

# Synergetic Segmentation and Registration of TLS Point Clouds using Geometric and Radiometric Information

vorgelegt von  
M. Sc.  
Mathias Burger

von der Fakultät VI – Planen Bauen Umwelt  
der Technischen Universität Berlin  
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften  
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Martin Kada  
Gutachter: Prof. Dr.-Ing. Frank Neitzel  
Gutachter: Prof. Dr.-Ing. Ingo Neumann  
Gutachter: Prof. Dr. habil. Hans-Gerd Maas

Tag der wissenschaftlichen Aussprache: 20. Januar 2022

Berlin 2022



MATHIAS BURGER

# Synergetic Segmentation and Registration of TLS Point Clouds using Geometric and Radiometric Information



# Synergetic Segmentation and Registration of TLS Point Clouds using Geometric and Radiometric Information

## ABSTRACT

The separation between regions of certain homogeneity of a data set in 2D or 3D is termed segmentation. Registration or referencing, on the other hand, denotes the process of transferring different data sets into a common reference frame or system.

Since the appearance of Terrestrial Laser Scanning (TLS) – combined with quasi-area-based acquisition methods and the associated enormous amount of data in the form of so-called *point clouds* – in engineering geodesy and related fields such as architecture and civil engineering, segmentation as well as registration have been among the most important processing steps with regard to an automated evaluation and further processing of *point clouds*.

Several procedures have been established, however, each of them representing independent and detached strategies in their solution of the individual process steps, whose disadvantages are mostly expressed on the segmentation side in incompleteness, the unilateral use of information or inefficiency as well as on the registration side by the lack of a suitable *stochastic model*.

This thesis first of all emphasizes the importance of segmentation and registration, as well as the necessity to use an adequate *stochastic model* in the context of Terrestrial Laser Scanning. As a result of the discussed disadvantages of existing methods, the motivation points as well as the prerequisites for the following chapters result.

A decisive component in any evaluation with measured quantities, also in association with segmentation and registration procedures, is the choice of a suitable individual weighting of the observations via the knowledge of the respective precision. Up to now, the precision-limiting factors influencing the measuring components in TLS could not be mapped in an all-embracing model. Consequently, a new *stochastic model* addressing this issue is presented.

A recent look on the topic of segmentation with TLS mostly shows the less effective application to a 3D data structure in combination with already referenced *point clouds* and the monolateral use of solely 3D information. Therefore, a new segmentation method is presented that addresses these problems and guarantees almost complete segmentation results based on the natural data structure of a single scan with proven and very efficient image processing routines. The examples show the versatile applicability of the segmentation algorithm using both 3D and intensity information on urban and natural object structures in the individual *point clouds*.

In the last chapter the loop closes. A unique synergetic segmentation and registration procedure is presented, which uses the information gain from both process steps mutually. Based on extensive segments of the segmentation of single scans, plane sub-segments are detected with a special sub-segmentation procedure, incorporating the derived *stochastic model*. By means of a plane-based matching procedure, approximate values are initially derived for a subsequent comprehensive registration procedure of the individual scans. Finally, the *point clouds* of the single scans can be transferred into a common coordinate system using the external transformation parameters determined from the registration step. Simultaneously, the existing matching information of corresponding planes can be used to complement segments that are not completely captured due to different scanning perspectives on the object. Thus, the result of this processing chain is a complete registered and segmented 3D point representation of an object previously acquired with a TLS from different perspectives.

## Synergetic Segmentation and Registration of TLS Point Clouds using Geometric and Radiometric Information

### KURZFASSUNG

Die Abgrenzung von Bereichen gewisser Homogenität eines Datensatzes im 2D oder 3D wird als sogenannte Segmentierung bezeichnet. Als Registrierung oder Referenzierung versteht man hingegen eine Überführung der verschiedenen Datensätze in ein gemeinsames Bezugs- oder auch Referenzsystem.

Seit dem Einzug des Terrestrischen Laserscannings (TLS) – verbunden mit quasi-flächenhaften Erfassungsmethoden und der damit einhergehenden enormen Datenmenge in Form von sog. *Punktwolken* – in die Ingenieurgeodäsie und verwandte Fachbereiche wie u. a. Architektur und das Bauingenieurwesen, zählen die Segmentierung als auch die Registrierung zu den wesentlichen Prozessierungsschritten im Hinblick auf eine automatisierte Auswertung und Weiterverarbeitung von *Punktwolken*.

Etabliert haben sich einige Verfahren, die aber jeweils in ihrer Lösung der einzelnen Prozessschritte voneinander unabhängige und losgelöste Strategien darstellen, deren Nachteile sich auf der Segmentierungsseite zumeist in Unvollständigkeit, dem einseitigen Nutzen von Informationen oder Ineffizienz sowie auf Registrierungsseite durch Nichtvorhandensein oder Fehlen eines geeigneten *stochastischen Modells* äußern.

Die vorliegende Arbeit stellt zunächst die Wichtigkeit der Segmentierung und Registrierung, als auch die damit einhergehende Notwendigkeit zur Nutzung eines adäquaten *stochastischen Modells*, im Kontext des TLS heraus. Aus den andiskutierten Nachteilen bestehender Verfahren ergeben sich die Motivationspunkte sowie die Voraussetzungen für die nachfolgenden Kapitel.

Eine maßgebliche Komponente bei jeglichen Auswertungen mit gemessenen Größen, auch in Verbindung mit Segmentierungs- und Registrierungsverfahren, ist die Wahl von einer geeigneten individuellen Gewichtung der Beobachtungen über die Kenntnis der jeweiligen Präzision. Bislang konnten die präzisionslimitierenden Einflussfaktoren auf die Messkomponenten beim TLS nicht in einem allumfänglichen Modell abgebildet werden, wodurch ein neues *stochastisches Modell* vorgestellt wird, das sich diesem Sachverhalt widmet.

Ein aktueller Blick zum Thema Segmentierung mit TLS zeigt zumeist die weniger effektive Anwendung auf eine 3D-Datenstruktur in Verbindung mit bereits referenzierten *Punktwolken* und der einseitigen Nutzung ausschließlicher 3D-Informationen. Daher wird ein neuartiges Segmentierungsverfahren vorgestellt das sich den Problemen annimmt und auf Basis der durch die Scanentstehung natürlichen Datenstruktur eines Einzelscans mit bewährten und sehr effizienten Bildverarbeitungsroutinen nahezu

vollständige Segmentierungsergebnisse gewährleistet. Die Beispiele zeigen die vielseitige Anwendbarkeit des Segmentierungsalgorithmus unter Nutzung von 3D- als auch Intensitätsinformationen auf urbane als auch natürliche Objektstrukturen in den einzelnen *Punktwolken*.

Im letzten Kapitel schließt sich der Kreis. Es wird ein einzigartiges synergetisches Segmentierungs- und Registrierungsverfahren vorgestellt, welches den Informationsgewinn aus beiden Prozessschritten wechselseitig ausnutzt. Basierend auf flächenhaften Segmenten der Segmentierung von Einzelscans werden ebene Untersegmente unter Einbeziehung des abgeleiteten *stochastischen Modells* mit einem speziellen Subsegmentierungsverfahren detektiert. Über ein ebenenbasiertes Matchingverfahren werden zunächst Näherungswerte für ein sich anschließendes allumfassendes Registrierungsverfahren der Einzelscans generiert. Schließlich können die *Punktwolken* der Einzelscans über die aus dem Registrierungsschritt bestimmten äußeren Transformationsparameter in ein gemeinsames Koordinatensystem überführt werden und zugleich über die vorhandene Matchinginformation korrespondierender Ebenen Segmente vervollständigt werden, die durch unterschiedliche Scanperspektive auf das Objekt nicht vollständig erfasst werden. Somit ist das Resultat dieser Verarbeitungskette eine vollständige registrierte und segmentierte 3D-Punktrepräsentation eines zuvor aus unterschiedlichen Perspektiven mit einem TLS aufgenommenen Objektes von Interesse.

# Publications

The presented thesis stems from various ideas and illustrations that have appeared previously in the following publications:

- [17] M. BURGER, D. WUJANZ, and F. NEITZEL: “Segmentierung von Punktwolken anhand von geometrischen und radiometrischen Informationen”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2017, pp. 116–128.
- [18] M. BURGER, D. WUJANZ, and F. NEITZEL: “Synergetische Segmentierung und Registrierung von Punktwolken”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2018, pp. 130–144.
- [130] D. WUJANZ, M. BURGER, M. METTENLEITER, and F. NEITZEL: “An intensity-based stochastic model for terrestrial laser scanners”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 125 (2017), pp. 146–155.
- [131] D. WUJANZ, M. BURGER, M. METTENLEITER, and F. NEITZEL: “Modellbasierte Standpunktplanung für terrestrische Laserscanner unter Nutzung eines intensitätsbasierten stochastischen Modells”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2016, pp. 60–71.
- [132] D. WUJANZ, M. BURGER, M. METTENLEITER, F. NEITZEL, F. TSCHIRSCHWITZ, and T. KERSTEN: “Ein intensitätsbasiertes stochastisches Modell für terrestrische Laserscanner - Erste Untersuchungen der Z+ F IMAGER 5006h und 5010”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2017, pp. 16–26.
- [133] D. WUJANZ, M. BURGER, F. TSCHIRSCHWITZ, T. NIETZSCHMANN, F. NEITZEL, and T. KERSTEN: “Bestimmung von intensitätsbasierten stochastischen Modellen für terrestrische Laserscanner basierend auf 3D-Punktwolken”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2018, pp. 155–166.
- [134] D. WUJANZ, M. BURGER, F. TSCHIRSCHWITZ, T. NIETZSCHMANN, F. NEITZEL, and T. P. KERSTEN: “Determination of intensity-based stochastic models for terrestrial laser scanners utilising 3D-point clouds”. In: *Sensors* 18.7 (2018), p. 2187.
- [136] D. WUJANZ, M. METTENLEITER, M. BURGER, and F. NEITZEL: “Viewpoint planning for terrestrial laser scanning utilising an intensity based stochastic model”. In: *3rd Joint International Symposium on Deformation Monitoring (JISDM)*. 2016.



# Contents

	PAGE
<b>o</b> INTRODUCTION	<b>I</b>
o.1 Processing and Analysis of TLS Measurements . . . . .	2
o.1.1 Current Evaluation Strategy . . . . .	4
o.2 Quality in the context of TLS . . . . .	5
o.2.1 Quality models for TLS . . . . .	7
o.2.1.1 Stochastic Modelling . . . . .	9
o.3 Aim and Scope of Research . . . . .	9
o.4 Thesis Organisation . . . . .	10
o.4.1 Main Structure of the Thesis . . . . .	11
o.4.2 Implementation of Functions . . . . .	12
<b>I</b> FUNDAMENTALS AND RELATED WORK	<b>13</b>
I.1 Methods of 3D-Data Acquisition . . . . .	13
I.2 Laser scanning . . . . .	14
I.2.1 Terrestrial Laserscanner – Scanning Principle and Measurements . . . . .	15
I.2.2 Electro-optical Distance Measurement (EDM) – Principles and Differences	18
I.2.2.1 Comparison of AMCW with TOF method . . . . .	23
I.2.3 Principle of Angular Measurement . . . . .	24
I.2.4 Error Influences on Measurements . . . . .	25
I.2.4.1 Influencing Factors on Distance Measurement . . . . .	26
I.2.4.2 Influencing Factors on Intensity Measurement . . . . .	30
I.2.4.3 Influencing Factors on Angle Measurement . . . . .	34
I.2.5 Calibration of Terrestrial Laser Scanners (TLSs) . . . . .	36
I.2.6 Interaction of Measurement Resolution and -Quality . . . . .	36
I.3 Coordinate Systems in the context of TLS . . . . .	37
I.3.1 Outer Orientation in the context of TLS . . . . .	39
I.4 Spatial Data Representation and Indexing . . . . .	39
I.5 Graph Theory and Connected Component Labeling . . . . .	42
I.6 Representation of Rotations in 3D space . . . . .	46
I.6.1 Parameterization using Euler Angles . . . . .	48
I.6.2 Parameterization using Quaternions . . . . .	49
I.7 Classical 3D-Helmert Transformation . . . . .	50
I.8 Concepts of Adjustment Theory . . . . .	52
I.8.1 Adjustment Problem . . . . .	52
I.8.2 The Functional Model . . . . .	52
I.8.2.1 Linearisation . . . . .	54

1.8.3	The Stochastic Model . . . . .	56
1.8.4	Least Squares Method ( <b>LSM</b> ) . . . . .	57
1.8.5	The Gauß Markov Model ( <b>GMM</b> ) . . . . .	59
1.8.6	The Gauß Helmert Model ( <b>GHM</b> ) . . . . .	60
1.8.7	The Gauß Helmert Model ( <b>GHM</b> ) with Constraints . . . . .	64
1.8.8	Detection and Elimination of Outliers in an Observation Series . . . . .	67
1.8.8.1	Overall Model Test . . . . .	67
1.8.8.2	Individual Test of Observations . . . . .	68
1.8.9	Quality Assessment . . . . .	68
1.8.9.1	Accuracy measures . . . . .	69
1.8.9.2	Reliability measures . . . . .	69
1.9	On Stochastic Modelling of TLS Measurements . . . . .	70
1.10	On the Segmentation of TLS Point Clouds . . . . .	71
1.11	On the Registration of TLS Point Clouds . . . . .	74
<b>2</b>	<b>A COMPREHENSIVE STOCHASTIC MODEL FOR TLS</b> . . . . .	<b>77</b>
2.1	Intensity Behaviour – SNR and Precision of Distance . . . . .	79
2.1.1	Experimental Arrangement and Data Acquisition . . . . .	79
2.1.2	Influence of Object Distance and Radiometry to derive Distance Noise . . . . .	82
2.2	Further Factors Influencing the Distance Stochastics . . . . .	84
2.2.1	Impact of the Incidence Angle . . . . .	84
2.2.2	Effects caused by different Materials . . . . .	85
2.3	The Stochastic Model of a TLS . . . . .	89
2.3.1	Derivation of a Stochastic Model for Point Clouds . . . . .	93
2.4	Validation of the Stochastic Model . . . . .	95
2.4.1	Interpretation of the Adjustment Result as Verification . . . . .	95
2.4.2	Validation based on Planar Surfaces . . . . .	96
2.4.3	Validation using Arbitrary Surfaces . . . . .	103
2.5	Deriving Stochastic Models based on 3D-Measuring Mode . . . . .	104
2.6	Validity for pulsed <b>TLS</b> . . . . .	106
2.7	Conclusion . . . . .	107
2.7.1	Acceptance and Fields of Application . . . . .	108
<b>3</b>	<b>SEGMENTATION BASED ON GEOMETRIC AND RADIOMETRIC INFORMATION</b> . . . . .	<b>109</b>
3.1	Sample Datasets . . . . .	110
3.2	Data Preprocessing for Information Retrieval . . . . .	112
3.2.1	Local Neighbourhood Analysis and Convolution . . . . .	112
3.2.2	Describing Local Changes within the Data . . . . .	115
3.2.3	Efficient Calculation of Point Normals . . . . .	118
3.3	New Segmentation Algorithm for TLS Data . . . . .	123
3.3.1	Pixel-based Segmentation . . . . .	125
3.3.2	Region-based Segmentation . . . . .	128
3.3.3	Overall Segmentation and Region Analysis . . . . .	131

3.4	Segmentation Results . . . . .	134
3.4.1	Influence of Homogeneity Criteria . . . . .	137
3.4.2	Performance Evaluation . . . . .	138
3.5	Conclusion . . . . .	139
<b>4</b>	<b>SYNERGETIC SEGMENTATION AND REGISTRATION</b>	<b>141</b>
4.1	Sample Dataset . . . . .	142
4.2	Synergetic Segmentation and Registration of TLS Data . . . . .	143
4.3	Segmentation Process . . . . .	144
4.3.1	Pre-segmentation of individual Scans . . . . .	144
4.3.2	Sub-Segmentation into Planar Regions . . . . .	145
4.4	Matching Process . . . . .	152
4.5	Registration Process . . . . .	164
4.6	Assignment process to entire segments . . . . .	174
4.7	Results . . . . .	178
4.8	Conclusion . . . . .	183
<b>5</b>	<b>CONCLUSION AND OUTLOOK</b>	<b>185</b>
5.1	Contribution to Science and Engineering . . . . .	186
5.2	Ideas for Improvement and Research Perspective . . . . .	187
<b>APPENDIX A COMPLEMENTARY INFORMATION</b>		<b>189</b>
A.1	Basic Mathematics . . . . .	189
A.1.1	Rodrigues Formulation and Quaternions . . . . .	189
A.1.2	Interrelation between Cartesian and Polar Coordinates . . . . .	192
A.1.3	Point Normal calculation within gridded data using SVD . . . . .	193
A.2	Transformation of arbitrary surfaces . . . . .	194
A.3	Basic Matlab Code . . . . .	197
A.3.1	General . . . . .	197
A.3.2	Precalculation . . . . .	198
A.3.3	Connected Component Labeling (CCL) . . . . .	199
A.3.4	Stochastic Model . . . . .	201
A.3.5	Segmentation . . . . .	202
A.3.6	Matching . . . . .	208
A.3.7	Registration . . . . .	209
<b>REFERENCES</b>		<b>211</b>



# Listing of figures

		PAGE
Figure 1	Evaluation strategy and processing chain for TLS . . . . .	2
Figure 2	Relationship between accuracy, precision and correctness of measured values according to (SCHWARZ 1995) and (WITTE and SPARLA 2015, p. 69) . . . . .	5
	A    High precision but low accuracy . . . . .	5
	B    Ideal case of precision and accuracy . . . . .	5
Figure 3	Structure and dependencies of chapters . . . . .	II
Figure 1.1	Active 3D acquisition methods based on (GÜHRING 2002, p. 14) . . . . .	14
Figure 1.2	Basic components of a TLS based on (METTENLEITER et al. 2015, p. 14) . . . . .	16
Figure 1.3	Creation of a grid by incremental scanning of the TLS . . . . .	17
Figure 1.4	Principle of pulse method . . . . .	19
Figure 1.5	Principle of laser triangulation . . . . .	20
Figure 1.6	Amplitude modulation of a carrier wave . . . . .	21
Figure 1.7	Principle of phase difference method . . . . .	22
Figure 1.8	Schematic representation of an angle encoder for according to (METTENLEITER et al. 2015) . . . . .	25
Figure 1.9	General distribution of the distance noise $\sigma_\rho$ for the mean distance $\rho_m$ . . . . .	30
Figure 1.10	Interaction of the factors influencing the intensity with respect to the radar equation . . . . .	31
Figure 1.11	Intensity behaviour of the TLS Z+F IMAGER <sup>®</sup> 5006h as a function of the measured distance $\rho$ for different panels, each with different reflectivity. . . . .	32
Figure 1.12	Intensity behaviour of the TLS Z+F IMAGER <sup>®</sup> 5006h as a function of the incidence angle $\alpha$ at different distances. . . . .	33
Figure 1.13	Axis errors for a TLS based on NEITZEL (2006) . . . . .	35
	A    Tilting axis error . . . . .	35
	B    Collimation axis error . . . . .	35
	C    Coll. axis eccentricity . . . . .	35
Figure 1.14	Local coordinate system of a TLS . . . . .	38
Figure 1.15	Pixel coordinate system after scanning of a TLS . . . . .	38
Figure 1.16	Outer orientation of TLS standpoints with respect to the superordinate system . . . . .	39
Figure 1.17	Formation of an octree structure for unstructured data . . . . .	40
	A    3D Voxels of octree . . . . .	40
	B    Octree branching structure . . . . .	40
Figure 1.18	Information of the TLS represented in different Data Layers . . . . .	41
Figure 1.19	Representation of a point cloud structure array in MatLab . . . . .	41
Figure 1.20	From a greyscale image via a binary representation to a graph representation . . . . .	43

	A	Greyscale image . . . . .	43
	B	Binary image . . . . .	43
	C	Graph of <b>CC</b> . . . . .	43
Figure 1.21		Adjacency matrix and visualization from the direct connections of a binary matrix . . . . .	44
	A	Adjacency matrix . . . . .	44
	B	Connection of nodes . . . . .	44
Figure 1.22		Principle of spatial similarity transformation . . . . .	51
Figure 1.23		Linearisation of a function according to NIEMEIER (2008) p. 122 . . . . .	54
Figure 1.24		Adjustment models according to the <b>LSM</b> . . . . .	58
Figure 2.1		Imprinted ALUCORE <sup>®</sup> panels with different radiometric properties (grey values varying from 0-255) . . . . .	79
	A	Panel 1 l . . . . .	79
	B	Panel 1 d . . . . .	79
	C	Panel 2 l . . . . .	79
	D	Panel 2 d . . . . .	79
	E	Panel 3 l . . . . .	79
	F	Panel 3 d . . . . .	79
	G	Panel 4 l . . . . .	79
	H	Panel 4 d . . . . .	79
Figure 2.2		Standard measurement setup for the derivation of the stochastic model . .	80
Figure 2.3		Intensity behaviour of the Avalanche Photo Diode ( <b>APD</b> ) for the <b>TLS Z+F IMAGER<sup>®</sup> 5006h</b> . . . . .	80
	A	Intensity versus noise . . . . .	80
	B	Intensity versus SNR . . . . .	80
Figure 2.4		Histograms of the intensity measurements for the panel 4 l at different distances resulting from the maximum sampling rate of 1016 kHz, see also <a href="#">Section 1.2.6</a> . . . . .	81
	A	Intensity histogram Panel 4 l at 5 m . . . . .	81
	B	Intensity histogram Panel 4 l at 15 m . . . . .	81
	C	Intensity histogram Panel 4 l at 25 m . . . . .	81
	D	Intensity histogram Panel 4 l at 35 m . . . . .	81
Figure 2.5		Histograms of the distance measurements for the panel 4 l at different distances corresponding to the intensity histograms shown in <a href="#">Figure 2.4</a> . . .	82
	A	Distance histogram Panel 4 l at 5 m . . . . .	82
	B	Distance histogram Panel 4 l at 15 m . . . . .	82
	C	Distance histogram Panel 4 l at 25 m . . . . .	82
	D	Distance histogram Panel 4 l at 35 m . . . . .	82
Figure 2.6		Precision of the distance measurement $\sigma_\rho$ as a function of the mean intensity $\bar{A}$ for the sampling rates of 1016 kHz (blue dots), 508 kHz (red dots) and 127 kHz (black dots) for the <b>TLS Z+F IMAGER<sup>®</sup> 5006h</b> . . . . .	83

Figure 2.7	Color-coded classification of the influences of object distance $\rho$ and radiometry on the precision of distance measurements $\sigma_\rho$ at 127 kHz . . . . .	84
Figure 2.8	Influence of the incidence angle $\alpha$ onto the distance precision $\sigma_\rho$ for two different object distances . . . . .	85
	A Acrylic wall plaster . . . . .	86
	B Beech Panel . . . . .	86
	C Pumice . . . . .	86
	D Red clinker . . . . .	86
	E Resin coated plywood . . . . .	86
	F Styrofoam . . . . .	86
	G White silicate plaster . . . . .	86
Figure 2.10	Spectral reflectance curves for different materials of a Perkin Elmer Lambda 19 spectrometer covering the wavelength range 350-2500 nm . . . . .	87
Figure 2.11	Influence of material samples from <a href="#">Figure 2.9</a> under varying acquisition configurations captured at 127 kHz . . . . .	88
Figure 2.12	Effect of roughness and smoothness for different materials under differing acquisition configurations captured at 127 kHz . . . . .	89
Figure 2.13	Characteristic intensity range to describe the distance noise for a fictitious <i>laser scanner</i> . . . . .	90
Figure 2.14	Regression curve through the data points with respect to the sampling rate of 508 kHz . . . . .	93
Figure 2.15	Representation of precision in 3D for the object presented in <a href="#">Figure 3.1A</a> . . . . .	95
Figure 2.16	Arrangement of the panels for verification of the stochastic model for test scenario A, characterized by red rectangles (left), and interchange of the panels (right) . . . . .	96
	A Acquisition configuration . . . . .	96
	B Replacement of panels . . . . .	96
Figure 2.17	Intensity behaviour of the <a href="#">TLS Z+F IMAGER<sup>®</sup> 5006h</a> as a function of the incidence angle $\alpha$ at different distances. . . . .	103
Figure 2.18	Comparison of the reference stochastic model based on an sampling rate of 508 kHz with 3D residual procedure . . . . .	105
Figure 2.19	Comparison of the reference stochastic model based on an sampling rate of 508 kHz with "quasi range" procedure . . . . .	106
Figure 2.20	Stochastic model of the MTA 1 zone of a Riegl VZ-400i laser scanner . . . . .	107
Figure 3.1	Front view of the test objects Belvedere (left), Orangery (center) and landslide (right) . . . . .	110
	A Teahouse Belvedere . . . . .	110
	B Orangery Potsdam . . . . .	110
	C Landslide . . . . .	110
Figure 3.2	Geometry layers <b>X</b> , <b>Y</b> and <b>Z</b> related to the standpoint in <a href="#">Figure 3.1A</a> . . . . .	111
	A Geometry layer <b>X</b> . . . . .	111
	B Geometry layer <b>Y</b> . . . . .	111

	c	Geometry layer $\mathbf{Z}$ . . . . .	III
Figure 3.3		Intensity layer $\mathbf{I}$ and corresponding precision layer $\Sigma_{XYZ}$ related to the standpoint in <a href="#">Figure 3.1A</a> . . . . .	112
	A	Intensity layer $\mathbf{I}$ . . . . .	112
	B	Precision layer $\Sigma_{XYZ}$ . . . . .	112
Figure 3.4		Convolution of a data matrix $\mathbf{M}$ with a kernel $\mathbf{K}$ . . . . .	114
Figure 3.5		Kernel $\mathbf{K}$ with size $3 \times 3$ at position $(i, j)$ in matrix $\mathbf{M}$ . . . . .	115
Figure 3.6		Computed information layers regarding curvature, point spacing and change in intensity . . . . .	117
	A	Point density layer $\mathbf{D}$ . . . . .	117
	B	Curvature layer $\mathbf{C}$ . . . . .	117
	C	Intensity change layer $\Delta\mathbf{I}$ . . . . .	117
Figure 3.7		Calculation of point normals within the matrix structure of coordinates $\mathbf{X}$ , $\mathbf{Y}$ and $\mathbf{Z}$ . . . . .	119
Figure 3.8		Computed layers regarding the normal vector components $\mathbf{n}_x$ , $\mathbf{n}_y$ and $\mathbf{n}_z$ . . . . .	120
	A	Normal layer $\mathbf{n}_x$ . . . . .	120
	B	Normal layer $\mathbf{n}_y$ . . . . .	120
	C	Normal layer $\mathbf{n}_z$ . . . . .	120
Figure 3.9		Regions for the evaluation of methods for computation of point normals . . . . .	122
Figure 3.10		Process diagram of the segmentation algorithm . . . . .	124
Figure 3.11		Results of pixel-based segmentation (binary matrix left) and region-based segmentation (color-coded regions right) . . . . .	125
	A	Binary representation . . . . .	125
	B	Segmented cloud . . . . .	125
Figure 3.12		Computed information layers regarding curvature, point spacing and change in intensity . . . . .	126
	A	Density histogram $\mathbf{D}$ . . . . .	126
	B	Curvature histogram $\mathbf{C}$ . . . . .	126
	C	Intensity change histogram $\Delta\mathbf{I}$ . . . . .	126
Figure 3.13		Run Length Encoding ( <a href="#">RLE</a> ) in the context of Connected Component ( <a href="#">CC</a> ) Labeling for a binary matrix . . . . .	128
	A	Binary matrix extract . . . . .	128
	B	Runs of binary matrix . . . . .	128
	C	Labeled runs . . . . .	128
Figure 3.14		<i>Graph</i> representations of the connectivity for the individual runs in <a href="#">Figure 3.13B</a> . . . . .	129
	A	Graph region 1 . . . . .	129
	B	Graph region 2 . . . . .	129
	C	Graph region 3 . . . . .	129
	D	Graph region 4 . . . . .	129
Figure 3.15		Representation of a regions structure array in <a href="#">MatLab</a> . . . . .	133
Figure 3.16		Results of the region-based segmentation (color-coded regions) of the front view of the test objects Belvedere (left) and inner dome in the outer area of the Orangery (right) . . . . .	134

	A	Teahouse Belvedere segmented . . . . .	134
	B	Orangery Potsdam segmented . . . . .	134
Figure 3.17		Detailed view from the overall result in <a href="#">Figure 3.16A</a> of the region-based segmentation (color-coded regions) of the front view of the test object Belvedere	135
	A	Belvedere detail 1 . . . . .	135
	B	Belvedere detail 2 . . . . .	135
	C	Belvedere detail 3 . . . . .	135
Figure 3.18		Detailed view from the overall result in <a href="#">Figure 3.16B</a> of the region-based segmentation (color-coded regions) of the inner dome in the outer area of the Orangery . . . . .	135
	A	Orangery detail 1 . . . . .	135
	B	Orangery detail 2 and 3 . . . . .	135
	C	Orangery detail 4 . . . . .	135
Figure 3.19		Results of the region-based segmentation (color-coded areas) of a small area of the landslide in Obergurgel . . . . .	136
Figure 3.20		Upper part of the segmented red pillar in <a href="#">Figure 3.11B</a> right without point spacing information (left pillar) and with (right pillar) . . . . .	138
	A	Homog. criterion $H_C$ . . . . .	138
	B	Homog. criterion $H_C \wedge H_D$ . . . . .	138
Figure 4.1		Overview of the standpoints (SP <sub>1-3</sub> ) and acquisition areas (green, red and blue delimited) that led to the data A, B and C in <a href="#">Figure 4.2</a> , in relation to the object. . . . .	142
Figure 4.2		Views onto the object, respectively data slices, emerging from the different positions in <a href="#">Figure 4.1</a> . . . . .	143
	A	View Standpoint 2 . . . . .	143
	B	View Standpoint 1 . . . . .	143
	C	View Standpoint 10 . . . . .	143
Figure 4.3		Process diagram of the synergetic segmentation and registration algorithm	143
Figure 4.4		Presegmented regions (color-coded) based on the region-based segmentation in <a href="#">Chapter 3</a> for the datasets A, B and C in <a href="#">Figure 4.2</a> of the test object Belvedere	144
	A	View Standpoint 2 . . . . .	144
	B	View Standpoint 1 . . . . .	144
	C	View Standpoint 10 . . . . .	144
Figure 4.5		Detected planes from sub-segmentation on the basis of the pre-segmented <i>point clouds</i> , i. e. datasets A, B and C in <a href="#">Figure 4.4</a> . . . . .	145
	A	View Standpoint 2 . . . . .	145
	B	View Standpoint 1 . . . . .	145
	C	View Standpoint 10 . . . . .	145
Figure 4.6		Computed information layers regarding curvature, point spacing and change in intensity . . . . .	151
	A	Initial region . . . . .	151
	B	Iteration 1 . . . . .	151

	C	Iteration 2 . . . . .	151
	D	Iteration 3 . . . . .	151
	E	Iteration 4 . . . . .	151
	F	Iteration 5 . . . . .	151
Figure 4.7		Representation of a regions structure array in <a href="#">MatLab</a> . . . . .	151
Figure 4.8		Result of plane matching with respect to standpoint pairs 2 with 1 (upper left image pair), 10 with 1 (lower right image pair) and 10 with 2 (upper right and lower left image) . . . . .	161
	A	Match 2 to 1 . . . . .	161
	B	Match 1 to 2 . . . . .	161
	C	Match 10 to 2 . . . . .	161
	D	Match 2 to 10 . . . . .	161
	E	Match 1 to 10 . . . . .	161
	F	Match 10 to 1 . . . . .	161
Figure 4.9		Corresponding plane pair $k$ in <i>point clouds</i> $i$ and $j$ in relation to a superordinate coordinate system . . . . .	164
Figure 4.10		Assignment of individual segments to entire segments based on matching of subsegments . . . . .	175
	A	Matching regions in scan 2 . . . . .	175
	B	Matching regions in scan 1 . . . . .	175
	C	Matching regions in scan 10 . . . . .	175
	D	Subregion matches in scan 2 . . . . .	175
	E	Subregion matches in scan 1 . . . . .	175
	F	Subregion matches in scan 10 . . . . .	175
Figure 4.11		Detailed views of the region parts of a surface patch captured from scan station 1 and 10 and its combination to an entire region . . . . .	177
	A	Main regions from scan station 1 . . . . .	177
	B	Combined regions from station 1 and 10 . . . . .	177
Figure 4.12		Horizontal cross sections to validate the registration result . . . . .	178
Figure 4.13		Individual cuts from the registered <i>point cloud</i> at different heights . . . . .	179
	A	Cut at $z=-0.5$ m . . . . .	179
	B	Cut at $z=8.0$ m . . . . .	179
	C	Cut at $z=13$ m . . . . .	179
Figure 4.14		Assigned segments (plane in orange and pillar in red) for the <i>point clouds</i> of the positions 2 (left), 1 (middle) and 10 (right) . . . . .	180
	A	Assignment standpoint 2 . . . . .	180
	B	Assignment standpoint 1 . . . . .	180
	C	Assignment standpoint 10 . . . . .	180
Figure 4.15		Detailed views (front view and top view) of the points of the planar region (depicted in the respective standpoint color based on <a href="#">Figure 4.1</a> ) merged from the three perspectives (depicted in orange) in <a href="#">Figures 4.14A</a> to <a href="#">4.14C</a> . . . . .	180
	A	Plane front view . . . . .	180
	B	Plane top view . . . . .	180

Figure 4.16	Detailed views (front view and top view) of the points of the pillar (depicted in the respective standpoint color based on <a href="#">Figure 4.1</a> ) merged from the three perspectives (depicted in red) in <a href="#">Figures 4.14A to 4.14C</a> . . . . .	181
A	Pillar front view . . . . .	181
B	Pillar top view . . . . .	181
Figure A.1	<i>Rodrigues</i> rotation around a unit vector $\mathbf{u}$ . . . . .	190
Figure A.2	Mathematical relationship between cartesian and polar coordinates . . . . .	192



# Listing of tables

		PAGE
Table 1	Quality model for <a href="#">TLS</a> , acquisition and <i>point cloud</i> . . . . .	7
Table 2	Quality models for <a href="#">TLS</a> 's evaluation steps segmentation and registration . .	8
Table 1.1	Comparison of different methods for <a href="#">EDM</a> with regard to their performance data according to ( <a href="#">METTENLEITER et al. 2015</a> , p. 9) . . . . .	18
Table 1.2	Comparison of <a href="#">TOF</a> and <a href="#">AMCW</a> methods according to ( <a href="#">WEHR and LOHR 1999</a> )	24
Table 1.3	Combination of various measuring rates with angular resolutions for a <a href="#">TLS Z+F IMAGER® 5006h</a> (based on <a href="#">METTENLEITER et al. 2015</a> , p. 49) . . . . .	37
Table 2.1	Estimated stochastic model parameters for different sampling rates of the <a href="#">TLS Z+F IMAGER® 5006h</a> . . . . .	93
Table 2.2	Empirical standard deviation $s_0$ after plane adjustment and mean intensity $I_m$ for different scenarios $A_i$ of the measuring arrangement $A$ . . . . .	100
Table 2.3	Empirical $SDs$ $s_0$ based on different unrealistic <i>stochastic models</i> for scenario $A_4$ from <a href="#">Table 2.2</a> . . . . .	101
Table 2.4	Empirical standard deviation $s_0$ after plane adjustment and mean intensity $I_m$ for different sampling rates of scenario $A_4$ from <a href="#">Table 2.2</a> . . . . .	102
Table 3.1	Comparison of different methods for normal calculation on a raster data structure . . . . .	123
Table 3.2	Segmentation parameters used to evaluate the various example data sets . .	137
Table 3.3	Criteria for various processed data sets to evaluate the segmentation algorithm	138
Table 4.1	Results for various processed data sets to evaluate the segmentation algorithm	146
Table 4.2	Listing of initial transformation parameters . . . . .	155
Table 4.3	Values and Standard Deviations ( $SDs$ ) for the <i>relative transformation parameters</i> of the scanner standpoint pairs after the matching process of <a href="#">Section 4.4</a>	162
Table 4.4	Values and Standard Deviations ( $SDs$ ) for the <i>exterior orientations</i> of the individual scanner standpoints after the registration process of <a href="#">Section 4.5</a> .	182



# Listing of code

	PAGE
Code 1.1	Filling a point cloud structure in MatLab . . . . . 42
Code 1.2	Labeling connected components of a graph . . . . . 44
Code 3.1	Conversion of <b>TLS</b> data based on polar acquisition in a matrix structure . . . . . III
Code 3.2	Region-based calculation of standard deviations for a matrix <b>M</b> using an modified convolution approach . . . . . 116
Code 3.3	Efficient point normal calculation by means of filtering under consideration of the point spacing . . . . . 121
Code 3.4	Pixel-based segmentation with all information layers . . . . . 127
Code 3.5	Labeling connected components of a graph consisting of run lengths . . . . . 130
Code 3.6	Segmenting of regions based on homogeneity criteria . . . . . 132
Code 3.7	Access to the 3D coordinates of a specific region in the point cloud . . . . . 134
Code 4.1	Definition of a planar region . . . . . 148
Code 4.2	Region splitting and merging . . . . . 149
Code 4.3	Determine initial rotation for matching planes . . . . . 154
Code 4.4	Determine initial translation for matching planes . . . . . 155
Code 4.5	Matching of planes by known datum in case of transformation parameters . . . . . 156
Code 4.6	Matching of planes between the various adjacent and geometrically overlap- ping point clouds of individual scan pairs . . . . . 163
Code 4.7	Access the indices of planes corresponding to a specific region . . . . . 174
Code 4.8	Matching of regions from individual point clouds based on matching planes . . . . . 176
Code A.1	Transforming 3D points to another coordinate system . . . . . 197
Code A.2	Expanding a matrix <b>M</b> for filtering purposes . . . . . 197
Code A.3	Calculation of the point normals by means of filtering . . . . . 198
Code A.4	Calculation of the point curvature by means of filtering . . . . . 198
Code A.5	Calculation of the point density by means of filtering . . . . . 199
Code A.6	Labeling connected components by flood filling . . . . . 200
Code A.7	Run length encoding for connected components labeling . . . . . 200
Code A.8	Computes precision layers for a point cloud structure . . . . . 201
Code A.9	Decision of splitting a region . . . . . 202
Code A.10	Decision of merging two regions . . . . . 203
Code A.11	Splitting a region . . . . . 204
Code A.12	Merging regions . . . . . 205
Code A.13	Split and merge algorithm . . . . . 207
Code A.14	Extending Code for direct plane matching if same correspondences are present among different scan pairs . . . . . 208

Code A.15	Transform planes to another coordinate system . . . . .	209
Code A.16	Fitting a plane using SVD . . . . .	209

# Listing of algorithms

	PAGE
Algorithm 3.1    Filtering a matrix by a kernel. . . . .	114



# Acronyms

ALS	Airborne Laser Scanner . . . . .	15
AMCW	Amplitude Modulated Continuous Wave . . . . .	15
APD	Avalanche Photo Diode . . . . .	xiv
BFS	Breadth First Search . . . . .	43
BIM	Building Information Modeling	
CC	Connected Component . . . . .	xvi
CAD	Computer-Aided Design	
CCA	Connected Component Analysis . . . . .	42
CCL	Connected Component Labeling . . . . .	xi
CM	Cofactor Matrix . . . . .	57
CMM	Coordinate Measuring Machine	
CPU	Central Processor Unit . . . . .	16
CT	Computer Tomography	
CV	Computer Vision . . . . .	42
DFS	Depth First Search . . . . .	43
DIC	Digital Image Correlation . . . . .	13
DMD	Dulmage Mendelsohn Decomposition . . . . .	46
DVS	Destructive Volume Scanning	
EDM	Electro-optical Distance Measurement . . . . .	ix
EDMU	Electro-optic Distance Measurement Unit . . . . .	16
EVD	Eigen Value Decomposition . . . . .	194
FMCW	Frequency Modulated Continuous Wave . . . . .	23
FOV	Field Of View . . . . .	15
GHM	Gauß Helmert Model . . . . .	x
GMM	Gauß Markov Model . . . . .	x
GPS	Global Positioning System	
ICP	Iterative Closest Point . . . . .	75
IMU	Inertial Measuring Unit . . . . .	74
LSA	Least Squares Adjustment . . . . .	49

LSE	Least Squares Estimation . . . . .	62
LSM	Least Squares Method . . . . .	x
MatLab	Matrix Laboratory . . . . .	12
MLS	Mobile Laser Scanner . . . . .	15
MMS	Mobile Mapping System . . . . .	15
NDVI	Normalized Differenced Vegetation Index . . . . .	188
RLE	Run Length Encoding . . . . .	xvi
RMS	root mean square . . . . .	28
SD	Standard Deviation . . . . .	xxi
SNR	Signal-to-Noise Ratio . . . . .	23
SVD	Singular Value Decomposition . . . . .	123
TLS	Terrestrial Laser Scanner/Terrestrial Laser Scanning . . . . .	ix
TOF	Time Of Flight . . . . .	15
VCM	Variance-Covariance Matrix . . . . .	56
VCP	Variance-Covariance Propagation . . . . .	94
WLS	Weighted Least Squares . . . . .	61
1D	one dimensional . . . . .	71
2D	two dimensional . . . . .	71
3D	three dimensional . . . . .	3

# Symbols

## BASIC ARITHMETICS AND MATHEMATICAL OPERATORS

$\mathbb{R}^3$	Quantity of all real numbers in 3D space
$\mathbf{I}$	Identity matrix

## COORDINATE SYSTEMS AND TRANSFORMATIONS

$\mathbf{R}$	Rotation matrix in 3D space
$\mathbf{R}_\omega$	Rotation matrix with respect to the X-axis
$\mathbf{R}_\phi$	Rotation matrix with respect to the Y-axis
$\mathbf{R}_\kappa$	Rotation matrix with respect to the Z-axis
$\lambda$	Scale factor of transformation
$\mathbf{x}$	Vector in the start coordinate system
$\mathbf{X}$	Vector in the target coordinate system
$\mathbf{X}_0$	Translation vector from start to target system

## ADJUSTMENT CALCULUS AND STATISTICS

$\mathbf{l}$	Vector of observations
$\mathbf{v}$	Vector of residuals
$\mu_x$	Expected value for $x$
$\tilde{x}$	True value $x$
$\sigma_x$	Precision of $x$
$\sigma_0$	Theoretical reference standard deviation
$s_0$	Empirical reference standard deviation
$\hat{\mathbf{x}}$	Estimated vector of adjusted unknowns $\mathbf{x}$
$\Delta\hat{\mathbf{x}}$	Estimated vector of reduced unknowns, i. e. corrections for $\mathbf{x}$
$\mathbf{x}^0$	Initial vector for $\mathbf{x}$
$\Psi$	Condition vector
$\mathbf{w}$	Vector of misclosures
$\mathbf{k}$	Vector of correlates
$\gamma$	Constraint vector
$\Sigma_{ll}$	Variance covariance matrix of observations
$\mathbf{Q}_{ll}$	Cofactor matrix of observations
$\mathbf{P}$	Weight matrix of observations
$\mathbf{J}$	Jacobian matrix
$\mathbf{A}$	Coefficient matrix of linearized functional model

<b>C</b>	Constraint matrix
<b>B</b>	Coefficient matrix according to residuals
<b>N</b>	Normal matrix
$Q_{\hat{x}\hat{x}}$	Cofactor matrix of adjusted unknowns
$Q_{vv}$	Cofactor matrix of residuals

#### MEASUREMENTS AND INFLUENCING FACTORS

$\theta$	Tilting angle
$\phi$	Direction of rotation
$\rho$	Distance measurement
<b>I</b>	Intensity measurement
$\Delta\theta$	Angular resolution for direction $\theta$
$\Delta\phi$	Angular resolution for tilting angle $\phi$
$\Delta\rho$	Distance resolution
$\sigma_{\theta}$	Angular precision for $\theta$
$\sigma_{\phi}$	Angular precision for $\phi$
$\sigma_{\rho}$	Distance precision

DEDICATED TO MY FATHER.



# Acknowledgments

THE MAJORITY OF THE IDEAS regarding the focus of research in the present thesis, as well as investigations carried out, emerged within the scope of the research project "TLS++ – Multispectral Terrestrial Laser Scanning: development, integration, calibration", funded by the Federal Ministry for Economic Affairs and Energy (BMWi), funding code KF3315401WM4. For the always pleasant collaboration I would like to thank the cooperation partners, the Society for the Advancement of Applied Computer Science (GFaI) from Berlin and the Zoller + Fröhlich GmbH (Z+F) from Wangen / Allgäu.

First of all, I wish to express my gratitude to Prof. Dr. Frank Neitzel, who recruited me in 2015 to his Institute of Geodesy and Geoinformation Science (IGG) at Technische Universität Berlin (TUB) and gave me the opportunity to pursue a PhD in this great environment. As my supervisor and former boss, he not only has always an open ear for subject-related as well as private topics, but is constantly attentive to the advancement and personal development of his staff. I am just as grateful to him for his openness regarding any support of my ideas as I am for the numerous thematic suggestions and his honest criticism.

Moreover, my thanks go to Prof. Dr. habil. H.-G. Maas from the Institute of Photogrammetry and Remote Sensing at the Technische Universität Dresden (TUD), as well as Prof. Dr. I. Neumann from the Geodetic Institute (GIH) of the Leibniz University Hannover (LUH) for their acceptance and function as external reviewers of the doctoral committee and Prof. Dr. M. Kada for the chairmanship.

I am grateful for the really fantastic time with former colleagues and friends at the TUB. I enjoyed the scientific discussions, that sometimes got out of hand, as well as private conversations. I would like to mention in particular the sessions on adjustment calculation with S. Weisbrich and G. Malissiovas. On the one hand, I wish to thank Dr. D. Wujanz in a special way for his constant support in all matters, as well as for the great cooperation, also with regard to publications, not least for his help with proofreading and, above all, for his friendship.

Last but not least I would also like to thank my family. My parents Hanno and Rita, who always believe in me. My wife Beata for her persistent understanding and trust.



# 0

## Introduction

TERRESTRIAL LASER SCANNING (TLS)<sup>1</sup> has meanwhile reached a high acceptance level in applied as well as engineering geodesy and is consequently used in numerous fields of application such as kinematic laser scanning (BOEDER et al. 2010), deformation monitoring (MECHELKE et al. 2012), (LINDENBERGH and PIETRZYK 2015), archaeology and cultural heritage (BÖHLER and MARBS 2004) or biomass estimation (TILLY et al. 2013).

While in the past, in contrast to *total stations*, the quasi pioneering technology of *laser scanners*, it has always been possible to ensure areal acquisition in the photogrammetric field, the evaluation, however, was limited to the pointwise discretization of objects for the various applications, partly due to the insufficient computing and storage capacity at that time. For instance, in the monitoring of buildings and natural objects with regard to movements and deformations, spatial discretization was in the past a fundamental method of engineering geodesy, see (BRUNNER 2007).

Since the appearance of TLS in engineering geodesy, a paradigm shift from this previously point-based to a now widely characterized approach by linear or surface-based measurement and evaluation methods has taken place (KUHLMANN et al. 2013).

---

<sup>1</sup> In the following, TLS is used to denote both the instrument (Terrestrial Laser Scanner) and the measuring method (Terrestrial Laser Scanning).

0.1 PROCESSING AND ANALYSIS OF TLS MEASUREMENTS

The quasi-surface acquisition method associated with TLS, see Section 1.2.1 and Figure 1.3, and the resulting vast amount of data for an acquisition, also called scan, is mapped in a so-called *point cloud*. In order to capture an object in its entirety, or to determine changes to an object, for example, several scans of potentially other instrument positions<sup>2</sup> with different perspectives onto the object are necessary. A classical evaluation strategy with the central processes from the planning of the measuring project via acquisition and evaluation towards the final product is shown in Figure 1 and is usually pursued for the above mentioned fields of application.

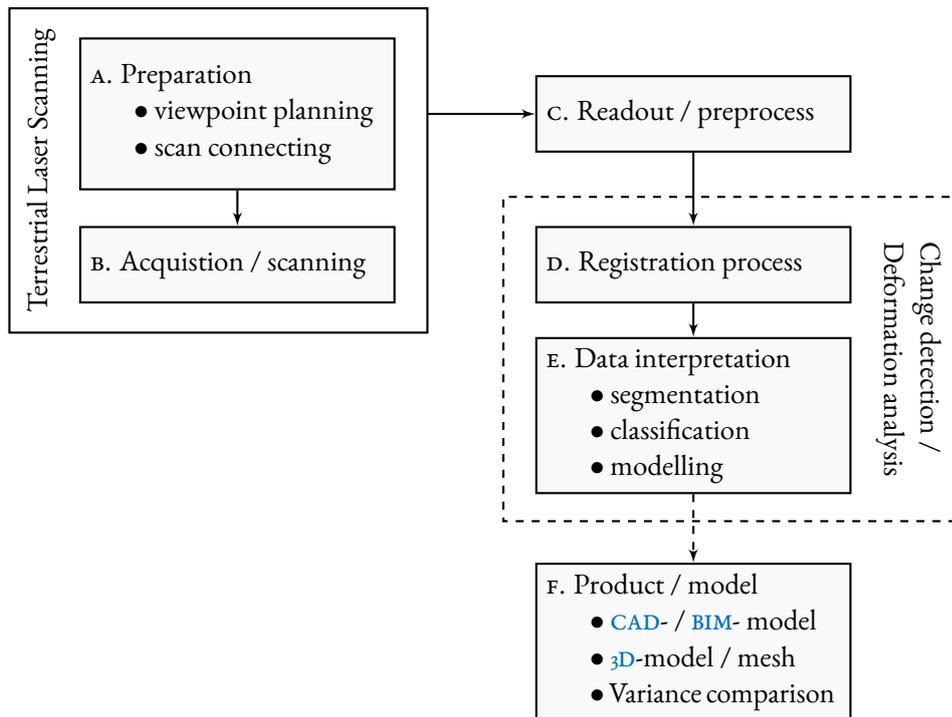


Figure 1: Evaluation strategy and processing chain for TLS.

The general workflow for the use of a TLS according to Figure 1 usually starts with a

- A. preparation step involving a viewpoint planning under engineering geodetic aspects (WUJANZ et al. 2016b), as well as the definition of the geometrically overlapping scans, followed by the actual
- B. acquisition based on the pre-planning, the

<sup>2</sup> Instrument positions are also simply denoted as standpoints in the following.

- C. readout of the data per scan from the scanner and possibly a pre-processing such as e. g. filtering or data reduction is performed, usually followed by a
- D. referencing of the individual scans into a common coordinate system, see also [Section 1.II](#), whose referenced overall *point cloud* is subjected to an
- E. interpretation step with regard to a segmentation, see also see [Section 1.IO](#), classification or a modeling of areas, see also [Section 4.3.2](#), of the *point cloud* in order to finally obtain a finished
- F. product or model in the form of a CAD- or BIM-model, 3D-model or similar.

According to KUHLMANN et al. (2013) the so-called engineering geodesy comprises the complete set of methods for the evaluation and modelling of the measured quantities as well as the derived estimated parameters. Among them, the planning process, e. g. also the viewpoint planning, as well as the measuring process itself, see left part of [Figure 1](#), and the complete subsequent evaluation and interpretation, see right part of [Figure 1](#), represent the essential core competences of engineering geodesy (KUHLMANN et al. 2014). Within the core competence,

- D. Registration, see [Section 1.II](#), as well as
- E. Segmentation, see [Section 1.IO](#)

belong to the fundamental processing steps with regard to an automated evaluation and further processing of *point clouds*.

SEGMENTATION IN GENERAL: Segmentation (lat. *segmentum* – section) is the subdivision of a whole into individual expedient segments (i.e. sections, parts or elements) that can be segregated from one another.

REGISTRATION IN GENERAL: The task of registration or referencing (lat. *referre* – refer to something) is to transfer the measurement data (2D or 3D) of individual view- or standpoints given in a local coordinate system from an instrument – e. g. captured with a TLS – into a common (superordinate) coordinate system. The result of a registration is a complete dataset composed of measurements, e. g. three dimensional (3D) points, captured from several connected<sup>3</sup> standpoints. This process is of utmost importance in the completion of a dataset.

Considering the engineer-geodetic challenge to detect geometric changes with regard to a deformation analysis of an object or building, the segmentation of unchanged areas for a subsequent unadulterated registration spanning several epochs is indispensable. Hence, the quality of the segmentation and registration results rises and falls with the quality of the statement of a deformation analysis.

<sup>3</sup> See [Section 1.II](#) how a connection could look like

## INTRODUCTION

### 0.1.1 CURRENT EVALUATION STRATEGY

When looking at the current evaluation strategy of *TLS point clouds*, see [Figure 1](#), with regard to the registration, see [Section 1.11](#), as well as the segmentation of point clouds, see [Section 1.10](#), it can generally be stated that the individual processing steps of the methodologies, relying on different data concepts, are carried out independently of each other and in sequential order.

In ([MAHMOUDABADI et al. 2016](#)) the segmentation of a single scan is efficiently solved, yet the context for registration is not discussed. A registered *point cloud* is used by RIVEIRO et al. ([2016](#)), however, the previous registration is not related to the segmentation. In both cases, the focus remains on segmentation.

**REMARK:** In order to be able to perform a subsequent segmentation in the second processing step after the registration, e. g. ([RIVEIRO et al. 2016](#)), the extensive buildup of topological relationships between individual points is again required, see [Section 1.4](#).

If one considers registration methods, planes are segmented in ([GIELSDORF 2009](#)) first, which serve afterwards as identical information for the registration, but they have no further purpose.

RABBANI et al. ([2006](#)) extend this concept by detecting several geometric primitives in the form of cylinders, spheres and also planes in the *point clouds* and using this information for registration and modeling. However, the registration also takes a leading role.

**IMPORTANT NOTE:** Existing strategies for solving both tasks are, however, based on a registration that is independent of the segmentation in its processing step. A mutual support between segmentation and registration methods, which is hereinafter referred to as "*synergetic segmentation and registration*", has not yet been applied.

**IT IS WORTH MENTIONING THAT** currently all methods, whether for registration or segmentation, are based on either no quality models or incomplete ones due to missing, or not fully modeled, stochastic information for the measured values, see [Section 0.2.1.1](#) and also [Section 1.9](#).

0.2 QUALITY IN THE CONTEXT OF TLS

The quality assessment of measuring methods and systems on the one hand, and the results of a measuring and evaluation process on the other hand, has always been one of the key aspects in geodesy and therefore also takes on a major role in conjunction with TLS.

QUALITY IN GENERAL: Quality (lat. *qualitas* – condition, feature, characteristic or status) is defined according to the valid standard DEUTSCHES INSTITUT FÜR NORMUNG E.V. (2015) for quality management as the degree to which a set of inherent characteristics of an object fulfills requirements and thus indicates the extent to which an object or result meets existing requirements.

By definition, the term quality can be broadly defined and can be transferred either to a measuring instrument, the measuring process or the evaluation and its results. As a basis for any rating, a so-called *quality model* with characteristics and parameters serves (SCHWIEGER and ZHANG 2019).

KUHLMANN et al. (2014) also dedicates a separate section to quality assessment and quality management within the core competencies of engineering geodesy.

One of the most important characteristics of quality established in geodesy is the accuracy whose measure or parameter – with respect to the result of various evaluation processes as well as the *adjustment calculation*, see Section 1.8, – is given as so-called Standard Deviation (SD).

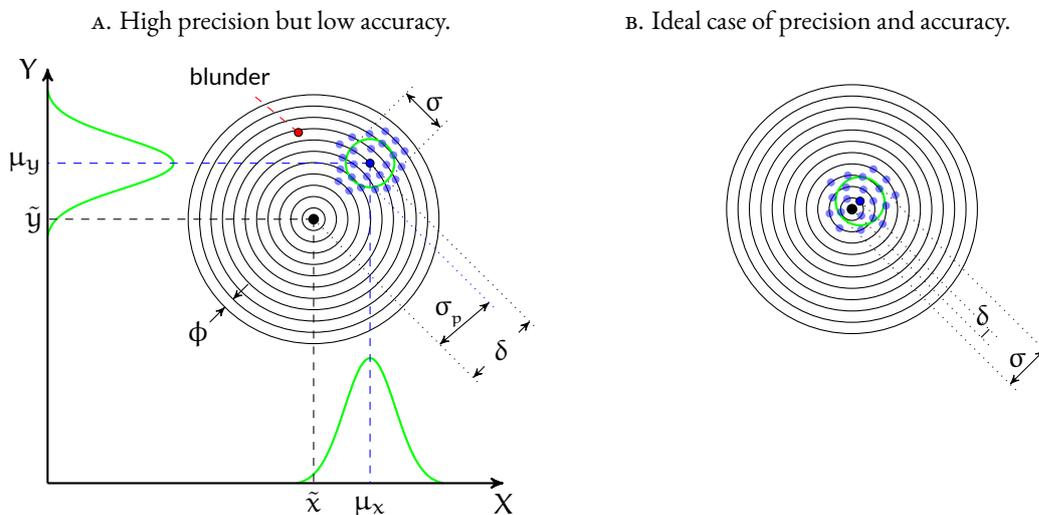


Figure 2: Relationship between accuracy, precision and correctness of measured values according to (SCHWARZ 1995) and (WITTE and SPARLA 2015, p. 69).

## INTRODUCTION

In [Figure 2](#), the concept of accuracy, including both *systematic* and *random errors*, see also [Sections 1.2.4](#) and [1.8.2](#), is illustrated graphically, where

- $\sigma$  represents the so-called precision  $\sigma$ , indicates the size of the random measurement deviations of a measurement series around its mean value, approaching the expected values  $\mu_x, \mu_y$  with an increasing number of repeated measurements,
- $\delta$  denotes the trueness, the size of the systematic deviation as the difference between the mean or expected values  $\mu_x, \mu_y$  of the measurement series and the values  $\tilde{x}, \tilde{y}$  recognized as correct values,
- $\sigma_p$  denotes the accuracy of a single point of the series of measurements and
- $\phi$  indicates the resolution with which the relevant measuring instrument generates the measured values.

Accuracy according to [Figure 2A](#), thus describes the correspondence between the values  $\tilde{x}, \tilde{y}$  considered to be correct and single measured values  $x_i, y_i$  and can thus be regarded as a generic term for correctness and precision.

NOTE: A measurement result is accurate if it is theoretically free of *systematic* and *random errors*.

The ideal case is shown in the right [Figure 2B](#), when the true values  $\tilde{x}, \tilde{y}$  coincides with the expected values  $\mu_x, \mu_y$  within the precision  $\sigma$ , i.e.  $\delta < \sigma$ . If  $\delta$  is even within the resolution  $\phi$  of the measuring instrument, one speaks of the measurements being free of *systematic errors*, or transferred to a measuring instrument that has been comprehensively calibrated, see also [Section 1.2.5](#).

Not to be neglected in the quality assessment, however, are also the reliability criteria, see also [Section 1.8.9.2](#), so-called *redundancy components*<sup>4</sup> as described in NEITZEL (2010). With respect to the adjustment calculation, [Section 1.8](#), not only the *stochastic model*, see [Section 1.8.3](#) or especially [Section 1.9](#), but also the functional model, see [Section 1.8.2](#), is crucial for an unambiguous quality assessment of the results. As NEITZEL (2010) points out, the matching of a complete functional relationship between the observations (*model building*) and the searched parameters with an appropriate *stochastic model* for the observations is indispensable.

---

<sup>4</sup> The *redundancy components* of the observations, see also [Section 1.8.9.2](#), indicate to what extent the respective observation is controlled by the measurement configuration. In the case of sufficiently well controlled observations based on an adequate measurement configuration, measurement errors in the data material can then ideally also be detected and filtered out, see [Section 1.8.8](#).

## 0.2.1 QUALITY MODELS FOR TLS

When transferring the concept of quality to TLS, special *quality models* can be formulated within the context of a quality assessment, describing on the one hand the quality of the instrument, the acquisition and the resulting *point clouds*, see Table 1, and on the other hand also that of the evaluation processes registration and segmentation, see Table 2.

Table 1: Quality model for TLS, acquisition and *point cloud*.

QUALITY MODELS	QUALITY CHARACTERISTICS	QUALITY PARAMETERS
INSTRUMENT	calibration and accuracy	parameters with STD
	measuring resolution	$\Delta\rho, \Delta\phi, \Delta\theta$
	unambiguous max. distance	$\rho_{max}$
	stochastic model	precision $\sigma_\rho, \sigma_\phi, \sigma_\theta$
ACQUISITION	measuring arrangement	viewpoint plan
	sampling rate	low, normal, hig, premium
	angular resolution	medium, high, super or ultra high
POINT CLOUD	degree of discretization	point spacing $\Delta x, \Delta y, \Delta z$
	point precision	$\sigma_x, \sigma_y, \sigma_z$
	information content	geometric, radiometric
	completeness	via registration

Accordingly, a TLS can be qualitatively assessed by means of the criteria whether and how extensively it has been calibrated with a certain accuracy, cf. Section 1.2.5, which maximum unambiguous distance  $\Delta\rho$  and maximum angular resolution  $\Delta\phi, \Delta\theta$  can be achieved device-technically, cf. Section 1.2.6 and Section 1.2.2.1, especially Table 1.2, how large the maximum achievable measuring distance  $\rho_{max}$  is, see Table 1.2, and to what extent precision statements ( $\sigma_\rho, \sigma_\phi, \sigma_\theta$ ), cf. also Table 1.2 and see Section 2.1, for the raw measuring information are possible based on a *stochastic model*.

However, the quality of the acquisition is directly related to the choice of the angular resolution and acquisition rate for the distance measurement, see also Section 1.2.6, and the optimization of the measurement setup in the course of a viewpoint planning as e. g. in (WUJANZ et al. 2016b).

By means of the degree of discretization, expressed by the point spacing  $\sigma_x, \sigma_y, \sigma_z$ , on the object, i. e. how much point information is located in a certain area, the knowledge of the precision of these points  $\sigma_x, \sigma_y, \sigma_z$ , the information content of geometric as well as radiometric information and, if applicable, the completeness by a registration procedure, the quality of the *point clouds* generated with a TLS can be characterized. Since the knowledge about the point precisions depends on the one hand

## INTRODUCTION

on the use of an adequate *stochastic model*, cf. [Section 2.3.1](#), and on the other hand the point spacing on the object depends on the chosen, or planned measurement configuration, as well as the selected acquisition parameters, cf. [Section 1.2.1](#), especially [Figure 1.3](#), there is a direct connection between the choice of the [TLS](#) as well as the type of recording with the quality of a resulting *point cloud*.

Table 2: Quality models for [TLS](#)'s evaluation steps segmentation and registration.

QUALITY MODELS	QUALITY CHARACTERISTICS	QUALITY PARAMETERS
SEGMENTATION	completeness	–
	diversity in objects	curved, planar, cylindrical, etc.
	separation precision	geometric and radiometric
REGISTRATION	precision	parameters with STD
	reliability	redundancy components

Actually, in the case of [TLS](#), due to the complexity of the measurement and evaluation process, one must speak of *quality modeling* over several process steps, as NEUNER (2019) explains referring to the example of registration with targets, cf. also [Section 1.11](#). As already illustrated in [Table 1](#), the individual model categories are intertwined and only in combination form a comprehensive quality picture. Hence, not least the quality of a *point cloud* resulting from the choice of the instrument and the acquisition configuration is decisive for the quality of subsequent processing steps such as segmentation and registration, see [Table 2](#).

When considering segmentation methods with regard to [TLS](#), the primary goal is to obtain a complete segmentation result, see also [Section 1.10](#), especially [Item 1](#), and, if possible, to cover a large diversity in the recognition of objects, such as planar, curved or even cylindrical surface elements, with a minimum of additional information. The precision of the separability of the individual objects from each other depends on the one hand on the precision  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$  of the points themselves and on the other hand on the degree of discretization, respectively the point spacing  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$ , on the object, cf. also [Table 1](#). Accordingly, the segmentation result is also directly defined by the quality during the acquisition and the use of a [TLS](#) with a qualitative *stochastic model*.

Registration techniques rely on methods of adjustment calculus, see also [Section 1.8](#), and regardless of the design of the functional model, see also [Section 1.8.2](#), a stochastic model, see [Section 1.8.3](#), is always underlying, which can be derived from the precisions of the observation pairs used for registration, cf. e. g. [Equation \(4.24\)](#). Reliability measures such as redundancy components, see [Section 1.8.9.2](#), also provide insight into the controllability of the observations in registration procedures and can be used in terms of the calculation of normalized residuals to significantly identify poor observation data, see [Section 1.8.8.2](#).

## O.2.1.1 STOCHASTIC MODELLING

*Stochastic models* constitute one of the essential quality characteristics in the use of TLS, see Table 1, since, as explained in the previous section, they influence the quality of both the *point clouds* arising from the acquisition and the processing methods to be applied to them.

For instance, this becomes apparent when using the point precision to estimate the significance to which points belong by definition to a plane in the course of a plane segmentation, cf. Section 4.3.2.

When considering registration procedures, the influence of the observations can thus be controlled via the weight derived from the respective precision on the adjustment result, see also Section 1.8.3 and (NIEMEIER 2008, p. 124). Hence, the use of a meaningful *stochastic model* for the referencing of TLS scans is of utmost importance for an adequate precision statement of the registration parameters, not least since only then erroneous observations as well as weak or false correspondences can be correctly identified and eliminated, cf. Section 1.8.8.2.

SUITABLE *stochastic models* are indispensable for both segmentation and referencing algorithms.

However, modeling the quality for the reflectorless distance measurement of a TLS as a precision characteristic is a considerable challenge in practice owing to the numerous precision-limiting environmental influences and its complexity, see Section 1.2.4.1. So far, various approaches exist to incorporate *stochastic models*, see Section 1.9, however, they could not be mapped in their completeness to consider all influences in a unified and comprehensive model and consequently have only moderate explanatory power.

## O.3 AIM AND SCOPE OF RESEARCH

The research topics of this dissertation are thematically classified into the modeling of quality in TLS, inspired by the motifs in Section 0.2, especially Section 0.2.1 and Section 0.2.1.1, on the one hand, and the evaluation process of TLS measurements, inspired by the motifs in Section 0.1, especially Section 0.1.1, on the other hand.

RESEARCH FOCUS I: As a consequence of the missing or only partially modeled *stochastic models* for TLS, see Section 0.2.1.1 and also Section 1.9, previous *quality models* for TLS, such as in Table 1, are incomplete, nevertheless, as described in Section 0.2.1, having a great relevance and significance on evaluation procedures and their quality, as well as the segmentation and registration in the context of TLS. Hence, Chapter 2 investigates whether an all-embracing *stochastic model* that covers all environmental influences can be established in order to enhance and complement the *quality model*.

## INTRODUCTION

RESEARCH FOCUS 2: The potential of segmentation as well as registration methods for TLS currently is not fully exploited caused by the sequential and independent approach to evaluation, see Section 0.1.1 and Figure 1, and hence potential synergies cannot be used. Moreover, the lack of or the incomplete application of stochastic information on the basis of a *stochastic model* for the evaluation procedures is a further weakness. As a consequence, Chapter 4 examines in how far partial information derived from different segmentations in Chapter 3 can be extracted and used in the course of a registration procedure and, conversely, investigate how the registration information obtained might serve a comprehensive segmentation.

CONSEQUENTLY, IN SUMMARY, this thesis introduces a new method that exploits the existing topology of single scans for efficient segmentation by using extracted features, enriched with stochastic from a new stochastic model, as corresponding registration information to complement the previous segmentation result.

### 0.4 THESIS ORGANISATION

The individual chapters of the present thesis are to be seen in a certain context, which results in a structure according to Figure 3.

First of all, Chapter 0 initiates by emphasizing the significance of TLS and the associated evaluation steps of segmentation and registration, which culminates to the motivational reasons for the main chapters.

Building on this, Chapter 1 provides a brief insight into existing methods (Sections 1.9 to 1.11) and illustrates underlying principles and evaluation methods, to be seen in the context of *laser scanning*, with relevance to and as a prerequisite for the main chapters. Among them are the genesis of the measurement, as well as influences on the measurement within TLS in Section 1.2, the representation of the measurement in Section 1.4, mathematical background such as coordinate systems, *graph theory*, HELMERT's transformation or the definition of rotations in Section 1.3 and Sections 1.5 to 1.7, as well as concepts of adjustment calculus, in particular the Gauß Helmert Model (GHM) and the accompanying elimination of outliers in Section 1.8.

In the first major Chapter 2, a new *stochastic model* is established by means of the intensity behaviour and influencing factors on the distance measurement of a TLS in Sections 2.1 and 2.2, followed by the evaluation and derivation in Section 2.3 and the demonstration of suitability in Section 2.4. The derivation of *stochastic models* in 3D measuring mode as well as the usability for pulse-based TLS are shown in Section 2.5 and Section 2.6 for completeness.

The presentation of a new, very efficient method for the segmentation of *point clouds* based on geometric as well as radiometric information is given in Chapter 3 referring to an example data set

from [Section 3.1](#), preprocessed in [Section 3.2](#). After the description of the segmentation algorithm in [Section 3.3](#), results with concluding evaluation are shown in [Sections 3.4](#) and [3.5](#).

Taking advantage of a pre-segmentation, see [Section 4.3.1](#), of the individual *point clouds* of the example data set in [Section 4.1](#) provided by the segmentation algorithm implemented in [Chapter 3](#), planar regions with derived stochastic information based on [Chapter 2](#) are segmented and matched in [Section 4.3.2](#) and [Section 4.4](#) of [Chapter 4](#) in order to reference the point clouds in [Section 4.5](#) and finally to complement the individual segmentations to an overall segmentation in [Section 4.6](#) via the existing matching. The final results and a closing discussion of the segmentation algorithm can be found in [Sections 4.7](#) and [4.8](#).

A concluding consideration of the individual chapters of this thesis with reflections for further investigations ends with [Chapter 5](#).

#### 0.4.1 MAIN STRUCTURE OF THE THESIS

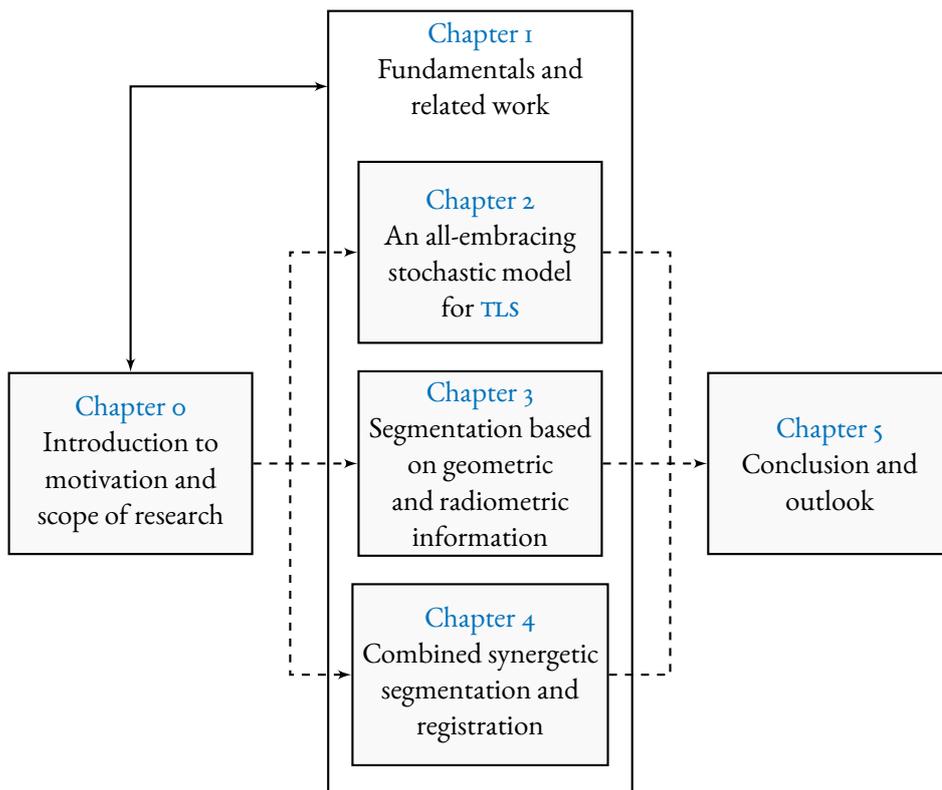


Figure 3: Structure and dependencies of chapters.

## INTRODUCTION

### 0.4.2 IMPLEMENTATION OF FUNCTIONS

The algorithms and the functions contained therein for the evaluation of the *laser scans* with regard to segmentation and registration were implemented exclusively in the interpreter language Matrix Laboratory ([MatLab](#))<sup>5</sup>.

---

<sup>5</sup> [MatLab](#)<sup>®</sup> is a proprietary software developed by the U.S. company MathWorks<sup>®</sup> (<https://www.mathworks.com/products/matlab.html>) to overcome mathematical problems, mainly numerical calculations based on matrices, and to graphically visualize the results.

# 1

## Fundamentals and Related Work

THE FOLLOWING CHAPTER DEALS with the general theoretical and functional relationships as well as the basics in the field of **TLS** for the subsequent main chapters and serve as a guide. In particular, a categorization of the measurement method is given in [Section 1.1](#), followed by measurement principles as well as the resulting measurement quantities and error influences in [Section 1.2](#). Processing methods of relevance for this thesis are presented in [Sections 1.9 to 1.11](#), as well as the required data representation in [Section 1.4](#). Special functional relationships and methods of particular concern are presented in [Sections 1.5 to 1.7](#), and an introduction of adjustment methods is finally given with [Section 1.8](#).

### 1.1 METHODS OF 3D-DATA ACQUISITION

Measuring methods enabling the measurement of **3D** information are characterized by a variety of features and properties. Basically, a categorization into

- A. passive and
- B. active methods

can be made in this context. Whereas active methods, see [Figure 1.1](#), transmit energy to the object to be captured in order to subsequently receive the emitted radiation, passive methods, among them imaging methods such as *photogrammetry* and especially Digital Image Correlation (**DIC**), operate with existing radiation.

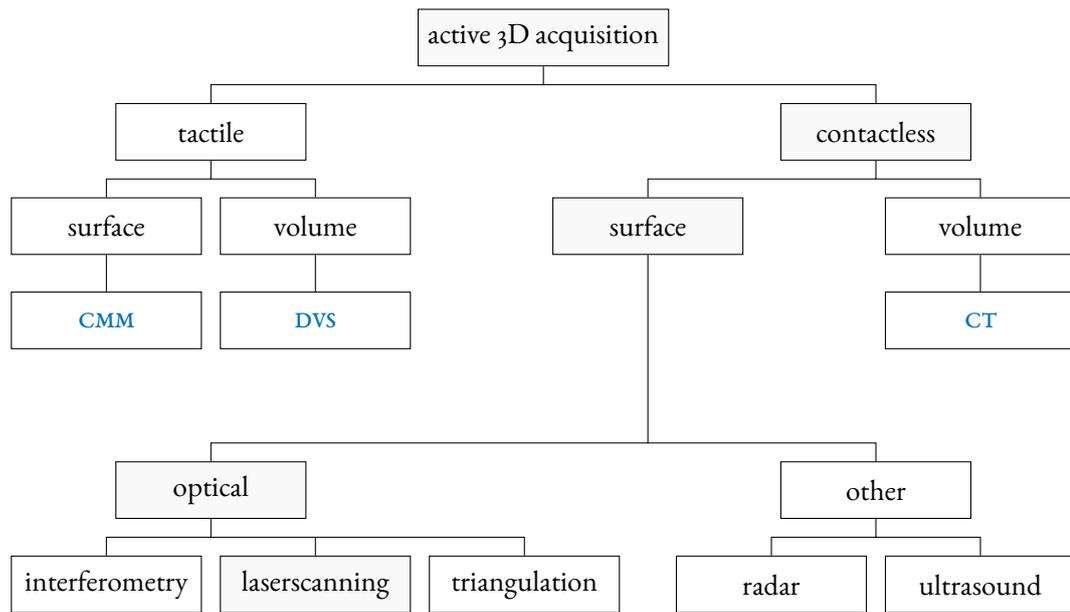


Figure 1.1: Active 3D acquisition methods based on (GÜHRING 2002, p. 14).

Scanning laser measuring systems as well as TLSs are categorized within the active methods among those that operate contactless and areal<sup>1</sup>, cf. Figure 1.1, having an optical measuring component.

## 1.2 LASER SCANNING

In general, the term *laser scanning* refers to the 3D capturing of an object’s surface by temporal and spatial scanning with a laser beam (DEUMLICH and STAIGER 2002, p. 403). The combination of scanning under the spatial angles of rotation  $\phi$  and  $\theta$  with distance measurement  $\rho$  is reflected in the so-called polar measuring principle, see Figure 1.3. The simultaneous and mostly incremental recording of these polar elements enables the conversion of the 1D distance measurement to 3D cartesian object coordinates, see Equation (1.1). The entirety of this set of points that is formed during the acquisition process of an object is referred to as *point cloud*.

The basic functionality of scanning devices is similar, but they can be further differentiated by various beam deflection systems

- rotating mirrors,
- oscillating mirrors,

<sup>1</sup> In contrast to *imaging methods*, the acquisition of measured values with *laser scanning* is strictly speaking quasi-areal based due to the mode of operation, see also Figure 1.3.

- rotating / oscillating mirrors

and kinds of EDMUs, see Section 1.2.2. *Laser scanning* is used in various fields of application, according to which three general categories of *laser scanners* can be distinguished:

1. Terrestrial Laser Scanners (TLSs)
2. Mobile Laser Scanners (MLSs)
3. Airborne Laser Scanners (ALSs)

According to their name, TLSs are used for *terrestrial* or stationary ground-based *laser scanning*, capturing the environment spherically from a stationary position. Most current TLS use beam deflection systems in the form of rotating mirrors and differ significantly in their potential range, realized with common EDM (rangefinder) methods, such as Time Of Flight (TOF) and Amplitude Modulated Continuous Wave (AMCW) principles, see Section 1.2.2. Depending on the device type and the corresponding EDMU, these are suitable in practice, e. g. for the surveying of industrial plants, larger buildings up to high structures or opencast mines.

*Laser scanners* mounted on a mobile moving platform are referred to as MLSs and ideally have only a single vertical beam deflection unit, thus they are also known as *profile scanners* according to their scanning mode. In addition to the MLS, a GPS unit and an IMU are usually installed on the platform so that the data can be transferred to a common reference frame by *direct georeferencing*. This kind of system is termed a Mobile Mapping System (MMS). For elongated and extensive target areas, e. g. tunnels or roadways, the Terrestrial Laser Scanning is not really suitable due to the requirement of many individual standpoints and therefore more time-efficient MMSs are used. According to the environmental requirements, also different distance measuring methods are used in certain devices.

The functionality of ALS is similar to that of MLS, whereby some have an oscillating mirror as beam deflection unit and, due to the one-sided orientation caused by the application, only require a comparatively small Field Of View (FOV) (typically 40° to 70°) in contrast to MLS (360°), (METTENLEITER et al. 2015, p. 30). A major aspect of ALS is the possibility of multi-target capability<sup>2</sup> in combination with a long range, allowing only pulse propagation methods to be used, for example in forest areas, to scan both treetops and the forest floor.

### 1.2.1 TERRESTRIAL LASERSCANNER – SCANNING PRINCIPLE AND MEASUREMENTS

The main internal components a TLS consists of as a so-called *panorama scanner*<sup>3</sup> are the following

<sup>2</sup> The almost new TLS Riegl VZ-400i also has a multi-target EDMU embedded.

<sup>3</sup> *Panorama scanners* are capable of rotating around their standing axis, enabling them, in contrast to *camera view scanners* with a maximum FOV of approximately 50° x 50°, to capture the environment except their footprint in a 360° scan.

## FUNDAMENTALS AND RELATED WORK

- A. Electro-optic Distance Measurement Unit (**EDMU**)
- B. motor and standing axis ( $\phi$ ) with angle encoder (light green)
- C. device base to connect the **TLS** with a tribrach on a tripod
- D. rotating mirror
- E. motor and tilting axis ( $\theta$ ) with angle encoder (light green)
- F. Central Processor Unit (**CPU**)

depicted in [Figure 1.2](#). Within the device, continuous laser light is emitted by the diode of the **EDMU** (A), which incident the rotating mirror (D) of the vertical beam deflection unit (E) and is deflected perpendicularly. The signal reflected by the object then returns to the **EDMU** in the same way and is recorded by a photo diode and evaluated.

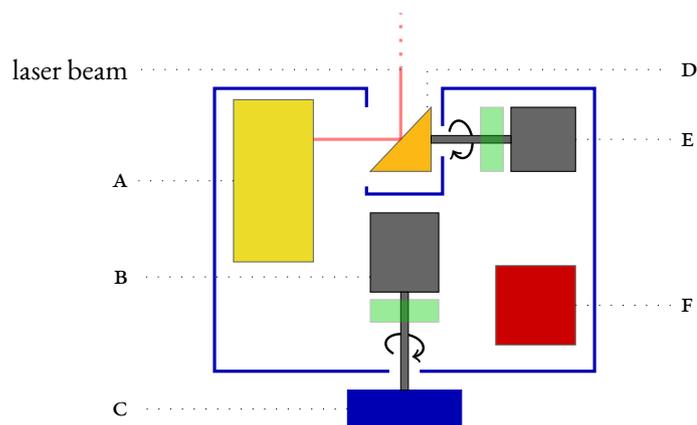


Figure 1.2: Basic components of a **TLS** based on ([METTENLEITER et al. 2015](#), p. 14).

Due to the circular deflection  $\theta$  ( $360^\circ$ ) of the laser beam with a high rotational speed of typically 25 to 50 Hz of the rotating mirror coupled with the measuring rate of up to 2 million distance values per second of the **EDMU**, up to 20000 **2D** points are obtained on a profile plane orthogonal to the tilting axis. A further, significantly slower rotation  $\phi$  ( $180^\circ$ ) performed by the horizontal deflection unit (E) around the standing axis of the device results in up to 5000 profiles or 50 million **3D** points per *laser scan* ([METTENLEITER et al. 2015](#), p. 14). This incremental scanning of the object, see [Figure 1.3](#), is coordinated by the control unit of the **CPU**, so that the angle encoders of the two deflection units (B and E) capture the two angle values  $\theta$  and  $\phi$  synchronously with the distance measurement  $\rho$ , see [Section 1.2.2](#), in the **EDMU**.

The elementary observations of a **TLS** are the *polar elements*<sup>4</sup>  $\phi$  (direction of rotation),  $\theta$  (tilting angle), the corresponding reflectorless distance measurement  $\rho$  as well as from the interaction with the object resulting intensity  $I$ . Due to the incremental row ( $i$ ) and column ( $j$ ) scanning of the laser scanner in a predefined angular increment  $\Delta\theta = \Delta\phi$ , the **3D** coordinates, see also [Section A.1.2](#),

$$\mathbf{x}_{i,j} = \begin{bmatrix} x_{i,j} \\ y_{i,j} \\ z_{i,j} \end{bmatrix} = \begin{bmatrix} \rho_{i,j} \sin(\theta_0 + i\Delta\theta) \cos(\phi_0 + j\Delta\phi) \\ \rho_{i,j} \sin(\theta_0 + i\Delta\theta) \sin(\phi_0 + j\Delta\phi) \\ \rho_{i,j} \cos(\theta_0 + i\Delta\theta) \end{bmatrix} \quad (1.1)$$

have a definite position in a raster, where  $\rho_{i,j}$  denotes the distance corresponding to  $\theta_{i,j} = (\theta_0 + i\Delta\theta)$  and  $\phi_{i,j} = (\phi_0 + j\Delta\phi)$ . [Figure 1.3](#) shows the scanning of a building and the formation of such a grid.

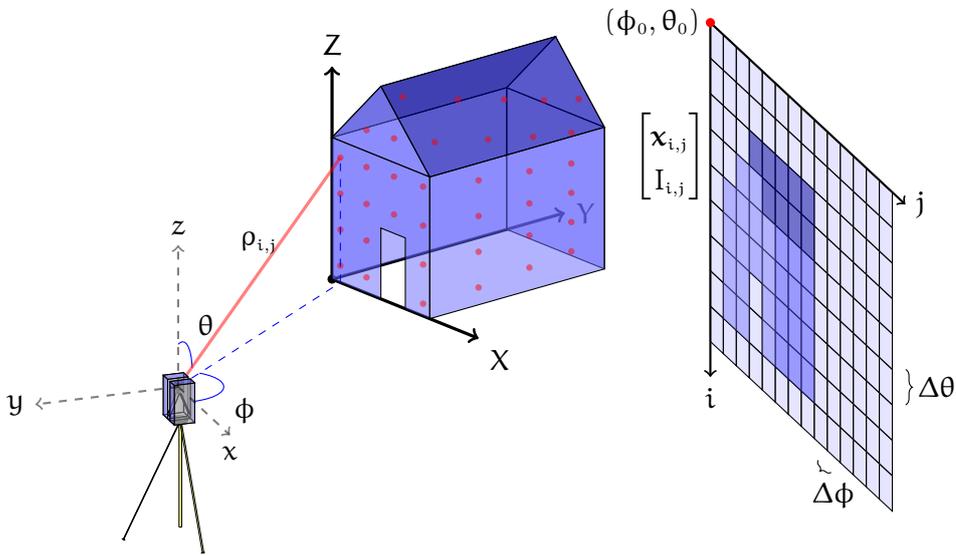


Figure 1.3: Creation of a grid by incremental scanning of the **TLS**.

The control unit of the **TLS**'s **CPU**, see [Figure 1.2](#) and denoted by  $\mathbb{F}$ , is responsible for coordinating the measurements with the movements of the deflection units, denoted by  $\mathbb{B}$  and  $\mathbb{E}$  in [Figure 1.2](#), whereby the pixel-by-pixel aggregation<sup>5</sup> of the measured values is already carried out in the **EDMU**, [Figure 1.2 A](#), ([METTENLEITER et al. 2015](#), p. 21).

<sup>4</sup> Usually, the manufacturers of **TLSs** only provide **3D** coordinates as the result of an acquisition, which can be converted into *polar coordinates* if required.

<sup>5</sup> With oscillating scanning heads, line shifts may occur under certain circumstances, so that the topology in the **2D** grid representation, see [Figure 1.3](#) on the right, is no longer given precisely. However, this error can simply be corrected by means of the known *polar elements*, but is not of importance for the *panorama scanner* used in the investigation, as it has a rotating scanning head.

1.2.2 ELECTRO-OPTICAL DISTANCE MEASUREMENT (EDM) – PRINCIPLES AND DIFFERENCES

For a better understanding of the connections and the error budget of the measured values in the following sections, the basic principles of EDM are explained in this section, with reference to e. g. (DEUMLICH and STAIGER 2002) for a more detailed explanation.

Various methods are suitable for EDM, differing mainly in their application, measurement range and expected precision, such as

- pulse propagation time method (TOF),
- multi waveform method,
- triangulation method,
- phase difference method (AMCW),
- frequency difference method (FMCW),
- interferometry.

Table 1.1 lists the most common selection criteria for certain measurement methods, see Section 1.2.2, in order to define the measurement method for the respective measurement task.

Table 1.1: Comparison of different methods for EDM with regard to their performance data according to (METTENLEITER et al. 2015, p. 9).

METHOD	MEASURING RANGE	CHARACTERISTICS
pulse propagation	1 m – several km	fast, new standards due to multi-waveform method
triangulation	1 m – 30 m	cheap, robust for short distances, strongly dependent on the surface
interferometry	10 μm – 50 m	slow, expensive, very high accuracy
phase difference	max. 300 m	very fast, reasonable costs for high accuracies

Among the most common EDM methods used in modern *laser scanners* are the pulse propagation method, also known as Time Of Flight (TOF) principle, and the phase difference method.

**PULSE PROPAGATION METHOD:** For the **TOF** method, very short light pulses are generated in the **EDMU** via a high-frequency oscillator with a clock frequency of around 15 GHz. A light pulse emitted by the transmitter diode at time  $t_e$  with a certain amplitude  $A_e$  is reflected at an object surface and after a certain time  $t_r$  registered by the receiver of the **EDMU** with an amplitude  $A_r$ , see [Figure 1.4](#).

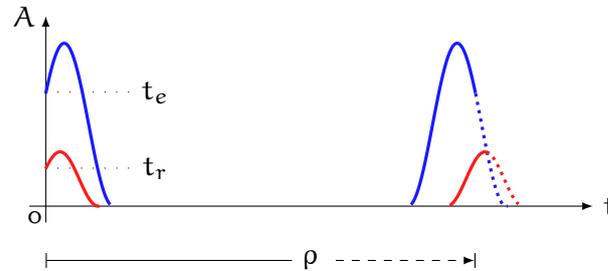


Figure 1.4: Principle of pulse method.

The distance to the object

$$\rho = \frac{c \Delta t}{2n_r} \tag{1.2}$$

is determined by the measured time difference  $\Delta t$ , which is proportional to the travelled path, where

- $c$  speed of light in vacuum
- $\Delta t = t_r - t_e$  time difference between emitted pulse at  $t_e$  and received pulse at  $t_r$ , see [Figure 1.4](#)
- $n_r$  refractive index of medium

An essential advantage of the **TOF** method compared to the phase difference method, see [Section 1.2.2](#), is the unambiguousness of the distance measurement and the possibility of measuring larger distances.

However, the **TOF** method is less accurate due to the too high demands on time measurement resulting from the enormous speed of the light.

**MULTI-WAVEFORM METHOD:** The multi waveform method is based on the classical **TOF** method, see [Section 1.2.2](#), whereby all information is recorded within the time window between transmission and receipt. If several objects are in the beam path within this time period, this leads to multiple reflections, e. g. due to the partial transmittance of some object surfaces, whose received pulse sequences can be evaluated individually.

For example, the distances to several objects in one spatial direction can be determined or any interfering objects can be filtered out. In particular **ALS** systems are able to achieve a higher temporal and hence spatial resolution ([ROTH, THOMPSON, et al. 2008](#)).

## FUNDAMENTALS AND RELATED WORK

**TRIANGULATION METHOD:** Triangulation is one of the oldest existing measurement methods and dates back to the 8th century. As the name triangulation suggests, the geometry within a triangle is used under certain assumptions (known quantities) to calculate the desired measured quantity.

A large variety of modern so-called active and passive triangulation methods are used in current measuring instruments. Active triangulation methods

- laser triangulation,
- light section triangulation,
- structured light triangulation,

use at least one (structured) light source in combination with at least one camera sensor, whereas passive methods are based on classical photogrammetry with only one or more 2D cameras.

Contrary to other methods, laser triangulation indirectly determines the distance to the object. The laser beam, which is only used for signalization, hits an object and is detected as a light spot at a certain position in the photosensitive sensor, see [Figure 1.5](#).

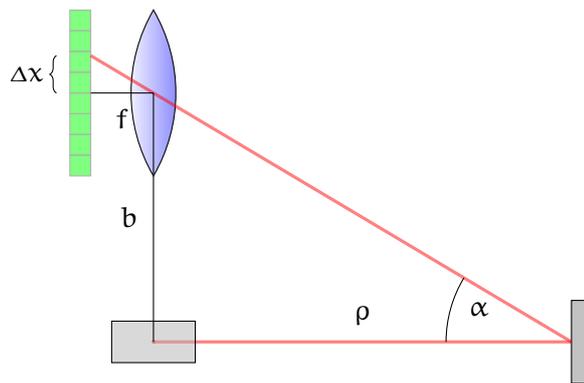


Figure 1.5: Principle of laser triangulation.

In a simplified representation, the displacement  $\Delta x$  of the light spot on the sensor and the focal length  $f$  have the same ratio as the object distance to be determined  $\rho$  and the base distance  $b$  between laser diode and camera, representing the tangent of the triangulation angle  $\alpha$ .

$$\frac{b}{\rho} = \frac{\Delta x}{f} = \tan \alpha \quad \Rightarrow \quad \rho = \frac{bf}{\Delta x} \quad (1.3)$$

**PHASE DIFFERENCE METHOD:** In contrast to the [TOF](#) method, with the phase difference method<sup>6</sup> continuous high-frequency laser light (continuous wave as carrier wave) is amplitude-modulated with a

<sup>6</sup> The phase difference method is also often referred to as the phase comparison method and, due to the formation of the measurement signal, the term [AMCW](#) method is also common.

sinusoidal signal (modulation wave) whose modulation frequency  $f_m$  is generated in a high-frequency oscillator of the EDMU, see Figure 1.6.

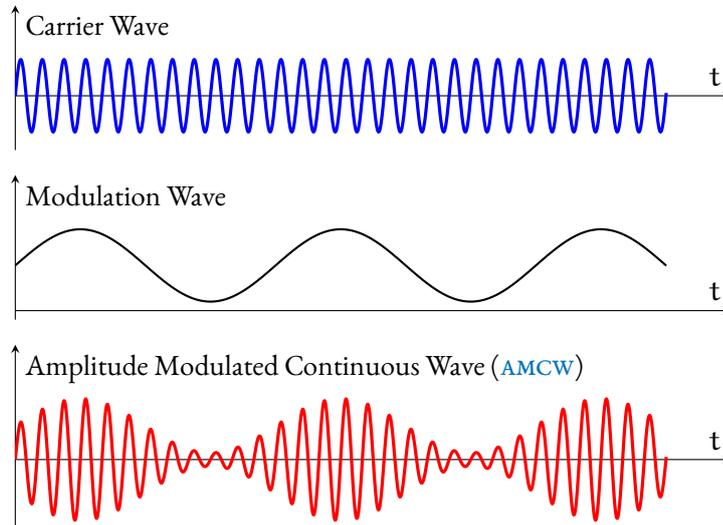


Figure 1.6: Amplitude modulation of a carrier wave.

The resulting amplitude-modulated signal (AMCW)  $A_e(t)$ , see (1.4a), is permanently emitted by the transmitting diode in the EDMU during a measurement process and the signal  $A_r(t)$ , see (1.4b), backscattered at the object is recorded phase-shifted by the receiving photodiode.

$$A_e(t) = A_{e_0} [\sin(\omega t)], \quad (1.4a)$$

$$A_r(t) = A_{r_0} [\sin(\omega t + \Delta\phi)], \quad (1.4b)$$

where

- $A_{e_0}, A_{r_0}$  maximum amplitudes of emitted and measured reflected signal
- $\Delta\phi = \phi_r - \phi_e$  phase difference between emitted signal with phase  $\phi_e$  and measured reflected signal with phase  $\phi_r$  which is equivalent to  $\Delta\phi = \omega\Delta t$
- $\omega = 2\pi f_m$  angular frequency with modulation frequency  $f_m$

Figure 1.7 illustrates the principle of the phase difference method according to the emitted and reflected signal, see (1.4). The received signal evaluated in the EDMU results in the measured maximum amplitude  $A_{r_0}$  (intensity) and the measured phase difference  $\Delta\phi$ , see (1.4b), leading to the length  $\Delta\lambda = \frac{\lambda\Delta\phi}{2\pi}$  of the shifted signal.

The distance to the object  $\rho = \frac{n\lambda + \Delta\lambda}{2}$  is calculated from the wave remainder  $\Delta\lambda$  and the number  $n$  of integer wavelengths  $\lambda$ .

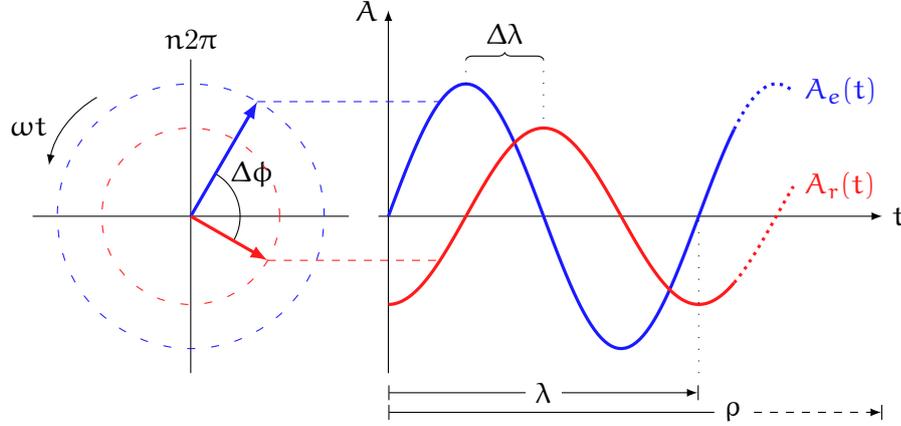


Figure 1.7: Principle of phase difference method.

Due to the unknown number of integer wavelengths, at least two modulation frequencies are applied in practice. The so-called coarse channel with  $f_{m_c}$  only serves to determine a coarse but unambiguous distance value, whereas the fine channel with  $f_{m_f}$  provides very precise but ambiguous distance values.

The wavelength  $\lambda_c$  resulting from the coarse channel limits at once the maximum measuring distance  $\rho = \frac{\lambda_c}{2}$ . The coarse distance  $\rho_c$ , see (1.5a), results, so to speak, directly as a wave remainder.

Via the coarsely determined distance, the ambiguities regarding the fine channel can be determined, resulting in the final distance  $\rho_f$ , see (1.5b).

$$\rho_c = \frac{\Delta\phi_c \lambda_c}{4\pi}, \quad (1.5a)$$

$$\rho_f = \frac{\lambda_f}{2} \left( n + \frac{\Delta\phi_f}{2\pi} \right), \quad (1.5b)$$

where

- $\Delta\phi_c, \Delta\phi_f$  measured phase differences of coarse and fine modulated carrier waves
- $\lambda_c = \frac{c}{f_{m_c}}$  wavelength of the coarse signal based on the coarse modulation frequency  $f_{m_c}$
- $n = \lfloor \frac{2\rho_c}{\lambda_f} \rfloor$  number of complete wavelengths  $\lambda_f$  of the fine signal

Another possibility of signal propagation with the phase difference method is to simultaneously amplitude-modulate the carrier signal with two sinusoidal signals of different frequencies (METTENLEITER et al. 2015, p. 10).

$$A_e(t) = A_{e_0} [\sin(\omega_c t) + \sin(\omega_f t)], \quad (1.6a)$$

$$A_r(t) = A_{r_0} [\sin(\omega_c t + \Delta\phi_c) + \sin(\omega_f t + \Delta\phi_f)], \quad (1.6b)$$

where

$A_{e_0}, A_{r_0}$	maximum amplitude of emitted signal and measured reflected signal
$\Delta\phi_c, \Delta\phi_f$	phase differences between emitted and reflected signal of coarse and fine modulated signal, see <a href="#">Equation (1.6b)</a>
$\omega_c, \omega_f$	coarse and fine angular frequencies based on coarse and fine modulation frequencies $f_{m_c}$ and $f_{m_f}$

A so-called frequency-selective calculation of the phase differences of both measuring channels results in a unambiguous and precise distance.

**FREQUENCY DIFFERENCE METHOD:** Unlike the phase difference method, see [Section 1.2.2](#), the frequency difference method frequency modulates the transmitter signal at a constant amplitude.

The Frequency Modulated Continuous Wave (**FMCW**) method, named according to the signal generation, records the difference in frequencies between the transmitted and received signal, which is proportional to the distance.

By using several such frequency modulated signals, the distance is determined unambiguously. Similar to multi waveform, see [Section 1.2.2](#), the **FMCW** method can be used to measure several targets in one spatial direction.

#### I.2.2.1 COMPARISON OF AMCW WITH TOF METHOD

In [Table 1.2](#), the **AMCW** method is compared to the **TOF** method with regard to the ambiguity range, the distance resolution and the distance uncertainty<sup>7</sup>.

The maximum unambiguous distance  $\rho_{max}$  is limited in the phase measurement by the use of the coarse modulation frequency  $f_{m_c} = \frac{c}{\lambda_c}$ , whereby the time  $t_{mi}$  after which sufficient intensity is still received for signal evaluation is decisive in the *propagation time method*. The strength of the laser pulse allows the maximum distance to be influenced in the **TOF** method, however, the Signal-to-Noise Ratio (**SNR**), having a negative effect on the signal evaluation, rises with decreasing distance.

<sup>7</sup> With distance uncertainty, the term precision of a distance measurement, used in the following, is understood.

Table 1.2: Comparison of **TOF** and **AMCW** methods according to (WEHR and LOHR 1999).

CRITERION	TOF	AMCW
unambiguous max. distance	$\rho_{\max} = \frac{c}{2} t_f$	$\rho_{\max} = \frac{\lambda_c}{2}$
distance resolution	$\Delta\rho = \frac{c}{2} \Delta t_o$	$\Delta\rho = \frac{\lambda_f}{4\pi} \Delta\phi$
distance uncertainty	$\sigma_\rho = \frac{c}{2} t_{\text{rise}} \frac{1}{\sqrt{\text{SNR}}}$	$\sigma_\rho = \frac{\lambda_f}{4\pi} \frac{1}{\sqrt{\text{SNR}}}$

For the **AMCW** method, the resolution  $\Delta\rho$  of the distance measurement is limited on the one hand by the fine modulation frequency  $f_{m_f} = \frac{c}{\lambda_f}$  – the higher the modulation frequency, the higher the resolution – and on the other hand by the phase measurement  $\Delta\phi$ .

A theoretical distance resolution of  $\Delta\rho \approx 0.05$  mm is obtained from a commonly used fine modulation frequency of  $f_{m_f} = 96$  MHz and the possibility of digital phase measurement with a resolution of up to  $\frac{1}{10000}$  of a period (DEUMLICH and STAIGER 2002, p. 163).

The resolution of the time  $\Delta t_o$  is crucial for the distance resolution of the *propagation time method*. With a signal propagation of 15 GHz, this leads to a temporal resolution of  $\Delta t_o \approx 0.066$  ns, resulting in a distance resolution of  $\Delta\rho \approx 10.0$  mm.

Concerning the distance uncertainty  $\sigma_\rho$ , the determination of the time  $t_{\text{rise}}$  of the ascending pulse is of importance for the **TOF** method, respectively the fine modulation wavelength  $\lambda_f$  for the **AMCW** method. In addition, the square root of the **SNR**, see also Section 1.2.4.1, has an inversely proportional effect on the determination of the uncertainty.

NOTE: A reduction of the noise component and thus of the stochastic effects, see Section 1.2.4.1, i. e. an improvement of the **SNR**, can be realized by repeated measurements and subsequent averaging. *Systematic error* components present in the measurements, see Section 1.2.4, are mostly compensated by the device manufacturers in the laboratory or determined in a calibration process, see Section 1.2.5.

### 1.2.3 PRINCIPLE OF ANGULAR MEASUREMENT

The two deflection units, see Figure 1.2,

- beam deflection unit with respect to tilting axis (E)
- deflection unit with respect to standing axis (B)

of modern **TLSs** are equipped with so-called encoders, that determine the respective deflection angles<sup>8</sup>,

<sup>8</sup> If the rotation axis of the instrument is aligned vertical the deflection angles  $\theta$  and  $\phi$  are named zenith angle and horizontal direction.

tilting angle  $\theta$  and direction  $\phi$  referring to the instrument system according to NEITZEL (2006), see also Section 1.3. The schematic layout of an angle encoder is shown in Figure 1.8.

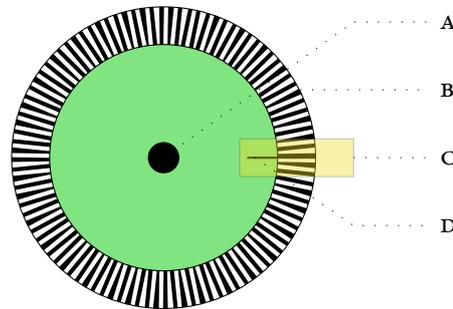


Figure 1.8: Schematic representation of an angle encoder for according to (METTENLEITER et al. 2015).

The circular encoder disk, see Figure 1.8, made of glass is mounted on the respective rotor axis (A) and has a very fine incremental radial line graduation (B) imprinted on the outer ring. The read head (C) attached on the outer ring of the encoder disk is equipped with an optical sensor capable of counting the number of lines during rotation.

The known number of lines on the disc determines the possible angular resolution<sup>9</sup>  $\Delta\theta = \Delta\phi$  of the scanner, resulting in the actual angular position

$$\theta = \phi = n_l \Delta\phi \quad \text{with} \quad \Delta\theta = \Delta\phi \quad (1.7)$$

according to the number  $n_l$  of currently counted lines. The reference point or zero point (D) of the angle measurement is marked by a separate dash, called the index, according to which the counter  $n_l$  is reset to 0 after one complete rotation.

#### 1.2.4 ERROR INFLUENCES ON MEASUREMENTS

The error budget within the TLS is very complex, according to which various influencing groups on the measured values

- influences stemming from the measurement system (errors caused by the EDMU, see Section 1.2.4.1, axis errors, see Section 1.2.4.3)
- influences arising from the signal path (atmospheric influences, see Section 1.2.4.1)
- influences stemming from the acquisition configuration (distance to the object and incidence angle), see also Section 1.2.4.2

<sup>9</sup> Modern TLS, such as the Z+F IMAGER<sup>®</sup> 5006h, are equipped with angle encoders, that have 200000 line graduations, yielding an angular resolution of  $\Delta\theta = \Delta\phi = 0.0018^\circ$ .

- influences originating from the measured object (color, material, humidity, etc.), see also [Section 1.2.4.2](#)

can be differentiated from each other. With regard to some specific error influences on the measured values during **TLS**, a general distinction is made between *random* and *systematic errors*, see also [Section 1.8.2](#). As the name suggests, *systematic errors* systematically falsify measurements, i.e. by defined amounts.

If *systematic influences* can be evidenced significantly via a suitable measurement setup, they may be transferred into an adequate *error model* in case all dependencies can be modeled. In the context of a *system calibration*, as e. g. in [Section 1.2.5](#) for a **TLS**, calibration parameters (errors from the **EDMU** and axis errors) can be finally determined and taken into account via suitable correction terms to the original measured values.

Errors of a *random* nature are caused within the device by the physical imperfection of the **EDMU** and the beam deflection unit, moreover, they include all stochastic effects caused by other influences that cannot be incorporated in a correction model.

#### 1.2.4.1 INFLUENCING FACTORS ON DISTANCE MEASUREMENT

**ERRORS CAUSED BY THE RANGEFINDER** Errors arising from the **EDMU** can be distinguished as listed below

- zero error  $k_0$
- scale error  $\Delta s$
- cyclic phase error  $\Delta c p$
- phase inhomogeneities

and sum up to the overall error

$$\rho_r - \rho_m = \Delta\rho + v_\rho = k_0 + \Delta s + \Delta c p \quad (1.8)$$

from the difference between reference  $\rho_r$  and measured  $\rho_m$  distance and a small residual error  $v_\rho$ .

**ZERO ERROR:** For reflectorless distance measurement of a **TLS**, the beam deflection system with e. g. rotating mirror causes signal delays in optics and electronics resulting in so-called *blind paths* of the measuring beam. The *blind path* is the distance the emitted signal travels from its transmitting point (electronic zero) to the zero point of the measuring instrument (mechanical zero), ideally coinciding with the **TLS**'s axes, see also [Section 1.2.4.3](#).

The constant and distance-independent path has a systematic effect on all distance values and is referred to as the zero error.

Due to the strong temperature dependence of the error<sup>10</sup> by heating effects within the instrument

$$k_0 = \rho_r - \rho_m, \quad (I.9)$$

it is continuously detected in the laser scanner via a reference distance  $\rho_r$  and the measured values  $\rho_m$  are compensated by it (METTENLEITER et al. 2015, p. 49).

**SCALE ERROR:** The scale for EDM is determined by the oscillator frequency for time measurement in the TOF method and by the modulation frequency  $f_{m_f}$  for fine measurement in the AMCW method.

The measurement frequency is falsified due to temperature-related (temperature dependence of the quartz resonance frequency) as well as operation time-dependent (ageing of the quartz and change in the operating voltage) circumstances.

The resulting deviation (frequency error  $\Delta f$ ) of the actual frequency  $f$  of the EDM compared to the nominal frequency  $f_0$  results in a distance error proportional to the distance or a scale deviation (scale error  $\Delta s$ ), which has to be taken into account by a scale correction.

The scale error  $\Delta s$  is determined by the manufacturers TLSs by comparing measured distance values  $\rho_{m_i}$  with precisely determined reference distance values  $\rho_{r_i}$  on a comparator track (interferometric measuring principle, see (JOECKEL et al. 2008)).

Using a suitable functional model<sup>11</sup>, see also Section 1.8.2,

$$\rho_{r_i} = \rho_{m_i}(1 + \Delta s) \quad \text{where} \quad \Delta s = \frac{f_0 - f}{f} = \frac{\Delta f}{f} \quad (I.10)$$

the scale error  $\Delta s$  can be determined, for example, by using a GMM, see Section 1.8.5.

**CYCLIC PHASE ERROR:** The transmitting and receiving units of the EDMU are located very close to each other, causing an electrical superposition of parts of the transmitter signal with the distance signal detected in the receiver, termed *electrical crosstalk*.

Furthermore, scattering infrared radiation can reach the receiver directly from the transmitter and be superimposed with the incoming light of the measurement. Due to the use of the same

<sup>10</sup>When correcting for the zero error  $k_0$  in the device, it is assumed that the measured distances  $\rho_m$  have already been compensated for all other systematic errors.

<sup>11</sup>In reality, the zero error and all other *systematic errors*, such as the parameters for describing the cyclic phase error, are determined simultaneously via an comprehensive functional model, otherwise the errors cannot be separated.

optical components that are equally traversed by the emitted and received beams, multiple reflections may also occur. In both cases, this is referred to as *optical crosstalk*.

The aforementioned causes act in their entirety as a cyclic phase error that occurs periodically with a multiple of the fine modulation wavelength  $\lambda_f$  and can be distance-dependent.

Accordingly, a correction model results whose parameters can be determined by measuring reference distances  $\rho_{r_i}$  with a length comparator, see e. g. similar to (DEUMLICH and STAIGER 2002, p. 166) or as part of a self-calibration (e. g. BAE and LICHTI 2007)

$$\Delta c p = A_{c_1} \sin \left( (\rho_{m_i} - \phi_f) \frac{4\pi}{\lambda_f} \right) + A_{c_2} \cos \left( (\rho_{m_i} - \phi_f) \frac{4\pi}{\lambda_f} \right) \quad (I.II)$$

where  $A_{c_1}, A_{c_2}$  are the correction amplitudes for the cyclic correction and  $\phi_f$  is the phase shift.

**PHASE INHOMOGENEITIES:** Since the transmitter diodes cannot be manufactured without errors, the modulation phase may be dependent on the transmitter location on the diode surface. This leads to the fact that not all modulated beams have the same phase position at the same time, one speaks of phase inhomogeneity (DEUMLICH and STAIGER 2002, p. 164). Using high-quality diodes combined with optical mixing of the beams before the signal is transmitted, phase inhomogeneities can be largely reduced.

**NOTE:** The root mean square (RMS) value  $\epsilon_\rho = \sqrt{\frac{1}{n} \sum_{i=1}^n v_{\rho_i}}$  of the remaining errors  $v_{\rho_i}$  after compensation of the 1, . . . , n distances to the true reference distances are specified by some manufacturers of TLSs as so-called *linearity error*, e. g. (METTENLEITER et al. 2015, p. 51), and are given for a certain distance. For the TLS Z+F IMAGER® 5006h primarily used in this thesis, this error is less than 1 mm per 50 m and indicates the accuracy of the distance measurement.

**ATMOSPHERIC INFLUENCES** Further influencing factors on the EDM are different atmospheric conditions such as

- temperature<sup>12</sup>,
- air pressure,
- and relative humidity,

<sup>12</sup>The atmospheric influences act in their magnitude as in the indicated order, whereas the temperature has the largest influence.

acting during measurements along the measuring path. These have a direct effect on the velocity of propagation  $c$  of the measuring beam and thus on the determination of the distance.

Suitable correction terms can be used to determine the so-called *refraction coefficient*  $n$  prevailing under the current meteorological conditions. Hence the correction value

$$e_n = \rho \left( \frac{n_{ref} - n}{n} \right)^{13} \quad (1.12)$$

for a distance measurement  $\rho$  results, also referred to as 1st velocity correction due to the impact of the atmospheric influences on the velocity of propagation  $c$ , where  $n_{ref} = \frac{c_0}{f_m \lambda_m}$  is the reference refraction coefficient, which depends on the modulation frequency  $f_m$  and the wavelength  $\lambda_m$ .

The correction value acts like a scale error, see also [Section 1.2.4.1](#), depending on the distance to be determined. For long distances and extreme atmospheric conditions, the error influence may be so large that it has to be taken into account.

**STOCHASTIC EFFECTS** Unknown and uncorrectable *systemic errors*, see [Section 1.2.4](#), as well as *random errors* are reflected in the distance noise, indicating the precision of the distance measurement.

Since the **EDM** is based on the analysis of the received laser light, see [Section 1.2.2](#), the distance noise (precision  $\sigma_\rho$ ) depends on the energy of the received light (amplitude  $A_{r0}$ , see [Equation \(1.4b\)](#) or [Equation \(1.6b\)](#)) ([METTENLEITER et al. 2015](#), p. 51) and is therefore also subject to all other factors influencing the intensity measurement, see [Section 1.2.4.2](#).

Due to the very complex relationships of the different influencing factors on the intensity, see [Equation \(1.15\)](#), in particular the correlation of the influence of the material properties on the influences of the incidence angle and the distance, see [Figure 1.11](#), the distance noise  $\sigma_\rho$  of a **TLS** is always indicated in practice for a certain distance  $\rho$  under an incidence angle  $\alpha$  of  $0^\circ$  and for the respective surface with a known reflectivity  $\gamma_\lambda$ .

In order to determine the **TLSs** distance noise  $\sigma_\rho$ , distances to different panels, e. g. as shown in [Figure 2.1](#), are usually determined on a measuring track by the manufacturer during factory calibration ([METTENLEITER et al. 2015](#), pp. 50-51), see [Section 1.2.5](#). By means of a large number  $n$  of single measurements  $\rho_i$ , randomly distributed around their mean value

$$\rho_m = \frac{1}{n} \sum_{i=1}^n \rho_i, \quad (1.13)$$

their deviations  $\Delta\rho_i = \rho_i - \rho_m$  from the mean value  $\rho_m$  can be determined, resulting in the typical normally distributed characteristic according to Gauss, see [Figure 1.9](#), based on their frequencies  $f$ .

<sup>13</sup> The refraction coefficient  $n_r$  in [Section 1.2.2](#) corresponds with the correction to the expression  $(1 + \frac{n_{ref} - n}{n})$

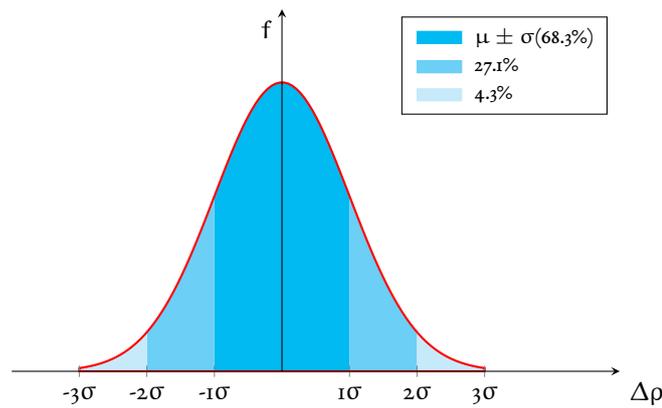


Figure 1.9: General distribution of the distance noise  $\sigma_\rho$  for the mean distance  $\rho_m$ .

Depending on the distance, incidence angle, material properties and sampling rate, see [Table 1.3](#), an individual distribution of the distance residuals is obtained. The wider the distribution, the less precise the distance measurement. The manufacturers specify the so-called  $1\sigma$  noise, see [Figure 1.9](#), as the decisive parameter.

HOWEVER due to the comprehensive factors influencing the intensity measurement, see [Section 1.2.4.2](#), and especially the complex interrelationship between the influences, compare [Equation \(1.15\)](#), hence also on the distance measurement, researches such as SOUDARISSANANE et al. (2011), ELKHRACHY and NIEMEIER (2006) and ZÁMEČNÍKOVÁ et al. (2014) consider the aspects independently of each other.

#### 1.2.4.2 INFLUENCING FACTORS ON INTENSITY MEASUREMENT

Factors that directly influence the intensity of the reflected laser light and hence on the EDM can be divided into the following three groups:

- A. Scanner properties (frequency of electromagnetic wave, beam divergence<sup>14</sup>)
- B. Scanning geometry (incidence angle, object distance)
- C. Object properties (roughness, reflective behaviour of the irradiated surface)

[Figure 1.10](#) illustrates the relationships between the influencing factors of groups A, B and C, that weaken the energy of the emitted signal strength  $A_{e0}$ .

<sup>14</sup>The laser spot size depends on the measuring distance, the angle of incidence and the beam divergence and has a direct effect on the backscattering behaviour, see also ([PESCI et al. 2011](#))

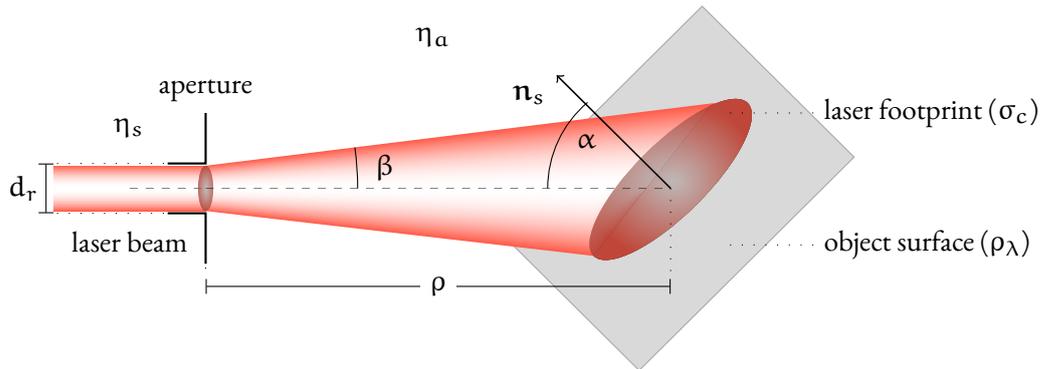


Figure 1.10: Interaction of the factors influencing the intensity with respect to the radar equation.

A functional relationship between the emitted  $A_{e_0}$  and the received signal strength  $A_{r_0}$  can be indicated by means of the radar equation

$$A_{r_0} = \frac{A_{e_0} d_r^2}{4\pi\rho^4 4\beta^2} \eta_s \eta_a \sigma_c \quad (1.14)$$

– both radar electromagnetic and laser waves follow the same principles –, see e. g. (WAGNER et al. 2006), where

$d_r$ [m]	receiver aperture diameter of the TLS
$\eta_s$	system transmission factor with respect to the optical TLS components
$\eta_a$	atmospheric attenuation factor
$\rho$ [m]	range (distance) from scanner to object
$\beta = \frac{\lambda}{\pi}$ [rad]	beam divergence depending on the wavelength $\lambda$ of the emitted light
$\alpha$ [rad]	incidence or scattering angle, defined as the angle between the incident beam and the surface normal $\mathbf{n}_s$
$\sigma_c = \gamma_\lambda A \cos \alpha$ [m <sup>2</sup> ]	backscatterer cross section comprising the product of the target reflectivity $\gamma_\lambda$ , the target area $A = \pi\rho^2 4\beta^2$ and the influence $\cos \alpha$ of the incidence angle $\alpha$ (scattering angle) on the object surface in relation to an isotropic scatterer (JELALIAN 1992)

By inserting the backscatter cross section  $\sigma_c$  into the original radar equation, see (1.14), a simplified form for a so-called laser equation is obtained:

$$A_{r_0} = \frac{A_{e_0} d_r^2 \gamma_\lambda \cos \alpha}{4\rho^2} \eta_s \eta_a \quad (1.15)$$

Obviously, (1.15) depends solely on the instrument properties  $A$  ( $d_r, \eta_s, A_{e_0}$ ) as well as an atmospheric

factor  $\eta_\alpha$ , the scanning geometry  $B(\rho, \alpha)$  and the object properties  $c(\gamma_\lambda)$ .

Assuming that the parameter  $\eta_s$  is constant related to a certain instrument (HÖFLE and PFEIFER 2007), a common characteristic parameter  $C_i = A_{e0} d_r^2 \eta_s$  for the TLS can be specified with the other constant instrument-related parameters  $d_r$  and  $A_{e0}$ .

At least during a TLS capture, the atmospheric conditions, see also Section 1.2.4.1, remain almost constant, thus the factor  $\eta_\alpha$  is to be regarded as conditionally constant. In a further simplified representation, the received signal amplitude

$$A_{r0} \propto \frac{C_i \gamma_\lambda \cos \alpha \eta_\alpha}{4\rho^2} \quad (1.16)$$

by the constancy of the other terms emphasizes the proportionality to the material reflectivity  $\gamma_\lambda$  and to the impact of the incidence angle by  $\cos \alpha$ , as well as the reversed proportionality to the square of the range  $\frac{1}{\rho^2}$ .

In order to verify the theoretical regularities of the distance with  $\frac{1}{\rho^2}$ , the angle of incidence with  $\cos \alpha$ , as well as the surface reflectivity  $\gamma_\lambda$  with regard to the intensity behavior, panels with different radiometric properties, see Figure 2.1, can be measured at defined distances and at different incidence angles in the *iD*-mode<sup>15</sup> of the TLS.

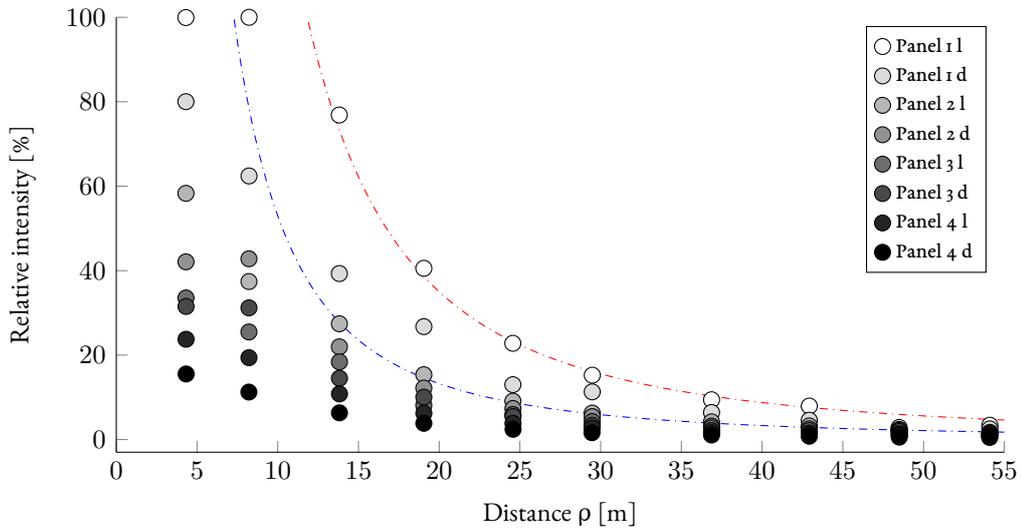


Figure 1.11: Intensity behaviour of the TLS Z+F IMAGER® 5006h as a function of the measured distance  $\rho$  for different panels, each with different reflectivity..

<sup>15</sup> Most TLSs do not provide a *iD*-mode due to the loss of eye safety from a static laser beam, but profile *2D*- or *3D* modes can be used alternatively.

Figure 1.11 shows the mean intensity values

$$I_m = \frac{1}{n} \sum_{i=1}^n I_i \quad (1.17)$$

obtained from the  $i = 1 \dots n$  measured individual intensities  $I_i$  in percent for the panels in Figure 2.1 as a function of the distance  $\rho$ . The blue and red curves in Figure 1.11 illustrate the percentage decrease in intensity for the 1l and 2l panels and confirm the theoretical loss in intensity by the ratio  $\frac{1}{\rho^2}$ .

Below 10 m, the data values no longer follow this trend, as the scanner in this area seems to slightly attenuate the very high signal intensity in order to avoid overmodulation (BLASKOW and SCHNEIDER 2014). Basically, darker panels feature lower intensity values compared to lighter ones due to their lower reflectivity – the darker the panels, the lower the reflection behaviour and hence also the intensity. The percentage decrease in intensity due to different radiometric properties, e. g. of panel 1l and panel 4d with more than 80%, is highest at shorter distances.

The influence of the incidence angle on the intensity is shown in Figure 1.12. The graph shows the evaluation of the intensity of a panel with average reflectivity at 10 and 30 m by varying the incident angle  $\alpha$ .

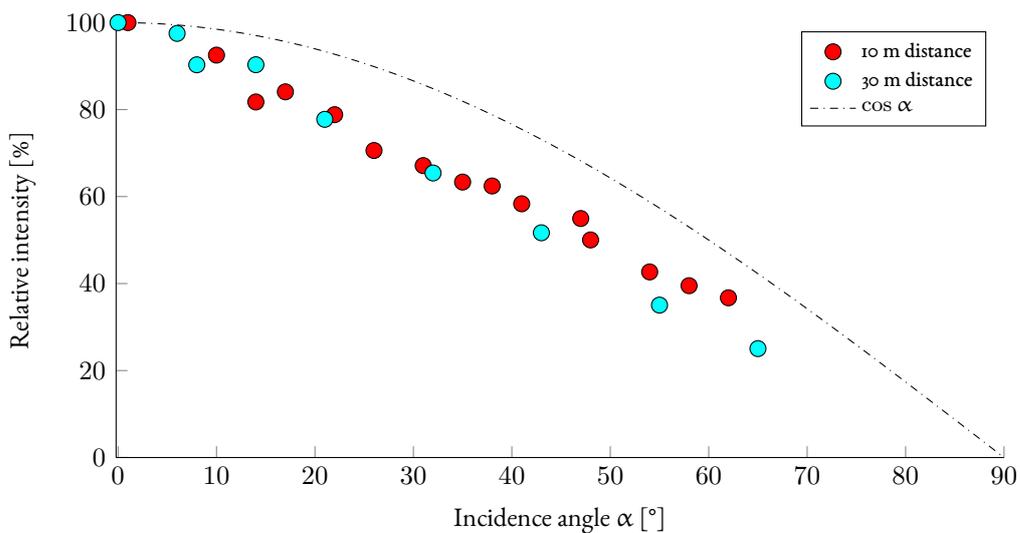


Figure 1.12: Intensity behaviour of the TLS Z+F IMAGER<sup>®</sup> 5006h as a function of the incidence angle  $\alpha$  at different distances..

The intensity behavior, or the intensity drop with increasing incidence angle, for both distances is largely linear but still slightly different from the degree of decrease, so that the influence of the incidence angle  $\alpha$  is not strictly independent of that of the distance. Furthermore, the behavior is not comparable to the theoretical one of  $\cos \alpha$ .

Even if *systematics* in the decrease of the intensity with increasing distance  $\rho$  as well as with incident angle  $\alpha$  can be observed, the effects cannot be represented in a generic model. On the one hand, due to a slight correlation between distance and incident angle, see [Figure 1.12](#), as well as the very different drop behavior of the intensity at different radiometric surface properties, see [Figure 1.11](#). Further effects not yet considered, such as object roughness and the material itself, lead to more unknown factors and dependencies, making modeling impossible without prior knowledge.

IN THE CASE OF purely *systematic effects*, as [Section 1.2.4.1](#), which can be significantly detected and separated by means of a suitable measurement arrangement, a modeling and thus also an evaluation in a *mathematical model*, such as [GMM](#) or [GHM](#), is possible. However, since the effects are not purely *systematic* but also *random* and, moreover, cannot be modeled, they are all reflected in the intensity decrease.

SINCE all influences are reflected in the intensity and also affect the distance noise, as shown in [Section 1.2.4.1](#), as well as the fact that the distance measurement is based on the evaluation of the backscattered intensity ([METTENLEITER et al. 2015](#), p. 51), the question arises whether a general relationship between the intensity and noise behavior of the distance measurement can be established, see the findings in [Chapter 2](#).

#### 1.2.4.3 INFLUENCING FACTORS ON ANGLE MEASUREMENT

As with distance measurement, cf. [Section 1.2.4.1](#), there are also error components in angle measurement, which are referred to as *axis errors* in the following.

AXIS ERRORS are systematic deviations of the axes within the local coordinate system of a polar measuring system, such as a *total station* or *laser scanner*, see also [Section 1.3](#). According to DEUMLICH and STAIGER (2002) p. 205-206 and 211-212, an error-free idealized polar measuring system is present if

- the tilting axis **ta** is perpendicular to the rotation axis **ra**,
- the collimation axis **ca** is normal to the tilting axis **ta** and
- the collimation axis **ca** intersects the rotation axis **ra**.

In addition to the above enumeration, DEUMLICH and STAIGER (2002) mention that the rotation axis **ra** should remain strictly plumb, however, this is not an internal instrument error, rather a set-up error. Since *laser scanning*, in contrast to *tacheometry*, the transfer of individual standpoints due to different evaluation methods usually takes place in a local coordinate system at the beginning, a strict vertical rotation axis is not necessarily required. Especially since today's compensators ensure a safe perpendicular positioning in case of rough levelling.

These systematic deviations from the ideal are illustrated in [Figures 1.13A to 1.13C](#) and are referred to subsequently as *tilting axis error*  $i$ , *collimation axis error*  $c$  and *collimation axis eccentricity*  $e$ .

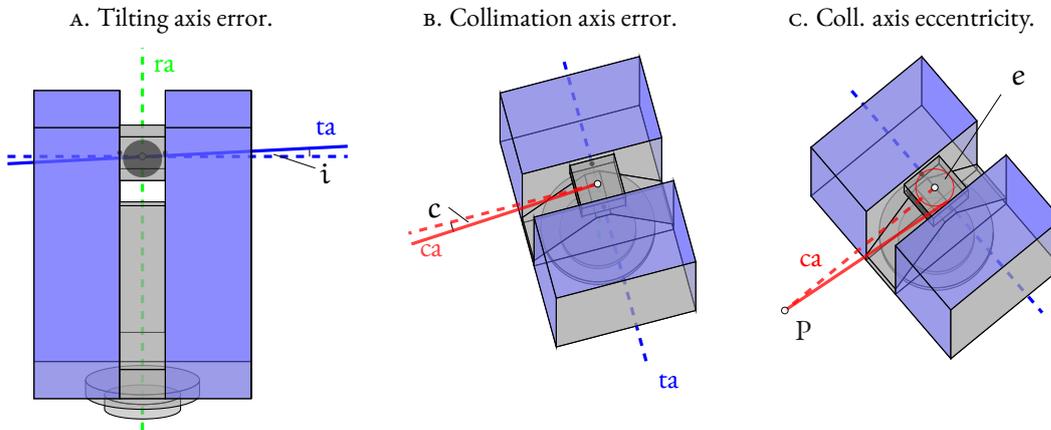


Figure 1.13: Axis errors for a TLS based on NEITZEL (2006).

From the definition of the axis errors according to [Figure 1.13A](#), the following angle conditions between the axes

$$\alpha_{tra} - i = 90^\circ \quad (1.18a)$$

$$\alpha_{tca} - c = 90^\circ \quad (1.18b)$$

$$\alpha_{rca} = \theta - h \quad \text{with } h \text{ as } \textit{height index error} \quad (1.18c)$$

result with

$\alpha_{tra}$  angle between tilting and rotation axis,

$\alpha_{tca}$  angle between tilting and collimation axis and

$\alpha_{rca}$  angle between rotation and collimation axis depending on direction measurement  $\theta$ .

For the determination of the angular errors, e. g. also for the Z+F IMAGER<sup>®</sup> 5006h, reference targets of a test field measured with high accuracy are used, which are compared with the measurements extracted from the *laser scan* data (METTENLEITER et al. 2015, p. 55). As with the distance correction in [Section 1.2.4.1](#), the differences can be included in a *correction model* and the angular errors, corresponding to axis errors  $i$  and  $c$ , can be determined accordingly, whereby the RMS errors  $\epsilon_\theta$  and  $\epsilon_\phi$  resulting from the calibration specify the angular accuracy of the TLS.

### 1.2.5 CALIBRATION OF TLSs

The aim of instrument calibration is to determine the internal *systematic error* components of a measuring device, in this case the errors caused by the EDMU, see [Section 1.2.4.1](#), and axis errors, see [Section 1.2.4.3](#), of a TLS. The calibration parameters or *systematic errors* are modelled as a function of the actual measured values and possible additional parameters in a comprehensive *functional model* (p. 1, para. 2, cl. 1, [PETROVIC 2003](#)), here *calibration model*, so that ideally only random error components, see also [Section 1.2.4.1](#), remain.

**NOTE:** In order to determine valid calibration parameters, suitable measurement configurations and evaluation methods that guarantee sufficient error impacts are necessary.

In principle, a distinction can be made between two calibration types in the context of TLS:

1. individual and independent determination of errors within distance and direction measurement
2. combined determination of all instrumental error components as part of a system self-calibration

Independent calibration parameters regarding the EDMU of a TLS can be determined by measuring and comparing reference distances, see also [Section 1.2.4.1](#), ideally on a length comparator such as those used by most instrument manufacturers.

The separate determination of the axis errors can be carried out according to NEITZEL ([2006](#)) referring to a vectorial functional description<sup>16</sup> of the errors by STAHLBERG ([1997](#)). An overview of procedures for *component* as well as *system calibration* of TLSs is given by SCHULZ ([2008](#)).

With regard to the joint determination of the calibration parameters for the EDMU as well as the angle measuring unit, ([RIETDORF et al. 2004](#); [RIETDORF 2005](#)) proposes a procedure in the wake of a concatenated transformation of *laser scans* based on observed planar patches.

A slightly different functional error approach inspired by LICHTI ([2007](#)) for self-calibration of TLSs is used by BAE and LICHTI ([2007](#)).

### 1.2.6 INTERACTION OF MEASUREMENT RESOLUTION AND -QUALITY

The angular resolution of a laser scanner, here TLS Z+F IMAGER<sup>®</sup> 5006h, is directly related to the sampling rate, resulting in various combinations according to [Table 1.3](#).

<sup>16</sup>The determination of the axis errors in this functional relationship presupposes the measurement of both faces in analogy to a tacheometer.

Table 1.3: Combination of various measuring rates with angular resolutions for a TLS Z+F IMAGER<sup>®</sup> 5006h (based on METTENLEITER et al. 2015, p. 49).

ANGULAR RESOLUTION [°]	SAMPLING RATES [kHz]			
	LOW	NORMAL	HIGH	PREMIUM
Medium (0.072)	254	127	63.5	63.5
High (0.036)	508	254	127	63.5
Super high (0.018)	1016	508	254	127
Ultra high (0.009)	1016	1016	508	254

The angular resolution is the parameter that influences the degree of discretization at the object, which is also distance and object dependent (angle of incidence). The basic measuring frequency of the TLS used is a constant 1016 kHz for all sampling rates. However, the sampling rate enables the precision to be affected. For a sampling rate of 256 kHz, a single target point can thus be measured 4 times, resulting in a approx.  $\sqrt{4}$ -times lower distance measurement noise after averaging, cf. Figure 2.6.

Depending on the measuring task and effort estimation, taking into account recording time<sup>17</sup> and quantity of data, the suitable combination of quality (sampling rate) and resolution at the object (angular resolution) can be selected according to Table 1.3.

THE KNOWLEDGE about the sampling rate is hereafter of major importance with respect to the derivation and utilization of the proper *stochastic model*, cf. Chapter 2.

### 1.3 COORDINATE SYSTEMS IN THE CONTEXT OF TLS

With regard to TLS, several coordinate systems are used, which are presented here for completeness and further understanding for the following chapters.

SCANNER COORDINATE SYSTEM The coordinate system of a TLS is a 3D cartesian right-handed coordinate system. This local scanner coordinate system can be arbitrarily oriented, depending on how the scanner is mounted on the tripod, see Figure 1.14.

The three-dimensional Cartesian coordinates of a point can be calculated from the current directional values  $\phi$ ,  $\theta$  and the distance  $\rho$  according to Figure A.2 by Equation (A.13).

<sup>17</sup>The recording time is proportional to the resolution and quality. For far away objects, a higher angular resolution coupled with a better quality (lower sampling rate) is therefore recommended.

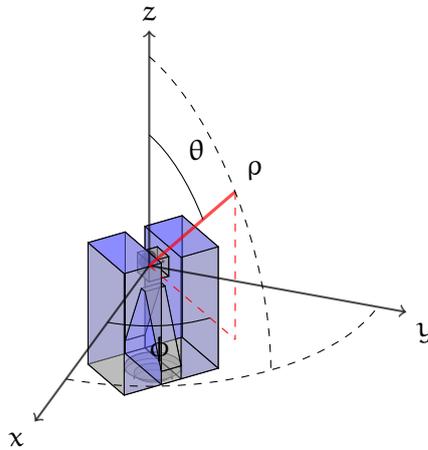


Figure 1.14: Local coordinate system of a TLS.

**PIXEL COORDINATE SYSTEM** Due to the polar operating principle resulting in regular incremental scanning –  $\Delta\phi$ ,  $\Delta\theta$  – by TLSs of its environment, see Figure 1.3, the measured values can be mapped into a grid structure, see Figure 1.15.

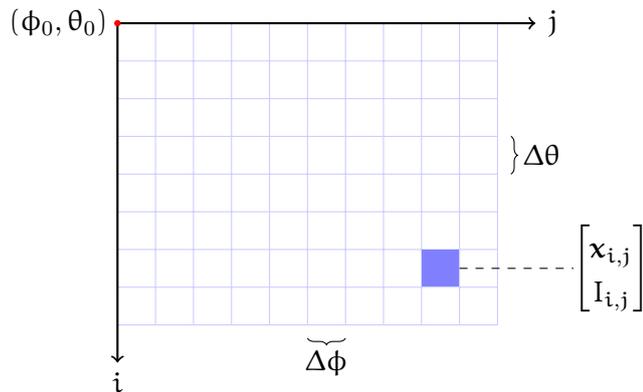


Figure 1.15: Pixel coordinate system after scanning of a TLS.

This raster structure can be interpreted as a *pixel coordinate system* and can be spanned for both the intensity values  $I_{i,j}$  and the resulting geometry, the coordinate values  $\mathbf{x}_{i,j}$ , see also Figure 1.18.

The origin of this coordinate system is defined by the beginning of the scan ( $\phi_0 = 0$ ,  $\theta_0 = 0$ ). It is spanned by the positive column  $j$  and row axis  $i$  and is  $n \times m$  in size, with  $n$  the size of a scanline  $i$  and  $m$  the number of scanlines  $j$ .

**REMARK:** The data structure corresponds to a matrix or raster data structure as defined in Section 1.4.

## I.3.1 OUTER ORIENTATION IN THE CONTEXT OF TLS

The relation of the sensor coordinate system, see Section 1.3, to a superordinate coordinate system is called *outer orientation*<sup>18</sup> within the scope of TLS. Figure 1.16 shows several local scanner coordinate systems, like in Section 1.3 for each scan station  $i$  in relation to a superordinate coordinate system.

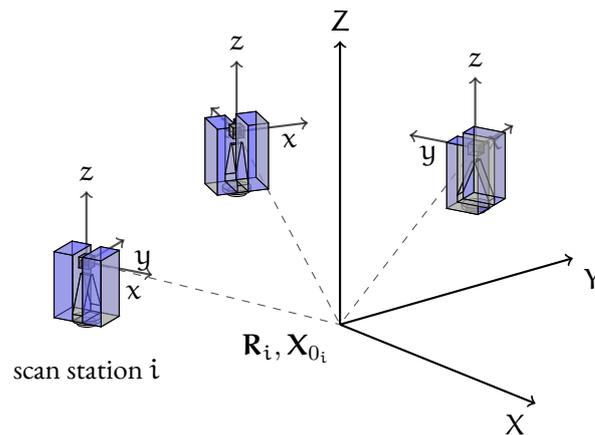


Figure 1.16: Outer orientation of TLS standpoints with respect to the superordinate system.

By means of the *outer orientation* of a scanner standpoint  $i$ , the measured values or local coordinates  $\mathbf{x}$  can be transformed into the superordinate coordinate system using  $\mathbf{X} = \mathbf{X}_{0_i} + \mathbf{R}_i \mathbf{x}$ <sup>19</sup>.

REMARK: In the course of a concatenated transformation of *laser scans*, in Section 4.5, *exterior orientations* of individual scanner stations are determined.

## I.4 SPATIAL DATA REPRESENTATION AND INDEXING

Depending on the measuring device used to acquire spatial data or whether these data have already been further processed, e. g. in the context of referencing, they are available either

- A. structured (e. g. topology by grid or matrix, *octree*, *kd-tree*) or
- B. unstructured (no topology as non-structured *point cloud*).

<sup>18</sup> The term outer orientation or exterior orientation stems from the photogrammetric domain and is mostly used within the context of bundle adjustment.

<sup>19</sup> The formulation is similar to the spatial similarity transformation, see Section 1.7, whereby the scale is assumed to be constant at 1 due to the true to scale measurements of the TLS.

If the 3D data is available in unstructured form, it must be structured hierarchically in order to carry out spatial neighbourhood analyses such as labelling, see also Section 1.5. In forming the most commonly used hierarchical 3D data structure (SAMET 2006, p. 211), an *octree*, the environment is subdivided into cubic volumes, so-called *voxels*, see Figure 1.17A.

**OCTREE:** An octree (lat. *octo* – eight and engl. *tree*) is a data structure in computer science, whose rooted tree has *nodes* with either eight direct *children* or none at all.

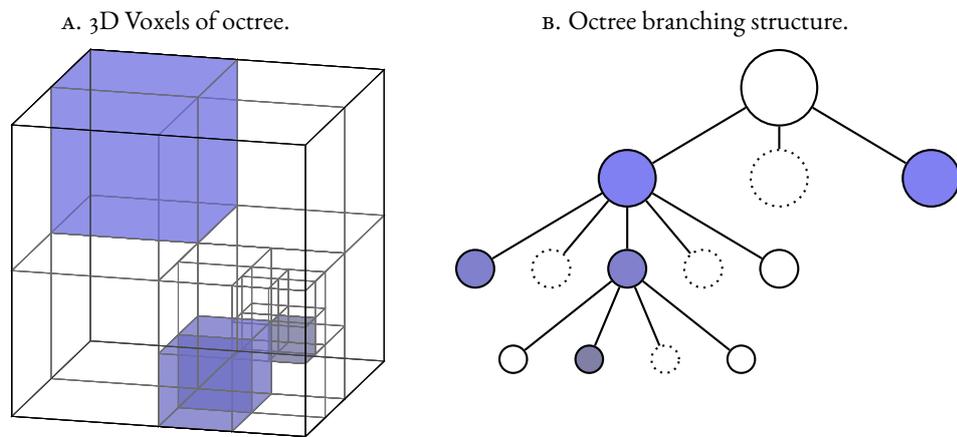


Figure 1.17: Formation of an octree structure for unstructured data.

For a given *point cloud* volume, the space is recursively divided into eight voxels according to Figure 1.17A until a desired level of detail or voxel size is reached.

**NOTE:** The more information content, i. e. the denser the 3D points, the higher the level of detail of the object is represented and the finer the subdivision.

This subdivision can be represented as a so-called tree structure, see Figure 1.17B, in which a fast and efficient search can be carried out for e. g. a neighbourhood analysis.

By storing the number of scan lines and their length while scanning with a *TLS*, the 3D data can be stored as a regular grid, see Section 1.3 and Figure 1.15, resulting in a structured data form reduced in complexity by one dimension.

Each four corresponding measured values  $x_{i,j}$ ,  $y_{i,j}$ ,  $z_{i,j}$  and  $I_{i,j}$  are stored at their pixel position, which leads to the layer structure in Figure 1.18, whereby the coordinate values were previously calculated from the original *polar elements*  $\theta_{i,j}$ ,  $\phi_{i,j}$  and  $\rho_{i,j}$ , see Equation (1.1).

#### I.4. SPATIAL DATA REPRESENTATION AND INDEXING

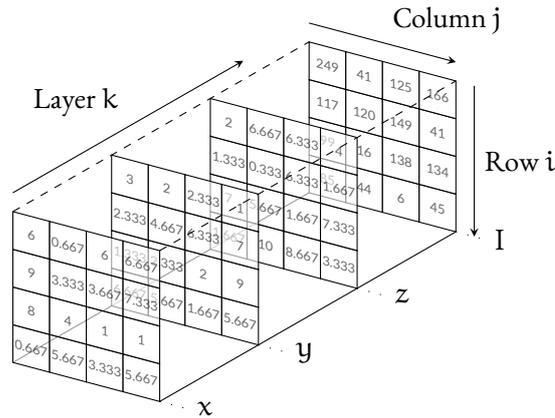


Figure I.18: Information of the TLS represented in different Data Layers.

REMARK: If the original data structure is used by the scanning method of a TLS, see also Figures I.3 and I.14, the formation of an *octree*, see Figure I.17, can be omitted, since the known adjacencies result in a more or less direct tree structure, cf. Section I.5.

In *MatLab*, the layer structure according to Figure I.18, as well as further information of the scan, such as dimensions or additional information, can be stored in a so-called structure array, see Figure I.19.

STRUCTURE ARRAY: A structure is a data type that groups related data using data containers called fields. Each field can contain data of any type or size.

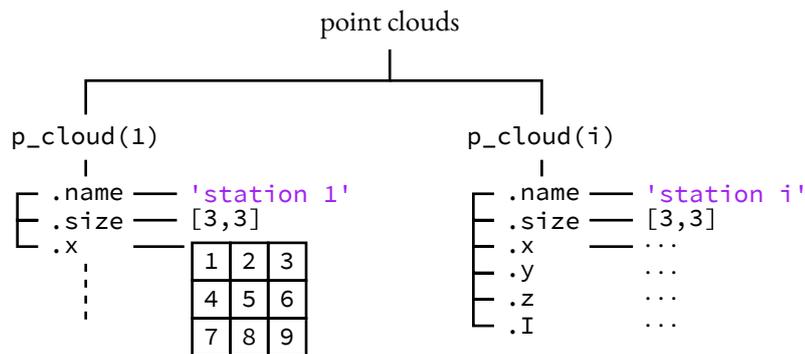


Figure I.19: Representation of a point cloud structure array in *MatLab*.

In *MatLab* syntax, the structure according to Figure I.19 for `p_cloud(1)` results as follows by Code I.1.

Code 1.1: Filling a point cloud structure in MatLab.

```

1 p_cloud(1).name = 'station 1';
2 p_cloud(1).size = [3,3];
3 p_cloud(1).x = [1,2,3;4,5,6;7,8,];

```

Access to, for example, the x-component of the `p_cloud` data structure with index 1, can then be accessed very easily by `p_cloud(1).x`.

## 1.5 GRAPH THEORY AND CONNECTED COMPONENT LABELING

Labeling `CCs`<sup>20</sup> is one of the most used and efficient method in Computer Vision (`CV`) to detect contiguous regions, mostly in digital images but also data of higher dimensionality (`SAMET and TAMMINEN 1988`), see also [Section 1.4](#).

`SHAPIRO (1996)` defines Connected Component Labeling (`CCL`) as an operator whose

input is a binary image and [...] output is a symbolic image in which the label assigned to each pixel is an integer uniquely identifying the connected component to which that pixel belongs.

`CCL` can be understood as an application of *graph theory*.

**GRAPH DEFINITION:** A graph  $G$  is a pair  $G = (V, E)$  where  $V$  is a finite set of vertices, also called nodes or points of  $G$  and  $E \subseteq \{\{x, y\} \mid (x, y) \in V^2 \wedge x \neq y\}$  is a subset, the edges, also called links or lines – i. e. an edge is associated with two distinct vertices – of  $G$  (`BENDER and WILLIAMSON 2010`).

On the basis of so-called vertices, containing the information<sup>21</sup>, and connecting edges, representing the adjacencies<sup>22</sup>, a certain graph can be constructed, see also [Figure 1.17](#).

For the labeling, i. e. the identification of connected regions, a special algorithm traverses this *graph* and labels the vertices on the basis of the connectivity and the relating neighboring values described by a heuristic<sup>23</sup>.

<sup>20</sup>`CCL` is also often used synonymously with Connected Component Analysis (`CCA`), region labeling or region extraction and blob extraction or blob discovery.

<sup>21</sup>For the `CCL`, various types of information, such as `3D` data in an *octree* data structure, see also [Figure 1.17A](#), or various image information as `2D` raster can be considered.

<sup>22</sup>Depending on the medium, either `3D` or `2D`, the connectivity is determined. For example, raster structures such as images can have a 4- (direct) or 8- connected neighborhood, see [Figure 3.5](#), whereas a *voxel* structure in an *octree*, see also [Figure 1.17A](#), can have either a 6- (directly adjacent faces), 18- (adjacent faces and edges) or 26- (adjacent faces, edges and corners) connected neighborhood.

<sup>23</sup>A heuristic concerning a voxel structure with `3D` points can be a distance criterion between points of two adjoining voxel cells or concerning a raster structure for example the difference of the grey values of adjoining pixels.

If the problem is transferred to a grey-value image, see [Figure 1.20A](#), a binary image can be obtained by simple thresholding, [Figure 1.20B](#)<sup>24</sup>.

REMARK: The pixels that correspond to a defined gray value range are classified via thresholding.

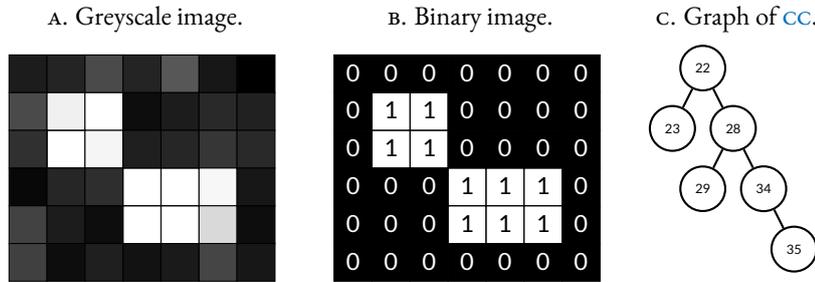


Figure 1.20: From a greyscale image via a binary representation to a graph representation.

The addressed vertices or nodes represent the foreground pixels marked with 1, where 0 represents the background pixels. The objective is to assign the foreground pixels of a defined neighborhood, here a direct connectivity, to a region respectively to a label. If one starts from the left upper pixel (linear index 22) of the right pixel cluster, the *graph* in [Figure 1.20C](#) results, which must be traversed for complete labeling of the contiguous pixel group. The mentioned edges represent the connections to the vertices, denoted by their linear indices 22 to 35, based on the neighborhood relationship.

For labeling by traversing such a *graph* there are basically two different approaches, named *graph traversal methods*:

1. Depth First Search (DFS)
2. Breadth First Search (BFS)

However, the main difference between the two methods of navigating through a *graph* lies in the different search order. DFS, as the name implies, visits the child vertices before visiting the sibling vertices, whereas the reverse rule applies to the BFS.

In practice, different data structures are used to represent the neighbourhood relations of individual vertices in *graphs* (CORMEN et al. 2001, pp.528–531):

- Adjacency list
- Adjacency matrix

<sup>24</sup>Based on the Matlab example from the connected component labeling blog of Steve Eddins: <https://blogs.mathworks.com/steve/2007/03/20/connected-component-labeling-part-3/>

- Incidence matrix

In the following the **2D adjacency matrix** is used, which has rows and columns for each *graph* node. [Figure 1.21B](#) shows the graphical representation of the *adjacency matrix*, see [Figure 1.21A](#), from the example in [Figure 1.20B](#) for a direct neighbourhood. A non-zero element indicates a connection between the nodes and thus defines the neighborhood.

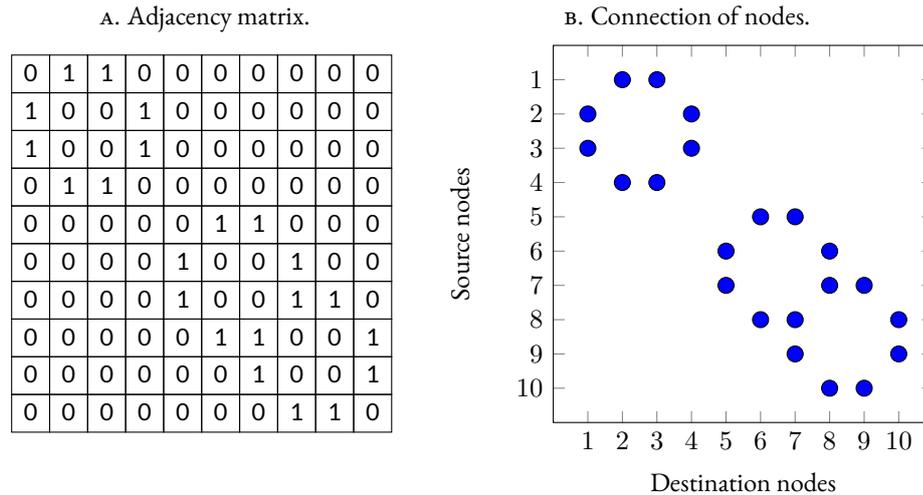


Figure 1.21: Adjacency matrix and visualization from the direct connections of a binary matrix.

In general, a distinction is made between two ways of processing algorithms for labeling **CCs**:

1. One pass over the image
2. Two passes over the image

While one pass algorithms visit each pixel only once, two pass algorithms work in two steps. Both *graph traversing*, see example in [Figure 1.20](#) and [Code 1.2](#), and *flood* or *seed filling*, see also [Code A.6](#), are among the one pass algorithms.

Code 1.2: Labeling connected components of a graph.

```

1 function CC = get_cc_of_graph(BW, conn)
2 %input:  binary matrix BW and connectivity conn 4 or 8
3 %output: struct CC of indices for connected components
4
5 % pad BW with zeros to ensure neighbourhood of border pixels
6 BW = padarray(BW,[1 1],0,'both');
7
8 [M,~] = size(BW);

```

```

9
10 % linear connectivity neighbour offsets (n_off)
11 if isequal(conn,4)
12     n_off = [-1;M;1;-M]; %direct connectivity
13 elseif isequal(conn,8)
14     n_off = [-1;M;1;-M;M+1;M-1;-M+1;-M-1]; %full connectivity
15 end
16
17 % numbering of nodes based on find functions linear indexing
18 lin_ind = find(BW); n = length(lin_ind);
19 nodes = zeros(size(BW)); nodes(BW) = (1:n);
20
21 % find all the nonzero neighbours
22 neighb_idx = bsxfun(@plus,lin_ind,n_off');
23 neighb = BW(neighb_idx);
24
25 % pairwise connections
26 lin_pix_ind_r = repmat(lin_ind,conn,1);
27 pairs = [nodes(lin_pix_ind_r(neighb)),nodes(neighb_idx(neighb))];
28
29 % sparse adjacency matrix
30 A = sparse([pairs(:,1);(1:n)'],[pairs(:,2);(1:n)'],1);
31
32 % Dulmage-Mendelsohn decomposition of adjacency matrix
33 [p,~,r,~] = dmperm(A); lin_ind_u = lin_ind(p);
34
35 % unpad array
36 [ri,ci] = ind2sub(size(BW),lin_ind_u); BW = BW(2:end-1,2:end-1);
37 lin_ind_u = sub2ind(size(BW),ri-1,ci-1);
38
39 for i = 1:length(r)-1
40     CC.PixelIdxList{i} = lin_ind_u(r(i):r(i+1)-1);
41 end

```

If the function `get_cc_of_graph` is applied to the binary image `BW` in [Figure 1.20B](#) with the neighborhood parameter `conn = 4`, a structure `CC` with linear indices of the pixel-connected regions is obtained. Access to the indices `[8;9;14;15]` of the first region is obtained with `CC.PixelIdxList{1,1}`, likewise access to the indices `[22;23;28;29;34;35]` of the second region with `CC.PixelIdxList{1,2}`, see also [Figure 1.20C](#). If, in addition, an extended neighbourhood with `conn = 8` is chosen, the present example results in a single region.

The graph-based labeling algorithm is structured in the following steps:

1. Define and number nodes of foreground pixels in the graph (lines 17 to 19)
2. Find all the pairwise node connections (lines 21 to 27) based on the neighbourhood definition `n_off` (lines 10 to 15)
3. Build up a *graph representation*, i. e. *adjacency matrix* (lines 29 to 30), from the pairwise connections `pairs`

4. Perform *equivalence class resolution*<sup>25</sup> by computing the Dulmage Mendelsohn Decomposition (DMD)<sup>26</sup> of *adjacency matrix*  $A$  (lines 32 to 33) in order to get the *CCs*

HARALICK and SHAPIRO (1992) describes at pp. 28-48 a simple two pass algorithm in which temporary labels are assigned to the foreground pixels in the first step in order to subsequently relabel them in the second step based on the mapping of *equivalence class resolution*. An extension of this technique, in which the *equivalence class resolution* is performed in the first pass with a *union find* method, is described by SEDGEWICK (1998) at pp. 11-20.

The labeling of *CCs* can be used in the context of region-based segmentation to find contiguous regions with pixels of similar characteristics. For this purpose, an optimization of a *graph*-based approach is presented in Section 3.3.2.

## 1.6 REPRESENTATION OF ROTATIONS IN 3D SPACE

Rotations in  $\mathbb{R}^3$  are generally transformations and serve amongst others to describe the spatial orientation of different sensors, see also Section 1.3.1, in a coordinate system, as well as to transform 3D data into a common coordinate system, e. g. in the context of a *spatial similarity transformation* as in Section 1.7, and are realized via so-called *rotation matrices* such as

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (1.19)$$

The quadratic rotation matrix  $\mathbf{R}$  is defined as an orthogonal matrix and must consequently fulfil the characteristic

$$\mathbf{R}^T \mathbf{R} = \mathbf{R} \mathbf{R}^T = \mathbf{I}, \quad (1.20)$$

where  $\mathbf{I}$  represents the unit matrix. In addition, two further characteristics

$$\mathbf{R}^T = \mathbf{R}^{-1}, \quad \det \mathbf{R} = 1 \quad (1.21)$$

result from orthogonality. From the demand for orthogonality, see Equations (1.20) to (1.21), six constraints for the nine elements of the *rotation matrix* arise

<sup>25</sup>The result of equivalence class resolution is the mapping information of subsets of temporary labels to the same object.

<sup>26</sup>If the matrix is an *adjacency matrix*, the *CCs* of a *graph* can be calculated by applying the DMD.

$$\begin{aligned}
 r_{11}^2 + r_{12}^2 + r_{13}^2 &= 1, \\
 r_{21}^2 + r_{22}^2 + r_{23}^2 &= 1, \\
 r_{31}^2 + r_{32}^2 + r_{33}^2 &= 1,
 \end{aligned}
 \tag{1.22a}$$

and moreover

$$\begin{aligned}
 r_{11}r_{21} + r_{12}r_{22} + r_{13}r_{23} &= 0, \\
 r_{21}r_{31} + r_{22}r_{32} + r_{23}r_{33} &= 0, \\
 r_{11}r_{31} + r_{12}r_{32} + r_{13}r_{33} &= 0,
 \end{aligned}
 \tag{1.22b}$$

which are referred to as orthogonality conditions<sup>27</sup>. With respect to  $\mathbf{R}$ , the Equations (1.22a) to (1.22b) mean that each row represents a unit vector and each column is orthogonal to each other. A point  $\mathbf{x}_p$ , ( $p = 1, \dots, n$ ), in a local system can thus be transferred via  $\mathbf{R}$  to a superior system

$$\mathbf{X}_p = \mathbf{R}\mathbf{x}_p, \quad \mathbf{x}_p = \mathbf{R}^T\mathbf{X}_p$$

and vice versa. Within an adjustment (cf. Section 1.8) the nine parameters of the *rotation matrix* are usually not determined directly, rather by specific sets of parameters (*rotation parameters*, see Sections 1.6.1 to 1.6.2). These differ in

- the number of parameters,
- the potential occurrence of singularities,
- the uniqueness,
- the geometrical interpretation and
- the computational effort (STALLMANN 2008).

Generally, a *rotation matrix* can be described by three independent parameters, as in Section 1.6.1 when using EULER angles, hence when using more parameters, as e. g. in Section 1.6.2 for parameterization with *Quaternions*, dependencies among the parameters exist, which have to be considered as constraints in the context of an adjustment, cf. Section 1.8.

---

<sup>27</sup>Strictly speaking, the six orthogonality conditions represent three orthogonality and three normalization conditions (KRAUS 2004, p. xx)

However, if the nine elements of  $\mathbf{R}$  shall be determined directly as *rotation parameters*, the six *orthogonality conditions* – Equations (1.22a) to (1.22b) – must be considered as *constraints*, cf. Section 1.8, in order to describe the interdependencies of the parameters within an adjustment.

This issue is very interesting because the parameters  $r_{ij}$  in the overdetermined case can be calculated from a linear adjustment in the course of an *affine transformation* (FANG 2015), thus providing good approximations to determine the parameters of the *spatial similarity transformation*, see also Section 1.7, by introducing the aforementioned *constraints*.

### 1.6.1 PARAMETERIZATION USING EULER ANGLES

The EULER angles<sup>28</sup> ( $\omega, \phi, \kappa$ ) according to *Leonhard Euler* (EULER 1775) represent the three rotation angles in sequence around the coordinate axes ( $X, Y, Z$ ). The entire rotation  $\mathbf{R}$  results from the product of three single rotations  $\mathbf{R}_\omega, \mathbf{R}_\phi$  and  $\mathbf{R}_\kappa$  with

$$\mathbf{R}_{\omega\phi\kappa} = \mathbf{R}_\omega \mathbf{R}_\phi \mathbf{R}_\kappa.$$

Considering the three single rotations

$$\mathbf{R}_\omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \omega & -\sin \omega \\ 0 & \sin \omega & \cos \omega \end{bmatrix}, \quad \mathbf{R}_\phi = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix}, \quad \mathbf{R}_\kappa = \begin{bmatrix} \cos \kappa & -\sin \kappa & 0 \\ \sin \kappa & \cos \kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

the full rotation reads

$$\mathbf{R}_{\omega\phi\kappa} = \begin{bmatrix} c\phi c\kappa & -c\phi s\kappa & s\phi \\ c\omega s\kappa + s\omega s\phi c\kappa & c\omega c\kappa - s\omega s\phi s\kappa & -s\omega c\kappa \\ s\omega s\kappa - c\omega s\phi c\kappa & s\omega c\kappa + c\omega s\phi s\kappa & c\omega c\phi \end{bmatrix} \quad (1.23)$$

with the abbreviations  $c$  for  $\cos$  and  $s$  for  $\sin$  respectively. The rotation sequence is of decisive importance when using rotation matrices and defines the order in which the individual rotations take place. Rotations around *co-rotated* axes – e. g. (1.23) – refer to the previously reached coordinate system, whereas the single rotations of *space-bound* axes are related to the initial system (STALLMANN 2008). As a consequence, the rotation around *fixed* coordinate axes is swapped mathematically in comparison to *co-rotated* such as  $\mathbf{R}_{\kappa\phi\omega} = \mathbf{R}_\kappa \mathbf{R}_\phi \mathbf{R}_\omega$ .

<sup>28</sup>The EULER angles (EULER 1775) are also often referred to as CARDANO angles (CARDANO 1560)

In geodesy, the usual rotation sequences for *co-rotated* axes include the aforementioned Equation (1.23) and the sequence  $\phi - \omega - \kappa$  frequently used in photogrammetry (cf. KRAUS 1994; SCHWIDEFSKY and ACKERMANN 1976). Theoretically, all combinations for parameterization are conceivable.

A MAJOR DISADVANTAGE of all parameterizations with three elements (EULER angles) is that *singularities* might occur. This circumstance is caused by the fact that the description is not unique due to the interdependencies of the angles. In the case of unfavourable angle constellations (e. g.  $\phi = 90^\circ$ ,  $\omega = 90^\circ$ ,  $\kappa = 0^\circ$ ) some single rotations can be represented by an infinite number of angles. As a result, a degree of freedom is lost. The phenomenon is known as the so-called *gimbal lock*.

### 1.6.2 PARAMETERIZATION USING QUATERNIONS

Another possibility to describe a rotation in space is the use of algebraic rotation parameters, also known as *Quaternions*<sup>29</sup>. In contrast to parameterization with EULER angles (cf. Section 1.6.1), *no singularities* can occur when using *quaternions*, which is a considerable advantage.

In addition, the expressions in (1.24) are linked bilinear, reducing their sensitivity to approximate values in terms of Least Squares Adjustment (LSA). They also provide good numerical stability paired with good convergence behavior.

SINCE the parameterization is unique, in comparison to the *Cardan* rotation, see Section 1.6.1, no conventions have to be considered regarding the rotation sequence.

The *Quaternion* rotation is closely related to other conventions of rotation, including the one with EULER angles. The connections between different parametrisations can be found, e. g. , in (DIEBEL 2006). A very strong relationship exists between the rotation with *Rodrigue* parameters, to which PUJOL et al. (PUJOL et al. 2012) also refers. A brief derivation is given in Section A.1.1, which leads to the representation

$$\mathbf{R}_{q_0 q_x q_y q_z} = \begin{bmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_x q_y + q_0 q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_0 q_x) \\ 2(q_x q_z - q_0 q_y) & 2(q_y q_z + q_0 q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}. \quad (1.24)$$

<sup>29</sup>The *Quaternions* are also referred to as HAMILTON *Quaternions* or HAMILTON numbers according to Sir *William Rowan Hamilton*, but they were independently discovered three years earlier by *Olinde Rodrigues* (RODRIGUES 1840).

The rotation parameters must satisfy the condition (cf. Equation (A.6))

$$\|\mathbf{q}\| = q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1, \quad (1.25)$$

since the rotation matrix, see Equation (1.24), unlike Equation (1.23), depends on four parameters<sup>30</sup>. To compensate for the over-parameterization, the *auxiliary condition*, see (1.25), has to be considered in the context of an adjustment in terms of a constraints for the unknowns. By taking (1.25) into account, the requirements for an orthogonal rotation matrix, see Equations (1.20) to (1.21), are met.

### 1.7 CLASSICAL 3D-HELMERT TRANSFORMATION

With the help of the *spatial similarity transformation*<sup>31</sup>, three dimensional (3D) cartesian coordinates can be transformed in shape into a specific target system. It serves as a fundamental basis and mathematical framework for most registration procedures, especially in the context of TLS. The relationship between the source and target system can be illustrated using vector addition.

$$\mathbf{X} = \mathbf{X}_0 + \lambda \mathbf{R}\mathbf{x} \quad (1.26)$$

where

$\mathbf{X}_0 = [X_0, Y_0, Z_0]^T$	Translation of the origin of the initial system $\mathbf{O}_i$ with respect to the target system $\mathbf{O}_t$
$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$	Rotation matrix specifying the rotation ( $\omega, \phi, \kappa$ ) of the coordinate axes of the initial system in relation to the target system, see Equation (1.23)
$\lambda$	Scale factor of the initial system in relation to the target system

A point P, see Figure 1.22, represented by its vector  $\mathbf{X}$  in the target system, can thus be described as the sum of the translation vector  $\mathbf{X}_0$  and the vector  $\mathbf{x}$  in the initial system. To ensure the validity of the vector addition, the vector  $\mathbf{x}$  is transferred via  $\mathbf{R}$  into a system parallel to the initial system and scaled with  $\lambda$ , see (1.26).

<sup>30</sup>Often variants  $(q_0, q_1, q_2, q_3)$ ,  $(w, x, y, z)$  and  $(d, a, b, c)$  are used instead of the parameter set  $(q_0, q_x, q_y, q_z)$  for the designation of *Quaternions* in the literature.

<sup>31</sup>The *spatial similarity transformation* is also often referred to as *3D-Helmert transformation*. Due to the seven parameters to be determined, the term *7-parameter transformation* is also common

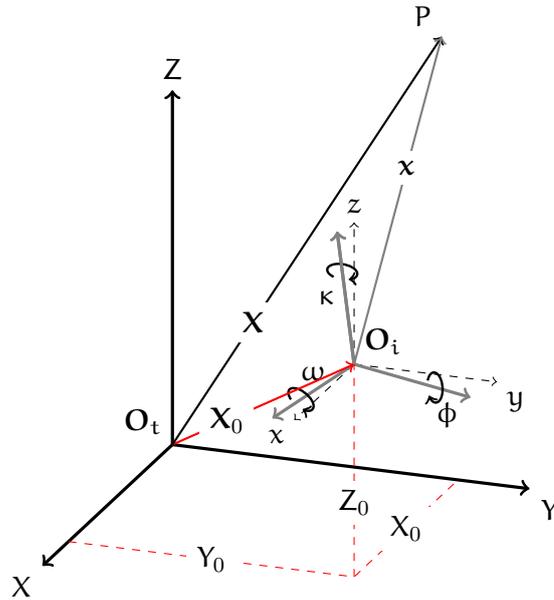


Figure I.22: Principle of spatial similarity transformation.

Spatial coordinates  $\mathbf{x}_i = [x_i, y_i, z_i]^T$  of the points  $p_i (i = 1, \dots, k)$  in the initial system are transformed into the spatial coordinates  $\mathbf{X}_i = [X_i, Y_i, Z_i]^T$  of the homologous points  $P_i (i = 1, \dots, k)$  in the target system via the transformation rule of the *3D-Helmert Transformation*  $\mathbf{X}_i = \mathbf{X}_0 + \lambda \mathbf{R} \mathbf{x}_i$ , see also [Equation \(1.26\)](#), where  $k$  is the number of homologous points. In total seven parameters have to be determined, the three translation parameters  $X_0, Y_0, Z_0$ , the three rotation parameters  $\phi, \omega, \kappa$  and a common scale factor  $\lambda$ . Thus, the *spatial similarity transformation* represents a special case of the *spatial affine transformation*, in which a separate scale and a shear parameter are applied for each coordinate axis.

NOTE: If algebraic rotation parameters, see [Section 1.6.2](#), are used instead of trigonometric parameters, see [Section 1.6.1](#), for the parameterization of the rotation matrix, this results in an additional parameter to be determined for the rotation parameters. This procedure offers some advantages both for the calculation and the determination of approximate values.

The six parameters for translation and rotation, or seven parameters when using *Quaternions*, comply with the parameters of the so-called *exterior orientation*<sup>32</sup>, see also [Section 1.3.1](#).

<sup>32</sup>The *exterior orientation* is also referred to in the literature as *outer orientation*, mostly in the context of photogrammetric issues.

REMARKS: For the unambiguous solution of the seven or eight parameters when using quaternions, at least two *full control points* and one or two further coordinate components are required. At least three *homologous points* (*control points*) must be provided for an adjustment. These may not lie on a straight line in space.

## 1.8 CONCEPTS OF ADJUSTMENT THEORY

The methods of adjustment calculus constitute a decisive basis for many calculation methods used in geodesy. In this thesis, adjustment problems are treated in the context of the evaluation of the *stochastic model*, see [Section 2.3](#), the segmentation of planes, see [Section 4.3.2](#), and the registration procedure in [Section 4.5](#).

### 1.8.1 ADJUSTMENT PROBLEM

If more observations than unknowns are present ( $n > u$ ) in a problem statement, where

$n$  Number of observations

$u$  Number of unknowns,

then this problem is overdetermined and is termed an adjustment problem. However, even in the case of a definite problem ( $n = u$ ), the system of observations and to be estimated parameters is determined. Accuracy measures, see [Section 1.8.9.1](#), can also be derived via the *law of variance propagation*, but no control or reliability, cf. [Section 1.8.9.2](#), of such a minimum configuration is ensured.

The task of the adjustment is to adapt the unknown parameters linked to the measured values via a *functional model*, cf. [Section 1.8.2](#), in a *mathematical model*<sup>33</sup>, see [Sections 1.8.5](#) to [1.8.7](#), reasonably to the empirical measured data defined by a *stochastic model*, cf. [Section 1.8.3](#). This is done by introducing so-called corrections<sup>34</sup> for the observations and specifying an additional requirement for a function of these corrections, i. e. residuals. The most common requirement is that the squared sum of the weighted residuals should become minimal, see [Equation \(1.41\)](#).

### 1.8.2 THE FUNCTIONAL MODEL

Geodesy is concerned with describing reality based on physics or geometry in the form of measured quantities. In most cases, however, not all the parameters that constitute the respective phenomena can be measured. For this reason, a functional relationship (*functional model*) must be established between

<sup>33</sup> The *mathematical model* consists of the *functional model* and *stochastic model*.

<sup>34</sup> In geodesy the term residuals has become established for this kind of corrections, used in this thesis.

the actual observed measured values<sup>35</sup> and the unknown quantities. With regard to the least squares method, see Section 1.8.4, the model often appears in the form

$$\tilde{\mathbf{l}} = f(\tilde{\mathbf{x}}). \quad (1.27)$$

In this form the quantity of observations, represented by vector  $\tilde{\mathbf{l}}$ , is mapped via their functional relationship to the quantity of true to be estimated parameters  $\tilde{\mathbf{x}}$ .

However, the representation is not contradiction-free, since there are no error-free measurements from a technical and human point of view. In reality, the vector  $\tilde{\mathbf{l}}$  is falsified by various deviations<sup>36</sup> resulting in the following realistic model

$$\mathbf{l} - \epsilon_r - \epsilon_s - \epsilon_g = f(\hat{\mathbf{x}}), \quad (1.28)$$

where

- $\mathbf{l}$  Observation vector that contains according to (1.28) all measurements with the above mentioned error influences, that must be reduced by these,
- $\hat{\mathbf{x}}$  Parameter vector with stochastic estimates,
- $\epsilon_r$  Random errors caused by non controllable influences of the measuring instruments and the erroneous human observer,
- $\epsilon_s$  Systematic errors resulting from inadequate or unfavourable measuring arrangements or measuring processes or from an insufficient modeling of the functional model or as a result from a decalibrated instrument,
- $\epsilon_g$  Gross errors that can occur, e. g. due to incorrect allocations of the measured quantities  $\mathbf{l}$  with the functional relationship  $f(\hat{\mathbf{x}})$ .

NOTE: In practice, however, it is usually not possible to describe the systematic errors in a functional way, since one either does not know the correlation in which the errors occur or they cannot be separated from each other.

In the case of known systematics, the error can be taken into consideration with a suitable measurement configuration. Gross errors cannot be detected directly either, but can be eliminated or taken into account using special methods such as BAARDAS *data snooping* (BAARDA 1968).

For the reasons mentioned above, it is therefore assumed that the measured values  $\mathbf{l}$  are only affected by random errors  $\epsilon_r$ . By renaming the random errors to the residuals

<sup>35</sup> Observed measured values are referred to simply as observations in the following.

<sup>36</sup> All these deviations, whether random or systematic, are usually referred to as errors.

$$\mathbf{v} = -\mathbf{e}_r,$$

conventionally used in geodesy, the so-called error equations

$$\mathbf{l} + \mathbf{v} = f(\hat{\mathbf{x}}) \tag{1.29}$$

can be introduced.

### 1.8.2.1 LINEARISATION

If the functional relationship  $f(\hat{\mathbf{x}})$  is non-linear, a so-called *linearisation* is required. The *Taylor series expansion* approach is used to determine function values  $f$  of a non-linear function at a position  $x^0 + \Delta x$  if the start value of function  $f(x^0)$  is known and the correction  $\Delta x$  is small, see [Figure 1.23](#).

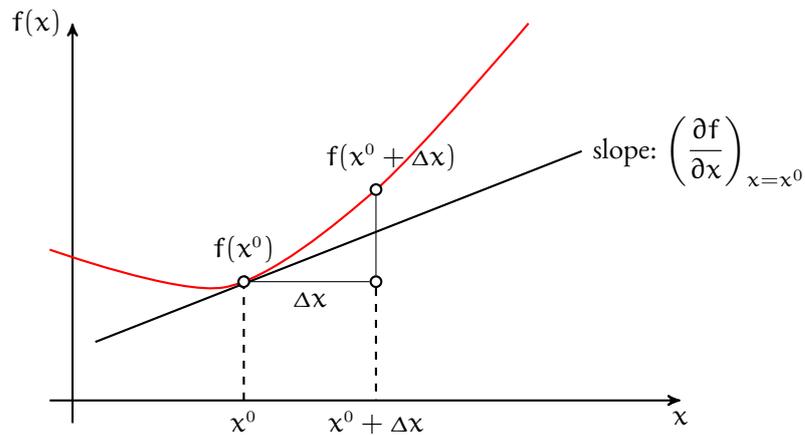


Figure 1.23: Linearisation of a function according to NIEMEIER (2008) p. 122.

The function value for  $f(x^0 + \Delta x)$  can be calculated with the *Taylor series*, knowing that the partial derivative of the function  $f$  in  $x^0$  can be interpreted as the gradient of the tangent at the respective point with

$$f(x^0 + \Delta x) = f(x^0) + \left(\frac{\partial f}{\partial x}\right)_{x=x^0} \Delta x + \frac{1}{2} \left(\frac{\partial f}{\partial x}\right)_{x=x^0}^2 \Delta x^2 + \dots \tag{1.30}$$

Due to the requirement for linearity, the linearisation in (1.30) is limited to the first term for adjustment purposes. [Figure 1.23](#) shows at a glance that this is only an approximate solution. As a consequence, good approximate values  $x^0$  must be provided for higher order functions, with a severely non-linear characteristic, so that the corrections  $\Delta x$  remain small. In addition, an iterative procedure is always required.

Transferring the procedure from Equation (1.30) to the functional model of the  $n$  observation equations  $i = 1, \dots, n$ , see (1.29), with the  $u$  unknown parameters  $j = 1, \dots, u$ , we obtain

$$l_i + v_i = f_i(x_1^0, x_2^0, \dots, x_u^0) + \left( \frac{\partial f_i}{\partial x_1} \right)_{x^0} \hat{x}_1 + \left( \frac{\partial f_i}{\partial x_2} \right)_{x^0} \hat{x}_2 + \dots + \left( \frac{\partial f_i}{\partial x_u} \right)_{x^0} \hat{x}_u. \quad (1.31)$$

If the partial derivatives in (1.31) with their abbreviations

$$a_{ij} = \left( \frac{\partial f_i}{\partial x_j} \right)$$

are summarised according to the order of observations  $l_i$  and unknowns  $\hat{x}_j$  in a  $n \times u$  matrix<sup>37</sup>

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1u} \\ a_{21} & a_{22} & \cdots & a_{2u} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nu} \end{bmatrix}, \quad (1.32)$$

the linearised error equations can be specified in matrix form

$$\mathbf{l} + \mathbf{v} = \mathbf{x}^0 + \mathbf{A}\hat{\mathbf{x}} = f(\mathbf{x}^0) + \mathbf{A}\hat{\mathbf{x}}. \quad (1.33)$$

Introducing the so-called truncated observation vector

$$\Delta \mathbf{l} = \mathbf{l} - \mathbf{l}^0 = \mathbf{l} - f(\mathbf{x}^0) \quad (1.34)$$

yields the known relation

$$\Delta \mathbf{l} + \mathbf{v} = \mathbf{A}\hat{\mathbf{x}}, \quad (1.35)$$

to be solved by a Gauss-Markov model, see Section 1.8.5.

**CONCLUDING REMARK:** An insufficient functional modelling inevitably leads to a systematic falsification of the desired parameters. Good approximate values are mandatory for non-linear problems<sup>38</sup>.

<sup>37</sup>The matrix  $\mathbf{A}$  is often called A-matrix, but also design matrix.

<sup>38</sup>In the linear case, the searched unknowns  $\hat{\mathbf{x}}$  are obtained directly, in the nonlinear case via suitable approximate values only corrections for the unknowns are obtained, hence in the following  $\Delta \hat{\mathbf{x}}$  is used instead of  $\hat{\mathbf{x}}$  within the nonlinear case.

1.8.3 THE STOCHASTIC MODEL

For the observed quantities  $l_i$ , it applies that these were observed with a certain precision. This a priori information can be taken on the basis of empirical values or instrument-related accuracy specifications are available, which are usually preferable.

These specifications are of crucial importance for an adjustment, since the searched parameters  $\hat{\mathbf{x}}$  and their precisions are directly influenced by the precisions of the individual observations  $l_i$  (NIEMEIER 2008). Therefore, a reliable accuracy statement of the respective measured values is even more important. For use in linearized adjustment models, see Sections 1.8.5 to 1.8.6, the precision information is presented in a matrix

$$\Sigma_{ll} = \begin{bmatrix} \sigma_1^2 & & & 0 \\ & \sigma_i^2 & & \\ & & \ddots & \\ 0 & & & \sigma_n^2 \end{bmatrix} \quad (1.36)$$

as standard deviations  $\sigma_i$  of the respective measured values  $l_i$ . This matrix is referred to as the Variance-Covariance Matrix (VCM) of the  $n$  observations or also called *stochastic model* of adjustment calculation. On the main diagonal the variances are listed in the order of the observations. The secondary diagonal elements represent the covariances and are set to zero for the most commonly assumed case of uncorrelated observations.

If the observations can be considered to be equally precise and uncorrelated, the Variance-Covariance Matrix (VCM) results with

$$\sigma_1^2 = \sigma_2^2 = \sigma_i^2 \dots = \sigma_n^2 = \sigma^2$$

to a matrix

$$\Sigma_{ll} = \begin{bmatrix} \sigma^2 & & & 0 \\ & \sigma^2 & & \\ & & \ddots & \\ 0 & & & \sigma^2 \end{bmatrix} = \sigma^2 \mathbf{I}, \quad (1.37)$$

which is obtained by multiplying the uniform *variance factor*<sup>39</sup>  $\sigma^2$  by the unit matrix  $\mathbf{I}$ . For the process

<sup>39</sup>Also denoted as variance of the unit of weight or sigma a posteriori.

of adjustment, the above representation (1.37) is generalised to a weighted approach

$$\mathbf{Q}_u = \frac{1}{\sigma_0^2} \boldsymbol{\Sigma}_u = \frac{1}{\sigma_0^2} \begin{bmatrix} \sigma_1^2 & & & 0 \\ & \sigma_i^2 & & \\ & & \ddots & \\ 0 & & & \sigma_n^2 \end{bmatrix} \quad (1.38)$$

according to NIEMEIER (2008) p. 124. This matrix is formed over the product of the reciprocal global precision factor  $\sigma_0^2$  and the **VCM** from Equation (1.36) and is known as the Cofactor Matrix (**CM**) of the observations. It only contains precision ratios with regard to the globally applied accuracy. Based on the **CM** of the observations expressed by (1.38), the weight matrix

$$\mathbf{P} = \mathbf{Q}_u^{-1} \quad (1.39)$$

results, which is incorporated in the minimization task of Equation (1.41).

#### 1.8.4 LEAST SQUARES METHOD (**LSM**)

The most commonly used estimation method to solve adjustment problems is the Least Squares Method (**LSM**)<sup>40</sup>. It was discovered by *C.F. Gauss* and *A.M. Legendre* more or less at the same time at the beginning of the 19th century. When adjusting according to the **LSM**, the target function of the residuals  $\mathbf{v}$  is that the sum of all squared residuals

$$\sum_{i=1}^n v_i^2 = \mathbf{v}^T \mathbf{v} \rightarrow \min \quad (1.40)$$

should become minimal. If a *stochastic model*, see Section 1.8.3, is also considered, so that the observations cannot be regarded as equally weighted and uncorrelated, the requirement of the least square sum of the weighted residuals

$$\sum_{i=1}^n p_i v_i v_i = \mathbf{v}^T \mathbf{P} \mathbf{v} \rightarrow \min \quad (1.41)$$

must be met. This method has the prominent characteristic that the standard deviation of each adjusted quantity becomes minimal and thus pursues the goal of minimal variance.

<sup>40</sup>Also known as *error square method* or *L2-Norm*.

A fundamental prerequisite for the applicability of **LSM** is that the observation errors are approximately normal distributed random quantities. Regarding the necessity of normally distributed measurements R.F. HELMERT (HELMERT 1924) writes:

Lässt sich nachweisen, dass das Fehlervorkommen der Gaußschen Form (remark: *normal distribution*) [...] entspricht, so erhalten wir [...] durch die Methode der kleinsten Quadrate die wahrscheinlichsten Werte der Unbekannten. Zugleich besitzen diese Werte die größten Gewichte bzw. die kleinsten mittleren Fehler. Entspricht aber das Fehlervorkommen dem Gaußschen Fehlergesetze nicht, so haben wir nicht mehr die wahrscheinlichsten Werte der Unbekannten, dagegen in ihrer Bestimmung immer noch die kleinsten mittleren Fehler [...].

Furthermore, the observations should ideally be free of gross errors, as these have a significant influence on the normal distribution of the measured values. In addition, the functional model, see Section 1.8.2 must be described sufficiently so that no systematics can affect the random errors. If the above criteria cannot be fulfilled, the **LSM** provides unrealistic model parameters.

In order to be able to perform an adjustment with a high percentage of gross errors, alternative estimation methods, such as the *L1-Norm*, or procedures for identifying and eliminating these are used, see e. g. (BAARDA 1968) in case of an maximum error budget of 3-5% (NIEMEIER 2008, p. 218). In geodesy there are three common adjustment models, see Section 1.8.4, which are based on the **LSM**.

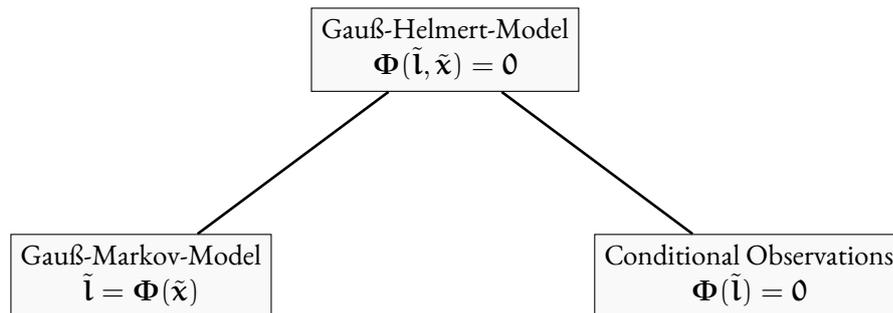


Figure 1.24: Adjustment models according to the **LSM**.

The Gauß Helmert Model (**GHM**), see Section 1.8.6, is a superordinate model, see Section 1.8.4, since all other models can be derived from it – it is therefore also referred to as the *general case of adjustment calculation*<sup>41</sup>. It may contain several observation quantities in a functional context, cf. Equation (1.50), which are formulated as so-called condition equations.

The most common and most popular adjustment model is the Gauß Markov Model (**GMM**). Only one measured quantity may occur in the functional context of the **GMM**, see Section 1.8.5, but each

<sup>41</sup>R.F. HELMERT (HELMERT 1924) also defines the *general case of adjustment calculation* as *conditional observations with unknowns*

problem definition in a **GHM** can be transferred into an equivalent adjustment for a **GMM** by introducing so-called *pseudo observations* for the additional measured quantities.

The rather less common model is the adjustment according to conditional observations. In the case of conditional observations, the functional relationship consists only of observations, i.e. without unknowns.

FINALLY, all models pursue the minimisation of the same *objective function*, see Equation (1.40), and thus represent all adjustment procedures in the sense of the Least Squares Method (**LSM**).

### 1.8.5 THE GAUSS MARKOV MODEL (**GMM**)

The adjustment based on the Gauß Markov Model (**GMM**) results in the linear or linearized functional model rearranged according to the residuals

$$\mathbf{v} = \mathbf{A}\hat{\mathbf{x}} - \mathbf{l} \quad (1.42)$$

and introduced in Section 1.8.2, see also Equation (1.35). According to the minimisation principle of *least squares* and the given weight matrix **P** it is claimed

$$\mathbf{\Omega} = \mathbf{v}^T \mathbf{P} \mathbf{v} \rightarrow \min . \quad (1.43)$$

Inserting the function of the residuals (1.42) into (1.43) and performing a few matrix operations now leads to the final *objective function*

$$\begin{aligned} \mathbf{v}^T \mathbf{P} \mathbf{v} &= (\hat{\mathbf{x}}^T \mathbf{A}^T - \mathbf{l}^T) \mathbf{P} (\mathbf{A}\hat{\mathbf{x}} - \mathbf{l}) \\ &\Rightarrow \hat{\mathbf{x}}^T \mathbf{A}^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}} - \hat{\mathbf{x}}^T \mathbf{A}^T \mathbf{P} \mathbf{l} - \mathbf{l}^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}} + \mathbf{l}^T \mathbf{P} \mathbf{l} \\ &\Rightarrow \hat{\mathbf{x}}^T \mathbf{A}^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}} - 2\hat{\mathbf{x}}^T \mathbf{A}^T \mathbf{P} \mathbf{l} + \mathbf{l}^T \mathbf{P} \mathbf{l} \end{aligned} \quad (1.44)$$

for which the minimum has to be determined. To determine the minimum of a function, the first derivative of this function must be set to zero – since the gradient in a point, cf. Figure 1.23, is mathematically equal to zero at a minimum – and must be resolved according to the searched term. The first derivative of (1.44) following the variable  $\hat{\mathbf{x}}$  then yields

$$\frac{\partial(\mathbf{v}^T \mathbf{P} \mathbf{v})}{\partial \hat{\mathbf{x}}} = 2(\mathbf{A}^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}} - \mathbf{A}^T \mathbf{P} \mathbf{l}). \quad (1.45)$$

As it can easily be seen, the first derivative of (1.45) becomes zero if the expression in brackets becomes zero, which results in the equation system

$$\mathbf{A}^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}} - \mathbf{A}^T \mathbf{P} \mathbf{l} = 0 \quad \Rightarrow \quad \mathbf{A}^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^T \mathbf{P} \mathbf{l} \quad (1.46)$$

to be solved for the **LSM**. If a left multiplication with  $(\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1}$  is then performed, the solution vector

$$\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{P} \mathbf{l} \quad (1.47)$$

of the system is obtained, whereby it is assumed that the inverse of  $\mathbf{A}^T \mathbf{P} \mathbf{A}$  exists respectively that the matrix has full column rank.

The expression  $\mathbf{A}^T \mathbf{P} \mathbf{A}$  in (1.47) is usually referred to as the *normal equation matrix* and is therefore briefly called  $\mathbf{N}$ , while the other part  $\mathbf{A}^T \mathbf{P} \mathbf{l}$  of the system in (1.46) is titled as the so-called right side and is often named  $\mathbf{n}$ . Thus Equation (1.46) can also be written as

$$\mathbf{N} \hat{\mathbf{x}} = \mathbf{n}. \quad (1.48)$$

The adjusted final parameters are obtained in the non-linear case

$$\hat{\mathbf{x}} = \mathbf{x}^0 + \Delta \hat{\mathbf{x}} \quad (1.49)$$

via the approximate values of the unknowns  $\mathbf{x}^0$  and  $\Delta \hat{\mathbf{x}} = \hat{\mathbf{x}}$  from (1.47), the vector of adjusted reduced unknowns, or simplified corrections, for the unknowns.

IT CAN BE STATED THAT:

1. With a non-linear model a linearization must be carried out at the location of suitable approximate values  $\mathbf{x}^0$  (cf. Section 1.8.2.1) and an iterative calculation must be carried out.
2. The higher the degree of nonlinearity, the more accurate the approximate values must be in order to ensure a solution converging to the global minimum of the *objective function*.

#### 1.8.6 THE GAUSS HELMERT MODEL (GHM)

In contrast to the **GMM**, see Section 1.8.5, the functional relationship of the **GHM** is not formulated by observations as a function of the unknowns and possible fixed quantities, but by so-called *conditional equations*. In Equation (1.42) it is assumed that each observation can be represented as a function of the searched parameters.

In the following model, however, several observations and unknowns can also occur in a functional context. The observation vector  $\mathbf{l}$ , the residuals  $\mathbf{v}$  and the unknowns  $\mathbf{x}$  are thus linked by  $r$  nonlinear, differentiable *conditional equations* of the form

$$\Psi_i(\mathbf{v}, \mathbf{x}) = h_i(\mathbf{l} + \mathbf{v}, \mathbf{x}) = 0 \quad , \text{ where } i = 1, \dots, r. \quad (1.50)$$

For the complete and correct linearisation, see [Section 1.8.2.1](#), of [\(1.50\)](#), the approximate values for the residuals  $\mathbf{v}^0$  and the unknowns  $\mathbf{x}^0$  are introduced. Accordingly, a complete linearisation must be carried out at the locations  $\mathbf{x}^0$  of the unknowns and the residuals  $\mathbf{v}^0$ .

**IMPORTANT NOTE:** The linearisation in many standard works of adjustment calculation, as for example in [\(NIEMEIER 2008\)](#) or [\(JÄGER et al. 2005\)](#), is however erroneously performed only at the location  $\mathbf{x}^0$  of the unknowns, which is equivalent to an additional linearisation for the residuals at the location  $\mathbf{v}^0 = 0$ . The results of the analytical strategy used in this literature therefore only correspond to an approximate solution.

Netherless, the necessity of this proper linearisation was pointed out early in [\(POPE 1972\)](#). In [\(LENZMANN and LENZMANN 2004\)](#) the issue is taken up again and a correct solution scheme is presented. The strict evaluation of the [GHM](#) described in the following is based on the explanations in [\(LENZMANN and LENZMANN 2004\)](#), resulting in the entirely linearised *condition equations*

$$\mathbf{f}(\mathbf{v}, \hat{\mathbf{x}}) = \Psi_i(\mathbf{v}^0, \mathbf{x}^0) + \mathbf{B}(\mathbf{v} - \mathbf{v}^0) + \mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}^0) = 0. \quad (1.51)$$

The linearisation of [\(1.50\)](#) is based on the above considerations with the *Jacobians* of the partial derivatives

$$\mathbf{B}(\mathbf{v}, \hat{\mathbf{x}}) = \left. \frac{\partial \Psi_i(\mathbf{v}, \mathbf{x})}{\partial \mathbf{v}} \right|_{\mathbf{v}^0, \mathbf{x}^0} \quad \text{and} \quad \mathbf{A}(\mathbf{v}, \hat{\mathbf{x}}) = \left. \frac{\partial \Psi_i(\mathbf{v}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{v}^0, \mathbf{x}^0} \quad (1.52)$$

at locations  $\mathbf{x}^0$  and  $\mathbf{v}^0$  respectively. The fact that an adjustment problem with a design matrix  $\mathbf{A}$ , see [\(1.52\)](#), in which residuals  $\mathbf{v}$  can occur, needs not to be solved by frequently propagated *total least squares* techniques is explained in [\(NEITZEL and PETROVIC 2008\)](#).

According to БӨСК ([БӨСК 1961](#)) a strict linearisation, as mentioned above, is indispensable, if the residuals  $\mathbf{v}$  are not differentially small in relation to the observations  $\mathbf{l}$ . Observations  $l_i$  with a low weight  $p_i = \frac{1}{\sigma_i}$ , respectively a large standard deviation  $\sigma_i$ , are strongly corrected in the sense of the adjustment, thus a correct linearisation is mandatory in such cases.

Anyway, there are also special problems where no residuals in the design matrix arise and these can be solved with an incomplete linearisation.

From [\(1.51\)](#) the current vector of misclosures

$$\mathbf{w} = -\mathbf{B}\mathbf{v}^0 + \Psi_i(\mathbf{v}^0, \mathbf{x}^0) \quad (1.53)$$

with the approximate values  $\mathbf{v}^0$  and  $\mathbf{x}^0$  results from the previous iteration step.

The addition  $-\mathbf{B}\mathbf{v}^0 + \Psi_i(\mathbf{v}^0, \mathbf{x}^0)$  guarantees the requirement of the Weighted Least Squares ([WLS](#)) sum

for the residuals, cf. Equation (1.41), and can be derived algebraically (GRÜNDIG 2003). Considering Equations (1.51) to (1.53), the linearised *condition equation*

$$\mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}^0) + \mathbf{B}\mathbf{v} + \mathbf{w} = \mathbf{0} \quad (1.54)$$

to be minimised with respect to the GHM can hence be specified. Under the secondary condition (1.54) and the main condition with the requirement of Least Squares Estimation (LSE), see Equation (1.41), it is now a challenge to find a minimum. This can be done by using the so-called *Lagrange function* (LAGRANGE 1788), which allows to solve minimization problems with several secondary conditions. The complete target function

$$\Omega = \mathbf{v}^T \mathbf{P} \mathbf{v} - 2\mathbf{k}^T (\mathbf{B}\mathbf{v} + \mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}^0) + \mathbf{w}) \rightarrow \min \quad (1.55)$$

to be minimized is thus obtained. In (1.55)  $\hat{\mathbf{x}}$  and  $\mathbf{v}$  are to be determined in such a way that  $\mathbf{v}^T \mathbf{P} \mathbf{v}$  becomes minimal under fulfilment of the secondary condition (1.54), where  $\mathbf{k}$  represents the so-called *Lagrange multiplier*<sup>42</sup>.

The minimum of (1.55) can then be obtained using the partial derivatives

$$\frac{\partial \Omega}{\partial \mathbf{v}} = 2\mathbf{v}^T \mathbf{P} - 2\mathbf{k}^T \mathbf{B} \quad (1.56a)$$

$$\frac{\partial \Omega}{\partial (\hat{\mathbf{x}} - \mathbf{x}^0)} = -2\mathbf{k}^T \mathbf{A} \quad (1.56b)$$

$$\frac{\partial \Omega}{\partial \mathbf{k}^T} = -2(\mathbf{B}\mathbf{v} + \mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}^0) + \mathbf{w}) \quad (1.56c)$$

according to the present unknowns  $\mathbf{v}$ ,  $\hat{\mathbf{x}} - \mathbf{x}^0$  and  $\mathbf{k}$ . Regarding the zero determination, the derivatives (1.56a) to (1.56c) are set to zero in the context of the minimization task, resulting in the three equations

$$-\mathbf{P}\mathbf{v} + \mathbf{B}^T \mathbf{k} = \mathbf{0}, \quad (1.57a)$$

$$\mathbf{A}^T \mathbf{k} = \mathbf{0}, \quad (1.57b)$$

$$\mathbf{B}\mathbf{v} + \mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}^0) + \mathbf{w} = \mathbf{0}. \quad (1.57c)$$

<sup>42</sup>In the parlance of geodesy and adjustment calculation these multipliers are called *correlates*. They serve as an indicator for the influence of the respective condition on the adjustment result.

Rearranging (1.57a) to  $\mathbf{P}\mathbf{v}$  and multiplying it by  $\mathbf{P}^{-1}$ , the residuals

$$\mathbf{v} = \mathbf{Q}_{ll}\mathbf{B}^T\mathbf{k} \quad (1.58)$$

can be determined taking into account that  $\mathbf{P}^{-1} = \mathbf{Q}_{ll}$ , where  $\mathbf{Q}_{ll}$  is the CM of the observations  $l$ . Inserting (1.58) into (1.57c) results in the common expression

$$\mathbf{B}\mathbf{Q}_{ll}\mathbf{B}^T\mathbf{k} + \mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}^0) + \mathbf{w} = \mathbf{0}. \quad (1.59)$$

With the second condition, respectively (1.57b), and the *conditional equation*, expressed by (1.59), the entire equation system to be solved

$$\begin{bmatrix} \mathbf{B}\mathbf{Q}_{ll}\mathbf{B}^T & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{k}} \\ (\hat{\mathbf{x}} - \mathbf{x}^0) \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{0} \end{bmatrix} = \mathbf{0} \quad (1.60)$$

can be formulated in block matrix notation. For  $\hat{\mathbf{x}} - \mathbf{x}^0$ , the abbreviation  $\Delta\hat{\mathbf{x}}$  can be written. From the current approximate values  $\mathbf{x}^0$  of the unknowns and their adjusted corrections  $\Delta\hat{\mathbf{x}}$ , the final unknowns result in analogy to the GMM, cf. Section 1.8.5 and equation (1.49)

$$\hat{\mathbf{x}} = \mathbf{x}^0 + \Delta\hat{\mathbf{x}} \quad \text{and} \quad \hat{\mathbf{v}} = \mathbf{Q}_{ll}\mathbf{B}^T\hat{\mathbf{k}}, \quad (1.61)$$

whereas the adjusted residuals arise from (1.58). After each iteration step, the estimated residuals and unknowns are introduced into the adjustment as new approximate values  $\mathbf{x}^0 = \hat{\mathbf{x}}$  and  $\mathbf{v}^0 = \hat{\mathbf{v}}$  until a certain break condition is reached. A possible termination criterion

$$\max \Psi_i(\hat{\mathbf{v}}, \hat{\mathbf{x}}) = \max [0 \quad \dots \quad 0]^T < t_c \quad (1.62)$$

is thus reached if the maximum element of the vector (1.50) resulting from the original nonlinear condition equations is smaller than a bound  $t_c$ , e. g.  $t_c = 10^{-7}$ .

**NOTE:** An iterative approach is absolutely necessary for non-linear problems of any kind, since the approximate values are often not so accurate at the beginning and the corrections  $\Delta\hat{\mathbf{x}}$  of the unknowns are therefore slightly larger. A possible break condition could thus also be formulated for the unknown corrections, as it is frequently applied in practice.

**FINALLY, IT SHOULD BE NOTED THAT** regardless of the chosen *functional model*, the general case of adjustment calculation provides an identical solution for the parameters to be estimated as the GMM, see Section 1.8.5. However, the prerequisite for this is that a correct linearisation is carried out, as can be found in (LENZMANN and LENZMANN 2004), among others. When transferring the GHM into

an equivalent **GMM** in the context of so called *quasi observations*, there are also no limitations as long as the residuals in the linearisation are taken into account.

#### 1.8.7 THE GAUSS HELMERT MODEL (**GHM**) WITH CONSTRAINTS

In addition to the usual condition equations, see [Equation \(1.50\)](#), the unknown parameters may also be subject to other mathematical-geometric conditions. This is the case, for example, when quaternions are used to parameterize the *rotation matrix*, see [Section 1.6.2](#).

In contrast to the conditional equations in [Equation \(1.50\)](#), the so-called constraint equations do not contain observation content, but merely represent relations between unknowns and constants. The unknowns  $\mathbf{x}$  and the constants  $\mathbf{c}$  are thus linked via  $r$  nonlinear and differentiable conditions of the general form

$$\gamma_i(\mathbf{x}) = h_i(\mathbf{x}, \mathbf{c}) = \mathbf{0}^{43} \quad , \text{ where } i = 1, \dots, r. \quad (1.63)$$

Since no observations with stochastic information in the form of weights occurs in [\(1.63\)](#), these do not represent weighted condition equations in contrast to [\(1.50\)](#) and thus have no controllable influence on the adjustment result. The influence on the adjustment is infinite, as desired in a fixed mathematical-geometrical context, hence this type of condition is also termed deterministic condition<sup>44</sup>.

Within the context of the adjustment, these additional conditions must be included, leading to a conditional adjustment with unknowns and conditions between them (**GHM** with constraints). For this [\(1.63\)](#) are linearized analogously to [Equation \(1.51\)](#) under the introduction of the approximate values  $\mathbf{x}^0$  as follows

$$\begin{aligned} \mathbf{g}(\mathbf{x}) &= \mathbf{C}(\hat{\mathbf{x}} - \mathbf{x}^0) + \gamma(\mathbf{x}^0) = \mathbf{c} \\ &= \mathbf{C}\Delta\hat{\mathbf{x}} + \gamma(\mathbf{x}^0) = \mathbf{c} \end{aligned} \quad , \quad (1.64)$$

where

$$\mathbf{C}(\mathbf{x}^0) = \frac{\partial \gamma(\mathbf{x}^0)}{\partial \mathbf{x}^0} \quad (1.65)$$

is the matrix of the partial derivatives and

$$\mathbf{w}_r = \gamma(\mathbf{x}^0) - \mathbf{c} \quad (1.66)$$

<sup>43</sup>To prevent a rank deficiency, the deterministic constraint equations must be mutually linearly independent.

<sup>44</sup>Also referred to as a mandatory condition.

is the condition vector. By using  $\mathbf{c} = \mathbf{0}$  the linearized constraint equation reads

$$\mathbf{C}\Delta\hat{\mathbf{x}} + \mathbf{w}_r = \mathbf{0}. \quad (1.67)$$

With (1.67) and Equation (1.51), the task to be solved represents a minimization problem with constraints, which can be set up using the *Lagrange function*, cf. Section 1.8.6 and especially Equation (1.55). The objective function to be minimized thus results from Equations (1.41), (1.54) and (1.67) to

$$\Omega = \mathbf{v}^T \mathbf{Q}_{11}^{-1} \mathbf{v} - 2\mathbf{k}^T (\mathbf{B}\mathbf{v} + \mathbf{A}\Delta\hat{\mathbf{x}} + \mathbf{w}) - 2\mathbf{k}_r^T (\mathbf{C}\Delta\hat{\mathbf{x}} + \mathbf{w}_r) \rightarrow \min. \quad (1.68)$$

In order to find the minimum of the function according to (1.68), the derivatives are formed according to the unknowns, resulting in the conditional equations

$$\frac{\partial \Omega}{\partial \mathbf{v}} = \mathbf{Q}_{11}^{-1} \mathbf{v} - \mathbf{B}^T \mathbf{k} = \mathbf{0}, \quad (1.69a)$$

$$\frac{\partial \Omega}{\partial \Delta\hat{\mathbf{x}}} = -\mathbf{A}^T \mathbf{k} - \mathbf{C}^T \mathbf{k}_r = \mathbf{0}, \quad (1.69b)$$

$$\frac{\partial \Omega}{\partial \mathbf{k}} = \mathbf{B}\mathbf{v} + \mathbf{A}\Delta\hat{\mathbf{x}} + \mathbf{w} = \mathbf{0}, \quad (1.69c)$$

$$\frac{\partial \Omega}{\partial \mathbf{k}_r} = \mathbf{C}\Delta\hat{\mathbf{x}} + \mathbf{w}_r = \mathbf{0}. \quad (1.69d)$$

Assuming that the adjustment task consists of  $n$  conditional equations and  $r$  constraint equations and that  $u$  unknowns are to be determined, then the system of equations

$$\begin{bmatrix} \mathbf{Q}_{11}^{-1} & -\mathbf{B}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}^T & \mathbf{0} & -\mathbf{C}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{v}} \\ \hat{\mathbf{k}} \\ \Delta\hat{\mathbf{x}} \\ \hat{\mathbf{k}}_r \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{w} \\ \mathbf{0} \\ -\mathbf{w}_r \end{bmatrix} \quad (1.70)$$

to be solved can be constructed with the equations Equations (1.69a) to (1.69d) in block matrix form. The system of (1.70) is very complex and difficult to solve using the standard algorithm of the *gaussian elimination method*. For this reason it is reduced to solve  $\Delta\hat{\mathbf{x}}$  and  $\hat{\mathbf{k}}_r$ . However, rearranging (1.69a) to  $\hat{\mathbf{v}}$  and thus replacing  $\hat{\mathbf{v}}$  in (1.69c) results in

$$\hat{\mathbf{k}} = (\mathbf{B}\mathbf{Q}_{11}\mathbf{B}^T)^{-1} (-\mathbf{A}\Delta\hat{\mathbf{x}} - \mathbf{w}). \quad (1.71)$$

By inserting (1.71) in (1.69b) and excluding  $\Delta\hat{\mathbf{x}}$  follows

$$(\mathbf{A}^\top(\mathbf{B}\mathbf{Q}_{11}\mathbf{B}^\top)^{-1}\mathbf{A})\Delta\hat{\mathbf{x}} - \mathbf{C}^\top\mathbf{k}_r = -\mathbf{A}^\top(\mathbf{B}\mathbf{Q}_{11}\mathbf{B}^\top)^{-1}\mathbf{w}. \quad (1.72)$$

With (1.71) and (1.72) the reduced system of equations

$$\begin{bmatrix} \mathbf{A}^\top(\mathbf{B}\mathbf{Q}_{11}\mathbf{B}^\top)^{-1}\mathbf{A} & -\mathbf{C}^\top \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\hat{\mathbf{x}} \\ \hat{\mathbf{k}}_r \end{bmatrix} = \begin{bmatrix} -\mathbf{A}^\top(\mathbf{B}\mathbf{Q}_{11}\mathbf{B}^\top)^{-1}\mathbf{w} \\ -\mathbf{w}_r \end{bmatrix}^{45} \quad (1.73)$$

results, see also (NEITZEL and JOHANNES 2009). With the abbreviations

$$\begin{aligned} \mathbf{M} &= \mathbf{B}\mathbf{Q}_{11}\mathbf{B}^\top \\ \mathbf{N} &= \mathbf{A}^\top\mathbf{M}^{-1}\mathbf{A} \end{aligned} \quad (1.74)$$

leads after (1.72) and rearrangement

$$\Delta\hat{\mathbf{x}} = \mathbf{N}^{-1}(\mathbf{C}^\top\mathbf{k}_r - \mathbf{A}^\top\mathbf{M}^{-1}\mathbf{w}). \quad (1.75)$$

Inserted in (1.69d) and rearranged to  $\mathbf{k}_r$  results in

$$\mathbf{k}_r = (\mathbf{C}\mathbf{N}^{-1}\mathbf{C}^\top)^{-1}(\mathbf{C}\mathbf{N}^{-1}\mathbf{A}^\top\mathbf{M}^{-1}\mathbf{w} - \mathbf{w}_r). \quad (1.76)$$

From (1.71), taking into account the above abbreviations, results

$$\mathbf{k} = -\mathbf{M}^{-1}(\mathbf{A}\Delta\hat{\mathbf{x}} + \mathbf{w}) \quad (1.77)$$

and under the transition of (1.69a)

$$\mathbf{v} = \mathbf{Q}_{11}\mathbf{B}^\top\mathbf{k}. \quad (1.78)$$

<sup>45</sup>Certain tasks may require some unknowns to be fixed during the adjustment process so that they are not corrected. In addition, stochastic a priori information may be available in the form of standard deviations for different unknowns resulting from e. g. a previous measuring step or campaign. Similar to the CM of observations  $\mathbf{Q}_{11}$  an a priori CM can then be established for the  $\mathbf{u}$  unknowns:  $\mathbf{Q}_{xx_{ap}} = \frac{1}{\sigma_0^2} \begin{bmatrix} \sigma_1^2 & \sigma_2^2 & \dots & \sigma_u^2 \end{bmatrix}$ . The matrix allows each unknown to be assigned an individual weight. A major advantage of introducing a priori precision information in this form is that no need to introduce it via so-called *pseudo observations*. Since unbiased estimates for the unknowns are available, requirement  $\Delta\hat{\mathbf{x}}^\top\mathbf{Q}_{xx_{ap}}^{-1}\Delta\hat{\mathbf{x}}$  must be established in addition to the *Lagrange function* of the LSM in (1.68) (CASPARY 2007). The equation system to be solved from (1.73) is then extended via the normal equation matrix  $\mathbf{N} = \mathbf{Q}_{xx_{ap}}^{-1} + \mathbf{A}^\top(\mathbf{B}\mathbf{Q}_{11}\mathbf{B}^\top)^{-1}\mathbf{A}$ .

NOTE: Considering Equations (1.75) to (1.78), unlike (1.73), the system can also be solved via a *Cholesky factorization*<sup>46</sup>, which drastically reduces the memory requirements and the computing time rapidly decreases.

The unknowns are then obtained via the approximate values  $\mathbf{x}^0$  and the adjusted corrections from (1.75), which together with the vector of residuals from (1.78) enter the following iteration step as new approximate values.

#### 1.8.8 DETECTION AND ELIMINATION OF OUTLIERS IN AN OBSERVATION SERIES

The detection and elimination of outliers occupies a very high priority in the adjustment calculation, see Section 1.8, and the general statistical evaluation of measurement data.

NOTE: Only data that have been filtered of errors allow a distinct and meaningful interpretation of the final results.

##### 1.8.8.1 OVERALL MODEL TEST

The so called *global test*<sup>47</sup> (NIEMEIER 2008, p. 167) is used for the overall assessment of the adjustment result. It compares the *empirical standard deviation*  $s_0$  after the adjustment (a posteriori) with the previously assumed theoretical a-priori value  $\sigma_0$ , the *standard deviation of the unit weight*<sup>48</sup>

$$T_f = \frac{\sigma_0}{s_0} \quad (1.79)$$

where

$$s_0 = \frac{\mathbf{v}^T \mathbf{Q}_{11}^{-1} \mathbf{v}}{r} \quad (1.80)$$

and  $\mathbf{v}$  denotes the residual vector and  $\mathbf{Q}_{11}$  the CM of the observations. The value  $r$  describes the degree of freedom, also used as redundancy, of the specified adjustment problem. Depending on how the adjustment was formulated, the degree of freedom  $r$  is calculated generally by  $r = n - u$  or in case of existing restrictions (GHILANI 2010, p. 429) by  $r = n - u + c$ , where  $n$  represents the number of observations and  $u$  the set of unknown parameters as well as  $c$  that of the constraint equations.

<sup>46</sup>Also known as *Cholesky decomposition*. The matrix must be positively definite.

<sup>47</sup>Alternatively the term *overall model test* is used, e. g. by (TEUNISSEN 2000, p. 93).

<sup>48</sup>Also referred to as *theoretical reference variance*

<sup>49</sup>See also Equation (1.38) and Equation (1.41)

The evaluation is done with a test, see e. g. (TEUNISSEN 2000, p. 93; NIEMEIER 2008, p. 167). If the test turns out to be positive, this may indicate gross outliers in the data material. However, the test can also be affected by an incomplete modelling (functional model), which, strictly speaking, can only be proven when there are no more gross errors in the measured data. Furthermore, a positive test can also indicate an incorrect stochastic model.

#### 1.8.8.2 INDIVIDUAL TEST OF OBSERVATIONS

The concept of detecting gross errors (*blunders*) in data is known as *data snooping* (BAARDA 1968). If the hypothesis, see NIEMEIER (2008) p. 167, of the *global test*, see Section 1.8.8.1, is rejected, a deeper insight into the measured data is obligatory. This analysis is also called *local test* because it only refers to the individual observation  $l_i$ .

However, the approach is based on the assumption that a gross error mainly affects only the respective correction or residual  $v_i$  of the individual observation  $l_i$ . As test variable the so-called normalized residual

$$NV_i = \frac{v_i}{\sigma_{v_i}} = \frac{v_i}{\sigma_0 \sqrt{q_{vv_{ii}}}} \quad (1.81)$$

is used. For the calculation of the test quantities  $NV_i$ , the diagonal elements  $q_{vv_{ii}}$  of the CM of the residuals  $Q_{vv}$  are used, see also Equation (1.85). An important aspect for the calculation of (1.81) is that the theoretical standard deviation of the unit of weight  $\sigma_0$  is to be used, since the empirical value  $s_0$  could already be falsified by outliers.

The assessment of the test value is again based on a statistical test procedure, e. g. (NEITZEL 2010). Strictly speaking, the test is only valid for the one observation  $l_i$  with the largest value  $NV_i$ .

NOTE that this method is only suitable for datasets that contain no more than 3-5% of outliers in relation to all observations (NIEMEIER 2008, p. 218).

#### 1.8.9 QUALITY ASSESSMENT

In all fields of geodesy, it is of central importance to judge the adjustment result. This is done in the form of precision assessments of the estimated parameters. Here the statement on the reliability of the accuracy measures is of major importance, not at least for the detection of measurement errors, see Section 1.8.8.2. Furthermore, the choice of the *functional model*, see Section 1.8.2, is not insignificant and directly related to the reliability of the precision of the estimated parameters.

I.8.9.1 ACCURACY MEASURES

The knowledge of how precisely the unknowns were determined is as essential as the determination of the unknowns itself. However, a meaningful evaluation presupposes a correct *mathematical* and *stochastic model*, cf. Sections 1.8.2 and 1.8.3. The CM of the unknowns

$$\mathbf{Q}_{\Delta\hat{x}\Delta\hat{x}} = \mathbf{N}^{-1} - \mathbf{N}^{-1}\mathbf{C}\mathbf{Q}_{k_r k_r}\mathbf{C}^T\mathbf{N}^{-1}{}^{50} \quad (1.82)$$

can be formed by applying the *law of covariance propagation* with regard to Equation (1.75), considering

$$\mathbf{Q}_{k_r k_r} = (\mathbf{C}\mathbf{N}^{-1}\mathbf{C}^T)^{-1}, \quad (1.83)$$

whereupon the precision measures of the quantities to be estimated can be derived. From the diagonal elements  $q_{xx_i}$  of matrix (1.82), the standard deviations

$$\sigma_{x_i} = s_0\sqrt{q_{xx_i}} \quad (1.84)$$

can be specified for the unknowns using the empirical weighting factor  $s_0$ .

I.8.9.2 RELIABILITY MEASURES

In addition to precision, see Section 1.8.9.1, reliability is an important quality criterion. It refers to the control of the observations and thus characterizes the quality of the realization. The reliability measures are formed from the CM of the residuals

$$\mathbf{Q}_{vv} = \mathbf{Q}_{ll}\mathbf{B}^T\mathbf{Q}_{kk}\mathbf{B}\mathbf{Q}_{ll}, \quad (1.85)$$

which is obtained by applying the *law of covariance propagation* to (1.78), taking into account the CM of the correlates with respect to conditions

$$\mathbf{Q}_{kk} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{A}\mathbf{Q}_{\Delta\hat{x}\Delta\hat{x}}\mathbf{A}^T\mathbf{M}^{-1}. \quad (1.86)$$

The derivation of reliability measures is closely related to the *redundancy components* (FÖRSTNER 1979) of the observations  $l_i$

$$\mathbf{Q}_{vv}\mathbf{P} = \mathbf{Q}_{ll}\mathbf{B}^T\mathbf{Q}_{kk}\mathbf{B}, \quad (1.87)$$

<sup>50</sup>The structure of the CM follows the vector of unknowns  $\Delta\hat{x}$ .

which can be derived from (1.85). The diagonal elements  $r_i$  of the matrix (1.87) represent the *redundancy components*. The sum of all *redundancy components*

$$\sum_{i=1}^n r_i = \text{trace}(\mathbf{Q}_{vv}\mathbf{P}) = r \quad (1.88)$$

results in the total redundancy of the adjustment problem.

**IMPORTANT NOTE:** The *redundancy component*  $r_i$  indicates the amount by which a possible measurement error is reflected in the correction  $v_i$ . In order to be able to detect gross errors in the observations, see e. g. Section 1.8.8.2, sufficiently high *redundancy components* are therefore required. The redundancy is thus a measure of the controllability of an observation.

## 1.9 ON STOCHASTIC MODELLING OF TLS MEASUREMENTS

As for all other surveying instruments, the decision whether a specific sensor is suitable for a certain task or not is made on the basis of the specified accuracy, see Section 1.2.4.1 and the measurement noise, see also Section 1.2.4.1 and more general Section 0.2. The individual weighting of single observations based on a suitable stochastic model, cf. Sections 0.2.1.1 and 1.8.3, is of immense importance for many applications in the field of TLS, for instance:

- Identification of outliers related to adjustment calculation e. g. (BAARDA 1968), see also Section 1.8.8.2
- Statistically significant identification of deformations cf. e. g. (WUJANZ 2016)
- Sensor calibration of TLS's e. g. (LICHTI 2007), see also Section 1.2.5
- Registration (GRANT et al. 2012; BURGER et al. 2018), see Section 1.11 and segmentation (BURGER et al. 2017), see Section 1.10, of *point clouds*
- Direct georeferencing of TLS (SCAIONI 2005)
- Modelling the uncertainty of a MMS (MEZIAN et al. 2016)

Due to the individual magnitude of the uncertainty of individual observed distances of TLSs caused by the influencing factors affecting the distance measurement, see especially Section 1.2.4.1, no constant weights can be assumed for the observations (SODARISSANANE et al. 2011). Consequently, methods have been developed in the past to address this issue. BÖHLER et al. (2003) in an early publication on the topic derives the noise of the EDMU of a TLS from residuals of adjusted planes previously scanned. A similar approach was pursued by HEISTER (2006), who chose spherical targets instead of planes.

Even if the use of scanned geometric primitives to derive the distance stochastic seems self-explanatory at first glance, some influences with negative effects on the stochastic, such as

- accuracy of angle encoders,
- spatial resolution of the data as well as redundancy,
- accuracy of the applied geometric primitives and
- processing software,

see also (WUJANZ et al. 2017a), cannot be neglected. However, the majority of authors, including VOEGTLE et al. (2008) and SOUDARISSANANE et al. (2011), follow this approach.

Strictly speaking, however, the observations and thus also the distance measurement should be considered independently of each other in order to avoid dependencies in order to draw conclusions about the surface properties.

An alternative method is pursued by ZÁMEČNÍKOVÁ et al. (2014), determining the deviations between reference measurements and the observations of a TLS on a calibration track.

Another very simple *stochastic model*, referring only to the dependence of the distance to the object, see Figure 1.11, is used by ELKHRACHY and NIEMEIER (2006).

## 1.10 ON THE SEGMENTATION OF TLS POINT CLOUDS

Apart from the registration, see Sections 0.1 and 1.11, the segmentation belongs to the central work steps that contribute to the automation with regard to the further processing of *point clouds*, which have been captured e. g. with a TLS, cf. Section 0.1.

It often serves as the basis for the subsequent classification of segments into related object classes and enables, for instance, the extraction of characteristics (features) that can serve as information for a registration.

Basically, a distinction can be made between three major data structures to which segmentation algorithms can be applied:

- one dimensional (1D) data structure (e. g. list or array)
- two dimensional (2D) data structure (e. g. image grid or matrix, quadtree, graph),
- three dimensional (3D) or spatial data structure (e. g. octree, kd-tree)

The representation of a point cloud as a list or grid belongs to the group of non-hierarchical data structures, while the tree structures quadtree or octree (SAMET 2006, p. 211) belong to the hierarchical ones due to their structure.

The main purpose of applying appropriate data structures for the segmentation of point clouds is to quickly determine neighbouring points for selection and extraction. Spatial data structures are usually built up in this context in order to reproduce the topology lost due to the registration of *point clouds* as a result of the solely *cartesian* description.

The relations in a grid, such as from a [TLS](#) scan, however, are directly defined, which means that no time-consuming and computation-intensive topology creation as in the case of registered datasets is required. In this thesis, this advantage is exploited and, as a consequence, one focuses on the segmentation of a single scan in raster representation.

Regardless of the data structure on which segmentation algorithms are applied, two different categories can be differentiated:

- knowledge-based segmentation,
- data-driven segmentation.

In the case of knowledge-based or model-based approaches, a previous knowledge about the shape and form of the objects to be extracted is assumed. Using such a predefined model, suitable objects can be searched specifically ([PU and VOSSELMAN 2009](#)). However, if very heterogeneous data is provided, model-based concepts are less effective and a data-driven one is preferable. The segmented regions, when using a data-driven approach, do not have to correspond to the real objects – they can be assigned to the objects by a further analysis.

A data-driven segmentation based on a grid data structure (e. g. image grid) can be carried out in the following manners:

- point- or pixel-based (simple thresholding),
- edge-based (defined by contours),
- region- or area-based (using neighbourhood relations),
- hybrid (combination of methods).

The new segmentation approach outlined in [Chapter 3](#) below is based on a combination of a pixel-based and a region-based approach, which generally provide better results for noisy data compared to an edge-based solution.

With the region-based evaluation of *point clouds*, the segmentation pursues the goal of grouping points in a neighborhood into subsets (segments or even regions) according to predefined homogeneity criteria  $H$ .

Considering a data set  $P$ , which has to be split into  $n$  segments  $P_i$  ( $i = 1 \dots n$ ), the following five requirements can be demanded on a segmentation algorithm ([HOROWITZ and PAVLIDIS 1974](#); [JIANG and BUNKE 1997](#)):

1.  $\bigcup_{i=1}^n P_i = P$
2.  $P_i$  ( $i = 1 \dots n$ )
3.  $P_i \cap P_j = \emptyset$ , for  $i \neq j$

4.  $H(P_i) = \text{true}$ , for  $i = 1, 2 \dots n$
5.  $H(P_i \cup P_j) = \text{false}$ , for  $i \neq j$

The quantity of all segments  $\bigcup_{i=1}^n P_i$  should on one hand match the entire data set  $P$  (Item 1). On the other hand, each point may only be assigned to one segment (Item 3), in which individual segment points have a certain relationship – e. g. connected in a neighborhood of 4 or 8 – (Item 2). Again, all points  $P_i$  of a region must satisfy a homogeneity criteria  $H$  (Item 4), however adjacent segments should be distinguishable by  $H$  (Item 5).

IT SHOULD BE NOTED THAT most segmentation methods do not fulfill all mentioned requirements and thus often an attenuation for 1. and 3. is condoned.

For the solution of the task of segmenting *point clouds* obtained with TLS, see also Section 0.1, there are many scientific contributions so far, which on the one hand pursue different strategies and on the other hand are based on different data concepts.

A well known method for segmentation is the *split and merge* procedure, which WANG and TSENG (2004) apply to *lidar* data. However, it has the disadvantage that the subdivision of the data does not follow their natural structures and the segmentation therefore depends on the merge order.

In contrast, the also widely used *region growing* technique, which is used for example by VO et al. (2015) on the basis of an *octree* data structure, see also Section 1.4, provides directly contiguous regions. Nevertheless, it is much more computationally intensive and the segmentation result depends on the choice of the so-called *seed points*. In addition, without further analysis, it only fulfils the criteria by Item 3 and Item 4 of the requirements for a segmentation algorithm.

RABBANI et al. (2006) present a similar approach on the example of industrial plant segmentation, which also assumes an unstructured data representation.

BARNEA and FILIN (2013) use a *point cloud* in polar representation and take advantage of image information for segmentation in addition to geometry.

The data structure of the *point cloud* as a grid is taken from MAHMOUDABADI et al. (2016) and, like RIVEIRO et al. (2016), uses a *connected component* method, see also Section 1.5, for segmentation. Indeed, additional intensity and image information is also used, but first all data layers are processed separately and then combined to form a single result.

REMARK: Due to the various advantages and disadvantages, a skilful combination of methods is often used to achieve satisfactory results.

## 1.II ON THE REGISTRATION OF TLS POINT CLOUDS

Registration or referencing in general denotes an orientation procedure in geodesy and can be applied to both 2D or 3D data. The task of registration in the context of TLS is to transfer the *point clouds* resulting after the acquisition from individual standpoints with local coordinate systems into a common (superordinate) coordinate system, cf. also Section 0.1. The result of a registration is a complete dataset composed of 3D points captured from several connected standpoints. This process is of utmost importance in the completion of a dataset.

The process of transferring a *point cloud* into another coordinate system is known as transformation, Section 1.7. In order to transform several *point clouds* present in a local coordinate system into a common coordinate system, there are different strategies that can be reasonable depending on the type of task, the present data and the availability of additional measuring instruments or sensors.

In principle, the following registration procedures can be distinguished:

- Use of Artificial Targets
- Georeferenced Approaches
- Data-Driven Approaches

As a standard in Terrestrial Laser Scanning (TLS), the use of special targets in the form of coded targets or spheres, which are distributed in the object space and subsequently identified in the individual scans and assigned to each other, has become established. Registration then takes place exclusively via the corresponding target centres estimated from the targets. The automatic identification of such targets is presented in (ABMAYR et al. 2008). A major advantage of using target marks is that the individual point clouds do not require to overlap apart from the target itself. A disadvantage is the additional time required to attach the target marks or the fact that it is not possible to place them at all, since the object space cannot or must not be entered.

The so-called georeferencing uses additional sensors to determine the position and orientation of the measuring system in the superior (superordinate) system. If additional sensors, e. g. GPS receivers for position determination and Inertial Measuring Units (IMUs) for orientation determination are used together with the actual acquisition system – e. g. TLS – on a surveying platform, this is referred to as direct georeferencing. A prerequisite for such a measuring concept is the knowledge of the relative orientations of the measuring systems to each other determined by a simultaneous calibration as well as a synchronous acquisition. Kinematic laser scanning is an example of such a concept (HESSE 2007), although there are also static applications (PAFFENHOLZ et al. 2010). An example of direct georeferencing for photogrammetric applications is described in (CRAMER et al. 2000).

Another type of direct georeferencing is known as tracking. An exterior, superordinate, system, usually a camera or multi-camera system, observes a measuring sensor and records its position and

orientation for each measuring position. Since the exterior system must remain unchanged, the measuring volume is limited.

One possibility of "indirect" georeferencing in terrestrial laser scanning is to measure the aforementioned targets or distinct object points with a total station in the context of a network measurement and to use the corresponding points identified in the respective laser scans for a direct transformation into a superior coordinate system.

Regarding data-driven solutions, the information used for registration are obtained literally from the captured data itself. No additional signalisation in the form of targets or the use of external sensors as in the previously presented procedures is required. The following methods can be distinguished from each other:

- surface-based,
- feature-based,
- geometry based.

Surface-based methods, such as the Iterative Closest Point (**ICP**) algorithm (**BESL and MCKAY 1992**), are very popular and are included in many commercial software solutions. Since **ICP** algorithms determine adjacent point pairs in order to determine transformation parameters in an iterative procedure, the respective point clouds must be approximately aligned to each other. Otherwise, the algorithm may converge to an incorrect local solution. Another surface-based method is referred to as "*least squares surface matching*" (**AKCA 2007; GRUEN and AKCA 2005**).

A further approach is to detect and match feature points in adjacent scenes, which can then be used for registration. A distinction is made between **3D**- (e. g. **RUSU et al. 2009**) and **2D**-features (e. g. **BOEHM and BECKER 2007**), which are obtained either solely via a distinctive geometry description in the *point cloud* itself or by using the intensity information and its subsequent allocation to **3D**-space. Feature detectors that use both geometry and radiometry in combination can minimise mismatches by providing a more explicit description.

When registering *point clouds*, however, finding corresponding information in the *point clouds* to be referenced is a challenge. Due to the quasi-areal mode of operation of the **TLS**, see **Section 1.2.1**, no direct point correspondences can be formed between adjacent scans, but must necessarily be derived from the geometry or radiometry in the environment. To overcome this circumstance, one can benefit from the advantages of the quasi-areal acquisition of the **TLS**, see **Figure 1.3**, by deriving the parameters of geometric primitives in the *point clouds* and using them as information for registration. These data-driven procedures based on the geometry of surface elements provide direct and unambiguous correspondences via approximated straight lines, planes, cylinders or spheres in the contiguous scans and make use of the high redundancy of the surface-based recording.

## FUNDAMENTALS AND RELATED WORK

Methods among them, which are based on the extraction of straight lines in 3D, are used by LICHTENSTEIN and BENNING (2010) or (VON HANSEN et al. 2008). Plane-based methods, that have their strengths especially in urban environments due to the favourable availability of surfaces, have been addressed in various publications (RIETDORF et al. 2004; VON HANSEN 2006; GIELSDORF 2009; DOLD 2010; PREVITALI et al. 2014).

A contribution which uses both planes, cylinders and spheres describe RABBANI et al. (2006).

REMARK: All contributions deal with registration techniques detached from the workflow of the actual segmentation.

*Das, wobei unsere Berechnungen versagen, nennen wir Zufall.*

Albert Einstein

# 2

## A comprehensive stochastic model for TLS

DESPITE THE WELL-ESTABLISHED TLSs, suitable models capable to describe the stochastic characteristics of reflectorless EDMUs, as evidenced e. g. by the research of BÖHLER et al. (2003) and SOUDARISSANANE et al. (2011), have only recently been proposed. From a geodetic point of view, this situation has been unsatisfactory, since a comprehensive knowledge of the precision is crucial in addition to the accuracy of measurements, cf. Section 0.2 and figure 2, for a variety of applications and the modeling of quality, see Sections 0.2.1 and 0.2.1.1. For the first time, therefore, a major breakthrough for a comprehensive *stochastic model* has been achieved with contributions (WUJANZ et al. 2016a; WUJANZ et al. 2016c; WUJANZ et al. 2017a; WUJANZ et al. 2017b; WUJANZ et al. 2018a; WUJANZ et al. 2018b), providing the initial basis for this chapter.

The reason why the stochastic properties have not been well understood so far is mainly caused by the poor understanding of the stochastic behavior of reflectorless EDMUs, leading to the development of TLS. In order to better understand the influences on distance noise, these are therefore considered separately in various contributions on this topic, see also Section 1.9:

- object distance (ELKHRACHY and NIEMEIER 2006),
- surface properties (ZÁMEČNÍKOVÁ et al. 2014) and
- different incidence angles (SOUDARISSANANE et al. 2011).

The influence of the object distance is largely independent of that of the incidence angle, see also Figure 1.12, allowing both to be considered separately in a stochastic model. However, the influences are distorted by the unknown surface properties  $\gamma_\lambda$ , see also Figure 1.11, which cannot be determined

without further effort. In addition, the determination of incidence angles for individual points in the neighbourhood is not particularly reliable due to the ratio of local point density and noise. For these reasons, a separate consideration of different influences is not practicable and purposeful, since the effects cannot be modeled explicitly due to their unknownness and interdependence.

Existing methods, see [Section 1.9](#), for the derivation of distance noise are also subject to several influences, which can significantly falsify the outcome.

The incompleteness of previous stochastic models, as well as the inappropriate approach to derive such models, leads in summary to three arguments ([WUJANZ et al. 2017a](#), p. 147), which are reflected in the motivation for this chapter:

1. The input parameters for the stochastic model should be independent to other elementary observations or derived quantities such as Cartesian coordinates.
2. All previously mentioned influences have an impact onto the signal's strength and hence the precision of distance measurements. A causal separation of influencing factors cannot be made.
3. *Raw intensity values*<sup>1</sup> are capable of capturing those influences provoked by the acquisition configuration as well as radiometric properties of a sampled surface. Hence, intensity values should be suitable to assign stochastic properties to rangefinders, if a characteristic behaviour can be observed.

To address these problems, a novel intensity-based stochastic model for the reflectorless EDMU of a Z+F IMAGER<sup>®</sup> 5006h is experimentally derived. The influences of the scanning geometry as well as the interaction between the emitted signal and the surface, see [Section 1.2.4.1](#), are considered completely.

To gain a better understanding of the relationships between the intensity behavior of an EDMU and the distance noise  $\sigma_\rho$ , [Section 2.1.1](#) introduces an experimental setup resulting in measurements representing the characteristics of an APD. In [Section 2.2](#), the remaining effects of the incidence angle  $\alpha$  and certain material properties on this characteristic curve are presented, followed by the derivation of the stochastic model in [Section 2.3](#). Based on two different experiments, see [Section 2.4](#), the stochastic model was successfully validated for three selected sampling rates. In case a 1D scan mode is not available for a certain TLS, a possibility to derive the stochastic model in 3D measurement mode is presented in [Section 2.5](#). Finally, the applicability of the general principle for determining stochastic models for TOF TLSs is demonstrated in [Section 2.6](#).

---

<sup>1</sup> Usually the intensity values for a data set are given either in a gray value range from 0-255 or normalized from 0-1, whereby they are always individually distributed over the entire range. As a result, the intensities from different data sets do not correspond to each other due to the use of different gray value stretching functions, thus *raw intensity values* are used. *Raw intensity values* recorded by the applied TLS Z+F IMAGER<sup>®</sup> 5006h theoretically range from zero to five million increments (Inc) and can be specially exported with the manufacturer software Z+F LaserControl<sup>®</sup>.

## 2.1 INTENSITY BEHAVIOUR – SNR AND PRECISION OF DISTANCE

The relationship between **SNR** and the achievable precision of measurements is also widely used in other fields of geodesy, such as photogrammetry (**ACKERMANN 1984**), leading to the motivation to find such a relationship in **TLS**.

## 2.1.1 EXPERIMENTAL ARRANGEMENT AND DATA ACQUISITION

For a deeper understanding of the dependency of signal strength to signal noise, equivalent with intensity to intensity noise for a **TLS**, one can take advantage of the influence of distance and surface reflectivity on the received intensity, see **Figure 1.ii**. A representative intensity characteristic is obtained by covering the largest possible intensity spectrum.

For this purpose, four so-called Alucore<sup>®2</sup> panels with a side length of 5 cm were used, coated on both sides with a different dull shade of grey (from white to black) and thus showing different surface reflectivities, see **Figure 2.i**.

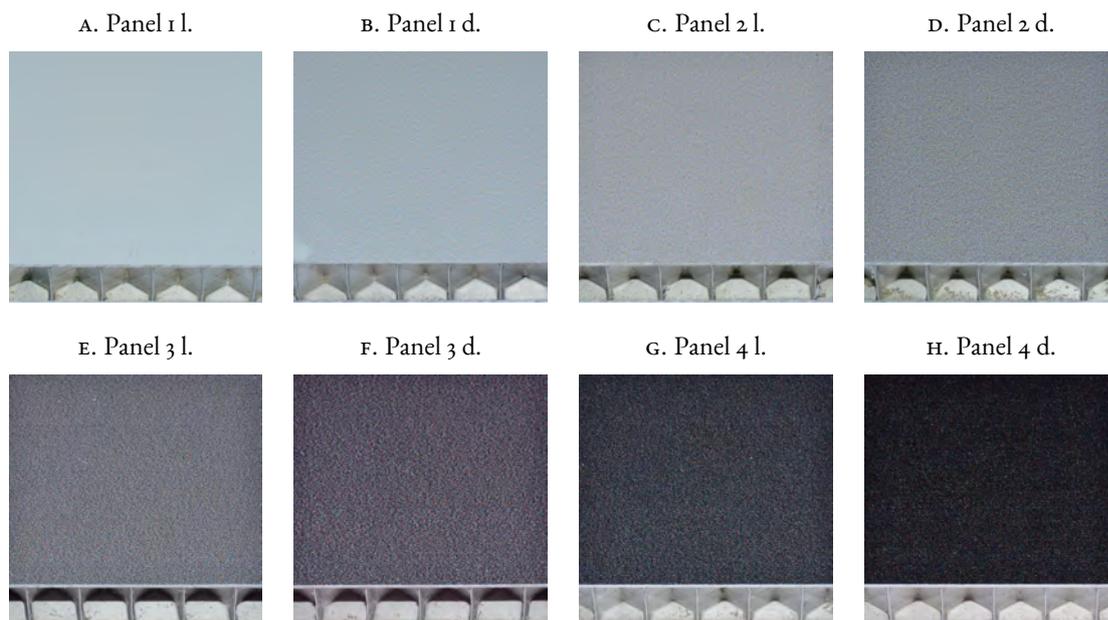


Figure 2.i: Imprinted ALUCORE<sup>®</sup> panels with different radiometric properties (grey values varying from 0-255).

<sup>2</sup> ALUCORE<sup>®</sup> is an aluminium composite panel consisting of two cover plates and an aluminium honeycomb core for versatile applications in architecture, transport and industry, characterised by its optimum planarity, formability and colour variety.

The panels attached to a tripod with a holder were finally captured<sup>3</sup> in 5 m steps at different distances between 5 and 44 m at an incidence angle of  $\sim 0^\circ$  in **iD** mode of the **TLS**, see [Figure 2.2](#), whereby the panels with their light (l) and dark (d) sides were interchanged at each position, resulting in 64 data sets.

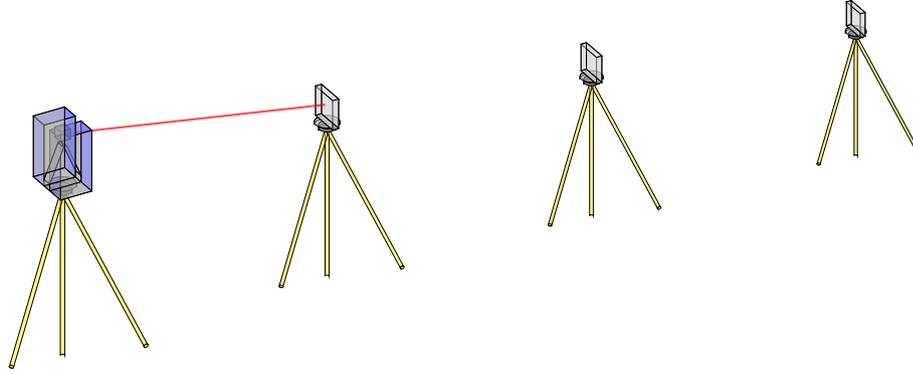


Figure 2.2: Standard measurement setup for the derivation of the stochastic model.

By data acquisition in **iD** mode, repeated observations of the distance between scanner and panel, as well as the intensity in the particular measurement rate, see also [Section 1.2.6](#), of 1016 kHz are recorded.

Based on  $n = 1000$  measurements with  $i = 1, \dots, n$  for the distance  $\rho_i$  and intensity  $A_i$ , of each data set, stochastic measures were derived to describe the behaviour of the **TLS's EDMU**. The mean values of the raw intensity values  $\bar{A} = \frac{\sum_{i=1}^n A_i}{n}$  and distances  $\bar{\rho} = \frac{\sum_{i=1}^n \rho_i}{n}$  as well as their corresponding empirical Standard Deviations (**SDs**)  $\sigma_A = \sqrt{\frac{\sum_{i=1}^n \Delta A_i^2}{n-1}}$  and  $\sigma_\rho = \sqrt{\frac{\sum_{i=1}^n \Delta \rho_i^2}{n-1}}$  were calculated based on the observations reduced by the mean values  $\Delta A_i = A_i - \bar{A}$  and  $\Delta \rho_i = \rho_i - \bar{\rho}$ .

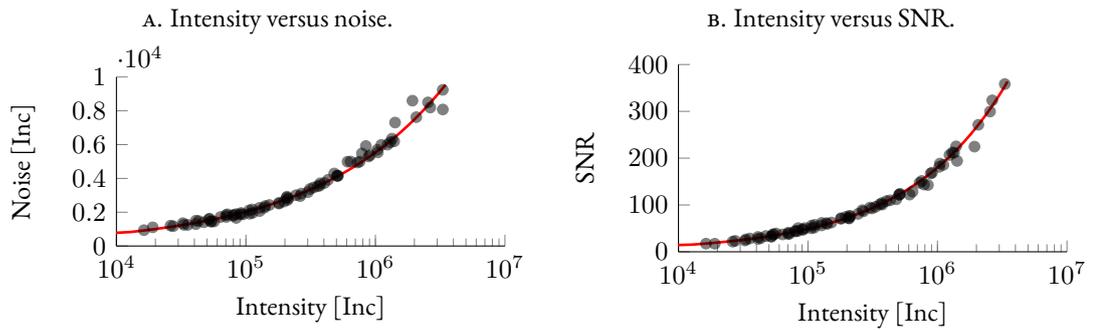


Figure 2.3: Intensity behaviour of the Avalanche Photo Diode (**APD**) for the **TLS Z+F IMAGER**<sup>®</sup> 5006h.

<sup>3</sup> In order to avoid possible falsification effects due to different ambient light conditions, the measurements were carried out in a cellar without windows.

## 2.1. INTENSITY BEHAVIOUR – SNR AND PRECISION OF DISTANCE

If one now applies the intensity noise  $\sigma_A$  as a function of the mean intensity  $\bar{A}$ , the characteristic curve of the EDMU's photodiode is shown in Figure 2.3A on the left. The amount of signal noise leading to the noise characteristic in Figure 2.3A is illustrated in Figure 2.4, which depicts the noise distributions for panel 4 l at different distances. Figures 2.4A to 2.4D indicate that the histograms become narrower with increasing range, resulting in lower signal noise  $\sigma_A$ .

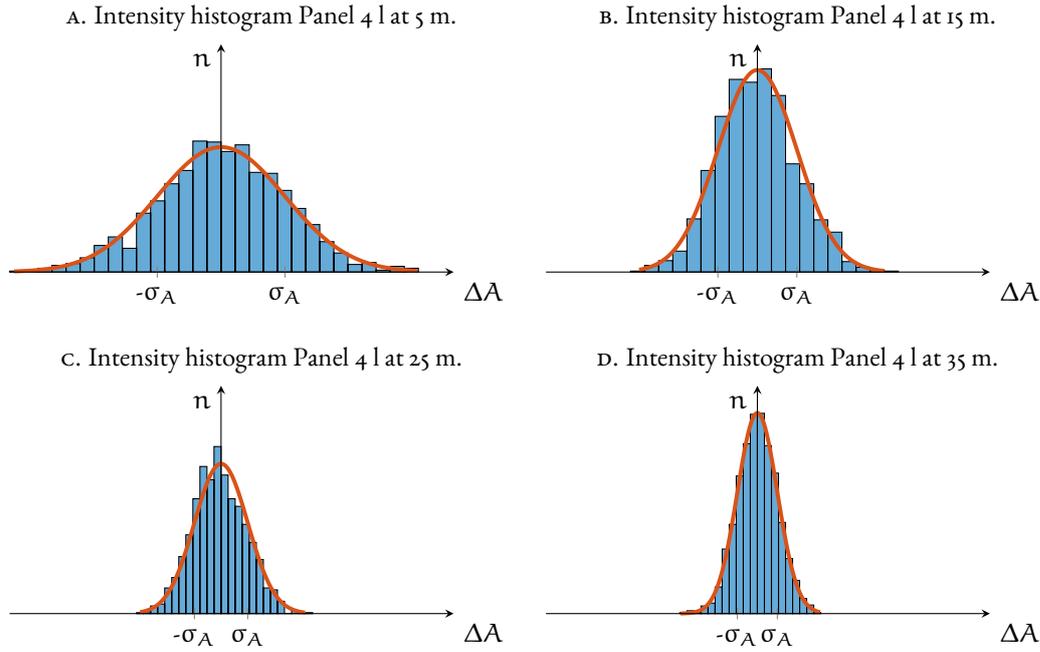


Figure 2.4: Histograms of the intensity measurements for the panel 4 l at different distances resulting from the maximum sampling rate of 1016 kHz, see also Section 1.2.6..

With the definition of the Signal-to-Noise Ratio (SNR)

$$\text{SNR} = \frac{\bar{A}}{\sigma_A} \quad (2.1)$$

between the mean signal amplitude  $\bar{A}$  and the SD of the amplitude  $\sigma_A$ , a further characteristic curve of the mean intensity as a function of the SNR can be represented in Figure 2.3B on the right.

Figure 2.3A shows that on the one hand the signal noise of the APD increases with increasing signal strength, but on the other hand Figure 2.3B shows a simultaneous increase of the SNR with increasing signal strength. Since the SNR is directly related to the precision of the distance measurement (METTENLEITER et al. 2015, p. 51), larger SNRs should therefore result in more favorable distance precisions  $\sigma_\rho$ . Consequently, and due to the recognizable characteristic of the SNR, see Figure 2.3B, it is obvious to obtain a comparable characteristic for the precision of the distance, cf. Section 2.1.2.

IT SHOULD BE NOTED THAT each observation series contains potential outliers having a direct impact on the determination of mean values and standard deviations for intensity and thus also on the characteristic curves, see [Figure 2.6](#), hence it is necessary to eliminate them beforehand.

However, in practice data snooping ([BAARDA 1968](#)), see also [Section 1.8.8.2](#), or so-called robust estimators ([ROUSSEEUW and LEROY 1987](#)) are used to detect outliers, presuppose the stochastic information to be determined itself, and thus are not suitable. Assuming normally distributed measurements ([GHILANI 2010](#), p. 525 ff.), erroneous observations were therefore consequently rejected by applying the so-called  $3\sigma$ -rule ([PUKELSHEIM 1994](#)). The outlier rate of the cleaned data series was on average 0.29% and maximum 1.19%.

### 2.1.2 INFLUENCE OF OBJECT DISTANCE AND RADIOMETRY TO DERIVE DISTANCE NOISE

In comparison to the intensity histograms, see [Figure 2.4](#), the corresponding distance histograms, see [Figure 2.5](#), behave differently with regard to the Panel 4 l. With increasing distance, broader distributions are obtained for the distance measurements  $\rho$ , i. e. less precise ( $\sigma_\rho$ ) mean values  $\bar{\rho}$  for the distances.

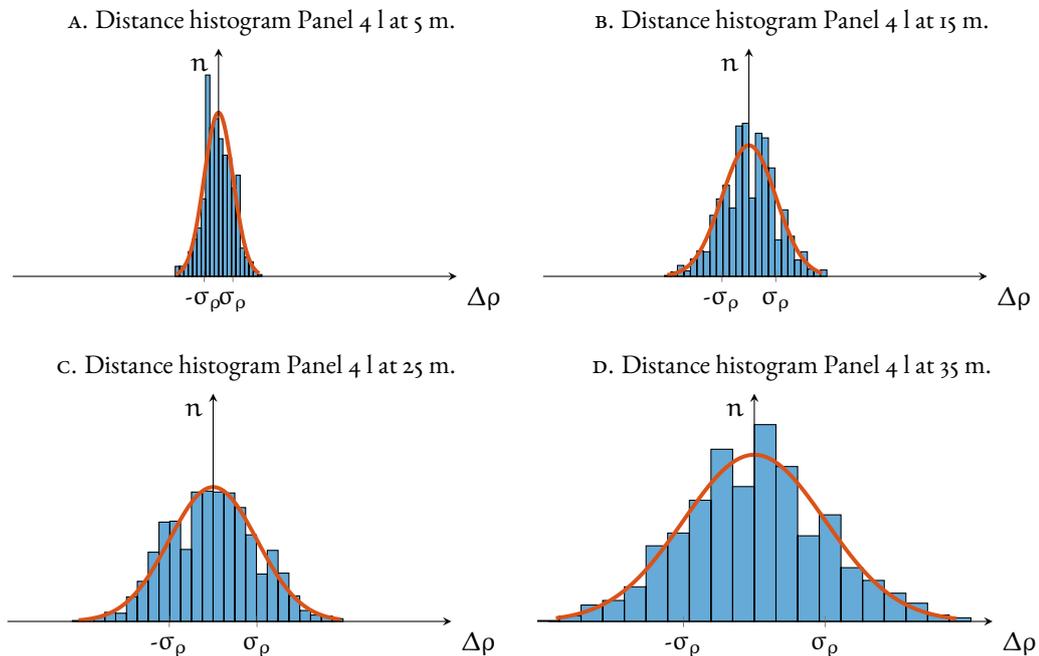


Figure 2.5: Histograms of the distance measurements for the panel 4 l at different distances corresponding to the intensity histograms shown in [Figure 2.4](#).

Applying the precision of the distances  $\sigma_\rho$  as a function of the average signal strength  $\bar{A}$ , see Figure 2.6, results in a reciprocal relationship – represented by blue data points related to the maximum sampling rate of 1016 kHz – to Figure 2.3A due to the distance histograms which are inversely proportional to the intensities.

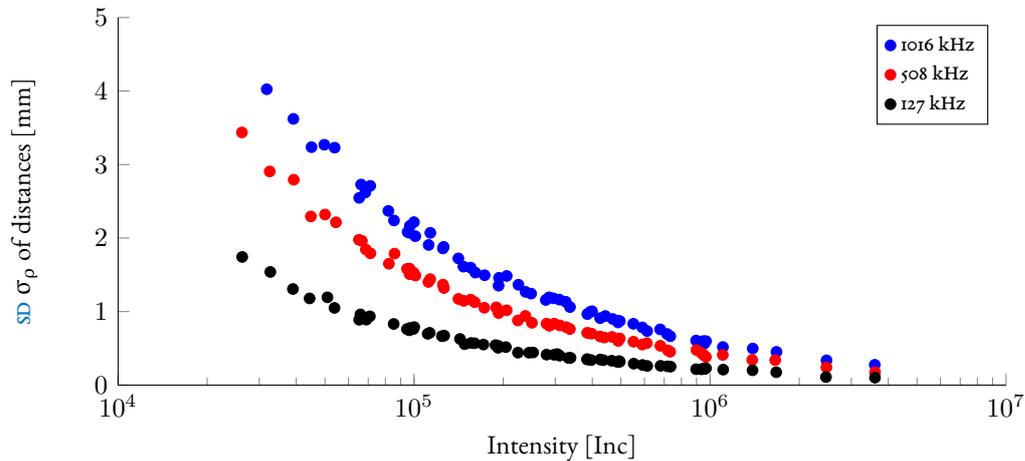


Figure 2.6: Precision of the distance measurement  $\sigma_\rho$  as a function of the mean intensity  $\bar{A}$  for the sampling rates of 1016 kHz (blue dots), 508 kHz (red dots) and 127 kHz (black dots) for the TLS Z+F IMAGER<sup>®</sup> 5006h.

Red data points characterize measurements recorded at a sampling rate of 508 kHz and black data points indicate those recorded at 127 kHz. The different sampling rates result in three curves expressing the characteristics of the Avalanche Photo Diode (APD), whereby the precision of the distance  $\sigma_\rho$  increases with increasing signal strength regardless of which sampling rate has been applied.

The averaging of the measured values when using different sampling rates, see Section 1.2.6, has a direct effect on the noise of the distance measurement, which is reflected in an offset of the courses. The precision of the distance measurements with respect to the basic measurement rate follows approximately the relation  $\sqrt{k}$  – corresponds to the standard deviation of the mean value (GHILANI 2010, p. 20) –, where  $k$  is the ratio between the basic (1016 kHz) and the selected sampling rate.

For example, a distance recorded at a sampling rate of 508 kHz has a  $\sqrt{2}$  times lower noise compared to the corresponding one recorded at the basic sampling rate of 1016 kHz.

The noise behaviour of the intensity measurement, see Figure 2.3, as well as that of the distance measurement, see Figure 2.6, results from the use of the intensity-influencing quantities of distance paired with different surface reflectivities (panels in Figure 2.1) by the experimental setup in Figure 2.2.

Figure 2.7 shows a more detailed analysis of the 127 kHz data points from Figure 2.6, highlighting the influence of range on distance noise with circle edges color-coded from red (5 m) to blue (44 m). The circle areas are coloured in the shades of grey of the respective panels used, see Figure 2.1, and thus represent the influence of radiometry on the distance noise.

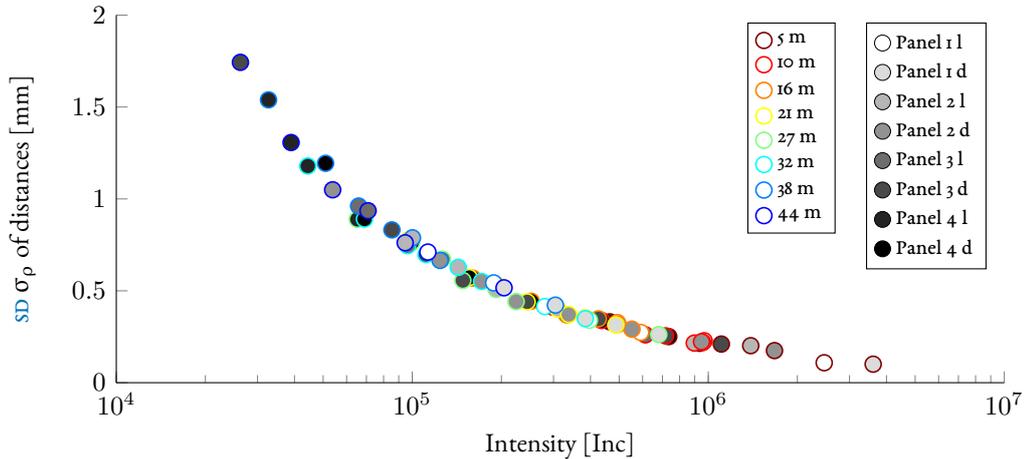


Figure 2.7: Color-coded classification of the influences of object distance  $\rho$  and radiometry on the precision of distance measurements  $\sigma_\rho$  at 127 kHz.

**INTERIM CONCLUSION:** In particular, overlapping data points show that an explicit cause for the deterioration of distance noise  $\sigma_\rho$  can neither be clearly assigned to the radiometric properties of a panel nor to the object distance. This confirms the hypothesis presented above, see Item 2, that a causal separation of the influencing factors onto the SNR cannot be made.

## 2.2 FURTHER FACTORS INFLUENCING THE DISTANCE STOCHASTICS

In this section, the influence of the incidence angle  $\alpha$  and the effect of different materials with special reflectance properties  $\gamma$  on the 127 kHz data series trend (black dots) shown in Figure 2.6 is examined.

### 2.2.1 IMPACT OF THE INCIDENCE ANGLE

It has already been proven by several publications, such as SOUDARISSANANE et al. (2011) or ZÁMEČNÍKOVÁ et al. (2014), that the incidence angle  $\alpha$  has a significant influence on reflectorless distance measurement of EDMUs, accompanied by an increasing loss of intensity with growing incidence angles, see also Figure 1.12.

## 2.2. FURTHER FACTORS INFLUENCING THE DISTANCE STOCHASTICS

To clarify the effects of the incidence angle  $\alpha$  on the characteristic course of the distance noise determined so far, see Figure 2.6, the panel 4 l, see Figure 2.16, was recorded in an experiment at object distances of 15 and 30 m in *id* mode at incidence angles<sup>4</sup> between 16 and a maximum of 63°.

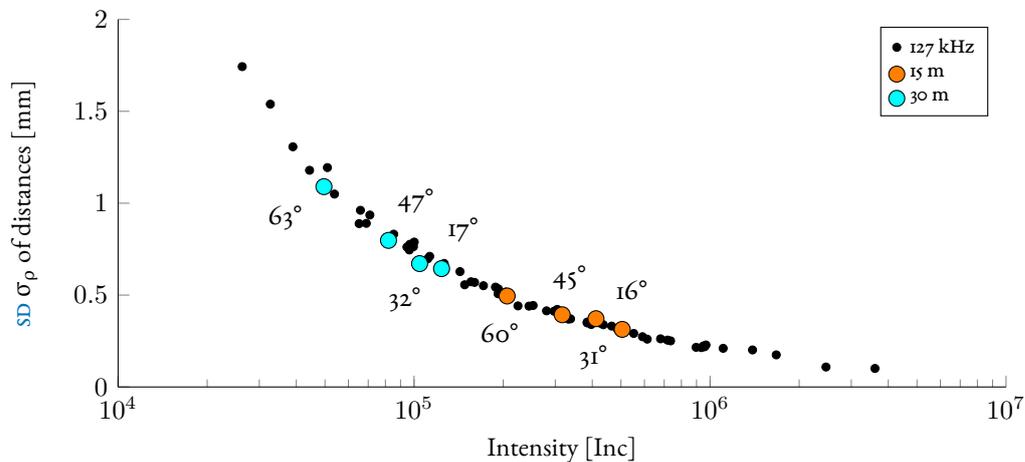


Figure 2.8: Influence of the incidence angle  $\alpha$  onto the distance precision  $\sigma_\rho$  for two different object distances.

The measurements were evaluated in analogy to Section 2.1.1 and are plotted in Figure 2.8 together with the black dotted data series from Figure 2.6 recorded at the same measuring rate of 127 kHz. The points are colored according to their object distance, according to Figure 2.7 – orange represents 15 m and light blue 30 m.

As to be expected, the observation group at 15 m also shows better distance precision at higher intensity values compared to the observation group at 30 m. Within an observation group, the signal strength equally decreases with increasing incidence angles, which is reflected in a higher distance noise.

**INTERIM CONCLUSION:** However, it is obvious that these observations follow the general trend of the other observations (black dots), recorded at an incidence angle of  $\sim 0^\circ$ .

### 2.2.2 EFFECTS CAUSED BY DIFFERENT MATERIALS

The interaction between the emitted laser signal and the object surface, which is also termed object or surface reflectivity  $\gamma$ , is the subject of intense scientific attention with regard to TLS, such as (ZÁMEČNÍKOVÁ et al. 2014).

<sup>4</sup> The incidence angles were determined by the normal vector of the panel surface and the direction vector of the measuring beam.

The object reflectivity for a certain object cannot be determined directly due to the strong dependencies on other factors influencing the intensity, see [Section 1.2.4.1](#), especially [Equation \(1.15\)](#), hence no noteworthy results have been obtained in the past to describe these effects. [Sections 2.1.2](#) to [2.2.1](#), in particular [Figure 2.7](#) and [Figure 2.8](#), demonstrated that raw intensity values are suitable for mapping the effects caused both by the acquisition configuration and by radiometric properties.

In order to prove whether raw intensity values are also appropriate for considering the effects caused by different materials, samples<sup>5</sup> of common construction materials –[Figure 2.9](#)– with different radiometric and surface-related (e. g. roughness) properties were investigated.



Figure 2.9: Material samples of different characteristics to examine the stochastic model.

To derive the actual reflectivity  $\gamma_\lambda$  of the materials at different wavelengths, the samples were measured with a Perkin Elmer Lambda 19 spectrometer, providing the material spectra in [Figure 2.10](#).

<sup>5</sup> The material samples have a size of  $\sim 5 \times 5$  cm and vary in thickness from 1 to 2 cm.

## 2.2. FURTHER FACTORS INFLUENCING THE DISTANCE STOCHASTICS

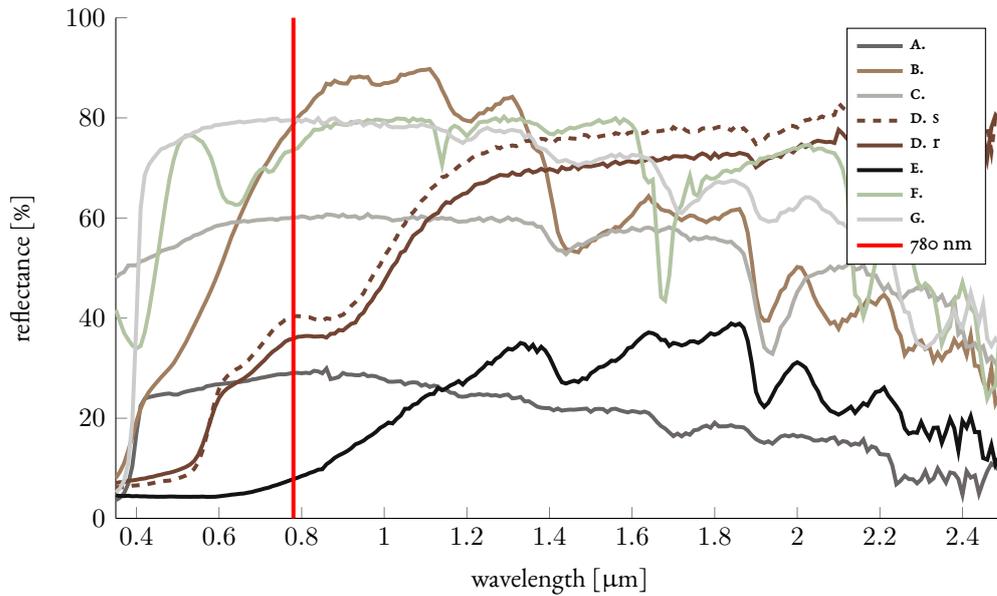


Figure 2.10: Spectral reflectance curves for different materials of a Perkin Elmer Lambda 19 spectrometer covering the wavelength range 350-2500 nm.

From the readings of the spectral curves at the wavelength of the applied TLS of 780 nm (red vertical line in Figure 2.10), the following reflectivities are obtained for the different materials:

- A. Acrylic-based wall plaster: 29.1% [Figure 2.9A](#)
- B. Beech panel: 78.7% [Figure 2.9B](#)
- C. Pumice (rough | smooth): 53.7% | 60.1% [Figure 2.9C](#)
- D. Red clinker: (rough | smooth): 35.8% | 40.4% [Figure 2.9D](#)
- E. Resin coated plywood: 7.8% [Figure 2.9E](#)
- F. Styrofoam: 73.6% [Figure 2.9F](#)
- G. White silicate plaster: 79.4% [Figure 2.9G](#)

For the first study, material samples A, B, E, F and G, see [Figure 2.9](#), were scanned from a distance of 15 m and 30 m as well as at incidence angles of  $0^\circ$  and  $20^\circ$ . The observations evaluated equivalent to [Section 2.1.1](#) are shown in [Figure 2.11](#) together with the reference trend (black dots) from [Figure 2.6](#) at the same measurement rate of 127 kHz.

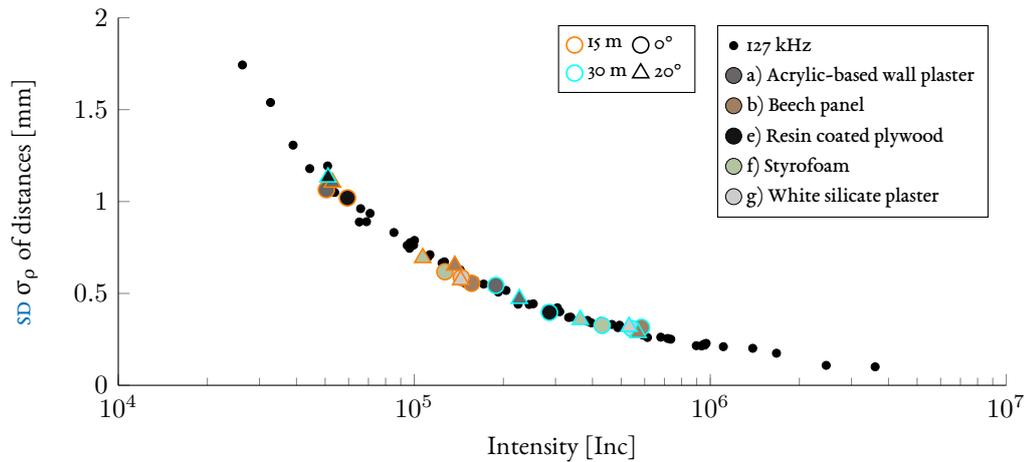


Figure 2.II: Influence of material samples from Figure 2.9 under varying acquisition configurations captured at 127 kHz.

To be able to assign the data patches to the corresponding materials and their acquisition configuration, the pattern face was colored in the corresponding material color, where the shape represents the incidence angle –circles for 0° and triangles for 20°– and the color of the shape edge –orange for 15 and light blue for 30 m– indicates the object distance.

**INTERIM CONCLUSION:** Regardless of which acquisition configuration the materials originate from or which immensely different levels of reflectivity they have, see list above, all observations follow the reference trend. Hence it can be concluded that the raw intensity values are also capable of expressing influences caused by different materials.

The two sides of the material samples c and d have very different degrees of roughness<sup>6</sup>, which is also reflected in different reflectivity values for the rough and smooth side, see material list above.

To also analyze the influences of the material’s roughness, both sides of the material samples c and d were again observed under different acquisition configurations (0 and 20°, 15 and 30 m) with the same sampling rate of 127 kHz applied before as a result of a second experiment.

Figure 2.12 shows the measurement of the rough side of the material by triangular data patches and the measurement of the smooth side by round data patches. The material type is represented by the respective material color of the surface patches.

<sup>6</sup> Sawing the pumice stone produces a porous cut surface with higher roughness. For clinker, the adhesive surface is comparatively smooth to the decorative granular outer layer. As a result, the corresponding reflection ratios are clearly different.

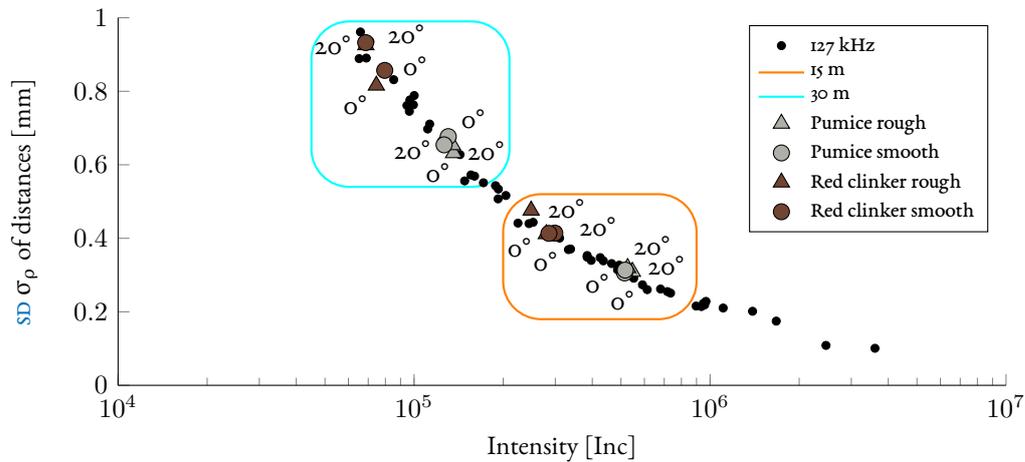


Figure 2.12: Effect of roughness and smoothness for different materials under differing acquisition configurations captured at 127 kHz.

In [Figure 2.12](#), four clusters can be identified as a result of different reflection ratios, which can be reduced to two clusters – rectangles in orange for 15 m and light blue for 30 m – depending on the object distance.

**INTERIM CONCLUSION:** As in the previous investigations, see [Section 2.1.2](#) and [Section 2.2.1](#), of other influencing factors, it is also evident here that raw intensity values are capable of mapping the influences from different material roughnesses on the previously determined course (black dots) regardless of the recording geometry.

### 2.3 THE STOCHASTIC MODEL OF A TLS

The results of the aforementioned experimental series, see [Sections 2.1](#) to [2.2](#), confirm the arguments under [Item 2](#) and [Item 3](#), according to which stochastic values for individual distances can be derived from raw intensity values.

[Figure 2.13](#) shows the relevant intensity range on the abscissa axis and the corresponding distance noise on the ordinate for a fictitious TLS. The lower and upper abscissa values of the black intensity curve represent the usable intensity range for the determination of the *stochastic model*, i. e. the range that should ideally be covered by intensity measurements of a TLS via a suitable acquisition configuration, see also [Figure 2.2](#), so that a meaningful model can be derived.

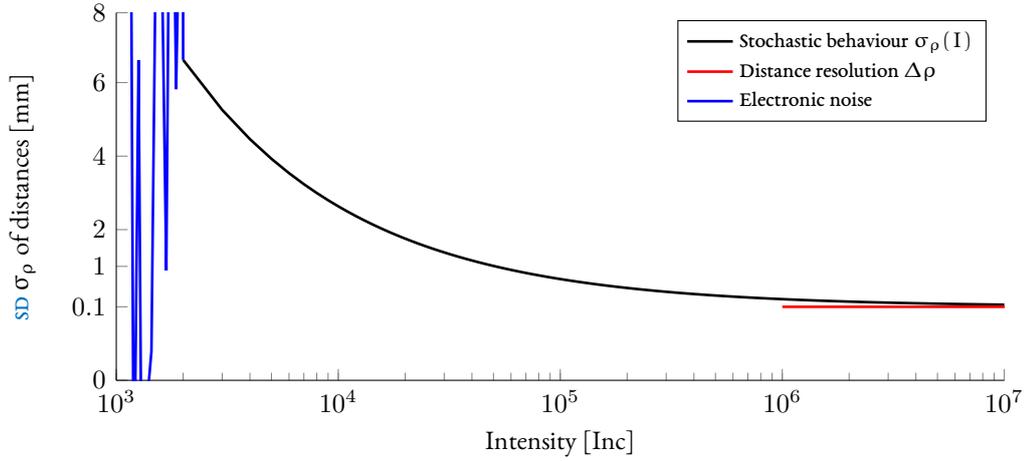


Figure 2.13: Characteristic intensity range to describe the distance noise for a fictitious *laser scanner*.

With decreasing intensity and increasing distance noise, a blue jagged area<sup>7</sup> is finally reached in which a correct distance measurement is no longer possible. This is due to electronic noise, occurring when the SNR, see also Figure 2.3B, becomes smaller than 2, i. e. the noise component begins to exceed the signal component.

In the higher intensity range, the black curve asymptotically approaches the red line, indicating the maximum resolution of the TLS applied.

NOTE: The stochastic model does not allow conclusions to be drawn below the resolution of the EDMU, since the distance measurements recorded within this intensity range differ randomly by the magnitude of the maximum resolution  $\Delta\rho$ .

Based on the characteristic decrease in distance precision of the APD in Figure 2.6, a power function –  $f : x \mapsto \alpha x^r \quad \alpha, r \in \mathbb{R}$  – ideally serves as *functional model*

$$\sigma_\rho = \alpha I^b + c \quad \text{or} \quad \sigma_\rho = \alpha I^{b^8} \quad (2.2)$$

to describe the stochastic model. Within an Least Squares Adjustment (LSA), the unknown parameters  $\hat{\alpha}$  and  $\hat{b}$  can be determined by adding the residuals  $v_{I_i}$  to the observed intensities  $I_i$ <sup>9</sup> and introducing

<sup>7</sup> The measurements recorded in the area of electronic noise have not been taken into account for further processing.

<sup>8</sup> For the TLS used, the parameter  $c$ , which describes an additional offset of the precision curve, was not significant and hence was not considered in the functional model. For other *laser scanners*, however, it might be of some importance!

<sup>9</sup> The intensities  $I_i$  are equivalent to the mean intensities  $\bar{A}$  derived in Section 2.1.1.

the standard deviation of the distance measurements  $\sigma_{\rho_i}$  as fixed parameters<sup>10</sup>, leading to so-called *condition equations*:

$$0 = \hat{\mathbf{a}}(I_i + \nu_{I_i})^{\hat{\mathbf{b}}} - \sigma_{\rho_i} \quad \text{ii} \quad (2.3)$$

The adjustment problem with the nonlinear *condition equations* of the type according to (2.3), can ideally be solved in a **GHM**, see [Section 1.8.6](#), by linearizing, see [Section 1.8.2.1](#), the conditional equations at the location of suitable approximate values  $\mathbf{x}^0$  and  $\mathbf{v}^0$ .

After introducing the vector of the corrections to the approximate values

$$\Delta \hat{\mathbf{x}} = \begin{bmatrix} \Delta \hat{\mathbf{a}} & \Delta \hat{\mathbf{b}} \end{bmatrix}^T, \quad (2.4)$$

the *Jacobian matrix*

$$\mathbf{J}_1 = \begin{bmatrix} (I_1 + \nu_{I_1}^0)^{b^0} & \mathbf{a}^0 (I_1 + \nu_{I_1}^0)^{b^0} \ln(I_1 + \nu_{I_1}^0) \\ \vdots & \vdots \\ (I_i + \nu_{I_i}^0)^{b^0} & \mathbf{a}^0 (I_i + \nu_{I_i}^0)^{b^0} \ln(I_i + \nu_{I_i}^0) \\ \vdots & \vdots \\ (I_n + \nu_{I_n}^0)^{b^0} & \mathbf{a}^0 (I_n + \nu_{I_n}^0)^{b^0} \ln(I_n + \nu_{I_n}^0) \end{bmatrix} \quad (2.5)$$

can be set up with the partial derivatives of the conditional equations according to the unknowns  $\hat{\mathbf{a}}$  and  $\hat{\mathbf{b}}$ . Concerning the **GHM** the *Jacobian matrix*

$$\mathbf{J}_2 = \begin{bmatrix} \mathbf{a}^0 \mathbf{b}^0 (I_1 + \nu_{I_1}^0)^{b^0-1} & & & 0 \\ & \mathbf{a}^0 \mathbf{b}^0 (I_i + \nu_{I_i}^0)^{b^0-1} & & \\ & & \ddots & \\ 0 & & & \mathbf{a}^0 \mathbf{b}^0 (I_n + \nu_{I_n}^0)^{b^0-1} \end{bmatrix} \quad (2.6)$$

<sup>10</sup>Since the derivation of the measured values for the intensity and the distance measurement in [Section 2.1.1](#) is performed in the same way and the accuracy information  $\sigma_I$  of the respective mean intensity observation  $\bar{A}$  is taken into account in the adjustment, the observed precision  $\sigma_\rho$  of the distance measurement is assumed to be constant.

<sup>11</sup>The condition equation can also be reformulated into the form  $I_i + \nu_{I_i} = \left(\frac{\sigma_{\rho_i}}{\hat{\mathbf{a}}}\right)^{\frac{1}{\hat{\mathbf{b}}}}$  as a so-called *observation equations*, allowing an evaluation in a **GMM**, see [Section 1.8.5](#). However, due to the unfavorable differential quotients for the convergence behavior, an evaluation in the **GHM**, see [Section 1.8.6](#), is preferred in this case.

with the partial derivatives of the *condition equations* according to the residuals

$$\hat{\mathbf{v}} = \left[ \hat{v}_{I_1} \quad \cdots \quad \hat{v}_{I_i} \quad \cdots \quad \hat{v}_{I_n} \right]^T \quad (2.7)$$

for the observations  $I_i$ , with  $i = 1, \dots, n$ , is additionally obtained. Considering the *condition vector*

$$\Psi(\mathbf{v}^0, \mathbf{x}^0) = \begin{bmatrix} \mathbf{a}^0(I_1 + v_{I_1}^0)^{b^0} - \sigma_{\rho_1} \\ \vdots \\ \mathbf{a}^0(I_i + v_{I_i}^0)^{b^0} - \sigma_{\rho_i} \\ \vdots \\ \mathbf{a}^0(I_n + v_{I_n}^0)^{b^0} - \sigma_{\rho_n} \end{bmatrix} \quad (2.8)$$

in Equation (1.53) and the individual precisions of the intensities  $\sigma_{I_i}$  in the **CM** of the observations

$$\mathbf{Q}_{11} = \frac{1}{\sigma_0^2} \begin{bmatrix} \sigma_{I_1}^2 & & & 0 \\ & \ddots & & \\ & & \sigma_{I_i}^2 & \\ & & & \ddots \\ 0 & & & & \sigma_{I_n}^2 \end{bmatrix}, \quad (2.9)$$

the *Jacobians*  $\mathbf{J}_1$ , see (2.5), with  $\mathbf{A} = \mathbf{J}_1$  and  $\mathbf{J}_2$ , see (2.6), with  $\mathbf{B} = \mathbf{J}_2$  can now be used in Equation (1.60) to obtain the solution  $\Delta\hat{\mathbf{x}}$ , see (2.4).

The unknowns  $\hat{\mathbf{x}} = [\hat{a} \ \hat{b}]^T$  and residuals  $\hat{\mathbf{v}}$  are finally obtained from Equation (1.61).

**NOTE:** Since this is only the solution of a linearized substitute problem, the unknowns  $\hat{\mathbf{x}}$  and residuals  $\hat{\mathbf{v}}$  based on Equation (1.58) are to be introduced into the adjustment as new approximate values  $\mathbf{x}^0$  and  $\mathbf{v}^0$ . This iterative calculation is carried out until a selected break condition, cf. Equation (1.62), is reached. Possible outliers were eliminated by *data snooping*, see Section 1.8.8.2, after convergence in the respective iteration step.

Table 2.1 lists the results of the regressions for any sampling rate of the curves in Figure 2.6. Each parameter set  $[\hat{a} \ \hat{b}]$ <sup>12</sup> describes an individual distance noise behaviour of the EDMU in the TLS applied

<sup>12</sup>The estimated parameters of the stochastic model refer to *raw intensity values* as input for the *functional model* in Equation (2.2), yielding the precision for distances in meters.

for each sampling rate via the functional relationship in (2.2), where the values in round brackets are the standard deviations ( $1\sigma$ ) of the respective adjusted parameters.

Table 2.1: Estimated stochastic model parameters for different sampling rates of the TLS Z+F IMAGER<sup>®</sup> 5006h.

SAMPLING RATE [kHz]	MODEL PARAMETERS	
	$\hat{a}$ ( $\sigma_a$ )	$\hat{b}$ ( $\sigma_b$ )
1016	1.3168 (0.0414)	-0.5577 (0.0027)
508	1.1742 (0.0363)	-0.5756 (0.0027)
127	0.4696 (0.0209)	-0.5587 (0.0047)

Figure 2.14 shows the regression curve through the data points based on the adjusted parameters  $\hat{a}$  and  $\hat{b}$ , see Table 2.1, of the *stochastic model* by applying Equation (2.2) for the sampling rate of 508 kHz.

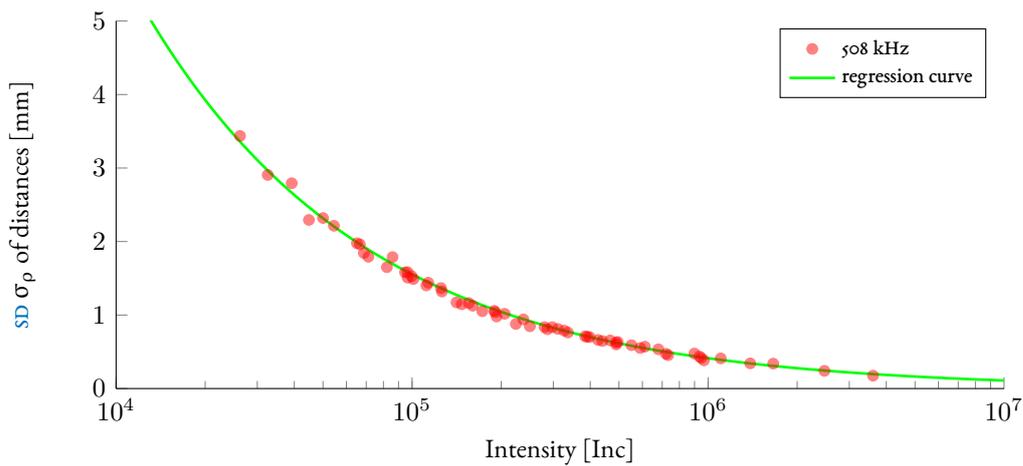


Figure 2.14: Regression curve through the data points with respect to the sampling rate of 508 kHz.

### 2.3.1 DERIVATION OF A STOCHASTIC MODEL FOR POINT CLOUDS

The precision of a 3D point obtained by a TLS depends on the precision of the corresponding original polar measurements, i. e. the precision of the direction  $\sigma_\phi$  and tilt angle  $\sigma_\theta$  as well as the precision of the distance  $\sigma_\rho$ , as illustrated by the functional relationship given by Equation (A.13).

The precision of the angle measurements can be taken from the manufacturer's data sheet for a TLS, resulting in  $\sigma_\phi = \sigma_\theta = 0.007^\circ$  for the Z+F IMAGER<sup>®</sup> 5006h used (ZOLLER+FRÖHLICH 2010).

On the basis of the observed raw intensity  $I$  for a 3D point and the proposed stochastic model parameters related to the sampling rate in [Table 2.1](#), [Equation \(2.2\)](#) can be used to determine the precision of the distance measurement  $\sigma_\rho$ .

By means of Variance-Covariance Propagation (VCP) ([GHILANI 2010](#), p. 86 ff.), it is now possible to determine the precision for the individual coordinate values of a 3D point, taking

$$\Sigma_{xx_i} = \mathbf{F}_i \Sigma_{ll_i} \mathbf{F}_i^T, \quad (2.10)$$

and considering the functional or design matrix

$$\mathbf{F}_i = \begin{bmatrix} \sin \theta_i \cos \phi_i & \rho_i \sin \theta_i \sin \phi_i & \rho_i \cos \theta_i \cos \phi_i \\ \sin \theta_i \sin \phi_i & \rho_i \sin \theta_i \cos \phi_i & \rho_i \cos \theta_i \sin \phi_i \\ \cos \theta_i & 0 & \rho_i \sin \theta_i \end{bmatrix}, \quad (2.11)$$

containing the partial derivatives with respect to the measured quantities via the functional relationship in [Equation \(A.13\)](#), as well as the corresponding CM

$$\Sigma_{ll_i} = \text{diag} \left[ \sigma_{\rho_i}^2 \quad \sigma_{\phi_i}^2 \quad \sigma_{\theta_i}^2 \right] \quad (2.12)$$

by knowing the precision of the polar elements. The precision of the 3D point, i. e. point error

$$\sigma_{3D_i} = \sqrt{\sum \text{trace} \Sigma_{xx_i}}, \quad (2.13)$$

can be easily calculated from the diagonal elements of the individual by [\(2.10\)](#) determined CM

$$\Sigma_{xx_i} = \begin{bmatrix} \sigma_{x_i}^2 & \sigma_{xy_i} & \sigma_{xz_i} \\ \sigma_{yx_i} & \sigma_{y_i}^2 & \sigma_{yz_i} \\ \sigma_{zx_i} & \sigma_{zy_i} & \sigma_{z_i}^2 \end{bmatrix}, \quad (2.14)$$

containing the precision information of the corresponding coordinate values.

[Figure 2.15](#) shows a color coded representation of 3D precision for an entire dataset calculated by [function](#) `compute_point_precision`, see [Code A.8](#), of the presented object in [Figure 3.1A](#).

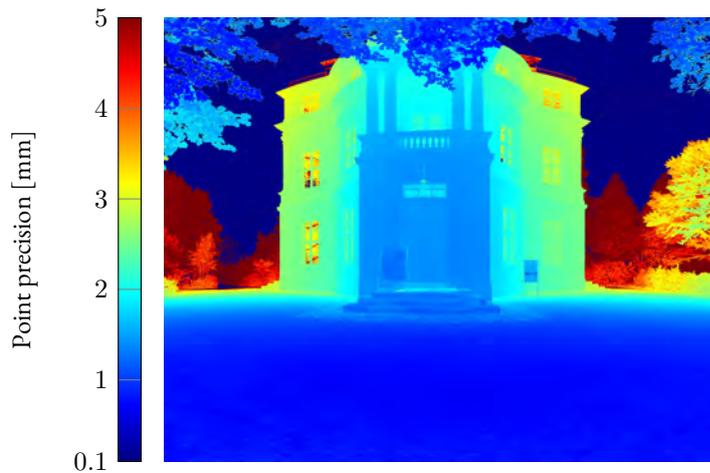


Figure 2.15: Representation of precision in 3D for the object presented in Figure 3.1A.

## 2.4 VALIDATION OF THE STOCHASTIC MODEL

This section focuses on the validation of the novel *stochastic model* presented in Section 2.3, for which two independent methods have been developed.

The first method presented in Section 2.4.2 is based on an overall analysis of the stochastic properties of the TLS, comprising the assumed precisions for directions, tilt angles and distances.

Since the first method is based on the evaluation of a geometric shape in form of a planar surface, its precision in planarity must be significantly below the precision of the scanner to be examined, an independent method is presented in Section 2.4.3. The second method is characterized by the use of unfavorable or non-smooth surface properties and focuses solely on the precision of the distances.

### 2.4.1 INTERPRETATION OF THE ADJUSTMENT RESULT AS VERIFICATION

Both methods for validating the *stochastic model* are based on the same concept that takes advantage of the properties of the *global test* presented in Section 1.8.8.1.

Under the assumption that the *functional model* was chosen appropriately and that there are no outliers present in the observation material, the *global test* is an indicator of the suitability of the *stochastic model* used.

USUALLY the statement whether the calculated *empirical SD*  $s_0$ , see Equation (1.80), matches the *SD of the unit weight*  $\sigma_0$  is made with the *chi-square test*, see Section 1.8.8.1. However, the test is limited to problem cases with a redundancy  $f < 20$ , since the error probability increases significantly with

increasing redundancy (FORNELL and LARCKER 1981; SMITH and McMILLAN 2001; NEITZEL and PETROVIC 2008).

As the redundancies exceed this limit immensely due to the numerous observations for the subsequent evaluations, a criterion established in practice is used according to which the stochastic model is assumed to be correct if  $s_0$  falls into the interval of  $0.7 < s_0 < 1.3$  with the assumption of  $\sigma_0 = 1$  (MÜLLER 2000, p. 345). The test then can be interpreted as follows:

- $s_0 > 1$ : Weighting of observations was too optimistic  
Observations are less precise than the model assumption
- $s_0 < 1$ : Weighting of observations was too pessimistic  
Observations are more precise than the model assumption

#### 2.4.2 VALIDATION BASED ON PLANAR SURFACES

For the verification process of the *stochastic model*, four planar panels with a side length of 50 cm, identical in material and radiometric properties to those in Figure 2.1, were scanned in 3D mode under different acquisition configurations, see Figure 2.16.

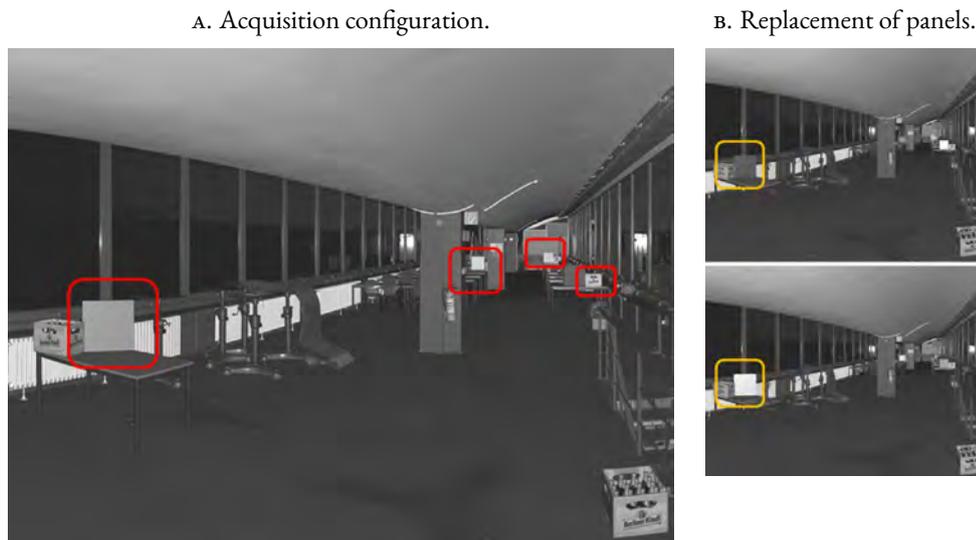


Figure 2.16: Arrangement of the panels for verification of the stochastic model for test scenario A, characterized by red rectangles (left), and interchange of the panels (right).

In a further working step, the 3D points of the panels were extracted from the *point cloud*<sup>13</sup> and the

<sup>13</sup> Since *point clouds* can usually only be exported in *cartesian coordinates*, a conversion into *polar coordinates* by Section A.1.2 is performed, representing the elementary observations of the scanner

## 2.4. VALIDATION OF THE STOCHASTIC MODEL

separate amounts of data were each subjected to a *plane regression*.

In order to ensure the appropriate *functional model* assumption postulated in [Section 2.4.1](#), here the assumption as a plane, the front and rear sides of each individual panel were checked for planarity with a GOM ATOS I structured light scanner. The inspection resulted in an average planarity of  $< 0.1$  mm, as also specified by the manufacturer, which is below the resolution of the [TLS](#) examined and thus an accurate model assumption could be proven.

The *functional model* for a planar surface with the common use of *cartesian coordinates* reads

$$n_x x_i + n_y y_i + n_z z_i - d = 0 \quad (2.15)$$

with  $i = 1, \dots, n$  points on the plane by considering the *constraint*

$$n_x^2 + n_y^2 + n_z^2 = 1 \quad (2.16)$$

for the length of the normal vector. Expressing the *cartesian coordinates* in [\(2.15\)](#) as a function of corresponding *polar coordinates* with [Equation \(A.13\)](#) results in the desired functional relationship

$$n_x \rho_i \sin \theta_i \cos \phi_i + n_y \rho_i \sin \theta_i \sin \phi_i + n_z \rho_i \cos \theta_i - d = 0 \quad (2.17)$$

of elementary observations. Assuming that the elementary observations  $\theta_i$ ,  $\phi_i$  and  $\rho_i$  are only subject to *random errors*, see also [Section 1.8.2](#), residuals for directions  $v_{\phi_i}$ , tilt angles  $v_{\theta_i}$  and distances  $v_{\rho_i}$  are introduced, leading to

$$\begin{aligned} & \hat{n}_x (\rho_i + v_{\rho_i}) \sin(\theta_i + v_{\theta_i}) \cos(\phi_i + v_{\phi_i}) + \\ & \hat{n}_y (\rho_i + v_{\rho_i}) \sin(\theta_i + v_{\theta_i}) \sin(\phi_i + v_{\phi_i}) + \\ & \hat{n}_z (\rho_i + v_{\rho_i}) \cos(\theta_i + v_{\theta_i}) - \hat{d} = 0 \end{aligned} \quad (2.18)$$

where

- $\hat{n}_x, \hat{n}_y, \hat{n}_z$  are the unknown adjusted values of the components of the normal vector of the estimated plane, as well as
- $\hat{d}$  the unknown adjusted perpendicular distance from the estimated plane to the origin of the local coordinate system of the scanner.

Summarizing the residuals in the vector

$$\hat{\mathbf{v}} = \left[ \hat{v}_{\phi_1} \quad \dots \quad \hat{v}_{\phi_n} \quad \hat{v}_{\theta_1} \quad \dots \quad \hat{v}_{\theta_n} \quad \hat{v}_{\rho_1} \quad \dots \quad \hat{v}_{\rho_n} \right]^T \quad (2.19)$$

and the precision relations of the observations with the *SD of the unit weight*  $\sigma_0 = 1$  in a *weight matrix*

$$\mathbf{P} = \text{diag} \left[ \frac{1}{\sigma_{\phi_1}^2} \quad \cdots \quad \frac{1}{\sigma_{\phi_n}^2} \quad \frac{1}{\sigma_{\theta_1}^2} \quad \cdots \quad \frac{1}{\sigma_{\theta_n}^2} \quad \frac{1}{\sigma_{\rho_1}^2} \quad \cdots \quad \frac{1}{\sigma_{\rho_n}^2} \right]^{14}, \quad (2.20)$$

the *objective function* to be minimized of an *LSA* is given by Equation (1.41). The functional relation (2.18) corresponds to the form according to Equation (1.50), which leads with the *constraint* of (2.16) to a nonlinear *LSA* with *condition equations* and constraints between the unknowns (MIKHAIL and ACKERMANN 1976).

Ideally, an iterative solution can be achieved with an evaluation and appropriate linearization according to LENZMANN and LENZMANN (2004) in a *GHM* with constraints, see Section 1.8.7.

Based on (2.18), the *condition vector* with the current approximate values for the residuals and unknowns is obtained

$$\Psi(\mathbf{v}^0, \mathbf{x}^0) = \left[ \Psi_1^0 \quad \cdots \quad \Psi_i^0 \quad \cdots \quad \Psi_n^0 \right]^T \quad (2.21)$$

by using the individual conditions

$$\begin{aligned} \Psi_i^0 = & \mathbf{n}_x^0(\rho_i + v_{\rho_i}^0) \sin(\theta_i + v_{\theta_i}^0) \cos(\phi_i + v_{\phi_i}^0) + \\ & \mathbf{n}_y^0(\rho_i + v_{\rho_i}^0) \sin(\theta_i + v_{\theta_i}^0) \sin(\phi_i + v_{\phi_i}^0) + \\ & \mathbf{n}_z^0(\rho_i + v_{\rho_i}^0) \cos(\theta_i + v_{\theta_i}^0) - \hat{d}. \end{aligned} \quad (2.22)$$

Similarly, the *constraint vector* based on (2.16) is obtained

$$\gamma^0 = \mathbf{n}_x^{0,2} + \mathbf{n}_y^{0,2} + \mathbf{n}_z^{0,2} - 1 \quad (2.23)$$

with the current approximate values for the unknowns. The *Jacobian matrix* with the partial derivatives of the *condition equations* in (2.21) according to the unknowns is obtained

$$\mathbf{J}_1 = \begin{bmatrix} \frac{\partial \Psi_1^0}{\partial n_x^0} & \frac{\partial \Psi_1^0}{\partial n_y^0} & \frac{\partial \Psi_1^0}{\partial n_z^0} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \Psi_i^0}{\partial n_x^0} & \frac{\partial \Psi_i^0}{\partial n_y^0} & \frac{\partial \Psi_i^0}{\partial n_z^0} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \Psi_n^0}{\partial n_x^0} & \frac{\partial \Psi_n^0}{\partial n_y^0} & \frac{\partial \Psi_n^0}{\partial n_z^0} & -1 \end{bmatrix} \quad (2.24)$$

<sup>14</sup>The precision of the directions  $\sigma_{\phi_i}$  and the tilt angles  $\sigma_{\theta_i}$  were set to  $0.007^\circ$  according to the manufacturer's specifications and the precision of the distances  $\sigma_{\rho_i}$  stems from the proposed stochastic model of TLS as already used previously in Section 2.3.1

based on Equation (1.52) by using the substitutions

$$\begin{aligned}\frac{\partial \Psi_i^0}{\partial n_x^0} &= (\rho_i + v_{\rho_i}^0) \sin(\theta_i + v_{\theta_i}^0) \cos(\phi_i + v_{\phi_i}^0), \\ \frac{\partial \Psi_i^0}{\partial n_y^0} &= (\rho_i + v_{\rho_i}^0) \sin(\theta_i + v_{\theta_i}^0) \sin(\phi_i + v_{\phi_i}^0), \\ \frac{\partial \Psi_i^0}{\partial n_z^0} &= (\rho_i + v_{\rho_i}^0) \cos(\theta_i + v_{\theta_i}^0).\end{aligned}\quad (2.25a)$$

With the partial derivatives of the *condition equations* in (2.21) according to the residuals

$$\begin{aligned}\frac{\partial \Psi_i^0}{\partial v_{\phi_i}^0} &= -n_x^0 (\rho_i + v_{\rho_i}^0) \sin(\theta_i + v_{\theta_i}^0) \sin(\phi_i + v_{\phi_i}^0) \\ &\quad + n_y^0 (\rho_i + v_{\rho_i}^0) \sin(\theta_i + v_{\theta_i}^0) \cos(\phi_i + v_{\phi_i}^0) \\ \frac{\partial \Psi_i^0}{\partial v_{\theta_i}^0} &= n_x^0 (\rho_i + v_{\rho_i}^0) \cos(\theta_i + v_{\theta_i}^0) \cos(\phi_i + v_{\phi_i}^0) \\ &\quad + n_y^0 (\rho_i + v_{\rho_i}^0) \cos(\theta_i + v_{\theta_i}^0) \sin(\phi_i + v_{\phi_i}^0) - n_z^0 (\rho_i + v_{\rho_i}^0) \sin(\theta_i + v_{\theta_i}^0), \\ \frac{\partial \Psi_i^0}{\partial v_{\rho_i}^0} &= n_x^0 \sin(\theta_i + v_{\theta_i}^0) \cos(\phi_i + v_{\phi_i}^0) \\ &\quad + n_y^0 \sin(\theta_i + v_{\theta_i}^0) \sin(\phi_i + v_{\phi_i}^0) + n_z^0 \cos(\theta_i + v_{\theta_i}^0),\end{aligned}\quad (2.26a)$$

the *Jacobian matrix*

$$\mathbf{J}_2 = \begin{bmatrix} \frac{\partial \Psi_1}{\partial v_{\phi_1}^0} & 0 & 0 & \frac{\partial \Psi_1}{\partial v_{\theta_1}^0} & 0 & 0 & \frac{\partial \Psi_1}{\partial v_{\rho_1}^0} & 0 & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & \frac{\partial \Psi_n}{\partial v_{\phi_n}^0} & 0 & 0 & \frac{\partial \Psi_n}{\partial v_{\theta_n}^0} & 0 & 0 & \frac{\partial \Psi_n}{\partial v_{\rho_n}^0} \end{bmatrix}\quad (2.27)$$

results based on Equation (1.52) analogous to the *Jacobian matrix*  $\mathbf{J}_1$ , see (2.24). Finally, the *Jacobian*

$$\mathbf{J}_3 = \begin{bmatrix} 2n_x^0 & 2n_y^0 & 2n_z^0 & 0 \end{bmatrix}\quad (2.28)$$

is obtained with the partial derivatives of the constraint equations in (2.23) after the unknowns. Using the *Jacobians* with  $\mathbf{A} = \mathbf{J}_1$ ,  $\mathbf{B} = \mathbf{J}_2$  and  $\mathbf{C} = \mathbf{J}_3$  and forming the *vectors of misclosures* according to Equations (1.53) and (1.66) using the *condition* and *constraint vectors* in (2.21) and (2.23), the solution for the vector of reduced unknowns  $\Delta \hat{\mathbf{x}} = [\Delta \hat{n}_x \ \Delta \hat{n}_y \ \Delta \hat{n}_z \ \Delta \hat{d}]^T$  is obtained with  $\mathbf{Q}_{ll} = \mathbf{P}^{-1}$ , see (2.20), taking Equation (1.73).

SINCE this is like in Section 2.3 only the solution of a linearized substitute problem, the unknowns  $\hat{\mathbf{x}}$  and residuals  $\hat{\mathbf{v}}$  based on Equation (1.61) are to be introduced into the adjustment as new approximate values  $\mathbf{x}^0$  and  $\mathbf{v}^0$ . This iterative calculation is carried out until a selected stop criterion, cf. Equation (1.62), is reached.

During the iteration steps, after each convergence<sup>15</sup> of  $s_0$ , BAARDA's *data snooping* (BAARDA 1968), see Section 1.8.8.2, was applied to detect and remove outliers in the observation material.

Two scenarios, A and B, with different acquisition configurations, were prepared to verify the *stochastic model* by satisfying the combination of distance (ELKHRACHY and NIEMEIER 2006) and incidence angle (GRANT et al. 2012) dependent models.

Regarding scenario A, panels with different radiometric properties, cf. Figure 2.1, were repeatedly measured at constant positions, highlighted by red rectangles in Figure 2.16A, by exchanging them among each other, highlighted by orange rectangles in Figure 2.16B, with a constant sampling rate of 508 kHz in order to quantify the influences of different radiometric properties while maintaining the same acquisition configuration on the new stochastic model.

Table 2.2 contains the empirical SDs  $s_0$  based on the adjusted planes, as well as the mean intensity  $I_m$ , for the different realisations of scenario A. For scenarios A1-A4, the corresponding scan geometry<sup>16</sup> is given in the first column of the table.

Table 2.2: Empirical standard deviation  $s_0$  after plane adjustment and mean intensity  $I_m$  for different scenarios A1 of the measuring arrangement A.

SCAN GEOMETRY	A1		A2		A3		A4	
	$s_0$	$I_m$ [Inc]						
Pos. 1: 4.94 m   12°	1.16	1.519·10 <sup>6</sup>	1.05	3.701·10 <sup>5</sup>	1.13	9.041·10 <sup>5</sup>	1.16	1.503·10 <sup>6</sup>
Pos. 2: 12.32 m   5°	1.11	9.142·10 <sup>5</sup>	1.11	1.025·10 <sup>6</sup>	1.01	2.533·10 <sup>5</sup>	1.04	5.283·10 <sup>5</sup>
Pos. 3: 19.42 m   7°	1.01	1.101·10 <sup>5</sup>	0.99	2.448·10 <sup>5</sup>	1.03	4.159·10 <sup>5</sup>	1.03	4.322·10 <sup>5</sup>
Pos. 4: 25.88 m   8°	0.95	1.395·10 <sup>5</sup>	0.97	2.313·10 <sup>5</sup>	0.92	2.308·10 <sup>5</sup>	0.96	6.054·10 <sup>4</sup>

IT CAN BE CONCLUDED THAT the proposed *stochastic model* is capable of describing effects caused by different radiometric properties as well as the acquisition configuration, cf. Section 1.2.4.1,

<sup>15</sup> Means that the difference of a computed value from the previous to the actual iteration step has reached a value close to zero in terms of a stop criteria

<sup>16</sup> The recording geometry refers to the distances between the centroids of the extracted planes and the origin of the local scanner coordinate system, as well as the incidence angle, defined as the angle between the plane normal and the incident beam of the TLS.

## 2.4. VALIDATION OF THE STOCHASTIC MODEL

since according to MÜLLER (2000), as can be seen in Table 2.2, all *empirical SDs*  $s_0$  fall within the given interval of  $0.7 < s_0 < 1.3$ .

To illustrate the influence of poorly chosen precision information for the observations or unsuitable *stochastic models* on the adjustment result, expressed by the *empirical SD*  $s_0$ , the measurements of scenario A4<sup>17</sup> from Table 2.2 were reprocessed.

For this purpose, different weight matrices  $\mathbf{P}_{c1}$ ,  $\mathbf{P}_{c1.5}$  and  $\mathbf{P}_{c2}$  were created, respectively based on the assumption of equally precise distances with  $\sigma_\rho = 1$  mm,  $\sigma_\rho = 1.5$  mm and  $\sigma_\rho = 2$  mm, and used as new precision models for the measured values from scenario A4 in further plane adjustments.

Table 2.3: Empirical *SDs*  $s_0$  based on different unrealistic *stochastic models* for scenario A4 from Table 2.2.

WEIGHT MATRIX	POSITION 1	POSITION 2	POSITION 3	POSITION 4
	$s_0$	$s_0$	$s_0$	$s_0$
$\mathbf{P}$	1.16	1.04	1.03	0.96
$\mathbf{P}_{c1}$	0.48	0.68	0.77	1.96
$\mathbf{P}_{c1.5}$	0.32	0.46	0.52	1.37
$\mathbf{P}_{c2}$	0.24	0.34	0.39	1.05

Table 2.3 compares the original *empirical SDs* based on the evaluation of the new derived stochastic model  $\mathbf{P}$  for the positions from scenario A4, see last column of Table 2.2, with the *empirical SDs* based on the unrealistic models  $\mathbf{P}_{c1}$ ,  $\mathbf{P}_{c1.5}$  and  $\mathbf{P}_{c2}$ . For the weight matrix  $\mathbf{P}_{c1}$  there are significant deviations (up to 0.96 for position 4) from the  $s_0$  of the individual positions to the *SD of the unit of weight*  $\sigma_0$  and thus also to the  $s_0$  of the proposed weight matrix  $\mathbf{P}$  (up to 1.0 for position 4), therefore 3 of 4 values do not lie within the required acceptance range.

IT CAN BE STATED THAT  $\mathbf{P}_{c1}$ , as well as the other weight matrices  $\mathbf{P}_{c1.5}$  and  $\mathbf{P}_{c2}$ , are not suitable for describing the stochastic properties of the TLS used due to the above findings.

FURTHERMORE, IT SHOULD BE NOTED THAT with regard to  $s_0$  in terms of a position (table columns 2, 3, 4 or 5 in Table 2.3), even changes in stochastics within submillimeter range, i. e. from 1 to 1.5 and from 1.5 to 2 mm, may lead to major changes in  $s_0$ , indicating a high sensitivity for the correct choice of a *stochastic model*.

<sup>17</sup>Due to its largest variation in intensity between the individual panel positions and the accompanying clearer significance for the experiment, scenario A4 was preferred to the others.

To estimate the sole influence of different radiometric properties on the distance noise  $\sigma_\rho$ , the highest and lowest intensity values  $I_m$  for a constant scanning geometry, here with respect to position 1 in [Table 2.2](#), were used in [Equation \(2.2\)](#) with the parameters of the *stochastic model* for the relevant 508 kHz from [Table 2.1](#) to calculate the corresponding precisions.

For the intensity values 370.104 and 1.519.370 the values 0.73 mm and 0.33 mm result for the distance noise  $\sigma_\rho$ , leading to a difference in precision<sup>18</sup> of 0.4 mm solely due to the radiometric variation.

AS ALREADY STATED BEFORE, see [Table 2.3](#), these submillimeter differences have a considerable effect on the *empirical SD*  $s_0$ , which clearly argues against the validity of range and/or incidence angle dependent *stochastic models* for TLS's EDMUs.

IN SUMMARY, IT CAN BE CONCLUDED that the previous results based on scenario A, see [Figure 2.16](#), support the argument in [Item 3](#) that raw intensity values are able to consider the influences caused by the acquisition configuration and the radiometric properties.

In order to check whether the newly developed *stochastic model* is capable of describing the distance noise of the EDMU differentiated by different sampling rates, with scenario B individual panels were repeatedly recorded with a constant acquisition configuration but with different sampling rates.

Table 2.4: Empirical standard deviation  $s_0$  after plane adjustment and mean intensity  $I_m$  for different sampling rates of scenario A4 from [Table 2.2](#).

SCAN GEOMETRY	B1 (1016 kHz)	B2 (508 kHz)	B3 (127 kHz)
	$s_0$   $I_m$ [Inc]	$s_0$   $I_m$ [Inc]	$s_0$   $I_m$ [Inc]
Pos. 1: 10.09 m   11°	1.02   4.398·10 <sup>5</sup>	1.02   4.424·10 <sup>5</sup>	1.04   4.452·10 <sup>5</sup>
Pos. 2: 12.17 m   13°	1.07   7.798·10 <sup>5</sup>	1.08   7.846·10 <sup>5</sup>	0.99   7.895·10 <sup>5</sup>
Pos. 3: 15.74 m   10°	1.04   5.517·10 <sup>5</sup>	1.04   5.536·10 <sup>5</sup>	0.98   5.589·10 <sup>5</sup>
Pos. 4: 17.02 m   9°	1.01   3.674·10 <sup>5</sup>	0.98   3.690·10 <sup>5</sup>	1.05   3.725·10 <sup>5</sup>

[Table 2.4](#) shows that the *empirical SD*  $s_0$  for the different sampling rates all fall within the required interval of  $0.7 < s_0 < 1.3$ , which confirms the validity of the parameter sets for the sampling rates in [Table 2.1](#) and thus the *stochastic models*.

<sup>18</sup> If the measurements had been carried out with a sampling rate of 1016 kHz, this would even lead to a precision difference of 0.56 mm.

## 2.4. VALIDATION OF THE STOCHASTIC MODEL

### 2.4.3 VALIDATION USING ARBITRARY SURFACES

In order to also provide a possibility to verify the *stochastic model*, which is on the one hand independent of a geometric precision of the test object, as in Section 2.4.2 of the planarity of the panels, and on the other hand only considers the distance component  $\rho$  regardless of the other observations  $\theta$  and  $\phi$ , a further procedure based on repeated distance measurements  $\mathbf{l} = [\dots \rho_i \dots]^T$  is presented.

The solution expressed in a GMM, according to the calculation of a weighted mean is obtained straightforward by Equation (1.47) with the design matrix

$$\mathbf{A} = [1 \quad \dots \quad 1]^T \quad (2.29)$$

and the weight matrix

$$\mathbf{P} = \text{diag} \left[ \dots \quad \frac{1}{\sigma_{\rho_i}^2} \quad \dots \right], \quad (2.30)$$

where  $\sigma_0 = 1$  and the precisions  $\rho_i$  via the corresponding intensities are based on the proposed *stochastic model* according to Equation (2.2).

On the basis of the residuals according to Equation (1.42), the *empirical SDs*  $s_0$  were determined with Equation (1.80) using the weight matrix from (2.30) and presented in Figure 2.17 together with the  $s_0$  of the procedure described in Section 2.4.2. The  $s_0$  values from Section 2.4.2 are represented by triangles and the  $s_0$  values based on this section by circles, where the different colors indicate the sampling rate on which the data points are based. The two vertical green lines define the intensity range on which the experiments are based and indicate the bandwidth within the proposed *stochastic model* has been verified.

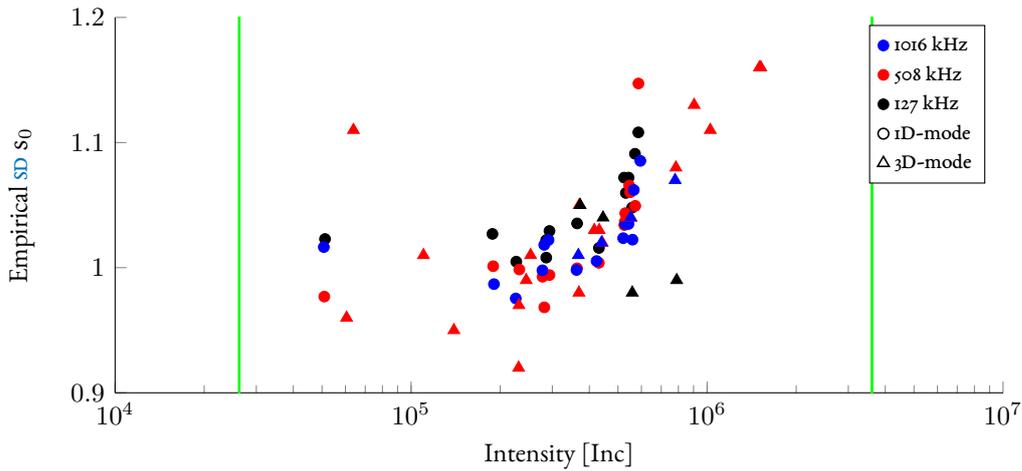


Figure 2.17: Intensity behaviour of the **TLs** Z+F IMAGER<sup>®</sup> 5006h as a function of the incidence angle  $\alpha$  at different distances..

IN SUMMARY all empirical standard deviations  $s_0$  with values between 0.92 and 1.16 fall within the desired interval of  $0.7 < s_0 < 1.3$ . In addition, the mean value of 1.03 indicates that the specific weights based on the *stochastic model* fit very well to the corresponding observations.

IT SHOULD BE NOTED THAT  $s_0$  increases slightly with increasing signal strength, which can be attributed to the fact that the precision of the distance measurement gradually coincides with the maximum resolution of the distance of 0.1 mm (ZOLLER+FRÖHLICH 2010).

CONCLUDING, IT CAN BE STATED THAT the proposed *stochastic models*, see Table 2.1, could be verified using both independent methods in this section.

## 2.5 DERIVING STOCHASTIC MODELS BASED ON 3D-MEASURING MODE

A major drawback of the method for deriving the stochastic model presented in Section 2.3 is the necessity of a repeatable measurement of a single distance  $\rho$ , requiring the beam deflection unit of the TLS to be switched off. The so-called 1D mode is only released by a few manufacturers, which is mainly due to eye safety issues. Therefore, this section focuses on the subject of deriving *stochastic models* even during normal operation of a TLS and is founded on the findings contained in the research contributions of (WUJANZ et al. 2018a) and (WUJANZ et al. 2018b).

To derive intensity-based stochastic models in normal operation, i. e. based on 3D *point clouds*, larger planar panels, such as in Figure 2.16, have to be used in contrast to the 1D mode, cf. Figure 2.1, thus ensuring sufficient evaluable information. This is due to the fact that in the 3D mode one has to rely on a planar analysis due to the missing correspondence and repeatability of single measurements. However, the basic measurement setup can be taken over from that in Figure 2.2.

In principle, two approaches to derive *stochastic models* in the 3D measurement mode can be distinguished:

- A. Interpretation of residuals as stochastics of distance measurement
- B. Derivation of EDMU stochastics based on "quasi-ranges"

Considering approach A, planes are estimated from the respective measurements on the panels, with  $i = 1, \dots, n$  points on the plane, cf. Section 2.4.2 and Equation (2.15), and the mean residual

$$\Delta p_m = \frac{\sum_{i=1}^n \Delta p_i}{n} \quad (2.31)$$

from the individual residuals

$$\Delta p_i = |n_x x_i + n_y y_i + n_z z_i - d|^{19} \quad (2.32)$$

of the points  $[x_i, y_i, z_i]$  to the adjusting plane is determined, serving as a representative quantity for the stochastic of the distance measurement –  $\sigma_\rho \hat{=} \Delta p_m$ .

REMARK: The general idea to use residuals from a plane adjustment as stochastic quantity for reflectorless measurements is not new (BÖHLER et al. 2003).

Furthermore, the average raw intensity  $I_m$  is determined for these measurements and compared to the corresponding calculated precision values  $\sigma_\rho$  for the distance. In Figure 2.18, the standard deviations  $\sigma_\rho$  of the distances determined at the same sampling rate of 508 kHz are plotted on the vertical axis, the raw mean intensity values  $I_m$  on the horizontal axis and compared to the stochastic reference model from Figure 2.14 (light green line) arising on the 1D mode.

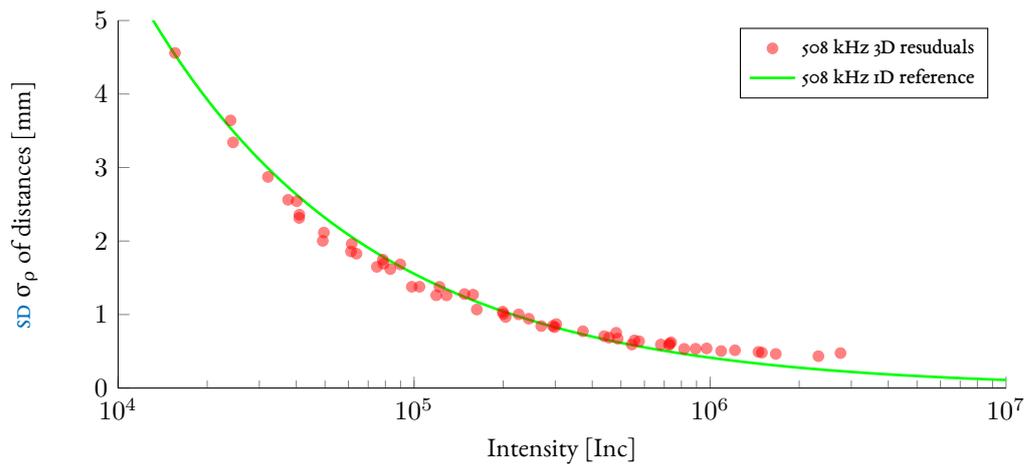


Figure 2.18: Comparison of the reference stochastic model based on an sampling rate of 508 kHz with 3D residual procedure.

As can be observed in Figure 2.18, the data points, depicted in red, calculated on the basis of the 3D mode follow the reference course (green line) almost completely, with minor differences in magnitude of up to 0.17 mm for higher intensity values.

<sup>19</sup>Since the residuals  $\Delta p_i$  of the plane adjustment depend on the orientation of the sample, the residuals may have to be corrected for this influence. This can be done simply by using the specified normal vector indicating the orientation of the sample.

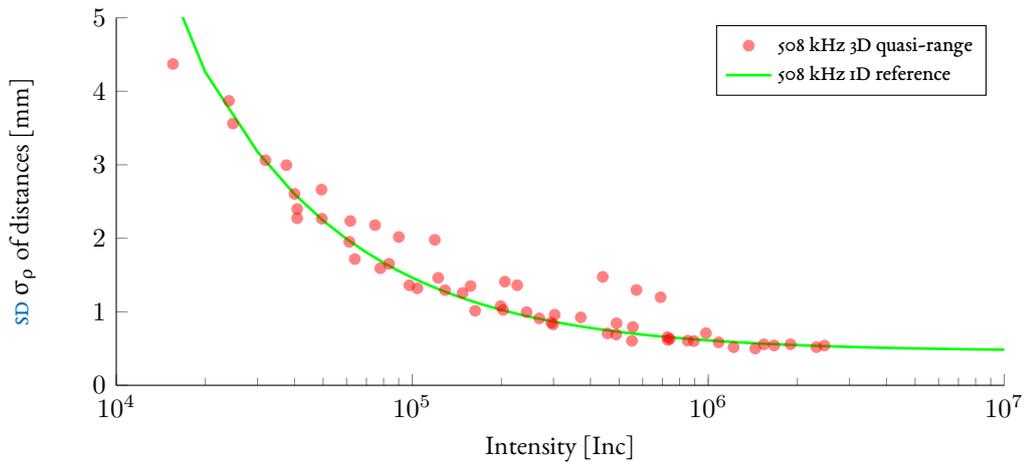


Figure 2.19: Comparison of the reference stochastic model based on an sampling rate of 508 kHz with "quasi range" procedure.

Another possibility to derive the stochastic model in 3D measurement mode is offered by variant B. Here the polar measurement values on the panel are determined from the *cartesian coordinates*, cf. Section A.1.2, and residuals to the mean measurement distance  $\bar{\rho}$  in a defined region are derived. Those residuals that lie below the distance resolution of the TLS applied, here 0.1 mm, are used to determine the measurement uncertainty.

Figure 2.19 shows the evaluation (red data points), comparable with Figure 2.18, for the stochastic reference model (green line). Again, only minor differences at higher intensities can be observed, but the course of the calculated data points is more noisy compared to Figure 2.18.

IT CAN BE STATED THAT the derivation of EDMU stochastics based on "quasi ranges" (method B) is obviously very sensitive to the chosen acquisition configuration, hence method A is preferred.

## 2.6 VALIDITY FOR PULSED TLS

The focus of the previous derivation of *stochastic models* was on TLSs that operate with the *phase comparison method*, cf. also Section 1.2.2, see Section 2.3 and Section 2.5. Nevertheless, in this section we will discuss whether the method can be transferred to TLSs that operate with the pulse propagation method and is based on the findings in (WUJANZ et al. 2018a) and (WUJANZ et al. 2018b).

For this purpose, a scanner from this category, the Riegl VZ-400i, was used and an intensity-based stochastic model according to Section 2.5, variant A has been derived, see Figure 2.20.

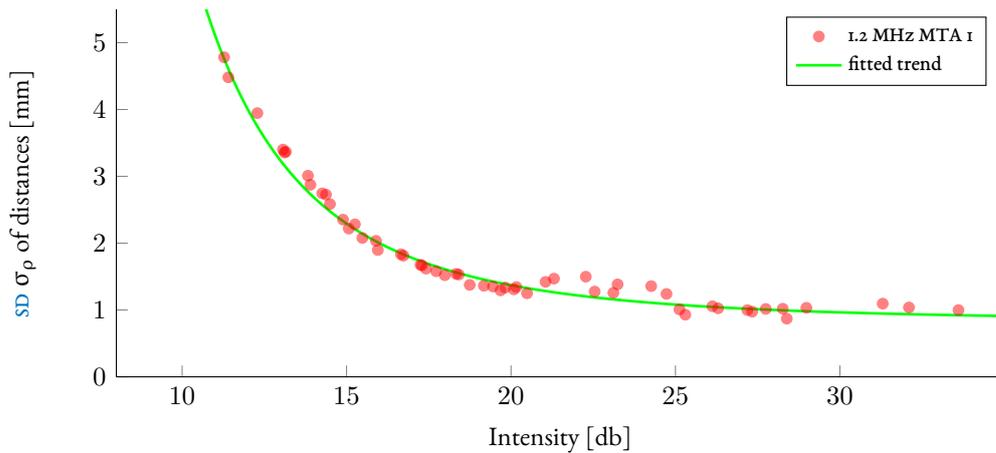


Figure 2.20: Stochastic model of the MTA I zone of a Riegl VZ-400i laser scanner.

Analogous to [Section 2.3](#), a stochastic model can be derived, which was extended by one parameter  $c$ , cf. [Equation \(2.2\)](#), due to the different intensity behavior of the diode.

## 2.7 CONCLUSION

Referring to [Section 1.9](#) existing efforts to derive *stochastic models* were elaborated, the importance and requirements of those models were emphasized, consequently the necessity of a new all-encompassing *stochastic model* was pointed out.

Subsequently, the intensity behavior of the [APD](#) employed was evaluated using a suitable measuring arrangement, and associated with the precision of the distance measurement in [Section 2.1](#).

Further distance and radiometry related factors influencing the stochastics of the distance measurement, such as the angle of incidence or different material properties, were investigated in [Section 2.2](#), showing that the *stochastic model* is capable of covering all effects.

Finally, the characteristic trend in [Figure 2.14](#) has been described with a functional model, see [Equation \(2.2\)](#), and the parameters of the stochastic model have been determined, whereupon the validation in [Section 2.4](#) shows the usability.

In [Section 2.5](#), however, the applicability of the derivation of stochastic models in [3D](#) measurement mode was demonstrated and in [Section 2.6](#) for scanners that operate pulse-based.

IN SUMMARY IT CAN BE CONCLUDED that a giant step towards an all-embracing *stochastic model*, applicable for different *laser scanner* types, could be developed, which maps all influencing factors on the distance measurement in one model and considers them completely.

The new *stochastic model* is a basic module for processing the measured values in [Chapters 3](#) and [4](#) and is already being used or discussed also by other scientists, see [Section 2.7.1](#).

#### 2.7.1 ACCEPTANCE AND FIELDS OF APPLICATION

Meanwhile, the intensity-based stochastic model has gained wide acceptance and is used for a variety of scientific research questions, as listed below:

- [44] E. HEINZ, M. METTENLEITER, H. KUHLMANN, and C. HOLST: “Strategy for determining the stochastic distance characteristics of the 2D Laser Scanner Z+ F Profiler 9012A with special focus on the close range”. In: *Sensors* 18.7 (2018), p. 2253.
- [56] G. KERÉKES and V. SCHWIEGER: “Elementary error model applied to terrestrial laser scanning measurements: study case arch dam Kops”. In: *Mathematics* 8.4 (2020), p. 593.
- [57] G. KERMARREC: “On Estimating the Hurst Parameter from Least-Squares Residuals. Case Study: Correlated Terrestrial Laser Scanner Range Noise”. In: *Mathematics* 8.5 (2020), p. 674.
- [58] G. KERMARREC, H. ALKHATIB, and I. NEUMANN: “On the sensitivity of the parameters of the intensity-based stochastic model for terrestrial laser scanner. Case study: B-spline approximation”. In: *Sensors* 18.9 (2018), p. 2964.
- [59] G. KERMARREC, M. LÖSLER, and J. HARTMANN: “Analysis of the temporal correlations of TLS range observations from plane fitting residuals”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 171 (2021), pp. 119–132.
- [60] G. KERMARREC, I. NEUMANN, H. ALKHATIB, and S. SCHÖN: “The stochastic model for Global Navigation Satellite Systems and terrestrial laser scanning observations: A proposal to account for correlations in least squares adjustment”. In: *Journal of Applied Geodesy* 13.2 (2019), pp. 93–104.
- [107] B. SCHMITZ, C. HOLST, T. MEDIC, D. D. LICHTI, and H. KUHLMANN: “How to Efficiently Determine the Range Precision of 3D Terrestrial Laser Scanners”. In: *Sensors* 19.6 (2019), p. 1466.
- [138] X. ZHAO, G. KERMARREC, B. KARGOLL, H. ALKHATIB, and I. NEUMANN: “Influence of the simplified stochastic model of TLS measurements on geometry-based deformation analysis”. In: *Journal of Applied Geodesy* 13.3 (2019), pp. 199–214.

# 3

## Segmentation based on Geometric and Radiometric Information

THE SEGMENTATION BELONGS, apart from the registration to the most important operational steps contributing to the automation regarding the subsequent processing of point clouds, see also [Section 0.1](#). From [Section 1.10](#) it emerges that a variety of different segmentation methods for *point clouds* captured with a [TLS](#) already exist. However, most of these methods consider *point clouds* from different standpoints already transformed into a *superordinate coordinate system* (registered, see also [Section 1.11](#)) and are therefore dependent on a recovery of the topology by the prior construction of suitable 3D data structures, see [Section 1.4](#). In contrast, if one assumes a single [TLS](#) position, the data is available as a grid (in polar representation, see especially [Figure 1.3](#)) due to the line- and column-wise operating principle of the *laser scanner*. The known topology, as in an image matrix, enables very efficient neighborhood analyses with already proven image processing routines, from which geometric as well as radiometric information can be obtained for the segmentation of a single scan.

First of all, [Section 3.2](#) describes the pre-processing steps based on the data of the test objects presented in [Section 3.1](#), providing the information required for segmentation. On this basis, the new procedure for segmenting [TLS point clouds](#) is presented in [Section 3.3](#). An illustration and more detailed analysis of further results of the procedure presented, followed by a conclusion in [Section 3.5](#), is given in [Section 3.4](#).

### 3.1 SAMPLE DATASETS

The new segmentation algorithm shall be explained in the following by means of processing a sample data set and validated by more detailed analyses with the use of two further data sets. The views of the test objects

- A teahouse in the park of the Charlottenburg Palace ("Belvedere"),
- B inner dome of the outdoor facility of the Orangery in Potsdam,
- C landslide in Obergurgel

captured with [TLSs](#) and upon whom the data sets are based are depicted in [Figure 3.1](#).



Figure 3.1: Front view of the test objects Belvedere (left), Orangery (center) and landslide (right).

[Figure 3.1A](#) shows the front view of the 24 m high, 17 m wide and 13 m deep tea house "Belvedere" constructed in the late Baroque and the position of the [TLS Z+F IMAGER<sup>®</sup> 5006h](#) with equipment from which the building was scanned. The data set recorded from this perspective serves as the basis for the representation of the segmentation algorithm in the following [Sections 3.2](#) to [3.3](#).

The individual geometry layers of the data set related to the scan position shown in [Figure 3.1A](#) are depicted color-coded in [Figure 3.2](#) according to their coordinate values. The range of values extends from -15 (blue) to 15 m (red), whereas the value passage of 0 m was assigned white.

Due to the incremental scanning of a [TLS](#) in predefined angular increments, see [Figure 1.3](#), the individual measured values refer to the respective scan line and column, so that the measured values are directly available in such a raster structure.

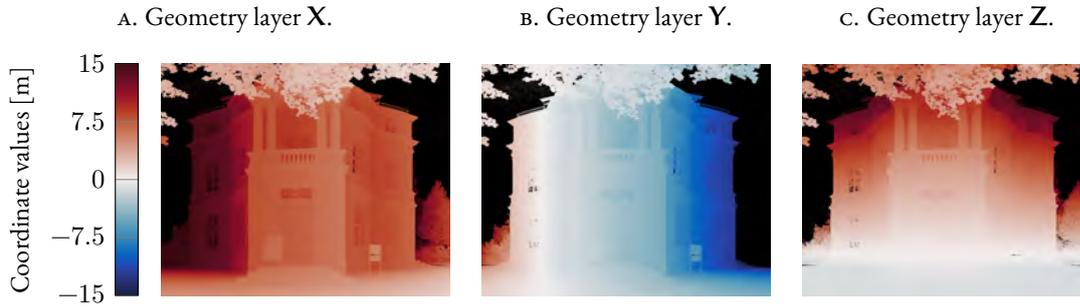


Figure 3.2: Geometry layers **X**, **Y** and **Z** related to the standpoint in [Figure 3.1A](#).

However, during data export the values are stored in various formats, e. g. the proprietary ptx-format in a list. With the known number of scan columns  $c$  and lines  $r$ , the raster structure (matrix) is reproducible again. [Code 3.1](#) shows the `function` `rearrange_2_matrix` which performs the conversion of a point cloud as a list `p_cloud_l` into a matrix structure `p_cloud`.

Code 3.1: Conversion of `TLS` data based on polar acquisition in a matrix structure.

```

1 function p_cloud = rearrange_2_matrix(p_cloud_l)
2 %input: struct lists p_cloud_l.p3D and p_cloud_l.int
3 %output: struct matrices p_cloud.x,.y,.z and p_cloud.int
4
5 % known scan rows r and columns c of scanned point cloud
6 r = p_cloud_l.size(1,1); c = p_cloud_l.size(1,2);
7
8 p_cloud = struct('x',{[]},'y',{[]},'z',{[]},'int',{[]});
9 for i = 1:r
10     p_cloud.x(i,:) = p_cloud_l.p3D(((i-1)*c+1):(i*c),1);
11     p_cloud.y(i,:) = p_cloud_l.p3D(((i-1)*c+1):(i*c),2);
12     p_cloud.z(i,:) = p_cloud_l.p3D(((i-1)*c+1):(i*c),3);
13     p_cloud.int(i,:) = p_cloud_l.int(((i-1)*c+1):(i*c),1);
14 end

```

The intensity information **I** corresponding to the geometry, see [Figure 3.2](#), is color-coded in [Figure 3.3A](#), where the decrease in intensity from object near areas in the center (red) to farther away and with a larger angle of incidence to the right and left edge regions (yellow via green to blue) is obvious, see also [Section 1.2.4.1](#).

On the basis of the raw intensity values, the new *stochastic model* presented in [Chapter 2](#), see [Equation \(2.2\)](#), can be used to calculate the precision for each individual **3D** point according to [Section 2.3.1](#) with [Equation \(2.13\)](#). The resulting precision layer  $\Sigma_{XYZ}$  with the color bar ranging from blue (<1 mm) to red (>8 mm) is shown in [Figure 3.3B](#) to the right of the intensity layer [Figure 3.3A](#). It can be

clearly seen that the point precision is inversely proportional to the strength of the intensity, reflecting the behaviour of the EDMU observed in Section 2.1.

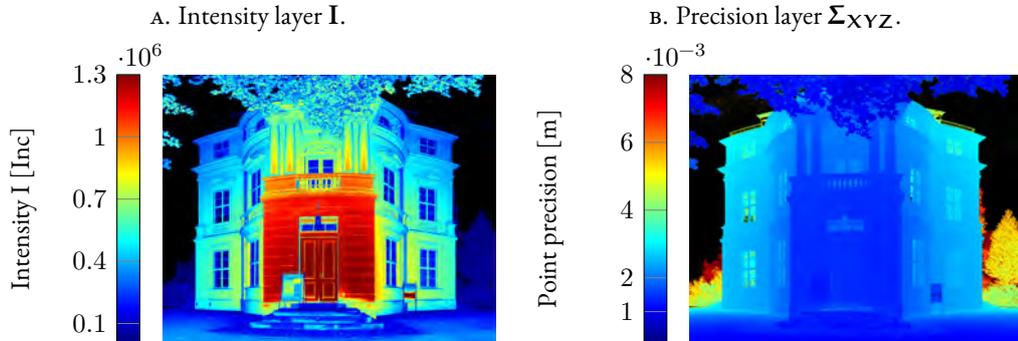


Figure 3.3: Intensity layer  $I$  and corresponding precision layer  $\Sigma_{XYZ}$  related to the standpoint in Figure 3.1A.

For the evaluation of the segmentation results, the so-called "egyptian basket arch portal" of the orangery in the new garden of Potsdam, Figure 3.1B, was also scanned with the TLS Z+F IMAGER<sup>®</sup> 5006h, resulting in another data set. Due to their different geometric elements (pillars, planes, curved surfaces) and the complexity of their composition, the two data sets of the buildings, see Figures 3.1A to 3.1B, are particularly suitable for evaluating a segmentation algorithm.

In order to also analyze the applicability of the strategy pursued by the segmentation algorithm with regard to natural objects, a third data set, see Figure 3.1C, was captured<sup>1</sup> with the new TLS Z+F IMAGER<sup>®</sup> 5016 of a landslide in Obergurgel.

### 3.2 DATA PREPROCESSING FOR INFORMATION RETRIEVAL

For the subsequent segmentation, the informations contained in the data layers in Figures 3.2 to 3.3 must be preprocessed. In particular, point normals that represent the surface condition of the object in a local environment are suitable for deriving geometric criteria. Radiometric information in the form of intensity values can additionally contribute to a differentiated separation of objects in the case of a more homogeneous surface geometry. With regard to an additional classification into individual object classes, cf. Figure 2.10, intensity values also take on an important role.

#### 3.2.1 LOCAL NEIGHBOURHOOD ANALYSIS AND CONVOLUTION

Efficient neighbourhood analyses are needed to derive such criteria in local areas. With the mathematical formulation of a raster or matrix structure as a function  $f(x, y)$  of two variables  $x$  and  $y$ , indicating

<sup>1</sup> The TLS measurements were carried out by Daniel Wujanz.

the position of a value  $f(x, y)$  within the structure, a third function

$$g(x, y) = f(x, y) \star h(x, y) = \int_{x'=-\infty}^{+\infty} \int_{y'=-\infty}^{+\infty} f(x', y') h(x - x', y - y') dx' dy' \quad (3.1)$$

is obtained by means of the so-called *convolution* with a function  $h(x, y)$ . The product  $f(x, y) \star h(x, y)$ <sup>2</sup> in (3.1) of the two functions with discrete domains  $D_1 \subseteq \mathbb{Z}$  and  $D_2 \subseteq \mathbb{Z}$  leads to *discrete convolution*<sup>3</sup>:

$$g(x, y) = f(x, y) \star h(x, y) = \sum_{x' \in D_1} \sum_{y' \in D_2} f(x', y') h(x - x', y - y') \quad (3.2)$$

Local linear image processing operations can be realized by a *discrete convolution*<sup>4</sup> according to (3.2) with the substitutions  $f = \mathbf{K}$  and  $h = \mathbf{M}$

$$\mathbf{M}^*(i, j) = \sum_{p=1}^w \sum_{q=1}^w \mathbf{M}(i - p + k_c, j - q + k_c) \mathbf{K}(p, q), \quad (3.3)$$

where

- $\mathbf{M}$  is the original matrix to be convolved
- $\mathbf{K}$  is the quadratic convolution matrix or kernel<sup>5</sup> of size  $w \times w$  and center position  $k_c = \frac{w}{2} + 0.5$
- $\mathbf{M}^*(i, j)$  is the resulting pixel or value at the position  $(i, j)$  in the matrix  $\mathbf{M}^*$

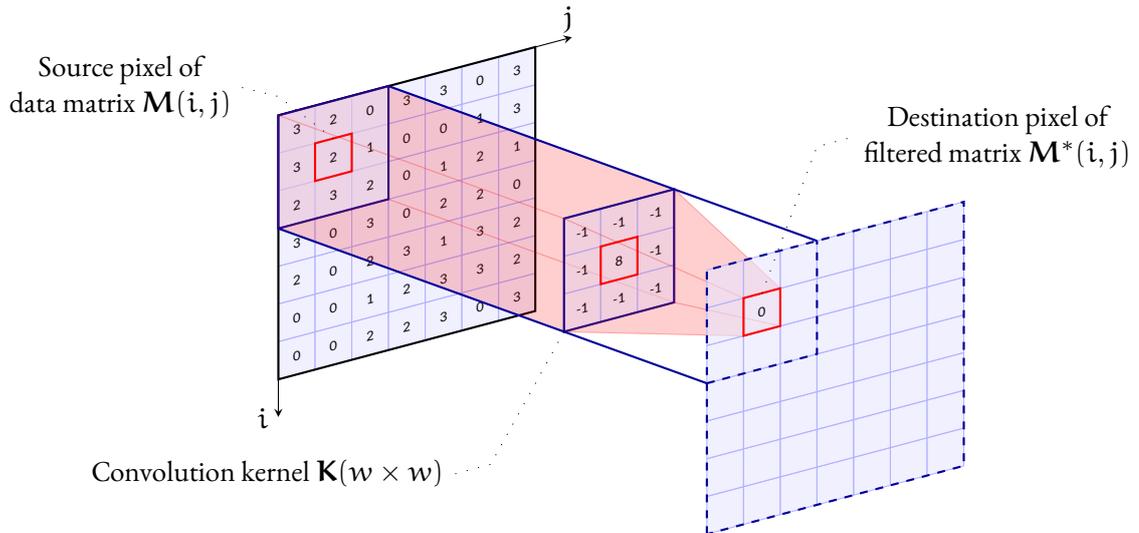
The convolution according to (3.3) describes the successive shift of the kernel  $\mathbf{K}$  to each position  $(i, j)$  of the matrix  $\mathbf{M}$ , see Figure 3.4, whereby the horizontally and vertically flipped kernel values are multiplied by the corresponding matrix values within the mask and summed up in order to store them at the position  $(i, j)$  in the filtered matrix  $\mathbf{M}^*$  weighted by the kernel elements.

<sup>2</sup> The arithmetic operator  $\star$  symbolically represents the convolution operation

<sup>3</sup> The convolution of two functions can also be performed with the so called *convolution theorem*  $f(x, y) \star h(x, y) = \mathcal{F}^{-1} \mathcal{F} f(x, y) \star \mathcal{F} h(x, y)$  in frequency domain, where  $\mathcal{F}$  is denoted by the Fourier Transformation.

<sup>4</sup> The convolution can be expressed by  $\mathbf{M}^*(i, j) = \sum_{p=1}^w \sum_{q=1}^w \mathbf{M}(i + p - k_c, j + q - k_c) \mathbf{K}(p, q)$  as a linear filtering and subsequently used in this sense. For symmetrical filter matrices  $\mathbf{K}$ , the convolution is equivalent to filtering.

<sup>5</sup> The convolution matrix is often referred to as a filter operator, filter mask or kernel in the context of filtering.


 Figure 3.4: Convolution of a data matrix  $\mathbf{M}$  with a kernel  $\mathbf{K}$ .

The process of *discrete convolution* based on Equation (3.3) and its graphical representation in Figure 3.4, can be described algorithmically in the case of filtering with the following Algorithm 3.1:

Algorithm 3.1: Filtering a matrix by a kernel.

---

**input :** A matrix  $\mathbf{M}$  of size  $r \times c$  and a filter kernel  $\mathbf{K}$  of size  $w \times w$   
**output :** A filtered matrix  $\mathbf{M}^*$  of same size

```

for  $i \leftarrow k_c$  to  $r - 1 + c$  do
    for  $j \leftarrow k_c$  to  $c - 1 + c$  do
         $M_{ij} \leftarrow 0$ 
        for  $p \leftarrow 1$  to  $w$  do
            for  $q \leftarrow 1$  to  $w$  do
                 $M_{pq} \leftarrow \mathbf{M}[i + p - k_c, j + q - k_c] \mathbf{K}[p, q]$ 
                 $M_{ij} \leftarrow M_{ij} + M_{pq}$ 
            end
        end
         $\mathbf{M}^*[i, j] \leftarrow M_{ij}$ 
    end
end
return  $\mathbf{M}^*$ 
    
```

---

## 3.2.2 DESCRIBING LOCAL CHANGES WITHIN THE DATA

The derivation of local changes in the data is a prerequisite for the pixel-based segmentation in [Section 3.3.1](#). The extent of the scatter of measured values can be easily expressed by the Standard Deviation ([SD](#)). This idea can be transferred to the values within a filter mask  $\mathbf{K}$ . Considering a  $3 \times 3$  filter mask applied to a matrix  $\mathbf{M}$ , see [Figure 3.4](#), with its neighborhood values, see [Figure 3.5](#), the [SD](#)

$$\sigma_{M_{i,j}^*} = \sqrt{\frac{\sum_{p=1}^w \sum_{q=1}^w (\mathbf{M}(i+p-k_c, j+q-k_c) - \bar{M})^2}{n_k - 1}} \quad (3.4)$$

can be given to describe the mean change of the values within the mask, where  $w = 3$  and consequently  $c = 2$  and the number of kernel elements is  $n_k = 9$ .

$M_{j-1}^{i-1}$	$M_{j-1}^i$	$M_{j-1}^{i+1}$
$M_{j-1}^i$	$M_j^i$	$M_{j+1}^i$
$M_{j-1}^{i+1}$	$M_{j+1}^{i+1}$	$M_{j+1}^{i+1}$

Figure 3.5: Kernel  $\mathbf{K}$  with size  $3 \times 3$  at position  $(i, j)$  in matrix  $\mathbf{M}$ .

Using the so-called *Steiner's displacement theorem*<sup>6</sup>, see e. g. ([CHAN et al. 1983](#)), the expression of the empirical variance beneath the square root in [\(3.4\)](#) with respect to the mean  $\bar{M}$  can be reformulated to

$$\sigma^2_{M_{i,j}^*} = \frac{\sum_{p=1}^w \sum_{q=1}^w \mathbf{M}^2(i+p-k_c, j+q-k_c) - n_k \bar{M}^2}{n_k - 1}. \quad (3.5)$$

Substituting the mean value  $\bar{M} = \frac{\sum_{p=1}^w \sum_{q=1}^w \mathbf{M}(i+p-k_c, j+q-k_c)}{n_k}$  for [\(3.5\)](#) and reformulating yields

$$\begin{aligned} \sigma^2_{M_{i,j}^*} &= \frac{1}{n_k - 1} \sum_{p=1}^w \sum_{q=1}^w \mathbf{M}^2(i+p-k_c, j+q-k_c) \\ &\quad - \frac{1}{n_k(n_k - 1)} \left( \sum_{p=1}^w \sum_{q=1}^w \mathbf{M}(i+p-k_c, j+q-k_c) \right)^2. \end{aligned} \quad (3.6)$$

Using the calculation expression for the variance of values  $\sigma^2_{M_{i,j}^*}$  within a kernel in [\(3.6\)](#), the local calculation of the [SD](#) with respect to a kernel  $\mathbf{K}$  applied to an entire matrix  $\mathbf{M}$  can be described by means of *convolution*, see [Equation \(3.3\)](#).

<sup>6</sup> Also known as Huygens theorem in literature and has a second relation to mechanics.

The `function` `calc_change_in_region`<sup>7</sup>, see [Code 3.2](#), calculates the local standard deviations by using two separate convolutions applying an average filter

$$\mathbf{K} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad (3.7)$$

here with an edge length of  $w = 3$ . A first convolution is performed with a weighting of the filter elements by  $\frac{1}{n_k - 1} = \frac{1}{8}$  on the squared elements in the matrix  $\mathbf{M}^2$  (corresponds to the first part of (3.6)). The second convolution is applied to the original matrix  $\mathbf{M}$  and then the result is squared and weighted with  $\frac{1}{n_k(n_k - 1)} = \frac{1}{72}$  (corresponds to the second part of (3.6)). The root of the difference between the two convolution matrices (`conv1` and `conv2` in [Code 3.2](#)), represented by `M_filt`, finally contains the local SDs.

Code 3.2: Region-based calculation of standard deviations for a matrix  $\mathbf{M}$  using an modified convolution approach.

```

1 function M_filt = calc_change_in_region(M, wK)
2 %input: matrix M and kernel width wK
3 %output: filtered matrix M_filt with standard deviations
4
5 % build kernel
6 K = ones(wK);
7
8 % Number of Kernel elements
9 nK = wK^2; nK1 = nK-1;
10
11 % Expand M that interpolation is valid at the boundaries
12 [M, rows, cols] = expand_borders(M, wK);
13
14 %filter: rewritten convolution for computing standard deviation
15 conv1 = filter2(K/nK1,M.^2);
16 conv2 = filter2(K,M).^2/(nK*nK1);
17
18 M_filt = sqrt(conv1-conv2); M_filt = M_filt(rows,cols);

```

By applying an average filter, as in [Equation \(3.7\)](#), a slight smoothing<sup>8</sup> of the data takes place at the same time, which is advantageous for the region-based approach in [Section 3.3.2](#) unlike edge-based methods.

<sup>7</sup> Uses `function` `expand_borders` in [Section A.3](#), see [Code A.2](#)

<sup>8</sup> The degree of smoothing depends on the size of the filter mask  $w$ , ideally defined by the point spacing on the object, cf. [Equation \(3.15\)](#), and increases with increasing mask size.

NOTE that the average filter in Equation (3.7) can be separated, which means that it can be represented by multiplying two single one-dimensional operators:

$$\mathbf{K} = \mathbf{k}_1 \mathbf{k}_2 = \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.8)$$

For the convolution process this implies that in the case of separability, the two-dimensional convolution can be reduced to two one-dimensional operations by applying the second operator to the intermediate result of the first.

REMARK: The application of a 2D filter of size  $w \times w$  requires for each position  $(i, j)$   $w^2$  read accesses and multiplications, as well as  $w^2 - 1$  additions. In a separated application, the computational effort is reduced to only  $2w$  read accesses and multiplications, as well as  $2(w - 1)$  additions, resulting in a significant saving of computation time.

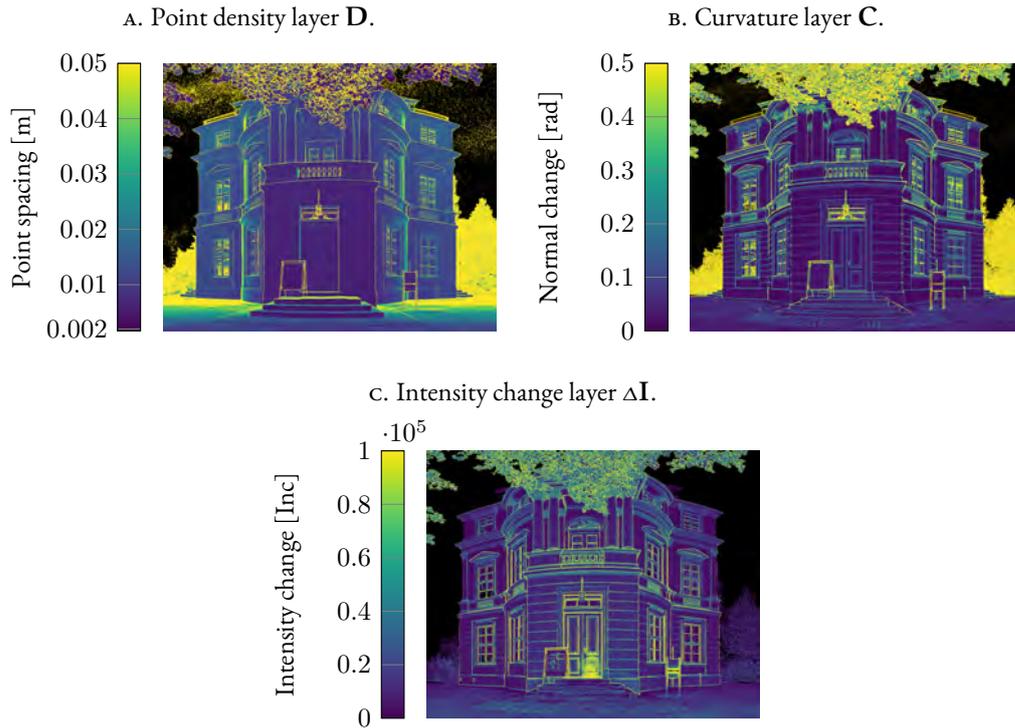


Figure 3.6: Computed information layers regarding curvature, point spacing and change in intensity.

Transferring the previously described principle for the local calculation of changes in the data to the intensity layer **I**, see [Figure 3.3A](#), [Code 3.2](#) with the use of `I_filt=calc_change_in_region(I,3)` can be carried out to calculate the matrix with the SDs of the intensities  $\Delta\mathbf{I}=\mathbf{I\_filt}$ , see [Figure 3.6C](#). These can be interpreted as an average measure of the intensity change within the mask used for the calculation.

NOTE: For the segmentation algorithm in [Section 3.3](#), it is not mandatory to perform a radiometric calibration of the intensity values in advance by means of a compensation for its influencing factors, see also [Section 2.1.2](#), as described in ([ERIC and GÖRING 2017](#)), since only changes are of relevance.

Within a mask, see [Figure 3.5](#), adjacent points do not necessarily have to be neighboured in the 3D due to the incremental scanning, see [Figure 1.3](#). Calculating the changes  $\Delta\mathbf{X}$ ,  $\Delta\mathbf{Y}$  and  $\Delta\mathbf{Z}$  in analogy to the intensity change  $\Delta\mathbf{I}$  by applying the `function calc_change_in_region` for each geometry layer in [Figure 3.2](#), i. e. for  $\Delta\mathbf{X}=\mathbf{x\_filt}$  and  $\mathbf{X}=\mathbf{x}$  using `x_filt=calc_change_in_region(x,3)` with a mask size  $w = 3$  indicating a direct neighbourhood for the calculation, cf. [Figure 3.5](#), the layer  $\mathbf{D} = \sqrt{\Delta\mathbf{X}^2 + \Delta\mathbf{Y}^2 + \Delta\mathbf{Z}^2}$  that represents the spatial point change can be calculated with `function calc_point_density`, see [Code A.5](#). The layer **D**, see [Figure 3.6A](#), is a measure for the mean point distance (mean point spacing) within the used mask and can be applied as an additional geometry criterion in [Section 3.3](#) to exclude "spatially wrong"<sup>9</sup> neighbours during segmentation.

Again, the local changes  $\Delta\mathbf{n}_x$ ,  $\Delta\mathbf{n}_y$  and  $\Delta\mathbf{n}_z$  of the previously calculated layers  $\mathbf{n}_x$ ,  $\mathbf{n}_y$  and  $\mathbf{n}_z$  of the normal components, see [Figure 3.8](#), from [Section 3.2.3](#) can be determined equivalent to the calculation of the density layer **D**, see above, with the same `function calc_change_in_region`. The local change of the point normals, which can also be interpreted as curvature of the surface, results in layer  $\mathbf{C} = \sqrt{\Delta\mathbf{n}_x^2 + \Delta\mathbf{n}_y^2 + \Delta\mathbf{n}_z^2}$ , see [Figure 3.6B](#), by applying `function calc_point_curvature`, see [Code A.4](#). The information in layer **C** is the primary input for the geometric part of the segmentation algorithm in [Section 3.3](#), since it includes the change of the geometry of the object.

### 3.2.3 EFFICIENT CALCULATION OF POINT NORMALS

The efficient and precise<sup>10</sup> calculation of point normals, describing the surface geometry in a small neighborhood, is indispensable for segmentation on a geometric basis, particularly as they serve as

<sup>9</sup> Means that a 3D point could be neighboured in the matrix data structure in terms of the neighbourhood definition, see also [Figure 3.5](#), but spatially it may lie on another surface and therefore its distance to the relating neighbouring point exceeds the usual point spacing in a certain area, i. e. within the kernel mask.

<sup>10</sup> Precise in this context means that the point normals are determined with minimum variation in the geometry, i. e. variation in the normal vector components.

input information for the calculation of the curvature, see [Section 3.2.2](#), which constitutes one of three basic sets of information for segmentation.

An efficient and common calculation method of the normal direction in a point can be realized via the two direction vectors to the respective point neighbors. Considering the direct neighbours, see also [Figure 3.5](#), of a certain position  $(i, j)$  in the respective geometry layers, see [Figure 3.7](#), the direction vectors

$$\mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \mathbf{X}(i, j-1) - \mathbf{X}(i, j+1) \\ \mathbf{Y}(i, j-1) - \mathbf{Y}(i, j+1) \\ \mathbf{Z}(i, j-1) - \mathbf{Z}(i, j+1) \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \begin{bmatrix} \mathbf{X}(i-1, j) - \mathbf{X}(i+1, j) \\ \mathbf{Y}(i-1, j) - \mathbf{Y}(i+1, j) \\ \mathbf{Z}(i-1, j) - \mathbf{Z}(i+1, j) \end{bmatrix} \quad (3.9)$$

can be calculated via the coordinate differences in row ( $i$ ) and column ( $j$ ) direction.

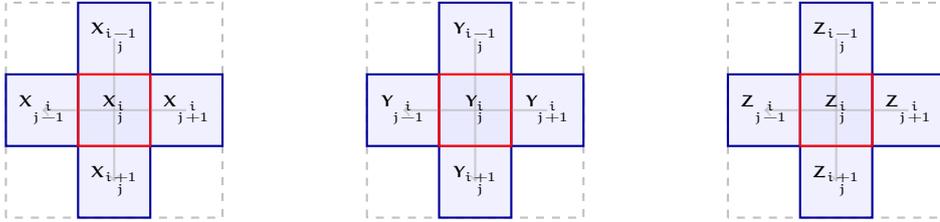


Figure 3.7: Calculation of point normals within the matrix structure of coordinates  $\mathbf{X}$ ,  $\mathbf{Y}$  and  $\mathbf{Z}$ .

Forming the cross product of these two vectors, a vector

$$\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \times \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \begin{bmatrix} -(a_y b_z - a_z b_y) \\ -(a_z b_x - a_x b_z) \\ -(a_x b_y - a_y b_x) \end{bmatrix} \quad (3.10)$$

is obtained that is perpendicular to the plane spanned by the vectors. It represents the geometry in the local neighborhood and can be understood as surface normal in point  $(i, j)$ .

The calculation for an entire data set  $\mathbf{M}$  in raster format can ideally be implemented using two filtering procedures, cf. also [Equation \(3.8\)](#). The differences of the neighborhood values can be determined by the gradients

$$\nabla_j = \frac{\partial \mathbf{M}(i, j)}{\partial j} = \frac{1}{2} (\mathbf{M}(i, j-1) - \mathbf{M}(i, j+1)) = \frac{1}{2} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad (3.11a)$$

$$\nabla_i = \frac{\partial \mathbf{M}(i, j)}{\partial i} = \frac{1}{2} (\mathbf{M}(i+1, j) - \mathbf{M}(i-1, j)) = \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T, \quad (3.11b)$$

corresponding to the first derivative, in column and row direction of the matrix  $\mathbf{M}$ , from which the one-dimensional filter masks are derived.

By applying the kernel  $\mathbf{K}_r = \frac{1}{2}[1 \ 0 \ -1]$ , cf. (3.11a), to each coordinate layer  $\mathbf{X}$ ,  $\mathbf{Y}$  and  $\mathbf{Z}$  in Figure 3.2 by means of filtering, see Section 3.2.1, the three matrices

$$\begin{aligned} \mathbf{a}_x &= \mathbf{K}_r \star \mathbf{X} \\ \mathbf{a}_y &= \mathbf{K}_r \star \mathbf{Y} \\ \mathbf{a}_z &= \mathbf{K}_r \star \mathbf{Z} \end{aligned} \quad (3.12)$$

with the vector components in column direction are obtained as a result. Again, by applying the kernel  $\mathbf{K}_c = \frac{1}{2}[-1 \ 0 \ 1]^T$ , cf. (3.11b), the matrices

$$\begin{aligned} \mathbf{b}_x &= \mathbf{K}_c \star \mathbf{X} \\ \mathbf{b}_y &= \mathbf{K}_c \star \mathbf{Y} \\ \mathbf{b}_z &= \mathbf{K}_c \star \mathbf{Z} \end{aligned} \quad (3.13)$$

with the vector components in row direction are obtained in analogy. With the element by element calculation of the matrices in (3.12) and (3.13) according to (3.10)

$$\begin{bmatrix} \mathbf{n}_x(i, j) \\ \mathbf{n}_y(i, j) \\ \mathbf{n}_z(i, j) \end{bmatrix} = \begin{bmatrix} \mathbf{a}_x(i, j) \\ \mathbf{a}_y(i, j) \\ \mathbf{a}_z(i, j) \end{bmatrix} \times \begin{bmatrix} \mathbf{b}_x(i, j) \\ \mathbf{b}_y(i, j) \\ \mathbf{b}_z(i, j) \end{bmatrix}, \quad (3.14)$$

the matrices  $\mathbf{n}_x$ ,  $\mathbf{n}_y$  and  $\mathbf{n}_z$  result with its corresponding normal vector components.

According to this approach, `function calc_point_normals`, see Code A.3 calculates the layers of the normal vector components, see Figure 3.8, depending on the kernel size `wk` by applying the respective filter masks `mask_r` and `mask_c`.

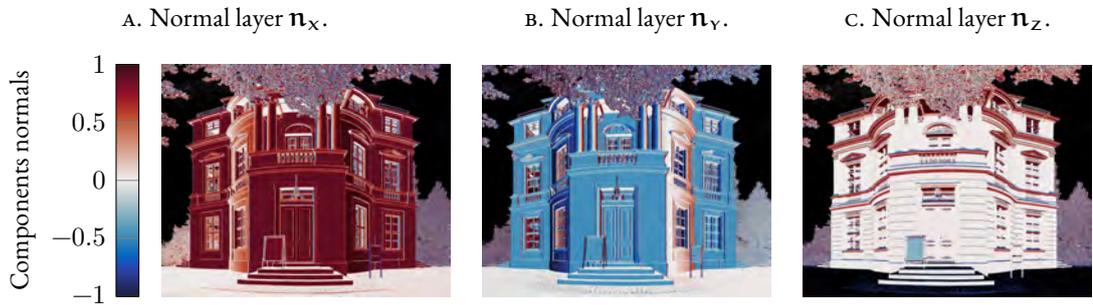


Figure 3.8: Computed layers regarding the normal vector components  $\mathbf{n}_x$ ,  $\mathbf{n}_y$  and  $\mathbf{n}_z$ .

Due to the incremental scanning of the TLS, see Figure 1.3, and changes in the object geometry, different point spacings result, see Figure 3.6A. When using the filter masks  $\mathbf{K}_r$  and  $\mathbf{K}_c$  with a length  $w = 3$  – termed “uniform filter mask” in the following – to calculate the point normals according to Code A.3, different precisions for the calculated normals result for different areas with varying point spacings. This is reflected in higher noise after the calculation of the curvature information in areas where the points are more dense. This is due to the fact that in regions with a small point spacing, the direction vectors, see Equation (3.9), for calculating the normals span a smaller area, causing more geometrically instable normals. In order to take this circumstance into account, the size of the filter mask

$$w = \begin{cases} 3, & \text{for } \mathbf{D}(i, j) \geq 0.02 \\ 5, & \text{for } 0.01 \geq \mathbf{D}(i, j) < 0.02 \\ 7, & \text{for } 0.005 \geq \mathbf{D}(i, j) < 0.01 \\ 9, & \text{for } \mathbf{D}(i, j) < 0.005 \end{cases} \quad (3.15)$$

can be selected as a function of the local point spacing – termed “adapted filter mask” in the following – in  $\mathbf{D}(i, j)$ , see Figure 3.6A.

function `calc_point_normals_dii`, see Code 3.3, follows this principle by determining the mask positions `pos_mask` in the `switch - case` statement (lines 13 to 25) for different ranges of point spacings based on the information in layer  $\mathbf{D}$ , see Figure 3.6A, (`p_cloud.d` in Code 3.3) and applying the appropriate filter masks to them.

Code 3.3: Efficient point normal calculation by means of filtering under consideration of the point spacing.

```

1 function p_cloud = calc_point_normals_d(p_cloud)
2 %input: struct p_cloud.x, p_cloud.y, p_cloud.z
3 %output: struct p_cloud.nx, p_cloud.ny, p_cloud.nz
4
5 [rows,cols] = size(p_cloud.x);
6
7 normals_x = zeros(rows*cols,1);
8 normals_y = zeros(rows*cols,1);
9 normals_z = zeros(rows*cols,1);
10
11 % choose individual mask size depending on point spacing
12 for i = 1:4
13     switch i
14         case 1
15             wK = 3;

```

<sup>ii</sup> Uses function `expand_borders`, see Code A.2, and `calc_point_normals`, see Code A.3 in Section A.3

## SEGMENTATION BASED ON GEOMETRIC AND RADIOMETRIC INFORMATION

```
16         pos_mask = find(p_cloud.d>=0.02);
17     case 2
18         wK = 5;
19         pos_mask = find(p_cloud.d >= 0.01 & p_cloud.d < 0.02);
20     case 3
21         wK = 7;
22         pos_mask = find(p_cloud.d >= 0.005 & p_cloud.d < 0.01);
23     case 4
24         wK = 9;
25         pos_mask = find(p_cloud.d < 0.005);
26     end
27
28     % compute normals for specific mask size wK
29     p_cloud = calc_point_normals(p_cloud,wK);
30
31     normals_x(pos_mask) = normals_x(pos_mask)+p_cloud.nx(pos_mask);
32     normals_y(pos_mask) = normals_y(pos_mask)+p_cloud.ny(pos_mask);
33     normals_z(pos_mask) = normals_z(pos_mask)+p_cloud.nz(pos_mask);
34 end
35
36 p_cloud.nx = reshape(normals_x,rows,cols);
37 p_cloud.ny = reshape(normals_y,rows,cols);
38 p_cloud.nz = reshape(normals_z,rows,cols);
```

In order to differentiate between various methods for calculating point normals and to determine the effects of using different filter masks, unique or adapted, four spatially distributed regions were selected in the data set, see [Figure 3.9](#).



Figure 3.9: Regions for the evaluation of methods for computation of point normals.

### 3.3. NEW SEGMENTATION ALGORITHM FOR TLS DATA

For the data set, see [Figure 3.2](#), the point normals were calculated using the methods of simple filtering (uniform mask), see [Code A.3](#), filtering with different masks (adapted mask), see [Code 3.3](#), and Singular Value Decomposition (SVD), introduced in [Section A.1.3](#), followed by the corresponding curvature information for each result.

Subsequently, the mean SDs of the values within the regions were calculated for each curvature layer. Since the curvature describes the change of the point normals in a region, the calculated SDs in homogeneous areas represent a criterion for the remaining noise of the normals, which in turn characterizes the quality of the method used. [Table 3.1](#) lists the results of the regions for the different methods, as well as the processing time for the calculation of the point normal.

Table 3.1: Comparison of different methods for normal calculation on a raster data structure.

CALC. METHOD	PROC. TIME [s]	NOISE [SD]			
		REGION 1	REGION 2	REGION 3	REGION 4
SVD	148.96009	0.00695	0.02596	0.01637	0.01146
uniform mask	0.48976	0.00935	0.030479	0.01674	0.01284
adapted mask	2.69816	0.00526	0.01040	0.00948	0.00590

One can easily recognize the enormous differences in the processing time of the methods based on filtering compared to the calculation with SVD, which is much slower. For the remaining noise in the individual regions, values of a similar magnitude emerge for the method based on simple filtering and SVD, whereby generally, as was to be expected, in the regions with a narrower point spacing (regions 2 and 3) the noise is higher in comparison to the other regions, indicating point normals with less precision. The method adjusting the filter mask on the basis of the point spacing shows a significant noise reduction of up to factor 3 in these areas, which indicates a better normal determination compared to the other methods.

### 3.3 NEW SEGMENTATION ALGORITHM FOR TLS DATA

On the basis of the pre-processing steps explained in [Section 3.2](#), this section focuses on the workflow of segmentation, which is essentially based on a combined pixel- and region-based solution, see [Section 3.3.1](#) and [Section 3.3.2](#). The entire process diagram of the algorithm is shown in [Section 3.3](#).

Starting from a laser scan available as a list of 3D coordinates  $x$ ,  $y$ ,  $z$  and intensity values  $I$ , the first step is the conversion with [Code 3.1](#) into the equivalent polar representations, see [Figure 3.2](#) and [Figure 3.3A](#).

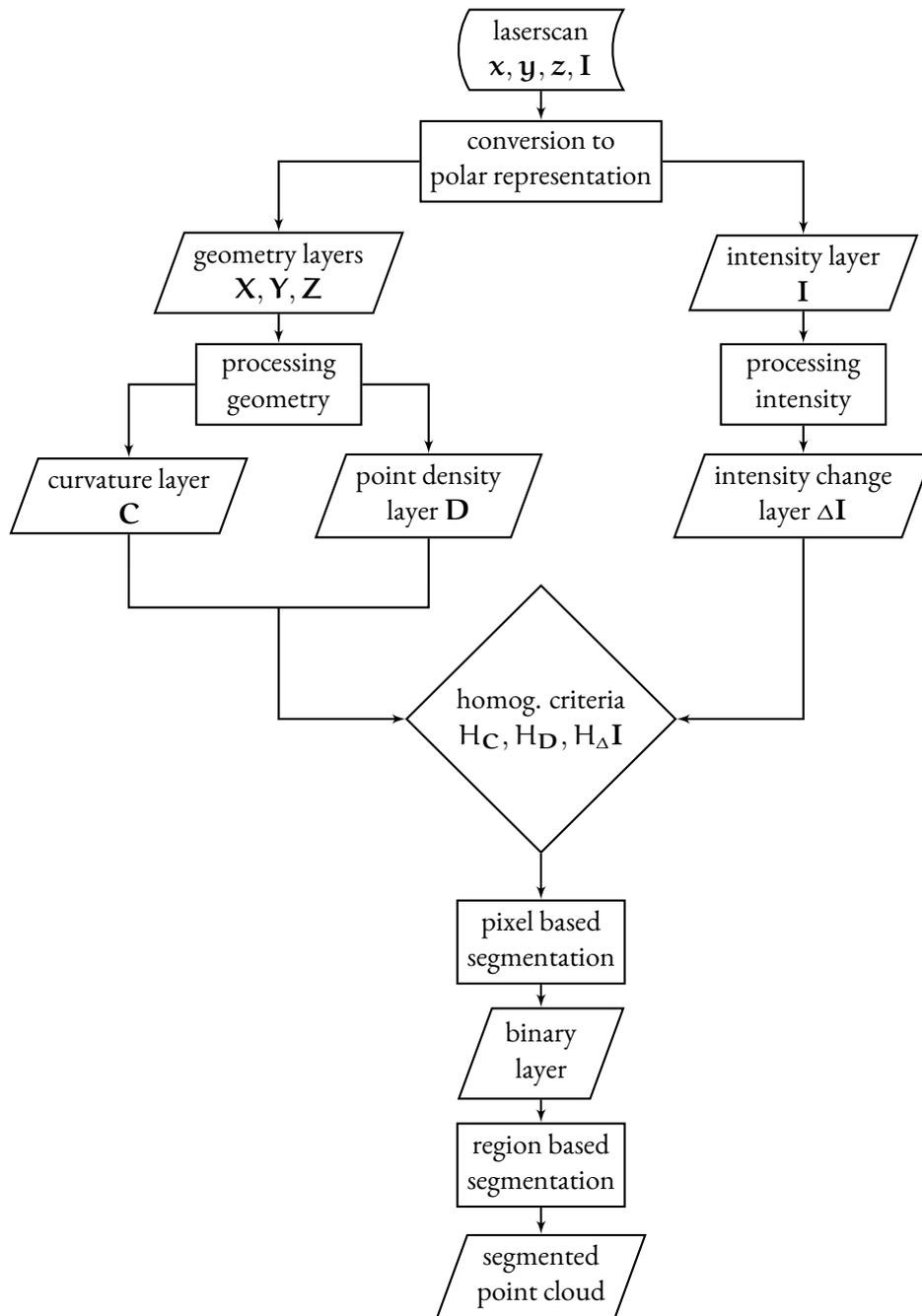


Figure 3.10: Process diagram of the segmentation algorithm.

Subsequently, the pre-processing of the geometry and intensity information, provided as  $X$ -,  $Y$ -,  $Z$ -layer and  $I$ -layer, is carried out in [Section 3.2](#), yielding the basic layers  $C$  (curvature information),  $D$  (point spacing information) and  $\Delta I$  (change in intensity) for the subsequent segmentation.

### 3.3. NEW SEGMENTATION ALGORITHM FOR TLS DATA

By means of previously determined homogeneity criteria  $H_C$ ,  $H_D$  and  $H_{\Delta I}$  for curvature, point spacing, and change of intensity, to be satisfied by the regions being segmented, the pixel-based segmentation, see Section 3.3.1, provides a purely binary representation, see Figure 3.11A, without knowledge of any neighborhood relationships.

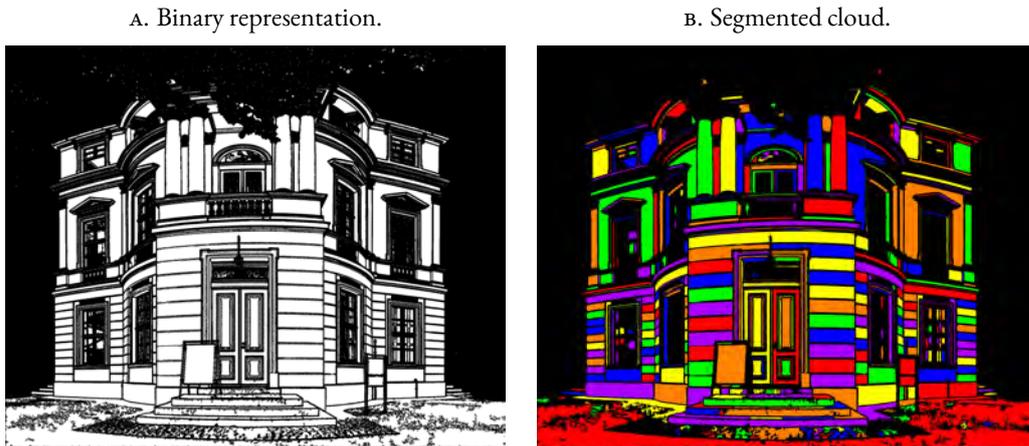


Figure 3.11: Results of pixel-based segmentation (binary matrix left) and region-based segmentation (color-coded regions right).

On the basis of the white pixels in the left Figure 3.11A, meeting the homogeneity requirements, in the last step a neighborhood analysis groups the connected pixels with the region-based segmentation from Section 3.3.2 into regions. Figure 3.11B on the right shows the final segmented point cloud with the separated regions, each assigned a different<sup>12</sup> color.

#### 3.3.1 PIXEL-BASED SEGMENTATION

The pixel-based segmentation is the fundament for the following region-based approach in Section 3.3.2. Using previously defined homogeneity criteria  $H_C$ ,  $H_D$  and  $H_{\Delta I}$  for the segmentation base layers  $C$ ,  $D$  and  $\Delta I$  determined in Section 3.2, the values within each layer can be divided into two classes.

Each value within these matrices is checked for homogeneity according to its specific criteria following the rules in (3.16a) to (3.16c). A value thus meets the homogeneity requirements if it is smaller than the

<sup>12</sup> Due to the recurring discrete color map and the sequential color assignment, adjacent regions can be of the same color and yet spatially separated.

predefined homogeneity threshold and is labeled with 1, otherwise with 0.

$$\mathbf{B}_C(i, j) = \begin{cases} 1, & \mathbf{C}(i, j) \leq H_C \\ 0, & \mathbf{C}(i, j) > H_C \end{cases} \quad (3.16a)$$

$$\mathbf{B}_D(i, j) = \begin{cases} 1, & \mathbf{D}(i, j) \leq H_D \\ 0, & \mathbf{D}(i, j) > H_D \end{cases} \quad (3.16b)$$

$$\mathbf{B}_{\Delta I}(i, j) = \begin{cases} 1, & \Delta \mathbf{I}(i, j) \leq H_{\Delta I} \\ 0, & \Delta \mathbf{I}(i, j) > H_{\Delta I} \end{cases} \quad (3.16c)$$

For the determination of the threshold values for homogeneity, a histogram- based approach can be applied, where [Figure 3.12](#) shows the histograms of the layers in [Figure 3.6](#).

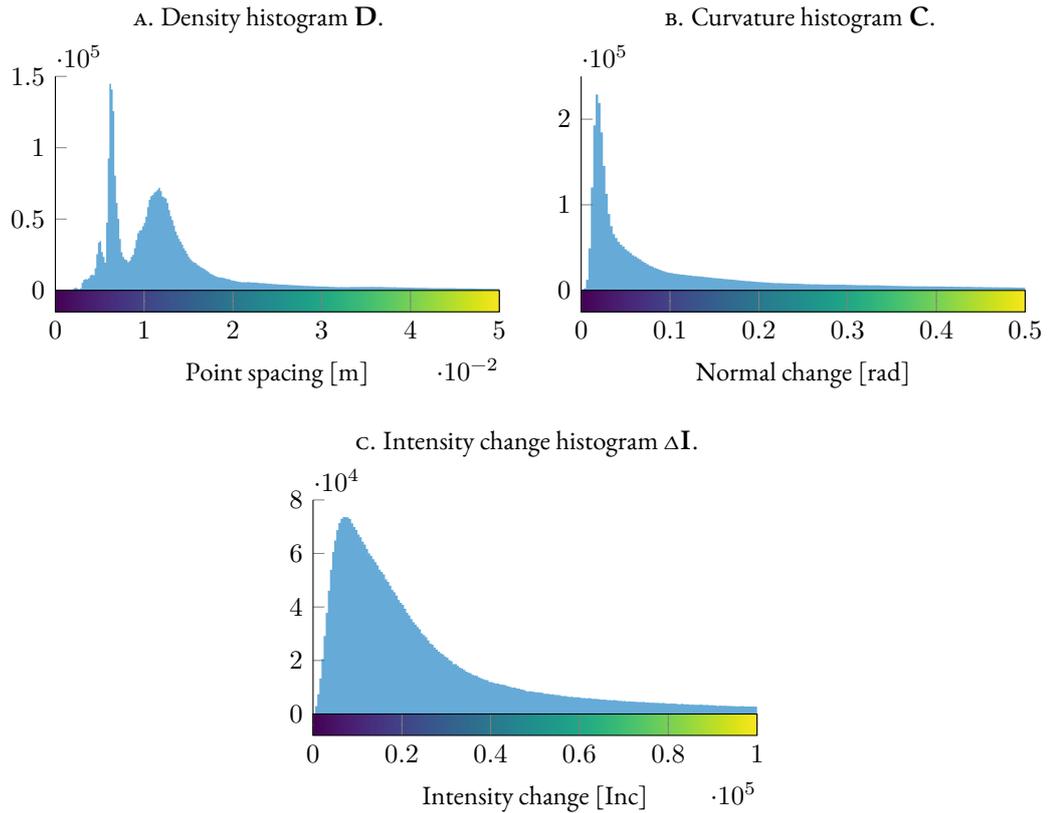


Figure 3.12: Computed information layers regarding curvature, point spacing and change in intensity.

### 3.3. NEW SEGMENTATION ALGORITHM FOR TLS DATA

The histogram for the characterization of the point spacing, see [Figure 3.12A](#), shows the most frequent point spacings up to 2 cm, whereby point spacings from 5 cm are no longer caused by the scanning process and the object geometry, so that the criterion  $H_D=0.05$  can be selected.

Homogeneous curvature information in [Figure 3.12B](#) can be found in the range between 0 and 0.1 rad with the highest frequencies, a meaningful threshold value  $H_C$ , can therefore be found behind the highest peak between 0.05 and 1. Values above represent non-homogeneous edge information.

Minor intensity changes within the magnitude of three times the  $SD$  of the intensities, see also [Figure 2.3A](#), or constant intensity conditions can be found in [Figure 3.12C](#) up to approx.  $3 \cdot 10^4$ , or  $4 \cdot 10^4$  Inc after the peak. Values above represent abrupt intensity changes due to strong changes in the object geometry or texture, therefore a favorable threshold value is given with  $H_{\Delta I}=3 \cdot 10^4$ .

The conjunction of the binary matrices in [Equations \(3.16a\) to \(3.16c\)](#) results in a binary matrix

$$\mathbf{B}_{C,D,\Delta I} = \mathbf{B}_C \wedge \mathbf{B}_D \wedge \mathbf{B}_{\Delta I} \quad (3.17)$$

in which the pixels are labelled with  $I$ , also labelled with  $I$  at the same locations in the single binary matrices, characterized by those pixels which satisfy all homogeneity requirements equally.

The example [Code 3.4](#) shows the pixel-based segmentation based on the binaryzation of the matrices `p_cloud.c`, `p_cloud.d` and `p_cloud.di` (or rather  $C,D$  and  $\Delta I$ ) by the threshold values `curv_threshold`, `dist_threshold` and `int_threshold` (or rather  $H_C$ ,  $H_D$  and  $H_{\Delta I}$ ).

Code 3.4: Pixel-based segmentation with all information layers.

```

1 %% pixel based segmentation
2
3 % homogeneity criteria (thresholds)
4 curv_threshold = 0.05;
5 dist_threshold = 0.05;
6 int_threshold = 4*10^4;
7
8 % get binary representations
9 bin_mat_c = p_cloud.c < curv_threshold;
10 bin_mat_d = p_cloud.d < dist_threshold;
11 bin_mat_di = p_cloud.di < int_threshold;
12
13 % pixels that meet all criteria
14 bin_mat = bin_mat_c & bin_mat_d & bin_mat_di;

```

The result of the pixel-based segmentation of the data set represented by [Figures 3.2A to 3.2C](#) and [3.3A](#) applying the homogeneity criteria  $H_C = 0.05$ ,  $H_D = 0.05$  and  $H_{\Delta I} = 3 \cdot 10^4$  is the binary representation in [Figure 3.11A](#).

CONCLUDING REMARKS: Neighborhood relations between the pixels are not considered with this procedure. The segmentation result for the subsequent region-based segmentation, see Section 3.3.2, depends primarily on the appropriate choice of homogeneity criteria.

The homogeneity criteria may have to be determined again for further data sets of the same object generated from other standpoints, if the conditions regarding geometry or radiometry at the object differ strongly from the previous ones.

In this case, an automated derivation of homogeneity criteria is recommended to ensure an individual and optimal evaluation for each *point cloud* of a standpoint. One can take advantage of proven thresholding methods for gray level histograms in image segmentation, such as the method according to OTSU (1979). For simple, i. e. clearly separable histograms with a defined bi-class distribution of the values, a global method is sufficient, whereas for histograms with a diverse distribution, a division via a multi-level thresholding method is necessary.

For instance, to handle varying geometric conditions within a scan, it can also be useful to apply a local, so-called adaptive thresholding.

### 3.3.2 REGION-BASED SEGMENTATION

With the pixel-based segmentation in Section 3.3.1, a representation of values already exists that meets the required criteria for segmentation with regard to surface geometry, intensity and point spacing. These values labelled in a binary matrix with 1 have no affiliation to their neighborhoods so far, see also the neighborhood relations in Figure 3.5, as a result a subsequent grouping of the neighborly connected pixels in regions must take place. Such a segmentation of a binary data representation can be realized very efficiently with methods of the so-called Connected Components (CCs) labeling, see Section 1.5.

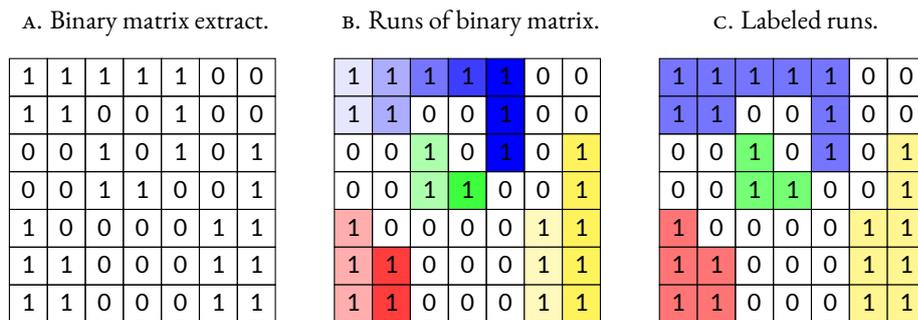


Figure 3.13: Run Length Encoding (RLE) in the context of Connected Component (CC) Labeling for a binary matrix.

Considering a section of the binary matrix from Figure 3.11A, see Figure 3.13A, the matrix can be represented in row direction for each column as a sequence (so-called runs) of individual pixel values

solely by the starting pixel and the length of the sequence<sup>13</sup>, so-called runlength (HARALICK and SHAPIRO 1992, pp.40), see Figure 3.13B. Hence the name of this method, which is known as Run Length Encoding (RLE) in the context of lossless data compression.

Assuming a direct neighborhood, cf. also Figure 3.7, runs in a subsequent column are adjacent if there is at least one pixel in the same row, leading to the following connections after Figure 3.13B:

- run 3 is connected to run 1
- run 4 is connected to run 2
- run 5 is connected to run 3
- run 7 is connected to run 5
- run 8 is connected to run 6
- run 9 is connected to run 7
- run 11 is connected to run 10

The dependencies of the runs to each other result in the following *graphs* in Figure 3.14, whose nodes now correspond to the individual runs in contrast to a single pixel representation as in Section 1.5, i. e. Figure 1.20C.

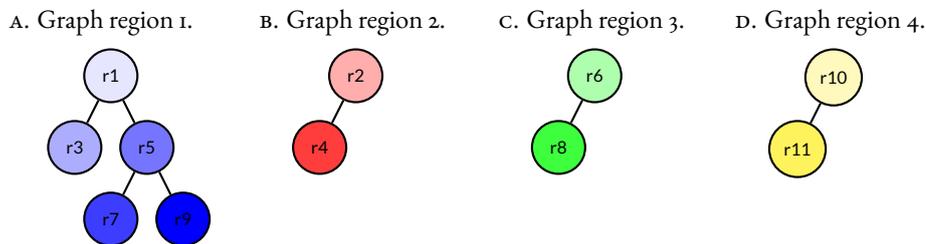


Figure 3.14: *Graph* representations of the connectivity for the individual runs in Figure 3.13B.

The graphs in Figures 3.14A to 3.14D each represent four independent regions via the connected runs, labeled with different colors in Figure 3.13C.

The graph-based runlength labeling algorithm is called by `CC = get_cc_of_rl_graph(BW, conn)`<sup>14</sup> under the input parameters `BW` (binary image) and `conn = 4` (connectivity) and is composed of the following steps:

<sup>13</sup> A sequence can also be described with its starting and ending pixels.

<sup>14</sup> Uses function `rlenc`, see Code A.7 in Section A.3

## SEGMENTATION BASED ON GEOMETRIC AND RADIOMETRIC INFORMATION

1. Finding runs applying `function` `rlenc` by computing the run lengths of each run (lines 17 to 19) while
2. determining which runs on the previous column are connected (lines 26 to 42)
3. Construct a *graph representation*, i. e. a sparse *adjacency matrix* (lines 50 to 51), from the pairwise connected runs `pairs` (line 48)
4. Compute *equivalence classes*, see also [Section 1.5](#), (lines 53 to 54) in order to label the runs according to the class they belong to (lines 56 to 65)

Code 3.5: Labeling connected components of a graph consisting of run lengths.

```

1 function CC = get_cc_by_rle(BW, conn)
2 %input: binary matrix BW and connectivity conn 4 or 8
3 %output: struct CC of indices for connected components
4
5 % connectivity offset
6 if isequal(conn,4)
7     c = 0; %direct connectivity
8 elseif isequal(conn,8)
9     c = 1; %full connectivity
10 end
11
12 [rows,cols] = size(BW);
13 runs = cell(cols,1);
14
15 n_labels = 0;
16 pairs = [];
17 for i = 1:cols
18     column = BW(:,i);
19     [runs_col,labels] = rlenc(column);
20
21     if ~isempty(runs_col)
22         runs{i} = [runs_col+(i-1)*rows,labels+n_labels];
23         n_labels_col = labels(end);
24         n_labels = n_labels+n_labels_col;
25
26         % check which column runs are connected
27         if i > 1 && ~isempty(runs{i-1})
28             % iterate over runs
29             n_runs = numel(runs{i}(:,3));
30             for j = 1:n_runs
31                 % runs are connected if they overlap
32                 ind_j_i = (runs{i}(j,1)-rows >= runs{i-1}(:,1)-c...
33                     & runs{i}(j,1)-rows <= runs{i-1}(:,2)+c)...
34                     | (runs{i}(j,2)-rows >= runs{i-1}(:,1)-c...
35                     & runs{i}(j,2)-rows <= runs{i-1}(:,2)+c)...
36                     | (runs{i}(j,1)-rows < runs{i-1}(:,1)...
37                     & runs{i}(j,2)-rows > runs{i-1}(:,2));
38                 n_id = sum(ind_j_i);
39                 pairs_tmp = [repmat(runs{i}(j,3),n_id,1),...
40                     runs{i-1}(ind_j_i,3)];

```

```

41         pairs = [pairs;pairs_tmp];
42     end
43 end
44 end
45 end
46
47 pairsd = [pairs(:,2),pairs(:,1)];  pairs = [pairs;pairsd];
48 pairs = sortrows(pairs);
49
50 % sparse adjacency matrix
51 A = sparse([pairs(:,1);(1:n_labels)'],[pairs(:,2);(1:n_labels)'],1);
52
53 % Dulmage-Mendelsohn decomposition of adjacency matrix
54 [p,~,r,~] = dmperm(A);
55
56 runs = cell2mat(runs);
57 n_nodes = length(r)-1;
58 for i = 1:n_nodes
59     ind_runs = p(r(i):r(i+1)-1);  runs_reg = runs(ind_runs,:);
60     n_runs = numel(ind_runs);  pix_idx = [];
61     for j = 1:n_runs
62         pix_idx_tmp = runs_reg(j,1):runs_reg(j,2);
63         pix_idx = [pix_idx,pix_idx_tmp];
64     end
65     CC.PixelIdxList{i} = pix_idx;
66 end

```

NOTE: This method is very efficient because each pixel is scanned only once and the connectivity analysis can be determined from the runs alone. The number of runs is much smaller compared to the number of foreground pixels used by the other methods, e. g. in [Section 1.5](#) and [Code 1.2](#), resulting in a smaller adjacency matrix.

### 3.3.3 OVERALL SEGMENTATION AND REGION ANALYSIS

The overall segmentation is composed of the pixel-based, see [Section 3.3.1](#), and region-based segmentation, see [Section 3.3.2](#), and describes the final process of the new segmentation algorithm, in which a complete segmentation of a point cloud is obtained on the basis of the pre-processed information channels by means of the transfer of the homogeneity criteria, in [Section 3.3](#).

A point cloud structure `p_cloud` with all the information layers pre-processed in [Section 3.2](#) and the variable `det_params`, which contains detection parameters, is transferred to [function](#) `get_homog_regions`<sup>15</sup>, which performs the overall segmentation.

<sup>15</sup> Uses [function](#) `get_cc_by_rle`, see [Code 3.5](#) in [Section 3.3.2](#)

The algorithm in [Code 3.6](#) detects contiguous regions that satisfy a minimum number of points `min_points`, have a certain neighborhood `conn`, and fulfill homogeneity criteria based on threshold values `curv_thresh`, `dist_thresh`, `int_thresh`, all defined in `det_params`, and is structured in:

1. Perform pixel-based segmentation (lines 13 to 18), see also [Section 3.3.1](#), according to homogeneity thresholds in `det_params` (lines 8 to 11)
2. Perform region-based segmentation (lines 20 to 22)
3. Further processing including separation of regions having a minimum number of points (lines 25 to 29) and individual region analysis (lines 34 to 62)

Code 3.6: Segmenting of regions based on homogeneity criteria.

```

1 function [ regions ] = get_homog_regions(p_cloud, det_params)
2 %input 1: struct p_cloud with information .x,.y,.z,.i,.c,.d,.di
3 %input 2: detection parameters det_params
4 %output: struct regions
5
6 cloud_size = size(p_cloud.x);
7
8 % read detection parameters
9 min_points = det_params(1,1);   conn = det_params(2,1);
10 curv_thresh = det_params(3,1); dist_thresh = det_params(4,1);
11 int_thresh = det_params(5,1);
12
13 %% pixel based segmentation
14 bin_mat_c = p_cloud.c < curv_thresh;
15 bin_mat_d = p_cloud.d < dist_thresh;
16 bin_mat_di = p_cloud.di < int_thresh;
17 % only pixel that satisfy all criteria
18 bin_mat = bin_mat_c & bin_mat_d & bin_mat_di;
19
20 %% region based segmentation
21 % get connected components
22 CC = get_cc_by_rle(bin_mat,conn);
23
24 %% further processing
25 % get only regions that satisfy min_points
26 n_points_region = cellfun(@numel,CC.PixelIdxList);
27 idx_over = find(n_points_region > min_points);
28 [~,idx_sort] = sort(n_points_region(idx_over),'descend');
29 idx_over = idx_over(idx_sort);
30
31 % number of regions
32 num_regions = numel(idx_over);
33
34 % analyze regions
35 for i = 1:num_regions
36     regions(i).region_id = i;
37     idx_p_r = CC.PixelIdxList{idx_over(1,i)};
38     regions(i).point_indices = idx_p_r;

```

```

39
40 % compute boundary
41 regions(i).bbox = get_bbox(cloud_size,idx_p_r);
42
43 % extract 3D information and accuracy
44 p_3D = [p_cloud.x(idx_p_r),p_cloud.y(idx_p_r),p_cloud.z(idx_p_r)];
45 p_3D_p = [p_cloud.s_x(idx_p_r),p_cloud.s_y(idx_p_r),...
46 p_cloud.s_z(idx_p_r)];
47
48 % center of gravity
49 regions(i).cog = mean(p_3D)';
50
51 % plane approximation
52 [plane_approx_region,plane_dists] = fitplane(p_3D);
53 regions(i).plane_params = plane_approx_region;
54
55 % average point accuracy
56 regions(i).av_point_acc = mean(sqrt(p_3D_p(:,1).^2 ...
57 +p_3D_p(:,2).^2+p_3D_p(:,3).^2));
58
59 % maximum distance to plane
60 regions(i).max_plane_dist = max(plane_dists);
61
62 % mean intensity of region
63 regions(i).mean_intensity = mean(p_cloud.i(idx_p_r));
64 end

```

The result of `function get_homog_regions` is a structure `regions`, arranged in [Figure 3.15](#).

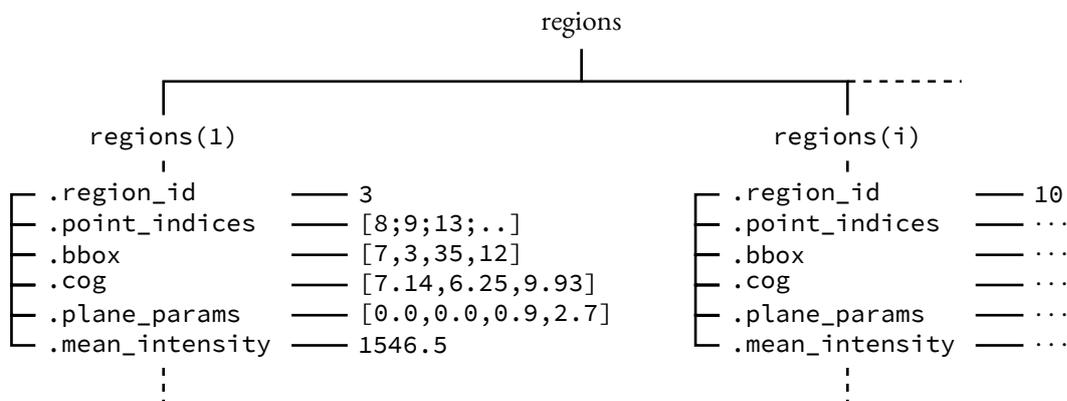


Figure 3.15: Representation of a `regions` structure array in [MatLab](#).

This includes information, among others, such as the average intensity `.mean_intensity`, the center of gravity `.cog` or the bounding box `.bbox` of a region `regions(i)`, and whether a region represents a plane with `.plane_params` or not.

NOTE that the extended region analysis based on the existing geometry, as well as radiometric information is important for the further processing of the *point clouds*, such as the sub-segmentation in [Section 4.3.2](#) or the matching procedure in [Section 4.4](#).

Code 3.7: Access to the 3D coordinates of a specific region in the point cloud.

```

1 x_region = p_cloud(1).x(region(3).point_indices);
2 y_region = p_cloud(1).y(region(3).point_indices);
3 z_region = p_cloud(1).z(region(3).point_indices);

```

The [Code 3.7](#) example illustrates the extraction of the 3D coordinates `x_region`, `y_region` and `z_region` of region 3 in *point cloud* 1, used e. g. to access the coordinates for estimation purposes.

### 3.4 SEGMENTATION RESULTS

In the following, some segmentation results that have been generated with the algorithm previously described in [Section 3.3](#) will be examined in more detail.

[Figure 3.16A](#) on the left shows the full details of the segmentation result for the "Belvedere" test data set used in [Section 3.3](#) to introduce the algorithm, see also [Figure 3.1A](#). The [Figure 3.16B](#) on the right illustrates the segmentation of the second test data set, a scan of an inner dome in the outer area of the "Orangery" in Potsdam, see [Figure 3.1B](#).

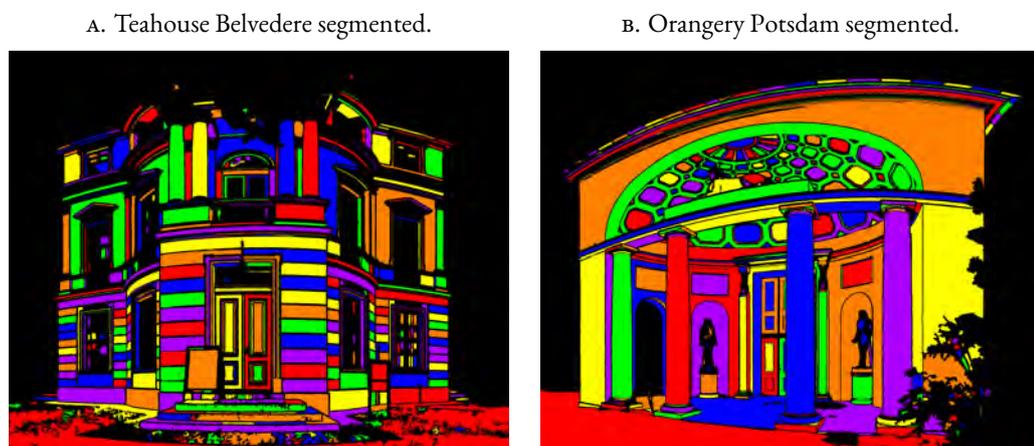


Figure 3.16: Results of the region-based segmentation (color-coded regions) of the front view of the test objects Belvedere (left) and inner dome in the outer area of the Orangery (right).

The objects in [Figure 3.16](#) are composed of a large variety of geometric elements. It can be seen that both in the left [Figure 3.16A](#) and in the right [Figure 3.16B](#) pillars, planes and curved surface elements

were recognized equally by the algorithm, characterizing the data-driven approach in a particular manner. Also the completeness of the segmentation results and the variation of the segment sizes are an indication for a functioning segmentation algorithm.

For a closer look on both segmentation results, some detailed views of representative areas are presented with Figure 3.17 for the "Belvedere" object in Figure 3.16A as well with Figure 3.18 for the "Orangery" object in Figure 3.16B.



Figure 3.17: Detailed view from the overall result in Figure 3.16A of the region-based segmentation (color-coded regions) of the front view of the test object Belvedere.

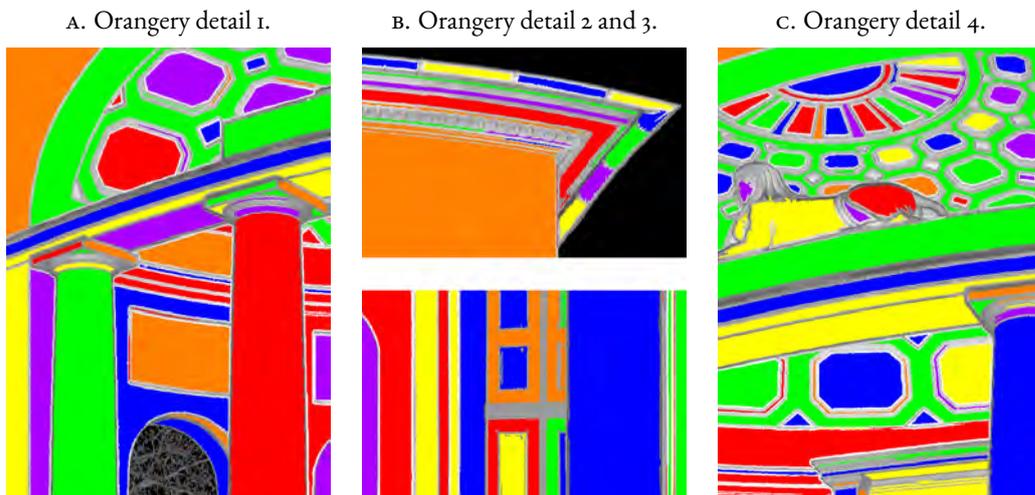


Figure 3.18: Detailed view from the overall result in Figure 3.16B of the region-based segmentation (color-coded regions) of the inner dome in the outer area of the Orangery.

Even areas delimited by small depressions, such as fugues, were reliably segmented. Evidence of this can be found on the facade of the example "Belvedere", see the detailed views of separated stone parts of the facade in [Figures 3.17A](#) and [3.17C](#), and on the segments of the dome of the "Orangery" data set, see the detailed views of indentations in [Figures 3.18A](#) and [3.18C](#) as well the separation from just minimal indentations or fugues for the door parts (lower part of [Figure 3.18B](#)) or the rim stones at the roof (upper part of [Figure 3.18B](#)).

In addition, it can be observed for all detailed views in [Figures 3.17](#) and [3.18](#) that the segmentation extends almost to the respective actual segment boundaries, caused by spatial or radiometric changes, either by depressions, gaps or edges, geometry jumps as well as color changes.

To also validate the applicability of the segmentation algorithm with regard to natural objects and conditions, a part of the data set of the landslide area in [Figure 3.1C](#) was evaluated. [Figure 3.19](#) shows the segmented boulder field, whereby the upper and lower area, in which only trees and shrubs occurred, compare with [Figure 3.1C](#), was not considered.

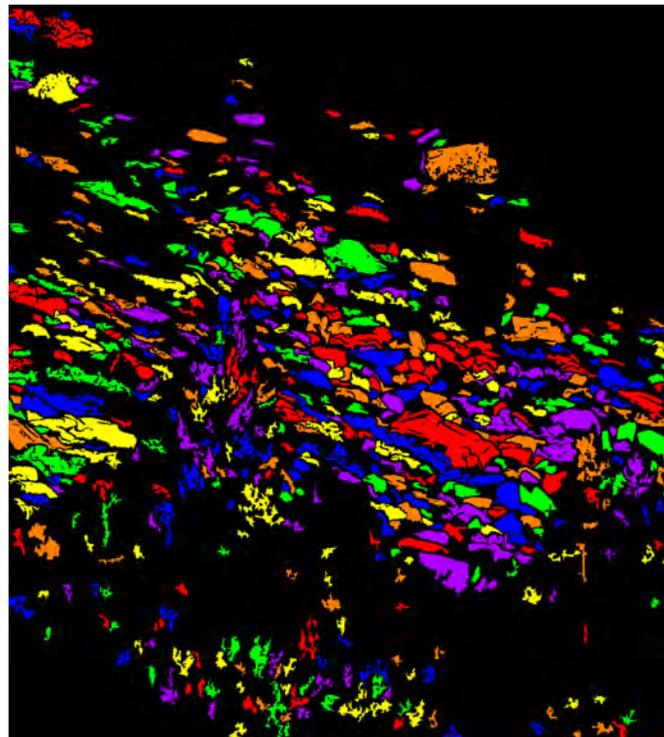


Figure 3.19: Results of the region-based segmentation (color-coded areas) of a small area of the landslide in Obergurgel.

As can easily be seen, both large and small boulders and single stones could be reliably detected and separated from each other, confirming the applicability of the method for natural objects.

Table 3.2 lists the segmentation criteria that `function get_homog_regions`, see Code 3.6, uses as input parameters for the various data sets that lead to the segmentation results in Figure 3.16 and Figure 3.19.

Table 3.2: Segmentation parameters used to evaluate the various example data sets.

DATASET	SEGMENTATION PARAMETERS				
	MIN. POINTS	CONNECTIVITY	$H_C$	$H_D$	$H_{\Delta I}$
Belvedere	100	4-connected	0.05	0.05	$3.0 \cdot 10^4$
Orangery	100	4-connected	0.08	0.05	$3.0 \cdot 10^4$
Landslide	15	4-connected	0.3	0.05	–

Furthermore, the new segmentation method meets almost all the requirements for an algorithm described in Section 1.10. The first requirement, that the segmented areas in their entirety should correspond to the complete data set, is, as with many other methods, only valid in an attenuated form due to the limitation by the choice of homogeneity criteria<sup>16</sup>. Only the fifth requirement, cf. Section 1.10 and Item 5, is not fulfilled by the homogeneity criteria uniformly selected for the entire segmentation range. The algorithm differs strongly from other methods, for instance the "Region Growing", which meet only a few of these requirements.

#### 3.4.1 INFLUENCE OF HOMOGENEITY CRITERIA

Figure 3.20 shows a detailed view of the upper part of the right red pillar depicted in Figure 3.11B and indicates how the use of the point spacing information of layer **D**, see also Figure 3.6A affects the segmentation of a pillar. In Figure 3.20A on the left, the points meet the homogeneity criterion  $H_C$  for the curvature information of layer **C**, see Figure 3.6B, but it can be seen that not all points are spatially adjacent. However, in Figure 3.20B on the right, layer **D** is used as additional information, which allows a clear and distinct separation even in 3D space. The points now also fulfil the homogeneity criterion  $H_D$  for complying with an average point distance and are now also spatially adjacent.

<sup>16</sup>The homogeneity criteria can practically also be selected individually for certain areas in a matrix by subdividing the matrix and performing the binaryization, see Section 3.3.1, for each subarea and reassembling these to an entire binary matrix.

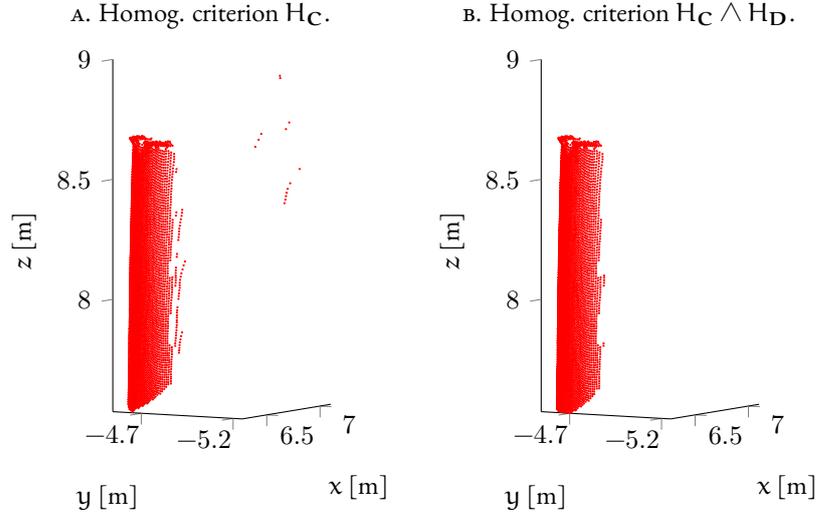


Figure 3.20: Upper part of the segmented red pillar in Figure 3.11B right without point spacing information (left pillar) and with (right pillar).

### 3.4.2 PERFORMANCE EVALUATION

Table 3.3 lists various parameters related to the datasets of each standpoint required to capture the entire object shown in Figure 3.1A. Among these datasets, a distinction is made between the number of data points, the number of regions segmented by the segmentation algorithm explained in Section 3.3 and the processing time required for segmentation with and without region analysis.

Table 3.3: Criteria for various processed data sets to evaluate the segmentation algorithm.

DATASET	NO. OF POINTS	NO. OF SEGMENTS	PROC. TIME [s]	
			CCL	CCL + ANALYSIS
1	$5.204 \cdot 10^6$	487	0.0707	2.2269
2	$4.938 \cdot 10^6$	453	0.0595	1.8995
3	$2.186 \cdot 10^6$	198	0.0261	0.4999
4	$7.367 \cdot 10^6$	511	0.1040	3.4385
5	$1.673 \cdot 10^6$	48	0.0148	0.1230
6	$1.966 \cdot 10^6$	234	0.0215	0.4191
7	$3.330 \cdot 10^6$	323	0.0364	1.0278
8	$7.794 \cdot 10^6$	304	0.1070	3.4419
9	$6.391 \cdot 10^6$	420	0.0755	3.0125

IN THIS CONTEXT, the pure region-based segmentation in terms of the `CCL` without any preprocessing steps, see [Section 3.3.2](#) and especially the efficiency of `function get_cc_by_rle` by [Code 3.5](#) and as part of [Code 3.6](#) line 22, is compared against the latter in addition to the subsequent region analysis concerning the part termed as `%% further processing` (lines 24 to 62) of the `function get_homog_regions`, see [Code 3.6](#), and the result of the analysis as structure of additional information in [Figure 3.15](#).

It can be seen that the processing time of the segmentation algorithm does not depend on the number and size of the regions found, but solely on the size of the data set. This is mainly due to the highly efficient region analysis algorithm, see [Section 3.3.2](#), which scans each pixel only once.

### 3.5 CONCLUSION

With the final result representation in [Section 3.4](#) it has been demonstrated that the segmentation algorithm is able to provide an almost complete and very detailed segmentation independent of the geometry and size of the regions to be segmented, see also [Figures 3.16A to 3.16B](#) and [Figure 3.19](#). Furthermore, it could be illustrated that the additional use of information, such as the point spacing information as depicted in [Figure 3.20](#), improved the segmentation results by a clearer spatial delineation. Due to the way the presented method is designed, further information can be used for segmentation without additional effort, such as the precision of the 3D coordinates, which can be used by the development of a new *stochastic model* as introduced by [Chapter 2](#). In general, due to the raster data structure, cf. [Section 1.4](#), the method is clearly superior to other methods in terms of execution time, see [Table 3.3](#), by using efficient image processing routines, especially the region-based analysis in [Section 3.3.2](#) of the data.

**FINAL REMARK:** The result of the segmentation from this chapter serves as a source for a sub-segmentation into planar regions, see [Section 4.3.2](#), for the registration procedure, presented in [Chapter 4](#).



*Probleme kann man niemals mit derselben Denkweise lösen, durch die sie entstanden sind.*

Albert Einstein

# 4

## Synergetic Segmentation and Registration

NUMEROUS ESTABLISHED SOLUTIONS ARE KNOWN for the sole and independent task of segmentation or registration, see [Sections 1.10](#) and [1.11](#). The essential work steps regarding the automated processing of *point clouds* are based on different strategies and data concepts, see [Sections 0.1](#) and [1.10](#), and each represent stand-alone solutions. Thus existing strategies for the solution of both tasks in the first processing step are based on a registration detached from the segmentation. A challenge here is the subsequent processing step of segmentation, requiring a complex re-establishment of topological relationships between individual points, cf. [Section 1.4](#).

In ([MAHMOUDABADI et al. 2016](#)) the segmentation of a single scan is efficiently solved, but the context of the registration is not discussed. A registered *point cloud* is used by [RIVEIRO et al. \(2016\)](#), whereby the previous registration is not related to the segmentation. In both cases, the segmentation is in focus.

Considering registration procedures, in ([GIELSDORF 2009](#))<sup>1</sup> planes are segmented first, serving consequently as identical information for registration, but they have no other relevance. [RABBANI et al. \(2006\)](#) extend this concept by detecting several geometrical primitives in shape of cylinders, spheres and planes within the *point clouds* and using those informations for registration and modelling. However, the focus is mainly on registration.

---

<sup>1</sup> The concept of registration in this chapter follows the idea according to [GIELSDORF \(2009\)](#).

REMARK: A mutual support of segmentation and registration, which can also be characterized as "synergetic segmentation and registration", does not exist yet.

Consequently, this chapter introduces a new methodology that exploits the existing topology of individual scans for efficient segmentation in order to use extracted features as corresponding information for registration to complement the previous segmentation result.

Based on an example data set, see [Section 4.1](#), the entire process chain of the algorithm from segmentation in [Section 4.3](#) through matching and registration in [Sections 4.4](#) and [4.5](#) to the overall segmentation in [Section 4.6](#) is presented. A final presentation of the results, followed by a conclusion, is given in [Section 4.7](#).

#### 4.1 SAMPLE DATASET

The new approach for a synergetic procedure for the collaborative segmentation and registration of *point clouds* will be presented in the following using the processing of an example data set. The test object is the teahouse in the park of the Charlottenburg Palace "Belvedere", see [Figure 3.1A](#), already used to introduce the segmentation algorithm in [Chapter 3](#).

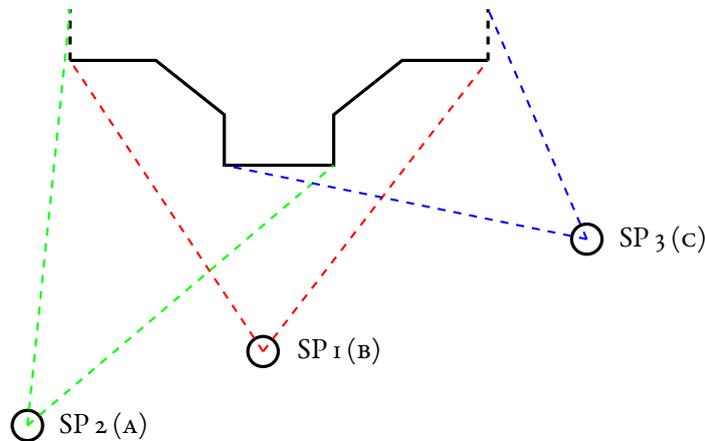


Figure 4.1: Overview of the standpoints (SP<sub>1-3</sub>) and acquisition areas (green, red and blue delimited) that led to the data A, B and C in [Figure 4.2](#), in relation to the object..

The test data set consists of data captured by a [TLS](#) from three standpoints. [Figure 4.1](#) shows the locations of the standpoints and the respective fields of view (red for SP<sub>1</sub>, green for SP<sub>2</sub> and blue for SP<sub>3</sub>) of the *laser scanner* on the test object in top view. [Figure 4.2](#) shows the different views onto the object from the different positions based on the intensity data.

#### 4.2. SYNERGETIC SEGMENTATION AND REGISTRATION OF TLS DATA

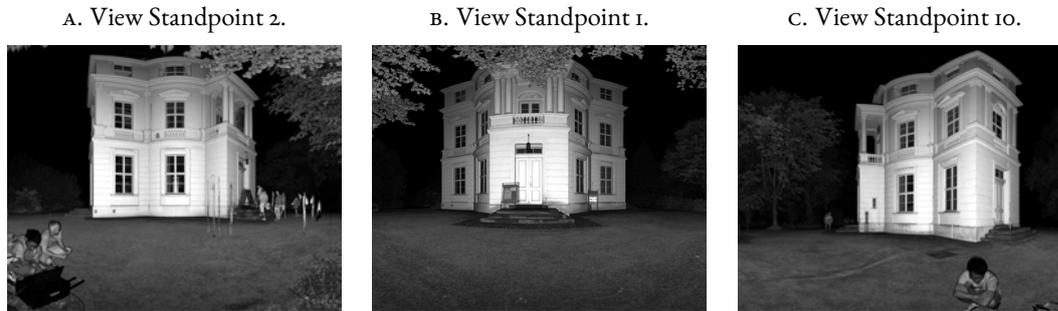


Figure 4.2: Views onto the object, respectively data slices, emerging from the different positions in Figure 4.1.

#### 4.2 SYNERGETIC SEGMENTATION AND REGISTRATION OF TLS DATA

Considering an object captured with a TLS from several standpoints and the resulting *point clouds* in individual local coordinate systems, Section 4.2 shows the algorithm of synergetic segmentation and registration.

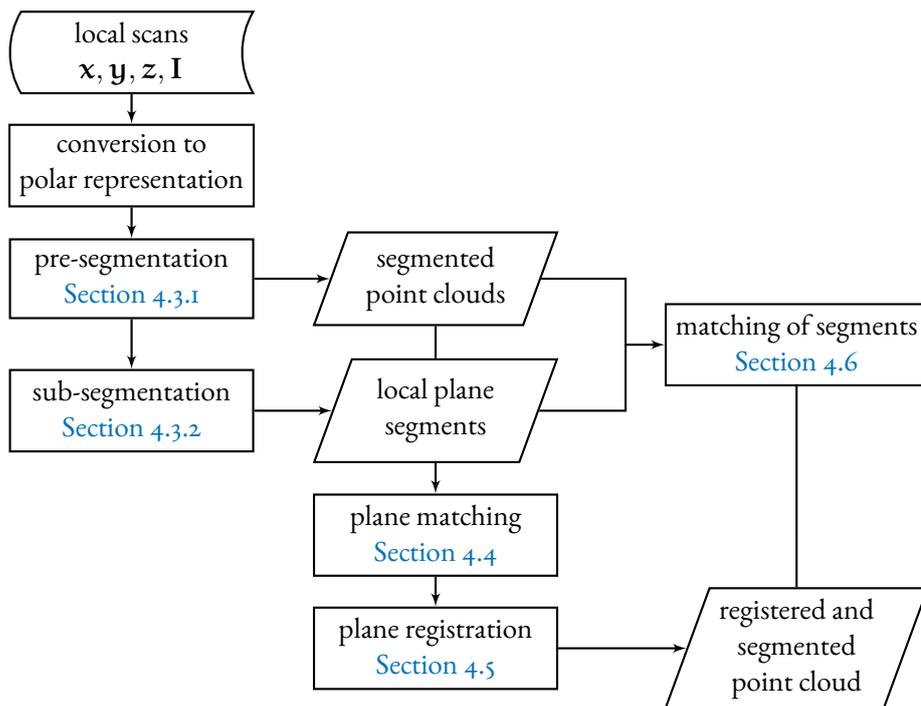


Figure 4.3: Process diagram of the synergetic segmentation and registration algorithm.

Initially, the conversion of the individual *point clouds* into their polar representation takes place with Code 3.1, and subsequently the pre-segmentation procedure, see Section 4.3.1, is applied. The resulting

segments are subjected to the sub-segmentation process, see [Section 4.3.2](#), which results in local planes for the respective *point clouds*. With the following matching procedure, see [Section 4.4](#), on the one hand the correspondences of planes of neighbouring scan or *point cloud* pairs are provided and on the other hand the approximate values for the subsequent final registration, see [Section 4.5](#), are determined. Using the calculated parameters of the *outer orientations* and the direct correspondence of local planes to the original segments obtained from the sub-segmentation algorithm, see [Section 4.3.2](#), individual local sub-segments can be assigned to each other, see [Section 4.6](#).

**REMARK:** The sub-segments form entire segments in the registered superordinate *point cloud*, composed of the local<sup>2</sup> *point clouds* of the standpoints, cf. [Section 1.3](#), transformed via the *outer orientations*, cf. [Section 1.3.1](#), into the common, i. e. superordinate coordinate system. As a result, a complete *point cloud* registered and segmented via several standpoints is obtained.

### 4.3 SEGMENTATION PROCESS

#### 4.3.1 PRE-SEGMENTATION OF INDIVIDUAL SCANS

Starting from the *point clouds* of the respective standpoints in a raster structure, see [Figure 4.2](#), the segmentation algorithm presented in [Chapter 3](#) was applied. [Figure 4.4](#) shows the results of the segmentation for the *point clouds* of positions 1 ([Figure 4.2B](#)), 2 ([Figure 4.2A](#)) and 10 ([Figure 4.2C](#)) from the front of the tea house "Belvedere", see also [Figure 3.1A](#).

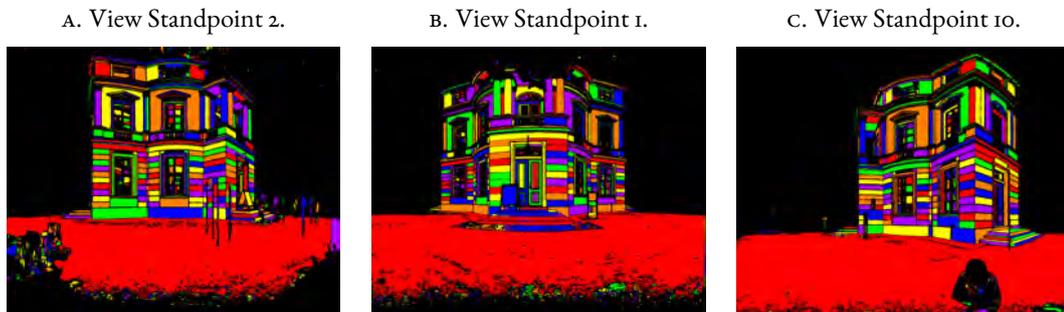


Figure 4.4: Presegmented regions (color-coded) based on the region-based segmentation in [Chapter 3](#) for the datasets A, B and C in [Figure 4.2](#) of the test object Belvedere.

The efficient neighborhood analysis using proven image processing algorithms, see [Section 3.2.1](#), characterizes the algorithm in comparison to 3D segmentation methods, especially with regard to processing

<sup>2</sup> Local in the context of *point clouds* describes the initial coordinate system, or the scanner coordinate system, see also [Section 1.3](#).

time, see [Table 3.3](#). The subdivision of the data is performed in segmentation based on the natural structure of the object, in contrast to conventional "split and merge" methods. The almost complete and very detailed segmentation, see [Figures 4.4A to 4.4C](#), reveals in regions that have been reliably detected independently of their geometry (planes, pillars, curved surfaces) and size. By using additional information, e. g. point distance information or intensity change, see [Section 3.4.1](#), the segmentation results could be improved by a clearer spatial segregation. This is reflected among others in the fact that even small depressions, e. g. fugues, can separate individual segments from each other.

#### 4.3.2 SUB-SEGMENTATION INTO PLANAR REGIONS

The registration that follows the segmentation uses information by planar surfaces in form of plane correspondences to derive the transformation parameters. Therefore the point clouds have to be subdivided into planar regions in the first step. For each segment originating from the pre-segmentation, see [Figure 4.4](#), a sub-segmentation into planar segments takes place using a special "split and merge" procedure. [Figure 4.5](#) shows the planes detected on the basis of the pre-segmented point clouds of positions 1, 2 and 10.

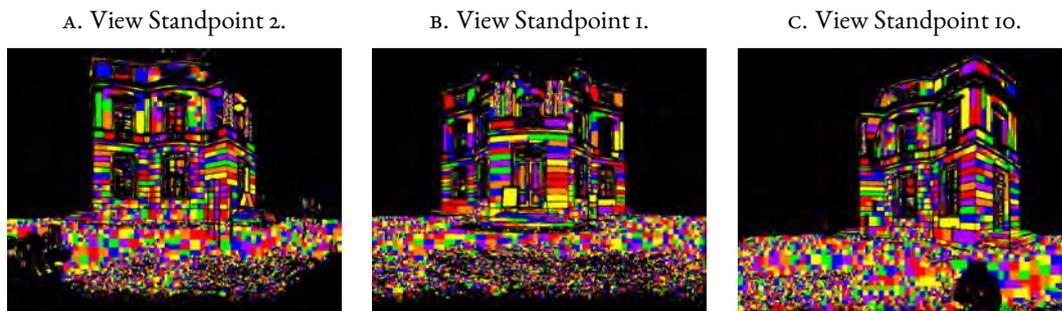


Figure 4.5: Detected planes from sub-segmentation on the basis of the pre-segmented *point clouds*, i. e. datasets A, B and C in [Figure 4.4](#).

[Table 4.1](#) summarises the result of the sub-segmentation from [Figure 4.5](#) compared to the previous segments, see [Figure 4.4](#), for the three positions. Specified is the number of split segments detected in the corresponding processing time based on a given number of original segments. Concerning the large soil segments, the majority and the smallest subdivisions resulted from the large variation of the geometry of the worn out subsoil.

Table 4.1: Results for various processed data sets to evaluate the segmentation algorithm.

STANDPOINT	NO. OF POINTS	SEGMENTS	SUB-SEGMENTATION	
			SUB-SEGMENTS	PROC. TIME [s]
1	$8.825 \cdot 10^6$	969	6573	25.726
2	$8.725 \cdot 10^6$	910	4771	19.678
10	$8.361 \cdot 10^6$	661	4068	16.738

The division of a segment from the pre-segmentation is carried out until the points  $P_i = [x_i \ y_i \ z_i]^T$  with  $i = 1, \dots, n$  of a sub-segment sufficiently satisfy a plane equation in the form

$$n_x x_i + n_y y_i + n_z z_i - d = 0 \quad (4.1)$$

under the condition

$$n_x^2 + n_y^2 + n_z^2 = 1. \quad (4.2)$$

The unknowns are the components  $n_x$ ,  $n_y$  and  $n_z$  of the normal vector of the plane and the perpendicular distance  $d$  of the plane from the coordinate origin. To determine the unknown parameters, the coordinate values  $x_i$ ,  $y_i$ ,  $z_i$  are reduced to their center of gravity

$$x_{cg} = x_i - \frac{1}{n} \sum_{i=1}^n x_i, \quad y_{cg} = y_i - \frac{1}{n} \sum_{i=1}^n y_i, \quad z_{cg} = z_i - \frac{1}{n} \sum_{i=1}^n z_i \quad (4.3)$$

and the vectors with the reduced coordinate values are combined in an augmented matrix

$$\mathbf{D} = \begin{bmatrix} x_1 - x_{cg} & y_1 - y_{cg} & z_1 - z_{cg} \\ x_2 - x_{cg} & y_2 - y_{cg} & z_2 - z_{cg} \\ \vdots & \vdots & \vdots \\ x_n - x_{cg} & y_n - y_{cg} & z_n - z_{cg} \end{bmatrix}. \quad (4.4)$$

On  $\mathbf{D}^T \mathbf{D}$ <sup>3</sup>, a Singular Value Decomposition (SVD)<sup>4</sup>, e. g. GOLUB and VAN LOAN (2013) p. 76 ff. and

<sup>3</sup> The SVD on  $\mathbf{D}$  directly is still possible, but not recommended if the plane consists of a large number of points, e. g. for a plane represented by 7k points the calculation is approx. times 5k faster, a plane with 13.3k points (double amount) approx. times 14k (3 times more).

<sup>4</sup> For more information how the SVD is applied programmatically, see also the function `fitPlane` by Code A.16.

see also [Section A.1.3](#),

$$\mathbf{U}\Sigma\mathbf{V}^T = \text{SVD}(\mathbf{D}^T\mathbf{D}) \quad (4.5)$$

can now be performed that provides the best column rank approximation with respect to the quadratic error. The desired normal vector  $\mathbf{n} = [n_x \ n_y \ n_z]^T$  then corresponds to the last column of the matrix  $\mathbf{V}$ . Finally, the orthogonal distance of the searched plane to the origin can be indicated by

$$d = \frac{\sum (n_x x_i + n_y y_i + n_z z_i)}{n}. \quad (4.6)$$

**IMPORTANT REMARK:** Usually, the determination of plane parameters by [SVD](#) is to be considered as an approximate solution, since the coordinates are assumed to be equally weighted and uncorrelated quantities. However, during the split and merge process, a rigorous solution as in [Section 2.4.2](#) was intentionally omitted for the sake of performance. Comparisons revealed no significant differences in the detection of planes between the two methods. For the use of the computed plane parameters as observables with corresponding [SD](#) for the subsequent matching procedure in [Section 4.4](#), as well as for the registration process in [Section 4.5](#), a rigorous solution is finally computed according to [Section 2.4.2](#).

For the assessment of a plane defined according to [Equation \(4.1\)](#) with regard to the further splitting into sub-segments, the individual absolute point distances

$$\epsilon_i = |n_x x_i + n_y y_i + n_z z_i - d| \quad (4.7)$$

to the adjusted plane are compared against their respective precision

$$\sigma_{\epsilon_i} = n_x^2 \sigma_{x_i}^2 + n_y^2 \sigma_{y_i}^2 + n_z^2 \sigma_{z_i}^2 \text{ }^5. \quad (4.8)$$

In so far a plane is defined as a plane if 95% of the set of all point distances  $\epsilon_i$  in the sense of error distribution does not exceed its individual precision  $\sigma_{\epsilon_i}$  by a factor of three ( $3\sigma$  rule). The 95% error interval was chosen due to [BAARDA's data snooping \(BAARDA 1968\)](#) capability of 5%, which is applied in the ensuing rigorous plane adjustment according to [Section 2.4.2](#).

<sup>5</sup> The precision of the point distances  $\sigma_{\epsilon_i}$  is calculated from the adjusted plane parameters and the individual point accuracies  $\sigma_{x_i}$ ,  $\sigma_{y_i}$ ,  $\sigma_{z_i}$  based on the stochastic model from [Chapter 2](#) using [VCP](#), see also [Section 2.3.1](#) and especially [Equation \(2.13\)](#).

Further secondary criteria can be defined by a minimum number of points and a minimum segment width and height. The `function` `predicate_is_plane`<sup>6</sup>, see [Code 4.1](#), checks a region for planarity using the additional criteria (lines 11 to 14) and the main criterion (line 34) and returns `plane_flag=true` in case of confirmation otherwise `false`.

Code 4.1: Definition of a planar region.

```

1 function plane_flag = predicate_is_plane(region)
2
3 % get parameters of the region
4 w = region.bbox(1,3);
5 h = region.bbox(1,4);
6 r_p = region.plane_params;
7 r_ids = region.point_indices;
8 num_elements = numel(r_ids);
9
10 % define plane
11 if num_elements < 100 || w < 3 || h < 3
12     plane_flag = 0;
13     return;
14 end
15
16 % points and precision
17 p_3D = [p_cloud.x(r_ids),p_cloud.y(r_ids),p_cloud.z(r_ids)];
18 p_3D_p = [p_cloud.s_x(r_ids),p_cloud.s_y(r_ids),p_cloud.s_z(r_ids)];
19
20 % point to plane distances and precision
21 plane_dists = abs(r_p(1,1).*p_3D(:,1)+r_p(2,1).*p_3D(:,2)...
22 +r_p(3,1).*p_3D(:,3)-r_p(4,1));
23 plane_dist_acc = sqrt((r_p(1,1).*p_3D_p(:,1)).^2 ...
24 +(r_p(2,1).*p_3D_p(:,2)).^2+(r_p(3,1).*p_3D_p(:,3)).^2);
25
26 % threshold parameters for defining a adequate plane
27 perc_thresh = 0.95;
28
29 % gets plane distances smaller 3 times their sd
30 p_dists_in_sig_range = plane_dists < 3*plane_dist_acc;
31
32 % percentage of inliers
33 perc = nnz(p_dists_in_sig_range)/numel(p_dists_in_sig_range);
34
35 % is plane if min. 95% are inlier
36 plane_flag = perc_thresh <= perc;

```

The [Code 4.2](#) excerpt shows the main process of splitting and merging a region of the `function` `split_merge`, see [Code A.13](#), which is divided into the following 4 steps:

<sup>6</sup> Is used by `function` `predicate_split`, see [Code A.9](#), and `merge_regions`, see [Code A.12](#) in [Section A.3](#)

1. Lines 4 to 14: try to divide main region, see [function split\\_region \(Code A.11\)](#), if region is not dividable, then check by [function predicate\\_is\\_plane](#), see [Code 4.1](#), if the region is already planar
2. Lines 21 to 36: successive splitting of all subregions until they can be assigned to a planar region in `plane_regions` according to `predicate_is_plane`
3. Lines 38 to 48: try to merge existing subregions that may have been unfavourably divided by the [function merge\\_regions](#), see [Code A.12](#)
4. Line 17 and 51: Repeat steps 2. and 3. until all subregions have been assigned (Line 49, `subregions = []`).

Code 4.2: Region splitting and merging.

```

1 %% region splitting and merging
2
3 % try to split main region first
4 subregions = split_region(region,@predicate_split);
5
6 if isempty(subregions)
7     flag_is_plane = predicate_is_plane(region);
8     if flag_is_plane
9         plane_regions(1) = region;
10    else
11        plane_regions = [];
12    end
13    return;
14 end
15
16 n_plane_regions = 0;
17 while ~isempty(subregions)
18
19     n_sub_subregions = 0;
20     n_subregions = numel(subregions);
21     for i = 1:n_subregions
22         % try to split subregions
23         subregions_tmp = split_region(subregions(i),@predicate_split);
24         if isempty(subregions_tmp)
25             plane_flag = predicate_is_plane(subregions(i));
26             if plane_flag
27                 plane_regions(num_plane_regions+1) = subregions(i);
28                 n_plane_regions = num_plane_regions+1;
29             end
30         else
31             for j = 1:numel(subregions_tmp)
32                 sub_subregions(n_sub_subregions+j) = subregions_tmp(j);
33             end
34             n_sub_subregions = numel(sub_subregions);
35         end
36     end
37
38     if exist('sub_subregions','var')
39         subregions = sub_subregions;

```

```

40     clear 'sub_subregions' 'n_sub_subregions';
41     % try to merge some subregions
42     [merged_plane_regions, subregions] = merge_regions(...
43     subregions,@predicate_merge);
44     if ~isempty(merged_plane_regions)
45         for j = 1:numel(merged_plane_regions)
46             plane_regions(n_plane_regions+1) = ...
47             merged_plane_regions(j);
48             n_plane_regions = num_plane_regions+1;
49         end
50     end
51     else
52         subregions = [];
53     end
54 end
55
56 if ~exist('plane_regions','var')
57     plane_regions = [];
58 end

```

The division into regions by applying the `split_region`<sup>7</sup> function, see [Code A.11](#), is based on the segment width  $w$  and height  $h$ , always dividing by the longest segment side. In addition, the

$$\text{ratio} = \frac{n}{wh} \quad (4.9)$$

of the content (number of points  $n$ ) to the segment size ( $wh$ ) is taken into account, ensuring the best possible ratio after splitting for all subregions.

The merge of previously unfavorably subdivided regions is attempted by the `merge_regions` function, see [Code A.12](#), which verifies whether neighboring regions represent the same plane according to `predicate_merge`, see [Code A.10](#). Hence, adjacent planes are identical if they border each other to a certain extent and have a significantly similar orientation in space (plane normal). This is the case if the angle

$$\angle(\mathbf{n}_1, \mathbf{n}_2) = \arccos \frac{\langle \mathbf{n}_1 | \mathbf{n}_2 \rangle}{\|\mathbf{n}_1\| \cdot \|\mathbf{n}_2\|} \quad (4.10)$$

between the plane normals  $\mathbf{n}_1$  and  $\mathbf{n}_2$  is below a defined threshold value. A merge attempt is thus performed according to [Code A.10](#) if the angle  $\angle(\mathbf{n}_1, \mathbf{n}_2)$  of the normal vectors of two adjacent planes is less than 0.02 radians, i. e. about 1 degree.

[Figure 4.6](#) shows the behavior of the "split and merge" algorithm in [Code A.13](#) of successive splitting into planar subregions based on an initial region.

<sup>7</sup> Uses function `predicate_split`, see [Code A.9](#) in [Section A.3](#), to decide whether to divide

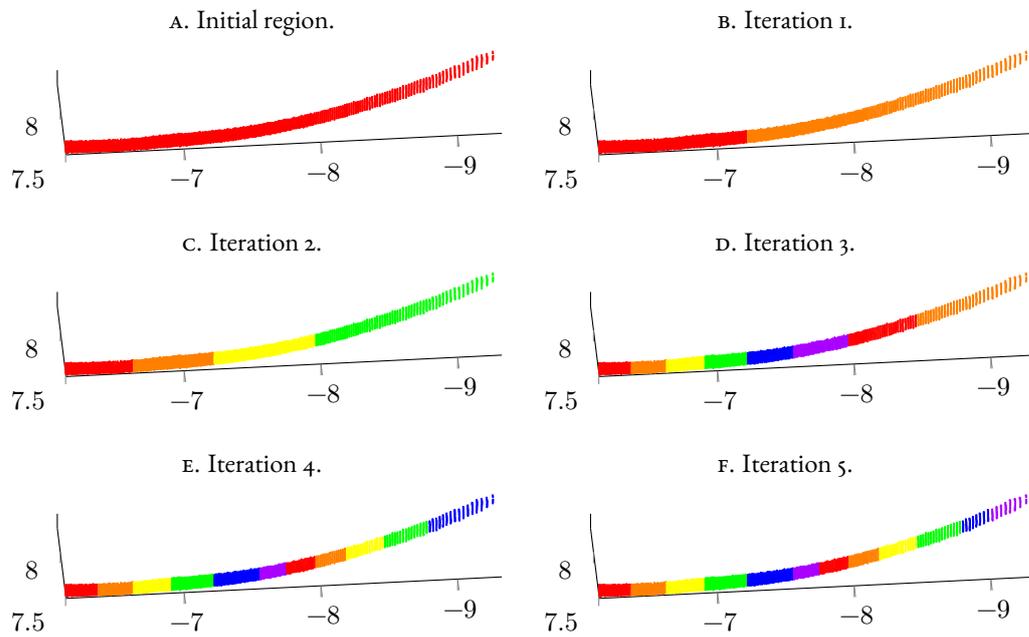


Figure 4.6: Computed information layers regarding curvature, point spacing and change in intensity.

The result of `function split_merge`, see [Code A.13](#), is a structure `plane_regions`, which is based on the structure `regions` in [Figure 3.15](#), see [Figure 4.7](#).

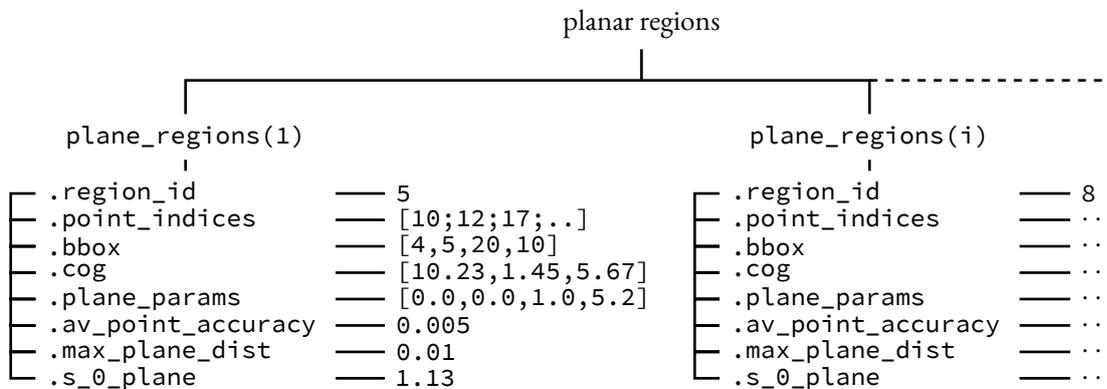


Figure 4.7: Representation of a regions structure array in `MatLab`.

In the structure `plane_regions` the `region_id` guarantees the clear assignment to the region from which the respective plane is derived, that is of relevance for the region merging in [Section 4.6](#). Furthermore, for the matching process in [Section 4.4](#) the centroid `cog`, the plane parameters `plane_params`

and quality criteria of the plane, like the mean point accuracy of points on the plane `av_point_acc` or the maximum plane distance `max_plane_dist` are stored.

**CONCLUDING REMARK:** Ordinary *split and merge* methods, such as the one used in (GIELSDORF 2009), are applied to an entire dataset, thus the successive splitting without prior information results in segments that do not extend to the geometry boundaries of the dataset. The distinctive feature of the developed *split and merge* method is that the subdivision of the segments relies on the pre-segmentation in Chapter 3, see also Figure 4.4, and thus follows the present geometry in the data set reaching up to the geometry transitions (segment boundaries).

#### 4.4 MATCHING PROCESS

On the basis of the planes found in Section 4.3.2 for each point cloud of the three standpoints, see also Figure 4.5, correspondences between the planes are searched for each pair of standpoints. The matching procedure is a three-step procedure and includes the

- initial determination of the rotation matrix  $\mathbf{R}^{08}$ ,
- initial determination of the translation vector  $\mathbf{t}^0$ ,
- improvement of  $\mathbf{R}$ ,  $\mathbf{t}$  and the matching via a rigorous adjustment in a GHM

between the two local coordinate systems of adjacent standpoints. For two identical planes of a standpoint pair in the defined source system  $s$  and target system  $t$ , the equations read as follows

$$\begin{aligned} \mathbf{n}_s \mathbf{x}_{s_i} - d_s &= 0 \\ \mathbf{n}_t \mathbf{x}_{t_i} - d_t &= 0 \end{aligned} \tag{4.11}$$

where

- $\mathbf{n}_s$  and  $\mathbf{n}_t$  are the normal vectors of the plane pair
- $d_s$  and  $d_t$  are the perpendicular distances of the planes from the coordinate origin
- $\mathbf{x}_{s_i}$  and  $\mathbf{x}_{t_i}$  are single points on the planes in the source and target system satisfying the equations.

<sup>8</sup> It is recommended to use rotation matrices with quaternions, see e. g. (LUHMANN 2018, p. 63 ff.) and Section 1.6.2, as these allow fast convergence in an adjustment, see also Section 1.8, with low demands on the quality of the approximate values.

For the initial determination of the rotation between both systems, expressed by the rotation matrix  $\mathbf{R}^0$ , with

$$\mathbf{R}^0 \mathbf{n}_s - \mathbf{n}_t = \mathbf{0}^9 \quad (4.12)$$

a direct relationship between the normal vectors  $\mathbf{n}_s$  and  $\mathbf{n}_t$  of the plane pair of Equation (4.11) can be established. Moreover, if one assumes levelled TLS during the acquisition the determination of  $\mathbf{R}^0$  using (4.12) is reduced to the 2D case. Thus a single rotation  $\kappa$  around the standing axis, see Figure 1.2 B., of the TLS can be assumed, which corresponds to a rotation around the Z-axis, see also Section 1.6.1  $\mathbf{R}_\kappa$ , resulting in

$$\mathbf{R}^0 = \mathbf{R}_Z = \begin{bmatrix} \cos \kappa & -\sin \kappa & 0 \\ \sin \kappa & \cos \kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.13)$$

With the substitutions  $a^0 = \cos \kappa$  and  $o^0 = \sin \kappa$  two linear *conditional equations*

$$\begin{aligned} a^0 n_{s_x} - o^0 n_{s_y} - n_{t_x} &= 0 \\ o^0 n_{s_x} + a^0 n_{s_y} - n_{t_y} &= 0 \end{aligned} \quad (4.14)$$

are obtained from (4.12). Based on two correspondences  $(n_{s_x} \ n_{s_y} | n_{t_x} \ n_{t_y})$ , the transformation parameters  $a$  and  $o$  can be easily determined via the linear equation system

$$\begin{bmatrix} n_{t_x} \\ n_{t_y} \end{bmatrix} = \begin{bmatrix} n_{s_x} & -n_{s_y} \\ n_{s_x} & n_{s_y} \end{bmatrix} \begin{bmatrix} a^0 \\ o^0 \end{bmatrix}. \quad (4.15)$$

In an iterative procedure, appropriate normal vector pairs can be used to determine  $\mathbf{R}^0$ , the most likely solution being the one with the most correspondences. In order to limit the enormous number of possible plane pairs due to the large number of planes found, see also Table 4.1, and to increase efficiency<sup>10</sup>, planes on the floor and ceiling are excluded as they are not relevant for determining the rotation around the Z-axis anyway. A faster and better way to determine the rotation parameters is to systematically change the angle  $\kappa$  and apply the resulting *rotation matrix*, see Equation (4.13), to the normals in the source system and find corresponding normals in the target system for each iteration step.

<sup>9</sup> The normal vector of a plane in the source system  $\mathbf{n}_s$  rotated by  $\mathbf{R}^0$  must correspond to the equivalent normal vector  $\mathbf{n}_t$  of the plane in the target system.

<sup>10</sup> For the planes found of standpoint 1, see Table 4.1, there are more than 31 million correspondence possibilities to the planes of standpoint 2.

Code 4.3 shows the determination of the rotation parameters, according to which the best rotation corresponds to the one with the most correspondences. In the first pass, the rotation angle  $\kappa$  is found with a precision of the initial step width of one degree, whereupon a new interval is finally built around this value with a step width smaller by a power of ten, thus improving the precision. The process is repeated twice until a precision one hundredth of a degree is reached.

Code 4.3: Determine initial rotation for matching planes.

```

1 %input: planes_source planes_target
2 %output: rotation parameters a, o
3
4 % exclude planes on the floor and ceiling
5 ind_source = find(abs(planes_source(3,:)) < 0.5);
6 ind_target = find(abs(planes_target(3,:)) < 0.5);
7
8 n_s = planes_source(1:3,ind_source);
9 n_t = planes_target(1:3,ind_target);
10 n_n_s = numel(ind_source);
11
12 %threshold for normal matching
13 n_thr = 0.001;
14
15 corresp_struct = zeros(360,3);
16 for i = 1:360
17     kappa_tmp = pi()*(i/180);
18     a = cos(kappa_tmp);
19     o = sin(kappa_tmp);
20     R_tmp = [a -o 0;o a 0;0 0 1];
21     n_st = R_tmp*n_s;
22     for j=1:n_n_s
23         i_nx = n_t(1,:) < n_st(1,j)+n_thr & n_t(1,:) > n_st(1,j)-n_thr;
24         i_ny = n_t(2,:) < n_st(2,j)+n_thr & n_t(2,:) > n_st(2,j)-n_thr;
25         i_nz = n_t(3,:) < n_st(3,j)+n_thr & n_t(3,:) > n_st(3,j)-n_thr;
26     end
27     i_all = i_nx & i_ny & i_nz;
28     corresp_struct(i,:) = [a,o,nnz(i_all)];
29 end

```

To determine the initial translation, it is assumed that the foot point  $\mathbf{x}_{F_s} = \mathbf{n}_s d_s$  of the plane in the source system transformed with

$$\mathbf{x}_{F_t} = \mathbf{R}^0 \mathbf{x}_{F_s} + \mathbf{t} \quad (4.16)$$

into the target system should correspond to the foot point  $\mathbf{x}_{F_t} = \mathbf{n}_t d_t$  of the plane in the target

system. Taking Equation (4.11) and Equation (4.12) into account, this leads to a further relationship

$$\mathbf{d}_s + \mathbf{n}_t \mathbf{t}^0 - \mathbf{d}_t = 0, \quad (4.17)$$

in order to be able to determine the translation  $\mathbf{t}^0$  between the local systems. On the basis of respectively three correspondences from the totality of all found plane correspondences for the determination of the rotation matrix  $\mathbf{R}^0$ , translation parameters can now be determined by solving the linear equation system

$$\begin{bmatrix} \mathbf{d}_{t_1} - \mathbf{d}_{s_1} \\ \mathbf{d}_{t_2} - \mathbf{d}_{s_2} \\ \mathbf{d}_{t_3} - \mathbf{d}_{s_3} \end{bmatrix} = \begin{bmatrix} \mathbf{n}_{t_{x_1}} & \mathbf{n}_{t_{y_1}} & \mathbf{n}_{t_{z_1}} \\ \mathbf{n}_{t_{x_2}} & \mathbf{n}_{t_{y_2}} & \mathbf{n}_{t_{z_2}} \\ \mathbf{n}_{t_{x_3}} & \mathbf{n}_{t_{y_3}} & \mathbf{n}_{t_{z_3}} \end{bmatrix} \begin{bmatrix} t_x^0 \\ t_y^0 \\ t_z^0 \end{bmatrix} \quad (4.18)$$

based on (4.17), where the indices 1 to 3 indicate three correspondences between the planes in the source and target system. The most probable translation  $\mathbf{t}^0$  is again the one that results from the majority of correspondences. The function `translation_from_3_corresp` shows the calculation of the translation parameters  $\mathbf{T}$  from 3 plane correspondences.

Code 4.4: Determine initial translation for matching planes.

```

1 function T = translation_from_3_corresp(planes_source, planes_target)
2 %input planes_source, planes_target
3 %output translation T
4
5 l = (planes_target(4,:)-planes_source(4,:))';
6
7 A = [planes_target(1,:) planes_target(2,:) planes_target(3,:)'];
8
9 T = A \ l;

```

Table 4.2 shows the initial transformation parameters for the scan pairs of positions 1 with 2 and 10 with 1, determined according to the procedures described above.

Table 4.2: Listing of initial transformation parameters.

SCAN PAIR	INITIAL TRANSFORMATION PARAMETERS					
	$\omega^0$ [°]	$\phi^0$ [°]	$\kappa^0$ [°]	$t_x^0$ [m]	$t_y^0$ [m]	$t_z^0$ [m]
2 to 1	0	0	32.0290	6.9438	7.1909	0.0761
10 to 1	0	0	-41.2465	0.4941	-14.0535	0.1659

Taking the initially found transformation parameters between the respective scan pairs as a basis, using `function` `match_planes_by_datum`<sup>ii</sup>, a final, so-called date-dependent matching can be carried out, providing the finite plane correspondences for the following calculation step.

Code 4.5: Matching of planes by known datum in case of transformation parameters.

```

1 function matches = match_planes_by_datum(p_t, p_s, R, T, m_thr)
2 %input: - structs p_t, p_s of target and source planes
3 %       - R, T the transformation parameters
4 %       - m_thr(1:3,1) a matching threshold array
5 %output: - matches(n,2) of n matches for a scanpair, where
6 %         matches(i,1) is target and matches(i,2) source
7
8 % matching thresholds
9 n_t = m_thr(1,1); %change in plane normal
10 d_t = m_thr(2,1); %change in plane dist to origin
11 d_cogs_t = m_thr(3,1); %max dist for matching cogs
12
13 % extract necessary plane information from struct
14 p_t_cog = reshape(extractfield(p_t, 'cog'),3,[]);
15 p_s_cog = reshape(extractfield(p_s, 'cog'),3,[]);
16 p_t = reshape(extractfield(p_t, 'plane_params'),8,[]);
17 p_s = reshape(extractfield(p_s, 'plane_params'),8,[]);
18
19 % transform planes from source to target system with initial R, T
20 p_s_t = transform_planes(p_s,R,T);
21
22 % transform pcogs from source to target system with initial R, T
23 p_s_cogs_t = transform_points(p_s_cog,R,T);
24
25 %% find for transformed planes corresponding planes in target system
26 matches = [];
27
28 for i = 1:numel(p_s_t(1,:))
29     % component nx ny nz within thresholds
30     i_nx = p_t(1,:) < p_s_t(1,i)+n_t & p_t(1,:) > p_s_t(1,i)-n_t;
31     i_ny = p_t(2,:) < p_s_t(2,i)+n_t & p_t(2,:) > p_s_t(2,i)-n_t;
32     i_nz = p_t(3,:) < p_s_t(3,i)+n_t & p_t(3,:) > p_s_t(3,i)-n_t;
33     % component d within thresholds
34     i_d = p_t(4,:) < p_s_t(4,i)+d_t & p_t(4,:) > p_s_t(4,i)-d_t;
35
36     i_all = i_nx & i_ny & i_nz & i_d;
37
38     m_i = find(i_all);
39     % for all found planes get distance from cogs
40     if ~isempty(m_i)

```

<sup>ii</sup> Uses `function` `transform_planes` which performs the transformation of plane parameters to another coordinate system, see [Code A.15](#), and `function` `transform_points` for equivalently transforming points, see also [Code A.1](#)

```

41     n_matches = numel(m_i);
42     d_cogs = zeros(n_matches,1);
43     for j = 1:n_matches
44         dX = p_s_cogs_t(:,i)-p_t_cog(:,m_i(1,j));
45         d_cogs(j,1) = sqrt(dX(1,1).^2+dX(2,1).^2+dX(3,1).^2);
46     end
47     i_cog_m = find(d_cogs < d_cogs_t);
48     if ~isempty(i_cog_m)
49         matches_t = [m_i(1,i_cog_m)',ones(numel(i_cog_m),1)*i];
50         matches = [matches;matches_t];
51     end
52 end
53 end

```

Based on the condition [Equations \(4.12\)](#) and [\(4.17\)](#), a rigorous adjustment can be carried out in a [GHM](#) to reveal and eliminate erroneous correspondences<sup>12</sup> and hence improve the transformation parameters.

Introducing the residuals  $\mathbf{v}_{n_s} = [v_{n_{s_x}} \ v_{n_{s_y}} \ v_{n_{s_z}}]^T$ ,  $\mathbf{v}_{d_s}$  and  $\mathbf{v}_{n_t} = [v_{n_{t_x}} \ v_{n_{t_y}} \ v_{n_{t_z}}]^T$ ,  $\mathbf{v}_{d_t}$  for the observation pairs  $\mathbf{n}_s = [n_{s_x} \ n_{s_y} \ n_{s_z}]^T$ ,  $\mathbf{d}_s$  and  $\mathbf{n}_t = [n_{t_x} \ n_{t_y} \ n_{t_z}]^T$ ,  $\mathbf{d}_t$ , four conditions

$$\begin{aligned}
 (\mathbf{d}_s + \mathbf{v}_{d_s}) + (\mathbf{n}_{t_x} + \mathbf{v}_{n_{t_x}})\hat{t}_x + (\mathbf{n}_{t_y} + \mathbf{v}_{n_{t_y}})\hat{t}_y + (\mathbf{n}_{t_z} + \mathbf{v}_{n_{t_z}})\hat{t}_z - (\mathbf{d}_t + \mathbf{v}_{d_t}) &= 0 \\
 \hat{r}_{11}(\mathbf{n}_{s_x} + \mathbf{v}_{n_{s_x}}) + \hat{r}_{12}(\mathbf{n}_{s_y} + \mathbf{v}_{n_{s_y}}) + \hat{r}_{13}(\mathbf{n}_{s_z} + \mathbf{v}_{n_{s_z}}) - (\mathbf{n}_{t_x} + \mathbf{v}_{n_{t_x}}) &= 0 \\
 \hat{r}_{21}(\mathbf{n}_{s_x} + \mathbf{v}_{n_{s_x}}) + \hat{r}_{22}(\mathbf{n}_{s_y} + \mathbf{v}_{n_{s_y}}) + \hat{r}_{23}(\mathbf{n}_{s_z} + \mathbf{v}_{n_{s_z}}) - (\mathbf{n}_{t_y} + \mathbf{v}_{n_{t_y}}) &= 0 \\
 \hat{r}_{31}(\mathbf{n}_{s_x} + \mathbf{v}_{n_{s_x}}) + \hat{r}_{32}(\mathbf{n}_{s_y} + \mathbf{v}_{n_{s_y}}) + \hat{r}_{33}(\mathbf{n}_{s_z} + \mathbf{v}_{n_{s_z}}) - (\mathbf{n}_{t_z} + \mathbf{v}_{n_{t_z}}) &= 0
 \end{aligned} \tag{4.19}$$

emerge from each plane match of the source system  $s$  to the target system  $t$ , where

- $\hat{t}_x, \hat{t}_y, \hat{t}_z$  are the unknown adjusted values of the components of the translation vector  $\hat{\mathbf{t}}$ , as well as
- $\hat{r}_{11}, \hat{r}_{12}, \dots, \hat{r}_{33}$  the unknown adjusted elements of the *rotation matrix*  $\hat{\mathbf{R}}$ , expressed by the *quaternions*  $\hat{\mathbf{q}} = [\hat{q}_0 \ \hat{q}_x \ \hat{q}_y \ \hat{q}_z]^T$  to be determined, see also [Equation \(1.24\)](#).

From the rotation matrix  $\mathbf{R}^0$  initially found, approximate values for the rotation parameters can be derived in the form of quaternions  $\mathbf{q}^0 = [q_0^0 \ q_x^0 \ q_y^0 \ q_z^0]^T$ , to be used together with the approximate values of the translation parameters  $\mathbf{t}^0 = [t_x^0 \ t_y^0 \ t_z^0]^T$  and the original residuals  $v_{n_{s_x}}^0, v_{n_{s_y}}^0, v_{n_{s_z}}^0, v_{d_s}^0$

<sup>12</sup>Neighboring planes in different scans may lead to several matches among each other within the matching thresholds, possibly including matches that are not so favorable for registration.

as well as  $v_{n_{t_x}}^0, v_{n_{t_y}}^0, v_{n_{t_z}}^0, v_{d_t}^0$  initialized with 0, yielding the approximate *condition equations*

$$\Psi_k^0 = \begin{bmatrix} (d_s + v_{d_s}^0) + (n_{t_x} + v_{n_{t_x}}^0)t_x^0 + (n_{t_y} + v_{n_{t_y}}^0)t_y^0 + (n_{t_z} + v_{n_{t_z}}^0)t_z^0 - (d_t + v_{d_t}^0) \\ r_{11}^0(n_{s_x} + v_{n_{s_x}}^0) + r_{12}^0(n_{s_y} + v_{n_{s_y}}^0) + r_{13}^0(n_{s_z} + v_{n_{s_z}}^0) - (n_{t_x} + v_{n_{t_x}}^0) \\ r_{21}^0(n_{s_x} + v_{n_{s_x}}^0) + r_{22}^0(n_{s_y} + v_{n_{s_y}}^0) + r_{23}^0(n_{s_z} + v_{n_{s_z}}^0) - (n_{t_y} + v_{n_{t_y}}^0) \\ r_{31}^0(n_{s_x} + v_{n_{s_x}}^0) + r_{32}^0(n_{s_y} + v_{n_{s_y}}^0) + r_{33}^0(n_{s_z} + v_{n_{s_z}}^0) - (n_{t_z} + v_{n_{t_z}}^0) \end{bmatrix} \quad (4.20)$$

assuming  $n$  plane matches, with  $\{k = 1 : n\}$  based on (4.19). The entire *condition vector*

$$\Psi^0 = [\Psi_1^0 \quad \dots \quad \Psi_k^0 \quad \dots \quad \Psi_n^0]^T \quad (4.21)$$

is therefore composed of the vectors  $\Psi_k^0$  from the individual plane matches. For a plane match  $k$  with respect to (4.20), the initial residuals can be summarized in a vector

$$\mathbf{v}_k^0 = [v_{n_{s_x}}^0 \quad v_{n_{s_y}}^0 \quad v_{n_{s_z}}^0 \quad v_{d_s}^0 \quad | \quad v_{n_{t_x}}^0 \quad v_{n_{t_y}}^0 \quad v_{n_{t_z}}^0 \quad v_{d_t}^0]^T, \quad (4.22)$$

forming the entire vector of residuals

$$\mathbf{v}^0 = [\mathbf{v}_1^0 \quad \dots \quad \mathbf{v}_k^0 \quad \dots \quad \mathbf{v}_n^0]^T \quad (4.23)$$

for  $n$  matches. According to the order of the residual vector  $\mathbf{v}^0$  in (4.23), the precision relations of the observations are given by the choice of the *SD of the unit weight* of  $\sigma_0 = 0.01$  in the *CM* of observations

$$\mathbf{Q}_{ll} = \text{diag} [Q_{ll_1} \quad \dots \quad Q_{ll_k} \quad \dots \quad Q_{ll_n}] \quad (4.24)$$

with the individual *CMs* of observations

$$\mathbf{Q}_{ll_k} = \text{diag} [Q_{ll_s} \quad | \quad Q_{ll_t}] \quad (4.25)$$

for each plane match  $k$ , containing the respective *CMs* of observations

$$\mathbf{Q}_{ll_s} = \frac{s_0^2}{\sigma_0^2} \mathbf{Q}_{x \times s} \quad \text{and} \quad \mathbf{Q}_{ll_t} = \frac{s_0^2}{\sigma_0^2} \mathbf{Q}_{x \times t}^{13}, \quad (4.26)$$

<sup>13</sup> The precision of the plane components stems from the plane adjustment of the plane segmentation procedure in Section 4.3.2, i. e. the adjustment after Section 2.4.2, and is based on the individual point precisions derived from the proposed stochastic model of the *TLS* as already used previously in Section 2.3.1.

for the plane matches  $k$  in the source system  $s$  as well the target system  $t$ , using the corresponding **CMs** of adjusted unknowns  $\mathbf{Q}_{xx_s}$ ,  $\mathbf{Q}_{xx_t}$  and the *empirical reference variances*  $s_{0_s}^2$ ,  $s_{0_t}^2$ , computed after each rigorous plane adjustment following [Section 2.4.2](#) via [Equation \(1.82\)](#).

[Equations \(4.23\)](#) and [\(4.24\)](#) result in the minimization of the *objective function* under the secondary condition (*constraint vector*)

$$\gamma^0 = q_0^0{}^2 + q_x^0{}^2 + q_y^0{}^2 + q_z^0{}^2 - 1 \quad (4.27)$$

with respect to the rotation parameters according to ??, ideally providing an iterative solution to the problem of [Equation \(4.21\)](#) in a **GHM** with *constraints* between the unknowns, see also [Section 1.8.7](#). The *Jacobian matrix* with the partial derivatives of the *condition equations* in [Equation \(2.21\)](#) according to the unknowns

$$\mathbf{J}_1 = \left[ \mathbf{J}_{1_1} \quad \cdots \quad \mathbf{J}_{1_k} \quad \cdots \quad \mathbf{J}_{1_n} \right]^T \quad (4.28)$$

is obtained with

$$\mathbf{J}_{1_k} = \left[ \frac{\partial \Psi_k^0}{\partial q_0^0} \quad \frac{\partial \Psi_k^0}{\partial q_x^0} \quad \frac{\partial \Psi_k^0}{\partial q_y^0} \quad \frac{\partial \Psi_k^0}{\partial q_z^0} \quad \frac{\partial \Psi_k^0}{\partial t_x^0} \quad \frac{\partial \Psi_k^0}{\partial t_y^0} \quad \frac{\partial \Psi_k^0}{\partial t_z^0} \right] \quad (4.29)$$

for each plane match  $k$  based on [Equation \(4.20\)](#), where

$$\frac{\partial \Psi_k^0}{\partial t_x^0} = \begin{bmatrix} n_{t_x} + v_{n_{t_x}}^0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \frac{\partial \Psi_k^0}{\partial t_y^0} = \begin{bmatrix} n_{t_y} + v_{n_{t_y}}^0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \frac{\partial \Psi_k^0}{\partial t_z^0} = \begin{bmatrix} n_{t_z} + v_{n_{t_z}}^0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and

$$\frac{\partial \Psi_k^0}{\partial q_0^0} = \begin{bmatrix} 0 \\ 2q_0^0(n_{s_x} + v_{n_{s_x}}^0) - 2q_z^0(n_{s_y} + v_{n_{s_y}}^0) + 2q_y^0(n_{s_z} + v_{n_{s_z}}^0) \\ 2q_z^0(n_{s_x} + v_{n_{s_x}}^0) + 2q_0^0(n_{s_y} + v_{n_{s_y}}^0) - 2q_x^0(n_{s_z} + v_{n_{s_z}}^0) \\ -2q_y^0(n_{s_x} + v_{n_{s_x}}^0) + 2q_x^0(n_{s_y} + v_{n_{s_y}}^0) + 2q_0^0(n_{s_z} + v_{n_{s_z}}^0) \end{bmatrix}$$

$$\begin{aligned}
 \frac{\partial \Psi_k^0}{\partial q_x^0} &= \begin{bmatrix} 0 \\ 2q_x^0(n_{s_x} + v_{n_{s_x}}^0) + 2q_y^0(n_{s_y} + v_{n_{s_y}}^0) + 2q_z^0(n_{s_z} + v_{n_{s_z}}^0) \\ 2q_y^0(n_{s_x} + v_{n_{s_x}}^0) - 2q_x^0(n_{s_y} + v_{n_{s_y}}^0) - 2q_0^0(n_{s_z} + v_{n_{s_z}}^0) \\ 2q_z^0(n_{s_x} + v_{n_{s_x}}^0) + 2q_0^0(n_{s_y} + v_{n_{s_y}}^0) - 2q_x^0(n_{s_z} + v_{n_{s_z}}^0) \end{bmatrix} \\
 \frac{\partial \Psi_k^0}{\partial q_y^0} &= \begin{bmatrix} 0 \\ -2q_y^0(n_{s_x} + v_{n_{s_x}}^0) + 2q_x^0(n_{s_y} + v_{n_{s_y}}^0) + 2q_0^0(n_{s_z} + v_{n_{s_z}}^0) \\ 2q_x^0(n_{s_x} + v_{n_{s_x}}^0) + 2q_y^0(n_{s_y} + v_{n_{s_y}}^0) + 2q_z^0(n_{s_z} + v_{n_{s_z}}^0) \\ -2q_0^0(n_{s_x} + v_{n_{s_x}}^0) + 2q_z^0(n_{s_y} + v_{n_{s_y}}^0) - 2q_y^0(n_{s_z} + v_{n_{s_z}}^0) \end{bmatrix} \\
 \frac{\partial \Psi_k^0}{\partial q_z^0} &= \begin{bmatrix} 0 \\ -2q_z^0(n_{s_x} + v_{n_{s_x}}^0) - 2q_0^0(n_{s_y} + v_{n_{s_y}}^0) + 2q_x^0(n_{s_z} + v_{n_{s_z}}^0) \\ 2q_0^0(n_{s_x} + v_{n_{s_x}}^0) - 2q_z^0(n_{s_y} + v_{n_{s_y}}^0) + 2q_y^0(n_{s_z} + v_{n_{s_z}}^0) \\ 2q_x^0(n_{s_x} + v_{n_{s_x}}^0) + 2q_y^0(n_{s_y} + v_{n_{s_y}}^0) + 2q_z^0(n_{s_z} + v_{n_{s_z}}^0) \end{bmatrix}.
 \end{aligned}$$

With the partial derivatives of the *conditional equations* concerning a plane match  $k$ , see [Equation \(4.20\)](#), with respect to the residuals

$$\mathbf{J}_{2_k} = \begin{bmatrix} 0 & 0 & 0 & 1 & t_x^0 & t_y^0 & t_z^0 & -1 \\ r_{11}^0 & r_{12}^0 & r_{13}^0 & 0 & -1 & 0 & 0 & 0 \\ r_{21}^0 & r_{22}^0 & r_{23}^0 & 0 & 0 & -1 & 0 & 0 \\ r_{31}^0 & r_{32}^0 & r_{33}^0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}, \quad (4.30)$$

a further *Jacobian matrix*

$$\mathbf{J}_2 = \text{diag} \left[ \mathbf{J}_{2_1} \quad \cdots \quad \mathbf{J}_{2_k} \quad \cdots \quad \mathbf{J}_{2_n} \right] \quad (4.31)$$

results in analogy to the *Jacobian matrix*  $\mathbf{J}_1$ , see [Equation \(4.28\)](#). Finally, the *Jacobian matrix*

$$\mathbf{J}_3 = \begin{bmatrix} 2q_0^0 & 2q_x^0 & 2q_y^0 & 2q_z^0 & 0 & 0 & 0 \end{bmatrix} \quad (4.32)$$

is obtained with the partial derivatives of the *constraint equation* in [Equation \(4.27\)](#) according to the unknowns. Using the *Jacobians* with  $\mathbf{A} = \mathbf{J}_1$ ,  $\mathbf{B} = \mathbf{J}_2$  and  $\mathbf{C} = \mathbf{J}_3$  and forming the *vectors of misclosures*

$$\mathbf{w}_r = \boldsymbol{\gamma}^0 \quad \text{and} \quad \mathbf{w} = -\mathbf{B}\mathbf{v}^0 + \boldsymbol{\Psi}^0 \quad (4.33)$$

according to Equations (1.53) and (1.66) using the *condition* and *constraint vectors* in Equation (4.21) and Equation (4.27), the solution for the vector of reduced unknowns  $\Delta\hat{\mathbf{x}} = [\Delta\hat{q}_0 \ \Delta\hat{q}_x \ \Delta\hat{q}_y \ \Delta\hat{q}_z \ \Delta\hat{t}_x \ \Delta\hat{t}_y \ \Delta\hat{t}_z]^T$  is obtained taking Equation (1.73).

NOTE: Since this is like in Section 2.3 and Section 2.4.2 only the solution of a linearized substitute problem, the unknowns  $\hat{\mathbf{x}}$  and corrections  $\hat{\mathbf{v}}$  based on Equation (1.61) are to be introduced into the adjustment as new approximate values  $\mathbf{x}^0$  and  $\mathbf{v}^0$ . This iterative calculation is carried out until a selected break condition, cf. Equation (1.62), is reached. After each convergence for the actually observation material, BAARDA's *data snooping* (BAARDA 1968), see Section 1.8.8.2, was applied to detect and remove outliers in form of incorrect or improper plane matches.

Figure 4.8 shows the allocated planes of standpoint pairs 2 with 1 (upper image pair) and 1 with 10 (lower image pair), cf. the overview of the standpoints in Figure 4.1.

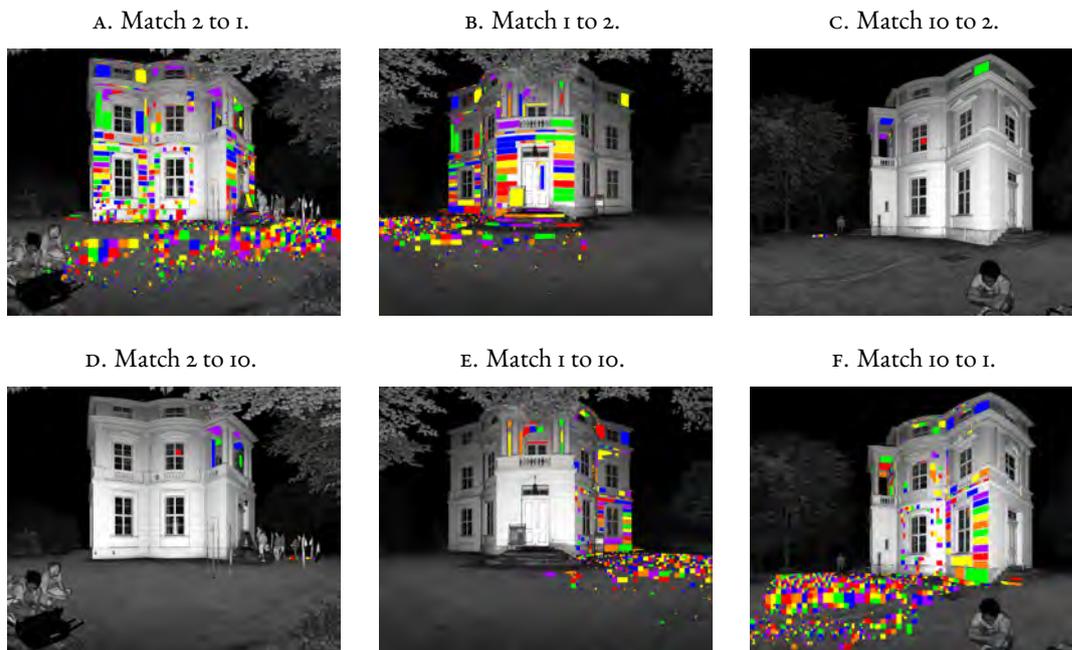


Figure 4.8: Result of plane matching with respect to standpoint pairs 2 with 1 (upper left image pair), 10 with 1 (lower right image pair) and 10 with 2 (upper right and lower left image).

The intermediate matching of the planes with respect to the standpoints 2 and 10 cannot be performed according to the above procedure due to the small overlap of the two scans. On the one hand, however, direct matches can be created from the other two matching pairs via identical IDs for position 1. Assuming, for example, a plane match  $[200, 8]$  for the pair of viewpoints 2 with 1, as well as another

plane match [155, 8] for the pair of viewpoints 10 with 1, where the plane with the ID 200 in scan 2 corresponds to the plane respectively ID 8 in scan 1 and accordingly the IDs 155 with 8 in scan 10, and scan 1, a direct match [155, 200] between the pair of viewpoints 10 with 2 can be specified using the same ID 8 of both matches.

On the other hand, the calculation of the relative transformation parameters between the point cloud of standpoint pair 10 and 2 can be derived from the other two transformations, see Table 4.3, by using the conventions

$$\begin{aligned}\mathbf{R}_{10,2} &= \mathbf{R}_{2,1}^T \mathbf{R}_{10,1} \\ \mathbf{t}_{10,2} &= \mathbf{R}_{2,1}^T (\mathbf{t}_{10,1} - \mathbf{t}_{2,1})\end{aligned}\quad (4.34)$$

where

- $\mathbf{R}_{2,1}, \mathbf{t}_{2,1}$  are the relative transformation parameters from scan 2 to scan 1 and
- $\mathbf{R}_{10,1}, \mathbf{t}_{10,1}$  are the relative transformation parameters from scan 10 to scan 1 as well as
- $\mathbf{R}_{10,2}, \mathbf{t}_{10,2}$  the computed ones describing the transformation between scan 10 and scan 2.

Table 4.3 shows the final results of the matching process described above in the form of the transformation parameters from standpoint to standpoint on the basis of the matched plane pairs of the individual standpoints, Figure 4.8.

Table 4.3: Values and Standard Deviations (SDs) for the *relative transformation parameters* of the scanner standpoint pairs after the matching process of Section 4.4.

SCAN PAIR	MATCHES	RELATIVE TRANSFORMATION PARAMETERS					
		$\hat{\omega}$ [°]	$\hat{\phi}$ [°]	$\hat{\kappa}$ [°]	$\hat{t}_x$ [m]	$\hat{t}_y$ [m]	$\hat{t}_z$ [m]
		$\sigma_\omega$	$\sigma_\phi$	$\sigma_\kappa$	$\sigma_{t_x}$	$\sigma_{t_y}$	$\sigma_{t_z}$
2 to 1	658	0.002062	-0.021111	-32.159455	6.9529	7.1930	0.0175
		0.002175	0.000958	0.000679	0.0014	0.0027	0.0003
10 to 1	650	-0.015842	-0.004972	-41.287382	0.5029	-14.0508	0.1670
		0.001432	0.001677	0.000973	0.0011	0.0021	0.0002

Table 4.3 indicates that the translations  $\hat{t}_x, \hat{t}_y, \hat{t}_z$  between a scan pair of the standpoints were determined with a SD in the millimetre to tenth of a millimetre range, corresponding to the magnitude of the point precision of the TLS applied. The rotation between a scan pair, described by  $\hat{\omega}, \hat{\phi}, \hat{\kappa}$ , have been determined with a SD in the range of a few angular seconds.

CONCLUDING REMARK: The transformation parameters determined in this section, see [Table 4.3](#), represent *relative transformation parameters* between a source and a target coordinate system as in the *spatial similarity transformation* in [Section 1.7](#). In this case between two local *scanner coordinate systems*, see also [Section 1.3](#), of adjacent viewpoints.

Based on the individual calculated transformation parameters among the directly adjacent scan pairs, the `function` `match_planes`<sup>14</sup>, see [Code 4.6](#), can now be used to determine final plane matches among all linked scan pairs, even those that are not directly adjacent.

Code 4.6: Matching of planes between the various adjacent and geometrically overlapping point clouds of individual scan pairs.

```

1 function p_matches = match_planes(p_viewpoint, init_pose, m_pairs)
2 %input: - p_viewpoint{i} cell array of planes,
3 %       where i is the scan number, see order of match_pairs
4 %       - init_pose{i} initial pose between scan pairs
5 %       in the form [R,T], see match_pairs, e.g. from 2 to 1
6 %       - e.g. m_pairs = [1 2;1 3;2 3];
7 %       corresp. to scan overlap matches 1-2, 1-10, 2-10
8 %output: - r_matches of regions in scans
9
10 %% datum dependent matching
11
12 % matching thresholds
13 n_t = 0.005; %change in plane normal
14 d_t = 0.1; %change in plane normal
15 d_cogs_t = 0.3; %max dist for matching cogs
16 m_thr = [n_t;d_t;d_cogs_t];
17
18 n_match_pairs = numel(m_pairs(:,1));
19 p_matches = cell(1,n_match_pairs);
20 for i = 1:n_match_pairs
21     p_t = p_viewpoint{m_pairs(i,1)};
22     p_s = p_viewpoint{m_pairs(i,2)};
23     R = init_pose{1,i}(:,1:3); T = init_pose{1,i}(:,4);
24     p_matches{i} = match_planes_by_datum(p_t,p_s,R,T,m_thr);
25 end

```

Thus, even taking [Equation \(4.34\)](#) into account, the correspondences of the only moderately overlapping *point clouds* can be taken into account with respect to position 2 with 10.

Another possibility to generate further potential matching pairs between only poorly overlapping scans is to consider the same plane indices across different scan pairs. For example, a match in the *point*

<sup>14</sup>Uses `function` `match_planes_by_datum` which performs a matching of planes based on initial transformation parameters, see [Code 4.5](#)

*cloud* pair 1 - 2, which is also a match with the same ID in the *point cloud* pair 1 - 10, indicates that there is also a match with this ID for the pair 2 - 10. Such a direct assignment procedure can be used to complete missing matches, thus adding [Code A.14](#) to the end of the `function` `match_planes`, see [Code 4.6](#). The planes assigned to each other finally serve as initial information for the subsequent chained transformation process in [Section 4.5](#).

#### 4.5 REGISTRATION PROCESS

The planes found in [Section 4.3.2](#) and matched in [Section 4.4](#) serve as identical information for a simultaneous registration of *point clouds* from several standpoints. Similar to a *bundle adjustment* in photogrammetry, where image points associated to several images serve as identical information, a so-called *outer orientation*, see also [Section 1.3.1](#), is determined for each standpoint (scanning position).

Via the respective *exterior orientations*, the *point clouds* of the corresponding scanning positions can finally be transferred to the superordinate coordinate system via e. g. `function` `transform_points`, see [Code A.1](#). [Figure 4.9](#) illustrates the relationship between two local, corresponding planes from two adjacent *point clouds* in relation to the superordinate coordinate system.

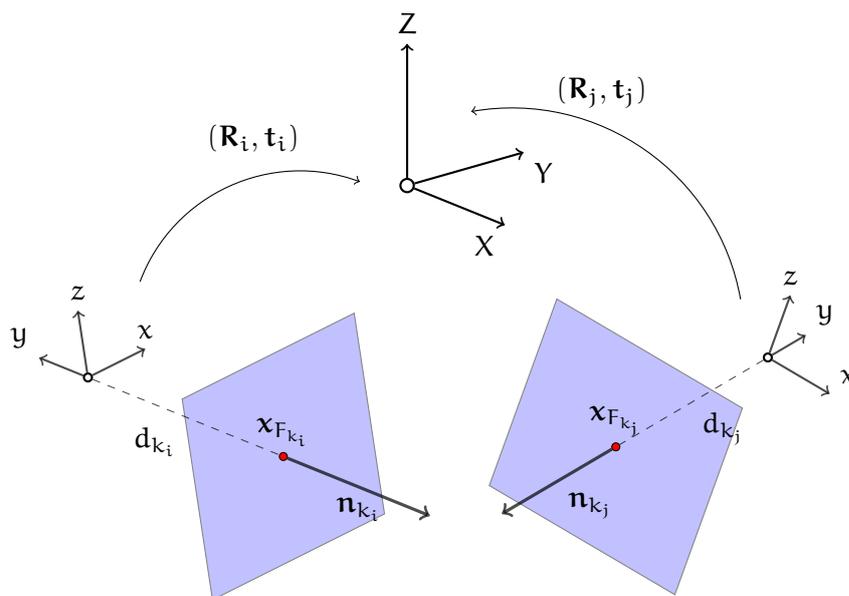


Figure 4.9: Corresponding plane pair  $k$  in *point clouds*  $i$  and  $j$  in relation to a superordinate coordinate system.

For an identical pair of planes  $k$  occurring in the *point clouds*  $i$  and  $j$ , see [Figure 4.9](#), the equations

$$\begin{aligned}
 \mathbf{n}_{k_i} \mathbf{x} - d_{k_i} &= 0 \\
 \mathbf{n}_{k_j} \mathbf{x} - d_{k_j} &= 0
 \end{aligned} \tag{4.35}$$

can be given analogously to Equation (4.11). With the transformation of the normal vectors  $\mathbf{n}_{k_i}$  and  $\mathbf{n}_{k_j}$  into the superordinate system  $\mathbf{n}_{k_{glob}}$  one obtains directly

$$\begin{aligned}\mathbf{n}_{k_{glob}} &= \mathbf{R}_i \mathbf{n}_{k_i} = \mathbf{R}_j \mathbf{n}_{k_j} \\ &\Rightarrow \mathbf{R}_i \mathbf{n}_{k_i} - \mathbf{R}_j \mathbf{n}_{k_j} = \mathbf{0},\end{aligned}\tag{4.36}$$

resulting in three *conditional equations*

$$\begin{aligned}\hat{r}_{i11} \hat{n}_{i_x} + \hat{r}_{i12} \hat{n}_{i_y} + \hat{r}_{i13} \hat{n}_{i_z} - \hat{r}_{j11} \hat{n}_{j_x} - \hat{r}_{j12} \hat{n}_{j_y} - \hat{r}_{j13} \hat{n}_{j_z} &= 0 \\ \hat{r}_{i21} \hat{n}_{i_x} + \hat{r}_{i22} \hat{n}_{i_y} + \hat{r}_{i23} \hat{n}_{i_z} - \hat{r}_{j21} \hat{n}_{j_x} - \hat{r}_{j22} \hat{n}_{j_y} - \hat{r}_{j23} \hat{n}_{j_z} &= 0 \\ \hat{r}_{i31} \hat{n}_{i_x} + \hat{r}_{i32} \hat{n}_{i_y} + \hat{r}_{i33} \hat{n}_{i_z} - \hat{r}_{j31} \hat{n}_{j_x} - \hat{r}_{j32} \hat{n}_{j_y} - \hat{r}_{j33} \hat{n}_{j_z} &= 0\end{aligned}\tag{4.37}$$

for a plane match  $k$  concerning the *point clouds*  $i$  and  $j$ , where

$\hat{r}_{i11}, \hat{r}_{i12}, \dots, \hat{r}_{i33}$  are the unknown adjusted elements of the *rotation matrix*  $\hat{\mathbf{R}}_i$  for *point cloud*  $i$ , expressed by the *quaternions*  $\hat{\mathbf{q}}_i = [\hat{q}_{i_0} \hat{q}_{i_x} \hat{q}_{i_y} \hat{q}_{i_z}]^T$  to be determined, see also Equation (1.24) as well as

$\hat{r}_{j11}, \hat{r}_{j12}, \dots, \hat{r}_{j33}$  the unknown adjusted elements of the *rotation matrix*  $\hat{\mathbf{R}}_j$  for *point cloud*  $j$  and

$$\hat{n}_{i_x} = n_{i_x} + v_{n_{i_x}}$$

$\hat{n}_{i_y} = n_{i_y} + v_{n_{i_y}}$  are the corrected normal vector component observations of *point cloud*  $i$  after introducing the residuals as well as

$$\hat{n}_{i_z} = n_{i_z} + v_{n_{i_z}}$$

$$\hat{n}_{j_x} = n_{j_x} + v_{n_{j_x}}$$

$\hat{n}_{j_y} = n_{j_y} + v_{n_{j_y}}$  the corrected normal vector component observations of *point cloud*  $j$

$$\hat{n}_{j_z} = n_{j_z} + v_{n_{j_z}}$$

Considering the same assumption as in Section 4.4, that the plane footpoints  $\mathbf{x}_{F_{k_i}} = \mathbf{n}_{k_i} d_{k_i}$  and  $\mathbf{x}_{F_{k_j}} = \mathbf{n}_{k_j} d_{k_j}$  of corresponding planes  $k$  in the *point clouds*  $i$  and  $j$ , cf. Figure 4.9, transformed into the superordinate system via  $\mathbf{x}_{F_{k_{glob}}} = \mathbf{R}_i \mathbf{x}_{F_{k_i}} + \mathbf{t}_i = \mathbf{R}_j \mathbf{x}_{F_{k_j}} + \mathbf{t}_j$ <sup>16</sup>, see also Equation (4.16),

<sup>15</sup> Indicates that the local normal vectors in the superordinate coordinate system must be identical.

<sup>16</sup> A similar procedure, which uses the centers of gravity of the planes for the translation determination, is described in GIELSDORF (2009).

should correspond to each other, with the dot product

$$\begin{aligned}
 \mathbf{d}_{k_{\text{glob}}} &= \langle \mathbf{n}_{k_{\text{glob}}}, \mathbf{x}_{F_{k_{\text{glob}}}} \rangle \\
 &= \langle (\mathbf{R}_i \mathbf{n}_{k_i}), (\mathbf{R}_i \mathbf{n}_{k_i} \mathbf{d}_{k_i} + \mathbf{t}_i) \rangle = \langle (\mathbf{R}_j \mathbf{n}_{k_j}), (\mathbf{R}_j \mathbf{n}_{k_j} \mathbf{d}_{k_j} + \mathbf{t}_j) \rangle^{17} \\
 &\Rightarrow \mathbf{d}_{k_i} + \langle \mathbf{R}_i \mathbf{n}_{k_i}, \mathbf{t}_i \rangle - \mathbf{d}_{k_j} - \langle \mathbf{R}_j \mathbf{n}_{k_j}, \mathbf{t}_j \rangle
 \end{aligned} \quad (4.38)$$

another *conditional equation* emerges

$$\begin{aligned}
 &\hat{\mathbf{d}}_i + (\hat{r}_{i11} \hat{\mathbf{n}}_{i_x} + \hat{r}_{i12} \hat{\mathbf{n}}_{i_y} + \hat{r}_{i13} \hat{\mathbf{n}}_{i_z}) \hat{\mathbf{t}}_{i_x} + (\hat{r}_{i21} \hat{\mathbf{n}}_{i_x} + \hat{r}_{i22} \hat{\mathbf{n}}_{i_y} + \hat{r}_{i23} \hat{\mathbf{n}}_{i_z}) \hat{\mathbf{t}}_{i_y} \\
 &+ (\hat{r}_{i31} \hat{\mathbf{n}}_{i_x} + \hat{r}_{i32} \hat{\mathbf{n}}_{i_y} + \hat{r}_{i33} \hat{\mathbf{n}}_{i_z}) \hat{\mathbf{t}}_{i_z} - \hat{\mathbf{d}}_j - (\hat{r}_{j11} \hat{\mathbf{n}}_{j_x} + \hat{r}_{j12} \hat{\mathbf{n}}_{j_y} + \hat{r}_{j13} \hat{\mathbf{n}}_{j_z}) \hat{\mathbf{t}}_{j_x} \\
 &- (\hat{r}_{j21} \hat{\mathbf{n}}_{j_x} + \hat{r}_{j22} \hat{\mathbf{n}}_{j_y} + \hat{r}_{j23} \hat{\mathbf{n}}_{j_z}) \hat{\mathbf{t}}_{j_y} - (\hat{r}_{j31} \hat{\mathbf{n}}_{j_x} + \hat{r}_{j32} \hat{\mathbf{n}}_{j_y} + \hat{r}_{j33} \hat{\mathbf{n}}_{j_z}) \hat{\mathbf{t}}_{j_z} = 0
 \end{aligned} \quad (4.39)$$

for a plane match  $k$  concerning the *point clouds*  $i$  and  $j$ , where

$$\begin{aligned}
 \hat{\mathbf{t}}_{i_x}, \hat{\mathbf{t}}_{i_y}, \hat{\mathbf{t}}_{i_z} &\quad \text{are the unknown adjusted values of the components of the position vector } \hat{\mathbf{t}}_i \text{ of} \\
 &\quad \text{the } \textit{exterior orientation} \text{ for } \textit{point cloud } i \text{ as well as} \\
 \hat{\mathbf{t}}_{j_x}, \hat{\mathbf{t}}_{j_y}, \hat{\mathbf{t}}_{j_z} &\quad \text{the unknown adjusted values of the components of the position vector } \hat{\mathbf{t}}_j \text{ of the} \\
 &\quad \textit{exterior orientation} \text{ for } \textit{point cloud } j \text{ and} \\
 \hat{\mathbf{d}}_i = \mathbf{d}_i + \nu_{d_i} &\quad \text{are the corrected observations for the orthogonal distances to the plane origins} \\
 \hat{\mathbf{d}}_j = \mathbf{d}_j + \nu_{d_j} &\quad \text{of } \textit{point clouds } i \text{ and } j \text{ after introducing the residuals}
 \end{aligned}$$

From [Section 4.4](#) the transformation information between the individual *point clouds* can be used as approximate information for the individual *exterior orientations*  $i$  with  $\mathbf{q}_i^0 = [q_{i0}^0 \ q_{i_x}^0 \ q_{i_y}^0 \ q_{i_z}^0]^T$  and  $\mathbf{t}_i^0 = [t_{i_x}^0 \ t_{i_y}^0 \ t_{i_z}^0]^T$ , to be used together with the approximate residuals  $\nu_{n_{i_x}}^0, \nu_{n_{i_y}}^0, \nu_{n_{i_z}}^0, \nu_{d_i}^0$  for a plane  $k$ , in order to obtain the approximate *condition equations* as vector

$$\begin{aligned}
 \Psi_{k_{ij}}^0 &= \\
 &\left[ \begin{array}{l}
 \mathbf{d}_i^0 + \mathbf{r}_{i1}^0 \mathbf{n}_i^0 t_{i_x}^0 + \mathbf{r}_{i2}^0 \mathbf{n}_i^0 t_{i_y}^0 + \mathbf{r}_{i3}^0 \mathbf{n}_i^0 t_{i_z}^0 - \mathbf{d}_j^0 - \mathbf{r}_{j1}^0 \mathbf{n}_j^0 t_{j_x}^0 - \mathbf{r}_{j2}^0 \mathbf{n}_j^0 t_{j_y}^0 - \mathbf{r}_{j3}^0 \mathbf{n}_j^0 t_{j_z}^0 \\
 \mathbf{r}_{i11}^0 \mathbf{n}_{i_x}^0 + \mathbf{r}_{i12}^0 \mathbf{n}_{i_y}^0 + \mathbf{r}_{i13}^0 \mathbf{n}_{i_z}^0 - \mathbf{r}_{j11}^0 \mathbf{n}_{j_x}^0 - \mathbf{r}_{j12}^0 \mathbf{n}_{j_y}^0 - \mathbf{r}_{j13}^0 \mathbf{n}_{j_z}^0 \\
 \mathbf{r}_{i21}^0 \mathbf{n}_{i_x}^0 + \mathbf{r}_{i22}^0 \mathbf{n}_{i_y}^0 + \mathbf{r}_{i23}^0 \mathbf{n}_{i_z}^0 - \mathbf{r}_{j21}^0 \mathbf{n}_{j_x}^0 - \mathbf{r}_{j22}^0 \mathbf{n}_{j_y}^0 - \mathbf{r}_{j23}^0 \mathbf{n}_{j_z}^0 \\
 \mathbf{r}_{i31}^0 \mathbf{n}_{i_x}^0 + \mathbf{r}_{i32}^0 \mathbf{n}_{i_y}^0 + \mathbf{r}_{i33}^0 \mathbf{n}_{i_z}^0 - \mathbf{r}_{j31}^0 \mathbf{n}_{j_x}^0 - \mathbf{r}_{j32}^0 \mathbf{n}_{j_y}^0 - \mathbf{r}_{j33}^0 \mathbf{n}_{j_z}^0
 \end{array} \right] \quad (4.40)
 \end{aligned}$$

<sup>17</sup>Indicates that the local orthogonal distances to a plane's origin in the superordinate coordinate system must be identical.

assuming  $n$  plane matches, with  $\{k = 1 : n\}$  for *point cloud* pairs  $i$  and  $j$  based on [Equations \(4.37\)](#) and [\(4.39\)](#), where

$$\begin{aligned}
 \mathbf{r}_{i_1}^0 \cdots \mathbf{r}_{i_3}^0 & \text{ are the substitutions for rowwise matrix elements, such as for e. g. } \mathbf{r}_{i_1}^0 = \\
 & [r_{i_{11}}^0 \ r_{i_{12}}^0 \ r_{i_{13}}^0] \text{ concerning the } \textit{exterior orientation } i \text{ as well as} \\
 \mathbf{r}_{j_1}^0 \cdots \mathbf{r}_{j_3}^0 & \text{ the equivalent substitutions concerning } \textit{exterior orientation } j \text{ and} \\
 \mathbf{n}_i^0 = [n_{i_x}^0 \ n_{i_y}^0 \ n_{i_z}^0]^T & \text{ is the initial corrected normal vector with its components e. g. } n_{i_x}^0 = n_{i_x} + \\
 & v_{n_{i_x}}^0 \text{ of the plane match in } \textit{point cloud } i \text{ as well as} \\
 \mathbf{n}_j^0 = [n_{j_x}^0 \ n_{j_y}^0 \ n_{j_z}^0]^T & \text{ the equivalent initial corrected normal vector with its components e. g. } n_{j_x}^0 = \\
 & n_{j_x} + v_{n_{j_x}}^0 \text{ in } \textit{point cloud } j \text{ and} \\
 d_i^0 = d_i + v_{d_i}^0 & \text{ the initial corrected orthogonal distances to the plane origins of } \textit{point clouds} \\
 d_j^0 = d_j + v_{d_j}^0 & i \text{ and } j \text{ after introducing the residuals.}
 \end{aligned}$$

The entire *condition vector* for all *point cloud* pairs  $i$  and  $j$

$$\Psi^0 = [\Psi_{12}^0 \ \Psi_{13}^0 \ \cdots \ \Psi_{23}^0 \ \Psi_{24}^0 \ \cdots \ \Psi_{ij}^0]^T \quad (4.41)$$

is therefore composed of the *point cloud* pair vectors

$$\Psi_{ij}^0 = [\Psi_{1ij}^0 \ \cdots \ \Psi_{2ij}^0 \ \cdots \ \Psi_{kij}^0 \ \cdots \ \Psi_{nij}^0]^T \quad (4.42)$$

containing the individual plane matches  $k$  from [Equation \(4.40\)](#). The initial residuals for a pair of *point clouds* are summarized in a vector

$$\mathbf{v}_{ij}^0 = [v_{1ij}^0 \ v_{2ij}^0 \ v_{3ij}^0 \ \cdots \ v_{kij}^0 \ \cdots \ v_{nij}^0]^T \quad (4.43)$$

consisting of the individual plain matches

$$\mathbf{v}_{kij}^0 = [v_{n_{i_x}}^0 \ v_{n_{i_y}}^0 \ v_{n_{i_z}}^0 \ v_{d_i}^0 \ | \ v_{n_{j_x}}^0 \ v_{n_{j_y}}^0 \ v_{n_{j_z}}^0 \ v_{d_j}^0]^T. \quad (4.44)$$

Thus the entire vector of residuals for all *point cloud* pairs reads

$$\mathbf{v}^0 = [v_{12}^0 \ v_{13}^0 \ \cdots \ v_{23}^0 \ v_{24}^0 \ \cdots \ v_{ij}^0]^T. \quad (4.45)$$

According to the order of the residual vector  $\mathbf{v}^0$  in [\(4.45\)](#), the precision relations of the observations are given in equivalence to [Section 4.4](#), cf. [Equation \(4.24\)](#), by the choice of the *SD of the unit weight* of

<sup>18</sup> For a simplified representation, the same number of  $\{k = 1 : n\}$  plane matches per scan pair  $ij$  is assumed.

$\sigma_0 = 0.01$  in the **CM** of observations

$$\mathbf{Q}_{ll} = \text{diag} \left[ \mathbf{Q}_{ll_{12}} \quad \mathbf{Q}_{ll_{13}} \quad \cdots \quad \mathbf{Q}_{ll_{23}} \quad \mathbf{Q}_{ll_{24}} \quad \cdots \quad \mathbf{Q}_{ll_{ij}} \right] \quad (4.46)$$

with the individual **CMs** of observations

$$\mathbf{Q}_{ll_{ij}} = \text{diag} \left[ \mathbf{Q}_{ll_{1,ij}} \quad \mathbf{Q}_{ll_{2,ij}} \quad \mathbf{Q}_{ll_{3,ij}} \quad \cdots \quad \mathbf{Q}_{ll_{k,ij}} \quad \cdots \quad \mathbf{Q}_{ll_{n,ij}} \right] \quad (4.47)$$

for a pair of *point clouds*  $ij$  and its submatrices

$$\mathbf{Q}_{ll_{k,ij}} = \text{diag} \left[ \mathbf{Q}_{ll_i} \quad | \quad \mathbf{Q}_{ll_j} \right] \quad (4.48)$$

of the individual plane match  $k$ , composed of the respective **CMs** of observations

$$\mathbf{Q}_{ll_i} = \frac{s_{0i}^2}{\sigma_0^2} \mathbf{Q}_{xx_i} \quad \text{and} \quad \mathbf{Q}_{ll_j} = \frac{s_{0j}^2}{\sigma_0^2} \mathbf{Q}_{xx_j} \quad ^{19}, \quad (4.49)$$

for the plane matches  $k$  in the system  $i$  and  $j$ , using the corresponding **CMs** of adjusted unknowns  $\mathbf{Q}_{xx_i}$ ,  $\mathbf{Q}_{xx_j}$  and the *empirical reference variances*  $s_{0i}^2$ ,  $s_{0j}^2$ , computed after each rigorous plane adjustment following Section 2.4.2 via Equation (1.82).

As in Section 4.4 an iterative solution to the problem case of Equation (4.41) in a **GHM** with *constraints*, see also Section 1.8.7, between the rotation parameters for each *exterior orientation* of the individual *point clouds* takes place.

The rotational part assuming  $i = 1, \dots, m$  *exterior orientations*, represented by *quaternions*, is thus subject to the *secondary condition*

$$\boldsymbol{\gamma}_n^0 = \begin{bmatrix} q_{1_0}^0{}^2 + q_{1_x}^0{}^2 + q_{1_y}^0{}^2 + q_{1_z}^0{}^2 - 1 \\ \vdots \\ q_{i_0}^0{}^2 + q_{i_x}^0{}^2 + q_{i_y}^0{}^2 + q_{i_z}^0{}^2 - 1 \\ \vdots \\ q_{m_0}^0{}^2 + q_{m_x}^0{}^2 + q_{m_y}^0{}^2 + q_{m_z}^0{}^2 - 1 \end{bmatrix}. \quad (4.50)$$

However, for the matching procedure in Section 4.4, the relationship between the start and target systems predetermines the *geodetic datum*, unlike the registration procedure, the datum is not directly

<sup>19</sup>The precision of the plane components stems from the plane adjustment of the plane segmentation procedure in Section 4.3.2, i. e. the adjustment after Section 2.4.2, and is based on the individual point precisions derived from the proposed stochastic model of the **TLS** as already used previously in Section 2.3.1.

defined by the mathematical relationship. For the present case, two common possibilities for defining the *geodetic datum* are reasonable:

- A. Specifying the datum in one of the local scanner coordinate system of a TLS's station
- B. Definition of the datum via the approximate values of the *exterior orientations* to be estimated, i. e. transformation parameters into the superordinate coordinate system, in the sense of a *free adjustment*.

For case A, each parameter of the relevant *exterior orientation* yields one *constraint equation* for the selected fix station  $i$ , four

$$q_{i_0} = \text{const.}, \quad q_{i_x} = \text{const.}, \quad q_{i_y} = \text{const.} \quad \text{and} \quad q_{i_z} = \text{const.} \quad (4.51)$$

to define the rotation and another three

$$t_{i_x} = \text{const.}, \quad t_{i_y} = \text{const.}, \quad \text{and} \quad t_{i_z} = \text{const.} \quad (4.52)$$

to define the translation, setting each of the components to a constant value. Usually, the translation components are set to  $t_{i_x} = t_{i_y} = t_{i_z} = 0$  and the rotation components are set as unit quaternion  $\mathbf{q}_i = [q_{i_0} \ q_{i_x} \ q_{i_y} \ q_{i_z}]^T = [1 \ 0 \ 0 \ 0]^T$  in order to choose a reference coordinate system that is neither rotated nor translated.

THE CONSIDERATION OF a fixed *exterior orientation* during the adjustment is straightforward. The corresponding parameters are introduced as constant quantities, consequently also no partial derivatives are formed and do not cause any correction. From the technical side of programming, the definition of the datum can easily be assigned to another local coordinate system reducing the *Jacobians* by the corresponding columns.

In case B, applied in this section, the datum is defined distributed over all  $i = 1, \dots, m$  *exterior orientations* for the local coordinate systems according to the idea of a *free adjustment*. In doing so, after GIELSDORF (2009) three *constraint equations*

$$\gamma_t^0 = \begin{bmatrix} \sum \hat{t}_{i_x} - \sum t_{i_x}^0 \\ \sum \hat{t}_{i_y} - \sum t_{i_y}^0 \\ \sum \hat{t}_{i_z} - \sum t_{i_z}^0 \end{bmatrix} = \begin{bmatrix} \sum \Delta \hat{t}_{i_x} \\ \sum \Delta \hat{t}_{i_y} \\ \sum \Delta \hat{t}_{i_z} \end{bmatrix} \quad (4.53)$$

are considered to determine the translation and another three

$$\begin{aligned}
 \gamma_r^0 &= \begin{bmatrix} \sum 2(\hat{q}_{i_0} \hat{q}_{i_x} + \hat{q}_{i_y} \hat{q}_{i_z}) - \sum 2(q_{i_0}^0 q_{i_x}^0 + q_{i_y}^0 q_{i_z}^0) \\ \sum 2(\hat{q}_{i_x} \hat{q}_{i_z} - \hat{q}_{i_0} \hat{q}_{i_y}) - \sum 2(q_{i_x}^0 q_{i_z}^0 - q_{i_0}^0 q_{i_y}^0) \\ \sum 2(\hat{q}_{i_x} \hat{q}_{i_y} + \hat{q}_{i_0} \hat{q}_{i_z}) - \sum 2(q_{i_x}^0 q_{i_y}^0 + q_{i_0}^0 q_{i_z}^0) \end{bmatrix} \\
 &= \begin{bmatrix} \sum 2(\Delta \hat{q}_{i_0} \Delta \hat{q}_{i_x} + \Delta \hat{q}_{i_y} \Delta \hat{q}_{i_z}) \\ \sum 2(\Delta \hat{q}_{i_x} \Delta \hat{q}_{i_z} - \Delta \hat{q}_{i_0} \Delta \hat{q}_{i_y}) \\ \sum 2(\Delta \hat{q}_{i_x} \Delta \hat{q}_{i_y} + \Delta \hat{q}_{i_0} \Delta \hat{q}_{i_z}) \end{bmatrix}
 \end{aligned} \tag{4.54}$$

to determine the rotation.

The *Jacobian matrix* with the partial derivatives of the *condition equations* according to the unknowns

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{J}_{a_{12_1}} & \mathbf{J}_{a_{12_2}} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \mathbf{J}_{a_{13_1}} & 0 & \mathbf{J}_{a_{13_3}} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \mathbf{J}_{a_{23_2}} & \mathbf{J}_{a_{23_3}} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \mathbf{J}_{a_{24_2}} & 0 & \mathbf{J}_{a_{24_4}} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \mathbf{J}_{a_{ij_i}} & 0 & \mathbf{J}_{a_{ij_j}} \end{bmatrix} \tag{4.55}$$

is composed of the matrices

$$\begin{aligned}
 \mathbf{J}_{a_{ij_i}} &= \begin{bmatrix} \mathbf{J}_{1_{ij_i}} & \mathbf{J}_{2_{ij_i}} & \mathbf{J}_{3_{ij_i}} & \mathbf{J}_{k_{ij_i}} & \cdots & \mathbf{J}_{n_{ij_i}} \end{bmatrix}^T \\
 \mathbf{J}_{a_{ij_j}} &= \begin{bmatrix} \mathbf{J}_{1_{ij_j}} & \mathbf{J}_{2_{ij_j}} & \mathbf{J}_{3_{ij_j}} & \mathbf{J}_{k_{ij_j}} & \cdots & \mathbf{J}_{n_{ij_j}} \end{bmatrix}^T
 \end{aligned} \tag{4.56}$$

for each neighboured *point clouds* i and j with

$$\begin{aligned}
 \mathbf{J}_{k_{ij_i}} &= \begin{bmatrix} \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{i_0}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{i_x}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{i_y}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{i_z}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{i_x}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{i_y}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{i_z}^0} \end{bmatrix} \\
 \mathbf{J}_{k_{ij_j}} &= \begin{bmatrix} \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{j_0}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{j_x}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{j_y}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{j_z}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{j_x}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{j_y}^0} & \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{j_z}^0} \end{bmatrix}
 \end{aligned} \tag{4.57}$$

for each plane match k based on [Equation \(4.40\)](#), where

$$\frac{\partial \Psi_{k_{ij}}^0}{\partial t_{i_x}^0} = \begin{bmatrix} \mathbf{r}_{i_1}^0 & \mathbf{n}_i^0 & 0 & 0 & 0 \end{bmatrix}^T \quad \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{i_y}^0} = \begin{bmatrix} \mathbf{r}_{i_2}^0 & \mathbf{n}_i^0 & 0 & 0 & 0 \end{bmatrix}^T \quad \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{i_z}^0} = \begin{bmatrix} \mathbf{r}_{i_3}^0 & \mathbf{n}_i^0 & 0 & 0 & 0 \end{bmatrix}^T$$

and

$$\begin{aligned} \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{i_0}^0} &= \begin{bmatrix} 2(\mathbf{t}_{i_x}^0 [q_{i_0}^0 - q_{i_z}^0 q_{i_y}^0]^T + \mathbf{t}_{i_y}^0 [q_{i_z}^0 q_{i_0}^0 - q_{i_x}^0]^T + \mathbf{t}_{i_z}^0 [-q_{i_y}^0 q_{i_x}^0 q_{i_0}^0]^T) \mathbf{n}_i^0 \\ 2(q_{i_0}^0 n_{i_x}^0 - q_{i_z}^0 n_{i_y}^0 + q_{i_y}^0 n_{i_z}^0) \\ 2(q_{i_z}^0 n_{i_x}^0 + q_{i_0}^0 n_{i_y}^0 - q_{i_x}^0 n_{i_z}^0) \\ 2(-q_{i_y}^0 n_{i_x}^0 + q_{i_x}^0 n_{i_y}^0 + q_{i_0}^0 n_{i_z}^0) \end{bmatrix} \\ \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{i_x}^0} &= \begin{bmatrix} 2(\mathbf{t}_{i_x}^0 [q_{i_x}^0 q_{i_y}^0 q_{i_z}^0]^T + \mathbf{t}_{i_y}^0 [q_{i_y}^0 - q_{i_x}^0 - q_{i_0}^0]^T + \mathbf{t}_{i_z}^0 [q_{i_z}^0 q_{i_0}^0 - q_{i_x}^0]^T) \mathbf{n}_i^0 \\ 2(q_{i_x}^0 n_{i_x}^0 + q_{i_y}^0 n_{i_y}^0 + q_{i_z}^0 n_{i_z}^0) \\ 2(q_{i_y}^0 n_{i_x}^0 - q_{i_x}^0 n_{i_y}^0 - q_{i_0}^0 n_{i_z}^0) \\ 2(q_{i_z}^0 n_{i_x}^0 + q_{i_0}^0 n_{i_y}^0 - q_{i_x}^0 n_{i_z}^0) \end{bmatrix} \\ \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{i_y}^0} &= \begin{bmatrix} 2(\mathbf{t}_{i_x}^0 [-q_{i_y}^0 q_{i_x}^0 q_{i_0}^0]^T + \mathbf{t}_{i_y}^0 [q_{i_x}^0 q_{i_y}^0 q_{i_z}^0]^T + \mathbf{t}_{i_z}^0 [-q_{i_0}^0 q_{i_z}^0 - q_{i_y}^0]^T) \mathbf{n}_i^0 \\ 2(-q_{i_y}^0 n_{i_x}^0 + q_{i_x}^0 n_{i_y}^0 + q_{i_0}^0 n_{i_z}^0) \\ 2(q_{i_x}^0 n_{i_x}^0 + q_{i_y}^0 n_{i_y}^0 + q_{i_z}^0 n_{i_z}^0) \\ 2(-q_{i_0}^0 n_{i_x}^0 + q_{i_z}^0 n_{i_y}^0 - q_{i_y}^0 n_{i_z}^0) \end{bmatrix} \\ \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{i_z}^0} &= \begin{bmatrix} 2(\mathbf{t}_{i_x}^0 [-q_{i_z}^0 - q_{i_0}^0 q_{i_x}^0]^T + \mathbf{t}_{i_y}^0 [q_{i_0}^0 - q_{i_z}^0 q_{i_y}^0]^T + \mathbf{t}_{i_z}^0 [q_{i_x}^0 q_{i_y}^0 q_{i_z}^0]^T) \mathbf{n}_i^0 \\ 2(-q_{i_z}^0 n_{i_x}^0 - q_{i_0}^0 n_{i_y}^0 + q_{i_x}^0 n_{i_z}^0) \\ 2(q_{i_0}^0 n_{i_x}^0 - q_{i_z}^0 n_{i_y}^0 + q_{i_y}^0 n_{i_z}^0) \\ 2(q_{i_x}^0 n_{i_x}^0 + q_{i_y}^0 n_{i_y}^0 + q_{i_z}^0 n_{i_z}^0) \end{bmatrix} \end{aligned}$$

are the partial derivatives regarding *exterior orientation*  $\mathbf{i}$  and moreover

$$\frac{\partial \Psi_{k_{ij}}^0}{\partial t_{j_x}^0} = [-\mathbf{r}_{j_1}^0 \mathbf{n}_j^0 \quad 0 \quad 0 \quad 0]^T \quad \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{j_y}^0} = [-\mathbf{r}_{j_2}^0 \mathbf{n}_j^0 \quad 0 \quad 0 \quad 0]^T \quad \frac{\partial \Psi_{k_{ij}}^0}{\partial t_{j_z}^0} = [-\mathbf{r}_{j_3}^0 \mathbf{n}_j^0 \quad 0 \quad 0 \quad 0]^T$$

and

$$\frac{\partial \Psi_{k_{ij}}^0}{\partial q_{j_0}^0} = \begin{bmatrix} -2(\mathbf{t}_{j_x}^0 [q_{j_0}^0 - q_{j_z}^0 q_{j_y}^0]^T + \mathbf{t}_{j_y}^0 [q_{j_z}^0 q_{j_0}^0 - q_{j_x}^0]^T + \mathbf{t}_{j_z}^0 [-q_{j_y}^0 q_{j_x}^0 q_{j_0}^0]^T) \mathbf{n}_j^0 \\ 2(-q_{j_0}^0 n_{j_x}^0 + q_{j_z}^0 n_{j_y}^0 - q_{j_y}^0 n_{j_z}^0) \\ 2(-q_{j_z}^0 n_{j_x}^0 - q_{j_0}^0 n_{j_y}^0 + q_{j_x}^0 n_{j_z}^0) \\ 2(q_{j_y}^0 n_{j_x}^0 - q_{j_x}^0 n_{j_y}^0 - q_{j_0}^0 n_{j_z}^0) \end{bmatrix}$$

$$\begin{aligned}
 \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{j_x}^0} &= \begin{bmatrix} -2(\mathbf{t}_{j_x}^0 [q_{j_x}^0 \ q_{j_y}^0 \ q_{j_z}^0]^T + \mathbf{t}_{j_y}^0 [q_{j_y}^0 \ -q_{j_x}^0 \ -q_{j_0}^0]^T + \mathbf{t}_{j_z}^0 [q_{j_z}^0 \ q_{j_0}^0 \ -q_{j_x}^0]^T) \mathbf{n}_j^0 \\ 2(-q_{j_x}^0 n_{j_x}^0 - q_{j_y}^0 n_{j_y}^0 - q_{j_z}^0 n_{j_z}^0) \\ 2(-q_{j_y}^0 n_{j_x}^0 + q_{j_x}^0 n_{j_y}^0 + q_{j_0}^0 n_{j_z}^0) \\ 2(-q_{j_z}^0 n_{j_x}^0 - q_{j_0}^0 n_{j_y}^0 + q_{j_x}^0 n_{j_z}^0) \end{bmatrix} \\
 \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{j_y}^0} &= \begin{bmatrix} -2(\mathbf{t}_{j_x}^0 [-q_{j_y}^0 \ q_{j_x}^0 \ q_{j_0}^0]^T + \mathbf{t}_{j_y}^0 [q_{j_x}^0 \ q_{j_y}^0 \ q_{j_z}^0]^T + \mathbf{t}_{j_z}^0 [-q_{j_0}^0 \ q_{j_z}^0 \ -q_{j_y}^0]^T) \mathbf{n}_j^0 \\ 2(q_{j_y}^0 n_{j_x}^0 - q_{j_x}^0 n_{j_y}^0 - q_{j_0}^0 n_{j_z}^0) \\ 2(-q_{j_x}^0 n_{j_x}^0 - q_{j_y}^0 n_{j_y}^0 - q_{j_z}^0 n_{j_z}^0) \\ 2(q_{j_0}^0 n_{j_x}^0 - q_{j_z}^0 n_{j_y}^0 + q_{j_y}^0 n_{j_z}^0) \end{bmatrix} \\
 \frac{\partial \Psi_{k_{ij}}^0}{\partial q_{j_z}^0} &= \begin{bmatrix} -2(\mathbf{t}_{j_x}^0 [-q_{j_z}^0 \ -q_{j_0}^0 \ q_{j_x}^0]^T + \mathbf{t}_{j_y}^0 [q_{j_0}^0 \ -q_{j_z}^0 \ q_{j_y}^0]^T + \mathbf{t}_{j_z}^0 [q_{j_x}^0 \ q_{j_y}^0 \ q_{j_z}^0]^T) \mathbf{n}_j^0 \\ 2(q_{j_z}^0 n_{j_x}^0 + q_{j_0}^0 n_{j_y}^0 - q_{j_x}^0 n_{j_z}^0) \\ 2(-q_{j_0}^0 n_{j_x}^0 + q_{j_z}^0 n_{j_y}^0 - q_{j_y}^0 n_{j_z}^0) \\ 2(-q_{j_x}^0 n_{j_x}^0 - q_{j_y}^0 n_{j_y}^0 - q_{j_z}^0 n_{j_z}^0) \end{bmatrix}
 \end{aligned}$$

are the equivalent partial derivatives regarding *exterior orientation*  $j$ . With the partial derivatives of the *conditional equations* concerning a plane match  $k$ , see [Equation \(4.40\)](#), with respect to the residuals in [Equation \(4.45\)](#)

$$\mathbf{J}_{b_{k_{ij}}} = \begin{bmatrix} \mathbf{r}_{i_1}^0 \mathbf{t}_i^0 & \mathbf{r}_{i_2}^0 \mathbf{t}_i^0 & \mathbf{r}_{i_3}^0 \mathbf{t}_i^0 & 1 & -\mathbf{r}_{j_1}^0 \mathbf{t}_j^0 & -\mathbf{r}_{j_2}^0 \mathbf{t}_j^0 & -\mathbf{r}_{j_3}^0 \mathbf{t}_j^0 & 1 \\ \mathbf{r}_{i_{11}}^0 & \mathbf{r}_{i_{12}}^0 & \mathbf{r}_{i_{13}}^0 & 0 & -\mathbf{r}_{j_{11}}^0 & -\mathbf{r}_{j_{12}}^0 & -\mathbf{r}_{j_{13}}^0 & 0 \\ \mathbf{r}_{i_{21}}^0 & \mathbf{r}_{i_{22}}^0 & \mathbf{r}_{i_{23}}^0 & 0 & -\mathbf{r}_{j_{21}}^0 & -\mathbf{r}_{j_{22}}^0 & -\mathbf{r}_{j_{23}}^0 & 0 \\ \mathbf{r}_{i_{31}}^0 & \mathbf{r}_{i_{32}}^0 & \mathbf{r}_{i_{33}}^0 & 0 & -\mathbf{r}_{j_{31}}^0 & -\mathbf{r}_{j_{32}}^0 & -\mathbf{r}_{j_{33}}^0 & 0 \end{bmatrix}, \quad (4.58)$$

where

$\mathbf{r}_{i_1}^0 \cdots \mathbf{r}_{i_3}^0$  are the substitutions for columnwise matrix elements, such as for e. g.  $\mathbf{r}_{i_1}^0 = [r_{i_{11}}^0 \ r_{i_{21}}^0 \ r_{i_{31}}^0]^T$  concerning the *exterior orientation*  $i$  as well as

$\mathbf{r}_{j_1}^0 \cdots \mathbf{r}_{j_3}^0$  the equivalent substitutions concerning *exterior orientation*  $j$ ,

and the resulting matrices

$$\mathbf{J}_{b_{ij}} = \text{diag} \left[ \mathbf{J}_{b_{1ij}} \quad \mathbf{J}_{b_{2ij}} \quad \mathbf{J}_{b_{3ij}} \quad \cdots \quad \mathbf{J}_{b_{kij}} \quad \cdots \quad \mathbf{J}_{b_{nij}} \right] \quad (4.59)$$

containing all matches  $k$  for a point cloud combination  $i, j$  a further *Jacobian matrix*

$$\mathbf{J}_b = \text{diag} \left[ \mathbf{J}_{b_{12}} \quad \mathbf{J}_{b_{13}} \quad \cdots \quad \mathbf{J}_{b_{23}} \quad \mathbf{J}_{b_{24}} \quad \cdots \quad \mathbf{J}_{b_{ij}} \right] \quad (4.60)$$

results. Finally, the *Jacobian matrices* with the partial derivatives of the *constraint equations* according to the unknowns, respectively  $i = 1, \dots, m$  *exterior orientations*

$$\mathbf{J}_n = \text{diag} \begin{bmatrix} 2q_{1_0}^0 & 2q_{1_x}^0 & 2q_{1_y}^0 & 2q_{1_z}^0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2q_{i_0}^0 & 2q_{i_x}^0 & 2q_{i_y}^0 & 2q_{i_z}^0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2q_{m_0}^0 & 2q_{m_x}^0 & 2q_{m_y}^0 & 2q_{m_z}^0 & 0 & 0 & 0 \end{bmatrix} \quad (4.61)$$

to take the normalization of the rotation *quaternions* in Equation (4.50) into account as well

$$\mathbf{J}_t = \left[ \frac{\partial \gamma_t^0}{\partial t_1^0} \quad \cdots \quad \frac{\partial \gamma_t^0}{\partial t_i^0} \quad \cdots \quad \frac{\partial \gamma_t^0}{\partial t_m^0} \right] \quad (4.62)$$

and

$$\mathbf{J}_r = \left[ \frac{\partial \gamma_r^0}{\partial q_1^0} \quad \cdots \quad \frac{\partial \gamma_r^0}{\partial q_i^0} \quad \cdots \quad \frac{\partial \gamma_r^0}{\partial q_m^0} \right] \quad (4.63)$$

with the partial derivatives for the *Jacobian matrix*  $\mathbf{J}_t$  according to the initial translation components for an individual *exterior orientation* of the constraints  $\gamma_t^0$ , see Equation (4.53)

$$\frac{\partial \gamma_t^0}{\partial t_i^0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.64)$$

an moreover for the *Jacobian matrix*  $\mathbf{J}_r$  according to the initial rotations components of the constraints  $\gamma_r^0$ , see Equation (4.54)

$$\frac{\partial \gamma_r^0}{\partial q_i^0} = \begin{bmatrix} 2q_{i_x}^0 & 2q_{i_0}^0 & 2q_{i_z}^0 & 2q_{i_y}^0 & 0 & 0 & 0 \\ -2q_{i_y}^0 & 2q_{i_z}^0 & -2q_{i_0}^0 & 2q_{i_x}^0 & 0 & 0 & 0 \\ 2q_{i_z}^0 & 2q_{i_y}^0 & 2q_{i_x}^0 & 2q_{i_0}^0 & 0 & 0 & 0 \end{bmatrix} \quad (4.65)$$

for the definition of the *geodetic datum* based on Equations (4.53) and (4.54) are obtained. Thus the

overall *Jacobian matrix* considering all *constraint equations* reads

$$\mathbf{J}_c = \begin{bmatrix} \mathbf{J}_n & \mathbf{J}_t & \mathbf{J}_r \end{bmatrix}^T. \quad (4.66)$$

Using the *Jacobians* with  $\mathbf{A} = \mathbf{J}_a$ ,  $\mathbf{B} = \mathbf{J}_b$  and  $\mathbf{C} = \mathbf{J}_c$  and forming the *vectors of misclosures*

$$\mathbf{w}_r = \begin{bmatrix} \gamma_n^0 & \gamma_t^0 & \gamma_r^0 \end{bmatrix}^T \quad \text{and} \quad \mathbf{w} = -\mathbf{B}\mathbf{v}^0 + \Psi^0 \quad (4.67)$$

according to [Equations \(1.53\)](#) and [\(1.66\)](#) using the *condition* and *constraint vectors* in [Equation \(4.41\)](#) and [Equations \(4.50\)](#), [\(4.53\)](#) and [\(4.54\)](#), the solution for the vector of reduced unknowns

$$\Delta \hat{\mathbf{x}} = \begin{bmatrix} \dots & \Delta \hat{q}_{i_0} & \Delta \hat{q}_{i_x} & \Delta \hat{q}_{i_y} & \Delta \hat{q}_{i_z} & \Delta \hat{t}_{i_x} & \Delta \hat{t}_{i_y} & \Delta \hat{t}_{i_z} & \dots \end{bmatrix}^T \quad (4.68)$$

is obtained taking [Equation \(1.73\)](#). The results of the registration procedure are presented in [Section 4.7](#) with the *outer orientation* parameters in [Table 4.4](#).

NOTE: According to [Section 4.4](#) and the unknowns  $\hat{\mathbf{x}}$  and corrections  $\hat{\mathbf{v}}$  based on [Equation \(1.61\)](#) are to be introduced into the adjustment as new approximate values  $\mathbf{x}^0$  and  $\mathbf{v}^0$ . This iterative calculation is carried out until a selected break condition, cf. [Equation \(1.62\)](#), is reached. After each convergence for the actually observation material, BAARDA's *data snooping* (BAARDA 1968), see [Section 1.8.8.2](#), was applied to detect and remove outliers in form of incorrect or improper plane matches.

#### 4.6 ASSIGNMENT PROCESS TO ENTIRE SEGMENTS

The matching procedure in [Section 4.4](#) automatically provides direct correspondences of the planes of adjacent *scan* or *point cloud* pairs segmented with the subsegmentation algorithm in [Section 4.3.2](#). Since the planar subsegments are derived from the segments of the segmentation algorithm in [Section 4.3.1](#), direct correspondences of these local planes to the original segment are given, reflecting the structure in [Figure 4.7](#).

The plane indices `plane_ids` of the planes emerging from a specific region can be obtained from [Code 4.7](#) taking e. g. region with `region_id = 3` by considering the structures of a region, see also [Figure 3.15](#), and a plane, see also [Figure 4.7](#), as follows.

Code 4.7: Access the indices of planes corresponding to a specific region.

```
1 plane_ids = find(extractfield(plane_regions, 'region_id') == 3);
```

#### 4.6. ASSIGNMENT PROCESS TO ENTIRE SEGMENTS

Consequently, the correspondences of the segments of different *point clouds* containing one or more matches of subplanes are assigned automatically.

Figure 4.10 shows in row 1 a selection of main segments to be merged in the colors yellow, green, red and orange, originating from the *point cloud* segmentation of viewpoints 2, 1 and 10, see also Figures 4.10A to 4.10C. The line below indicates the sub-segments resulting from these main segments in the shape of planar areas, see Figures 4.10D to 4.10F, whereas the patches filled out in the respective color indicate direct plane matches among the *point clouds*. There may also be so-called direct correspondences, whose region of origin corresponds to a single plane match, as far as the criteria for a plane according to the definition are fulfilled, e.g. the respective yellow and green segments in scan 2 and 1.

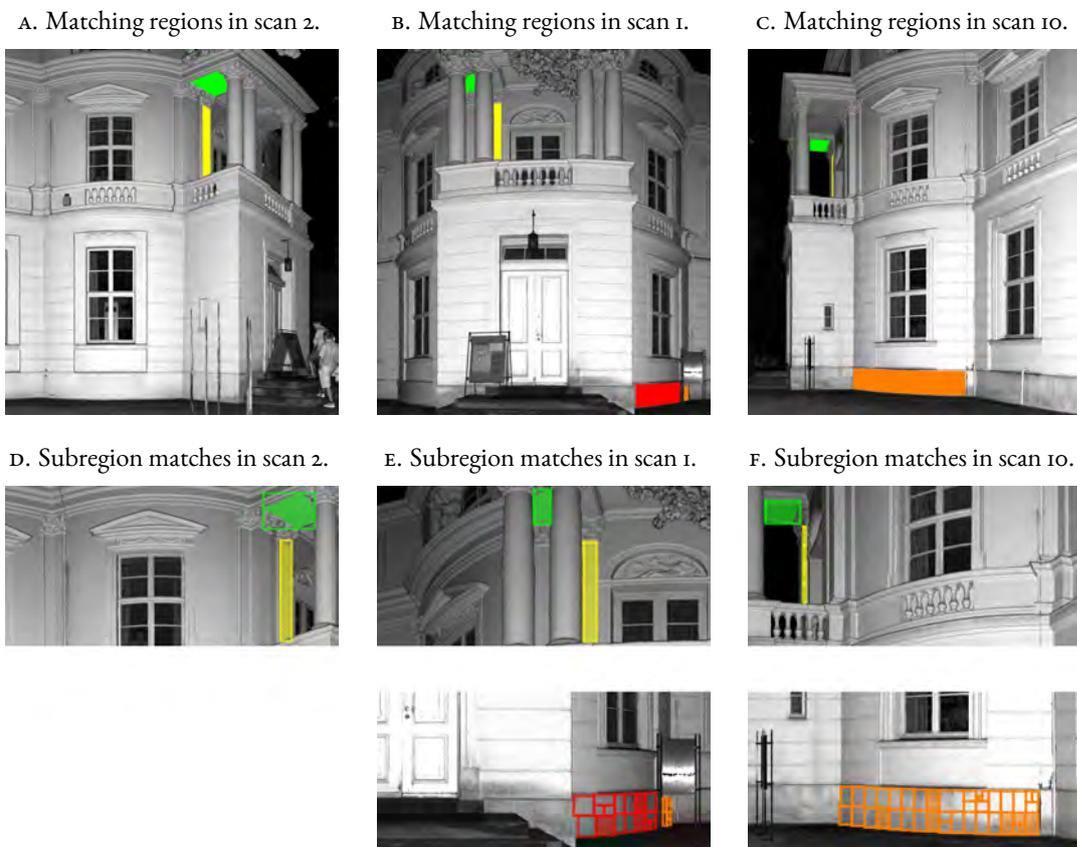


Figure 4.10: Assignment of individual segments to entire segments based on matching of subsegments.

Looking at the separation into planes for the lower partial surface, see Figure 4.10E and Figure 4.10F, shown in a red and orange partial area for scan 1, as well as a larger orange one in scan 10, the following matches among the planes result, indicated by the color-filled areas:

- plane 2661 in scan 1 matches plane 5041 in scan 10

- plane 2667 in scan 1 matches plane 5047 in scan 10
- plane 2669 in scan 1 matches plane 5049 in scan 10
- plane 2676 in scan 1 matches plane 5989 in scan 10
- plane 2678 in scan 1 matches plane 5993 in scan 10
- plane 2685 in scan 1 matches plane 5990 in scan 10
- plane 2690 in scan 1 matches plane 5992 in scan 10
- plane 2690 in scan 1 matches plane 5993 in scan 10

NOTE: Following the definition of a plane match, see also [Code 4.1](#) or [Equation \(4.1\)](#), under certain circumstances, as can be seen above, several matches can be created between planes if they are geometrically close together.

Code 4.8: Matching of regions from individual point clouds based on matching planes.

```

1 function r_matches = match_regions(p_viewpoint, p_matches, m_pairs)
2 %input: - p_viewpoint{i} cell array of planes,
3 %       where i is the scan number, see match_pairs
4 %       - e.g. m_pairs = [1 2;1 3;2 3];
5 %       corresp. to scan overlap matches 1-2, 1-10, 2-10
6 %       - p_matches of planes in scan pairs, see match_pairs
7 %output: - r_matches of regions in scans
8
9 n_match_pairs = numel(m_pairs(:,1));
10 n_all_matches = sum(cellfun(@numel,p_matches)) / 2;
11 n_stations = numel(p_viewpoint);
12 r_matches = zeros(n_all_matches,n_stations);
13 idx = 0;
14
15 %% build region matches
16 for i = 1:n_match_pairs
17     n_matches = numel(p_matches{1,i}(:,1));
18     for j = 1:n_matches
19         idx = idx+1;
20         p_scan_t = p_viewpoint{m_pairs(i,1)}(p_matches{1,i}(j,1));
21         p_scan_s = p_viewpoint{m_pairs(i,2)}(p_matches{1,i}(j,2));
22         r_matches(idx,m_pairs(i,1)) = p_scan_t.region_id;
23         r_matches(idx,m_pairs(i,2)) = p_scan_s.region_id;
24     end
25 end
26
27 [r_matches,~,~] = unique(r_matches,'rows');
```

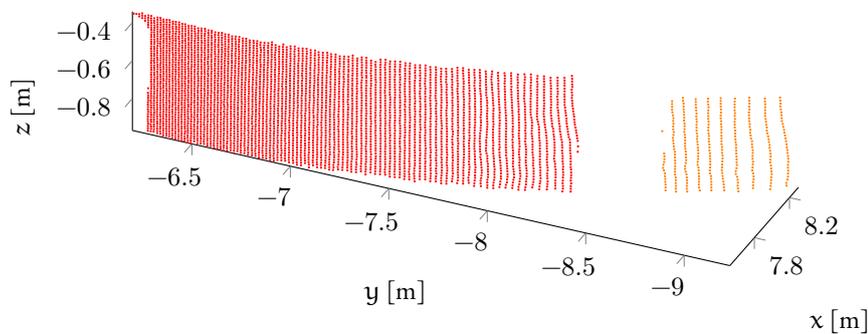
#### 4.6. ASSIGNMENT PROCESS TO ENTIRE SEGMENTS

By knowing the assignment of the planes ids to the respective region via the corresponding `region_id`, from which the planes are derived, the `function match_regions`<sup>20</sup>, see [Code 4.8](#), can now be used to form correspondences, stored in `r_matches`, between the individual main segments of the scans:

- region **18** in scan 1 matches region **3** in scan 10
- region **231** in scan 1 matches region **3** in scan 10

Consequently, the same correspondence of the regions with the id 18 and 231 in scan 1 to the similar region with the id 3 in scan 10 allows to form a uniform region and to merge the area covered by the sign in scan 1, reflected in 2 subregions (red and orange), see [Figure 4.10B](#).

A. Main regions from scan station 1.



B. Combined regions from station 1 and 10.

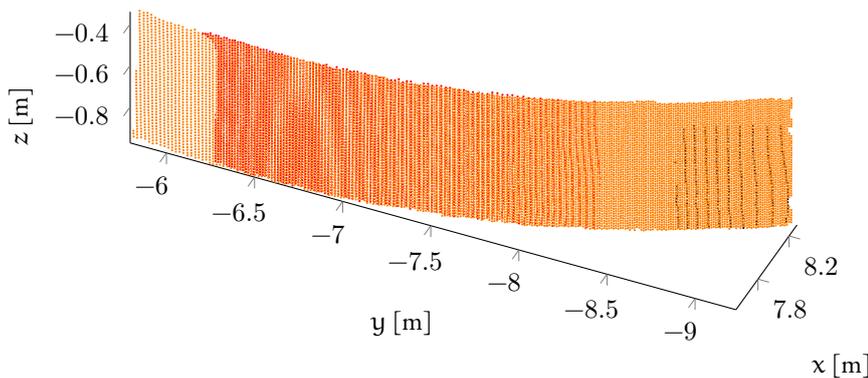


Figure 4.11: Detailed views of the region parts of a surface patch captured from scan station 1 and 10 and its combination to an entire region.

<sup>20</sup>Uses `p_matches` with planar matches for *point cloud* pairs, build by `function match_planes`, see [Code 4.6](#)

By means of the final registration parameters, the *outer orientations* in Table 4.4, the individual local partial segments, derived from each *point cloud* of a scan, can be transformed into a common, superordinate coordinate system in the same manner as the whole *point clouds* of the standpoints.

Ultimately, the partial segments can be joined together in the registered superior *point cloud* via the matching information to entire segments, obtaining a uniform overall segmentation in addition to a complete registered *point cloud*. Figure 4.11<sup>21</sup> shows the union of the individual segments of the orange and red regions from the above example, see Figure 4.10B and Figure 4.10C.

#### 4.7 RESULTS

The procedure for synergetic segmentation and registration presented in Sections 4.3 to 4.5 was applied to the three *point clouds* of the standpoints, see also Figure 4.1 and Figure 4.2, of the object "Belvedere". In the following, some of the obtained results are examined in more detail.

An indication of the result of the overall registration shall initially be given by an overview based on various cross sections normal to the Z-axis in the XY-plane at different heights, see Section 4.7.



Figure 4.12: Horizontal cross sections to validate the registration result.

The individual sections through the registered *point clouds* are shown in Figures 4.13A to 4.13C, with the points marked in green representing the object scanned from station 2, the red ones that were scanned from station 1 and the blue ones those scanned from station 10, see also Figure 4.1.

In each of the illustrations, the more present color indicates which standpoint has mainly contributed to the completion of the object through the different perspective on the object and the associated point density in certain areas.

<sup>21</sup>The orange points in Figure 4.11A are depicted in black for Figure 4.11B to ensure better visualisation.

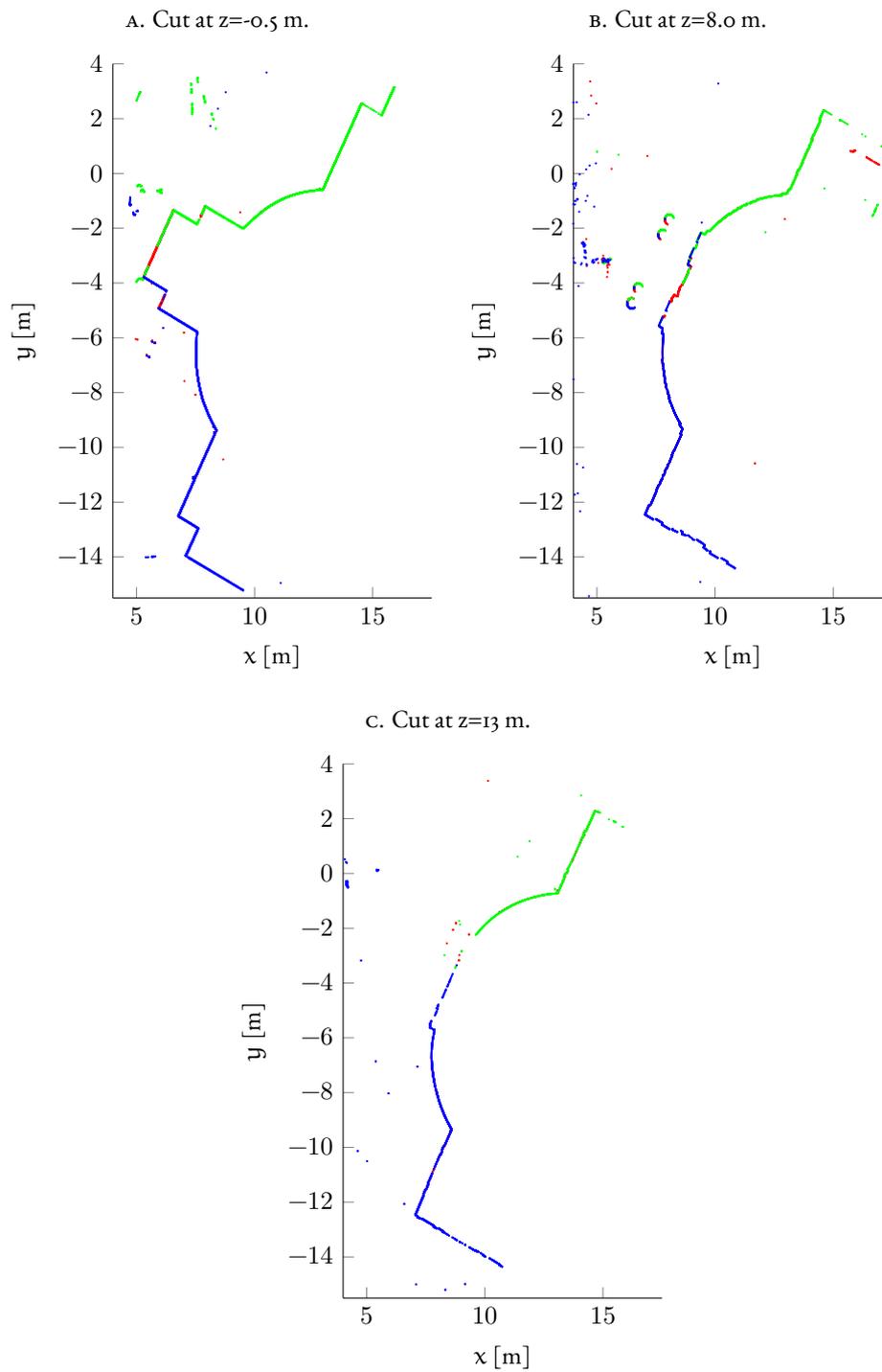


Figure 4.13: Individual cuts from the registered *point cloud* at different heights.

For a detailed evaluation of the registration result, two spatially distributed segments of different geometry types (area in orange and pillar in red) in the three adjacent scans were assigned to each other in [Figure 4.14](#) according to the principle described in [Section 4.6](#).

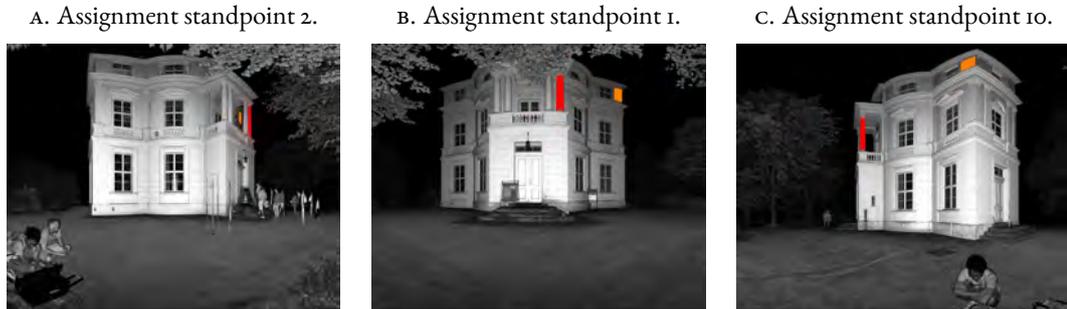


Figure 4.14: Assigned segments (plane in orange and pillar in red) for the *point clouds* of the positions 2 (left), 1 (middle) and 10 (right).

[Figure 4.15](#) shows a more detailed analysis of the planar region captured from the three different perspectives of the standpoints, see [Figures 4.14A](#) to [4.14C](#), and depicted in orange.

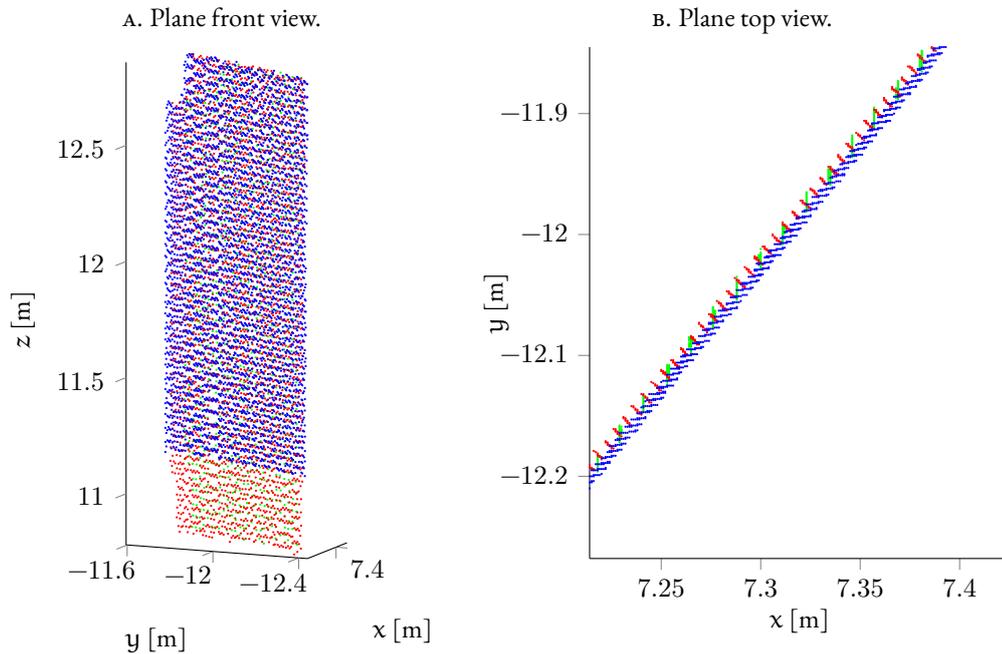


Figure 4.15: Detailed views (front view and top view) of the points of the planar region (depicted in the respective standpoint color based on [Figure 4.1](#)) merged from the three perspectives (depicted in orange) in [Figures 4.14A](#) to [4.14C](#).

Figure 4.16 shows a more detailed analysis of the pillar captured from the three different perspectives of the standpoints, see Figures 4.14A to 4.14C, and depicted in red.

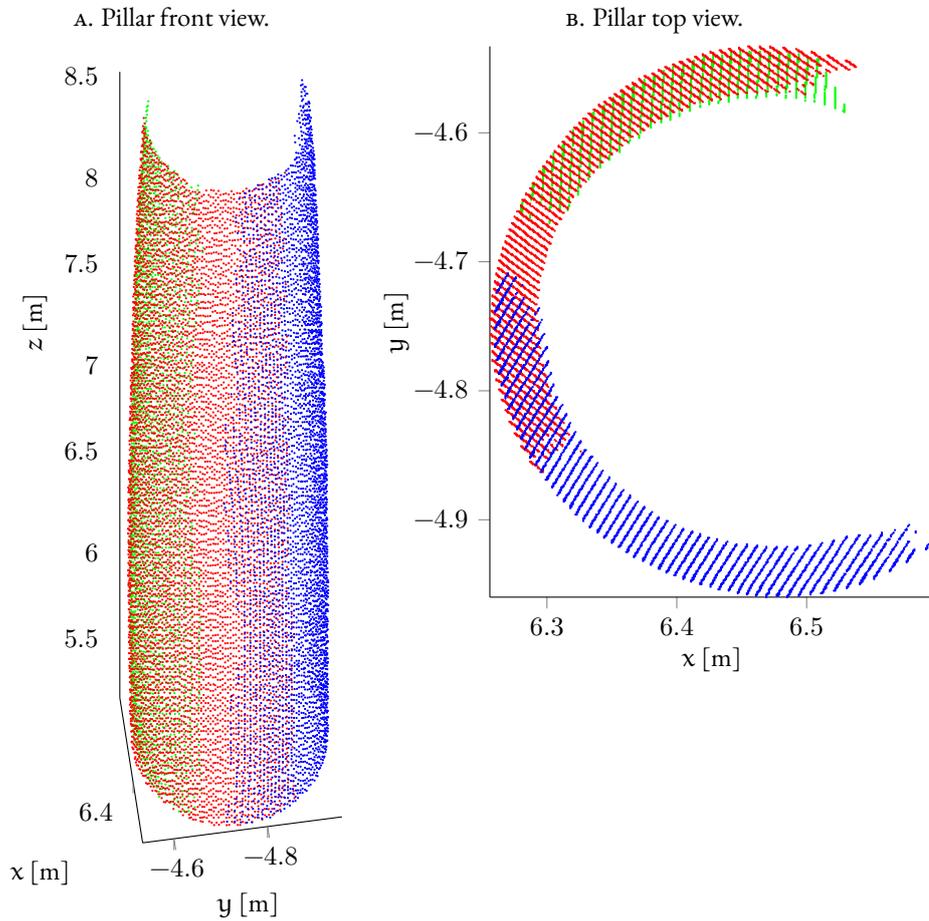


Figure 4.16: Detailed views (front view and top view<sup>22</sup>) of the points of the pillar (depicted in the respective standpoint color based on Figure 4.1) merged from the three perspectives (depicted in red) in Figures 4.14A to 4.14C.

On the one hand, Figures 4.15A and 4.16A illustrate the merging of the subsegments resulting from the different perspectives and related occlusions into one overall segment. The green colored areas represent those captured from position 2 (Figure 4.14A) the red colored areas represent those captured from position 1 (Figure 4.14B) and the blue colored areas represent those captured from position 10 (Figure 4.14C), see also Figure 4.1. If, on the other hand, one considers the respective top views in

<sup>22</sup>As the pillar tapers, its top view results in a wider ring.

Figures 4.15B and 4.16B of the segments merged via the registration, see the results in Table 4.4, one can conclude that the subsegments in the overlap area superimpose each other within the measurement noise of the TLS employed. The quality of the unification of the planar region in Figure 4.15A also appears to be in the range of the measurement noise, even though the segment is far outside the main overlapping and thus matching range of scan 1 and 2 of the respective positions, see Figures 4.14A and 4.14B. Both are important factors for a precise and reliable registration.

For a more detailed assessment of the exterior orientations, see also Section 1.3.1, Table 4.4 lists the results of the registration determined by the procedure in Section 4.5. For the calculated *rotation angles*<sup>23</sup>  $\hat{\omega}$ ,  $\hat{\phi}$ ,  $\hat{\kappa}$  and the translation parameters  $\hat{t}_x$ ,  $\hat{t}_y$ ,  $\hat{t}_z$  their SD is given.

Table 4.4: Values and Standard Deviations (SDs) for the *exterior orientations* of the individual scanner standpoints after the registration process of Section 4.5.

SCAN POSITION	EXTERIOR ORIENTATION PARAMETERS					
	$\hat{\omega}$ [°]	$\hat{\phi}$ [°]	$\hat{\kappa}$ [°]	$\hat{t}_x$ [m]	$\hat{t}_y$ [m]	$\hat{t}_z$ [m]
	$\sigma_{\omega}$	$\sigma_{\phi}$	$\sigma_{\kappa}$	$\sigma_{t_x}$	$\sigma_{t_y}$	$\sigma_{t_z}$
2	-0.327306	-0.003297	-32.153729	6.9690	7.2250	0.0027
	0.002556	0.002691	0.004260	0.0020	0.0023	0.0003
1	0.030773	0.033118	0.005023	-0.0072	-0.014	0.0211
	0.002086	0.002486	0.001029	0.0014	0.0027	0.0003
10	0.282756	-0.055904	-41.300534	0.4939	-14.0685	0.1607
	0.002086	0.002486	0.001029	0.0014	0.0027	0.0003

Table 4.4 indicates that the positions (translations to the *superordinate coordinate system*, see also Section 1.3.1)  $\hat{t}_x$ ,  $\hat{t}_y$ ,  $\hat{t}_z$  of the standpoints were determined with a SD in the millimetre to tenth of a millimetre range, which corresponds to the magnitude of the global point precision of the TLS applied. The vertical component  $\hat{t}_z$  was consistently determined more precisely than the other two, which can be attributed to the dominant number of existing horizontally aligned plane matches, see Figure 4.8.

The rotation angles  $\hat{\omega}$ ,  $\hat{\phi}$ ,  $\hat{\kappa}$  with respect to the orientations of the standpoints have been determined with a SD in the range of a few angular seconds. Similar to the *bundle adjustment* in *photogrammetry* with image coordinates, the registration method presented in Section 4.5 allows the consideration of connections in the form of identical planes across several standpoints, having a positive effect on the determination of the *exterior orientations* due to geometric stabilization and increased redundancy.

<sup>23</sup>The conversion of the parameters from *rotation matrices* with *quaternions*, see also Equation (1.2.4), into the common EULER angles  $\hat{\omega}$ ,  $\hat{\phi}$ ,  $\hat{\kappa}$  is shown e. g. in (LUHMANN 2018, p. 65).

## 4.8 CONCLUSION

The concluding presentation of individual results in [Section 4.7](#) showed that, on the one hand, the information obtained by a presegmentation to individual scans in [Section 4.3.1](#) can be used in the form of a split into planes, see [Section 4.3.2](#), for a registration, see [Sections 4.4 to 4.5](#), and, on the other hand, the orientation and matching information derived from the registration can be used in [Section 4.6](#) to complete the segmentation result.

The highly efficient pre-segmentation algorithm, see [Section 4.3.1](#) and also [Chapter 3](#), which classifies regions independently of their geometry and size based on additional information, coupled with the precise registration procedure in [Section 4.5](#), which is independent of point density and distribution on the object, provide a complementary simultaneous solution to both problems.

The need for the time-consuming build-up of a topology in 3D, see also [Sections 1.4](#) and [3.3.2](#), is completely obsolete here with regard to the registered *point cloud*, hence leading to a much more efficient procedure compared to others. With the high precision, compare [Tables 4.3](#) and [4.4](#), and by *outlier detection*, cf. [Section 1.8.8.2](#), reliable registration based on planes, see section [Section 4.5](#), many *point clouds* of individual standpoints can be transferred very efficiently into a superordinate context.



# 5

## Conclusion and Outlook

WITH THE FURTHER DEVELOPMENT OF THE TLS and its associated more precise and discrete quasi-area acquisition methods, today's TLSs now have a very broad field of application in a variety of disciplines, see [Chapter 0](#). Accordingly, the *point clouds* resulting from the acquisition, as well as the processing methods to be applied to them from the core competence area of engineering geodesy, see [Section 0.1](#) and ([KUHLMANN et al. 2013](#)), as well as the segmentation and registration, are also subject to increasingly higher quality demands in the form of so-called quality models, see [Section 0.2.1](#). On the basis of the disadvantages of present methods, see [Section 0.1.1](#), in particular the detached processing of segmentation and registration without the use of a meaningful *stochastic model* for a quality characteristic, the main research topics emerged in [Section 0.3](#). In [Chapter 1](#), fundamental topics relevant within the main chapters with regard to the recording and evaluation of TLS measurements were addressed first. Finally, with [Chapter 2](#) it was possible for the first time to derive and validate an all-encompassing *stochastic model* based solely on the intensity measurement of a TLSs using a suitable measurement and evaluation concept, considering all environmental influences affecting the reflectorless distance measurement. The segmentation procedure in [Chapter 3](#), operating on the natural data structure of *point clouds* and thus being very efficient, is capable of processing both geometric and radiometric data as well as in combination of a TLS to effectively subdivide individual *point clouds* of a scan into different geometrically related regions, such as planes, cylinders or generally curved surfaces, without any additional information. Following on this, in [Chapter 4](#) the segments from the pre-segmentation in [Section 4.3.1](#) could be exploited to identify significant planes, see [Section 4.3.2](#) using stochastic information from the *stochastic model* derived in [Section 2.3](#). These were subsequently used as corresponding information within

## CONCLUSION AND OUTLOOK

overlapping *point clouds* from multiple viewpoints for registration into a common coordinate system, see [Section 4.5](#), whereby the matching information could in turn be used to complete the individual segmentations from [Section 4.3.1](#) to form an overall segmentation, see also [Section 4.6](#). The procedure for the combined segmentation and registration of *laser scans* was called *Synergetic Segmentation and Registration* due to the mutual use and knowledge gain of information.

### 5.1 CONTRIBUTION TO SCIENCE AND ENGINEERING

The present thesis contributes decisively to two main research foci, see also [Section 0.3](#), within the core competencies of engineering geodesy, see among others ([KUHLMANN et al. 2013](#)), in the context of **TLS**:

1. Derivation and validation of a new and comprehensive *stochastic model* for **TLS**, see [Chapter 2](#).
2. *Synergetic segmentation and registration* of **TLS point clouds**, see [Chapter 4](#).

However, the fact that stochastic properties of **TLSs** can be derived from simply the raw intensity values while reflecting any environmental influences affecting the reflectance led to the first development of a meaningful *stochastic model* in [Chapter 2](#), contributing to the complement of previous quality models for **TLSs**, see also [Section 0.2.1](#), especially [Table 1](#). Due to the facile transferability and applicability of the findings to other scanner types, the new *stochastic model* has already gained wide acceptance within engineering and applied disciplines of geodesy and beyond, see [Section 2.7.1](#).

- The knowledge about the accuracy, thus also the precision of observations, cf. [Section 0.2](#) and [Figure 2](#), takes an essential role with regard to the modeling of the quality of measuring instruments, in particular of **TLSs**, see also [Section 0.2.1](#), as well as the evaluation of measuring and evaluation methods since the early days of geodesy.
- With the new and meaningful *stochastic model* for **TLSs**, very realistic precision statements can now be met regarding the respective observations, in particular concerning the reflectorless distance measurement, cf. [Equation \(2.2\)](#), which, in turn, allows the full potential of the several evaluation strategies with respect to **TLS**, such as segmentation and registration methods, to be exploited for the first time.
- On the one hand, this can be decisive for the statistical statement of the affiliation of points to a region or especially to a plane in the case of a plane segmentation, see [Section 4.3.2](#), but also in the classical sense for the weighting of the observations in the course of a usual evaluation process in a **GMM** or **GHM**, e. g. also in case of registration, see [Equation \(4.24\)](#), and especially the associated detection of outliers in the data, see also [Section 1.8.8.2](#).
- Also e. g. with regard to a deformation analysis on the basis of **TLS** data, the knowledge of the precision about the **3D** points is crucial for a significant test decision whether and to which extent regions have deformed or changed, see also e. g. ([KAUKER and SCHWIEGER 2017](#); [ZHAO et al. 2019](#)), hence the new *stochastic model* is also of enormous importance in this context.

## 5.2. IDEAS FOR IMPROVEMENT AND RESEARCH PERSPECTIVE

DUE TO the complexity of many evaluation processes, a **VCP** over several process steps is often unavoidable and therefore the basic statement about the precision of the input measured values is even more important.

When looking at the previous detached and independent evaluation strategy for segmentation and registration processes, see [Section 0.1.1](#), a joint solution has been proposed for the first time in [Chapter 3](#) and [Chapter 4](#) with which both processing steps are carried out as efficiently as possible in combination with maximum precision taking advantage of mutual synergies.

- By using the original scan data with its natural data structure resulting from the creation process, see [Section 1.2.1](#) and [figure 1.3](#), a fast and well separated segmentation result on single scans can be achieved with the help of an extremely efficient **CCA** in [Section 3.3.2](#) basing on the geometric and radiometric information, whereas one could conceivably perform a pre-segmentation on the **CPU** of a **TLS** directly in the field and thus achieve a time saving in the overall evaluation chain.
- In [Section 4.3](#) of [Chapter 4](#), a new algorithm for the segmentation of planar patches based on individual regions from the pre-segmentation was presented thereupon, which, in contrast to other classical *split and merge* methods, is oriented by the pre-segmentation according to its natural, in the data present, features and thus also results in planes with an extension to the segment boundaries.
- The subsequent plane matching in [Section 4.4](#) serves not only for correspondence generation between planes of different *point clouds* for the plane-based registration in [Section 4.5](#), but is also equally information provider for the assignment procedure of the individual regions from the scan-based pre-segmentations for completion to overall regions, see [Section 4.6](#).
- Consequently, the plane-based segmentation with observation information in the form of plane correspondences contributes to the success of the registration on the one hand, and on the other hand the registration via the matching information participates in a uniform and complete segmentation result, whereby a segmentation on an already registered *point cloud*, which is usual in practice, accompanied by the again complex and time-consuming reconstruction of the topology, cf. also [Section 1.10](#), is omitted.

MOREOVER, by continuously carrying meaningful stochastic information based on the *stochastic model* throughout the entire evaluation process, false or unreliable plane correspondences can be stably and reliably detected and eliminated during registration, which has a positive effect on the reliability and precision of the registration result.

## 5.2 IDEAS FOR IMPROVEMENT AND RESEARCH PERSPECTIVE

A closer look at the current development of **TLSs** (cf. e. g. [JUNTILA et al. 2017](#); [LI et al. 2016](#); [MATIKAINEN et al. 2017](#)), shows that in the near future there will be *laser scanners* with several information channels due to the integration of diodes with different wavelengths. By the additional information gain of the intensity channels besides the distance channel, further perspectives and optimization possibilities for

## CONCLUSION AND OUTLOOK

the evaluation of *laser scans* will arise (KRUEGER et al. 2017), similarly as they are also currently possible e. g. after the fusion with camera images, see among others (LICHTI 2005; BRELL et al. 2017).

- Considering the matching process in Section 4.4, it is currently purely geometric based. However, applying additional intensity information or combinations of relative differences between various intensity channels on the one hand would reduce the search space for possible plane correspondences and exclude false correspondences more reliably as well. This would result in an increase of reliability as well as efficiency of the matching process.
- Even with regard to a classification of the individual scans into different object classes, for example according to the spectral characteristic of different materials as shown in Figure 2.10, the general segmentation algorithm from Chapter 3 can be applied directly without any adaptation. The creation of so-called *index layers* (radiometric coefficients) from a special combination of some intensity channels for each material or at least a material category or object class, e. g. vegetation, asphalt, stone, metal, etc., can be used to classify certain objects such as building materials, as for example in the case of a definition of an Normalized Differenced Vegetation Index (NDVI) in the context of a supervised or unsupervised image classification.

**SHORT REMARK:** For the use of several intensity channels within the classification or for the improvement of the matching process, a radiometric calibration per intensity channel is not mandatory, since by difference formation or index determination between various channels, only relative values are considered, having comparable values in a certain examination area, unlike absolute values in different scans. The prerequisite is that the environmental influences have a comparable effect on all diodes respectively intensity channels.

In accordance to the registration process in Section 4.5 and the quality modeling of TLSs, see Section 0.2.1, further optimizations regarding a comprehensive evaluation are conceivable:

- To cover the remaining systematics of a TLS system and to take them into account as far as possible in the evaluation, a so-called simultaneous calibration, as e. g. also mentioned in Section 1.2.5, is a suitable approach. *Systematic error* components of the TLS which are not completely covered during the *system calibration* by the manufacturer at delivery can thus ideally be determined in the context of a simultaneous calibration within the registration process and applied to the observations in order to enhance the *point cloud*. In the ideal case, only *random errors* in the sense of noise remain, represented solely by the new *stochastic model*.
- For an individual consideration also of other geometries depending on the presence in the data during the registration process, it is conceivable to reference via the general formulation of *second order surfaces* or *quadrics*, see also Section A.2. With a successive significance analysis, the parameters of the original model can be reduced until the model fits to the data. The increased flexibility and better, or more ideal and large-scale, fitting across multiple geometry models ideally allows a more reliable and even more accurate registration across multiple TLS viewpoints.



# Complementary Information

This additional chapter provides supporting information and supplementary material for the individual chapters [Chapters 1](#) to [4](#). Among them are basic mathematical relationships and source code as functions, which can be found in other listed functions as source code examples in the main chapters.

## A.1 BASIC MATHEMATICS

### A.1.1 RODRIGUES FORMULATION AND QUATERNIONS

The considerations that a rotation in  $\mathbb{R}^3$  can be expressed as a function of a single rotation angle  $\phi$  and the three components  $u_x$ ,  $u_y$ ,  $u_z$  of a unit vector  $\mathbf{u}$  along the rotation axis can be traced back to *Leonhard Euler* ([EULER 1775](#)) and *Olinde Rodrigues* ([RODRIGUES 1840](#)). The three orientation angles  $\alpha$ ,  $\beta$  and  $\gamma$  between the rotation axis and the base vectors are used to describe the position of the rotation axis, see [Figure A.1](#), according to which the depicted vector

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \cos \alpha \\ \cos \beta \\ \cos \gamma \end{bmatrix} \quad (\text{A.1})$$

is obtained.

COMPLEMENTARY INFORMATION

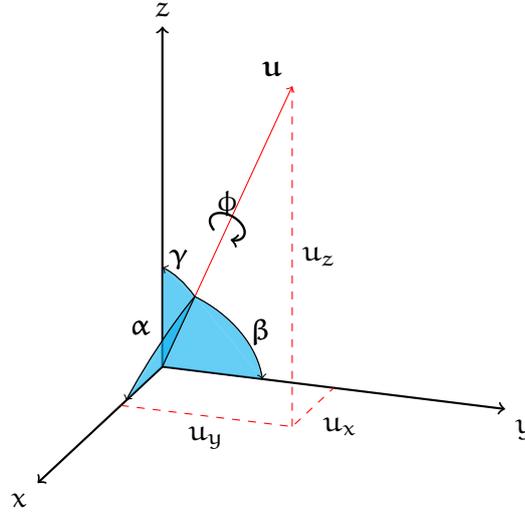


Figure A.1: Rodrigues rotation around a unit vector  $\mathbf{u}$ .

According to Rodrigues (RODRIGUES 1840) a vector  $\mathbf{x}$  in space can be rotated by the unit vector  $\mathbf{u}$ , see Equation (A.1), which represents the rotation axes, and the rotation angle  $\phi$  as follows:

$$\mathbf{x}_{\text{rot}} = \mathbf{x} + \sin \phi (\mathbf{u} \times \mathbf{x}) + (1 - \cos \phi) \mathbf{u} \times (\mathbf{u} \times \mathbf{x}) \quad (\text{A.2})$$

With the formation of a skew symmetric cross product matrix

$$\mathbf{U} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}, \quad (\text{A.3})$$

so that

$$\mathbf{U}\mathbf{x} = \mathbf{u} \times \mathbf{x} \quad (\text{A.4})$$

applies and the substitution of the equivalent expression in the Rodrigues formula, see (A.2), one obtains

$$\mathbf{x}_{\text{rot}} = \mathbf{x} + \sin \phi (\mathbf{U}\mathbf{x}) + (1 - \cos \phi) (\mathbf{U}^2\mathbf{x}). \quad (\text{A.5})$$

By excluding the vector  $\mathbf{x}$ , to be rotated under the subcondition

$$\|\mathbf{u}\| = u_x^2 + u_y^2 + u_z^2 = 1, \quad (\text{A.6})$$

the known rotation matrix is obtained in *Rodrigue* form

$$\mathbf{R}_{(\mathbf{u},\phi)} = \mathbf{I} + \sin \phi \mathbf{U} + (1 - \cos \phi) \mathbf{U}^2 = \mathbf{R}_{(u_x, u_y, u_z, \phi)} = \begin{bmatrix} u_x^2 + (1 - u_x^2) c\phi & u_x u_y (1 - c\phi) - u_z s\phi & u_x u_z (1 - c\phi) + u_y s\phi \\ u_x u_y (1 - c\phi) + u_z s\phi & u_y^2 + (1 - u_y^2) c\phi & u_y u_z (1 - c\phi) - u_x s\phi \\ u_x u_z (1 - c\phi) - u_y s\phi & u_y u_z (1 - c\phi) + u_x s\phi & u_z^2 + (1 - u_z^2) c\phi \end{bmatrix}, \quad (\text{A.7})$$

where  $s\phi = \sin \phi$  and  $c\phi = \cos \phi$  respectively. By means of the double trigonometric functions

$$\sin 2\phi = 2 \sin \phi \cos \phi \quad \cos 2\phi = 2 \cos^2 \phi - \sin^2 \phi$$

and the trigonometric *Pythagoras*<sup>†</sup>

$$\sin^2 \phi + \cos^2 \phi = 1,$$

the following connections

$$\begin{aligned} \sin \phi &= 2 \sin \frac{\phi}{2} \cos \frac{\phi}{2} \\ \cos \phi &= \cos^2 \frac{\phi}{2} - \sin^2 \frac{\phi}{2} \\ 1 - \cos \phi &= 2 \sin^2 \frac{\phi}{2} \end{aligned} \quad (\text{A.8})$$

can be indicated, Substituting the equivalent expressions in (A.7) according to the relations, see [Section A.1.1](#), and considering that a *Quaternion*  $\mathbf{q} = [q_0, q_x, q_y, q_z]^T$  can be represented with the *Rodrigue* parameters (cf. [DAI 2015](#); and [PUJOL et al. 2012](#))

$$\mathbf{q} = \cos \frac{\phi}{2} + \mathbf{u} \sin \frac{\phi}{2},$$

<sup>†</sup> *Phytagoras* (\* around 570 B.C.; † after 510 B.C.) was an ancient Greek philosopher and pioneer of mathematics and science. The well-known *Phytagoras theorem* was named after him.

COMPLEMENTARY INFORMATION

we obtain the rotation matrix expressed by *Quaternions*<sup>2</sup>

$$\mathbf{R}_{(q_0, q_x, q_y, q_z)} = \begin{bmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_x q_y + q_0 q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_0 q_x) \\ 2(q_x q_z - q_0 q_y) & 2(q_y q_z + q_0 q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}, \quad (\text{A.9})$$

subject to the auxiliary condition

$$\|\mathbf{q}\| = q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1. \quad (\text{A.10})$$

A.1.2 INTERRELATION BETWEEN CARTESIAN AND POLAR COORDINATES

The relationship between cartesian and polar coordinates can be illustrated rather simple by [Figure A.2](#) using two schematic sketches.

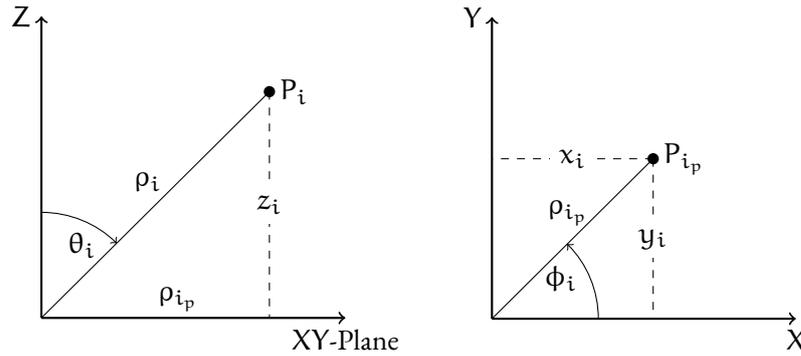


Figure A.2: Mathematical relationship between cartesian and polar coordinates.

Concerning the left part of [Figure A.2](#), the formulas

$$z_i = \rho_i \cos \theta_i \quad \text{and} \quad \rho_{i_p} = \rho_i \sin \theta_i \quad (\text{A.11})$$

can be derived and equivalently for the right part the formulas

$$x_i = \rho_{i_p} \cos \phi_i \quad \text{and} \quad y_i = \rho_{i_p} \sin \phi_i. \quad (\text{A.12})$$

<sup>2</sup> The *Quaternions* are also referred to as HAMILTON *Quaternions* or HAMILTON numbers according to Sir *William Rowan Hamilton*, but they were independently discovered three years earlier by *Olinde Rodrigues* (RODRIGUES 1840).

From (A.11) and (A.12), the conversion from polar to cartesian coordinates can be given with

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} \rho_i \sin \theta_i \cos \phi_i \\ \rho_i \sin \theta_i \sin \phi_i \\ \rho_i \cos \theta_i \end{bmatrix}. \quad (\text{A.13})$$

Moreover the back conversion from cartesian to polar coordinates can be performed by

$$\begin{aligned} \theta_i &= \arccos \frac{z_i}{\rho_i}, \\ \phi &= \arctan \frac{y_i}{x_i} \quad \text{and} \\ \rho_i &= \sqrt{x_i^2 + y_i^2 + z_i^2} \end{aligned} \quad (\text{A.14})$$

### A.I.3 POINT NORMAL CALCULATION WITHIN GRIDDED DATA USING SVD

Similar to Section 3.2, the point normal calculation with SVD is based on the successive analysis and evaluation of small data areas. Due to the gridded data structure, see also Figure 3.2, the same evaluation principle can be applied as for filtering with Algorithm 3.1. The red mask in Figure 3.5 is shifted consecutively and pixel by pixel (for each position  $i, j$ ) over the entire grid  $\mathbf{M}$ , see Figure 3.4, and simultaneously for each layer  $\mathbf{M} = \mathbf{X}, \mathbf{M} = \mathbf{Y}, \mathbf{M} = \mathbf{Z}$  from Figure 3.2. At each position  $(i, j)$  the coordinate values

$$\begin{aligned} \mathbf{x}_{ij} &= \begin{bmatrix} \mathbf{X}_{j-1}^{i-1} & \mathbf{X}_j^{i-1} & \mathbf{X}_{j+1}^{i-1} & \mathbf{X}_{j-1}^i & \mathbf{X}_j^i & \mathbf{X}_{j+1}^i & \mathbf{X}_{j-1}^{i+1} & \mathbf{X}_j^{i+1} & \mathbf{X}_{j+1}^{i+1} \end{bmatrix}^T \\ \mathbf{y}_{ij} &= \begin{bmatrix} \mathbf{Y}_{j-1}^{i-1} & \mathbf{Y}_j^{i-1} & \mathbf{Y}_{j+1}^{i-1} & \mathbf{Y}_{j-1}^i & \mathbf{Y}_j^i & \mathbf{Y}_{j+1}^i & \mathbf{Y}_{j-1}^{i+1} & \mathbf{Y}_j^{i+1} & \mathbf{Y}_{j+1}^{i+1} \end{bmatrix}^T \\ \mathbf{z}_{ij} &= \begin{bmatrix} \mathbf{Z}_{j-1}^{i-1} & \mathbf{Z}_j^{i-1} & \mathbf{Z}_{j+1}^{i-1} & \mathbf{Z}_{j-1}^i & \mathbf{Z}_j^i & \mathbf{Z}_{j+1}^i & \mathbf{Z}_{j-1}^{i+1} & \mathbf{Z}_j^{i+1} & \mathbf{Z}_{j+1}^{i+1} \end{bmatrix}^T \end{aligned} \quad (\text{A.15})$$

lying within the mask of size  $3 \times 3$  are taken, reduced to their center of gravity<sup>3</sup>

<sup>3</sup> Assuming that the points represent a plane of the form according to Equation (2.15), the distance  $d$  to the origin can be eliminated by the centroid reduction of the points therein, so that only the information of the normal vector is contained in the centroid reduced data.

## COMPLEMENTARY INFORMATION

$$\begin{aligned}
 \mathbf{x}_{ij_{cg}} &= \mathbf{x}_{ij} - w_{n_k} \sum \mathbf{x}_{ij} \\
 \mathbf{y}_{ij_{cg}} &= \mathbf{y}_{ij} - w_{n_k} \sum \mathbf{y}_{ij} \\
 \mathbf{z}_{ij_{cg}} &= \mathbf{z}_{ij} - w_{n_k} \sum \mathbf{z}_{ij},
 \end{aligned}
 \tag{A.16}$$

where  $w_{n_k} = \frac{1}{9}$  is the weighting factor for the  $n_k = 9$  elements for the corresponding mask size, and summarized in a matrix

$$\mathbf{D} = \begin{bmatrix} \mathbf{x}_{ij_{cg}} & \mathbf{y}_{ij_{cg}} & \mathbf{z}_{ij_{cg}} \end{bmatrix}.
 \tag{A.17}$$

A Singular Value Decomposition ([SVD](#)), see also [GOLUB and VAN LOAN \(2013\)](#) p.76 ff., can now be performed on  $\mathbf{D}^T \mathbf{D}$  with

$$\mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \text{SVD}(\mathbf{D}^T \mathbf{D}),
 \tag{A.18}$$

where

$\mathbf{U}$ ,  $\mathbf{V}$  are the left and right orthogonal matrices containing the respective singular vectors and  $\mathbf{\Sigma}$  is the diagonal matrix comprising the singular values to its singular vectors.

The singular vector  $\mathbf{s}_v$  belonging to the smallest singular value in  $\mathbf{\Sigma}$  from the right orthogonal matrix (last column of  $\mathbf{V}$ ), contains the desired normal vector information  $\mathbf{n}$ , which represents the point normal of the reference point within a mask. For each mask position, the calculated normal vector components are stored in a separate layer at the same position (mask center point), as in [Figure 3.8](#).

**REMARK:** The [SVD](#) can be reached by solving an Eigen Value Decomposition ([EVD](#)) on  $\mathbf{D}^T \mathbf{D}$ , since singular values have a direct relationship to eigenvalues ([GOLUB and VAN LOAN 2013](#), p. 77).

### A.2 TRANSFORMATION OF ARBITRARY SURFACES

The transformation of planar surface elements, see [Section 4.5](#), can be derived from a general surface in  $\mathbb{R}^3$ . In [3D](#) space, a so-called *quadric* can be used to describe an arbitrary surface in space, which is also called *second order surface* or *quadratic surface*:

$$f(\mathbf{X}) = \mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{b}^T \mathbf{X} + c = 0
 \tag{A.19}$$

where

## A.2. TRANSFORMATION OF ARBITRARY SURFACES

$\mathbf{X} = [X, Y, Z]^T$  is a transformed point via (A.21) into a specific coordinate system satisfying (A.19),  
 $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$  is a symmetric matrix containing the coefficients for the quadratic part,  
 $\mathbf{b} = [b_1, b_2, b_3]^T$  is the vector containing the linear part and  
 $c$  is the constant part of (A.19).

Due to the symmetry of  $\mathbf{A}$ , Equation (A.19) is formulated into

$$a_{11}X^2 + a_{22}Y^2 + a_{33}Z^2 + 2a_{12}XY + 2a_{13}XZ + 2a_{23}YZ + p_1X + p_2Y + p_3Z + c = 0. \quad (\text{A.20})$$

From the general design of a quadric, various objects in space, including cylinders, cones, planes, hyperboloid, etc., but also curved surfaces, can be derived or described by reducing (A.20) by a few parameters. From the transformation of a point  $\mathbf{x}$  into a superordinate coordinate system

$$\mathbf{X} = \mathbf{R}\mathbf{x} + \mathbf{t}, \quad (\text{A.21})$$

the quadric described by (A.19) can be reformulated as follows

$$f(\mathbf{R}\mathbf{x} + \mathbf{t}) = (\mathbf{x}^T \mathbf{R}^T + \mathbf{t}^T) \mathbf{A} (\mathbf{R}\mathbf{x} + \mathbf{t}) + 2\mathbf{b}^T (\mathbf{R}\mathbf{x} + \mathbf{t}) + c, \quad (\text{A.22})$$

which leads to

$$\mathbf{t}^T \mathbf{A} \mathbf{t} + \mathbf{t}^T \mathbf{A} \mathbf{R} \mathbf{x} + \mathbf{x}^T \mathbf{R}^T \mathbf{A} \mathbf{t} + \mathbf{x}^T \mathbf{R}^T \mathbf{A} \mathbf{R} \mathbf{x} + 2\mathbf{b}^T \mathbf{t} + 2\mathbf{b}^T \mathbf{R} \mathbf{x} + c. \quad (\text{A.23})$$

Assuming symmetric, so  $\mathbf{A} = \mathbf{A}^T$  as mentioned above, we can say that

$$\mathbf{x}^T \mathbf{R}^T \mathbf{A} \mathbf{t} = (\mathbf{x}^T \mathbf{R}^T \mathbf{A} \mathbf{t})^T \quad (\text{A.24})$$

and moreover

$$(\mathbf{x}^T \mathbf{R}^T \mathbf{A} \mathbf{t})^T = \mathbf{t}^T \mathbf{A} \mathbf{R} \mathbf{x}, \quad (\text{A.25})$$

resulting by applying (A.23) to (A.26) in

$$\mathbf{t}^T \mathbf{A} \mathbf{t} + \mathbf{t}^T \mathbf{A} \mathbf{R} \mathbf{x} + \mathbf{t}^T \mathbf{A} \mathbf{R} \mathbf{x} + \mathbf{x}^T \mathbf{R}^T \mathbf{A} \mathbf{R} \mathbf{x} + 2\mathbf{b}^T \mathbf{t} + 2\mathbf{b}^T \mathbf{R} \mathbf{x} + c. \quad (\text{A.26})$$

COMPLEMENTARY INFORMATION

After summarizing only the linear terms of [Equation \(A.26\)](#), yields

$$2\mathbf{t}^T \mathbf{A} \mathbf{R} \mathbf{x} + 2\mathbf{b}^T \mathbf{R} \mathbf{x} = 2(\mathbf{t}^T \mathbf{A} + \mathbf{b}^T) \mathbf{R} \mathbf{x}, \quad (\text{A.27})$$

which leads to

$$\mathbf{b}^{*\top} = (\mathbf{t}^T \mathbf{A} + \mathbf{b}^T) \mathbf{R} \quad (\text{A.28})$$

and by transposing

$$(\mathbf{b}^{*\top})^T = (\mathbf{t}^T \mathbf{A} + \mathbf{b}^T) \mathbf{R}^T \quad (\text{A.29})$$

and reformulating leads to

$$\mathbf{b}^* = \mathbf{R}^T (\mathbf{A} \mathbf{t} + \mathbf{b}) \quad (\text{A.30})$$

and finally to the transformed parameters

$$\begin{aligned} \mathbf{A}^* &= \mathbf{R}^T \mathbf{A} \mathbf{R} \\ \mathbf{b}^* &= \mathbf{R}^T (\mathbf{A} \mathbf{t} + \mathbf{b}) \\ \mathbf{c}^* &= \mathbf{t}^T \mathbf{A} \mathbf{t} + 2\mathbf{b}^T \mathbf{t} + \mathbf{c} \end{aligned} \quad (\text{A.31})$$

Taking the transformed parameters from [\(A.31\)](#) into account, [Equation \(A.19\)](#) can be represented uniformly

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{A}^* \mathbf{x} + \mathbf{b}^{*\top} \mathbf{x} + \mathbf{c}^*, \quad (\text{A.32})$$

or equivalently in matrix form

$$\mathbf{x}^T \begin{bmatrix} \mathbf{c}^* & \mathbf{b}^{*\top} \\ \mathbf{b}^* & \mathbf{A}^* \end{bmatrix} \mathbf{x}. \quad (\text{A.33})$$

Considering the exclusively linear parameters  $\mathbf{b}^*$  with and  $\mathbf{c}^*$  one finally obtains

$$\mathbf{b}^{*\top} \mathbf{x} + \mathbf{c}^* \quad \text{with} \quad \mathbf{b}^* = \mathbf{R}^T (\mathbf{t} + \mathbf{b}) \quad \text{and} \quad \mathbf{c}^* = 2\mathbf{b}^T \mathbf{t} + \mathbf{c} \quad \text{excluding} \quad \mathbf{A}, \quad (\text{A.34})$$

corresponding to a transformed plane equation according to [Equations \(4.12\)](#) and [\(4.17\)](#).

REMARK: With the general formulation of the transformation of surfaces in space, any surface elements present in a *point cloud* can be estimated and used as identical information in the context of a registration process, see e. g. [Section 4.5](#).

### A.3 BASIC MATLAB CODE

#### A.3.1 GENERAL

Code A.1: Transforming 3D points to another coordinate system.

```

1 function XYZ = transform_points(xyz, R, T)
2 %input: - xyz(3,n) point list [x;y;z]
3 %      - R, T the transformation parameters
4 %output: - XYZ(3,n) the transformed points
5
6 XYZ = R*xyz;
7 XYZ(1,:) = XYZ(1,:)+T(1,1);
8 XYZ(2,:) = XYZ(2,:)+T(2,1);
9 XYZ(3,:) = XYZ(3,:)+T(3,1);

```

Code A.2: Expanding a matrix  $M$  for filtering purposes.

```

1 function [M, rows, cols] = expand_borders(M, wK)
2 %input: matrix M and kernel size wK
3 %output1: expanded matrix M for filtering with kernel wKxwK
4 %output2/3: valid filter indices
5
6 [m,n] = size(M);
7
8 % kernel center
9 k_c = wK/2+0.5;
10
11 rows = k_c:m+k_c-1; cols = k_c:n+k_c-1;
12
13 rep_1 = 3*M(1,:)-3*M(2,:)+M(3,:);
14 rep_2 = 3*M(m,:)-3*M(m-1,:)+M(m-2,:);
15 M = [repmat(rep_1,k_c-1,1);M;repmat(rep_2,k_c-1,1)];
16
17 rep_3 = 3*M(:,1)-3*M(:,2)+M(:,3);
18 rep_4 = 3*M(:,n)-3*M(:,n-1)+M(:,n-2);
19 M = [repmat(rep_3,1,k_c-1),M,repmat(rep_4,1,k_c-1)];

```

## COMPLEMENTARY INFORMATION

### A.3.2 PRECALCULATION

Code A.3: Calculation of the point normals by means of filtering.

```
1 function p_cloud = calc_point_normals(p_cloud, wK)
2 %input: struct p_cloud.x, p_cloud.y, p_cloud.z and kernel width wK
3 %output: struct p_cloud.nx, p_cloud.ny, p_cloud.nz
4
5 switch wK
6     case 3
7         mask_r = [1 0 -1]/2;
8         mask_c = [-1;0;1]/2;
9     case 5
10        mask_r = [1 1 0 -1 -1]/4;
11        mask_c = [-1;-1;0;1;1]/4;
12    case 7
13        mask_r = [1 1 1 0 -1 -1 -1]/6;
14        mask_c = [-1;-1;-1;0;1;1;1]/6;
15    case 9
16        mask_r = [1 1 1 1 0 -1 -1 -1 -1]/8;
17        mask_c = [-1;-1;-1;-1;0;1;1;1;1]/8;
18 end
19
20 % expand borders for filtering operation
21 [xx,rows,cols] = expand_borders(p_cloud.x,wK);
22 [yy,~,~] = expand_borders(p_cloud.y,wK);
23 [zz,~,~] = expand_borders(p_cloud.z,wK);
24
25 % compute matrix gradients
26 ax = filter2(mask_r,xx); ax = ax(rows,cols);
27 ay = filter2(mask_r,yy); ay = ay(rows,cols);
28 az = filter2(mask_r,zz); az = az(rows,cols);
29
30 bx = filter2(mask_c,xx); bx = bx(rows,cols);
31 by = filter2(mask_c,yy); by = by(rows,cols);
32 bz = filter2(mask_c,zz); bz = bz(rows,cols);
33
34 % compute cross product to get normals
35 nx = -(ay.*bz-az.*by); ny = -(az.*bx-ax.*bz); nz = -(ax.*by-ay.*bx);
36
37 mag = sqrt(nx.*nx+ny.*ny+nz.*nz);
38 d = find(mag==0); mag(d) = eps*ones(size(d));
39 p_cloud.nx = nx./mag; p_cloud.ny = ny./mag; p_cloud.nz = nz./mag;
```

Code A.4: Calculation of the point curvature by means of filtering.

```
1 function p_cloud = calc_point_curvature(p_cloud)
2 %input: struct p_cloud.nx, p_cloud.ny, p_cloud.nz
3 %output: struct p_cloud.c
```

```

4
5 xx = p_cloud.nx;
6 yy = p_cloud.ny;
7 zz = p_cloud.nz;
8
9 % expand borders for filtering operation
10 [xx, rows, cols] = expand_borders(xx,3);
11 [yy,~,~] = expand_borders(yy,3);
12 [zz,~,~] = expand_borders(zz,3);
13
14 % compute change for all layers
15 xx_filt = calc_change_in_region(xx,3);
16 yy_filt = calc_change_in_region(yy,3);
17 zz_filt = calc_change_in_region(zz,3);
18
19 % calculate curvature
20 p_cloud.c = sqrt(xx_filt.^2+yy_filt.^2+zz_filt.^2);
21
22 p_cloud.c = p_cloud.c(rows,cols);

```

Code A.5: Calculation of the point density by means of filtering.

```

1 function p_cloud = calc_point_density(p_cloud)
2 %input: struct p_cloud.x, p_cloud.y, p_cloud.z
3 %output: struct p_cloud.d
4
5 xx = p_cloud.x;
6 yy = p_cloud.y;
7 zz = p_cloud.z;
8
9 % expand borders for filtering operation
10 [xx,rows,cols] = expand_borders(xx,3);
11 [yy,~,~] = expand_borders(yy,3);
12 [zz,~,~] = expand_borders(zz,3);
13
14 % compute change for all layers
15 xx_filt = calc_change_in_region(xx,3);
16 yy_filt = calc_change_in_region(yy,3);
17 zz_filt = calc_change_in_region(zz,3);
18
19 % calculate density
20 p_cloud.d = sqrt(xx_filt.^2+yy_filt.^2+zz_filt.^2);
21
22 p_cloud.d = p_cloud.d(rows,cols);

```

### A.3.3 CONNECTED COMPONENT LABELING (CCL)

COMPLEMENTARY INFORMATION

Code A.6: Labeling connected components by flood filling.

```

1 function CC = get_cc_by_flood_fill(BW, conn)
2 %input: binary matrix BW and connectivity conn 4 or 8
3 %output: struct CC of indices for connected components
4
5 [M,~] = size(BW);
6
7 % linear connectivity neighbour offsets (n_off)
8 if isequal(conn,4)
9     n_off = [-1;M;1;-M]; %direct connectivity
10 elseif isequal(conn,8)
11     n_off = [-1;M;1;-M;M+1;M-1;-M+1;-M-1]; %full connectivity
12 end
13
14 % start with nearest foreground pixel and label it
15 idx = find(BW,1); idx_region = idx; label_val = 1;
16
17 while ~isempty(idx)
18     % set the just labeled pixels to 0
19     BW(idx) = 0;
20
21     % find all the nonzero neighbours
22     neighb_idx = bsxfun(@plus,idx,n_off');
23     neighb_idx = unique(neighb_idx(:));
24
25     idx = neighb_idx(BW(neighb_idx));
26     idx_region = [idx_region;idx];
27
28     % if empty --> region labeled and find next foreground pixel
29     if isempty(idx)
30         CC.PixelIdxList{label_val} = idx_region;
31         idx = find(BW,1); idx_region = idx;
32         label_val = label_val+1;
33     end
34 end

```

Code A.7: Run length encoding for connected components labeling.

```

1 function [runs, labels] = rlenc(col_vect)
2 %input: column of an binary matrix BW
3 %output: runs with start and end pixel, corresp. label
4
5 p_idx = find(col_vect);
6
7 if isempty(p_idx)
8     runs = []; labels = []; return;
9 end
10
11 num_p = numel(p_idx); label = 1;
12 if num_p == 1

```

```

13     runs(1,:) = [p_idx(1),p_idx(1)];
14 else
15     runs(1,:) = [p_idx(1),[]];
16 end
17
18 for i = 1:num_p-1
19     % end of run
20     if p_idx(i+1) > p_idx(i)+1
21         % end current run
22         runs(label,2) = p_idx(i);
23         % start new run
24         label = label+1;    runs(label,1) = p_idx(i+1);
25     end
26     % end of last run
27     if i == num_p-1
28         runs(label,2) = p_idx(end);
29     end
30 end
31
32 labels = (1:label)';

```

#### A.3.4 STOCHASTIC MODEL

Code A.8: Computes precision layers for a point cloud structure.

```

1 function p_cloud = compute_point_precision(p_cloud, filter)
2 %input 1: struct p_cloud with information .x,.y,.z,.i
3 %output: struct p_cloud with add. information .sx,.sy,.sz
4
5 switch filter
6     case 0
7         params = load('params_1MHZ.mat');
8     case 1
9         params = load('params_12MHZ.mat');
10    case 2
11        params = load('params_18MHZ_new.mat');
12 end
13
14 params = params.power_func_params;
15
16 % angular accuracy
17 angular_acc = 0.007;
18
19 % conversions
20 RHO = 180/pi;
21
22 x = double(p_cloud.x);    y = double(p_cloud.y);
23 z = double(p_cloud.z);    int = double(p_cloud.int);
24

```

## COMPLEMENTARY INFORMATION

```
25 % polar coordinates from cartesian
26 d = sqrt(x.^2+y.^2+z.^2);
27 az = atan_4q(y,x)*RH0;
28 vz = acos(z./d)*RH0;
29
30 % compute dist accuracies
31 dist_acc = params(1,1)*power(int,params(2,1));
32
33 % Error propagation for a Z+F Imager 5006
34 az = az/RH0;
35 vz = vz/RH0;
36 angular_acc = angular_acc/RH0;
37
38 dx_d_d = cos(az).*sin(vz);
39 dx_d_az = -d.*sin(az).*sin(vz);
40 dx_d_vz = d.*cos(az).*cos(vz);
41
42 s_x = sqrt((dx_d_d.^2).*(dist_acc.^2)+(dx_d_az.^2).*...
43 (angular_acc.^2)+(dx_d_vz.^2).*(angular_acc.^2));
44
45 dy_d_d = sin(az).*sin(vz);
46 dy_d_az = d.*cos(az).*sin(vz);
47 dy_d_vz = d.*sin(az).*cos(vz);
48
49 s_y = sqrt((dy_d_d.^2).*(dist_acc.^2)+(dy_d_az.^2).*...
50 (angular_acc.^2)+(dy_d_vz.^2).*(angular_acc.^2));
51
52 dz_d_d = cos(vz);
53 dz_d_vz = -d.*sin(vz);
54
55 s_z = sqrt((dz_d_d.^2).*(dist_acc.^2)+(dz_d_vz.^2).*...
56 (angular_acc.^2));
57
58 p_cloud.s_x = s_x;
59 p_cloud.s_y = s_y;
60 p_cloud.s_z = s_z;
```

### A.3.5 SEGMENTATION

Code A.9: Decision of splitting a region.

```
1 function split_flag = predicate_split(region)
2
3 % get parameters of the region
4 w = region.bbox(1,3);
5 h = region.bbox(1,4);
6
7 % stop split, if points in region are less than 100 or w or h < 5
8 if numel(region.point_indices) < 100 || w < 5 || h < 5
```

```

9     split_flag = 0;
10  else
11     % test if region is already plane region
12     flag = predicate_is_plane(region);
13
14     content_ratio = numel(region.point_indices)/(w*h);
15
16     if w > h
17         bbox_size_ratio = h/w;
18     else
19         bbox_size_ratio = w/h;
20     end
21
22     content_ratio_thresh = 0.5;
23     size_ratio_thresh = 0.1;
24
25     % compute flag
26     split_flag = ~flag || content_ratio < content_ratio_thresh || ...
27     bbox_size_ratio < size_ratio_thresh;
28  end

```

Code A.10: Decision of merging two regions.

```

1  function merge_flag = predicate_merge(region_pairs)
2
3  % threshold parameters for merging two planes
4  angle_thresh = 0.02; % [radian] equivalent to approx 1.0 degree
5
6  % get parameters of the region pairs
7  w_1 = region_pairs(1).bbox(1,3);
8  h_1 = region_pairs(1).bbox(1,4);
9
10 w_2 = region_pairs(2).bbox(1,3);
11 h_2 = region_pairs(2).bbox(1,4);
12
13 % check if planes are adjacent
14 c_size = cloud_size;
15 up = region_pairs(2).point_indices-1;
16 up_r = region_pairs(2).point_indices-1+c_size(1,1);
17 up_l = region_pairs(2).point_indices-1-c_size(1,1);
18 down = region_pairs(2).point_indices+1;
19 down_r = region_pairs(2).point_indices+1+c_size(1,1);
20 down_l = region_pairs(2).point_indices+1-c_size(1,1);
21 point_indices_2 = unique([up;up_r;up_l;down;down_r;down_l]);
22 num_conn_elements = nnz(ismember(region_pairs(1).point_indices,...
23 point_indices_2));
24
25 % calculate flag
26 min_conn = min([w_1;h_1;w_2;h_2]);
27 if num_conn_elements < min_conn/4
28     merge_flag = 0;

```

COMPLEMENTARY INFORMATION

```

29     return;
30 end
31
32 % check if planes have similar normal direction
33 theta = acos(min(1,max(-1,region_pairs(1).plane_params(1:3,1).'*...
34 region_pairs(2).plane_params(1:3,1)/norm(region_pairs(1). ...
35 plane_params(1:3,1))/norm(region_pairs(2).plane_params(1:3,1))));
36
37 merge_flag = theta < angle_thresh;

```

Code A.11: Splitting a region.

```

1 function subregions = split_region(region, func)
2
3 flag = feval(func,region);
4 if flag
5
6     w = region.bbox(1,3);
7     h = region.bbox(1,4);
8
9     % if content ratio is not high enough
10    content_ratio_thresh = 0.6;
11    content_ratio = numel(region.point_indices)/(w*h);
12
13    if content_ratio < content_ratio_thresh
14
15        % check for optimal division
16        subreg_d_w = divide_region(region,cloud_size,'w');
17        subreg_d_h = divide_region(region,cloud_size,'h');
18
19        content_ratio_d_w1 = numel(subreg_d_w(1).point_indices)/...
20        (subreg_d_w(1).bbox(1,3)*subreg_d_w(1).bbox(1,4));
21        content_ratio_d_w2 = numel(subreg_d_w(2).point_indices)/...
22        (subreg_d_w(2).bbox(1,3)*subreg_d_w(2).bbox(1,4));
23        content_ratio_d_h1 = numel(subreg_d_h(1).point_indices)/...
24        (subreg_d_h(1).bbox(1,3)*subreg_d_h(1).bbox(1,4));
25        content_ratio_d_h2 = numel(subreg_d_h(2).point_indices)/...
26        (subreg_d_h(2).bbox(1,3)*subreg_d_h(2).bbox(1,4));
27        content_ratio_d_w = content_ratio_d_w1+content_ratio_d_w2;
28        content_ratio_d_h = content_ratio_d_h1+content_ratio_d_h2;
29
30        if content_ratio_d_w >= content_ratio_d_h
31            subregions = subreg_d_w;
32        else
33            subregions = subreg_d_h;
34        end
35    else
36        if w > h || w == h
37            subregions = divide_region(region,cloud_size,'w');
38        else
39            subregions = divide_region(region,cloud_size,'h');

```

```

40     end
41 end
42
43 % compute and add plane parameters
44 flag_subregion = true(2,1);
45 for i = 1:2
46     if numel(subregions(i).point_indices) > 2
47         p_3D = [p_cloud.x(subregions(i).point_indices),...
48                p_cloud.y(subregions(i).point_indices),...
49                p_cloud.z(subregions(i).point_indices)];
50         p_3D_p = [p_cloud.s_x(subregions(i).point_indices),...
51                 p_cloud.s_y(subregions(i).point_indices),...
52                 p_cloud.s_z(subregions(i).point_indices)];
53
54         [plane_approx_region, plane_dists] = fitplane (p_3D);
55
56         subregions(i).plane_params = plane_approx_region;
57         subregions(i).av_point_acc = mean(sqrt(p_3D_p(:,1).^2 ...
58                +p_3D_p(:,2).^2+p_3D_p(:,3).^2));
59         subregions(i).max_plane_dist = max(plane_dists);
60
61         % add cog and region id
62         subregions(i).cog = mean(p_3D)'; %real 3D cog
63         subregions(i).region_id = region.region_id;
64         flag_subregion(i,1) = true;
65     else
66         flag_subregion(i,1) = false;
67     end
68 end
69 % delete too small subregions
70 subregions = subregions(flag_subregion);
71 else
72     subregions = [];
73 end

```

Code A.12: Merging regions.

```

1 function [plane_regions, subregions] = merge_regions(regions, func)
2
3 % select regions with special properties
4 n_regions = numel(regions);
5
6 max_plane_dist = zeros(n_regions,1);
7 av_point_acc = zeros(n_regions,1);
8
9 for i = 1:n_regions
10     max_plane_dist(i,1) = regions(i).max_plane_dist;
11     av_point_acc(i,1) = regions(i).av_point_acc;
12 end
13
14 % get mergeable regions

```

## COMPLEMENTARY INFORMATION

```

15 ind_merge_planes = max_plane_dist < av_point_acc;
16 mergeable_regions = regions(ind_merge_planes);
17
18 if ~isempty(mergeable_regions)
19     ind_p_r = 0;
20     merged_pairs = [];
21
22     % create region pairs
23     n_mergeable_regions = numel(mergeable_regions);
24     pair_combs = combnk(1:n_mergeable_regions,2);
25
26     while ~isempty(pair_combs)
27         num_pair_combs = numel(pair_combs)/2;
28
29         for i = 1:num_pair_combs
30             region_pairs(1) = mergeable_regions(pair_combs(i,1));
31             region_pairs(2) = mergeable_regions(pair_combs(i,2));
32
33             % test if regions are connected having similar orientation
34             flag = feval(func,region_pairs);
35             if flag
36                 merged_region_tmp.bbox = [];
37
38                 % combine two regions to one region and test it
39                 merged_region_tmp.point_indices = [region_pairs(1)...
40                 .point_indices;region_pairs(2).point_indices];
41
42                 % update bounding box
43                 bbox = get_bbox(cloud_size,...
44                 merged_region_tmp(i).point_indices);
45                 merged_region_tmp(i).bbox = bbox;
46
47                 p_3D = [p_cloud.x(merged_region_tmp.point_indices)...
48                 ,p_cloud.y(merged_region_tmp.point_indices)...
49                 ,p_cloud.z(merged_region_tmp.point_indices)];
50                 p_3D_p = [p_cloud.s_x(merged_region_tmp.point_indices)...
51                 ,p_cloud.s_y(merged_region_tmp.point_indices)...
52                 ,p_cloud.s_z(merged_region_tmp.point_indices)];
53
54                 [plane_approx_region, plane_dists] = fitplane(p_3D);
55
56                 merged_region_tmp.plane_params = plane_approx_region;
57
58                 merged_region_tmp.av_point_acc = mean(sqrt(...
59                 p_3D_p(:,1).^2+p_3D_p(:,2).^2+p_3D_p(:,3).^2));
60                 merged_region_tmp.max_plane_dist = max(plane_dists);
61
62                 % add cog and region id
63                 merged_region_tmp.cog = mean(p_3D)'; %real 3D cog
64                 merged_region_tmp.region_id = region.region_id;
65
66                 % test if merged regions are part of a unique plane
67                 flag_p = predicate_split(merged_region_tmp);
68

```

```

69         if ~flag_p
70             plane_regions(ind_p_r+1) = merged_region_tmp;
71             merged_pairs(ind_p_r+1,:) = pair_combs(i,:);
72             ind_p_r = ind_p_r+1;
73
74             % update pair combs
75             select = ~(pair_combs(:,1) == pair_combs(i,1))...
76                 & ~(pair_combs(:,1) == pair_combs(i,2))...
77                 & ~(pair_combs(:,2) == pair_combs(i,1))...
78                 & ~(pair_combs(:,2) == pair_combs(i,2));
79             pair_combs = pair_combs(select,:);
80             break;
81         else
82             pair_combs(i,:) = [];
83             break;
84         end
85     else
86         pair_combs(i,:) = [];
87         break;
88     end
89 end
90 end
91
92 if isempty(merged_pairs)
93     plane_regions = [];
94     subregions = regions;
95 else
96     i_not_merged = setxor((1:n_mergeable_regions)',merged_pairs(:));
97     subregions_not_merged = mergeable_regions(i_not_merged);
98     not_mergeable_regions = regions(~ind_merge_planes);
99     subregions = subregions_not_merged;
100    n_sub_n_merged = numel(subregions_not_merged);
101
102    for i = 1:numel(not_mergeable_regions)
103        subregions(n_sub_n_merged+i) = not_mergeable_regions(i);
104    end
105 end
106 else
107     plane_regions = [];
108     subregions = regions;
109 end

```

Code A.13: Split and merge algorithm.

```

1 function plane_regions = split_merge(region, p_cloud, cloud_size)
2 %input: structs p_cloud. ..., region. ..., size cloud_size
3 %output: struct plane_regions. ...
4
5 % try to split main region first
6 subregions = split_region(region,@predicate_split);
7

```

## COMPLEMENTARY INFORMATION

```
8 if isempty(subregions)
9     flag_is_plane = predicate_is_plane(region);
10    if flag_is_plane
11        plane_regions(1) = region;
12    else
13        plane_regions = [];
14    end
15    return;
16 end
17
18 % split merge procedure
19 n_plane_regions = 0;
20 while ~isempty(subregions) ...
21
22     if ~exist('plane_regions','var')
23         plane_regions = [];
24     end
25
26 % functions for splitting and merging
27 function subregions = split_region(region,func) ...
28
29 function [plane_regions, subregions] = merge_regions(regions,func) ...
30
31 % definition of a planar region
32 function plane_flag = predicate_is_plane(region) ...
33
34 % decision functions for splitting or merging
35 function merge_flag = predicate_merge(region_pairs) ...
36
37 function split_flag = predicate_split(region) ...
```

### A.3.6 MATCHING

Code A.14: Extending Code for direct plane matching if same correspondences are present among different scan pairs.

```
1 %% direct matching
2
3 % find pairs of matching pairs for same standpoint indices of target
4 target_ids = unique(m_pairs(:,1));
5 for i = 1:numel(target_ids)
6     id_s_target{i} = find(m_pairs(:,1)==target_ids(i,1));
7     if numel(id_s_target{i}) > 1
8         combs = combnk(id_s_target{i},2);
9         n_combs = numel(combs(:,1));
10        for j = 1:n_combs
11            p_pair = [m_pairs(combs(j,1),2),m_pairs(combs(j,2),2)];
12            p_pairs = [p_pair;p_pair(1,2),p_pair(1,1)];
13            [is_comb,id_pairs] = ismember(p_pairs,m_pairs,'rows');
```

```

14         if any(is_comb)
15             d_match = [];
16             n_match = numel(p_matches{1,combs(j,1)}(:,1));
17             for k = 1:n_match
18                 m_ids{k} = find(p_matches{1,combs(j,2)}(:,1)...
19                     == p_matches{1,combs(j,1)}(k,1));
20                 if ~isempty(m_ids{k})
21                     for l = 1:numel(m_ids{k})
22                         d_match_t = [p_matches{1,combs(j,1)}(k,2)...
23                             ,p_matches{1,combs(j,2)}(m_ids{k}(l),2)];
24                         d_match = [d_match;d_match_t];
25                     end
26                 end
27             end
28             p_matches{1,id_pairs(is_comb)} = ...
29             [p_matches{1,id_pairs(is_comb)};d_match];
30             [tmp,~,~] = unique(p_matches{1,3},'rows');
31             p_matches{1,id_pairs(is_comb)} = tmp;
32         end
33     end
34 end
35 end

```

## A.3.7 REGISTRATION

Code A.15: Transform planes to another coordinate system.

```

1 function planes_t = transform_planes(planes, R, T)
2 %input: - planes(1:4,n) with n planes
3 %       where [nx;ny;nz;d] = planes(:,i)
4 %output: - planes_t transformed planes
5
6 planes_t = R*planes(1:3,:);
7 planes_t(4,:) = planes(4,:)+T'*R*planes(1:3,:);

```

Code A.16: Fitting a plane using SVD.

```

1 function [plane_params, plane_dists] = fitplane(XYZ)
2
3 [npts,rows] = size(XYZ);
4
5 % center of gravities
6 X_cog = sum(XYZ(:,1))/npts;
7 Y_cog = sum(XYZ(:,2))/npts;
8 Z_cog = sum(XYZ(:,3))/npts;
9

```

## COMPLEMENTARY INFORMATION

```
10 % reduced coordinates
11 X_red = XYZ(:,1)-X_cog;
12 Y_red = XYZ(:,2)-Y_cog;
13 Z_red = XYZ(:,3)-Z_cog;
14
15 % augmented matrix
16 A = [X_red Y_red Z_red];
17
18 N = A'*A;
19
20 %perform svd and get normal vector
21 [U,S,V] = svd(N);
22
23 n_v = V(:,end);
24
25 % Flip the normal vector if it is not pointing towards the sensor
26 p1 = - [X_cog,Y_cog,Z_cog];
27 p2 = n_v';
28
29 angle = atan2(norm(cross(p1,p2)),p1*p2');
30 if angle > pi/2 || angle < -pi/2
31     n_v = -n_v;
32 end
33
34 d = (sum(n_v(1,1).*XYZ(:,1))+sum(n_v(2,1).*XYZ(:,2))...
35 +sum(n_v(3,1).*XYZ(:,3)))/npts;
36
37 plane_params = [n_v;d];
38
39 plane_dists = abs(n_v(1,1).*XYZ(:,1)+n_v(2,1).*XYZ(:,2)...
40 +n_v(3,1).*XYZ(:,3)-d);
```

# References

- [1] T. ABMAYR, F. HÄRTL, G. HIRZINGER, D. BURSCHKA, and C. FRÖHLICH: “A correlation based target finder for terrestrial laser scanning”. In: *Journal of Applied Geodesy* 2.3 (2008), pp. 131–137.
- [2] F. ACKERMANN: “Digital image correlation: performance and potential application in photogrammetry”. In: *The Photogrammetric Record* 11.64 (1984), pp. 429–439.
- [3] D. AKCA: “Matching of 3D surfaces and their intensities”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 62.2 (2007), pp. 112–121.
- [4] W. BAARDA: “A testing Procedure for use in geodetic networks”. In: *Publications on Geodesy, New Series* 2.5 (1968).
- [5] K.-H. BAE and D. LICHTI: “On-site self-calibration using planar features for terrestrial laser scanners”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 36.3 (2007), pp. 14–19.
- [6] S. BARNEA and S. FILIN: “Segmentation of terrestrial laser scanning data using geometry and image information”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 76 (2013), pp. 33–48.
- [7] E. A. BENDER and S. G. WILLIAMSON: *Lists, Decisions and Graphs*. S. Gill Williamson, 2010.
- [8] P. J. BESL and N. D. MCKAY: “A method for registration of 3-D shapes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (Feb. 1992), pp. 239–256.
- [9] R. BLASKOW and D. SCHNEIDER: “Analysis and correction of the dependency between laser scanner intensity values and range”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 40.5 (2014), p. 107.
- [10] R. BÖCK: “Allgemeinste Formulierung der Ausgleichsrechnung nach der Methode der kleinsten Quadratsummen”. In: *Zeitschrift für Geodäsie, Geoinformation und Landmanagement (ZfV)* 86 (1961), pp. 98–106.
- [11] V BOEDER, T. KERSTEN, C HESSE, T THIES, and A SAUER: “Initial experience with the integration of a terrestrial laser scanner into the mobile hydrographic multi sensor system on a ship”. In: *ISPRS Istanbul Workshop 2010 on Modeling of optical airborne and spaceborne Sensors*. 2010.
- [12] J BOEHM and S BECKER: “Automatic Marker-Free Registration of Terrestrial Laser Scans using Reflectance Features”. In: *Proceedings of 8th Conference on Optical 3D Measurement Techniques*. Zürich, Schweiz, 2007, pp. 338–344.

## REFERENCES

- [13] W. BÖHLER and A. MARBS: “3D scanning and photogrammetry for heritage recording: a comparison”. In: *Proceedings of the 12th International Conference on Geoinformatics*. Gavle University Press, Sweden, 2004, pp. 291–298.
- [14] W. BÖHLER, M. B. VICENT, A. MARBS, et al.: “Investigating laser scanner accuracy”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 34.5 (2003), pp. 696–701.
- [15] M. BRELL, K. SEGL, L. GUANTER, and B. BOOKHAGEN: “Hyperspectral and lidar intensity data fusion: A framework for the rigorous correction of illumination, anisotropic effects, and cross calibration”. In: *IEEE Transactions on Geoscience and Remote Sensing* 55.5 (2017), pp. 2799–2810.
- [16] F. K. BRUNNER: “On the methodology of Engineering Geodesy”. In: *Journal of Applied Geodesy* 1 (2007), pp. 57–62.
- [17] M. BURGER, D. WUJANZ, and F. NEITZEL: “Segmentierung von Punktwolken anhand von geometrischen und radiometrischen Informationen”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2017, pp. 116–128.
- [18] M. BURGER, D. WUJANZ, and F. NEITZEL: “Synergetische Segmentierung und Registrierung von Punktwolken”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2018, pp. 130–144.
- [19] G. CARDANO: *De subtilitate*. 1560.
- [20] W. CASPARY: *Auswertung von Messdaten: Statistische Methoden für Geo- und Ingenieurwissenschaften*. Oldenbourg Wissenschaftsverlag, 2007.
- [21] T. F. CHAN, G. H. GOLUB, and R. J. LEVEQUE: “Algorithms for computing the sample variance: Analysis and recommendations”. In: *The American Statistician* 37.3 (1983), pp. 242–247.
- [22] T. H. CORMEN, C. STEIN, R. L. RIVEST, and C. LEISERSON: “Representations of graphs”. In: *Introduction to Algorithms* (2001), pp. 527–531.
- [23] M. CRAMER, D. STALLMANN, and N. HAALA: “Direct georeferencing using GPS/inertial exterior orientations for photogrammetric applications”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 33.3 (2000), pp. 198–205.
- [24] J. S. DAI: “Euler-Rodrigues formula variations, quaternion conjugation and intrinsic connections”. In: *Mechanism and Machine Theory* 92 (2015), pp. 144–152.
- [25] F. DEUMLICH and R. STAIGER: *Instrumentenkunde der Vermessungstechnik*. 9th ed. Herbert Wichmann Publishing, Heidelberg, 2002.
- [26] DEUTSCHES INSTITUT FÜR NORMUNG E.V.: *Quality management systems - Fundamentals and vocabulary*. en. Standard DIN EN ISO 9000:2015. International Organization for Standardization, 2015.

## REFERENCES

- [27] J. DIEBEL: “Representing attitude: Euler angles, unit quaternions, and rotation vectors”. In: *Matrix* 58.15-16 (2006), pp. 1–35.
- [28] C. DOLD: Ebenenbasierte Verfahren für die automatische Registrierung terrestrischer Laserscans. Series C, Nr. 646. Committee for Geodesy (DGK), Munich, 2010.
- [29] I. ELKHRACHY and W. NIEMEIER: “Stochastic assessment of terrestrial laser scanner”. In: *Proceedings of the ASPRS Annual Conference, Reno, United States of America*. 2006.
- [30] V. ERÍĆ and M. GÖRING: “Intensity of the terrestrial laser scanning data: incident angle and surface reflectance effects”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2017.
- [31] L. EULER: “Formulae generales pro translatione quacunque corporum rigidorum”. In: *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 20 (1775), pp. 189–207.
- [32] X. FANG: “Weighted total least-squares with constraints: a universal formula for geodetic symmetrical transformations”. In: *Journal of geodesy* 89.5 (2015), pp. 459–469.
- [33] C. FORNELL and D. F. LARCKER: “Evaluating Structural Equation Models with Unobservable Variables and Measurement Error”. In: *Journal of Marketing Research* 18.1 (1981), pp. 39–50.
- [34] W. FÖRSTNER: “Das Programm TRINA zur Ausgleichung und Gütebeurteilung geodätischer Lagenetze”. In: *Zeitschrift für Geodäsie, Geoinformation und Landmanagement (ZfV)* 104.2 (1979), pp. 61–72.
- [35] C. D. GHILANI: Adjustment computations: spatial data analysis. John Wiley & Sons, 2010.
- [36] F. GIELSDORF: “Ebenendetektion, Matching und verkettete Transformation von Laserscans”. In: *Allgemeine Vermessungs-Nachrichten (AVN)* 11-12 (2009), pp. 391–395.
- [37] G. H. GOLUB and C. F. VAN LOAN: Matrix computations. Vol. 3. Johns Hopkins University Press, 2013.
- [38] D. GRANT, J. BETHEL, and M. CRAWFORD: “Point-to-plane registration of terrestrial laser scans”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 72 (2012), pp. 16–26.
- [39] A. GRUEN and D. AKCA: “Least squares 3D surface and curve matching”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 59.3 (2005), pp. 151–174.
- [40] L. GRÜNDIG: Grundlagen der Ausgleichungsrechnung. lecture notes. 2003.
- [41] J. GÜHRING: “3D-Erfassung und Objektrekonstruktion mittels Streifenprojektion”. PhD thesis. 2002.
- [42] S. W. R. HAMILTON: “LXXVIII. On quaternions; or on a new system of imaginaries in Algebra”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and*

## REFERENCES

- Journal of Science* 25.169 (1844), pp. 489–495. eprint: <https://doi.org/10.1080/14786444408645047>.
- [43] R. M. HARALICK and L. G. SHAPIRO: Computer and robot vision. Vol. 1. Addison-Wesley, 1992.
- [44] E. HEINZ, M. METTENLEITER, H. KUHLMANN, and C. HOLST: “Strategy for determining the stochastic distance characteristics of the 2D Laser Scanner Z+ F Profiler 9012A with special focus on the close range”. In: *Sensors* 18.7 (2018), p. 2253.
- [45] H. HEISTER: “Zur standardisierten Überprüfung von terrestrischen Laserscannern (TLS)”. In: *Schriftenreihe des DVW* 51 (2006), pp. 35–44.
- [46] F. R. HELMERT: Die Ausgleichsrechnung nach der Methode der kleinsten Quadrate: mit Anwendungen auf die Geodäsie und die Theorie der Messinstrumente. 3rd ed. BG Teubner, 1924.
- [47] C HESSE: Hochauflösende kinematische Objekterfassung mit terrestrischen Laserscannern. Series C, Nr. 608. Committee for Geodesy (DGK), Munich, 2007.
- [48] B. HÖFLE and N. PFEIFER: “Correction of laser scanning intensity data: Data and model-driven approaches”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 62.6 (2007), pp. 415–433.
- [49] S. L. HOROWITZ and T. PAVLIDIS: “Picture segmentation by a directed split-and-merge procedure”. In: vol. International Joint Conference on Pattern Recognition. 2. 1974, pp. 424–433.
- [50] R. JÄGER, T. MÜLLER, H. SALER, and R. SCHWÄBLE: Klassische und robuste Ausgleichungsverfahren: ein Leitfaden für Ausbildung und Praxis von Geodäten und Geoinformatikern. Herbert Wichmann Publishing, Heidelberg, 2005.
- [51] A. V. JELALIAN: Laser radar systems. Artech House, 1992.
- [52] X. JIANG and H. BUNKE: “Gewinnung und Analyse von Tiefenbildern”. In: *Dreidimensionales Computersehen*. Springer, 1997. Chap. Segmentierung.
- [53] R. JOECKEL, M. STOBER, and W. HUEP: Elektronische Entfernungsmessung und ihre Integration in aktuelle Positionierungsverfahren. Herbert Wichmann Publishing, Heidelberg, 2008.
- [54] S. JUNTILA, M. VASTARANTA, X. LIANG, H. KAARTINEN, A. KUKKO, S. KAASALAINEN, M. HOLOPAINEN, H. HYYPPÄ, and J. HYYPPÄ: “Measuring leaf water content with dual-wavelength intensity data from terrestrial laser scanners”. In: *Remote Sensing* 9.1 (2017), p. 8.
- [55] S. KAUKER and V. SCHWIEGER: “A synthetic covariance matrix for monitoring by terrestrial laser scanning”. In: *Journal of Applied Geodesy* 11.2 (2017), pp. 77–87.
- [56] G. KERESKES and V. SCHWIEGER: “Elementary error model applied to terrestrial laser scanning measurements: study case arch dam Kops”. In: *Mathematics* 8.4 (2020), p. 593.

- [57] G. KERMARREC: “On Estimating the Hurst Parameter from Least-Squares Residuals. Case Study: Correlated Terrestrial Laser Scanner Range Noise”. In: *Mathematics* 8.5 (2020), p. 674.
- [58] G. KERMARREC, H. ALKHATIB, and I. NEUMANN: “On the sensitivity of the parameters of the intensity-based stochastic model for terrestrial laser scanner. Case study: B-spline approximation”. In: *Sensors* 18.9 (2018), p. 2964.
- [59] G. KERMARREC, M. LÖSLER, and J. HARTMANN: “Analysis of the temporal correlations of TLS range observations from plane fitting residuals”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 171 (2021), pp. 119–132.
- [60] G. KERMARREC, I. NEUMANN, H. ALKHATIB, and S. SCHÖN: “The stochastic model for Global Navigation Satellite Systems and terrestrial laser scanning observations: A proposal to account for correlations in least squares adjustment”. In: *Journal of Applied Geodesy* 13.2 (2019), pp. 93–104.
- [61] K. KRAUS: “Geometrische Informationen aus Photographien und Laserscanneraufnahmen”. In: *Photogrammetrie*. 7th ed. Vol. 1. Walter de Gruyter, Berlin, New York, 2004.
- [62] K. KRAUS: “Grundlagen und Standardverfahren”. In: *Photogrammetrie*. 5th ed. Vol. 1. Ferdinand Dümmler, Bonn, 1994.
- [63] D. KRUEGER, M. BURGER, D. WUJANZ, and N. HEUWOLD: “3D-Datenverarbeitung mit multispektralen Laserscandaten”. In: *3D-NordOst 2017. 20. Anwendungsbezogener Workshop zur Erfassung, Modellierung, Verarbeitung und Auswertung von 3D-Daten*. 2017, pp. 53–60.
- [64] H. KUHLMANN, V. SCHWIEGER, A. WIESER, and W. NIEMEIER: “Engineering Geodesy—Definition and Core Competencies”. In: *Journal of Applied Geodesy* 8.4 (2014), pp. 327–334.
- [65] H. KUHLMANN, V. SCHWIEGER, A. WIESER, and W. NIEMEIER: “Ingenieurgeodäsie—Definition, Kernkompetenzen und Alleinstellungsmerkmale”. In: *Zeitschrift für Geodäsie, Geoinformation und Landmanagement (ZfV)* 138.6 (2013), pp. 391–399.
- [66] J. L. LAGRANGE: *Mécanique Analytique*. Paris: Desaint, 1788.
- [67] L. LENZMANN and E. LENZMANN: “Strenge Auswertung des nichtlinearen Gauß-Helmert-Modells”. In: *Allgemeine Vermessungs-Nachrichten (AVN)* III.2004 (2004), pp. 68–72.
- [68] Z. LI, D. L. JUPP, A. H. STRAHLER, C. B. SCHAAF, G. HOWE, K. HEWAWASAM, E. S. DOUGLAS, S. CHAKRABARTI, T. A. COOK, I. PAYNTER, et al.: “Radiometric calibration of a dual-wavelength, full-waveform terrestrial lidar”. In: *Sensors* 16.3 (2016), p. 313.
- [69] M. LICHTENSTEIN and W. BENNING: “Registrierung von Punktwolken auf der Grundlage von Objektprimitiven”. In: *Allgemeine Vermessungs-Nachrichten (AVN)* 6 (2010), pp. 202–207.

## REFERENCES

- [70] D. D. LICHTI: “Error modelling, calibration and analysis of an AM–CW terrestrial laser scanner system”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 61.5 (2007), pp. 307–324.
- [71] D. D. LICHTI: “Spectral filtering and classification of terrestrial laser scanner point clouds”. In: *The Photogrammetric Record* 20.III (2005), pp. 218–240.
- [72] R. LINDENBERGH and P. PIETRZYK: “Change detection and deformation analysis using static and mobile laser scanning”. In: *Applied Geomatics* 7.2 (2015), pp. 65–74.
- [73] T. LUHMANN: *Nahbereichsphotogrammetrie: Grundlagen-Methoden-Beispiele*. Ed. by VDE. 4th ed. Herbert Wichmann Publishing, Heidelberg, 2018.
- [74] H. MAHMOUDABADI, M. J. OLSEN, and S. TODOROVIC: “Efficient terrestrial laser scan segmentation exploiting data structure”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 119 (2016), pp. 135–150.
- [75] L. MATIKAINEN, K. KARILA, J. HYYPPÄ, P. LITKEY, E. PUTTONEN, and E. AHOKAS: “Object-based analysis of multispectral airborne laser scanner data for land cover classification and map updating”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 128 (2017), pp. 298–313.
- [76] K. MECHELKE, M. LINDSTAEDT, H. STERNBERG, and T. KERSTEN: “Bauwerksmonitoring mit terrestrischem Laserscanning - eine Machbarkeitsstudie”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2012, pp. 55–62.
- [77] M. METTENLEITER, F. HÄRTL, S. KRESSER, and C. FRÖHLICH: *Laserscanning-Phasenbasierte Lasermesstechnik für die hochpräzise und schnelle dreidimensionale Umgebungserfassung*. Ed. by M. INDUSTRIE. Vol. 371. Die Bibliothek der Technik, 2015.
- [78] M. MEZIAN, B. VALLET, B. SOHEILIAN, and N. PAPANODITIS: “Uncertainty propagation for terrestrial mobile laser scanner”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 41 (2016), pp. 331–335.
- [79] E. M. MIKHAIL and F. E. ACKERMANN: *Observations and least squares*. IEP-A Dun-Donneley Publisher, 1976.
- [80] G. MÜLLER: *Handbuch Ingenieurgeodäsie: Eisenbahnbau*. Ed. by M. MÖSER, G. MÜLLER, H. SCHLEMMER, and H. WERNER. Herbert Wichmann Publishing, Heidelberg, 2000.
- [81] F. NEITZEL: “Ausgleichsrechnung–Modellbildung, Auswertung, Qualitätsbeurteilung”. In: *Qualitätsmanagement geodätischer Mess- und Auswerteverfahren, Beiträge zum 93. DVW-Seminar* 93 (2010), pp. 95–127.
- [82] F. NEITZEL: “Investigation of axes errors of terrestrial laser scanners”. In: *5th International Symposium Turkish-German Joint Geodetic Days, Berlin*. 2006.
- [83] F. NEITZEL and L. JOHANNES: “Anwendung des tachymetrischen Laserscannings in der Überwachung eines historischen Bauwerkes”. In: *Photogrammetrie - Laser-*

- scanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2009, pp. 127–136.
- [84] F. NEITZEL and S. PETROVIC: “Total least squares (TLS) im Kontext der Ausgleichung nach kleinsten Quadraten am Beispiel der ausgleichenden Geraden”. In: *Zeitschrift für Geodäsie, Geoinformation und Landmanagement (ZfV)* 133 (2008), pp. 141–148.
- [85] H. NEUNER: “Qualitätsbetrachtungen zu TLS-Daten”. In: *Qualitätssicherung geodätischer Mess- und Auswerteverfahren, Beiträge zum 180. DVW-Seminar* 95 (2019), pp. 69–89.
- [86] W. NIEMEIER: *Ausgleichsrechnung: Statistische Auswertemethoden*. Walter de Gruyter, 2008.
- [87] N. OTSU: “A threshold selection method from gray-level histograms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66.
- [88] J.-A. PAFFENHOLZ, H. ALKHATIB, and H. KUTTERER: “Direct geo-referencing of a static terrestrial laser scanner”. In: *Journal of Applied Geodesy* 4.3 (2010), pp. 115–126.
- [89] A. PESCI, G. TEZA, and E. BONALI: “Terrestrial laser scanner resolution: numerical simulations and experiments on spatial sampling optimization”. In: *Remote Sensing* 3.1 (2011), pp. 167–184.
- [90] S. PETROVIC: *Parameterschätzung für unvollständige funktionale Modelle in der Geodäsie*. Series C, Nr. 563. Committee for Geodesy (DGK), Munich, 2003.
- [91] A. J. POPE: “Some pitfalls to be avoided in the iterative adjustment of nonlinear problems”. In: *Photogrammetric Engineering*. Vol. 38. 6. Annual Meeting of the American Society of Photogrammetry. 1972, p. 597.
- [92] M. PREVITALI, L. BARAZZETTI, R. BRUMANA, and M. SCAIONI: “Laserscan registration using planar features”. In: 40.5 (2014), pp. 501–508.
- [93] S. PU and G. VOSSELMAN: “Knowledge based reconstruction of building models from terrestrial laser scanning data”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 64.6 (2009), pp. 575–584.
- [94] J. PUJOL et al.: “Hamilton, Rodrigues, Gauss, quaternions, and rotations: a historical reassessment”. In: *Communications in Mathematical Analysis* 13.2 (2012), pp. 1–14.
- [95] F. PUKELSHEIM: “The Three Sigma Rule”. In: *The American Statistician* 48.2 (1994), pp. 88–91.
- [96] T. RABBANI, F. VAN DEN HEUVEL, and G. VOSSELMANN: “Segmentation of point clouds using smoothness constraint”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 36.5 (2006), pp. 248–253.
- [97] A. RIETDORF: *Automatisierte Auswertung und Kalibrierung von scannenden Messsystemen mit tachymetrischem Messprinzip*. Series C, Nr. 582. Committee for Geodesy (DGK), Munich, 2005.

## REFERENCES

- [98] A. RIETDORF, F. GIELSDORF, and L. GRUENDIG: “A concept for the calibration of terrestrial laser scanners”. In: *Proceedings of INGEO 2004 and FIG Regional Central and Eastern European Conference of Engineering Surveying*. Bratislava, Slovakia. Vol. II. 2004, p. 13.
- [99] B. RIVEIRO, M. J. DEJONG, and B. CONDE: “Automated processing of large point clouds for structural health monitoring of masonry arch bridges”. In: *Automation in Construction* 72 (2016), pp. 258–268.
- [100] O. RODRIGUES: *Annales de Gergonne*. 1840.
- [101] R. ROTH, J THOMPSON, et al.: “Practical application of multiple pulse in air (MPiA) Lidar in large-area surveys”. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 37.Part B1 (2008), pp. 183–188.
- [102] P. J. ROUSSEUW and A. M. LEROY: *Robust regression and outlier detection*. Vol. 589. John Wiley & Sons, 1987.
- [103] R. B. RUSU, N. BLODOW, and M. BEETZ: “Fast point feature histograms (FPFH) for 3D registration”. In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. 2009, pp. 3212–3217.
- [104] H. SAMET: *Foundations of multidimensional and metric data structures*. Morgan Kaufmann, 2006.
- [105] H. SAMET and M. TAMMINEN: “Efficient component labeling of images of arbitrary dimension represented by linear bintrees”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.4 (1988), pp. 579–586.
- [106] M. SCAIONI: “Direct georeferencing of TLS in surveying of complex sites”. In: *Proceedings of the ISPRS Working Group 4* (2005), pp. 22–24.
- [107] B. SCHMITZ, C. HOLST, T. MEDIC, D. D. LICHTI, and H. KUHLMANN: “How to Efficiently Determine the Range Precision of 3D Terrestrial Laser Scanners”. In: *Sensors* 19.6 (2019), p. 1466.
- [108] T. SCHULZ: “Calibration of a terrestrial laser scanner for engineering geodesy”. PhD thesis. ETH Zurich, 2008.
- [109] W. SCHWARZ: “Vermessungsverfahren im Maschinen-und Anlagenbau”. In: *Schriftenreihe des DVW* (1995).
- [110] K. SCHWIDEFSKY and F. ACKERMANN: *Photogrammetrie: Grundlagen, Verfahren, Anwendungen*. 7th ed. BG Teubner, Stuttgart, 1976.
- [111] V. SCHWIEGER and L. ZHANG: “Qualität in der Ingenieurgeodäsie – Begriff und Modellierung”. In: *Qualitätssicherung geodätischer Mess-und Auswerteverfahren, Beiträge zum 180. DVW-Seminar* 95 (2019), pp. 9–30.
- [112] R. SEDGEWICK: *Algorithms in C*. 3rd ed. Addison-Wesley, 1998.
- [113] L. G. SHAPIRO: “Connected component labeling and adjacency graph construction”. In: *Machine Intelligence and Pattern Recognition*. Vol. 19. Elsevier, 1996, pp. 1–30.

## REFERENCES

- [114] T. D. SMITH and B. F. McMILLAN: “A Primer of Model Fit Indices in Structural Equation Modeling.” In: *Mathematics* (2001).
- [115] S. SOUDARISSANANE, R. LINDENBERGH, M. MENENTI, and P. TEUNISSEN: “Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.4 (2011), pp. 389–399.
- [116] C. STAHLBERG: “Eine vektorielle Darstellung des Einflusses von Ziel- und Kippachsenfehler auf die Winkelmessung”. In: *Zeitschrift für Geodäsie, Geoinformation und Landmanagement (ZfV)* 122 (1997), pp. 225–235.
- [117] D. STALLMANN: DGAP Notes. Mar. 2008.
- [118] P. TEUNISSEN: Adjustment theory: an introduction. VSSD Press, 2000.
- [119] N. TILLY, D. HOFFMEISTER, Q. CAO, V. LENZ-WIEDEMANN, Y. MIAO, and G. BARETH: “Precise plant height monitoring and biomass estimation with terrestrial laser scanning in paddy rice”. In: *Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences Conference, Antalya, Turkey*. Vol. III3. 2013.
- [120] A.-V. VO, L. TRUONG-HONG, D. F. LAEFER, and M. BERTOLOTTO: “Octree-based region growing for point cloud segmentation”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 104 (2015), pp. 88–100.
- [121] T. VOEGTLE, I. SCHWAB, and T. LANDES: “Influences of different materials on the measurements of a terrestrial laser scanner (TLS)”. In: *Proc. of the XXI Congress, The International Society for Photogrammetry and Remote Sensing*. Vol. 37. 2008, pp. 1061–1066.
- [122] W. VON HANSEN: “Robust automatic marker-free registration of terrestrial scan data”. In: *Proc. Photogramm. Comput. Vis* 36 (2006), pp. 105–110.
- [123] W. VON HANSEN, H. GROSS, and U. THOENNESSEN: “Line-based registration of terrestrial and airborne LIDAR data”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 37 (2008), pp. 161–166.
- [124] W. WAGNER: “Radiometric calibration of small-footprint full-waveform airborne laser scanner measurements: Basic physical concepts”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 65.6 (2010), pp. 505–513.
- [125] W. WAGNER, A. ULLRICH, V. DUCIC, T. MELZER, and N. STUDNICKA: “Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner”. In: *ISPRS journal of Photogrammetry and Remote Sensing* 60.2 (2006), pp. 100–112.
- [126] M. WANG and Y.-H. TSENG: “Lidar data segmentation and classification based on octree structure”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 35 (2004), pp. 1682–1750.
- [127] A. WEHR and U. LOHR: “Airborne laser scanning—an introduction and overview”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 54.2-3 (1999), pp. 68–82.

## REFERENCES

- [128] B WITTE and P SPARLA: Vermessungskunde und Grundlagen der Statistik für das Bauwesen. 8th ed. Herbert Wichmann Publishing, Heidelberg, 2015.
- [129] D. WUJANZ: Terrestrial laser scanning for geodetic deformation monitoring. Series C, Nr. 775. Committee for Geodesy (DGK), Munich, 2016.
- [130] D. WUJANZ, M. BURGER, M. METTENLEITER, and F. NEITZEL: “An intensity-based stochastic model for terrestrial laser scanners”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 125 (2017), pp. 146–155.
- [131] D. WUJANZ, M. BURGER, M. METTENLEITER, and F. NEITZEL: “Modellbasierte Standpunktplanung für terrestrische Laserscanner unter Nutzung eines intensitätsbasierten stochastischen Modells”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2016, pp. 60–71.
- [132] D. WUJANZ, M. BURGER, M. METTENLEITER, F. NEITZEL, F. TSCHIRSCHWITZ, and T. KERSTEN: “Ein intensitätsbasiertes stochastisches Modell für terrestrische Laserscanner - Erste Untersuchungen der Z+ F IMAGER 5006h und 5010”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2017, pp. 16–26.
- [133] D. WUJANZ, M. BURGER, F. TSCHIRSCHWITZ, T. NIETZSCHMANN, F. NEITZEL, and T. KERSTEN: “Bestimmung von intensitätsbasierten stochastischen Modellen für terrestrische Laserscanner basierend auf 3D-Punktwolken”. In: *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage*. Herbert Wichmann Publishing, Heidelberg, 2018, pp. 155–166.
- [134] D. WUJANZ, M. BURGER, F. TSCHIRSCHWITZ, T. NIETZSCHMANN, F. NEITZEL, and T. P. KERSTEN: “Determination of intensity-based stochastic models for terrestrial laser scanners utilising 3D-point clouds”. In: *Sensors* 18.7 (2018), p. 2187.
- [135] D. WUJANZ, C. HOLST, F. NEITZEL, H. KUHLMANN, W. NIEMEIER, and V. SCHWIEGER: “Survey configuration for terrestrial laser scanning”. In: *Allgemeine Vermessungs-Nachrichten (AVN)* 123.6 (2016), pp. 158–169.
- [136] D. WUJANZ, M. METTENLEITER, M. BURGER, and F. NEITZEL: “Viewpoint planning for terrestrial laser scanning utilising an intensity based stochastic model”. In: *3rd Joint International Symposium on Deformation Monitoring (JISDM)*. 2016.
- [137] M. ZÁMEČNÍKOVÁ, A. WIESER, H. WOSCHITZ, and C. RESSL: “Influence of surface reflectivity on reflectorless electronic distance measurement and terrestrial laser scanning”. In: *Journal of Applied Geodesy* 8.4 (2014), pp. 311–326.
- [138] X. ZHAO, G. KERMARREC, B. KARGOLL, H. ALKHATIB, and I. NEUMANN: “Influence of the simplified stochastic model of TLS measurements on geometry-based deformation analysis”. In: *Journal of Applied Geodesy* 13.3 (2019), pp. 199–214.
- [139] ZOLLER+FRÖHLICH: Z+F Imager 5006h Owners Manual. 2010.