

Unterschrift des Betreuers



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

D I P L O M A R B E I T

Registrierung von TLS Scans in Waldgebieten mittels ICP

Ausgeführt am

Department für Geodäsie und Geoinformation
Forschungsgruppe Photogrammetrie
der Technischen Universität Wien

unter der Anleitung von

Univ.Prof. Dipl.-Ing. Dr.techn. Norbert Pfeifer
Senior Scientist Dipl.-Ing. Dr.techn. Markus Hollaus

durch

David Reifeltshammer, BSc

MatrNr.: 00927256

Hirschengasse 19/1/8

1060 Wien - Österreich

Wien, Oktober 2017

David Reifeltshammer

Vielen Dank an Univ.Prof. Dr. Norbert Pfeifer, Dr. Markus Hollaus,
meinen Freunden und meiner Familie, ohne welche die
Erstellung dieser Diplomarbeit nicht
möglich gewesen wäre.

Zusammenfassung

Die immer schneller werdende Entwicklung von terrestrischen Laserscannern erhöht den Einsatz dieser Aufnahmemethode bei zahlreichen Anwendungen. Dabei ist der wichtigste Schritt der Datenverarbeitung, die Registrierung der einzelnen Scans zu einer gemeinsamen 3D-Punktwolke. Dazu müssen während der Feldmessungen sogenannte Passpunkte vermarktet werden und von mehreren TLS Standpunkten sichtbar sein. Mittels dieser Punkte kann im Postprocessing die Orientierung der einzelnen Punktwolken bestimmt werden.

Somit stellt sich die Frage, die Scans anfangs nur grob zu registrieren und dann die genaue Einpassung mit dem Iterative Closest Point Algorithmus zu realisieren. Dabei können bei der Feldarbeit die Passpunkte minimiert beziehungsweise ganz weggelassen werden.

Ein großes Problem bei diesem Verfahren ist, dass die Laufzeit des ICP- Algorithmus sehr stark von der Anzahl der verwendeten Datenpunkte abhängig ist. Um diese möglichst gering zu halten, werden die Punktwolken im Vorfeld stark reduziert. Da die Aufnahme in einem Waldstück stattgefunden hat, war die Überlegung, einzelne Bäume zur Registrierung zu verwenden. Dazu werden solche Bereiche automatisch detektiert, aus dem kompletten Scan ausgeschnitten und mit den verbleibenden Datenpunkten die Registrierung neu bestimmt.

Zusätzlich wurden noch Szenarien getestet, bei denen Schichten in verschiedenen Höhen aus den Punktwolken geschnitten und diese Bereiche für den ICP- Algorithmus verwendet wurden. Zur Beurteilung des am besten geeigneten Punktebereichs wurden Laufzeit, statistische Kennwerte und die grafische Darstellung herangezogen.

Bei der Bewertung zeigt sich, dass es auf jeden Fall sinnvoll ist, die Daten vorher aufzubereiten und nur noch gewisse Bereiche der Punktwolke für die Bestimmung der Registrierung zu verwenden. Auch wenn diese Vorbereitung einen nicht vernachlässigbaren Anteil der Laufzeit ausmacht, ist die komplette Dauer um einiges kürzer, als mit dem gesamten Scan. Erst bei der Verwendung von Punkten über einer Höhe von 6 m wird die Registrierung zunehmend schlechter und ist nicht mehr zu empfehlen.

Abstract

The accelerating development of terrestrial laser scanners increases the use of this recording method in numerous applications. Thereby, the most important step in data processing is the registration of the individual scans into a common 3D point cloud. To ensure this, so-called control points must be marked during the field measurements and be visible by several TLS positions. By means of these points, the orientation of the individual point clouds can be determined in the postprocessing stage.

However, there is the idea of registering the scans only roughly at the beginning and then realizing the exact fit with the Iterative Closest Point algorithm. Over the course of the field work the amount of points can be minimized or omitted completely.

A major problem with this method is that the runtime of the ICP algorithm is highly depending on the number of data points used. To keep the number as low as possible, the point clouds are greatly reduced in advance. Since the recording took place in a piece of woodland, the idea was to use individual trees for registration. For this purpose, such areas are automatically detected, cut out of the complete scan and the registration is redefined with the remaining data points.

In addition, scenarios were tested in which layers at different heights were cut from the point clouds. These areas were then used for the ICP algorithm. In order to assess the most suitable point range, criteria such as runtime, statistical values and the graphical representation were analyzed.

The evaluation shows that it is always useful to prepare the data beforehand and only to use certain areas of the point clouds for determining the registration. Even if this preparation takes a non-negligible part of the runtime, the entire duration is much shorter than with an entire scan. It is only above a height of 6 m that registration becomes increasingly worse and is no longer recommended.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	3
2.1	Terrestrisches Laserscanning (TLS)	4
2.1.1	Distanzmessung	5
2.1.2	Verknüpfung von Punktwolken	10
2.1.3	Georeferenzierung	13
2.2	Datenverarbeitung	15
2.2.1	Orientation and Processing of Airborne Laser Scanning Data (OPALS)	15
2.2.2	Matlab	17
2.3	Iterative Closest Point (ICP)	18
2.3.1	Korrespondierende Punkte	19
2.3.2	Funktionsweise des ICP	23
2.4	Simultaneous Localization and Mapping (SLAM)	26
3	Datenerfassung	29
3.1	Untersuchungsgebiet	29
3.2	Geräte	30
3.3	Messablauf	31
3.4	Kombination der Messdaten	34
3.5	Referenzwerte der Stämme (DBH)	35
4	Methodik	37
4.1	Vorbereitung	38
4.1.1	Höhe über DGM	39
4.1.2	Normalvektoren	40
4.1.3	Lokale Maxima detektieren	42

4.2	Bäume extrahieren	45
4.3	Schichten aus Punktwolken schneiden	47
4.4	ICP anwenden	49
4.5	Validierung	52
5	Ergebnisse und Diskussion	54
5.1	Laufzeit	55
5.2	Statistische Kennwerte	61
5.3	Grafische Darstellung	64
6	Resümee und Ausblick	70

Kapitel 1

Einleitung

Mit der stetigen Entwicklung geodätischer Messinstrumente, wie in diesem Fall, dem Laserscanner, sinken die Anschaffungskosten und es wird immer leichter, Daten mit diesem Gerätetyp aufzunehmen. Dadurch steigt die Menge an neu gewonnenen Rohdaten, die zur weiteren Verwendung entsprechend aufbereitet werden müssen. Im Falle des Laserscanners ist es die Registrierung der einzelnen Punktwolken, welche von verschiedenen Standpunkten aufgenommen wurden. Dieser Prozess soll natürlich mit möglichst effizienten, genauen und am besten automatisierten Methoden abgewickelt werden.

Im Zuge dieser Arbeit wurden ein Workflow und Methodiken entwickelt, um grob registrierte Punktwolken mit der Verwendung des Iterative Closest Point Algorithmus besser zu einander auszurichten. [Chen und Medioni, 1992] und [Besl und McKay, 1992] registrierten erstmals zwei Datensätze, ohne Pass- oder Identpunkte zu verwenden. Es wird hierbei iterativ der Abstand zwischen korrespondierenden Punktepaaren verringert. Dadurch kann während der Feldarbeit Zeit eingespart werden, da weniger Aufwand für das Anbringen von Reflektorfolien zur Registrierung getrieben werden muss. Im besten Fall kann dieser Schritt komplett entfallen, wenn einige Passpunkte direkt aus markanten Stellen der Punktwolke gewählt werden und so eine grobe Orientierung erreicht wird. Die genaue Registrierung erfolgt dann mit dem zuvor erwähnten Algorithmus.

Da der ICP- Algorithmus bei großen Datenmengen sehr laufzeitintensiv ist, wird in dieser Arbeit versucht, die Anzahl der Punkte zuvor durch verschiedene Methoden zu reduzieren, um die Dauer des ICP möglichst gering zu halten und trotzdem gleichwertige Ergebnisse zu erhalten. Nachdem das Testgebiet der Aufnahmen ein Waldstück in Vorarlberg war, entstand die Überlegung, automatisch einzelne Bäume aus der Punktwolke zu detektieren und nur diese Bereiche für die Bestimmung der Transformations-

parameter zu verwenden. Zusätzlich wurden Versuche mit Schnitten durch die Daten in verschiedenen Höhen angestellt, um die Menge der Daten und somit die Laufzeit noch weiter zu verkürzen.

Die vorliegende Arbeit beschreibt in Kapitel 2 die Grundlagen von terrestrischem Laserscanning inklusive der verschiedenen Distanzmessverfahren. Er werden Methoden zur Verknüpfung der einzelnen Scans und die Georeferenzierung der gesamten Punktwolke behandelt.

Weiters wird die Datenverarbeitung mit OPALS inklusive den Modulen und eine Einführung in den Iterative Closest Point Algorithmus gegeben.

In Kapitel 3 wird die Akquisition der Rohdaten beschrieben. Es wird eine Übersicht des Untersuchungsgebietes, der verwendeten Geräte sowie die Messung selbst und die Kombination der verschiedenen Messmethoden gegeben.

Die Aufbereitung und Prozessierung der Daten ist in Kapitel 4 beschrieben. Es wird der Workflow aufgezeigt und die einzelnen Schritte daraus näher erklärt. Die Ergebnisse dieser Verarbeitung werden dann in Kapitel 5 diskutiert. Die Bewertung der einzelnen Szenarios erfolgt durch folgende Punkte:

- Laufzeit
- Statistische Kennwerte
- Grafische Darstellung

Kapitel 6 fasst noch einmal die Erkenntnisse zusammen und gibt einen Ausblick auf weitere Forschungsmöglichkeiten.

Kapitel 2

Grundlagen

Um die angewandten Methoden zur Datenbeschaffung und weiters die Verarbeitung besser verstehen zu können, werden in diesem Kapitel die grundlegenden Verfahren und Methoden näher beschrieben.

Im Abschnitt 2.1 werden verschiedene Bereiche des terrestrischen Laserscanning (TLS) behandelt. Unter anderem gibt es einen Einblick in drei verschiedene Distanzmessverfahren und deren Eigenschaften. Es werden Methoden aufgezeigt, wie die Scans der unterschiedlichen TLS-Standpunkte zueinander ausgerichtet und die Verknüpfung untereinander hergestellt werden kann. Abschließend wird noch die Georeferenzierung diskutiert. Dies ist die Transformation der Punktwolke in ein übergeordnetes Koordinatensystem, in diesem Fall das Landeskoordinatensystem.

In der vorliegenden Arbeit basiert die Datenverarbeitung größtenteils auf zwei Programmen, die in Abschnitt 2.2 beschrieben werden. Zum einen handelt es sich um OPALS, welches von der Technischen Universität Wien entwickelt wird und für die Verarbeitung von Laserscannerdaten geeignet ist, zum anderen wird Matlab verwendet, um die Daten darzustellen und Analysen durchzuführen.

Das Hauptaugenmerk dieser Arbeit liegt aber in der Verwendung des ICP-Algorithmus. Richtigerweise würde dieser auch zum Abschnitt 2.2 gehören, wird aber auf Grund der Komplexität und Wichtigkeit dieses Verfahren hier extra in Abschnitt 2.3 diskutiert.

Abschließend wird noch eine kurze Einführung in *Simultaneous Localization and Mapping (SLAM)* gegeben, da dieses Verfahren durch den immer größer werdenden Einsatz von mobilen Aufnahmesystemen zusehends wichtiger wird.

2.1 Terrestrisches Laserscanning (TLS)

Mehr als zwei Jahrzehnte ist es her, dass Laserscanning in der Geodäsie zum ersten Mal verwendet wurde. Durch die stetige Entwicklung ist dieses Verfahren mittlerweile zu einer sehr wichtigen und effizienten Methode geworden, um Daten zu erfassen. Der Unterschied zur Photogrammetrie liegt in der verschiedenen Art der Aufnahme. Während die Photogrammetrie ohne zusätzliche Energiequellen auskommt, um ein Objekt abzubilden, wird beim Laserscanner ein Strahl ausgesandt und wieder empfangen. Solch eine Vorgehensweise wird auch als aktives Verfahren bezeichnet. Dieses Laserlicht ist stark gebündelt, monochromatisch und kohärent. [vgl. Kraus, 2004, S. 449ff]

Das Ergebnis einer solchen Laserscanner Messkampagne ist eine Punktwolke bestehend aus einer Menge von räumlichen Einzelpunkten. Jeder dieser Punkte kann, je nach Gerät, zusätzliche Informationen (Amplitude, Number of Echos, Farbinformationen, ...) gespeichert haben, die zur späteren Auswertung und Analyse verwendet werden können.

Prinzipiell gibt es zwei verschiedene Arten, einen Laserscanner zu verwenden. Diese unterscheiden sich entscheidend im Standort des Geräts. Erfolgt die Messung im bewegten Zustand, spricht man von kinematischem Laserscanning. Hierbei wird der Scanner auf Plattformen wie Auto, Flugzeug oder Unmanned Aerial Vehicle (UAV) montiert und von dort aus die Daten aufgenommen. Wird hingegen der Laserscanner während der Messung nicht bewegt, spricht man von terrestrischem Laserscanning. Da die Daten für die vorliegende Arbeit ausschließlich mit der zweiten Methode aufgenommen wurden, wird im Weiteren näher auf diese eingegangen.

Bei einem terrestrischen Laserscanner TLS spricht man von einem polaren Messsystem. Ebenso fallen elektronische Tachymeter, Lasertracker und ähnliche Instrumente in diese Kategorie. Diese haben als Gemeinsamkeit, dass sie zwei Winkel (Horizontalwinkel ϕ_i und Zenitwinkel Θ_i) und eine Schrägstrecke S_i (siehe 2.1.1) messen, um die Koordinaten des Objektpunkts P_i zu bestimmen. Abbildung 2.1 veranschaulicht dieses Messprinzip. [vgl. Rietdorf, 2005]

Zur Bestimmung der kartesischen Koordinaten des Objektpunkts wird Formel 2.1 verwendet.

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} S_i \cdot \sin(\Theta_i) \cos(\phi_i) \\ S_i \cdot \sin(\Theta_i) \sin(\phi_i) \\ S_i \cdot \cos(\Theta_i) \end{pmatrix} \quad (2.1)$$

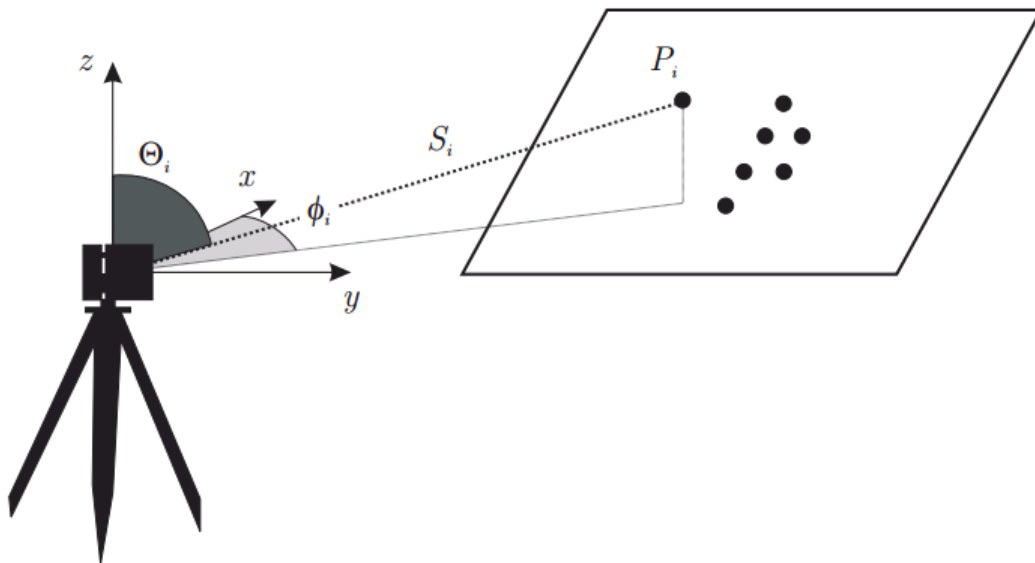


Abbildung 2.1: Messprinzip des TLS [Rietdorf, 2005]

Abbildung 2.2 zeigt die Funktionsweise eines Laserscanners. Im Zuge einer Messung dreht sich das Gerät einmal langsam um die Z-Achse. Aus dieser Drehung erhält man den Horizontalwinkel ϕ der polaren Beobachtungen. Zusätzlich wird der Laserstrahl mechanisch mit einem rotierenden Spiegel um den Winkel β abgelenkt. Dieser tastet ein vertikales Profil ab und speichert den Wert inklusive Schrägstrecke zu jedem Objektpunkt. [vgl. Zeinzinger, 2013]

Im Gegensatz zum Horizontalwinkel kann, aus konstruktionstechnischer Sicht, beim Vertikalwinkel keine volle Umdrehung gemessen werden. Der Öffnungswinkel variiert je nach Hersteller und Gerät und kann dem jeweiligen Datenblatt entnommen werden.

Die umgerechneten kartesischen Koordinaten der polaren Beobachtungen ergeben eine sogenannte Punktwolke mit verschiedensten Informationen. Diese können nun weiter verarbeitet werden.

2.1.1 Distanzmessung

Die polaren Beobachtungen beim solchen Messinstrumenten bestehen, wie oben erwähnt, aus zwei räumlichen Winkeln und einer Schrägstrecke. Die beiden Winkel werden geräteintern, elektronisch über Teilkreise abgetastet, umgewandelt und gespeichert.

Die Distanzmessung soll praktischerweise ohne vordefinierte Ziele, wie Spiegel oder Prismen, erfolgen. Man spricht von reflektorloser Distanzmessung, wobei der Laserstrahl von verschiedenen Oberflächen reflektiert wird.

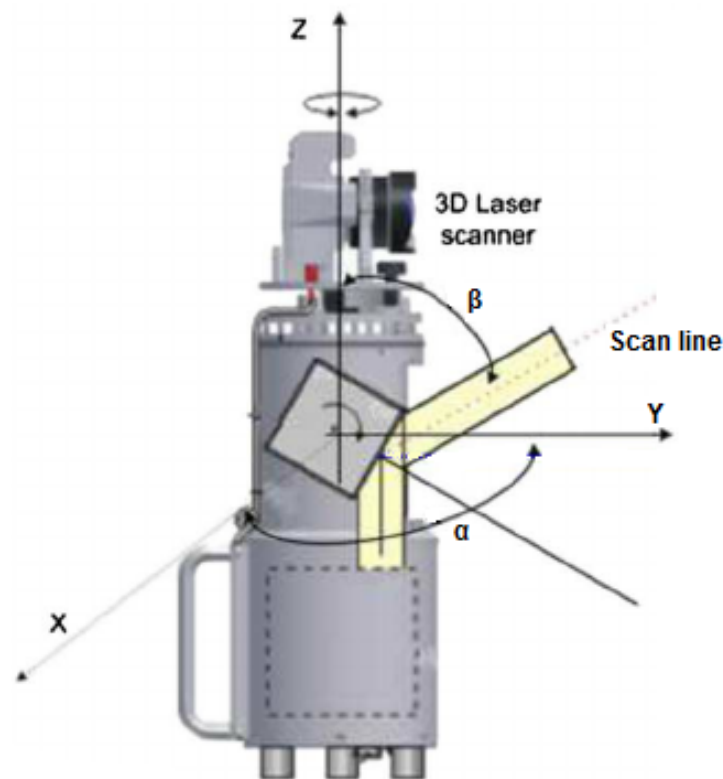


Abbildung 2.2: Prinzipielle Funktionsweise des TLS [nach Li u. a., 2009]

Der Begriff Laserscanning wird verwendet, wenn nicht nur zu einem bestimmten Punkt gemessen wird, sondern eine gesamte Fläche aufgenommen wird. Dadurch können Objekte aus der Realität in ein digitales Modell übertragen werden. [vgl. Rietdorf, 2005]

Nachfolgend werden die drei wichtigsten Verfahren zur reflektorlosen Distanzbestimmung näher erklärt.

- **Impulslaufzeitverfahren**

Das am häufigsten verwendete Verfahren bei Laserscannern ist das Impulslaufzeitverfahren. Es ist auch unter dem Begriff TOF (Time Of Flight) in der Literatur zu finden. Vom Sender ausgehend wird ein sehr kurzer Impuls (ca. 5 ns) am Objekt reflektiert und vom Empfänger wieder registriert. Dieses Schema ist in Abbildung 2.3 skizziert. [Hesse, 2008]

Die gemessene Distanz ergibt sich nun mit der Formel 2.2. Nach [Kraus, 2004] multipliziert man die Laufzeitdifferenz mit einer sogenannten Gruppengeschwindigkeit c_g . Diese ist um 0.03% geringer, als die Lichtgeschwindigkeit. In Abbildung 2.3 ist gut zu sehen, dass die, durch die Laufzeit berechnete Distanz noch halbiert werden muss, um auf die tatsächliche Distanz zu kommen.

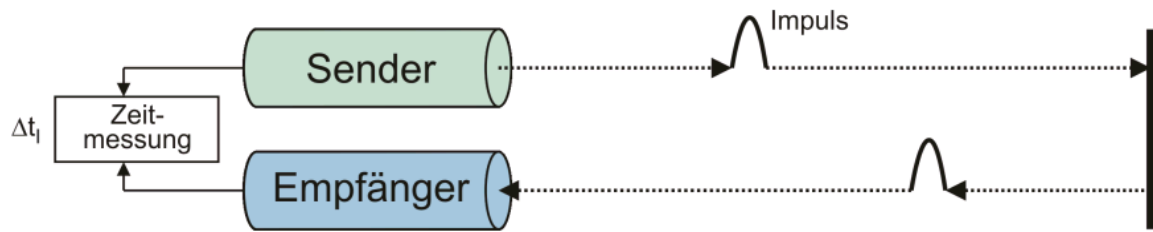


Abbildung 2.3: Prinzip des Impulslaufzeitverfahrens [Hesse, 2008]

$$S = \frac{c_g \cdot \Delta t}{2} \quad (2.2)$$

Um die erreichbaren Genauigkeiten aufzuzeigen, kann Formel 2.2 nach $\sigma_{\Delta t}$ umgeformt werden und man bekommt für eine gewünschte Streckengenauigkeit von 1mm Formel 2.3. [Nach Hesse, 2008]

$$\sigma_{\Delta t} = \frac{2 * \sigma S}{c_g} = \frac{2 \cdot 0.001m}{3 \cdot 10^8 m \cdot s^{-1}} = 0.0066ps \quad (2.3)$$

Hier ist ersichtlich, dass die Genauigkeit der Distanzmessung sehr stark von der Präzision der Zeitmessung abhängig ist. Die gemessene Entfernung fließt eigentlich nicht in diese Gleichung ein. [vgl. Pfeifer und Briese, 2007; Hesse, 2008]

Ein Laserscanner kann grundsätzlich auch die Intensität des zurückkommenden Signals messen. In der Literatur wird teilweise auch von Amplitude gesprochen. Es kann aber auch vorkommen, dass so ein Impuls nicht immer eindeutig reflektiert wird, sondern in versetzten Anteilen wieder empfangen wird. Diese einzelnen Signale werden als Echos oder Returns bezeichnet. Abbildung 2.4 zeigt eine solche empfangene Welle. Es sind die drei kleinen Spitzen von den Ästen erkennbar und als letzter Impuls das Dach des Hauses. Diese Echos werden vom ersten, zweiten, ... bis zum letzten Echo durchgezählt. [vgl. Zeinzinger, 2013]

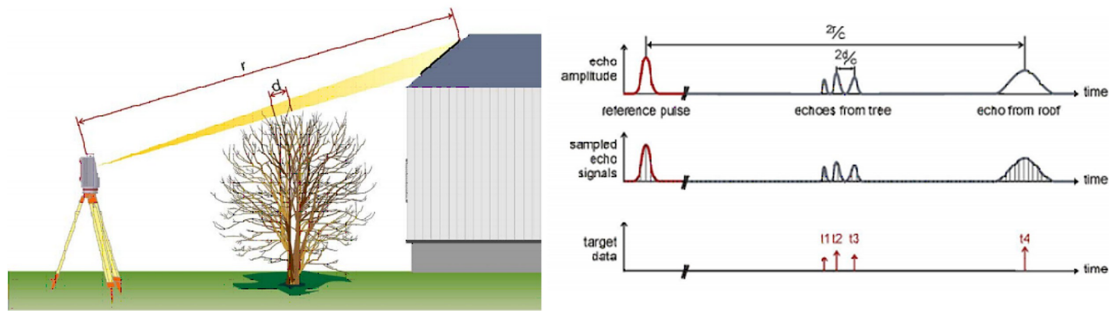


Abbildung 2.4: Unterschiedliche Intensitäten und Returns [RIEGL VZ-2000 manual, 2014]

• Phasenvergleichsverfahren

Im Vergleich zum Impulslaufzeitverfahren kann bei dieser Methode schneller und in höherer Genauigkeit gemessen werden. Nachteilig ist jedoch die niedrigere Reichweite, die sich auf ca 100 m beschränkt.

Abbildung 2.5 zeigt, dass das Phasenreststück $\Delta\varphi$ eindeutig aus der Differenz des ausgesandten φ^S und empfangenen φ^E Signals bestimmt werden kann. Nun fehlt aber noch die Anzahl der vollen durchlaufenen Phasen N wie in Formel 2.4 zu sehen ist. Dies wird außerhalb des Eindeutigkeitsbereichs realisiert, indem man Messungen mit unterschiedlichen, niederfrequenten Wellenlängen durchführt. Somit kann eine genaue Distanzmessung nur in Kombination mit Grob- und Feinmessung erfolgen. [vgl. Zeinzinger, 2013]

$$S = \frac{1}{2} \cdot \left(N + \frac{\varphi^E - \varphi^S}{2\pi} \right) \cdot \lambda \quad (2.4)$$

Durch Umformen der Formel 2.4 erhält man eine Angabe bezüglich Genauigkeit der gemessenen Strecke. In Formel 2.5 ist ersichtlich, dass die Genauigkeit umgekehrt proportional zur Wellenlänge steht. Das erklärt auch die schlechte Einsetzbarkeit bei größeren Distanzen. [Hesse, 2008].

$$\sigma_S^2 = \frac{1}{4\pi} \cdot (2\lambda^2 \cdot \sigma_\varphi^2 + \Delta\varphi^2 \cdot \sigma_\lambda^2) \quad (2.5)$$

Nach [Kahmen, 2006] kann als Größenordnung für die Angabe der Genauigkeiten auch $\frac{\lambda}{4000}m$ bis $\frac{\lambda}{8000}m$ angenommen werden.

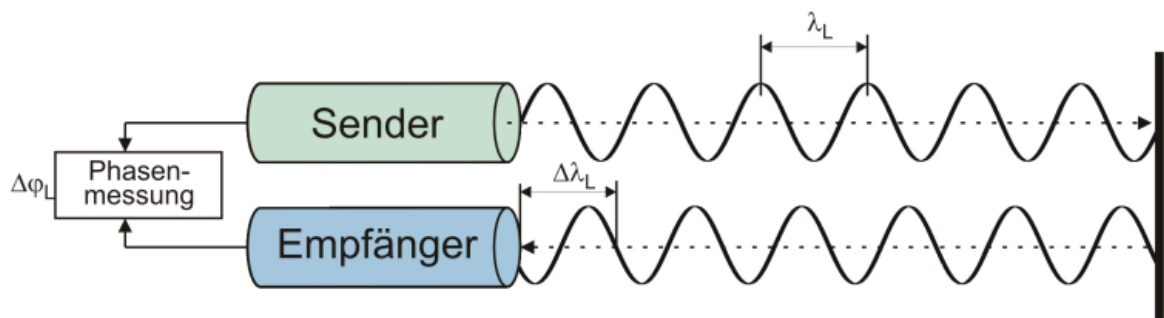


Abbildung 2.5: Prinzip des Phasenvergleichsverfahren [Hesse, 2008]

- **Lichtschnittverfahren**

Ein völlig anderes Aufnahmeprinzip ist das sogenannte Lichtschnittverfahren. Es wird hauptsächlich in extremen Nahbereichen verwendet. Hier spricht man von Entfernungen von wenigen Dezimetern bis hin zu zwei Metern, wo es aber im Vergleich zu den anderen zwei Verfahren, die genauesten Ergebnisse liefert. [Kraus, 2004]

Bei dieser Methode wird ein Laserstrahl nicht zur Messung der Entfernung verwendet, sondern nur, um eine Laser-Licht-Ebene auf das Objekt zu projizieren. Dieser Strahl muss soweit abgelenkt werden, dass das gesamte Objekt einmal beleuchtet wird. Mit einer Digitalkamera wird dann jedes dieser Profile aufgenommen und in Bildkoordinaten gespeichert. [Zeinzinger, 2013]

Abbildung 2.6 zeigt die Konfiguration des Messaufbaus. Hier ist auch ersichtlich, dass der Abstand B (meistens als Basis bezeichnet) zwischen Laser und CCD-Kamera bekannt sein muss. Durch einen räumlichen Vorwärtsschnitt werden die Koordinaten des Objektpunktes P bestimmt. Herleitung und nähere Erklärungen sind in [Kraus, 2004] zu finden. [Zeinzinger, 2013]

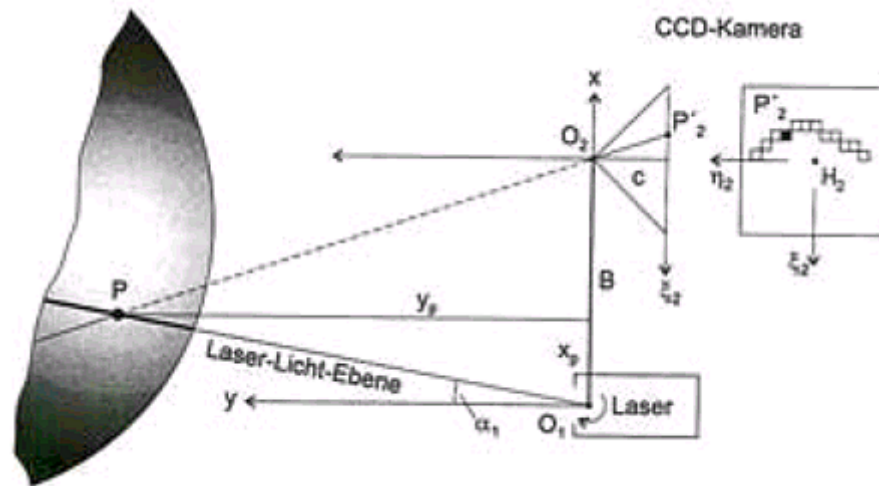


Abbildung 2.6: Prinzip des Lichtschnittverfahrens [Kraus, 2004]

2.1.2 Verknüpfung von Punktwolken

Da es in der Praxis meistens nicht ausreichend ist, die Daten von einem einzigen Standpunkt aufzunehmen, ist es notwendig, den Scanner im Zuge einer Messkampagne mehrmals an neuen Orten aufzustellen. Im Gegensatz zu einer einzigen Aufstellung hat das den Vorteil, dass diverse Schatten vermieden werden und die Bäume von allen Seiten aufgenommen werden. Die Herausforderung in der Nachbearbeitung der Daten besteht jetzt darin, diese einzelnen Punktwolken wieder in ein gemeinsames Koordinatensystem zu bringen. Dieser Vorgang wird auch Referenzierung genannt.

Das Problem beim Laserscanning im Vergleich zur terrestrischen Vermessung mit Tachymetern ist, dass beim Scanvorgang immer verschiedene Punkte aufgenommen werden. Dadurch geht der Vorteil der eindeutigen, identen Punkte verloren, die normalerweise für eine Transformation verwendet werden. Hierfür gibt es drei verschiedene Ansätze um die Referenzierung trotzdem erfolgreich durchzuführen. [Zeinzinger, 2013]

- **Verknüpfungspunkte**

Um doch wieder eindeutige Punkte zu bekommen, können spezielle Zielmarken, oder auch "Targets" genannt, in der Messregion und außen herum angebracht werden. Diese Ziele gibt es in verschiedenen Ausführungen, beginnend von ebenen Ausdrucken aus Papier im Schachbrett oder Dreiecksmuster, über retro-reflektierende Zielmarken, bis hin zu räumlichen Objekten wie Kugeln oder Zylindern (siehe Abbildung 2.7).

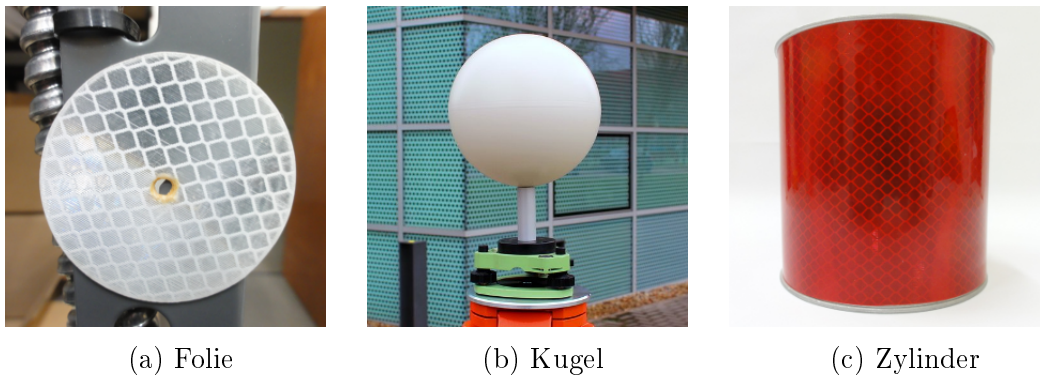


Abbildung 2.7: Verschiedene künstliche Zielmarken [unavco.org, 2017]

Ein Problem bei den ebenen Zielen ist, dass die Sichtbarkeit in Abhängigkeit vom Blickwinkel stark eingeschränkt sein kann. Abhelfen kann hier ein Gestell mit verstellbarer Steh- und Kippachse. Dieses muss aber vor jedem neuen Scan entsprechend ausgerichtet werden. Positiv ist aber der relativ günstige Anschaffungspreis. Gegenüber Verdeckungen sind die räumlichen Ziele robuster. [Bienert, 2013]

Diese angebrachten Targets haben im Scan einen sehr hohen Intensitätswert und sind dort leicht zu finden. Geeignete Software berechnet dann aus diesen Marken den Mittelpunkt und kann so korrespondierende Punkte für die Transformation finden.

Diese Methode hat den Nachteil, dass im Vorhinein alle Marken im und um das Messgebiet verteilt werden müssen. Auch ist darauf zu achten, dass mindestens drei identische Targets von zwei verschiedenen Standorten gesehen werden.

- **Geometrische Primitive**

Ein weiterer Ansatz, um idente Punkte für die Orientierung zu finden, ist die Verwendung von geometrischen Primitiven. Dazu zählen Formen wie Kugeln, Würfel, Zylinder im räumlichen Umfeld. Es gibt natürlich auch im 2D-Raum Objekte wie Ebenen, Kurven oder Linien. Zu bevorzugen sind hier die Primitiven im 3D-Raum um die winkelabhängigen Probleme zu minimieren. Bei Kugeln und Zylindern wird hier im Postprocessing der Mittelpunkt der Figuren bestimmt und anhand dieser Identpunkte die Registrierung durchgeführt. Ebene Primitive werden mit sogenannten Matching Algorithmen zur Deckung gebracht. [Zeinzinger, 2013]

Abhängig vom jeweiligen Aufnahmegebiet muss überlegt werden, ob genügend natürliche Formen vorhanden sind, oder künstliche Objekte in dem Areal positio-



Abbildung 2.8: Verwendete Messkugeln bei der Aufnahme [Wang u. a., 2016]

niert werden müssen. Bei Messungen von Industrieanlagen oder Innenaufnahmen von Gebäuden wird vermutlich eine aufreichend große Anzahl von geometrischen Primitiven vorhanden sein, seien es diverse Rohre, Wände, Säulen oder sonstige eindeutig definierbare Objekte.

Schwieriger wird es bei Arbeiten, bei denen wenige gut verwendbare Formen zu finden sind. Dort ist es sinnvoll, zusätzliche Objekte aufzustellen, um eine erfolgreiche Referenzierung der Punktwolken zu bekommen. Abbildung 2.8 zeigt eine künstliche Kugel, welche bei der Aufnahme der Daten für diese Arbeit verwendet wurde.

- **Komplette Punktwolke**

Eine weitere Möglichkeit ist das Berechnen der Orientierung anhand der gesamten Punktwolke anstatt einzelne Identpunkte zu verwenden. Voraussetzung ist hier ein ausreichend großer Überlappungsbereich der unterschiedlichen Punktwolken. Dieses Verfahren wird Iterative Closest Point (ICP) genannt und auf Grund seiner Komplexität und Wichtigkeit in dieser Arbeit, in Abschnitt 2.3 genauer behandelt.

2.1.3 Georeferenzierung

Unter dem Begriff *Georeferenzierung* versteht man den Zusammenhang zwischen räumlichen Daten und deren Bezug zur Erdoberfläche. In der vorliegenden Arbeit ist darunter die Transformation vom Sensorkoordinatensystem in ein gemeinsames übergeordnetes, in diesem Fall das Landeskoordinatensystem, gemeint. [Haring, 2007]

Grundsätzlich ist aber für jeden Anwendungsbereich zu überlegen, welche Anforderungen gegeben sind. So macht es zum Beispiel bei Indoor-Aufnahmen nicht unbedingt Sinn, die Daten ins Landeskoordinatensystem zu transformieren, sondern es genügt ein gemeinsames lokales Projektkoordinatensystem.

Um die aufgenommenen Daten in ein globales Koordinatensystem zu transformieren, benötigt man die äußere Orientierung des Laserscanners. Dazu gehören die Standpunktkoordinaten (X_0, Y_0, Z_0) und die Orientierungswinkel (ω, ϕ, κ) des TLS. Sind diese Unbekannten bestimmt, können mittels Formel 2.6 die Koordinaten im globalen System berechnet werden. [Kraus, 2004]

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} + R_{\omega\phi\kappa} \begin{pmatrix} S_i \cdot \sin(\Theta_i) \cos(\phi_i) \\ S_i \cdot \sin(\Theta_i) \sin(\phi_i) \\ S_i \cdot \cos(\Theta_i) \end{pmatrix} \quad (2.6)$$

Zur Ermittlung der Parameter der äußeren Orientierung gibt es [nach Haring, 2007] drei verschiedene Möglichkeiten.

- **Direkte Georeferenzierung**

Bei dieser Methode werden die Parameter des Laserscanners mit Hilfe zusätzlicher Sensoren bestimmt. Eine häufige Vorgehensweise ist die Verwendung eines GNSS zur Bestimmung der Standpunktkoordinaten und einer inertialen Messeinheit IMU für die Orientierungen.

Theoretisch könnte man mit diesen Informationen die Koordinaten der Objektpunkte im globalen Koordinatensystem berechnen und diese für weitere Zwecke verwenden, ohne zusätzliche Passpunkte zwischen den Punktwolken zu definieren. Dazu müssten aber alle verwendeten Sensoren optimal aufeinander abgestimmt sein, was in der Praxis quasi unmöglich ist. Allein schon die Genauigkeit der GNSS-Messungen bringt erhebliche Verschlechterungen für die Qualität der aufgenommenen Daten. Dazu kommt, dass diese Art der Georeferenzierung völlig ohne Kontrolle abläuft und daher nicht wirklich zu empfehlen ist.

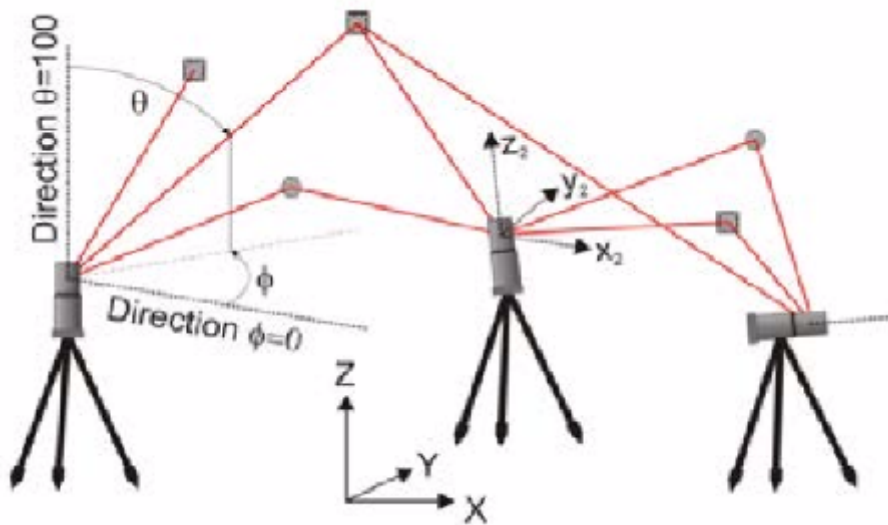


Abbildung 2.9: Pass (Quadrate)- und Verknüpfungspunkte von drei TLS-Standpunkten bei der indirekten Georeferenzierung [Kraus, 2004]

- **Indirekte Georeferenzierung**

Im Gegensatz zur direkten Bestimmung der äußeren Orientierungsparameter gibt es die indirekte Methode, die mit Hilfe von koordinativ bekannten Passpunkten, die benötigten Parameter bestimmt. Dazu werden verschiedene Zielmarken tachymetrisch eingemessen und deren Koordinaten berechnet. Diese Marken finden sich in der Punktwolke wieder und können zur Bestimmung der Parameter verwendet werden. Aus drei Passpunkten können neun Beobachtungsgleichungen aufgestellt werden und daraus die sechs unbekanntenen Orientierungsparameter bestimmt werden. Aufgrund dieser Überbestimmung von drei Beobachtungen erweist sich diese Methode als kontrolliert.

Abbildung 2.9 zeigt die Methode der indirekten Georeferenzierung mittels Pass- und Verknüpfungspunkten. Zweitere sind Punkte, die in mindestens zwei Scans erkennbar sind, aber keine übergeordneten Koordinaten besitzen. Diese werden zur weiteren Stabilisierung des Systems verwendet.

In der Praxis werden jedoch weitaus mehr als drei Passpunkte verwendet, um die Qualität der Objektpunkte zu erhöhen. Auch werden normalerweise die einzelnen Punktwolken vorher relativ zueinander ausgerichtet und erst danach die Transformation ins übergeordnete Koordinatensystem durchgeführt. [Zeinzinger, 2013]

- **Integrierte Georeferenzierung**

Die dritte Methode der Georeferenzierung kombiniert sowohl die zusätzlichen Sensoren (GNSS, IMU) als auch die bekannten Passpunkte. Dadurch wird eine höhere Genauigkeit und Zuverlässigkeit erzielt. Es können auch Kalibrierungsparameter von kombinierten Systemen bei der Ausgleichung mitbestimmt und somit etwaige Fehler kompensiert werden.

2.2 Datenverarbeitung

Im Folgenden werden zwei Programme vorgestellt, die bei der Realisierung dieser Arbeit hauptsächlich zum Einsatz gekommen sind. Zum einen handelt es sich um die Anwendung OPALS, welche die Forschungsgruppe Photogrammetrie und Fernerkundung die TU Wien entwickelt und speziell für die Prozessierung von Laserscannerdaten verwendet wird.

Matlab heißt das zweite Programm und ist eine kommerzielle Software zum Behandeln von mathematischen Problemen und zur Visualisierung derer. Der Name dieser Anwendung leitet sich von MATrix LABoratory ab, da es speziell für das Arbeiten mit Matrizen ausgelegt ist. [de.wikipedia.org, 2017]

Um die aufgenommenen Daten der Feldmessung in ein vom Computer lesbares Format zu bekommen, sind noch weitere Programme notwendig. Großteils wird hier *RiSCAN PRO* verwendet, welches vom Hersteller des Laserscanners entwickelt wird. Da diese Aufbereitung jedoch in einem vorhergehenden Projekt erfolgt ist, wird in dieser Arbeit nicht näher darauf eingegangen, sondern es werden diese vorhandenen Daten verwendet.

2.2.1 Orientation and Processing of Airborne Laser Scanning Data (OPALS)

Bei OPALS handelt es sich um ein Programm, das aus einer Anzahl von klar voneinander abgetrennten Modulen besteht. Jede dieser Komponente hat seine eigene Funktionalität, die je nach Bedarf verwendet und nacheinander kombiniert werden kann. Die Software eignet sich perfekt zum Prozessieren von großen Datenmengen, die bei Laserscanning Anwendungen in der Regel anfallen. Eigentlich ist OPALS für Airborne Laserscanning (ALS)-Daten entwickelt worden, ist aber mit der richtigen Verwendung

von Parametern auch für TLS geeignet.

OPALS kann auf drei verschiedene Arten ausgeführt werden:

- **Kommandozeilenprogramm:** Es sind sowohl Windows, als auch Linux Version vorhanden. Diese Methode erlaubt das Erstellen von Shell-Skripten zum automatisierten Abarbeiten von einzelnen Schritten. Der Zugriff über Kommandozeilen Skripts wird auch in dieser Arbeit verwendet.
- **Python Modul:** Ermöglicht das Verwenden von OPALS Funktionen in eigenen Python Programmen.
- **C++ Klasse über DLL Zugriff:** Ähnlich wie das Python Modul nur für C++ Programme.

Ein eigens entwickelter Datenmanager ODM verwaltet die Punkte auf effiziente Art und ermöglicht einen schnellen räumlichen Zugriff. Zusätzlich besteht die Möglichkeit, zu den Punkten verschiedenste, vordefinierte und eigene, Attribute zu verspeichern. [Mandlbürger u. a., 2010]

OPALS besteht aus knapp 40 einzelnen Modulen (Stand: 07/2017), wobei jedoch nur einige wenige in dieser Arbeit verwendet werden. Diese werden im Folgenden kurz beschrieben:

- **AddInfo**
Fügt einem Punkt beliebige vor- oder benutzerdefinierte Attribute hinzu. Die Werte können mittels verschiedenen Algorithmen und Filtern berechnet werden.
- **Normals**
Für einige Verarbeitungsschritte ist die lokale Ebeneninformation erforderlich. Diese wird mit *opalsNormals* berechnet. Auch hier können verschiedenste Nachbarschaftsbeziehungen, Berechnungsmethoden und ähnliches angegeben werden.
- **Import/Export**
Grundlegende Funktion von OPALS ist das Einlesen von verschiedenen Datenformaten in das eigene ODM-Format und der umgekehrte Weg zum Ausgeben der prozessierten Daten in diverse Formate. Auch hier kann wieder mit Filtern gearbeitet werden, um nur die relevanten Informationen zu verwenden.

- **Cell**

Mit dem Modul *Cell* ist es möglich, aus diversen Eingabe Dateien, verschiedene Rasterdaten zu erstellen. Diese werden dann meist als GeoTIFF abgespeichert, um den Raumbezug nicht zu verlieren. Als wichtigste Parameter benötigt man eine Rastergröße, das zu analysierende Attribut und die gewünschte Kenngröße (engl. *feature*). Hier gibt es zum Beispiel, die Punktdichte, Minimum, Maximum, Summe. In dieser Arbeit werden zum Bestimmen der Baumpositionen die Features Punktdichte *pdens* und die Suche nach dem nächsten Punkt vom Zentrum *center* verwendet.

- **StatFilter**

Dieses Modul verwendet Rasterdaten um statistische Kennwerte zu berechnen. Das Ergebnis ist ebenfalls ein Raster, wo zu jedem Pixel die neuen Informationen gespeichert sind. Zusätzlich zur Angabe des gewünschten Kennwertes, können verschiedene Nachbarschaften und Suchfenster angegeben werden.

- **Algebra**

Erstellt einen neuen Datensatz durch Kombination unterschiedlicher anderer Datensätze. Mit bestimmten Formeln, die sowohl generisch, als auch im Python Format möglich sind, können neue Werte berechnet werden.

- **ICP**

Das Modul wendet den ICP Algorithmus an, um Transformationsparameter für die Orientierung von mehreren Punktwolken zu bestimmen. Während der Datenverarbeitung dieser Arbeit, war das ICP Modul nur in der Windows Version verfügbar.

Die genauen Beschreibungen und restlichen Module können auf der Website des Projekts nachgelesen werden. [OPALS Team, 2017b]

2.2.2 Matlab

Das Programm *Matlab* wird seit den 70er Jahren vom amerikanischen Unternehmen *MathWorks* (vorher an der Universität New Mexico) entwickelt. Es handelt sich dabei um ein Programm zum Berechnen und Visualisieren von Daten.

Der Name leitet sich von *MATrix LABoratory* ab, wobei hier schon zu erkennen ist, dass das Hauptaugenmerk der Software auf dem Arbeiten mit Matrizen liegt.

Es gibt zahlreiche Erweiterungen zu der Grundversion, die Toolboxes genannt werden. Hier kann der Benutzer je nach Anwendungsbereich wählen und die Funktionalität anpassen.

Matlab kann interaktiv über direkte Befehle in der Kommandozeilenumgebung der Benutzeroberfläche (GUI) bedient werden. Bei komplexeren Aufgaben wird eine höhere Skript- bzw. Programmiersprache angeboten. Diese Anwendungen werden beim Ausführen in der GUI interpretiert und ausgewertet. [Teschl, 2013]

Im Zuge dieser Arbeit wird Matlab größtenteils zur Visualisierung der Ergebnisse verwendet. Die Schnittstelle zu den, mit OPALS prozessierten Daten, erfolgt über ASCII-Dateien. Diese enthalten die Koordinaten (X, Y, Z) , Normalvektoren (nX, nY, nZ) und die Höhe über dem Geländemodell (*NormalizedZ*) der Objektpunkte. Mit Matlab können diese Informationen einfach eingelesen und je nach Bedarf visualisiert werden.

2.3 Iterative Closest Point (ICP)

Eine alternative Methode zum Orientieren von Laserscanner Punktwolken ist der ICP-Algorithmus. Dieser hat den Vorteil, dass keine Pass- oder Verknüpfungspunkte notwendig sind. Das erhöht die Wirtschaftlichkeit vor allem bei großen Messkampagnen. Außerdem ist es nicht immer möglich, diese Punkte gleichmäßig zu verteilen. Gerade in größeren Höhen ist die Dichte an selbst definierten Passpunkten eher gering. Dies verschlechtert die Geometrie für die Transformation und diese Bereiche weisen höhere Klaffungen auf. Eine Hauptanforderung für ein positives Ergebnis des ICP Algorithmus ist jedoch eine grobe Vororientierung der einzelnen Punktwolken. Somit werden doch einige Verknüpfungspunkte in den Scans benötigt, um diese ungefähre Registrierung durchzuführen. [OPALS Team, 2017a]

Die Methode mit Iterative Closest Point nutzt die Geometrie der gegebenen Punktwolken und bringt diese in ein gemeinsames Koordinatensystem. Es wird versucht, die Abstände der einzelnen Punktwolken zu minimieren und daraus die Transformationsparameter zu bestimmen.

Das Problem bei dieser Methode ist jedoch, dass die vorhandenen Daten schon grob orientiert sein müssen, da sonst die Gefahr besteht, dass der Algorithmus nicht konvergiert bzw. nur lokale Minima gefunden werden. Es ist daher erforderlich, trotzdem einige Pass- oder Verknüpfungspunkte zu definieren, oder auf Methoden der direkten Georeferenzierung (Abschnitt 2.1.3) zurückzugreifen. Auch gibt es [nach Böhm und

Haala, 2005] Ansätze, mittels Low-Cost-GNSS Empfänger und digitalem Kompass die Näherungswerte zu bestimmen. [Haring, 2007]

Die Grundidee des ICP-Algorithmus kommt aus der Computer Vision und wird dort in der Qualitätskontrolle angewandt. Maßgeblich wurde er von [Besl und McKay, 1992; Chen und Medioni, 1992] entwickelt und erweitert. Mit der Methode sollte ein gefertigtes Werkstück mit dem CAD-Modell verglichen werden, um so etwaige Abweichungen aufzudecken.

Es gibt somit zwei Punktwolken [vgl. Haring, 2007]:

- **Modell-Punktwolke:** diese ist das vorgegebene CAD-Modell im übergeordneten Koordinatensystem.

$$M = \{m_1, m_2, \dots, m_{N_M}\}$$

- **Daten-Punktwolke:** liegt im Scanner-Koordinatensystem vor und soll ins übergeordnete System transformiert werden.

$$D = \{d_1, d_2, \dots, d_{N_D}\}$$

Für die Transformation zwischen den beiden Koordinatensystemen sind sechs Parameter notwendig. In Gleichung 2.7 sind die Unbekannten zu sehen und es werden der Einfachheit halber nun homogene Matrizen angeschrieben. Die Richtungswinkel (α, β und γ) werden in der Rotationsmatrix ($r_{11} \dots r_{33}$) verwendet. [Chen und Medioni, 1992]

$$T = T(\alpha, \beta, \gamma, t_x, t_y, t_z) = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.7)$$

2.3.1 Korrespondierende Punkte

Nach [Haring, 2007] wird im ersten Schritt zu jedem Datenpunkt d_i ein Punkt aus der Modell-Punktwolke gesucht, der den kleinsten euklidischen Abstand d besitzt. Diese wird aus der Wurzel der Summe der einzelnen Vektorkomponenten berechnet (siehe Gleichung 2.8).

$$d(\vec{x}_1, \vec{x}_2) = \|\vec{x}_1 - \vec{x}_2\| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (2.8)$$

Mittels der Gleichung 2.9 wird zu einem beliebigen Datenpunkt d_i die kürzeste euklidische Distanz zu einem, in der Modell-Punktwolke M enthaltenen, Punkt m_j gesucht.

$$d(d_i, M) = \min_{j \in \{1, \dots, N_M\}} (d(d_i, m_j)) \quad (2.9)$$

Dieses Punktepaar (d_i, m_j) wird *korrespondierende Punkte* genannt. Es sei zu erwähnen, dass ein Modellpunkt durchaus mit mehreren Datenpunkten ein Paar bilden kann. Aus diesen N_D Paaren wird nun eine Transformation $\hat{\mathbf{T}}$ berechnet und die Datenpunkte transformiert. Da es sich beim ICP-Algorithmus um ein iteratives Verfahren handelt, können nun diese Punkte für einen weiteren Durchgang verwendet werden. Wie oben schon erwähnt, kann es vorkommen, dass bei schlechten Näherungswerten der ICP nicht konvergiert bzw. ein falsches Ergebnis liefert. Daher ist es sinnvoll, geeignete Startwerte für die Transformation $(\mathbf{T}^{(0)})$ anzugeben.

Formel 2.10 zeigt die daraus entstehende Transformationskette nach N_I Iterationsschritten: [vgl. Haring, 2007]:

$$\mathbf{T} = \mathbf{T}^{(N_I)} = \hat{\mathbf{T}}^{(N_I)} \cdot \hat{\mathbf{T}}^{(N_I-1)} \cdot \dots \cdot \hat{\mathbf{T}}^{(1)} \cdot \mathbf{T}^{(0)} = \prod_{r=0}^{N_I-1} (\hat{\mathbf{T}}^{(N_I-r)}) \cdot \mathbf{T}^{(0)} \quad (2.10)$$

Grundsätzlich werden zum Suchen der korrespondierenden Punkte zwei Methoden angewandt:

- **Punkt-zu-Punkt:** Der Abstand wird direkt aus der euklidischen Distanz zwischen Daten- und Modellpunkt berechnet. [vgl. Besl und McKay, 1992]
Abbildung 2.10 zeigt diese Methode beispielhaft.

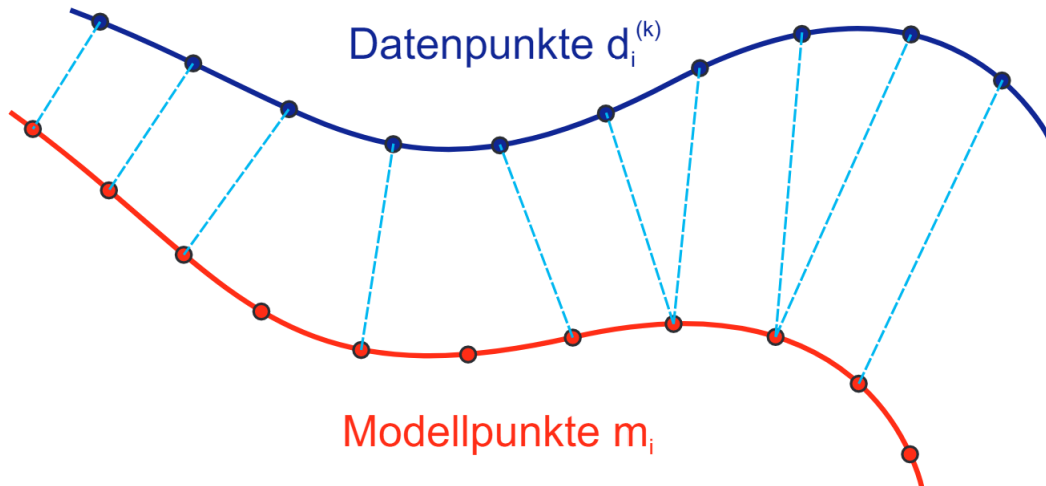


Abbildung 2.10: Abstand vom Datenpunkt zum euklidisch nächsten Modellpunkt [Haring, 2007]

Vorteile [nach Pulli, 1999; Mitra u. a., 2004; Haring, 2007]:

- + keine Normalvektoren notwendig (erspart aufwendige Rechenzeit)
 - + robuster gegenüber schlechten Näherungswerten
 - **Punkt-zu-Ebene:** Hier wird beim korrespondierenden Modellpunkt eine Tangente angelegt und der Normalabstand vom Datenpunkt zu dieser verwendet. [Chen und Medioni, 1992]
- In Abbildung 2.11 ist die Methode schematisch dargestellt.

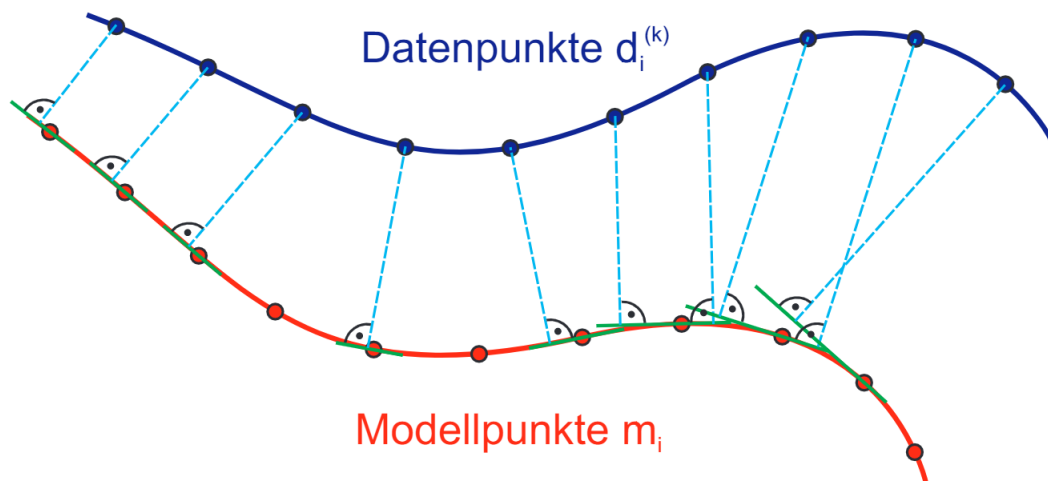


Abbildung 2.11: Normalabstand vom Datenpunkt zur Tangente vom euklidisch nächsten Modellpunkt [Haring, 2007]

Eine weitere Möglichkeit der Punkt-zu-Ebene Methode wird Normal-Shooting genannt und ist in Abbildung 2.12 dargestellt. Hier wird von jedem Datenpunkt die Flächennormale berechnet und mit dem Modell geschnitten. Es werden also nicht die Modellpunkte berücksichtigt, sondern interpolierte Punkte zwischen diesen. An diesen Schnittpunkten wird nun eine Tangente angelegt und der Normalabstand vom Datenpunkt zu dieser ist die verwendete Distanz.

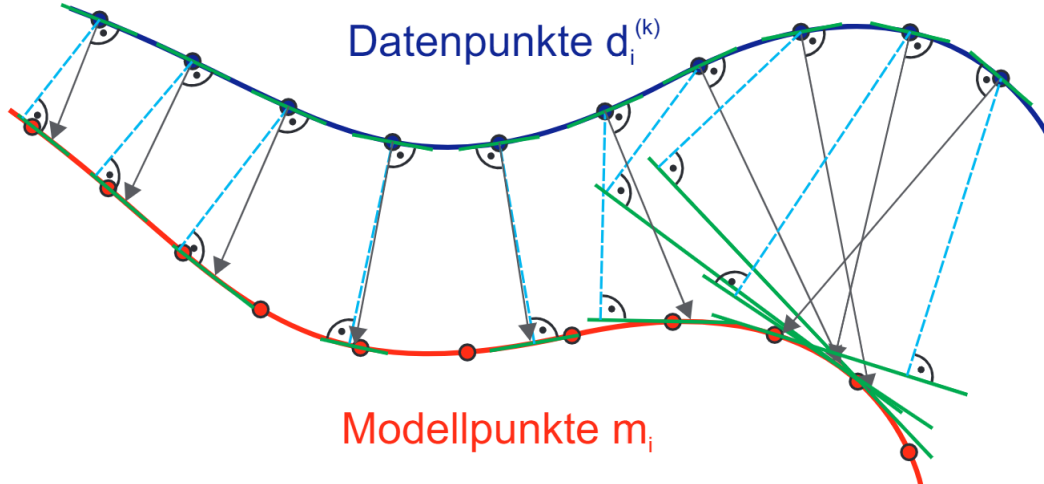


Abbildung 2.12: Normal-Shooting: Normalabstand vom Datenpunkt zur Tangente vom Schnittpunkt von Flächennormalen und Modell [Haring, 2007]

Nach [Rusinkiewicz und Levoy, 2001] liefert die Normal-Shooting Methode schneller ein Ergebnis, birgt jedoch bei schlechten Näherungswerten die Gefahr, zu einer falschen Lösung zu konvergieren. Es wird zwar der Abstand der beiden Punktwolken minimiert, auf Grund der schlecht verteilten, korrespondierenden Punkte aber nur ein lokales Minimum gefunden.

In Abbildung 2.13 ist der Vergleich der zwei Ansätze dargestellt.

Vorteile der Punkt-zu-Ebene Methoden [nach Pulli, 1999; Mitra u. a., 2004; Haring, 2007]:

- + falsch korrespondierende Punktpaare wirken sich weniger stark aus
- + benötigt weniger Iterationen
- + genauere, bzw. realistischere Ergebnisse

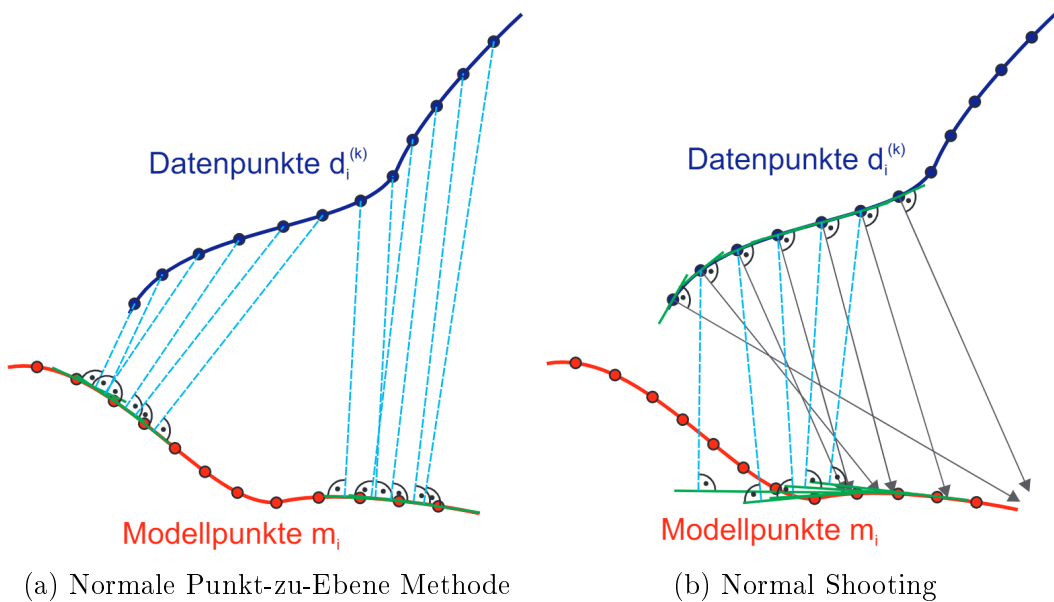


Abbildung 2.13: Vergleich der beiden Punkt-zu-Ebene Methoden bei schlechten Näherungswerten. (a) ist viel weniger anfällig im Gegensatz zum Normal-Shooting (b), wo das Konvergieren zu einem lokalen Minimum droht.

[Haring, 2007]

2.3.2 Funktionsweise des ICP

Nachdem nun die verschiedenen Methoden zum Aufsuchen der korrespondierenden Punkte beschrieben wurden, wird in diesem Abschnitt der generelle Iterative Closest Point- Algorithmus erläutert.

Die ersten Ansätze des ICP beschränkten sich zunächst auf die Verwendung von nur zwei Punktwolken. Erst später wurden Ideen für eine beliebige Anzahl vorgestellt.

Der Ablauf wird in mehrere Schritte unterteilt [vgl. Rusinkiewicz und Levoy, 2001; Glira u. a., 2015; Haring, 2007; OPALS Team, 2017a], welcher in Abbildung 2.14 beispielhaft dargestellt ist:

1. Auswahl der Punkte:

Im ersten Schritt werden Punkte ausgewählt, die für die Prozessierung verwendet werden sollen. Hier gibt es auch wieder diverse Ansätze, um die Anzahl der Punkte und somit den Rechenaufwand zu minimieren. Eine sinnvolle Variante ist, alle Modell- und Datenpunkte außerhalb des Überlappungsbereiches zu werfen, da diese keinen Beitrag zur Lösung liefern. Auch eine Selektion anhand von Flächennormalen und deren Winkel ist denkbar.

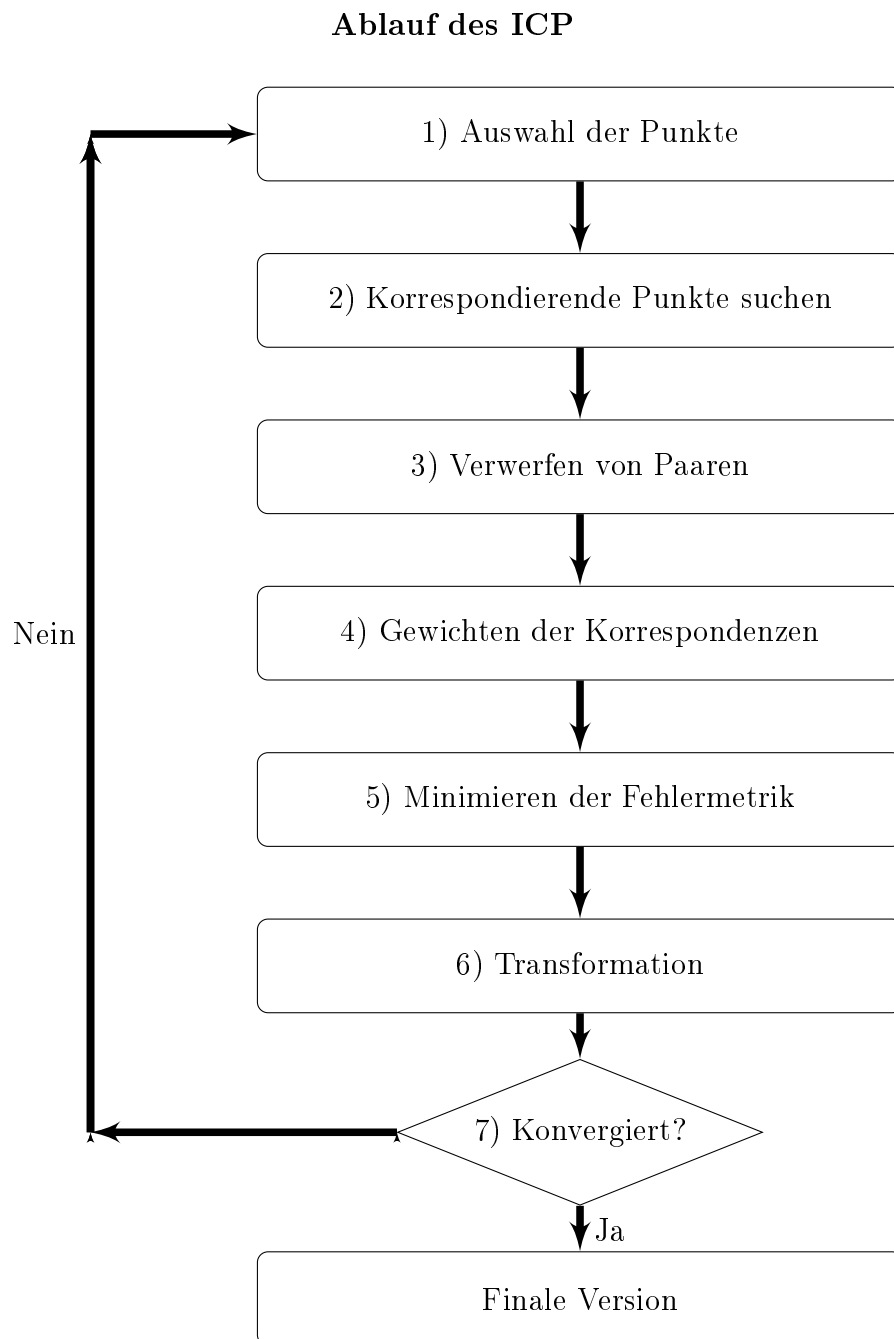


Abbildung 2.14: Funktionsweise des Iterative Closest Point - Algorithmus

2. Korrespondierende Punkte suchen:

Nach dem Ausdünnen der Punkte geht es darum, die korrespondierenden Punktpaare zu finden. Die verschiedenen Verfahren und Details dazu wurden bereits in Abschnitt 2.3.1 ausführlich diskutiert.

3. Verwerfen von Paaren:

Es besteht nun die Möglichkeit, Punktpaare anhand verschiedener Faktoren zu verwerfen. Diese Ausreißer können beispielsweise abhängig von der Distanz zwischen den korrespondierenden Punkten, dem Winkel zwischen den Normalvektoren oder der Rauheit detektiert werden. Es kann jedoch nach diesem Schritt trotzdem nicht garantiert werden, dass alle Ausreißer entfernt worden sind, so dass beim Minimieren der Fehlermetrik auf ein robustes Verfahren zurückgegriffen werden muss.

4. Gewichten der Korrespondenzen:

Nach dem Entfernen der unbrauchbaren Paare können nun die verbleibenden Korrespondenzen gewichtet werden. Hierzu werden normalerweise die gleichen Attribute, wie beim Detektieren der Ausreißer verwendet. Ziel ist dabei, den besser geeigneten Punktpaaren einen größeren Einfluss zum Bestimmen der Transformationsparameter zukommen zu lassen. Ist das nicht gewünscht, werden die Gewichtungen aller Paare gleich gesetzt.

5. Minimierung der Fehlermetrik:

Dieser Schritt ermittelt nun die Transformationsparameter um die Punktwolken möglichst gut einander anzunähern. Dazu wird die Methode der kleinsten Quadrate angewendet, um die Summe der Quadrate der Abstände der korrespondierenden Punkte zu minimieren. Bei der Punkt-zu-Punkt Methode gibt es eine geschlossene Lösung, die Parameter zu bestimmen. Bei der Punkt-zu-Ebene Methode, muss dagegen auf Linearisierung oder Verfahren zur Lösung nicht-linearer Gleichungssysteme (z.B. Levenberg-Marquardt) zurückgegriffen werden.

6. Transformation:

Diese berechneten Transformationsparameter werden nun auf die Datenpunkte angewandt. Dazu wird eine 6-Parameter Transformation (3 Rotationen, 3 Translationen) verwendet. Diese Art der Transformation wird auch *starr* (engl. *rigid*) genannt. Sie ist vergleichbar mit einer 3D-Helmert-Transformation, bei der der Maßstab auf 1 gesetzt wird.

7. Prüfen auf Konvergenz:

Im letzten Schritt des Algorithmus wird überprüft, ob das Ergebnis zufriedenstellend ist. Falls *Ja*, wird der ICP beendet und die letzte angewandte Transformation ist die finale Version. Wenn jedoch noch keine Konvergenz eingetreten ist, wird ein neuer Durchgang bei *Schritt 1* gestartet. Von diesem iterativen Vorgehen kommt auch der Name des Iterative Closest Point-Algorithmus.

Als Abbruchkriterium kann beispielsweise der mittlere Abstand der Punktwolken oder die relative Änderung zwischen zwei Iterationsschritten herangezogen werden. Auch ist es sinnvoll, die maximale Anzahl an Iterationen zu begrenzen, um im Fall einer schlechten, oder unmöglichen Orientierung, eine Endlosschleife zu vermeiden.

Im Falle dieser Arbeit werden aber nicht nur zwei Punktwolken (Modell und Daten) zueinander orientiert, sondern durchaus mehr Scans mit unterschiedlichen Überlappungsbereichen.

Hierzu gibt es Ansätze ([Chen und Medioni, 1992]), eine Punktwolke zu fixieren und diejenige, mit der größten Überlappung dazu auszurichten. Danach werden die zwei Scans vereint und Transformationsparameter für den nächsten bestimmt. Das wird so oft wiederholt, bis alle Punktwolken zu einer Gesamten vereint sind. Das Problem hierbei ist jedoch, dass sich ein Fehler beliebig weit fortpflanzt, weil die bereits orientierten Scans, durch einen weiteren, nicht mehr verändert werden [vgl. Bergevin u. a., 1996].

Eine weitere Möglichkeit wurde von [Neugebauer, 1997] vorgestellt und verwendet die Punkt-zu-Ebene Methode. Er erweitert diese für die Verwendung von beliebig vielen Punktwolken und bestimmt die Transformationsparameter von allen korrespondierenden Punkten gleichzeitig. Auch hier wird das nicht-lineare Gleichungssystem mittels Levenberg-Marquardt-Methode gelöst.

2.4 Simultaneous Localization and Mapping (SLAM)

Obwohl dieses Verfahren in dieser Arbeit nicht direkt behandelt wird, ist es trotzdem erwähnenswert, da es durch stetig steigende Anzahl von mobilen Aufnahmesystemen, immer interessanter wird. Es kommt aus der Robotik und behandelt die Thematik, Daten aufzunehmen und gleichzeitig diese Daten zu verwenden, um sich selbst zu orientieren und ihr “Wissen“ über die Umgebung erweitern. Es ist also durchaus mit künstlicher Intelligenz vergleichbar.

Die Idee dahinter besteht darin, ein mobiles Aufnahmegerät in eine, ihm unbekannte Umgebung zu setzen und es dort zu bewegen. Dabei kann die ganze Bandbreite von mobilen Robotern, UAV bis hin zu tragbaren 3D-Scannern verwendet werden. Es ist mit unterschiedlichsten Sensoren zur Datenaufnahme ausgestattet. Dabei können von Kamera, Laserscanner, über Radar, bis hin zu GNSS-Empfängern oder Beschleunigungssensoren alle möglichen Kombinationen verwendet werden. Anhand dieser Daten erzeugt nun das System eine Karte von seiner Umgebung und extrahiert verschiedenste markante Punkte (sogente *landmarks*) um später innerhalb der Karte navigieren zu können. Dies wird *Mapping* genannt.

Gleichzeitig wird aber auch versucht, diese detektierten Punkte mit bekannten, früheren Aufnahmen zu vergleichen und somit die eigene Position zu ermitteln (*Localization*). Sobald übereinstimmende Stellen gefunden werden, weiß das System, wo es sich befindet und kann so die vorher aufgenommenen Daten zu der bestehenden Karte hinzufügen. Hierbei werden ebenfalls Überlappungen der Aufnahmen verwendet, um den bekannten Bereich zu erweitern. [Vgl. Riisgaard und Blas, 2003; Nirmalakumari u. a., 2017]

Abbildung 2.15 zeigt noch den groben Ablauf, der während SLAM passiert.

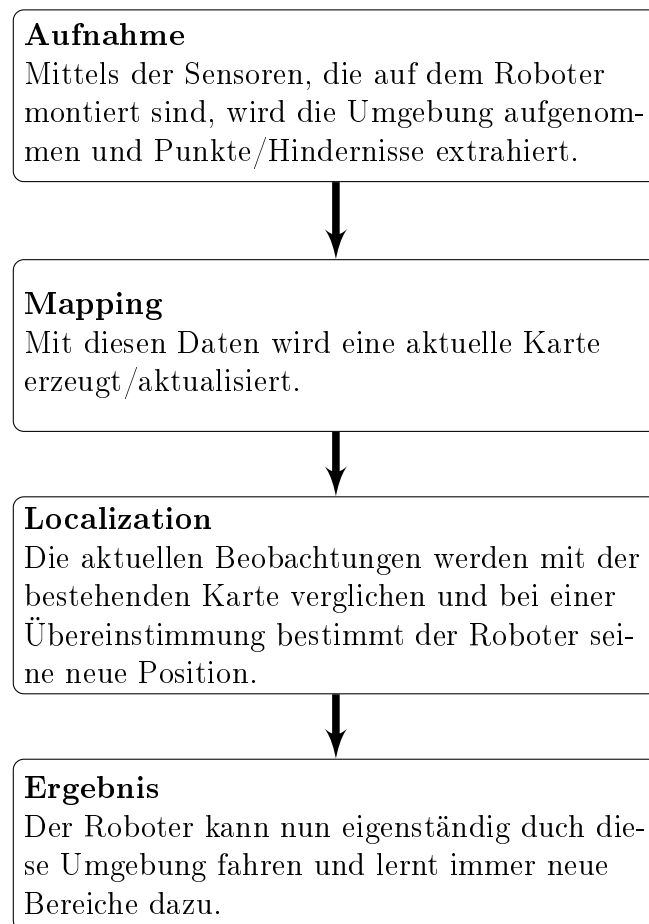


Abbildung 2.15: In SLAM durchgegangene Schritte [vgl. Nirmalakumari u. a., 2017]

Kapitel 3

Datenerfassung

In diesem Kapitel werden Informationen zur Datenakquisition beschrieben, um die Daten aus der Natur zu digitalisieren und für die Weiterverarbeitung in Kapitel 4 vorzubereiten.

Abschnitt 3.1 gibt einen Einblick in das Untersuchungsgebiet und dessen Eigenheit. Außerdem wird erklärt, warum genau dort das Projekt gemacht wird. Danach werden die verwendeten Geräte mit ihren Spezifikationen beschrieben (siehe 3.2) und in Abschnitt 3.3 deren Verwendung aufgezeigt. Ebenso wird erklärt, welche Funktion diese während des Messablaufs hatten. Um für die Weiterverarbeitung geeignete Daten zu erhalten, müssen die Informationen aller Geräte kombiniert werden, was in Abschnitt 3.4 erläutert wird. Abschließend wird noch kurz auf die manuelle Messung der Brusthöhendurchmesser (engl: *Diameter at Breast Height (DBH)*) eingegangen.

3.1 Untersuchungsgebiet

Grundsätzlich hätte für diese Arbeit jedes beliebige Stück Wald gescannt und weiterverarbeitet werden können, in dem vorzugsweise nur wenige Sträucher und niedere Bäume vorhanden sind. Wenn die Baumstämme in den Punktwolken klar zu erkennen sind, erleichtert das die automatische Detektion immens.

Die Daten für diese Arbeit wurden in Vorarlberg in der Ortschaft Düns (siehe Abbildung 3.1) aufgenommen, da dort zeitgleich ein anderes Projekt ([BioSLIDE Team, 2017]) Aufnahmen gemacht hat und somit Referenzdaten zur Verfügung stehen. Für dieses Projekt wurde ein Gebiet benötigt, in dem nachweislich Erdbeben stattfinden. Dies bedeutet aber auch, dass das Areal relativ steil ist, was die Datenaufnahme und

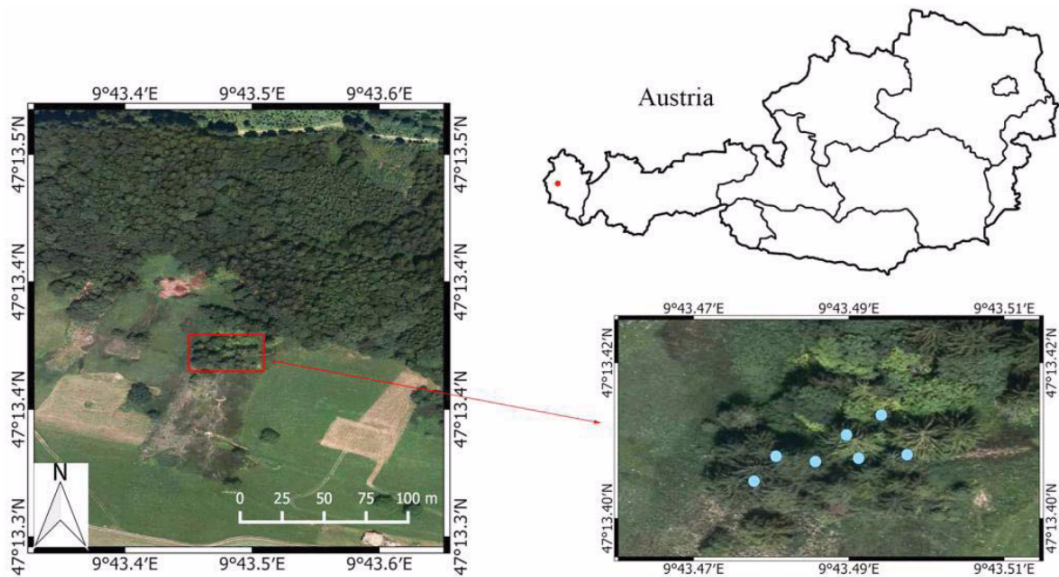


Abbildung 3.1: Lage des Messgebiets in Düns, Vorarlberg. Die türkisen Punkte stellen die Standpunkte des TLS dar. [Wang u. a., 2016]

weitere Verarbeitung etwas erschwert. Dem Geländemodell, welches in Abbildung 4.2 dargestellt ist, kann ein Gefälle von ca. 45% von NO nach SW und einer mittleren Seehöhe von 860 m entnommen werden. Die violetten Punkte stellen die Positionen der einzelnen Bäume dar und sind auf einer Fläche von knapp 600 m² verteilt. Die vorherrschende Baumart in diesem Bereich des Nadelwaldes ist die Fichte mit vereinzelt Tannen und Buchen.

3.2 Geräte

Da die Daten dieser Messkampagne nicht nur für diese Arbeit benötigt werden, ist es notwendig, die Daten in ein einheitliches Koordinatensystem zu transformieren, um die unterschiedlichen Messungen danach kombinieren zu können. Deshalb werden zusätzlich zu den Laserscanneraufnahmen noch terrestrische und satellitengestützte Messsysteme verwendet. Die Anwendung dieser wird in Abschnitt 3.3 im Detail beschrieben.

Für die Laserscannermessungen wurde ein Gerät der Firma *Riegl* vom Typ *VZ-2000* verwendet. Um die TLS Punkte in das österreichische Landeskoordinatensystem zu bekommen, waren noch Messungen von Fixpunkten und den Referenzfolien notwendig. Dies geschah mit zwei Geräten der Firma *Leica Geosystems* mit Hauptsitz in der Schweiz. Genauer wurde hier das *System 1200+* verwendet, welches für diese Anfor-

derungen mehr als ausreichend ist. In Tabelle 3.1 sind die wichtigsten Spezifikationen der Geräte angegeben.

Tabelle 3.1: Spezifikationen der verwendeten Geräte

Riegl VZ-2000 [RIEGL Laser Measurement Systems GmbH, 2017]	
Genauigkeit	8 mm (bei 150 m)
Auflösung Hz/V	0.0024°/0.0015°
Scanfenster Hz/v	360°/−40° bis 60°
Scanrate	396 000 Messungen/s
Reichweite	2.5 m bis 2050 m (bei 50 Hz)
Leica TPS1200+ [Leica Geosystems AG, 2017]	
Genauigkeit Hz/V (Std. Abw.)	0.1" (0.3 mgon)
Genauigkeit Distanz	1 mm + 1.5 ppm
Zentriergenauigkeit (Std. Abw.)	0.5" (0.2 mgon)
Leica GPS1200+ [Leica Geosystems AG, 2017]	
Genauigkeit Lage	10 mm + 1 ppm
Genauigkeit Höhe	20 mm + 1 ppm
Anmerkung	Genauigkeitsangaben sind für günstige Bedingungen (Satellitengeometrie und Anzahl, Beobachtungsdauer, Atmosphäre, ...)
Satellitensysteme	GPS und GLONASS

3.3 Messablauf

Nachdem nun die verwendeten Geräte erklärt sind, wird im Folgenden der genaue Messablauf beschrieben. Es werden die unterschiedlichen Anwendungsgebiete der einzelnen Instrumente, sowie die Verwendung der aufgenommenen Daten beschrieben.

- **GNSS-Messung:**

Damit die Laserscannerdaten mit anderen Projekten kombinierbar sind, müssen diese in ein einheitliches Koordinatensystem gebracht werden. Dazu wird das Gauß-Krüger System im Meridianstreifen 28 (EPSG: 31257) verwendet, welches das offizielle Landeskoordinatensystem im Bereich Vorarlberg ist.

Da die satellitengestützten Aufnahmemethoden im Wald nicht funktionieren, werden vier Fixpunkte rund um das Messgebiet vermarktet und mit dem GNSS-



Abbildung 3.2: Lage der Festpunkte durch GNSS-Messung (gelb) und Reflektorfolien zum Referenzieren der Punktwolken (rot)

Gerät aufgenommen. Um nun diese Punkte für die Stationierung der Totalstation verwenden zu können, müssen sie ins richtige Koordinatensystem gebracht werden. Dazu werden Festpunkte vom Bundesamt für Eich- und Vermessungswesen (BEV) rund um das Untersuchungsgebiet ebenfalls mit dem GNSS-Gerät aufgenommen. Danach können anhand der Koordinaten der amtlichen Punkte, welche im Gauß-Krüger System vorliegen, Transformationsparameter berechnet und die vier Fixpunkte rund um das Messgebiet bestimmt werden. Für die Aufnahme wurden Echtzeit Korrekturen von EPOSA verwendet, um die Koordinaten der Fixpunkte in ausreichender Genauigkeit zu bekommen. Diese Punkte wurden nun für die terrestrischen Messungen verwendet, um den Standpunkt der Totalstation im Wald zu bestimmen. In Abbildung 3.2 sind die Punkte, die mit GNSS gemessenen wurden, in *gelb* dargestellt.

- **TPS-Messung:**

Um die Scans in ein einheitliches System zu bringen und zueinander auszurichten, wurden elf Reflektorfolien an den Baumstämmen befestigt. Mit der Totalstation *TPS1200+* wurden diese aufgenommen und gespeichert. Damit diese Targets nun Koordinaten im Gauß-Krüger System bekommen, mussten die vier zuvor gemessenen Fixpunkte ebenfalls tachymetrisch aufgenommen werden. Dann konnten mittels freier Stationierung der Standpunkt der Totalstation berechnet und Transformationsparameter ermittelt werden. Mit diesen Werten wurden an-

schließlich die Koordinaten der Reflektorfolien berechnet und in ein einheitliches System gebracht. In Tabelle 3.2 ist die Standardabweichung der Stationierung aufgelistet, welche mit dem Programm *Geosi* berechnet wurde. Das Ergebnis der TPS Messung sind die roten Punkte in Abbildung 3.2, welche die Position der Reflektorfolien an den Bäumen darstellen.

Tabelle 3.2: Genauigkeit der Registrierung

Standardabweichung	
Orientierung	0.0224 gon
Höhe	8.5 mm

- **TLS-Messung:**

Mit dem Laserscanner werden nun die einzelnen Punktwolken aufgenommen. Es muss darauf geachtet werden, dass in jedem Scan genügend Reflektorfolien zu sehen sind, um später eine erfolgreiche Registrierung durchzuführen zu können. Theoretisch sind drei Identpunkte ausreichend, was jedoch in der Praxis nicht ratsam ist, da dadurch Ausreißer nur schwer zu detektieren sind. Auch ist es sinnvoll, die einzelnen Bäume von möglichst allen Seite aufzunehmen, um die Scanschatten zu minimieren und die weitere Verarbeitung der Daten zu vereinfachen. Im Zuge dieser Arbeit wurden mehr als 150 000 000 von sieben verschiedenen Scanpositionen erfasst. Abbildung 3.1 zeigt die Standorte des Laserscanners durch die türkisen Punkte.

Nach der Aufnahme werden die sieben Punktwolken mittels *RiSCAN PRO*, ein Programm vom TLS Hersteller *Riegl*, anhand der Folien und den künstlichen Targets (Styroporkugeln) zueinander ausgerichtet. Tabelle 3.3 zeigt die erreichten Genauigkeiten, welche später durch den ICP Algorithmus noch verbessert werden sollen.

Tabelle 3.3: Genauigkeit der Orientierung [Wang u. a., 2016]

Scan Name	082512	084146	080033	081609	074902	075500	074451
Verk. Pkt.	4	4	4	5	5	6	5
Std [mm]	4.5	5.4	7.2	8.1	12.4	7.8	4.8

3.4 Kombination der Messdaten

Nachdem die Daten mit den einzelnen Geräten erfasst sind, geht es darum, diese Informationen zu kombinieren und eine *verwendbare* Punktwolke zu erstellen. Darunter versteht man die erfolgreiche Registrierung der Scans zueinander mit kartesischen Koordinaten der Datenpunkte im Gauß-Krüger M28 Koordinatensystem.

Dazu werden die Reflektorfolien sowie die, mit GNSS gemessenen Fixpunkte mit der Totalstation aufgenommen und die Daten gespeichert. Im Postprocessing wird dann mit dem Programm *Geosi* mittels freier Stationierung der Standpunkt des TPS bestimmt. Die Genauigkeit dieser Stationierung ist in Tabelle 3.2 zu finden. Die Orientierungsgenauigkeit von 0.0224 gon entspricht einer Lagegenauigkeit von ungefähr 10 mm im Bereich der Messungen. Dies ist auf jeden Fall ausreichend für diesen Anwendungsbereich.

Ausgehend von dieser Stationierung werden ebenfalls mit *Geosi* die Koordinaten der Reflektorfolien bestimmt. Die Genauigkeit dieser Punkte ergibt sich aus der Stationsgenauigkeit und der Messgenauigkeit des Geräts, wo sich ebenfalls ein Wert von ca. 10 mm ergibt.

Da die Fixpunkte satellitengestützt aufgenommen wurden und somit deren Genauigkeit schlechter ist, ist die Positionsgenauigkeit der Reflektorfolien im Bezug auf das Landeskoordinatensystem in der Größenordnung von 2 cm bis 5 cm. Höhenmäßig sind eher die schlechteren Werte anzunehmen. Der Grund für diese eher höheren Werte ist einerseits die Abschattung durch das steile Gelände, sowie die Nähe zu den Bäumen, die den Empfang der Satellitensignale stören. Die Geometrie der Koordinaten der Folien ist jedoch in der Genauigkeit des Tachymeters, da alle diese Punkte von einem Standpunkt und somit mit der gleichen Stationierung aufgenommen wurden.

Mittels den Reflektorfolien wurden nun mit dem Programm *RiSCAN PRO* die einzelnen Punktwolken zueinander registriert und georeferenziert. Dazu werden in jedem Scan die gemessenen Folien (diese haben durch ihre hohe Reflektanz einen sehr hohen Intensitätswert) gesucht und diesen die zugehörigen Koordinaten zugewiesen. Danach führt das Programm die entsprechenden Schritte durch und verarbeitet die Messdaten. Dies geschah jedoch im Zug eines anderen Projekts ([BioSLIDE Team, 2017]) und wurde in dieser Arbeit nicht weiter behandelt.

Zur weiteren Verarbeitung der Messdaten in der vorliegenden Arbeit werden die Scans mit Gauß-Krüger Koordinaten der Datenpunkte verwendet. Diese sollen nun mit den in Kapitel 4 beschriebenen Methoden ausgedünnt und besser zueinander registriert werden.



Abbildung 3.3: Position der Bäume mit händisch gemessenem DBH

3.5 Referenzwerte der Stämme (DBH)

Um Referenzwerte zur Überprüfung der Ergebnisse zu haben, wurden im Zuge der Feldmessung die Durchmesser der Stämme noch händisch mit einem Maßband erhoben. Dies geschah in einer Höhe von 1.3m über dem Boden, da dies bei der Forstinventur ein gängiger Wert ist und wird Brusthöhendurchmesser genannt. In der Literatur wird oft auch der englische Begriff *Diameter at Breast Height (DBH)* verwendet. Im Bereich des Messgebiets wurden insgesamt 27 Bäume mit dem Maßband gemessen, bei denen der durchschnittliche Durchmesser bei ca. 32 cm mit einer Standardabweichung von ca. 14 cm liegt.

In Abbildung 3.3 und Tabelle 3.4 sind die geografischen Positionen mit zugehörigem Stammdurchmesser dargestellt.

Tabelle 3.4: DBH der gemessenen Bäumen mit statistischen Kennwerten

1	51.7 cm	10	33.4 cm	19	16.2 cm
2	55.9 cm	11	58.7 cm	20	36.2 cm
3	38.1 cm	12	31.2 cm	21	19.0 cm
4	42.0 cm	13	37.9 cm	22	30.5 cm
5	36.3 cm	14	41.6 cm	23	9.7 cm
6	42.6 cm	15	14.7 cm	24	9.9 cm
7	23.3 cm	16	23.1 cm	25	58.1 cm
8	25.2 cm	17	21.4 cm	26	36.7 cm
9	34.5 cm	18	20.5 cm	27	17.4 cm
Durchschnitt				32.1 cm	
Standardabweichung				14.1 cm	

Die Mittelpunkte der Stämme wurden manuell aus der Punktwolke in einer Höhe von 1.3 m herausgesucht. Die Nummerierung erfolgte ebenfalls händisch aus den Scannerdaten, indem die bei der Feldmessung angebrachten Nummern, den jeweiligen Bäumen zugeordnet wurden. Abbildung 3.4 zeigt solche Zettel mit der jeweiligen Nummerierung.



Abbildung 3.4: Zettel mit Nummern zur späteren Bestimmung der Stämme [vgl. Wang u. a., 2016]

Kapitel 4

Methodik

Im Zuge dieser Diplomarbeit werden verschiedene Methoden entwickelt, um die Daten des Laserscanners zu verarbeiten. Das Hauptaugenmerk richtet sich dabei auf die Referenzierung der einzelnen Punktwolken von den jeweiligen Standpunkten des TLS zueinander. Hier wird vor allem mit dem, von OPALS zur Verfügung gestellten, Modul ICP gearbeitet. Aber auch einige andere Prozessierungsschritte müssen im Vorfeld durchgeführt werden, um diese entsprechend aufzubereiten. Ein großes Problem im Umgang mit Laserscannerdaten ist die große Menge an Einzelpunkten, die die Verarbeitung erschwert. Es wäre natürlich möglich, den kompletten Datensatz mittels ICP Algorithmus zueinander zu referenzieren, was aber einen erheblichen Rechenaufwand und somit Zeitaufwand bedeutet. Daher wird in dieser Arbeit auch darauf geachtet, die Punktwolken vorher soweit auszudünnen, dass die Berechnungen weniger lang dauern und trotzdem noch gleichwertige Ergebnisse erzielt werden. Hierbei ist der Ansatz, nur die Baumstämme in verschiedenen Höhenschichten für die Registrierung zu verwenden.

In diesem Kapitel werden die angewandten Methoden beschrieben und die Idee dahinter genauer erklärt. Um eine Vorstellung zu bekommen, wie die Verarbeitung der Daten vonstatten geht, wird hier kurz der Ablauf der einzelnen Schritte erklärt.

Abschnitt 4.1 beschreibt die Vorbereitung der Laserscannerdaten und die automatische Detektion von Baumstämmen. In Punkt 4.2 werden die sichtbaren Baumanteile pro Scanposition aus der gesamten Punktwolke herausgeschnitten. Abschließend wird in Abschnitt 4.4 noch der ICP Algorithmus und dessen Anwendung behandelt.

Abbildung 4.1 zeigt grafisch den Ablauf der Datenverarbeitung.

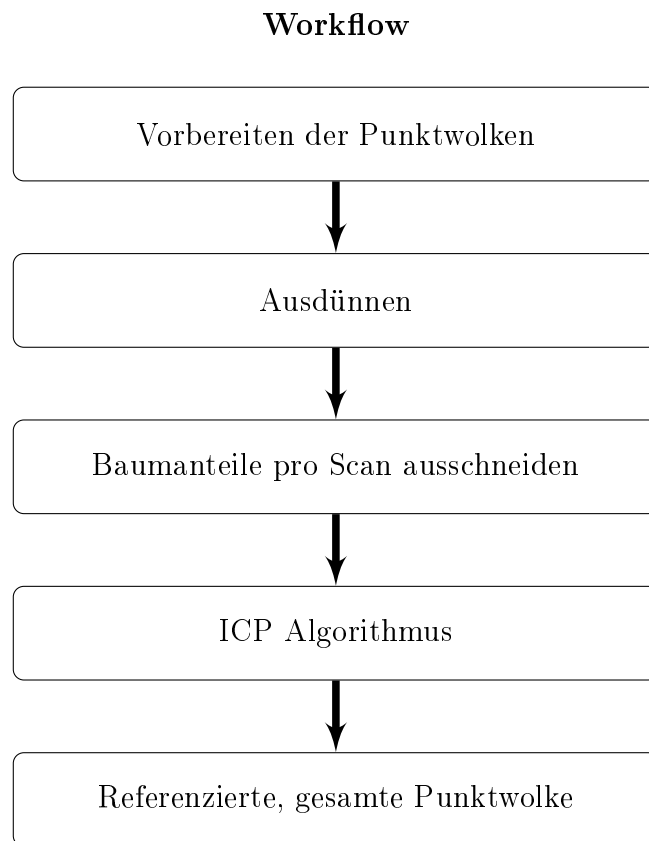


Abbildung 4.1: Workflow zum Verarbeiten der TLS Daten

4.1 Vorbereitung

Ziel der Vorbereitung ist es, die Punktwolken von unpassenden Punkten zu befreien und Koordinaten von einigen einzelnen Bäumen automatisch zu ermitteln. Mit diesen Informationen können die Datenpunkte extrem reduziert werden, was sich hoffentlich positiv auf die Laufzeit der Prozessierung auswirkt. Mit den Koordinaten der Bäume werden die Baumstämme aus der Punktwolke extrahiert, da diese von äußeren Einflüssen, wie zum Beispiel dem Wind, am wenigsten betroffen sind und somit die stabilsten Bereiche darstellen. Mit den verbleibenden Punkten wird dann der ICP Algorithmus in verschiedenen Höhenschichten angewendet und verglichen. Damit diese Reduzierung erreicht werden kann, muss der Datensatz um verschiedenste Attribute erweitert werden, was mit dem OPALS Modul *AddInfo* und *Normals* möglich ist.

Folgende Schritte werden durchgeführt, um die Daten für die weitere Prozessierung vorzubereiten:

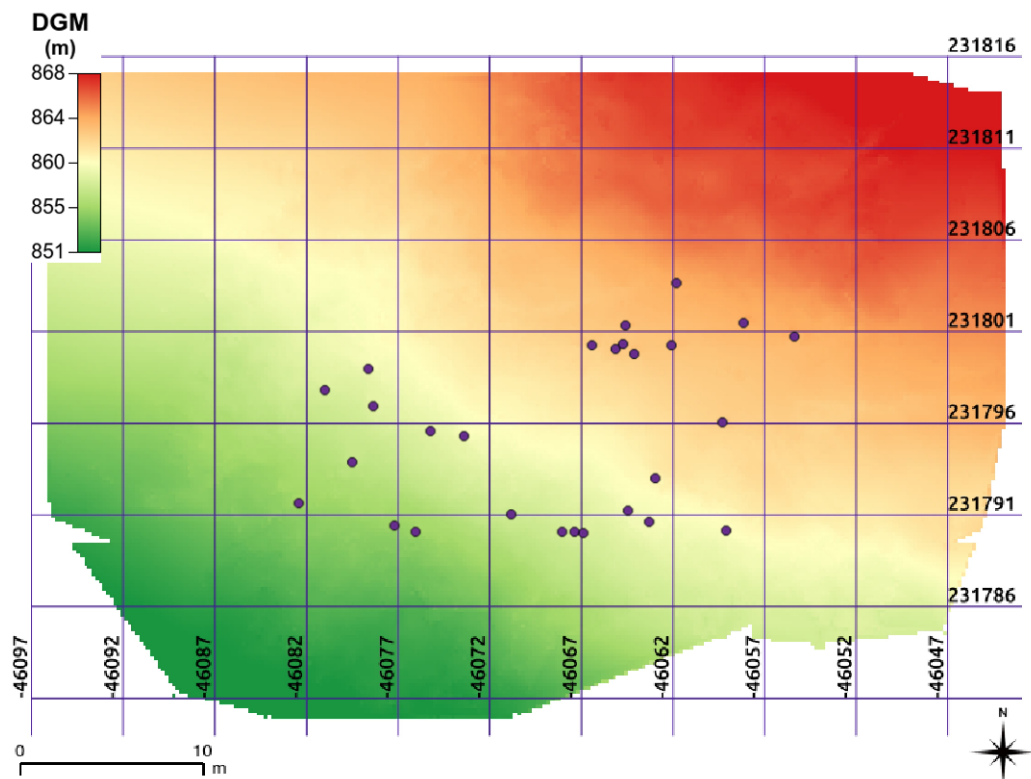


Abbildung 4.2: Geländemodell vom Auswertebereich mit Baumpositionen

4.1.1 Höhe über DGM

Von großem Interesse bei dieser Arbeit ist die Punkthöhe über dem digitalen Geländemodell (DGM). Mit dieser Information können verschiedene Schichten aus dem Datensatz herausgeschnitten und unterschiedliche Ergebnisse produziert werden.

Abbildung 4.2 zeigt das Geländemodell vom Bereich der TLS Messungen. Anhand der Farbcodierung kann man sehen, dass das Gebiet von SW nach NO um ungefähr 20 Meter ansteigt, was die Verarbeitung etwas erschwert.

Um jedem Datenpunkt die Höhe über dem DGM hinzuzufügen, wird das vorhin erwähnte Modul *addInfo* verwendet.

```

1  opalsAddInfo
2      -inf scan.odm
3      -gridFile DGM.tif
4      -attribute "NormalizedZ=z-r[0]"

```

Code 4.1: addInfo NormalizedZ

Wie in Code 4.1 zu sehen ist, wird das Modul mittels verschiedener Parameter gesteuert. Zur Berechnung des Attributes *NormalizedZ* wird nur eine sehr einfache Formel angegeben, welche nachfolgend kurz erklärt ist.

- **-inf:** Hier wird die Datei angegeben, die später das neue Attribut enthalten soll. Außerdem verwendet das Modul Werte aus dem hier angegeben Datensatz. In diesem Fall werden die X-, Y- und Z-Koordinaten genutzt, um die Höhe über dem Gelände zu erhalten.
- **-gridFile:** Dieser Parameter beinhaltet das zu Grunde liegende Geländemodell von dem Bereich, wo die Messungen durchgeführt worden sind. Es handelt sich hier um ein so genanntes *GeoTIFF* Datenformat, welches zusätzlich zu den Pixelwerten eine Georeferenzierung speichern kann. Das macht man sich bei dieser Anwendung zunutze, um zu jedem aufgenommenen Punkt einen Wert für die Geländehöhe zu bekommen. Generiert wurde dieses Modell an der TU Wien am Institut für Photogrammetrie und Fernerkundung mit Hilfe von OPALS. Dabei kam ein hierarchischer Ansatz nach [Pfeifer und Mandlbürger, 2008] zur Verwendung und das Ergebnis ist dieses Geländemodell mit einer Auflösung von 20 cm.
- **-attribute:** Die eigentliche Berechnung erfolgt dann mit der angegebenen Formel. Hier wird zu jedem X-Y-Paar eines Messpunktes aus dem Geländemodell eine entsprechende Höheninformation gesucht und danach die Differenz zum Z-Wert in dem *NormalizedZ*-Attribut gespeichert.

4.1.2 Normalvektoren

Eine sehr wichtige Information der Punktwolke sind die lokalen Normalvektoren der einzelnen Punkte. Diese werden später zum Ausdünnen und vor allem beim ICP-Algorithmus zum Ermitteln der korrespondierenden Punkte bei der Punkt-zu-Ebene Methode benötigt.

In diesem Fall wird für jeden Punkt und einer gewissen Anzahl von Nachbarpunkten eine Ebene abgeschätzt und die Flächennormale verwendet, um die Komponenten des Normalvektors zu bestimmen. Somit besitzt jeder Datenpunkt drei einzelne Werte für die Richtungsanteile des Vektors in den verschiedenen Achsrichtungen (nX, nY, nZ). Zusätzlich wird noch ein Kennwert für die Genauigkeit der Ebene, die Standardabweichung der Residuen, *Sigma0* berechnet. Anhand dieses Wertes können sogenannte

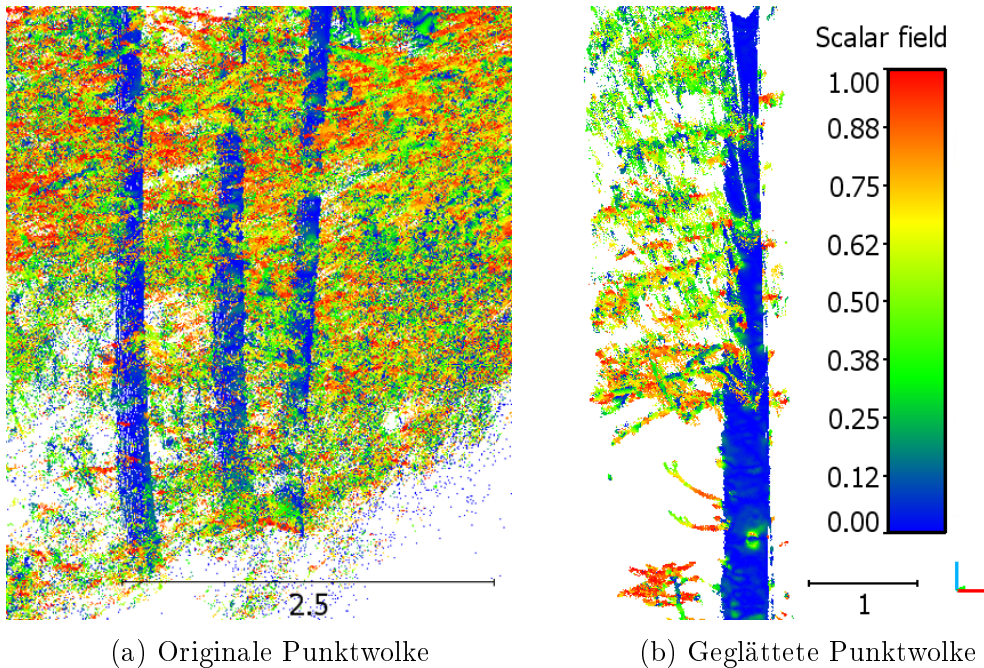


Abbildung 4.3: Farbige Darstellung der Z-Komponente nZ des Normalvektors. Deutlich zu erkennen sind die Baumstämme in Blau.

raue Punkte entfernt werden. Bei dieser Arbeit wird dieser Grenzwert bei zwei Zentimetern gesetzt. Dadurch fällt schon ein Großteil der unwesentlichen Punkte aus dem Datensatz heraus. Abbildung 4.3 zeigt recht deutlich den Unterschied vor und nach der Glättung. Vor allem Nadeln/Blätter, dünne Zweige und bodennahe, kleinere Gewächse fallen durch diese Filterung aus der Punktwolke.

In Abbildung 4.3 sind zwei Ausschnitte der Datenpunktwolke zu sehen, die anhand der nZ -Komponente eingefärbt sind. Recht deutlich zu sehen sind hier die Baumstämme mit einem sehr kleinen Wert, was eine vertikale Ebene bedeutet. In Abbildung 4.3(a) ist ein kompletter Scan mit allen Punkten dargestellt. 4.3(b) zeigt einen Teil eines einzelnen Baumes, der aus einem Scan extrahiert wurde und die *rauen* Punkte entfernt sind.

Zum Berechnen der Normalvektoren für die Datenpunkte wird Code 4.2 verwendet. Die Parameter *neighb*, *searchMode*, *selMode* und *searchRadius* definieren die Nachbarschaftsbeziehungen der verwendeten Punkte. In diesem Fall werden maximal 999 nächste Punkte im Umkreis von fünf Zentimetern im 3D-Raum (x, y, z) für die Berechnung ausgewählt. Diese Werte haben sich bei anderen Projekten mit TLS Messungen bewährt. Durch die hohe Anzahl an Nachbarpunkten und dem kleinen Suchradius werden alle Punkte aus dieser Umgebung für die Bestimmung der Ebene verwendet. Diese wird mit der besten Anpassung durch die Datenpunkte geschätzt und die Flächen-

normale inklusive dem Rauheitsfaktor σ_0 gespeichert. Der Parameter *normalsAlg* gibt die Methoden an, mit der die Normalen geschätzt werden. Hier wird zum Berechnen ein Ebenen Ausgleich nach der Methode kleinster Quadrate verwendet, da dieser die kürzeste Laufzeit besitzt.

```

1      opalsNormals
2      -inf scan.odm
3      -neighb 999
4      -searchMode d3
5      -selMode nearest
6      -searchRadius 0.05
7      -normalsAlg simplePlane

```

Code 4.2: Normalvektoren berechnen

4.1.3 Lokale Maxima detektieren

In diesem Schritt geht es darum, die Positionen von potentiellen Baumstämmen automatisch zu detektieren. Es ist nicht notwendig, jeden einzelnen Baum zu finden, sondern es reicht, wenn einige aus der Punktwolke gefunden werden können. Hier ist es jedoch wichtig, dass die Koordinaten relativ gut (ungefähr $\pm 75\text{cm}$) zu den tatsächlichen Baummittelpunkten passen. Sonst würde im nächsten Schritt ein Bereich ohne Baumstamm ausgeschnitten werden.

Um dies zu erreichen, werden diesmal mehrere OPALS Module nacheinander verwendet, was in Code 4.3 zu sehen ist. Begonnen wird mit *opalsCell*, um zwei Rasterbilder zu erstellen. Eines davon ist eine Punktdichtekarte (*pdens*), das andere wird mit dem Feature *center* generiert und beinhaltet den nächsten Punkt zum Zentrum der Zellgröße. Dies wird benötigt, um die Information der *Z*-Koordinate der Punkte nicht zu verlieren, da im Rasterbild immer nur ein Wert zu einem *X-Y*-Paar gespeichert werden kann. Abbildung 4.5(a) und 4.5(b) zeigen die zwei erstellten Bilder. Um einen besseren Eindruck dieser Schritte zu erhalten, ist in Abbildung 4.4(a) die Punktdichtekarte zu sehen. Die helleren Stellen deuten auf eine höhere Dichte und somit auf potentielle Positionen von Baumstämmen hin.

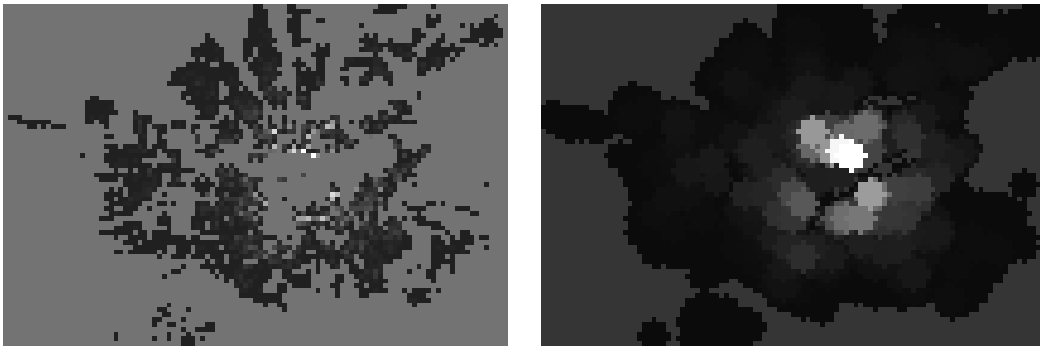
(a) OpalsCell - Feature *pdens*(b) OpalsStatFilter - Feature *max*

Abbildung 4.4: Punktdichtekarte (a) der Scans und zugehöriger statistischer Analyse (b) zum Detektieren von Bäumen. (Pixelgröße: 1.3)

```

1  opalsCell
2    -inf scan.odm
3    -outf cellImage.tif
4    -feature pdens center
5    -cel 1.3
6
7  opalsStatFilter
8    -inf cellImage_pdens.tif
9    -outf cellImage_stat.tif
10   -feature max
11   -kernelSize 5
12   -kernelShape circle
13
14  opalsAlgebra
15    -inf cellImage_pdens.tif cellImage_stat.tif
16    -outf cellImage_localMax.tif
17    -formula "r[0]==r[1]"
18
19  opalsAlgebra
20    -inf cellImage_center.tif cellImage_localMax.tif
21    -outf cellImage_final.tif
22    -formula "r[0]*r[1]"

```

Code 4.3: Lokale Maxima finden

Zu erwähnen ist, dass in Abbildung 4.5 die grünen Zellen das Maximum und die roten das Minimum darstellen. Der Farbverlauf zwischen diesen beiden hängt mit der Höhen- bzw Punktdichteinformation zusammen. Alle weißen Bereiche sind sogenannte *Not a Number (NaN)* Werte, also Zellen ohne Inhalt.

Nachfolgend wird das Modul *StatFilter* angewandt, um das vorher erstellte Punktdichtebild 4.5(a) zu analysieren. Es wird ein kreisförmiger Kernel mit einem Durchmesser

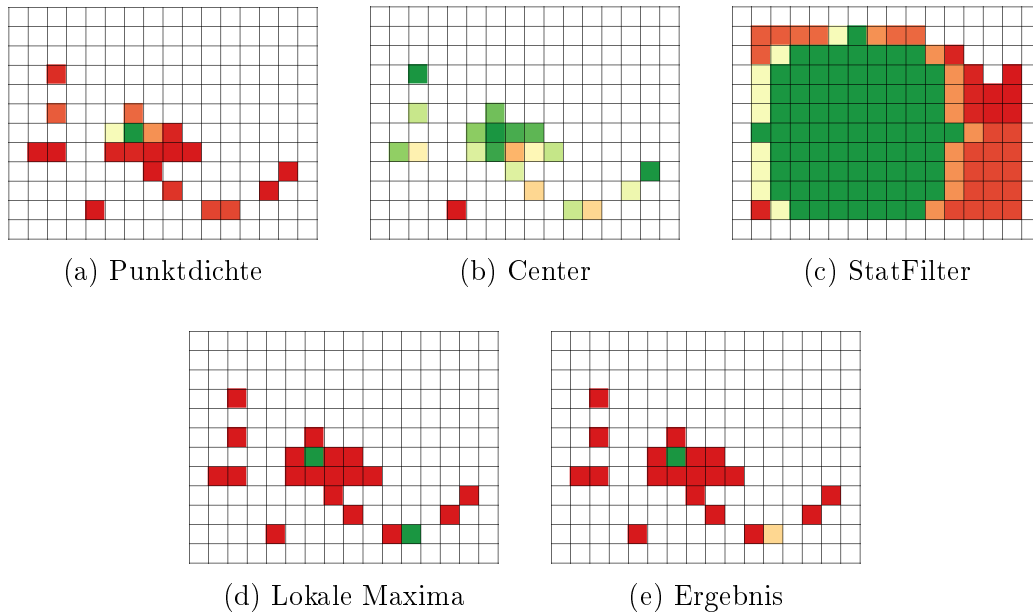


Abbildung 4.5: Notwendige Zwischenschritte zur automatischen Baumstammdetektion (Pixelgröße: 0.5)

von fünf Bildpunkten über das Bild “geschoben“ und jeweils der maximalen Wert unter der Fläche in die Zelle im Zentrum des Kreises geschrieben. Das entstandene Bild ist in Abbildung 4.5(c) dargestellt. In Grafik 4.4(b) ist der kreisförmige, fünf Pixel Kernel zu sehen, der die maximale Punktdichte in dem Bereich ermittelt.

Nun werden Punktdichte 4.5(a) und das Ergebnis des statistisch ausgewerteten Bild 4.5(c) mittels Modul *Algebra* miteinander kombiniert. Es wird verglichen, ob die Pixel an der gleichen Position in beiden Bildern denselben Werte haben. Ist dies der Fall, wird eine 1 gespeichert, andernfalls eine 0. Besitzt eine der beiden Zellen einen NaN-Eintrag, wird dieser ins Ergebnis übertragen. Das entstandene Bild 4.5(d) beinhaltet nun die lokalen Maxima der Ausgangspunktwolke.

Abschließend werden ebenfalls mit dem Modul *Algebra* die vorher im *Zentrums*-Bild 4.5(b) gespeicherten *Z*-Koordinaten ins Ergebnis übertragen. Dazu werden die Pixelwerte mit den lokalen Maxima 4.5(d) multipliziert und so entsteht ein Rasterbild (Abbildung 4.5(e) mit den Koordinaten der Stammpositionen und zugehöriger Höheninformation.

In Tabelle 4.1 ist ein Soll-Ist Vergleich der gefundenen und manuell gesuchten (siehe Referenzdaten in Abschnitt 3.5) Koordinaten der Baumstammmittelpunkte. Die Werte stammen aus dem GeoTIFF-Bild 4.5(e) und repräsentieren die grüne bzw. orange Zelle. Das Ergebnis dieser Verarbeitungsschritte ist eine Liste von 3D Koordinaten (X , Y , Z), die zur weiteren Prozessierung benötigt wird.

Tabelle 4.1: Soll/Ist Vergleich der Baumkoordinaten

Baum 1	X [m]	Y [m]	Baum 2	X [m]	Y [m]
Soll	- 46073.328	231795.252	Soll	- 46066.622	231798.804
Ist	- 46073.304	231795.190	Ist	- 46066.740	231789.940
Δ	0.024	0.062	Δ	- 0.118	- 0.136

4.2 Bäume extrahieren

Nachdem nun einige Baumstämme detektiert sind, werden diese Anteile aus der Punktwolke jedes Scans herausgeschnitten. Code 4.4 generiert somit aus jeder Punktwolke neue Dateien, die nur noch die Punkte enthalten, die im Bereich eines Baumes sind. Die Grenzwerte für die Region werden von den Stamm-Koordinaten $\pm 0.75m$ berechnet. In Abbildung 4.6 ist ein solcher ausgeschnittener Baum zu sehen. Im Grundriss ist gut ersichtlich, dass der Laserscanner auf der linken Seite gestanden ist, da der Scanschatten auf der gegenüberliegenden Seite zu erkennen ist. Im Aufriss 4.6(b) ist erkennbar, dass nur der halbe Baum erfasst werden konnte.

```

1  opalsExport
2  -inf scan#.odm
3  -outf scan#_t#.odm
4  -filter "Region [minX minY maxX maxY]"
5
6  opalsExport
7  -inf scan#_t#.odm
8  -outf scan#_trees.odm

```

Code 4.4: Bäume aus der Punktwolke extrahieren

Der zweite *Export* Aufruf vereinigt alle einzelnen Baumanteile pro Scanposition in eine eigene Datei. Somit entsteht eine Punktwolke, die alle sichtbaren Teile der Bäume von einem Standpunkt enthält. Diese wird später beim ICP Algorithmus verwendet. Abbildung 4.7 zeigt Draufsicht und Aufriss der so erstellten Punktwolke. In 4.7(a) ist auch gut erkennbar, dass der Grundriss der ausgeschnittenen Bereiche ein Quadrat mit der Seitenlänge von 1.5 Metern ist. Die Einfärbungen in den Grafiken stellen die Höhe der Punkte über dem Geländemodell dar.

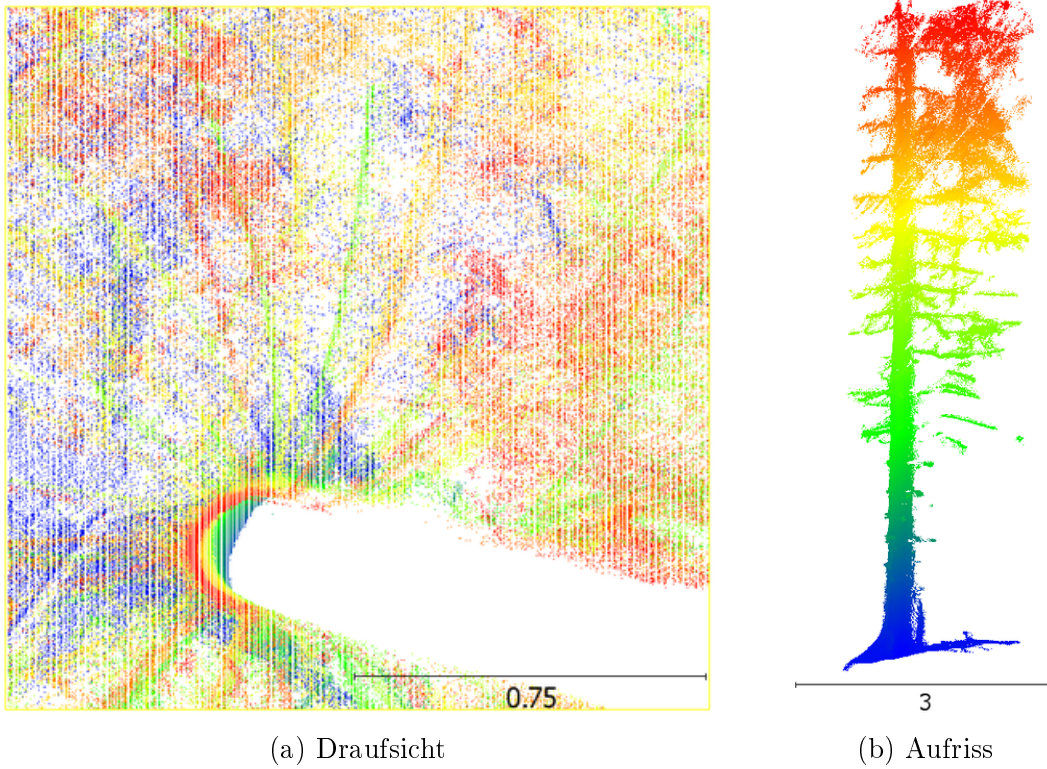


Abbildung 4.6: Aus einem Scan detektierter Baum in der Draufsicht (a) und im Aufriss (b)

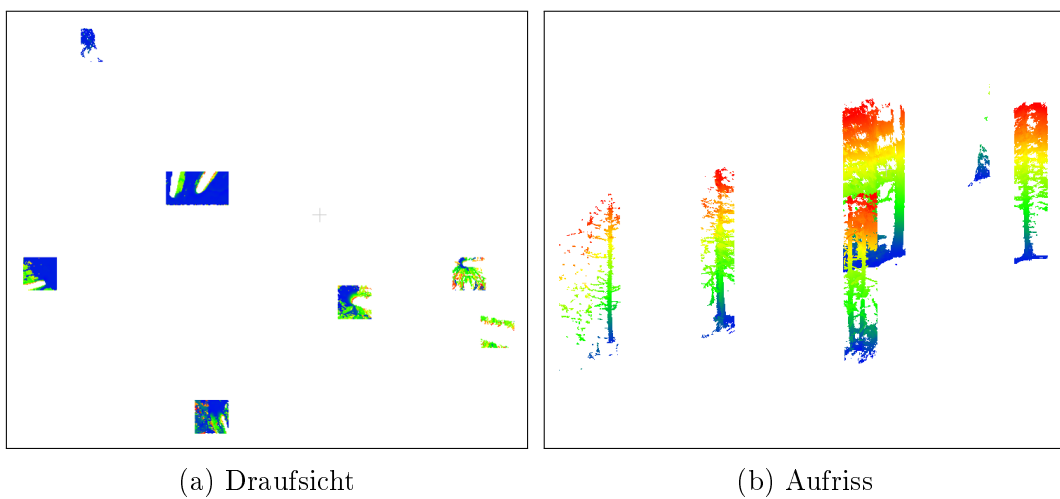


Abbildung 4.7: Kombinierte Baumanteile pro Scan in der Draufsicht (a) und im Aufriss (b)

Tabelle 4.2: Verwendete Schichten mit zugehörigem OPALS Filter

Höhenbereich	Abb 4.8	Filter Syntax
Komplett	(a)	[]
Ohne 1-7.5 m	(b)	[NormalizedZ > 7.5 AND NormalizedZ < 1]
Höher als 6 m	(c)	[NormalizedZ > 6]
Kleiner als 2.5 m	(d)	[NormalizedZ < 2.5]
Nur 2-6 m	(e)	[NormalizedZ > 2 AND NormalizedZ < 6]
Ohne 2-6 m	(f)	[NormalizedZ > 6 AND NormalizedZ < 2]

4.3 Schichten aus Punktwolken schneiden

Da es in dieser Arbeit darum geht, den ICP-Algorithmus mit verschiedenen Schichten der Punktwolke zu testen, werden nun Bereiche aus den Scans geschnitten. Ziel ist es, einen Vergleich der Ergebnisse anzustellen und herauszufinden, welche Höhenschichten am besten für die Berechnung der Orientierung geeignet sind. Diese Schichten richten sich nach der Höhe über dem digitalen Geländemodell (DGM) und sind in Abbildung 4.8 grafisch dargestellt. Die grünen Bereiche symbolisieren die verwendeten Bereiche aus den Scans. Rot wird zur Bestimmung der Orientierung ignoriert.

In Tabelle 4.2 ist zu den jeweiligen Bereichen der zugehörige Filter für das OPALS Modul *Export* (siehe Code 4.5) angeführt.

```

1  opalsExport
2  -inf scan#_trees.odm
3  -outf scan#_trees_2-6.odm
4  -filter "Generic [NormalizedZ > 2 AND NormalizedZ < 6]"

```

Code 4.5: Schicht aus der Punktwolke schneiden

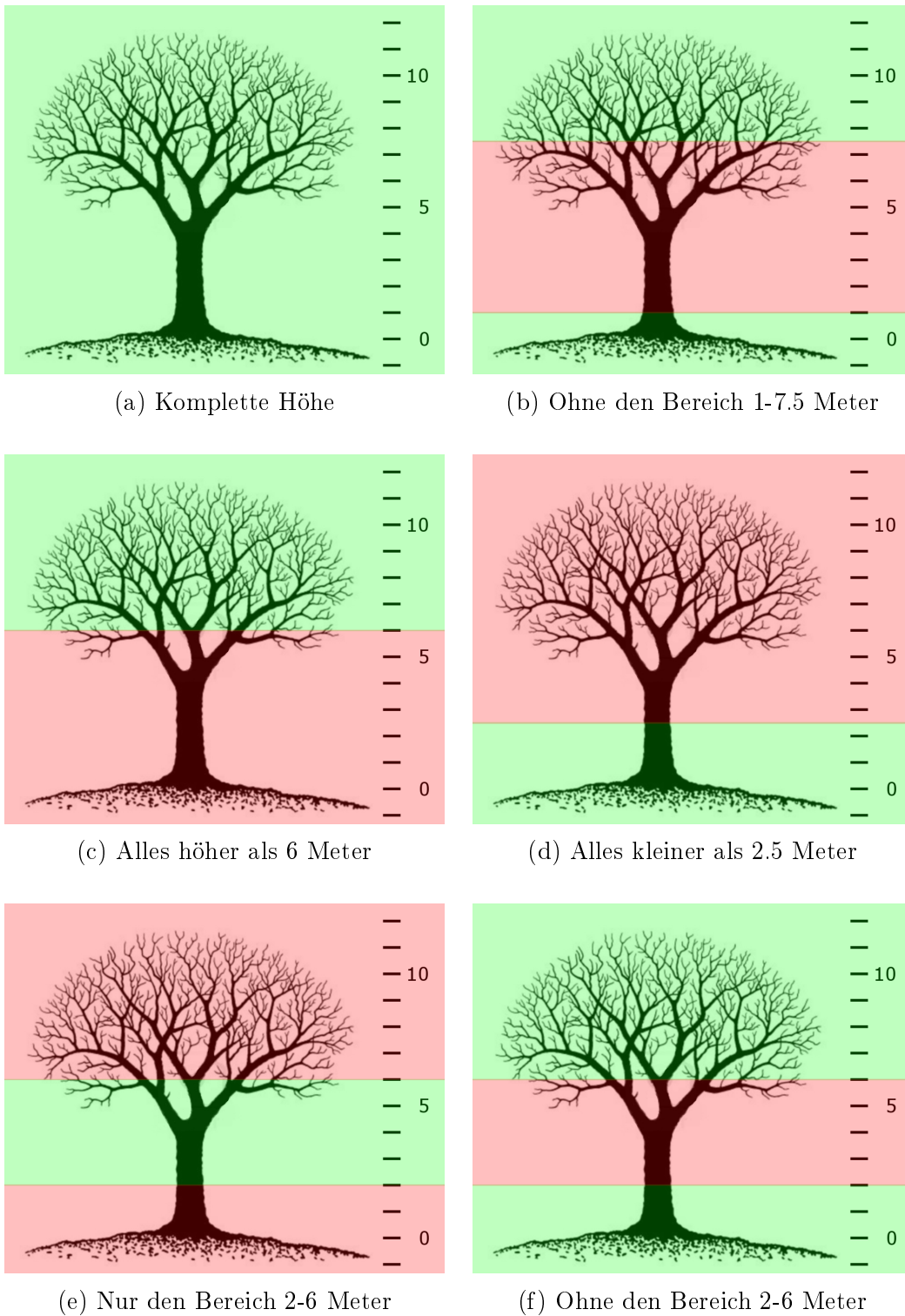


Abbildung 4.8: Verwendete (grün) Punktwolkenanteile [Baum: clipart-work.net, 2017]

Auf die Punktwolke angewendet ergibt sich dann Abbildung 4.9, in der gut der leere Streifen im Bereich von 2-6 Metern zu sehen ist. Die Einfärbung der Punkte stellt deren Höhe über dem Geländemodell dar.

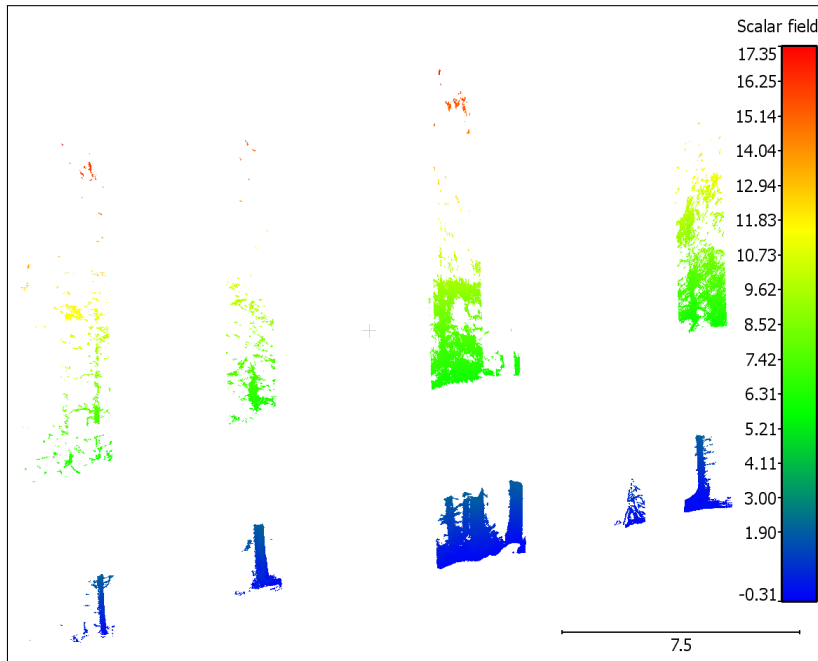


Abbildung 4.9: Scan ohne die Punkte im Bereich von 2-6 Metern im Aufriss

4.4 ICP anwenden

Nachdem die Punktwolken entsprechend vorbereitet sind, können nun die Transformationsparameter für die gemeinsame Orientierung bestimmt werden. Dazu wird das OPALS Modul *ICP* verwendet. Der Aufruf erfolgt ebenfalls über ein Kommandozeilen Skript, welches in Code 4.6 ausschnittsweise angegeben ist.

Als Eingabeparameter werden die zuvor erzeugten Punktwolken verwendet, welche die verschiedenen Höhenschichten enthalten. Das Ergebnis des Iterative Closest Point Moduls wird in dem Ordner *version01* gespeichert. Darin findet man in diesem Fall, zusätzlich zu den von OPALS generierten Log-Files, eine Grafik (engl. *plot*) und die Datei mit den Transformationsparametern *icp_param.xml*.

```

1  opalsICP
2  -inf scan#_trees.odm
3  -outDirectory version01
4  -samplingDist 0.05
5  -trafoType rigid
6  -voxelSize 0.5
7  -searchRadius 0.05
8  -maxSigma 0.01
9  -fixedFile scan1_trees.odm
10 -redPoint -46070 231795 875
11 -plot true
12 -transformData false
13 -outParamFile version1\icp_param.xml

```

Code 4.6: ICP auf die Scans anwenden

In Tabelle 4.3 sind die einzelnen Parameter, die beim Aufruf des Moduls angegeben werden, kurz erklärt. Genauere Informationen findet man auf der Homepage des Projekts [OPALS Team, 2017a].

Tabelle 4.3: Verwendete ICP Parameter mit Beschreibung

Parameter	Beschreibung
-inf	Gibt an, welche Eingabe- ODM Dateien für die Berechnung herangezogen werden.
-outDirectory	In diesem Ordner befinden sich danach die Protokolle von OPALS, die Grafiken (falls gewünscht) und eventuelle transformierte Punktwolken.
-samplingDist	Dieser Parameter gibt den mittleren Abstand zwischen den korrespondierenden Punkten an. Es kann auch in alle Koordinatenachsen ein unterschiedlicher Wert angegeben werden, was aber hier nicht notwendig ist.
-trafoType	Hier gibt es die Möglichkeit zwischen verschiedenen Transformationen auszuwählen. Bei dieser Arbeit wird die starre (engl. <i>rigid</i> , 3 Rotationen und 3 Translationen) Transformation verwendet.

-voxelSize	Mit Hilfe dieses Werts wird die Punktwolke in einer schlechteren Auflösung dargestellt, um überlappende Bereiche zu finden. Angegeben wird die Kantenlänge der Würfel, die die Punktwolke repräsentieren sollen. Es ist vergleichbar mit einem Pixel mit bestimmter Größe, nur im 3D Raum. Sobald sich ein Datenpunkt innerhalb eines solchen Würfels befindet, wird dieser zur Hülle hinzugefügt.
-searchRadius	Dieser Wert gibt den Radius der Kugel an, die die Punkte für die ausgleichende Ebene der Punkt-zu-Ebene Methode beinhaltet. Da es sich hier um TLS Daten handelt, kann dieser Wert eher klein gehalten werden, da die Punktdichte im Vergleich zu ALS sehr hoch ist.
-maxSigma	Hier wird ein Filterkriterium für die Qualität der korrespondierenden Punkte eingeführt. Zu raue Punkte werden als Ausreißer detektiert und nicht verwendet.
-fixedFile	Der Algorithmus benötigt eine Eingangspunktwolke, zu der alle anderen orientiert werden. Im Fall dieser Arbeit wird die Punktwolke mit den meisten Datenpunkten verwendet.
-redPoint	Speziell für große Koordinaten ist es entscheidend, das Zentrum des Koordinatensystems in die Nähe dieser Punkte zu "schieben", da die berechneten Parameter sonst eine sehr hohe Korrelation aufweisen.
-plot	Ist der Schalter, über den dem Benutzer eine grafische Ausgabe der einzelnen Iterationsschritte angezeigt wird. Abbildung 4.10 zeigt eine solche Grafik.
-transformData	Dieser Wert (<i>true/false</i>) gibt an, ob die Ausgangspunktwolken mit den berechneten Parametern gleich transformiert werden sollen.
-outParamFile	Hier wird der Speicherort für die XML-Datei mit den Transformationsparametern angegeben.

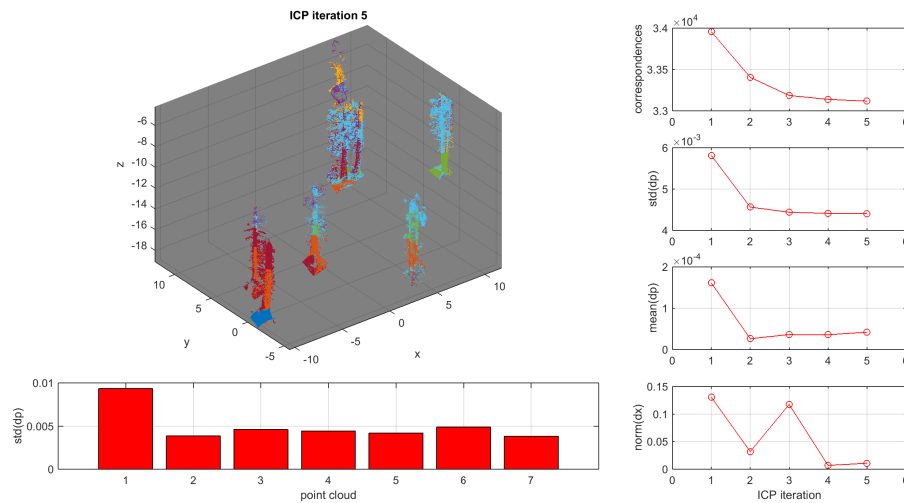


Abbildung 4.10: Beispiel für die grafische Ausgabe der Iterationen mit gesetztem *plot* Parameter.

Um nun abschließend mit den Transformationsparametern die einzelnen Punktwolken zu transformieren, wird Code 4.7 verwendet. Hier wird wieder das OPALS Modul *Export* angewandt, welches die vorher generierte Parameter-Datei nutzt. Die Transformation geschieht über sogenanntes *Parameter Mapping*. Hier werden die Rotationen und Translationen zum Transformieren für die jeweilige Punktwolke aus dem File gelesen und angewendet. Das Ergebnis ist eine Verbesserung der Orientierung durch den ICP Algorithmus.

```

1  opalsExport
2  -inf scan#_trees.odm
3  -outf scan#_trees_icp.odm
4  -inParamFiles version01\icp_param.xml
5  -paramMapping "trafo=opalsICP.outTrafpars me.inFile[0].stem().←
    startswith( Path(you.getIdGridMov()).stem() )"

```

Code 4.7: Transformationsparameter auf Punktwolken anwenden

4.5 Validierung

Als Grundlage für die Bewertung der einzelnen Szenarien werden großteils die Informationen aus den OPALS- Protokollen verwendet. Es wird die Laufzeit der jeweiligen Schritte analysiert und untereinander verglichen. Zusätzlich werden noch die Dauer des ICP- Algorithmus mit den unterschiedlichen Datensätzen angegeben und die Auswir-

kung der vorhergehenden Datenreduktion gezeigt.

Neben der Prozessierungsdauer ist die Standardabweichung und der Mittelwert der korrespondierenden Punktpaare ein Kriterium für eine Verbesserung beziehungsweise Verschlechterung, was ebenfalls in Kapitel 5 diskutiert wird. Dazu werden die Szenarien untereinander verglichen und auch Differenzen zu den Kennwerten aus *RiSCAN PRO* berechnet. Auch hier bilden die Protokolle des OPALS Moduls als Grundlage für die Auswertung.

Um eine bessere Vorstellung dieser Werte zu bekommen und die Auswirkungen der Transformationsparameter auf die Punktwolke zu zeigen, werden unterschiedliche Szenarien grafisch in Form von Schnitten durch einen Baum dargestellt und diskutiert. Zur Erstellung der Abbildungen wird *Matlab* verwendet, wo mit Hilfe von Skripts verschiedenste Parameter getestet und dargestellt werden können.

Kapitel 5

Ergebnisse und Diskussion

Nachdem in Kapitel 4 die Methoden zur Datenverarbeitung beschrieben wurden, werden nun die Resultate bewertet. Genauer wird analysiert, wie sich die verschiedenen ICP Durchgänge auf die Laufzeit auswirken und eine Verbesserung beziehungsweise Verschlechterung der Registrierung im Vergleich zu den originalen Daten aus *RiSCAN PRO* diskutiert.

Als Kennwerte werden die Werte aus den Protokollen der jeweiligen ICP Durchläufen verwendet. Dazu zählen zum Beispiel Standardabweichung, Mittelwert oder korrespondierende Punkte. Zusätzlich werden die Ergebnisse nach der neuen Registrierung in Matlab dargestellt und verschiedene Schichten zum Vergleich ausgeschnitten und besprochen.

Begonnen wird in Abschnitt 5.1 mit einer Analyse der Laufzeit und der Angabe der Punkte zu den jeweiligen Durchgängen. Dabei wird auch auf das verwendete Rechnersystem eingegangen und die verwendete Hard- beziehungsweise Software angeführt. Danach werden in Abschnitt 5.2 die statistischen Kennwerte diskutiert und abschließend in Abschnitt 5.3 die Ergebnisse mit verschiedenen Schnitten grafisch dargestellt und verglichen.

Bevor jedoch die Ergebnisse besprochen werden, sind in Tabelle 5.1 noch kurz die Beschriftungen der nachfolgenden Grafiken und Diagramme erklärt. Diese beschreiben die verwendeten Punkte, die in einem Iterative Closest Point Algorithmus Durchgang prozessiert werden.

Tabelle 5.1: Diagrammbeschriftung

complete	Komplette Punktwolke ohne die <i>rauen</i> Punkte
trees	<i>complete</i> Punktwolke mit ausgeschnittenen Baumbereichen
trees_2-6	<i>trees</i> Punktwolke im Bereich von 2 m bis 6 m
trees_no2-6	<i>trees</i> ohne den Bereich von 2 m bis 6 m
trees_no1-7.5	<i>trees</i> ohne den Bereich von 1 m bis 7.5 m
trees_gt6	<i>trees</i> mit allen Punkten über 6 m
trees_lt2.5	<i>trees</i> mit allen Punkten unter 2.5 m

5.1 Laufzeit

Um laufzeitmäßig einigermaßen vergleichbare Ergebnisse produzieren zu können, müssen die Berechnungen immer auf einem ähnlich leistungsstarken System durchgeführt werden. Tabelle 5.2 zeigt die wichtigsten Hard- und Softwarekomponenten des Systems, auf dem die Prozessierung stattfindet. Es sei zu erwähnen, dass es sich beim Betriebssystem des Servers um eine Linux Distribution handelt. Da jedoch das ICP Modul bisher nur auf Windows Systemen funktioniert, muss für diese Arbeit eine virtuelle Maschine aufgesetzt werden, worauf dann OPALS ausgeführt wird. Folglich verschlechtert sich dadurch die Performance, was sich aber bei allen Szenarien gleich auswirkt und somit das Ergebnis nicht verfälscht.

Bevor jedoch mit der Analyse der Laufzeiten begonnen wird, werden die verwendeten Punktmengen genauer betrachtet. Diese Anzahl hat zweifelsohne den größten Einfluss auf die Dauer der Berechnungen. In Abbildung 5.1 ist die Menge der gesamten Datenpunkte der einzelnen Szenarien in *Blau* zu sehen. Die komplette Punktwolke ist dabei fast um den Faktor 14 größer, als die nächstgrößere mit den ausgeschnittenen Bäumen. Die Balken in dem Diagramm 5.1 täuschen bei der Darstellung der Punktmenge, da die Y-Achse logarithmisch skaliert ist, um überhaupt einen Unterschied bei den anderen Punktwolken zu sehen. Auffallend ist außerdem, dass *trees_gt6* vergleichsweise sehr wenige Datenpunkte enthält. Das lässt darauf schließen, dass in den höheren Bereichen eine viel geringere Punktdichte vorherrscht. Dies kann einerseits durch die Abschattung der tiefer liegenden Äste, andererseits auch durch den schlechteren Einfallswinkel und längeren Strecken begründet werden.

Tabelle 5.2: Verwendete Hard- und Software

Hardware	
Prozessor	Intel® Xeon® E3-1230 v3 @ 3.30 GHz
Arbeitsspeicher	32 GB @ 1600 MHz
Mainboard	MSI Z87-G43W
Festplatte	Seagate 1 TB SATA II @ 7200 RPM
Software	
Betriebssystem	Ubuntu Server 14.04.3 LTS
VirtualBox Version	v5.0.40
VirtualBox Spezifikation	
VBox Arbeitsspeicher	15 GB @ 1600 MHz
VBox Gast System	Windows 7 Professional x64 SP1
OPALS	v2.1.5 x64

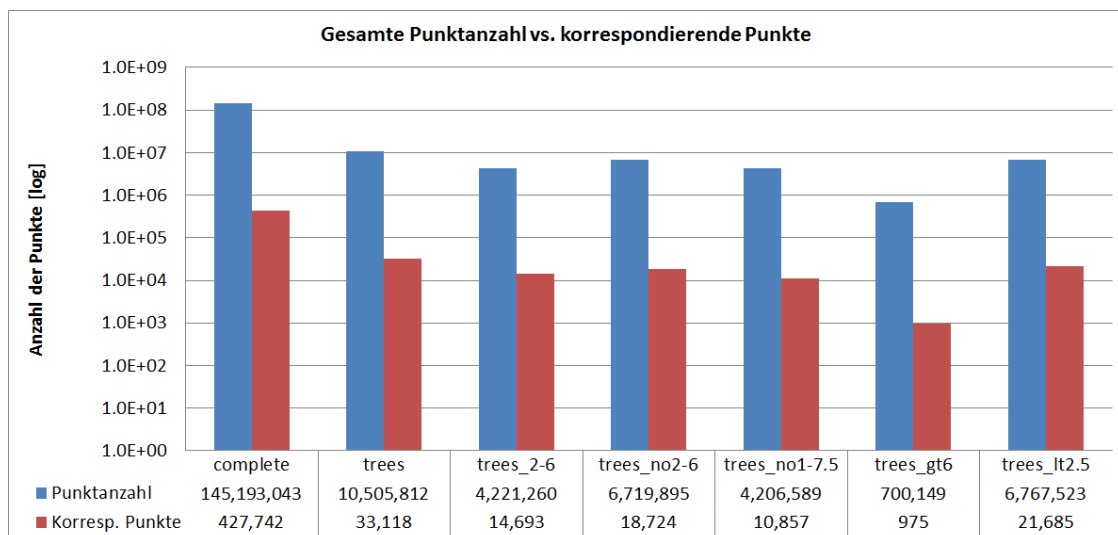


Abbildung 5.1: Anzahl der gesamten Punkte im Vergleich zu den korrespondierenden Punkten (Y-Achse ist logarithmisch skaliert)

Die roten Balken in der Abbildung 5.1 zeigen die Anzahl der korrespondierenden Punktpaare, die im letzten Iterationsschritt des ICP Algorithmus zur Bestimmung der Registrierungsparameter verwendet werden. Stellt man diesen Wert dann im Vergleich zu der gesamten Punktzahl, ergibt das ein Verhältnis von ungefähr 0.3% (siehe Abbildung 5.2). Einzig das Szenario mit den Punkten über 6 m Höhe (*trees_gt6*) hat verhältnismäßig einen geringeren Anteil an korrespondierenden Punkten, was auch wieder auf die niedrigere Punktdichte zurückzuführen ist. Es werden durch den eingeschränkten Suchbereich bei der Nachbarschaftssuche von 5 cm einfach weniger Punkte gefunden.

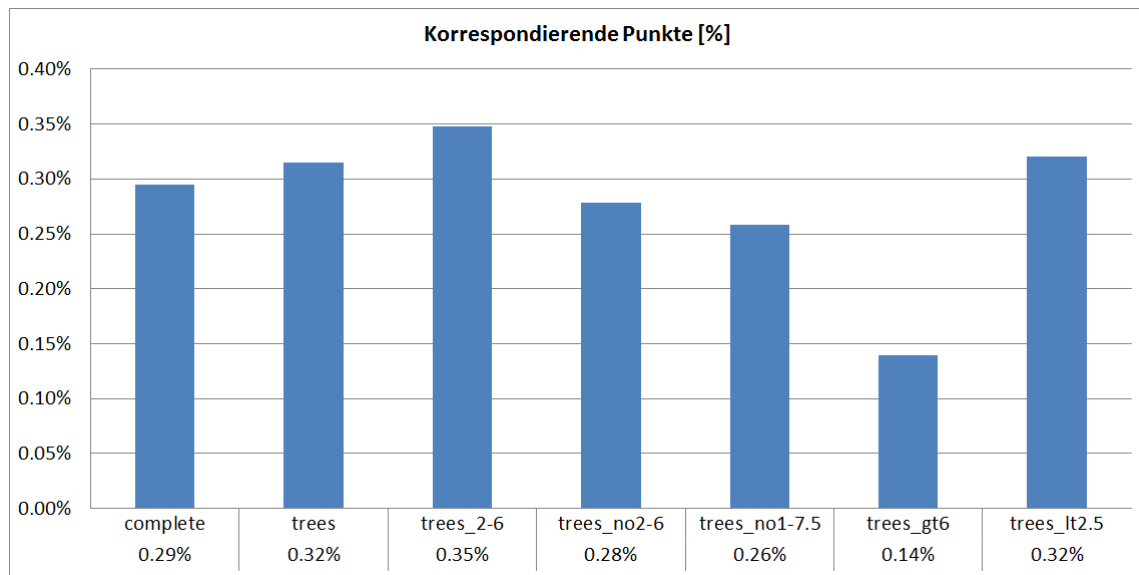
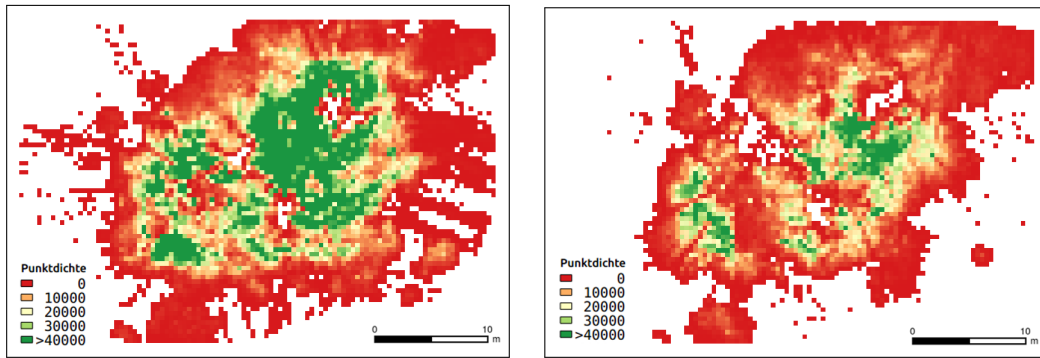


Abbildung 5.2: Anzahl der korrespondierenden Punkte pro Szenario

In Abbildung 5.3 sind die unterschiedlich hohen Punktdichten dargestellt. Sie wurden mit einer Zellgröße von 0.5 m berechnet und im Bereich von 0 bis 100 000 Punkten / Fläche eingefärbt. Die grünen Areale stellen dabei die dichteren Gebiete dar, welche klarerweise im Zentrum des Messgebiets sind, da diese Bereiche von mehreren TLS Standpunkten aufgenommen wurden.



(a) Schicht: 2 m bis 6 m über DGM (b) Schicht: höher als 6 m über DGM

Abbildung 5.3: Punktdichte in verschiedenen Höhenschichten (Pixelgröße: 0.5)

Nachdem nun die Anzahl der Datenpunkte und korrespondierende Punktepaare des ICP Algorithmus diskutiert sind, wird nachfolgend eine Laufzeitanalyse angestellt. Abbildung 5.4 zeigt die jeweiligen Laufzeiten der verschiedenen Durchgänge. Auch hier ist wieder zu beachten, dass die Y-Achse logarithmisch skaliert ist, da die Laufzeit der kompletten Punktwolke im Vergleich zum Rest ungemein höher ist. Um das Ganze in Zahlen auszudrücken: die Zeit für das ICP Modul bei der *complete* Punktwolke beträgt knapp 11 Stunden während die Nächstlangsammere etwas weniger als 10 Minuten braucht.

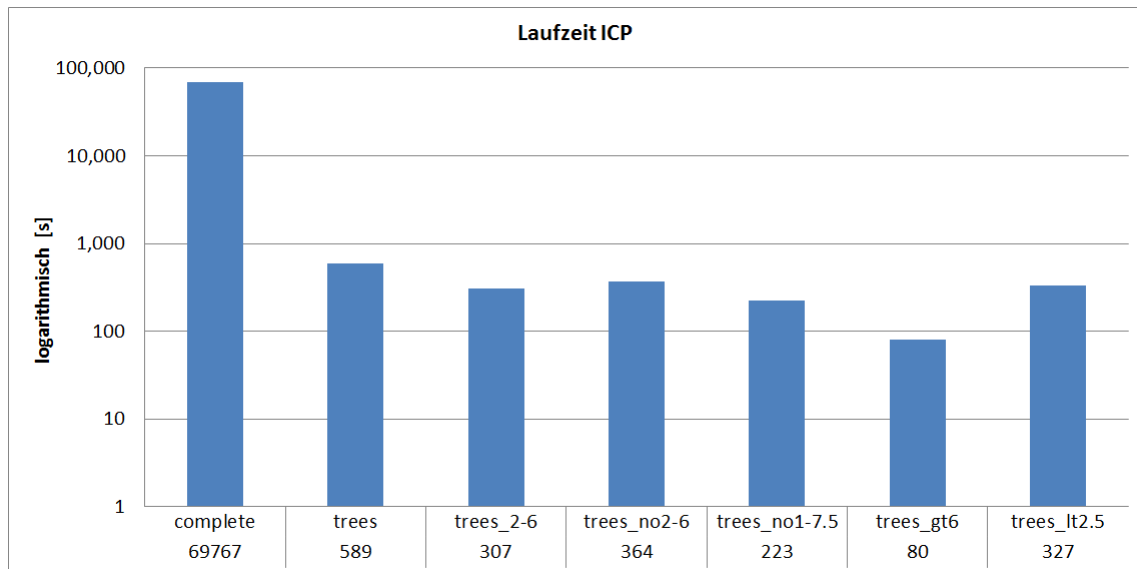


Abbildung 5.4: Laufzeitvergleich des ICP-Algorithmus mit verschiedenen Schichten (Y-Achse ist logarithmisch)

Nun wird überprüft, ob die Laufzeit linear mit der Anzahl der verwendeten Punkte zusammenhängt. Dazu wird die Dauer für die Berechnung von 10 000 Datenpunkte bestimmt, welche in Abbildung 5.5 dargestellt ist. Hier ist zu sehen, dass auch trotz

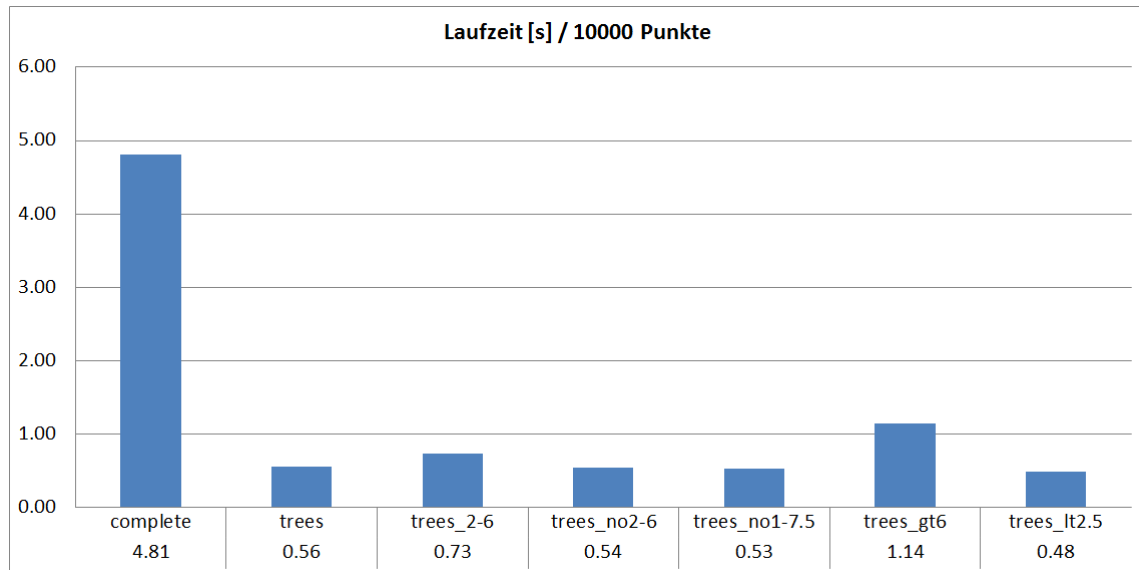


Abbildung 5.5: Laufzeit pro 10 000 Datenpunkte des ICP-Algorithmus mit verschiedenen Schichten

Division der Laufzeit durch die sehr große Punktzahl des *complete* Datensatzes, nur ein sehr schlechter Wert erreicht wird.

Auffallend ist außerdem, dass das laufzeitmäßig schnellste Szenario bei der Verhältnissbildung einen fast doppelt so hohen Wert aufweist, wie die anderen Testbereiche. Das könnte an etwaigen Initialisierungen des Iterative Closest Point Algorithmus liegen, die im Vorfeld erledigt werden müssen, bevor überhaupt mit den Berechnungen begonnen wird.

Selbiger Schritt wird nun wiederholt, um das Verhältnis zwischen Laufzeit und korrespondierenden Punkten zu analysieren. Hierbei reicht es, wenn man die Dauer pro 100 Punktpeare angibt, da hier viel höhere Werte auftreten. Grafisch ist das Ergebnis in Abbildung 5.6 dargestellt, wo zu sehen ist, dass sich das Phänomen von vorhin wiederholt und sogar um einiges verstärkt.

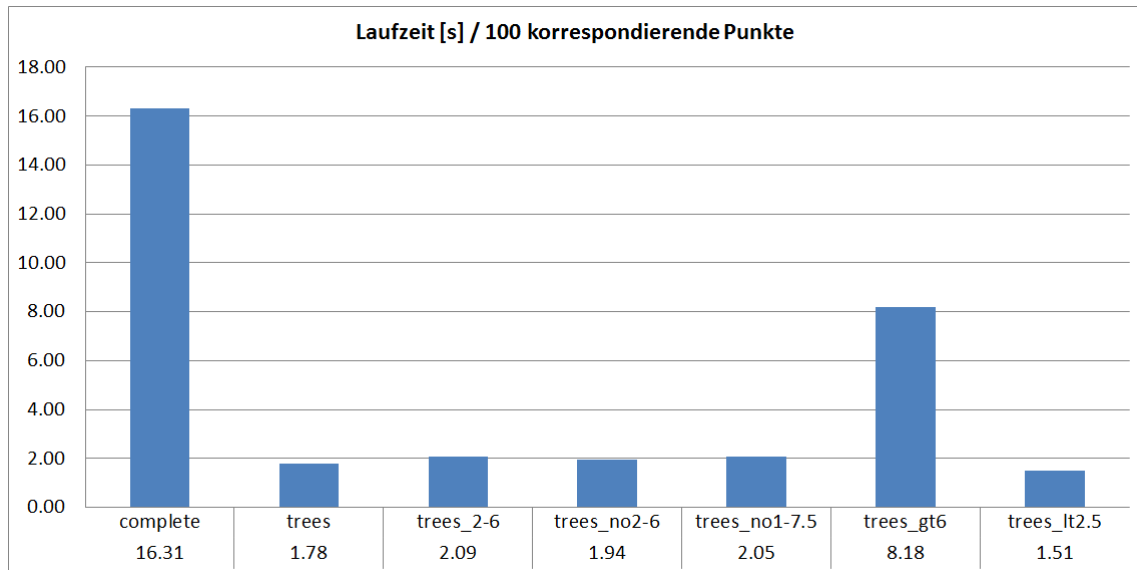


Abbildung 5.6: Laufzeit pro 100 korrespondierender Datenpunkte des ICP-Algorithmus mit verschiedenen Schichten

Diese Verbesserung der Laufzeit ist natürlich nur mit entsprechender Vorbereitung der originalen Scandaten möglich. Diese Prozessierungszeit darf fairerweise auch nicht vernachlässigt werden, da es sich doch um einige rechenintensive Schritte handelt. Abbildung 5.7 zeigt die einzelnen Vorgänge mit deren Laufzeit. Es werden zusätzlich zu den angeführten Schritten noch einige andere angewandt, welche sich aber in der gesamten Laufzeit nicht auswirken.

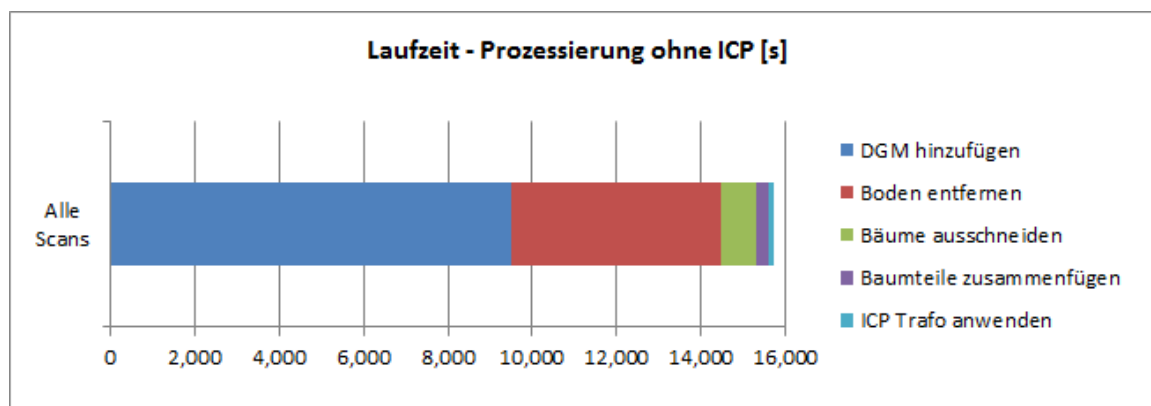


Abbildung 5.7: Laufzeit der Verarbeitungsschritte exklusive dem ICP-Algorithmus

Addiert man nun diese zusätzliche Prozessierungszeit zu der Dauer des ICP- Algorithmus, ergibt sich die Gesamtlaufzeit, welche in Abbildung 5.8 dargestellt ist. Trotz vorheriger aufwendiger Verarbeitung der Punktwolken, ist die gesamte Ausführungsdauer noch um einiges schneller, als die Verwendung der kompletten, originalen Daten.

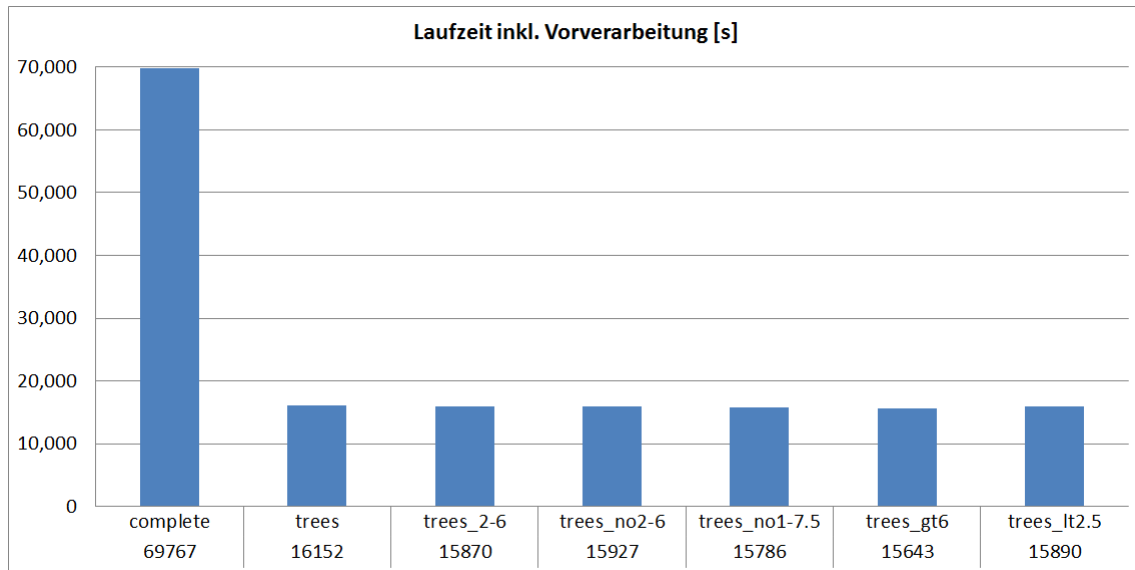


Abbildung 5.8: Gesamte Laufzeit der Datenprozessierung inklusive Vorverarbeitung

5.2 Statistische Kennwerte

Da jedoch die noch so schnelle Laufzeit bei schlechten Resultaten der Registrierung irrelevant ist, wird in diesem Abschnitt die Qualität des ICP Algorithmus anhand von Standardabweichung und Mittelwert bewertet und so auf geeignete Szenarien geschlossen.

Eine bewährte Methode zur Qualitätsbewertung ist der Vergleich der Standardabweichungen der verschiedenen Szenarien. Zur Berechnung wird der Punkt-zu-Ebene Abstand der korrespondierenden Punktpaare herangezogen. Je kleiner dieser Wert ist, desto besser sind die einzelnen Punktwolken zueinander orientiert. In Diagramm 5.9 sind die Standardabweichungen für die unterschiedlichen Szenarien dargestellt. Diese wird nach der fünften Iteration berechnet, indem die Abweichungen der einzelnen Scans zur fixierten Punktwolke verwendet werden. Der Versuch mit dem vollständigen Scan *complete* und der Schicht über 6m zeigt einen doppelt so hohen Wert, wie der Rest der Testbereiche. Dies lässt sich für den ersten Datensatz durch die viel größere Anzahl an korrespondierenden Punkte erklären, da hierbei auch eine Vielzahl an Ausreißern dabei ist. Für die Schicht mit den Punkten über 6m ist, wie schon oben erwähnt, die niedrigere Punktdichte der Grund für den erhöhten Wert und auch die schlechteren Ergebnisse.

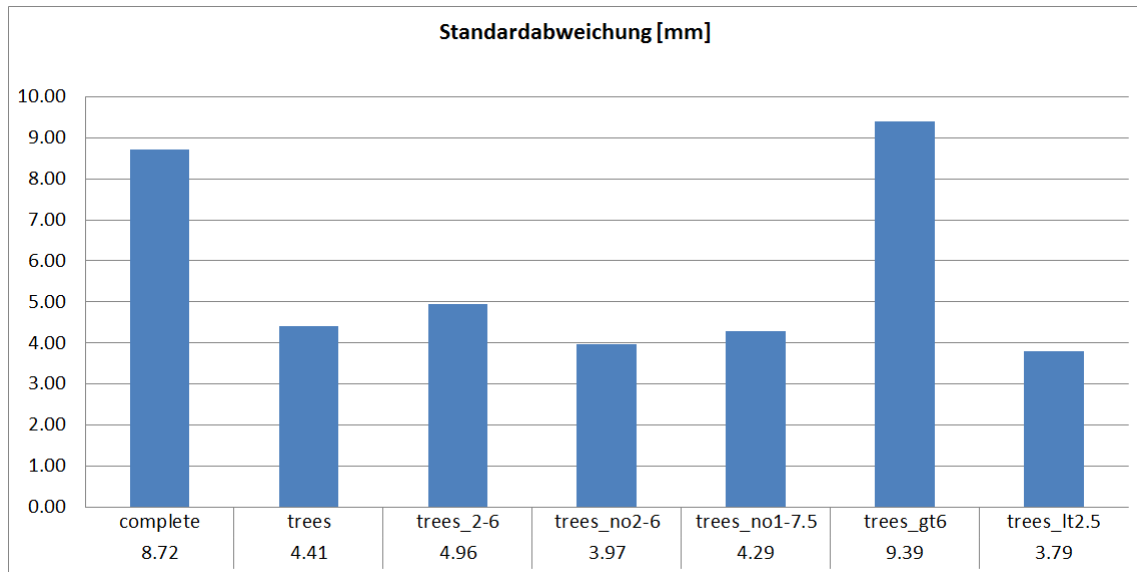


Abbildung 5.9: Standardabweichung der registrierten Punktwolken nach der 5. Iteration

In Tabelle 3.3 ist bereits die Registrierungsqualität aus *RiSCAN Pro* aufgelistet. Diese Werte sind pro Punktwolke angegeben und können mit den Ergebnissen des ICP verglichen werden. Dazu wird jeweils die Differenz von $Std_{ICP} - Std_{RiSCAN}$ berechnet, wie in Tabelle 5.3 zu sehen ist. Ein negativer Wert bedeutet, dass die Standardabweichung des ICP kleiner ist und somit ein besseres Ergebnis erzielt wurde.

Tabelle 5.3: Differenz der Standardabweichung zwischen *RiSCAN*- und ICP Registrierung

Scan [mm]	complete	trees	2-6	no2-6	no1-7.5	gt6	lt2.5
082512	10.64	4.84	0.00	-0.83	4.56	0.00	4.39
084146	3.90	-0.81	-0.03	-2.47	-0.85	6.97	-1.67
080033	0.18	-3.32	-3.24	2.03	-4.27	0.00	-4.18
081609	1.66	-4.27	-4.37	-3.84	-3.09	-0.28	-4.17
074902	-3.58	-7.51	-6.51	-8.21	-8.10	-2.46	-8.56
075500	0.45	-3.37	-2.90	-3.67	-3.56	-0.49	-3.99
074451	2.30	-0.59	0.09	-0.76	-0.78	0.00	-0.76
Mittelwert	2.22	-2.15	-2.42	-2.54	-2.30	0.53	-2.71

Nach Mittelwertbildung der Standardabweichungen der Scanpositionen ergibt sich Diagramm 5.10. Auch hier ist zu sehen, dass die *complete* und die *trees_gt6* Punktwolke sogar einen schlechteren Wert erzielen, als die originalen Daten.

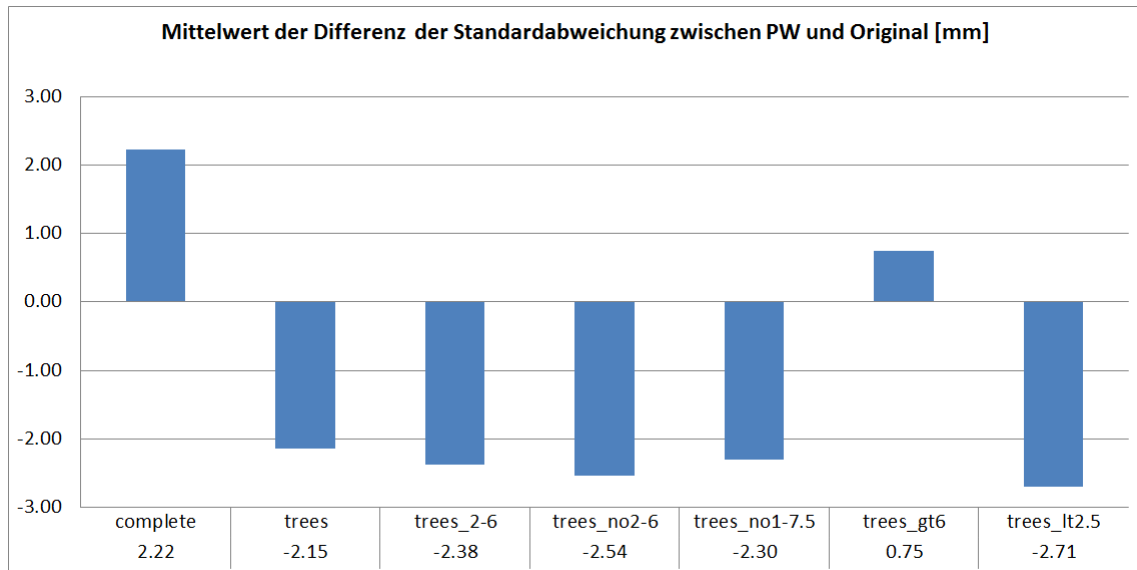


Abbildung 5.10: Mittelwert der Standardabweichungs Differenz zwischen den einzelnen Punktwolken und der Ausgangsregistrierung

Bei der Analyse des Mittelwerts zeigt sich ein ähnliches Bild. Die Punktwolke mit den höher gelegenen Datenpunkten hat auch hier mit Abstand den größten Wert. Die Verwendung der kompletten Scans hingegen erzielt einen Mittelwert von fast *Null*. Dieser kann auch wieder auf die vergleichsweise große Anzahl von korrespondierenden Punktepaaren zurückzuführen sein. Diagramm 5.11 zeigt die Werte der verschiedenen Szenarien. Im Letzten ist der Mittelwert negativ, was sich durch die Berechnungsmethode des Punkt-zu-Ebenen Abstandes erklären lässt. Je nachdem, ob der Datenpunkt unter oder über der verwendeten Ebene liegt, können Werte kleiner *Null* vorkommen.

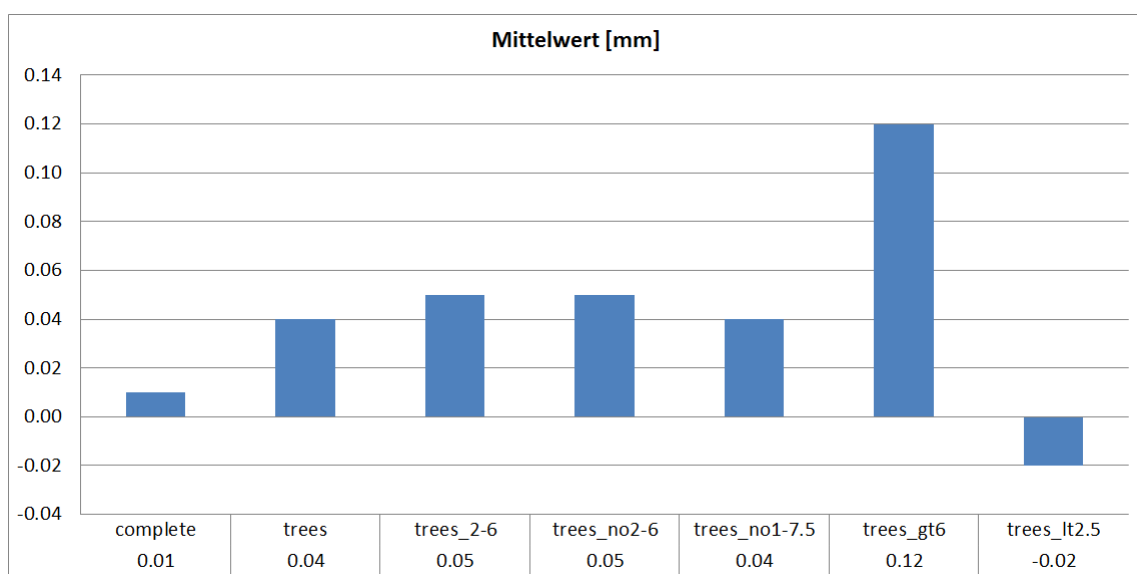


Abbildung 5.11: Mittelwert der registrierten Punktwolken nach der 5. Iteration

In Diagramm 5.12 ist die Veränderung der Kennwerte von Szenario *trees_2-6* nach jedem Iterationsschritt dargestellt. Es ist gut zu sehen, dass die Änderung der Parameterzuschläge (*Norm*) für die Transformation zu Beginn sehr stark ist und gegen Ende hin minimal wird. Die Standardabweichung bleibt relativ konstant bei einem halben Zentimeter und der Mittelwert verbessert sich auf 0.05 mm.

Diese Parameterzuschläge beziehen sich auf die Änderung der Zuschläge zu den unbekanntem Transformationsparametern. Dieser Wert wird normalerweise von einer Iteration zur nächsten immer kleiner, da die Punktwolken immer besser angepasst sind, was in Diagramm 5.12 in Grün gut zu sehen ist.

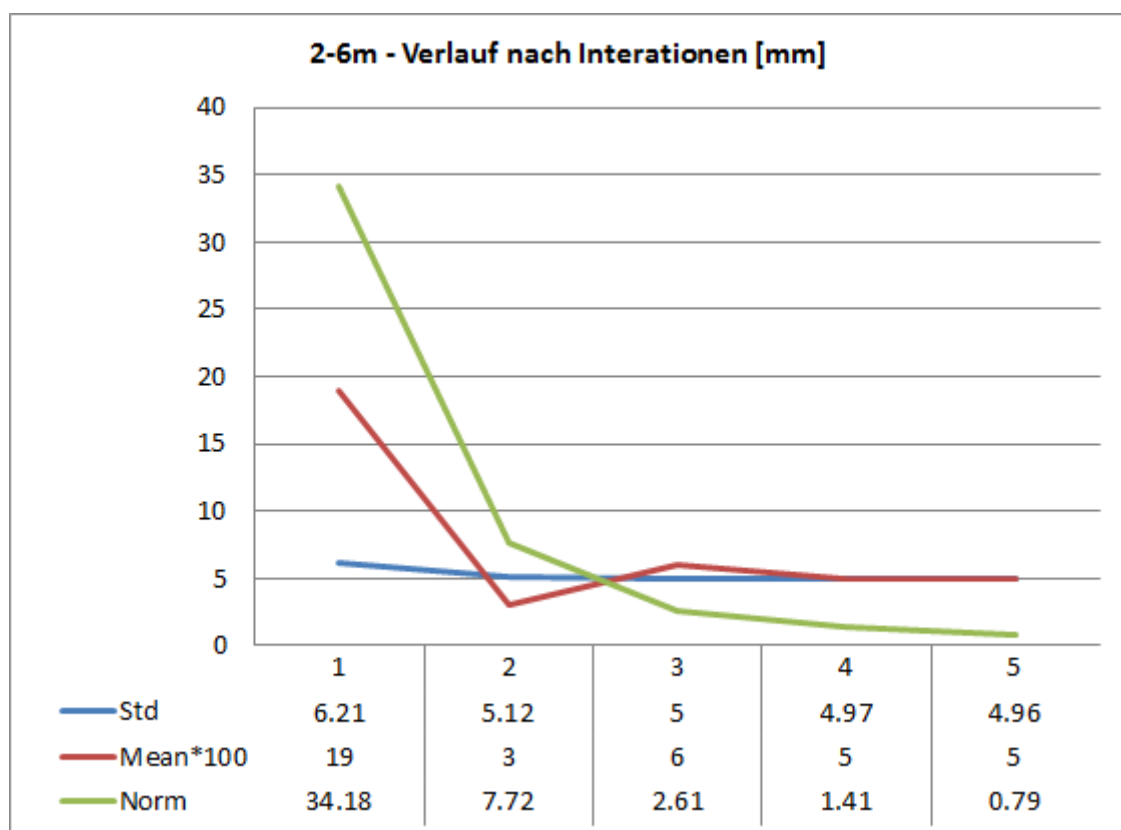


Abbildung 5.12: Änderung der Kennwerte bei jeder Iteration

5.3 Grafische Darstellung

Nachdem die Laufzeiten und statistischen Kennwerte diskutiert sind, folgen in diesem Abschnitt noch grafische Darstellungen der Ergebnisse, um die Auswirkungen des ICP Algorithmus auf die transformierte Punktwolke zu veranschaulichen. Genauer werden

Schnitte in verschiedenen Höhen durch diverse Bäume gelegt und die Unterschiede der verschiedenen Szenarien aufgezeigt.

In den nachfolgenden Grafiken werden die unterschiedlichen Scanpositionen jeweils in den gleichen Farben dargestellt, um eine bessere Vergleichbarkeit zu ermöglichen. Die Abbildungen sind in Ausschnitte von (a) bis (d) unterteilt und stellen verschiedene Szenarien aus den Punktwolken dar.

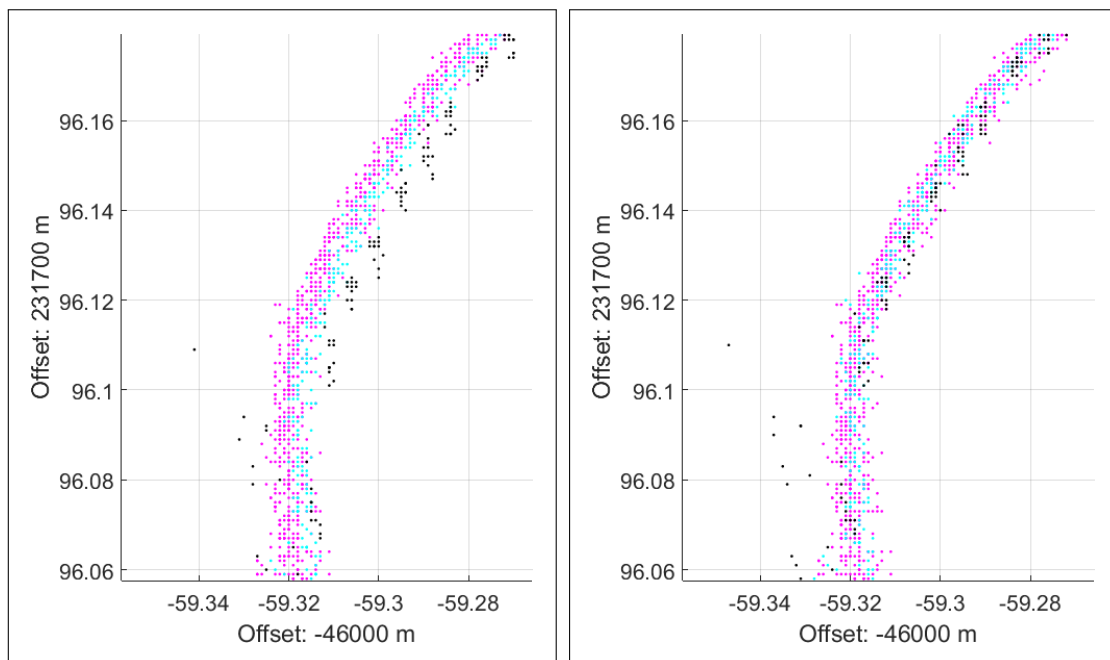
- (a) zeigt die Registrierung, die mittels *RiSCAN Pro* durchgeführt wurde. Diese Orientierung wird als Vorregistrierung für den ICP- Algorithmus verwendet.
- (b) ist das Ergebnis unter Verwendung der Bäume im Bereich 2 m bis 6 m über dem Geländemodell.
- (c) stellt die transformierten Punktwolken dar, wobei zur Berechnung der Parameter nur Punkte über 6 m verwendet wurden.
- (d) wird erreicht, wenn zur Bestimmung der Transformationsparameter alle Datenpunkte der Scans verwendet werden.

Es sei vorweggenommen, dass nicht nur diese vier Szenarien durchgespielt wurden, sondern, wie schon in Abbildung 4.8 gezeigt, auch noch einige andere Schichten getestet wurden. In diesen Ergebnissen lassen sich jedoch mit freiem Auge keinerlei Unterschiede erkennen und sie sind daher nicht mehr extra dargestellt. Nach der Neubestimmung der Registrierung gleichen die nicht gezeigten Punktwolken den Abbildungen (b) bzw (d).

Abbildung 5.13 zeigt einen Schnitt in der Höhe von 2.5 m durch den Baum Nummer 4. Die Breite der Schicht ist 10 cm und wurde auf eine Ebene projiziert, um die Anzahl der Punkte und somit die Sichtbarkeit des Ergebnisses zu erhöhen.

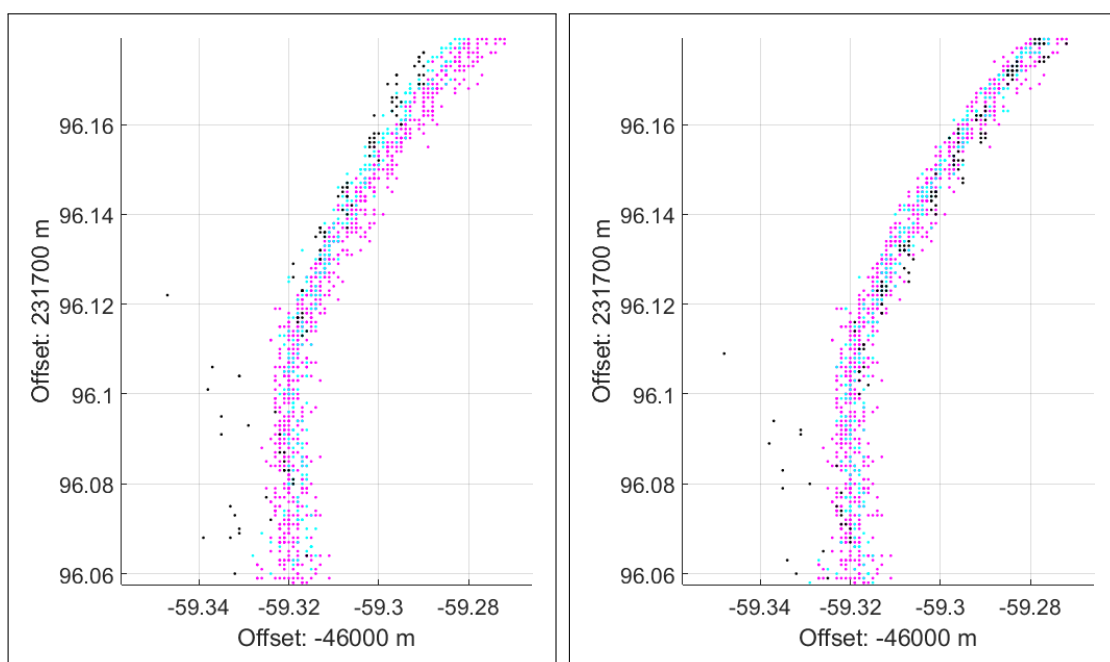
Das Bild der Ausgangspunktwolke (a) zeigt deutlich die nebeneinander liegenden, verschieden farbigen Punkte, die den Stamm eines Baumes darstellen sollen. Nach Bestimmung und Anwendung der Transformationsparameter mittels ICP unter Verwendung der Schicht *trees_2-6* bzw der gesamten Scandaten, ergeben sich die Bilder (b) und (d). Hier ist, wie schon vorhin erwähnt, in der grafischen Darstellung kein Unterschied der Ergebnisse zu erkennen. Die einzelnen Scans decken sich optimal und zeigen den Teil des Baumstamms. Die Punktwolke in *Magenta* wurde bei der Berechnung fixiert und die übrigen bestmöglich dazu orientiert. Anders sieht das Resultat bei Verwendung der *trees_gt6* Schicht aus, was allein schon nach Betrachtung der statistischen Kennwerte

in Abschnitt 5.2 nicht verwunderlich ist. Nach der Transformation sind die Punkte im Vergleich zur fixierten Punktwolke auf die andere Seite verschoben.



(a) Szenario: *original*

(b) Szenario: *trees_2-6*



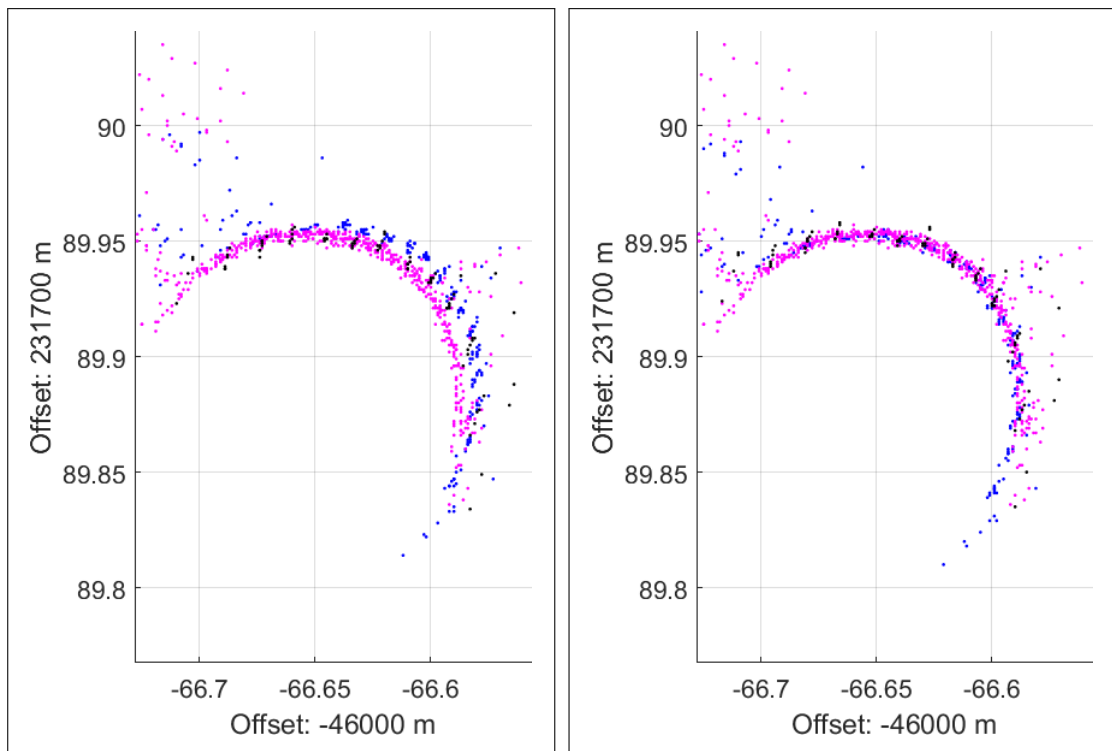
(c) Szenario: *trees_gt6*

(d) Szenario: *complete*

Abbildung 5.13: Schnitt durch die Punktwolke im Bereich von Baum Nr. 4 in 2.5 m Höhe

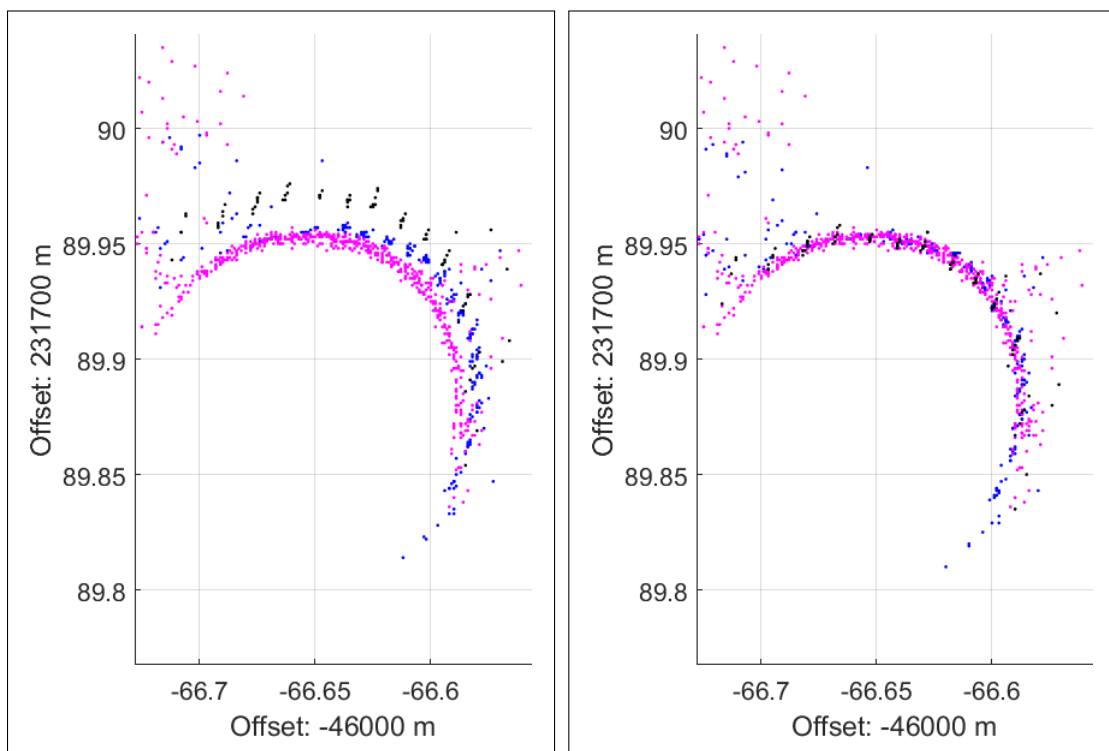
Ein ähnliches Bild zeigt sich in Abbildung 5.14, wobei der Schnitt diesmal durch die Punktwolke bei Baum Nummer 8 und in einer Höhe von 4.5 m gelegt ist. Darstellung (b) und (d) können wiederum mit dem Auge nicht unterschieden werden, was sich auch bei den restlichen, nicht dargestellten Schichten zeigt. Das Szenario mit den höher gelegenen Datenpunkten (*trees_gt6*) weist im Vergleich zum Schnitt in 2.5 m Höhe eine noch größere Klaffung auf. Das lässt darauf schließen, dass die Registrierung nach oben hin immer schlechter wird.

Abschließend wird in Abbildung 5.15 noch ein Schnitt durch einen Baumstamm gezeigt, dessen Durchmesser ungefähr 45 cm ist und der von allen Seiten gescannt wurde. Es ist hier allerdings schwierig, größere Unterschiede zwischen den einzelnen Szenarien auszumachen. Am besten ist noch ein Abdriften der schwarzen Punktwolke in Abbildung (c) in Richtung Norden zu erkennen. Vergleichsweise kompakter und besser überlappend zeigen sich die verschiedenen Scans in Abbildung (b) und (d). Ebenso verkleinert sich die *Lücke* zwischen der grünen, türkisen und pinken Punktwolke nach Anwendung des ICP- Algorithmus.



(a) Szenario: *original*

(b) Szenario: *trees_2-6*



(c) Szenario: *trees_gt6*

(d) Szenario: *complete*

Abbildung 5.14: Schnitt durch die Punktwolke im Bereich von Baum Nr. 8 in 4.5 m Höhe

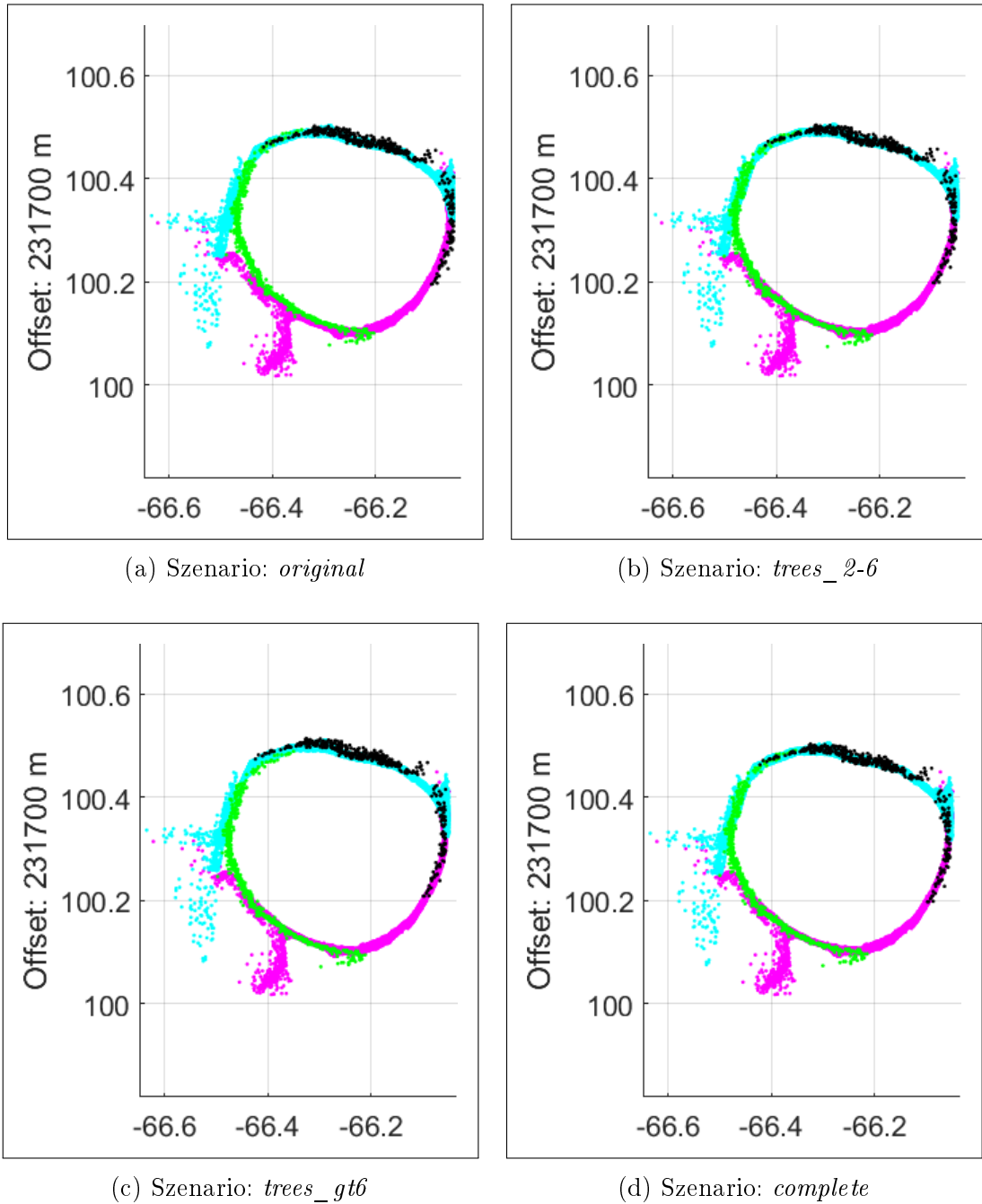


Abbildung 5.15: Schnitt durch die Punktwolke im Bereich von Baum Nr. 2 in 2 m Höhe

Kapitel 6

Resümee und Ausblick

Wie in den Ergebnissen in Kapitel 5 zu sehen ist, ist eine Vorverarbeitung der Punktwolken auf jeden Fall sinnvoll, bevor der Iterative Closest Point Algorithmus auf die Daten angewandt wird. Die Laufzeit der Bestimmung von Transformationsparametern zur besseren Registrierung nimmt mit geringerer Punkteanzahl immens ab.

Die in dieser Arbeit durchgeführten Schritte zur Datenreduktion bestehen aus:

1. Bestimmen der Höhe über DGM
2. Entfernen des Bodens, bodennahe und *raue* Punkte
3. Entfernen der Punkte mit zu hohem Wert der Z-Komponente des Normalvektors
4. Ausschneiden detektierter Bäume aus Punktwolken

Diese Vorbereitung hat natürlich auch eine nicht zu vernachlässigende Laufzeit, die vor allem mit der Verwendung des digitalen Geländemodells einen wesentlichen Anteil der Zeit ausmacht. Hier könnte in weiteren Arbeiten eventuell untersucht werden, ob diese Bestimmung der Höhe über dem DGM vereinfacht oder optimiert werden kann.

Weiters spricht für diese hier verwendete Methodik, dass sich die statistischen Kennwerte im Vergleich zur ursprünglichen Registrierung deutlich verbessert haben. Das zeigt sich auch in der grafischen Darstellung der Schnitte durch die Punktwolke in Abschnitt 5.3. Es sei jedoch zu erwähnen, dass zu den Werten aus *RiSCAN PRO* keine Informationen über deren Berechnung vorliegen.

Sowohl in der grafischen Beurteilung, als auch in den statistischen Kennwerten zeigt sich, dass das behandelte Szenario gar nicht wirklich ausschlaggebend ist, sondern alle

ein vergleichbares Resultat liefern und mit freiem Auge oft nicht mehr unterschieden werden kann. Einzig der Testfall mit Datenpunkten über 6 m Höhe liefert ein deutlich schlechteres Ergebnis. Dies lässt sich durch die geringere Punktdichte in höheren Bereichen erklären.

Für diese Arbeit war es ausreichend, nur ein paar der vorhandenen Bäume automatisch zu detektieren und zur Bestimmung der neuen Registrierung zu verwenden. Es wäre in Zukunft jedoch auch interessant, wie sich die Ergebnisse verändern, wenn alle, in den Punktwolken enthaltenen Bäume bei der Anwendung des ICP- Algorithmus berücksichtigt werden. Es könnte auch untersucht werden, wie sich die Punktwolken verhalten, wenn wirklich nur der Baumstamm verwendet wird und nicht ein Quader mit einer Grundfläche von 1.5 m x 1.5 m mitsamt allen Ästen des Baumes aus den Scandaten geschnitten wird.

Eine Grundvoraussetzung des Iterative Closest Point Algorithmus ist, wie schon oben erwähnt, eine grobe Registrierung der Punktwolken. Diese wurde hier mit Reflektorfolien im Feld und *RiSCAN PRO* im Postprocessing gewährleistet. Da die verwendeten Daten für diese Arbeit ebenfalls Grundlage für ein anderes Projekt sind, ist diese grobe Registrierung relativ genau ausgefallen. Hier könnte noch weiter geforscht werden, wie akkurat diese erste Orientierung der Punktwolken sein muss, um die Passpunkte im Feld zu minimieren und die Arbeit effizienter zu machen. Die beste Lösung wäre natürlich der komplette Verzicht auf eine Vermarkung bei der Aufnahme und die grobe Registrierung nur anhand von markanten Punkten zu erreichen.

Literaturverzeichnis

- [Bergevin u. a. 1996] BERGEVIN, Robert ; SOUCY, Marc ; GAGNON, Hervé ; LAURENDEAU, Denis: Towards a general multi-view registration technique. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (1996), Nr. 5, S. 540–547
- [Besl und McKay 1992] BESL, Paul J. ; MCKAY, Neil D.: A Method for Registration of 3-D Shapes. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (1992), Februar, Nr. 2, S. 239–256. – URL <http://dx.doi.org/10.1109/34.121791>. – ISSN 0162-8828
- [Böhm und Haala 2005] BÖHM, J. ; HAALA, N.: Efficient integration of aerial and terrestrial laser data for virtual city modeling using Lasermaps. In: *International Archives of Photogrammetry and Remote Sensing XXXVI*, 3/W19 (2005), S. 192 – 197. – Enschede, Niederlande
- [Bienert 2013] BIENERT, Anne: *Automatische Extraktion von 3D-Baumparametern aus terrestrischen Laserscannerdaten*, TU Dresden, Dissertation, 2013
- [BioSLIDE Team 2017] BIOSLIDE TEAM: *BioSLIDE*. 2017. – URL <http://bioslide.geo.tuwien.ac.at/>
- [Chen und Medioni 1992] CHEN, Yang ; MEDIONI, Gérard: Object Modelling by Registration of Multiple Range Images. In: *Image Vision Comput.* 10 (1992), April, Nr. 3, S. 145–155. – URL [http://dx.doi.org/10.1016/0262-8856\(92\)90066-C](http://dx.doi.org/10.1016/0262-8856(92)90066-C). – ISSN 0262-8856
- [de.wikipedia.org 2017] DE.WIKIPEDIA.ORG: *Matlab - Wikipedia*. 2017. – URL <https://de.wikipedia.org/wiki/Matlab>
- [Glira u. a. 2015] GLIRA, Philipp ; PFEIFER, Norbert ; BRIESE, Christian ; RESSL, Camillo: A Correspondence Framework for ALS Strip Adjustments based on Variants of the ICP Algorithm Korrespondenzen für die ALS-Streifenausgleichung auf Basis

- von ICP. In: *Photogrammetrie - Fernerkundung - Geoinformation* 2015 (2015), Nr. 4, S. 275–289. – URL <http://www.ingentaconnect.com/content/schweiz/pfg/2015/00002015/00000004/art00001>. – ISSN 1432-8364
- [Haring 2007] HARING, Alexander: *Die Orientierung von Laserscanner- und Bilddaten bei der fahrzeuggestützten Objekterfassung*, TU Vienna, Dissertation, 2007
- [Hesse 2008] HESSE, C.: *Hochauflösende kinematische Objekterfassung mit terrestrischen Laserscannern*. Fachrichtung Geodäsie und Geoinformatik der Leibniz-Univ., 2008 (Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz-Universität Hannover)
- [Kahmen 2006] KAHMEN, Heribert: *Angewandte Geodäsie: Vermessungskunde*. Berlin, New York : Walter de Gruyter, 2006. – ISBN 3-11-018464-8
- [Kraus 2004] KRAUS, Karl: *Photogrammetrie, Band 1, Geometrische Informationen aus Photographien und Laserscanneraufnahmen*. Berlin : Walter de Gruyter, 2004. – ISBN 3-11-017708-0
- [Leica Geosystems AG 2017] LEICA GEOSYSTEMS AG: *Datasheet Leica 1200+*. 2017. – URL <http://www.leica-geosystems.com/>
- [Li u. a. 2009] LI, Baoshun ; CAO, Qixin ; ZHANG, Lei ; LEE, Jay ; LENG, Chuntao ; LI, Yaoliang: 3D surface modeling and measuring system for pneumatic caisson. In: *Journal of Computing in Civil Engineering* 24 (2009), Nr. 3, S. 223–240
- [Mandlbürger u. a. 2010] MANDLBURGER, G. ; OTEPKA, J. ; KAREL, W. ; WÖHRER, B. ; WAGNER, W. ; PFEIFER, N.: OPALS (Orientation and Processing of Airborne Laser Scanning data) - Konzept und Anwendungsbeispiele einer wissenschaftlichen Laserscanning Software. In: *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V.* 19 (2010), S. 376 – 387. – URL https://publik.tuwien.ac.at/files/PubDat_188132.pdf. – Vortrag: Dreiländertagung - 30. Wissenschaftlich-Technische Jahrestagung der DGPF, Wien; 01.07.2010 - 03.07.2010. – ISSN 0942-2870
- [Mitra u. a. 2004] MITRA, Niloy J. ; GELFAND, Natasha ; POTTMANN, Helmut ; GUIBAS, Leonidas: Registration of Point Cloud Data from a Geometric Optimization Perspective. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. New York, NY, USA : ACM, 2004 (SGP '04), S. 22–31. – URL <http://doi.acm.org/10.1145/1057432.1057435>. – ISBN 3-905673-13-4

- [Neugebauer 1997] NEUGEBAUER, Peter J.: Geometrical cloning of 3D objects via simultaneous registration of multiple range images. In: *Shape Modeling and Applications, 1997. Proceedings., 1997 International Conference on IEEE* (Veranst.), 1997, S. 130–139
- [Nirmalakumari u. a. 2017] NIRMALAKUMARI, K ; PATIL, Puja V. ; KAUR, Amrit P.: Simultaneous Localization and Mapping. (2017)
- [OPALS Team 2017a] OPALS TEAM: *Opals Module ICP*. 2017. – URL <http://www.ipf.tuwien.ac.at/opals/html/ModuleICP.html>
- [OPALS Team 2017b] OPALS TEAM: *Opals Projektseite*. 2017. – URL <http://www.ipf.tuwien.ac.at/opals>
- [Pfeifer und Briese 2007] PFEIFER, Norbert ; BRIESE, Christian: *Laser scanning-Principles and applications*. na, 2007
- [Pfeifer und Mandlbürger 2008] PFEIFER, Norbert ; MANDLBURGER, Gottfried: *LIDAR Data Filtering and DTM Generation*. 11 2008
- [Pulli 1999] PULLI, Kari: Multiview Registration for Large Data Sets. In: *Proceedings of the 2Nd International Conference on 3-D Digital Imaging and Modeling*. Washington, DC, USA : IEEE Computer Society, 1999 (3DIM'99), S. 160–168. – URL <http://dl.acm.org/citation.cfm?id=1889712.1889738>. – ISBN 0-7695-0062-5
- [RIEGL Laser Measurement Systems GmbH 2017] RIEGL LASER MEASUREMENT SYSTEMS GMBH: *Datasheet RIEGL VZ-2000*. 2017. – URL <http://www.riegl.com/>
- [RIEGL VZ-2000 manual 2014] RIEGL VZ-2000 MANUAL: *General description and Data interfaces*. 2014. – URL www.riegl.com
- [Rietdorf 2005] RIETDORF, A.: *Automatisierte Auswertung und Kalibrierung von scannenden Messsystemen mit tachymetrischem Messprinzip*. Verlag der Bayer. Akad. der Wiss., 2005 (Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissenschaften. Reihe C: Dissertationen). – ISBN 9783769650211
- [Riisgaard und Blas 2003] RIISGAARD, Søren ; BLAS, Morten R.: SLAM for Dummies. In: *A Tutorial Approach to Simultaneous Localization and Mapping* 22 (2003), Nr. 1-127, S. 126

- [Rusinkiewicz und Levoy 2001] RUSINKIEWICZ, Szymon ; LEVOY, Marc: Efficient Variants of the ICP Algorithm. In: *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, 2001, S. 145–152
- [Teschl 2013] TESCHL, Susanne: MATLAB - Eine Einführung. (2013). – URL <http://staff.technikum-wien.at/~teschl/MatlabSkriptum.pdf>
- [unavco.org 2017] UNAVCO.ORG: *Non-profit university-governed consortium*. 2017. – URL www.unavco.org
- [Wang u. a. 2016] WANG, Di ; HOLLAUS, Markus ; SCHMALTZ, Elmar ; WIESER, Martin ; REIFELTSHAMMER, David ; PFEIFER, Norbert: Tree Stem Shapes Derived from TLS Data as an Indicator for Shallow Landslides. In: *Procedia Earth and Planetary Science* 16 (2016), S. 185 – 194. – URL <http://www.sciencedirect.com/science/article/pii/S1878522016300200>. – The Fourth Italian Workshop on Landslides. – ISSN 1878-5220
- [Zeinzinger 2013] ZEINZINGER, Birgit: *Absolute und relative Orientierung von TLS - Punktwolken mit TLS - Radoms*, Department für Geodäsie und Geoinformation, Diplomarbeit, 2013. – URL http://publik.tuwien.ac.at/files/PubDat_227505.pdf

Abbildungsverzeichnis

2.1	Messprinzip TLS	5
2.2	Funktionsweise TLS	6
2.3	Distanzmessung: Impulslaufzeitverfahren TLS	7
2.4	TLS Echos	8
2.5	Distanzmessung: Phasenvergleichsverfahren	9
2.6	Distanzmessung: Lichtschnittverfahren	10
2.7	Zielmarken	11
2.8	Messkugeln	12
2.9	Indirekte Georeferenzierung	14
2.10	ICP - Punkt zu Punkt Methode	21
2.11	ICP - Punkt zu Ebene Methode	21
2.12	ICP - Punkt zu Ebene Methode - Normal-Shooting	22
2.13	Vergleich Punkt-zu-Ebene Methoden	23
2.14	Funktionsweise ICP	24
2.15	Funktionsweise SLAM	28
3.1	Messgebiet	30
3.2	Festpunkte und Reflektorfolien	32
3.3	Position der Bäume	35
3.4	Nummerierung der Bäume	36
4.1	Workflow	38
4.2	DGM des Bereiches	39

4.3	nZ-Komponente im Scan	41
4.4	Punktdichtekarte	43
4.5	Notwendige Zwischenschritte zur automatischen Baumstammdetektion (Pixelgröße: 0.5)	44
4.6	Einzelner Baum im Scan	46
4.7	Kombinierte Baumanteile	46
4.8	Verwendete (grün) Punktwolkenanteile [Baum: clipart-work.net, 2017] .	48
4.9	Scan mit herausgeschnittenem Bereich	49
4.10	Ergebnis Plot vom ICP	52
5.1	Punkteverhältnis	56
5.2	Korrespondierende Punkte	57
5.3	Punktdichte in verschiedenen Höhenschichten (Pixelgröße: 0.5)	58
5.4	Laufzeit	58
5.5	Laufzeit pro Datenpunkt	59
5.6	Laufzeit pro korrespondierendem Datenpunkt	60
5.7	Laufzeit der Vorbereitung	60
5.8	Laufzeit inklusive Vorbereitung	61
5.9	Standardabweichung nach dem ICP	62
5.10	Differenz der Standardabweichungen	63
5.11	Mittelwert nach dem ICP	63
5.12	Verlauf pro Iteration	64
5.13	Schnitt: Baum 4, Höhe: 2.5 m	66
5.14	Schnitt: Baum 8, Höhe: 4.5 m	68
5.15	Schnitt: Baum 2, Höhe: 2 m	69

Codeverzeichnis

4.1	addInfo NormalizedZ	39
4.2	Normalvektoren berechnen	42
4.3	Lokale Maxima finden	43
4.4	Bäume aus der Punktwolke extrahieren	45
4.5	Schicht aus der Punktwolke schneiden	47
4.6	ICP auf die Scans anwenden	50
4.7	Transformationsparameter auf Punktwolken anwenden	52

Tabellenverzeichnis

3.1	Spezifikationen der verwendeten Geräte	31
3.2	Genauigkeit der Registrierung	33
3.3	Genauigkeit der Orientierung [Wang u. a., 2016]	33
3.4	DBH der gemessenen Bäumen mit statistischen Kennwerten	36
4.1	Soll/Ist Vergleich der Baumkoordinaten	45
4.2	Verwendete Schichten mit zugehörigem OPALS Filter	47
4.3	Verwendete ICP Parameter mit Beschreibung	50
5.1	Diagrammbeschriftung	55
5.2	Verwendete Hard- und Software	56
5.3	Differenz der Standardabweichung zwischen <i>RiSCAN</i> - und ICP Registrierung	62

Abkürzungsverzeichnis

TLS	Terrestrisches Laserscanning
DGM	Digitales Geländemodell
GPS	Global Positioning System
TPS	Terrestrisches Positionierungs System
GLONASS	Global Navigation Satellite System
GNSS	Global Navigation Satellite System
ICP	Iterative Closest Point
SLAM	Simultaneous Localization and Mapping
DBH	Diameter at Breast Height
OPALS	Orientation and Processing of Airborne Laser Scanning Data
SW	Süd-West
NO	Nord-Ost
TU	Technische Universität
ALS	Airborne Laserscanning
UAV	Unmanned Aerial Vehicle
IMU	Inertiale Measurement Unit
ODM	OPALS Data Manager
GUI	Graphical User Interface
ASCII	American Standard Code for Information Interchange
CAD	Computer Aided Design
GeoTIFF	Geo Tagged Image File Format
NaN	Not a Number
XML	Extensible Markup Language
ppm	parts per million
EPSG	European Petroleum Survey Group Geodesy

EPOSA Echtzeit Positionierung Austria

RPM Rounds Per Minute

LTS Long Term Support

BEV Bundesamt für Eich- und Vermessungswesen

