



Carsten Fries  
Martin Schuhfuß

# Realisierung eines 3D-Laserscanners

auf Basis der Lichtschnitttechnik



Dieses Buch befasst sich mit der Realisierung eines dreidimensionalen Laserscanners auf Basis der Lichtschnitttechnik. Es wird die Entwicklung eines hochgenauen Laserscanners, ausgehend von den verwendeten Materialien bis zur Softwarerealisierung und -Implementierung detailliert beschrieben. Der technische Aufbau des Laserscanners besteht aus einer Industriekamera, einem Laser mit einer zylindrischen Linse für die Linienprojektion und einem Drehteller, der von einem präzise positionierbaren Schrittmotor angetrieben wird. Der resultierende Laserscanner wird unter analytischen Kriterien wie Präzision, Geschwindigkeit, Struktur- und Materialeinfluss untersucht. Dabei veranschaulichen zahlreiche Vermessungsergebnisse die Leistungsfähigkeit der Objektvermessung.

#### **Carsten Fries, Martin Schuhfuß**



Carsten Fries und Martin Schuhfuß studierten Informatik an der Hochschule für Angewandte Wissenschaften Hamburg. Dort ist Carsten Fries als wissenschaftlicher Mitarbeiter angestellt und forscht aktiv im Bereich Photogrammetrie und Computer Vision. Martin Schuhfuß arbeitet seit 2007 bei der spot-media AG in Hamburg im Bereich E-Commerce.



9 783639 281538 978-3-639-28153-8



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Stand der Technik . . . . .	1
1.2. Zielsetzung . . . . .	2
1.3. Gliederung . . . . .	4
<b>2. Technologien und Verfahren</b>	<b>5</b>
2.1. Taktile Verfahren . . . . .	5
2.2. Berührungslose Verfahren . . . . .	7
2.2.1. Lichtlaufzeit . . . . .	7
2.2.2. Triangulation . . . . .	7
<b>3. Grundlagen</b>	<b>11</b>
3.1. Mathematische Grundlagen . . . . .	12
3.1.1. Koordinatentransformation . . . . .	12
3.1.2. Bildtransformationen . . . . .	14
3.1.3. Perspektivische Transformation $\mathbb{P}^2 \mapsto \mathbb{P}^2$ . . . . .	15
3.1.4. Lösung linearer Ausgleichsprobleme . . . . .	16
3.1.5. Nichtlineare Ausgleichsrechnung . . . . .	18
3.2. Grundlagen der Bildverarbeitung . . . . .	20
3.2.1. Lineare Filterung mit Faltungskernen . . . . .	20
3.2.2. Kantendetektion . . . . .	20
3.2.3. Hough-Raum und Hough-Transformation . . . . .	23
3.2.4. Bilineare Interpolation . . . . .	24
<b>4. Technischer Aufbau</b>	<b>25</b>
4.1. Gerüst . . . . .	25
4.2. Verkleidung . . . . .	28
4.3. Kamera . . . . .	29
4.4. Laser . . . . .	30

---

4.5. Drehmesstisch . . . . .	31
4.6. Kalibrierungsvorrichtung . . . . .	32
<b>5. Realisierung</b>	<b>35</b>
5.1. Kalibrierung . . . . .	39
5.1.1. Vorbereitung . . . . .	40
5.1.2. Bestimmung der Referenzpunkte in der Bildebene . . . . .	42
5.1.3. Berechnung der Transformationsmatrix . . . . .	49
5.2. Bildaufnahme . . . . .	51
5.3. Bildanalyse . . . . .	53
5.3.1. Vorverarbeitung des Kamerabildes . . . . .	54
5.3.2. Extraktion der Oberflächenpunkte aus dem Bild . . . . .	55
5.3.3. Berechnung der Koordinaten in der Laserebene . . . . .	56
5.4. Rekonstruktion der 3D-Koordinaten . . . . .	57
5.5. Registrierung . . . . .	58
5.6. Vernetzung . . . . .	60
5.7. Modellierung . . . . .	61
5.8. Nachbearbeitung . . . . .	62
5.8.1. Beseitigung von Punktstörungen . . . . .	62
5.8.2. Lochbeseitigung . . . . .	63
5.8.3. Glätten der Oberfläche . . . . .	63
5.8.4. Simplifizierung . . . . .	64
5.8.5. Formatierung der Punktmenge . . . . .	66
<b>6. Implementierung</b>	<b>69</b>
6.1. Betriebssystemunterstützung . . . . .	69
6.2. Grafische Oberfläche . . . . .	70
<b>7. Auswertung</b>	<b>79</b>
7.1. Genauigkeit der Kalibrierung . . . . .	79
7.2. Messgenauigkeit . . . . .	80
7.3. Präzision . . . . .	82
7.4. Geschwindigkeit . . . . .	84
7.5. Struktur- und Materialeinfluss . . . . .	86
7.6. Messergebnisse . . . . .	87
7.6.1. Problematisches Messobjekt . . . . .	87
7.6.2. Komplexe Form . . . . .	88

---

7.6.3. Geometrische Formen . . . . .	89
7.6.4. Weitere Messresultate . . . . .	91
<b>8. Zusammenfassung und Ausblick</b>	<b>95</b>
8.1. Stand der Arbeit . . . . .	95
8.2. Weiterentwicklungen . . . . .	96
<b>A. Verwendete Hilfsmittel</b>	<b>99</b>
A.1. Bibliotheken . . . . .	99
A.2. Bild- und Grafikprogramme . . . . .	99
A.3. Softwaresysteme zur Darstellung und Weiterverarbeitung von Punkt- mengen . . . . .	100
A.4. Weitere Hilfsmittel . . . . .	100
<b>B. Codebeispiele</b>	<b>101</b>
B.1. Interfaces . . . . .	101
B.2. Aufruf der Kalibrierung und Objektvermessung . . . . .	105
B.3. Bildaufnahme . . . . .	107
<b>Abbildungsverzeichnis</b>	<b>109</b>
<b>Tabellenverzeichnis</b>	<b>111</b>
<b>Quellenverzeichnis</b>	<b>113</b>



# 1. Einleitung

## 1.1. Stand der Technik

Die dreidimensionale Objektvermessung dient zur Erfassung von Oberflächenstrukturen und der Bestimmung der räumlichen Geometrie von Körpern. Durch die bis heute rasante technologische Entwicklung ist die 3D-Objektvermessung in vielen Bereichen vertreten.

In der Qualitätskontrolle erfassen Vermessungssysteme Produkte und Werkstücke im Produktionsprozess dreidimensional, um anschließend einen Soll-Ist-Vergleich durchzuführen. Somit können frühzeitig Produktionsfehler, die durch Umwelteinflüsse wie wechselnder Temperatur und Luftdruck oder Maschinenverschleiß entstehen, erkannt und sofort notwendige Korrekturen vorgenommen werden.

Im Bereich der Orthopädie sind Behandlungsverläufe und Therapieergebnisse ohne gesundheitsschädliche Strahlenbelastung messbar. Animierte Bewegungsstudien werden zum Beispiel durchgeführt, um Prothesen vorab zu testen. Somit kann die bestmögliche Passform bei Prothesen, Schuhen oder Kleidung für jede Person erzielt werden.

Die Archäologie rekonstruiert virtuell beschädigte wertvolle Objekte, wie jahrhundertalte Skulpturen oder Ausgrabungsstätten, um ihre ursprüngliche Form zu erhalten. Zahlreiche digitale Kunstkataloge und Objektausstellungen sind in den letzten Jahren durch berührungslose, hochgenaue und schnelle 3D-Objektscanner entstanden.

Dieser kleine Einblick von vielen möglichen Anwendungsgebieten zeigt, dass sich die 3D-Objektvermessung in der Industrie, Fertigung und Forschung etabliert hat. In der Zukunft wird sich die Technologie und damit die Leistungsfähigkeit der technischen Geräte weiterentwickeln, was eine noch größere Verbreitung der 3D-Objektvermessung erwarten lässt.

## 1.2. Zielsetzung

Das Ziel dieser Arbeit ist die Realisierung eines Laserscanners auf Basis der Lichtschnitttechnik, zur 3D-Vermessung von Gegenständen.

Das zu vermessende Objekt soll in den Scanner auf einen per Schrittmotor gesteuerten Drehteller gestellt werden. Während des Messvorgangs wird der Drehteller dann langsam um 360 Grad gedreht, während eine Kamera für jeden Einzelschritt der Drehung ein Bild speichert. Aus den Einzelbildern kann dann ein 3D-Modell des Messobjektes errechnet werden.

Dieses Buch wurde von zwei bestehenden Arbeiten inspiriert. Die eine ermöglicht die Kalibrierung der Systemkomponenten und die Erfassung von Weltkoordinaten einer Schnittebene (Schuhfuss, 2007). In einer zweiten entwickelte ein Diplomand ein passendes Steuerungssystem, mit dem ein Drehteller über einen Schrittmotor und einer Positioniersteuerung ansteuerbar ist (Reimann, 2008).

Im Rahmen dieser Arbeit soll ein Gesamtsystem so entwickelt werden, dass alle zur Vermessung notwendigen Schritte über eine grafische Benutzeroberfläche bedienbar werden:

- Der Anwender ist schrittweise durch den Kalibrierungsvorgang zu geleiten.
- Nachdem die Systemkomponenten aufeinander kalibriert sind, sollen Gegenstände, die in ihrer Oberflächenstruktur oder deren Körper zu bestimmen sind, einzeln in einer abgedunkelten Umgebung auf einem per Schrittmotor ansteuerbaren Drehteller platziert werden können, um nachfolgend über eine Software den Messvorgang zu steuern. Die ermittelten Weltkoordinaten der Schnittebenen sollen im Messvorgang zu 3D-Weltkoordinaten umgerechnet werden können, sodass nach Vollendung des Messvorgangs eine Punktwolke entsteht.
- Mit Hilfe zwei zu integrierende Programme sind Punktwolken in flächenbasierte Modelle überzuführen und darzustellen.

- 
- Ein textuell und grafisch dokumentierter Sourcecode des Messsystems ist in die grafische Benutzeroberfläche einzubinden. Im Laufe der Entwicklung entstandene Messergebnisse sind zur Veranschaulichung der Leistungsfähigkeit des Laserscanners in einer gesonderten Ansicht darzustellen.
  - Bei dem resultierenden Laserscannersystem ist besonders darauf zu achten, dass soft- und hardwareseitig der Licht- und Materialeinfluss minimal zu halten, und die Geschwindigkeit zu maximieren ist.

## 1.3. Gliederung

Die vorliegende Arbeit ist in 8 Kapitel gegliedert. Diese Kapitel haben folgende Inhalte:

**Kapitel 1** beschreibt den heutigen Stand der dreidimensionalen Objektvermessung und erläutert das Ziel dieser Arbeit.

**Kapitel 2** gibt einen Überblick über existierende Technologien und Verfahren.

**Kapitel 3** stellt für das notwendige Verständnis dieser Arbeit Mathematik- und Bildverarbeitungsgrundlagen vor.

**Kapitel 4** beschreibt den technischen Aufbau des Laserscanners.

**Kapitel 5** beschreibt detailliert die einzelnen Phasen zur Ermittlung dreidimensionaler Punktwolken und deren Weiterverarbeitung.

**Kapitel 6** erläutert die Bedienung der entwickelten Software zur Steuerung und Handhabung des Laserscanner-Systems. Ebenso werden die Voraussetzungen der Softwarenutzung beschrieben.

**Kapitel 7** bewertet die Messergebnisse des Laserscanners in Bezug auf Präzision, Geschwindigkeit, Struktur- und Materialeinfluss.

**Kapitel 8** fasst die Resultate dieser Arbeit zusammen und stellt mögliche Weiterentwicklungen vor.

## 2. Technologien und Verfahren

Für die räumliche Abtastung von Objekten zur dreidimensionalen digitalen Rekonstruktion ist eine Vielzahl von Messprinzipien bekannt. Diese Messprinzipien sind in taktile- und berührungslose Verfahren unterteilt. Nachfolgend werden typische und das in dieser Arbeit angewendete Verfahren erläutert.

### Inhalt

---

2.1. Taktile Verfahren . . . . .	5
2.2. Berührungslose Verfahren . . . . .	7
2.2.1. Lichtlaufzeit . . . . .	7
2.2.2. Triangulation . . . . .	7
Punkttriangulation . . . . .	8
Lichtschnitttechnik . . . . .	8
Codierter Lichtansatz . . . . .	8

---

### 2.1. Taktile Verfahren

Mit den taktilen Verfahren werden Messungen mit einem sehr geringen Verlust an Genauigkeit vorgenommen. Zur oberflächigen taktilen Vermessung wird eine Koordinatenmessmaschine in der industriellen Qualitäts- und Fertigungskontrolle eingesetzt. Die Vermessung findet mit einem präzise gefertigten Tastkopf statt, der die Oberflächenstruktur des Objektes schrittweise abtastet. Nach jeder Berührung des Tastkopfes mit dem Objekt wird eine Messung angestoßen, welche die aktuelle Position ermittelt und speichert. Nachdem die Oberflächenstruktur durch die gemessenen Positionswerte ermittelt wurde, kann die Form rekonstruiert werden. Der Messbereich ist dabei in X-, Y- und Z-Richtung bis zu mehreren Metern möglich. Dabei werden Genauigkeiten von  $0,1 \mu m$  erzielt. Da der Tastkopf das Objekt bei der Vermessung

berührt, ist das Risiko der Beschädigung oder gar Zerstörung von empfindlichen Objekten gegeben. Der Kaufpreis von taktilen Messinstrumenten ist sehr hoch. Des Weiteren benötigt das Verfahren viel Zeit bei der Erfassung von komplexen Objekten.

Aufgrund der genannten Nachteile ist dieses Verfahren nicht für den Anwendungsbereich dieser Arbeit geeignet. Daher wird nicht weiter auf dieses Verfahren eingegangen.

## 2.2. Berührungslose Verfahren

Berührungslose Messverfahren haben in den letzten Jahren immer mehr an Bedeutung gewonnen. Ein Objekt zeiteffizient und zerstörungsinvariant vermessen zu können, ohne es zu berühren, führte zu zahlreichen Verfahren. In diesem Unterkapitel wird auf die aktiven Messmethoden der berührungslosen Verfahren eingegangen. Passive Messmethoden benötigen im Gegensatz zu aktiven Messmethoden keine zusätzliche Lichtquelle und greifen daher nicht in den zu vermessenden Bereich ein. Mehr Informationen über passive Verfahren finden sich in (Gühring, 2001) [S.14-17].

### 2.2.1. Lichtlaufzeit

Bei dem zur Abstandsmessung eingesetztem Lichtlaufzeit-Verfahren wird die Laufzeit (engl. „Time-of-Flight“) eines Laserlichtimpulses zum Objekt und zurück gemessen. Das Verfahren benötigt einen Lichtsensor und eine Lichtquelle. Die Abstandsmessung beginnt mit der Aussendung eines Lichtimpulses der Lichtquelle in Richtung des Objektes. Dieses reflektiert den Lichtimpuls und ermöglicht dadurch die Aufnahme des ausgesandten Lichtimpulses durch den Lichtsensor. Mit der gegebenen Ausbreitungsgeschwindigkeit des Signals (Lichtgeschwindigkeit) und der gemessenen Dauer des Lichtimpulses, kann der Abstand zwischen dem Lichtsensor und dem Objekt ermittelt werden, indem die Hälfte der gemessenen Dauer mit der Ausbreitungsgeschwindigkeit des Signals multipliziert wird. Mit diesem Messverfahren sind in Abhängigkeit der störungsfreien Lichtimpulsübertragung und der Reflexionsverhältnisse, hohe Genauigkeiten über mehrere Kilometer möglich. Für Präzisionsmessungen im Nahbereich ist das Lichtlaufzeit-Verfahren eher nicht geeignet.

### 2.2.2. Triangulation

Bei den Triangulationsverfahren stehen eine Lichtquelle, ein elektronischer Bildwandler und das zu vermessende Objekt in einem Dreieck mit den zugehörigen drei Triangulationswinkeln zueinander.

Während die Lichtquelle einen Punkt-, eine Linie- oder ein Lichtmuster auf das Objekt projiziert, wird die beleuchtete Oberfläche von einem elektronischen Bildwandler,

zumeist eine CMOS-, CCD-Kamera oder ein PSD erfasst. PSD steht für „Position Sensitive Device“ und ist ein Sensor welcher zur ein- oder zweidimensionalen Ortsbestimmung eines Lichtpunktes verwendet wird. Die Position des Messpunktes wird anhand einer einmaligen Kalibrierung der Lichtquelle mit dem elektronischen Bildwandler ermittelt. Laserlicht bietet den großen Vorteil der Verwendung sehr dünner Lichtpunkte bzw. Lichtstreifen. In Kombination mit einem hochauflösenden elektronischen Bildwandler, können somit Genauigkeiten von  $0,01\text{ mm}$  erzielt werden.

Nachfolgend werden drei typische Triangulationsverfahren, die sich in der Lichtquelle unterscheiden, erläutert.

### **Punkttriangulation**

Das Punkttriangulationsverfahren benutzt als Lichtquelle einen Punktlaser, sodass nur ein Messpunkt pro Zeiteinheit, erfasst werden kann. Aufgrund der hohen Zeitintensität bei einigen tausend Messpunkten, wurden das Lichtschnittverfahren und der codierte Lichtschnittansatz entwickelt.

### **Lichtschnitttechnik**

In dieser Arbeit wurde das Lichtschnittverfahren angewendet, welches es ermöglicht, mit einem Linienlaser eine gesamte Linie und damit einen Lichtschnitt des Objektes gleichzeitig zu vermessen. Dies erhöht die Geschwindigkeit enorm.

Eine Weiterentwicklung des Lichtschnittverfahrens ist die gleichzeitige Projektion von mehreren Lichtstreifen. Dies ermöglicht die Erfassung des gesamten Objektes bzw. der maximal möglichen Größe des Messbereiches und reduziert damit die Anzahl der aufzunehmenden Bilder. Die Schwierigkeit bei dieser Erweiterung liegt jedoch in der eindeutigen Zuordnung der projizierten Lichtstreifen im Kamerabild.

Die Fortentwicklung dieses Gedankens führt zum codierten Lichtschnittverfahren.

### **Codierter Lichtansatz**

Beim „Codierten Lichtansatz“ wird ein binär-, farbcodiertes oder Moiré-Muster mit einem Lichtmusterprojektor flächenhaft auf das Objekt projiziert. Das Problem bei einfachen Lichtmustern ist die Verdeckung von projiziertem Licht, sodass eine eindeutige

Identifizierung und Indizierung zwischen projiziertem und im Kamerabild sichtbarem Laserlicht nicht vollständig gegeben ist. Dieses Problem wird behoben, indem ein für das System bekanntes kodierte Muster auf das zu vermessene Objekt projiziert wird.

Aus Gründen der Fehlertoleranz wird beim binärcodiertem Ansatz vorzugsweise der Graycode in der Praxis verwendet.

Nähere Informationen über den codierten Lichtansatz finden sich in Pagés u. a. (2003), Zhang u. a. (2002) und Bunke und Jiang (1997).



# 3. Grundlagen

Da sich diese Arbeit sehr viel mit den Bereichen der linearen Algebra und der digitalen Bildverarbeitung beschäftigt, soll in diesem Kapitel auf die Grundlagen und verwendeten Verfahren etwas näher eingegangen werden.

## Inhalt

---

3.1. Mathematische Grundlagen . . . . .	12
3.1.1. Koordinatentransformation . . . . .	12
Translation . . . . .	12
Rotation . . . . .	13
3.1.2. Bildtransformationen . . . . .	14
Affine Transformationen . . . . .	14
3.1.3. Perspektivische Transformation $\mathbb{P}^2 \mapsto \mathbb{P}^2$ . . . . .	15
3.1.4. Lösung linearer Ausgleichsprobleme . . . . .	16
QR-Zerlegung . . . . .	17
Singularwert-Zerlegung . . . . .	18
3.1.5. Nichtlineare Ausgleichsrechnung . . . . .	18
3.2. Grundlagen der Bildverarbeitung . . . . .	20
3.2.1. Lineare Filterung mit Faltungskernen . . . . .	20
3.2.2. Kantendetektion . . . . .	20
Canny-Algorithmus . . . . .	22
3.2.3. Hough-Raum und Hough-Transformation . . . . .	23
3.2.4. Bilineare Interpolation . . . . .	24

---

## 3.1. Mathematische Grundlagen

### 3.1.1. Koordinatentransformation

Mit einer Koordinatentransformation können Koordinaten bezüglich eines Koordinatensystems verschoben, rotiert und skaliert werden. In diesem Unterkapitel wird auf die Translation und Rotation im 2D und 3D-Raum eingegangen.

#### Translation

Die Translation im zweidimensionalen Raum ermöglicht die Verschiebung von Koordinaten in zwei Richtungen auf eine andere Raumposition. Bei einem Koordinatensystem mit einer X und Y-Achse, kann positiv und negativ in diese beiden Richtungen verschoben werden. Daraus ergibt sich, dass ein Punkt  $P_q$  mit den Koordinaten  $(x_q, y_q)$  auf die Zielposition  $P_z(x_z, y_z)$  abgebildet wird, indem der positive bzw. negative Versatz  $\Delta x, \Delta y$  auf den Punkt  $P_q$  addiert wird. Es lassen sich damit die Translationsgleichungen

$$\begin{aligned}x_z &= x_q + \Delta x \\y_z &= y_q + \Delta y\end{aligned}\tag{3.1}$$

und die Translationsmatrix

$$\begin{bmatrix} x_z \\ y_z \end{bmatrix} = \begin{bmatrix} x_q \\ y_q \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}\tag{3.2}$$

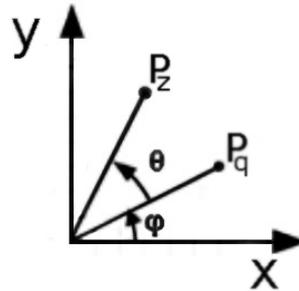
aufstellen.

Für den 3D-Raum ist das Koordinatensystem um die Z-Achse erweitert und damit ändert sich die Translationsmatrix zu

$$\begin{bmatrix} x_z \\ y_z \\ z_z \end{bmatrix} = \begin{bmatrix} x_q \\ y_q \\ z_q \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}\tag{3.3}$$

## Rotation

Bei einer Rotation im zweidimensionalen Raum werden das Koordinatensystem bzw. die zu drehenden Objektpunkte rotiert. Nachfolgend werden die Rotationsgleichungen hergeleitet.



**Abbildung 3.1.:** Der trigonometrische Zusammenhang einer Rotation.

In einem Koordinatensystem mit einer x- und y-Achse, soll ein Punkt  $P_q$  auf den Punkt  $P_z$  gedreht werden. Der Punkt  $P_q$  ist in Polarkoordinaten beschrieben und hat somit einen Ursprungsabstand  $r$  und einen Winkel  $\varphi$ . Dabei ist der Winkel  $\varphi$  zwischen der x-Achse und der Linie, die den Ursprungsabstand darstellt, aufgespannt. Dadurch ergeben sich die Polarkoordinaten für den Punkt  $P_q$

$$\begin{aligned}x_q &= r \cdot \cos \varphi \\y_q &= r \cdot \sin \varphi\end{aligned}\tag{3.4}$$

Zur Ermittlung der Zielpunktcoordinate  $P_z(r, \varphi + \theta)$ , die um den Rotationswinkel  $\theta$  zusätzlich gedreht ist, lassen sich mit den Additionstheoremen

$$\begin{aligned}\cos(\varphi + \theta) &= \cos \varphi \cdot \cos \theta - \sin \varphi \cdot \sin \theta \\ \sin(\varphi + \theta) &= \cos \varphi \cdot \sin \theta + \sin \varphi \cdot \cos \theta\end{aligned}\tag{3.5}$$

und unter Verwendung von  $x_q = r \cdot \cos \varphi$  und  $y_q = r \cdot \sin \varphi$  die gesuchten Rotationsgleichungen

$$\begin{aligned}x_z &= x_q \cdot \cos \theta - y_q \cdot \sin \theta \\ y_z &= x_q \cdot \sin \theta + y_q \cdot \cos \theta\end{aligned}\tag{3.6}$$

herleiten.

Bei einer Rotation ist darauf zu achten, dass die Drehung um den Koordinatenursprung und nicht um die Objektmitte stattfindet. Deshalb wird üblicherweise zuerst das Objekt in das Rotationszentrum mit einer Translation  $\underline{T}^{-1}$  verschoben. Anschließend erfolgt mit einer Rotationsmatrix  $\underline{R}$  die Drehung. Nach der Rotation muss der Punkt mit der Translation  $\underline{T}$  zurückverschoben werden. Insgesamt ergibt sich eine von rechts nach links zu lesende Matrix  $\underline{M} = \underline{T} \cdot \underline{R} \cdot \underline{T}^{-1}$ , die aus zwei Matrizenmultiplikationen zusammengesetzt ist.

Die Rotation im 3D-Raum unterscheidet sich vom zweidimensionalen Raum darin, dass jeweils um die x-, y- und z-Koordinatenachse gedreht werden kann. Es ergeben sich damit die drei Rotationsmatrizen:

$$\underline{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (3.7)$$

$$\underline{R}_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3.8)$$

$$\underline{R}_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

### 3.1.2. Bildtransformationen

#### Affine Transformationen

In der Geometrie bezeichnen affine Transformationen alle auf einen Punkt bzw. einer Menge von Punkten anwendbaren geometrischen Operationen wie Rotation, Skalierung, Scherung, Spiegelung und Verschiebung. Alle diese Operationen haben gemeinsam, dass die Abbildungen von 3 Punkten, die auf einer Geraden liegen ihrerseits auch kollinear sind. Des Weiteren sind auch die Abbildungen von parallelen Geraden ebenfalls parallel. Mathematisch lassen sich affine Transformationen als eine

lineare Transformation  $\mathbf{T}$  gefolgt von einer Verschiebung  $\vec{b}$  beschreiben.

$$\vec{a} \mapsto \mathbf{T}\vec{a} + \vec{b} = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \end{pmatrix} \quad (3.10)$$

Um alle affinen Transformationen (inklusive der Verschiebung) in einer einzigen Transformationsmatrix abbilden zu können, werden die Vektoren um eine dritte Komponente erweitert, die in diesem Fall konstant gleich eins ist. Dadurch lassen sich die affinen Transformationen vereinfacht darstellen mit

$$\vec{a} \mapsto \mathbf{T}\vec{a} = \begin{pmatrix} t_{11} & t_{12} & b_x \\ t_{21} & t_{22} & b_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ 1 \end{pmatrix} \quad (3.11)$$

### 3.1.3. Perspektivische Transformation $\mathbb{P}^2 \mapsto \mathbb{P}^2$

Mit der perspektivischen Abbildung können Translationen, Rotationen, Skalierungen, Scherungen, Spiegelungen und projektive Verzerrungen vorgenommen werden. Dabei ist die Aufgabe einer perspektivischen Transformation  $\underline{H}$

$$\underline{H} \cdot \vec{P} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \longrightarrow \vec{P}^* = \begin{bmatrix} x^* \\ y^* \end{bmatrix} \quad (3.12)$$

die Abbildung eines Punktes der Form  $\vec{P} = (x, y, 1)^T$  auf den zugehörigen projizierten Punkt  $\vec{P}^* = (x^*, y^*)^T$  der zweidimensionalen Bildebene einer Kamera. Hierbei ist  $\vec{P}$  in homogenen Koordinaten und  $\vec{P}^*$  in euklidischen Koordinaten dargestellt. Die Rückwandlung von homogenen Koordinaten in euklidische Koordinaten wird ermöglicht, indem durch den dritten Wert des Multiplikationsergebnisses von  $\underline{H} \cdot \vec{P}$  dividiert wird:

$$x^* = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \quad (3.13)$$

$$y^* = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \quad (3.14)$$

Die Koeffizienten  $h_{ij}$  aus der Abbildungsmatrix  $\underline{H}$  sind von den so genannten intrinsischen und extrinsischen Kameraparametern abhängig. Zusätzlich ist zu beachten, dass diese Abbildungsart parallele Geraden nicht auf parallele Geraden abbildet und somit nicht paralleleerhaltend ist. Geraden werden jedoch immer auf Geraden abgebildet.

Die intrinsischen Parameter beschreiben den Zusammenhang des Kamera- und Bildkoordinatensystems. Zu den intrinsischen Parametern gehört die interne Geometrie der Kamera und damit Eigenschaften wie Brennweite, Pixelkoordinaten der Bildmitte, sowie der Skalierungsfaktor.

Die extrinsischen Parameter beschreiben bezüglich eines Weltkoordinatensystems die Position und Ausrichtung der Kamera im Raum und damit die äußere Orientierung der Kamera.

Zur Ermittlung der Zielposition von gemessenen Bildkoordinaten ermöglicht die inverse perspektivische Transformation, die Berechnung der Koordinaten in der Streifenebene.

Weitere und ausführlichere Informationen zu diesem Thema finden sich in (Rogers und Adams, 1976), (Hartley und Zisserman, 2000) oder (Meisel, 2008a).

### 3.1.4. Lösung linearer Ausgleichsprobleme

Lineare Ausgleichsprobleme treten überall dort auf, wo die Parameter eines mathematischen Modells aus einer Reihe von Messwerten bestimmt werden müssen. Da Messungen in der Praxis immer mit Messfehlern behaftet sind, lassen sich die Parameter des Modells nicht direkt aus den Messwerten bestimmen. Daher ist es notwendig, ein Verfahren anzuwenden mit dem die bestmögliche Näherung an die tatsächlichen Werte der Modell-Parameter ermittelt werden kann.

Durch Einsetzen der einzelnen Messwerte in die Modellgleichung entsteht ein Gleichungssystem mit ebenso vielen Gleichungen wie Messwerten. Dieses Gleichungssystem lässt sich in Matrixschreibweise als  $\mathbf{M}\vec{x} = \vec{b}$  darstellen, wobei  $\vec{x}$  der Vektor der gesuchten Modell-Parameter ist und  $\mathbf{M}$  und  $\vec{b}$  die Messwerte enthalten. Dieses lineare Gleichungssystem ist in der Regel nicht eindeutig lösbar. Eine eindeutige Lösung existiert genau dann nicht, wenn  $\mathbf{M} \in \mathbb{R}^{m \times n}$  eine Matrix mit  $m > n$  ist und  $\mathbf{M}$  keine linear abhängigen Zeilenvektoren  $\vec{m}_i$  enthält. Weniger formal ausgedrückt, existiert

dann keine eindeutige Lösung, wenn das zugehörige Gleichungssystem mehr Gleichungen als Unbekannte enthält.

Die bestmögliche Näherung dieses linearen Ausgleichsproblems ist durch denjenigen Vektor  $\vec{x}$  gegeben, für den die Summe der Fehlerquadrate

$$\sum_{i=1}^m \varepsilon_i^2 = |\vec{\varepsilon}|^2 = |\mathbf{M}\vec{x} - \vec{b}|^2 \quad (3.15)$$

minimal ist.

Es lässt sich zeigen, dass ein solches lineares Ausgleichsproblem immer eindeutig lösbar ist. Die Beweisführung und weitergehende Erklärungen hierzu finden sich u.a. in (Mackens und Voss, 1993). Die Lösung ist dann gegeben durch

$$\vec{x} = \mathbf{M}^\dagger \vec{b} \quad \text{mit } \mathbf{M}^\dagger = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \quad (3.16)$$

Die Matrix  $\mathbf{M}^\dagger$  wird verallgemeinerte Inverse (auch Pseudoinverse oder Moore-Penrose-Inverse) der Matrix  $\mathbf{M}$  genannt.

In dieser Arbeit wurden zwei unterschiedliche Verfahren zur Lösung der linearen Ausgleichsprobleme eingesetzt, die im Folgenden kurz vorgestellt werden sollen. Beschreibungen der Einsatzgebiete finden sich in Kapitel 5 in den Abschnitten 5.1.2 auf S.44 und 5.1.3 auf S.49

## QR-Zerlegung

Bei der QR-Zerlegung wird die Matrix  $\mathbf{M} \in \mathbb{R}^{m \times n}$  in zwei Matrizen  $\mathbf{Q}$  und  $\mathbf{R}$  zerlegt, so dass  $\mathbf{M} = \mathbf{Q}\mathbf{R}$  gilt. Hierbei ist  $\mathbf{Q}$  eine orthogonale Matrix und  $\mathbf{R}$  eine obere Dreiecksmatrix. Orthogonale Matrizen sind Matrizen, deren Inverse gleich ihrer Transponierten ist. Daher gilt  $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$ . Zur Berechnung der QR-Zerlegung wird die Householder-Transformation verwendet. Bei der Householder-Transformation handelt es sich um ein numerisches Verfahren zur Berechnung der QR-Zerlegung einer Matrix. Details über den Algorithmus und dessen Implementierung finden sich in (Press u. a., 1992).

Es ergibt sich damit für das lineare Ausgleichsproblem  $\mathbf{M}\vec{x} = \vec{b}$

$$\mathbf{Q}\mathbf{R}\vec{x} = \vec{b} \quad \Leftrightarrow \quad \mathbf{R}\vec{x} = \mathbf{Q}^T \vec{b} \quad (3.17)$$

Da es sich bei  $\mathbf{R}$  um eine obere Dreiecksmatrix handelt, lässt sich dieses Gleichungssystem nun durch Rückwärtseinsetzen (Gauss-Elimination) nach  $\vec{x}$  auflösen.

## Singulärwert-Zerlegung

In Gleichung (3.16) wurde gezeigt, dass es durch Verwendung der Pseudoinversen eine allgemeine Lösung für das lineare Ausgleichsproblem gibt. Mit der Singulärwertzerlegung (*engl. Singular Value Decomposition*, kurz SVD) existiert ein zwar aufwändiges aber auch numerisch sehr stabiles Verfahren, mit dem die Pseudoinverse  $\mathbf{M}^\dagger$  berechnet werden kann.

Jede Matrix  $\mathbf{M} \in \mathbb{R}^{m \times n}$  kann in drei Matrizen  $\mathbf{U} \in \mathbb{R}^{m \times m}$ ,  $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$  und  $\mathbf{V} \in \mathbb{R}^{n \times n}$  zerlegt werden, so dass gilt:

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (3.18)$$

Hierbei sind die Matrizen  $\mathbf{U}$  und  $\mathbf{V}$  orthogonale Matrizen, die Matrix  $\mathbf{\Sigma}$  ist eine Diagonalmatrix und enthält die singulären Werte von  $\mathbf{M}$  als Diagonalelemente. Die Pseudoinverse  $\mathbf{M}^\dagger$  der Matrix  $\mathbf{M}$  lässt sich nun berechnen durch

$$\mathbf{M}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^T \quad (3.19)$$

In dieser Gleichung ist  $\mathbf{\Sigma}^\dagger$  ebenfalls eine Diagonalmatrix, deren Diagonalelemente die Kehrwerte der Diagonalelemente aus  $\mathbf{\Sigma}$  sind, also  $\sigma_{ii}^\dagger = \frac{1}{\sigma_{ii}}$ .

Basierend auf den singulären Werten einer Matrix lässt sich auch die Konditionszahl  $\kappa$  einer Matrix als das Verhältnis des größten zu dem kleinsten singulären Wert definieren. Diese Zahl gibt Aufschluss darüber wie empfindlich die Unbekannte  $\vec{x}$  des linearen Gleichungssystems auf kleine Veränderungen der Werte in  $\mathbf{M}$  bzw.  $\vec{b}$  reagiert. Bei einer großen Konditionszahl wird von einem schlecht konditionierten Gleichungssystem gesprochen, bei kleiner Konditionszahl von einem gut konditionierten. Weitere Informationen zur Singulärwertzerlegung und ihrer Anwendung finden sich in (Mackens und Voss, 1993), (Gramlich, 2004), (Jackèl, 2005) und (Weisstein, 2006). Die Implementierung des SVD-Algorithmus wird in (Press u. a., 1992) näher beschrieben.

### 3.1.5. Nichtlineare Ausgleichsrechnung

Die nichtlineare Ausgleichsrechnung wird eingesetzt, wenn mehr nichtlinear zusammenhängende Messwerte  $y_i$  als unbekannte Parameter  $x_1, \dots, x_n$  eines Modells vorliegen und damit keine eindeutige Lösung der unbekannt Parameter berechnet

werden kann. Zur Lösung des Ausgleichsproblems wird versucht, mit den möglichst kleinsten Abweichungen, eine gegebene nichtlineare Modellfunktion in die vorhandenen Messwerte einzupassen.

Das in dieser Arbeit verwendete numerische Lösungsverfahren ist der Gauß-Newton-Algorithmus. Im Wesentlichen wird versucht die unbekannt Parameter so an die gesuchte Modellfunktion anzupassen, dass die Summe der Fehlerquadrate minimal ist. Im Falle einer Funktion  $f(x_1, \dots, x_n)$  mit  $n$  unbekannt Parametern gilt:

$$\sum_{i=1}^m |f(x_i) - y_i|^2 \rightarrow \min \quad (x_* \in \mathfrak{R}^n, m > n) \quad (3.20)$$

Mit Hilfe der Gauß-Newton-Iteration

$$|f(x) + f'(x) \cdot \Delta x| = 0 \quad (x \leftarrow x + \Delta x) \quad (3.21)$$

kann das nichtlineare Ausgleichsproblem iterativ bestimmt werden. Somit wird wie beim Newton-Verfahren, die Funktion in jedem Iterationsschritt durch eine lineare Näherung ersetzt. Mit einem Standardverfahren kann das dabei entstehende lineare Ausgleichsproblem gelöst werden.

Eine detaillierte Beschreibung des Gauß-Newton-Verfahrens ist in (Überhuber, 1995) [S.226-246 und S.339-341] beschrieben.

## 3.2. Grundlagen der Bildverarbeitung

In der Implementierung zu dieser Arbeit wurden zahlreiche bekannte Verfahren der Bildverarbeitung eingesetzt. Die wichtigsten dieser Verfahren sollen nun ein wenig näher betrachtet werden.

### 3.2.1. Lineare Filterung mit Faltungskernen

Viele Bildverarbeitungsverfahren wie z.B. die unterschiedlichen Verfahren zur Kantendetektion (siehe Abschnitt 3.2.2) oder das Weichzeichnen eines Bildes lassen sich mithilfe von Faltungs- oder Filterkernen und dem Faltungsoperator realisieren.

Der Grauwert eines jeden Pixels im gefilterten Bild ergibt sich hierbei aus einer gewichteten Summe der Grauwerte benachbarter Pixel im Quellbild. Welche benachbarten Pixel mit welcher Gewichtung in diese Summe mit eingehen, wird mit dem Faltungskern angegeben.

Mathematisch ausgedrückt ergibt sich der Grauwert  $f^*(x, y)$  eines gefilterten Bildes  $f^*$  aus den Grauwerten des Quellbildes  $f$  mit gleichen Dimensionen und einem Faltungskern  $k$  der Größe  $m \times n$  (wobei  $m$  und  $n$  ungerade Zahlen sind) durch die Gleichung (Gonzales und Woods, 2002):

$$f^*(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b k(s, t) f(x + s, y + t) \quad \text{mit} \begin{cases} a = \frac{m-1}{2} \\ b = \frac{n-1}{2} \end{cases} \quad (3.22)$$

### 3.2.2. Kantendetektion

Ein sehr häufig im Rahmen der digitalen Bildverarbeitung eingesetztes Verfahren ist die Kantendetektion. Im Wesentlichen geht es dabei darum, die Grauwertübergänge (die Kanten) in einem Bild hervorzuheben und homogene Bereiche, also Bereiche mit ähnlichen Farb- bzw. Grauwerten, zu unterdrücken. Im Folgenden wird lediglich das Verfahren der Kantenerkennung in Grauwertbildern näher erläutert. In Farbbildern werden die Gradienten entweder komponentenweise oder über die Formulierung des

Farbkontrastes berechnet. Das Verfahren hierzu wird in (Gonzales und Woods, 2002), s.335 ff. näher erläutert.

Die Kontinuität eines Bildes in einem bestimmten Bereich kann mithilfe der Gradienten des Bildes untersucht werden. Der Gradient  $\nabla f$  einer Funktion mit mehreren Variablen ist ein Vektor, dessen Komponenten die partiellen Ableitungen der Funktion enthalten:

$$\nabla f(x,y) = \left( G_x \quad G_y \right)^T = \left( \frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right)^T$$

Um diese zu bestimmen, werden Faltungskerne (vgl. Abschnitt 3.2.1) eingesetzt, mit deren Hilfe die diskreten partiellen Ableitungen der Grauwertfunktion des Bildes berechnet werden können. Ein Beispiel für einen solchen Satz von Faltungskernen sind die Sobel-Faltungskerne, die in Abbildung 3.2 dargestellt sind.

-1	-2	-1
0	0	0
1	2	1

(a)

1	0	-1
2	0	-2
1	0	-1

(b)

**Abbildung 3.2.:** Sobel-Faltungskerne: (a) zeigt den Faltungskern für die Gradienten in  $x$ -Richtung, (b) den für die Gradienten in  $y$ -Richtung

Das Ergebnis der Anwendung dieser Faltungskerne sind 2 Grauwertbilder  $G_x$  und  $G_y$ , die die erste bzw. zweite Komponente der Gradienten darstellen. Hieraus lassen sich nun die Gradientenbeträge  $G$  sowie die Richtung der Gradienten  $\alpha$  bestimmen:

$$G = \sqrt{G_x^2 + G_y^2} \quad (3.23a)$$

und

$$\alpha = \arctan \frac{G_y}{G_x} \quad (3.23b)$$

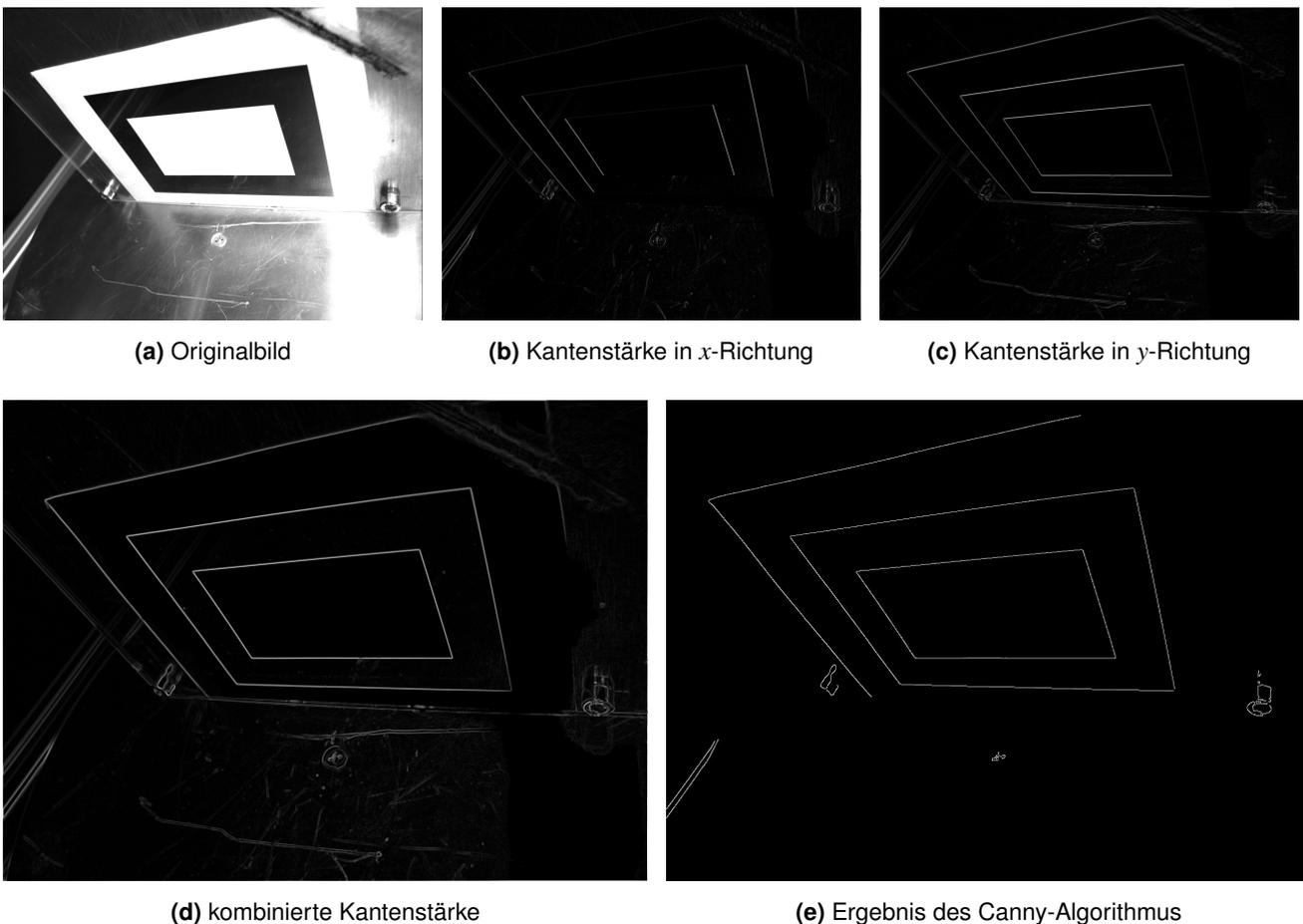
Es ist zu beachten, dass es sich bei  $G$ ,  $G_x$ ,  $G_y$  und  $\alpha$  jeweils um Funktionen mit den Parametern  $x$  und  $y$  handelt.

Die Abbildungen 3.3a–e auf S. 22 zeigen ein Beispiel für die Kantenerkennung mithilfe der Sobel-Faltungskerne.

## Canny-Algorithmus

In dieser Arbeit wurde für die Kantenerkennung der Canny-Algorithmus eingesetzt. Bei diesem Algorithmus handelt es sich um ein auf den Sobel-Faltungskernen aufbauendes Verfahren, welches gegenüber der direkten Verwendung des Gradientenbildes einige Vorteile bietet:

- Da das Quellbild vor der Berechnung der Gradienten mit einem gaußschen Weichzeichner mit einstellbarer Größe und Varianz geglättet wird, lässt sich über diese beiden Parameter der gewünschte Detaillierungsgrad des Ergebnisses steuern.
- Alle Kanten innerhalb des Bildes werden auf eine Breite von einem Pixel ausgedünnt.
- Das Kantenbild ist ein Binärbild.



**Abbildung 3.3.:** Beispiel zur Kantendetektion.

### 3.2.3. Hough-Raum und Hough-Transformation

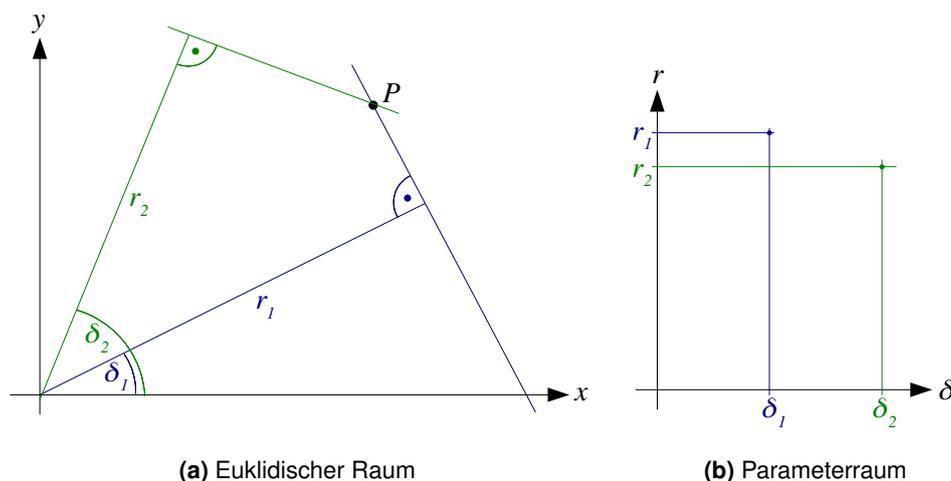
In einem Bild die darin enthaltenen Geraden zu finden, ist ein häufig zu lösendes Problem in der Bildverarbeitung. Ein Verfahren, mit denen dieses Problem gelöst werden kann ist die Hough-Transformation.

Jede Gerade im euklidischen Raum lässt sich durch die zwei Parameter  $\delta$  und  $r$  ihrer normierten Normalen beschreiben. Der Parameter  $\delta$  gibt hierbei den Winkel der Normalen zur  $x$ -Achse des Koordinatensystems und  $r$  die Länge der Normalen an.

Bei der Hough-Transformation handelt es sich um ein Verfahren, mit dem ein Bild auf einen Parameterraum („*Hough-Raum*“) mit den beiden Parametern  $\delta$  und  $r$  abgebildet wird. Hierzu wird meistens ein binäres Kantenbild verwendet, wie es zum Beispiel durch den Canny-Algorithmus (vgl. Abschnitt 3.2.2) erzeugt wird.

Bei der Abbildung existiert zu jedem Punkt im Bild eine bestimmte Anzahl von Punkten im Parameterraum, durch die Geraden beschrieben werden, die durch diesen Punkt verlaufen. Da der Parameterraum die beiden Parameter  $\delta$  und  $r$  nicht kontinuierlich, sondern diskret erfasst, ist die Anzahl der Punkte im Parameterraum begrenzt.

Die Transformation wird in Abbildung 3.4 veranschaulicht.



**Abbildung 3.4.:** Veranschaulichung der Hough-Transformation: der Punkt  $P$  lässt sich durch eine unendliche Anzahl von hypothetischen Geraden beschreiben, von denen hier zwei dargestellt sind. Jeder Punkt im Parameterraum (dem Hough-Raum) beschreibt eine dieser hypothetischen Geraden.

Für jeden Bildpunkt, dessen Intensitätswert über einem anzugebenden Schwellwert liegt, werden nun im Parameterraum die Werte aller Punkte inkrementiert, durch die Geraden beschrieben werden, die durch diesen Bildpunkt verlaufen könnten.

Wurden alle Pixel des Bildes bearbeitet, lassen sich die in dem Bild vorhandenen Geraden aus dem Parameterraum ablesen. Hierzu werden innerhalb des Parameterraums alle lokalen Maxima, deren Wert größer als ein angegebener Schwellwert (dieser beschreibt die Mindestanzahl von Bildpunkten, die auf der durch den Punkt im Parameterraum beschriebenen Geraden liegen und damit die Mindestlänge der Geraden) ist, gesucht.

### 3.2.4. Bilineare Interpolation

Interpolationsverfahren werden in der Bildverarbeitung eingesetzt, um Farb- bzw. Grauwerte an Positionen zwischen einzelnen Pixeln zu ermitteln. Bei der bilinearen Interpolation wird hierzu ein gewichteter Mittelwert der nächsten vier benachbarten Pixel gebildet. Für den interpolierten Grauwert  $f(x_s, y_s)$  im Punkt  $(x_s, y_s)$  erhält man somit:

$$\begin{aligned} f(x_s, y_s) &= f(x_0, y_0) (1 - x_s)(1 - y_s) + f(x_0 + 1, y_0) x_s(1 - y_s) \\ &+ f(x_0, y_0 + 1) (1 - x_s)y_s + f(x_0 + 1, y_0 + 1) x_s y_s \end{aligned} \quad (3.24)$$

# 4. Technischer Aufbau

In diesem Kapitel wird auf die benötigten Baumaterialien und Hardwarekomponenten zur Realisierung eines vollständigen 3D-Laserscannersystems eingegangen. Eine detailgetreue Darstellung des Laserscanners in Form eines Bauplans, ermöglicht es anderen Personen das System nachzubauen und zu nutzen.

## Inhalt

---

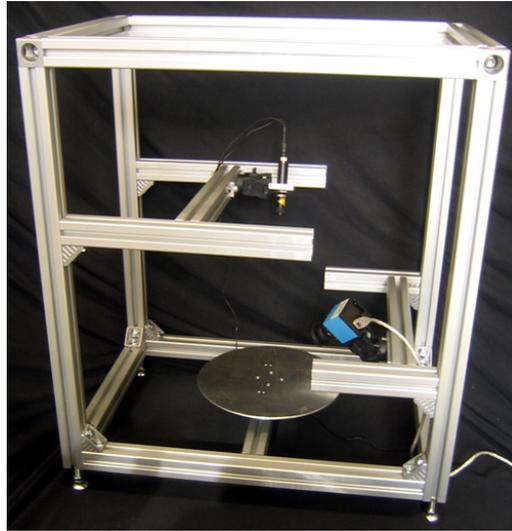
4.1. Gerüst . . . . .	25
4.2. Verkleidung . . . . .	28
4.3. Kamera . . . . .	29
4.4. Laser . . . . .	30
4.5. Drehmesstisch . . . . .	31
4.6. Kalibrierungsvorrichtung . . . . .	32

---

## 4.1. Gerüst

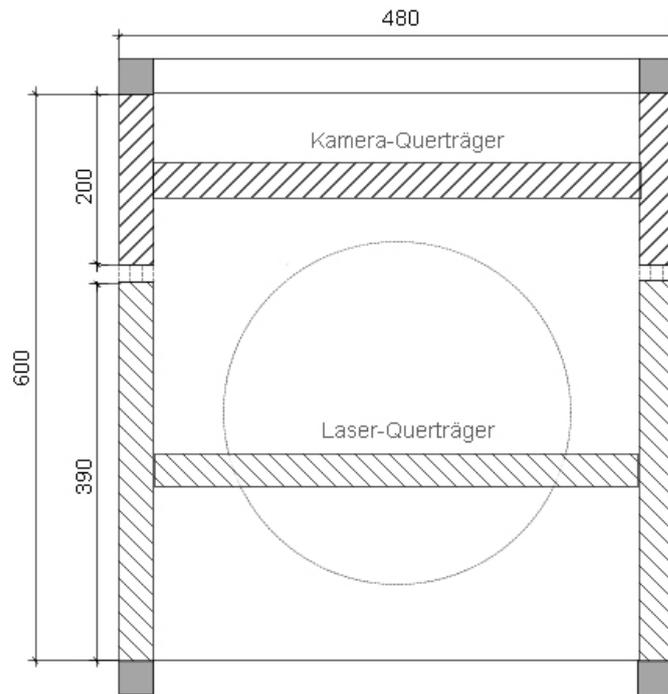
Das Gerüst besteht weitestgehend aus quadratischen Rexroth-Strebenprofilen der Firma Bosch. Zu finden auf der Internetseite der Firma Bosch (Bosch, 2009). Diese haben die Maße von 40 *mm* x 40 *mm*. Mit zwölf dieser Strebenprofile wird ein Raum aufgezogen. An jeweils zwei der vier vertikalen Strebenprofile ist eine Trägerkonstruktion befestigt. Diese besteht aus weiteren zwei vertikal verstellbaren Strebenprofile und einem horizontal verstellbaren Querträger. Dabei ist ein Querträger zur Befestigung einer Kamera oder eines Lasers vorhanden. Das letzte sich im Gerüst befindende Strebenprofil ist als Querträger für einen mit Schrittmotor ansteuerbaren Drehteller zwischen zwei auf dem Boden befindenden Strebenprofilen montiert. Vier in der Höhe verstellbare Gerüstfüße sind zur Ausrichtung des Gesamtgestells an dem unteren Ende der vertikalen Strebenprofile angebracht.

Befestigt werden die genannten Bestandteile des Gerüsts mit Winkelstützen, Würfelverbindern, Nutensteinen und Zentralschrauben.



**Abbildung 4.1.:** Montiertes Grundgerüst mit verstellbaren Kamera- und Laserquerträger und einem befestigten Drehteller.

Eine Aufsicht des Laserscanners zur Betrachtung der montierten Strebenprofile sowie der Trennung der beiden Querträger für die Befestigung einer Kamera und einem Laser ist in Abbildung 4.2 dargestellt.



**Abbildung 4.2.:** Draufsicht des Grundgerüstes mit Maßen in *mm*.

Die zugehörigen Maße der Strebenprofile sind in der Tabelle 4.1 aufgelistet.

Gerüstfläche	Breite	Höhe
Oberseite	480 <i>mm</i>	680 <i>mm</i>
Rückseite	480 <i>mm</i>	740 <i>mm</i>
Linke-Seite	680 <i>mm</i>	740 <i>mm</i>
Rechte-Seite	680 <i>mm</i>	740 <i>mm</i>
Grundseite	480 <i>mm</i>	600 <i>mm</i>

**Tabelle 4.1.:** Beschreibung der Gerüstmasse in *mm*.

## 4.2. Verkleidung

Die Verkleidung des Laserscanners setzt sich aus fünf Verkleidungsflächen und einem zugeschnittenen Tuch zusammen. Jede Verkleidungsfläche hat acht Löcher, so dass mit Nutensteinen, die in die Strebenprofile eingelegt werden, die Verkleidungsflächen mit dem Gerüst verbunden werden können. Jedes Verkleidungsstück ist außerdem von innen mit einer schwarzen nicht Licht reflektierenden Folie beklebt. Um die Kanten des Laserscanners runder darzustellen, wurden im Verhältnis zu den Gerüstmassen die Verkleidungsstücke um jeweils 6 mm verkürzt. Die Maße der Verkleidungsstücke sind in der Tabelle 4.2 aufgelistet.

Ein zugeschnittenes Tuch ist so an die offene Vorderseite des Laserscanners montiert, das zu scannende Objekte einerseits mit Lichteinfall auf den Drehteller gelegt werden können und andererseits, um im Scanvorgang einen komplett abgedunkelten Laserscan zu ermöglichen.

Verkleidungsfläche	Breite	Höhe
Oberseite	474 mm	674 mm
Rückseite	474 mm	734 mm
Linke-Seite	674 mm	734 mm
Rechte-Seite	674 mm	734 mm
Grundseite	474 mm	594 mm

**Tabelle 4.2.:** Auflistung der Verkleidungsmasse in *mm*.

Der komplett verkleidete Laserscanner ist in der folgenden Abbildung 4.3 dargestellt.



**Abbildung 4.3.:** Verkleidung des Laserscanners.

## 4.3. Kamera

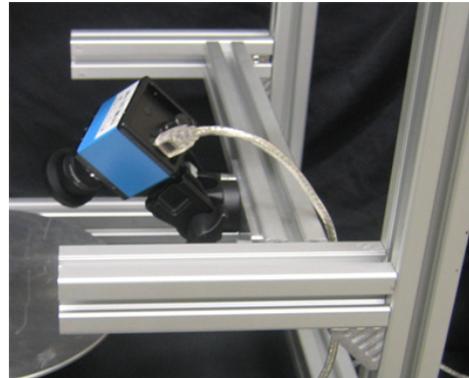
In dieser Arbeit wird die CCD-Monochrom-Kamera „DMK 41BF02“ der Firma „The Imaging Source“ eingesetzt. Sie verfügt über eine Auflösung von  $1280 \times 960$  Bildpunkten und überträgt bis zu 15 Bilder pro Sekunde über eine Firewire-Schnittstelle. CCD steht für „Charge-Coupled Device“ und ist ein lichtempfindliches Bauteil welches aus einer Vielzahl von Fotodioden besteht. Als Kameraobjektiv kommt ein hochauflösendes Objektiv „H0514-MP“, das speziell für den Nahbereich entwickelt wurde, zum Einsatz.

Die Kamera lässt sich in der Tiefe und Höhe durch den am Grundrahmen befestigten verstellbaren Kameraträger beliebig positionieren. Zusätzlich ermöglicht die Halterung der Kamera eine Positionierung in der Breite und einen über  $180^\circ$  Schwenkbereich der Kamera in vertikaler und horizontaler Richtung.

Die Abbildung 4.4 zeigt die am Querträger montierte Kamera.



(a) Nahaufnahme der Kamera



(b) Kameramontage am Querträger

**Abbildung 4.4.:** Darstellung der montierten Kamera.

Detaillierte Informationen bezüglich der angebotenen Kameras der Firma „The Imaging Source“ finden sich auf (The-Imaging-Source, 2009a). Zu verwendende Objektive sind unter (The-Imaging-Source, 2009b) erhältlich.

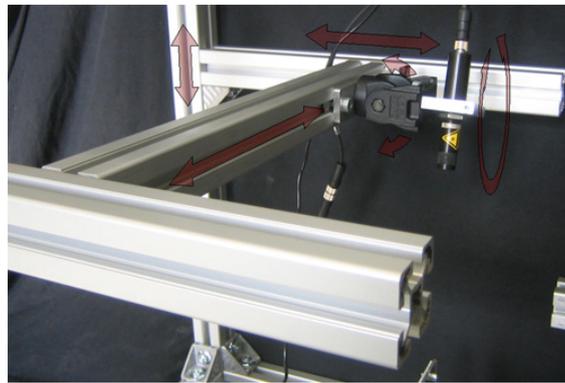
## 4.4. Laser

Ein Linienlaser mit der Typenbezeichnung „Z10V-TTL-SLG30“ der Firma Z-Laser wird in dieser Arbeit verwendet. Dieser Laser hat eine gaußsche Lichtverteilung entlang der Laserlinie mit einer sehr gut fokussierbaren Optik. Der Öffnungswinkel beträgt  $30^\circ$ . Weitere Informationen des Lasers finden sich auf (Zimmermann und Lang, 2004).

Wie die Kamera ist der Laser ebenfalls auf einem Querträger mit einer Halterung montiert, sodass eine beliebige Positionierung und Ausrichtung eingestellt werden kann. Die Abbildung 4.5 stellt dies grafisch dar.



(a) Nahaufnahme des Lasers



(b) Laserbefestigung am Querträger

**Abbildung 4.5.:** Darstellung des angebrachten Lasers.

## 4.5. Drehmesstisch

Ein Drehmesstisch kommt in der Messtechnik zum Einsatz, wenn ein Objekt nicht aus einem erfassten Blickwinkel komplett vermessen werden kann. Es ist deshalb notwendig, das zu vermessende Objekt zu drehen. Der montierte Drehmesstisch besteht aus einem ansteuerbaren Drehteller, der das Objekt trägt und rotiert. Um die Achsenansteuerung des Drehtellers umzusetzen, wurde zusätzlich eine Positionierungssteuerung installiert und konfiguriert. In dieser Arbeit kam die universale Positionierungssteuerung „PS 90“ in Kombination eines Drehmesstisches „DMT65+SM240“ der Firma OWIS zum Einsatz. Für nähere Informationen wird auf (OWIS, 2009) verwiesen.



(a) Ansteuerbarer Drehteller

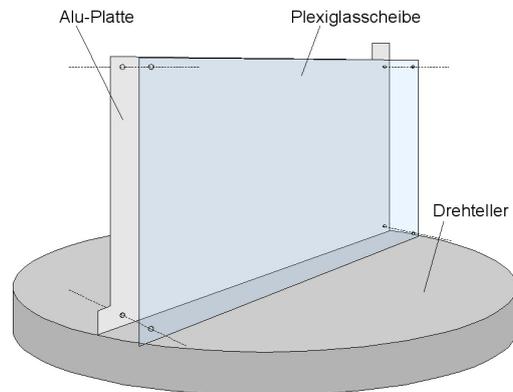


(b) Universal-  
Positionierungssteuerung  
PS 90

**Abbildung 4.6.:** Drehmesstisch in Kombination mit einer Positionierungssteuerung zur vollständigen Ansteuerung eines Drehtellers.

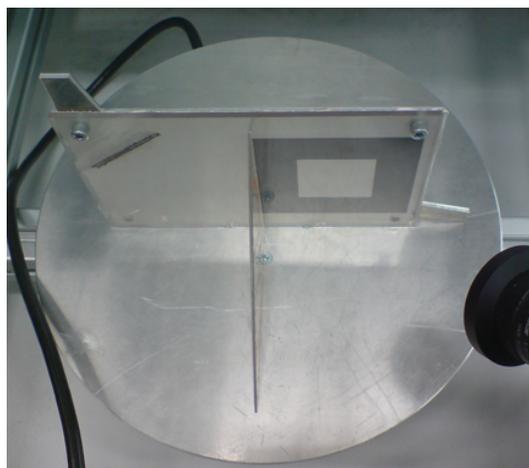
## 4.6. Kalibrierungsvorrichtung

Damit die Komponenten des Laserscanners zueinander kalibriert werden können, bedarf es einer Vorrichtung (Abbildung 4.7) mit Kalibrierungsmerkmalen.



**Abbildung 4.7.:** Aufbau der Kalibrierungsvorrichtung.

Eine Kalibrierungsvorrichtung besteht aus einer Halterung und einem Kalibrierungsbild, das im Kalibrierungsprozess zu identifizierende Merkmale enthält. Die Vorrichtung wurde aus einer Aluminiumplatte und einer Plexiglasscheibe zusammengesetzt. Befestigt wird die Vorrichtung mit zwei Schrauben, die in zwei auf der Rotationsachse des Drehtellers befindenden Löchern verschraubt werden. Die Abbildung 4.8 zeigt die Vorrichtung mit dem eingesetzten Kalibrierungsmuster. Aus der Abbildung ist ersichtlich, dass die Vorrichtung unbedingt in einem  $90^\circ$  Winkel und damit orthogonal zum Drehteller platziert werden muss. Wird die Vorrichtung nicht orthogonal platziert, treten im Kalibrierungsprozess sehr hohe Ungenauigkeiten auf, die im Messvorgang zu großen Vermessungsfehlern führen. Deshalb wird zur Überprüfung der Rechtwinkligkeit ein Geodreieck verwendet.

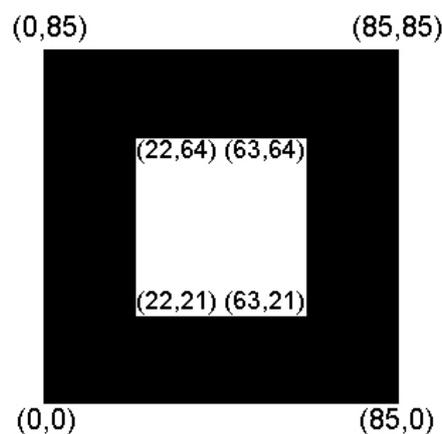


**Abbildung 4.8.:** Kalibrierungsvorrichtung mit eingelegtem Kalibrierungsmuster.

Das aktuelle Kalibrierungsmuster hat sich gegenüber zahlreichen anderen Mustervariationen durchgesetzt. Bei diesem Kalibrierungsbild werden die gesuchten Merkmale am schnellsten und ohne Falschdeutung von Kanten im Kalibrierungsprozess gefunden.

Das Kalibrierungsbild hat eine weiße Grundfarbe. Mit Hilfe eines aufgedruckten schwarzen Rahmens ergeben sich acht Eckpunkte, die im Kalibrierungsprozess erfasst werden müssen. Diese Merkmale sind im System bekannt. Das Kalibrierungsbild wird zwischen die Aluminiumplatte und der Plexiglasscheibe so eingespannt, dass die linke untere Ecke des schwarzen Rahmens direkt auf den Mittelpunkt des Drehtellers liegt.

Nachfolgend ist das Kalibrierungsmuster dargestellt. Es enthält acht Merkmalspunkte, die als Koordinatenpunkte der Form  $(x,y)$  im System bekannt sind.



**Abbildung 4.9.:** Kalibrierungsmuster mit acht im System bekannten Merkmalspunkten.

Zusammengefasst besteht der technische Aufbau des Laserscanners aus einer Industriekamera, einem Laser mit einer zylindrischen Linse für die Linienprojektion und einem Drehteller, der von einem hochgenau positionierbaren Schrittmotor angetrieben wird.



# 5. Realisierung

Die Messprozedur zur Oberflächenrekonstruktion ist in acht Phasen gegliedert, die in den nachfolgenden Unterkapiteln ausführlich erläutert werden. Das Übersichtsdiagramm (Abbildung 5.2) veranschaulicht den schematischen Ablauf der 3D-Objektvermessung, ausgehend von der Kalibrierung, über die Erfassung der Laserlinien, bis zur Rekonstruktion der gesuchten Oberflächenstruktur in Form von dreidimensionalen Koordinatenpunkten. Die ersten vier Phasen sind notwendig, um die Oberflächenstruktur zu erfassen. Vier weitere optionale Phasen ermöglichen ein noch detailgetreueres, platzsparendes, modelliertes und digitales Abbild des zu vermessenden Objektes.

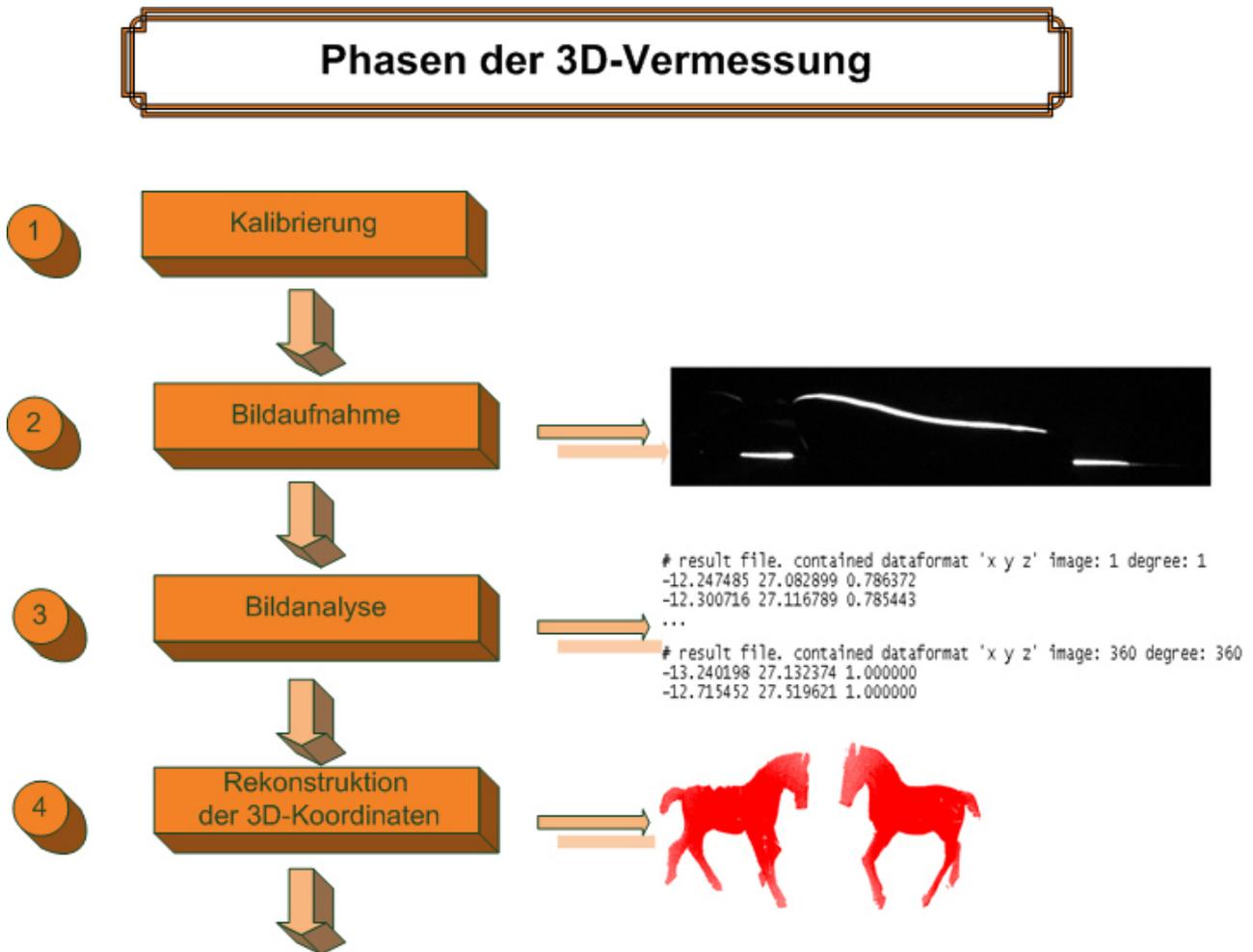
## Inhalt

---

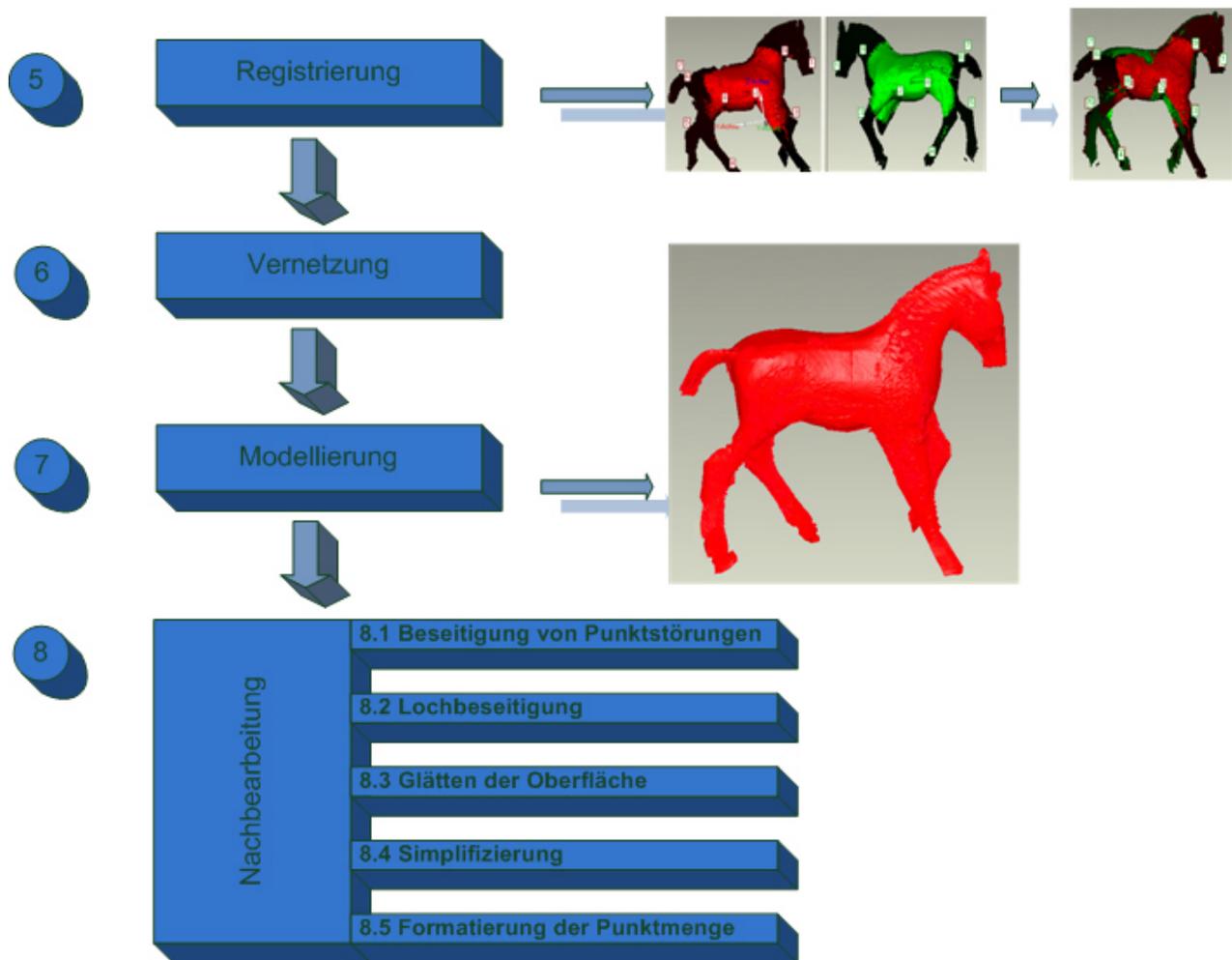
5.1. Kalibrierung . . . . .	39
5.1.1. Vorbereitung . . . . .	40
Ausrichtung des Lasers . . . . .	40
Ausrichtung und Einstellung der Kamera . . . . .	41
5.1.2. Bestimmung der Referenzpunkte in der Bildebene . . . . .	42
Algorithmus zur Bestimmung der Referenzpunkte . . . . .	42
Schritt 1: Näherungsweise Bestimmung der Kantenposition . . . . .	43
Schritt 2: Bestimmung der exakten Kantenposition . . . . .	44
Schritt 3: Sortieren der Geraden und Bestimmung der Schnittpunkte . . . . .	48
5.1.3. Berechnung der Transformationsmatrix . . . . .	49
5.2. Bildaufnahme . . . . .	51
5.3. Bildanalyse . . . . .	53
5.3.1. Vorverarbeitung des Kamerabildes . . . . .	54
5.3.2. Extraktion der Oberflächenpunkte aus dem Bild . . . . .	55
5.3.3. Berechnung der Koordinaten in der Laserebene . . . . .	56
5.4. Rekonstruktion der 3D-Koordinaten . . . . .	57

5.5. Registrierung . . . . .	58
5.6. Vernetzung . . . . .	60
5.7. Modellierung . . . . .	61
5.8. Nachbearbeitung . . . . .	62
5.8.1. Beseitigung von Punktstörungen . . . . .	62
5.8.2. Lochbeseitigung . . . . .	63
5.8.3. Glätten der Oberfläche . . . . .	63
5.8.4. Simplifizierung . . . . .	64
5.8.5. Formatierung der Punktmenge . . . . .	66

---



**Abbildung 5.1.:** Schematische Darstellung der Oberflächenrekonstruktion.



**Abbildung 5.2.:** Schematische Darstellung der Oberflächenrekonstruktion.

## 5.1. Kalibrierung

Die erste Stufe eines Messsystems ist die Kalibrierung der Systemkomponenten zueinander.

Im dem nun folgenden Kapitel werden die Hintergründe, die Konzeption und die mathematischen Zusammenhänge der implementierten Verfahren für die Kalibrierung näher erläutert. Die hierbei verwendeten mathematischen und bildverarbeitungstechnischen Grundlagen wurden in Kapitel 3 ab S.11 beschrieben.

Um eine Objektvermessung durchführen zu können, ist es notwendig aus Koordinaten im Kamerabild (der Bildebene) die entsprechenden Koordinaten in der Laserebene berechnen zu können. Wie in Abschnitt 3.1.3 bereits erwähnt wurde, wird hierfür die Transformationsmatrix einer perspektivischen Transformation benötigt, mit der Koordinaten in der Bildebene auf Koordinaten in der Laserebene abgebildet werden können.

Für die Berechnung der Transformationsmatrix werden, wie im Folgenden gezeigt wird, mindestens vier Punkte benötigt, deren Koordinaten in der Laserebene ( $\vec{l}$ , gemessen in *mm*) genau bekannt sind. Die dazugehörigen Koordinaten der Abbildungen dieser Punkte ( $\vec{c}$ , gemessen in *pixel*) können dann im Kamerabild bestimmt werden (siehe Abschnitt 5.1.2).

Um die Koordinaten der Referenzpunkte angeben zu können ist die Wahl eines Bezugspunktes (Ursprung des Koordinatensystems der Laserebene) notwendig. Dieser wird zunächst auf die linke untere Ecke des Kalibrierungsmusters gelegt.

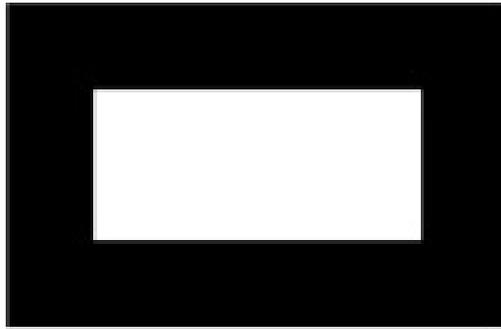
Im Folgenden soll nun die Konzeption des für die Kalibrierung entwickelten Verfahrens näher erläutert werden. Das hier beschriebene Kalibrierungsverfahren ist an das in der Publikation „*Building an Accurate Range Finder with Off the Shelf Components*“ (Jezouin u. a., 1988) vorgestellte Verfahren angelehnt.

In diesem Verfahren wird ein Kalibrierungsmuster mit bekannten Abmessungen in die Laserebene gebracht, wodurch eine Reihe von Referenzpunktpaaren ermittelt werden kann. Diese Referenzpunktpaare ermöglichen es, die Parameter der bereits erwähnten perspektivischen Transformation zu berechnen.

In den folgenden Abschnitten werden die einzelnen Schritte, die nötig sind um die Kalibrierung durchzuführen, im Detail erläutert und an Beispielen verdeutlicht.

### 5.1.1. Vorbereitung

Ein Kalibrierungsmuster, wie in Abbildung 5.3 dargestellt, wird auf ein Blatt Papier gedruckt, welches in die Kalibriervorrichtung eingespannt wird (siehe Abbildung 4.7 auf S. 32).



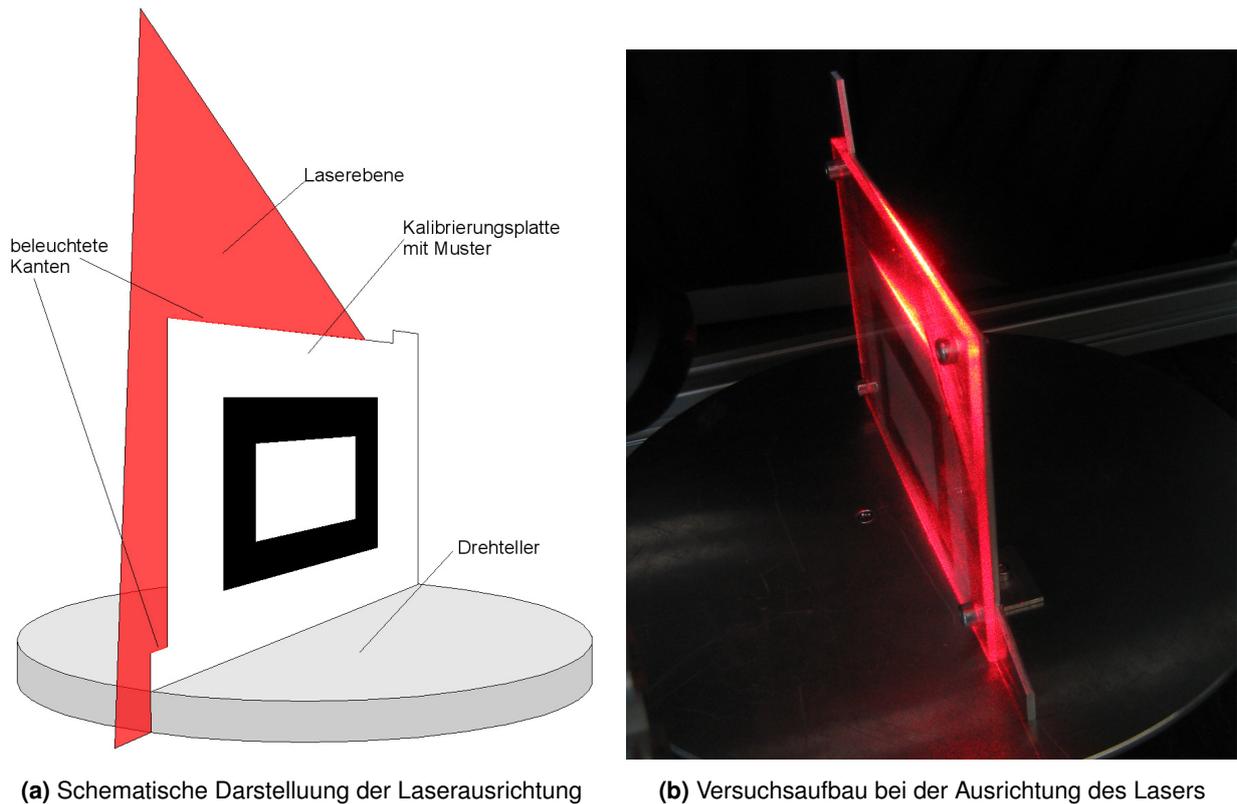
**Abbildung 5.3.:** Beispielhaftes Kalibrieremuster

Hierbei sind die genauen Abmessungen des Kalibrierungsmusters von entscheidender Bedeutung. Der Bezugspunkt für die Koordinaten der acht Eckpunkte des Kalibrierungsmusters ist hierbei frei wählbar.

#### **Ausrichtung des Lasers**

Da die Kalibriervorrichtung auf dem Drehteller verschraubt und damit unbeweglich ist, muss der Laser nun so justiert werden, dass er sowohl die obere Kante der Kalibriervorrichtung als auch die herausstehende Ecke am unteren Ende der Platte beleuchtet (siehe Abbildung 5.4 auf S. 41).

Die Linse des Lasers sollte so eingestellt werden, dass die auf dem Drehteller sichtbare Linie möglichst scharf zu erkennen ist. Ist die Laserebene ausgerichtet, können die Eckpunkte des Kalibrierungsmusters als bekannte Punkte in der Laserebene betrachtet werden.



**Abbildung 5.4.:** Ausrichtung des Lasers

### Ausrichtung und Einstellung der Kamera

Nachdem der Laser ausgerichtet wurde, müssen nun die Ausrichtung und Einstellungen der Kamera vorgenommen werden. Die Kamera sollte so positioniert werden, dass der Winkel zwischen Kamera und Laser etwa 45 Grad beträgt. Das Kalibrierungsmuster darf auf keinen Fall über den Rand des Kamerabildes hinausragen. Ist die Position und Ausrichtung der Kamera eingestellt, müssen noch die Werte für Brennweite und Blende des Objektivs sowie die Empfindlichkeit und Belichtungszeit des Sensors justiert werden.

Grundsätzlich ist es hierbei empfehlenswert die Größe der Blendenöffnung der Kamera so klein wie möglich zu halten, da daraus ein großer Tiefenschärfebereich resultiert, wodurch die Kanten und Eckpunkte des Kalibrierungsmusters auf ihrer gesamten Breite scharf abgebildet werden können. Der Tiefenschärfebereich eines Objektivs ist der Tiefenbereich (mindestabstand bis höchstabsand vor dem Objektiv) in dem ein Objekt vor der Kamera scharf abgebildet werden kann.

Die ebenfalls daraus resultierende kleinere Lichtausbeute kann durch eine entsprechende Beleuchtung, eine längere Belichtungszeit oder ein Anheben der Empfindlichkeit des Sensors kompensiert werden.

Nachdem die Einstellungen von Laser und Kamera vorgenommen wurden, müssen die Halterungen arretiert werden, um Fehler durch ein versehentliches Verstellen zu vermeiden.

### 5.1.2. Bestimmung der Referenzpunkte in der Bildebene

Um die Koordinaten  $\vec{c}$  der acht Referenzpunkte in der Bildebene so genau wie möglich zu bestimmen, sind einige Schritte notwendig. Es wird zunächst eine Abschätzung über die Positionen und Richtungen der Kanten des Kalibrierungsmusters vorgenommen. Anschließend werden diese Abschätzungen korrigiert und die Eckpunkte des Kalibrierungsmusters als Schnittpunkte der korrigierten Kantengeraden bestimmt.

Abbildung 3.3a auf S. 22 zeigt ein Beispiel für ein zur Kalibrierung verwendetes Kamerabild. Die einzelnen Schritte des Kalibrierungsverfahrens werden im Folgenden an diesem Beispiel verdeutlicht.

#### Algorithmus zur Bestimmung der Referenzpunkte

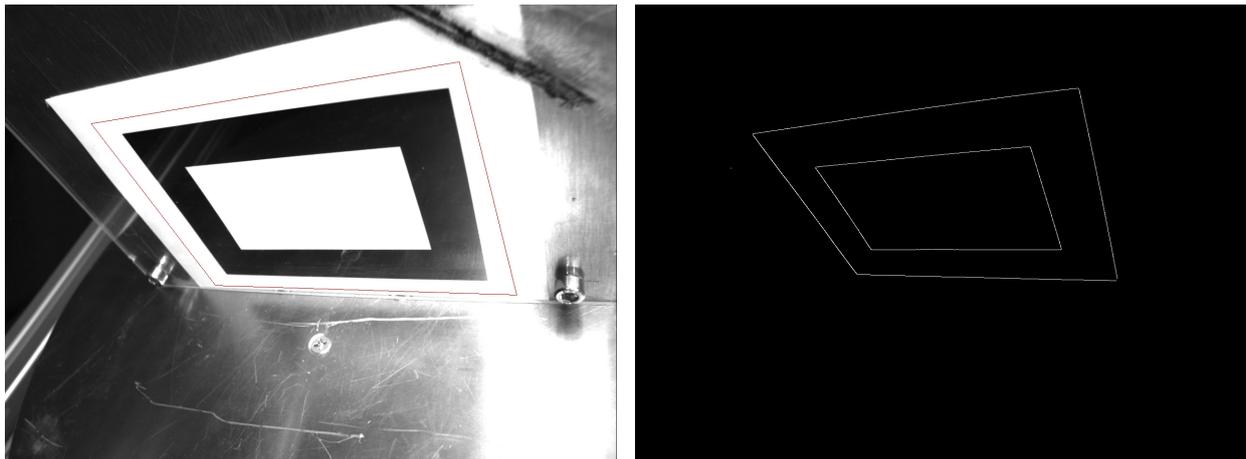
Im Folgenden wird das für die Bestimmung der Koordinaten der Referenzpunkte entwickelte Verfahren als Algorithmus beschrieben. Eine detaillierte Beschreibung der einzelnen Schritte ist in den folgenden Abschnitten zu finden.

1. Aus dem Quellbild wird mit dem Canny-Algorithmus ein Kantenbild und ein Gradientenrichtungsbild erzeugt.
2. Alle Punkte des Kantenbildes, die außerhalb der *Region of Interest* liegen, werden auf Null gesetzt.
3. Mit Hilfe der Hough-Transformation werden die ungefähren Positionen und Richtungen der acht Kantengeraden des Kalibrierungsmusters bestimmt.
4. Für jede der in Schritt 3 gefundenen Geraden:
  - a) Es werden die Start und Endpunkte von zu der Gerade senkrechten Geraden mit einer bestimmten Breite und einem festen Abstand zueinander berechnet.
  - b) Für jede dieser Senkrechten:

- i. Mittels bilinearer Interpolation werden die Intensitätswerte der auf der Senkrechten liegenden Punkte berechnet.
  - ii. Innerhalb des so berechneten Intensitätsverlaufes wird die exakte Position des Wendepunktes bestimmt.
  - iii. Aus der Position und Richtung der Senkrechten und der Position des Wendepunktes innerhalb des Intensitätsverlaufes werden die Bildkoordinaten des Wendepunktes berechnet.
- c) Aus den in Schritt 4b bestimmten Kantenpunkten wird eine Ausgleichsgerade berechnet.
5. Die in Schritt 4 berechneten Ausgleichsgeraden werden anhand ihrer Steigungswinkel als entweder horizontale oder vertikale Geraden kategorisiert.
  6. Die jeweils vier horizontalen und vertikalen Geraden werden aufsteigend nach ihrem Abstand zum Ursprung sortiert.
  7. Es werden die Schnittpunkte der horizontalen und vertikalen Geraden berechnet, in denen sich eine äußere mit einer äußeren bzw. eine innere mit einer inneren Gerade schneidet. Die so bestimmten Schnittpunkte sind die Bildkoordinaten der acht Referenzpunkte.

### **Schritt 1: Näherungsweise Bestimmung der Kantenposition**

Für die Bestimmung der näherungsweise Kantenpositionen wird die Hough-Transformation verwendet. Die hierfür benötigten Kanteninformationen werden mithilfe des Canny-Algorithmus berechnet. Das durch den Canny-Algorithmus erzeugte Kantenbild ist in Abbildung 3.3e auf S. 22 dargestellt. In diesem Bild lässt sich auch erkennen, dass hierbei noch zu viele Kanten, also solche, die nicht zu dem Kalibrierungsmuster gehören, erkannt werden. Da es nur sehr schwer möglich wäre solche Probleme zu vermeiden bzw. automatisch zu erkennen, wird dem Kalibrierungsprogramm die „*Region of interest*“ (ROI) angegeben. Hierbei handelt es sich um ein Polygon, das den Bereich des Bildes umschließt, der bei der Kalibrierung berücksichtigt werden soll. Bereiche ausserhalb dieses Polygons werden in allen Schritten der Kalibrierung ignoriert („*Clipping*“). Abbildung 5.5 auf S. 44 zeigt die ROI in dem Beispielbild und das Kantenbild nach dem Clipping.



(a) Kalibrierungsbild mit eingezeichneter ROI

(b) Kantenbild nach Clipping

**Abbildung 5.5.:** Kalibrierungsbild mit *Region of interest*

In diesem Kantenbild können nun mittels Hough-Transformation (siehe Abschnitt 3.2.3 auf S.23) die signifikanten Geraden bestimmt werden. Die so gefundenen Geraden sind in Abbildung 5.6 auf S. 45 dargestellt.

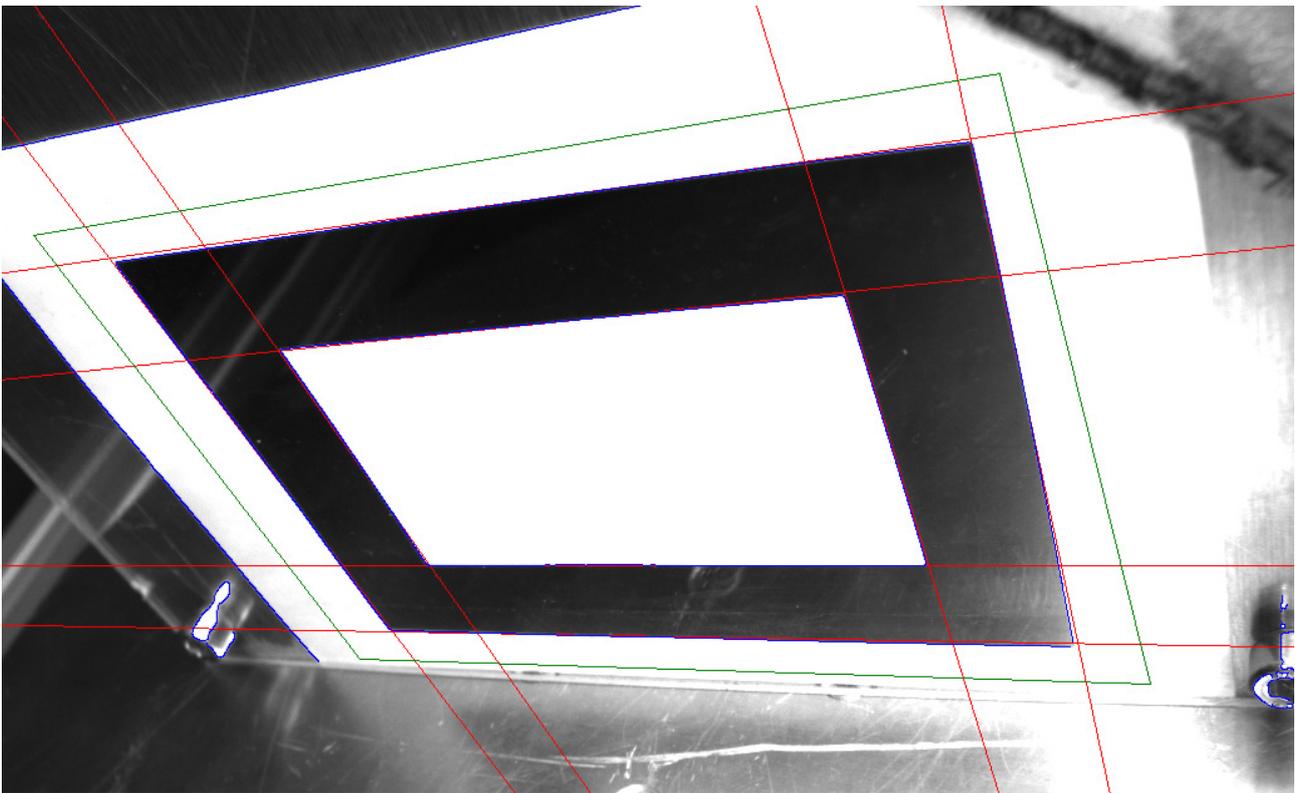
### Schritt 2: Bestimmung der exakten Kantenposition

Im nun folgenden Schritt werden die von der Hough-Transformation nur ungenau bestimmten Geraden mit einem speziell hierfür entwickelten Verfahren einer Korrektur unterzogen, um die exakte Position der Kante zu bestimmen. Hierzu werden zunächst in festen Abständen Senkrechten auf jeder der näherungsweise bestimmten Geraden errichtet. Sowohl der Abstand zwischen den Senkrechten als auch die Länge können dem Kalibrierungsprogramm als Parameter übergeben werden.

Mittels bilinearer Interpolation kann der Intensitätsverlauf entlang dieser Senkrechten berechnet werden. Dieses Vorgehen wird auch als „*Edge-Sampling*“ bezeichnet.

Innerhalb des dadurch berechneten Intensitätsverlaufes („*Sample*“) kann nun die exakte Position der Kante bestimmt werden. Diese wird hierbei definiert als der Wendepunkt des Intensitätsverlaufes, also die Nullstelle der zweiten Ableitung, die mithilfe des Laplace-Operators berechnet werden kann. Zwei Beispiele für solche Intensitätsverläufe und deren zweite Ableitungen sind in Abbildung 5.7 gegeben.

Um die genaue Position eines Kantenpunktes innerhalb eines Samples zu ermitteln, werden zunächst die Intensitätswerte sowie die zweite Ableitung eines Samples untersucht. Da der Intensitätsverlauf aufgrund von Rauschen in der Regel mehr als

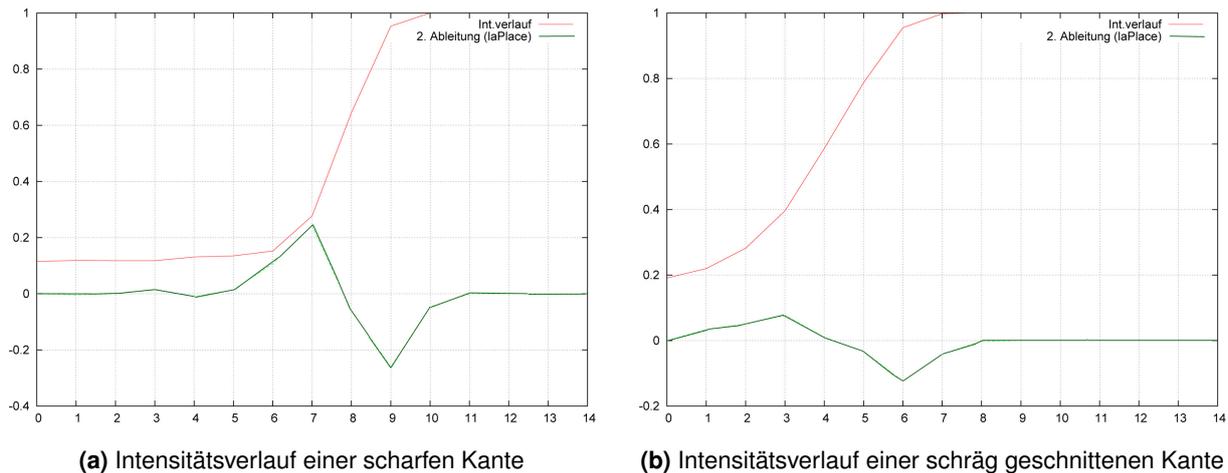


**Abbildung 5.6.:** Kalibrierungsbild mit näherungsweise bestimmten Kantengeraden:  
 ÄUSSERER- UND INNERER UMRISSE DES SCHWARZEN RAHMENS: durch den Canny-Algorithmus ermittelte Kantenpunkte; RECHTECK UM DEN SCHWARZEN RAHMEN: Region of Interest; VERTIKALE- UND HORIZONTALE LINIEN ENTLANG DES SCHWARZEN RAHMENS: lokale Maxima des Hough-Raums nach der Hough-Transformation.

einen Wendepunkt aufweist, wird derjenige Nulldurchgang der zweiten Ableitung gesucht, der zwischen ihrem Maximal- und Minimalwert liegt. Die exakte Position des Nulldurchganges innerhalb des Samples wird mittels linearer Interpolation berechnet.

In wenigen ungünstigen Fällen kann es vorkommen, dass die Senkrechte eine andere Kante oder einen Eckpunkt des Kalibrierungsmusters schneidet. Um diese Fälle zu erkennen und damit Fehler bei der Kantenkorrektur zu verhindern, werden noch eine Reihe von Überprüfungen durchgeführt. In der folgenden Liste sind alle Bedingungen, die erfüllt sein müssen, um einen gefundenen Kantenpunkt zu verifizieren, aufgeführt.

- Die zweite Ableitung des Intensitätsverlaufes muss mindestens einen Nulldurchgang aufweisen.
- Maximum und Minimum der zweiten Ableitung dürfen nicht zu weit auseinander-



**Abbildung 5.7.:** Beispiele für Intensitätsverläufe: OBERE FUNKTION: Intensitätsverlauf senkrecht zu der Kantengeraden; UNTERE FUNKTION: diskrete zweite Ableitung des Intensitätsverlaufes.

der liegen. Dabei kennzeichnen das Maximum und Minimum der zweiten Ableitung den oberen und unteren Wendepunkt der ersten Ableitung und damit den Anfangs- und Endpunkt des Kantenverlaufes.

- Die Differenz von Minimal- und Maximalwert der zweiten Ableitung darf nicht zu klein sein.
- Die Differenz der Intensitätswerte an den Stellen des Minimums und Maximums der zweiten Ableitung muss ausreichend groß sein.
- Der Winkel zwischen der Richtung des Samples und der Gradientenrichtung (diese wurde bereits im ersten Schritt durch den Canny-Operator berechnet) darf nicht zu groß sein.

Die Vergleichswerte für diese Bedingungen können dem Kalibrierungsprogramm als Parameter übergeben werden, jedoch hat es sich gezeigt, dass die Voreinstellungen in den meisten Fällen zu sehr guten Ergebnissen führen.

Die Koordinaten des gefundenen Kantenpunktes können jetzt aus der Position und Richtung des Samples und der Kantenposition innerhalb dieses Samples berechnet werden.

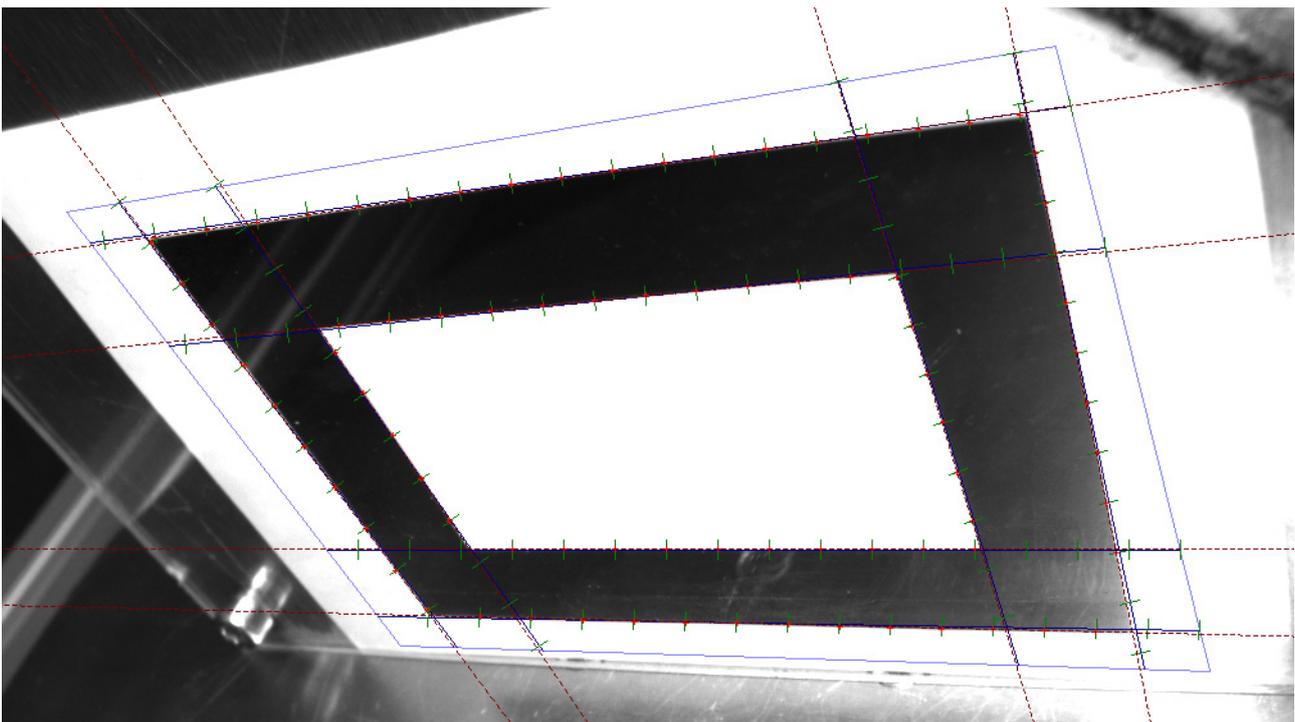
Für jede der im ersten Schritt bestimmten Kantengeraden kann nun aus den  $n$  gefundenen Kantenpunkten  $\vec{p}_i$ ,  $i = 1, \dots, n$  eine Ausgleichsgerade berechnet werden. Basierend auf der Modellgleichung  $y = a_1x + a_0$  für die zu bestimmende Gerade und

den bekannten Punkten  $\vec{p}_i = (x_i \ y_i)^T$  ergibt sich ein lineares Ausgleichsproblem mit

$$\begin{pmatrix} x_0 & 1 \\ x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} \approx \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad (5.1)$$

Die Lösung dieses Ausgleichsproblems im Sinne der kleinsten Fehlerquadrate wird mithilfe der QR-Zerlegung (siehe Abschnitt 3.1.4) berechnet. Somit ist eine optimale Näherung an die Parameter  $a_0$  und  $a_1$  der Geradengleichung der korrigierten Kantengerade bekannt.

Abbildung 5.8 zeigt das Kalibrierungsbild in dem die näherungsweise bestimmten Geraden, die Edge-Samples, die gefundenen Kantenpunkte sowie die korrigierten Kantengeraden eingezeichnet sind.



**Abbildung 5.8.:** Veranschaulichung des Kantenkorrekturverfahrens: VERTIKALE- UND HORIZONTALE LINIEN ENTLANG DES SCHWARZEN RAHMENS: näherungsweise bestimmte Geraden; DIE RAHMENKANTE SCHNEIDENDEN KURZEN LINIEN: Edge-Samples; ROT: Kantenpunkte; GESTRICHELT: korrigierte Geraden.

### Schritt 3: Sortieren der Geraden und Bestimmung der Schnittpunkte

Um nun die Schnittpunkte der Geraden und damit die Eckpunkte des Kalibrierungsmusters bestimmen zu können, müssen die korrigierten Geraden zunächst in horizontale und vertikale Geraden gruppiert und anschließend von oben nach unten bzw. von links nach rechts sortiert werden.

Um die Geraden als entweder horizontal oder vertikal zu kategorisieren ist es naheliegend, sie Anhand ihrer Steigung so zu gruppieren, dass Geraden mit ähnlicher Steigung in jeweils einer Gruppe zusammengefasst werden. Anstelle der Steigungen  $m_i$  der Geraden werden hierzu jedoch die Steigungswinkel  $\alpha_i = \arctan(m_i)$ ,  $\alpha_i \in [-\pi, \pi]$  verwendet, da sich mit ihnen die Ähnlichkeit zweier Steigungen unabhängig von ihrem Betrag beschreiben lässt.

Die Steigungen zweier vertikaler Geraden mit Steigungswerten von  $m_1 = 100$  und  $m_2 = 1000$  ( $\Delta m = 900$ ) sind sich genauso ähnlich wie die zweier horizontaler Geraden mit Steigungen  $m_3 = \frac{1}{100}$  und  $m_4 = \frac{1}{1000}$  ( $\Delta m = 0.009$ ).

Es lässt sich zeigen, dass sich die acht Geraden unabhängig vom Blickwinkel der Kamera und damit unabhängig vom Grad der perspektivischen Verzerrung immer anhand des Mittelwertes der Beträge ihrer Steigungswinkel  $\alpha_{avg} = \frac{1}{n} \sum_{i=0}^n |\alpha_i|$  in zwei Gruppen von je vier Geraden einordnen lassen. Eine Gerade wird als vertikale Gerade bezeichnet, falls  $|\alpha_i| \geq \alpha_{avg}$ , für horizontale Geraden gilt  $|\alpha_i| < \alpha_{avg}$ .

Bevor nun die Schnittpunkte berechnet werden können, müssen die Geraden noch von links nach rechts bzw. von oben nach unten sortiert werden. Für diese Sortierung wird der Abstand der Geraden zum Ursprung des Koordinatensystems als Vergleichskriterium verwendet.

Um abschließend die Koordinaten der Eckpunkte des Kalibrierungsmusters zu erhalten, müssen die Schnittpunkte der horizontalen mit den vertikalen Geraden berechnet werden. Von den insgesamt 16 Schnittpunkten werden jedoch nur die 8 Schnittpunkte benötigt, in denen sich eine der äußeren horizontalen Geraden mit einer der äußeren vertikalen Geraden bzw. eine der inneren horizontalen Geraden mit einer der inneren vertikalen Geraden schneidet.

Diese acht Schnittpunkte müssen von links nach rechts und von oben nach unten durchnummeriert werden, um den Abgleich mit den bekannten Koordinaten in der

Laserebene zu ermöglichen. Die Nummerierung muss bezogen auf das Kalibrierungsmuster erfolgen, da die Koordinaten der Schnittpunkte in der Bildebene aufgrund der perspektivischen Verzerrung unter Umständen eine andere Reihenfolge ergeben könnten.

Die Nummerierungssequenz wird durch die Reihenfolge bei der Berechnung der Schnittpunkte sichergestellt. Unter Auslassung der nicht benötigten Schnittpunkte werden die Schnittpunkte von oben nach unten und von links nach rechts nacheinander berechnet.

### 5.1.3. Berechnung der Transformationsmatrix

Wie bereits am Anfang dieses Kapitels auf S.39 beschrieben wurde, ist es das Ziel der Kalibrierung, die Transformationsmatrix einer perspektivischen Transformation zu bestimmen, mit der sich Koordinaten in der Bildebene auf Koordinaten in der Laserebene abbilden lassen. Wie diese Transformationsmatrix aus den nun vorliegenden Referenzpunktpaaren berechnet wird, soll im Folgenden gezeigt werden.

Die Gleichung für die Berechnung der Koordinaten eines Punktes  $\vec{l}$  in der Laserebene aus den Koordinaten eines Punktes  $\vec{c}$  in der Bildebene lautet

$$\vec{l}^* = \mathbf{H}\vec{c}^* \Leftrightarrow \begin{pmatrix} x_l^* \\ y_l^* \\ h_l^* \end{pmatrix} = \begin{pmatrix} h_{11} & \cdots & h_{13} \\ \vdots & \ddots & \vdots \\ h_{31} & \cdots & h_{33} \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} \quad \text{mit} \quad \begin{cases} x_l = \frac{x_l^*}{h_l^*} \\ y_l = \frac{y_l^*}{h_l^*} \end{cases} \quad (5.2)$$

Die Anzahl der in dieser Gleichung auftretenden Unbekannten kann noch auf acht Unbekannte reduziert werden, indem das gesamte Gleichungssystem mit  $\frac{1}{h_{33}}$  multipliziert wird. Dadurch erhalten wir eine neue Transformationsmatrix  $\mathbf{B}$  mit  $b_{ik} = \frac{h_{ik}}{h_{33}}$  und können obige Gleichung umschreiben in

$$\vec{l}^* = \mathbf{B}\vec{c}^* \Leftrightarrow \begin{pmatrix} x_l^* \\ y_l^* \\ h_l^* \end{pmatrix} = \begin{pmatrix} b_{11} & \cdots & b_{13} \\ \vdots & \ddots & b_{23} \\ b_{31} & b_{32} & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} \quad \text{mit} \quad \begin{cases} x_l = \frac{x_l^*}{h_l^*} \\ y_l = \frac{y_l^*}{h_l^*} \\ b_{ik} = \frac{h_{ik}}{h_{33}} \end{cases} \quad (5.3)$$

Aufgrund der homogenen Gleichheit (vgl. Abschnitt 3.1.3 auf S.15) gilt  $\frac{1}{h_{33}}\vec{l}^* \sim \vec{l}^*$ . Beide Koordinaten beschreiben also denselben Punkt im euklidischen Raum. Daher kann die Multiplikation in diesem Fall vernachlässigt werden.

Die acht Werte für  $b_{ik}$  sind also die aus den Referenzpunktpaaren zu bestimmenden Unbekannten. Um diese mit der Methode der kleinsten Quadrate bestimmen zu können, wird ein lineares Gleichungssystem  $\mathbf{M}\vec{x} = \vec{b}$  benötigt, in dem die Parameter  $b_{ik}$  nur noch in dem Vektor der Unbekannten  $\vec{x}$  auftauchen.

Um dies zu erreichen wird die obige Gleichung Komponentenweise aufgeschrieben und die Umwandlung der homogenen Koordinaten von  $\vec{l}^*$  in die euklidischen Koordinaten von  $\vec{l}$  mit  $x_l = \frac{x_l^*}{h_l^*}$  und  $y_l = \frac{y_l^*}{h_l^*}$  durchgeführt. Für die Komponenten von  $\vec{l}$  ergibt sich hiermit:

$$\begin{aligned} x_l &= \frac{x_l^*}{h_l^*} = \frac{b_{11}x_c + b_{12}y_c + b_{13}}{b_{31}x_c + b_{32}y_c + 1} \\ \Leftrightarrow x_l(b_{31}x_c + b_{32}y_c + 1) &= b_{11}x_c + b_{12}y_c + b_{13} \\ \Leftrightarrow x_l &= b_{11}x_c + b_{12}y_c + b_{13} - x_c x_l b_{31} - x_l y_c b_{32} \end{aligned} \quad (5.4a)$$

sowie

$$\begin{aligned} y_l &= \frac{y_l^*}{h_l^*} = \frac{b_{21}x_c + b_{22}y_c + b_{23}}{b_{31}x_c + b_{32}y_c + 1} \\ \Leftrightarrow y_l(b_{31}x_c + b_{32}y_c + 1) &= b_{21}x_c + b_{22}y_c + b_{23} \\ \Leftrightarrow y_l &= b_{21}x_c + b_{22}y_c + b_{23} - x_c y_l b_{31} - y_l y_c b_{32} \end{aligned} \quad (5.4b)$$

Mit den beiden Gleichungen (5.4a) und (5.4b) lässt sich nun ein Gleichungssystem zur Bestimmung der acht Unbekannten  $b_{ik}$  aufstellen. Dabei entstehen aus jedem der  $n$  Referenzpunktpaare zwei Gleichungen. Daher werden zur Berechnung dieser Unbekannten  $n \geq 4$  linear unabhängige Paare von zusammengehörigen Vektoren  $\vec{c}_k = \begin{pmatrix} x_{c_k} & y_{c_k} \end{pmatrix}^T$  und  $\vec{l}_k = \begin{pmatrix} x_{l_k} & y_{l_k} \end{pmatrix}^T$ , mit  $k = 1, \dots, n$  benötigt, um eine eindeutige Lösung zu erhalten.

Dieses lineare Ausgleichsproblem zur Bestimmung der Unbekannten  $b_{ik}$  lässt sich in

Matrixschreibweise nun folgendermaßen darstellen:

$$\mathbf{M}\vec{x} = \vec{b} \Leftrightarrow \begin{pmatrix} x_{c_1} & y_{c_1} & 1 & 0 & 0 & 0 & -x_{l_1}x_{c_1} & -x_{l_1}y_{c_1} \\ 0 & 0 & 0 & x_{c_1} & y_{c_1} & 1 & -y_{l_1}x_{c_1} & -y_{l_1}y_{c_1} \\ \vdots & \vdots \\ x_{c_n} & y_{c_n} & 1 & 0 & 0 & 0 & -x_{l_n}x_{c_n} & -x_{l_n}y_{c_n} \\ 0 & 0 & 0 & x_{c_n} & y_{c_n} & 1 & -y_{l_n}x_{c_n} & -y_{l_n}y_{c_n} \end{pmatrix} \begin{pmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{21} \\ b_{22} \\ b_{23} \\ b_{31} \\ b_{32} \end{pmatrix} \approx \begin{pmatrix} x_{l_1} \\ y_{l_1} \\ \vdots \\ x_{l_n} \\ y_{l_n} \end{pmatrix} \quad (5.5)$$

Für die Lösung dieses Gleichungssystems werden alle acht bekannten Referenzpunktpaare in diese Matrix eingesetzt, wodurch insgesamt 16 Gleichungen zur Bestimmung der 8 Unbekannten zur Verfügung stehen.

Da dieses Ausgleichsproblem oftmals schlecht konditioniert ist, und weniger rechenintensive Verfahren dabei an ihre Grenzen stoßen, wird die Lösung durch Berechnung der Pseudoinversen  $\mathbf{M}^\dagger$  mithilfe der Singulärwertzerlegung ermittelt:

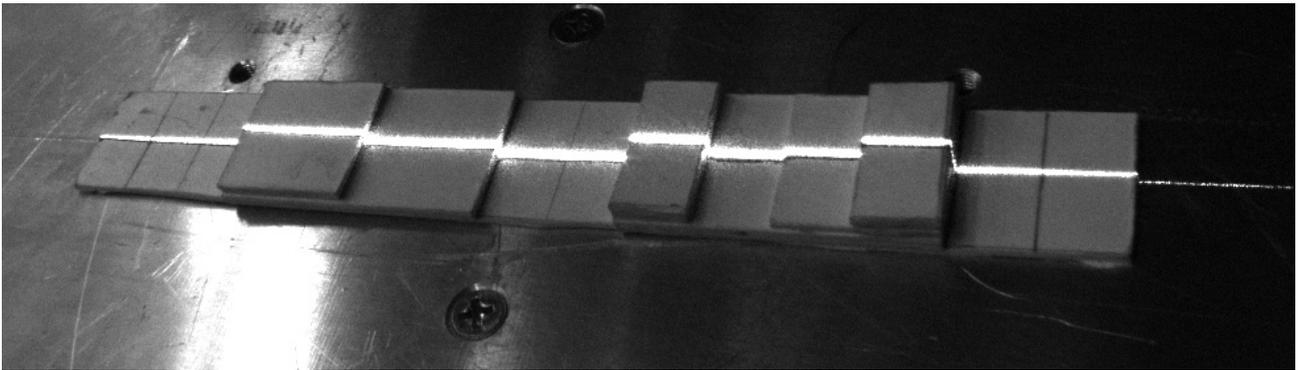
$$\vec{x} = \mathbf{M}^\dagger \vec{b} \Leftrightarrow \vec{x} = \mathbf{V}\mathbf{\Sigma}^\dagger \mathbf{U}^T \quad \text{mit } SVD(\mathbf{M}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (5.6)$$

## 5.2. Bildaufnahme

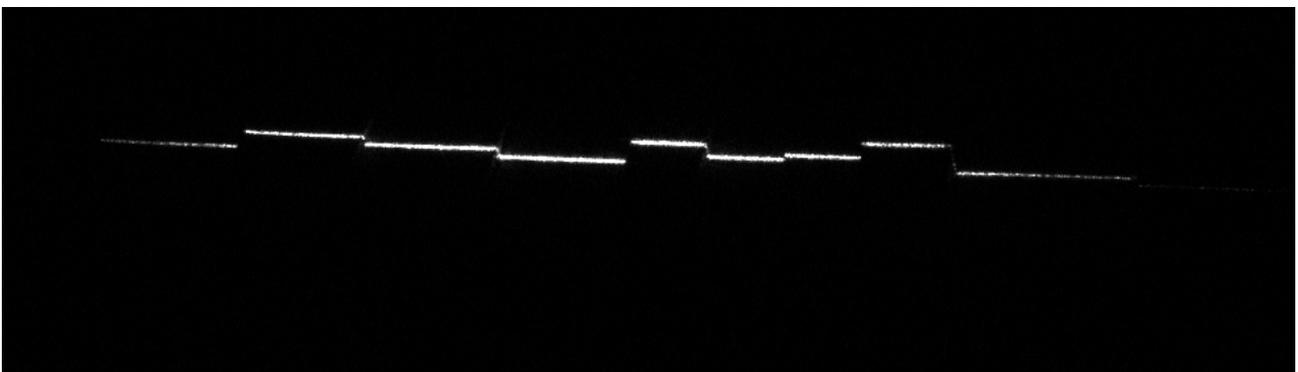
Nachdem die Kalibrierung erfolgreich abgeschlossen ist, kann ein Objekt auf den Drehteller platziert und vermessen werden. In der Aufnahmephase dreht sich der Drehteller um 360 Grad. In dieser Phase werden in vorher festgelegten Abschnitten, Bilder mit der Kamera aufgenommen und die zugehörigen Drehwinkel für die spätere Verarbeitung zwischengespeichert. Die aufgenommenen Bilder zeigen jeweils einen Lichtschnitt und damit einen weißen Linienschnitt vom Objektumriss.

Um die spätere Verarbeitung der Bilder zu vereinfachen, wird die Messvorrichtung abgedunkelt und das Objektiv der Kamera so eingestellt, dass nur noch die von dem Laser beleuchteten Punkte in dem Bild scharf zu erkennen sind. Gegebenenfalls könnte hierzu auch ein monochromatischer Filter eingesetzt werden. Monochromatische Filter sind Filter, die vor das Objektiv einer Kamera geschraubt werden können und nur einen sehr kleinen Bereich der Wellenlängen des sichtbaren Lichtes durchlassen.

Das Ergebnis einer einzelnen Aufnahme ist ein Bild, wie es in den Abbildungen 5.9b und 5.11 gezeigt ist.



(a) Testobjekt, bei normalem Umgebungslicht mit hoher Empfindlichkeit aufgenommen



(b) Testobjekt, in abgedunkeltem Scanner mit Messeinstellungen aufgenommen

**Abbildung 5.9.:** Beispielbilder Objektvermessung

## 5.3. Bildanalyse

Nach der Aufnahmephase werden in der Bildanalyse die Punkte auf der Laserlinie präzise und schnell erfasst.

Um die Erfassungsgeschwindigkeit zu steigern ist eine Bildvorverarbeitung vor jeder Bildanalyse von Nöten. In dieser wird ein so genanntes „Clipping“ durchgeführt. Clipping bezeichnet das Betrachten eines bestimmten Bereiches, der die zu verarbeitenden Informationen enthält. Dieser Bereich, der als „Region of Interest“ bezeichnet ist, wird vorab der Objektvermessung durch ein mit vier Punkten aufgespanntes Polygon gebildet. Der Geschwindigkeitsgewinn liegt in der nachfolgenden reduzierten Anzahl an Bildpixeln über die iteriert wird.

Die Detektion der Laserlinie im Bild startet mit einer Iteration über die Bildpixel beginnend in der linken oberen Ecke der „Region of Interest“. Bei jedem Pixel wird überprüft, ob dieser einen Helligkeitsschwellwert übersteigt. In dem Fall, dass der Helligkeitwert des Pixels größer als der Schwellwert ist, wird er als erkannter Punkt markiert. Andernfalls ist der Punkt nicht weiter von Bedeutung und wird ignoriert.

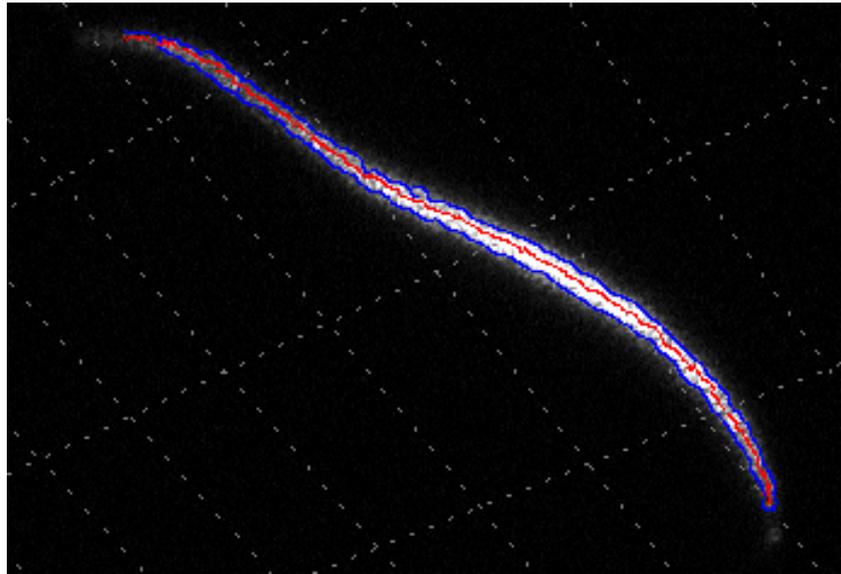
Ein vor dem Vermessungsstart einstellbarer Reflexionsfilter überprüft im aktivierten Zustand den Pixel im Fall einer Markierung, um eine weitere Bedingung. Da ein Punkt durch eine Reflexion o.ä. entstehen kann, findet an dieser Stelle eine Überprüfung der umliegenden Pixel statt, um eine Bildstörung auszuschließen.

Im Nachhinein werden mit Hilfe aller markierten Punkte, die Mittelpunkte der Laserlinie ermittelt. Im letzten Abschnitt der Bildakquisition müssen diese erkannten Mittelpunkte von der Bildebene in die Laserebene transformiert werden. Dies geschieht, indem jeder erkannte Mittelpunkt der Laserlinie mit der in der Kalibrierung berechneten inversen perspektivischen Transformationsmatrix multipliziert wird.

Wie ein Lichtschnitt im Detail zu analysieren ist, wird im nachfolgenden erläutert.



(a) Originalbild



(b) Analysierter Lichtschnitt

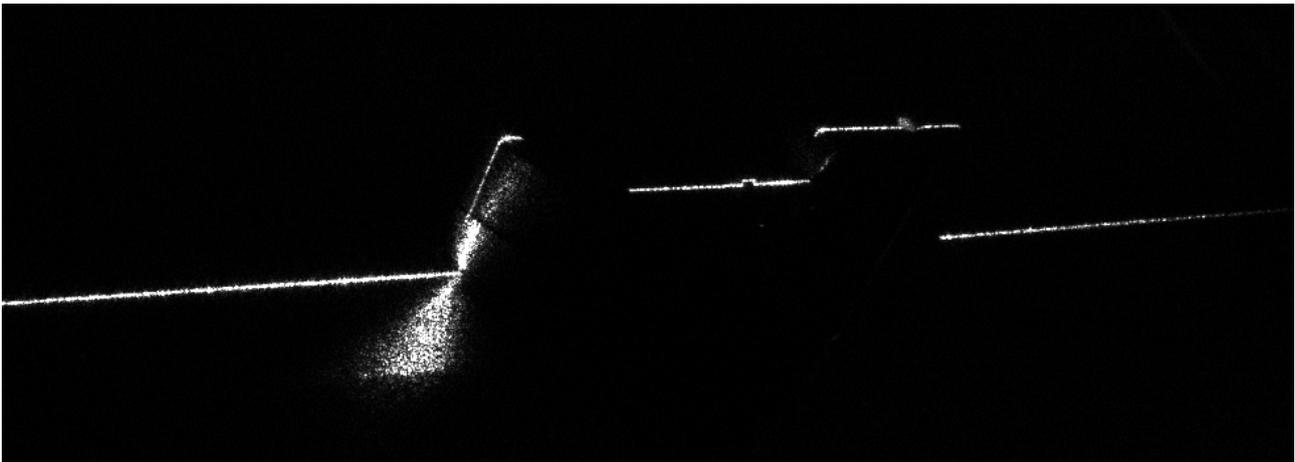
**Abbildung 5.10.:** Analyse eines Lichtschnittes. Die äußeren markierten Punkte stellen die identifizierten Kantenpunkte der Laserlinie dar. Die inneren Punkte kennzeichnen die ermittelten Mittelpunkte.

### 5.3.1. Vorverarbeitung des Kamerabildes

Um die durch den Laser beleuchteten Punkte im Kamerabild zu finden, wird aus dem Quellbild zunächst mithilfe des Canny-Algorithmus ein Kantenbild erzeugt.

Da es bei ungünstigen Bedingungen je nach aktueller Ausrichtung und Oberflächenbeschaffenheit des Objektes zu Bildstörungen durch diffuse Reflektionen kommen kann, wird basierend auf diesem Kantenbild zunächst eine Vorverarbeitung bzw. Filterung des Bildes durchgeführt. Dabei wird für jedes einzelne der Pixel, die als Kantenpunkte gefunden wurden, überprüft, ob es sich dabei tatsächlich um einen Kantenpunkt der Laserlinie handelt oder ob der Punkt aus einer Bildstörung resultiert.

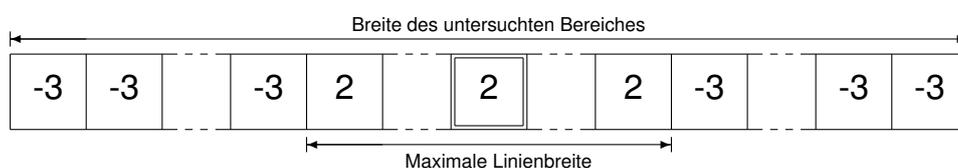
Die Grundidee des hierzu entwickelten Verfahrens ist, dass die Umgebung von jedem der gefundenen Kantenpunkte auf ihre Eigenschaften bezüglich der Linienhaftigkeit untersucht wird. Die Auswertbarkeit der abgebildeten Laserlinie hängt maßgeblich von der Breite ihrer Abbildung ab. Sie ist nur dann sinnvoll auswertbar, wenn sie eine bestimmte Breite von wenigen Pixeln nicht überschreitet. Die durch diffuse Reflektionen verursachten Störungen, wie sie beispielsweise in Abbildung 5.11 auf S. 55 gezeigt sind, lassen sich ebenfalls an größeren Bereichen mit hoher Intensität erken-



**Abbildung 5.11.:** Beispiel für diffuse Reflektionen

nen. Es werden also diejenigen Pixel von der Weiterverarbeitung ausgeschlossen, die an der Kante eines zu großen Bereiches mit hoher Intensität liegen.

Um dies zu erreichen, wird die nähere Umgebung von jedem der gefundenen Kantenpunkte untersucht. Hierzu werden, ähnlich wie bei der Kalibrierung, zunächst die Intensitätswerte einer bestimmten Anzahl Pixel entlang der Gradientenrichtung mittels bilinearer Interpolation berechnet. Auf die so berechnete Bildzeile wird nun der in Abbildung 5.12 dargestellte Filterkern angewendet. Das Ergebnis dieser Faltung wird jedoch nicht, wie bei der linearen Filterung üblich, in ein Zielbild geschrieben, sondern als Bewertungszahl für das aktuelle Pixel verwendet. Ist dieser Wert kleiner oder gleich Null, so wird das Pixel nicht als Kantenpunkt der Laserlinie gewertet und in dem Kantenbild auf Null gesetzt. Ist der Wert größer Null, so wird der Punkt beibehalten.

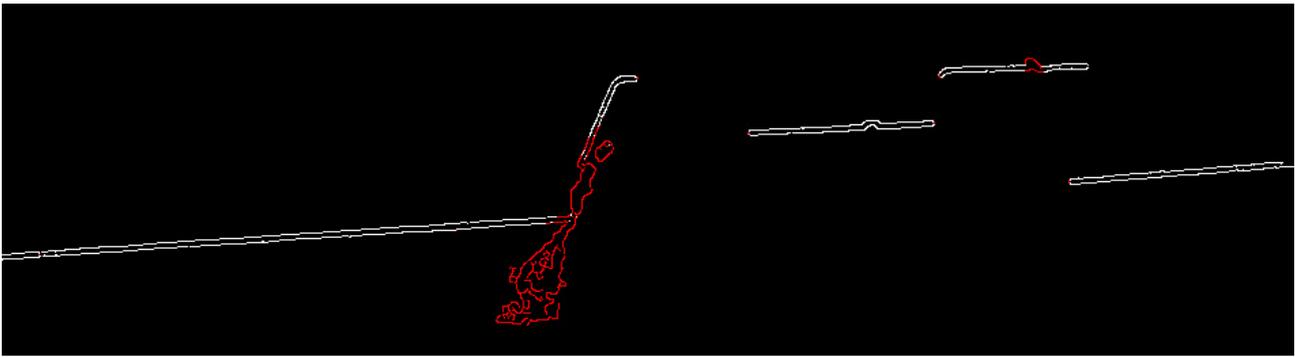


**Abbildung 5.12.:** Filtermaske zur Unterdrückung von diffusen Reflektionen

Das Ergebnis der Anwendung dieses Filters ist in 5.13 dargestellt.

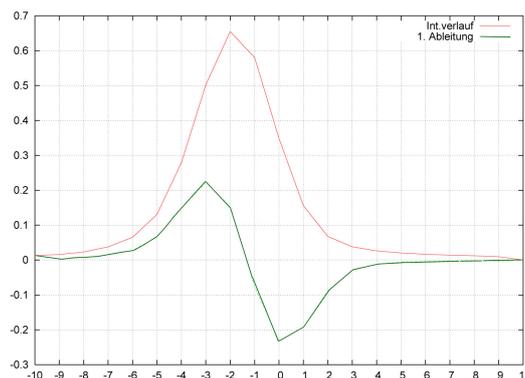
### 5.3.2. Extraktion der Oberflächenpunkte aus dem Bild

Nachdem die Vorverarbeitung durchgeführt wurde, werden nun für alle nicht gefilterten Kantenpunkte die Mittelpunkte der Laserlinie bestimmt. Hierzu werden in den bereits im vorigen Schritt für jeden der Kantenpunkte berechneten Linienquerschnitten



**Abbildung 5.13.:** Beispiel Reflektionsfilter: GRAU: gefilterte Kantenpunkte; WEISS: beibehaltene Kantenpunkte

(siehe Abbildung 5.14 auf S. 56) die genauen Positionen des Maximums bestimmt. Diese lässt sich durch Bestimmung der Nullstelle der diskreten ersten Ableitung der Intensitätswerte berechnen. Die subpixelgenaue Position der Nullstelle wird per linearer Interpolation berechnet.



**Abbildung 5.14.:** Beispiel für einen Querschnitt durch die Laserlinie: OBERE FUNKTION: Intensitätsverlauf; UNTERE FUNKTION: diskrete erste Ableitung der Intensitätswerte

### 5.3.3. Berechnung der Koordinaten in der Laserebene

Im letzten Schritt müssen nun noch aus den im Kamerabild ermittelten Koordinaten  $\vec{c}$  die entsprechenden Koordinaten  $\vec{l}$  der Laserebene berechnet werden. Hierzu müssen die Koordinaten lediglich mit der durch die Kalibrierung bestimmten Transformationsmatrix  $\mathbf{B}$  multipliziert (vgl. Gleichung 5.3 in Abschnitt 5.1.3) und das Ergebnis in euklidische Koordinaten umgewandelt werden:

$$\vec{l}^* = \mathbf{B}\vec{c}^* \quad \text{mit } \vec{c}^* = \begin{pmatrix} c_x & c_y & 1 \end{pmatrix}^T \quad (5.7)$$

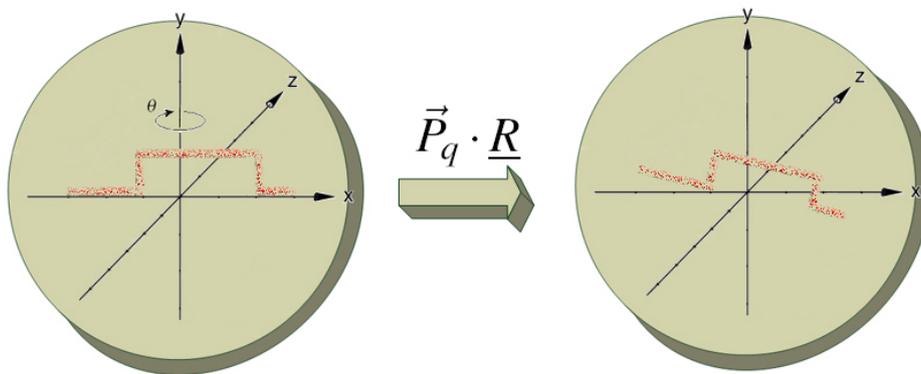
## 5.4. Rekonstruktion der 3D-Koordinaten

Die vollständige Rekonstruktion der 3D-Koordinaten erfordert eine weitere Transformation. In der Bildaufnahmephase wird zu jedem Bild der Drehwinkel  $\theta$  zwischengespeichert. Diese Bildinformation wird verwendet, um die erkannten Koordinatenpunkte aus der Bildanalysephase an die richtige Position im Weltkoordinatensystem zu positionieren. Mit Hilfe einer Rotationsmatrix  $\underline{R}$  werden die erkannten Koordinaten  $P_q(x_q, y_q, 1)$  mit einer Rotation um die Y-Achse des Weltkoordinatensystems an die richtige Position  $P_z(x_z, y_z, z_z)$  rotiert:

$$\vec{P}_z = \begin{bmatrix} x_z \\ y_z \\ z_z \end{bmatrix} = \vec{P}_q \cdot \underline{R} = \begin{bmatrix} x_q \\ y_q \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (5.8)$$

Die Herleitung der Rotationsmatrix wurde im Kapitel 3.8 auf Seite 14 beschrieben.

Die Nachfolgende Abbildung 5.15 stellt diese Rotation grafisch dar.



**Abbildung 5.15.:** Rekonstruktion der 3D-Koordinaten.

In der rechten Grafik sind die um  $\theta$  rotierten Koordinatenpunkte eines Lichtschnittes dargestellt.

Die somit ermittelten 3D-Koordinaten beschreiben die Oberflächenstruktur des zu vermessenden Objektes.

Dabei bilden alle 3D-Koordinaten eine Koordinatenmenge, die als so genannte „Punktwolke“ bezeichnet wird. Die beschreibenden Informationen der Punktwolke sind in Abbildung 5.16 dargestellt.

```
# result file. contained dataformat 'x y z' image: 1 degree: 1
-12.247485 27.082899 0.786372
-12.300716 27.116789 0.785443
...
# result file. contained dataformat 'x y z' image: 2 degree: 2
-12.932054 27.027191 0.549012
-12.655300 27.342342 0.558677
...
# result file. contained dataformat 'x y z' image: 3 degree: 3
-12.622746 27.447149 0.339842
-12.838274 26.924780 0.328547
...
# result file. contained dataformat 'x y z' image: 360 degree: 360
-13.240198 27.132374 1.000000
-12.715452 27.519621 1.000000
...
```

**Abbildung 5.16.:** Formatierung der 3D-Datenmenge.

## 5.5. Registrierung

Die geometrischen Eigenschaften eines zu vermessenden Objektes lassen es nicht immer zu, dass ein Objekt in einem Scanvorgang vollständig vermessen werden kann.

Beispielhaft sei ein Spielzeugauto auf einem Drehteller positioniert. Dabei sind die Konturen durch die Rotation des Drehtellers von oben und von den Seiten von dem Laser beleuchtet und von der Kamera gut zu erfassen. Der Fahrzeuguntergrund kann nicht vom Laser beleuchtet und des Weiteren auch nicht von der Kamera erfasst werden. Eine vollständige Rekonstruktion der Oberflächenstruktur ist daher nicht mit einem einzigen Scanvorgang möglich. Um den Fahrzeuguntergrund und damit alle Konturen des Objektes erfassen zu können, wird eine zweite Objektvermessung durchgeführt. In dieser wird das Auto auf dem Kopf positioniert.

Die Notwendigkeit für mehrere Objektvermessungen kann ebenfalls durch Anforderungen an das zu vermessende Objekt gegeben sein. Eine Anforderung ist zum Beispiel die hochgenaue Erfassung der Türen und Scheinwerfer eines Kraftfahrzeuges. In diesem Fall bieten sich für diese Fahrzeugkomponenten Nahaufnahmen an, die in einer Registrierung anschließend zusammengefasst werden.

Zwei Registrierungsverfahren zur Vereinigung von überlappenden Oberflächenscans werden weitestgehend unterschieden. Bei der manuellen Registrierung wird mit Hilfe

von Passpunkten gearbeitet. Dieses Verfahren kann nur produktiv angewendet werden, wenn mindestens drei korrespondierende Passpunktpaare in den zu registrierenden Punktwolken existieren, und diese ein Dreieck bilden. Um eine gute Vereinigung der Punktwolken zu erzielen, sollten möglichst viele Passpunkte in den Punktwolken markiert werden. Bei den Triangulations- und Laufzeitverfahren ist diese stabile Registrierungsart derzeit das Standardverfahren.

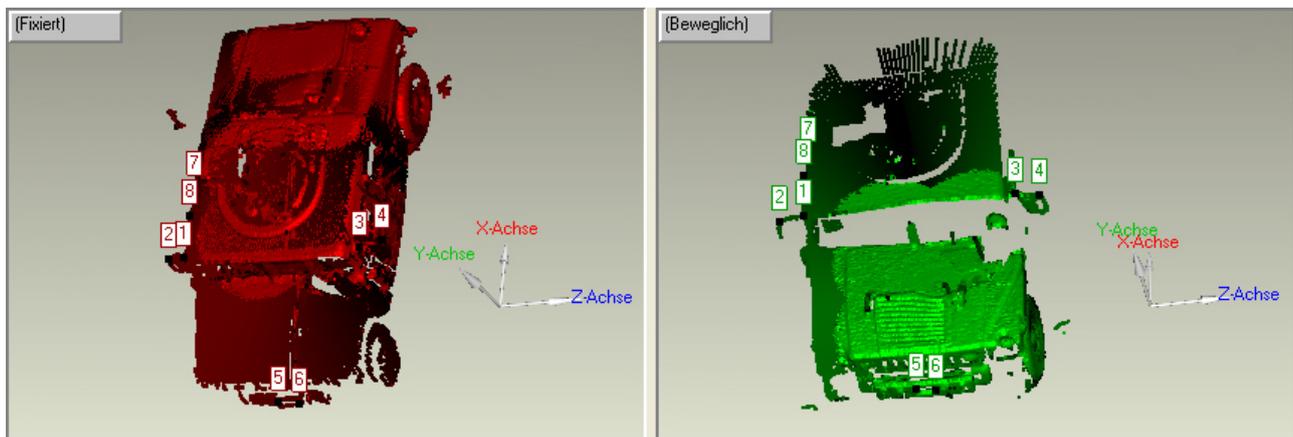
Das zweite Verfahren um Punktwolken zu vereinigen, ist die automatische Registrierung. Mit Hilfe von Algorithmen werden charakteristisch übereinstimmende Punktpaare bzw. Merkmale in den Punktwolken gesucht, um damit anschließend die Registrierung durchzuführen. Der populärste Algorithmus ist der „Iterative Closest Point (ICP)“ Algorithmus auf dem zahlreiche Weiterentwicklungen basieren. Vorgestellt 1992 von Paul J. Besl und Neil D. McKay. Informationen über den Algorithmus und über Weiterentwicklungen finden sich in (Besl und McKay, 1992), (Maurer u. a., 1996) und (Masuda und Yokoya, 1994).

Bei der Registrierung von zwei Punktwolken  $P_1$  und  $P_2$  versucht der „Iterative Closed Point“ Algorithmus zu jedem Punkt aus der Punktmenge  $P_2$  den korrespondierenden Punkt in der Punktmenge  $P_1$  zu finden. Mit Hilfe der korrespondierenden Punkte wird eine Transformation  $\underline{T}$ , die aus einer Rotation und einer Translation besteht, berechnet. Diese Transformation ermöglicht die Registrierung der Punktwolke  $P_2$  in  $P_1$ . Somit wird der Algorithmus in drei Phasen eingeteilt. Diese werden iterative durchlaufen:

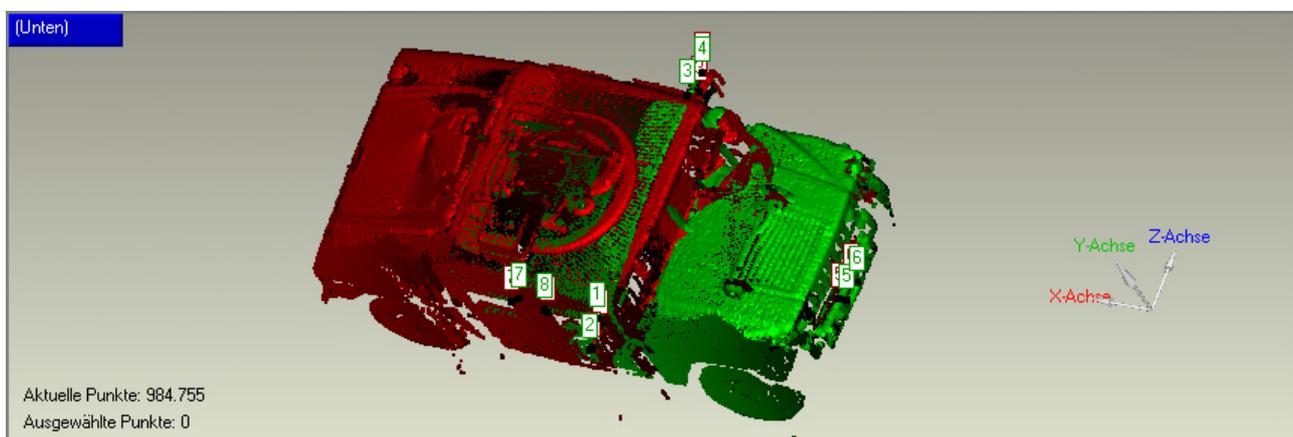
1. Suche nach den korrespondierenden Punkten aus  $P_1$  und  $P_2$
2. Berechnung der Transformation  $T$  zur Registrierung
3. Anwendung der errechneten Transformation  $T$

Der Algorithmus terminiert wenn der Fehler der letzten Iteration einen Schwellwert unterschreitet oder die maximale Anzahl an Iteration erreicht wird. In der Praxis unterscheiden sich die Abbruchkriterien.

Im Nachfolgenden ist eine mit dem Softwareprodukt „Geomagic Studio 9“ durchgeführte manuelle Registrierung von zwei Oberflächenscans eines Spielzeugautos dargestellt. Für softwarespezifische Informationen wird auf (Geomagic, 2009) verwiesen.



**Abbildung 5.17.:** Korrespondierende Passpunkte müssen vor der manuellen Registrierung markiert werden.



**Abbildung 5.18.:** Ergebnis einer manuellen Registrierung zweier Oberflächenscans zu einer Punktwolke.

## 5.6. Vernetzung

Die Vernetzungsphase befasst sich mit der Erzeugung eines Polygonnetzes, welches im Sprachgebrauch häufig auch als „Mesh“ oder „Wireframe“ bezeichnet wird. Gebräuchliche Polygone bestehen aus drei, vier oder fünf Eckpunkten. Ein Eckpunkt wird im Singular als „Vertex“ und im Plural als „Vertice“ bezeichnet.

In dieser Arbeit wurde das Programm „Point Cloud“ der Firma SYCODE in die entwickelte Software integriert. Produktinformationen finden sich auf (Sycode, 2009). Es ermöglicht eine Erzeugung eines Polygonnetzes, bestehend aus Polygonen mit jeweils drei Eckpunkten. Dazu werden alle Punkte einer Punktmenge analysiert und über eine Dreiecksvermaschung zu einem Polygonnetz schrittweise aufgebaut. Die

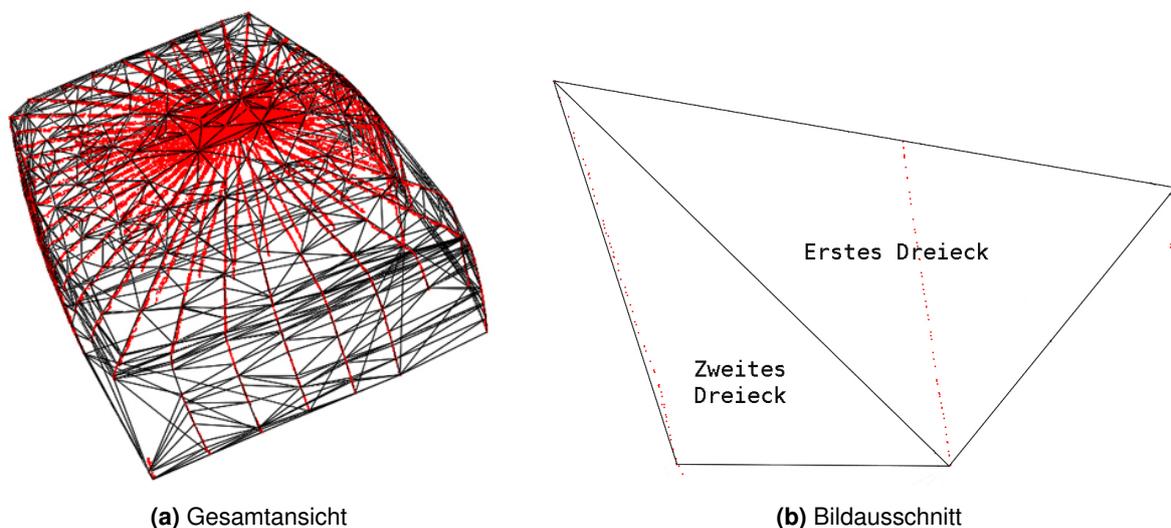
Dreiecksvermaschung wird ebenfalls als Triangulierung bezeichnet. Für die Vermaschung wird zum Beispiel die räumliche Delaunay-Triangulation verwendet. Detaillierte Informationen zur räumlichen Delaunay-Triangulation sind in (Fang und Piegler, 1995) und (Cignonia u. a., 1998) nachzuschlagen.

In dem Fall, dass eine Vernetzung mit aus drei Punkten bestehenden Polygonen durchgeführt werden soll, geht die 2D-Delaunay-Triangulation folgendermaßen vor.

Drei Punkte die ein Dreieck bilden werden gesucht. Dabei dürfen innerhalb eines Kreises, auf dem diese Punkte liegen, keine anderen Punkte vorhanden sein. Diese entstehenden Dreiecke des Dreiecksnetzes erfüllen somit alle die so genannte Umkreisbedingung. Dadurch wird der kleinste Innenwinkel über alle Dreiecke maximiert.

Die räumliche Delaunay-Triangulation baut auf der 2D-Delaunay-Triangulation auf. Dort besteht statt der Umkreis- eine Umkugelbedingung, welche aus jeweils vier Punkten ein Tetraeder bildet.

Die Abbildung 5.19 stellt das Ergebnis einer vernetzten Punktwolke dar.



**Abbildung 5.19.:** Resultat einer Dreiecksvermaschung.

## 5.7. Modellierung

Nachdem die Grundlage zur Flächenerzeugung geschaffen wurde, können aufbauend auf einem Polygonnetz, Flächen modelliert werden. Dazu werden Polygone gesucht, die Flächen lokal approximieren. Dabei ist es das Ziel den geometrischen Fehler der Approximation möglichst zu minimieren. Wissenswertes über die Modellierung

von Punktwolken beschreibt das Paper „Computing and Rendering Point Set Surfaces“ in (Alexa u. a., 2003) anschaulich.

Mit Hilfe des integrierten Programms „Point Cloud“ ist es möglich Polygonnetze modelliert darzustellen. Zwei Arten der Modellierung sind dabei verfügbar und zu unterscheiden. Auf der einen Seite kann ein konvexes Polygonnetz mit einer Wölbung nach außen und zum anderen ein konkaves Polygonnetz mit einer Wölbung nach innen erzeugt werden.

Ein Grauwert kennzeichnet den Helligkeits- oder Intensitätswert eines einzelnen Oberflächenpunktes. Grauwertinformationen waren aufgrund der notwendigen kompletten Abdunkelung des Laserscanners während der Vermessung nicht zu erfassen, sodass auch keine Helligkeitsintensitäten der Punkte im „Point Cloud“-Programm Verwendung fand. Dadurch wird das in der Modellierungsphase entstehende Flächenmodell in einer manuell ausgewählten Farbe dargestellt. Realisierbar wäre für jede Objektvermessung ein zweiter simultaner Scanvorgang ohne vollständige Lichtabschottung. Dies würde die Komplexität und Geschwindigkeit für jede komplette Objektvermessung ansteigen lassen. Es könnten aber Helligkeitsinformationen mit der in dieser Arbeit verwendeten Schwarz/Weiß-Kamera mit erfasst werden. Eine Farbkamera wäre mit diesem Prinzip sogar in der Lage, Farbinformationen bezüglich der Objektoberfläche zu erfassen, um mit diesen zusätzlichen Informationen die Originaltextur des Objektes farbig zu rekonstruieren.

## **5.8. Nachbearbeitung**

In der Nachbearbeitungsphase werden, um dem ursprünglichen Objekt noch genauer zu entsprechen, Veränderungen an Punktwolken, Polygonnetzen oder modellierten Modellen durchgeführt. Die fünf Nachbearbeitungsphasen werden nachfolgend erläutert.

### **5.8.1. Beseitigung von Punktstörungen**

Vereinzelnde Punkte die nicht zum vermessenden Objekt gehören, treten immer wieder in Messergebnisse auf. Diese Störungen können leicht erkannt und entfernt wer-

den. Dafür stellt „Geomagic Studio 9“ Operatoren zur Verfügung, mit dem sich falsche Messwerte manuell markieren und entfernen lassen.

### 5.8.2. Lochbeseitigung

Die Beseitigung von Löchern in der Oberflächenstruktur wird häufig als „Hole Filling“ bezeichnet. Oft wird nach dem Scanvorgang festgestellt, dass bestimmte Bereiche nicht vom Scanner erfasst wurden. Diese entstandenen Löcher im Messergebnis der Oberflächenstruktur sind zum Beispiel auf Abschattungen zurückzuführen.

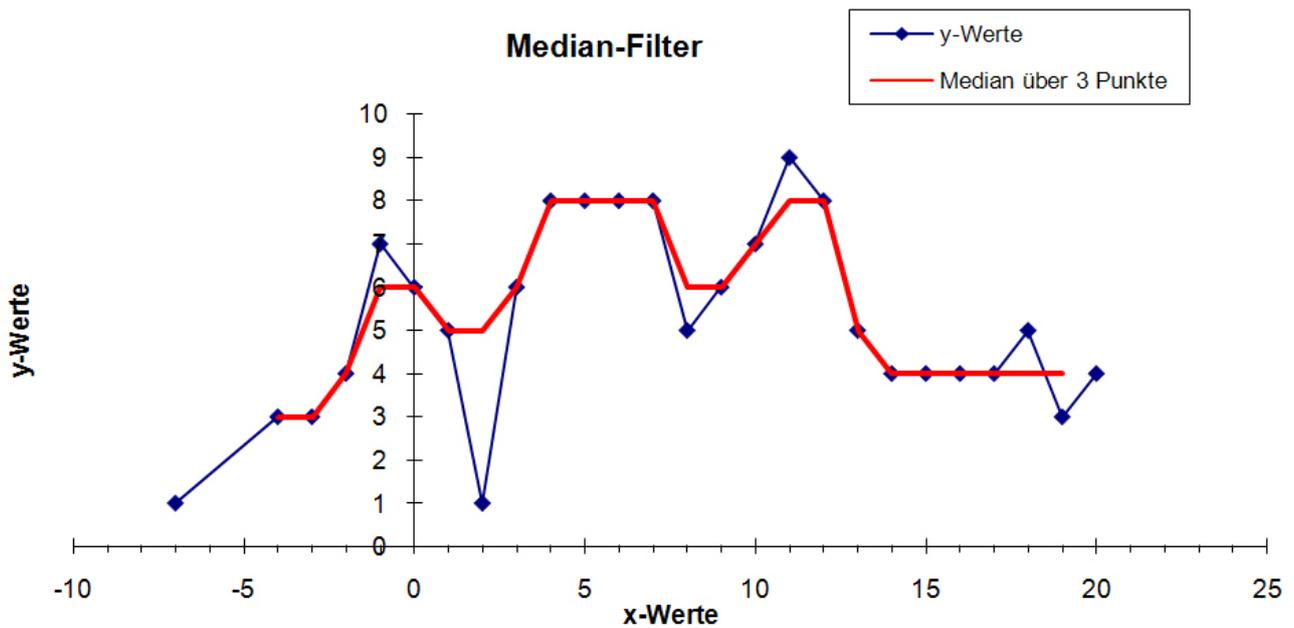
„Geomagic Studio 9“ ermöglicht die manuelle und automatische Lochbeseitigung. Manuell können Punkte zur Lochfüllung einzeln gesetzt werden. Da dies ziemlich viel Zeit in Anspruch nimmt, bietet sich die automatische präzise Lochbeseitigung an. Dazu muss nur ein Bereich detektiert werden, der den zu füllenden Bereich enthält.

### 5.8.3. Glätten der Oberfläche

Auf dem Markt existieren zahlreiche Glättungsverfahren, die je nach Oberflächenstruktur des Polygonnetzes ihre Verwendung finden. Sie werden angewendet um Kanten und Ecken eines Polygonnetzes feiner darzustellen. Zu den bekannten Glättungsverfahren gehören Laplace Smoothing, Anisotropische Diffusion, Multiresolution Analysis, Mean Curvature Flow, Gauß-, Median-, Wiener-, Bilateral Filter und weitere.

In dem Fall, dass viele Ausreißer in einer Punktmenge existieren, wird der Median-Filter vorzugsweise, ohne Verschmierungseffekte eingesetzt. In der Bildverarbeitung wird der Median einer Bildpixelmenge folgendermaßen ermittelt. Zuerst werden alle Helligkeitswerte der Bildpixelmenge aufsteigend sortiert. Anschließend wird der Wert in der Mitte, der sortierten Bildpixelmenge, ausgewählt. Dieser ist der Median.

Die in Abbildung 5.20 dargestellte Grafik stellt die Wirkung eines Median-Filters dar. Dabei wurde der Median jeweils aus drei benachbarten Werten berechnet.



**Abbildung 5.20.:** Einwirkung eines Median-Filters.

Auf Polygonnetze bezogen wird statt der Helligkeitswerte die Richtung der Einheitsnormalen verwendet. Somit erfolgt eine Änderung der Flächennormalen.

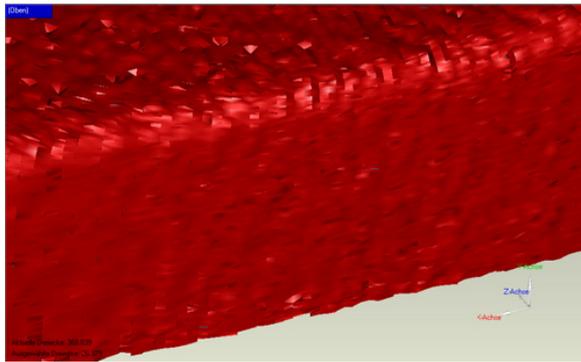
Beim Glättungsverfahren ist es besonders wichtig, dass die geometrischen Merkmale wie Kanten und Ecken erhalten bleiben. Das vermessene Objekt soll nicht vom Originalobjekt verändert, sondern von Störungen befreit werden.

Die in Abbildung 5.21 dargestellten Grafiken zeigen ein mit dem Softwareprodukt „Geomagic Studio 9“ angewandtes Glättungsverfahren. Erkennbar sind der Erhalt der Kanten und die erfolgreiche Oberflächenglättung.

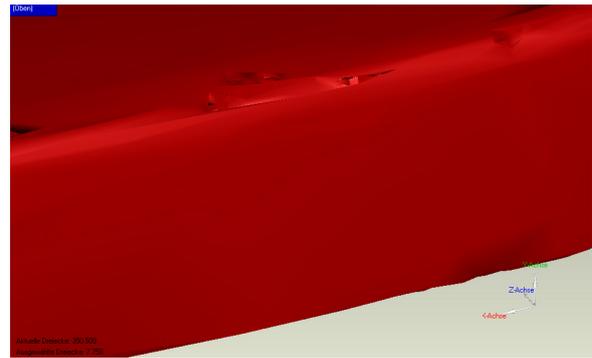
Nähere Informationen über gängige Glättungsverfahren und eine intensive Befassung mit einer Erweiterung des Bilateralen Filters finden sich in (Haase, 2005) und (Jones, 2003).

#### 5.8.4. Simplifizierung

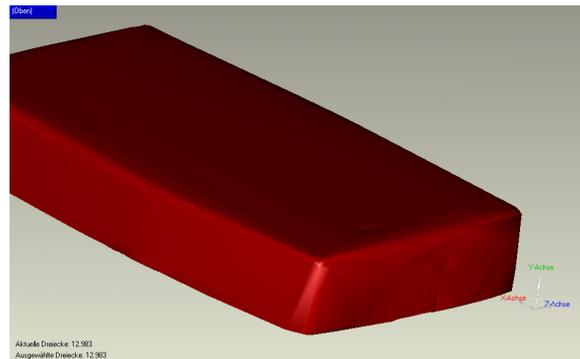
Die Simplifizierung befasst sich mit der Reduzierung einer Punktmenge, die eine Punktwolke, ein Polygonnetz oder ein modelliertes Modell beschreibt. Einzelne Flächen eines modellierten Modells werden durch Eckpunkte beschrieben. Eine Reduzierung dieser Punktmenge wird, indem breite und glatte Flächen mit weniger Polygonen beschrieben werden, erzielt. So können zum Beispiel zwei Flächen die je-



(a) Kante vor der Glättung



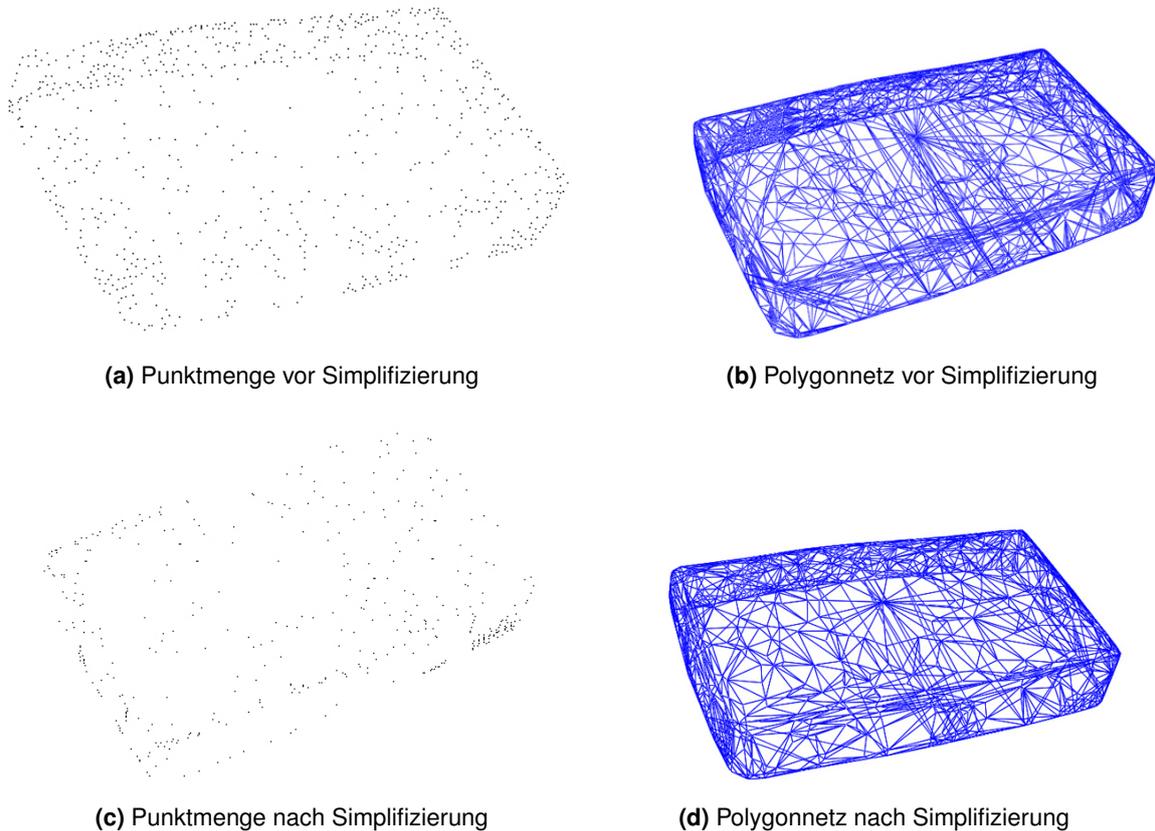
(b) Kante nach der Glättung



(c) Geglättetes Modell

**Abbildung 5.21.:** Ergebnis eines erfolgreich angewandten Glättungsverfahrens.

weils durch drei Eckpunkte definiert sind, auf der gleichen Ebene liegen und derart positioniert sind, dass eine Vereinigung beider Flächen zu einer gemeinsamen Flächenbeschreibung führt, von sechs auf drei Beschreibungspunkten reduziert werden. Dabei erfolgt die Flächenreduzierung mit geringer oder sogar ohne Verlust an Genauigkeit. Weitere Informationen bezüglich Simplifizierungsverfahren sind in (Keßler, 2008) nachzuschlagen.



**Abbildung 5.22.:** Simplifizierung einer Punktmenge.

Die Punktmenge, die ein Polygonnetz beschreibt sind in Abbildung a und c zu sehen. Es ist gut zu erkennen, dass sich mit einer reduzierten Punktmenge (Abbildung c) ein gleich gutes Polygonnetz erzeugen lässt.

### 5.8.5. Formatierung der Punktmenge

Im letzten Schritt des Realisierungsprozesses wird die Datenmenge, die eine Punktwolke, ein Polygonnetz oder ein modelliertes Modell beschreibt, für die Weiterverarbeitung in anderweitigen Softwareprodukten formatiert abgespeichert.

Damit die objektbeschreibende Punktmenge in „Point Cloud“, „Geomagic Studio“, „AutoCAD“ und in weiteren Softwareprodukten Weiterverwendung finden kann, werden in dieser Phase die Beschreibungsinformationen des Objektes, wie in Abbildung 5.23 dargestellt abgespeichert.

---

```
-12.247485 27.082899 0.786372  
-12.300716 27.116789 0.785443  
-12.121478 27.036463 0.788571  
-12.086360 27.025757 0.789184  
-11.971641 27.002216 0.791187  
-11.826903 26.986923 0.793713  
-11.935146 26.711134 0.791824
```

**Abbildung 5.23.:** Formatierung der 3D-Datenmenge.



# 6. Implementierung

In diesem Kapitel ist die in dieser Arbeit entwickelte Software zur Vermessung von Gegenständen detailliert beschrieben.

Die Software wurde komplett in C++ geschrieben, arbeitet pfadunabhängig und hat eine grafische Benutzeroberfläche, die zur Bedienung des Laserscanners und zur Verarbeitung der Messdaten dient.

## Inhalt

---

6.1. Betriebssystemunterstützung . . . . .	69
6.2. Grafische Oberfläche . . . . .	70

---

## 6.1. Betriebssystemunterstützung

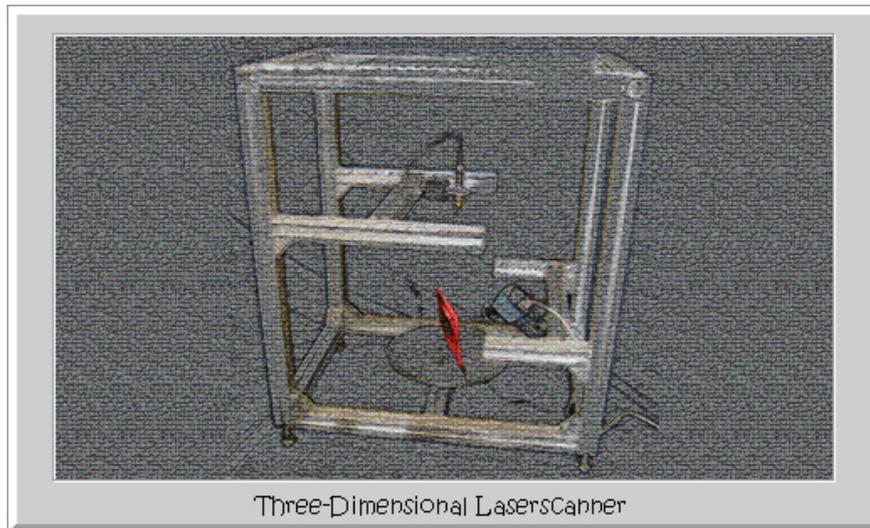
Das Softwareprodukt wurde so entworfen und entwickelt, dass es auf allen Windows Betriebssystemen einwandfrei funktionieren sollte. Die in der Tabelle 6.1 aufgelisteten Betriebssysteme wurden mit der entwickelten Software erfolgreich getestet.

Betriebssystem
Windows XP Professional (SP3)
Windows XP Media Center Edition (SP3)
Windows Vista Home Premium (SP1)
Windows Vista Business
Windows 7

**Tabelle 6.1.:** Auflistung der getesteten Betriebssysteme.

## 6.2. Grafische Oberfläche

Die Laserscannersoftware wird mit einer ausführbaren Datei gestartet. Daraufhin erscheint ein so genannter „Splash Screen“, der den Laserscanner ohne Verkleidung mit seinen einzelnen Komponenten grafisch darstellt.



**Abbildung 6.1.:** Splash Screen beim Starten der Laserscannersoftware.

Nachdem die Software vollständig geladen ist, erscheint das Hauptmenü. In dem Menü sind vier wichtige Funktionsmöglichkeiten von Bedeutung. Diese sind nummeriert in Abbildung 6.2 dargestellt.

1. Die Nummer eins kennzeichnet die Möglichkeit über „View -> Go to View“ zu jedem gewünschten Ansichtenfenster zu wechseln.
2. Eine komplett auf Doxygen (van Heesch, 2009) basierende Code-Dokumentation wird über den mit der Nummer zwei markierten Weg „Help -> Documentation“ aufgerufen. Mit dieser Dokumentation kann der komplette ausführlich dokumentierte Sourcecode des Laserscanners betrachtet werden. Aufgrund der komplexen Sourcecode-Dokumentation wird in dieser Arbeit nicht detailliert auf den entwickelten Sourcecode eingegangen. Der Sourcecode ist für wissenschaftliche Zwecke über den Autor zu erhalten.
3. Mit dem mit der Nummer drei gekennzeichneten Button wechselt die Ansicht zum Setupbereich des Laserscanners um ein Objekt zu vermessen.

4. Durch das Betätigen des zweiten Buttons auf diesem Ansichtsfenster werden dem Nutzer Scanergebnisse von Objekten, die im Laufe des Entwicklungsprozesses entstanden sind, grafisch dargestellt. Diese sollen einen kleinen Eindruck vermitteln, welche Ergebnisse der Laserscanner erzielen kann.

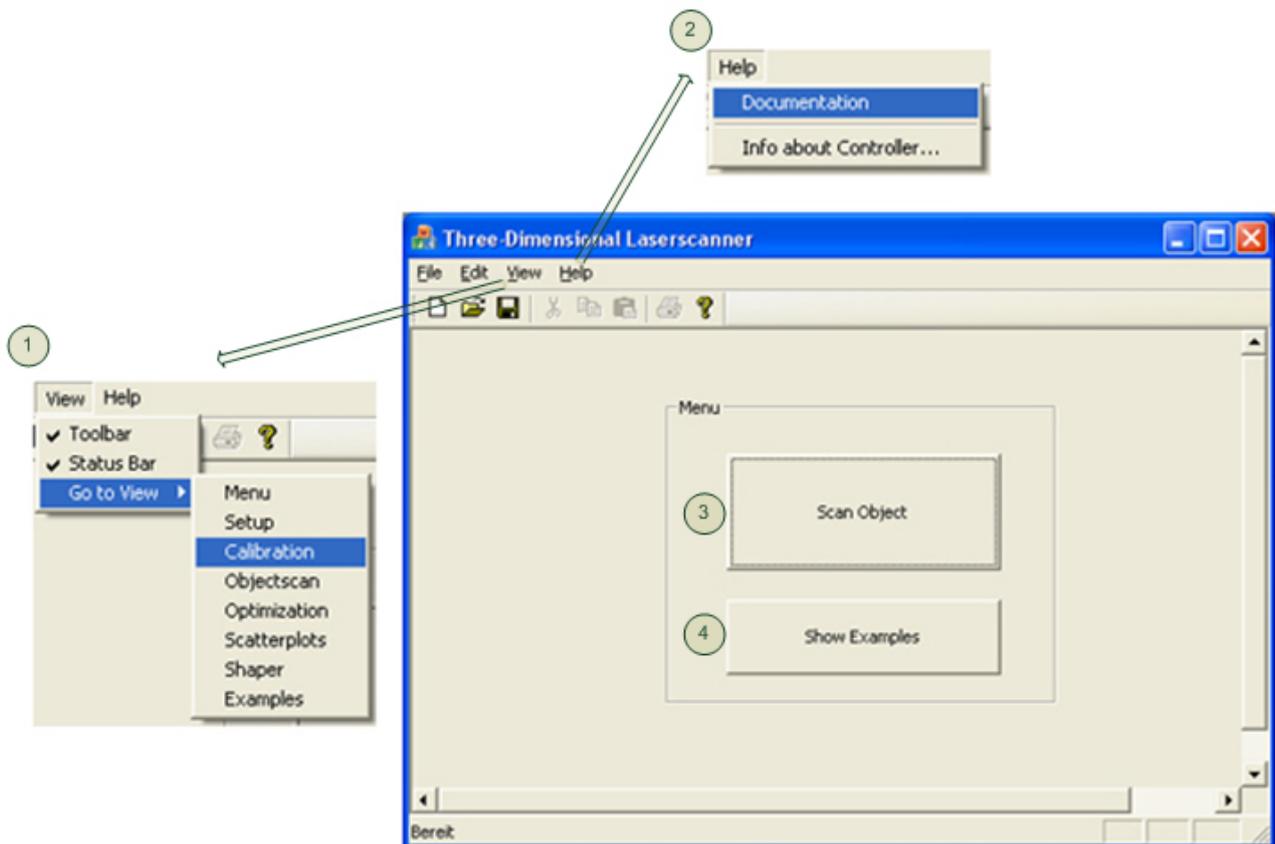
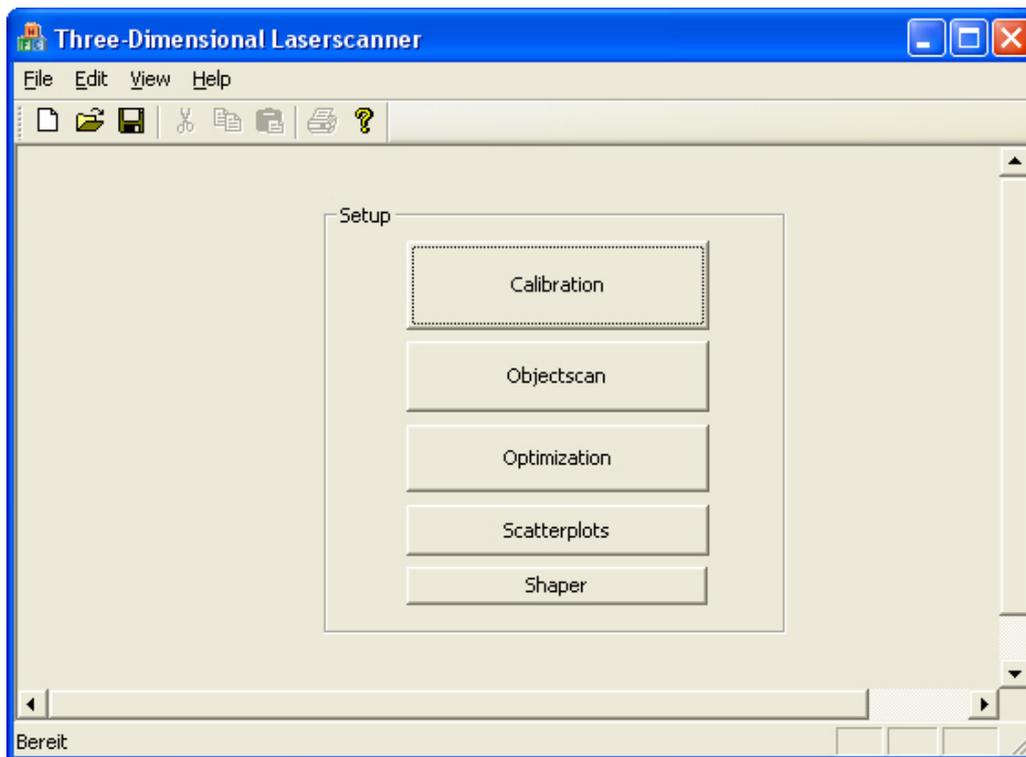
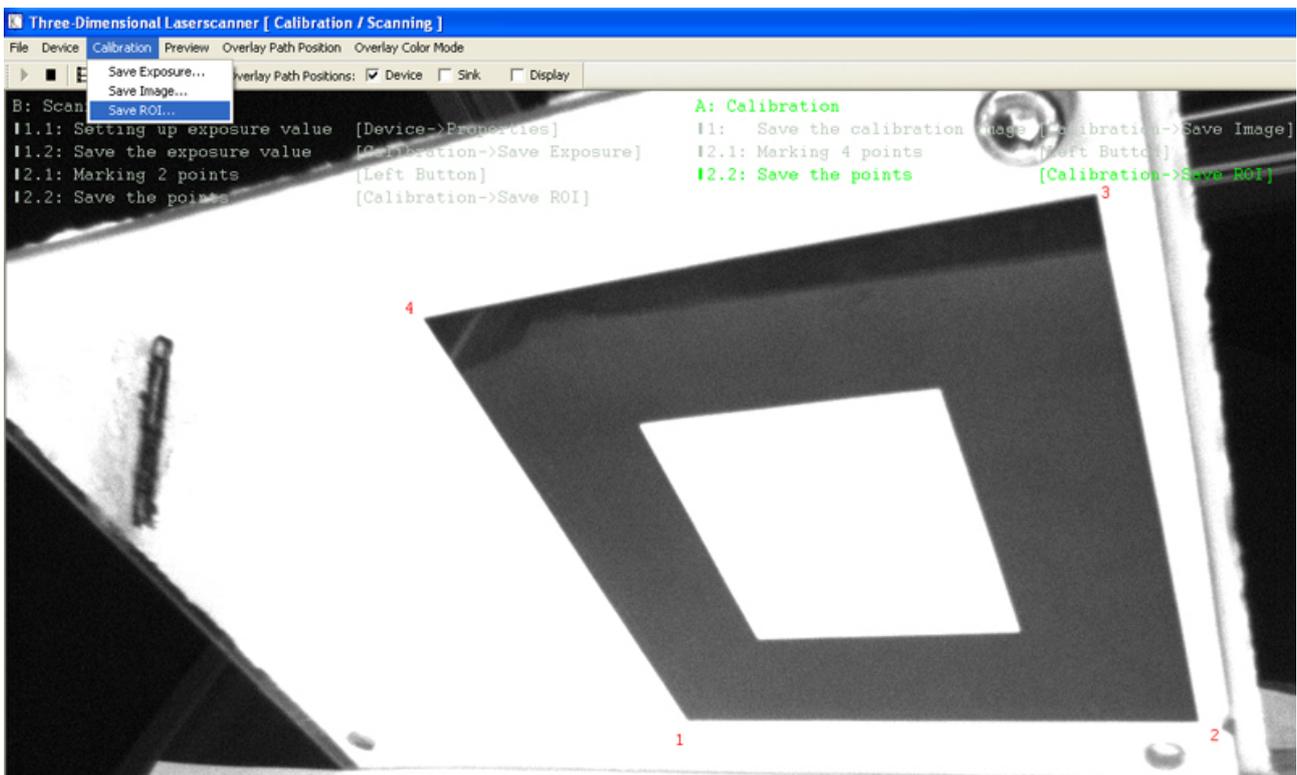


Abbildung 6.2.: Hauptmenü.



**Abbildung 6.3.:** Setupmenü.

Das Setupmenü ist die wichtigste Ansicht im gesamten Programm. Ausgehend von dem obersten- von insgesamt fünf vertikal ausgerichteten Buttons, werden diese fünf Arbeitsabläufe schrittweise nach unten abgearbeitet, um ein komplett modelliertes Objekt zu rekonstruieren. Liegen bereits Kalibrierungsergebnisse, Messdaten oder sogar vollständig modellierte Modelle vor, können auch gewünschte Schritte vom Benutzer übersprungen werden.



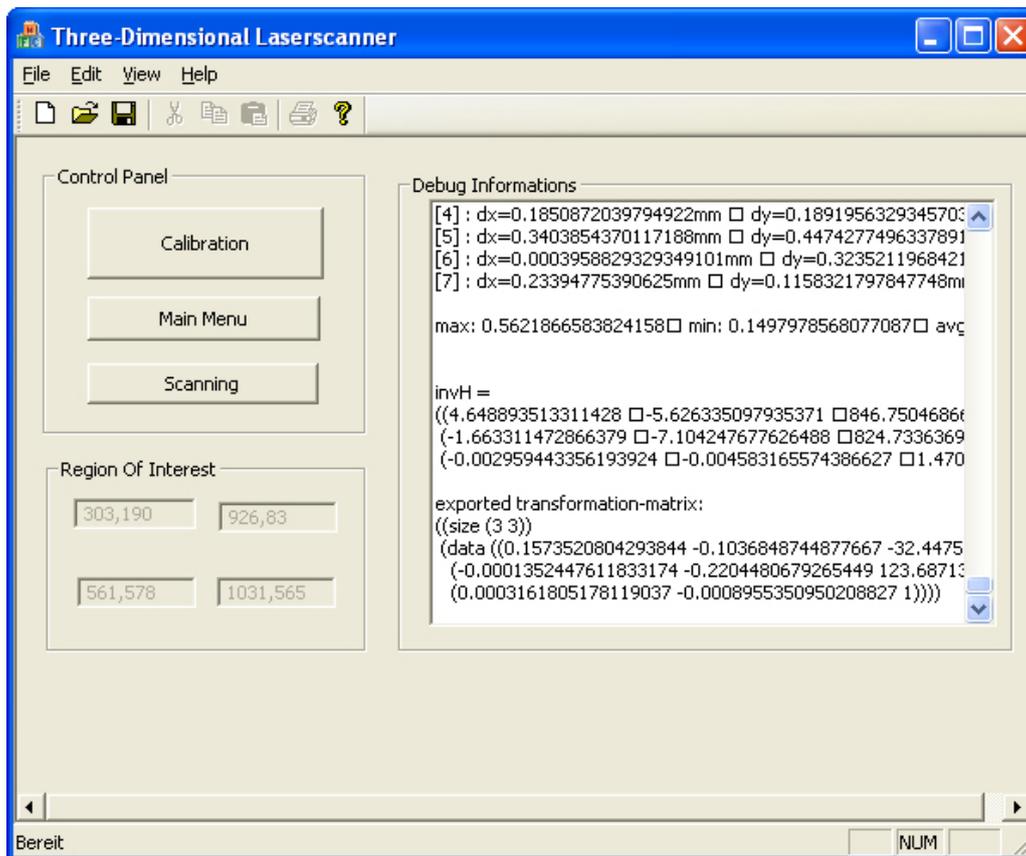
**Abbildung 6.4.:** Ein Kalibrierungsbild zu speichern und eine „Region of Interest“ mit 4 Punkten aufzuspannen sind Aufgaben der Kalibrierungsvorbereitung.

### Stufe 1: Kalibrierung

Der erste Schritt einer vollständigen Vermessung ist die Kalibrierung. Beim Betätigen des Kalibrierungsbuttons wird ein Fenster (Abbildung 6.4) geöffnet, welches beim Anschluss der Laserscanner-Kamera einen Livescreen erscheinen lässt. Im Livestream sollte die zuvor auf dem Drehteller montierte Kalibrierungsplatte mit eingelegtem Kalibrierungsmuster zu sehen sein. Der an dieser Stelle weiterentwickelte Programmcode basiert auf einem von der Firma „The Imaging Source“ entworfenem Sourcecode, der es ermöglicht, einen Livescreen einer Kamera darzustellen und in diesen Text hineinzuschreiben. Die Internetadresse zum Sourcecode, der es ermöglicht einen Livescreen darzustellen, ist in (The-Imaging-Source, 2008b) angegeben. Der Sourcecode wurde in dieser Arbeit so weiterentwickelt, dass der Benutzer textuell schrittweise durch die vorbereitenden Kalibrierungsschritte geleitet wird. Des Weiteren wurde die Möglichkeit eine „Region of Interest“ in den Livestream zu zeichnen entwickelt und integriert. Dazu werden mit der Maus vier Punkte markiert, welche die „Region of Interest“ aufspannen.

Sind alle Kalibrierungsschritte erfolgreich durchgeführt worden, wechselt der Anlei-

tungstext im Livescreen seine Farbe und das Fenster kann geschlossen werden. Das Schließen des Fensters verursacht die Anzeige der eingestellten Kalibrierungsparameter und ermöglicht den Kalibrierungsstart mit dem in Abbildung 6.5 zu sehenden Button, mit der Aufschrift „Calibration“. Daraufhin läuft im Hintergrund der Kalibrierungsprozess, der im Kapitel 5.1 beschrieben wurde. Im Anschluss an dem Kalibrierungsvorgang erscheinen in dem Feld „Debug Informations“ alle berechneten Informationen, wie zum Beispiel die Koeffizienten der Ebene-zu-Ebene perspektivischen Transformationsmatrix.



**Abbildung 6.5.:** Das Ergebnis einer Kalibrierung.

Links unten im Bild sind die 4 Punkte der „Region of Interest“ platziert. Rechts im Bild sind alle wichtigen Informationen einschließlich der Ebene-zu-Ebene perspektivischen Transformationsmatrix des Kalibrierungsprozesses abgelegt.

## Stufe 2: Objektvermessung

Anschließend kann zum nächsten Schritt übergegangen werden. Dieser ist die Objektvermessung (Button „Scanning“). Dort erscheint wieder das Fenster mit einem Livescreen. Der Livescreen sollte das auf dem Drehteller platzierte und zu vermes-

sende Objekt mit einem gut ausgerichteten Laser anzeigen. Zuerst muss der Belichtungswert der Kamera eingestellt und gespeichert werden. Der Belichtungswert bestimmt die Lichtintensität durch die das Objekt belichtet wird. Je höher der Belichtungswert ist, desto länger, breiter und intensiver ist die Beschaffenheit der Laserlinie. Dabei ist zu beachten, dass die Laserlinie optimal so eingestellt wird, dass die Linie circa so lang wie der Objektdurchmesser ist und eine Breite von 2-4 *mm* aufweist.

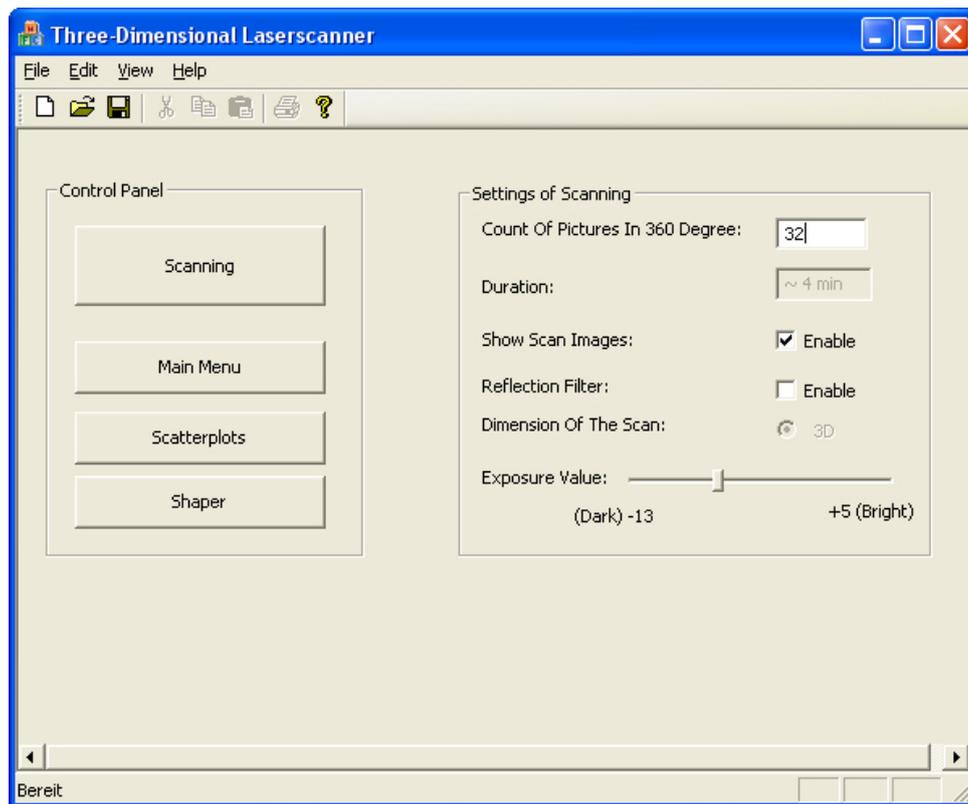
Eine mit „Maple 12“ (Maplesoft, 2009) berechnete Umrechnungsfunktion transformiert den aktuell eingestellten Belichtungswert  $x_q$  in einen anderen Wertebereich, sodass dieser Belichtungswert später in der Objektvermessung geladen und verwendet werden kann. Ausgehend von der Modellfunktion 6.1 wurde mit Maple und den gegebenen Abbildungskordinaten, die Quell- und Zielbelichtungswerte, eine nichtlineare Ausgleichsrechnung durchgeführt. Die nichtlineare Ausgleichsrechnung wurde im Kapitel 3.1.5 auf der Seite 18 beschrieben.

Die im Resultat der Ausgleichsrechnung ermittelten drei Koeffizienten  $a$ ,  $b$  und  $c$  der Modellfunktion sind in der Formel 6.2 zu sehen. Im zweiten Schritt muss wie in den Vorbereitungseinstellungen zur Kalibrierung eine „Region of Interest“ festgelegt werden. In diesem Fall sollte die „Region of Interest“ den Bereich definieren, indem sich das zu vermessende Objekt während der Bildaufnahme befindet. Dieser Bereich sollte unter dem Aspekt der Zeiteffizienz möglichst exakt bestimmt werden, da sonst Bildflächen analysiert werden, die keine zu erfassenden Messpunkte der Laserlinie enthalten.

$$x_z = \frac{a}{\exp^{b \cdot x_q}} + c \quad (6.1)$$

$$x_z = \frac{9.58}{\exp^{0.0259 \cdot x_q}} - 9.27 \quad (6.2)$$

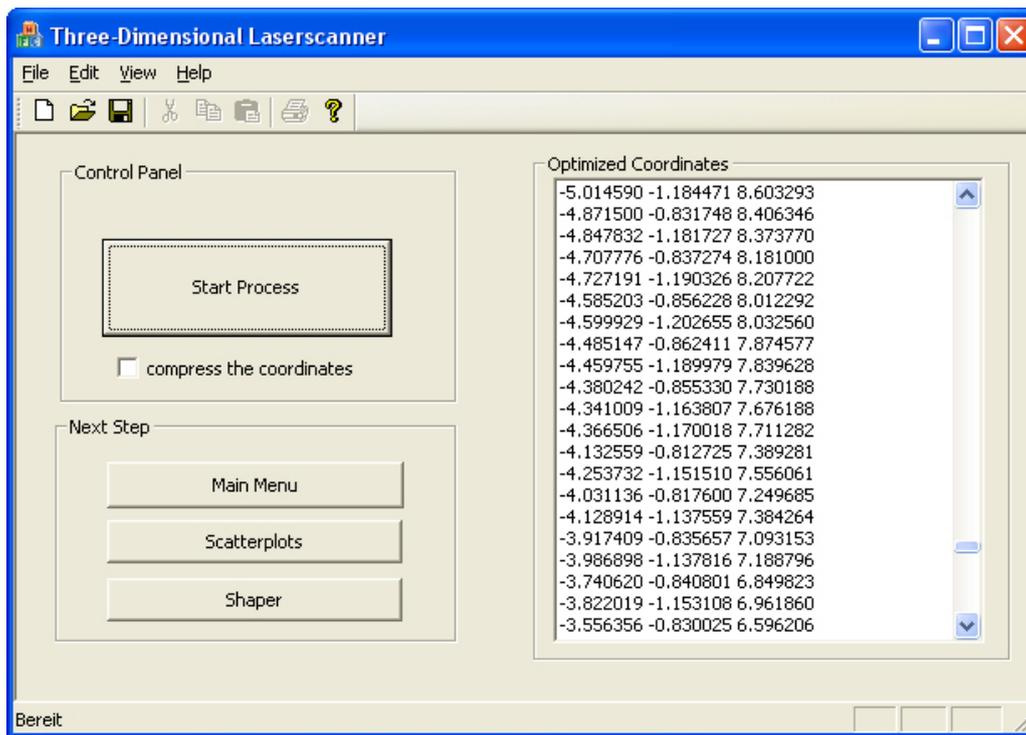
Nachdem die Vorbereitungsschritte der Objektvermessung durchgeführt wurden, kann das Fenster geschlossen werden. Im Anschluss erscheint ein Fenster (Abbildung 6.6) mit weiteren Einstellungsmöglichkeiten zur Objektvermessung. Nach Angabe der insgesamt aufzunehmenden Bilder innerhalb einer 360 Grad Drehtellerrotation wird der Scanvorgang mit dem mit „Scanning“ bezeichneten Button gestartet. Alle anderen Einstellungen sind optional.



**Abbildung 6.6.:** Allgemeine Einstellungen zur Objektvermessung.

Im Anschluss der Vermessung erscheint ein Popup, der die tatsächliche Vermessungsdauer und die Anzahl der erfassten Objektpunkte wiedergibt.

Die Datenmenge, die erfasst wurde, muss nachfolgend formatiert werden. Dies ist notwendig, damit die Datenmenge zur Weiterverwendung in integrierten- und externen Bildverarbeitungsprogrammen möglich ist. Dazu wird das Fenster (Abbildung 6.7) zur Optimierung aufgerufen. Ebenfalls wurde die Möglichkeit, die Datenmenge auf *mm*-Schritte zu simplifizieren, implementiert.



**Abbildung 6.7.:** Die Formatierung der Datenmenge muss nach der Objektvermessung für interne- und externe Bildverarbeitungsprogramme angepasst werden.

### Stufe 3: Darstellung

Zwei integrierte Programme namens „Scatter3D“ (Heitler, 2007) und „PointCloud“ (Sycode, 2009) lassen sich innerhalb des Programmes aufrufen. Scatter3D wird über den Button „Scatterplot“ gestartet und stellt veranschaulichte Punktmengen dar. PointCloud wird über den Button „Shaper“ ausgeführt und ermöglicht die Darstellung, Bearbeitung und Modellierung von Punktmengen.



# 7. Auswertung

Das siebte Kapitel beschreibt die Leistung der Objektvermessung bezüglich Präzision, Geschwindigkeit, Struktur- und Materialeinfluss und stellt erfolgreiche Scanergebnisse im letzten Unterkapitel vor.

## Inhalt

---

7.1. Genauigkeit der Kalibrierung . . . . .	79
7.2. Messgenauigkeit . . . . .	80
7.3. Präzision . . . . .	82
7.4. Geschwindigkeit . . . . .	84
7.5. Struktur- und Materialeinfluss . . . . .	86
7.6. Messergebnisse . . . . .	87
7.6.1. Problematisches Messobjekt . . . . .	87
7.6.2. Komplexe Form . . . . .	88
7.6.3. Geometrische Formen . . . . .	89
7.6.4. Weitere Messresultate . . . . .	91

---

## 7.1. Genauigkeit der Kalibrierung

Die maximal erzielbare Genauigkeit lässt sich näherungsweise als die Genauigkeit der Kalibrierung berechnen. Für jedes der  $n$  Paare von Referenzpunkten  $(\vec{c}_i, \vec{l}_i)$  wird hierzu der Punkt  $\vec{c}_i$  in der Bildebene auf einen Punkt  $\vec{l}'_i = \mathbf{B}\vec{c}_i$  in der Laserebene transformiert. Der Abstand des so berechneten Punktes  $\vec{l}'_i$  von dem tatsächlichen Punkt  $\vec{l}_i$  ist die durch die Approximation der Transformationsmatrix verursachte Abweichung. Die mittlere Abweichung ergibt sich daraus mit

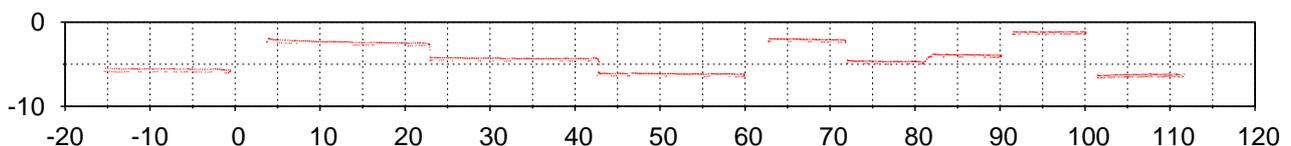
$$E_{avg} = \frac{1}{n} \sum_{i=0}^n |\vec{l}_i - \mathbf{B}\vec{c}_i| \quad (7.1)$$

In den durchgeführten Versuchen lagen die Werte der durchschnittlichen Genauigkeit im Bereich von  $0.09\text{mm}$  bis  $0.24\text{mm}$ . Diese Genauigkeit ist von zahlreichen Faktoren abhängig. Den vermutlich entscheidenden Einfluss hat die Ausrichtung der Kamera relativ zu der Laserebene und damit die Stärke der auftretenden perspektivischen Verzerrungen. Hierüber lässt sich aber anhand der wenigen durchgeführten Messungen keine genaue Aussage treffen. Einen weiteren Einfluss haben auch die intrinsischen Parameter der Kamera, die in dieser Arbeit nicht berücksichtigt wurden. Die intrinsischen Parameter der Kamera beschreiben die inneren Eigenschaften der Kamera und damit die Abbildung der Bildebene auf die Pixelebene. Werden diese Parameter berücksichtigt, kann die durch die Kamera verursachte Linsenverzeichnung im Bild ausgeglichen werden. Des Weiteren ist die Wahl der Kalibrierungsparameter wichtig. Insbesondere hat der Abstand der Edge-Samples (vgl. Abschnitt 5.1.2 auf S.44) und damit die Anzahl der für die Approximation der Kantengeraden verwendeten Punkte einen Einfluss auf die maximale Genauigkeit.

## 7.2. Messgenauigkeit

Die Messgenauigkeit des Systems ist nicht nur von der Genauigkeit der Kalibrierung sondern auch von einigen anderen Faktoren abhängig. Zu diesen Faktoren gehört insbesondere die Genauigkeit, mit der die Koordinaten der Mittelpunkte der Laserlinie bestimmt werden.

Um diese Genauigkeiten zu untersuchen, wurde ein Messobjekt hergestellt (siehe Abbildung 5.9 auf S. 52), dessen Abmessungen und Eigenschaften hinreichend genau bekannt sind. Das aus insgesamt 1356 Punkten bestehende Ergebnis der Vermessung dieses Objektes ist in Abbildung 7.1 dargestellt.



**Abbildung 7.1.:** Ergebnis der Vermessung des Testobjektes. Die Koordinaten sind in  $\text{mm}$  angegeben; Ursprung des Koordinatensystems ist der untere linke Eckpunkt des Kalibrierungsmusters (vgl. Abschnitt 5.1)

Um nun einige Kennzahlen für die Messgenauigkeit aus diesen Punkten zu berechnen, wurde zunächst die sogenannte Wiederholgenauigkeit der Messung der

$y$ -Koordinaten bestimmt. Hierzu wurde das Ergebnis in mehrere Bereiche eingeteilt, in denen die Höhe des Messobjektes jeweils konstant ist. Für jeden dieser Bereiche wurden nun die wichtigsten statistischen Kennzahlen für die  $y$ -Werte berechnet. Die Ergebnisse dieser Auswertung sind in Tabelle 7.1 zusammengefasst.

$x$	$n$	$y_m$	$y_{min}$	$y_{max}$	$\bar{y}_{arith}$	$\sigma_y$
$[-15, 0]$	54	0 mm	-6.0326 mm	-5.5028 mm	-5.7090 mm	0.1736 mm
$[5, 20]$	138	4 mm	-2.7676 mm	-2.1286 mm	-2.3871 mm	0.1356 mm
$[25, 40]$	149	2 mm	-4.6542 mm	-4.2318 mm	-4.3375 mm	0.0956 mm
$[45, 60]$	167	0 mm	-6.4873 mm	-6.0600 mm	-6.1749 mm	0.0956 mm
$[62, 70]$	91	4.5 mm	-2.3689 mm	-1.9398 mm	-2.0933 mm	0.1136 mm
$[75, 80]$	58	2 mm	-5.0034 mm	-4.6600 mm	-4.7387 mm	0.1103 mm
$[83, 90]$	90	3 mm	-4.1963 mm	-3.8547 mm	-3.9499 mm	0.1102 mm
$[92, 100]$	110	5 mm	-1.4695 mm	-1.1196 mm	-1.2459 mm	0.1075 mm
$[103, 110]$	101	0 mm	-6.5662 mm	-6.1599 mm	-6.3276 mm	0.1259 mm

**Tabelle 7.1.:** Messgenauigkeit in  $y$ -Richtung:  $x$  gibt den untersuchten Bereich des Ergebnisses auf der  $x$ -Achse an;  $n$  ist die in diesem Bereich befindliche Anzahl an Messpunkten;  $y_m$  gibt den Erwartungswert, bezogen auf die niedrigste Oberfläche des Objektes an;  $y_{min}$  und  $y_{max}$  geben den maximalen und minimalen  $y$ -Wert an;  $\bar{y}_{arith}$  und  $\sigma_y$  geben den arithmetischen Mittelwert und die Standardabweichung an.

An diesen Daten lässt sich gut erkennen, dass mit dem realisierten Messsystem sehr genaue Messungen möglich sind. Vergleicht man zum Beispiel die recht ungenau gemessenen Erwartungswerte mit den dazugehörigen Mittelwerten, stellt man fest, dass die maximale Abweichung kaum größer als ein halber Millimeter ist. Dieser Wert entspricht in etwa der Genauigkeit, mit der das Messobjekt von Hand vermessen werden konnte.

Die von links nach rechts kleiner werdenden Mittelwerte in den Bereichen, in denen der Erwartungswert gleich Null ist, lassen darauf schließen, dass die untere Kante des Kalibrierungsmusters bei der Kalibrierung nicht exakt parallel zu der Oberfläche des Drehtellers war. Anhand der Werte der Standardabweichungen und der maximalen Abweichung vom Mittelwert ( $\sim 0.3$  mm) lässt sich gut erkennen, dass die Objektvermessung unter optimalen Bedingungen (genaue Ausrichtung des Kalibrierungsmusters und des Lasers sowie optimale Einstellungen von Laser und Kamera) in der Lage ist, eine sehr hohe Genauigkeit zu erreichen.

### 7.3. Präzision

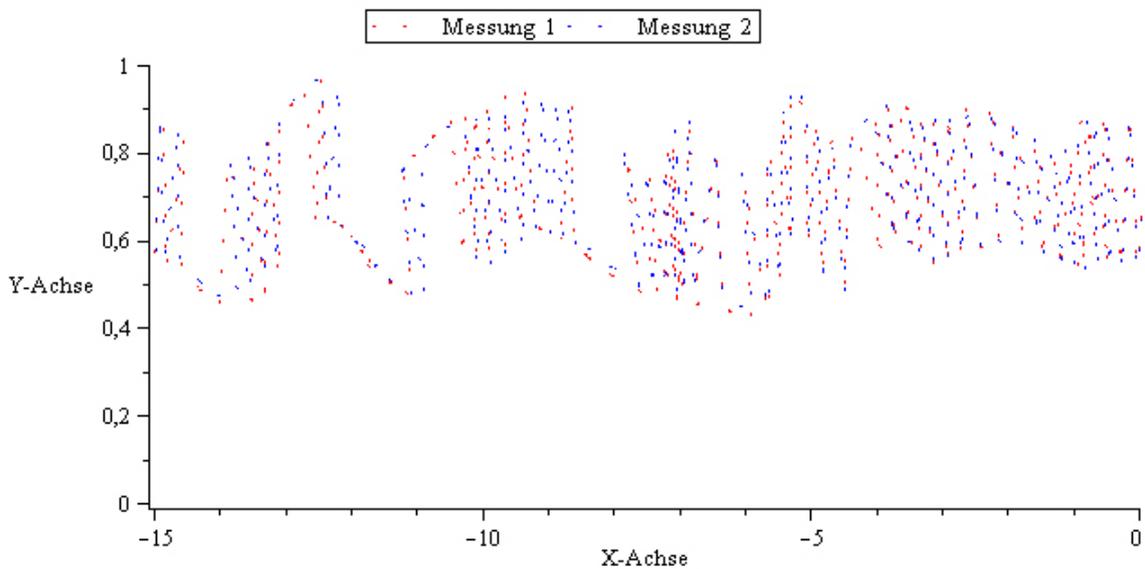
Im vorherigen Kapitel 7.2 wurde die Messgenauigkeit der Objektvermessung ermittelt. Dabei wurde ein Messobjekt per Hand- und mit dem Laserscanner vermessen, um nachfolgend die resultierenden Ergebnisse zu vergleichen.

Zur Überprüfung der Messpräzision in Bezug auf einen vollständig zu vermessenden Gegenstand, wurde in dieser Arbeit eine Schachtel vermessen.

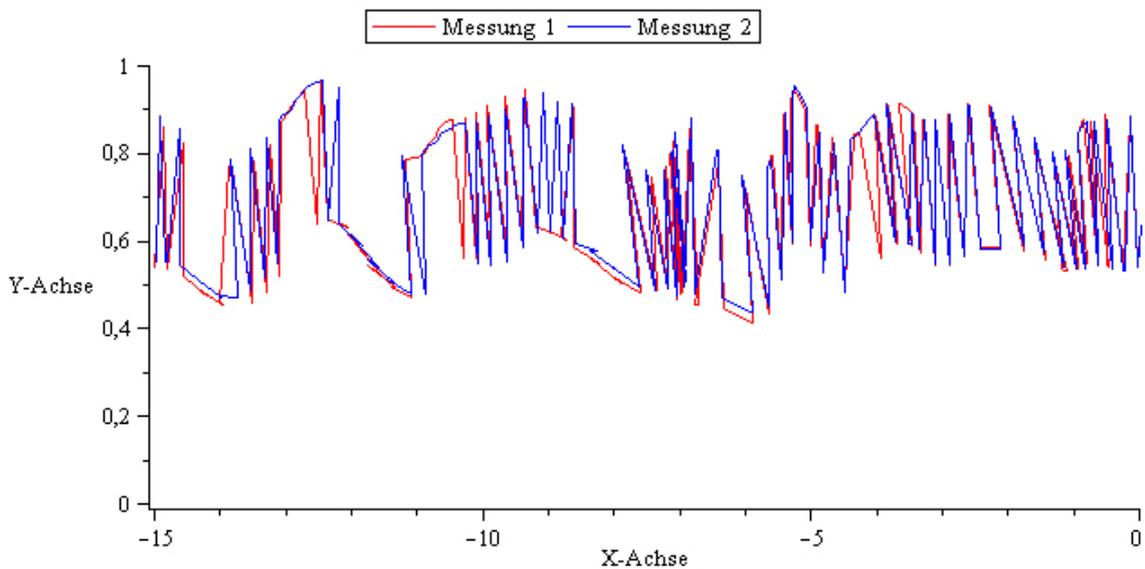
Die zufällige Abweichung von Werten  $x_i \in \Delta x$  um einen Mittelwert  $\bar{x}_{arith}$ , wird durch die Präzision beschrieben. Angegeben wird sie als „Standartabweichung  $\sigma$ “:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x}_{arith})^2}{n - 1}} \quad \begin{array}{l} x_i = \text{Messwert} \\ \bar{x}_{arith} = \text{Arithmetischer Mittelwert} \\ n = \text{Anzahl Messwerte} \end{array} \quad (7.2)$$

Um die Präzision des Laserscanners zu ermitteln, wurde dieser Gegenstand jeweils unter denselben Umständen mehrfach vermessen. Die somit ermittelten Resultate spiegeln die Wiederholpräzision wieder. Die Nachfolgende Abbildung 7.2 und Tabelle 7.2 stellen Kennzahlen der Wiederholpräzision im Bezug der  $y$ -Richtung in einem festgelegtem Bereich  $\Delta x$  dar.



(a) Punktdarstellung



(b) Liniendarstellung

**Abbildung 7.2.:** Vergleich zweier Messungen desselben Objektes.

Zwei Vermessungen vom gleichen Messobjekt wurden nacheinander unter denselben Umständen durchgeführt. Die beiden Grafiken zeigen die Messwerte in unterschiedlichen Darstellungsweisen in einem Bereich  $[-15,0]$  und damit einen Ausschnitt der Punktmenge die bei der Vermessung erfasst wurde. Es ist gut zu erkennen, dass die zweite- der ersten Messung näherungsweise entspricht. Dies lässt auf eine hohe Wiederholpräzision schließen. Alle Angaben sind in *mm* dargestellt.

Messung	Aufnahmewinkel $\theta$	Bereich $\Delta x$	Anzahl Punkte $n$	$y_{min}$	$y_{max}$	$\bar{y}_{arith}$	$\sigma_y$
1	90°	[-20,20]	347	0,132	1,054	0,663	0,213
1	180°	[-15,10]	226	0,123	0,971	0,618	0,207
1	270°	[-20,20]	354	0,248	0,987	0,663	0,187
1	360°	[-15,10]	196	0,062	0,927	0,599	0,207
2	90°	[-20,20]	350	0,132	1,071	0,670	0,215
2	180°	[-15,10]	218	0,123	0,971	0,631	0,211
2	270°	[-20,20]	353	0,248	0,987	0,675	0,186
2	360°	[-15,10]	197	0,062	0,930	0,600	0,206

**Tabelle 7.2.:** Kennzahlen der Wiederholpräzision.

$\Delta x$  ist der analysierte Bereich in dem sich die Punkte befinden.  $y_{min}$  und  $y_{max}$  kennzeichnen den jeweils minimalen und maximalen  $y$ -Wert.  $\bar{y}_{arith}$  und  $\sigma_y$  stellen den arithmetischen Mittelwert und die Standardabweichung dar. Alle Maßeinheiten, ausgenommen der Messungsnummer, dem Aufnahmewinkel und der Anzahl erfasster Punkte sind in *mm* angegeben.

In wesentlicher Abhängigkeit zur Kalibrierung wurde eine Wiederholpräzision von circa einem zwanzigstel Millimeter festgestellt. Das Resultat dieser Kennzahlen stimmt im Wesentlichen mit der ermittelten Messgenauigkeit von Kapitel 7.2 überein.

## 7.4. Geschwindigkeit

Die Geschwindigkeit ist ein weiteres wichtiges Kriterium für die praktische Anwendbarkeit. Im Laufe der Entwicklung des Laserscanners wurde ein besonders großes Augenmerk auf die Vermessungsdauer gelegt. Die Dauer einer Vermessung hängt von vielen Faktoren ab:

- Zum einen dauert der Kalibrierungsvorgang in etwa zwei Minuten. Er wird nur einmalig, wenn die Kamera oder der Laser neu ausgerichtet wurde, ausgeführt. Größere Zeitschwankungen treten im Kalibrierungsvorgang nicht auf.
- Zum anderen sind Vermessungsphasen vorhanden, in denen Bilder (Lichtschritte) vom Objekt aufgenommen, die Akquisition der Laserlinie- und die Re-

konstruktion der 3D-Koordinaten, durchzuführen sind. Diese Phasen weisen eine hohe Zeitintensität auf.

Um die Geschwindigkeit zu verbessern wurde der näherungsweise bestimmte Multiplikationsfaktor  $F$ , der die benötigte Vermessungsdauer eines Bildes repräsentiert, stark gesenkt. Die nachfolgend dargestellte Tabelle 7.3 zeigt den Entwicklungsverlauf des Multiplikationsfaktors  $F$ . Alle dargestellten Werte sind Durchschnittswerte, die aus mehreren Scanvorgängen, mit unterschiedlichen Messobjekten ermittelt wurden.

Entwicklungsphase	Multiplikationsfaktor F	Vermessungsdauer	Erfasste Oberflächenpunkte	Optimierung
1	>0,50			
2	0,50	50 <i>min</i>	20.000	Codeoptimierung
3	0,45	45 <i>min</i>	60.000	Vom Ursprung aufgespannte ROI.
4	0,27	27 <i>min</i>	140.000	Komplette Raum-Abdunkelung mittels Verkleidung.
5	0,22	22 <i>min</i>	145.000	Diverse Optimierungen
6	0,10	10 <i>min</i>	145.000	Verbesserte ROI (Region of Interest)

**Tabelle 7.3.:** Entwicklungsphasen des Multiplikationsfaktors F. Wertangaben sind in Abhängigkeit zu 100 Bildern in 360° angegeben.

Es ist deutlich zu erkennen, dass sich die Geschwindigkeit im Laufe der Entwicklung um mehr als das Fünffache verbessert hat. Die Formel 7.3 beschreibt die ungefähr benötigte Zeit für eine komplette Objektvermessung.

$$\text{Vermessungsdauer} = (\text{Kalibrierzeit}) + \text{„Anzahl Bilder in } 360^\circ \text{“} * \text{„Multiplikationsfaktor } F \text{“} \quad (7.3)$$

## 7.5. Struktur- und Materialeinfluss

Beeinflussbare Faktoren eines zu vermessenden Objektes sind vor der Vermessung besonders zu beachten.

Das Material der Oberfläche sollte die Laserlinie so aufnehmen, dass eine saubere und deutliche Laserlinie auf der Oberflächenstruktur des Objektes abgebildet wird. Matte Oberflächen und Material wie Holz, Ton oder Lehm sind dafür besonders gut geeignet. Glas, beispielsweise im Fenster oder Spiegel, ist dagegen mit keinem erfolgreichen Ergebnis zu erfassen.

Ebenfalls können durch eventuell vorhandene Vertiefungen, Nischen oder Einschnitte in der Oberflächenstruktur Vermessungslücken entstehen. Stofftiere sind von der Materialbeschaffenheit oft sehr plüschig und weisen eine haarige Oberflächenstruktur auf. Das kann zu vielen Lücken im Messresultat führen.

## 7.6. Messergebnisse

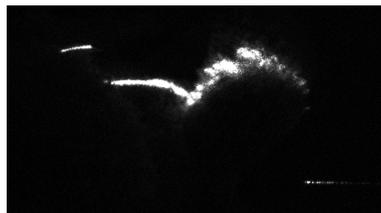
In diesem Unterkapitel werden erfolgreiche Messergebnisse und durch Probleme in der Struktur- und Materialbeschaffenheit entstandene erfolglose Vermessungen, vorgestellt. Alle Angaben bezüglich der Punktmengen sind gerundet angegeben.

### 7.6.1. Problematisches Messobjekt

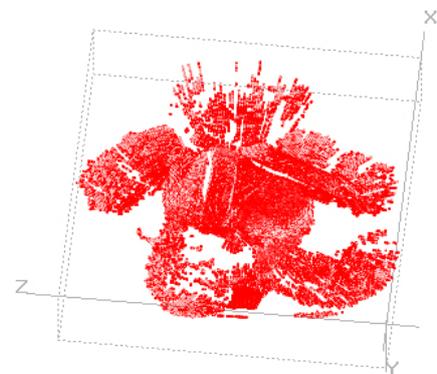
Ein zu vermessendes Objekt, welches erhebliche Probleme in der Struktur- und Materialbeschaffenheit aufweist, ist der in Abbildung 7.3 dargestellte Stoffteddy. Die Grafiken verdeutlichen die Schwierigkeit der Erfassung einer haarigen Oberflächenstruktur. Viele große Lücken sind dadurch im Messresultat entstanden. In Nachbearbeitungsphasen konnten diese trotz 400.000 erfasster Oberflächenpunkte, nicht originalgetreu geschlossen werden.



(a) Originalbild



(b) Lichtschnitt

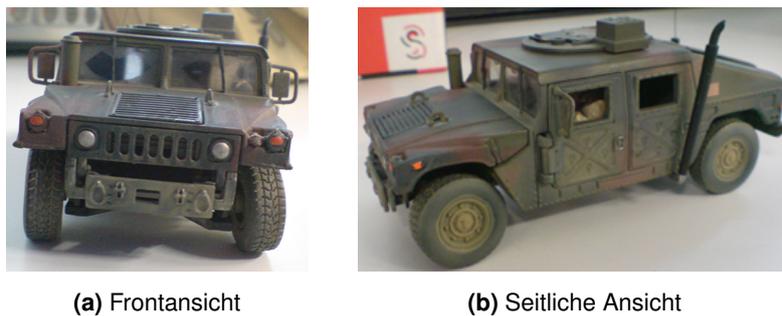


(c) Punktwolke

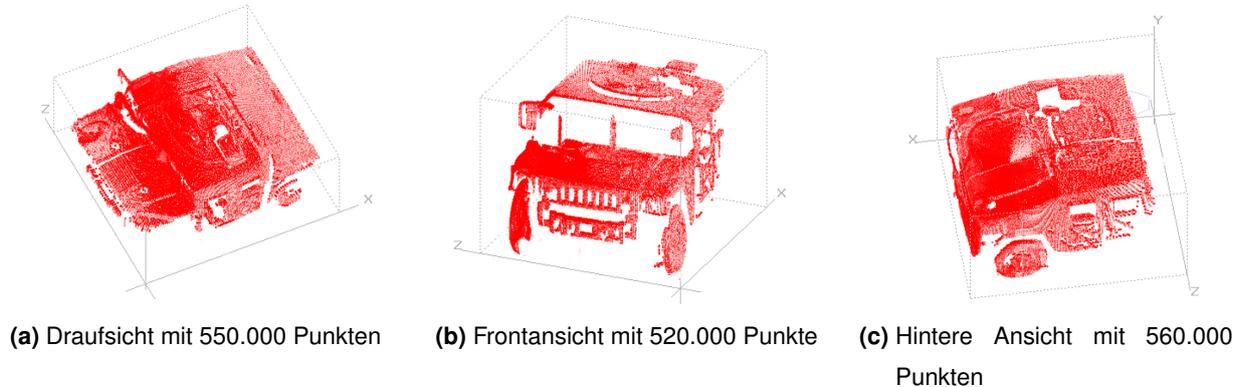
**Abbildung 7.3.:** Stoffteddy mit einer erheblichen Anzahl von Lücken im Messresultat.

## 7.6.2. Komplexe Form

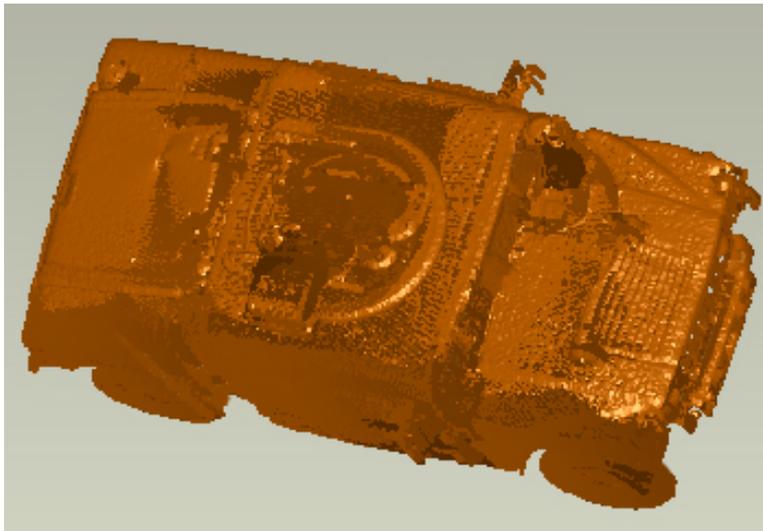
Als ein sehr komplexes Messobjekt wurde ein Spielzeugauto vermessen. Das Ergebnis brachte jedoch aufgrund der guten Oberflächenbeschaffenheit, ein sehr detailgetreues Messresultat mit sich. In Abbildung 7.5 sind die Ergebnisse von drei Objektvermessungen in Form von Punktwolken zu betrachten. Ausgehend davon wurden diese drei Punktwolken in einem Registrierungsprozess vereinigt. Das modellierte Endresultat zeigt die Abbildung 7.6.



**Abbildung 7.4.:** Originalbilder eines Spielzeugautos.



**Abbildung 7.5.:** Punktwolken des Spielzeugautos.



(a) Ergebnis von drei registrierten Flächenmodellen

**Abbildung 7.6.:** Modellerte Darstellung des Spielzeugautos mit insgesamt 1.600.000 Punkten.

### 7.6.3. Geometrische Formen

Geometrische Modelle lassen sich durch ihre geometrischen Eigenschaften sehr gut vermessen und modellieren. Ausgehend von Messergebnissen in Form von Punktwolken veranschaulichen die nachfolgenden Bilder die vollständige Rekonstruktion übergehend vom Polygon- zum endgültigen modellierten Modell. Ein Puzzle aus Holz und eine Schachtel aus Papier wurden dafür rekonstruiert.

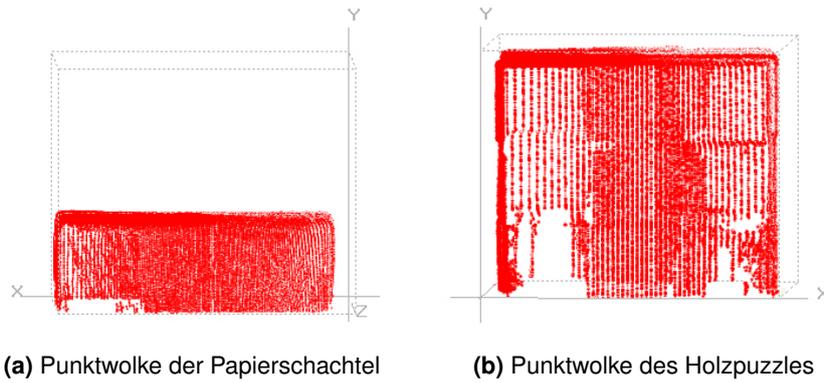


(a) Papierschachtel

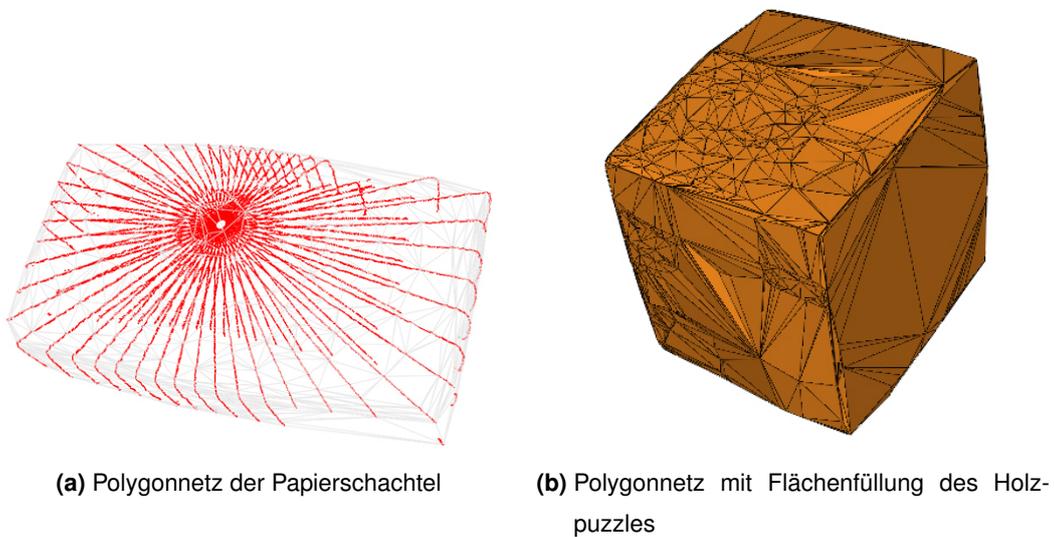


(b) Holzpuzzle

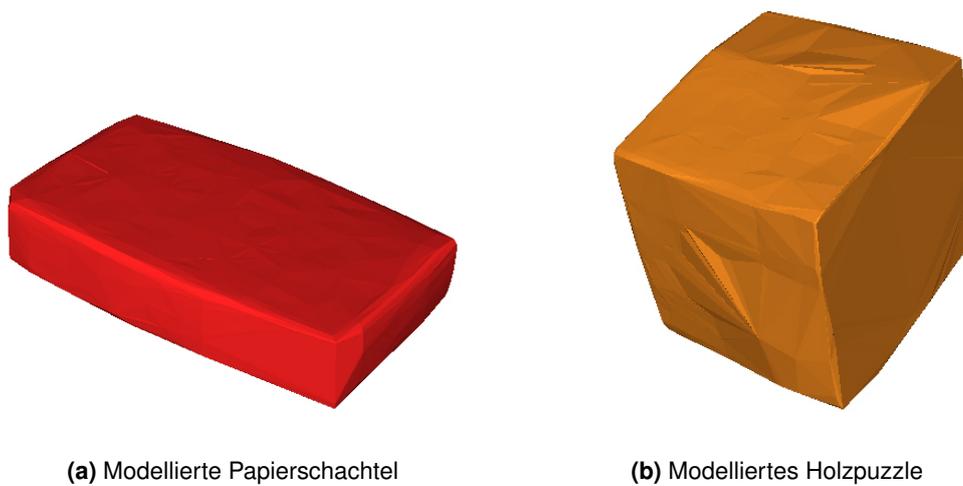
**Abbildung 7.7.:** Originalbilder der zu vermessenden Objekte.



**Abbildung 7.8.:** Resultate zweier Objektvermessungen.



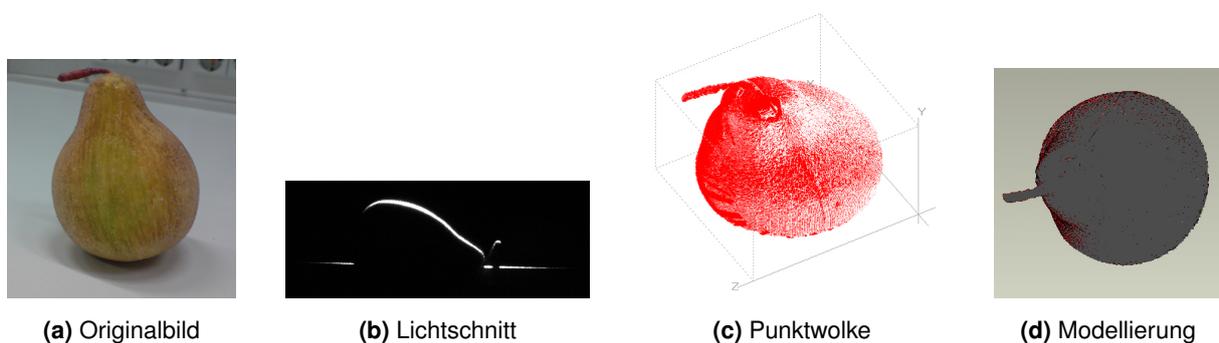
**Abbildung 7.9.:** Vernetzungsergebnisse der Punktmengen.



**Abbildung 7.10.:** Flächenbasierte Darstellung der Messobjekte.

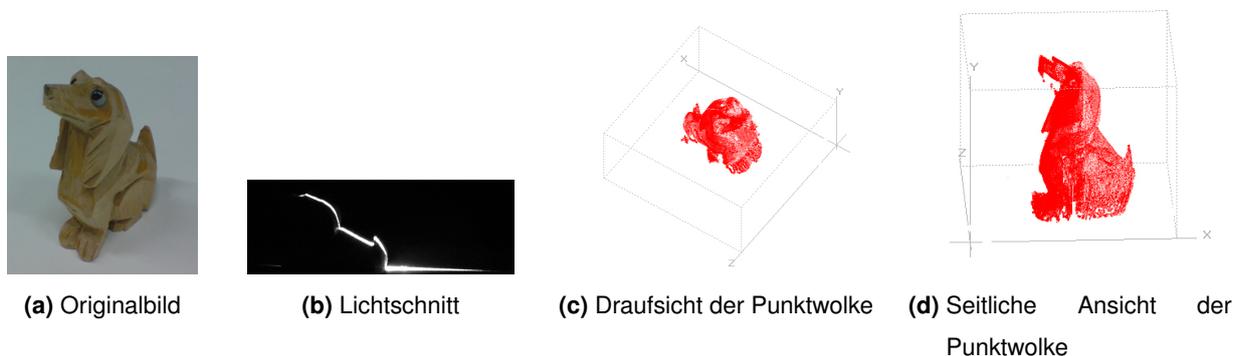
### 7.6.4. Weitere Messresultate

Eine aus Plastik produzierte Dekobirne wurde in einem Scanvorgang vermessen. Insgesamt erfasste der Laserscanner 470.000 Punkte. Nachbearbeitungsphasen, wie das Korrigieren von Messfehlern und das Füllen von Löchern erbrachten eine korrigierte Punktmenge mit 500.000 Punkten. Ausgehend von einer optimierten Punktmenge, wurde so ein detailgetreues flächenbasiertes Modell aus 1.000.000 Punkten rekonstruiert.

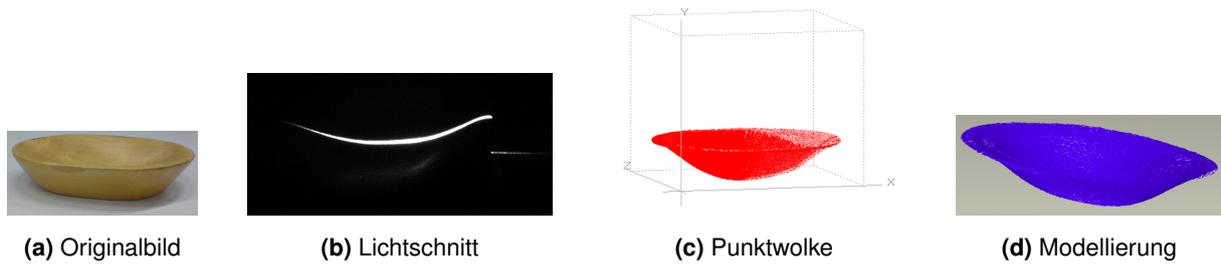


**Abbildung 7.11.:** Dekobirne aus Plastik.

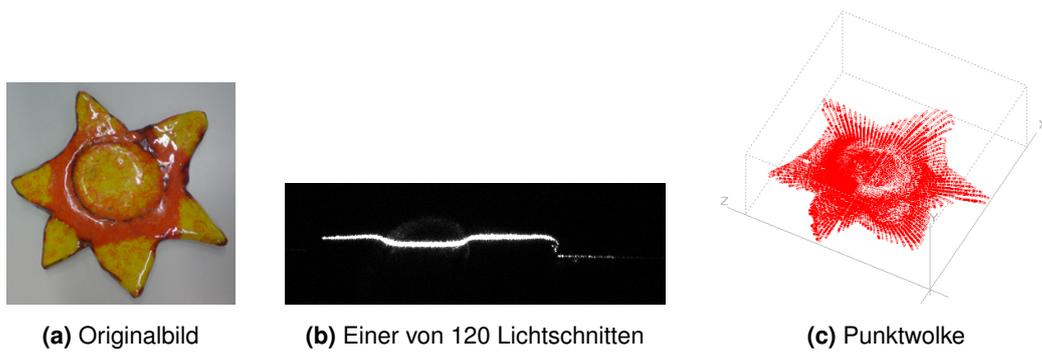
Weitere ausgewählte Ergebnisse von Scanvorgängen sind den nachfolgenden Abbildungen 7.12 bis 7.17 zu entnehmen.



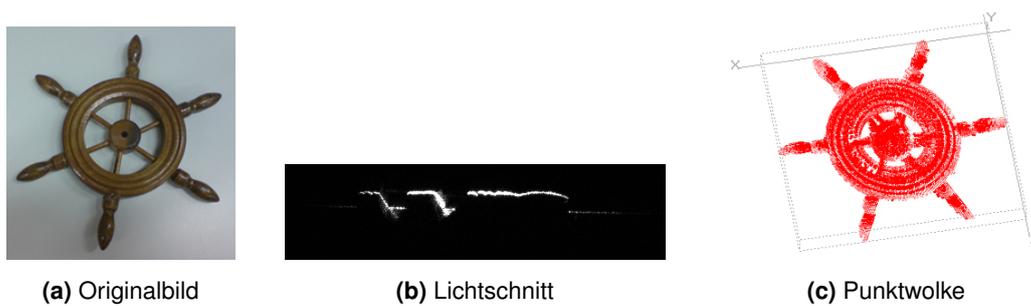
**Abbildung 7.12.:** Messergebnis mit rund 360.000 erfassten Objektpunkten einer Hundefigur aus Holz.



**Abbildung 7.13.:** Schale aus Holz.



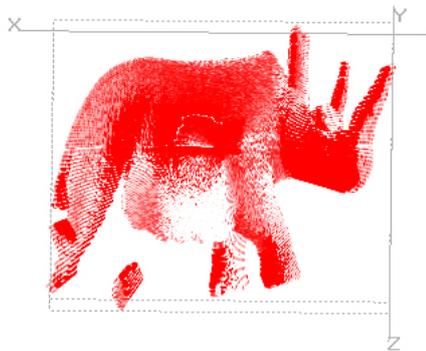
**Abbildung 7.14.:** Stern aus Ton.



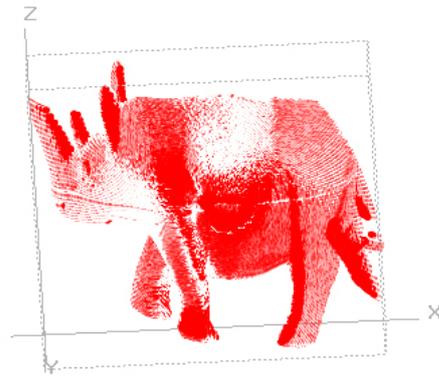
**Abbildung 7.15.:** Ruder aus Holz.



(a) Originalbild



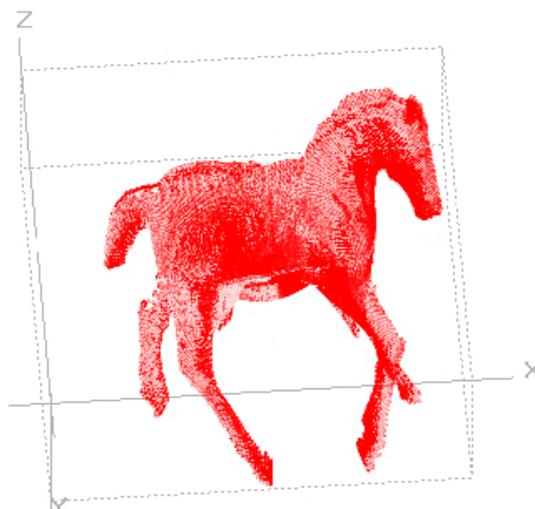
(b) Punktwolke - Scan 1



(c) Punktwolke - Scan 2

**Abbildung 7.16.:** Nashorn aus Holz.

(a) Originalbild



(b) Punktwolke

**Abbildung 7.17.:** Holzfigur eines Pferdes.



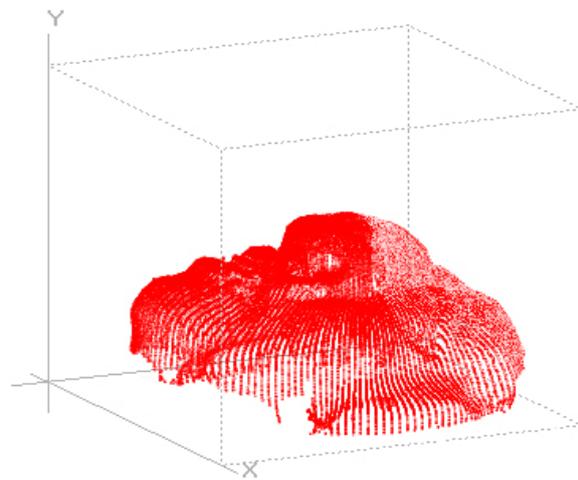
(a) Draufsicht des Originalbildes



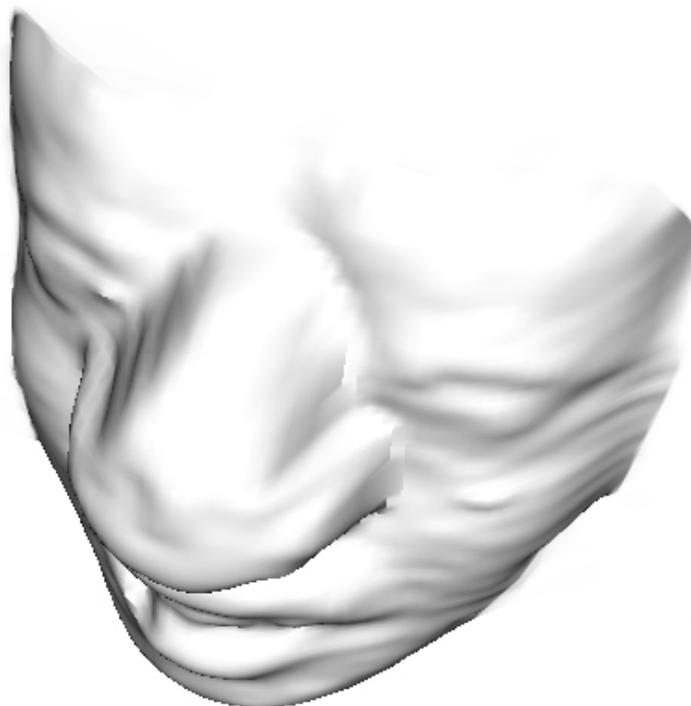
(b) Seitliche Ansicht des Originalbildes



(c) Lichtschnitt



(d) Punktwolke



(e) Modellierung

**Abbildung 7.18.:** Gesicht aus Ton.

# 8. Zusammenfassung und Ausblick

Zunächst werden die wesentlichen Ergebnisse der vorliegenden Arbeit zusammengefasst. In einem Ausblick folgen mögliche Weiterentwicklungsmöglichkeiten.

## Inhalt

---

8.1. Stand der Arbeit . . . . .	95
8.2. Weiterentwicklungen . . . . .	96

---

### 8.1. Stand der Arbeit

Das Ziel dieser Arbeit bestand in der Realisierung eines 3D-Vermessungssystems auf Basis der Lichtschnitttechnik.

In dieser Arbeit wurden die mathematischen und bildverarbeitungstechnischen Grundlagen der Lichtschnitttechnik erläutert. Die Ansätze für eine Kalibrierung der Systemkomponenten und die 3D-Vermessung einer Objektoberfläche beschrieben und implementiert. Des Weiteren wurden Untersuchungen über die mit diesen Ansätzen realisierbaren Genauigkeiten durchgeführt.

Die Realisierungsgrundlage bestand aus der Kalibrierung der Systemkomponenten zueinander, die 2D-Objektvermessung und der Ansteuerung eines Drehtellers mit Hilfe einer Positionierungseinheit.

Der Laserscanner, der im Wesentlichen aus einer Kamera, einem Linienlaser, einem ansteuerbaren Drehteller, einer Positionierungseinheit und einem verkleideten Gerüst besteht, erzielt dank seiner zwanzigstel Millimeter Präzision sehr gute Messergebnisse. Aufgrund der verbesserten Messgeschwindigkeit, dem reduzierten Licht- und Materialeinfluss, können detailgetreue Rekonstruktionen von Gegenständen, schnell und ohne Störungen, erfasst werden.

Eine grafische Oberfläche, die zur Steuerung des Laserscanners und zur Nachbearbeitung von Punktmengen entwickelt wurde, bietet eine übersichtliche und einfache Handhabung über die Vermessung von Gegenständen und die anschließenden Weiterverarbeitungsmöglichkeiten.

Zusammengefasst wurde ein Laserscanner entwickelt, mit dem sich Gegenstände detailgetreu dreidimensional rekonstruieren lassen.

## 8.2. Weiterentwicklungen

Basierend auf den in dieser Arbeit gewonnen Erkenntnissen, können weitere Projekte entstehen.

Am Anfang dieser Arbeit wurden im Kapitel 2.2.2 die drei typischen Triangulationsverfahren, die Punktriangulation, die Lichtschnitttechnik und der codierte Lichtansatz beschrieben. Eine Erweiterung des Lichtschnittverfahrens, sodass mehrere Lichtstreifen gleichzeitig auf das zu vermessende Objekt projiziert werden, ermöglicht die Erfassung des gesamten Objektes bzw. der maximal möglichen Größe des Messbereiches. Dies führt zu einer Reduzierung der aufzunehmenden Bilder.

Für den Vergleich der Triangulationsverfahren bezüglich der Geschwindigkeit und Genauigkeit, würde eine Untersuchung des codierten Lichtansatzes Aufschluss geben. Beim codierten Lichtansatz wird ein binär-, farbcodiertes oder Moiré-Muster mit einem Lichtmusterprojektor flächenhaft auf das Objekt projiziert.

Des Weiteren würde ein besseres Kalibrierungsverfahren zu einer deutlichen Verbesserung der Genauigkeit führen. Der Einfluss der genauen Ausrichtung von Laser und Kamera könnte untersucht werden. Eventuell lassen sich mit anderen Kalibrierungsverfahren die Referenzpunkte noch genauer bestimmen. Des Weiteren lässt sich die Genauigkeit der Kalibrierung durch das Ausgleichen der Linsenverzeichnungen verbessern.

Das in dieser Arbeit entwickelte Kalibrierungsverfahren ist in seiner Genauigkeit eingeschränkt, wenn es um zehntel- oder gar hundertstel-Millimeter geht, da in diesen Genauigkeitsbereichen schon eine winzige Abweichung der Kalibriervorrichtung von ihrer Position oder ein kleiner Fehler bei der Ausrichtung des Lasers ausreicht, um die

---

Positionsgenauigkeit der Referenzpunkte schlechter als die gewünschte Messgenauigkeit werden zu lassen. Hier stellt sich also die Frage, ob es Kalibrierungsverfahren gibt, bei denen diese Probleme nicht oder nur in abgeschwächter Form auftreten.

Weitestgehend finden Schwarz/Weiß-Kameras in Laserscannern ihre Verwendung. Eine mögliche Untersuchung würde herausfinden, ob mit geringem Geschwindigkeitsverlust Farbinformationen bezüglich des zu vermessenden Objektes gewonnen werden können. Eine Farbkamera könnte in Kombination mit einer Belichtungseinheit agieren, um Farbwerte der Objekt Oberfläche zu erhalten. Diese Farbinformationen würden eine noch detailgetreuere Darstellung von Punktmengen und Flächenmodellen realisieren.



# A. Verwendete Hilfsmittel

Mit Hilfe von Bibliotheken, Softwareprodukten und weiteren weitestgehend öffentlich freien Hilfsmitteln, konnte die in dieser Arbeit beschriebene Laserscannersoftware entwickelt werden. Im Folgenden wird auf die verwendeten Hilfsmittel eingegangen.

## Inhalt

---

A.1. Bibliotheken . . . . .	99
A.2. Bild- und Grafikprogramme . . . . .	99
A.3. Softwaresysteme zur Darstellung und Weiterverarbeitung von Punktmengen .	100
A.4. Weitere Hilfsmittel . . . . .	100

---

## A.1. Bibliotheken

Die in der Bildverarbeitung verbreitete objektorientierte „LTI-Lib“ (Alvarado u. a., 2005) hat in dieser Arbeit häufig Verwendung gefunden. Für mathematische Algorithmen oder Bildverarbeitungsverfahren kam die umfangreiche LTI-Bibliothek, in der neuesten Version 1.9 zum Einsatz.

„Boost“ (Boost, 2009) ist eine C++-Bibliothek welche viele Hilfsfunktionen zur Programmiererleichterung zur Verfügung stellt. Die Version 1.3 fand zur Textformatierung Gebrauch.

## A.2. Bild- und Grafikprogramme

Der im Kapitel 4 grafisch dargestellte Bauplan des Laserscannergerüsts wurde mit „CADEMIA 1.4“ entworfen. CADEMIA ist ein Softwaresystem zur Lösung von Aufga-

ben im Bauingenieurwesen. Als öffentlich freie Software zu finden unter (Beucke u. a., 2009).

„Adobe Photoshop CS3“ wurde für die Bildverarbeitung eingesetzt. Kommerziell erwerbbar auf (Adobe-Systems, 2009).

Das „Microsoft Office Visio 2007“ (Microsoft-Corporation, 2008) -Produkt ist eine weit verbreitete Visualisierungs-Software und fand für Ablaufpläne Verwendung.

### **A.3. Softwaresysteme zur Darstellung und Weiterverarbeitung von Punktmengen**

„Geomagic Studio 9“ (Geomagic, 2009) kam weitestgehend für die Lochbeseitigung, Registrierung und Modellierung von Punktwolken zum Einsatz.

Mit „Scatter3D“ (Heitler, 2007) konnten zahlreiche Punktwolken dargestellt werden.

„PointCloud“ (Sycode, 2009) diente größtenteils als zweites Modellierungswerkzeug. Ausgehend von geladenen Punktmengen wurden mit diesem Programm, Polygonnetze und Flächenmodelle erstellt.

Des Weiteren wurde „AutoCAD 2008“ (Autodesk, 2009) zur Punktmengenverarbeitung genutzt.

### **A.4. Weitere Hilfsmittel**

„Microsoft Visual Studio 2008“ (Microsoft-Corporation, 2007) ist ein Entwicklungswerkzeug, mit dem entwickelt und verifiziert wurde.

Die Bildverarbeitungssoftware „IC Imaging Control 3“ (The-Imaging-Source, 2008a) kam zur Belichtungseinstellung und als Grundlage zur Darstellung von Livescreens zum Einsatz.

Des Weiteren fand die zur universellen Positioniersteuerung „PS90“ (OWIS, 2009) und „Maple 12“ (Maplesoft, 2009) zur nichtlinearen Ausgleichsrechnung Verwendung.

# B. Codebeispiele

## Inhalt

---

B.1. Interfaces . . . . .	101
B.2. Aufruf der Kalibrierung und Objektvermessung . . . . .	105
B.3. Bildaufnahme . . . . .	107

---

## B.1. Interfaces

In diesem Abschnitt sind die Programmierschnittstellen der Klassen `lts::Calibration` und `lts::Lts` in reduzierter Form dargestellt. Die vollständigen Schnittstellendefinitionen und eine deutlich ausführlichere Dokumentation findet sich in den entsprechenden Header-Dateien.

```
1 namespace lts {
2     class Calibration {
3         public:
4             // parameters-class, see Listing B.2
5             class parameters { /* ... */ };
6
7             // constructors
8             Calibration ();
9             Calibration( const Calibration::parameters &p );
10
11            // getters and setters
12            void setParameters( const lts::Calibration::parameters &p );
13            void setRefPoints( const std::vector<lts::fpoint> &refPoints );
14            lts::dmatrix &getTransformationMatrix ();
15
```

```

16     // apply-methods
17     void apply( const lti::image &srcImage );
18     void apply( const lti::image &srcImage,
19                 const std::vector<lti::fpoint> &refPoints );
20 };
21 };

```

**Listing B.1:** public-interface der Klasse `lts::Calibration`

```

1 namespace lts {
2     class Calibration {
3     public:
4         class parameters {
5             public:
6                 // default- and copy-constructor
7                 parameters();
8                 parameters(const parameters &other);
9                 parameters& copy(const parameters& other);
10
11                 // operator for output to ostream
12                 friend std::ostream& operator <<( std::ostream &os,
13                                                     const Calibration::parameters &p );
14
15                 // parameters (public attributes)
16                 // configuration for the hough line-transform
17                 int hltBaseValue;
18                 int hltAccuracy;
19                 float hltLmdStdDevFactor;
20                 float hltLmdHysteresis;
21
22                 // distance between two edge-samples
23                 float edgeSampleDistance;
24
25                 // width of an edge-sample
26                 float edgeSampleWidth;
27
28                 // minimal difference between minimum and maximum of the
29                 // laPlacian within the edge-sample
30                 float minLPValueDiff;
31
32                 // maximum distance between indices of minimum and maximum of the laPlacian

```

```

33         int edgeMinMaxIdxDistance;
34
35         // definition of the region of interest
36         bool roiDefined;
37         lti::polygonPoints roi;
38     };
39     // ...
40 };
41 };

```

**Listing B.2:** Die Parameterklasse `lts::Calibration::parameters`

```

1 namespace lts {
2     class Lts {
3     public:
4         // parameters-class, see Listing B.4
5         class parameters { /* ... */ };
6
7         // constructors
8         Lts();
9         Lts( const Lts::parameters &p );
10        Lts( const Lts::parameters &p, const lti::dmatrix &trMatrix );
11
12        // accessors
13        void setParameters( const Lts::parameters &params );
14        void setTransformationMatrix( const lti::dmatrix &trMatrix );
15        void getPointList( std::vector<lti::fpoint> &res );
16
17        // apply-methods
18        void apply( lti::image &srcImage );
19        void apply( lti::image &srcImage, const lti::dmatrix &trMatrix );
20    };
21 };

```

**Listing B.3:** public-interface der Klasse `lts::Lts`

```

1 namespace lts {
2     class Lts {
3     public:
4         class parameters {
5         public:
6             // default- and copy-constructor

```

```
7     parameters ();
8     parameters(const parameters &other);
9     parameters& copy(const parameters& other);
10
11     // operator for output to ostreams
12     friend std::ostream& operator <<( std::ostream &os,
13         const Calibration::parameters &p );
14
15     // parameters (public attributes)
16     // cannyEdges parameters
17     float ceTresholdMin;
18     float ceTresholdMax;
19     float ceKernelSize;
20     float ceVariance;
21
22     // reflection-filter settings
23     bool rfEnabled;
24     float rfSize;
25     float rfMaxLineWidth;
26     float rfTreshold;
27     int rfOuterScore;
28     int rflInnerScore;
29 };
30 // ...
31 };
32 };
```

**Listing B.4:** Die Parameterklasse `lts::Lts::parameters`

## B.2. Aufruf der Kalibrierung und Objektvermessung

Im Folgenden werden zwei kurze Code-Beispiele zur Verwendung der Klassen `lts::Calibration` und `lts::Lts` gegeben.

```
1 lts::Calibration::parameters clbParams;
2
3 // setup calibration parameters...
4
5 // load the source image
6 lti::loadImage loader;
7 lti::image srcImage;
8 bool isLoading = loader.load( src_fname, srcImage );
9 if (!isLoading) { /* handle error */ }
10
11 std::vector<lti::fpoint> refPoints;
12 // fill in reference points...
13
14 lts::Calibration clb( clbParams );
15
16 // run the calibration
17 clb.apply( srcImage, refPoints );
18
19 // retrieve the transformation-matrix
20 lti::dmatrix trMatrix = clb.getTransformationMatrix();
```

**Listing B.5:** Beispiel für einen Aufruf der Kalibrierung

```
1 lts::Lts::parameters ltsParams;
2
3 // setup parameters ...
4
5 // load the source image
6 lti::loadImage loader;
7 lti::image srcImage;
8 bool isLoading = loader.load( src_fname, srcImage );
9 if (!isLoading) { /* handle error */ }
10
11 lti::dmatrix trMatrix;
12 // load the transformation-matrix...
```

```
13
14 lts :: Lts l( ltsParams , trMatrix );
15
16 // run the survey
17 l . apply( srcImage );
18
19 // retrieve the resulting points.
20 std::vector< lti::fpoint > result;
21 l . getPointList( result );
```

**Listing B.6:** Beispiel für einen Aufruf der Objektvermessung

## B.3. Bildaufnahme

Der nachfolgende Programmausschnitt verdeutlicht die Bildaufnahmephase.

```
1 int picNr    = 0;
2 double degree = 0.0;
3
4 // calculate stepsize
5 double stepsize = ((double)360.0 / (double)scanParam->MEMBUFS) * (double)0.05;
6
7 // rotation time
8 double sleep    = ((double)360.0 / (double)scanParam->MEMBUFS) * (double)150.0;
9
10 // connection to the ps90 is stable
11 if (stepmotor.IsConnected())
12 {
13     // current position
14     double pos = 0;
15
16     // start live mode for fast snapping
17     grabber.startLive( true );
18
19     // take MEMBUFS picture
20     for (int i =0 ; i<scanParam->MEMBUFS ; i++)
21     {
22
23         // set the current position of the axes
24         stepmotor.SetPositionEx(1,0);
25
26         // do rotation
27         stepmotor.MoveEx(1, stepsize , true );
28
29         // wait to reach the correct position
30         Sleep(sleep);
31
32         // snap one image
33         pSink->snapImages(1);
34
35         // increment position
```

```
36     pos = pos + stepsize ;
37 }
38
39 // stop the live video
40 grabber.stopLive ();
41
42 ...
```

**Listing B.7:** Handhabung der Bildaufnahme

# Abbildungsverzeichnis

3.1. Der trigonometrische Zusammenhang einer Rotation . . . . .	13
3.2. Sobel-Faltungskerne . . . . .	21
3.3. Beispiel zur Kantendetektion . . . . .	22
3.4. Veranschaulichung der Hough-Transformation . . . . .	23
4.1. Grundgerüst des Laserscanners . . . . .	26
4.2. Draufsicht des Grundgerüsts . . . . .	27
4.3. Verkleidung des Laserscanners . . . . .	28
4.4. Kamera . . . . .	29
4.5. Laser . . . . .	30
4.6. Drehmesstisch und Positionierungssteuerung . . . . .	31
4.7. Aufbau der Kalibrierungsvorrichtung . . . . .	32
4.8. Kalibrierungsvorrichtung mit eingelegtem Kalibrierungsmuster . . . . .	32
4.9. Kalibrierungsmuster . . . . .	33
5.1. Schematische Darstellung der Oberflächenrekonstruktion . . . . .	37
5.2. Schematische Darstellung der Oberflächenrekonstruktion . . . . .	38
5.3. Beispielhaftes Kalibriermuster . . . . .	40
5.4. Ausrichtung des Lasers . . . . .	41
5.5. Kalibrierungsbild mit <i>Region of interest</i> . . . . .	44
5.6. Kalibrierungsbild mit näherungsweise bestimmten Kantengeraden . . . . .	45
5.7. Beispiele für Intensitätsverläufe senkrecht zu einer Kantengeraden . . . . .	46
5.8. Veranschaulichung des Kantenkorrekturverfahrens . . . . .	47
5.9. Beispielbilder Objektvermessung . . . . .	52
5.10. Analyse eines Lichtschnittes . . . . .	54
5.11. Beispiel für diffuse Reflektionen . . . . .	55
5.12. Filtermaske zur Unterdrückung von diffusen Reflektionen . . . . .	55
5.13. Beispiel Reflektionsfilter . . . . .	56
5.14. Beispiel für einen Querschnitt durch die Laserlinie . . . . .	56

5.15. Rekonstruktion der 3D-Koordinaten . . . . .	57
5.16. Formatierung der 3D-Datenmenge . . . . .	58
5.17. Korrespondierende Passpunkte einer Registrierung . . . . .	60
5.18. Ergebnis einer manuellen Registrierung . . . . .	60
5.19. Resultat einer Dreiecksvermaschung . . . . .	61
5.20. Einwirkung eines Median-Filters . . . . .	64
5.21. Ergebnis eines erfolgreich angewandten Glättungsverfahrens . . . . .	65
5.22. Simplifizierung einer Punktmenge . . . . .	66
5.23. Formatierung der 3D-Datenmenge zur Weiterverwendung . . . . .	67
6.1. Splash Screen der Laserscannersoftware . . . . .	70
6.2. Hauptmenü . . . . .	71
6.3. Setupmenü . . . . .	72
6.4. Aufgaben der Kalibrierungsvorbereitung . . . . .	73
6.5. Das Ergebnis einer Kalibrierung . . . . .	74
6.6. Allgemeine Einstellungen zur Objektvermessung . . . . .	76
6.7. Formatierungsanpassung der Datenmenge . . . . .	77
7.1. Ergebnis der Vermessung des Testobjektes . . . . .	80
7.2. Vergleich zweier Messungen desselben Objektes . . . . .	83
7.3. Stoffteddy mit erheblichen Lücken im Messresultat . . . . .	87
7.4. Originalbilder eines Spielzeugautos . . . . .	88
7.5. Punktwolken des Spielzeugautos . . . . .	88
7.6. Modellierte Darstellung eines Spielzeugautos . . . . .	89
7.7. Originalbilder von zu vermessenden Objekten . . . . .	89
7.8. Resultate zweier Objektvermessungen . . . . .	90
7.9. Vernetzungsergebnisse von Punktmengen . . . . .	90
7.10. Flächenbasierte Darstellung von Messobjekten . . . . .	90
7.11. Dekobirne aus Plastik . . . . .	91
7.12. Messergebnis einer Hundefigur . . . . .	91
7.13. Schale aus Holz . . . . .	92
7.14. Stern aus Ton . . . . .	92
7.15. Ruder aus Holz . . . . .	92
7.16. Nashorn aus Holz . . . . .	93
7.17. Holzfigur eines Pferdes . . . . .	93
7.18. Gesicht aus Ton . . . . .	94

# Tabellenverzeichnis

4.1. Beschreibung der Gerüstmasse . . . . .	27
4.2. Auflistung der Verkleidungsmasse . . . . .	28
6.1. Auflistung der getesteten Betriebssysteme . . . . .	69
7.1. Messgenauigkeit in y-Richtung . . . . .	81
7.2. Kennzahlen der Wiederholpräzision . . . . .	84
7.3. Entwicklungsphasen des Multiplikationsfaktors F . . . . .	85



# Quellenverzeichnis

## Literaturquellen

- [Alexa u. a. 2003] ALEXA, Marc ; BEHR, Johannes ; COHEN-OR, Daniel ; FLEISHMAN, Shachar ; LEVIN, David ; SILVA, Claudio T.: *Computing and Rendering Point Set Surfaces*. IEEE Transactions on Visualization and Computer Graphics, 2003
- [Ansorge und Oberle 2000] ANSORGE, Rainer ; OBERLE, Hans J.: *Mathematik für Ingenieure*. Bd. 1: *Mathematik für Ingenieure, Band 1, 3. überarbeitete Auflage*. Berlin : Wiley-VCH, 2000. – ISBN 3-527-40309-4
- [Benker 2003] BENKER, Hans: *Mathematische Optimierung mit Computeralgebra-systemen: Einführung für Ingenieure, Naturwissenschaftler und Wirtschaftswissenschaftler unter Anwendung von Mathematica, Maple, Mathcad, Matlab und Excel*. Springer, 2003. – ISBN 3-540-44118-2
- [Überhuber 1995] ÜBERHUBER, Christopher W.: *Computer-Numerik 2*. Springer, 1995. – ISBN 3-540-59152-4
- [Bertagnolli 2004] BERTAGNOLLI, Frank: *Vollautomatische robotergestützte 3D-Oberflächenvermessung zur Qualitätssicherung von Werkstücken*. In: VDI/VDE Gesellschaft Mess- und Automatisierungstechnik (Hrsg.): *Sensoren und Messsysteme 2004*, VDI-Berichte 1829, 2004. – ISBN 3-18-091829-2
- [Besl und McKay 1992] BESL, Paul J. ; MCKAY, Neil D.: *A method for registration of 3-D shapes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992
- [Brüderlin und Meier 2001] BRÜDERLIN, Beat ; MEIER, Andreas: *Computergrafik und geometrisches Modellieren*. Vieweg+Teubner, 2001. – ISBN 3-519-02948-0

- [Bunke und Jiang 1997] BUNKE, Horst ; JIANG, Xiaoyi: *Dreidimensionales Computersehen - Gewinnung und Analyse von Tiefenbildern*. Springer, 1997. – ISBN 3-540-60797-8
- [Cignonia u. a. 1998] CIGNONIA, Paolo ; MONTANIA, Claudio ; SCOPIGNO, Roberto: *DeWall - A fast divide and conquer Delaunay triangulation algorithm in E-d*. Elsevier Science Ltd., 1998
- [Fang und Piegl 1995] FANG, Tsung-Pao ; PIEGL, Les A.: *Delaunay triangulation in three dimensions*. IEEE Computer Graphics and Applications, 1995
- [Gonzales und Woods 2002] GONZALES, Rafael C. ; WOODS, Richard E.: *Digital Image Processing, 2nd Edition*. New Jersey : Prentice-Hall, Inc., 2002. – ISBN 0-201-18075-8
- [Gramlich 2004] GRAMLICH, Günter: *Anwendungen der linearen Algebra mit MATLAB*. Leipzig : Fachbuchverlag Leipzig, 2004. – ISBN 3-446-22655-9
- [Haase 2005] HAASE, Jens: *Glättung von Polygonnetzen in medizinischen Visualisierungen*. Diplomarbeit - Institut für Simulation und Graphik, 2005
- [Haberäcker 1989] HABERÄCKER, Peter: *Digitale Bildverarbeitung - Grundlagen und Anwendungen*. München/Wien : Hanser, 1989. – ISBN 3-446-15644-5
- [Hofer und Hutter 2009] HOFER, Simon ; HUTTER, Matthias: *Delaunay-Triangulation - Konstruktion und Anwendung*. Berner Fachhochschule - Technik und Informatik, 2009
- [Jähne u. a. 1995] JÄHNE, Bernd ; MASSEN, Robert ; NICKOLAY, Bertram ; SCHARFENBERG, Harald: *Technische Bildverarbeitung - Maschinelles Sehen*. Berlin/Heidelberg/New York : Springer, 1995. – ISBN 3-540-58641-5
- [Jones 2003] JONES, Thouis R.: *Feature Preserving Smoothing of 3D Surface Scans*. Master thesis - Massachusetts Institute of Technology, 2003
- [Kaiser und Kecher 2008] KAISER, Ulrich ; KECHER, Christoph: *C/C++*. Galileo Press, 2008. – ISBN 978-3-89842-839-2
- [Keßler 2008] KESSLER, Christoph: *Simplifizierung polygonaler Oberflächen*. Diplomarbeit - Fraunhofer Institut Techno- und Wirtschaftsmathematik, 2008

- [Klette u. a. 1996] KLETTE, Reinhard ; KOSCHAN, Andreas ; SCHLÜNS, Karsten: *Computer Vision - Räumliche Information aus digitalen Bildern*. Braunschweig/Wiesbaden : Vieweg, 1996. – ISBN 3-528-06625-3
- [Kraus 2000] KRAUS, Karl: *Photogrammetrie: Topographische Informationssysteme*. Walter de Gruyter, 2000. – ISBN 3-110-18164-9
- [Mackens und Voss 1993] MACKENS, Wolfgang ; VOSS, Heinrich: *Mathematik 1 für Studierende der Ingenieurwissenschaften*. Aachen : HECO-Verlag, 1993. – ISBN 3-930121-00-X
- [Masuda und Yokoya 1994] MASUDA, Takeshi ; YOKOYA, Naokazu: *A robust method for registration and segmentation of multiple rangeimages*. Proceedings of the 1994 Second CAD-Based Vision Workshop, 1994. – ISBN 0-818-65310-8
- [Maurer u. a. 1996] MAURER, Calvin R. ; ABOUTANOS, Georges B. ; DAWANT, Benoit M. ; MACIUNAS, Robert J. ; FITZPATRICK, Michael: *Registration of 3-D images using weighted geometrical features*. IEEE Transactions on Medical Imaging, 1996
- [Nidermair 2006] NIDERMAIR, Elke: *Latex: Das Praxisbuch*. Franzis, 2006. – ISBN 3-77236-930-8
- [Nidermair und Nidermair 2006] NIDERMAIR, Elke ; NIDERMAIR, Michael: *L<sup>A</sup>T<sub>E</sub>X: Das Praxisbuch*. Poing : Franzis Verlag, 2006. – ISBN 3-7723-6930-8
- [Pagés u. a. 2003] PAGÉS, Jordi ; SALVI, Joaquim ; GARCIA, Rafael ; MATABOSCH, Carles: *Overview of coded light projection techniques for automatic 3D profiling*. IEEE International Conference on Robotics and Automation, 2003. – ISBN 0-7803-7736-2
- [Paulus und Hornegger 1995] PAULUS, Dietrich W. R. ; HORNEGGER, Joachim: *Pattern Recognition and Image Processing in C++*. Braunschweig/Wiesbaden : Vieweg, 1995. – ISBN 3-528-05491-3
- [Press u. a. 1992] PRESS, William H. ; TEUKOLSKY, Saul A. ; VETTERLING, William T. ; FLANNERY, Brian P.: *Numerical Recipes in C, 2nd Edition*. Cambridge : Cambridge University Press, 1992. – ISBN 0-521-43108-5
- [Reimann 2008] REIMANN, Christian: *Entwicklung eines Steuerungsmoduls für einen 3D-Laserscanner*. Diplomarbeit. 2008

- [Rogers und Adams 1976] ROGERS, D. F. ; ADAMS, J. A.: *Mathematical Elements for Computer Graphics*. New York : McGraw-Hill, 1976. – ISBN 0-07-053530-2
- [Schuhfuss 2007] SCHUHFUSS, Martin: *Realisierung eines Laser-Triangulationssensors zur 3D-Objektvermessung*. Bachelorarbeit. 2007
- [Stroustrup 2000] STROUSTRUP, Bjarne: *The C++ Programming Language, Special Edition*. Boston : Addison-Wesley, 2000. – ISBN 0201700735
- [Tönnies 2005] TÖNNIES, Klaus D.: *Grundlagen der Bildverarbeitung*. München : Pearson Studium, 2005. – ISBN 3-8273-7155-4
- [Willms 2008] WILLMS, André: *Einstieg in Visual C++ 2008*. Galileo Press, 2008. – ISBN 978-3-8362-1193-2
- [Wolf 2006] WOLF, Jürgen: *C++ von A bis Z*. Galileo Press, 2006. – ISBN 3-89842-643-2
- [Zhang u. a. 2002] ZHANG, Li ; CURLESS, Brian ; SEITZ, Steven M.: *Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming*. Department of Computer Science and Engineering, 2002

## Internetquellen

- [Adobe-Systems 2009] ADOBE-SYSTEMS: *Adobe Photoshop*. Zugriffsdatum: 17.05.2009. 2009. –  
URL: <http://www.adobe.com/de/products/photoshop/>
- [Alvarado u. a. 2005] ALVARADO, Pablo ; DOERFLER, Peter ; CANZLER, Ulrich: *LTI-Lib*. Homepage des LTI-Lib-Projektes der RWTH Aachen, Stand: 24.11.2005. 2005. –  
URL: <http://ltilib.sourceforge.net>
- [Autodesk 2009] AUTODESK: *AutoCAD*. Zugriffsdatum: 17.05.2009. 2009. –  
URL: <http://www.autodesk.de>
- [Beucke u. a. 2009] BEUCKE, Karl ; BURGER, Christian ; FIRMENICH, Berthold ; NOCKEN, Peter M.: *CADEMIA*. Zugriffsdatum: 17.05.2009. 2009. –  
URL: <http://www.cademia.org>

- [Boost 2009] BOOST: *Boost C++ Libraries*. Stand: 02.05.2009. 2009. –  
URL: <http://www.boost.org>
- [Bosch 2009] BOSCH: *Bosch Rexroth*. Zugriffsdatum: 08.04.2009. 2009. –  
URL: <http://www.boschrexroth.com>
- [Geomagic 2009] GEOMAGIC: *Geomagic Studio*. Zugriffsdatum: 17.05.2009. 2009.  
–  
URL: <http://www.geomagic.com/de/products/studio/>
- [Gühring 2001] GÜHRING, Dipl.-Ing. J.: *3D-Erfassung und Objektrekonstruktion mittels Streifenprojektion*. Dissertation, Universität Stuttgart, Stand: 21.11.2001. 2001. –  
URL: [http://deposit.ddb.de/cgi-bin/dokserv?idn=980546680&dok\\_var=d1&dok\\_ext=pdf&filename=980546680.pdf](http://deposit.ddb.de/cgi-bin/dokserv?idn=980546680&dok_var=d1&dok_ext=pdf&filename=980546680.pdf)
- [Hartley und Zisserman 2000] HARTLEY, R. I. ; ZISSERMAN, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. – URL <http://www.robots.ox.ac.uk/~vgg/azbook.html>
- [van Heesch 2009] HEESCH, Dimitri van: *Doxygen*. Zugriffsdatum: 30.04.2009. 2009. –  
URL: <http://www.doxygen.org>
- [Heitler 2007] HEITLER, Dr. W.: *Scatter3D*. Stand: 02.06.2007. 2007. –  
URL: <http://www.st-andrews.ac.uk/~wjh/scatter3d/>
- [Jackèl 2005] JACKÈL, D.: *3D Computer Vision*. Erschienen als Teil des Skriptes zur Vorlesung 'Computer Vision 2' des Studiengangs Informatik an der Universität Rostock. 2005. – URL [http://www.icg.informatik.uni-rostock.de/Lehre/CV2/script0304/06\\_3dcv.pdf](http://www.icg.informatik.uni-rostock.de/Lehre/CV2/script0304/06_3dcv.pdf)
- [Jezouin u. a. 1988] JEZOUIN, J. L. ; SAINT-MARC, P. ; MEDIONI, G.: Building an Accurate Range Finder with Off the Shelf Components. In: *CVPR'88 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Ann Arbor, MI, June 5–9, 1988)*. Washington, DC. : Computer Society Press, Juni 1988, S. 195–200. – URL [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?isNumber=5032&prod=CNF&arnumber=196236&arSt=+195&ared=+200&arNumber=196236](http://ieeexplore.ieee.org/xpls/abs_all.jsp?isNumber=5032&prod=CNF&arnumber=196236&arSt=+195&ared=+200&arNumber=196236)

- [Lichtblau und Weisstein 2004] LICHTBLAU, Daniel ; WEISSTEIN, Eric W.: *Condition Number*. Erschienen auf der Webseite mathworld.wolfram.com. 2004. – URL <http://mathworld.wolfram.com/ConditionNumber.html>
- [Link und Sättele 2009] LINK, Matthias ; SÄTTELE, Manuel: *Digitale Archäologie*. Stand: 16.04.2009. 2009. – URL: <http://www.digitale-architektur.de>
- [Maplesoft 2009] MAPLESOFT: *Maple*. Zugriffsdatum: 17.05.2009. 2009. – URL: <http://www.maplesoft.com/products/maple/>
- [Meisel 2008a] MEISEL, Prof. Dr.-Ing. A.: *3D Bildverarbeitung*. Vortragsfolien zu Grundlagen der 3D-Bildverarbeitung, Stand: 03.03.2008. 2008. – URL: <http://www.informatik.haw-hamburg.de/1708.html>
- [Meisel 2008b] MEISEL, Prof. Dr.-Ing. A.: *Robot Vision*. Vorlesungsskript des Faches Robot Vision, Stand: 07.03.2008. 2008. – URL: [http://www.informatik.haw-hamburg.de/wp\\_robot\\_vision.html](http://www.informatik.haw-hamburg.de/wp_robot_vision.html)
- [Microsoft-Corporation 2007] MICROSOFT-CORPORATION: *Microsoft Visual Studio*. Stand: 19.11.2007. 2007. – URL: <http://msdn.microsoft.com/de-de/vstudio/products/>
- [Microsoft-Corporation 2008] MICROSOFT-CORPORATION: *Microsoft Office Visio*. Zugriffsdatum: 02.12.2008. 2008. – URL: <http://office.microsoft.com/visio/>
- [OWIS 2009] OWIS: *OWIS Stufen*. Zugriffsdatum: 16.05.2009. 2009. – URL: <http://www.owis-staufen.de>
- [Rauth 2009] RAUTH, Hans-Helmut: *Wenzel GearTec*. Zugriffsdatum: 16.05.2009. 2009. – URL: <http://www.wenzel-geartec.de>
- [Schenk 2008] SCHENK, Christian: *MiKTeX*. Stand: 08.02.2008. 2008. – URL: <http://www.miktex.org>
- [Sycode 2009] SYCODE: *Point Cloud*. Zugriffsdatum: 17.05.2009. 2009. – URL: [http://www.sycode.com/products/point\\_cloud/](http://www.sycode.com/products/point_cloud/)

- [The-Imaging-Source 2008a] THE-IMAGING-SOURCE: *IC Imaging Control*. Zugriffsdatum: 21.09.2008. 2008. –  
URL: <http://www.imagingcontrol.com>
- [The-Imaging-Source 2008b] THE-IMAGING-SOURCE: *Overlay*. Zugriffsdatum: 21.09.2008. 2008. –  
URL: [http://www.imagingcontrol.com/en\\_US/support/documentation/class/Overlay.htm](http://www.imagingcontrol.com/en_US/support/documentation/class/Overlay.htm)
- [The-Imaging-Source 2009a] THE-IMAGING-SOURCE: *Cameras*. Stand: 16.05.2009. 2009. –  
URL: <http://www.theimagingsource.com/products/cameras/>
- [The-Imaging-Source 2009b] THE-IMAGING-SOURCE: *Lenses*. Stand: 16.05.2009. 2009. –  
URL: <http://www.theimagingsource.com/products/optics/lenses/>
- [Weisstein 2006] WEISSTEIN, Eric W.: *Singular Value Decomposition*. Erschienen auf der Webseite [mathworld.wolfram.com](http://mathworld.wolfram.com). 2006. – URL <http://mathworld.wolfram.com/SingularValueDecomposition.html>
- [Weisstein 2007] WEISSTEIN, Eric W.: *QR Decomposition*. Erschienen auf der Webseite [mathworld.wolfram.com](http://mathworld.wolfram.com). 2007. – URL <http://mathworld.wolfram.com/QRDecomposition.html>
- [Zimmermann und Lang 2004] ZIMMERMANN, Kurt M. ; LANG, Thomas: *Z-LASER*. Stand: 14.01.2004. 2004. –  
URL: <http://www.z-laser.com>