

# Exemplarische Implementierung des Lichtschnittverfahrens mittels einer selbstgebauten Scan-Vorrichtung

David Wichter

Master-Projektstudium

Betreuer: Professor Dr. Jürgen Lohscheller

Trier, Abgabedatum

---

## **Kurzfassung**

TODO

---

# Inhaltsverzeichnis

<b>1 Einleitung .....</b>	<b>1</b>
<b>2 Das Verfahren in der Theorie .....</b>	<b>2</b>
2.1 Das mathematische Modell einer Kamera.....	2
2.1.1 Die Kamera Kalibrierung .....	2
2.2 Der Lichschnitt .....	2
<b>3 Implementierung .....</b>	<b>3</b>
3.1 Scanner Konstruktion .....	3
3.2 Der Scavorgang .....	5
3.2.1 Kalibrierung der internen Kameraparameter .....	6
3.2.2 Datenerhebung .....	7
3.2.3 Farbkalibrierung .....	7
3.2.4 Kalibrierung der externen Kameraparameter.....	10
3.2.5 Bestimmung der Laserlinie .....	10
<b>4 Ergebnisse.....</b>	<b>13</b>
4.1 Laserlinienbestimmung .....	13
4.2 Messgenauigkeit .....	14
<b>5 Fazit .....</b>	<b>18</b>
<b>Literaturverzeichnis .....</b>	<b>19</b>

---

## **Abbildungsverzeichnis**

3.1	Frontal Ansicht . . . . .	4
3.2	Seitliche Ansicht . . . . .	4
3.3	Nahaufnahme der Millimeter-Anzeige . . . . .	4
3.4	Der schematische Ablauf des Scavorgangs . . . . .	6
3.5	Berechnung der Distanz zwischen zwei Linien . . . . .	9
3.6	Extraktion der Laserlinie . . . . .	11
4.1	Farbkanäle einer einzelnen Bildspalte . . . . .	15
4.2	Gegenüberstellung von optimaler Schrägen und gemessener Punktwolke . . . . .	16

# **1**

---

## **Einleitung**

TODO

## **2**

---

### **Das Verfahren in der Theorie**

#### **2.1 Das mathematische Modell einer Kamera**

TODO

##### **2.1.1 Die Kamera Kalibrierung**

TODO

#### **2.2 Der Lichschnitt**

TODO

# 3

---

## Implementierung

### 3.1 Scanner Konstruktion

Um das Lichtschnittverfahren angemessen umsetzen zu können, müssen zuerst brauchbare Daten erhoben werden. Mit Daten sind in diesem Kontext die Fotografien gemeint, auf denen das zu vermessende Objekt inklusive der projizierten Laserlinie zu sehen ist. Um diese Reihe von Bildern zu erstellen, ist eine Vorrichtung von Nöten, die es ermöglicht, die Webcam und den Laser so zu fixieren, dass das zu vermessene Objekt mit Leichtigkeit fotografierbar ist. Zusätzlich müssen sowohl der Laser als auch die Kamera so beweglich sein, dass das zu vermessene Objekt in kontrollierten Abständen mit dem Laser abgetastet werden kann, sich das räumliche Verhältnis zwischen Webcam und Laser sich jedoch nicht verändert. Für das vorliegende Projekt wurde versucht, dies unter Berücksichtigung einer einfachen und günstigen Konstruktion zu realisieren. Die in dieser Ausarbeitung vorgeschlagene Lösung setzt darauf, Kamera und Laser übereinander auf einer Pappelholzplatte zu fixieren. Diese Holzplatte befindet sich auf zum Boden senkrecht befindlichen Schlaufen aus stabilem Gurt, welche um zwei PVC-Röhren gespannt sind. Besagte PVC-Röhren ruhen in horizontaler Lage parallel zum Boden in zwei Holzstreben, welche wiederrum senkrecht auf einer Basis aus dünnem Pappelholz geklebt sind. Die Gesamtkonstruktion kann in Abb. 3.1 und 3.2 und betrachtet werden.

Die vom Laser aufgespannte Ebene befindet sich somit parallel zum Grund, während die Kamera schräg von oben auf das zu vermessene Objekt schaut. Mittels des Gurts kann die Holzplatte, auf der Kamera und Laser montiert sind, von unten nach oben bewegt werden, um das Objekt so abzutasten. Um zu kontrollieren, wie weit der Laser über das Objekt bewegt wurde, ist auf der Rückseite des Gurts eine Anzeige installiert, welche sich gen Boden bewegt sobald Laser und Kamera nach oben steigen. Die Anzeige zeigt an den Seiten der Holzstreben, an denen eine Skala in Millimeter-Papier angebracht ist, wie weit sich die Holzplatte von der Startposition aus nach oben bewegt hat. In Abb. 3.3 ist eine Nahaufnahme dieser Anzeige zu sehen.

Ein wesentlicher Vorteil der vorgenommenen Konstruktion liegt in der Parallelität der Laser-Ebene zum Boden. Auf diese Weise muss man sich bei der späteren

**Abb. 3.1.** Frontal Ansicht**Abb. 3.2.** Seitliche Ansicht**Abb. 3.3.** Nahaufnahme der Millimeter-Anzeige

Bildverarbeitung keine Sorgen machen, dass die Laser-Linie auf den Untergrund fällt, auf dem das zu vermessene Objekt ruht. Wäre dies der Fall, müssten bei der späteren Auslesung die Teile der Laser-Linie, die auf das Objekt fallen, von denen, die auf den Untergrund fallen, aufwendig getrennt werden. Ebenfalls macht der Aufbau die Nachbearbeitung der gemessenen Weltkoordinaten im Gegensatz zu vergleichbaren Konstruktionen (z.B. bei schwenkbaren Scannern) besonders simpel: Da sich die Apparatur zwischen Aufnahmen lediglich entlang der Z-Achse bewegt und dieser Abstand bekannt ist, kann auf den Z-Achsen-Anteil des Messungsergebnis der besagte Abstand einfach addiert werden.

Der Gesamtaufbau setzt vor allem auf eine praktikable Lösung, die problemlos nachgebaut werden kann und sich dem zu Grunde liegenden Problem so nähert, dass anschließende Verarbeitungen und Berechnungen erleichtert werden. Zudem

ist die Konstruktion kostengünstig. In Tabelle 3.1 können die Preise in Euro eingesehen werden, die alle Komponenten (außer Kleinteile wie einzelne Schrauben, Holzleim etc.) gekostet haben. Zusammen kommt der Scanner auf einen Preis von ca. 45,14€

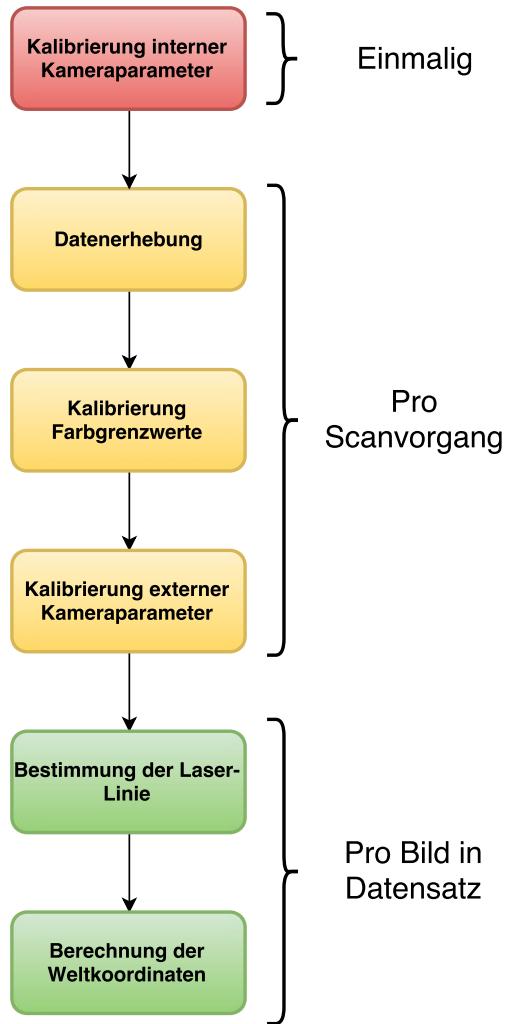
Komponente	Preis in Euro
Kamera „TeckNet C016 USB HD Webcam“	13,99 €
Linienlaser	ca. 20 €
PVC-Röhre	1,69 €
Holzleiste Buche (Seitenstreben)	3,79 €
Pappel-Sperrholz	2,29 €
Gurtband	3,38 €

**Tabelle 3.1.** Kosten der einzelnen Scanvorrichtungskomponenten

## 3.2 Der Scavorgang

Um ein zu vermessendes Objekt in eine Reihe von abgetasteten Weltkoordinaten umzuwandeln wird in der vorliegenden Implementierung ein Prozess von verschiedenen Stufen vorgeschlagen. Der Gesamtprozess ist schematisch in Abb. 3.4 dargestellt. Die einzelnen Stufen des Prozesses sind dabei nach der Abhängigkeit ihrer Ergebnisse und der daraus folgenden Anzahl an nötigen Wiederholungen sortiert. Mit anderen Worten: Die Aktionen, die nur einmal ausgeführt werden müssen um mehrere Scavorgänge ausführen zu können werden zuerst vorgenommen, während Aktionen die beispielsweise pro aufgenommenen Bild ausgeführt werden müssen, am Schluss folgen. Dies ist für denjenigen, der den Scavorgang durchführt eine geeignete Reihenfolge, da so Teilergebnisse für mehrere Scanprozesse wiederverwendet werden können und bei erfolglosen Stufen der Gesamtprozess nicht von neu angestoßen werden muss.

Die Kalibrierung der internen Kameraparameter beispielsweise ist unabhängig von den anderen Umständen des eigentlichen Scavorgangs und muss nicht für jede Messreihe erneut stattfinden, sondern lediglich wenn sich die Fokus-Einstellung der Kamera ändert. In der darauf folgenden Stufe des Scannens werden die nötigen Fotografien aufgenommen, was im Folgenden als Datenerhebung bezeichnet wird. Anschließend werden anhand einiger weniger dieser Bilder weitere Kalibrierungen vorgenommen, welche allerdings für den Rest des Datensatzes persistent sind und nur von Scandurchlauf zu Scandurchlauf wiederholt werden müssen. Zuletzt werden auf allen Bildern des Datensatzes die beiden Schritte der Laserlinienerkennung und Weltkoordinatenberechnung ausgeführt, welche für jedes Bild des Datensatzes andere Ergebnisse liefern und damit am Ende der Pipeline stehen. Die Weltkoordinaten, die als Resultat eines jeden Bildes berechnet werden, werden gesammelt und stehen als Punkte-Wolke zur weiteren Verarbeitung zur Verfügung. Im Folgenden werden nun die einzelnen Stufen des Scavorgangs einzeln näher erläutert.



**Abb. 3.4.** Der schematische Ablauf des Scavorgangs

### 3.2.1 Kalibrierung der internen Kameraparameter

In Abschnitt 2.1.1 wurde die Theorie hinter der Kamerakalibrierung bereits erläutert, daher soll das Verfahren an dieser Stelle lediglich aus praktischer Sicht betrachtet werden. Aus Gründen, welche in 3.2 dargelegt wurden, ist der Kamerakalibrierungsprozess in zwei zeitlich voneinander getrennte Teile gespalten. Am Anfang steht hierbei die Bestimmung der internen Kameraparameter, insbesondere der internen Kameramatrix. Um diese abzuschätzen wurde auf die in Matlab integrierte Kamerakalibrierungsapp zurückgegriffen, welche aus einem Set von vorher aufgenommenen Bildern die Kameraparameter (wie in 2.1.1 beschrieben) schätzt. Dabei wird die Reihe der Kalibrierungsbilder in die App importiert und anschließend von jener verarbeitet. Bevor die nun geschätzte interne Kameramatrix als eigenständige Datei exportiert werden kann, sollten die sogenannten Reprojection Errors begutachtet werden. Diese stellen eine Metrik dar, welche besagt wie erfolgreich die Kalibrierung verlaufen ist und ergibt sich aus dem Abstand zwischen den im Bild erkannten Schachbrett-Punkten und den auf das Bild zurück projizierten korrespondierenden Weltkoordinaten.

(mehr dazu auch bei [Mat] und [Sta]). Laut [Mat] gilt jeder durchschnittliche Fehler von unter einem Pixel als akzeptabel:

„As a general rule, reprojection errors of less than one pixel are acceptable.“

Bei den dieser Ausarbeitung mitgelieferten Kalibrierungsbildern wurde ein durchschnittlicher Reprojection Error von 0,3572 erreicht. Am Ende der Kalibrierung muss die interne Kameramatrix gespeichert und für die weiteren Schritte hinterlegt werden.

### 3.2.2 Datenerhebung

In der Phase der Datenerhebung werden die Fotografien des zu vermessenen Objektes aufgenommen. Dabei kommt die Konstruktion zum Tragen die in Kapitel 3.1 beschrieben wurde. Zuerst muss die Messplatte in ihre Startposition gebracht werden, das heißt die Millimeter-Anzeige muss einen Abstand von 0 Millimetern anzeigen (vgl. Abb. 3.3). Anschließend muss das Schachbrettmuster, das auch bei der Kalibrierung der internen Kameraparametern zum Einsatz kam, so platziert werden, dass zum einen die Kamera alle Quadrate des Musters erkennen kann und zum anderen das zu vermessene Objekt so aufgestellt werden kann, dass die Laserlinie auf die unterste Peripherie des Objektes projiziert werden kann, also da, wo die Messung beginnen soll. Aufgrund der Konstruktion des Scanners muss das Schachbrettmuster dafür auf einer Erhöhung von ca. 9 cm ruhen. Für das erste Bild ist es jedoch wichtig, eine Aufnahme des Musters zu machen, auf dem das zu vermessene Objekt nicht abgebildet ist. Dieses Bild wird für die Kalibrierung der externen Kameraparameter benötigt, welche im kommenden Abschnitt 3.2.4 beschrieben ist. Von nun an ist es für den Rest des Scavorgangs nicht mehr erlaubt, den Neigungswinkel der Kamera zu ändern.

Anschließend wird das zu vermessene Objekt so auf dem Schachbrettmuster platziert, dass die auf das Objekt projizierte Laserlinie in ihrer ganzen Länge zu erkennen ist und es wird eine Aufnahme gemacht. Nun muss die Messplatte mithilfe der Millimeter-Anzeige um einen vorher festgelegten Millimeter-Betrag nach oben bewegt werden. Der festgelegte Betrag muss für die spätere Berechnung der Weltkoordinaten festgehalten werden. Anschließend wird die nächste Fotografie geschossen. Dabei ist zu beachten, dass sich zwischen den Bildern nichts anderes als die Messplatte bewegen sollte und der vorher festgelegte Abstand so genau wie möglich eingehalten wird. In dieser Manier wird nun das gesamte Objekt mit dem Laser abgetastet und je nach gewähltem Abstand ergeben sich so mehr oder weniger Bilder, die für die folgenden Phasen abgespeichert werden.

### 3.2.3 Farbkalibrierung

Die Laserlinie, welche auf das zu vermessene Objekt projiziert wird und die nötigen Informationen zur Bestimmung der Tiefeninformation der Weltkoordinaten liefert, muss in jedem aufgenommenen Bild neu lokalisiert werden (vgl. den kommenden Abschnitt 3.2.5). Der Ansatz, die Laserlinie im Bild zu bestimmen, basiert auf einem globalen Farbgrenzwert, mit dem die Laserlinie so gleichmäßig wie möglich segmentiert werden kann. Die Bestimmung und Speicherung dieses Grenzwertes geschieht in dem Schritt der Farbkalibrierung. Unter den Voraussetzungen dass

1. die projizierte Laserlinie den höchsten Rot-Anteil im Bild besitzt und
2. sich die Licht- und Farbumgebung des Datensatzes zwischen Bildern nicht signifikant ändert

kann der globale Grenzwert anhand eines einzelnen Bildes aus dem Datensatz ermittelt werden, welcher im Anschluss für die restlichen Bilder der Messreihe

wiederverwendet werden kann.

Die Farbkalibrierung soll also den Grenzwert liefern, mit der die spätere Lokalisierung der Laserlinie in Pixelkoordinaten am besten funktioniert. Gearbeitet wird dabei in dem YCbCr-Farbraum, da dieser es auf eine einfache Weise erlaubt, Farbinformationen von Helligkeitsinformationen zu trennen und den Rotanteil dennoch in einem einzelnen Kanal (dem Cr-Kanal) zu isolieren. Der Ansatz, den die vorliegende Implementierung verfolgt, ist, den Benutzer einmalig die Laserlinie mittels eines Splines nachzeichnen zu lassen, und diesen „Goldstandard“ automatisch unter Anwendung des in 3.2.5 beschriebenen Verfahrens und steigender Cr-Grenzwerte mit vielen extrahierten Laserlinien zu vergleichen. Der Cr-Grenzwert, mit dem diejenige Laserlinie extrahiert wurde welche dem Goldstandard am nächsten kommt, ist der optimale Grenzwert und wird für die kommenden Laserlinienlokalisierungen abgespeichert.

Um zu bestimmen, wie nah eine Pixellinie dem vom Nutzer bestimmten Spline und damit der Laserlinie kommt, ist eine Metrik in Form einer Distanzfunktion von Nöten. Pixellinien, bei der diese Distanzfunktion geringe Ergebnisse liefert, sind sich also näher und somit ähnlicher als Linien, die große Werte liefern. Im Folgenden wird die verwendete Distanzfunktion definiert.

Sei die Breite des gegebenen Bildes in Pixeln definiert als  $w$  und die Höhe in Pixeln als  $h$ . Die Spalten des Bildes lassen sich also mit Elementen der Spaltenindex-Menge  $C = \{1, 2, 3, \dots, w\}$  indizieren, während die Reihen sich analog mit Indizes der Reihenindex-Menge  $R = \{1, 2, 3, \dots, h\}$  bestimmen lassen. Die Position eines Pixels im Bild kann also mittels eines Tupels  $p = (c, r)$  mit  $c \in C$  und  $r \in R$  bestimmen. Eine Pixellinie sei nun definiert als eine rechtseindeutige Relation  $L \in C \times R$ , welche durch eine Menge von geordneten Tupeln gegeben ist (Beispiel:  $L = \{(1, 500), (2, 501), (3, 500), (5, 499)\}$ ). Die Rechtseindeutigkeit ist hierbei essentiell: Keiner Spalte des Bildes darf mehr als eine Reihe zugewiesen werden. Sei weiterhin mit  $\text{Dom}(L) = \{c \in C \mid \exists r \in R : (c, r) \in L\}$  die Definitionsmenge einer gegebenen Pixellinie  $L$  gemeint.

Seien nun zwei Pixellinien  $L_1$  und  $L_2$  gegeben. Die Menge

$$S_{1,2} = \{\text{abs}(r_1 - r_2) \mid \exists c \in C : (c, r_1) \in L_1 \wedge (c, r_2) \in L_2\} \quad (3.1)$$

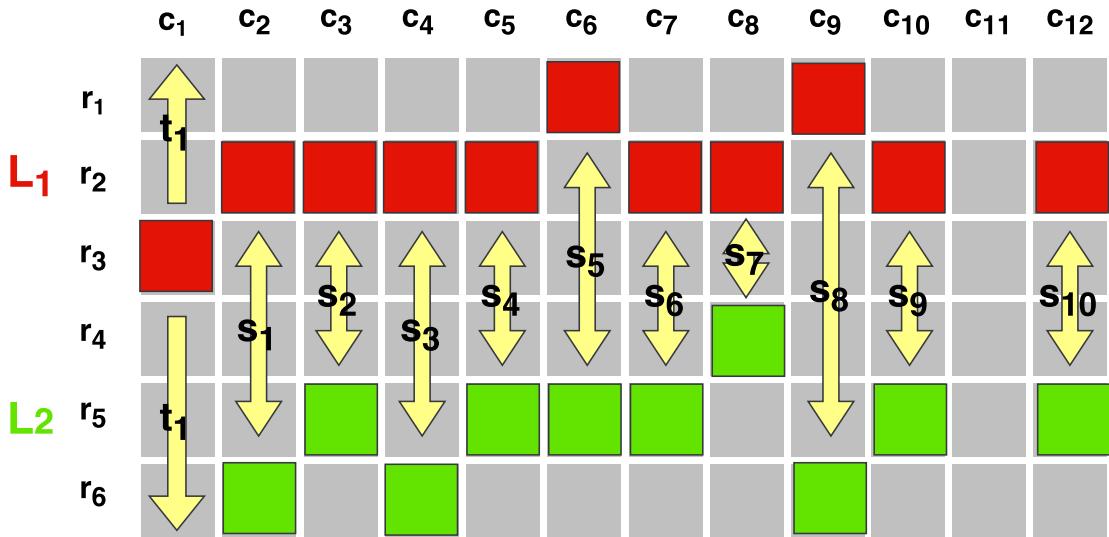
ist dann die Menge aller vertikalen Pixel-Abstände in Pixelspalten, in denen beide Pixellinien eine Pixelkoordinate besitzen.

$$T_{1,2} = (\text{Dom}(L_1) \cup \text{Dom}(L_2)) \setminus (\text{Dom}(L_1) \cap \text{Dom}(L_2)) \quad (3.2)$$

ist die Menge aller Spaltenindizes beider Linien, die in der jeweils anderen Pixellinie auf keine Reihe abbilden. Mittels dieser beiden Mengendefinitionen kann nun die Distanzfunktion  $D : (C \times R) \times (C \times R) \rightarrow \mathbb{R}$  definiert werden als:

$$D(L_1, L_2) = \frac{1}{|S_{1,2}| + |T_{1,2}|} \sum_{x \in S_{1,2}} x + (|T_{1,2}| * h) \quad (3.3)$$

$D$  berechnet also das arithmetische Mittel aller vertikalen Pixelabstände der beiden Pixellinien. Dort wo die beiden Pixellinien sich horizontal überlappen, kann dafür der Pixelabstand genommen werden. In den Spalten, wo jedoch nur eine Linie definiert ist, kann kein direkter Abstand berechnet werden, daher wird ein „Bestrafungs-Betrag“ in Form der Höhe des Bildes auf die Summe gerechnet. Dieser ermöglicht, dass Pixellinien mit viel horizontaler Überschneidung näher bzw. in diesem Kontext ähnlicher zueinander eingestuft werden, als Linien, die horizontale Lücken aufweisen. Abb. 3.5 verbildlicht die Berechnung der Metrik etwas deutlicher.



**Abb. 3.5.** Berechnung der Distanz zwischen zwei Linien. Zu sehen ist ein Schema der Berechnung aus Gleichung 3.3. Die einzelnen grauen Kästchen stellen Pixel aus einem Bild mit Spalten  $C = \{c_1 \dots c_{12}\}$  und Reihen  $R = \{r_1 \dots r_6\}$  dar. Die rot und grün eingefärbten Pixel sind Elemente von  $L_1$  und  $L_2$  und  $S = \{s_1 \dots s_2\}$  und  $T = \{t_1\}$  entsprechen den Definitionen aus Gleichungen 3.1 und 3.2

Zusammengefasst läuft die Farbkalibrierung folgender Maßen ab: Zuerst wird der Benutzer dazu aufgefordert mittels einiger Punkte die horizontal verlaufende projizierte Laserlinie mittels eines Splines nachzuzeichnen. Anschließend wird der globale Cr-Grenzwert mit null initialisiert. Die erste Laserlinie wird mit diesem Grenzwert lokalisiert (siehe dazu 3.2.5) und wird mit der oben beschriebenen Metrik  $D$  wird die Ähnlichkeit der gefundenen Linie zum „Goldstandard“ bestimmt. Der aktuelle Cr-Grenzwert nun als am besten passender Grenzwert vermerkt. Als nächstes wird der Grenzwert um einen bestimmten Wert erhöht und die Laserlinie wird wieder lokalisiert. Wenn diese Linie näher an der vom Benutzer festgelegten Pixellinie (dem Goldstandard) liegt, wird der neue, erhöhte Cr-Grenzwert als optimal vermerkt. Dieses Verfahren geht so lange weiter, bis für einen neuen Grenzwert keine Linie gefunden werden kann, die näher am Goldstandard liegt. Der gefundene Grenzwert wird nun für die folgenden Schritte des Scavorgangs gespeichert.

### 3.2.4 Kalibrierung der externen Kameraparameter

Um Koordinaten der projizierten Laserlinie auf dem zu vermessenden Objekt nicht nur in Kamera- sondern in Weltkoordinaten zu berechnen, ist es nötig, die Position und Rotation der Kamera im Weltkoordinatensystem zu kennen. Diese Informationen erhält man im Schritt der Kalibrierung der externen Kameraparameter. Sofern die Kamera zwischen Bildaufnahmen nicht bewegt wird, muss diese Kalibrierung nur einmal ausgeführt werden. Dafür ist eine Aufnahme des Schachbrettmusters notwendig, welches wie in 3.2.2 beschrieben aufgenommen werden muss. Anhand dieses Bildes werden aus dem Schachbrettmuster Punkte im Weltkoordinatensystem und deren korrespondierende Bildpunkte generiert. Mittels ersterer und letzterer Punktemengen und den internen Kameraparametern aus Schritt 3.2.1 kann Matlab anschließend die Position und Rotation der Kamera gemäß dem in Kapitel 2.1.1 vorgestellten Verfahren ermitteln.

### 3.2.5 Bestimmung der Laserlinie

Bevor aus der Laserlinie Tiefeninformationen gelesen werden können, muss die Position der Linie in Bildkoordinaten bestimmt werden. Da sich die Laserlinie von Bild zu Bild verformen kann und nicht für den gesamten Datensatz gleich bleibt muss dieser Schritt, als erster Schritt des Verfahrens auf jedem Bild des Datensatzes ausgeführt werden. Als erstes wird hierfür die Laserlinie mittels des Cr-Grenzwerts segmentiert, der in Kapitel 3.2.3 ermittelt wurde. Man erhält eine Maske, in der alle Pixel, die unter dem Cr-Grenzwert liegen, geschwärzt werden. Es bleibt also nur die Segmentierung der Laserlinie übrig (vgl. Abb. 3.6). Nun kann für jede Spalte abgeschätzt werden, wo in dieser Spalte die Laserlinie auftritt, falls sie überhaupt in dieser Bildspalte zu finden ist. Dies geschieht indem ein arithmetisches Mittel über die Reihenindizes der gegebenen Spalte berechnet wird, welche mit den Rotwerten des jeweils indizierten Pixels gewichtet wird. Gegeben seien die bestehenden Definitionen aus Abschnitt 3.2.3. Sei zudem  $Red(c, r)$  für  $c \in C$  und  $r \in R$  der Rotwert des Pixels  $(c, r)$ . Die Menge aller zu einem Spaltenindex  $c \in C$  gehörenden Reihenindizes sei definiert als:

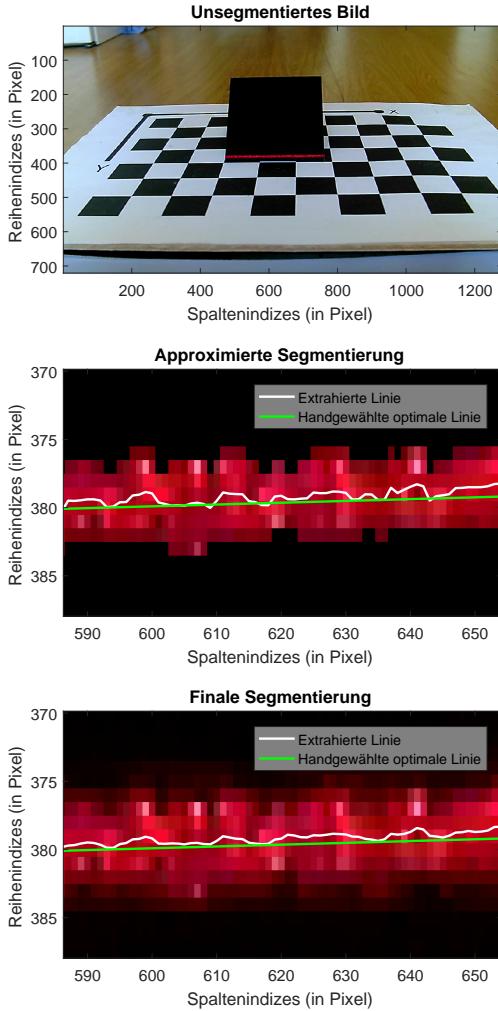
$$R_c = \{r \in R \mid (c, r) \in C \times R\} \quad (3.4)$$

Soll nun die Laserlinie in der gegebenen Spalte  $c \in C$  angenähert werden, wird der korrespondierende Reihenindex berechnet als:

$$LineRow(c) = \frac{\sum_{x \in R_c} Red(c, x) * x}{\sum_{x \in R_c} Red(c, x)} \quad (3.5)$$

Das Ergebnis ist eine subpixelgenaue Bestimmung der Höhe der Laserlinie in Spalte  $c$ . Die gesamte extrahierte Pixellinie für ein gegebenes segmentiertes Bild ist also definiert als:

$$L_{auto} = \{(c, r) \in C \times R \mid \forall c: r = LineRow(c) \wedge r > 0\} \quad (3.6)$$



**Abb. 3.6.** Extraktion der Laserlinie.

linie dar aufgrund derer im folgenden Verlauf des Scavorgangs in Weltkoordinaten berechnet werden.

### Berechnung der Weltkoordinaten

Am Ende des Scavorgangs steht die Berechnung der Weltkoordinaten. Hier werden für jedes aufgenommene Bild die Bildkoordinaten der in Schritt 3.2.5 bestimmten Pixellinie  $L_{auto}$  wie folgt umgerechnet. Sei

$$\vec{p}_{Bild} = \begin{pmatrix} x \\ y \end{pmatrix} \quad (3.7)$$

ein als Vektor interpretierter Bildkoordinatenpunkt  $(x, y)$  in  $L_{auto}$ . Der Punkt kann mit der internen Kameramatrix  $K$  in das Kamerakoordinatensystem umgewandelt werden. Der Tiefenanteil, also die Z-Koordinate, wird hierbei noch nicht berechnet, sondern erst einmal auf 1 gesetzt:

Da sich aus verschiedenen Gründen nicht immer zur Gänze darauf verlassen werden kann, dass die Segmentierung die Laserlinie in ihrer Gesamtheit erfasst wird (vgl. Abschnitt 4.1), kann es sein, dass eine verzerrte Version der Laserlinie berechnet wird, wenn lediglich das segmentierte Bild als Basis für  $L_{auto}$  herangezogen wird. Daraus wird der Prozess zweimal durchgeführt: Zuerst wird eine Annäherung  $L_{auto}'$  mit dem segmentierten Bild berechnet. Anschließend werden in jeder Pixelspalte der approximierten Linie die Rotwerte des unmaskierten Bildes in einem festgelegten Abstand oberhalb und unterhalb des aktuellen Reihenindex wieder „eingebettet“, wobei die Pixel außerhalb dieses Bereichs jedoch nach wie vor geschwärzt bleiben. Diese neue Segmentierung wird nun als Basis für die Berechnung von  $L_{auto}$  herangezogen, welche wie durch Gleichung 3.6 beschrieben bestimmt wird. Dies führt zu einer genaueren Annäherung an die Optimallinie und einem generell „weicheren“ Verlauf der extrahierten Pixellinie (siehe Abb. 3.6).  $L_{auto}$  stellt die Bildkoordinaten der projizierten Laser-

$$\vec{p_{Kamera}} = K * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.8)$$

Sei nun  $T$  und  $R$  die Kameratranslation und -rotation aus Abschnitt 3.2.4. Dann kann eine Gerade  $g$  definiert werden, die durch den Ortsvektor  $\vec{o_g}$  als das Zentrum der Kamera in Weltkoordinaten und als Richtungsvektor  $\vec{d_g}$  den im Weltkoordinatensystem ausgedrückten Vektor  $\vec{p_{Kamera}}$  gegeben ist:

$$\vec{o_g} = R^{-1} * T \quad (3.9)$$

$$\vec{d_g} = R^{-1} * \vec{p_{Kamera}} \quad (3.10)$$

$$g = \vec{o_g} + \lambda * \vec{d_g} \quad (3.11)$$

Der gesuchte Punkt  $p_{Welt}$  liegt auf dieser Geraden. Um nun zu berechnen, wo auf der Geraden der Punkt liegt, kann der Schnittpunkt von  $g$  mit der Ebene  $e$ , die durch den Laser aufgespannt wird, berechnet werden. Der Ortsvektor  $\vec{o_e}$  von  $e$  ist durch den Aufbau des Scanners gegeben: Er befindet sich 120 mm in positiver Z-Richtung unter dem Zentrum der Kamera. Der Normalenvektor  $\vec{n_e}$  lässt sich ebenfalls aus der Scanner-Konstruktion bestimmen; dieser ist identisch mit der Z-Achse.

$$\vec{o_e} = \vec{o_g} + \begin{pmatrix} 0 \\ 0 \\ 120 \end{pmatrix} \quad (3.12)$$

$$\vec{n_e} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.13)$$

$$\vec{n_e} \cdot (\vec{x} - \vec{o_e}) = 0 \quad (3.14)$$

Der Schnittpunkt  $p_{Welt}$  ergibt sich dann aus:

$$\lambda = \frac{\vec{n_e} \cdot (\vec{o_e} - \vec{o_g})}{\vec{n_e} \cdot \vec{d_g}} \quad (3.15)$$

Durch das Einsetzen des errechneten  $\lambda$  in  $g$  kann nun  $p_{Welt}$  berechnet werden (siehe zu dieser Berechnung auch [Kow])

Durch die oben beschriebene Berechnung kann nun für jeden Punkt in  $L_{auto}$  eine Weltkoordinate berechnet werden. Am Ende dieses Prozesses muss nun noch der Z-Achsen-offset, der sich aus der Verschiebung des Scanner-Apparats zwischen Bildaufnahmen ergibt, auf die Koordinaten addiert werden. So wird für jede Aufnahme der Messreihe ein Satz von Weltkoordinaten errechnet, welcher zu einer Punktemenge hinzugefügt wird. Sind alle Aufnahmen verarbeitet, ist der Scavorgang abgeschlossen und das zu vermessene Objekt wurde erfolgreich abgebildet. Die resultierende Punktwolke steht dann zur weiteren Verarbeitung zur Verfügung.

# 4

---

## Ergebnisse

Jedes Messverfahren muss sich vor allem durch seine Genauigkeit behaupten. Im folgenden Kapitel werden in dieser Hinsicht die Ergebnisse vorgestellt, die das in Kapitel 3 implementierte Verfahren erbracht hat. Dabei wird auch darauf eingegangen, warum die erläuterten Ansätze verfolgt wurden, welche Schwierigkeiten sich ergeben und inwiefern diese sich auf die Messergebnisse ausgewirkt haben. Für die folgenden Erklärungen wurde die ebenfalls vorliegende exemplarische Messreihe einer im Raum liegenden schiefen Ebene verwendet, welche sich aufgrund ihrer geometrisch einfachen Form leicht beschreiben und als Validierungsobjekt verwenden lässt.

### 4.1 Laserlinienbestimmung

Die Lokalisierung der Laserlinie bestimmt in Bildkoordinaten, welche Teile des Bildes für die Messung interessant sind. Jegliche Ungenauigkeiten, die in diesem Schritt auftreten, propagieren sich also unweigerlich in den Folgeschritten zu immer größeren Messfehlern. Eine sorgfältige Bildverarbeitung ist hier also von hoher Bedeutung.

Wie in Abschnitt 3.2.3 und 3.2.5 beschrieben, wird für die Segmentierung der Laserlinie auf einen globalen Farbgrenzwert gesetzt. Dieser Ansatz erweist zwar eine geringe Komplexität, ist aber nicht ohne Nachteile. In Abb. 4.1 sind exemplarisch die Farbkanalwerte für die Pixel einer einzelnen Bildspalte aus einem ausgewählten Bild (welches in Abb. 3.6 ganz oben zu sehen ist) sowohl im RGB- als auch im YCbCr-Farbraum dargestellt. Dabei befinden sich auf der X-Achse die Reihenindizes und auf der Y-Achse die Kanalwerte im Bereich von 0 bis 255. Wie man der Abbildung erkennen kann, befindet sich die Laserlinie im X-Achsenabschnitt zwischen ca. 370 bis 380, erkennbar durch den hohen Ausschlag im Rot-Kanal. Möchte man nun einen globalen Farbgrenzwert ansetzen um die Laserlinie in ihrer gesamten Breite von ca. 10 Pixel zu erfassen und dabei alle anderen Bildanteile auszublenden, müssten alle Pixel mit einem Rotkanalwert von ca. unter 50 ausgeblendet (also geschwärzt) werden. Dabei bleiben aber noch die Pixel übrig, die sowohl einen hohen Rot- als auch einen hohen Blau und Grün-Anteil aufweisen und somit ebenfalls nicht zur Laserlinie gehören.

Es würde sich nun also anbieten, für den Grün- und Blau-Kanal einen Obergrenzwert festzulegen, der zusätzlich alle Pixel schwärzt, deren Grün- und Blau-Anteile über diesen Grenzwerten liegen. Wie aus Abb. 4.1 allerdings hervorgeht, ist es nicht möglich, solche Grenzwerte zu finden und gleichzeitig alle Pixel der Laserlinie zu erhalten. Vor allem im X-Achsenabschnitt von 0 bis 150 sind die Grünkanalwerte nicht hoch genug um durch die Obergrenze geschwärzt zu werden und gleichzeitig die Pixel im X-Achsenabschnitt von 370 bis 380 intakt zu lassen (ähnliches gilt für die Blau-Werte in anderen Teilen des Bildes). Eine Alternative bestünde darin, den Rotkanal-Untergrenzwert zu erhöhen, um die hohen Rotkanalwerte im X-Achsenabschnitt von 0 bis 150 auszublenden, was allerdings in einer generellen Schmälerung der Laserliniensegmentierung resultiert. In der vorliegenden Implementierung wurde nun anstatt eines RGB- ein YCbCr-Grenzwert verwendet. Dieser ist ebenfalls in Abb. 4.1 abgebildet. Das geschilderte Problem lässt sich zwar auch in diesem Farbraum nicht gänzlich vermeiden, jedoch besser kontrollieren. Wie in der Abbildung zu sehen ist, kann mit einem Untergrenzwert von ca. 150 die Laserlinie in einem Großteil ihrer Breite erfasst werden ohne dabei andere Farbkanäle in Betracht ziehen zu müssen. Indem nur ein Grenzwert angepasst werden muss eignet sich dieser Farbraum somit auch zur besseren automatischen Grenzwertfindung, welche sicherstellen soll, dass die Schmälerung der Laserliniensegmentierung minimal ausfällt.

Ohne eine lokale Betrachtung der Bilddaten lässt sich mit einem globalen Ansatz die Segmentierung der Laserlinie also nicht zur Gänze beseitigen. Daher wird in der vorliegenden Implementierung wie in 3.2.5 beschrieben darauf gesetzt, nach einer groben Lokalisierung auf Basis der Segmentierung eine feinere Bestimmung auf Basis der umliegenden unsegmentierten Pixeldaten vorzunehmen. So wird der Nachteil eines globalen Farbgrenzwertes abgeschwächt ohne zu viel Komplexität hinzuzufügen. Die Ergebnisse der Laserlinienbestimmung bewegen sich dank dieses Ansatzes in einem akzeptablen Fehlerraum. In 3.6 ist eine mit dem Verfahren erkannte Laserlinie zu sehen, welche neben einem vom Benutzer vorgenommenen Goldstandard und vor dem Hintergrund des betreffenden Bildes dargestellt ist. Der Cr-Grenzwert der in diesem Beispiel gewählt wurde ist durch dass Verfahren ermittelt worden, welches in 3.2.3 beschrieben wurde. Die Distanz zwischen der optimalen und der ermittelten Linie beträgt 5.0443 Pixel.

## 4.2 Messgenauigkeit

Die Genauigkeit der Berechnung der Weltkoordinaten bestimmt wie erfolgreich das Messverfahren in seiner Gesamtheit funktioniert, da diese das angestrebte Endprodukt der Messung darstellen. Um diese Genauigkeit zu bestimmen, wurde für die vorliegende Implementierung die zu Eingang des Kapitels erwähnte schiefe Fläche zweimal vermessen: Einmal händisch um nötige Daten für die mathematische Beschreibung der Schrägen zu sammeln, und einmal mit dem Lichtschnittverfahren um diese Messung mit der händischen Messung zu vergleichen.

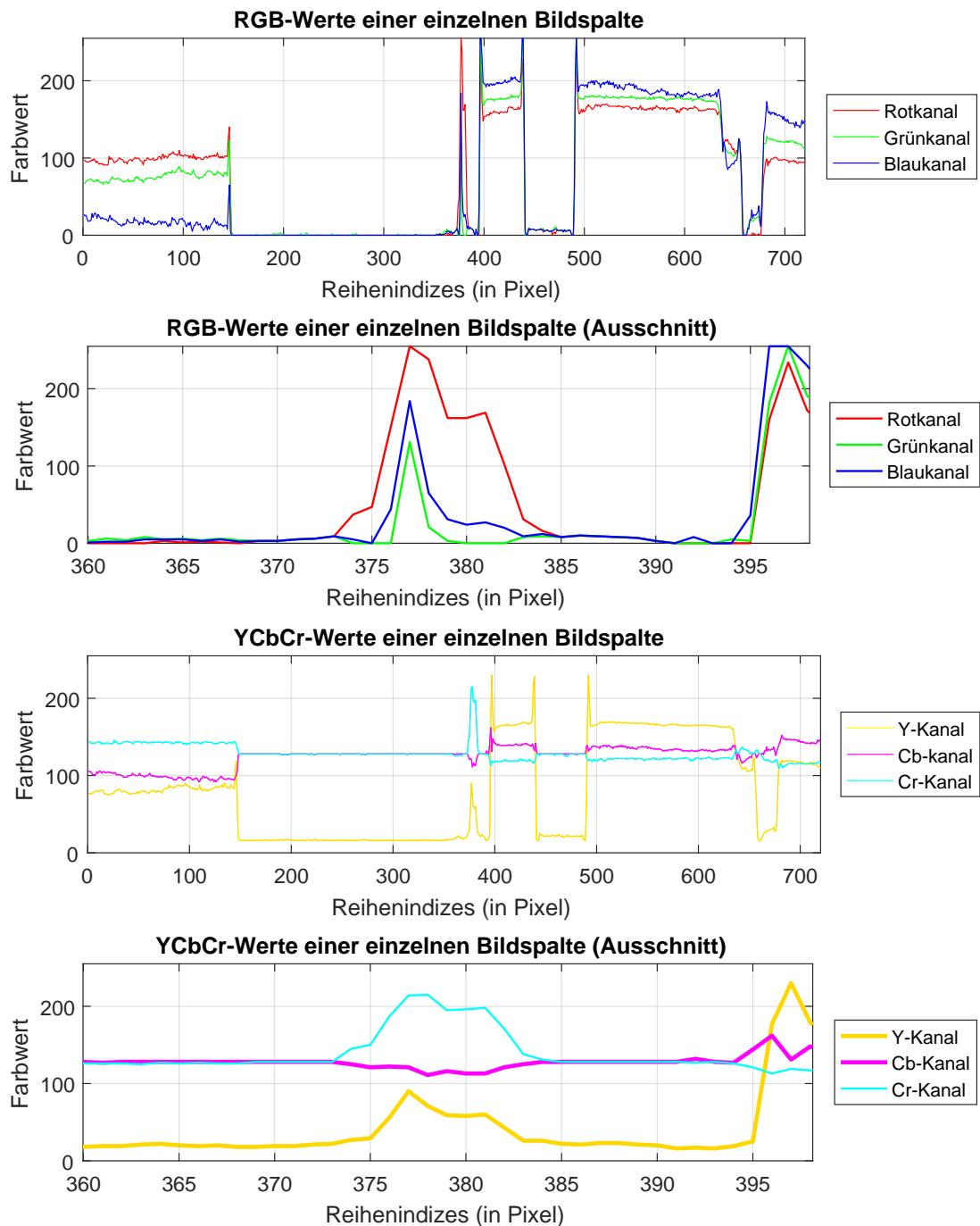
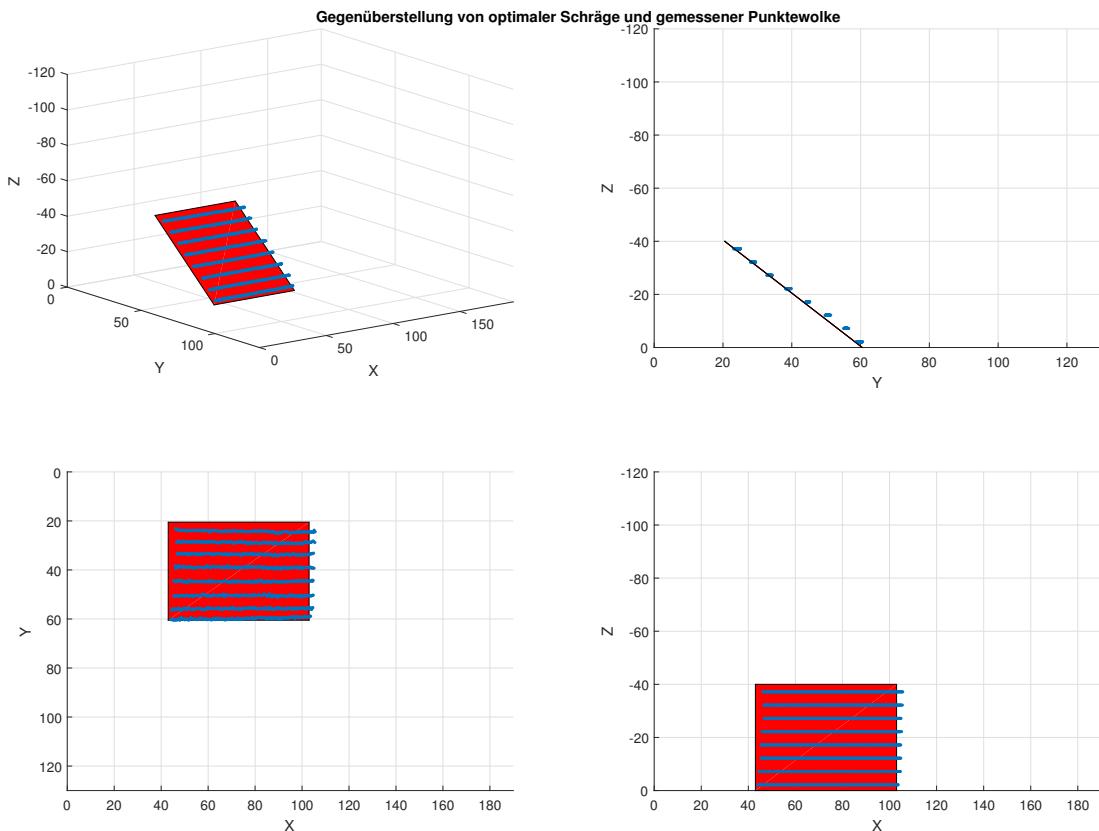


Abb. 4.1. Farbkanäle einer einzelnen Bildspalte

Durch die händische Messung lässt sich die Schräge mathematisch vergleichsweise einfach als eine durch vier Seiten begrenzte Ebene beschreiben und eignet sich daher zur Berechnung einer Metrik, mit der die Exaktheit des Verfahrens bewertet werden kann. Dafür wird für jede Weltkoordinate der kürzeste Abstand zur Ebene berechnet. Es wird immer der kürzeste Abstand zum jeweils nächsten Merkmal der Schräge ermittelt, also entweder zur Ebene an sich, zu einer der vier Seiten oder einer der vier Eckpunkte der Fläche. Anschließend wird das arithmetische Mittel dieser Abstände gebildet, um zu ermitteln, wie weit eine Koordinate im Durchschnitt von der eigentlichen Fläche entfernt ist. In 4.2 ist die gemessene Punktewolke der Schräge entgegengestellt. Gemessen wurden hier 2168 Punkte, welche im Durchschnitt 0.9083 Millimeter mit einer Standardabweichung  $\sigma = 0.5360$  von der optimalen Schräge entfernt liegen.



**Abb. 4.2.** Gegenüberstellung von optimaler Schräge und gemessener Punktewolke

Es ist auf der X-Y-Ebene zu sehen, dass die einzelnen vermessenen Punkte-Linien eine Schiefe aufweisen und nicht im perfekten rechten Winkel zur X-Achse stehen. Dies ist der Tatsache geschuldet, dass der Laser in der Bildaufnahmen nicht perfekt parallel zur Bodenebene steht. Als Resultat neigt sich auch auf den Bildern die Laserlinie mit einer minimalen Steigung, was den oben zu sehenden Effekt hervorbringt. Dies lässt sich vor allem mit erhöhter Sorgfalt während

des Vermessens mildern, zeigt jedoch, dass Außeneinflüsse im direkten Vergleich zu mathematisch optimalen Konstrukten selten zur Gänze zu vermeiden sind. Ein Problem des Verfahrens liegt ebenfalls in der Gefahr, dass die mathematische Beschreibung des Validierungsobjektes das Objekt idealer beschreibt als in der Realität abgebildet. Somit kann die Messung zu einem gewissen Grad niemals an die mathematische Beschreibung heran reichen und der Messung wird auf diese Weise „Unrecht getan“; sie wird also schlechter eingeschätzt als sie eigentlich ist. Dennoch kann bei genügender Sorgfalt der händischen Vermessung des Objektes damit gerechnet werden, dass die Metrik aussagekräftig ist. Auch wenn die verwendete Metrik nur für das simple Validierungsobjekte sprechen kann, wird davon ausgegangen dass sich das implementierte Verfahren für komplexere Objekte ähnlich genau verhält.

# **5**

---

## **Fazit**

TODO

---

## Literaturverzeichnis

- Kow. KOWDLE, ADARSH: *Back projection - 2D points to 3D.*  
[http://chenlab.ece.cornell.edu/people/adarsh/publications/  
BackProjection.pdf](http://chenlab.ece.cornell.edu/people/adarsh/publications/BackProjection.pdf).
- Mat. *Single Camera Calibration App.*  
[https://de.mathworks.com/help/vision/ug/single-camera-  
calibrator-app.html](https://de.mathworks.com/help/vision/ug/single-camera-calibrator-app.html).
- Sta. *Stack Overflow - How is the reprojection error calculated in Matlab's triangulate function?*  
[http://stackoverflow.com/questions/29095349/how-is-the-  
reprojection-error-calculated-in-matlabs-triangulate-function-  
sadly](http://stackoverflow.com/questions/29095349/how-is-the-reprojection-error-calculated-in-matlabs-triangulate-function-sadly).