



FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis
in
Robotics, Cognition, Intelligence

**Stereo-Vision mit Zeilenkameras in der
industriellen Bildverarbeitung**

Marcel Sauer



FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis
in
Robotics, Cognition, Intelligence

**Stereo-Vision mit Zeilenkameras in der
industriellen Bildverarbeitung**

**Stereo Vision Using Line-Scan Cameras in
Industrial Applications**

Bearbeiter: Marcel Sauer
Aufgabensteller: Prof. Dr. Carsten Steger
Betreuer: Prof. Dr. Carsten Steger
Abgabedatum: 16. Dezember 2013

Ich versichere, dass ich diese Master's Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Garching, den 16. Dezember 2013

Marcel Sauer

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Nomenklatur	v
1. Einleitung	1
2. Aufgabenstellung	2
3. Stand der Technik	3
4. Allgemeine theoretische Grundlagen	5
4.1. Stereo-Vision	5
4.2. Zeilenkameras	7
4.3. Starre Abbildungen	9
4.3.1. Pose	9
4.3.2. Homogene Matrix	10
4.4. Levenberg-Marquardt-Algorithmus	13
5. Problemspezifische theoretische Grundlagen	15
5.1. Einschränkungen im Kontext dieser Arbeit	15
5.2. Kameramodell	17
5.2.1. Kameraparameter	17
5.2.2. Unbestimmtheiten	20
5.2.3. Abbildung eines Punktes zu einer Bildkoordinate	22
5.2.4. Abbildung einer Bildkoordinate zu einem Strahl	23
5.3. Stereo-Aufbau	23
5.3.1. Parameter	24
5.3.2. Gesamtes Modell	24
5.4. N-Kamera-Setup	26
5.4.1. Anpassung der Parameter	26
5.4.2. Anpassung des Modells	27
6. Implementierung der Kalibrierung	28
6.1. Vorbereiten der Kalibrierung	28
6.2. Aufbau der grundsätzlichen Kalibrierung	29
6.3. Ausschluss von Parametern	30
6.4. Vermeidung von lokalen Minima	30
6.5. Stabilität und Verringerung der Komplexität	32
6.6. Fehlerkorrektur der Mittelpunktextraktion	33
6.7. Erweiterung auf N Kameras	34
7. Ansatz zur Rekonstruktion	35
7.1. Matching anhand von angenäherten rektifizierten Bildern	35

7.2. Matching anhand von Epipolarkurven	37
8. Ergebnisse	39
8.1. Vorgehen	39
8.2. Simulierte Daten	40
8.2.1. Kalibrierung	41
8.2.2. Betrachtung der Längenabweichung	42
8.2.3. Rekonstruktion	42
8.3. Reale Daten	46
8.3.1. Versuchsaufbau	46
8.3.2. Kalibrierung	46
8.3.3. Betrachtung der Längenabweichung	49
8.3.4. Rekonstruktion	52
8.4. Modifikation des Levenberg-Marquardt-Verfahrens	57
9. Ausblick	61
10. Fazit	62
A. Anhang	63
A.1. Homogene Elementartransformationen	63
A.2. Invertierung einer homogenen Matrix	64
A.3. Methoden zum Vergleichen von Bildausschnitten	65
Literatur	66

Abbildungsverzeichnis

1.	Along-Track- und Across-Track-Konfigurationen (Fernerkundung)	3
2.	Abbildung eines Pixels auf einen Strahl	5
3.	Epipolarkurve bei Flächenkameras	6
4.	Rektifizierung von Bildern	6
5.	Koordinatensystem und Trajektorie einer idealen Zeilenkamera	8
6.	Bild einer Kalibrierplatte	15
7.	Ideale Along-Track- und Across-Track-Konfigurationen	16
8.	Parameter einer Zeilenkamera	18
9.	Simulation mit Beispielparametern	19
10.	Einfluss von C_x	19
11.	Einfluss von κ	20
12.	Einfluss von $S_y C_y$	21
13.	Einfluss von f	21
14.	Einfluss von S_x	21
15.	Einfluss des Bewegungsvektors	22
16.	Aufbau mit zwei Kameras	24
17.	Aufbau mit drei Kameras	26
18.	Aufnahme und Simulation einer Kalibrierplatte	33
19.	Tiefenrekonstruktion in einer Along-Track-Konfiguration	36
20.	Matching auf Basis allgemeiner Epipolarkurven	38
21.	Definition der Abstände zur Berechnung der Längenabweichung	40
22.	Längenabweichung des Testfalls SIMALONG, Datensatz 1	43
23.	Simulierte Bilder des Testfalls SIMALONG	43
24.	Ausschnitte der Korrespondzsuche bei Testfall SIMALONG	44
25.	Verlauf der normalisierten Kreuzkorrelation in Testfall SIMALONG	44
26.	Rekonstruierte Tiefe der Kalibrierplatte als Graustufen des Testfalls SIMALONG	45
27.	Rekonstruierte Fläche in 3D: Testfall SIMALONG	45
28.	Rekonstruierte Fläche in 2D: Testfall SIMALONG	46
29.	Versuchsaufbau	47
30.	Aufnahme einer Kalibrierplatte in Testfall TEST7	49
31.	Fehler bei der Längenrekonstruktion: Testfall TEST1, Datensatz 1	51
32.	Fehler bei der Längenrekonstruktion: Testfall TEST7, Datensatz 3, lokales Minimum	52
33.	Aufnahme einer Platine (Testfall TEST6)	53
34.	Rekonstruierte Tiefe einer Platine als Graustufen (Testfall TEST6)	54
35.	Seitliche 2D-Ansicht einer rekonstruierten Platine (Testfall TEST6)	54
36.	Aufnahme einer flachen Korkplatte (Testfall TEST7)	55
37.	Aufnahme einer gebogenen Korkplatte (Testfall TEST7)	55
38.	Tiefe der Korkplatten als Graustufen (Testfall TEST7)	56
39.	Rekonstruierte Oberflächen der Korkplatten aus Testfall TEST7	56
40.	Konvergenzverhalten in Testfall TEST1	57

41.	Verhalten von λ in Testfall TEST1	58
42.	Konvergenzverhalten in Testfall TEST3	59
43.	Konvergenzverhalten in Testfall TEST2	60
44.	Konvergenzverhalten in Testfall TEST3, abweichende Implementierung von Methode 2	60

Nomenklatur

Koordinatensysteme

$s, s1, s2$	allgemeine Koordinatensysteme
$c, c1, c2, cn$	Kamerakoordinatensysteme
$o, o1, o2, om$	Koordinatensysteme von Kalibrierkörpern

3D-Raum

${}^s p$	Punkt in homogenen Koordinaten definiert im Koordinatensystem s
${}^s v$	Vektor in homogenen Koordinaten definiert im Koordinatensystem s
${}^{s2} {}_{s1} P$	Pose, Beschreibung einer starren Abbildung von $s1$ nach $s2$
${}^{s2} {}_{s1} H$	Homogene Transformationsmatrix, transformiert Punkte und Vektoren von $s1$ nach $s2$ (${}^{s2} p = {}^{s2} {}_{s1} H {}^{s1} p$)
p_x, p_y, p_z	x -, y - und z -Koordinate eines Punktes
v_x, v_y, v_z	x -, y - und z -Koordinate eines Vektors
${}^s x, {}^s y, {}^s z$	x -, y - und z -Richtung des Koordinatensystems s

Bildkoordinaten

${}^{p1} r, {}^{p2} r,$	Zeile im Bild der ersten bzw. der zweiten Kamera
${}^{p1} c, {}^{p2} c,$	Spalte im Bild der ersten bzw. der zweiten Kamera

1. Einleitung

Stereo-Vision bezeichnet eine Vorgehensweise, um 3D-Daten aus zwei zweidimensionalen Bildern zu extrahieren. Üblich ist hierfür die Verwendung von Kameras, die durch ein Lochkameramodell beschrieben werden.

Diese Arbeit befasst sich mit Stereo-Vision unter Verwendung von Zeilenkameras. Eine Zeilenkamera nimmt zu einem Zeitschritt nur eine Zeile auf [SRT⁺¹¹]. Für die Erzeugung eines zweidimensionalen Bildes wird eine Bewegung vorausgesetzt. Das Bild entsteht durch das Aneinandersetzen von zu verschiedenen Zeitpunkten aufgenommenen Bildzeilen.

Zeilenkameras können eine deutlich höhere Auflösung einer Zeile erreichen als herkömmliche Kameras. Ein aktuelles Beispiel ist die raL12288-66km von Basler. Diese löst eine Zeile mit über 12 000 Pixeln auf und hat eine Aufnahmerate von bis zu 66 000 Zeilen pro Sekunde. Zudem sind Zeilenkameras günstiger als hochauflösende Flächenkameras. [Bas13]

Die Motivation, Zeilenkameras anstelle von Flächenkameras zu verwenden, ist die Aussicht, eine genauere Rekonstruktion bei günstigerer Hardware zu erreichen.

Der Fokus dieser Arbeit ist dabei die Anwendung in einem industriellen Umfeld. Für ein solches wird hier angenommen, dass die vorherrschende Bewegung linear ist und die relative Pose zwischen den Zeilenkameras konstant bleibt.

Damit unterscheidet sich das Problem von bisherigen Anwendungen von Stereo-Zeilenkameras. Diese werden hauptsächlich an Satelliten und Flugzeugen montiert. In diesen Anwendungsfällen ändert sich die äußere Orientierung und Position des Sensors zu jedem Zeitschritt. Nur in Spezialfällen bleibt die relative Pose zwischen den Kameras konstant. Dagegen werden hier Fälle betrachtet, welche in einem industriellen Umfeld auftreten können.

Eine Bewegung zwischen den Kameras und dem Objekt entsteht, da entweder das zu rekonstruierende Objekt oder die Kameras geführt werden. Dabei kann das Objekt beispielsweise auf einem Förderband liegen. Ebenso ist es denkbar, die Zeilenkameras an einem Schlitten zu montieren und diesen über dem Objekt zu bewegen.

Die Anwendungsfälle sind beispielsweise die Fehlererkennung in Produkten, ein Objekt-matching anhand von rekonstruierten 3D-Daten und die Ermittlung von Objektposen für weitere Aufgaben.

2. Aufgabenstellung

In dieser Arbeit soll die Nutzung von Stereo-Vision mit Zeilenkameras untersucht werden. Dabei soll sich die Arbeit auf das industrielle Umfeld beschränken. Die Annahme dazu ist die Existenz einer linearen Bewegung des zu rekonstruierenden Objektes relativ zu den Kameras. Weiterhin bleibt die relative Pose zwischen den Kameras konstant.

Hierfür muss ein Modell gefunden werden, welches einen solchen Stereo-Aufbau beschreibt. In diesem Modell werden die Kameramodelle (Abbildung von einem Punkt im 3D-Raum zu einer Bildkoordinate), die relevanten Posen und der Bewegungsvektor vereint. Bei den Kameramodellen ist darauf zu achten, dass sie reale Kameras möglichst gut beschreiben. Aus diesem Grund sollen die Kameramodelle Verzeichnungen und Fehler in der Position des Hauptpunktes¹ beinhalten.

Ausgehend von dem Modell des Stereo-Aufbaus müssen die inneren Parameter der Kameras und die äußeren Parameter der Konfiguration bestimmt werden. Dieser Vorgang wird Kalibrierung genannt und soll den Schwerpunkt der Arbeit bilden.

Ein weiteres Interesse betrifft die Rekonstruktion. Hier soll ermittelt werden, wie eine Rekonstruktion durchgeführt werden kann, und in welchem Maße man auf bekannte Vorgehensweisen aus Stereo-Vision mit Flächenkameras zurückgreifen kann.

Für das Erreichen der Ziele wird diese Arbeit durch MVTEC unterstützt. MVTEC ist ein weltweit operierendes Unternehmen, spezialisiert auf Bildverarbeitung jeder Art.

Eines der Produkte von MVTEC ist HALCON. HALCON bietet ein Framework für Bildverarbeitungsanwendungen mit einer umfangreichen Sammlung an Operatoren und Bildeinzugsschnittstellen. Diese können über Sprachanbindungen mit verschiedenen Programmiersprachen verwendet werden. Zusätzlich enthält HALCON mit HDevelop eine IDE². Durch die Verwendung einer dynamischen Skriptsprache können in dieser Bildverarbeitungslösungen effektiv entwickelt werden.

Neben einer Lizenz von HALCON wurde auch die Hardware und ein Laborplatz für die Versuchsaufnahmen zur Verfügung gestellt.

¹Schnittpunkt der optischen Achse des Objektivs mit der Sensorebene.

²Integrierte Entwicklungsumgebung (engl. integrated development environment).

3. Stand der Technik

Zeilenkameras werden bereits zur Stereo-Rekonstruktion genutzt. Dabei beschränkt sich die Nutzung größtenteils auf Anwendungsfälle in der Fernerkundung. Für die Vermessung der Erdoberfläche werden Zeilenkameras an Satelliten und Flugzeugen montiert ([Whi90], [FR05]).

Aus diesem Kontext stammen auch die Terme Along-Track und Across-Track. In einer Along-Track-Konfiguration sind zwei Zeilenkameras an einem Flugzeug oder Satelliten befestigt. Diese nehmen die Landschaft quer zur Flugrichtung auf. Dabei ist eine Kamera nach vorne und eine Kamera nach hinten geneigt. Abbildung 1a zeigt eine solche Konfiguration.

Bei einer Across-Track-Konfiguration werden zwei näherungsweise parallele Bahnen benötigt. Auf beiden Bahnen wird die Landschaft quer zur Bewegungsrichtung aufgenommen. Solch eine Konfiguration ist in Abbildung 1b gezeigt.

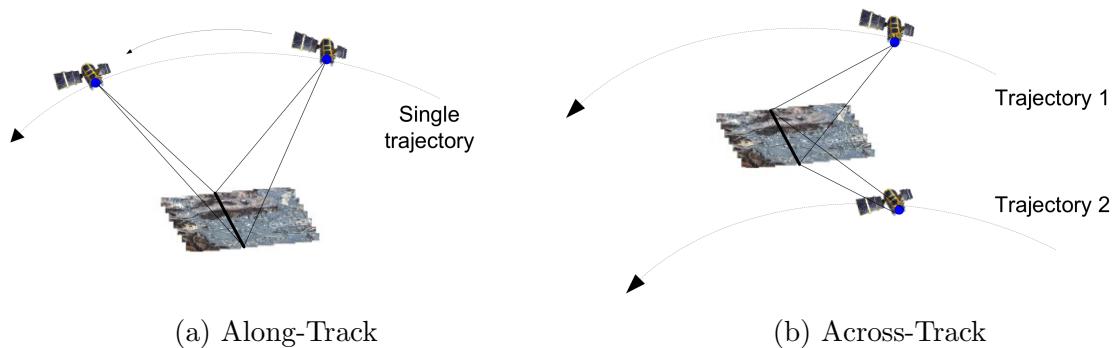


Abbildung 1: Along-Track- und Across-Track-Konfigurationen in der Fernerkundung (aus [Oh11]).

In [HSH05] wird ein Verfahren vorgestellt, um anhand von mehreren Zeilenkameras in einer Along-Track-Konfiguration 3D-Daten zu rekonstruieren. Dabei werden auch Verzeichnungen und Fehler in der Position des Hauptpunktes betrachtet. Mit dem Kontext dieser Arbeit ist aber die Annahme einer näherungsweisen Along-Track-Konfiguration nicht vereinbar.

Ein Ansatz zur näherungsweisen Rektifizierung von Bildern verschiedener idealer Zeilenkameras wird in [WHL11] gezeigt. Dies ist allerdings nur gültig, wenn sich die Zeilenkameras linear bewegen und in einer Along-Track- oder einer Across-Track-Konfiguration befinden. Weiterhin werden die Epipolarkurven³ durch Geraden angenähert. Dies könnte bei einer Rekonstruktion zu Problemen führen.

Einen anderen Ansatz zur näherungsweisen Rektifizierung zeigt [MKJH06]. Hierbei wird die Kamera durch eine Parallelprojektion beschrieben. Dies ist allerdings nur vertretbar, wenn der Öffnungswinkel der Kamera sehr klein ist. Daher sind bei einer industriellen Anwendung zu große Fehler durch das unpassende Kameramodell zu erwarten.

³Abbildung einer Bildkoordinate eines Bildes auf mögliche Bildkoordinaten des zweiten Bildes.

In [GH97] wird vorgestellt, wie eine lineare Bewegung orbitale Bewegungen von Satelliten annähern kann, so dass mit Hilfe eines weniger komplexen Modells die Kameras kalibriert und Tiefen rekonstruiert werden können. Viele Aspekte dieser Veröffentlichung lassen sich auf diese Arbeit übertragen. Allerdings werden für die Kameras ideale Modelle ohne Verzeichnung verwendet.

Wie [DRS11] herausstellt, besteht unter bestimmten Bedingungen die Möglichkeit, mit einer linearen Methode eine Zeilenkamera mit planaren Körpern zu kalibrieren. Hierfür muss eine lineare Bewegung vorhanden sein. Weiterhin dürfen keine Verzeichnungen vorliegen und der Bewegungsvektor muss orthogonal zu der vom Sensor erfassten Ebene ausgerichtet sein.

Im Kontext dieser Arbeit soll ein Modell einer Zeilenkamera verwendet werden, welches sowohl Verzeichnungen als auch Fehler in der Position des Hauptpunktes betrachtet. Weiterhin soll die äußere Konfiguration nicht eingeschränkt werden.

4. Allgemeine theoretische Grundlagen

Für das Verständnis der weiteren Arbeit wird neben algebraischen Grundkenntnissen auf den folgenden Grundlagen aufgebaut.

4.1. Stereo-Vision

Stereo-Vision beschreibt die Rekonstruktion von 3D-Daten aus Bildern zweier Kameras. Zwischen den Kameras besteht eine starre Transformation.

Sind alle Parameter eines Aufbaus bekannt, ist es möglich die Koordinaten eines Punktes p , welcher sich auf den Bildkoordinaten $(^{p1}r, ^{p1}c)$ und $(^{p2}r, ^{p2}c)$ abbildet, zu rekonstruieren. Einer Bildkoordinate kann ein Strahl im 3D-Raum zugeordnet werden. Der Ursprung des Strahls ist ein Sensorelement des Sensors der Kamera. Weiterhin durchdringt jeder Strahl das Projektionszentrum. Als Entfernung auf dem Strahl wird hier das Erreichen einer bestimmten z -Koordinate im Kamerakoordinatensystem angesehen. Abbildung 2 zeigt einen Strahl und die Definition des Kamerakoordinatensystems.

Aus den Strahlen zu den Bildkoordinaten beider Bilder und der Verbindungsgeraden b zwischen den Projektionszentren entsteht ein Dreieck. Der Schnittpunkt der Strahlen bildet einen Eckpunkt dieses Dreiecks. An diesem Eckpunkt befindet sich der Punkt p .

Betrachtet man zu dem Strahl einer Bildkoordinate $(^{p1}r', ^{p1}c')$ einen Punkt p' in der Entfernung ^{c1}z , kann die Projektion des Punktes p' auf $(^{p2}r', ^{p2}c')$ vorhergesagt werden. Durch das Variieren der Distanz ^{c1}z von 0 bis ∞ entsteht eine Linie im Bild der zweiten Kamera. Diese wird Epipolarkurve genannt. In Abbildung 3 wird die Erzeugung einer Epipolarkurve gezeigt.

Die Epipolarbedingung sagt aus, dass sich der Punkt p' in dem Bild der zweiten Kamera irgendwo auf der Epipolarkurve abbilden muss. Aus der Position $(^{p2}r', ^{p2}c')$ auf der Epipolarkurve kann auf die Tiefe ^{c1}z geschlossen werden.

Für eine Rekonstruktion muss die Koordinate $(^{p2}r', ^{p2}c')$ gefunden werden. Durch die Epipolarbedingung wird die allgemeine 2D-Suche auf eine 1D-Suche reduziert.

In realen Anwendungen wird dazu ein Bildausschnitt des ersten Bildes um eine Koordinate $(^{p1}r', ^{p1}c')$ mit Ausschnitten aus dem zweiten Kamerabild verglichen. Diese werden entlang der Epipolarkurve gebildet.

Die Verwendung von Stereo-Vision zur 3D-Rekonstruktion mit Flächenkameras ist umfassend erforscht. Ein Standardwerk hierzu ist [HZ04].

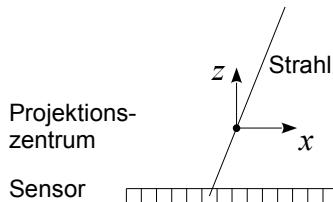


Abbildung 2: Abbildung eines Pixels auf einen Strahl.

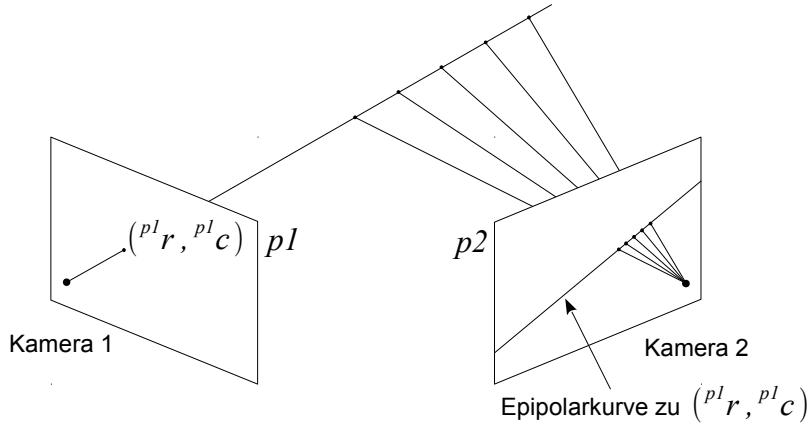


Abbildung 3: Die Epipolarkurve einer Flächenkamera. Große Punkte stellen die Projektionszentren der Kameras dar. p_1 und p_2 beschreiben die virtuellen Bildebenen der Kameras.

Bei der Nutzung von Flächenkameras lässt sich die Rekonstruktion vereinfachen. Wie in [HZ04] beschrieben, bilden sich Epipolarkurven in Bildern von idealen Lochkameras auf Geraden ab. Dies ist auch in Abbildung 3 sichtbar.

Bilder realer Flächenkameras können entzerrt werden, so dass diese dem Lochkamera-modell entsprechen.

Sind zudem die Ebenen, welche in einer (entzerrten) Bildzeile abgebildet werden, und die Verbindungsgeraden zwischen den Projektionszentren koplanar, bilden sich alle Epipolarkurven in einer Zeile auf eine Zeile des Bildes der anderen Kamera ab. Diese Eigenschaft kann allgemein hergestellt werden, indem die Kamerabilder geeignet transformiert werden. Dieser Schritt wird Rektifizierung genannt. Abbildung 4 zeigt ein Bildpaar vor und nach der Rektifizierung.

Elementar für dieses Vorgehen ist die genaue Bestimmung der Parameter der Kameramodelle und der relativen Pose zwischen den Kameras. Zu diesem Thema existieren viele Veröffentlichungen. Ausführlich wird dies in [SUW08] unter Betrachtung von Verzeichnungen und Fehlern in der Position des Hauptpunktes beschrieben. Weiterhin stellt [FP08] eine Möglichkeit vor, Flächenkameras in einem Stereo-Aufbau zu kalibrieren.

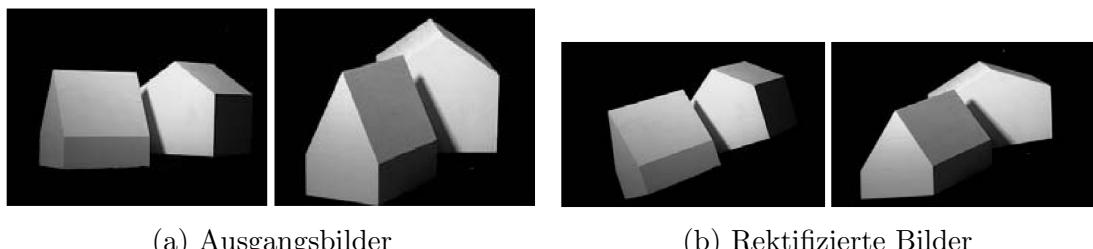


Abbildung 4: Rektifizierung eines simulierten Bildpaars (aus [HZ04]).

[Hei00] behandelt die Kalibrierung einzelner verzeichnungsbehafteter Flächenkameras mit kreisförmigen Kontrollmarken. Führt man diese mit zwei Kameras gleichzeitig durch, kann auch der ganze Stereo-Aufbau kalibriert werden.

4.2. Zeilenkameras

Zeilenkameras (engl. line scan cameras oder pushbroom cameras) sind Kameras, bei denen der Sensor nur aus einem 1D-Array besteht. Demnach ist es zu einem Zeitpunkt nur möglich, eine einzige Bildzeile aufzunehmen. Allerdings ist die mögliche Auflösung einer Bildzeile bei Zeilenkameras deutlich höher als bei Flächenkameras. Die Zeilenrate⁴ ist bei dieser Bauart variabel gehalten.

Bei der Existenz einer Bewegung, sei es die Kamera, ein Objekt im Sichtfeld der Kamera oder auch beides, ist es möglich durch das Aneinandersetzen von Zeilen ein 2D-Bild zu erzeugen. Dabei hängt die Auflösung in Bewegungsrichtung ausschließlich von der Zeilenrate ab.

Um verwertbare Bilder mit hohen Zeilenrate zu erzeugen, ist es notwendig, dass ausreichend Licht auf das Sensorarray trifft. Dies ist durch eine zusätzliche Beleuchtung oder eine Öffnung der Blende des Objektives möglich. Da eine Öffnung der Blende allerdings die Schärfentiefe⁵ negativ beeinflusst, sollte bei der Verwendung von Zeilenkameras auf eine ausreichende Lichtzufuhr geachtet werden.

Für die Beschreibung des Modells einer idealen Zeilenkamera definieren wir in dem Projektionszentrum der Kamera ein Koordinatensystem. Dieses ist so ausgerichtet, dass die x -Achse der Ausrichtung des Sensorarrays entspricht. Die z -Achse zeigt in Blickrichtung der Kamera. Die verbleibende y -Achse wird so definiert, dass das finale System ein rechtshändiges Orthogonalsystem bildet. Da x und somit auch y vom Vorzeichen her nicht festgelegt sind, wird definiert, dass die Bewegung zum Zeitpunkt $t = 0$ eine positive y -Komponente besitzen soll. Abbildung 5 zeigt die Definition des Koordinatensystems. Die Bewegung wird durch eine Trajektorie⁶ beschrieben. Dabei wird angenommen, dass sich nur die Kamera bewegt. Das Kamerakoordinatensystem liegt in dem Projektionszentrum zum Zeitpunkt $t = 0$. In diesem ist die Trajektorie definiert. Mit einer Variation der Zeit wird die Position und die Ausrichtung der Kamera anhand einer Trajektorie modifiziert. Es entsteht ein mitlaufendes Koordinatensystem, welches zum Zeitpunkt $t = 0$ mit dem Kamerakoordinatensystem identisch ist.

Abbildung 5 zeigt die Definition der Koordinatensysteme und der Trajektorie.

⁴Die Anzahl aufgenommener Bildzeilen pro Sekunde, analog zu der Framerate bei Flächenkameras.

⁵Der Tiefenbereich, der scharf dargestellt werden kann.

⁶Allgemeine Abbildung von Zeit auf Position und Orientierung.

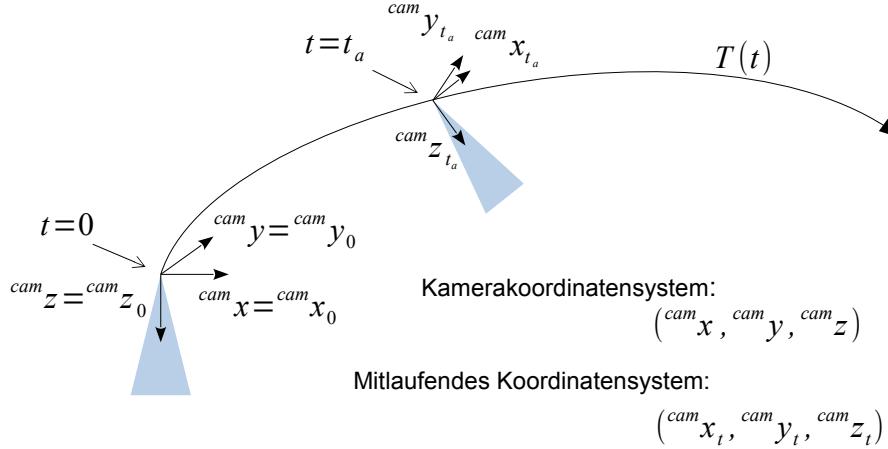


Abbildung 5: Koordinatensystem und Trajektorie einer idealen Zeilenkamera. Durch die Dreiecke wird der Sichtbereich der Kamera angedeutet.

Das komplette Modell lässt sich mit folgenden Gleichungen beschreiben:

$$m u = T_x(t, p) \quad (1)$$

$$0 = T_y(t, p) \quad (2)$$

$$m f = T_z(t, p) \quad (3)$$

$$r = t \text{ lps} \quad (4)$$

$$c = u/S_x + C_x \quad (5)$$

Dabei beschreibt die Trajektorie $T(t, p)$ die Transformation des Punktes p von dem Kamerakoordinatensystem in das mitlaufende Koordinatensystem zum Zeitpunkt t . Sollte die y -Koordinate des Punktes p in dem mitlaufenden Koordinatensystem ($T_y(t, p)$) gleich 0 sein, ist die Gleichung (2) erfüllt. Diese Zeitpunkte werden nachfolgend t^* genannt. Der Punkt bildet sich dann auf die Zeile r in Abhängigkeit von der Zeilenrate lps ab (Gleichung (4)). Dies beschreibt eine Parallelprojektion entlang der Trajektorie.

Abhängig von der Tiefe des Punktes $T_z(t^*, p)$ und der Brennweite f ergibt sich aus Gleichung (3) die normierte Distanz m . Durch m und $T_x(t^*, p)$ wird in Gleichung (1) die metrische x -Koordinate u , in welcher sich der Punkt p auf das Sensorarray abbildet, bestimmt. Das antiproportionale Verhalten von m und u bewirkt eine perspektivische Projektion. Gleichung (5) rechnet die metrische Größe u in eine Spalte c um. Hierbei geht der Abstand zweier benachbarter Sensorelemente S_x und die Position des Hauptpunktes auf dem Sensor C_x ein.

Die Form der Trajektorie wird durch den Anwendungsfall bestimmt. Bei der Fernerkundung mit Satelliten sind beispielsweise Orbitalmodelle und Annäherungen an Ellipsen üblich [GH95]. Die Bewegung von Flugzeugen kann aus Sensordaten interpoliert werden [HSH05].

Innerhalb dieser Arbeit wird als Trajektorie eine lineare Bewegung vorausgesetzt, bei welcher auch die Orientierung konstant bleibt. Somit lassen sich die Gleichungen (1), (2) und (3) vereinfachen. Gleichungen (6) bis (10) zeigen das komplette System für eine lineare Bewegung.

$$m u = p_x - t v_x \quad (6)$$

$$0 = p_y - t v_y \quad (7)$$

$$m f = p_z - t v_z \quad (8)$$

$$r = t lps \quad (9)$$

$$c = u/S_x + C_x \quad (10)$$

Der Bewegungsvektor v wird in dem Kamerakoordinatensystem definiert. Dieser beschreibt Richtung und Geschwindigkeit der Bewegung der Kamera. Wechselt man das Referenzsystem bleibt die Position der Kamera fix und das Objekt unter der Kamera bewegt sich mit $-v$.

4.3. Starre Abbildungen

Starre Abbildungen beschreiben die Transformationen zwischen Koordinatensystemen. Mit ihnen ist es möglich, Punkte und Vektoren, die gegenwärtig im System $s1$ definiert sind, in ein anderes System $s2$ zu transformieren. Dabei wird von orthonormalen Systemen ausgegangen.

Folgend werden zwei Arten vorgestellt, wie starre Abbildungen definiert werden können.

4.3.1. Pose

Die Pose beschreibt eine starre Abbildung mit 6 Parametern. Diese sind $t_x, t_y, t_z, \alpha, \beta$ und γ . t_x, t_y und t_z stellen eine Verschiebung dar. α, β und γ sind Winkel für Rotationen um die x, y und z -Achse.

Die komplette Transformation lässt sich durch folgende Abfolge beschreiben:

$$p^{s2} = {}_{s1}^{s2}P(p^{s1}) \quad (11)$$

$$= (\text{Trans}(t_x, t_y, t_z) \circ \text{Rot}_x(\alpha) \circ \text{Rot}_y(\beta) \circ \text{Rot}_z(\gamma))(p^{s1}) \quad (12)$$

Von rechts nach links gelesen sind die Rotationen und die Translation auf das jeweils vorherige Koordinatensystem bezogen.

Auf diese Weise erreicht man eine von mehreren möglichen minimalen Parametrisierungen einer starren Abbildung.

Bei der Nutzung der Parametrisierung von Posen durch eine Translation und drei Winkel können alle starren Abbildungen dargestellt werden. Die Jacobi-Matrix verliert allerdings einen Rang, wenn der mittlere Winkel ungünstig liegt. Mit der hier genutzten Definition tritt dieser Fall auf, wenn β auf 90° oder 270° steht. In diesem Fall verursacht die Drehung um β , dass die Rotationsachsen für α und γ linear abhängig sind. Somit wirken α und

γ in die gleiche oder in die entgegengesetzte Richtung. Das führt dazu, dass die Jacobi-Matrix nicht erlaubt, um die Achse zu drehen, welche orthogonal auf der Achse von β und der gemeinsamen von α und γ steht. Dies kann vor allem in Optimierungsaufgaben zu Problemen führen

4.3.2. Homogene Matrix

Eine weitere Möglichkeit, starre Abbildungen darzustellen, sind homogene Matrizen. Mit Hilfe homogener Matrizen lassen sich Punkte und Vektoren zwischen Koordinatensystemen transformieren. Homogene Punkte und Vektoren entstehen aus der Erweiterung der Koordinaten um ein vierter Element. Bei Punkten ist dieses Element 1. Vektoren werden mit einer 0 aufgefüllt.

Eine homogene Matrix ist eine 4×4 -Matrix. Für die hier betrachteten Fälle von starren Abbildungen besteht die letzte Zeile aus $[0, 0, 0, 1]$ ⁷. Die obere linke 3×3 -Matrix beschreibt die Rotation. Die einzelnen Vektoren in dieser Matrix sind, der Reihe nach, die x , y und z -Achse des Ausgangssystems definiert im Zielsystem. Diese haben jeweils die Länge 1⁸ und stehen orthogonal zueinander, so dass diese ein rechtshändiges System bilden.

Die letzte Spalte der homogenen Matrix stellt die Verschiebung dar. Hier sind t_x , t_y und t_z direkt ablesbar. Im Zielsystem ist der Punkt $(t_x, t_y, t_z)^T$ der Ursprung des Ausgangssystems.

In Kombination mit homogenen Punkten und Vektoren lassen sich Punkte und Vektoren durch eine Matrixmultiplikation von dem Ausgangssystem in das Zielsystem transformieren.

Umwandlung: Pose zu homogener Matrix

Wie in Gleichung (12) zu sehen ist, beschreibt eine Pose eine Transformation, welche aus drei Rotationen, gefolgt von einer Translation aufgebaut ist.

Diese Elementartransformationen lassen sich auch als homogene Matrizen darstellen. Wie diese definiert sind, ist im Anhang A.1 gezeigt. Die gesamte Transformation lässt sich als Produkt der Teiltransformationen beschreiben.

⁷Dies ist nur für affine Abbildungen gültig. Allgemeine homogene Matrizen können durch andere Einträge auch Projektionen darstellen.

⁸Dies gilt nur, da wir in diesem Kontext keine Skalierungen und keine Scherungen betrachten.

$${}_{s1}^{s2}H = \text{Hom}({}_{s1}^{s2}P) \quad (13)$$

$$= \text{Trans}(t_x, t_y, t_z) R_x(\alpha) R_y(\beta) R_z(\gamma) \quad (14)$$

$$= \begin{pmatrix} {}_{s1}^{s2}R & {}_{s1}^{s2}T \\ 000 & 1 \end{pmatrix} \quad (15)$$

$${}_{s1}^{s2}R = \begin{pmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta \\ \cos \alpha \sin \gamma + \sin \alpha \sin \beta \cos \gamma & \cos \alpha \cos \gamma - \sin \alpha \sin \beta \sin \gamma & -\sin \alpha \cos \beta \\ \sin \alpha \sin \gamma - \cos \alpha \sin \beta \cos \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma & \cos \alpha \cos \beta \end{pmatrix} \quad (16)$$

$${}_{s1}^{s2}T = (t_x, t_y, t_z)^T \quad (17)$$

$${}_{s2}^s p = {}_{s1}^{s2}H {}_{s1}^{s1}p \quad (18)$$

Mit Hilfe der Gleichung (18) ist es möglich, Punkte und Vektoren zwischen Koordinatensystemen zu transformieren.

Das Verhalten der Transformation in Bezug auf die Parameter der Pose kann durch die Jacobi-Matrix lokal linear angenähert werden. Hierfür werden die partiellen Ableitungen der Koordinaten von ${}_{s2}^s p$ nach den Posenparametern mit Hilfe von Gleichung (18) berechnet. Gleichung (19) zeigt den Aufbau der Jacobi-Matrix.

$$J_{{}_{s1}^{s2}P}({}_{s1}^{s1}p) = \begin{pmatrix} \frac{\delta {}_{s2}^s p_x}{\delta t_x} & \frac{\delta {}_{s2}^s p_x}{\delta t_y} & \frac{\delta {}_{s2}^s p_x}{\delta t_z} & \frac{\delta {}_{s2}^s p_x}{\delta \alpha} & \frac{\delta {}_{s2}^s p_x}{\delta \beta} & \frac{\delta {}_{s2}^s p_x}{\delta \gamma} \\ \frac{\delta {}_{s2}^s p_y}{\delta t_x} & \ddots & & & & \\ \frac{\delta {}_{s2}^s p_z}{\delta t_x} & & & & & \end{pmatrix} \quad (19)$$

Ist die vierte Koordinate von p eine 1, und somit p semantisch ein Punkt, ist die linke 3×3 -Teilmatrix aus Gleichung (19) eine Einheitsmatrix. Auf einen Vektor dagegen hat die Verschiebung zwischen den Koordinatensystemen keinen Einfluss. Somit ist dort die linke 3×3 -Teilmatrix nur mit Nullen gefüllt.

Umwandlung: Homogene Matrix zu Pose

Will man aus einer gegebenen homogenen Matrix eine Pose rekonstruieren, kann man die Parameter t_x , t_y und t_z direkt ablesen. Für die Bestimmung der Winkel müssen die Komponenten der Rotationsmatrix mit denen aus Gleichung (16) gleichgesetzt werden. In dieser Arbeit wird dieses Vorgehen, wie in Gleichung (20) beschrieben, abgekürzt.

$${}_{s2}^s P = \text{Pose}({}_{s2}^s H) \quad (20)$$

Invertierung

Homogene Matrizen, die starre Abbildungen beschreiben, lassen sich vereinfacht invertieren.

$$\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} H = \begin{pmatrix} s_2 \\ s_1 \end{pmatrix} H^{-1} \quad (21)$$

$$= \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \quad (22)$$

$$= \begin{pmatrix} r_{11} & r_{21} & r_{31} & -r_{11}t_x - r_{21}t_y - r_{31}t_z \\ r_{12} & r_{22} & r_{32} & -r_{12}t_x - r_{22}t_y - r_{32}t_z \\ r_{13} & r_{23} & r_{33} & -r_{13}t_x - r_{23}t_y - r_{33}t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (23)$$

Eine ausführliche Herleitung hierzu ist im Anhang A.2 zu finden.

4.4. Levenberg-Marquardt-Algorithmus

Der Levenberg-Marquardt-Algorithmus ist ein iteratives, gradientenbasiertes Optimierungsverfahren zur Minimierung eines quadratischen Fehlers. Es handelt sich dabei um eine Abwandlung des Gauß-Newton-Verfahrens [NW06].

Beide Verfahren minimieren den quadratischen Fehler $\epsilon^T \epsilon$ eines Problems. Im Kontext dieser Arbeit ist ϵ die Abweichung zwischen Modellvorhersagen und Beobachtungen.

Die zu optimierende Funktion, hier $\epsilon = \text{Prob}(\text{par})$, liefert einen Vektor mit Fehlerwerten ϵ . Ihre Parameter sind durch den Vektor par beschrieben. Weiterhin muss die Jacobi-Matrix $J = J_{\text{Prob}}(\text{par})$ bekannt sein. Diese enthält die partiellen Ableitungen der Fehler nach den Parametern. Gleichungen (24), (25) und (26) zeigen kurz den Aufbau der Vektoren ϵ und par und der Jacobi-Matrix J .

$$\epsilon = (\epsilon_1 \ \epsilon_2 \ \dots \ \epsilon_n)^T \quad (24)$$

$$\text{par} = (\text{par}_1 \ \text{par}_2 \ \dots \ \text{par}_m)^T \quad (25)$$

$$J = \begin{pmatrix} \frac{\delta \epsilon_1}{\delta \text{par}_1} & \frac{\delta \epsilon_1}{\delta \text{par}_2} & \dots & \frac{\delta \epsilon_1}{\delta \text{par}_m} \\ \frac{\delta \epsilon_2}{\delta \text{par}_1} & \frac{\delta \epsilon_2}{\delta \text{par}_2} & \dots & \frac{\delta \epsilon_2}{\delta \text{par}_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta \epsilon_n}{\delta \text{par}_1} & \frac{\delta \epsilon_n}{\delta \text{par}_2} & \dots & \frac{\delta \epsilon_n}{\delta \text{par}_m} \end{pmatrix} \quad (26)$$

Gauß-Newton

Das Gauß-Newton-Verfahren ist ein iteratives Verfahren. Ausgehend von einem Startparametersatz $\text{par} = \text{parStart}$ wird wie folgt vorgegangen:

Zuerst wird der Fehlervektor $\epsilon = \text{Prob}(\text{par})$ und die Jacobi-Matrix $J = J_{\text{Prob}}(\text{par})$ bestimmt. Darauf folgend kann das Gleichungssystem $J^T J \Delta = -J^T \epsilon$ gelöst werden. Der neue Parametersatz ergibt sich aus $\text{par} = \text{par} + \Delta$.

Abgebrochen wird nach einer maximalen Anzahl Iterationen oder bei Unterschreiten einer Schranke bei der Verbesserung des quadratischen Fehlers $\epsilon^T \epsilon$.

Bei einem linearen Verhalten von $\text{Prob}(\text{par})$ in allen Parametern würde eine Iteration genügen, um Δ so zu bestimmen, dass der quadratische Fehler $\epsilon^T \epsilon$ mit $\epsilon = \text{Prob}(\text{par} + \Delta)$ minimal ist.

Weist die Funktion kein lineares Verhalten auf, ist die Konvergenz des Verfahrens nicht garantiert. Ausführlich wird das Gauß-Newton-Verfahren in [NW06] behandelt.

Levenberg-Marquardt

Bei dem Levenberg-Marquardt-Verfahren wird der Term $J^T J$ augmentiert. Die zu lösende Gleichung ist dann $(J^T J + \lambda I) \Delta = -J^T \epsilon$.

Mit einem positiven Wert für λ nahe 0 ist eine Iteration ähnlich einer Iteration mit dem Gauß-Newton-Verfahren. Große Werte für λ dagegen approximieren das System mit $\lambda I = -J^T \epsilon$. Da $J^T \epsilon$ der Summe aller Fehlergradienten entspricht, geht das Verfahren für große Werte für λ in ein Gradientenabstiegsverfahren über. Die Schrittgröße verringert sich dabei mit steigendem λ .

Das Levenberg-Marquardt-Verfahren geht somit von einem Gauß-Newton-Verfahren mit steigendem λ in einen Gradientenabstieg über.

Als Startwert für λ wird beispielsweise das 10^{-3} -fache des Durchschnittes der Elemente auf der Hauptdiagonalen von $J^T J$ gewählt [HZ04].

Innerhalb der Iterationen wird im Levenberg-Marquardt-Verfahren λ in jedem Schritt angepasst. Ist der quadratische Fehler $Err = \epsilon^T \epsilon$ kleiner als der bisherige, wird λ durch den Faktor 10 geteilt. Sollte dagegen der Fehler größer als der bisherige sein, wird das Ergebnis verworfen und λ mit 10 multipliziert.

Im Folgenden ist der Algorithmus in Pseudocode dargestellt.

```

Data: parStart, Prob,  $J_{\text{Prob}}$ ,  $\lambda$ 
Result: parOpt, Err
iter = 0;
 $\epsilon$  = Prob(parStart);
 $J$  =  $J_{\text{Prob}}(\text{parStart})$ ;
parOpt = parStart;
Err =  $\epsilon^T \epsilon$ ;
while (not result-is-good) and (iter < maxIter) do
     $A = J^T J + \lambda I$ ;
     $h = -J^T \epsilon$ ;
     $\Delta = A^{-1} h$ ;
    parTmp = parOpt +  $\Delta$ ;
     $\epsilon Tmp$  = Prob(parTmp);
    ErrTmp =  $\epsilon Tmp^T \epsilon Tmp$ ;
    if ErrTmp < Err then
         $\lambda = \lambda / 10$ ;
        parOpt = parTmp;
        Err = ErrTmp;
         $\epsilon = \epsilon Tmp$ ;
         $J$  =  $J_{\text{Prob}}(\text{parOpt})$ ;
    else
        |  $\lambda = 10 \lambda$ ;
    end
    iter = iter + 1;
end
```

Algorithmus 1: Levenberg-Marquardt-Algorithmus

Der Algorithmus wird ausführlich in [NW06] behandelt. In [HZ04] wird der Algorithmus mit den wichtigsten Eigenschaften kurz dargestellt. Abweichend zu dem hier vorgestellten Algorithmus wird in [PTVF92] die Hauptdiagonale bei der Erstellung von A mit $1 + \lambda$ multipliziert.

5. Problemspezifische theoretische Grundlagen

Neben den in Kapitel 4 vorgestellten allgemeinen Grundlagen werden für die weitere Arbeit spezielle Grundlagen benötigt. Diese umfassen insbesondere das verwendete Kameramodell und die Modellierung eines Stereo-Aufbaus.

5.1. Einschränkungen im Kontext dieser Arbeit

Für den Einsatz von Stereo-Vision im industriellen Einsatz können Einschränkungen getroffen werden. Mit diesen wird das Problem stark vereinfacht. In industriellen Anwendungen kann oftmals von linearen Bewegungen ausgegangen werden. Ein Beispiel hierfür ist eine Befestigung zweier Zeilenkameras über einem Förderband. Ein äquivalenter Fall ist die Befestigung der Kameras auf einem Schlitten, welcher mit einer konstanten Geschwindigkeit linear geführt wird.

Dies führt zu den folgenden Vereinfachungen:

- Die relative Pose zwischen den Kameras ist konstant.
- Es herrscht eine gleichförmige lineare Bewegung zwischen Kameras und zu rekonstruierendem Objekt.

Weiterhin ist im industriellen Umfeld die Möglichkeit gegeben, Kalibrierkörper für die Bestimmung der Modellparameter zu verwenden. Abbildung 6 zeigt einen möglichen Kalibrierkörper.

Damit unterscheidet sich der Aufbau klar von Anwendungsfällen der Fernerkundung. Dort herrschen keine linearen Bewegungen. Stattdessen werden die Bewegungen beispielsweise mit Ellipsen angenähert oder aus Sensorwerten (GPS, Gyroskop, Accelerometer) interpoliert. Dabei ändert sich auch die Orientierung der Kamera zwischen dem Aufnehmen zweier Zeilen.

Weiterhin ist die relative Pose zwischen den Kameras nicht zwangsläufig konstant. Oft befindet sich am Träger eine nach vorne und eine nach hinten ausgerichtete Kamera. Somit können Aufnahmen zweier zeitlich versetzter Zeitintervalle der Kameras verglichen werden. Die relative Pose ist aber abhängig von der Zeit.

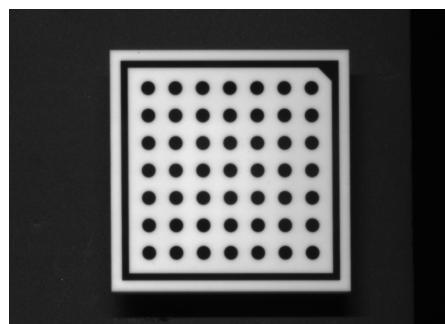


Abbildung 6: Bild einer Kalibrierplatte der Größe $30\text{mm} \times 30\text{mm}$ mit 49 Marken.

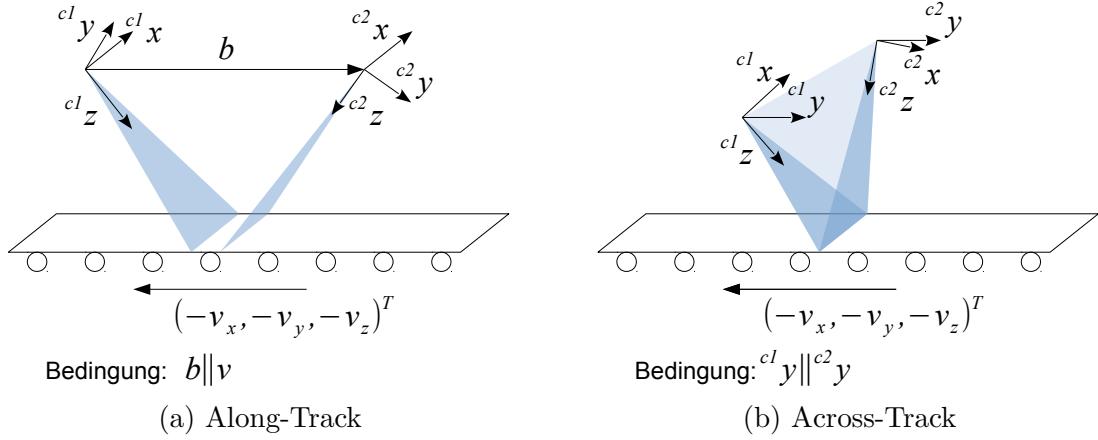


Abbildung 7: Ideale Along-Track- und Across-Track-Konfigurationen.

Für die Kalibrierung des Aufbaus stehen auch keine Kalibrierkörper zur Verfügung. Stattdessen müssen bekannte Kontrollpunkte auf der Erdoberfläche verwendet werden.

Unterschiede zu konventionellem Stereo

Bei Stereo-Vision mit Zeilenkameras entstehen die Bilder durch eine Parallelprojektion entlang des Bewegungsvektors und eine perspektivische Projektion entlang der Sensorausrichtung.

Die Mischung dieser beiden Projektionsarten führt zu einer komplexen Epipolargeometrie. Während bei Flächenkameras, nachdem Verzeichnungen korrigiert wurden, die Epipolarkurven sich stets auf Geraden abbilden, ist bei Stereo-Vision mit Zeilenkameras die Abbildung abhängig von der relativen Pose zwischen den Kameras und dem Bewegungsvektor. Werden verzeichnungsfreie Zeilenkameras betrachtet, haben die Epipolarkurven in diesem Aufbau die Form von Hyperbeln, wobei die Form der Hyperbel in einem Bild auch von der betrachteten Spalte im anderen Bild abhängt (siehe dazu [GH97], [HMJK05a], [HMJK05b], [Kim00]). Aus diesem Grund ist es bei einem allgemeinen Aufbau nicht möglich, die Bilder zu rektifizieren.

Wie [HMJK05b] herausstellt, bestehen allerdings Ausnahmen in zwei Fällen. Dabei handelt es sich um eine Idealisierung der aus der Fernerkundung bekannten Fälle Along-Track und Across-Track. Diese sind in Abbildung 7 gezeigt.

Bei dem idealen Along-Track-Fall muss die Verbindungsstrecke zwischen den Projektionszentren parallel zur Bewegungsrichtung sein. Ist das der Fall, bilden die Spaltenkoordinaten des Bildes Winkel orthogonal zur Bewegungsrichtung ab. Die Spalten des einen Bildes können Spalten des anderen Bildes zugeordnet werden. Dementsprechend bilden sich Epipolarkurven auf Geraden in y -Richtung ab. Die entscheidenden Komponenten für eine Rekonstruktion basieren in diesem Fall auf denen der Parallelprojektion. Diese sind die Zeilenrate der Kameras und die Geschwindigkeit der Bewegung.

Der ideale Across-Track-Fall setzt dagegen voraus, dass die Aufnahmeebenen⁹ parallel sind. Dies hat zur Folge, dass beide Kameras in einer Bildzeile Abbildungen derselben Ebene aufnehmen. In dem idealen Across-Track-Fall verlaufen die Epipolarkurven entlang der x -Richtung in den Bildern. Betrachtet man nur eine Zeile aus dem einen Bild und die korrespondierende Zeile aus dem anderen Bild entsteht somit eine Geometrie, welche der von zwei korrespondierenden Zeilen bei Stereo-Vision mit Flächenkameras entspricht. Demnach basiert die Rekonstruktion auf der perspektivischen Projektion. Entscheidend ist hier hauptsächlich die Auflösung der Zeilen und die Distanz der auf die zz -Ebene der Kamerakoordinaten projizierten Projektionszentren.

5.2. Kameramodell

In Abschnitt 4.2 wurde ein allgemeines Kameramodell für ideale Zeilenkameras vorgestellt. Neben einem Modell für eine allgemeine Trajektorie wurde auch das Modell für eine lineare Bewegung präsentiert.

Für eine reale Anwendung sollte nicht von einer idealen Kamera ausgegangen werden. Das hier verwendete Modell stammt aus [SUW08]. Dieses betrachtet zusätzlich Fehler in der Position des Hauptpunktes und eine radiale Verzeichnung der verwendeten Optik. Folgende Gleichungen beschreiben das Modell. Diese definieren die Abbildung einer Koordinate $(x, y, z)^T$ im Raum auf den Bildpunkt $(r, c)^T$ in dem Kamerabild. Es wird angenommen, dass sowohl der Vektor v als auch der Punkt $(x, y, z)^T$ im Kamerakoordinatensystem definiert sind.

$$m D u = x - t v_x \quad (27)$$

$$-m D p_v = y - t v_y \quad (28)$$

$$m f = z - t v_z \quad (29)$$

$$D = 1/(1 + \kappa (u^2 + p_v^2)) \quad (30)$$

$$p_v = S_y C_y \quad (31)$$

$$r = t \quad (32)$$

$$c = u/S_x + C_x \quad (33)$$

Abweichend zum idealen Modell (Gleichungen (6) bis (10)) wurde auf die Zeilenrate lps verzichtet. Stattdessen sind die Geschwindigkeitskomponenten v_x , v_y und v_z in [m/Zeile] gegeben ($v[m/\text{Zeile}] = v[m/s]/lps$).

5.2.1. Kameraparameter

Abbildung 8 zeigt die geometrische Entsprechung der verschiedenen Parameter. Dabei ist die geometrische Interpretation von κ , dem Parameter für die Verzeichnung, nicht erkennbar.

⁹Ebene, die durch das Sensorarray und das Projektionszentrum, also x - und z -Achse des Kamerakoordinatensystems, aufgespannt wird.

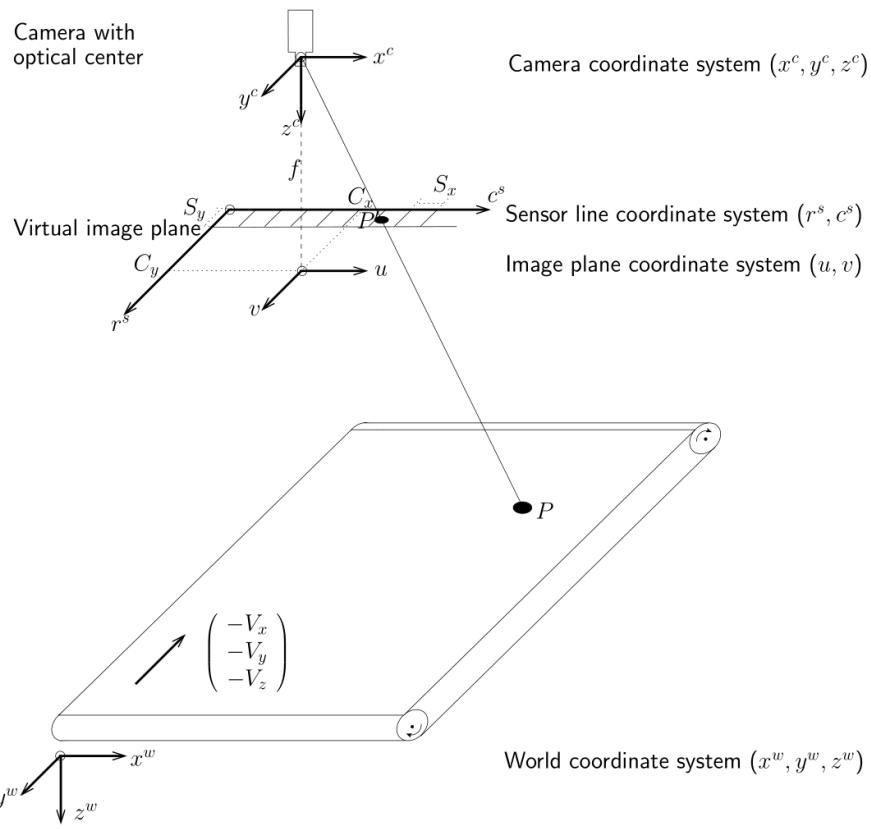


Abbildung 8: Parameter einer Zeilenkamera (aus [MVT13a]).

Parameter	Wert	Einheit	Beschreibung
f	0.5	m	Brennweite
κ	0	$\frac{1}{m^2}$	Verzeichnungskoeffizient
S_x	$5e^{-5}$	$\frac{m}{Pixel}$	Abstand der Sensorelemente
S_y	$5e^{-5}$	$\frac{m}{Pixel}$	y -Ausdehnung eines Sensorelements
C_x	500.5	Pixel	x -Position des Hauptpunktes
C_y	0	Pixel	y -Position des Hauptpunktes
v_x	0	$\frac{m}{Zeile}$	x -Komponente des Bewegungsvektors
v_y	$5e^{-5}$	$\frac{m}{Zeile}$	y -Komponente des Bewegungsvektors
v_z	0	$\frac{m}{Zeile}$	z -Komponente des Bewegungsvektors

Tabelle 1: Beispieldatensatz für eine Zeilenkamera.

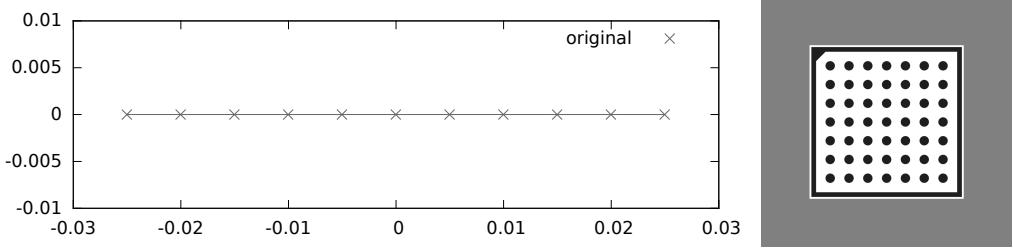


Abbildung 9: Simulation mit Beispielparametern.

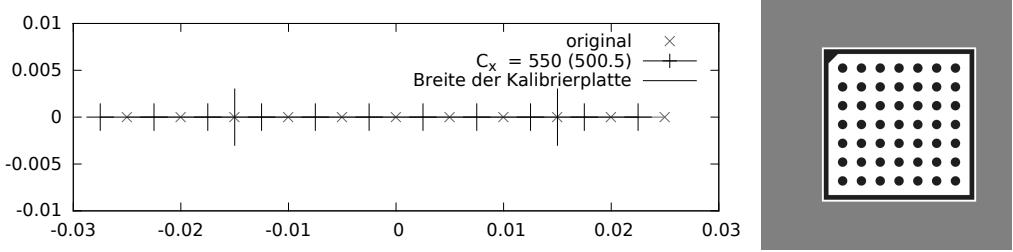


Abbildung 10: Einfluss von C_x .

In diesem Abschnitt werden die Einflüsse der verschiedenen Kameraparameter gezeigt. Tabelle 1 zeigt hierfür einen Beispielparametersatz. Ausgehend von diesen Parametern und einer Bildgröße von 1001×1001 Pixeln können Bilder simuliert werden. Als zu simulierendes Objekt wird die in Abbildung 6 gezeigte Kalibrierplatte verwendet. Der Mittelpunkt der Kalibrierplatte wird in Kamerakoordinaten um $(0, 0.025, 0.5)^T$ verschoben. Somit besteht ein Abstand zur Kamera und die Kalibrierplatte ist komplett in dem entstehenden Bild zu sehen.

Verschneidet man zusätzlich die Strahlen der einzelnen Pixel zum Zeitpunkt $t = 0$ mit der xy -Ebene bei $z = 0.5$ können die Punkte gezeigt werden, welche der Sensor zur Zeit $t = 0$ erfasst. Die so ermittelten Punkte zeigen die Form der resultierenden Aufnahmefläche. In Abbildung 9 sind die so erzeugten Bilder gezeigt. Bei der Visualisierung der aufgenommenen Zeile ist jedes hundertste Pixel als Punkt markiert.

In den folgenden Abbildungen ist die Ursprungszeile mit abgebildet. Die Änderung der Parameter von den Beispielwerten ist mit angegeben. Dabei steht in Klammern der Ausgangswert.

Eine Verschiebung des Sensorarrays in x -Richtung wurde in Abbildung 10 simuliert. Verglichen mit dem Ausgangspараметern verursacht eine Erhöhung von C_x eine Verschiebung der aufgenommenen Punkte nach links. Somit wird die simulierte Kalibrierplatte im Bild nach rechts verschoben.

Abbildung 11 zeigt die Bilder bei einer Veränderung von κ . Ein positiver Wert (Abbildung 11a) führt dazu, dass die Pixel sich in der Bildzeile stauchen. Dabei werden diese mehr gestaucht, umso weiter sie von $x = 0$ entfernt sind. Folglich wird die Kalibrierplatte zu den Rändern in x -Richtung gestreckt. Ein negativer Wert führt zu einem Strecken der Bildzeile und somit einem Stauchen der simulierten Kalibrierplatte. Abbildung 11b zeigt dieses Verhalten.

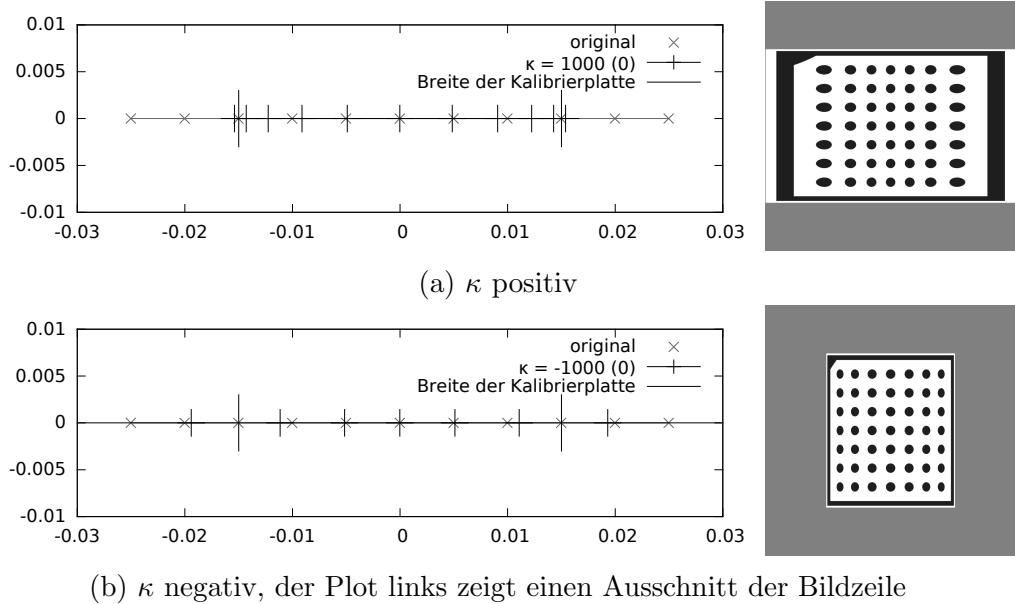


Abbildung 11: Einfluss von κ .

In Abbildung 12 wird der Einfluss des Produkts $S_y C_y$ mit und ohne Verzeichnung gezeigt. Die Verzeichnung führt dazu, dass die Bildzeile zu den Rändern hin sich zu $y = 0$ biegt ($\kappa > 0$) oder sich von $y = 0$ entfernt ($\kappa < 0$).

Eine Variation von f oder S_x führt zu einer Skalierung der Bildzeile. Dies wird in den Abbildungen 13 und 14 gezeigt. Bei $\kappa = 0$ und $p_v = 0$ ist der Einfluss der Parameter ähnlich.

Die Parameter v_x , v_y und v_z haben keine Auswirkung auf die Abbildung der Bildzeile bei $t = 0$. Die simulierten Bilder der Kalibrierplatte werden aber direkt beeinflusst. Abbildung 15 verdeutlicht dies.

5.2.2. Unbestimmtheiten

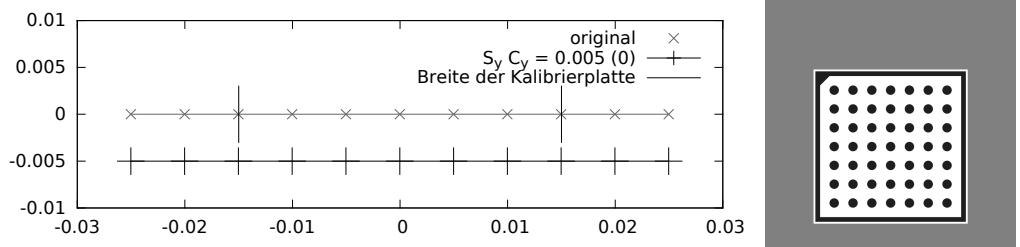
Bei der Diskussion der Kameraparameter sind Unbestimmtheiten erkennbar. Eine Unbestimmtheit besteht, wenn mehrere Parametersätze zu demselben Modell führen.

Offensichtlich ist dies bei den Parametern S_y und C_y . Diese gehen nur als Produkt ein. Somit ist ein Modell mit S_y und C_y äquivalent zu einem Modell mit den Parametern $a S_y$ und C_y/a .

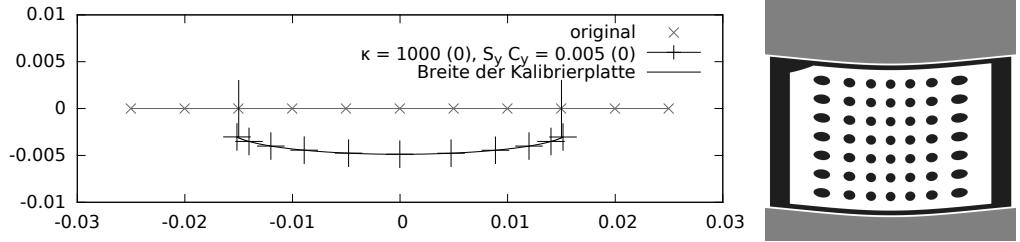
Komplexer ist dieser Zusammenhang bei der Skalierung einer Bildzeile (siehe hierzu Abbildungen 13 und 14). Bei $\kappa = 0$ und $p_v = 0$ ist das Modell mit f und S_x gleich einem Modell mit $a f$ und $a S_x$.

Dies lässt sich auf beliebige Werte für κ und p_v erweitern.

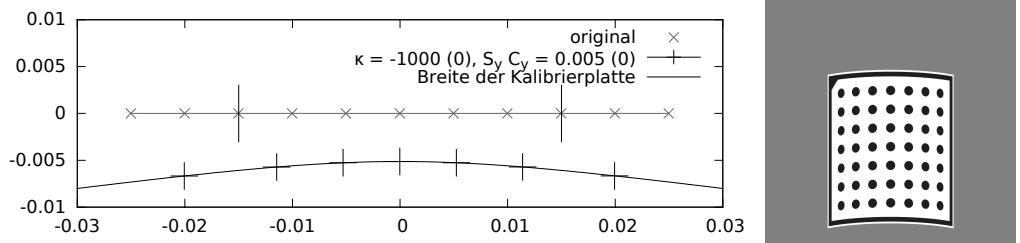
Substituiert man f mit $a f$, simuliert man eine Anpassung von f . Aus Gleichung (29) folgt, dass m mit m/a ersetzt werden muss. Damit die Gleichungen (27) und (28) gelten, wird D konstant gehalten. Folglich werden u und p_v mit $a u$ und $a p_v$ ersetzt. p_v wiederum hängt ab von S_y und C_y . Da diese Parameter allerdings nur p_v beeinflussen, kann der



(a) ohne Verzeichnung



(b) mit Verzeichnung (κ positiv)



(c) mit Verzeichnung (κ negativ, der Plot links zeigt einen Ausschnitt der Bildzeile)

Abbildung 12: Einfluss von $S_y C_y$.

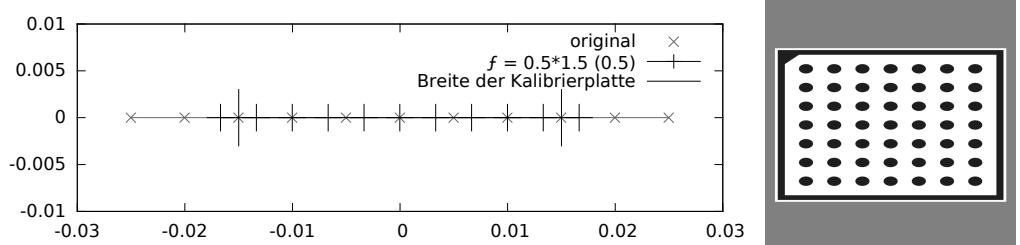


Abbildung 13: Einfluss von f .

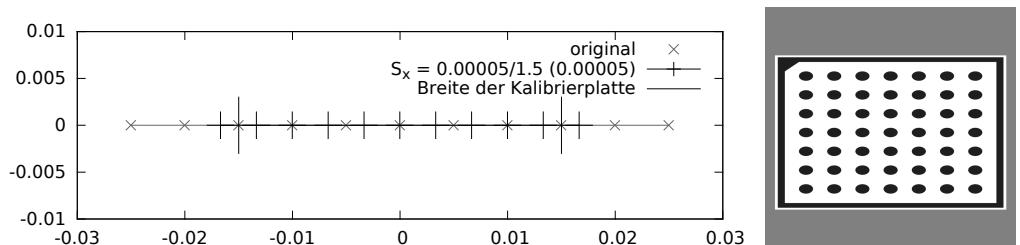
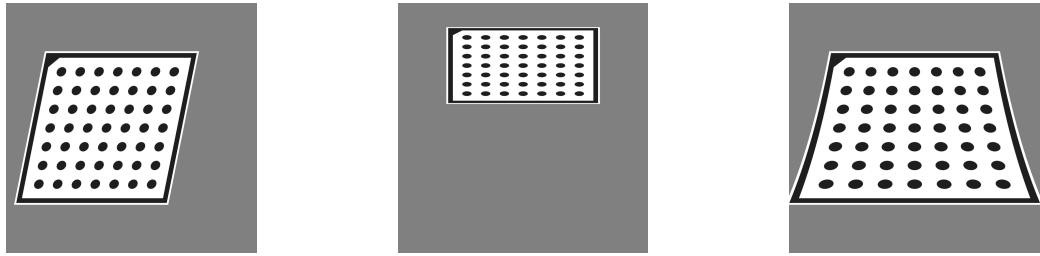


Abbildung 14: Einfluss von S_x .



(a) v_x von 0 auf $1 \cdot 10^{-5}$ (b) v_y von $5 \cdot 10^{-5}$ auf $10 \cdot 10^{-5}$ (c) v_z von 0 auf $2.5 \cdot 10^{-4}$

Abbildung 15: Einfluss des Bewegungsvektors.

Faktor a beliebig auf die Terme aufgeteilt werden. D ist abhängig von u und p_v . Damit D konstant bleibt, muss κ diese Einflüsse ausgleichen. Dies ist möglich, indem κ durch κ/a^2 substituiert wird. Ersetzt man nun S_x mit $a S_x$ ist auch Gleichung (33) erfüllt. Somit ändern sich die Gleichungen des Kameramodells zu den folgenden:

$$(m/a) D (a u) = x - t v_x \quad (34)$$

$$-(m/a) D (a p_v) = y - t v_y \quad (35)$$

$$(m/a) (a f) = z - t v_z \quad (36)$$

$$D = 1/(1 + (\kappa/a^2)((u a)^2 + (p_v a)^2)) \quad (37)$$

$$a p_v = a S_y C_y \quad (38)$$

$$r = t \quad (39)$$

$$c = (a u)/(a s_x) + c_x \quad (40)$$

Aus all diesen Gleichungen kann a gekürzt werden. Somit ist das System mit den genannten Anpassungen der Parameter äquivalent zum Ausgangssystem.

Damit das System eindeutig ist, empfiehlt es sich, S_x und S_y als positive Konstanten anzusehen.

5.2.3. Abbildung eines Punktes zu einer Bildkoordinate

Für die Zuordnung von einem Punkt zu einer Bildkoordinate müssen die Gleichungen (27) bis (33) gelöst werden. Um diese Gleichungen robust zu lösen, müssen auf Grund der quadratischen Einflüsse von u und p_v mehrere Fallunterscheidungen beim Bestimmen der Größen m und t durchgeführt werden. Für das Erstellen dieser Arbeit wurde eine effiziente und robuste Funktion zum Lösen der Gleichungen von MVTec zur Verfügung gestellt. Genauso wurde auch die Berechnung der Jacobi-Matrix bereitgestellt. Die Lösung der Gleichungen basiert auf einer in mehrere Fälle unterteilte Bestimmung von m . Sobald m bestimmt ist, kann auch t ermittelt werden. Für die Bestimmung der Jacobi-Matrix wird dem gleichen Schema gefolgt, um die partiellen Ableitungen zu bilden.

In der weiteren Arbeit wird die Abbildung von Punkten in Kamerakoordinaten zu Bildkoordinaten, wie in Gleichung (41) angegeben, verwendet. Hierbei werden die Kamera-

parameter (f , κ , S_x , S_y , C_x und C_y) als cp zusammengefasst. Weiterhin ist der Bewegungsvektor v und der Punkt p in dem Koordinatensystem der Kamera definiert.

$$\begin{pmatrix} r \\ c \end{pmatrix} = \text{lineCam}(v, cp, p) \quad (41)$$

Die entstehende Jacobi-Matrix hat folgende Form:

$$J_{\text{lineCam}} = \begin{pmatrix} \frac{\delta v_x}{\delta r} & \frac{\delta v_y}{\delta r} & \frac{\delta v_z}{\delta r} & \frac{\delta f}{\delta r} & \frac{\delta \kappa}{\delta r} & \frac{\delta S_x}{\delta r} & \frac{\delta S_y}{\delta r} & \frac{\delta C_x}{\delta r} & \frac{\delta C_y}{\delta r} & \frac{\delta x}{\delta r} & \frac{\delta y}{\delta r} & \frac{\delta z}{\delta r} \\ \frac{\delta v_x}{\delta c} & \frac{\delta v_y}{\delta c} & \frac{\delta v_z}{\delta c} & \frac{\delta f}{\delta c} & \frac{\delta \kappa}{\delta c} & \frac{\delta S_x}{\delta c} & \frac{\delta S_y}{\delta c} & \frac{\delta C_x}{\delta c} & \frac{\delta C_y}{\delta c} & \frac{\delta x}{\delta c} & \frac{\delta y}{\delta c} & \frac{\delta z}{\delta c} \end{pmatrix} \quad (42)$$

$$J_{\text{lineCam}} = \begin{pmatrix} J_v^{2 \times 3} & J_{cp}^{2 \times 6} & J_p^{2 \times 3} \end{pmatrix} \quad (43)$$

5.2.4. Abbildung einer Bildkoordinate zu einem Strahl

Die Abbildung von einer Bildkoordinate zu einem Strahl im 3D-Raum ist mit Hilfe der Gleichungen einfach umsetzbar. Da m die Tiefe kodiert, können die Gleichungen einmal für $m = 0$ gelöst werden. Der resultierende Punkt beschreibt das Projektionszentrum der Kamera und kann als Ursprung des Strahls angesehen werden. Ein weiterer Punkt auf dem Strahl kann beispielsweise mit einem Setzen von $m = 1$ berechnet werden. Somit können Ursprung und Richtungsvektor des Strahls bestimmt werden.

5.3. Stereo-Aufbau

Mit Hilfe des Kameramodells aus Abschnitt 5.2 und den in Abschnitt 4.3.1 vorgestellten Posen kann ein Stereo-Aufbau beschrieben werden. Dabei wird die erste Kamera als Referenzkamera betrachtet. Die Posen beschreiben Transformationen von oder zu dem System der Referenzkamera. Die wichtigste Pose dabei beschreibt die relative Pose der zweiten Kamera. Der Bewegungsvektor wird im Referenzsystem definiert und kann bei Bedarf in das System der zweiten Kamera transformiert werden. Für die Kalibrierung sind auch die Posen zu den Kalibrierkörpern in den Bildern relevant. Der Aufbau ist in Abbildung 16 mit den relevanten Posen visualisiert. Die Posen beschreiben die starre Abbildung in Richtung der abgebildeten Pfeile.

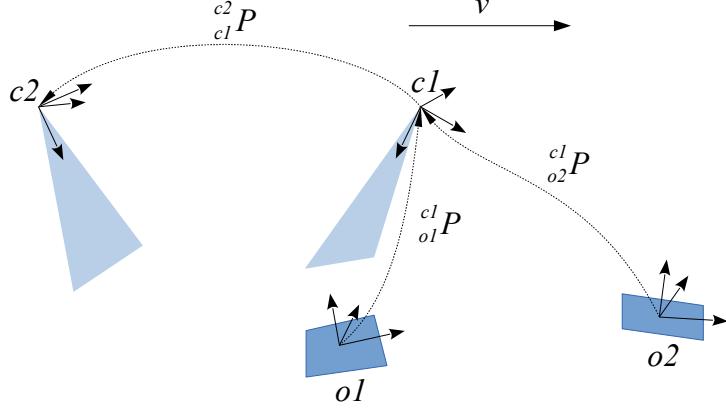


Abbildung 16: Ein Aufbau mit zwei Kameras. Definition von Koordinatensystemen und Posen. $o1$ und $o2$ sind die Koordinatensysteme von Kalibrierkörpern.

5.3.1. Parameter

Aus der verbalen Beschreibung kann auch die Parametrisierung abgeleitet werden.

$$\begin{aligned}
 par_{\text{stereoSetup}} = [& \underbrace{\underbrace{v_x, v_y, v_z}_{c^1 v = [v_x, v_y, v_z, 0]^T}, \\
 & \underbrace{\underbrace{f_{c1}, \kappa_{c1}, S_{x c1}, S_{y c1}, C_{x c1}, C_{y c1},}_{cp_{c1}} \\
 & \underbrace{\underbrace{f_{c2}, \kappa_{c2}, S_{x c2}, S_{y c2}, C_{x c2}, C_{y c2},}_{cp_{c2}} \\
 & \underbrace{\underbrace{c_1^2 t_x, c_1^2 t_y, c_1^2 t_z, c_1^2 \alpha, c_1^2 \beta, c_1^2 \gamma,}_{c_1^2 P} \\
 & \underbrace{\underbrace{c_1^1 t_x, c_1^1 t_y, c_1^1 t_z, c_1^1 \alpha, c_1^1 \beta, c_1^1 \gamma,}_{c_1^1 P} \\
 & \vdots \\
 & \underbrace{\underbrace{c_M^1 t_x, c_M^1 t_y, c_M^1 t_z, c_M^1 \alpha, c_M^1 \beta, c_M^1 \gamma]}_{c_M^1 P}]^T \quad (44)
 \end{aligned}$$

Das gesamte Modell hat also $3 + 6 + 6 + 6 = 21$ Parameter für die Konfiguration des Aufbaus und $6 M$ weitere für die Posen der M Kalibrierkörper.

5.3.2. Gesamtes Modell

Mit Hilfe dieser Parametrisierung lässt sich nun ein homogener Punkt ${}^{om}p$ in dem Koordinatensystem des Kalibrierkörpers m in Zeilen und Spalten beider Kameras projizieren. Hierfür wird eine Funktion `stereoSetup` definiert (Gleichung (45)). Deren Rückgabe setzt sich aus den Modellen der Zeilenkameras zusammen (Gleichungen (47) und (50)).

$$\begin{pmatrix} {}^{c1}r \\ {}^{c1}c \\ {}^{c2}r \\ {}^{c2}c \end{pmatrix} = \text{stereoSetup}({}^{c1}v, cp_{c1}, cp_{c2}, {}^{c1}P, m, {}^{om}p) \quad (45)$$

mit

$${}^{c1}p = \text{Hom}({}^{c1}P) {}^{om}p \quad (46)$$

$$\begin{pmatrix} {}^{c1}r \\ {}^{c1}c \end{pmatrix} = \text{camModel}({}^{c1}v, cp_{c1}, {}^{c1}p) \quad (47)$$

und

$${}^{c2}p = \text{Hom}({}^{c2}P) {}^{c1}p \quad (48)$$

$${}^{c2}v = \text{Hom}({}^{c2}P) {}^{c1}v \quad (49)$$

$$\begin{pmatrix} {}^{c2}r \\ {}^{c2}c \end{pmatrix} = \text{camModel}({}^{c2}v, cp_{c2}, {}^{c2}p) \quad (50)$$

Auch lässt sich die Jacobi-Matrix für das gesamte Modell für gegebene Parameter und einen gegebenen Punkt bilden.

$$J_{\text{stereoSetup}} = \begin{pmatrix} {}^{c1}J_v & {}^{c1}J_{cp1} & 0^{2 \times 6} & 0^{2 \times 6} & {}^{c1}J_{{}^{oi}P} & \dots & {}^{c1}J_{{}^{oM}P} \\ {}^{c2}J_v^* & 0^{2 \times 6} & {}^{c2}J_{cp2} & {}^{c2}J_{{}^{c1}P} & {}^{c2}J_{{}^{oi}P} & \dots & {}^{c2}J_{{}^{oM}P} \end{pmatrix} \quad (51)$$

mit

$$J_{\text{cam1}} = \begin{pmatrix} {}^{c1}J_v & {}^{c1}J_{cp1} & {}^{c1}J_p \end{pmatrix} \quad (52)$$

$$J_{\text{cam2}} = \begin{pmatrix} {}^{c2}J_v & {}^{c2}J_{cp2} & {}^{c2}J_p \end{pmatrix} \quad (53)$$

$$\begin{pmatrix} {}^{c1}H \\ {}^{c2}H \end{pmatrix} = \begin{pmatrix} {}^{c1}R & {}^{c1}T \\ {}^{c2}R & {}^{c2}T \\ 0 & 0 & 1 \end{pmatrix}^{-1} = \text{Hom}({}^{c2}P)^{-1} \quad (54)$$

$${}^{c2}J_v^* = \begin{pmatrix} {}^{c1}R & {}^{c2}J_v^T \end{pmatrix}^T \quad (55)$$

$${}^{c2}J_{{}^{c1}P} = {}^{c2}J_p J_{{}^{c1}P}({}^{c1}p) + {}^{c2}J_v J_{{}^{c1}P}({}^{c1}v) \quad (56)$$

$${}^{c1}J_{{}^{om}P} = {}^{c1}J_p J_{{}^{om}P}({}^{om}p) \quad (57)$$

$${}^{c2}J_{{}^{om}P} = {}^{c2}J_p {}^{c1}R J_{{}^{om}P}({}^{om}p) \quad (58)$$

$$\forall i \neq m \quad {}^{c1}J_{{}^{oi}P} = 0^{2 \times 6} \quad (59)$$

$$\forall i \neq m \quad {}^{c2}J_{{}^{oi}P} = 0^{2 \times 6} \quad (60)$$

Dabei lassen sich die Terme ${}^{c1}J_v$, ${}^{c1}J_{cp1}$ und ${}^{c2}J_{cp2}$ in Gleichung (51) direkt aus den Jacobi-Matrizen der Kameramodelle der beiden Kameras setzen. Diese sind in den Gleichungen (52) und (53) gezeigt.

Da der Bewegungsvektor im Kamerakoordinatensystem der ersten Kamera definiert ist, müssen die partiellen Ableitungen des Bewegungsvektors von Kamera 2 (${}^{c2}J_v$) zuerst, wie in der Gleichung (55) gezeigt, in das System der ersten Kamera transformiert werden. Für die Ableitungen der Posen kommt die mehrdimensionale Kettenregel¹⁰ zum Einsatz.

¹⁰ $f(x) = g(h(x)) \rightarrow J_f(x) = J_g(h(x)) J_h(x)$

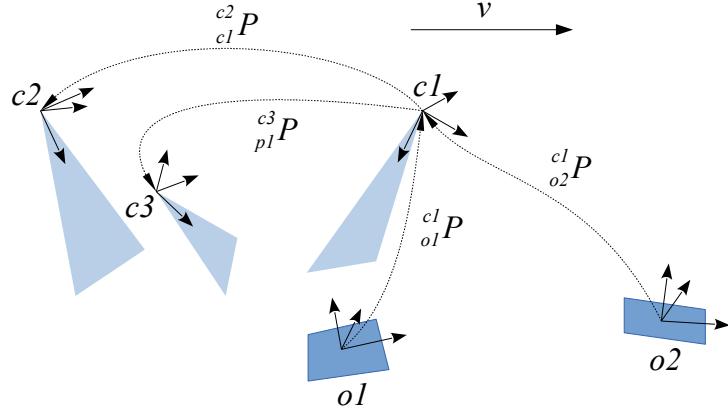


Abbildung 17: Ein Aufbau mit drei Kameras. Definition von Koordinatensystemen und Posen. $o1$ und $o2$ sind die Koordinatensysteme von Kalibrierkörpern.

Die Pose von Kamera 1 zu Kamera 2 beeinflusst sowohl die Position des Punktes ${}^{c2}P$ als auch den Bewegungsvektor ${}^{c2}v$. Somit werden die Teileinflüsse in der Gleichung (56) summiert.

5.4. N-Kamera-Setup

In Abschnitt 5.3 wurde die Abbildung eines Punktes in Kalibrierkörperkoordinaten in Bildpunkte zweier Kameras und die dazugehörige Jacobi-Matrix definiert. Diese Definition lässt sich auf $N \geq 1$ Kameras erweitern. Eine Rekonstruktion ist aber nur mit $N \geq 2$ Kameras möglich.

5.4.1. Anpassung der Parameter

Die erste Kamera wird als Referenzkamera betrachtet. Weitere Kameras werden durch ihre inneren Parameter und die relative Pose zur Referenzkamera definiert. Abbildung 17 zeigt die Transformationen, welche in der Parametrisierung für 3 Kameras verwendet werden.

Der Parametersatz erweitert sich somit:

$$\begin{aligned}
par_{NCams} = & \left[\underbrace{\begin{matrix} v_x, & \underbrace{v_y, & v_z}, \\ c^1 v = [v_x, v_y, v_z, 0]^T \end{matrix}}, \right. \\
& \underbrace{\begin{matrix} f_{c1}, \kappa_{c1}, S_{x\,c1}, S_{y\,c1}, C_{x\,c1}, C_{y\,c1}, \\ cp_{c1} \end{matrix}}, \\
& \vdots \\
& \underbrace{\begin{matrix} f_{cN}, \kappa_{cN}, S_{x\,cN}, S_{y\,cN}, C_{x\,cN}, C_{y\,cN}, \\ cp_{cN} \end{matrix}}, \\
& \underbrace{\begin{matrix} {}^{c2}t_x, {}^{c2}t_y, {}^{c2}t_z, {}^{c2}\alpha, {}^{c2}\beta, {}^{c2}\gamma, \\ {}^{c1}P \end{matrix}}, \\
& \vdots \\
& \underbrace{\begin{matrix} {}^{cN}t_x, {}^{cN}t_y, {}^{cN}t_z, {}^{cN}\alpha, {}^{cN}\beta, {}^{cN}\gamma, \\ {}^{c1}P \end{matrix}}, \\
& \underbrace{\begin{matrix} {}^{c1}t_x, {}^{c1}t_y, {}^{c1}t_z, {}^{c1}\alpha, {}^{c1}\beta, {}^{c1}\gamma, \\ {}^{o1}P \end{matrix}}, \\
& \vdots \\
& \left. \underbrace{\begin{matrix} {}^{c1}t_x, {}^{c1}t_y, {}^{c1}t_z, {}^{c1}\alpha, {}^{c1}\beta, {}^{c1}\gamma]^T \\ {}^{oM}P \end{matrix}} \right]
\end{aligned} \tag{61}$$

Mit N Kameras und M Bildern ergeben sich somit $3 + 6N + 6(N-1) + 6M$ Parameter

5.4.2. Anpassung des Modells

Die Ausgabe des N -Kamera-Modells sind N Paare von Zeile und Spalte im Bild der jeweiligen Kameras.

Das Vorgehen zu der Erzeugung der Ausgaben und dem Bilden der Jacobi-Matrix kann aus den Gleichungen für das Stereo-Modell abgeleitet werden.

Alle Kameras können wie die Kamera 2 aus dem Stereo-Aufbau behandelt werden. Hierfür muss die $c_1^1 P$ mit $(0, 0, 0, 0, 0, 0)^T$ definiert werden.

6. Implementierung der Kalibrierung

In diesem Kapitel wird die Kalibrierung beschrieben. Diese baut auf dem in Abschnitt 5.3 vorgestelltem Modell eines Stereo-Aufbaus mit Zeilenkameras auf. Aufgabe ist die Ermittlung der Modellparameter. Diese enthalten innere Parameter der Kameras und äußere Parameter für die Konfiguration. In diesem sind Nichtlinearitäten durch trigonometrische Funktionen in Posen und Verzeichnungen bei den Kameras enthalten. Somit ist es notwendig, ein nichtlineares Optimierungsverfahren zu verwenden. Hierfür wird das iterative Levenberg-Marquardt-Verfahren verwendet. Dieses wurde in Abschnitt 4.4 vorgestellt.

Die Kalibrierung basiert auf den Mittelpunkten einzelner Kalibrierkörper-Marken. Ziel der Kalibrierung ist es, die Modellparameter so zu bestimmen, dass die Vorhersage der Mittelpunkte möglichst genau zu den Marken der beobachteten Kalibrierkörpern passt.

6.1. Vorbereiten der Kalibrierung

Die Voraussetzung für die Kalibrierung ist die Extraktion der Mittelpunkte der Marken aus Bildern der Kalibrierkörper. Für eine gute Kalibrierung ist dabei eine subpixelgenaue Bestimmung notwendig. In dieser Arbeit werden hierfür Methoden aus HALCON verwendet. Mit einer ersten Methode wird die Region, in welcher sich der Kalibrierkörper befindet, bestimmt. Hierfür wird mit Hilfe eines Schwellwertoperators eine möglichst große konvexe Region gesucht, welche die erwartete Anzahl an Marken als Löcher aufweist.

Innerhalb der Region eines Kalibrierkörpers wird nachfolgend nach den Marken gesucht. Auf der Kalibrierplatte bestehen die Marken aus schwarz gefüllten Kreisen auf weißem Hintergrund. Mit Hilfe von Kantenoperationen und Zusammenhangskomponenten werden die Umrisse der Strukturen innerhalb der Region extrahiert. Nachfolgend werden Umrisse selektiert, welche ein ellipsenförmiges Aussehen haben. Stimmen diese in der Anzahl mit der Beschreibung der Kalibrierplatte überein werden die subpixelgenau extrahierten Mittelpunkte der Ellipsen zurückgegeben. Die Rückgabe der Mittelpunkte erfolgt in der Reihenfolge, welche in der Beschreibung der Kalibrierplatte festgelegt wurde. Die Mittelpunkte der Kalibriermarken werden innerhalb der Methode auch verwendet, um anhand grober Startwerte für Kameraparameter und Bewegungsvektor die relative Pose von Kalibrierplatte zur jeweiligen Kamera zu schätzen.

Weitere Informationen zu diesem Vorgehen sind in [MVT13b] unter „find_caltab“ und „find_marks_and_pose“ zu finden.

Mit Hilfe dieser Methoden werden pro Kamera und pro Kalibrierkörperaufnahme die Marken zur Kalibrierung extrahiert und die Pose ${}^{cn}P_{om}$ geschätzt.

Für eine Schätzung der Pose von Kamera 1 zu Kamera 2 wird bei einer beliebigen Aufnahme m die Pose aus einer Transformationskette berechnet. Diese ist in (62) gezeigt.

$$\overset{c2}{c1}H = \overset{c2}{om}H \overset{om}{c1}H \quad (62)$$

$$\overset{c2}{c1}H = \text{Hom}(\overset{c2}{om}P) \text{Hom}(\overset{c1}{om}P)^{-1} \quad (63)$$

$$\overset{c2}{c1}P = \text{Pose}(\text{Hom}(\overset{c2}{om}P) \text{Hom}(\overset{c1}{om}P)^{-1}) \quad (64)$$

6.2. Aufbau der grundsätzlichen Kalibrierung

Die Nutzung des Levenberg-Marquardt-Verfahrens setzt eine zu optimierende Funktion und deren Jacobi-Matrix voraus. Weiterhin muss der Parametersatz definiert werden.

Als Parameter werden hier die in Abschnitt 5.3 vorgestellten Parameter verwendet.

Ziel der Optimierung ist die Vorhersage der Mittelpunkte der Kalibriermarken. Demnach entsteht pro Kalibriermarke pro Bild ein Zeilen- und ein Spaltenfehler.

Dieser Fehler ist die Rückgabe der zu optimierenden Funktion

Für ein Bild m mit einer Kalibriermarke k bei $\overset{om}{p}_k$ lässt sich somit die zu optimierende Funktion wie folgt definieren:

$$\begin{aligned} \text{onePoint}(\text{par}_{\text{stereoSetup}}) &= \begin{pmatrix} \Delta^{c1}r_{m,k} \\ \Delta^{c1}c_{m,k} \\ \Delta^{c2}r_{m,k} \\ \Delta^{c2}c_{m,k} \end{pmatrix} \\ &= \text{stereoSetup}(\overset{c1}{v}, \overset{c1}{cp_{c1}}, \overset{c1}{cp_{c2}}, \overset{c2}{c1}P, m, \overset{om}{p}_k) - \begin{pmatrix} \overset{c1}{r}_{m,k}^{in} \\ \overset{c1}{c}_{m,k}^{in} \\ \overset{c2}{r}_{m,k}^{in} \\ \overset{c2}{c}_{m,k}^{in} \end{pmatrix} \end{aligned} \quad (65)$$

Die extrahierten Mittelpunkte $(r_{m,k}^{in}, c_{m,k}^{in})$ bleiben konstant. Somit unterscheiden sich J_{onePoint} und $J_{\text{stereoSetup}}$ nicht.

Für mehrere Bilder mit mehreren Kalibriermarken muss die Funktion onePoint für alle Kalibriermarken aller Bilder ausgeführt werden. Folgend müssen die einzelnen Fehler Δ und dazu entsprechende Jacobi-Matrizen zu einem gemeinsamen Fehlervektor und einer gemeinsamen Jacobi-Matrix zusammengefügt werden.

Hier wurde die finale zu optimierende Funktion wie folgt definiert:

$$\text{allPoints}(\text{par}_{\text{stereoSetup}}) = \begin{pmatrix} \Delta^{c1}r_{1,1} \\ \Delta^{c1}c_{1,1} \\ \vdots \\ \Delta^{c1}c_{1,K} \\ \Delta^{c2}r_{1,1} \\ \vdots \\ \Delta^{c2}c_{1,K} \\ \Delta^{c1}r_{2,1} \\ \vdots \\ \Delta^{c2}c_{M,K} \end{pmatrix} \quad (66)$$

Die Reihenfolge der Zeilen in der gemeinsamen Jacobi-Matrix muss dem Fehlervektor angeglichen werden.

6.3. Ausschluss von Parametern

Wie bereits in Abschnitt 5.2.2 gezeigt wurde, bestehen Unbestimmtheiten im Kamera-modell. Da diese bei λ nahe 0 zu einer singulären Matrix A führen, sollten einige Parameter festgehalten werden.

Ähnliches gilt, wenn man von außen eine sichere Information einfließen lassen will. Ein Festhalten eines Parameters des gesamten Modells lässt sich durch ein Herauslöschen von Elementen aus dem Parametervektor und der Spalten aus der Jacobi-Matrix bewerk-stelligen.

Konstant gehaltene Parameter sind immer noch Parameter des Stereo-Aufbaus. Allerdings sind diese keine Parameter der Optimierung.

Zur Vermeidung der Unbestimmtheiten des Kameramodells werden die Parameter S_x und S_y der Kameras von dem Benutzer explizit gesetzt und von der Optimierung ausge-schlossen. Abweichungen von den realen Werten für S_x und S_y können durch die anderen Parameter kompensiert werden.

6.4. Vermeidung von lokalen Minima

Selbst nach der Eliminierung der Unbestimmtheiten besitzt die Fehlerfunktion mehrere Minima. Mit Hilfe des Levenberg-Marquardt-Verfahrens können die Startparameter so lange variiert werden, bis ein lokales Minimum gefunden wurde. Ob dieses Minimum auch das globale Minimum ist, kann nicht bestimmt werden.

Aus diesem Grund werden mehrere Optimierungen unter verschiedenen Bedingungen durchgeführt und das Ergebnis mit dem minimalen Fehler zurückgegeben. Folgend sind einige Möglichkeiten für die Änderungen der Bedingungen aufgeführt:

- Startparameter zufällig variieren.
- Einzelne Parameter vorerst festhalten und in einer zweiten Optimierung mit frei geben.
- Einzelne Parameter oder Parametergruppen in definierten Stufen abtasten und als Startbedingung vorschreiben.

Die zufällige Variation der Parameter sollte aber im industriellen Umfeld nicht eingesetzt werden. Zwar ist bei einer steigenden Anzahl an Versuchen das Auffinden des globalen Minimums immer wahrscheinlicher, doch sind die Ergebnisse nicht reproduzierbar.

Auch ist das definierte Abtasten von Parametern nur mit wenigen Parametern effizient möglich.

Aus diesem Grund wurden folgende Vorgehensweisen implementiert (ausgehend von ei-nem allgemeinen Ausschluss von S_x und S_y , wie er sich aus Abschnitt 5.2.2 und 6.3 ergibt):

1. Alle Parameter optimieren.
2. Kameraparameter fix halten und vorerst nur Posen und Bewegungsvektor optimieren, darauf folgend alle Parameter optimieren.
3. Erst ohne die Kameraparameter f optimieren, danach alle Parameter optimieren.
4. Erst ohne die Kameraparameter C_y optimieren, danach alle Parameter optimieren.
5. α und β der Posen von Kalibrierkörper zu Kamera 1 auf 0 setzen, dann alle Parameter optimieren.
6. Optimieren mit wenigen Bildern, rekonstruieren aller Kalibrierkörper-Posen, optimieren aller Parameter.

Mit Hilfe von Methode 5 werden die Posen so ausgerichtet, dass die Kamera 1 senkrecht auf den Kalibrierkörper schaut. Dies kann bei der Umgehung lokaler Minima auf Grund ungünstig initialisierter Posen hilfreich sein.

Methode 6 greift noch weiter in die Parameter der Kalibrierkörper-Posen ein. Die Voraussetzung hierbei ist, dass eine gute Lösung mit wenigen Bildern gefunden wird. Die Güte hierbei bezieht sich nicht auf das Fehlermaß der Optimierung, sondern auf die Rekonstruktionsfehler der Marken. Mit Hilfe des Bewegungsvektors, den Kameraparametern und der Transformation zwischen den Kameras aus dem Ergebnis der Optimierung mit wenigen Bildern können die Posen zu allen Kalibrierkörpern geschätzt werden.

Durch die extrahierten Mittelpunkte und die Kameraparameter lassen sich die Strahlen im 3D-Raum berechnen. Da das Modell und die extrahierten Mittelpunkte fehlerhaft sind, haben die Strahlen keinen Schnittpunkt. Stattdessen wird das folgende Gleichungssystem gelöst:

$${}^{c1}p_{c1} + a {}^{c1}v_{cross} = {}^{c1}p_{c2} + b {}^{c1}v_{c2} + c {}^{c1}v_{cross} \quad (67)$$

$${}^{c1}v_{cross} = \frac{{}^{c1}v_{c1} \times {}^{c1}v_{c2}}{|{}^{c1}v_{c1} \times {}^{c1}v_{c2}|} \quad (68)$$

$${}^{c1}p_{out} = {}^{c1}p_{c1} + a {}^{c1}v_{cross} - \frac{c}{2} {}^{c1}v_{cross} \quad (69)$$

Semantisch dürfen die Richtungen der Strahlen aus den beiden Kamerabildern nicht linear abhängig sein. Somit ist auch das Kreuzprodukt ${}^{c1}v_{cross}$ der Richtungen definiert. c beschreibt den minimalen Abstand der Strahlen und somit den verbleibenden Fehler bei dem Verschneiden. Der resultierende Punkt ${}^{c1}p_{out}$ hat den minimalen Abstand zu beiden Strahlen.

Mit dieser Methode werden Punkte von dem Kalibrierkörper rekonstruiert. Diese werden so gewählt, dass zwei Punkte im Koordinatensystem des Kalibrierkörpers nur eine Verschiebung in x -Richtung aufweisen. Somit lässt sich die x -Achse ${}^{c1}v_{xAxis}$ des Kalibrierkörperkoordinatensystems rekonstruieren. Gleicher gilt für die y -Achse. Mit Hilfe des Kreuzproduktes lässt sich auch die z -Achse rekonstruieren. Falls eine Marke in dem

Ursprung des Kalibrierkörperkoordinatensystems liegt, kann diese direkt verwendet werden. Andernfalls kann der Ursprung ${}^{c1}p_0$ aus den bisher berechneten Punkten bestimmt werden.

In den Gleichungen (70) und (71) wird die Rekonstruktion der Pose gezeigt.

$${}^{c1}_oH = \begin{pmatrix} \frac{{}^{c1}v_{xAxis}}{{}^{c1}v_{xAxis}} & \frac{{}^{c1}v_{yAxis}}{{}^{c1}v_{yAxis}} & \frac{{}^{c1}v_{xAxis} \times {}^{c1}v_{yAxis}}{{}^{c1}v_{xAxis} \times {}^{c1}v_{yAxis}} & {}^{c1}p_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (70)$$

$${}^{c1}_oP = \text{Pose}({}^{c1}_oH) \quad (71)$$

Die so ermittelten Posen sollten sehr nahe an den realen Posen liegen und somit lokale Minima aufgrund ungünstig initialisierter Posen vermeiden.

6.5. Stabilität und Verringerung der Komplexität

Das Levenberg-Marquardt-Verfahren basiert auf der Invertierung von A . Weiterhin wird A näherungsweise durch $J^T J$ angenähert. Da die Größenordnungen der Werte in der Jacobi-Matrix sich stark unterscheiden, kann dies zu numerischen Problemen führen. Aus diesem Grund wurde A in der Implementierung, wie in [Mar63] vorgeschlagen, normiert. Ausgehend von einem Erwartungswert von 0 innerhalb jeder Spalte der Jacobi-Matrix werden diese anhand ihrer Standardabweichung normiert. Dies führt dazu, dass die Matrix A ohne Augmentierung auf der Hauptdiagonalen nur Einsen besitzt. Eine Skalierung der Spalten der Jacobi-Matrix wirkt sich auch auf die Vektoren h und Δ aus. Vor der Anpassung der Parameter durch Δ muss die Normierung rückgängig gemacht werden.

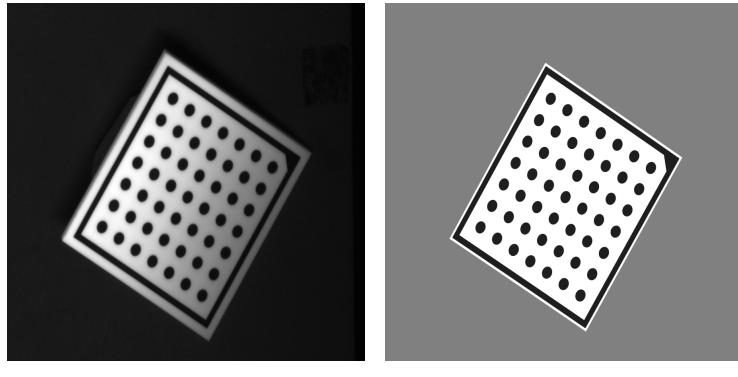
Die Invertierung der gesamten Matrix A kann auch vermieden werden. Dafür lässt sich die Struktur der Jacobi-Matrix ausnutzen.

$$J = \begin{pmatrix} p^1 J_v & p^1 J_{cp} & p^1 J_{c1P} & p^1 J_{c1P} & 0^{* \times 6} & \dots & 0^{* \times 6} \\ p^2 J_v & p^2 J_{cp} & p^2 J_{c1P} & 0^{* \times 6} & p^2 J_{c1P} & \dots & 0^{* \times 6} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p^M J_v & p^M J_{cp} & p^M J_{c1P} & 0^{* \times 6} & 0^{* \times 6} & \dots & p^M J_{c1P} \end{pmatrix} \quad (72)$$

Wie in [HZ04] gezeigt, kann in dieser Struktur das Problem auf M Invertierungen von 6×6 -Matrizen und eine Cholesky-Zerlegung des verbleibenden vollbesetzten Teils der Matrix A reduziert werden.

Ein weiterer Punkt, an dem das Verfahren modifiziert werden kann, ist die Anpassung von λ . Das Anpassen von λ um den Faktor 10 pro Iteration führt dazu, dass schnell eine geeignete Größenordnung gefunden wird. Wurde diese gefunden, kann das Anpassen nochmalige Anpassen von λ dazu führen, dass λ über mehrere Iterationen zwischen zwei Werten wechselt. Somit wird jede zweite Berechnung verworfen.

Betrachtet man die Historie von λ , kann man diese einbeziehen, um die Konvergenz zu beschleunigen. Beispielsweise kann man bei einer erfolgreichen Iteration auf das Verringern von λ verzichten, falls es in der vorherigen Iteration erhöht und davor nicht erhöht wurde.



(a) Aufnahme

(b) Simulation

Abbildung 18: Aufnahme und Simulation einer Kalibrierplatte. Eine passende Simulation setzt eine erfolgreiche Kalibrierung voraus.

Zusätzlich dazu könnte λ innerhalb der Bereiche, in denen sich Annahme und Ablehnung abwechseln, um einen kleineren Faktor variieren. Nachdem die Größenordnung für λ gefunden wurde, wird bei erfolgreichen Iterationen λ nur noch um den kleineren Faktor verringert. Wurde in der letzten Iteration das Ergebnis angenommen und das Ergebnis aus dieser Iteration verworfen, wird λ mit dem kleineren Faktor multipliziert. Bei einer abweichenden Historie wird bei einem verworfenen Ergebnis der große Faktor aufgeschlagen.

6.6. Fehlerkorrektur der Mittelpunktextraktion

Die Extraktion der Mittelpunkte, welche in den vorbereitenden Schritten zur Kalibrierung durchgeführt wurde, kann leicht fehlerbehaftet sein. Optional kann deswegen versucht werden, den Fehler in einer weiteren Optimierung auszugleichen. Die Ausgangsthese dazu ist, dass der Vektor vom realen Mittelpunkt zum extrahierten Mittelpunkt sich in realen und simulierten Daten ähneln.

Basierend auf dem Ergebnis der vorherigen Kalibrierung sind die Kameraparameter und Posen der Kalibrierkörper bis auf einen geringen Fehler bekannt. HALCON bietet nun die Möglichkeit unter Vorgabe des Bewegungsvektors, der Kameraparameter und der Pose von Kalibrierkörper zu Kamera ein Bild zu simulieren. Abbildung 18 zeigt ein solches reales Bild und ein mit den ermittelten Parameter rekonstruiertes Bild.

Genau wie mit den realen Bildern können auch aus den simulierten Bildern die Mittelpunkte der Marken ermittelt werden. Zu den simulierten Bildern kann dieser allerdings auch genau berechnet werden.

Für einen Ausgleich des Fehlers werden die Verschiebungen von extrahierten Mittelpunkten zu berechneten Mittelpunkten aus den simulierten Daten auf die extrahierten Punkte aus den realen Daten aufgeschlagen. Mit den so korrigierten Punkten wird eine erneute Kalibrierung vorgenommen. Auf die in Abschnitt 6.4 vorgestellten Verfahren zur Umgehung lokaler Minima kann hier verzichtet werden. Da die Startparameter schon

annähernd optimal sind und der Fehler bei der Extraktion der Mittelpunkte nur gering ist, können alle Parameter (ohne S_x und S_y) optimiert werden.

6.7. Erweiterung auf N Kameras

Das hier vorgestellte Verfahren kann einfach auf ein Setup mit N Kameras erweitert werden. Der Ausgangspunkt dafür ist das Modell für N Kameras, welches im Abschnitt 5.4 vorgestellt wurde. Die Definition des Parametervektors wird direkt übernommen. Die Jacobi-Matrix und die Einträge des Fehlervektors sind so aufgebaut, dass die Zeilen/Elemente in drei geschachtelten Schleifen gefüllt werden können. Die äußere Schleife läuft über die Aufnahmen der Kalibrierkörper. In der mittleren Schleife wird über die N Kameras iteriert. Die innere Schleife indiziert die einzelnen Marken auf den Kalibrierkörpern.

Auf diese Art ist eine Implementierung einfach durchführbar. Mit einem Einführen der $c_1^1 P$ als Einheitspose kann die Kamera 1 auch wie die anderen Kameras betrachtet werden.

Zu beachten ist allerdings, dass das in Abschnitt 6.4 unter 6 vorgestellte Verfahren mit der Rekonstruktion der Kalibrierkörper-Posen mit $N = 1$ Kameras nicht möglich ist. Bei der Erweiterung auf $N > 3$ Kameras muss eine Möglichkeit gefunden werden, den Punkt mit minimalen Abstand zu allen Strahlen zu finden.

7. Ansatz zur Rekonstruktion

Das Ziel eines Stereo-Aufbaus ist meist die Rekonstruktion von Objekten anhand von verschiedenen Aufnahmen. Durch das in Abschnitt 5.3 vorgestellte Modell wird ein Stereo-Aufbau mit Zeilenkameras definiert. Mit Hilfe der in Abschnitt 6 vorgestellten Kalibrierung können die dazugehörigen Parameter bestimmt werden.

Die Grundlage der Rekonstruktion ist die Epipolargeometrie und die sich damit ergebende Epipolarbedingung (siehe Abschnitt 4.1). Diese sagt aus, dass bei der Suche nach einer Korrespondenz eines Punktes aus dem Bild der ersten Kamera nur entlang der Epipolarkurve in dem Bild der zweiten Kamera gesucht werden muss.

Um Korrespondenzen zu finden, wird ein Ausschnitt aus dem ersten Bild mit verschiedenen Ausschnitten entlang der Epipolarkurve aus dem zweiten Kamerabild verglichen. Dieser Prozess wird im Folgenden Matching genannt. Für ein Vergleichen der Ausschnitte können verschiedene Methoden verwendet werden. Beispiele hierfür sind die Summe der absoluten Differenzen, die Summe der quadratischen Differenzen und die normalisierte Kreuzkorrelation [MVT13b]. Die Rechnungen dazu sind im Anhang A.3 beschrieben. Bei einer Rekonstruktion mit Flächenkameras ist ein Rektifizieren der Bilder möglich. Dies ist auch bei zwei Spezialfällen mit Zeilenkameras möglich. Diese Fälle, der ideale Along-Track-Fall und der ideale Across-Track-Fall, sind in Abschnitt 5.1 beschrieben. Für allgemeine Fälle kann aber nur von der Epipolarkurve direkt ausgegangen werden. Diese beiden Möglichkeiten der Rekonstruktion sind folgend weiter ausgeführt.

7.1. Matching anhand von angenäherten rektifizierten Bildern

Das Vorgehen für die Rekonstruktion anhand der Methoden aus der Stereo-Vision mit Flächenkameras unterscheidet sich bei den Spezialfällen.

In einer idealen Across-Track-Konfiguration entspricht die Geometrie der eines Aufbaus mit Flächenkameras.

Bei einem Abweichen von einer idealen Kamera ist dieses Vorgehen aber nur möglich, wenn die Aufnahmeflächen der Kameras einer Ebene gleichen. Abbildung 12 zeigt eine Konfiguration, bei der die Aufnahmefläche gebogen ist. Dies tritt auf, wenn κ und $S_y C_y$ von Null abweichen. Somit sind die Aufnahmeflächen nur annähernd parallel und es muss mit verbleibenden Fehlern gerechnet werden.

Ist κ oder $S_y C_y$ bei beiden Kameras sehr nahe oder gleich Null, lassen sich Zeilen der verschiedenen Kamerabilder einander zuordnen. Die Algorithmen für die Rekonstruktion gehen von idealen Kameras aus. Diese Kameras sind gleich ausgerichtet ($c^1 z \parallel c^2 z$). Weiterhin müssen die x -Achsen der Kamerakoordinatensysteme parallel zur Verbindungsline zwischen den Projektionszentren sein ($c^1 x \parallel c^2 x \parallel b$). Die Brennweite beträgt für beide idealen Kameras f_0 .

Nachdem die Zeilen der realen Kameras auf die virtuellen Bildzeilen projiziert wurden, können Tiefen rekonstruiert werden. Für jeden Punkt in einem Bild kann entlang der zugehörigen Zeile im anderen Bild eine Korrespondenz gesucht werden.

Die Differenz der Spalten in den transformierten Bildern $p^1 c$ und $p^2 c$, in welcher die Korrespondenzen gefunden wurden, bezeichnet man als Disparität. Aus der Disparität

kann die ${}^{c1'}z$ -Koordinate rekonstruiert werden. Gleichung (73) zeigt die Rechnung, wenn die Position des Hauptpunktes (C_x) bei beiden Kameras gleich ist.

$${}^{c1'}p_z = f_0 \frac{b_x}{\frac{p1'c - p2'c}{p1'c}} \quad (73)$$

Diese Vorgehensweise ist in Standardwerken zur Stereo-Vision ausführlich beschrieben. An dieser Stelle sei auf [HZ04] verwiesen.

Eine Along-Track-Konfiguration weicht geometrisch von Aufbauten mit Flächenkameras ab. Hierbei verlaufen die Epi polarkurven entlang der y -Achse der Bilder. Durch ein Rotieren der Bilder um 90° kann die Vorgehensweise von Stereo-Vision mit Flächenkameras angewendet werden. Die Geometrie, und somit die Berechnung der Tiefe aus der Disparität, unterscheidet sich allerdings. Diese ist in Abbildung 19 gezeigt.

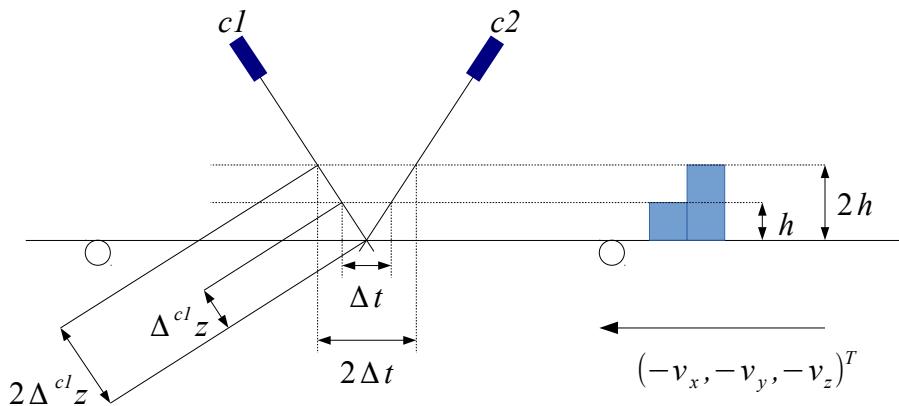


Abbildung 19: Tiefenrekonstruktion in einer Along-Track-Konfiguration.

Zwei korrespondierende Spalten (der nicht gedrehten Bilder) nehmen zur gleichen Zeit einen Punkt in einer Tiefe (${}^{c1}z$ – Achse) von ${}^{c1}z_0$ auf. In dieser Tiefe würden Korrespondenzen in den beiden Bildern in den gleichen Zeilen liegen. Weicht die Tiefe davon ab, werden die Korrespondenzen zu verschiedenen Zeiten aufgenommen. Somit entsteht die Disparität als Differenz der Zeilenkoordinaten. Diese Differenz verhält sich proportional zur Abweichung von ${}^{c1}z_0$. Folglich lässt sich die Tiefe durch ${}^{c1}z = a(p^2r - p^1r) + {}^{c1}z_0$ berechnen. Der Faktor a wird durch die Geschwindigkeit der Bewegung und der Richtung der Strahlen bestimmt.

Abhängig von den Kameraparametern können die Werte a und ${}^{c1}z_0$ von Spalte zu Spalte variieren. Im Gegensatz zum Across-Track-Fall besteht im Along-Track-Fall für dieses Vorgehen keine Einschränkung der Kameraparameter.

7.2. Matching anhand von Epipolarkurven

Eine Rekonstruktion kann auch auf allgemeine Konfigurationen angewendet werden. Wie bereits in Abschnitt 5.2 beschrieben, ist die Form einer Epipolarkurve in einer allgemeinen Konfiguration mit idealen Zeilenkameras und einer linearen Bewegung eine Hyperbel. Weiterhin ist die Form der Hyperbel abhängig von der Spalte, zu welcher die Epipolarkurve gesucht wird. Dies verhindert eine Rektifizierung, welche für das in Abschnitt 7.1 vorgestellte Verfahren elementar ist.

Da die Epipolarbedingung bestehen bleibt, werden hier die Epipolarkurven direkt verwendet, um zu einem Ausschnitt im ersten Bild Ausschnitte im zweiten Bild zu erstellen. Für einzelne Pixel aus dem ersten Bild kann mit Hilfe der Epipolarkurve die erwartete Koordinate innerhalb des zweiten Bildes für verschiedene Tiefen bestimmt werden.

Für einen Vergleich mit einem Ausschnitt in der Tiefe $c^1 z_d$ wird anhand der Epipolarkurven zu jedem Pixel innerhalb des Ausschnittes im Bild der ersten Kamera die erwartete Position im Bild der zweiten Kamera ermittelt. Die Bildinformationen an diesen Stellen bilden den Inhalt des Vergleichsausschnittes.

Dem Ausgangspixel wird die Tiefe zugeordnet, in welcher von der Vergleichsfunktion die maximale Ähnlichkeit ermittelt wurde.

Abbildung 20 visualisiert das Vorgehen. In ihr sind oben die Bilder der beiden Kameras gezeigt. Die Aufgabe ist die Bestimmung der Tiefe $c^1 z$ zu dem markierten Punkt im linken Bild. Zu diesem ist im rechten Bild ein Ausschnitt der Epipolarkurve eingezeichnet. Mittig sind in der Abbildung Bildausschnitte für die folgenden Vergleiche dargestellt. Links ist der Ausschnitt um den markierten Punkt aus dem ersten Bild gezeigt. Weiterhin sind rechts exemplarisch drei Ausschnitte für verschiedene Tiefen dargestellt.

Jedes Pixel innerhalb des linken Ausschnittes wird einzeln über die zugehörige Epipolarkurve transformiert. Dies lässt sich durch die Richtung der Kante im ersten und die Form des Kreises im zweiten Ausschnitt erkennen.

Mit Hilfe der normalisierten Kreuzkorrelation werden die Ausschnitte für die verschiedenen Tiefen bewertet. Die Ergebnisse dazu sind im Diagramm unten in der Abbildung 20 gezeigt.

Hierbei stellt ein Wert von 1 die maximale Ähnlichkeit dar. Das beste Ergebnis wird mit dem zweiten Vergleichsausschnitt erreicht. Aus dem Diagramm folgt, dass die $c^1 z$ -Koordinate des Vergleichspunktes bei 0.394m liegt.

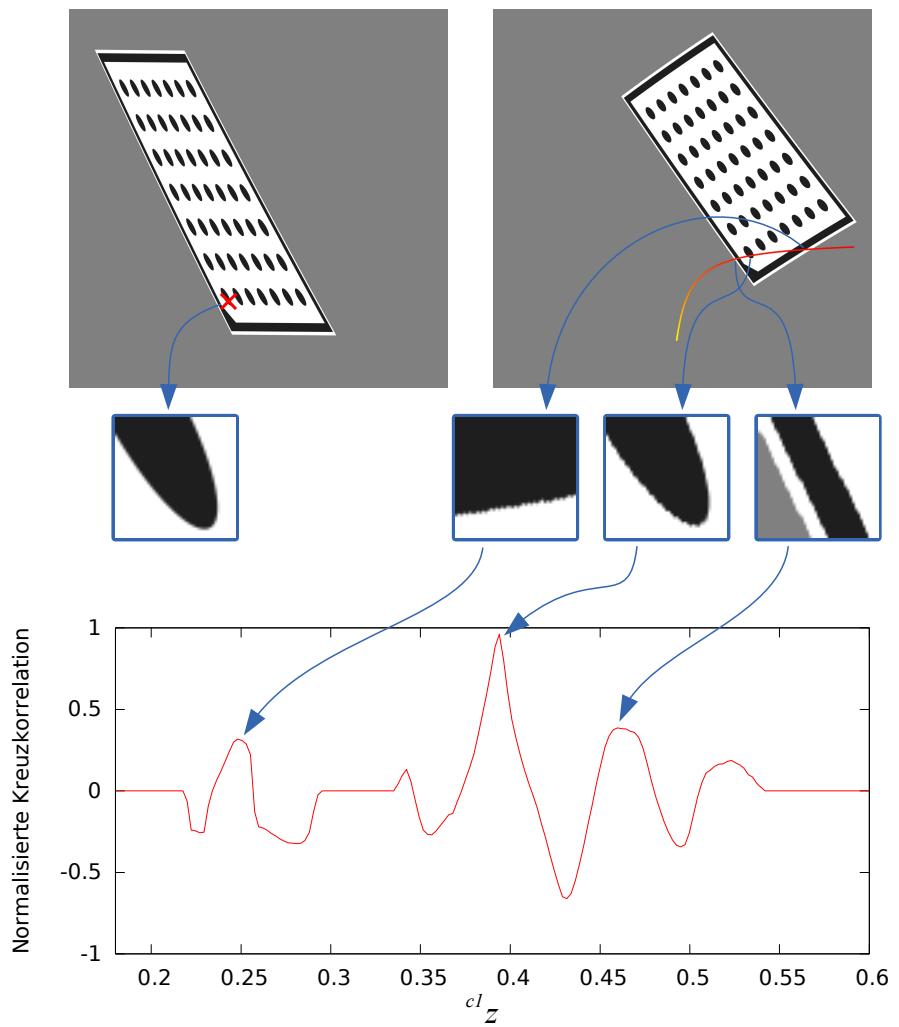


Abbildung 20: Matching auf Basis allgemeiner Epipolarkurven.

8. Ergebnisse

Nachfolgend werden die vorgestellten Verfahren zur Kalibrierung (Kapitel 6) und zur Rekonstruktion (Kapitel 7) angewendet und bewertet.

Der Ausgangspunkt für die Bewertung der Kalibrierung ist der Rückprojektionsfehler und die an der VDI/VDE-Richtlinie 2634 [VDI02] angelehnte Längenabweichung. Weiterhin wird der Fehler bei der Rekonstruktion der Punkte betrachtet, zwischen welchen die Abstände für die Längenabweichung berechnet werden.

Als Kalibrierkörper wurde für diese Arbeit eine $30mm \times 30mm$ große Kalibrierplatte mit 49 Marken in Form von schwarzen Kreisen auf weißem Hintergrund verwendet. Diese wird in Abbildung 6 gezeigt.

8.1. Vorgehen

Für die folgenden Ergebnisse wurden Daten simuliert oder aufgenommen. Bei der Kalibrierung wurden für jede Teiloptimierung maximal 500 Iterationen zugelassen. Sollte 20 Mal in Folge λ erhöht werden, endet eine Optimierung ebenfalls.

Die Rückprojektionsfehler ϵ_{out} zu den Kalibrierungen entstehen aus einer Normierung des quadratischen Fehlers ϵ_{sq} der Optimierung mit der Anzahl der Kameras N , der Anzahl der Kalibrieraufnahmen M und der Anzahl der Marken auf dem Kalibrierkörper P . Die Rechnung ist in Gleichung (74) gezeigt.

$$\epsilon_{out} = \sqrt{\frac{\epsilon_{sq}}{MNP}} \quad (74)$$

Die Kalibrierbilder werden jeweils in drei Datensätze unterteilt. Pro Testfall werden vier Kalibrierungen durchgeführt. Dabei werden einmal alle Datensätze ($1 \cup 2 \cup 3$) und weiterhin Kombinationen aus zwei Datensätzen ($1 \cup 2$, $1 \cup 3$ und $2 \cup 3$) verwendet.

Für die Bewertung, angelehnt an die VDI/VDE-Richtlinie 2634 [VDI02], werden Kalibrierungen mit zwei Datensätzen verwendet und die Punkte aus dem verbleibenden Datensatz für die Berechnung der Längenabweichung genutzt. Diese Vorgehensweise nennt sich „Cross Validation“. Sie stellt sicher, dass die Ergebnisse auf Daten beruhen, welche nicht in die Modellbildung eingeflossen sind.

Abbildung 21 zeigt das Muster, welches zur Berechnung der Längenabweichung verwendet wird. In diesem Muster werden die Ränder und die Diagonalen der Kalibrierplatte vermessen.

Für diese Berechnungen werden die Markenpunkte, wie in Abschnitt 6.4 bei Methode 6 beschrieben, rekonstruiert. Dabei werden zu den Bildpunkten der Kameras die Strahlen im 3D-Raum berechnet und der Punkt mit dem geringsten Abstand zu beiden Strahlen bestimmt. Neben der Verwendung der so rekonstruierten Punkte zum Berechnen der räumlichen Abstände und der Längenabweichung wird auch der Fehler bei der Rekonstruktion der Punkte betrachtet. Als Fehlermaß wird dabei der Abstand betrachtet, um welchen sich die Strahlen verfehlten.

Für die Rekonstruktion wurde immer von allgemeinen Epipolarkurven ausgegangen. Das Verfahren dazu wurde in Abschnitt 7.2 vorgestellt. Als Vergleichsfunktion wurde die

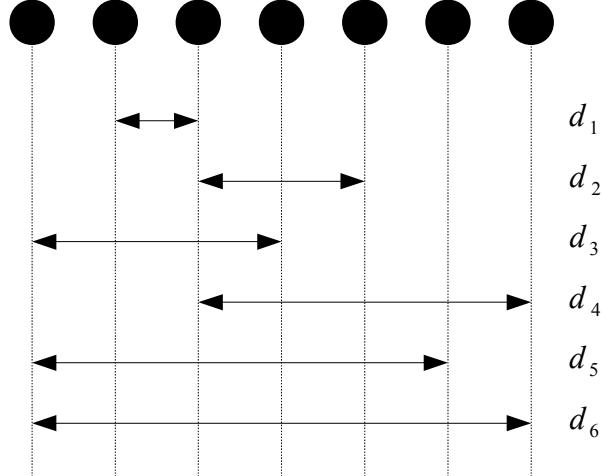


Abbildung 21: Definition der Abstände zur Berechnung der Längenabweichung.

normalisierte Kreuzkorrelation (siehe Anhang A.3) eingesetzt. Diese liefert Ausgaben im Bereich von -1 bis 1 . Dabei ist 1 die maximale Ähnlichkeit.

8.2. Simulierte Daten

Die Kalibrierung ist ein Teilprozess, welchem die Aufnahmen des Kalibrierkörpers und die Extraktion der Marken voranstehen. Durch die Nutzung simulierter Daten werden in den folgenden Testfällen Fehler bei den Aufnahmen und Fehler bei der Extraktion der Marken ausgeschlossen. Da die simulierten Daten zusätzlich auf dem verwendeten Modell (siehe Abschnitt 5.3.2) basieren, sind auch Fehler ausgeschlossen, welche nicht parametrisiert werden können. Für die Erzeugung der Daten wurden Simulationsparameter vorgegeben und die Mittelpunkte der Kalibriermarken in die Bildebenen der Kameras projiziert.

Folgend werden 3 Testfälle betrachtet.

SIMALONG simulierte eine ideale Along-Track-Konfiguration.

SIMACROSS simulierte eine ideale Across-Track-Konfiguration.

SIMFREE simulierte eine von den idealen Fällen abweichende Konfiguration.

In den Testfällen werden jeweils 6 Aufnahmen von Kalibrierkörpern simuliert. Die simulierten Kameras besitzen eine Verzeichnung und einen von $(\frac{1}{2} \text{Width}, 0)$ abweichenden Hauptpunkt. Die Startparameter der Kameras sind frei von Verzeichnungen und der Hauptpunkt liegt mittig auf dem Sensorarray bei $(\frac{1}{2} \text{Width}, 0)$. Weiterhin wurden für die ersten 4 Aufnahmen die Startparameter der Posen leicht verändert. Die Posen der verbleibenden Aufnahmen wurden stark variiert. Weitere Parameter (der Bewegungsvektor und die Pose zwischen den Kameras) wurden nur leicht abgeändert.

Für die Rekonstruktion wurden mit Hilfe von HALCON aus den Simulationsparametern Aufnahmen von Kalibrierplatten simuliert.

8.2.1. Kalibrierung

Bei der Kalibrierung ist das Umgehen der lokalen Minima von größtem Interesse. Hierfür werden die in Abschnitt 6.4 beschriebenen Methoden verwendet.

In Tabelle 2 werden die Ergebnisse der Herangehensweisen für den SIMALONG-Fall miteinander verglichen.

Wie an den Rückprojektionsfehlern in Tabelle 2 bei allen Bildern ($1 \cup 2 \cup 3$) erkennbar ist, bleiben einige Methoden in lokalen Minima hängen. Nur die Methoden, welche in die Posen eingreifen, finden das globale Minimum. Auch wenn die Methoden (1) und (2) - (4) zu ähnlichen Fehlern führen, variieren die Parameter deutlich. In der gegebenen Aufteilung endeten diese in zwei verschiedenen lokalen Minima.

Dagegen wird in dem Fall ohne die letzten zwei Bilder ($1 \cup 2$) von mehreren Vorgehensweisen das globale Minimum gefunden. In den verbleibenden Fällen sind wieder die schlecht initialisierten Posen des Datensatzes 3 mit in der Kalibrierung, so dass Methoden (1) - (4) das Minimum nicht finden.

Am robustesten verhält sich hier das Verfahren (6). Für dieses wurden aber die Bilder zur ersten Näherung der Parameter (vgl. Abschnitt 6.4) händisch ausgewählt. Hierfür wurden die ersten 4 Bilder verwendet.

Die Fehler verhalten sich auch in den anderen Testfällen ähnlich. Diese sind in den Tabellen 3 und 4 gezeigt. Auch hier führt Methode (6) immer zu einem guten Ergebnis.

Verfahren	Fehler $1 \cup 2 \cup 3$	Fehler $1 \cup 2$	Fehler $1 \cup 3$	Fehler $2 \cup 3$
Start	$2.364558 \cdot 10^{+2}$	$2.758526 \cdot 10^{+2}$	$1.272317 \cdot 10^{+2}$	$2.746843 \cdot 10^{+2}$
(1) Alle Parameter	$2.352849 \cdot 10^{+1}$	$5.153568 \cdot 10^{-7}$	$2.347808 \cdot 10^{+1}$	$2.235880 \cdot 10^{+1}$
(2) Pose zuerst	$2.324097 \cdot 10^{+1}$	$1.449005 \cdot 10^{-6}$	$2.318003 \cdot 10^{+1}$	$2.175120 \cdot 10^{+1}$
(3) zuerst ohne f	$2.324094 \cdot 10^{+1}$	$5.440411 \cdot 10^{-7}$	$2.318003 \cdot 10^{+1}$	$2.370675 \cdot 10^{+1}$
(4) zuerst ohne c_y	$2.324097 \cdot 10^{+1}$	$6.537388 \cdot 10^{-7}$	$2.415645 \cdot 10^{+1}$	$2.175257 \cdot 10^{+1}$
(5) α und β auf 0	$6.941054 \cdot 10^{-7}$	$1.641906 \cdot 10^{+2}$	$7.758327 \cdot 10^{-7}$	$6.497073 \cdot 10^{-7}$
(6) Posen rekonstruieren	$5.773582 \cdot 10^{-7}$	$4.655025 \cdot 10^{-7}$	$6.431055 \cdot 10^{-7}$	$5.132458 \cdot 10^{-7}$
(7) weitere Iterationen	$5.550760 \cdot 10^{-7}$	$3.344116 \cdot 10^{-7}$	$6.430817 \cdot 10^{-7}$	$5.059510 \cdot 10^{-7}$

Tabelle 2: Rückprojektionsfehler der verschiedenen Methoden des Testfalls SIMALONG.

Verfahren	Fehler $1 \cup 2 \cup 3$	Fehler $1 \cup 2$	Fehler $1 \cup 3$	Fehler $2 \cup 3$
Start	$3.751870 \cdot 10^{+2}$	$3.875986 \cdot 10^{+2}$	$3.385085 \cdot 10^{+2}$	$3.968313 \cdot 10^{+2}$
(1) Alle Parameter	$2.038878 \cdot 10^{+1}$	$2.954792 \cdot 10^{-7}$	$1.867868 \cdot 10^{+1}$	$1.964039 \cdot 10^{+1}$
(2) Pose zuerst	$1.984172 \cdot 10^{+1}$	$2.698698 \cdot 10^{-7}$	$2.155525 \cdot 10^{+1}$	$1.859449 \cdot 10^{+1}$
(3) zuerst ohne f	$1.980668 \cdot 10^{+1}$	$1.454398 \cdot 10^{-6}$	$1.860773 \cdot 10^{+1}$	$1.899378 \cdot 10^{+1}$
(4) zuerst ohne c_y	$1.969652 \cdot 10^{+1}$	$7.867980 \cdot 10^{-8}$	$1.860929 \cdot 10^{+1}$	$1.879367 \cdot 10^{+1}$
(5) α und β auf 0	$4.910829 \cdot 10^{-7}$	$5.473795 \cdot 10^{-7}$	$7.581929 \cdot 10^{+1}$	$5.063426 \cdot 10^{-7}$
(6) Posen rekonstruieren	$1.033758 \cdot 10^{-7}$	$1.330069 \cdot 10^{-7}$	$1.487111 \cdot 10^{-7}$	$1.418803 \cdot 10^{-7}$
(7) weitere Iterationen	$1.033758 \cdot 10^{-7}$	$7.867980 \cdot 10^{-8}$	$1.487111 \cdot 10^{-7}$	$1.415811 \cdot 10^{-7}$

Tabelle 3: Rückprojektionsfehler der verschiedenen Methoden des Testfalls SIMACROSS.

Verfahren	Fehler $1 \cup 2 \cup 3$	Fehler $1 \cup 2$	Fehler $1 \cup 3$	Fehler $2 \cup 3$
Start	$7.069855 \cdot 10^{+2}$	$7.347378 \cdot 10^{+2}$	$6.408442 \cdot 10^{+2}$	$7.409213 \cdot 10^{+2}$
(1) Alle Parameter	$2.988082 \cdot 10^{+1}$	$6.944328 \cdot 10^{-7}$	$3.133387 \cdot 10^{+1}$	$2.734275 \cdot 10^{+1}$
(2) Pose zuerst	$2.988082 \cdot 10^{+1}$	$2.077478 \cdot 10^{-7}$	$3.129527 \cdot 10^{+1}$	$2.733474 \cdot 10^{+1}$
(3) zuerst ohne f	$2.988082 \cdot 10^{+1}$	$3.748354 \cdot 10^{-6}$	$3.129463 \cdot 10^{+1}$	$2.733334 \cdot 10^{+1}$
(4) zuerst ohne c_y	$2.988082 \cdot 10^{+1}$	$4.219072 \cdot 10^{-7}$	$3.129463 \cdot 10^{+1}$	$2.733332 \cdot 10^{+1}$
(5) α und β auf 0	$2.711422 \cdot 10^{+1}$	$1.390338 \cdot 10^{+2}$	$3.028510 \cdot 10^{+1}$	$5.900109 \cdot 10^{-7}$
(6) Posen rekonstruieren	$2.513665 \cdot 10^{-7}$	$5.798462 \cdot 10^{-7}$	$3.896538 \cdot 10^{-7}$	$3.928042 \cdot 10^{-7}$
(7) weitere Iterationen	$2.513665 \cdot 10^{-7}$	$1.999677 \cdot 10^{-7}$	$2.840018 \cdot 10^{-7}$	$3.928042 \cdot 10^{-7}$

Tabelle 4: Rückprojektionsfehler der verschiedenen Methoden des Testfalls SIMFREE.

8.2.2. Betrachtung der Längenabweichung

Wie beschrieben, wurden verschiedene Markenpositionen in dem 3D-Raum rekonstruiert und die Distanzen mit den erwarteten Distanzen verglichen.

Abbildung 22 zeigt das Ergebnis für den Testfall SIMALONG, kalibriert mit $2 \cup 3$. Die Distanzen wurden aus dem verbleibenden Datensatz 1 entnommen. Abgebildet ist hier neben den tatsächlichen Fehlern bei den erwarteten Längen auch der Durchschnittsfehler und die Standardabweichung. Weiterhin ist rechts das Verhalten der Distanz gezeigt, um welche sich die Strahlen bei dem Rekonstruieren der einzelnen Punkte verfehlt haben. Wie an der Skalierung der y -Achsen erkennbar ist, sind die Fehler sehr klein. Eine Rekonstruktion mit diesen Parametern ist demnach möglich.

Bei den weiteren Möglichkeiten, die Datensätze zum Kalibrieren und Rekonstruieren aufzuteilen, und auch bei den weiteren simulierten Testfällen bleiben die Ergebnisse ähnlich.

8.2.3. Rekonstruktion

Das allgemeine Rekonstruktionsverfahren wurde anhand von einer simulierten Kalibrierplatte getestet. Diese wurde in ein Bild mit $2048px \times 2048px$ abgebildet. Abbildung 23 zeigt ein simuliertes Bilderpaar. Deutlich zu sehen ist dabei die starke simulierte Verzeichnung der Objektive.

Hier wird versucht, diese Kalibrierplatte flächendeckend zu rekonstruieren.

Problematisch ist dies beispielsweise an dem markierten Punkt im linken Bild. Zu diesem ist ein Ausschnitt der Epipolarkurve im rechten Bild gezeigt. Dieser Ausschnitt beschreibt einen Unterschied von $37mm$ in der c1z -Koordinate.

In diesem Bereich wurde nach Korrespondenzen gesucht. Der hierzu gehörende Bildausschnitt ist in Abbildung 24a gezeigt. Als Fenstergröße wurde hier $45px \times 45px$ gewählt. Die zwei ähnlichsten Bildausschnitte entlang der Epipolarkurve aus dem rechten Bild sind in Abbildungen 24b und 24c gezeigt.

Der mittlere Bildausschnitt ist dabei die richtige Korrespondenz. Der rechte Bildausschnitt zeigt den Schnittpunkt mit dem Rahmen.

Die Kalibrierplatte liegt schief im Raum. Das Modell geht aber davon aus, dass innerhalb des Bildausschnitts die c1z -Koordinate konstant bleibt. Somit kommt es zu nicht model-

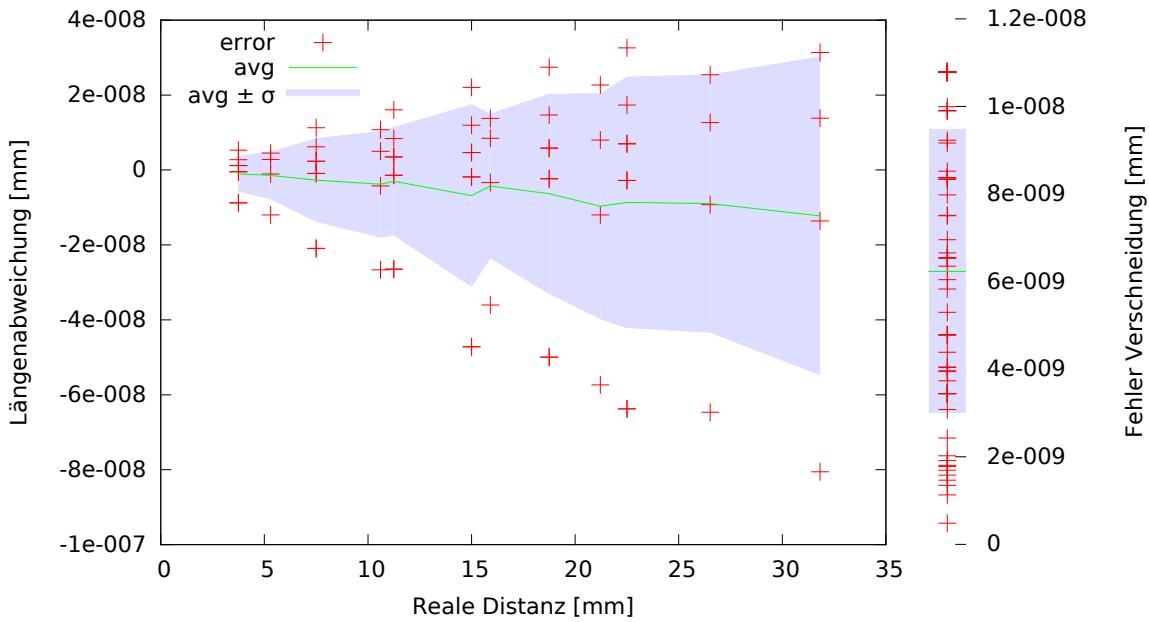


Abbildung 22: Längenabweichung des Testfalls SIMALONG, Datensatz 1.

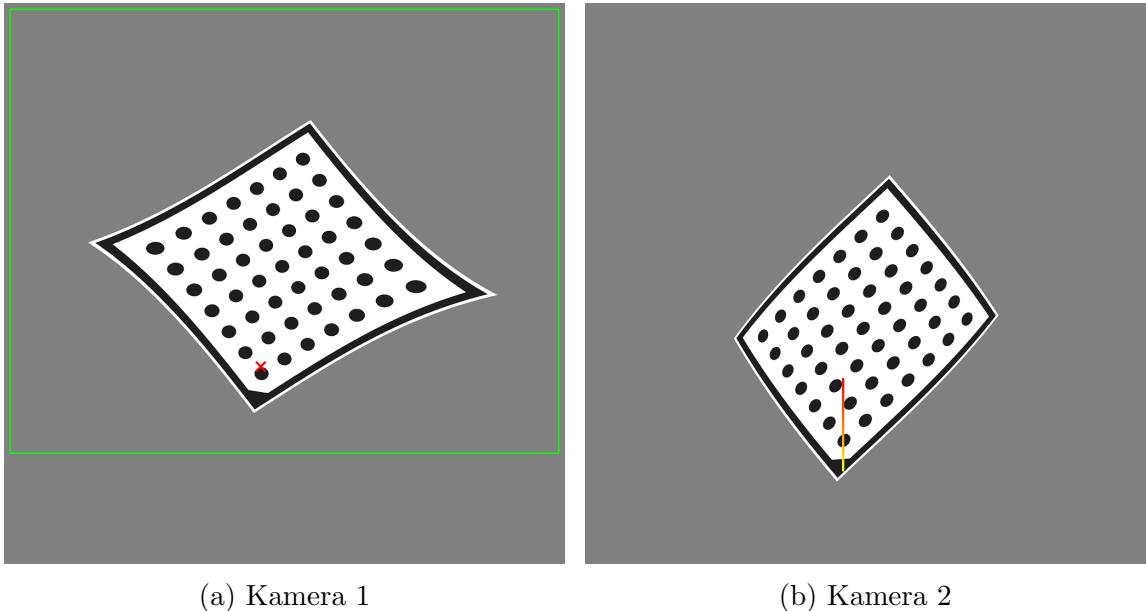


Abbildung 23: Simulierte Bilder aus dem Testfall SIMALONG. Zu dem markierten Punkt im linken Bild ist im rechten Bild ein Ausschnitt der Epipolarkurve gezeigt. Für eine Rekonstruktion werden nur Pixel innerhalb des Rahmens verwendet.



Abbildung 24: Ausschnitte der Korrespondenzsuche bei Testfall SIMALONG.

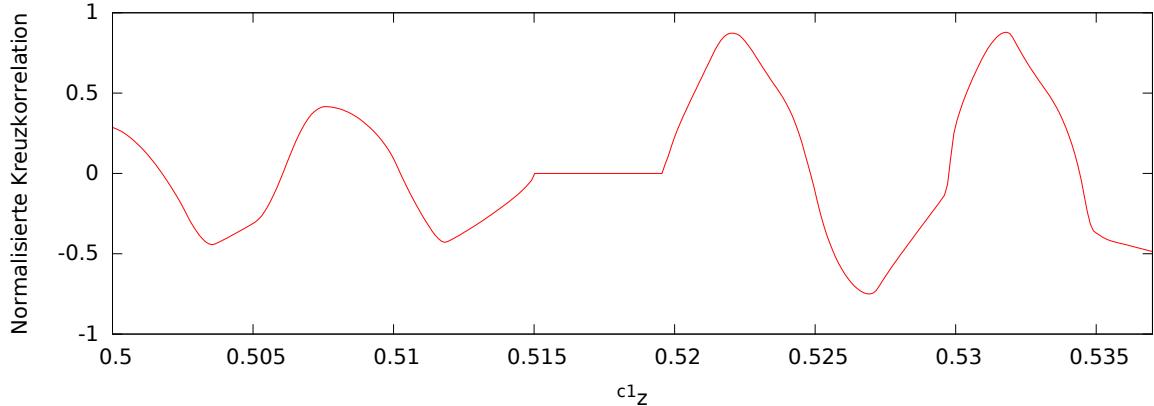


Abbildung 25: Verlauf der normalisierten Kreuzkorrelation in Abhängigkeit zur Tiefe in Testfall SIMALONG.

lierten Verzerrungen. In diesem Beispiel führt das dazu, dass beide Vergleichsausschnitte ähnliche Bewertungen nach der normalisierten Kreuzkorrelation erhalten.

Der Verlauf der normalisierten Kreuzkorrelation in Abhängigkeit zur Tiefe $c^1 z$ ist in Abbildung 25 gezeigt. Durch die zwei ähnlichen Maxima ist keine eindeutige Zuordnung möglich.

Für die flächendeckende Rekonstruktion wurde sich auf den Inhalt des grünen Rahmens in Abbildung 23a beschränkt. Zu den enthaltenen Pixeln können die Ausschnitte für das Matching erzeugt werden, ohne die Grenzen des rechten Bildes zu übertreten.

Abbildung 26 zeigt die ermittelte $c^1 z$ -Koordinate in Graustufen. Dabei wurde innerhalb des Rahmens aus Abbildung 23a Bereiches jedes fünfte Pixel in x - und y -Richtung verwendet. Farbige Pixel in der Ausgabe wurden ausgeschlossen. Der Grund hierfür ist entweder eine zu geringe Varianz der Grauwerte innerhalb des Ausschnittes (türkis) oder mehrere ähnlich gute Korrespondenzen (gelb). Ein Beispiel für ähnliche Korrespondenzen wurde in den Abbildungen 24 und 25 gezeigt. Hier kann dem Punkt keine klare Tiefe zugeordnet werden.

In Abbildung 27 sind die rekonstruierten kartesischen Koordinaten gezeigt. Hierbei ist erkennbar, dass die Verzeichnung in der Ausgabe kompensiert wurde.

In einer yz -Ansicht in Kalibrierkörperkoordinaten (Abbildung 28) ist der Fehler in der rekonstruierten Tiefe erkennbar. Bei diesem Testfall variiert dieser um maximal $\pm 0.5\text{mm}$.

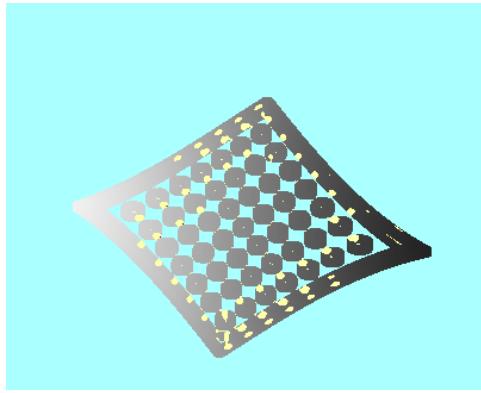


Abbildung 26: Rekonstruierte Tiefe der Kalibrierplatte als Graustufen des Testfalls SIMALONG.

Die verbleibenden Fehler entstehen durch die Annahme, dass die ^{c1}z -Koordinate innerhalb eines Vergleichsausschnittes konstant ist. Da die Kalibrierplatte, wie in Abbildung 26 erkennbar, aber deutlich in der ^{c1}z -Koordinate variiert, kommt es zu einem Nebeneffekt. Ein markantes Muster, wie beispielsweise der Übergang von Schwarz nach Weiß bei den Marken, wird in mehreren Ausschnitten abgetastet und auch in dem anderen Bild richtig zugeordnet. Den Abschnitten werden dabei ähnliche ^{c1}z -Koordinaten zugewiesen. Gültig ist diese allerdings nur, wenn der Übergang zwischen Schwarz und Weiß genau in der Mitte des Ausschnittes liegt. Die ähnlichen ^{c1}z -Koordinaten führen auf Grund der Verdrehung zu verschiedenen ^{o1}z -Koordinaten.

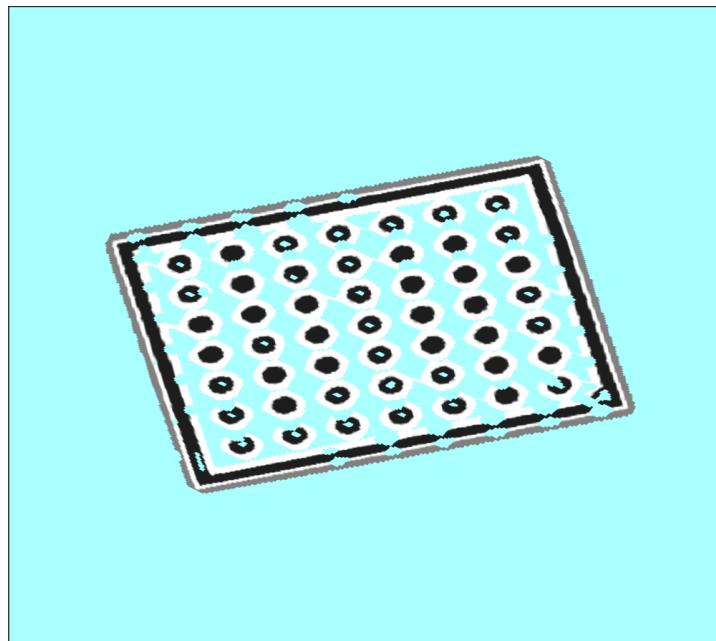


Abbildung 27: Rekonstruierte Fläche in 3D: Testfall SIMALONG.

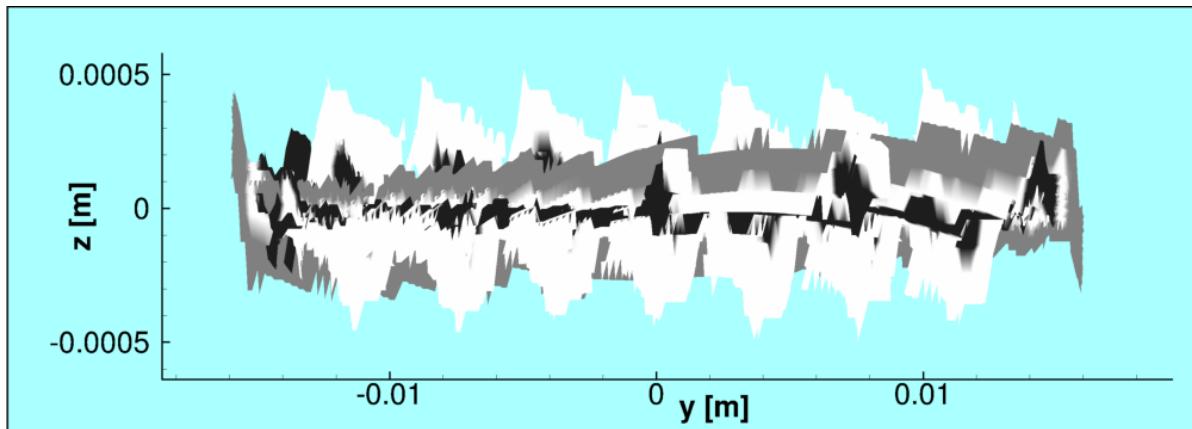


Abbildung 28: Rekonstruierte Fläche in 2D: Testfall SIMALONG.

8.3. Reale Daten

Die Vorgehensweise, welche bei den simulierten Daten angewandt wurde, soll hier auf realen Daten durchgeführt werden.

8.3.1. Versuchsaufbau

Für die Aufnahme der Daten wurde die in Abbildung 6 gezeigte $30\text{mm} \times 30\text{mm}$ Kalibrierplatte verwendet. Als Ersatz für ein industrielles Förderband erzeugte ein Linearschlitten die Bewegung. Zur Befestigung der Zeilenkameras wurden diese an einem Gerüst montiert. Eine zusätzliche Beleuchtungsquelle mit einer Zylinderlinse wurde verwendet, um die notwendige Helligkeit zu erreichen.

Als Kameras kamen Basler racer raL2048-48gm mit einer Zeilenauflösung von 2048 Pixeln zum Einsatz. Diese nehmen die Zeilen in Graustufen auf.

Abbildung 29 zeigt diesen Versuchsaufbau.

Mit diesem Setup wurden mehrere Testfälle aufgenommen. Diese sind in Tabelle 5 aufgeführt.

Abbildung 30 zeigt die Aufnahme einer Kalibrierplatte aus Testfall TEST7. In dieser Aufnahme wurde die Kalibrierplatte schräg gestellt. Somit unterscheiden sich die Abbildungen in den beiden Bildern deutlich.

8.3.2. Kalibrierung

Für jeden Testfall wurden 4 Kalibrierungen mit unterschiedlichen Datensätzen durchgeführt. Neben der eigentlichen Kalibrierung wurde der in Abschnitt 6.6 beschriebene Ansatz zur Korrektur des entstehenden Fehlers bei der Extraktion der Markenmittelpunkte in einer Iteration durchgeführt.

In Tabelle 6 sind die entstandenen Fehler aufgelistet. Zusätzlich wurde auch aufgelistet, welche aus den in Abschnitt 6.4 vorgestellten Verfahren zu vergleichbar guten Ergebnissen führen und welches Verfahren letztendlich verwendet wurde.

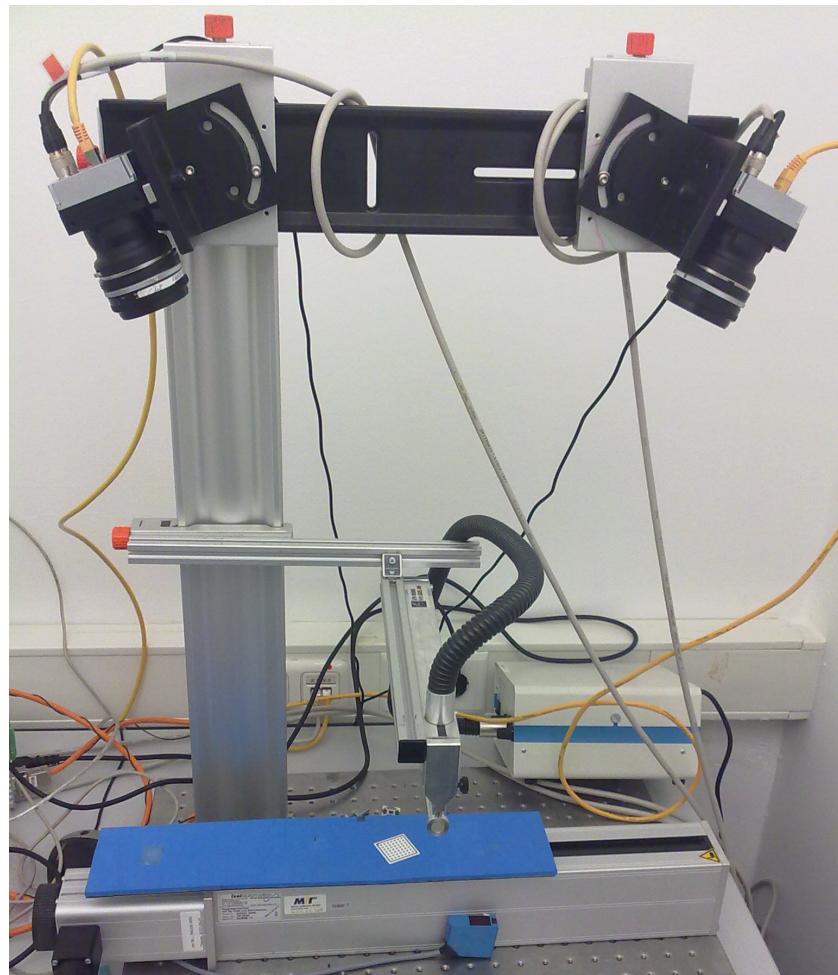


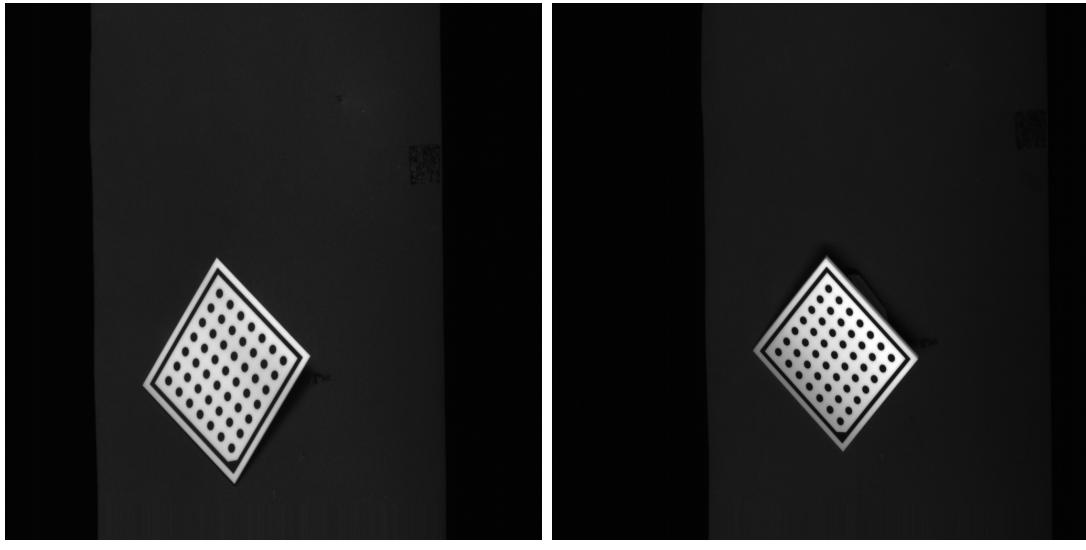
Abbildung 29: Versuchsaufbau

Testfall	Kalibrier- aufnahmen	Bildzeilen	Geschwindig- keit	Konfiguration
TEST1	14	1024	langsam	grob Along-Track
TEST2	14	1024	langsam	nahe Along-Track
TEST3	19	1024	langsam	grob Along-Track
TEST4	16	1024	langsam	stark abweichend
TEST5	18	1024	langsam	grob Across-Track
TEST6	79	4096	schnell	nahe Along-Track
TEST7	40	2048	mittel	nahe Along-Track

Tabelle 5: Testfälle für Aufnahmen von realen Daten.

Testfall	Datensätze	erfolgreiche Verfahren	Startfehler	Fehler	Fehler mit Mittelpunktausgleich
TEST1	$1 \cup 2 \cup 3$	(1),(2),(3),(4),(5),(6)	$2.882296 \cdot 10^{+1}$	$6.187207 \cdot 10^{-2}$	$6.185032 \cdot 10^{-2}$
	$1 \cup 2$	(1), <u>(2)</u> ,(3),(4),(5),(6)	$2.433470 \cdot 10^{+1}$	$6.564908 \cdot 10^{-2}$	$6.561800 \cdot 10^{-2}$
	$1 \cup 3$	(1),(2), <u>(3)</u> ,(4),(5),(6)	$2.130418 \cdot 10^{+1}$	$5.630446 \cdot 10^{-2}$	$5.609438 \cdot 10^{-2}$
	$2 \cup 3$	(1),(2),(3), <u>(4)</u> ,(5),(6)	$2.477566 \cdot 10^{+1}$	$6.472442 \cdot 10^{-2}$	$6.476494 \cdot 10^{-2}$
TEST2	$1 \cup 2 \cup 3$	(1), <u>(2)</u> ,(3),(4),(6)	$2.440928 \cdot 10^{+1}$	$7.093547 \cdot 10^{-2}$	$7.131228 \cdot 10^{-2}$
	$1 \cup 2$	(1), <u>(2)</u> ,(3),(4),(6)	$2.733073 \cdot 10^{+1}$	$7.276330 \cdot 10^{-2}$	$7.323637 \cdot 10^{-2}$
	$1 \cup 3$	(1), <u>(2)</u> ,(3),(4),(6)	$2.252000 \cdot 10^{+1}$	$6.938923 \cdot 10^{-2}$	$6.989377 \cdot 10^{-2}$
	$2 \cup 3$	(1), <u>(2)</u> ,(3),(4),(5),(6)	$2.802498 \cdot 10^{+1}$	$7.525355 \cdot 10^{-2}$	$7.546518 \cdot 10^{-2}$
TEST3	$1 \cup 2 \cup 3$	(1),(2), <u>(3)</u> ,(4)	$3.474985 \cdot 10^{+1}$	$1.049844 \cdot 10^{-1}$	$1.052021 \cdot 10^{-1}$
	$1 \cup 2$	(1),(2),(3),(4),(5)	$3.909778 \cdot 10^{+1}$	$6.056702 \cdot 10^{-2}$	$6.082659 \cdot 10^{-2}$
	$1 \cup 3$	(1),(2),(3), <u>(4)</u> ,(5)	$3.756790 \cdot 10^{+1}$	$1.145051 \cdot 10^{-1}$	$1.148378 \cdot 10^{-1}$
	$2 \cup 3$	(1),(3), <u>(5)</u>	$2.826573 \cdot 10^{+1}$	$1.211999 \cdot 10^{-1}$	$1.213678 \cdot 10^{-1}$
TEST4	$1 \cup 2 \cup 3$	(6)	$2.896521 \cdot 10^{+1}$	$1.019117 \cdot 10^{-1}$	$1.017857 \cdot 10^{-1}$
	$1 \cup 2$	(6)	$2.914255 \cdot 10^{+1}$	$9.011182 \cdot 10^{-2}$	$8.984061 \cdot 10^{-2}$
	$1 \cup 3$	(6)	$2.484246 \cdot 10^{+1}$	$1.021462 \cdot 10^{-1}$	$1.024511 \cdot 10^{-1}$
	$2 \cup 3$	(6)	$3.380984 \cdot 10^{+1}$	$9.207384 \cdot 10^{-2}$	$9.245920 \cdot 10^{-2}$
TEST5	$1 \cup 2 \cup 3$	(1),(2),(3),(4),(5),(6)	$2.267761 \cdot 10^{+2}$	$6.898575 \cdot 10^{-2}$	$6.919999 \cdot 10^{-2}$
	$1 \cup 2$	(1), <u>(2)</u> ,(3),(4),(5),(6)	$2.445456 \cdot 10^{+2}$	$7.073491 \cdot 10^{-2}$	$7.109359 \cdot 10^{-2}$
	$1 \cup 3$	(1),(2),(3),(4),(5),(6)	$2.065305 \cdot 10^{+2}$	$6.490975 \cdot 10^{-2}$	$6.516267 \cdot 10^{-2}$
	$2 \cup 3$	(1),(2),(3),(4),(5), <u>(6)</u>	$2.276505 \cdot 10^{+2}$	$6.779947 \cdot 10^{-2}$	$6.821667 \cdot 10^{-2}$
TEST6	$1 \cup 2 \cup 3$	(6)	$1.564119 \cdot 10^{+2}$	$1.241604 \cdot 10^{-1}$	$1.242766 \cdot 10^{-1}$
	$1 \cup 2$	(6)	$1.135477 \cdot 10^{+2}$	$1.334211 \cdot 10^{-1}$	$1.335341 \cdot 10^{-1}$
	$1 \cup 3$	(6)	$1.673420 \cdot 10^{+2}$	$1.317571 \cdot 10^{-1}$	$1.318599 \cdot 10^{-1}$
	$2 \cup 3$	(6)	$1.661015 \cdot 10^{+2}$	$9.714182 \cdot 10^{-2}$	$9.727147 \cdot 10^{-2}$
TEST7	$1 \cup 2 \cup 3$	(6)	$6.347376 \cdot 10^{+1}$	$6.511264 \cdot 10^{-2}$	$6.523670 \cdot 10^{-2}$
	$1 \cup 2$	(1), <u>(2)</u> ,(3),(4),(5), <u>(6)</u>	$3.526078 \cdot 10^{+1}$	$5.429342 \cdot 10^{-2}$	$5.558599 \cdot 10^{-2}$
	$1 \cup 3$	(6)	$7.001981 \cdot 10^{+1}$	$6.824031 \cdot 10^{-2}$	$6.834919 \cdot 10^{-2}$
	$2 \cup 3$	(6)	$7.625243 \cdot 10^{+1}$	$6.442345 \cdot 10^{-2}$	$6.457548 \cdot 10^{-2}$

Tabelle 6: Rückprojektionsfehler der Kalibrierungen mit realen Daten. Die unterstrichenen Vorgehensweisen wurden weiter verwendet.



(a) Kamera 1

(b) Kamera 2

Abbildung 30: Aufnahme einer Kalibrierplatte in Testfall TEST7.

Dabei wird oft mit mehreren Vorgehensweisen ein guter Parametersatz gefunden. Die wenigsten Probleme mit der Kalibrierung entstehen bei dem Testfall TEST5, einer groben Across-Track-Konfiguration. Hier konvergieren alle Verfahren auf das gleiche Minimum. Der resultierende Fehler ist sehr gering.

Als Problemfälle erweisen sich die Testfälle 4, 6 und 7. Die Häufigkeit, mit der nur Methode (6) zu einem guten Ergebnis führt, weist auf ungünstig initialisierte Posen hin. Inwieweit der Ansatz zu der Korrektur der Mittelpunkte verwendet werden kann, ist nicht klar ersichtlich. Es ergibt sich keine konsistente Verringerung des Rückprojektionsfehlers. In manchen Testfällen wirkt sich dieser Schritt sogar negativ aus.

8.3.3. Betrachtung der Längenabweichung

Für die Bestimmung der Güte anhand der VDI/VDE-Richtlinie werden die Ergebnisse aus den Kalibrierungen mit zwei Datensätzen verwendet und anhand des Dritten verschiedene Längen rekonstruiert.

In Abbildung 31 sind die Ergebnisse aus Testfall TEST1, bezogen auf die Rekonstruktion von Längen aus Datensatz 1, gezeigt. Der geringe Fehler bei dem Verschneiden der Strahlen weist auf ein konsistentes System hin. Da auch die maximale Längenabweichung gering ist, können gute Ergebnisse bei der Rekonstruktion erwartet werden.

Bei allen Testfällen können hier ähnlich gute Ergebnisse erreicht werden. Diese sind in Tabelle 7 zusammengefasst.

Insgesamt verhalten sich die verschiedenen Fehlertypen (Längenabweichung, Fehler bei der Verschneidung und Rückprojektionsfehler) ähnlich. In allen Testfällen werden vergleichbare Dimensionen erreicht.

Die Längenabweichung eignet sich auch um Probleme bei der Kalibrierung nachzuweisen. Mit einer ungünstig initialisierten Startpose zwischen den Kameras wird beispielsweise

Testfall und Datensatz		Maximale Längen- abweichung [mm]	Maximaler Fehler der Verschneidung [mm]	Rückprojektionsfehler
TEST1	3	-0.022178	0.013572	0.065649
	2	0.047778	0.019738	0.056304
	1	0.031838	0.030031	0.064724
TEST2	3	-0.044991	0.013182	0.072763
	2	-0.031151	0.014495	0.069389
	1	-0.026060	0.014730	0.075254
TEST3	3	-0.078468	0.025474	0.060567
	2	0.038281	0.015490	0.114505
	1	0.019125	0.012959	0.121200
TEST4	3	-0.065259	0.017449	0.090112
	2	-0.037156	0.016532	0.102146
	1	0.070865	0.010456	0.092074
TEST5	3	0.015688	0.009480	0.070735
	2	0.041676	0.010873	0.064910
	1	-0.036548	0.009925	0.067799
TEST6	3	-0.025867	0.020856	0.133421
	2	0.019170	0.013220	0.131757
	1	0.042579	0.021167	0.097142
TEST7	3	0.021321	0.019087	0.054293
	2	0.016678	0.011308	0.068240
	1	0.015827	0.018921	0.064423

Tabelle 7: Maximale Fehler der Längenabweichung und zugehörige Rückprojektionsfehler. Es wurde keine Mittelpunktkorrektur angewendet. Die maximale Längenabweichung bezieht sich auf die Absolutwerte. Die Vorzeichen wurden bei der Ausgabe erhalten.

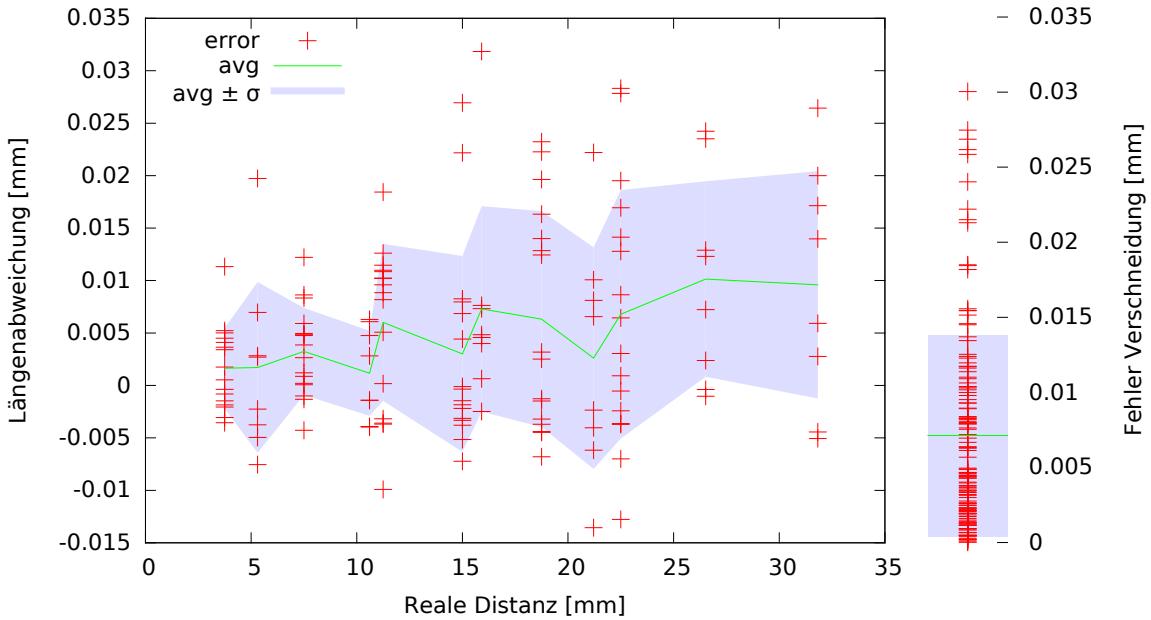


Abbildung 31: Fehler bei der Längenrekonstruktion in Testfall TEST1 kalibriert mit Datensätzen 2 und 3, erzeugt mit Datensatz 1.

in Testfall TEST7 in einer Kalibrierung mit den Datensätzen $1 \cup 2$ ein lokales Minimum gefunden. Der Rekonstruktionsfehler beträgt hier 0.61. Dieser ist größer als die Rekonstruktionsfehler aus Tabellen 6 und 7. Deutlicher zu erkennen ist das nicht optimale Minimum durch die Längenabweichungen.

Abbildung 32 zeigt die mit dem verbleibenden Datensatz 3 bestimmten Längenabweichungen. Diese hat eine maximale Abweichung von 1.00mm bei einer realen Länge von 22.5mm . Somit liegt der Fehler hier um Größenordnungen oberhalb der in Tabelle 7 aufgelisteten Fehler.

Die Kalibrierung hat hier einen Parametersatz ermittelt, welcher einen unauffälligen Rückprojektionsfehler verursacht. Auch der Fehler bei der Verschneidung der Strahlen ist gering, wie am rechten Rand der Abbildung 32 zu sehen ist. Das Modell ist allerdings mit diesem Parametersatz nicht in der Lage Längen gut zu reproduzieren. Somit entsteht ein großer Fehler in der Längenabweichung.

Die Anwendung der Mittelpunktkorrektur (siehe Abschnitt 6.6) hatte in den Testfällen keinen markanten Unterschied bewirkt.

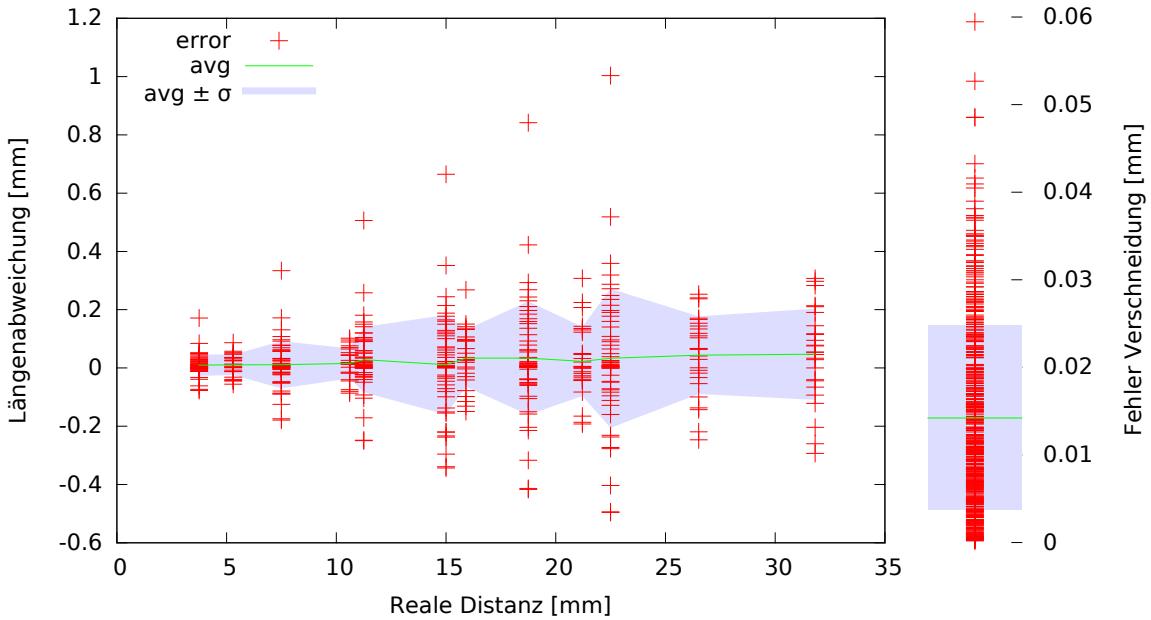


Abbildung 32: Fehler bei der Längenrekonstruktion in Testfall TEST7 kalibriert mit Datensätzen 1 und 2, erzeugt mit Datensatz 3. Durch eine ungünstige Startpose zwischen den Kameras wird bei der Kalibrierung das optimale Minimum nicht gefunden.

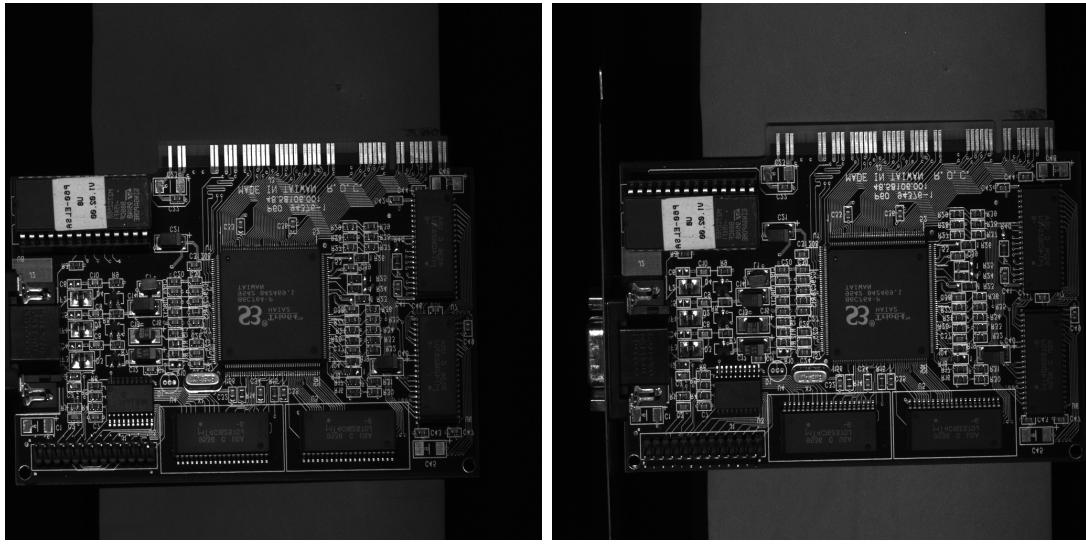
8.3.4. Rekonstruktion

Eine 3D-Rekonstruktion mit dem in Abschnitt 7.2 beschriebenen Verfahren ist mit Einschränkungen verbunden. Damit ein Punkt rekonstruiert werden kann, muss seine Umgebung eine markante Textur aufweisen. Weiterhin sollte die Geometrie keine Sprünge aufweisen. Auch wird angenommen, dass die Normale an dem Punkt annähernd in Richtung $-c^1 z$ ausgerichtet ist. Andernfalls werden die Fehler groß, welche durch nicht modellierte perspektivische Verzerrungen entstehen.

In Abbildung 33 sind die Aufnahmen einer Platine aus Testfall TEST6 gezeigt. Aufgrund der Unstetigkeiten der Oberfläche und der Verdeckungen ist eine flächendeckende Höhenrekonstruktion mit dem vorgestellten Verfahren nicht möglich. Weiterhin sind viele sich wiederholende Muster erkennbar.

In Abbildung 34 sind die rekonstruierten Höhen als Graustufen gezeigt. Diese sind einmal mit und einmal ohne die Filterung nach uneindeutigen Tiefen rekonstruiert. Die Höhenunterschiede zwischen den großen Bauteilen und der Platine selbst sind klar erkennbar. Da neben der Schrift aber keine Textur auf den Bauteilen erkennbar ist, können nicht die kompletten Oberflächen der Bauteile rekonstruiert werden.

Abbildung 35 zeigt eine Seitenansicht der Platine mit der gefilterten Rekonstruktion. Deutlich zu sehen ist die Grundfläche der Platine. Der hohe Block links gehört zu dem Sockel mit aufgestecktem Chip. Die rekonstruierte Höhe dieses Blocks beträgt 10mm. Bei den flachen Chips wurde die Höhe mit 3.6mm bis 3.8mm rekonstruiert.



(a) Kamera 1

(b) Kamera 2

Abbildung 33: Aufnahme einer Platine (Testfall TEST6).

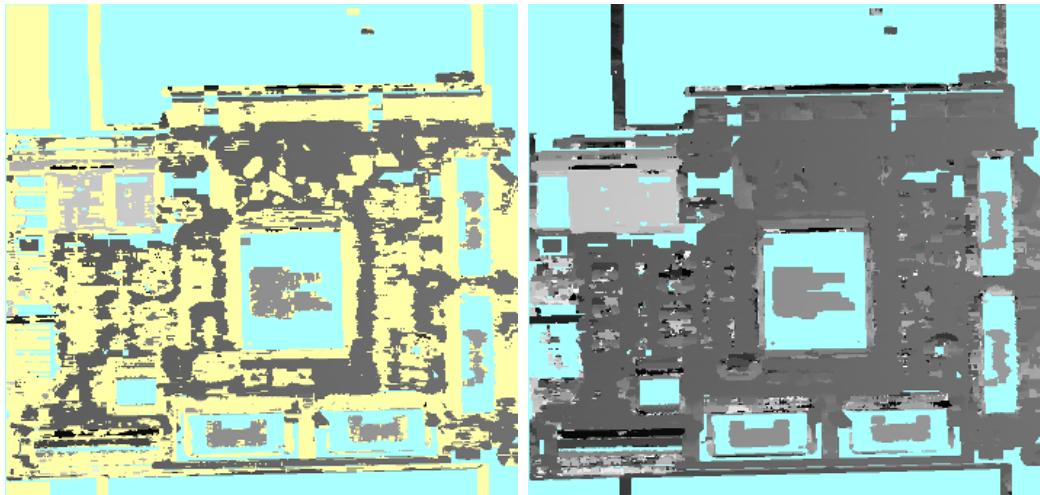
Aus den Diagonalen aus Abbildung 35 lässt sich die Blickrichtung der Kamera 1 ableiten. An diesen Diagonalen sind falsche Tiefenzuordnungen erkennbar, die nicht durch den Filter erfasst wurden. Sie entstehen, wenn zwei benachbarte Punkte den Filter passieren, aber unterschiedlichen Tiefen zugeordnet werden. Weitere falsche Zuordnungen, ohne richtig zugeordnete Nachbarn, sind unterhalb der Platine und oberhalb der Chips erkennbar.

Besser funktioniert das Verfahren bei Objekten mit einer ausgeprägten Textur. Abbildungen 36 und 37 zeigen die Aufnahmen einer flachen und einer gebogenen Korkplatte aus Testfall TEST7. Die gebogene Platte steht in der Mitte ca. 8mm von der Oberfläche des Linearschlittens ab.

Abbildungen 38a und 38b zeigen die Tiefe als Graustufen. Die flache Korkplatte lässt sich über die gesamte Fläche gut rekonstruieren. Bei der gebogenen Korkplatte treten am rechten Rand Probleme auf. Wie an dem Helligkeitsverlauf erkennbar ist, variiert die ^{cl}z -Koordinate stark. Dies führt zu einem großen Einfluss der perspektivischen Verzerrung. Folglich lassen sich die Punkte am rechten Rand nicht gut rekonstruieren.

Deutlicher werden diese Probleme auch in den rekonstruierten Oberflächen. Diese sind in Abbildung 39 gezeigt. Hierbei wurde auf die Auftragung der Grauwerte verzichtet.

Während bei der Rekonstruktion der flachen Platte nur leichte Fehler vorhanden sind, weist die gebogene Platte in der rechten Hälfte größere Fehler auf.



(a) gefiltert

(b) ungefiltert

Abbildung 34: Rekonstruierte Tiefe einer Platine als Graustufen (Testfall TEST6).

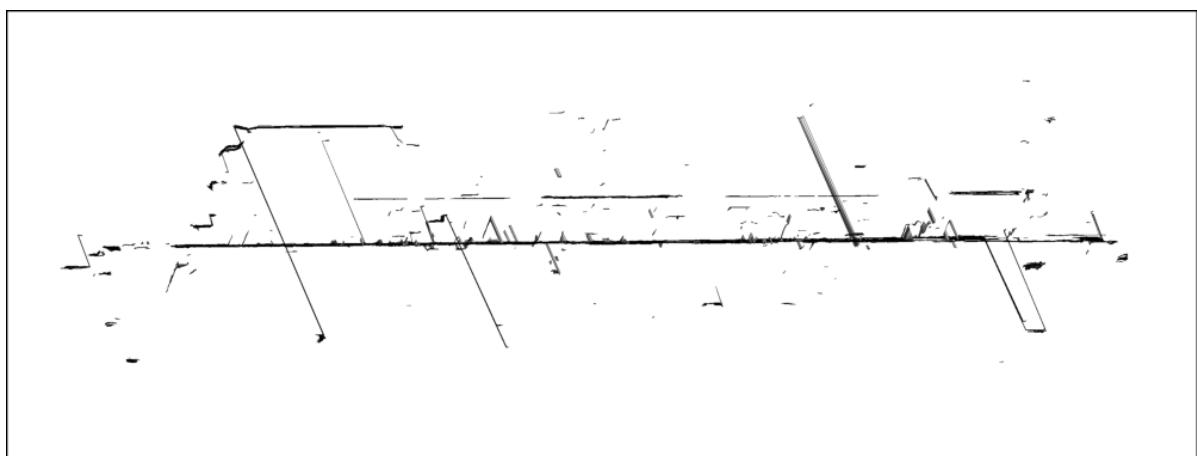
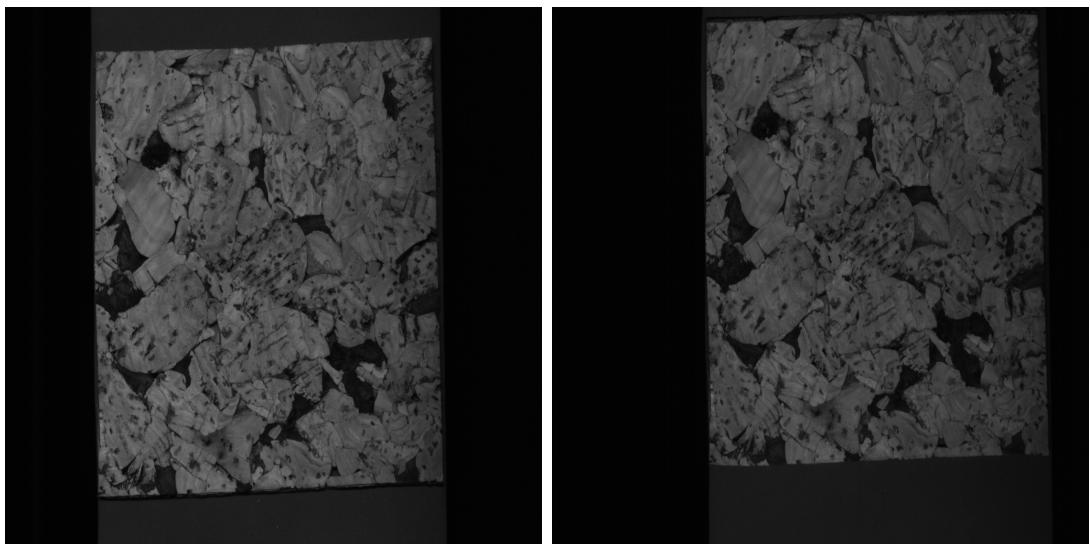


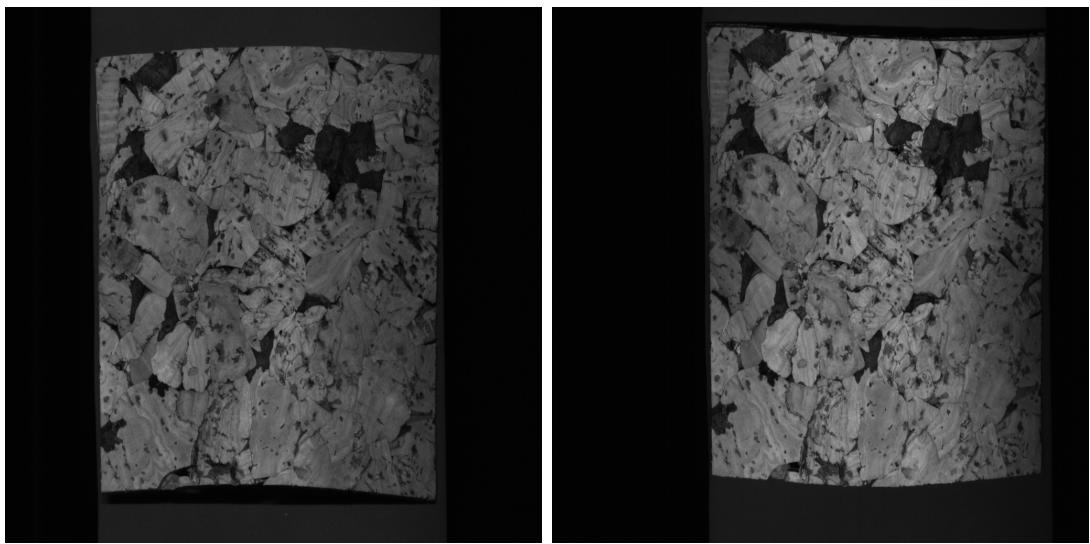
Abbildung 35: Seitliche 2D-Ansicht einer rekonstruierten Platine (Testfall TEST6).



(a) Kamera 1

(b) Kamera 2

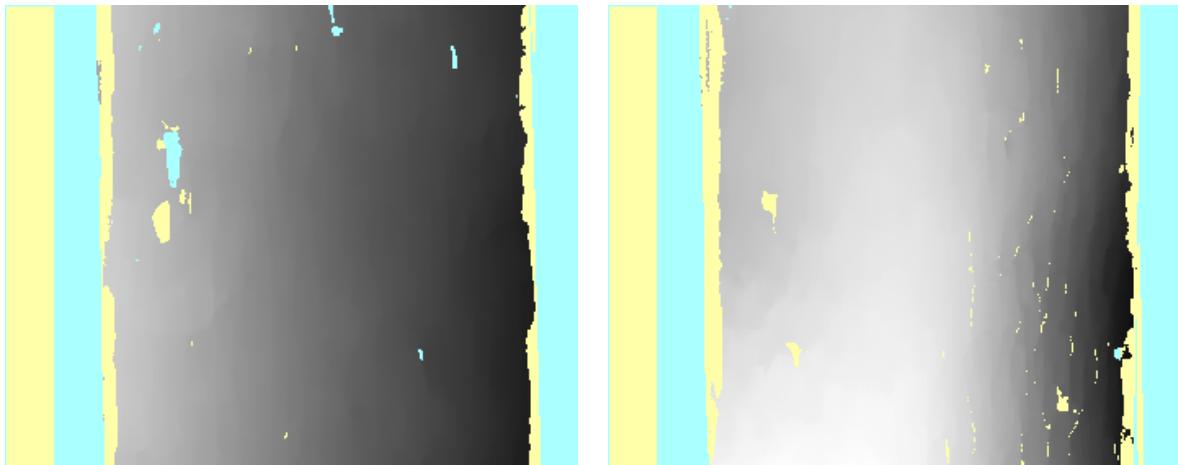
Abbildung 36: Aufnahme einer flachen Korkplatte (Testfall TEST7).



(a) Kamera 1

(b) Kamera 2

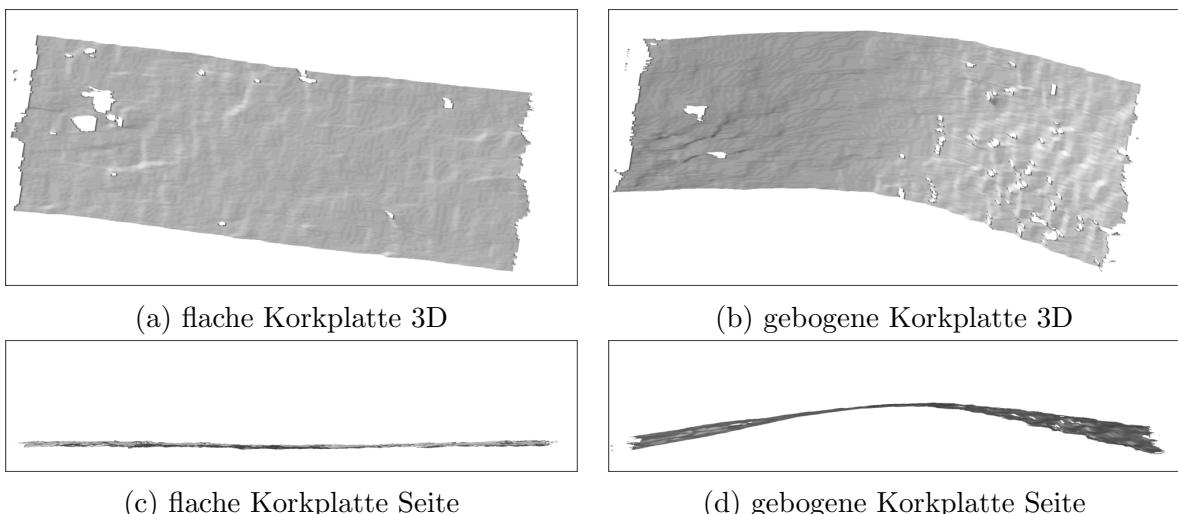
Abbildung 37: Aufnahme einer gebogenen Korkplatte (Testfall TEST7).



(a) flache Korkplatte

(b) gebogene Korkplatte

Abbildung 38: Tiefe der Korkplatten als Graustufen (Testfall TEST7).



(a) flache Korkplatte 3D

(b) gebogene Korkplatte 3D

(c) flache Korkplatte Seite

(d) gebogene Korkplatte Seite

Abbildung 39: Rekonstruktion der Oberflächen der Korkplatten aus Testfall TEST7.

8.4. Modifikation des Levenberg-Marquardt-Verfahrens

Unabhängig von dem Problem der Kalibrierung wurden in Abschnitt 6.5 Modifikationen der Levenberg-Marquardt-Iterationen vorgeschlagen. Hierbei wird die Anpassung von λ verändert. Damit soll die Konvergenz des Verfahrens beschleunigt werden. Um dies zu erreichen wird versucht, die Anzahl der erfolgreichen Iterationen zu erhöhen. Als eine erfolgreiche Iteration werden hier Iterationen angesehen, welche zu einem kleineren Fehlerwert führen.

Zwei Methoden wurden vorgeschlagen. Die erste Methode hält λ konstant, wenn eine Iteration erfolgreich ist und vorher genau ein Mal λ verringert wurde. Die zweite Methode führt einen kleinen Faktor ein, welcher das Einstellen von λ verbessern soll. Diese Methoden lassen sich auch kombinieren.

Zum Testen der Methoden werden die Kalibrierungen der realen Testfälle mit allen Kalibrierbildern verwendet. Dabei wurde das Verfahren mit der Rekonstruktion der Posen gewählt.

Mit der unveränderten Implementierung und wenigen Bildern werden die Posen rekonstruiert.

Darauf folgend werden die verschiedenen Methoden angewendet, um die zweite Optimierung in der Kalibrierung durchzuführen. Weiterhin wurde λ für alle Verfahren mit 0.001 initialisiert.

Anstatt des Faktors 10 wurde der Faktor 8 verwendet. In Methode 2 wird zusätzlich 2 als kleiner Faktor verwendet. Mit diesen Werten ist der große Faktor eine Potenz des kleinen Faktors ($8 = 2^3$).

Abbildung 40 zeigt das Verhalten des Rückprojektionsfehlers mit den Iterationen. Dabei beschränkt sich die Ansicht auf die ersten 200 Iterationen und zeigt nur die Ergebnisse erfolgreicher Iterationen.

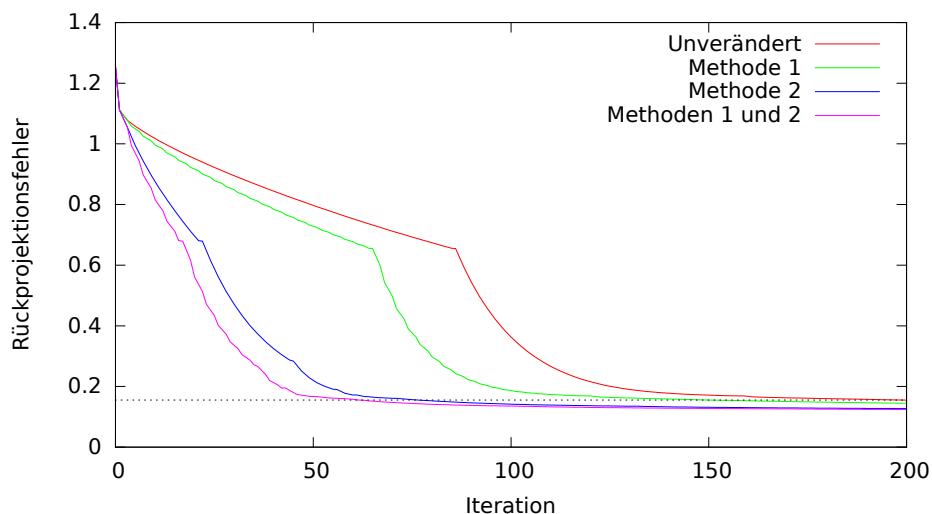


Abbildung 40: Konvergenzverhalten in Testfall TEST1.

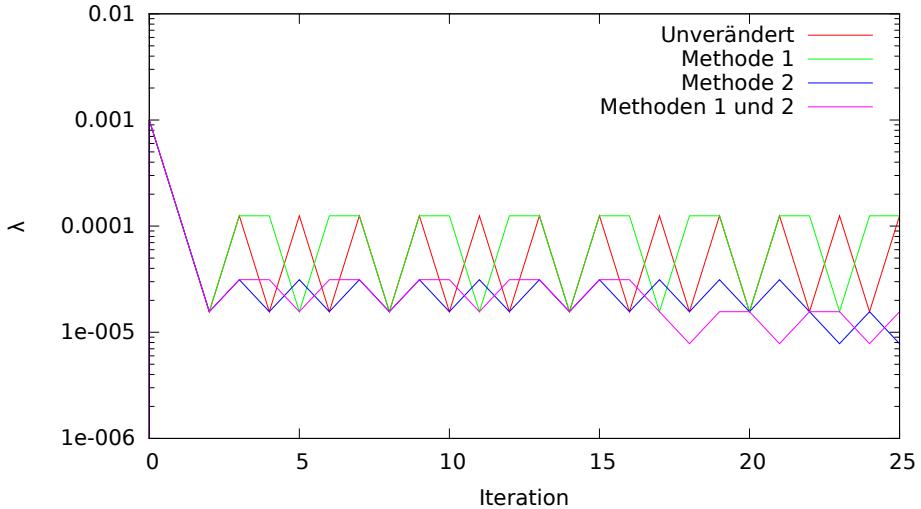


Abbildung 41: Verhalten von λ in Testfall TEST1.

In diesem Testfall konvertieren die Implementierungen mit den genannten Methoden schneller als die unveränderte Implementierung. Methode 2 hat dabei einen größeren Einfluss als Methode 1. Zusammen angewendet wird der Fehler am schnellsten minimiert. In Abbildung 41 ist für die ersten 25 Iterationen der Verlauf von λ gezeigt. Erfolgslose Iterationen sind durch ein Steigen von λ erkennbar. Bei der unveränderten Implementierung ist das Wechseln zwischen zwei Werten für λ erkennbar. Mit der Methode 1 wird auf viele Iterationen verzichtet, welche auch nicht konvergieren würden. Im Gegenzug wird aber in der Hälfte der Fälle eine mögliche Reduktion von λ erst eine Iteration später erkannt. Durch die Methode 2 wird ein Bereich von λ verwendet, der nur mit dem großen Faktor nicht erreichbar ist. Dies ermöglicht die Nutzung eines kleineren Wertes für λ . Somit wird der Fehler schneller minimiert. Die kombinierte Methode spart im Vergleich zu Methode 2 wiederum Iterationen ein.

Die Auswirkungen der verschiedenen Methoden sind verhalten sich in allen Testfällen ähnlich.

Verhältnismäßig geringe Unterschiede sind im Testfall TEST3 aufgetreten. Der Konvergenzverlauf hierzu ist in Abbildung 42 gezeigt.

Zusammengefasst werden die Ergebnisse der neuen Methoden zur Anpassung von λ in Tabelle 8.

Hier lässt sich die geringe Reduktion an Iterationen in Testfall TEST3 erkennen. Beim Testfall TEST2 ist die Kalibrierung mit der unveränderten Implementierung bei Iteration 217 konvergiert. Mit Methode 1 ist hier das Verfahren erst mit 227 Iterationen konvergiert und der finale Fehler ist minimal schlechter ($3 \cdot 10^{-7}$). Optisch wirken alle Modifikationen deutlich schneller (vgl. Abbildung 43).

Allgemein lässt sich die Konvergenz, vor allem in Verbindung beider Methoden, stark beschleunigen. Jodoch kann jede Modifizierung dazu führen, dass das Verfahren auf ein anderes Minimum konvergiert.

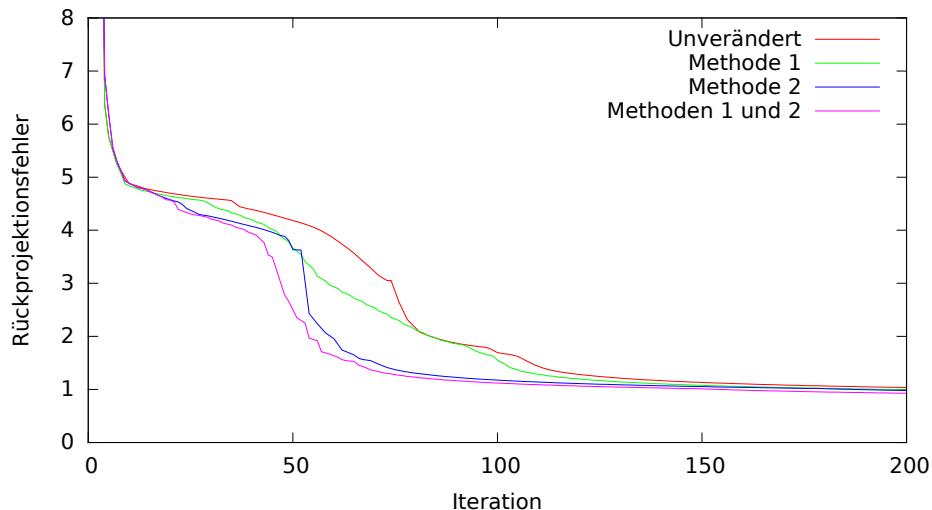


Abbildung 42: Konvergenzverhalten in Testfall TEST3.

Testfall	Benötigte Iterationen (Anteil erfolgreicher Iterationen)			
	Unverändert	Methode 1	Methode 2	Methoden 1 und 2
TEST1	199 (51.26%)	150 (68.00%)	77 (55.84%)	63 (71.43%)
TEST2	197 (51.78%)	227 (67.40%)	106 (50.94%)	73 (73.97%)
TEST3	199 (51.26%)	171 (67.25%)	163 (55.21%)	135 (69.63%)
TEST4	200 (51.50%)	153 (67.97%)	83 (56.63%)	68 (72.06%)
TEST5	200 (51.50%)	152 (67.76%)	96 (57.29%)	80 (71.25%)
TEST6	199 (51.26%)	158 (67.72%)	65 (58.46%)	49 (73.47%)
TEST7	199 (51.26%)	159 (67.92%)	98 (55.10%)	79 (69.62%)

Tabelle 8: Benötigte Iterationen zum Erreichen des Fehlers der unveränderten Vorgehensweise nach 200 Iterationen. In Klammern ist hinter den Iterationen der Anteil erfolgreicher Iterationen angegeben.

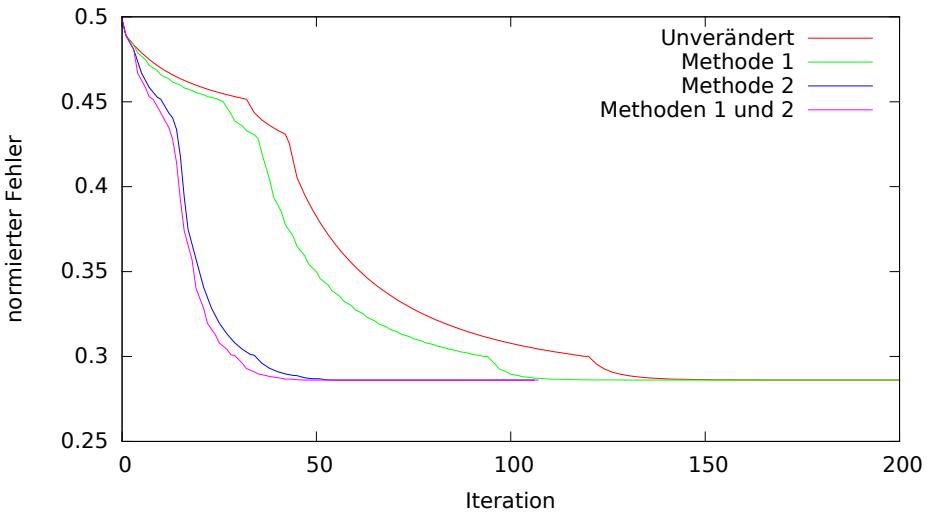


Abbildung 43: Konvergenzverhalten in Testfall TEST2.

Ein Negativbeispiel hierzu ist in Abbildung 44 zu sehen. Abweichend zu der vorgestellten Implementierung wurde λ bei Methode 2 in der ersten Iteration mit dem kleinen Faktor modifiziert. Dies führt prinzipiell dazu, dass mit der Verwendung von Methode 2 eventuell die ersten Iterationen verschwendet werden. Im Zusammenspiel von Methode 1 und Methode 2 konvergiert die Optimierung in einem anderen Minimum.

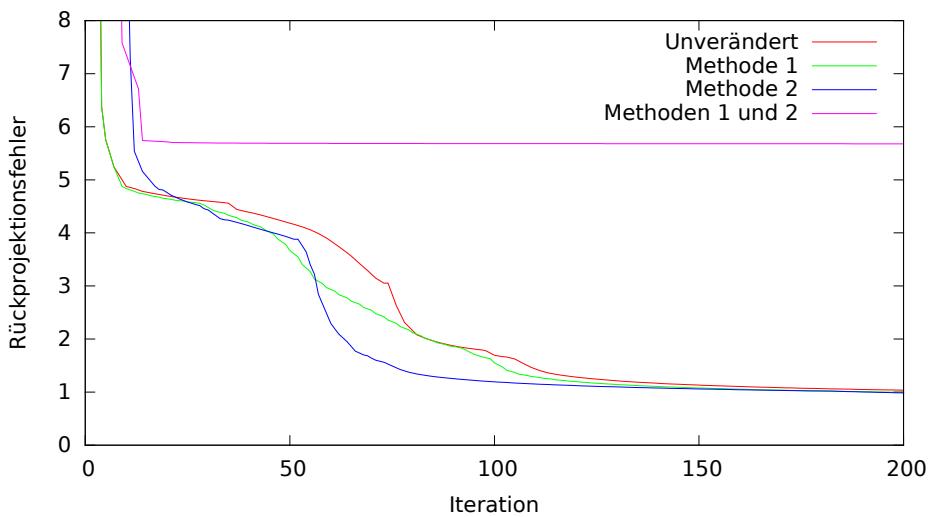


Abbildung 44: Konvergenzverhalten in Testfall TEST3, abweichende Implementierung von Methode 2.

9. Ausblick

Die vorgestellte Kalibrierung aus Kapitel 6 hat gute Ergebnisse gezeigt. In Kapitel 8 ist die Methode, erst mit wenigen Bildern zu kalibrieren, hervorgehoben. Für diese wurden anhand der ersten Kalibrierung die Posen aller Kalibrierkörper rekonstruiert.

In dieser Richtung kann die Kalibrierung noch weiter verbessert werden. Wichtig ist dabei, dass die häufische Auswahl der „wenigen Bilder“ wegfällt. Für diese Arbeit wurden fix immer die ersten 4 Bilder verwendet.

Allgemein kann es sich lohnen, hier einen anderen Ansatz zu verfolgen. Beispielsweise könnten zuerst die Kameras einzeln kalibriert werden. Aus dem Kalibrierergebnissen können die Posen zu den Bildern entnommen werden. Nun kann für jedes Bild die Pose zwischen den Kameras berechnet werden.

Mit einer geeigneten Metrik könnten davon die ähnlichsten Posen ausgewählt werden. Die entsprechenden Bilder könnten für die erste Kalibrierung verwendet werden. Somit werden auch gute Startparameter für die relative Pose der zweiten Kamera ermittelt. Wie in Abschnitt 8.3.3 gezeigt wurde, können ungünstige Startparameter dieser Pose das Auffinden des globalen Minimums verhindern.

Aus den Kalibrierungen mit realen Daten ging hervor, dass sich der Rückprojektionsfehler und der Fehler aus der Längenrekonstruktion unterschiedlich verhalten. Trotz eines geringen Rückprojektionsfehlers kann eine große Längenabweichung vorkommen. Eventuell ließe sich eine bessere Kalibrierung erreichen, wenn die Fehler aus der Längenrekonstruktion direkt mit einbezogen würden. Weiterhin bietet die Längenabweichung ein Fehlermaß, mit welchem sich bewerten lässt, ob die Kalibrierung erfolgreich war.

Die Rekonstruktion für allgemeine Geometrien der Epipolarkurven funktioniert prinzipiell. Dabei wurden aber keine geneigten Oberflächen und Verdeckungen modelliert. Weiterhin ist diese Vorgehensweise nicht so effizient umsetzbar, wie die Rekonstruktion auf Basis rektifizierter Bilder. Für eine industrielle Anwendung dieser Vorgehensweise wird deswegen abgeraten. Hier sollte sich auf die möglichst genaue Herstellung eines idealen Falls konzentriert werden. In diesen Fällen kann eine Rektifizierung durchgeführt werden und auf die Vorgehensweise der Stereo-Rekonstruktion mit Flächenkameras zurückgegriffen werden.

Somit sind die in Abschnitt 5.1 vorgestellten idealen Along-Track- und Across-Track-Konfigurationen von Bedeutung. Für die Erstellung einer Across-Track-Konfiguration ist auch die in [LAHM13] vorgestellte Methode zur Ausrichtung einer Kamera interessant. Die Bewertung, inwieweit eine ideale Konfiguration erreicht wurde, sollte anhand einer Kalibrierung getroffen werden. Zusammen mit einer Vorgabe des relevanten Tiefenbereichs können die Epipolarkurven-Ausschnitte aller Spalten berechnet werden. Relevant ist die Differenz des Maximums und Minimums der Spaltenkoordinaten (Along-Track) oder der Zeilenkoordinaten (Across-Track) der Epipolarkurven. Hier sind Werte deutlich unter einem Pixel wünschenswert.

10. Fazit

Eine Nutzung von Zeilenkameras für Stereo-Vision ist möglich. Unter den Einschränkungen eines industriellen Umfeldes kann ein solcher Aufbau modelliert werden. Die Grundannahme dabei ist die statische Pose zwischen den Kameras und das Vorherrschen einer linearen Bewegung.

Selbst mit dieser Vereinfachung ist die Kalibrierung eine anspruchsvolle Aufgabe. Der Hauptgrund dafür sind lokale Minima. Viele davon treten durch die Posen zu den Kalibrierkörpern auf.

Somit ist es sinnvoll, verschiedene Strategien bei der Kalibrierung zu verfolgen. Eine erfolgreiche Strategie ist es, zunächst nur mit wenigen Bildern zu kalibrieren und danach die Posen zu allen Kalibrierkörpern zu rekonstruieren. Diese Strategie benötigt allerdings eine erfolgreiche Kalibrierung mit wenigen Bildern. Ein Ansatz für eine automatisierte Ermittlung erfolgversprechender Bilder wurde in Kapitel 9 vorgestellt.

Anhand von simulierten und realen Daten wurde die Kalibrierung überprüft. Dabei kann ein geringer Rückprojektionsfehler in der Größenordnung eines Zehntel Pixels in den verschiedenen Testfällen erreicht werden. Zusätzlich wurde, angelehnt an die VDI/VDE-Richtlinie 2634 [VDI02], auch ein geringer Fehler bei der Rekonstruktion von Längen nachgewiesen.

Da allgemein bei einem solchen Aufbau keine Rektifizierung der Kamerabilder möglich ist, wurde ein Verfahren vorgestellt, mit welchem direkt auf Basis der Epipolarkurven eine 3D-Rekonstruktion durchgeführt werden kann. Mit diesem wurden Geometrien aus simulierten und realen Bildern rekonstruiert.

Für eine industrielle Anwendung sind aber herkömmliche Verfahren, welche von Rektifizierungen der Aufnahmen ausgehen, zu bevorzugen. Voraussetzung hierfür ist, dass eine ideale Along-Track- oder eine ideale Across-Track-Konfiguration (siehe Abschnitt 5.1) hergestellt wurde.

A. Anhang

A.1. Homogene Elementartransformationen

Verschiebung um den Vektor $(t_x, t_y, t_z)^T$:

$$Trans(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation um die x -Achse um den Winkel α :

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation um die y -Achse um den Winkel β :

$$R_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation um die z -Achse um den Winkel γ :

$$R_z(\gamma) = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A.2. Invertierung einer homogenen Matrix

Gegeben ist eine homogene Matrix H_{s1}^{s2} , gesucht ist deren Inverse H_{s2}^{s1} .

$$\begin{aligned} {}_{s2}^{s1}H &= ({}_{s1}^{s2}H)^{-1} \\ &= \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \end{aligned}$$

Diese Matrix H_{s1}^{s2} lässt sich in eine Translationsmatrix T und eine Rotationsmatrix R aufspalten.

$$\begin{aligned} {}_{s2}^{s1}H &= \left(\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right)^{-1} \\ &= (T \ R)^{-1} \end{aligned}$$

Weiterhin lässt sich eine Rotationsmatrix durch Eulerwinkel beschreiben.

$$\begin{aligned} {}_{s2}^{s1}H &= (T \ R_x(\alpha) \ R_y(\beta) \ R_z(\gamma))^{-1} \\ &= R_z(\gamma)^{-1} \ R_y(\beta)^{-1} \ R_x(\alpha)^{-1} \ T^{-1} \end{aligned}$$

Das Inverse einer Rotation um einen Winkel ϕ ist die Rotation um den Winkel $-\phi$. Da $\sin(-\phi) = -\sin(\phi)$ und $\cos(-\phi) = \cos(\phi)$ ist, entspricht das Inverse einer Rotationsmatrix ihrer Transponierten. Dies ist bei Rotationen um die Hauptachsen leicht nachvollziehbar. Es gilt aber für allgemeine Rotationsmatrizen.

Eine Translation lässt sich umkehren, indem der Vektor invertiert wird. Dies entspricht einer Multiplikation von t_x , t_y und t_z mit -1 .

$$\begin{aligned} {}_{s2}^{s1}H &= R_z(\gamma)^T \ R_y(\beta)^T \ R_x(\alpha)^T \ T^{-1} \\ &= (R_x(\alpha) \ R_y(\beta) \ R_z(\gamma))^T \ T^{-1} \\ &= R^T \ T^{-1} \\ &= \begin{pmatrix} r_{11} & r_{21} & r_{31} & -r_{11}t_x - r_{21}t_y - r_{31}t_z \\ r_{12} & r_{22} & r_{32} & -r_{12}t_x - r_{22}t_y - r_{32}t_z \\ r_{13} & r_{23} & r_{33} & -r_{13}t_x - r_{23}t_y - r_{33}t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

A.3. Methoden zum Vergleichen von Bildausschnitten

Für das Auffinden von Korrespondenzen werden Bildausschnitte zweier Bilder mit einander verglichen. Diese Ausschnitte werden hier als A und B bezeichnet. Sie haben die gleichen Dimensionen s_i und s_j . A_{ij} bezeichnet den Grauwert an der Stelle (i, j) und ist als ein Byte ($0 - 255$) gegeben. Weiterhin ist $\bar{A} = (\sum_{i=1}^{s_i} \sum_{j=1}^{s_j} A_{ij}) / (s_i s_j)$. Gleiche Definitionen gelten für B .

Summe der absoluten Differenzen

Rechnung:

$$d_{\text{sad}} = \frac{1}{s_i s_j} \sum_{i=1}^{s_i} \sum_{j=1}^{s_j} (abs(A_{ij} - B_{ij}))$$

Wertebereich:

$$0 \leq d_{\text{sad}} \leq 255$$

Optimum ($A = B$):

$$0$$

Summe der quadratischen Differenzen

Rechnung:

$$d_{\text{ssd}} = \frac{1}{s_i s_j} \sum_{i=1}^{s_i} \sum_{j=1}^{s_j} (A_{ij} - B_{ij})^2$$

Wertebereich:

$$0 \leq d_{\text{ssd}} \leq 255^2$$

Optimum ($A = B$):

$$0$$

Normalisierte Kreuzkorrelation

Rechnung:

$$d_{\text{ncc}} = \sum_{i=1}^{s_i} \sum_{j=1}^{s_j} \frac{(A_{ij} - \bar{A})(B_{ij} - \bar{B})}{\sqrt{\left[\sum_{i=1}^{s_i} \sum_{j=1}^{s_j} (A_{ij} - \bar{A})^2 \right] \left[\sum_{i=1}^{s_i} \sum_{j=1}^{s_j} (B_{ij} - \bar{B})^2 \right]}}$$

Wertebereich:

$$-1 \leq d_{\text{ncc}} \leq 1$$

Optimum helligkeitsinvariant ($A = x B + y$, $x > 0$):

$$1$$

Literatur

- [Bas13] Basler AG. Basler racer line scan cameras. http://www.baslerweb.com/media/documents/BAS1309_racer_Brochure_Web_de.pdf, October 2013.
- [DRS11] Jamil Draréni, Sébastien Roy, and Peter F. Sturm. Plane-Based Calibration for Linear Cameras. *International Journal of Computer Vision*, 91(2):146–156, 2011.
- [FP08] Yasutaka Furukawa and Jean Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. In *CVPR*, 2008.
- [FR05] Peter Fricker and Arthur Rohrbach. Pushbroom scanners provide highest resolution earth imaging information in multispectral bands. *IPSRS Hannover Workshop: High Resolution Earth Imaging for Geospatial Information: Proceedings*, 2005.
- [GH95] Rajiv Gupta and Richard I. Hartley. Camera Estimation for Orbiting Pushbroom Imaging Systems. Proceedings Second Asian Conference Computer Vision, December 1995.
- [GH97] Rajiv Gupta and Richard I. Hartley. Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):963–975, September 1997.
- [Hei00] Janne Heikkilä. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, October 2000.
- [HMJK05a] Ayman F. Habib, Michel Morgan, Soo Jeong, and Kyung-Ok Kim. Analysis of epipolar geometry in linear array scanner scenes. *The Photogrammetric Record*, 20(109):27–47, 2005.
- [HMJK05b] Ayman F. Habib, Michel F. Morgan, Soo Jeong, and Kyung-Ok Kim. Epipolar Geometry of Line Cameras Moving with Constant Velocity and Attitude. *ETRI Journal*, 27(2):172–180, 2005.
- [HSH05] Heiko Hirschmüller, Frank Scholten, and Gerd Hirzinger. Stereo Vision Based Reconstruction of Huge Urban Areas from an Airborne Pushbroom Camera (HRSC). In *Pattern Recognition*, pages 58–66. Springer, 2005.
- [HZ04] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, April 2004.
- [Kim00] Taejung Kim. A study on the epipolarity of linear pushbroom images. *Photogrammetric Engineering and Remote Sensing*, 66(8):961–966, 2000.

- [LAHM13] Erik Lilienblum, Ayoub Al-Hamadi, and Bernd Michaelis. A coded 3d calibration method for line-scan cameras. In Joachim Weickert, Matthias Hein, and Bernt Schiele, editors, *GCPR*, volume 8142 of *Lecture Notes in Computer Science*, pages 81–90. Springer, 2013.
- [Mar63] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.
- [MKJH06] Michel Morgan, K Kim, Soo Jeong, and Ayman Habib. Epipolar resampling of space-borne linear array scanner scenes using parallel projection. *Photogrammetric engineering and remote sensing*, 72(11):1255–1263, 2006.
- [MVT13a] MVtec. HALCON Solution Guide III-C 3D Vision, 2013.
- [MVT13b] MVtec. HALCON/HDevelop Referenzhandbuch, 2013.
- [NW06] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, 2nd edition, 2006.
- [Oh11] Jaehong Oh. Novel Approach to Epipolar Resampling of HRSI and Satellite Stereo Imagery-based Georeferencing of Aerial Images. Master’s thesis, The Ohio State Univiversity, 2011.
- [PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 2nd edition, 1992.
- [SRT⁺11] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and João Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundation and Trends in Computer Graphics and Vision*, 6:1–183, January 2011.
- [SUW08] Carsten Steger, Markus Ulrich, and Christian Wiedemann. *Machine Vision Algorithms and Applications*. Wiley-VCH. Wiley, 2008.
- [VDI02] VDI/VDE 2634 Blatt 1:2002-05. Optische 3D-Messsysteme — Bildgebende Systeme mit punktförmiger Antastung, May 2002.
- [Whi90] Preston White. *SPOT User’s Handbook*. SPOT Image Corporation, 1990.
- [WHL11] Mi Wang, Fen Hu, and Jonathan Li. Epipolar resampling of linear pushbroom satellite imagery by a new epipolarity model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3):347–355, May 2011.