

Beiträge zur Analyse,
Modellierung und Kalibrierung
von Kameras und 3D-Tiefensensoren

Dissertation

zur Erlangung des Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

David Norbert Fiedler

Dortmund
2014

Tag der mündlichen Prüfung: 18.03.2014

Dekan: Prof. Dr.-Ing. Gernot Fink

Gutachter: Prof. Dr. Heinrich Müller
Prof. Dr. Andreas Kolb

Widmung

Für Carolin.

Danksagung

Ich möchte mich bei allen bedanken, die mich bei der Entstehung dieser Dissertation begleitet haben. An erster Stelle danke ich meinem Betreuer, Herrn Prof. Dr. Heinrich Müller, der mich mit seinen Ratschlägen und produktiven Diskussionen in zahlreichen Abendstunden unterstützte und der mir insbesondere in der letzten Phase den nötigen zeitlichen Freiraum gewährte. Herrn Prof. Dr. Kolb danke ich für die Erstellung des Gutachtens. Ich danke meinen Kollegen, die während meiner Zeit am Lehrstuhl immer für ein angenehmes Arbeitsklima gesorgt haben. Besonders danke ich Marcel Gaspar und Thomas Wiederkehr, auf deren offenes Ohr und Hilfsbereitschaft ich immer zählen konnte.

Ein besonderer Dank gilt meinen Eltern, Schwiegereltern und vor allem meiner Frau, die mich immer aufs neue motivierten und deren vielfältige Unterstützung das Gelingen dieser Dissertation ermöglichte.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Kameras und 3D-Tiefensensoren	1
1.1.1	Kameras	2
1.1.2	3D-Tiefensensoren	2
1.2	Abbildungsverhalten von Kameras und 3D-Tiefensensoren	3
1.2.1	Kameragestützte Materialflusssysteme	4
1.2.2	LKW Ladungsüberwachung	5
1.2.3	Oberflächenerfassung und Kartographierung mit Flugobjekten	7
1.3	Zielsetzung und Beiträge	9
1.3.1	Kameras	10
1.3.2	3D-Tiefensensoren	11
1.3.3	Eigenanteil der Beiträge	13
1.4	Gliederung der Arbeit	13
2	Modellierung und Kalibrierung von Kameras	15
2.1	Technischer Aufbau	15
2.2	Lochkameramodell	17
2.2.1	Intrinsische und extrinsische Kameraparameter	19
2.2.2	Pixelbasierte Bildebene	20
2.3	Linsenverzerrung	21
2.3.1	Radiale Verzerrung	21
2.3.2	Tangentiale Verzerrung	23
2.3.3	Mathematische Verzerrungsmodelle	24
2.4	Kamerakalibrierung	26
2.4.1	Intrinsische Kalibrierung	27
2.4.2	Extrinsische Kalibrierung	31
2.4.3	Globale Kalibrierung	32
2.5	Forward- und Backward-Modelle	32
2.5.1	Forward- und Backward-Verzerrungsmodelle	32
2.5.2	Invertierung der Verzerrungsrichtung	33
2.5.3	Entkoppeltes Backward-Verzerrungsmodell	34
3	Partikelfilter basierte Linsenverzerrungskorrektur	37
3.1	Motivation und Problemspezifikation	37
3.1.1	Motivation	38
3.1.2	Problemspezifikation	41

3.1.2.1	Zielfunktion	41
3.1.2.2	Weitere Fehlerfunktionen	43
3.2	Lösung	44
3.2.1	Grundidee	44
3.2.2	Funktionsweise und Anpassung des iterativen Partikelfilters	46
3.2.2.1	Resampling	48
3.2.2.2	Drift	50
3.2.2.3	Diffusion	51
3.2.2.4	Measurement	52
3.2.2.5	Normalization	54
3.2.2.6	Fusion der Hypothesen	55
3.2.3	Iterative Optimierung der Linsenentzerrungsparameter	55
3.2.3.1	Initialisierung der Partikel	56
3.2.3.2	Optimierungsprozess und Abbruchkriterien	57
3.2.4	Konfiguration relevanter Parameter des Partikelfilters	58
3.2.4.1	Anzahl der Partikel	59
3.2.4.2	Diffusion-Kontrollparameter	61
3.2.4.3	Standardabweichung bei der Initialisierung der Parametervektoren	63
3.2.4.4	Distanzfunktion beim Drift-Schritt	63
3.2.5	Effiziente Berechnung entzerrter Bildpunkte	64
3.2.6	Effiziente Korrektur verzerrter Bilder	64
3.3	Evaluierung der Entzerrungsleistung	67
3.3.1	Beschreibung des Evaluierungssystems	68
3.3.2	Auswirkung des Zufallsprozesses	68
3.3.3	Variierende Brennweite	70
3.3.4	Verrauschte Daten	73
3.3.5	Unschärfe Bilddaten durch fehlerhafte Fokussierung	74
3.3.6	Fischaugen-Objektiv	76
3.3.7	Vergleich mit der OCamCalib-Toolbox	77
3.4	Direkter Vergleich der Optimierungsverfahren	78
3.4.1	Beschreibung des Evaluierungssystems	78
3.4.2	Evaluierung	80
3.5	Grenzen des Partikelfilter-Ansatzes	81
4	3D-Tiefensensoren	83
4.1	3D-Laserscanner	83
4.2	Time-of-Flight Kameras	84
4.3	Stereo-Kamerasysteme	85
4.3.1	Triangulierungsbasierte Distanzberechnung	85
4.3.2	Stereo-Kalibrierung	88
4.3.3	Disparitätsbasierte Distanzberechnung	90
4.4	Ansätze mit strukturiertem Licht	93
4.4.1	Lichtmusterübersicht	94
4.4.2	Vorteile und kommerzielle Produkte	96
4.5	Kinect als Low-Cost-Tiefensensor	97
4.5.1	Strukturiertes Punktmuster	99

4.5.2	Berechnung der Disparitätswerte	100
4.5.3	Mathematisches Modell zur Tiefenbestimmung	102
4.5.4	Umrechnung der Tiefendaten in 3D-Punktpositionen	103
4.5.5	Kalibrierung der Tiefenberechnungsparameter	104
4.5.6	Tiefenauflösung	105
4.5.7	Datenzugriff und SDKs	106
4.6	Potential und Probleme kostengünstiger Tiefensensoren	107
5	Analyse und Korrektur von Messfehlern des Tiefensensors Kinect	109
5.1	Problemstellungen und Lösungsansätze	110
5.1.1	Problemstellungen	110
5.1.2	Lösungsansätze	111
5.2	Komplexer sensorortsabhängiger Messfehler	112
5.2.1	Senkrechte Artefakte im Tiefenbild	112
5.2.2	Fehler beim Messen planarer Oberflächen	113
5.2.3	Planaritätsbasierter Korrekturansatz	114
5.2.3.1	Berechnung der Delta-Plane	115
5.2.3.2	Reproduzierbarkeit	118
5.2.3.3	Delta-Plane basierter Korrekturansatz	120
5.2.3.4	Quantisierung der Tiefenmesswerte	121
5.2.3.5	Distanzabhängigkeit der Delta-Plane	122
5.2.3.6	Multi-Delta-Plane basierte Korrektur unter Berücksichtigung der Messdistanz	127
5.2.4	Evaluierung	130
5.2.4.1	Delta-Plane basierte Korrekturleistung	130
5.2.4.2	Delta-Plane versus Multi-Delta-Plane	131
5.2.4.3	Untersuchung typischer Innenraumszenarien	135
5.3	Online-Gewinnung von Referenzmessdaten	139
5.3.1	Kalibrierobjekte	140
5.3.1.1	Modell des Kalibrierobjekts	140
5.3.1.2	Realisierung der Kalibrierobjekte	141
5.3.1.3	Sichtbarkeit des Schachbrettmusters	143
5.3.2	Stereo-Kalibrierung der Kinect	144
5.3.3	Lagebestimmung des Kalibrierobjekts mittels der RGB-Kamera	144
5.3.4	Schätzung der Kalibrierobjektlage bezüglich der IR-Kamera	147
5.3.5	Berechnung der Referenztiefenmessung	147
5.3.5.1	Pixelbasierte Referenztiefenmessung	148
5.3.5.2	Flächenbasierte Referenztiefenmessung	151
5.3.6	Genauigkeitsanalyse	151
5.3.6.1	Laserdistanzbasierte Bestimmung der Referenzmesspunkt- lage	152
5.3.6.2	Evaluierung der Referenzmesspunkt- lage	155
5.3.6.3	Evaluierung der Messgenauigkeit	155
5.3.6.4	Evaluierung des Messrauschens	157
5.3.6.5	Messrauschen bei flächenbasierter Referenzmessung	158
5.4	Auswirkungen von Umwelteinflüssen auf den Kinect-Tiefensensor	159
5.4.1	Thermischer Einfluss auf die internen Kameras der Kinect	160

5.4.1.1	Bildbasierter Vergleich	161
5.4.1.2	Optical-Flow basierter Vergleich	162
5.4.1.3	Kameraparameter basierter Vergleich	163
5.4.2	Thermischer Einfluss auf die Projektion des strukturierten Lichts	165
5.4.3	Auswirkung der Umwelt- und thermalen Einflüsse auf Dis- tanzmessungen	167
5.4.3.1	Versuchsaufbau und Distanzmessungen	167
5.4.3.2	Distanzveränderungen während der Aufwärmphase	168
5.4.3.3	Distanzunterschiede zwischen Thermalzuständen .	169
5.4.3.4	Rauschuntersuchung in beiden Thermalzuständen .	171
5.4.3.5	Einfluss des kinectinternen Lüfters	172
5.4.3.6	Auswirkung natürlicher Luftbewegungen	173
5.4.3.7	Einfluss natürlicher Temperaturschwankungen . . .	174
5.4.3.8	Korrelation zwischen den Distanzmessungen und der Temperatur	175
5.4.3.9	Einfluss des Stand-By-Betriebs und Verbindungsun- terbrechungen	176
5.4.3.10	Temperaturabhängigkeit der extern montierten RGB- Kamera	178
5.5	Untersuchung der Temperatúrauswirkung auf den MDP basierten An- satz	179
5.5.1	Auswirkung auf das gesamte Tiefenbild	179
5.5.2	Auswirkung auf die MDP	180
6	Zusammenfassung	183
	Literaturverzeichnis	185

1

Einleitung

Das Ziel, die Leistungsfähigkeit des Menschen bei der Erfassung, der Analyse und der Interpretation seiner dreidimensionalen Umwelt durch seine visuelle Sinneswahrnehmung mit der Hilfe von Computern nachzubilden, bleibt unerreicht. Dennoch wurden in den vergangenen drei Jahrzehnten beachtliche Fortschritte in dem Bereich des Visual Computing und seinen Teilbereichen Computer Vision, Computergraphik und Maschinelles Lernen erzielt. Insbesondere sind Methoden, Vorgehensweisen und mathematische Modelle aus dem Teilgebiet Computer Vision für eine Vielzahl heutiger Anwendungsbereiche unverzichtbar, wie z.B. in der Fertigungsindustrie, in der medizinischen Bildverarbeitung, beim Motion Capturing, bei Fahrassistenzsystemen oder der Sicherheitsüberwachung.

Alle diese und weitere Anwendungen haben gemeinsam, dass der automatischen Verarbeitung und Analyse eine Erfassung der visuellen Daten durch entsprechende Sensoren vorausgehen muss. Diese Dissertation befasst sich mit der Analyse, der Modellierung und der Kalibrierung von Kameras und 3D-Tiefensensoren, die eine wesentliche Rolle bei der Erfassung der realen Umgebung in zahlreichen Anwendungen spielen. Der folgende Abschnitt stellt die Gegebenheiten bei Kameras und 3D-Tiefensensoren einführend dar. Abschnitt 1.2 wendet sich dem Abbildungsverhalten dieser Geräte zu und beleuchtet verschiedene Aspekte hiervon an drei Projekten, an denen der Autor mitgewirkt hat. Ausgehend davon stellt Abschnitt 1.3 die Zielsetzung der Dissertation vor und gibt einen Überblick über ihre Resultate. Abschnitt 1.4 beschreibt die Gliederung der Arbeit.

1.1 Kameras und 3D-Tiefensensoren

Kameras und 3D-Tiefensensoren sind heute komplexe Systeme, die auf optoelektronischen Bauteilen basieren. Einer der wichtigsten und am häufigsten eingesetzten Sensoren im Bereich Computer Vision ist die Kamera, die eine beobachtete Szene auf einen zweidimensionalen Sensorchip abbildet und ein Rasterbild zur Verfügung stellt [Luh10]. Dieses Rasterbild ist die Ausgangsbasis für zahlreiche digitale Bildverarbeitungsalgorithmen, auf denen viele Applikationen aufbauen. Aufgrund des technischen Fortschritts und immer weiter sinkender Herstellungskosten für Elektronikbauteile rücken auch 3D-Tiefensensoren [FGG⁺13] immer mehr in den Vordergrund. Diese liefern direkt dreidimensionale Informationen der beobachteten Szene, die auch als Tiefendaten bezeichnet werden. Gegenüber Kameras ermöglichen diese 3D-Tiefensensoren eine weiterführende Analyse der Szene.

1.1.1 Kameras

Den Kern einer Kamera bildet der Sensor (Sensorchip), der aus einem zweidimensionalen Feld von fotoempfindlichen Sensorelementen besteht, die in den meisten Fällen auf der CCD- oder der CMOS-Technologie [ALL⁺07] basieren. Diese Sensoren sind in der Lage, zweidimensionale Informationen über die Helligkeitsverteilung auf ihrer Fläche zu liefern. Um die Lichtstrahlung, die von der Szene ausgeht, optimal auf den Sensor zu bündeln, wird eine entsprechende Optik eingesetzt (Objektiv). Somit ist es möglich, das Abbild (Projektion) einer Szene in Form des bereits erwähnten Rasterbildes zu gewinnen. Die einzelnen Sensorelemente werden als Pixel bezeichnet, abkürzend für die englische Bezeichnung "picture element". Die Sensoren unterscheiden sich hinsichtlich der Größe und der Anzahl der Pixel, wodurch die zur Verfügung stehende Bildauflösung, aber auch weitere Qualitätsmerkmale, wie z.B. das Rauschen, beeinflusst werden.

Es gibt zahlreiche, auf elektronischen Sensoren basierende Kameratypen, die sich je nach Einsatzgebiet unterscheiden können, und von denen hier nur wenige Beispiele genannt werden. Das bekannteste Beispiel stellen Fotokameras dar, in denen Sensoren den ursprünglichen analogen Film ersetzen und typischerweise sehr hohe Auflösungen aufweisen (z.B. 21 Megapixel). Auch Video- und Industriekameras basieren auf ähnlichen Sensoren, wobei diese typischerweise für höhere Bildausleseraten bei mittleren Auflösungen konzipiert sind.

Generell lassen sich sensorbasierte Kameras z.B. bezüglich des wahrgenommenen Frequenzspektrums und dem damit verbundenen konkreten Einsatzgebiet unterscheiden [Luh10]. Die genannten Fotokameras aus dem Unterhaltungssektor, wie z.B. Kompakt-Fotokameras, Spiegelreflex-Kameras, Handy-Kameras oder Webcams, arbeiten im für das menschliche Auge sichtbaren Bereich des Lichts (Wellenlänge von 380 bis 720 nm). Infrarot-Kameras nutzen spezielle Frequenzfilter und arbeiten im für Menschen unsichtbaren Bereich des Lichtspektrums (720 bis 3000 nm). Sie werden häufig bei Überwachungskameras und Nachtsichtgeräten eingesetzt.

Neben einem Sensor haben alle Kameratypen gemeinsam, dass sie ein entsprechendes Objektiv verwenden, das eine wesentliche Rolle bzgl. der Qualität der Rasterbilddaten spielt.

1.1.2 3D-Tiefensensoren

Ein Sensor oder ein System aus mehreren Einzelsensoren, die in der Lage sind, dreidimensionale Informationen einer zu erfassenden Szene zu liefern, werden im Folgenden als 3D-Tiefensensoren bezeichnet. Dabei liegen die von den Sensoren ermittelten, dreidimensionalen Informationen typischerweise entweder als Distanz- oder Tiefenwerte bzw. direkt als eine Menge von dreidimensionalen Punktkoordinaten, einer sog. Punktwolke, vor.

Zu den bekanntesten 3D-Tiefensensoren zählen Laserscanner [Bla04], die Distanzwerte einer gegebenen Szene mit Hilfe eines in unterschiedliche Richtungen ausgesendeten Laserstrahls ermitteln. Dieser Sensor wird z.B. bei der Gebäude- und Landschaftsvermessung verwendet, kommt aber auch bei kürzeren Distanzen, z.B. zur Kollisions-

erkennung bei autonomen Fahrzeugen, zum Einsatz. Typischerweise liefern Laser-scanner Tiefendaten in Form von Punktwolken. Eine Technologie für Tiefensensoren ist Time-of-Flight (ToF) [LSBR99].

Eine spezielle Ausprägung von 3D-Tiefensensoren sind kameraähnliche Geräte, die Tiefendaten in Bildform, also als zweidimensionales Rasterbild ähnlich dem von Kameras liefern. Dabei ist die von jedem Pixel zur Verfügung gestellte Information nicht die Lichtintensität, sondern die Distanz zum Szenenobjekt, welches an der Pixelposition zu beobachten ist. Aufgrund der hohen Bildwiederholrate werden diese Sensoren für dynamische Anwendungsszenarien wie z.B. der Benutzerinteraktion eingesetzt. Ein Beispiel für ein derartiges bildgebendes Gerät ist die sog. PMD-Kamera [LSKK10], die nach dem ToF-Prinzip arbeitet.

Stereo-Kamerasysteme [HZ03], bestehend aus zwei nebeneinander angeordneten Kameras, können ebenfalls als 3D-Tiefensensoren dienen. Dabei beobachten beide Kameras einen gemeinsamen Ausschnitt einer Szene. Die Abschätzung der Tiefe der beobachteten Objekte ergibt sich aufgrund der in den Bildern zu beobachtenden Parallaxe. Aufgrund der hohen Auflösungen von Kameras sind hohe Tiefenaufösungen möglich, so dass Stereo-Kamerasysteme z.B. für präzise Vermessungsaufgaben verwendet werden können.

Ein weiterer Typ von 3D-Tiefensensoren basiert auf strukturiertem Licht [Gen11]. Das Funktionsprinzip ähnelt dem der Stereo-Kameras, jedoch wird eine Kamera gegen einen Projektor getauscht, der ein bekanntes Lichtmuster in die Szene projiziert. Aus den Verzerrungen des ursprünglichen Musters kann auf die 3D-Struktur der Szene geschlossen werden. Während der letzten drei Jahre hat sich der ursprünglich für den Unterhaltungssektor entwickelte Microsoft-Kinect¹ Sensor als Tiefensensor etabliert. Die geringen Kosten, die kleine Bauweise, die hohe Bildwiederholrate sowie ein umfassendes Framework zeichnen diesen Sensor aus, so dass er in kürzester Zeit Einzug in eine Vielzahl von Anwendungsbereichen des Visual Computing gefunden hat.

1.2 Abbildungsverhalten von Kameras und 3D-Tiefensensoren

Kameras und 3D-Tiefensensoren dienen der Erfassung von Informationen über die reale Umgebung. In vielen Anwendungen des Visual Computing ist es notwendig, das Abbildungsverhalten, also die Art und Weise, wie die reale Umgebung erfasst wird, zu kennen. Das Abbildungsverhalten wird üblicherweise mit Hilfe mathematischer Modelle beschrieben, die von Parametern abhängen. Liegen die Parameterwerte des Modells eines konkreten Sensors vor, so kann es zum Rückschluss von den durch den Sensor erfassten Informationen auf die reale Umgebung verwendet werden. Die Ermittlung der Werte der Modellparameter und somit des konkreten Abbildungsverhaltens eines gegebenen Sensors ist Gegenstand der Kalibrierung. Die Notwendigkeit einer individuellen Kalibrierung ergibt sich etwa durch Toleranzen bei der Fertigung eines Sensors, die Einfluss auf das Abbildungsverhalten haben. Aber auch die Gege-

¹<http://www.xbox.com/en-US/kinect>

benheiten des Einsatzortes des Sensors, wie z.B. die Umgebungstemperatur, können Auswirkungen haben.

Im Folgenden werden Projekte vorgestellt, an denen der Autor der Dissertation beteiligt war, und an denen die Relevanz des korrekten Abbildungsverhaltens exemplarisch verdeutlicht wird.

1.2.1 Kameragestützte Materialflusssysteme

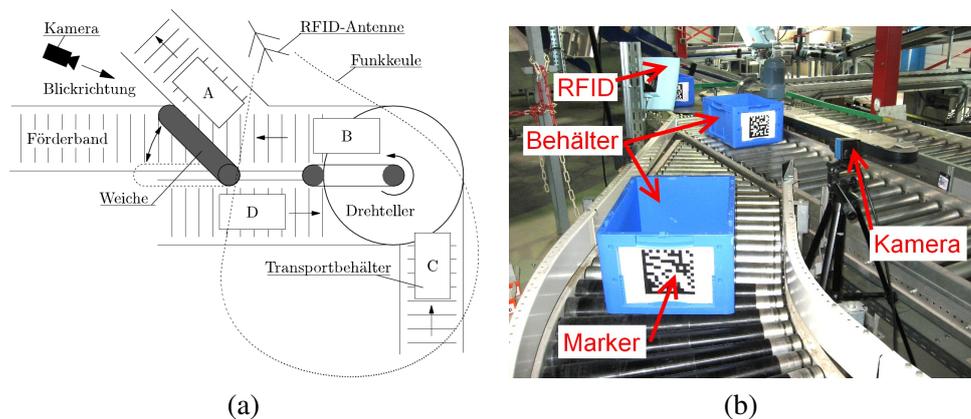


Abbildung 1.1: In (a) ist der Ausschnitt einer schematischen Darstellung einer Förderanlage gezeigt. Die realistische Abbildung in (b) zeigt die mit Markern versehenen Transportbehälter (Bins) sowie die Kamera an einer Gabelungsstelle der Förderanlage. (Bildquelle: [WFH⁺10])

Die nachfolgende Darstellung stützt sich auf die Publikationen [WFH⁺10] und [TWF⁺11], die der Autor der Dissertation mitverfasst hat.

Dabei geht es um einen im Logistikkontext neuartigen Ansatz für die kontinuierliche Detektion, Lokalisierung und Identifikation von Paketen bzw. Transportbehältern (sog. Bins) auf Rollen- und Bandfördersystemen, wie sie typischerweise in automatisierten Logistikzentren wie Postzentralen und Versandhäusern zum Einsatz kommen. Üblicherweise werden Barcodescanner und Lichtschrankensysteme im Eingangsbereich des Fördersystems platziert und nur dort die sequenzielle Identifizierung und Positionsbestimmung der Bins durchgeführt. Im weiteren Verlauf wird darauf vertraut, dass die Reihenfolge und der Abstand der Bins während des Transports unverändert bleiben, was z.B. bei Gabelungsstellen an Weichen relevant ist.

Bei der in den Publikationen vorgeschlagenen Lösung für dieses Problem wurde unter anderem eine kostengünstige Kamera verwendet, um die Bins an kritischen Stellen der Förderanlage zu beobachten (siehe Abb. 1.1). Mit Hilfe der Kamera war eine exakte Detektion, Lokalisierung (Reihenfolge, Position usw.) und Identifizierung durchführbar, so dass eine präzise Weichensteuerung ermöglicht wurde. Somit konnten nicht nur die Ausfallsicherheit erhöht, sondern auch die Abstände zwischen den Bins reduziert, der Durchsatz der Anlage gesteigert und Kosten minimiert werden.

Da die Bins grundsätzlich in Bewegung waren und die Kamera fortlaufend Bilder aufzeichnete, konnten neu in das Kamerabild eintretende Bins und deren 2D-Marker mit Hilfe von Differenzbildern [GW01] detektiert werden. Die Position und Lage der Bins wurde mit einem Analyse-durch-Synthese-Ansatz [WWS07] unter Verwendung eines geometrischen Modells der Bins bestimmt. Um dies zu ermöglichen, war im Vorfeld eine geometrische Kalibrierung der Kamera notwendig. Aufgrund der bekannten Lage der Bins war auch die Lage der Marker bekannt. Somit war es möglich, eine perspektivische Korrektur des Bildausschnitts durchzuführen, in dem der Marker enthalten war, um den 2D-Code des Markers (DataMatrix-Code [Len02]) zeilenweise auslesen zu können (Identifikation). Um die genaue Position der Bins im weiteren Verlauf zuverlässig vorhersagen zu können, wurde ein auf dynamischen Modellen basierendes probabilistisches Tracking-Verfahren verwendet, welches auf dem sog. Partikelfilter basiert und mit Hilfe des sog. Condensation-Algorithmus berechnet wird.

Der Einsatz von Kameras, die eine auf das Kameraobjektiv zurückzuführende, nicht-lineare Bildverzerrung (Linsenverzerrung) aufwiesen, führte zu Ungenauigkeiten bei der 3D-Lage- und Positionsbestimmung und insbesondere zu deutlichen Problemen beim zeilenweisen Auslesen des Codes. Um dieses Problem zu beheben, war eine Korrektur der Linsenverzerrung notwendig, deren Modellparameter im Rahmen einer im Vorfeld durchgeführten Kalibrierung bestimmt werden mussten.

1.2.2 LKW Ladungsüberwachung

Das nun geschilderte und noch nicht abgeschlossene Projekt wird mit Unterstützung eines Logistikunternehmens durchgeführt. Die Grundideen dazu entstammen einer Zusammenarbeit des Autors mit zwei weiteren Kollegen.

Transportunternehmen, die LKWs, wie in Abb. 1.2(a) dargestellt, verwenden, sind darauf angewiesen, die zur Verfügung stehende Ladefläche (siehe Abb. 1.2(c)) optimal zu nutzen, um unnötige Fahrzeiten und Fahrtstrecken zu vermeiden. Typischerweise werden während einer Tagestour Güter entnommen, aber auch neue Waren hinzugefügt. Die Abschätzung, ob kurzfristige, während der Tour eintreffende Routenänderungen sinnvoll sind, ist oftmals davon abhängig, welche Kapazitäten im Laderaum eines LKWs vorhanden sind.

Das Ziel des Projekts ist die automatische und fortlaufende Erfassung der sich im Laderaum des LKWs befindenden Güter. Dafür kann ein modellbasierter Rekonstruktionsansatz des Beladungszustands verwendet werden, der sowohl ein dreidimensionales geometrisches Modell für den LKW-Laderaum als auch für die transportierten Güter nutzt. Aufgrund der bekannten Maße des Laderaums kann dieser als Quader approximiert werden. Da in der Regel alle Güter auf Europaletten transportiert werden, die eine definierte Grundfläche besitzen, kann auch hier vereinfachend ein entsprechendes Quadermodell verwendet werden, wobei die Höhe der Beladung auf der Palette zu ermitteln ist. Da der LKW typischerweise nach dem Stack-Prinzip mit den Gütern be- und endladen wird (nur die vordersten Güter vom Eingang der Ladefläche aus betrachtet sind zugänglich) kann ein fortlaufend aktualisiertes Gesamtmodell, welches die Position und Größe aller Güterobjekte erfasst, aufgebaut werden. So lässt sich auch die freie Fläche und das freie Volumen im Laderaum bestimmen.

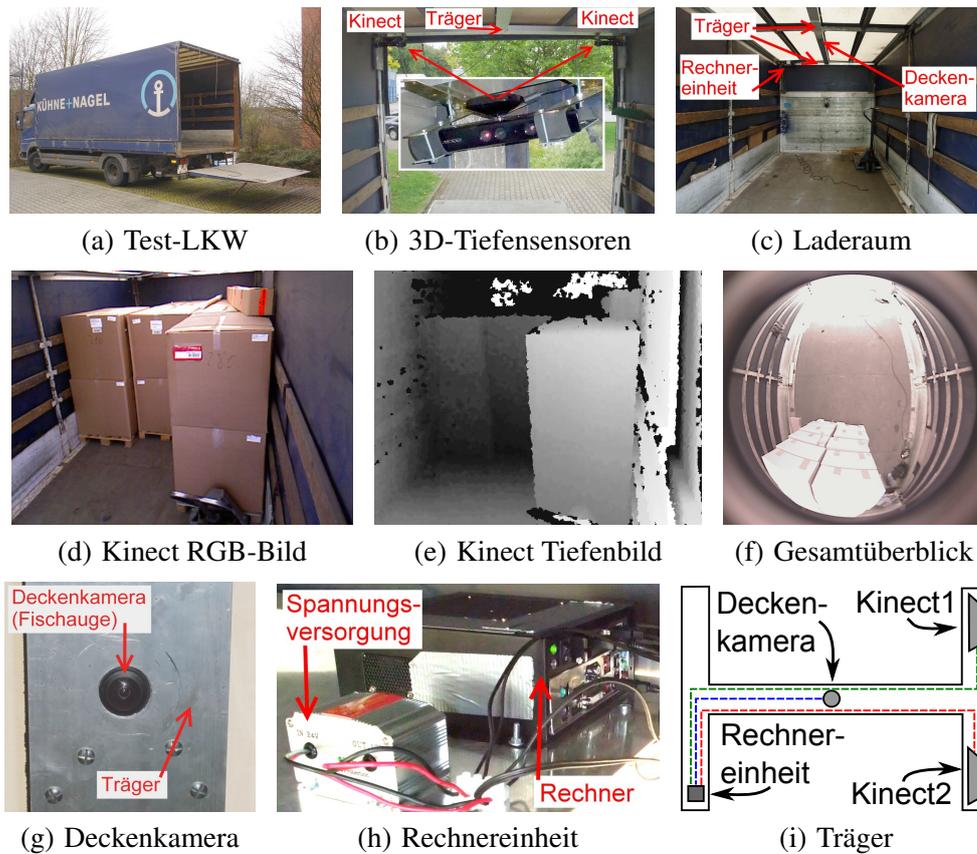


Abbildung 1.2: Der Laderaum (c) des im Rahmen des Projekts verwendeten LKWs (a) wurde mit 3D-Tiefensensoren (b) sowie einer Fischaugen-Kamera (g) ausgestattet. Die Sensordaten (d,e,f) wurden von der Rechneinheit (h) gesammelt und weiterverarbeitet. In dem unter der LKW-Decke montierten H-förmigen Aluminiumträger (i) sind alle Hardware-Komponenten sowie Daten- und Spannungsleitungen verbaut worden.

Zur Erfassung dreidimensionaler Informationen der Szene wurden zwei Kinect Tiefensensoren eingesetzt. Beide Kinects wurden links und rechts in den Ecken des Laderaums direkt am Eingangsbereich installiert (vgl. Abb. 1.2(b)) und in Richtung der Rückwand des Laderaums (siehe Abb. 1.2(c)) derart ausgerichtet, dass sie ein möglichst großes Laderaumvolumen erfassen konnten. Durch die in jeder Kinect verbaute RGB-Kamera ist zusätzlich zu den von den 3D-Tiefensensoren gelieferten Informationen eine stereobasierte 3D-Datengewinnung in dem von beiden Kameras beobachteten Bereich des Laderaums möglich. Die Abb. 1.2(d) zeigt das RGB-Kamerabild der rechten Kinect, in dem einige Güter zu erkennen sind. Das zeitgleich aufgezeichnete Tiefenbild ist in Abb. 1.2(e) dargestellt, dabei deuten unterschiedliche Helligkeiten unterschiedliche Entfernungen der beobachteten Objekte an. Zusätzlich wurde mittig unter der Decke des Laderaums eine handelsübliche 2D-Webcam mit einem speziellen Weitwinkel-Objektiv (Fischaugen-Objektiv) installiert (siehe Abb. 1.2(g)), um einen Gesamtüberblick des Laderaums zu ermöglichen (vgl. Abb. 1.2(f)). Alle Hardware-

komponenten inklusive der Rechereinheit wurden an einem Aluminiumträger unter der Decke des LKWs montiert (siehe Abb. 1.2(i)).

Um eine Fusion der Daten der unterschiedlichen Sensoren zu ermöglichen, muss die relative Lage aller Sensoren zu einem globalen Koordinatensystem bekannt sein, in dem das Beladungsmodell definiert ist. Dieses Problem kann durch eine extrinsische Kalibrierung gelöst werden [Zha98]. Dazu wurde das globale Koordinatensystem mit Hilfe eines in der Laderaumecke positionierten Kalibrierobjekts definiert. Das Kalibrierobjekt konnte von allen Sensoren (Fischaugen-Kamera, kinectinterne RGB-Kameras) erfasst und die Kalibrierung somit durchgeführt werden. Die relative Lage zwischen der kinectinternen RGB-Kamera und dem Tiefensensor kann mittels der Stereo-Kalibrierung [Sch07] ermittelt werden. Im Falle der Deckenkamera war insbesondere eine Korrektur der starken Linsenverzerrung notwendig. Im Allgemeinen ist die Qualität der Positionsbestimmung von der Güte der Verzerrungskorrektur abhängig, mit welchem Umfang bleibt jedoch eine offene Fragestellung.

In dem gegebenen Szenario sind die Sensoren verschiedenen Umwelteinflüssen, bedingt durch Jahres- und Tageszeiten, ausgesetzt. Beispielsweise können wechselnde Lichtverhältnisse einen Einfluss auf das Sensorrauschen der Kameras haben. Des Weiteren sind die Sensoren starken Temperaturschwankungen ausgesetzt, die, insbesondere im Fall der 3D-Tiefensensoren, einen signifikanten Einfluss auf die ermittelten Tiefendaten und somit auf das Laderaummodell haben können.

1.2.3 Oberflächenerfassung und Kartographierung mit Flugobjekten

Die Erarbeitung der Grundidee, der methodischen und mathematischen Vorgehensweisen sowie die Leitung des nun geschilderten Projekts wurden vom Autor, zusammen mit einem weiteren Kollegen durchgeführt. Die Realisierung des Konzepts erfolgte im Rahmen einer studentischen Projektgruppe. Details und Ergebnisse zu diesem Projekt sind in [BHK⁺10] zusammengefasst. Ziel des Projekts namens "FlyCam" war es, hoch aufgelöste Oberflächenkarten mit Hilfe von ferngesteuerten, kamerabestückten Flugobjekten zu generieren. Die aktuellen Kamerabilder sollten vom Flugobjekt per Funk an einen Rechner übertragen und dort bereits während des Flugs online zu einer Karte zusammengesetzt werden. Die Abb. 1.3(b) verdeutlicht die Grundidee. Die online erstellte Karte ermöglicht dem Piloten, der das Flugobjekt fernsteuert, ein Feedback über bereits erfasste und noch ausstehende Teile des gewünschten Gebiets zu erhalten. Als Flugobjekt wurde ein Mikrokopter, wie in Abb. 1.3(a) dargestellt, verwendet.

Die aus den Kamerabildern resultierenden 2D-Karten ähneln denen aus den bekannten Diensten "Google Earth"¹ oder "Microsoft Bing Maps"², sind allerdings aktueller und deutlich höher aufgelöst. Das entwickelte System in Kombination mit einem Mikrokopter kann z.B. bei der Einsatzplanung in Katastrophengebieten oder der Erfassung archäologischer Funde unter minimalem Risiko- und Kostenaufwand eingesetzt wer-

¹<http://earth.google.com>

²<http://www.bing.com/maps/>

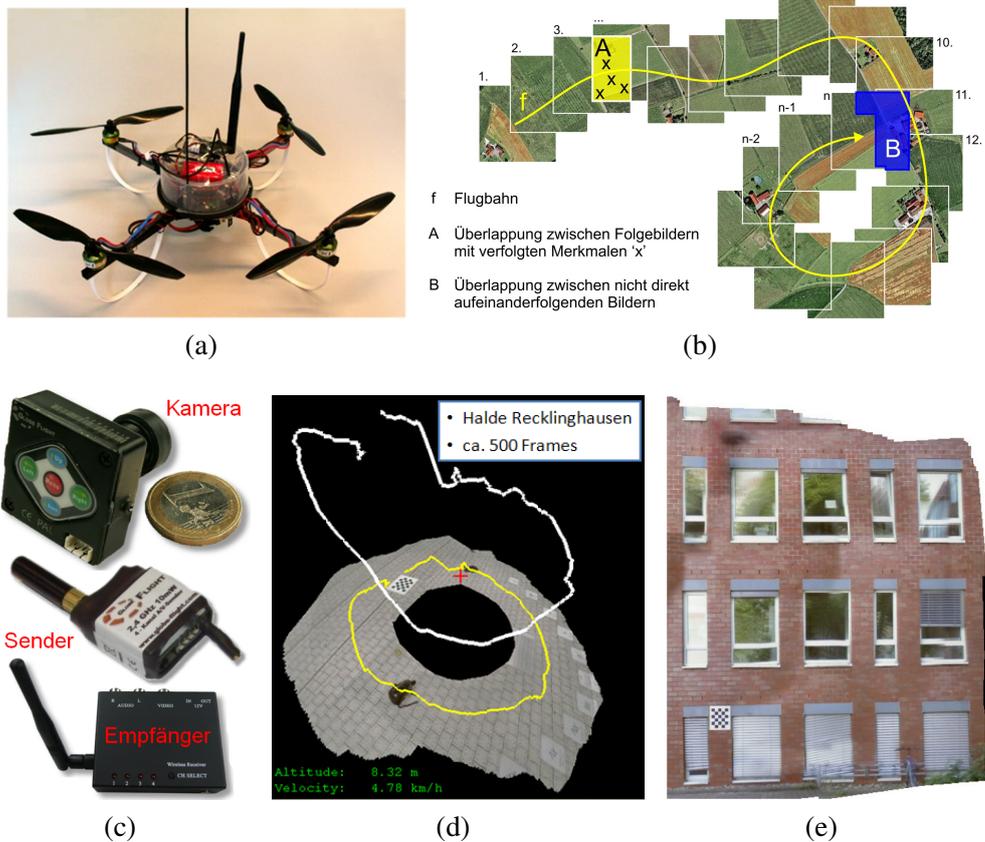


Abbildung 1.3: In (a) ist der ferngesteuerte Mikrokoopter gezeigt, an dem eine Mikro-kamera montiert wurde (c). Die Bildübertragung erfolgte mittels einer Analogfunkstrecke (c). Das Grundprinzip des Kartenaufbaus aus Einzelbildern ist in (b) verdeutlicht. (d) zeigt die 3D-Visualisierung eines Zwischenergebnisses (Kreisflug) aus 500 Einzelbildern inklusive Flugbahn (weiß) und ihrer Projektion auf den Boden (gelb). Bei Änderung der Blickrichtung der Kamera sind auch Fassadenaufzeichnungen (400 Einzelbilder) möglich (e). (Bildquelle: [BHK⁺ 10])

den. Durch eine Anpassung der Kameraausrichtung sind z.B. detaillierte Erfassungen von Gebäudefassaden und deren Überprüfung möglich (siehe Abb. 1.3(e)).

Aufgrund der geringen Nutzlast des Mikrokoopters wurde eine leichte Analogkamera und ein analoges Bildübertragungssystem verwendet (siehe Abb. 1.3(c)). Die aufgrund der Analogtechnik entstandenen Artefakte (z.B. Interlacing [WT08]) mussten nach der Digitalisierung in einem Vorverarbeitungsschritt behandelt werden. Anschließend sind im aktuellen und vorangegangenen Bild, bei denen eine Überlappung vorausgesetzt werden konnte, korrespondierende Punkte mit Hilfe von SIFT-Merkmalen [Low04] berechnet worden (vgl. Bereich A in Abb. 1.3(b)). Auf der Basis dieser Punkte und unter Verwendung des RANSAC-Algorithmus [FB81] wurde die Homographie [HZ03] zwischen den Bildern bestimmt. Mit Hilfe der Homographie konnte das aktuelle Bild derart verzerrt werden, dass es in die Bildebene des Vorgängerbildes eingefügt werden

konnte. Dies war insbesondere bei Flugmanövern von Relevanz, bei denen die Kamera nicht senkrecht zum Boden ausgerichtet war. Es folgte das sog. Stitching [Sze06] der überlappenden Bilder. Über die Zeit entstand ein Mosaik aus den Einzelbildern, welches schließlich die resultierende Karte ergab.

Eine rein auf den Bilddaten basierende Ermittlung der Flugbahn war ebenfalls ein zu lösendes Problem. Dazu wurde auf die zu erfassende Oberfläche ein planares Kalibrierobjekt gelegt, welches als sichere Startunterlage und als Definition eines Weltkoordinatensystems diente. Nach dem Start wurde das Kalibrierobjekt von der Kamera erfasst und ihre 3D-Flugposition relativ zum Weltkoordinatensystem bestimmt. Mit Hilfe der fortlaufend berechneten Homographien und der bekannten Kameraposition zum Zeitpunkt $t - 1$ konnte auf die 3D-Position zum Zeitpunkt t geschlossen werden, ohne dass dabei das Kalibrierobjekt zu beobachten war. Abb. 1.3(d) zeigt das Zwischenergebnis eines Kreisflugs mit der ermittelten Flugbahn. Der sich über die Zeit aufakkumulierte Abweichungsfehler konnte durch die Detektion bereits überflogener und im System erfasster Gebietsteile während des Flugs korrigiert werden (vgl. Bereich B in Abb. 1.3(b)). Neben den Online-Funktionalitäten wurden auch Offline-Korrekturen realisiert, die die Qualität der ermittelten Karte in einem globalen Optimierungsschritt verbesserten.

Die beschriebene Kamerapositionsbestimmung und die damit verbundene Kartengenerierung erforderte kalibrierte Kameras. Dabei war die Güte der Kalibrierung entscheidend für die Qualität der Ergebnisse. Aufgrund der Nebenbedingungen (leichte Bauweise, geringer Energieverbrauch, geringe Kosten) konnten keine hochwertigen Kameras mit großen Objektiven eingesetzt werden, so dass die Kamerabilder eine deutliche, nichtlineare Linsenverzerrung aufwiesen. Eine möglichst genaue Korrektur der Verzerrung war insbesondere für eine zuverlässige und korrekte Berechnung der Homographien unerlässlich und bildete einen wichtigen Bestandteil bei der Vorverarbeitung des Online-Bilddatenstroms.

Bei der Übertragung des Projekts auf reale Anwendungsfälle im größeren Maßstab (große Flugdrohne) stellt sich die Frage nach einem potentiellen Einfluss von Temperaturschwankungen, bedingt durch das Klima oder die Flughöhe, auf die Kameraoptik und somit auf das Gesamtsystem, der gegebenenfalls im Sensormodell zu berücksichtigen ist.

1.3 Zielsetzung und Beiträge

Wie im vorherigen Abschnitt dargelegt, ist die möglichst genaue Kenntnis des Abbildungsverhaltens von Kameras und 3D-Tiefensensoren eine wesentliche Voraussetzung für zahlreiche Anwendungen des Visual Computing. Diese Dissertation liefert Beiträge zur Analyse des Abbildungsverhaltens von Kameras und 3D-Tiefensensoren sowie zu deren Modellierung und Kalibrierung, die eine verbesserte Ermittlung des Abbildungsverhaltens ermöglichen. Die damit verbundenen übergeordneten Ziele sind die Steigerung der Realitätsnähe der erfassten Daten und Abschätzungen der Abweichung der erfassten Daten von der Realität.

1.3.1 Kameras

Gängige Kalibrierverfahren von Kameras schätzen in einem ersten Schritt die sog. intrinsischen und extrinsischen Modellparameter. Erst in einem zweiten Schritt wird typischerweise ein globaler Optimierungsansatz durchgeführt, der zusätzlich zu den intrinsischen und extrinsischen Parametern die des Linsenverzerrungsmodells mit berücksichtigt. Dabei wird die Parameteroptimierung üblicherweise anhand einer Zielfunktion durchgeführt, die von den vom Sensor zur Verfügung gestellten Daten abhängig ist. Insbesondere aufgrund der extrinsischen Parameter kommt es schnell zu einer großen Gesamtparameteranzahl, die sich negativ auf die Stabilität numerischer Optimierungsverfahren auswirken kann. Steht lediglich die präzise Korrektur der Linsenverzerrung im Vordergrund, wie dies in der vorliegenden Dissertation der Fall ist, so stellt die Mitoptimierung der intrinsischen und insbesondere der zahlreichen extrinsischen Parameter ein Hindernis dar.

Für die Optimierung werden typischerweise deterministische Verfahren wie z.B. der Levenberg-Marquardt-Algorithmus [Mor77] eingesetzt. Diese sind nur dann in der Lage, ein globales Optimum zu erreichen, wenn sich die geschätzten Parameter bereits in der Nähe des globalen Optimums befinden. Anderenfalls kann nur ein lokales Optimum erreicht werden. Somit ist eine genaue Schätzung der intrinsischen und extrinsischen Parameter im ersten Schritt maßgeblich für das Erreichen eines globalen Optimums. Dabei ist jedoch problematisch, dass bei den Schätzungen typischerweise die in den Sensordaten vorhandene Linsenverzerrung zunächst ignoriert wird, die Schätzung also auf fehlerhaften Daten beruht, was folglich die Qualität der Schätzung und somit auch die des Optimierungsergebnisses beeinträchtigen kann. Ein weiteres Problem stellt das Rauschen in den vom Sensor zur Verfügung gestellten Daten dar, welches eine Auswirkung auf die von diesen Daten abhängige Zielfunktion hat. Durch dieses Rauschen können zusätzliche lokale Minima entstehen, die zu einer vorzeitigen Terminierung der deterministischen Optimierungsverfahren führen können. Die Folge sind suboptimale Linsenverzerrungsparameter, die folglich auch nur eine suboptimale, von diesen Parametern abhängige Korrektur der Verzerrung zulassen.

Die Dissertation präsentiert ein neuartiges Verfahren zur Linsenverzerrungskorrektur, das bessere Korrekturleistungen als gängige alternative Verfahren bietet. Dieses kann auch unter schwierigen Bedingungen, wie z.B. bei starker Verzerrung oder deutlichem Rauschen, stabilere Korrekturergebnisse liefern.

Eine Besonderheit des Verfahrens ist die Wahl der Zielfunktion. Diese basiert auf der Eigenschaft von projektiven Kameras, die gerade Linien einer Szene auf gerade Linien im Kamerabild abbilden. Wird diese Eigenschaft verletzt, so wird dies auf eine vorhandene Linsenverzerrung zurückgeführt [DF01]. Es wird gezeigt, wie diese Eigenschaft der Zielfunktion mit Hilfe von einfachen Kalibrierobjekten automatisch, präzise und zuverlässig ermittelt werden kann.

Eine weitere Besonderheit ist das Verfahren zur Optimierung der Linsenmodellparameter, wobei diese ohne die Abhängigkeit von den intrinsischen und extrinsischen Parametern bestimmt werden können. Dabei wird ein auf dem Partikelfilter [DJ11] basierender Ansatz, konkret der Condensation-Algorithmus [IB98], der seinen Ursprung im Kontext der Objektverfolgung hat (Tracking), derart angepasst, dass er die beschrie-

bene Zielfunktion nutzt, um die gesuchten Modellparameter stochastisch und iterativ zu optimieren.

Das vorgeschlagene Verfahren wurde in zahlreichen Experimenten mit verschiedenen Kameras und Objektiven untersucht, von denen einige Resultate im Folgenden zusammengefasst werden. Trotz der Zufallsprozesse im Optimierungsverfahren, war die erreichte Entzerrungsqualität bei 1000 getesteten Durchgängen stets sehr stabil (Variationskoeffizient von 0.71%). Das Verfahren wurde zunächst mit einem alternativen Verfahren verglichen, welches durch die OpenCV-Bibliothek [BK08] zur Verfügung gestellt wird. Bei Zoom-Objektiven mit variabler Brennweite konnte der nach der Korrektur verbliebene Verzerrungsfehler (Residuum) im Vergleich, insbesondere bei kurzen Brennweiten, deutlich auf bis zu 17% gegenüber dem Alternativverfahren herabgesenkt werden. Bei der Handhabung unscharfer Bilddaten konnte der Fehler auf bis zu 76% verringert werden. Der störende Einfluss des Rauschens, insbesondere bei wachsendem Rauschen, war bei dem vorgeschlagenen Verfahren deutlich geringer (siehe Abb. 3.20). Die Untersuchung von Fischaugen-Kameras beim Vergleich mit einem anderen Verfahren (OCamCalib [Sca05]) ergab ebenfalls, je nach Objektiv, eine Fehlerreduktion auf 61% bis 71%. In einem direkten Vergleich der Optimierer (stochastisch gegen deterministisch) lieferte der Einsatz des in dieser Dissertation beschriebenen stochastischen Optimierers bessere Ergebnisse. Insbesondere in Gegenwart von deutlichem Rauschen vergrößerte sich die Differenz zwischen den Resultaten (verbliebener Verzerrungsfehler) beider Verfahren auf das 2.5-fache gegenüber dem Fall ohne Rauschen.

1.3.2 3D-Tiefensensoren

Im Gegensatz zu Kameras, die im Wesentlichen aus einem Objektiv und einem Sensorchip bestehen, stellen 3D-Tiefensensoren deutlich komplexere Systeme dar, die sich in der Regel aus mehreren Komponenten zusammensetzen, deren Zusammenwirkung die 3D-Informationsgewinnung erst ermöglicht. Bedingt durch technische Einschränkungen, Fertigungstoleranzen oder Rauschen können die Einzelkomponenten (Einzelsensoren) fehlerhaft sein bzw. fehlerhaft arbeiten, und durch das komplexe Zusammenspiel in einem Fehler der 3D-Informationsgewinnung resultieren. Liegen die vom 3D-Tiefensensor ermittelten Informationen als Tiefen- oder Distanzwerte in einer physikalischen Bezugsgröße wie z.B. Millimetern vor, so können diese Informationen auch als Tiefenmessung aufgefasst, und fehlerhafte Informationen als messfehlerbedingt interpretiert werden.

Die Dissertation befasst sich mit der Analyse und Korrektur von Tiefenmessfehlern. Dies geschieht anhand des 3D-Tiefensensors Kinect. Das Prinzip der Vorgehensweise ist jedoch auf andere Tiefensensoren übertragbar, insbesondere auf solche, die auf strukturiertem Licht basieren. Im Einzelnen werden die im Folgenden aufgeführten Beiträge präsentiert.

Aufdeckung der Messfehlerbeschaffenheit. Der Beitrag der Dissertation zur Analyse des Tiefenmessfehlers liegt im Aufzeigen geeigneter Experimente und Vorgehensweisen, die in der Lage sind, die Beschaffenheit und Struktur des Fehlers aufzudecken.

Als Resultat der Experimente mit planaren Objekten konnte ein Messfehler nachgewiesen werden (z.B. eine Abweichung von bis zu 57 mm bei ca. 1.6 m Distanz), der eine komplexe ortsabhängige Struktur im Sensorfeld und eine Abhängigkeit von der Messtiefe aufwies.

Verfahren zur Messfehlerkorrektur. Auf diesen Erkenntnissen aufbauend, wird als weiterer Beitrag ein Korrekturverfahren zur Verbesserung der Qualität der Tiefenmessdaten vorgeschlagen. Das Verfahren besteht in einem heuristischen Korrekturmodell der Tiefeninformation. Es benutzt auf effiziente Weise Look-Up Tabellen, die auf empirischen Messdaten planarer Objekte beruhen. Es ermöglicht eine Korrektur zur Laufzeit (online). Mit Hilfe dieses Korrekturverfahrens konnte der Fehler bis auf 29% reduziert werden. Die Korrekturleistung in natürlichen Szenen ist anhand der Abb. 5.15 erkennbar.

Online-Referenzmessdatengewinnung. Im Zusammenhang mit der Online-Korrektur von Messfehlern steht auch die interaktive Überprüfbarkeit der Messgenauigkeit durch die Online-Gewinnung von Referenzmessdaten. Eine solche Überprüfung ist insbesondere bei Tiefensensoren interessant, die in der Lage sind, 3D-Daten in Echtzeit zu liefern. Hierfür präsentiert die Dissertation eine entsprechende Vorgehensweise.

Aufdeckung umweltbedingter Auswirkungen. Des Weiteren greift die Dissertation das Problem der Auswirkung von Umwelteinflüssen auf die Tiefenmessung auf. Dazu werden Experimente beschrieben und durchgeführt, in denen Umwelteinflüsse kontrolliert manipuliert und ihre Auswirkungen systematisch untersucht werden. Dabei wird ermittelt, wie sich Umwelteinflüsse auf die einzelnen Komponenten auswirken und wie diese Auswirkungen mit dem Einfluss auf das Gesamtmesssystem zusammenhängen.

Die Temperatur stellte den wichtigsten experimentell ermittelten Umwelteinfluss dar. Neben Auswirkungen auf die kinectinternen Kameras (Abbildungsverschiebungen von bis zu 6.7 Pixeln, Änderungen intrinsischer Parameter) konnten Änderungen in den Tiefenmesswerten festgestellt werden, wie z.B. während der Aufwärmphase nach der Aktivierung der Kinect (ca. 20 mm), bei der automatischen Aktivierung des kinect-internen Lüfters (12 mm) oder bedingt durch natürliche Temperaturschwankungen von ca. 2 mm pro 1°C.

Regeln zur Fehlervermeidung. Basierend auf diesen Ergebnissen werden Regeln hinsichtlich der Umwelteinflüsse abgeleitet, die zur Fehlervermeidung bei der Kalibrierung und Verwendung des Kinect-Sensors verhelfen. Konkret wurden sechs Regeln aufgestellt, wie z.B. die Notwendigkeit der Beachtung einer bis zu 60-minütigen Aufwärmzeit.

Experimenteller Nachweis der Robustheit des Korrekturverfahrens. Aufgrund der Erkenntnisse bzgl. der Umwelteinflüsse ist eine Abhängigkeit des vorgeschlagenen Korrekturverfahrens des komplexen ortsabhängigen Messfehlers von der Temperatur denkbar. Durch eine entsprechende experimentelle Untersuchung konnte jedoch Unabhängigkeit nachgewiesen werden.

1.3.3 Eigenanteil der Beiträge

Die in der Dissertation erarbeiteten Beiträge, Experimente und Resultate beruhen auf eigenen Ideen des Autors, bei deren Konkretisierung Herr Prof. Dr. Müller beratend zur Seite stand.

Die im vorangegangenen Abschnitt beschriebenen Beiträge zur *Aufdeckung umweltbedingter Auswirkungen* und die damit verbundenen *Regeln zur Fehlervermeidung* sind in der Publikation [FM13] veröffentlicht worden, die der Autor der Dissertation verfasst hat. Die Ideen und Experimente wurden vom Autor eigenständig entwickelt und umgesetzt. Der Beitrag des Koautors (Prof. Dr. Müller) bei dieser Publikation beschränkt sich auf technische Diskussionen zu Einzelaspekten.

Zwei weitere Publikationen [WFH⁺10] und [TWF⁺11] stehen im Motivationskontext in der Einleitung der Dissertation (siehe Abschnitt 1.2.1), sind jedoch nicht Gegenstand des wissenschaftlichen Beitrags der Arbeit.

1.4 Gliederung der Arbeit

Die vorliegende Dissertation lässt sich grob in zwei Teile gliedern, wobei die Kapitel 2 und 3 den ersten Teil darstellen, der schwerpunktmäßig dem Gebiet der Kameras gewidmet ist. Gegenstand des zweiten Teils der Arbeit, bestehend aus den Kapiteln 4 und 5, sind 3D-Tiefensensoren. Die beiden Teile sind jedoch keineswegs als disjunkt, sondern vielmehr als aufeinander aufbauend zu betrachten. Jeder der beiden Teile ist derart strukturiert, dass die jeweiligen ersten Kapitel (Kapitel 2 und 4) als Einführung in das entsprechende Themengebiet zu verstehen sind. Dabei werden weitestgehend bekannte Verfahren und Konzepte aufgearbeitet, zusammengeführt und stellenweise aus anderen Perspektiven dargestellt, um für die Arbeit relevante Aspekte hervorzuheben und gleichzeitig eine kompakte Darbietung zu ermöglichen. Die Kapitel 3 und 5 stellen Resultatskapitel dar, in denen eigene, neue Methoden und Verfahren vorgestellt oder bereits bestehende miteinander kombiniert werden. Außerdem werden dort Experimentdurchführungen beschrieben und Ergebnisse entsprechender Evaluierungen präsentiert. Das Kapitel 6 schließt die Arbeit mit einer kurzen Zusammenfassung ab.

2

Modellierung und Kalibrierung von Kameras

Das Kapitel 2 dient der Einführung in das Themengebiet der Kameras mit dem Schwerpunkt ihrer Modellierung und Kalibrierung. Dabei werden Verfahren und Methoden aus der Literatur derart ausgewählt, aufgearbeitet und zusammengeführt, dass sie die relevanten Aspekte abdecken und hervorheben, auf denen das nachfolgende Kapitel 3 aufbaut.

Für die meisten Anwendungen ist es notwendig, das Abbildungsverhalten einer Kamera zu kennen, um beschreiben zu können, wie dreidimensionale Objekte in der Welt auf die zweidimensionale Bildebene, die im Fall von Digitalkameras aus einem 2D-Pixelsensor besteht, abgebildet werden. Ist dieses Abbildungsverhalten bekannt, ist es möglich, Rückschlüsse auf die durch die Kamera beobachtete Welt zu ziehen. Dies kann z.B. die Ermittlung der Entfernung, Lage oder Größe von Objekten sein. Typischerweise wird dieses Abbildungsverhalten mit Hilfe eines Modells beschrieben, dessen Modellparameter vor der Anwendung zu bestimmen sind.

Nach einer kurzen Beschreibung des technischen Aufbaus typischer Kameras (Abschnitt 2.1), wird zunächst das Lochkameramodell beschrieben und sowohl auf die intrinsischen als auch auf die extrinsischen Kameraparameter eingegangen (Abschnitt 2.2). Anschließend wird das Kameramodell um ein Linsenverzerrungs-Modell erweitert (Abschnitt 2.3). Es folgt die Darstellung eines typischen Kalibrierungsansatzes zur Bestimmung der Modellparameter (Abschnitt 2.4). Anschließend werden sog. Forward- und Backward-Modelle diskutiert (Abschnitt 2.5) und die damit im Zusammenhang stehende Linsenverzerrungsfunktion geschildert, welche die durch das Objektiv verursachte Verzerrung aufhebt. Zudem wird eine Entkopplung des Linsenmodells vom restlichen geometrischen Modell der Kamera beschrieben (Abschnitt 2.5.3), so dass eine separate Betrachtung sowie Kalibrierung des Linsenmodells ermöglicht wird. Diese separate Betrachtung des Linsenmodells ist schließlich Ausgangspunkt für das Kapitel 3.

2.1 Technischer Aufbau

Die beiden wesentlichen Komponenten einer digitalen Kamera sind ein 2D-Pixelsensor und ein Objektiv. Das Objektiv erfasst Lichtstrahlen der beobachteten Szene und lenkt diese auf den im Kameragehäuse verbauten Sensor, auf dem ein zweidimensionales Abbild der Szene entsteht. Der Sensor dient der Erfassung der Information der Helligkeitsverteilung der abgebildeten Szene auf der Sensorfläche und der Weitergabe dieser

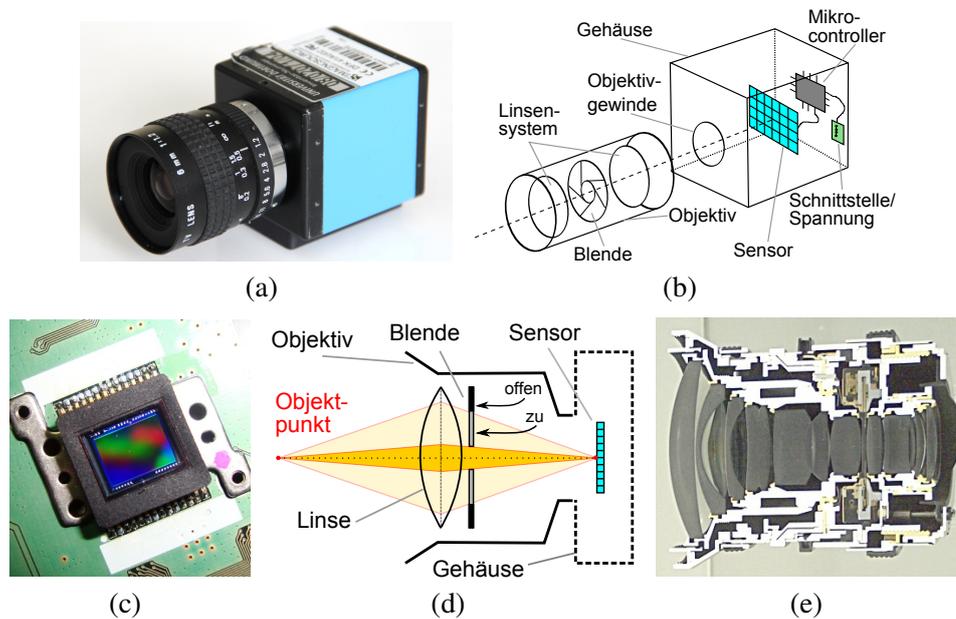


Abbildung 2.1: In (a) wird als Beispiel für eine Kamera die Industriekamera "IS DFK 21AU618" gezeigt. Der schematische Aufbau ist in (b) dargestellt. Das Objektiv besteht im Wesentlichen aus einer Linse bzw. einem Linsensystem und einer Blende. Die wesentlichen Komponenten im Inneren des Gehäuses sind der 2D-Pixelsensor sowie ein Mikrocontroller zur Aufbereitung des Bildsignals und zur Weitergabe der Daten über eine entsprechende Schnittstelle. In (c) ist ein 6 Megapixel CCD-Sensor abgebildet. (d) verdeutlicht das Abbildungsprinzip anhand eines einfachen Objektivs. In (e) ist beispielhaft ein komplexes Zeiss-Distagon-Objektiv im Querschnitt gezeigt. (Bildquelle (e): Klaus-Dieter Grüniger, (CC: BY-NC-SA), <http://www.schule-bw.de>)

Information an einen typischerweise ebenfalls im Gehäuse vorhandenen Mikrocontroller, der die Information verarbeitet und mittels einer Schnittstelle z.B. an einen Computer überträgt. In Abb. 2.1(a) ist ein Beispiel für eine digitale Industriekamera aus dem Hause "Imaging Source" abgebildet (DFK 21AU618). Die einzelnen genannten Komponenten sind schematisch in Abb. 2.1(b) aufgeführt.

Der Sensor besteht aus vielen einzelnen fotoempfindlichen Elementen, die in Form einer zweidimensionalen Matrix angeordnet sind, und die in den meisten Fällen auf der CCD- oder der CMOS-Technologie basieren. Abb. 2.1(c) zeigt einen CCD-Sensor einer handelsüblichen Kompaktkamera, mit einer Auflösung von 2840×2130 Pixeln (6 Megapixel). Jedes fotoempfindliche Element ist in der Lage, einfallendes Licht in elektrische Ladung umzuwandeln. Die entstehende Ladung ist in etwa proportional zur Menge des einfallenden Lichts. Die Ladung wird über einen kurzen Zeitraum (Integrationszeitraum) angesammelt und zwischengespeichert. Anschließend erfolgt eine von der Technologie abhängige Auslesephase, in der die gespeicherte Ladung durch entsprechende integrierte Schaltkreise in eine Spannung umgewandelt wird. Diese Spannung ist proportional zur Ladung und somit zum über den Integrationszeitraum eingefallenen Licht. Über entsprechende Analog-Digital-Wandler steht die Lichtmenge

somit auch als digitaler Wert zur Verfügung. Diese Informationen aller Sensorelemente können in digitaler Form weiter verarbeitet werden. Schließlich steht ein digitales Bild der Szene zur Verfügung. Aufgrund einer Reihe unterschiedlicher physikalischer und elektrischer Effekte (Dunkelrauschen, Ausleserauschen, Photonenrauschen usw.) entspricht das erfasste nicht dem ursprünglichen Bildsignal, sondern ist mit einer zufallsbedingten Abweichung überlagert, die Rauschen genannt wird. Rauschen ist ein Qualitätsmerkmal von digitalen Sensoren und ist unter anderem von der Größe der Sensorelemente, der Gesamtsensorgröße und der zur Verfügung stehenden Lichtmenge abhängig. Details zur Funktionsweise der CMOS- und CCD-Technologie sind z.B. in [ALL⁺07] und [Luh10] zu finden.

Die Menge des auf den Sensor einfallenden Lichts ist unter anderem von dem verwendeten Objektiv abhängig. Es besteht im Wesentlichen aus einer Linse bzw. einem Linsensystem, das das einfallende Licht auf den Sensor bündelt, sowie einer Blende, über die (unter anderem) die Lichtmenge reguliert werden kann. Die Abb. 2.1(d) zeigt schematisch ein vereinfachtes Objektiv, bestehend aus nur einer Linse und einer variablen Blende. Von einem Objektpunkt (rot) gehen Lichtstrahlen in alle Richtungen aus, von denen einige auf die Linse treffen. Die Linse lenkt diese Lichtstrahlen so um, dass sie im Idealfall wieder auf einen einzigen Punkt auf den Sensor treffen. Dabei bestimmt die Blende, ob alle auf die Linse treffenden Strahlen den Sensor erreichen (Blende offen), oder nur die weiter in der Linsenmitte verlaufenden Strahlen die Blende passieren können. Nach diesem Prinzip kann die Lichtintensität, die die Sensorelemente erreicht, geregelt werden.

Einfache Kameras (z.B. günstige Webcams, einfache Handycameras) besitzen oftmals nur eine kleine Linse und eine entsprechend kleine Blende, so dass wegen der geringen Lichtmenge tendenziell höheres Rauschen im Bild zu erwarten ist, während lichtstarke Objektive große Linsen verwenden. Diese sind in der Regel komplexer und bestehen aus einer Vielzahl von Linsen, wie z.B. in Abb. 2.1(e) gezeigt. Dort ist der Querschnitt von einem Zeiss-Objektiv vom Typ "Distagon" mit über 10 unterschiedlichen Linsen abgebildet. Der Strahlengang bei derartigen Objektiven ist um ein Vielfaches komplexer als in der Darstellung in Abb. 2.1(d). Die Abbildungsgeometrie, die zahlreichen optischen Effekte und die Fehlerquellen bei der Abbildung wie etwa die chromatische Aberration, die dabei eine Rolle spielen, sollen hier nicht weiter ausgeführt werden. Weiterführende Details können z.B. in [Hec05] und [Luh10] nachgeschlagen werden.

Bestimmte Arten des typischerweise durch Objektive verursachten Verzerrungsfehlers werden zu einem späteren Zeitpunkt gesondert beschrieben (Abschnitt 2.3), da dieser erst für weiterführende Teile der Arbeit relevant sein wird. Der folgende Abschnitt erläutert ein von der Realität abstrahierendes, mathematisches Kameramodell, welches für die meisten Anwendungsgebiete, in denen Kameras eingesetzt werden, verwendet wird.

2.2 Lochkameramodell

Die Grundidee für das nachfolgend beschriebene Kameramodell stammt von der Lochkamera ab, die in den Anfängen der Fotografie genutzt wurde und in Abb. 2.2(a) skizziert ist. Bei der Lochkamera fallen Lichtstrahlen einer Szene durch ein einzi-

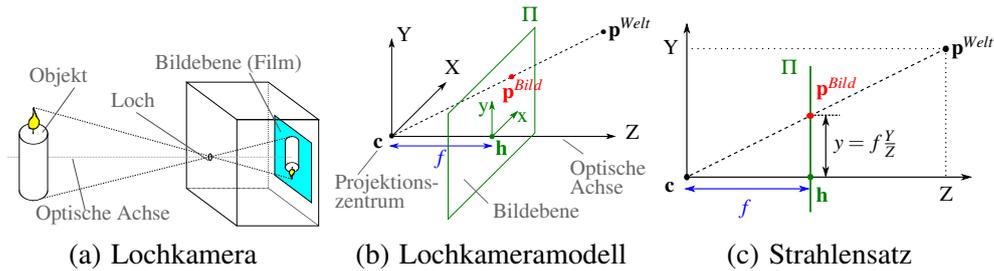


Abbildung 2.2: In (a) wird die Funktionsweise der Lochkamera anhand einer Skizze demonstriert. Das Objekt in der Szene (Kerze) wird auf die Bildebene innerhalb der Lochkamera abgebildet. Durch die Verschiebung der Bildebene Π vor das Projektionszentrum \mathbf{c} ergibt sich das typische Lochkameramodell [HZ03, MT96] mit dem Hauptpunkt \mathbf{h} und der Brennweite f , wie in (b) dargestellt. Die Berechnung der y -Koordinate des projizierten Punktes $\mathbf{p}^{Bild} = (x, y)^T$ mit Hilfe des Strahlensatzes wird in (c) verdeutlicht.

ges kleines Loch in das Innere einer abgedunkelten Zelle. Auf ihrer Rückseite, die der Projektionsebene entspricht, befindet sich ein Film oder Fotopapier. Ein Objekt in der Szene wird punktgespiegelt auf die Bildebene abgebildet. Das Loch stellt dabei das Projektionszentrum dar, und der Abstand zwischen dem Loch und der Bildebene wird Brennweite genannt. Die optische Achse entspricht dem Strahl, der senkrecht zur Bildebene steht und durch das Projektionszentrum verläuft. Der Schnittpunkt von optischer Achse und Bildebene wird Hauptpunkt (engl. "principal point") genannt.

Durch eine Verschiebung der Bildebene entlang der optischen Achse auf die gegenüberliegende Seite des Projektionszentrums erhält man das typische Lochkameramodell, wie es z.B. in [HZ03] oder [MT96] definiert und in Abb. 2.2(b) dargestellt ist. Dabei wird das Projektionszentrum $\mathbf{c} = (0, 0, 0)^T$ als Ursprung eines kartesischen Koordinatensystems angenommen, dessen Z -Achse der optischen Achse entspricht und die Bildebene Π im Hauptpunkt \mathbf{h} schneidet. Die Brennweite f entspricht der Distanz zwischen \mathbf{c} und Π . Die Bildebene wiederum besitzt ihr eigenes zweidimensionales Koordinatensystem, welches im Hauptpunkt entspringt. Wird ein Punkt $\mathbf{p}^{Welt} \in \mathbb{R}^3$ nun auf die Bildebene projiziert, so befindet sich dieser an dem Schnittpunkt der Bildebene mit dem Strahl, der durch \mathbf{p}^{Welt} und das Projektionszentrum \mathbf{c} definiert wird. Daher wird diese Projektion auch Zentralprojektion genannt. Sei $\mathbf{p}^{Bild} = (x, y)^T$ der Punkt in der Bildebene, der sich durch die Projektion von $\mathbf{p}^{Welt} = (X, Y, Z)^T$ ergibt. Dann gilt durch die Anwendung des Strahlensatzes die folgende Beziehung zwischen den Koordinaten der beiden Punkte:

$$\frac{x}{f} = \frac{X}{Z} \Leftrightarrow x = f \frac{X}{Z}, \quad (2.1)$$

$$\frac{y}{f} = \frac{Y}{Z} \Leftrightarrow y = f \frac{Y}{Z}. \quad (2.2)$$

Der Zusammenhang wird unter Verwendung des Strahlensatzes für die Berechnung von y durch Abb. 2.2(c) verdeutlicht.

Um die Zentralprojektion bequem mit Hilfe von Matrixmultiplikationen beschreiben

zu können, wird die homogene Darstellung der Vektoren genutzt. Dabei ist allgemein die Transformation $\mathbf{v} \mapsto \tilde{\mathbf{v}}$ eines Vektors von kartesischen zu homogenen Koordinaten sowie umgekehrt gegeben durch

$$\mathbf{v} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix} \mapsto \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \\ 1 \end{pmatrix} = \tilde{\mathbf{v}} \quad \text{sowie} \quad \tilde{\mathbf{v}} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \\ v_n \end{pmatrix} \mapsto \begin{pmatrix} v_0/v_n \\ v_1/v_n \\ \vdots \\ v_{n-1}/v_n \end{pmatrix} = \mathbf{v}. \quad (2.3)$$

Somit sind homogene Vektoren bis auf einen Skalierungsfaktor eindeutig. Seien nach dieser Definition die homogenen Darstellungen der Punkte $\mathbf{p}^{Welt} = (X, Y, Z)^T$ und $\mathbf{p}^{Bild} = (x, y)^T$ gegeben durch $\tilde{\mathbf{p}}^{Welt} = (X, Y, Z, 1)^T$ und $\tilde{\mathbf{p}}^{Bild} = (x, y, 1)^T$. Unter Verwendung der Gleichungen (2.1) und (2.2) ergibt sich der folgende Zusammenhang zwischen $\tilde{\mathbf{p}}^{Welt}$ und $\tilde{\mathbf{p}}^{Bild}$:

$$Z \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = Z \begin{pmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \\ 1 \end{pmatrix} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{M}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.4)$$

Entsprechend [Sch07] lautet die Projektionsgleichung in kompakter Schreibweise

$$s \tilde{\mathbf{p}}^{Bild} = \mathbf{M} \tilde{\mathbf{p}}^{Welt}, \quad (2.5)$$

mit der 3×4 Matrix \mathbf{M} , die Projektionsmatrix genannt wird, und einem beliebigen homogenen Skalierungsfaktor $s \neq 0$. Da $\tilde{\mathbf{p}}^{Bild}$ in homogener Darstellung vorliegt, hat eine Skalierung mit einem beliebigen Faktor $s \neq 0$ keinen Einfluss auf den eigentlichen Punkt \mathbf{p}^{Bild} . Im Fall der Gleichung (2.4) gilt $s = Z$. Für beispielsweise $\tilde{\mathbf{p}}^{Welt} = (2X, 2Y, 2Z, 1)^T$ würde $s = 2Z$ gelten.

2.2.1 Intrinsische und extrinsische Kameraparameter

Die Projektionsmatrix \mathbf{M} aus Gleichung (2.5) kann dargestellt werden als Produkt von zwei weiteren Matrizen $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ und $\mathbf{A} \in \mathbb{R}^{3 \times 4}$, wobei sich $\mathbf{A} = [\mathbf{R} \ \mathbf{t}]$ aus einer Rotationsmatrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ und einem Translationsvektor $\mathbf{t} \in \mathbb{R}^{3 \times 1}$ zusammensetzt, so dass $\mathbf{M} = \mathbf{K}\mathbf{A}$ gilt. Die Matrix \mathbf{K} wird intrinsische Matrix oder Kamerakalibrierungsmatrix genannt und speichert die intrinsischen Parameter der Kamera, die bislang nur die Brennweite f beinhaltet, jedoch im Folgenden noch erweitert wird. Die Matrix \mathbf{A} wird extrinsische Matrix genannt und beschreibt die Transformation zwischen dem Weltkoordinatensystem und dem Kamerakoordinatensystem. In den bisherigen Betrachtungen wurde angenommen, dass die beiden Koordinatensysteme übereinstimmen, weshalb \mathbf{R} der Identitätsmatrix und \mathbf{t} dem Nullvektor entspricht:

$$s \tilde{\mathbf{p}}^{Bild} = \underbrace{\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{A}} \tilde{\mathbf{p}}^{Welt}. \quad (2.6)$$

Im allgemeinen Fall jedoch lautet die erweiterte Projektionsgleichung [Zha98]

$$s\tilde{\mathbf{p}}^{Bild} = \mathbf{K}[\mathbf{R} \ \mathbf{t}]\tilde{\mathbf{p}}^{Welt}, \quad (2.7)$$

und man erhält schließlich den gesuchten projizierten Punkt \mathbf{p}^{Bild} durch Aufhebung der homogenen Darstellung wie bereits in Gleichung (2.3) beschrieben durch

$$s\tilde{\mathbf{p}}^{Bild} = s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} sx \\ sy \\ s \end{pmatrix} \mapsto \begin{pmatrix} sx/s \\ sy/s \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{p}^{Bild}. \quad (2.8)$$

2.2.2 Pixelbasierte Bildebene

Die Projektionsgleichung (2.7) beschreibt die Abbildungseigenschaften einer idealen Lochkamera. Für reale digitale Kameras, die statt einer ideellen Bildebene einen zweidimensionalen, pixelbasierten CCD- oder CMOS-Sensorchip besitzen (vgl. Abschnitt 2.1), muss die bisherige intrinsische Matrix erweitert werden. Im bisherigen Lochkameramodell wurde angenommen, dass das Bildkoordinatensystem im Zentrum der Bildebene im Hauptpunkt \mathbf{h} entspringt und beide Koordinatenachsen gleich skaliert sowie orthogonal zueinander sind. Im Fall pixelbasierter Sensoren ist jedoch eine Reihe von Unterschieden zu berücksichtigen [HZ03].

Das Pixelkoordinatensystem entspringt in der Regel nicht im Zentrum der Bildebene, sondern in einer der Ecken des Pixelsensors (typischerweise oben links). Die Lage des Hauptpunktes \mathbf{h} ist somit nicht mehr wie bislang angenommen bei $(0,0)$, sondern an den Pixelkoordinaten $\mathbf{h} = (u_0, v_0)$, siehe Abb. 2.3. Die Pixel müssen nicht zwangsläufig quadratisch sein. Werden die Bildkoordinaten in Pixeln gemessen, hat dies zur Folge, dass unterschiedliche Skalierungsfaktoren für die jeweiligen Richtungen vorliegen können. Wird angenommen, dass die Anzahl der Pixel pro Distanzeinheit in u - bzw. v -Richtung m_u bzw. m_v beträgt, so muss die Brennweite f in der bisherigen intrinsischen Matrix ersetzt werden durch $\alpha = fm_u$ bzw. $\beta = fm_v$. Somit repräsentieren α bzw. β die Brennweite, ausgedrückt in Pixeln in u - bzw. v -Richtung.

Weiterhin ist es möglich, dass die Achsen des Pixelkoordinatensystems nicht orthogonal zueinander sind. Dies ist z.B. der Fall, wenn der Sensor eine leichte fertigungsbedingte Schiefe aufweist. Zu diesem Zweck wird die intrinsische Matrix um den Parameter γ erweitert. Dieser Parameter ist in den meisten heute gefertigten Sensoren nicht relevant [HZ03], er wird jedoch der Vollständigkeit halber berücksichtigt.

Unter Verwendung der erweiterten intrinsischen Matrix

$$\mathbf{K} = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.9)$$

ist ein abgebildeter Punkt nicht mehr in Bildkoordinaten, sondern in Pixelkoordinaten (in homogener Darstellung) gegeben:

$$s\tilde{\mathbf{p}}^{Pix} = \mathbf{K}[\mathbf{R} \ \mathbf{t}]\tilde{\mathbf{p}}^{Welt}. \quad (2.10)$$

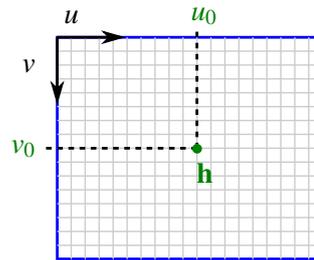


Abbildung 2.3: Das Pixelkoordinatensystem hat seinen Ursprung im Gegensatz zum Bildkoordinatensystem nicht mehr im Hauptpunkt, sondern typischerweise in der oberen linken Sensorecke. Die Lage des Hauptpunktes wird im Pixelkoordinatensystem definiert durch $\mathbf{h} = (u_0, v_0)$.

2.3 Linsenverzerrung

In den bisherigen Betrachtungen wurde das Lochkameramodell verwendet. Ein Nachteil der Lochkamera ist jedoch die geringe Lichtstärke, die zur Verfügung steht, da das Loch möglichst klein sein muss, um scharfe Abbildungen zu erlauben. Insbesondere bei der Nutzung von CMOS- und CCD-Sensorchips hätte dies einen hohen Rauschpegel zur Folge. Deshalb werden anstelle des Lochs optische Linsensysteme (Objektive) eingesetzt, die deutlich mehr einfallendes Licht zulassen. Der Nachteil beim Einsatz von Objektiven ist jedoch, dass die Abbildung nichtlinear verzerrt wird. Die beiden wichtigsten Verzerrungsarten sind die sog. radiale und die tangentielle Verzerrung.

2.3.1 Radiale Verzerrung

Die radiale Verzerrung ist die dominierende Verzerrungsart, die sich besonders häufig bei Kameras mit einem großen Blickwinkel (kleine Brennweite) beobachten lässt. Je weiter ein projizierter Punkt im Bild vom Hauptpunkt \mathbf{h} entfernt ist, desto stärker ist die Auswirkung der radialen Verzerrung. Somit ist sie am stärksten an den Bildrändern zu beobachten, wie z.B. in Abb. 2.4(e), bei der ein Büroflur dargestellt und eine deutliche Krümmung der Türrahmen erkennbar ist.

Für die Verdeutlichung der Ursache der radialen Verzerrung wird statt des einfachen Lochkameramodells kurzzeitig ein erweitertes Modell betrachtet, welches sowohl die Linse als auch die Blende der Kamera berücksichtigt, wie in Abb. 2.4(a) dargestellt und in [Luh10] detailliert beschrieben. Die Beschaffenheit eines Objektivs kann bzgl. der Positionierung der Linse und der Blende unterschiedlich sein. Im Idealfall (mittig in Abb. 2.4(a)) sollte sich die Blende (rot) möglichst an der gleichen Position befinden wie die Linse (blau). Dies ist oftmals technisch nur schwer realisierbar, so dass die Blende einen gewissen Abstand zur Linse aufweist. Je nach Bauart, kann die Blende dabei, von der Bildebene aus betrachtet, entweder vor oder hinter der Linse positioniert sein. Diese Position ist entscheidend für die Art der radialen Verzerrung. Dabei entsteht die sog. Kissenverzerrung, wenn sich die Blende vor, und die sog. Tonnenverzerrung, wenn sich die Blende hinter der Linse befindet. Der graue Kreis auf der Bildebene

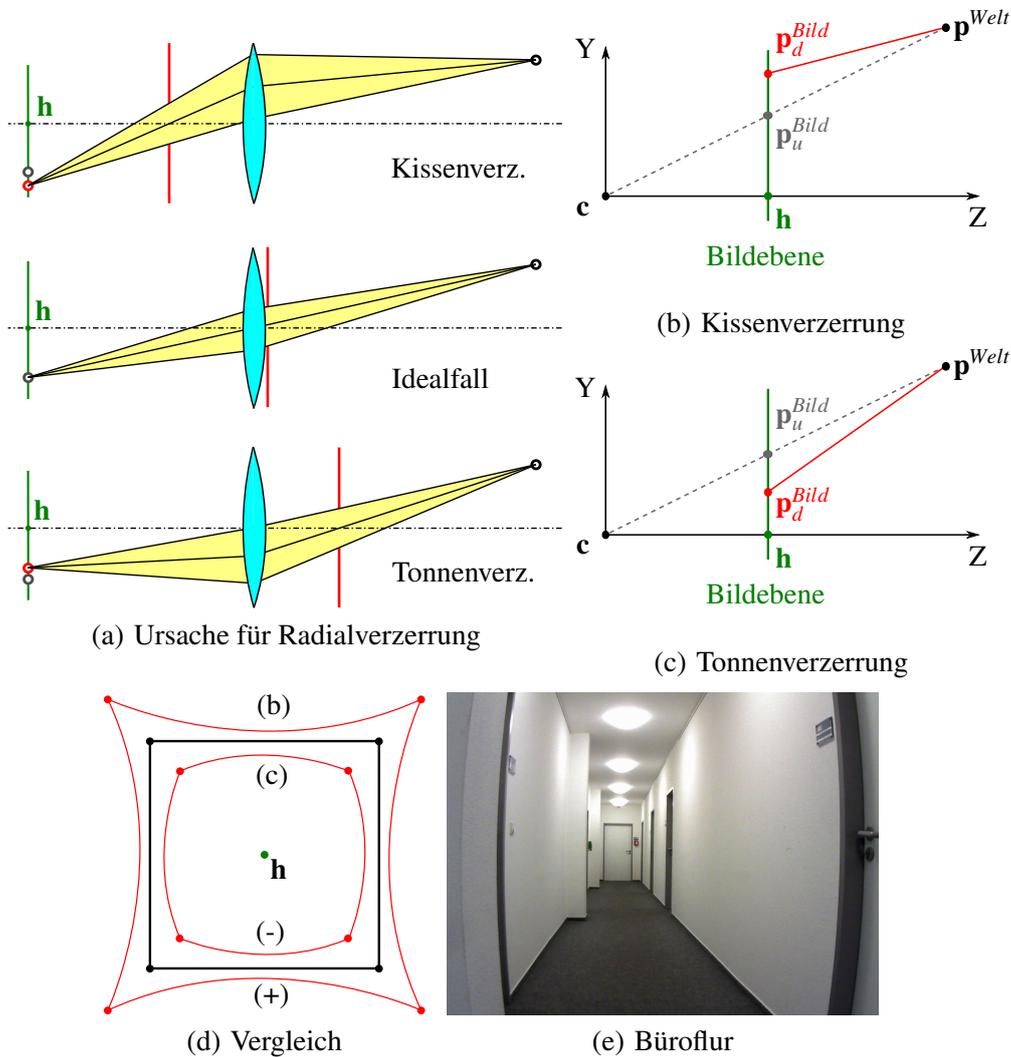


Abbildung 2.4: In (a) wird die von der relativen Position der Blende (rot) zur Linse (blau) abhängige Art der Verzerrung veranschaulicht, die eine Verschiebung des unverzerrten Punktes (grauer Kreis) auf den verzerrten Punkt (roter Kreis) in der Bildebene (grün) zur Folge hat. Bei Betrachtung des einfachen Lochkameramodells entspricht dies der Verschiebung des unverzerrten Punktes p_u^{Bild} auf den verzerrten Punkt p_d^{Bild} . Im Fall der Tonnenverzerrung (c) wird der Punkt hin zum Hauptpunkt verschoben, während er im Fall der Kissenverzerrung (b) weiter von ihm entfernt projiziert wird. In (d) wird in Anlehnung an [WCH92] die Auswirkung beider Verzerrungsarten auf ein ideal projiziertes Quadrat in der Bildebene dargestellt. Das Bild in (e) zeigt einen Büroflur, der von einer Webcam mit einer Brennweite von 6 mm aufgezeichnet wurde. Die Türrahmen wurden aufgrund der Tonnenverzerrung an den Bildrändern deutlich gekrümmt. (Bildquelle (a): Paul van Walreen, <http://toothwalker.org>)

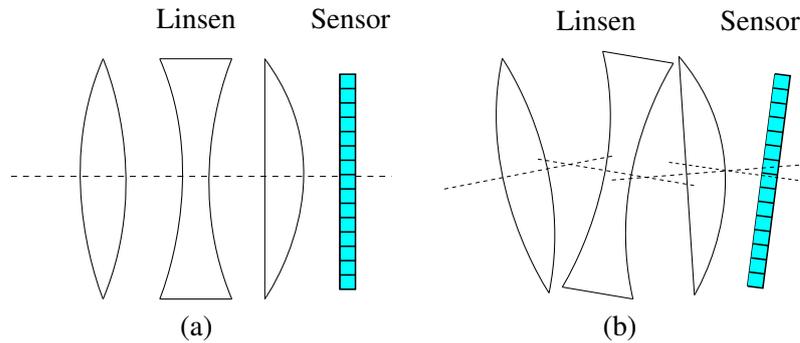


Abbildung 2.5: Kollinearität bei Linsensystemen. Die optischen Achsen der einzelnen Linsen sind durch die gestrichelten Geraden dargestellt. Im Idealfall sind diese kollinear (a). Durch Produktionsfehler ist dies jedoch nicht immer gewährleistet, so dass das in (b) überspitzt dargestellte Szenario auftreten kann.

links in Abb. 2.4(a) deutet die ideale Position eines Punktes nach der Projektion an. Die tatsächlich abgebildeten Punkte in den beiden Verzerrungsfällen sind durch die roten Kreise markiert. Die entsprechende Abweichung zwischen den Kreisen verändert sich somit mit der Position der Blende. Die beiden Skizzen in 2.4(b) und 2.4(c) verdeutlichen diese Abweichung für das vereinfachte Lochkameramodell. Der eigentlich erwartete Projektionsstrahl ist gestrichelt dargestellt, während der durch das Linsen-Blenden-System abgelenkte Strahl durchgezogen abgebildet ist. Je nach Beschaffenheit des Objektivs, wird somit ein Punkt \mathbf{p}^{Welt} nicht auf den eigentlichen unverzerrten Punkt \mathbf{p}_u^{Bild} auf der Bildebene projiziert, sondern auf den verzerrten Punkt \mathbf{p}_d^{Bild} abgelenkt. Die durch die Ablenkung verursachte Verschiebung geschieht immer radial, also entlang eines Strahls zwischen dem Hauptpunkt \mathbf{h} und dem projizierten Punkt. Der Abstand zwischen diesen beiden Punkten (der Radius) bestimmt das Ausmaß der Verzerrung. Liegt mit wachsendem Radius eine immer stärker werdende Verzerrung weg vom Hauptpunkt vor, so entspricht dies der Kissenverzerrung (positive Verzerrung, durch (+) in Abb. 2.4(d) dargestellt), während eine Verzerrung hin zum Hauptpunkt der Tonnenverzerrung oder auch der sog. "Fischaugen-Verzerrung" entspricht (negative Verzerrung, durch (-) in Abb. 2.4(d) dargestellt). Die schematische Darstellung der Auswirkung beider Verzerrungsarten auf ein ursprünglich ideal projiziertes Quadrat ist in Abb. 2.4(d) verdeutlicht.

2.3.2 Tangentiale Verzerrung

Aufgrund von Produktionstoleranzen lässt es sich in der Praxis nicht immer sicherstellen, dass die optischen Achsen der einzelnen Linsen in Kameraobjektiven sowie der Sensorchip exakt kollinear ausgerichtet sind, siehe Abb. 2.5(b). Dadurch wird ein weiterer Verzerrungsfehler verursacht, der auch Dezentrierung genannt wird [HS97] und der eine von der Radialrichtung abweichende Verschiebungskomponente besitzen kann. Die tangentielle Verzerrung ist weniger dominant als die radiale Verzerrung, jedoch kann diese insbesondere bei kostengünstigen Kameras deutlich ausgeprägt sein.

2.3.3 Mathematische Verzerrungsmodelle

Das bisherige Kameramodell ist, ohne Berücksichtigung der im vorangegangenen Abschnitt beschriebenen Linsenverzerrung, nicht in der Lage, die tatsächliche Abbildungsposition eines Punktes vorherzusagen. Deshalb muss das Kameramodell um ein Verzerrungsmodell erweitert werden, so dass die bislang unverzerrten Punkte des Kameramodells auf die verzerrten Punkte abgebildet werden können. Der Einfachheit halber soll zunächst angenommen werden, dass die folgenden Betrachtungen nicht in Pixelkoordinaten, sondern in Bildkoordinaten mit dem Hauptpunkt im Bildkoordinatenursprung stattfinden (siehe Abschnitt 2.2.1).

Sei $\mathbf{p}_u^{Bild} = \mathbf{p}_u = (x_u, y_u)$ der vom Kameramodell vorhergesagte unverzerrte Punkt in der Bildebene und $\mathbf{p}_d^{Bild} = \mathbf{p}_d = (x_d, y_d)$ der entsprechende verzerrte Punkt (zur besseren Übersicht wird auf den oberen Index Bild verzichtet). Angelehnt an die Notation in [Hei00] kann das Verzerrungsmodell beschrieben werden durch

$$\mathbf{p}_d = \mathbf{p}_u + \mathcal{F}_D(\mathbf{p}_u, \delta_D), \quad (2.11)$$

wobei die Verzerrungsfunktion $\mathcal{F}_D(\mathbf{p}_u, \delta_D)$ vom unverzerrten Punkt \mathbf{p}_u und dem Parametervektor δ_D abhängt und den Verschiebungsvektor $(\Delta_x, \Delta_y)^T$ von \mathbf{p}_u nach \mathbf{p}_d liefert:

$$\mathcal{F}_D(\mathbf{p}_u, \delta_D) = \begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix}. \quad (2.12)$$

Die radiale Verzerrungskomponente, die mit zunehmendem Abstand zum Hauptpunkt $\mathbf{h} = (0, 0)$ eine stärkere Ausprägung besitzt, kann wie folgt beschrieben werden [HS97, Sla80]:

$$\mathcal{F}_D^{Rad}(\mathbf{p}_u, \delta_D^{Rad}) = \begin{pmatrix} x_u(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 + \dots) \\ y_u(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 + \dots) \end{pmatrix} = \begin{pmatrix} \Delta_x^{Rad} \\ \Delta_y^{Rad} \end{pmatrix}. \quad (2.13)$$

Dabei entspricht $r = \sqrt{x_u^2 + y_u^2}$ dem Radius und $\kappa_1, \kappa_2, \kappa_3, \dots$ den radialen Koeffizienten, die im Parametervektor $\delta_D^{Rad} = (\kappa_1, \kappa_2, \kappa_3, \dots)$ zusammengefasst sind. In der Praxis ist es üblich, sich auf die ersten drei Terme zu beschränken [Bro71], wobei laut [BK08] der dritte von κ_3 abhängige Term insbesondere für stark verzerrende Kameras (Fischaugen-Kameras) genutzt werden kann.

Die tangentielle Verzerrungskomponente ist ebenfalls vom Radius r abhängig und lässt sich laut [Bro66] durch die Verzerrungsfunktion

$$\mathcal{F}_D^{Tan}(\mathbf{p}_u, \delta_D^{Tan}) = \begin{pmatrix} [2\rho_1 x_u y_u + \rho_2(r^2 + 2x_u^2)] [1 + \rho_3 r^2 + \rho_4 r^4 + \dots] \\ [2\rho_2 x_u y_u + \rho_1(r^2 + 2y_u^2)] [1 + \rho_3 r^2 + \rho_4 r^4 + \dots] \end{pmatrix} = \begin{pmatrix} \Delta_x^{Tan} \\ \Delta_y^{Tan} \end{pmatrix} \quad (2.14)$$

mit den Koeffizienten $\rho_1, \rho_2, \rho_3, \rho_4, \dots$ beschreiben, aus denen sich der Parametervektor $\delta_D^{Tan} = (\rho_1, \rho_2, \rho_3, \rho_4, \dots)$ zusammensetzt. Nach Aussage von [FB86] sind die Koeffizienten ρ_3, ρ_4, \dots selten signifikant, wodurch die Verzerrungsfunktion in der reduzierten Variante nur von den Koeffizienten ρ_1 und ρ_2 abhängt und, wie z.B. in [HS97] verwendet, von folgender Form ist:

$$\mathcal{F}_D^{Tan}(\mathbf{p}_u, \delta_D^{Tan}) = \begin{pmatrix} 2\rho_1 x_u y_u + \rho_2(r^2 + 2x_u^2) \\ 2\rho_2 x_u y_u + \rho_1(r^2 + 2y_u^2) \end{pmatrix} = \begin{pmatrix} \Delta_x^{Tan} \\ \Delta_y^{Tan} \end{pmatrix}. \quad (2.15)$$

Die Verzerrungsfunktion aus Gleichung (2.12) kann jetzt aus den beiden Verzerrungskomponenten wie folgt zusammengesetzt werden [MMB04]:

$$\mathcal{F}_D(\mathbf{p}_u, [\delta_D^{Rad}, \delta_D^{Tan}]) = \mathcal{F}_D^{Rad}(\mathbf{p}_u, \delta_D^{Rad}) + \mathcal{F}_D^{Tan}(\mathbf{p}_u, \delta_D^{Tan}) = \begin{pmatrix} \Delta_x^{Rad} + \Delta_x^{Tan} \\ \Delta_y^{Rad} + \Delta_y^{Tan} \end{pmatrix} \quad (2.16)$$

mit dem zusammengesetzten Parametervektor $\delta_D = [\delta_D^{Rad}, \delta_D^{Tan}]$.

In den bisherigen Betrachtungen wurde die Verzerrung in Bildkoordinaten durchgeführt. Wie bereits in Abschnitt 2.2.2 erwähnt, kann in der Praxis angenommen werden, dass $\gamma = 0$ in der intrinsischen Matrix (siehe Gleichung (2.9)) gilt. Somit ist die Beziehung zwischen dem verzerrten Punkt $\mathbf{p}_d^{Pix} = (u_d, v_d)^T$ in Pixelkoordinaten und dem unverzerrten Punkt $\mathbf{p}_u = (x_u, y_u)$ in Bildkoordinaten gegeben durch

$$\mathbf{p}_d^{Pix} = \begin{pmatrix} u_d \\ v_d \end{pmatrix} = \begin{pmatrix} m_u x_d \\ m_v y_d \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} m_u(x_u + \Delta_x^{Rad} + \Delta_x^{Tan}) \\ m_v(y_u + \Delta_y^{Rad} + \Delta_y^{Tan}) \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \quad (2.17)$$

mit dem Hauptpunkt $\mathbf{h} = (u_0, v_0)$ im Pixelkoordinatensystem und den Skalierungsfaktoren m_u und m_v aus Abschnitt 2.2.2, welche die Anzahl der Pixel pro Distanzeinheit in u - bzw. v -Richtung angeben. Diese können aus dem Datenblatt der Kamera entnommen werden. Liegt ein solches Datenblatt nicht vor und spielt sowohl die tatsächliche Anzahl der Pixel pro Distanzeinheit als auch die tatsächliche Brennweite einer Kamera keine Rolle, so kann diese als $f = 1$ angenommen werden, da andere Werte von f lediglich eine Skalierung der Bildebene bedeuten [MT96]. Durch den Zusammenhang der Parameter $\alpha = fm_u$ und $\beta = fm_v$, wie in Abschnitt 2.2.2 eingeführt, kann Gleichung (2.17) dahingehend abgeändert werden, dass nun die direkt aus der intrinsischen Matrix \mathbf{K} bekannten Parameter α und β verwendet werden können:

$$\mathbf{p}_d^{Pix} = \begin{pmatrix} u_d \\ v_d \end{pmatrix} = \begin{pmatrix} \alpha x_d \\ \beta y_d \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} \alpha(x_u + \Delta_x^{Rad} + \Delta_x^{Tan}) \\ \beta(y_u + \Delta_y^{Rad} + \Delta_y^{Tan}) \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix}. \quad (2.18)$$

Durch sequenzielle Anwendung der in Abschnitt 2.2.1 beschriebenen Projektion und der anschließenden Verzerrung, wie durch Formel (2.18) beschrieben, ist es möglich, die Pixelposition eines projizierten Punktes auf dem Sensor einer realen Kamera mit einem Objektiv (Linsensystem) vorherzusagen. Dieser Vorgang sei zusammengefasst durch die Projektionsfunktion

$$\mathbf{p}_d^{Pix} = \mathcal{P}(\mathbf{p}^{Welt}, \mathbf{R}, \mathbf{t}, \alpha, \beta, \gamma, u_0, v_0, \kappa_1, \kappa_2, \kappa_3, \rho_1, \rho_2), \quad (2.19)$$

die in Abhängigkeit der Parameter einen Punkt \mathbf{p}^{Welt} im Weltkoordinatensystem auf den Punkt \mathbf{p}_d^{Pix} unter Berücksichtigung des beschriebenen Kamera- und Verzerrungsmodells abbildet. Es gelte zusätzlich die folgende Kurzschreibweise, wobei die einzelnen Parameter der intrinsischen Matrix \mathbf{K} sowie dem Verzerrungs-Parametervektor δ_D zu entnehmen sind:

$$\mathbf{p}_d^{Pix} = \mathcal{P}(\mathbf{p}^{Welt}, \mathbf{R}, \mathbf{t}, \mathbf{K}, \delta_D). \quad (2.20)$$

Neben dem beschriebenen Verfahren existieren weitere Alternativen, jedoch stellt dieses Verfahren das populärste dar. Beispielsweise nutzen die bekannte freie Bibliothek

OpenCV [Bra00] sowie die "Camera calibration toolbox for Matlab" [Bou10] das beschriebene Vorgehen bei ihrer Implementierung, um die Projektion von Punkten inklusive Verzerrung zu berechnen. Dieses beschriebene Verfahren wird auch bei der Kamerakalibrierung benötigt, auf die im folgenden Abschnitt 2.4 eingegangen wird.

2.4 Kamerakalibrierung

Die in Gleichung (2.20) eingeführte Projektionsfunktion, die einen 3D-Punkt, entsprechend des beschriebenen Kamera- sowie Verzerrungsmodells abbildet, ist abhängig von mehreren Modellparametern.

Ziel der Kalibrierung ist, diese Modellparameter derart zu bestimmen, dass der Abstand $d = \|\hat{\mathbf{p}} - \mathcal{P}(\mathbf{p}^{Welt}, \mathbf{R}, \mathbf{t}, \mathbf{K}, \delta_D)\|$ zwischen dem vom Modell vorhergesagten (mit der Projektionsfunktion berechneten) Punkt und dem tatsächlichen im Bild beobachtbaren Punkt $\hat{\mathbf{p}}$ minimal ist (siehe Abb. 2.6(a)). Dieser Abstand d wird auch als Reprojektionsfehler bezeichnet. Um mehrere solcher Punktpaare zu generieren, kann ein Objekt mit bekannter Geometrie mit der Kamera aus verschiedenen Ansichten aufgezeichnet werden. Diese Objekte, die im Folgenden Kalibrierobjekte genannt werden, besitzen oftmals Merkmale, die in den Bildern einfach zu identifizieren sind, wie z.B. die Eckpunkte eines Schachbrettmusters (siehe Abb. 2.6(b), vgl. Abschnitt 5.3.1.1).

Ein typischer Ansatz bei der Kalibrierung ist die Aufstellung einer Kostenfunktion, die von den gesuchten Parametern abhängt und den Reprojektionsfehler berücksichtigt. Die gesuchten optimalen Modellparameter sind dann genau diejenigen, die die Kostenfunktion minimieren. Seien \mathbf{p}_j^{Welt} , $j = 0, \dots, m-1$, Punkte im Weltkoordinatensystem, die zum Kalibrierobjekt (siehe Abb. 2.6(b)) gehören, und seien \mathbf{R}_i und \mathbf{t}_i die Rotationsmatrix und der Translationsvektor, die die Lage der Kamera relativ zu dem Kalibrierobjekt in der i -ten Ansicht, $i = 0, \dots, n-1$, beschreiben (siehe Abb. 2.6(c)). Dann liefert die Projektion jedes Punktes \mathbf{p}_j^{Welt} für jede Ansicht i den Punkt $\mathbf{p}_{i,j}$:

$$\mathbf{p}_{i,j} = \mathcal{P}(\mathbf{p}_j^{Welt}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{K}, \delta_D). \quad (2.21)$$

Seien nun $\hat{\mathbf{p}}_{i,j}$ die in den Kamerabildern beobachtbaren 2D-Punkte (Merkmalspositionen) für die jeweiligen Kameraansichten, die zu den 3D-Punkten \mathbf{p}_j^{Welt} gehören. Dann lässt sich die Kostenfunktion beschreiben durch

$$\mathcal{H}(\mathbf{R}_i, \mathbf{t}_i, \mathbf{K}, \delta_D) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \|\hat{\mathbf{p}}_{i,j} - \mathcal{P}(\mathbf{p}_j^{Welt}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{K}, \delta_D)\|^2. \quad (2.22)$$

Die Suche nach den optimalen Parametern entspricht einem nichtlinearen Optimierungsproblem, welches z.B. mit dem Levenberg-Marquardt-Algorithmus [Mor77] berechnet werden kann.

Kritisch bei dieser Vorgehensweise ist jedoch die Wahl der initialen Parameterwerte. Bezüglich der Verzerrungsparameter könnte initial angenommen werden, dass die Linse zunächst ideal ist und keinerlei Verzerrung verursacht:

$$\kappa_1 = \kappa_2 = \kappa_3 = \rho_1 = \rho_2 = 0. \quad (2.23)$$

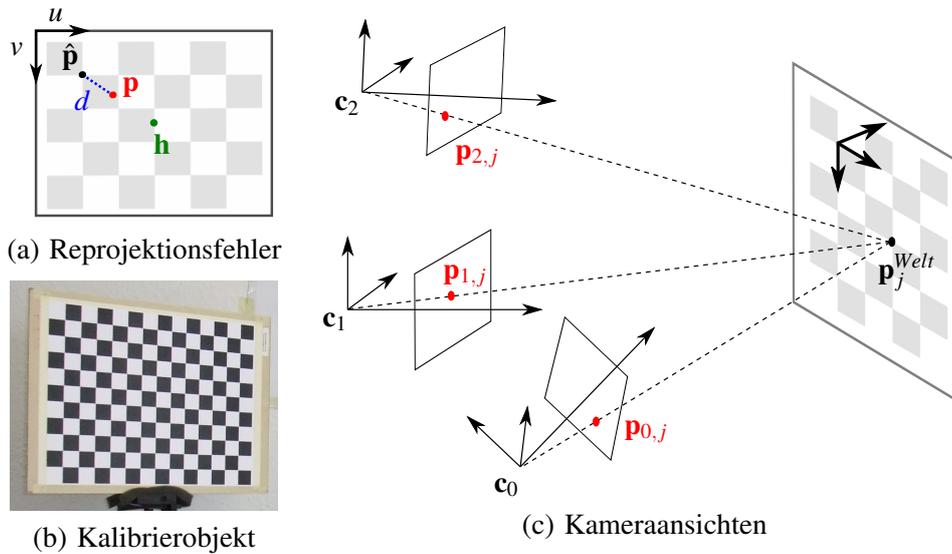


Abbildung 2.6: In (a) wird der Reprojektionsfehler d verdeutlicht, der dem Abstand zwischen dem mit Hilfe des Kamera- und Verzerrungsmodells berechneten Punkt \mathbf{p} und dem tatsächlich beobachtbaren Punkt $\hat{\mathbf{p}}$ entspricht (a). In (c) wird in verschiedenen Kameraansichten (gleiche Kamera, unterschiedliche Positionen) ein Punkt \mathbf{p}_j^{Welt} auf dem Kalibrierobjekt (vgl. (b)), der in Weltkoordinaten bekannt ist, auf die jeweiligen Bildpunkte $\mathbf{p}_{i,j}$ projiziert.

Der Hauptpunkt liegt optimalerweise in der Mitte des Pixelsensors, weshalb

$$\mathbf{h} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} \frac{w_{sensor}}{2} \\ \frac{h_{sensor}}{2} \end{pmatrix} \quad (2.24)$$

ein sinnvoller initialer Wert ist, wobei w_{sensor} und h_{sensor} der Anzahl der Pixel des Sensors in der Breite und der Höhe entsprechen.

Leider ist es nicht möglich, vergleichbar einfache Annahmen für die Initialisierung der Lageparameter \mathbf{R}_i und \mathbf{t}_i sowie den weiteren Parametern der intrinsischen Matrix \mathbf{K} zu machen. Wie diese dennoch sinnvoll initialisiert werden können, wird in den Abschnitten 2.4.1 und 2.4.2 behandelt.

2.4.1 Intrinsische Kalibrierung

Davon ausgehend, dass ein planares Kalibrierobjekt wie in Abb. 2.6(b) verwendet wird, können mit dem in [Zha98] beschriebenen Verfahren die intrinsischen Parameter, wie nachfolgend beschrieben, geschätzt werden. Dabei wird bei der Schätzung die Linsenverzerrung ignoriert. Die Grundidee ist die Annahme, dass sich die Abbildung von einer planaren Ebene (Kalibrierobjekt) auf eine andere Ebene (Sensorchip) mit einer Homographie beschreiben lässt. Dabei wird definiert, dass das Koordinatensystem des planaren Kalibrierobjekts im Weltkoordinatenursprung entspringt und in der von der X- und Y-Achse aufgespannten Ebene liegt (vgl. Abschnitt 5.3.1.1). Somit gilt für alle 3D-Punkte, die in der Kalibrierobjektebene liegen, dass die Z-Koordinate gleich

Null ist: $\mathbf{p}^{Welt} = (X, Y, 0)^T$. Für $\tilde{\mathbf{p}}^{Welt} = (X, Y, 0, 1)^T$ gilt somit unter Anwendung der Gleichung (2.10)

$$s\tilde{\mathbf{p}}^{Pix} = \mathbf{K}(\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3 \ \mathbf{t}) \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}, \quad (2.25)$$

wobei \mathbf{K} die intrinsische Matrix, \mathbf{t} den Translationsvektor und $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ die Spalten der Rotationsmatrix \mathbf{R} bezeichnen. Da die Einträge des Spaltenvektors \mathbf{r}_3 immer mit Null multipliziert werden und somit keinen Beitrag liefern, kann die Gleichung vereinfacht werden zu

$$s\tilde{\mathbf{p}}^{Pix} = \mathbf{K}(\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}) \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t})\tilde{\mathbf{p}}^{Obj}. \quad (2.26)$$

Dabei bezeichnet $\tilde{\mathbf{p}}^{Obj}$ die homogene Darstellung des zweidimensionalen Punktes $\mathbf{p}^{Obj} = (X, Y)^T$ im Koordinatensystem des planaren Kalibrierobjekts. Somit kann die Beziehung zwischen Punkten auf dem Kalibrierobjekt und dem Abbild auf dem Kamerasensor mittels einer Homographie \mathbf{H} beschrieben werden durch

$$s\tilde{\mathbf{p}}^{Pix} = \mathbf{H}\tilde{\mathbf{p}}^{Obj} \quad (2.27)$$

mit

$$\mathbf{H} = \mathbf{K}(\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}). \quad (2.28)$$

Für die Bestimmung der 3×3 Matrix \mathbf{H} , die bis auf einen beliebigen Skalierungsfaktor eindeutig ist und somit acht Freiheitsgrade besitzt, müssen mindestens vier Punktpaare vorliegen (siehe Abb. 2.7). Ein Punktpaar besteht aus den Koordinaten markanter Punkte im Kalibrierobjekt (z.B. Ecken des Schachbrettmusters), die aufgrund der bekannten Geometrie des Kalibrierobjekts zur Verfügung stehen, sowie den Pixelkoordinaten im Kamerabild, an denen die markanten Punkte wiedergefunden werden können. Details zur Berechnung der Homographie sind z.B. in [HZ03] zu finden.

Sei nun $\mathbf{H} = (\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3)$ die (bis auf einen Skalierungsfaktor) berechnete Homographie mit den Spaltenvektoren $\mathbf{h}_1, \mathbf{h}_2$ und \mathbf{h}_3 . Dann gilt mit Gleichung (2.28)

$$(\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3) = \lambda \mathbf{K}(\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}), \quad (2.29)$$

wobei $\lambda \neq 0$ einen Skalierungsfaktor bezeichnet.

Aus Gleichung (2.29) lässt sich folgender Zusammenhang zwischen den Spalten der Homographie und den Spalten der Rotationsmatrix ableiten, der in den Gleichungen (2.32) und (2.33) genutzt werden kann:

$$\mathbf{h}_1 = \lambda \mathbf{K}\mathbf{r}_1 \Leftrightarrow \lambda \mathbf{r}_1 = \mathbf{K}^{-1}\mathbf{h}_1, \quad (2.30)$$

$$\mathbf{h}_2 = \lambda \mathbf{K}\mathbf{r}_2 \Leftrightarrow \lambda \mathbf{r}_2 = \mathbf{K}^{-1}\mathbf{h}_2. \quad (2.31)$$

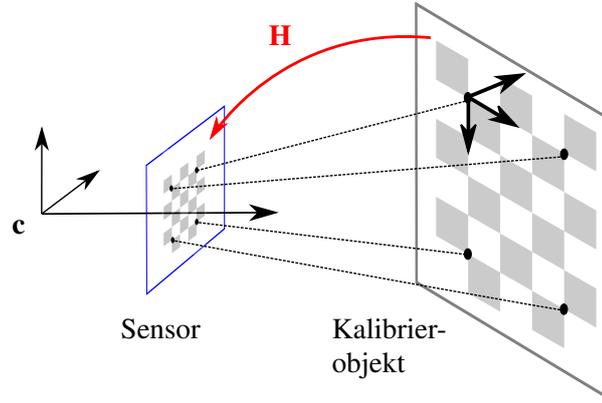


Abbildung 2.7: Die Abbildung von Punkten des Kalibrierobjekts auf Punkte des Kamerasensors kann mathematisch durch eine Homographie \mathbf{H} beschrieben werden.

Da die Spalten der Rotationsmatrix orthogonal zueinander sind, gilt

$$\begin{aligned}
 \mathbf{r}_1^T \mathbf{r}_2 &= 0 \\
 \Leftrightarrow (\lambda \mathbf{r}_1)^T (\lambda \mathbf{r}_2) &= 0 \\
 \Leftrightarrow (\mathbf{K}^{-1} \mathbf{h}_1)^T (\mathbf{K}^{-1} \mathbf{h}_2) &= 0 \\
 \Leftrightarrow \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 &= 0 \\
 \Leftrightarrow \mathbf{h}_1^T \mathbf{B} \mathbf{h}_2 &= 0
 \end{aligned} \tag{2.32}$$

mit der Matrix $\mathbf{B} = \mathbf{K}^{-T} \mathbf{K}^{-1}$ und $\mathbf{K}^{-T} = (\mathbf{K}^{-1})^T = (\mathbf{K}^T)^{-1}$. Somit liefert die Gleichung (2.32) die erste Nebenbedingung für die Bestimmung der intrinsischen Parameter. Die zweite Nebenbedingung ergibt sich durch die Nutzung der Eigenschaft, dass die Spalten einer Rotationsmatrix normiert sind auf die Länge 1, wodurch gilt:

$$\begin{aligned}
 \mathbf{r}_1^T \mathbf{r}_1 &= \mathbf{r}_2^T \mathbf{r}_2 \\
 \Leftrightarrow (\lambda \mathbf{r}_1)^T (\lambda \mathbf{r}_1) &= (\lambda \mathbf{r}_2)^T (\lambda \mathbf{r}_2) \\
 \Leftrightarrow (\mathbf{K}^{-1} \mathbf{h}_1)^T (\mathbf{K}^{-1} \mathbf{h}_1) &= (\mathbf{K}^{-1} \mathbf{h}_2)^T (\mathbf{K}^{-1} \mathbf{h}_2) \\
 \Leftrightarrow \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_1 &= \mathbf{h}_2^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 \\
 \Leftrightarrow \mathbf{h}_1^T \mathbf{B} \mathbf{h}_1 &= \mathbf{h}_2^T \mathbf{B} \mathbf{h}_2 \\
 \Leftrightarrow \mathbf{h}_1^T \mathbf{B} \mathbf{h}_1 - \mathbf{h}_2^T \mathbf{B} \mathbf{h}_2 &= 0.
 \end{aligned} \tag{2.33}$$

Für die symmetrische Matrix \mathbf{B} gilt

$$\begin{aligned}
 \mathbf{B} = \mathbf{K}^{-T} \mathbf{K}^{-1} &= \begin{pmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{pmatrix} \\
 &= \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ b_{13} & b_{23} & b_{33} \end{pmatrix}.
 \end{aligned} \tag{2.34}$$

Ziel ist es im Folgenden, ein Gleichungssystem, basierend auf der Matrix \mathbf{B} , aufzustellen, das die beiden Nebenbedingungen aus den Gleichungen (2.32) und (2.33) nutzt. Durch die Lösung des Gleichungssystems erhält man die Einträge der Matrix \mathbf{B} , mit deren Hilfe schließlich die eigentlichen intrinsischen Parameter der Matrix \mathbf{K} berechnet werden können.

Aufgrund der Symmetrie kann \mathbf{B} kompakt dargestellt werden durch den Vektor

$$\mathbf{b} = (b_{11}, b_{12}, b_{22}, b_{13}, b_{23}, b_{33})^T. \quad (2.35)$$

Für den i -ten Spaltenvektor der Homographie \mathbf{H} gelte $\mathbf{h}_i = (h_{1i}, h_{2i}, h_{3i})^T$. Um die beiden Nebenbedingungen, die von der Form $\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j$ sind, kompakter beschreiben zu können, wird der Vektor

$$\mathbf{v}_{ij} = (h_{1i}h_{1j}, h_{1i}h_{2j} + h_{2i}h_{1j}, h_{2i}h_{2j}, h_{3i}h_{1j} + h_{1i}h_{3j}, h_{3i}h_{2j} + h_{2i}h_{3j}, h_{3i}h_{3j})^T \quad (2.36)$$

definiert und es gilt

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b}. \quad (2.37)$$

Mit den beiden Nebenbedingungen aus den Gleichungen (2.32) und (2.33) und der kompakten Schreibweise kann das Gleichungssystem

$$\begin{pmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{11})^T \end{pmatrix} \mathbf{b} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2.38)$$

aufgestellt werden. Für eine gegebene Homographie und den zwei Nebenbedingungen erhält man somit zwei Gleichungen. Dies reicht nicht aus, um die sechs Unbekannten des Vektors \mathbf{b} (bzw. der Matrix \mathbf{B}) zu bestimmen. Stehen jedoch mehrere Ansichten des Kalibrierobjekts zur Verfügung (vgl. Abb. 2.6(c)), wobei jede ihre eigene Homographie liefert, so kommen pro Ansicht zwei weitere Gleichungen in der Art, wie in Gleichung (2.38) beschrieben, hinzu. Durch die Stapelung von n solcher Gleichungen für n Ansichten, erhält man das Gleichungssystem

$$\mathbf{V} \mathbf{b} = \mathbf{0}, \quad (2.39)$$

wobei \mathbf{V} eine $2n \times 6$ Matrix darstellt. Somit kann \mathbf{B} (bis auf einen Skalierungsfaktor) bestimmt werden, falls mindestens $n \geq 3$ Ansichten ($n \geq 3$ Homographien) zur Verfügung stehen. Der Lösungsvektor \mathbf{b} für das Gleichungssystem in (2.39) ist gegeben durch den Eigenvektor zum kleinsten Eigenwert der Singulärwertzerlegung (SVD) der Matrix \mathbf{V} . Details zur Bestimmung des Lösungsvektors sind in [HZ03] beschrieben.

Mit dem beschriebenen Verfahren lässt sich die Matrix \mathbf{B} bis auf einen Skalierungsfaktor bestimmen, so dass $\mathbf{B} = \lambda \mathbf{K}^{-T} \mathbf{K}^{-1}$ mit $\lambda \neq 0$ gilt. Die intrinsischen Parameter sowie λ können schließlich aus den Matrixeinträgen von \mathbf{B} extrahiert werden [Zha98]:

$$v_0 = (b_{12}b_{13} - b_{11}b_{23}) / (b_{11}b_{22} - b_{12}^2), \quad (2.40)$$

$$\lambda = b_{33} - (b_{13}^2 + v_0(b_{12}b_{13} - b_{11}b_{23})) / b_{11}, \quad (2.41)$$

$$\alpha = \sqrt{\lambda / b_{11}}, \quad (2.42)$$

$$\beta = \sqrt{\lambda b_{11} / (b_{11}b_{22} - b_{12}^2)}, \quad (2.43)$$

$$\gamma = -b_{12} \alpha^2 \beta / \lambda, \quad (2.44)$$

$$u_0 = \gamma v_0 / \beta - b_{13} \alpha^2 / \lambda. \quad (2.45)$$

2.4.2 Extrinsische Kalibrierung

Die extrinsische Kalibrierung hat zum Ziel, die Rotationsmatrix \mathbf{R} und den Translationsvektor \mathbf{t} zu bestimmen, welche die Transformation zwischen dem Kamera- und dem Weltkoordinatensystem beschreiben. Davon ausgehend, dass das planare Kalibrierobjekt im Koordinatenursprung liegt (vgl. Abschnitt 2.4.1), wird zunächst die Homographie $\mathbf{H} = (\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3)$ bestimmt, die die Merkmalspunkte des Kalibrierobjekts auf die Punkte im Kamerabild abbildet (vgl. Abb. 2.7). Zusammen mit den bereits geschätzten intrinsischen Parametern der Matrix \mathbf{K} (vgl. Abschnitt 2.4.1) können die extrinsischen Parameter wie folgt geschätzt werden [Zha98]:

$$\mathbf{r}_1 = \mu \mathbf{K}^{-1} \mathbf{h}_1, \quad (2.46)$$

$$\mathbf{r}_2 = \mu \mathbf{K}^{-1} \mathbf{h}_2, \quad (2.47)$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2, \quad (2.48)$$

$$\mathbf{t} = \mu \mathbf{K}^{-1} \mathbf{h}_3. \quad (2.49)$$

Dabei sind $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ die Spalten der Rotationsmatrix \mathbf{R} und μ ein Skalierungsfaktor mit

$$\mu = 1 / \|\mathbf{K}^{-1} \mathbf{h}_1\| = 1 / \|\mathbf{K}^{-1} \mathbf{h}_2\|. \quad (2.50)$$

Aufgrund von Rauschen kann die rechte Gleichheitsbedingung möglicherweise nicht eingehalten werden, weshalb in der Praxis getrennt betrachtete Skalierungsfaktoren μ_1, μ_2, μ_t eingesetzt werden [Bra00] und die Gleichungen (2.46) bis (2.49) dann wie folgt lauten:

$$\mathbf{r}_1 = \mu_1 \mathbf{K}^{-1} \mathbf{h}_1 \quad \text{mit } \mu_1 = 1 / \|\mathbf{K}^{-1} \mathbf{h}_1\|, \quad (2.51)$$

$$\mathbf{r}_2 = \mu_2 \mathbf{K}^{-1} \mathbf{h}_2 \quad \text{mit } \mu_2 = 1 / \|\mathbf{K}^{-1} \mathbf{h}_2\|, \quad (2.52)$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2, \quad (2.53)$$

$$\mathbf{t} = \mu_t \mathbf{K}^{-1} \mathbf{h}_3 \quad \text{mit } \mu_t = 2 / (\|\mathbf{K}^{-1} \mathbf{h}_1\| + \|\mathbf{K}^{-1} \mathbf{h}_2\|). \quad (2.54)$$

Die Spalten einer Rotationsmatrix müssen orthonormal sein. Durch die Skalierung mit μ (bzw. μ_1, μ_2) ist die Normierung sichergestellt. Die Orthogonalitätsbedingung könnte jedoch aufgrund verrauschter Daten verletzt sein. Sei $\mathbf{R} = \mathbf{USV}^T$ die Singulärwertzerlegung [HZ03, DBH02] der Rotationsmatrix \mathbf{R} . Dann gilt, dass die Spalten der Matrizen \mathbf{U} und \mathbf{V} orthonormal sind. Statt der ursprünglichen Matrix \mathbf{R} kann die Matrix

$$\mathbf{R}' = \mathbf{UV}^T \quad (2.55)$$

verwendet werden, die die Orthonormalitätsbedingung erfüllt und der nachgewiesen werden kann, dass sie der ursprünglichen Matrix \mathbf{R} bezüglich der Frobeniusnorm am ähnlichsten ist [Zha98].

Sollen die extrinsischen Parameter für mehrere Ansichten geschätzt werden, wie es z.B. für die Kostenfunktion in Gleichung (2.22) notwendig ist, wird zunächst für jede Ansicht i die zugehörige Homographie \mathbf{H}_i bestimmt und anschließend \mathbf{R}'_i und \mathbf{t}_i , wie beschrieben, geschätzt.

2.4.3 Globale Kalibrierung

In den Abschnitten 2.4.1 und 2.4.2 wurden die intrinsischen und extrinsischen Parameter geschätzt. Dabei wurde jedoch angenommen, dass keine Linsenverzerrung vorliegt, was in der Praxis selten zutrifft. Aus diesem Grund ist die Schätzung der Parameter ungenau, weshalb diese weiter zu optimieren sind. Da nun sämtliche Parameter der in Gleichung (2.22) definierten Kostenfunktion $\mathcal{H}(\mathbf{R}_i, \mathbf{t}_i, \mathbf{K}, \delta_D)$, wie am Anfang des Abschnitts 2.4 erläutert, initialisiert sind, kann diese minimiert werden. Anschließend stehen alle Modellparameter der Kamera zur Verfügung und ihre Abbildungseigenschaften können somit als bekannt angenommen werden.

Kritisch ist dabei zu sehen, dass die so bestimmten Modellparameter zwar lokal, jedoch nicht zwangsläufig global optimal sein müssen. Ein globales Optimum wird nur erreicht, wenn die initialisierten Parameter von entsprechender Güte sind, also bereits in der Nähe des globalen Optimums liegen. Da jedoch eine starke Linsenverzerrung zu einer starken Verfälschung der initialen Schätzung der intrinsischen und extrinsischen Parameter führen kann, ist dies unter Umständen nicht gewährleistet.

Deshalb wird im Folgenden eine alternative Herangehensweise beschrieben, bei der die Linsenverzerrung unabhängig vom restlichen Kameramodell betrachtet werden kann.

2.5 Forward- und Backward-Modelle

Mit dem in den vorangegangenen Abschnitten erläuterten Kameramodell ist es möglich, die Projektion von 3D-Punkten in der Welt auf 2D-Punkte in der Bildebene zu beschreiben. Modelle, die diese vorwärts orientierte Projektion beschreiben, werden in [Hei00] mit "Forward Camera Model" bezeichnet.

Mit "Backward Camera Model" werden hingegen solche Kameramodelle bezeichnet, die diese Projektion rückwärts durchführen, also zu einem gegebenen 2D-Punkt in der Bildebene den dazugehörigen Strahl in der Welt beschreiben, auf dem der entsprechende 3D-Punkt liegen muss. Eine Darstellung beider Modelle als Pipeline ist in Abb. 2.8 gegeben.

2.5.1 Forward- und Backward-Verzerrungsmodelle

Analog zu dieser Bezeichnung soll mit dem Forward-Verzerrungsmodell das Linsenverzerrungsmodell bezeichnet werden, welches die Abbildung von unverzerrten Bildpunkten \mathbf{p}_u auf verzerrte Bildpunkte \mathbf{p}_d beschreibt. Ein solches Forward-Verzerrungsmodell wurde bereits in Gleichung (2.11) beschrieben durch

$$\mathbf{p}_d = \mathbf{p}_u + \mathcal{F}_D(\mathbf{p}_u, \delta_D). \quad (2.56)$$

Das mit Backward bezeichnete Verzerrungsmodell soll hingegen verzerrte Bildpunkte \mathbf{p}_d zurück auf die unverzerrten Punkte \mathbf{p}_u abbilden, was einer Entzerrung entspricht.

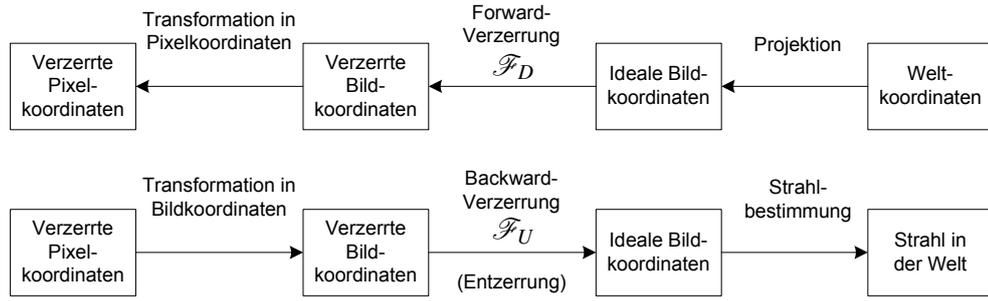


Abbildung 2.8: Pipeline-Darstellung des Forward-Kameramodells (oben) und des Backward-Kameramodells (unten).

Deshalb werden nachfolgend die Begriffe Backward-Verzerrungsmodell und Entzerrungsmodell synonym verwendet. Ein solches Entzerrungsmodell kann beschrieben werden durch

$$\mathbf{p}_u = \mathbf{p}_d + \mathcal{F}_U(\mathbf{p}_d, \delta_U) \quad (2.57)$$

mit der Entzerrungsfunktion \mathcal{F}_U und dem Entzerrungsparametervektor δ_U .

Die Grundstruktur des Forward- und des Backward-Verzerrungsmodells ist vergleichbar, jedoch müssen in den Formeln (2.13) und (2.15) für das Backward-Verzerrungsmodell x_u und y_u ersetzt werden durch x_d und y_d , da nun \mathbf{p}_d und nicht \mathbf{p}_u Ausgangspunkt ist, so dass z.B. für den Radius $r = \sqrt{x_d^2 + y_d^2}$ gilt.

Angenommen, es liegt eine Menge von Punktpaaren $\mathbf{p}_{u,i} = (x_{u,i}, y_{u,i})^T$ (unverzerrt) und $\mathbf{p}_{d,i} = (x_{d,i}, y_{d,i})^T$ (verzerrt) vor. Dann ist es im Idealfall möglich, beide Verzerrungsmodelle mit ihren jeweiligen Verzerrungsparametern δ_D und δ_U unabhängig voneinander hinsichtlich derselben Linse so auf den Punktpaaren zu optimieren, dass für einen Punkt \mathbf{p}_d gilt:

$$\mathbf{p}_d = \underbrace{[\mathbf{p}_d + \mathcal{F}_U(\mathbf{p}_d, \delta_U)]}_{\mathbf{p}_u} + \mathcal{F}_D(\underbrace{[\mathbf{p}_d + \mathcal{F}_U(\mathbf{p}_d, \delta_U)]}_{\mathbf{p}_u}, \delta_D). \quad (2.58)$$

Dabei muss jedoch beachtet werden, dass sogar im Fall derselben betrachteten Linse δ_D und δ_U in der Regel unterschiedliche Parameter enthalten. Um die Verzerrung in beiden Richtungen durchführen zu können (also eine Verzerrung durch das Forward-Verzerrungsmodell und eine Entzerrung durch das Backward-Verzerrungsmodell), müssten also die optimalen Parameter beider Modelle bekannt sein. Das wiederum bedeutet, dass zwei Optimierungsvorgänge notwendig wären.

2.5.2 Invertierung der Verzerrungsrichtung

Eine gängige Alternative zur Optimierung der Parameter beider Modelle besteht darin, nur die Parameter des Forward-Verzerrungsmodells, wie in Abschnitt 2.4 beschrieben, zu bestimmen und dann die analytische inverse Lösung zur Gleichung (2.56) zu berechnen. Dies ist jedoch nur für sehr einfache Linsenverzerrungsmodelle möglich, in denen z.B. nur die radiale Verzerrung für lediglich den ersten von κ_1 abhängigen Term

berücksichtigt wird [DF01]. Für ein komplexes Modell wie in den Gleichungen (2.13) und (2.14) definiert, ist dies allerdings nicht mehr möglich [Hei00, DF01].

Statt der exakten Berechnung kann durch das Lösen der Fixpunktgleichung

$$\mathbf{p}_u = \mathbf{p}_d - \mathcal{F}_D(\mathbf{p}_u, \delta_D), \quad (2.59)$$

die sich durch die Umstellung der Gleichung (2.56) ergibt, ein entzerrter Punkt \mathbf{p}_u mit Hilfe der Verzerrungsfunktion \mathcal{F}_D numerisch approximiert werden. Die iterative Lösung der Fixpunktgleichung ist dann gegeben durch

$$\begin{aligned} \mathbf{p}_u^0 &:= \mathbf{p}_d, \\ \mathbf{p}_u^i &:= \mathbf{p}_d - \mathcal{F}_D(\mathbf{p}_u^{i-1}, \delta_D), \quad i > 0. \end{aligned} \quad (2.60)$$

Laut [Hei00] sind in der Praxis mindestens drei Iterationen notwendig, um starke Verzerrungen zu kompensieren. In der OpenCV-Implementierung [Bra00], die, wie in Abschnitt 2.3.3 erwähnt, das Forward-Verzerrungsmodell nutzt, werden fünf Iterationen verwendet.

Sind nach der globalen Optimierung aller Parameter des Forward-Kameramodells (siehe Abschnitt 2.4.3), insbesondere die des Forward-Verzerrungsparameters δ_D bekannt, ist es möglich, sowohl entzerrte Punkte zu verzerren als auch verzerrte Punkte mit Hilfe der beschriebenen approximativen Lösung zu entzerren.

Die betrachtete Invertierung der Verzerrungsrichtung kann auch auf einen Punkt $\mathbf{p}_d^{Pix} = (u_d, v_d)^T$ im Pixelkoordinatensystem angewendet werden. Dazu muss dieser zunächst mit Hilfe der inversen intrinsischen Matrix \mathbf{K}^{-1} in Bildkoordinaten $\mathbf{p}_d = (x_d, y_d)^T$ transformiert werden. Seien $\tilde{\mathbf{p}}_d^{Pix}$ und $\tilde{\mathbf{p}}_d$ die beiden genannten Punkte in homogenen Koordinaten, dann gilt $\tilde{\mathbf{p}}_d = \mathbf{K}^{-1} \tilde{\mathbf{p}}_d^{Pix}$ mit

$$\mathbf{K}^{-1} = \begin{pmatrix} \frac{1}{\alpha} & \frac{-\gamma}{\alpha\beta} & \frac{-\beta u_0 + \gamma v_0}{\alpha\beta} \\ 0 & \frac{1}{\beta} & \frac{-v_0}{\beta} \\ 0 & 0 & 1 \end{pmatrix} \quad \text{bzw.} \quad \mathbf{K}^{-1} = \begin{pmatrix} \frac{1}{\alpha} & 0 & \frac{-u_0}{\alpha} \\ 0 & \frac{1}{\beta} & \frac{-v_0}{\beta} \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.61)$$

für den Fall, dass $\gamma \neq 0$ bzw. $\gamma = 0$ gilt. Nach der Entzerrung liegt \mathbf{p}_u vor und der entsprechende homogene Punkt $\tilde{\mathbf{p}}_u$ kann wieder in Pixelkoordinaten zurück transformiert werden durch $\tilde{\mathbf{p}}_u^{Pix} = \mathbf{K} \tilde{\mathbf{p}}_u$.

2.5.3 Entkoppeltes Backward-Verzerrungsmodell

Das Forward-Verzerrungsmodell (siehe Abschnitt 2.4) ist direkt gekoppelt an das Lochkameramodell. Mit dem Ziel, den Reprojektionsfehler mittels der Kostenfunktion (siehe Gleichung (2.22)) zu minimieren, wird nach angemessener Initialisierung der Verzerrungsparameter δ_D zusammen mit allen anderen intrinsischen und extrinsischen Parametern optimiert. Bei der Schätzung der intrinsischen und extrinsischen Parameter wird jedoch fälschlicherweise angenommen, dass keine Verzerrung vorliegt. Dies ist in der Realität selten der Fall, weshalb die Qualität der Schätzung, wie bereits in Abschnitt 2.4.3 erwähnt, von der tatsächlich vorliegenden Linsenverzerrung abhängt. Folglich wäre es wünschenswert, wenn die Verzerrungsparameter bereits bei der

Schätzung von \mathbf{K} sowie \mathbf{R}_i und \mathbf{t}_i vorliegen würden. Dies ist aber bei Verwendung des Forward-Verzerrungsmodells nicht möglich, weil zur Bestimmung der Verzerrungsparameter wiederum die vom Lochkameramodell vorhergesagten, (nicht beobachtbaren) unverzerrten Punkte notwendig sind. Somit kann die gegenseitige Abhängigkeit aufgrund der gegebenen Kopplung nicht aufgelöst werden.

Dieses Problem lässt sich umgehen, wenn die Verzerrungsparameter unabhängig vom restlichen Lochkameramodell bestimmt würden. Dabei kann ausgenutzt werden, dass ideale projektive Kameras die Eigenschaft der Geradlinigkeit (engl. "straightness") bei der Projektion von Linien erhalten [HZ03]. Das bedeutet, dass eine gerade Linie in der Szene auf eine gerade Linie im Kamerabild abgebildet wird. Wenn hingegen im Kamerabild eine gerade Linie gebogen erscheint, so kann im Umkehrschluss angenommen werden, dass die Linsenverzerrung dafür verantwortlich ist.

Die Grundidee ist nun, das Backward-Verzerrungsmodell zu nutzen (welches einem Entzerrungsmodell entspricht) und die Parameter so zu wählen, dass die beobachteten, gekrümmten Linien im Bild wieder gerade verlaufen. Typischerweise wird bei dem Vorgehen eine Kostenfunktion aufgestellt, die die Geradlinigkeit der beobachteten Linien nach einer Entzerrung mit den aktuellen verfügbaren Parametern beurteilt. Die optimalen Entzerrungsparameter sind dann jene, die diese Kostenfunktion minimieren.

Diese Idee wurde bereits in [Bro71] aufgegriffen, bei der mit der Kamera das Bild einer Lotlinie (Schnurlot) aufgezeichnet wurde. In den beiden Arbeiten [AF01, DF01] werden natürliche Szenen verwendet, von denen angenommen wird, dass sie gerade Linien enthalten (Türrahmen, Bilderrahmen usw.). Diese Linien werden in Form von Liniensegmenten im Kamerabild gesucht und weiter verarbeitet. Bevor die Variante des entkoppelten Backward-Verzerrungsmodells beschrieben wird, die im Gegensatz zu den beiden genannten Arbeiten einen erweiterten Parametersatz besitzt, soll hervorgehoben werden, dass das Backward-Verzerrungsmodell nur dann wirklich von den intrinsischen und extrinsischen Parametern entkoppelt ist, wenn es direkt auf den Pixelkoordinaten definiert wird. Denn die in Abschnitt 2.5.1 erläuterten Varianten des Forward- und Backward-Verzerrungsmodells arbeiten auf den Bildkoordinaten und nicht direkt auf den Pixelkoordinaten (siehe Abb. 2.8). Erst eine zusätzliche Transformation, die von den intrinsischen Parametern α, β, u_0 und v_0 abhängt, liefert dort die Beziehung zu den Pixelkoordinaten (siehe Gleichung (2.18)). Nachfolgend wird eine mathematische Formulierung des Backward-Verzerrungsmodells beschrieben, welches, wie gefordert, direkt auf den Pixelkoordinaten definiert ist.

In Gleichung (2.17) wurden in Pixelkoordinaten gegebene verzerrte Punkte durch $\mathbf{p}_d^{Pix} = (u_d, v_d)^T$ bezeichnet. Analog dazu sollen die entzerrten Punkte $\mathbf{p}_u^{Pix} = (u_u, v_u)^T$ lauten. Das entkoppelte Backward-Verzerrungsmodell soll gegeben sein durch

$$\mathbf{p}_u^{Pix} = \mathbf{p}_d^{Pix} + \mathcal{F}_U^{Pix}(\mathbf{p}_d^{Pix}, \delta_U^{Pix}) \quad (2.62)$$

mit

$$\begin{aligned} \mathcal{F}_U^{Pix}(\mathbf{p}_d^{Pix}, \delta_U^{Pix}) &= \begin{pmatrix} \bar{u}_d(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6) + [2\rho_1 \bar{u}_d \bar{v}_d + \rho_2(r^2 + 2\bar{u}_d^2)] \\ \bar{v}_d(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6) + [2\rho_2 \bar{u}_d \bar{v}_d + \rho_1(r^2 + 2\bar{v}_d^2)] \end{pmatrix} \\ &= \begin{pmatrix} \Delta_u^{Pix} \\ \Delta_v^{Pix} \end{pmatrix} \end{aligned} \quad (2.63)$$

und

$$\bar{u}_d = u_d - c_u, \quad \bar{v}_d = v_d - c_v, \quad r = \sqrt{\bar{u}_d^2 + \bar{v}_d^2}. \quad (2.64)$$

Der Vektor $(\bar{u}_d, \bar{v}_d)^T$ entspricht dem Differenzvektor zwischen dem verzerrten Punkt $(u_d, v_d)^T$ und dem Verzerrungszentrum $(c_u, c_v)^T$, also dem Vektor, dessen Betrag dem Radius r entspricht. Im Abschnitt 2.3.3 wurde das Verzerrungszentrum im Hauptpunkt $\mathbf{h} = (0, 0)^T$ angenommen. Da der Hauptpunkt aber Teil der intrinsischen Parameter ist, von denen das Modell entkoppelt sein soll, wird hier das Verzerrungszentrum explizit durch c_u und c_v in Pixelkoordinaten beschrieben. Somit gehören c_u und c_v mit zu den Parametern des entkoppelten Backward-Verzerrungsmodells und werden mit in den Parametervektor aufgenommen, so dass gilt:

$$\delta_U^{Pix} = (\kappa_1, \kappa_2, \kappa_3, \rho_1, \rho_2, c_u, c_v). \quad (2.65)$$

O.B.d.A. wurde hier die Einschränkung auf die ersten drei radialen Koeffizienten $\kappa_1, \kappa_2, \kappa_3$ und die ersten zwei tangentialen Koeffizienten ρ_1, ρ_2 , entsprechend der Argumentationen in Abschnitt 2.3.3, durchgeführt, weil dies dem Modell entspricht, welches in Kapitel 3 eingesetzt wird.

Im Folgenden soll für die Funktion $f_U^{Pix}(\mathbf{p}_d^{Pix}, \delta_U^{Pix})$, die als Backward-Verzerrungsfunktion bzw. Entzerrungsfunktion bezeichnet werden soll, gelten:

$$\mathbf{p}_u^{Pix} = f_U^{Pix}(\mathbf{p}_d^{Pix}, \delta_U^{Pix}) = \mathbf{p}_d^{Pix} + \mathcal{F}_U^{Pix}(\mathbf{p}_d^{Pix}, \delta_U^{Pix}). \quad (2.66)$$

Geht aus dem Kontext hervor, dass nur Pixelkoordinaten betrachtet werden, soll auf den Index Pix verzichtet werden dürfen und es gilt:

$$\mathbf{p}_u = f_U(\mathbf{p}_d, \delta_U) = \mathbf{p}_d + \mathcal{F}_U(\mathbf{p}_d, \delta_U). \quad (2.67)$$

Wie die Parameter der Entzerrungsfunktion f_U mit einem stochastischen Ansatz optimiert werden können, wird in Kapitel 3 behandelt.

3

Partikelfilter basierte Linsenverzerrungskorrektur

Viele Anwendungen im Bereich Computer Vision sind auf eine präzise Korrektur der Linsenverzerrung angewiesen, um eine korrekte Funktionsweise sicherzustellen. Gegenstand des Kapitels ist die Beschreibung eines neuen Verfahrens zur Linsenverzerrungskorrektur und der Nachweis seiner Leistungsfähigkeit anhand von Experimenten, insbesondere im Vergleich mit gängigen bestehenden Verfahren. Ausgangspunkt ist das im vorangegangenen Kapitel erläuterte entkoppelte Backward-Verzerrungsmodell (Entzerrungsmodell). Um eine präzise Korrektur zu ermöglichen, müssen optimale Parameter des Modells bestimmt werden. Es wird eine Zielfunktion vorgeschlagen, die das Maß der Linsenverzerrung beurteilen kann und die auf den Projektionseigenschaften gerader Linien in Kombination mit einfachen Kalibrierobjekten basiert. Anstatt eines deterministischen Optimierers, wird ein stochastischer Optimierungsansatz genutzt. Konkret wird der Condensation-Algorithmus, der zur Familie der Partikelfilter gehört, derart abgewandelt, dass er als nichtdeterministischer Optimierer für die Linsenparameter unter Einsatz der vorgeschlagenen Zielfunktion dient. Alle nötigen Parameter des vorgeschlagenen Optimierungsverfahrens werden bzgl. der Laufzeit und der resultierenden Korrekturleistung analysiert. Um verzerrte Bilder mit Hilfe des entkoppelten Backward-Verzerrungsmodells und den ermittelten optimierten Parametern effizient zu korrigieren, wird ein auf Look-Up Tabellen basierendes Bildkorrekturverfahren beschrieben.

In Abschnitt 3.1 werden die Motivation und Problemspezifikation erläutert. Als Lösung wird in Abschnitt 3.2 das neue Verfahren zur Linsenverzerrungskorrektur dargestellt. Die Evaluierung der Entzerrungsleistung sowie die Vergleiche mit alternativen Verfahren sind Gegenstand des Abschnitts 3.3. Der Abschnitt 3.4 widmet sich dem direkten Vergleich der Optimierer. Die Grenzen des vorgeschlagenen Verfahrens sind schließlich in Abschnitt 3.5 aufgezeigt.

3.1 Motivation und Problemspezifikation

Mit dem in Abschnitt 2.4 beschriebenen Kalibrierverfahren für das Forward-Kameramodell ist man in der Lage, die Linsenverzerrung zu bestimmen und mit Hilfe der in Abschnitt 2.5.2 erläuterten iterativen Lösung der Fixpunktgleichung auch die inverse Verzerrungsrichtung, also die Korrektur der Linsenverzerrung, zu berechnen. Dieses Kalibrierverfahren findet in der Literatur sowie in der Praxis häufig Verwendung, was insbesondere durch die verfügbaren Implementierungen in der bekannten freien

Bibliothek "OpenCV" [Bra00] oder auch der "Camera calibration toolbox for Matlab" [Bou10] gefördert wird. Das Verfahren liefert bereits eine gute Linsenkorrekturleistung, jedoch kann diese noch weiter verbessert werden, insbesondere bei wachsendem Rauschen und starker Linsenverzerrung.

3.1.1 Motivation

Die folgenden Überlegungen stellen den Ausgangspunkt für das in diesem Kapitel beschriebene Linsenkorrekturverfahren dar.

Beschränkung auf linsenrelevante Parameter. Das Kalibrierverfahren für das Forward-Kameramodell betrachtet die Linsenverzerrungsparameter bei der Optimierung nicht gesondert, sondern erst in einem globalen Optimierungsschritt (siehe Abschnitt 2.4.3), zusammen mit fünf intrinsischen sowie bei n Ansichten mit $6n$ extrinsischen Parametern (je drei für jede Rotationsmatrix, die sich mit Hilfe der Rodrigues-Formel [HZ03] als 3D-Rotationsvektor darstellen lässt, und drei für jeden Translationsvektor). Im Allgemeinen wirkt sich eine steigende Parameteranzahl negativ auf die Stabilität und das Auffinden eines globalen Optimums aus. Da im Folgenden der Fokus auf die Linsenverzerrungskorrektur gesetzt wird, soll ein Ansatz verwendet werden, der sich ausschließlich auf die linsenrelevanten Parameter konzentriert.

Hierzu wurden im Abschnitt 2.5.3 bereits einige Vorüberlegungen gemacht, die hier aufgegriffen werden, und die es ermöglichen, die Verzerrungskorrektur der Linse separat zu betrachten und zu modellieren. Dabei wird die Eigenschaft genutzt, dass gerade Linien nach der Projektion ebenfalls gerade sein müssen, und eine Verletzung dieser Eigenschaft auf eine vorhandene Linsenverzerrung zurückzuführen ist.

Nutzung eines umfangreichen linsenrelevanten Parametersatzes. In den Arbeiten [AF01] und [DF01], die auf der beschriebenen Eigenschaft des Erhalts gerader Linien basieren, wird für den Vergleich der dort vorgeschlagenen Verfahren nur ein reduzierter Parametersatz (nur je ein Parameter für die radiale (κ_1) und tangential Komponente (ρ_1)) verwendet. In [EMF03] wird zwar ein Entzerrungsmodell beschrieben, welches dem in Gleichung (2.63) ähnelt, jedoch werden nur κ_1 und κ_2 genutzt, und die tangential Verzerrung wird gänzlich vernachlässigt. Insbesondere bei kostengünstig gefertigten Kameras und solchen mit einem großen Öffnungswinkel (geringe Brennweite) können die reduzierten Parametersätze zu verringerten Korrekturleistungen führen. Deshalb soll im Folgenden ein umfangreicher Parametersatz mit $\kappa_1, \kappa_2, \kappa_3, \rho_1, \rho_2, c_u, c_v$ genutzt und somit sowohl die radiale als auch die tangential Verzerrung berücksichtigt werden, wie im Abschnitt 2.5.3 beschrieben.

Robuste Detektion gerader Linien. Wie bereits in Abschnitt 2.5.3 erwähnt, verwenden die Arbeiten [AF01, DF01, EMF03] natürliche Szenen, um die als Geraden angenommenen Liniensegmente zu extrahieren. Ein Vorteil dieser Verfahren ist, dass eine Linsenentzerrung auf den Bildern durchgeführt werden kann, ohne den expliziten Einsatz von Kalibrierobjekten. Die Nutzung natürlicher Szenen hat jedoch auch Nachteile. Der in Abb. 3.1(a) dargestellte Galerieraum liefert zahlreiche gerade Linien, jedoch ist die Verfügbarkeit einer solchen Szene als gering einzuschätzen. In

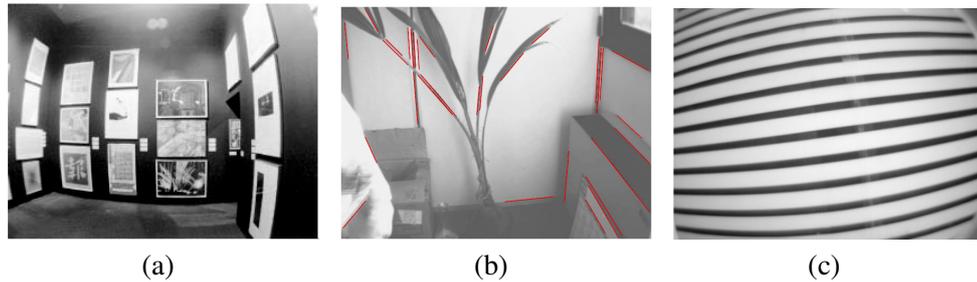


Abbildung 3.1: Zur Kalibrierung verwendete, verzerrte Bilder natürlicher Szenen aus den Arbeiten [AF01, DF01], aus denen Linien für die Linsenkalibrierung extrahiert wurden, die in der Welt als gerade angenommen worden sind. Zu sehen ist ein Raum in einer Kunstgalerie (a), ein Büro mit Topfpflanze (b) und eine Fenster-Jalousie (c).

Abb. 3.1(b) sind die automatisch ermittelten Kantenzüge rot markiert, unter anderem auf der Zimmerpflanze, die nicht zwangsläufig gerade gewachsen sein muss. Daran wird eine potentielle Fehlerquelle des Verfahrens deutlich, da offensichtlich nicht alle als gerade angenommenen Linien in Wirklichkeit gerade sind. Diese Fehlerquelle ist als umso kritischer einzustufen, je stärker die Linsenverzerrung ist, da bei stärkerer Verzerrung auch die Akzeptanz der gekrümmten Liniensegmente im Bild steigen muss. Auch die in Abb. 3.1(c) gezeigte Jalousie kann Verformungen aufweisen, die die geforderte Eigenschaft von geraden Linien verletzt. Somit werden Anforderungen an natürliche Szenen gestellt, die möglicherweise nicht erfüllt sind und somit eine Fehlerquelle darstellen.

Ist jedoch eine möglichst präzise und zuverlässige Korrektur gefordert, so kann die in den genannten Arbeiten genutzte Eigenschaft gerader Linien mit einem einfachen planaren Kalibrierobjekt, wie es z.B. in [Tsa85, Zha98] verwendet oder in Abb. 5.18 dargestellt ist, kombiniert werden. Die Extraktion der benötigten Linien aus dem Bild eines planaren Kalibrierobjekts mit einem einfachen Schachbrettmuster hat die folgenden Vorteile: Die Ecken des Schachbrettmusters liefern automatisch eine Gitterstruktur aus Linien, die bis auf Toleranzen des verwendeten Druckers als ideal und deutlich genauer als Linien in natürlichen Szenen angenommen werden können. Dies steigert die Robustheit des Verfahrens. Zudem kann das Auffinden der Kreuzungspunkte aufgrund des hohen Kontrastes und der bekannten Geometrie zuverlässig und automatisch geschehen. Ein Beispiel für aufgefundene Kreuzungspunkte im Schachbrettmuster des im Folgenden verwendeten Kalibrierobjekts ist in Abb. 3.3(b) gegeben.

Überwindung lokaler Optima und Robustheit gegenüber Rauschen. Bei den bislang betrachteten Verfahren werden die Modellparameter in einem globalen Schritt mit Hilfe von deterministischen Optimierungsverfahren bestimmt. Speziell bei der Optimierung der nichtlinearen Linsenverzerrungsparameter werden die Kostenfunktionen typischerweise mit dem Levenberg-Marquardt-Algorithmus minimiert, der einer Kombination aus dem Gradientenabstiegs- und dem Gauss-Newton-Verfahren entspricht, die ebenfalls beide deterministische Verfahren sind. Ausgehend von einem initialen Punkt im Parameterraum ist das Optimierungsverfahren in der Lage, den Punkt im Pa-

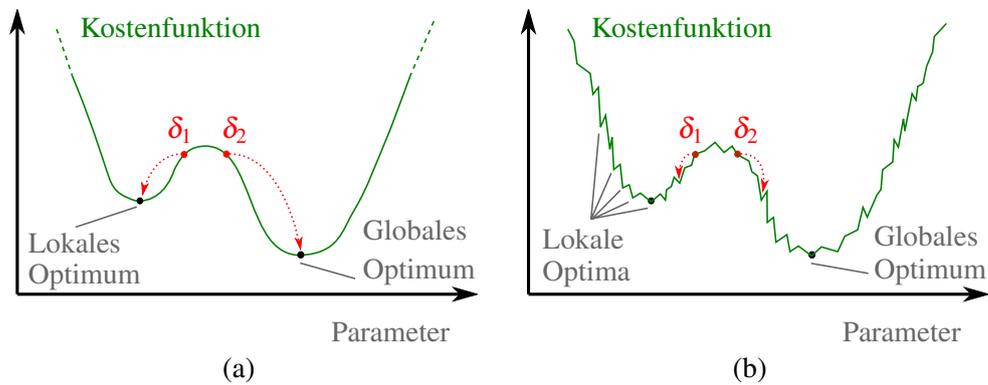


Abbildung 3.2: Die initialen Startpunkte für das Optimierungsproblem sollen durch δ_1 und δ_2 dargestellt werden. In Abhängigkeit von dem Startpunkt kann die Suche nur in einem lokalen oder aber im globalen Optimum enden (a). In (b) soll die Auswirkung verrauschter Daten verdeutlicht werden, die die Suche nach dem globalen Optimum weiter erschwert.

parameterraum zu finden, der ein Optimum darstellt. Existieren jedoch mehrere Optima, so hängt es von dem initialen Startpunkt ab, ob bei dem Gradientenabstieg nur ein lokales oder ein globales Optimum erreicht wird (siehe Abb. 3.2(a)). Somit ist, wie bereits in Abschnitt 2.4.3 erwähnt, die Wahl der initialen Parameter von Bedeutung.

Im Fall von CCD- und CMOS-Kerasensoren kommt erschwerend hinzu, dass die Daten ein nichtdeterministisches Rauschen aufweisen. Da dieses Rauschen eine Auswirkung auf die von den Daten abhängige Zielfunktion hat, wird die Suche nach einem globalen Optimum erschwert, da durch das Rauschen eine Vielzahl lokaler Optima eingeführt werden könnte, in denen das deterministische Optimierungsverfahren potenziell vorzeitig terminieren würde (siehe Abb.3.2(b)).

Stochastische Optimierungsverfahren sind im Gegensatz zu deterministischen Verfahren in der Lage, einen Zufallsprozess in die Suchstrategie zu integrieren und mehrere Lösungswege simultan zu verfolgen und zu beurteilen [GHM04]. Dadurch besteht generell die Möglichkeit, trotz ungünstig gewählter initialer Parameter und verrauschter Daten ein lokales Optimum zu überwinden und das globale Optimum zu erreichen.

Ein stochastischer Optimierungsansatz, bei dem eine Wahrscheinlichkeitsverteilung für die Modellparameter mit Hilfe des sog. "Sum-of-Gaussians (SOG)"-Ansatzes geschätzt wird, ist z.B. in [CBDM09] zu finden. Neben einem reduzierten intrinsischen Parametersatz wird dort lediglich ein Teil der radialen Verzerrungsparameter optimiert und die Tangentialverzerrung gar nicht berücksichtigt. In [QC02] werden die intrinsischen Parameter unter Verwendung des sog. "Sequential-Importance-Sampling (SIS)"-Algorithmus geschätzt, der ursprünglich aus dem Bereich der Partikelfilter [DJ11] stammt. Zwar wird die Linsenverzerrung dort gänzlich vernachlässigt, jedoch scheint ein Partikelfilter basierter Ansatz für die Optimierung bei verrauschten Daten ein vielversprechender Ansatz zur präzisen Bestimmung der linsenrelevanten Parameter zu sein.

3.1.2 Problemspezifikation

Aus den Vorüberlegungen ergibt sich nun die folgende Problemspezifikation. Gegeben sei ein Kalibrierobjekt mit einem Schachbrettmuster wie in Abb. 3.3(b) dargestellt. Als Eingabe sollen ein einzelnes Bild, eine Menge von Bildern oder ein Live-Videostrom der Kamera, deren Linsenverzerrung korrigiert werden soll, vorliegen, die das verzerrt abgebildete Kalibrierobjekt enthalten. Dabei dürfen die Bilder Sensorrauschen sowie Unschärfe, z.B. aufgrund von falscher Fokussierung, aufweisen. Gesucht sind die optimalen Parameter $\kappa_1, \kappa_2, \kappa_3, \rho_1, \rho_2, c_u, c_v$, mit deren Hilfe eine schnelle Berechnung von einzelnen korrigierten Bildpositionen ermöglicht werden soll, ohne auf eine approximative, iterative Lösung zugreifen zu müssen (vgl. Abschnitt 2.5.2), sowie eine Möglichkeit, ein komplettes entzerrtes Kamerabild effizient zu berechnen.

Die Aufgabe der Linsenverzerrungskorrektur wird als stochastisches Optimierungsproblem betrachtet. Dabei gilt es die nachfolgend definierte Zielfunktion zu minimieren, die auf dem Prinzip gerader Linien basiert. Diese Funktion wird zum einen verwendet, um die aktuelle Güte der Parameter während des Optimierungsprozesses zu beurteilen, und zum anderen um die Stärke der Linsenverzerrung in einem gegebenen Bild zu bewerten.

Ausgehend von dem verzerrten Kamerabild, welches das planare Kalibrierobjekt mit einem Schachbrettmuster beinhaltet, werden zunächst die Koordinaten aller Kreuzungspunkte $\hat{\mathbf{p}}_{i,j}$, die von jeweils vier Schachbrettkästchen umgeben sind, extrahiert (siehe Abb. 3.3(b)). Aufgrund des gewählten Schachbrettmusters kann die Extraktion vollautomatisch und mit Subpixelgenauigkeit, z.B. mit Hilfe entsprechender OpenCV-Methoden [Bra00] geschehen.

Bei Betrachtung eines idealen, unverzerrten Schachbrettmusters mit quadratischen Schachbrettkästchen definieren die Kreuzungspunkte ein regelmäßiges Gitter (siehe Abb. 3.3(a)), so dass sich jeder ideale Kreuzungspunkt $\mathbf{p}_{i,j}$ auf dem Schnittpunkt zweier Gitterlinien in der i -ten Zeile und j -ten Spalte des Gitters befindet. Durch entsprechende Verbindung außen liegender Kreuzungspunkte lassen sich horizontale, vertikale sowie diagonale Geraden definieren, wie in Abb. 3.3(c) dargestellt. Im Idealfall liegen die entsprechenden inneren Kreuzungspunkte exakt auf den jeweiligen Geraden. Liegt jedoch eine durch die Linse verursachte Verzerrung vor, wird diese Eigenschaft verletzt, und die inneren Punkte weisen einen gewissen Abstand (Abweichung) zu den Geraden auf (siehe Abb. 3.3(d)). Diese Abweichung wird für die Beschreibung des Verzerrungsfehlers genutzt, wobei gelten soll, dass diese um so größer wird, je mehr Verzerrung durch die Linse verursacht wird.

3.1.2.1 Zielfunktion

Seien $\hat{\mathbf{p}}_{i,j}, i = 0, \dots, k-1, j = 0, \dots, l-1$, die extrahierten verzerrten Kreuzungspunkte (KP) der i -ten Zeile und j -ten Spalte des Gitters, k die Anzahl der horizontalen und l die Anzahl der vertikalen Linien. Die Zielfunktion \mathcal{E}_{KP} , die auch als Fehlerfunktion (engl. "error function") betrachtet werden kann und die von der Menge der extrahier-

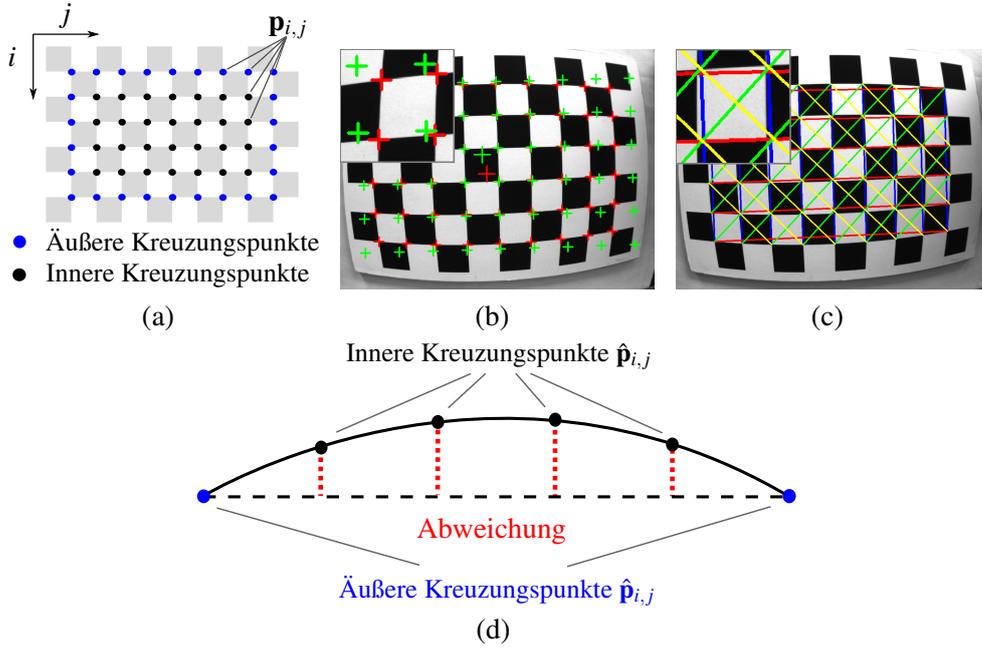


Abbildung 3.3: Innere und äußere Schachbrett-Kreuzungspunkte $\mathbf{p}_{i,j}$, die von jeweils vier Schachbrettkästchen umgeben sind (a). Die automatisch gefundenen (beobachtbaren) Kreuzungspunkte $\hat{\mathbf{p}}_{i,j}$ in dem verzerrten Bild des Kalibrierobjekts sind in (b) durch die roten Kreuze markiert. Die grün markierten Punkte entsprechen den entzerrten Punktkoordinaten $\mathbf{p}_{i,j}$, auf denen die Kreuzungspunkte zu erwarten sind, wenn keine Linsenverzerrung vorliegen würde. Die Zielfunktion basiert auf den in (c) gezeigten Abständen von den horizontalen (rot), vertikalen (blau) sowie diagonalen Geraden (fallend: gelb, steigend: grün) zu den Kreuzungspunkten, durch die die Geraden idealerweise verlaufen müssten, es aufgrund der Verzerrung jedoch nicht tun. Die Abweichungen von den beobachtbaren inneren Kreuzungspunkten zur idealen Verbindungslinie zwischen zwei äußeren Kreuzungspunkten sind in (d) dargestellt.

ten, verzerrten Kreuzungspunkte $\{\hat{\mathbf{p}}_{i,j}\}$ abhängt, wird definiert durch

$$\mathcal{E}_{KP}(\{\hat{\mathbf{p}}_{i,j}\}) = e_h + e_v + e_{d_+} + e_{d_-}, \quad (3.1)$$

wobei

$$e_h = \frac{1}{k \cdot (l-2)} \sum_{i=0}^{k-1} \sum_{j=1}^{l-2} \frac{|d_G(\hat{\mathbf{p}}_{i,j}, \mathcal{G}(\hat{\mathbf{p}}_{i,0}, \hat{\mathbf{p}}_{i,l-1}))|}{\|\mathcal{G}(\hat{\mathbf{p}}_{i,0}, \hat{\mathbf{p}}_{i,l-1})\|} \quad (3.2)$$

die mittleren normierten Abweichungen aller horizontalen und

$$e_v = \frac{1}{l \cdot (k-2)} \sum_{j=0}^{l-1} \sum_{i=1}^{k-2} \frac{|d_G(\hat{\mathbf{p}}_{i,j}, \mathcal{G}(\hat{\mathbf{p}}_{0,j}, \hat{\mathbf{p}}_{k-1,j}))|}{\|\mathcal{G}(\hat{\mathbf{p}}_{0,j}, \hat{\mathbf{p}}_{k-1,j})\|} \quad (3.3)$$

jene aller vertikalen Geraden betrachtet. Mit e_{d_+} und e_{d_-} werden die mittleren normierten Abweichungen für die steigenden und fallenden Diagonalen bezeichnet und auf analoge Art und Weise berechnet. $\mathcal{G}(\mathbf{p}, \mathbf{q})$ bezeichnet den Geradenabschnitt zwischen den beiden Punkten \mathbf{p} und \mathbf{q} und $|d_G(\mathbf{s}, \mathcal{G})|$ den Betrag der Distanz zwischen

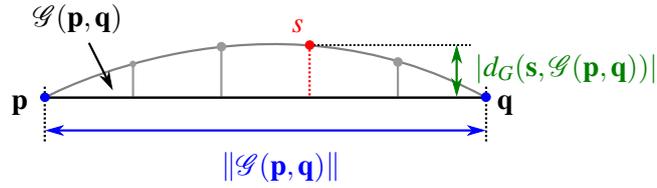


Abbildung 3.4: Verdeutlichung der wesentlichen Komponenten der Gleichungen (3.2) und (3.3).

einem Punkt s und dem Geradenabschnitt \mathcal{G} , siehe Abb. 3.4. Dabei werden in den Gleichungen (3.2) und (3.3) die Punkte \mathbf{p} und \mathbf{q} immer derart gewählt, dass die senkrechte Projektion von s auf den Geradenabschnitt stets zwischen den Punkten \mathbf{p} und \mathbf{q} liegt.

Durch die Multiplikation mit den Faktoren $\frac{1}{k \cdot (l-2)}$ und $\frac{1}{l \cdot (k-2)}$ in den Gleichungen (3.2) und (3.3) wird die Invarianz gegenüber der Anzahl der Zeilen k und Spalten l gewährleistet. Somit wird sichergestellt, dass der Wert der Zielfunktion \mathcal{E}_{KP} unabhängig von der verwendeten Anzahl der Kreuzungspunkte des Schachbrettmusters ist.

Aufgrund der projektiven Kameraeigenschaften erscheint das Kalibrierobjekt im Kamerabild größer, wenn es näher an die Kamera gehalten wird. Dadurch wird auch der durch zwei äußere Kreuzungspunkte \mathbf{p} und \mathbf{q} definierte Geradenabschnitt $\mathcal{G}(\mathbf{p}, \mathbf{q})$ sowie der Abstand eines Punktes s zu dem Geradenabschnitt länger. Deshalb wird eine Normierung der jeweiligen Abstände $|d_G(s, \mathcal{G}(\mathbf{p}, \mathbf{q}))|$ mit der Länge des Geradenabschnitts $\|\mathcal{G}(\mathbf{p}, \mathbf{q})\|$ durchgeführt. Zusätzlich sichert diese Normierung eine Bewertung der Verzerrung, unabhängig von der vorliegenden Auflösung des Bildes. Dies ist eine wichtige Eigenschaft, um die Vergleichbarkeit zu gewährleisten, falls sich z.B. bedingt durch unterschiedliche Bildentzerrungsverfahren die Auflösung der Bilder ändern sollte. Da sowohl $|d_G(s, \mathcal{G}(\mathbf{p}, \mathbf{q}))|$ als auch $\|\mathcal{G}(\mathbf{p}, \mathbf{q})\|$ in Pixeln gegeben sind, folgt aus deren Quotient (siehe Formeln (3.2) und (3.3)), dass der durch \mathcal{E}_{KP} ermittelte Abweichungsfehler ohne Einheit ist.

Somit beschreibt die Zielfunktion bzw. Fehlerfunktion \mathcal{E}_{KP} mit den beschriebenen Normierungen den ermittelten Verzerrungsfehler anhand der Kreuzungspunkte innerhalb des Bildbereichs, in dem sich das Kalibrierobjekt befindet. Da die Verzerrung in der Bildmitte gering ist, zu den Bildrändern hin jedoch stärker wird, sollte das Kalibrierobjekt einen möglichst großen Teil des Kamerabildes abdecken. Es sei angemerkt, dass es für den beschriebenen Ansatz keine Einschränkungen oder Vorgaben bzgl. der Orientierung der Kalibrierobjekte in den Kalibrierbildern gibt, da die projektive Eigenschaft gerader Linien auch bei geneigten und somit perspektivisch verzerrt abgebildeten Kalibriermustern erhalten bleibt.

3.1.2.2 Weitere Fehlerfunktionen

Zusätzlich zur Fehlerfunktion \mathcal{E}_{KP} , die den Verzerrungsfehler auf Basis der Kreuzungspunkte ermittelt, wird mit $\mathcal{E}_{Bild}(\mathbf{B})$ nachfolgend die Fehlerfunktion bezeichnet,

die als Argument ein Kamerabild \mathbf{B} entgegennimmt. Sei

$$\mathcal{F}_{ExtKP}(\mathbf{B}) = \{\hat{\mathbf{p}}_{i,j}\} \quad (3.4)$$

eine Funktion, die für ein Bild \mathbf{B} , welches das Kalibrierobjekt beinhaltet, die Menge der Kreuzungspunkte $\{\hat{\mathbf{p}}_{i,j}\}$ extrahiert. Dann gilt

$$\mathcal{E}_{Bild}(\mathbf{B}) = \mathcal{E}_{KP}(\mathcal{F}_{ExtKP}(\mathbf{B})). \quad (3.5)$$

Als Erweiterung zur Berechnung des Verzerrungsfehlers auf einem einzigen Bild wird weiterhin die Funktion $\mathcal{E}_{Bildmenge}(\mathcal{B})$ definiert, die den mittleren Verzerrungsfehler der $m \geq 1$ Bilder einer gegebenen Bildmenge $\mathcal{B} = \{\mathbf{B}_0, \dots, \mathbf{B}_{m-1}\}$ liefert:

$$\mathcal{E}_{Bildmenge}(\mathcal{B}) = \frac{1}{m} \sum_{\xi=0}^{m-1} \mathcal{E}_{KP}(\mathcal{F}_{ExtKP}(\mathbf{B}_{\xi})). \quad (3.6)$$

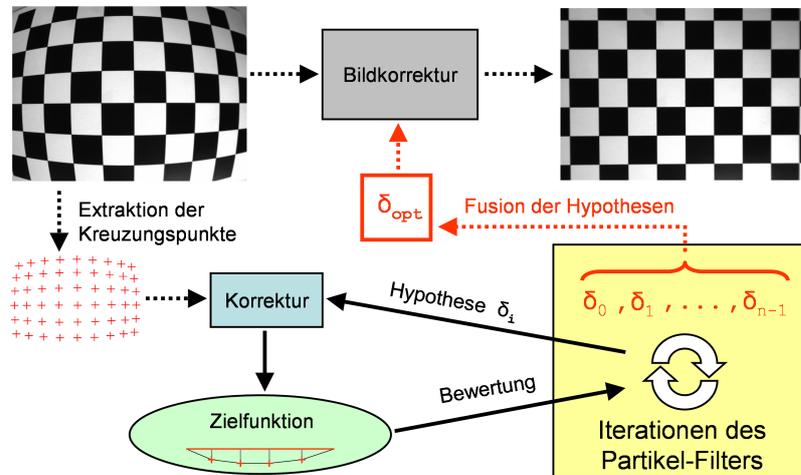
3.2 Lösung

Wie das spezifizierte Problem unter Berücksichtigung der Vorüberlegungen gelöst werden kann, wird in diesem Abschnitt erläutert. Die Grundidee des Lösungsansatzes wird beschrieben und in den darauf folgenden Abschnitten genauer ausgearbeitet. Der vorgeschlagene Lösungsansatz ermöglicht eine stochastische Optimierung der linsenrelevanten Parameter unter Berücksichtigung der in Abschnitt 3.1.2.1 formulierten Zielfunktion. Es werden Terminierungskriterien ausgearbeitet sowie die Parameter, die für den Optimierungsansatz relevant sind, untersucht.

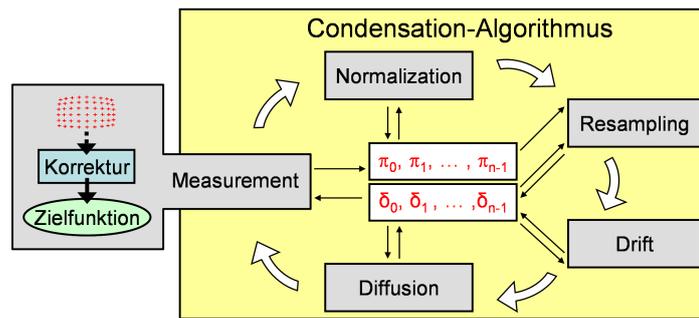
3.2.1 Grundidee

Den Kern der Lösung stellt der in [IB98] beschriebene Condensation-Algorithmus dar, mit dessen Hilfe die gesuchten Parameter des Entzerrungsmodells stochastisch optimiert werden. Der Condensation-Algorithmus gehört zur Familie der sequenziellen Monte-Carlo-Methoden (SMC-Methoden), die auch Partikelfilter genannt werden [DdFG01].

In seinem ursprünglichen Anwendungsgebiet wird der Condensation-Algorithmus verwendet, um Objekte in Videosequenzen zu verfolgen. Dabei werden die Objektkonturen durch B-Spline-Kurven repräsentiert. Ziel ist es, die multidimensionale Wahrscheinlichkeitsverteilung für die positions- und orientierungsrelevanten Parameter mit Hilfe einer Stichprobe (engl. "sample") zu approximieren. Bei dieser Approximation besteht die Stichprobe aus sog. Partikeln, die jeweils eine Hypothese über die aktuelle Objektposition und -orientierung speichern. Da sich die zu verfolgenden Objekte bewegen können, muss dies in jeder Iteration des Algorithmus berücksichtigt werden. Dabei wird für jede Hypothese eine Vorhersage generiert, die einen deterministischen und einen nichtdeterministischen Bewegungsanteil enthält. Als deterministischer Anteil wird z.B. ein lineares Bewegungsmodell des Objekts angenommen, so dass sich die vorhergesagte Position z.B. aus der aktuellen Position und der geschätzten Objektgeschwindigkeit berechnen lässt. Zu dieser vorhergesagten Position wird ein nichtdeterministischer (zufälliger) Bewegungsanteil hinzugefügt, der die Unsicherheit der



(a)



(b)

Abbildung 3.5: Übersicht über die Grundidee zur Partikelfilter basierten Optimierung der Entzerrungsparameter (a). Die einzelnen Hypothesen, die durch $\delta_0, \dots, \delta_{n-1}$ dargestellt sind, beschreiben jeweils einen konkreten Parametersatz des Entzerrungsmodells. Die Hypothesen werden innerhalb des iterativen Optimierungsprozesses (durch die beiden Kreisfeile angedeutet) fortlaufend verbessert. Dabei wird innerhalb einer Iteration jede Hypothese über einen Parametervektor δ_i entsprechend seiner Entzerrungsleistung beurteilt, wobei die Entzerrung anhand der aus dem verzerrten Bild extrahierten Kreuzungspunkte durchgeführt und das Resultat durch die Zielfunktion bewertet wird. Diese Bewertung der Hypothese wird als Gewichtungsfaktor π_i hinterlegt und wirkt sich auf die nächste Iteration des Partikelfilters aus. Ist ein entsprechendes Terminierungskriterium erreicht, werden die Hypothesen geeignet fusioniert und der optimale Parametervektor δ_{opt} bestimmt. Dieser wird bei der Bildkorrektur verwendet, so dass schließlich die Linsenverzerrung im Kamerabild entfernt werden kann. In (b) sind die Teilschritte des Condensation-Algorithmus, die innerhalb einer Iteration des Partikelfilters durchlaufen werden, sowie der Bezug auf die Hypothesen δ_i und die Gewichtungsfaktoren π_i der Partikel verdeutlicht.

Positionsschätzung oder ein gewisses Rauschen ausdrückt. Um die durch das Bewegungsmodell vorhergesagten Hypothesen zu beurteilen, nutzt der Algorithmus visuelle Beobachtungen, die auf Übereinstimmungen der B-Spline-Kurven mit der Objektkontur an den durch die Hypothesen vermuteten Positionen im Kamerabild basieren. Diese Beobachtungen werden genutzt, um die Partikel zu gewichten. Dabei werden Partikel, deren Hypothesen sich gut mit der Beobachtung im Bild decken, stärker gewichtet, wodurch ausgedrückt wird, dass die jeweilige Hypothese eine höhere Wahrscheinlichkeit besitzt. Über die Zeit hinweg liegt eine immer bessere Approximation der durch die Partikel dargestellten Wahrscheinlichkeitsverteilung der gesuchten Parameter vor. Der Punkt im Parameterraum, der das Maximum der Wahrscheinlichkeitsverteilung darstellt, entspricht der wahrscheinlichsten Hypothese über die Position und Lage des verfolgten Objekts im Bild. Der beschriebene Algorithmus erweist sich als besonders robust gegenüber Rauschen sowie fehlerhaften oder unvollständigen Beobachtungen.

Die Grundidee ist nun, den Condensation-Algorithmus für den hier betrachteten Kontext anzupassen und für die Bestimmung optimaler Entzerrungsparameter zu nutzen. Anstatt der ursprünglichen Objektlage und -position sollen die jeweiligen Partikel nun Hypothesen über mögliche Parametervektoren δ_i speichern. Ziel ist es, die Hypothesen (und somit die Linsenentzerrungsparameter) über die Iterationen des Partikelfilters immer weiter zu verbessern und schließlich die optimalen Parameter zu erhalten. Um eine Hypothese über einen Parametervektor δ_i zu beurteilen, werden zunächst die Kreuzungspunkte des Schachbrettmusters aus einem gegebenen verzerrten Kamerabild extrahiert. Anschließend werden die Positionen der Kreuzungspunkte unter Verwendung des Parametervektors δ_i und des Entzerrungsmodells korrigiert und die Güte der Korrektur mit Hilfe der Zielfunktion bewertet. Die Bewertung der Korrektur wird für die Gewichtung π_i des Partikels verwendet, zu dem die jeweilige Hypothese gehört, wobei ein hohes Gewicht eine "gute" bzw. wahrscheinliche Hypothese ausdrückt und umgekehrt. Über die Iterationen des Partikelfilters werden wahrscheinliche Hypothesen aufrecht erhalten, und unwahrscheinliche verworfen.

Im ursprünglichen Anwendungsszenario änderten sich die Lage- und Positionsparameter der sich bewegenden Objekte fortlaufend. Im Gegensatz dazu werden die Abbildungseigenschaften des Objektivs und somit die gesuchten Parameter des Linsenentzerrungsmodells als statisch angenommen. Daher ist zu erwarten, dass über die Iterationen hinweg die durch die Hypothesen approximierte Wahrscheinlichkeitsverteilung konvergiert und der iterative Prozess beendet werden kann. Die resultierenden Hypothesen, die die finale Wahrscheinlichkeitsverteilung der Modellparameter im Entzerrungsparameterraum repräsentieren, können geeignet fusioniert werden, so dass der optimale Parametervektor zur Verfügung steht, der schließlich zur Bildentzerrung genutzt werden kann. Die Abb. 3.5 verdeutlicht die beschriebene Grundidee.

3.2.2 Funktionsweise und Anpassung des iterativen Partikelfilters

Die einzelnen Schritte des Condensation-Algorithmus werden im Folgenden beschrieben und so angepasst, dass dieser als stochastischer Optimierer für die gesuchten Entzerrungsparameter dient. Gesucht ist die Wahrscheinlichkeitsverteilung im multidimensionalen

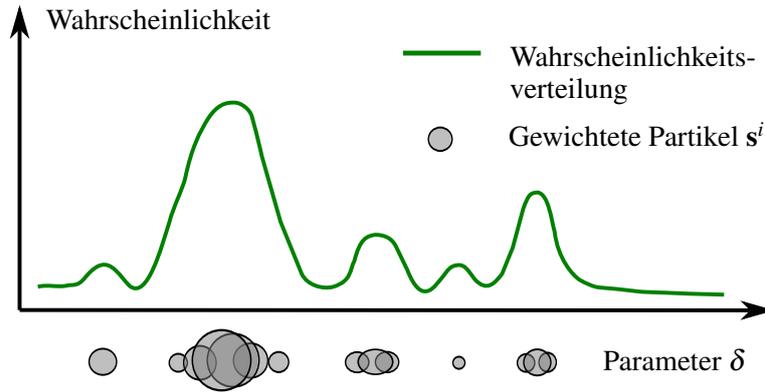


Abbildung 3.6: Approximation der Wahrscheinlichkeitsverteilung durch die Partikel am eindimensionalen Beispiel. Die Partikel $s^i = (\delta^i, \pi^i)$ sind durch Ellipsen dargestellt, wobei die Position der Ellipsen vom konkreten Parametervektor δ^i abhängt, und ihre Größe dem Partikelgewicht π^i entspricht. Häufungen von Partikeln sowie große Partikelgewichte sind an den Stellen vorzufinden, die hohen Wahrscheinlichkeiten der Verteilung entsprechen. Somit stellen die Partikel diskrete Abtastungen im Parameterraum dar.

mensionalen Entzerrungsparameterraum, deren Maximum bei der Parameterkonfiguration angenommen wird, die die optimale Linsenentzerrung ermöglicht. Als Approximation wird diese Wahrscheinlichkeitsverteilung ("posterior density") repräsentiert durch die Partikelmenge ("sample set")

$$\mathcal{S}_t = \{s_t^0, \dots, s_t^{n-1}\} = \{(\delta_t^0, \pi_t^0), \dots, (\delta_t^{n-1}, \pi_t^{n-1})\} \quad (3.7)$$

aus n Partikeln zum Zeitpunkt t . Der Parameter t kann auch als Iterationsparameter verstanden werden, über den der Condensation-Algorithmus die approximierende Wahrscheinlichkeitsverteilung immer weiter anpasst. Jedes Partikel $s_t^i = (\delta_t^i, \pi_t^i)$ besteht dabei aus dem Zustandsvektor ("state vector") δ_t^i , der im gegebenen Kontext eine Hypothese der Entzerrungsparameter speichert (siehe Gleichung (2.65)), und dem Partikel-Gewichtungsfaktor π_t^i . Der Gewichtungsfaktor beschreibt dabei die Güte des Zustandsvektors des Partikels, die mit Hilfe der Zielfunktion beurteilt werden kann. Die Abb. 3.6 verdeutlicht den Zusammenhang zwischen den Partikeln mit ihren Gewichtungsfaktoren und der zu approximierenden Wahrscheinlichkeitsverteilung für den eindimensionalen Fall. Die Gesamtdichte ergibt sich durch eine normierte additive Mischverteilung ("additive Gaussian mixture"). Dabei wird der Einfluss auf den Beitrag einer einzelnen Normalverteilung zur Gesamt-Mischverteilung durch das Gewicht und den Zustandsvektor eines Partikels bestimmt, Details siehe [IB98].

Ein Iterationsschritt von der "alten" Partikelmenge \mathcal{S}_{t-1} zum Zeitpunkt $t-1$ zu der "neuen" Partikelmenge \mathcal{S}_t zum Zeitpunkt t besteht aus den fünf Schritten *Resampling*, *Drift*, *Diffusion*, *Measurement* und *Normalization*, die jeweils in den nachfolgenden Abschnitten detailliert erläutert werden und in Abb. 3.5(b) dargestellt sind. Es folgt die Beschreibung, wie die einzelnen Partikel bzw. Hypothesen der Entzerrungsparameter sinnvoll fusioniert werden können, um schließlich die gesuchten optimalen Entzerrungsparameter zu erhalten.

3.2.2.1 Resampling

Jedes Partikel mit seinem Parametervektor stellt einen Abtastpunkt im Parameterraum dar. Da der Parameterraum mehrdimensional und die Anzahl der Partikel konstant ist, wobei diese um Rechenzeit zu sparen möglichst klein sein soll, liegt nur eine spärliche Abtastung vor. Bereiche der Wahrscheinlichkeitsverteilung mit sehr geringen Wahrscheinlichkeiten sind uninteressant und können vernachlässigt werden. Das bedeutet, dass sie nach Möglichkeit nicht von Partikeln abgetastet werden sollten. Die Bereiche mit einer hohen Wahrscheinlichkeit hingegen sollten möglichst genau abgetastet werden, da dort die optimalen Parameter vermutet werden. Deshalb wird in jedem Iterationsschritt eine Neu- bzw. Umabtastung durchgeführt ("resampling"), um die beschriebene Eigenschaft zu begünstigen.

Sei die Partikelmenge $\mathcal{S}_{t-1} = \{(\delta_{t-1}^i, \pi_{t-1}^i) | i = 0, \dots, n-1\}$ zum Zeitpunkt $t-1$ gegeben. Dabei gilt, dass δ_{t-1}^i eines Partikels mit einem großen Gewichtungsfaktor π_{t-1}^i "gute" Entzerrungsparameter speichert. Aus dieser Menge werden nun n Partikel mit Zurücklegen zufällig gezogen, wobei ein bestimmtes Partikel $(\delta_{t-1}^i, \pi_{t-1}^i)$ mit der Wahrscheinlichkeit π_{t-1}^i gezogen wird. Dadurch wird erreicht, dass Partikel mit "guten" Parametern häufiger und sogar mehrfach, während Partikel mit kleinen Gewichtungsfaktoren und den korrespondierenden "schlechten" Entzerrungsparametern seltener oder gar nicht gezogen werden. Dies führt dazu, dass sich die Partikel in dem Bereich des Parameterraums häufen, wo bereits gute Parameter vorgefunden worden sind.

Das zufällige Ziehen mit Zurücklegen wird abweichend von der im Condensation-Algorithmus vorgeschlagenen Strategie durchgeführt, da diese Nachteile besitzt, die im Folgenden erläutert werden.

Nachteile der ursprünglichen Partikelwahl. Bei dem Resampling-Schritt, wie er ursprünglich in [IB98] beschrieben wird, wird zunächst für jedes Partikel $(\delta_{t-1}^i, \pi_{t-1}^i)$ ein kumulatives Gewicht g_{t-1}^i berechnet, für das gilt

$$g_{t-1}^i = \sum_{m=0}^i \pi_{t-1}^m. \quad (3.8)$$

Anschließend wird für jede Ziehung eine Zufallszahl r aus dem Intervall $[0, 1]$ berechnet. Da die kumulativen Gewichte bzgl. des Index i aufsteigend sortiert sind, kann eine binäre Suche durchgeführt werden, um das Partikel s_{t-1}^j mit dem kleinsten Index j auszuwählen, so dass für dessen kumulatives Gewicht $g_{t-1}^j \geq r$ gilt. Die so gezogenen n Partikel sind das Resultat des Resamplings.

Dieses Verfahren hat zwei Nachteile. Zum einen muss nach der Berechnung der kumulativen Gewichte für jedes der n zu ziehenden Partikel eine Zufallszahl berechnet und anschließend eine binäre Suche auf den kumulativen Gewichten durchgeführt werden. Dies stellt einen hohen Rechenaufwand dar, der sich im schlimmsten Fall mit $O(n \log n)$ abschätzen lässt. Zum anderen besteht die Gefahr eines unangemessenen Schwundes ("depletion") der Partikel, der sich am folgenden Beispiel verdeutlichen lässt. Es sei angenommen, dass vor dem Resampling eine uniforme Verteilung der Wahrscheinlichkeit vorliegt (Gleichverteilung), die Partikel eine gleichmäßige Abtastung im Parameterraum darstellen und für alle Partikelgewichte $\pi_{t-1}^i = \frac{1}{n}$ gilt. Dadurch,

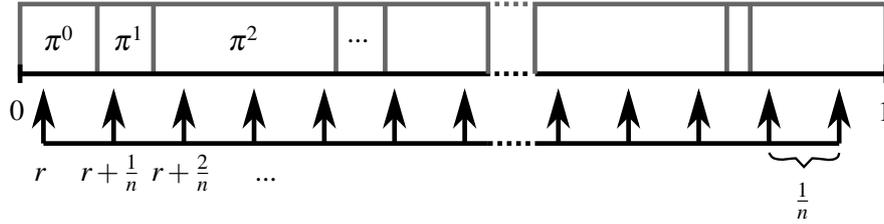


Abbildung 3.7: Kerinidee des "low variance sampling"-Algorithmus. Die Partikelgewichte π^i definieren die Größe der Teilintervalle im Gesamtintervall $[0, 1]$. Die Pfeile deuten die regelmäßige Abtastung im Abstand von $\frac{1}{n}$ an. Der Versatz r ("offset") muss nur einmal zufällig aus dem Intervall $[0, \frac{1}{n}]$ bestimmt werden. Es wird jeweils das Partikel gewählt, auf dessen Intervall ein Pfeil zeigt, so dass Partikel mit großen Gewichten öfter ausgewählt werden.

dass n Zufallszahlen generiert und die Partikel in Abhängigkeit von diesen ausgewählt werden, ist es möglich, dass einzelne Partikel häufiger gewählt werden. Aufgrund der konstanten Partikelanzahl hat dies im Umkehrschluss zur Folge, dass andere Partikel gar nicht gewählt werden. Aufgrund des Schwundes der Partikel in einigen Bereichen des Parameterraums und der Häufungen in anderen würden die Partikel nach dem Resampling nicht mehr die angenommene, uniforme Wahrscheinlichkeitsverteilung approximieren.

Effiziente Partikelwahl ohne Schwund. Aufgrund der beschriebenen Nachteile wird hier, abweichend zum ursprünglichen Vorgehen, für das Resampling das in [TBF05] vorgeschlagene Verfahren eingesetzt, da dieses den unangemessenen Partikelschwund berücksichtigt und zudem eine effizientere Partikelwahl mit dem Zeitaufwand von $O(n)$ anstatt $O(n \log n)$ im schlimmsten Fall ermöglicht.

Es sei angenommen, dass alle Partikelgewichte normiert sind und ihre Summe sich zu 1 aufaddiert. Stellt man sich die Partikelgewichte graphisch wie in Abb. 3.7 vor, so entsprechen diese jeweils einem Teilintervall des Gesamtintervalls $[0, 1]$. Es wird nur eine einzige Zufallszahl r aus dem Intervall $[0, \frac{1}{n}]$ berechnet, die einen zufälligen Versatz ("offset") beschreibt. Ausgehend von diesem Versatz wird das gesamte Intervall in regelmäßigen Abständen von $\frac{1}{n}$ abgetastet. Dabei wird jeweils das Partikel gewählt, in dessen Teilintervall die regelmäßige Abtastung fällt. Somit bleibt die Eigenschaft, dass ein Partikel proportional zur Gewichtung gewählt wird, erhalten. Dies liegt daran, dass die Intervalle von Partikeln mit einem großen Gewicht größer sind als von jenen mit einem kleinen Gewicht. Das Verfahren wird in [TBF05] als "low variance sampling" bezeichnet, weil das Ziehen der Partikel systematisch und somit mit einer deutlich kleineren Varianz durchgeführt werden kann, im Vergleich zu dem Fall, dass bei jeder Ziehung eine neue Zufallszahl generiert wird. Der erwähnte Geschwindigkeitsvorteil ergibt sich dadurch, dass zum einen die binäre Suche für jedes Partikel mit der im schlimmsten Fall nötigen Laufzeit $O(\log n)$ wegfallen kann, und zum anderen dadurch, dass nicht mehr n , sondern nur eine einzige Zufallszahl berechnet werden muss. Der Algorithmus 1 fasst das beschriebene Verfahren als Pseudocode zusammen und liefert für die gegebene Partikelmenge $\mathcal{S}_{t-1} = \{(\delta_{t-1}^i, \pi_{t-1}^i) | i = 0, \dots, n-1\}$ die neu abgetastete Menge $\hat{\mathcal{S}}_t$. Es muss jedoch beachtet werden, dass nach dem "Resampling"

die Partikelgewichte nicht mehr gültig sind und in den Schritten "Measurement" (Abschnitt 3.2.2.4) und "Normalization" (Abschnitt 3.2.2.5) neu berechnet werden müssen.

Algorithmus 1 : Resampling-Algorithmus (low variance sampling)

Input : Partikelmenge $\mathcal{S}_{t-1} = \{(\delta_{t-1}^i, \pi_{t-1}^i) | i = 0, \dots, n-1\}$ zum Zeitpunkt $t-1$

Output : Die Partikelmenge $\hat{\mathcal{S}}_t$ nach dem Resampling

begin

```

 $\hat{\mathcal{S}}_t := \emptyset$ 
 $i := 0$ 
 $c := \pi_{t-1}^0$ 
 $r := \text{rand}(0, \frac{1}{n})$ 
for  $j := 0$  to  $n-1$  do
     $u = r + j \cdot \frac{1}{n}$ 
    while  $u > c$  do
         $i := i + 1$ 
         $c := c + \pi_{t-1}^i$ 
     $\hat{\mathcal{S}}_t := \hat{\mathcal{S}}_t \cup (\delta_{t-1}^i, \pi_{t-1}^i)$ 
return  $\hat{\mathcal{S}}_t$ 

```

3.2.2.2 Drift

Der mit "Drift" bezeichnete Schritt beschreibt die in Abschnitt 3.2.1 erwähnte deterministische Bewegungsvorhersage. Dabei wurde ursprünglich das Bewegungsmodell des zu verfolgenden Objekts zum gegebenen Zeitschritt ausgewertet und der Zustandsvektor deterministisch manipuliert, um so die geschätzte Objektbewegung zu berücksichtigen. Da im gegebenen Kontext davon ausgegangen wird, dass das Linsensystem über die Zeit unverändert bleibt, ist eine Manipulation des Zustandsvektors, der die linsenrelevanten Parameter speichert, prinzipiell nicht notwendig. Eine Ausnahme bilden jedoch die Parameter des Verzerrungszentrums c_u und c_v , in dem Fall, dass die betrachtete Kamera nahezu keine Linsenverzerrung aufweist. Bei Annahme des Extremfalls, dass ein Linsensystem gänzlich ohne Linsenverzerrung vorliegt, würde für die optimalen radialen und tangentialen Parameter gelten, dass diese gleich Null sind. So würde aus der Gleichung (2.63) des Backward-Verzerrungsmodells folgen, dass die Parameter c_u und c_v beliebige Werte annehmen könnten, ohne das Ergebnis zu verändern. Damit läge bezüglich c_u und c_v eine uniforme Wahrscheinlichkeitsverteilung vor. Zusammen mit der zufallsbedingten Manipulation, wie sie im nächsten Abschnitt geschildert wird, hätte dies über die Iterationen hinweg zur Folge, dass das Verzerrungszentrum nicht in einer optimalen Position konvergieren, sondern fortwährend Zufallsbewegungen ("random walk") durchführen würde, was negative Auswirkungen auf die später beschriebene Konvergenz des Verfahrens hätte (siehe Abschnitt 3.2.3.2). Um diese Zufallsbewegung zu beschränken und somit zu verhindern, dass das Verzerrungszentrum an beliebigen Positionen angenommen wird, wird ein deterministisches

Bewegungsmodell eingeführt, welches sich ähnlich einem inversen Kraftfeld verhält, dessen Zentrum in der Bildmitte liegt. Somit wird bewirkt, dass eine deterministische Verschiebung des Verzerrungszentrums in Richtung der Bildmitte durchgeführt wird, die um so größer ausfällt, je weiter sich das Verzerrungszentrum von der Bildmitte entfernt. Dies ist legitim, da sich das Verzerrungszentrum typischerweise in der Umgebung der Bildmitte befindet. Liegen also sehr kleine radiale und tangentialen Verzerrungskoeffizienten vor, kann der Drift für c_u und c_v berücksichtigt werden durch

$$c_u := c_u - (c_u - w_{sensor}/2) \cdot \mathcal{D}(\|c_u - w_{sensor}/2\|), \quad (3.9)$$

$$c_v := c_v - (c_v - h_{sensor}/2) \cdot \mathcal{D}(\|c_v - h_{sensor}/2\|), \quad (3.10)$$

wobei die polynomielle Distanzfunktion der Form

$$\mathcal{D}(d) = \lambda_0 d^0 + \lambda_1 d^1 + \lambda_2 d^2 + \dots \quad (3.11)$$

gewählt und als Faktor verwendet wird.

3.2.2.3 Diffusion

Im Gegensatz zum vorherigen "Drift"-Schritt stellt der mit "Diffusion" bezeichnete Schritt den nichtdeterministischen Teil der ursprünglichen Bewegungsvorhersage dar. Durch das Resampling werden insbesondere Partikel mit einem großen Gewicht mehrfach gezogen, wodurch mehrere Partikel mit identischen Zustandsvektoren (Linsenverzerrungsparametern) vorliegen. Diese Häufung stellt eine schlechte Abdeckung des Parameterraums durch die verfügbare Partikelanzahl dar. Auch der deterministische "Drift"-Schritt ändert nichts an diesem Umstand, da er alle identischen Partikel auf die gleiche Weise manipulieren würde. Um diese Häufung zu verhindern, führt der "Diffusion"-Schritt eine von einem Zufallsprozess abhängige Manipulation aller Einträge des Zustandsvektors durch. Dieser Schritt verhindert jedoch nicht nur mehrfache identische Partikel, sondern sorgt auch für eine zufallsbedingte Exploration des Parameterraums in der lokalen Umgebung eines aktuell betrachteten Parametervektors. Auf diese Weise können lokale Minima potentiell überwunden und die Suche nach dem globalen Optimum fortgesetzt werden.

Die Änderungen an dem Zustandsvektor können formal beschrieben werden durch

$$\delta_t^i[j] := \delta_t^i[j] + \psi \mathcal{N}(0, \sigma_t^j), \quad (3.12)$$

wobei $\delta_t^i[j]$ den j -ten Eintrag des Zustandsvektors des i -ten Partikels zum Zeitpunkt t darstellt und ψ den Diffusion-Kontrollparameter. Um den Zufallsprozess zu modellieren, wird für den gegebenen Kontext die Funktion $\mathcal{N}(0, \sigma_t^j)$ verwendet, die eine Zufallszahl entsprechend der Normalverteilung mit dem Mittelwert Null und der Standardabweichung σ_t^j generiert. Dabei gilt

$$\sigma_t^j = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (\delta_t^i[j] - \overline{\delta_t[j]})^2} \quad (3.13)$$

und

$$\overline{\delta_t[j]} = \frac{1}{n} \sum_{i=0}^{n-1} \delta_t^i[j]. \quad (3.14)$$

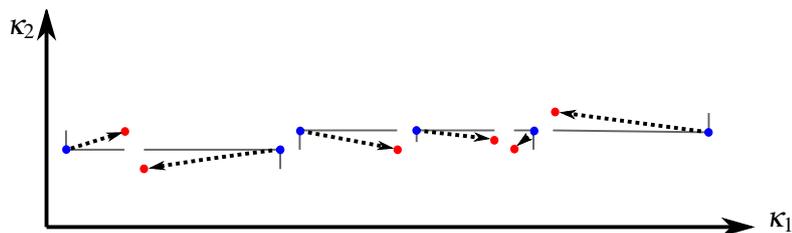


Abbildung 3.8: Verdeutlichung des "Diffusion"-Schritts am zweidimensionalen Beispiel mit den Parametern κ_1 und κ_2 . Die ursprünglichen und die veränderten Parameterwerte sind durch die blauen und roten Punkte dargestellt. Der "Diffusion"-Schritt führt eine vom Zufallsprozess bedingte, nichtdeterministische Änderung an den Parameterwerten durch, wobei das Ausmaß der Änderung von der Streuung der Partikel bzgl. des betrachteten Parameters abhängig ist. In dem Beispiel liegt für κ_1 eine größere Streuung als für κ_2 vor, weshalb die erwarteten Änderungen entlang der horizontalen Achse stärker ausfallen können.

Die jeweilige Standardabweichung σ_i^j wird individuell für jeden Eintrag des Zustandsvektors bestimmt und ist, über alle Partikel hinweg betrachtet, abhängig von der Streuung jedes einzelnen linsenrelevanten Parameters (z.B. κ_1). Die Parameter ψ und σ_i^j des Terms $\psi \mathcal{N}(0, \sigma_i^j)$ in Gleichung (3.12) regeln somit das Ausmaß oder die Schrittweite der nichtdeterministischen Änderung der einzelnen Entzerrungsparameter, wobei sie konzeptionell unterschiedlich sind, d.h. ψ ist ein Kontrollparameter und σ_i^j eine aktuelle stochastische Größe.

Die beschriebene Berechnung von σ_i^j hat folgende Eigenschaften: Konzentrieren sich die über alle n Partikel betrachteten Werte bzgl. eines Entzerrungsparameters (z.B. κ_2) auf eine einzige Region, so wird nach den Überlegungen der vorangegangenen Abschnitte angenommen, dass sich der optimale Wert in dieser Region befinden muss. In diesem Fall ist die Varianz des Parameters gering und somit auch die Wahrscheinlichkeit, sich durch die Diffusion weit von dieser Region zu entfernen. Ist hingegen eine solche Region noch nicht gefunden und die Partikel bzgl. eines Entzerrungsparameters (z.B. κ_1) nicht gehäuft, sondern weit gestreut, so ist die randomisierte Schrittweite größer und somit auch die Explorationsumgebung erweitert, die insbesondere bei der Überwindung lokaler Minima hilfreich ist. Die Abb. 3.8 verdeutlicht diesen Sachverhalt im zweidimensionalen Fall.

3.2.2.4 Measurement

Im ursprünglichen Condensation-Algorithmus werden im "Measurement"-Schritt die in den Partikeln gespeicherten Hypothesen anhand des aktuell beobachtbaren Kamerabildes, welches das zu verfolgende Objekt beinhaltet, überprüft. Somit bildet der "Measurement"-Schritt abstrakt betrachtet die Schnittstelle zwischen der approximierten Wahrscheinlichkeitsverteilung, die eine Schätzung des Systemzustands darstellt, und der mittels der Kamera verfügbaren Beobachtungen des Systems, die in jeder Iteration zur Aktualisierung und Anpassung der Approximation dienen sollen. Das Ziel des Schritts ist konkret die Messung bzw. Beurteilung der Qualität der durch die Par-

tikel geschätzten Position und Lage des Objekts anhand von Übereinstimmungen mit Bildmerkmalen wie z.B. starken Gradienten, die auf Objektkanten im Bild hindeuten. Im betrachteten Kontext soll die Qualität der in den Partikeln gespeicherten Entzerrungsparameter anhand ihrer Entzerrungsleistung bemessen und in den Partikelgewichten hinterlegt werden. Da der "Resampling"-Schritt (siehe Abschnitt 3.2.2.1) bevorzugt Partikel mit einem großen Gewicht wählt, sollten jene Partikel (δ_i^i, π_i^i) , deren Parametervektor δ_i^i eine gute Entzerrungsleistung liefert, ein großes Partikelgewicht π_i^i erhalten und umgekehrt. Zur Beurteilung der Entzerrungsleistung eines Parametervektors δ_i^i wird der verbliebene Grad an Verzerrung nach einer Korrektur unter Verwendung des Parametervektors δ_i^i herangezogen, der nachfolgend auch verbliebener Verzerrungsfehler genannt wird. Anstatt des zu verfolgenden Objekts muss im betrachteten Kontext ein verzerrtes Kamerabild \mathbf{B} mit dem beschriebenen Kalibrierobjekt (siehe Abb. 3.3(b)) vorhanden sein. Mit Hilfe der Funktion \mathcal{F}_{ExtKP} kann im Vorfeld die Menge der verzerrten Kreuzungspunkte $\hat{\mathcal{M}}_{KP} = \{\hat{\mathbf{p}}_{a,b}\} = \mathcal{F}_{ExtKP}(\mathbf{B})$ des Kalibrierobjekts aus \mathbf{B} extrahiert werden (siehe Gleichung (3.4)). Sei weiterhin die Funktion $\mathcal{F}_{UndistKP}$, die für die Menge der verzerrten Kreuzungspunkte $\hat{\mathcal{M}}_{KP}$ die Menge der entzerrten Kreuzungspunkte \mathcal{M}_{KP} unter Verwendung des Parametervektors δ berechnet, gegeben durch

$$\begin{aligned} \mathcal{M}_{KP} &= \mathcal{F}_{UndistKP}(\hat{\mathcal{M}}_{KP}, \delta) \\ &= \{\mathbf{p}_{a,b} | \mathbf{p}_{a,b} = f_U(\hat{\mathbf{p}}_{a,b}, \delta), a = 0, \dots, k-1, b = 0, \dots, l-1\}. \end{aligned} \quad (3.15)$$

Dabei entsprechen k und l der Anzahl der Zeilen und Spalten des Kreuzungspunktgitters (siehe Abschnitt 3.1.2.1) und der in Gleichung (2.67) definierten Entzerrungsfunktion f_U .

Der "Measurement"-Schritt kann dann in die folgenden Subschritte eingeteilt werden, die für jedes Partikel (δ_i^i, π_i^i) , $i = 0, \dots, n-1$, durchgeführt werden, um die jeweiligen Gewichte π_i^i zu bestimmen:

- Berechnung der Menge der entzerrten Kreuzungspunkte \mathcal{M}_{KP} aus den im Vorfeld extrahierten verzerrten Kreuzungspunkten $\hat{\mathcal{M}}_{KP}$ unter Verwendung des Parametervektors δ_i^i durch

$$\mathcal{M}_{KP} = \mathcal{F}_{UndistKP}(\hat{\mathcal{M}}_{KP}, \delta_i^i). \quad (3.16)$$

- Berechnung des verbliebenen Verzerrungsfehlers \mathbf{v} auf der korrigierten Kreuzungspunktmenge \mathcal{M}_{KP} mit der in Abschnitt 3.1.2.1 definierten Zielfunktion (Fehlerfunktion) \mathcal{E}_{KP} :

$$\mathbf{v} = \mathcal{E}_{KP}(\mathcal{M}_{KP}). \quad (3.17)$$

- Festsetzung des Partikelgewichts π_i^i auf den Wert

$$\pi_i^i = \frac{1}{\mathbf{v}}, \quad (3.18)$$

da der verbliebene Verzerrungsfehler in einem reziproken Verhältnis zur Güte des verwendeten Parametervektors steht.

3.2.2.4.1 Nutzung mehrerer Kalibrierbilder. In den bisherigen Betrachtungen wurde nur ein Bild berücksichtigt. Um jedoch eine größere Anzahl von Kreuzungspunkten zu erlauben, ohne ein anderes Kalibrierobjekt mit mehr Schachbrettkästchen verwenden zu müssen, soll im Folgenden der verbliebene Verzerrungsfehler auch für mehrere Bilder definiert werden. Sei $\mathcal{B} = \{\mathbf{B}_0, \dots, \mathbf{B}_{m-1}\}$ die Menge von $m \geq 1$ Bildern. Dann soll die Funktion, die den verbliebenen Verzerrungsfehler (Residuum) für \mathcal{B} unter Verwendung des Entzerrungsparametervektors δ bestimmt, gegeben sein durch

$$\mathcal{E}_{Residuum}(\mathcal{B}, \delta) = \frac{1}{m} \sum_{j=0}^{m-1} \mathcal{E}_{KP}(\mathcal{F}_{UndistKP}(\mathcal{F}_{ExtKP}(\mathbf{B}_j), \delta)). \quad (3.19)$$

Dann können für jedes Partikel $(\delta_t^i, \pi_t^i), i = 0, \dots, n-1$, die Partikelgewichte bei Verwendung mehrerer Bilder $\mathbf{B}_0, \dots, \mathbf{B}_{m-1}$ berechnet werden durch

$$\pi_t^i = \frac{1}{\mathcal{E}_{Residuum}(\mathcal{B}, \delta_t^i)}. \quad (3.20)$$

Dabei können, wie im ursprünglich beschriebenen Vorgehen, die Kreuzungspunkte der einzelnen Bilder einmal im Vorfeld berechnet werden, um die wiederholte Berechnung bei jeder Iteration zu vermeiden.

3.2.2.4.2 Live-Kalibrierung unter Verwendung eines Videostroms. In manchen Fällen kann es bequemer sein, nicht erst eine Menge von Kalibrierbildern vorauszuwählen und zu speichern, sondern direkt die Live-Videoansicht der Kamera (Live-Stream), in der das Kalibrierobjekt enthalten ist, zu nutzen. Da der Condensation-Algorithmus ursprünglich so konzipiert ist, dass er die Objektverfolgung, also die Schätzung des Zustandsvektors in Echtzeit, auf fortlaufenden Videobildern durchführen kann, ist es möglich diese Eigenschaft auch in dem hier vorliegenden Kontext zu nutzen.

Damit in jeder Partikelfilter-Iteration t das aktuelle Live-Bild \mathbf{B}_{Live} berücksichtigt werden kann, muss lediglich der "Measurement"-Schritt dahingehend angepasst werden, dass in Formel (3.20) die Menge der Bilder \mathcal{B} bei jeder Iteration aus der einelementigen Bildmenge besteht, die nur \mathbf{B}_{Live} enthält, so dass das Partikelgewicht gegeben ist durch

$$\pi_t^i = \frac{1}{\mathcal{E}_{Residuum}(\{\mathbf{B}_{Live}\}, \delta_t^i)}. \quad (3.21)$$

Im Gegensatz zum Abschnitt 3.2.2.4.1 können die extrahierten Kreuzungspunkte nicht im Vorfeld berechnet, sondern müssen für jedes Live-Bild neu extrahiert werden.

3.2.2.5 Normalization

Nach der Definition der Wahrscheinlichkeitsverteilung gilt, dass das Integral unter der dazugehörigen Dichtefunktion auf 1 normiert ist [Geo07]. Da die Partikelgewichte auch mit Wahrscheinlichkeiten assoziiert werden (z.B. beim "Resampling", bei dem

das Partikelgewicht die Wahrscheinlichkeit beschreibt, mit der ein Partikel gezogen wird), soll für diese ebenfalls gelten, dass sie sich zu 1 aufsummieren. Da diese Eigenschaft jedoch durch den "Measurement"-Schritt nicht gewährleistet ist, wird im "Normalization"-Schritt für alle Partikel $(\delta_t^i, \pi_t^i), i = 0, \dots, n - 1$, die folgende Neuberechnung der Partikelgewichte π_t^i durchgeführt:

$$\pi_t^i := \frac{\pi_t^i}{\sum_{i=0}^{n-1} \pi_t^i}. \quad (3.22)$$

Nach dieser Normierung ist ein kompletter Iterationsschritt des Partikelfilters abgeschlossen und die Partikelmenge \mathcal{S}_t ist vorbereitet für den "Resampling"-Schritt zum Zeitpunkt $t + 1$.

3.2.2.6 Fusion der Hypothesen

Am Ende jedes kompletten Iterationsschrittes bietet jedes der n Partikel (δ_t^i, π_t^i) eine Hypothese δ_t^i mit der Wahrscheinlichkeit π_t^i über den geschätzten optimalen Entzerrungsparametervektor zum Zeitpunkt t . Um den optimalen Parametervektor zu bestimmen, der dem Erwartungswert der Partikelmenge entspricht, kann aufgrund der Normierung der Gewichte π_t^i der *gewichtete mittlere Entzerrungsparametervektor* δ_t^{GM} bestimmt werden durch

$$\delta_t^{GM} = \sum_{i=0}^{n-1} \delta_t^i \pi_t^i. \quad (3.23)$$

Somit entspricht δ_t^{GM} der Fusion aller Hypothesen zum gegebenen Zeitpunkt t .

Aufgrund der relativ einfachen Struktur sind auch weitere Kandidaten für den optimalen Parametervektor zum Zeitpunkt t denkbar, wie z.B. der Parametervektor des Partikels (δ_t^j, π_t^j) mit dem *maximalen* Gewicht

$$\delta_t^{MAX} = \delta_t^j \text{ mit } \pi_t^j \geq \pi_t^i \forall i = 0, \dots, n - 1, \quad (3.24)$$

oder der Parametervektor δ_t^{TT} , der das gewichtete Mittel aus den zehn wahrscheinlichsten Parametervektoren darstellt (Top Ten). Jedoch weist, über die Iterationen hinweg betrachtet, der gewichtete mittlere Parametervektor δ_t^{GM} die stabilsten Eigenschaften und somit die geringste Sprunghaftigkeit auf.

3.2.3 Iterative Optimierung der Linsenentzerrungsparameter

In Abschnitt 3.2.2 wurde detailliert beschrieben, wie ein vollständiger Iterationsschritt des Partikelfilters aufgebaut ist und wie er für den vorliegenden Kontext angepasst wurde. Nach der Erläuterung der Initialisierung des Partikelfilters werden im Folgenden der gesamte iterative Optimierungsprozess der Linsenentzerrungsparameter sowie geeignete Abbruchkriterien erläutert.

3.2.3.1 Initialisierung der Partikel

Vor der ersten Iteration des Partikelfilters müssen die Parametervektoren δ_0^i und Gewichtungsfaktoren π_0^i der n Partikel für den Zeitpunkt $t = 0$ initialisiert werden. Bezüglich der Partikelwahrscheinlichkeiten wird der Einfachheit halber eine Gleichverteilung angenommen, so dass für alle Gewichte

$$\pi_0^i = \frac{1}{n}, \quad i = 0, \dots, n-1, \quad (3.25)$$

gelten soll. Je nach Art des repräsentierten Parameters, also $\kappa_1, \kappa_2, \kappa_3, \rho_1, \rho_2, c_u$ oder c_v , der von der j -ten Komponente des Parametervektors repräsentiert wird, muss der initiale Wert passend gewählt werden. Wie bereits in Abschnitt 2.4 verwendet, könnte für jedes Partikel ($i = 0, \dots, n-1$) für die ersten fünf Parametervektorkomponenten (radiale und tangentiale Koeffizienten) die Initialisierung

$$\delta_0^i[j] = 0, \quad j = 0, \dots, 4, \quad (3.26)$$

sowie für das Verzerrungszentrum

$$\delta_0^i[5] = w_{sensor}/2, \quad (3.27)$$

$$\delta_0^i[6] = h_{sensor}/2, \quad (3.28)$$

mit den Pixelkoordinaten des Sensormittelpunkts in Frage kommen. Diese Initialisierung ist jedoch aus folgenden Gründen ungeeignet. Im "Diffusion"-Schritt (siehe Abschnitt 3.2.2.3) ist die nichtdeterministische Manipulation jeder Parametervektorkomponente j abhängig von der Streuung der Werte bzgl. dieser Komponente (über alle Partikel hinweg betrachtet). Bei der vorgeschlagenen Initialisierung sind jedoch alle n Parametervektoren identisch. Folglich ist die über alle Parametervektoren betrachtete Standardabweichung bzgl. einer Komponente j gleich Null und der "Diffusion"-Schritt hätte keinerlei Auswirkung auf den Parametervektor, womit die Partikelmenge über alle Iterationsschritte hinweg unverändert bleiben würde.

Durch die Integration eines Zufallsprozesses in die Initialisierungsphase kann dies verhindert werden. Dazu wird die aus Abschnitt 3.2.2.3 bekannte Funktion \mathcal{N} verwendet, um normalverteilte Zufallszahlen zu generieren. Somit wird für die Initialisierung der Parameter des Verzerrungszentrums

$$\delta_0^i[5] = \mathcal{N}\left(\frac{w_{sensor}}{2}, \frac{w_{sensor}}{100}\right), \quad (3.29)$$

$$\delta_0^i[6] = \mathcal{N}\left(\frac{h_{sensor}}{2}, \frac{h_{sensor}}{100}\right), \quad (3.30)$$

für alle $i = 0, \dots, n-1$ gewählt, wobei der Mittelwert der normalverteilten Zufallszahlen jeweils vom Sensormittelpunkt abhängt und die Standardabweichung 1% der Sensorbreite bzw. -höhe, gegeben in Pixeln, beträgt.

Für die radialen und tangentialen Koeffizienten in den Komponenten $j = 0, \dots, 4$ erfolgt die Initialisierung für alle $i = 0, \dots, n-1$ durch

$$\delta_0^i[j] = \mathcal{N}(0, \sigma). \quad (3.31)$$

Während die Wahl der Standardabweichung für das Verzerrungszentrum relativ unkritisch ist, ist eine geeignete Wahl von σ deutlich relevanter für den Optimierungsprozess und wird experimentell zusammen mit weiteren für den Partikelfilter relevanten Parametern im Abschnitt 3.2.4.3 untersucht. Es sei angemerkt, dass potenziell durch eine ungünstige Streuung keine Samples in der Nähe des globales Minimums gesetzt werden könnten, was die Gefahr birgt, ein globales Minimum zu übersehen. Jedoch konnte die Wirksamkeit des Verfahrens experimentell nachgewiesen werden, wie später im Evaluierungskapitel deutlich wird.

Auf der wie beschrieben initialisierten Partikelmenge $\mathcal{S}_0 = \{(\delta_0^i, \pi_0^i) | i = 0, \dots, n-1\}$ kann der iterative stochastische Optimierungsprozess gestartet werden.

3.2.3.2 Optimierungprozess und Abbruchkriterien

Ist die Initialisierung durchgeführt und eine Menge von $m \geq 1$ verzerrten Bildern $\mathcal{B} = \{\mathbf{B}_0, \dots, \mathbf{B}_{m-1}\}$ mit dem Kalibrierobjekt gegeben, so kann der iterative Optimierungsprozesses gestartet werden. Dieser hat zum Ziel, die optimalen Entzerrungsparameter zu finden, die zu der Linse passen, mit der die Bilder in \mathcal{B} erstellt worden sind. Während des Prozesses wird erwartet, dass "gute" Partikel bestehen bleiben und "schlechte" über die Zeit hinweg verschwinden, so dass sich die Partikelmenge in der Umgebung des Parameterraums konzentriert, in der sich der gesuchte optimale Parametervektor befindet. Somit stellt der in Abschnitt 3.2.2.6 eingeführte gewichtete, mittlere Entzerrungsparametervektor δ_t^{GM} , der der Fusion aller Hypothesen entspricht, die zum Zeitpunkt t optimalste Lösung dar. Angefangen bei $t = 0$ wird erwartet, dass δ_t^{GM} über die Iterationsschritte $t = 0, 1, \dots$ immer bessere Linsenentzerrungsparameter zur Verfügung stellt.

Für den Abbruch des iterativen Optimierungsprozesses sind mehrere Kriterien denkbar. Der Prozess kann z.B. beendet werden, sobald die Veränderungen von δ_t^{GM} vom vorherigen zum aktuellen Zeitschritt kleiner als ein vorgegebener Schwellenwert $\tau > 0$ ist:

$$\|\delta_t^{GM} - \delta_{t-1}^{GM}\| < \tau. \quad (3.32)$$

Ein alternatives Terminierungskriterium kann mit Hilfe der in Gleichung (3.19) definierten Funktion $\mathcal{E}_{Residuum}(\mathcal{B}, \delta)$ beschrieben werden. Diese Funktion berechnet für die gegebene Bildmenge \mathcal{B} den verbliebenen Verzerrungsfehler nach Anwendung der Entzerrung mittels des gegebenen Parametervektors δ . Das Abbruchkriterium lautet dann

$$|\mathcal{E}_{Residuum}(\mathcal{B}, \delta_t^{GM}) - \mathcal{E}_{Residuum}(\mathcal{B}, \delta_{t-1}^{GM})| < \tau. \quad (3.33)$$

Wird angenommen, dass δ_t^{GM} über die Iterationen hinweg "besser" wird, so kann geschlossen werden, dass $\mathcal{E}_{Residuum}$ immer kleinere Werte liefert und schließlich konvergiert, so dass die Änderung zwischen zwei Zeitschritten kleiner als τ ist.

Die individuelle Berechnung der Standardabweichung σ_t^j für den j -ten Parametervektoreintrag im "Diffusion"-Schritt in Gleichung (3.13) spielt eine wichtige Rolle für die Terminierungseigenschaft des Optimierungsprozesses. Aufgrund der Häufung in

der Nähe des Optimums folgt, dass σ_t^j und somit auch die randomisierte Schrittweite kleiner wird, so dass sich die Partikel im darauf folgenden Schritt noch weiter häufen. Dies resultiert wiederum in einem noch kleineren σ_{t+1}^j , wodurch letztendlich eine Konvergenz erwartet wird.

Trotz der beschriebenen Eigenschaft ist zu beobachten, dass die auf dem Zufallsprozess basierenden Änderungen der Zustandsvektoren im "Diffusion"-Schritt dafür sorgen, dass δ_t^{GM} in der Nähe der Optimums fluktuiert. Dies gilt folglich auch für den verbliebenen Verzerrungsfehler $\mathcal{E}_{Residuum}(\mathcal{B}, \delta_t^{GM})$, was die Wahl eines geeigneten Schwellenwerts τ erschweren kann. Die Wahl des Diffusion-Kontrollparameters ψ in Gleichung (3.12) hat einen wichtigen Einfluss auf das Fluktuationsverhalten, wie später im Abschnitt 3.2.4.2 verdeutlicht wird. Ein geeignet gewählter Wert für ψ kann die Fluktuation minimieren, jedoch nicht gänzlich verhindern, wie in Abb. 3.12 zu sehen ist. Als Lösung wird der *beste gewichtete mittlere Entzerrungsparametervektor* δ_t^{BGM} vorgeschlagen, der am Ende eines Iterationsschritts nur dann aktualisiert wird, wenn der aktuelle *gewichtete mittlere Entzerrungsparametervektor* δ_t^{GM} einen geringeren verbliebenen Verzerrungsfehler aufweist:

$$\delta_t^{BGM} := \begin{cases} \delta_t^{GM} & \mathcal{E}_{Residuum}(\mathcal{B}, \delta_t^{GM}) < \mathcal{E}_{Residuum}(\mathcal{B}, \delta_{t-1}^{BGM}) \\ \delta_{t-1}^{BGM} & \text{sonst.} \end{cases} \quad (3.34)$$

Somit kann garantiert werden, dass die Funktion $\mathcal{E}_{Residuum}(\mathcal{B}, \delta_t^{BGM})$ monoton fallend ist. Wird im Verlauf des Optimierungsprozesses eine größere Anzahl an Iterationen als ein vorgegebener Schwellenwert τ_{iter} durchgeführt, ohne dass sich der Wert von $\mathcal{E}_{Residuum}(\mathcal{B}, \delta_t^{BGM})$ verändert, kann der Optimierungsprozess als abgeschlossen betrachtet und abgebrochen werden.

3.2.4 Konfiguration relevanter Parameter des Partikelfilters

Die Konfiguration des Partikelfilters hat einen wesentlichen Einfluss auf den Optimierungsprozess. Die Anzahl der Iterationen und die nötige Laufzeit bis zur Terminierung hängen z.B. von n , der Anzahl der verwendeten Partikel, dem Diffusion-Kontrollparameter ψ , der Standardabweichung σ der normalverteilten initialen Werte der Parametervektoren δ_0^i sowie der Wahl der Distanzfunktion beim Drift-Schritt ab. Die optimale Konfiguration dieser Parameter wurde experimentell ermittelt. Für alle Konfigurationstests wurde die gleiche Eingabe-Bildmenge aus insgesamt sechs Bildern verwendet, die ein Kalibrierobjekt in unterschiedlichen Posen enthalten, welches $9 \times 6 = 54$ innere Kreuzungspunkte zur Verfügung stellt. Alle Bilder wurden mit derselben USB-Kamera (Hersteller: ImagingSource, Modell: DFK31BU03) mit einer Auflösung von 1024×768 Pixeln unter Verwendung eines Zoom-Objektivs mit variabler Brennweite (Hersteller: Computar, 3.5-10.5 mm), das auf 4 mm Brennweite fixiert wurde, erstellt. Die Bilder sind in der oberen Reihe in Abb. 3.9 dargestellt. Alle Tests sind auf einem Windows-XP Notebook mit einem Intel Mobile Core i5-520M Prozessor mit 2.4 GHz und 3 GB RAM durchgeführt worden.

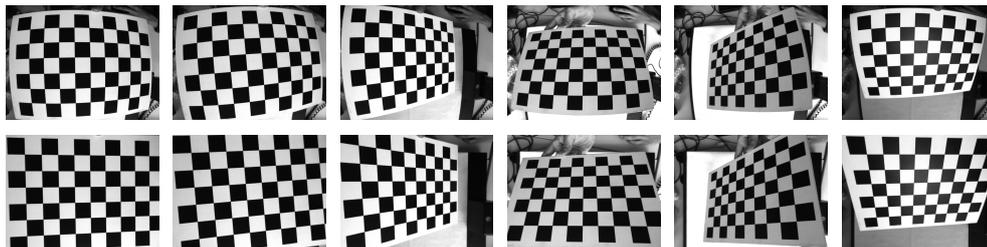


Abbildung 3.9: Die obere Reihe zeigt die sechs Bilder, die unter anderem für die Ermittlung der für die Konfiguration des Partikelfilters relevanten Parameter verwendet worden sind. Die Bilder weisen eine deutliche Linsenverzerrung auf. Die untere Reihe zeigt die jeweiligen Bilder nach der Entzerrung unter Anwendung des beschriebenen Verfahrens in Abschnitt 3.2.6.

3.2.4.1 Anzahl der Partikel

Zunächst soll der Einfluss der Partikelanzahl untersucht werden. Die Abb. 3.10 zeigt die Entwicklung des verbliebenen Verzerrungsfehlers $\mathcal{E}_{Residuum}(\mathcal{B}, \delta_t^{GM})$ im Verlauf der ersten 30 Iterationsschritte des Verfahrens, das auf den sechs verzerrten Bildern angewendet worden ist, die in der oberen Reihe der Abb. 3.9 zu sehen sind. Dabei wurde jeweils eine unterschiedliche Anzahl von Partikeln zwischen 10 und 10000 untersucht. Es sei angemerkt, dass der Optimierungsprozess nach den ersten 30 Iterationen noch nicht abgeschlossen war. Weiterhin sei darauf hingewiesen, dass die Werte des verbliebenen Verzerrungsfehlers $\mathcal{E}_{Residuum}$ in diesem sowie in den nachfolgenden Abschnitten mit dem Faktor 1000 skaliert worden sind, um in einem angenehm lesbaren Wertebereich zu liegen.

Während die Verwendung einer großen Partikelanzahl (200 bis 10000) in einer schnellen und monotonen Senkung des Verzerrungsfehlers resultierte, wies die Nutzung weniger Partikel (10 bis 100) Verläufe auf, bei denen der Verzerrungsfehler über die Iterationen hinweg deutlich langsamer gesunken ist. In dem Fall der Verwendung von nur 10 Partikeln waren deutliche Schwankungen des Verzerrungsfehlers zu erkennen, so dass der Werteverlauf bereits innerhalb der ersten 30 Iterationsschritte nicht-monoton war. Die Verläufe der Verzerrungsfehler für $n \geq 200$ Partikel verhielten sich relativ ähnlich. Der verbliebene Verzerrungsfehler am Ende der 30 Iterationen sowie die Anzahl der berechenbaren Iterationen pro Sekunde sind in Tabelle 3.1 zusammengefasst. Dort ist zu erkennen, dass mit steigender Partikelanzahl der Verzerrungsfehler tendenziell kleiner wird. Bei der Verwendung von $n \geq 200$ Partikeln konnten nach der 30. Iteration keine signifikanten Unterschiede des verbliebenen Verzerrungsfehlers zwischen den Messreihen festgestellt werden, während die Anzahl der pro Sekunde berechenbaren Iterationen annähernd linear mit steigender Partikelanzahl fiel.

Um die optimale Partikelanzahl zu ermitteln, ist ein weiterer Test durchgeführt worden, der die reale Laufzeit des Algorithmus berücksichtigte. Es wurde die Laufzeit gemessen, die der Algorithmus benötigte, um eine bestimmte Güte der Entzerrung zu erreichen, also einen bestimmten Schwellenwert τ des verbliebenen Verzerrungsfehlers zu unterschreiten.

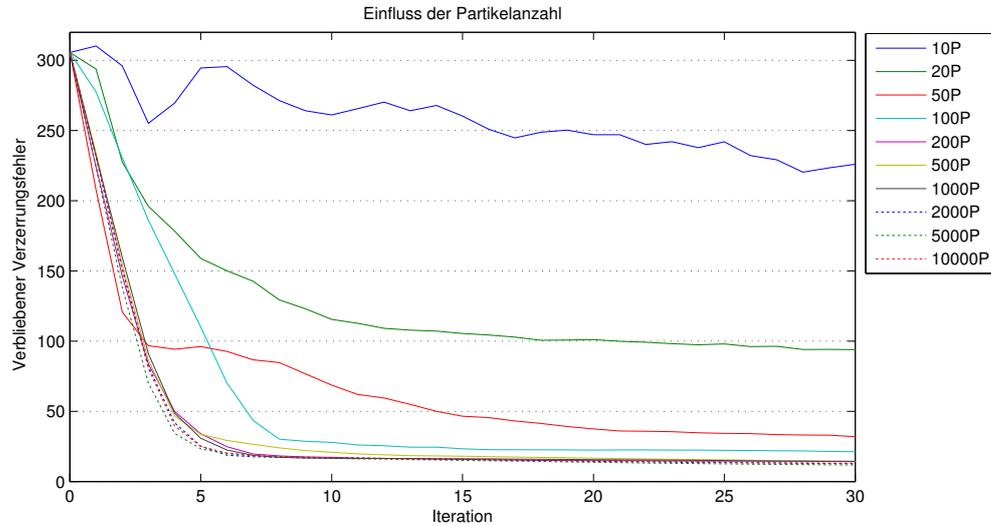


Abbildung 3.10: Vergleich der Verläufe des mit Faktor 1000 skalierten verbliebenen Verzerrungsfehlers $\mathcal{E}_{Residuum}(\mathcal{B}, \delta_t^{GM})$ innerhalb der ersten 30 Iterationen unter Variation der verwendeten Partikelanzahl zwischen 10 und 10000.

Partikelanzahl	Fehler	Iter./Sek.	Partikelanzahl	Fehler	Iter./Sek.
10	225.9	1572.1	500	14.2	93.9
20	94.0	1166.0	1000	14.1	48.0
50	31.9	667.5	2000	12.7	24.3
100	21.2	401.5	5000	11.8	9.5
200	14.4	220.7	10000	12.9	4.7

Tabelle 3.1: Angaben des mit Faktor 1000 skalierten verbliebenen Verzerrungsfehlers nach Abarbeitung der ersten 30 Iterationen des Optimierungsprozesses sowie der Berechnungsgeschwindigkeit (Anzahl der Iterationen pro Sekunde) bei variierender Partikelanzahl.

Um τ zu ermitteln, wurde zunächst auf denselben sechs Bildern wie in den bisherigen Tests eine Kamerakalibrierung unter Verwendung der OpenCV-Implementierung durchgeführt. Die Implementierung entspricht den Ausführungen im Abschnitt 2.4. Anschließend wurden ebenfalls mit entsprechend vorhandenen OpenCV-Methoden die optimierten Parameter verwendet, um die Bilder zu entzerren. Für die so ermittelte entzernte Bildmenge ist der Verzerrungsfehler entsprechend der Formel (3.6) ermittelt worden, der bei $9.78 \cdot 10^{-3}$ lag. Dieser Wert wurde als Schwellenwert τ für das Abbruchkriterium genutzt. Somit sollte der Optimierungsprozess erst dann abgebrochen werden, wenn die Entzerrungsleistung mindestens "so gut" ist wie die der OpenCV-Bibliothek. Die vollständige Partikelfilter basierte Optimierung wurde für eine variierende Anzahl von Partikeln durchgeführt. Um gültige Aussagen über den Optimierungsprozess, der von einem Zufallsprozess abhängt, machen zu können, sind die Tests für jede Partikelanzahl jeweils 100 mal wiederholt worden. Die Testergebnisse sind in Abb. 3.11 in Form eines Box-Whisker-Plots [Tuk77] aufgezeigt.

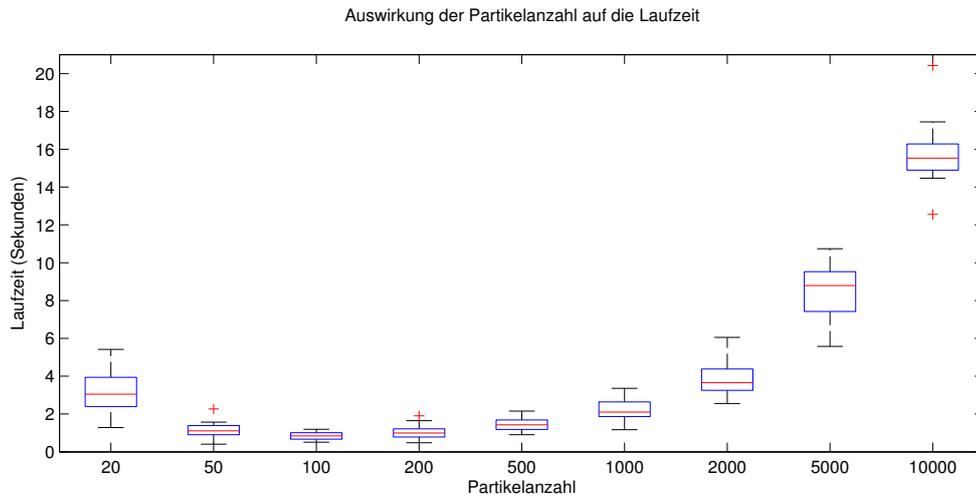


Abbildung 3.11: Visualisierung der Verteilung der Laufzeitergebnisse von jeweils 100 Testwiederholungen pro Partikelanzahl unter Verwendung des Box-Whisker-Plots [Tuk77]. Die Box deutet die Verteilung von 50% aller Ergebniswerte an. Der rote Balken zeigt den Median, der die Box in das obere und untere Quartil aufteilt. Die Whisker (Antennen) deuten den Bereich der Daten innerhalb des 1.5-fachen des Interquartilabstands an. Die roten Kreuze außerhalb des Whisker-Bereichs werden als Ausreißer interpretiert. Die besten Laufzeitresultate wurden bei Verwendung von 100 und 200 Partikeln erreicht.

Die besten Ergebnisse wurden für die Anzahl von 100 und 200 Partikeln ermittelt. In Kombination mit den in Tabelle 3.1 und Abb. 3.10 erarbeiteten Ergebnissen, wurde daher für alle nachfolgend beschriebenen Tests die Partikelanzahl auf 200 festgelegt. Um eine mit der OpenCV-Bibliothek vergleichbare Entzerrungsleistung zu erreichen, lag die mittlere Laufzeit (bei 200 Partikeln) bei 1.039 Sekunden.

3.2.4.2 Diffusion-Kontrollparameter

Der Kontrollparameter ψ aus Gleichung (3.12) steuert das Ausmaß der nichtdeterministischen Manipulation des Entzerrungsparametervektors im "Diffusion"-Schritt. Dadurch hat er auf der einen Seite Einfluss auf die Veränderung von einem Iterationsschritt zum nächsten und folglich auch auf die Anzahl notwendiger Gesamtschritte. Auf der anderen Seite spielt er eine wichtige Rolle bei der Verwendung des in Gleichung (3.33) definierten Terminierungskriteriums und der geeigneten Wahl von τ , denn größer gewählte Werte für ψ ermöglichen größere Veränderungen im "Diffusion"-Schritt, haben jedoch auch eine stärkere Fluktuation (Schwankung) in der Nähe des Optimums zur Folge.

Im ersten Test wurden 1000 Iterationen durchgeführt und die Schwankungen in Abhängigkeit von ψ analysiert. Es sei angemerkt, dass 1000 Optimierungsschritte (bei Verwendung von 200 Partikeln) bereits eine sehr große Iterationsanzahl darstellt, bei der alle signifikanten Änderungen des Optimierungsprozesses (des mittleren gewich-

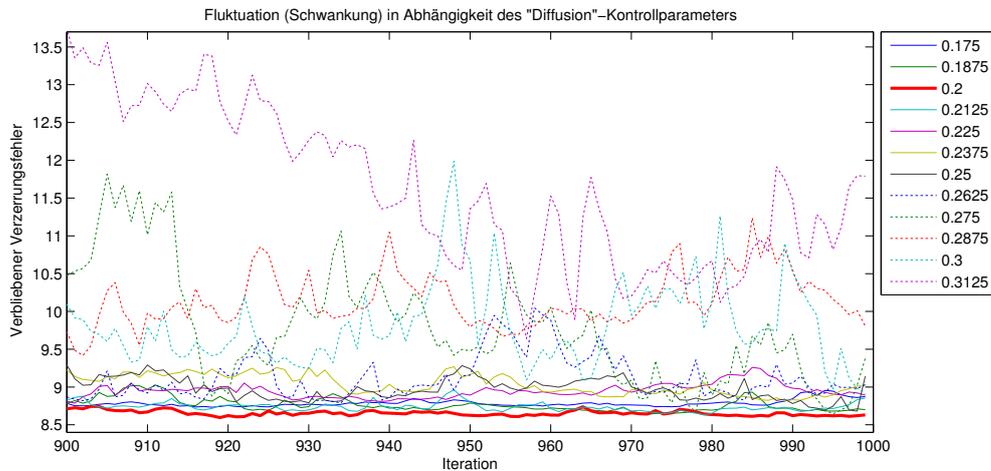


Abbildung 3.12: Fluktuation des mit Faktor 1000 skalierten verbliebenen Verzerrungsfehlers in den Iterationen 900 bis 1000 für unterschiedliche Werte des Kontrollparameters ψ . Die geringsten Schwankungen konnten bei Verwendung von $\psi = 0.2$ beobachtet werden.

teten Entzerrungsparametervektors δ_t^{GM}) als abgeschlossen angenommen werden können. Abb. 3.12 zeigt den Verlauf des verbliebenen Verzerrungsfehlers für unterschiedliche Werte von ψ zwischen 0.175 und 0.3125 für die Iterationsschritte 900 bis 1000. Die besten Resultate wurden für $\psi = 0.2$ erreicht. Insbesondere größere Werte wiesen eine deutlich stärkere Fluktuation des verbliebenen Verzerrungsfehlers auf.

Der zweite Test analysiert den Einfluss von ψ auf die Anzahl der benötigten Iterationen des Algorithmus. Der Algorithmus terminierte, sobald die Anzahl der Iterationen ohne Veränderung des verbliebenen Verzerrungsfehlers unter Verwendung von δ_t^{BGM} , wie in Gleichung (3.34) definiert, einen vorgegebenen Iterationsschwellenwert τ_{iter} überschritten hat.

Die Abb. 3.13 zeigt den Verzerrungsfehler sowie die Anzahl der benötigten Iterationen bis zur Terminierung, unter Verwendung des relativ strengen Kriteriums $\tau_{iter} = 100$ für unterschiedliche Werte von ψ . Dabei wurden die Tests auch in diesem Fall 100 mal wiederholt. Die angegebenen Werte in Abb. 3.13 entsprechen den jeweiligen Mittelwerten der Untersuchungen. Während die Anzahl der benötigten Iterationsschritte mit steigenden Werten für ψ sank, nahm der verbliebene Verzerrungsfehler zu. Dieses Verhalten ist nachvollziehbar, da größere ψ größere Veränderungen (Schrittweiten) im "Diffusion"-Schritt nach sich ziehen und folglich eine frühere Terminierung bei dem gewählten Kriterium erreicht wird. Allerdings wird die Qualität des optimalen Parametervektors δ_t^{BGM} aufgrund der stärkeren Fluktuation und folglich geringeren Konzentration der Partikel in der Nähe des Optimums geringer, was für einen größeren verbliebenen Verzerrungsfehler sorgt. Somit stellt die Wahl von ψ einen Kompromiss zwischen Laufzeit und Qualität dar. Zusammen mit der Berücksichtigung der Fluktuation (siehe Abb. 3.12) soll der Qualität eine höhere Priorität eingeräumt werden, so dass im Folgenden $\psi = 0.2$ verwendet wird.

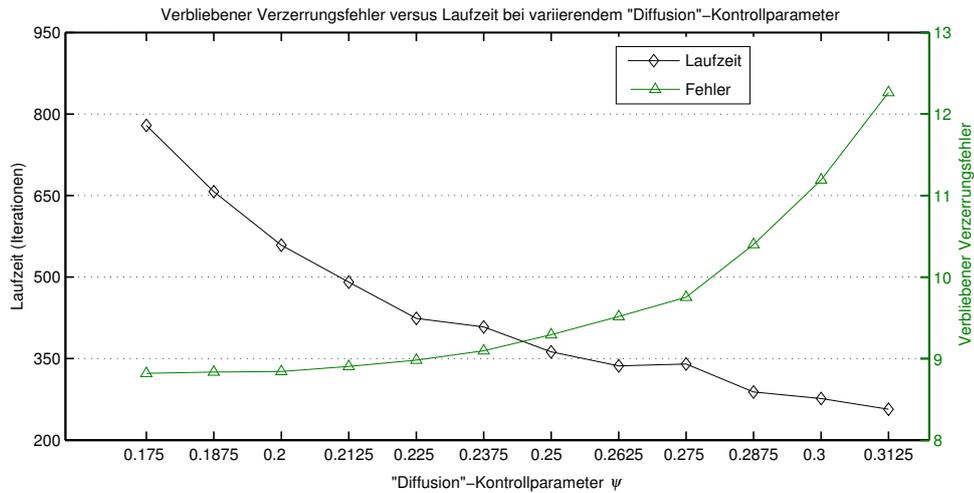


Abbildung 3.13: Mit Faktor 1000 skaliertes verbliebener Verzerrungsfehler (grün) und benötigte Laufzeit (schwarz) für unterschiedliche Werte des Kontrollparameters ψ . Der Optimierungsprozess galt als abgeschlossen, wenn $\tau_{iter} = 100$ Iterationsschritte unter Verwendung von δ_t^{BGM} (siehe Gleichung (3.34)) ohne Veränderung des Fehlers erfolgt sind. Die angegebenen Werte entsprechen jeweils den Mittelwerten aus 100 Messwiederholungen.

3.2.4.3 Standardabweichung bei der Initialisierung der Parametervektoren

In Abschnitt 3.2.3.1 wurde die Initialisierung der Partikel beschrieben. Dabei spielte insbesondere in Gleichung (3.31) die Verwendung des Zufallsprozesses eine wichtige Rolle, bei dem normalverteilte Zufallszahlen $\mathcal{N}(0, \sigma)$ für die Initialisierung der Entzerrungsparameter verwendet wurden. Da die Wahl des Parameters σ ebenfalls einen Einfluss auf den Optimierungsprozess hat, wurde der Verlauf des verbliebenden Verzerrungsfehlers innerhalb der ersten 50 Iterationsschritte für unterschiedliche Werte von σ zwischen 0.005 und 1.5 untersucht. Die Ergebnisse sind in Abb. 3.14 zu sehen. Die besten Resultate waren für $\sigma = 0.1$ zu beobachten. Kleine Werte wiesen deutlich langsamere Verbesserungen während des Optimierungsprozesses auf, während größere sogar für eine zwischenzeitliche Verschlechterung im Hinblick auf den Verzerrungsfehler sorgten. Neben $\sigma = 0.1$ lieferte auch $\sigma = 0.2$ gute Resultate. Jedoch zeigt das vorliegende Experiment, dass der gemessene Verzerrungsfehler bei $\sigma = 0.1$ etwas geringer war als bei $\sigma = 0.2$ ($10.13 \cdot 10^{-3}$ gegenüber $11.34 \cdot 10^{-3}$). Demzufolge wurde für die weiteren Experimente $\sigma = 0.1$ genutzt.

3.2.4.4 Distanzfunktion beim Drift-Schritt

In Abschnitt 3.2.2.2 wurde der Drift-Schritt des Condensation-Algorithmus beschrieben, der einen deterministischen Einfluss auf die Parameter des Verzerrungszentrums hat. Dabei ist eine polynomielle Distanzfunktion verwendet worden (siehe Gleichung (3.11)), die von den Parametern $\lambda_0, \lambda_1, \dots$ abhängt und die in den Gleichun-

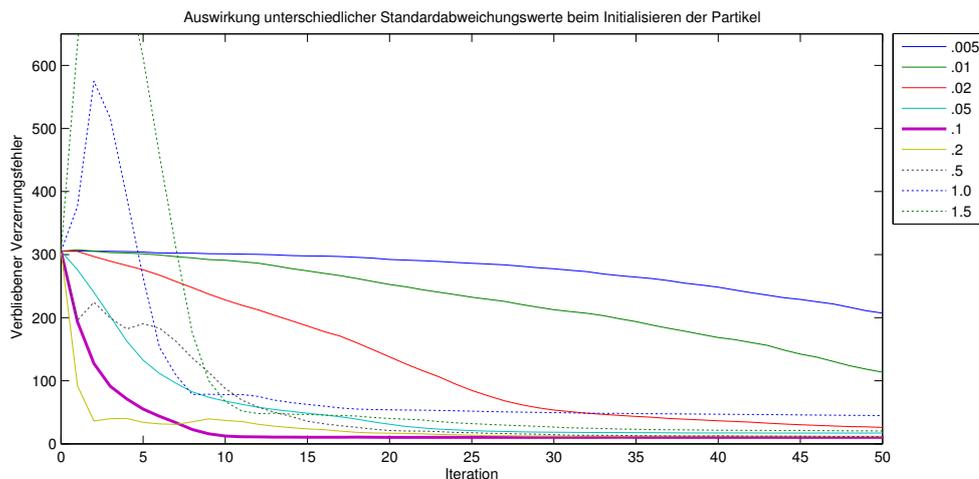


Abbildung 3.14: Verlauf des mit Faktor 1000 skalierten verbliebenen Verzerrungsfehlers bei unterschiedlicher Wahl des Parameters σ , der die Standardabweichung der normalverteilten Zufallszahlen bei der Initialisierung der Verzerrungsparameter regelt. Die besten Resultate sind für $\sigma = 0.1$ zu beobachten.

gen (3.9) und (3.10) als Faktor genutzt wurde. In Voruntersuchungen hat sich herausgestellt, dass bereits die Verwendung des ersten Parameters λ_0 ausreichend war, so dass die weiteren Parameter vernachlässigt werden konnten. Für nachfolgende Experimente wurde $\lambda_0 = 0.001$ verwendet.

3.2.5 Effiziente Berechnung entzerrter Bildpunkte

Nachdem die Konfiguration des Partikelfilters vorliegt, können die Entzerrungsparameter stochastisch optimiert werden. Sie liegen nach der Terminierung zum Zeitpunkt t im Parametervektor δ_t^{GM} vor, der die Fusion aller Hypothesen repräsentiert (siehe Gleichung (3.23)). Mit dem hier genutzten entkoppelten Backward-Verzerrungsmodell (Entzerrungsmodell) kann für einen beobachtbaren, verzerrten Punkt \mathbf{p}_d der entsprechende entzerrte Punkt \mathbf{p}_u direkt und effizient unter Verwendung von δ_t^{GM} durch

$$\mathbf{p}_u = f_U(\mathbf{p}_d, \delta_t^{GM}) \quad (3.35)$$

berechnet werden, wobei f_U die in Abschnitt 2.5.3 definierte Entzerrungsfunktion bezeichnet (siehe Gleichung (2.67)). Dies ist genauer und deutlich effizienter, verglichen mit der Verwendung des Forward-Verzerrungsmodells, bei dem die entzerrten Pixelkoordinaten iterativ mit Hilfe der Fixpunktgleichung approximiert werden müssen (siehe Abschnitt 2.5.2).

3.2.6 Effiziente Korrektur verzerrter Bilder

Im Folgenden wird ein Ansatz vorgeschlagen, bei dem die stochastisch optimierten Entzerrungsparameter genutzt werden können, um gegebene verzerrte Bilder zu entzerren.

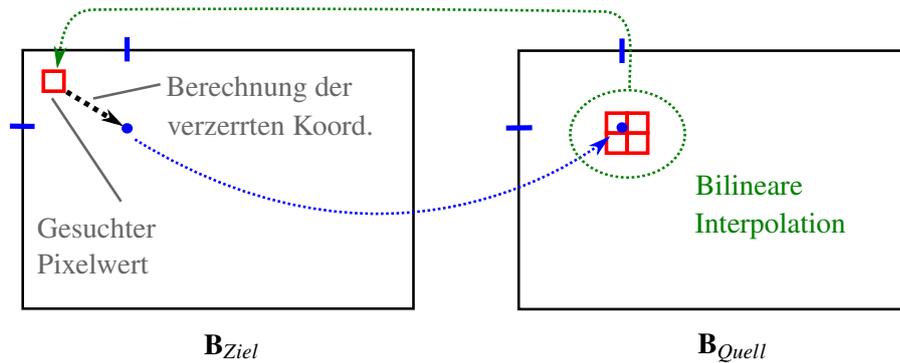


Abbildung 3.15: Grundidee für die Berechnung eines entzerrten Bildes. Der Wert des Pixels, das in rot dargestellt ist, soll in \mathbf{B}_{Ziel} berechnet werden. Zu dieser Pixelposition wird mit Hilfe der Look-Up Tabelle die entsprechende verzerrte Position bestimmt (blauer Punkt). Diese Position wird im verzerrten Bild \mathbf{B}_{Quell} aufgegriffen und eine bilineare Interpolation der Pixelwerte in \mathbf{B}_{Quell} durchgeführt. Der Ergebniswert der Interpolation wird schließlich in das rot markierte Pixel im entzerrten Bild \mathbf{B}_{Ziel} gespeichert. Nach Abarbeitung aller Pixel von \mathbf{B}_{Ziel} liegt das gesuchte entzerrte Bild vor.

Für eine Kamera ist es typischerweise notwendig, nicht nur ein einziges Bild zu entzerrern, sondern eine Menge von Bildern oder sogar einen fortlaufenden Videostrom, der in Echtzeit zu verarbeiten ist. Um die Berechnungen der entzerrten Bilder zur Laufzeit effizient durchführen zu können, wird eine Look-Up Tabelle genutzt, die offline berechnet werden kann. Dies ist generell ein sinnvolles und gebräuchliches Vorgehen, wenn die Linseneigenschaften der Kamera als konstant angenommen werden können. Beispielsweise bietet OpenCV [Bra00] Methoden an, um eine Look-Up Tabelle mit Hilfe des Forward-Verzerrungsmodells zu bestimmen. Im Gegensatz dazu beschreibt der hier vorgeschlagene Ansatz, wie die Bildentzerrung unter Verwendung des Backward-Verzerrungsmodells (Entzerrungsmodells) berechnet werden kann.

Bevor die Berechnung der Look-Up Tabelle beschrieben wird, soll die Grundidee geschildert werden. Sei \mathbf{B}_{Quell} das gegebene verzerrte und \mathbf{B}_{Ziel} das gesuchte entzerrte Bild. Der Wert eines einzelnen Pixels im entzerrten Bild \mathbf{B}_{Ziel} soll durch bilineare Interpolation der Pixelwerte aus \mathbf{B}_{Quell} in der Umgebung der entsprechenden verzerrten Position berechnet werden. Abb. 3.15 veranschaulicht diese Vorgehensweise. Die Entzerrungsfunktion $f_U(\mathbf{p}_d, \delta)$ aus Gleichung (2.67) bildet jedoch eine verzerrte Punktposition \mathbf{p}_d auf eine entsprechende entzerrte Punktposition \mathbf{p}_u ab, weshalb f_U nicht direkt verwendet werden kann, um die Position für die bilineare Interpolation zu bestimmen.

Um die gesuchten Positionen dennoch mit Hilfe von f_U zu berechnen, wird die Idee des iterativen Ansatzes aufgegriffen, die bereits beim Invertieren der Verzerrungsrichtung in Abschnitt 2.5.2 verwendet worden ist. Die gesuchten Positionen werden im Folgenden durch zwei Look-Up Tabellen zur Verfügung gestellt, wobei die eine die Pixelposition in u -Richtung und die andere jene in v -Richtung speichert. Die beiden Look-Up Tabellen seien dargestellt durch die $h \times w$ Matrizen \mathbf{M}^u und \mathbf{M}^v und werden nachfolgend auch Look-Up Matrizen genannt. Die einzelnen Einträge der Matrizen \mathbf{M}^u und \mathbf{M}^v seien m_{ij}^u und m_{ij}^v mit $i = 0, \dots, h - 1$ und $j = 0, \dots, w - 1$, so dass je ein Matrixeintrag pro Pixel des zu berechnenden entzerrten Bildes \mathbf{B}_{Ziel} mit der Höhe h

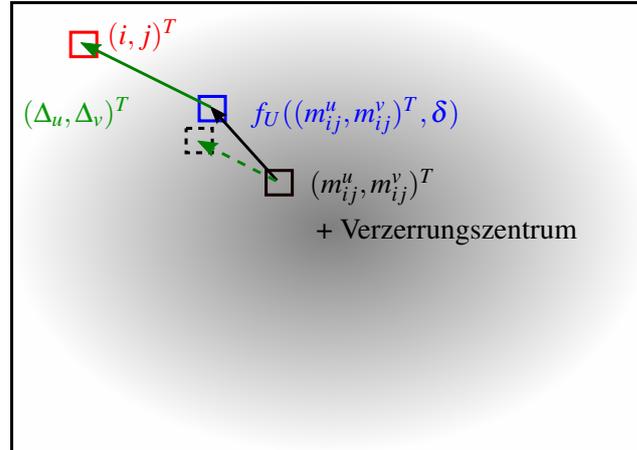


Abbildung 3.16: Ziel ist es, die Position $(m_{ij}^u, m_{ij}^v)^T$ (schwarzer Kasten) so zu bestimmen, dass durch die Anwendung der Entzerrung $f_U((m_{ij}^u, m_{ij}^v)^T, \delta)$ (schwarzer Pfeil) möglichst exakt die Position $(i, j)^T$ erreicht wird, so dass sich blauer und roter Kasten überlagern. Solange $(m_{ij}^u, m_{ij}^v)^T$ nicht ideal bestimmt ist, entspricht $(\Delta u, \Delta v)^T$ dem Abweichungsvektor (grüner Pfeil). Die Position $(m_{ij}^u, m_{ij}^v)^T$ wird iterativ durch den gewichteten Abweichungsvektor (grün gestrichelter Pfeil) korrigiert und ergibt die neue, verbesserte Schätzung für $(m_{ij}^u, m_{ij}^v)^T$ (schwarz gestrichelter Kasten). Dieser Prozess wird im Algorithmus 2 innerhalb der While-Schleife so lange durchgeführt bis der Betrag von $(\Delta u, \Delta v)^T$ (Länge des grünen Pfeils) kleiner als ein gegebener Schwellenwert ist.

und der Breite w existiert. Ziel ist es, dass die Einträge m_{ij}^u und m_{ij}^v die verzerrte Position $(m_{ij}^u, m_{ij}^v)^T$ darstellen, die zu der entzerrten Pixelposition $(i, j)^T$ gehört. Mit anderen Worten: die Look-Up Matrizen \mathbf{M}^u und \mathbf{M}^v sollen so bestimmt werden, dass unter Verwendung des optimalen Entzerrungsparametervektors δ die entzerrte Position $(i, j)^T$ mit der Position $f_U((m_{ij}^u, m_{ij}^v)^T, \delta)$ möglichst übereinstimmt, so dass die Zielfunktion \mathcal{F}_{AbwLUT} gegeben durch

$$\mathcal{F}_{AbwLUT}(\mathbf{M}^u, \mathbf{M}^v, \delta) = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \|f_U((m_{ij}^u, m_{ij}^v)^T, \delta) - (i, j)^T\|^2 \quad (3.36)$$

minimiert wird. Der Algorithmus 2 liefert die gesuchten Look-Up Matrizen. Der Vektor $(\Delta u, \Delta v)^T$ beschreibt die aktuelle Abweichung zwischen der tatsächlichen entzerrten Position $(i, j)^T$ und der geschätzten entzerrten Position $f_U((m_{ij}^u, m_{ij}^v)^T, \delta)$. Diese Abweichung wird im Verlauf des Algorithmus für jeden Look-Up Matrixeintrag minimiert. Die gesuchten verzerrten Positionen $(m_{ij}^u, m_{ij}^v)^T$ werden bei jeder Iteration der While-Schleife durch den gewichteten Vektor $\lambda (\Delta u, \Delta v)^T$ mit $\lambda \in [0, 1]$ korrigiert, bis der Betrag des Abweichungsvektors $(\Delta u, \Delta v)^T$ kleiner als ein gegebener Schwellenwert τ ist. Die Abb. 3.16 verdeutlicht einen Durchgang der inneren While-Schleife des Algorithmus.

In der Praxis haben sich der Gewichtungsfaktor $\lambda = 0.5$ und als Schwellenwert $\tau = 10^{-8}$ bewährt. Bei den gewählten Parametern kam es nach 5 bis 20 Iterationen

Algorithmus 2 : Berechnung der Look-Up Matrizen

Input : Entzerrungsfunktion f_U , Parametervektor δ , Bildhöhe h , Bildbreite w ,
 Terminierungsschwellenwert τ , Gewichtungsfaktor λ

Output : Look-Up Matrizen \mathbf{M}^u und \mathbf{M}^v

```

for  $i := 0$  to  $h - 1$  do
    for  $j := 0$  to  $w - 1$  do
         $m_{ij}^u := j$  //Initialisierung der Look-Up Matrizen
         $m_{ij}^v := i$ 

for  $i := 0$  to  $h - 1$  do
    for  $j := 0$  to  $w - 1$  do
         $\Delta u := \infty$  //Initiale Abweichung
         $\Delta v := \infty$ 
        while  $\sqrt{\Delta u^2 + \Delta v^2} > \tau$  do
             $(u, v)^T := f_U((m_{ij}^u, m_{ij}^v)^T, \delta)$ 
             $\Delta u := j - u$ 
             $\Delta v := i - v$ 
             $m_{ij}^u := m_{ij}^u + \lambda \Delta u$  //Korrektur der Einträge
             $m_{ij}^v := m_{ij}^v + \lambda \Delta v$ 
    
```

pro Eintrag zur Terminierung. Der Terminierungsschwellenwert ist dabei bereits relativ strikt gewählt worden, da aufgrund der Wahl von τ der tolerierte Abweichungsfehler maximal 10^{-8} Pixel betragen durfte. Da die Berechnung der Look-Up Matrizen nur einmal offline durchgeführt werden muss, ist der Zeitaufwand von 5 bis 20 Iterationen pro Pixel unkritisch. Somit wurde beim Kompromiss zwischen Offline-Berechnungszeit und Genauigkeit zugunsten der Genauigkeit entschieden.

Liegen die Look-Up Matrizen \mathbf{M}^u und \mathbf{M}^v vor, kann für das Bild \mathbf{B}_{Ziel} der gesuchte Wert des Pixels an der Position $(i, j)^T$ schließlich als bilineare Interpolation der Pixelwerte an den vier Positionen $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ im Bild \mathbf{B}_{Quell} berechnet werden, wobei

$$\begin{aligned}
 \mathbf{a} &= (\lfloor m_{ij}^u \rfloor, \lfloor m_{ij}^v \rfloor)^T & \mathbf{b} &= (\lfloor m_{ij}^u \rfloor + 1, \lfloor m_{ij}^v \rfloor)^T \\
 \mathbf{c} &= (\lfloor m_{ij}^u \rfloor, \lfloor m_{ij}^v \rfloor + 1)^T & \mathbf{d} &= (\lfloor m_{ij}^u \rfloor + 1, \lfloor m_{ij}^v \rfloor + 1)^T
 \end{aligned} \tag{3.37}$$

gilt. Beispielbilder, die mit dem beschriebenen Verfahren nach der Optimierung der Parameter korrigiert worden sind, befinden sich in der unteren Reihe in Abb. 3.9.

3.3 Evaluierung der Entzerrungsleistung

Im Folgenden soll die Entzerrungsleistung des auf dem Partikelfilter basierenden Ansatzes (PFA) mit dem auf der OpenCV-Implementierung basierten Ansatz (OCVA) verglichen werden. Beim PFA wird das entkoppelte Backward-Verzerrungsmodell verwendet (vgl. Abschnitt 2.5.3), und die Entzerrungsparameter werden mittels der nicht-deterministischen, iterativen, Partikelfilter basierten Optimierung bestimmt (vgl. Abschnitt 3.2.3). Beim OCVA wird das Forward-Verzerrungsmodell (vgl. Abschnitt 2.3.3)

genutzt, und die Linsenparameter werden zusammen mit den intrinsischen und extrinsischen Parametern nach entsprechender Initialisierung mit dem deterministischen Levenberg-Marquardt-Algorithmus optimiert (vgl. Abschnitt 2.4). Es sei angemerkt, dass sowohl beim PFA als auch beim OCVA dieselbe Anzahl an linsenmodellrelevanten Parametern genutzt wird. Konkret werden drei radiale und zwei tangentielle Parameter verwendet (vgl. Abschnitt 2.3.3 und 2.5.3). Die Verwendung der gleichen Parameteranzahl bei beiden Ansätzen ist notwendig, um eine bessere Entzerrungsleistung nur aufgrund eines differenzierteren Modells auszuschließen. Bei den nachfolgenden Experimenten werden unterschiedliche Kamera- und Linsentypen sowie der Einfluss von Rauschen und Unschärfe evaluiert.

3.3.1 Beschreibung des Evaluierungssystems

Zunächst müssen die optimalen Parameter bestimmt werden. Anstatt diese nur anhand eines einzigen Bildes zu bestimmen, wurde eine Menge von verzerrten Bildern mit unterschiedlichen Ansichten des Kalibrierobjekts verwendet. Dies war notwendig für die Durchführung der Kalibrierung mit OpenCV. Um auszuschließen, dass eine Überanpassung der Parameter bzgl. einer Bildmenge die Evaluierungsergebnisse verfälscht, sind zwei unterschiedliche Bildmengen für jede Kamera und Linsenkombination genutzt worden: die "Kalibrierbildmenge" (KBM) und die "Evaluierungsbildmenge" (EBM). Die KBM ist für die Partikelfilter basierte Optimierung der Entzerrungsparameter sowie für die OpenCV-Kamerakalibrierung eingesetzt worden. Die optimierten Parameter wurden verwendet, um die verzerrten Bilder der EBM zum einen mit der OpenCV-Bibliothek und zum anderen mit dem in Abschnitt 3.2.6 beschriebenen, auf den Look-Up Matrizen basierenden Verfahren zu korrigieren. Die resultierenden entzerrten Bildmengen sollen OCV-EBM (auf dem OCVA basierend) und PF-EBM (auf dem PFA basierend) bezeichnet werden. Für die jeweiligen korrigierten Bildmengen OCV-EBM und PF-EBM wurde der in Gleichung (3.6) definierte Verzerrungsfehler berechnet, der nach der Korrektur noch vorliegt. Anschließend wurden die Ergebnisse verglichen. Abb. 3.17 zeigt das Evaluierungssystem im Überblick. Bei der Verarbeitung aller Bildmengen ist immer die gleiche Routine zur Extraktion der Kreuzungspunkte genutzt worden, um zu verhindern, dass Unterschiede in den Routinen einen Einfluss auf den Vergleich haben können.

3.3.2 Auswirkung des Zufallsprozesses

Zunächst wurde der Einfluss des Zufallsprozesses auf die Optimierungsergebnisse untersucht. Dabei ist das in Abb. 3.17 beschriebene Evaluierungssystem verwendet worden. Die sechs Bilder in der oberen Reihe der Abb. 3.9 wurden für die KBM genutzt. Die EBM bestand ebenfalls aus sechs Bildern, die mit der gleichen Kamera aufgezeichnet wurden und ähnliche Ansichten des Kalibrierobjekts beinhalten. Der komplette Optimierungsprozess wurde 1000 mal wiederholt und die Bilder der EBM unter Verwendung der jeweiligen optimalen Parameter korrigiert. Die gleiche Bildmenge ist mittels OpenCV korrigiert worden. Die Verzerrungsfehler nach der Korrektur sind in Tabelle 3.2 zusammengefasst. Gegenüber dem OCVA (100%) kann-

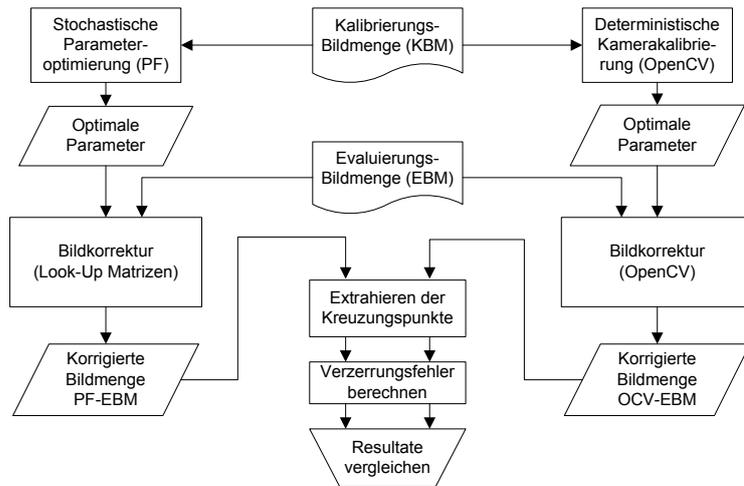


Abbildung 3.17: Das Evaluierungssystem im Überblick. Zunächst werden die Parameter beider Verfahren auf der Kalibrierbildmenge KBM im Vorfeld optimiert. Die Bilder in der Evaluierbildmenge EBM werden unter Verwendung der optimalen Parameter von den jeweiligen Verfahren korrigiert und ergeben die Bildmengen PF-EBM und OCV-EBM. Schließlich werden die Verzerrungsfehler, die die korrigierten Bilder noch beinhalten, miteinander verglichen.

	OCVA	PFA Min.	PFA Max.	PFA Mittelw.	PFA St.-Abw.
Fehler	26.5652	9.2601	9.9216	9.3895	0.0669
Anteil	100.0%	34.9%	37.3%	35.3%	

Tabelle 3.2: Mit Faktor 1000 skalierter verbliebener Verzerrungsfehler in der EBM nach Korrektur unter Verwendung des OCVA und des PFA, der 1000 mal wiederholt worden ist.

te der Fehler unter Verwendung des PFA auf fast ein Drittel reduziert werden (im Mittel: 35.3%). Der Zufallsprozess hat eine relativ geringe Auswirkung auf den resultierenden verbliebenen Verzerrungsfehler, was für eine hohe Stabilität des Verfahrens spricht und an der relativ kleinen Standardabweichung der 1000 Wiederholungen zu erkennen ist. Diese beträgt $0.0669 \cdot 10^{-3}$ bei einem mittleren Verzerrungsfehler von $9.3895 \cdot 10^{-3}$, was einem Variationskoeffizient (relative Standardabweichung) von $v = 100\% \cdot 0.0669 \cdot 10^{-3} / 9.3895 \cdot 10^{-3} = 0.71\%$ entspricht.

Die mittlere Anzahl benötigter Iterationen lag bei 557 mit einer Standardabweichung von 127. Die mittlere Berechnungsdauer entsprach 2.5 Sekunden mit dem 2.4 GHz Intel Mobile Core i5-520M Notebook-Prozessor. Zwar war der OCVA mit einer benötigten Rechenzeit von 0.1 Sekunden schneller, jedoch sei angemerkt, dass die Kalibrierung in der Regel nur einmal im Vorfeld durchgeführt wird, weshalb der größere Zeitaufwand nicht als kritisch zu beurteilen ist. Zudem sei angemerkt, dass der Schwerpunkt bei der vorliegenden Implementierung zwar nicht auf der Geschwindigkeit des Verfahrens lag, der Partikelfilter aber aufgrund seiner generellen Struktur in einem hohen Maß parallelisierbar ist, da z.B. die Berechnung der einzelnen Partikelgewich-

te (siehe Abschnitt 3.2.2.4) unabhängig voneinander durchgeführt werden kann. Wie in [GTA⁺12] berichtet, konnte bei der Nutzung eines vergleichbaren Prozessors nahezu eine Vervierfachung der Berechnungsgeschwindigkeit erreicht werden. Bei der Berechnung mit Hilfe des Grafikkartenprozessors (GPU) wurde sogar eine 35-fache Beschleunigung beobachtet. Folglich ist davon auszugehen, dass bei einer entsprechenden Implementierung des PFA eine vergleichbare Rechenzeit wie beim OCVA erreicht werden kann.

Als erstes Beispiel für den Einfluss des Zufallsprozesses auf den resultierenden Parametervektor sind die Ergebnisse für das Verzerrungszentrum in Abb. 3.18(a) visualisiert. Die Pixelpositionen der Verzerrungszentren für alle 1000 Wiederholungen des PFA sind durch graue Kreuze, und die mittlere Position (477.2, 417.9) durch das schwarze Kreuz gekennzeichnet. Details zur Verteilung der Werte sind der Abb. 3.18(b) zu entnehmen. Die Standardabweichungen in u - und v -Richtung betragen 2.17 und 2.76 Pixel. Das mit dem OCVA ermittelte Verzerrungszentrum liegt bei (473.2, 428.9) und ist mit dem roten Stern markiert. Beide Ansätze weisen eine deutliche Verschiebung gegenüber dem geometrischen Bildmittelpunkt (512.0, 384.0) auf. Bei Betrachtung der Verteilung der Verzerrungszentren und der mittleren Position der 1000 PFA Wiederholungen weicht das OCVA Resultat um das 1.9 bzw. 4.9-fache der Standardabweichung in u - bzw. v -Richtung ab. Somit liegt eine signifikante Abweichung vor.

Als zweites wurde der Einfluss des Zufallsprozesses auf die Parameter des radialen Verzerrungskorrekturmodells betrachtet. In Abb. 3.18(c) ist zwar ein leicht ansteigender Unterschied für die 1000 resultierenden Korrekturfunktionen für wachsende Radien erkennbar, dennoch weisen alle Kurven eine vergleichbare Charakteristik auf. Beim größten betrachteten Radius (400 Pixel) lag der Mittelwert der Verzerrungskorrektur bei 57.94 Pixel mit einer Standardabweichung von 0.125 Pixel. Die Differenz zwischen dem maximalen und minimalen ermittelten Wert lag bei 0.84 Pixel.

3.3.3 Variierende Brennweite

Für den nächsten Test wurde eine USB-Kamera (ImagingSource, DFK31BU03) mit einer Auflösung von 1024x768 Pixeln und einem Zoom-Objektiv (Computar T3Z3510CS 1/3" 3.5-10.5 mm) verwendet. Die Bilder für die KBM und EBM wurden mit unterschiedlichen Brennweiten zwischen $f = 3.67$ mm und $f = 8.93$ mm aufgezeichnet. Durch die Reduzierung der Brennweite vergrößert sich der Blickwinkel ("field of view"), wodurch die Linsenverzerrung an Einfluss gewinnt. In Abb. 3.19 wird der verbliebene Verzerrungsfehler für die unterschiedlichen Brennweiten nach Anwendung der Bildkorrekturen für die beiden Ansätze, den OCVA und PFA, graphisch gegenüber gestellt. Die durchgezogenen Linien stellen die Mittelwerte dar. Die Standardabweichungen beim PFA sind durch die gestrichelten Linien angedeutet, wobei der Wert der Standardabweichung durch den Abstand (Differenz) zwischen der oberen und der unteren gestrichelten Linie angezeigt ist. Die gewählte Darstellung der Standardabweichung wird in Abb. 3.20 deutlicher, da bei den dort aufgeführten Ergebnissen größere Standardabweichungen vorliegen. Die zugrundeliegenden numerischen Details sind in der Tabelle 3.3 zusammengefasst. Die Standardabweichung des

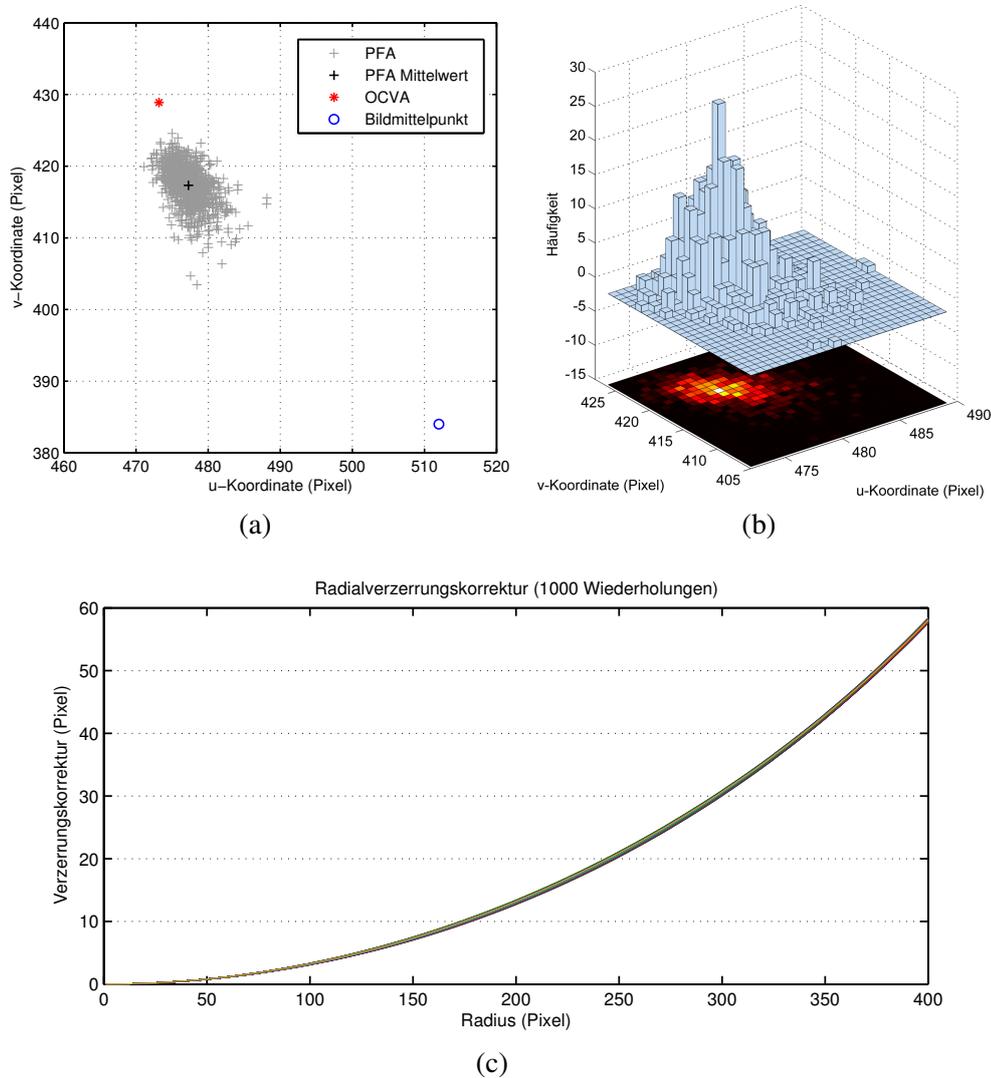


Abbildung 3.18: Visualisierung des Einflusses des Zufallsprozesses auf den resultierenden Parametervektor. In (a) sind die Positionen der Verzerrungszentren der 1000 Optimierungsdurchgänge des PFA (grau), die mittlere Position (schwarz), das mit dem OCVA ermittelte Verzerrungszentrum (rot) sowie der geometrische Bildmittelpunkt (blau) dargestellt. Die Positionen der geschätzten Verzerrungszentren beider Verfahren sind deutlich vom Bildzentrum entfernt. Das OCVA Resultat liegt außerhalb des Bereichs der PFA basierten Schätzungen. Die Verteilung der 1000 Verzerrungszentren ist anhand eines Histogramms in (b) verdeutlicht. In (c) sind die radialen Korrekturfunktionen eingezeichnet, die sich durch Verwendung des zugrundeliegenden Korrekturmodells mit den 1000 ermittelten radialen Parametersätzen ergeben. Alle Kurven weisen eine vergleichbare Charakteristik auf.

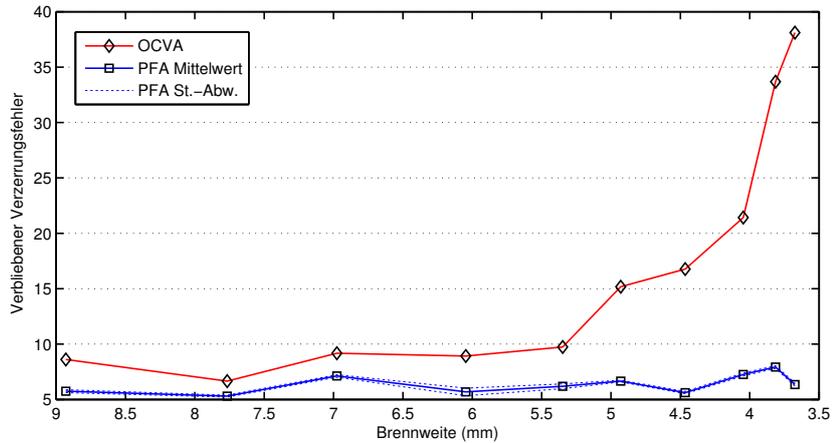


Abbildung 3.19: Vergleich des mit Faktor 1000 skalierten verbliebenen Verzerrungsfehlers nach Anwendung des OCVA (rot) und PFA (blau) für unterschiedliche Brennweiten zwischen $f = 3.67$ mm und $f = 8.93$ mm. Für die 100 Wiederholungen des PFA sind die Mittelwerte (durchgezogene blaue Linie) sowie die Standardabweichungen (gestrichelte blaue Linie) abgebildet. Die Verwendung des OCVA hinterlässt insbesondere bei kleinen Brennweiten (starken Verzerrungen) einen deutlich größeren Verzerrungsfehler.

Brennweite f	8.93	7.77	6.98	6.05	5.35	4.93	4.47	4.05	3.81	3.67
OCVA	8.61	6.66	9.18	8.93	9.74	15.17	16.77	21.42	33.68	38.11
PFA Mittelw.	5.75	5.30	7.12	5.68	6.19	6.66	5.62	7.25	7.93	6.35
PFA St.-Abw.	0.13	0.08	0.12	0.34	0.21	0.07	0.10	0.13	0.12	0.14
Anteil %	66.7	79.6	77.6	63.7	63.6	43.9	33.5	33.9	23.5	16.7

Tabelle 3.3: Numerische Zusammenfassung der Resultate des mit Faktor 1000 skalierten verbliebenen Verzerrungsfehlers nach Anwendung des OCVA und PFA für unterschiedliche Brennweiten zwischen $f = 3.67$ und $f = 8.93$ mm. Für die Auswertung wurde der PFA 100 mal wiederholt. Die letzte Zeile gibt den Anteil des Fehlers des PFA gegenüber dem des OCVA prozentual an.

verbliebenen Verzerrungsfehlers über die 100 Wiederholungen des PFA war auch bei diesem Experiment relativ gering und lag zwischen $0.07 \cdot 10^{-3}$ und $0.34 \cdot 10^{-3}$ (vgl. Tabelle 3.3), was trotz des Zufallsprozesses für die Stabilität des Verfahrens spricht. Bei allen gemessenen Brennweiten hat die PFA basierte Korrektur durchgehend bessere Ergebnisse geliefert. Während der verbliebene Verzerrungsfehler bei Verwendung des PFA über die gesamte Brennweitenspanne relativ stabil blieb, stieg der Fehler unter Verwendung des OCVA insbesondere bei kürzer werdenden Brennweiten (größeren Blickwinkeln) und den damit verbundenen stärkeren Linsenverzerrungen in den Bildern deutlich an. Die geringste Verbesserung wurde bei $f = 7.77$ mm Brennweite erzielt, bei der der Fehler unter Anwendung des OCVA (100%) auf 79.6% reduziert werden konnte. Bei der kleinsten getesteten Brennweite von $f = 3.67$ mm hingegen, konnte der Fehler mit dem PFA sogar auf nahezu ein Sechstel reduziert werden (16.7%).

3.3.4 Verrauschte Daten

Verrauschte Daten erschweren die Suche nach einem globalen Optimum aufgrund einer potentiell steigenden Anzahl lokaler Minima. Insbesondere bei steigendem Rauschen konnte experimentell gezeigt werden, dass die Linsenverzerrungskorrektur basierend auf dem PFA gute Ergebnisse liefert. Um den Einfluss des Rauschens zu untersuchen, ist wie folgt vorgegangen worden. Zunächst wurden die Positionen der Kreuzungspunkte aus den Bildern in der KBM extrahiert. Diese Positionen wurden manipuliert (verrauscht). Auf diesen verrauschten Eingabedaten wurden die Parameter beider Ansätze (OCVA und PFA) optimiert (kalibriert), wodurch die verrauschten Daten Einfluss auf die Bestimmung der optimalen Parameter hatten. Die durch das Rauschen beeinflussten Parameter beider Ansätze sind dann verwendet worden, um die Bilder in der EBM zu korrigieren. Anschließend wurde, wie in den vorangegangenen Abschnitten, der verbliebene Verzerrungsfehler ermittelt und verglichen.

Um die Positionen $\mathbf{p} = (p_u, p_v)^T$ eines Kreuzungspunktes in der KBM zu manipulieren, ist jeweils eine zufällige Verschiebung in u - und v -Richtung durchgeführt worden. Die verrauschte Position $\check{\mathbf{p}} = (\check{p}_u, \check{p}_v)^T$ wurde durch das Hinzuaddieren einer normal verteilten Zufallszahl $\mathcal{N}(0, \sigma_r)$ mit dem Mittelwert 0 und der Standardabweichung σ_r wie folgt berechnet:

$$\check{\mathbf{p}} = \begin{pmatrix} \check{p}_u \\ \check{p}_v \end{pmatrix} = \begin{pmatrix} p_u + \mathcal{N}(0, \sigma_r) \\ p_v + \mathcal{N}(0, \sigma_r) \end{pmatrix}. \quad (3.38)$$

Somit konnte über den Parameter σ_r die Stärke des Rauschens kontrolliert werden. In Abb. 3.20 wird der verbliebene Verzerrungsfehler für steigende Werte von σ_r zwischen 0.0 und 2.5 (in Schritten von 0.25) für unterschiedliche Kameras und Brennweiten für beide Ansätze verglichen. Dabei wurde das Experiment für jeden Wert von σ_r 100 mal wiederholt. Bei jeder Wiederholung ist die zufallsbedingte Manipulation der Kreuzungspunkte neu berechnet sowie die Optimierungen mit den beiden Ansätzen erneut durchgeführt worden.

Insgesamt ließ sich beobachten, dass der verbliebene Verzerrungsfehler bei Verwendung des PFA durchweg geringer war. Die Abb. 3.20(a) und 3.20(b) zeigen die Resultate bei Verwendung des Zoom-Objektivs für die Brennweiten $f = 8.9$ mm und $f = 7.0$ mm. Zusätzlich wurde das Experiment mit einem hochqualitativen Objektiv (Computar, $f = 5$ mm) sowie einem kostengünstigen Mikroobjektiv von geringer Qualität ($f = 6$ mm) durchgeführt. Die entsprechenden Ergebnisse sind in den Abb. 3.20(c) und 3.20(d) dargestellt. Die mit dem hochqualitativen Objektiv aufgezeichneten Bilder wiesen bereits eine relativ geringe Ausgangsverzerrung auf. Folglich gelang es mit beiden Ansätzen adäquate Parameter zu finden. Der verbliebene Verzerrungsfehler war im Fall von $\sigma_r = 0.0$ (kein zusätzliches Rauschen) beim OCVA (Mittelwert: $7.15 \cdot 10^{-3}$) nur geringfügig größer als beim PFA (Mittelwert: $6.91 \cdot 10^{-3}$). Mit zunehmendem Rauschen stiegen der Mittelwert (durchgezogene Linie) sowie die Standardabweichung (gestrichelte Linie) des verbliebenen Verzerrungsfehlers bei beiden Ansätzen tendenziell an. Jedoch ist deutlich zu beobachten, dass der Fehler bei Verwendung des PFA mit steigendem Rauschen langsamer anstieg als bei Verwendung des OCVA. Somit kann der PFA als robuster gegenüber Rauschen angenommen werden. Diese Tendenz ließ sich bei allen vier getesteten Objektiven beobachten. Bezüglich der

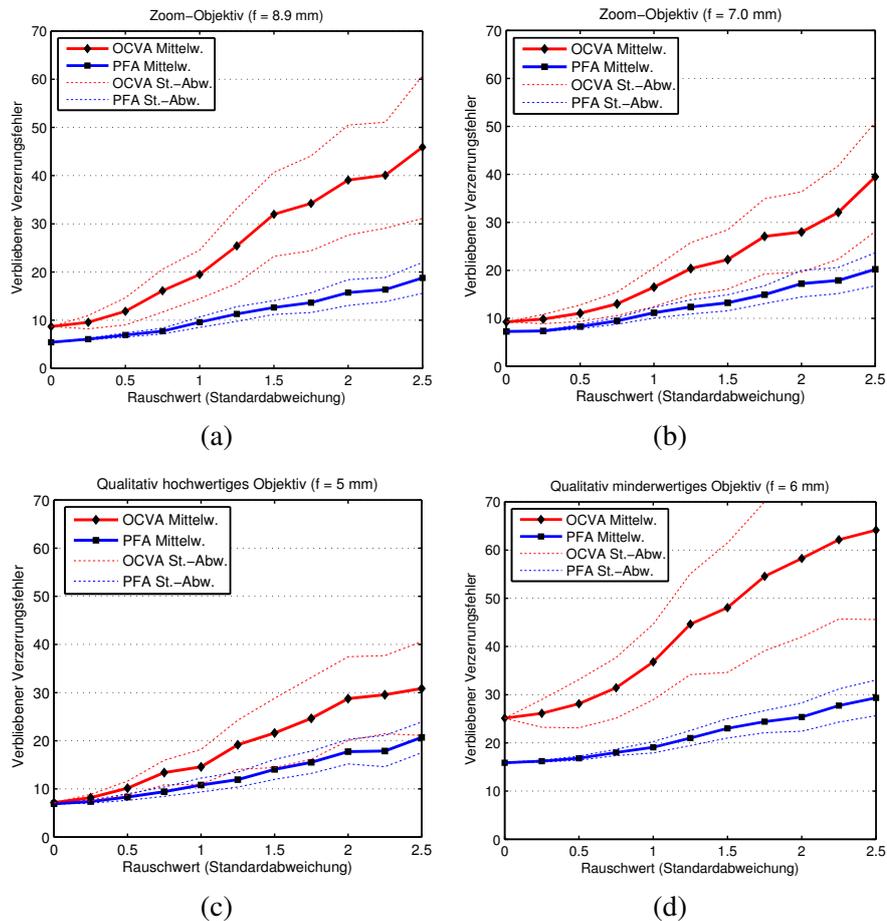


Abbildung 3.20: Vergleich des mit Faktor 1000 skalierten verbliebenen Verzerrungsfehlers bei zunehmend verrauschten Daten unter Verwendung verschiedener Objektive. Die durchgezogene und die gestrichelte Linie zeigen den Mittelwert und die Standardabweichung für je 100 Wiederholungen pro Rauschwert für beide Ansätze. Im Vergleich zum OCVA erweist sich der PFA als robuster gegenüber wachsendem Rauschen.

Standardabweichung des Fehlers konnte ebenfalls festgestellt werden, dass diese mit zunehmendem Rauschen zwar bei beiden Ansätzen angestiegen ist, jedoch deutlich weniger bei Verwendung des PFA.

3.3.5 Unschärfe Bilddaten durch fehlerhafte Fokussierung

In Anwendungsgebieten, in denen einfache Kameras ohne Zoom und Autofokus eingesetzt werden, ist es notwendig, den Fokus so einzustellen, wie es das Anwendungsszenario verlangt. Dies kann zur Folge haben, dass unscharfe Aufnahmen des Kalibrierobjekts entstehen, weil z.B. der Fokus zwar optimal für das vorgesehene Anwendungsszenario, nicht jedoch für die Aufzeichnung des Kalibrierobjekts eingestellt ist. Eine

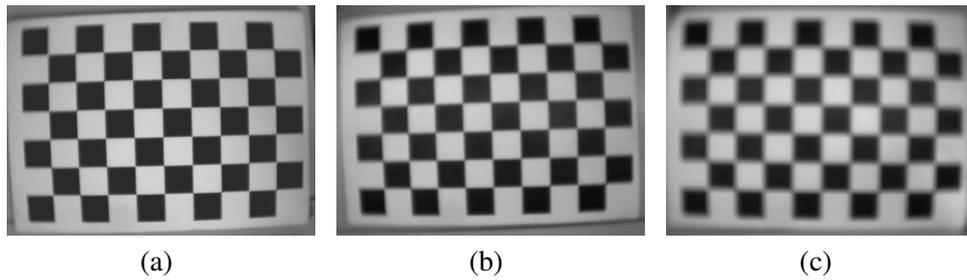


Abbildung 3.21: Beispielbilder aus der KBM, aufgezeichnet mit einer einfachen, kostengünstigen Webcam. Die verschiedenen Unschärfegrade wurden durch unterschiedliche Fokuseinstellungen der Kamera verursacht: (a) geringe Unschärfe, (b) mittlere Unschärfe und (c) starke Unschärfe.

Unschärfe	OCVA	PFA Mittelw.	Anteil (%)	PFA St.-Abw.
gering	23.2594	19.6078	84.5	0.124
mittel	56.7383	48.4202	85.9	0.373
stark	200.1100	153.6700	76.8	16.178

Tabelle 3.4: Vergleich des mit Faktor 1000 skalierten verbliebenen Verzerrungsfehlers unter Verwendung des OCVA und des PFA (100 Wiederholungen) für Bilder mit geringer, mittlerer und starker Unschärfe.

Fokusanpassung durchzuführen, um scharfe Abbildungen des Kalibrierobjekts zu erhalten, ist keine Option, da diese Anpassung Auswirkungen auf das Linsensystem und somit auf die Verzerrung haben könnte. Insbesondere bei Verwendung kostengünstiger Kameras können die Bilder auch aufgrund von Mängeln und der Verwendung qualitativ minderwertiger Materialien im Produktionsprozess Unschärfe aufweisen.

Zur Evaluierung der Entzerrungsleistung des PFA wurde eine kostengünstige Webcam (TCM 234282, 640×480 Pixel) verwendet, um die Bilder für die KBM und die EBM mit unterschiedlicher Unschärfe durch falsche Fokuseinstellungen aufzuzeichnen. In Abb. 3.21 sind für drei Unschärfestufen (gering, mittel, stark) je ein Beispiel aus der jeweiligen KBM dargestellt. Für den Vergleich wurden 100 Wiederholungen mit dem PFA durchgeführt und der verbliebene Verzerrungsfehler mit dem des OCVA verglichen. Wie in Tabelle 3.4 zusammengefasst, konnte der Fehler mit dem PFA bei starker Unschärfe auf 76.8% reduziert werden.

Es sei angemerkt, dass starke Unschärfe sowie Sensorrauschen zu einer ungenauen und instabilen Positionsbestimmung bei der Extraktion der Kreuzungspunkte des Kalibrierobjekts führten und dem Optimierungsprozess somit verfälschte (verrauschte) Daten zugrunde lagen. Über die 100 Wiederholungen hinweg betrachtet, hatten diese verfälschten Daten im Fall der starken Unschärfe (siehe Abb. 3.21(c)) eine relativ große Standardabweichung des verbliebenen Verzerrungsfehlers von $16.178 \cdot 10^{-3}$ zur Folge. Die Standardabweichung bei geringer und mittlerer Unschärfe war dazu verhältnismäßig klein, was darauf zurückgeführt wird, dass der Extraktionsalgorithmus die tatsächliche Kreuzungspunktlage bei diesen beiden Unschärfegraden deutlich genauer ermitteln konnte.

3.3.6 Fischaugen-Objektiv

Als Fischaugen-Objektive werden Objektive bezeichnet, die einen sehr großen Blickwinkel ("field of view") besitzen. Solche Objektive werden z.B. in Überwachungsszenarios eingesetzt, in denen ein Überblick über einen großen Bereich (Raum, Korridor) benötigt wird. Aufgrund des großen Blickwinkels weisen Fischaugen-Objektive typischerweise sehr starke Verzerrungen auf, die eine besondere Herausforderung darstellen.

Die Verzerrungskorrekturleistung wurde für ein Fischaugen-Objektiv mit einer Brennweite von 2.6 mm, kombiniert mit einer PAL Kamera (GF-OSC FPV) mit 420 Bildzeilen (TV-Linien), ermittelt. Die Abb. 3.22(a) zeigt ein Beispielbild aus der KBM, während 3.22(b) und 3.22(c) jeweils ein Bild aus der EBM nach der Entzerrung unter Verwendung des OCVA und des PFA vergleichen. Das Ergebnis der OCVA basierten Entzerrung wies noch einen deutlich sichtbaren Verzerrungsfehler im Randbereich auf, während der PFA sichtbar bessere Resultate lieferte. Die Ergebnisse für den verbliebenen Verzerrungsfehler wurden für den OCVA und den PFA (1000 Wiederholungen) ermittelt und in Tabelle 3.5 zusammengefasst. Gegenüber dem OCVA konnte der Fehler mit Hilfe des PFA im Mittel auf 35.1% reduziert werden.

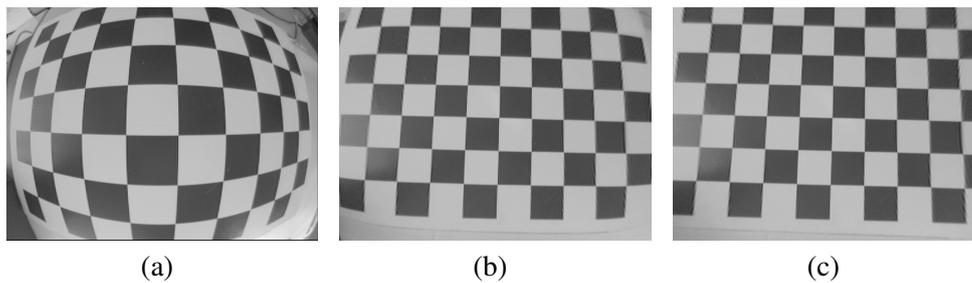


Abbildung 3.22: Ein Beispielbild aus der KBM mit starker Verzerrung, verursacht durch ein Fischaugen-Objektiv ($f = 2.6$ mm Brennweite), ist in (a) dargestellt. In (b) bzw. (c) ist je ein Bild aus der EBM dargestellt, das sich nach der Verzerrungskorrektur unter Verwendung des OCVA bzw. des PFA ergibt. Während in (b) noch eine Verzerrung erkennbar ist, konnte mit dem PFA die Verzerrung in (c) nahezu vollständig korrigiert werden.

	OCVA	PFA Min.	PFA Max.	PFA Mittelwert	PFA St.-Abw.
Fehler	117.176	39.7658	48.8026	41.1166	0.5222
Anteil	100%	33.9%	41.6%	35.1%	

Tabelle 3.5: Der mit Faktor 1000 skalierte verbliebene Verzerrungsfehler in der EBM nach Korrektur der durch ein Fischaugen-Objektiv verursachten Verzerrung unter Verwendung des OCVA und des PFA (1000 Wiederholungen).

3.3.7 Vergleich mit der OCamCalib-Toolbox

Da Fischaugen-Objektive aufgrund der starken Verzerrung eine besondere Herausforderung darstellen, wurde die Korrekturleistung des PFA zusätzlich mit den Resultaten der OCamCalib-Toolbox für MATLAB [Sca05] verglichen, die frei zugänglich ist und auf den Arbeiten [SS07, SMS06] basiert. Diese Toolbox unterstützt unter anderem die Verzerrungskorrektur von omnidirektionalen Objektiven, insbesondere von Spiegellinsenobjektiven (katadioptrische Systeme) und Fischaugen-Objektiven mit einem Blickwinkel bis zu 190° .

Für den ersten Test wurden dieselben Bilddaten der Fischaugen-Kamera aus Abschnitt 3.3.6 verwendet. Mit der OCamCalib-Toolbox ist zunächst die Kalibrierung auf der KBM und anschließend die Korrektur der Bilder der EBM durchgeführt worden. Für die korrigierte Bildmenge wurde der verbliebene Verzerrungsfehler ermittelt. Dieser lag bei $57.9 \cdot 10^{-3}$. Im Vergleich zum OCVA ($117.2 \cdot 10^{-3}$) konnte mit der Toolbox der verbliebene Verzerrungsfehler somit auf 49.4% reduziert werden. Dies ist bereits

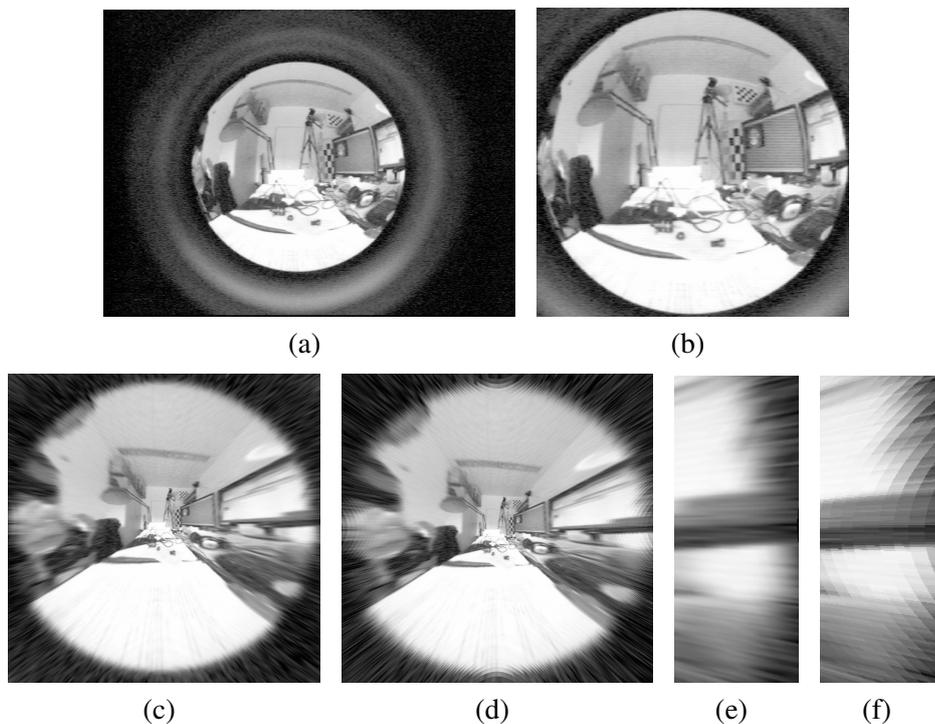


Abbildung 3.23: In (a) ist ein Super-Weitwinkel Fischaugen-Objektiv (160° Blickwinkel, 2 mm Brennweite) verwendet worden. Das Bild in (a) wurde auf den relevanten Bereich zugeschnitten (400×400 Pixel) und ist in (b) dargestellt. Dieser Bildausschnitt wurde unter Verwendung des PFA bzw. der OCamCalib-Toolbox entzerrt und die Resultate sind in (c) bzw. (d) abgebildet. Mittig an den Bildrändern lassen sich deutliche interpolationsbedingte Artefakte in den mit der OCamCalib-Toolbox berechneten Resultaten feststellen. Diese sind besonders gut an dem Vergleich zwischen (e) und (f) zu erkennen, die jeweils einen Ausschnitt (mittlerer Bereich des rechten Bildrandes) aus den Bildern (c) und (d) vergrößert darstellen.

eine deutliche Verbesserung gegenüber dem OCVA. Wie der Tabelle 3.5 entnommen werden kann, konnte mit dem PFA unter Verwendung derselben KBM und EBM sogar ein noch kleinerer verbliebener Verzerrungsfehler von $41.1 \cdot 10^{-3}$ erreicht werden, was einer Reduzierung auf 35.1% entspricht. Im direkten Vergleich der OCamCalib-Toolbox mit dem PFA konnte der Fehler somit auf 71.0% reduziert werden.

Für einen weiteren Vergleichstest wurde zwar die gleiche Kamera, jedoch ein Super-Weitwinkel-Objektiv mit 2 mm Brennweite und einem Blickwinkel von 160° verwendet. Dieses Fischaugen-Objektiv ist so konstruiert, dass der gesamte, kreisförmige Abbildungsbereich vollständig auf den Sensor passt (vgl. Abb. 3.23(a)). Alle aufgezeichneten Bilder sind gleichermaßen auf den relevanten Bildausschnitt der Größe 400×400 Pixel zugeschnitten (vgl. Abb. 3.23(b)) und in die KBM und EBM aufgeteilt worden. Der Vergleich beider Verfahren hat ergeben, dass der verbliebene Verzerrungsfehler unter Verwendung der Toolbox bei $122.4 \cdot 10^{-3}$ und beim PFA bei $74.5 \cdot 10^{-3}$ lag. Somit konnte der Fehler auf 60.9% reduziert werden. Entzerrte Beispielbilder sind in Abb. 3.23(c) und 3.23(d) dargestellt. Bei genauer Betrachtung der entzerrten Bilder werden beim Resultat der OCamCalib-Toolbox Artefaktbildungen mittig an den Bildrändern sichtbar. Das PFA-Resultat, welches auf dem in Abschnitt 3.2.6 beschriebenen Bildentzerrungsalgorithmus basiert, wies eine derartige Artefaktbildung nicht auf.

3.4 Direkter Vergleich der Optimierungsverfahren

Im Abschnitt 3.3 wurde die Entzerrungsleistung des Partikelfilter-Ansatzes (PFA) hauptsächlich mit der des OpenCV-Ansatzes (OCVA) verglichen. Zwar benutzten beide Ansätze ein vergleichbares, polynomielles Linsenmodell mit der gleichen Anzahl an Parametern, jedoch wurden beim OCVA zusätzlich zu den linsenrelevanten Parametern die intrinsischen und extrinsischen Parameter bei der Optimierung mit berücksichtigt (siehe Abschnitt 2.4). Zusätzlich unterschieden sich die jeweiligen verwendeten Optimierungsverfahren. Im Fall des PFA wurde der in Abschnitt 3.2.2 beschriebene nichtdeterministische (stochastische) Parameteroptimierungsansatz verwendet. Beim OCVA sind die linsenverzerrungsrelevanten Parameter mit dem deterministischen Levenberg-Marquardt-Algorithmus [Mor77] optimiert worden.

Ziel dieses Abschnitts ist es, die Leistungsfähigkeit der Optimierer unter Ausschluss aller anderen verfahrensbedingten Unterschiede (Forward- bzw. Backward-Verzerrungsmodell, zusätzliche Bestimmung der intrinsischen und extrinsischen Parameter) miteinander zu vergleichen.

3.4.1 Beschreibung des Evaluierungssystems

Der deterministische Ansatz soll im Prinzip demselben Vorgang wie beim PFA entsprechen (siehe linke Seite des Evaluierungssystems in Abb. 3.17), jedoch wird nicht der iterative Condensation-Algorithmus (siehe Abschnitt 3.2.2), sondern der deterministische Levenberg-Marquardt-Algorithmus verwendet, um den optimalen Parametervektor δ^{LMA} des entkoppelten Backward-Verzerrungsmodells, wie in Abschnitt 2.5.3

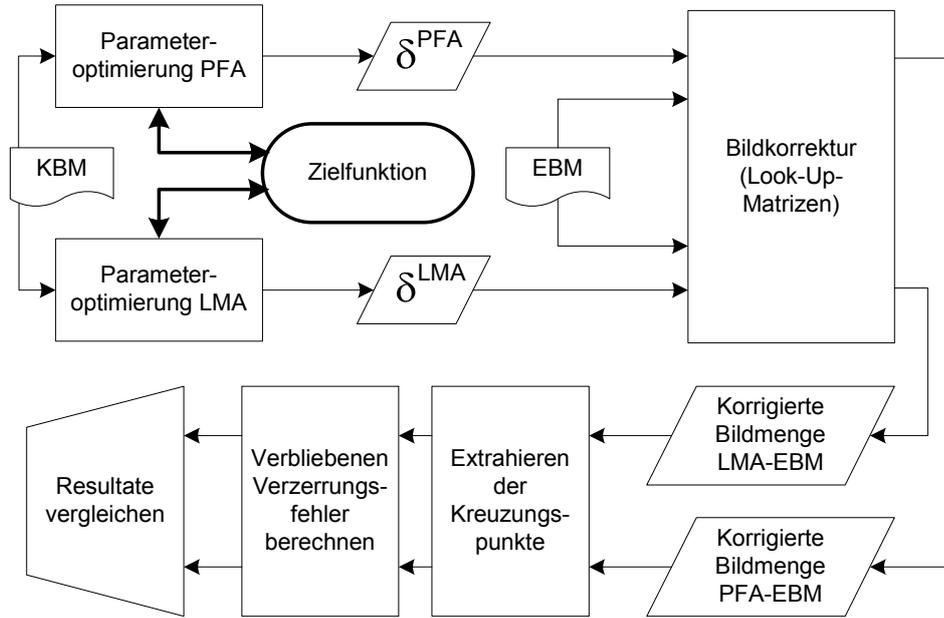


Abbildung 3.24: Beschreibung des Evaluierungssystems für den Vergleich der Optimierungsverfahren. Unter Verwendung der gleichen Zielfunktion wird die Kalibrierbildmenge (KBM) genutzt, um die optimalen Entzerrungsparametervektoren δ^{PFA} bzw. δ^{LMA} des entkoppelten Backward-Verzerrungsmodells mit Hilfe des stochastischen Partikelfilter-Ansatzes bzw. des deterministischen Levenberg-Marquardt-Algorithmus zu bestimmen. Die Evaluierbildmenge (EBM) wird jeweils mit dem in Abschnitt 3.2.6 beschriebenen Bildentzerrungsverfahren unter Verwendung von δ^{PFA} und δ^{LMA} korrigiert. Für die beiden korrigierten Bildmengen LMA-EBM und PFA-EBM werden jeweils der verbliebene Verzerrungsfehler ermittelt und die Resultate verglichen.

definiert, zu bestimmen. Verwendet wurde *levmar* [Lou04], eine frei verfügbare C++ Implementierung des Levenberg-Marquardt-Algorithmus, die wiederum auf der ausgereiften Bibliothek *LAPACK* [ABD⁺90] basiert. Der auf dem Levenberg-Marquardt-Algorithmus basierende Ansatz (LMA) soll, analog zum PFA, den Parametervektor auf einer Kalibrierbildmenge $\mathcal{B}^{KBM} = \{\mathbf{B}_0, \dots, \mathbf{B}_{m-1}\}$ durch die Minimierung der Funktion

$$\mathcal{E}_{Residuum}(\mathcal{B}^{KBM}, \delta^{LMA}) = \frac{1}{m} \sum_{j=0}^{m-1} \mathcal{E}_{KP}(\mathcal{F}_{UndistKP}(\mathcal{F}_{ExtKP}(\mathbf{B}_j), \delta^{LMA})) \quad (3.39)$$

optimieren. Diese Funktion (verbliebener Verzerrungsfehler) ist bereits im "Measurement"-Schritt zur Berechnung der Partikelgewichte verwendet worden (vgl. Abschnitt 3.2.2.4.1). Die Initialisierung des Parametervektors wurde in Anlehnung an die in Abschnitt 3.2.3.1 beschriebene Initialisierung des PFA wie folgt durchgeführt:

$$\delta^{LMA} = (\kappa_1, \kappa_2, \kappa_3, \rho_1, \rho_2, c_u, c_v) = \left(0, 0, 0, 0, 0, \frac{w_{sensor}}{2}, \frac{h_{sensor}}{2}\right). \quad (3.40)$$

Nachdem die optimalen Parametervektoren δ^{LMA} und δ^{PFA} mit beiden Ansätzen bestimmt worden sind, wurden die Bilder der EBM unter Verwendung dieser Parame-

ter korrigiert und der verbliebene Verzerrungsfehler für den Vergleich der Verfahren herangezogen. Eine detaillierte Beschreibung des Evaluierungssystems ist in Abb. 3.24 dargestellt.

3.4.2 Evaluierung

Für die Evaluierung sind insgesamt zehn verschiedene KBM und EBM ausgewertet worden. Die Bildmengen wurden mit dem aus Abschnitt 3.3.3 bekannten Zoom-Objektiv für unterschiedliche Brennweiten zwischen $f = 3.67$ und $f = 8.93$ mm erzeugt. Die Auswertungen wurden in Abb. 3.25 zusammengefasst und einmal ohne (durchgezogene Linien) und einmal mit hinzugefügtem Rauschen (gestrichelte Linien), wie in Abschnitt 3.3.4 beschrieben, durchgeführt. Obwohl der PFA als auch der LMA die jeweiligen optimalen Entzerrungsparameter δ^{PFA} und δ^{LMA} unter Verwendung der gleichen Zielfunktion auf jeweils den gleichen Eingabedaten (KBM) bestimmt haben, und auch die Evaluierung auf jeweils den gleichen Bilddaten (EBM) durchgeführt worden ist, lassen sich Unterschiede zwischen den beiden Ansätzen beobachten. Der PFA lieferte durchgängig bessere Ergebnisse. Insbesondere bei kleiner werdenden Brennweiten (stärkeren Verzerrungen) wurde der Unterschied deutlicher (siehe Abb. 3.25). Lag zusätzlich hinzugefügtes Rauschen vor ($\sigma_r = 2.5$), so wiesen zwar beide Ansätze einen insgesamt größeren verbliebenen Verzerrungsfehler auf, jedoch wurde der Vorteil des PFA, insbesondere im Brennweitenbereich 5.35 bis 8.93 mm, deutlich. In diesem Bereich erhöhte sich der Mittelwert der Differenz des verbliebenen Verzerrungsfehlers zwischen dem LMA und dem PFA auf das 2.5-fache gegenüber dem Fall ohne Rauschen, was für eine größere Robustheit des PFA spricht.

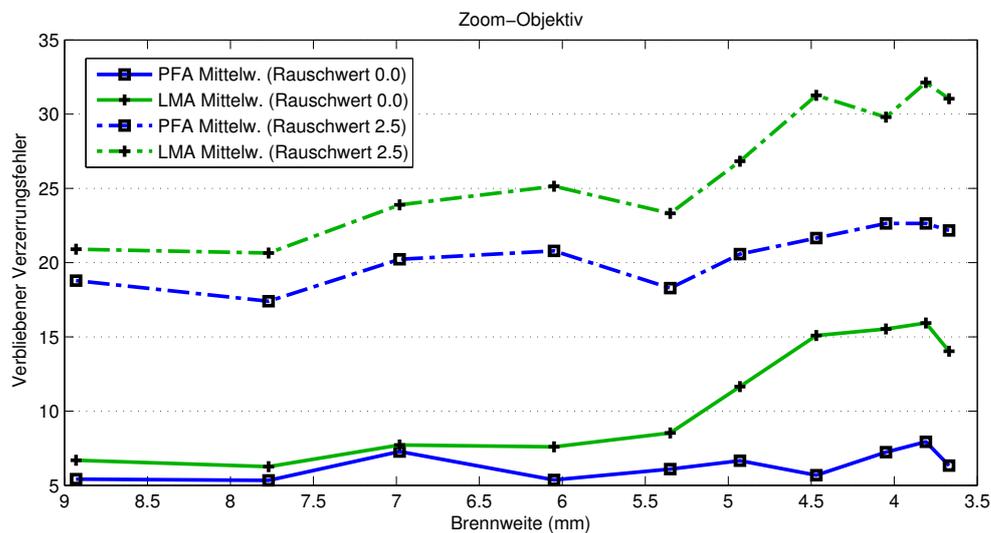


Abbildung 3.25: Vergleich des mit Faktor 1000 skalierten verbliebenen Verzerrungsfehlers für 10 unterschiedliche Brennweiten zwischen $f = 3.67$ mm und $f = 8.93$ mm ohne ($\sigma_r = 0.0$) und mit hinzugefügtem Rauschen ($\sigma_r = 2.5$). Die PFA bzw. LMA basierten Resultate sind in blau bzw. grün dargestellt. Die Ergebnisse mit hinzugefügtem Rauschen sind durch gestrichelte Linien angedeutet.

3.5 Grenzen des Partikelfilter-Ansatzes

Bei den zahlreichen Kameras und Objektiven, die im Rahmen dieser Arbeit, aber auch im Vorfeld, untersucht worden sind, hat sich die Philips SPC900NC Kamera (55° Blickwinkel), die nahezu keine Linsenverzerrung aufweist, als problematisch herausgestellt. Um einen Vergleich dieser Kamera mit einigen der in den vorangegangenen Abschnitten verwendeten Kameras bzw. Objektiven bzgl. der Verzerrung vor und nach der Korrektur zu machen, sei auf die Tabelle 3.6 verwiesen. Sogar die Bilder des hochqualitativen Objektivs (HQ) (vgl. Abschnitt 3.3.4) wiesen im Originalzustand eine mehr als doppelt so große Verzerrung auf. Wie in Abb. 3.26 zu erkennen ist, waren die PFA basierten Resultate, insbesondere bei steigendem Rauschen, auch im Fall einer geringen vorliegenden Verzerrung, durchgehend besser als die des OCVA. Bei den Rauschwerten $\sigma = 0.0$ und $\sigma = 0.25$ lieferte der PFA ebenfalls leicht bessere Ergebnisse als der LMA. Ab einem Rauschwert von $\sigma = 0.5$ jedoch konnten mit dem LMA bessere Ergebnisse erzielt werden. Das Experiment wurde mit jeweils einer neu auf-

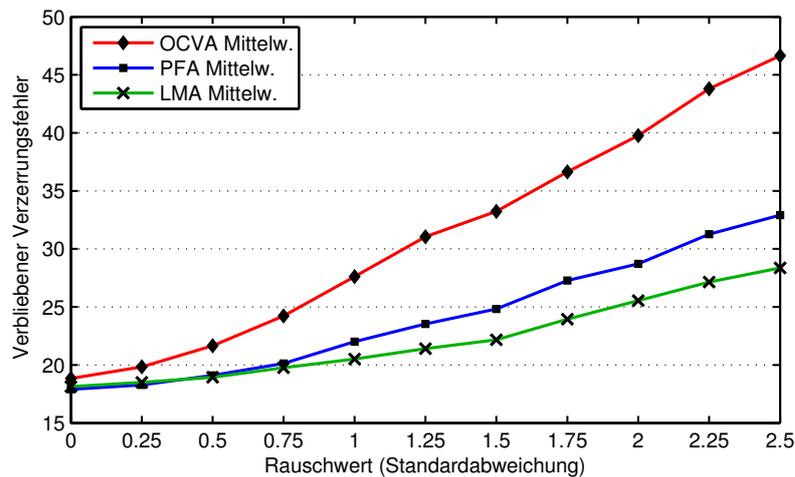


Abbildung 3.26: Der mit Faktor 1000 skalierte verbliebene Verzerrungsfehler nach Korrektur der EBM der Kamera Philips SPC900NC. Die angegebenen Werte entsprechen jeweils Mittelwerten aus 100 Durchgängen bei verschiedenen Rauschwerten zwischen $\sigma = 0.0$ und $\sigma = 2.5$.

	Zoom (f = 3.67)	Zoom (f = 8.93)	Computar (HQ)	Philips SCP900
EBM vor Korrektur	481.01	80.2239	54.521	23.4794
Anteil verbl. Verz. (%)	1.32	6.76	12.67	76.21

Tabelle 3.6: Dargestellt sind die mit Faktor 1000 skalierten Verzerrungsfehler der EBMs für einige der bereits aus vorherigen Abschnitten bekannten Kameras bzw. Objektiv im unkorrigierten Zustand sowie der verbliebene Verzerrungsfehler nach der Korrektur mit dem PFA, prozentual zum ursprünglichen Fehler. Die Bilder der Philips SCP900 Kamera weisen bereits im ursprünglichen Zustand extrem wenig Verzerrung auf, so dass die Korrektur nur einen vergleichsweise kleinen Verbesserungsanteil (%) bringt.

gezeichneten KBM und EBM dreimal wiederholt, um zufällige Störfaktoren bei der Bilderfassung auszuschließen. Dabei ergaben sich vergleichbare Resultate.

Als Erklärungsansatz wird die im "Drift"-Schritt beschriebene Problematik angenommen, dass aufgrund der sehr kleinen Verzerrungsparameter zunehmend eine Gleichverteilung der Wahrscheinlichkeit bzgl. der Position des Verzerrungszentrums entsteht, wie in Abschnitt 3.2.2.2 beschrieben. Diese resultiert in der dort erwähnten fortwährenden Zufallsbewegung ("random walk") des Verzerrungszentrums und somit in einem schlechteren Konvergenzverhalten, was schließlich zu schlechteren Ergebnissen führt. Somit bringen Kameras bzw. Objektive, die nahezu keine Linsenverzerrung aufweisen, das PFA basierte Verfahren an seine Grenzen. Diese Schwäche des PFA bei Kameras mit extrem geringer Verzerrung ist jedoch insofern vertretbar, als dass sie nur in Fällen von extrem geringer Verzerrung aufgetreten ist, die je nach Präzisionsanspruch, möglicherweise nicht weiter korrigiert werden muss.

Da sowohl der LMA als auch der OCVA auf einem deterministischen Optimierungsansatz basieren, sei auf die Annahme hingewiesen, dass die bessere Entzerrungsleistung des LMA (insbesondere bei steigendem Rauschen) zum einen auf die in Abschnitt 3.1.2.1 vorgeschlagene Zielfunktion und zum anderen auf die in Abschnitt 3.1 geforderte Beschränkung auf linsenrelevante Parameter zurückgeführt werden kann.

Abschließend sei angemerkt, dass nur bei der Philips SCP900NC Kamera und nur bei einem hinzugefügtem Rauschen ($\sigma \geq 0.5$) der LMA bessere Ergebnisse geliefert hat. In allen anderen getesteten Fällen hat der PFA die besseren Resultate ergeben, insbesondere im Vergleich mit dem OCVA.

4

3D-Tiefensensoren

Das Kapitel 4 gibt zum einen eine Einführung in das Thema "3D-Tiefensensoren", bei der die bereits im Abschnitt 1.1.2 des Einleitungskapitels genannten Sensoren aufgegriffen werden. Zum anderen wird eine Übersicht über ausgewählte, weitestgehend bekannte Verfahren und Konzepte gegeben, die für den vorliegenden Kontext aufgearbeitet worden sind und als Grundlage für das Kapitel 5 dienen.

Eine zuverlässige, genaue und schnelle Analyse von dreidimensionalen Szenen ist Grundvoraussetzung für eine Vielzahl von Anwendungsbereichen. Die Rekonstruktion von Objektgeometrie, Objektverfolgung, Objekterkennung, Lagebestimmung, Roboternavigation, Kollisionserkennung oder die Gestenerkennung stellen nur einige Beispiele dar. Es existieren zahlreiche Methoden und Systeme zur Erfassung der 3D-Tiefeninformation, von denen hier nur einige hervorgehoben werden. Die ersten zwei Beispiele betreffen 3D-Laserscanner (Abschnitt 4.1) und Time-of-Flight Kameras (Abschnitt 4.2), deren Funktionalität und Messeigenschaften kurz skizziert werden. In Abschnitt 4.3 folgt eine detaillierte Beschreibung des stereokamerabasierten Distanzmessverfahrens, das auf der Kombination von zwei üblichen Kameras (vgl. Kapitel 2) basiert, sowie die Schilderung eines Stereo-Kalibrierungsverfahrens. Der Abschnitt 4.4 verdeutlicht die Funktionsweise von Tiefensensoren, die auf dem Prinzip des strukturierten Lichts arbeiten, woraufhin in Abschnitt 4.5 die Kinect, ein auf diesem Prinzip basierender Tiefensensor, detailliert behandelt wird. Neben der Erläuterung der technischen Details, erfolgt die Beschreibung der mathematischen Grundlagen dieses Tiefensensors, die mit denen des Stereo-Kamerasystems eng verwandt sind. Dieser Abschnitt bildet den Schwerpunkt des Kapitels. Der abschließende Abschnitt 4.6 verdeutlicht das Potential des Tiefensensors und liefert somit die Motivation für weiterführende Untersuchungen.

4.1 3D-Laserscanner

Aufgrund der hohen Genauigkeit und der großen Reichweite stellen 3D-Laserscanner einen Standard bei der Erfassung von 3D-Daten dar. Dabei wird ein Laserstrahl durch einen Spiegel abgelenkt und so eine systematische Abtastung der zu erfassenden Szene durchgeführt. Das von den Szenenobjekten reflektierte Licht wird dann von einem lichtempfindlichen Sensor erfasst und gemessen. Das Ergebnis der Abtastung ist eine dreidimensionale Punktwolke der Oberfläche der Szene.

Für die Tiefenbestimmung kommen unterschiedliche Messprinzipien zum Einsatz [Luh10, Mar10]. Bei der laufzeitbasierten Variante ("Time-of-Flight") wird z.B. die Zeitdifferenz zwischen dem Aussenden des Lichtimpulses und dem Ankommen des

reflektierten Lichts gemessen. Aufgrund der bekannten Geschwindigkeit des Lichts kann aus der Zeitdifferenz die Entfernung des Objekts in Strahlrichtung ermittelt werden. Mit diesem Verfahren können große Objekte (z.B. Gebäude, Industrieanlagen, geologische Objekte usw.) mit einer Geschwindigkeit zwischen 1000 und 10000 Messpunkten pro Sekunde in Distanzen von über 1000 m vermessen werden. Die typische Messgenauigkeit liegt im Millimeter- bis Zentimeterbereich, je nach Ausführung und maximaler Messreichweite. Beispielsweise hat der "Optek ILRIS-LR¹" Scanner eine Messgenauigkeit von ± 4 cm (ermittelt bei einer Distanz von 100 m) und eine maximale Reichweite von bis zu 1330 m.

Deutlich schnellere und genauere Messungen, die jedoch mit einer wesentlich geringeren maximalen Messdistanz verbunden sind, bietet das auf dem Prinzip der Phasenverschiebung basierende Messverfahren. Dabei wird moduliertes Laserlicht mit einer festen Frequenz ausgesendet. Anhand der Verschiebung der Phase des empfangenen (reflektierten) Laserlichts gegenüber dem ausgesendeten Licht kann die Distanz berechnet werden. Ein typischer auf diesem Verfahren basierender Scanner, wie z.B. der "Faro LS 880²", tastet die Szene mit einer Geschwindigkeit von bis zu 120000 3D-Punkten pro Sekunde ab. Dabei wird eine Genauigkeit von ± 3 mm (ermittelt bei einer Distanz von 20 m) und einer maximalen Messdistanz von 50 m erzielt. Ein Scan mit einer Auflösung von 7 Millionen 3D-Punkten kann laut Herstellerangaben innerhalb von einer Minute durchgeführt werden. Die maximale Auflösung ist mit bis zu 700 Millionen Punkten angegeben. In [Bla04] ist eine Übersicht verschiedener Verfahren sowie einiger kommerziell verfügbarer 3D-Laserscanner dargestellt.

4.2 Time-of-Flight Kameras

Mit einem Anschaffungspreis von bis zu 120000 EUR^{3,4} sind 3D-Laserscanner sehr teuer. Zudem sind die großen Messbereiche für viele Anwendungen, insbesondere für Innenraumanwendungen, nicht notwendig. Eine kompaktere, leichtere und kostengünstigere Alternative bei der Realisierung des ToF-Prinzips in bildgebenden Systemen stellen die sog. Time-of-Flight Kameras dar. Das Funktionsprinzip ist mit dem der Laserscanner verwandt, jedoch wird die vorliegende Szene nicht sequenziell mit dem Laserstrahl abgetastet, sondern die Tiefeninformation der gesamten Szene gleichzeitig erfasst. Eine weit verbreitete ToF-Kamera ist die sog. PMD-Kamera (Photonic Mixed Device), die die Szene aktiv mit moduliertem, infrarotem Licht ausleuchtet. Das von der Szene reflektierte Licht wird von einer Kameraoptik erfasst und auf ein zweidimensionales Sensorfeld, bestehend aus sog. "Smart Pixels", abgebildet. Die Phasenverschiebung des einfallenden Lichts wird in jedem Pixel des Sensors ausgewertet und der Tiefenwert, der zu dem optischen Pfad vom Objekt zum jeweiligen Pixel gehört, steht somit zu Verfügung. Details zur Funktionsweise können den Arbeiten [LSKK10, KRI06, Lan00, GYB04] entnommen werden. Sowohl das Funktionsprinzip als auch eine Übersicht über Integrationsmöglichkeiten der PMD-Kameratechnologie in typische Einsatzgebiete im Kontext von "Computer Graphics" sind in [KBKL10]

¹<http://www.optech.ca>

²<http://www2.faro.com>

³<http://www.urbica.net>

⁴<http://3dlasersurveys.com>

aufgeführt.

Die PMD-Kamera liefert ein Tiefenbild, aus dem sich eine 3D-Punktwolke der Szenenoberfläche berechnen lässt, wobei die Anzahl der 3D-Punkte der Anzahl der Sensorpixel entspricht. Je nach Modell liefern PMD-Kameras zwischen 20 und 60 Tiefenbilder pro Sekunde und sind somit deutlich besser für dynamische Szenen geeignet als Laserscanner. Das Modell "SwissRanger SR-2" sowie der aktuellste Nachfolger "SR-4000" des Herstellers Mesa Imaging¹ besitzt beispielsweise eine Auflösung von 160×124 bzw. 176×144 Pixel. Eine etwas höhere Auflösung von 204×204 Pixel bietet der "CamCube3" von PMD Technologies². Somit liegt die derzeitige Auflösung unterhalb von 0.1 Megapixel und damit deutlich unter den durch Laserscanner verfügbaren Auflösungen. Maximale Messdistanzen liegen typischerweise zwischen 5 und 10 m und die Genauigkeiten (laut Spezifikationen) bewegen sich im Bereich von 1-2 cm. Genauere Untersuchungen zur Messgenauigkeit der genannten Modelle sowie eine Übersicht der Spezifikationen unterschiedlicher weiterer Modelle sind in [PR12] zu finden.

Mit einem Preis von ca. 4000 EUR³ sind die genannten PMD-Kameras zwar deutlich günstiger als Laserscanner, jedoch weisen sie für diesen Preis nur eine sehr geringe Auflösung auf. Auch neueste Generationen der PMD-Kameras in Kleinbauweise, wie z.B. die "PMD[vision] CamBoard nano⁴", ermöglichen Preise um 600 EUR, allerdings liefern diese ebenfalls nur eine geringe Auflösung von 160×120 Pixel und sind lediglich für den Einsatz im Nahbereich zwischen 0.05 und 0.5 m optimiert⁵.

4.3 Stereo-Kamerasysteme

Ein Stereo-Kamerasystem stellt eine weit verbreitete Möglichkeit der 3D-Datengewinnung dar. Im Gegensatz zu Laserscannern oder den PMD-Kameras, die spezielle Hardware zur Distanzberechnung benötigen, besteht ein Stereo-Kamerasystem aus der Kombination von z.B. zwei handelsüblichen Kameras, wie sie in Kapitel 2 detailliert beschrieben worden sind. Da solche Kameras (z.B. Spiegelreflexkameras, Kompaktdigitalkameras, Webcams, Handykameras) vergleichsweise kostengünstig und im hohen Maß verfügbar sind, ist es möglich, neben der Nutzung kommerzieller Systeme, 3D-Daten durch ein individuell zusammengestelltes Stereo-Kamerapaar zu gewinnen. Die Grundlagen zur Distanzberechnung mit Hilfe von zwei Kameras sowie der zu erwartende Fehler bei der Distanzberechnung werden in den nachfolgenden Abschnitten beschrieben.

4.3.1 Triangulierungsbasierte Distanzberechnung

Wie in Abb. 4.1 dargestellt, seien zwei Kameras gegeben, die so ausgerichtet sind, dass beide einen Punkt x betrachten, der sich im von beiden Kameras beobachtba-

¹<http://www.mesa-imaging.ch>

²<http://www.pmdtec.com>

³<http://www.acroname.com>

⁴<http://www.pmdtec.com>

⁵<https://www.cayim.com>

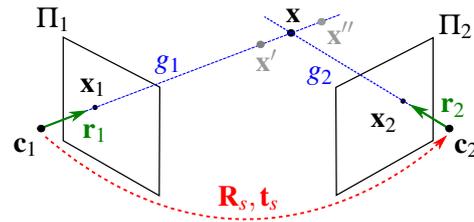


Abbildung 4.1: Stereo-Kamerakonfiguration zur Berechnung der 3D-Position eines Raumpunktes x durch Triangulierung. Die Kameras 1 und 2 sind durch ihre jeweiligen Kamerazentren c_1 bzw. c_2 sowie deren Projektionsebenen Π_1 bzw. Π_2 dargestellt. Die Kameras sind so ausgerichtet, dass der 3D-Punkt x auf jeweils x_1 bzw. x_2 projiziert wird. Die jeweiligen Projektionsstrahlen sind durch die Geraden g_1 bzw. g_2 (blau) verdeutlicht, die durch die Projektionszentren und die Richtungsvektoren r_1 und r_2 (grün) definiert werden. Die Mehrdeutigkeit der Punktposition x entlang des Projektionsstrahls bei Verwendung einer einzigen Kamera ist durch die Punkte x' und x'' (grau) angedeutet. Die relative Lage der Kameras zueinander wird durch die Rotation R_s und die Translation t_s beschrieben.

ren Ausschnitt der Szene befindet. Der Punkt x wird auf die Punkte x_1 und x_2 auf den jeweiligen Bildebenen Π_1 und Π_2 der Kamera 1 und Kamera 2 abgebildet. Die beiden Punkte x_1 und x_2 werden als korrespondierend bezeichnet, da sie zum gleichen Objekt gehören. Typischerweise müssen diese korrespondierenden Punkte in den einzelnen Bildern, z.B. anhand von Merkmalen wie den SIFT-Features [Low04], erst gefunden werden. Weitere Ansätze sind in [Sze06] beschrieben. Um den Suchraum für korrespondierende Punkte von der gesamten Bildebene auf eine einzige Gerade zu reduzieren, kann die sog. Fundamentalmatrix genutzt werden, deren Eigenschaften sich mit Hilfe der Epipolargeometrie beschreiben lassen. Details dazu sind z.B. in [HZ03, Luh10] zu finden. Im Folgenden sei angenommen, dass das Korrespondenzproblem bereits exakt gelöst ist und entsprechende Punktpaare zur Verfügung stehen. Die nachfolgenden Beschreibungen beziehen sich auf die Arbeiten [Ric11] und [Bro06].

Bei der Betrachtung einer einzigen Kamera (z.B. der Kamera 1 in Abb. 4.1) ist die Lage des Punktes x im Raum nicht eindeutig bestimmbar, da alle Punkte, die auf dem gleichen Strahl bzw. der gleichen Gerade g_1 (blau) liegen, auf den Punkt x_1 abgebildet werden (z.B. die grauen Punkte x' und x''). Mit Hilfe von zwei Kameras kann der Punkt jedoch eindeutig bestimmt werden, da dieser genau im Schnitt der beiden Geraden g_1 und g_2 (Projektionsstrahlen) liegen muss, die durch das jeweilige Kamerazentrum sowie den projizierten Punkt in der Bildebene definiert sind. Somit kann das Problem der Bestimmung der 3D-Position des Punktes durch die Berechnung des Schnittpunktes der beiden Geraden gelöst werden.

Um die Geraden im Raum berechnen zu können, müssen die Kameras kalibriert sein. Zusätzlich muss die relative Lage der beiden Kameras in dem Stereo-Kamerasystem, also die Rotation R_s und die Translation t_s , bekannt sein. Diese Transformation (Stereo-Transformation) kann mit Hilfe der Stereo-Kalibrierung, wie in Abschnitt 4.3.2 beschrieben, berechnet werden. Die Einzelkalibrierung der beiden Kameras kann entsprechend Abschnitt 2.4 durchgeführt werden, so dass die intrinsischen Kameramatrizen K_1 und K_2 zur Verfügung stehen. Die Richtungsvektoren r_1 und r_2 (grün) im

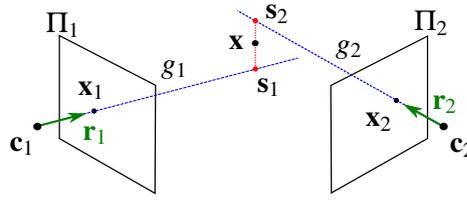


Abbildung 4.2: In der Praxis ist es aufgrund mehrerer Fehlerquellen (z.B. Rauschen) möglich, dass die beiden Projektionsstrahlen bzw. Geraden g_1 und g_2 (blau) windschief sind und keinen Schnittpunkt besitzen. Nach der Bestimmung der Punkte s_1 und s_2 , die die Strecke (rot) definieren, deren Länge dem kleinsten Abstand zwischen g_1 und g_2 entspricht, kann als approximative Lösung für \mathbf{x} die Mitte dieser Strecke gewählt werden.

Koordinatensystem der jeweiligen Kamera lassen sich durch

$$\mathbf{r}_1 = \left(\frac{p_x}{p_z}, \frac{p_y}{p_z}, 1 \right)^T \quad \text{mit} \quad \mathbf{p} = (p_x, p_y, p_z)^T = \mathbf{K}_1^{-1} \tilde{\mathbf{x}}_1, \quad (4.1)$$

$$\mathbf{r}_2 = \left(\frac{q_x}{q_z}, \frac{q_y}{q_z}, 1 \right)^T \quad \text{mit} \quad \mathbf{q} = (q_x, q_y, q_z)^T = \mathbf{K}_2^{-1} \tilde{\mathbf{x}}_2, \quad (4.2)$$

berechnen, wobei $\tilde{\mathbf{x}}_1$ bzw. $\tilde{\mathbf{x}}_2$ die homogene Darstellung des Punktes \mathbf{x}_1 bzw. \mathbf{x}_2 und \mathbf{K}_1^{-1} bzw. \mathbf{K}_2^{-1} die inverse intrinsische Matrix der jeweiligen Kamera bezeichnet (siehe Gleichung (2.61)). O.B.d.A. sei angenommen, dass der Koordinatenursprung $\mathbf{0}$ im Zentrum der Kamera 1 liegt. Dann lassen sich die beiden Geraden g_1 und g_2 in Punkt-Richtungsform durch

$$g_1 : \mathbf{a}_1 = \mathbf{0} + \lambda \mathbf{r}_1, \quad (4.3)$$

$$g_2 : \mathbf{a}_2 = \mathbf{R}_s^T (-\mathbf{t}_s) + \underbrace{\mu \mathbf{R}_s^T \mathbf{r}_2}_{\mathbf{r}'_2} \quad (4.4)$$

definieren, wobei \mathbf{R}_s und \mathbf{t}_s die Stereo-Transformation beschreiben und $\mathbf{r}'_2 = \mathbf{R}_s^T \mathbf{r}_2$ dem Richtungsvektor der Geraden g_2 im Koordinatensystem der Kamera 1 entspricht. Durch die Gleichsetzung von (4.3) und (4.4) entsteht ein Gleichungssystem aus drei Gleichungen mit den zwei Unbekannten λ und μ . Somit können diese eindeutig bestimmt werden. Durch Einsetzen der Lösung für z.B. λ in die Gleichung (4.3) kann der gesuchte 3D-Szenenpunkt \mathbf{x} in Abb. 4.1 und somit auch seine Distanz bestimmt werden.

Als Folge von Rauschen, Rundungsfehlern, ungenauen Positionsbestimmungen bei der Detektion und Lokalisierung der korrespondierenden Punkte \mathbf{x}_1 und \mathbf{x}_2 oder einer unpräzisen Kalibrierung, ist es in der Praxis nicht vermeidbar, dass die beiden Geraden windschief sein können und somit keinen gemeinsamen Schnittpunkt besitzen (siehe Abb. 4.2). Als Approximation für den gesuchten 3D-Punkt \mathbf{x} kann der Punkt gewählt werden, der sich in der Mitte der kürzesten Verbindungsstrecke (rot) zwischen den beiden Geraden g_1 und g_2 befindet. Diese Strecke wird durch die Punkte s_1 und s_2 (rot) definiert. Somit ist \mathbf{x} gegeben durch

$$\mathbf{x} = \frac{1}{2}(\mathbf{s}_1 + \mathbf{s}_2). \quad (4.5)$$

Gesucht werden die Parameter λ und μ , um die Punkte \mathbf{s}_1 und \mathbf{s}_2 auf den Geraden g_1 und g_2 zu bestimmen. Diese lassen sich wie folgt berechnen [Bro06]:

$$\lambda = \frac{\mathbf{n}_1 * \mathbf{c}_2 - \mathbf{n}_1 * \mathbf{c}_1}{\mathbf{n}_1 * \mathbf{r}_1}, \text{ mit } \mathbf{n}_1 = \mathbf{r}'_2 \times (\mathbf{r}'_2 \times \mathbf{r}_1), \quad (4.6)$$

$$\mu = \frac{\mathbf{n}_2 * \mathbf{c}_1 - \mathbf{n}_2 * \mathbf{c}_2}{\mathbf{n}_2 * \mathbf{r}'_2}, \text{ mit } \mathbf{n}_2 = \mathbf{r}_1 \times (\mathbf{r}_1 \times \mathbf{r}'_2). \quad (4.7)$$

Durch Einsetzen der beiden Parameter in die jeweilige Geradengleichung (4.3) bzw. (4.4) können \mathbf{s}_1 und \mathbf{s}_2 bestimmt und \mathbf{x} schließlich mit Formel (4.5) berechnet werden.

Wird das beschriebene Verfahren nun für jedes korrespondierende Pixelpaar der Kamerabilder durchgeführt, ergibt sich eine 3D-Punktvolke, die die dreidimensionale Struktur der Szene darstellt. Tendenziell gilt, dass bei angenommener unveränderter Korrespondenzfindungsgenauigkeit die resultierende Punktvolke umso dichter bestimmt werden kann, je größer die Auflösung der Kameras ist. Da typische Kameras Sensoren mit einer Auflösung von mehreren Megapixeln besitzen, können auf diese Weise hoch aufgelöste Punktvolken erzeugt werden. Für den Fall, dass die Szene nicht statisch ist, sei angemerkt, dass es notwendig ist, die Bilder mit beiden Kameras synchron aufzuzeichnen, da sonst Ungenauigkeiten bei der Positionsbestimmung von bewegten Objekten entstehen.

Es sei angemerkt, dass bei den Erläuterungen in diesem Abschnitt auf die Berücksichtigung der Linsenverzerrung verzichtet worden ist, um den Fokus nur auf die Grundidee der Distanzberechnung zu legen. In der Praxis ist die Linsenverzerrung jedoch zu berücksichtigen.

4.3.2 Stereo-Kalibrierung

Die Grundidee für diesen Abschnitt basiert auf den Beschreibungen in [BK08], die hier jedoch detaillierter ausgearbeitet und ergänzt werden. Ziel des Abschnitts ist die Bestimmung aller relevanten Parameter beider Kameras des Stereo-Kamerasystems. Dazu zählt neben den intrinsischen und den Verzerrungsparametern insbesondere auch die relative Lage beider Kameras zueinander, die durch die Rotationsmatrix \mathbf{R}_s und den Translationsvektor \mathbf{t}_s beschrieben werden kann. Es sei zunächst angenommen, dass beide Kameras bereits einzeln für sich wie im Abschnitt 2.4 beschrieben, kalibriert worden sind. Um \mathbf{R}_s und \mathbf{t}_s zu bestimmen, muss das Kalibrierobjekt wie in Abb. 4.3 dargestellt, so in der Szene platziert werden, dass es in beiden Kamerabildern sichtbar ist. Die Berechnung der Rotationsmatrix \mathbf{R}_1 und des Translationsvektors \mathbf{t}_1 zwischen dem Kalibrierobjekt und der Kamera 1, ausgehend vom Abbild des Kalibrierobjekts, basiert im Wesentlichen auf der in Abschnitt 2.4.2 geschilderten extrinsischen Kalibrierung. Eine detaillierte Beschreibung der Vorgehensweise für die Berechnung von \mathbf{R}_1 und \mathbf{t}_1 ist in Abschnitt 5.3.3 gegeben. Die Lage der Kamera 2, relativ zum Kalibrierobjekt, kann analog berechnet werden, so dass auch \mathbf{R}_2 und \mathbf{t}_2 zur Verfügung stehen. Die gesuchte Stereo-Transformation (in Abb. 4.3 in rot dargestellt) kann dann

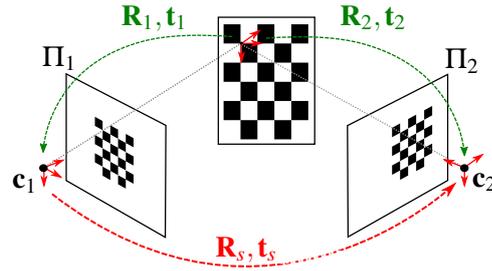


Abbildung 4.3: Bestimmung der Stereo-Transformation, bestehend aus der Rotationsmatrix \mathbf{R}_s und dem Translationsvektor \mathbf{t}_s , mit Hilfe eines Kalibrierobjekts, das von beiden Kameras gleichzeitig erfasst wird. Die gesuchte Stereo-Transformation kann mit Hilfe der Transformationen zwischen dem Koordinatensystem des Kalibrierobjekts und dem der jeweiligen Kamera ausgedrückt werden $(\mathbf{R}_1, \mathbf{t}_1, \mathbf{R}_2, \mathbf{t}_2)$.

wie folgt berechnet werden [BK08]:

$$\mathbf{R}_s = \mathbf{R}_2 \mathbf{R}_1^T, \quad (4.8)$$

$$\mathbf{t}_s = \mathbf{R}_2 (\mathbf{R}_1^T (-\mathbf{t}_1)) + \mathbf{t}_2 = \mathbf{R}_s (-\mathbf{t}_1) + \mathbf{t}_2. \quad (4.9)$$

In homogener Darstellung ist die entsprechende Stereo-Transformation gegeben durch

$$\mathbf{T}_s = \begin{pmatrix} \mathbf{R}_s & \mathbf{t}_s \\ \mathbf{0} & 1 \end{pmatrix}. \quad (4.10)$$

Werden mit dem Stereo-Kamerasystem direkt mehrere Posen des Kalibrierobjekts aufgezeichnet, kann diese resultierende Bildmenge für die Einzelkalibrierung der jeweiligen Kamera genutzt werden (vgl. Abschnitt 2.4), und eine separate Aufzeichnung für die Einzelkalibrierung entfällt. Wird das zu Beginn dieses Abschnitts beschriebene Verfahren für jede Kalibrierobjektpose durchgeführt, so ergeben sich die Stereo-Transformationen $\mathbf{T}_s^0, \dots, \mathbf{T}_s^{n-1}$, deren Werte idealerweise identisch, aufgrund von Rauschen, Rundungsfehlern usw. jedoch leicht unterschiedlich ausfallen können. Dabei entspricht n der Anzahl der verschiedenen Posen. Um diese Transformationen zu kombinieren, kann z.B. der Median der jeweiligen Einträge der Matrizen \mathbf{T}_s^i herangezogen werden [BK08]. Weitere Verfeinerungen können dann durch simultane Minimierung des Reprojektionsfehlers für beide Kameras unter Berücksichtigung der Stereo-Transformation durchgeführt werden. Die zu minimierende Kostenfunktion \mathcal{H}_s kann als Kombination der in Gleichung (2.22) beschriebenen Kostenfunktion \mathcal{H} wie folgt definiert werden (vgl. Abschnitt 2.4):

$$\mathcal{H}_s(\mathbf{R}_1^i, \mathbf{t}_1^i, \mathbf{K}_1, \delta_1, \mathbf{K}_2, \delta_2, \mathbf{R}_s, \mathbf{t}_s) = \quad (4.11)$$

$$\mathcal{H}(\mathbf{R}_1^i, \mathbf{t}_1^i, \mathbf{K}_1, \delta_1) + \mathcal{H}(\mathbf{R}_s \mathbf{R}_1^i, \mathbf{R}_s \mathbf{t}_1^i + \mathbf{t}_s, \mathbf{K}_2, \delta_2). \quad (4.12)$$

Dabei beschreiben \mathbf{R}_1^i und \mathbf{t}_1^i , $i = 0, \dots, n-1$, die Transformationen zwischen dem Kalibrierobjekt in der jeweiligen Pose und der Kamera 1. Statt analog für Kamera 2 vorzugehen, werden $\mathbf{R}_s \mathbf{R}_1^i$ und $\mathbf{R}_s \mathbf{t}_1^i + \mathbf{t}_s$ verwendet, um die Parameter der Stereo-Transformation mit zu optimieren, wobei im Idealfall (kein Rauschen usw.)

$$\mathbf{R}_2^i = \mathbf{R}_s \mathbf{R}_1^i, \quad (4.13)$$

$$\mathbf{t}_2^i = \mathbf{R}_s \mathbf{t}_1^i + \mathbf{t}_s \quad (4.14)$$

gelten würde. Die verfeinerten Parameter können z.B. mit dem Levenberg-Marquardt-Algorithmus [Mor77] berechnet werden.

4.3.3 Disparitätsbasierte Distanzberechnung

Die nachfolgenden Beschreibungen der disparitätsbasierten Distanzberechnung basieren primär auf den Ausführungen in [BK08, Sch07, Luh10] sowie auf weiteren im Folgenden genannten Quellen. In Abb. 4.4(a) ist ein Kamerasystem im Stereo-Normalfall gezeigt. Dieser stellt einen Spezialfall der allgemeinen Stereo-Kamerakonfiguration dar, wie sie z.B. in Abb. 4.1 zu sehen ist. Im Stereo-Normalfall sind die Kameras so ausgerichtet, dass die Bildebenen beider Kameras koplanar sind und beide Kameras die gleiche Brennweite f besitzen. Die Kamerazentren \mathbf{c}_1 und \mathbf{c}_2 liegen dabei üblicherweise auf der X-Achse des Weltkoordinatensystems, und beide optischen Achsen verlaufen parallel zur Z-Achse. Die Strecke zwischen den Kamerazentren wird Basis genannt, und ihre Länge (Entfernung zwischen den Kameras) sei mit b bezeichnet. Es ist nicht zwangsläufig notwendig, zwei baugleiche und entsprechend ausgerichtete reale Kameras zu verwenden. Stattdessen gibt es generell die Möglichkeit, die Bilder, die von zwei kalibrierten Kameras in allgemeiner Stereo-Kamerakonfiguration aufgenommen worden sind, so zu verzerren, dass sie dem Stereo-Normalfall entsprechen. Diese Transformation ist unter "Stereo-Rektifizierung" [SH11, Har99] oder "Berechnung von Normalbildern" [Luh10] bekannt. Nach dieser Transformation besitzen korrespondierende Punkte in beiden Bildern die gleiche v -Koordinate. Somit ist auch der Suchraum bei der Suche nach korrespondierenden Punkten auf eine Pixelzeile eingeschränkt, was eine schnellere und zuverlässigere Korrespondenzbestimmung ermöglicht. Folglich sind für die Bestimmung der Tiefe lediglich die u -Koordinaten relevant.

Seien für einen Objektpunkt \mathbf{x} die korrespondierenden Punkte in Pixelkoordinaten der jeweiligen rektifizierten Bilder durch $\mathbf{x}_1 = (u'_1, v'_1)^T$ und $\mathbf{x}_2 = (u'_2, v'_2)^T$ gegeben. Dann lässt sich laut [GFMP08] die Tiefe berechnen mit

$$z = \frac{bf}{d}, \quad (4.15)$$

wobei b der Basislänge, f der Brennweite gemessen in Pixeln und d der Disparität, also der Differenz der Pixelkoordinaten der beiden korrespondierenden Punkte in u -Richtung, gegeben durch

$$d = u'_1 - u'_2, \quad (4.16)$$

entspricht. Die Abb. 4.4(b) verdeutlicht diesen Zusammenhang. Ist die Tiefe z bekannt, so kann die 3D-Position des Punktes $\mathbf{x} = (x, y, z)^T$ berechnet werden durch

$$x = \frac{z(u'_1 - u_{0,1})}{f}, \quad (4.17)$$

$$y = \frac{z(v'_1 - v_{0,1})}{f}, \quad (4.18)$$

wobei $\mathbf{h}_1 = (u_{0,1}, v_{0,1})^T$ dem Hauptpunkt der Kamera 1 entspricht.

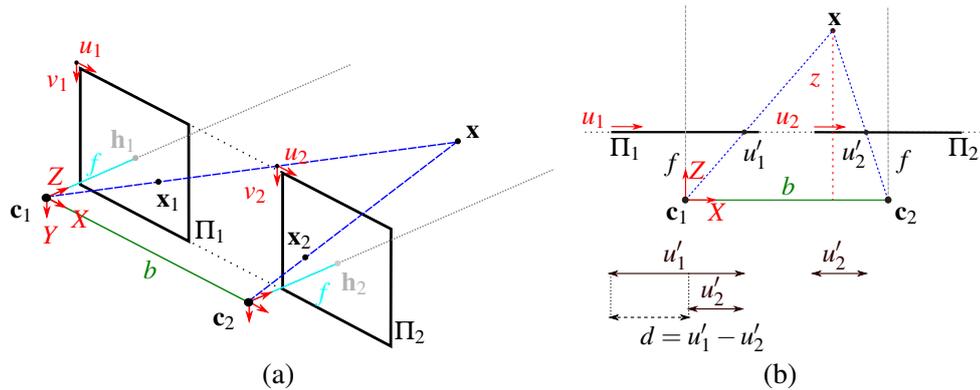


Abbildung 4.4: In (a) ist ein Kamerasystem im Stereo-Normalfall gezeigt. Dabei liegen die Bildebenen Π_1 und Π_2 in einer gemeinsamen Ebene, und die optischen Achsen (grau gestrichelt) verlaufen parallel. Sie sind definiert durch die Kamerazentren \mathbf{c}_1 bzw. \mathbf{c}_2 und die Hauptpunkte \mathbf{h}_1 bzw. \mathbf{h}_2 (grau). Die Brennweite f (hellblau) wird im Normalfall für beide Kameras als identisch angenommen. Die Kamerazentren liegen auf der X -Achse des Weltkoordinatensystems, und die Verbindungsgerade (grün) wird als Basis bezeichnet. Die Länge der Basis ist b (grün). Der 3D-Punkt \mathbf{x} wird auf \mathbf{x}_1 und \mathbf{x}_2 in den Bildebenen projiziert und liegt an den Pixelkoordinaten $\mathbf{x}_1 = (u'_1, v'_1)^T$ und $\mathbf{x}_2 = (u'_2, v'_2)^T$ vor. Da die Disparität in Richtung v_1 bzw. v_2 durch die Rektifizierung eliminiert wurde, spielen nur noch Positionen entlang den Pixelkoordinaten u_1 bzw. u_2 eine Rolle und eine vereinfachte Darstellung, wie in (b), ist ausreichend für die Berechnung der Disparität d und schließlich der Entfernung z des Punktes \mathbf{x} .

Der Fehler ε_z der berechneten Tiefe ("depth error") wächst quadratisch mit zunehmender Tiefe z und kann wie folgt abgeschätzt werden [GFMP08, BK08]:

$$\varepsilon_z = \frac{z^2}{fb} \Delta d. \quad (4.19)$$

Dabei entspricht Δd dem kleinsten erfassbaren Disparitätsschritt. Dieser ist abhängig von der Genauigkeit der Lokalisierung der korrespondierenden Bildpunkte, die üblicherweise in Pixel- oder Subpixelgenauigkeit durchgeführt werden kann. Bei Betrachtung der Gleichung (4.19) wird deutlich, dass der Fehler ε_z bei fester Entfernung z genau dann geringer wird, wenn

- die Brennweite f der Kameras,
- die Länge der Basis b (Abstand zwischen den beiden Kameras),
- die Sensorauflösung (bei gleichbleibender Größe des Sensors) oder
- die Genauigkeit der Lokalisierung der korrespondierenden Bildpunkte

erhöht wird, wobei die letzten beiden Punkte eine Auswirkung auf Δd haben. Dadurch ist der Fehler und damit verbunden auch die Genauigkeit bei Stereo-Kamerasystemen

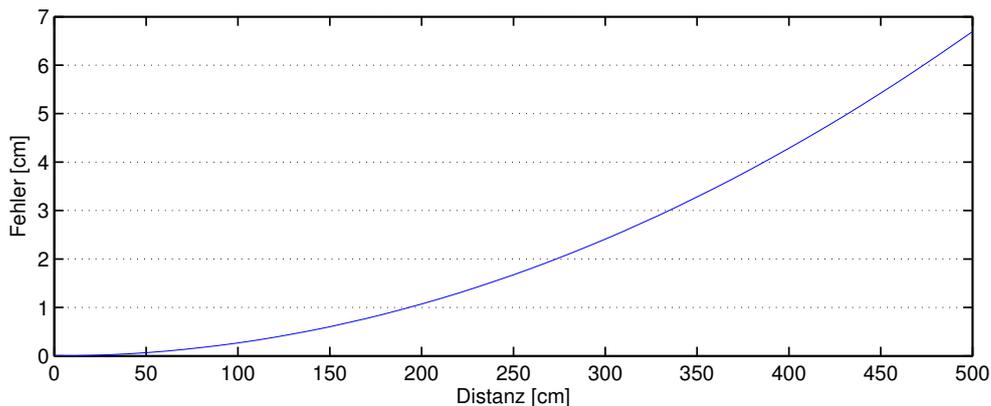


Abbildung 4.5: Verlauf des Fehlers ε_z bei steigender Distanz zum Objekt für das geschilderte Beispiel eines Stereo-Kamerasystems im Normalfall, bestehend aus zwei Kompaktkameras.

skalierbar. Dies hat den Vorteil, dass die für eine vorliegende Anwendung benötigte Genauigkeit über die geeignete Wahl der aufgelisteten Parameter erreicht werden kann.

Zur Verdeutlichung der Genauigkeit wird ein Rechenbeispiel unter Verwendung konkreter Parameter einer handelsüblichen Kompaktkamera im Preissegment unter 100 EUR durchgeführt (Canon PowerShot A2500). Diese Kamera hat die folgenden Spezifikationen: Sensorgröße: 6.17×4.55 mm, Auflösung: 4608×3456 Pixel, Brennweite 5.0 mm. Die Basislänge wird auf $b = 100$ mm festgelegt. Wird angenommen, dass die Genauigkeit der Lokalisierung der korrespondierenden Bildpunkte (nur) 1 Pixel genau ist, so gilt:

$$\Delta d = \frac{1 \text{ Pixel} \cdot 6.17 \text{ mm}}{4608 \text{ Pixel}} = 1.339 \cdot 10^{-3} \text{ mm}. \quad (4.20)$$

Abb. 4.5 zeigt den typischen Verlauf des Fehlers ε_z in Abhängigkeit von der Tiefe z für dieses Beispiel für Distanzen bis 5 m. Daran ist erkennbar, dass das Stereo-Kamerasystem sehr genaue Tiefenwerte bei geringen Distanzen liefern kann. Bei einer Tiefe von z.B. 50 cm liegt der Fehler ε_z bei 0.067 cm. Bei 4 m beträgt der Fehler aufgrund des quadratischen Zusammenhangs bereits 4.2 cm. Es sei angemerkt, dass bei diesem Beispiel keine subpixelgenaue Korrespondenzbestimmung angenommen wurde und sich der Fehler beispielsweise durch Verdopplung der Basislänge auf 20 cm halbieren lässt. Dies verdeutlicht die genannte Skalierbarkeit der Genauigkeit von Stereo-Kamerasystemen.

Kommerzielle Stereo-Kamerasysteme wie z.B. das AXIOS 3D¹ versprechen bei Distanzmessungen von bis zu 4 m eine Genauigkeit von 0.5 cm. Ein relativ neuartiges kommerzielles und sehr kompaktes Stereo-Kamerasystem, kombiniert mit einer aktiven, infraroten Beleuchtung (drei IR-LEDs) ist die "Leap Motion"², die als hand-

¹<http://www.axios3d.de>

²<https://www.leapmotion.com/>

gestenbasiertes Eingabegerät dient und auf die Positionsbestimmung der Hand und der Finger(-spitzen) im Nahbereich spezialisiert ist. In [WBRF13] wurde die Messgenauigkeit untersucht. Die ermittelte Genauigkeit lag unterhalb von 0.25 cm bei einer maximalen Reichweite von 1 m.

Beim Vergleich der in Abschnitt 4.2 angegebenen Genauigkeit einer typischen PMD-Kamera, die laut Spezifikation unterhalb von 2 cm liegt, wäre diese bei dem beschriebenen Beispiel-Stereo-Kamerasystem bei etwa 2.7 m erreicht (vgl. Abb. 4.5). In der Praxis ist die Genauigkeit der PMD-Kamera jedoch deutlich geringer als spezifiziert. Beispielsweise wurden in [LSKK10] für die PMD-Kamera SwissRanger3000 auch nach der Korrektur des systematischen Wiggling-Messfehlers ("wiggling error") noch Abweichungen bis ca. 10 cm bei Messdistanzen unterhalb von 7 m beobachtet.

Je nach Einsatzgebiet und notwendiger Messdistanz kann mit Stereo-Kamerasystemen eine angemessene Tiefenmessgenauigkeit erreicht werden, insbesondere weil diese über die Konfigurierbarkeit der Parameter wie Brennweite, Basislänge, Auflösung usw. flexibel gewählt werden kann (vgl. Formel (4.19)). In Verbindung mit den typischerweise hohen Auflösungen der dabei verwendeten Kameras, die hoch aufgelöste 3D-Punktwolken ermöglichen, stellt ein Stereo-Kamerasystem eine kostengünstige Alternative zu Laserscannern und PMD-Kameras dar.

Der Nachteil eines Stereo-Kamerasystems ist jedoch, dass die Suche nach korrespondierenden Punkten und die Berechnung der rektifizierten Bilder, insbesondere mit steigender Auflösung, sehr rechenaufwendig sein kann. Hinzu kommt, dass geeignete Punktkorrespondenzen typischerweise nur gefunden werden können, wenn die beobachteten Objekte hinreichend texturiert sind und eine ausreichende Beleuchtung vorliegt. Der nachfolgende Abschnitt beschreibt ein alternatives Verfahren zur 3D-Tiefendatengewinnung, welches die genannten Nachteile umgeht.

4.4 Ansätze mit strukturiertem Licht

Ansätze mit strukturiertem Licht ("structured light") bieten eine weitere Möglichkeit, 3D-Tiefendaten der Szene zu gewinnen. Diese Ansätze sind verwandt mit denen der Stereo-Kamerasysteme, jedoch werden typischerweise statt zwei Kameras ein Projektor verwendet, der die Szene mit einem vordefinierten Lichtmuster ausleuchtet, und nur eine Kamera, die das Lichtmuster in der Szene erfasst, siehe Abb. 4.6(a). Durch die Projektion des Musters auf die dreidimensionale Szenenstruktur lassen sich Veränderungen bzw. Verzerrungen des ursprünglichen Musters im Kamerabild erkennen, die einen Rückschluss auf die Struktur der Szene zulassen. Zunächst müssen Projektor und Kamera kalibriert und ihre Lage relativ zueinander bestimmt werden, wie z.B. in [FS11] vorgeschlagen. Sind diese Parameter bekannt, so lässt sich die Geometrie des Objekts mit Hilfe der Triangulierung bestimmen. Dies ist methodisch vergleichbar mit der Triangulierung in Abschnitt 4.3.1, wobei die Korrespondenz jetzt dadurch gegeben ist, dass das Muster des Projektors bekannt ist und lediglich im Kamerabild wiedererkannt werden muss, siehe Abb. 4.6(b).

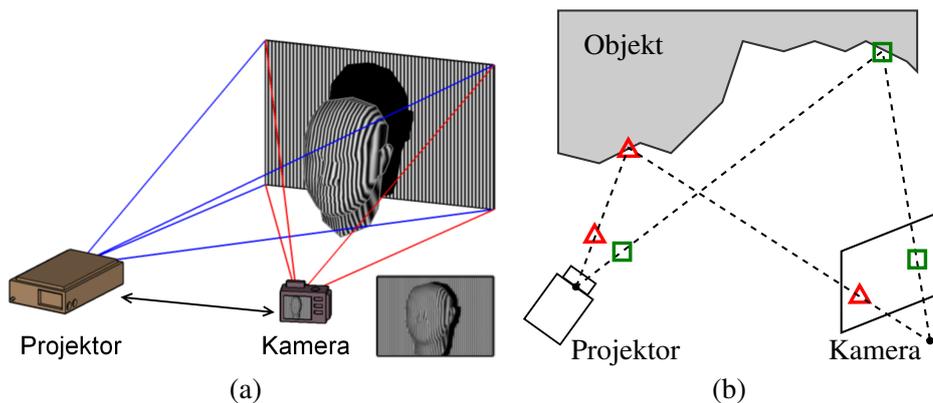


Abbildung 4.6: In (a) ist der technische Grundaufbau bei Ansätzen mit strukturiertem Licht dargestellt. Im Gegensatz zum Stereo-Kamerasystem, bei dem zwei Kameras verwendet werden, wird eine der Kameras durch einen Projektor ersetzt, dessen relative Lage zur Kamera als bekannt vorausgesetzt wird (durch schwarzen Doppelpfeil angedeutet). Dieser wirft ein Lichtmuster (z.B. ein Streifenmuster) in die zu beobachtende Szene. Dieses Muster wird von der Kamera erfasst und die Verzerrung des Musters für die Rekonstruktion der Geometrie des Objekts (Kopf) genutzt. Die Rekonstruktion der Geometrie basiert auf der Triangulierung und ist in (b) verdeutlicht, wobei eindeutig zu identifizierende Teile des bekannten Lichtmusters vom Projektor in die Szene geworfen werden (hier abstrakt durch Dreieck und Quadrat dargestellt) und im Kamerabild wiedergefunden werden können. (Bildquelle (a): Alces Technology, Inc., www.alcestech.com)

4.4.1 Lichtmusterübersicht

In der Literatur sind eine Vielzahl von Verfahren zu finden, die auf unterschiedlichen Mustern des strukturierten Lichts basieren. Eine Übersicht sowie weiterführende Arbeiten sind z.B. in [Gen11, SdMFCV02, LJL07] aufgeführt. Zur Verdeutlichung werden hier nur einige Musterarten kurz angesprochen und in Abb. 4.7 gezeigt, die auf den Beschreibungen in [Gen11] basieren und wie folgt unterschieden werden können. Zunächst werden sequenzielle ("Multi-Shot") und einzelne ("Single-Shot") Projektionen der Muster unterschieden, wobei die sequenziellen Ansätze das Grundmuster über die Zeit variieren und mehrere Aufnahmen notwendig sind.

Binäre Streifenmuster wie in Abb. 4.6(a) sind ein typisches Beispiel für einen "Multi-Shot" Ansatz, bei dem nacheinander unterschiedlich viele Streifen (z.B. 2, 4, 8, 16 usw.) auf das Objekt projiziert werden (vgl. Abb. 4.7(a)). Aus der sich so ergebenden binären Kombination ist eine Zuordnung (Korrespondenzbildung) zwischen dem idealen und dem von der Kamera beobachteten Muster möglich. Um bei gleichbleibender Genauigkeit die Anzahl der nötigen Muster zu verringern, können z.B. mehrere Graustufen verwendet werden (vgl. 4.7(b)). Dieser Ansatz gilt als robust, ist jedoch ungeeignet für dynamische Szenen, da mehrere zeitlich aufeinander folgende Bilder nötig sind.

Im Gegensatz zu "Multi-Shot" Ansätzen können "Single-Shot" Ansätze die Geometrie aus einem einzigen Bild rekonstruieren, indem mehr Information in ein einzelnes

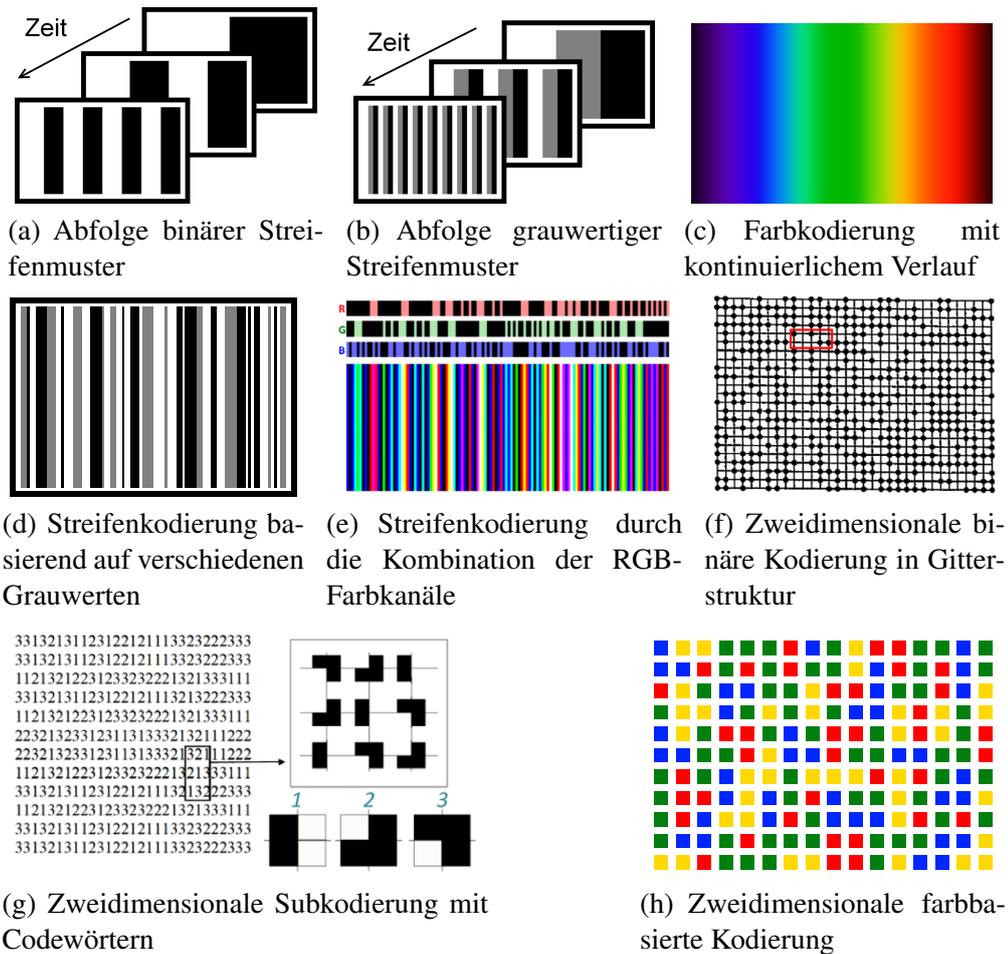


Abbildung 4.7: Beispiele für unterschiedliche Muster des strukturierten Lichts (Bildquelle (e,f,g): [Gen11]).

Muster untergebracht wird. Kontinuierliche Farbmuster sind ein solches Beispiel, bei dem unterschiedliche Farben eine eindeutige Positionierung erlauben, und das Muster einem Regenbogenverlauf ähnelt (vgl. Abb. 4.7(c)). Es ist jedoch nachteilig, dass die feinen Farbunterschiede möglicherweise nicht genau genug differenziert werden können und somit die Genauigkeit der Rekonstruktion sinkt. Alternativ kann das robuste Streifenprinzip dahingehend erweitert werden, dass eine unregelmäßige Kombination aus verschiedenen hellen Streifen verwendet wird, so dass dadurch ein Streifencode definiert wird, der sich durch die Nachbarschaft eines Streifens ergibt und wiedererkannt werden kann. Dies wird auch Streifenindizierung ("Stripe Indexing") genannt und ist gleichbedeutend damit, dass jeder Streifen einen eindeutigen und wiedererkennbaren Index trägt (vgl. Abb. 4.7(d)). Dieses Prinzip ist direkt auf die RGB-Farbkanäle erweiterbar, so dass ein Farbstreifencode vorliegt (vgl. Abb. 4.7(e)).

Um eine noch höhere Informationsdichte zu erreichen, können anstatt des eindimensionalen Streifenmusters zweidimensionale Gittermuster genutzt werden. Diese können wiederum nur binär sein (vgl. Abb. 4.7(f)) oder komplexere Submuster beinhalten, die kurze Codewörter darstellen (vgl. Abb. 4.7(g)). Dabei wird die Umgebung inner-

halb eines vordefinierten Fensters im Kamerabild untersucht und die Korrespondenz aufgrund der eindeutigen Verteilung des Musters bestimmt. Auch hier ist eine Erweiterung um eine Farbinformation möglich, um die Kombinationsmöglichkeiten der 2D-Kodierung zu erhöhen. Somit können noch kleinere Suchfenster genutzt und dennoch eine eindeutige Wiedererkennung gewährleistet werden (vgl. Abb. 4.7(h)). Dabei ist es wichtig, die zweidimensionalen Muster so zu wählen, dass sie sich nach Möglichkeit nicht wiederholen. Diese können z.B. mit Hilfe von Zufallszahlen generiert werden ("pseudo-random patterns"). Für weitere Details siehe [PD09, DP07].

4.4.2 Vorteile und kommerzielle Produkte

Ansätze mit strukturiertem Licht haben gegenüber Stereo-Kamerasystemen den Vorteil, dass die Szenenobjekte durch das strukturierte Licht mit virtuellen Merkmalen ("features") versehen werden, wodurch es nicht mehr notwendig ist, dass Objekte z.B. eine Texturierung aufweisen, die die benötigten Merkmale für die Korrespondenzbestimmung liefert. Zudem sind auf strukturiertem Licht basierende Ansätze nicht von externen Lichtquellen abhängig und funktionieren auch bei geringer Beleuchtung. Tatsächlich wäre die Abwesenheit von externem Licht sogar der Optimalfall, da übermäßige Lichteinstrahlung den Detektionsvorgang stören könnte, wenn z.B. die externe Lichtquelle stärker als der Projektor ist (direkte Sonneneinstrahlung). Bei typischen Innenraumanwendungen ist dies in der Regel jedoch unproblematisch.

Ein Beispiel für einen kommerziellen, auf strukturiertem Licht basierenden Tiefensensor ist der "HDI Advance R3 3D-Scanner" aus dem Hause "3D3Solutions"¹. Dieser erfasst die Objektgeometrie in einer Distanz von 1 m mit einer angegebenen hohen Genauigkeit von 0.1 mm (0.5 mm bei maximal 3.7 m Distanz). Jedoch hat dieser auch einen sehr hohen Preis von 17000 EUR. Zur Berechnung der 3D-Informationen wird zusätzlich ein leistungsstarker PC benötigt (2GHz CPU mit 4 Kernen, 4GB RAM, Grafikkarte mit 512MB RAM), mit dem etwa einen Scan der Szene pro Sekunde durchgeführt werden kann (1.13 Hz).

Ein weniger präzises, jedoch erheblich kostengünstigeres und als 3D-Tiefensensor verwendbares Produkt ist die seit Ende 2010 von der Firma Microsoft vertriebene Kinect², siehe Abb 4.8. Diese ist ursprünglich als Eingabesteuergerät für die Microsoft Xbox³ Spielekonsole konzipiert worden. Mit Hilfe der Kinect lassen sich Körpergesten erkennen und zur Spielsteuerung nutzen. Die Erkennung der Gesten basiert dabei auf Tiefen- bzw. Disparitätsbildern. Diese Bilder werden mit dem Ansatz des strukturierten Lichts ermittelt und direkt auf der Kinect-Hardware mit einer Wiederholrate von 30 Hz berechnet. Durch zur Verfügung stehende Schnittstellen ist es möglich, die Tiefenbilder direkt auszulesen und somit für eigene Zwecke zu verwenden. In weniger als zweieinhalb Jahren verkaufte Microsoft 24 Millionen Exemplare⁴. Dieses Interesse an dem Sensor ist nicht nur dem Unterhaltungssektor zuzuschreiben, sondern auch der Tatsache, dass aufgrund des extrem niedrigen Preises von nur rund 100 EUR, der

¹<http://www.3d3solutions.com>

²<http://www.xbox.com/de-DE/Kinect>

³<http://www.xbox.com/de-DE>

⁴<http://www.gamespot.com/news/kinect-sales-reach-24-million-6403766>



Abbildung 4.8: Die Microsoft Kinect mit den wichtigsten internen Komponenten. Der Abstand zwischen IR-Kamera und Projektor beträgt etwa 7.5 cm, zwischen IR-Kamera und RGB-Kamera etwa 2.5 cm. Neben den optischen Sensoren besitzt die Kinect ein Mikrofon-Feld aus 4 Mikrofonen und einen schwenkbaren Standfuß.

hohen Verfügbarkeit und der leichten Bauweise, die Kinect einen kostengünstigen und schnellen 3D-Tiefensensor darstellt, der innerhalb der letzten Jahre zunehmend im Bereich der Forschung und Entwicklung eingesetzt wurde.

Es existieren auch kommerziell angebotene Softwareprodukte wie z.B. "KScan3D"¹, die auf die Verwendung der Kinect als Eingabesensor spezialisiert sind und mit dieser Kombination ein 3D-Scanner-System zur Erzeugung von 3D-Objektmodellen anbieten. KScan3D unterstützt unter anderem den Tiefensensor "ASUS Xtion Pro Live"², der deutlich weniger verbreitet ist als die Kinect, jedoch auf vergleichbarer Hardware basiert. Aufgrund des höheren Preises konnte sich dieser jedoch nicht gegen die Kinect durchsetzen.

Da die Kinect für den kommerziellen Sektor sowie für die Forschung und Entwicklung relevant ist, werden in den nachfolgenden Abschnitten Details zur Kinect und ihrer Funktionsweise erläutert. Auf ihr Potential wird außerdem in Abschnitt 4.6 näher eingegangen.

4.5 Kinect als Low-Cost-Tiefensensor

Neben einem aktiv schwenkbaren Standfuß sowie weiteren Sensoren, wie z.B. einem Mikrofon-Feld aus insgesamt vier Mikrofonen, besteht die Kinect im Wesentlichen aus einer RGB-Kamera mit einer Auflösung von 1280×1024 Pixeln ($f = 6.1$ mm, Pixelgröße $5.2 \mu\text{m}$ [SJP11]) und einer Kombination aus einem Laserprojektor, der ein infrarotes Punktmuster erzeugt sowie einer Infrarotkamera, ebenfalls mit einer Auflösung von 1280×1024 Pixeln ($f = 2.9$ mm, Pixelgröße $2.8 \mu\text{m}$ [SJP11]), die das Muster erfasst. Der Projektor hat in etwa einen Abstand von 7.5 cm zur IR-Kamera, während die RGB-Kamera etwa 2.5 cm von dieser entfernt ist (siehe Abb. 4.8). Die IR-Kamera besitzt einen Filtervorsatz, der nur infrarotes Licht passieren lässt. Diese Kombination aus IR-Projektor und IR-Kamera bildet den Kern des Tiefensensors. Der Tiefensensor besitzt einen horizontalen und vertikalen Öffnungswinkel von 57°

¹<http://www.kscan3d.com/>

²http://www.asus.com/Multimedia/Xtion_PRO_LIVE/

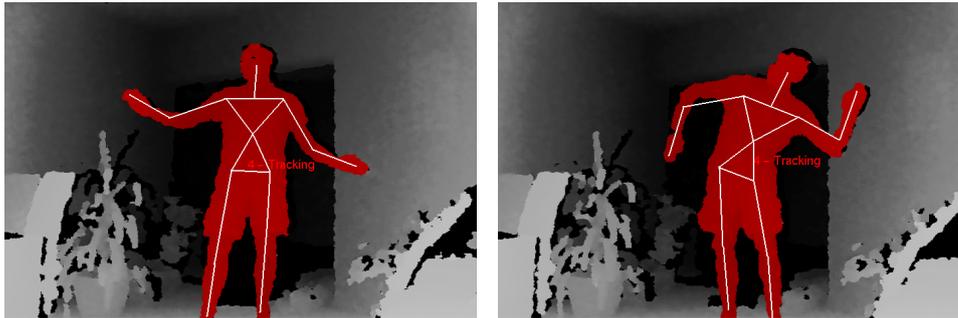


Abbildung 4.9: Bestimmung der Spielerpose mit Hilfe der durch die Kinect berechneten Tiefenbilder. Der rot markierte Bereich segmentiert den Spieler, in den das vereinfachte Skelettmodell (weiß) eingezeichnet ist. Die so ermittelte Pose dient als Eingabe zur Steuerung der Anwendung.

und 43° . Die Auswertung des erfassten IR-Kamerabildes inklusive Disparitätsberechnung wird vollständig durch die Kinect-Hardware durchgeführt. Das Resultat ist ein Disparitätsbild mit einer Auflösung von 640×480 Pixeln, in dem für jedes Pixel die berechnete Disparität zur Verfügung steht. Die Kinect besitzt eine USB-Schnittstelle, über die sich simultan das aktuelle Disparitätsbild und das RGB-Kamerabild mit einer Bildwiederholrate von 30 Hz zum PC übertragen lassen. Die Übertragung des Bildes der Infrarotkamera ist ebenfalls möglich. Laut Microsoft ist die Kinect für eine Distanz zwischen 1.2 und 3.5 m ausgelegt. Dieser Distanzbereich ist jedoch als Angabe für die Nutzung der Kinect als Eingabegerät zu verstehen. Die tatsächlich messbaren Distanzen liegen z.B. bei Verwendung des OpenNI-Frameworks [Ope10] im Bereich zwischen 0.4 und 10 m.

Wie bereits erwähnt, ist die Kinect als Eingabegerät konzipiert, wobei die Steuerung der Applikation (z.B. ein Videospiel) durch Körpergesten erfolgen soll. Dabei beobachtet die Kinect den oder die sich im Sichtfeld befindenden Spieler und es wird eine modellbasierte Schätzung der Körperposen der Spieler durchgeführt. Die Körperpose wird durch ein vereinfachtes Modell des menschlichen Skeletts repräsentiert, siehe Abb. 4.9. Die Anpassung der Skelettkonfiguration ist derart durchzuführen, dass das geschätzte Skelett möglichst optimal zu den beobachteten Tiefen- bzw. Disparitätsdaten passt. Details zur Bestimmung der Körperpose aus Tiefendaten sind z.B. in [YWY⁺11] erläutert.

Die wesentlichen Schritte der Verarbeitungspipeline bei der Nutzung der Kinect als Eingabegerät sehen somit wie folgt aus:

1. Projektion des strukturierten Punktmusters mit dem IR-Laserprojektor.
2. Erfassung des von der Szene reflektierten Musters mit der IR-Kamera.
3. Berechnung der Disparitäten für jedes Pixel im IR-Kamerabild (Disparitätsbild).
4. Umrechnung der Disparitätswerte in Tiefenwerte (Tiefenbild).
5. Schätzung der Skelettkonfiguration des Spielers anhand des Tiefenbildes.
6. Klassifikation der Spielerpose anhand der Skelettkonfiguration.
7. Steuerung der Anwendung entsprechend der klassifizierten Spielerpose.

Da im Kontext der vorliegenden Dissertation die Kinect lediglich als Tiefensensor betrachtet wird, sind im Folgenden nur die Schritte 1-4 relevant.

4.5.1 Strukturiertes Punktmuster

Das Muster des strukturierten Lichts, das der in der Kinect verbaute Projektor (siehe Abb. 4.10(a)) erzeugt, ist ein zweidimensionales, in einem Gitter angeordnetes, binäres Punktmuster. Dieses lässt sich der Abb. 4.7(f) in der Musterübersicht in Abschnitt 4.4 zuordnen. Als Lichtquelle verwendet der Projektor eine 830 nm Laserdiode mit einer Leistung von 60 mW.¹ Der Laserstrahl passiert im Wesentlichen zwei sog. diffraktive optische Elemente ("Diffractive Optical Elements", DOEs), die physikalische Eigenschaften der Lichtbeugung nutzen [SPSL10], siehe Abb. 4.10(b). Das erste Element erzeugt das eigentliche Grundmuster ("speckle pattern"). Innerhalb dieses Musters sind die Punkte unkorreliert angeordnet. Das bedeutet, dass bei Verwendung eines Suchfensters die Lage des Fensters innerhalb des Musters anhand der Punktkon-

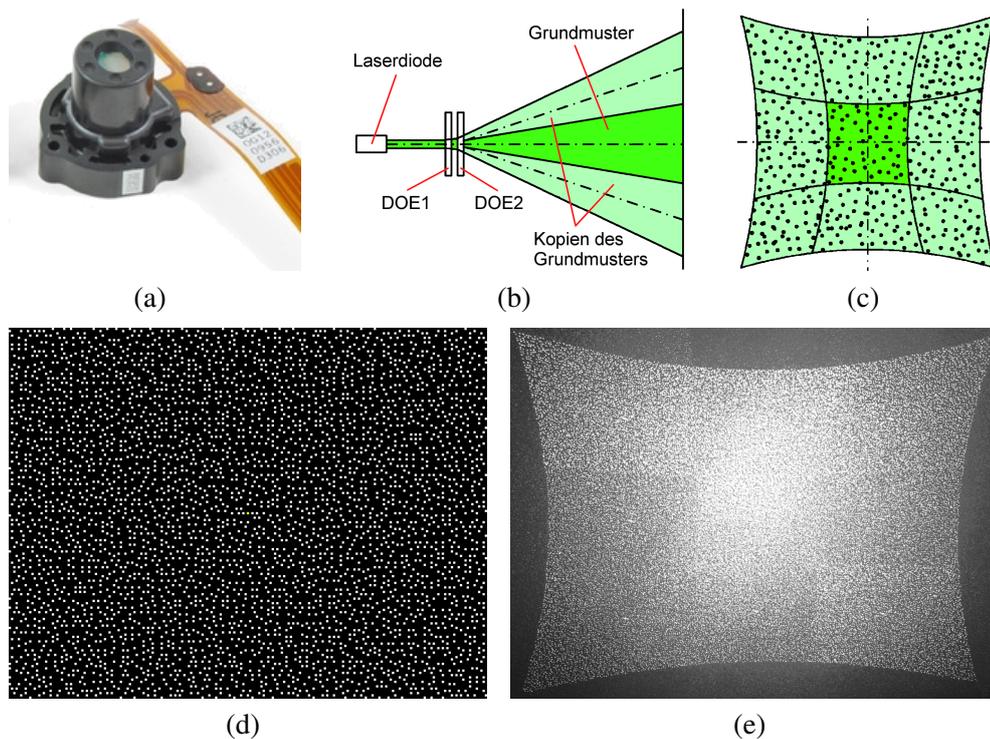


Abbildung 4.10: Der Laserprojektor (a) verwendet eine IR-Laserdiode und zwei diffraktive optische Elemente (DOEs). Das DOE1 (b) erzeugt aus dem Lichtstrahl das Grundmuster, welches idealisiert in (d) dargestellt ist, während das DOE2 die Vielfältigung des Grundmusters durchführt, so dass insgesamt ein Mosaik bestehend aus 3×3 Feldern entsteht (c) [SPSL10]. Das reale Ergebnis der Projektion ist in (e) dargestellt. (Bildquellen: (a): <http://www.ifixit.com> (Lizenz: CC BY-NC-SA 3.0), (d): <http://azttm.wordpress.com>)

¹http://openkinect.org/wiki/Hardware_info

figuration, die sich innerhalb des Fensters befindet, eindeutig bestimmt werden kann. Die Abb. 4.10(d) zeigt das ideale Punktmuster nach Ausrichtung in ein regelmäßiges zweidimensionales Gitter, welches durch "reverse engineering" ermittelt wurde¹. Das Gitter, in dem die Punkte liegen, besteht aus 211 Spalten und 165 Zeilen. Das zweite DOE splittet und vervielfacht (kopiert) das Muster so, dass eine 3×3 Matrix des ursprünglichen Musters nahtlos nebeneinander projiziert wird (siehe Abb. 4.10(c)). Erst so ergibt sich das Gesamtmuster des strukturierten Lichts (siehe Abb. 4.10(e)), dessen Gesamtöffnungswinkel größer ist als der der IR-Kamera. Somit füllt das Punktmuster den vollständigen Bildbereich der IR-Kamera aus und eine Auswertung der Disparität wird für jedes Pixel möglich. Das Muster besteht insgesamt aus 633×495 potentiellen Punktpositionen. Dies entspricht weniger als der Hälfte der Auflösung, die die IR-Kamera besitzt (1280×1024 Pixel), so dass das Muster entsprechend dem Abtasttheorem [Wah84] korrekt erfasst werden kann. Da die verwendete Optik des Projektors nicht ideal ist, weist das projizierte Muster eine kissenförmige Verzerrung auf (vgl. Abb. 4.10(e)). Aufgrund des verwendeten infraroten Lichts ist das Punktmuster weder für das menschliche Auge noch für die RGB-Kamera sichtbar, da diese über einen IR-Filter verfügt, der infrarotes Licht blockt.

4.5.2 Berechnung der Disparitätswerte

In Abschnitt 4.3.3 konnte die Disparität unter Verwendung des Stereo-Kamerasystems im Normalfall dadurch ermittelt werden, dass zunächst die beiden zur Verfügung stehenden Kameras ein Bild der aktuellen Szene erfassten. In diesen Bildern wurden die korrespondierenden Punkte aufgesucht und schließlich die Disparität anhand des Unterschieds der Positionen der korrespondierenden Punkte ermittelt.

Da der Kinect-Tiefensensor mit dem Ansatz des strukturierten Lichts arbeitet, ist jedoch eine Kamera durch den Projektor ersetzt worden. Somit steht zum Zeitpunkt t nur ein aktuelles Bild \mathbf{B}_t der Szene zur Verfügung, welche mit dem Punktmuster beleuchtet ist. Um dennoch Disparitäten berechnen zu können, auf denen das mathematische Modell zur Tiefenbestimmung in Abschnitt 4.5.3 basieren wird, kann das Referenzbild \mathbf{B}_{ref} verwendet werden, welches jede Kinect ab Werk in einem internen ROM speichert. Dabei beinhaltet \mathbf{B}_{ref} das mit dem eigenen Projektor generierte Muster, welches senkrecht auf eine Referenzebene Π_{ref} mit bekanntem Referenzabstand z_{ref} projiziert (vgl. Abb. 4.12) und mit der IR-Kamera erfasst worden ist [ZSMSL07]. Somit kann die Disparität mittels der Bilder \mathbf{B}_t und \mathbf{B}_{ref} bestimmt werden. Ziel ist es, für jedes Pixel in \mathbf{B}_t die Disparität zu berechnen, so dass ein Disparitätsbild \mathbf{D}_t entsteht und für weiterführende Berechnungen genutzt werden kann. Eine Übersicht des Vorgehens ist in Abb. 4.11 dargestellt und wird nachfolgend erläutert.

Sei (u_t, v_t) die aktuell betrachtete Pixelposition, für die der Eintrag $\mathbf{D}_t(u_t, v_t)$ im Disparitätsbild bestimmt werden soll. Gesucht ist die Position (u_{ref}, v_{ref}) im Referenzbild \mathbf{B}_{ref} , für die eine Korrespondenz des Punktmusters ("speckle pattern") vorliegt, also jene Position in \mathbf{B}_{ref} , an der sich ein eindeutig identifizierbarer Lichtpunkt \mathbf{s} ("speckle") aus \mathbf{B}_t wiedererkennen lässt (vgl. Abb. 4.11(a)). Wie in Abschnitt 4.5.1 beschrieben, ist das Punktmuster unkorreliert, so dass unter Verwendung eines zweidimensionalen

¹<http://azttm.wordpress.com/2011/04/03/kinect-pattern-uncovered/>

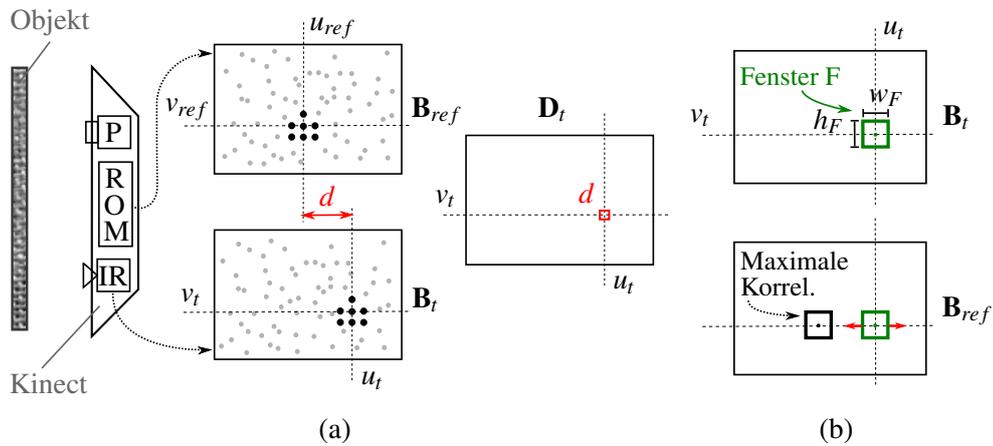


Abbildung 4.11: In (a) wird das Gesamtsystem zur Berechnung des Disparitätsbildes \mathbf{D}_t verdeutlicht. Die IR-Kamera der Kinect erfasst das aktuelle Bild \mathbf{B}_t zum Zeitpunkt t , in dem das auf ein Objekt projizierte Punktmuster zu sehen ist. Gleichzeitig steht das im ROM der Kinect vorliegende Referenztiefbild \mathbf{B}_{ref} zur Verfügung. Um den Disparitätsbildeintrag $\mathbf{D}_t(u_t, v_t)$ zu bestimmen, muss die Position des korrespondierenden Punktmusters im Referenzbild \mathbf{B}_{ref} gefunden werden. Dazu wird \mathbf{B}_{ref} mittels eines Suchfensters F (grün), das in der Umgebung der Position (u_t, v_t) platziert wird, durch die Verschiebung des Fensters abgesucht, um die Position (u_{ref}, v_{ref}) zu finden, die die maximale Korrelation mit dem Fenster in \mathbf{B}_t an der aktuellen Position (u_t, v_t) aufweist, siehe (b). Die Parameter w_F und h_F definieren die Suchfenstergröße. d ist die gesuchte Disparität, die im Disparitätsbild \mathbf{D}_t an der aktuellen Position abgespeichert wird.

Fensters F mit der Breite w_F und der Höhe h_F ein bestimmter Punkt innerhalb des Musters anhand seiner direkten Umgebung eindeutig identifiziert werden kann. Laut [Kue11, KM12] wird davon ausgegangen, dass die Kinect zur Berechnung der korrespondierenden Positionen ein Fenster der Größe $w_F = 9, h_F = 9$ verwendet. Um die korrespondierende Position zu berechnen, wird das Fenster F an der aktuellen Position (u_t, v_t) in \mathbf{B}_t zentriert. Das gleiche Fenster wird im Referenzbild \mathbf{B}_{ref} eingesetzt, um die Position in \mathbf{B}_{ref} zu finden, an der die Korrelation zwischen dem Fensterausschnitt in \mathbf{B}_t und dem Fensterausschnitt in \mathbf{B}_{ref} maximal ist. Dies ist die gesuchte Position (u_{ref}, v_{ref}) . Sie lässt sich durch die Verschiebung des Fensters in \mathbf{B}_{ref} in Subpixelgenauigkeit ermitteln (durch rote Pfeile in Abb. 4.11(b) angedeutet). Die Disparität d ist die horizontale Verschiebung zwischen den beiden korrespondierenden Positionen mit $d = u_t - u_{ref}$. Diese entspricht auch dem gesuchten Eintrag im Disparitätsbild \mathbf{D}_t an der aktuellen Position:

$$\mathbf{D}_t(u_t, v_t) = u_t - u_{ref}. \quad (4.21)$$

Wird dies für alle Positionen $u_t = 0, \dots, w_D - 1, v_t = 0, \dots, h_D - 1$ berechnet, dann steht das gesamte Disparitätsbild zur Verfügung. Dabei bezeichnen w_D und h_D die Breite und Höhe des Disparitätsbildes.

Wie erwähnt, wiederholt sich der unkorrelierte Teil des Punktmusters aufgrund der verwendeten Vervielfältigung als 3×3 Mosaik (vgl. Abschnitt 4.5.1). Somit ist auch die

maximale Disparität, die mit Hilfe der Korrelation eindeutig bestimmt werden kann, beschränkt auf etwa 1/3 der Bildbreite des IR-Kamerabildes.

4.5.3 Mathematisches Modell zur Tiefenbestimmung

Wie in Abb. 4.12 verdeutlicht, entspricht die geometrische Anordnung von IR-Kamera und Projektor dem Stereo-Normalfall mit dem Unterschied, dass der Projektor die zweite Kamera ersetzt (vgl. Abschnitt 4.3.3). Ein einzelner Lichtpunkt s ("speckle") des Punktmusters definiert einen Strahl, der vom Projektionszentrum c_P ausgeht und entlang dessen das Licht des Lichtpunktes s bei der Projektion verläuft. Der Strahl sei durch die Gerade g_s repräsentiert. Da das Punktmuster als konstant angenommen wird, ist auch die zu s gehörende Richtung des Strahls als konstant anzunehmen. Sei Π_{ref} eine Referenzebene mit bekanntem Referenzabstand z_{ref} zur Basis. Der Punkt, auf dem der Lichtpunkt s auf die Referenzebene trifft, entspricht dem Schnittpunkt von g_s und Π_{ref} und sei mit x_{ref} bezeichnet. Dieser wird auf den Punkt x'_{ref} in der Kameraebene Π_{IR} abgebildet.

Nun sei angenommen, dass in der Szene ein Objekt vorliegt, dessen Objektebene Π_{obj} eine von z_{ref} abweichende Distanz z_{obj} besitzt. Dann wird Π_{obj} von g_s im Punkt x_{obj} geschnitten und ist im Punkt x'_{obj} in der Projektionsebene der IR-Kamera zu beobachten. Da die Richtung des Projektionsstrahls konstant ist, hat die veränderte Distanz der

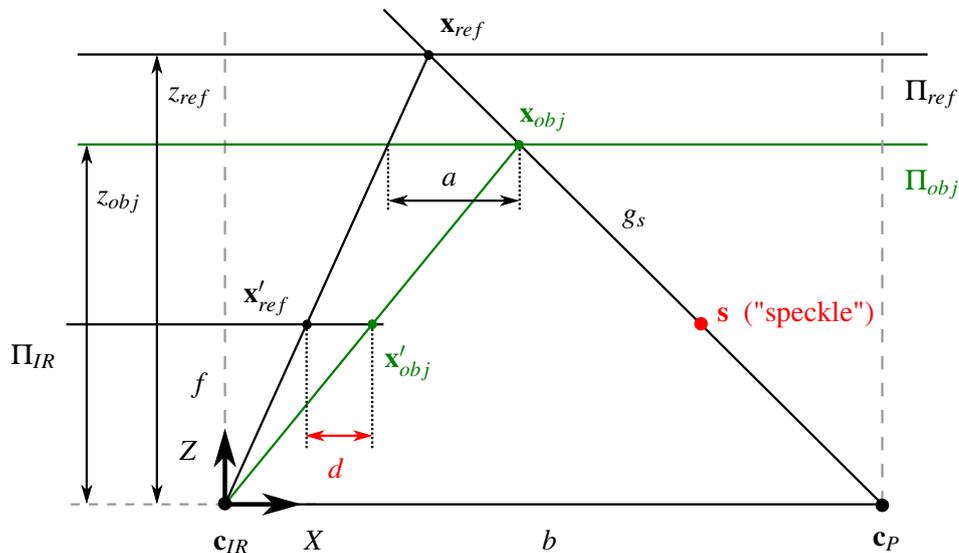


Abbildung 4.12: Geometrischer Zusammenhang bei der Berechnung des Objektabstands. Der Laserprojektor projiziert einen Lichtpunkt s entlang des Strahls, der durch die Gerade g_s repräsentiert wird, die die Referenzebene Π_{ref} im Abstand z_{ref} im Punkt x_{ref} bzw. die Objektebene Π_{obj} im Abstand z_{obj} im Punkt x_{obj} schneidet. Die Länge der Basis zwischen dem IR-Kamerazentrum c_{IR} und dem Zentrum des Projektors c_P ist mit b bezeichnet. Die Projektionsebene der Kamera Π_{IR} hat den Abstand f zum Kamerazentrum. Die Disparität d entspricht der horizontalen Abweichung des projizierten Punktes x'_{obj} vom Punkt x'_{ref} in Π_{IR} .

Objektebene gegenüber der Referenzebene eine horizontale Verschiebung des Punktes \mathbf{x}'_{obj} gegenüber dem Referenzpunkt \mathbf{x}'_{ref} in der IR-Kamerabildebene zur Folge. Diese Verschiebung ist die Disparität d . Sie lässt sich bestimmen durch $d = u'_{obj} - u'_{ref}$ mit $\mathbf{x}'_{obj} = (u'_{obj}, v'_{obj})^T$ und $\mathbf{x}'_{ref} = (u'_{ref}, v'_{ref})^T$.

Durch Anwendung der Strahlensätze lassen sich die Gleichungen

$$\frac{a}{b} = \frac{z_{ref} - z_{obj}}{z_{ref}} \quad (4.22)$$

und

$$\frac{d}{f} = \frac{a}{z_{obj}} \quad (4.23)$$

aufstellen [KE12]. Dabei ist b die Länge der Basis, f die Brennweite der IR-Kamera und a die Länge der Strecke zwischen dem Punkt \mathbf{x}_{obj} und dem Schnittpunkt zwischen der Objektebene Π_{obj} und dem Projektionsstrahl, der das Projektionszentrum \mathbf{c}_{IR} der IR-Kamera mit dem Referenzpunkt \mathbf{x}_{ref} verbindet (siehe Abb. 4.12). Durch die Umstellung der Gleichung (4.23) nach a , das Einsetzen in (4.22) und die Umformung nach z_{obj} ergibt sich die gesuchte Entfernung zur Objektebene in Abhängigkeit der Disparität und der übrigen, konstanten Parameter [KE12]:

$$\frac{dz_{obj}}{bf} = \frac{z_{ref} - z_{obj}}{z_{ref}} \Leftrightarrow \quad (4.24)$$

$$z_{obj} = \frac{z_{ref}}{\left(1 + \frac{z_{ref}}{bf} d\right)}. \quad (4.25)$$

Es sei angenommen, dass die konstanten Parameter f, b und z_{ref} durch eine entsprechende Kalibrierung bekannt sind (siehe Abschnitt 4.5.5).

Ist $\mathbf{D}(u, v)$ das aktuelle Disparitätsbild, welches für jede Pixelposition (u, v) die ausgewertete Disparität zur Verfügung stellt (Details siehe Abschnitt 4.5.2), dann kann ein entsprechendes Tiefenbild $\mathbf{T}(u, v)$ durch pixelweise Auswertung der Gleichung (4.25) unter Verwendung von $\mathbf{D}(u, v)$ berechnet werden.

4.5.4 Umrechnung der Tiefendaten in 3D-Punktpositionen

Gesucht sind die 3D-Koordinaten des Punktes $\mathbf{x}_{obj} = (x_{obj}, y_{obj}, z_{obj})^T$. Ist die Tiefe z_{obj} entsprechend der Gleichung (4.25) bekannt, so lassen sich x_{obj} und y_{obj} berechnen durch [KE12]

$$x_{obj} = \frac{z_{obj}}{f} (u'_{obj} - u_0 + \Delta_u), \quad (4.26)$$

$$y_{obj} = \frac{z_{obj}}{f} (v'_{obj} - v_0 + \Delta_v), \quad (4.27)$$

wobei $\mathbf{x}'_{obj} = (u'_{obj}, v'_{obj})^T$ den in der Bildebene Π_{IR} beobachtbaren Punkt, $(u_0, v_0)^T$ den Hauptpunkt der IR-Kamera und Δ_u sowie Δ_v Terme der Linsenverzerrungskorrektur bezeichnen.

Liegt ein aktuelles Tiefenbild \mathbf{T} der Kinect vor, so kann für jedes Pixel $\mathbf{T}(u, v)$ ein 3D-Punkt und somit für das gesamte Tiefenbild eine 3D-Punktwolke \mathbf{W} berechnet werden. Seien $\mathbf{w}^0, \dots, \mathbf{w}^{n-1}$ die Punkte der Punktwolke \mathbf{W} und $n = w_T h_T$ die Anzahl der Punkte, wobei w_T bzw. h_T der Breite bzw. Höhe des Tiefenbildes entsprechen. Sei $\mathcal{F}_{Pos2Index}(u, v)$ eine Funktion, die für die Pixelposition (u, v) den entsprechenden Index i des Wolkenpunktes \mathbf{w}^i mit

$$i = \mathcal{F}_{Pos2Index}(u, v) = w_T v + u \quad (4.28)$$

bestimmt. Dann können die Wolkenpunkte $\mathbf{w}^i = (w_x^i, w_y^i, w_z^i)^T$ aus dem Tiefenbild \mathbf{T} für alle Pixelpositionen (u, v) , $u = 0, \dots, w_T - 1$, $v = 0, \dots, h_T - 1$, wie folgt berechnet werden:

$$w_x^{\mathcal{F}_{Pos2Index}(u,v)} = \frac{T(u, v)}{f} (u - u_0 + \Delta_u), \quad (4.29)$$

$$w_y^{\mathcal{F}_{Pos2Index}(u,v)} = \frac{T(u, v)}{f} (v - v_0 + \Delta_v), \quad (4.30)$$

$$w_z^{\mathcal{F}_{Pos2Index}(u,v)} = T(u, v). \quad (4.31)$$

4.5.5 Kalibrierung der Tiefenberechnungsparameter

In Abschnitt 4.5.3 wurde die Berechnung der gesuchten Tiefe z_{obj} bei gegebener Disparität d mit Hilfe der Gleichung (4.25) beschrieben. Dabei wurden die konstanten Parameter f, b und z_{ref} als bekannt vorausgesetzt. Diese müssen also vor der Anwendung kalibriert werden.

Die Brennweite f kann mit Hilfe der Kamerakalibrierung, wie z.B. in Abschnitt 2.4 beschrieben, bestimmt werden. Neben der unbekannt Basislänge b und der Distanz zur Referenzebene z_{ref} kommt hinzu, dass die Kinect nicht die realen Disparitätswerte (gegeben in Subpixelgenauigkeit) überträgt, sondern eine unbekannt Normierung dieser Werte durchführt und sie als 11-Bit Ganzzahl im Wertebereich zwischen 0 und 2047 überträgt. Der vermutete Grund für diese Normierung ist eine Bandbreitenlimitierung, die bei der Übertragung der tatsächlichen Disparitäten, z.B. als Gleitkommazahlen mit je 32 Bits, überschritten werden könnte. Somit müssen zusätzlich ein Umrechnungsmodell sowie dessen Parameter geschätzt und kalibriert werden. Üblicherweise wird in der Literatur ein lineares Umrechnungsmodell angenommen [KE12, Kue11, KM12]:

$$d = \alpha d_n + \beta. \quad (4.32)$$

Dabei ist $d_n \in [0, 2047]$ die normalisierte und d die tatsächliche Disparität, und α sowie β sind die zu schätzenden Modellparameter.

Durch das Einsetzen der Gleichung (4.32) in die Gleichung (4.25) ergibt sich

$$z_{obj} = \frac{z_{ref}}{\left(1 + \frac{z_{ref}}{bf} (\alpha d_n + \beta)\right)}. \quad (4.33)$$

Durch die Bildung des Kehrwerts und entsprechende weitere Äquivalenzumformungen folgt [KE12]:

$$\frac{1}{z_{obj}} = \left(\frac{\alpha}{fb}\right) d_n + \left(\frac{1}{z_{ref}} + \frac{\beta}{fb}\right). \quad (4.34)$$

Die Gleichung (4.34) drückt den linearen Zusammenhang zwischen der inversen Distanz zum Objekt und der von der Kinect zur Verfügung gestellten korrespondierenden normalisierten Disparität d_n aus. Liegen für m Objektpunkte mit bekannter Entfernung z_{obj}^i , $i = 0, \dots, m-1$, die entsprechenden, von der Kinect zur Verfügung gestellten Disparitäten d_n^i vor, so lassen sich die gesuchten Parameter α, β, b und z_{ref} der linearen Gleichung (4.34) z.B. durch Minimierung der Zielfunktion

$$\mathcal{F}(f, b, \alpha, \beta, z_{ref}, d_n^i) = \sum_{i=0}^{m-1} \left\| \left(\frac{\alpha}{fb} \right) d_n^i + \left(\frac{1}{z_{ref}} + \frac{\beta}{fb} \right) - \frac{1}{z_{obj}^i} \right\|^2 \quad (4.35)$$

bestimmen [KE12].

In Arbeiten wie z.B. [SJP11, HKH12] sowie (nach geeigneter Variablensubstitution) in [HKH11], wird gegenüber Gleichung (4.25) die folgende deutlich einfachere Relation zwischen der Tiefe z_{obj} und der Disparität d verwendet:

$$z_{obj} = \frac{1}{c_0 d + c_1}. \quad (4.36)$$

Da die Variablen f, z_{ref} , und b in Gleichung (4.25) unbekannt sind, jedoch als konstant angenommen werden können, kann Gleichung (4.25) in Gleichung (4.36) überführt werden durch

$$z_{obj} = \frac{z_{ref}}{\left(1 + \frac{z_{ref}}{bf} d\right)} = \frac{1}{\frac{1}{z_{ref}} \left(1 + \frac{z_{ref}}{bf} d\right)} = \frac{1}{\left(\frac{1}{z_{ref}}\right) + \left(\frac{1}{bf}\right) d} = \frac{1}{c_0 + c_1 d}, \quad (4.37)$$

wobei die Substitutionen $c_0 = \frac{1}{z_{ref}}$ und $c_1 = \frac{1}{bf}$ durchgeführt worden sind. Wird weiterhin die Disparität d durch die normierte Disparität d_n ersetzt (siehe Gleichung (4.32)), so bleibt die Struktur der Gleichung aufgrund des linearen Zusammenhangs zwischen d und d_n erhalten:

$$z_{obj} = \frac{1}{c_0 + c_1(\alpha d_n + \beta)} = \frac{1}{(c_0 + c_1\beta) + c_1\alpha d_n} = \frac{1}{\gamma_0 + \gamma_1 d_n}. \quad (4.38)$$

Dabei wurden die Substitutionen $\gamma_0 = c_0 + c_1\beta$ und $\gamma_1 = c_1\alpha$ durchgeführt. Somit müssen nur die beiden Variablen γ_0 und γ_1 , die zu den intrinsischen Parametern des Tiefensensors gezählt werden können, wie zuvor beschrieben, z.B. mit Hilfe einer Menge von Punkten mit bekanntem Abstand und den dazugehörigen Disparitäten der Kinect, bestimmt werden. Auf dieser Grundidee basieren auch die Kalibrierungen der Parameter des Tiefensensors in den Arbeiten [KE12, SJP11, HKH12, HKH11].

4.5.6 Tiefenauflösung

Wie in Abschnitt 4.5.5 erwähnt, werden 11 Bits für die Übertragung eines Disparitätswertes verwendet. Von diesen 11 Bits wird jedoch 1 Bit für die Information aufgewendet, um zu kodieren, ob der übertragene Wert gültig ist [KE12]. Ungültige Werte können z.B. durch Störungen des aufzufindenden Punktmusters entstehen, beispielsweise durch Reflexionen an der Objektoberfläche, und zur Folge haben, dass keine

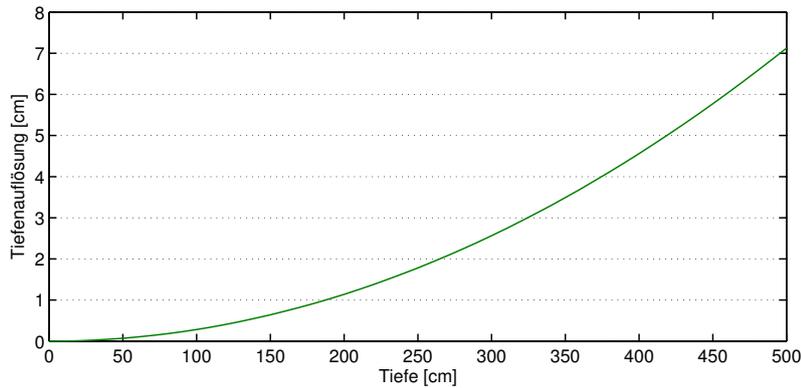


Abbildung 4.13: Verlauf der Tiefenauflösung der Kinect in Abhängigkeit der Tiefe. Auswertung von $\Delta z = \left(\frac{\alpha}{fb}\right) z^2$ mit $\frac{\alpha}{fb} = 2.85 \cdot 10^{-6} \text{ mm}^{-1}$ im Wertebereich bis 5 m.

Korrespondenz im Punktmuster wiedergefunden werden kann (vgl. Abschnitt 4.5.2). Somit verbleiben 10 Bits für die Kodierung der normalisierten Disparität, und das Disparitätsbild kann folglich nur maximal 1024 unterschiedliche Disparitätsstufen aufweisen. Die Abschätzung der Tiefenauflösung bzw. der Schrittweite zwischen zwei aufeinanderfolgenden Tiefenwerten ist in [KE12] untersucht worden und durch die Gleichung

$$\Delta z = \left(\frac{\alpha}{fb}\right) z^2 \quad (4.39)$$

gegeben, wobei f und b die Brennweite und die Länge der Basis und α einen der Umrechnungsparameter der normalisierten Disparität darstellen (siehe Gleichung (4.32)). Das bedeutet, dass die Tiefenauflösung eine von der Tiefe quadratisch abhängige Funktion ist, die mit zunehmender Tiefe abnimmt (Schrittweite nimmt mit der Tiefe quadratisch zu). Dieser Zusammenhang ist bereits für das Stereo-Kamerasystem beobachtet worden (vgl. Abschnitt 4.3.3).

In Abb. 4.13 ist der Verlauf der Tiefenauflösung in Abhängigkeit von der Tiefe entsprechend der Gleichung (4.39) dargestellt, wobei der in [KE12] für die Kinect ermittelte Wert $\frac{\alpha}{fb} = 2.85 \cdot 10^{-6} \text{ mm}^{-1}$ verwendet worden ist. Gemäß dieser Auswertung besitzt die Kinect eine geschätzte Tiefenauflösung von 0.28 cm in einer Distanz von 1 m und von etwa 2 cm in 2.65 m Entfernung.

4.5.7 Datenzugriff und SDKs

Um mit dem PC auf die Kinect zugreifen zu können gibt es drei verschiedene Möglichkeiten. Kurz nach Verkaufsstart wurde die Bibliothek "libfreenect" von der "OpenKinect-Community"¹ inklusive Treiber zur Verfügung gestellt. Diese bietet einen nicht-offiziellen ("gehackten") Zugriff auf die Kinect-Daten, insbesondere auf die Disparitätsdaten. Die Möglichkeit, auf diese Daten zugreifen zu können und die Kinect somit als neuartigen und kostengünstigen Tiefensensor zu verwenden, hat in der

¹<http://openkinect.org/>

Forschungsgemeinschaft großes Interesse hervorgerufen, so dass auch wissenschaftliche Veröffentlichungen wie z.B. [KE12, SJP11, HKH12, HKH11] erschienen sind, die sich die Umrechnung der (normierten) Disparitätsdaten sowie die dafür notwendige Kalibrierung zum Gegenstand gemacht haben (siehe auch Abschnitt 4.5.5).

Einige Zeit später wurde vom OpenNI-Framework [Ope10] in Zusammenarbeit mit der Firma PrimeSense¹, die die Technologie des Tiefensensors entwickelte, ein offizielles Software Development Kit (SDK) veröffentlicht. Da dieses SDK offiziell unterstützt wird, stehen den Entwicklern, im Gegensatz zur OpenKinect-Community, alle geheimgehaltenen technischen Details sowie Informationen zur Kalibrierung der Kinect ab Werk (z.B. Informationen zur Lage und Entfernung der gespeicherten Referenzebene, vgl. Abschnitt 4.5.2 und 4.5.3) zur Verfügung. Das Framework kapselt die Umrechnung der Disparitätsdaten in Distanzwerte und stellt nur die Distanzwerte in Millimetern über entsprechende Schnittstellen zur Verfügung. Somit ist eine explizite Kalibrierung der Parameter für die Umrechnungsformel (siehe Gleichung (4.33)) nicht mehr notwendig.

Aufgrund des großen Interesses an dem SDK für die Kinect hat zuletzt auch Microsoft selbst das "Kinect für Windows-SDK"² veröffentlicht und bietet, genau wie das OpenNI-Framework, nur den direkten Zugriff auf die umgerechneten Distanzwerte in Millimetern, jedoch nicht auf die normierten Disparitätswerte.

4.6 Potential und Probleme kostengünstiger Tiefensensoren

Die Tabelle 4.1 fasst Stärken und Schwächen der in diesem Kapitel erläuterten Tiefensensoren (Laserscanner, PMD-Kamera, Stereo-Kamera, Kinect) zusammen. Eine hohe Genauigkeit bei großen Messdistanzen sowie die hohe Auflösung sind deutliche Stärken von Laserscannern. Demgegenüber stehen jedoch die sehr hohen Kosten (bis 120000 EUR), die geringe Messwiederholrate (je nach Punktwolkendichte deutlich unter 1 Hz) sowie das relativ hohe Gewicht (typischerweise ca. 10 kg). Sind hingegen seitens des Anwendungsbereiches keine großen Messdistanzen erforderlich, wie z.B. bei typischen Innenraumanwendungen, so können auch die anderen Tiefensensoren eingesetzt werden. Im Gegensatz zu Laserscannern bieten PMD-Kameras hohe Messwiederholraten und haben eine kompakte Bauform. Zwar liegen sie im mittleren Preissegment (ca. 4000 EUR), allerdings ist der größte Nachteil die geringe Punktwolkendichte, bedingt durch die geringe Auflösung des Sensors. Aufgrund der Möglichkeit des Einsatzes handelsüblicher, günstiger Kameras (z.B. der Kompaktkameras aus Abschnitt 4.3.3) können Stereo-Kamerasysteme im niedrigen Preissegment realisiert werden (ca. 200 EUR), die aufgrund der hohen Auflösung eine hohe Punktwolkendichte bieten. Aufgrund des quadratischen Zusammenhangs zwischen der Messdistanz und Tiefenauflösung (sehr genau im Nahbereich, ungenau bei größeren Distanzen) wird die Genauigkeit der Stereo-Systeme in der Tabelle als mittelmäßig eingestuft. Es sei jedoch angemerkt, dass die benötigte Genauigkeit durch eine geeignete Wahl

¹<http://www.primesense.com>

²<http://www.microsoft.com/en-us/kinectforwindows/>

Eigenschaft	Laserscanner	PMD-Kamera	Stereo-Kamera	Kinect
Kosten	-	∅	+	+
Punktwolkendichte	+	-	+	∅
Genauigkeit	+	∅	∅	∅
Messwiederholrate	-	+	∅	+
Umgebungsanforderung (Beleuchtung, Textur)	+	+	-	+
Größe / Gewicht	-	+	+	+
Reichweite	+	∅	∅	∅

Tabelle 4.1: Übersicht der Stärken und Schwächen verschiedener Tiefensensoren. Legende: (-) Schwäche, (∅) Mittelmaß, (+) Stärke.

der Parameter (Brennweite, Basislänge, Auflösung usw.) flexibel konfigurierbar ist. Der größte Nachteil bei Stereo-Systemen ist die Anforderung an die Umgebung bzw. die Szene, da diese hinreichend beleuchtet und ausreichend Merkmale für die Korrespondenzfindung aufweisen muss (z.B. durch texturierte Oberflächen). Dies ist in vielen Fällen nicht gegeben, so dass Stereo-Systeme nur bedingt eingesetzt werden können. Dieser Nachteil kann durch Ansätze mit strukturiertem Licht, bei dem die nötigen Merkmale aktiv erzeugt werden, umgangen werden. Insbesondere durch die extrem kostengünstige Kinect (ca. 100 EUR), die mit einem kontrastreichen, infraroten Laserpunktmuster arbeitet, eine mittlere Punktwolkenauflösung bietet (7 mal mehr Wolkenpunkte im Vergleich mit PMD-Kameras), eine vergleichbare Tiefenauflösung wie Stereo-Kamerasysteme aufweist, ein geringes Gewicht sowie eine kompakte Bauform besitzt und zudem einen Großteil des Rechenaufwands bereits auf der eigenen Hardware durchführt, wurde ein Tiefensensor zur Verfügung gestellt, dessen Potential in vielen Anwendungsbereichen und zahlreichen Publikationen erkannt wurde. Das große Interesse wird auch an einer Vielzahl von in Verbindung mit Konferenzen gehaltenen Workshops, wie z.B. [ICR13, WDI12, Per12] deutlich, in denen die Kinect und ihre Anwendungsmöglichkeiten thematische Schwerpunkte darstellen.

Der geringe Preis, der den Schlüsselfaktor des Erfolgs der Kinect in den unterschiedlichen Anwendungsbereichen darstellt, hat jedoch auch den Nachteil, dass keine hochwertigen Hardwarekomponenten verbaut werden können, und als Folge dessen fehlerhafte und instabile Messungen auftreten können, die unterschiedliche Ursachen haben. Die nähere Untersuchung dieser und weiterer Probleme erfolgt in Kapitel 5.

5

Analyse und Korrektur von Messfehlern des Tiefensensors Kinect

Gegenstand dieses Kapitels ist die Analyse und Korrektur von Messfehlern des kostengünstigen Tiefensensors Kinect. Die dargelegte Vorgehensweise stellt eine allgemeine Vorgehensweise dar, deren Prinzip auf andere Tiefensensoren übertragbar ist, insbesondere auf solche, die auf strukturiertem Licht basieren.

Die korrekte Arbeitsweise vieler Anwendungen aus dem Bereich des Visual Computing hängt von möglichst zuverlässigen und genauen 3D-Tiefenmessdaten ab. Der 3D-Tiefensensor Kinect stellt ein komplexes System aus zahlreichen Einzelkomponenten dar, die ihrerseits fehlerhaft sein und eine Auswirkung auf die Tiefenmessung haben können. Aufgrund des komplexen Zusammenspiels der Einzelkomponenten ist keine triviale Abschätzung des Messfehlers möglich. In entsprechenden Experimenten, in denen planare Oberflächen betrachtet werden, wird ein Messfehler ermittelt und beschrieben. Von diesen Erkenntnissen ausgehend, werden Korrekturverfahren vorgeschlagen, die auf empirischen Messdaten planarer Objekte beruhen und eine effiziente, auf Look-Up Tabellen basierende Korrektur erlauben. Die Kinect stellt die Messdaten online zur Verfügung. Um auch die Qualität der Messdaten online beurteilen zu können, wird des Weiteren eine Methode zur Online-Ermittlung von Referenzmessdaten vorgeschlagen, die auf Kalibrierobjekten basiert und sich vollständig mit der in der Kinect integrierten Hardware durchführen lässt.

Aus dem Kontext anderer Sensoren ist bekannt, dass Umwelteinflüsse, wie z.B. die Temperatur, Auswirkungen auf die Messdaten haben können, die jedoch bislang nicht für den Kinect-Sensor untersucht worden sind. Insbesondere die Komplexität des Sensors, der niedrige Kaufpreis und die damit verbundene Verarbeitungsqualität rechtfertigen die Vermutung, dass die einzelnen Komponenten und somit die Messungen der Kinect ebenfalls von Umwelteinflüssen abhängig sind, was experimentell untersucht wird. Aufgrund der Erkenntnisse bzgl. der Umwelteinflüsse wird zusätzlich geprüft, ob der zu Beginn vorgeschlagene Korrekturansatz gegenüber Umwelteinflüssen robust ist.

Der Abschnitt 5.1 fasst die für dieses Kapitel relevanten Problemstellungen in einer allgemeinen Form zusammen, die nicht nur auf den Kinect-Sensor beschränkt zu betrachten sind. Zudem präsentiert es die Lösungsansätze im Einzelnen und stellt dabei einen Bezug zur Struktur des gesamten Kapitels 5 her. Der komplexe, sensorortsabhängige Messfehler wird in Abschnitt 5.2 untersucht sowie korrigiert und die Kor-

rekturleistung evaluiert. Der Ansatz zur Online-Berechnung von Referenzmessdaten ist Gegenstand des Abschnitts 5.3. Die Auswirkungen der Umwelteinflüsse werden in Abschnitt 5.4 und schließlich die Tragfähigkeit des in Abschnitt 5.2 erarbeiteten Korrekturverfahrens angesichts der Erkenntnisse über Umwelteinflüsse in Abschnitt 5.5 untersucht.

5.1 Problemstellungen und Lösungsansätze

Dieser Abschnitt präsentiert zunächst wesentliche Problemstellungen, die sich bei der Analyse und Korrektur von Messfehlern von Sensoren im Allgemeinen ergeben. Daran schließt sich eine Übersicht von Ansätzen zur Lösung der Probleme für den Kinect-Sensor an, die den Schwerpunkt der weiteren Abschnitte des Kapitels bilden.

5.1.1 Problemstellungen

Im Allgemeinen ist davon auszugehen, dass reale Tiefensensoren nicht ideal sein können, sondern immer ein gewisser Messfehler vorhanden ist. Eine Ursache für Messfehler ist z.B. das Sensorrauschen, durch das die Messung einer zufallsbedingten Verfälschung unterliegt. Zwar lässt sich dieses Problem nicht verhindern, doch ist es notwendig, das Sensorrauschen zu berücksichtigen. Ein weiterer Grund für Messfehler ist die Fertigungsungenauigkeit. Diese ist besonders bei Messsystemen relevant, die sich aus der Kombination mehrerer Sensoren ergeben. Hinzu kommt bei solchen Systemen, dass sich Fehler der Einzelsensoren, die etwa aufgrund minderer Qualität der verarbeiteten Materialien (aus Gründen der Wirtschaftlichkeit) oder aufgrund von Fertigungstoleranzen vorhanden sind, im Gesamtsystem im schlimmsten Fall noch verstärken können. Ein auf strukturiertem Licht basierender Tiefensensor wie die in diesem Kapitel betrachtete Kinect stellt ein solches komplexes Messsystem dar, das aus zahlreichen Einzelkomponenten (Laserprojektor, Kamera, usw.) besteht.

Vor diesem Hintergrund besteht die erste Problemstellung in der *Ermittlung des Messfehlers*. Die Aufgabe hierbei ist, Zusammenhänge zwischen dem zu erfassenden Datum und dem gemessenen Datum zu ermitteln. Ein solcher Zusammenhang kann in der Abschätzung der Abweichung zwischen einem Ist- und Soll-Datum (\hat{d} und d) in der Form eines gewählten Fehlermaßes E in Form einer Fehlerschranke ε bestehen: $E(\hat{d}, d) < \varepsilon$. Ideal wäre eine Funktion \mathcal{F} , die den Zusammenhang zwischen einem gegebenen und gemessenen Datum präzise beschreibt: $\hat{d} = \mathcal{F}(d)$. Aufgrund der aufgeführten Gegebenheiten wird es jedoch in der Regel allenfalls möglich sein, eine Approximation einer derartigen Funktion angeben zu können.

Ein Ansatz zur Herleitung einer messfehlerbeschreibenden Funktion besteht in der physikbasierten, mathematischen Modellierung des Sensorverhaltens. Diese wird gerade bei komplexen Sensoren schwierig sein. Eine Alternative sind mathematische Ersatzmodelle, die empirisch aus Daten gewonnen werden. Deren Erstellung, aber auch die Kalibrierung physikbasierter Modelle, führt auf eine weitere Problemstellung, die *Gewinnung von Referenzdaten zur Messfehleranalyse*.

Ein spezieller Aspekt der Messfehlerermittlung ist die Abhängigkeit des Messfeh-

lers von Umwelteinflüssen. So kann das Messverhalten beispielsweise von der Umgebungstemperatur abhängig sein, was für die Kinect im Rahmen dieser Dissertation deutlich herausgestellt werden konnte. Dies führt auf ein drittes Problem, die *Bestimmung des Einflusses von Umwelteinflüssen auf den Messfehler*. Dies wiederum stellt eine Erweiterung der ersten Problemstellung dar, bei der die zu ermittelnde Abhängigkeit von weiteren Parametern, z.B. der Umgebungstemperatur, abhängt.

Die Messfehleranalyse erlaubt die Beurteilung der Tragfähigkeit von Schlussfolgerungen aus Messungen. Ein weitergehender Schritt ist die *Korrektur von Messfehlern*, welche die vierte behandelte Problemstellung darstellt. Hierbei geht es um die Ermittlung einer Funktion, die einem Messwert den ursprünglichen Datenwert zuordnet. Diese Funktion ist also invers zu der erwähnten Fehlerfunktion der ersten Problemstellung. Sofern die Fehlerkorrektur das Ziel der Fehleranalyse ist, könnte es zweckmäßig sein, in der ersten Problemstellung gleich den inversen Zusammenhang zu approximieren.

5.1.2 Lösungsansätze

Die gewählte Methode zur Bearbeitung der geschilderten Probleme ist die experimentelle Analyse.

Zur *Ermittlung des Messfehlers* werden die in Abschnitt 5.2 beschriebenen Experimente durchgeführt, die Artefakte in den Sensordaten verdeutlichen. Daran schließt sich eine quantitative experimentelle Untersuchung zur Ermittlung von Messfehlern anhand eines einfachen Eingabeszenarios an, dessen geometrische Gegebenheiten vollständig bekannt sind. Der dabei erkannte Messfehler weist eine Ortsabhängigkeit im aufgenommenen Tiefenbild auf, die quantitativ erfasst wird.

Darauf basierend folgt ein Ansatz zur *Korrektur dieses Messfehlers*. Er besteht in der Angabe einer tabellarisch beschriebenen Korrekturfunktion, die experimentell erstellt wird. Dieser Ansatz wird auf zweierlei Weise konkretisiert. Die Wirkung der resultierenden Korrekturverfahren wird an dem einfachen Eingabeszenario und an weiteren, komplexeren Testszenarien überprüft.

Abschnitt 5.3 präsentiert einen Ansatz, mit dem das Problem der *Gewinnung von Referenzdaten* angegangen wird. Er beruht auf planaren Kalibrierobjekten und benötigt lediglich die bereits in der Kinect integrierte Hardware. Mit dem beschriebenen Ansatz können Referenzmessdaten während der laufenden Anwendung (online) berechnet werden, so dass eine interaktive Überprüfung der Messgenauigkeit des Tiefensensors ermöglicht wird. Der Beschreibung der Referenzdatengewinnung folgt eine Genauigkeitsanalyse, bei der auch das Rauschen berücksichtigt wird.

Die Lösung der Fragestellung, ob und inwiefern einzelne Komponenten und somit der gesamte Tiefensensor von *Umwelteinflüssen* abhängen, wird in einer Reihe von Experimenten in Abschnitt 5.4 beantwortet. Dabei werden verschiedene Möglichkeiten der kontrollierten Manipulation der Umweltbedingungen erarbeitet und Methoden zur Beurteilung der Auswirkungen vorgeschlagen. Des Weiteren werden Zusammenhänge zwischen den Umwelteinflüssen auf einzelne Komponenten und den Auswirkungen auf den resultierenden Tiefenmessfehler experimentell herausgestellt und verdeutlicht.

Aufgrund der Erkenntnisse aus Abschnitt 5.4 ist es notwendig zu überprüfen, inwiefern der in Abschnitt 5.2 vorgeschlagene Korrekturansatz temperaturabhängig ist. Die Beschreibung und Auswertung der dazu geeigneten Experimente sind Gegenstand des Abschnitts 5.5.

5.2 Komplexer sensorortsabhängiger Messfehler

Bei der Betrachtung des Tiefenbildes der Kinect, in dem eine komplexe Szene beobachtet wird, fallen Messfehler, die von der Position im Tiefenbild abhängen, nicht auf. Besteht die beobachtete Szene jedoch z.B. aus einem großen planaren Objekt, wie z.B. einem Tisch oder einer Wand, dann wird deutlich, dass die Verteilung der gemessenen Distanzen nicht zu dem realen Objekt passt. Diese Beobachtung wird umso deutlicher, je paralleler das Objekt zur Kinect ausgerichtet wird. Im dem speziellen Fall, dass die optische Achse der IR-Kamera senkrecht auf das planare Objekt trifft, ist zu erwarten, dass die von der Kinect ermittelten Tiefenwerte (Z-Werte), bis auf ein Rauschen, identisch sind. Dies ist jedoch nicht der Fall. Zu dem beschriebenen Fehler kommt hinzu, dass das von der Kinect gelieferte Tiefenbild standardmäßig eine variable Anzahl von senkrecht verlaufenden Artefakten aufweist.

Bevor im Folgenden auf den Fehler beim Messen von planaren Oberflächen genauer eingegangen wird, soll kurz auf die senkrechten Artefakte und ihre Beseitigung Bezug genommen werden.

5.2.1 Senkrechte Artefakte im Tiefenbild

Die senkrechten Artefakte, die als equidistante, sprunghafte Kanten im Tiefenbild erkennbar sind, teilen das Bild in mehrere sichtbare Streifensegmente ein (siehe Abb. 5.1). Die beobachtete Anzahl dieser sprunghaften Kanten variiert zwischen 0 und 27 und ist abhängig von der beobachteten Szene. Die Anzahl ändert sich mit einer gewissen zeitlichen Trägheit (schrittweise Anpassung ca. alle 60 Frames). Nähere Informationen zur automatischen Bestimmung der Anzahl konnten nicht ermittelt werden (Anfragen bei Entwicklern des OpenNI-Frameworks [Ope10] wurden mit dem Hinweis auf Geheimhaltung solcher Implementierungsdetails abgewiesen).

Die beschriebenen Artefakte sind auch in Tiefenbildern, die z.B. in den Arbeiten [HKH12, SJP11] abgebildet sind, vorzufinden, jedoch wird dort nicht auf diese eingegangen. Da die Anzahl und Position der Streifen automatisch gewählt werden, führt dies zwangsläufig zu Verfälschungen bei potentiellen Korrekturansätzen, da das Streifenmuster weder konstant noch vorhersagbar ist. Durch die Manipulation des nicht kommentierten Eintrags "GMCMModel" auf den Wert Null in der Konfigurationsdatei "GlobalDefaultsKinect.ini" des Prime-Sense Treibers ist es gelungen, die Artefaktbildung zu unterdrücken und somit eine szenenunabhängige Stabilität des Tiefenbildes zu erreichen. Rechts in Abb. 5.1 ist das entsprechende Tiefenbild nach der Anpassung der Konfigurationsdatei gezeigt. In den nachfolgenden Abschnitten werden nur Tiefenbilder ohne die beschriebenen Artefakte genutzt.

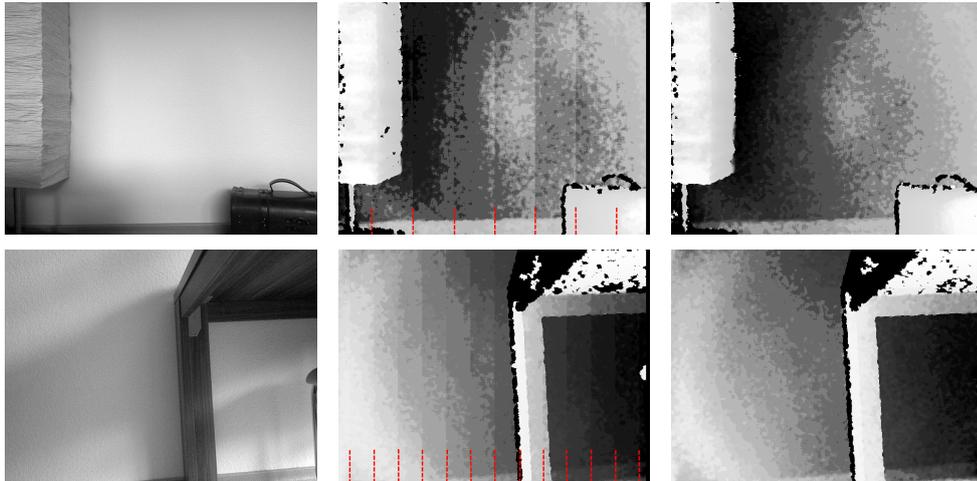


Abbildung 5.1: *Abgebildet sind zwei Beispielszenen, in denen eine unterschiedliche Anzahl an Streifensegmenten zu erkennen ist (mittig). Die sprunghaften Kanten zwischen den Streifen sind durch rote Linien angedeutet. Das Tiefenbild derselben Szene nach Beseitigung der Artefaktbildung ist jeweils rechts dargestellt.*

5.2.2 Fehler beim Messen planarer Oberflächen

Der am Anfang von Abschnitt 5.2 erwähnte Fehler beim Messen von planaren Objekten soll im Folgenden näher untersucht werden. Um die Auswirkung des Fehlers zu verdeutlichen, wurde die Kinect zunächst flach auf einen Tisch gelegt und in Richtung der Zimmerdecke ausgerichtet. Anschließend ist ihre Lage so lange angepasst worden, bis der Tiefensensor möglichst parallel zur Zimmerdecke ausgerichtet war. Um die Ausrichtung zu beurteilen, wurde die Standardabweichung aller Tiefenwerte im Tiefenbild online ausgewertet und die Ausrichtung bei der geringsten gemessenen Standardabweichung von $\sigma = 8.8$ mm fixiert. Die mittlere gemessene Entfernung (Z-Wert) zur Decke lag bei 1666.1 mm. Zu einem späteren Zeitpunkt wurde zusätzlich die Schwankung der Standardabweichung über die Zeit hinweg untersucht (1000 aufeinanderfolgende Frames). Die ermittelten Werte lagen zwischen 8.61 und 9.08 mm mit einer Schwankung von $\sigma_{Zeit} = 0.087$ mm.

Die Auswertung der für das beschriebene Szenario erfassten Tiefenwerte ist in Abb. 5.2 dargestellt. Trotz optimaler Ausrichtung sind insbesondere in Abb. 5.2(c) deutliche Abweichungen von der zugrundeliegenden planaren Oberfläche zu erkennen. Die Verteilung der Abweichungen weist eine komplexe Struktur auf, die vom Ort im Tiefenbild des Sensors abhängt. Anstatt wie im Idealfall nur einen einzigen Eintrag zu besitzen, weist das dazugehörige, logarithmisch skalierte Histogramm insgesamt 13 unterschiedliche gemessene Tiefenwerte im Bereich zwischen 1626 und 1723 mm auf. Dies entspricht Abweichungen zur mittleren gemessenen Entfernung im Bereich von -40 bis 57 mm. Dieses Problem wurde im Rahmen dieser Dissertation unabhängig von den Arbeiten [HKH12, SJP11] festgestellt, wird dort jedoch auch beobachtet. Die dort verwendeten Korrekturansätze ignorieren jedoch die Artefakte im Tiefenbild (vgl.

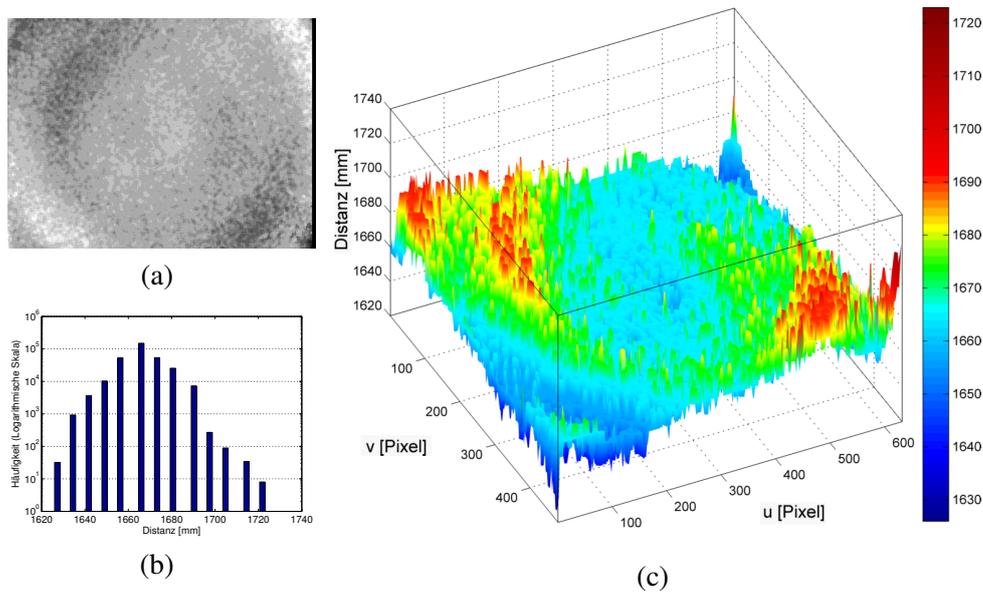


Abbildung 5.2: Ergebnisse der Tiefenmessung der Kinect, die frontal auf ein ca. 1.7 m entferntes planares Objekt gerichtet wurde (Zimmerdecke). In (a) ist das entsprechende Tiefenbild und in (b) das Histogramm (logarithmisch skaliert) der ermittelten Tiefenwerte abgebildet. Eine dreidimensionale Ansicht der Tiefenwerte ist in (c) dargestellt, in der insbesondere starke Abweichungen in den Bildecken zu erkennen sind. Die Farbskala ist in Millimetern angegeben.

Abschnitt 5.2.1) und basieren direkt auf den Disparitätswerten, die bei dem in dieser Dissertation genutzten "OpenNI-Framework" [Ope10] sowie beim "Microsoft Kinect for Windows SDK" [WA12] nicht mehr zur Verfügung gestellt werden, sondern bei denen die Umrechnung als Black-Box gekapselt und nur die Distanzwerte in Millimetern geliefert werden (vgl. Abschnitt 4.5.7).

Das im Folgenden vorgeschlagene Verfahren ist in der Lage, den beschriebenen Messfehler direkt auf den Tiefendaten basierend zu korrigieren und dabei keinerlei technische Hilfsmittel außer einem planaren Objekt (z.B. einer Wand oder der Zimmerdecke) zu benötigen.

5.2.3 Planaritätsbasierter Korrekturansatz

Um den in Abschnitt 5.2.2 beobachteten komplexen, sensorortsabhängigen Messfehler zu korrigieren, wird eine zweidimensionale Look-Up Tabelle (LUT) berechnet, die als "Delta-Plane" bezeichnet wird und die für jede Position des Tiefenbildes einen Korrekturwert speichert. Die Verwendung einer LUT für die Korrektur hat nicht nur Vorteile bzgl. des Berechnungsaufwands, sondern ermöglicht auch die Handhabung beliebig komplexer Fehlermuster. Würde eine Einschränkung des zur Verfügung stehenden Speichers vorliegen (eine LUT ist in der Regel speicherintensiv), so könnte auf eine andere Approximation der LUT zurückgegriffen werden, die z.B. eine größere Auflösung besitzt oder polynombasiert ist, was jedoch hier nicht betrachtet wird.

5.2.3.1 Berechnung der Delta-Plane

Die notwendigen Eingabedaten zur Berechnung der Delta-Plane resultieren aus einem Tiefenbild, das bei der Betrachtung eines frontal zur Kinect ausgerichteten planaren Objekts entstanden ist, welches den gesamten Bildbereich des Tiefenbilds einnimmt. Idealerweise müsste das Tiefenbild rauschfrei und die Kinect exakt parallel zur Objektebene ausgerichtet sein. Dies wäre jedoch unpraktikabel. Deshalb wird stattdessen eine Menge von Tiefenbildern des planaren Objekts verwendet und eine Art Durchschnittsberechnung durchgeführt, um eine Überanpassung ("overfitting") an das Rauschen eines einzigen Tiefenbildes sowie mögliche Abweichungen von der idealen, parallelen Ausrichtung der Kinect zu kompensieren. Die Kinect wurde auf einem Stativ frontal vor einer Wand positioniert. Es sind mehrere Bilder bei gleicher Distanz aufgezeichnet worden, während die Kinect geringfügig um die horizontale und vertikale Achse geneigt wurde (Neigungen innerhalb von $\pm 2.5^\circ$). Die so erfassten Tiefenbilder seien durch die Menge $\mathcal{T} = \{\mathbf{T}_0, \dots, \mathbf{T}_{m-1}\}$ bezeichnet. Der Gesamttablauf bei der Berechnung der Delta-Plane ist in Abb. 5.3 veranschaulicht. Die dort dargestellten einzelnen Schritte werden nachfolgend detailliert erläutert.

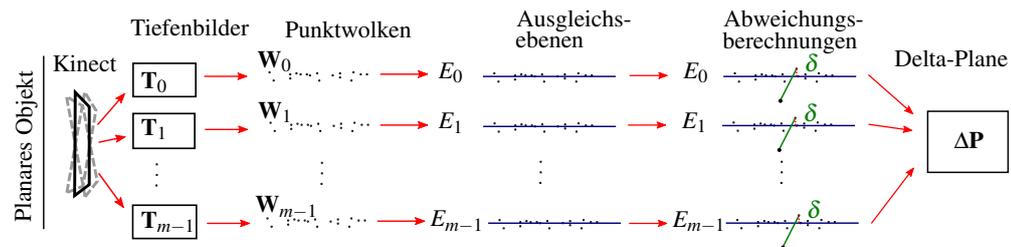


Abbildung 5.3: Übersicht der Einzelschritte zur Berechnung der Delta-Plane $\Delta\mathbf{P}$. Nach der Erfassung mehrerer Tiefenbilder \mathbf{T}_j eines planaren Objekts unter geringfügigem Neigen der Kinect werden zunächst die dazugehörigen Punktwolken \mathbf{W}_j berechnet. Für jede Punktwolke wird eine Ausgleichsebene E_j bestimmt. Auf diesen basierend werden die Abweichungen zum Idealfall (Planarität der Punktwolke) berechnet. Die resultierende Delta-Plane ist schließlich die Kombination der gewichteten Abweichungsberechnungen und beinhaltet das gesuchte, komplexe Fehlermuster.

Zu jedem Tiefenbild \mathbf{T}_j wird die entsprechende 3D-Punktwolke \mathbf{W}_j , wie in Abschnitt 4.5.4 beschrieben, berechnet (jedem Pixel im Tiefenbild wird ein 3D-Punkt zugeordnet). Diese bilden die Menge der Punktwolken $\mathcal{W} = \{\mathbf{W}_0, \dots, \mathbf{W}_{m-1}\}$. Aufgrund des beobachteten planaren Objekts (Wand) müssten die Punkte jeder Punktwolke idealerweise in einer Ebene liegen. Das ist in der Praxis jedoch nicht der Fall (vgl. Abb. 5.2(c)). Um die Abweichung vom Idealfall abzuschätzen, wird zunächst für jede Punktwolke \mathbf{W}_j eine Ausgleichsebene E_j durch die folgenden Schritte bestimmt:

1. Berechnung des Wolkenzentrums \mathbf{z}_j für die n_j Punkte $\mathbf{w}_j^0, \dots, \mathbf{w}_j^{n_j-1}$ der Punktwolke \mathbf{W}_j durch

$$\mathbf{z}_j = \frac{1}{n_j} \sum_{i=0}^{n_j-1} \mathbf{w}_j^i. \quad (5.1)$$

2. Konstruktion der $n_j \times 3$ Matrix \mathbf{A}_j , deren Zeilenvektoren \mathbf{a}_j^i den um das Wolkenszentrum zentrierten Punkten der Punktwolke \mathbf{W}_j entsprechen:

$$\mathbf{A}_j = \begin{pmatrix} \mathbf{a}_j^0 \\ \vdots \\ \mathbf{a}_j^{n_j-1} \end{pmatrix} = \begin{pmatrix} \mathbf{w}_j^0 - \mathbf{z}_j \\ \vdots \\ \mathbf{w}_j^{n_j-1} - \mathbf{z}_j \end{pmatrix}. \quad (5.2)$$

3. Berechnung der Singulärwertzerlegung [HZ03] der Matrix $\mathbf{A}_j = \mathbf{U}_j \mathbf{S}_j \mathbf{V}_j^T$, wobei \mathbf{U}_j eine $n_j \times n_j$, \mathbf{S}_j eine $n_j \times 3$ und \mathbf{V}_j eine 3×3 Matrix ist.
4. Bestimmung der gesuchten Ebene E_j in Normalform durch

$$E_j : (\mathbf{x} - \mathbf{z}_j) * \mathbf{n}_j = 0, \quad (5.3)$$

wobei $\mathbf{n}_j = \mathbf{v}_j^2$ gilt und \mathbf{v}_j^2 der dritten Spalte der Matrix $\mathbf{V}_j = (\mathbf{v}_j^0 \ \mathbf{v}_j^1 \ \mathbf{v}_j^2)$ und der Wolkenschwerpunkt \mathbf{z}_j dem Stützvektor der Ebene entspricht.

Diese Schritte werden für alle m Punktwolken durchgeführt, um die entsprechenden Ausgleichsebenen E_0, \dots, E_{m-1} zu erhalten. Die beiden ersten Spaltenvektoren \mathbf{v}_j^0 und \mathbf{v}_j^1 in Schritt 4 entsprechen den Richtungsvektoren, die die Ebene E_j aufspannen. Das Kreuzprodukt der beiden Vektoren entspricht dem Normalenvektor dieser Ebene. Da die Spaltenvektoren von \mathbf{V}_j orthonormal sind, gilt $\mathbf{v}_j^0 \times \mathbf{v}_j^1 = \mathbf{v}_j^2$ und \mathbf{v}_j^2 ist bereits der gesuchte Normalenvektor von E_j . Es sei angemerkt, dass bei den beschriebenen Berechnungsschritten einer Ausgleichsebene E_j o.B.d.A. davon ausgegangen wird, dass alle 3D-Punkte gültig sind. Als nicht gültig wird dabei ein 3D-Punkt bezeichnet, der keinen gültigen Distanzeintrag im entsprechenden Pixel des dazugehörigen Tiefenbildes besitzt (Distanzwert gleich Null). Ungültige Distanzeinträge können z.B. bei Reflexionen entstehen, die die korrekte Erkennung des Punktmusters verhindern. Entsprechende 3D-Punkte fallen im Punkt $(0, 0, 0)^T$ zusammen und gehören nicht zu der Punktwolke, die das planare Objekt repräsentieren. Sollten in der Praxis solche Punkte vorhanden sein, müssten diese von der Berechnung der Ausgleichsebene ausgeschlossen werden, da sie Ausreißer darstellen und die Lage der Ebene verfälschen würden.

Die Delta-Plane wird als zweidimensionales Array der Größe $w_T \times h_T$ definiert, das mit $\Delta\mathbf{P}$ abgekürzt wird. Dabei entsprechen w_T und h_T der Breite und Höhe des Tiefenbildes der Kinect. Der Algorithmus 3 beschreibt die Berechnung der einzelnen Einträge $\Delta\mathbf{P}(u, v)$, $u = 0, \dots, w_T - 1$, $v = 0, \dots, h_T - 1$, der Delta-Plane. Dabei wird berücksichtigt, dass ungültige 3D-Punkte vorkommen können. Die Eingabeparameter sind die Punktwolken \mathbf{W}_j , die Ausgleichsebenen E_j , das Kamerazentrum der Infrarotkamera \mathbf{c}_{IR} sowie die distanzabhängige Funktion $f_d(z)$. Zu den Ausgabeparametern des Algorithmus zählt neben $\Delta\mathbf{P}$ die mittlere Z-Entfernung d_{mean} vom Kamerazentrum zu den relevanten Bereichen der Ausgleichsebenen. d_{mean} beschreibt somit die Distanz, in der die Delta-Plane konstruiert worden ist. In den Algorithmuszeilen 3.1 bis 3.6 finden die Parameterinitialisierungen statt, wobei $\mathbf{N}(u, v)$ die Anzahl der Einträge der Delta-Plane an der Position (u, v) speichert. Dies ist für die Mittelwertbildung der einzelnen Einträge von $\Delta\mathbf{P}$ notwendig, da die Anzahl aufgrund vorkommender ungültiger 3D-Punkte variieren kann. Die Funktion $\mathcal{F}_{Pos2Index}(u, v)$ in Zeile 3.11 liefert zur aktuellen Position (u, v) den Index des dazugehörigen 3D-Punktes der Punktwolke. Details sind

Algorithmus 3 : Berechnung der Delta-Plane

Input : Punktwolken $\mathbf{W}_0, \dots, \mathbf{W}_{m-1}$, Ausgleichsebenen E_0, \dots, E_{m-1} , Zentrum der IR-Kamera \mathbf{c}_{IR} , distanzabhängige Funktion $f_d(z)$

Output : Delta-Plane $\Delta\mathbf{P}$, mittlere Ausgleichsebenentfernung d_{mean}

```

3.1  $d_{mean} := 0$ 
3.2  $n_{valid} := 0$  // Anzahl gültiger 3D-Punkte einer Punktwolke
3.3 for  $u := 0$  to  $w - 1$  do
3.4   for  $v := 0$  to  $h - 1$  do
3.5      $\Delta\mathbf{P}(u, v) := 0$  //Initialisierung
3.6      $\mathbf{N}(u, v) := 0$  // 2D-Array, speichert Anzahl der Einträge in  $\Delta\mathbf{P}$ 

3.7 for  $j := 0$  to  $m - 1$  do
3.8    $n_{valid} := 0$ 
3.9   for  $u := 0$  to  $w - 1$  do
3.10    for  $v := 0$  to  $h - 1$  do
3.11       $i := \mathcal{F}_{Pos2Index}(u, v)$  // Index des 3D-Punktes berechnen
3.12      if  $\mathbf{w}_j^i \neq (0, 0, 0)^T$  then
3.13         $n_{valid} := n_{valid} + 1$ 
3.14         $\lambda := \frac{(\mathbf{z}_j - \mathbf{c}_{IR}) * \mathbf{n}_j}{(\mathbf{w}_j^i - \mathbf{c}_{IR}) * \mathbf{n}_j}$  // Schnittpunktparameter
3.15         $\mathbf{s} := (s_x, s_y, s_z)^T = \mathbf{c}_{IR} + \lambda(\mathbf{w}_j^i - \mathbf{c}_{IR})$  // Schnittpunkt
3.16         $\delta := s_z - w_{j,z}^i$  // Tiefendifferenz,  $\mathbf{w}_j^i = (w_{j,x}^i, w_{j,y}^i, w_{j,z}^i)^T$ 
3.17         $\Delta\mathbf{P}(u, v) := \Delta\mathbf{P}(u, v) + \delta / f_d(w_{j,z}^i - c_{IR,z})$  // Akkumulation
3.18         $\mathbf{N}(u, v) := \mathbf{N}(u, v) + 1$ 
3.19         $d_{mean} := d_{mean} + (s_z - c_{IR,z})$ 
3.20     $d_{mean} := d_{mean} / n_{valid}$ 
3.21  $d_{mean} := d_{mean} / m$ 
3.22 for  $u := 0$  to  $w - 1$  do
3.23   for  $v := 0$  to  $h - 1$  do
3.24      $\Delta\mathbf{P}(u, v) := \Delta\mathbf{P}(u, v) / \mathbf{N}(u, v)$  //Mittelwertbildung
    
```

dem Abschnitt 4.5.4 zu entnehmen. Die Gültigkeitsüberprüfung des aktuell betrachteten 3D-Punktes \mathbf{w}_j^i findet in Zeile 3.12 statt. Die Zeilen 3.14 bis 3.16 bilden den Kern der Abweichungsberechnung. Dabei wird für jeden aktuellen 3D-Punkt \mathbf{w}_j^i zunächst eine Gerade g (in Punkt-Richtungsform) konstruiert, die durch \mathbf{w}_j^i und den Ursprung der IR-Kamera \mathbf{c}_{IR} verläuft:

$$g : \mathbf{x} = \mathbf{c}_{IR} + \lambda(\mathbf{w}_j^i - \mathbf{c}_{IR}). \quad (5.4)$$

Als nächstes wird der Schnittpunkt \mathbf{s} zwischen der Geraden g und der aktuellen Ausgleichsebene $E_j : (\mathbf{x} - \mathbf{z}_j) * \mathbf{n}_j = 0$ (siehe Gleichung (5.3)) bestimmt. Dazu wird λ in der Gleichung in Zeile 3.14 durch Einsetzen der Geradengleichung in die Ebenengleichung sowie entsprechende Umformungen berechnet. In δ in Zeile 3.16 wird die Z-Komponente des Differenzvektors zwischen dem Schnittpunkt \mathbf{s} , in dem die Gerade g die Ausgleichsebene E_j schneidet, und dem 3D-Punkt \mathbf{w}_j^i zwischengespeichert (Tiefendifferenz). Die Abb. 5.4 verdeutlicht die Abweichungsberechnung.

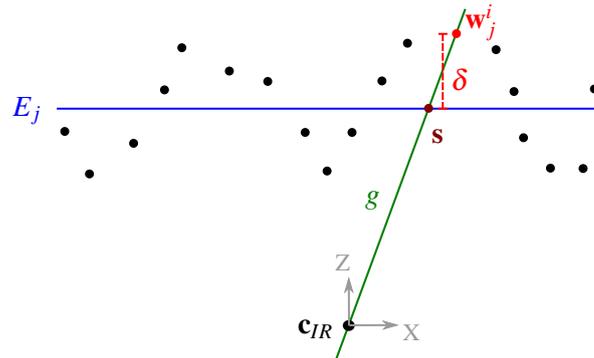


Abbildung 5.4: Veranschaulichung der Berechnung der Tiefendifferenz δ (gestrichelte rote Linie) zwischen dem aktuell betrachteten 3D-Punkt w_j^i (roter Punkt) und dem Schnittpunkt s (braun). Der Schnittpunkt s entsteht durch den Schnitt der Geraden g (grün), definiert durch das Kamerazentrum und den aktuellen Punkt w_j^i , mit der Ausgleichsebene E_j (blau).

Bevor die Abweichung zur aktuellen Position in der Delta-Plane hinzuaddiert wird, wird diese gewichtet mit $1/f_d(z)$, wobei $f_d(z)$ eine distanzabhängige Funktion ist und z der Tiefe bzw. der Z-Distanz zwischen dem Kamerazentrum und dem aktuell betrachteten 3D-Punkt w_j^i entspricht (siehe Algorithmuszeile 3.17). Ziel der Division ist es, eine möglichst distanzunabhängige Delta-Plane zu erhalten. Details zur Wahl von $f_d(z)$ werden im Abschnitt 5.2.3.5 näher erläutert. In den letzten Schritten (Zeilen 3.22 bis 3.24) wird schließlich der Mittelwert aller aufaddierten Abweichungen für die jeweilige Position (u, v) der Delta-Plane berechnet. Nach Ablauf des Algorithmus 3 steht die gesuchte Delta-Plane zur Verfügung.

Es sei angemerkt, dass die Position des Kamerazentrums c_{IR} der IR-Kamera in der Algorithmusbeschreibung zur besseren Verdeutlichung explizit aufgeführt wurde. Jedoch wird angenommen, dass der Ursprung des Koordinatensystems, in dem die 3D-Punkte definiert sind, im Kamerazentrum liegt, so dass $c_{IR} = (c_{IR,x}, c_{IR,y}, c_{IR,z})^T = (0, 0, 0)^T$ gilt und folglich Vereinfachungen der Notation in den Zeilen 3.14, 3.15, 3.17 und 3.19 möglich sind.

5.2.3.2 Reproduzierbarkeit

Die entsprechend des in Abschnitt 5.2.3.1 beschriebenen Verfahrens berechnete Delta-Plane wird auf ihre Reproduzierbarkeit hin untersucht. Dies ist notwendig, da bei der Datenaufzeichnung das geringfügige Schwenken der Kinect einen Einfluss auf die resultierende Delta-Plane haben könnte.

Das Experiment wurde wie folgt durchgeführt. Zunächst sind 36 Tiefenbilder einer 1155 mm entfernten Wand unter geringfügigem Schwenken frontal aufgezeichnet worden. Diese Bildermenge ist zufällig in drei Bilderpools mit je 12 Bildern aufgeteilt worden. Aus jedem Pool wurde jeweils eine Delta-Plane erzeugt, wobei die distanzabhängige Funktion zunächst konstant auf $f_d(z) = 1$ gesetzt worden ist. Die drei so entstandenen Delta-Planes sind mit ΔP_1 , ΔP_2 und ΔP_3 bezeichnet und in den Abb. 5.5(a)

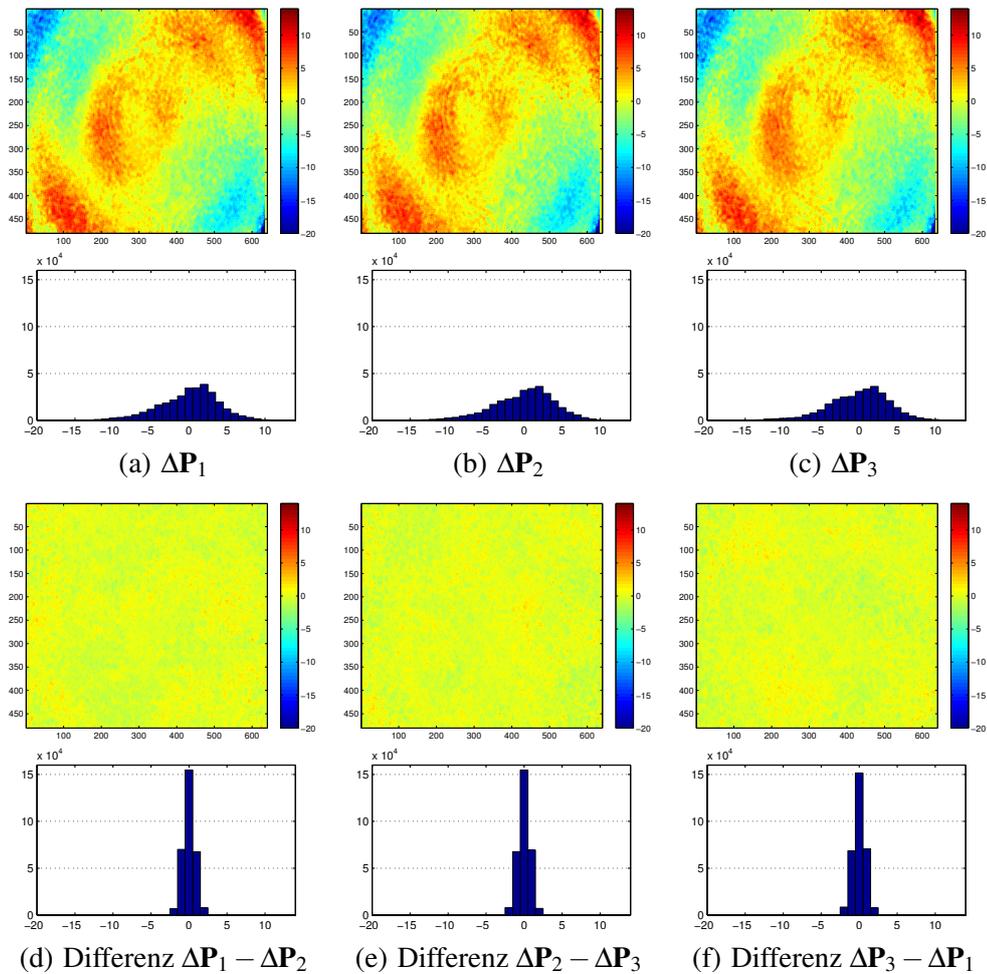


Abbildung 5.5: Verdeutlichung der Reproduzierbarkeit der Delta-Plane. In (a) bis (c) sind die Delta-Planes der drei Bilderpools aus je 12 Tiefenbildern dargestellt, die in einer Entfernung von 1155 mm zu Wand erstellt worden sind. Die Farbskala ist in Millimetern angegeben. (d) bis (f) zeigen die Differenz von je zwei Delta-Planes. Die dazugehörigen Histogramme sind jeweils unter den Abbildungen zu sehen.

bis 5.5(c) dargestellt, wobei die farbkodierten Werte die durchschnittlichen Abweichungen von den Ausgleichsebenen, gegeben in Millimetern, ausdrücken. In diesen Abbildungen ist die charakteristische Struktur des komplexen sensorortsabhängigen Fehlers deutlich zu erkennen. Die drei Delta-Planes sind sich sehr ähnlich, was für eine hohe Reproduzierbarkeit bei der Anwendung des in Abschnitt 5.2.3.1 beschriebenen Verfahrens spricht. Die jeweiligen Histogramme zu den Werten in den Delta-Planes sind ebenfalls vergleichbar.

Um die Unterschiede zwischen den Delta-Planes genauer zu untersuchen, wurde die pixelweise Differenz zwischen je zwei Delta-Planes berechnet. Das Resultat der Differenz $\Delta\mathbf{P}_1 - \Delta\mathbf{P}_2$ ist in Abb. 5.5(d), $\Delta\mathbf{P}_2 - \Delta\mathbf{P}_3$ in Abb. 5.5(e) und $\Delta\mathbf{P}_3 - \Delta\mathbf{P}_1$ in Abb. 5.5(f) dargestellt. Die Differenzwerte liegen Größtenteils bei Null (vgl. Histogramme in 5.5(d) bis 5.5(f)) und sind lediglich durch ein sichtbares Rauschen überlagert. Die Auswertung der dazugehörigen Histogramme ergibt, dass sich 95% aller Differenzwerte im Wertebereich von ± 1 mm befinden. Zieht man die erwartete Tiefenauflösung für die vorliegende Distanz von 1155 mm heran, die 3.80 mm beträgt (vgl. Abb. 4.13), so lässt dies ebenfalls die Schlussfolgerung zu, dass das Verfahren zur Berechnung der Delta-Plane eine hohe Reproduzierbarkeit aufweist. Somit kann das Verfahren als robust gegenüber den Abweichungen angenommen werden, die durch das geringfügige Schwenken der Kinect bei der Tiefenbildaufnahme verursacht wurden.

5.2.3.3 Delta-Plane basierter Korrekturansatz

Wie am Anfang des Abschnitts 5.2.3 erwähnt, wird die nach Ablauf des Algorithmus 3 zur Verfügung stehende Delta-Plane $\Delta\mathbf{P}$ als Look-Up Tabelle (LUT) verwendet, in der für jedes Pixel des Tiefenbildes ein individueller Korrekturwert gespeichert wird. Sei \mathbf{T} ein aktuelles Tiefenbild der Kinect. So lässt sich das korrigierte Tiefenbild \mathbf{T}' pixelweise durch

$$\mathbf{T}'(u, v) = \mathbf{T}(u, v) + \Delta\mathbf{P}(u, v) \cdot f_d(\mathbf{T}(u, v)) \quad (5.5)$$

berechnen, wobei $u = 0, \dots, w_T - 1$, $v = 0, \dots, h_T - 1$ und $f_d(\mathbf{T}(u, v))$ die distanzabhängige Gewichtungsfunktion ist (vgl. Abschnitt 5.2.3.1 und 5.2.3.5), die für die aktuelle Distanz $\mathbf{T}(u, v)$ ausgewertet wird.

Wie in Abschnitt 4.5.6 erläutert, überträgt die Kinect die Disparitätswerte als 10 Bit Ganzzahlwerte mit maximal 1024 Stufen. Folglich unterliegen auch die vom OpenNI-Framework umgerechneten und zur Verfügung gestellten Tiefenmesswerte im Tiefenbild \mathbf{T} einer Quantisierung. Durch die Anwendung der Gleichung (5.5) können jedoch korrigierte Tiefenmesswerte entstehen, die zwischen den quantisierten Tiefenmesswerten liegen. Ist dies nicht erwünscht, so kann eine zusätzliche Quantisierung mit Hilfe der Funktion $\mathcal{F}_{QuantMapper}(x)$ durchgeführt werden, die einen beliebigen Tiefenwert x auf einen der quantisierten Tiefenmesswerte abbildet. Die Delta-Plane basierte Korrektur unter Berücksichtigung der Quantisierung ist dann gegeben durch

$$\mathbf{T}'_Q(u, v) = \mathcal{F}_{QuantMapper}(\mathbf{T}(u, v) + \Delta\mathbf{P}(u, v) \cdot f_d(\mathbf{T}(u, v))). \quad (5.6)$$

Details zur Untersuchung der Quantisierungsstufen und der Funktion $\mathcal{F}_{QuantMapper}(x)$ sind in Abschnitt 5.2.3.4 beschrieben.

5.2.3.4 Quantisierung der Tiefenmesswerte

Um die quantisierten Tiefenmesswerte zu ermitteln, die vom OpenNI-Framework zur Verfügung gestellt werden, sind zunächst eine Menge an Tiefenbildern von Szenen aufgezeichnet worden, in denen Objekte in unterschiedlichen Distanzen zur Kinect vorkamen. Um möglichst lückenlos alle quantisierten Tiefenmesswerte zu erfassen, waren besonders Szenen geeignet, in denen die Kinect z.B. mehrere Bilder eines langen Flurs beobachtete (vgl. Abb. 2.4(e)) oder schräg auf Wände ausgerichtet war, so dass aufgrund der Perspektive ein kontinuierlicher Entfernungsverlauf zu beobachten war, der viele unterschiedliche Tiefenwerte in den Tiefenbildern zur Folge hatte. Insgesamt wurden ca. 4 Millionen Messpunkte ausgewertet. Von den theoretisch 1024 möglichen (10 Bit) konnten 764 unterschiedliche Tiefenmesswerte zwischen 374 und 8995 mm ermittelt werden.

Das OpenNI-Framework gibt die Distanzen ganzzahlig in Millimetern an. Bis zu einer Distanz von 609 mm ist jeder Tiefenwert vertreten, so dass der Abstand Δz zwischen zwei Tiefenmesswerten 1 mm beträgt (Quantisierungsschrittweite). Zwischen 611 und 845 mm beträgt Δz entweder 1 oder 2 mm. Der größte und letzte ermittelte Abstand bei der Distanz von 8995 mm beträgt $\Delta z = 228$ mm. Eine Übersicht der Quantisierungsschrittweiten für die ersten 700 ermittelten Tiefenmesswerte ist in Abb. 5.6(a) dargestellt.

Der Abstand zwischen den ermittelten quantisierten Tiefenmesswerten wurde mit der in Abschnitt 4.5.6 beschriebenen Tiefenauflösung der Kinect verglichen. Wie in

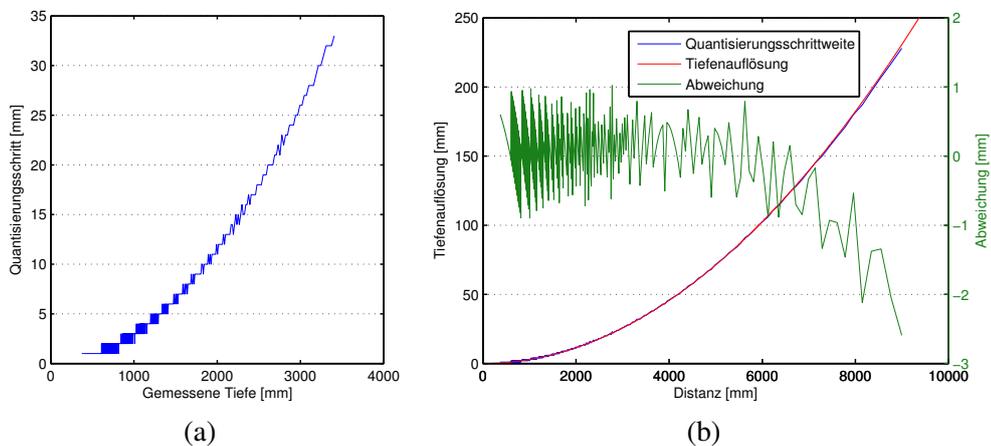


Abbildung 5.6: In (a) wird die Quantisierungsschrittweite für die ersten 700 durch das OpenNI-Framework zur Verfügung gestellten, quantisierten Tiefenmesswerte dargestellt. Zwischen 374 und 609 mm beträgt die Quantisierungsschrittweite nur 1 mm. Mit wachsender Distanz steigt diese annähernd quadratisch an. Die Quantisierungsschrittweite für alle Tiefenmesswerte bis 8995 mm sind in (b) gezeigt (blaue Kurve). Der Verlauf der in Abschnitt 4.5.6 ermittelten Tiefenauflösung ist durch die rote Kurve dargestellt. Die nur sehr geringfügige Abweichung der Quantisierungsschrittweite zur vorhergesagten Tiefenauflösung ist durch die grüne Kurve verdeutlicht (rechte Skala). Für Distanzen unterhalb von 7000 mm liegt die Abweichung im Bereich von ± 1 mm.

Abb. 5.6(b) zu erkennen ist, entsprechen die Quantisierungsschrittweiten für die jeweiligen ermittelten Tiefenmesswerte des OpenNI-Frameworks, bis auf eine geringfügige Abweichung (durch grüne Kurve dargestellt), der durch die Gleichung (4.39) vorhergesagten Tiefenauflösung.

Nachdem die einzelnen quantisierten Messwerte ermittelt worden sind, kann auch die in Abschnitt 5.2.3.3 verwendete Funktion $\mathcal{F}_{QuantMapper}(x)$, die einen beliebigen Tiefenwert x auf einen der ermittelten quantisierten Tiefenmesswerte abbildet, definiert werden. Seien q_0, q_1, \dots, q_{n-1} die ermittelten Tiefenmesswerte mit $n = 764$. Dann ist der gesuchte quantisierte Wert q_i jener, der dem Tiefenwert x am nächsten liegt:

$$\mathcal{F}_{QuantMapper}(x) = q_i, \text{ mit } |x - q_i| \leq |x - q_j|, \forall j \neq i. \quad (5.7)$$

Im Fall der Uneindeutigkeit (x liegt genau zwischen zwei Quantisierungswerten) wird der Quantisierungswert mit dem kleineren Index gewählt. Um insbesondere die Rechenzeit während des laufenden Betriebs (online) zu optimieren, wurde eine Look-Up-Tabelle (LUT) für alle Tiefenwerte zwischen 0 und 10000 mm in 1 mm Schritten angelegt, in der das Quantisierungsergebnis vorberechnet wurde und in der der gesuchte Quantisierungswert an der Tabellenposition $\lfloor x + 0.5 \rfloor$ nachgeschlagen werden kann.

5.2.3.5 Distanzabhängigkeit der Delta-Plane

Wird die distanzabhängige Gewichtungsfunktion aus Abschnitt 5.2.3.1 (Zeile 17 im Algorithmus 3) zunächst nicht berücksichtigt bzw. konstant auf $f_d(z) = 1$ gesetzt, so wird, wie in Abb. 5.7 zu sehen, die Distanzabhängigkeit der Delta-Plane deutlich. Dort sind zwei Delta-Planes visualisiert, deren zugrunde liegenden Tiefenbilder in den Distanzen 1500 bzw. 2250 mm gewonnen worden sind. Die Struktur der Abweichung ist bei beiden Distanzen zwar ähnlich, jedoch unterscheiden sich die Werte hinsichtlich ihrer Ausprägung deutlich. Dabei ist die Ausprägung bei größeren Distanzen stärker. Mit Struktur soll im Folgenden die Verteilung der Abweichungswerte gemeint sein, die bei der Visualisierung der Delta-Plane in Abb. 5.7 am Höhenprofil erkennbar ist. Die Distanzabhängigkeit der Delta-Plane soll durch eine geeignete Wahl der Funktion $f_d(z)$ möglichst kompensiert werden. Um die Beschaffenheit von $f_d(z)$ beurteilen zu können, ist es notwendig, das Ausprägungsmaß der Abweichungen in den Delta-Planes bei unterschiedlichen Distanzen bewerten zu können. Die Bewertungsfunktion, die ein skalares Maß liefert, welches dem Mittelwert aller Beträge der Einträge $\Delta\mathbf{P}(u, v)$ der Delta-Plane entspricht, sei gegeben durch

$$\mathcal{F}_{MeanAbs}(\Delta\mathbf{P}) = \frac{1}{w_{\Delta\mathbf{P}} \cdot h_{\Delta\mathbf{P}}} \sum_{u=0}^{w_{\Delta\mathbf{P}}-1} \sum_{v=0}^{h_{\Delta\mathbf{P}}-1} |\Delta\mathbf{P}(u, v)|, \quad (5.8)$$

wobei $w_{\Delta\mathbf{P}}$ bzw. $h_{\Delta\mathbf{P}}$ die Breite bzw. die Höhe der Delta-Plane bezeichnet.

5.2.3.5.1 Ermittlung der Gewichtungsfunktion. Um die Gewichtungsfunktion $f_d(z)$ zu ermitteln, durch die die stabilsten Delta-Planes bei unterschiedlichen Distanzen ermöglicht werden, wurde wie folgt vorgegangen. In $n = 11$ verschiedenen

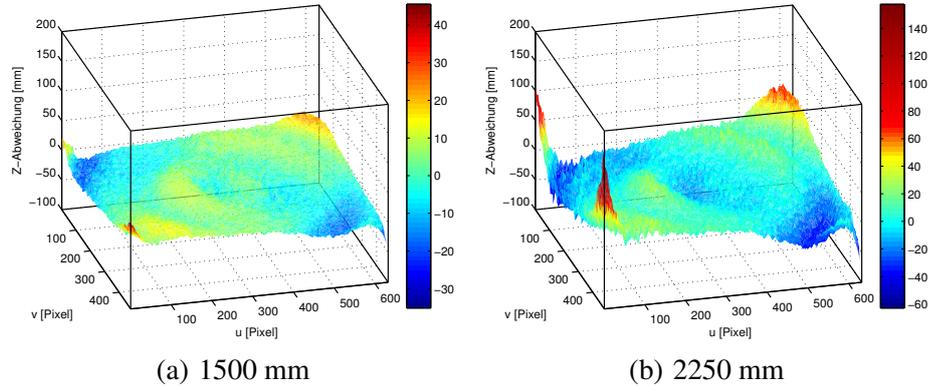


Abbildung 5.7: Verdeutlichung der Distanzabhängigkeit der Delta-Planes. Die zugrunde liegenden Tiefenbilder für die in (a) dargestellte Delta-Plane wurden in 1500 mm und in (b) in 2250 mm Entfernung zur Wand aufgezeichnet. Zwar ist die Struktur ähnlich, jedoch ist die Ausprägung der Abweichungen bei der größeren Distanz deutlich stärker. Bei der Berechnung dieser Delta-Planes ist die distanzabhängige Gewichtungsfunktion zunächst nicht berücksichtigt bzw. konstant auf $f_d(z) = 1$ gesetzt worden.

Distanzen zwischen 750 und 3250 mm sind in 250 mm Schritten jeweils 10 Tiefenbilder entsprechend der Beschreibung in Abschnitt 5.2.3.1 aufgezeichnet worden. Aus den jeweiligen Tiefenbildern in den n Distanzen wurden die Delta-Planes $\Delta\mathbf{P}_0, \dots, \Delta\mathbf{P}_{n-1}$ unter Verwendung einer zu untersuchenden Gewichtungsfunktion $f_d(z)$ berechnet. Für jede Delta-Plane wurde anschließend die skalare Bewertungsfunktion aus Gleichung (5.8) ausgewertet, so dass der Vektor $\mathbf{s} = (s_0, \dots, s_{n-1})$ mit $s_i = \mathcal{F}_{\text{MeanAbs}}(\Delta\mathbf{P}_i)$, $i = 0, \dots, n-1$, entstand. Dabei können die Vektoreinträge den n unterschiedlichen Distanzen über den Index i zugeordnet werden.

Dieses Vorgehen wurde für die vier Gewichtungsfunktionen

$$f_d(z) = 1, \quad f_d(z) = z, \quad f_d(z) = z^2, \quad f_d(z) = z^3 \quad (5.9)$$

durchgeführt. Somit standen vier Vektoren, die mit $\mathbf{s}^{(1)}$, $\mathbf{s}^{(z)}$, $\mathbf{s}^{(z^2)}$ und $\mathbf{s}^{(z^3)}$ bezeichnet werden, zur Verfügung. Um die Werte der Elemente zwischen den vier Vektoren vergleichbar zu machen, war eine Normierung der Elemente notwendig. Sei $\mathbf{s}^{(j)}$ mit

$$\mathbf{s}^j = \left(s_0^{(j)}, \dots, s_{n-1}^{(j)} \right), \quad j \in \{1, z, z^2, z^3\}, \quad (5.10)$$

einer dieser vier Vektoren. Dann ergibt sich der jeweilige Vektor $\check{\mathbf{s}}^{(j)}$ mit den normierten Werten durch

$$\check{\mathbf{s}}^{(j)} = \left(\check{s}_0^{(j)}, \dots, \check{s}_{n-1}^{(j)} \right) = \left(\frac{s_0^{(j)}}{\min(\mathbf{s}^{(j)})}, \dots, \frac{s_{n-1}^{(j)}}{\min(\mathbf{s}^{(j)})} \right), \quad (5.11)$$

wobei $\min(\mathbf{s})$ das kleinste Element des Vektors \mathbf{s} liefert. Die so ermittelten Vektoreinträge wurden den unterschiedlichen Distanzen über den Index $i = 0, \dots, n-1$ zugeordnet und sind als Kurven in Abb. 5.8 dargestellt, wobei $\check{\mathbf{s}}^{(1)}$ die blaue, $\check{\mathbf{s}}^{(z)}$ die grüne,

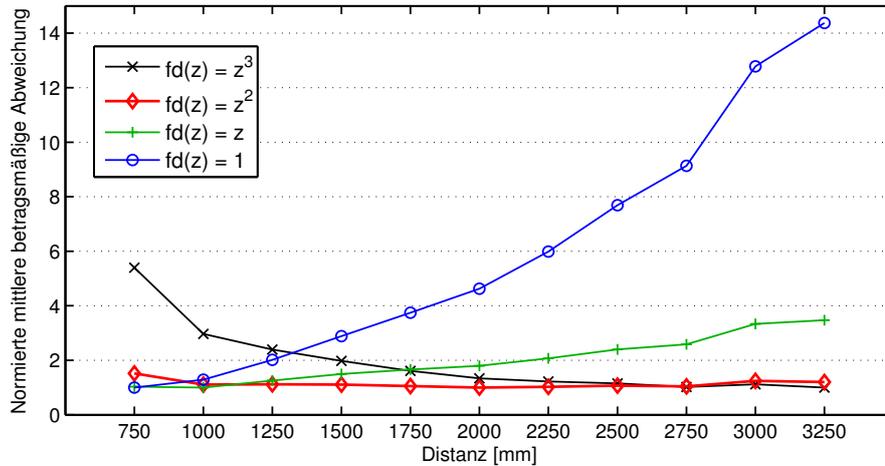


Abbildung 5.8: Distanzabhängigkeit der Delta-Plane für unterschiedlich gewählte Gewichtungsfunktionen f_d . Die größte Stabilität kann für $f_d(z) = z^2$ beobachtet werden.

$\xi(z^2)$ die rote und $\xi(z^3)$ die schwarze Kurve ergab. Die Kurven repräsentieren somit die normierte mittlere betragsmäßige Abweichung einer Delta-Plane in Abhängigkeit von der verwendeten Gewichtungsfunktion $f_d(z)$ und der Distanz zwischen Kinect und Wand. Durch die Normierung (Division der Werte durch den Minimalwert) besitzen alle Kurven ihr Minimum bei 1 und die Werte auf der Y-Achse drücken Vielfache des minimalen Wertes aus.

Ohne Berücksichtigung der Gewichtungsfunktion, also dem Fall $f_d(z) = 1$, ist anhand der blauen Kurve ein deutlicher Anstieg zu erkennen. Zwischen der minimalen (750 mm) und der maximalen untersuchten Distanz (3250 mm) liegt mehr als ein Faktor 14 vor. Für $f_d(z) = z$ ist ebenfalls ein Anstieg zu verzeichnen, jedoch in einem geringeren Ausmaß (Faktor 3.47 bei 3250 mm). Der stabilste Verlauf kann für $f_d(z) = z^2$ (rote Kurve) beobachtet werden, der mit Ausnahme der Distanz 750 mm (Faktor 1.52), durchgängig einen Faktor zwischen 1 und 1.24 aufweist. Bei der Verwendung von $f_d(z) = z^3$ liegt wieder ein deutlich größerer Faktor von bis zu 5.5 vor, jedoch mit der Besonderheit, dass, im Gegensatz zu den anderen Verläufen, ein fallender Trend bei steigender Distanz zu beobachten ist.

Aufgrund der beschriebenen Normierung (Division durch Minimalwert) haben sowohl $f_d(z) = z^2$ als auch $f_d(z) = kz^2$, wobei k ein konstanter Vorfaktor mit $k > 0$ ist, den gleichen Verlauf in der oben betrachteten Auswertung zur Folge. Da k letztendlich einen Skalierungsfaktor für die in einer Delta-Plane $\Delta\mathbf{P}$ gespeicherten Werte darstellt, erscheint es zweckmäßig, k so zu wählen, dass die gespeicherten Werte möglichst aussagekräftig sind. Als aussagekräftige Einheit für jeden Eintrag $\Delta\mathbf{P}(u, v)$ wird die Anzahl der Tiefenquantisierungsstufen erachtet. Um dies zu erreichen, wird $k = 2.85 \cdot 10^{-6}$ gewählt. Dadurch erhält man die Gleichung zur Abschätzung der Tiefenauflösung Δz (siehe Gleichung (4.39)) mit der in Abschnitt 4.5.6 verwendeten Konstante $\frac{\alpha}{fb} = 2.85 \cdot 10^{-6}$:

$$f_d(z) = kz^2 = (2.85 \cdot 10^{-6})z^2 = \left(\frac{\alpha}{fb}\right)z^2 = \Delta z. \quad (5.12)$$

Dabei gibt die Tiefenauflösung (vgl. Abb. 4.13) den erwarteten Distanzsprung bis zum nächsten möglichen Tiefenwert an. Wie in der Abb. 5.6(b) in Abschnitt 5.2.3.4 verdeutlicht, entspricht die Größe der Distanzsprünge (bis auf eine geringfügige Abweichung) der Größe der Quantisierungsstufen.

Wird k bei der Erzeugung einer Delta-Plane $\Delta\mathbf{P}$ wie beschrieben gewählt, so wird jeder Eintrag $\Delta\mathbf{P}(u, v)$, der die Abweichung (Distanz) von der Ausgleichsebene an der Stelle (u, v) beschreibt, durch $f_d(z)$ geteilt. Da $f_d(z)$ der Größe einer Quantisierungsstufe bei der gegebenen Distanz z entspricht, speichert folglich jeder Eintrag $\Delta\mathbf{P}(u, v)$ die Abweichung wie gewünscht als Anzahl der Tiefenquantisierungsstufen. Trotz des Bezugs zu den Quantisierungsstufen sei angemerkt, dass die in $\Delta\mathbf{P}$ gespeicherten Werte reellwertig sind, und die Abbildung auf die tatsächlichen quantisierten Tiefenwerte erst stattfindet, wenn der endgültige korrigierte Tiefenwert vorliegt (siehe Gleichung (5.6) in Abschnitt 5.2.3.3).

Für alle nachfolgenden Untersuchungen wird die Gewichtungsfunktion entsprechend Gleichung (5.12) festgelegt, da diese die größte Distanzunabhängigkeit der Abweichungswerte der Delta-Plane lieferte.

5.2.3.5.2 Struktur der Werteverteilung. Zwar sind die Abweichungswerte nach geeigneter Wahl von $f_d(z)$, wie im vorangegangenen Abschnitt gezeigt, relativ stabil gegenüber der Distanz, jedoch muss die Struktur der Verteilung der Werte bzw. das Verteilungsmuster näher untersucht werden. In der Abb. 5.9 sind die Delta-Planes abgebildet, die für die bereits verwendeten Distanzen zwischen 750 und 3250 mm erzeugt worden sind. Die Abweichung, gegeben als Anzahl der Quantisierungsstufen, ist farbig dargestellt. Zwar lagen bei diesem Experiment die Extremwerte im Bereich zwischen -11.76 und 14.36, jedoch wurde die Farbskala auf den Wertebereich von -5 bis 5 normiert, um eine bessere Farbauflösung im mittleren Bereich zu erhalten. In Abhängigkeit von der Distanz sind verschiedene Strukturen in der Werteverteilung zu erkennen. Die stärksten Unterschiede können bei geringen Distanzen beobachtet werden. Besonders starke Änderungen sind in den Ecken der Delta-Plane erkennbar.

Um die distanzbedingten Veränderungen der Struktur weiter zu untersuchen, wurde der Verlauf unterschiedlicher minimaler und maximaler Werte der Delta-Planes für steigende Distanzen ermittelt. Das Resultat der Auswertung ist in Abb. 5.10 dargestellt. Zunächst wurde für jede Delta-Plane das absolute Minimum sowie das Maximum ermittelt (schwarze Kurven). Da diese potenziell Ausreißer sein könnten, wurde zusätzlich wie folgt verfahren. Alle Einträge der jeweiligen Delta-Plane sind in einen Vektor geschrieben und sortiert worden. Anschließend wurde von den n kleinsten Werten der Mittelwert berechnet, wobei n 1% der Gesamtanzahl der Vektoreinträge entspricht. Weiterhin ist der Mittelwert für die n größten Werte bestimmt worden. Diese beiden Mittelwerte sind durch die roten Kurven angegeben. Das beschriebene Vorgehen wurde analog für 0.1% der Vektoreinträge durchgeführt (blaue Kurven). Zusätzlich ist der Mittelwert über alle Vektoreinträge berechnet worden (grüne Kurve).

Über alle Distanzen hinweg lag der Mittelwert sehr stabil zwischen 0.0372 und 0.0092 und somit nahe bei Null. Die drei Kurven der kleinsten Werte (minimaler Wert, die kleinsten 0.1% sowie die kleinsten 1% aller Werte) sind in der Abb. 5.10 gestrichelt dargestellt und weisen einen deutlichen Anstieg bis zu einer Distanz von 1250 mm auf.

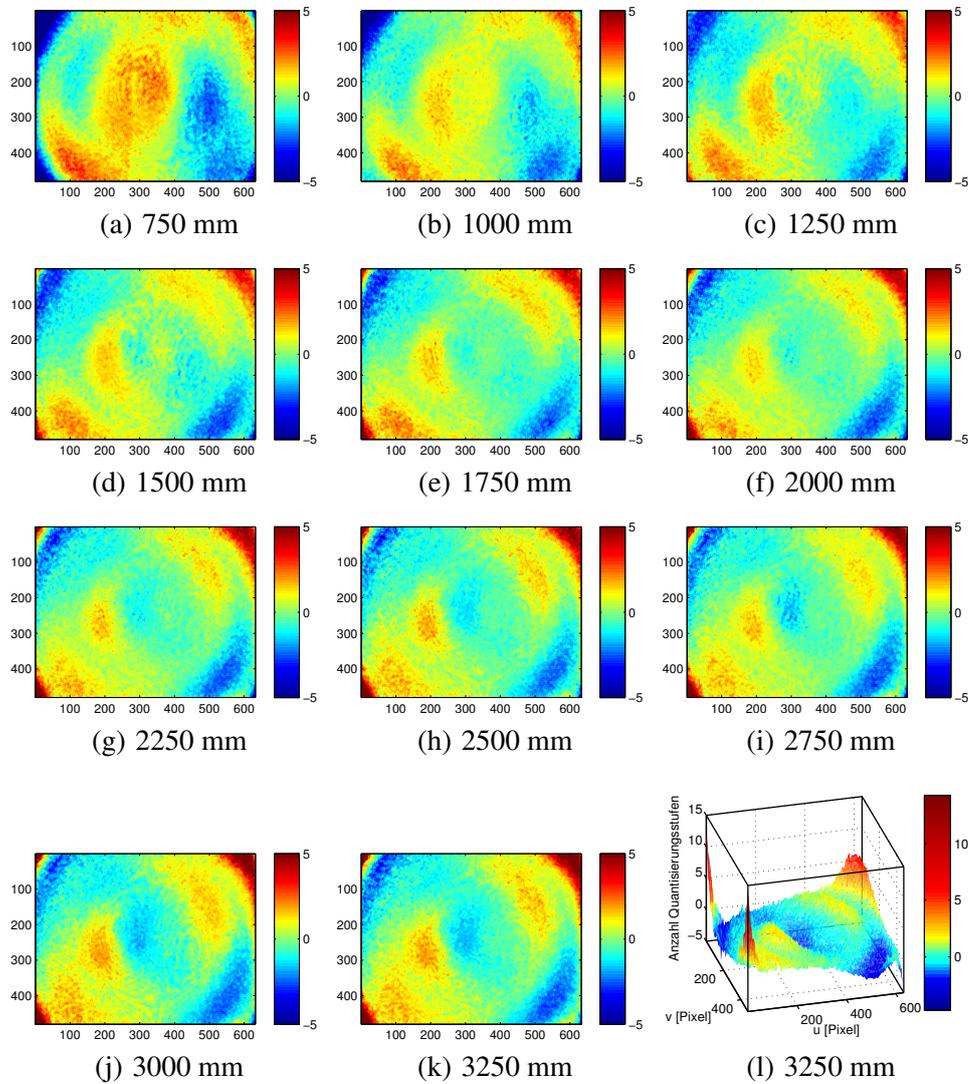


Abbildung 5.9: Abhängigkeit der Struktur der Delta-Planes (Verteilung der Werte bzw. der Anzahl der Quantisierungsstufen) von der Distanz. Die Farbskalen sind bei den Abb. (a) bis (k) auf den Bereich -5 bis 5 Quantisierungsstufen festgelegt. Besonders stark zeigen sich die strukturellen Unterschiede bei den geringen Distanzen. In (l) ist die Delta-Plane für 3250 mm als 3D-Höhenprofil dargestellt, worin die starke Abweichung in den Ecken besonders deutlich wird. Insgesamt bewegt sich die Anzahl der Quantisierungsstufen im Bereich -12 bis 15, siehe Abb. 5.10.

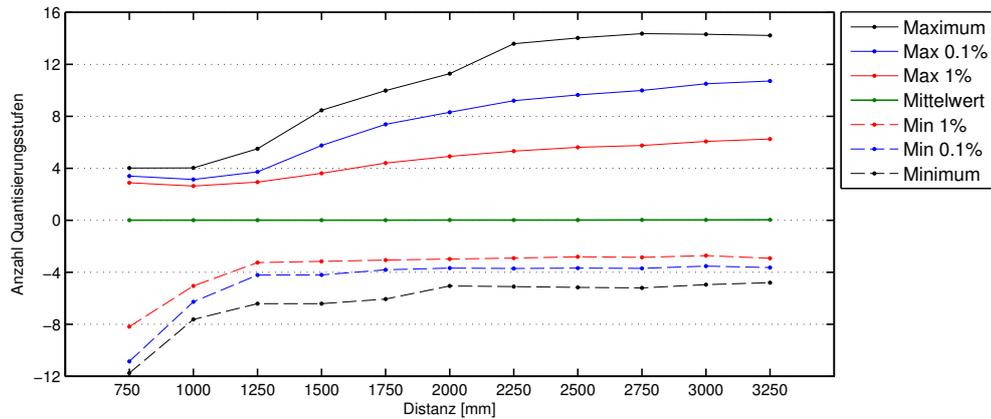


Abbildung 5.10: Visualisierung der Verläufe der minimalen (gestrichelte Linien) und der maximalen Werte (durchgezogene Linien). Neben dem absoluten Minimum bzw. Maximum der jeweiligen Delta-Plane (schwarz) wird, nach einer Sortierung aller Abweichungswerte, der Mittelwert der n kleinsten bzw. größten Werte bestimmt und in rot eingezeichnet. Dabei entspricht n 1% aller der in der Delta-Plane gespeicherten Werte. Analog werden die Werte für 0.1% betrachtet (blau). Der Mittelwert aller Abweichungswerte der Delta-Plane (grün) liegt durchgängig nahe bei Null. Eine Abhängigkeit der dargestellten Werte, insbesondere bei geringen Distanzen, ist deutlich zu erkennen.

Die entsprechenden größten Werte sind mit durchgezogenen Kurven dargestellt. Diese weisen die stärksten Veränderungen im mittleren Distanzbereich zwischen 1250 und 2250 mm auf und flachen zum Ende hin ab. Diese Verläufe verdeutlichen, dass die Änderung der Struktur für Distanzen ab 2250 mm zwar nur noch gering ausfallen. Jedoch besteht, insgesamt betrachtet, insbesondere unterhalb von 1250 mm eine deutliche Abhängigkeit von der Distanz. Die Beobachtung ist auch anhand der Abb. 5.9 nachvollziehbar. Das Auseinanderklaffen der durchgezogenen Linien (positive Abweichungswerte in der Delta-Plane) in Abb. 5.10 ist darauf zurückzuführen, dass die Extremwerte (bis zu 14.36) verhältnismäßig selten vorkommen, siehe rot gefärbte Bereiche in den Ecken der Delta-Plane in Abb. 5.9(1).

5.2.3.6 Multi-Delta-Plane basierte Korrektur unter Berücksichtigung der Messdistanz

Wie in Abschnitt 5.2.3.5.2 verdeutlicht, ist die Struktur der Werteverteilung der Delta-Plane abhängig von der Distanz. Als Konsequenz aus dieser Beobachtung wird ein Korrekturansatz vorgeschlagen, der dieser Abhängigkeit gerecht wird. Der in Abschnitt 5.2.3.3 beschriebene Korrekturansatz ging davon aus, dass es nur eine Delta-Plane gibt. Dieser Ansatz wird im Folgenden auf mehrere Delta-Planes erweitert, wobei für die n einzelnen Delta-Planes $\Delta P_i, i = 0, \dots, n - 1$, gelten soll, dass diese für unterschiedliche Distanzen erzeugt worden sind, ähnlich wie dies im Abschnitt 5.2.3.5 durchgeführt wurde. Diese Erweiterung wird als Multi-Delta-Plane bezeichnet und

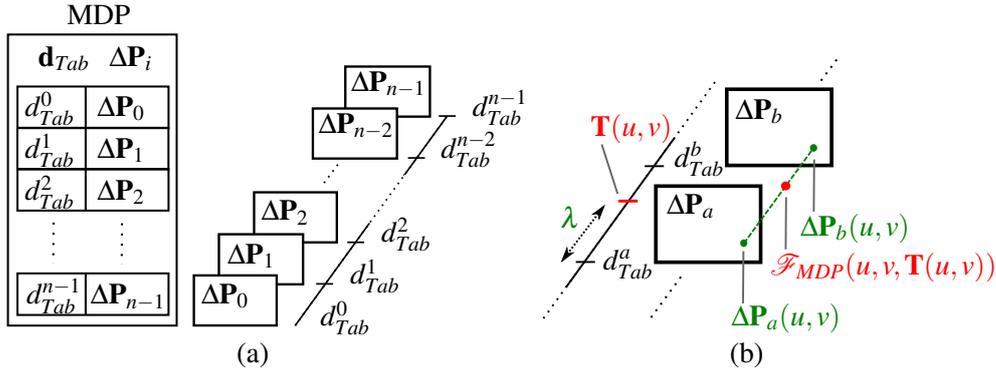


Abbildung 5.11: In (a) ist der Aufbau der Multi-Delta-Plane (MDP) verdeutlicht, die im Wesentlichen aus n Delta-Planes ΔP_i sowie einer dazugehörigen Distanztabelle mit den Distanzen d_{Tab}^i besteht. Dabei entsprechen die d_{Tab}^i den Distanzen, in denen die jeweilige Delta-Plane ΔP_i erzeugt worden ist. (b) verdeutlicht die Berechnung des Korrekturfaktors durch die Funktion $\mathcal{F}_{MDP}(u, v, \mathbf{T}(u, v))$ (rot), basierend auf der Interpolation der entsprechenden Einträge (grün) aus den beiden benachbarten Delta-Planes ΔP_a und ΔP_b , deren Distanzen d_{Tab}^a und d_{Tab}^b das Intervall bilden, in dem der aktuell zu korrigierende Tiefenwert $\mathbf{T}(u, v)$ liegt. Dabei ist λ der Interpolationsfaktor, der zur Berechnung von $\mathcal{F}_{MDP}(u, v, \mathbf{T}(u, v))$ genutzt werden kann.

mit MDP abgekürzt. Zusätzlich zu den Delta-Planes speichert die MDP eine Tabelle, in der die Distanzen hinterlegt sein müssen, in denen die jeweilige Delta-Plane ΔP_i erzeugt worden ist. Diese Distanzen seien mit $\mathbf{d}_{Tab} = (d_{Tab}^0, d_{Tab}^1, \dots, d_{Tab}^{n-1})$ bezeichnet, wobei jedes d_{Tab}^i die Distanz speichert, für die ΔP_i erzeugt worden ist. Diese Distanz wird direkt durch den Algorithmus 3 in Abschnitt 5.2.3.1 berechnet. Dabei gilt $d_{Tab}^i < d_{Tab}^j \quad \forall i < j$. Abb. 5.11(a) verdeutlicht den wesentlichen Aufbau der MDP.

Sei $\mathbf{T}(u, v)$ ein aktuell gegebener Tiefenwert aus dem Tiefenbild \mathbf{T} an der Position (u, v) . Analog zur Gleichung (5.5) in Abschnitt 5.2.3.3, sei der korrigierte Tiefenwert $\mathbf{T}'(u, v)$ gegeben durch

$$\mathbf{T}'(u, v) = \mathbf{T}(u, v) + \mathcal{F}_{MDP}(u, v, \mathbf{T}(u, v)) \cdot f_d(\mathbf{T}(u, v)). \quad (5.13)$$

Dabei bezeichnet $\mathcal{F}_{MDP}(u, v, \mathbf{T}(u, v))$ eine orts- und tiefenabhängige Funktion, die einen Korrekturfaktor liefert und auf der MDP basiert. Dieser Korrekturfaktor steht in Gleichung (5.13) an der Stelle, an der ursprünglich der direkte Zugriff auf die einfache Delta-Plane erfolgte. Die Ermittlung dieses Korrekturfaktors wird im Folgenden beschrieben und durch die Abb. 5.11(b) verdeutlicht.

Zunächst werden anhand des aktuell gegebenen Tiefenwerts $\mathbf{T}(u, v)$ die beiden Indizes a und b identifiziert, so dass $d_{Tab}^a \leq \mathbf{T}(u, v) < d_{Tab}^b$ mit $b = a + 1$ gilt. Sind a und b bekannt, so ergibt sich der gesuchte Korrekturfaktor durch die lineare Interpolation der entsprechenden Einträge der am besten geeigneten Delta-Planes ΔP_a und ΔP_b . Dabei ist ihre Eignung dadurch gegeben, dass die Distanz, in der sie erzeugt worden sind, der aktuell betrachteten Distanz $\mathbf{T}(u, v)$ am nächsten liegt.

Algorithmus 4 : Berechnung des Korrekturfaktors mit Hilfe der Multi-Delta-Plane.

Input : Aktueller Tiefenwert $\mathbf{T}(u, v)$, Anzahl der Delta-Planes n , aktuelle Position (u, v) ,
Delta-Planes $\Delta\mathbf{P}_0, \dots, \Delta\mathbf{P}_{n-1}$, Distanztabelle $d_{Tab}^0, d_{Tab}^1, \dots, d_{Tab}^{n-1}$.

Output : Korrekturfaktor $\mathcal{F}_{MDP}(u, v, \mathbf{T}(u, v))$.

```

i := 0
while  $i < n \wedge \mathbf{T}(u, v) > d_{Tab}^i$  do
  |  $i := i + 1$ 
 $a := i - 1$ 
 $b := i$ 
if  $a < 0$  then
  |  $a := 0$ 
if  $b \geq n$  then
  |  $b := n - 1$ 
 $\lambda := 0$ 
if  $a \neq b$  then
  |  $\lambda := \frac{\mathbf{T}(u, v) - d_{Tab}^a}{d_{Tab}^b - d_{Tab}^a}$ 
 $\mathcal{F}_{MDP}(u, v, \mathbf{T}(u, v)) := (1 - \lambda)\Delta\mathbf{P}_a(u, v) + \lambda\Delta\mathbf{P}_b(u, v)$ 

```

Der Interpolationsfaktor λ kann durch

$$\lambda = \frac{\mathbf{T}(u, v) - d_{Tab}^a}{d_{Tab}^b - d_{Tab}^a} \quad (5.14)$$

bestimmt werden. Der Korrekturfaktor an der Position (u, v) für die Tiefe $\mathbf{T}(u, v)$ ist schließlich gegeben durch

$$\mathcal{F}_{MDP}(u, v, \mathbf{T}(u, v)) = (1 - \lambda)\Delta\mathbf{P}_a(u, v) + \lambda\Delta\mathbf{P}_b(u, v). \quad (5.15)$$

Während die Distanz bei der Korrektur unter Verwendung der einfachen Delta-Plane nicht beachtet wurde, wird bei der durch die Gleichung (5.13) beschriebenen Korrektur die individuelle Distanz an der aktuellen Position im Tiefenbild mit berücksichtigt. Dies ermöglicht es, auch Tiefenbilder mit komplexen Szenen, bestehend aus verschiedenen Objekten und unterschiedlichen Distanzen, zu korrigieren.

Der Algorithmus 4 fasst die Berechnung des Korrekturfaktors unter Berücksichtigung von Sonderfällen der Indizes zusammen. Dabei wird für den Fall, dass für eine aktuell vorliegende Distanz $\mathbf{T}(u, v) \notin [d_{Tab}^0, d_{Tab}^{n-1}]$ gilt, diese also außerhalb des durch die MDP abgedeckten Tiefenintervalls liegt, der Korrekturwert direkt aus $\Delta\mathbf{P}_0$ bzw. $\Delta\mathbf{P}_{n-1}$ entnommen. Es sei jedoch angemerkt, dass alternativ auch eine lineare Extrapolation außerhalb des Intervalls basierend auf $\Delta\mathbf{P}_0$ und $\Delta\mathbf{P}_1$ bzw. $\Delta\mathbf{P}_{n-2}$ und $\Delta\mathbf{P}_{n-1}$ möglich wäre.

Um die korrigierten Tiefenwerte $\mathbf{T}(u, v)$ auf die quantisierten Tiefenwerte abzubilden, kann analog zum Abschnitt 5.2.3.3 die Quantisierungsfunktion $\mathcal{F}_{QuantMapper}(x)$ (siehe Gleichung (5.7)) verwendet werden. Dann sind die quantisierten, korrigierten Tiefenwerte gegeben durch

$$\mathbf{T}'_Q(u, v) = \mathcal{F}_{QuantMapper}(\mathbf{T}(u, v) + \mathcal{F}_{MDP}(u, v, \mathbf{T}(u, v)) \cdot f_d(\mathbf{T}(u, v))). \quad (5.16)$$

5.2.4 Evaluierung

Zunächst werden die Korrekturleistung des Delta-Plane basierten Ansatzes (vgl. Abschnitt 5.2.3.3) evaluiert und dessen Grenzen bei Betrachtung unterschiedlicher Distanzen aufgezeigt. Anschließend wird der Multi-Delta-Plane (MDP) basierende Ansatz (siehe Abschnitt 5.2.3.6) für die Korrektur herangezogen, der die Distanz mit berücksichtigt und eine deutlich bessere Korrekturleistung liefert. Bei dem MDP basierenden Ansatz werden konkret neun Delta-Planes eingesetzt, wodurch für das Korrekturmodell die Speicherung von neun Skalarwerten pro Pixel notwendig ist. Es sei angemerkt, dass bei den nachfolgenden Evaluierungsexperimenten eine strikte Trennung zwischen den Tiefenbildern, die zur Erzeugung der Delta-Planes bzw. der MDP verwendet worden sind, und den Tiefenbildern, anhand denen die Korrekturleistung ermittelt wird, eingehalten wird.

5.2.4.1 Delta-Plane basierte Korrekturleistung

Für das erste Experiment wurde die Delta-Plane $\Delta\mathbf{P}$ aus 8 frontal aufgezeichneten Tiefenbildern erzeugt. Dabei ist die Kinect in einem Abstand von 1230 mm zur Wand positioniert worden. Zu einem späteren Zeitpunkt wurde ein Tiefenbild \mathbf{T} in einer vergleichbaren Entfernung gewonnen und ist in Abb. 5.12(a) in der 3D-Ansicht bzw. in Abb. 5.12(c) in 2D dargestellt. Darin sind die Abweichungen von einer idealen Ebene deutlich erkennbar. Anschließend wurde das Tiefenbild \mathbf{T} unter Verwendung der Delta-Plane $\Delta\mathbf{P}$ mit dem in Abschnitt 5.2.3.3 beschriebenen Verfahren korrigiert. Das Resultat \mathbf{T}' ist in Abb. 5.12(b) und 5.12(d) dargestellt. Durch die Korrektur konnte das komplexe ortsabhängige Fehlermuster entfernt werden, und die rekonstruierten Tiefenwerte entsprachen, bis auf das vorhandene Rauschen, dem erwarteten, planaren Objekt. Bei der Betrachtung der Abb. 5.12(d) deutet die Verteilung der Tiefenwerte eine geringfügige Schräge an. Diese ist, abgesehen von dem Rauschmuster, daran zu erkennen, dass eine Tendenz zu größeren Distanzwerten in der unteren rechten und zu kleineren in der oberen linken Ecke vorliegt. Die Schräge ist darauf zurückzuführen, dass die Kinect bei der Aufzeichnung von \mathbf{T} nicht ideal frontal zur Wand ausgerichtet war.

Um die Korrekturleistung neben dem visuellen Vergleich zusätzlich statistisch zu beurteilen, wurden für die Tiefenwerte in \mathbf{T} und \mathbf{T}' die dazugehörigen Punktwolken \mathbf{W} und \mathbf{W}' , wie in Abschnitt 4.5.4 beschrieben, berechnet. Anschließend wurden jeweils eine Ausgleichsebene in jede dieser Punktwolken gelegt (vgl. Abschnitt 5.2.3.1) und die Abweichungen der Wolkenpunkte ausgewertet. Der Mittelwert aller Abweichungen lag bei $-8.3 \cdot 10^{-6}$ mm. Die Standardabweichung verringerte sich insgesamt von 5.35 auf 2.46 mm und konnte somit auf unter 46% des Ausgangswertes reduziert werden. Die mittlere absolute Abweichung von der Ausgleichsebene lag vor der Korrektur bei 4.40 mm und nach der Korrektur bei 1.84 mm. Zusätzlich wurden die minimalen Messwerte ermittelt, die vorher bei -21.12 mm und nachher bei -15.87 mm lagen. Analog sind die maximalen Werte betrachtet worden, bei denen eine Änderung von 28.78 auf 13.99 mm beobachtet werden konnte. Das Balkendiagramm in Abb. 5.12(e) fasst die Auswertung zusammen.

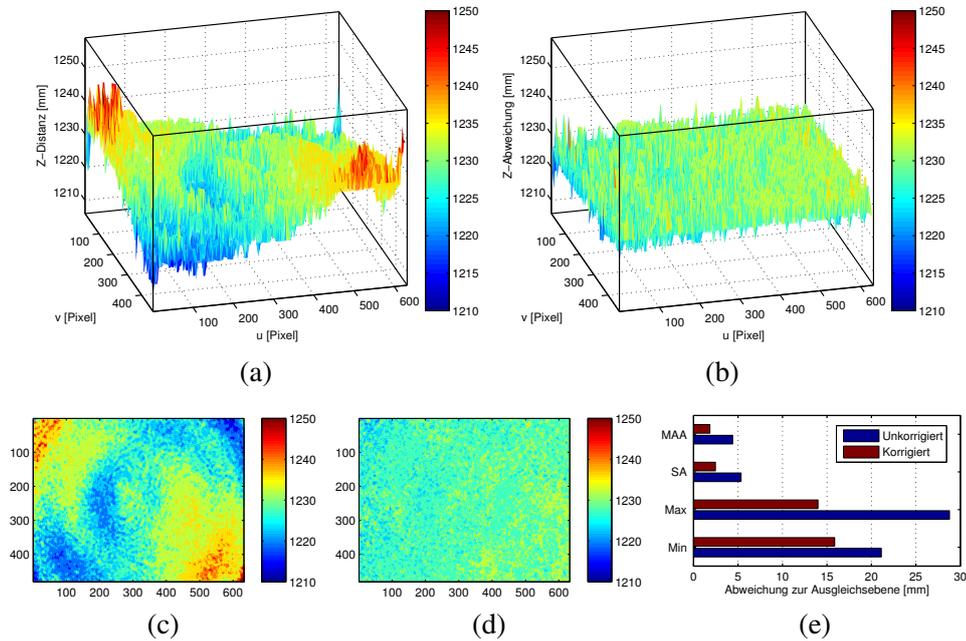


Abbildung 5.12: In (a) und (c) sind die 3D- bzw. 2D-Visualisierungen eines unkorrigierten Tiefenbildes bei der Aufnahme einer Wand dargestellt. Das Ergebnis nach der Delta-Plane basierten Korrektur ist in (b) und (d) zu sehen. Die Farbskala gibt die Distanzen in Millimetern an. Das Balkendiagramm in (e) vergleicht statistische Werte vor (blau) und nach der Korrektur (braun), wobei Min abkürzend für den Minimalwert, Max für Maximalwert, SA für die Standardabweichung und MAA für die mittlere absolute Abweichung stehen. Das Vorzeichen der Minimalwerte wurde invertiert, um eine kompaktere Darstellung zu ermöglichen.

5.2.4.2 Delta-Plane versus Multi-Delta-Plane

Die Delta-Plane, die in Abschnitt 5.2.4.1 verwendet worden ist, wurde in diesem Abschnitt erneut genutzt. Diese ist für eine Distanz von 1230 mm generiert worden. Der Mittelwert der erfassten Tiefenwerte der für den Test genutzten Tiefenbilder betrug 739, 982, 2451 und 2937 mm. Diese Tiefenbilder sind in den Abbildungen 5.13(a), 5.13(d), 5.13(g) und 5.13(j) dargestellt (linke Spalte der Abb. 5.13). Die mittlere Spalte in Abb. 5.13 zeigt die Resultate nach der Korrektur unter Verwendung der Delta-Plane. Es ist zu erkennen, dass das Fehlermuster zwar geringer ausfällt, jedoch in keinem der vier Tiefenbilder entfernt werden konnte. Besonders intensiv ist der verbliebene Fehler in den Ecken der Tiefenbilder sichtbar, was insbesondere anhand der 3D-Ansicht in der Abb. 5.13(n) deutlich wird. Dieser Korrekturfehler ist auf die Beobachtungen aus Abschnitt 5.2.3.5.2 zurückzuführen, in denen eine Distanzabhängigkeit der Struktur der Korrekturwerte in der Delta-Plane beobachtet wurde. Da die vorliegenden Bilder in einer von der Erzeugungsdistanz der Delta-Plane abweichenden Entfernung aufgezeichnet wurden, kann der Fehler nicht vollständig beseitigt werden.

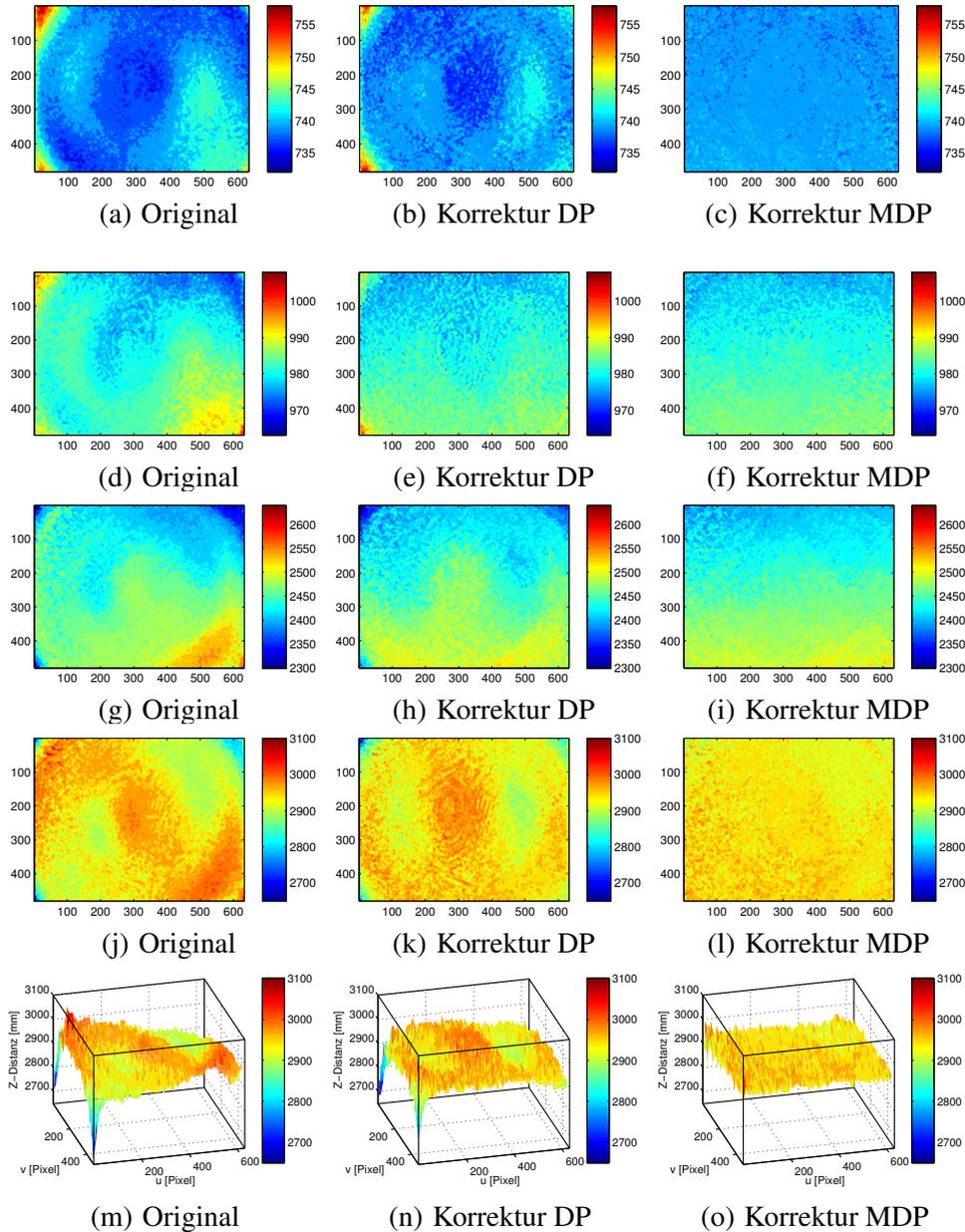


Abbildung 5.13: Die linke Spalte zeigt unkorrigierte Tiefenbilder, die in unterschiedlichen Entfernungen aufgezeichnet worden sind. Die mittlere gemessene Distanz lag bei 739 mm (a), 982 mm (d), 2451 mm (g) bzw. 2937 mm (j). Die mittlere Spalte visualisiert die Korrekturergebnisse bei Verwendung des Delta-Plane basierten Ansatzes, wobei die Delta-Plane für eine Distanz von 1230 mm erzeugt worden ist. Das distanzabhängige komplexe Fehlermuster konnte nicht beseitigt werden. In der rechten Spalte hingegen konnte das Muster mit Hilfe der MDP korrigiert werden. Es sind lediglich leichte Neigungen der beobachteten Wand in (f) und (j) erkennbar. Die zugrundeliegenden Daten sind bei den letzten beiden Zeilen identisch, jedoch verdeutlicht die 3D-Ansicht insbesondere die Korrekturleistung in den Tiefenbildecken.

	Block 1			Block 2		
	Original	DP	MDP	Original	DP	MDP
Mittl. Tiefe [mm]	739.1	739.0	739.0	981.6	981.2	981.2
Min [mm]	-7.76	-7.03	-4.75	-14.77	-10.66	-10.66
Max [mm]	19.48	18.34	5.67	25.96	20.41	10.05
SA [mm]	2.71	2.12	0.79	3.58	2.32	1.65
MAA [mm]	1.95	1.39	0.50	2.72	1.76	1.29
	Block 3			Block 4		
	Original	DP	MDP	Original	DP	MDP
Mittl. Tiefe [mm]	2450.7	2450.4	2450.5	2936.8	2936.3	2936.3
Min [mm]	-194.6	-179.3	-62.2	-301.9	-263.2	-95.0
Max [mm]	139.1	67.1	85.4	143.2	91.1	113.5
SA [mm]	22.61	18.28	11.29	34.47	28.67	16.18
MAA [mm]	16.72	13.70	8.87	27.12	21.52	12.73

Tabelle 5.1: Statistische Auswertung der Distanzwerte für die in Abb. 5.13 dargestellten Tiefenbilder. Die Blocknummer gibt die jeweilige Zeile in der Abb. 5.13 an. Die erste Zeile zeigt den Mittelwert aller Tiefenwerte des jeweiligen Tiefenbildes. Für die Abweichungen der Wolkenpunkte von den entsprechenden Ausgleichsebenen wurden der minimale (Min) und der maximale Wert (Max), die Standardabweichung (SA) und die mittlere absolute Abweichung (MAA) berechnet. Neben dem unkorrigierten Tiefenbild (Original) sind auch die Ergebnisse für die mit dem Delta-Plane basierten Verfahren (DP) sowie mit dem Multi-Delta-Plane basierten Verfahren (MDP) angegeben.

Zu den visuellen Auswertungen ist zusätzlich eine statistische Auswertung durchgeführt worden. Zum einen wurde der Mittelwert über alle Tiefenwerte berechnet. Dabei ist festgestellt worden, dass das Korrekturverfahren in der Praxis nahezu keine Veränderung dieses Mittelwerts zur Folge hatte (Mittelwert nach Korrektur betrug 99.962% bis 99.983% des Ausgangswertes). Zum anderen wurden erneut die Ausgleichsebenen für die zu den Tiefenbildern gehörenden Punktwolken bestimmt. Die minimale und maximale Abweichung, die Standardabweichung und die mittlere absolute Abweichung sind in Tabelle 5.1, zusammen mit den genannten Mittelwerten der Tiefenwerte, für alle vier Bilder angegeben. Dabei wurden Angaben sowohl für das unkorrigierte Tiefenbild (Original) als auch für das durch die Delta-Plane korrigierte Tiefenbild gemacht. Die visuell erkennbare Verringerung des Fehlers konnte auch numerisch, z.B. anhand der Standardabweichung, beobachtet werden, die für das erste Bild (739.1 mm) von 2.71 auf 2.12 mm, für das zweite (981.6 mm) von 3.58 auf 2.32 mm, für das dritte (2450.7 mm) von 22.61 auf 18.25 mm und das vierte (2939.3 mm) von 34.47 auf 28.67 mm reduziert werden konnte. Dies entspricht einer Reduktion auf 78.0%, 64.8%, 80.8% bzw. 83.2%. Um einen schnelleren Überblick über diese und die weiteren Korrekturleistungen zu ermöglichen, stellt die Abb. 5.14 die ermittelten Größen im Verhältnis zu den Auswertungen des Originalbildes in Form eines Balkendiagramms dar. Dabei wird das Verhältnis in Prozent ausgedrückt.

Wie in Abb. 5.13 gezeigt, deutet die visuelle Auswertung der Delta-Plane basierten Korrektur auf einen verbliebenen Fehler hin (mittlere Spalte). Um zu untersuchen, inwieweit das MDP basierte Verfahren in der Lage ist, eine bessere Korrekturleistung zu bieten, wurde zunächst eine MDP erzeugt. In Abschnitt 5.2.3.5.2 konnte beobach-

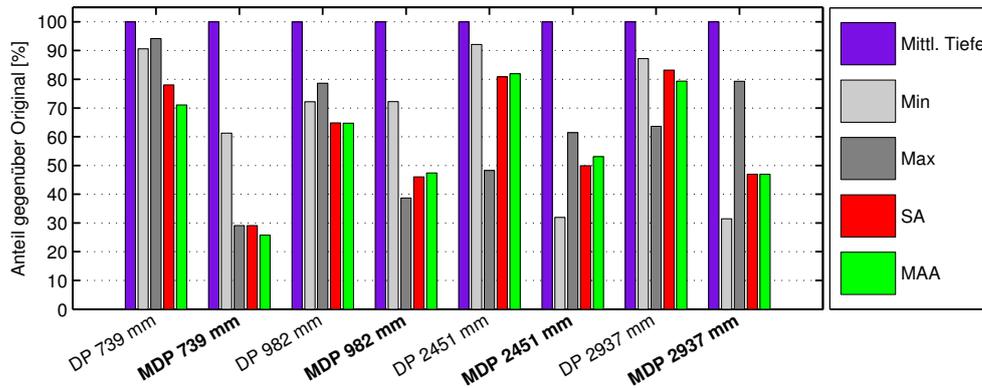


Abbildung 5.14: Kompakte Visualisierung der statistischen Auswertung aus Tabelle 5.1. Dargestellt sind jeweils die ermittelten Werte der korrigierten Bilder unter Verwendung der Delta-Plane (DP) bzw. der Multi-Delta-Plane (MDP) im Verhältnis zum unkorrigierten Originalbild. Das Verhältnis ist in Prozent ausgedrückt.

tet werden, dass die Struktur der Delta-Planes innerhalb der Distanzen bis 2250 mm einer stärkeren Veränderung unterliegt, als bei Distanzen oberhalb dieser Grenze (vgl. Abb. 5.10). Folglich wurden für die MDP sieben Delta-Planes für die Distanzen zwischen 750 und 2250 mm in 250 mm Schritten und zwei weitere in 500 mm Schritten bis zu einer Distanz von 3250 mm verwendet. Für die Erzeugung jeder der insgesamt neun Delta-Planes entsprechend Abschnitt 5.2.3.1, wurden jeweils neun Tiefenbilder in den genannten Distanzen aufgezeichnet.

Wie an der rechten Spalte der Abb. 5.13 erkennbar, konnte der Fehler mit Hilfe des MDP basierten Verfahrens, bis auf das Rauschen, behoben werden. Alle mit der MDP korrigierten Tiefenbilder lassen die planare Struktur der Wand erkennen, wobei in den Abb. 5.13(f) und 5.13(i) ein Verlauf der Tiefenwerte zu sehen ist. Dieser ist erneut darauf zurückzuführen, dass die Kinect bei der Aufnahme der entsprechenden Bilder (relativ zur Wand) geringfügig um die horizontale Achse geneigt war. Die korrigierten Tiefenbilder in den Abb. 5.13(i) und 5.13(l) sind Beispiele für den funktionierenden Interpolationsansatz bei Verwendung der MDP. Die dazugehörigen unkorrigierten Originalbilder haben nämlich eine mittlere Distanz von 2451 bzw. 2937 mm und liegen somit in etwa mittig in dem Intervall zwischen den beiden für die Interpolation verwendeten Delta-Planes der MDP, deren Erzeugungsdistanzen 2250 und 2750 mm bzw. 2750 und 3250 mm betragen haben.

Während bei der Verwendung des Delta-Plane basierten Ansatzes insbesondere in den Ecken noch deutliche Fehler in der 3D-Ansicht zu erkennen waren (siehe Abb. 5.13(n)), konnten diese mit Hilfe des MDP basierten Ansatzes beseitigt werden (siehe Abb. 5.13(o)). In der unteren linken Ecke des Tiefenbildes lag der dort gemessene, unkorrigierte Tiefenwert bei 2649 mm. Dies entspricht einer Abweichung von -288 mm vom Mittelwert (2937 mm). Mit der Delta-Plane konnte die Abweichung lediglich auf -183 mm (63.5%) reduziert werden, während mit der MDP eine Reduktion bis auf -20 mm (6.9%) möglich war.

Die weiterführenden statistischen Auswertungen in Tabelle 5.1 bestätigen ebenfalls, dass die Korrekturleistung des MDP basierten Ansatzes die des Delta-Plane basierten Ansatzes übertrifft. Wie dem Balkendiagramm in Abb. 5.14 zu entnehmen ist, konnte die Standardabweichung der Abstände zur Ausgleichsebene bei allen vier mit dem MDP basierten Ansatz korrigierten Tiefenbildern auf unterhalb von 50% des Ausgangswertes reduziert werden. Bei dem Tiefenbild mit der durchschnittlichen Distanz von 739 mm konnte die Standardabweichung sogar auf 29.1% herabgesetzt werden (von 2.71 auf 0.79 mm), während der Delta-Plane basierte Ansatz nur eine Reduzierung auf 78.0% (von 2.71 auf 2.12 mm) erreichte. Die durchschnittliche Distanz aller Tiefenwerte blieb nach der Korrektur des Tiefenbildes nahezu unverändert (99.96% bis 99.99%), so dass keine signifikante Verschiebung der Tiefendaten bei Verwendung der MDP beobachtet werden konnte. Die Gegenüberstellung der weiteren statistischen Größen ist ebenfalls der Abb. 5.14 zu entnehmen.

5.2.4.3 Untersuchung typischer Innenraumszenarien

In den vorangegangenen Abschnitten wurde lediglich ein einziges, bildfüllendes planares Objekt verwendet, welches bis auf geringfügige Abweichungen frontal zur Kinect ausgerichtet gewesen ist. Im Folgenden werden typische Innenraumszenen betrachtet, bei denen mehr Freiheitsgrade (Ausrichtung, Anzahl verschiedener Objekte, Distanzumfang) gegeben sind, und anhand denen eine visuelle Verdeutlichung des Einflusses des ortsabhängigen, komplexen Fehlers aufgezeigt wird. Zudem wird gezeigt, inwiefern der Fehler mit Hilfe des MDP basierten Verfahrens (vgl. Abschnitt 5.2.3.6) behoben werden kann.

In Abb. 5.15 sind insgesamt fünf unterschiedliche Ansichten in verschiedenen Räumen gezeigt. Für jede Ansicht sind das Kamerabild (linke Spalte) und sowohl das unkorrigierte Originaltiefenbild (mittlere Spalte) als auch das Tiefenbild nach der Korrektur unter Verwendung des MDP basierten Verfahrens (rechte Spalte) abgebildet. Um eine Darstellung der Tiefenwerte zu erzielen, die gleichzeitig einen hohen Distanzumfang sowie eine differenzierte Anzeige kleinerer Distanzunterschiede ermöglicht, waren die folgenden Schritte notwendig. Zunächst wurde darauf verzichtet, den Wertebereich des Tiefenbildes eindeutig auf den Grauwertebereich abzubilden. Dies hätte aufgrund des großen Tiefenumfanges der Szene und nur 256 zur Verfügung stehenden Grauwertstufen (8 Bit) zur Folge gehabt, dass viele Details verloren gegangen oder für das menschliche Auge nur schwer zu erkennen gewesen wären. Stattdessen wurden Grauwerte periodisch wiederverwendet. Wie in Abschnitt 4.5.7 beschrieben, werden die Tiefenbilder vom OpenNI-Framework in Millimetern zur Verfügung gestellt. Für die Darstellung der Tiefenbilder als Graustufenbild ist zunächst je ein Millimeter auf eine der 256 verfügbaren Graustufen abgebildet worden. Somit kommt es in der Darstellung der Distanz alle 256 mm zu einem Überlauf der Grauwerte, der hier bewusst eingesetzt wird. Anschließend wurden die Grauwerte einer Histogrammglättung unterzogen. Dabei sind häufig vorkommende Werte auf einen größeren Helligkeitsbereich abgebildet worden, wodurch bei diesen Werten der Kontrast erhöht werden konnte. Zuletzt wurden die histogrammgeglätteten Grauwerte auf Falschfarben abgebildet. Durch die beschriebenen Schritte entstanden bei kontinuierlichen Tiefenverläufen deutliche Kanten, die Isolinien ähneln. Diese kommen z.B. in den Abb. 5.15(o) besonders zum

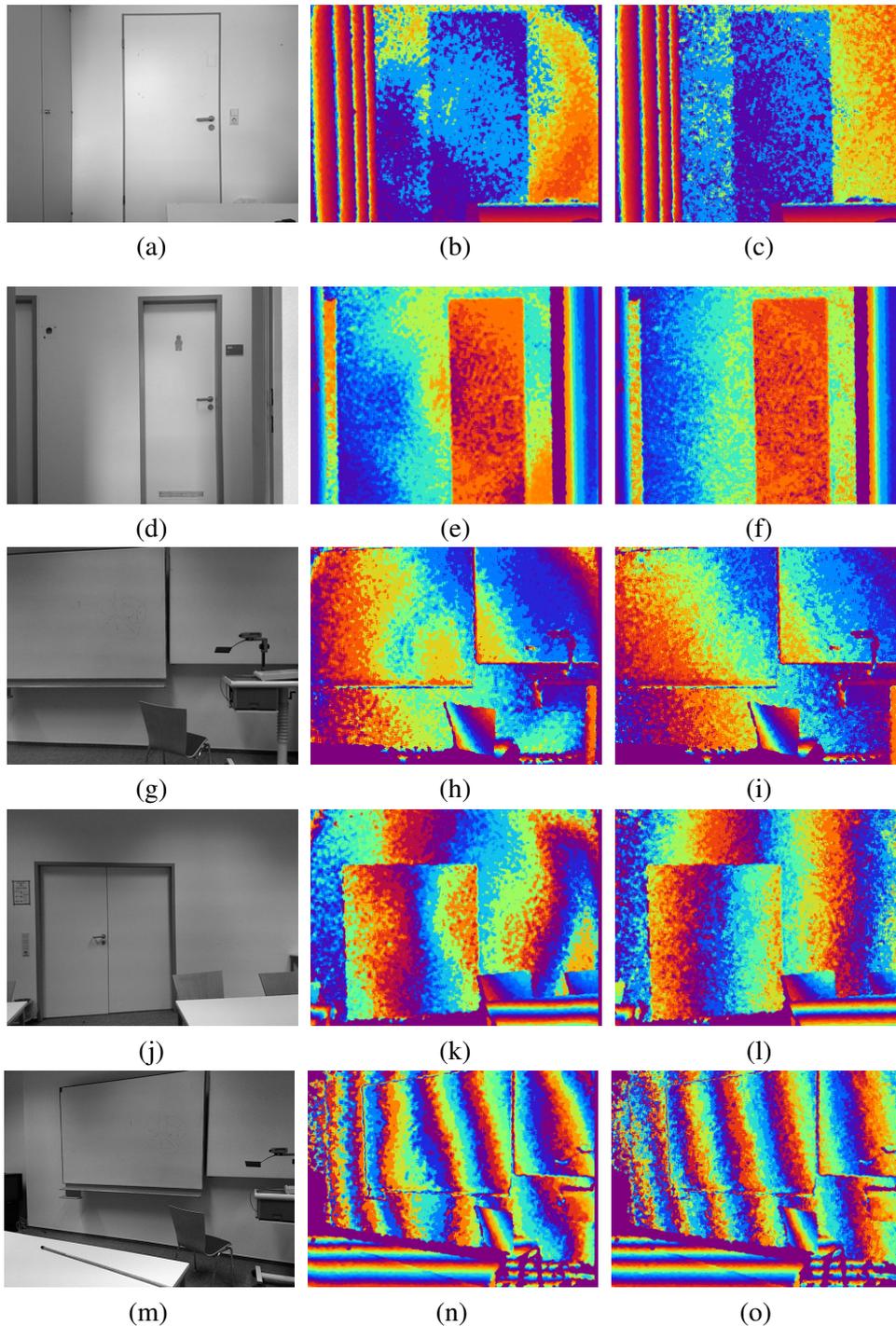


Abbildung 5.15: Visuelle Auswertung verschiedener Ansichten in Innenräumen. Jede Zeile entspricht einer Beispielszene. Die erste Spalte gibt die Kamerabilder wieder. Die dazugehörigen, mit dem komplexen ortsabhängigen Fehler überlagerten Tiefenbilder sind jeweils in der mittleren Spalte in Falschfarben dargestellt. Die MDP basierten Korrekturergebnisse sind in der rechten Spalte gezeigt.

Vorschein. Details zur Histogrammglättung und zur Falschfarbenabbildung können in [GW01] nachgeschlagen werden.

In den unkorrigierten Tiefenbildern ist die Überlagerung mit dem Fehlermuster, das einem Negativabdruck der in Abb. 5.9 dargestellten Delta-Planes ähnelt, in der mittleren Spalte der Abb. 5.15 zu erkennen. Im ersten Beispiel ist rechts neben der Tür eine deutliche mondformige Verfälschung sichtbar, die nach der Korrektur behoben ist. Die eigentlich leicht schräg verlaufende Wand mit den beiden Türen im zweiten Beispiel weist eine Reihe von nicht vorhandenen Höhenprofilen auf, wie z.B. in der Mitte des Tiefenbildes. Nach der Korrektur ist ein gleichmäßiger Verlauf im gesamten Tiefenbildbereich entsprechend der leichten Schräge von links nach rechts zu erkennen. Besonders hervorstechend ist im Tiefenbild des dritten Beispiels die auf der Tafelcke zu erkennende, jedoch in Wirklichkeit nicht vorhandene kreisförmige Auswölbung (gelb), die durch das MDP basierte Verfahren entfernt werden konnte. Die letzten beiden Beispiele zeigen den Fehler anhand von deutlich schräg verlaufenden Objekten (Tafel, Wände, Türen). Durch die genutzte Farbdarstellung, kombiniert mit den schrägen Objekten, wird das periodische Farbmuster deutlich. Die Verläufe ähneln den erwähnten Isolinien. Da die meisten der Objekte planar sind, müssten die Isolinien im Idealfall parallel verlaufen. In den gegebenen unkorrigierten Tiefenbildern sind jedoch deutliche Krümmungen erkennbar. Die erwartete Objektstruktur ist in den korrigierten Bildern deutlich besser wiederzuerkennen.

Im Abschnitt 5.2.4.2 wurde bereits auf die starke Verfälschung in den Eckbereichen der Tiefenbilder hingewiesen. Diese wird im Kontext typischer Innenraumszenen anhand der in Abb. 5.16(a) gezeigten Szene ebenfalls untersucht, in der im Wesentlichen ein Tageslichtprojektor vor zwei an der Wand befestigten großen Tafeln zu sehen ist. Aufgrund der gewählten Farbdarstellung ist der Fehler in den Eckbereichen zwar sichtbar, sein Ausmaß jedoch schwer abzuschätzen (siehe Abb. 5.16(b) und die korrigierte Variante in Abb. 5.16(c)). Deshalb wurden zu den beiden Tiefenbildern die entsprechenden Punktwolken berechnet (vgl. Abschnitt 4.5.4) und diese mit OpenGL¹ dreidimensional visualisiert. Dabei ist die Farbe jedes Punktes in Abhängigkeit von seiner Z-Koordinate ebenfalls mit der Falschfarbendarstellung gerendert worden. Es sei jedoch angemerkt, dass für die 3D-Ansicht eine andere Farbauflösung verwendet wurde, weshalb der Farbverlauf nicht mit dem in den 2D-Darstellungen übereinstimmt. Die 3D-Ansicht der Szene vor und nach der Korrektur ist in den Abb. 5.16(d) und 5.16(e) gezeigt. Bei der Erstellung der 3D-Ansicht ist die Kameraperspektive gegenüber der Ursprünglichen leicht verändert worden, indem die virtuelle Kamera, in Blickrichtung betrachtet, nach oben rechts verschoben und anschließend nach links rotiert worden ist. In dieser Ansicht ist die Verzerrung in den beiden oberen Ecken der Punktwolke deutlich zu erkennen, während sie nach der Korrektur fast vollständig beseitigt werden konnte. Bei genauerer Betrachtung sind mehrere Verformungen und Verzerrungen auch im Bereich der Tafeln erkennbar, während die korrigierte Version der Punktwolke einen planaren Eindruck vermittelt, was auch der realen Geometrie der Objekte entspricht.

¹<http://www.opengl.org/>

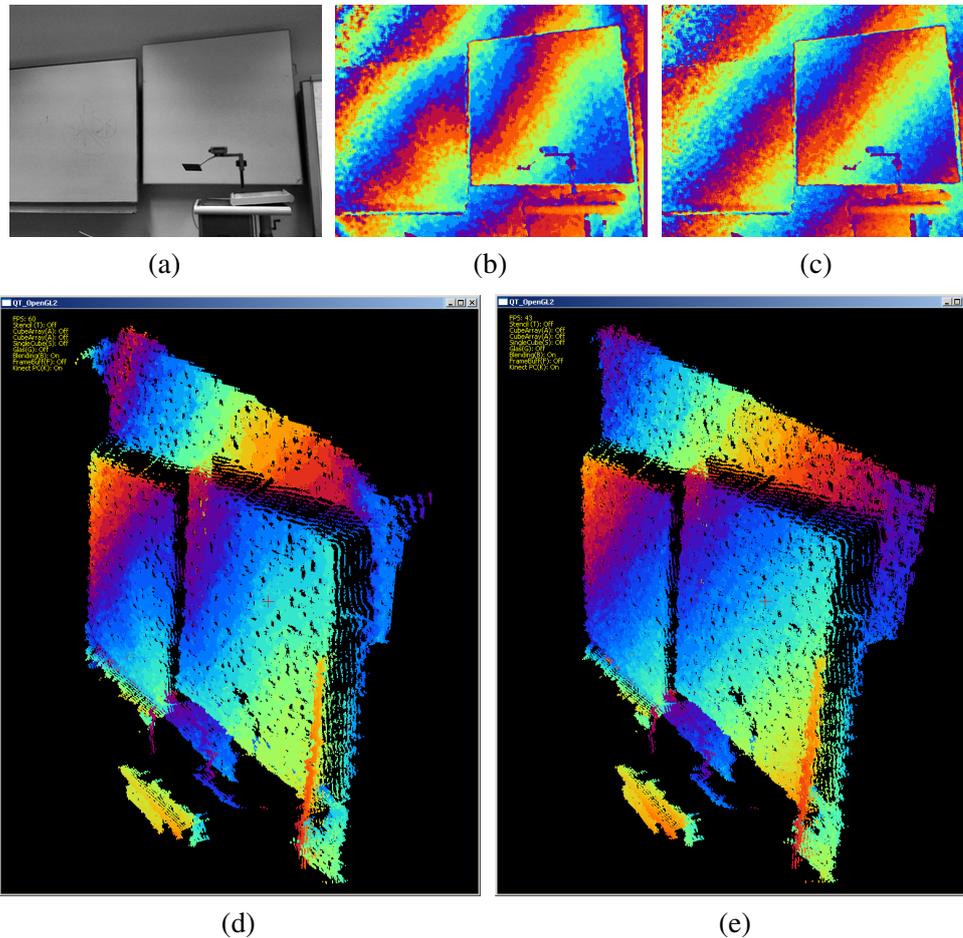


Abbildung 5.16: Verdeutlichung des Einflusses des Fehlermusters auf die dreidimensionale Struktur der durch den Tiefensensor erfassten Szene. Das Kamerabild des gegebenen Szenenausschnitts ist in (a) dargestellt. (b) und (c) zeigen das Tiefenbild vor und nach der Korrektur. Die zu den jeweiligen Tiefenbildern entsprechenden Punktwolken sind in (d) und (e) gegeben. Neben den deutlichen Fehlern in den Bildecken lassen sich bei sorgfältiger Betrachtung weitere Verzerrungen der dargestellten Tafeln erkennen (z.B. obere rechte Ecke der rechten Tafel), welche im korrigierten Bild verringert sind.

5.3 Online-Gewinnung von Referenzmessdaten

Um die Güte der Tiefenmessung des Kinect-Sensors zu beurteilen, ist es notwendig, diese in geeigneter Form mit Referenzmessdaten vergleichen zu können. Ein starres Messszenario stellt eine klassische Möglichkeit zur Gewinnung von Referenzmessdaten dar, weil dabei die Informationen aufgrund der bekannten Geometrie des Messaufbaus zur Verfügung stehen. Dies schränkt allerdings die Gewinnung von Referenzmessdaten auf nur eine gegebene Konfiguration ein. Alternativ könnte z.B. ein Industrie-Roboter verwendet werden, bei dem die Lage und Position des Roboterarms sehr genau bekannt sind, um somit Referenzmessdaten dynamisch zu gewinnen. Beispielsweise wird in [WBRF13] die Genauigkeit des bereits in Abschnitt 4.3.3 genannten LeapMotion-Sensors mit Hilfe des Roboters "Kuka Robot KR 125/3" untersucht. Derartige Roboter sind jedoch oftmals nicht verfügbar oder sehr kostspielig. Eine weitere Alternative stellt der Einsatz von kalibrierten CCD-Kameras dar, um Referenzmessungen zu erhalten. Beispielsweise wird in [LSKK10] eine hochauflösende zusätzliche CCD-Kamera im Rahmen der Kalibrierung einer Time-Of-Flight-Kamera eingesetzt. Der Einsatz von zwei Spiegelreflex-Kameras als Stereo-System zur Referenzdatenermittlung findet z.B. in [SJP11] Verwendung. In [SBK08] wird sogar ein Multi-Kamerasystem eingesetzt, um Referenzmessungen mit Hilfe eines Kalibrierobjekts zu generieren. Solche Systeme liefern in der Regel eine hohe Genauigkeit, jedoch haben sie auch Nachteile. Der Einsatz mehrerer hochwertiger Kameras ist im Vergleich zum Anschaffungspreis der Kinect teuer. Zudem müssen die Kameras in einer starren Konfiguration zueinander montiert werden, was je nach Anzahl, Größe und Gewicht der Kameras aufwendig und ebenfalls kostspielig sein sowie zudem die Mobilität einschränken kann.

Im Folgenden werden einige der genannten Aspekte aufgegriffen und ein Verfahren beschrieben, mit dem Referenzmessdaten für die Kinect gewonnen werden können, und welches die folgenden Vorteile hat:

- Die für das Verfahren notwendige Hardware ist bereits in der Kinect integriert.
- Das Verfahren basiert auf einfachen Kalibrierobjekten, die kostengünstig erstellt werden können.
- Es erlaubt eine hohe Flexibilität bei der Positionierung der Kinect und des Kalibrierobjekts.
- Es ist in der Lage, Referenzmessdaten interaktiv und im laufenden Betrieb (online) zur Verfügung zu stellen, wodurch eine Ad-hoc Überprüfung der Genauigkeit der Tiefendaten ermöglicht wird.

Die Grundidee liegt in der Kalibrierung und Verwendung der bereits in der Kinect verbauten RGB-Kamera zur Auffindung und Lokalisierung eines planaren Kalibrierobjekts mit bekannter Geometrie. Die Nutzung der internen RGB-Kamera wurde z.B. in [CBF11] vorgeschlagen, jedoch diente sie dort als Teil eines Stereo-Systems im Verbund mit der internen IR-Kamera, während sie in dieser Dissertation als Mono-Kamera zur Lagebestimmung des Kalibrierobjekts eingesetzt wird. Ist die Lage des Kalibrierobjekts im Koordinatensystem der internen RGB-Kamera bekannt, wird es in das Koordinatensystem des Tiefensensors transformiert und ist somit auch dort indirekt bekannt. Zeitgleich wird das planare Kalibrierobjekt vom Tiefensensor der Kinect

erfasst und liegt als Tiefenbild vor. Da das Kalibrierobjekt, je nach Entfernung zum Sensor, nur einen Teil des Tiefenbildes einnimmt, muss der zum Kalibrierobjekt gehörende Bereich im Tiefenbild identifiziert werden. Dies kann aufgrund der indirekt bestimmten Lage durch die Projektion des Kalibrierobjekts in das Tiefenbild erfolgen. Die Tiefendaten innerhalb des identifizierten Bereichs können nun mit den Referenzdaten online verglichen werden, die sich aus der bekannten Lage des Kalibrierobjekts berechnen lassen. Ein vergleichbares Vorgehen ist in [LK07] für eine PMD-Kamera in Kombination mit einer extern montierten hochwertigen CCD-Kamera beschrieben worden. Das im Rahmen dieser Dissertation vorgeschlagene Verfahren wurde unabhängig von dieser Arbeit entwickelt.

Die folgenden Abschnitte gehen detailliert auf das hier skizzierte Verfahren und seine Messgenauigkeit sowie auf die Probleme, die bei der Realisierung dieses Verfahrens aufgetreten und gelöst worden sind, ein. Zunächst erfolgt jedoch eine Beschreibung der erstellten und genutzten Kalibrierobjekte, die in ähnlicher Form bereits in den Kapiteln 2 und 3 verwendet worden sind.

5.3.1 Kalibrierobjekte

Die Kalibrierobjekte bestehen im Wesentlichen aus einem planaren Objekt, auf das ein Schachbrettmuster aufgedruckt bzw. aufgeklebt worden ist. Das Schachbrettmuster eignet sich aufgrund des hohen Kontrastes und der einfachen Struktur hervorragend, um bestimmte Merkmale auf dem Kalibrierobjekt eindeutig in den Kamerabildern wiedererkennen zu können. Nachdem das Modell des Kalibrierobjekts beschrieben wurde, werden Realisierungsdetails besprochen.

5.3.1.1 Modell des Kalibrierobjekts

Als Merkmalspunkte werden die Eckpunkte \mathbf{p}_j^{Welt} verwendet, die jeweils von vier Schachbrettkästchen umgeben sind (siehe Abb. 5.17). Diese Punkte, auch Kontrollpunkte genannt, wurden z.B. in den Abschnitten 2.4.1 und 3.1.2.1 genutzt. Es wird zunächst angenommen, dass der Koordinatenursprung des Kalibrierobjekts, der im Punkt \mathbf{p}_0^{Welt} definiert sein soll, mit dem Weltkoordinatenursprung übereinstimmt. Weiterhin soll gelten, dass das planare Kalibrierobjekt in der von der X- und Y-Achse des Weltkoordinatensystems aufgespannten Ebene liegt, und für alle 3D-Punkte in der Kalibrierobjektebene die Z-Koordinate gleich Null ist, also $\mathbf{p}_j^{Welt} = (X_j, Y_j, 0)^T$. Die Kontrollpunkte liegen auf dem Kalibrierobjekt in regelmäßigen Abständen von Δx und Δy in x - bzw. y -Richtung. Die Punkte sind der Reihe nach durchnummeriert. Korrespondierende Punkte im Kamerabild können z.B. mit der OpenCV-Bibliothek [BK08] automatisch und mit Subpixelgenauigkeit extrahiert werden (vgl. Abschnitt 3.1.2). Die Anzahl der Kontrollpunkte wird durch die Anzahl der Zeilen und Spalten der Schachbrettkästchen definiert: $n_{KP} = (n_{Zeilen} - 1)(n_{Spalten} - 1)$. Um Symmetrien des Schachbrettmusters zu verhindern und somit eine eindeutige Orientierungsbestimmung zu ermöglichen, soll für die Anzahl der Zeilen und Spalten

$$(n_{Zeilen} \% 2) + (n_{Spalten} \% 2) = 1, \quad (5.17)$$

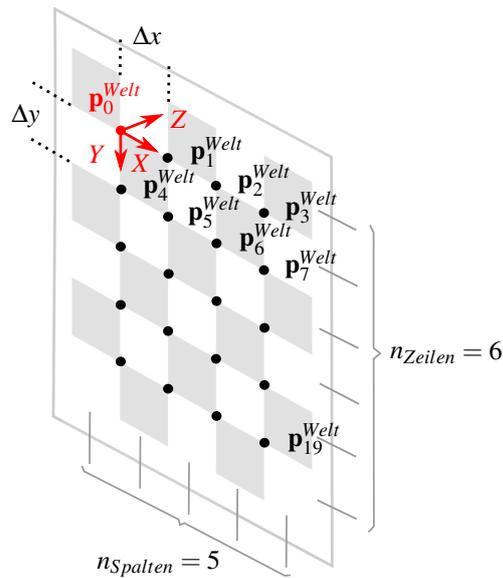


Abbildung 5.17: Modell des Kalibrierobjekts mit einem bedruckten Schachbrettmuster der Größe 6×5 . Die von jeweils vier Schachbrettkästchen umgebenen Eckpunkte definieren die Kontrollpunkte $\mathbf{p}_j^{\text{Welt}}$. Der Koordinatenursprung des Kalibrierobjekts liegt im rot markierten Punkt $\mathbf{p}_0^{\text{Welt}}$ und stimmt mit dem Weltkoordinatenursprung überein. Von da aus liegen alle anderen Punkte im regelmäßigen Gitter im Abstand Δx und Δy in der X-Y-Ebene des Weltkoordinatensystems.

gelten, wobei % der Modulo-Operation entspricht.

Bislang wurden die Kontrollpunkte $\mathbf{p}_j^{\text{Welt}} = (X_j, Y_j, 0)^T$ als 3D-Punkte mit $Z = 0$ betrachtet. Je nach Kontext (z.B. bei der Homographiebestimmung) soll $\mathbf{p}_j^{\text{Obj}} = (X_j, Y_j)^T$ den gleichen Kontrollpunkt als 2D-Punkt in der Kalibrierobjektebene bezeichnen. Das dazugehörige Koordinatensystem beschränkt sich auf die X- und Y-Achse in Abb. 5.17.

Der nachfolgende Abschnitt geht auf Details ein, die bei der Realisierung der Kalibrierobjekte zu beachten sind.

5.3.1.2 Realisierung der Kalibrierobjekte

Die verwendeten Kalibrierobjekte mit dem aufgeklebten Schachbrettmuster unterscheiden sich bei der Realisierung je nach verwendetem Einsatzbereich in den folgenden Eigenschaften:

- Anzahl der Zeilen und Spalten der Schachbrettkästchen,
- Größe des Kalibrierobjekts,
- Frequenzbereich der Sichtbarkeit des Schachbrettmusters.

Auf kurzen Distanzen zwischen 500 und 1000 mm können kleine Kalibrierobjekte mit vielen Schachbrettkästchen genutzt werden, da diese im Kamerabild noch hinreichend

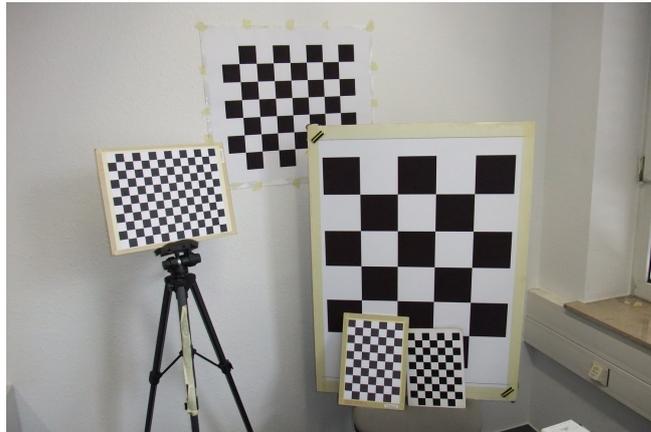


Abbildung 5.18: Gezeigt sind einige der im Rahmen dieser Arbeit entstandenen Kalibrierobjekte. Dabei stellt die Wand eine Erweiterung der planaren Fläche, in der sich das aufgeklebte Schachbrettmuster (oben) befindet, dar.

groß für die automatische Detektion sind. Für größere Distanzen sollten größere Kalibrierobjekte mit wenigen Schachbrettkästchen genutzt werden. Die Abb. 5.18 zeigt einige der erstellten und verwendeten Kalibrierobjekte. Während bei der Erstellung kleiner Kalibrierobjekte die Planarität einfach gewährleistet werden kann, ist dies bei größeren nicht trivialerweise der Fall.

Im Rahmen dieser Arbeit entstand ein kostengünstiges, mobiles Kalibrierobjekt der Größe 750×1000 mm, bei dem besondere Sorgfalt auf Planarität gelegt worden ist, da diese Einfluss auf die Genauigkeit der Lagebestimmung hat. Dabei wurde ein Brett, das leicht konkav geformt ist, unter hoher Luftfeuchtigkeit mit einem DIN A0 großen Schachbrettmuster beklebt. Durch die hohe Luftfeuchtigkeit beim Aufkleben (Raum künstlich mit Wasserdampf angereichert) konnte eine Wellung des Papiers dauerhaft verhindert werden, da die normale Luftfeuchtigkeit zu späteren Zeitpunkten deutlich geringer war und das Papier dadurch auf Spannung blieb.

An der Rückseite ist ein stabiler Holzstab mit einem Abstandshalter zum Brett hin montiert worden, der das leicht konkave Brett zurück wölbte. Je nach Höhe des Abstandshalters konnte somit der Grad der Wölbung des Bretts eingestellt werden. Zusätzlich sorgte die bestehende Spannung für eine erhöhte Stabilität des Kalibrierobjekts. Zur Überprüfung der Planarität an der Vorderseite wurde ein langes Aluminiumrohr verwendet. Dieses ist in mehreren Richtungen an das Kalibrierobjekt angelegt und der maximale Abstand zwischen Rohr und Kalibrierobjekt überprüft worden. Dieser lag unterhalb von 1 mm. Wird die Breite des Kalibrierobjekts von 750 mm betrachtet, so lag der Planaritätsfehler unterhalb von 0.15%. Der Einfluss des Planaritätsfehlers wurde z.B. für das in [Zha98] vorgestellte Kalibrierverfahren untersucht und als Störgröße identifiziert, die jedoch in der vorliegenden Größenordnung als unproblematisch eingestuft werden kann (deutlich kleiner als 3%). Das beschriebene Kalibrierobjekt ist auf einem mobilen, rotierbaren sowie höhenverstellbaren Untersatz montiert worden.

Für Experimente, die kein mobiles, jedoch ein möglichst großes Kalibrierobjekt benötigen, wurde ein Schachbrettmuster ausgedruckt und auf eine flache weiße Wand

geklebt (siehe Abb. 5.18 oben), wodurch eine planare Fläche der Wandgröße zur Verfügung stand. Dies ist insbesondere dann notwendig, wenn das Bild bei größeren Entfernungen vollständig durch die Kalibrierobjektebene abgedeckt werden soll.

5.3.1.3 Sichtbarkeit des Schachbrettmusters

Eine Besonderheit der erstellten Kalibrierobjekte ist der erwähnte Frequenzbereich, in dem das gedruckte Schachbrettmuster sichtbar ist, was insbesondere für die Sichtbarkeit im infraroten (IR) Bereich gilt. In diesem Kontext werden zwei entgegengesetzte Anforderungen an die Reflexionseigenschaft der Kalibrierobjekte gestellt.

Anforderung 1: Bei der Kalibrierung der intrinsischen Parameter der IR-Kamera der Kinect sowie der Stereo-Kalibrierung zwischen der IR-Kamera und der internen RGB-Kamera ist es notwendig, dass das gleiche Kalibrierobjekt von beiden Kameras sowohl im infraroten als auch im normal sichtbaren Bereich erkannt werden kann. Dies ist der Fall für das linke Kalibrierobjekt in den Abb. 5.19(a) und 5.19(b).

Anforderung 2: Bei den Experimenten, bei denen die Tiefeninformation der Kinect benötigt wird, ist ein Kalibrierobjekt, welches im IR-Bereich sichtbar ist, problematisch. Dies hängt mit der Projektion des strukturierten infraroten Lichts (Punktmuster) der Kinect und den unterschiedlichen Reflexionseigenschaften von unbedruckten weißen und bedruckten schwarzen Kästchen des Schachbrettmusters zusammen. Ist das Schachbrettmuster im IR-Bereich sichtbar, so kann das infrarote Punktmuster derart reflektiert werden, dass es von der IR-Kamera falsch oder sogar gar nicht detektiert werden kann, und folglich keine korrekte Tiefenberechnung möglich ist (siehe Abb. 5.19(c)). Somit ist es im Gegensatz zum Anwendungsfall der Kalibrierung notwendig, dass das Schachbrettmuster bei den Experimenten im infraroten Bereich möglichst unsichtbar ist, wie es für das rechte Kalibrierobjekt in der Abb. 5.19(b) der Fall ist.

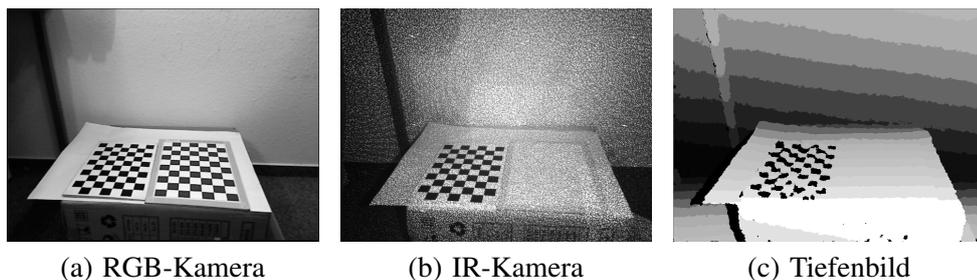


Abbildung 5.19: Zwei unterschiedlich gefertigte Kalibrierobjekte (links: Laserdrucker, rechts: Tintenstrahldrucker) werden von der Kinect aufgezeichnet. Das Bild der internen RGB-Kamera (grauwertkonvertiert) ist in (a), das Bild der IR-Kamera in (b) und das resultierende Tiefenbild in (c) dargestellt. Das linke Kalibrierobjekt ist sowohl im sichtbaren als auch im infraroten Bereich deutlich zu erkennen. Dies führt zu Fehlern bei der Berechnung des Tiefenbildes. Das rechte Kalibrierobjekt hingegen ist im IR-Bereich nahezu unsichtbar, wodurch Störungen und somit Fehler im Tiefenbild ausbleiben.

Um diesen entgegengesetzten Anforderungen gerecht zu werden, wurden die Schachbrettmuster mit unterschiedlichen Drucktechniken gefertigt. Es konnte festgestellt werden, dass das Druckbild eines Laserdruckers im infraroten Bereich vollständig sichtbar und somit für die Kalibrierung der Kameras geeignet ist. Das Druckbild eines Tintenstrahldruckers hingegen ist im infraroten Bereich größtenteils unsichtbar und somit für die störungsfreie Erfassung der Tiefendaten bestens geeignet. Tabelle 5.2 fasst die Ergebnisse zusammen.

Anwendungskontext	Sichtbarkeitsanforderung	Drucktechnik
Kalibrierung	IR-Sichtbarkeit	Laser
Tiefenmessung	IR-Unsichtbarkeit	Tintenstrahl

Tabelle 5.2: Zusammenfassung der Sichtbarkeitsanforderung in Abhängigkeit vom Anwendungskontext und der Drucktechnik.

5.3.2 Stereo-Kalibrierung der Kinect

Um in den nachfolgenden Abschnitten kalibrierte Kameras voraussetzen zu können, wurde die Stereo-Kalibrierung, wie in Abschnitt 4.3.2 beschrieben, für die Kinect durchgeführt. Dabei wurden die interne IR- und RGB-Kamera sowie zusätzlich eine extern montierte USB-Webcam (1280×1024 Pixel, 6 mm Objektiv) kalibriert, siehe Abb. 5.20. Es sei angemerkt, dass für das in den nächsten Abschnitten beschriebene Verfahren, wie zu Beginn des Abschnitts 5.3 erwähnt, die externe Kamera nicht notwendig ist. Sie stellt lediglich eine Alternative zur kinectinternen RGB-Kamera dar. Die Stereo-Kalibrierung ist sowohl zwischen der internen RGB- und IR-Kamera als auch zwischen der externen USB-Webcam und der IR-Kamera durchgeführt worden. Das verwendete Kalibrierobjekt wurde entsprechend der Überlegungen aus Abschnitt 5.3.1.3 gewählt und ist sowohl im normalen als auch im infraroten Bereich sichtbar. Da während der Stereo-Kalibrierung die Tiefenmessung nicht relevant ist, wurde der Projektor abgedeckt, um zu verhindern, dass das infrarote Punktmuster ("speckle pattern") die Detektion des Kalibrierobjekts im IR-Bild stört. Die ermittelten Parameter aller drei Kameras sowie die Stereo-Konfiguration sind der Tabelle 5.3 zu entnehmen. Statt der neun Koeffizienten der Rotationsmatrix wurde die Rotation zwischen den Kamerasystemen mit Hilfe der drei Roll-Pitch-Yaw Rotationswinkel dargestellt [Die06].

5.3.3 Lagebestimmung des Kalibrierobjekts mittels der RGB-Kamera

Die in Abschnitt 5.3.1 beschriebenen Kalibrierobjekte sollen genutzt werden, um Online-Referenzmessdaten zu gewinnen, mit denen die Tiefendaten der Kinect verglichen werden können. Aufgrund der bekannten Geometrie der Kalibrierobjekte ist es möglich, ihre Lage mit Hilfe einer einzigen Kamera zu bestimmen. Da die Kinect bereits über eine interne RGB-Kamera verfügt, die bislang nicht anderweitig verwen-

Parameter	α	β	u_0	v_0	κ_1	κ_2	κ_3	ρ_1	ρ_2
RGB	1046.05	1047.96	655.30	519.87	0.1826	-0.5064	0.4289	0.0007	0.0016
IR	587.01	587.72	321.16	237.45	-0.0994	0.4529	-0.6237	-0.0012	-0.0001
Ext. RGB	1311.74	1310.83	644.24	480.27	-0.4186	0.2594	-0.1157	0.0002	0.0003

Stereo-Konfiguration	t_x	t_y	t_z	θ_{roll}	θ_{pitch}	θ_{yaw}
RGB \rightarrow IR	25.05726	-0.32295	-0.55536	-0.00107	-0.00074	0.00158
Ext. RGB \rightarrow IR	-42.62241	0.61663	32.43970	0.00103	0.02204	0.02513

Tabelle 5.3: Ergebnisse der Kalibrierung der intrinsischen und der Verzerrungsparameter (obere Tabelle) sowie der Stereo-Konfiguration, bestehend aus der Translation und der Rotation (gegeben als Roll-Pitch-Yaw Rotationswinkel) zwischen den jeweiligen Kamerasystemen (untere Tabelle).

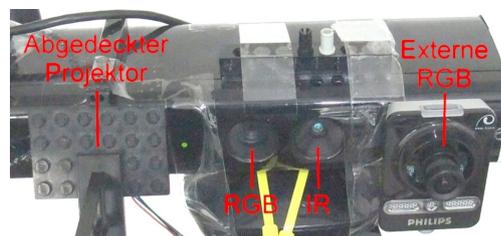


Abbildung 5.20: Die Kinect mit dem abgedeckten Projektor, der internen IR- und RGB-Kamera sowie der extern montierten RGB-Kamera (USB-Webcam) mit einem 6 mm Objektiv. Beide RGB-Kameras unterstützen eine Auflösung von 1280×1024 Pixeln.

det wird, soll diese im Folgenden für die Lagebestimmung eingesetzt werden. Zudem liefert sie bereits eine relativ hohe Auflösung (1280×1024 Pixel), was insbesondere bei weiter distanzierten Kalibrierobjekten von Vorteil ist. Alternativ kann aber auch eine extern montierte RGB-Kamera, wie z.B. in Abb. 5.20 dargestellt, eingesetzt werden. Es wird davon ausgegangen, dass die RGB-Kamera bereits kalibriert ist und die intrinsische Matrix \mathbf{K}_{RGB} sowie der Verzerrungsparametervektor δ_{RGB} bekannt sind (siehe Abschnitt 2.4). Weiterhin sei angenommen, dass die Kamera die 3D-Kontrollpunkte \mathbf{p}_j^{Welt} auf dem Kalibrierobjekt auf die verzerrten Bildpunkte $\hat{\mathbf{p}}_j$ projiziert. Die Rotationsmatrix $\mathbf{R}_{K \rightarrow RGB}$ und der Translationsvektor $\mathbf{t}_{K \rightarrow RGB}$, die die Transformation zwischen dem Koordinatensystem des Kalibrierobjekts (gleichzeitig Weltkoordinatensystem) und dem der RGB-Kamera beschreiben, können durch die folgenden Schritte bestimmt werden:

1. Erfassung des Kamerabildes \mathbf{B} , das das gesuchte Kalibrierobjekt enthält.
2. Extraktion der verzerrten Kontrollpunkte $\hat{\mathbf{p}}_j$ aus dem Bild \mathbf{B} mit der in Abschnitt 3.1.2.2 eingeführten Funktion $\mathcal{F}_{ExtKP}(\mathbf{B})$.
3. Berechnung der zu den Punkten $\hat{\mathbf{p}}_j$ gehörenden entzerrten Punkte \mathbf{p}_j unter Verwendung von δ_{RGB} , wie in Abschnitt 2.5.2 beschrieben.
4. Erzeugung der entsprechenden 2D-Kontrollpunkte $\mathbf{p}_j^{Obj} = (X_j, Y_j)^T$ in der Kalibrierobjektebene für jeden 3D-Kontrollpunkt $\mathbf{p}_j^{Welt} = (X_j, Y_j, 0)^T$ (vgl. Abschnitt 5.3.1.1).

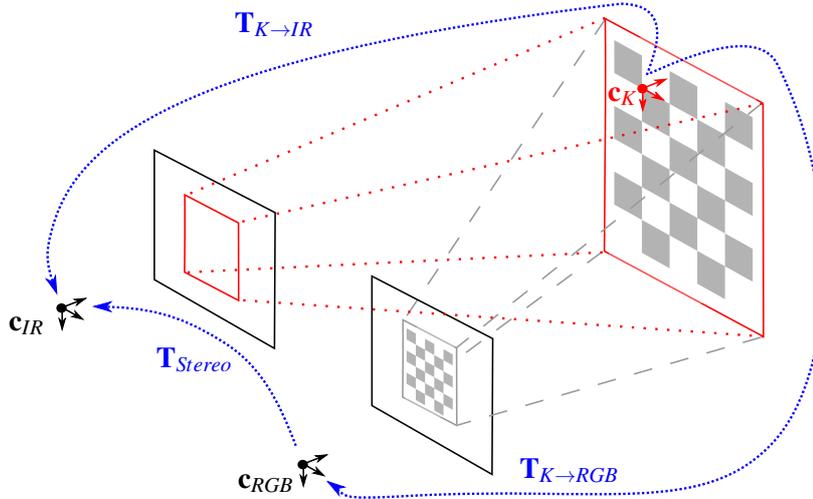


Abbildung 5.21: Das Kalibrierobjekt wird auf die Bildebene der kalibrierten RGB-Kamera abgebildet und die Transformation $\mathbf{T}_{K \rightarrow RGB}$ kann dann mit Hilfe der Homographie berechnet werden. $\mathbf{T}_{K \rightarrow IR}$ lässt sich aus der Kombination von $\mathbf{T}_{K \rightarrow RGB}$ und der Stereo-Transformation \mathbf{T}_{Stereo} zwischen der RGB- und der IR-Kamera ermitteln. Die Lage des Kalibrierobjekts zur IR-Kamera ist somit indirekt bekannt. Mit der ebenfalls kalibrierten IR-Kamera und $\mathbf{T}_{K \rightarrow IR}$ ist es schließlich möglich, mit Hilfe des Kameramodells den gesuchten Bereich im IR-Kamerabild zu bestimmen, auf den das Kalibrierobjekt projiziert werden würde (rot dargestellt).

5. Berechnung der Homographie \mathbf{H} , basierend auf den 2D-Kontrollpunkten \mathbf{p}_j^{Obj} in der Kalibrierobjektebene und den entzerrten Punkten \mathbf{p}_j in der Bildebene.
6. Berechnung von $\mathbf{R}_{K \rightarrow RGB}$ und $\mathbf{t}_{K \rightarrow RGB}$ unter Verwendung der Homographie \mathbf{H} und der intrinsischen Matrix \mathbf{K}_{RGB} , wie in Abschnitt 2.4.2 beschrieben.
7. Verfeinerung von $\mathbf{R}_{K \rightarrow RGB}$ und $\mathbf{t}_{K \rightarrow RGB}$ durch Minimierung des Reprojektionsfehlers

$$d = \sum_{j=0}^{n_{KP}-1} \left\| \hat{\mathbf{p}}_j - \mathcal{P}(\mathbf{p}_j^{Welt}, \mathbf{R}_{K \rightarrow RGB}, \mathbf{t}_{K \rightarrow RGB}, \mathbf{K}_{RGB}, \delta_{RGB}) \right\|^2 \quad (5.18)$$

mit der Anzahl der Kontrollpunkte n_{KP} und der Projektionsfunktion \mathcal{P} aus Gleichung (2.20), die die Projektion der 3D-Kontrollpunkte des Kalibrierobjekts \mathbf{p}_j^{Welt} entsprechend des Kamera- und Verzerrungsmodells vorhersagt.

Die zu $\mathbf{R}_{K \rightarrow RGB}$ und $\mathbf{t}_{K \rightarrow RGB}$ entsprechende Transformation $\mathbf{T}_{K \rightarrow RGB} \in \mathbb{R}^{4 \times 4}$ in homogener Darstellung sei schließlich gegeben durch

$$\mathbf{T}_{K \rightarrow RGB} = \begin{pmatrix} \mathbf{R}_{K \rightarrow RGB} & \mathbf{t}_{K \rightarrow RGB} \\ \mathbf{0} & 1 \end{pmatrix} \quad (5.19)$$

mit $\mathbf{0} = (0 \ 0 \ 0)$. Diese Transformation ist unter anderem in Abb. 5.21 verdeutlicht.

5.3.4 Schätzung der Kalibrierobjektlage bezüglich der IR-Kamera

Da die IR-Kamera der Kinect das infrarote Punktmuster ("speckle pattern") beobachtet und dieses im IR-Bild extrem dominant ist, ist es nicht möglich, die Lage des Kalibrierobjekts mit Hilfe des IR-Bildes direkt zu bestimmen. Hinzu kommt, dass das verwendete Schachbrettmuster im infraroten Bereich nahezu unsichtbar bleibt, um das infrarote Punktmuster und somit den Tiefensensor nicht zu stören (vgl. Abschnitt 5.3.1.3). Da jedoch die Lage des Kalibrierobjekts bzgl. der RGB-Kamera aus Abschnitt 5.3.3 und die Lage der beiden Kameras zueinander aus der Stereo-Kalibrierung aus Abschnitt 4.3.2 bekannt sind, kann die Kalibrierobjektlage bzgl. der IR-Kamera indirekt geschätzt werden durch

$$\mathbf{T}_{K \rightarrow IR} = \begin{pmatrix} \mathbf{R}_{K \rightarrow IR} & \mathbf{t}_{K \rightarrow IR} \\ \mathbf{0} & 1 \end{pmatrix} = \mathbf{T}_{Stereo} \mathbf{T}_{K \rightarrow RGB}, \quad (5.20)$$

wie in Abb. 5.21 veranschaulicht. Somit kann die Kalibrierobjektebene im Koordinatensystem der IR-Kamera wie folgt in Parameterdarstellung beschrieben werden:

$$\mathbf{a} = \mathbf{t}_{K \rightarrow IR} + \lambda \underbrace{\begin{bmatrix} \mathbf{R}_{K \rightarrow IR} \begin{pmatrix} \Delta x \\ 0 \\ 0 \end{pmatrix} \\ \mathbf{v}_1 \end{bmatrix}}_{\mathbf{v}_1} + \mu \underbrace{\begin{bmatrix} \mathbf{R}_{K \rightarrow IR} \begin{pmatrix} 0 \\ \Delta y \\ 0 \end{pmatrix} \\ \mathbf{v}_2 \end{bmatrix}}_{\mathbf{v}_2}. \quad (5.21)$$

Dabei entsprechen Δx und Δy den Modellparametern des Kalibrierobjekts (Ausmaß der Schachbrettkästchen, vgl. Abschnitt 5.3.1.1) und \mathbf{v}_1 und \mathbf{v}_2 den Richtungsvektoren der Ebene. Die entsprechende Kalibrierobjektebene in Normalform ist gegeben durch

$$E_K : (\mathbf{a} - \mathbf{t}_{K \rightarrow IR}) * \mathbf{n}_{IR}^K = 0 \quad (5.22)$$

mit dem Normalenvektor $\mathbf{n}_{IR}^K = \mathbf{v}_1 \times \mathbf{v}_2$.

5.3.5 Berechnung der Referenztiefenmessung

Der Kinect Tiefensensor liefert Tiefenmessungen in Form eines Tiefenbildes mit einer Auflösung von 640×480 Pixeln. Wie zu Beginn des Abschnitts 5.3 beschrieben, soll die optische Referenzmessung mit Hilfe der RGB-Kamera für frei wählbare Positionierungen der Kinect bzw. des Kalibrierobjekts möglich sein. Je nach Entfernung zur Kinect ist das Kalibrierobjekt nur in einem Ausschnitt des Tiefenbildes zu sehen. Es ist notwendig, diesen Bereich zu identifizieren, da die Referenzmessung sich nur auf diesen beschränken soll. Die Abb. 5.22 zeigt ein Bildpaar, bestehend aus der RGB-Kameraaufnahme und dem dazugehörigen Tiefenbild der Kinect. Der relevante Bereich ist in beiden Bildern durch das rote Polygon angedeutet.

Im RGB-Bild kann der relevante Bereich z.B. direkt mit Hilfe der aus dem Bild extrahierten Kontrollpunkte bestimmt werden. Dies ist jedoch im Tiefenbild nicht direkt möglich. Um den Bereich indirekt zu bestimmen, wird als Zwischenschritt die in Abschnitt 5.3.4 indirekt berechnete Lage des Kalibrierobjekts bzgl. der kalibrierten IR-Kamera genutzt. Dabei werden die vier äußersten Kontrollpunkte des Kalibrierobjekts



Abbildung 5.22: Aufnahmen einer Szene mit dem Kinect Sensor. Das RGB-Kamerabild (nach Graustufenkonvertierung) und das zugehörige Tiefenbild sind in (a) und (b) dargestellt. Der relevante Bereich in beiden Bildern ist durch das rote Polygon verdeutlicht. Für Tiefenwerte, die innerhalb des Polygons liegen, können Referenzmessungen online berechnet werden.

in das Bild der IR-Kamera mit Hilfe der Projektionsfunktion \mathcal{P} aus Gleichung (2.20) projiziert. Diese vier projizierten Punkte definieren das Polygon des relevanten Bereichs im IR-Bild. Die berechneten Koordinaten der projizierten Punkte müssen anschließend um jeweils 3 Pixel in u - und v -Richtung verschoben werden. Dies entspricht dem Offset zwischen dem IR-Bild und dem Tiefenbild, von dem in [SJP11] berichtet wurde und der im Rahmen dieser Dissertation bestätigt werden konnte. Der Offset steht im Zusammenhang mit der Ausdehnung des verwendeten Fensters bei der korrelationsbasierten Berechnung der Disparität. Das durch die verschobenen Punkte definierte Polygon definiert schließlich den gesuchten relevanten Bereich im Tiefenbild (vgl. Abb. 5.22(b)). Durch die Einführung eines Steuerparameters ist es zudem möglich, die Größe des relevanten Bereichs zu skalieren. Dies ist beim Einsatz großer Kalibrierobjekte relevant, deren planare Oberfläche deutlich über das Schachbrettmuster hinausragt (vgl. Schachbrettmuster auf der Wand in Abb. 5.18(b)).

5.3.5.1 Pixelbasierte Referenztiefmessung

Bei Verwendung des OpenNI-Frameworks [Ope10] liefert jedes Pixel des Tiefenbildes die Tiefe bzw. Z -Koordinate eines Objektpunktes in der Welt, der auf das jeweilige Pixel projiziert werden würde. Ziel dieses Abschnitts ist es, für jedes Pixel, das im relevanten Bereich des Tiefenbildes liegt, einen Referenztiefmesswert zu ermitteln und somit eine pixelweise Abweichung bestimmen zu können. Sei $\hat{\mathbf{q}} = (\hat{q}_u, \hat{q}_v)^T$ eine Pixelposition im Tiefenbild. Mit Hilfe des Punkt-in-Polygontests [Hec94] oder des Scanline-Verfahrens [Fel92] kann geprüft werden, ob sich das Pixel innerhalb des durch das Polygon definierten relevanten Bereichs befindet. Ist dies der Fall, wird der Referenzmesswert für das Tiefenbildpixel an der Position $\hat{\mathbf{q}}$ durch die folgenden Schritte berechnet, wobei davon ausgegangen wird, dass das Weltkoordinatensystem in der IR-Kamera liegt, die IR-Kamera bereits kalibriert ist und somit die intrinsische Ma-

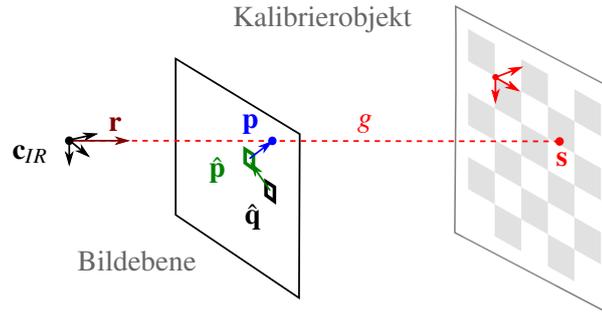


Abbildung 5.23: Berechnung des Referenztieftwertes für das Pixel an der Position $\hat{\mathbf{q}}$ im Tiefenbild. Die entsprechende Position im IR-Bild ergibt sich durch den Offset $(-3, -3)^T$ und ist durch $\hat{\mathbf{p}}$ dargestellt (grün). Die entsprechende entzerrte Bildposition ist durch \mathbf{p} angedeutet (blau). Die Gerade g verläuft durch das Projektionszentrum \mathbf{c}_{IR} in Richtung des Vektors \mathbf{r} und schneidet die Bildebene der IR-Kamera im Punkt \mathbf{p} . Der Referenztieftwert für das Pixel an der Position $\hat{\mathbf{q}}$ im Tiefenbild ergibt sich aus der Z -Komponente des Schnittpunktes \mathbf{s} zwischen der Geraden g und dem Kalibrierobjekt.

trix \mathbf{K}_{IR} sowie der Verzerrungsparametervektor δ_{IR} bekannt sind (siehe Abschnitt 2.4):

1. Bestimmung der zu der Position $\hat{\mathbf{q}} = (\hat{q}_u, \hat{q}_v)^T$ im Tiefenbild entsprechenden Position $\hat{\mathbf{p}}$ im Pixelkoordinatensystem der IR-Kamera unter Berücksichtigung des Offsets zwischen dem IR- und dem Tiefenbild mit $\hat{\mathbf{p}} = (\hat{q}_u - 3, \hat{q}_v - 3)^T$.
2. Berechnung der zu $\hat{\mathbf{p}}$ gehörenden entzerrten Pixelposition \mathbf{p} unter Verwendung von δ_{IR} , wie in Abschnitt 2.5.2 beschrieben.
3. Berechnung der Geraden g im 3D-Raum (Strahl), die durch den Ursprung der IR-Kamera \mathbf{c}_{IR} verläuft und die Bildebene im Punkt \mathbf{p} schneidet.
4. Berechnung des Schnittpunktes $\mathbf{s} = (s_x, s_y, s_z)^T$ der Geraden g mit der Kalibrierobjektebene, deren Lage bzgl. der IR-Kamera aus Abschnitt 5.3.4 bekannt ist.
5. Bestimmung des Referenzmesswertes d_{ref} für die an der Position $\hat{\mathbf{q}}$ gespeicherte Tiefe im Tiefenbild mit $d_{ref} = s_z$.

Abb. 5.23 verdeutlicht das Vorgehen. Die in Schritt 3 beschriebene Gerade sei in der Parameterdarstellung durch $g : \mathbf{a} = \mathbf{c}_{IR} + \lambda \mathbf{r}$ gegeben, wobei \mathbf{r} der Richtungsvektor ist und das Zentrum \mathbf{c}_{IR} der IR-Kamera (im vorliegenden Fall als $\mathbf{c}_{IR} = (0, 0, 0)^T$ angenommen) dem Stützvektor entspricht. Sei $\tilde{\mathbf{p}} = (p_u, p_v, 1)$ die homogene Darstellung des in Schritt 3 berechneten Punktes \mathbf{p} in Pixelkoordinaten. Dann lässt sich der 3D-Richtungsvektor \mathbf{r} bestimmen durch

$$\mathbf{r} = \begin{pmatrix} p'_x/p'_z \\ p'_y/p'_z \\ 1 \end{pmatrix} \quad \text{mit} \quad \mathbf{p}' = \begin{pmatrix} p'_x \\ p'_y \\ p'_z \end{pmatrix} = \mathbf{K}_{IR}^{-1} \tilde{\mathbf{p}}, \quad (5.23)$$

wobei \mathbf{K}_{IR}^{-1} die inverse intrinsische Matrix der IR-Kamera bezeichnet (siehe Gleichung (2.61)). Um den Schnittpunkt \mathbf{s} in Schritt 4 zu berechnen, muss der entsprechende Parameter λ bestimmt werden. Dazu wird die Geradengleichung von g in die

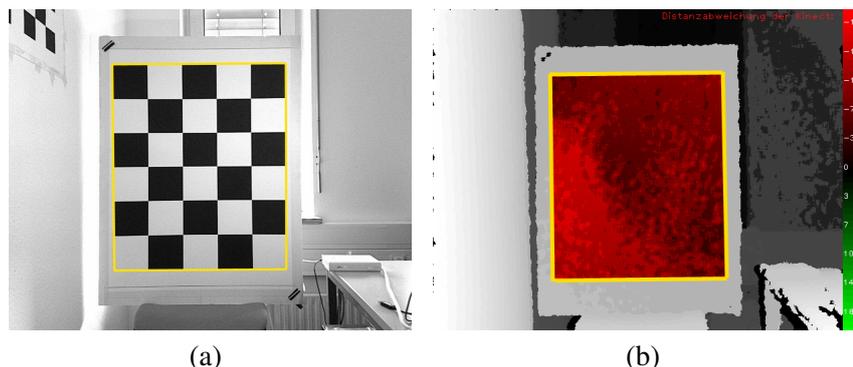


Abbildung 5.24: Das RGB-Bild einer Szene mit dem Kalibrierobjekt ist in (a) und das dazugehörige Tiefenbild der Kinect in (b) dargestellt. Das Tiefenbild ist überlagert mit den farblich dargestellten pixelweisen Abweichungen in Millimetern (siehe Farbskala) zwischen dem Tiefenbild und der Referenzmessung d_{ref} . Nur Pixel innerhalb des relevanten Bereichs, der vollständig zum planaren Kalibrierobjekt gehören muss (durch gelbes Polygon dargestellt), werden ausgewertet.

Ebenengleichung $(\mathbf{a} - \mathbf{t}_{K \rightarrow IR}) * \mathbf{n}_{IR}^K = 0$ des Kalibrierobjekts (vgl. Gleichung 5.22) eingesetzt:

$$\begin{aligned} (\mathbf{c}_{IR} + \lambda \mathbf{r} - \mathbf{t}_{K \rightarrow IR}) * \mathbf{n}_{IR}^K &= 0 \\ \Leftrightarrow \frac{(\mathbf{t}_{K \rightarrow IR} - \mathbf{c}_{IR}) * \mathbf{n}_{IR}^K}{\mathbf{r} * \mathbf{n}_{IR}^K} &= \lambda. \end{aligned} \quad (5.24)$$

Unter Verwendung des so bestimmten λ ist der Schnittpunkt schließlich gegeben durch $\mathbf{s} = \mathbf{c}_{IR} + \lambda \mathbf{r}$ und der Referenztiefenwert d_{ref} steht nach Schritt 5 zur Verfügung. In Abb. 5.24(b) ist die pixelweise Abweichung des durch die Kinect ermittelten Tiefenwertes vom Referenztiefenwert d_{ref} innerhalb des relevanten Bereichs (gelbes Polygon) farbig dargestellt.

Die Laufzeit des beschriebenen Verfahrens ist auf einem PC (Prozessor: Core i7-2600, Nutzung nur eines Kerns) untersucht worden. Für die Lagebestimmung des Kalibrierobjekts war weniger als 1 ms notwendig. Die Laufzeit der genannten Schritte 1 bis 5 für alle Pixel innerhalb des relevanten Bereichs mit der Größe, wie sie in etwa in Abb. 5.24(b) zu sehen ist, lag bei ca. 8 ms inklusive der Mehrfachauswertung des Punkt-in-Polygontests. Somit kann die Berechnung für den relevanten Bereich in weniger als 10 ms durchgeführt und eine Framerate von über 100 FPS erreicht werden. In dieser Rechnung nicht mitberücksichtigt sind jedoch die Detektion des Kalibrierobjekts und die Extraktion der Kreuzungspunkte aus dem 1280×1024 Pixel großen RGB-Kamerabild, welche den wesentlichen Berechnungsaufwand verursachen. Bei der verwendeten Implementierung wird für diese je nach Bildinhalt und Rauschpegel zwischen 24 und 100 ms, in der Regel etwa 60 ms benötigt. Mit dem verwendeten PC konnten folglich insgesamt Frameraten zwischen 9 und 29 FPS erreicht werden. Somit können die Referenzmessdaten im Echtzeitbetrieb in der Regel für jeden zweiten Frame der Kinect berechnet werden, so dass die Genauigkeit der online übertragenen

Tiefenmessung für eine beliebige Konfiguration von Sensor und Kalibrierobjekt in beliebigen Bildteilen des Tiefenbildes verfügbar und visualisierbar ist. Dies ermöglicht dem menschlichen Betrachter während des laufenden Betriebs flexibel Information über die Qualität der Tiefenmessung der Kinect zu erhalten und Fehler abzuschätzen.

5.3.5.2 Flächenbasierte Referenztiefenmessung

Bei den späteren Analysen des Einflusses der Temperatur auf die Tiefenmessung der Kinect (siehe Abschnitt 5.4) wird nicht die Distanz einzelner Pixel, sondern die der gesamten Fläche des Tiefenbildes innerhalb des relevanten Bereichs betrachtet. Dieses Vorgehen weist bzgl. des Rauschens der Kinect-Tiefenmesswerte eine größere Stabilität gegenüber der Betrachtung einzelner Pixel auf und reduziert zudem das zu verarbeitende Datenaufkommen. Dies ist notwendig, weil bei den Tests längere Zeitreihen ausgewertet werden. Somit soll auch eine Referenztiefenmessung bestimmt werden, die die gleiche Fläche des relevanten Bereichs berücksichtigt. Der Algorithmus 5 fasst die Berechnung zusammen. Dabei bezeichnet $\mathbf{T}(u, v)$ den Tiefenwert aus dem Tiefenbild und $d_{ref}(u, v)$ den ermittelten Referenztiefenmesswert entsprechend Abschnitt 5.3.5.1 an der Pixelposition (u, v) . Der relevante Bereich ist durch das Polygon \mathcal{P} definiert. Die Ausgaben des Algorithmus sind die mittleren Distanzen \mathcal{D}_T (basierend auf dem Tiefenbild) sowie \mathcal{D}_M (basierend auf der Modellebene des Kalibrierobjekts) für den gesamten relevanten Bereich.

Algorithmus 5 : Berechnung der mittleren Distanzen, basierend auf dem Kinect-Tiefenbild bzw. dem mit der Modellebene des Kalibrierobjekts berechneten Referenztiefenmesswert.

Input : Tiefenbildeinträge $\mathbf{T}(u, v)$, Referenzmesswerte $d_{ref}(u, v)$, Tiefenbildhöhe h , Tiefenbildbreite w , Polygon \mathcal{P}

Output : Mittlere Distanzen $\mathcal{D}_T, \mathcal{D}_M$

$\mathcal{D}_T := 0$

$\mathcal{D}_M := 0$

$n := 0$

for $v := 0$ **to** $h - 1$ **do**

for $u := 0$ **to** $w - 1$ **do**

if $(u, v)^T \in \mathcal{P}$ **then**

$n := n + 1$

$\mathcal{D}_T := \mathcal{D}_T + \mathbf{T}(u, v)$

$\mathcal{D}_M := \mathcal{D}_M + d_{ref}(u, v)$

$\mathcal{D}_T := \mathcal{D}_T / n$

$\mathcal{D}_M := \mathcal{D}_M / n$

5.3.6 Genauigkeitsanalyse

Zur Abschätzung der Güte der online berechneten Referenzmessdaten sind diese mit Hilfe eines Laser-Distanzmessgerätes nachgeprüft worden. Eingesetzt wurde das



Abbildung 5.25: Die online berechneten Referenzdaten sollen ausgehend von einem fixen Referenzmesspunkt (Rasterloch oberhalb der IR-Kamera, siehe roter Pfeil in (a)) mit Hilfe eines Laser-Distanzmessgerätes (b) verifiziert werden.

BOSCH DLE 70 Professional (siehe Abb. 5.25(b)), das besonders geeignet ist, um die Distanzen mit Hilfe eines Anschlagstifts ausgehend von einem definierten Punkt zu messen. Die Lage und die Distanzen zum planaren Kalibrierobjekt lassen sich, wie in Abschnitt 5.3.4 beschrieben, relativ zum Ursprung des Koordinatensystems der IR-Kamera bestimmen. Dieser befindet sich allerdings im Inneren des Gehäuses der Kinect und ist somit für den Anschlagstift nicht erreichbar, weshalb Vergleichsmessungen mit dem Laser-Distanzmessgerät nicht unmittelbar durchgeführt werden können. Um dennoch Messungen mit dem DLE 70 zu ermöglichen, wurde ein fixer externer Referenzmesspunkt festgelegt, wie in Abb. 5.25(a) dargestellt. Dadurch wird gewährleistet, dass der Anschlagstift des Laser-Distanzmessgerätes immer am gleichen Referenzpunkt angelegt und ein Abrutschen beim Messen verhindert werden kann.

5.3.6.1 Laserdistanzbasierte Bestimmung der Referenzmesspunktlage

Wie die Lage des Referenzmesspunktes mit Hilfe des Laser-Distanzmessgerätes bestimmt werden kann, wird nachfolgend erläutert. Seien die vier äußersten Kreuzungspunkte (Kontrollpunkte) des Schachbrettmusters auf dem Kalibrierobjekt mit \mathbf{p}_0 , \mathbf{p}_1 , \mathbf{p}_2 und \mathbf{p}_3 bezeichnet (siehe Abb. 5.26(a)). Wie in Abschnitt 2.4.1 beschrieben, soll auch hier das Kalibrierobjekt in der X-Y-Ebene des Weltkoordinatensystems angenommen werden. Somit sollen im Folgenden \mathbf{p}_0 , \mathbf{p}_1 , \mathbf{p}_2 und \mathbf{p}_3 3D-Raumpunkte sein, deren Z-Komponente gleich Null ist. Der Punkt $\mathbf{p}_0 = (0, 0, 0)^T$ entspricht dem Koordinatenursprung des Kalibrierobjekts und gleichzeitig dem Weltkoordinatenursprung. Da die Geometrie des Kalibrierobjekts als bekannt angenommen werden kann, sind auch die Positionen der Punkte \mathbf{p}_i , bekannt (vgl. Abschnitt 5.3.1.1).

Um die Lage des Referenzmesspunktes \mathbf{p}_{mp} relativ zum Kalibrierobjekt zu bestimmen, wird zunächst der Abstand zwischen dem Referenzmesspunkt und den vier äußersten Kreuzungspunkten des Schachbrettmusters auf dem Kalibrierobjekt gemessen,

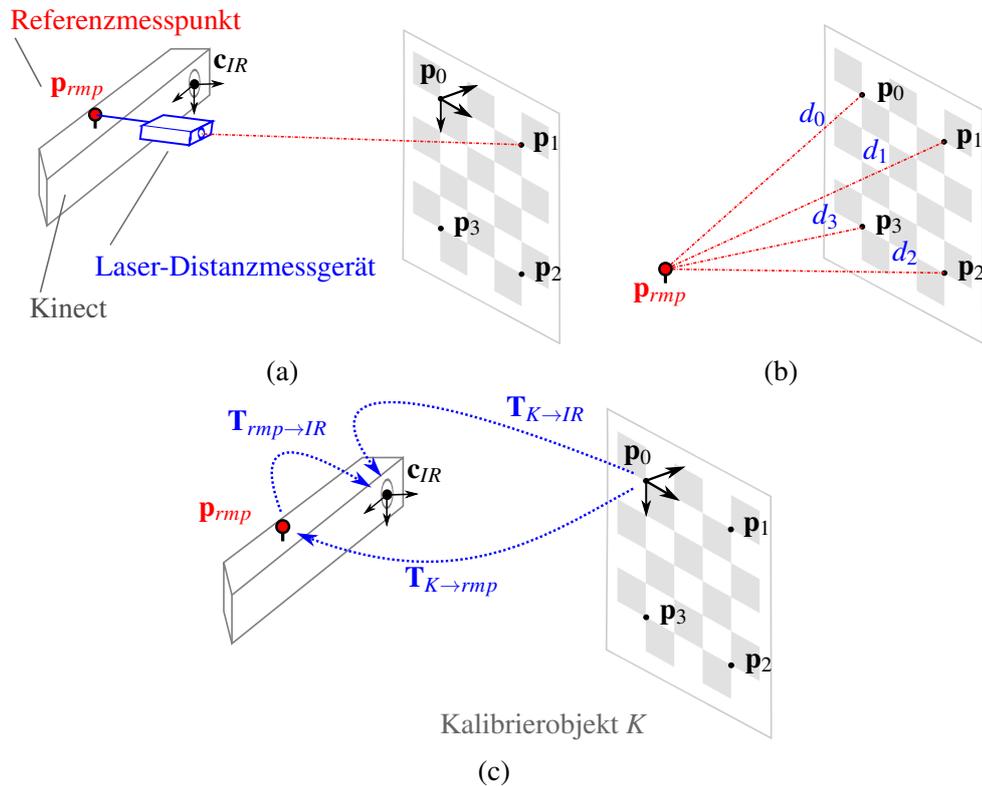


Abbildung 5.26: Das Laser-Distanzmessgerät wird mit dem Anschlagstift am Referenzmesspunkt angelegt (a), und die Distanzen d_i zu den vier äußersten Schachbrettkreuzungen des Kalibrierobjekts werden gemessen (b). (c) verdeutlicht die Transformationen zwischen den verschiedenen Koordinatensystemen.

zu denen die Raumpunkte \mathbf{p}_i gehören. Dies liefert die vier Laserdistanzen d_0, d_1, d_2 und d_3 , wie in Abb. 5.26(b) dargestellt. Da die Richtung des Lasermessstrahls nicht bekannt ist, ist es nicht direkt möglich, von den Distanzen d_i auf die Lage des Referenzmesspunktes zu schließen. Deshalb wird die Suche nach \mathbf{p}_{rmp} als Minimierungsproblem definiert. Gesucht sind somit die drei Positionskoordinaten des Punktes $\mathbf{p}_{rmp} = (p_{rmp}^x, p_{rmp}^y, p_{rmp}^z)^T$ (im Koordinatensystem des Kalibrierobjekts), dessen Abstände zu den Punkten \mathbf{p}_i , mit den gemessenen Laserdistanzen d_i möglichst übereinstimmen. Eine entsprechende Kostenfunktion kann z.B. durch

$$\mathcal{H}(\mathbf{p}_{rmp}) = \sum_{i=0}^3 (\|\mathbf{p}_{rmp} - \mathbf{p}_i\| - d_i)^2 \quad (5.25)$$

beschrieben werden. Die optimale Position von \mathbf{p}_{rmp} kann dann z.B. mit dem Levenberg-Marquardt-Algorithmus [Mor77] anhand der Kostenfunktion berechnet werden.

Die Lösung ist jedoch nicht eindeutig, da der gesuchte Punkt \mathbf{p}_{rmp} , der die Kostenfunktion in Gleichung (5.25) minimiert, sowohl vor als auch hinter dem Kalibrierobjekt liegen könnte, weshalb eine sinnvolle Initialisierung notwendig ist. Da der Referenzmesspunkt in der Nähe der IR-Kamera montiert ist, eignet sich die Position ihres Koor-

datenursprungs (aus Sicht des Koordinatensystems des Kalibrierobjekts) zur Initialisierung. Sei die Koordinatensystemtransformation zwischen dem Kalibrierobjekt und der IR-Kamera gegeben durch die Rotationsmatrix $\mathbf{R}_{K \rightarrow IR}$ und den Translationsvektor $\mathbf{t}_{K \rightarrow IR}$. Dann erhält man die gewünschte Initialisierung \mathbf{p}_{rmp}^{init} für den Levenberg-Marquardt-Algorithmus durch

$$\mathbf{p}_{rmp}^{init} = (\mathbf{R}_{K \rightarrow IR})^T (-\mathbf{t}_{K \rightarrow IR}). \quad (5.26)$$

Nach der Optimierung liegt \mathbf{p}_{rmp} im Koordinatensystem des Kalibrierobjekts vor. Durch

$$\mathbf{p}_{rmp}^{IR} = \mathbf{R}_{K \rightarrow IR} \mathbf{p}_{rmp} + \mathbf{t}_{K \rightarrow IR} \quad (5.27)$$

erhält man die Position des Referenzmesspunktes relativ zur IR-Kamera. Die Transformation zwischen Referenzmesspunkt und IR-Kamera kann nun beschrieben werden durch

$$\mathbf{R}_{rmp \rightarrow IR} = \mathbf{I}, \quad (5.28)$$

$$\mathbf{t}_{rmp \rightarrow IR} = \mathbf{p}_{rmp}^{IR}, \quad (5.29)$$

wobei \mathbf{I} die Identitätsmatrix ist und somit angenommen wird, dass zwischen den Koordinatensystemen keine Rotation vorliegt.

Seien nun $\mathbf{T}_{K \rightarrow IR}$ und $\mathbf{T}_{rmp \rightarrow IR}$ mit

$$\mathbf{T}_{K \rightarrow IR} = \begin{pmatrix} \mathbf{R}_{K \rightarrow IR} & \mathbf{t}_{K \rightarrow IR} \\ \mathbf{0} & 1 \end{pmatrix} \quad \text{und} \quad \mathbf{T}_{rmp \rightarrow IR} = \begin{pmatrix} \mathbf{R}_{rmp \rightarrow IR} & \mathbf{t}_{rmp \rightarrow IR} \\ \mathbf{0} & 1 \end{pmatrix} \quad (5.30)$$

die entsprechenden Transformationen in homogener Darstellung (siehe 5.26(c)). Dann ist die Transformation $\mathbf{T}_{K \rightarrow rmp}$ zwischen dem Kalibrierobjekt und dem Referenzmesspunkt gegeben durch

$$\mathbf{T}_{K \rightarrow rmp} = \mathbf{T}_{rmp \rightarrow IR}^{-1} \mathbf{T}_{K \rightarrow IR}. \quad (5.31)$$

Mit Hilfe dieser Transformation können für beliebige Positionierungen des Kalibrierobjekts die Distanzen d'_i zwischen den Punkten \mathbf{p}_i auf dem Kalibrierobjekt und dem Referenzmesspunkt berechnet werden durch

$$d'_i = \sqrt{\left(\frac{q_{i,x}}{q_{i,w}}\right)^2 + \left(\frac{q_{i,y}}{q_{i,w}}\right)^2 + \left(\frac{q_{i,z}}{q_{i,w}}\right)^2} \quad (5.32)$$

mit

$$\tilde{\mathbf{q}}_i = \begin{pmatrix} q_{i,x} \\ q_{i,y} \\ q_{i,z} \\ q_{i,w} \end{pmatrix} = \mathbf{T}_{K \rightarrow rmp} \begin{pmatrix} \mathbf{p}_i \\ 1 \end{pmatrix}. \quad (5.33)$$

Die berechneten Distanzen d'_i können schließlich mit den Laserdistanzmessungen d_i , $i \in \{0, \dots, 3\}$, verglichen werden.

5.3.6.2 Evaluierung der Referenzmesspunktlage

Bevor im nächsten Abschnitt die absolute Genauigkeit der Berechnung der Kalibrierobjektlage mit dem in Abschnitt 5.3.4 beschriebenen Verfahren abgeschätzt werden kann, muss die konkrete Position des Referenzmesspunktes bekannt sein. Da die Berechnung der Lage des Referenzmesspunktes \mathbf{p}_{rmp}^{IR} von der Genauigkeit des Laser-Distanzmessgerätes¹ sowie der Transformation $\mathbf{T}_{K \rightarrow IR}$ abhängig ist, werden mehrere Messungen durchgeführt und dabei die folgenden Merkmale variiert:

- Größe und Muster des Kalibrierobjekts
- Distanz zwischen Kalibrierobjekt und Referenzmesspunkt
- Verwendete Kamera (kinectinterne RGB-Kamera, externe Webcam)

Für die konkrete Bestimmung des in Abb. 5.25(a) gezeigten Referenzmesspunktes wurden 21 Messungen durchgeführt. Der Mittelwert und die Standardabweichung der Referenzmesspunktlage sind in Tabelle 5.4 angegeben. Die z-Komponente der Position hat den wichtigsten Einfluss auf die Distanzberechnung und konnte mit einer Standardabweichung von ca. 0.5 mm bestimmt werden.

	x-Position [mm]	y-Position [mm]	z-Position [mm]
Mittelwert	0.1616	-22.1366	-5.4769
Standardabweichung	0.7857	1.4453	0.5001

Tabelle 5.4: Ermittelte 3D-Position des Referenzmesspunktes (siehe Abb. 5.25(a)) relativ zum Koordinatensystem der IR-Kamera aus insgesamt 21 Messungen.

5.3.6.3 Evaluierung der Messgenauigkeit

Um die absolute Genauigkeit der mit Hilfe von Kalibrierobjekten und der RGB-Kamera berechneten Referenztiefmesswerte zu ermitteln, wurde das große Kalibrierobjekt (ca. 750×1000 mm, vgl. Abschnitt 5.3.1) in acht Abständen zwischen 900 und 3000 mm (in 300 mm Schritten) vor der Kinect platziert. Die Entfernungen vom Referenzmesspunkt zu den vier äußersten Kreuzungspunkten des Kalibrierobjektmusters wurden entsprechend der Gleichung (5.32) berechnet und die tatsächlichen Entfernungen mit dem Laser-Distanzmessgerät ermittelt. Für die Berechnung wurde sowohl die kinectinterne RGB-Kamera als auch zum Vergleich zusätzlich die extern montierte USB-Webcam herangezogen. Die Abb. 5.27(a) und 5.27(b) zeigen den Verlauf der absoluten Abweichungen für beide Kameras bei steigender Distanz. Bei der Distanz von ca. 900 mm lag der absolute Fehler bei beiden genutzten Kameras unterhalb von 0.7 mm. Die maximale ermittelte Abweichung für die interne RGB-Kamera

¹Die Messgenauigkeit des verwendeten Laser-Distanzmessgerätes ist mit ± 1.5 mm bei einer maximalen Messdistanz von 70 m angegeben. Bei den hier vorliegenden Distanzen wird eine höhere Genauigkeit angenommen.

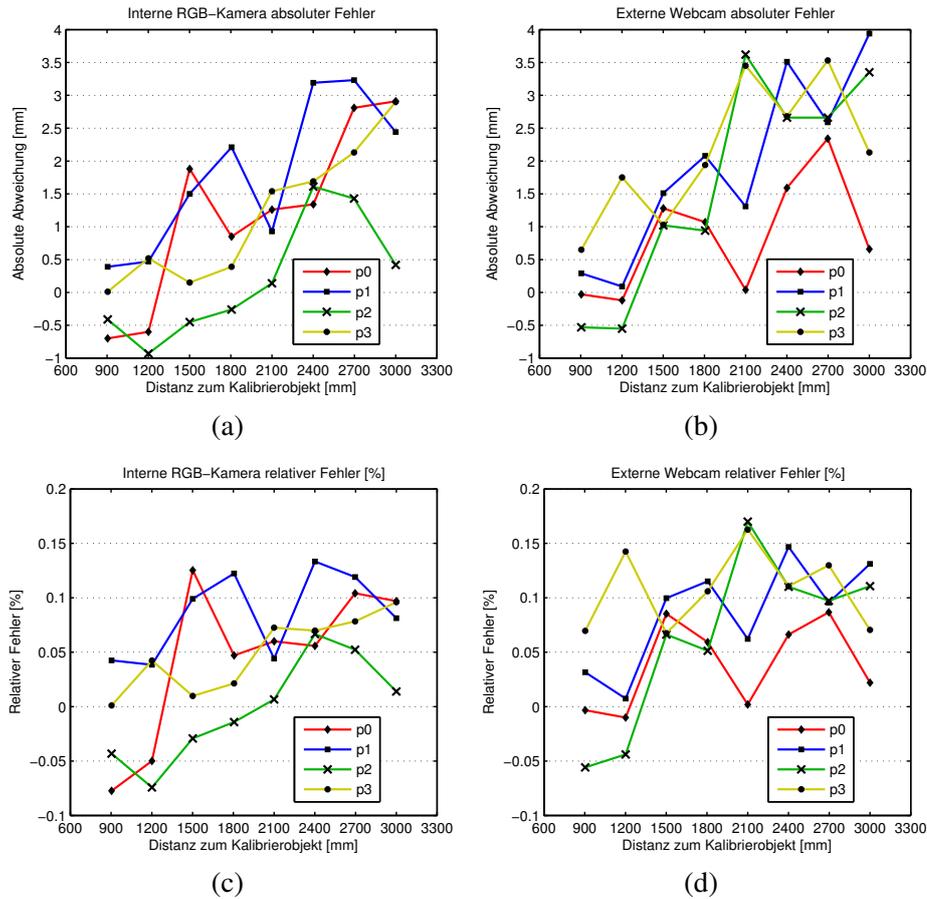


Abbildung 5.27: Die Abweichung zwischen den berechneten Distanzen der Punkte $\mathbf{p}_0, \dots, \mathbf{p}_3$ zum Referenzmesspunkt und der mit dem Laser-Distanzmessgerät ermittelten Distanzen für steigende Entfernungen sind in (a) für die interne RGB-Kamera und in (b) für die externe USB-Kamera (Webcam) dargestellt. In (c) und (d) sind für die beiden Kameras die relativen Abweichungen in Prozent angegeben.

betrug 3.23 mm bei ca. 2700 mm Entfernung und für die externe Webcam 3.94 mm bei ca. 3000 mm Entfernung.

Um den Abweichungsfehler unabhängig von der Messdistanz zu betrachten, sei der relative prozentuale Distanzfehler $d_i^{\Delta\%}$ gegeben durch

$$d_i^{\Delta\%} = \frac{100\%(d'_i - d_i)}{d_i}, \quad (5.34)$$

wobei d'_i der berechneten und d_i der mit dem Lasermessgerät ermittelten Distanz vom Referenzmesspunkt zum Kreuzungspunkt $\mathbf{p}_i, i \in \{0, \dots, 3, \}$ des Schachbrettmusters auf dem Kalibrierobjekt entsprechen. Die Auswertung des relativen Messfehlers ist für beide Kameras in den Abb. 5.27(c) und 5.27(d) zusammengefasst. Im Fall der internen RGB-Kamera wurde ein maximaler relativer Fehler von 0.133% ermittelt. Bei der externen Webcam lag dieser bei 0.170%.

5.3.6.4 Evaluierung des Messrauschens

Typische Kamerasensoren weisen ein gewisses Bildrauschen auf. Da die Bestimmung der Lage des Kalibrierobjekts auf diesen verrauschten Bilddaten basiert, hat das Sensorrauschen auch eine Auswirkung auf die Referenzmessdaten. Das Rauschen wurde wie im vorherigen Abschnitt für die Entfernungen des Kalibrierobjekts zwischen 900 und 3000 mm in Schritten von 300 mm für beide Kameras untersucht. Um das Distanzrauschen zu beurteilen, sind pro Entfernungsschritt die Distanzen zwischen dem Referenzmesspunkt und den vier Punkten p_i auf dem Kalibrierobjekt jeweils 100 mal berechnet und die Standardabweichungen bestimmt worden. Die Ergebnisse für die kinectinterne RGB-Kamera und die extern montierte Webcam sind graphisch in den Abb. 5.28(a) und 5.28(b) dargestellt. Bei beiden Kameras lag die Standardabweichung unterhalb von 0.061 mm bei der Distanz von 900 mm und unterhalb von 0.496 mm bei einer Distanz von 3000 mm. Tendenziell wurde das Rauschen mit steigender Distanz größer.

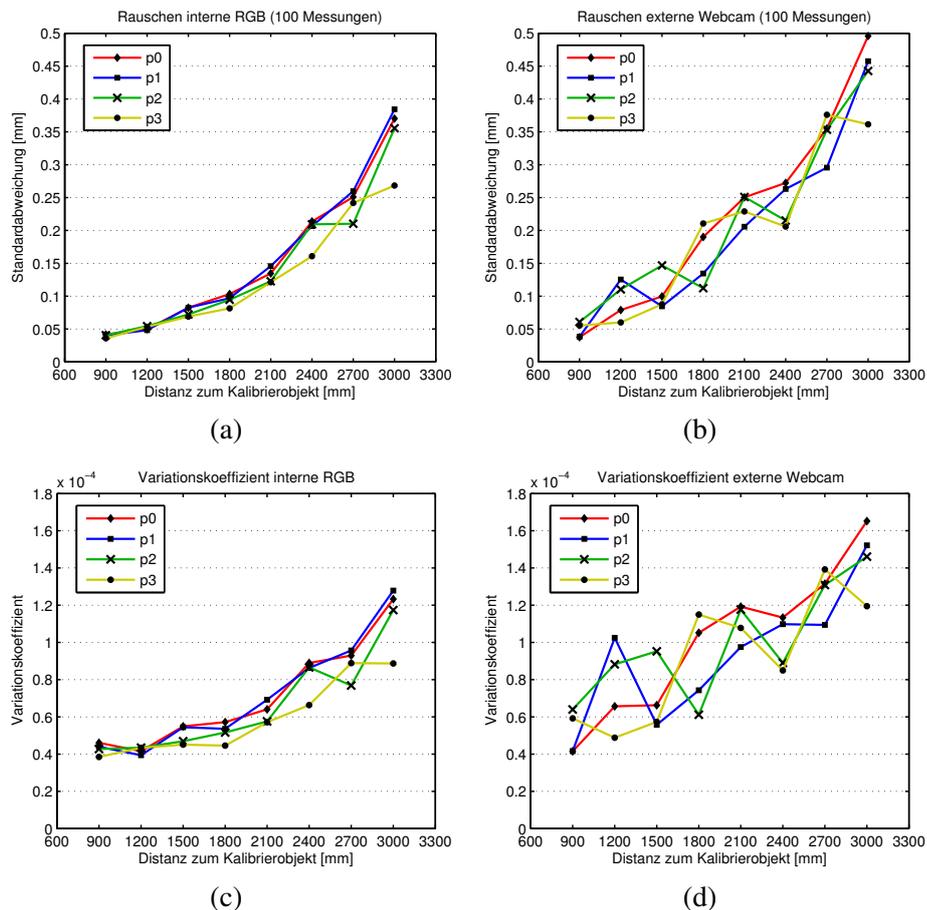


Abbildung 5.28: Rauschuntersuchung. Verlauf der Standardabweichungen für jeweils 100 Messungen pro Punkt p_i für unterschiedliche Distanzen zum Kalibrierobjekt für die kinectinterne RGB-Kamera (a) und die extern montierte Webcam (b). Das distanznormierte Rauschen wird durch den Variationskoeffizienten in (c) und (d) ausgedrückt.

Um das Rauschen unabhängig von der jeweiligen Distanz zu betrachten, wurde zusätzlich der Variationskoeffizient [Koh04] der Messreihen bestimmt, der keine Einheit besitzt und durch $v = \frac{\sigma}{\mu}$ definiert ist, wobei σ und μ die Standardabweichung und der Mittelwert der jeweiligen 100 Messwerte sind. Die Auswertung ist in den Abb. 5.28(c) und 5.28(d) für beide Kameras zusammengefasst. Der Variationskoeffizient lag bei beiden Kameras unterhalb von $6.4 \cdot 10^{-5}$ bei 900 mm und unterhalb von $1.7 \cdot 10^{-4}$ bei der Distanz von 3000 mm. Somit lag im betrachteten Distanzbereich die maximale Standardabweichung unterhalb von 0.17 mm pro 1000 mm Distanz.

5.3.6.5 Messrauschen bei flächenbasierter Referenzmessung

Für die in Abschnitt 5.3.5.2 online berechneten durchschnittlichen Distanzen innerhalb der gesamten Fläche des relevanten Bereichs ist das Rauschen untersucht worden. Die Kinect wurde frontal vor das Kalibrierobjekt in verschiedenen Entfernungen zwischen 900 und 3000 mm positioniert. Wie in den vorangegangenen Abschnitten ist das Rauschen der flächenbasierten Referenzmessung anhand der Standardabweichung der 100 Messwiederholungen pro Entfernungsschritt (300 mm) ermittelt und in Abb. 5.29(a) für beide RGB-Kameras (intern und extern) zusammengefasst worden. Bei diesem Versuchsaufbau wurde eine zusätzliche Lichtquelle verwendet, die das Kalibrierobjekt direkt beleuchtete. Damit sollte neben einer möglichst optimalen Ausleuchtung eine geringere Abhängigkeit vom Umgebungslicht erreicht werden.

Bei diesem Versuchsaufbau hat die flächenbasierte Referenzmessung unter Verwendung der externen Webcam bei zunehmender Distanz ein deutlich geringeres Rauschen gegenüber der internen RGB-Kamera aufgewiesen (ca. 45% des Rauschanteils bei 3000 mm). Während für die interne RGB-Kamera bei der flächenbasierten Referenzmessung vergleichbare Rauschwerte wie für die kreuzungspunktbasiereten Messungen im Abschnitt 5.3.6.4 ermittelt werden konnten, waren die Rauschwerte der externen Webcam bei diesem Vergleich auf etwa die Hälfte gesunken. Somit hatte der Einsatz der zusätzlichen Lichtquelle zwar keinen signifikanten Vorteil bei der Nutzung der internen RGB-Kamera, wohl aber bei Verwendung der externen Webcam. In einer gesonderten Untersuchung konnte bzgl. des Rauschens ebenfalls eine hohe Empfindlichkeit der externen Webcam gegenüber Veränderungen der Lichtverhältnisse bestätigt werden. Dies wird auf eine sensible Auto-Gain-Funktion der externen Webcam zurückgeführt, die zur Folge hat, dass das Bildrauschen bei zunehmendem Lichteinfall deutlich abnimmt.

Beim Vergleich der Kurven der Variationskoeffizienten in den Abb. 5.29(b) und 5.28(d) wird insbesondere deutlich, dass die flächenbasierten Messungen einen weniger sprunghaften Verlauf als die kreuzungspunktbasiereten Messungen aufweisen, was auf den glättenden Charakter der Mittelwertbildung beim flächenbasierten Messansatz zurückzuführen ist.

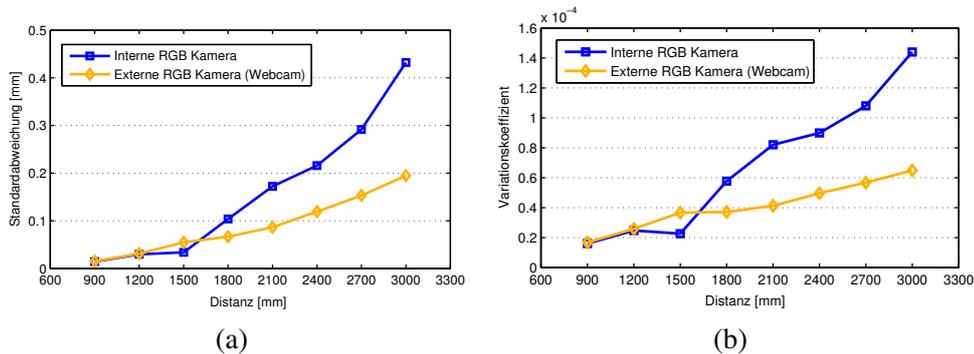


Abbildung 5.29: Vergleich der Standardabweichungen für jeweils 100 Wiederholungen der internen RGB-Kamera und der extern montierten Webcam unter Verwendung der flächenbasierten Referenzmessung. In (a) sind die absoluten Rauschwerte bei den untersuchten Distanzen abgebildet. Der dazugehörige distanznormierte Variationskoeffizient ist in (b) dargestellt.

5.4 Auswirkungen von Umwelteinflüssen auf den Kinect-Tiefensensor

In zahlreichen Arbeiten wird die Kinect als Tiefensensor verwendet [SHBS11, SS11, BRB⁺11, KE12, SJP11, WWJ11]. Diese und weitere Arbeiten haben gemeinsam, dass sie vor der Nutzung der Kinect eine Kamerakalibrierung (z.B. intrinsische Parameter, Verzerrungsparameter) sowie teilweise eine Kalibrierung der Modellparameter zur Tiefenberechnung (Tiefenkalibrierung) mit dem Ziel durchführen, die Messgenauigkeit zu erhöhen. Jedoch wird die Auswirkung von Umwelteinflüssen auf die Kinect, insbesondere der thermischen Einflüsse, bei diesen Arbeiten weder in der Kalibrier- noch während der Mess- oder Evaluierungsphase berücksichtigt. Dabei ist im allgemeinen Kontext von Kameras und Tiefensensoren die Abhängigkeit von Temperaturbedingungen in der Literatur nicht unbekannt. Im Folgenden werden einige Beispielarbeiten genannt.

Bereits in einer frühen Publikation von 1958 ist das Problem von Temperaturunterschieden und ihren Auswirkungen auf Linsensysteme und Kameraoptiken in der Luftbildfotografie diskutiert worden [Hot58]. Die untersuchten Kameras wurden in großen Gefrierschränken gelagert, um niedrige Umgebungstemperaturen in großen Flughöhen zu simulieren. Unter diesen Bedingungen wurde schließlich die Kamerakalibrierung durchgeführt, um korrekte Parameter für den späteren Einsatz während des Fluges zu erhalten. In einer anderen Arbeit ist die Robustheit unterschiedlicher Kalibrieremethoden und ihre Auswirkung auf die Kalibrierengenauigkeit bei variierenden Temperaturbedingungen getestet worden [PP08]. Weiterhin wurde z.B. in [BKS09] eine Methode zur Temperaturkalibrierung einer Infrarotkamera mit Hilfe eines Wasserbades präsentiert. Die Temperaturabhängigkeit der Tiefenmesswerte während der Aufwärmphase der Swiss Ranger SR2 Time-of-Flight Kamera wurde in [KRI06] analysiert. In [SC10] ist der Einfluss von Temperaturschwankungen auf die intrinsischen Parameter

von Spiegelreflexkameras (SLR-Kameras) überprüft worden. Jedoch wurden der Temperatureinfluss sowie weitere Umwelteinflüsse im Kontext des Kinect-Tiefensensors bislang weder untersucht noch berücksichtigt.

Die nachfolgenden Abschnitte beschreiben eine Reihe von Experimenten, in denen analysiert wird, wie und in welchem Ausmaß Distanzmessungen sowie einzelne Komponenten der Kinect durch Temperaturveränderungen, Luftdurchzug, Kühlung, Unterbrechung der Stromzufuhr usw. beeinflusst werden. Zusätzlich werden aus den Untersuchungsergebnissen praktische Regeln abgeleitet, die bei der Nutzung der Kinect zu beachten sind, um sowohl Fehler in der Kalibrierungs- als auch in der Messphase zu reduzieren. Die meisten der nachstehenden Untersuchungen und Ergebnisse sind in [FM13] veröffentlicht worden. Die Verwendung von Abbildungen aus dieser Publikation ist durch den Verlag genehmigt worden¹.

5.4.1 Thermischer Einfluss auf die internen Kameras der Kinect

Um den thermischen Einfluss auf das optische Linsensystem der beiden internen Kameras zu ermitteln, wurde die Kinect in zwei verschiedenen thermischen Zuständen untersucht. Im ersten Fall, nachfolgend als "kalter Zustand" bezeichnet, wurde die Kinect durch einen extern montierten Lüfter gekühlt (siehe Abb. 5.30(a)). Der Lüfter erzeugte einen leichten Luftstrom, der direkt durch das Gehäuse der Kinect geleitet wurde und somit die internen Komponenten der Kinect in etwa auf die Umgebungstemperatur abkühlte. Im zweiten Fall wurde der Lüfter deaktiviert und die Kinect durch die Verarbeitung des Farb- und Tiefendatenstroms (Online-Modus, aktiver Betriebsmodus) über eine Zeitspanne von 45 Minuten aufgewärmt. Der so erreichte Zustand wird nachfolgend "warmer Zustand" bezeichnet. Bei dem verwendeten Lüfter handelt es sich um einen vibrationsarmen Radiallüfter, der in der Anlaufphase langsam beschleunigt wurde, um potentielle Kräfte, die kleine Bewegungen bzw. Verschiebungen der



Abbildung 5.30: Die Kinect mit dem extern montierten Radiallüfter (links), der beliebig aktiviert werden kann (a). Im aktivierten Zustand strömt die Luft durch das Kinect-Gehäuse und kühlt dadurch die internen Komponenten ab. In (b) ist das bei den bildbasierten Experimenten verwendete texturreiche Poster abgebildet.

¹Mit freundlicher Genehmigung von Springer Science+Business Media, Original-Publikation: [FM13], Lizenznummer: 3254740445314.

Kinect zur Folge haben könnten, auszuschließen. Durch die Möglichkeit, den Lüfter nach Belieben zu aktivieren, konnte der Wärmezustand der Kinect beeinflusst werden.

5.4.1.1 Bildbasierter Vergleich

In einem statischen Versuchsaufbau wurde die Kinect vor einem texturreichen Poster der Größe 110×85 cm (siehe Abb. 5.30(b)) in einer Distanz von etwa 90 cm auf einem Kamerastativ fixiert. Bei der gegebenen Konstellation nahm das Poster die gesamte Bildfläche beider interner Kameras (RGB und IR) randlos ein. Der Projektor der Kinect wurde, wie in Abschnitt 5.3.2 beschrieben, abgedeckt, um die Aufzeichnung des Posters im IR-Kamerabild nicht zu stören. Gleichzeitig ist eine Lichtquelle mit hohem infraroten Lichtanteil verwendet worden, um genügend Licht für die IR-Kamera zur Verfügung zu stellen. Die Kinect wurde zunächst in den kalten Zustand versetzt und das Poster mit beiden Kameras aufgezeichnet. Anschließend wurde die Aufzeichnung nach Erreichen des warmen Zustands wiederholt. Die gemessene Raumtemperatur lag während des Experiments bei 27.6°C .

Zwischen den in beiden Zuständen aufgezeichneten Bildern konnten zwei Effekte festgestellt werden. Im warmen Zustand nahm zum einen die Bildunschärfe zu. Zum anderen konnte eine Art Zoom- oder Vergrößerungseffekt beobachtet werden, der sich dadurch bemerkbar machte, dass die Bildinhalte eine leichte Verschiebung von der Bildmitte zu den Bildrändern hin aufwies. Die beschriebenen Effekte werden anhand der vergrößerten Ausschnitte in Abb. 5.31 deutlich. Diese Ausschnitte wurden jeweils dem rechten Bildrand der Kamerabilder entnommen (siehe Pfeile in Abb. 5.31(a))

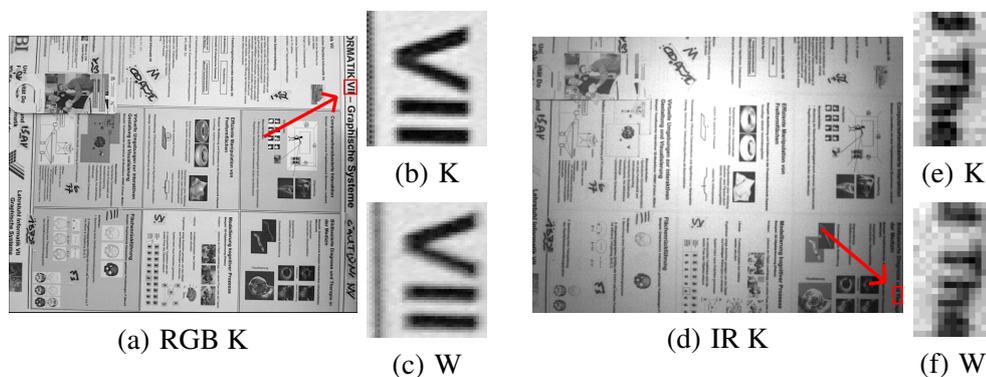


Abbildung 5.31: Die Aufzeichnung des Posters aus Abb. 5.30(b) im kalten Zustand der Kinect (mit K gekennzeichnet) ist in (a) bzw. (d) für die RGB- (1280×1024 Pixel) bzw. die IR-Kamera (640×480 Pixel) dargestellt. (b) und (e) zeigen Vergrößerungen der jeweils rot markierten kleinen Bereiche am rechten Rand des RGB- bzw. IR-Bildes. Nach einer Aufwärmphase von 45 Minuten wurde das Poster in dem unveränderten Versuchsaufbau erneut aufgezeichnet. Die Ausschnitte in (c) bzw. (f) haben exakt die gleiche Größe und liegen an der gleichen Pixelposition, stammen jedoch aus den im warmen Zustand aufgezeichneten Bildern (mit W gekennzeichnet). Beim Vergleichen lässt sich in diesen Ausschnitten eine Verschiebung zur rechten Seite sowie eine Zunahme an Unschärfe feststellen.

und 5.31(d)). Beim Vergleich dieser Ausschnitte lässt sich eine entsprechende Verschiebung nach rechts erkennen.

Trotz umfangreicher verbauter Hardware wird aufgrund des geringen Preises angenommen, dass die Verarbeitungsqualität der Kinect sowie die gewählten Materialien der Kameraoptiken bzw. der Kameragehäuse von geringer Qualität sind und die thermischen Änderungen deshalb Verschiebungen bzw. Verformungen dieser Teile zur Folge haben, wodurch auch die oben genannten Effekte hervorgerufen worden sind.

5.4.1.2 Optical-Flow basierter Vergleich

Für jedes Bildpaar des Posters (siehe Abb. 5.30(b)), bestehend aus je einer Aufnahme im kalten und im warmen Zustand, wurde der optische Fluss ("optical flow") berechnet [BWS05]. Das Ergebnis des optischen Flusses zwischen zwei Bildern ist ein Vektorfeld, das jedem Pixel einen Bewegungsvektor zuordnet. Die Bewegungsvektoren beschreiben dabei die Verschiebung eines lokalen Bildinhalts an der betrachteten Pixelposition. Zur Visualisierung der Bewegungen (des Vektorfelds) wurde die in [BSL⁺11] vorgeschlagene Farbkodierung verwendet (siehe Abb. 5.32(a)). Diese basiert auf dem HSV-Farbraum [GW01], wobei die H-Komponente ("Hue", Farbton) die Richtung und die S-Komponente ("Saturation", Sättigung) den Betrag des Bewegungsvektors wiedergibt. Die Resultate für die RGB- und IR-Kamera sind in den Abb. 5.32(b) und 5.32(c) dargestellt.

Bei Betrachtung der Farbverteilung des optischen Flusses wird deutlich, dass bei beiden Kameras im Bereich der Bildmitte keine oder nur eine sehr geringe Veränderung vorliegt (erkennbar an der weißen Farbe, minimale Sättigung). Mit wachsendem Abstand zur Bildmitte hingegen lässt sich ein zu den Bildrändern hin orientierter und stärker werdender Bewegungstrend feststellen (satter werdende Farben). Diese Beobachtung passt zu dem bereits im vorherigen Abschnitt beschriebenen Zoom-Effekt. Der maximal ermittelte optische Fluss lag bei der RGB-Kamera bei 6.7 Pixeln und bei der IR-Kamera bei 1.8 Pixeln. Detaillierte Angaben sind der Tabelle 5.5 zu entnehmen. Es sei angemerkt, dass der kleinere Wert in Relation zur kleineren Auflösung der IR-Kamera zu sehen ist und bei vergleichbarer Auflösung etwa 3.6 Pixel betragen würde.

Kamera	Bildauflösung	Maximaler Fluss	Horizontal min / max	Vertikal min / max
RGB	1280×1024	6.7338	-6.440 / 4.897	-5.044 / 3.678
IR	640×480	1.7853	-1.666 / 1.581	-1.461 / 1.115

Tabelle 5.5: Quantitative Ergebnisse des optischen Flusses für die RGB- und IR-Kamera. Alle Werte sind in Pixeln angegeben.

Zur Überprüfung der Stabilität des Verfahrens zur Berechnung des optischen Flusses gegenüber dem Bildrauschen wurde das Experiment zu einem späteren Zeitpunkt unter Verwendung eines ähnlichen Posters wiederholt. Dabei wurde das Poster jeweils zehnmal in beiden Wärmezuständen aufgezeichnet. Für jede der zehn Paarungen, bestehend aus je einer kalten und einer warmen Aufzeichnung, ist der optische Fluss

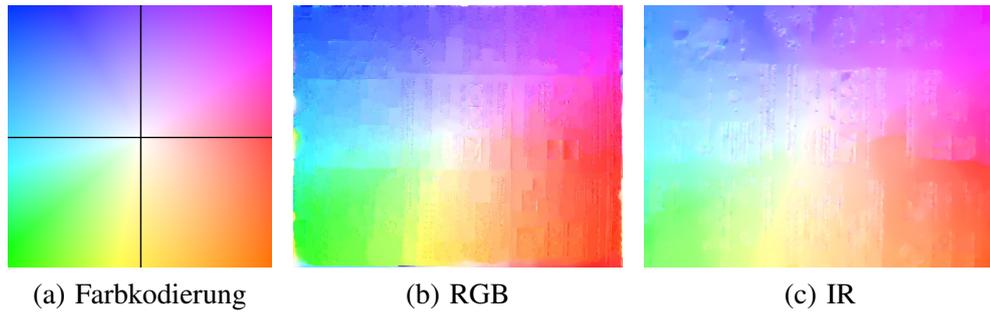


Abbildung 5.32: Visualisierung der Auswirkung des Wechsels des Wärmezustands auf die Kamerabilder mit Hilfe des optischen Flusses. Die Richtung eines Bewegungsvektors (Verschiebung) wird im Farbkodierungsschema (a) durch den Farbton und der Betrag durch die Farbsättigung dargestellt. Veränderungsfreie (bewegungsfreie) Gebiete sind in weiß gekennzeichnet. Ein sattes Grün deutet z.B. eine starke Bewegung nach unten links an. Das Ergebnis des optischen Flusses, angewendet auf das Poster beinhaltende Bildpaar der RGB- bzw. der IR-Kamera, ist in (b) bzw. (c) abgebildet.

ausgewertet worden. Die Mittelwerte und die Standardabweichungen für den maximalen Fluss sowie die minimalen und maximalen Werte für den Fluss in horizontaler und vertikaler Richtung sind in Tabelle 5.6 zusammengefasst. Die Standardabweichungen deuten auf einen geringen Einfluss des Rauschens hin.

Kamera	Bildauflösung		Maximaler Fluss	Horizontal min / max	Vertikal min / max
RGB	1280×1024	μ	6.6565	-6.579 / 4.789	-4.940 / 3.495
		σ	0.3737	0.384 / 0.169	0.111 / 0.223
IR	640×480	μ	1.8151	-1.721 / 1.640	-1.330 / 1.051
		σ	0.0670	0.089 / 0.089	0.023 / 0.060

Tabelle 5.6: Einfluss des Rauschens in den Eingabebildern (Poster) auf die Resultate des optischen Flusses. Die Mittelwerte (μ) und die Standardabweichungen (σ) sind für zehn Posterpaare ausgewertet worden. Alle Werte sind in Pixeln angegeben.

5.4.1.3 Kameraparameter basierter Vergleich

Falls die in den Abschnitten 5.4.1.1 und 5.4.1.2 beobachteten Effekte durch eine thermisch bedingte Deformation oder Verschiebung des optischen Linsensystems der Kameras hervorgerufen worden ist, so müsste auch eine Veränderung der Kameramodellparameter (intrinsische sowie Verzerrungsparameter) beobachtbar sein. Um dies zu überprüfen wurde zunächst bei abgedecktem Projektor ein im infraroten Bereich sichtbares Kalibrierobjekt in unterschiedlichen Posen von beiden internen Kameras aufgezeichnet. Dieses Vorgehen wurde sowohl im kalten als auch im warmen Zustand der Kinect wiederholt. Anschließend wurden auf diesen Bildmengen die Kamerakalibrierungen, wie in Abschnitt 2.4 beschrieben, durchgeführt. Die Resultate der Kalibrierungen sind in der Tabelle 5.7 zusammengefasst. Anhand des Vergleichs der ermit-

	α	β	u_0	v_0	κ_1	κ_2	κ_3	ρ_1	ρ_2
RGB									
Kalt	1041.60	1043.29	656.70	520.56	0.1831	-0.5199	0.4520	0.00012	0.00122
Warm	1046.25	1048.14	656.87	523.67	0.1918	-0.5546	0.5011	0.00069	0.00144
Differenz	4.65	4.85	0.17	3.10	0.0086	-0.0347	0.0492	0.00057	0.00022
IR									
Kalt	586.02	586.78	321.27	239.21	-0.1014	0.4648	-0.6413	-0.00112	-0.00009
Warm	587.14	588.04	322.70	238.15	-0.1128	0.5179	-0.7146	-0.00213	-0.00013
Differenz	1.12	1.25	1.43	-1.06	-0.0114	0.0531	-0.0733	-0.00101	-0.00004

Tabelle 5.7: Vergleich der Kalibrierparameter der RGB- (oberer Tabellenteil) und der IR-Kamera (unterer Tabellenteil), ermittelt für den kalten und warmen Zustand der Kinect.



Abbildung 5.33: Verdeutlichung des Zusammenhangs zwischen der Brennweite und dem Zoom-Effekt (Vergrößerung). In (a) wird im zweidimensionalen Fall der Blickwinkel für eine kleine (grün) und eine große Brennweite (rot) dargestellt. Die resultierende Szene, hier aus Äpfeln bestehend, erscheinen im Kamerabild (b) entsprechend klein und in (c) entsprechend groß.

telten Parameter in den beiden Wärmezuständen wird deutlich, dass eine Deformation bzw. Verschiebung im optischen Linsensystem stattgefunden hat.

Bei beiden Kameras wurde im warmen Zustand eine größere Brennweite ermittelt. Im Allgemeinen hat eine Vergrößerung der Brennweite eine Verkleinerung des Blickwinkels zur Folge. Somit wird ein kleinerer Ausschnitt der Szene auf die Bildebene abgebildet, was einer Vergrößerung bzw. einem Zoom der sichtbaren Objekte entspricht (vgl. Abb 5.33). Folglich ist die beobachtete Brennweitenänderung konsistent mit den Beobachtungen in den Abschnitten 5.4.1.1 und 5.4.1.2.

Zur Überprüfung der Stabilität des Kalibrierverfahrens und somit der der Brennweitenparameter wurden beide Kameras zu einem späteren Zeitpunkt erneut kalibriert. Dabei wurde das Kalibrierobjekt in beiden Wärmezuständen in verschiedenen Posen vor jeder Kamera ausgerichtet und für jede Pose sind jeweils zehn Aufnahmen des Kalibrierobjekts aufgezeichnet worden. Die Parameter wurden anschließend zehnmal mit dem Kalibrierverfahren ermittelt, um den Einfluss des Bildrauschens zu bestimmen. Die Mittelwerte sowie die Standardabweichungen aller Parameter der zehn Kalibrierdurchgänge sind der Tabelle 5.8 zu entnehmen. Die geringen Standardabweichungen verdeutlichen die Stabilität der ermittelten Parameterwerte.

Alle drei Experimente (Bildbasiert, optischer Fluss, Kalibrierparameter) verdeutlichen den Einfluss der unterschiedlichen Wärmezustände auf die internen Kameras der Ki-

		α	β	u_0	v_0	κ_1	κ_2	κ_3	ρ_1	ρ_2
RGB										
Kalt	μ	1041.35	1041.95	655.10	520.52	0.1943	-0.5836	0.5490	-0.00021	0.00092
	σ	0.0839	0.0806	0.1147	0.0866	0.0006	0.0033	0.0051	0.00002	0.00004
Warm	μ	1046.03	1046.69	655.44	521.03	0.2031	-0.6340	0.6438	-0.00018	0.00103
	σ	0.0948	0.0968	0.1204	0.0498	0.0005	0.0031	0.0056	0.00002	0.00004
Diff.		4.68	4.74	0.34	0.51	0.0088	-0.0504	0.0948	0.00003	0.00011
IR										
Kalt	μ	586.17	585.96	321.26	237.74	-0.1028	0.4851	-0.6933	-0.00229	-0.00038
	σ	0.0511	0.0517	0.0748	0.0359	0.0006	0.0032	0.0051	0.00002	0.00003
Warm	μ	587.48	587.29	320.99	238.67	-0.0986	0.4597	-0.6434	-0.00150	-0.00052
	σ	0.0472	0.0430	0.0372	0.0352	0.0005	0.0025	0.0036	0.00002	0.00002
Diff.		1.31	1.33	-0.27	0.93	0.0042	-0.0254	0.0499	0.00079	-0.00014

Tabelle 5.8: Untersuchung der Auswirkung von Bildrauschen auf die Stabilität der ermittelten Parameterwerte. Angegeben sind jeweils der Mittelwert (μ) und die Standardabweichung (σ) der Parameter für je zehn Kalibrierdurchgänge der RGB- (oberer Tabellenteil) und der IR-Kamera (unterer Tabellenteil). Die Differenz bezieht sich auf die jeweiligen Mittelwerte.

nect, insbesondere der RGB-Kamera. Wird z.B. die Kalibrierung der Kameras in einem anderen Wärmezustand als nachfolgende Messungen durchgeführt, hat dies zur Folge, dass die Parameter nicht mehr passend sind und somit Messfehler verursacht werden können. Diese Erkenntnis ist z.B. für die Arbeiten [HKH11, ZZ11, SJP11] von besonderer Bedeutung, da dort die RGB-Kamera in den Tiefenkalibrierprozess mit einbezogen wird und sich folglich thermisch bedingte Änderungen negativ auf den Kalibrierprozess auswirken können. Weiterhin hat diese Erkenntnis eine Relevanz für Arbeiten wie z.B. [NLB⁺11], die eine 3D-Szenenrekonstruktion, basierend auf der Fusion der Bilddaten der RGB-Kamera und den Tiefendaten, durchführen ("RGB-Depth registration"), da eine Veränderung des betrachteten Bildausschnitts (Zoom-Effekt) Auswirkungen auf die Resultate haben kann. Um Fehler zu vermeiden, wird deshalb die folgende erste Regel aufgestellt:

Regel 1: Die Durchführung der Kamerakalibrierung und nachfolgende Messungen sollten im gleichen thermischen Zustand der Kinect durchgeführt werden.

5.4.2 Thermischer Einfluss auf die Projektion des strukturierten Lichts

Wie in Abschnitt 4.5.1 beschrieben, wird das strukturierte Punktmuster ("speckle pattern") vom Projektor in die Szene geworfen und von der IR-Kamera erfasst. Wie sich thermische Veränderungen auf die optischen Elemente und somit auf das projizierte Muster auswirken, wird nachfolgend untersucht. Wie aus Abschnitt 5.4.1 bekannt, haben die thermischen Änderungen einen Einfluss auf die IR-Kamera. Um die Auswirkungen auf den Projektor unabhängig von diesen beobachten zu können, wurde das Experiment mit Hilfe von zwei Kinects durchgeführt, die im Folgenden als projizierende Kinect und beobachtende Kinect bezeichnet werden. Der Versuchsaufbau ist in

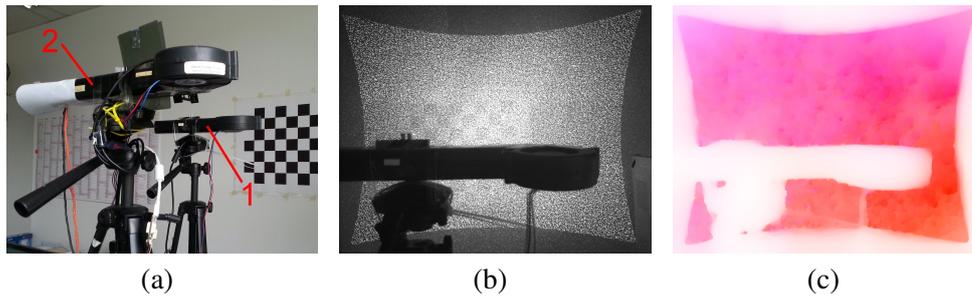


Abbildung 5.34: Der Versuchsaufbau für die Untersuchung des Einflusses thermaler Änderungen auf das strukturierte Punktmuster ("speckle pattern") ist in (a) dargestellt. Die projizierende Kinect (1) wirft das Punktmuster zunächst im warmen und dann im kalten Zustand an die Wand. Die IR-Kamera der beobachtenden Kinect (2) erfasst die Bilder des Punktmusters von beiden Wärmezuständen (b). Das Resultat des optischen Flusses (vom warmen zum kalten Zustand) zwischen diesen beiden Bildern ist in (c) gegeben. Die rötliche Färbung in den durch das Punktmuster abgedeckten Bereichen deutet auf eine Bewegungstendenz in Richtung der positiven u -Achse hin (Rechtsbewegung, vgl. Farbkodierung in Abb. 5.32(a)).

Abb. 5.34(a) zeigt. Die projizierende Kinect wurde im Abstand von 1.05 m frontal vor einer planaren Wand positioniert. Da das Punktmuster normalerweise einen größeren Bereich bestrahlt als die IR-Kamera bei gleichem Abstand erfassen kann, wurde die beobachtende Kinect 0.4 m hinter der projizierenden Kinect aufgestellt. Somit konnte das gesamte Punktmuster von der IR-Kamera der beobachtenden Kinect erfasst werden. Um stabile thermische Bedingungen der beobachtenden Kinect zu sichern, war ihr extern montierter Lüfter bei diesem Experiment permanent in Betrieb. Der externe Lüfter der projizierenden Kinect war zunächst deaktiviert. Beide Kinects wurden vor Experimentbeginn eine Stunde lang betrieben, um stabile Temperaturen der internen Komponenten sicherzustellen. Die Raumtemperatur lag während der gesamten Experimentphase bei 28.5°C.

Zunächst ist das Muster der projizierenden Kinect im warmen Zustand aufgezeichnet worden. Anschließend wurde ihr externer Lüfter für 30 Minuten aktiviert und schließlich im kalten Zustand erneut von der IR-Kamera der beobachtenden Kinect erfasst (siehe Abb. 5.34(b)). Um Bewegungen der projizierenden Kinect auszuschließen, ist der Lüfter langsam beschleunigt worden. Zudem war die Kinect auf einem Stativ fixiert. Auf diesen beiden Bildern wurde der bereits in Abschnitt 5.4.1.1 beschriebene Optical-Flow-Ansatz angewendet. Das Resultat des optischen Flusses vom warmen zum kalten Zustand ist in Abb. 5.34(c) dargestellt. Während für alle Bildteile außerhalb des Punktmusters keine Bewegung beobachtbar war (durch weißliche Färbung angedeutet), wies die vom Punktmuster abgedeckte Fläche deutliche Verschiebungstendenzen nach rechts auf (an der rötlichen Färbung erkennbar, vgl. Farbkodierung in Abb. 5.32(a)). Der berechnete optische Fluss lag in u -Richtung im Intervall von $[0.025, 1.397]$ und in v -Richtung im Intervall von $[-0.383, 0.449]$ Pixeln. Die weißlich gefärbte Fläche, die in Abb. 5.34(c) an der Stelle zu sehen ist, die zur projizierenden Kinect gehört, verdeutlicht die starre Konfiguration der beiden Kinects zueinander.

5.4.3 Auswirkung der Umwelt- und thermalen Einflüsse auf Distanzmessungen

Zum einen wird im Folgenden untersucht, wie die in Abschnitt 5.4.1 beschriebenen Effekte sich indirekt auf Distanzmessungen auswirken, die auf der internen RGB-Kamera basieren. Zum anderen wird überprüft, wie sich die Umwelteinflüsse auf die von der Kinect ermittelten Distanz- bzw. Tiefenwerte (Tiefenbild) auswirken. Nach einer kurzen Schilderung des Versuchsaufbaus und der verwendeten Messverfahren werden die einzelnen Experimente und die Versuchsergebnisse erläutert.

5.4.3.1 Versuchsaufbau und Distanzmessungen

Der statische Versuchsaufbau ist in Abb. 5.35 gegeben und bestand im Wesentlichen aus der Kinect mit dem bereits aus Abschnitt 5.4.1 bekannten extern montierten Lüfter, der beliebig aktiviert werden konnte, und dem Kalibrierobjekt der Größe 0.75×1.0 m. Das Kalibrierobjekt wurde in etwa 1.5 m von der Kinect entfernt platziert. Da bei den Experimenten die Tiefenmessung (Tiefenbild) der Kinect benötigt wurde, ist ein im infraroten Bereich unsichtbares Schachbrettmuster eingesetzt worden, um die Erfassung des infraroten strukturierten Lichts nicht zu stören (vgl. Abschnitt 5.3.1.3). Zur Stabilisierung der Beleuchtungsbedingungen für die RGB-Kamera wurde eine Kompaktleuchtstofflampe eingesetzt. Diese emittiert nur einen verhältnismäßig geringen Anteil an infraroter Strahlung und verhindert somit eine Überbelichtung, die die Detektion des strukturierten Lichts stören könnte. Weiterhin wurde ein Temperatursensor eingesetzt, der ca. 10 cm unterhalb der Kinect montiert war und zur Erfassung der Umgebungstemperatur diente.

Die im Folgenden untersuchten Distanzmessungen wurden auf zweierlei Arten bestimmt. Die erste Distanzmessung basierte, ausgehend vom aktuellen RGB-Bild, wel-

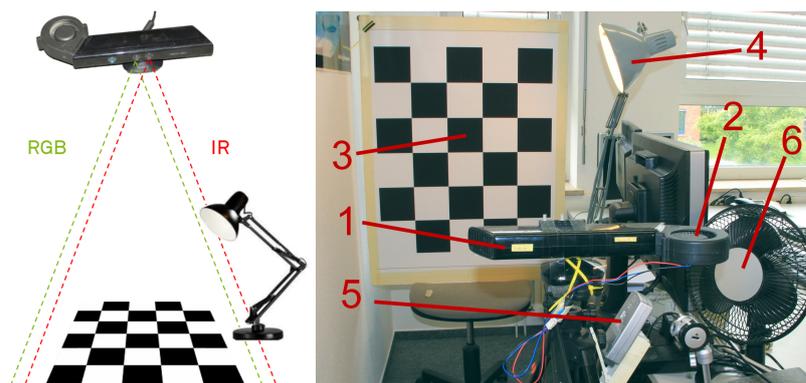


Abbildung 5.35: Der statische Versuchsaufbau mit der Kinect (1), dem extern montierten Lüfter (2), dem Kalibrierobjekt (3), einer Kompaktleuchtstofflampe (4), einem Thermometer (5) und einem Tischventilator (6). Der Abstand zwischen der Kinect und dem Kalibrierobjekt lag bei etwa 1.5 m, so dass das Kalibrierobjekt vollständig in das RGB-Kamerabild sowie in das Tiefenbild passte.

ches das Kalibrierobjekt beinhaltet, direkt auf der Schätzung der dreidimensionalen Lage des Kalibrierobjekts relativ zur Kamera (vgl. Abschnitt 5.3.3). Nach der Transformation in das IR-Kamerakoordinatensystem (vgl. Abschnitt 5.3.4) wurden Geraden konstruiert, die durch die Pixel im relevanten Bereich verliefen und die Modellebene des Kalibrierobjekts schnitten (vgl. Abschnitt 5.3.5.1). Der Mittelwert der Distanzen in Z-Richtung zwischen dem Kamerazentrum und den Schnittpunkten ergab schließlich die gesuchte Distanz. Diese, auf der Modellebene basierte Distanz wird nachfolgend mit \mathcal{D}_M bezeichnet.

Die zweite Distanzmessung basierte direkt auf dem Tiefenbild der Kinect. Dazu wurde zunächst der relevante Bereich im Tiefenbild ermittelt (siehe Abb. 5.22). Durch die Berechnung des Mittelwerts der Tiefenwerte aller Pixel, die sich innerhalb des relevanten Bereichs befanden, ergab sich schließlich die zweite Distanzmessung. Die auf dem Tiefenbild basierende Distanzmessung wird im weiteren Verlauf mit \mathcal{D}_T bezeichnet. Weitere Details zu den Distanzberechnungen können dem Abschnitt 5.3.5.2 und dem Algorithmus 5 entnommen werden.

Es sei angemerkt, dass die im Folgenden durchgeführten Experimentreihen teilweise zu unterschiedlichen Zeitpunkten stattfanden und der Versuchsaufbau teilweise neu aufgestellt worden ist, so dass der tatsächliche Abstand zwischen der Kinect und dem Kalibrierobjekt variierte. Innerhalb einer Experimentreihe war der Versuchsaufbau jedoch immer statisch, so dass gemessene Distanzveränderungen nur auf Umwelt- und Temperatureinflüsse zurückzuführen sind.

5.4.3.2 Distanzveränderungen während der Aufwärmphase

Um alle Komponenten der Kinect vollständig abzukühlen, wurde die Strom- und USB-Verbindung für drei Stunden getrennt. Während des Experiments lag die Umgebungstemperatur bei ca. 27.7°C . Der externe Lüfter blieb deaktiviert. Die Kinect wurde gestartet und die Veränderungen der Modellebenen basierten Distanzmessung \mathcal{D}_M sowie der Tiefenbild basierten Distanzmessung \mathcal{D}_T während der Aufwärmphase der Kinect beobachtet. Die charakteristischen Verläufe innerhalb von 135 Minuten sind in Abb. 5.36 dargestellt. Die dazugehörigen quantitativen Details sind in der Tabelle 5.9 zusammengefasst. Wie dieser zu entnehmen ist, veränderte sich die Distanz \mathcal{D}_M trotz der statischen Szene im Verlauf des Experiments um -6.97 mm. 90% dieser Veränderung war erst nach etwa 60 Minuten Aufwärmzeit vollzogen. Es wird angenommen, dass diese Distanzänderung eine direkte Folge des bereits in Abschnitt 5.4.1.1 beob-

Distanzmessung	Beginn [mm]	Ende [mm]	Veränderung [mm]	90% Veränderung [min]
Tiefenbild \mathcal{D}_T	1484.54	1504.03	19.49	41.0
Modellebene \mathcal{D}_M	1499.68	1492.71	-6.97	62.9

Tabelle 5.9: Quantitative Details zu den Verläufen der Distanzen aus Abb. 5.36. Angegeben sind die jeweiligen Distanzen bei Beginn und am Ende des Experiments sowie die Veränderung in den Distanzwerten. Zusätzlich ist angegeben, nach welcher Zeit 90% der Gesamtänderung vollzogen war.

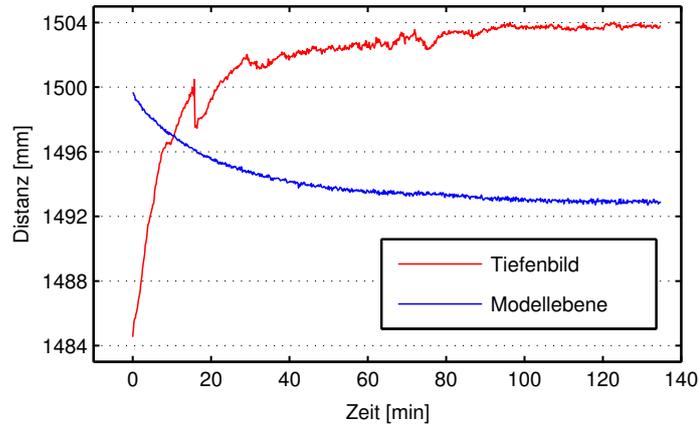


Abbildung 5.36: Der Verlauf der Tiefenbild basierten Distanzmessung \mathcal{D}_T (rot) sowie der Modellebenen basierten Distanzmessung \mathcal{D}_M (blau) innerhalb der ersten 135 Minuten der Aufwärmphase nach Aktivierung der Kinect. Die sprunghafte Veränderung der roten Kurve in der 16. Minute des Experiments wird in Abschnitt 5.4.3.3 diskutiert.

achteten Zoom-Effekts ist, der im warmen Zustand zu beobachten war. Somit erscheint auch das Kalibrierobjekt im RGB-Kamerabild vergrößert, was aufgrund der perspektivischen Projektion üblicherweise nur der Fall ist, wenn sich das Objekt näher an der Kamera befindet. Folglich verringerte sich fälschlicherweise die gemessene Distanz \mathcal{D}_M , obwohl die Szene unverändert blieb.

Die gemessene Distanz \mathcal{D}_T hingegen stieg im Verlauf des Experiments um 19.49 mm an, was bei der vorliegenden Entfernung zum Kalibrierobjekt einer Veränderung von 1.3% während der Aufwärmphase entspricht. 90% dieser Veränderung war nach 41 Minuten abgeschlossen. Ein vergleichbares Verhalten der Tiefenmesswerte wurde z.B. für die SwissRanger SR2 Time-of-Flight Kamera in [KRI06] beschrieben, deren Distanzmessung sich während der Aufwärmphase um etwa 12 mm innerhalb von ca. 30 Minuten erhöhte.

Das beschriebene Experiment wurde bei einer Umgebungstemperatur von 22.5°C wiederholt. Die ermittelten Distanzveränderungen betragen 19.68 mm (90% innerhalb von 42 Minuten) für \mathcal{D}_T und -6.12 mm (90% innerhalb von 56 Minuten) für \mathcal{D}_M .

Als Schlussfolgerung der Beobachtungen wird die folgende Regel zur Fehlervermeidung aufgestellt:

Regel 2: Die Beachtung einer Aufwärmzeit von bis zu 60 Minuten ist notwendig, um stabile Messbedingungen zu erhalten.

5.4.3.3 Distanzunterschiede zwischen Thermalzuständen

Bei konstanter Umgebungstemperatur von 27.5°C wurde der extern montierte Lüfter (siehe Abb. 5.30(a)) alternierend aktiviert und wieder deaktiviert, um zwischen den

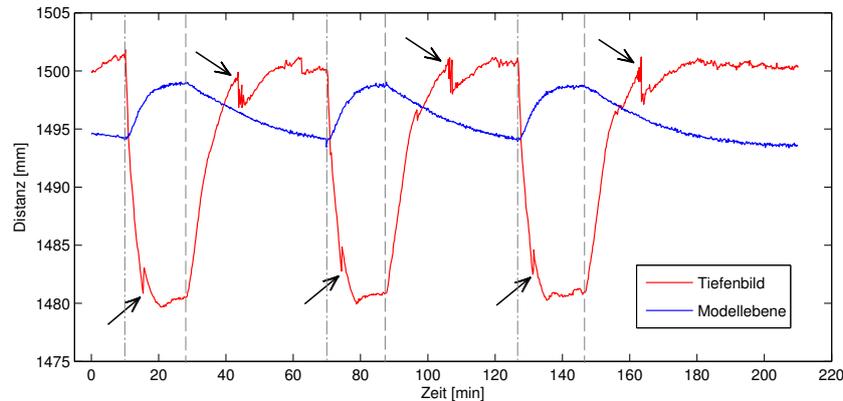


Abbildung 5.37: Der Verlauf der Distanzen \mathcal{D}_T (rot) und \mathcal{D}_M (blau) bei einer konstanten Umgebungstemperatur von ca. 27.5°C während alternierender Phasen mit Kühlung durch den extern montierten Lüfter. Die strichpunktierten Linien markieren Zeitpunkte, an denen der Lüfter aktiviert und die gestrichelten, an denen er deaktiviert worden ist. Innerhalb der Kühlphasen verringerte sich \mathcal{D}_T während \mathcal{D}_M anstieg. Die Pfeile markieren Zeitpunkte, an denen sprunghafte Distanzänderungen beobachtet werden konnten.

Wärmezuständen der Kinect zu wechseln. Es wurden drei Abkühl- und Aufwärmphasen untersucht und der Verlauf der beiden Distanzen aufgezeichnet (siehe Abb. 5.37). Die Abkühlphasen waren nach etwa 10 bzw. 18 Minuten bei Betrachtung von \mathcal{D}_T bzw. \mathcal{D}_M abgeschlossen. Für \mathcal{D}_M war die längste ermittelte Aufwärmphase nach 61 Minuten beendet. Dies ist vergleichbar mit den Resultaten aus Abschnitt 5.4.3.2. Für \mathcal{D}_T dauerte die Aufwärmphase bis zu 33 Minuten. Trotz konstanter Raumtemperatur und dem statischen Versuchsaufbau hatten die unterschiedlichen Wärmezustände der Kinect einen signifikanten Einfluss auf beide Distanzmessarten. Während der Kühlphasen veränderte sich der gemessene Distanzwert \mathcal{D}_M um bis zu 5.67 mm und \mathcal{D}_T um bis zu -22.75 mm. In den Abkühlphasen vollzog sich der Zoom-Effekt in entgegengesetzter Richtung (Vergrößerung des Blickwinkels, Objekte erscheinen kleiner), weshalb im Gegensatz zur in Abschnitt 5.4.3.2 untersuchten Aufwärmphase ein wachsender Distanzwert \mathcal{D}_M gemessen wurde.

Als Ursache für die gemessene negative Distanzänderung für \mathcal{D}_T wird der Einfluss der Thermalzustände auf das strukturierte Licht angenommen, wie in Abschnitt 5.4.2 beschrieben. Die Kinect besitzt ein vorberechnetes Referenzpunktmuster, auf dem die Disparitäts- und die Tiefenberechnungen basieren (vgl. Abschnitt 4.5.3). Sei die in Abb. 5.38 in rot dargestellte, durchgezogene Linie der vorberechnete Projektionsstrahl eines bestimmten Punktes des Punktmusters. Dieser Punkt wird auf ein Objekt projiziert (durch \mathbf{w}' angedeutet) und ist im IR-Kamerabild an der Position \mathbf{w} sichtbar. Entsprechend der Berechnungsformel (4.25) in Abschnitt 4.5.3, kann die Distanz $d_{w'}$ berechnet werden. Bei einer Änderung des Wärmezustands von warm zu kalt lag, wie in Abschnitt 5.4.2 beobachtet, eine Rechtsverschiebung des Punktmusters vor (vgl. Abb. 5.34(c)). Wird nun durch den thermalen Einfluss der tatsächliche Projektionsstrahl von der angenommenen (vorberechneten) Richtung abgelenkt (gestrichelte rote

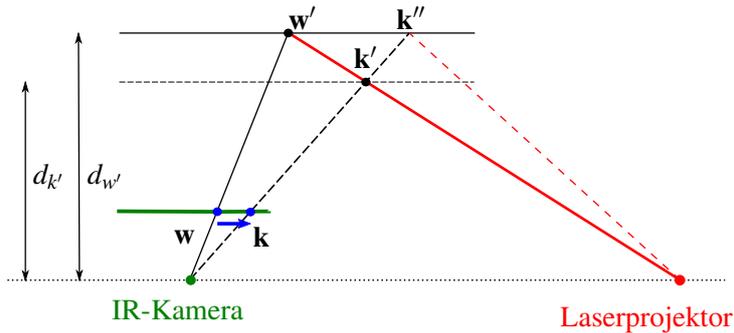


Abbildung 5.38: Die Veränderung des Wärmezustands (von warm nach kalt) hat eine Verschiebung des projizierten strukturierten Punktmusters ("speckle pattern") nach rechts zur Folge (siehe Abschnitt 5.4.2). Demnach kann auch eine Verschiebung des Musters im Bild der IR-Kamera beobachtet werden (blauer Pfeil). Dies hat eine Auswirkung auf die beobachtete Disparität und folglich auf die resultierende Distanzbeurteilung $d_{k'} < d_{w'}$ (vgl. Abschnitt 4.5.3).

Linie), so wird der Punkt des Punktmusters auf eine verschobene Position auf dem Objekt projiziert (\mathbf{k}'') und folglich an einer nach rechts verschobenen Position \mathbf{k} im Bild der IR-Kamera beobachtet (blauer Pfeil). Da das gespeicherte Punktmuster jedoch als statisch angenommen und die Ablenkung des Strahls somit nicht berücksichtigt wird, wird bei der Berechnung vom ursprünglichen Projektionsstrahl ausgegangen (durchgezogene rote Linie) und folglich eine geringere Distanz $d_{k'} < d_{w'}$ zum Punkt \mathbf{k}' ermittelt.

Die Pfeile in Abb. 5.37 markieren die Zeitpunkte, an denen sprunghafte Veränderungen der Distanzmessung \mathcal{D}_T in der Größenordnung von 2 bis 4 mm beobachtet wurden. Diese Sprünge traten in Fällen auf, in denen eine rapide Änderung der gemessenen Distanzwerte vorlag. Bei fallenden Distanzwerten war eine Aufwärtskorrektur zu beobachten und umgekehrt. Es sei angemerkt, dass an den markierten Zeitpunkten auch eine sprunghafte und globale Veränderung der Werte in den entsprechenden Tiefenbildern beobachtet werden konnte, die die Grundlage der Berechnungen von \mathcal{D}_T darstellen.

5.4.3.4 Rauschuntersuchung in beiden Thermalzuständen

Wie sich unterschiedliche thermische Zustände auf das Rauschen auswirken, wird nachstehend beschrieben. Das Rauschen in den Distanzmessungen \mathcal{D}_T und \mathcal{D}_M bei Verwendung des statischen Versuchsaufbaus wurde anhand von Standardabweichungen ermittelt. Die Raumtemperatur lag bei 23.7°C während des gesamten Experiments. Es sind jeweils über 660 Distanzmessungen innerhalb eines Zeitraums von 15 Minuten erfasst und die Standardabweichungen berechnet worden. Die Evaluierung des kalten Zustands wurde nach einer Abkühlphase von 30 Minuten durchgeführt, in der der extern montierte Lüfter aktiviert war. Der warme Zustand ist nach einer 60 minütigen Aufwärmphase bei deaktiviertem Lüfter untersucht worden. Tabelle 5.10 fasst die Ergebnisse zusammen.

	Minimalwert	Maximalwert	Differenz	μ	σ
Kalter Zustand					
Tiefenbild \mathcal{D}_T [mm]	1457.07	1457.58	0.51	1457.32	0.103
Modellebene \mathcal{D}_M [mm]	1487.49	1487.81	0.32	1487.66	0.058
Warmer Zustand					
Tiefenbild \mathcal{D}_T [mm]	1474.12	1474.86	0.74	1474.53	0.134
Modellebene \mathcal{D}_M [mm]	1482.50	1482.72	0.22	1482.61	0.038

Tabelle 5.10: Untersuchung des Messrauschens in beiden Wärmeszuständen der Kinect bei konstanter Umgebungstemperatur (23.7°C) unter Verwendung des statischen Versuchsaufbaus. Der Rauschwert ist jeweils als Standardabweichung σ mit dem dazugehörigen Mittelwert μ der mehr als 660 zugrundeliegenden Messpunkte angegeben. Zusätzlich sind die ermittelten Extremwerte sowie deren Differenz aufgeführt.

Bei Betrachtung des Übergangs vom kalten zum warmen Zustand konnte für \mathcal{D}_T eine Rauscherhöhung von 30.1% festgestellt werden. Für \mathcal{D}_M lag hingegen eine Rauschreduktion um 34.5% vor. Als Ursache für die Reduktion wird die in Abschnitt 5.4.1.1 beschriebene Bildunschärfe angenommen. Diese ist bei der RGB-Kamera im warmen Zustand beobachtet worden und besitzt eine Tiefpasscharakteristik, die sich rauschreduzierend auf die Detektion des Kalibrierobjekts im RGB-Bild und somit auch auf die Bestimmung der Lage der Modellebene des Kalibrierobjekts auswirkt. Der in Tabelle 5.10 angegebene Rauschwert für \mathcal{D}_M im warmen Zustand ist vergleichbar mit dem bereits in Abschnitt 5.3.6.5 ermittelten Rauschwert für die flächenbasierten Referenzmessungen. In beiden Wärmeszuständen konnte für \mathcal{D}_M ein geringeres Rauschen als für \mathcal{D}_T ermittelt werden.

5.4.3.5 Einfluss des kinectinternen Lüfters

Die Kinect besitzt einen kleinen, innerhalb des Gehäuses verbauten Lüfter (siehe Abb. 5.39(a)), der eingesetzt wird, um eine Überhitzung der internen Komponenten zu verhindern. Es existiert keine Schnittstelle, die eine manuelle Kontrolle über den internen Lüfter erlaubt. Stattdessen wird dieser automatisch aktiviert, falls die Temperatur einen gewissen Schwellenwert überschreitet.

Bei dem durchgeführten Experiment war der externe Lüfter deaktiviert und die Aufwärmphase der Kinect bereits zuvor abgeschlossen. Die Umgebungstemperatur stieg im Verlauf des Experiments von 29.9 auf 31.1°C an. Der interne Lüfter wurde bei Erreichen einer Umgebungstemperatur von ca. 30.5°C automatisch aktiviert (Aktivierungsschwellenwert). Der Verlauf der Distanzmessungen ist in Abb. 5.39(b) dargestellt. Der Aktivierungszeitpunkt lag bei Minute 13 und ist durch die gestrichelte Linie angedeutet.

Die automatische Aktivierung wirkte sich in einer Distanzänderung von -12.19 bzw. 2.38 mm bei Betrachtung von \mathcal{D}_T bzw. \mathcal{D}_M aus. Tendenziell haben die Auswirkungen eine vergleichbare Charakteristik wie bereits in Abschnitt 5.4.3.3 beobachtet. Jedoch fallen die Veränderungen geringer aus (-12.19 mm gegenüber -22.75 mm bei \mathcal{D}_T und

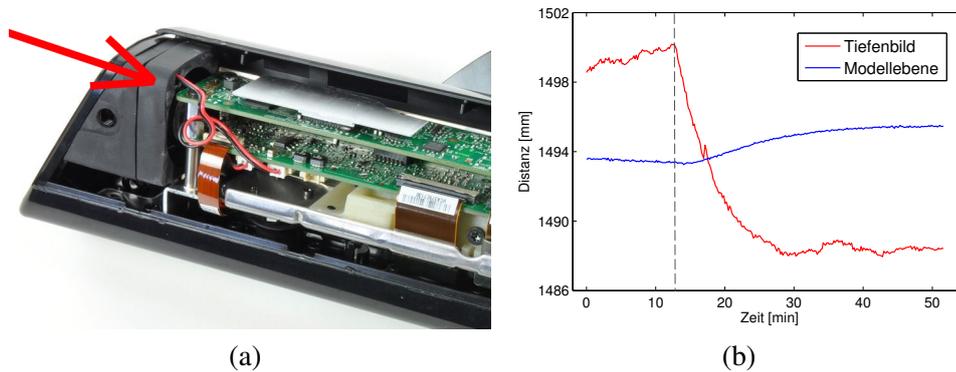


Abbildung 5.39: Der interne Lüfter (a) wird bei hohen Temperaturen zum Schutz der internen Komponenten automatisch aktiviert. Der Zeitpunkt der Aktivierung während des Experiments (Minute 13) ist durch die gestrichelte Linie angedeutet. Die Auswirkung ist anhand der Verläufe der gemessenen Distanzen in (b) verdeutlicht. Während der Abkühlphase ist die in Abschnitt 5.4.3.3 diskutierte sprunghafte Änderung in der auf dem Tiefenbild basierten Distanzmessung zu beobachten (Minute 17). (Bildquelle (a): <http://www.ifixit.com> (Lizenz: CC BY-NC-SA 3.0))

2.38 mm gegenüber 5.67 mm bei \mathcal{D}_M), was auf den weniger leistungsstarken Lüfter zurückgeführt wird. Dennoch ist der Einfluss des internen Lüfters signifikant, weshalb die folgende Regel aufgestellt wird:

Regel 3: Es ist notwendig, die Aktivierung des internen Lüfters zu beachten, falls die Kinect bei einer Umgebungstemperatur genutzt wird, die in der Nähe des Aktivierungsschwellenwerts liegt.

5.4.3.6 Auswirkung natürlicher Luftbewegungen

In einigen Vortests wurde der Einfluss natürlicher Luftbewegungen, wie z.B. ein leichter Durchzug bedingt durch gleichzeitig geöffnete Fenster und Türen beobachtet. Da diese spontanen und oftmals nur kurz andauernden Luftbewegungen nicht kontrollierbar und somit schwer experimentell auswertbar sind, wurden sie mit Hilfe eines Tischventilators (siehe Abb. 5.35) simuliert. Die Raumtemperatur lag bei 27.5°C . Sowohl der interne als auch der externe Lüfter waren deaktiviert und die Kinect war im warmen Zustand. Der Tischventilator wurde in 30 cm Entfernung auf die Kinect ausgerichtet und auf niedrigster Stufe für nur 10 Sekunden aktiviert. Der Einfluss auf \mathcal{D}_M war nicht signifikant. Bei Betrachtung von \mathcal{D}_T konnte jedoch trotz der sehr kurzen Zeitdauer des simulierten Durchzugs eine Veränderung von -3.22, -3.14 und -3.18 mm für drei Wiederholungen des Experiments ermittelt werden. Die notwendige Aufwärmzeit, um wieder die Ausgangsmesswerte zu erreichen, lag zwischen 5 und 6 Minuten. Dieses

Experiment verdeutlicht die Sensibilität der Kinect im warmen Zustand auf natürliche Umwelteinflüsse, selbst wenn die Umgebungstemperatur konstant bleibt. Deshalb wird die folgende Regel abgeleitet:

Regel 4: Wird die Kinect im warmen Zustand betrieben, so müssen Durchzug und Luftbewegungen vermieden werden.

5.4.3.7 Einfluss natürlicher Temperaturschwankungen

Die Sensibilität der Kinect gegenüber natürlichen Temperaturschwankungen in einem alltäglichen Anwendungsszenario wird in diesem Abschnitt verdeutlicht. Für das Experiment wurde der statische Versuchsaufbau (vgl. Abb. 5.35) in einem Büroraum der Größe $5 \times 3 \times 3$ m durchgeführt. Der zur Erfassung der Raumtemperatur verwendete Temperatursensor wurde etwa 10 cm unterhalb der Kinect montiert. Vor Beginn des Experiments war die Kinect in einer hinreichend langen Aufwärmphase im Online-Modus. Dabei waren sowohl die Tür als auch die Fenster geschlossen und die Raumtemperatur lag bei 26.2°C . Das Experiment ist einen Tag lang durchgeführt worden (9:53–23:59 Uhr, 846 Minuten) und sowohl der interne als auch der externe Lüfter waren deaktiviert. Zu Beginn des Experiments (Minute 0) wurde das Fenster geöffnet, jedoch blieben die Tür und die Außenjalousie des Fensters während des Experiments geschlossen, um den störenden Einfluss von Durchzug und Luftbewegungen zu verhindern (siehe Abschnitt 5.4.3.6). Aufgrund des offenen Fensters sorgten wetterbedingte Außentemperaturänderungen (Mischung aus regnerischen und sonnigen Abschnitten) für Temperaturschwankungen im Innenraum. In Abb. 5.40 ist der Verlauf der Innenraumtemperatur (grüne Kurve) sowie der Einfluss auf die beiden Distanzmessungen abgebildet. 497 Minuten nach Experimentbeginn wurde das Fenster geschlossen, so dass die Raumtemperatur wieder deutlich anstieg.

Bei Betrachtung des Verlaufs der auf dem Tiefenbild basierten Distanz \mathcal{D}_T wird eine positive Korrelation mit der Temperatur deutlich. Der Verlauf von \mathcal{D}_M weist hinge-

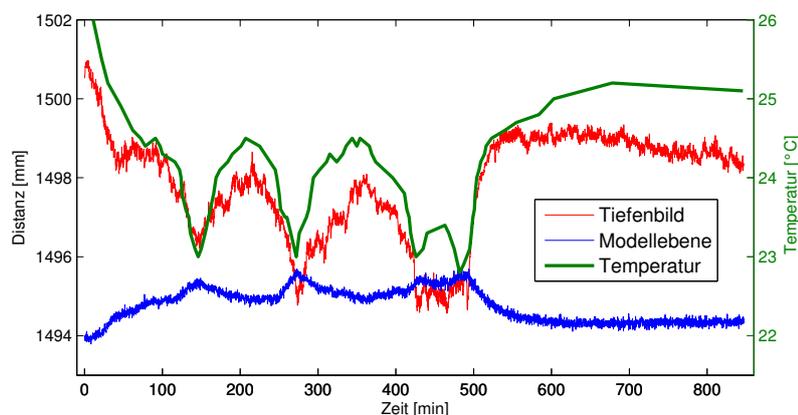


Abbildung 5.40: Verlauf der beiden Distanzmessungen \mathcal{D}_T (rot) und \mathcal{D}_M (blau) sowie der natürlich verursachten Schwankungen der Umgebungstemperatur (grün) im Verlauf eines Tages.

	Minimalwert	Maximalwert	Differenz	mm/°C
Temperatur [°C]	22.8	26.2	3.40	-
Tiefenbild \mathcal{D}_T [mm]	1494.57	1500.96	6.39	1.88
Modellebene \mathcal{D}_M [mm]	1493.79	1495.69	1.90	-0.56

Tabelle 5.11: *Quantitative Details zu den Verläufen der Distanzen sowie der natürlichen Temperaturschwankungen in Abb. 5.40. Die letzte Spalte gibt die Veränderung in den Distanzmesswerten pro 1 °C an.*

gen eine negative Korrelation auf. Dies ist konsistent mit dem bereits beobachteten Zoom-Effekt (siehe Abschnitt 5.4.1.1). Wie in Tabelle 5.11 zusammengefasst, lag der maximale Temperaturunterschied während des Experiments bei 3.4°C. Diese Temperaturspanne hatte eine maximale Änderung von -1.99 mm in \mathcal{D}_M und 6.39 mm in \mathcal{D}_T verursacht, was einer Veränderung von -0.56 und 1.88 mm pro 1°C entsprechen würde. Genau wie für \mathcal{D}_T wurde für die SR2 Time-of-Flight Kamera in [KRI06] eine Distanzzunahme bei steigender Temperatur beobachtet. Mit 8 mm pro 1°C bei ca. 2 m Objektstand war diese jedoch stärker ausgeprägt.

5.4.3.8 Korrelation zwischen den Distanzmessungen und der Temperatur

In diesem Experiment werden die Erkenntnisse aus Abschnitt 5.4.3.7 unter klar definierten Temperaturveränderungen bestätigt und genauer untersucht. Wie in der Abb. 5.41(a) dargestellt, wurde dazu ein Temperatursensor unmittelbar an den Lüftungsschlitzen der Kinect montiert. Im Gegensatz zur Messung mit dem Raumtemperatursensor (vgl. Abb. 5.35) kann davon ausgegangen werden, dass die gemessene Temperatur eher der tatsächlichen kinectinternen Temperatur entspricht, da die warme Abluft durch die Lüftungsschlitze direkt am Temperatursensor vorbeigeführt wird. Vor Beginn des Experiments wurde die Kinect aufgewärmt. Die gemessene Temperatur an den Lüftungsschlitzen lag bei 27.1°C, während die Raumtemperatur 22.0°C betrug. Die Außentemperatur lag mit 11.0°C deutlich darunter. Von Minute 5 bis Minute 75 wurde das Außenfenster geöffnet, um den Raum stetig abzukühlen. Ab der 75. Minute wurde das Fenster geschlossen und der Raum wieder erwärmt. Der gemessene Temperaturverlauf an den Lüftungsschlitzen sowie der Einfluss der Temperaturänderung auf die Distanzmessungen sind in Abb. 5.41(b) gezeigt. Dort markiert die gestrichelte Linie den Zeitpunkt des Wechsels von der Abkühl- zur Aufwärmphase des Raums.

Der in Abschnitt 5.4.3.7 beobachtete korrelative Zusammenhang konnte in diesem Experiment in einer deutlichen Ausprägung bestätigt werden. Der Korrelationskoeffizient der Korrelation zwischen \mathcal{D}_T und der Temperatur lag bei 0.96852, während sich ein negativer Koeffizient von -0.99562 für die Korrelation zwischen \mathcal{D}_M und der Temperatur beobachten ließ. Diese nahe an 1 bzw. -1 liegenden Koeffizienten deuten auf einen annähernd linearen Zusammenhang mit der Temperatur hin.¹ Der maximale Temperaturunterschied betrug 6.34°C. Pro 1°C entspricht dies einer Veränderung von

¹Nach Definition ist der Korrelationskoeffizient ein Maß für die lineare Abhängigkeit von zwei Zufallsvariablen, wobei ein idealer linearer Zusammenhang genau dann vorliegt, wenn der Korrelationskoeffizient exakt 1 bzw. -1 beträgt [BS08].

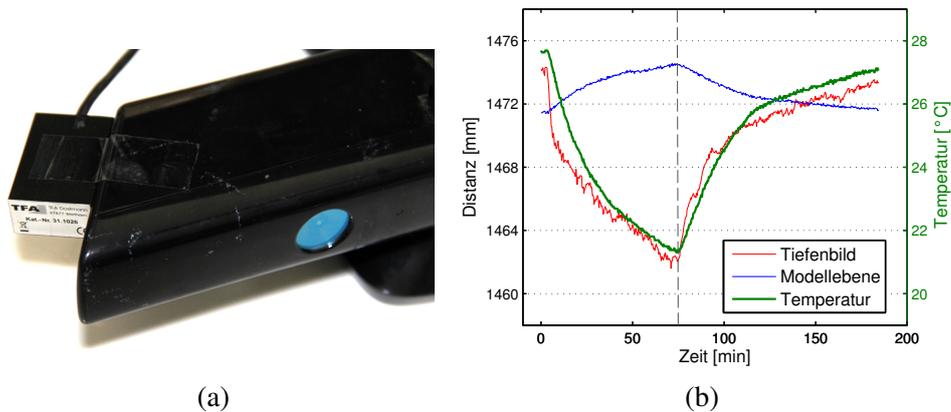


Abbildung 5.41: In (a) ist der direkt an den Lüftungsschlitzen der Kinect montierte Temperatursensor abgebildet. Der Verlauf der dort gemessenen Temperatur (grün) sowie jener der beiden Distanzmessungen ist in (b) abgebildet. Im ersten Teil des Experiments wurde die Umgebungstemperatur abgesenkt, während sie ab Minute 75 (gestrichelte Linie) wieder in etwa auf das Ausgangsniveau gebracht worden ist.

	Min	Max	Diff.	mm/°C	Korrelationskoeff.
Temperatur [°C]	21.31	27.69	6.38		
Tiefenbild \mathcal{D}_T [mm]	1461.60	1474.32	12.72	1.99	0.96852
Modellebene \mathcal{D}_M [mm]	1471.39	1474.56	3.17	-0.50	-0.99562

Tabelle 5.12: Quantitative Details zum Einfluss der kontrollierten Temperaturänderung auf die in Abb. 5.41 dargestellten Distanzverläufe. In der letzten Spalte ist der Koeffizient der Korrelation zwischen der Distanz und der Temperatur gegeben.

1.99 mm für \mathcal{D}_T und -0.50 mm für \mathcal{D}_M . Quantitative Details sind der Tabelle 5.12 zu entnehmen. Aufgrund der hier gewonnenen Erkenntnisse und den Beobachtungen aus Abschnitt 5.4.3.7 wird die folgende Regel aufgestellt:

Regel 5: Um stabile Distanzmessungen durchführen zu können, ist eine konstante Umgebungstemperatur notwendig.

5.4.3.9 Einfluss des Stand-By-Betriebs und Verbindungsunterbrechungen

Die in diesem Abschnitt beschriebenen Experimente sind unter konstanten Umgebungsbedingungen durchgeführt worden (Raumtemperatur 27°C, deaktivierter interner und externer Lüfter, Tür und Fenster geschlossen, kein Durchzug). Vor dem Beginn jedes Experiments wurde die Kinect eine Stunde lang im Online-Modus (Verarbeitung des RGB- und Tiefendatenstroms) aufgewärmt, bis ein stabiler Zustand der Messwerte erreicht war. Untersucht wurde, wie sich kurze Arbeitsunterbrechungen innerhalb von längeren Nutzungsperioden der Kinect auf die Distanzmessungen auswirken. Die im Folgenden untersuchten drei Unterbrechungstypen des Online-Modus sind die Tren-

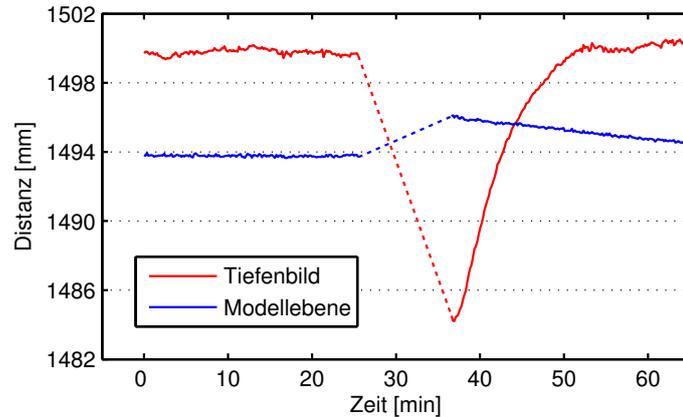


Abbildung 5.42: Veranschaulichung der zehnmütigen Unterbrechung der USB-Verbindung anhand der Verläufe der Distanzmessungen \mathcal{D}_T (rot) und \mathcal{D}_M (blau). Die gestrichelten Linienabschnitte deuten eine Unterbrechungsphase für die jeweiligen Distanzmessungen an.

nung der USB-Verbindung, die Trennung der Spannungsversorgung und das Betreiben der Kinect im Stand-By-Modus. Diese Untersuchung hat für die Praxis eine große Relevanz, weil solche Unterbrechungen während der Arbeit mit der Kinect (z.B. bei der Softwareentwicklung) oder beim Umbau eines Versuchsaufbaus typischerweise auftreten.

Typ 1: Trennung der USB-Verbindung. Die USB-Verbindung der Kinect zum PC ist jeweils für die Dauer von 2, 5 und 10 Minuten unterbrochen worden. Anschließend wurde die Verbindung wieder aufgebaut und die beiden Distanzmessungen sind fortgeführt worden. Zwischen den drei Unterbrechungen ist die notwendige Aufwärmzeit beachtet worden. Die Veränderung der Distanzwerte zwischen dem letzten Messzeitpunkt vor der Trennung und dem ersten Messzeitpunkt nach der Wiederverbindung wurde ausgewertet. Diese Werte lagen für die drei Unterbrechungszeiträume bei -6.12, -10.38 und -14.66 mm bei Betrachtung der auf dem Tiefenbild basierten Distanz \mathcal{D}_T . Bei der auf der Modellebene basierten Distanz \mathcal{D}_M wurden Veränderungen von 0.32, 1.21 und 2.36 mm gemessen. Es sei angemerkt, dass die ermittelten Werte zwar relativ klein, aber dennoch valide sind, da \mathcal{D}_T und \mathcal{D}_M auf durchschnittlichen Distanzen innerhalb des relevanten Bereichs basieren (vgl. Abschnitt 5.3.5.2), welche bei stabilen Umgebungsbedingungen einen geringen Rauschpegel unterhalb von $\sigma_{\mathcal{D}_T} = 0.14$ mm und $\sigma_{\mathcal{D}_M} = 0.06$ mm besitzen (vgl. Abschnitt 5.4.3.4). Für das Beispiel der zehnmütigen Unterbrechung ist der Verlauf beider Distanzmessungen in Abb. 5.42 dargestellt. Um erneut stabile Messwerte nach der zehnmütigen Unterbrechung zu erreichen, waren Aufwärmzeiten von 18 bzw. 57 Minuten für \mathcal{D}_T bzw. \mathcal{D}_M notwendig.

Typ 2: Trennung der Spannungsversorgung. Es konnte festgestellt werden, dass die Trennung der Spannungsversorgung (externes Netzteil) die gleichen Resultate aufwies, wie die zuvor beschriebene Trennung der USB-Verbindung. Der Grund dafür liegt darin, dass die Kinect den Energieverbrauch bereits vollständig einstellt, sobald

die USB-Verbindung getrennt wird. Folglich erzeugen die internen Hardwarekomponenten keine Wärme mehr, wie dies auch im Fall der Trennung der Spannungsversorgung ist. Aufgrund der resultierenden Abkühlung folgt eine Beeinflussung der gemessenen Distanzen.

Typ 3: Betrieb im Stand-By-Modus. Wenn die Kinect keinerlei Datenstrom verarbeitet (heruntergefahrener OpenNI-XnSensorServer [Ope10]), jedoch sowohl die Verbindung zur USB-Schnittstelle als auch zur Spannungsversorgung besteht, befindet sich die Kinect in einem Stand-By-Modus (durch grün blinkende LED an der Frontseite erkennbar). Bei Betrachtung einer Applikation, die die Kinect verwendet, ist das der typische Zustand zwischen dem Zeitpunkt des Beendens der Applikation und dem erneuten Zugriff auf die Kinect.

Nach der Aufwärmphase wurde der Stand-By-Modus für 15 Minuten betreten, bevor anschließend wieder in den Online-Modus zurück gewechselt und die Distanzmessung fortgesetzt wurde. Die gemessenen Veränderungen in \mathcal{D}_T und \mathcal{D}_M lagen bei -3.09 und 0.73 mm. Um den maximalen Einfluss zu ermitteln, wurde die Kinect für 10 Stunden in den Stand-By-Modus versetzt. Es konnte eine Änderung von -5.95 und 1.67 mm bei \mathcal{D}_T und \mathcal{D}_M festgestellt werden. Dies entspricht etwa 25% der Veränderungen während der Aufwärmphase (vgl. Abschnitt 5.4.3.2) und der Verwendung des externen Lüfters (vgl. Abschnitt 5.4.3.3). Im Gegensatz zum Unterbrechungstyp 1 und 2, verbrauchen die internen Komponenten der Kinect im Stand-By-Modus weiterhin Energie. Eine Messung am externen Netzteil ergab, dass im Online-Modus eine Leistungsaufnahme von etwa 4.7 Watt vorlag. Im Stand-By-Modus war diese geringer, lag aber noch bei ca. 3.4 Watt. Diese Leistungsaufnahme der internen Komponenten und die damit verbundene Abwärme verhinderte ein vollständiges Abkühlen im Stand-By-Modus und ermöglichte die vergleichsweise geringen Änderungen in der Distanzmessung von nur etwa 25%. Um Messfehler bei der Arbeit mit der Kinect (z.B. Softwareentwicklung), die typischerweise von Unterbrechungen geprägt sein kann, zu minimieren, wird die folgende Regel aufgestellt:

Regel 6: Die Kinect sollte möglichst immer im Online-Modus gehalten werden. Ist dies nicht möglich, bietet der Stand-By-Modus die beste Alternative.

5.4.3.10 Temperaturabhängigkeit der extern montierten RGB-Kamera

In Abschnitt 5.4.3.7 und 5.4.3.8 ist verdeutlicht worden, dass die auf der internen RGB-Kamera basierende Distanzmessung \mathcal{D}_M bei einer Entfernung von etwa 1.5 m eine Temperaturabhängigkeit von ca. -0.50 mm pro 1°C aufweist. Um diese Abhängigkeit in Relation mit einer weiteren Kamera in Beziehung setzen zu können, wurde die Distanzberechnung mit Hilfe der in Abschnitt 5.3.2 erwähnten externen RGB-Kamera (USB-Webcam) durchgeführt. Die Temperatur ist wie in Abschnitt 5.4.3.8 direkt an den Ventilationsschlitzen ermittelt worden. In der ersten Experimentphase erfolgte eine Abkühlung des Raums durch das Öffnen des Fensters (Außentemperatur ca. 9°C). Nach etwa 20 Minuten wurde der Raum bei geschlossenem Fenster wieder erwärmt.

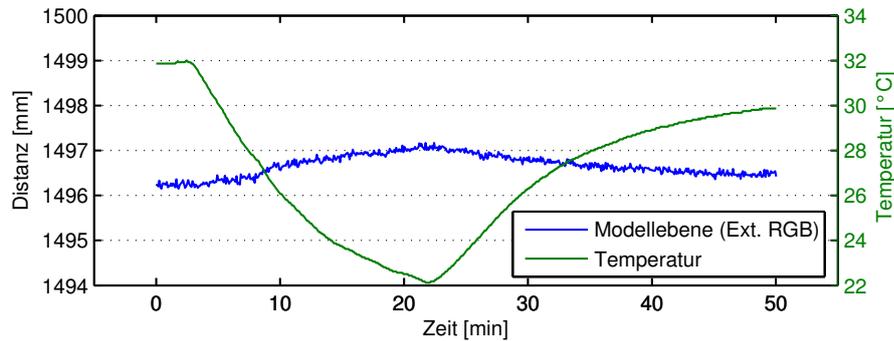


Abbildung 5.43: Temperaturabhängigkeit der Distanzmessung \mathcal{D}_M (blau) basierend auf der extern montierten RGB-Kamera (siehe Abb. 5.20). Die gemessene Distanz verhält sich reziprok zur Temperatur. Trotz einer großen Temperaturschwankung von ca. 10°C ändert sich die gemessene Distanz nur geringfügig um etwa 1 mm.

Das in Abschnitt 5.4.3.7 und 5.4.3.8 beobachtete reziproke Verhältnis zwischen der Temperatur und der gemessenen Distanz \mathcal{D}_M konnte auch für die extern montierte Kamera festgestellt werden (siehe Abb. 5.43). Trotz des großen maximalen Temperaturunterschieds von 9.9°C lag die maximale Distanzveränderung bei insgesamt nur 1.02 mm. Dies entsprach einer Veränderung von $-0.103\text{ mm pro }1^\circ\text{C}$, was in Relation zum Rauschwert (Standardabweichung von ca. 0.055 mm, vgl. Abb. 5.29(a)) verhältnismäßig gering war (etwa das 1.9-fache der Standardabweichung). Somit erwies sich die auf der kinectinternen RGB-Kamera basierende Distanzmessung im Vergleich als deutlich temperaturabhängiger (-0.50 mm gegenüber $-0.103\text{ mm pro }1^\circ\text{C}$, etwa Faktor 5). Dies ist eine relevante Erkenntnis für den Fall, dass während der Experimente Temperaturschwankungen vorliegen und \mathcal{D}_M als Referenztiefenmesswert verwendet werden soll.

5.5 Untersuchung der Temperatureuswirkung auf den MDP basierten Ansatz

In Abschnitt 5.2.3.6 wurde das Multi-Delta-Plane (MDP) basierte Korrekturverfahren beschrieben, mit dessen Hilfe der komplexe sensorortsabhängige Fehler korrigiert werden konnte. Inwiefern die Temperatur einen Einfluss auf das Korrekturverfahren hat, wird im Folgenden geprüft. Dabei wird zunächst der Einfluss auf das gesamte Tiefenbild und anschließend auf die MDP geprüft.

5.5.1 Auswirkung auf das gesamte Tiefenbild

Der Abschnitt 5.4 hat ausführlich dargelegt, dass Temperaturunterschiede eine deutliche Auswirkung auf die Komponenten sowie das Messverhalten des Kinect-Tiefensensors haben. Dabei wurde jeweils der Tiefenbildausschnitt untersucht, in dem sich das Kalibrierobjekt befand. Dieses ist primär so platziert worden, dass es den (typischerweise wichtigsten) mittleren Bereich des Sensors abdeckte. Da der MDP-Ansatz

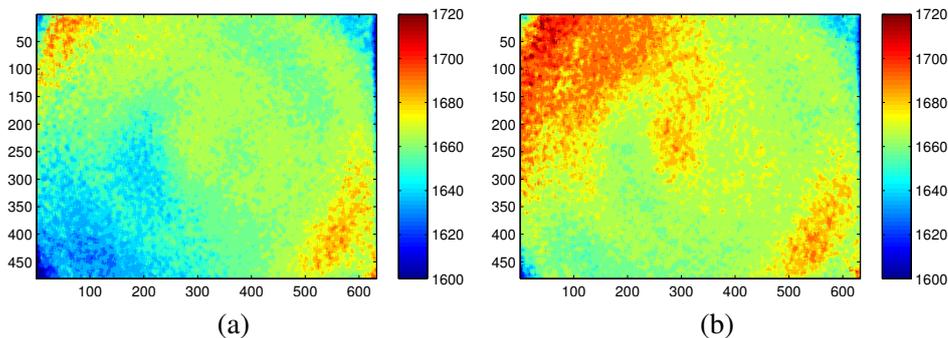


Abbildung 5.44: Verdeutlichung des Einflusses der Temperatur auf die Tiefenbilder. Zur Erzeugung der Tiefenbilder wurde die Kinect frontal auf eine etwa 1680 mm entfernte Wand gerichtet und auf einem Stativ fixiert. Bei der Aufzeichnung des Tiefenbildes in (a) betrug die Temperatur an den Lüftungsschlitzen der Kinect 25.18°C , in (b) 30.43°C . Die Veränderungen der Distanzwerte sind anhand der Farbskala erkennbar, deren Werte in Millimetern angegeben sind.

auf einzelnen Delta-Planes basiert, die ihrerseits den gesamten Bereich der Tiefenbilder zur Erzeugung benötigen, wird zunächst der Temperatureinfluss für das gesamte Tiefenbild und anschließend der Einfluss auf den MDP basierten Ansatz untersucht.

Für die Berechnung der in Abschnitt 5.2.3.1 beschriebenen Delta-Plane wurden Tiefenbilder eines frontal positionierten, planaren Objekts, in diesem Falle einer Wand, aufgezeichnet. Dieses Vorgehen wurde übernommen und die Kinect in der Distanz von 1680 mm zur Wand platziert. Anschließend ist je ein Tiefenbild bei der an den Lüftungsschlitzen gemessenen Temperatur von 25.18 und 30.43°C aufgezeichnet worden. Dabei wurde sichergestellt, dass die Position der Kinect zwischen den Aufzeichnungen unverändert geblieben ist. Die Abb. 5.44 zeigt die beiden erfassten Tiefenbilder. Die durch die Temperatur verursachte Veränderung der Tiefenwerte ist deutlich erkennbar.

Um die in Abb. 5.44 dargestellten Veränderungen in Relation zu den in dem Abschnitt 5.4.3.8 gewonnenen Erkenntnissen zu setzen, wurde ein Bereich innerhalb der in Abb. 5.44(a) und 5.44(b) dargestellten Tiefenbildern definiert, dessen Größe und Position dem durch den gelben Rahmen gekennzeichneten Bereich in Abb. 5.24 entspricht. Der Bereich hatte eine Größe von 258×296 Pixeln, dessen obere linke Ecke an der Pixelposition (187, 100) lag. Innerhalb dieses Ausschnitts wurde jeweils der Mittelwert der Tiefenwerte berechnet und anschließend die Differenz ermittelt, die bei 10.12 mm lag. Der Temperaturunterschied zwischen den beiden Aufnahmen betrug 5.25°C . Der Quotient dieser beiden Werte ergibt eine Veränderung von 1.93 mm pro 1°C und ist somit mit dem in Abschnitt 5.4.3.8 ermittelten Wert vergleichbar, der bei 1.99 mm pro 1°C lag.

5.5.2 Auswirkung auf die MDP

Nach der beschriebenen Voruntersuchung der Tiefenbilder, die eine deutliche temperaturbedingte Änderung aufwiesen, wurde der Temperatureinfluss auf das MDP basierte

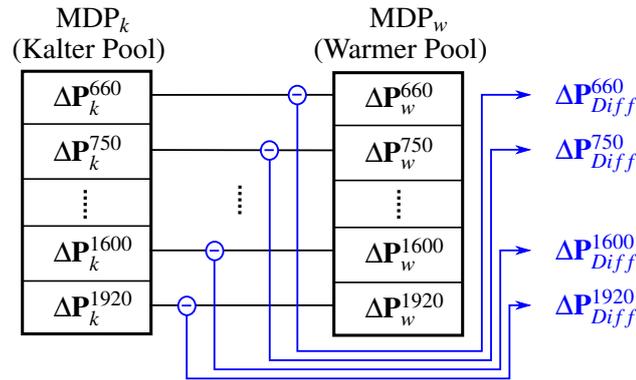


Abbildung 5.45: Darstellung des Verfahrens für die Berechnung der Differenz-Delta-Planes ΔP_{Diff}^d , die im Rahmen der Überprüfung des MDP basierten Verfahrens auf Temperaturabhängigkeit ausgewertet werden.

Verfahren getestet. Dazu sind zwei Pools an Delta-Planes erzeugt worden, wobei die Tiefenbilder, die für die Erzeugung der Delta-Planes für den ersten Pool (kalter Pool) verwendet wurden, bei einer gemessenen Temperatur von 24.00°C , und für den zweiten Pool (warmer Pool) bei 31.18°C aufgezeichnet worden sind. Jeder Pool bestand aus sieben Delta-Planes, die für die Distanzen 660, 750, 860, 1020, 1250, 1600 und 1920 mm erzeugt wurden (zwischen den aufgezählten Distanzen, mit Ausnahme der letzten Distanz, liegen jeweils etwa 60 der ermittelten Distanzquantisierungsschritte, siehe Abschnitt 5.2.3.4).

Um zu überprüfen, ob die Temperatur einen Einfluss auf das MDP basierte Verfahren haben kann, wurde jede Delta-Plane aus dem kalten Pool mit der bzgl. der Distanz entsprechenden Delta-Plane aus dem warmen Pool verglichen, siehe Abb. 5.45. Unterscheiden sich die Delta-Planes deutlich, so ist dies ein notwendiges, jedoch nicht zwangsläufig hinreichendes Kriterium dafür, dass die Temperatur für diesen Unterschied verantwortlich ist (es könnte auch weitere, unbekannte Ursachen geben). Sind die Delta-Planes hingegen vergleichbar, so kann eine Temperaturabhängigkeit ausgeschlossen werden.

Um die Ähnlichkeiten zweier Delta-Planes ΔP_w^d und ΔP_k^d zu beurteilen, wurde die Differenz zwischen den jeweiligen Einträgen berechnet:

$$\Delta P_{Diff}^d(u, v) = \Delta P_w^d(u, v) - \Delta P_k^d(u, v). \quad (5.35)$$

Dabei deuten w bzw. k die Herkunft einer Delta-Plane an (warmer bzw. kalter Pool) und d gibt die Distanz an. ΔP_{Diff}^d wird im Folgenden auch Differenz-Delta-Plane (DDP) genannt. Sowohl die Visualisierung der DDPs als auch die dazugehörigen Histogramme sind exemplarisch für drei der sieben genannten Distanzen in Abb. 5.46 dargestellt. In Abb. 5.46(g) ist sowohl der Verlauf der Mittelwerte als auch der der Standardabweichungen der Differenzwerte in den DDPs für alle sieben Distanzen angegeben. Die Standardabweichung liegt bei allen Distanzen unterhalb von 0.36 Quantisierungsstufen und der Mittelwert im Intervall $[-2.40 \cdot 10^{-3}, 0.76 \cdot 10^{-3}]$. In den Abb. 5.46(a) bis 5.46(c) ist keine erkennbare Struktur der Verteilung zu sehen. Es ist lediglich ein Rauschmuster festzustellen. Die Histogramme verdeutlichen, dass sich die in den DDPs vorkommenden Werte deutlich um den Nullpunkt häufen. Da die in

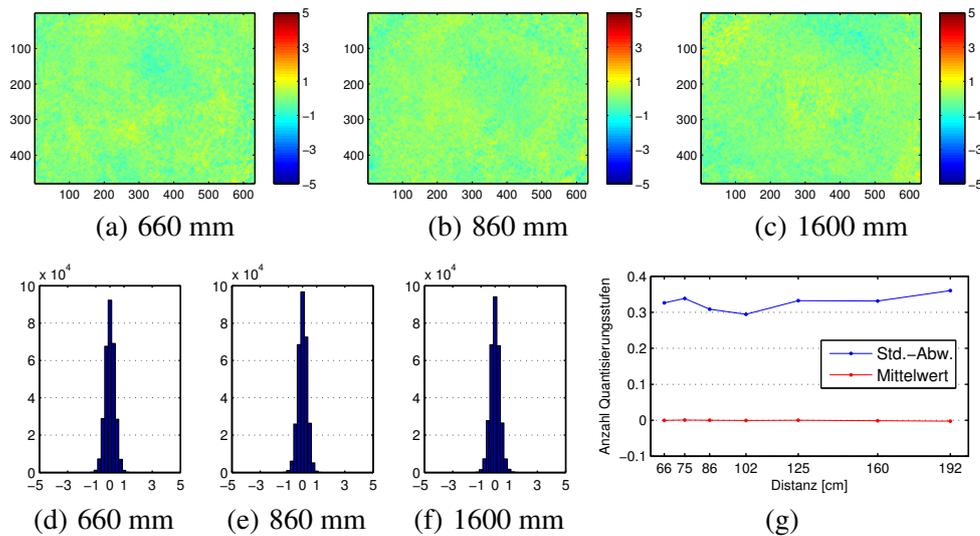


Abbildung 5.46: Visualisierung der Werte der Differenz-Delta-Planes (DDPs) für unterschiedliche Distanzen. Die Farbskala gibt die Anzahl der abweichenden Distanzquantisierungsstufen an. Die dargestellten Differenzwerte in (a) bis (c) lassen im Wesentlichen nur ein Rauschmuster erkennen. In (d) bis (f) sind die dazugehörigen Histogramme zu sehen. Die Standardabweichung und der Mittelwert der Werte der DDPs für die Distanzen zwischen 660 und 1920 mm sind in (g) dargestellt (aus Platzgründen ist die X-Achse in cm angegeben). Die gezeigten Auswertungen deuten auf sehr geringe Differenzen und folglich auf sehr ähnliche Delta-Planes hin.

den Delta-Planes gespeicherten Werte als Anzahl der Quantisierungsstufen angegeben sind (vgl. Abschnitt 5.2.3.5.1), gilt dies auch für die Werte in den DDPs. 99.99% aller Werte in den DDPs in den Abb. 5.46(a) bis 5.46(c) liegen im Intervall $[-0.80, 0.96]$ und befinden sich somit innerhalb von ± 1 Quantisierungsstufen. Da die Kinect ein gewisses Rauschen besitzt, und dieses sich in den vom OpenNI-Framework gelieferten Tiefenbildern nur in ganzen Tiefenquantisierungsstufen manifestieren kann, ist die Schlussfolgerung möglich, dass sich die beobachteten Differenzen in den DDPs innerhalb des Rauschniveaus der Kinect bewegen, und die Delta-Planes als von der Temperatur unbeeinflusst betrachtet werden können.

Das bedeutet, dass sich zwar die Tiefenwerte in den Tiefenbildern durch die Temperatur verändern, die Struktur des komplexen ortsabhängigen Fehlers jedoch erhalten bleibt, so dass auch die Delta-Planes ihre Gültigkeit bei Temperaturänderungen nicht verlieren. Dies hat zweierlei Konsequenzen. Zum einen muss bei der Anwendung der in Abschnitt 5.2.3.6 beschriebenen MDP basierten Korrektur die Temperatur nicht beachtet werden. Zum anderen spielt die konkrete Temperatur bei der Aufzeichnung der Tiefenbilder, aus denen die Delta-Planes für die MDP gewonnen werden, keine Rolle. Die genannten Konsequenzen verdeutlichen die Stabilität und die Praktikabilität des vorgeschlagenen MDP basierten Korrekturverfahrens.

6

Zusammenfassung

Der erste Teil dieser Dissertation war einem neuen Verfahren zur Linsenverzerrungskorrektur gewidmet. Zunächst wurde ein parameterabhängiges Korrekturmodell für die Korrektur der Linsenverzerrung gewählt und anschließend eine Zielfunktion definiert, anhand der die Entzerrungsparameter optimiert wurden. Dabei sind sowohl das Korrekturmodell als auch die Zielfunktion unabhängig vom Lochkameramodell und somit von den extrinsischen und intrinsischen Parametern. Die vorgeschlagene Zielfunktion basiert auf der Annahme, dass gerade Linien in der Szene auf gerade Linien im Kamerabild projiziert werden, und dass eine Verletzung dieser Eigenschaft auf eine vorhandene Linsenverzerrung zurückzuführen ist. Zur Erfassung der Linienstrukturen wurden einfache schachbrettartige Kalibrierobjekte eingesetzt. Der aus dem Kontext der Objektverfolgung (Tracking) bekannte Condensation-Algorithmus wurde derart angepasst, dass er als stochastischer partikelfilterbasierter Optimierer zur Berechnung der optimalen Entzerrungsparameter dient. Nach einer umfassenden Untersuchung der Kontrollparameter des Condensation-Algorithmus wurde die Korrekturleistung des vorgeschlagenen Verfahrens mit existierenden Verfahren verglichen, wobei insbesondere bei starker Linsenverzerrung und zunehmendem Rauschen mit dem neuen Ansatz deutlich bessere Ergebnisse erzielt werden konnten. Um einen direkten Vergleich der Optimierer unter Verwendung des vorgeschlagenen Verfahrens zu ermöglichen, wurde der stochastische partikelfilterbasierte Optimierer gegen den deterministischen Levenberg-Marquardt-Algorithmus ausgetauscht. Mit Ausnahme eines untersuchten Spezialfalls ist das vorgeschlagene stochastische Verfahren überlegen.

Der zweite Teil der Dissertation lieferte Beiträge im Bereich der 3D-Tiefensensoren, die anhand des Kinect-Tiefensensors erarbeitet wurden, jedoch auch auf andere Tiefensensoren übertragbar sind. Zunächst wurde auf die Messfehlerbeschaffenheit in den Tiefenbildern der Kinect eingegangen, die sowohl vom Sensorort (Position im Tiefenbild) als auch von der Messdistanz abhängig war. Es wurden zwei auf planaren Oberflächen basierende Korrekturansätze (Delta-Plane und MDP) vorgestellt, die auf einem heuristischen Korrekturmodell basieren und durch eine tabellarische Beschreibung effizient eingesetzt werden können. Die Korrekturleistung wurde anhand unterschiedlicher Testszenarien sowohl quantitativ als auch qualitativ beurteilt und demonstriert. Einen weiteren Beitrag stellte ein homographiebasiertes Verfahren zur Online-Referenzmessdatengewinnung dar, welches mit einfachen Kalibrierobjekten sowie der bereits in der Kinect integrierten Hardware vollständig durchführbar ist. Dieses Verfahren ermöglicht eine kostengünstige und interaktive Aufdeckung und Beurteilung von Messfehlern während des laufenden Betriebs. Ein weiterer Teil der Dissertation widmete sich der Aufdeckung unterschiedlicher umweltbedingter Auswirkungen auf den Tiefensensor sowie auf die internen Kameras der Kinect, wobei insbesondere der

Temperatur ein wesentlicher Einfluss nachgewiesen werden konnte. Als Konsequenz aus den Beobachtungen und Experimenten ist eine Reihe von praktischen Regeln abgeleitet worden, die zur Fehlervermeidung bzw. Fehlerreduktion während des Kalibrier- und Messvorgangs der Kinect beitragen. Zuletzt wurde ein experimenteller Nachweis der Robustheit des MDP basierten Korrekturverfahrens gegenüber Umwelteinflüssen erbracht.

Literaturverzeichnis

- [ABD⁺90] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof, and D. Sorensen. LAPACK: a portable linear algebra library for high-performance computers. In *Proceedings of the ACM/IEEE conference on Supercomputing*, Supercomputing '90, pages 2–11, Los Alamitos, CA, USA, IEEE Computer Society Press, 1990.
- [AF01] M. Ahmed and A. Farag. Non-metric calibration of camera lens distortion. In *Proceedings of the International Conference on Image Processing*, volume 2 of *ICIP*, pages 157–160. IEEE Computer Society, October 2001.
- [ALL⁺07] H. H. Austmann, J. Langner, W. Laveure, E. Hering, K. Bressler, J. Strauß, J. Gutekunst, and W. Streib. *Elektronik für Ingenieure und Naturwissenschaftler*. Springer-Lehrbuch. Springer London, Limited, 2007.
- [BHK⁺10] A. Boukercha, K. Hölscher, B. Korth, J. Köster, P. Libuschewski, T. Opfermann, K. Rösner, L. Rothacker, D. Siedhoff, and F. Vielhauer. Fly-Cam - Oberflächenerfassung mit ferngesteuerten, kamerabestückten Flugobjekten. Endbericht PG 531, Projektgruppenleitung: D. Fiedler, T. Wiederkehr, Herausgeber: Lehrstuhl VII, Graphische Systeme, Fakultät für Informatik, TU Dortmund, 2010.
- [BK08] G. Bradski and A. Kaehler. *Learning OpenCV*. O'Reilly Media Inc., first edition, 2008.
- [BKS09] S. M. Bower, J. Kou, and J. R. Saylor. A method for the temperature calibration of an infrared camera using water as a radiative source. *Review of Scientific Instruments*, 80(9):095107, 2009.
- [Bla04] F. Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1):231–243, 2004.
- [Bou10] J. Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/ [Zugegriffen am 09.07.2013], 2010.

- [Bra00] G. Bradski. The OpenCV Library, Referenzversion 2.1. *Dr. Dobb's Journal of Software Tools*, 2000.
- [BRB⁺11] K. Berger, K. Ruhl, C. Brümmer, Y. Schröder, A. Scholz, and M. Magnor. Markerless motion capture using multiple color-depth sensors. In *Proceedings of Vision, Modeling and Visualization (VMV)*, pages 317–324. Eurographics Association, October 2011.
- [Bro66] D. C. Brown. Decentering distortion of lenses. *Photometric Engineering*, 32(3):444–462, 1966.
- [Bro71] D. C. Brown. Close-range camera calibration. *Photometric Engineering*, 37(8):855–866, 1971.
- [Bro06] C. Brockmann. *Remote Vision Based Multi Gesture Interaction in Natural Indoor Environments*. PhD thesis, TU Dortmund, Lehrstuhl Informatik VII, 2006.
- [BS08] I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik*. Harri Deutsch Verlag, 2008.
- [BSL⁺11] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, March 2011.
- [BWS05] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, February 2005.
- [CBDM09] J. Civera, D. R. Bueno, A. J. Davison, and J. M. M. Montiel. Camera self-calibration for sequential bayesian structure from motion. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'09*, pages 3411–3416, Piscataway, NJ, USA., IEEE Press, 2009.
- [CBF11] W.-C. Chiu, U. Blanke, and M. Fritz. Improving the kinect by cross-modal stereo. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 116.1–116.10. BMVA Press, September 2011.
- [DBH02] P. Deuffhard, F. A. Bornemann, and A. Hohmann. *Numerische Mathematik*. Number v. 1 in De Gruyter Lehrbuch. De Gruyter, 2002.
- [DdFG01] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Statistics for engineering and information science. Springer, 2001.
- [DF01] F. Devernay and O. Faugeras. Straight lines have to be straight: automatic calibration and removal of distortion from scenes of structured environments. *Machine Vision and Applications*, 13(1):14–24, August 2001.

-
- [Die06] J. Diebel. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. Technical report, Stanford University, 2006.
- [DJ11] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. In *The Oxford Handbook of Nonlinear Filtering*, pages 656–704. Oxford University Press, 2011.
- [DP07] D. Desjardins and P. Payeur. Dense stereo range sensing with marching pseudo-random patterns. In *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision, CRV '07*, pages 216–226, Washington, DC, USA, IEEE Computer Society, 2007.
- [EMF03] M. T. El-Melegy and A. A. Farag. Nonmetric lens distortion calibration: Closed-form solutions, robust estimation and model selection. In *Proceedings of the 9th IEEE International Conference on Computer Vision - Volume 2, ICCV '03*, pages 554–559, Washington, DC, USA, IEEE Computer Society, 2003.
- [FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.
- [FB86] J. G. Fryer and D. C. Brown. Lens distortion for close-range photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 52(1):51–58, 1986.
- [Fel92] W.D. Fellner. *Computergrafik*. Reihe Informatik. BI-Wissenschaftsverlag, 1992.
- [FGG⁺13] A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige. *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*. Advances in Computer Vision and Pattern Recognition. Springer, 2013.
- [FM13] D. Fiedler and H. Müller. Impact of thermal and environmental conditions on the kinect sensor. In X. Jiang et al., editor, *Advances in Depth Image Analysis and Applications*, volume 7854 of *Lecture Notes in Computer Science*, pages 21–31. Springer, 2013.
- [FS11] S. Fernandez and J. Salvi. Planar-based camera-projector calibration. In *Proceedings of the 7th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 633–638, Dubrovnik, IEEE Explore Digital Library, September 4-6 2011.
- [Gen11] J. Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011.
- [Geo07] H. O. Georgii. *Stochastik: Einführung in die Wahrscheinlichkeitstheorie und Statistik*. De Gruyter Lehrbuch. De Gruyter, 2007.
-

- [GFMP08] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, CVPR 2008*, Anchorage, Alaska, USA, IEEE Computer Society, June 24-26 2008.
- [GHM04] J. E. Gentle, W. Härdle, and Y. Mori. *Handbook of computational statistics: concepts and methods*. Springer handbooks of computational statistics. Springer-Verlag Berlin Heidelberg, 2004.
- [GTA⁺12] M. Goodrum, M. Trotter, A. Aksel, S. Acton, and K. Skadron. Parallelization of particle filter algorithms. In *Proceedings of the International Conference on Computer Architecture (ISCA)*, pages 139–149, Saint-Malo, France, Springer, 2012.
- [GW01] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.
- [GYB04] S. B. Gokturk, H. Yalcin, and C. Bamji. A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, volume 3 of *CVPRW '04*, Washington, DC, USA, IEEE Computer Society, 2004.
- [Har99] R. I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2):115–127, November 1999.
- [Hec94] P. S. Heckbert. *Graphics Gems Four*. The Graphics Gems Series. AP Professional, 1994.
- [Hec05] E. Hecht. *Optik*. Oldenbourg, 2005.
- [Hei00] J. Heikkilä. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, October 2000.
- [HKH11] C. D. Herrera, J. Kannala, and J. Heikkilä. Accurate and practical calibration of a depth and color camera pair. In *Proceedings of the 14th international conference on Computer analysis of images and patterns - Volume Part II, CAIP'11*, pages 437–445, Berlin, Heidelberg, Springer-Verlag, 2011.
- [HKH12] D. Herrera, J. Kannala, and J. Heikkilä. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2058–2064, 2012.
- [Hot58] J. Hothmer. Possibilities and limitations for elimination of distortion in aerial photographs. *The Photogrammetric Record*, 2(12):426–445, 1958.

-
- [HS97] J. Heikkilä and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, CVPR '97*, pages 1106–1112, Washington, DC, USA, IEEE Computer Society, 1997.
- [HZ03] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, second edition, 2003.
- [IB98] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.
- [ICR13] Workshop Kinect untangled: from basics to applications, held in conjunction with the International Conference on Recent Advances in Neurorehabilitation (ICRAN 2013). Polytechnic Innovation City, Valencia (Spain), <http://icranconference.com/> [Zugegriffen am 05.08.2013], March 6th 2013.
- [KBKL10] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight cameras in computer graphics. *Computer Graphics Forum*, 29(1):141–159, 2010.
- [KE12] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.
- [KM12] K. Konolige and P. Mihelich. Technical description of kinect calibration. http://www.ros.org/wiki/kinect_calibration/technical/ [Zugegriffen am 31.07.2013], 2012.
- [Koh04] W. Kohn. *Statistik: Datenanalyse und Wahrscheinlichkeitsrechnung*. Statistik und ihre Anwendungen. Springer, 2004.
- [KRI06] T. Kahlmann, F. Remondino, and H. Ingensand. Calibration for increased accuracy of the range imaging camera SwissRanger. In *Proceedings of the ISPRS Commission V Symposium on Image Engineering and Vision Metrology, International Society for Photogrammetry and Remote Sensing (ISPRS)*, volume 36 part 5, Dresden, Germany, September 25-27 2006.
- [Kue11] T. Kuehn. The kinect sensor platform. In *Advances in Media Technology*, number 2, pages 1–4. Technische Universität München, 2011.
- [Lan00] R. Lange. *3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology*. PhD thesis, University of Siegen, June 2000.
- [Len02] B. Lenk. *Handbuch der automatischen Identifikation 2. 2D-Codes: Matrixcodes, Staplecodes, Composite Codes, Dotcodes*. Handbuch der automatischen Identifikation. Lenk Monika Fachbuchverlag, 2002.
-

- [LJL07] K. H. Lee, C. Je, and S. W. Lee. Color-stripe structured light robust to surface color and discontinuity. In *Proceedings of the Asian Conference on Computer Vision, ACCV (2)*, volume 4844 of *Lecture Notes in Computer Science*, pages 507–516. Springer, 2007.
- [LK07] M. Lindner and A. Kolb. Calibration of the intensity-related distance error of the PMD ToF-camera. In *Proceedings of the International Society for Optics and Photonics (SPIE)*, volume 6764, September 10th 2007.
- [Lou04] M. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. <http://www.ics.forth.gr/~lourakis/levmar/> [Zugegriffen am 20.05.2013], July 2004.
- [Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.
- [LSBR99] R. Lange, P. Seitz, A. Biber, and Schwarte R. Time-of-flight range imaging with a custom solid state image sensor. In *Proceedings of the International Society for Optics and Photonics (SPIE), Laser Metrology and Inspection*, volume 3823, pages 180–191, September 9th 1999.
- [LSKK10] M. Lindner, I. Schiller, A. Kolb, and R. Koch. Time-of-flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding*, 114(12):1318–1328, December 2010.
- [Luh10] T. Luhmann. *Nahbereichsphotogrammetrie: Grundlagen, Methoden und Anwendungen*. Wichmann Herbert, 2010.
- [Mar10] A. Marbs. Laserscanning zur dreidimensionalen Objekterfassung: Grundlagen und Anwendungen. Technical report, i3mainz - Institut für Raumbezogene Informations- und Messtechnik, FH Mainz, 2010.
- [MMB04] C. McGlone, E. Mikhail, and J. Bethel. *Manual of Photogrammetry*. American Society for Photogrammetry and Remote Sensing, 5th edition, 2004.
- [Mor77] J. J. Moré. The Levenberg-Marquardt Algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, volume 630, pages 105–116. Springer, Berlin, 1977.
- [MT96] R. Mohr and B. Triggs. Projective geometry for image analysis. In *Proceedings of the 18th International Symposium on Photogrammetry & Remote Sensing (ISPRS '96)*, Vienna, Autriche, 1996.
- [NLB⁺11] D. Neumann, F. Lugauer, S. Bauer, J. Wasza, and J. Hornegger. Real-time RGB-D mapping and 3-D modeling on the GPU using the random ball cover data structure. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1161–1167. IEEE, November 6-13 2011.

-
- [Ope10] OpenNI. Open Natural Interaction. The standard framework for 3D sensing. <http://openni.org/> [Zugegriffen am 29.05.2013], 2010.
- [PD09] P. Payeur and D. Desjardins. Structured light stereoscopic imaging with dynamic pseudo-random patterns. In *Proceedings of the 6th International Conference on Image Analysis and Recognition, ICIAR '09*, pages 687–696, Berlin, Heidelberg, Springer-Verlag, 2009.
- [Per12] Workshop on kinect in pervasive computing, held in conjunction with pervasive 2012. Newcastle, UK, <http://pervasiveconference.org/2012/> [Zugegriffen am 05.08.2013], June 18th 2012.
- [PP08] P. Podbreznik and B. Potocnik. Influence of temperature variations on calibrated cameras. *WASET World Academy of Science, Engineering and Technology*, 43:55–61, July 2008.
- [PR12] D. Piatti and F. Rinaudo. SR-4000 and CamCube3.0 Time of Flight (ToF) Cameras: Tests and Comparison. *Remote Sensing*, 4(4):1069–1089, 2012.
- [QC02] G. Qian and R. Chellappa. Bayesian self-calibration of a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 2351 of *Lecture Notes in Computer Science*, pages 277–293. Springer, 2002.
- [Ric11] J. Richarz. *Videobasierte Gestenerkennung in einer intelligenten Umgebung*. PhD thesis, Technische Universität Dortmund, 2011.
- [SBK08] I. Schiller, C. Beder, and R. Koch. Calibration of a pmd-camera using a planar calibration pattern together with a multi-camera setup. In *Proceedings of the 21th International Society for Photogrammetry and Remote Sensing Congress (ISPRS)*, 2008.
- [SC10] M. J. Smith and E. Cope. The effects of temperature variation on single-lens-reflex digital camera calibration parameters. In *Proceedings of the ISPRS Commission V Mid-Term Symposium on Close Range Image Measurement Techniques, International Society for Photogrammetry and Remote Sensing (ISPRS)*, volume 38 part 5, pages 554–559, Newcastle, UK, June 21-24 2010.
- [Sca05] D. Scaramuzza. OCamCalib: Omnidirectional Camera Calibration Toolbox for Matlab. <https://sites.google.com/site/scarobotix/ocamcalib-toolbox/> [Zugegriffen am 21.05.2013], 2005.
- [Sch07] O. Schreer. *Stereoanalyse und Bildsynthese*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
-

- [SdMFCV02] A. M. Sa, E. S. de Medeiros Filho, P. C. Carvalho, and L. Velho. Coded structured light for 3d-photography: An overview. *RITA*, 9(2):203–219, 2002.
- [SH11] H. Su and B. He. Stereo rectification of calibrated image pairs based on geometric transformation. *International Journal of Modern Education and Computer Science (IJMECS)*, 3(4):17–24, July 2011.
- [SHBS11] J. Stowers, M. Hayes, and A. Bainbridge-Smith. Quadrotor helicopter flight control using hough transform and depth map from a microsoft kinect sensor. In *Proceedings of the 12th IAPR Conference on Machine Vision Applications (MVA2011)*, Nara, JAPAN, pages 352–356, June 13-15 2011.
- [SJP11] J. Smisek, M. Jancosek, and T. Pajdla. 3D with kinect. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1154–1160. IEEE, 2011.
- [Sla80] C. Slama. *Manual of Photogrammetry, Fourth Edition*. American Society for Photogrammetry and Remote Sen, 4th edition, June 1980.
- [SMS06] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easy calibrating omnidirectional cameras. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2006.
- [SPSL10] A. Shpunt, B. Pesach, and PRIME SENSE LTD. Optical pattern projektor (patent). US Patent, Pub. No.: US 2010/0284082 A1, Pub. Date: November 11th, 2010, Assignee: PrimeSense LTD, Tel Aviv (IL), 2010.
- [SS07] D. Scaramuzza and R. Siegwart. *A Practical Toolbox for Calibrating Omnidirectional Cameras*. Swiss Federal Institute of Technology, 2007.
- [SS11] E. Stone and M. Skubic. Evaluation of an inexpensive depth camera for in-home gait assessment. *Journal of Ambient Intelligence and Smart Environments*, 3(4):349–361, December 2011.
- [Sze06] R. Szeliski. Image alignment and stitching: a tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104, January 2006.
- [TBF05] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*. Intelligent robotics and autonomous agents. The MIT Press, August 2005.
- [Tsa85] R.Y. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1985.
- [Tuk77] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

-
- [TWF⁺11] C. Timm, F. Weichert, D. Fiedler, C. Prasse, H. Müller, M. ten Hompel, and P. Marwedel. Decentralized control of a material flow system enabled by an embedded computer vision system. In *Proceedings of the IEEE International Conference on Communications, Workshop on Embedding the Real World into the Future Internet, ICC'11*, pages 1–5, Kyoto, Japan, June 2011.
- [WA12] J. Webb and J. Ashley. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress, Berkely, CA, USA, 1st edition, 2012.
- [Wah84] F. M. Wahl. *Digitale Bildsignalverarbeitung: Grundlagen, Verfahren, Beispiele*. Nachrichtentechnik (Springer-Verlag). Springer, 1984.
- [WBRF13] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013.
- [WCH92] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, October 1992.
- [WDI12] International workshop on depth image analysis (WDIA 2012), held in conjunction with 21st international conference on pattern recognition (ICPR 2012). Tsukuba International Congress Center, Tsukuba Science City, Japan, <http://www.icpr2012.org/index.html> [Zugegriffen am 05.08.2013], November 10th 2012.
- [WFH⁺10] F. Weichert, D. Fiedler, J. Hegenberg, H. Müller, C. Prasse, M. Roidl, and M. ten Hompel. Marker-based tracking in support of RFID controlled material flow systems. *Logistics Research*, 2(1):13–21, June 2010.
- [WT08] C.-H. Wu and S.-C. Tai. *De-interlacing Algorithm Using Motion Adaptive and Edge-based Analysis*. VDM Verlag, Germany, 2008.
- [WWJ11] M. Weinmann, S. Wursthorn, and B. Jutzi. Semi-automatic image-based co-registration of range imaging data with different characteristics. In U. Stilla, F. Rottensteiner, H. Mayer, B. Jutzi, and M. Butenuth, editors, *Photogrammetric Image Analysis PIA11. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38 (3/W22), pages 119–124, 2011.
- [WWS07] H. Wuest, F. Wientapper, and D. Stricker. Adaptable model-based tracking using analysis-by-synthesis techniques. In W. G. Kropatsch, M. Kampel, and A. Hanbury, editors, *CAIP*, volume 4673 of *Lecture Notes in Computer Science*, pages 20–27. Springer, 2007.
- [YWY⁺11] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3D pose estimation from a single depth image. In *Proceedings of the International Conference on Computer Vision, ICCV '11*, pages 731–738, Washington, DC, USA, IEEE Computer Society, 2011.
-

- [Zha98] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 1998.
- [ZSMSL07] Z. Zalevski, A. Shpunt, A. Maizels, and PRIME SENSE LTD. Method and system for object reconstruction (patent). World Intellectual Property Organisation, Int. Pub. No.: WO 2007/043036 A1, Pub. Date: April 19th, 2007, Applicant: PrimeSense LTD, Tel Aviv (IL), 2007.
- [ZZ11] C. Zhang and Z. Zhang. Calibration between depth and color sensors for commodity depth cameras. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '11*, pages 1–6, Washington, DC, USA, IEEE Computer Society, 2011.