



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

## **Masterarbeit**

Carsten Fries

Kamerabasierte Identifizierung und Lokalisierung von Gegenständen  
für flexible Roboter

Carsten Fries

Carsten.Fries@haw-hamburg.de

Kamerabasierte Identifizierung und Lokalisierung von Gegenständen  
für flexible Roboter

Masterarbeit eingereicht im Rahmen der Masterprüfung  
im Studiengang Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Andreas Meisel  
Zweitgutachter: Prof. Dr. rer. nat. Stephan Pareigis

Abgegeben am 12. August 2011

**Carsten Fries**

**Thema der Masterarbeit**

Kamerabasierte Identifizierung und Lokalisierung von Gegenständen für flexible Roboter

**Stichworte**

3D-Rekonstruktion, Identifikation, Lokalisierung, Objekterkennung, SURF, SIFT, Skalierungsinvariante Merkmale, Merkmalsextraktion, Greifarm

**Kurzzusammenfassung**

In dieser Arbeit wird eine bildbasierte Erkennung und Lokalisierung von Gegenständen vorgestellt. Einmalig muss dafür eine 3D-Oberflächenrekonstruktion für jeden Gegenstand erfolgen, wobei jedem Oberflächenpunkt gegenstands-spezifische Merkmale zugeordnet werden. Die Rekonstruktion basiert auf perspektivisch unterschiedlichen Kameraaufnahmen. In den Aufnahmen werden Merkmale gesucht, welche in anderen Aufnahmen zuverlässig wiedererkannt werden können. Dies ermöglicht die Suche nach Merkmalskorrespondenzen zwischen Bildern, wodurch die 3D-Rekonstruktion erfolgt. Auf Basis des resultierenden 3D-Merkmalmodells (Oberflächenpunkte + Merkmale) kann ein Gegenstand in einem beliebigen Bild identifiziert und seine räumliche Lage und Orientierung bestimmt werden. Des Weiteren kommt ein Knickarm-Roboter für Mobilisierungstätigkeiten zum Einsatz. Mit dem Wissen über die räumliche Lage und Orientierung sind dadurch Greif- und Bewegungsvorgänge an Gegenständen durchführbar.

**Carsten Fries**

**Title of the paper**

Camera-based identification and localization of objects for flexible robots

**Keywords**

3d reconstruction, identification, localization, object detection, SURF, SIFT, scale-invariant features, feature extraction, manipulator-arm

**Abstract**

This thesis presents a camera-based identification and localization of objects. In this regard, a unique 3d surface reconstruction is performed for each object, in which object-specific features are assigned to each surface point. The reconstruction is based on different perspective camera views. In these images, features are extracted which can be reliably detected in other views. This allows finding of feature correspondences between images, thereby enabling the 3d reconstruction. The so-called 3d feature model (surface points + features) is used to identify an object in an arbitrary image and to determine its spatial position and orientation. Furthermore, an articulated robot performs mobilization activities. With the known spatial pose, gripping and moving operations can be performed on objects.

---

„Freude an der Arbeit lässt das Werk trefflich geraten.“

*(Aristoteles, griechischer Philosoph, † 322 v. Chr.)*

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Stand der Technik . . . . .	1
1.2	Zielsetzung . . . . .	2
1.3	Gliederung . . . . .	4
<b>2</b>	<b>Vergleichbare Arbeiten</b>	<b>5</b>
2.1	Sensorbasierte Methoden . . . . .	5
2.2	Bildbasierte Methoden . . . . .	5
2.2.1	Markerbasierte Verfahren . . . . .	6
2.2.2	Markerlose Verfahren . . . . .	7
<b>3</b>	<b>Grundlagen</b>	<b>11</b>
3.1	Koordinatensysteme . . . . .	11
3.1.1	Kamerakoordinatensystem . . . . .	11
3.1.2	Bildkoordinatensystem . . . . .	12
3.1.3	Gelenkkoordinatensysteme vom Roboter . . . . .	12
3.1.4	Weltkoordinatensystem . . . . .	12
3.1.5	Endeffektorkoordinatensystem am Greifer . . . . .	13
3.1.6	Objektkoordinatensystem . . . . .	13
3.2	Singulärwertzerlegung . . . . .	13
3.3	Fundamentalmatrix . . . . .	14
3.3.1	Die Fundamentalmatrix F . . . . .	14
3.3.2	Berechnung der Fundamentalmatrix . . . . .	15
3.3.3	Robuste Schätzung der Fundamentalmatrix . . . . .	17
<b>4</b>	<b>Technischer Aufbau</b>	<b>19</b>
4.1	Roboterarm . . . . .	19
4.2	Steuerungskontroller . . . . .	22
4.3	Computer . . . . .	23
4.4	Kameras . . . . .	23
4.5	Drehmesstisch . . . . .	25
<b>5</b>	<b>Generierung eines 3D-Merkmalmodells</b>	<b>26</b>
5.1	Kamerakalibrierung . . . . .	26
5.1.1	Intrinsische Kameraparameter . . . . .	27
5.1.2	Ermittlung der intrinsischen Kameraparameter . . . . .	30
5.1.3	Kalibrierergebnisse . . . . .	31
5.2	Merkmalsextraktion . . . . .	33
5.2.1	SIFT . . . . .	34
5.2.2	SURF . . . . .	41
5.3	Initiale Bestimmung von Weltpunkten . . . . .	42
5.3.1	Suche nach Merkmalskorrespondenzen . . . . .	43
5.3.2	Filterung . . . . .	47
5.3.3	Berechnung der 3D-Weltpunkte . . . . .	51
5.3.4	Anlegen eines 3D-Merkmalmodells . . . . .	62
5.4	Weltpunkte aus weiteren Aufnahmen . . . . .	63
5.4.1	Suche nach Merkmalskorrespondenzen . . . . .	65

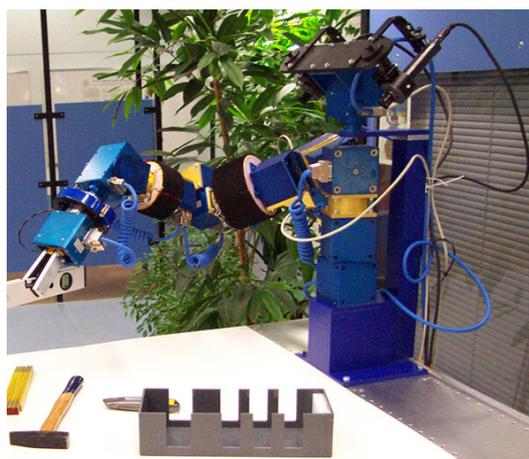
5.4.2	Filterung	66
5.4.3	Berechnung der 3D-Weltpunkte	66
5.4.4	Erweiterung des 3D-Merkmalmodells	69
<b>6</b>	<b>Posenbestimmung</b>	<b>70</b>
6.1	3D-RANSAC	71
6.2	Normalisierte Direct Linear Transformation	72
6.3	Verbessertes nicht-iteratives Verfahren	75
<b>7</b>	<b>Objektmobilisierung mit einem Greifarm</b>	<b>88</b>
7.1	Vorbedingungen	88
7.2	Startposition anfahren	89
7.3	Identifizierung	90
7.4	Berechnung der Zielpose abhängig zur Objektpose	91
7.5	Korrektur der Zielpose	100
7.6	Inverse Kinematik der Zielpose	100
7.7	Berechnung der Greifpose abhängig zur Zielpose	106
7.8	Inverse Kinematik der Greifpose	107
7.9	Mobilisierung	107
<b>8</b>	<b>Ergebnisse</b>	<b>109</b>
8.1	3D-Merkmalmodelle	109
8.1.1	Testobjekt	109
8.1.2	Papierschachtel	116
8.1.3	Tee-Packung	119
8.1.4	VHS-Kassette	121
8.2	Identifikation von Gegenständen	122
8.3	Mobilisierung von Gegenständen	123
<b>9</b>	<b>Zusammenfassung und Ausblick</b>	<b>128</b>
9.1	Stand der Arbeit	128
9.2	Weiterentwicklungen	129
<b>A</b>	<b>Codebeispiele</b>	<b>130</b>
<b>B</b>	<b>Inhalt der beiliegenden DVD</b>	<b>133</b>
	<b>Abbildungsverzeichnis</b>	<b>134</b>
	<b>Tabellenverzeichnis</b>	<b>137</b>
	<b>Algorithmenverzeichnis</b>	<b>138</b>
	<b>Abkürzungsverzeichnis</b>	<b>139</b>
	<b>Formelzeichen</b>	<b>140</b>
	<b>Index</b>	<b>143</b>
	<b>Quellenverzeichnis</b>	<b>146</b>

# 1 Einleitung

## 1.1 Stand der Technik

Die bis heute rasante technologische Entwicklung führte zu einer zunehmenden Anzahl an existierenden Assistenzrobotern. Sie kommen für unterstützende Tätigkeiten wie Handlings-, Pick & Place-, Manipulations-, Prüf- und Nachbearbeitungsaufgaben zum Einsatz. Dabei werden die Tätigkeiten ortsfest, portabel oder mobil und gegebenenfalls autonom durchgeführt. Portable und mobile Roboter sind nicht auf einen Aufstellungsort beschränkt, welches einen größeren Einsatzort mit sich bringt. Roboter bieten im Vergleich zu Menschen konstante Fähigkeiten im Bereich der Präzision, Geschwindigkeit, Kraft und Ausdauer. In der Industrie ist dabei die Kostenreduktion bei gleichzeitiger Produktivitätssteigerung ein wichtiges Kriterium für den ansteigenden Einsatz von Assistenzrobotern [HM05]. Immer verbreiteter wird der Fokus auf autonome sensorbasierte Roboter gelegt. Sie können deutlich mehr als nur fest einprogrammierte Handlungsabläufe auszuführen. Messbare physikalische und chemische Eigenschaften wie Wärmestrahlung, Temperatur, Feuchtigkeit, Druck, Schall, Helligkeit oder Beschleunigung werden über Sensoren erfasst, bewertet und führen zu einer den Umwelteinflüssen angemessenen Reaktion [SHW04].

Häusliche Assistenzroboter finden heutzutage vermehrt Gebrauch. Ein Haushaltsroboter benötigt für seine einwandfreie Verwendung u. a. Verfahren zur häuslichen Navigation. Beispielsweise müssen dafür Hindernisse erkannt und umfahren werden. Dabei liefern Sensoren wichtige Informationen zur Erstellung und Aktualisierung von Umgebungskarten. Diese dienen zur Selbstlokalisierung sowie zur Routenplanung innerhalb des Arbeitsbereiches. Dies ermöglicht unter anderem Serviceleistungen wie Staubsaugen, Fensterreinigung, Wegführung, Such- und Bringdienste. Des Weiteren kann ein Assistenzroboter als Mensch-Maschine-Schnittstelle eingesetzt werden. Beispielhaft fordert eine videobasierte Personen- und Gestenerkennung einen Assistenzroboter zu festgelegten Reaktionen auf. So könnten gefährliche Situationen wie Personenstürze oder akustische Notrufe wahrgenommen und auf diese schnell und effizient reagiert werden. Der Einsatz einer künstlichen Haut, bei der die Interaktion über Berührungen stattfindet, stellt eine weitere Interaktionsmöglichkeit dar.



**Abbildung 1.1:** Assistenzroboter CoRA. [SHW04]

Das Institut für Neuroinformatiker der Ruhr-Universität Bochum entwickelt seit über zehn Jahren an einem autonomen Assistenzroboter namens *Cooperative Robot Assistant* (CoRA). Eine schwenk- und neigbare Stereokamera ermöglicht das Erkennen und Lokalisieren eines angelernten Gegenstandes. Nachdem die relative räumliche Position des Gegenstandes bekannt ist, kann dieser mit einer Zwei-Finger-Hand gegriffen werden.

Die Abbildung 1.1 stellt den ortsfesten Assistenzroboter in der genannten Ausübung dar. Dabei besitzt der Roboter sieben Freiheitsgrade und eine Sprachschnittstelle, sodass Befehle über ein Mikrofon aufnehmbar sind. Eine weitere Fähigkeit ist die vorhandene künstliche Haut, welche Berührungen wahrnimmt [SHW04].

Ein zweiter innovativer Assistenzroboter ist der mobile Haushaltsroboter *Care-O-bot 3*. Dieser wurde 2008 vom Fraunhofer-Institut für Produktionstechnik und Automatisierung in Stuttgart entwickelt. Im Vergleich zum Roboter der Ruhr-Universität Bochum ermöglicht eine Kombination aus Stereo- und Lichtlaufzeitkamera das Identifizieren, Lokalisieren und Greifen von antrainierten Gegenständen. Für diese Aufgabe verfügt der Care-O-bot über eine multimodale Mensch-Maschine-Interaktion sowie der Manipulationsmöglichkeit mit Hilfe eines hochflexiblen Armes mit sieben Freiheitsgraden und einer 3-Finger-Hand. Im Gegensatz zum Assistenzroboter der Universität Bochum ist der Roboter nicht ortsfest. Die mobile Plattform ermöglicht zusammen mit den angebauten Kameras eine omnidirektionale Navigation bei der zusätzlich eine 3D-Umgebungskarte aufgebaut wird. Die Navigation wird dabei von einer Hinderniserkennung unterstützt. Die Abbildung 1.2 stellt den Care-O-bot 3 dar.



**Abbildung 1.2:** Care-O-bot 3. [FRA09]

Die Anwendungsbreite von Assistenzrobotern reicht noch viel weiter hinaus. Durch die rasante technologische Weiterentwicklung ist darauf zu schließen, dass sich in Zukunft die Assistenzroboter weiter verbreiten und in neue Anwendungsbereiche etablieren werden.

Weitere Informationen über den Praxiseinsatz von Assistenzrobotern finden sich in [HM05], [ICH05] und [NEU10].

## 1.2 Zielsetzung

In dieser Arbeit ist eine autonome Identifizierung, Lokalisierung und Mobilisierung von Gegenständen zu entwickeln. Dabei soll der Lösungsansatz auf Basis von Bildmerkmalen realisiert werden. Für ein derartiges Vorhaben müssen die Merkmale in perspektivisch unterschiedlichen Aufnahmen eindeutig wiedererkennbar sein. In dem Fall, dass ein Merkmal falsch klassifiziert wird, beeinflusst dies die Identifizierung und nachfolgend die Lokalisierung erheblich. Aus diesem Grund ist eine falsche Merkmalszuordnung auszuschließen.

Damit Gegenstände auch aus unterschiedlichen Blickwinkeln identifiziert und lokalisiert werden können, muss eine Kamerabewegung möglich sein. Des Weiteren muss für die Modellgenerierung die Kamera oder der Gegenstand bewegbar sein. Zum einen wegen der Kamerabewegung und andererseits zur Mobilisierung von Gegenständen soll der Katana-Greifarm der Firma [NEU10] zum Einsatz kommen. Aufgrund dessen sind die nachfolgend aufgelisteten Eigenschaften des Roboters zu beachten. Dabei entspricht die maximale Bewegungsgeschwindigkeit einem Meter pro Sekunde.

- Bewegungsgeschwindigkeit von 90° pro Sekunde
- Maximale Nutzlast von 400 g

- Am Endeffektor zu montierende Kamera mit einem Weitwinkelobjektiv
- Ortsgebundener Roboterarm mit einem Bewegungsradius von 60 cm

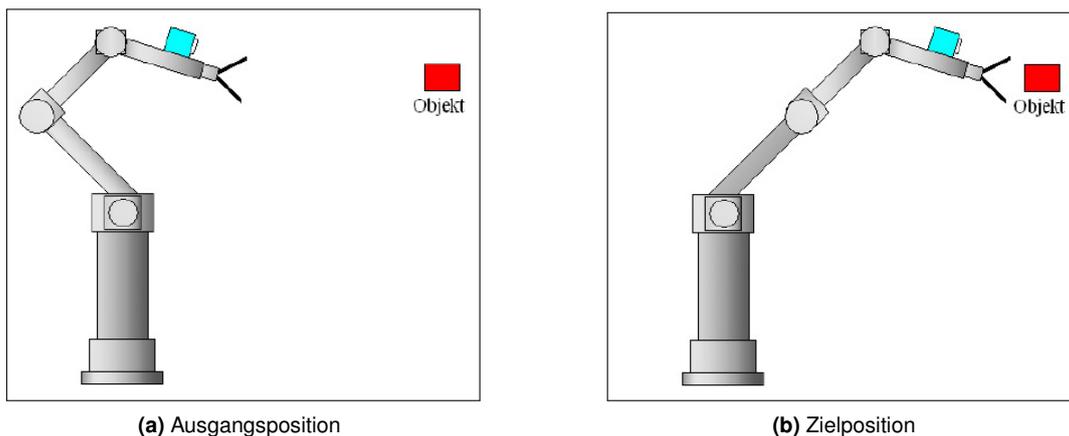
Eine weitere Bedingung ist die Position des zu greifenden Gegenstandes. Ein Gegenstand muss sich im sichtbaren und greifbaren Umfeld des Greifarmes befinden. Für den Greifvorgang ermöglicht ein montierter Winkelgreifer das Greifen von Gegenständen. Dieser besteht aus zwei Fingern, welche mit Infrarot-, Kraft-, Leitfähigkeitssensoren und *Encoder*n für die Greiferöffnung bestückt werden können.

Der zu verwendende 4 kg leichte Knickarmroboter ist in Abbildung 1.3 dargestellt.



**Abbildung 1.3:** Katana-Greifarm. [NEU10]

Die nachfolgende Abbildung 1.4 veranschaulicht die gesamt zu behandelnde Problemstellung. Die Grafik 1.4a stellt die Ausgangsposition des Roboters dar. Es soll der rot-dargestellte Gegenstand erkannt, lokalisiert und bewegt werden. Die türkis-markierte Kamera liefert Bildinformationen. Diese Aufnahmen ermöglichen das Identifizieren und Lokalisieren eines Gegenstandes, sodass sich der Knickarmroboter in die Zielposition 1.4b bewegen und den Gegenstand im anschließenden greifen kann.



**Abbildung 1.4:** Identifizierung, Lokalisierung und Greifen eines Gegenstandes. [WAG11]

## 1.3 Gliederung

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

- Kapitel 1** beschreibt den heutigen Stand der autonomen Assistenzroboter und erläutert das Ziel dieser Arbeit.
- Kapitel 2** gibt einen Überblick über existierende Lösungskonzepte zur Korrespondenzsuche und Posenbestimmung.
- Kapitel 3** stellt für das notwendige Verständnis dieser Arbeit Mathematik- und Bildverarbeitungsgrundlagen vor. Des Weiteren werden diverse Koordinatensysteme vorgestellt.
- Kapitel 4** beschreibt den technischen Aufbau bestehend aus einem Roboterarm, einem Steuerungskontroller, einem Computer, diverser Kameras und einem Drehmesstisch.
- Kapitel 5** erläutert wie aus perspektivisch unterschiedlichen Kameraaufnahmen ein 3D-Merkmalmodell (Oberflächenpunkte + Merkmale) von einem Gegenstand rekonstruiert wird.
- Kapitel 6** beschreibt die relative räumliche Orientierung zwischen einer Kamera (-aufnahme) und einem Gegenstand.
- Kapitel 7** erläutert die Bewegung eines Gegenstandes mit einem Roboterarm.
- Kapitel 8** bewertet die Messergebnisse der Identifikation, Lokalisierung und Mobilisierung von Gegenständen in Bezug auf Präzision, Geschwindigkeit, Struktur- und Materialeinfluss. Des Weiteren werden Messresultate vorgestellt, welche die Leistungsfähigkeit des in dieser Arbeit implementierten Verfahrens veranschaulichen.
- Kapitel 9** fasst die Resultate dieser Arbeit zusammen und stellt mögliche Weiterentwicklungen vor.

## 2 Vergleichbare Arbeiten

### Inhalt

---

2.1	Sensorbasierte Methoden . . . . .	5
2.2	Bildbasierte Methoden . . . . .	5
2.2.1	Markerbasierte Verfahren . . . . .	6
2.2.2	Markerlose Verfahren . . . . .	7

---

Diese Arbeit beschäftigt sich mit dem identifizieren, lokalisieren und bewegen von Gegenständen. In diesem Kapitel werden bestehende Lösungsansätze vorgestellt und verglichen. Anmerkend sei erwähnt, dass die Identifizierung zur Lokalisierung gehört und das Wissen über die räumliche Lage mitsamt Ausrichtung eine Mobilisierung ermöglicht. Aus diesem Grund ist in diesem Kapitel das Augenmerk auf die Posenbestimmung gelegt. Dabei beschreibt eine Pose die räumlichen Lage und Orientierung eines Gegenstandes relativ zu einem Bezugspunkt, wie beispielsweise einer Kamera.

Die bestehenden Verfahren zur Posenbestimmung werden nach [MÜC07] in sensorbasierte- und bildbasierte Methoden eingeordnet. Die nachfolgenden zwei Unterkapitel gehen auf diese ein.

### 2.1 Sensorbasierte Methoden

Bei der sensorbasierten Posenbestimmung wird mit ein oder mehreren Sensoren gearbeitet. Unter anderem können diese magnetisch, akustisch, funkbasiert oder inertial sein. Beispielsweise ist ein funkbasiertes Verfahren das globale Navigationssatellitensystem namens *Global Positioning System* (GPS). GPS basiert auf 24 weltumfassend verteilten Satelliten, welche über die Atomzeit synchronisiert sind. Diese versenden in kontinuierlichen zeitlichen Abständen Nachrichten, die von GPS-Empfängern entgegengenommen werden. Dabei beinhaltet eine Nachricht z.B. die Satellitenkennung, den Sendezeitstempel und die Satellitenposition. Als erstes berechnet ein Empfänger aus dem mitgesandten Sende- und dem eigenen Empfangszeitstempel die Laufzeit der Nachricht. Anschließend lässt sich aus der Laufzeit und der bekannten Übertragungsgeschwindigkeit die Entfernung zum Satelliten berechnen. Abschließend ist die Empfängerposition durch eine Triangulation zu bestimmen, indem mehrere Satellitennachrichten ausgewertet werden [BRA10]. Dabei ist eine Lokalisierungs-genauigkeit von einem Meter möglich. Allerdings kann durch geeignete Differenzverfahren eine Genauigkeit von einem Zentimeter erzielt werden. Dies führt zu dem sogenannten *Differential Global Positioning System* (DGPS) [GGE09]. Die Kommunikation mit einem GPS-Satelliten ist in Räumen bzw. Gebäuden oftmals problematisch. Die GPS-Signale sind üblicherweise zu schwach um Wände und Decken störungsfrei zu durchdringen. Aus diesem Grund werden seit ein paar Jahren Verfahren weiterentwickelt, die Signale auf Basis der *Ultra-Breitband-Technologie* (UWB) aussenden und empfangen [RML10].

Eine detaillierte Übersicht über sensorbasierte Verfahren sind in [TEG06] zu finden.

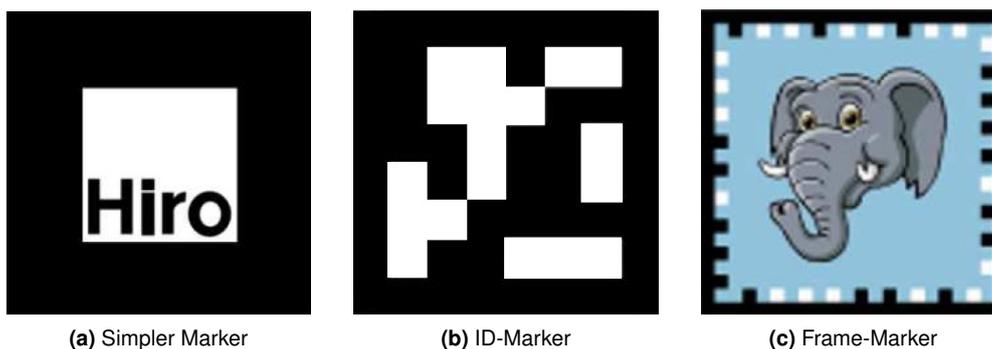
### 2.2 Bildbasierte Methoden

Die bildbasierten Methoden unterteilen sich in markerbasierte und markerlose Verfahren. Diese verwenden eine oder mehrere Kameras und sind aus diesem Grund üblicherweise preisgünstiger als die sensorbasierten Verfahren aus Kapitel 2.1. Die bildbasierten Verfahren berechnen eine Pose anhand von Punktkorrespondenzen in perspektivisch unterschiedlichen Bildern. Dabei unterscheiden sich die Verfahren in der Detektion von künstlichen-

oder natürlichen Bildmerkmalen sowie deren Korrespondenzsuche. Im Nachfolgenden werden markerbasierte und markerlose Verfahren vorgestellt.

### 2.2.1 Markerbasierte Verfahren

Bei den markerbasierten Verfahren finden künstliche Marker Verwendung. Dabei ist die Geometrie und Textur eines Markers bekannt. Die Abbildung 2.1 veranschaulicht drei Arten von künstlichen Markern. Ein einfacher Marker zeigt die Grafik 2.1a. Dieser besitzt eine weiße Grundfläche auf der ein schwarzer Rahmen aufgedruckt ist. Die jeweils vier äußeren- und inneren Ecken sind die Merkmale, welche in jeder Kameraaufnahme zur Korrespondenzfindung gesucht werden. Es gibt mehrere Methoden die Bildposition der Eckpunkte zu bestimmen [JÄH05, SCH00]. Einen Ansatz beschreibt [FS10](S.42-49). Zuerst werden alle Kanten in der Kameraaufnahme gesucht. Dafür kann der Canny-Algorithmus zum Einsatz kommen [CAN86]. Dieser erzeugt aus einem Grauwertbild ein Kanten- und Gradientenrichtungsbild. Aus diesem Grund muss ein Farbbild in ein Grauwertbild umgewandelt werden. Anschließend bestimmt die Hough-Transformation die Position und Ausrichtung jeder Kantengerade. Dabei definiert die Geradensteigung ob es sich um eine vertikale- oder horizontale Linie handelt. Zu Letzt gibt jeder Schnittpunkt einer vertikalen- mit einer horizontalen Gerade Aufschluss über die gesuchten acht Kantenpositionen im Bild. An dieser Stelle angelangt fehlt jedoch noch die Information um welchen Kantenpunkt es sich handelt. Denn wenn die Ecken nicht eindeutig identifizierbar sind, lassen sich keine Korrespondenzen zwischen Bildern ermitteln. Aus diesem Grund besitzt der Marker noch eine weitere Bezugsquelle innerhalb des schwarzen Rahmens. Dort befindet sich beispielsweise ein Wort, welches erkannt wird und anschließend die Indizierreihenfolge der Ecken festlegt [FIA05].

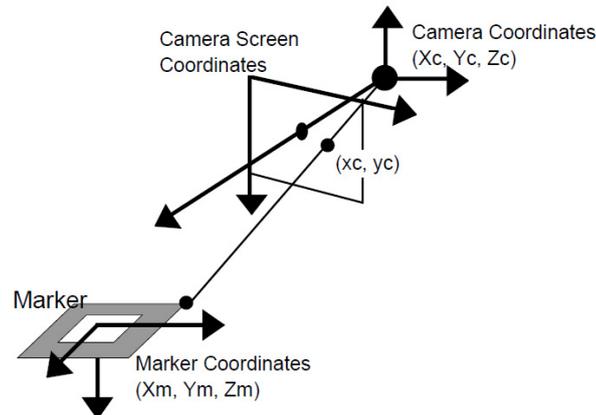


**Abbildung 2.1:** Drei typische künstliche Marker. (a) Einfacher Marker (b) Marker mit einer binären Kennung (c) Marker ist binärkodiert und invariant gegenüber Teilverdeckung [MAI09]

Die Grafik 2.1b besitzt im Gegensatz zum einfachen Marker eine binäre Kodierung. Ein solches Bitmuster ermöglicht eine schnelle und zuverlässige Indizierung der Eckpunkte. In der Grafik 2.1c ist der sogenannte Frame-Marker abgebildet. Dieser beinhaltet kein Bitmuster im Inneren des Markers. Die Binärkodierung des Markers befindet sich lediglich am Innenrand. Dabei besitzt jeder Rand neun Quadrate, welche entweder schwarz oder weiß sind. Insgesamt verfügt damit ein Marker über  $4 \cdot 9 = 36$  Quadrate, wobei jeweils neun eine Ecke indizieren. Daraus ergeben sich  $2^9 = 512$  Kombinationsmöglichkeiten zur Indizierung. Ein Vorteil bei diesem Marker ist die Invarianz gegenüber einer Teilverdeckung. Denn Quadrate werden im Uhrzeigersinn interpretiert, sodass lediglich eine vollständige Seite mit jeweils den neun Quadraten detektiert werden muss, um alle vier Ecken zu indizieren. Im Inneren des Markers kann gegebenenfalls eine Grafik eingebettet werden. Dies ist jedoch nicht notwendig und wird im Allgemeinen im Detektionsalgorithmus nicht berücksichtigt. [WLS08] evaluierte den Frame-Marker auf einem Smartphone, dass mit einem 400 MHz Prozessor und einer Kamera mit einer 320 x 240 Bildauflösung ausgestattet war. Dabei wurden die Kanten in einer Kameraaufnahme in 0.9 ms extrahiert. Insgesamt konnte ein Marker mit seinen Merkmalen in einer Zeit von 4.1 ms erkannt werden.

Es existieren noch weitere Marker, wie beispielsweise der Split- oder Dot-Marker. Detaillierte Informationen sind in [FIA05] und [WLS08] zu finden.

Wie bereits auf Seite 5 erwähnt, findet die relative Lokalisierung anhand von Punktkorrespondenzen in perspektivisch unterschiedlichen Bildern statt. Die Abbildung 2.2 veranschaulicht dies. Die Kanten eines Markers werden eindeutig identifiziert, was wiederum zu einer fehlerfreien Korrespondenzsuche führt. Eine abschließende Triangulation bestimmt die Translation (Lage) und Rotation (Orientierung) zwischen Kamera und Marker.



**Abbildung 2.2:** Relative Posenbestimmung mit künstlichen Markern. [KB99]

Der Vorteil der Verwendung von Markern liegt in der Wiedererkennungsrate und deren Positionsberechnung. Ein im Kamerabild sichtbarer Marker wird schnell und korrekt wiedererkannt. Aufgrund der bekannten Markergeometrie ist eine sehr genaue Kantenposition bestimmbar. Diese genannten Eigenschaften führen zu fehlerfreien Korrespondenzen. Der große Nachteil des markerbasierten Ansatzes ist das manuelle Anbringen des Markers an den Gegenständen. Je nachdem wie viele Marker benötigt werden, ist dies sehr aufwendig [KB99].

### 2.2.2 Markerlose Verfahren

Die Nachteile des markerbasierten Ansatzes sind bei den markerlosen Verfahren nicht vorhanden. Diese arbeiten mit natürlichen Markern, welche jeder Gegenstand bzw. jedes Objekt besitzt. Dies können Ecken, Kanten, Flächen, Konturen oder Nachbarschaftsinformationen bzgl. eines Pixels sein. Jede Kameraaufnahme enthält abhängig von Aufnahme und Auflösung eine Vielzahl von natürlichen Markern. Die Problematik liegt in der Berechnungsdauer und der eindeutigen Identifizierung. Zahlreiche Forschungsergebnisse haben jedoch gezeigt, dass eine falsche Zuordnung von natürlichen Markern in einem adäquaten Berechnungsaufwand fast vermieden werden kann [Low04, Hof09].

Der sogenannte *Harris Corner Detector* erschien erstmals in [HS88]. Mit diesem ist es möglich die Ecken in einem Bild zu extrahieren. Lediglich die Ecken ohne Zusatzinformationen zu verwenden, ist jedoch durch ihre schlechte Unterscheidbarkeit nicht als wiedererkennbares Merkmal geeignet. Dagegen wurde 1999 ein Verfahren namens *Scale Invariant Feature Transform* (SIFT) veröffentlicht, welches eine gute Merkmalsdetektion besitzt. Diese Merkmale verfügen über eine große Invarianz gegenüber Rotation, Translation, Belichtung und Verzerrung, und können damit gut in unterschiedlichen Bildern wiedererkannt werden [Low99]. Ein weiterer Merkmalsdetektor wurde in [BETG08] veröffentlicht, welcher ähnliche Eigenschaften wie die SIFT-Merkmale besitzt. Gleichermäßen wie SIFT arbeitet dieser mit Nachbarschaftsinformationen von einzelnen Bildpunkten.

In [WAG11] wird eine zweistufige Methode, zur Lösung der in dieser Arbeit gestellten Problemstellung, beschrieben. Das Verfahren setzt sich aus einer direkten Steuerung und einer bildpunktbasierter Regelung zusammen.

#### 1. Direkte Steuerung

Auf Basis einer 3D-Teilrekonstruktion eines Gegenstandes kann die aktuelle Objektlage und -ausrichtung relativ zur Kamera berechnet werden. Dafür wird eine Kameraaufnahme ausgewertet um anschließend eine definierte Greifpose abhängig zur berechneten Objektpose zu bestimmen. Danach kann ein Knickarm-Roboter die berechnete Greifpose anfahren.

## 2. Bildpunkt-basierte Regelung

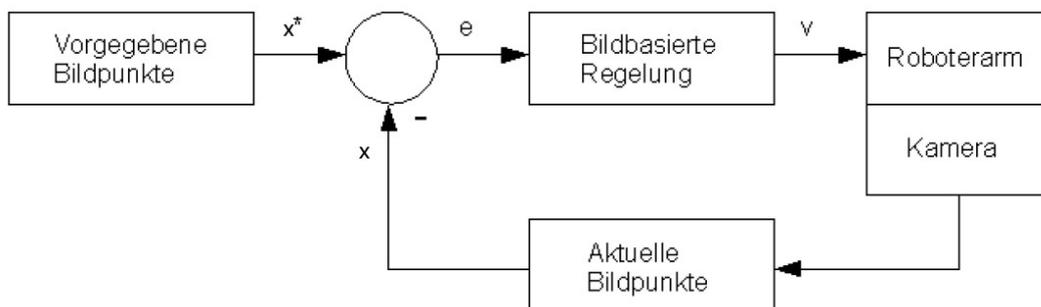
Da die direkte Steuerung die gewünschte Greifpose nicht exakt berechnet, kommt anschließend eine bildpunkt-basierte Regelung zum Einsatz. Dabei findet ein kontinuierlicher Vergleich zwischen der aktuellen- und einer initial festgelegten Zielansicht statt.

Die bildpunkt-basierte Regelung wird als sogenanntes *2D Visual Servoing*-Verfahren bezeichnet und arbeitet grundlegend in Abhängigkeit einer Regelung. Diese wird im Folgenden erläutert.

Eine am Roboter-Endeffektor montierte Kamera nimmt in kontinuierlichen Abständen Aufnahmen auf. Dabei enthält eine Aufnahme den zu lokalisierenden und zu bewegenden Gegenstand.

Das Verfahren basiert auf einer vorab gespeicherten Zielansicht, aus der mindestens vier Gegenstandsmerkmale als Sollgröße  $\mathbf{x}^*$  festzulegen sind. Damit setzt sich die Zielansicht durch die Orientierung von mindestens vier bekannten Gegenstandsmerkmalen (Siehe Merkmale, Detektion sowie Merkmalsvergleich ab S. 43) zusammen. Diesbezüglich besteht eine Merkmalsorientierung aus einer  $x$ - und  $y$ -Bildkoordinate. Aus Sicherheitsgründen wurden in [WAG11]  $n \geq 4$  bekannte Merkmale detektiert und als Zielansicht definiert.

In Abbildung 2.3 ist der verwendete Regelkreis abgebildet.



**Abbildung 2.3:** Regelkreisbasierte Posenbestimmung. [WAG11]

In der Regelung ist die Sollgröße  $\mathbf{x}^*$  die Orientierung der bekannten Merkmale aus der Zielansicht. Die Istgröße  $\mathbf{x}$  ist die aktuelle Orientierung dieser Merkmale. Die Differenz der Ist- und Sollgröße gibt die momentane Regelabweichung  $\mathbf{e}$  an. Aus der Regelabweichung wird die Bewegung des Knickarmroboters in Richtung zur Zielansicht abgeleitet. Für die Berechnung der Roboterbewegung wird dabei ein Regelgesetz aus [CH06] verwendet. Die Bewegung wird in Geschwindigkeiten  $\mathbf{v}$  bezüglich des Knickarm-Basiskoordinatensystems angegeben, wobei sich das Basiskoordinatensystem im Fuß des Roboters befindet. Die Regelabweichung wird dabei im Verlauf minimiert. Der Knickarmroboter nähert sich dadurch der Zielansicht (Regelabweichung  $\approx 0$ ) an und lokalisiert letztendlich den gesuchten Gegenstand.

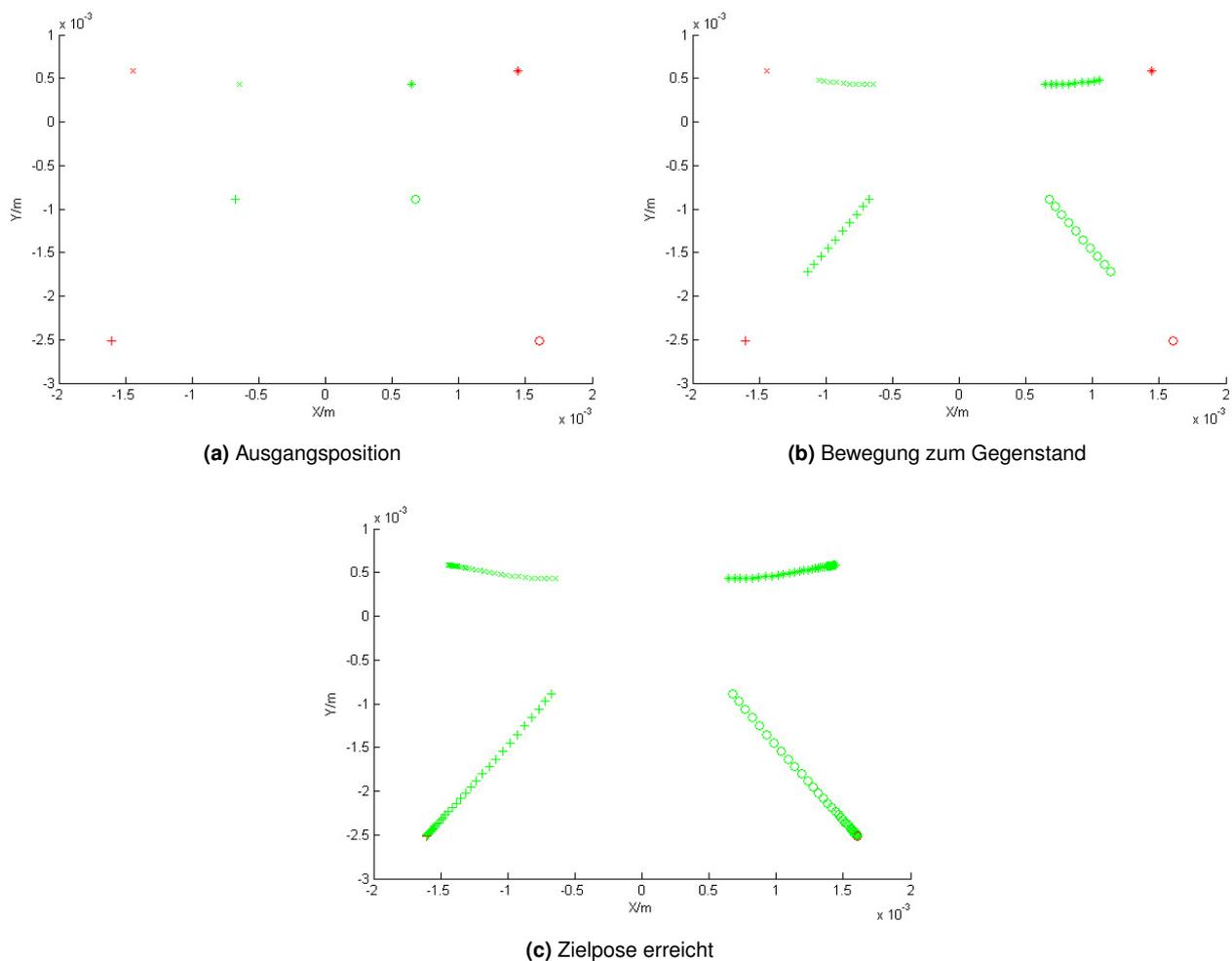
Befindet sich die Zielansicht jedoch zu weit von der Startansicht entfernt, kann das Verfahren instabil werden. Des Weiteren besteht eine Instabilität, wenn die Verdrehung der Kamera um die optische Achse zwischen der Start- und Zielpose zu groß ist. (Vgl. [CH06])

Zusammenfassend ist der *2D Visual Servoing*-Algorithmus durch folgende sechs Schritte beschrieben.

1. Zielansicht (Orientierung von mindestens vier Merkmalen) festlegen
2. Aufnahme eines Kamerabildes
3. Detektion der aktuellen Merkmale  $\mathbf{x}$ , welche mit den vorgegebenen Merkmalen  $\mathbf{x}^*$  korrespondieren
4. Triangulation der Merkmalskorrespondenzen  $\mathbf{x}$  und  $\mathbf{x}^*$  zur Entfernungsmessung
5. Bewegung des Knickarmroboters, anhand der Regelabweichung  $\mathbf{e}$  und der Abstandswerte, in Richtung zur Zielansicht ermitteln und durchführen
6. Wiederhole ab 2. bis Regelabweichung  $\approx 0$

Der Vorteil eines solchen Regelkreises ist die kontinuierliche Korrektur der aktuellen Roboterposition. Dies führt zu einer sehr genauen Lokalisierung eines gesuchten Gegenstandes. Des Weiteren ist das Verfahren robust gegenüber Kalibrierungsungenauigkeiten.

Die nachfolgende Abbildung 2.4 veranschaulicht das, mit der Matlab Robotics- [COR96] sowie der Epipolar Geometry Toolbox [MP05], simulierte Verfahren. Die erste Grafik 2.4a stellt die Ausgangsposition des Knickarmroboters dar. Die grün-markierten Merkmale symbolisieren die detektierten Merkmale in der aktuellen Aufnahme. Die rot-dargestellten Merkmale kennzeichnen die Merkmale der Zielansicht. Im zweiten Bild 2.4b hat sich der Knickarmroboter auf Basis der Regelabweichung in Richtung zum gesuchten Gegenstand bewegt. Die bis dahin grün-markierten Merkmale symbolisieren den Bewegungsverlauf der Kamera sowie der Merkmalsorientierung. In der letzten Grafik 2.4c ist die Regelabweichung gegen null konvergiert. Der Knickarmroboter befindet sich in der Zielpose und hat damit den Gegenstand lokalisiert.



**Abbildung 2.4:** Simulierter Ablauf des *2D Visual Servoing*-Verfahrens. [WAG11]

Ein Nachteil macht sich an dieser Stelle bemerkbar. Eine Lokalisierung erfolgt immer mit einer zwangsweisen Roboterbewegung zum Gegenstand, sodass dieser nicht ohne Kamerabewegung exakt lokalisiert werden kann. Des Weiteren ist die kontinuierliche Bildauswertung sehr zeitaufwendig. Dadurch benötigt das Verfahren von [WAG11] mehrere Minuten um einen Gegenstand zu greifen.

Weitere Informationen über das *2D Visual Servoing*-Verfahren sind aus [WAG11] und [CH06] zu entnehmen.

Im Gegensatz zum Visual-Servoing-Verfahren wurde in [EBE09] ein rein modellbasierter Ansatz zur Posenbestimmung untersucht. Bei dieser Methode wird in einem initialen Vorgang ein 3D-Modell von einem Gegenstand erstellt. Dafür wird eine monokulare Kamera und ein Stereoalgorithmus verwendet, welcher in perspektivisch

unterschiedlichen Aufnahmen Korrespondenzen von künstlichen SIFT-Markern (Bildmerkmale) sucht und trianguliert. Das entstehende 3D-Merkmalmodell ist ein rekonstruiertes Oberflächenmodell bestehend aus 3D-Punkten und natürlichen Markern. Jeder Oberflächenpunkt wird somit über die natürlichen Marker indiziert. Das 3D-Merkmalmodell ermöglicht in einem aktiven Prozess die relative Lokalisierung zwischen einer Kameraaufnahme und dem Gegenstand. Dafür wird eine Aufnahme auf natürliche Marker untersucht. Die gefundenen Marker werden anschließend mit den Markern des 3D-Merkmalmodells verglichen. Abschließend führt die 2D-zu-3D-Korrespondenzmenge unter Verwendung der *Direct Linear Transformation* (DLT) zu der gesuchten relativen Pose. Weiterführend wurde ein lineares Optimierungsverfahren zur robusteren Posenschätzung integriert.

Der entwickelte Lösungsansatz benötigt eine kalibrierte Kamera. Da eine Falschzuordnung von SIFT-Merkmalen nicht ausgeschlossen werden konnte, erfolgt während der Modellerstellung ein manueller Benutzereingriff. Dabei wird die erste Schätzung der Epipolargeometrie durch eine manuelle Selektion von SIFT-Merkmalen fehlerfrei realisiert. Einen ähnlichen modellbasierten Ansatz verfolgte [LF05] ohne manuellen Eingriff.

# 3 Grundlagen

In diesem Kapitel werden die benötigten Grundlagen der Bildverarbeitung und der Mathematik beschrieben.

## Inhalt

---

3.1	Koordinatensysteme . . . . .	11
3.1.1	Kamerakoordinatensystem . . . . .	11
3.1.2	Bildkoordinatensystem . . . . .	12
3.1.3	Gelenkkoordinatensysteme vom Roboter . . . . .	12
3.1.4	Weltkoordinatensystem . . . . .	12
3.1.5	Endeffektorkoordinatensystem am Greifer . . . . .	13
3.1.6	Objektkoordinatensystem . . . . .	13
3.2	Singulärwertzerlegung . . . . .	13
3.3	Fundamentalmatrix . . . . .	14
3.3.1	Die Fundamentalmatrix F . . . . .	14
3.3.2	Berechnung der Fundamentalmatrix . . . . .	15
3.3.3	Robuste Schätzung der Fundamentalmatrix . . . . .	17

---

## 3.1 Koordinatensysteme

Ein Koordinatensystem wird zur Positions- und Orientierungsangabe verwendet. In dieser Arbeit existieren diverse Koordinatensysteme, welche in bedeutender Relation zueinander stehen. Mit Hilfe von Koordinatentransformationen können dabei Koordinatenpunkte in ein anderes Bezugssystem umgerechnet werden.

Nachfolgend werden die sechs elementaren Koordinatensysteme dieser Arbeit vorgestellt. Die Beschreibung erfolgt in Bezug auf den Roboter-Greifarm (Siehe Kapitel 4.1).

### 3.1.1 Kamerakoordinatensystem

Das 3D-Kamerakoordinatensystem besteht aus drei senkrecht aufeinander stehenden Koordinatenachsen  $(x_{cam}, y_{cam}, z_{cam})$  und dem Koordinatenursprung (Schnittpunkt der Achsen). Es definiert die Position und Orientierung des Betrachters. Die mit  $z_{cam}$  gekennzeichnete Achse wird als sogenannte optische Achse, die sich vom Bildhauptpunkt durch das Linsenzentrum aufspannt, bezeichnet. Das Kamerakoordinatensystem hat einen Linsenmittelpunkt, der als Ursprung benannt ist. Da außerdem die Koordinatenachsen in jedem Punkt senkrecht aufeinander stehen, bezeichnet man dieses als orthogonal.

Die Grafik 3.1a stellt das Koordinatensystem der Kamera dar.

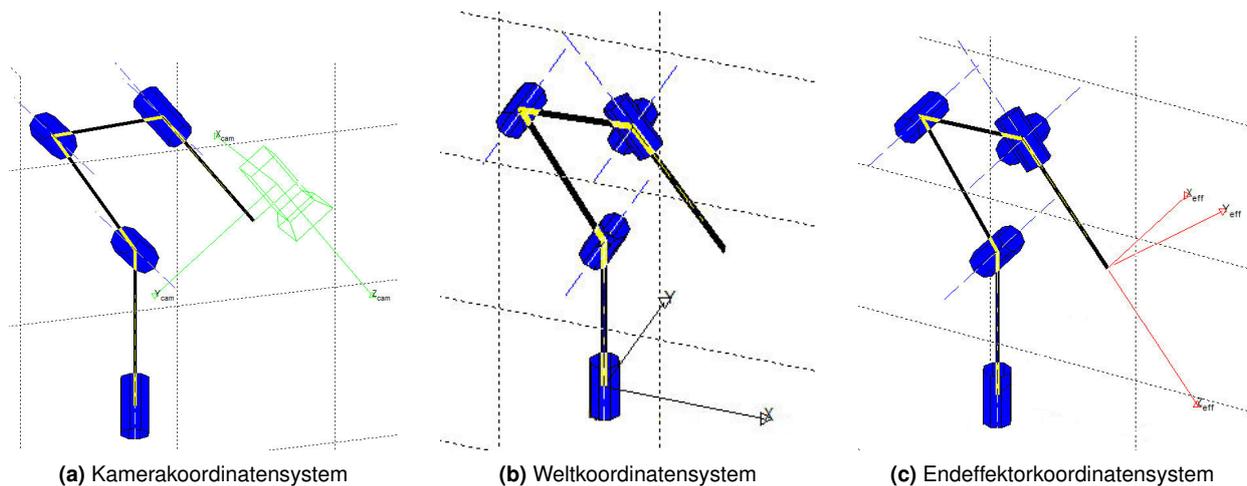


Abbildung 3.1: Kamera-, Welt- und Endeffektorkoordinatensystem.

### 3.1.2 Bildkoordinatensystem

Ein Bildkoordinatensystem  $(x, y)$  ist ein rechtwinkliges zweidimensionales Koordinatensystem, welches eine zur Bildebene parallele Koordinatenebene besitzt. Der Ursprung befindet sich im Projektionszentrum und ist im Allgemeinen die linke untere oder obere Bildecke. Das Bildkoordinatensystem ist ein Bezugssystem, welches die Position der Pixel im Bildaufnehmer angibt. Dabei ergeben sich die Bildkoordinaten aus der Zeilen- und Spaltenmatrix des Bildaufnehmers.

Die intrinsischen Kameraparameter beschreiben den Zusammenhang des Kamera- und Bildkoordinatensystems. Zu den intrinsischen Parametern gehört die interne Geometrie der Kamera und damit Eigenschaften wie Bildweite, Pixelkoordinaten der Bildmitte, Scherungsfaktor sowie die Linsenverzeichnung. Weitere Informationen sind im Kapitel 5.1.1 ab S. 27 zu finden.

### 3.1.3 Gelenkkoordinatensysteme vom Roboter

Der Katana-Roboter besitzt fünf Gelenke. Das erste Drehgelenk befindet sich am Fuße des Roboters. Darüber kommen drei Knickgelenke und anschließend, zur Rotation des Endeffektors, das fünfte Gelenk (Drehgelenk). Jedes der insgesamt fünf Robotergelenke besitzt ein eigenes Koordinatensystem. Diese werden ausgehend vom ersten Drehgelenk durchnummeriert:

$$(x_{g1}, y_{g1}, z_{g1}) \dots (x_{g5}, y_{g5}, z_{g5}) \quad (3.1)$$

### 3.1.4 Weltkoordinatensystem

Weltkoordinaten werden bezüglich eines 3D-Weltkoordinatensystems spezifiziert und sind auf die betrachtete Szene bezogen. Es ist das Hauptkoordinatensystem, welches allen anderen Koordinatensystemen zugrunde liegt. Damit ist es ein gemeinsames Bezugssystem für sämtliche Objekte einer Szene.

Der Ursprung des Weltkoordinatensystems befindet sich im Rotationspunkt des ersten Gelenks. Die Drehachse des ersten Gelenks entspricht der  $z$ -Achse. Die Abbildung 3.1b veranschaulicht das Weltkoordinatensystem mit der Achsenbeschriftung  $(X, Y, Z)$ .

### 3.1.5 Endeffektorkoordinatensystem am Greifer

Das Endeffektorkoordinatensystem befindet sich am Winkelgreifer und liegt am sogenannten *Tool Center Point* (TCP). Aus der Sicht vom Greifarm zeigt die  $x$ -Achse nach links, die  $y$ -Achse nach oben und die  $z$ -Achse weg vom Endeffektor. In der Abbildung 3.1c ist das Koordinatensystem des Winkelgreifers mitsamt seiner Achsenbeschriftung  $(x_{\text{eff}}, y_{\text{eff}}, z_{\text{eff}})$  dargestellt.

### 3.1.6 Objektkoordinatensystem

Die  $x$ -,  $y$ - und  $z$ -Achse des Objektkoordinatensystems ist durch  $(x_{\text{obj}}, y_{\text{obj}}, z_{\text{obj}})$  gekennzeichnet.

## 3.2 Singulärwertzerlegung

Die Singulärwertzerlegung ermöglicht eine Faktorisierung einer Matrix  $\mathbf{M}$  in drei Matrizen  $\mathbf{U}$ ,  $\mathbf{S}$  und  $\mathbf{V}$  (Gleichung 3.2).

$$\text{svd}(\mathbf{M}) = [\mathbf{U}, \mathbf{S}, \mathbf{V}] \quad (3.2)$$

Dabei ergibt die Multiplikation der drei Matrizen  $\mathbf{U}$ ,  $\mathbf{S}$  und  $\mathbf{V}$  die ursprüngliche Matrix

$$\mathbf{M} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T \quad (3.3)$$

$\mathbf{U}$  ist eine unitäre  $m \times m$  Matrix mit orthogonalen Spalten,  $\mathbf{S}$  eine  $m \times n$  Diagonalmatrix und  $\mathbf{V}^T$  ist die Adjungierte einer unitären  $n \times n$  Matrix. Eine unitäre Matrix  $\mathbf{X}$  kennzeichnet eine quadratische Matrix, bei der die Spalten zueinander orthonormal sind. Die genannte Konstellation tritt auf, wenn  $\mathbf{X}$  transponiert mit sich selbst multipliziert wird und daraus die Einheitsmatrix  $\mathbf{I}$  resultiert. In der Gleichung 3.4 ist diese Konstellation dargestellt.

$$\mathbf{X}^T \cdot \mathbf{X} = \mathbf{I} \quad (3.4)$$

Zusammenfassend ergibt sich die nachfolgende Gleichung 3.5.

$$\begin{bmatrix} m_{11} & \cdots & m_{1n} \\ m_{21} & \cdots & m_{2n} \\ \vdots & \ddots & \vdots \\ m_{m1} & \cdots & m_{mn} \end{bmatrix} = \begin{bmatrix} u_{11} & \cdots & u_{1m} \\ u_{21} & \cdots & u_{2m} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mm} \end{bmatrix} \cdot \begin{bmatrix} d_1 & & & \vdots \\ & \ddots & & \vdots \\ & & d_s & \vdots \\ \hline & & & \vdots \\ \cdots & 0 & \cdots & \vdots \\ & & & \vdots \\ \vdots & & & \vdots \end{bmatrix} \cdot \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{bmatrix}^T \quad (3.5)$$

Die Spalten von  $\mathbf{U}$  heißen linke Singulärvektoren. Dementsprechend werden die Spalten von  $\mathbf{V}$  als rechte Singulärvektoren bezeichnet. Die Diagonalmatrix  $\mathbf{S}$  enthält die  $s$  Singulärwerte  $d$  der  $m \times n$  Matrix  $\mathbf{M}$ . Diese sind echt positiv und absteigend der Größe nach geordnet. Resultierend folgt die Bedingung

$$d_1 \geq d_2 \geq \cdots \geq d_s > 0 \quad (3.6)$$

Beispielsweise können die Singulärwerte von  $\mathbf{M}$  über die Eigenwerte  $\Lambda$  von  $\mathbf{M}^T \cdot \mathbf{M}$  mit

$$d_i = \sqrt{\Lambda_i} \quad \text{wobei} \quad \Lambda_i \in \Lambda(\mathbf{M}^T \cdot \mathbf{M}) \quad (3.7)$$

berechnet werden (Vgl. [ÜBE95], S. 385).

Des Weiteren ist die Singulärwertzerlegung als *Singular Value Decomposition* (SVD) bekannt und in [ROD04] und [ÜBE95] beschrieben.

### 3.3 Fundamentalmatrix

Dieses Kapitel geht auf die Fundamentalmatrix ein, welche in dieser Arbeit mehrfach Verwendung findet.

#### 3.3.1 Die Fundamentalmatrix $\mathbf{F}$

Nehmen zwei voneinander abweichend positionierte Kameras ein Objekt im dreidimensionalen Raum auf, entstehen zwei unterschiedliche Bilder. Die Fundamentalmatrix  $\mathbf{F}$  beschreibt eine Abbildung zwischen diesen Bildern. Die Grafik 3.2 stellt diese Beziehung grafisch dar. Die Projektionszentren der Kameras befinden sich an Position  $\mathbf{C}$  und  $\mathbf{C}'$ . Diese spannen eine Verbindungsgerade auf, welche als Basislinie bezeichnet wird. Die Basislinie schneidet jede Bildebene in den sogenannten Epipolen  $\mathbf{e}$  und  $\mathbf{e}'$ . Ein Raumpunkt  $\mathbf{X}$  spannt mit der Basislinie eine sogenannte Epipolarebene auf. In dieser Epipolarebene befinden die korrespondierenden Bildpunkte  $\mathbf{x}$  und  $\mathbf{x}'$ . Weiterführend schneidet eine Epipolarebene die zwei Bildebenen in den korrespondierenden Epipolarlinien<sup>1</sup>  $\mathbf{l}$  und  $\mathbf{l}'$ .

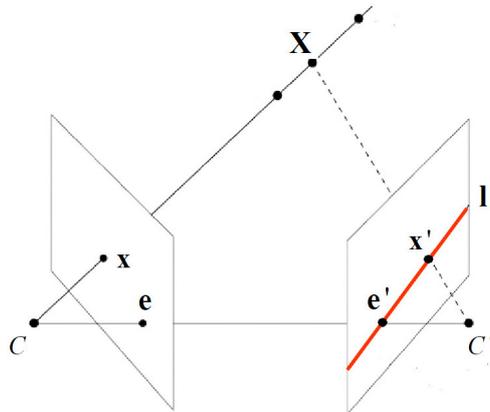


Abbildung 3.2: Beziehung zwischen zwei Bildern. [MEI08A]

Eine Veränderung der Raumpunktposition  $\mathbf{X}$  hat eine Drehung der Epipolarebene um die Basislinie zur Folge. Die Grafik 3.3 veranschaulicht diese Abhängigkeit.

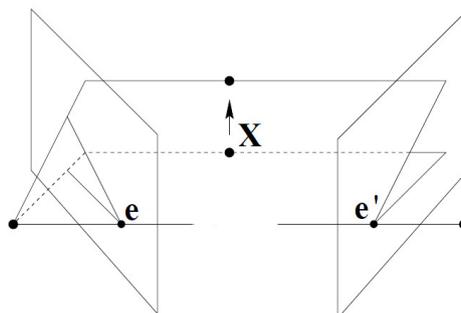


Abbildung 3.3: Epipolarebenen zweier Bilder bei Veränderung des Raumpunktes  $\mathbf{X}$ . [CAN07]

<sup>1</sup>Kernlinie ist eine synonyme Bezeichnung.

Es existieren korrespondierende Punkte  $\mathbf{x}$  und  $\mathbf{x}'$ , welche zwei perspektivisch unterschiedlichen Aufnahmen zugeordnet sind. Dann kann die Fundamentalmatrix  $\mathbf{F}$  mit den Korrespondenzen berechnet werden. Denn bei einer Punktkorrespondenz gilt, dass sich der korrespondierende Punkt  $\mathbf{x}'$  von  $\mathbf{x}$  auf der Epipolarlinie  $\mathbf{l}'$  befindet. Die Gleichung 3.8 beschreibt diesen Zusammenhang. Dabei besitzt  $\mathbf{x}$  eine  $x$ - und  $y$ -Bildkoordinate und ist als homogene Koordinate  $[x, y, 1]^T$  angegeben. Der Umkehrschluss ist in der Gleichung 3.9 aufgeführt. Dementsprechend ist  $\mathbf{x}'$  durch  $[x', y', 1]^T$  definiert.

$$\mathbf{F} \cdot \mathbf{x} = \mathbf{l}' \quad (3.8)$$

$$\mathbf{F}^T \cdot \mathbf{x}' = \mathbf{l} \quad (3.9)$$

Da sich der Punkt  $\mathbf{x}'$  auf der Epipolarlinie  $\mathbf{l}'$  befindet, gilt die Gleichung 3.10.

$$\mathbf{x}'^T \cdot \mathbf{l}' = 0 \quad (3.10)$$

Wird Gleichung 3.8 in 3.10 eingesetzt, folgt die nachfolgende Gleichung 3.11, welche als lineares Gleichungssystem aufgestellt werden kann.

$$\underbrace{\mathbf{x}'^T}_{1 \times 3 \text{ Vektor}} \cdot \underbrace{\mathbf{F}}_{3 \times 3 \text{ Matrix}} \cdot \underbrace{\mathbf{x}}_{3 \times 1 \text{ Vektor}} = 0 \quad (3.11)$$

Des Weiteren wird  $\mathbf{x}'^T \cdot \mathbf{F} \cdot \mathbf{x} = 0$  als sogenannte Epipolarbedingung bezeichnet. Die Gleichung 3.11 ermöglicht, mit mindestens acht korrespondierenden Bildpunkten, die Berechnung der Fundamentalmatrix durch ein lineares Gleichungssystem. Das in dieser Arbeit verwendete Verfahren zur Berechnung der Fundamentalmatrix ist im anschließenden Kapitel 3.3.2 beschrieben.

### 3.3.2 Berechnung der Fundamentalmatrix

In diesem Kapitel wird der normierte lineare 8-Punkte-Algorithmus (Vgl. [HZ04, LH81]) verwendet. Dieser benötigt mindestens acht korrespondierende Punkte zwischen zwei Bildern. Hier sei auf den 7-Punkte-Algorithmus verwiesen, welcher lediglich sieben Korrespondenzen benötigt. Informationen bezüglich des 7-Punkte-Algorithmus sind in [HZ04] auf Seite 281 zu finden.

Die Fundamentalmatrix  $\mathbf{F}$  ist eine  $3 \times 3$  Matrix. Besitzt diese einen Rang von zwei, ist sie singulär und damit nicht invertierbar. Sie hat neun Matrixelemente, verfügt jedoch über sieben Freiheitsgrade. Damit ist die Fundamentalmatrix überparametrisiert.

Gesucht sei eine Matrix  $\mathbf{F}$  bei der  $n \geq 8$  Punktkorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  die Gleichung 3.11 erfüllen.

Dafür werden im ersten Schritt des normalisierten linearen 8-Punkte-Algorithmus die Bildkoordinaten  $\mathbf{x}_i$  und  $\mathbf{x}'_i$  normalisiert. Die Bildkoordinaten werden aufgrund der hohen Koordinatenschwankungen in digitalen Bildern normalisiert. Auftretende Koordinatenschwankungen im Bereich von 1 bis über 1000 können zu numerischen Schwierigkeiten führen. Nähere Informationen sind in [HZ04] ab Seite 107 zu entnehmen.

Die Normalisierung erfolgt mit einer punktwweisen Schwerpunktverschiebung des Ursprungs  $\mathbf{o} = [0, 0]^T$  sowie durch das Skalieren vom mittleren Abstand der Punkte zum Schwerpunkt auf  $\sqrt{2}$ . Dies wird in einer metrischen Transformationsmatrix  $\mathbf{T}$  zusammengefasst. Sei  $\mathbf{t} = [t_x, t_y]^T$  die Schwerpunktverschiebung und  $\lambda = [\lambda_x, \lambda_y]^T$  die Skalierung, dann ergibt sich eine in Gleichung 3.12 dargestellte Transformationsmatrix  $\mathbf{T}$ .

$$\mathbf{T} = \begin{bmatrix} \lambda_x & 0 & t_x \\ 0 & \lambda_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

Unter der Verwendung von 3.12 kann die Normalisierung

$$\hat{\mathbf{x}}_i \sim \mathbf{T} \cdot \mathbf{x}_i \quad (3.13)$$

$$\hat{\mathbf{x}}'_i \sim \mathbf{T}' \cdot \mathbf{x}'_i \quad (3.14)$$

der Punkte vorgenommen werden.

Im zweiten Schritt wird für die normierten homogenen Punkte  $\hat{\mathbf{x}}_i$  und  $\hat{\mathbf{x}}'_i$  ein lineares Gleichungssystem

$$\hat{\mathbf{A}} \cdot \hat{\mathbf{f}} = 0 \quad (3.15)$$

aufgebaut.  $\hat{\mathbf{f}}_{1 \times 9} = [\hat{F}_{11}, \hat{F}_{12}, \hat{F}_{13}, \hat{F}_{21}, \hat{F}_{22}, \hat{F}_{23}, \hat{F}_{31}, \hat{F}_{32}, \hat{F}_{33}]^T$  beinhaltet die Elemente der Fundamentalmatrix  $\hat{\mathbf{F}}$  in Vektorschreibweise. Die Matrix  $\hat{\mathbf{A}}$  enthält alle normierten Punktkorrespondenzen, was zum nachfolgenden linearen homogenen Gleichungssystem 3.16 führt.

$$\begin{bmatrix} \hat{x}'_1 \hat{x}_1 & \hat{x}'_1 \hat{y}_1 & \hat{x}'_1 & \hat{y}'_1 \hat{x}_1 & \hat{y}'_1 \hat{y}_1 & \hat{y}'_1 & \hat{x}_1 & \hat{y}_1 & 1 \\ \vdots & \vdots \\ \hat{x}'_n \hat{x}_n & \hat{x}'_n \hat{y}_n & \hat{x}'_n & \hat{y}'_n \hat{x}_n & \hat{y}'_n \hat{y}_n & \hat{y}'_n & \hat{x}_n & \hat{y}_n & 1 \end{bmatrix} \cdot \hat{\mathbf{f}} = 0 \quad (3.16)$$

Das Gleichungssystem kann bis auf einen Skalierungsfaktor genau berechnet werden. Der zum kleinsten Singulärwert von  $\hat{\mathbf{A}}$  zugehörige Vektor bestimmt die Fundamentalmatrix  $\hat{\mathbf{F}}$ .

Die berechnete Fundamentalmatrix ist aufgrund von Messfehlern üblicherweise nicht wie gefordert singulär. Besitzt die Fundamentalmatrix einen vollen Rang<sup>2</sup>, dann schneiden sich die Epipolarlinien nicht in einem gemeinsamen Punkt. Aus diesem Grund muss die nachfolgend aufgelistete Singularitätsbedingung berücksichtigt werden.

$$\det(\hat{\mathbf{A}}) = 0 \quad (3.17)$$

Der kleinste Singulärwert  $d_3$  ist auf null zu setzen, damit die Singularitätsbedingung aus Gleichung 3.17 erfüllt wird. Dafür wird die Singulärwertzerlegung (Siehe Kapitel 3.2) von  $\hat{\mathbf{F}} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$  berechnet und der kleinste Singulärwert aus der 3x3 Diagonalmatrix  $\mathbf{S}$  eliminiert. Die korrigierte Fundamentalmatrix  $\hat{\mathbf{F}}_k$  ergibt sich aus der anschließenden Rekombinierung und ist in der Gleichung 3.18 dargestellt.

$$\hat{\mathbf{F}}_k = \mathbf{U} \cdot \text{diag}(d_1, d_2, 0) \cdot \mathbf{V}^T = \mathbf{U} \cdot \underbrace{\begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{d_3=0} \cdot \mathbf{V}^T \quad (3.18)$$

Im letzten Schritt des 8-Punkte-Algorithmus muss die korrigierte Fundamentalmatrix  $\hat{\mathbf{F}}_k$  mit

$$\mathbf{F} = \mathbf{T}'^T \cdot \hat{\mathbf{F}}_k \cdot \mathbf{T} \quad (3.19)$$

denormalisiert werden.

Im nachfolgend dargestellten Algorithmus 1 ist der Ablauf des 8-Punkte-Algorithmus zusammengefasst.

Ein vereinfachtes Codebeispiel des implementierten 8-Punkte-Algorithmus ist im Anhang A.1 zu finden.

<sup>2</sup>Eine Matrix besitzt einen vollen Rang, wenn ihre Determinante ungleich null ist. Alle Matrixzeilen- und spalten sind folglich linear unabhängig.

**Algorithmus 1** Normalisierter 8-Punkte-Algorithmus

1: **Vorbedingung:**  $n \geq 8$  Punktkorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$

2: **Gesucht:** Eine Matrix  $\mathbf{F}$  bei der  $n \geq 8$  Punktkorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  die Gleichung  $\mathbf{x}'^T \cdot \mathbf{F} \cdot \mathbf{x} = 0$  erfüllen.

3: **Algorithmus:**

(i) **Normalisierung:** Verschiebe den Ursprung  $\mathbf{o} = [0, 0]^T$  der Punkte in den Schwerpunkt und führe eine Skalierung durch, sodass der mittleren Schwerpunktabstand  $\sqrt{2}$  beträgt:  $\hat{\mathbf{x}}_i \sim \mathbf{T} \cdot \mathbf{x}_i$  und  $\hat{\mathbf{x}}'_i \sim \mathbf{T}' \cdot \mathbf{x}'_i$ .

(ii) **Berechne** auf Basis der normierten Korrespondenzen eine Fundamentalmatrix  $\hat{\mathbf{F}}$  mit Hilfe:

(a) **Lineare Berechnung:** Sei  $\hat{\mathbf{f}} = [\hat{\mathbf{F}}_{11}, \dots, \hat{\mathbf{F}}_{33}]^T$  und

$$\hat{\mathbf{A}} = \begin{bmatrix} \hat{x}'_1 \hat{x}_1 & \hat{x}'_1 \hat{y}_1 & \hat{x}'_1 & \hat{y}'_1 \hat{x}_1 & \hat{y}'_1 \hat{y}_1 & \hat{y}'_1 & \hat{x}_1 & \hat{y}_1 & 1 \\ \vdots & \vdots \\ \hat{x}'_n \hat{x}_n & \hat{x}'_n \hat{y}_n & \hat{x}'_n & \hat{y}'_n \hat{x}_n & \hat{y}'_n \hat{y}_n & \hat{y}'_n & \hat{x}_n & \hat{y}_n & 1 \end{bmatrix},$$

dann stelle für die normierten homogenen Punkte  $\hat{\mathbf{x}}_i$  und  $\hat{\mathbf{x}}'_i$  ein lineares Gleichungssystem  $\hat{\mathbf{A}} \cdot \hat{\mathbf{f}} = 0$  auf. Verwende die Singulärwertzerlegung zur Berechnung von  $\hat{\mathbf{F}}$ . Der zum kleinsten Singulärwert von  $\hat{\mathbf{A}}$  zugehörige Vektor bestimmt  $\hat{\mathbf{F}}$ .

(b) **Singularitätsbedingung:** Damit  $\hat{\mathbf{F}}$  singular ist, muss  $\det(\hat{\mathbf{A}}) = 0$  erfüllt sein. Dafür berechne die Singulärwerte  $\hat{\mathbf{F}} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$  und eliminiere den kleinsten Singulärwert  $d_3$  aus der Diagonalmatrix  $\mathbf{S}$  mit anschließender Rekombinierung  $\hat{\mathbf{F}}_k = \mathbf{U} \cdot \text{diag}(d_1, d_2, 0) \cdot \mathbf{V}^T$ .

(iii) **Denormalisierung:**  $\mathbf{F} = \mathbf{T}'^T \cdot \hat{\mathbf{F}}_k \cdot \mathbf{T}$

9: **Resultat:**  $\mathbf{F}$

**3.3.3 Robuste Schätzung der Fundamentalmatrix**

Mit Hilfe des 8-Punkte-Algorithmus kann die Fundamentalmatrix berechnet werden. Jedoch ist die resultierende Fundamentalmatrix im Normalfall nicht optimal, da fehlerbehaftete Merkmalskorrespondenzen die Schätzung der Fundamentalmatrix beeinflussen.

Der *RANdom SAMple Consensus (RANSAC)* wird gängigerweise herangezogen, um die Fundamentalmatrix bzw. ein Modell mit Hilfe der verfügbaren Merkmalskorrespondenzen und dem 8-Punkte-Algorithmus optimal zu schätzen. Bei dieser Schätzung der Fundamentalmatrix werden, mit einer großen Wahrscheinlichkeit, alle Fehlzuordnungen aufgedeckt. Diese falschen Merkmalskorrespondenzen werden als *Outlier* bezeichnet. Dagegen besitzen sogenannte *Inlier* eine korrekte Zuordnung. Anhand der berechneten Inlier kann eine optimale Fundamentalmatrix bestimmt werden. Nachfolgend wird diesbezüglich der RANSAC-Algorithmus vorgestellt.

Der Wert  $k$  gibt die Mindestanzahl an Punktkorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  an, welche zur Schätzung eines Modells benötigt werden. In diesem Fall ist das zu schätzende Modell die  $3 \times 3$  Fundamentalmatrix  $\mathbf{F}$ . Dadurch müssen acht Korrespondenzen für die lineare Berechnung von  $\mathbf{F}$  vorhanden sein.

Die Eingabemenge, auf deren Basis die Schätzung des Modells stattfindet, sind  $n \geq k$  Punktkorrespondenzen. Dann werden aus der Eingabemenge  $k$  Punktkorrespondenzen zufällig ausgewählt um anschließend eine Schätzung des Modells durchzuführen.

Existiert ein erstes Modell, ist die Anzahl der Inlier aus der Menge  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  zu prüfen. Dabei findet die Prüfung in Abhängigkeit eines Toleranz- bzw. Abstandsfaktors  $\alpha$  statt. Erreicht die Inlieranzahl einen festgelegten Schwellwert  $v$ , terminiert das Verfahren und es wird noch ein letztes Mal ein Modell mit allen Inlier geschätzt. Andererseits sind die vorherigen Schritte solange zu wiederholen, bis ausreichend Inlier gefunden wurden oder die maximale Anzahl an Iterationen  $N$  erreicht ist. Im Fall eines Abbruchs wird das Modell mit dem größten Datensatz (Iteration mit der größten Inlieranzahl) bestimmt und zur letzten Modellschätzung herangezogen. Das resultierende Modell ist die optimal geschätzte Fundamentalmatrix  $\mathbf{F}_{\text{opt}}$ .

Die maximale Anzahl an Iterationen wird folgendermaßen bestimmt:

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^k)} \quad (3.20)$$

$p$  sei die Wahrscheinlichkeit dass mindestens ein ausreißerfreier Datensatz ausgewählt wird. Der relative Ausreißeranteil ist durch den Parameter  $\varepsilon$  gekennzeichnet.

Bei einer Wahrscheinlichkeit  $p$  von 0.99 und einer Anwendung des RANSAC-Algorithmus zur linearen Berechnung der Fundamentalmatrix (Anzahl benötigter Punktkorrespondenzen:  $k = 8$ ), ergibt sich die notwendige Iterationsanzahl

Ausreißeranteil $\varepsilon$	5%	10%	20%	30%	40%	50%	60%	70%
Iterationen $N$	5	9	26	78	272	1177	7025	70188

**Tabelle 3.1:** Notwendige Anzahl an Iterationen in Abhängigkeit des Ausreißeranteils.

in Abhängigkeit der Anzahl an Outlier.

In dieser Arbeit wurden die in der Tabelle 3.2 zusammengefassten RANSAC-Parameter verwendet. Diese empirisch ermittelten Parameter führen zu einer hohen Wahrscheinlichkeit, dass alle Outlier aufgedeckt und somit eine optimale Fundamentalmatrix berechnet werden kann.

Parameter	Wert
$p$	0.99
$k$	8
$\varepsilon$	0.05
$N$	5

**Tabelle 3.2:** Zur Schätzung der Fundamentalmatrix verwendete RANSAC-Parameter.

Im nachfolgend dargestellten Algorithmus 2 ist das verwendete RANSAC-Verfahren zusammengefasst. Zusätzlich ist der in dieser Arbeit implementierte RANSAC-Algorithmus stark vereinfacht im Anhang A.2 vorhanden. Weitere Informationen bezüglich des RANSAC-Verfahrens sind in [FB87] und [ROD04] ab Seite 143 zu finden.

---

#### Algorithmus 2 RANdom SAMple Consensus (RANSAC)

---

- 1: **Vorbedingung:**  $n \geq k$  Punktkorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  mit  $k = 8$
  - 2: **Gesucht:** Fundamentalmatrix  $\mathbf{F}_{\text{opt}}$  (Modell), welche aus einer fehlerbehafteten Korrespondenzmenge berechnet wird.
  - 3: **Algorithmus:**
    - (i) **Modellschätzung:** Wähle per Zufall aus der Eingabemenge  $k$  Punktkorrespondenzen aus und führe eine Schätzung des Modells durch.
    - (ii) **Inlier:** Ermittle in Abhängigkeit eines Abstandsfaktors  $\alpha$  die Anzahl an Inlier aus der Menge  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ .
      - (a) **Terminiere**, falls die Inlieranzahl den festgelegten Schwellwert  $v$  erreicht hat und berechne anhand aller Inlier die optimale Fundamentalmatrix  $\mathbf{F}_{\text{opt}}$ .
      - (b) **Abbruch** bei dem Erreichen der maximalen Iterationsanzahl  $N$  und wähle den größten- von allen bisher ermittelten Datensätzen aus. Anhand des gewählten Datensatzes (Korrespondenzen der Iteration mit der größten Inlieranzahl) ist die Fundamentalmatrix  $\mathbf{F}_{\text{opt}}$  zu berechnen.
      - (c) **Wiederhole** (i), wenn der Schwellwert  $v$  und die maximale Iterationsanzahl noch nicht erreicht wurde.
  - 9: **Resultat:**  $\mathbf{F}_{\text{opt}}$ , Inlier
-

# 4 Technischer Aufbau

In diesem Kapitel wird auf die in dieser Arbeit verwendeten Hardwarekomponenten eingegangen.

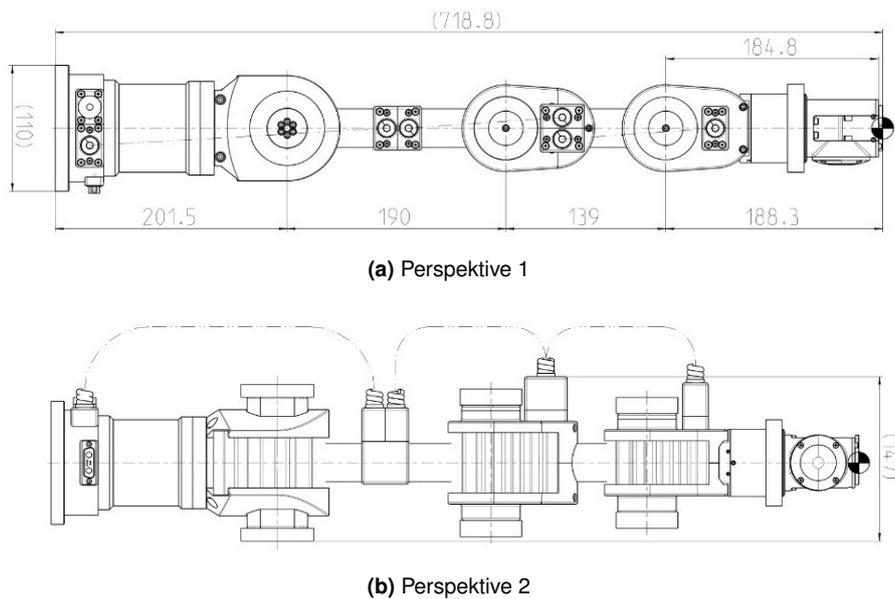
## Inhalt

4.1	Roboterarm . . . . .	19
4.2	Steuerungskontroller . . . . .	22
4.3	Computer . . . . .	23
4.4	Kameras . . . . .	23
4.5	Drehmesstisch . . . . .	25

## 4.1 Roboterarm

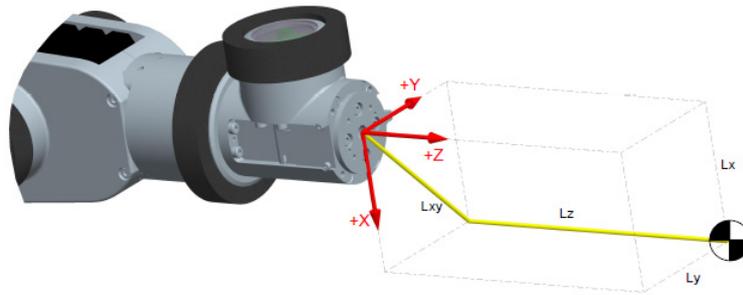
Bei dem eingesetzten Knickarmroboter handelt es sich um einen 4.8 kg leichten Katana-Greifarm der Firma [NEU10]. Dieser besitzt inklusive Greifer sechs Freiheitsgrade und kann ortsgebunden Tätigkeiten ausführen. Dabei sind Bewegungen in einem Radius von circa 60 cm möglich.

Die Millimetermaße des Greifarmes sind in der Abbildung 4.1 skizziert.

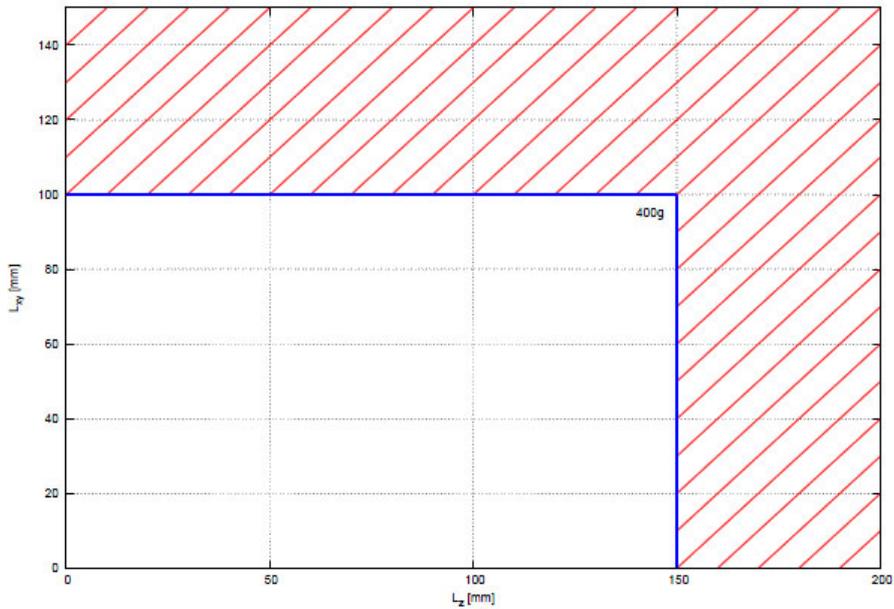


**Abbildung 4.1:** Dimensionen des Greifarmes. [NEU10]

Einen sicheren Umgang zwischen Mensch und Maschine wird durch die maximale- Kraft von 4 N sowie der Nutzlast von 400 g (ohne Greifer) gewährleistet. Die Grafik 4.2 veranschaulicht die maximal zu wirkenden Kräfte am Flanschsprung  $[X, Y, Z] = [0, 0, 0]$ . Der Nutzlastschwerpunkt wird durch den radialen Abstand  $L_{xy}$  und dem axialen Abstand  $L_z$  bestimmt. Dabei darf  $L_{xy}$  die Länge von 10 cm sowie  $L_z$  den Abstand von 15 cm nicht übersteigen (Siehe Nutzlastdiagramm 4.3), damit die maximale Nutzlast am Schwerpunkt wirken darf. Das Überschreiten der maximal zulässigen Nutzlast führt zu einer Beschädigung der Getriebe und Motoren.



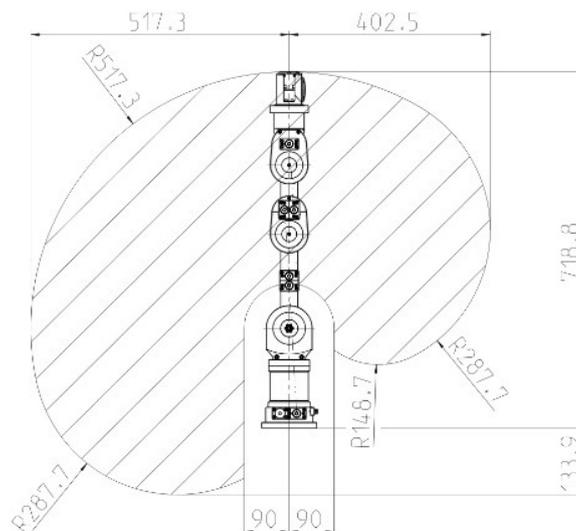
**Abbildung 4.2:** Nutzlastschwerpunkt des Greifarmes. [NEU10]



**Abbildung 4.3:** Nutzlastdiagramm des Greifarmes. [NEU10]

Die maximale Bewegungsgeschwindigkeit beträgt bei gleichzeitiger Bewegung aller Gelenke 90° pro Sekunde. Dieser Wert entspricht einer Bewegung von einem Meter pro Sekunde.

Die Abbildung 4.4 veranschaulicht den Umgebungsbereich der vom Roboter angefahren werden kann.



**Abbildung 4.4:** Arbeitsraum des Greifarmes. [NEU10]

Die nachfolgende Tabelle 4.1 listet die möglichen Verfahrwinkel der Gelenke auf.

Gelenk	Winkel absolut	Winkel relativ
1	339°	+/- 169.5°
2	132°	+102° / -30°
3	245°	+/- 122.5°
4	224°	+/- 112°
5	336°	+/- 168°

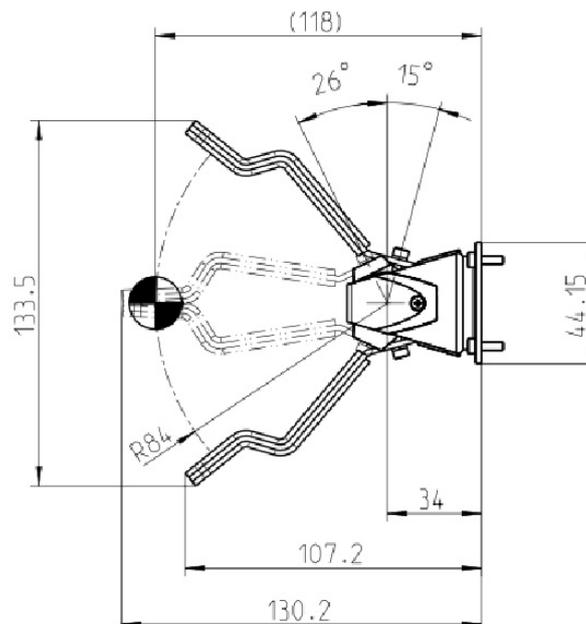
**Tabelle 4.1:** Verfahrwinkel der Gelenke.

Ein montierter Winkelgreifer ermöglicht das Greifen von Gegenständen. Dieser besteht aus zwei Fingern, welche mit Infrarot-, Kraft-, Leitfähigkeitssensoren und *Encodern* für die Greiferöffnung bestückt werden können. Der in dieser Arbeit verwendete *Katana-Greifarm 450 - 6M180* mit montiertem Winkelgreifer ist anschließend dargestellt.



**Abbildung 4.5:** Katana-Greifarm mit fünf Gelenken und einem Zwei-Finger-Greifer. [NEU10]

Die Maße [mm] und Verfahrwinkel [°] des Winkelgreifers sind in Abbildung 4.6 skizziert.

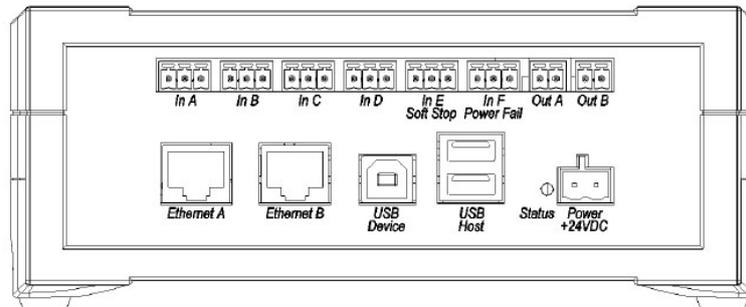


**Abbildung 4.6:** Maße und Verfahrwinkel des Winkelgreifers. [NEU10]

Des Weiteren benötigt der Roboter eine Leistung von 96 W (24 V, 4 A) und besitzt eine Wiederholgenauigkeit (Präzision) von +/- 0.1 mm.

## 4.2 Steuerungskontroller

Ein Ansteuerungskontroller wird als Schnittstelle zwischen Computer und Roboter benötigt. Dieser ist in Abbildung 4.7 dargestellt und wird in naher Umgebung zum Roboter platziert. Das in dieser Arbeit verwendete Steuerungskabel ermöglicht einen Maximalabstand von 10 m. Für die Kommunikation zwischen Computer und Kontroller stehen zwei RJ45 Ethernet-Anschlüsse sowie ein USB-Port zur Verfügung.

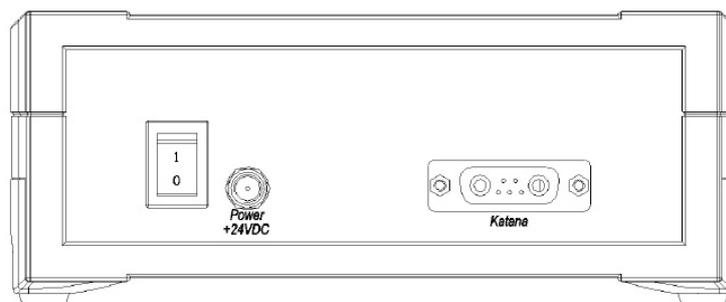


**Abbildung 4.7:** Kontroller zur Ansteuerung des Roboters (Frontansicht). [NEU10]

In dem Fall, dass die Steuerungsbefehle über einen Ethernet-Anschluss gesendet werden, ist die Kontroller-IP-Adresse 192.168.168.232 mit der zugehörigen Netzwerkmaske 255.255.255.0 zu verwenden. Erfolgt dagegen die Ansteuerung über USB, muss die IP-Adresse 192.168.1.1 mit der Netzwerkmaske 255.255.0.0 benutzt werden.

Des Weiteren stehen sechs digitale Eingänge zur Steuerung, zwei digitale Ausgänge und zwei USB-Host-Anschlüsse für Firmware-Updates sowie *Standalone*-Programmspeicher zur Verfügung.

In der nachfolgenden Abbildung 4.8 ist die Rückseite des Steuerungskontrollers abgebildet. Auf dieser Seite befindet sich der Hauptschalter mit dem der Kontroller und der Roboter an- sowie ausgeschaltet werden kann. An dieser Stelle sei daraufhin gewiesen, dass alle Gelenke des Greifarmes im ausgeschalteten Zustand nicht einrasten und somit dieser in einer ungünstigen Gelenkstellung in sich zusammen fällt. Dies kann zu Schädigungen am Greifarm führen. Die Datenübertragung zwischen Kontroller und Roboter erfolgt über den in Abbildung 4.8 dargestellten D-SUB-Anschluss.



**Abbildung 4.8:** Rückseite des Steuerungskontrollers. [NEU10]

Weitere technische Daten sind nachfolgend aufgelistet:

- Mainboard: PPC MPC5200 400 MHz
- Festplatte: 32 MB Flash
- Arbeitsspeicher: 64 MB RAM
- Betriebssystem: Embedded GNU / Linux, Kernel 2.4 mit Xenomai

### 4.3 Computer

Die Steuerungsbefehle kommen von einem Stand-PC oder einem Notebook der Marke ASUS. Das Notebook verfügt über eine *Intel Core Duo* CPU mit 1 GHz und 2 GB Arbeitsspeicher. Als Betriebssystem kam Windows 7 Professional als 32-bit Version zum Einsatz.

Die in dieser Arbeit evaluierten Messdaten beziehen sich überwiegend auf den leistungsfähigeren Stand-PC *HP Compaq 8100 Elite*. Dieser besitzt einen *Intel Core i5* Prozessor mit 2.67 GHz und 4 GB an Arbeitsspeicher. Ebenfalls wie beim Notebook ist das Betriebssystem Windows 7 Professional in der 32-bit Version zum Einsatz gekommen. Die verwendete Grafikkarte gehört zu der *ATI Radeon HD 4600 Serie* und verfügt über einen 1 GB großen DDR3-Arbeitsspeicher und einer 600 MHz Grafikprozessor (engl. *Graphics Processing Unit*, kurz GPU).

Eine Leistungsstarke Grafikkarte ist aus dem Grund von Vorteil, da die in dieser Arbeit verwendete MATLAB-Version Operationen auf der GPU auslagern kann.

### 4.4 Kameras

In dieser Arbeit finden drei unterschiedliche Kameras Verwendung.

#### 1. Drehtellerkamera

Die CCD<sup>3</sup> monochrome Kamera *DMK 41BF02* der Firma [THE09A] ist an verstellbaren Strebenprofilen eines per Schrittmotor ansteuerbaren Drehtellers montiert. Zusätzlich ermöglicht die Halterung der Kamera eine Positionierung in der Breite und einen über 180° Schwenkbereich der Kamera in vertikaler und horizontaler Richtung. Die Kamera verfügt über eine maximale Auflösung von 1280 x 960 Bildpunkten und überträgt bis zu 15 Bilder pro Sekunde über eine FireWire-Schnittstelle. Als Kameraobjektiv kam ein, speziell für den Nahbereich entwickeltes, hochauflösendes Objektiv *H0514-MP* zum Einsatz [THE09B].

Der ansteuerbare Drehteller hat für die 3D-Modellgenerierung mehrere nützliche Eigenschaften. Zum einen ermöglicht dieser eine gute grafische Verifizierung der rekonstruierten Kamerapositionen. Andererseits wird eine gleich verteilte Raumabdeckung der Kameraaufnahmen erzielt. Dies vermeidet Lücken im 3D-Modell und die in Kapitel 5.3.2.3 aufgelisteten degenerativen Daten (Merkmalskorrespondenzen).

Die Abbildung 4.9 zeigt den ansteuerbaren Drehteller in Verbindung mit einer CCD monochromen Kamera.



(a) Ansteuerbarer Drehteller



(b) Kamera

**Abbildung 4.9:** Einsatz einer CCD monochromen Kamera in Kombination mit einem ansteuerbaren Drehteller.

<sup>3</sup>CCD steht für *Charge-Coupled Device* und ist ein lichtempfindliches Bauteil welches aus einer Vielzahl von Fotodioden besteht.

## 2. Handykamera

Mit Hilfe der Sony Ericsson Handykamera konnte das entwickelte Verfahren, zur Rekonstruktion von 3D-Merkmalmodellen, aus beliebigen Posen sowie außerhalb des Labors getestet werden. Des Weiteren erlaubt diese Kamera im Vergleich zu den anderen beiden Kameras die Aufnahme von Farbbildern. Sie verfügt über eine maximale Auflösung von 3264 x 2448 Bildpunkten. In der Grafik 4.10 ist die Handykamera abgebildet.



**Abbildung 4.10:** Sony Ericsson C905 Handykamera. [SON10]

## 3. Roboterkamera

Die dritte verwendete Kamera ist eine sehr kleine CCD-FireWire-Kamera *Guppy F-080B/C* der Firma [ALL10], welche am Endeffektor des Roboters befestigt wurde. Sie wird aufgrund des beweglichen Greifarmes für die Identifizierung und Lokalisierung von Gegenständen verwendet. Für einen möglichst großen Sichtbereich sorgt ein Weitwinkelobjektiv *C60402KP - H416 (KP)* des Herstellers [PEN10]. Aufgrund des Weitwinkelobjektivs sind die Aufnahmen verzerrt. Diese Verzerrung ist mit den ermittelten radialen und tangentialen Verzerrungskoeffizienten (Vgl. S. 32) herauszurechnen. Das Objektiv wiegt im Gegensatz zur 50 g schweren Kamera 120 g und verfügt über eine Brennweite von 4,2 mm.

Mit Hilfe des ortsgebundenen Roboterarmes kann die Kamera in einem Radius von 60 cm bewegt werden. Dabei können pro Sekunde bis zu 30 Grauwertaufnahmen, mit einer maximalen Auflösung von 1032 x 778 Pixeln, aufgenommen werden. Die Abbildung 4.11 zeigt die dritte verwendete Kamera.



**Abbildung 4.11:** Am Katana-Greifarm befestigte FireWire-Kamera.

Die Kalibrierungsergebnisse der Kameras und damit die ermittelten intrinsischen Kameraparameter sind im Kapitel 5.1.3 auf

- S. 31 - Drehtellerkamera *DMK 41BF02*
- S. 32 - Sony Ericsson C905 Handykamera
- S. 32 - Guppy F-080B/C am Robotergreifarm

zu finden.

## 4.5 Drehmesstisch

Der verwendete Drehmesstisch besteht aus einem ansteuerbaren Drehteller, der einen zu vermessenden Gegenstand trägt und rotiert. Um die Achsenansteuerung des Drehtellers umzusetzen, wurde zusätzlich eine Positionierungssteuerung installiert und konfiguriert. In dieser Arbeit kam die Positionierungssteuerung *PS 90* in Kombination eines Drehmesstisches *DMT65+SM240* der Firma [OWI10] zum Einsatz.

Wie schon bereits Kapitel 4.4 erwähnt wurde, wird der Drehmesstisch für die 3D-Modellgenerierung (Siehe Kapitel 5) bevorzugt verwendet.



(a) Ansteuerbarer Drehteller



(b) Positionierungs-  
steuerung PS  
90

**Abbildung 4.12:** Drehmesstisch in Kombination mit einer Positionierungssteuerung zur Ansteuerung eines Drehtellers.

# 5 Generierung eines 3D-Merkmalmodells

In diesem Kapitel wird die Generierung eines 3D-Merkmalmodells von einem Gegenstand beschrieben. Dieses wird für die Objekterkennung und -lokalisierung benötigt und basiert auf perspektivisch unterschiedlichen Kameraaufnahmen. Einmalig muss dafür eine 3D-Oberflächenrekonstruktion des Gegenstandes erfolgen, wobei jedem Oberflächenpunkt gegenstands-spezifische Merkmale zugeordnet werden. Dafür sind möglichst viele perspektivisch unterschiedliche Kameraaufnahmen aufzunehmen. In den Aufnahmen werden Merkmale gesucht, welche in anderen Aufnahmen zuverlässig wiedererkannt werden können. Dies ermöglicht die Suche nach Merkmalskorrespondenzen zwischen Bildern, wodurch die 3D-Rekonstruktion erfolgt. Auf Basis des resultierenden 3D-Merkmalmodells (Oberflächenpunkte + Merkmale) kann ein Gegenstand in einem beliebigen Bild identifiziert und seine räumliche Lage und Orientierung bestimmt werden.

Die nachfolgenden vier Unterkapitel beschreiben die Generierung eines 3D-Merkmalmodells ausführlich.

## Inhalt

---

5.1	Kamerakalibrierung . . . . .	26
5.1.1	Intrinsische Kameraparameter . . . . .	27
5.1.2	Ermittlung der intrinsischen Kameraparameter . . . . .	30
5.1.3	Kalibrierergebnisse . . . . .	31
5.2	Merkmalsextraktion . . . . .	33
5.2.1	SIFT . . . . .	34
5.2.2	SURF . . . . .	41
5.3	Initiale Bestimmung von Weltpunkten . . . . .	42
5.3.1	Suche nach Merkmalskorrespondenzen . . . . .	43
5.3.2	Filterung . . . . .	47
5.3.3	Berechnung der 3D-Weltpunkte . . . . .	51
5.3.4	Anlegen eines 3D-Merkmalmodells . . . . .	62
5.4	Weltpunkte aus weiteren Aufnahmen . . . . .	63
5.4.1	Suche nach Merkmalskorrespondenzen . . . . .	65
5.4.2	Filterung . . . . .	66
5.4.3	Berechnung der 3D-Weltpunkte . . . . .	66
5.4.4	Erweiterung des 3D-Merkmalmodells . . . . .	69

---

## 5.1 Kamerakalibrierung

Das in dieser Arbeit verwendete Verfahren zur Generierung von 3D-Merkmalmodellen basiert auf den bekannten intrinsischen Kameraparametern. Oftmals sind diese durch den Hersteller bekannt. Aufgrund der abweichenden Kameraparameter innerhalb der gleichen Baureihe, ist eine Kalibrierung dennoch notwendig. Aus diesem Grund wird mit einer Kalibrierung sichergestellt, dass die intrinsischen Kameraparameter spätere Berechnungen nicht verfälschen.

### 5.1.1 Intrinsische Kameraparameter

Die intrinsischen Parameter beschreiben die interne Geometrie einer Kamera und somit Eigenschaften wie

- Bildweite  $\mathbf{f} = [f_x, f_y]$
- Verschiebung des Bildhauptpunktes  $\Delta\mathbf{h} = [\Delta h_x, \Delta h_y]$
- Scherungskoeffizient  $s$
- Linsenverzeichnung  $\mathbf{k}_{1 \times 5}$

Die Bildweite (engl. *focal length*) definiert den Abstand der Optik zur Sensorfläche.<sup>4</sup> Sie enthält einen horizontalen und vertikalen Pixelwert. In dieser Arbeit besitzt die Bildweite die Einheit *Pixel*. Vom Hersteller wird diese üblicherweise in mm oder  $\mu\text{m}$  angegeben. Der Zusammenhang ist nachfolgend in Gleichung 5.1 beschrieben.

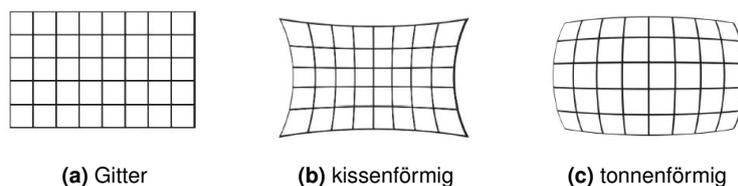
$$\mathbf{f}_{[\text{px}]} = \mathbf{f}_{[\text{mm}]} \rightarrow \mathbf{f}_{[\text{px}]} = \frac{\text{Bildma\ss e}_{[\text{px}]}}{\text{Sensorma\ss e}_{[\text{mm}]}} \cdot \mathbf{f}_{[\text{mm}]} \quad (5.1)$$

Die Abbildung der Bildweite von Millimeter nach Pixel setzt eine bekannte Bildweite in Millimeter sowie die Bild- und Sensorma\ss e voraus. Das sogenannte Seitenverh\u00e4ltnis<sup>5</sup> ( $f_x / f_y$ ) der Pixel wird als *aspect ratio* bezeichnet. Resultiert ein Verh\u00e4ltnis ungleich eins, dann ist der Bildsensor nicht quadratisch.

Der Durchsto\ss punkt der optischen Achse durch die Bildebene ergibt den Bildhauptpunkt (engl. *principal point*). Dieser sollte in der Regel in der N\u00e4he des Bildmittelpunktes liegen. Bei einer Kameraaufl\u00f6sung von 640 x 480 Pixel m\u00fcsste sich der Bildhauptpunkt etwa an der Position  $[x, y] = [320, 240]$  befinden.

Der Scherungskoeffizient  $s$  (engl. *skew coefficient*) definiert den Winkel zwischen der  $x$ - und  $y$ -Sensorachse. Ein Scherungskoeffizient von null bedeutet, dass rechteckige Pixel vorliegen.

Die Linsenverzeichnung  $\mathbf{k}_{1 \times 5}$  beschreibt in dieser Arbeit die radialsymmetrische und tangentiale Verzerrung der Optik. Diese setzt sich aus drei radialen- und zwei tangentialen Koeffizienten zusammen. Die radialsymmetrische Linsenverzeichnung entsteht durch Lichtstrahlenablenkung und -brechung an der Linse des Objektivs. Die Intensit\u00e4t der radialsymmetrischen Verzeichnung steigt je weitwinkliger ein Objektiv ist und nimmt ausgehend vom Bildhauptpunkt zum Bildrand zu. Es wird zwischen zwei Verzeichnungsarten unterschieden. Verl\u00e4uft die Kr\u00fcmmung zur Bildmitte hin, wird von einer kissenf\u00f6rmigen Verzeichnung gesprochen. Verl\u00e4uft die Kr\u00fcmmung dagegen von der Bildmitte weg, ist von einer tonnenf\u00f6rmigen Verzeichnung die Rede. Die nachfolgende Abbildung 5.1 stellt diese beiden Verzeichnungstypen dar.



**Abbildung 5.1:** Verzeichnungsarten einer Kamera. [GRA07]

Ist der Bildsensor nicht exakt orthogonal zur optischen Achse ausgerichtet, entsteht eine tangentiale Linsenverzeichnung. Dies hat eine Scherung des Bildes zur Folge. Bei der tangentialen Verzeichnung ist zwischen der vertikalen und horizontalen Verzeichnung zu unterscheiden.

Nachfolgend werden die Matrizen zur Ver- und Entzerrung der Bildpunkte hergeleitet.

<sup>4</sup>Die sogenannte Brennweite definiert dagegen den Abstand der Linsenhauptebene zum Brennpunkt.

<sup>5</sup>Breiten/H\u00f6hen -Verh\u00e4ltnis des Bildaufnehmers.

Die Zentralprojektion (Grafik 5.2) beschreibt die Abbildung eines Raumpunktes  $[X, Y, Z]^T$  auf einen Bildpunkt  $[x, y]^T$ .

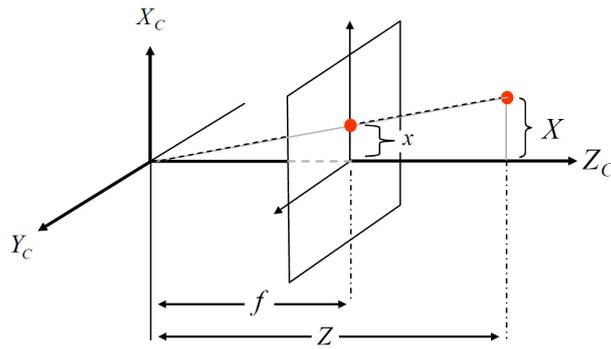


Abbildung 5.2: Zentralprojektion. [MEI08A]

Mit Hilfe des Strahlensatzes

$$\frac{X}{Z} = \frac{x}{f_x} \quad (5.2)$$

$$\frac{Y}{Z} = \frac{y}{f_y} \quad (5.3)$$

kann die Abbildung beschrieben werden.

Eine Umstellung der Gleichungen 5.2 und 5.3 nach  $x$  und  $y$  führt zusammengefasst zur Gleichung 5.4.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_x \cdot \frac{X}{Z} \\ f_y \cdot \frac{Y}{Z} \end{bmatrix} \quad (5.4)$$

Mit der Berücksichtigung des Bildhauptpunktes  $[h_x, h_y]^T$ , führt dies zur nachfolgend dargestellten Transformation (Gleichung 5.5).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f_x \cdot \frac{X}{Z} + h_x \\ f_y \cdot \frac{Y}{Z} + h_y \end{bmatrix} \quad (5.5)$$

Zusammengefasst unter Verwendung des Scherungsfaktors  $s$ , ergibt sich die in Gleichung 5.6 beschriebene 3 x 3 Kamera- bzw. Kalibriermatrix  $\mathbf{K} = [\mathbf{K}_x, \mathbf{K}_y, 1]^T$ . Diese enthält die intrinsischen Kameraparameter, mit denen Koordinaten vom Kamera- zum Bildkoordinatensystem abgebildet werden können.

$$\begin{bmatrix} \mathbf{K}_x \\ \mathbf{K}_y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & f_x \cdot s & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

Die Verzerrung von Bildpixeln bzw. -punkten  $[x, y]^T$  ist durch die Gleichung 5.8 beschrieben. Diese gilt unter der Annahme  $r^2 = x^2 + y^2$  und  $[\Delta x, \Delta y]^T$  sei die tangentielle Linsenverzeichnung, welche sich nach Gleichung 5.7 berechnen lässt.

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 2 \cdot k_3 \cdot x \cdot y + k_4 \cdot [r^2 + 2 \cdot x^2] \\ 2 \cdot k_4 \cdot x \cdot y + k_3 \cdot [r^2 + 2 \cdot y^2] \end{bmatrix} \quad (5.7)$$

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = [1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_5 \cdot r^6] \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (5.8)$$

Für die Entzerrung von Bildpunkten folgt der Umkehrschluss. Zu Beginn muss der Hauptpunkt  $\mathbf{h} = [h_x, h_y]^T$  vom verzerrten Bildpunkt  $[x_v, y_v]^T$  subtrahiert werden (Formel 5.9).

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} - \begin{bmatrix} h_x \\ h_y \end{bmatrix} \quad (5.9)$$

Anschließend ist durch die Bildweite  $\mathbf{f} = [f_x, f_y]^T$  zu dividieren und die Scherung  $s$  zu kompensieren. Mit diesen drei Schritten ergibt sich für die  $y$ -Koordinate

$$y_d = \frac{y_v - h_y}{f_y} \quad (5.10)$$

und für die  $x$ -Koordinate

$$x_d = \frac{x_v - h_x}{f_x} - s \cdot y_d \quad (5.11)$$

Die zwei Gleichungen 5.10 und 5.11 können auch zusammengefasst werden. Dies zeigt die nachfolgende Gleichung 5.12.

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} \frac{x_v - h_x}{f_x} - s \cdot \frac{y_v - h_y}{f_y} \\ \frac{y_v - h_y}{f_y} \end{bmatrix} \quad (5.12)$$

Im nächsten Schritt ist die Linsenverzeichnung zu kompensieren. Dafür kann beispielsweise eine iterative Vorgehensweise durchgeführt werden. Dabei ist  $[x_d, y_d]^T$  der Startwert  $[x, y]^T$  der Iteration. Diese besteht aus den fünf nachfolgenden Phasen:

1.  $r^2 = x^2 + y^2$

2. Radiale Linsenverzeichnung berechnen

$$k_{\text{radial}} = [1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_5 \cdot r^6]$$

3. Tangentiale Linsenverzeichnung berechnen

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 2 \cdot k_3 \cdot x \cdot y + k_4 \cdot [r^2 + 2 \cdot x^2] \\ 2 \cdot k_4 \cdot x \cdot y + k_3 \cdot [r^2 + 2 \cdot y^2] \end{bmatrix}$$

4. Tangentiale Linsenverzeichnung subtrahieren

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

5. Radiale Linsenverzeichnung dividieren

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{x}{k_{\text{radial}}} \\ \frac{y}{k_{\text{radial}}} \end{bmatrix}$$

In dieser Arbeit hat sich gezeigt, dass die Iteration 20-mal durchlaufen werden sollte.

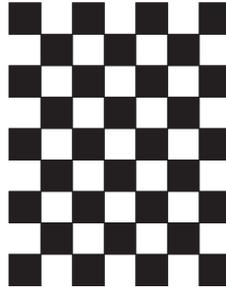
Abschließend müssen, unter der Berücksichtigung der Kamera-Kalibriermatrix  $\mathbf{K}$ , die entzerrten Koordinaten  $[x, y]^T$  mit  $\mathbf{K}$  multipliziert werden.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_x \\ \mathbf{K}_y \\ 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (5.13)$$

Die beschriebene Entzerrungsmethode wird in dem Kapitel 5.3.3 (*Berechnung der 3D-Weltpunkte*) auf alle zu entzerrende Bildpunkte (Merkmale) angewendet.

### 5.1.2 Ermittlung der intrinsischen Kameraparameter

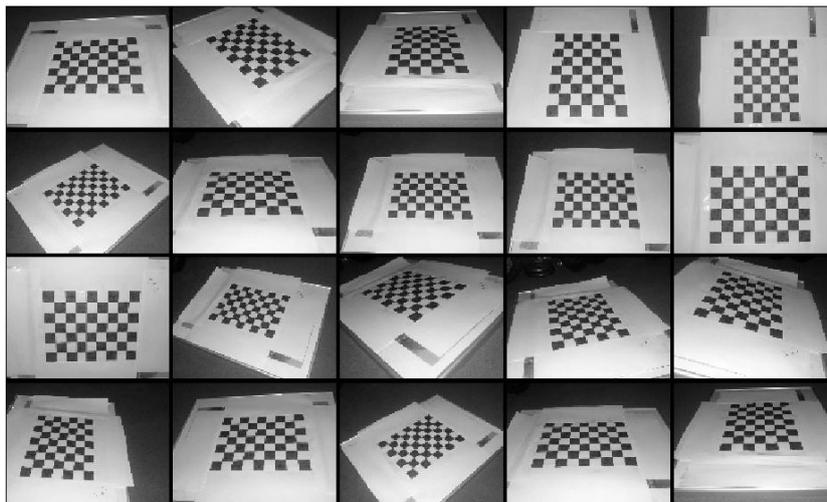
Dass in Matlab implementierte Kalibrierverfahren *Camera Calibration Toolbox* wurde in dieser Arbeit zur Kalibrierung verwendet. Es basiert auf einem bekannten Kalibriermuster, welches in der Abbildung 5.3 dargestellt ist. Das Kalibriermuster ist ein ebenes Schachbrettmuster bestehend aus 7 x 9 Quadraten. Jedes Quadrat besitzt eine Fläche von 30 mm x 30 mm.



**Abbildung 5.3:** Ebenes Kalibriermuster.

Die *Camera Calibration Toolbox* basiert auf der Kalibrierungsmethode von [ZHA00] und wurde von [BOU08] weiterentwickelt und in Matlab implementiert.

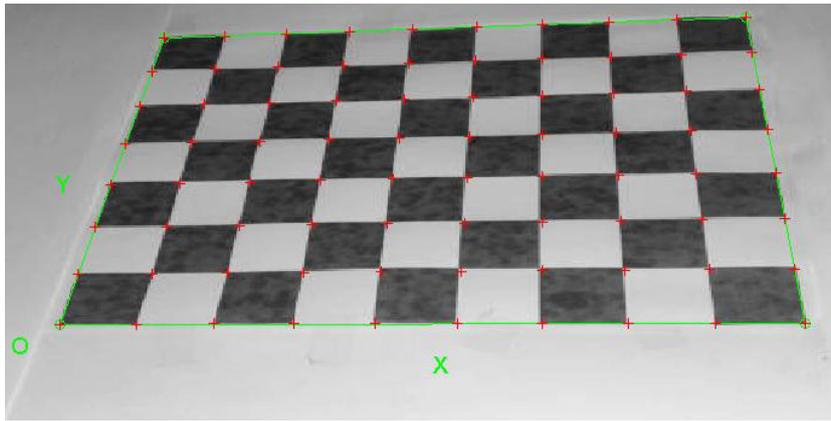
Vorbereitend wurde das Kalibriermuster 5.3 in naher Umgebung der Kamera positioniert. Für den Kalibriervorgang sind mindestens zwei- jedoch möglichst viele perspektivisch unterschiedliche Aufnahmen des Kalibrier-musters aufzunehmen. In diesem Fall wurden 20 Aufnahmen vom Kalibriermuster aufgenommen. Die resultierenden Aufnahmen der Größe 640 x 480 sind in der Abbildung 5.4 dargestellt.



**Abbildung 5.4:** 20 unterschiedliche Aufnahmen vom Kalibriermuster.

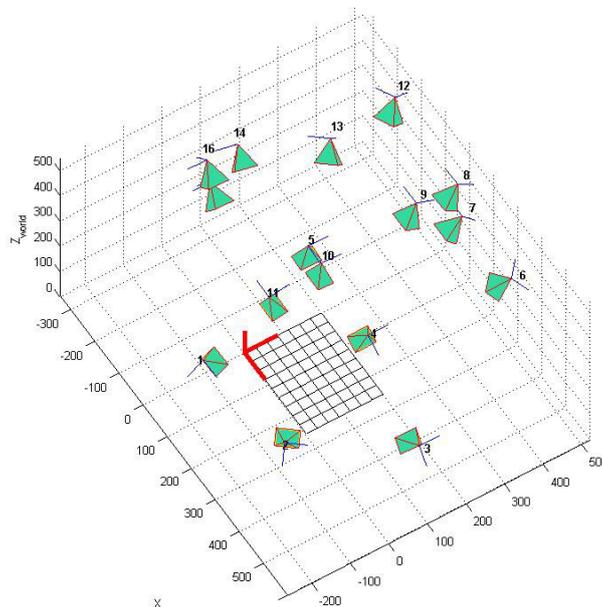
Nachdem 20 Aufnahmen des Schachbrettmusters aufgenommen und in der *Camera Calibration Toolbox* geladen sind, müssen die vier äußeren Eckpunkte des Schachbretts manuell ausgewählt werden. Der zu Beginn gesetzte Eckpunkt bildet den Ursprung eines rechtwinkligen Koordinatensystems. Dieser muss sich in allen Ansichten an der gleichen Ecke befinden. Anhand der bekannten Kästchengröße können somit die Eckpunkte aller Quadrate grob bestimmt werden. Anschließend erfolgt eine subpixelgenaue Bestimmung aller Eckpunkte.

Die nachfolgende Abbildung 5.5 zeigt die detektierten Eckpunkte eines Schachbrettmusters. Diese sind durch rote Kreuze gekennzeichnet. Die vier grün umkreisten Eckpunkte spannen den Bereich auf, indem die Suche nach den Eckpunkten stattfindet. Das grüne **O** symbolisiert den Ursprung von dem die *X* und *Y*-Achse des Koordinatensystems ausgehen.



**Abbildung 5.5:** Gefundene Eckpunkte des Schachbrettmusters.

Die bekannten Eckpunkte in allen Ansichten ermöglichen die Bestimmung der intrinsischen Kameraparameter. Die Darstellung der ursprünglichen Kamerapositionen und -ausrichtungen ist abschließend in Abbildung 5.6 veranschaulicht. Es signalisiert mit der korrekten perspektivischen Darstellung der 20 Kameras ein erfolgreiches Kalibrierergebnis.



**Abbildung 5.6:** Weltansicht der Kameraposen.

### 5.1.3 Kalibrierergebnisse

In dieser Arbeit wurden drei verschiedene Kameras verwendet. Dieses Kapitel beschreibt die Kalibrierergebnisse und damit die ermittelten intrinsischen Kameraparameter  $\mathbf{K}$ . Eine ausführliche Hardwarebeschreibung der Kameras ist im Kapitel 4 (*Technischer Aufbau*) ab Seite 19 zu finden.

Jede einzelne Kamera musste für das Verfahren, zur Generierung eines 3D-Merkmalmodells sowie zur Lokalisierung von Gegenständen, kalibriert werden. Als Kalibrieremethode fand das in Kapitel 5.1.2 beschriebene Verfahren der *Matlab Calibration Toolbox* Verwendung. Bei allen drei Kalibrierresultaten wurden zwanzig Aufnahmen eines Schachbrettmusters aus unterschiedlichen Perspektiven aufgenommen. Die horizontale-, vertikale Bildweite  $[f_x, f_y]$  und der Bildhauptpunkt  $[h_x, h_y]$  sind in Pixel angegeben. Der Scherungsfaktor  $s$  und die Linsenverzeichnung  $\mathbf{k}_{1 \times 5}$  sind Koeffizienten. Alle Angaben wurden auf zwei Nachkommastellen gerundet.

Eine Sony Ericsson C905 Handykamera kommt für flexible Aufnahmen im Freien zum Einsatz. Die nachfolgende Tabelle 5.1 beinhaltet alle ermittelten intrinsischen Kameraparameter.

(Intrinsischer) Parameter	Bezeichner	Wert	Abweichung
Auflösung	$[a_x, a_y]$	[640 , 320]	
Bildweite	$[f_x, f_y]$	[675.75 , 675.64]	$\pm [4.04 , 4.08]$
Bildhauptpunkt	$[h_x, h_y]$	[314.58 , 237.42]	$\pm [3.22 , 4.19]$
Scherungskoeffizient	$s$	[0.00]	$\pm [0.00]$
Verzeichnung	$\mathbf{k}_{1 \times 5}$	[0.23 , -1.01 , 0.00 , 0.00 , 0.00]	$\pm [0.023 , 0.15 , 0.00 , 0.00 , 0.00]$
Pixelfehler	$[e_x, e_y]$	[0.17 , 0.44]	

**Tabelle 5.1:** Intrinsische Parameter der Sony Ericsson C905 Handykamera.

Zusätzlich zur Handykamera wird eine CCD monochrome Kamera *DMK 41BF02* in Kombination mit einem ansteuerbaren Drehteller, eingesetzt. Aufgrund des Drehtellers kommt sie für gleichmäßig rotierte Gegenstandsaufnahmen zum Einsatz. Diese Aufnahmen erzielen bei der Generierung der 3D-Merkmalmodelle sehr gute Ergebnisse (Siehe Kapitel 8). Als Kameraobjektiv wird ein hochauflösendes Objektiv *H0514-MP* verwendet. Die Abbildung 4.9 veranschaulicht die CCD monochrome Kamera in Kombination eines per Schrittmotor ansteuerbaren Drehtellers.

Nachfolgend sind in Tabelle 5.2 die Kalibrierergebnisse aufgeführt.

(Intrinsischer) Parameter	Bezeichner	Wert	Abweichung
Auflösung	$[a_x, a_y]$	[1280 , 960]	
Bildweite	$[f_x, f_y]$	[1072.42 , 1076.41]	$\pm [13.06 , 11.80]$
Bildhauptpunkt	$[h_x, h_y]$	[675.61 , 549.75]	$\pm [8.09 , 11.83]$
Scherungskoeffizient	$s$	[0.00]	$\pm [0.00]$
Verzeichnung	$\mathbf{k}_{1 \times 5}$	[-0.15 , 0.18 , 0.01 , 0.00 , 0.00]	$\pm [0.01 , 0.04 , 0.00 , 0.02 , 0.00]$
Pixelfehler	$[e_x, e_y]$	[0.61 , 0.64]	

**Tabelle 5.2:** Intrinsische Parameter der Drehtellerkamera.

Bei der dritten verwendeten Kamera handelt es sich um eine sehr kleine FireWire-Kamera (Abbildung 4.11). Diese ist mit einem Weitwinkelobjektiv *C60402KP - H416 (KP)* ausgestattet. Die Kamera ist an einem Knickarmroboter (Vgl. *Technischer Aufbau*, Kapitel 4.1) in der Nähe des Winkelgreifers montiert. Mit Hilfe des ortsbundenen Roboterarmes kann die Kamera in einem Radius von circa 60 cm bewegt werden.

Aufgrund des Weitwinkelobjektivs sind die Aufnahmen bei dieser Kamera verzerrt. Diese tonnenförmige Verzeichnung muss rausgerechnet werden. Dazu müssen die auf Seite 29 angegebenen Formeln 5.9 bis 5.13, zur Korrektur der Linsenverzeichnung, verwendet werden. Nachfolgend sind in Tabelle 5.3 die Ergebnisse der Kamerakalibrierung aufgelistet.

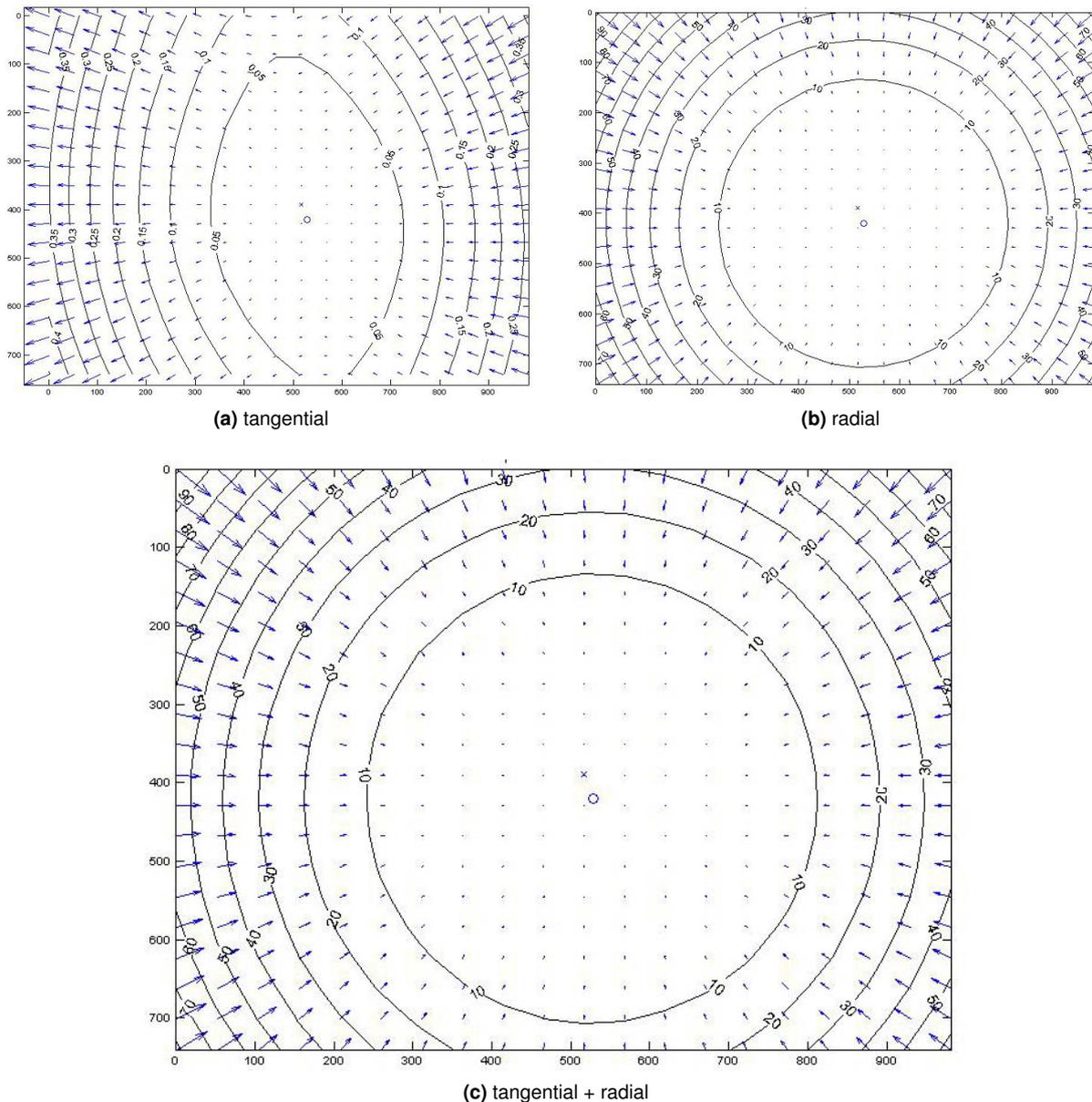
(Intrinsischer) Parameter	Bezeichner	Wert	Abweichung
Auflösung	$[a_x, a_y]$	[1032 , 778]	
Bildweite	$[f_x, f_y]$	[923.56 , 923.65]	$\pm [7.69 , 6.73]$
Bildhauptpunkt	$[h_x, h_y]$	[527.92 , 419.87]	$\pm [3.05 , 6.23]$
Scherungskoeffizient	$s$	[0.00]	$\pm [0.00]$
Verzeichnung	$\mathbf{k}_{1 \times 5}$	[-0.38 , 0.18 , 0.00 , 0.00 , 0.00]	$\pm [0.01 , 0.04 , 0.00 , 0.00 , 0.00]$
Pixelfehler	$[e_x, e_y]$	[0.21 , 0.23]	

**Tabelle 5.3:** Intrinsische Parameter der Katana-Roboterkamera.

Die Abbildung 5.7 veranschaulicht die tangentielle und radiale Verzeichnung. In dieser besitzt die  $x$ - und  $y$ -Achse Pixelwerte.

Die Genauigkeit der Kalibrierung hängt von der Anzahl der verwendeten Kalibrierungspunkte sowie deren Genauigkeit ab. Mindestens 48 Kalibrierungspunkte und deren subpixelgenauen Berechnung führten zu einer präzisen Kalibrierung der intrinsischen Kameraparameter.

Die intrinsischen Kameraparameter, welche in dem Kalibrierprozess ermittelt wurden, finden in dieser Arbeit in dem Kapitel 5.3.3 Gebrauch.



**Abbildung 5.7:** Tangentiale und radiale Verzeichnung der Roboterkamera.

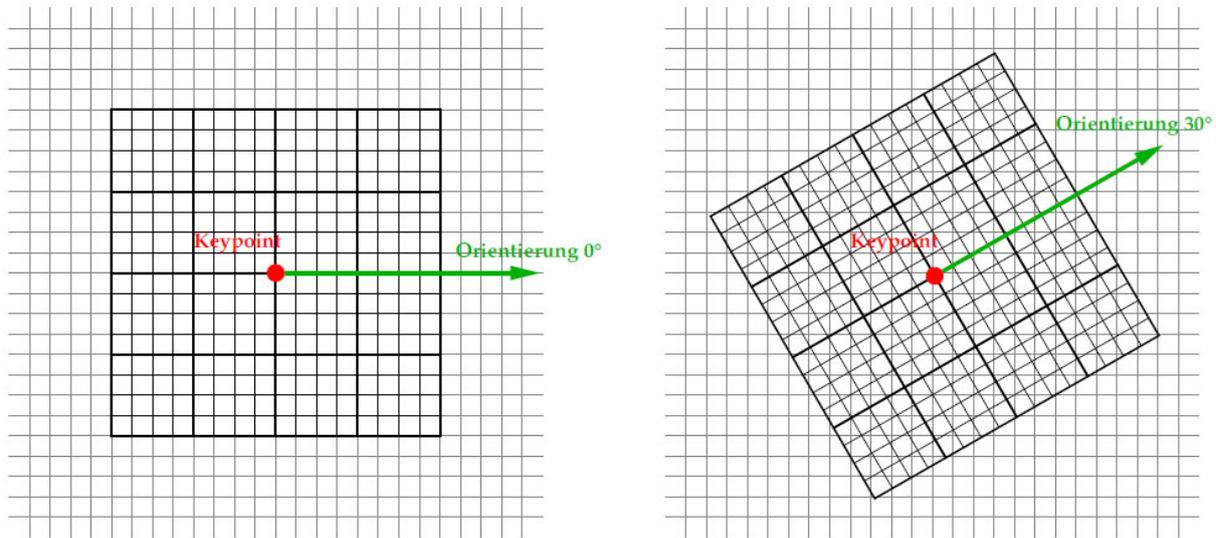
Nachträgliche Änderungen der Kameraeigenschaften wie Auflösung oder Zoom führen, aufgrund veränderter intrinsischer Kameraparameter, zu einer zwangsweisen Neukalibrierung.

## 5.2 Merkmalsextraktion

Das Verfahren zur 3D-Modellgenerierung basiert auf Kameraaufnahmen, welche aus unterschiedlichen Perspektiven aufgenommen werden. Um eine Beziehung zwischen diesen Aufnahmen zu beschreiben, sind korrespondierende Merkmale zwischen den Aufnahmen zu finden. Der Rotations-, Translations-, Skalierungs- und Verzerrungsgrad beeinflusst unter anderem die Wiedererkennung von Merkmalen in unterschiedlichen Aufnahmen. Aufgrund dessen sollten die Merkmale einen hohen Grad an Robustheit gegenüber den genannten Eigenschaften aufweisen.

Ein Merkmal wird durch seine Bildposition  $[x, y]$ , seine Hauptorientierung und einen Merkmalsvektor gekennzeichnet. Die Hauptorientierung definiert die Merkmalsregion. Der Merkmalsvektor beschreibt diese Umgebung.

Er beinhaltet somit Erkenntnisse über sein Umfeld. Berechnet wird der Merkmalsvektor relativ zur Hauptorientierung. Wird ein Bild beispielsweise rotiert, dann haben übereinstimmende Merkmale den gleichen Merkmalsvektor aber eine unterschiedliche Hauptorientierung. Nachfolgend ist in Abbildung 5.8 die Hauptorientierung eines Merkmales mit seinen 16 x 16 Umgebungspixeln<sup>6</sup> dargestellt.



**Abbildung 5.8:** Rotation einer 16 x 16 Pixel großen Merkmalsumgebung anhand seiner Hauptorientierung. [ZIM07]

Der Merkmalsvektor dient zum Vergleich<sup>7</sup> von Merkmalen und bringt eine Invarianz bezüglich Helligkeits- und Kontrastveränderungen mit sich. Er wird auch als sogenannter Deskriptor bezeichnet.

Zwei geeignete Verfahren zur Merkmalsextraktion werden im Nachfolgenden vorgestellt.

### 5.2.1 SIFT

Das Verfahren zur skaleninvarianten Merkmalsextraktion (engl. Scale-Invariant Feature Transform, kurz SIFT) wurde erstmals 1999 von LOWE<sup>8</sup> veröffentlicht. Dieses Verfahren ermöglicht das Auffinden von invarianten Merkmalen bezüglich

- Rotation
- Translation
- Skalierung
- Änderung der Lichtverhältnisse
- Teilweiser affiner Verzerrung

in Bildern. Auf Basis dieser Merkmale kann eine zuverlässige Beziehung zwischen perspektivisch unterschiedlichen Aufnahmen beschrieben werden.

<sup>6</sup>Aufgrund der Rotation müssen gegebenenfalls Pixel, welche sich nach der Drehung zwischen zwei oder mehreren Pixelfeldern befinden, interpoliert werden.

<sup>7</sup>Zwei Merkmale aus verschiedenen Aufnahmen können unter anderem mit der euklidischen Distanz ihrer Merkmalsvektoren verglichen werden.

<sup>8</sup>Veröffentlichung des SIFT-Algorithmus: [Low99]  
Weiterentwicklungen sind in [Low01], [BL02] und [Low04] beschrieben.

Die Arbeitsweise des SIFT-Algorithmus gliedert sich in vier Schritte. Im ersten Schritt sind anhand Extrema im Skalenraum potentielle skalierungsinvariante Merkmale zu ermitteln. Anschließend erfolgt durch eine Stabilitätsuntersuchung eine Filterung der Merkmalskandidaten. Für alle verbliebenen Merkmalskandidaten wird dabei die exakte Position bestimmt. Im dritten Schritt ist zu jedem potentiellen Merkmalspunkt eine Hauptorientierung zu berechnen. Eine Hauptorientierung ist für die Rotationsinvarianz von Merkmalen von Nöten. Zur vollständigen Beschreibung eines Merkmales wird im letzten Schritt ein 128-dimensionaler Merkmalsvektor ermittelt. Dieser dient zum Vergleich von Merkmalen.

Nachfolgend werden die vier Phasen des SIFT-Algorithmus ausführlich beschrieben.

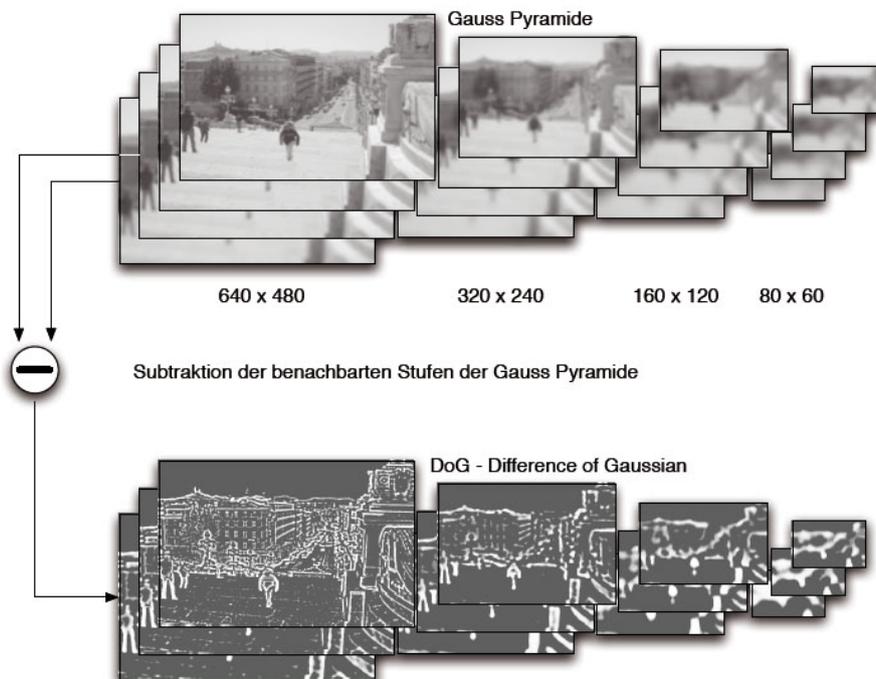
### 5.2.1.1 Ermittlung potentieller Merkmale

Die Grafik 5.9 veranschaulicht die Suche nach robusten Merkmalskandidaten. Auf der oberen Seite sind von links nach rechts vier Gauß-Pyramiden dargestellt. Eine Pyramide besteht aus fortlaufend stärker geglätteten Abbildungen des zu analysierenden Bildes **B**. Dabei kommt ein Gauß-Filter  $G_{\sigma}(x,y)$  aufgrund seiner Invarianz gegenüber Skalierung zum Einsatz. Unter Verwendung von unterschiedlichen Varianzen<sup>9</sup>  $\sigma$  ergibt sich eine in der Gleichung 5.14 angegebene Faltungsgleichung, welche auf das zu untersuchende Bild **B** angewendet wird.

$$B(x,y) * G_{\sigma}(x,y) = B(x,y) * \left( \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \right) \quad (5.14)$$

Dabei ist  $B(x,y)$  eine Funktion, die den Helligkeitswert an der Bildposition  $[x,y]$  des Bildes **B** wiedergibt.

Jeweils zwei benachbarte Abbildungen der Gauß-Pyramide werden im Anschluss der Glättung subtrahiert. Diese Subtraktion wird als *Difference of Gaussian* (DoG) bezeichnet. Die untere Seite der Abbildung 5.9 stellt die Differenzbilder dar.



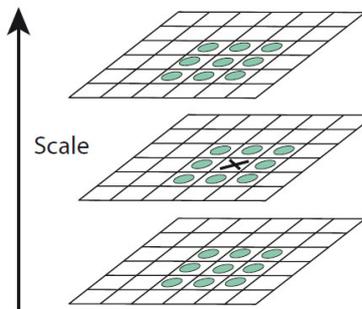
**Abbildung 5.9:** Ermittlung der Extrema im Skalenraum. [HEY05]

Dieser beschriebene Vorgang wird  $n$ -mal mit jeweils um 50 % verkleinerten Abbildungen durchgeführt. Dabei ist als Ausgangsbild jeweils die letzte Abbildung aus der vorherigen Pyramide zu verwenden. Die somit entstehende Folge von unterschiedlichen Auflösungen eines Bildes bauen einen sogenannten *Skalenraum* auf. Die

<sup>9</sup>Die Varianz ist ein Maß für die Streuung bezüglich einer Messgröße.

Bildverkleinerung erfolgt, indem jeder zweite Pixel in  $x$ - und  $y$ -Richtung verwendet wird. Aufgrund der vorherigen Gaußglättung beinhaltet jeder Pixel Details von seinen Nachbarpixeln. Dadurch muss beim Verkleinern der Bilder keine zusätzliche Pixelinterpolation durchgeführt werden.

In der letzten Phase werden die DoG-Pyramiden auf minimale und maximale Pixelwerte untersucht. Ein Maximum ist gefunden, wenn alle der 26 benachbarten Pixel kleiner als er selbst sind. Die 26 Nachbarn ergeben sich aus den acht Nachbarn derselben Ebene und den jeweils neun Nachbarn der niedrigeren und höheren Ebene. Ein Minimum ist im umgekehrten Fall gefunden. Die Grafik 5.10 veranschaulicht die Nachbarschaft.



**Abbildung 5.10:** Potentieller Merkmalspixel wird mit seinen 26 Nachbarn verglichen. [Low04]

### 5.2.1.2 Filterung und Lokalisierung potentieller Merkmalspunkte

Alle im ersten Schritt ermittelten Extremwerte  $L(x,y)$  sind Merkmalskandidaten. Im zweiten Schritt sind diese in einer Stabilitätsanalyse genauer zu untersuchen. Erfüllt ein Merkmalskandidat die festgelegten Stabilitätsbedingungen, ist seine exakte Position und Skalierung zu bestimmen. Andererseits ist der Merkmalskandidat nicht weiter von Bedeutung und wird verworfen. Stabilitätsmerkmale sind unter anderem Kontrast (Rauschunterdrückung), Klassifikation in Ecken, homogene Regionen und Kanten.

Ein Kantenpunkt kann anhand seiner zwei Hauptkrümmungen erkannt werden. Jeder Kantenpunkt besitzt eine große Hauptkrümmung quer über die Kante. Die zweite Hauptkrümmung verläuft entlang der Kante und ist deswegen sehr klein. Mit dem Verhältnis zwischen den beiden Hauptkrümmungen wird anschließend ermittelt, ob sich ein Punkt auf einer Kante befindet. Mit Hilfe einer  $2 \times 2$  Hesse-Matrix lässt sich das Verhältnis der Hauptkrümmungen berechnen.

Die Hesse-Matrix  $\mathbf{H}$  ist in der nachfolgenden Gleichung 5.18 dargestellt.

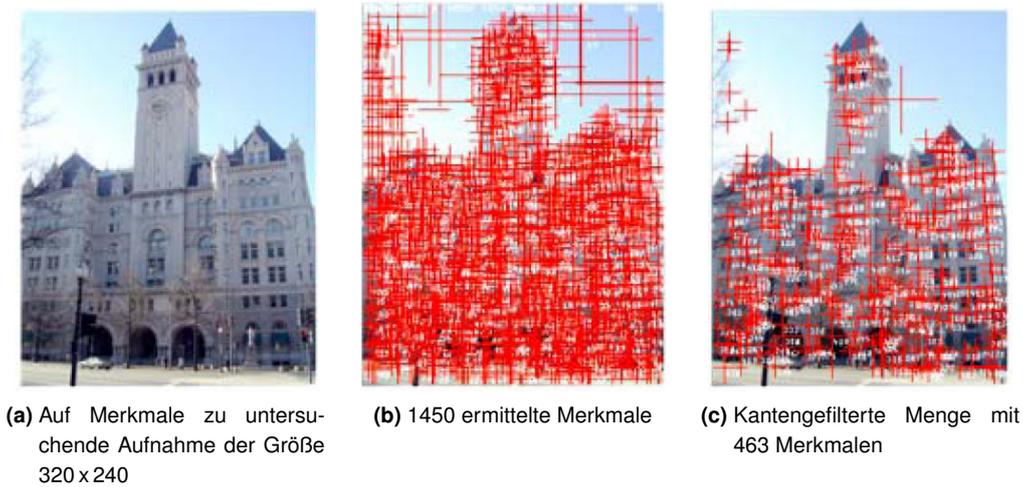
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (5.15)$$

In dieser gibt  $D_{xx}$ ,  $D_{xy}$  und  $D_{yy}$  die zweifache Ableitung des Differenzbildes in der jeweiligen  $x/y$ -Richtung an.

Näheres ist in [ZIM07] ab Seite 27 beschrieben und gegebenenfalls zu entnehmen.

Ein Beispiel einer mit einem Kantenfilter angewandten Merkmalsfilterung, ist in der nachfolgenden Abbildung 5.11 dargestellt. Ausgehend eines Eingangsbildes (a) wurde dieses auf potentielle robuste Merkmale analysiert. Auf Basis der extrahierten Merkmalsmenge (b) konnten mit Hilfe eines Kantenfilters, Merkmale die sich auf Kantenregionen befanden, entfernt werden. Das rechte Bild (c) der Abbildung veranschaulicht die reduzierte Merkmalsmenge.

Dieser Fall ist ein gutes Beispiel für die gebräuchliche Anwendung eines Kantenfilters. Kommt keine Kantenfilterung zum Einsatz, dann ist die Wiedererkennung eines Kantenpunktes in einer anderen Aufnahme wahrscheinlich nicht möglich. Der Grund hierfür ist, dass Punkte, welche sich auf derselben Kante befinden, oftmals die gleichen Merkmalseigenschaften haben und damit nicht zu unterscheiden sind.



**Abbildung 5.11:** Filterung von Merkmalen, welche sich auf Kanten befinden. [HEY05]

### 5.2.1.3 Bestimmung der Hauptorientierungen

Die Hauptorientierung eines Merkmalspunktes kann mit den umliegenden Bildgradienten seines gaußgefilterten Bildes bestimmt werden. Dabei ist das gaußgefilterte Bild, welches der Skalierung des Merkmalspunktes am nächsten kommt, zu verwenden. Ein Gradienten kennzeichnet die Richtung des größten Helligkeitsanstiegs und wird als Vektor mit seiner Länge  $m(x,y)$  und Orientierung  $\theta(x,y)$  angegeben. Die Berechnung eines Gradienten bezüglich eines Punktes  $L(x,y)$ , ist nachfolgend in den Gleichungen 5.16 und 5.17 beschrieben.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (5.16)$$

$$\theta(x,y) = \tan^{-1} \frac{L(x+1,y) - L(x-1,y)}{L(x,y+1) - L(x,y-1)} \quad (5.17)$$

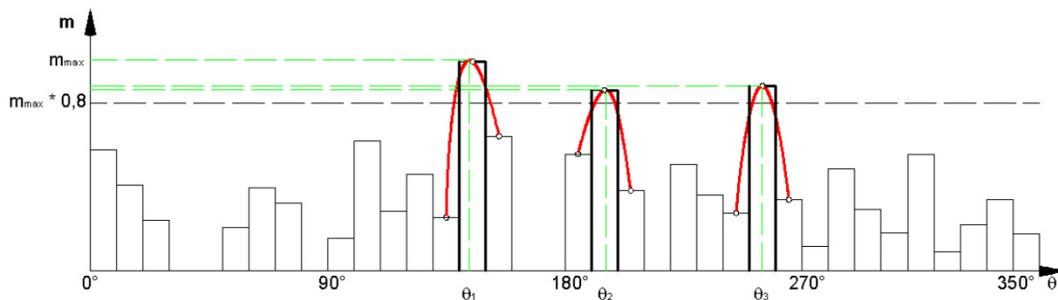
Für die Ermittlung der Merkmalsorientierung sind alle Gradienten in seiner näheren Umgebung zu berechnen. Anschließend sind die ermittelten Gradientenorientierungen  $\theta_i$  zu gewichten. Die erste Gewichtung erfolgt mit der jeweiligen Gradientenlänge  $m_i$ . Das hat zur Folge, dass Gradienten mit einer größeren Länge stärker bewertet werden. Die zweite Gewichtung erfolgt in Abhängigkeit eines Gewichtungsfaktors  $\alpha$ . Dieser bewirkt, dass äußere Bereiche der 360°-Umgebung gedämpft einbezogen werden.

Nachfolgend ist ein Histogramm anhand der gewichteten Orientierungen  $\theta_{g_i}$  und den jeweiligen Gradientenlängen aufzubauen. Das entstehende Orientierungshistogramm ist in 36 Winkelbereiche unterteilt und hat somit eine Klassenbreite von 10°. Alle Gradientenlängen sind anhand ihrer gewichteten Orientierung an der passenden Stelle im Histogramm aufzuaddieren.

Ist das Histogramm vollständig aufgebaut, kann die Hauptorientierung des Merkmales bestimmt werden. Dafür wird die größte Gradientenlänge  $m_{\max}$  aus dem Histogramm ermittelt. Die Hauptorientierung des Merkmales wird anschließend mit der Länge  $m_{\max}$  und dem dazugehörigen Orientierungswinkel  $\theta_i$  zusammengesetzt.

In dem gesonderten Fall, dass die Länge eines weiteren Gradienten mindestens achtzig Prozent des größten Betrags aufweist, ist ein weiterer Merkmalspunkt zu erzeugen. Dies führt aufgrund der Möglichkeit den Merkmalspunkt für mehrere Richtungen zu betrachten, zu einer stabilitätssteigernden Eigenschaft. Beispielhaft sind derartige Merkmale (gleiche Position + mehrere Orientierungen) an Ecken zu finden.

Die nachfolgend dargestellte Abbildung 5.12 veranschaulicht ein Orientierungshistogramm. Die  $x$ -Achse kennzeichnet die Richtung in Grad und besitzt eine Klassenbreite von 10°. Die  $y$ -Achse stellt die akkumulierten Gradientenlängen dar. Die größte Gradientenlänge bestimmt die Richtung des Merkmales. Des Weiteren wurden zwei zusätzliche Richtungen ermittelt und dem Merkmal zugeordnet.



**Abbildung 5.12:** Richtungsbestimmung eines Merkmales mit einem Orientierungshistogramm. [RON05]

Diese eindeutige Hauptorientierung eines Merkmales ist für die Rotationsinvarianz von Nöten.

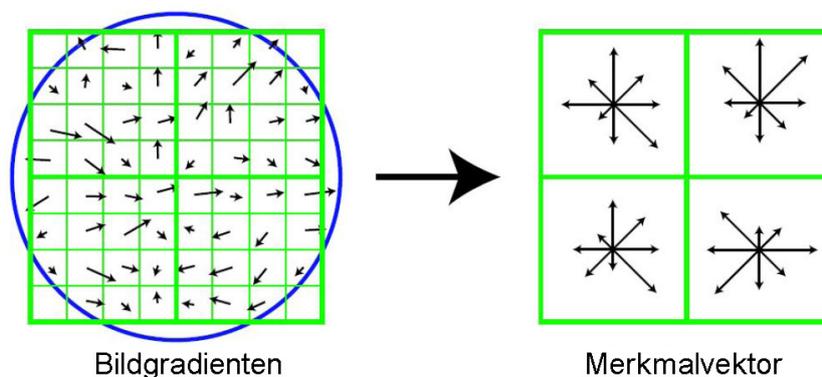
#### 5.2.1.4 Erzeugung der Merkmalsvektoren

Ein Merkmal wird durch seine Bildposition  $[x, y]$ , seiner Hauptorientierung und einem Merkmalsvektor gekennzeichnet. Ein 128-dimensionaler Merkmalsvektor ist die dritte und letzte fehlende Information, zur vollständigen Beschreibung eines robusten Merkmales. Anhand der Hauptorientierung, welche die Merkmalsregion definiert, beschreibt der Merkmalsvektor diese Umgebung. Er beinhaltet somit Erkenntnisse über sein Umfeld. Berechnet wird dieser relativ zur Hauptorientierung. Wird ein Bild beispielhaft rotiert, haben übereinstimmende Merkmale den gleichen Merkmalsvektor aber eine unterschiedliche Hauptorientierung. In der Abbildung 5.8 auf Seite 34 wurde die Hauptorientierung eines Merkmales mit seinen  $16 \times 16$  Umgebungspixeln dargestellt. Aufgrund der Rotation müssen gegebenenfalls Pixel, welche sich nach der Drehung zwischen zwei oder mehreren Pixelfeldern befinden, interpoliert werden.

Der Merkmalsvektor dient somit zum Vergleich von Merkmalen und bringt die Invarianz bezüglich Helligkeits- und Kontrastveränderungen mit sich. Er wird auch als sogenannter *Deskriptor* bezeichnet.

LOWE<sup>10</sup> verwendet einen  $4 \times 4 \times 8 = 128$ -dimensionalen Deskriptor. Der Deskriptor setzt sich aus  $4 \times 4 = 16$  Orientierungshistogrammen, welche jeweils eine gleichgroße Fläche abdecken, zusammen. Jedes Histogramm wird ähnlich wie in Schritt 3 *Bestimmung der Hauptorientierungen* berechnet. Die Klassenbreite beträgt in diesem Fall  $45^\circ$ . Damit hat ein Histogramm  $360^\circ / 45^\circ = 8$  Winkelbereiche, welche jeweils  $45^\circ$  seiner Subregion abdecken.

Die Abbildung 5.13 veranschaulicht einen  $2 \times 2 \times 8$ -dimensionalen Deskriptor. Im linken Bild der Abbildung sind die ermittelten Gradienten dargestellt. Diese beschreiben das Umfeld des Merkmales. Der blaue Kreis kennzeichnet dabei die Gewichtung der Gradienten. Die rechte Seite zeigt einen aus  $2 \times 2 = 4$  Histogrammen bestehenden Deskriptor, wobei jedes Histogramm acht Richtungsbereiche beinhaltet. Die acht Richtungspfeile stellen anhand ihrer Länge und Orientierung die Stärke und den Richtungsverlauf in den jeweiligen  $45^\circ$ -Bereich dar.



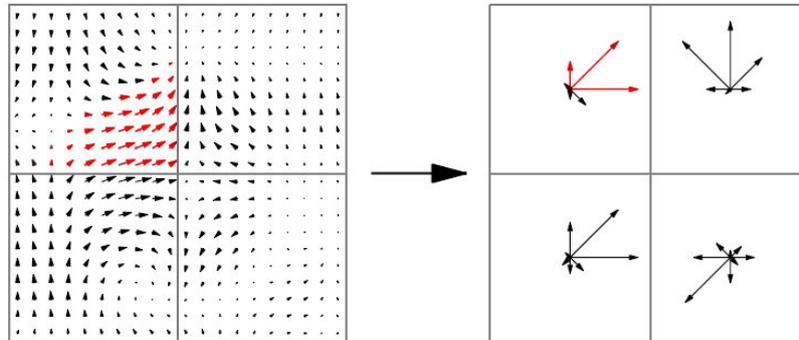
Bildgradienten

Merkmalsvektor

**Abbildung 5.13:**  $2 \times 2 \times 8$ -dimensionaler Deskriptor eines Merkmales. Dieser beschreibt seine  $8 \times 8$  Pixelumgebung, welche in vier Subregionen mit jeweils  $4 \times 4$  Pixeln aufgeteilt wurde. [BAR07]

<sup>10</sup>In [Low04] und [Low99] beschrieben.

In der nachfolgenden Abbildung 5.14 ist ein weiterer  $2 \times 2 \times 8$ -dimensionaler Deskriptor abgebildet. Dieser Deskriptor dient lediglich zur Veranschaulichung und ist aufgrund dessen eher unwahrscheinlich in der Praxis anzutreffen. Der abstrakte Deskriptor stellt anschaulich den Übergang zwischen den Gradienten der Merkmalsumgebung (links) und den vier Orientierungshistogrammen (rechts) dar. Die Umgebung des Merkmales ist dabei in vier Subregionen eingeteilt. Aus jeder Subregion ist ein Orientierungshistogramm mit jeweils acht Richtungseinträgen zu bestimmen. In der sich oben links befindenden ersten Subregion, zeigen viele große Gradienten nach oben rechts. Das aus diesen Gradienten berechnete Orientierungshistogramm, stellt anhand der stark nach oben rechts zeigenden Richtungspeile, den Zusammenhang dar.



**Abbildung 5.14:** Abstrakter  $2 \times 2 \times 8$ -dimensionaler Deskriptor. [SCH06]

Um eine Invarianz bezüglich Beleuchtungsveränderungen zu ermöglichen, wird der Gradientenbetrag des Deskriptors, indem er durch seine Länge dividiert wird, auf Einheitslänge normiert. Dies erfolgt, da eine Kontrastveränderung eine gleichmäßige Erhöhung aller Gradientenbeträge zur Folge hat, welche durch die Normalisierung aufgehoben wird.

Der gesamte Ablauf der Merkmalsextraktion ist in Abbildung 5.15 zusammenfassend dargestellt. Ein Eingangsbild ist auf Merkmale zu analysieren. (1a) Dazu wird das Eingangsbild fortlaufend geglättet. Es entsteht eine Gauß-Pyramide. Das letzte gaußgeglättete Bild wird anschließend um 50% runterskaliert und bildet das Eingangsbild für die nächste Gauß-Pyramide. Dieser Vorgang wird  $n$ -mal fortgeführt. (1b) Jeweils zwei benachbarte Bilder aus der Gauß-Pyramide werden miteinander subtrahiert. Die subtrahierten Bilder bauen DoG-Pyramiden auf. (1c) Jeder Punkt in der DoG-Pyramide wird auf einen Extrempunkt geprüft. (2) Anschließend wird jedes gefundene Maxima einer Stabilitätsanalyse unterzogen. (3) Anhand der Umgebungsgradienten eines Merkmals wird die Hauptorientierung des Merkmales bestimmt. (4) Deskriptoren beschreiben die Merkmalsumgebung und sind im letzten Schritt zu ermitteln.

Ein Objekt wird somit durch eine Menge von SIFT-Merkmalen beschrieben. Anhand dieser robusten Merkmale kann ein Gegenstand in einem rotierten, verschobenen, skalierten, teilweise affin verzerrten und unter einem anderen Lichtverhältnis aufgenommen Bild, wiedererkannt werden. Dies bildet die Grundlage zur 3D-Objekt- und Lageerkennung von Gegenständen.

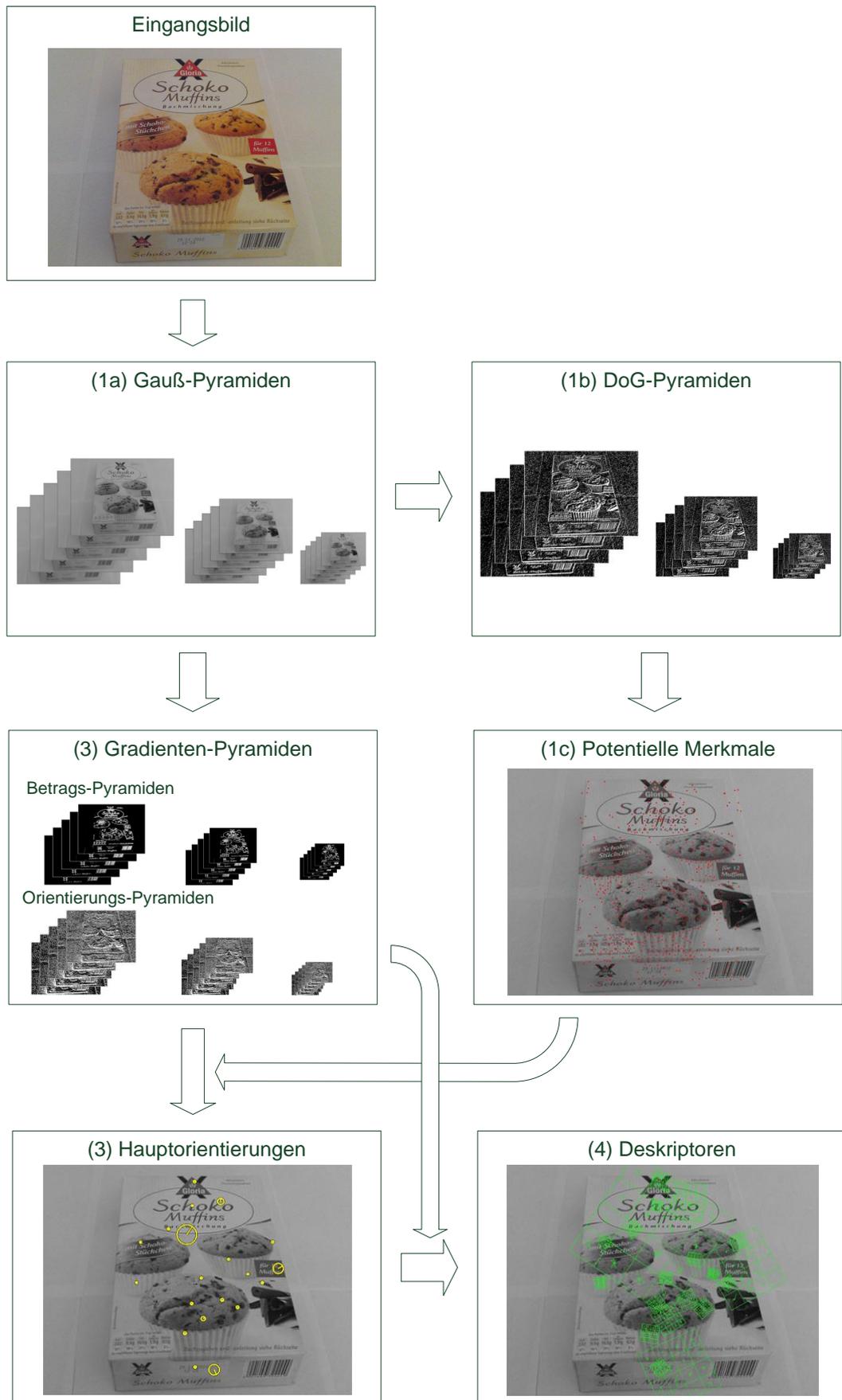


Abbildung 5.15: Ablauf einer Merkmalsextraktion.

## 5.2.2 SURF

SURF (engl. *Speeded-Up Robust Features*) wurde erstmals in [BTG06] veröffentlicht. SURF besitzt im Vergleich zu SIFT einen ähnlichen Algorithmus. Er hat jedoch einen deutlichen Geschwindigkeitsvorteil. Dieser wird durch eine effizientere Suche nach Merkmalskandidaten erzielt.

Ein Großteil der Geschwindigkeitssteigerung von SURF liegt an der Verwendung von Integral- anstelle von Intensitätsbildern. Bei einem Integralbild sind die Pixelsummen innerhalb eines rechteckigen Bildausschnittes zu berechnen. Für dieses Vorgehen ist die Bezeichnung *Summed Area Tables* gebräuchlich. Darüber hinaus ist bei der Verwendung von Integralbildern keine Unterabtastung notwendig.

SIFT sucht in DoG-Pyramiden<sup>11</sup> Merkmalskandidaten. Beim SURF-Algorithmus werden dagegen potentielle Merkmale mit Hilfe der Determinante der Hesse-Matrix zeiteffizient gesucht. Besitzt die Determinante der Hesse-Matrix an einer Bildposition ein lokales Maximum, dann ist ein Merkmalskandidat gefunden.

Die 2 x 2 Hesse-Matrix  $\mathbf{H}$  ist von dem Bildpunkt  $\mathbf{x} = [x, y]$  und der Skalierung  $\sigma$  abhängig. Die Berechnung von  $\mathbf{H}$  ist in der nachfolgenden Gleichung 5.18 aufgeführt.

$$\mathbf{H}(\mathbf{x}, \sigma) = \begin{bmatrix} F_{xx}(\mathbf{x}, \sigma) & F_{xy}(\mathbf{x}, \sigma) \\ F_{xy}(\mathbf{x}, \sigma) & F_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \quad (5.18)$$

$F_{xx}(\mathbf{x}, \sigma)$  ist die Faltung der zweiten partiellen Ableitung der Gauß-Funktion  $\frac{\delta^2}{\delta x^2} \cdot G(\sigma)$  nach  $x$ , mit der Bildposition  $\mathbf{x}$ .  $F_{yy}$  und  $F_{xy}$  sind analog zu berechnen.

Mit Hilfe der Determinante der Hesse-Matrix

$$\det(\mathbf{H}) = F_{xx} \cdot F_{yy} - F_{xy}^2 \quad (5.19)$$

lässt sich feststellen ob ein lokales Maximum und somit ein Merkmalskandidat vorliegt. Ist das Resultat positiv, dann handelt es sich um einen Merkmalskandidat. Andererseits liegt ein Sattelpunkt und damit kein Extremwert vor.  $F_{xx}$ ,  $F_{yy}$  und  $F_{xy}$  werden üblicherweise für die Faltung eines Bildes in Filtermasken diskretisiert. Die Diskretisierung ist in [BETG08] beschrieben und führt zu einem Geschwindigkeitsgewinn.

Die Invarianten des SIFT-Verfahrens gelten größtenteils auch beim SURF-Verfahren. SURF weist einen Präzisionsverlust auf, welcher jedoch in [BETG08] als akzeptabel eingeschätzt wird.

Detaillierte Informationen bezüglich SURF sind in [BFG06] und [BETG08] zu finden.

Die Abbildungen 5.16 und 5.17 veranschaulichen die Merkmale, die mit Hilfe des SIFT- und SURF-Verfahrens, detektiert wurden. Eine in dem Kapitel 5.1.3 beschriebene Sony C905 Handykamera lieferte dafür zwei perspektivisch unterschiedliche Aufnahmen einer quaderförmigen Papierschachtel.

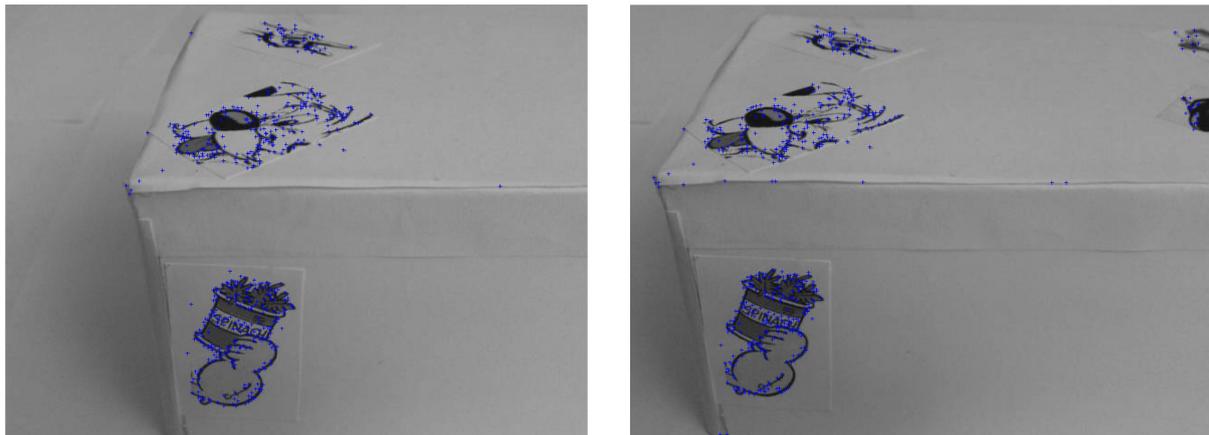
Die Aufnahmen wurden mit einer Auflösung von 640 x 320 Pixeln aufgenommen. In Abhängigkeit zur Auflösung<sup>12</sup> listet Tabelle 5.4 die Anzahl der detektierten Merkmale sowie die benötigte Berechnungsdauer<sup>13</sup> des SIFT/SURF-Verfahrens auf.

Für einen Geschwindigkeitsgewinn ist es möglich die Detektion der Merkmale sowie die Berechnung der Deskriptoren auf dem Grafikprozessor durchzuführen. Dieser Ansatz wurde, mit einem evaluierten Geschwindigkeitsgewinn, für SIFT in [ZIM07] und SURF in [CG08] umgesetzt.

<sup>11</sup>DoG steht für *Difference of Gaussian*. Bei einer DoG-Pyramide wird jede Pyramidenstufe aus jeweils zwei voneinander subtrahierten Gaußbildern aufgebaut.

<sup>12</sup>Veränderbare Parameter wie u. a. die Anzahl der Oktaven der *Difference-of-Gaussian* (DoG)-Pyramide oder der Kantengrenzbereich, indem robuste Merkmale gesucht werden, wurden aufgrund empirischer Untersuchungen nicht verändert.

<sup>13</sup>Die Berechnung wurde auf einem Laptop mit einer *Intel Core Duo* CPU (1 GHz), einem 2 GB Arbeitsspeicher und einem 32-bit Betriebssystem (Windows 7 Professional) durchgeführt. Der Programmcode wurde größtenteils in Matlab geschrieben.



(a) Aufnahme 1

(b) Aufnahme 2

**Abbildung 5.16:** Extrahierte SIFT-Merkmale.

(a) Aufnahme 1

(b) Aufnahme 2

**Abbildung 5.17:** Extrahierte SURF-Merkmale.

Verfahren	Aufnahme	Extrahierte Merkmale	Berechnungsdauer [Sekunden]
SIFT	1	642	3.84
SURF	1	122	0.14
SIFT	2	648	3.78
SURF	2	118	0.15

**Tabelle 5.4:** Extrahierte Merkmale und benötigte Berechnungsdauer mit SIFT/SURF.

### 5.3 Initiale Bestimmung von Weltpunkten

In diesem Kapitel wird die initiale Ermittlung von 3D-Weltpunkten (Oberflächenpunkte)<sup>14</sup> eines Gegenstandes beschrieben. Dazu werden zwei bis drei Kameraaufnahmen, mit unterschiedlichen Perspektiven zum Gegenstand, verwendet.

<sup>14</sup>Oberflächenpunkte werden fortlaufend als sogenannte *Weltpunkte* bezeichnet.

### 5.3.1 Suche nach Merkmalskorrespondenzen

Im ersten Schritt sind robuste Merkmale aus den zwei bzw. drei Aufnahmen zu extrahieren. Diese sollten einen hohen Grad an Invarianz bezüglich Rotation, Translation, Skalierung, Änderung der Lichtverhältnisse und teilweiser affiner Verzerrung aufweisen. Für die Suche nach Merkmalen kommt entweder das SIFT- oder das SURF-Verfahren zum Einsatz. Diese wurden in dem Kapitel 5.2 (*Merkmalsextraktion*) beschrieben.

Anschließend müssen die Merkmalskorrespondenzen zwischen den ersten zwei Aufnahmen  $\mathbf{B}_1 \leftrightarrow \mathbf{B}_2$  gesucht werden. Aus Sicherheitsgründen kann eine dritte Aufnahme  $\mathbf{B}_3$  zur Verifikation der Korrespondenzen herangezogen werden. Denn die Gefahr besteht, dass die erste Korrespondenzmenge  $\{\mathbf{B}_1 \leftrightarrow \mathbf{B}_2\}$  fehlerbehaftete Merkmalszuordnungen beinhaltet. Aus diesem Grund wird die zweite Korrespondenzmenge  $\{\mathbf{B}_2 \leftrightarrow \mathbf{B}_3\}$  ermittelt. Die Schnittmenge  $\{\mathbf{B}_1 \leftrightarrow \mathbf{B}_2\} \cap \{\mathbf{B}_2 \leftrightarrow \mathbf{B}_3\}$  zwischen diesen Mengen ist mit einer sehr großen Wahrscheinlichkeit fehlerfrei. Aus diesem Grund sind für weitere Berechnungen (Filterung + Triangulation) nur die Korrespondenzen aus den ersten zwei Bildern zu verwenden, welche sich in der Schnittmenge befinden.

Im Nachfolgenden werden zwei Verfahren zur Korrespondenzsuche vorgestellt.

#### 5.3.1.1 Merkmalsvergleich auf Basis des euklidischen Abstandes

Die Suche nach Merkmalskorrespondenzen erfolgt üblicherweise auf Basis des euklidischen Abstandes. Der euklidische Abstand zweier Vektoren  $\mathbf{v}_1$  und  $\mathbf{v}_2$  wird mit der Formel 5.20 berechnet.  $n$  kennzeichnet die Länge der Vektoren.

$$\sqrt{\sum_{i=1}^n (\mathbf{v}_{1i} - \mathbf{v}_{2i})^2} \quad (5.20)$$

Beispielsweise existiert ein 128-dimensionale Deskriptor  $\mathbf{D}_{1 \times 128}$  aus der ersten Aufnahme und  $j$  Deskriptoren  $\mathbf{D}_{j \times 128}$  aus der Aufnahme 2. Somit ist der korrespondierende Deskriptor  $\mathbf{D}_j$  derjenige, welcher die geringste euklidische Distanz zum Deskriptor  $\mathbf{D}_1$  aufweist. Dieses Vorgehen wird als *Brute Force*-Ansatz bezeichnet.

Aus Effizienzgründen wurden in dieser Arbeit die Korrespondenzen der Merkmale nicht auf Basis des euklidischen Abstandes ermittelt. Anstelle dessen wird das Skalarprodukt zwischen Einheitsvektoren

$$\mathbf{D}_i \cdot \mathbf{D}_j = |\mathbf{D}_i| \cdot |\mathbf{D}_j| \cdot \cos \angle (\mathbf{D}_i, \mathbf{D}_j) \quad (5.21)$$

gebildet. Dabei entspricht  $\cos \angle (\mathbf{D}_i, \mathbf{D}_j)$  dem Kosinus des von den beiden Deskriptoren eingeschlossenen Winkels. Des Weiteren ist zu beachten, dass die Deskriptoren normiert sind und damit die Länge eines Deskriptors eins ergibt (Gleichung 5.22).

$$|\mathbf{D}| = 1 \quad (5.22)$$

Resultierend wird der euklidische Abstand approximiert, indem der Winkel

$$\angle (\mathbf{D}_i, \mathbf{D}_j) = \arccos \left( \frac{\mathbf{D}_i \cdot \mathbf{D}_j}{|\mathbf{D}_i| \cdot |\mathbf{D}_j|} \right) \quad (5.23)$$

$$\angle (\mathbf{D}_i, \mathbf{D}_j) = \arccos (\mathbf{D}_i \cdot \mathbf{D}_j) \quad (5.24)$$

zwischen zwei Deskriptoren betrachtet wird.

Weiterführend müssen die Merkmalskorrespondenzen eine Zusatzbedingung erfüllen. Der Grund dafür klärt ein Beispiel.

Es existieren zehn eindimensionale Deskriptoren  $\mathbf{D}_1, \dots, \mathbf{D}_{10}$ . Dabei besitzt  $\mathbf{D}_1$  den Wert 1,  $\mathbf{D}_2$  den Wert 2,  $\dots$ , und  $\mathbf{D}_{10}$  den Wert 10. Bei einer Korrespondenzsuche wären jeweils die Nachbarn die beste Korrespondenz. Aus diesem Grund wird eine Zusatzbedingung eingeführt, welche für die Einzigartigkeit eines Merkmales zuständig ist.

In der nachfolgenden Gleichung 5.25 ist die Zusatzbedingung

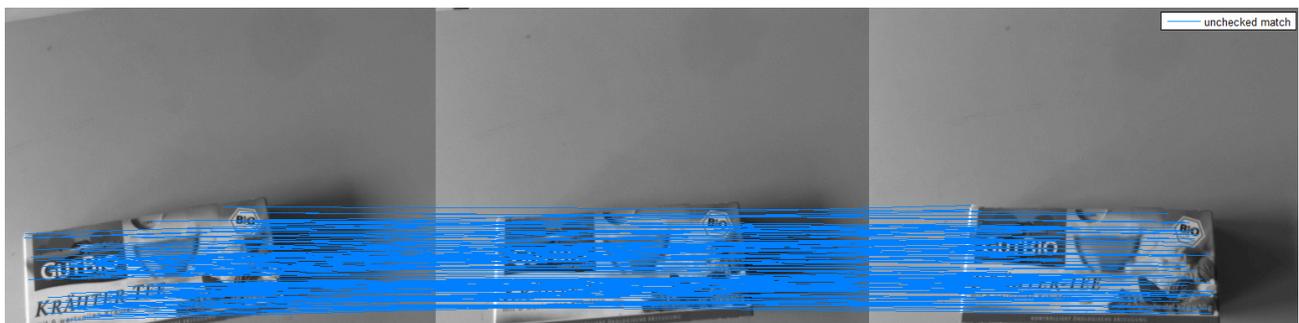
$$\mathbf{D}_i < \text{threshold} \cdot \mathbf{D}_j \quad \begin{cases} \mathbf{D}_i : \text{Kleinsten Abstand/Winkel} \\ \mathbf{D}_j : \text{Zweit-kleinsten Abstand/Winkel} \end{cases} \quad (5.25)$$

aufgeführt. Dabei ist *threshold* ein beliebiger Wert zwischen 0,0001 und 1. Dieser ist ein Maß für die Einzigartigkeit eines Merkmales und bedeutet Folgendes. Eine Merkmalskorrespondenz ist nur dann vorhanden, wenn das Verhältnis vom kleinsten- zum zweit-kleinsten euklidischen Abstand einen Schwellwert nicht übersteigt. Dies gilt analog wenn das Skalarprodukt zwischen Einheitsvektoren anstelle vom euklidischen Abstand herangezogen wird.

In dieser Arbeit wird ein Threshold von 0,1 verwendet. Die nachfolgende Abbildung 5.18 stellt die, mit dem approximierten euklidischen Abstand ermittelten, Merkmalskorrespondenzen dar. Dabei veranschaulicht die Grafik 5.18a die Merkmalskorrespondenzen aus den ersten zwei Bildern. Des Weiteren sind in der zweiten Grafik 5.18b die Merkmalskorrespondenzen des zweiten und dritten Bildes zusätzlich dargestellt.



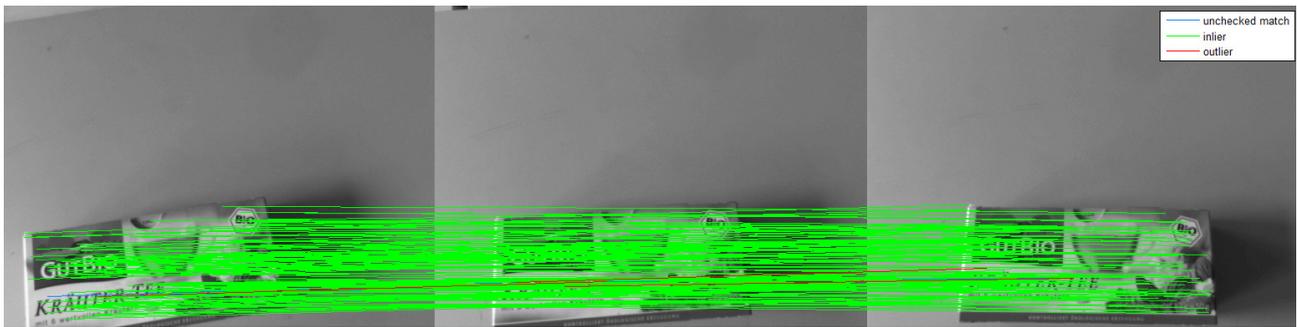
(a) SURF-Korrespondenzen Bild 1 ↔ 2



(b) SURF-Korrespondenzen Bild 1 ↔ 2 und Bild 2 ↔ 3

**Abbildung 5.18:** SURF-Korrespondenzen zwischen den ersten- und den nächsten zwei Kameraaufnahmen.

Die Schnittmenge zwischen den zwei Korrespondenzmengen veranschaulicht die nachfolgende Abbildung 5.19. Dort sind die Korrespondenzen aus der Schnittmenge grün dargestellt.



**Abbildung 5.19:** Schnittmenge zwischen zwei Korrespondenzmengen.

Die Tabelle 5.5 veranschaulicht die gefundenen Korrespondenzen tabellarisch.

Verfahren	Aufnahme	Merkmale	Korrespondenzen	Berechnungsdauer [Sek.]
SURF	1,2	406+401	227 (Menge A)	
SURF	2,3	401+390	228 (Menge B)	
SURF	1,2,3	406+401+390	154 (Schnittmenge $A \cap B$ )	0.4

**Tabelle 5.5:** Korrespondenzanalyse der ersten drei Aufnahmen.

### 5.3.1.2 Verbesserte Korrespondenzsuche mit einem Suchbaum

Ein Vergleich mit allen Merkmalsvektoren ist aufgrund der hohen Vektordimension sehr zeitintensiv. Eine bessere Lösung ist der Aufbau eines sogenannten k-d Baumes. Dieser, zur Verwaltung von Merkmalspunkten k-dimensionale Suchbaum, erleichtert das Finden vom nächsten Nachbarn bzw. der Merkmalskorrespondenzen.

#### 1. Aufbau eines k-d Baumes

Der k-d Baum ist ein binärer Suchbaum der Tiefe  $t$ , welcher einen Raum in  $2^t$  Unterräume aufteilt. Jeder Knoten gliedert jeweils zwei Unterräume auf seine zwei Nachfolgerknoten auf. Die Blätter des Baumes repräsentieren die Merkmalspunkte mit ihren Deskriptoren. Mit  $n$  Merkmalsvektoren ergibt sich daraus eine Maximaltiefe von  $\log_2 n$ .

Für den Aufbau eines k-d Baumes sind die höheren Baumebenen priorisiert zu gewichten. Dafür sind die Dimensionen der Merkmalsvektoren anhand ihrer Varianz zu sortieren. In Gleichung 5.26 ist die Berechnung der Varianz nachfolgend dargestellt.  $\bar{v}$  kennzeichnet den Deskriptormittelwert.

$$\frac{1}{n} \cdot \sum_{i=1}^n (v_i - \bar{v})^2 \quad (5.26)$$

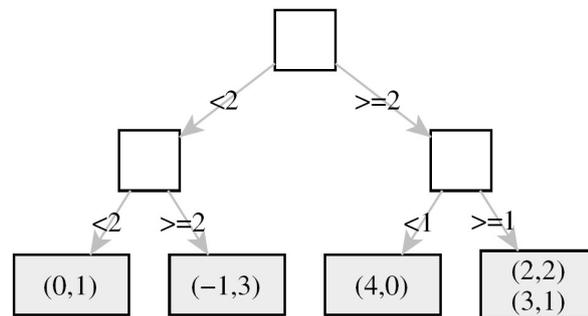
Nachdem die Vektordimensionen anhand ihrer Varianz sortiert wurden, müssen die Zweigbedingungen für den Wurzelknoten bestimmt werden. Dazu wird der Median über die höchst-priorisierte Dimension aller Deskriptoren gebildet. Die priorisierte Dimension ist das erste Element aus der sortierten Menge. Der Median einer geordneten Menge  $\mathbf{M}^{\text{sort}}$  ist folgendermaßen zu berechnen:

$$\tilde{\mathbf{x}} = \begin{cases} \mathbf{M}_{\frac{n+1}{2}}^{\text{sort}} & n \text{ ungerade} \\ \frac{1}{2} \cdot \left( \mathbf{M}_{\frac{n}{2}}^{\text{sort}} + \mathbf{M}_{\frac{n+1}{2}}^{\text{sort}} \right) & n \text{ gerade} \end{cases} \quad (5.27)$$

Es liegen beispielhaft drei zweidimensionale Punkte  $\mathbf{x}_1(0,2)$ ,  $\mathbf{x}_2(1,4)$  und  $\mathbf{x}_3(2,8)$  vor. Dann ist der Median der zweiten Dimension 4. Die Deskriptoren sind anschließend, anhand der in der zweiten Dimension befindenden Werte, in den Baum einzuordnen. In dem Fall, dass der Wert echt kleiner als der Median ist, wird der Deskriptor in den linken Ast des Wurzelknotens eingeordnet. Ansonsten ist der Deskriptor in

den rechten Ast einzusortieren. Anschließend ist der Median über alle Deskriptoren eines Unterraumes in Bezug auf die zweithöchst-priorisierte Dimension zu bestimmen. Das Verfahren läuft iterativ wie bisher beschrieben fort. Der Median bewirkt einen ausgeglichen verteilten Suchbaum.

Die Abbildung 5.20 veranschaulicht einen einsortierten k-d Baum im zweidimensionalen Fall. Der k-d Baum wurde mit den Vektoren  $\mathbf{v}_1(0,1)$ ,  $\mathbf{v}_2(-1,3)$ ,  $\mathbf{v}_3(4,0)$ ,  $\mathbf{v}_4(2,2)$  und  $\mathbf{v}_5(3,1)$  aufgebaut. Die erste Dimension weist die größte Varianz auf. Aufgrund dessen wurde der Median über die erste Dimension aller fünf Vektoren gebildet. Aus den Vektorwerten der ersten Dimension  $\{0, -1, 2, 3, 4\}$  wird der Median 2 ermittelt. Dieser Wert ist die Entscheidungsbedingung für den Wurzelzweig. Bei den Vektoren  $\mathbf{v}_1(0,1)$  und  $\mathbf{v}_2(-1,3)$  ist der erste Dimensionswert echt kleiner als 2. Das hat zur Folge, dass diese beiden Vektoren in den linken Ast einzusortieren sind. Die Vektoren  $\mathbf{v}_3(4,0)$ ,  $\mathbf{v}_4(2,2)$  und  $\mathbf{v}_5(3,1)$  müssen dementsprechend in den rechten Ast einsortiert werden. Anschließend wird der Baum nach dem gleichen Verfahren weiter aufgeteilt. Der Median für die zwei neuen Knoten ist anhand der zweiten Dimension zu berechnen. Für den linken Knoten ergibt sich somit ein Median von  $(1+3)/2 = 2$ . Der rechte Knoten besitzt einen Median von 1. Die Vektoren sind dementsprechend weiter nach unten in den Baum einzusortieren.

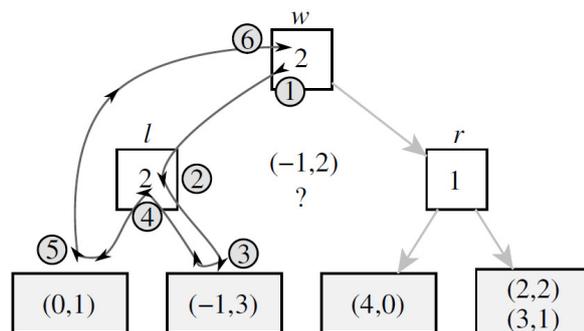


**Abbildung 5.20:** Aufbau eines k-d Baumes mit einer zweidimensionalen Punktmenge. [HÄR06]

## 2. Suche im k-d Baum

Beispielhaft soll der nächste Nachbar (Merkmalskorrespondenz), im aufgebauten k-d Baum (Vgl. *Aufbau eines k-d Baumes*), zum Vektor  $\mathbf{v}_x(-1,2)$  gefunden werden. Die Zweigbedingungen sind jeweils an den Ästen vermerkt. Die Blätter sind Deskriptoren und somit die potentiellen nächsten Nachbarn des Vektors  $\mathbf{v}_x$ .

Nachfolgend ist die Suche des nächsten Nachbarn zum Vektor  $\mathbf{v}_x$  anhand der Abbildung 5.21 beschrieben.



**Abbildung 5.21:** Suchablauf in einem k-d Baum. [HÄR06]

- a) Auf der Ebene der Wurzel  $w$  ist aufgrund der Priorisierung die erste Dimension des Vektors  $\mathbf{v}_x$  zu betrachten. Anhand des Wertes der ersten Dimension wird entschieden, ob nach links oder rechts im Baum traversiert wird.  $-1$  ist echt kleiner als der Zweigbedingungswert  $2$ . Aufgrund dessen ist nach links im Baum zu traversieren. In einer sogenannten *Backtracking-Warteschlange*  $Q$  ist der quadratische Abstand  $(2 - (-1))^2 = 9$  zwischen der ersten Dimension zur Wurzel zu vermerken. Dieser kennzeichnet den Mindestabstand, den alle Deskriptoren des rechten Unterbaumes zum Vektor  $\mathbf{v}_x$  besitzen. Die Warteschlange enthält somit einen Eintrag:  $Q[(w, 9)]$

- b) Am Nachfolgerknoten  $l$  ist im Anschließenden die zweite Dimension des Vektors  $\mathbf{v}_x$  zu betrachten. Die zweite Dimension des Vektors  $\mathbf{v}_x$  enthält den Wert 2. Da 2 größer gleich der Zweigbedingung des Knotens  $l$  ist, wird nach rechts traversiert. Der Mindestabstand des Knotens  $l$  ist in die Warteschlange mit dem Wert  $(2-2)^2 = 0$  einzutragen. 0 kennzeichnet den Mindestabstand aller Vektoren im linken Blatt des Knotens  $l$ . Dieser wird in der Warteschlange vermerkt:  $\mathbf{Q}[(l,0), (w,9)]$
- c) Im Anschluss ist der Vektor  $\mathbf{v}_x$  mit allen an dem Blatt befindenden Vektoren zu vergleichen.  $(-1,3)$  ist der einzige Vektor an diesem Ort. Damit ist dieser der beste nächste Nachbar des Vektors  $\mathbf{v}_x$ . Der quadratische Abstand zwischen diesen beiden Vektoren beträgt 1. Der quadratische Abstand zwischen zwei Vektoren wird ermittelt, indem die beiden quadratischen Abstände der zwei Dimensionen summiert werden.
- d) Der sogenannte *Backtracking*-Schritt kommt an dieser Stelle zum Einsatz. Dazu ist der Knoten mit dem kleinsten Mindestabstand aus der *Backtracking*-Warteschlange zu entnehmen. In dem Fall hat  $l$  den geringsten Mindestabstand. In diesem Teilbaum wird der alternative Ast betrachtet.
- e) An diesem alternativen Ast werden erneut alle Vektoren des Blattes mit dem Vektor  $\mathbf{v}_x$  verglichen. Der Vektor  $(0,1)$  hat dabei einen quadratischen Abstand von 2. Da der Abstand nicht besser geworden ist, bleibt der nächste Nachbar mit dem Abstand von 1, der Vektor  $(-1,3)$ .
- f) Im letzten Schritt wird noch geprüft, ob in der Warteschlange ein weiterer Knoten existiert, welcher einen kleineren Abstand als der bisher ermittelte kleinste Abstand von 1 besitzt. Der Abstand 9 von  $w$  ist größer als der Abstand 1 von  $(-1,3)$ . Damit bleibt der Vektor  $(-1,3)$  der nächste Nachbar und somit der am besten übereinstimmende Vektor zu  $\mathbf{v}_x$ .

[BL97] und [Low04] verwenden zur effizienteren Suche im k-d Baum diesen verbesserten Suchalgorithmus mit *Backtracking*. Diese Methode wird auch als *Best-Bin-First* (BBF) bezeichnet. Dabei ist in diesem modifizierten k-d Baum eine 95% Suchübereinstimmung bei verbesserter Geschwindigkeit erzielbar.

Für weitere Informationen bezüglich der Erzeugung und Suche in k-d Bäumen ist auf die Literatur [HEY05] und [BL97] verwiesen.

Zur Vollständigkeit sei erwähnt, dass die Dimension eines Merkmalsvektors reduziert werden kann. Die sogenannte Hauptkomponentenanalyse (engl. *Principal Component Analysis*, kurz PCA) und Diskriminanzanalyse (engl. *Linear Discriminant Analysis*, kurz LDA) sind Verfahren, welche eine Dimensionsreduzierung realisieren können. In [KRZ00] und [FHT96] sind die Hauptkomponenten- und Diskriminanzanalyse ausführlich beschrieben und gegebenenfalls zu entnehmen.

## 5.3.2 Filterung

Zu diesem Zeitpunkt wurden die Merkmalskorrespondenzen aus den ersten zwei Kameraaufnahmen extrahiert, wobei alle Korrespondenzen, welche sich nicht in der zweiten Korrespondenzmenge  $\{\mathbf{B}_2 \leftrightarrow \mathbf{B}_3\}$  befinden, entfernt wurden. Damit konnte die erste Korrespondenzmenge  $\{\mathbf{B}_1 \leftrightarrow \mathbf{B}_2\}$  grob auf fehlerbehaftete Korrespondenzen verifiziert werden. In diesem Unterkapitel werden drei weitere Filtermaßnahmen vorgestellt.

### 5.3.2.1 Eliminierung vorhandener Duplikate

In der Merkmalsextraktion des SIFT-Verfahrens wird ein Merkmal in einem gesonderten Fall (gleiche Position +  $\geq 2$  Orientierungen) mehrmals extrahiert. Befindet sich beispielhaft ein Merkmal an einer Ecke, besitzt dieses zwei Hauptorientierungen. Aufgrund dessen wird das Merkmal für jede Hauptorientierung separat extrahiert. Dies hat eine erhöhte Wiedererkennungsrates des Merkmals zur Folge.

In einem späteren Abschnitt dieser Arbeit wird anhand der Merkmalskorrespondenzen die räumliche Beziehung zwischen zwei Aufnahmen ermittelt. Diese Beziehung wird mit einem Algorithmus berechnet, welcher acht Merkmalskorrespondenzen verwendet. Befinden sich in den acht Merkmalskorrespondenzen ein oder mehrere

Duplikate (gleicher Merkmalsvektor + unterschiedliche Orientierung), wird die Beziehung zwischen den Aufnahmen falsch berechnet. Dies wird vermieden, indem die Duplikate eliminiert werden.

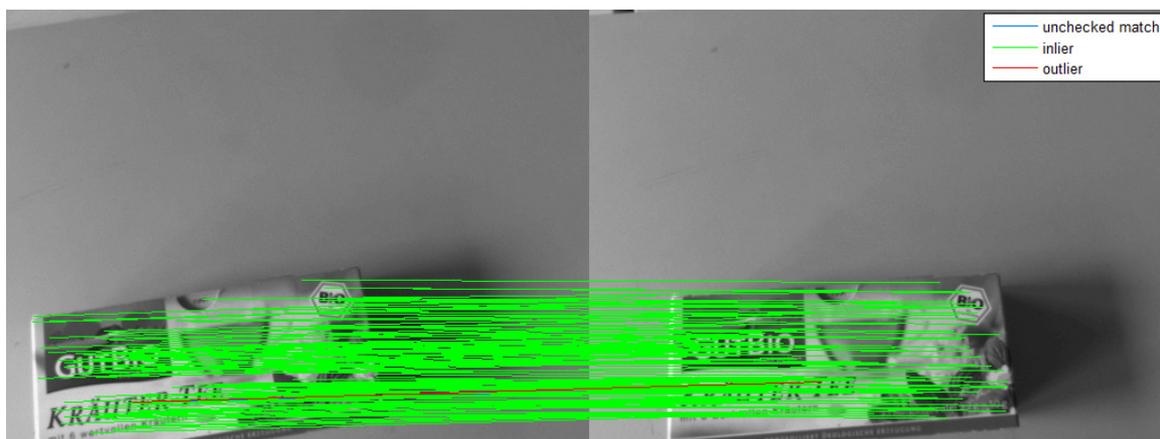
Des Weiteren kann ein Deskriptor beim Merkmalsvergleich mehrfach als beste Korrespondenz ausgewählt werden. Dies beeinflusst auch die Berechnung der Beziehung zwischen zwei Bildern und wird ebenfalls als Duplikat deklariert und entfernt. Die nachfolgende Tabelle 5.6 listet die eliminierten Duplikate zwischen zwei Aufnahmen auf.

Verfahren	Aufnahme	Merkmale	Verifizierte Korrespondenzen	Duplikate	Berechnungsdauer [Sekunden]
SIFT	1,2	652+670	249	11	0.01
SURF	1,2	406+401	154	1	0.0025

**Tabelle 5.6:** Duplikate der Aufnahme 1 und 2.

### 5.3.2.2 Aussortieren von Fehlzuordnungen

Fehlzuordnungen zwischen Merkmalen treten vereinzelt bei der Korrespondenzsuche auf. Diese falschen Merkmalskorrespondenzen werden als *Outlier* bezeichnet. *Inlier* besitzen dagegen eine richtige Zuordnung. Die Abbildung 5.22 veranschaulicht zwei falsch zugeordnete Merkmale. Diese sind rot gekennzeichnet. Inlier sind mit einer grünen Linie miteinander verbunden.



**Abbildung 5.22:** Fehlzuordnung zwischen zwei Merkmalen.

Mit Hilfe des 8-Punkte-Algorithmus kann die Fundamentalmatrix, welche die Beziehung zwischen zwei Kameraaufnahmen beschreibt, berechnet werden. RANSAC wird gängigerweise herangezogen, um die Fundamentalmatrix mit Hilfe der verfügbaren Merkmalskorrespondenzen und dem 8-Punkte-Algorithmus optimal zu schätzen. Bei dieser Schätzung der Fundamentalmatrix werden, mit einer großen Wahrscheinlichkeit, alle Outlier aufgedeckt.

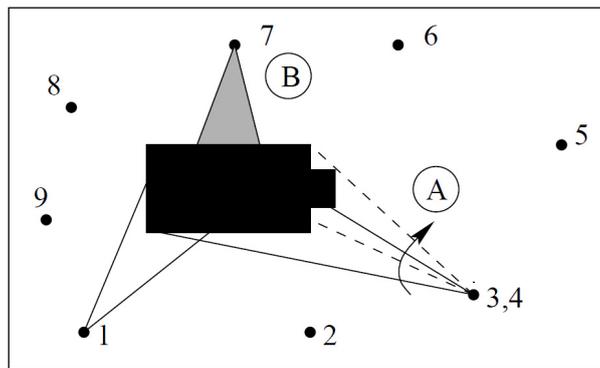
Im Kapitel 3.3.3 (*Robuste Schätzung der Fundamentalmatrix*) wurde diese Vorgehensweise beschrieben und findet an dieser Stelle zur Ermittlung der Outlier sowie der Berechnung der optimalen Fundamentalmatrix Gebrauch.

### 5.3.2.3 Erkennung einer degenerierten Merkmalsmenge

Unter Verwendung von 2D-Bildkorrespondenzen können die zugehörigen 3D-Punkte berechnet werden. Allerdings existieren zwei Situationen (sogenannte Degenerationsfälle), bei denen keine Tiefeninformationen bzw. 3D-Punkte rekonstruierbar sind. Dabei kann lediglich die planare Homographie von zwei Aufnahmen bestimmt werden. Dieses Unterkapitel beschreibt die Überprüfung, ob sich mit den übrig gebliebenen Inlier Tiefeninformationen rekonstruieren lassen.

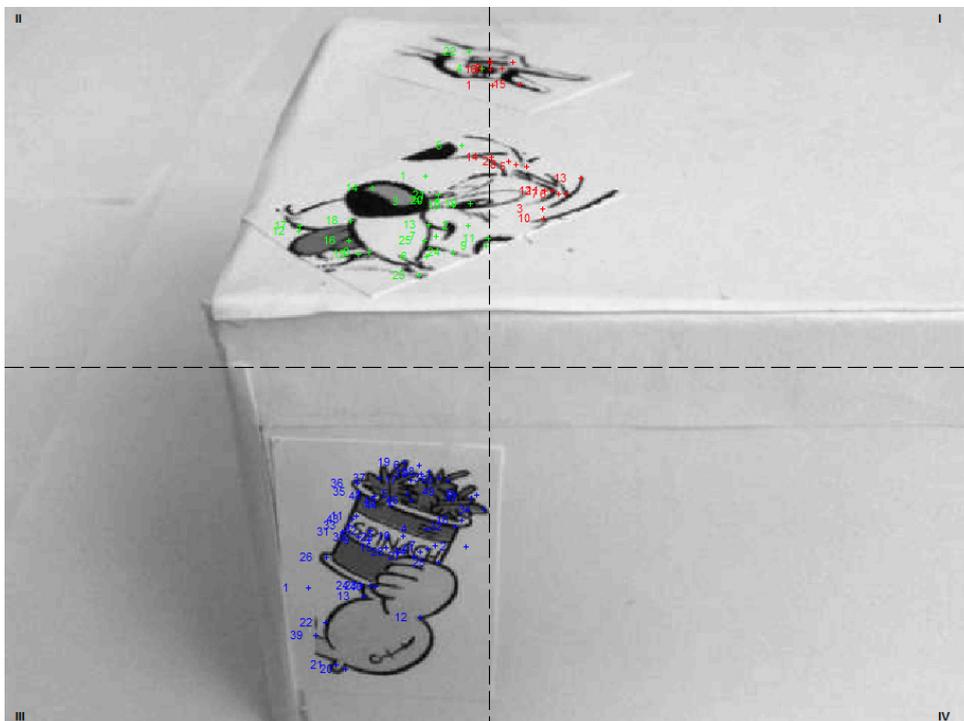
Es werden zwei Arten von Degenerationen unterschieden. Beträgt die Translation zwischen der ersten- und zweiten Kameraposition null, liegt eine Bewegungsdegeneration vor. Befinden sich alle Weltpunkte der zu triangulierenden Merkmalskorrespondenzen auf einer planaren Fläche<sup>15</sup>, resultiert eine Strukturdegeneration. In diesen zwei Situationen können die Weltpunkte nicht berechnet werden.

In der Abbildung 5.23 sind die zwei Degenerationsarten grafisch dargestellt. Die Kameraaufnahme an Position 3 und 4 befinden sich an derselben Position. Die Kamera wurde lediglich um ihr optisches Zentrum rotiert. Dies hat eine Bewegungsdegeneration, welche in der Abbildung mit *A* gekennzeichnet ist, zur Folge. Die Kameraaufnahme an Position 7 enthält, aufgrund der Sicht auf eine planare Fläche, keine räumlichen Objektinformationen. Damit entsteht eine Strukturdegeneration, die mit einem *B* gekennzeichnet ist.



**Abbildung 5.23:** Struktur- und Bewegungsdegeneration. [TFZ99]

Wie in dieser Arbeit Degenerationen vermieden und erkannt werden, wird im Nachfolgenden erläutert. In dem Fall, dass eine bewegliche Kamera zur Aufnahme Verwendung findet, ist die Bewegungsdegeneration durch eine Translation ungleich null zu vermeiden. Wird dagegen, wie in dem Unterkapitel 5.1.3 beschrieben, ein ansteuerbarer Drehteller in Kombination mit einer ortsfesten Kamera verwendet, kann eine Bewegungsdegeneration vollkommen ausgeschlossen werden.



**Abbildung 5.24:** Ineffiziente Methode zur Erkennung einer Strukturdegeneration.

Eine Strukturdegeneration kann in zwei verschiedenen Phasen überprüft werden. Wenn lediglich die Merkmalskorrespondenzen vorhanden sind, dann ist eine Strukturdegeneration nur grob erkennbar. Jede Kameraaufnahme

<sup>15</sup>In diesem Fall werden die 3D-Weltpunkte als *koplanar* bezeichnet.

me wird in vier gleich große Flächen aufgeteilt. Diese vier Areale sind in der Grafik 5.24 mit einer Quadrantennummerierung gekennzeichnet. Eine Strukturdegeneration ist vorhanden, wenn die Merkmale in nur einem Areal vorliegen.

Allerdings ist dies keine gute Methode zur Erkennung einer Strukturdegeneration. Es lassen sich zahlreiche Beispiele nennen, bei denen die Erkennung fehlschlägt. Beispielsweise befindet sich eine Dose komplett in einem Quadranten und es wurden nicht planare Merkmale auf der Seitenfläche des Kreiszylinders detektiert. Resultierend würde die Erkennung eine Strukturdegeneration detektieren. Dabei besteht keinesfalls eine Strukturdegeneration, da die Merkmale auf der Seitenfläche des Kreiszylinders verteilt sind.

Aus diesem Grund wurde die genannte Methode zur Erkennung einer Strukturdegeneration durch ein adaptiertes Verfahren ersetzt. Das entwickelte Verfahren kommt in einer späteren Phase *Triangulation und Bündelblockausgleich* auf S. 56 zum Einsatz. Dieses erkennt eine Strukturdegeneration anhand von  $n \geq 5$  triangulierten Weltpunkte  $\mathbf{X}_{3 \times n}$  zuverlässig. Das Verfahren wird im nachfolgenden vorgestellt.

In dem Fall, dass die triangulierten 3D-Weltpunkte näherungsweise planar sind, sind diese zu verwerfen. Das Vorhandensein einer planaren Weltpunktmenge kann mit dem arithmetischen Mittelwert und der Singulärwertzerlegung ermittelt werden. Zu Beginn wird dafür der arithmetische Mittelwert

$$\bar{\mathbf{X}}_{\text{arith}} = \frac{1}{n} \cdot \sum_{i=1}^n \mathbf{X}_i \quad (5.28)$$

der aktuell triangulierten Weltpunktmenge berechnet. Dieser resultiert in einem  $3 \times 1$  Vektor  $\bar{\mathbf{X}}_{\text{arith}}$ . Anschließend beschreiben die Varianzen in einer  $3 \times 3$  Matrix

$$\sigma_{3 \times 3}^2 = \begin{bmatrix} \sigma_{11} & \dots & \sigma_{13} \\ \vdots & \ddots & \vdots \\ \sigma_{31} & \dots & \sigma_{33} \end{bmatrix} = \left( \mathbf{X}_{3 \times n} - \left( \bar{\mathbf{X}}_{\text{arith}} \cdot \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}_{1 \times n} \right) \right) \cdot \left( \mathbf{X}_{3 \times n} - \left( \bar{\mathbf{X}}_{\text{arith}} \cdot \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}_{1 \times n} \right) \right)^T \quad (5.29)$$

die Abweichung jedes Weltpunktes zum arithmetischen Mittelwert und geben damit das Streuungsmaß der Weltpunkte wieder. Danach kann die Singulärwertzerlegung auf die Varianzen  $\sigma^2$  angewendet werden. Das Resultat ist die Zerlegung der Matrix  $\sigma^2$  in drei Matrizen  $\mathbf{U}$ ,  $\mathbf{S}$  und  $\mathbf{V}$ . Die unitäre  $3 \times 3$  Matrix  $\mathbf{U}$  und die Adjungierte einer unitären  $3 \times 3$  Matrix  $\mathbf{V}$  sind aufgrund der Quadrierung

$$\left( \mathbf{X}_{3 \times n} - \left( \bar{\mathbf{X}}_{\text{arith}} \cdot \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}_{1 \times n} \right) \right) \cdot \left( \mathbf{X}_{3 \times n} - \left( \bar{\mathbf{X}}_{\text{arith}} \cdot \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}_{1 \times n} \right) \right)^T \quad (5.30)$$

identisch und werden des Weiteren nicht betrachtet. Die  $3 \times 3$  Diagonalmatrix

$$\mathbf{S} = \begin{bmatrix} d_{11} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & d_{33} \end{bmatrix} \quad (5.31)$$

gibt die Streuung<sup>16</sup> der Weltpunktmenge an und wird anhand ihrer Singulärwerte zur Erkennung einer planaren Punktmenge herangezogen. In dem Fall, dass der (Singulär-) Wert  $d_{33}$  im Verhältnis zum Singulärwert  $d_{22}$  sehr klein ist, liegt eine Strukturdegeneration vor. Dies führt zu der abschließende Abfrage:

$$(d_{33}/d_{22}) < 0.001 \quad (5.32)$$

<sup>16</sup>Im Anbetracht der zu überprüfenden Strukturdegeneration.

Für weitere Ansätze zur Erkennung und Vermeidung von degenerierten Daten sei auf [TFZ99] und [HZ04] ab Seite 295 verwiesen.

### 5.3.3 Berechnung der 3D-Welpunkte

In diesem Unterkapitel sind die vier Phasen

1. Bestimmung der Projektionsmatrizen
2. Triangulation und Bündelblockausgleich
3. Filterung anhand Rückprojektion
4. Korrektur der Skalierung

beschrieben. Mit diesen Phasen lassen sich detailgetreue 3D-Oberflächenpunkte berechnen. Dafür werden die aus den Kapiteln 5.3.1 und 5.3.2 ermittelten Inlier  $\mathbf{x} \leftrightarrow \mathbf{x}'$  sowie deren zugehörige Fundamentalmatrix  $\mathbf{F}$  (Vgl. Kapitel 3.3) verwendet. Die intrinsischen Kameraparameter  $\mathbf{K}$  aus Kapitel 5.1.3 sind des Weiteren notwendig.

Anmerkend sei auf die Kapitel 3.3.2 und 3.3.3 verwiesen, in denen die Berechnung der Fundamentalmatrix mit mindestens acht korrespondierenden Bildpunkten beschrieben wurde. Dabei wird die sogenannte Epipolarbedingung  $\mathbf{x}'^T \cdot \mathbf{F} \cdot \mathbf{x} = 0$  betrachtet, damit  $\mathbf{F}$  über ein lineares Gleichungssystem bestimmt werden kann.

#### 5.3.3.1 Bestimmung der Projektionsmatrizen

Eine  $3 \times 4$  Projektionsmatrix  $\mathbf{P}$  bildet homogene<sup>17</sup> 3D-Welpunkte  $\mathbf{X}_i$  auf homogene 2D-Bildpunkte  $\mathbf{x}_i$  einer Kameraaufnahme ab (Gleichung 5.33).

$$\mathbf{x}_i = \mathbf{P} \cdot \mathbf{X}_i \quad (5.33)$$

Die Projektionsmatrix kann mit der ermittelten Fundamentalmatrix  $\mathbf{F}_{3 \times 3}$ , den intrinsischen Kameraparametern

$$\mathbf{K} = \begin{bmatrix} f_x & f_x \cdot s & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5.34)$$

und den Merkmalskorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  berechnet werden. Dafür wird die Fundamentalmatrix zusammen mit den intrinsischen Kameraparametern  $\mathbf{K}$  zu einer sogenannten *Essentialmatrix*  $\mathbf{E}_{3 \times 3}$  erweitert (Vgl. [EBE09], S. 35):

$$\mathbf{E} = \mathbf{K}^T \cdot \mathbf{F} \cdot \mathbf{K} \quad (5.35)$$

Das Projektionszentrum der Kamera 1 kann einfachheitshalber in den Weltursprung gelegt werden. Dies führt zur Projektionsmatrix aus Gleichung 5.36, bei der  $\mathbf{I}$  die Einheitsmatrix kennzeichnet.

$$\mathbf{P}_1 = \left[ \mathbf{I}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.36)$$

<sup>17</sup>Die Transformation von kartesischen- zu homogenen Koordinaten wird durch die Abbildung  $[X, Y, Z]^T \mapsto [X, Y, Z, 1]^T$  bzw.  $[x, y]^T \mapsto [x, y, 1]^T$  beschrieben.

Unter Einbeziehung der intrinsischen Kameraparameter wird der Rotationsanteil von  $\mathbf{P}_1$  ersetzt (Gleichung 5.37).

$$\mathbf{P}_1 = \left[ \mathbf{K}^T, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right] \quad (5.37)$$

Im Folgenden ist die zweite Projektionsmatrix  $\mathbf{P}_2$  in Abhängigkeit zu  $\mathbf{P}_1$  zu berechnen. Für diese Matrix ist eine  $3 \times 3$  Rotationsmatrix und ein  $3 \times 1$  Translationsvektor

$$\mathbf{P}_2 = [\mathbf{R}, \mathbf{t}] \quad (5.38)$$

gesucht, welche einen Weltpunkt  $\mathbf{X}_i$  in das zweite Kamerakoordinatensystem transformiert:

$$\mathbf{x}'_i = \mathbf{P}_2 \cdot \mathbf{X}_i \quad (5.39)$$

Um die gesuchte Projektionsmatrix  $\mathbf{P}_2$  und damit die Rotation  $\mathbf{R}$  und Translation  $\mathbf{t}$  zu berechnen, wird die aus Gleichung 5.35 ermittelte Essentialmatrix faktorisiert (Vgl. [HZ04], S. 275 ff.). Eine Faktorisierung hat eine Zerlegung der Matrix in ihre Einzelteile zu Folge. Diese können für die Herleitung der Rotation und Translation verwendet werden. Die Gleichung 5.40 zeigt die Zerlegung der Essentialmatrix in das Kreuzprodukt einer Translation mit einer Rotation, welches auf die zweite Projektionsmatrix bezogen ist.

$$\mathbf{E} = [\mathbf{t}] \times \mathbf{R} \quad (5.40)$$

Weiterführend kann eine schiefsymmetrische Matrix  $\mathbf{S}$  und eine Rotation  $\mathbf{R}$  aus der Essentialmatrix gebildet werden.

$$\mathbf{E} = [\mathbf{t}] \times \mathbf{R} = \mathbf{S} \cdot \mathbf{R} \quad (5.41)$$

Unter Verwendung der Hilfsmatrizen

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.42)$$

$$\mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.43)$$

und der gesonderten Singulärwertzerlegung von  $\mathbf{E}$  (Vgl. Kapitel 3.3.2, S. 16, Gleichung 3.18)

$$\text{svd}(\mathbf{E}) = \mathbf{U} \cdot \text{diag}(d_1, d_2, 0) \cdot \mathbf{V}^T = \mathbf{U} \cdot \underbrace{\begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{d_3=0} \cdot \mathbf{V}^T \quad (5.44)$$

ergibt sich für die schiefsymmetrische Matrix  $\mathbf{S}$  die Gleichung 5.45.

Die Singulärwertzerlegung (engl. *Singular Value Decomposition*, kurz SVD) ist im Kapitel 3.2 auf S. 13 ausführlich beschrieben.

Bei der Singulärwertzerlegung von  $\mathbf{E}$  wird die Essentialmatrix in die drei Matrizen  $\mathbf{U}$  (Orthogonale Matrix),  $\mathbf{S}$  und  $\mathbf{V}$  aufgeteilt. Aus diesen drei Matrizen können die Singulärwerte  $d_1$ ,  $d_2$  und  $d_3$  erfasst werden. Eine Multiplikation der einzelnen Matrizen miteinander führt zu der ursprünglichen Essentialmatrix:  $\mathbf{E} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$ .

$$\mathbf{S} = \mathbf{U} \cdot \mathbf{Z} \cdot \mathbf{U}^T \quad (5.45)$$

Für die Rotation gibt es zwei mögliche Faktorisierungen, welche sich in der Transponierten von  $\mathbf{W}$  unterscheiden:

$$\mathbf{R}_1 = \mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^T \quad (5.46)$$

$$\mathbf{R}_2 = \mathbf{U} \cdot \mathbf{W}^T \cdot \mathbf{V}^T \quad (5.47)$$

Die Translation der Projektionsmatrix  $\mathbf{P}_2$  liegt im linken Nullraum<sup>18</sup> der Essentialmatrix und lässt sich als letzter Spaltenvektor von  $\mathbf{U}$  ermitteln:

$$\mathbf{t} = \mathbf{U} \cdot [0, 0, 1]^T \quad (5.48)$$

Das Vorzeichen der Essentialmatrix ist jedoch nicht eindeutig. Aufgrund dessen bestehen vier Möglichkeiten wie die Projektionsmatrix  $\mathbf{P}_2$  zusammengesetzt ist. Diese sind in den Gleichungen 5.49 bis 5.52 dargestellt.

$$\mathbf{P}_2 = [\mathbf{R}_1, \mathbf{t}] = \left[ [\mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^T], \mathbf{U} \cdot [0, 0, 1]^T \right] \quad (5.49)$$

$$\mathbf{P}_2 = [\mathbf{R}_1, -\mathbf{t}] = \left[ [\mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^T], -\mathbf{U} \cdot [0, 0, 1]^T \right] \quad (5.50)$$

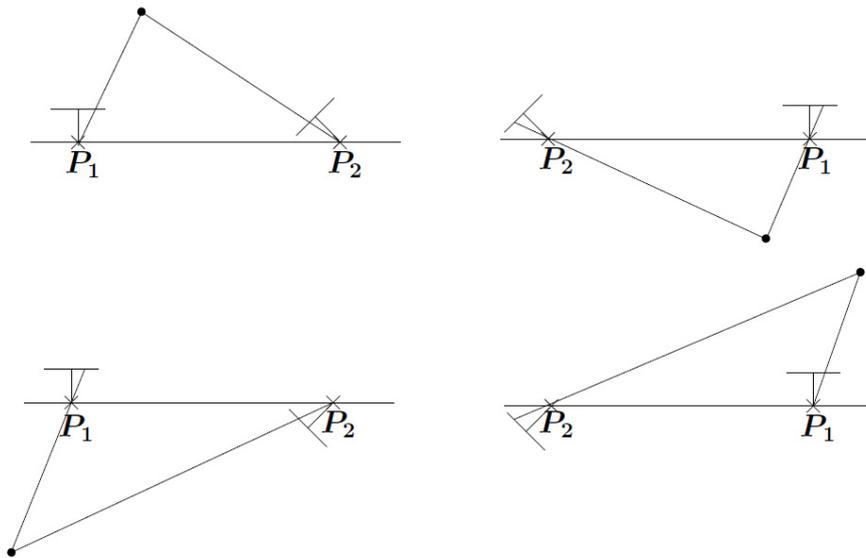
$$\mathbf{P}_2 = [\mathbf{R}_2, \mathbf{t}] = \left[ [\mathbf{U} \cdot \mathbf{W}^T \cdot \mathbf{V}^T], \mathbf{U} \cdot [0, 0, 1]^T \right] \quad (5.51)$$

$$\mathbf{P}_2 = [\mathbf{R}_2, -\mathbf{t}] = \left[ [\mathbf{U} \cdot \mathbf{W}^T \cdot \mathbf{V}^T], -\mathbf{U} \cdot [0, 0, 1]^T \right] \quad (5.52)$$

Die Abbildung 5.25 veranschaulicht die vier Lösungsmöglichkeiten der zweiten Projektionsmatrix. Bei den oberen beiden Grafiken ist im Gegensatz zu den Unteren die zweite Kamera um 180° entlang der Basislinie gedreht. Die linken beiden Grafiken stellen die positive Translation entlang der Basislinie dar. Im Gegensatz dazu besitzen die rechten beiden Grafiken eine negative Translation entlang der Basislinie.

Die Auswahl der richtigen Projektionsmatrix aus den Gleichungen 5.49 bis 5.52 bestimmt ein triangulierter Punkt. Denn befindet sich ein triangulierter Punkt vor den beiden Kamerazentren, folgt dass die verwendete Projektionsmatrix die Richtige ist. Aufgrund dessen muss mindestens eine zufällig gewählte 2D-Korrespondenz  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  trianguliert werden. Um die Situation der Verwendung einer fehlerbehafteten Korrespondenz oder eines falsch triangulierten Weltpunktes zu vermeiden, sollten alle Merkmalskorrespondenzen trianguliert und im Anschluss dessen der arithmetische Mittelwert zur Entscheidung der korrekten Projektionsmatrix herangezogen werden. Das Triangulationsverfahren und damit die Berechnung der 3D-Weltpunkte aus Punktkorrespondenzen wird im nächsten Abschnitt 5.3.3.2 erläutert.

<sup>18</sup>Der Nullraum einer Matrix sind alle Elemente, welche auf null bzw. das neutrale Element abgebildet werden. Des Weiteren wird der Nullraum auch als Kern bezeichnet.



**Abbildung 5.25:** Die Berechnung der zweiten Projektionsmatrix besitzt vier Lösungen. [HZ04]

### 5.3.3.2 Triangulation und Bündelblockausgleich

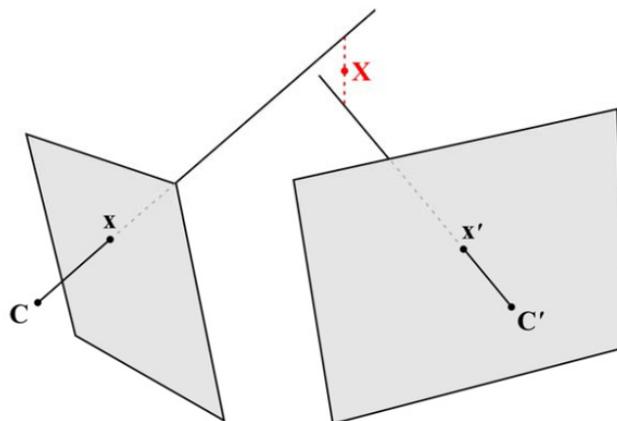
Sind die Projektionsmatrizen  $\mathbf{P}_1$  und  $\mathbf{P}_2$  vorhanden, dann kann der dreidimensionale Ursprung für alle Merkmalskorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  zurückgerechnet werden. Würden die Korrespondenzen keine geringfügigen Fehler beinhalten, könnten die Rücktransformationen

$$\mathbf{x}_i = \mathbf{P}_1 \cdot \mathbf{X}_i \quad (5.53)$$

und

$$\mathbf{x}'_i = \mathbf{P}_2 \cdot \mathbf{X}_i \quad (5.54)$$

direkt für die Berechnung der 3D-Punkte  $\mathbf{X}$  verwendet werden. Dies ist jedoch im Normalfall nicht möglich, da sich die Projektionsstrahlen nicht in einem gemeinsamen Punkt schneiden (Abbildung 5.26).



**Abbildung 5.26:** Projektionsstrahlen schneiden sich nicht in einem gemeinsamen Weltpunkt  $\mathbf{X}$ . [ROD04]

Damit erfüllt die Projektion eines Weltpunktes nicht die Epipolarbedingung  $\mathbf{x}'^T \cdot \mathbf{F} \cdot \mathbf{x} = 0$  (Siehe S. 15, Gleichung 3.11). Aus diesem Grund sind mit Hilfe der Projektionsmatrizen die Merkmalskorrespondenzen gesondert zu triangulieren<sup>19</sup>. Die Triangulation erfolgt mit einer Minimierung des Rückprojektionsfehlers

$$\min_{\mathbf{X}_i} \left[ \|\mathbf{P}_1 \cdot \mathbf{X}_i - \mathbf{x}_i\|^2 \cdot \|\mathbf{P}_2 \cdot \mathbf{X}_i - \mathbf{x}'_i\|^2 \right] \quad (5.55)$$

eines 3D-Weltpunktes  $\mathbf{X}_i$  zu seinen zwei Bildpunkten  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  (Vgl. [EBE09], S. 39). Nachfolgend kann die Gleichung 5.55 in ein homogenes Gleichungssystem umgeformt und mit der Singulärwertzerlegung gelöst werden.

Das Ergebnis der linear homogenen Triangulation sind die gesuchten ersten 3D-Weltpunkte von einem Gegenstand. Im Gegensatz zur linear homogenen Triangulation existiert eine alternative Möglichkeit die Weltpunkte zu berechnen. Anstatt den Rückprojektionsfehler zu minimieren, kann die Lage der Bildpunkte

$$\min_{\mathbf{x}_i, \mathbf{x}'_i} \left[ L(\mathbf{x}_i, \mathbf{l}_i)^2 + L(\mathbf{x}'_i, \mathbf{l}'_i)^2 \right] \quad (5.56)$$

bezüglich der Epipolargeometrie korrigiert werden. Dabei kennzeichnet  $L$  den senkrechten euklidischen Abstand des Punktes  $\mathbf{x}_i$  bzw.  $\mathbf{x}'_i$  zu seiner zugehörigen Epipolarlinie  $\mathbf{l}_i$  bzw.  $\mathbf{l}'_i$ . Die Abbildung 5.27 veranschaulicht die Fehlerminimierung im Bildbereich. Detaillierte Informationen zum alternativen Verfahren sind in [HZ04] ab S. 315 beschrieben.

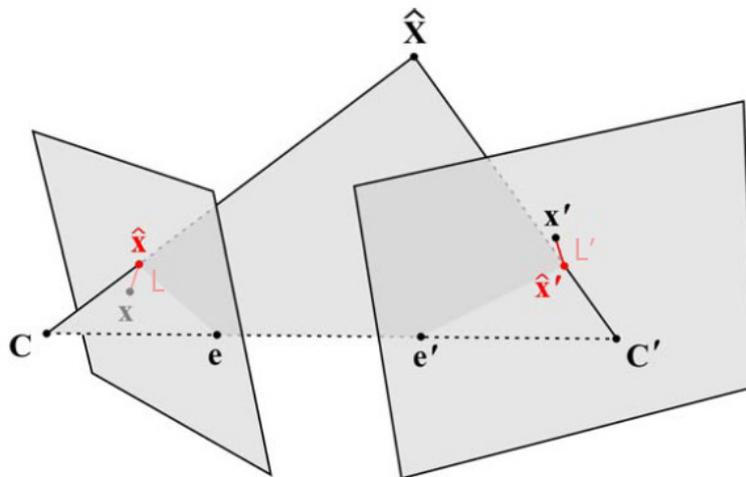
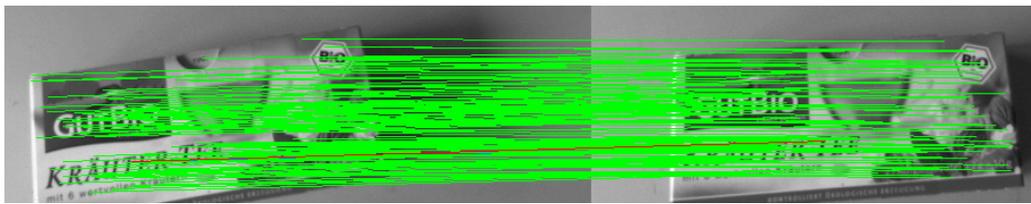


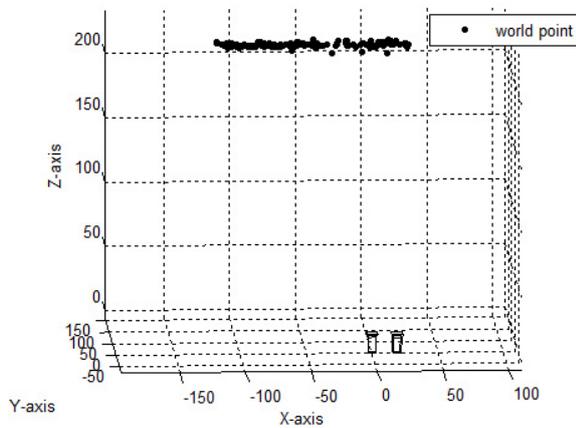
Abbildung 5.27: Korrektur der Bildpunkte. [ROD04]

Die triangulierten Weltpunkte sind in der folgenden Abbildung 5.28 dargestellt.

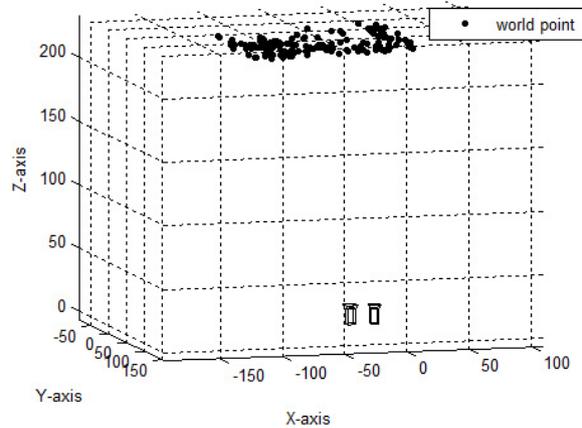
<sup>19</sup>Die Korrektur der Weltpunkte sowie der Projektionsstrahlen bzw. Projektionsmatrizen wird als sogenannter *Sparse Bundle Adjustment* bezeichnet. Dafür sind die rekonstruierten 3D-Weltpunkte zu ihren Bildpunkten zu projizieren. Die Summe der quadrierten Differenz, zwischen Merkmalspunkt und deren Rückprojektion, gibt Aufschluss über die Optimierungsmöglichkeiten. Für weitere Informationen ist auf [TMHF00] und [HZ04] verwiesen.



(a) Verwendete 2D-Korrespondenzen



(b) 3D-Punkte (Perspektive 1)



(c) 3D-Punkte (Perspektive 2)

**Abbildung 5.28:** Ermittelte 3D-Weltpunkte.

An diesem Punkt angelangt, werden alle triangulierten Weltpunkte noch einmal verifiziert. Einerseits werden die Weltpunkte auf einer näherungsweisen planaren Fläche untersucht um eine Strukturdegeneration auszuschließen (Vgl. Kapitel 5.3.2.3 ab S. 50). In dem Fall, dass keine Strukturdegeneration vorliegt erfolgt eine weitere Überprüfung eines Weltpunktes auf Basis seiner Raumposition im Verhältnis zu allen anderen Weltpunkten.

Der Gültigkeitsbereich wird folgendermaßen berechnet. Zuerst wird der Schwerpunkt der Punktmenge ermittelt. Dafür wurden zwei Abstandsmaße evaluiert. Der arithmetische Mittelwert erzielte, aufgrund seiner starken Auswirkung auf Ausreißer, schlechte Ergebnisse. Dagegen bewies sich der Median, welcher auch als Zentralwert bekannt ist, als Ausreißerresistent und damit als geeigneter Schwerpunkt. Der Median

$$\tilde{\mathbf{X}} = \begin{cases} \mathbf{X}_{\frac{n+1}{2}}^{\text{sort}} & n \text{ ungerade} \\ \frac{1}{2} \cdot (\mathbf{X}_{\frac{n}{2}}^{\text{sort}} + \mathbf{X}_{\frac{n+1}{2}}^{\text{sort}}) & n \text{ gerade} \end{cases} \quad (5.57)$$

gibt die Grenze zwischen zwei Hälften einer geordneten Menge  $\mathbf{X}^{\text{sort}}$  an.

Nachdem der Schwerpunkt der Weltpunkte bestimmt wurde, ist der arithmetische Mittelwert zwischen den Weltpunkten und dem Schwerpunkt zu berechnen. Dieser Mittelwert

$$\bar{\mathbf{d}}_{\text{arith}} = \frac{1}{n} \cdot \sum_{i=1}^n |\mathbf{x}_i^{\text{sort}} - \tilde{\mathbf{X}}| \quad (5.58)$$

ermöglicht die Spezifikation eines Gültigkeitsbereiches ausgehend vom Schwerpunkt (Abbildung 5.29a). Des Weiteren wird dieser mit einem Faktor  $\alpha$  verstärkt, welcher die Größe des Gültigkeitsbereiches beeinflusst. Die triangulierten Weltpunkte, welche sich innerhalb des Gültigkeitsbereiches befinden, werden nicht aussortiert. Die Abbildung 5.29 veranschaulicht diese Überprüfung. Der ermittelte Schwerpunkt (Median) der Weltpunktmenge ist durch ein  $x$  gekennzeichnet. Der arithmetische Mittelwert der Weltpunktmenge wird mit einem schwarz ausgefalteten Dreieck dargestellt. Der Gültigkeitsbereich ist durch einen grauen Quader visualisiert. Grün dargestellte Weltpunkte liegen innerhalb des Gültigkeitsbereiches, dagegen sind rote Weltpunkte als fehlerhaft verifiziert worden.

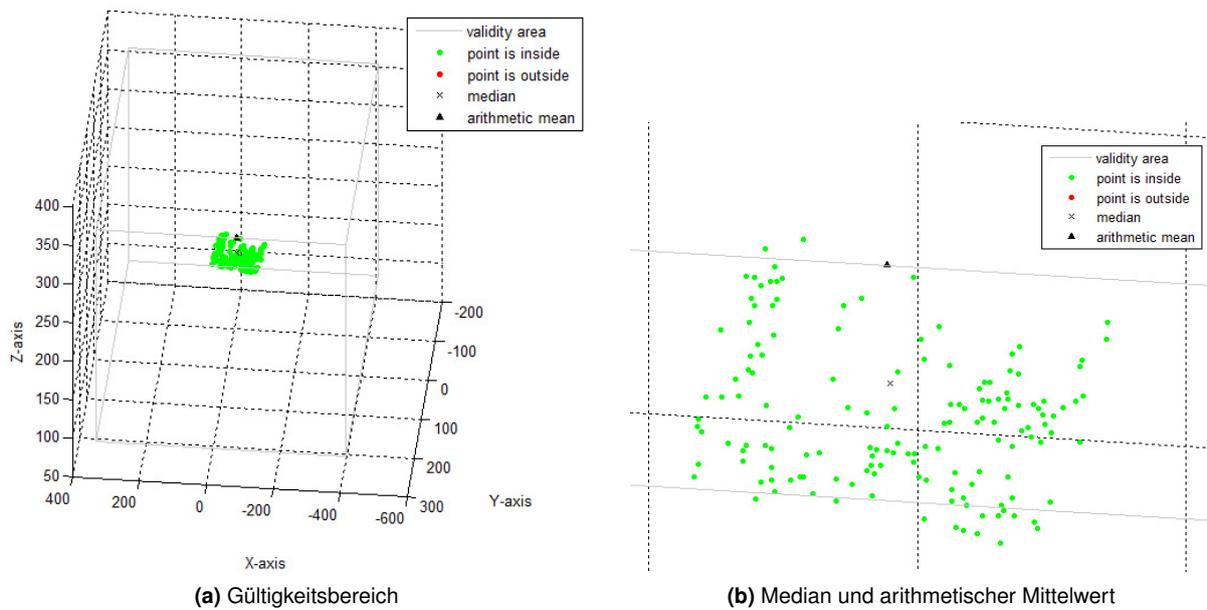


Abbildung 5.29: Detektion falsch triangulierter Weltpunkte.

Der prozentuale Anteil  $\mathbf{X}_{\text{out}[\%]}$ , der  $k$  fehlerhaft triangulierten Weltpunkte im Verhältnis zu der gesamten Triangulationsmenge, ermöglicht die Erkennung ob die Triangulation komplett fehlgeschlagen ist.

$$\mathbf{X}_{\text{out}[\%]} = \frac{100}{n} \cdot k \quad (5.59)$$

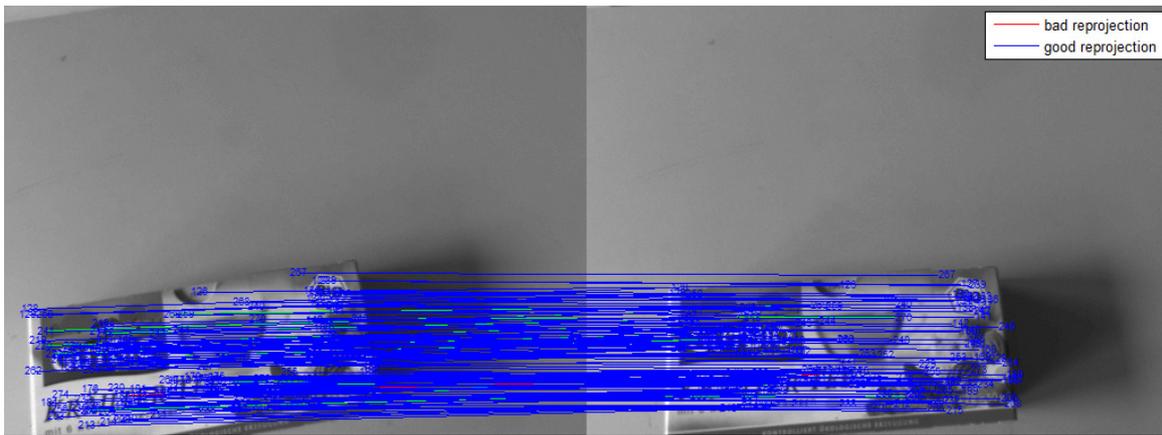
Übersteigt der prozentuale Anteil  $\mathbf{X}_{\text{out}[\%]}$  einen gewissen Schwellwert, sind alle triangulierten Weltpunkte zu verwerfen. Eine erneute Merkmalsuche oder ein wiederholter Merkmalsvergleich mit anderen Such- oder Vergleichsparametern führt üblicherweise zu einer anschließend erfolgreichen Triangulation. Alternativ kann das aktuell verwendete Aufnahme paar  $\mathbf{B}_1 + \mathbf{B}_2$  verworfen werden. Der Gebrauch von  $\mathbf{B}_1 + \mathbf{B}_3$ ,  $\mathbf{B}_2 + \mathbf{B}_3$  oder  $\mathbf{B}_3 + \mathbf{B}_4$  führten im Praxisversuch zu einer nachfolgend erfolgreichen Triangulationsmenge.

Im letzten Schritt müssen jedem 3D-Weltpunkt seine Merkmale zugeordnet werden. Dies ermöglicht in einem späteren Merkmalsvergleich, welcher zwischen zwei Bildern durchgeführt wird, dass der dreidimensionale Ursprung und damit die räumliche Lage eines Merkmales bekannt sind.

### 5.3.3.3 Filterung anhand Rückprojektion

Die Eliminierung der triangulierten Weltpunkte, die einen zu großen Rückprojektionsfehler aufweisen, sollte vorgenommen werden. Denn sind weitere Aufnahmen und damit neue Weltpunkte sowie Merkmale in die bestehende Menge aller Weltpunkt zu integrieren, wird auf die aktuelle Merkmals- und Weltpunktmenge Bezug genommen. Aus diesem Grund sollten nur Weltpunkte beibehalten werden, welche auf die Projektionsmatrix bestmöglich abgestimmt sind.

Die nachfolgende Abbildung 5.30 veranschaulicht die Filterung anhand des Rückprojektionsfehlers. Die Verbindung der Merkmalskorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  sind grün dargestellt. Die blaue bzw. rote Linie zeigt die Verbindung zwischen den rückprojizierten Weltpunkten. Steigt der Rückprojektionsfehler über einen gewissen Schwellwert, wird diese Linie rot, andernfalls blau dargestellt.



**Abbildung 5.30:** Filterung von Weltpunkten anhand des Rückprojektionsfehlers.

#### 5.3.3.4 Korrektur der Skalierung

Die Triangulation ist bis auf einen Skalierungsfaktor  $scale$  genau. Aus diesem Grund müssen die initialen Weltpunkte  $\mathbf{X}$  richtig skaliert werden.

Zum einen kann der Skalierungsfaktor manuell bestimmt werden. Dafür ist der Abstand  $\mathbf{d}_{\text{known}}$  von zwei divergenten Oberflächenpunkten  $\mathbf{X}_i$  und  $\mathbf{X}_j$  auszumessen. Danach muss der aktuelle Abstand dieser Weltpunkte mit

$$\mathbf{d} = \mathbf{X}_i - \mathbf{X}_j \quad (5.60)$$

und

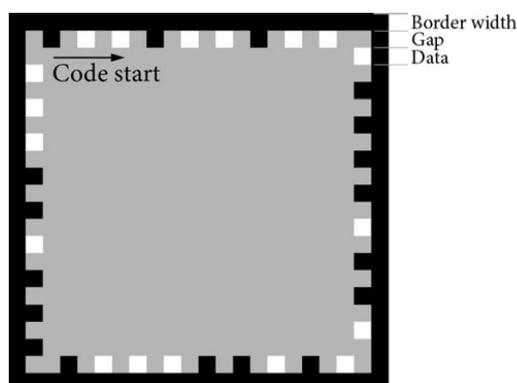
$$\mathbf{d}_{\text{tria}} = \sqrt{d_x^2 + d_y^2 + d_z^2} \quad (5.61)$$

berechnet werden. Anschließend kann der bekannte- durch den aktuellen Abstand dividiert werden.

$$scale = \mathbf{d}_{\text{known}} / \mathbf{d}_{\text{tria}} \quad (5.62)$$

Das Ergebnis ist der unbekannte Skalierungsfaktor.

Zum anderen ist der Skalierungsfaktor auch autonom berechenbar. Dafür wird für die ersten beiden Kameraaufnahmen ein sogenannter *Frame-Marker*



**Abbildung 5.31:** Binärkodierter Frame-Marker. [WLS08]

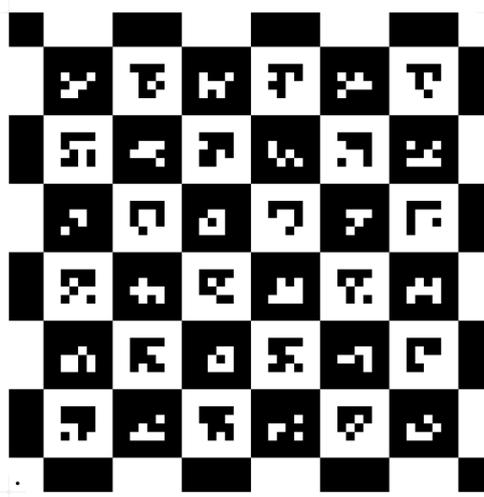
neben den zu vermessenden Gegenstand positioniert. Die Maße des Kalibrierungsmusters sind dabei bekannt. Bei der Triangulation (Vgl. Kapitel 5.3.3.2) sind dann die vier äußeren Marker-Eckpunkte bzw. deren Korrespondenzen zusammen mit den anderen natürlichen Merkmalskorrespondenzen des Gegenstandes zu triangulieren. Die vier Eckpunkte werden anhand einer Binärcodierung am Innenrand des Markers erkannt und indiziert. Dafür besitzt jeder Rand neun Quadrate, welche entweder schwarz oder weiß sind. Insgesamt verfügt damit ein Marker über  $4 \cdot 9 = 36$  Quadrate, wobei jeweils neun eine Ecke indizieren. Daraus ergeben sich  $2^9 = 512$  Kombinationsmöglichkeiten zur Indizierung. Ein Vorteil bei diesem Marker ist die Invarianz gegenüber einer Teilverdeckung. Denn Quadrate werden im Uhrzeigersinn interpretiert, sodass lediglich eine vollständige Seite mit jeweils den neun Quadraten detektiert werden muss, um alle vier Ecken zu indizieren. Die eindeutige Zuordnung der Eckpunkte ermöglicht eine robuste Eckpunktkorrespondenzsuche in den ersten beiden Kameraaufnahmen.

Die triangulierten vier Eckpunkte können nachfolgend mit ihren bekannten Maßen verglichen werden. Der gesuchte Skalierungsfaktor lässt sich somit durch

$$scale = \frac{1}{4} \cdot \sum_{i=1}^4 [d_{\text{known}_i} / d_{\text{tria}_i}] \quad (5.63)$$

ermitteln.

In dieser Arbeit wurde ein drittes Verfahren implementiert, welches sich als robust und sehr genau herausgestellt hat. Dieses ist ähnlich dem Vorherigen ein autonomes Verfahren und benötigt aus diesem Grund keinen manuellen Benutzereingriff. Als Grundlage dient das nachfolgend dargestellte Kalibrierungsmuster 5.32, welches aus  $6 \times 6 = 36$  binärcodierten Markern entworfen wurde.



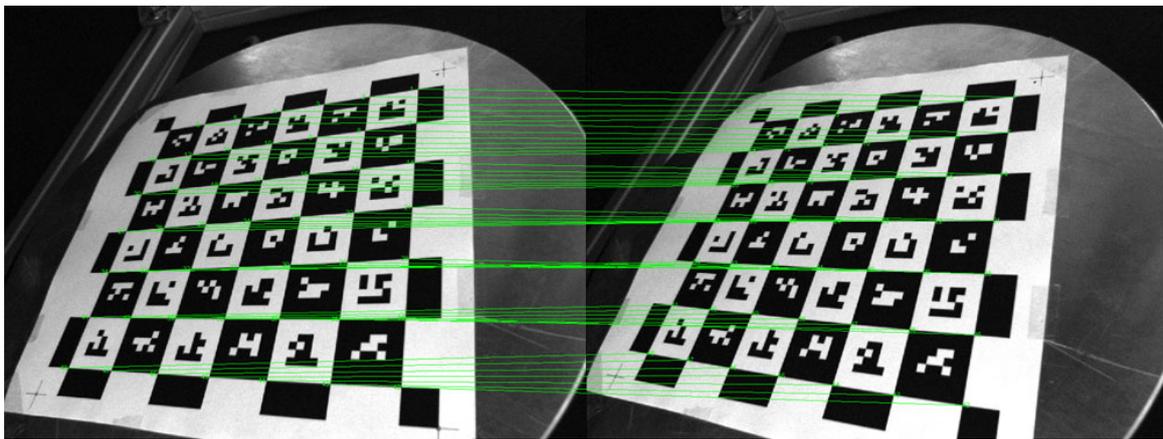
**Abbildung 5.32:** Binärcodiertes Markerfeld mit bekannter Geometrie.

Dabei sind die Anordnung, die Binärcodierung und die Maße jedes Markers bekannt. Das ermöglicht eine fehlerfreie Detektion sowie Indizierung von insgesamt  $4 \times 36 = 144$  Eckpunkten in jeder Kameraaufnahme<sup>20</sup>. Damit sind in jeweils zwei divergenten Aufnahmen 144 Korrespondenzen vorhanden.

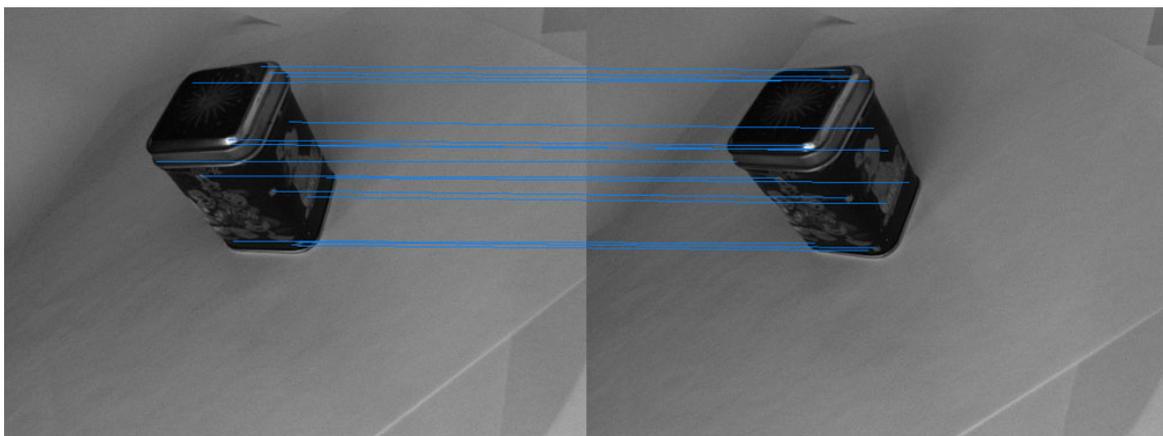
Diese 144 Korrespondenzen werden zusammen mit der ermittelten Korrespondenzmenge der natürlichen Merkmale vom Gegenstand trianguliert. Dabei ist zu beachten, dass die zwei Aufnahmen vom  $6 \times 6$  Markerfeld (Abbildung 5.33a) aus der gleichen Perspektive wie die zwei Gegenstandsaufnahmen (Abbildung 5.33b) aufzunehmen sind. Das ist beispielsweise mit einem Drehteller leicht zu realisieren, indem zuerst ein Gegenstand und anschließend das  $6 \times 6$  Markerfeld in gleichen Rotationsabständen aufgenommen werden.

Aufgrund der bekannten Markeranordnung muss der genannte Aufnahmevorgang nicht, wie die Abbildung 5.33 darstellt, getrennt ablaufen. Das in dieser Arbeit implementierte Verfahren detektiert auch komplett verdeckte Marker anhand einer Schätzung. Die Schätzung ist aufgrund der bekannten Maße sehr genau. Dadurch kann

<sup>20</sup>In dem Fall, dass ein Eckpunkt nicht doppelt vorhanden sein soll, ergeben sich  $7 \times 7 = 49$  Eckpunkte.



(a) Korrespondenzen des Markerfeldes



(b) Korrespondenzen vom Gegenstand

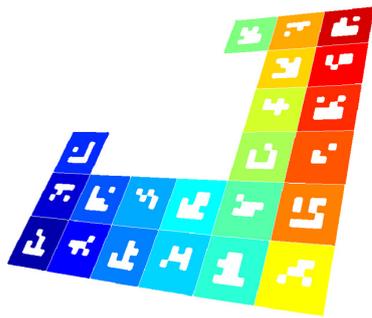
**Abbildung 5.33:** Marker und natürliche Korrespondenzen. Jeweils aus den zwei gleichen Perspektiven aufgenommen.

das Markerfeld gleichzeitig mit der Suche nach natürlichen Merkmalen am Gegenstand detektiert werden. Die Abbildung 5.34 veranschaulicht in drei Phasen eine Detektion von Markern und die Schätzung von verdeckten Markern.

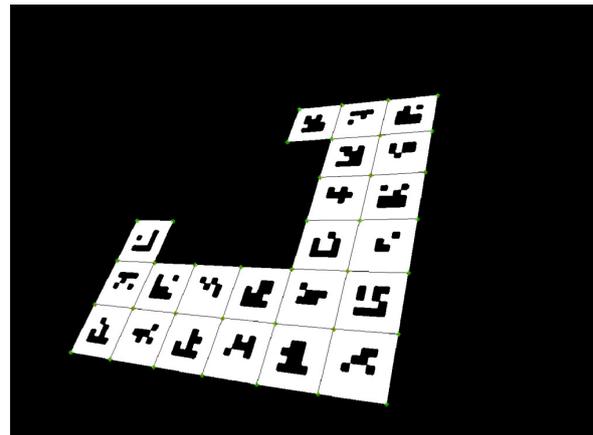
In der ersten Phase werden alle sichtbaren Marker in einer Kameraaufnahme kartiert. Dafür wird die Aufnahme in ein Grauwertbild konvertiert um anschließend die Kanten mit dem Sobel-Filter zu extrahieren. Danach muss das gefilterte Bild von Rauschen u. ä. befreit werden. Jeder Pixel mit dem näherungsweisen Grauwert von null wird invertiert, wenn dieser zwei Nachbarpixel, welche nicht Verbunden sind sowie einen Grauwert ungleich null besitzen. Anschließend erfolgt die Anwendung eines Schwellwertverfahrens mit einem 3 x 3 Median-Filter. Danach wird jeder isolierte Pixel entfernt bzw. auf null gesetzt. Abschließend erfolgt eine Invertierung des Kantenbildes und eine Kartierung der Marker (Grafik 5.34a).

Die zweite Phase beschäftigt sich mit der Suche nach den vier äußersten Eckpunkten von jedem Marker (Grafik 5.34b). Dafür sind die Eckpunkte anhand des Kantenbildes zu finden. Der Schnittpunkt einer vertikalen mit einer horizontalen Gerade liefert eine ungefähre Lokalisierung der Eckpunkte. Anschließend verbessert die Suche nach Sattelpunkten die Lokalisierung subpixelgenau. Zu diesem Zeitpunkt wurden alle in der Kameraaufnahme sichtbaren äußeren Eckpunkte jedes Markers erkannt.

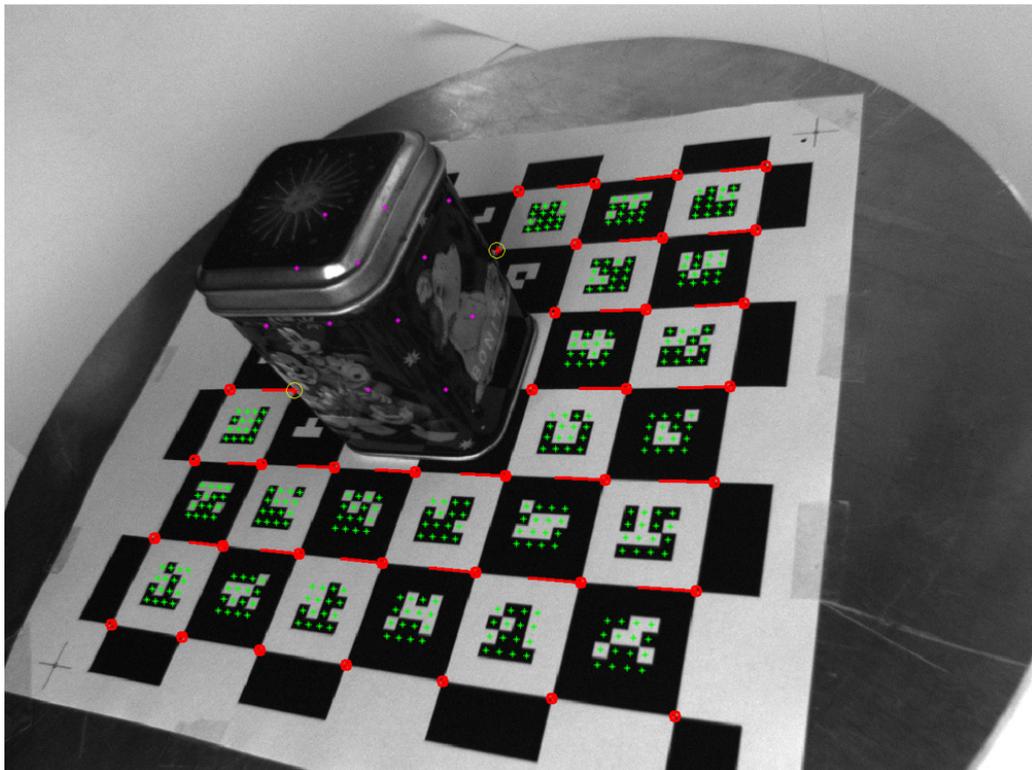
In der dritten Phase werden auf Basis aller Eckpunkte die Binärkodierungen entschlüsselt, die sichtbaren Marker indiziert und abschließend die verdeckten Marker geschätzt. Dabei ist die Schätzung aufgrund der bekannten Maße und Anordnung des Markerfeldes sehr robust. Die Grafik 5.34c stellt die sicher detektierten und geschätzten Marker mit ihren Eckpunkten dar. In jedem sichtbaren Marker wurde jedes Binärfeld des markerspezifischen Binärkodes mit einem grünen Symbol gekennzeichnet. Ein weißes Kästchen entspricht einer eins, ein schwarzes



(a) Kantenextraktion + Kartierung



(b) Äußere Markereckpunkte



(c) Markerdetektion + Schätzung verdeckter Marker

**Abbildung 5.34:** Detektion von Markern trotz Teilverdeckung.

Kästchen einer null. Aus der Grafik ist damit ersichtlich, dass ein Marker durch eine  $4 \times 4 = 16$  Binärcodierung indiziert wird. Die roten Symbole kennzeichnen die detektierten Eckpunkte der sichtbaren Marker. Die lila-farbenden Symbole stellen die geschätzten Eckpunkte der verdeckten Marker dar.

Der beschriebene Algorithmus wurde in dieser Arbeit adaptiert und ist in [AHH10] zu finden.

Für die abschließende Berechnung des Skalierungsfaktors sind die triangulierten  $k \in [1 \dots 144]$  Eckpunkte mit ihren bekannten Maßen zu vergleichen. Der gesuchte Skalierungsfaktor lässt sich gleichermaßen nach der Formel 5.63 berechnen.

Anmerkend sei an dieser Stelle ein weiterer Vorteil der Verwendung eines Markerfeldes beschrieben. In dieser Arbeit wurde im Kapitel 5.3.2.3 erwähnt, dass eine Strukturdegeneration auftritt, wenn koplanare Merkmalskorrespondenzen trianguliert werden. Da üblicherweise mehr als eine natürliche Merkmalskorrespondenz vom Gegenstand detektiert wird, ist ein Auftreten einer Strukturdegeneration ausgeschlossen. Die Merkmalskorrespondenzen des Markerfeldes liegen auf einer Ebene ungleich des Gegenstandes. Aus diesem Grund sind

Merkmalskorrespondenzen auf unterschiedlichen Ebenen vorhanden, wodurch eine Strukturdegeneration unmöglich ist.

In diesem Unterkapitel wurden drei Verfahren für die Bestimmung des Skalierungsfaktors vorgestellt. Im Nachfolgenden wird die Skalierungskorrektur anhand des Skalierungsfaktors erläutert. Die Korrektur der Skalierung der Weltpunkte erfolgt durch die in Gleichung 5.64 dargestellte Multiplikation des Skalierungsfaktors mit allen Weltpunkten.

$$\hat{\mathbf{X}}_i = \mathbf{X}_i \cdot scale \quad (5.64)$$

Die zwei Projektionsmatrizen  $\mathbf{P}_1$  und  $\mathbf{P}_2$  der Form

$$\mathbf{P}_{3 \times 4} = \left[ \begin{array}{c|c} \mathbf{P}_R & \mathbf{P}_t \\ \hline \underbrace{3 \times 3 \text{ Rotationsanteil}} & \underbrace{3 \times 1 \text{ Translationsanteil}} \end{array} \right] \quad (5.65)$$

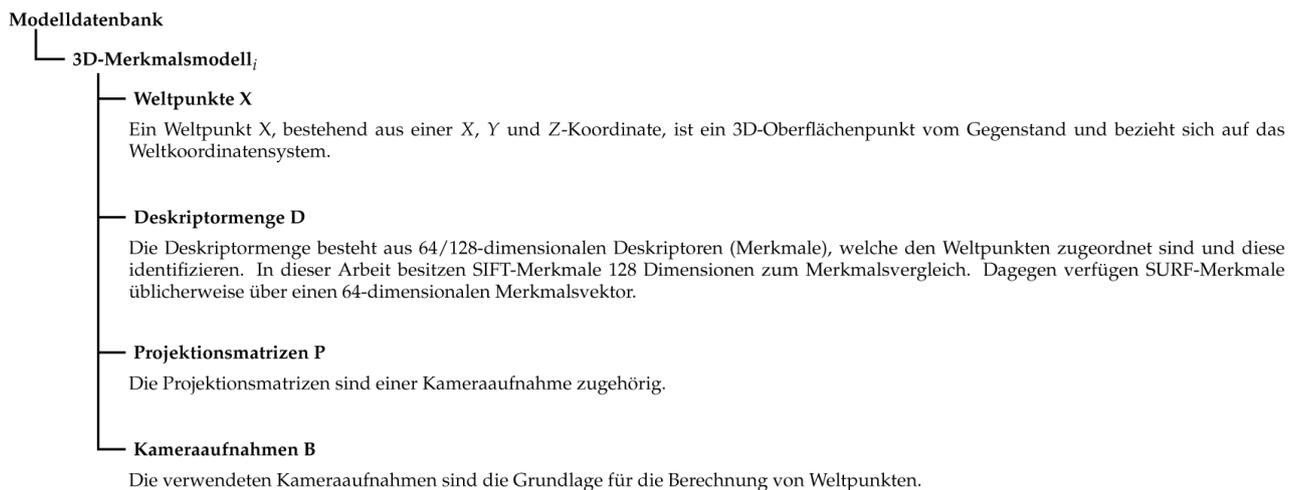
sind mit

$$\hat{\mathbf{P}} = [\mathbf{P}_R \mid \mathbf{P}_t \cdot scale] \quad (5.66)$$

zu korrigieren. In der Gleichung 5.66 ist ersichtlich, dass lediglich der Translationsanteil mit dem Skalierungsfaktor multipliziert werden muss.

### 5.3.4 Anlegen eines 3D-Merkmalmodells

Eine Datenbank mit allen Weltpunkten sowie allen Deskriptoren (Merkmalen), welche die Weltpunkte eindeutig kennzeichnen, ist für eine robuste Identifizierung und Lokalisierung von Gegenständen notwendig. Der Aufbau der Modelldatenbank ist in Abbildung 5.35 dargestellt. Die Modelldatenbank ist in 3D-Merkmalmodelle untergliedert, welche jeweils einen Gegenstand zugehörig sind. Jedes 3D-Merkmalmodell beinhaltet sämtliche Oberflächenpunkte mit den zugehörigen Deskriptoren, die Projektionsmatrizen und die Kameraaufnahmen.



**Abbildung 5.35:** Aufbaustruktur der Modelldatenbank.

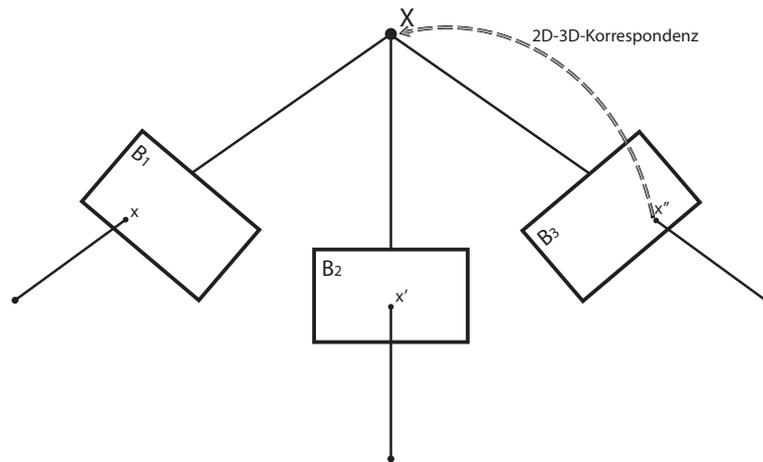
## 5.4 Weltpunkte aus weiteren Aufnahmen

Dieses Kapitel beschreibt die Extraktion und Einreihung von Weltpunkten sowie Merkmalen, welche sich in weiteren Aufnahmen befinden.

Für einen hohen Wiedererkennungsgrad eines Gegenstandes sollten Merkmale aus weiteren Aufnahmen in das bereits bestehende 3D-Merkmalmodell hinzugefügt werden. In dieser Arbeit wurden zwei Ansätze, zur Erweiterung der bereits im Merkmalsmodell bestehenden Merkmale, entwickelt und evaluiert.

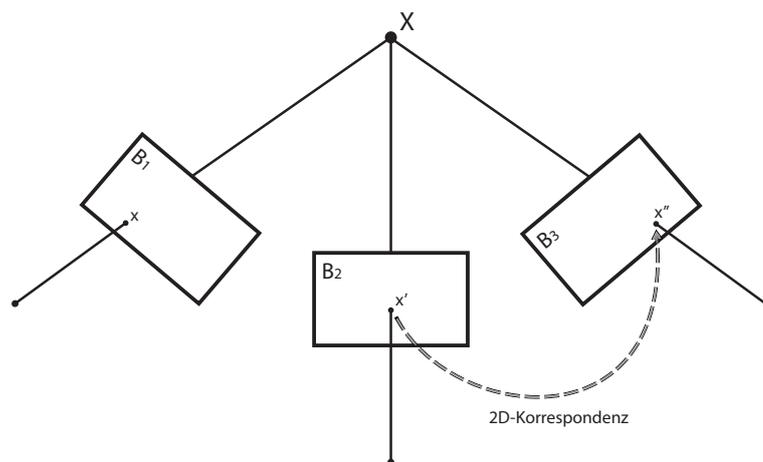
Beide Verfahren werden nachfolgend kurz erläutert.

### 1. Methode



**Abbildung 5.36:** Merkmalsvergleich zwischen einem Bild und einem 3D-Merkmalmodell.

Bei der ersten Methode ist eine zu den ersten zwei Bildern ( $\mathbf{B}_1, \mathbf{B}_2$ ) perspektivisch abweichende Aufnahme  $\mathbf{B}_3$  aufzunehmen. Um neue Weltpunkte für das 3D-Merkmalmodell berechnen zu können, wird zuerst die Projektionsmatrix  $\mathbf{P}_3$  des aktuellen Bildes  $\mathbf{B}_3$  benötigt. Wird ein Merkmalsvergleich zwischen der Aufnahme  $\mathbf{B}_3$  und den Merkmalen aller Weltpunkte durchgeführt, kann auf Basis dieser 3D-Welt- zu 2D-Bild-Korrespondenzen die Projektionsmatrix  $\mathbf{P}_3$  berechnet werden (Abbildung 5.36). Beispielsweise ermöglichen die Verfahren DLT, EPnP oder POSIT die Berechnung von  $\mathbf{P}_3$  mit Hilfe der 3D-2D-Korrespondenzen. Diese werden im Kapitel 6 ab Seite 70 beschrieben.



**Abbildung 5.37:** Merkmalsvergleich zwischen zwei Bildern.

Im Anschließenden sind die Merkmale zwischen der aktuellen Aufnahme  $\mathbf{B}_3$  und der vorherigen Aufnahme

$\mathbf{B}_2$  zu vergleichen (Abbildung 5.37) und gegebenenfalls mit einer weiteren Aufnahme  $\mathbf{B}_4$  zu verifizieren<sup>21</sup>. Alle Merkmalskorrespondenzen welche zu einem noch nicht vorhandenen Weltpunkt führen, sind mit dem Triangulationsverfahren (Siehe *Berechnung von 3D-Weltpunkten* auf S. 54) zu berechnen.

Wurden viele Aufnahmen flächendeckend um einen Gegenstand aufgenommen, ist ein komplettes 3D-Merkmalmodell vom Gegenstand möglich. Grundsätzlich gilt, je mehr Merkmale im 3D-Modell vorhanden sind, desto höher ist die Wahrscheinlichkeit, dass die räumliche Position eines Gegenstandes später in einer Aufnahme zurückgerechnet werden kann.

Die erste Methode kennzeichnet sich durch die vier nachfolgenden Punkte aus.

- Die Projektionsmatrix kann direkt aus den 3D-2D-Korrespondenzen berechnet werden. In dem Fall, dass die 3D-Weltkoordinaten fehlerbehaftet sind, wird die Projektionsmatrix falsch berechnet. Der resultierende Fehler wird sich in alle nachfolgenden Berechnungen, wie beispielsweise der Projektionsmatrizen  $\mathbf{P}_{p+1,p+2,\dots,p+n}$  oder der Weltpunkte  $\mathbf{X}_{i+1,i+2,\dots,i+m}$ , übertragen.
- Die Berechnung von Weltpunkten erfolgt durch eine Triangulation, welche bis auf einen unbekanntem Skalierungsfaktor genau ist. Aus diesem Grund muss eine Korrektur der Skalierung erfolgen. Die Projektionsmatrizen  $\mathbf{P}_{p,p>2}$  werden anhand der 3D-2D-Korrespondenz berechnet. Aus diesem Grund erben die Projektionsmatrizen  $\mathbf{P}_{p,p>2}$  die richtige Skalierung. Deswegen müssen lediglich die ersten zwei Projektionsmatrizen  $\mathbf{P}_1, \mathbf{P}_2$  und die Weltpunkte der initialen Triangulation korrigiert werden.
- Die Aufnahmen sind aneinander gebunden. Ein neu einzureihendes Bild wird immer mit der letzten Aufnahme verglichen. Dadurch kann es vorkommen, dass das Verfahren plötzlich keine Aufnahmen mehr einreihen kann. Der genannte Fall tritt auf, wenn eine einzureihende Aufnahme  $\mathbf{B}_{b+1}$  eine zu große perspektivische Abweichung zur vorgehenden Aufnahme  $\mathbf{B}_b$  aufweist. In dem Fall werden zu wenige oder sogar keine Merkmalskorrespondenzen gefunden. Da üblicherweise die nachfolgenden Aufnahmen  $\mathbf{B}_{b+2,b+3,\dots,b+n}$  noch perspektivisch unterschiedlicher sind, können keine Merkmalskorrespondenzen gefunden und damit keine weiteren Weltpunkte berechnet werden.
- Pro Iterationsschritt wird eine neue Aufnahme verarbeitet. In dem Fall, dass eine Korrespondenzmenge verifiziert werden soll, wird eine zusätzliche Aufnahme benötigt.

## 2. Methode

Die zweite Methode ist ähnlich der Ersten konzipiert, jedoch unterscheidet sich diese durch die nachfolgenden fünf Merkmale.

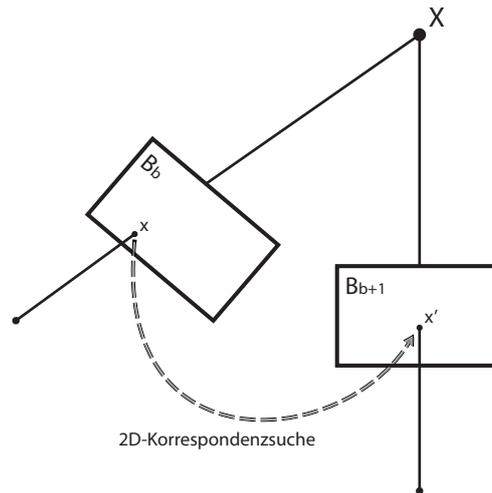
- Jeweils die (verifizierten) Bild-Korrespondenzen zwischen zwei Aufnahmen  $\mathbf{B}_b$  und  $\mathbf{B}_{b+1}$  bestimmen die Projektionsmatrizen.
- Die Berechnung von Weltpunkten erfolgt durch eine Triangulation, welche bis auf einen unbekanntem Skalierungsfaktor genau ist. Aus diesem Grund muss eine Korrektur der Skalierung erfolgen.
- Die Projektionsmatrizen  $\mathbf{P}_{p,p>2}$  stehen in keiner Relation zu den ersten beiden  $\mathbf{P}_{p \in \{1,2\}}$ . Aus diesem Grund ist die Kamerabewegung von  $\mathbf{P}_{p-1,p-2,\dots,p-n}$  nicht in den Projektionsmatrizen  $\mathbf{P}_{p,p>2}$  enthalten, sodass die triangulierten Weltpunkte korrigiert werden müssen. Die Korrektur erfolgt durch eine 3D-Korrespondenzsuche zwischen den neu triangulierten Weltpunkten und den bisher eingespeisten Weltpunkten des 3D-Merkmalmodells.
- Die Bildreihenfolge ist unabhängig voneinander.
- Es werden pro Iterationsschritt zwei neue Aufnahmen analysiert. In dem Fall, dass eine Korrespondenzmenge verifiziert werden soll, wird eine zusätzliche Aufnahme benötigt.

<sup>21</sup>Grundsätzlich sind fehlerfreie Korrespondenzen in zwei Bildern zu finden. In dieser Arbeit werden die Korrespondenzen  $\{\mathbf{B}_2 \leftrightarrow \mathbf{B}_3\}$  aus Sicherheitsgründen durch eine weitere Aufnahme  $\mathbf{B}_4$  verifiziert. Die Verifikation ergibt sich aus der Schnittmenge  $\{\mathbf{B}_2 \leftrightarrow \mathbf{B}_3\} \cap \{\mathbf{B}_3 \leftrightarrow \mathbf{B}_4\}$ .

Die Auswertung der beiden Verfahren zeigte im Praxiseinsatz, dass die zweite Methode im Hinblick auf die Fehleranfälligkeit und Präzision dem ersten Verfahren überlegen ist. Des Weiteren ist das erste Verfahren leider abhängig von der Bildreihenfolge. Denn das nächste hinzugefügte Bild bzw. deren Merkmale beziehen sich auf die letzte Aufnahme.

Aus diesen Gründen ist die zweite Methode zu bevorzugen und wird nachfolgend detailliert beschrieben.

#### 5.4.1 Suche nach Merkmalskorrespondenzen

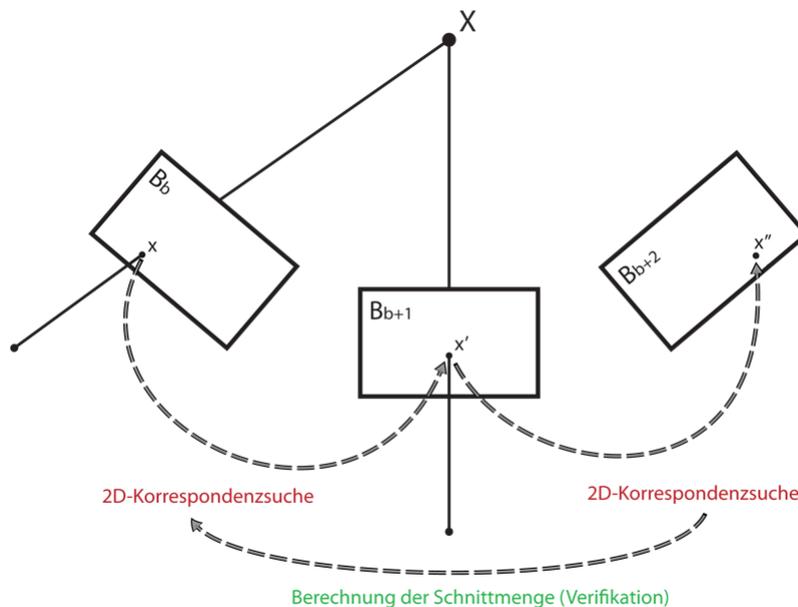


**Abbildung 5.38:** Suche nach Merkmalskorrespondenzen zwischen zwei Bildern.

Zu Beginn müssen die vorhandenen Merkmalskorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  zwischen den zwei neuen Aufnahmen  $\mathbf{B}_{b,b>2}$  und  $\mathbf{B}_{b+1,b>2}$  ermittelt werden (Abbildung 5.38).

Des Weiteren kann zur Verifikation der Merkmalskorrespondenzen die darauffolgende Korrespondenzmenge  $\{\mathbf{B}_{b+1,b>2} \leftrightarrow \mathbf{B}_{b+2,b>2}\} \rightarrow \{\mathbf{x}'_i \leftrightarrow \mathbf{x}''_i\}$  bestimmt werden. Die Schnittmenge zwischen  $\{\mathbf{B}_{b,b>2} \leftrightarrow \mathbf{B}_{b+1,b>2}\}$  und  $\{\mathbf{B}_{b+1,b>2} \leftrightarrow \mathbf{B}_{b+2,b>2}\}$  ergibt die potentiell fehlerfreien Merkmalskorrespondenzen. Nur diese sind in den nachfolgenden Filtermaßnahmen und in der anschließenden Triangulation zu verwenden.

Die nachfolgende Abbildung 5.39 veranschaulicht die Berechnung der Schnittmenge und die daraus resultierende Verifikation der Merkmalskorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ .



**Abbildung 5.39:** Verifikation von Merkmalskorrespondenzen.

Es sei darauf hingewiesen, dass dieses Unterkapitel die Korrespondenzfindung nicht im Detail wiedergibt. Im Kapitel 5.3.1 ab Seite 43 ist die Suche nach Merkmalskorrespondenzen ausführlich beschrieben worden und gegebenenfalls zu entnehmen.

## 5.4.2 Filterung

Eine Filterung nicht verwendbarer Korrespondenzen ist wie bei der initialen Bestimmung von Weltpunkten durchzuführen. Die Filterungsmaßnahmen wurden im Kapitel 5.3.2 beschrieben. Dabei findet eine Merkmalsfilterung auf Duplikate (Siehe S. 47), falschen Korrespondenzen (S. 48) und einer degenerierten Korrespondenzmenge (S. 48) statt.

## 5.4.3 Berechnung der 3D-Weltpunkte

Dieses Unterkapitel ist in die vier Phasen

1. Bestimmung der Projektionsmatrizen
2. Triangulation und Bündelblockausgleich
3. Filterung anhand Rückprojektion
4. Korrektur der neuen Weltpunkte

unterteilt.

### 5.4.3.1 Bestimmung der Projektionsmatrizen

Wie im Kapitel 5.3.3.1 beschrieben erfolgt anhand der Merkmalskorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ , der Fundamentalmatrix  $\mathbf{F}$  und den intrinsischen Kameraparametern  $\mathbf{K}$  die Ermittlung der Projektionsmatrix  $\mathbf{P}_p$  und  $\mathbf{P}_{p+1}$  der beiden Aufnahmen  $\mathbf{B}_b$  und  $\mathbf{B}_{b+1}$ .

### 5.4.3.2 Triangulation und Bündelblockausgleich

Das Kapitel 5.3.3.2 erläutert anhand von zwei Projektionsmatrizen  $\mathbf{P}_p$  und  $\mathbf{P}_{p+1}$  die Triangulation sowie den Bündelblockausgleich. Damit jeweils der dreidimensionale Ursprung für die Merkmalskorrespondenzen  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$  ermittelt werden kann, ist das gleiche Vorgehen an dieser Stelle notwendig.

### 5.4.3.3 Filterung anhand Rückprojektion

Anschließend sind alle triangulierten Weltpunkte, welche eine schlechte Rückprojektion aufweisen, zu entfernen. Die Filterung wurde ebenfalls bei der initialen Bestimmung von Weltpunkten beschrieben (Vgl. Seite 57).

#### 5.4.3.4 Korrektur der neuen Weltpunkte

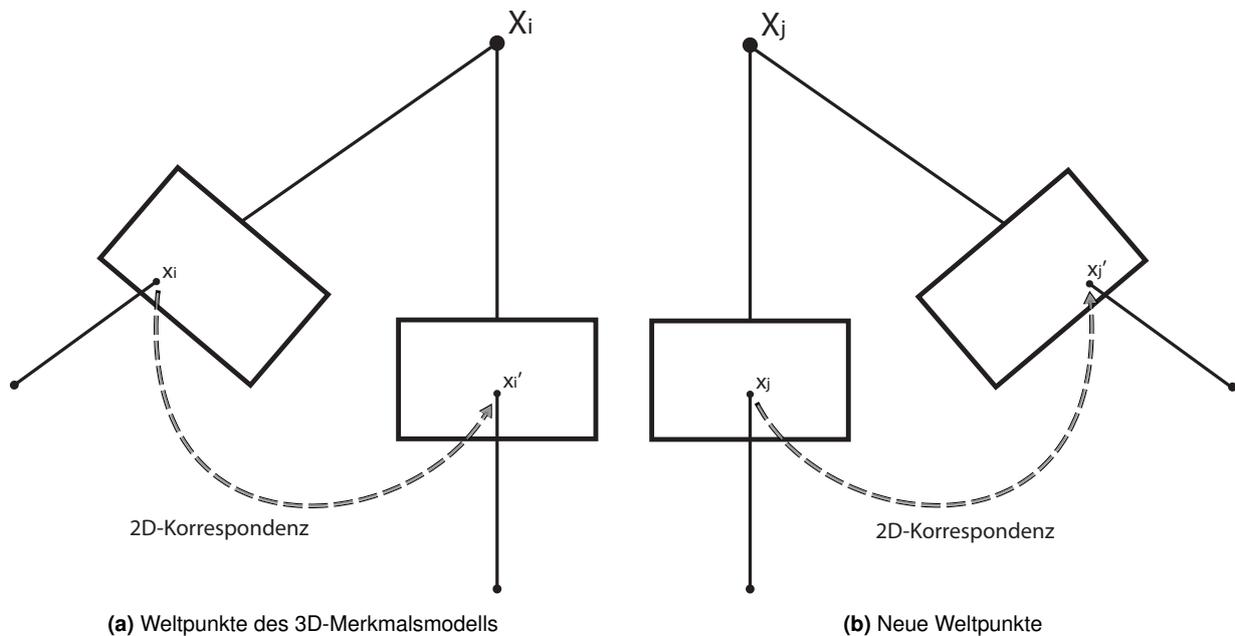
An dieser Stelle wird der Unterschied zur initialen Bestimmung von Weltpunkten deutlich. Die Triangulation ist bis auf einen Skalierungsfaktor genau. Aus diesem Grund mussten die initialen Weltpunkte richtig skaliert werden. Im Kapitel 5.3.3.4 wurde für die ersten Weltpunkte die korrekte Skalierung ermittelt.

In diesem Kapitel 5.4 (*Weltpunkte aus weiteren Aufnahmen*) stehen die Projektionsmatrizen  $\mathbf{P}_{p,p>2}$  in keiner Relation zu den ersten beiden  $\mathbf{P}_{p \in \{1,2\}}$ . Aus diesem Grund ist die Kamerabewegung und -rotation nicht in den Projektionsmatrizen  $\mathbf{P}_{p,p>2}$  enthalten, sodass die neu triangulierten Weltpunkte  $\mathbf{X}_j$  korrigiert werden müssen.

Genaugenommen müssen nur die neuen Weltpunkte korrigiert werden, welche noch nicht im 3D-Merkmalmodell vorhanden sind. Bei den schon vorhandenen Weltpunkten sollten jedoch die neuen Deskriptoren (Merkmale) mit in das 3D-Merkmalmodell hinzugefügt werden. Aus diesem Grund wird die 3D-Korrespondenzmenge

$$\{\mathbf{X}_i \in \text{3D-Merkmalmodell}\} \longleftrightarrow \mathbf{X}_j \quad (5.67)$$

zwischen dem bestehenden 3D-Merkmalmodell und den aktuell ermittelten Weltpunkten  $\mathbf{X}_j$  ermittelt (Abbildung 5.40).



**Abbildung 5.40:** Korrektur von neuen Weltpunkten  $\mathbf{X}_j$  relativ zu den bestehenden Weltpunkten  $\mathbf{X}_i$  des 3D-Merkmalmodells.

Anschließend ist die Deskriptormenge der Weltpunkte  $\mathbf{X}_i$  mit den Deskriptoren der Korrespondenzen  $\mathbf{X}_j$  zu erweitern. Danach wird die Transformationsmatrix  $\mathbf{T}$ , welche die Abbildung

$$\mathbf{X}_j \mapsto \{\mathbf{X}_i \in \text{3D-Merkmalmodell}\} \quad (5.68)$$

beschreibt, berechnet. Die Transformationsmatrix wird für die Weltpunkte

$$\mathbf{X}_n = \forall \mathbf{X}_j \notin \{\mathbf{X}_i \leftrightarrow \mathbf{X}_j\} \quad (5.69)$$

, die sich noch nicht im 3D-Merkmalmodell befinden, benötigt. Diese werden mit der Transformationsmatrix  $\mathbf{T}_{3 \times 4} = [\mathbf{R}_{3 \times 3} \mid \mathbf{t}_{3 \times 1}]$  in Relation zu den ersten Kameraaufnahmen gebracht und damit auch richtig skaliert

(Gleichung 5.70). Denn die Weltpunkte und Projektionsmatrizen, welche sich derzeit im 3D-Merkmalmodell befinden, wurden im Kapitel 5.3.3.4 richtig skaliert. Diese korrekte Skalierung wird durch  $\mathbf{T}$  vererbt.

$$\hat{\mathbf{X}}_n = \mathbf{T} \cdot \mathbf{X}_n \quad (5.70)$$

Die Berechnung von  $\mathbf{T}$  wird im Kapitel 6 ab Seite 86 ausführlich beschrieben. Des Weiteren befindet sich im Anhand A.4 auf der Seite 131 der Programmiercode in Kurzform.

Die nachfolgende Abbildung 5.41 veranschaulicht die räumliche Anpassung der neuen Weltpunkte. Die linke Grafik 5.41a stellt die Korrespondenzmenge  $\mathbf{X}_i^{\text{modell}} \leftrightarrow \mathbf{X}_j^{\text{neu}}$  vor der Punkt Korrektur grafisch dar. Dabei sind die korrespondierenden Weltpunkte des 3D-Merkmalmodells mit schwarz ausgefüllten Kreisen gekennzeichnet. Dagegen besitzen die Korrespondenzen der neuen Weltpunktmenge ein nicht ausgefülltes Quadrat. Jeweils ein nicht ausgefüllter Kreis symbolisiert einen neuen Weltpunkt, welcher keine Korrespondenz zum 3D-Merkmalmodell besitzt.

Des Weiteren sind Korrespondenzen mit einer grünen oder roten Linie verbunden. In dem Fall, dass der Abstand  $|\mathbf{X}_j^{\text{neu}} - \mathbf{X}_i^{\text{modell}}|$  über einen Schwellwert  $v$  steigt, wird dieser rot-, andernfalls grün dargestellt.

Der Schwellwert

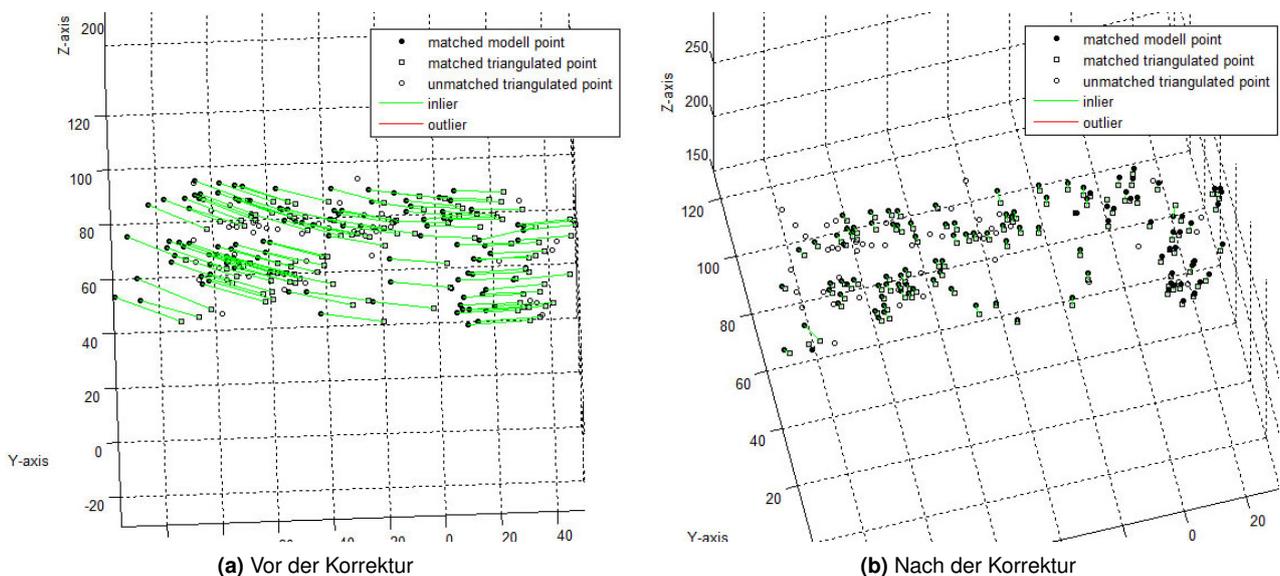
$$v = \alpha \cdot \bar{d}_{\text{arith}} = \alpha \cdot \frac{1}{n} \cdot \sum_{i=1}^n |\mathbf{X}_j^{\text{neu}} - \mathbf{X}_i^{\text{modell}}| \quad (5.71)$$

wird durch den arithmetischen Mittelwert aller Abstände und einem Verstärkungsfaktor  $\alpha$  berechnet. Dies ermöglicht eine robustere Berechnung der Transformationsmatrix  $\mathbf{T}$ , indem  $\mathbf{T}$  lediglich mit den Korrespondenzen ermittelt wird, welche den Schwellwert nicht übersteigen.

Weiterführend sind die  $n$  neuen Weltpunkte

$$\mathbf{X}_n^{\text{neu}} \notin \{\mathbf{X}_i^{\text{modell}} \leftrightarrow \mathbf{X}_j^{\text{neu}}\} \quad (5.72)$$

, welche nicht in der Korrespondenzmenge enthalten sind, mit der berechneten  $3 \times 4$  Transformationsmatrix  $\mathbf{T}$  zu korrigieren. Die rechte Grafik 5.41b der nachfolgenden Abbildung stellt alle Weltpunkte nach der räumlichen Korrektur dar.



**Abbildung 5.41:** Räumliche Korrektur von neuen Weltpunkten.

#### 5.4.4 Erweiterung des 3D-Merkmalmodells

In dieser letzten Phase sind die aus den Filtermaßnahmen übrig gebliebenen Weltpunkte in das zugehörige 3D-Merkmalmodell hinzuzufügen. Dabei werden zugehörige Informationen wie u. a. die Deskriptoren (Merkmale) jedem Weltpunkt zugeordnet.

Die Datenstruktur der Modelldatenbank wurde im Kapitel 5.3.4 (*Anlegen eines 3D-Merkmalmodells*) in der Abbildung 5.35 dargestellt.

# 6 Posenbestimmung

## Inhalt

---

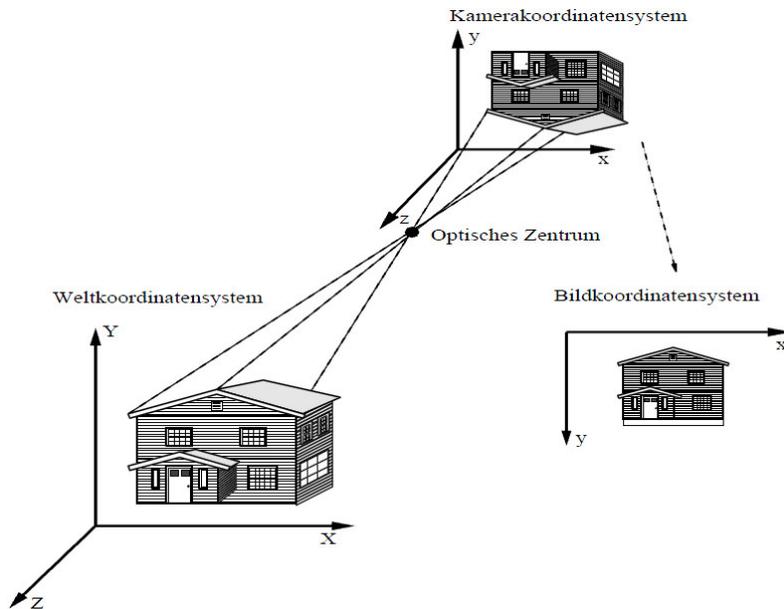
6.1	3D-RANSAC . . . . .	71
6.2	Normalisierte Direct Linear Transformation . . . . .	72
6.3	Verbessertes nicht-iteratives Verfahren . . . . .	75

---

Die Suche nach der relativen Pose anhand von Punktkorrespondenzen wird als sogenanntes *Perspective n-Point* (PnP)-Problem bezeichnet. Die relative Pose zwischen zwei Koordinatensystemen wird mit einer 3 x 3 Rotationsmatrix  $\mathbf{R}$  und einem 3 x 1 Translationsvektor  $\mathbf{t}$  beschrieben. Die zusammengefasste Rotation und Translation in einer gemeinsamen Transformationsmatrix  $\mathbf{T}$  zeigt die Gleichung 6.1.

$$\mathbf{T} = [\mathbf{R} \mid \mathbf{t}] \tag{6.1}$$

Die Grafik 6.1 veranschaulicht die Abbildung zwischen einer Szene (Welt- bzw. Objektkoordinatensystem) und dem Kamera- oder Bildkoordinatensystem. Die zu bestimmende Pose erfolgt auf Basis von 3D-Welt- zu 2D-Bild-Punktkorrespondenzen  $\mathbf{X} \leftrightarrow \mathbf{x}$  und den intrinsischen Kameraparametern  $\mathbf{K}$ .



**Abbildung 6.1:** Zusammenhang des Welt-, Bild- und Kamerakoordinatensystems. [KS08]

Weiterführend ist die Abbildung von Welt-  $\mathbf{X} = [X, Y, Z]^T$  auf Bildkoordinaten  $\mathbf{x} = [x, y]^T$  mit der nachfolgenden Gleichung 6.2 möglich.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K} \cdot \mathbf{T} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \underbrace{\mathbf{K} \cdot [\mathbf{R} \mid \mathbf{t}]}_{\mathbf{P}} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{6.2}$$

In dieser entspricht die Multiplikation der Kalibriermatrix  $\mathbf{K}$  mit  $\mathbf{T}$  der singulären Projektionsmatrix  $\mathbf{P}$ .

Die Wahrscheinlichkeit, dass in einer 3D-zu-2D-Korrespondenzmenge inkorrekte bzw. unpräzise Korrespondenzen vorkommen ist recht hoch. Aus diesem Grund sollten diese Fehlzuordnungen aus der Korrespondenzmenge herausgefiltert werden, damit die Bestimmung der Pose nicht verfälscht wird. Das Kapitel 6.1 stellt eine Möglichkeit, zur Erkennung und Eliminierung, von fehlerbehafteten 3D-zu-2D-Punktkorrespondenzen vor.

Nachdem eine einwandfreie Korrespondenzmenge vorhanden ist, kann eine *optimale* Transformationsmatrix  $\mathbf{T}$  bestimmt werden. Dafür beschreibt das Unterkapitel 6.2 und 6.3 jeweils ein nicht-iteratives Verfahren. Zur Vollständigkeit halber wird an dieser Stelle auf ein iteratives Verfahren mit der Bezeichnung *Pose from Orthography and Scaling with Iterations* (kurz POSIT) verwiesen. Informationen sind aus [DDDS02] und [ODD96] zu entnehmen.

## 6.1 3D-RANSAC

Im Grundlagenkapitel 3.3.3 (*Robuste Schätzung der Fundamentalmatrix*) wurde die Berechnung der Fundamentalmatrix, mit einer Filterung auf fehlerfreie Merkmalskorrespondenzen, beschrieben. Dafür kam das RANSAC-Verfahren zum Einsatz, welches im Algorithmus 1 auf der Seite 17 vorgestellt wurde.

In dieser Arbeit wurde der RANSAC-Algorithmus für eine 3D-Welt- zu 2D-Bild-Korrespondenzmenge  $\mathbf{X} \leftrightarrow \mathbf{x}$  weiterentwickelt, sodass fehlerhafte Korrespondenzen erkannt werden. Dies ermöglicht eine optimale Berechnung der Projektionsmatrix  $\mathbf{P}$

$$\mathbf{x} = \mathbf{P} \cdot \mathbf{X} = [\mathbf{R}_p \mid \mathbf{t}_p] \cdot \mathbf{X} = \begin{bmatrix} R_{p11} & \cdots & R_{p13} & t_{p1} \\ \vdots & \ddots & \vdots & t_{p2} \\ R_{p31} & \cdots & R_{p33} & t_{p3} \end{bmatrix} \cdot \mathbf{X} \quad (6.3)$$

bzw. der Transformationsmatrix  $\mathbf{T}$

$$\mathbf{x} = \underbrace{\mathbf{K} \cdot \mathbf{T}}_{\mathbf{P}} \cdot \mathbf{X} \quad (6.4)$$

Das erweiterte RANSAC-Verfahren wird im Folgenden beschrieben.

1. Modellschätzung:

Wähle per Zufall aus der Eingabemenge  $k$  Korrespondenzen aus und führe eine Schätzung des Modells  $M$  (Fundamental-, Projektionsmatrix, o. a.) durch.

2. Inlier-Auswertung:

Ermittle, in Abhängigkeit eines Abstandsfaktors, die Anzahl an Inlier aus der Menge  $\mathbf{x} \leftrightarrow \mathbf{x}'$  bzw.  $\mathbf{X} \leftrightarrow \mathbf{x}$ .

a) *Terminiere*,

falls die Inlieranzahl einen festgelegten Schwellwert erreicht hat und berechne anhand aller Inlier das optimale Modell  $M_{\text{opt}}$ .

b) *Abbruch*

bei dem Erreichen einer maximalen Iterationsanzahl und wähle anschließend den größten- von allen bisher ermittelten Datensätzen aus. Anhand des gewählten Datensatzes (Korrespondenzen der Iteration mit der größten Inlieranzahl) ist das Modell  $M_{\text{opt}}$  zu berechnen.

c) *Wiederhole*

die Modellschätzung, wenn der Schwellwert unterschritten und die maximale Iterationsanzahl nicht erreicht wurde.

Im Kapitel 3.3.3 wurde beschrieben, wie die Fundamentalmatrix  $\mathbf{F}$  mit acht 2D-Merkmalsskorrespondenzen  $\mathbf{x} \leftrightarrow \mathbf{x}'$  zu berechnen ist. Anstelle dessen werden sechs 3D-zu-2D-Korrespondenzen  $\mathbf{X} \leftrightarrow \mathbf{x}$  verwendet, um die Transformationsmatrix  $\mathbf{T}$  zu berechnen. Dabei beschreibt diese Transformationsmatrix die Rotation und Translation zwischen dem Ursprung des Objekt- und Kamerakoordinatensystems.

Im Rahmen der Schätzung von  $\mathbf{T}$  wurden die nachfolgenden RANSAC-Einstellungen gewählt. Die maximale Anzahl an Iterationen  $N$  wurde auf 1000 Durchgänge gesetzt. Die Wahrscheinlichkeit, dass mindestens eine Korrespondenzmenge von sechs fehlerfreien 3D-2D-Korrespondenzen zufällig gewählt wird, bekam den Wert 0,99. Dieser Wert entspricht einer 99 prozentigen Wahrscheinlichkeit einer Korrespondenz-Untermenge, welche *ideale* und damit fehlerfreie Korrespondenzen beinhaltet. Im Gegensatz zur Berechnung der Fundamentalmatrix haben statistische Auswertungen<sup>22</sup> gezeigt, dass die Mindestanzahl an Iterationen nicht auf eins sondern auf einen recht großen Wert gesetzt werden sollte. Das liegt an dem Grund, dass 3D-2D-Korrespondenzen  $\mathbf{X} \leftrightarrow \mathbf{x}$  fehlerbehafteter sind als 2D-Korrespondenzen  $\mathbf{x} \leftrightarrow \mathbf{x}'$ . Aus diesem Grund wurden mindestens 20 Iterationen durchlaufen, um letztendlich die beste Korrespondenz-Untermenge für die Berechnung der *optimalen* Transformationsmatrix zu verwenden.

## 6.2 Normalisierte Direct Linear Transformation

Dieses Unterkapitel beschreibt die Berechnung der Projektionsmatrix  $\mathbf{P}$  aus 3D-zu-2D-Punktkorrespondenzen anhand der normalisierten *Direct Linear Transformation* (DLT). Des Weiteren wird beschrieben, wie aus der Projektionsmatrix die Transformationsmatrix  $\mathbf{T}$ <sup>23</sup> hergeleitet wird.

Der ursprüngliche DLT-Algorithmus wurde von [AAK71] und [MK75] entwickelt. Auf dieser Basis wurde in [HZ04] der normalisierte DLT-Algorithmus veröffentlicht. Dieser berechnet die relative Pose, zwischen dem Bild- und Weltkoordinatensystem, und damit die Projektionsmatrix  $\mathbf{P}$  schneller als iterative Minimierungsverfahren. Des Weiteren wird eine initiale Schätzung der Startwerte nicht benötigt. Die Berechnung der Transformationsparameter von  $\mathbf{P}$  erfolgt *direkt* in einer geschlossenen Form, wodurch keine Konvergenzprobleme auftreten. Die problematische Wahl eines günstigen Abbruchkriteriums ist nicht erforderlich. Es gibt allerdings auch einen großen Nachteil beim DLT-Verfahren. Die zwölf unbekanntenen Projektionsparameter

$$\mathbf{P} = \left[ \begin{array}{ccc|c} R_{p11} & \cdots & R_{p13} & t_{p1} \\ \vdots & \ddots & \vdots & t_{p2} \\ R_{p31} & \cdots & R_{p33} & t_{p3} \end{array} \right] \quad (6.5)$$

werden unabhängig voneinander geschätzt, was zu sehr großen Ungenauigkeiten führen kann. Für die Berechnung der unbekanntenen Projektionsparameter sind mindestens sechs Punktkorrespondenzen von Nöten.

Im Nachfolgenden ist der in dieser Arbeit verwendete normalisierte DLT-Algorithmus von [TV98] und [HZ04] zusammen mit einer Erweiterung der Projektionsmatrix zur Transformationsmatrix  $\mathbf{T}$ , beschrieben.

Für die Bestimmung der zwölf unbekanntenen Projektionsparameter müssen mindestens sechs Korrespondenzen  $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$  zur Verfügung stehen.

Zu Beginn sind alle Bildkoordinaten  $\mathbf{x}$  aus der 3D-2D-Korrespondenzmenge  $\mathbf{X} \leftrightarrow \mathbf{x}$  mit der Kamera-Kalibriermatrix  $\mathbf{K}$  zu korrigieren. Die Kalibriermatrix beinhaltet die intrinsischen Kameraparameter, wie die Hauptpunktverschiebung  $\mathbf{h}_{x,y}$ , die Bildweite  $\mathbf{f}_{x,y}$ , den Scherungsfaktor  $s$  und die Linsenverzeichnung  $\mathbf{k}_{1,x5}$ . Dafür ist die  $x$ -Komponente der Hauptpunktverschiebung von der  $x$ -Komponente des Bildpunktes  $\mathbf{x}_i \in (\mathbf{X} \leftrightarrow \mathbf{x})$  zu subtrahieren. Das Ergebnis wird anschließend durch die  $x$ -Komponente der Bildweite dividiert. Der analoge Fall gilt für die  $y$ -Komponente. Die verwendeten Kameras dieser Arbeit besitzen einen Scherungsfaktor von null. Dieser Wert repräsentiert rechteckige Pixel. Aus diesem Grund wird der Scherungsfaktor nicht weiter betrachtet. Abschließend sind bei einer vorhandenen Linsenverzeichnung die Bildkoordinaten zu entzerren.

<sup>22</sup>Bei der Evaluierung der Posenbestimmung wurden sowohl iterative- als auch nicht-iterative Verfahren herangezogen.

<sup>23</sup>Beschreibt die Abbildung zwischen Welt- und Kamerakoordinaten bzw. Objekt- und Kamerakoordinaten.

Eine ausführliche Beschreibung der intrinsischen Kameraparameter und der Bildpunktentzerrung wurde im Kapitel 5.1 (*Kamerakalibrierung*) vorgenommen.

Im zweiten Schritt sollten, aufgrund der hohen Koordinatenschwankungen in digitalen Bildern, die  $n$  Bildkoordinaten  $\mathbf{x}_i$  sowie die zugehörigen Weltkoordinaten  $\mathbf{X}_i$  normalisiert werden. Denn diese auftretenden Koordinatenschwankungen im Bereich von 1 bis über 1000 führen häufigerweise zu numerischen Schwierigkeiten (Vgl. [HZ04] ab S. 107).

Die Normalisierung beginnt mit einer punktweisen Schwerpunktverschiebung der Bild-

$$x_{i_x} = x_{i_x} - \bar{x}_{\text{centroid}_x} \quad (6.6)$$

$$x_{i_y} = x_{i_y} - \bar{x}_{\text{centroid}_y} \quad (6.7)$$

und Weltkoordinaten

$$X_{i_x} = X_{i_x} - \bar{X}_{\text{centroid}_x} \quad (6.8)$$

$$X_{i_y} = X_{i_y} - \bar{X}_{\text{centroid}_y} \quad (6.9)$$

$$X_{i_z} = X_{i_z} - \bar{X}_{\text{centroid}_z} \quad (6.10)$$

Der Schwerpunkt der Bildkoordinaten wird durch  $\bar{\mathbf{x}}_{\text{centroid}} = \frac{1}{n} \cdot \sum \mathbf{x}_i$  bestimmt. Der Weltkoordinatenschwerpunkt ist mit  $\bar{\mathbf{X}}_{\text{centroid}} = \frac{1}{n} \cdot \sum \mathbf{X}_i$  zu berechnen. Anschließend ist das Skalieren vom mittleren Abstand der Punkte zum Schwerpunkt auf  $\sqrt{2}$  bzw.  $\sqrt{3}$  durchzuführen. Dafür müssen die punktweisen Abstände mit

$$x_{\text{distances}_i} = \sqrt{x_{i_x}^2 + x_{i_y}^2} \quad (6.11)$$

$$X_{\text{distances}_i} = \sqrt{X_{i_x}^2 + X_{i_y}^2 + X_{i_z}^2} \quad (6.12)$$

berechnet werden. Danach erfolgt die Ermittlung des mittleren Abstandes mit Hilfe des arithmetischen Mittelwertes. Sei  $n$  die Anzahl der Punktkorrespondenzen, dann führen die Gleichungen 6.13 und 6.14 zum aktuellen Abstandsmittelwert.

$$\bar{x}_{\text{arith}} = \frac{1}{n} \cdot \sum_{i=1}^n x_{\text{distances}_i} \quad (6.13)$$

$$\bar{X}_{\text{arith}} = \frac{1}{n} \cdot \sum_{i=1}^n X_{\text{distances}_i} \quad (6.14)$$

Im letzten Schritt werden die Skalierungskoeffizienten  $[x_{\lambda_x}, x_{\lambda_y}]^T$  und  $[X_{\lambda_x}, X_{\lambda_y}, X_{\lambda_z}]^T$  so berechnet, dass der mittlere Abstand der Punkte zum Schwerpunkt  $\sqrt{2}$  bzw.  $\sqrt{3}$  beträgt. Die nachfolgenden Gleichungen 6.15 und 6.16 geben diese Berechnung wider. Dabei gilt der Abstandswert  $\sqrt{2}$  für die 2D-Bildkoordinaten und die Distanz  $\sqrt{3}$  für die 3D-Weltkoordinaten.

$$\mathbf{x}_\lambda = \frac{\sqrt{2}}{\bar{x}_{\text{arith}}} \quad (6.15)$$

$$\mathbf{X}_\lambda = \frac{\sqrt{3}}{\bar{X}_{\text{arith}}} \quad (6.16)$$

Abschließend erfolgt auf Basis der Skalierungskoeffizienten die Skalierung der Punktkorrespondenzen, sodass die Bild- und Weltkoordinaten normalisiert vorliegen:

$$\hat{\mathbf{x}}_i = \mathbf{x}_\lambda \cdot \mathbf{x}_i \quad (6.17)$$

$$\hat{\mathbf{X}}_i = \mathbf{X}_\lambda \cdot \mathbf{X}_i \quad (6.18)$$

Insgesamt kann die Schwerpunktverschiebung zusammen mit der Skalierung in einer metrischen Transformationsmatrix  $\mathbf{T}_b$  bzw.  $\mathbf{T}_w$  zusammengefasst werden. Nachfolgend ist diese für die Bildkoordinaten

$$\mathbf{T}_b = \begin{bmatrix} x_{\lambda_x} & 0 & -x_{\lambda_x} \cdot \bar{x}_{\text{centroid}_x} \\ 0 & x_{\lambda_y} & -x_{\lambda_y} \cdot \bar{x}_{\text{centroid}_y} \\ 0 & 0 & 1 \end{bmatrix} \quad (6.19)$$

und für die Weltkoordinaten

$$\mathbf{T}_w = \begin{bmatrix} X_{\lambda_x} & 0 & 0 & -X_{\lambda_x} \cdot \bar{X}_{\text{centroid}_x} \\ 0 & X_{\lambda_y} & 0 & -X_{\lambda_y} \cdot \bar{X}_{\text{centroid}_x} \\ 0 & 0 & X_{\lambda_z} & -X_{\lambda_z} \cdot \bar{X}_{\text{centroid}_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.20)$$

dargestellt. Diese sind für die Normalisierung der Weltkoordinaten

$$\hat{\mathbf{X}}_i \sim \mathbf{T}_w \cdot \mathbf{X}_i \quad (6.21)$$

als auch zur Normalisierung der Bildkoordinaten

$$\hat{\mathbf{x}}_i \sim \mathbf{T}_b \cdot \mathbf{x}_i \quad (6.22)$$

zu verwenden.

Nachdem die gesamte Punktmenge normalisiert wurde, kann die gesuchte Projektionsmatrix  $\mathbf{P}$  auf Basis der Gleichung 6.23 berechnet werden.

$$\mathbf{A} = \begin{bmatrix} X_{1_x} & X_{1_y} & X_{1_z} & 1 & 0 & 0 & 0 & 0 & -x_{1_x}X_{1_x} & -x_{1_x}X_{1_y} & -x_{1_x}X_{1_z} & -x_{1_x} \\ \vdots & \vdots \\ X_{n_x} & X_{n_y} & X_{n_z} & 1 & 0 & 0 & 0 & 0 & -x_{n_x}X_{n_x} & -x_{n_x}X_{n_y} & -x_{n_x}X_{n_z} & -x_{n_x} \\ 0 & 0 & 0 & 0 & X_{1_x} & X_{1_y} & X_{1_z} & 1 & -x_{1_y}X_{1_x} & -x_{1_y}X_{1_y} & -x_{1_y}X_{1_z} & -x_{1_y} \\ \vdots & \vdots \\ 0 & 0 & 0 & 0 & X_{n_x} & X_{n_y} & X_{n_z} & 1 & -x_{n_y}X_{n_x} & -x_{n_y}X_{n_y} & -x_{n_y}X_{n_z} & -x_{n_y} \end{bmatrix} \quad (6.23)$$

Dafür wird die Singulärwertzerlegung auf die konstruierte Matrix  $\mathbf{A}$  angewendet. Die Folge einer Singulärwertzerlegung sind die drei Matrizen  $\mathbf{U}$ ,  $\mathbf{S}$  und  $\mathbf{V}$  aus der Gleichung 6.24. Die Singulärwertzerlegung sowie die Eigenschaften der drei resultierenden Matrizen wurden im Kapitel 3.2 auf Seite 13 erläutert.

$$\text{svd}(\mathbf{A}) = [\mathbf{U}, \mathbf{S}, \mathbf{V}] \quad (6.24)$$

Die  $2n \times 12$  Elemente der  $\mathbf{V}$ -Matrix bestimmen die normalisierte  $3 \times 4$  Projektionsmatrix  $\hat{\mathbf{P}}$ . Im Anschluss ist  $\hat{\mathbf{P}}$  zu denormalisieren (Gleichung 6.25).

$$\mathbf{P} = \text{inv}(\mathbf{T}_b) \cdot \hat{\mathbf{P}} \cdot \mathbf{T}_w \quad (6.25)$$

Bei der Denormalisierung muss die Transformationsmatrix der Bildkoordinaten zuerst invertiert werden. Nach der Invertierung von  $\mathbf{T}_b$  ist diese mit der normalisierten Projektionsmatrix  $\hat{\mathbf{P}}$  sowie der Transformationsmatrix der 3D-Weltkoordinaten zu multiplizieren. Das Resultat ist die Projektionsmatrix  $\mathbf{P}$ , welche die Abbildung der Weltkoordinaten  $\mathbf{X}_i$  auf die Bildkoordinaten  $\mathbf{x}_i$  beschreibt.

An dieser Stelle erfolgt die Weiterentwicklung der Projektionsmatrix  $\mathbf{P}$  zur Transformationsmatrix  $\mathbf{T} = [\mathbf{R} \mid \mathbf{t}]$ , welche die Abbildung von Welt- bzw. Objekt- auf Kamerakoordinaten beschreibt. Dafür muss die Projektionsmatrix in Ihre Bestandteile, die intrinsischen Kameraparameter  $\mathbf{K}$  und den Rotations- und Translationsanteil  $(\mathbf{R}, \mathbf{t})$ , zerlegt werden. Dabei ist der Rotations- und Translationsanteil der Transformationsmatrix  $\mathbf{T}$  zugehörig.

Für die angesprochene Zerlegung kann einerseits die QR-Zerlegung<sup>24</sup> herangezogen werden. Andererseits ist die Faktorisierung von  $\mathbf{P}$  auch in einer geschlossenen Form möglich. Beide Herangehensweisen werden ausführlich in [KIM06] beschrieben und sind gegebenenfalls dort zu entnehmen.

Aus der Zerlegung von  $\mathbf{P}_{3 \times 4}$  resultiert die Transformationsmatrix  $\mathbf{T}_{3 \times 4} = [\mathbf{R}_{3 \times 3} \mid \mathbf{t}_{3 \times 1}]$  und die intrinsischen Kameraparameter  $\mathbf{K}_{3 \times 3}$ . Die Transformationsmatrix beschreibt die Abbildung vom Welt- zum Kamerakoordinatensystem. In dem Fall, dass die umgekehrte Abbildung gesucht wird, ist eine Invertierung notwendig. Die Invertierung erfolgt indem der Rotationsanteil  $\mathbf{R}$  transponiert und der Translationsanteil  $\mathbf{t}$  mit dem negativen transponierten Rotationsanteil multipliziert wird. Die Gleichungen 6.26 bis 6.28 beschreiben die Invertierung der Transformationsmatrix.

$$\mathbf{R}_{\text{inv}} = \mathbf{T}'_{(1:3,1:3)} \quad (6.26)$$

$$\mathbf{t}_{\text{inv}} = -\mathbf{R}_{\text{inv}} \cdot \mathbf{T}_{(1:3,4)} \quad (6.27)$$

$$\mathbf{T}_{\text{Kamera} \rightarrow \text{Welt}} = \text{inv}(\mathbf{T}) = [\mathbf{R}_{\text{inv}} \mid \mathbf{t}_{\text{inv}}] \quad (6.28)$$

### 6.3 Verbessertes nicht-iteratives Verfahren

Dieses Unterkapitel beschreibt einen zweiten nicht-iterativen Ansatz für die Bestimmung der Transformationsmatrix  $\mathbf{T}$  mit der Abbildung Welt-  $\mapsto$  Kamerakoordinaten und der Berechnung der Projektionsmatrix  $\mathbf{P}$  mit der Abbildung Welt-  $\mapsto$  Bildkoordinaten.

Typischerweise weisen nicht-iterative- gegenüber iterativen Verfahren eine geringere Genauigkeit auf, jedoch besitzen sie einen deutlich größeren Geschwindigkeitsvorteil (Vgl. Abbildung 6.2).

Im Nachfolgenden wird ein präzises nicht-iteratives Verfahren beschrieben, welches in einer Laufzeit von  $O(n)^{25}$  eine Lösung des *Perspective n-Point-Problems* berechnet. Das Verfahren wird als *Efficient PnP* (kurz EPnP) bezeichnet, basiert auf der in [MNL07] veröffentlichten Implementierung und benötigt mindestens vier 3D-zu-2D Punktkorrespondenzen sowie bekannte intrinsische Kameraparameter. Des Weiteren können die Punktmengen sowohl planar als auch nicht-planar vorliegen.

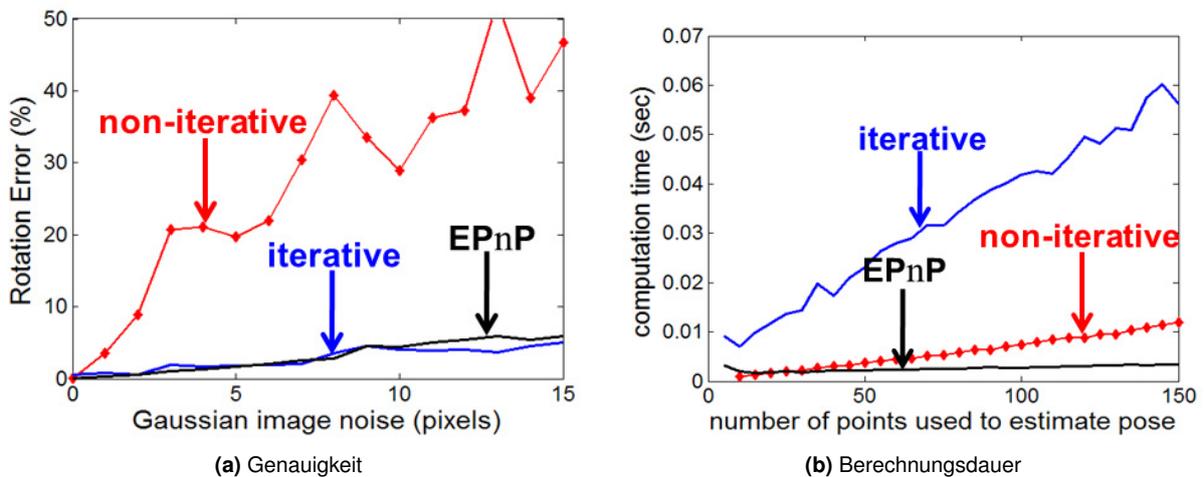
Bei EPnP und den typischen nicht-iterativen Verfahren, zur Bestimmung einer Transformationsmatrix  $\mathbf{T}$ , welche die Abbildung zwischen der Korrespondenzmenge  $\mathbf{X}_i \longleftrightarrow \mathbf{x}_i^{\text{camera}}$  beschreibt, werden zuerst die Koordinaten im Kamerakoordinatensystem berechnet. Denn sind diese Koordinaten bekannt, ist die Bestimmung der Rotation und Translation einer Transformation  $\mathbf{T}^{26}$  einfach. Zu Beginn versuchen die gängigen Algorithmen die Kamerakoordinaten zu ermitteln, indem sie die Tiefe der Punkte bestimmen. Die Grundidee des EPnP-Verfahrens liegt dagegen in der Verwendung von sogenannten *virtuellen* Kontrollpunkten. Ein Kontrollpunkt im Weltkoordinatensystem ist ein 3D-Punkt der zusammen mit zwei (Bei keiner planaren Eingabemenge) oder drei (Planare

<sup>24</sup>Faktorisierung einer Matrix  $\mathbf{M}$  in eine orthogonale Matrix  $\mathbf{Q}$  und eine obere Dreiecksmatrix  $\mathbf{R}$ . Dabei resultiert aus dem Produkt von  $\mathbf{Q}$  und  $\mathbf{R}$  die ursprüngliche Matrix  $\mathbf{M}$ .

<sup>25</sup>Die O-Notation beschreibt den Zeitaufwand einer Berechnung bezüglich eines Algorithmus und kennzeichnet den sogenannten *worst case*. Nachfolgend findet zur Veranschaulichung ein Wachstumsvergleich statt:  $1 < \log n < \sqrt{n} < n < n \cdot (\log n)^2 < n^2 < n^a < a^n$

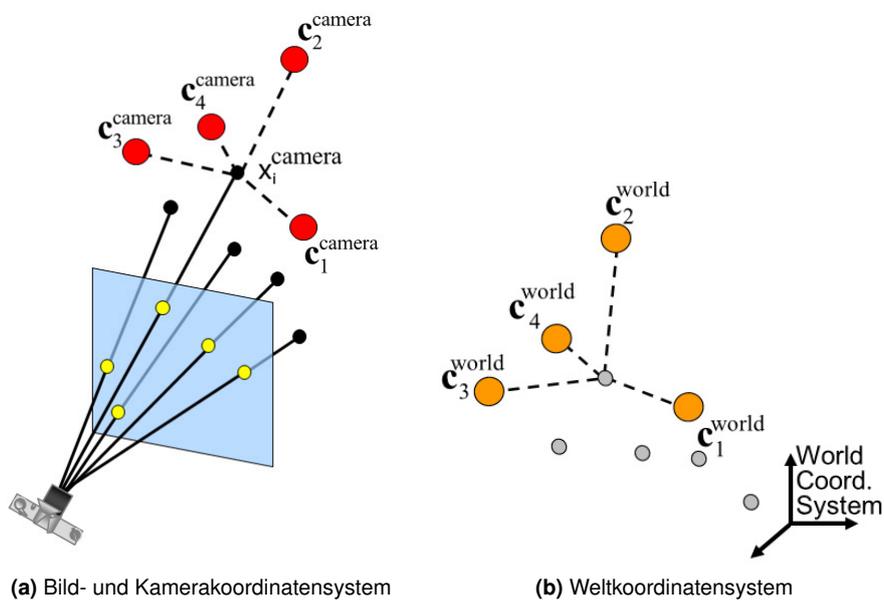
<sup>26</sup>Es handelt sich dabei um eine affine Transformation, dessen linearer Anteil eine orthogonale Transformation ist.

Eingabemenge) anderen Kontrollpunkten eine Beziehung zu allen Weltpunkten  $\mathbf{X}_i$  beschreibt. Für die korrespondierenden Kamerakoordinaten  $\mathbf{x}_i^{\text{camera}}$  gilt dies analog.



**Abbildung 6.2:** Geschwindigkeits- und Genauigkeitsvergleich zwischen iterativen- (Implementierung von [LHM00]), nicht-iterativen- (DLT mit Einbeziehung der intrinsischen Kameraparameter) und dem EPnP-Verfahren. [MNL10]

Die Kontrollpunkte werden in Relation zu den bekannten 3D-Weltpunkten  $\mathbf{X}_i$  gewichtet bestimmt, um anschließend die gesuchten Kontrollpunkte der unbekanntenen 3D-Kamerakoordinaten  $\mathbf{x}_i^{\text{camera}}$  zu berechnen. Die Abbildung 6.3 veranschaulicht die Beziehung zwischen den Koordinaten in den unterschiedlichen Koordinatensystemen.



**Abbildung 6.3:** Virtuelle Kontrollpunkte  $\mathbf{c}$ . [MNL10]

Im Anschluss werden die vier Phasen

1. Festlegen von vier Kontrollpunkten  $\mathbf{c}_j^{\text{world}}$  für  $j = 1, \dots, 4$
2. Bestimmen der Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$  für  $j = 1, \dots, 4$
3. Berechnung der Kamerapunkte  $\mathbf{x}_i^{\text{camera}}$  für  $i = 1, \dots, n$
4. Ermittlung einer Transformationsmatrix:  $\mathbf{x}_i^{\text{camera}} = \mathbf{T} \cdot \mathbf{X}_i$  für  $i = 1, \dots, n$

des Algorithmus sowie eine enthaltende optionale Verbesserung der Genauigkeit (Gauß-Newton Optimierung) erläutert. Die Phasen beschreiben zuerst den Fall einer planaren Eingabemenge, bei der vier Kontrollpunkte benötigt werden. Im Anschluss daran wird der Unterschied zum nicht-planaren Fall erläutert. Dort finden im Algorithmus lediglich drei Kontrollpunkte Verwendung. Nähere Information bei einer nicht-planaren Punktmenge sind ab Seite 84 zu finden.

### 1. Festlegen von vier Kontrollpunkten $\mathbf{c}_j^{\text{world}}$ für $j = 1, \dots, 4$

Zu Beginn müssen die Gewichtungen  $\eta_{ij}$  und die vier Kontrollpunkte  $\mathbf{c}_j^{\text{world}}$  so festgelegt werden, dass das Resultat der aufsummierten Einzelmultiplikationen den homogenen 3D-Weltpunkten  $\mathbf{X}_i$  für  $i = 1, \dots, n$  entsprechen.

$$\mathbf{X}_i = \sum_{j=1}^4 \eta_{ij} \cdot \mathbf{c}_j^{\text{world}} \quad (6.29)$$

Die Gewichtungen  $\eta_{ij}$  sind homogene baryzentrische Koordinaten. Diese beschreiben eine Punktposition  $\mathbf{X}_i$  in Relation zu einer bekannten Strecke, Dreieck oder wie in diesem Fall ein Tetraeder. Die Gleichung 6.29 zeigt die Berechnung der 3D-Weltpunkte als gewichtete Summe der Kontrollpunkte  $\mathbf{c}_j^{\text{world}}$  durch eine Linearkombination. Die Summe der Koeffizienten (Gewichtungen  $\eta_{ij}$ ) beträgt dabei eins.

Analog zur Gleichung 6.29 mit

$$\sum_{j=1}^4 \eta_{ij} = 1 \quad (6.30)$$

gilt die Gleichung 6.31, welche für die spätere Berechnung der Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$  und den homogenen Kamerakoordinaten  $\mathbf{x}_i^{\text{camera}}$  Verwendung findet.

$$\mathbf{x}_i^{\text{camera}} = \sum_{j=1}^4 \eta_{ij} \cdot \mathbf{c}_j^{\text{camera}} \quad (6.31)$$

Für eine hohe Stabilität können die Kontrollpunkte  $\mathbf{c}_j^{\text{world}}$  des Weltkoordinatensystems folgendermaßen festgelegt werden (Vgl. [MNL07]). Der erste Kontrollpunkt  $\mathbf{c}_1^{\text{world}}$  ist der Schwerpunkt der zugehörigen Punktmenge  $\mathbf{X}_i$ .

$$\mathbf{c}_1^{\text{world}} = \frac{1}{n} \cdot \sum_{i=1}^n \mathbf{X}_i \quad (6.32)$$

Die restlichen drei Kontrollpunkte  $\mathbf{c}_j^{\text{world}}$  für  $j = 2, \dots, 4$  sind anschließend durch die räumliche Verteilung der Punktmenge so festzulegen, dass sie die räumliche Richtung der Punktmenge angeben.

In dieser Arbeit wird aus Effizienzgründen folgende Alternative vorgenommen. Der erste Kontrollpunkt  $\mathbf{c}_1^{\text{world}}$  ist der Ursprung des Weltkoordinatensystems. Im Anschluss sind die restlichen drei Kontrollpunkte  $\mathbf{c}_j^{\text{world}}$  für  $j = 2, \dots, 4$  so festzulegen, dass sie zusammen die räumliche Richtung beschreiben. Die Gleichungen 6.33 bis 6.36 zeigen die Belegungen der einzelnen Kontrollpunkte.

$$\mathbf{c}_1^{\text{world}} = [0, 0, 0]^T \quad (6.33)$$

$$\mathbf{c}_2^{\text{world}} = [1, 0, 0]^T \quad (6.34)$$

$$\mathbf{c}_3^{\text{world}} = [0, 1, 0]^T \quad (6.35)$$

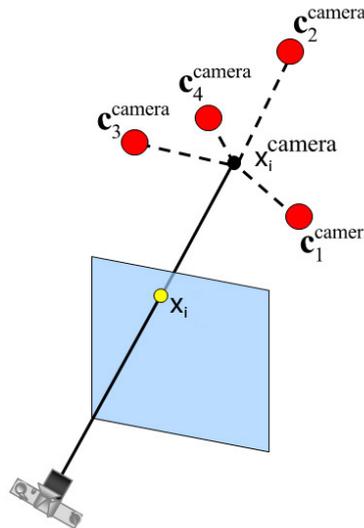
$$\mathbf{c}_4^{\text{world}} = [0, 0, 1]^T \quad (6.36)$$

## 2. Bestimmen der Kontrollpunkte $\mathbf{c}_j^{\text{camera}}$ für $j = 1, \dots, 4$

Die Berechnung der vier Kontrollpunkte der Kamerakoordinaten wird in den nachfolgenden sechs Sektionen (a) bis (f) beschrieben.

### a) Aufbau eines linearen Gleichungssystems $\mathbf{M} \cdot \mathbf{m} = 0$

Durch die vorhandenen intrinsischen Kameraparameter ist der zu jedem Punkt  $\mathbf{x}_i^{\text{camera}}$  im Kamerakoordinatensystem zugehörige 2D-Bildpunkt  $\mathbf{x}_i$  bekannt (Abbildung 6.4).



**Abbildung 6.4:** Zusammenhang zwischen Bildkoordinate  $\mathbf{x}_i$  und Kamerakoordinate  $\mathbf{x}_i^{\text{camera}}$ . [MNL10]

Anhand der Korrespondenz  $[x_i, x_i, 1]^T \leftrightarrow \mathbf{x}_i^{\text{camera}}$  zwischen 2D-Bildpunkt und seinem zugehörigen Punkt im Kamerakoordinatensystem kann die Gleichung 6.37 aufgestellt werden. Dabei beinhaltet  $w_{1 \times n}$   $n$  projektive Koeffizienten und  $\mathbf{K}_{3 \times 3}$  die intrinsischen Kameraparameter wie die Bildweite  $\mathbf{f} = [f_x, f_y]^T$ , den Scherungsfaktor  $s$  und den Bildhauptpunkt  $\mathbf{h} = [h_x, h_y]^T$ .

$$w_i \cdot [\mathbf{x}_i, 1]^T = \mathbf{K} \cdot \mathbf{x}_i^{\text{camera}} \quad \text{für } i = 1, \dots, n \quad (6.37)$$

Unter Einbeziehung von Gleichung 6.31 und den intrinsischen Kameraparametern

$$\mathbf{K} = \begin{bmatrix} f_x & f_x \cdot s & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix} \quad \text{mit } s = 0 \quad (6.38)$$

kann die nachfolgende Gleichung 6.39 konstruiert werden.

$$\forall i \in \{1, \dots, n\} : w_i \cdot \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \sum_{j=1}^4 \eta_{ij} \cdot \mathbf{c}_j^{\text{camera}} \quad (6.39)$$

Werden die projektiven Koeffizienten  $w_i$  mit  $\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}$  ausmultipliziert, ergibt sich der Vektor

$\begin{bmatrix} w_i \cdot x_{ix} \\ w_i \cdot x_{iy} \\ w_i \end{bmatrix}$  bei dem die letzte Reihe wiederum den projektiven Koeffizienten entsprechen. Mit der Gleichung 6.39 impliziert dass für die unbekannt projektiven Koeffizienten  $w_i$  folgendes:

$$w_i = [0, 0, 1] \cdot \sum_{j=1}^4 \eta_{ij} \cdot \mathbf{c}_j^{\text{camera}} = \sum_{j=1}^4 \eta_{ij} \cdot c_{jz}^{\text{camera}} \quad \text{für } i = 1, \dots, n \quad (6.40)$$

Dabei gibt  $c_{jz}^{\text{camera}}$  die z-Koordinate des Kontrollpunktes  $\mathbf{c}_j^{\text{camera}}$  an. Resultierend kann der Ausdruck 6.40 für die ersten zwei Reihen aus Gleichung 6.39 zur Substitution verwendet werden. Dadurch tauchen die projektiven Koeffizienten  $w$  nicht mehr in Gleichungen auf. Daraus ergeben sich zwei lineare Gleichungen

$$\sum_{j=1}^4 (\eta_{ij} \cdot f_x \cdot c_{jx}^{\text{camera}} + \eta_{ij} \cdot (h_x - x_{ix}) \cdot c_{jz}^{\text{camera}}) = 0 \quad (6.41)$$

und

$$\sum_{j=1}^4 (\eta_{ij} \cdot f_y \cdot c_{jy}^{\text{camera}} + \eta_{ij} \cdot (h_y - x_{iy}) \cdot c_{jz}^{\text{camera}}) = 0 \quad (6.42)$$

Anschließend kann mit Hilfe der zwei linearen Gleichungen 6.41 und 6.42 ein lineares Gleichungssystem  $\mathbf{M} \cdot \mathbf{m} = 0$  mit den 12 Unbekannten (Vier Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$  mit jeweils drei Dimensionen) aufgestellt werden. Bei dem Gleichungssystem beinhaltet  $\mathbf{m}$  die vier gesuchten Kontrollpunkte  $\mathbf{c}_{j3 \times 1}^{\text{camera}}$  als 12 x 1 Vektor und  $\mathbf{M}$  ist eine 2n x 12 konstante Matrix

$$\mathbf{M} = \begin{bmatrix} \eta_{11} \cdot f_x & 0 & \eta_{11} \cdot (h_x - x_{1x}) & \eta_{12} \cdot f_x & 0 & \eta_{12} \cdot (h_x - x_{1x}) & \eta_{13} \cdot f_x & 0 & \eta_{13} \cdot (h_x - x_{1x}) & \eta_{14} \cdot f_x & 0 & \eta_{14} \cdot (h_x - x_{1x}) \\ 0 & \eta_{11} \cdot f_y & \eta_{11} \cdot (h_y - x_{1y}) & 0 & \eta_{12} \cdot f_y & \eta_{12} \cdot (h_y - x_{1y}) & 0 & \eta_{13} \cdot f_y & \eta_{13} \cdot (h_y - x_{1y}) & 0 & \eta_{14} \cdot f_y & \eta_{14} \cdot (h_y - x_{1y}) \\ \vdots & \vdots \\ \eta_{n1} \cdot f_x & 0 & \eta_{n1} \cdot (h_x - x_{nx}) & \eta_{n2} \cdot f_x & 0 & \eta_{n2} \cdot (h_x - x_{nx}) & \eta_{n3} \cdot f_x & 0 & \eta_{n3} \cdot (h_x - x_{nx}) & \eta_{n4} \cdot f_x & 0 & \eta_{n4} \cdot (h_x - x_{nx}) \\ 0 & \eta_{n1} \cdot f_y & \eta_{n1} \cdot (h_y - x_{ny}) & 0 & \eta_{n2} \cdot f_y & \eta_{n2} \cdot (h_y - x_{ny}) & 0 & \eta_{n3} \cdot f_y & \eta_{n3} \cdot (h_y - x_{ny}) & 0 & \eta_{n4} \cdot f_y & \eta_{n4} \cdot (h_y - x_{ny}) \end{bmatrix}$$

mit den Bildkoordinaten  $\mathbf{x}_i$ , den Gewichtungen  $\eta_{ij}$  und den intrinsischen Kameraparametern aus der Matrix  $\mathbf{K}$ .

b) Beschreibung der Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$  für  $j = 1, \dots, 4$  als Linearkombination der Null-Eigenvektoren

Die Lösung des linearen Gleichungssystems  $\mathbf{M} \cdot \mathbf{m} = \mathbf{0}$  befindet sich im Nullraum<sup>27</sup> von  $\mathbf{M}$  und wird durch

$$\mathbf{m} = \sum_{i=1}^N \beta_i \cdot \mathbf{v}_i \quad (6.43)$$

ausgedrückt. Dabei sind  $\mathbf{v}_i$  für  $i = 1, \dots, N$  Paare aus Spalten des rechten Singulärvektors der Matrix  $\mathbf{M}$  korrespondierend zu den  $n$  Null-Singulärwerten<sup>28</sup> von  $\mathbf{M}$  (Vgl. Kapitel 3.2, S. 13). Die Null-Eigenvektoren der  $12 \times 12$  resultierenden Matrix aus  $\mathbf{M}^T \cdot \mathbf{M}$

$$\mathbf{M} \mathbf{m} = \mathbf{0} \quad \Rightarrow \quad \mathbf{M}^T \mathbf{M} \mathbf{m} = \mathbf{0}$$

**Abbildung 6.5:** Umformen des linearen Gleichungssystems zu optimalen Matrix- und Vektorgrößen. [MNLF10]

führen in einer Laufzeit von  $O(n)^{29}$  zu den  $\mathbf{v}_i$  Paaren (Gleichung 6.44).

$$\text{eig}(\mathbf{M}^T \cdot \mathbf{M}) = \mathbf{v}_{12 \times 12} \quad (6.44)$$

Bei den sogenannten Null-Eigenvektoren sind die zu den Eigenvektoren zugehörigen Eigenwerte null. Es sei jedoch zu beachten, dass ein Eigenvektor in keiner Situation dem Nullvektor entspricht.

In der Gleichung 6.43 sind die  $\mathbf{v}_i$  Paare durch die Eigenvektoren von  $(\mathbf{M}^T \cdot \mathbf{M})$  bekannt. Die vier Kontrollpunkte als  $12 \times 1$  Vektor  $\mathbf{m}$ , die Koeffizienten  $\beta_i$  und die Dimension  $N$  des Nullraumes von  $\mathbf{M}^T \cdot \mathbf{M}$  sind derzeit unbekannt. In Bezug auf die Dimension  $N$  fand [MNLF07] heraus, dass sie lediglich für  $N \in \{1, 2, 3, 4\}$  betrachtet werden muss. Aus diesem Grund sind die vier Koeffizienten  $\beta_i$  in Abhängigkeit seiner zugehörigen  $\mathbf{v}$  Werte zu betrachten:

$$N = 1 : \mathbf{m} = \beta_1 \cdot \mathbf{v}_1 \quad (6.45)$$

$$N = 2 : \mathbf{m} = \beta_1 \cdot \mathbf{v}_1 + \beta_2 \cdot \mathbf{v}_2 \quad (6.46)$$

$$N = 3 : \mathbf{m} = \beta_1 \cdot \mathbf{v}_1 + \beta_2 \cdot \mathbf{v}_2 + \beta_3 \cdot \mathbf{v}_3 \quad (6.47)$$

$$N = 4 : \mathbf{m} = \beta_1 \cdot \mathbf{v}_1 + \beta_2 \cdot \mathbf{v}_2 + \beta_3 \cdot \mathbf{v}_3 + \beta_4 \cdot \mathbf{v}_4 \quad (6.48)$$

Nachdem jedes  $N$  separat betrachtet wurde, bestimmt der kleinste Rückprojektionsfehler

$$\sum_{i=1}^n \text{dist}(\tilde{\mathbf{u}}, \mathbf{x}_i)^2 \quad (6.49)$$

<sup>27</sup>Der Nullraum einer Matrix sind alle Elemente, welche auf null bzw. das neutrale Element abgebildet werden. Des Weiteren wird der Nullraum auch als Kern bezeichnet.

<sup>28</sup>Mit Null-Singulärwerten sind singuläre Werte gemeint, die null sind.

<sup>29</sup>Die Berechnungsdauer bezieht sich auf ein ausreichend großes  $n \geq 15$ . Diese Berechnung von  $\mathbf{M}^T \cdot \mathbf{M}$  nimmt im Algorithmus die größte Zeit in Anspruch.

das zu verwendende  $N^{30}$ . Dabei ist der, in homogenen Koordinaten angegebene, Punkt

$$\tilde{\mathbf{u}} = \mathbf{K} \cdot [\mathbf{R} \mid \mathbf{t}] \cdot [\mathbf{X}_i, 1]^T \quad (6.50)$$

die Rückprojektion vom Weltpunkt  $\mathbf{X}_i$  zum Bildpunkt  $\mathbf{x}_i$ . Genaugenommen wird der Weltpunkt  $\mathbf{X}_i$  mit einer Transformation  $[\mathbf{R} \mid \mathbf{t}]$  in das Kamerakoordinatensystem transformiert. Anhand der intrinsischen Kameraparameter, welche sich in der Kamera-Kalibriermatrix  $\mathbf{K}$  befinden, kann der Punkt in das Bildkoordinatensystem transformiert werden. Anschließend ist dieser Punkt  $\tilde{\mathbf{u}}$  mit dem Bildpunkt  $\mathbf{x}_i$  vergleichbar.

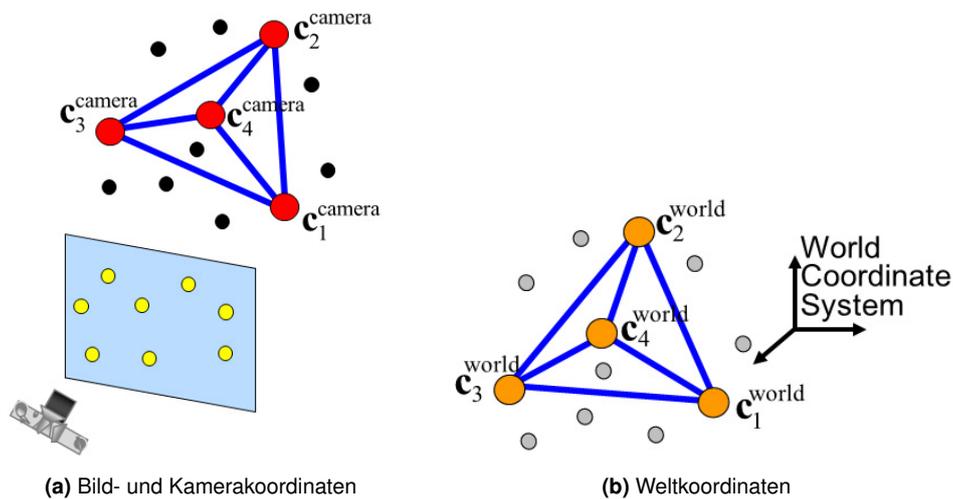
Zu diesem Zeitpunkt ist die passende Dimension  $N \in \{1, 2, 3, 4\}$  und damit die nachfolgend zu behandelnde Gleichung aus 6.45 bekannt. Bevor die Berechnung dieser Gleichung bzw. den  $\beta$ -Werten stattfinden kann, werden in der nächsten Phase die einzuhaltende Stabilitätsbedingungen erläutert.

### c) Hinzufügen von Stabilitätsbedingungen

Im dritten Schritt wird der Fokus auf die Abstände zwischen den vier Kamera-Kontrollpunkten  $\mathbf{c}_j^{\text{camera}}$  und den Abständen der vier Welt-Kontrollpunkte  $\mathbf{c}_j^{\text{world}}$  gelegt. Denn diese Distanzen müssen zueinander bewahrt werden. Die nachfolgende Gleichung

$$\|\mathbf{c}_i^{\text{camera}} - \mathbf{c}_j^{\text{camera}}\| = \|\mathbf{c}_i^{\text{world}} - \mathbf{c}_j^{\text{world}}\| \quad (6.51)$$

und die Abbildung 6.6 veranschaulichen eine Möglichkeit. Die sechs unbekanntene Kontrollpunktdistanzen der Kamera sind mit den zugehörigen sechs bekannten Kontrollpunktabständen der Welt gleichzustellen.



**Abbildung 6.6:** Bewahrung der Kontrollpunktabstände. [MNL10]

<sup>30</sup>Es hat sich in dieser Arbeit herausgestellt, dass ausgehend der ersten Linearkombination nur die Nächste betrachtet werden muss, wenn der Rückprojektionsfehler der aktuellen Linearkombination über einen Schwellwert liegt. Der Schwellwert wurde empirisch ermittelt und bezieht sich auf den mittleren Rückprojektionsfehler der gesamten Punktmenge.

d) Berechnung der Gewichte  $\beta_i$  abhängig zur Dimension ( $N \in \{1, 2, 3\}$ : Linearisierung |  $N = 4$ : Neularisierung)

In Abhängigkeit der ermittelten Dimension  $N \in \{1, 2, 3, 4\}$  wird das weitere Vorgehen beschrieben.

i. Linearkombination für  $N = 1$ :  $\mathbf{m} = \beta_1 \cdot \mathbf{v}_1$

Sei  $\mathbf{v}_1^{[i]}$  ein  $3 \times 1$  Vektor des  $12 \times 1$  Vektors  $\mathbf{v}_1$ , welcher zusammen mit  $\beta_1$  zu den Kontrollpunkten  $\mathbf{m}_{12 \times 1} = [(\mathbf{c}_1^{\text{camera}})^T, \dots, (\mathbf{c}_4^{\text{camera}})^T]^T$  korrespondiert. Dann können, mit der Gleichung 6.51, dem bekannten  $\mathbf{v}_1^{[i]}$  Vektor und den Kontrollpunkten  $\mathbf{c}_j^{\text{world}}$ , die Abstände

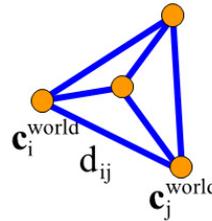
$$\|\beta_1 \cdot \mathbf{v}_1^{[i]} - \beta_1 \cdot \mathbf{v}_1^{[j]}\|^2 = \|\mathbf{c}_i^{\text{world}} - \mathbf{c}_j^{\text{world}}\|^2 \quad (6.52)$$

gleichgesetzt werden, um nachfolgend  $\beta_1$  in einer geschlossenen Form

$$\beta_1 = \frac{\sum_{\{i,j\} \in [1;4]} \|\mathbf{v}_1^{[i]} - \mathbf{v}_1^{[j]}\| \cdot \|\mathbf{c}_i^{\text{world}} - \mathbf{c}_j^{\text{world}}\|}{\sum_{\{i,j\} \in [1;4]} \|\mathbf{v}_1^{[i]} - \mathbf{v}_1^{[j]}\|^2} \quad (6.53)$$

zu berechnen.

Die Abbildung 6.7 veranschaulicht den Abstand  $d_{ij}$  zwischen dem Kontrollpunkt  $\mathbf{c}_i^{\text{world}}$  und  $\mathbf{c}_j^{\text{world}}$ .



**Abbildung 6.7:** Abstand zwischen zwei Kontrollpunkten. [MNL10]

ii. Linearkombination für  $N = 2$ :  $\mathbf{m} = \beta_1 \cdot \mathbf{v}_1 + \beta_2 \cdot \mathbf{v}_2$

In dem zweiten Fall wird die Abstandsgleichung zu

$$\|(\beta_1 \cdot \mathbf{v}_1^{[i]} + \beta_2 \cdot \mathbf{v}_2^{[i]}) - (\beta_1 \cdot \mathbf{v}_1^{[j]} + \beta_2 \cdot \mathbf{v}_2^{[j]})\|^2 = \|\mathbf{c}_i^{\text{world}} - \mathbf{c}_j^{\text{world}}\|^2 \quad (6.54)$$

Die zwei unbekanntenen  $\beta_1$  und  $\beta_2$  befinden sich in den quadratischen Termen und können mit einer Linearisierung berechnet werden. Diese beinhaltet die neu eingeführten Werte  $\{\beta_{11}, \beta_{12}, \beta_{22}\}$ , welche in Verbindung zueinander stehen:

$$\beta_{11} = \beta_1^2 \quad (6.55)$$

$$\beta_{12} = \beta_1 \cdot \beta_2 \quad (6.56)$$

$$\beta_{22} = \beta_2^2 \quad (6.57)$$

Mit Hilfe der vier Kontrollpunkte kann ein lineares Gleichungssystem von sechs Gleichungen aufgebaut werden, bei dem  $\beta_{ab}$  mit  $ab \in \{11, 12, 22\}$  unbekannt sind.

$$\mathbf{L} \cdot \boldsymbol{\beta} = \mathbf{D} \quad (6.58)$$

$$\begin{bmatrix} l_{1,1} & \cdots & l_{1,3} \\ \vdots & \ddots & \vdots \\ l_{6,1} & \cdots & l_{6,3} \end{bmatrix} \cdot \begin{bmatrix} \beta_{11} \\ \beta_{12} \\ \beta_{22} \end{bmatrix} = \begin{bmatrix} d_{12}^2 \\ d_{13}^2 \\ d_{14}^2 \\ d_{23}^2 \\ d_{24}^2 \\ d_{34}^2 \end{bmatrix} \quad (6.59)$$

Die  $6 \times 3$  Matrix  $\mathbf{L}$  aus Gleichung 6.58 enthält die durch  $\mathbf{v}_1$  und  $\mathbf{v}_2$  berechneten Elemente  $l_{1,1} \dots l_{6,3}$ . Der  $\mathbf{D}$ -Vektor beinhaltet die sechs quadratischen Abstände  $\|\mathbf{c}_i^{\text{world}} - \mathbf{c}_j^{\text{world}}\|^2$ . Die Lösung des Gleichungssystems erfolgt mit der Pseudoinversen<sup>31</sup> von  $\mathbf{L}$  und der Festlegung des Vorzeichens von  $\beta_a$ , sodass alle Kamerakoordinaten  $\mathbf{x}_i^{\text{camera}}$  eine positive  $z$ -Koordinate besitzen.

iii. Linearkombination für  $N = 3$ :  $m = \beta_1 \cdot \mathbf{v}_1 + \beta_2 \cdot \mathbf{v}_2 + \beta_3 \cdot \mathbf{v}_3$

Die sechs Abstandsgleichungen sind entsprechend wie beim zweiten Fall (Gleichung 6.54) aufzustellen. Dies führt wiederum zu einem linearen Gleichungssystem  $\mathbf{L} \cdot \boldsymbol{\beta} = \mathbf{D}$ , welches jedoch aufgrund der neuen Linearkombination  $\mathbf{x} = \beta_1 \cdot \mathbf{v}_1 + \beta_2 \cdot \mathbf{v}_2 + \beta_3 \cdot \mathbf{v}_3$  eine größere Dimension besitzt. Konstruiert aus den Elementen  $\mathbf{v}_1 \dots \mathbf{v}_3$  vergrößert sich  $\mathbf{L}$  zu einer quadratischen  $6 \times 6$  Matrix. Der Vektor  $\boldsymbol{\beta}$  steigt von drei auf sechs Dimensionen an  $[\beta_{11}, \beta_{12}, \beta_{22}]^T \rightarrow [\beta_{11}, \beta_{12}, \beta_{13}, \beta_{22}, \beta_{23}, \beta_{33}]^T$ . Die weitere Vorgehensweise ist wie beim zweiten Fall ( $N = 2$ ), außer dass nicht die Pseudoinverse von  $\mathbf{L}$  sondern deren Inverse zu berechnen ist.

iv. Linearkombination für  $N = 4$ :  $\mathbf{m} = \beta_1 \cdot \mathbf{v}_1 + \beta_2 \cdot \mathbf{v}_2 + \beta_3 \cdot \mathbf{v}_3 + \beta_4 \cdot \mathbf{v}_4$

Im vierten Fall sind in der Linearkombination  $\beta_j$  für  $j = 1, \dots, 4$  unbekannt. Für eine Berechnung dieser vier Werte ist eine sogenannte *Neularisierung* (Vgl. [KS99]) durchzuführen. Eine Linearisierung wie in den vorherigen Fällen reicht nicht mehr aus. Denn es liegen sechs quadratische Gleichungen vor, jedoch ergibt die Linearisierung zehn Unbekannte. Das unterbestimmte lineare Gleichungssystem ist nachfolgend in

$$\mathbf{L} \cdot \boldsymbol{\beta} = \mathbf{D} \quad (6.60)$$

und

<sup>31</sup>Die Pseudoinverse einer Matrix kann als Verallgemeinerung der inversen Matrix auf singuläre und nichtquadratische Matrizen aufgefasst werden. In dem Fall, dass ein lineares Gleichungssystem nicht zu lösen ist, findet die Pseudoinverse häufig Verwendung. Sie ermöglicht eine Annäherung der gesuchten Lösung. Für weitere Informationen sei auf das Kapitel 7.6 verwiesen. Dort wird ab Seite 104 die Berechnung der Pseudoinversen erläutert.

$$\begin{bmatrix} l_{1,1} & \cdots & l_{1,10} \\ \vdots & \ddots & \vdots \\ l_{6,1} & \cdots & l_{6,10} \end{bmatrix} \cdot \begin{bmatrix} \beta_{11} \\ \beta_{12} \\ \beta_{13} \\ \beta_{14} \\ \beta_{22} \\ \beta_{23} \\ \beta_{24} \\ \beta_{33} \\ \beta_{34} \\ \beta_{44} \end{bmatrix} = \begin{bmatrix} d_{12}^2 \\ d_{13}^2 \\ d_{14}^2 \\ d_{23}^2 \\ d_{24}^2 \\ d_{34}^2 \end{bmatrix} \quad (6.61)$$

dargestellt.

Die Neularisierung macht prinzipiell das Gleiche wie bei der Berechnung der Kontrollpunkte. Die Lösung für  $\beta_{ab}$  befindet sich im Nullraum des ersten homogenen linearen Gleichungssystems 6.52. Dabei müssen die quadratischen Abstandsgleichungen (Vgl. Gleichung 6.54) der neuen Linearkombination  $\beta_1 \cdot \mathbf{v}_1 + \beta_2 \cdot \mathbf{v}_2 + \beta_3 \cdot \mathbf{v}_3 + \beta_4 \cdot \mathbf{v}_4$  angepasst werden. Damit ist das Gleichungssystem erneut per Linearisierung zu lösen (Neularisierung).

Die neuen Abstandsgleichungen können durch die Kommutativität der Multiplikation

$$\beta_{ab} \cdot \beta_{cd} = \beta_a \cdot \beta_b \cdot \beta_c \cdot \beta_d = \beta_{a'b'} \cdot \beta_{c'd'} \quad (6.62)$$

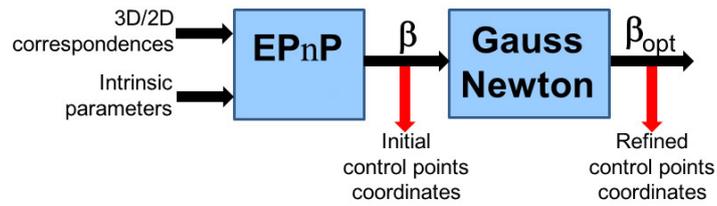
abgeleitet werden. In der Gleichung 6.62 repräsentiert  $\{a', b', c', d'\}$  jede Permutation der Zahlen  $\{a, b, c, d\}$ .

Die bisherigen Berechnungen von  $\beta$  für die vier Fälle ( $N \in \{1, 2, 3, 4\}$ ) wurden für eine nicht-planare Eingabemenge erläutert. Bei einer planaren Eingabemenge werden anstatt vier- lediglich drei Kontrollpunkte verwendet. Dies führt dazu, dass sich die Dimension der Matrix  $\mathbf{M}$  von  $2 \cdot n \cdot 12$  auf  $2 \cdot n \cdot 9$  verkleinert.  $\mathbf{M}$  verfügt damit über einen neun-dimensionalen Eigenvektor  $\mathbf{v}_i$ . Der größte Unterschied zum nicht-planaren Fall liegt darin, dass sich jetzt die Anzahl der quadratischen Gleichung von sechs auf drei reduziert. Aus diesem Grund reicht im Fall ( $N = 3$ ) eine Linearisierung nicht alleine aus, was die Heranziehung der Neularisierung zur Folge hat.

An dieser Stelle angekommen, sind die  $\beta$  Werte der gewählten Linearkombination bestimmt worden. Das ermöglicht die Berechnung der vier gesuchten Kamera-Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$ . Zunächst soll jedoch eine Möglichkeit zur Verbesserung der berechneten  $\beta$  Werte vorgestellt werden.

#### e) Steigerung der Genauigkeit durch Gauß-Newton Optimierung (Optional)

Die im letzten Schritt berechneten  $\beta_i$  Werte für  $i = 1, \dots, N$  können als Initialisierung der iterativen Gauß-Newton Optimierung verwendet werden (Abbildung 6.8). Dies führt zu einer hohen Genauigkeit bei niedrigem Zeitaufwand.



**Abbildung 6.8:** Korrektur der Kontrollpunkte mit dem Gauß-Newton-Algorithmus. [MNL10]

Die Lösung des linearen Gleichungssystems  $\mathbf{M} \cdot \mathbf{m} = 0$  wurde auf der Seite 80 durch

$$\mathbf{m} = \sum_{i=1}^N \beta_i \cdot \mathbf{v}_i \quad (6.63)$$

ausgedrückt. Für eine Steigerung der Genauigkeit<sup>32</sup> sind die  $\beta_i$  Werte zu verfeinern. Dies erfolgt mit dem Gauß-Newton-Algorithmus, indem eine Minimierung der Kontrollpunktabstände

$$error(\beta) = \sum_{(i,j) \text{ mit } i < j} \left( \|\mathbf{c}_i^{\text{camera}} - \mathbf{c}_j^{\text{camera}}\|^2 - \|\mathbf{c}_i^{\text{world}} - \mathbf{c}_j^{\text{world}}\|^2 \right) \quad (6.64)$$

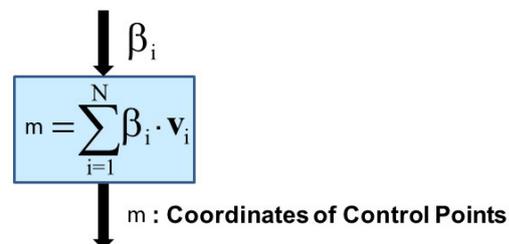
durchgeführt wird.

Abschließend sei erwähnt, dass diese Sektion (2.e) für die Bestimmung von  $\mathbf{T}$  bzw.  $\mathbf{P}$  optional ist. Das nicht-iterative EPnP-Verfahren wird an dieser Stelle um eine iterative Komponente (Gauß-Newton-Algorithmus) ergänzt. Allerdings ist der zeitliche Aufwand, aufgrund der geringen Dimension der  $\beta$  Werte, sehr gering.

f) Berechnung der Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$  für  $j = 1, \dots, 4$

Für die Berechnung der vier Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$  sind die  $\mathbf{v}_i$  Paare,  $\beta_i$  Werte für  $i = 1, \dots, N$  und die Dimension  $N$  notwendig.

In der Sektion (2.b) auf Seite 80 wurden die Dimension und  $\mathbf{v}$  bestimmt. Die Dimension konnte durch die Linearkombination mit den kleinsten Rückprojektionsfehler ermittelt werden. Die  $\mathbf{v}_i$  Paare resultierten aus den Eigenvektoren der Matrix  $(\mathbf{M}^T \cdot \mathbf{M})$ , welche dem linearen Gleichungssystem  $\mathbf{M} \cdot \mathbf{m} = 0$  bzw.  $\mathbf{M} \cdot \mathbf{M}^T \cdot \mathbf{m} = 0$  zugehörig sind. Nach der Einführung von Stabilitätsbedingungen ermöglichten die bekannten  $\mathbf{v}_i$  Paare und die Dimension, die Berechnung der  $\beta_i$  Werte. Dafür wurden vier verschiedenen Berechnungsmöglichkeiten ab Seite 81 vorgestellt. Diese ergaben sich in Abhängigkeit zur Dimension.



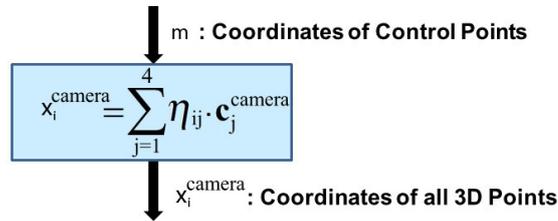
**Abbildung 6.9:** Ermittlung der Kontrollpunkte. [MNL10]

Die Abbildung 6.9 veranschaulicht das weitere Vorgehen in dieser Sektion (2.f).  $\mathbf{v}$  ist mit den  $\beta$  Werten einzeln zu multiplizieren. Die Summe der Einzelmultiplikationen ergibt die vier gesuchten Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$ .

<sup>32</sup>In Bezug auf die gesuchte Transformationsmatrix  $\mathbf{T}$ , welche die Abbildung  $\mathbf{x}_i^{\text{camera}} = \mathbf{T} \cdot \mathbf{X}_i$  beschreibt.

### 3. Berechnung der Kamerapunkte $\mathbf{x}_i^{\text{camera}}$ für $i = 1, \dots, n$

Mit Hilfe der in Sektion (2.f) berechneten vier Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$  sind die  $n$  Kamerapunkte  $\mathbf{x}_i^{\text{camera}}$  ermittelbar. Dafür sind die Gewichtungen  $\eta_{ij}$ , angegeben in homogenen baryzentrischen Koordinaten und beschrieben in Phase (1) auf Seite 77, heranzuziehen.

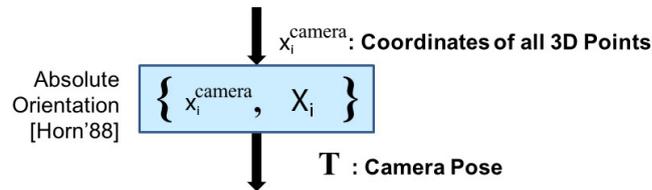


**Abbildung 6.10:** Berechnung der Kamerapunkte. [MNL10]

Die Abbildung 6.10 zeigt die Berechnung der Kamerapunkte als gewichtete Summe der Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$  durch eine Linearkombination. Die Summe der Koeffizienten (Gewichtungen  $\eta_{ij}$ ) beträgt dabei eins. Des Weiteren befinden sich die vier Kontrollpunkte  $\mathbf{c}_j^{\text{camera}}$  in dem  $12 \times 1$  Vektor  $\mathbf{m} = [(\mathbf{c}_1^{\text{camera}})^T, \dots, (\mathbf{c}_4^{\text{camera}})^T]^T$ .

### 4. Ermittlung einer Transformationsmatrix mit der Abbildung $\mathbf{x}_i^{\text{camera}} = \mathbf{T} \cdot \mathbf{X}_i$

Mit den korrespondierenden 3D-Punkten  $\mathbf{x}_i^{\text{camera}} \longleftrightarrow \mathbf{X}_i$  kann die Transformationsmatrix  $\mathbf{T}$  ermittelt werden (Vgl. Abbildung 6.11).



**Abbildung 6.11:** Bestimmung der Kamerapose durch Punktkorrespondenzen. [MNL10]

Dabei besteht  $\mathbf{T}$  aus der Rotation  $\mathbf{R}$  und Translation  $\mathbf{t}$  und beschreibt die Abbildung  $\mathbf{X}_i \mapsto \mathbf{x}_i^{\text{camera}}$ . Die Transformationsmatrix wird durch die Methode von [HHN88] bestimmt.

Zu Beginn ist der arithmetische Mittelwert

$$\bar{\mathbf{X}}_{\text{arith}} = \frac{1}{n} \cdot \sum_{i=1}^n \mathbf{X}_i \quad (6.65)$$

der Weltkoordinaten  $\mathbf{X}_i$  und

$$\bar{\mathbf{x}}_{\text{arith}}^{\text{camera}} = \frac{1}{n} \cdot \sum_{i=1}^n \mathbf{x}_i^{\text{camera}} \quad (6.66)$$

der Kamerakoordinaten  $\mathbf{x}_i^{\text{camera}}$  zu ermitteln. Dafür werden jeweils alle Welt- und Kamerakoordinaten aufsummiert und durch die Anzahl der aufaddierten Punkte dividiert. Ist der Schwerpunkt der jeweiligen Punktmenge bekannt, dann können die Punkte normalisiert werden. Die Gleichungen

$$\mathbf{x}_i^{\text{camera}} = \mathbf{x}_i^{\text{camera}} - \bar{\mathbf{x}}_{\text{arith}}^{\text{camera}} \quad \text{für } i = 1, \dots, n \quad (6.67)$$

und

$$\mathbf{X}_i = \mathbf{X}_i - \bar{\mathbf{X}}_{\text{arith}} \quad \text{für } i = 1, \dots, n \quad (6.68)$$

subtrahieren alle Punkte von ihrem zugehörigen Schwerpunkt. Anschließend sind die Korrespondenzen einzeln miteinander zu multiplizieren. Dabei muss für die Multiplikation jede Weltkoordinate  $\mathbf{X}$  transponiert werden. Die aufsummierten Resultate werden in einer  $3 \times 3$  Matrix

$$\mathbf{A} = \sum_{i=1}^n (\mathbf{x}_i^{\text{camera}}) \cdot \mathbf{X}_i^T \quad (6.69)$$

festgehalten, um sie mit der Singulärwertzerlegung (Vgl. Kapitel 3.2)

$$\text{svd}(\mathbf{A}) = [\mathbf{U} \mathbf{S} \mathbf{V}] \quad (6.70)$$

in die drei Matrizen  $\mathbf{U}$ ,  $\mathbf{S}$  und  $\mathbf{V}$  zu zerlegen. Für die Berechnung der  $3 \times 3$  Rotationsmatrix

$$\mathbf{R} = \mathbf{U} \cdot \mathbf{V}^T \quad (6.71)$$

ist  $\mathbf{U}$  mit der transponierten  $\mathbf{V}$ -Matrix zu multiplizieren. Das Vorzeichen der Rotationsmatrix bestimmt ihre Determinante. In dem Fall, dass die Determinante von  $\mathbf{R}$  unter null ist, muss das Vorzeichen negiert werden. Andernfalls ist das Vorzeichen richtig.

Der Translationsanteil

$$\mathbf{t} = (\bar{\mathbf{x}}_{\text{arith}}^{\text{camera}}) - \mathbf{R} \cdot (\bar{\mathbf{X}}_{\text{arith}}) \quad (6.72)$$

der gesuchten Transformationsmatrix  $\mathbf{T}$  wird anhand der ermittelten Rotationsmatrix und den zwei Schwerpunkten  $\bar{\mathbf{x}}_{\text{arith}}^{\text{camera}}$  und  $\bar{\mathbf{X}}_{\text{arith}}$  bestimmt. Dafür ist zuerst der Schwerpunkt der Weltkoordinaten mit der Rotationsmatrix zu multiplizieren. Das Berechnungsergebnis wird danach von dem Schwerpunkt der Kamerakoordinaten subtrahiert. Das Resultat ist der Translationsvektor der zusammen mit der Rotationsmatrix die Transformationsmatrix

$$\mathbf{T}_{3 \times 4} = [\mathbf{R}_{3 \times 3} \mid \mathbf{t}_{3 \times 1}] \quad (6.73)$$

vervollständigt. Die Matrix  $\mathbf{T}$  ermöglicht die Abbildung

$$\mathbf{x}_i^{\text{camera}} = \mathbf{T} \cdot \mathbf{X}_i \quad (6.74)$$

der Weltpunkte (Weltkoordinatensystem) auf die korrespondierenden Punkte im Kamerakoordinatensystem. Für eine Abbildung

$$\mathbf{x}_i = \mathbf{K} \cdot \mathbf{T} \cdot \mathbf{X}_i = \mathbf{P} \cdot \mathbf{X}_i \quad (6.75)$$

auf das Bildkoordinatensystem sind entsprechend die intrinsischen Kameraparameter  $\mathbf{K}$  heranzuziehen.

Im Anhand A.4 auf Seite 131 ist der Programmiercode der vierten Phase in Kurzform veröffentlicht.

# 7 Objektmobilisierung mit einem Greifarm

Dieses Kapitel beschreibt die räumliche Bewegung von Gegenständen mit einem Greifarm. Der Roboterarm besitzt inklusive Greifer sechs Freiheitsgrade und wurde im technischen Aufbau im Kapitel 4.1 beschrieben. Die Mobilisierung von greifbaren Gegenständen gliedert sich in neun Phasen, welche nachfolgend erläutert werden.

## Inhalt

---

7.1	Vorbedingungen . . . . .	88
7.2	Startposition anfahren . . . . .	89
7.3	Identifizierung . . . . .	90
7.4	Berechnung der Zielpose abhängig zur Objektpose . . . . .	91
7.5	Korrektur der Zielpose . . . . .	100
7.6	Inverse Kinematik der Zielpose . . . . .	100
7.7	Berechnung der Greifpose abhängig zur Zielpose . . . . .	106
7.8	Inverse Kinematik der Greifpose . . . . .	107
7.9	Mobilisierung . . . . .	107

---

## 7.1 Vorbedingungen

Die unterschiedlichen Formen und Materialien eines Gegenstandes führen dazu, dass jeder Gegenstand individuell gegriffen werden muss. Zu den Vorbedingungen gehört daher eine Abbildungsmatrix  $\mathbf{T}_{obj}^{\text{gripcam}_{init}} = \mathbf{T}_{obj}^{\text{targetcam}_{init}} \cdot \mathbf{T}_{\text{targetcam}_{init}}^{\text{gripcam}_{init}}$ , welche die Transformation vom Objekt- zum Kamerakoordinatensystem der Greifpose abbildet (Relative Pose zwischen Objekt- und Greifpose).

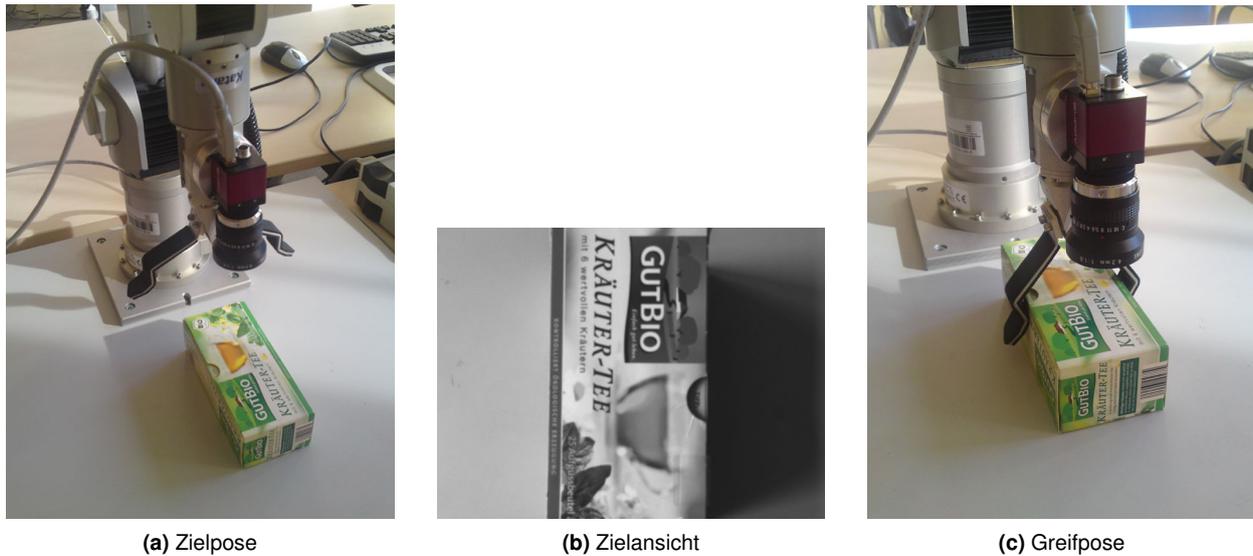
Falls nicht bereits vorhanden muss diese für den Gegenstand berechnet werden. Dafür sind drei Schritte notwendig. Zu Beginn wird der Greifarm in eine sogenannte Zielpose  $\mathbf{T}_{base}^{\text{targetcam}_{init}}$  gefahren (Grafik 7.1a). In dieser Pose wird eine Zielansicht (Grafik 7.1b) aufgenommen, welche anschließend auf Merkmale untersucht wird. Eine Korrespondenzsuche zwischen diesen Merkmalen und dem aus Kapitel 5 bekannten gegenstands-spezifischen 3D-Merkmalmodell führt zur Abbildungsmatrix  $\mathbf{T}_{obj}^{\text{targetcam}_{init}}$  (Relative Pose zwischen Objekt- und Zielpose). Damit ist es möglich den Roboterarm immer in diese Zielpose zu bewegen, wenn die Objektlage und -ausrichtung berechnet wurde. Die Berechnung von  $\mathbf{T}_{obj}^{\text{targetcam}_{init}}$ , anhand von 2D-zu-3D-Korrespondenzen, wurde im Kapitel 6 (*Posenbestimmung*) beschrieben.

Weiterführend kann die gewünschte Greifpose relativ zur Zielpose festgelegt werden. Dafür ist  $\mathbf{T}_{\text{targetcam}_{init}}^{\text{gripcam}_{init}}$  beliebig festzulegen. Im einfachsten Fall ist dies lediglich eine räumliche Verschiebung ohne eine Veränderung der Kameraausrichtung. Beispielsweise zeigt die Grafik 7.1c eine Greifpose, die ausgehend von der Zielpose um wenige Zentimeter in  $x$ - und  $z$ -Richtung verschoben wurde.

Es ist jedoch wichtig, dass die relative Pose  $\mathbf{T}_{\text{targetcam}_{init}}^{\text{gripcam}_{init}}$  im Basiskoordinatensystem  $base$  berechnet wird. Die relative Pose wäre sonst nur für den Gegenstand gültig, wenn sich dieser in der initialen räumlichen Lage und Ausrichtung befindet. Aus diesem Grund muss mit

$$\mathbf{T}_{\text{targetcam}_{init}}^{\text{gripcam}_{init}} = \text{inv} \left( \mathbf{T}_{base}^{\text{targetcam}_{init}} \right) \cdot \text{inv} \left( \mathbf{T}_{base}^{\text{gripcam}_{init}} \right) \quad (7.1)$$

der relative Versatz und die relative Ausrichtung zwischen Ziel- und Greifpose berechnet werden. Dadurch ist die relative Pose immer gültig, wenn sie als Transformation im Basiskoordinatensystem durchgeführt wird.



**Abbildung 7.1:** Initial festzulegende Zielpose, Zielansicht und Greifpose eines Gegenstandes.

Nachfolgend kann der Roboterarm anhand von  $\mathbf{T}_{obj}^{gripcam_{init}}$  in die festgelegte Greifpose bewegt werden, wenn die Pose  $\mathbf{T}_{base}^{obj}$  des Gegenstandes bekannt ist.

Dabei besitzt eine Pose die Form

$$\mathbf{T} = [\mathbf{R} \mid \mathbf{t}] = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \end{bmatrix} \quad (7.2)$$

und wurde im Kapitel 6 erläutert. In der Gleichung 7.2 geben  $\mathbf{t}$  und  $\mathbf{R}$  die relative räumliche Verschiebung und Orientierung an.

Alternativ zur Gleichung 7.2 erlaubt die sogenannte RPY-Winkelkonvention eine kompaktere Darstellung der ursprünglichen  $3 \times 3$  Rotationsmatrix. Die RPY-Winkelkonvention beschreibt die räumliche Rotation mit lediglich drei  $[\phi, \psi, \vartheta]$  anstatt neun  $[R_{11}, \dots, R_{33}]$  Parametern. Das Y steht für *gieren* (engl. *yaw*) und beschreibt die Rotation  $\phi$  um die  $x$ -Achse. Das P entstammt der englischen Bezeichnung *nicken* (engl. *pitch*) und definiert die Drehung  $\psi$  um die  $y$ -Achse. Die Rotation  $\vartheta$  um die  $z$ -Achse ist dem Buchstaben R von *rollen* (engl. *roll*) zugeordnet. Ebenfalls ist diese kompaktere Darstellung auch als ZYX-Eulerwinkel bekannt (Vgl. [HOR07]).

Zusammenfassend ergeben sich drei Vorbedingungen, welche für einen Greifvorgang erfüllt sein müssen. Ein dem Gegenstand zugehöriges 3D-Merkmalmodell aus Kapitel 5 (*Generierung eines 3D-Merkmalmodells*) und die zwei Transformationen  $\mathbf{T}_{obj}^{targetcam_{init}}$  und  $\mathbf{T}_{targetcam_{init}}^{gripcam_{init}}$  sind erforderlich.

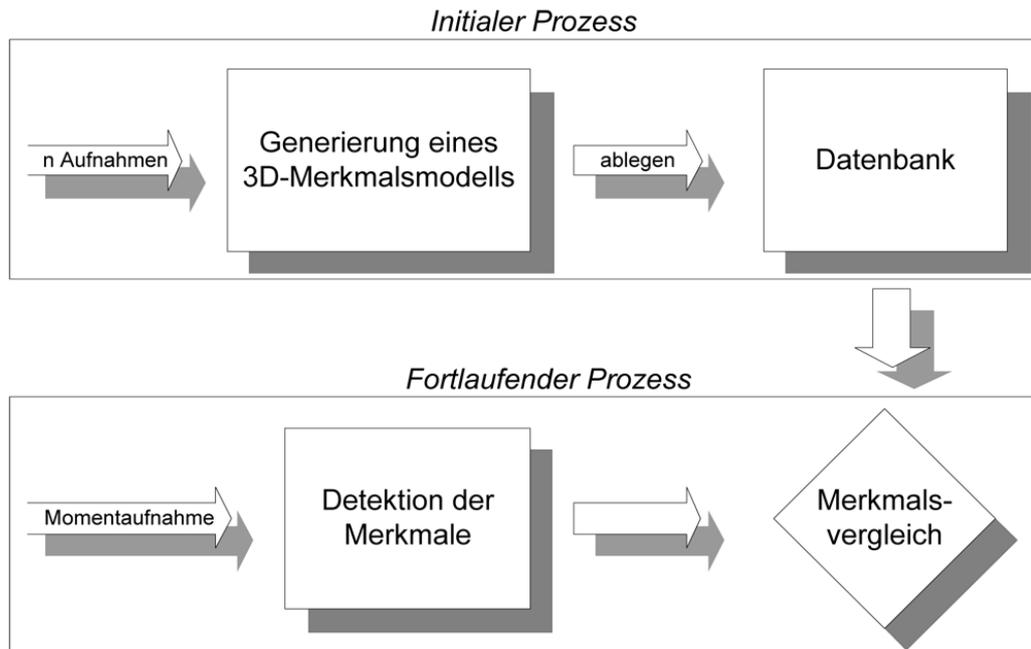
## 7.2 Startposition anfahren

Eine festgelegte Position und Ausrichtung des Knickarm-Roboters, aus der die Roboterkamera einen weitflächigen Blick auf seine greifbare Umgebung verfügt, ist initial anzufahren. Nachdem sich der Roboter in der Startposition befindet, wird eine erste Aufnahme der Umgebung aufgenommen.

## 7.3 Identifizierung

### a) Erkennung aller greifbaren Gegenstände

Die erste Möglichkeit besteht darin sich zuerst alle im Kamerabild erkannten Gegenstände anzeigen zu lassen. Dafür ist die aktuelle Aufnahme einer Merkmalsdetektion (Vgl. Kapitel 5.2) zu unterziehen. Anschließend führt ein Merkmalsvergleich mit allen 3D-Merkmalmodellen zu den in der Aufnahme enthaltenen Gegenständen. Dabei ist ein Merkmalsmodell einem Gegenstand zugeordnet und beinhaltet u. a. gegenstands-spezifische Merkmale.



**Abbildung 7.2:** Ablauf einer Objektidentifikation auf Basis von Merkmalen.

Der Merkmalsvergleich erfolgt nach Kapitel 5.3.1 (*Suche nach Merkmalskorrespondenzen*) und ist in der Abbildung 7.2 visualisiert. Aus der Korrespondenzmenge kann anschließend bestimmt werden, ob sich ein Gegenstand in einem Kamerabild befindet. Existieren mindestens drei Merkmalsübereinstimmungen<sup>33</sup> für einen Gegenstand, ist dieser mit einer sehr großen Wahrscheinlichkeit in der Aufnahme abgebildet. Dies führt zu einer statistischen Auswertung der Form:

Merkmalmodell	Merkmale der Momentaufnahme	Anzahl Korrespondenzen	Korrespondenzen [%]
Gegenstand i			
⋮			
Gegenstand j			

**Tabelle 7.1:** Statistische Form der Korrespondenzanalyse.

Die identifizierten Gegenstände erlauben dem Anwender die Auswahl eines zu greifenden Gegenstandes. Die Abbildung 7.3 veranschaulicht die Darstellung von potentiell greifbaren Gegenständen.

<sup>33</sup>Der genannte Wert gilt bei einem Schwellwert (*Threshold*) von 0,1. Dieser ist ein Maß für die Einzigartigkeit eines Merkmales und bedeutet Folgendes. Eine Merkmalskorrespondenz ist nur dann vorhanden, wenn das Verhältnis vom kleinsten- zum zweit-kleinsten euklidischen Abstand nicht den Schwellwert übersteigt. Dies gilt Analog wenn das Skalarprodukt zwischen Einheitsvektoren anstelle vom euklidischen Abstand herangezogen wird.



Abbildung 7.3: Identifizierte Gegenstände in einem Kamerabild.

#### b) Erkennung eines bestimmten Gegenstandes

Alternativ kann ein ausgewählter Gegenstand gesucht werden. Befindet sich der gesuchte Gegenstand nicht in der aktuellen Aufnahme, bewegt sich der Roboterarm in einer festgelegten Trajektorie. Während der Knickarm-Roboter die Trajektorie aus der Abbildung 7.4 abfährt, ist in kontinuierlichen Abständen eine Momentaufnahme aufzunehmen. Jede Aufnahme wird bezüglich des gesuchten Gegenstandes analysiert. Für die Analyse eines Bildes erfolgt wie bei (a) eine Merkmalsdetektion mit anschließendem Merkmalsvergleich mit dem gegenstands-spezifischen Merkmalsmodell.

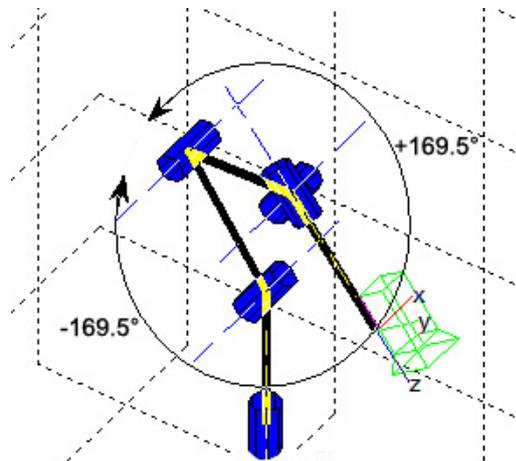


Abbildung 7.4: Trajektorie ausgehend der Startausrichtung.

## 7.4 Berechnung der Zielpose abhängig zur Objektpose

Im vorherigen Unterkapitel wurde der zu greifende Gegenstand ausgewählt. Für den Greifvorgang muss zu Beginn die Zielpose  $T_{base}^{targeteff}$  abhängig zur aktuellen Objektpose berechnet werden. Dabei kennzeichnet  $T_{base}^{targeteff}$  die Transformation vom Basis- zum Endeffektorkoordinatensystem der Zielpose. Erst wenn diese Abbildung bekannt ist, kann anschließend die Greifpose  $T_{base}^{grieffeff}$  ermittelt werden. Dieses Unterkapitel beschreibt, wie auf Basis der initial festgelegte Zielpose  $T_{obj}^{targetcaminit}$ , die gesuchte Zielpose  $T_{base}^{targeteff}$  zu berechnen ist.

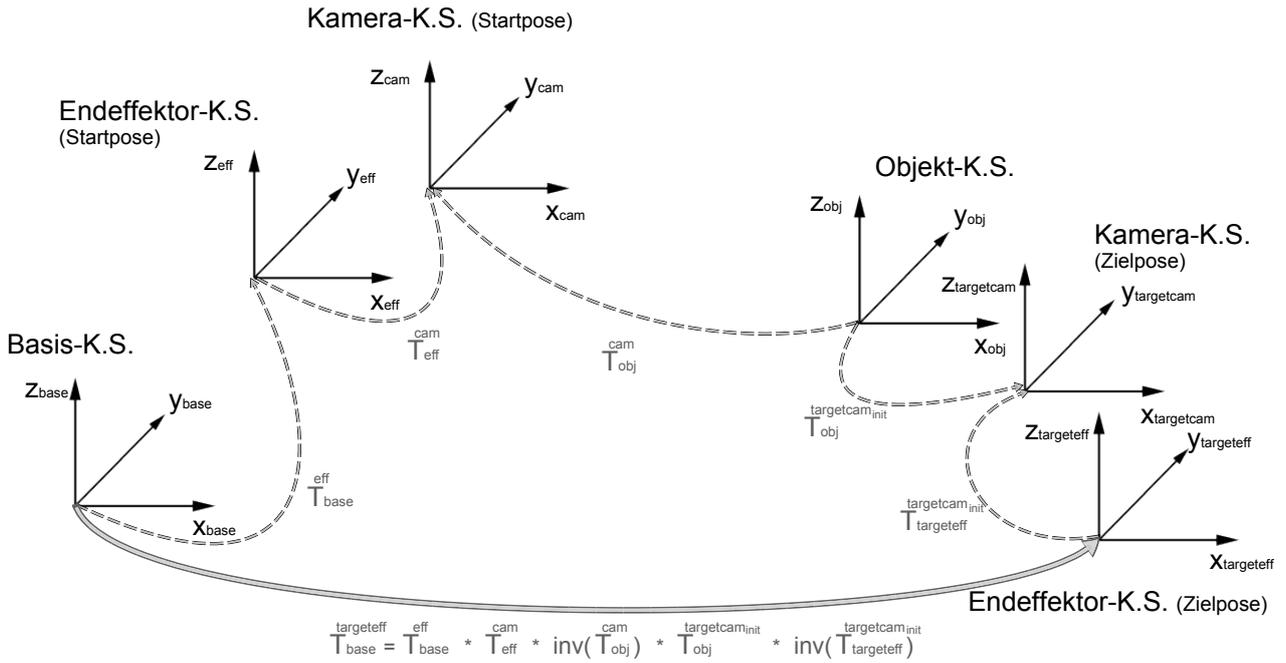


Abbildung 7.5: Transformation zwischen Koordinatensystemen (K.S.).

Gesucht sei die  $3 \times 4$  Transformationsmatrix  $\mathbf{T}_{base}^{targeteff}$ , welche Koordinaten vom Basiskoordinatensystem in das Koordinatensystem der Zielpose abbildet. An dieser Stelle sei auf das Grundlagenkapitel 3.1 verwiesen. Dieses beschreibt die in dieser Arbeit vorkommenden Koordinatensysteme und damit auch das Basiskoordinatensystem, das sich im FuÙe des Knickarm-Roboters befindet.

Die Transformation  $\mathbf{T}_{base}^{targeteff}$  lässt sich durch die nachfolgenden fünf Einzeltransformationen  $\mathbf{T}_{base}^{eff}$ ,  $\mathbf{T}_{eff}^{cam}$ ,  $\mathbf{T}_{cam}^{obj}$ ,  $\mathbf{T}_{obj}^{targetcam_{init}}$  und  $\mathbf{T}_{targetcam_{init}}^{targeteff}$  berechnen (Vgl. Grafik 7.5). Diese werden in den nachfolgenden fünf Sektionen (7.4a) bis (7.4e) hergeleitet.

a) Basis-  $\mapsto$  Endeffektorkoordinatensystem  $\mathbf{T}_{base}^{eff}$

Zu Beginn wird die in der Grafik 7.6 visualisierte Transformationsmatrix  $\mathbf{T}_{base}^{eff}$  gesucht. Diese Transformation vom Basis- zum Endeffektorkoordinatensystem kann mit der sogenannten Vorwärtskinematik berechnet werden. Des Weiteren ist diese Abbildung auch als direkte Kinematik bekannt. Für die Berechnung sind die aktuellen Gelenkwinkel und die Armlängen (Gelenkabstände) notwendig.

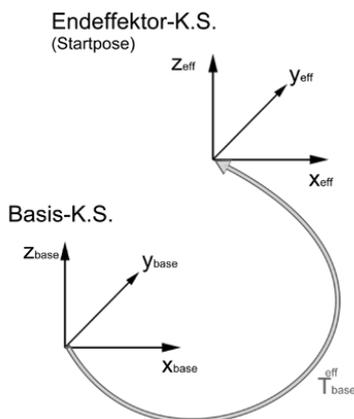


Abbildung 7.6: Basis-  $\mapsto$  Endeffektorkoordinatensystem  $\mathbf{T}_{base}^{eff}$ .

Die Armlängen des Knickarm-Roboters

$$l_{1,2} = 201.5 \quad (7.3)$$

$$l_{2,3} = 190.0 \quad (7.4)$$

$$l_{3,4} = 139.0 \quad (7.5)$$

$$l_{4,5,*} = 188.3 \quad (7.6)$$

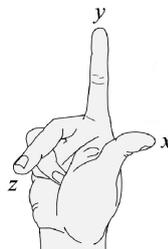
sind im Gegensatz zu den fünf Gelenkwinkeln  $\varphi_i$  unveränderbare Größen. Sie sind in der Maßeinheit *mm* angegeben und in der Abbildung 4.1 auf Seite 19 veranschaulicht. Die ersten drei Armlängen  $l_{1,2}$ ,  $l_{2,3}$  und  $l_{3,4}$  beschreiben jeweils die Länge zwischen zwei Gelenken. Dagegen kennzeichnet die Längenangabe  $l_{4,5,*}$  den Abstand vom vierten Gelenk bis zum Flansch (Siehe Kapitel 4.1).

In dieser Arbeit wird für die Berechnung der Vorwärtskinematik die sogenannte Denavit-Hartenberg-Konvention herangezogen [STA09, HD64, DH55]. Sie ist eine Vorschrift für die schrittweise Festlegung der (Gelenk-) Koordinatensysteme:

- i) Die  $z_n$ -Achse verläuft entlang der Bewegungsachse des  $n$ -ten Gelenks
- ii) Die  $x_n$ -Achse steht orthonormal auf der  $z_{n-1}$ -Achse und zeigt von ihr weg
- iii) Die Anordnung der  $y_n$ -Achse führt zu einem rechtshändigen Koordinatensystem

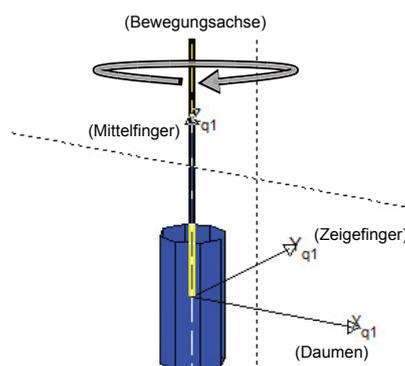
Für das erste Koordinatensystem (Basiskoordinatensystem) entfällt die zweite Vorschriftsregel (ii). Die  $x$ - und  $y$ -Achse sind dort frei wählbar. Allerdings muss ein rechtshändiges Koordinatensystem resultieren.

Bei einem rechtwinkligen rechtshändigen Koordinatensystem definiert der Daumen die  $x$ -Achse, der Zeigefinger gleicht der  $y$ -Achse und der Mittelfinger entspricht der  $z$ -Achse. Dabei bezieht sich die Achsenwahl auf die in Abbildung 7.7 ausgerichteten Finger einer rechten Hand. Diese Anwendung der Achsenzuordnung ist als Rechte-Hand-Regel bekannt.



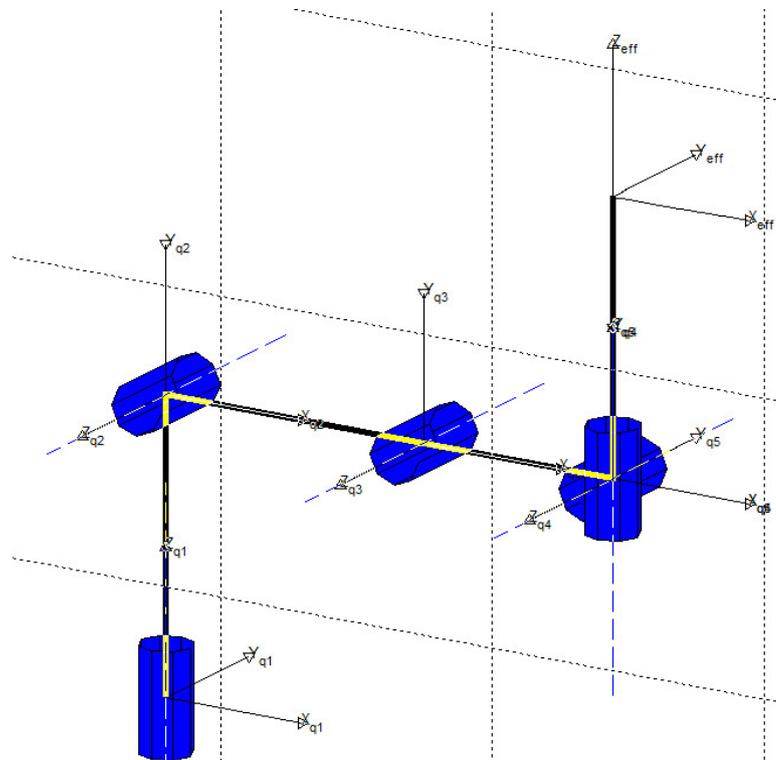
**Abbildung 7.7:** Rechte-Hand-Regel. [ROD04]

Das erste (Gelenk-) Koordinatensystem entspricht dem Basiskoordinatensystem und ist nachfolgend dargestellt.



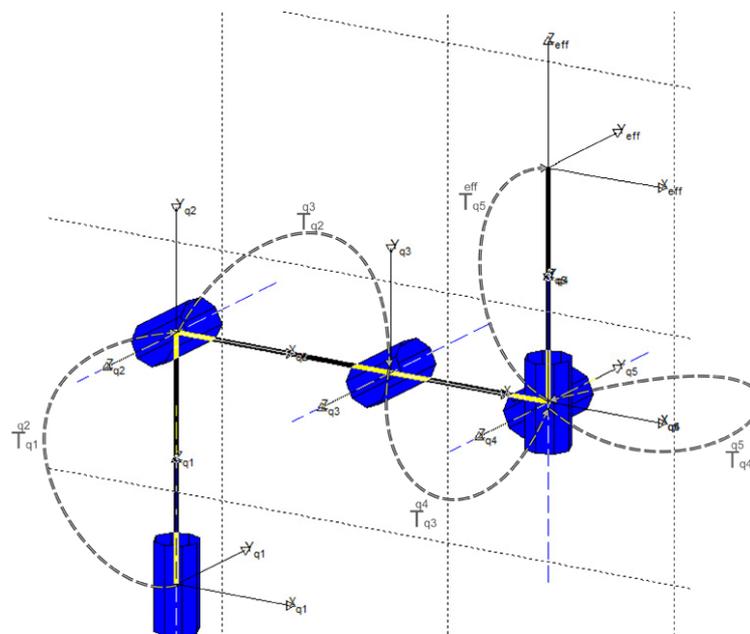
**Abbildung 7.8:** Koordinatensystem nach der Denavit-Hartenberg-Konvention.

Der Ursprung befindet sich im ersten Gelenk des Knickarm-Roboters. Die  $z$ -Achse liegt in der Drehachse des ersten Gelenkes  $q_1$ . Die  $x$ - und  $y$ -Achse werden so gewählt, dass ein rechtshändiges Koordinatensystem vorliegt. Die restlichen Koordinatensysteme sind entsprechend weiter nach der Denavit-Hartenberg-Konvention zu definieren. Die Abbildung 7.9 zeigt die in dieser Arbeit festgelegten (Gelenk-) Koordinatensysteme.



**Abbildung 7.9:** Festlegung der (Gelenk-) Koordinatensysteme.

Die Beziehungen zwischen den (Gelenk-) Koordinatensystemen bilden die fünf homogenen Transformationsmatrizen  $\mathbf{T}_{q_1}^{q_2}$  bzw.  $\mathbf{T}_{\text{base}}^{q_2}$ ,  $\mathbf{T}_{q_2}^{q_3}$ ,  $\mathbf{T}_{q_3}^{q_4}$ ,  $\mathbf{T}_{q_4}^{q_5}$  und  $\mathbf{T}_{q_5}^{\text{eff}}$  ab (Vgl. Abbildung 7.10).



**Abbildung 7.10:** Homogene Transformationen zwischen Koordinatensystemen.

Eine Transformation zwischen zwei Koordinatensystemen bzw. deren Lage und Orientierung zueinander wird im Allgemeinen mit sechs Parametern  $[x, y, z, \phi, \psi, \vartheta]$  (RPY-Winkelkonvention) bestimmt. Die so-

nannte Denavit-Hartenberg-Transformation ermöglicht diese Berechnung mit lediglich vier Parametern  $\alpha$ ,  $a$ ,  $d$  und  $\theta$ . Diese werden als sogenannte Denavit-Hartenberg-Parameter bezeichnet. [STA09]

Denavit-Hartenberg-Transformationen beschreiben die kinematische Kette eines Roboters und damit die räumlichen Beziehungen zwischen den (Gelenk-) Koordinatensystemen. Eine kinematische Kette besteht aus den fünf homogenen Transformationsmatrizen  $\mathbf{T}_{q_1}^{q_2}(\alpha_1, a_1, d_1, \theta_1)$  bzw.  $\mathbf{T}_{\text{base}}^{q_2}(\alpha_1, a_1, d_1, \theta_1)$ ,  $\mathbf{T}_{q_2}^{q_3}(\alpha_2, a_2, d_2, \theta_2)$ ,  $\mathbf{T}_{q_3}^{q_4}(\alpha_3, a_3, d_3, \theta_3)$ ,  $\mathbf{T}_{q_4}^{q_5}(\alpha_4, a_4, d_4, \theta_4)$  und  $\mathbf{T}_{q_5}^{\text{eff}}(\alpha_5, a_5, d_5, \theta_5)$  und wird ausgehend vom ersten Koordinatensystem erzeugt.  $\theta_i$  gibt dabei den Drehwinkel des Gelenkes  $i$  an.

Jede Denavit-Hartenberg-Transformation  $\mathbf{T}_{q_i}^{q_{i+1}}(\alpha_i, a_i, d_i, \theta_i)$  mit  $i \in \{1, 2, 3, 4\}$  und  $\mathbf{T}_{q_5}^{\text{eff}}(\alpha_5, a_5, d_5, \theta_5)$  besteht aus vier elementaren Transformationen,

- Rotation um die  $z$ -Achse mit dem Winkel  $\theta$

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.7)$$

- Translation der Länge  $d$  entlang der  $z$ -Achse

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.8)$$

- Translation der Länge  $a$  entlang der  $x$ -Achse

$$\begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.9)$$

- Rotation um die  $x$ -Achse mit dem Winkel  $\alpha$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.10)$$

welche von den vier Parametern  $\alpha$ ,  $a$ ,  $d$  und  $\theta$  abhängig sind. Bei einem Lineargelenk ist der Abstand  $d_i$  variabel und somit würde  $d_i$  auf den aktuellen Winkel  $\varphi_i$  des Gelenkes  $i$  gesetzt werden. Liegt dagegen ein Rotationsgelenk vor, ist der Winkel  $\theta_i$  variabel.  $\theta_i$  entspricht dann  $\varphi_i$ . In dieser Arbeit ist dies

$$\theta_i = \varphi_i \quad (7.11)$$

der Fall.

Für die Berechnung der Denavit-Hartenberg-Transformationen  $\mathbf{T}_{q_i}^{q_{i+1}}$  mit  $i \in \{1, 2, 3, 4\}$  und  $\mathbf{T}_{q_5}^{\text{eff}}$  sind die vier elementaren Transformationen 7.7 bis 7.10 ausgehend von 7.7 miteinander zu multiplizieren. Es ergibt sich für  $\mathbf{T}_{q_i}^{q_{i+1}}(\alpha_i, a_i, d_i, \theta_i)$  mit  $i \in \{1, 2, 3, 4\}$  und  $\mathbf{T}_{q_5}^{\text{eff}}(\alpha_5, a_5, d_5, \theta_5)$  die Gleichung 7.12. Dabei

kennzeichnet die allgemeine Notation  $\mathbf{T}_i^{i+1}(\alpha_i, a_i, d_i, \theta_i)$  die Abbildung von dem Koordinatensystem  $i$  zum Koordinatensystem  $i + 1$  unter Verwendung der transformations-bezogenen Parameter  $\alpha_i, a_i, d_i, \theta_i$ .

$$\mathbf{T}_i^{i+1}(\alpha_i, a_i, d_i, \theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cdot \cos(\alpha_i) & \sin(\theta_i) \cdot \sin(\alpha_i) & a_i \cdot \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cdot \cos(\alpha_i) & -\cos(\theta_i) \cdot \sin(\alpha_i) & a_i \cdot \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.12)$$

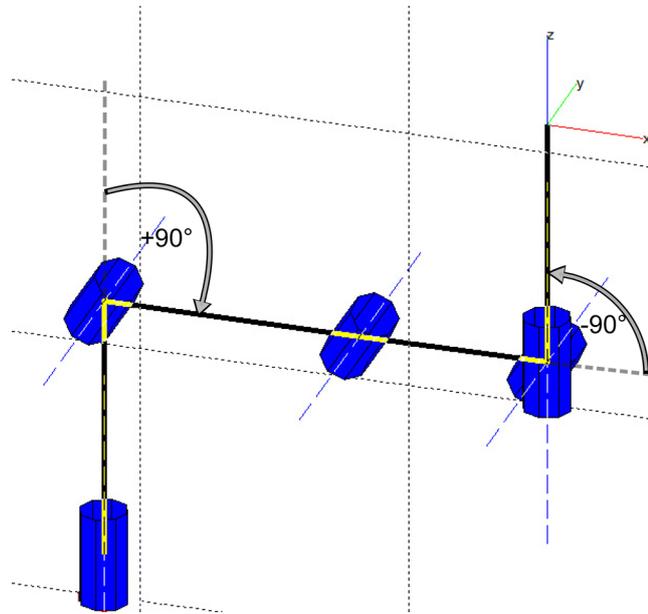
Die Gleichung 7.12 darf für die umgekehrte Abbildungsrichtung invertiert werden (Vgl. Gleichung 7.13).

$$\mathbf{T}_{i+1}^i(\alpha_i, a_i, d_i, \theta_i) = \text{inv}(\mathbf{T}_i^{i+1}(\alpha_i, a_i, d_i, \theta_i)) \quad (7.13)$$

Die Tabelle 7.2 beinhaltet die Parameter für die fünf Denavit-Hartenberg-Transformationen  $\mathbf{T}_{q_1}^{q_2}$ ,  $\mathbf{T}_{q_2}^{q_3}$ ,  $\mathbf{T}_{q_3}^{q_4}$ ,  $\mathbf{T}_{q_4}^{q_5}$  und  $\mathbf{T}_{q_5}^{\text{eff}}$ . Sie sind entsprechend der Abbildung 7.9 angegeben. Unter den Parametern kennzeichnet  $a_i$  den Abstand von  $z_{i-1}$  zu  $z_i$  entlang  $x_{i-1}$ . Der Winkel  $\alpha_i$  zwischen der  $z_{i-1}$ - und  $z_i$ -Achse wird um  $x_{i-1}$  gemessen. Dabei entspricht  $\pm \frac{\pi}{2}$  rad einem Winkel von  $\pm 90^\circ$  (Vgl. Abbildung 7.11). Der Abstand  $d_i$  von  $x_{i-1}$  zu  $x_i$  ist entlang der  $z_i$ -Achse zu messen. Der vierte Parameter ist  $\theta_i$ . Er definiert den Winkel zwischen  $x_{i-1}$  und  $x_i$  gemessen um die  $z_i$ -Achse.

Gelenk	Achswinkel $\alpha_i$	Achsabstand $a_i$	Achswinkel $\theta_i (= \varphi_i)$	Achsabstand $d_i$
1	$+\frac{\pi}{2}$	0	$\varphi_1$	$l_{1,2}$
2	0	$l_{2,3}$	$\varphi_2$	0
3	0	$l_{3,4}$	$\varphi_3$	0
4	$-\frac{\pi}{2}$	0	$\varphi_4$	0
5	0	0	$\varphi_5$	$l_{4,5,*}$

**Tabelle 7.2:** Denavit-Hartenberg-Parameter.



**Abbildung 7.11:** Kinematische Grundstellung definiert durch Denavit-Hartenberg-Parameter.

Zusammenfassend kann die gesuchte Abbildung  $\mathbf{T}_{\text{base}}^{\text{eff}}$  durch die kinematischen Kette

$$\mathbf{T}_{\text{base}}^{\text{eff}} = \mathbf{T}_{q_1}^{\text{eff}} = \mathbf{T}_{q_1}^{q_2} \cdot \mathbf{T}_{q_2}^{q_3} \cdot \mathbf{T}_{q_3}^{q_4} \cdot \mathbf{T}_{q_4}^{q_5} \cdot \mathbf{T}_{q_5}^{\text{eff}} = \left( \prod_{i=1}^4 \mathbf{T}_{q_i}^{q_{i+1}}(\alpha_i, a_i, d_i, \theta_i) \right) \cdot \mathbf{T}_{q_5}^{\text{eff}}(\alpha_5, a_5, d_5, \theta_5) \quad (7.14)$$

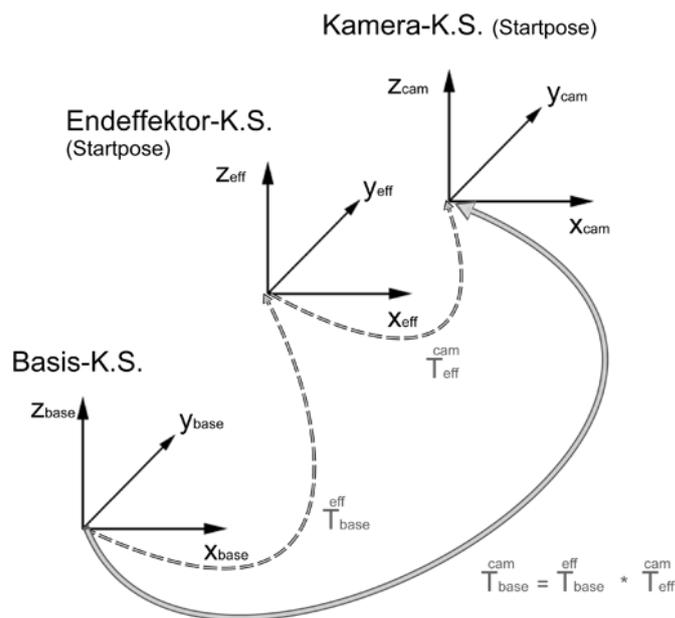
berechnet werden. Die Denavit-Hartenberg-Transformationen  $\mathbf{T}_{q_{i+1}}^{q_i}$  mit  $i \in \{1, 2, 3, 4\}$  und  $\mathbf{T}_{q_5}^{\text{eff}}$  beinhalten die vier festgelegten achsenspezifischen Denavit-Hartenberg-Parameter. Der Winkel  $\varphi_i$  entspricht  $\theta_i$  und ist der zugehörige Gelenkwinkel.

b) Endeffektor-  $\mapsto$  Kamerakoordinatensystem  $\mathbf{T}_{\text{eff}}^{\text{cam}}$

Die Transformation vom Endeffektor- zum Kamerakoordinatensystem der Startpose (Vgl. Grafik 7.12) wird durch eine einmalige Kalibrierung von  $\mathbf{T}_{\text{eff}}^{\text{cam}}$  ermittelt. Mit der Startpose ist die aktuelle Position und Ausrichtung der Roboterkamera gemeint. Die Kalibrierung wird als sogenannte Hand-Auge-Kalibrierung bezeichnet und wurde in dieser Arbeit einmalig vorab der Objektmobilisierung durchgeführt. Die Hand ist der Endeffektor, das Auge ist dementsprechend die Kamera. Die Kalibrierungsart unterscheidet sich leicht von der beschriebenen Kamerakalibrierung aus Kapitel 5.1.

Die angewandte Kalibrierungsmethode stammt ebenfalls von [BOU08] und benötigt gleichermaßen Kameraaufnahmen, welche aus unterschiedlichen Perspektiven ein bekanntes Schachbrettmuster aufnehmen. Damit jedoch eine räumliche Beziehung zwischen Kamera und Endeffektor ermittelt werden kann, ist für jede Aufnahme  $\mathbf{B}_i$  die Transformationsmatrix  $\mathbf{T}_{\text{base}}^{\text{eff}_i}$  zwischen zu speichern. Aus sämtlichen Kameraaufnahmen wird ein Schachbrettmuster dedektiert und die Kamerapose  $\mathbf{T}_{\text{base}}^{\text{cam}_i}$  zurückberechnet. Anhand der Zuordnung  $\mathbf{T}_{\text{base}}^{\text{cam}_i} \longleftrightarrow \mathbf{T}_{\text{base}}^{\text{eff}_i}$  kann die gesuchte Transformationsmatrix  $\mathbf{T}_{\text{eff}}^{\text{cam}}$  berechnet werden.

Für weitere Informationen bezüglich der Hand-Auge-Kalibrierung sei auf [BOU08] verwiesen.

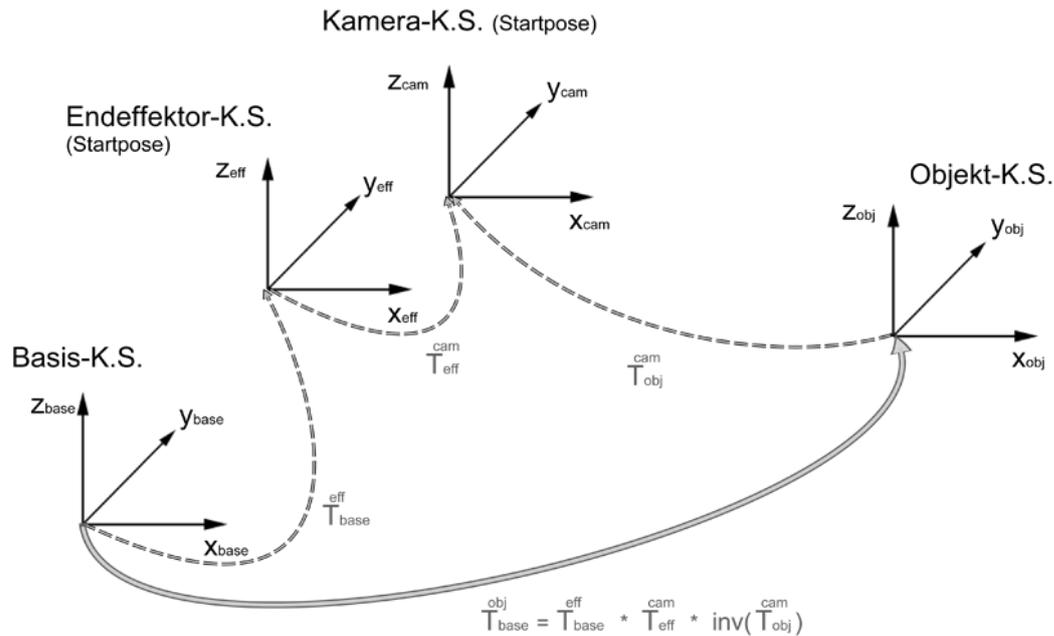


**Abbildung 7.12:** Transformation zum Kamerakoordinatensystem der Roboterkamera.

Abschließend kann die Abbildung von Koordinaten des Basis- auf das Kamerakoordinatensystem beschrieben werden. Die nachfolgende Gleichung 7.15 beschreibt diese Transformation.

$$\mathbf{T}_{\text{base}}^{\text{cam}} = \mathbf{T}_{\text{base}}^{\text{eff}} \cdot \mathbf{T}_{\text{eff}}^{\text{cam}} \quad (7.15)$$

c) Objekt-  $\mapsto$  Kamerakoordinatensystem  $\mathbf{T}_{obj}^{cam}$



**Abbildung 7.13:** Objekt-  $\mapsto$  Kamerakoordinatensystem  $\mathbf{T}_{obj}^{cam}$ .

Die relative Posenbestimmung des zu greifenden Gegenstandes zum Kamerakoordinatensystem erfolgt nach Kapitel 6 (*Posenbestimmung*) mit den gefundenen 3D-Welt-zu-2D-Bildkorrespondenzen. Dabei stammen die detektierten Merkmals- bzw. Bildkorrespondenzen aus der Phase 7.3 (*Identifizierung*) und wurden zwischen der aktuellen Aufnahme und dem Merkmalsmodell des Gegenstandes ermittelt.

Resultierend lässt sich die gesuchte Transformationsmatrix

$$\mathbf{T}_{obj}^{cam} = \text{Objekt-} \mapsto \text{Kamerakoordinaten} \quad (7.16)$$

bestimmen. Aus dieser folgt die Translation und Rotation zwischen der Kameraaufnahme und dem im Raum positionierten Gegenstand (Vgl. Grafik 7.13).

Mit Hilfe der Abbildungsmöglichkeit zwischen Koordinaten des Objekt- und Kamerakoordinatensystems kann die Transformation  $\mathbf{T}_{base}^{cam}$  vom Basis- zum Kamerakoordinatensystem ergänzt werden. Dafür ist  $\mathbf{T}_{base}^{cam}$  mit der invertierten Abbildungsmatrix  $\mathbf{T}_{obj}^{cam}$  zu multiplizieren. Dabei wird  $\mathbf{T}_{obj}^{cam}$  aufgrund der Abbildungsrichtung invertiert. Als Ergebnis folgt

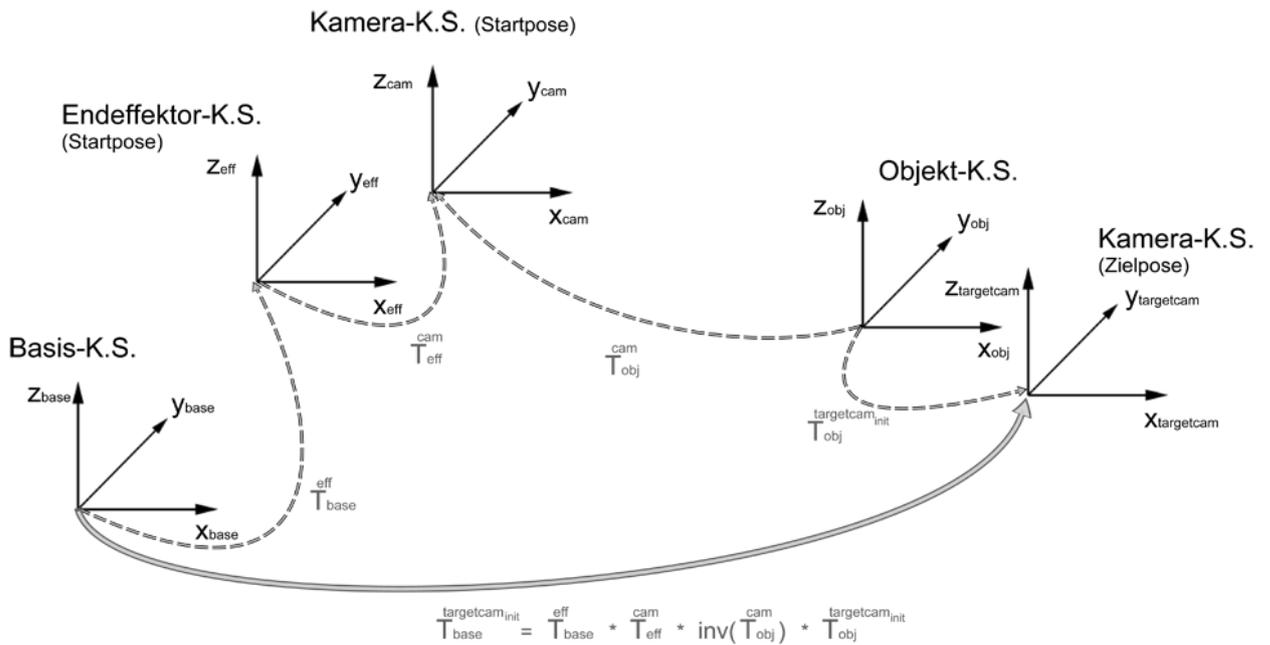
$$\mathbf{T}_{base}^{obj} = \mathbf{T}_{base}^{cam} \cdot \text{inv}(\mathbf{T}_{obj}^{cam}) \quad (7.17)$$

d) Objekt-  $\mapsto$  Kamerakoordinatensystem der Zielpose  $\mathbf{T}_{obj}^{\text{targetcam}_{init}}$

In der Phase 7.1 (*Vorbedingungen*) wurde erwähnt, dass eine Transformation  $\mathbf{T}_{obj}^{\text{targetcam}_{init}}$  benötigt wird, welche Koordinaten vom Objekt- zum Kamerakoordinatensystem der Zielpose abbildet. Eine Zielansicht wurde dafür in der Vorbereitungsphase aufgenommen, deren Merkmale mit den Merkmalsmodell des Gegenstandes verglichen und aus den Korrespondenzen die Transformationsmatrix

$$\mathbf{T}_{obj}^{\text{targetcam}_{init}} = \text{Objekt-} \mapsto \text{Kamerakoordinatensystem der Zielpose} \quad (7.18)$$

berechnet.



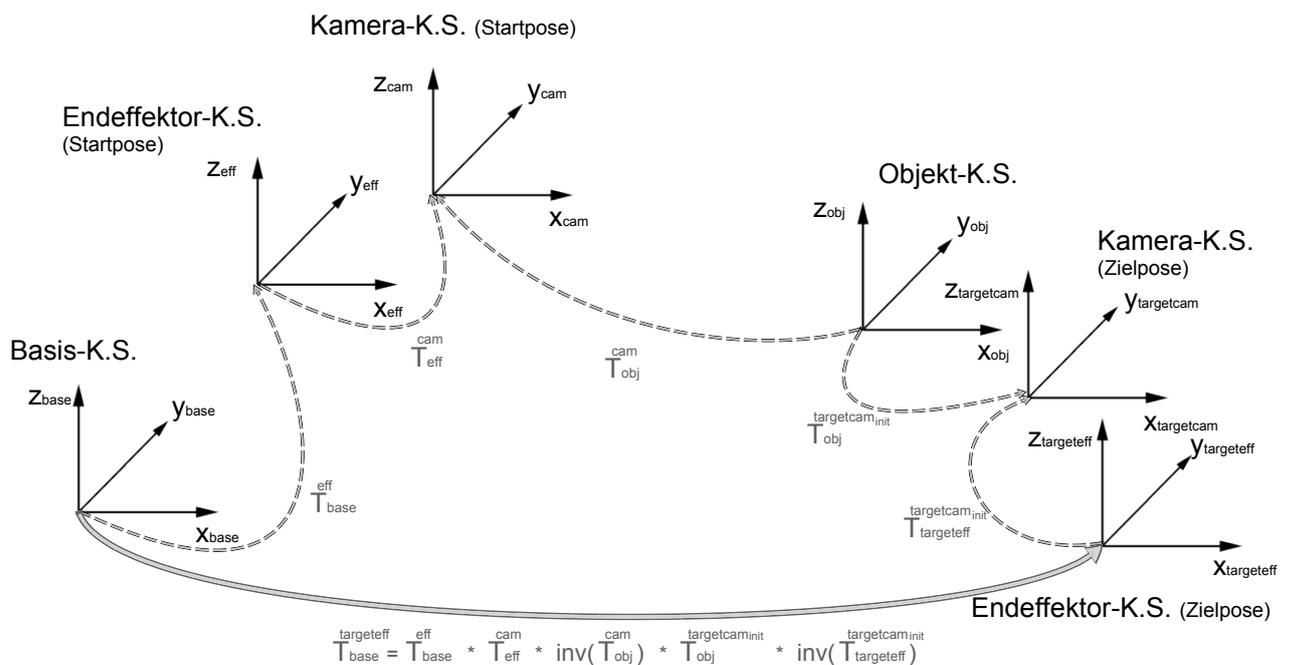
**Abbildung 7.14:** Visualisierung der Koordinatensystem (K.S.) - Transformationen.

Zusammen mit dem Wissen über die bekannte Objektlage und -ausrichtung aus der Sektion (8.4c) kann die Abbildung

$$\mathbf{T}_{\text{base}}^{\text{targetcam}_{\text{init}}} = \mathbf{T}_{\text{base}}^{\text{obj}} \cdot \mathbf{T}_{\text{obj}}^{\text{targetcam}_{\text{init}}} \quad (7.19)$$

vom Basis- auf das Kamerakoordinatensystem der Zielpose beschrieben werden (Vgl. Abbildung 7.14).

e) Endeffektor-  $\rightarrow$  Kamerakoordinatensystem der Zielpose  $\mathbf{T}_{\text{targeteff}}^{\text{targetcam}_{\text{init}}}$



**Abbildung 7.15:** Darstellung der Koordinatensystem (K.S.) - Transformationen.

Die Transformation vom Kamera- zum Endeffektorkoordinatensystem der Zielpose muss für die gesuchte Transformation  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  berücksichtigt werden. Aus diesem Grund wird die relative Pose zwischen

Kamera- und Endeffektorkoordinatensystem der Zielpose benötigt. Die Invertierung von  $\mathbf{T}_{\text{eff}}^{\text{cam}}$  beschreibt genau diese Abbildung und führt zu der gesuchten Transformation

$$\mathbf{T}_{\text{base}}^{\text{targeteff}} = \mathbf{T}_{q_1}^{\text{targeteff}} = \mathbf{T}_{\text{base}}^{\text{targetcam}_{\text{init}}} \cdot \text{inv} \left( \mathbf{T}_{\text{targeteff}}^{\text{targetcam}_{\text{init}}} \right) \quad (7.20)$$

In der Gleichung 7.20 ist  $\mathbf{T}_{\text{targeteff}}^{\text{targetcam}_{\text{init}}}$  mit  $\mathbf{T}_{\text{eff}}^{\text{cam}}$  aus der Sektion (7.4b) (*Endeffektor-  $\rightarrow$  Kamerakoordinatensystem der Startpose*) gleichgestellt.

## 7.5 Korrektur der Zielpose

Jeder Gegenstand ist individuell in Abhängigkeit seiner Form und seinem Material zu greifen. Die Zielpose  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  ist eine Vorstufe der Greifpose  $\mathbf{T}_{\text{base}}^{\text{grieff}}$ , welche möglichst exakt berechnet werden sollte. Wenn die Zielpose fehlerbehaftet bestimmt wird, folgt eine ungenaue Berechnung der Greifpose. Denn in diesem Moment wird davon ausgegangen, dass sich der Roboter ausgehend von der aktuellen Objektpose in der initial festgelegten Pose  $\mathbf{T}_{\text{obj}}^{\text{targetcam}_{\text{init}}}$  bzw.  $\mathbf{T}_{\text{obj}}^{\text{targeteff}}$  befindet. Aus diesem Grund ist es möglich die Zielpose beliebig oft zu berechnen bzw. zu korrigieren bevor die Greifpose ermittelt wird.

Die nachfolgende Abbildung 7.16 stellt die beschriebene Situation dar. Die erste Grafik (a) zeigt die initial festgelegte Pose  $\mathbf{T}_{\text{obj}}^{\text{targetcam}_{\text{init}}}$  zwischen der Objekt- und Zielpose. In der zweiten Grafik (b) wird der Gegenstand im Kamerabild erkannt, die Lage und Orientierung des Gegenstandes berechnet und anschließend die Zielpose abhängig zur neuen Objektpose ermittelt. Anschließend kann die berechnete Zielpose  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  angefahren werden (Grafik (c)). Es ist gut zu erkennen, dass die berechnete Zielpose nicht mit der Gewünschten übereinstimmt. In diesem Fall würde die Greifpose fehlerbehaftet berechnet werden. Aus diesem Grund eignet sich an dieser Stelle eine erneute Berechnung der Zielpose (d). Dafür ist in der neuen Roboterlage (c) ein Kamerabild aufzunehmen und auf Merkmale zu untersuchen. Anschließend beginnt erneut die Berechnung der Zielpose relativ zur Objektpose (Vgl. Kapitel 7.4, *Berechnung der Zielpose abhängig zur Objektpose*), jedoch mit einer räumlich näheren Ausgangslage was eine exaktere Posenbestimmung zur Folge hat.

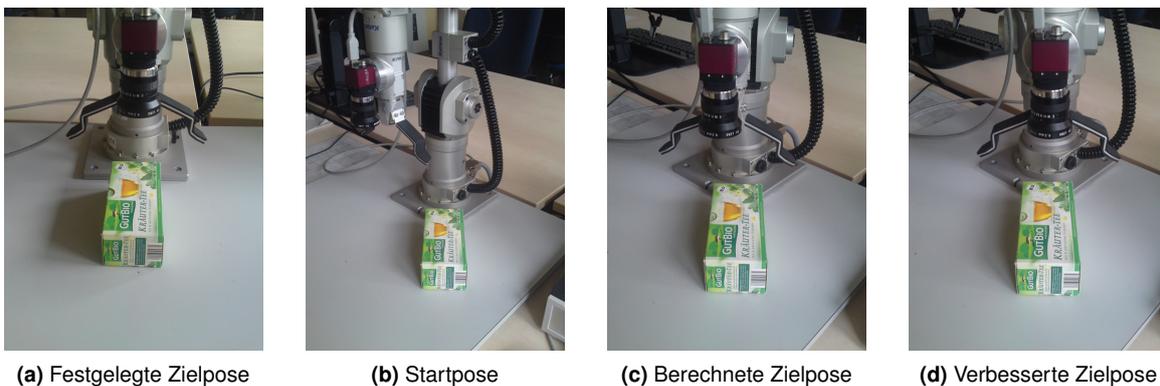


Abbildung 7.16: Erneute Berechnung der Zielpose.

## 7.6 Inverse Kinematik der Zielpose

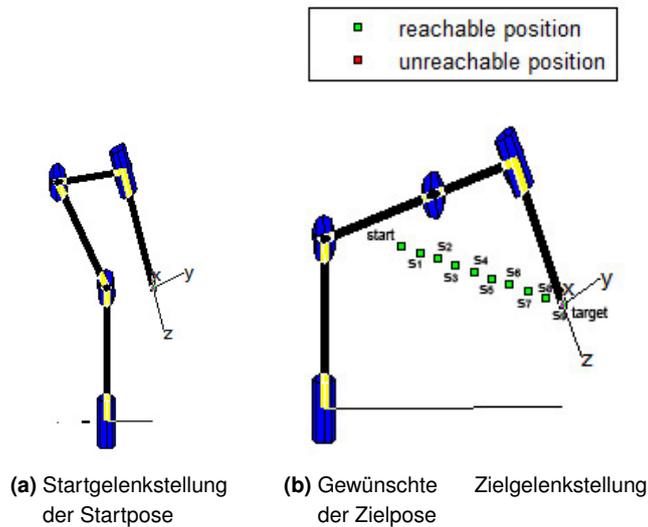
Die  $3 \times 4$  Transformationsmatrix  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  beschreibt die Abbildung vom Basis- auf das Endeffektorkoordinatensystem der Zielpose. Jedoch wird ein Knickarm-Roboter nicht über eine Raumpose, sondern über die Gelenkwinkel gesteuert.

Wie die fünf Gelenkwinkel  $\varphi_i$  in der Startgelenkstellung  $\varphi_{\text{cur}_i}$  eingestellt werden müssen, damit sich der Endeffektor des Roboters in die gewünschte Zielgelenkstellung  $\varphi_i$  bewegt, ist bis hierhin unbekannt. Die sogenannte

inverse Kinematik berechnet aus der Transformationsmatrix  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  die gesuchten Gelenkwinkel der Zielpose (Gleichung 7.21).

$$\mathbf{T}_{\text{base}}^{\text{targeteff}} \mapsto \varphi_i \quad \text{für } i = 1, \dots, 5 \quad (7.21)$$

Sie ist damit das logische Gegenstück zur Vorwärtskinematik, welche in der Sektion (8.4a) beschrieben wurde. Allerdings existiert im Gegensatz zur Vorwärtstransformation kein allgemeingültiges Lösungskonzept. Weiterhin ist die Lösung gängigerweise nicht eindeutig. Denn die Gelenke können meistens unterschiedlich eingestellt werden, damit sich der Endeffektor in der Zielpose  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  befindet.



**Abbildung 7.17:** Trajektorie mit  $s$  Trajektorienritten.

In dieser Arbeit wurde eine iterative Berechnung der vier von insgesamt fünf Gelenkwinkel  $\varphi_i$  für  $i = 2, \dots, 5$  der Zielgelenkstellung entwickelt. Dabei ist es besonders wichtig, dass der räumliche Abstand zwischen der Start- und Zielgelenkstellung nicht zu groß ist. Aus diesem Grund wird eine Trajektorie zwischen der Start- und Zielpose berechnet und in  $s$  Trajektorienritte zerlegt (Abbildung 7.17). Dadurch ist ein kleiner Abstand zwischen der Start- und Teilzielpose gewährleistet, wodurch die inverse Kinematik optimal bestimmt werden kann. Dafür ist die inverse Kinematik  $s$  mal zu berechnen. Das erste Mal wird anhand der Start-, Teilzielpose  $s_1$  und der Startgelenkstellung die Teilzielgelenkstellung iterativ bestimmt. Anschließend wird die berechnete Lösung verwendet um die nächste Teilzielgelenkstellung, welche sich räumlich näher an der Zielgelenkstellung befindet, zu berechnen. Dies wird solange fortgeführt bis die gesuchte Zielgelenkstellung bekannt ist. Im Sonderfall, dass keine Lösung für eine Teilzielgelenkstellung gefunden wird, ist dieser Trajektorienritt zu überspringen.

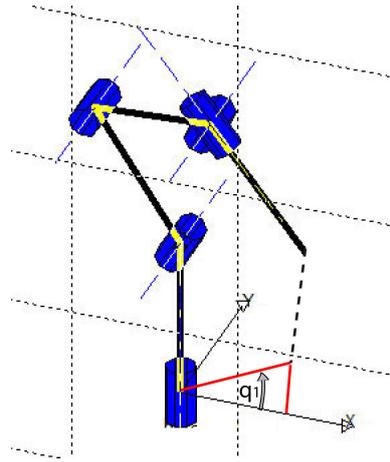
Die vier Gelenkwinkel  $\varphi_i$  für  $i = 2, \dots, 5$  der Zielgelenkstellungen sind lediglich iterativ zu berechnen. Das erste Gelenk kann unabhängig der Iteration und fehlerfrei bestimmt werden. Es handelt sich um ein Rotationsgelenk, welches nicht von der Ausrichtung des Endeffektors und nicht von der Endeffektorhöhe abhängt. Für den Rotationswinkel  $\varphi_1$  ist lediglich die  $x$ - und  $y$ -Position des Endeffektors relevant. Unter der Verwendung der Zielpose  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  und der trigonometrischen Tangensfunktion

$$\tan(\text{winkel}) = \frac{\text{Gegenkathete}}{\text{Ankathete}} \quad (7.22)$$

lässt sich der erste Gelenkwinkel der Zielgelenkstellung mit dem Arkustangens

$$\varphi_1 = \text{atan}(y, x) \quad (7.23)$$

direkt berechnen. Die Abbildung 7.18 veranschaulicht dies.



**Abbildung 7.18:** Direkte Berechnung des ersten Gelenks anhand der  $x$ - und  $y$ -Endeffektorkoordinate.

Die in dieser Arbeit angewandte iterative Berechnung der restlichen vier Gelenkwinkel  $\varphi_{t_2}, \dots, \varphi_{t_5}$  ist ein Minimierungsproblem der Form

$$\begin{aligned}
 \min_{\varphi_{cur_2}, \dots, \varphi_{cur_5}} \|F(\varphi_{t_1}, \varphi_{cur_2}, \dots, \varphi_{cur_5})\|_2^2 &= \left\| f_{kine}(\varphi_{t_1}, \varphi_{cur_2}, \dots, \varphi_{cur_5}) - \mathbf{T}_{base}^{targeteff} \right\|_2^2 \\
 &= \left\| \mathbf{T}_{base}^{eff}(\varphi_{t_1}, \varphi_{cur_2}, \dots, \varphi_{cur_5}) - \mathbf{T}_{base}^{targeteff} \right\|_2^2 \\
 &= \left\| [x_{diff}, y_{diff}, z_{diff}, \phi_{diff}, \psi_{diff}, \vartheta_{diff}] \right\|_2^2
 \end{aligned} \tag{7.24}$$

bei dem  $\left\| \mathbf{T}_{base}^{eff}(\varphi_{t_1}, \varphi_{cur_2}, \dots, \varphi_{cur_5}) - \mathbf{T}_{base}^{targeteff} \right\|_2^2$  einem nichtlinearen Gleichungssystem gleicht. Die Funktion  $f_{kine}$  entspricht der Vorwärtskinematik (Vgl. Sektion 7.4a, S. 92) und liefert Abhängig zur aktuell berechneten Gelenkstellung  $(\varphi_{t_1}, \varphi_{cur_2}, \dots, \varphi_{cur_5})$  die Raumposition und die Orientierung in RPY-Winkelkonvention.

Für die Minimierungsaufgabe aus der Gleichung 7.24 wurden drei Algorithmen evaluiert. Zum einen kann der Gauß-Newton-Algorithmus verwendet werden. Bei diesem Algorithmus muss die Funktion  $F(\varphi_{t_1}, \varphi_{cur_2}, \dots, \varphi_{cur_5})$  lediglich einmal nach den vier Funktionsparametern  $\varphi_{cur_2}, \dots, \varphi_{cur_5}$  (Partielle Ableitung) abgeleitet werden. Jedoch benötigt dieser viel Zeit, wenn das Residuum  $r = \left\| [x_{diff}, y_{diff}, z_{diff}, \phi_{diff}, \psi_{diff}, \vartheta_{diff}] \right\|_2^2$  von der Gleichung 7.24 sehr groß ist. Alternativ existiert der Levenberg-Marquardt-Algorithmus, welcher deutlich robuster ist und somit ein besseres Konvergenzverhalten zeigt. Dieser basiert auf einer Taylorentwicklung erster Ordnung und definiert ein quadratisches Ersatzproblem. Beim dritten Algorithmus *Trust-Region-Reflective* wird die Lösungskorrektur anhand einer Taylorentwicklung zweiter Ordnung berechnet. Dieser besitzt einen großen Vorteil gegenüber den anderen zwei Verfahren. Der *Trust-Region-Reflective*-Algorithmus beachtet die Rotations- und Knickgrenzbereiche jedes zu berechnenden Gelenkes. Dadurch ist jede Lösung für  $\min_{\varphi_{cur_2}, \dots, \varphi_{cur_5}} \|F(\varphi_{t_1}, \varphi_{cur_2}, \dots, \varphi_{cur_5})\|_2^2$  mit dem Katana-Roboter realisierbar.

Detaillierte Informationen bezüglich des Gauß-Newton-Algorithmus sind aus [EMNW05] zu entnehmen. Für Informationen über das *Trust-Region-Reflective*-Verfahren sei auf [CL92] und [CL96] verwiesen.

Im nachfolgenden wird die Minimierungsaufgabe mit dem Levenberg-Marquardt-Algorithmus erläutert (Vgl. [MOR78]). In jedem Iterationsschritt ist  $F(\varphi_{t_1}, \varphi_{cur_2}, \dots, \varphi_{cur_5})$  durch eine Linearisierung zu ersetzen. Des Weiteren wird das modifizierte Ersatzproblem

$$\min_{\varphi_{cur_2}, \dots, \varphi_{cur_5}} \left\| \underbrace{F(\varphi_{t_1}, \varphi_{cur_2}, \dots, \varphi_{cur_5})}_{1 \times 6 \text{ Vektor}} + \underbrace{J_{6 \times 4}(\varphi_{cur_2}, \dots, \varphi_{cur_5}) \cdot \begin{bmatrix} \Delta \varphi_{cur_2} \\ \vdots \\ \Delta \varphi_{cur_5} \end{bmatrix}}_{6 \times 1 \text{ Vektor}} \right\|_2^2 \tag{7.25}$$

betrachtet, bei dem  $\mathbf{J}$  die Jacobimatrix und  $\Delta\varphi$  die Gelenkveränderung kennzeichnet.

Eine Iteration unter Verwendung des Levenberg-Marquardt-Verfahrens besteht aus fünf Schritten:

- Bestimmung der Vorwärtskinematik  $\mathbf{T}_{\text{base}}^{\text{eff}}(\varphi_{t_1}, \varphi_{\text{cur}_2}, \dots, \varphi_{\text{cur}_5})$  der aktuellen Lösung
- Berechnung der Abweichung  $\mathbf{e}$  der Soll/Ist-Transformation
- Ermittlung der Jakobimatrix  $\mathbf{J}(\varphi_{\text{cur}_2}, \dots, \varphi_{\text{cur}_5})$  von dem Roboter
- Berechnung der neuen Lösung für  $\varphi_{\text{new}}$
- Überprüfung der Abbruchkriterien

Dadurch erfolgt die Annäherung des Startwertes  $\varphi_{\text{cur}_2}, \dots, \varphi_{\text{cur}_5}$  näherungsweise an die Ziellösung  $\varphi_{t_2}, \dots, \varphi_{t_5}$ . Der erste Startwert repräsentiert die momentane und bekannte Gelenkstellung  $\varphi_{\text{cur}_2}, \dots, \varphi_{\text{cur}_5}$  des Roboters. Im Nachfolgenden werden die genannten fünf Schritte a) bis f) erläutert. Anmerkend sei erwähnt, dass  $\varphi_{t_1}$  unabhängig der inversen Kinematik berechnet wurde. Damit ist die Lösung für den ersten Gelenkwinkel bekannt und für die Berechnung der Vorwärtskinematik zu verwenden. Jedoch wird die Lösung für den ersten Gelenkwinkel  $\varphi_{t_1}$  nicht verändert.

- Bestimmung der Vorwärtskinematik  $\mathbf{T}_{\text{base}}^{\text{eff}}(\varphi_{t_1}, \varphi_{\text{cur}_2}, \dots, \varphi_{\text{cur}_5})$  der aktuellen Lösung

Im ersten Schritt wird die Vorwärtskinematik des Startwertes bzw. der aktuellen Lösung berechnet. Dabei wird stets der fehlerfrei bestimmte Gelenkwinkel  $\varphi_{t_1}$  verwendet, welcher direkt Anhand der  $x$ - und  $y$ -Endeffektorkoordinate berechnet werden kann (Vgl. Abbildung 7.18).

In der Gleichung 7.26 sind  $\alpha, a, d, \varphi_{\text{cur}}$  die Denavit-Hartenberg-Parameter des Roboters (Vgl. Tabelle 7.2, S. 96). Dabei sind  $\alpha, a, d$  und  $\varphi_{\text{cur}_1}$  ( $\varphi_{\text{cur}_1} = \varphi_{t_1}$ ) bekannt und unveränderbare Größen. Im Gegensatz dazu ändern sich die vier Gelenkwinkel  $\varphi_{\text{cur}_2}, \dots, \varphi_{\text{cur}_5}$  pro Iteration.

$$\mathbf{T}_{\text{base}}^{\text{eff}}(\alpha, a, d, \varphi_{\text{cur}}) = \left( \prod_{i=1}^4 \mathbf{T}_{q_i}^{q_{i+1}}(\alpha_i, a_i, d_i, \varphi_{\text{cur}_i}) \right) \cdot \mathbf{T}_{q_5}^{\text{eff}}(\alpha_5, a_5, d_5, \varphi_{\text{cur}_5}) \quad (7.26)$$

Die Berechnung erfolgt gleichermaßen wie in der Sektion (7.4a) (*Basis-  $\mapsto$  Endeffektorkoordinatensystem  $\mathbf{T}_{\text{base}}^{\text{eff}}$* ), welche auf der Seite 92 beschrieben wurde.

- Berechnung der Abweichung  $\mathbf{e}$  der Soll/Ist-Transformation

Die Abweichung zwischen  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  und der im vorherigen Schritt (7.6a) berechneten Transformation  $\mathbf{T}_{\text{base}}^{\text{eff}}$  ergibt den aktuellen Transformationsfehler  $\mathbf{e}$ .

Die Differenz zwischen der Soll-  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  und Ist-Transformationsmatrix  $\mathbf{T}_{\text{base}}^{\text{eff}}$  wird im nachfolgenden erläutert.

Der Translationsanteil von  $\mathbf{T}_{\text{base}}^{\text{eff}}$  ist von  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  zu subtrahieren (Gleichung 7.27).

$$\mathbf{t}_{\text{diff}} = \begin{bmatrix} x_{\text{diff}} \\ y_{\text{diff}} \\ z_{\text{diff}} \end{bmatrix} = \mathbf{T}_{\text{base}}^{\text{targeteff}} \mathbf{t} - \mathbf{T}_{\text{base}}^{\text{eff}} = \begin{bmatrix} \mathbf{T}_{\text{base}}^{\text{targeteff}} & t_x \\ \mathbf{T}_{\text{base}}^{\text{targeteff}} & t_y \\ \mathbf{T}_{\text{base}}^{\text{targeteff}} & t_z \end{bmatrix} - \begin{bmatrix} \mathbf{T}_{\text{base}}^{\text{eff}} & t_x \\ \mathbf{T}_{\text{base}}^{\text{eff}} & t_y \\ \mathbf{T}_{\text{base}}^{\text{eff}} & t_z \end{bmatrix} \quad (7.27)$$

Die Differenz der Rotation erfolgt gleichermaßen, jedoch muss vorab jeweils die 3 x 3 Rotationsmatrix der Form

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (7.28)$$

in die RPY-Winkelkonvention  $[\phi \ \psi \ \vartheta]$  umgerechnet werden.

Zusammenfassend ergibt sich die Abweichung

$$\mathbf{e}_{6 \times 1} = [x_{\text{diff}}, y_{\text{diff}}, z_{\text{diff}}, \phi_{\text{diff}}, \psi_{\text{diff}}, \vartheta_{\text{diff}}]^T \quad (7.29)$$

zwischen der Soll- und Ist-Transformation.

c) Ermittlung der Jakobimatrix  $\mathbf{J}(\varphi_{\text{cur}_2}, \dots, \varphi_{\text{cur}_5})$  von dem Roboter

In diesem Schritt ist die sogenannte Jacobimatrix  $\mathbf{J}$  in Abhängigkeit zur aktuellen Roboter-Gelenkstellung  $\varphi_{\text{cur}_i}$  mit  $i=2, \dots, 5$  zu berechnen. Die  $6 \times n$  Jacobimatrix des Roboters beschreibt den Zusammenhang zwischen der differentiellen Bewegung aller Gelenke (Gelenkgeschwindigkeiten) und den kartesischen Geschwindigkeiten des Endeffektors. Wobei  $n$  die Anzahl an Gelenken wiedergibt. In dieser Arbeit wird ein Roboter mit fünf Gelenken verwendet. Jedoch ist die Jacobimatrix keine  $6 \times 5$  sondern eine  $6 \times 4$  Matrix, da ein Zielgelenkwinkel  $\varphi_{r_1}$  unabhängig der Iteration direkt berechnet werden kann. Die Jacobimatrix wird durch die partielle Ableitung von  $F$  nach den vier Gelenkwinkeln  $\varphi_2, \dots, \varphi_{\text{cur}_5}$  gebildet. Die Gleichung 7.30 stellt dies unter Verwendung von  $F = [f_1, \dots, f_6]$  (Vgl. Gleichung 7.24) dar.

$$\mathbf{J}_{6 \times 4}(\varphi_{\text{cur}_2}, \dots, \varphi_{\text{cur}_5}) = \begin{bmatrix} \frac{\delta f_1}{\delta \varphi_{\text{cur}_2}} & \frac{\delta f_1}{\delta \varphi_{\text{cur}_3}} & \frac{\delta f_1}{\delta \varphi_{\text{cur}_4}} & \frac{\delta f_1}{\delta \varphi_{\text{cur}_5}} \\ \frac{\delta f_2}{\delta \varphi_{\text{cur}_2}} & \frac{\delta f_2}{\delta \varphi_{\text{cur}_3}} & \frac{\delta f_2}{\delta \varphi_{\text{cur}_4}} & \frac{\delta f_2}{\delta \varphi_{\text{cur}_5}} \\ \frac{\delta f_3}{\delta \varphi_{\text{cur}_2}} & \frac{\delta f_3}{\delta \varphi_{\text{cur}_3}} & \frac{\delta f_3}{\delta \varphi_{\text{cur}_4}} & \frac{\delta f_3}{\delta \varphi_{\text{cur}_5}} \\ \frac{\delta f_4}{\delta \varphi_{\text{cur}_2}} & \frac{\delta f_4}{\delta \varphi_{\text{cur}_3}} & \frac{\delta f_4}{\delta \varphi_{\text{cur}_4}} & \frac{\delta f_4}{\delta \varphi_{\text{cur}_5}} \\ \frac{\delta f_5}{\delta \varphi_{\text{cur}_2}} & \frac{\delta f_5}{\delta \varphi_{\text{cur}_3}} & \frac{\delta f_5}{\delta \varphi_{\text{cur}_4}} & \frac{\delta f_5}{\delta \varphi_{\text{cur}_5}} \\ \frac{\delta f_6}{\delta \varphi_{\text{cur}_2}} & \frac{\delta f_6}{\delta \varphi_{\text{cur}_3}} & \frac{\delta f_6}{\delta \varphi_{\text{cur}_4}} & \frac{\delta f_6}{\delta \varphi_{\text{cur}_5}} \end{bmatrix} \quad (7.30)$$

Weitere Informationen über die Berechnung der Jacobimatrix sind aus [ANG07], [COR96] und [PSM81] zu entnehmen.

d) Berechnung der neuen Lösung  $\varphi_{\text{new}}$

Die Jacobimatrix  $\mathbf{J}$  aus dem vorherigen Schritt und der Transformationsfehler  $\mathbf{e}$  aus dem Schritt (7.6b) erlauben die Berechnung der approximierten Lösung für die gesuchte Zielgelenkstellung  $\varphi_r$ .

Eine Lösung liefert die Pseudoinverse der Jacobimatrix  $\mathbf{J}^+$  multipliziert mit dem Transformationsfehler (Gleichung 7.31).

$$\varphi_{\text{new}_{4 \times 1}} = \begin{bmatrix} \varphi_{\text{cur}_2} \\ \vdots \\ \varphi_{\text{cur}_5} \end{bmatrix} + \begin{bmatrix} \Delta \varphi_{\text{cur}_2} \\ \vdots \\ \Delta \varphi_{\text{cur}_5} \end{bmatrix} = \begin{bmatrix} \varphi_{\text{cur}_2} \\ \vdots \\ \varphi_{\text{cur}_5} \end{bmatrix} + \underbrace{\mathbf{J}_{4 \times 6}^+ \cdot \mathbf{e}_{6 \times 1}}_{4 \times 1 \text{ Vektor}} \quad (7.31)$$

Die Pseudoinverse der Jacobimatrix ist eine verallgemeinerte Inverse auf singuläre und nicht-quadratische Matrizen. Für die Berechnung wird die Singulärwertzerlegung (Vgl. Kapitel 3.2) herangezogen. Das Ergebnis der Pseudoinversen der  $6 \times 4$  Jacobimatrix  $\mathbf{J}$  ist die  $4 \times 6$  Matrix  $\mathbf{J}^+$ , sodass die Formeln 7.32 bis 7.35 gültig und hermitesch<sup>34</sup> sind.

$$\mathbf{J} \cdot \mathbf{J}^+ \cdot \mathbf{J} = \mathbf{J} \quad (7.32)$$

$$\mathbf{J}^+ \cdot \mathbf{J} \cdot \mathbf{J}^+ = \mathbf{J}^+ \quad (7.33)$$

$$\mathbf{J} \cdot \mathbf{J}^+ \quad (7.34)$$

$$\mathbf{J}^+ \cdot \mathbf{J} \quad (7.35)$$

Für eine ausführliche Beschreibung der Pseudoinversen wird auf [BUS09] und [COR96] verwiesen.

Eine andere Möglichkeit zur Berechnung der Gelenkveränderung  $\Delta\varphi$  bzw. der neuen Lösung  $\varphi_{\text{new}}$  besteht in dem Aufstellen und Lösen des nachfolgenden Gleichungssystems 7.36.

$$\mathbf{J}_{6 \times 4} \cdot \begin{bmatrix} \Delta\varphi_{\text{cur}_2} \\ \vdots \\ \Delta\varphi_{\text{cur}_5} \end{bmatrix} = \mathbf{e}_{6 \times 1} \quad (7.36)$$

Für die Lösung des Gleichungssystems kann ein Eliminationsverfahren herangezogen werden.

Im Anschluss nach der Berechnung von  $\varphi_{\text{new}}$ , welche die gesuchte Ziellösung  $\varphi_t$  besser annähert, ist anschließend zu prüfen ob eine weitere Iteration erfolgen muss.

#### e) Überprüfung der Abbruchkriterien

In dem Fall, dass die aktuelle Zielabweichung

$$\varphi_e = \|\Delta\varphi_{\text{cur}_2}, \dots, \Delta\varphi_{\text{cur}_5}\|_2 = \sqrt{\sum_{i=2}^5 |\Delta\varphi_{\text{cur}_i}|^2} \quad (7.37)$$

ein  $\varepsilon$  unterschreitet oder ein festgelegtes Iterationslimit erreicht wurde, konvergiert das Verfahren mit der Lösung

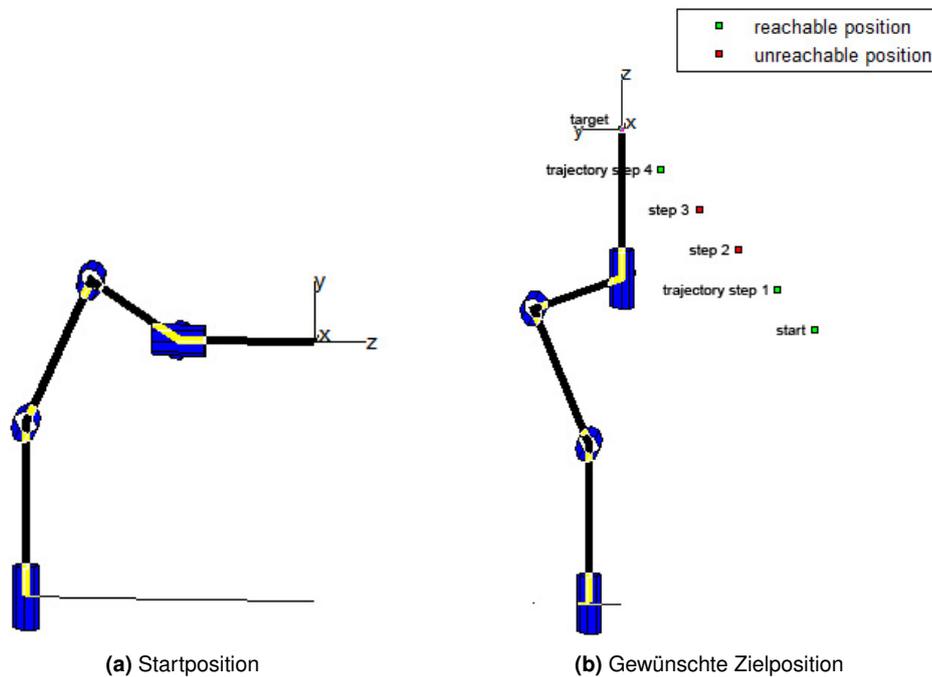
$$\varphi_{t_{5 \times 1}} = [\varphi_{t_1}, \varphi_{\text{new}_{4 \times 1}}^T]^T \quad (7.38)$$

Andernfalls wird der Schritt (7.6a) mit der aktuellen Näherung  $\varphi_{\text{new}}$  erneut durchgeführt. Dann erhält  $\varphi_{\text{cur}}$  den neuen Wert  $\varphi_{\text{new}}$ .

Im konvergierten Fall müssen die Gelenkwinkel  $\varphi_{t_i}$  für  $i=1, \dots, 5$  noch auf den gültigen Wertebereich der Gelenke überprüft werden. Im Kapitel 4.1 in der Tabelle 4.1 sind die möglichen Verfahrwinkel der einzelnen Gelenke aufgelistet. In der Situation, dass sich ein oder mehrere Gelenkwinkel außerhalb des gültigen Wertebereiches befinden, ist die Lösung  $\varphi_t$  unbrauchbar. Tritt diese Situation ein, kann ein minimal veränderter Start-  $\varphi_{\text{cur}}$  oder Zielwert  $\varphi_t$  zu einer erfolgreichen Lösung führen. In dieser Arbeit wird jedoch die Lösung der inversen Kinematik als unbrauchbar angesehen und die Berechnung der inversen Kinematik für den nachfolgenden Trajektorien-schritt bekommt dann nicht die Lösung aus der aktuellen-, sondern aus der letzten Trajektorienposition. Die

<sup>34</sup>Eine quadratische Matrix  $\mathbf{M}$  ist hermitisch, wenn ihre Transponierte  $\mathbf{M}^T$  der komplex Konjugierten  $\overline{\mathbf{M}}$  entspricht. Beispielsweise ist  $\mathbf{M}$  durch  $\underbrace{[\mathbf{J}_{6 \times 4} \cdot \mathbf{J}_{4 \times 6}^+]}_{6 \times 6 \text{ Matrix}}$  definiert und  $\mathbf{M}^T = \overline{\mathbf{M}}$  ist erfüllt.

Abbildung 7.19 veranschaulicht dies. Für den ersten Trajektorien-schritt bzw. -position konnte die inverse Kinematik, ohne die Gelenkbeschränkungen zu verletzen, berechnet werden (Grünes Dreieck). Das rote Dreieck im zweiten Trajektorien-schritt kennzeichnet eine Roboterposition, welche aufgrund der Gelenkbeschränkungen nicht exakt angefahren werden kann.



**Abbildung 7.19:** Trajektorie-basierte inverse Kinematik.

Zusammenfassend wird somit eine Trajektorie zwischen der Start- und Zielposition berechnet. Für jeden Trajektorien-schritt erfolgt eine iterative Berechnung der inversen Kinematik. Dadurch werden die gesuchten Gelenkwinkel  $\varphi_i$  für  $i = 2, \dots, 4$  sukzessiv ermittelt.

## 7.7 Berechnung der Greifpose abhängig zur Zielpose

Nachdem die Zielgelenkstellung  $\varphi_t$  der Zielpose  $\mathbf{T}_{\text{base}}^{\text{targeteff}}$  optimal berechnet und angefahren wurde, kann die Greifpose fehlerfrei bestimmt werden. Dafür wird die Transformation  $\mathbf{T}_{\text{targetcam}_{\text{init}}}^{\text{grip}_{\text{cam}_{\text{init}}}}$  und  $\mathbf{T}_{\text{eff}}^{\text{cam}}$  benötigt.  $\mathbf{T}_{\text{targetcam}_{\text{init}}}^{\text{grip}_{\text{cam}_{\text{init}}}}$  wurde in der Phase 7.1 (*Vorbedingungen*) speziell für einen Gegenstand festgelegt und beschreibt die relative Lage und Orientierung zwischen der initialen Ziel- und Greifpose. Üblicherweise ist der Gegenstand nicht gleichermaßen wie in der Phase 7.1 räumlich positioniert und ausgerichtet. Aus diesem Grund wurde in der Vorbereitungsphase die relative Pose im Basiskoordinatensystem ermittelt:

$$\mathbf{T}_{\text{targetcam}_{\text{init}}}^{\text{grip}_{\text{cam}_{\text{init}}}} = \text{inv} \left( \mathbf{T}_{\text{base}}^{\text{targetcam}_{\text{init}}} \right) - \text{inv} \left( \mathbf{T}_{\text{base}}^{\text{grip}_{\text{cam}_{\text{init}}}} \right) \quad (7.39)$$

Deshalb ist die Transformation  $\mathbf{T}_{\text{targetcam}_{\text{init}}}^{\text{grip}_{\text{cam}_{\text{init}}}}$  im Basiskoordinatensystem durchzuführen. Die Invertierung von  $\mathbf{T}_{\text{base}}^{\text{targetcam}_{\text{init}}}$  erzielt den Wechsel vom Kamerakoordinatensystem der Zielpose zum Basiskoordinatensystem. Jetzt darf der relative Versatz und die Ausrichtung zwischen Ziel- und Greifpose angepasst werden. Abschließend führt die Invertierung von  $\left( \mathbf{T}_{\text{targetcam}_{\text{init}}}^{\text{base}} - \mathbf{T}_{\text{targetcam}_{\text{init}}}^{\text{grip}_{\text{cam}_{\text{init}}}} \right)$  zur gesuchten Greifpose  $\mathbf{T}_{\text{base}}^{\text{grip}_{\text{cam}_{\text{init}}}}$ .

Für die Greifpose im Endeffektorkoordinatensystem ist die bekannte Abbildung  $\mathbf{T}_{\text{eff}}^{\text{cam}}$  aus der Sektion (7.4b) auf Seite 97 zu verwenden:

$$\mathbf{T}_{\text{base}}^{\text{gripeff}} = \mathbf{T}_{\text{base}}^{\text{grip}_{\text{cam}_{\text{init}}}} \cdot \text{inv} \left( \mathbf{T}_{\text{eff}}^{\text{cam}} \right) \quad (7.40)$$

## 7.8 Inverse Kinematik der Greifpose

Gleichermaßen wie in der Phase 7.6 (*Inverse Kinematik der Zielpose*) ist die inverse Kinematik der Greifpose zu berechnen. Das Resultat sind die fünf Gelenkwinkel, welche den Roboter in die Greifpose  $T_{base}^{grieff}$  bewegen.

## 7.9 Mobilisierung

Nach der Berechnung der Gelenkwinkel kann der gegriffene Gegenstand im Umfeld des Roboters beliebig bewegt und abgelegt werden (Abbildung 7.20).



(a) Startpose angefahren + Zielpose berechnen



(b) Zielpose angefahren + Greifpose berechnen



(c) Greifpose angefahren



(d) Abgabeplatz erreicht



(e) Observationsposition erreicht

**Abbildung 7.20:** Objektmobilisierung mit dem Knickarm-Roboter.

Die Ablage des Gegenstandes erfolgt nachdem der gewünschte Abgabeplatz angefahren wurde. Danach bewegt sich der Roboter zu einer definierten Observationsposition zurück. Aus dieser Lage können erneut Gegenstände identifiziert, lokalisiert und gegriffen werden.

Abschließend veranschaulicht die Abbildung 7.21 das gesamte Vorgehen zur Manipulation bzw. Mobilisierung von Gegenständen. Ausgehend einer Startposition (a) wird ein Gegenstand in dem Kamerabild (b) gesucht. Die Suche erfolgt durch einen Merkmalsvergleich zwischen den Merkmalen im Kamerabild und dem bekannten 3D-Merkmalmodell des Gegenstandes. Anschließend erfolgt die Berechnung der Lage und Orientierung des Gegenstandes anhand der Merkmalskorrespondenzmenge (Grafik c). Da die Objektpose nun bekannt ist, kann eine vorab definierte Zielpose (e) angefahren werden. Befindet sich die Roboterkamera in der Zielpose wird die vorab festgelegte Greifpose (f) eingenommen. Abschließend findet eine Mobilisierung des Gegenstandes statt und der Roboter wird in seine ursprüngliche Observationsposition zurückgefahren (Grafik g und h).

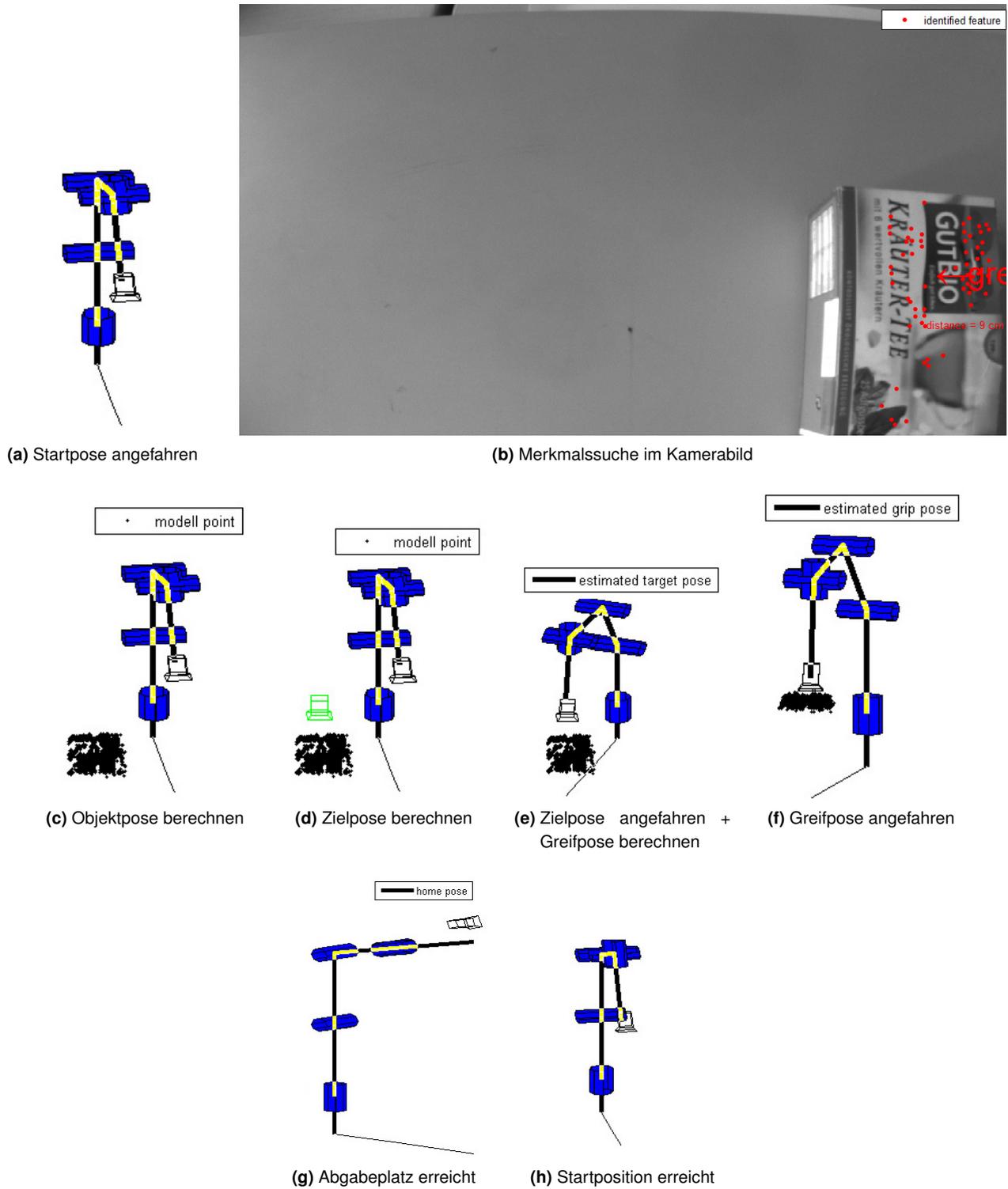


Abbildung 7.21: Prinzip der Objektmobilisierung.

# 8 Ergebnisse

Dieses Kapitel stellt experimentelle Ergebnisse, der in dieser Arbeit implementierten Verfahren, vor.

## Inhalt

---

8.1	3D-Merkmalmodelle . . . . .	109
8.1.1	Testobjekt . . . . .	109
8.1.2	Papierschachtel . . . . .	116
8.1.3	Tee-Packung . . . . .	119
8.1.4	VHS-Kassette . . . . .	121
8.2	Identifikation von Gegenständen . . . . .	122
8.3	Mobilisierung von Gegenständen . . . . .	123

---

## 8.1 3D-Merkmalmodelle

Das erste Unterkapitel präsentiert vier Messresultate. Diese veranschaulichen die Leistungsfähigkeit über das entwickelte Verfahren zur Generierung eines 3D-Merkmalmodells. Alle Angaben sind gerundet angegeben.

### 8.1.1 Testobjekt

Grundsätzlich ist die Generierung eines 3D-Merkmalmodells immer mit fehlerfreien- und damit *idealen* Merkmalskorrespondenzen realisierbar. In dieser Arbeit werden die Merkmalskorrespondenzen jedoch autonom berechnet, wodurch Korrespondenzen mit einer Abweichung im Bereich von  $[0; 1]$  Pixeln nicht auszuschließen sind. Mit Hilfe bekannter fehlerfreier Merkmalskorrespondenzen konnte, unabhängig von der Merkmalsdetektion und der Korrespondenzfindung, die in dieser Arbeit implementierte Generierung eines 3D-Merkmalmodells getestet werden.

Für das unabhängige Testen sind fehlerfreie Merkmalskorrespondenzen notwendig. Aus diesem Grund wurde eine quaderförmige Box als Testobjekt entworfen (Vgl. Abbildung 8.1).

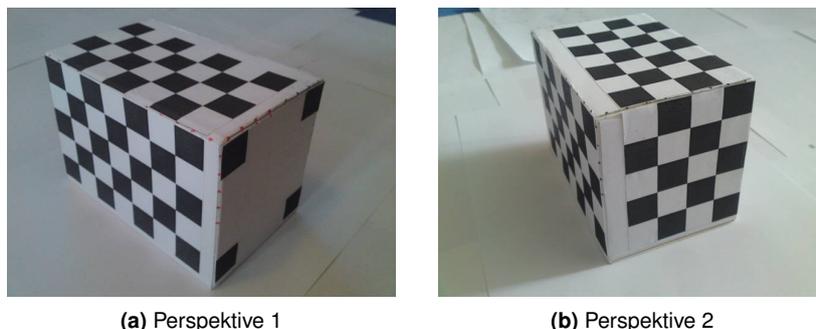
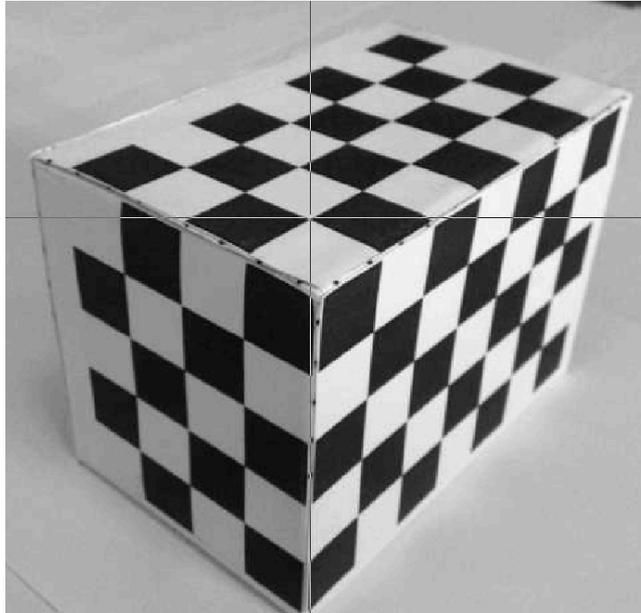


Abbildung 8.1: Testobjekt.

Auf diesem Testobjekt ist teilweise ein Schachbrettmuster abgebildet. Bei einem Schachbrettmuster sind die Ecken der Quadrate geeignete Merkmalskandidaten. Deshalb wurden vom Testobjekt elf Aufnahmen aus unterschiedlichen Perspektiven aufgenommen. Dabei kam eine Handykamera zum Einsatz (Siehe Kapitel 4.4, S. 24).

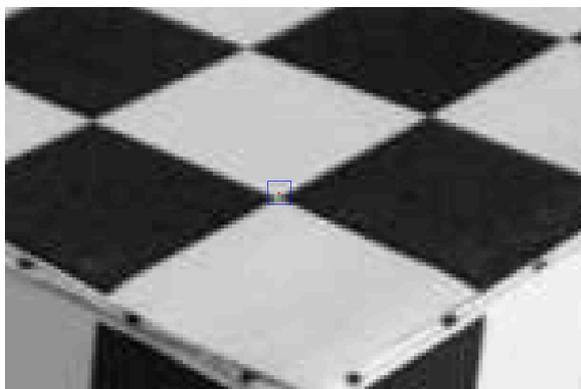
An dieser Stelle sei zu beachten, dass die Aufnahmen möglichst das komplette Testobjekt flächendeckend abdecken sollten. Denn dies führt in einem späteren Zeitpunkt zu einem vollständigen 3D-Merkmalmodell. Anschließend sind in jeder Aufnahme mit dem Mauszeiger die Merkmale (Ecken) sowie deren Korrespondenzen auszuwählen (Vgl. Abbildung 8.2).



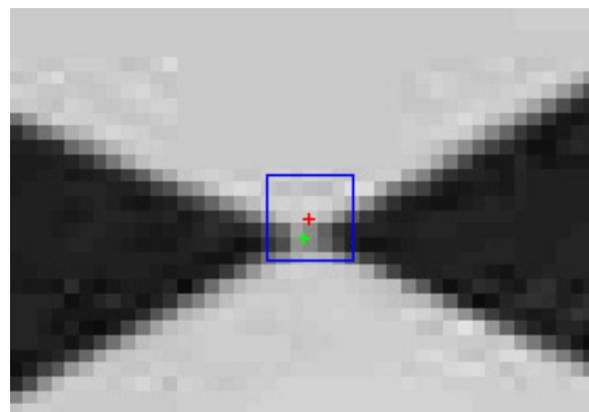
**Abbildung 8.2:** Markierung eines Merkmales (Ecke) mit dem Mauszeiger.

Im Praxisversuch hat sich gezeigt, dass sich die Eckpunkte nicht exakt mit der Maus selektieren lassen. Aus diesem Grund erwies sich diese Suche nach fehlerfreien Korrespondenzen als zu ungenau. Jedoch erzielte eine verbesserte Variante den Gewinn fehlerfreier Merkmalskorrespondenzen.

Das Markieren eines Merkmales wurde als grobe Selektierung gewertet. Nachdem eine Ecke vom Schachbrettmuster selektiert war, kam eine subpixelgenaue Korrektur zum Einsatz. Die Abbildung 8.3 veranschaulicht diese Korrektur.



(a) Subpixelgenaue Korrektur



(b) Nahaufnahme

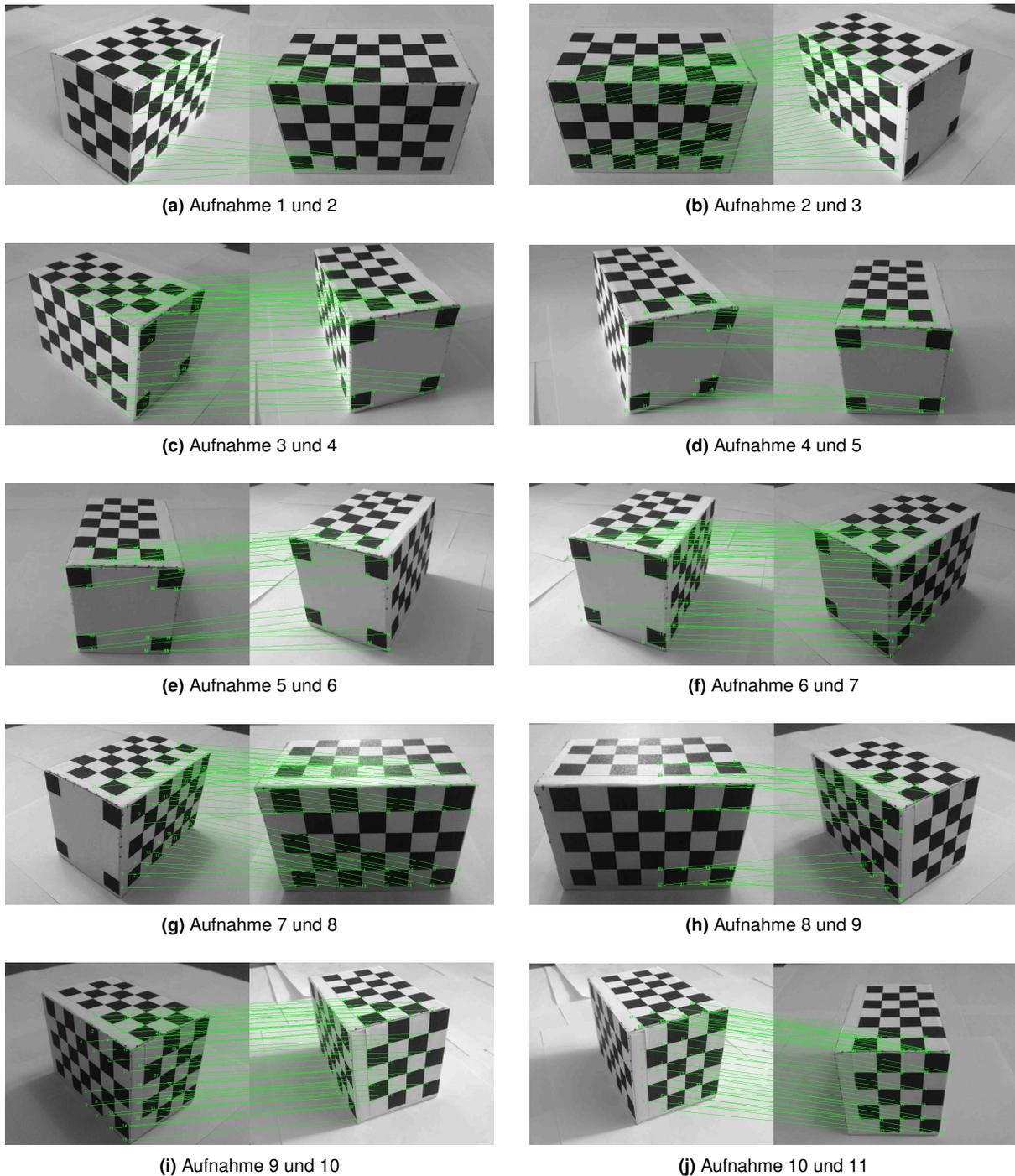
**Abbildung 8.3:** Subpixelgenaue Korrektur der selektierten Ecke.

Die mit dem Mauszeiger grob markierte Ecke ist durch ein rotes Kreuz gekennzeichnet. Ausgehend von dieser Bildposition  $\mathbf{x} = [x_x, x_y]^T$  wurde eine sogenannte *Region of Interest* (ROI) aufgespannt. Sie deckt die 5 x 5

Pixel große Umgebungsfläche von  $\mathbf{x}$  ab. In diesem blau umrandeten Bereich wird die Ecke auf 0,1 Pixel genau bestimmt. Das Ergebnis zeigt das grün dargestellte Kreuz an der Bildposition  $\hat{\mathbf{x}} = [\hat{x}_x, \hat{x}_y]^T$ .

Das in dieser Arbeit implementierte Verfahren zur subpixelgenauen Bestimmung einer Ecke bzw. eines Merkmales basiert auf einer Erweiterung des sogenannten *Harris Corner Detectors*<sup>35</sup>. Eine ausführliche Beschreibung des *Harris Corner Detectors* ist in [HS88] zu finden. Die in dieser Arbeit implementierte Erweiterung stammt aus [BOU08].

Die ermittelten Punktkorrespondenzen zwischen den insgesamt elf Aufnahmen sind in der Abbildung 8.4 abgebildet.



**Abbildung 8.4:** Korrespondenzmengen.

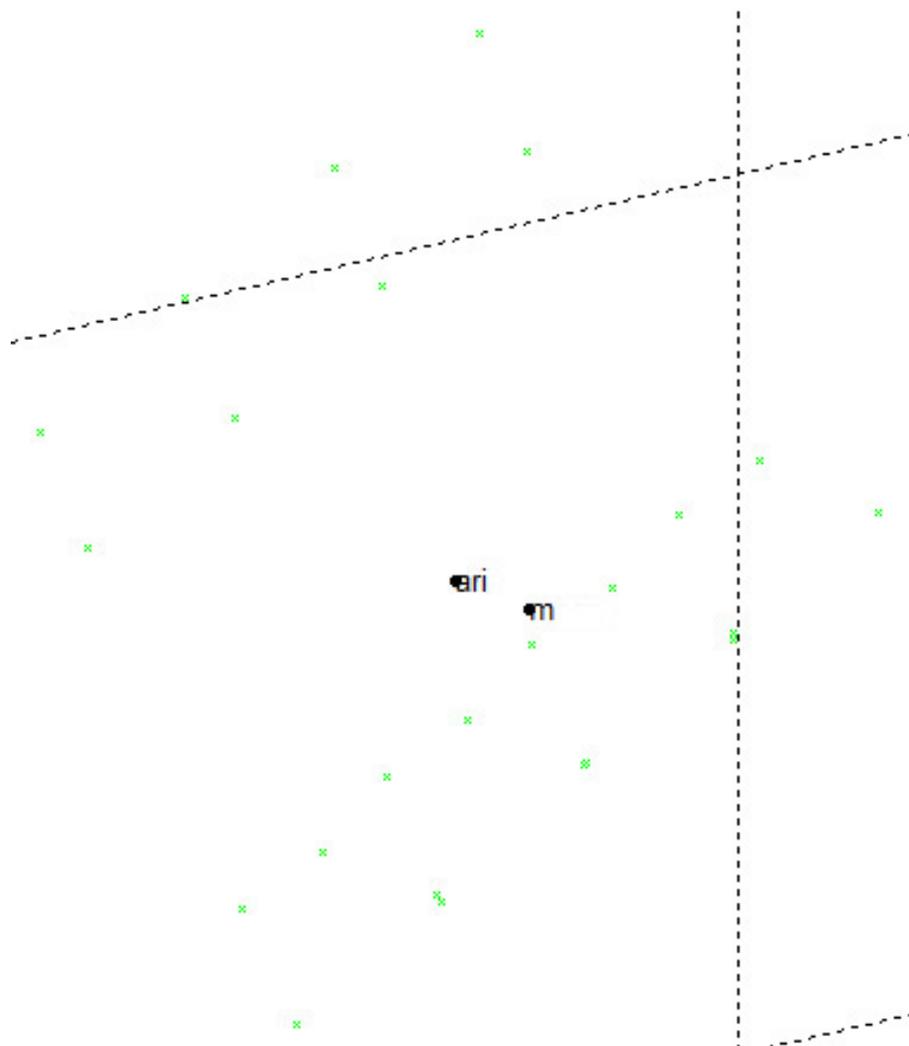
<sup>35</sup>Der Harris Corner Detector wird u. a. auch als *Plessy*-Punkt-detektor bezeichnet.

Nachdem die exakten Merkmalskorrespondenzen zwischen jeweils zwei Aufnahmen ermittelt ist, kann die Generierung eines 3D-Merkmalmodells getestet werden. Dafür wurden die Weltpunkte mit den Merkmalskorrespondenzen zwischen der ersten- und zweiten Kameraaufnahme  $\mathbf{B}_1 \leftrightarrow \mathbf{B}_2$  berechnet (Vgl. Kapitel 5.3, *Initiale Bestimmung von Weltpunkten*).

Weiterführend erfolgte unter anderem eine Überprüfung auf eine degenerierte Merkmalsmenge. Diese Überprüfung schlug nicht fehl. Bei der Markierung der Merkmale ist darauf zu achten, dass die Merkmalsmenge pro Bild nicht auf einer planaren Fläche liegt. Dadurch kann keine Strukturdegeneration vorliegen. Das Auftreten einer Bewegungsdegeneration ist durch die perspektivisch unterschiedlichen Aufnahmen ausgeschlossen.

Die Triangulation und damit die Berechnung der 3D-Weltpunkte  $\hat{\mathbf{X}}$  aus 2D-Punktcorrespondenzen  $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}_j$  erzielte im Anschluss der Filtermaßnahmen gute Resultate.

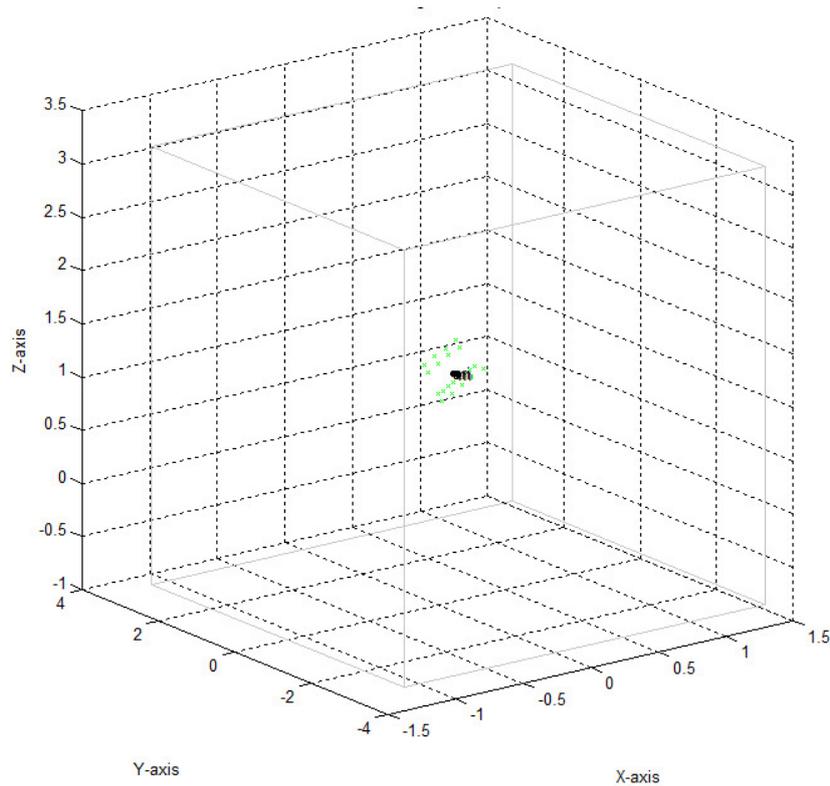
Die erste Filtermaßnahme prüft auf fehlerhaft triangulierte Weltpunkte. Dafür wird jede räumliche Weltpunktposition im Verhältnis zu allen Weltpunkten betrachtet. Zuerst muss der Schwerpunkt der Punktmenge ermittelt werden. Der Median ist Resistent gegen Ausreißer und kommt resultierend für die Berechnung des Schwerpunktes zum Einsatz. Die nachfolgende Abbildung 8.5 stellt den Median  $m$  sowie den arithmetischen Mittelwert  $ari$ , als alternativen Schwerpunkt, dar.



**Abbildung 8.5:** Arithmetischer Mittelwert  $ari$ - und Median  $m$  der triangulierten Punktmenge.

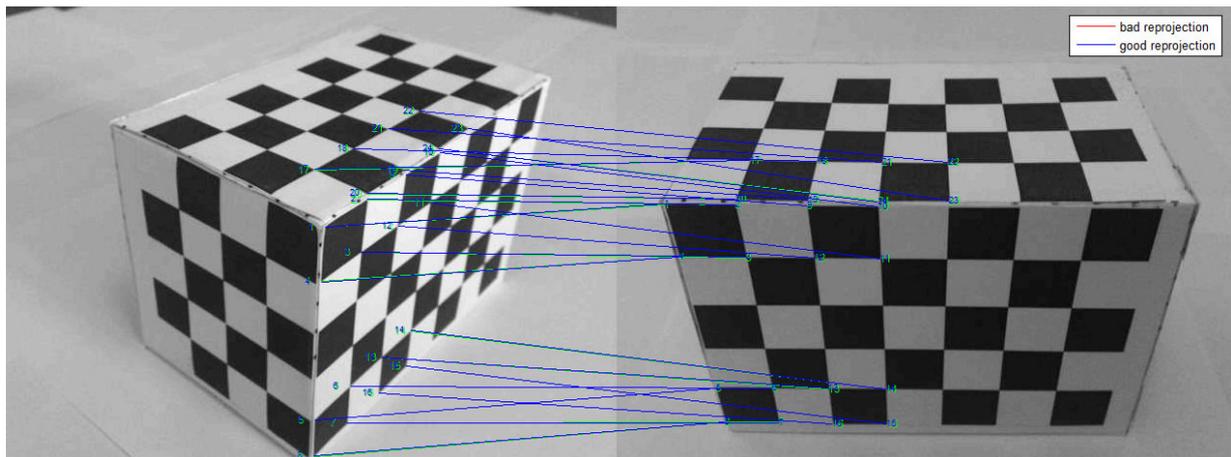
Nachdem der Schwerpunkt der Weltpunkte berechnet ist, muss der arithmetische Mittelwert zwischen den Weltpunkten und dem Schwerpunkt berechnet werden. Dieser Faktor verstärkte Mittelwert ermöglicht die Spezifikation eines Gültigkeitsbereiches ausgehend vom Schwerpunkt. Je größer der Faktor ist, desto größer wird der Gültigkeitsbereich. Analog gilt die Umkehrung. (Vgl. Kapitel 5.3.3.2, S. 56)

Der Gültigkeitsbereich ist in der Abbildung 8.6 durch ein graues Quader visualisiert. Grün gekennzeichnete Weltpunkte liegen innerhalb des Gültigkeitsbereiches, dagegen sind rote Weltpunkte als fehlerhaft verifiziert worden.



**Abbildung 8.6:** Filterung von fehlerhaft triangulierten Weltpunkten.

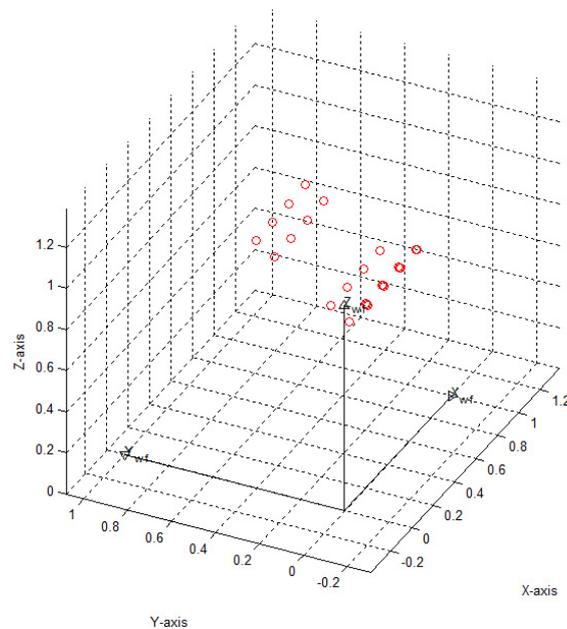
Im Anschluss der Aussortierung von falsch triangulierten Weltpunkten findet eine Filterung anhand des Rückprojektionsfehlers statt. Die Abbildung 8.7 stellt das Resultat dieser Filterung dar. Jeweils eine blaue bzw. rote Linie zeigt eine Punktkorrespondenz von einem rückprojizierten Weltpunkt. Steigt der Rückprojektionsfehler über einen gewissen Schwellwert, wird diese Linie rot, andernfalls blau dargestellt. Im gezeigten Beispiel liegt der Rückprojektionsfehler weit unter dem Schwellwert, sodass keine Weltpunkte aussortiert werden.



**Abbildung 8.7:** Rückprojektion der Weltpunkte zu den ursprünglichen Bildkoordinaten.

Im letzten Schritt muss die richtige Skalierung der Weltpunkte bestimmt und korrigiert werden. Dies liegt an dem Grund, dass die Triangulation bis auf einen unbekanntem Skalierungsfaktor genau ist (Vgl. Seite 58).

Die nachfolgende Abbildung 8.8 zeigt die korrigierten initialen Weltpunkte. Diese sind zusammen mit ihren Merkmalen in einem sogenannten 3D-Merkmalmodell abzulegen.

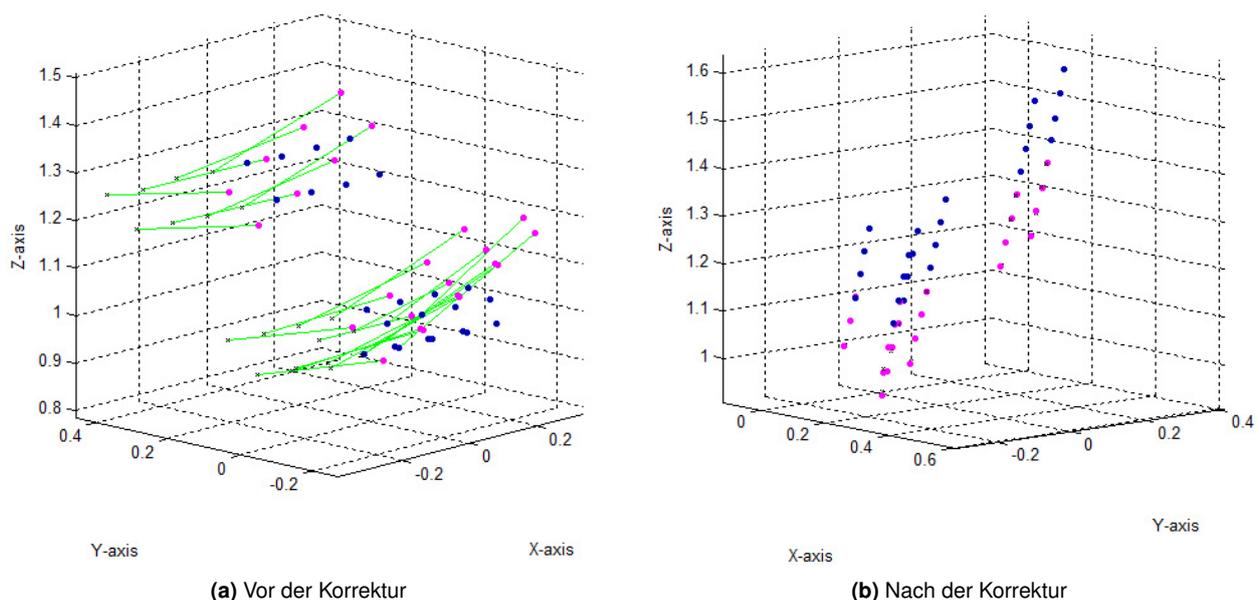


**Abbildung 8.8:** Resultierende Weltpunkte nach Durchführung diverser Filtermaßnahmen.

Für einen hohen Wiedererkennungsgrad eines Gegenstandes müssen Merkmale aus weiteren Aufnahmen in das bereits bestehende 3D-Merkmalmodell ergänzt werden. Aus diesem Grund sind gleichermaßen wie bisher die bekannten Merkmalskorrespondenzen zwischen den nachfolgenden Aufnahmen  $\mathbf{B}_2 \leftrightarrow \mathbf{B}_3$ ,  $\mathbf{B}_3 \leftrightarrow \mathbf{B}_4 \dots \mathbf{B}_{10} \leftrightarrow \mathbf{B}_{11}$  für weitere Triangulationen heranzuziehen.

Nachdem auch bei diesen triangulierten Weltpunkten die Filtermaßnahmen erfolgreich durchgeführt sind, muss eine räumliche Positionskorrektur der neuen Weltpunkte erfolgen (Vgl. Kapitel 5.4.3.4, S. 67).

Die Projektionsmatrizen  $\mathbf{P}_{p,p>2}$  stehen in keiner Relation zu den ersten Beiden  $\mathbf{P}_{p \in \{1,2\}}$ . Aus diesem Grund beinhalten die neuen Projektionsmatrizen  $\mathbf{P}_{p,p>2}$  nicht die Kamerabewegung und -rotation der ersten Beiden. Des Weiteren stimmt die Skalierung der 3D-Punkte nicht. Deshalb müssen die neuen Weltpunkte korrigiert werden. Die Suche nach den 3D-Korrespondenzen zwischen den bestehenden Weltpunkten des 3D-Merkmalmodells und den neu-triangulierten Weltpunkten erzielt die gewünschte räumliche Korrektur. Die nachfolgende Abbildung 8.9 zeigt die 3D-Punktcorrespondenzen sowie das Resultat der korrigierten Weltpunkte, welche durch die Korrespondenzmenge der beiden Aufnahmen  $\mathbf{B}_2$  und  $\mathbf{B}_3$  trianguliert wurden.



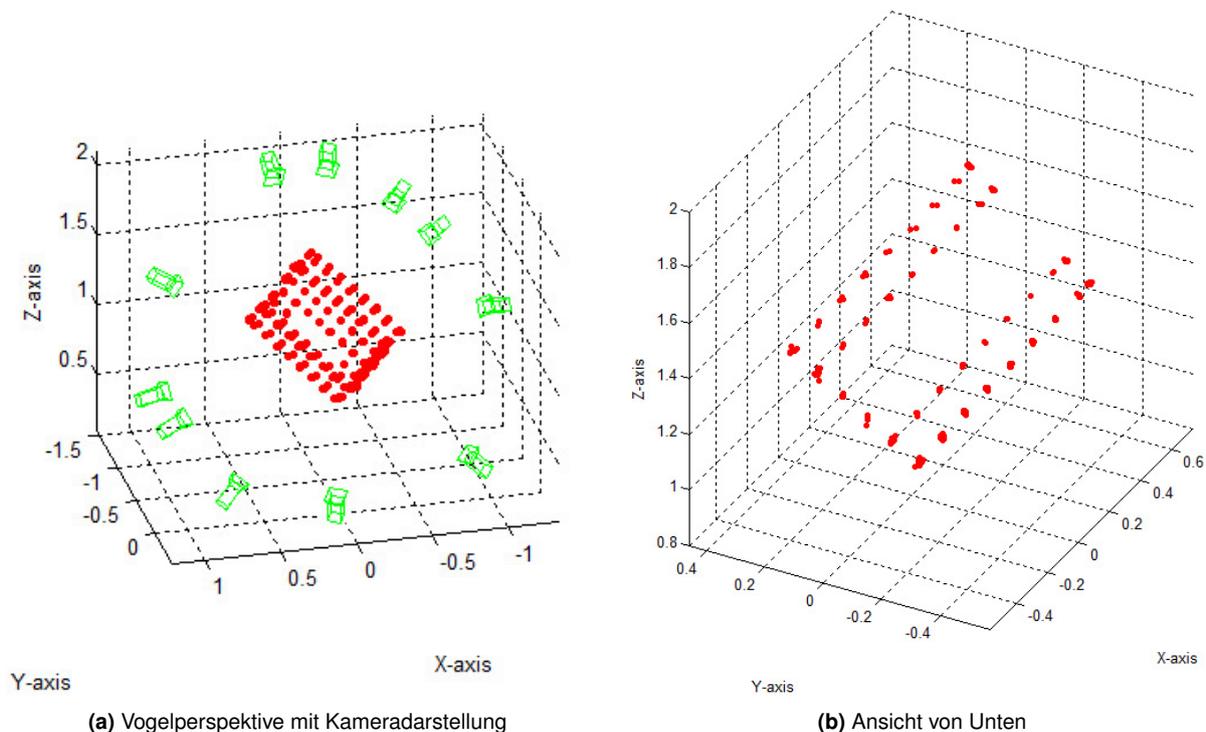
**Abbildung 8.9:** Räumliche Korrektur von neuen Weltpunkten.

Die linke Grafik 8.9a veranschaulicht die 3D-Punktkorrespondenzen  $\mathbf{X}_i^{\text{modell}} \leftrightarrow \mathbf{X}_j^{\text{new}}$  zwischen den Weltpunkten  $\mathbf{X}_i^{\text{modell}}$  des 3D-Merkmalmodells und den neuen Weltpunkten  $\mathbf{X}_j^{\text{new}}$ . Dabei sind Weltpunkte des 3D-Merkmalmodells pink gekennzeichnet. Die mit einem  $x$  symbolisierten Punkte stellen die neuen Weltpunkte dar. Mit einer grünen oder roten Linie sind diese mit den korrespondierenden Weltpunkten des 3D-Merkmalmodells verbunden. In dem Fall, dass der Abstand  $|\mathbf{X}_j^{\text{new}} - \mathbf{X}_i^{\text{modell}}|$  über ein Schwellwert  $v$  steigt, wird dieser rot-, andernfalls grün dargestellt.

In der rechten Grafik 8.9b sind, anhand der 3D-Korrespondenzmenge, die neuen Weltpunkte räumlich an die Weltpunkte des 3D-Merkmalmodells angepasst worden. Nach der Korrektur beinhalten die neuen Weltpunkte die Kamerabewegung, -rotation sowie die richtige Skalierung der initialen Weltpunkte.

Für die weiteren Aufnahmen  $\mathbf{B}_3 \leftrightarrow \mathbf{B}_4$  bis einschließlich  $\mathbf{B}_{10} \leftrightarrow \mathbf{B}_{11}$  wurden die neu triangulierten Weltpunkte gleichermaßen in Bezug auf die Korrespondenzmenge  $\mathbf{X}_i^{\text{modell}} \leftrightarrow \mathbf{X}_j^{\text{new}}$  durchgeführt. Resultierend konnte aus allen Triangulationen die in Abbildung 8.10 dargestellte Weltpunktmenge ermittelt werden.

Anmerkend zu der Grafik 8.10b sei erwähnt, dass die Schachbrettmuster der Testbox nicht gleichverteilt aufgelegt sind. Aus diesem Grund können sich die Punktabstände unterscheiden. Die abgebildeten 3D-Oberflächenpunkte wurden mit den Originalmaßen verifiziert. Dabei resultierte eine Genauigkeit im Millimeterbereich.



**Abbildung 8.10:** Erzielte Weltpunktmenge der Testbox. Dafür wurden elf perspektivisch unterschiedliche Aufnahmen von einer Handkamera verwendet.

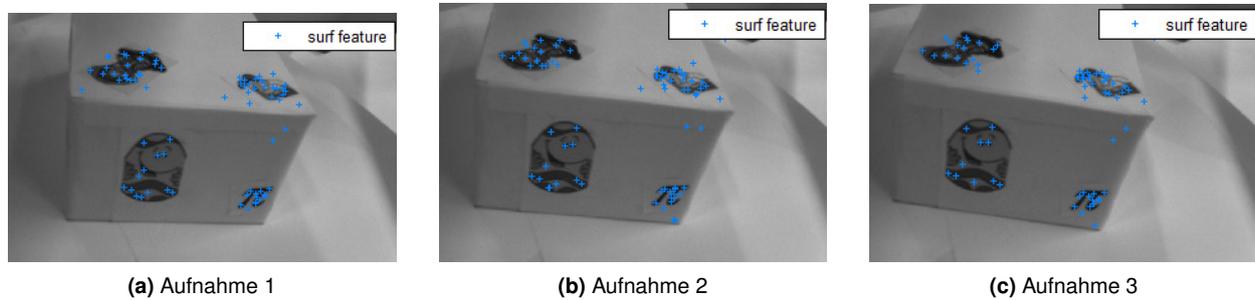
Abschließend sind in der Tabelle 8.1 die Messergebnisse der Modellgenerierung aufgelistet. Jeweils zwei der insgesamt elf Kameraaufnahmen verfügen über 24 Merkmalskorrespondenzen. Diese wurden mit der auf S. 109 beschriebenen Methode subpixelgenau ermittelt. Insgesamt resultierten 240 Korrespondenzen, welche zu  $10 \cdot 24 = 240$  Oberflächenpunkte trianguliert wurden. Die dafür benötigte Berechnungsdauer betrug 3 Sekunden.

Aufnahmen	Korrespondenzen	Oberflächenpunkte (3D-Weltpunkte)	Berechnungsdauer [Sek.]
1-11	240	240	3

**Tabelle 8.1:** Messresultat der Testbox.

### 8.1.2 Papierschachtel

Die Merkmalerkennung einer selbstgebastelten Papierschachtel ist in Abbildung 8.11 dargestellt. In einer Berechnungszeit von  $3 \times 0,08 \text{ s} = 0,24$  Sekunden konnten insgesamt  $81 + 79 + 83 = 243$  SURF-Merkmale mit einer Invarianz gegenüber Rotation, Translation, Skalierung, Änderung der Lichtverhältnisse und teilweiser affiner Verzerrung detektiert werden.



**Abbildung 8.11:** Detektierte SURF-Merkmale der Papierschachtel.

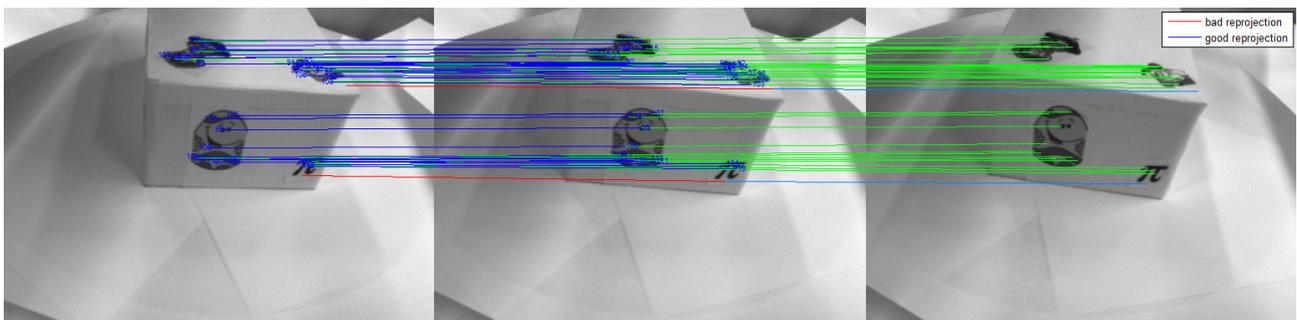
Anschließend wurden die Merkmalskorrespondenzen zwischen den ersten beiden-  $\mathbf{B}_1 \leftrightarrow \mathbf{B}_2$  und den nächsten beiden Aufnahmen  $\mathbf{B}_2 \leftrightarrow \mathbf{B}_3$  bestimmt. Dies zeigt die Abbildung 8.12. Aus den ersten beiden Bildern waren 48- und in  $\mathbf{B}_2 \leftrightarrow \mathbf{B}_3$  47 Merkmalskorrespondenzen ermittelbar.



**Abbildung 8.12:** Merkmalsvergleich zwischen den ersten beiden-  $\mathbf{B}_1 \leftrightarrow \mathbf{B}_2$  und den nächsten beiden Aufnahmen  $\mathbf{B}_2 \leftrightarrow \mathbf{B}_3$ .

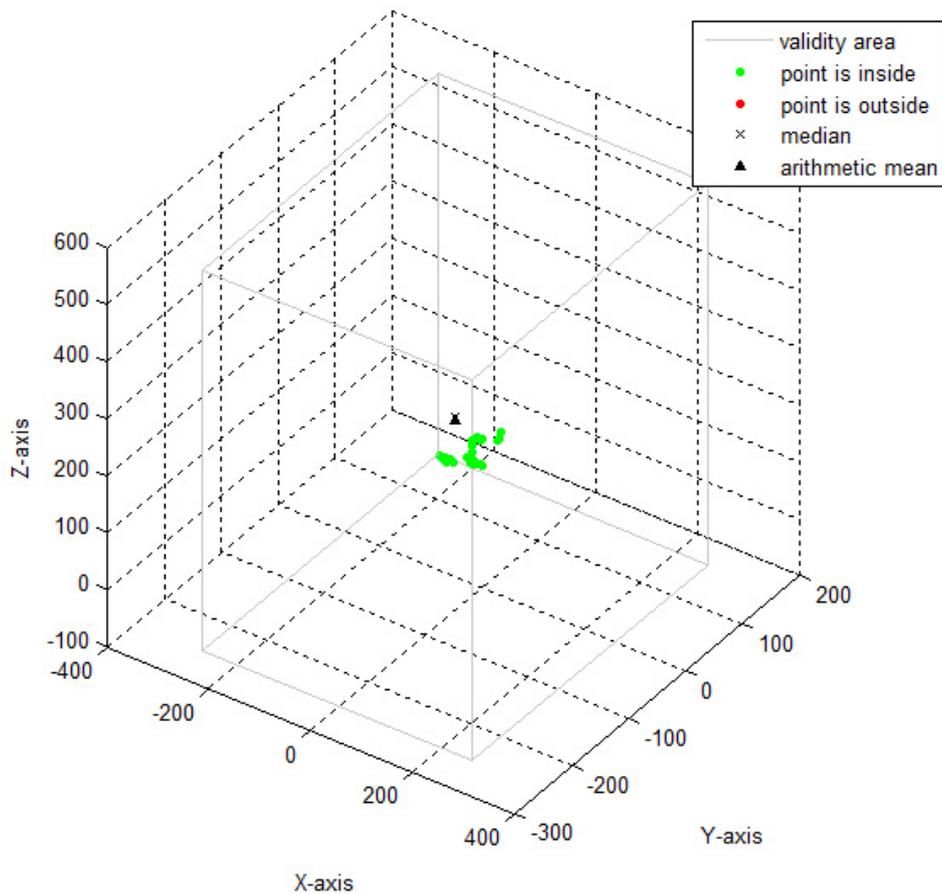
Die Schnittmenge  $\{\mathbf{B}_1 \leftrightarrow \mathbf{B}_2\} \cap \{\mathbf{B}_2 \leftrightarrow \mathbf{B}_3\}$  zwischen den zwei Korrespondenzmengen ergab 34 Treffer. Die Wahrscheinlichkeit von fehlerbehafteten Korrespondenzen wird damit minimiert. Diesbezüglich wurden lediglich die Korrespondenzen aus der Schnittmenge für weitere Berechnungen (Triangulation etc.) verwendet.

Weiterführend konnten durch Filtermaßnahmen bezüglich Outlier, Merkmalsduplikate, Rückprojektionsfehler der triangulierten Merkmalskorrespondenzen u. a. 33 exakte Korrespondenzen ermittelt werden (Siehe Abbildung 8.13).



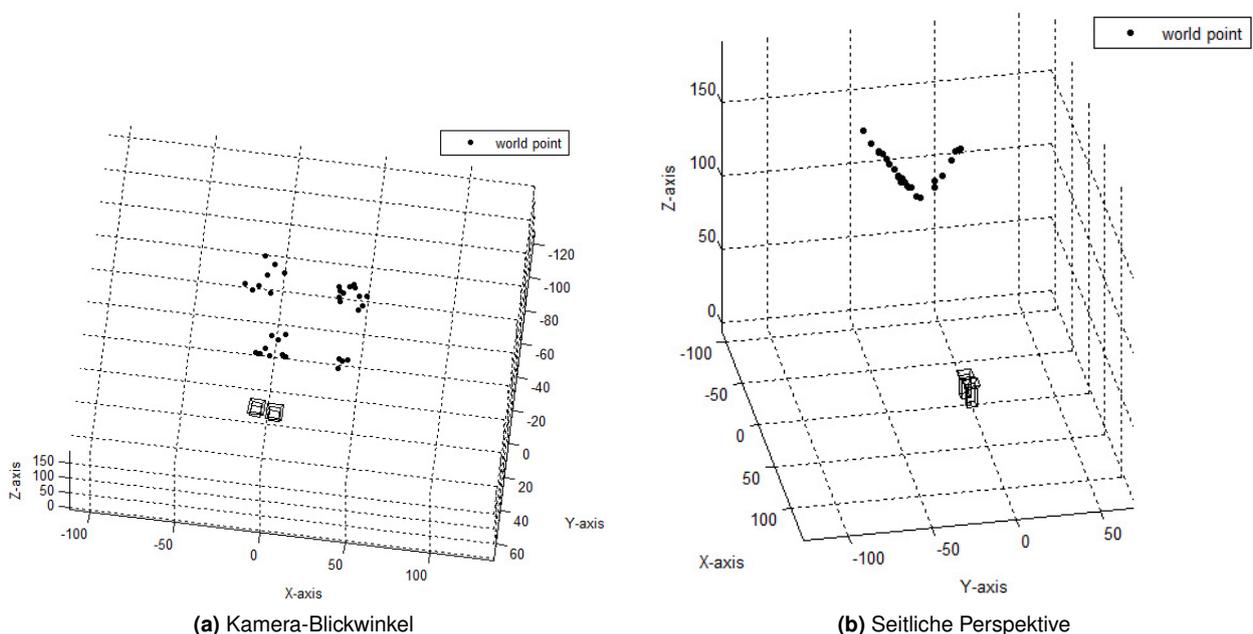
**Abbildung 8.13:** Ergebnis einer gefilterten Korrespondenzmenge mit Rückprojektionsanalyse in den ersten zwei Bildern.

Die fehlerfreien Korrespondenzen ermöglichen die Berechnung von 33 Oberflächenpunkten (3D-Weltpunkten). Im Anschluss der Triangulation sind die 3D-Punkte aus Sicherheitsgründen zu überprüfen. In der Abbildung 8.14 ist die Überprüfung dargestellt. Die Raumposition von einem 3D-Punkt darf im Verhältnis zu allen anderen 3D-Punkten nicht unrealistisch weit entfernt sein. Dafür wird ein Gültigkeitsbereich autonom berechnet, welcher im Kapitel 5.3.3.2 ab Seite 56 erläutert wurde.



**Abbildung 8.14:** Überprüfung der Oberflächenpunkte anhand eines Gültigkeitsbereiches.

Die beiden Abbildungen 8.15a und 8.15b stellen die Oberflächenpunkte grafisch dar. Der rechte Winkel in der Grafik 8.15b veranschaulicht eine gute Messgenauigkeit.



**(a)** Kamera-Blickwinkel

**(b)** Seitliche Perspektive

**Abbildung 8.15:** Ermittelte Oberflächenpunkte in den ersten zwei Bildern.

Die nachfolgende Tabelle 8.2 listet die Resultate der initialen Bestimmung von Weltpunkten auf. Anmerkend sei erwähnt, dass jedem Oberflächenpunkt jeweils ein Merkmal aus jedem Bild zugeordnet wird. Im derzeitigen 3D-Merkmalmodell (Oberflächenpunkte + Merkmale) besitzt damit jeder Oberflächenpunkt drei Merkmale bzw. Deskriptoren.

Verfahren	Aufnahme	Merkmale	Korrespondenzen	Oberflächenpunkte (3D-Weltpunkte)	Berechnungsdauer [Sek.]
SIFT	1,2	151	93 (Menge A)		
SIFT	2,3	149	82 (Menge B)		
SIFT	1,2,3	211	65 (Schnittmenge $A \cap B$ )	65	7.2
SURF	1,2	160	48 (Menge A)		
SURF	2,3	162	47 (Menge B)		
SURF	1,2,3	243	33 (Schnittmenge $A \cap B$ )	33	0.4

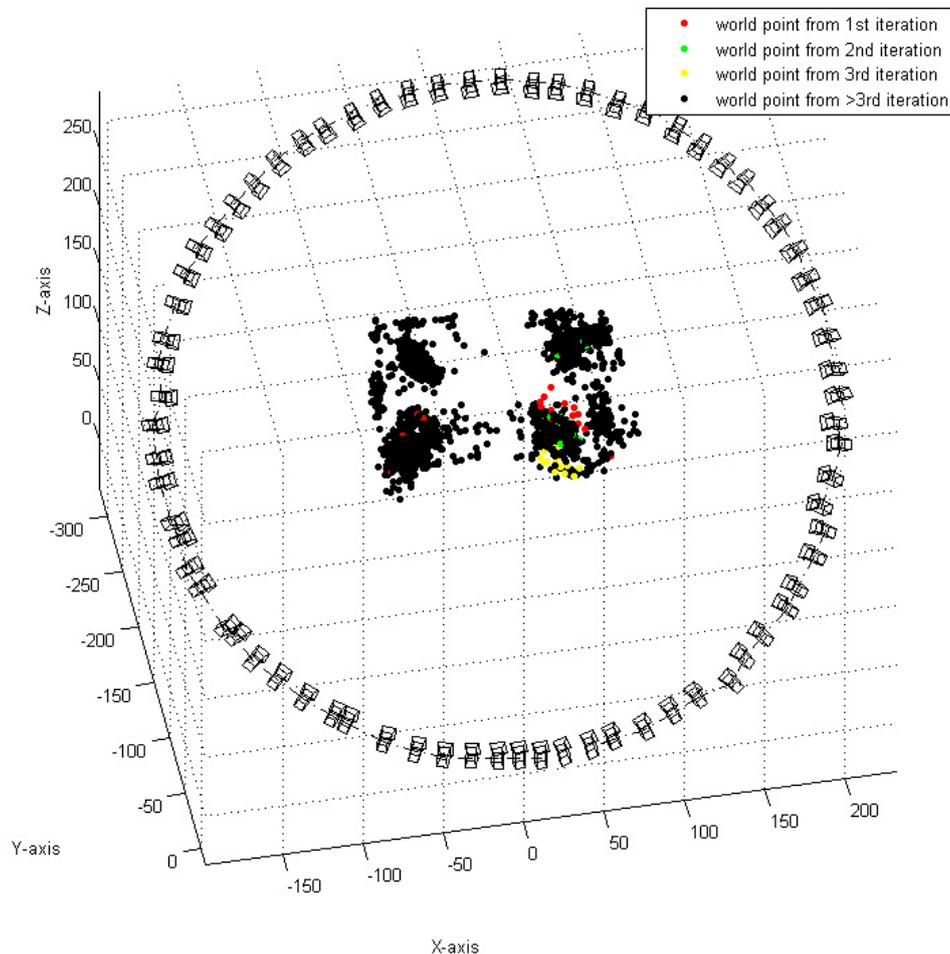
**Tabelle 8.2:** Messresultat der Papierschachtel.

Das aus den initialen Weltpunkten bestehende 3D-Merkmalmodell ist mit neuen Merkmalen sowie Weltpunkten zu erweitern. Dafür sind weitere 69 Aufnahmen analysiert worden. Mit den insgesamt 72 Aufnahmen konnten 1800 Weltpunkte und 5700 Merkmale ermittelt werden. Die Aufnahmen wurden mit der Drehtellerkamera (Vgl. Kapitel 4.4) in jeweils  $360^\circ / 72 = 5^\circ$  Schritten aufgenommen. Im Nachfolgenden sind in der Tabelle 8.3 die Messresultate der Modellgenerierung aufgelistet.

Verfahren	Aufnahmen	Merkmale	Oberflächenpunkte (3D-Weltpunkte)	Berechnungsdauer [Minuten]
SURF	72	5700	1800	0.45

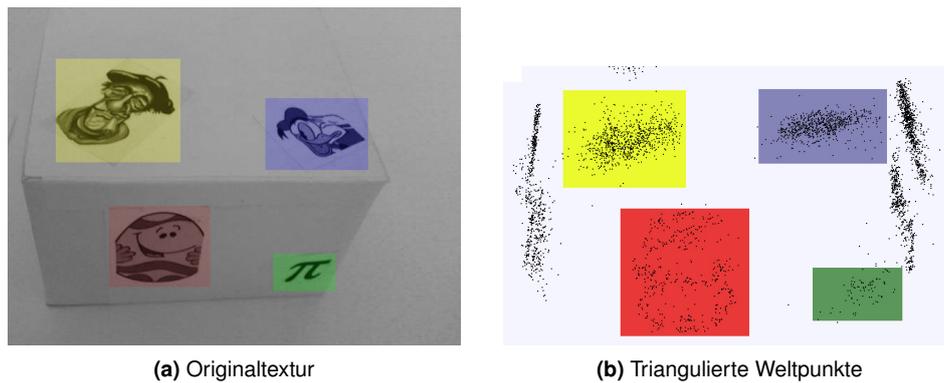
**Tabelle 8.3:** Kennzahlen der Modellgenerierung von einer Papierschachtel.

Die Abbildungen 8.16 veranschaulicht das Ergebnis der Modellgenerierung grafisch.



**Abbildung 8.16:** Draufsicht der triangulierten Weltpunktmenge aus insgesamt 72 Kameraaufnahmen.

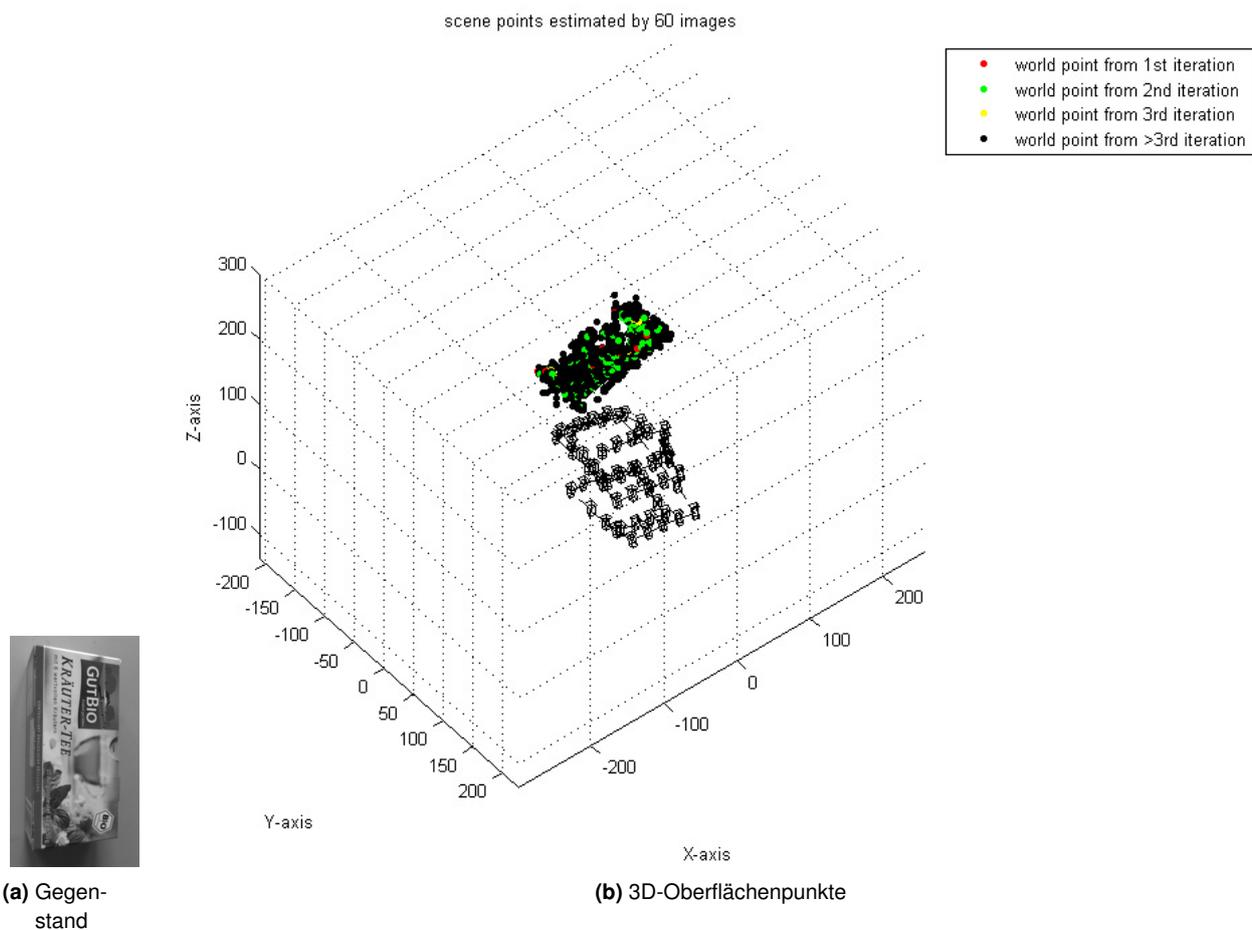
Eine Gegenüberstellung zwischen Bildsegmenten und den zugehörig triangulierten Oberflächenpunkten ist in der Abbildung 8.17 zu sehen.



**Abbildung 8.17:** Vergleich der rekonstruierten Oberflächenpunkte mit der zugehörigen Originaltextur.

### 8.1.3 Tee-Packung

Die Abbildung 8.18 zeigt das aus 60 Kameraaufnahmen entstandene 3D-Modell einer Tee-Packung.

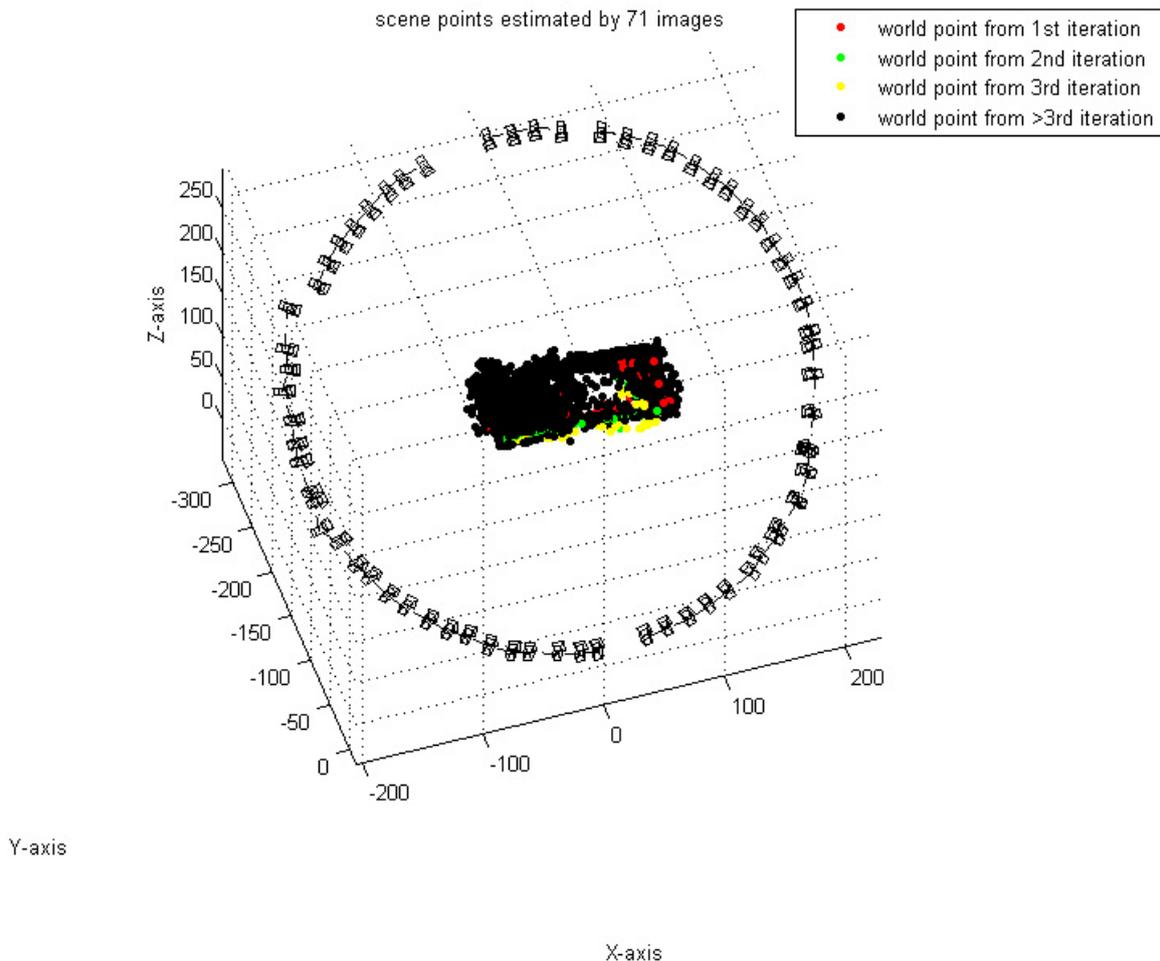


**Abbildung 8.18:** Oberflächenpunkte einer Tee-Verpackung.

Die benötigte Berechnungsdauer betrug 2,3 Minuten. Dabei wurden insgesamt 10300 3D-Oberflächenpunkte trianguliert. Zusammengefasst besitzen diese 27600 SURF-Merkmale. In der Grafik 8.18b sind die ermittelten 3D-Punkte, welche aus den ersten zwei Kameraaufnahmen (Erste Iteration) entstammen, rot dargestellt. Die

3D-Punkte aus den nächsten zwei Aufnahmen (Zweite Iteration) sind grün symbolisiert. Die Oberflächenpunkte der dritten Iteration sind gelb- und die der nachfolgenden Iterationen schwarz gekennzeichnet. Anhand der 60 dargestellten Kameras ist zu erkennen, dass mit dem Knickarm-Roboter eine Trajektorie abgefahren wurde und damit kein Drehteller zum Einsatz kam.

Unter der Verwendung eines Drehtellers wurde ein zweites 3D-Merkmalmodell von der Tee-Packung generiert. Die nächste Abbildung 8.19 stellt die 3D-Punkte und die Kamerapositionen mitsamt der Ausrichtungen grafisch dar. Der Drehteller wurde so angesteuert, dass in 5° Schritten eine Aufnahme mit der Roboterkamera (Bewegt sich nicht!) aufgenommen wird. Dies ist gut in der Abbildung zu erkennen.



**Abbildung 8.19:** Ermittelte Oberflächenpunkte unter Verwendung eines Drehtellers (Vogelperspektive).

Ersichtlicherweise ist der dargestellte Trajektorienkreis an mindestens zwei Stellen unterbrochen. An diesen Stellen sind die Kameraaufnahmen nicht dargestellt, weil die Merkmalsuche oder die Triangulation nicht *perfekte* Merkmalskorrespondenzen bzw. 3D-Punkte berechnet hat. Aus Sicherheitsgründen werden diese autonom verworfen.

Insgesamt konnten 6500 Oberflächenpunkte und 19200 SURF-Merkmale in einer Zeit von 1,2 Minuten berechnet werden. Es ist zu erkennen, dass sich die Anzahl der ermittelten Oberflächenpunkte und Merkmale im Verhältnis zum ersten 3D-Merkmalmodell verringert hat (Siehe S. 119). Das kann zum einen an der divergenten Lichtintensität während der Bildaufnahme liegen. Größtenteils ist der Unterschied jedoch an der Kameraentfernung zum Objekt zurückzuführen. Die vom Roboter abgefahrte Trajektorie befindet sich 7-40 cm vom Gegenstand entfernt (Vgl. Abbildung 8.18, S. 119). Dadurch ist die Texturfläche größer als die weit entfernten Aufnahmen am Drehteller. Denn hier ist nicht die Drehtellerkamera verwendet worden, sondern die Roboterkamera (Vgl. Kapitel 4.4). Aus diesem Grund wurde der Knickarm-Roboter in eine gewünschte Raumposition gefahren, sodass die Roboterkamera eine gute Sicht auf den Drehteller bzw. den Gegenstand besaß. Der räumliche Abstand zwischen der festpositionierte Roboterkamera und dem Drehtellermittelpunkt betrug in etwa 70 cm.

### 8.1.4 VHS-Kassette

In diesem Unterkapitel wird ein 3D-Merkmalmodell von einer VHS-Kassette vorgestellt. Besitzt ein Gegenstand viel Textur, können zahlreiche Merkmale detektiert werden. Aus diesem Grund konnten 16400 Oberflächenpunkte mitsamt 42000 SURF-Merkmale innerhalb von 3,3 Minuten berechnet werden. Die nachfolgenden Abbildungen 8.20 und 8.21 veranschaulichen dies.



Abbildung 8.20: VHS-Kassette.

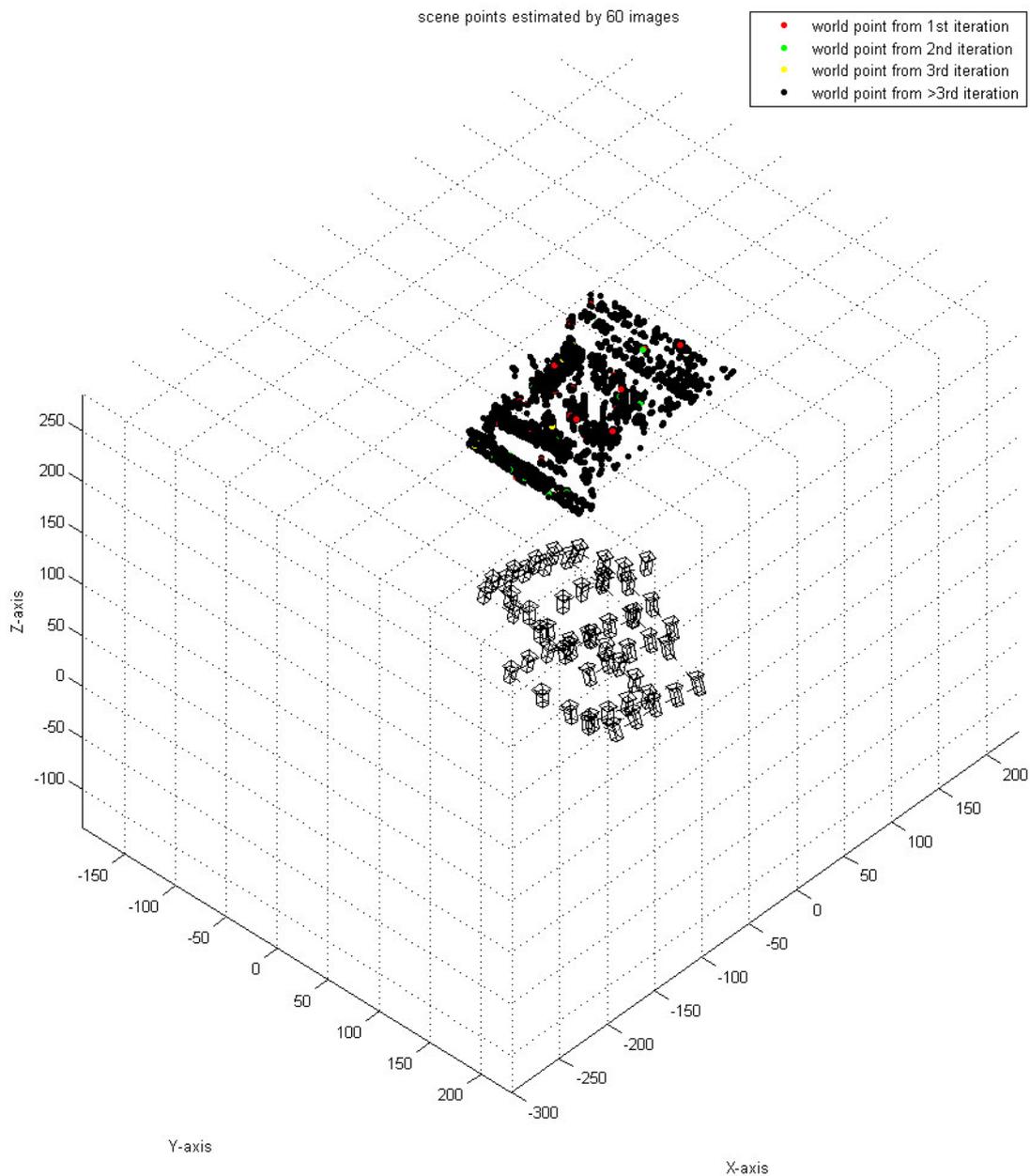


Abbildung 8.21: Berechnete Oberflächenpunkte einer VHS-Kassette.

Anmerkend zu den vorgestellten 3D-Merkmalmodellen kann über die Genauigkeit der berechneten 3D-Oberflächenpunkte folgendes ausgesagt werden. Bei jedem Gegenstand wurden vier markante räumliche Punkt-zu-Punkt Abstände gemessen. Der anschließende Vergleich mit den originalen Maßen ergab eine Genauigkeit im Millimeterbereich.

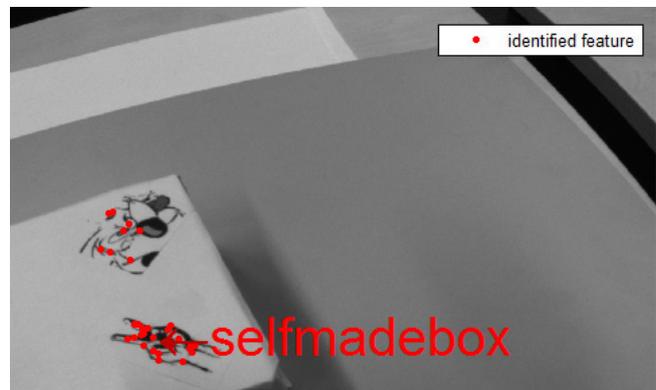
Die exakten Genauigkeitsangaben sind schwer zu bestimmen. In dieser Arbeit werden die Oberflächenpunkte lediglich aus den ermittelten Merkmalskorrespondenzen rekonstruiert. Aus diesem Grund werden markante Oberflächenpunkte, wie beispielsweise die acht äußeren Eckpunkte der quaderförmigen VHS-Kassette, nicht (exakt) berechnet. Dies zeigt auch, dass die Anzahl an robusten Merkmalen das Wichtigste für die Identifizierung und Posenbestimmung ist. Dagegen ist ein vollständiges 3D-Modell von einem Gegenstand deutlich unwichtiger.

## 8.2 Identifikation von Gegenständen

Die Identifizierung von Gegenständen wird durch einen Merkmalsvergleich zwischen den Merkmalen einer Kameraaufnahme und sämtlichen 3D-Merkmalmodellen erzielt. Eine Identifizierung eines Gegenstandes stellt die Abbildung 8.22 dar. Dort zeigt die Grafik (b) eine Kameraaufnahme, welche auf Merkmale untersucht wurde. Die detektierten Merkmale wurden mit den bestehenden 3D-Merkmalmodellen aller Gegenstände verglichen. Beim Merkmalsvergleich konnten 207 Korrespondenzen zwischen den Merkmalen der aktuellen Aufnahme und den Merkmalen des 3D-Merkmalmodells einer Papierschachtel gefunden werden. Dabei sind die 207 Merkmale 32 3D-Punkten zugeordnet. In der Grafik ist dies zu erkennen. Es sind 32 Bildpunkte rot markiert. Jeder rote Punkt kennzeichnet ein Merkmal, wobei sich typischerweise Merkmale mit leicht divergenten Deskriptor an der gleichen Bildposition befinden. Dies wurde zur verbesserten Wiedererkennung während der 3D-Merkmalmodellgenerierung anhand zahlreicher perspektivisch unterschiedlicher Kameraaufnahmen realisiert (Vgl. Kapitel 5.4 *Weltpunkte aus weiteren Aufnahmen*).



(a) Farbdarstellung der Papierschachtel



(b) Detektion in einem Bild

**Abbildung 8.22:** Detektion einer selbstgebauten Papierschachtel.

In der nachfolgenden Abbildung 8.23 wurden drei Gegenstände wiedererkannt. Eine VHS-Kassette mit 426-, ein Hustensaft mit 265- und eine Tee-Packung mit 376-Merkmal-korrespondenzen. Anmerkend sei erwähnt, dass sich zahlreiche Merkmale bzw. Merkmalskorrespondenzen mit leicht unterschiedlichen Deskriptoren an derselben Bildposition befinden.



Abbildung 8.23: Drei identifizierte Gegenstände in einem Kamerabild.

### 8.3 Mobilisierung von Gegenständen

An dieser Stelle werden vier experimentelle Ergebnisse vorgestellt. Als erstes sollte die Verpackung eines Hustensaftes gesucht, gegriffen und anschließend zu einem festgelegten Abgabeplatz gebracht werden. Die Verpackung ist 140 mm lang, 65 mm breit und hoch und besitzt mitsamt des Inhaltes ein Gewicht von 250 g. Die insgesamt acht Grafiken der Abbildung 8.24 veranschaulichen den Mobilisierungsvorgang der Hustensaftverpackung. Zu Beginn wurde die Objekt- und Zielpose (b) berechnet und von einer Startpose (a) aus angefahren. Die berechnete Zielpose entspricht der initial festgelegten Zielpose so exakt, dass keine Korrektur erfolgen musste. Aus diesem Grund wurde die Greifpose (e) als Nächstes angefahren. Anschließend konnte zu einer gewünschten Übergabeposition (h) gefahren werden. Dort kann eine Person den Gegenstand nach einer Aufforderung entgegennehmen.

Für die Mobilisierung musste lediglich eine Kameraaufnahme ausgewertet werden. Insgesamt hat sich der Knickarm-Roboter dreimal bewegt. Von der Start- zur Zielpose. Anschließend weiter zur Greifpose und abschließend zu dem gewünschten Abgabeplatz. Die benötigte Gesamtdauer betrug 4,7 s. Dabei wurde die Gelenkbewegung zur Zielpose und zum Abgabeplatz in einer limitierten Gelenkgeschwindigkeit<sup>36</sup> durchgeführt. Die Anfahrt von der Ziel- zur Greifpose wurde aus Sicherheitsgründen in 30% der limitierten Gelenkgeschwindigkeit ausgeführt.

Die Abbildung 8.25 stellt gleichermaßen wie bei der Hustensaft-Verpackung einen Such- und Bringvorgang dar. Eine Person möchte, dass der Knickarm-Roboter eine Tee-Packung sucht, greift und abschließend zureicht. Dabei ist die räumliche Pose des Gegenstandes wiederum unbekannt. Anhand einer einzigen Kameraaufnahme konnte der gesuchte Gegenstand in einem Bruchteil einer Sekunde identifiziert werden. Die anschließenden Posenbestimmungen des Gegenstandes und der -Zielpose (c) dauerte 0,8 s. Die Greifpose (e) konnte innerhalb von 0,5 s berechnet und angefahren werden. Die letzte Grafik (g) zeigt die Übergabeposition, wo der Gegenstand der Person übergeben wird.

<sup>36</sup>Mit der limitierten Gelenkgeschwindigkeit kann in eine Gelenkstellung gefahren werden, ohne dass an einem Gelenk ein Schaden bzw. ein erhöhter Verschleiß entsteht. Es ist jedoch möglich das 1,8-fache der limitierten Geschwindigkeit zu erreichen. Bei einer Überschreitung der limitierten Gelenkgeschwindigkeit muss allerdings die (End-) Beschleunigung/Geschwindigkeit beachtet werden, da sonst zu abrupt in eine Gelenkstellung gefahren wird.

Die Gesamtzeit des Such- und Bringvorganges betrug 5,3 s. Im Vergleich zur vorherigen Mobilisierung ist gut zu erkennen, dass die Identifizierung und Posenbestimmung trotz sehr unterschiedlicher Lichtverhältnisse ausgezeichnet funktioniert (Vgl. Abbildungen 8.24 und 8.25). Dies bestätigt die hohe Invarianz der Merkmale gegenüber Helligkeitsveränderungen. Hierbei wurden 128-dimensionale SURF-Merkmale verwendet (Vgl. Kapitel 5.2.2).



(a) Objektpose + Zielpose berechnen



(b) Zielpose angefahren



(c) Bewegung zur Greifpose



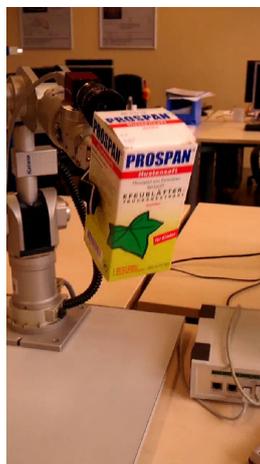
(d) Bewegung zur Greifpose



(e) Greifpose angefahren



(f) Abgabeplatz anfahren



(g) Abgabeplatz anfahren



(h) Übergabe

**Abbildung 8.24:** Mobilisierung einer Hustensaft-Verpackung.



(a) Objektpose + Zielpose berechnen



(b) Zielpose anfahren



(c) Zielpose erreicht



(d) Greifpose anfahren



(e) Greifpose erreicht



(f) Übergabeposition anfahren



(g) Übergabeposition erreicht

**Abbildung 8.25:** Such- und Bringvorgang einer Tee-Packung.

Das dritte experimentelle Ergebnis zeigt die nachfolgende Abbildung 8.26 mit insgesamt zwölf Grafiken. Es sollte eine Flasche (Unbekannte Pose) mit geringer Texturfläche und den damit resultierenden wenigen Merkmalen gegriffen werden, um anschließend einen Becher (Bekannte Pose) symbolisch mit Flüssigkeit zu befüllen. Es vergingen lediglich 4,5 s bis die Flasche vollständig gegriffen war. Dabei wurde ein Kamerabild analysiert und die Zielpose musste nicht korrigiert werden. Anschließend war der Becher mit einer festgelegten Bewegungsabfolge symbolisch mit Flüssigkeit zu befüllen. Nach dem Einschenkvorgang erfolgte eine Bewegung zum Übergabepunkt. Dort wird der Greifer durch eine Aufforderung geöffnet, sodass eine Person den Gegenstand entgegen nehmen kann.



(a) Objektpose + Zielpose berechnen



(b) Zielpose angefahren



(c) Bewegung zur Greifpose



(d) Greifpose erreicht



(e) Greifpose erreicht



(f) Einschenkpose anfahren



(g) Einschenkpose erreicht



(h) Einschenkvorgang



(i) Einschenkvorgang



(j) Einschenkvorgang



(k) Einschenkvorgang abgeschlossen

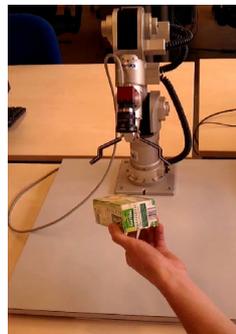


(l) Übergabeposition erreicht

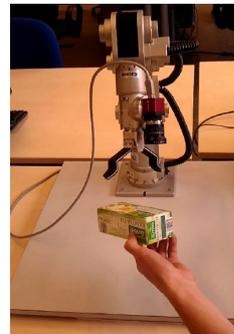
**Abbildung 8.26:** Posenbestimmung, Greifen und Bewegungsvorgänge einer Flasche.

Das vierte und letzte Beispiel zeigt keinen kompletten Mobilisierungs-Vorgang. Stattdessen soll lediglich ein Gegenstand verfolgt werden. Dies ist einfach zu realisieren, indem kontinuierlich die Zielpose berechnet und angefahren wird. Denn die Zielpose beschreibt den relativen räumlichen Versatz und die Orientierung zwischen dem Gegenstand und der Roboter-Kamera.

Die Abbildung 8.27 stellt eine Verfolgung eines Gegenstandes dar. Dabei wurde kontinuierlich die Zielpose (inkl. Objektpose) berechnet und angefahren. Dadurch passt sich der Roboter der Bewegung des Gegenstandes an.



(a) Objektpose + Zielpose berechnen



(b) Zielpose angefahren



(c) Objektpose + Zielpose berechnen



(d) Bewegung zur Zielpose



(e) Zielpose erreicht



(f) Objektpose + Zielpose berechnen



(g) Zielpose angefahren



(h) Objektpose + Zielpose berechnen



(i) Zielpose angefahren

**Abbildung 8.27:** Verfolgung eines Gegenstandes durch kontinuierliche Berechnung der Zielpose.

# 9 Zusammenfassung und Ausblick

Zunächst werden die wesentlichen Ergebnisse der vorliegenden Arbeit zusammengefasst. Anschließend folgen in einem Ausblick ein paar Weiterentwicklungsmöglichkeiten.

## Inhalt

---

9.1 Stand der Arbeit . . . . .	128
9.2 Weiterentwicklungen . . . . .	129

---

### 9.1 Stand der Arbeit

In dieser Arbeit wurde eine kamerabasierte Erkennung, Lokalisierung und Mobilisierung von Gegenständen beschrieben und untersucht.

Der dafür einzusetzende Knickarmroboter ist ein 4 kg leichter Katana-Greifarm. Dieser besitzt fünf Freiheitsgrade und kann Tätigkeiten ortsgebunden in einem Radius von 60 cm bewältigen. Des Weiteren ist für die Identifikation von Gegenständen eine Kamera notwendig. Diese wurde am Endeffektor befestigt und liefert Grauwertbilder zu einem nah positionierten Computer.

Das in dieser Arbeit beschriebene Verfahren basiert auf einem 3D-Merkmalmodell (Oberflächenpunkte + Merkmale) des zu identifizierenden und letztendlich zu greifenden Gegenstandes.

Die Generierung eines 3D-Merkmalmodells wurde im Kapitel 5 beschrieben und benötigt eine einmalige Kamerakalibrierung. Die aus der Kalibrierung gewonnenen intrinsischen Kameraparameter sind für die exakte Berechnung der Weltpunkte notwendig. Dabei entsprechen Weltpunkte den 3D-Oberflächenpunkten von einem Gegenstand.

Im Anschluss der Kalibrierung wurden im Kapitel 5.2 zwei Methoden zur Merkmalsextraktion vorgestellt. Dabei hat sich das Verfahren zur skaleninvarianten Merkmalstransformation (*Scale-Invariant Feature Transform*) als sehr robust herausgestellt. Dagegen wies das sogenannte *Speeded-Up Robust Features*-Verfahren einen beachtlichen Geschwindigkeitsvorteil auf. Nach der Merkmalsextraktion wurde die initiale Bestimmung von Weltpunkten erläutert. Diese gliedert sich in vier Phasen.

Am Anfang sind Merkmalskorrespondenzen zwischen Bildern zu finden. Anschließend ist eine Filterung erforderlich. Es müssen u. a. Duplikate, Fehlzugeordnungen oder eine degenerierte Merkmalsmenge erkannt und korrigiert werden. Die dritte Phase beschreibt wie Weltpunkte, anhand der Merkmalskorrespondenzen und den intrinsischen Kameraparametern, zu bestimmen sind. In einer letzten Phase müssen die gesamten Weltpunkte in ein 3D-Merkmalmodell abgelegt werden. Dabei sind zu jedem Weltpunkt die Merkmale, welche einen Weltpunkt eindeutig identifizieren, zugeordnet.

Im Anschluss an die initiale Bestimmung von Weltpunkten können weitere perspektivisch unterschiedliche Kameraaufnahmen einbezogen werden (Vgl. Kapitel 5.4). Dies führt zu einem noch detailgetreueren 3D-Merkmalmodell, wodurch die Wiedererkennungsrates eines Gegenstandes deutlich ansteigt.

Auf Basis des resultierenden 3D-Merkmalmodells konnte ein Gegenstand in einem beliebigen Bild identifiziert und seine räumliche Lage und Orientierung bestimmt werden. Mit dem Wissen über die Objektlage- und Ausrichtung sind dadurch Greif- und Bewegungsvorgänge an Gegenständen durchführbar (Vgl. Kapitel 6 und 7).

Zusammenfassend wurde ein autonomer Knickarm-Roboter entwickelt. Dieser kann Gegenstände, bei denen die Raumposition unbekannt ist, identifizieren, lokalisieren und mobilisieren. Diesbezüglich veranschaulicht das Kapitel 8 zahlreiche experimentelle Ergebnisse.

## 9.2 Weiterentwicklungen

Aufbauend auf den gewonnenen Erkenntnissen können weitere Projekte entstehen. In dieser Arbeit wird ein Gegenstand von einer festgelegten Greifpose gegriffen. Die Vorgehensweise ist für Gegenstände wie Tassen oder Flaschen gut geeignet, da diese nur in bestimmten Greifposen gegriffen werden dürfen. Dabei ist die Festlegung auf mehrere Greifposen ohne größeren Aufwand möglich.

Die Vorgehensweise sollte sich bei anderen Gegenständen verändern. Beispielsweise möchte eine Person einen Schlüssel, ein Portemonnaie oder ein Stift zugegriffen bekommen. Die Greifpose ist im beschriebenen Fall von untergeordneter Bedeutung. In dieser Situation kann die berechnete Objektpose herangezogen werden, um alle möglichen Greifposen autonom zu bestimmen. Dadurch ist ein Gegenstand nicht mehr von einer oder mehreren festgelegten Greifposen abhängig.

Des Weiteren sollten Druck- und Abstandssensoren am Greifer montiert werden. Diese würden den Greifvorgang vor allem bei empfindlichen Gegenständen aus Glas o. ä. sicherer machen.

# A Codebeispiele

In diesem Kapitel sind stark vereinfachte Codebeispiele aufgelistet. Der Code fand in dieser Arbeit Verwendung und ist in der Syntax von MATLAB dargestellt.

## Quellcode A.1: Bestimmung der Fundamentalmatrix mit dem 8-Punkte-Algorithmus

```
1 % normalisation
2 [x1 x1 T1 T2] = normalisepoints(x1,x2);
3
4 % constraint matrix
5 A = [x2(1,:)'.*x1(1,:) '   x2(1,:)'.*x1(2,:) '   x2(1,:) ' ...
6       x2(2,:)'.*x1(1,:) '   x2(2,:)'.*x1(2,:) '   x2(2,:) ' ...
7       x1(1,:) '             x1(2,:) '             ones(npts,1) ];
8
9 % singular value decomposition
10 [U,S,V] = svd(A,0);
11
12 % estimate the fundamental matrix with the smallest singular value
13 F = reshape(V(:,9),3,3)';
14
15 % repeat singular value decomposition
16 [U,S,V] = svd(F,0);
17
18 % eliminate the smallest value from the diagonal matrix of S
19 S = diag([S(1,1) S(2,2) 0]);
20
21 % recombine the improved fundamental matrix Fk
22 Fk = U * S * V';
23
24 % denormalise the fundamental matrix
25 F = T2' * Fk * T1;
```

## Quellcode A.2: RANdom SAMple Consensus (RANSAC)

```
1 % number of max. trials
2 N = 5;
3
4 % propability to take minimum one sample without outliers
5 p = 0.99;
6
7 % how many points we need to estimate the fundamental matrix F
8 k = 8;
9
10 % count of trials
11 trials = 0;
12
13 % best score of found inliers
14 maxscore = 0;
15
16
17 % break if the maximum of trials is reached
18 while N > trials
19
```

```

20     % take out k points
21     ind = randSelect(npts, k);
22
23     % fit the selected corresponding points into the given modell
24     M = getModell(fmatrix, x(:,ind));
25
26     % check inliers
27     [M inliers] = getInliers(M, x, alpha);
28
29     cntInliers = length(inliers);
30
31     % is this the largest set of inliers
32     if cntInliers > maxscore
33
34         maxscore = cntInliers;
35         maxinliers = inliers;
36         bestModell = M;
37
38         pOutliers = 1 - (cntInliers / npts)^k;
39
40         % refresh the number of max. trials N
41         N = log(1-p) / log(pOutliers);
42
43     end
44
45     % increment the trials
46     trials = trials + 1;
47
48 end

```

### Quellcode A.3: Ansteuerung einer Kamera

```

1 % show cameras which you have acces to (optional)
2 imagtool
3
4 % choose camera that you want to use (first parameter)
5 % and set the resolution (third parameter)
6 vid = videoinput('winvideo', 1, 'Y800_1280x960');
7
8 % set up the options
9 src = getselectedsource(vid);
10 vid.FramesPerTrigger = 1;
11 src.FrameRate = '7.5'; % or 3.75 | 7.5 | 15.0 | 30.0
12 src.ExposureMode = 'auto';
13 vid.ReturnedColorspace = 'rgb'; % or grayscale
14
15 % run preview
16 preview(vid);
17
18 % do snapshot
19 I = getsnapshot(vid);

```

**Quellcode A.4:** Berechnung einer Transformationsmatrix  $\mathbf{T}$ , welche die 3D-zu-3D-Abbildung  $\mathbf{x}_i^{camera} = \mathbf{T} \cdot \mathbf{X}_i$  (Im Allgemeinen  $\mathbf{X}_i = \mathbf{T} \cdot \mathbf{X}_j$ ) beschreibt

```

1 % syntax note
2 % cpnts: 3d-points (e.g. in the camera coordinate system)
3 % wpnts: 3d-points (e.g. in the world coordinate system)
4
5
6 % count of correspondings
7 n = size(cpnts,1);

```

```
8
9 % dimension of points
10 m = size(cpnts,2);
11
12 % calculate the arithmetic mean for each point amount
13 ccentroid = mean(cpnts);
14 wcentroid = mean(wpnts);
15
16 % refine camera- and worldpoints
17 for d=1:m
18     cpnts(:,d) = cpnts(:,d) - ccentroid(d) * ones(n,1);
19     wpnts(:,d) = wpnts(:,d) - wcentroid(d) * ones(n,1);
20 end
21
22 % build 3 x 3 matrix with the normalized camera- and worldpoints
23 % -> sum of correspondence multiplication
24 for i=1:n
25     M = M + cpnts(i,:) * wpnts(i,:);
26 end
27
28 % singular value decomposition on M
29 [U S V] = svd(M);
30
31 % estimate rotation matrix
32 R = U * V';
33
34 % check if rotation matrix has a negativ prefix
35 if det(R) < 0
36     R = -R;
37 end
38
39 % calculate translation vector
40 t = ccentroid' - R * wcentroid';
41
42 % build transformation matrix T := [ R | t ]
43 T(1:3,1:3) = R;
44 T(1:3,4) = t;
```

## **B Inhalt der beiliegenden DVD**

Dieser Arbeit liegt eine DVD mit dem nachfolgend aufgelisteten Inhalt bei.

### **\PDF**

Masterarbeit im PDF-Format.

### **\Ergebnisse**

Sammlung von zahlreichen Messresultaten in Form von Bildern, Screenshots und Videos.

### **\Software**

Dokumentierte entwickelte Software.

# Abbildungsverzeichnis

1.1	Assistenzroboter CoRA	1
1.2	Care-O-bot 3	2
1.3	Katana-Greifarm	3
1.4	Identifizierung, Lokalisierung und Greifen eines Gegenstandes	3
2.1	Typische künstliche Marker	6
2.2	Relative Posenbestimmung mit künstlichen Markern	7
2.3	Regelkreisbasierte Posenbestimmung	8
2.4	Simulierter Ablauf des 2D Visual Servoing Verfahrens	9
3.1	Kamera-, Welt- und Endeffektorkoordinatensystem	12
3.2	Beziehung zwischen zwei Bildern	14
3.3	Epipolarebenen zweier Bilder bei Veränderung eines Raumpunktes	14
4.1	Dimensionen des Greifarmes	19
4.2	Nutzlastschwerpunkt des Greifarmes	20
4.3	Nutzlastdiagramm des Greifarmes	20
4.4	Arbeitsraum des Greifarmes	20
4.5	Katana-Greifarm mit fünf Gelenken und einem Zwei-Finger-Greifer	21
4.6	Maße und Verfahrenswinkel des Winkelgreifers	21
4.7	Kontroller zur Ansteuerung des Roboters (Frontansicht)	22
4.8	Rückseite des Steuerungskontrollers	22
4.9	CCD-Kamera in Kombination mit einem Drehteller	23
4.10	Sony Ericsson C905 Handykamera	24
4.11	Am Katana-Greifarm befestigte FireWire-Kamera	24
4.12	Drehmesstisch und Positionierungssteuerung	25
5.1	Verzeichnungsarten einer Kamera	27
5.2	Zentralprojektion	28
5.3	Ebenes Kalibriermuster	30
5.4	20 unterschiedliche Aufnahmen vom Kalibriermuster	30
5.5	Gefundene Eckpunkte des Schachbrettmusters	31
5.6	Weltansicht der Kameraposen	31
5.7	Tangentiale und radiale Verzeichnung der Roboterkamera	33
5.8	Rotation einer 16 x 16 Pixel großen Merkmalsumgebung anhand seiner Hauptorientierung	34
5.9	Ermittlung der Extrema im Skalenraum	35
5.10	Potentieller Merkmalspixel wird mit seinen 26 Nachbarn verglichen	36
5.11	Filterung von Merkmalen, welche sich auf Kanten befinden	37
5.12	Richtungsbestimmung eines Merkmales mit einem Orientierungshistogramm	38
5.13	2x2x8-dimensionaler Deskriptor eines Merkmales	38
5.14	Abstrakter 2x2x8-dimensionaler Deskriptor	39
5.15	Ablauf einer Merkmalsextraktion	40
5.16	Extrahierte SIFT-Merkmale	42
5.17	Extrahierte SURF-Merkmale	42
5.18	SURF-Korrespondenzen zwischen den ersten- und den nächsten zwei Kameraaufnahmen	44
5.19	Schnittmenge zwischen zwei Korrespondenzmengen	45

5.20	Aufbau eines k-d Baumes mit einer zweidimensionalen Punktmenge . . . . .	46
5.21	Suchablauf in einem k-d Baum . . . . .	46
5.22	Fehlzuordnung zwischen zwei Merkmalen . . . . .	48
5.23	Struktur- und Bewegungsdegeneration . . . . .	49
5.24	Ineffiziente Methode zur Erkennung einer Strukturdegeneration . . . . .	49
5.25	Die Berechnung der zweiten Projektionsmatrix besitzt vier Lösungen . . . . .	54
5.26	Projektionsstrahlen schneiden sich nicht in einem gemeinsamen Weltpunkt . . . . .	54
5.27	Korrektur der Bildpunkte . . . . .	55
5.28	Ermittelte 3D-Weltpunkte . . . . .	56
5.29	Detektion falsch triangulierter Weltpunkte . . . . .	57
5.30	Filterung von Weltpunkten anhand des Rückprojektionsfehlers . . . . .	58
5.31	Binärkodierter Frame-Marker . . . . .	58
5.32	Binärkodiertes Markerfeld mit bekannter Geometrie . . . . .	59
5.33	Marker und natürliche Korrespondenzen . . . . .	60
5.34	Detektion von Markern trotz Teilverdeckung . . . . .	61
5.35	Aufbaustruktur der Modelldatenbank . . . . .	62
5.36	Merkmalsvergleich zwischen einem Bild und einem 3D-Merkmalsmodell . . . . .	63
5.37	Merkmalsvergleich zwischen zwei Bildern . . . . .	63
5.38	Suche nach Merkmalskorrespondenzen zwischen zwei Bildern . . . . .	65
5.39	Verifikation von Merkmalskorrespondenzen . . . . .	65
5.40	Korrektur von neuen Weltpunkten relativ zu den bestehenden Weltpunkten des 3D-Merkmalsmodells . . . . .	67
5.41	Räumliche Korrektur von neuen Weltpunkten . . . . .	68
6.1	Zusammenhang des Welt-, Bild- und Kamerakoordinatensystems . . . . .	70
6.2	Geschwindigkeits- und Genauigkeitsvergleich zwischen iterativen-, nicht-iterativen- und dem EPnP-Verfahren . . . . .	76
6.3	Virtuelle Kontrollpunkte . . . . .	76
6.4	Zusammenhang zwischen Bild- und Kamerakoordinate . . . . .	78
6.5	Umformen des linearen Gleichungssystems zu optimalen Matrix- und Vektorgrößen . . . . .	80
6.6	Bewahrung der Kontrollpunktabstände . . . . .	81
6.7	Abstand zwischen zwei Kontrollpunkten . . . . .	82
6.8	Korrektur der Kontrollpunkte mit dem Gauß-Newton-Algorithmus . . . . .	85
6.9	Ermittlung der Kontrollpunkte . . . . .	85
6.10	Berechnung der Kamerapunkte . . . . .	86
6.11	Bestimmung der Kamerapose durch 3D-Punktkorrespondenzen . . . . .	86
7.1	Initial festzulegende Zielpose, Zielansicht und Greifpose eines Gegenstandes . . . . .	89
7.2	Ablauf einer Objektidentifikation auf Basis von Merkmalen . . . . .	90
7.3	Identifizierte Gegenstände in einem Kamerabild . . . . .	91
7.4	Trajektorie ausgehend der Startausrichtung . . . . .	91
7.5	Transformation zwischen Koordinatensystemen (K.S.) . . . . .	92
7.6	Basis- $\mapsto$ Endeffektorkoordinatensystem $\mathbf{T}_{base}^{eff}$ . . . . .	92
7.7	Rechte-Hand-Regel . . . . .	93
7.8	Koordinatensystem nach der Denavit-Hartenberg-Konvention . . . . .	93
7.9	Festlegung der (Gelenk-) Koordinatensysteme . . . . .	94
7.10	Homogene Transformationen zwischen Koordinatensystemen . . . . .	94
7.11	Kinematische Grundstellung definiert durch Denavit-Hartenberg-Parameter . . . . .	96
7.12	Transformation zum Kamerakoordinatensystem der Roboterkamera . . . . .	97
7.13	Objekt- $\mapsto$ Kamerakoordinatensystem $\mathbf{T}_{obj}^{cam}$ . . . . .	98
7.14	Visualisierung der Koordinatensystem-Transformationen . . . . .	99
7.15	Darstellung der Koordinatensystem-Transformationen . . . . .	99
7.16	Erneute Berechnung der Zielpose . . . . .	100
7.17	Trajektorie mit $s$ Trajektorienschritten . . . . .	101

7.18 Direkte Berechnung des ersten Gelenks anhand der $x$ - und $y$ -Endeffektorkoordinate . . . . .	102
7.19 Trajektorie-basierte inverse Kinematik . . . . .	106
7.20 Objektmobilisierung mit dem Knickarm-Roboter . . . . .	107
7.21 Prinzip der Objektmobilisierung . . . . .	108
8.1 Testobjekt . . . . .	109
8.2 Markierung eines Merkmales mit dem Mauszeiger . . . . .	110
8.3 Subpixelgenaue Korrektur der selektierten Ecke . . . . .	110
8.4 Korrespondenzmengen . . . . .	111
8.5 Arithmetischer Mittelwert- und Median der triangulierten Punktmenge . . . . .	112
8.6 Filterung von fehlerhaft triangulierten Weltpunkten . . . . .	113
8.7 Rückprojektion der Weltpunkte zu den ursprünglichen Bildkoordinaten . . . . .	113
8.8 Resultierende Weltpunkte nach Durchführung diverser Filtermaßnahmen . . . . .	114
8.9 Räumliche Korrektur von neuen Weltpunkten . . . . .	114
8.10 Erzielte Weltpunktmenge der Testbox . . . . .	115
8.11 Detektierte SURF-Merkmale der Papierschachtel . . . . .	116
8.12 Merkmalsvergleich zwischen den ersten beiden- und den nächsten beiden Aufnahmen . . . . .	116
8.13 Ergebnis einer gefilterten Korrespondenzmenge mit Rückprojektionsanalyse in den ersten zwei Bildern . . . . .	116
8.14 Überprüfung der Oberflächenpunkte anhand eines Gültigkeitsbereiches. . . . .	117
8.15 Ermittelte Oberflächenpunkte in den ersten zwei Bildern . . . . .	117
8.16 Draufsicht der triangulierten Weltpunktmenge aus insgesamt 72 Kameraaufnahmen . . . . .	118
8.17 Vergleich der rekonstruierten Oberflächenpunkte mit der zugehörigen Originaltextur . . . . .	119
8.18 Oberflächenpunkte einer Tee-Verpackung . . . . .	119
8.19 Ermittelte Oberflächenpunkte unter Verwendung eines Drehtellers (Vogelperspektive) . . . . .	120
8.20 VHS-Kassette . . . . .	121
8.21 Berechnete Oberflächenpunkte einer VHS-Kassette . . . . .	121
8.22 Detektion einer selbstgebauten Papierschachtel . . . . .	122
8.23 Drei identifizierte Gegenstände in einem Kamerabild . . . . .	123
8.24 Mobilisierung einer Hustensaft-Verpackung . . . . .	124
8.25 Such- und Bringvorgang einer Tee-Packung . . . . .	125
8.26 Posenbestimmung, Greifen und Bewegungsvorgänge einer Flasche . . . . .	126
8.27 Verfolgung eines Gegenstandes durch kontinuierliche Berechnung der Zielpose . . . . .	127

# Tabellenverzeichnis

3.1	Notwendige Anzahl an Iterationen in Abhängigkeit des Ausreißeranteils . . . . .	18
3.2	Zur Schätzung der Fundamentalmatrix verwendete RANSAC-Parameter . . . . .	18
4.1	Verfahrwinkel der Gelenke . . . . .	21
5.1	Intrinsische Parameter der Sony Ericsson C905 Handykamera . . . . .	32
5.2	Intrinsische Parameter der Drehtellerkamera . . . . .	32
5.3	Intrinsische Parameter der Katana-Roboterkamera . . . . .	32
5.4	Extrahierte Merkmale und benötigte Berechnungsdauer mit SIFT/SURF . . . . .	42
5.5	Korrespondenzanalyse der ersten drei Aufnahmen . . . . .	45
5.6	Duplikate der Aufnahme 1 und 2 . . . . .	48
7.1	Statistische Form der Korrespondenzanalyse . . . . .	90
7.2	Denavit-Hartenberg-Parameter . . . . .	96
8.1	Messresultat der Testbox . . . . .	115
8.2	Messresultat der Papierschachtel . . . . .	118
8.3	Kennzahlen der Modellgenerierung von einer Papierschachtel . . . . .	118

# Algorithmenverzeichnis

- 1 Normalisierter 8-Punkte-Algorithmus . . . . . 17
- 2 RANdom SAmples Consensus (RANSAC) . . . . . 18

# Abkürzungsverzeichnis

BBF .....	Best-Bin-First
bzgl. ....	bezüglich
bzw. ....	beziehungsweise
CCD .....	Charge-Coupled Device
CPU .....	Central Processing Unit
DDR .....	Double Data Rate
DGPS .....	Differential Global Positioning System
DLT .....	Direct Linear Transformation
DoG .....	Difference of Gaussian
EPnP .....	Efficient Perspective n-Point
ff. ....	folgende / fortfolgende
GNU .....	GNU's Not Unix
GPS .....	Global Positioning System
GPU .....	Grafikprozessor (engl. Graphics Processing Unit)
IP .....	Internet Protocol
K. S. ....	Koordinatensystem
LDA .....	Diskriminanzanalyse (engl. Linear Discriminant Analysis)
o. a. ....	oder andere
PC .....	Personal Computer
PCA .....	Hauptkomponentenanalyse (engl. Principal Component Analysis)
PnP .....	Perspective n-Point
POSIT .....	Pose from Orthography and Scaling with Iterations
RAM .....	Random-Access Memory
ROI .....	Region of Interest
S. ....	Seite
Sek. ....	Sekunde(n)
SIFT .....	Skaleninvariante Merkmalstransformation (engl. Scale-Invariant Feature Transform)
SURF .....	Speeded-Up Robust Features
SVD .....	Singulärwertzerlegung (engl. Singular Value Decomposition)
TCP .....	Tool Center Point
u. a. ....	unter anderen / unter anderem
u. w. ....	und weitere
u. ä. ....	und ähnliches
USB .....	Universal Serial Bus
z. B. ....	zum Beispiel
UWB .....	Ultra-Breitband-Technologie (engl. Ultra Wide Band)

# Formelzeichen

'	.....	Punkte/Linien-Korrespondenz oder Inverse einer Matrix / eines Vektors
*	.....	Faltungsoperator
$\alpha$	.....	(Verstärkungs-) Faktor / Denavit-Hartenberg-Parameter
$\beta$	.....	Koeffizient
<b>A</b>	.....	Matrix mit normierten Punktkorrespondenzen
<b>a</b>	.....	Auflösung einer Kamera
<b>B</b>	.....	Bild
<b>C</b>	.....	Projektionszentrum
$\mathbf{c}^{\text{camera}}$	.....	Kontrollpunkt im Kamerakoordinatensystem
$\mathbf{c}^{\text{world}}$	.....	Kontrollpunkt im Weltkoordinatensystem
<b>d</b>	.....	Abstand zweier Punkte
<b>E</b>	.....	3 x 3 Essentialmatrix von zwei Bildern
<b>e</b>	.....	Epipol
<b>F</b>	.....	3 x 3 Fundamentalmatrix von zwei Bildern
<b>f</b>	.....	Bildweite in $x$ und $y$ -Richtung
<b>H</b>	.....	Hesse-Matrix
<b>h</b>	.....	Bildhauptpunkt
<b>I</b>	.....	Einheitsmatrix
$\mathbf{J}^+$	.....	Pseudoinverse der Jakobimatrix <b>J</b>
<b>J</b>	.....	Jakobimatrix
<b>K</b>	.....	3 x 3 Kamera-Kalibrierungsmatrix
$\mathbf{k}_{1 \times 5}$	.....	Drei radiale- und zwei tangentielle Linsenverzeichnungs-Koeffizienten
<b>L</b>	.....	6 x 3 Matrix, 2D- oder 3D-Abstand
<b>l</b>	.....	Epipolarlinie
<b>M</b>	.....	Matrix / Menge an Vektoren / Faktorisierung einer Matrix in drei Matrizen: $\mathbf{M} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$
$\mathbf{m}_{1 \times 12}$	.....	Beinhaltet vier 3D-Kontrollpunkte $\mathbf{c}_{j,1 \times 3}^{\text{camera}}$ für $j=1, \dots, 4$
<b>O</b>	.....	Ursprung des Schachbrettmusters
<b>o</b>	.....	Ursprung
<b>P</b>	.....	3 x 4 Projektionsmatrix zur perspektivischen Abbildung
<b>Q</b>	.....	Warteschlange
<b>R</b>	.....	3 x 3 Rotationsmatrix bzw. 1 x 3 Rotationsvektor bei einer RPY-Winkelkonvention
<b>S</b>	.....	Diagonalmatrix
<b>s</b>	.....	Auf $[0 \dots 1]$ normalisierte Trajektorien-schritte als $1 \times n$ Vektor
<b>T</b>	.....	3 x 4 oder 3 x 3 Transformationsmatrix
<b>t</b>	.....	Translationsvektor
<b>v</b>	.....	Vektor
<b>X</b>	.....	(Welt-) Punkt $[X, Y, Z]^T$ im Weltkoordinatensystem
<b>x</b>	.....	(Bild-) Punkt $[x, y]^T$ im Bildkoordinatensystem
$\mathbf{x}^*$	.....	Soll-Größe des Regelkreises bzw. die zu <b>x</b> korrespondierende Bildpunkte

$\mathbf{x}^{\text{camera}}$	(Kamera-) Punkt $[x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}}]^T$ im Kamerakoordinatensystem
$\Delta\varphi$	Gelenkveränderung in $^\circ$
$\Delta\mathbf{h}$	Bildhauptpunkt-Verschiebung in $x$ und $y$ -Richtung
$\varepsilon$	Eine sehr kleine Zahl, jedoch größer als null
$\eta$	Gewichtungskomponente bzw. homogene baryzentrische Koordinate
$\hat{\mathbf{f}}_{1 \times 9}$	1 x 9 Vektor mit den Matrizenelementen der Fundamentalmatrix $\mathbf{F}$
$\wedge$	Normalisierter bzw. Veränderter Wert
$\Lambda$	Eigenwert
$\lambda$	Skalierungskoeffizient
$\mathbb{P}^2, \mathbb{P}^3$	Projektive Ebene und projektiver Raum
$\nu$	Schwellwert
$\phi$	RPY-Winkelgröße: Rotation um die $x$ -Achse
$\psi$	RPY-Winkelgröße: Drehung um die $y$ -Achse
$\sigma$	Pyramidenstufe bzw. Skalierung des Bildes
$\theta$	Winkelgröße / Denavit-Hartenberg-Parameter
$\theta(x, y)$	Gradientenorientierung an der Bildposition $[x, y]$
$\tilde{\mathbf{u}}$	Rückprojizierter Weltpunkt in das Bildkoordinatensystem
$\times$	Kreuzprodukt
$\varphi$	Winkelgröße
$\vartheta$	RPY-Winkelgröße: Rotation um die $z$ -Achse
$T$	Transponierte einer Matrix / eines Vektors
arith	Arithmetischer Mittelwert einer Punktmenge
centroid	Schwerpunkt einer Punktmenge
distance	2D/3D-Punktabstand
$a$	Denavit-Hartenberg-Parameter
$\text{atan}$	Arkustangens-Funktion
$b$	Index
$B(x, y)$	Farb- bzw. Helligkeitswert an der Bildposition $[x, y]$
$\cos$	Cosinusfunktion
$D$	Zweifache Ableitung des Differenzbildes
$d$	Singulärwert / Denavit-Hartenberg-Parameter
$\det(\mathbf{A})$	Determinante der Matrix $\mathbf{A}$
$\text{diag}$	Überschreiben der Singulärwerte einer Diagonalmatrix
$\text{dist}$	Punktabstand
$e$	Abweichung, Fehler oder Exponentialfunktion
$\text{eig}$	Eigenwerte einer Matrix
$\text{error}$	Fehler einer Minimierungsfunktion
$F$	Faltungsfunktion der zweiten partiellen Ableitung der Gauß-Funktion
$\text{fkine}$	Funktion berechnet die Vorwärtskinematik
$G$	Gauß-Funktion
$i$	Index
$\text{inv}$	Inverse einer Matrix / eines Vektors
$j$	Index
$k$	Index
$L(\mathbf{x}, \mathbf{l})$	Abstandsfunktion, welche den senkrechten euklidischen Abstand des Punktes $\mathbf{x}$ zu seiner Epipolarlinie $\mathbf{l}$ berechnet

---

$L(x,y)$ .....	Extremwert an der Bildposition $[x,y]$
$\log$ .....	Natürlicher Logarithmus
$M$ .....	Ein Modell das im RANSAC-Algorithmus zur <i>Inlier</i> -Schätzung herangezogen wird
$m(x,y)$ .....	Gradientenlänge an der Bildposition $[x,y]$
$n,N$ .....	Natürliche Zahl repräsentative zur Angabe einer Länge oder Größe
$p$ .....	Index / Wahrscheinlichkeitsangabe
$s$ .....	Scherungsfaktor
$scale$ .....	Skalierungsfaktor
$\sin$ .....	Sinusfunktion
$svd(\mathbf{M})$ .....	Singulärwertzerlegung der Matrix $\mathbf{M}$
$t$ .....	Trajektorienstritte von der Start- zur Endposition
$\tan$ .....	Tangensfunktion
$v$ .....	Kamerageschwindigkeit
$w$ .....	Projektiver Koeffizient
$(x_{\text{eff}}, y_{\text{eff}}, z_{\text{eff}})$ .....	Punkt im Endeffektorkoordinatensystem
$(x_{\text{g}}, y_{\text{g}}, z_{\text{g}})$ .....	Punkt im Gelenkkoordinatensystem
$(x_{\text{obj}}, y_{\text{obj}}, z_{\text{obj}})$ .....	Punkt im Objektkoordinatensystem
$\min$ .....	Minimierungsaufgabe / Minimierungsfunktion

# Index

3D-Merkmalmodell .....	62
Erweiterung .....	69
8-Punkte-Algorithmus	
Denormalisierung .....	16
Lineares homogenes Gleichungssystem ...	16
Metrische Transformationsmatrix T .....	15
Normalisierung .....	16
Rekombinierung der Fundamentalmatrix ...	16
Singularitätsbedingung .....	16
<b>A</b>	
Absolute Orientierung .....	86
Rotationsanteil R .....	87
Transformationsmatrix T .....	87
Translationsanteil t .....	87
Arithmetischer Mittelwert .....	50, 56, 68, 86
Armlängen des Roboters .....	93
<b>B</b>	
Bündelblockausgleich .....	54, 66
Betrag des Gradienten .....	37
Bildweite .....	27
<b>D</b>	
Degeneration .....	48
Bewegung .....	48
Struktur .....	48
Denavit-Hartenberg	
Kinematische Kette .....	96, 103
Konvention .....	93
Parameter .....	93
Transformation .....	93
Determinante der Hesse-Matrix .....	41
Direkte Kinematik .....	92, 103
DLT	
Denormalisierung .....	74
Initiale Projektionsmatrix .....	74
Normalisierung	
Arithmetischer Mittelwert .....	73
Bildkoordinaten .....	74
Schwerpunkt der Bildkoordinaten .....	73
Schwerpunkt der Weltkoordinaten .....	73
Schwerpunktabstände .....	73
Skalierung .....	73
Weltkoordinaten .....	74
Singulärwertzerlegung .....	74
Drehmesstisch .....	25
Duplikate .....	47
<b>E</b>	
Epipolarlinie .....	15
EPnP	
Baryzentrische Koordinaten .....	77
Gewichtung .....	77
Initiale Kontrollpunkte .....	78
Intrinsische Kameraparameter .....	78
Kontrollpunkte	
Abstände .....	81
Der Bildkoordinaten .....	77
Der Weltkoordinaten .....	77
Schwerpunkt .....	77
Linearkombinationen .....	80
Minimierungsgleichung .....	85
Neularisierung .....	83
Null-Eigenvektor .....	80, 85
Projektive Koeffizienten .....	79
Rückprojektionsfehler	
Distanzfunktion .....	80
Rückprojektion .....	81
Substitution .....	79
Essentialmatrix .....	51
Hilfsmatrix .....	52
Kreuzprodukt .....	52
Rotationsmatrix .....	53
Schiefsymmetrische Matrix .....	53
Singulärwertzerlegung .....	52
Zerlegung .....	52
Euklidischer Abstand zwischen zwei Vektoren ...	43
<b>F</b>	
Faltung mit einem Gauß-Filter .....	35
Filterung .....	47, 66
Rückprojektion .....	57
Flansch .....	19
Fundamentalmatrix .....	14
<b>H</b>	
Hesse-Matrix .....	36, 41

- I**
- Identifizierung ..... 90
  - Inlier ..... 48
  - Intrinsische Kameraparameter ..... 28
  - Inverse Kinematik ..... 100
- J**
- Jacobimatrix ..... 104
- K**
- Kamerakalibrierung ..... 26
  - Knickarm ..... 19
  - Koordinatensystem
    - Rechtshändig ..... 93
    - Robotergelenke ..... 12
  - Korrektur der Linsenverzeichnung ..... 28
  - Korrespondenzen ..... 43
  - Korrigierter Pixelwert ..... 29
- L**
- Linsenverzeichnung
    - Korrektur ..... 28
    - Radialsymmetrisch ..... 27
    - Tangential ..... 28
  - Lokalisierung ..... 70, 91, 106
- M**
- Marker ..... 6
  - Matrix
    - Invertierung ..... 75
    - Unitär ..... 13
  - Median ..... 45, 56
  - Merkmal ..... 33
    - Extraktion ..... 33
    - Korrespondenzen ..... 43
    - SIFT ..... 34
    - SURF ..... 41
  - Mobilisierung ..... 107
- O**
- O-Notation ..... 75
  - Oberflächenpunkte ..... 66
  - Outlier ..... 48
- P**
- Posenbestimmung ..... 70, 91, 106
    - Nicht-Iteratives Verfahren ..... 72, 75
    - Transformationsmatrix ..... 70
  - Projektionsmatrix ..... 51, 66
    - Direkte Transformation der Weltpunkte ..... 54
    - Negative Lösung ..... 53
    - Positive Lösung ..... 53
    - Rotation und Translation ..... 52
    - Skalierungskorrektur ..... 62
    - Transformation ..... 52
  - Projektionsmatrix der Kamera
    - Mit Kameramatrix ..... 52
    - Ohne Kameramatrix ..... 51
  - Prozentualer Anteil ..... 57
  - Pseudoinverse ..... 104
  - Punkt auf Epipolarlinie ..... 15
  - Punktabstand ..... 58
- R**
- Rückwärtskinematik ..... 100, 107
  - Radialsymmetrische Linsenverzeichnung ..... 27
  - RANSAC ..... 18
  - Rechte-Hand-Regel ..... 93
  - Region of Interest (ROI) ..... 110
  - Relative Orientierung ..... 89
  - Richtung des Gradienten ..... 37
  - Roboterarm ..... 19
  - RPY-Winkelkonvention ..... 89
- S**
- Singulärwertzerlegung ..... 13, 53
  - Skalierung ..... 58
  - Standardabweichung ..... 45
  - Steuerungskontroller ..... 22
  - Strahlensatz ..... 28
- T**
- Tangentiale Linsenverzeichnung ..... 28
  - Transformation ..... 89
  - Transformationsmatrix ..... 89
  - Triangulation ..... 54, 66
    - Minimierung
      - Geometrischer Fehler im Bild ..... 55
      - Rückprojektionsfehler ..... 55
  - Triangulierung ..... 54, 66
- U**
- Unitäre Matrix ..... 13
- V**
- Varianz ..... 45, 50
  - Vorwärtskinematik ..... 92, 103
- W**
- Weltpunkte ..... 51, 66
    - Aus weiteren Aufnahmen ..... 63

Initiale Bestimmung .....	42
Skalierungskorrektur .....	58, 62, 67

**Z**

Zeitaufwand .....	75
Zentralwert .....	56

# Quellenverzeichnis

## Literaturquellen

- [AAK71] ABDEL-AZIZ, Y. I. ; KARARA, H. M.: Direct Linear Transformation from comparator coordinates into object space coordinates in close-range photogrammetry. Proceedings of the ASP Symposium on Close-Range Photogrammetry, S. 1-18, 1971
- [AHH10] ATCHESON, Bradley ; HEIDE, Felix ; HEIDRICH, Wolfgang: CALTag - High Precision Fiducial Markers for Camera Calibration. Siegen, Germany : 15<sup>th</sup> International Workshop on Vision, Modeling and Visualization, 2010
- [Bar07] BARSKI, Anatoli: Objekterkennung in 2D-Bildern und Tiefenbildern. Universität Karlsruhe, 2007
- [Übe95] ÜBERHUBER, Christoph: Computer-Numerik 2. Springer, 1995 (Computer Numerik). – ISBN 9–78354059–152–8
- [BETG08] BAY, Herbert ; ESS, Andreas ; TUYTELAARS, Tinne ; GOOL, Luc V.: Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, 2008
- [BFG06] BAY, Herbert ; FASEL, Beat ; GOOL, Luc V.: Interactive Museum Guide: Fast and Robust Recognition of Museum Objects. Computer Vision Laboratory (BIWI), 2006
- [BL97] BEIS, Jeffrey S. ; LOWE, David G.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. Conference on Computer Vision and Pattern Recognition, 1997
- [BL02] BROWN, Matthew ; LOWE, David G.: Invariant features from interest point groups. British Machine Vision Conference, 2002
- [Bra10] BRAUER, Henrick: Entwicklung eines Augmented Reality Frameworks auf Basis von Kamera-basierten Trackingverfahren. Masterarbeit - Hochschule für Angewandte Wissenschaften Hamburg, 2010
- [BTG06] BAY, Herbert ; TUYTELAARS, Tinne ; GOOL, Luc V.: SURF: Speeded Up Robust Features. Computer Vision (ECCV), 2006. – ISBN 978–3–540–33832–1
- [BTZ96] BEARDSLEY, Paul ; TORR, Philip ; ZISSERMAN, Andrew: 3D Model Aquisition from Extended Image Sequences. Springer-Verlag, 1996
- [Bus09] BUSS, Samuel R.: Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods. University of California, San Diego, La Jolla, 2009
- [Can86] CANNY, F. J.: A Computational Approach to Edge Detection. Nr. 6, S. 679-698, 1986
- [CG08] CORNELIS, Nico ; GOOL, Luc V.: Fast Scale Invariant Feature Detection and Matching on Programmable Graphics Hardware. Computer Vision and Pattern Recognition, 2008
- [CH06] CHAUMETTE, Francois ; HUTCHINSON, Seth: Visual servo control, Part I: Basic approaches. IEEE Robotics and Automation Magazine 13, Nr. 4, S. 82-90, 2006
- [CL92] COLEMAN, Thomas F. ; LI, Yuying: On the Convergence of Reflective Newton Methods for Large-scale Nonlinear Minimization Subject to Bounds. USA : Cornell University, 1992
- [CL96] COLEMAN, Thomas F. ; LI, Yuying: An interior trust region approach for nonlinear minimization subject to bounds. 1996

- [Cor96] CORKE, Peter I.: A Robotics Toolbox for MATLAB. IEEE Robotics and Automation Magazine 3, Nr. 1, S. 24-32, 1996. ISSN 1070-9932
- [DDDS02] DAVID, Philip ; DEMENTHON, Daniel F. ; DURAISWAMI, Ramani ; SAMET, Hanan: SoftPOSIT: Simultaneous Pose and Correspondence Determination. Computer Vision - EC-CV, S. 79-95, 2002
- [DH55] DENAVIT, Jacques. ; HARTENBERG, Richard S.: A kinematic notation for lower-pair mechanisms based on matrices. 1955
- [Ebe09] EBERT, Matthias: Modellbasierte Posebestimmung aus 2-D/3-D SIFT-Korrespondenzen. Diplomarbeit - Universität Koblenz, 2009
- [EMNW05] ENGELN-MÜLLIGES, Gisela ; NIEDERDRENK, Klaus ; WODICKA, Reinhard: Lineare und nichtlineare Approximation. Springer Berlin Heidelberg, S. 291-349, 2005. – ISBN 978-3-540-26353-1
- [FB87] FISCHLER, Martin A. ; BOLLES, Robert C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Graphics and Image Processing - SRI International, 1987. – ISBN 0-934613-33-8
- [FHT96] FAHRMEIR, Ludwig ; HAMERLE, Alfred ; TUTZ, Gerhard: Multivariate statistische Verfahren. 1996. – ISBN 3-11-013806-9
- [Fia05] FIALA, Mark: ARtag, a Fiducial Marker System Using Digital Techniques. Washington, DC, USA : IEEE Computer Society, 2005 (Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - S. 590-596). – ISBN 0-7695-2372-2
- [FS10] FRIES, Carsten ; SCHUHFUSS, Martin: Realisierung eines 3D-Laserscanners: auf Basis der Lichtschnitttechnik. VDM Verlag Dr. Müller, 2010. – ISBN 9-783-63928153-8
- [GL04] GORDON, Iryna ; LOWE, David G.: Scene Modelling, Recognition and Tracking with Invariant Image Features. International Symposium on Mixed and Augmented Reality (ISMAR) - University of British Columbia, 2004
- [Gra07] GRAF, Simone: Kamerakalibrierung mit radialer Verzeichnung: Die radiale essentielle Matrix. Dissertation - Universität Passau, 2007
- [HD64] HARTENBERG, Richard S. ; DENAVIT, Jacques: Kinematic Synthesis of Linkages. New York: McGraw-Hill, 1964
- [Hey05] HEYMANN, Sebastian: Implementierung und Evaluierung von Video Feature Tracking auf moderner Grafikkhardware. Diplomarbeit - Bauhaus Universität Weimar, 2005
- [HHN88] HORN, Berthold K. P. ; HILDEN, H. M. ; NEGAHDARIPOUR, Shariar: Closed-Form Solution of Absolute Orientation using Orthonormal Matrices. Journal of the Optical Society America, S. 1127-1135, 1988
- [HM05] HELMS, E. ; MEYER, C.: Assistor - Mensch und Roboter rücken zusammen: Einsatz von sicheren und einfach bedienbaren Assistenzrobotern im Produktionsalltag. Fraunhofer-Institut für Produktionstechnik und Automatisierung : Werkstattstechnik online, 2005. ISSN 1436-4980
- [Hof09] HOFMANN, Robert: Implementierung des SURF-Feature-Detektors auf der GPU mit Hilfe von CUDA. Universität Koblenz, 2009
- [Hor07] HORNING, Oliver: Bildbasierte Bahnplanung und Regelung von Robotern für Handhabungsaufgaben. VDI-Verlag, 2007. ISSN 978-3-18-511608-7
- [Här06] HÄRTEL, Volker: Effiziente Repräsentation von hochdimensionalen Merkmalsvektoren für die Korrespondenzfindung. Diplomarbeit - Christian Albrechts Universität zu Kiel, 2006
- [HS88] HARRIS, Chris ; STEPHENS, Mike: A Combined Corner and Edge Detector. 1988 (Proceedings of the 4th Alvey Vision Conference - S. 147-151)

- [HZ04] HARTLEY, Richard I. ; ZISSERMAN, Andrew: Multiple View Geometry in Computer Vision. Second. Cambridge University Press, 2004. – ISBN 0–521–54051–8
- [Ich05] ICHBIAH, Daniel: Roboter: Geschichte - Technik - Entwicklung. Knesebeck, 2005. – ISBN 3–896–60276–4
- [Jäh05] JÄHNE, Bernd: Digitale Bildverarbeitung. Springer, 2005
- [KB99] KATO, Hirokazu ; BILLINGHURST, Mark: Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. Washington, DC, USA : IEEE Computer Society, 1999 (Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality - S. 85 ff.). – ISBN 0–7695–0359–4
- [Kim06] KIM, Daesik: Camera Calibration. Intelligent Systems Research Center Sungkyunkwan University Suwon, Korea, 2006
- [Krz00] KRZANOWSKI, Wojtek J.: Principles of Multivariate Analysis: A User's Perspective. Oxford University Press, 2000. – ISBN 978–0198507086
- [KS08] KLIMENTJEW, Denis ; STROH, Andre: Grundlagen und Methodik der 3D-Rekonstruktion und ihre Anwendung für landmarkenbasierte Selbstlokalisierung humanoider Roboter. Universität Hamburg, 2008
- [LF05] LEPETIT, Vincent ; FUA, Pascal: Monocular model-based 3D tracking of rigid objects. Hanover, MA, USA : Now Publishers Inc., 2005 (Found. Trends. Comput. Graph. Vis. - S. 1-89). ISSN 1572–2740
- [LH81] LONGUET-HIGGINS, Hugh C.: A computer algorithm for reconstructing a scene from two projections. University of Sussex, 1981
- [LHM00] LU, Chien-Ping ; HAGER, Gregory D. ; MJOLSNESS, Eric: Fast and Globally Convergent Pose Estimation from Video Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, S. 610-622, 2000. ISSN 0162–8828
- [Lin94] LINDBERG, Tony: Scale-space theory: A basic tool for analysing structures at different scales. Journal of Applied Statistics, 1994
- [Low99] LOWE, David G.: Object Recognition from Local Scale-Invariant Features. International Conference on Computer Vision, 1999
- [Low01] LOWE, David G.: Local Feature View Clustering for 3D Object Recognition. International Conference on Computer Vision, 2001
- [Low04] LOWE, David G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 2004
- [Mai09] MAIERO, Jens: Image-based Tracking. Masterarbeit - Hochschule Bonn-Rhein-Sieg, 2009
- [Müc07] MÜCKE, Florian E.: Analyse GPU-basierter Feature Tracking Methoden für den Einsatz in der Augmented Reality. Diplomarbeit - Universität Augsburg, 2007
- [Mik02] MIKOLAJCZYK, Krystian: Detection of local features invariant to affine transformations. Institut National Polytechnique de Grenoble, 2002
- [MK75] MARZAN, G. T. ; KARARA, H. M.: A computer program for direct linear transformation solution of collinearity condition, and some applications of it. Proceedings of the Symposium on Close-Range Photogrammetric Systems, 1975
- [MNLf07] MORENO-NOGUER, Francesc ; LEPETIT, Vincent ; FUA, Pascal: Accurate Non-Iterative O(n) Solution to the PnP Problem. Rio de Janeiro, Brazil : IEEE International Conference on Computer Vision, 2007. ISSN 0920–5691

- [Mor78] MORÉ, Jorge J. ; WATSON, G. (Hrsg.): Lecture Notes in Mathematics. Bd. 630: The Levenberg-Marquardt algorithm: Implementation and theory. Springer Berlin / Heidelberg - S. 105-116, 1978
- [MP05] MARIOTTINI, Gian L. ; PRATTICHIZZO, Domenico: EGT: a Toolbox for Multiple View Geometry and Visual Servoing. IEEE Robotics and Automation Magazine 3, Nr. 12, 2005
- [ODD96] OBERKAMPF, Denis ; DEMENTHON, Daniel F. ; DAVIS, Larry S.: Iterative pose estimation using coplanar feature points. Computer Vision Image Understanding - Elsevier Science Inc., S. 495-511, 1996. ISSN 1077-3142
- [PSM81] PAUL, Richard P. ; SHIMANO, Bruce ; MAYER, Gordon E.: Differential Kinematic Control Equations for Simple Manipulators. IEEE Transactions on Man and Cybernetics Systems, Nr. 6, S. 456-460, 1981. ISSN 0018-9472
- [RML10] RUWANPURA, Janaka ; MOHAMED, Yasser ; LEE, Sang H.: Construction Research Congress 2010: Innovation for Reshaping Construction Practice. American Society of Civil Engineers, 2010 (Proceedings of the 2010 Construction Research Congress). – ISBN 9-78078441-109-4
- [Rod04] RODEHORST, Volker: Photogrammetrische 3D-Rekonstruktion im Nahbereich durch Auto-Kalibrierung mit projektiver Geometrie. wvb Wissenschaftlicher Verlag Berlin, 2004. – ISBN 9-783-93684-683-6
- [Ron05] RONCAT, Andreas: Nahbereichsphotogrammetrie. Diplomarbeit - Bauhaus Universität Weimar, 2005
- [Sch00] SCHARR, Hanno: Optimale Operatoren in der Digitalen Bildverarbeitung. Universität Heidelberg, 2000
- [Sch06] SCHATZ, Alexej: Visuelle Navigation mit Scale Invariant Feature Transform. Diplomarbeit - Universität Bielefeld, 2006
- [SHW04] SCHRAFT, Rolf D. ; HÄGELE, Martin ; WEGENER, Kai: Service-Roboter-Visionen. Hanser, 2004. – ISBN 9-783-44622840-5
- [Sta09] STARK, Georg: Robotik mit Matlab. Hanser Fachbuchverlag, 2009. – ISBN 9-7834464-1962-9
- [Teg06] TEGTMEIER, André: Augmented Reality als Anwendungstechnologie in der Automobilindustrie. Dissertation - Otto-von-Guericke-Universität Magdeburg, 2006
- [TFZ99] TORR, Philip H. S. ; FITZGIBBON, Andrew W. ; ZISSERMAN, Andrew: The Problem of Degeneracy in Structure and Motion Recovery from Uncalibrated Image Sequences. International Journal of Computer Vision, 1999. ISSN 0920-5691
- [TM93] TORR, Philip H. S. ; MURRAY, David W.: Outlier Detection and Motion Segmentation. SPIE volume 2059, 1993
- [TM97] TORR, Philip H. S. ; MURRAY, David W.: The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix. International Journal of Computer Vision, 1997
- [Tor95] TORR, Philip H. S.: Outlier Detection and Motion Segmentation. Dissertation - University of Oxford, 1995
- [Tor02] TORR, Philip H. S.: Bayesian Model Estimation and Selection for Epipolar Geometry and Generic Manifold Fitting. International Journal of Computer Vision, 2002
- [TV98] TRUCCO, Emanuele ; VERRI, Alessandro: Introductory Techniques for 3-D Computer Vision. Upper Saddle River, NJ, USA : Prentice Hall PTR, 1998. – ISBN 0-132-61108-2
- [TZ00] TORR, Philip H. S. ; ZISSERMAN, Andrew: MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. Computer Vision and Image Understanding, 2000

- [Wag11] WAGNER, Benjamin: Kamerageführte Positionierung eines Roboterarms für Handhabungsaufgaben. Masterarbeit - Hochschule für Angewandte Wissenschaften Hamburg, 2011
- [WLS08] WAGNER, Daniel ; LANGLOTZ, Tobias ; SCHMALSTIEG, Dieter: Robust and unobtrusive marker tracking on mobile phones. Los Alamitos, CA, USA : IEEE Computer Society, 2008 (IEEE / ACM International Symposium on Mixed and Augmented Reality - S. 121-124). – ISBN 978–1–4244–2840–3
- [Zha99] ZHANG, Zhengyou: Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. International Conference on Computer Vision, S. 666-673, 1999
- [Zha00] ZHANG, Zhengyou: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 2000
- [Zim07] ZIMMERMANN, Fabian: Keypoint-Detektion und Deskriptoren-Berechnung auf der Grafikkarte. Diplomarbeit - Technische Universität Kaiserslautern, 2007

## Internetquellen

- [ALL10] ALLIED: ALLIED - Vision Technologies. 2010 [HTTP://WWW.ALLIEDVISIONTEC.COM/DE/PRODUKTE/](http://www.alliedvisiontec.com/de/produkte/). – Zugriffsdatum: 09.08.2010
- [Ang07] ANGERMANN, Anne: Matlab - Simulink - Stateflow: Grundlagen, Toolboxen, Beispiele. Oldenbourg, 2007 [HTTP://BOOKS.GOOGLE.DE/BOOKS?ID=UNJHWTZR5AC](http://books.google.de/books?id=UNJHWTZR5AC). – ISBN 9–783486582–727. – Zugriffsdatum: 02.03.2011
- [Bou08] BOUGUET, Jean-Yves: Camera Calibration Toolbox for Matlab. 2008 [HTTP://WWW.VISION.CALTECH.EDU/BOUGUETJ/CALIB\\_DOC/](http://www.vision.caltech.edu/bouguetj/calib_doc/). – Zugriffsdatum: 11.07.2010, Stand 02.06.2008
- [Can07] CANTERAKIS, Nikos: Computer Vision I. Vorlesungsskript im Rahmen des Lehrstuhls für Mustererkennung der Universität Freiburg, 2007 [HTTP://LMB.INFORMATIK.UNI-FREIBURG.DE/LECTURES/COMPUTER\\_VISION/CVF\\_8.PDF](http://lmb.informatik.uni-freiburg.de/lectures/computer_vision/cvf_8.pdf). – Zugriffsdatum: 03.07.2010, Stand 26.11.2007
- [Fra09] FRAUNHOFER IPA: Care-O-bot 3 Serviceroboter. Fraunhofer-Institut für Produktionstechnik und Automatisierung, 2009 [HTTP://WWW.CARE-O-BOT.DE/](http://www.care-o-bot.de/). – Zugriffsdatum: 24.03.2011, Stand 29.10.2009
- [GGE09] GLEASON, Scott ; GEBRE-EGZIABHER, Demoz: GNSS Applications and Methods. Artech House, 2009 (GNSS Technology and Applications). [HTTP://BOOKS.GOOGLE.DE/BOOKS?ID=JUXAE3SHQROc](http://books.google.de/books?id=JUXAE3SHQROc). – ISBN 9–78159693–329–3. – Zugriffsdatum: 02.03.2011
- [KA00] KNABNER, Peter ; ANGERMANN, Lutz: Numerik partieller Differentialgleichungen: eine anwendungsorientierte Einführung. Springer, 2000 (Springer-Lehrbuch Masterclass). [HTTP://BOOKS.GOOGLE.DE/BOOKS?ID=1xPc6-Ed4Pgc](http://books.google.de/books?id=1xPc6-Ed4Pgc). – ISBN 9–78354066–231–0. – Zugriffsdatum: 17.04.2011
- [KS99] KIPNIS, Aviad ; SHAMIR, Adi: Advances in Cryptology - Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. Bd. 1666. Springer Berlin / Heidelberg, S. 788-788, 1999 [HTTP://DX.DOI.ORG/10.1007/3-540-48405-1\\_2/](http://dx.doi.org/10.1007/3-540-48405-1_2/). – ISBN 3–540–48405–1. – Zugriffsdatum: 06.03.2011
- [Mei08a] MEISEL, Andreas: 3D Bildverarbeitung. Vortragsfolien zu Grundlagen der 3D-Bildverarbeitung, 2008 [HTTP://WWW.INFORMATIK.HAW-HAMBURG.DE/1708.HTML](http://www.informatik.haw-hamburg.de/1708.html). – Zugriffsdatum: 12.02.2010, Stand 03.03.2008

- [Mei08b] MEISEL, Andreas: Robot Vision. Vorlesungsskript des Faches Robot Vision, 2008 [HTTP://WWW.INFORMATIK.HAW-HAMBURG.DE/WP\\_ROBOT\\_VISION.HTML](http://www.informatik.haw-hamburg.de/wp_robot_vision.html). – Zugriffsdatum: 12.02.2010, Stand 07.03.2008
- [MNLF10] MORENO-NOGUER, Francesc ; LEPETIT, Vincent ; FUA, Pascal: Accurate Non-Iterative O(n) Solution to the PnP Problem. Präsentationsfolien, 2010 [HTTP://WWW.IRI.UPC.EDU/PEOPLE/FMORENO/](http://www.iri.upc.edu/people/fmoreno/). – Zugriffsdatum: 11.10.2010
- [Neu10] NEURONICS AG: Neuronics AG - intelligent & personal robotics. 2010 [HTTP://WWW.NEURONICS.CH/](http://www.neuronics.ch/). – Zugriffsdatum: 04.10.2010
- [OWI10] OWIS: OWIS Staufen. 2010 [HTTP://WWW.OWIS-STAUFEN.DE](http://www.owis-staufen.de). – Zugriffsdatum: 04.10.2010
- [PEN10] PENTAX: PENTAX. 2010 [HTTP://SECURITY-SYSTEMS.PENTAX.DE/DE/PRODUCT/](http://security-systems.pentax.de/de/product/). – Zugriffsdatum: 09.08.2010
- [Son10] SONY ERICSSON MOBILE COMMUNICATIONS AB: Sony Ericsson Mobile Phones. 2010 [HTTP://WWW.SONYERICSSON.COM/CWS/PRODUCTS/MOBILEPHONES/](http://www.sonyericsson.com/cws/products/mobilephones/). – Zugriffsdatum: 30.04.2010
- [The09a] THE IMAGING SOURCE: Cameras. 2009 [HTTP://WWW.THEIMAGINGSOURCE.COM/PRODUCTS/CAMERAS/](http://www.theimagingsource.com/products/cameras/). – Zugriffsdatum: 18.07.2010, Stand 16.05.2009
- [The09b] THE IMAGING SOURCE: Lenses. 2009 [HTTP://WWW.THEIMAGINGSOURCE.COM/PRODUCTS/OPTICS/LENSES/](http://www.theimagingsource.com/products/optics/lenses/). – Zugriffsdatum: 18.07.2010, Stand 16.05.2009
- [TMHF00] TRIGGS, Bill ; MCLAUHLAN, P. ; HARTLEY, Richard ; FITZGIBBON, A.: Vision Algorithms: Theory and Practice (Lecture Notes in Computer Science). Bd. 1883: Bundle Adjustment - A Modern Synthesis. Springer-Verlag, S.298-372, 2000 [HTTP://LEAR.INRIALPES.FR/PUBS/2000/TMHF00/](http://lear.inrialpes.fr/pubs/2000/TMHF00/). – Zugriffsdatum: 07.03.2011

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Titel

**"Kamerabasierte Identifizierung und Lokalisierung von Gegenständen  
für flexible Roboter"**

im Sinne der Prüfungsordnung nach §22(4) bzw. §16(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 12. August 2011

Ort, Datum

Carsten Fries

Unterschrift