



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Benjamin Wagner

Kamerageführte Positionierung eines
Roboterarms für Handhabungsaufgaben

Benjamin Wagner
Kamerageführte Positionierung eines Roboterarms
für Handhabungsaufgaben

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 11. Februar 2011

Benjamin Wagner

Thema der Masterarbeit

Kamerageführte Positionierung eines Roboterarms für Handhabungsaufgaben

Stichworte

Visual Servoing, SIFT, Rekonstruktion von Raumpunkten, Kinematik

Kurzzusammenfassung

In dieser Arbeit wird ein generisches Verfahren zum kamerabasierten Greifen von ungeordneten Gegenständen für stationäre Roboterarme entwickelt. Der Fokus liegt hierbei auf der Positionierung des Endeffektors in eine Greiflage bezüglich eines Gegenstandes. Dies wird anhand einer hybriden Methode realisiert, welche aus einer direkten Steuerung (*open-loop approach*) und einer anschließenden regelbasierten Vorgehensweise (*closed-loop approach*) besteht. Diese Strategie bietet insgesamt eine genaue Einstellung der Greiflage und verringert gleichzeitig die Prozesszeit im Vergleich zu einer rein regelbasierten Methode. Weitere Aspekte dieser Arbeit sind anhand der Roboter-Kinematik, der Detektion von korrespondierenden Bildpunkten und der Rekonstruktion von Raumpunkten gegeben. Das Greif-Verfahren wird simulativ sowie experimentell auf Basis des Katana-Roboterarms vom Typ 450-6M180 verifiziert.

Title of the paper

Camera-based positioning of a robot arm for manipulation tasks

Keywords

Visual Servoing, SIFT, scene reconstruction, kinematics

Abstract

In this work, a generic method for vision-based grasping of unordered objects is developed for stationary robot arms. The focus lies on the positioning of the end effector in a gripping pose with respect to an object. This is achieved by using a hybrid method, which consists of an open- and a subsequent closed-loop approach. This strategy offers an accurate adjustment of the gripping position and simultaneously reduces the process time as compared to a purely closed-loop approach. Other aspects of this work are given by means of robot kinematics, the detection of corresponding image points and the reconstruction of space points. The gripping method is simulated and experimentally verified on the basis of the Katana robot arm of type 450-6M180.

Danksagung

Mein besonderer Dank gilt meinem betreuenden Prüfer Prof. Dr.-Ing. Andreas Meisel. Zunächst für die Ausgabe dieses interessanten Themas und für die Freiheiten die er mir bezüglich der Umsetzung gelassen hat. Des Weiteren danke ich ihm für die fachlichen Diskussionen und Ideen, welche das Projekt entscheidend voran gebracht haben.

Weiterhin bedanke ich mich bei allen Kommilitonen aus dem *Robot Vision Lab*, die mich bei der Ausarbeitung dieser Masterarbeit unterstützt haben. Insbesondere danke ich hierbei Carsten Fries für die anregenden Unterhaltungen zum Thema und für die Durchsicht meiner Arbeit.

Ebenso bedanke ich mich bei Verwandten und Freunden, die mir während des Studiums stets zur Seite standen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Problemstellung	1
1.3	Aufbau der Arbeit	2
2	Vergleichbare Arbeiten	3
2.1	Kamerabasiertes Greifen von Gegenständen	3
2.1.1	Direkte Steuerung	4
2.1.2	Visual Servoing	5
2.1.2.1	Überblick	6
2.1.2.2	Klassisches 2-D Visual Servoing	8
2.1.2.3	3-D Visual Servoing	9
2.1.2.4	2,5-D Visual Servoing	11
2.1.2.5	Switched-system Visual Servoing	12
2.1.2.6	Trajektorienplanung	12
2.1.2.7	Partitioned Visual Servoing	13
2.1.2.8	Direct Visual Servoing	14
2.1.3	Bestimmung von Greifparametern	14
2.1.3.1	Übersicht	15
2.1.3.2	Greifen von bekannten Objekten	16
2.1.3.3	Greifen von unbekanntem Objekten	17
2.1.3.4	Greifen von ähnlichen Objekten	18
2.2	Korrespondenzanalyse	19
2.2.1	Scale Invariant Feature Transform	19
2.2.2	Weitere Verfahren	22
3	Konzept	23
3.1	Übersicht	23
3.2	Systemkomponenten	25
4	Hardware	29
4.1	Katana-Roboterarm	29
4.1.1	Beschaffenheit	29
4.1.2	Steuerungsmöglichkeiten	30
4.1.3	Anwendungsbereiche	31
4.2	Kamera	32
5	Modellierung der Roboter-Kinematik	33
5.1	Pose eines starren Körpers	33
5.2	Rotationsmatrix	34
5.2.1	Elementare Rotationen	34
5.2.2	RPY-Winkelkonvention	35
5.3	Lineare Koordinatentransformation	36
5.4	Differentielle Transformation	36
5.5	Denavit-Hartenberg Konvention	37
5.6	Direkte Kinematik	38

5.7	Differentielle Kinematik	39
5.8	Inverse Kinematik	39
5.8.1	Numerische Lösung	40
5.8.2	Kinematische Singularitäten	41
5.9	Parameter des Katana-Roboters	42
5.9.1	Kinematische Kette	42
5.9.2	Kinematik	43
6	Modellierung der Kamera	44
6.1	Zentrales Projektionsmodell	44
6.2	Linsenverzeichnung	46
6.3	Entzerrung von Bildpunkten	48
7	Kalibrierung	50
7.1	Kamerakalibrierung	50
7.1.1	Intrinsische Parameter	50
7.1.2	Extrinsische Parameter	53
7.1.2.1	Ausgleichung	53
7.1.2.2	Robuste Schätzung	53
7.2	Hand-Auge-Kalibrierung	55
8	Direkte Steuerung	59
8.1	Beschreibung	59
8.2	Steuereinheit	59
9	Bildpunkt-basierte Regelung	61
9.1	Bildjacobimatrix	61
9.2	Stabilität der Regelung	62
10	Korrespondenzanalyse	65
10.1	Nearest-Neighbor-Algorithmus	66
10.2	Homography-Matrix	66
10.2.1	Beschreibung	66
10.2.2	Ausgleichung	67
10.2.3	Robuste Schätzung	69
10.3	Auswertung	70
11	Rekonstruktion von Raumpunkten	76
11.1	Epipolareometrie	77
11.1.1	Fundamentalmatrix	78
11.1.1.1	Übersicht	78
11.1.1.2	Ausgleichung	78
11.1.1.3	Degeneration	80
11.1.2	Essentialmatrix	84
11.1.2.1	Beschreibung	84
11.1.2.2	Zerlegung	85
11.2	Triangulation	87
11.3	Bündelblockausgleich	87
11.4	Skalierung der Rekonstruktion	88

11.5 Auswertung	88
12 Synthese	91
12.1 Vorbereitung eines Greifvorgangs	91
12.2 Durchführung eines Greifvorgangs	93
12.2.1 Suche nach dem Gegenstand	93
12.2.2 Direkte Steuerung	94
12.2.3 Bildpunktbasierte Regelung	94
12.2.4 Abschließende Greifbewegungen	97
12.3 Realisierung	97
13 Simulation	99
13.1 Camera retreat	99
13.2 Positionierung des Endeffektors	103
14 Experimentelle Ergebnisse	108
14.1 Gesamt-System	108
14.2 Einschränkungen bezüglich des Katana-Roboters	114
15 Zusammenfassung und Ausblick	116
A Mathematische Herleitungen	119
A.1 Homogene Koordinaten	119
A.2 Erste Ableitung der Rotationsmatrix und Winkelgeschwindigkeiten	119
A.3 Moore-Penrose-Inverse	119
B Algorithmen	121
B.1 RANSAC	121
B.2 RANSAC mit adaptiver Parameterbestimmung	122
B.3 Normalisierung von Bildpunkten	123
C Inhalt der CD-ROM	125
Symbolverzeichnis	126
Glossar	128
Abkürzungen	129
Abbildungsverzeichnis	130
Tabellenverzeichnis	132
Algorithmenverzeichnis	133
Quellenverzeichnis	134

1 Einleitung

1.1 Motivation

Die *International Federation of Robotics* (IFR) prognostiziert wachsende Verkaufszahlen für Industrie- und Serviceroboter in den kommenden Jahren (vgl. [INTB]). Im Bereich der Service-robotik werden laut [INTA] zum Beispiel Rettungs-, Inspektions- und medizinische Roboter verwendet. Diese werden im Gegensatz zu Industrierobotern oft in vorab nur wenig bekannten Umgebungen eingesetzt und erfordern daher einen hohen Grad an Autonomie. Eine von vielen Problematiken die sich daraus ergeben, ist die kameragestützte Manipulation von Gegenständen.

1.2 Problemstellung

Der stationäre Katana-Roboterarm vom Typ 450-6M180 wurde im Rahmen dieser Arbeit mit einer Kamera in der Nähe seines Endeffektors ausgestattet (vgl. Abbildung 1.1). Das Ziel ist es, dass der Roboterarm einen Gegenstand in seinem Arbeitsbereich autonom greifen können soll. Nach einem Greifvorgang kann ein Objekt manipuliert werden, indem es zum Beispiel an einer vorgegebenen Stelle im Arbeitsraum abgelegt wird. Gegenstände sollen ungeordnet im Arbeitsraum des Manipulators auftreten können und somit ist eine statische Programmierung ungeeignet.

Aus diesem Grund soll eine kamerabasierte Vorgehensweise umgesetzt werden. Die Ergebnisse dieser Arbeit sollen möglicherweise zukünftig für einen mobilen Greifarm im Kontext von Assistenzaufgaben im Haushalt verwertet werden. Nachstehend werden zusätzliche Anforderungen aufgelistet, die an das zu realisierende System und die Umgebung, in der das System eingesetzt werden soll, gestellt werden.

- Gegenstände, die vom Roboter gegriffen werden sollen, befinden sich sichtbar im Arbeitsraum des Greifarms. Partielle Verdeckungen werden durch das System nicht berücksichtigt.
- Es befinden sich keine Hindernisse zwischen dem Roboter und einem zu greifenden Objekt.
- Für einen Greifvorgang kommen nur starre Objekte in Betracht.



Abbildung 1.1: Katana-Greifarm vom Typ 450-6M180

1.3 Aufbau der Arbeit

In dieser Masterarbeit wird eine generische Methode zur Lösung der beschriebenen Problematik entwickelt. Zunächst werden vergleichbare Arbeiten hinsichtlich der Aufgabenstellung vorgestellt und auf dieser Basis wird ein Konzept für das Greifen von ungeordneten Gegenständen konstruiert. Der Kern der Arbeit umfasst die Kombination von zwei Verfahren zur Positionierung eines Greifarms bezüglich eines Gegenstandes. Dabei handelt es sich um eine direkte Steuerung und eine anschließende bildbasierte Regelung. Weitere Aspekte dieser Arbeit sind durch die Bildverarbeitung und die Kinematik des Roboters gegeben. Die entwickelte Methode zum kamerabasierten Greifen wird simulativ und experimentell auf Basis des Katana-Roboters untersucht.

2 Vergleichbare Arbeiten

Dieses Kapitel umfasst die Beschreibung von Ergebnissen auf dem Gebiet des kamerabasierten Greifens von Gegenständen. Weiterhin werden Verfahren zur Korrespondenzanalyse aus der Bildverarbeitung untersucht. Hierbei handelt es sich um die Ermittlung von übereinstimmenden Punkten in perspektivisch unterschiedlichen Bildern. Dieser Aspekt stellt eine wichtige Grundlage für kamerabasierte Greifvorgänge dar.

2.1 Kamerabasiertes Greifen von Gegenständen

Für die weiteren Erläuterungen, wird ein kamerageführter Greifvorgang zunächst in die folgende Sequenz unterteilt (vgl. [KC03], S. 1-2).

1. **Positionierung:** Eine Roboterhand wird auf Basis von Kameradaten in eine Greiflage bezüglich eines Gegenstands positioniert. Eine Lage besteht dabei aus einer Position und einer Orientierung.
2. **Greifen:** Ausgehend von einer Greiflage wird ein Objekt durch eine Roboterhand gegriffen. Bestenfalls muss die Hand in der Greiflage lediglich geschlossen werden.

Zur Durchführung der Positionierung können zwei Ansätze verfolgt werden (vgl. [KC02], S. 5-6). Entweder wird eine direkte Steuerung (*open-loop approach*) oder ein regelbasiertes Vorgehen (*closed-loop approach*) verwendet. Diese Ansätze setzen voraus, dass sich ein zu greifender Gegenstand bereits im Sichtbereich der Kamera befindet. Diese Bedingung wird durch Grafik 2.1 a veranschaulicht, wobei sich das rote Objekt im Sichtfeld der türkisen Kamera befindet. Sollte dies nicht der Fall sein, dann kann der Roboter seinen Arbeitsraum vorab zum Beispiel nach einem bestimmten Muster absuchen bis der Gegenstand wiedererkannt wurde. Nach der Positionierung kann das Greifen eines Objekts eingeleitet werden (sh. Grafik 2.1 b). Hierzu müssen sogenannte Greifparameter bekannt sein, zu denen auch die zur Positionierung notwendige Greiflage gehört. Im weiteren Verlauf dieses Abschnitts werden beide Ansätze zur Positionierung näher untersucht. Anschließend werden Verfahren zur Bestimmung von Greifparametern vorgestellt.

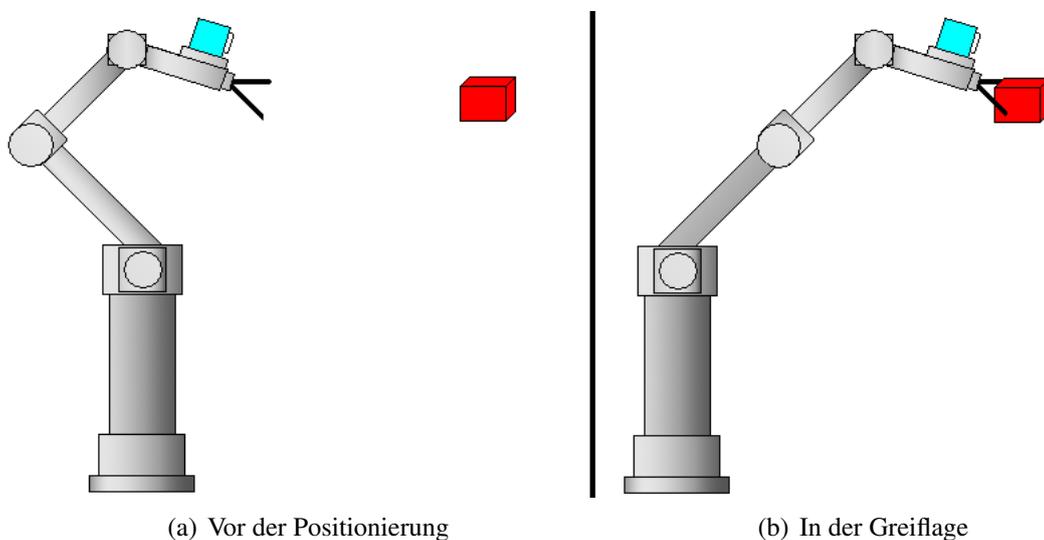


Abbildung 2.1: Einstellung der Greiflage

2.1.1 Direkte Steuerung

Die kamerabasierte Bewegung einer Roboterhand in eine vorgegebene Greiflage relativ zu einem Gegenstand, kann durch eine direkte Steuerung realisiert werden. Typischerweise wird das in Abbildung 2.2 dargestellte nachfolgend erläuterte Verhalten realisiert ([KC02], S. 5-6). Es existiert ein 3D-Modell des Gegenstands, der gegriffen werden soll. Ebenso ist die Greiflage der Roboterhand bezüglich des Objekts bekannt. Zunächst wird ein Bild der aktuellen Szene durch die Kamera aufgenommen und danach mit Hilfe der Bildverarbeitung ausgewertet. Anhand der resultierenden Bildinformationen und des 3D-Modells kann ein zu greifender Gegenstand in der aktuellen Szene wiedererkannt werden. Auf dieser Basis kann die aktuelle Pose (Lage) des Objekts bezüglich der Kamera ermittelt werden.

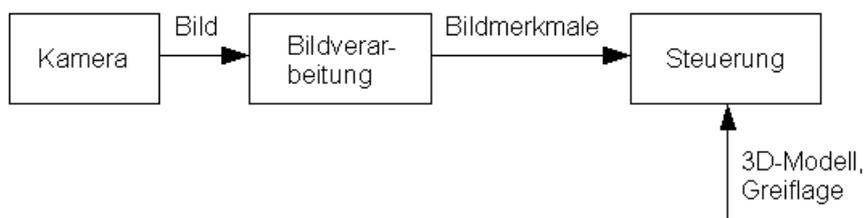


Abbildung 2.2: Direkte Steuerung

Um den bisherigen Ablauf exemplarisch zu verdeutlichen, wird die aktuelle Objektpose ${}^c X_o$ bezüglich der Kamera in Grafik 2.3 gezeigt. Diese Pose wird dazu verwendet, um den Endeffektor des Roboters in die vorgegebene Greiflage relativ zum Gegenstand zu bewegen. Die Greiflage wird in Grafik 2.3 durch die Objektlage ${}^{c^*} X_o$ bezüglich der Kamera repräsentiert. Diese Art der Repräsentation ist möglich, da die Pose zwischen der Roboterhand und der Kamera bekannt ist.

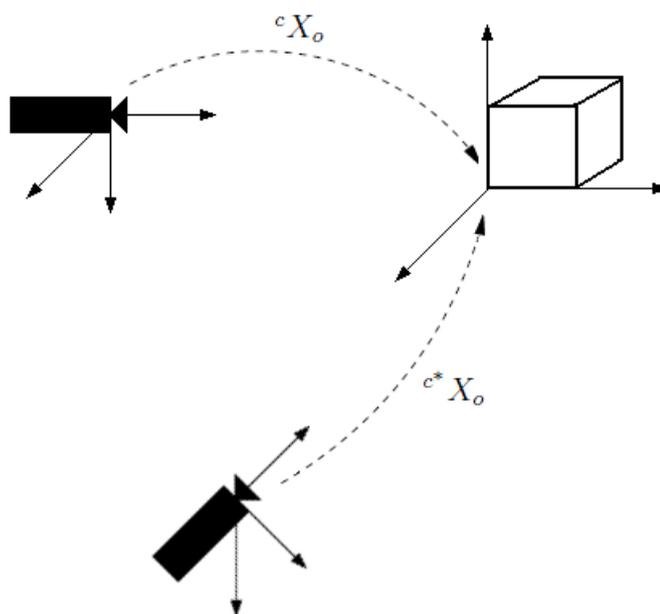


Abbildung 2.3: Objektpose bezüglich der Kamera in der Start- und Zielsituation

Das beschriebene Vorgehen zur Positionierung einer Roboterhand in eine Greiflage kann durch den folgenden Ablauf zusammengefasst werden.

1. **Identifikation:** Der Gegenstand des Interesses wird in der aktuellen Szene auf der Grundlage eines 3D-Modells wiedererkannt.

2. **Lokalisierung:** Daraufhin kann die aktuelle Pose des Objekts bezüglich der Kamera ermittelt werden.
3. **Positionierung:** Diese Pose ermöglicht es, die Roboterhand in die vorgegebene Greiflage relativ zum Gegenstand zu bewegen.

Abschließend erfolgt eine Bewertung des hier vorgestellten Verfahrens. Um das beschriebene Vorgehen umsetzen zu können, wird ein geometrisches Modell des zu greifenden Gegenstands benötigt. Das ist ein Nachteil, da die Erstellung eines solchen Modells prinzipiell aufwendig ist. Dies soll im nachstehenden Beispiel gezeigt werden. In [RLSP06] wird die kamerabasierte Erstellung eines 3D-Modells erläutert. Zu Beginn werden in der Regel 20 Bilder eines zu modellierenden Objekts aus unterschiedlichen Perspektiven rund um das Objekt erstellt. Durch überlappende Bilder können Teilrekonstruktionen des Gegenstands berechnet werden. Diese werden anschließend zu einer Gesamt-Rekonstruktion zusammengefasst. Für die Erstellung eines 3D-Modells werden in [RLSP06] auf den Seiten 17-18 folgende experimentelle Ergebnisse aufgeführt. Das Modell eines Spielzeug-Teddybärs konnte innerhalb von durchschnittlich ungefähr 34 Stunden erstellt werden. Die Berechnungen wurden dabei unter Verwendung von C++ Programmen via Linux auf einem PC mit einer Rechenleistung von 3 GHz durchgeführt. Bei der Verwendung von weniger aufwendigen Algorithmen konnte die Zeit für die Modellierung des Spielzeug-Teddybärs auf circa 3,5 Stunden reduziert werden. Allerdings führt eine Zeitreduzierung auch zu einer Minderung der Qualität des 3D-Modells.

Die Positionierung per direkter Steuerung erfolgt in einem einzigen Schritt. Daraus ergibt sich der Vorteil, dass die Ausführung mit der maximalen Geschwindigkeit des Roboters erfolgen kann. Die einmalige Auswertung der Kameradaten birgt allerdings auch Nachteile. Während sich der Roboter bewegt ist dieser blind und kann somit nicht auf unvorhersehbare Ereignisse reagieren. Aus diesem Grund eignet sich das Verfahren nur für Umgebungen, die sich ab der Bewegung des Roboters nicht mehr verändern. Ein schwerwiegender Nachteil der direkten Steuerung ist, dass die Genauigkeit der Ausführung direkt von der Genauigkeit des Endeffektors des Roboters abhängt ([HHC96], S. 1). Letzteres steht für die Toleranz bei der Einstellung der Gelenkwinkel zur Positionierung der Roboterhand. Daraus resultiert, dass die eingestellte Greiflage von der vorgegebenen Greiflage abweichen kann.

Diese Problematik kann ebenfalls hervorgerufen werden, wenn die berechnete aktuelle Objektpose cX_o bezüglich der Kamera aus Grafik 2.3 fehlerbehaftet ist. Die Ursachen dafür sind Bildrauschen ([PCCS06], S. 1), Kalibrierungsfehler bezüglich der Kamera und Ungenauigkeiten des verwendeten 3D-Modells ([MAA⁺06], S. 2). Alle diese Fehlerquellen treten in der Regel in einem geringen Maße auf. Beim Bildrauschen handelt es sich hier um eine gaußverteilte Abweichung der Pixelpositionen in einem Bild ([HS97A], S. 2). Dies kann durch Fehler bei der Digitalisierung und bei der Bestimmung der Position von Bildmerkmalen, wie zum Beispiel Ecken, entstehen. Bei Letzterem handelt es sich nicht um falsch detektierte Ecken, sondern um eine Abweichung der Eckenposition im Subpixelbereich.

2.1.2 Visual Servoing

Um einen Endeffektor auf Basis von Kameradaten in eine Greiflage bezüglich eines Gegenstandes zu positionieren, kann anstelle einer direkten Steuerung auch eine Regelung eingesetzt werden. Letzteres wird als *Visual Servoing* (VS) bezeichnet (sh. [HHC96], S. 1). In Abbildung

2.4 wird ein entsprechender Regelkreis in allgemeiner Form dargestellt. Die Kameradaten werden kontinuierlich ausgewertet, um die Ist-Größe des Regelkreises zu berechnen. Bei der Soll- und Ist-Größe handelt es sich entweder um Bildmerkmale oder räumliche Informationen, die in den Abschnitten 2.1.2.2 und 2.1.2.3 näher beschrieben werden. Die Soll-Größe repräsentiert dabei die zur Positionierung vorgegebene Greiflage. Die Differenz aus der Soll- und der Ist-Größe stellt den Regelfehler dar, der im Verlauf minimiert werden soll. Der Regelfehler wird auf eine Geschwindigkeit abgebildet, welche eine Bewegung des Endeffektors in Richtung der Greiflage ermöglicht. Wenn die Regelabweichung gegen 0 konvergiert, dann befindet sich die Hand des Roboters in der vorgegebenen Greiflage relativ zum Gegenstand. Die beschriebene Vorgehensweise wird laut ([KC02], S. 10) auch als *visual-motor model* bezeichnet. Hiermit kann eine genauere Positionierung durchgeführt werden als mit einer direkten Steuerung. Allerdings wird die Prozesszeit durch ein regelbasiertes Vorgehen erhöht.

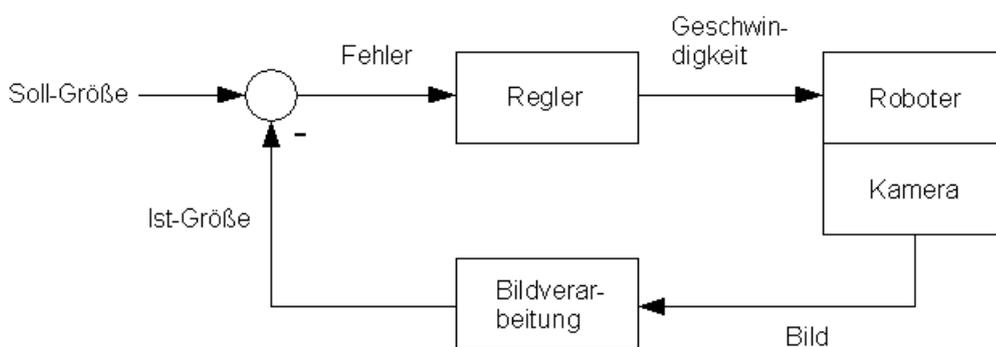


Abbildung 2.4: Regelkreis

Der Begriff *Visual Servoing* wurde 1979 eingeführt und die ersten Systeme auf diesem Gebiet wurden 1980 vorgestellt (sh. [HHC96], S. 1). Seitdem wurden viele unterschiedliche VS-Verfahren entwickelt. In den anschließenden Unterkapiteln 2.1.2.2 bis 2.1.2.8 werden die grundlegenden VS-Verfahren und einige aktuelle Forschungsergebnisse aus dem Bereich des VS beschrieben. Zunächst erfolgt jedoch eine Klassifizierung von VS-Systemen.

2.1.2.1 Überblick

Alle VS-Verfahren die in den folgenden Unterkapiteln 2.1.2.2 bis 2.1.2.8 beschrieben werden, können entweder als *dynamic look-and-move system* oder als *direct visual servo system* realisiert werden ([HHC96], S. 5-6). Im Folgenden wird die Bedeutung dieser Systeme erläutert.

- **Dynamic look-and-move system:** In diesem Fall werden die Gelenke eines Greifarms über Gelenkgeschwindigkeiten angesteuert. Hieraus ergibt sich eine kaskadierte Regelung, bei der ein Sub-System für die Regelung der Gelenkgeschwindigkeiten sorgt.
- **Direct visual servo system:** Hierbei entfällt die Geschwindigkeitsregelung und die Gelenke des Roboters werden direkt anhand von Gelenkwinkeln eingestellt.

Die meisten VS-Systeme werden hinsichtlich der ersten Variante umgesetzt, da dieses einige Vorteile aufweist. Ein Vorteil ist, dass sich der Roboter ständig bewegt und gleichzeitig die Kameradaten auswertet. Dieses Verhalten ist mit einem *direct visual servo system* nur schwer realisierbar. Hier kann eher nur ein Vorgehen im Sinne von *look-then-move* erreicht werden. Ein weiterer Vorteil der ersten Strategie ist, dass ein System mit idealisierter Achsendynamik

konstruiert werden kann. Dadurch kann eine Instabilität und somit ein Aufschwingen der Roboterstruktur vermieden werden. Im Fall eines *direct visual servo system* ist der visuelle Regler allein verantwortlich, um den Mechanismus stabil zu halten. Laut ([HHC96], S. 5) wäre dies ein besonders anspruchsvolles Regelungsproblem.

Eine weitere Kategorisierung von VS-Verfahren wird anhand der Kamera-Konfiguration vorgenommen. VS kann mit einer oder mit mehreren Kameras durchgeführt werden. Weiterhin wird zwischen einer *eye-in-hand* (EIH) und einer *eye-to-hand* (ETH) Konfiguration unterschieden, welche den Standort einer Kamera wie folgt angeben (sh. [HOR07]).

- **EIH:** Bei dieser Ausrichtung werden eine oder mehr Kameras in der Nähe der Roboterhand montiert (vgl. Abbildung 2.5 a).
- **ETH:** Wenn eine oder mehrere Kameras an einer statischen Position angebracht werden, dann spricht man von einer ETH-Lösung. Hierbei werden der Endeffektor und seine Zielposition, wie in Grafik 2.5 b gezeigt, kontinuierlich von außerhalb beobachtet.

Die EIH- und die ETH-Variante können ebenfalls miteinander kombiniert werden. Da in der vorliegenden Arbeit ein Greifroboter mit einer Kamera in der Nähe seiner Hand verwendet wird, werden im Folgenden ausschließlich EIH Lösungen untersucht.

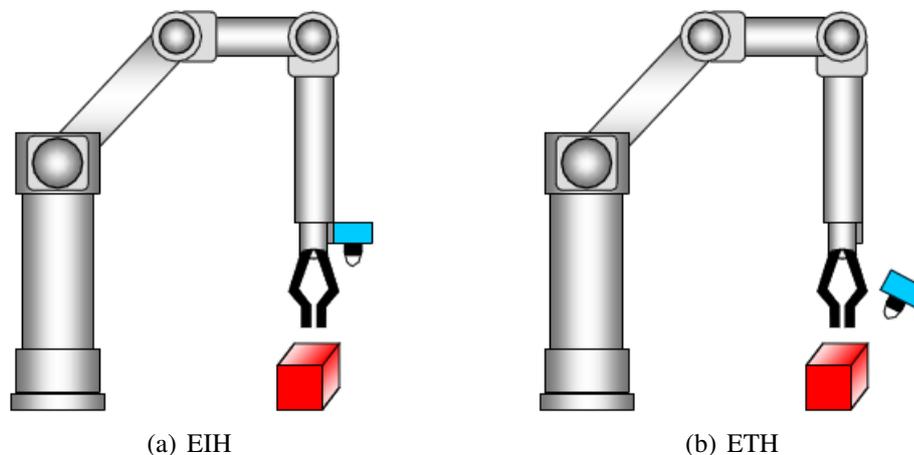


Abbildung 2.5: Hand-Auge-Konfiguration (Bildquelle: [SRV06])

Das Greifen von ungeordneten Gegenständen wurde in verschiedenen Arbeiten erfolgreich auf Basis des VS umgesetzt. In [KBCE05] wurde ein mobiler Greifroboter für den Haushaltsbereich realisiert. Durch [HOR07] wurde ein Greifsystem für einen mobilen Manipulator entwickelt, der im Logistik- und Pflegebereich eingesetzt werden soll. VS eignet sich neben dem Greifen allgemein für Anwendungsbereiche, in denen eine Relativ-Positionierung ausgeführt werden muss. Nachfolgend werden einige Beispiele genannt. In [CCO08] wird ein System für einen mobilen Roboter vorgestellt, der einer Linie auf dem Boden folgt. Die automatische Landung eines Flugzeugs in einer Simulation wird in [BC07] behandelt. Eine weitere Anwendung für den Flugbereich wird in [BGH⁺08] erläutert, bei der sich ein fliegender Roboter einer Zielpose relativ zu einem Objekt nähert. Ein Andockmanöver für ein autonomes Unterwasser-Fahrzeug wird in [LJK03] beschrieben. In der Medizin wird daran gearbeitet VS für die minimal-invasive Chirurgie einzusetzen, um bei Eingriffen ein Operationsinstrument

präzise bis zu einer vorgegebenen Lage führen zu können (vgl. [VMO05]). VS wird außerdem in der Automobilindustrie verwendet, um Bauteile in der Bewegung durch Roboter bearbeiten zu lassen (sh. [INO]).

In den folgenden Unterkapiteln 2.1.2.2 bis 2.1.2.3 werden zunächst die klassischen VS-Verfahren untersucht. Dabei handelt es sich um das bildpunktbasierte VS und das 3-D VS. Diese bilden die Grundlage für einige der fortgeschrittenen Methoden und Bereiche des VS ab Gliederungspunkt 2.1.2.4. In der vorliegenden Arbeit werden lediglich ein paar der wichtigsten Verfahren und Bereiche auf diesem Gebiet in der folgenden Reihenfolge vorgestellt.

- *2,5-D Visual Servoing*,
- *Switched-system Visual Servoing*,
- Trajektorienplanung,
- *Partitioned Visual Servoing* und
- *Direct Visual Servoing*.

Ein umfassenderer Überblick zu fortgeschrittenen VS-Strategien ist in [CH07] gegeben.

2.1.2.2 Klassisches 2-D Visual Servoing

Beim bildbasierten VS erfolgt die Regelung ausschließlich auf Basis von Bildmerkmalen, wie zum Beispiel markanten Bildpunkten, Linien, Ellipsen oder Bildmomenten (sh. [MC08], S. 1). Deshalb wird dieses Vorgehen auch als 2-D VS bezeichnet ([MCB99], S. 1). Im Folgenden wird die klassische Variante des 2-D VS, das bildpunktbasierte VS, vorgestellt (vgl. [CH06]).

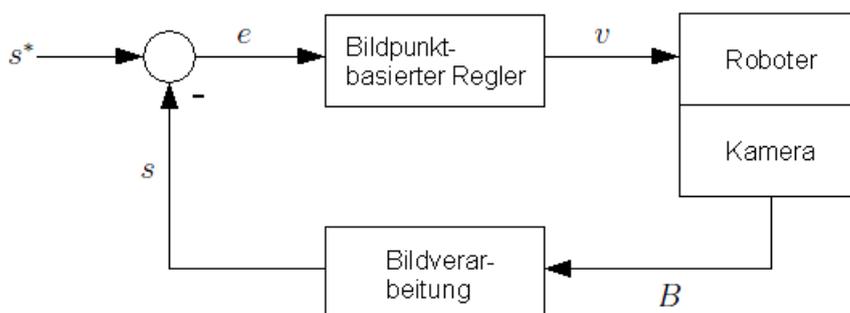


Abbildung 2.6: Bildpunktbasiertes VS

Der Regelfehler e ist durch die Abweichung der Koordinaten von korrespondierenden Bildpunkten aus 2 Ansichten gegeben und wird wie zu Beginn von Kapitel 2.1.2 auf eine Geschwindigkeit v abgebildet (sh. Abbildung 2.6). Bei den 2 Ansichten handelt es sich um eine Referenzansicht B^* und die aktuelle Ansicht B . Die Referenzansicht wird *offline* in einem *Teach-In* Schritt generiert und stellt einen zu greifenden Gegenstand so dar, wie dieser in der Greiflage von der Kamera gesehen werden soll. Aus B^* werden für die Regelung geeignete Bildpunkte extrahiert, deren Koordinaten die Soll-Größe s^* definieren ([KC02], S. 17). In jedem Regelschritt müssen die dazu korrespondierenden Bildpunkte in der aktuellen Ansicht B wiedergefunden werden. Deren Koordinaten ergeben die Ist-Größe s . Das Prinzip eines übereinstimmenden Bildpunkts wird in Grafik 2.7 dargestellt. Ein Raumpunkt P wird auf die

korrespondierenden Bildpunkte p^* und p in den Ansichten B^* und B entsprechend abgebildet. Die Größe c^* steht für die Kamera, wenn sich der Endeffektor in der Greiflage befindet. Die aktuelle Kamera wird durch die Größe c definiert. Wenn der Regelfehler aus Grafik 2.6 gegen 0 konvergiert, dann wird die Regelung erfolgreich beendet. In dieser Situation stimmen B und B^* überein.

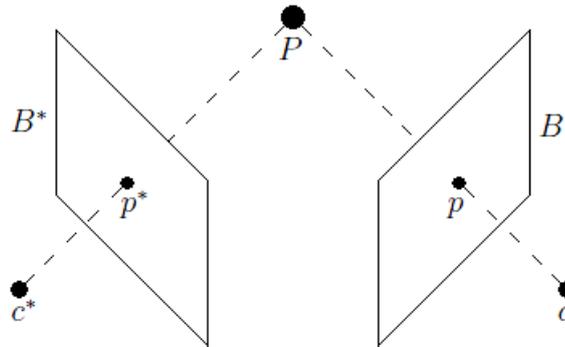


Abbildung 2.7: Korrespondierende Bildpunkte

Die bildpunktbasierete Regelung besitzt mehrere positive Eigenschaften. Zunächst ist das Verfahren laut ([CHA98], S. 1) robust gegenüber Kalibrierungsfehlern und Bildrauschen. Weiterhin werden nur wenig Informationen über den Gegenstand benötigt bezüglich dessen eine Roboterhand positioniert werden soll. Dabei handelt es sich um die in der *Offline-Phase* aufgenommene Zielansicht B^* . Dies birgt allerdings auch einen Nachteil, da die Referenzansicht manuell in einem *Teach-In* Schritt gespeichert werden muss. Eine weitere Einschränkung des Verfahrens ist, dass ein Objekt nur bezüglich der Referenzansicht wiedererkannt und gegriffen werden kann. Um dieses Problem zu umgehen, kann das Verfahren auf mehrere Zielansichten rundum einen Gegenstand erweitert werden.

Ein zusätzlicher Nachteil ist, dass im Regelgesetz der Tiefenabstand der nach s abgebildeten Objektpunkte relativ zur Kamera benötigt wird. Dadurch wird der Einsatz geeigneter Hardware oder spezieller Algorithmen zur Rekonstruktion von Raumpunkten erforderlich. Das schwerwiegendste Problem der bildpunktbasiereten Regelung ist der sogenannte *camera retreat* (vgl. [CH01], S. 4). Hierbei bewegt sich die Kamera erst vom Ziel weg und später wieder darauf zu. Dieser Fall tritt auf, sobald hohe Verdrehungen um die optische Achse der Kamera vorliegen. Die Verdrehung der Kamera bezieht sich dabei auf die Kamera in der Ziellage, welche zur Aufnahme der Zielansicht verwendet wurde. Je höher die Verdrehung um die optische Achse ist, desto stärker ist der beschriebene Effekt. Ein *camera retreat* führt zum Abbruch der Regelung, wenn die geforderte Bewegung durch den Greifarm auf Grund von Gelenkgrenzen nicht mehr ausgeführt werden kann.

2.1.2.3 3-D Visual Servoing

Ein weiteres grundlegendes Verfahren ist das positionsbasierte VS, bei dem auf Basis von räumlichen Informationen geregelt wird. Aus diesem Grund wird das Verfahren auch 3-D VS genannt ([MCB99], S. 1). Die Soll-Größe ist dabei durch die Objektpose ${}^{c^*}X_o$ bezüglich der Kamera aus Grafik 2.3 gegeben ([KC02], S. 12). Die Ist-Größe ist durch die aktuelle Objektpose cX_o bezüglich der Kamera definiert. Die Berechnung von cX_o erfolgt, wie in Abschnitt 2.1.1 beschrieben, auf Basis des aktuellen Bildes B (sh. Grafik 2.8). Dafür wird ein 3D-Modell

des Gegenstands benötigt, der gegriffen werden soll. Der Regelfehler e ergibt sich aus einer geeigneten Subtraktion der Posen ${}^c X_o^*$ und ${}^c X_o$ und wird mit Hilfe eines positionsbasierten Reglers auf eine Geschwindigkeit v abgebildet. Wenn e gegen 0 konvergiert, dann befindet sich der Endeffektor des Roboters in der Greiflage bezüglich des Objekts.

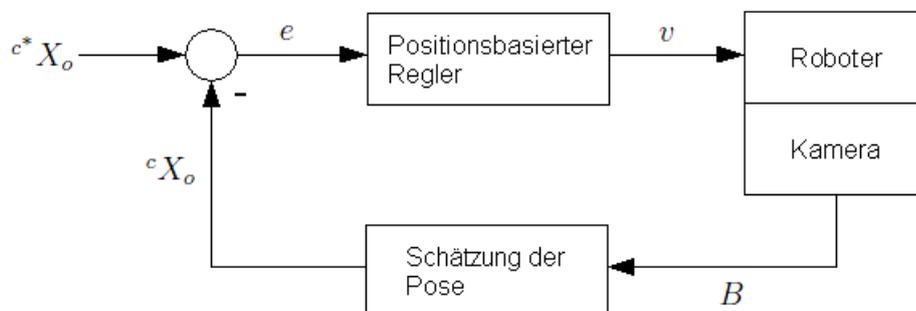


Abbildung 2.8: 3-D VS

Im Folgenden wird eine Evaluierung der hier vorgestellten Methode durchgeführt. Die Notwendigkeit eines 3D-Modells stellt einerseits einen Nachteil dar, da die Erstellung eines solchen Modells prinzipiell aufwendig ist (vgl. 2.1.1). Andererseits kann die Erstellung eines 3D-Modells automatisiert erfolgen. Dies ist ein Vorteil gegenüber dem bildpunktbasierten VS aus Gliederungspunkt 2.1.2.2 bei dem die Referenzdaten über einen *Teach-In* Schritt erfasst werden müssen. Ein weiterer Vorteil ist, dass ein Gegenstand auf Grund der Verwendung eines 3D-Modells aus jeder Ansicht wiedererkannt werden kann. Somit ist es auch möglich den Endeffektor aus jeder realisierbaren Lage in eine vorgegebene Greiflage bezüglich eines Gegenstands zu führen.

Der größte Nachteil des 3-D VS ist seine Anfälligkeit hinsichtlich Bildrauschen, Kalibrierungsfehlern ([CH06], S. 8) und Ungenauigkeiten des verwendeten 3D-Modells ([HHC96], S. 10). Alle diese Einflüsse treten in der Regel in einem geringen Maße auf und führen dazu, dass die aktuelle Objektpose ${}^c X_o$ bezüglich der Kamera fehlerbehaftet ist. Dies wirkt sich, ähnlich wie bei der direkten Steuerung aus Abschnitt 2.1.1, auf die Genauigkeit der Positionierung aus. Im Extremfall kann es sogar passieren, dass ein zu greifendes Objekt aus dem Blickfeld der Kamera verschwindet ([MC00], S. 2). Dies würde zu einem sofortigen Abbruch der Regelung führen.

Ein Vorteil des hier erläuterten Verfahrens ist, dass laut ([CH06], S. 6) eine Entkopplung der translatorischen und rotatorischen Freiheitsgrade (FHG) bezüglich der Ausgangsseite des positionsbasierten Reglers aus Grafik 2.8 vorliegt. Dies wirkt sich direkt auf die Bewegung des Roboters aus und es kann somit eine gerade verlaufende Trajektorie zwischen der Start- und der Greiflage erreicht werden. Die generierte Bewegung ist hier also geeigneter als im Fall der bildpunktbasierter Regelung aus 2.1.2.2, da dort ein *camera retreat* auftreten kann.

Im Folgenden werden aktuellere VS-Strategien erläutert, von denen einige auf den vorgestellten klassischen Verfahren aufbauen. Das Ziel dieser Verfahren ist es, das Verhalten von VS-Systemen zu verbessern.

2.1.2.4 2,5-D Visual Servoing

In [MCB99] wird das 2,5-D VS beschrieben, welches auf der bildpunktbasierter und der positionsbasierter Regelung aus den vorigen Abschnitten 2.1.2.2 bis 2.1.2.3 aufbaut. Aus diesem Grund wird das Verfahren auch als Hybrid VS bezeichnet (vgl. [CH07], S. 3). Das 2,5-D VS verbindet die Vorteile der klassischen Verfahren und vermeidet gleichzeitig deren Nachteile. Dabei wird lediglich die unter Gliederungspunkt 2.1.2.2 erläuterte Referenzansicht B^* zur Positionierung verwendet. Ein 3D-Modell eines Gegenstands, der gegriffen werden soll, wird nicht benötigt.

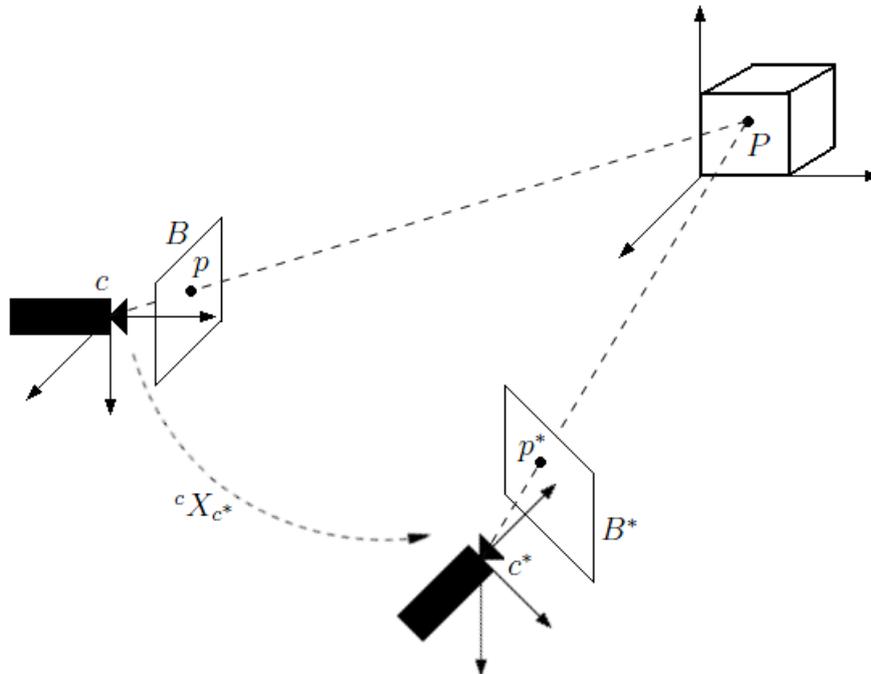


Abbildung 2.9: Pose zwischen der Kamera in der Start- und Zielsituation

Der Regelfehler e ist beim 2,5-D VS durch zwei Anteile e_1 und e_2 definiert. Die Abweichung e_1 ist durch Bildmerkmale gegeben und wird aus der Differenz der Koordinaten der übereinstimmenden Bildpunkte p und p^* berechnet (sh. Grafik 2.9). Im Gegensatz zur bildpunktbasierter Regelung wird hier nur ein Bildpunkt benötigt. Der Fehleranteil e_2 ist durch räumliche Informationen definiert, obwohl beim 2,5-D VS kein 3D-Modell zur Regelung genutzt wird. Stattdessen wird die Lage ${}^cX_{c^*}$ der Kamera c^* aus der Zielsituation bezüglich der aktuellen Kamera c in jedem Regelschritt berechnet, wobei die Kamera c^* mit der Zielansicht B^* assoziiert wird. Dieser Aspekt wird in Abbildung 2.9 veranschaulicht.

Da der Regelfehler e hier zum Teil aus Bildinformationen besteht, wird verhindert, dass ein Objekt wie beim 3-D VS aus dem Blickbereich der Kamera verschwinden kann. Außerdem bietet das 2,5-D VS, genau wie das 3-D VS, eine Entkopplung der translatorischen und rotatorischen Freiheitsgrad (FHG) bei der Bewegung des Roboters. Somit kann der unter Kapitel 2.1.2.2 beschriebene *camera retreat* vermieden werden. Der einzige Nachteil der hier erläuterten Methode, ist laut ([MCB99], S. 12) eine erhöhte Anfälligkeit gegenüber Bildrauschen im Vergleich zur bildpunktbasierter Regelung. Dies kann sich auf die Schätzung der beschriebenen Lage ${}^cX_{c^*}$ auswirken. Experimentelle Ergebnisse auf der Basis von Gegenständen mit künstlichen Marken zeigen die Robustheit des hybriden VS gegenüber Kalibrierungsfehlern.

2.1.2.5 Switched-system Visual Servoing

Bei Switched-system VS handelt es sich um einen Bereich, bei dem es laut ([GH03], S. 1) zu einer Umschaltung zwischen verschiedenen VS-Strategien kommt. Das Ziel ist es ein robustes Gesamtsystem zur Positionierung zu entwerfen, welches auf unzuverlässigen Einzelsystemen basiert. Das Gesamtsystem besteht dabei aus einem Entscheidungsmodul und zwei oder mehreren VS-Verfahren. Das Entscheidungsmodul bestimmt in jedem Regelschritt welches VS-Verfahren für den aktuellen Schritt verwendet werden soll. Dies geschieht in Abhängigkeit vom Systemzustand, von der Zeit oder beidem (sh. [GH03], S. 8).

Ein Verfahren das auf diese Weise funktioniert wird in [GH07] vorgestellt. In diesem Fall wird zwischen der bildpunktbasierten Regelung und dem 3-D VS aus den vorigen Gliederungspunkten 2.1.2.2 bis 2.1.2.3 umgeschaltet. Diese Konstellation beruht auf den folgenden Beobachtungen. Die bildpunktbasierte Regelung ist gut geeignet, wenn dies für das 3-D VS nicht gilt. Ebenso verhält sich das 3-D VS zuverlässig, wenn dies für die bildpunktbasierte Regelung nicht der Fall ist. Das Verfahren erreicht das im Folgenden beschriebene Umschaltverhalten, wobei mit der bildpunktbasierten Regelung begonnen wird. Wenn ein *camera retreat* auftritt, dann wird auf das 3-D VS umgeschaltet. Wenn der Gegenstand, wie unter Abschnitt 2.1.2.3 erläutert, aus dem Blickfeld der Kamera zu verschwinden droht, dann wird wieder auf die bildpunktbasierte Regelung zurückgeschaltet.

Auf diese Weise werden in [GH07] die Vorteile der klassischen Verfahren vereint und deren Nachteile vermieden. Allerdings bleibt die Notwendigkeit eines 3D-Modells von einem Gegenstand für das 3-D VS erhalten. Die hier erläuterte Methode wurde in Simulationen und anhand von Gegenständen mit künstlichen Marken in Experimenten größtenteils verifiziert. Es hat sich dabei allerdings herausgestellt, dass das System im schlechtesten Fall unendlich oft umschalten kann.

2.1.2.6 Trajektorienplanung

In [HOR07] wird ein Verfahren beschrieben, bei dem die bildpunktbasierte Regelung um eine Trajektorienplanung erweitert wird. Letztere wird zur Laufzeit durchgeführt und stellt die Grundlage für eine anschließende Positionierung eines Endeffektors in eine Greiflage bezüglich eines Gegenstands dar. Der Ablauf der Bahnplanung wird im Folgenden kurz erläutert. Als Ausgangspunkt dient der Regelfehler e , welcher durch die Abweichung der Koordinaten von korrespondierenden Bildpunkten aus der Referenzansicht B^* und der aktuellen Ansicht B gegeben ist (vgl. Kapitel 2.1.2.2). Auf dieser Basis ist es möglich die Bahn des Endeffektors schrittweise bis zur Greiflage, welche durch B^* definiert ist, zu berechnen. Dies gilt ebenso für die zu der Bahn des Endeffektors korrespondierende Trajektorie der verfolgten Punkte im Bild. Somit ist der Verlauf der Ist-Größe s aus Abschnitt 2.1.2.2 gegeben.

Im nächsten Schritt wird die berechnete Trajektorie des Endeffektors adaptiert. Dies geschieht anhand der Einpassung eines gaußförmigen Geschwindigkeitsprofils bezüglich der gegebenen Bahn des Endeffektors. Daraufhin kann der Verlauf der entsprechenden Bildpunktbahn linear interpoliert werden. Die interpolierten Stützstellen der Bildpunktbahn werden dazu verwendet, um die Positionierung des Endeffektors durch eine Folgeregelung zu realisieren. Dementsprechend wird die Soll-Größe s^* aus Abschnitt 2.1.2.2 im Verlauf der Positionierung angepasst. Jede interpolierte Stützstelle der Bildpunktbahn kann demnach mit einem Zielbild assoziiert

werden. So entsteht die in Abbildung 2.10 gezeigte Kette von abzuarbeitenden Zielbildern. Das letzte Zielbild entspricht dabei der Referenzansicht B^* .

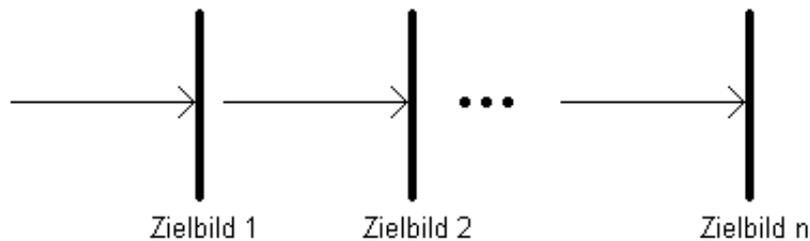


Abbildung 2.10: Folgeregelung

Das beschriebene Vorgehen ermöglicht eine Erweiterung des klassischen 2-D VS, wodurch sich folgende Vorteile ergeben. Durch die erläuterte Einpassung eines Geschwindigkeitsprofils, kann der Verlauf der Geschwindigkeit und die Prozesszeit modelliert werden. Da es sich hier um ein gaußförmiges Geschwindigkeitsprofil handelt, können ruckartige Bewegungen des Roboters vermieden werden. Dies ist besonders bei elastischen Systemen notwendig, damit es nicht zu Schwingungen kommt. Simulative und experimentelle Ergebnisse haben gezeigt, dass die tatsächliche Prozesszeit von der gewünschten Prozesszeit abweichen kann. Dieser Effekt wird mit sinkender gewünschter Prozesszeit verstärkt. Weiterhin erbt die hier beschriebene Methode die Vor- und Nachteile des klassischen 2-D VS aus Kapitel 2.1.2.2.

2.1.2.7 Partitioned Visual Servoing

Der Bereich *Partitioned VS* umfasst bildbasierte Verfahren, welche eine entkoppelte Regelung ermöglichen. Laut ([CH07], S. 4) besteht hierbei das Ziel Bildmerkmale zur Definition des Regelfehlers zu finden, welche möglichst eine vollständige Entkopplung der Freiheitsgrade bezüglich der Stellgröße bieten. Auf diese Weise kann der in Abschnitt 2.1.2.2 beschriebene *camera retreat* vermieden werden (vgl. [CH01], S. 4).

Im Folgenden wird eine Arbeit beschrieben, in der das erläuterte Vorgehen angewendet wird. Das Verfahren *Visual Servoing with Moments of SIFT Features* verwendet SIFT-Merkmale zur Regelung (sh. [HNSR06]), um eine Entkopplung der Freiheitsgrade bezüglich der Stellgröße zu erreichen. SIFT-Merkmale werden unter Abschnitt 2.2.1 im Rahmen der Korrespondenzanalyse noch näher erläutert. An dieser Stelle soll lediglich erläutert werden, dass es sich bei SIFT-Merkmalen um markante Bildpunkte handelt. Jeder markante Bildpunkt wird dabei anhand eines Merkmalsvektors beschrieben, um eine Unterscheidung zwischen verschiedenen markanten Bildpunkten zu ermöglichen. Auf der Basis von übereinstimmenden SIFT-Punkten aus der aktuellen Ansicht B und der Referenzansicht B^* (vgl. Grafik 2.7) können sogenannte Momente, wie zum Beispiel der Schwerpunkt, berechnet werden, um den Regelfehler e zu bestimmen. Durch die Eigenschaften dieser SIFT-Momente können die Freiheitsgrade bezüglich der Stellgröße größtenteils entkoppelt werden.

Simulationsergebnisse zeigen, dass das vorgestellte Verfahren robust funktioniert. Durch die Verwendung von SIFT-Merkmalen ergibt sich allerdings auch ein Nachteil. Wenn die perspektivischen Änderungen zwischen der Zielansicht und der aktuellen Ansicht zu hoch sind, dann können die SIFT-Punkte in der aktuellen Ansicht nicht wiedergefunden werden. Der Grund dafür ist, dass sich die erläuterten Merkmalsvektoren in diesem Fall ändern. Somit wird der

Bereich, in dem die Positionierung bezüglich der Referenzansicht ausgeführt werden kann, verkleinert.

2.1.2.8 Direct Visual Servoing

Verfahren in denen nur das Kamerabild zur Regelung verwendet wird, können als *Direct Visual Servoing* bezeichnet werden (vgl. [SM07]). Als Soll-Größe dient die Helligkeit jedes Pixels aus der Zielansicht. Die Ist-Größe ist durch die Helligkeit jedes Pixels aus dem aktuellen Kamerabild gegeben. Der Vorteil ist, dass die Bildverarbeitung zur Berechnung von visuellen Merkmalen für die Regelung eingespart werden kann.

Diese Vorgehensweise wird in [CMC08] verwendet. Das Verfahren besitzt die Einschränkung, dass die Regelung nur für sogenannte Lambert-Objekte robust durchgeführt werden kann. Bei einem Lambert-Objekt handelt es sich um einen Gegenstand mit einer stumpfen Oberfläche, die dafür sorgt dass auftreffendes Licht in alle Richtungen gleichmäßig reflektiert wird. Diese Eigenschaft besitzt zum Beispiel Papier. Das Gegenteil eines Lambert-Objekts ist ein Gegenstand mit einer spiegelnden Oberfläche. Experimentelle Ergebnisse zeigen, dass die Regelung im Fall eines Nicht-Lambert-Objekts nur für geringe Regelfehler funktioniert.

2.1.3 Bestimmung von Greifparametern

Ein Greifvorgang kann laut ([KC02], S. 1-2) in zwei Schritte unterteilt werden. Zunächst erfolgt die beschriebene Positionierung einer Roboterhand in eine Greiflage bezüglich eines Gegenstands per direkter Steuerung oder VS. Ausgehend von dieser Greiflage kann das Greifen des Gegenstands eingeleitet werden. Im einfachsten Fall muss die Hand in der Greiflage dafür lediglich geschlossen werden. Stattdessen ist es auch möglich, dass vorher eine weitere Bewegung der Roboterhand ausgehend von der Greiflage notwendig ist.

Letzteres kann zum Beispiel im Anschluss an das VS auftreten, da sich ein zu greifender Gegenstand für den gesamten Zeitraum der Regelung im Kamerabild befinden muss. Dies wird durch Abbildung 2.11 a verdeutlicht, in der ein Greifarm nach dem Vorgang der Positionierung dargestellt wird. Nach diesem Vorgang befindet sich das rote Objekt noch im Blickfeld der türkisen Kamera. Auf Grund der Lage der Kamera bezüglich der Hand des Roboters, reicht es nicht aus lediglich die Hand zu schließen um das Objekt zu greifen. Somit ist hier nach der Positionierung eine weitere Bewegung des Endeffektors notwendig.

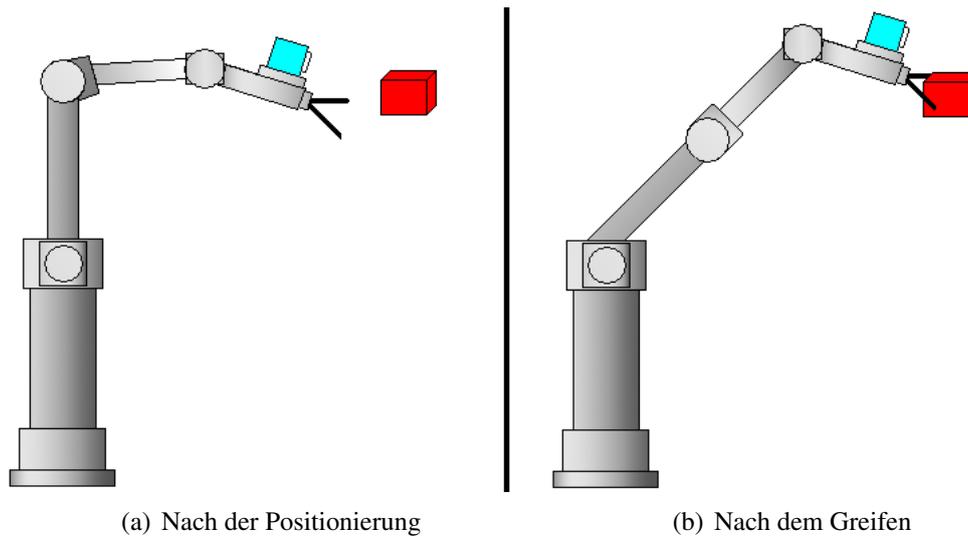


Abbildung 2.11: Greifvorgang

Erst nach dieser Bewegung befindet sich der Endeffektor in einer Lage, in der die Hand nur noch geschlossen werden muss (sh. Grafik 2.11 b). Dementsprechend ist es also möglich, dass ein erhöhter Aufwand für das Greifen nach der Positionierung betrieben werden muss. Unabhängig davon, kann das Greifen im Anschluss an die Positionierung über die zwei folgenden Vorgehensweisen gestaltet werden ([KC03], S. 1-2).

- **Direkte Steuerung:** Es werden vordefinierte Bewegungen ausgeführt, um ein Objekt zu greifen.
- **Regelung:** In diesem Fall erfolgt das Greifen über eine Regelung auf Basis von taktilen- und Kraftsensoren.

Für beide Vorgehensweisen werden sogenannte Greifparameter benötigt, zu denen auch die zur Positionierung notwendige Greiflage gehört. In den nachstehenden Abschnitten werden aktuelle Forschungsergebnisse hinsichtlich der Bestimmung dieser Greifparameter beschrieben.

2.1.3.1 Übersicht

Bei Greifparametern handelt es sich um Größen die im Verlauf des gesamten Greifvorgangs erforderlich sind. Diese Parameter sind nicht einheitlich definiert und können in unterschiedlichen Greifverfahren spezifische Ausprägungen annehmen. Der einzige Standard-Greifparameter ist die erläuterte Greiflage des Endeffektors relativ zu einem Gegenstand. Dieser Parameter wird zur Positionierung des Endeffektors via einer direkten Steuerung oder VS benötigt (vgl. Kapitel 2.1.1 bis 2.1.2). In der Literatur wird bei der Generierung von Greifparametern zwischen grundlegenden Methoden wie folgt unterschieden.

- **Teach-In:** Der Endeffektor wird *offline* per manueller Bedienung in eine geeignete Greiflage gefahren (sh. [PKB⁺10], S. 2). Von hier aus wird ein Gegenstand per Handsteuerung gegriffen. Die Greifparameter sind dann bekannt und müssen lediglich zur späteren Verwendung gespeichert werden.
- **Programming by Demonstration (PbD):** Hier wird das Greifen von Gegenständen *offline* durch einen Benutzer demonstriert (vgl. [EK05], S. 1). Das gezeigte Verhalten wird von einem Rechner interpretiert, um den Greifvorgang zur Laufzeit imitieren zu können.

- **Automatisierte Berechnung:** Die Berechnung von Greifparametern wird in einem Planungsschritt *offline* oder zur Laufzeit vollkommen autonom durchgeführt.

Das PbD-Vorgehen versucht das menschliche Greifen nachzuahmen und ist somit besonders für humanoide Roboter geeignet. Aus diesem Grund wird dieser Ansatz in der vorliegenden Arbeit nicht weiter vertieft. Stattdessen werden im Folgenden aktuelle Ergebnisse auf dem Gebiet der automatisierten Berechnung von Greifparametern auf Basis von Kameradaten vorgestellt. Arbeiten auf diesem Gebiet können laut ([BK10], S. 1) wie folgt kategorisiert werden. Als Kriterium dient dabei die Menge der Informationen, die über ein zu greifendes Objekt zur Verfügung stehen müssen.

- **Greifen von bekannten Objekten:** Gewöhnlicherweise wird ein 3D-Modell eines Gegenstands benötigt, um in einem *Offline*-Planungsschritt alle benötigten Greifparameter zu berechnen. Ein Objekt, das gegriffen werden soll, ist somit vorab bekannt.
- **Greifen von unbekanntem Objekten:** In diesem Fall liegen *offline* keine Informationen vor, die zur Erlangung von Greifparametern führen können. Eine Greifplanung kann somit erst zur Laufzeit ausgeführt werden.
- **Greifen von ähnlichen Objekten:** Es werden vorab bekannte Greiferfahrungen für verwandte Gegenstände verwendet. Verwandte Gegenstände können zum Beispiel eine ähnliche Form oder annähernd gleiche Texturen aufweisen.

In den folgenden Unterkapiteln wird jeweils ein repräsentatives Verfahren für jede der drei Kategorien beschrieben.

2.1.3.2 Greifen von bekannten Objekten

In [MAA⁺06] wird ein Verfahren beschrieben, welches Greifparameter zur Handhabung vorab bekannter Gegenstände berechnet. Um dies zu gewährleisten wird das 3D-Modell eines zu greifenden Gegenstands und der Roboterhand benötigt. Anhand dieser Vorgaben wird *offline* eine Greifanalyse durchgeführt, welche die benötigten Greifdaten berechnet und in einer Datenbank hinterlegt. Die Greifanalyse erfolgt auf Basis des Simulators *GraspIt!*, der in [MA04] erläutert wird. *GraspIt!* wurde in [MAA⁺06] erweitert, um die folgenden Greifparameter zu generieren.

Es wird die Greiflage der Roboterhand relativ zu einem Gegenstand berechnet. Die Greiflage kann wie unter Gliederungspunkt 2.1.1 durch die Objektlage ${}^c X_o$ bezüglich der Kamera repräsentiert werden. Weiterhin wird der Greiftyp festgelegt, welcher über die Form der Hand beim Greifen entscheidet. Zur Laufzeit wird der Greiftyp eingestellt, sobald sich die Roboterhand in der Greiflage befindet. Bei der letzten Größe handelt es sich um die Richtung der Annäherung. Diese gibt ausgehend von der Greiflage vor, in welche Richtung sich die Hand beim Greifen bewegen soll. Zur Laufzeit liegen somit objektspezifische Angaben zum Greifen vor.

Während der Ausführung kann ein Objekt, anhand des vorhandenen 3D-Modells wiedererkannt werden. Auf dieser Basis kann die aktuelle Objektpose ${}^c X_o$ bezüglich der Kamera aus Grafik 2.3 ermittelt werden. Dieses Vorgehen wird auch als Lokalisierung bezeichnet. Für das hier beschriebene Vorgehen werden in [MAA⁺06] keine experimentellen Ergebnisse bereitgestellt, da kein vollständiges Greifsystem umgesetzt wurde. Es wurden lediglich die erläuterten Module zur Lokalisierung und Greifanalyse entwickelt. Allerdings wird in der Arbeit beschrieben,

welche weiteren Module umzusetzen sind.

Es muss ein Modul vorhanden sein, welches nach der Lokalisierung einen geeigneten Greifvorgang aus der Datenbank auswählt. Zu diesem Zweck liegen für jedes Objekt mehrere Greifvorgänge in der Datenbank vor. Somit ist es möglich einen passenden Vorgang in Abhängigkeit von der aktuellen Situation zu bestimmen. Außerdem wird ein Modul zur Positionierung benötigt, um den Endeffektor in die ausgewählte Greiflage zu bewegen. Dies kann durch die direkte Steuerung aus Kapitel 2.1.1 oder via VS erfolgen. Abschließend soll ein regelbasiertes Greifen anhand der übrigen Greifparameter und taktiler Sensoren realisiert werden.

2.1.3.3 Greifen von unbekanntem Objekten

Im Fall von unbekanntem Objekten liegen vorab keine Informationen vor, die zur Erlangung von Greifparametern führen können. Diese Daten müssen somit während der Ausführung ermittelt werden. In [HK08] wird ein Ansatz für diese Problematik beschrieben. Zur Laufzeit wird eine 3D-Teilrekonstruktion eines zu greifenden Gegenstands anhand von zwei überlappenden Kamerabildern erstellt. Dieses Teilmodell besteht lediglich aus verrauschten 3D-Punkten. Eine Vervollständigung des Modells, zum Beispiel auf Basis einer Berechnung von Polygonen, wird nicht vorgenommen. Um mit diesen fehlerbehafteten unvollständigen Daten umgehen zu können, wird die 3D-Punktswolke durch eine Boxstruktur approximiert. Auf diese Weise können potenzielle Greifmöglichkeiten ermittelt werden.

Die Approximation durch eine Box-Konstellation wird exemplarisch in Abbildung 2.12 veranschaulicht, wobei das 3D-Modell einer Ente durch drei Boxen angenähert wurde. In diesem Beispiel wurde das vollständige 3D-Modell eines Objekts approximiert. Zur Laufzeit hingegen liegt nur die erläuterte Teilrekonstruktion in Form einer 3D-Punktswolke vor.

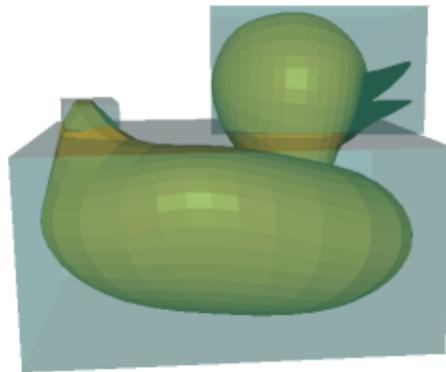


Abbildung 2.12: Box-Approximation (Bildquelle: [HK])

Das hier vorgestellte Verfahren bietet lediglich einen Ansatz zur Bestimmung von Greifdaten, da nur die Parameter für einen sogenannten *Pre-Grip* ermittelt werden können (sh. [HK08], S. 2). Das bedeutet, dass die ermittelten Greifdaten nicht für einen vollständigen Greifvorgang ausreichen. Im Folgenden werden die Greifparameter des hier vorgestellten Verfahrens erläutert, welche auf der Grundlage der beschriebenen Approximation durch eine Box-Konstellation generiert werden können. Die Greiflage der Roboterhand relativ zu einem Gegenstand kann nur partiell berechnet werden, wobei es sich um die Orientierung handelt. Diese sagt aus, wie die Hand des Roboters bezüglich eines Gegenstands ausgerichtet werden soll. Weiterhin kann der Greiftyp bestimmt werden, welcher die Form der Hand beim Greifen festlegt. Ein Vergleich

mit den Greifparametern der Methode aus dem vorigen Abschnitt 2.1.3.2 zeigt, dass die hier erläuterten Parameter nicht ausreichen um einen vollständigen Greifvorgang durchzuführen.

Experimentelle Ergebnisse zeigen, dass die Berechnung von Greifdaten für vorab unbekannte Objekte erfolgreich ausgeführt werden kann. Da diese Parameter allerdings nicht zum Greifen eines Objekts ausreichen, konnten diese somit auch nicht verifiziert werden. Es besteht somit die Möglichkeit das ein geplanter Greifvorgang in einer realen Situation nicht machbar ist (vgl. [HK08], S. 6). Ein Nachteil des vorgestellten Verfahrens ist, dass die erwähnte Approximation der Boxstruktur zur Laufzeit sehr zeitaufwändig ist. In einem repräsentativen Versuch wurde hierfür eine Dauer von 22 Sekunden ermittelt, wobei ein Rechner mit einer doppelten Intel Core2 Quad CPU mit 2,66 GHz verwendet wurde.

2.1.3.4 Greifen von ähnlichen Objekten

Im Folgenden wird ein Verfahren erläutert, welches auf der Annahme beruht, dass ähnlich aussehende Gegenstände auf eine ähnliche Art gegriffen werden können ([BK10], S. 1). Verwandte Gegenstände können zum Beispiel eine ähnliche Form oder annähernd gleiche Texturen aufweisen. In [SDN08] wird beim Greifen von verwandten Objekten auf vorab bekannte Greiferfahrungen zurückgegriffen. Dafür werden auf der Grundlage von maschinellem Lernen Greifpunkte in Bildern ermittelt. Hierbei handelt es sich um mehrere Punkte in einem Bild, welche eine Greifmöglichkeit repräsentieren. Dies wird exemplarisch in Grafik 2.13 veranschaulicht, wobei die nachträglich hinzugefügten roten Punkte die berechneten Greifpunkte darstellen.

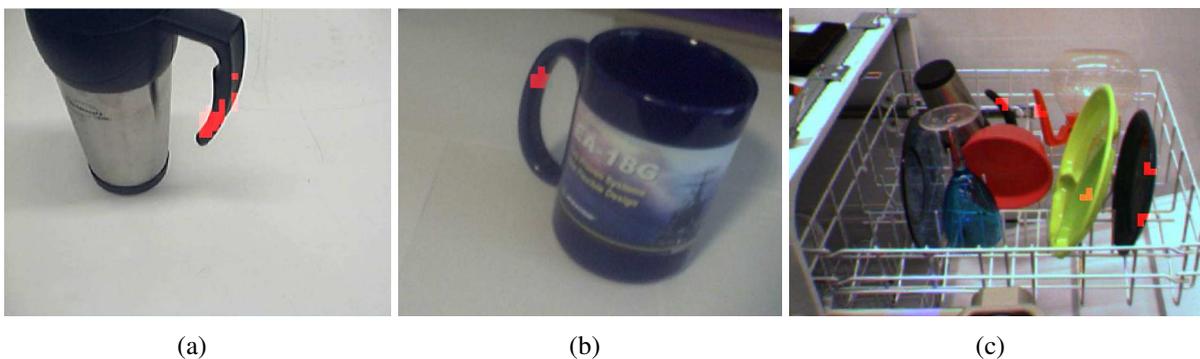


Abbildung 2.13: Ermittelte Greifpunkte (Bildquelle: [SDKN])

Um zur Laufzeit Greifpunkte für verwandte Gegenstände berechnen zu können, wird in [SDN08] *offline* ein überwachter Lernvorgang zur Bereitstellung von Greiferfahrungen durchgeführt. Für den Lernvorgang wurden synthetische Trainingsdaten in Form von gerenderten Bildern bereitgestellt. Für jedes Bild wurden anschließend die Greifpunkte vorgegeben. Während des Lernvorgangs wird der Zusammenhang zwischen Bildmerkmalen und den bekannten Greifpunkten hergestellt. Die Bildmerkmale basieren dabei auf Kanten, Texturen und Farben. Auf der Basis der gelernten Greifpunkte können zur Laufzeit in einem Bild Greifpunkte für Gegenstände ermittelt werden, die den Gegenständen aus den Trainingsdaten ähneln.

Um einen Gegenstand greifen zu können, müssen ermittelte Greifpunkte rekonstruiert werden. Dazu werden zwei oder mehr überlappende Ansichten eines Gegenstands von der Kamera aufgenommen und es werden die Greifpunkte in jedem Bild auf die beschriebene Weise ermittelt. Anschließend werden die korrespondierenden Greifpunkte (vgl. Abbildung 2.7) dazu verwendet, um die entsprechenden räumlichen 3D-Greifpunkte per Triangulation zu rekonstruieren.

Auf Basis der 3D-Greifpunkte ist eine Greiflage der Roboterhand bezüglich des zu handhabenden Objekts ermittelbar. Die 3D-Greifpunkte und die Greiflage stellen die Greifparameter in dem hier vorgestellten Verfahren dar.

Diese Parameter werden für einen Greifvorgang folgendermaßen verwendet. Zunächst wird die Roboterhand in die berechnete Greiflage positioniert und daraufhin wird die Hand in Richtung der 3D-Greifpunkte bewegt. Der restliche Greifvorgang wird in [SDN08] nicht beschrieben. In einem Experiment wurden die Greifpunkte für unterschiedliche Objekte *offline* gelernt. Anhand dieser Greiferfahrungen konnten ähnliche Objekte mit einer Erfolgsquote von 90 % gegriffen werden.

2.2 Korrespondenzanalyse

Dieses Kapitel umfasst eine Untersuchung von Verfahren zur Korrespondenzanalyse. Dabei handelt es sich um die Erfassung von übereinstimmenden Punkten in perspektivisch unterschiedlichen Bildern (vgl. Grafik 2.7). Diese Thematik stellt eine wichtige Grundlage für kamerabasierte Greifvorgänge dar.

2.2.1 Scale Invariant Feature Transform

In [LOW04] wird das Verfahren *Scale Invariant Feature Transform* (SIFT) erläutert, welches eine Berechnung von sogenannten markanten Bildpunkten ermöglicht. Diese können dazu verwendet werden, um eine Korrespondenzanalyse durchzuführen. Im Folgenden wird erläutert, worum es sich bei markanten Bildpunkten handelt und wie diese zur Bestimmung von korrespondierenden Bildpunkten genutzt werden können.

Ein anhand von SIFT ermittelter markanter Bildpunkt, kurz SIFT-Punkt, wird durch seine subpixelgenauen Koordinaten im Bild und einen 128-dimensionalen Merkmalsvektor \mathbf{d} beschrieben. In Abbildung 2.14 werden die berechneten SIFT-Punkte exemplarisch für eine Ansicht mittels ihrer Bildkoordinaten durch rote Markierungen dargestellt. Diese SIFT-Punkte können jeweils hinsichtlich des Vektors \mathbf{d} voneinander unterschieden werden, da die Bildinformationen aus der Nachbarschaft eines SIFT-Punkts bei der Berechnung von \mathbf{d} mitberücksichtigt werden.

In Abbildung 2.15 wird die gleiche Szene aus einer anderen Perspektive gezeigt, wobei die SIFT-Punkte grün dargestellt werden. Im Vergleich zur vorigen Ansicht 2.14 ändern sich die einzelnen Merkmalsvektoren \mathbf{d} bezüglich der übereinstimmenden Bildpunkte nicht oder nur geringfügig. An dieser Stelle sei angemerkt, dass eine Invarianz von \mathbf{d} hinsichtlich perspektivischer Änderungen nur partiell vorhanden ist. Je höher die perspektivischen Unterschiede sind, desto mehr ändert sich der Deskriptor \mathbf{d} . Die weiteren Eigenschaften der SIFT-Merkmalsvektoren werden später in diesem Unterkapitel vorgestellt.



Abbildung 2.14: Schachtel aus der ersten Perspektive



Abbildung 2.15: Schachtel aus der zweiten Perspektive

Anhand eines Vergleichs der Merkmalsvektoren besteht die Möglichkeit übereinstimmende Bildpunkte in beiden Ansichten zu ermitteln. Das Prinzip eines übereinstimmenden Bildpunktes wird in Grafik 2.7 gezeigt. Ein Raumpunkt P wird auf die korrespondierenden Bildpunkte p^* und p in den entsprechenden Ansichten B^* und B abgebildet. Wenn nun ein Merkmalsvektor d^* aus der ersten Ansicht 2.14 und ein Merkmalsvektor d aus der zweiten Ansicht 2.15 äquivalent oder annähernd gleich sind, dann handelt es sich bei den zugehörigen SIFT-Punkten mit

hoher Wahrscheinlichkeit um korrespondierende Bildpunkte. Ein Vergleich von Merkmalsvektoren kann hierbei zum Beispiel über den *Nearest-Neighbor*-Algorithmus stattfinden, welcher die Distanz zwischen zwei Vektoren bestimmt.

Dieses Vorgehen wurde exemplarisch für die SIFT-Punkte aus den Ansichten 2.14 und 2.15 durchgeführt. Anhand eines Vergleichs der Merkmalsvektoren aus beiden Ansichten wurden die in Grafik 2.16 gezeigten korrespondierenden Bildpunkte ermittelt, wobei die Ansichten 2.14 und 2.15 überlagert dargestellt werden. Die übereinstimmenden Bildpunkte wurden durch blaue Linien miteinander verbunden.

Die durch den *Nearest-Neighbor*-Algorithmus detektierten korrespondierenden Bildpunkte können mit Ausreißern behaftet sein. Hierbei handelt es sich um Bildpunkte die nicht übereinstimmen. Um diese falschen Übereinstimmungen herauszufiltern sind weitere Berechnungen notwendig, die an dieser Stelle nicht weiter vertieft werden. Die dargestellten Punkte aus Grafik 2.16 wurden hierbei bereits von Ausreißern bereinigt, um das Prinzip von übereinstimmenden Bildpunkten verdeutlichen zu können.



Abbildung 2.16: Übereinstimmende Bildpunkte bezüglich der überlagerten Bilder 2.14 und 2.15

Nachstehend werden die speziellen Eigenschaften der SIFT-Merkmalsvektoren beschrieben. Eine Einschränkung der SIFT-Methode ist, dass markante Bildpunkte nur in texturierten Bildbereichen detektiert werden können. Im Fall einer homogenen Fläche können somit keine SIFT-Punkte gefunden werden. SIFT-Merkmalsvektoren stellen invariante Merkmale dar, wie am erläuterten Beispiel in Grafik 2.16 verdeutlicht wurde. Die Merkmalsvektoren haben sich trotz einer Änderung der Perspektive nicht oder nur geringfügig verändert.

In ([LOW04], S. 1) wird erläutert, dass SIFT-Merkmalsvektoren bezüglich Rotation und Skalierung im Bild invariant sind. Weiterhin besteht eine partielle Invarianz gegenüber affiner Verzerrung, Änderung der Perspektive, Bildrauschen und Änderung der Helligkeit. Das

Bildrauschen ist hier als Abweichung der Intensität einzelner Pixel zu verstehen. Dies wird zum Beispiel durch das *Salt-and-Pepper*-Rauschen verursacht. Folglich bietet das SIFT-Verfahren eine robuste Möglichkeit zur Detektion von korrespondierenden Bildpunkten in realen wechselhaften Umgebungen.

In Grafik 2.16 wird eine weitere Eigenschaft des SIFT-Verfahrens deutlich. Es ist zu erkennen, dass die Schachtel aus Abbildung 2.15 am unteren Rand verzerrt dargestellt wird. Dies resultiert aus der Linsenverzeichnung der verwendeten Kamera. Die übereinstimmenden Bildpunkte aus Abbildung 2.16 zeigen, dass SIFT im Fall einer Linsenverzeichnung robuste Merkmale berechnet.

2.2.2 Weitere Verfahren

Neben der SIFT-Methode gibt es weitere Möglichkeiten zur Beschreibung von markanten Bildpunkten. In ([MS05], S. 5) wird das *Gradient Location and Orientation Histogram* (GLOH)-Verfahren erläutert, welches eine Erweiterung von SIFT darstellt. In einem Experiment wurde ermittelt, dass GLOH möglicherweise eine erhöhte Robustheit im Vergleich zu SIFT bietet. In dem Experiment wurde ein Objekt aus zwei unterschiedlichen Perspektiven aufgenommen und anschließend wurde eine Korrespondenzanalyse mit Hilfe des *Nearest-Neighbor*-Algorithmus ausgeführt. Dabei hat sich herausgestellt, dass GLOH geringfügig mehr korrekte Übereinstimmungen bezüglich der Bildpunkte liefert. Um die Genauigkeit von SIFT und GLOH eingehender vergleichen zu können, müssen laut ([MS05], S. 15) mehr Messungen durchgeführt werden. Es ist allerdings bekannt, dass die Berechnungen des GLOH-Verfahrens zeitaufwändiger sind als die der SIFT-Methode (vgl. [BETVG08], S. 5).

Weiterhin können markante Bildpunkte mit Hilfe des in [BETVG08] vorgestellten *Speeded-Up Robust Features* (SURF)-Verfahrens detektiert werden, welches ebenfalls eine Weiterentwicklung des SIFT-Algorithmus darstellt. In [BSP07] wird ein Vergleich von SIFT und SURF hinsichtlich der Robustheit und der Laufzeiteffizienz vorgenommen. Dabei werden die Eigenschaften der Verfahren hinsichtlich der Invarianz gegenüber den in Abschnitt 2.2.1 erläuterten Einflüssen berücksichtigt. Dazu wurde jeweils eine Messreihe für die Einflüsse Rotation, Skalierung, Änderung der Perspektive, Bildrauschen und Änderung der Helligkeit aufgenommen. In jeder Messreihe wurden die Einflüsse schrittweise verstärkt. In jedem Schritt wurde hierbei jeweils eine Korrespondenzanalyse für SIFT und SURF mit Hilfe des *Nearest-Neighbor*-Algorithmus durchgeführt, wobei das aktuelle Bild und das Ausgangsbild als Eingangsdaten dienten.

Die Ergebnisse dieser Messreihen zeigen, dass SIFT in fast allen Fällen deutlich mehr richtige Korrespondenzen bezüglich der Bildpunkte liefert. In wenigen Ausnahmen konnten auf Basis von SURF mehr korrekte Übereinstimmungen gefunden werden. Insgesamt ist SIFT also robuster als SURF. Allerdings wird für die Berechnung von SURF-Merkmalvektoren wesentlich weniger Rechenzeit benötigt als für SIFT-Merkmalvektoren (sh. [BETVG08], S. 1).

Auf der Grundlage der hier erläuterten Ergebnisse, lässt sich für die Korrespondenzanalyse mittels markanter Bildpunkte folgende Schlussfolgerung formulieren. Eine steigende Robustheit bringt eine erhöhte Rechenzeit mit sich, wobei die Robustheit durch die Anzahl der korrekten Bildpunkt-Korrespondenzen beschrieben wird.

3 Konzept

Im Folgenden wird das Konzept für das kamerabasierte Greifen von ungeordneten Gegenständen erläutert, welches im Rahmen dieser Masterarbeit entstanden ist. Als Grundlage dienen dabei die Erkenntnisse hinsichtlich der vergleichbaren Arbeiten aus Kapitel 2.

3.1 Übersicht

In Abschnitt 2.1 wurden Verfahren für das Greifen von ungeordneten Gegenständen auf Basis von Kameradaten untersucht. Ein Greifvorgang wurde hierbei in die nachstehenden Schritte unterteilt.

1. **Positionierung:** Eine Roboterhand wird auf Basis von Kameradaten in eine Greiflage bezüglich eines Gegenstands positioniert.
2. **Greifen:** Ausgehend von einer Greiflage wird ein Objekt durch eine Roboterhand gegriffen. Bestenfalls muss die Hand in der Greiflage lediglich geschlossen werden.

Bei der Untersuchung der Verfahren zur Positionierung hat sich herausgestellt, dass die direkte Steuerung eine ungenaue Methode darstellt. Dies kann dazu führen, dass die durch die Positionierung erreichte Greiflage von der vorgegebenen Greiflage abweichen kann. Das anschließende Greifen eines Gegenstands kann somit fehlschlagen.

Im Gegensatz dazu bietet das VS auf Grund des regelbasierten Vorgehens eine genauere aber auch langsamere Positionierung des Endeffektors in eine Greiflage. Des Weiteren hat sich ergeben, dass die meisten der vorgestellten VS-Verfahren nur unter bestimmten Bedingungen einwandfrei funktionieren.

In der vorliegenden Arbeit wird ein kombiniertes Vorgehen zur Positionierung angestrebt, um die Restriktionen der einzelnen Positionierungsverfahren kompensieren zu können. Der Ansatz besteht hierbei darin, das Prinzip der direkten Steuerung mit einer anschließenden Regelung zu koppeln. Letzteres soll anhand der bildpunktbasierter Regelung aus Abschnitt 2.1.2.2 realisiert werden. Auf diese Weise soll insgesamt eine genaue Einstellung der erwähnten Greiflage ermöglicht werden. Außerdem kann ein Greifvorgang mit einer höheren Geschwindigkeit ausgeführt werden, als im Fall einer rein regelbasierten Methode. Nachfolgend wird der prinzipielle Ablauf des Hybrid-Ansatzes erläutert.

Für eine direkte Steuerung zur Positionierung des Endeffektors wird üblicherweise ein vollständiges 3D-Modell eines zu greifenden Objekts benötigt. In dem hier erläuterten kombinierten Verfahren wird die direkte Steuerung hingegen auf Basis einer Teilrekonstruktion des Objekts durchgeführt. Diese Teilrekonstruktion soll mittels zweier Ansichten erstellt werden, welche die gleiche Szene zeigen. Dieser Aspekt wird in Grafik 2.7 skizziert. Beide Ansichten werden hierbei vorab in einem *Teach-In* Schritt aufgenommen.

Eine dieser beiden Ansichten stellt das Zielbild B^* dar, welches ebenfalls für die bildpunktbaasierte Regelung verwendet wird (vgl. Kapitel 2.1.2.2). B^* zeigt einen zu greifenden Gegenstand so, wie dieser in der Greiflage von der Kamera gesehen werden soll. Auf der Grundlage einer Teilrekonstruktion und B^* kann somit eine Greiflage bezüglich des Gegenstands berechnet werden (sh. Abschnitt 2.1.1), welche im ersten Schritt der Positionierung, während der

direkten Steuerung, als Vorgabe benötigt wird. Der Ablauf des Hybrid-Ansatzes wird in die nachstehende Sequenz aufgeteilt.

1. **Direkte Steuerung:** Anhand der Referenzansicht B^* kann ein zu greifendes Objekt im aktuellen Kamerabild wiedererkannt werden. Daraufhin kann mit Hilfe der Teilrekonstruktion des Objekts die aktuelle Objektlage relativ zur Kamera ermittelt werden. Dies ermöglicht die Bewegung des Roboters in Richtung der vorgegebenen Greiflage in einem einzigen Schritt. Aus den in Kapitel 2.1.1 beschriebenen Gründen wird die durch die direkte Steuerung erreichte Greiflage von der vorgegebenen Greiflage abweichen.
2. **Bildpunktbasierte Regelung:** Um diese Abweichung zu kompensieren, wird im Anschluss die bildpunktbasierte Regelung aus Kapitel 2.1.2.2 durchgeführt. Auf diese Weise wird das aktuelle Bild in das Zielbild B^* überführt. In diesem Fall befindet sich der Endeffektor in der vorgegebenen Greiflage.

Mittels der bildpunktbasierten Regelung kann die Ungenauigkeit der direkten Steuerung ausgeglichen werden. Dies wird gewährleistet, da die bildpunktbasierte Regelung robust gegenüber Kalibrierungsfehlern und Bildrauschen ist. Umgekehrt sorgt die direkte Steuerung dafür, dass eine Schwäche der bildpunktbasierten Regelung kompensiert wird. Dabei geht es um die Problematik des unter Gliederungspunkt 2.1.2.2 beschriebenen *camera retreat*, welcher zum Abbruch der Regelung führen kann.

Ein *camera retreat* tritt nur bei hohen Verdrehungen um die optische Achse auf. Die Verdrehung der Kamera bezieht sich dabei auf die Kamera in der Ziellage, welche zur Aufnahme der Zielansicht B^* verwendet wurde. Da die Abweichung zwischen der tatsächlichen und der vorgegebenen Greiflage nach der direkten Steuerung nur gering ist, treten während der Regelung somit keine hohen Verdrehungen um die optische Achse auf. Ein *camera retreat* kann folglich ausgeschlossen werden.

In Abschnitt 2.1.2.2 wurde erläutert, dass für die bildpunktbasierte Regelung der Tiefenabstand der nach s abgebildeten Objektpunkte relativ zur Kamera benötigt wird. Diese Daten sind hier anhand der beschriebenen partiellen Rekonstruktion eines Objekts bereits gegeben.

Nach der Positionierung soll das Greifen eines Objekts ausgehend von der Greiflage eingeleitet werden. Zur Ermittlung der hierfür notwendigen Parameter kann eine der Methoden angewendet werden, die in Abschnitt 2.1.3 untersucht wurden. Allerdings wird die zur Positionierung benötigte Referenzansicht B^* *offline* in einem *Teach-In* Schritt aufgenommen. Aus diesem Grund sollen die übrigen Greifparameter auf die gleiche Weise wie folgt bestimmt werden.

Da die durch B^* repräsentierte Greiflage mit Hilfe der bildpunktbasierten Regelung exakt eingestellt werden kann, können anschließend vordefinierte Bewegungen zum Greifen eines Objekts ausgeführt werden. Diese Greifbewegungen werden durch Abbildung 2.11 veranschaulicht. Ein zu greifender Gegenstand muss für den gesamten Zeitraum der bildbasierten Regelung im Kamerabild zu sehen sein. Somit kann es vorkommen, dass sich der Endeffektor nach der Positionierung in einer Lage befindet, von der aus eine weitere Bewegung notwendig ist, damit der Gegenstand gegriffen werden kann (sh. Grafik 2.11 a). Dabei handelt es sich um die Einstellung einer Pose, die vorab während eines *Teach-In* Vorgangs gespeichert wird.

Nach dieser zusätzlichen Bewegung muss die Roboterhand lediglich geschlossen werden (sh. Grafik 2.11 b). Wie weit sich die Hand schließen muss, um ein spezifisches Objekt greifen zu

können, wird ebenfalls vorab in einem *Teach-In* Schritt ermittelt und zur späteren Verwendung gesichert.

Der beschriebene *Teach-In* Schritt birgt den Nachteil, dass die Greifparameter nur manuell durch einen Benutzer ermittelt werden können. Eine weitere Einschränkung des hier vorgestellten Konzepts ist, dass ein Objekt nur bezüglich der Referenzansicht B^* wiedererkannt und gegriffen werden kann. Um dieses Problem zu umgehen, könnte das Verfahren allerdings mit geringem Aufwand auf mehrere Zielansichten und Teilrekonstruktionen rundum einen Gegenstand erweitert werden. Dies wird in der vorliegenden Arbeit jedoch nicht berücksichtigt. Die Eigenschaften des dargelegten Konzepts werden in der nachstehenden Tabelle zusammengefasst.

Vorteile	Nachteile
<ul style="list-style-type: none"> • Robust gegenüber Kalibrierungsfehlern und Bildrauschen. • Es werden nur wenig Informationen über einen Gegenstand benötigt, der gegriffen werden soll. Insbesondere wird kein vollständiges 3D-Modell eines Gegenstands gebraucht. • Ein Greifvorgang kann mit einer höheren Geschwindigkeit ausgeführt werden, als im Fall einer rein regelbasierten Methode. 	<ul style="list-style-type: none"> • Manueller <i>Teach-In</i> Vorgang zur Bestimmung der Greifparameter notwendig. • Ein Objekt kann nur bezüglich der Referenzansicht B^* wiedererkannt und gegriffen werden. Durch eine Erweiterung auf mehrere Referenzansichten und partielle Rekonstruktionen rundum einen Gegenstand kann dieser Nachteil beseitigt werden (wird hier jedoch nicht berücksichtigt).

Tabelle 3.1: Eigenschaften des Greifkonzepts

3.2 Systemkomponenten

Zur Realisierung des erläuterten Vorgehens sollen sukzessive einzelne Systemkomponenten entwickelt werden, welche als nächstes festgelegt werden. Als Ausgangspunkt dient hierfür der konzipierte Ablauf eines Greifvorgangs, der in Grafik 3.1 zusammengefasst wird.

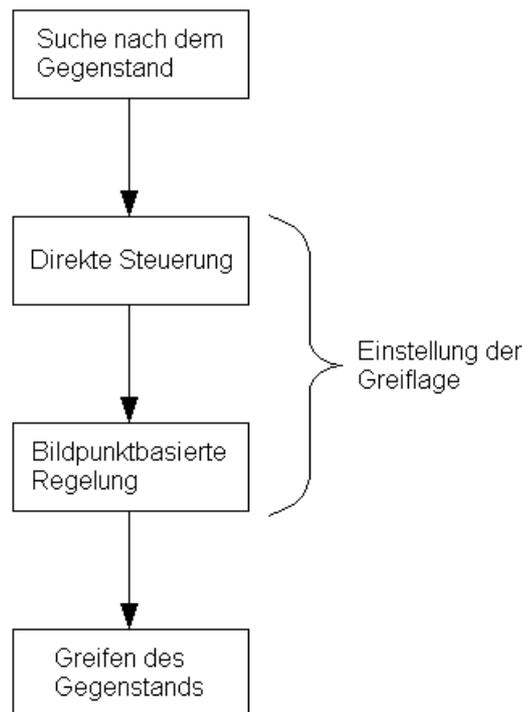


Abbildung 3.1: Ablauf eines Greifvorgangs

Ein Greifvorgang setzt sich hier somit aus den folgenden Schritten zusammen, wobei im weiteren Verlauf die bisher vernachlässigte initiale Suche nach einem Gegenstand berücksichtigt werden muss.

1. **Suche nach dem Gegenstand:** Bevor eine Positionierung der Roboterhand durchgeführt werden kann, muss sich zunächst der zu greifende Gegenstand im Blickfeld der Kamera befinden. Um dies zu erreichen sucht der Roboter seinen Arbeitsraum vorab nach einem bestimmten Muster ab, bis der Gegenstand wiedererkannt wird. Eine Wiedererkennung wird hierbei durch einen Abgleich der aktuellen Ansicht B und der Referenzansicht B^* ermöglicht.
2. **Direkte Steuerung:** Anschließend kann die aktuelle Gegenstandspose relativ zur Kamera anhand der Teilrekonstruktion des Gegenstands ermittelt werden. Hierauf erfolgt die Bewegung der Roboterhand in Richtung der vorgegebenen Greiflage in einem einzigen Schritt. Die direkte Steuerung wird durch Abbildung 3.2 dargestellt.
3. **Bildpunkt-basierte Regelung:** Die durch die direkte Steuerung erreichte Greiflage wird aus den erwähnten Gründen von der vorgegebenen Greiflage abweichen. Deshalb wird anschließend eine genaue Einstellung der Greiflage durch eine Regelung vorgenommen (vgl. Grafik 2.6), wobei B nach B^* überführt wird.
4. **Greifen:** Ausgehend von der Greiflage soll der Gegenstand anhand der im *Teach-In* Vorgang aufgenommenen Greifparameter gegriffen werden. Zunächst wird dazu eine vordefinierte Bewegung bezüglich der Greiflage ausgeführt. Abschließend wird die Hand des Roboters geschlossen, um den Gegenstand zu erfassen.

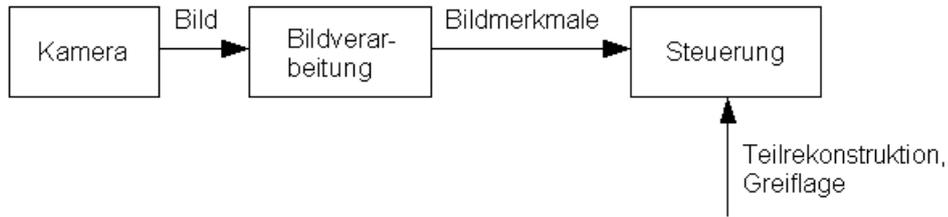


Abbildung 3.2: Direkte Steuerung

Die zu entwickelnden Systemkomponenten können wie folgt festgelegt werden. Für die Suche nach einem Gegenstand, muss dessen Wiedererkennung ermöglicht werden. Eine Wiedererkennung soll dabei anhand eines Abgleichs der aktuellen Ansicht B und der Referenzansicht B^* umgesetzt werden. Hierfür bietet sich die Korrespondenzanalyse aus Kapitel 2.2 an, welche einen Ansatz zur Erfassung von übereinstimmenden Punkten in perspektivisch unterschiedlichen Bildern zur Verfügung stellt. Um eine zuverlässige Detektion von korrespondierenden Bildpunkten zu realisieren, muss dieser Ansatz jedoch weiterentwickelt werden.

Mit Hilfe der Korrespondenzanalyse kann auch eine Teilaufgabe der bildpunktbasierteren Regelung aus Grafik 2.6 gelöst werden. Anhand der korrespondierenden Bildpunkte aus B und B^* sind die Ist-Größe s und die Soll-Größe s^* des Regelkreises gegeben. Eine weitere Systemkomponente ist durch die Rekonstruktion von Raumpunkten bezüglich zwei Ansichten gegeben, die für die Generierung eines Teilmodells eines Gegenstands notwendig ist.

Für das erläuterte hybride Positionierungsverfahren werden mehrere Module benötigt. Die direkte Steuerung setzt die Bestimmung der aktuellen Gegenstandspose relativ zur Kamera voraus. Diese Problematik wird nachstehend zusammenfassend als Lokalisierung bezeichnet. Für die direkte Steuerung wird weiterhin eine Einheit benötigt, welche die Einstellung der vorgegebenen Greiflage ermöglicht. Dieses Modul wird der Einfachheit halber als Steuereinheit bezeichnet.

Eine weitere Komponente des Systems ist durch den bildpunktbasierteren Regler aus Grafik 2.6 gegeben, der den Regelfehler e auf eine Geschwindigkeit v abbildet. Mit Hilfe von v wird die Bewegung des Endeffektors in Richtung der Greiflage ermöglicht.

Das vorgestellte kombinierte Positionierungsverfahren setzt voraus, dass die Roboter-Plattform anhand von räumlichen Lageinformationen gesteuert werden kann. Aus diesem Grund müssen die kinematischen Zusammenhänge des Katana-Greifarms aus Grafik 1.1 modelliert werden. Dabei handelt es sich um die Abhängigkeiten zwischen den Gelenkwinkeln des Roboterarms und den entsprechenden Lageinformationen. Insgesamt müssen somit die folgenden sechs Systemkomponenten entwickelt werden.

- Kinematik des Roboters,
- Lokalisierung eines Gegenstands bezüglich der Kamera,
- Steuereinheit für die direkte Steuerung,
- Bildpunktbasierter Regler,
- Korrespondenzanalyse,

- Rekonstruktion von Raumpunkten.

Im weiteren Verlauf dieser Arbeit werden die aufgelisteten Systemkomponenten in der gleichen Reihenfolge modelliert. Anschließend erfolgt eine Synthese der Komponenten zum Gesamtsystem, welches simulativ und experimentell auf Basis des Katana-Roboters verifiziert wird.

4 Hardware

Um die erläuterten Systemkomponenten modellieren zu können, muss zunächst die verwendete Hardware vorgestellt werden. Dabei handelt es sich um den Katana-Greifarm und die Kamera, die in der Nähe des Endeffektors des Roboters angebracht wurde (vgl. Abbildung 1.1). Zur Laufzeit werden die Kameradaten an einen PC gesendet und anschließend von einem Programm zur Berechnung von Steuerbefehlen ausgewertet. Die Steuerdaten werden an den Roboterarm geschickt, um den Endeffektor in Richtung eines zu greifenden Gegenstands zu bewegen.

4.1 Katana-Roboterarm

Dieses Unterkapitel umfasst eine Beschreibung der Beschaffenheit und der Steuerungsmöglichkeiten des Katana-Greifarms. Außerdem erfolgt eine Erläuterung der Anwendungsbereiche in denen der Roboter eingesetzt werden kann.

4.1.1 Beschaffenheit

Der Katana-Roboterarm vom Typ 450-6M180 besitzt eine anthropomorphe Struktur mit fünf FHG. Dabei handelt es sich um fünf Drehgelenke g_0 bis g_4 , die durch eine serielle Kette von Armgliedern miteinander verbunden sind. Zur Veranschaulichung des Aufbaus der einzelnen Gelenke werden deren Drehachsen z_0 bis z_4 in Abbildung 4.1 dargestellt. Anhand der fünf Drehgelenke kann eine Lage des Endeffektors relativ zu einem Bezugs-Koordinatensystem eingestellt werden. Die Zusammenhänge zwischen den entsprechenden Gelenkwinkeln und der Lage des Endeffektors werden in Abschnitt 5 näher erläutert.

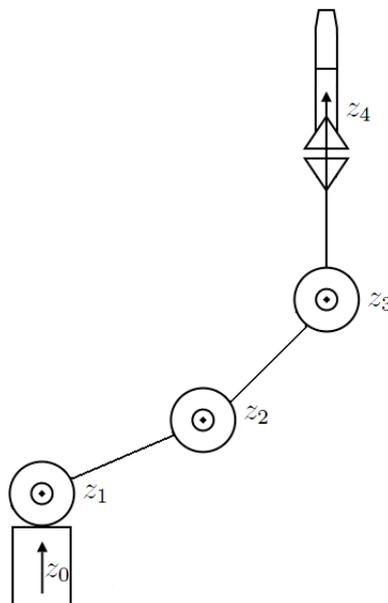


Abbildung 4.1: Drehachsen des Katana-Greifarms

Das Greifwerkzeug des Katana-Roboters besteht aus einem Winkelgreifer mit zwei Fingern, welcher mit Hilfe eines weiteren Drehgelenks g_5 bewegt wird. Für einen Greifvorgang stehen an der Roboterhand dabei keine taktilen- oder Kraftsensoren zur Verfügung. Die Nutzlast des Roboters beträgt laut Hersteller maximal 400 g, wobei diese Angabe noch nicht das Gewicht des Endeffektors berücksichtigt (vgl. [NEU08]). Da der Winkelgreifer 150 g wiegt, verbleiben somit 250 g für die weitere Nutzlast. Das Gewicht der beiden Finger am Winkelgreifer wird

in den Herstellerangaben nicht spezifiziert. Weiterhin muss das Gewicht der Kamera inklusive des Objektivs von insgesamt 170 g hinzugezogen werden. Diese Last kann allerdings nicht vollständig berücksichtigt werden, da die Kamera hinter der Roboterhand befestigt wurde. Insgesamt ist aber davon auszugehen, dass die übrige maximale Nutzlast für Greifgüter deutlich weniger als 250 g beträgt.

In Abbildung 4.2 wird ein Ausschnitt des Arbeitsraums des Roboters veranschaulicht, wobei die schraffierte Fläche die durch den Greifarm erreichbaren Bereiche vorgibt. Bei der Einheit der Maßangaben handelt es sich hierbei um Millimeter.

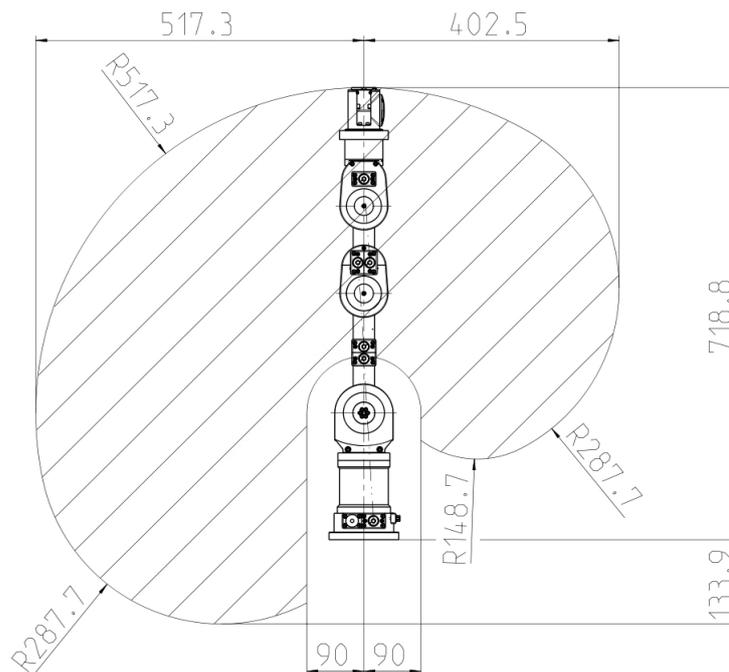


Abbildung 4.2: Arbeitsraum des Katana-Greifarms (Bildquelle: [NEU08])

4.1.2 Steuerungsmöglichkeiten

Im Folgenden werden die Steuerungsmöglichkeiten für den Katana-Roboterarm analysiert. Zur Ansteuerung des Roboters wird vom Hersteller die C++ Software-Bibliothek *Katana Native Interface* (KNI) zur Verfügung gestellt.

Anhand der Drehgelenke g_0 bis g_4 kann eine Pose des Endeffektors relativ zu einem Bezugs-Koordinatensystem eingestellt werden. Um dies zu erreichen besteht die Möglichkeit die Winkel der Gelenke direkt einzustellen. Eine weitere Steuerungsmodalität ist durch die Posen-Steuerung des Endeffektors gegeben, wobei eine Pose aus einer Position und einer Orientierung besteht. Die Position ist durch den sogenannten *Tool Center Point* (TCP) relativ zum Koordinatensystem des Roboters definiert. Das Koordinatensystem des Roboters wird in Grafik 4.3 gezeigt. Hierbei muss berücksichtigt werden, dass sich der Ursprung des Koordinatensystems im Fuß des Roboters befindet. Beim TCP handelt es sich um einen fiktiven Referenzpunkt am Endeffektor. In diesem Punkt wird das Endeffektor-Koordinatensystem aufgespannt, dessen Verdrehung zum Roboter-Koordinatensystem die Orientierung angibt. Die beschriebene Position und Orientierung bilden gemeinsam die Lage des Endeffektors bezüglich des Roboter-Koordinatensystems.

Experimentelle Ergebnisse zeigen, dass die Posen-Steuerung des Endeffektors mittels der KNI-Bibliothek des Herstellers nicht funktioniert, wobei die Bibliothek in der Version 4.0.0 vorliegt. Da das Konzept aus Abschnitt 3 eine Steuerung der Roboter-Plattform anhand von Lageinformationen voraussetzt, muss somit eine eigene Posen-Steuerung des Endeffektors entwickelt werden. Dieser Aspekt wird unter Gliederungspunkt 5 weiterverfolgt.

Die Steuerungsmöglichkeiten für den Winkelgreifer belaufen sich auf eine direkte Einstellung des Winkels des Drehgelenks g_5 .

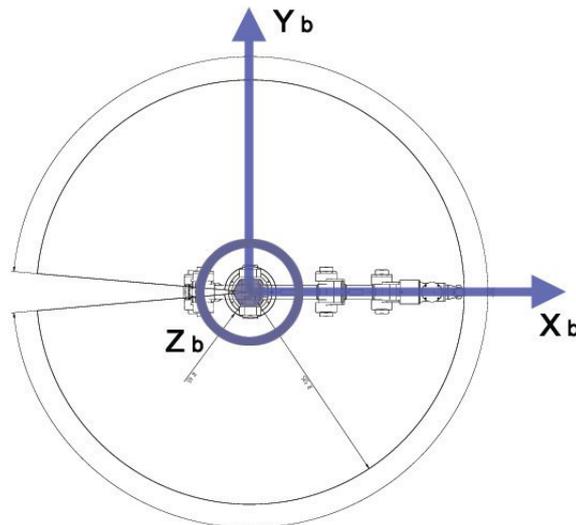


Abbildung 4.3: Koordinatensystem des Roboters (Bildquelle: [NEU08])

4.1.3 Anwendungsbereiche

Auf Grund der Konstellation der Drehgelenke (vgl. Grafik 4.1), können Greifvorgänge nur auf der in Grafik 4.4 a dargestellten Radialebene innerhalb des angedeuteten Bewegungskreises durchgeführt werden. Die Gelenkstruktur des Roboters lässt es hingegen nicht zu, dass die Roboterhand in die Pose aus Abbildung 4.4 b versetzt wird.

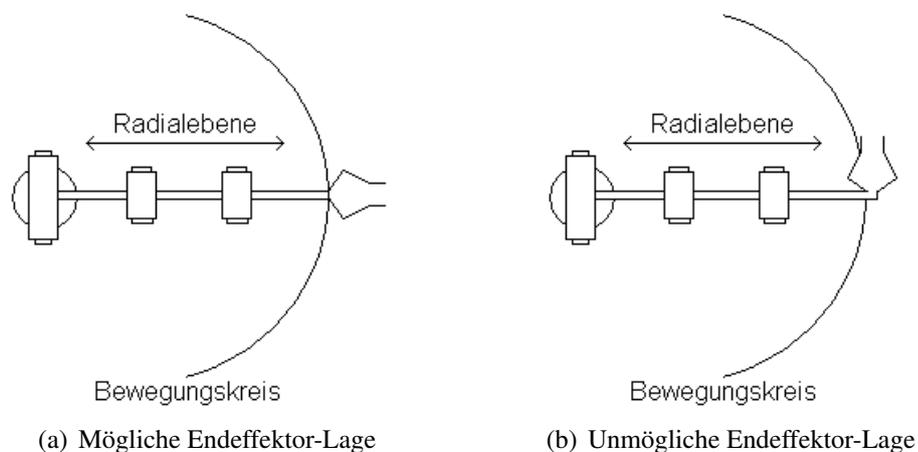


Abbildung 4.4: Katana-Roboter von oben

4.2 Kamera

Bei der Kamera handelt es sich um eine Guppy F-080B der Firma *Allied Vision Technologies*. Ein besonderes Merkmal dieser Kamera ist das geringe Gewicht von 50 g ohne Objektiv (vgl. [ALL]). Die Kamera sendet Schwarz-Weiß-Bilder bei einer maximalen Auflösung von 1032 x 778 Pixel. Weiterhin liefert die Kamera maximal 30 Bilder pro Sekunde.

Für die Kamera wurde ein Weitwinkelobjektiv mit der Bezeichnung H416 (KP) vom Hersteller *PENTAX* gewählt. Das Objektiv besitzt eine feste Brennweite von 4,2 mm und wiegt 120 g. Nachfolgend werden die beschriebenen Eigenschaften der Kamera inklusive des Objektivs zusammengefasst.

Darstellung: Schwarz-Weiß
Maximale Auflösung: 1032 x 778 Pixel
Maximale Bildrate: 30 Bilder pro Sekunde
Brennweite: 4,2 mm
Gesamt-Gewicht: 170 g

Tabelle 4.1: Eigenschaften der Kamera mit Objektiv

5 Modellierung der Roboter-Kinematik

Im Folgenden werden die kinematischen Zusammenhänge des Roboters entwickelt. Hierbei handelt es sich um die Abhängigkeiten zwischen den Gelenkwinkeln des Roboterarms und der Pose des Endeffektors relativ zu einem Bezugs-Koordinatensystem. Zu Beginn dieses Kapitels wird die Pose eines starren Körpers erläutert. Weiterhin wird die Berechnung der Orientierung als Bestandteil einer Pose laut RPY-Winkelkonvention beschrieben. Auf Basis einer Pose können lineare Koordinatentransformationen und differentielle Transformationen durchgeführt werden. Solche Transformationen werden im Verlauf dieser Arbeit in mehreren Situationen benötigt.

Anschließend wird die Berechnung der Endeffektor-Pose bezüglich des Roboter-Koordinatensystems mittels der direkten Kinematik vorgestellt. Dies dient als Grundlage zur Modellierung der inversen Kinematik, welche zur Steuerung des Greifarms verwendet werden soll. Für die nachfolgend aufgeführten Größen werden SI-Einheiten verwendet. Aus diesem Grund wird nicht jeweils auf die Einheit einer Größe eingegangen. Weiterhin werden alle eingeführten Größen zur Übersicht im Symbolverzeichnis am Ende der vorliegenden Arbeit aufgelistet.

5.1 Pose eines starren Körpers

In Abbildung 5.1 wird ein starrer Körper mit seinem orthonormalen Koordinatensystem $(KS)_1$ relativ zu einem orthonormalen Bezugs-Koordinatensystem $(KS)_0$ dargestellt. Eine Verallgemeinerung soll durch die Beschreibung $(KS)_i$ für das Körper-Koordinatensystem und $(KS)_j$ für das Bezugs-Koordinatensystem erfolgen. Die Pose des starren Körpers ist bezüglich $(KS)_j$ definiert und besteht aus einer Position und einer Orientierung (vgl. [SSVO09], S. 39-40). Die Position wird durch den Translationsvektor

$${}^j \mathbf{o}_i = [x \ y \ z]^T \quad (5.1)$$

bestimmt und beschreibt die Verschiebung des Ursprungs von $(KS)_i$ relativ zu $(KS)_j$. Die Orientierung ist durch die (3×3) Rotationsmatrix

$${}^j \mathbf{R}_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (5.2)$$

gegeben und definiert die Rotation von $(KS)_i$ relativ zu $(KS)_j$.

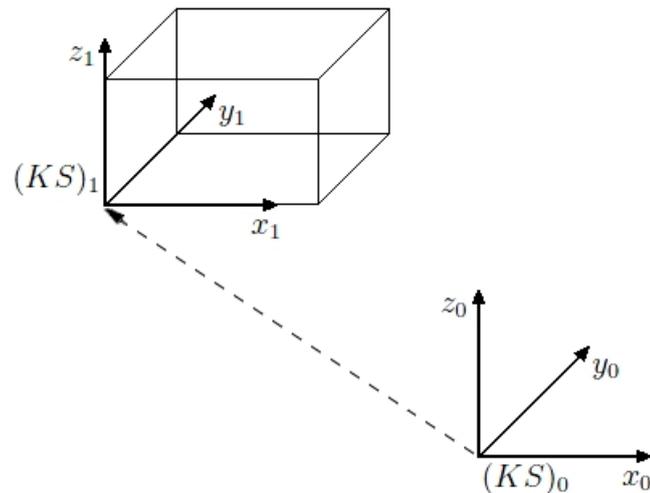


Abbildung 5.1: Pose eines starren Körpers

5.2 Rotationsmatrix

Nachfolgend wird die Berechnung der Rotationsmatrizen für elementare Drehungen beschrieben. Darauf aufbauend erfolgt die Komposition zu einer Gesamt-Rotationsmatrix laut RPY-Winkelkonvention. Dies ermöglicht eine kompakte Darstellung einer Orientierung.

5.2.1 Elementare Rotationen

Die grundlegenden Drehungen ϕ , ψ und θ hinsichtlich der Achsen x , y und z eines Bezugs-Koordinatensystems $(KS)_j$ werden in Abbildung 5.2 dargestellt. Dabei werden die einzelnen Achsen von $(KS)_j$ durch ihre Basisvektoren e_x , e_y und e_z repräsentiert. Die Achsen mit der Bezeichnung $e^{(B)}$ stellen das Koordinatensystem $(KS)_i$ dar, welches relativ zu $(KS)_j$ rotiert ist. Die Winkel ϕ , ψ und θ können dazu genutzt werden, um die Matrizen für die elementaren Rotationen zu definieren (sh. [SSVO09], S. 41-42):

$$\begin{aligned}
 \mathbf{R}_x(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \\
 \mathbf{R}_y(\psi) &= \begin{bmatrix} \cos(\psi) & 0 & \sin(\psi) \\ 0 & 1 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) \end{bmatrix}, \\
 \mathbf{R}_z(\theta) &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}.
 \end{aligned} \tag{5.3}$$

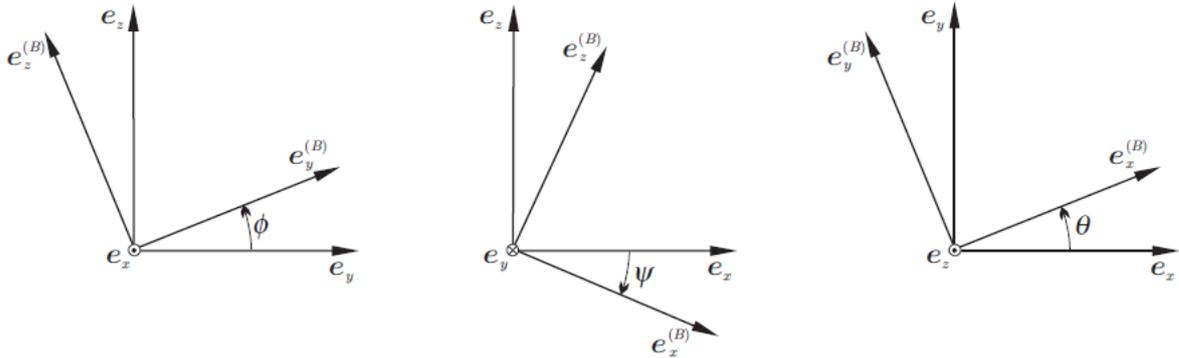


Abbildung 5.2: Elementare Rotationen (Bildquelle: [HEI])

5.2.2 RPY-Winkelkonvention

Eine Möglichkeit zur kompakten Darstellung einer Orientierung ist durch die *Roll-Pitch-Yaw* (RPY) Winkelkonvention, auch bekannt als *ZYX-Eulerwinkel*, gegeben. Dabei werden die elementaren Rotationsmatrizen aus Gleichung 5.3 per Linksmultiplikation mit

$$\boldsymbol{\eta} = [\phi \ \psi \ \theta]^T \quad (5.4)$$

zur Gesamt-Rotationsmatrix

$$\begin{aligned} \mathbf{R}(\boldsymbol{\eta}) &= \mathbf{R}_z(\theta) \mathbf{R}_y(\psi) \mathbf{R}_x(\phi) \\ &= \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi s_\phi - s_\theta c_\phi & c_\theta s_\psi c_\phi + s_\theta s_\phi \\ s_\theta c_\psi & s_\theta s_\psi s_\phi + c_\theta c_\phi & s_\theta s_\psi c_\phi - c_\theta s_\phi \\ -s_\psi & c_\psi s_\phi & c_\psi c_\phi \end{bmatrix} \end{aligned} \quad (5.5)$$

zusammengefasst (vgl. [SSVO09], S. 52). Die Notation $c_{(\cdot)}$ steht für $\cos(\cdot)$ respektive $s_{(\cdot)}$ für $\sin(\cdot)$. Der Zusammenhang der einzelnen Rotationen wird in Abbildung 5.3 gezeigt. Die jeweiligen Drehungen sind dabei relativ zum Bezugs-Koordinatensystem mit folgender Reihenfolge festgelegt.

1. yaw: Rotation um die x -Achse mit ϕ ,
2. pitch: Rotation um die y -Achse mit ψ ,
3. roll: Rotation um die z -Achse mit θ .

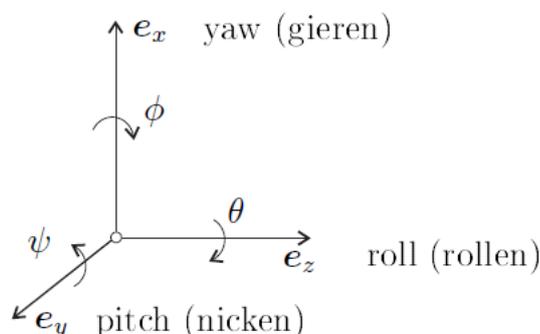


Abbildung 5.3: Rotationen laut RPY-Winkelkonvention (Bildquelle: [HEI])

In den folgenden Abschnitten muss $\boldsymbol{\eta}$ (Formel 5.4) für eine gegebene Rotationsmatrix (sh. Gleichung 5.2) berechnet werden. In ([HOR07], S. 16) wird dafür ein Koeffizientenvergleich

zwischen $\mathbf{R}(\boldsymbol{\eta})$ und der entsprechenden gegebenen Rotationsmatrix ${}^j\mathbf{R}_i$ durchgeführt. Für die Berechnung von $\boldsymbol{\eta}$ ergeben sich folgende Definitionen:

$$\begin{aligned}\theta &= \text{atan2}\left(\frac{r_{21}}{r_{11}}\right), \\ \psi &= \text{atan2}\left(\frac{-r_{31}}{c_\theta r_{11} + s_\theta r_{21}}\right), \\ \phi &= \text{atan2}\left(\frac{s_\theta r_{13} - c_\theta r_{23}}{c_\theta r_{22} - s_\theta r_{12}}\right).\end{aligned}\quad (5.6)$$

5.3 Lineare Koordinatentransformation

Nachfolgend wird die lineare Koordinatentransformation durch Translation und Rotation auf Basis der Pose aus Gliederungspunkt 5.1 beschrieben. Dies ermöglicht den Wechsel des Koordinatensystems bezüglich dessen ein Punkt $\mathbf{p} = [x \ y \ z]^T$ dargestellt wird. Die lineare Koordinatentransformation kann dabei durch die Verwendung von homogenen anstelle von euklidischen Koordinaten vereinfacht anhand einer Matrixmultiplikation ausgedrückt werden. Die Koordinaten eines homogenen Punktes $\tilde{\mathbf{p}}^i$ bezüglich $(KS)_i$ werden mittels der homogenen (4×4) Transformationsmatrix ${}^j\mathbf{T}_i$ in die Koordinaten des homogenen Punktes $\tilde{\mathbf{p}}^j$ relativ zum Bezugs-Koordinatensystem $(KS)_j$ transformiert ([SSVO09], S. 57):

$$\tilde{\mathbf{p}}^j = {}^j\mathbf{T}_i \tilde{\mathbf{p}}^i = \begin{bmatrix} {}^j\mathbf{R}_i & {}^j\mathbf{o}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \tilde{\mathbf{p}}^i. \quad (5.7)$$

Die Herleitung eines homogenen Punktes wird dabei unter Gliederungspunkt A.1 beschrieben. Eine umgekehrte Transformation erfolgt durch die Invertierung von ${}^j\mathbf{T}_i$ in der Gleichung

$$\tilde{\mathbf{p}}^i = ({}^j\mathbf{T}_i)^{-1} \tilde{\mathbf{p}}^j = {}^i\mathbf{T}_j \tilde{\mathbf{p}}^j. \quad (5.8)$$

Da \mathbf{R} eine orthogonale Matrix ist, ergibt sich für ${}^i\mathbf{T}_j$:

$${}^i\mathbf{T}_j = \begin{bmatrix} ({}^j\mathbf{R}_i)^T & -({}^j\mathbf{R}_i)^T {}^j\mathbf{o}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} {}^i\mathbf{R}_j & -{}^i\mathbf{R}_j {}^j\mathbf{o}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}. \quad (5.9)$$

5.4 Differentielle Transformation

Die Geschwindigkeit eines Koordinatensystems $(KS)_j$ ist gegeben durch

$${}^j\mathbf{v}_j = [\dot{x} \ \dot{y} \ \dot{z} \ w_x \ w_y \ w_z]^T. \quad (5.10)$$

Hierbei steht $[w_x \ w_y \ w_z]^T$ für die voneinander unabhängigen Winkelgeschwindigkeiten um die jeweiligen Achsen. Diese Repräsentation der Winkel ist somit anders als die RPY-Winkelkonvention aus Kapitel 5.2.2. Das Bezugs-Koordinatensystem $(KS)_j$ eines Geschwindigkeitsvektors ${}^j\mathbf{v}_j$ kann mit Hilfe der Transformationsmatrix ${}^i\mathbf{V}_j$ nach $(KS)_i$ gewechselt werden ([HOR07], S. 72):

$${}^i\mathbf{v}_j = {}^i\mathbf{V}_j ({}^i\mathbf{R}_j) {}^j\mathbf{v}_j = \begin{bmatrix} {}^i\mathbf{R}_j & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}^i\mathbf{R}_j \end{bmatrix} {}^j\mathbf{v}_j. \quad (5.11)$$

5.5 Denavit-Hartenberg Konvention

Die *Denavit-Hartenberg* (DH)-Konvention erlaubt eine minimale Beschreibung der Pose aus Kapitel 5.1 über vier sogenannte DH-Parameter d , θ , a und α . Diese eignen sich besonders dazu, um die Lage der einzelnen Komponenten eines seriellen Mehrkörpersystems (MKS) zu berechnen, welches aus aufeinander folgenden Gelenken j_1 bis j_n besteht. Auf diese Weise können die Abhängigkeiten zwischen den Gelenkwinkeln des Roboterarms und der Pose des Endeffektors relativ zum Roboter-Koordinatensystem hergestellt werden.

Zur Bestimmung der DH-Parameter des Gelenks j_{i+1} bezüglich des Gelenks j_i , wird jedes Gelenk j_i anhand eines Gelenk-Koordinatensystems $(KS)_{i-1}$ modelliert (vgl. Abbildung 5.4). Weiterhin müssen die nachstehenden Vorbereitungen getroffen werden (sh. [HOR07], S. 14-15).

1. Die Achse eines Gelenks j_i ist durch die Achse z_{i-1} des Koordinatensystems $(KS)_{i-1}$ gegeben.
2. Um den Ursprung von $(KS)_i$ ermitteln zu können, muss die gemeinsame Normale der Achsen z_{i-1} und z_i bestimmt werden. Der Schnittpunkt der gemeinsamen Normalen mit der Achse z_i definiert den Ursprung von $(KS)_i$.
3. Die Achse x_i liegt auf der gemeinsamen Normalen, wie in Grafik 5.4 dargestellt. Da es sich um ein Rechte-Hand-Koordinatensystem handelt, muss die Achse y_i zur Vervollständigung von $(KS)_i$ entsprechend dargestellt werden.

Anschließend können die DH-Parameter wie folgt bestimmt werden (sh. Grafik 5.4).

- d_i gibt die Verschiebung bezüglich der Achse z_{i-1} an.
- θ_i definiert den Drehwinkel um die Achse z_{i-1} .
- a_i beschreibt die Länge der gemeinsamen Normalen der Achsen z_{i-1} und z_i .
- α_i gibt den Drehwinkel um die Achse x_{i-1} an.

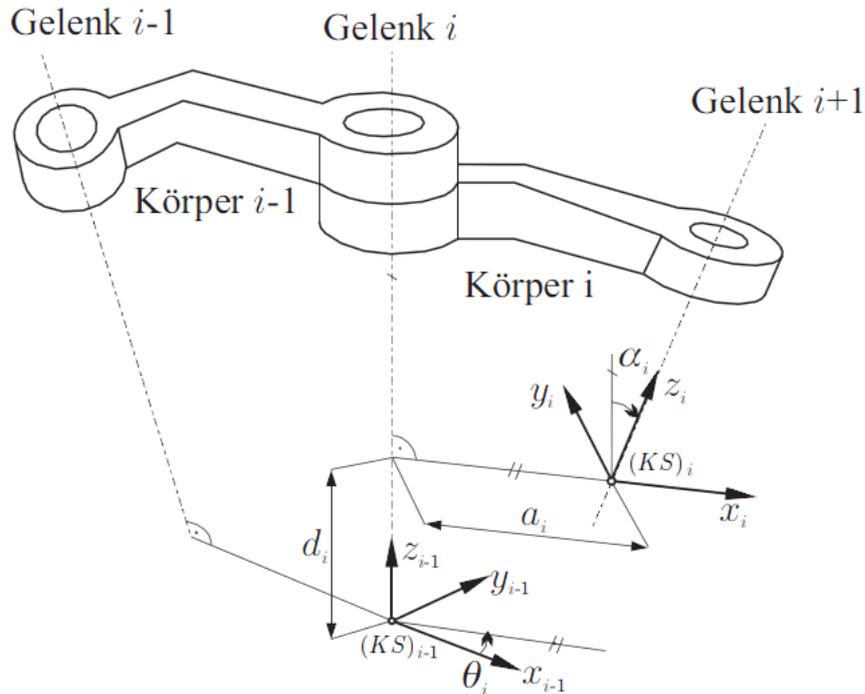


Abbildung 5.4: DH-Parameter

Mit Hilfe der DH-Parameter kann die homogene Transformationsmatrix aus Gleichung 5.7 auch wie folgt definiert werden:

$${}^{i-1}\mathbf{T}_i(q_i) = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.12)$$

Die Größe q_i stellt dabei die Gelenkstellung dar. Im Fall eines Drehgelenks ist $\theta_i = q_i$ und im Fall eines Schubgelenks ist $d_i = q_i$. Im Folgenden werden die Gelenkstellungen mit

$$\mathbf{q} = [q_1 \ q_2 \ \cdots \ q_{n-1} \ q_n]^T. \quad (5.13)$$

bezeichnet.

5.6 Direkte Kinematik

Die direkte Kinematik ermöglicht die Berechnung der Endeffektor-Pose eines Greifarms bezüglich des Roboter-Koordinatensystems $(KS)_0$ in Abhängigkeit der Gelenkstellungen \mathbf{q} . Die Endeffektor-Pose wird dabei folgendermaßen wie unter Gliederungspunkt 4.1.2 definiert. Die Position beschreibt den TCP des Endeffektors relativ zu $(KS)_0$. Die Orientierung gibt die Rotation des Endeffektor-Koordinatensystems $(KS)_e$ bezüglich $(KS)_0$ an. Im Folgenden wird das Roboter-Koordinatensystem $(KS)_0$ auch als Umwelt-Koordinatensystem bezeichnet.

Durch Rechtsmultiplikation der Transformationsmatrizen, die mittels der DH-Parameter berechnet werden, erfolgt die Beschreibung der Gesamt-Transformation ${}^0\mathbf{T}_e$ zwischen dem Umwelt-Koordinatensystem $(KS)_0$ und dem Koordinatensystem des Endeffektors $(KS)_e$ (sh. [SSVO09], S. 61):

$${}^0\mathbf{T}_e(\mathbf{q}) = {}^0\mathbf{T}_1(q_1) {}^1\mathbf{T}_2(q_2) \cdots {}^{n-2}\mathbf{T}_{n-1}(q_{n-1}) {}^{n-1}\mathbf{T}_e(q_n) \quad (5.14)$$

Diese Gesamt-Transformation beinhaltet die Endeffektor-Pose ${}^0\mathbf{p}_e$ bezüglich $(KS)_0$ (vgl. Abschnitt 5.3), welche sich aus den Komponenten

$${}^0\mathbf{p}_e = [x \ y \ z \ \phi \ \psi \ \theta]^T \quad (5.15)$$

zusammensetzt. Die RPY-Winkel ϕ , ψ und θ werden, wie in Formel 5.6 gezeigt, aus dem Rotationsanteil von ${}^0\mathbf{T}_e$ ermittelt.

5.7 Differentielle Kinematik

Mit Hilfe der differentiellen Kinematik können die Geschwindigkeiten der Gelenkwinkel $\dot{\mathbf{q}}$ auf die Geschwindigkeit des Endeffektors bezüglich des Umweltkoordinatensystems abgebildet werden. Die Herleitung der differentiellen Kinematik kann über die zeitliche Ableitung der direkten Kinematik aus Formel 5.15 vorgenommen werden:

$${}^0\dot{\mathbf{p}}_e = \frac{\Delta {}^0\mathbf{p}_e(\mathbf{q})}{\Delta t} = \frac{\partial {}^0\mathbf{p}_e(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_A(\mathbf{q}) \dot{\mathbf{q}}. \quad (5.16)$$

Die Größe $\mathbf{J}_A(\mathbf{q})$ stellt dabei die analytische Jacobimatrix der Kinematik dar. Da die in ${}^0\dot{\mathbf{p}}_e$ enthaltenen zeitlichen Ableitungen von Drehungen um bereits gedrehte Achsen keine physikalische Bedeutung haben, wird stattdessen die geometrische Jacobimatrix der Kinematik \mathbf{J}_G verwendet (vgl. [HOR07], S. 17). Letztere bildet $\dot{\mathbf{q}}$ auf die Geschwindigkeit des Endeffektors ${}^0\mathbf{v}_e$ (Gleichung 5.10) ab:

$${}^0\mathbf{v}_e = \mathbf{J}_G(\mathbf{q}) \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_T(\mathbf{q}) \\ \mathbf{J}_R(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_{T1} & \mathbf{J}_{T2} & \cdots & \mathbf{J}_{Tn} \\ \mathbf{J}_{R1} & \mathbf{J}_{R2} & \cdots & \mathbf{J}_{Rn} \end{bmatrix} \dot{\mathbf{q}}. \quad (5.17)$$

\mathbf{J}_G besteht hierbei aus einem translatorischen und einem rotatorischen Anteil, \mathbf{J}_T und \mathbf{J}_R . Die Jacobimatrizen \mathbf{J}_{Ti} und \mathbf{J}_{Ri} werden auf Basis der Berechnungen aus der direkten Kinematik bezüglich eines Dreh- oder Schubgelenks bestimmt ([SSVO09], S. 112).

$$\begin{bmatrix} \mathbf{J}_{Ti} \\ \mathbf{J}_{Ri} \end{bmatrix} = \begin{cases} \begin{bmatrix} {}^0\mathbf{z}_{i-1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}, & \text{für ein Schubgelenk} \\ \begin{bmatrix} {}^0\mathbf{z}_{i-1} \times ({}^0\mathbf{o}_e - {}^0\mathbf{o}_{i-1}) \\ {}^0\mathbf{z}_{i-1} \end{bmatrix}, & \text{für ein Drehgelenk.} \end{cases} \quad (5.18)$$

Die Größen ${}^0\mathbf{o}_e$ und ${}^0\mathbf{o}_{i-1}$ definieren den Translationsvektor aus Gleichung 5.7, der aus der entsprechenden Gesamt-Transformation ${}^0\mathbf{T}_e$ respektive ${}^0\mathbf{T}_{i-1}$ extrahiert wird (sh. Gleichung 5.14). Die Größe ${}^0\mathbf{z}_{i-1}$ bezeichnet die dritte Spalte der entsprechenden Rotationsmatrix ${}^0\mathbf{R}_{i-1}$ als Bestandteil der Gesamt-Transformation ${}^0\mathbf{T}_{i-1}$.

5.8 Inverse Kinematik

Um eine Steuerung der Roboter-Plattform anhand von räumlichen Lageinformationen ermöglichen zu können, muss eine inverse Kinematik

$$\mathbf{q} = f({}^0\mathbf{p}_e) \quad (5.19)$$

bekannt sein. Dies entspricht der Umkehrung der direkten Kinematik aus Gleichung 5.14. Wenn eine geschlossene Lösung für f vorhanden ist, dann können die Gelenkstellungen \mathbf{q} für eine anzufahrende Endeffektor-Pose ${}^0\mathbf{p}_e$ sofort berechnet werden. Laut ([SSVO09], S. 91) ist es allerdings möglich, dass eine geschlossene Lösung nur schwer oder gar nicht gefunden werden kann. In diesem Fall muss auf numerische Lösungsverfahren zurückgegriffen werden. Letztere Vorgehensweise wird auch in der vorliegenden Arbeit verwendet.

5.8.1 Numerische Lösung

Zur numerischen Lösung der inversen Kinematik soll das *Gauß-Newton-Verfahren* verwendet werden. Hierfür wird zunächst eine Residuenfunktion definiert, indem die linke Seite der Gleichung 5.14 von der rechten Seite wie folgt abgezogen wird.

$$\mathbf{r} = f(\mathbf{q}) - {}^0\mathbf{T}_{e^*} = {}^0\mathbf{T}_e(\mathbf{q}) - {}^0\mathbf{T}_{e^*} \quad (5.20)$$

Dabei repräsentiert die Transformation ${}^0\mathbf{T}_{e^*}$ die vorgegebene Lage des Endeffektors zu der die Gelenkwinkel \mathbf{q} gesucht sind. Um den Iterationsschritt des numerischen Verfahrens beschreiben zu können, muss vorerst die Residuenfunktion aus Gleichung 5.20 in eine kürzere Form gebracht werden. Hierfür kann \mathbf{r} anhand eines rotatorischen und eines translatorischen Anteils definiert werden. Letzteres ist durch die entsprechenden translatorischen Bestandteile von ${}^0\mathbf{T}_{e^*}$ und ${}^0\mathbf{T}_e(\mathbf{q})$ gegeben (sh. Formel 5.7):

$${}^0\dot{\mathbf{o}}_e = {}^0\mathbf{o}_e(\mathbf{q}) - {}^0\mathbf{o}_{e^*}. \quad (5.21)$$

Der rotatorische Anteil von \mathbf{r} soll anhand der Winkelgeschwindigkeiten ${}^0\mathbf{w}_e$ bestimmt werden, welche mittels der Rotationsmatrizen als Bestandteile von ${}^0\mathbf{T}_{e^*}$ und ${}^0\mathbf{T}_e(\mathbf{q})$ berechnet werden können (sh. Formel 5.7):

$$\begin{aligned} S({}^0\mathbf{w}_e) &= ({}^0\mathbf{R}_e(\mathbf{q}) - {}^0\mathbf{R}_{e^*}) {}^e\mathbf{R}_0(\mathbf{q}) \\ S({}^0\mathbf{w}_e) &= {}^0\dot{\mathbf{R}}_e {}^e\mathbf{R}_0(\mathbf{q}). \end{aligned} \quad (5.22)$$

Die Bestimmung von ${}^0\mathbf{w}_e$ wird in Abschnitt A.2 im Detail erläutert. Anhand des translatorischen Anteils ${}^0\dot{\mathbf{o}}_e$ und des rotatorischen Anteils ${}^0\mathbf{w}_e$ kann die Residuenfunktion aus Gleichung 5.20 redefiniert werden.

$$\mathbf{r}(\mathbf{q}) = {}^0\mathbf{v}_e(\mathbf{q}) = \begin{bmatrix} {}^0\dot{\mathbf{o}}_e \\ {}^0\mathbf{w}_e \end{bmatrix} \quad (5.23)$$

Da \mathbf{r} die Geschwindigkeit des Endeffektors bezüglich $(KS)_0$ beschreibt, wird \mathbf{r} im weiteren Verlauf durch die Bezeichnung ${}^0\mathbf{v}_e$ substituiert. Die hier vorgestellte inverse Kinematik kann als Minimierungsproblem im Sinne der Methode der kleinsten Quadrate definiert werden:

$$\min_{\mathbf{q}} \| {}^0\mathbf{v}_e(\mathbf{q}) \|_2^2. \quad (5.24)$$

Zur Lösung dieses nicht-linearen Problems soll das *Gauß-Newton-Verfahren* verwendet werden. Hierbei wird die Minimierungsaufgabe sukzessive gelöst indem es im jeweiligen Arbeitspunkt linearisiert wird. Der Iterationsschritt ist dabei durch

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \dot{\mathbf{q}} \quad (5.25)$$

gegeben. Als Startwert \mathbf{q}_0 können typischerweise die aktuellen Gelenkstellungen des Manipulators verwendet werden. Eine Verbesserung von \mathbf{q}_n wird durch $\dot{\mathbf{q}}$ erreicht, welches durch die Invertierung der differentiellen Kinematik aus Formel 5.17 gegeben ist:

$$\dot{\mathbf{q}} = \mathbf{J}_G^+(\mathbf{q}_n) {}^0\mathbf{v}_e(\mathbf{q}_n). \quad (5.26)$$

Zur Berechnung von $\dot{\mathbf{q}}$ wird die Größe ${}^0\mathbf{v}_e(\mathbf{q}_n)$ benötigt, welche in Gleichung 5.23 festgelegt wurde. Die Pseudoinverse \mathbf{J}_G^+ ermöglicht eine fehlerminimierte Lösung über die Methode der kleinsten Quadrate, um mit Singularitäten umgehen zu können. Singularitäten werden im nachfolgenden Unterkapitel näher untersucht. Die Herleitung der Pseudoinversen einer Matrix wird unter Gliederungspunkt A.3 durchgeführt.

Das Iterationsverfahren terminiert sobald $\dot{\mathbf{q}}$ gegen $\mathbf{0}$ konvergiert. Ein Abbruch erfolgt hingegen, wenn eine maximale Anzahl von Iterationsschritten überstiegen wird. Im Fall eines Abbruchs liegt für \mathbf{q} keine Lösung vor.

5.8.2 Kinematische Singularitäten

Um $\dot{\mathbf{q}}$ aus Formel 5.25 berechnen zu können, muss das lineare Gleichungssystem

$${}^0\mathbf{v}_e = \mathbf{J}_G(\mathbf{q})\dot{\mathbf{q}} \quad (5.27)$$

gelöst werden. Wenn für $\dot{\mathbf{q}}$ dabei keine eindeutige Lösung vorhanden ist, dann liegt eine kinematische Singularität vor. Diese Singularitäten können in die folgenden Klassen unterteilt werden (vgl. [SSVO09], S. 116).

- **Äußere Singularitäten** liegen am Rand des Arbeitsraums eines Manipulators. In diesem Fall ist der Roboterarm vollständig ausgestreckt oder zusammengefallen.
- **Innere Singularitäten** treten innerhalb des Arbeitsraums auf. Hierbei liegen zum Beispiel zwei oder mehr Gelenkachsen auf einer Linie.

Um beim Auftreten einer kinematischen Singularität trotzdem zu einer Lösung für $\dot{\mathbf{q}}$ zu gelangen, kann die Pseudoinverse \mathbf{J}_G^+ aus Gleichung 5.26 verwendet werden. Auf diese Weise wird für $\dot{\mathbf{q}}$ eine fehlerminimierte Lösung über die Methode der kleinsten Quadrate ermittelt. Im Fall eines redundanten Manipulators, können bei dieser Berechnung zusätzlich Optimierungskriterien berücksichtigt werden. Für einen redundanten Manipulator gilt dabei, dass die Dimension des Gelenkraums größer ist als die Dimension des Arbeitsraums (vgl. [HOR07], S. 19-20):

$$\dim(\mathbf{q}) > \dim({}^0\mathbf{p}_e). \quad (5.28)$$

Die Größe ${}^0\mathbf{p}_e$ beschreibt hierbei die Lage des Endeffektors relativ zu $(KS)_0$ (sh. Formel 5.15). Für Roboter mit redundanter Kinematik gilt, dass für $\dot{\mathbf{q}}$ unendlich viele Lösungen vorliegen. In diesem Fall kann mit Hilfe der Lagrange'schen Multiplikatortechnik ein Optimierungskriterium bei der Berechnung von $\dot{\mathbf{q}}$ hinzugezogen werden ([SSVO09], S. 124-127). Verschiedene Kriterien bei der Optimierung ermöglichen zum Beispiel die nachstehenden Effekte.

- **Vermeidung von singulären Konfigurationen:** Der Roboter hält sich bei der Bewegung von singulären Konfigurationen fern. In der Nachbarschaft einer Singularität führen selbst kleine Werte für ${}^0\mathbf{v}_e$ zu hohen Gelenkgeschwindigkeiten $\dot{\mathbf{q}}$. Dies kann hier vermieden werden.

- **Distanz zu einem Hindernis:** Während der Bewegung kann sich der Roboter von einem Hindernis fernhalten, um mit diesem nicht zu kollidieren.

5.9 Parameter des Katana-Roboters

Im Folgenden werden die Parameter des erläuterten Kinematik-Modells für den Katana-Roboter ermittelt. Hierfür wird eine kinematische Kette bezüglich der Gelenke entworfen, um die DH-Parameter aus Abschnitt 5.5 bestimmen zu können. Abschließend wird das kinematische Modell des Katana-Manipulators beschrieben.

5.9.1 Kinematische Kette

Eine kinematische Kette beschreibt die kinematischen Zusammenhänge zwischen den einzelnen Gelenken eines Manipulators. Hierfür wird die DH-Konvention aus Kapitel 5.5 verwendet. Um eine kinematische Kette für den Katana-Roboter entwerfen zu können, werden zunächst die Maße des Roboters benötigt.

In Abbildung 5.5 wird die Länge der einzelnen Armglieder des Greifarms dargestellt. Die Längenangaben werden in Tabelle 5.1 zusammengefasst und den Bezeichnungen l_1 bis l_{45} zugewiesen. Die Armglied-Länge l_1 beschreibt beispielsweise den Abstand des ersten und zweiten Gelenks. Die Armglied-Länge l_{45} beschreibt den Abstand zwischen dem vierten und fünften Gelenk zuzüglich des Abstands zwischen dem fünften Gelenk und dem TCP.

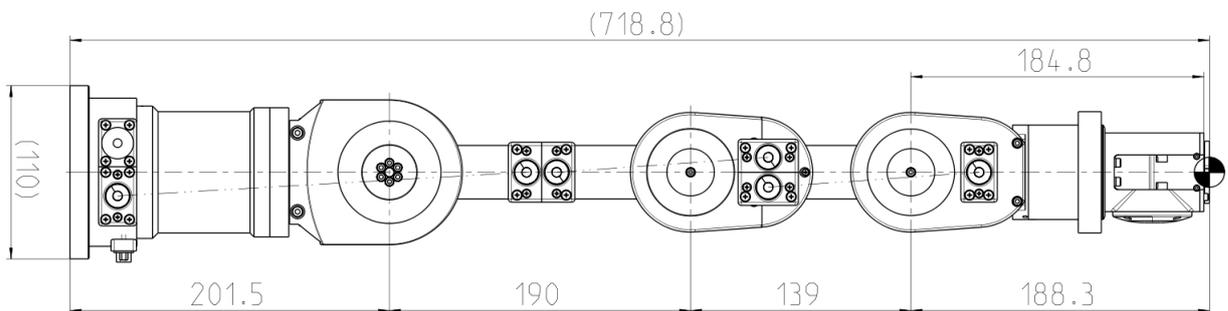


Abbildung 5.5: Abmessungen des Katana-Greifarms 6M180 (Bildquelle: [NEU08])

Armglied-Länge [mm]
$l_1 = 201,5$
$l_2 = 190$
$l_3 = 139$
$l_{45} = 188,3$

Tabelle 5.1: Länge der Armglieder

Grafik 5.6 zeigt die kinematische Kette des Katana-Greifarms in der kinematischen Grundstellung. Hierbei sind alle Gelenkwinkel q auf 0 gesetzt. Jedes Gelenk des Roboters wird entsprechend der DH-Konvention mit einem Koordinatensystem versehen. Das Koordinatensystem $(KS)_0$ stellt somit das Roboter-Koordinatensystem und $(KS)_5$ das Endeffektor-Koordinatensystem dar (vgl. Abschnitt 5.6). Das Koordinatensystem $(KS)_5$ wird daher im weiteren Verlauf mit $(KS)_e$ bezeichnet.

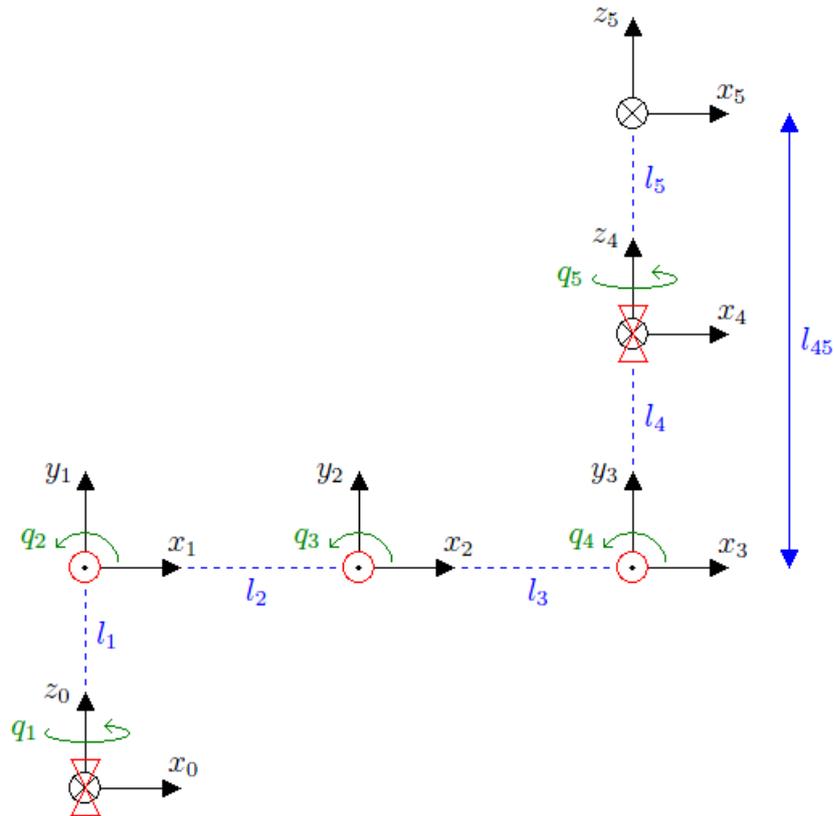


Abbildung 5.6: Kinematische Kette

Nachstehend werden die DH-Parameter des Katana-Greifarms in Tabelle 5.2 aufgeführt. Es ist anzumerken, dass das fünfte Gelenk zur Modellierung in das vierte Gelenk gelegt wird. Der Grund ist, dass nur die Länge l_{45} bekannt ist. Diese Modifikation hat dabei keine Auswirkungen auf das kinematische Modell.

i	α_i [rad]	a_i [mm]	θ_i [rad]	d_i [mm]
1	$\frac{\pi}{2}$	0	q_1	l_1
2	0	l_2	q_2	0
3	0	l_3	q_3	0
4	$-\frac{\pi}{2}$	0	q_4	0
5	0	0	q_5	l_{45}

Tabelle 5.2: DH-Parameter des Katana-Greifarms

5.9.2 Kinematik

Der Katana-Manipulator besitzt eine serielle Kinematik mit fünf Drehgelenken. Mit Hilfe der ermittelten DH-Parameter ergibt sich für die direkte Kinematik (sh. Formel 5.14):

$${}^0T_e(\mathbf{q}) = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) {}^3T_4(q_4) {}^4T_e(q_5). \quad (5.29)$$

Auf Basis dieser Transformationen sind alle Parameter bekannt die für das Modell der inversen Kinematik aus Abschnitt 5.8 benötigt werden.

6 Modellierung der Kamera

Auf Grund der Linse des Kameraobjektivs kommt es zu einer Verzerrung der Bildpunkte. Da im weiteren Verlauf dieser Arbeit entzerrte Bildpunkte benötigt werden, muss das Projektionsverhalten der Kamera modelliert werden. Das Kameramodell dient weiterhin als Basis für den bildpunkt-basierten Regler und die Rekonstruktion von Raumpunkten.

Zunächst erfolgt eine Erläuterung des zentralen Projektionsmodells, welches auch als Lochkameramodell bezeichnet wird. Dieses ermöglicht eine mathematische Beschreibung des Abbildungsverhaltens von Kameras mit fester Bildweite ohne Linsenverzeichnung. Da die Linse des Kameraobjektivs allerdings für eine Verzerrung sorgt, muss dies ebenfalls modelliert werden. Deshalb wird das zentrale Projektionsmodell um ein Modell für die Linsenverzeichnung erweitert.

6.1 Zentrales Projektionsmodell

Die projektive Abbildung eines Raumpunkts $\mathbf{p}^c = [x_c \ y_c \ z_c]^T$ bezüglich des Kamera-Koordinatensystems $(KS)_c$ auf einen Bildpunkt $\mathbf{p}^s = [x_s \ y_s]^T$ bezüglich des Sensor-Koordinatensystems $(KS)_s$ ist mittels des Strahlensatzes gegeben durch (sh. [HZ03], S. 153-154):

$$\mathbf{p}^s = \frac{f}{z_c} \begin{bmatrix} x_c \\ y_c \end{bmatrix}. \quad (6.1)$$

Dieser Zusammenhang wird durch Abbildung 6.1 verdeutlicht. Die Größe f steht dabei für die Bildweite und stellt einen intrinsischen Kameraparameter dar. Für die Modellierung der Projektion wird die Sensorebene $(KS)_s$ in Grafik 6.1 vor dem Kamerazentrum, der Ursprung des Kamera-Koordinatensystems $(KS)_c$, dargestellt. Im Gegensatz dazu befindet sich der Sensor in einer realen Kamera mit dem Abstand f hinter dem Kamerazentrum. Außerdem wird der Sensor im Vergleich zu Abbildung 6.1 mit π rad um die Achse x_s und anschließend mit π rad um die Achse y_s gedreht. Ein Projektionsstrahl, der die Punkte \mathbf{p}^c und \mathbf{p}^s miteinander verbindet, geht somit zunächst durch das Kamerazentrum und trifft dann auf die Sensorebene. Durch die Verwendung von homogenen Punkten $\tilde{\mathbf{p}}^c$ und $\tilde{\mathbf{p}}^s$ kann die Projektion in eine matrizielle Form gebracht werden:

$$\tilde{\mathbf{p}}^s = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{p}}^c. \quad (6.2)$$

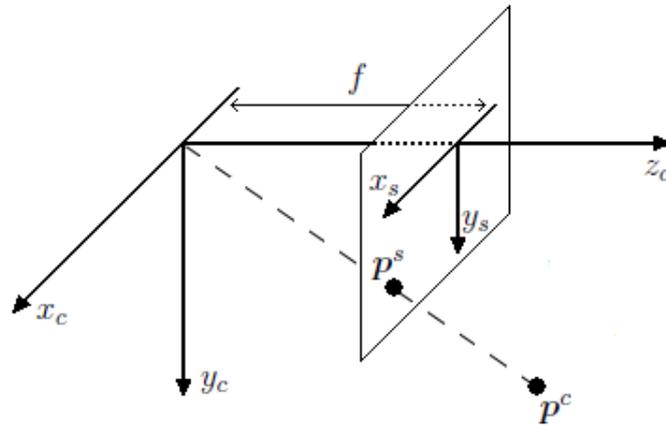


Abbildung 6.1: Zentralprojektion

Der Durchstoßpunkt der optischen Achse z_c mit der Sensorebene wird als Bildhauptpunkt bezeichnet und definiert den Ursprung von $(KS)_s$. Da der Ursprung eines Bildkoordinatensystems $(KS)_b$ im Verarbeitungssystem in der linken oberen Ecke liegt, muss p^s nach $p^b = [x_b \ y_b]^T$ bezüglich $(KS)_b$ transformiert werden (sh. Grafik 6.2). Hierbei handelt es sich um eine Translation, die durch die Verschiebung des Bildhauptpunkts $[x_0 \ y_0]^T$ gegeben ist:

$$\tilde{p}^b = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{p}^s. \quad (6.3)$$

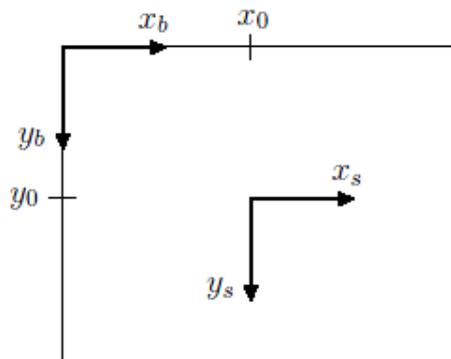


Abbildung 6.2: Sensor- und Bildkoordinatensystem

Formel 6.2 und 6.3 werden laut ([HZ03], S. 155) zusammengefasst durch:

$$\tilde{p}^b = \begin{bmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{p}^c = \underbrace{\begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} \end{bmatrix} \tilde{p}^c. \quad (6.4)$$

\mathbf{K} definiert dabei die Kalibriermatrix, welche alle intrinsischen Kameraparameter aufnimmt. Der Ausdruck $\mathbf{I}_{3 \times 3}$ steht für die (3×3) Einheitsmatrix. Bisher wurde ein Punkt p^c bezüglich des Kamera-Koordinatensystems $(KS)_c$ in die Bildebene projiziert. Da Objektpunkte allerdings häufig bezüglich eines Werkstück-Koordinatensystems $(KS)_w$ gegeben sind, muss ein Punkt p^w mittels der Transformation cT_w nach p^c konvertiert werden. ${}^cT_w({}^c p_w)$ soll auf Basis der Werkstück-Pose ${}^c p_w$ bezüglich der Kamera berechnet werden (vgl. Formel 5.7). Die Komponenten des Vektors ${}^c p_w$ werden hierbei als extrinsische Kameraparameter bezeichnet:

$${}^c\mathbf{p}_w = [x \ y \ z \ \phi \ \psi \ \theta]^T. \quad (6.5)$$

Die RPY-Winkel ϕ , ψ und θ werden, wie in Formel 5.6 gezeigt, aus dem Rotationsanteil von ${}^c\mathbf{T}_w$ ermittelt. Für die Gesamtprojektion von \mathbf{p}^w nach \mathbf{p}^b ergibt sich unter Verwendung von Gleichung 6.4 und ${}^c\mathbf{T}_w$:

$$\tilde{\mathbf{p}}^b = \underbrace{\mathbf{K} \begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{o}_w \end{bmatrix}}_{\mathbf{P}} \tilde{\mathbf{p}}^w. \quad (6.6)$$

Die Gesamtprojektion wird durch die Projektionsmatrix \mathbf{P} bestimmt. Für die Modellierung des Abbildungsverhaltens einer *Charge-coupled Device* (CCD)-Kamera müssen neben f , x_0 und y_0 weitere interne Kameraparameter berücksichtigt werden. Zunächst muss eine Skalierung m_x in x -Richtung und m_y in y -Richtung berücksichtigt werden. Beide Skalierungsfaktoren geben jeweils das Verhältnis zwischen der Anzahl der Pixel und der Seitenlänge des Sensors in der entsprechenden Achsen-Richtung an. Auf diese Weise ist es auch möglich nicht-quadratische Pixel zu modellieren. Im Fall von quadratischen Pixeln gilt $m_x = m_y$.

Weiterhin muss ein Scherungsfaktor s_k hinzugezogen werden, welcher den Winkel zwischen den Achsen x_s und y_s des Sensors repräsentiert. In der Regel sind diese Achsen orthogonal und daraus folgt $s_k = 0$. Sollte dies nicht der Fall sein, dann liegen keine rechtwinkligen Pixel vor. Die zusätzlichen intrinsischen Kameraparameter werden von der Kalibriermatrix \mathbf{K} wie folgt berücksichtigt (vgl. [HZ03], S. 156-157):

$$\mathbf{K} = \begin{bmatrix} fm_x & fm_x s_k & x_0 m_x \\ 0 & fm_y & y_0 m_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & \alpha_x s_k & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6.7)$$

Die internen Kameraparameter werden hierbei entsprechend durch die Größen α_x , α_y , u_0 , v_0 und s_k zusammengefasst.

6.2 Linsenverzeichnung

Im Folgenden wird das bisherige Kameramodell um eine Beschreibung für die Linsenverzeichnung erweitert. Hierbei werden die nachstehenden Verzeichnungsarten berücksichtigt (vgl. [WCH92], S. 4-5).

- **Radiale Verzeichnung:** Im Fall einer unvollkommenen Linsenform kommt es zu einer radialen Verzerrung δ_{rad} . Bildpunkte werden dabei bezüglich des Zentrums der Verzeichnung, dem Bildhauptpunkt aus Abbildung 6.1, entweder nach außen oder nach innen verschoben. Je größer der Abstand zwischen einem Bildpunkt und dem Verzeichnungszentrum ist, desto höher ist die radiale Verzerrung. Am Beispiel der unverzeichneten Linien des Gitters aus Grafik 6.3 a entsteht entweder eine kissen- oder eine tonnenförmige Verzeichnung von Linien in Grafik 6.3 b respektive c.
- **Dezentrierende Verzeichnung:** Die optischen Zentren von Linsen liegen im Idealfall kollinear auf der optischen Achse. Bei Abweichungen kommt es zur sogenannten dezentrierenden Verzeichnung δ_{dec} , welche aus einem radialen und einem tangentialen Anteil bestehen. Die tangentiale Verzerrung sorgt für eine seitliche Verschiebung, wie in Abbildung 6.4 veranschaulicht. Die Achsen stellen gleichzeitig unverzeichnete Linien dar, welche auf Grund der tangentialen Verzeichnung gekrümmt werden.

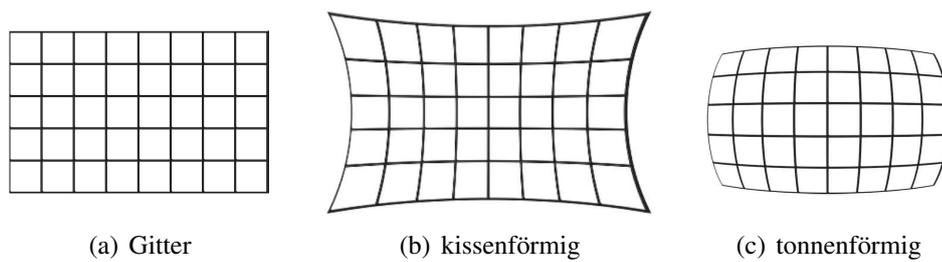


Abbildung 6.3: Arten der radialen Verzeichnung (Bildquelle: [GRA07])

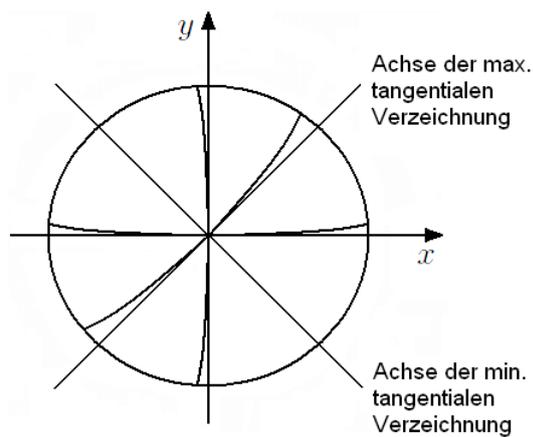


Abbildung 6.4: Tangentiale Verzeichnung von Linien

Die zu modellierende Linsenverzeichnung beinhaltet somit eine radiale Komponente δ_r und einen tangentialen Anteil δ_t . Diese beiden Verschiebungen werden in Abbildung 6.5 zusammengefasst dargestellt.

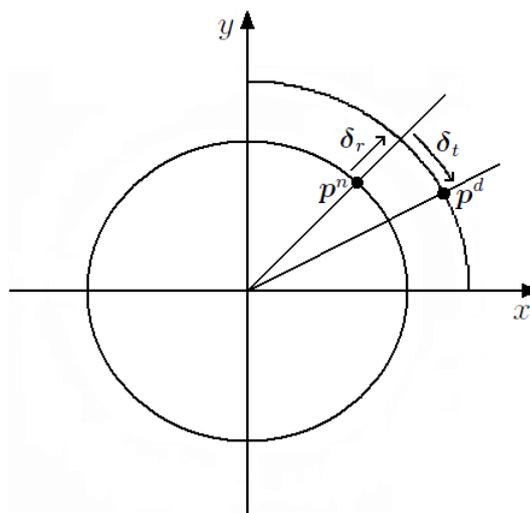


Abbildung 6.5: Radiale und tangentielle Verzeichnung eines Punktes

Als Ausgangssituation für das Verzeichnungsmodell dient hierbei ein normierter Bildpunkt $\mathbf{p}^n = [x_n \ y_n]^T$:

$$\mathbf{p}^n = \frac{1}{z_c} \begin{bmatrix} x_c \\ y_c \end{bmatrix}. \quad (6.8)$$

Dies entspricht der Gleichung 6.1, wobei f im vorliegenden Fall durch den Wert 1 ersetzt wurde. Weiterhin wird von einem normierten Bildpunkt \mathbf{p}^n der quadrierte Betrag

$$r^2 = x_n^2 + y_n^2 \quad (6.9)$$

benötigt, um die radiale Verzeichnung

$$\delta_{rad} = (k_{r1} * r^2 + k_{r2} * r^4 + k_{r3} * r^6) \mathbf{p}^n \quad (6.10)$$

und die dezentrierende Verzeichnung

$$\delta_{dec} = \begin{bmatrix} 2k_{t1}x_ny_n + k_{t2}(r^2 + 2x_n^2) \\ k_{t1}(r^2 + 2y_n^2) + 2k_{t2}x_ny_n \end{bmatrix} \quad (6.11)$$

berechnen zu können (vgl. [HS97B], S. 2). Die radialen Verzeichnungsparameter k_{r1} , k_{r2} , k_{r3} sowie die tangentialen Verzeichnungsparameter k_{t1} , k_{t2} stellen neben den Parametern der Kalibriermatrix weitere interne Kameraparameter dar. Das Gesamtmodell der Linsenverzeichnung ergibt sich somit aus

$$\mathbf{p}^d = \mathbf{p}^n + \delta_{rad} + \delta_{dec}. \quad (6.12)$$

Ein verzeichneter Bildpunkt \mathbf{p}^d wird anschließend mit Hilfe der Kalibriermatrix aus Gleichung 6.7 nach \mathbf{p}^b bezüglich des bildverarbeitenden Systems transformiert.

$$\tilde{\mathbf{p}}^b = \mathbf{K} \tilde{\mathbf{p}}^d. \quad (6.13)$$

6.3 Entzerrung von Bildpunkten

Im Verlauf dieser Arbeit werden entzerrte Bildpunkte benötigt und somit muss die Linsenverzeichnung kompensiert werden. Aus diesem Grund muss eine entsprechende Umkehrung des Kameramodells inklusive der Linsenverzeichnung gefunden werden. Das Ziel ist es, einen gegebenen Bildpunkt \mathbf{p}^b in einen normierten Bildpunkt \mathbf{p}^n (vgl. Formel 6.8) transformieren zu können. Um dies zu gewährleisten, wird zunächst Gleichung 6.13 in euklidischer Form dargestellt:

$$\mathbf{p}^b = \begin{bmatrix} \alpha_x(x_d + s_k y_d) + u_0 \\ \alpha_y y_d + v_0 \end{bmatrix}. \quad (6.14)$$

Zur Berechnung von \mathbf{p}^d kann dieser Zusammenhang wie folgt umgekehrt werden.

$$\begin{aligned} y_d &= \frac{y_b - v_0}{\alpha_y}, \\ x_d &= \frac{x_b - u_0}{\alpha_x} - s_k y_d \end{aligned} \quad (6.15)$$

Im nächsten Schritt muss die Linsenverzeichnung umgekehrt werden, um \mathbf{p}^d nach \mathbf{p}^n zu transformieren. Die Struktur der Gleichungen 6.9 bis 6.12 ermöglicht keine direkte Umkehrbarkeit der Linsenverzeichnung. Aus diesem Grund soll das Problem mit Hilfe einer nicht-linearen Optimierung

$$\min_{\mathbf{p}^n} \| \mathbf{p}^n + \delta_{rad} + \delta_{dec} - \mathbf{p}^d \|_2^2 \quad (6.16)$$

gelöst werden. Als Startwert wird \mathbf{p}^d verwendet, da sich dies in der Nähe der gesuchten Lösung für \mathbf{p}^n befindet. Zur Berechnung kann zum Beispiel der *Levenberg-Marquardt-Algorithmus* genutzt werden.

7 Kalibrierung

Dieses Kapitel umfasst die Kalibrierung der intrinsischen Kameraparameter für die hier verwendete Guppy-Kamera aus Gliederungspunkt 4.2. Somit wird eine Verwendung der Kameramodelle aus Abschnitt 6 ermöglicht. Weiterhin wird die Kalibrierung der extrinsischen Kameraparameter in Form der Pose eines Werkstücks bezüglich der Kamera erläutert. Dieses Vorgehen wird für das zu entwickelnde Positionierungsverfahren, bestehend aus der direkten Steuerung und der bildpunktbasierter Regelung, benötigt. Hierfür ist es ebenfalls notwendig, die Lage der Kamera relativ zum Endeffektor des Roboters zu berechnen. Dieses Vorgehen wird auch als Hand-Auge-Kalibrierung bezeichnet und wird im Folgenden näher beschrieben.

7.1 Kamerakalibrierung

Die Berechnung der internen Parameter der Guppy-Kamera wurde mit Hilfe der *Camera Calibration Toolbox* für Matlab durchgeführt (vgl. [BOU]). Dieses Werkzeug bietet ebenfalls die Funktionalität zur Ermittlung der externen Kameraparameter.

7.1.1 Intrinsische Parameter

Nachfolgend werden das Vorgehen und die Ergebnisse der Kalibrierung der internen Parameter der Guppy-Kamera zusammengefasst. Es wurde das Kalibrieremuster aus Grafik 7.1 verwendet, welches auf einer ebenen Oberfläche angebracht wurde. Im nächsten Schritt wurden 20 Bilder des Kalibrieremusters mit der Kamera aufgenommen (vgl. Abbildung 7.2).

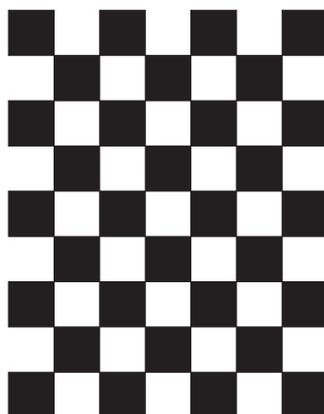


Abbildung 7.1: Kalibrieremuster

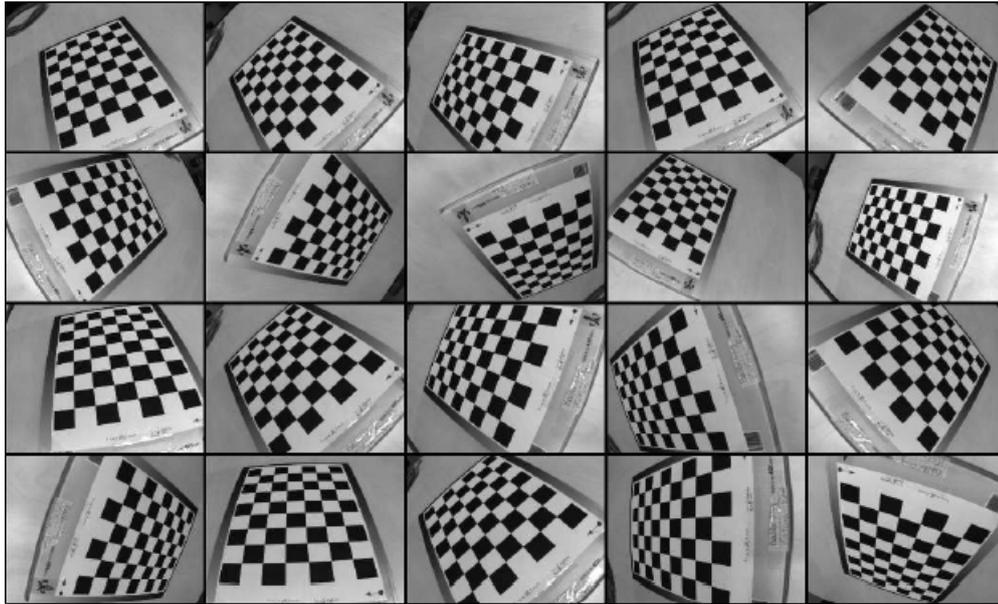


Abbildung 7.2: Bilder des Kalibrieramusters

Für jede der 20 Aufnahmen wurden anschließend die Ecken der inneren Quadrate des Kalibrieramusters in einem halbautomatischen Schritt bestimmt. Das Resultat wird exemplarisch für die erste Aufnahme in Grafik 7.3 gezeigt. Die roten Markierungen stellen dabei die detektierten Ecken dar, welche von grünen Linien eingerahmt werden. Diese Linien definieren dabei das orthogonale Koordinatensystem $(KS)_w$, dessen Ursprung durch das grüne O gekennzeichnet und durch die direkt benachbarte Ecke gegeben ist. Der Ursprung sowie die weiteren 3 grüne eingekreisten Extrempunkte wurden hierbei manuell in einer festdefinierten Reihenfolge bestimmt.

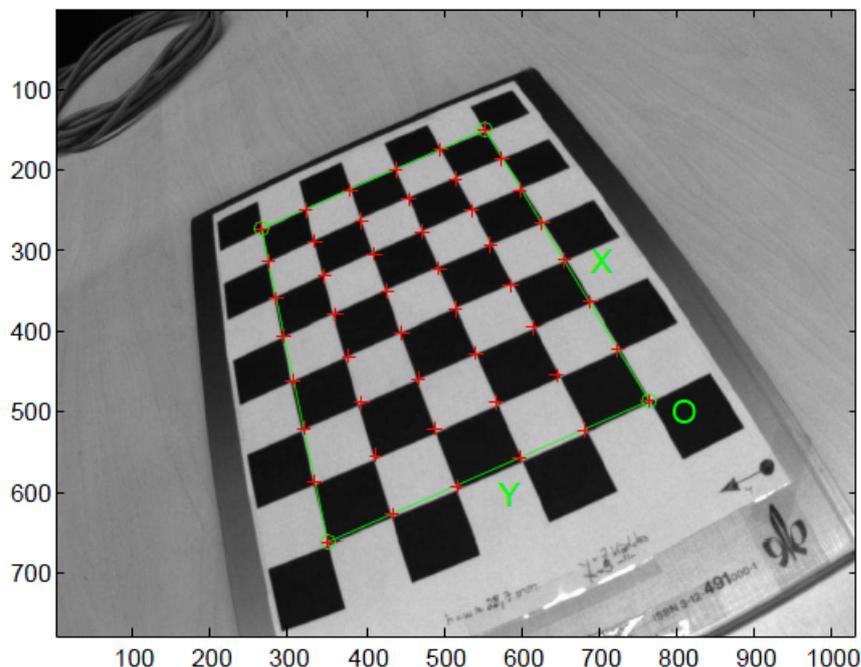


Abbildung 7.3: Eckpunkte des Kalibrieramusters

Auf diese Weise ist es möglich alle ermittelten Ecken in Form der Raumpunkte p_i^w (vgl. Abschnitt 6.1) bezüglich des Koordinatensystems $(KS)_w$ und in Form der Bildpunkte p_i^b gemäß

der Projektion einander zuzuordnen. Der Zusammenhang zwischen solchen korrespondierenden Punkten wird im weiteren Verlauf durch die Beschreibung $\mathbf{p}_i^b \leftrightarrow \mathbf{p}_i^w$ ausgedrückt. Die detektierten Ecken aus Grafik 7.3 stellen lediglich eine Näherung der tatsächlichen Position der einzelnen Ecken dar. Für eine möglichst genaue Kalibrierung werden die Ecken in einem weiteren Schritt subpixelgenau berechnet. Das Resultat wird in Grafik 7.4 dargestellt. Hier ist bereits zu erkennen, dass die Linse des Kameraobjektivs für eine tonnenförmige Verzeichnung der Bildpunkte sorgt.

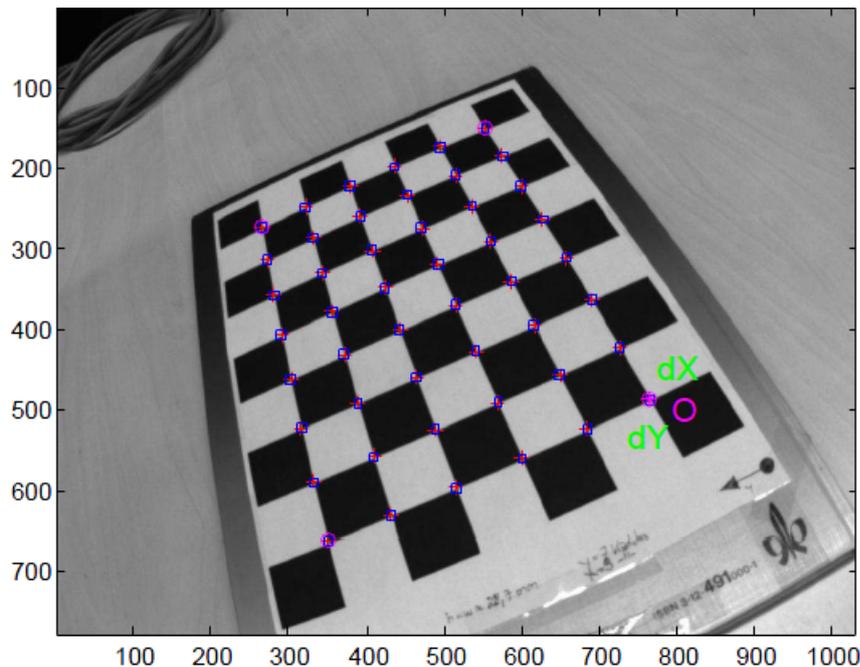


Abbildung 7.4: Subpixelgenaue Eckpunkte des Kalibrierungsmusters

Anhand der bildlich und räumlich übereinstimmenden Eckpunkte $\mathbf{p}_i^b \leftrightarrow \mathbf{p}_i^w$ sind alle Informationen gegeben, um die intrinsischen Kameraparameter inklusive der Linsenverzeichnungskoeffizienten aus Abschnitt 6.2 zu ermitteln. Das Kalibrierwerkzeug nutzt hierfür eine nicht-lineare Optimierung. Das Ergebnis der Berechnungen wird in der nachstehenden Tabelle zusammengefasst.

Parameter	Wert	Toleranz
α_x	932,96631	$\pm 1,33533$
α_y	933,98321	$\pm 1,36739$
u_0	527,13341	$\pm 1,92068$
v_0	406,31822	$\pm 2,04794$
s_k	0	± 0
$[k_{r1} \ k_{r2} \ k_{r3}]$	$[-0,38148 \ 0,17853 \ 0]$	$\pm [0,00548 \ 0,01865 \ 0]$
$[k_{t1} \ k_{t2}]$	$[0,00027 \ 0,00006]$	$\pm [0,00035 \ 0,00030]$

Tabelle 7.1: Kalibrierergebnisse für die Guppy-Kamera

Die Werte beziehen sich auf eine Bildauflösung von 1032 x 778 Pixel. Weiterhin wurden die Blenden des Kameraobjektivs arretiert, da aus einer Veränderung dieser Einstellung eine Abweichung der kalibrierten Werte folgen würde. Um ein Maß für die Genauigkeit der Kalibrierung angeben zu können, wird nachfolgend der Mittelwert \bar{m} und die Standardabweichung s der Reprojektionsfehler in x - und y -Richtung angegeben.

Parameter	Wert
\overline{m}_x	$2,2440 * 10^{-11}$
\overline{m}_y	$-1,7827 * 10^{-11}$
s_x	0,192
s_y	0,197

Tabelle 7.2: Mittelwert und Standardabweichung der Reprojektionsfehler

7.1.2 Extrinsische Parameter

Für die Kalibrierung der extrinsischen Kameraparameter wird zunächst eine Ausgleichsmethode vorgestellt, welche als Basis für eine robuste Schätzung unter Berücksichtigung von fehlerbehafteten Daten dient.

7.1.2.1 Ausgleichung

Die Berechnung der Lage eines Werkstücks bezüglich der Kamera soll ebenfalls mittels der *Camera Calibration Toolbox* für Matlab vorgenommen werden und wird hier lediglich kurz erläutert. Als Ausgangspunkt dient das Kameramodell mit Linsenverzeichnung aus Abschnitt 6.2. Auf dieser Basis kann das nicht-lineare Optimierungsproblem

$$\min_{{}^c\mathbf{R}_w, {}^c\mathbf{o}_w} \sum_{i=1}^n \| f(\mathbf{p}_i^w) - \mathbf{p}_i^b \|^2 \quad (7.1)$$

zur Berechnung der Werkstück-Lage (${}^c\mathbf{R}_w, {}^c\mathbf{o}_w$) formuliert werden. Hierbei müssen korrespondierende Bild- und Werkstückpunkte $\mathbf{p}_i^b \leftrightarrow \mathbf{p}_i^w$ berücksichtigt werden und es müssen alle intrinsischen Kameraparameter bekannt sein. In ([HZ03], S. 179) wird beschrieben, dass zur Schätzung von (${}^c\mathbf{R}_w, {}^c\mathbf{o}_w$) mindestens $n = 6$ übereinstimmende Punktepaare benötigt werden (vgl. [HZ03], S. 179).

Um an einen Startwert für die Optimierung zu gelangen, muss zwischen zwei Fällen unterschieden werden. Wenn sich alle Raumpunkte \mathbf{p}_i^w in einer koplanaren Anordnung befinden, dann kann eine erste Näherung der Werkstück-Lage durch eine Zerlegung der Homography-Matrix berechnet werden. Die Bedeutung der Homography-Matrix wird später unter Gliederungspunkt 10 im Kontext der Korrespondenzanalyse näher beschrieben. Für den Fall, dass die Raumpunkte \mathbf{p}_i^w nicht koplanar sind, kann ein Startwert mittels der sogenannten *Direct Linear Transformation* (DLT) ermittelt werden. Das Prinzip dieses Verfahrens wird ebenfalls in Abschnitt 10 vorgestellt.

7.1.2.2 Robuste Schätzung

Die Minimierung aus Gleichung 7.1 ermöglicht die Berechnung einer Ausgleichslösung für die Werkstück-Lage (${}^c\mathbf{R}_w, {}^c\mathbf{o}_w$). Hierbei wird das Modell bestmöglich bezüglich der übereinstimmenden Bild- und Werkstückpunkte $\mathbf{p}_i^b \leftrightarrow \mathbf{p}_i^w$ eingepasst. Wenn sich fehlerbehaftete Messungen in diesen Punktmengen befinden, dann wird folglich auch (${}^c\mathbf{R}_w, {}^c\mathbf{o}_w$) fehlerbehaftet sein. Dies hat wiederum negative Auswirkungen auf die Genauigkeit des konzipierten Positionierungsverfahrens zur Einstellung einer Greiflage.

Die Berechnung der zur Schätzung der extrinsischen Kameraparameter notwendigen Werkstückpunkte \mathbf{p}_i^w wird später in Kapitel 11 vorgenommen. Dabei wird sich zeigen, dass die

Punkte \mathbf{p}_i^w geringe Abweichungen enthalten werden. Um eine Fehlerfortpflanzung bezüglich $({}^c\mathbf{R}_w, {}^c\mathbf{o}_w)$ zu vermeiden, wird eine robuste Schätzung benötigt mit der die fehlerbehafteten Messungen bezüglich $\mathbf{p}_i^b \leftrightarrow \mathbf{p}_i^w$ kompensiert werden können.

Dies soll mit Hilfe des unter Abschnitt B.1 erläuterten *Random Sample Consensus* (RANSAC)-Algorithmus erreicht werden. Üblicherweise wird ein zu schätzendes Modell hierbei nur linearisiert berechnet, um Rechenzeit einzusparen. Erst nach der Terminierung des Algorithmus wird gegebenenfalls eine nicht-lineare Einpassung des Modells anhand der Konsensmenge durchgeführt.

Im Fall der hier zu berechnenden Werkstück-Lage $({}^c\mathbf{R}_w, {}^c\mathbf{o}_w)$ würde sich für eine initiale linearisierte Schätzung intuitiv das erläuterte Vorgehen zur Erlangung eines Startwerts für die Optimierung aus Abschnitt 7.1.2.1 anbieten. Hierbei wird ein Startwert je nach Anordnung der Punkte entweder anhand der Homography-Matrix oder mittels des DLT-Verfahrens ermittelt. Eine nicht-lineare Einpassung von $({}^c\mathbf{R}_w, {}^c\mathbf{o}_w)$ anhand der endgültigen Konsensmenge kann durch Formel 7.1 realisiert werden.

Im Rahmen dieser Arbeit hat sich allerdings herausgestellt, dass dieses Vorgehen keine zuverlässige robuste Schätzung ermöglicht. Vermutlich ist die linearisierte Schätzung zur Berechnung der Konsensmenge hierfür nicht exakt genug. Alternativ wurde die Berechnung der Konsensmenge in jedem Schritt anhand einer nicht-linearen Schätzung durchgeführt. Durch diese Vorgehensweise konnte die Zuverlässigkeit einer robusten Schätzung von $({}^c\mathbf{R}_w, {}^c\mathbf{o}_w)$ deutlich erhöht werden. Jedoch konnte die gesuchte Lage auch mit dieser Methode in manchen Konfigurationen nicht robust ermittelt werden.

Aus diesem Grund wurde in Anlehnung an den RANSAC-Algorithmus ein eigenes Vorgehen zur robusten Berechnung von $({}^c\mathbf{R}_w, {}^c\mathbf{o}_w)$ entworfen. Dabei wurde der RANSAC aus Abschnitt B.1 folgendermaßen adaptiert.

Ziel:

Robuste Schätzung der Werkstück-Lage $M = ({}^c\mathbf{R}_w, {}^c\mathbf{o}_w)$ bezüglich der Punktmenge $P = \{[(\mathbf{p}_i^w)^T (\mathbf{p}_i^b)^T]^T\}$, welche fehlerbehaftete Messungen enthält. Bei den Elementen von P handelt es sich um korrespondierende Bild- und Werkstückpunkte $\mathbf{p}_i^b \leftrightarrow \mathbf{p}_i^w$.

Vorbedingung:

Da zur Schätzung von M mindestens $n = 6$ übereinstimmende Punktepaare benötigt werden (vgl. Abschnitt 7.1.2.1), muss die Bedingung $|P| \geq n$ erfüllt sein.

Initialisierung:

Initialisiere $best_score = \infty$.

Algorithmus:

(1): Wähle zufällig n Punkte aus der Menge P aus und schätze das Modell M_i . Hierfür wird die Ausgleichsmethode aus Abschnitt 7.1.2.1 verwendet.

(2): Bestimme anhand aller Elemente aus P und des geschätzten Modells M_i den Fehler der Reprojektion $e_{rep,i}$ mittels Gleichung 7.2. Die Funktion f ist hierbei durch das Kameramodell mit Linsenverzeichnung aus Abschnitt 6.2 definiert.

(3): Wenn $e_{rep,i} < best_score$ gilt, dann setze $best_score = e_{rep,i}$ und $M = M_i$. Wiederhole den Vorgang in jedem Fall ab Schritt 1.

(4): Gilt nach N Wiederholungen $best_score \neq \infty$, dann terminiere. Die gesuchte Werkstück-Lage ist dann durch M gegeben. Unter der Bedingung $best_score = \infty$ terminiere mit einem Fehler.

Algorithmus 7.1: Robuste Schätzung der Werkstück-Lage $({}^c\mathbf{R}_w, {}^c\mathbf{o}_w)$

$$e_{rep} = \sum_{i=1}^n \| f(\mathbf{p}_i^w) - \mathbf{p}_i^b \|^2 \quad (7.2)$$

Der beschriebene Algorithmus 7.1 wählt die geschätzte Werkstück-Lage $({}^c\mathbf{R}_w, {}^c\mathbf{o}_w)$ aus, welche den geringsten Reprojektionsfehler e_{rep} hinsichtlich aller Elemente aus P erzeugt. Um das Modell mit dieser Methode tatsächlich näherungsweise korrekt schätzen zu können, müssen die fehlerbehafteten Punkte aus P demnach vorzugsweise normalverteilt bezüglich der tatsächlichen Werkstück-Lage $({}^c\mathbf{R}_w, {}^c\mathbf{o}_w)$ vorhanden sein.

7.2 Hand-Auge-Kalibrierung

Für die direkte Steuerung und die bildpunkt-basierte Regelung ist die Kenntnis über die Pose der Kamera relativ zum Endeffektor des Roboters notwendig. Da die Kamera am obersten Armglied des Katana-Roboters befestigt wurde (vgl. Grafik 1.1), befindet sich keines der Gelenke j_1 bis j_5 zwischen der Kamera und dem Endeffektor. Somit ist die Lage der Kamera bezüglich des Endeffektors konstant und muss lediglich einmalig mittels einer Hand-Auge-Kalibrierung

erfasst werden.

In Anlehnung an die Kamerakalibrierung wurde dazu die *Camera Calibration and Hand to Eye Calibration Toolbox* für Matlab verwendet (sh. [WEN]), welche auf der bereits erwähnten *Camera Calibration Toolbox* aufsetzt. In diesem Unterkapitel werden das Vorgehen und die Resultate der Hand-Auge-Kalibrierung erläutert.

In Abschnitt 7.1.1 wurde beschrieben, dass 20 Bilder des Kalibrierungsmusters aus Grafik 7.2 aufgenommen wurden. Für die Hand-Auge-Kalibrierung wurde in jedem der 20 Schritte bereits die Lage des Endeffektors bezüglich des Umwelt-Koordinatensystems $(KS)_0$ in Form der Gesamttransformation 0T_e aus Gleichung 5.14 gespeichert. Die gegebenen und gesuchten Größen werden dabei durch den Kalibrieraufbau aus Abbildung 7.5 veranschaulicht. Neben 0T_e ist auch die Lage cT_w des Kalibrierobjekts bezüglich der Kamera für jeden der 20 Schritte anhand der erläuterten Kamerakalibrierung gegeben. Im Zuge der Hand-Auge-Kalibrierung soll die Kamerapose eT_c relativ zum Endeffektor bestimmt werden.

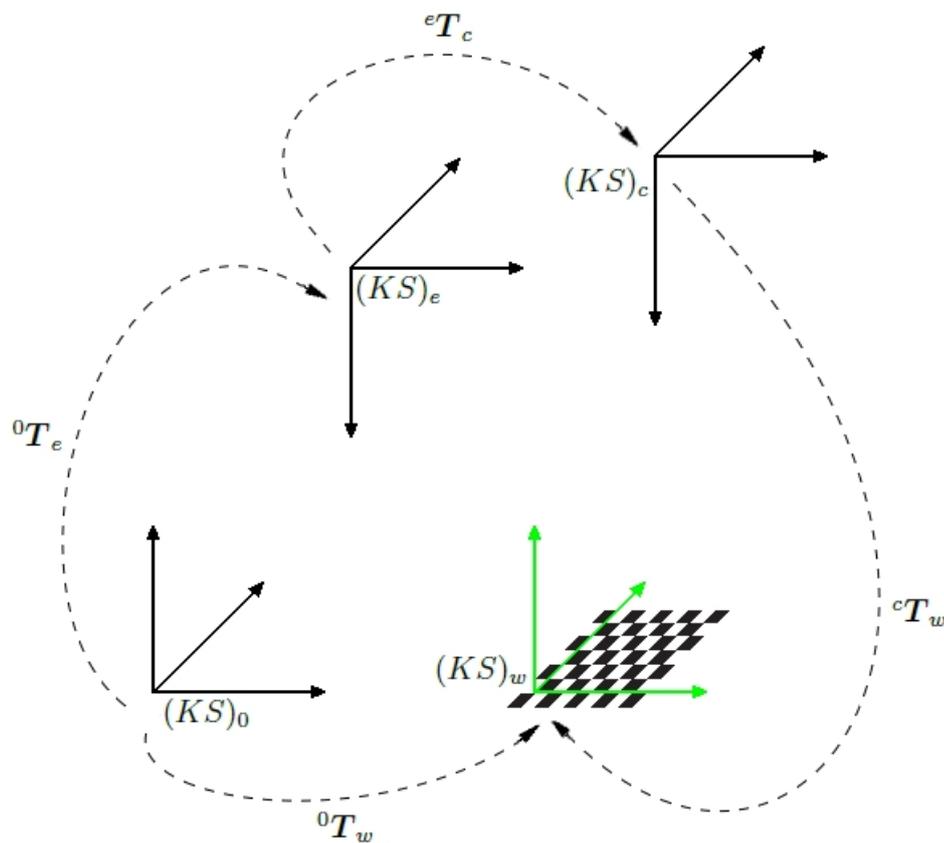


Abbildung 7.5: Aufbau zur Kalibrierung

Für die Berechnung der gesuchten Lage stehen in dem Kalibrierwerkzeug mehrere Verfahren zur Auswahl, von denen hier das in [HD95] beschriebene Vorgehen verwendet wurde. Als Grundlage für die Kalibrierung wird dabei das Modell

$${}^{c_1}T_{c_2} {}^cT_e = {}^cT_e {}^{e_1}T_{e_2} \quad (7.3)$$

angegeben, welches durch Grafik 7.6 veranschaulicht wird. Hierbei werden zwei unterschiedliche Stationen des Roboters genutzt, um die Größen ${}^{c_1}T_{c_2}$ und ${}^{e_1}T_{e_2}$ anhand der gegebenen Lageinformationen 0T_e und cT_w wie folgt berechnen zu können.

$$\begin{aligned} {}^{c_1}\mathbf{T}_{c_2} &= {}^{c_1}\mathbf{T}_w ({}^{c_2}\mathbf{T}_w)^{-1}, \\ {}^{e_1}\mathbf{T}_{e_2} &= ({}^0\mathbf{T}_{e_1})^{-1} {}^0\mathbf{T}_{e_2}. \end{aligned} \quad (7.4)$$

Die Berechnung von ${}^c\mathbf{T}_e$ wird laut ([HD95], S. 2) in einer geschlossenen Form durchgeführt. Zunächst wird der Rotationsanteil anhand von Einheits-Quaternionen berechnet, welche eine weitere Repräsentation der Orientierung als Bestandteil einer Pose darstellen. Anschließend kann der Translationsanteil für ${}^c\mathbf{T}_e$ ermittelt werden. Es ist zu beachten, dass die Lage zwischen dem Endeffektor und der Kamera in Gleichung 7.3 durch die Größe ${}^c\mathbf{T}_e$ definiert wird. Anhand der Invertierung von ${}^c\mathbf{T}_e$ kann die Lage ebenfalls in der zu Beginn angegebenen Form ${}^e\mathbf{T}_c$ dargestellt werden (vgl. Formel 5.8).

Abschließend werden die Ergebnisse der Hand-Auge-Kalibrierung für die hier verwendete Roboter-Kamera-Konfiguration in der nachstehenden Tabelle zusammengefasst. Die Lage der Kamera bezüglich des Endeffektors wird dabei hinsichtlich des Rotationsanteils in der kompakten RPY-Notation aus Gleichung 5.4 dargestellt. Die translatorischen Anteile werden in Millimeter und die Winkel in Radiant angegeben.

Parameter	Wert
${}^e\mathbf{o}_c$	$[-1,7853 \quad 50,7888 \quad -15,9702]^T$
${}^e\boldsymbol{\eta}_c$	$[-0,0083 \quad -0,0052 \quad -3,0955]^T$

Tabelle 7.3: Ergebnisse für die Hand-Auge-Kalibrierung

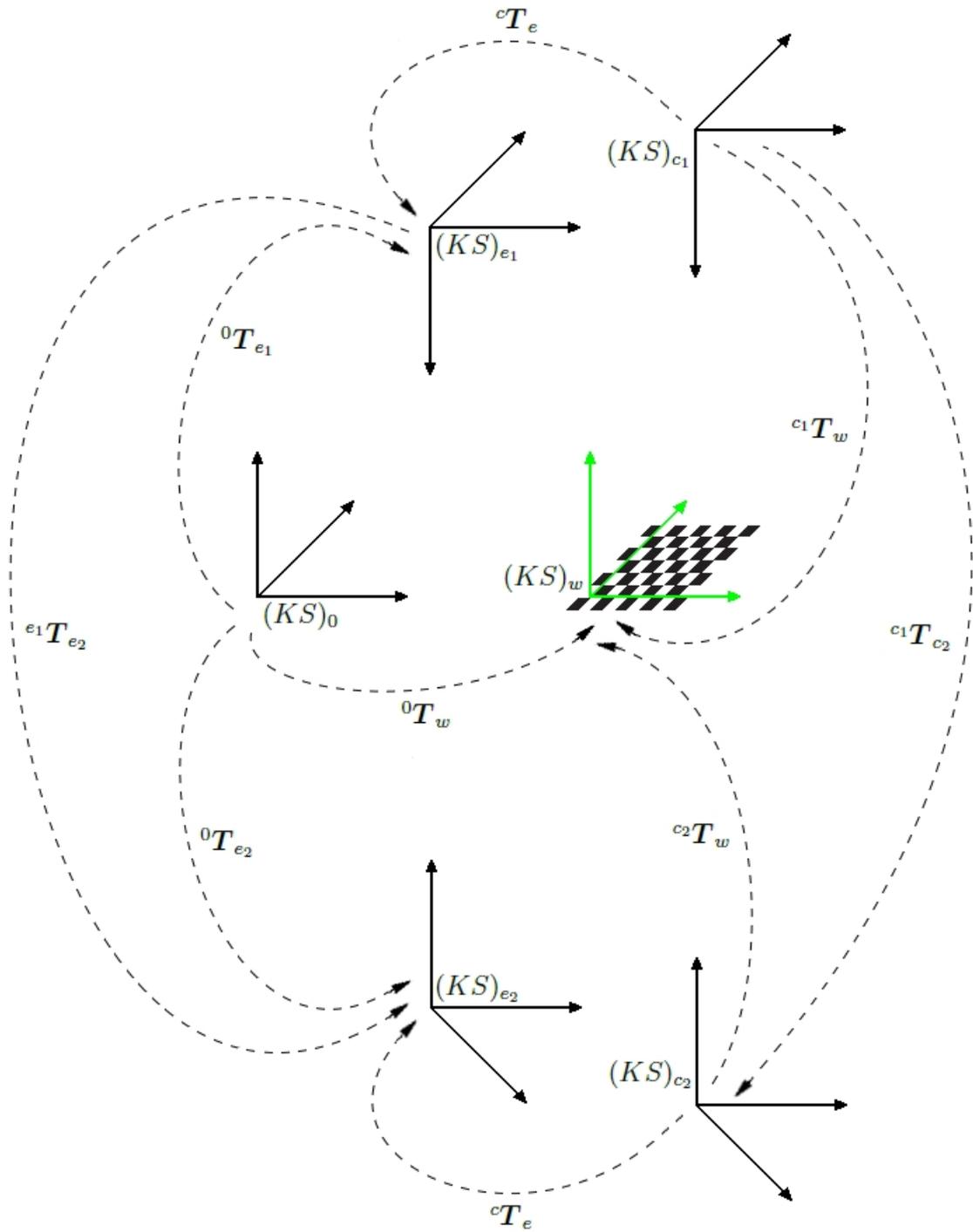


Abbildung 7.6: Modell für die Hand-Auge-Kalibrierung

8 Direkte Steuerung

Die Positionierung der Roboterhand in eine Greiflage bezüglich eines Gegenstands soll anhand des konzipierten Hybrid-Ansatzes aus Kapitel 3 vorgenommen werden. Im ersten Schritt der Positionierung wird dabei eine direkte Steuerung zur groben Einstellung der Greiflage durchgeführt (vgl. Abbildung 3.2).

8.1 Beschreibung

Die direkte Steuerung setzt voraus, dass ein zu greifendes Objekt im aktuellen Kamerabild dargestellt wird. Mit Hilfe einer Teilrekonstruktion des Objekts kann anschließend die aktuelle Werkstück-Lage cT_w relativ zur Kamera ermittelt werden. Dafür soll der vorgestellte Lokalisierungs-Algorithmus 7.1 eingesetzt werden. Neben einer Teilrekonstruktion für einen Gegenstand wird für dieses Vorgehen die Detektion von korrespondierenden Bild- und Werkstückpunkten $p_i^b \leftrightarrow p_i^w$ benötigt, wobei letztere bezüglich der Teilrekonstruktion gegeben sind. Diese Aspekte werden in den Kapiteln 10 bis 12 weiterverfolgt.

Sobald die aktuelle Werkstück-Lage cT_w bezüglich der Kamera bekannt ist, kann die Bewegung des Roboters in Richtung der vorgegebenen Greiflage in einem einzigen Schritt ausgeführt werden. Die vorgegebene Greiflage wird hierbei durch die Werkstück-Lage ${}^{c^*}T_w$ bezüglich der Kamera im Ziel repräsentiert. ${}^{c^*}T_w$ wird vorab anhand der Teilrekonstruktion des Gegenstands und der unter Gliederungspunkt 3 erläuterten Zielansicht B^* bestimmt. Hierbei sind die gleichen Berechnungen und Voraussetzungen wie im Fall der Bestimmung von cT_w zu berücksichtigen.

Um den Endeffektor des Roboters in Richtung der spezifizierten Greiflage ${}^{c^*}T_w$ bewegen zu können, muss letztere in eine Endeffektor-Lage ${}^0T_{e^*}$ relativ zum Umwelt-Koordinatensystem $(KS)_0$ transformiert werden. Diese Berechnung soll mittels der nachstehenden Steuereinheit umgesetzt werden.

8.2 Steuereinheit

Die erläuterten Lageinformationen cT_w und ${}^{c^*}T_w$, sowie die aktuelle Endeffektor-Pose 0T_e des kinematischen Modells aus Abschnitt 5.6 und die vorab kalibrierte Kamerapose eT_c relativ zum Endeffektor aus Kapitel 7.2 werden genutzt, um ${}^0T_{e^*}$ folgendermaßen zu bestimmen (sh. Grafik 8.1):

$$\begin{aligned} {}^0T_{e^*} &= {}^0T_e {}^eT_c {}^cT_w ({}^{c^*}T_w)^{-1} ({}^{e^*}T_{c^*})^{-1} \\ &= {}^0T_e {}^eT_c {}^cT_w {}^wT_{c^*} {}^{c^*}T_{e^*}. \end{aligned} \tag{8.1}$$

${}^{e^*}T_{c^*}$ ist hierbei äquivalent zu eT_c . Die berechnete Lage ${}^0T_{e^*}$ wird mit Hilfe der inversen Kinematik aus Abschnitt 5.8.1 in Gelenkstellungen q konvertiert, um den Roboter zu positionieren.

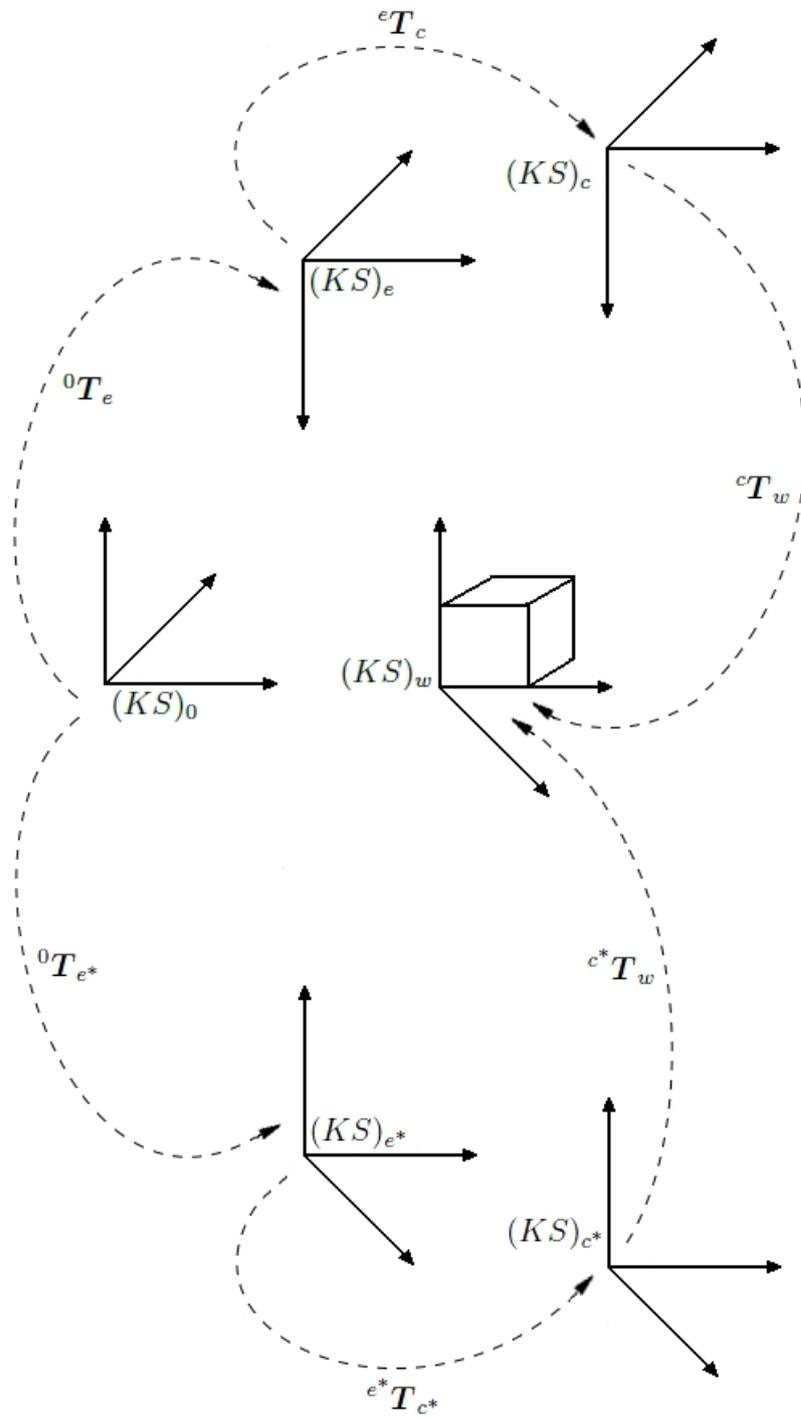


Abbildung 8.1: Verwendete Posen für die direkte Steuerung

9 Bildpunktbasierete Regelung

Aus den in Kapitel 2.1.1 erläuterten Gründen wird die durch die direkte Steuerung erreichte Greiflage von der vorgegebenen Greiflage abweichen. Um eine genaue Einstellung der Greiflage zu erreichen, wird deshalb im Anschluss eine bildpunktbasierete Regelung (vgl. Abbildung 2.6) durchgeführt. Durch die Überführung des aktuellen Bildes B in das Zielbild B^* , befindet sich der Endeffektor anschließend in der vorgegebenen Greiflage. Dieser Abschnitt enthält die Beschreibung des bildpunktbasiereten Reglers als Bestandteil des Regelkreises.

9.1 Bildjacobimatrix

Zur Beschreibung des Regelgesetzes muss zunächst die Bildjacobimatrix \mathbf{J}_i eingeführt werden, die in ([HOR07], S. 37-39) ausgehend von einer zeitlichen Ableitung der Gleichung 6.1 hergeleitet wird. \mathbf{J}_i bildet die Kamerageschwindigkeit ${}^c\mathbf{v}_c$ (vgl. Abschnitt 5.4) auf die Änderung $\dot{\mathbf{p}}^s = [\dot{x}_s \ \dot{y}_s]^T$ eines Bildpunktes ab:

$$\dot{\mathbf{p}}^s = \underbrace{\begin{bmatrix} -\frac{f}{z_c} & 0 & \frac{x_s}{z_c} & \frac{x_s y_s}{f} & -\frac{x_s^2 + f^2}{f} & y_s \\ 0 & -\frac{f}{z_c} & \frac{y_s}{z_c} & \frac{y_s^2 + f^2}{f} & -\frac{x_s y_s}{f} & -x_s \end{bmatrix}}_{\mathbf{J}_i} {}^c\mathbf{v}_c. \quad (9.1)$$

Die Bildjacobimatrix \mathbf{J}_i , welche auch als Interaktionsmatrix bezeichnet wird, enthält die Koordinaten eines aktuell betrachteten Bildpunktes $\mathbf{p}^s = [x_s \ y_s]^T$ und die Bildweite f . Außerdem beinhaltet \mathbf{J}_i die Tiefeninformation z_c des Objektpunktes $\mathbf{p}^c = [x_c \ y_c \ z_c]^T$ bezüglich der Kamera, welcher auf den aktuellen Bildpunkt \mathbf{p}^s projiziert wird. Bei einer Verwendung von n Bildpunkten statt einem Bildpunkt ergibt sich ein Stapel $\mathbf{J}_{i,ges}$ von Interaktionsmatrizen:

$$\dot{\mathbf{s}} = \begin{bmatrix} \dot{\mathbf{p}}_1^s \\ \vdots \\ \dot{\mathbf{p}}_n^s \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{J}_{i,1} \\ \vdots \\ \mathbf{J}_{i,n} \end{bmatrix}}_{\mathbf{J}_{i,ges}} {}^c\mathbf{v}_c. \quad (9.2)$$

$\mathbf{J}_{i,ges}$ wird im Folgenden der Einfachheit halber mit \mathbf{J}_i bezeichnet. Die Änderung der Bildpunkte wird durch $\dot{\mathbf{s}}$ zusammengefasst. Eine Invertierung der Gleichung 9.2 kann mit Hilfe der Pseudoinversen \mathbf{J}_i^+ der Bildjacobimatrix erreicht werden:

$${}^c\mathbf{v}_c = \mathbf{J}_i^+ \dot{\mathbf{s}}. \quad (9.3)$$

\mathbf{J}_i^+ ermöglicht hierbei eine fehlerminimierte Lösung über die Methode der kleinsten Quadrate, um mit Singularitäten umgehen zu können. Die Änderung der Bildpunkte $\dot{\mathbf{s}}$ stellt die Regelabweichung \mathbf{e} dar und ergibt sich durch die Subtraktion der aktuellen Bildpunkte $\mathbf{s} = [(\mathbf{p}_1^s)^T \ \dots \ (\mathbf{p}_n^s)^T]^T$ von den vorgegebenen Bildpunkten $\mathbf{s}^* = [(\mathbf{p}_1^{s*})^T \ \dots \ (\mathbf{p}_n^{s*})^T]^T$:

$$\dot{\mathbf{s}} = \mathbf{e} = \mathbf{s}^* - \mathbf{s}. \quad (9.4)$$

Die Soll-Größe des Regelkreises ist dabei durch \mathbf{s}^* definiert und wird aus der Referenzansicht B^* gewonnen. Die zu \mathbf{s}^* korrespondierenden Bildpunkte \mathbf{s} stellen die Ist-Größe dar und werden aus der aktuellen Ansicht B bezogen. Die Ermittlung von \mathbf{s} und \mathbf{s}^* wird in den Kapiteln 10 bis 12 weiterverfolgt. Dies gilt ebenso für die Tiefeninformation z_c der entsprechenden Objektpunkte \mathbf{p}_i^c , welche in der Interaktionsmatrix \mathbf{J}_i auftaucht. Unter Berücksichtigung einer

Regelverstärkung $\lambda > 0$ ergibt sich für die bildbasierte Regelung das proportionale Regelgesetz:

$${}^c v_c = \lambda J_i^+ e. \quad (9.5)$$

Im Fall der bildpunkt-basierten Regelung aus Abbildung 2.6 wird der Regelfehler e auf eine Geschwindigkeit v abgebildet, wobei letztere hier durch die Kamerageschwindigkeit ${}^c v_c$ konkretisiert (vgl. Grafik 9.1) wird. Mit Hilfe von ${}^c v_c$ kann eine Bewegung des Endeffektors in Richtung der Greiflage ermöglicht werden. Hierfür sind allerdings weitere Berechnungen notwendig, die während der Synthese der einzelnen Systemkomponenten in Kapitel 12 erläutert werden. Wenn die Regelabweichung gegen 0 konvergiert, dann stimmt B mit B^* überein. In diesem Fall befindet sich die Hand des Roboters in der vorgegebenen Greiflage relativ zum Gegenstand.

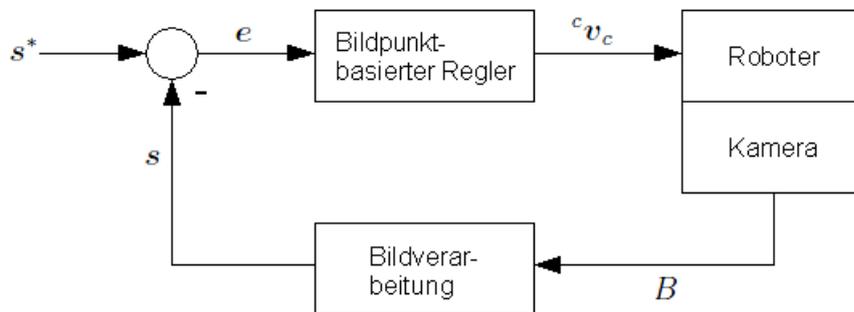


Abbildung 9.1: Bildpunkt-basierte Regelung

9.2 Stabilität der Regelung

Um die Kamerageschwindigkeit ${}^c v_c$ berechnen zu können, muss in jedem Regelschritt das lineare Gleichungssystem aus Formel 9.2 gelöst werden. Wenn drei Bildpunkte für die Regelung verwendet werden, dann ist das Gleichungssystem eindeutig bestimmt. Sind die zugehörigen Objektpunkte in diesem Fall beispielsweise kollinear, dann besitzt J_i nicht den vollen Rang (sh. [HOR07], S. 43). Somit liegt eine Singularität vor, die vermieden werden kann indem für die Regelung mehr als drei Bildpunkte verwendet werden. Diese Vorgehensweise wird auch in der vorliegenden Arbeit angewendet.

Die Nutzung von mehr als drei Bildpunkten hat weitere positive Effekte, die nachstehend kurz erläutert werden. Laut ([CH06], S. 2) gibt es bei der Verwendung von lediglich drei Bildpunkten vier unterschiedliche Kameralagen in denen der Regelfehler 0 ist. Für diese Konfiguration existieren also vier globale Minima von denen nur eins die richtige Lösung darstellt. Um diese Mehrdeutigkeiten zu vermeiden, sind auch hier mehr als drei Bildpunkte für die Regelung zu verwenden.

Ein weiterer Vorteil dieses Vorgehens betrifft die Berücksichtigung des Bildrauschen, das in Abschnitt 2.1.1 erläutert wurde. Dabei handelt es sich um eine gaußverteilte Abweichung der Pixelpositionen in einem Bild, welche negative Auswirkungen auf die Genauigkeit bei der Lösung von Gleichung 9.3 hat. Die Nutzung von mehr als drei beziehungsweise möglichst vielen Bildpunkten birgt dabei die Möglichkeit die Genauigkeit einer Ausgleichslösung zu erhöhen. Aus diesem Grund werden auch in dieser Arbeit so viele wie möglich aber mindestens

vier Bildpunkte ($n \geq 4$) zur Regelung verwendet.

Dadurch ergibt sich, dass das lineare Gleichungssystem aus Formel 9.2 in jedem Fall ($n \geq 4$) überbestimmt ist. Diese Eigenschaft hat Auswirkungen auf die Stabilität der Regelung, die im Folgenden beschrieben werden. Das erläuterte Gleichungssystem wird mit Hilfe der Pseudoinversen \mathbf{J}_i^+ aus Gleichung 9.3 gelöst. Da der Rang von \mathbf{J}_i bei einer Anzahl von ($n \geq 4$) Bildpunkten in jedem Fall sechs beträgt, gilt die folgende Bedingung (vgl. [HOR07], S. 44):

$$\dim(\ker(\mathbf{J}_i^+)) = 2n - \text{rang}(\mathbf{J}_i) = 2n - 6. \quad (9.6)$$

Im Fall von ($n \geq 4$) liegt also ein nichttrivialer Kern für \mathbf{J}_i^+ vor. Wenn somit für die Regelabweichung $e \in \ker(\mathbf{J}_i^+)$ gilt, dann liefert Gleichung 9.5 das Ergebnis ${}^c\mathbf{v}_c = 0$. Unter der zusätzlichen Bedingung, dass die Regelabweichung $e \neq 0$ ist, liegt somit ein lokales Minimum vor. In dieser Situation wird keine Bewegung mehr erzeugt, obwohl die Regelabweichung noch nicht abgeklungen ist. Dementsprechend befindet sich der Endeffektor nicht in der vorgegebenen Greiflage, weil das aktuelle Bild B nicht mit der Referenzansicht B^* übereinstimmt.

Da also die Möglichkeit besteht im Fall einer Anzahl von ($n \geq 4$) Bildpunkten in ein lokales Minimum zu geraten, kann mit der vorgestellten Methode laut ([CH06], S. 7-8) keine globale sondern lediglich eine lokale asymptotische Stabilität der Regelung gewährleistet werden. Da ein lokales Minimum anhand des Restfehlers $e \neq 0$ detektiert werden kann, könnte die Greiflage in dieser Situation trotzdem erreicht werden. Eine geeignete Vorgehensweise wäre zum Beispiel die Kamera so zu bewegen, dass die verfolgten Bildpunkte s anschließend noch im Kamerabild vorhanden sind.

Um diese Bedingung einzuhalten, könnte die Kamera lediglich um die optische Achse gedreht werden. Hierzu würde die Kamerageschwindigkeit ${}^c\mathbf{v}_c$ unter Vernachlässigung des Reglers mit entsprechenden Werten parametrisiert werden. Dies betrifft die Vektorkomponenten aus Gleichung 5.10. Um eine Drehung um die optische Achse zu ermöglichen würde lediglich die Komponente w_z eingestellt werden müssen:

$${}^c\mathbf{v}_c = [0 \ 0 \ 0 \ 0 \ 0 \ w_z]^T. \quad (9.7)$$

Nach der erläuterten Drehung der Kamera wären die verfolgten Bildpunkte s noch im Kamerabild und die Regelung könnte somit fortgesetzt werden. Da die beschriebene Maßnahme beim Erreichen eines lokalen Minimums bezüglich des verwendeten 5-FHG Katana-Roboters nur partiell umsetzbar wäre, soll dieser Ansatz in der vorliegenden Arbeit nicht weiter berücksichtigt werden. In ([HOR07], S. 44) wird erläutert, dass die Wahrscheinlichkeit während der Regelung in ein lokales Minimum zu geraten sehr gering ist. Somit ist eine Vernachlässigung von lokalen Minima zunächst möglich. Zukünftig sollten diese jedoch berücksichtigt werden.

Abschließend soll die lokale asymptotische Stabilität der bildpunkt-basierten Regelung näher untersucht werden. Laut ([HOR07], S. 42) ist die Regelung stabil, wenn der Realanteil der Eigenwerte der Matrix $\mathbf{J}_i^+ \mathbf{J}_i$ in jedem Regelschritt positiv ist:

$$\text{real}(\text{eig}(\mathbf{J}_i^+ \mathbf{J}_i)) > 0. \quad (9.8)$$

Unter der Bedingung das alle Parameter aus Gleichung 9.2 ideal und somit ohne Abweichungen vorhanden sind, existiert für die gesuchte Kamerageschwindigkeit ${}^c\mathbf{v}_c$ im Fall einer Anzahl von ($n \geq 4$) Bildpunkten genau eine Lösung. Es gilt dann $\mathbf{J}_i^+ \mathbf{J}_i = \mathbf{I}_{6 \times 6}$ und das Stabilitäts-

kriterium aus Formel 9.8 ist somit erfüllt. In diesem Fall können alle verfolgten Bildpunkte s auf einer geraden Bahn in die Ziel-Bildpunkte s^* überführt werden (vgl. Grafik 9.2 a).

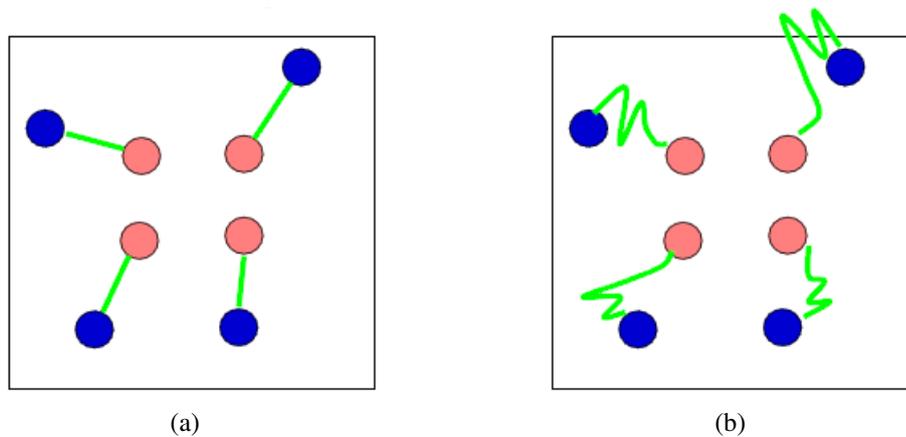


Abbildung 9.2: Bahn der verfolgten Bildpunkte s (blau) in Richtung der Ziel-Bildpunkte s^* (rosa). (Bildquelle: [CHA98])

Wenn die Größen aus Gleichung 9.2 hingegen nur approximativ vorliegen, dann kann vorab keine Aussage über die Stabilität der Regelung gemacht werden. Es ist allerdings davon auszugehen, dass das Stabilitätskriterium aus Gleichung 9.8 in der Regel nicht erfüllt ist. In diesem Fall kommt es zum Beispiel zu dem in Abbildung 9.2 b dargestellten Schwingungsverhalten bezüglich des Bildpunktverlaufs. Hierbei besteht die Gefahr, dass einige der verfolgten Bildpunkte s aus dem Blickfeld der Kamera geraten. Wenn die festgelegte Anzahl der benötigten Bildpunkte ($n \geq 4$) aus diesem Grund unterschritten wird, dann muss die Regelung abgebrochen werden. Dies ist jedoch nur bei hohen Parameterabweichungen zu erwarten.

Da die Parameter aus Gleichung 9.2 im realen System mit geringfügigen Unsicherheiten behaftet sein werden, ist davon auszugehen, dass es zu dem erläuterten Schwingungsverhalten und dem Verschwinden einiger der verfolgten Bildpunkte s aus dem Blickfeld der Kamera kommen kann. Demzufolge muss das Systemverhalten diesbezüglich bei der simulativen und experimentellen Verifikation des Gesamt-Systems in den Abschnitten 13 und 14 genauer untersucht werden.

10 Korrespondenzanalyse

Unter Abschnitt 3.2 wurde erläutert, dass eine kamerabasierte Wiedererkennung von Gegenständen zur Realisierung des Greifkonzepts ermöglicht werden muss. Eine Objektidentifikation ist dabei initial während der Suche nach einem Gegenstand erforderlich. Dies gilt ebenfalls in jedem Schritt der bildpunktbasierten Regelung.

Eine Wiedererkennung soll hier anhand eines Vergleichs zwischen der aktuellen Ansicht B und der Referenzansicht B^* umgesetzt werden. Dafür ist die erläuterte Korrespondenzanalyse aus Kapitel 2.2 prädestiniert, welche einen Ansatz zur Detektion von übereinstimmenden Punkten in perspektivisch unterschiedlichen Bildern zur Verfügung stellt. Um eine zuverlässige Erfassung von korrespondierenden Bildpunkten zu erreichen, soll dieser Ansatz hier weiterentwickelt werden.

Mit Hilfe der Korrespondenzanalyse können gleichzeitig die Soll- und Ist-Größe s^* respektive s für die bildpunktbasierte Regelung berechnet werden. Unter Gliederungspunkt 2.2 wurden verschiedene Verfahren zur Berechnung von markanten Bildpunkten erläutert, welche als Grundlage für eine Detektion von übereinstimmenden Bildpunkten dienen. Dabei hat sich gezeigt, dass eine steigende Robustheit hinsichtlich der Merkmalsvektoren auch eine erhöhte Rechenzeit mit sich bringt. Die Robustheit wird dabei anhand der Anzahl der korrekten Bildpunkt-Korrespondenzen gemessen.

Eine höhere Robustheit ermöglicht es somit, dass ein Gegenstand zuverlässiger wiedererkannt werden kann. Dies betrifft also den Grad der Invarianz gegenüber den unter Abschnitt 2.2.1 erläuterten Effekten, wie zum Beispiel die Änderung der Perspektive oder der Helligkeit. Da in dieser Arbeit allgemein eine möglichst robuste Lösung angestrebt wird, soll das SIFT-Verfahren zur Bestimmung von markanten Bildpunkten zum Einsatz kommen.

Ein durch SIFT ermittelter markanter Bildpunkt, kurz SIFT-Punkt, wird durch seine subpixelgenauen Koordinaten \mathbf{p}^b im Bild und einen 128-dimensionalen Merkmalsvektor \mathbf{d} beschrieben. SIFT-Punkte welche bezüglich der Referenzansicht B^* gegeben sind, werden durch die Menge

$$D^* = \left\{ \left[\begin{array}{c} \mathbf{p}_1^{b^*} \\ \mathbf{d}_1^* \end{array} \right], \dots, \left[\begin{array}{c} \mathbf{p}_n^{b^*} \\ \mathbf{d}_n^* \end{array} \right] \right\} \quad (10.1)$$

zusammengefasst. Entsprechend sind SIFT-Punkte aus der aktuellen Ansicht B durch die Menge

$$D = \left\{ \left[\begin{array}{c} \mathbf{p}_1^b \\ \mathbf{d}_1 \end{array} \right], \dots, \left[\begin{array}{c} \mathbf{p}_n^b \\ \mathbf{d}_n \end{array} \right] \right\} \quad (10.2)$$

definiert. Mit Hilfe der Korrespondenzanalyse gilt es nun die übereinstimmenden Bildpunkte $\mathbf{p}_i^{b^*} \leftrightarrow \mathbf{p}_i^b$ bezüglich der Mengen D und D^* zu ermitteln. Hierfür soll im ersten Schritt der unter Abschnitt 2.2.1 erwähnte *Nearest-Neighbor*-Algorithmus genutzt werden. Um jedoch eine zuverlässige Erfassung von korrespondierenden Bildpunkten zu erreichen, müssen anschließend weitere Berechnungen durchgeführt werden.

10.1 Nearest-Neighbor-Algorithmus

Der *Nearest-Neighbor*-Algorithmus ermöglicht die Suche nach dem dichtesten Nachbarn eines Deskriptors d^* hinsichtlich aller Deskriptoren d_i aus der Menge D unter Berücksichtigung eines Distanzmaßes. Für Letzteres wird hier die euklidische Norm zwischen zwei SIFT-Deskriptoren d^* und d ermittelt:

$$d_{dist} = \| d^* - d \|_2 . \quad (10.3)$$

Der dichteste Nachbar kann somit durch die geringste Distanz d_{dist} identifiziert werden. Die zu d^* und seinem dichtesten Nachbarn d assoziierten Bildpunkte p^{b^*} respektive p^b werden als korrespondierende Bildpunkte $p^{b^*} \leftrightarrow p^b$ gekennzeichnet.

Das beschriebene Vorgehen wird für alle Merkmalsvektoren d_i^* durchgeführt. Anschließend stehen die übereinstimmenden Bildpunkte $p_i^{b^*} \leftrightarrow p_i^b$ zur Verfügung. Hierbei handelt es sich jedoch lediglich um vermeintlich übereinstimmende Bildpunkte. Es kann also vorkommen, dass sich in der Menge der übereinstimmenden Punkte auch Ausreißer, also nicht korrespondierende Bildpunkte, befinden.

Der Grund dafür liegt zum einen an der Einfachheit des *Nearest-Neighbor*-Algorithmus und an der eingeschränkten Invarianz der SIFT-Merkmale, deren Eigenschaften unter Gliederungspunkt 2.2.1 beschrieben wurden. Mit Hilfe des hier erläuterten Vorgehens werden zum Beispiel auch dann korrespondierende Bildpunkte gefunden, wenn die Bilder B und B^* vollkommen unterschiedliche Szenen zeigen.

Zur Berechnung von SIFT-Punkten wird in dieser Arbeit die *VLFeat*-Bibliothek für Matlab genutzt (vgl. [VF]), welche auch eine Implementierung des *Nearest-Neighbor*-Algorithmus zur Verfügung stellt. Allerdings enthält Letztere eine Verbesserung gegenüber der hier erläuterten Vorgehensweise, welche in [LOW04] beschrieben wird. Dadurch wird erreicht, dass die ermittelten Übereinstimmungen $p_i^{b^*} \leftrightarrow p_i^b$ weniger Ausreißer enthalten als im Vergleich zur hier vorgestellten Methode. Um die restlichen falschen Übereinstimmungen aus $p_i^{b^*} \leftrightarrow p_i^b$ herauszufiltern, müssen weitere Berechnungen durchgeführt werden, welche im Folgenden dargelegt werden.

10.2 Homography-Matrix

Zur Bereinigung der bisher ermittelten vermeintlich übereinstimmenden Bildpunkte $p_i^{b^*} \leftrightarrow p_i^b$ von Ausreißern, soll die Homography-Matrix H verwendet werden. Auf der Basis einer Ausgleichungsmethode zur Berechnung von H wird dessen robuste Schätzung unter Berücksichtigung von Ausreißern erläutert. Letzteres ermöglicht auch das Extrahieren der Ausreißer aus $p_i^{b^*} \leftrightarrow p_i^b$.

10.2.1 Beschreibung

Die bisher detektierten vermeintlich übereinstimmenden Punkte $p_i^{b^*} \leftrightarrow p_i^b$ sind bezüglich des Bildkoordinatensystems $(KS)_b$ definiert und sind deshalb mit der unter Abschnitt 6.2 erläuterten Linsenverzeichnung behaftet. Die Homography-Matrix H ist laut ([FIT01], S. 2) allerdings im Allgemeinen nicht definiert, wenn die zur Berechnung von H notwendigen korrespondie-

renden Bildpunkte $\mathbf{p}_i^{b*} \leftrightarrow \mathbf{p}_i^b$ einer Linsenverzeichnung unterliegen.

Aus diesem Grund werden die Übereinstimmungen $\mathbf{p}_i^{b*} \leftrightarrow \mathbf{p}_i^b$ zunächst anhand der unter Gliederungspunkt 6.3 beschriebenen Vorgehensweise entzerrt. Die korrespondierenden Bildpunkte sind somit durch $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ normiert bezüglich des Sensor-Koordinatensystems gegeben (vgl. Formel 6.8).

Infolge des SIFT-Algorithmus kann es vorkommen, dass SIFT-Punkte mehrfach vorkommen. In diesem Fall bestehen für einen Bildpunkt mehrere unterschiedliche SIFT-Merkmalvektoren. Aus diesem Grund können einzelne Punkte in den Übereinstimmungen $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ doppelt vorkommen. Da sich dieser Sachverhalt negativ auf die folgenden Berechnungen auswirken kann, werden vorab alle doppelten Punkte aus \mathbf{p}_i^{n*} und \mathbf{p}_i^n entfernt. Dabei muss darauf geachtet werden, dass beim Entfernen eines Punktes auch sein vermeintlich übereinstimmender Punkt beseitigt wird.

Die (3×3) Homography-Matrix \mathbf{H} bildet einen Bildpunkt $\tilde{\mathbf{p}}^n$ auf einen korrespondierenden Bildpunkt $\tilde{\mathbf{p}}^{n*}$ wie folgt ab (vgl. [HZ03], S. 32-33):

$$\tilde{\mathbf{p}}^{n*} = \mathbf{H}\tilde{\mathbf{p}}^n = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tilde{\mathbf{p}}^n. \quad (10.4)$$

Es sei angemerkt, dass \mathbf{H} auch für jegliche Bildpunkte hinsichtlich der unterschiedlichen Koordinatensysteme des zentralen Projektionsmodells aus Kapitel 6.1 definiert werden kann. Der Einfachheit halber wird die Abbildung hier jedoch lediglich für normierte Bildpunkte eingeführt. Da es sich bei \mathbf{H} um eine nicht-singuläre Matrix handelt, ist die Umkehrung der Abbildung durch eine Invertierung von \mathbf{H} gegeben:

$$\tilde{\mathbf{p}}^n = \mathbf{H}^{-1} \tilde{\mathbf{p}}^{n*}. \quad (10.5)$$

10.2.2 Ausgleichung

Nachfolgend wird die *Direct Linear Transformation* (DLT) zur Berechnung einer Ausgleichslösung von \mathbf{H} auf der Grundlage von gegebenen übereinstimmenden Bildpunkten $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ vorgestellt. Als Ausgangspunkt wird hierfür Gleichung 10.4 genutzt. Da es sich bei \mathbf{H} um eine homogene Matrix handelt, kann diese mit einem Skalierungsfaktor $s \neq 0$ multipliziert werden ohne dabei eine Auswirkung auf die Abbildung zu nehmen (vgl. [HZ03], S. 33):

$$\tilde{\mathbf{p}}^{n*} = s\mathbf{H}\tilde{\mathbf{p}}^n. \quad (10.6)$$

In Formel 10.4 wurde s somit auf den Wert 1 festgelegt. Aus Gleichung 10.6 folgt, dass die Vektoren $\tilde{\mathbf{p}}^{n*}$ und $\mathbf{H}\tilde{\mathbf{p}}^n$ nicht gleich sind. Stattdessen besitzen sie zwar dieselbe Richtung aber sie können sich hinsichtlich ihrer Länge unterscheiden. Diese Eigenschaft ermöglicht es die erläuterten Zusammenhänge mit Hilfe des Kreuzproduktes auszudrücken (sh. [HZ03], S. 88-89):

$$\mathbf{0}_{3 \times 1} = \tilde{\mathbf{p}}^{n*} \times \mathbf{H}\tilde{\mathbf{p}}^n. \quad (10.7)$$

Durch die Umformung von Formel 10.7 ergibt sich das folgende lineare Gleichungssystem:

$$\mathbf{0}_{2 \times 1} = \underbrace{\begin{bmatrix} \mathbf{0}_{1 \times 3} & -(\mathbf{p}^n)^T & -1 & y_n^* (\mathbf{p}^n)^T & y_n^* \\ (\mathbf{p}^n)^T & 1 & \mathbf{0}_{1 \times 3} & -x_n^* (\mathbf{p}^n)^T & -x_n^* \end{bmatrix}}_{\mathbf{A}} \underbrace{[h_1 \cdots h_9]^T}_{\mathbf{h}}. \quad (10.8)$$

Hierbei wurde bereits eine linear abhängige Gleichung entfernt. Weiterhin wurde die dritte Zeile von $\tilde{\mathbf{p}}^{n*}$ und $\tilde{\mathbf{p}}^n$ zur Vereinfachung auf den Wert 1 festgelegt, so dass nur noch die euklidischen Koordinaten x_n^* und y_n^* , sowie der euklidische Punkt \mathbf{p}^n übrig bleiben. Bei einer Verwendung von n Punktepaaren ergibt sich

$$\mathbf{0} = \underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{bmatrix}}_{\mathbf{A}_{ges}} \mathbf{h}. \quad (10.9)$$

\mathbf{A}_{ges} wird im Folgenden der Einfachheit halber mit \mathbf{A} bezeichnet. Da es sich bei der gesuchten Größe \mathbf{H} um eine homogene (3×3) Matrix handelt, besitzt diese acht FHG. Aus diesem Grund muss zur Berechnung von \mathbf{h} die Bedingung $n \geq 4$ hinsichtlich der Anzahl der Punktepaare erfüllt sein.

Laut ([HZ03], S. 90) besitzt \mathbf{A} dann in jedem Fall den Rang $r = 8$, wenn die verwendeten n Punktepaare ohne Abweichungen vorliegen. Da die gesuchte Größe \mathbf{h} neun Einträge enthält, verfügt \mathbf{A} in diesem Fall über einen 1-dimensionalen Kern. Auf diese Weise kann für \mathbf{h} eine vom Nullvektor verschiedene Lösung bestimmt werden, welche allerdings nur bis auf eine Skalierung $s \neq 0$ gegeben ist. Da die entsprechende Homography-Matrix \mathbf{H} jedoch im Allgemeinen wie in Gleichung 10.6 mit einem Skalierungsfaktor $s \neq 0$ multipliziert werden kann, ist die Lösung für \mathbf{h} somit exakt.

Wenn das lineare Gleichungssystem aus Formel 10.9 überbestimmt ist ($n > 4$) und die genutzten Punktepaare fehlerbehaftet sind, dann liegt für \mathbf{h} nur noch die triviale Lösung $\mathbf{0}_{9 \times 1}$ vor. In diesem Fall erfolgt eine fehlerminimierte Schätzung für \mathbf{h} . Werden hingegen $n = 4$ fehlerbehaftete Punktepaare verwendet, dann liegt für \mathbf{h} eine eindeutige Lösung bis auf eine Skalierung $s \neq 0$ vor. Auf Grund der abweichenden Punkte ist diese Lösung allerdings nicht exakt.

In ([HZ03], S. 90) wird eine allgemeine Methode zur Bestimmung von \mathbf{h} beschrieben, welche im Fall von $n \geq 4$ korrespondierenden Bildpunkten angewendet werden kann. Dabei ist es unerheblich ob die Punktepaare exakt oder fehlerbehaftet vorliegen. Um dies zu erreichen wird ein Minimierungsproblem mit einer Nebenbedingung auf der Grundlage von Formel 10.9 formuliert:

$$\begin{aligned} \min_{\mathbf{h}} \|\mathbf{A}\mathbf{h}\|_2^2, \\ \text{so dass } \|\mathbf{h}\|_2^2 = 1. \end{aligned} \quad (10.10)$$

Zur Lösung dieses Problems kann die Lagrange'sche Multiplikator-Technik eingesetzt werden, welche die Nebenbedingung in die zu minimierende Funktion integriert. Nach einigen Schritten folgt hieraus das Eigenwertproblem

$$\mathbf{A}^T \mathbf{A} \mathbf{h} = \lambda \mathbf{h}, \quad (10.11)$$

wobei λ einen Lagrange'schen Multiplikator darstellt. Dabei wird eine vom Nullvektor verschiedene Lösung \mathbf{h} als Eigenvektor und das zugehörige λ als Eigenwert der Matrix $(\mathbf{A}^T \mathbf{A})$ bezeichnet. Die gesuchte Lösung für \mathbf{h} ist hierbei durch den Eigenvektor zum kleinsten Eigenwert definiert. Äquivalent dazu kann eine Singulärwertzerlegung für \mathbf{A} durchgeführt und die gesuchte Lösung für \mathbf{h} anhand des Eigenvektors zum kleinsten Singulärwert ermittelt werden.

Um die Genauigkeit der numerisch berechneten Lösung für \mathbf{H} zu erhöhen wird in ([HZ03], S. 109) der normalisierte DLT-Algorithmus vorgestellt. Hierbei müssen die übereinstimmenden Bildpunkte $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ zunächst anhand des unter Abschnitt B.3 beschriebenen Vorgehens in normalisierte Bildpunkte $\mathbf{p}_i^{r^*} \leftrightarrow \mathbf{p}_i^r$ transformiert werden.

Anschließend kann die Bestimmung der Homography-Matrix \mathbf{H}^r bezüglich der Punktepaare $\mathbf{p}_i^{r^*} \leftrightarrow \mathbf{p}_i^r$ durchgeführt werden. Dazu wird jede Koeffizientenmatrix \mathbf{A}_i aus Gleichung 10.8 mit den normalisierten Punktepaaren $\mathbf{p}_i^{r^*} \leftrightarrow \mathbf{p}_i^r$ parametrisiert. Nachdem \mathbf{H}^r erfasst wurde, kann eine Repräsentation der Homography-Matrix bezüglich der ursprünglichen Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ durch die Denormalisierung

$$\mathbf{H} = ({}^{r^*}\mathbf{T}_{n^*})^{-1} \mathbf{H}^r {}^r\mathbf{T}_n \quad (10.12)$$

erreicht werden. Die Transformationsmatrizen ${}^r\mathbf{T}_n$ und ${}^{r^*}\mathbf{T}_{n^*}$ sind auf Grund des erläuterten Normalisierungsvorgangs hinsichtlich der Bildpunkte gegeben (sh. Kapitel B.3).

Bei der Berechnung der Homography-Matrix \mathbf{H} kann es laut ([HZ03], S. 91-92) zu sogenannten degenerierten Konfigurationen bezüglich der verwendeten Punktepaare kommen. Im Rahmen einer minimalen Lösung, also bei $n = 4$ genutzten Punktepaaren, liegt eine degenerierte Konfiguration vor, sobald drei von vier Punkten bezüglich eines Bildes kollinear sind. Dies kann zu zwei unterschiedlichen Fällen führen.

Im ersten Fall sind die acht Einträge bezüglich der Koeffizientenmatrix \mathbf{A} aus Formel 10.9 nicht mehr linear unabhängig. \mathbf{A} besitzt dann einen Rang $r < 8$ und somit sind für \mathbf{H} mehrere Lösungen vorhanden. Im zweiten Fall existiert zwar eine exakte Lösung für \mathbf{H} , allerdings verfügt \mathbf{H} dann nur über den Rang $r = 1$. Eine solche Lösung ist unbrauchbar, da es sich bei \mathbf{H} , wie unter Gliederungspunkt 10.2.1 erläutert, um eine Matrix vollen Ranges handelt. Da degenerierte Konfigurationen also zu nicht verwendbaren Lösungen führen, müssen diese abgefangen werden.

10.2.3 Robuste Schätzung

Um die mit dem *Nearest-Neighbor*-Algorithmus detektierten vermeintlich übereinstimmenden Bildpunkte $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ von Ausreißern zu bereinigen, soll eine robuste Schätzung der Homography-Matrix \mathbf{H} vorgenommen werden. Diese Vorgehensweise ermöglicht auch das Extrahieren der Ausreißer aus $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$.

Für eine robuste Schätzung von \mathbf{H} soll der unter Abschnitt B.2 beschriebene RANSAC-Algorithmus mit adaptiver Parameterbestimmung genutzt werden. Hierbei wird der zu Beginn unbekannte Ausreißeranteil ϵ bezüglich $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ sukzessive ermittelt, da dieser zu Beginn nicht bekannt ist. Als nächstes wird erläutert wie die Parameter des Algorithmus B.2 zur robusten Schätzung von \mathbf{H} gesetzt werden müssen.

Da zur Berechnung einer minimalen Lösung für \mathbf{H} vier Punktpaare benötigt werden, gilt $n = 4$. Der Algorithmus erfordert, dass in jeder Iteration eine Schätzung von \mathbf{H} auf der Basis von $n = 4$ zufällig aus P ermittelten Elementen durchgeführt wird. Hierzu soll das normalisierte DLT-Verfahren aus Abschnitt 10.2.2 genutzt werden. Die Punktmenge P ist somit durch

$$P = \{[(\mathbf{p}_i^{r*})^T (\mathbf{p}_i^r)^T]^T\} \quad (10.13)$$

gegeben, wobei die Punktpaare $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ in normalisierte Übereinstimmungen $\mathbf{p}_i^{r*} \leftrightarrow \mathbf{p}_i^r$ transformiert wurden. Die zu schätzende Homography-Matrix ist demnach durch die Größe \mathbf{H}^r definiert.

Um degenerierte Schätzungen für \mathbf{H}^r zu vermeiden, müssen die erwähnten degenerierten Konfigurationen bezüglich der Bildpunkte abgefangen werden. Eine degenerierte Konfiguration tritt dabei dann auf, sobald drei der $n = 4$ zur Schätzung notwendigen Punkte bezüglich eines Bildes kollinear sind. In diesem Fall werden die ausgewählten Punkte verworfen und es werden zufällig $n = 4$ neue Elemente aus P zur Bestimmung von \mathbf{H}^r ausgewählt.

Im zweiten Schritt des Algorithmus B.2 soll die aktuelle Konsensmenge P_i bestimmt werden, deren Elemente innerhalb eines Distanzschwellwertes t bezüglich der aktuellen Schätzung \mathbf{H}_i^r liegen. P_i ist dabei eine Untermenge von P . Auf diese Weise soll festgestellt werden, bei welchen der Punktpaare aus P es sich tatsächlich um korrespondierende Bildpunkte bezüglich der aktuellen Schätzung \mathbf{H}_i^r handelt. Nachfolgend wird die hierzu erforderliche Berechnung am Beispiel eines Punktpaars $\mathbf{p}^{r*} \leftrightarrow \mathbf{p}^r$ durchgeführt.

Zunächst müssen die Distanzen $dist_1$ und $dist_2$ wie folgt bestimmt werden:

$$\begin{aligned} dist_1 &= \| \mathbf{H}_i^r \tilde{\mathbf{p}}^r - \tilde{\mathbf{p}}^{r*} \|_2^2, \\ dist_2 &= \| (\mathbf{H}_i^r)^{-1} \tilde{\mathbf{p}}^{r*} - \tilde{\mathbf{p}}^r \|_2^2. \end{aligned} \quad (10.14)$$

Hierbei muss darauf geachtet werden, dass die homogenen Vektoren aus den unter Gliederungspunkt 10.2.2 genannten Gründen vorab einheitlich skaliert werden müssen. Bei dem Punktpaar $\mathbf{p}^{r*} \leftrightarrow \mathbf{p}^r$ handelt es sich tatsächlich um korrespondierende Bildpunkte wenn die Bedingung

$$dist_1 + dist_2 < t \quad (10.15)$$

erfüllt ist. Nach der Terminierung des Algorithmus befinden sich die tatsächlich übereinstimmenden Bildpunkte in der Menge P_{best} (vgl. Schritt 4 des Algorithmus B.2). Hierbei handelt es sich um Punktpaare $\mathbf{p}_i^{r*} \leftrightarrow \mathbf{p}_i^r$ die von Ausreißern bereinigt wurden. Wenn die Indizes dieser Punktpaare bezüglich der Menge P mitgeführt werden, dann können die tatsächlich korrespondierenden Bildpunkte in der ursprünglichen Notation $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ ermittelt werden. Hierbei handelt es sich um die Punktpaare die zu Beginn dieses Unterkapitels normalisiert und nun von Ausreißern befreit wurden. Die neu gefundenen Übereinstimmungen $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ liegen dabei zwar subpixelgenau vor, sind aber mit dem unter Abschnitt 2.1.1 beschriebenen Bildrauschen behaftet.

10.3 Auswertung

Anhand der detektierten korrespondierenden Bildpunkte $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ kann die ansichtenbasierte Wiedererkennung von Gegenständen realisiert werden. Hierfür soll zunächst ein Beispiel gegeben werden. In Grafik 10.1 wird ein Handy gezeigt, welches in der zweiten Ansicht

10.2 wiedererkannt werden soll. Für beide Ansichten wurden die SIFT-Punkte berechnet und anhand des *Nearest-Neighbor*-Algorithmus ausgewertet. Hieraus resultieren die vermeintlichen Punktpaare aus Abbildung 10.3.

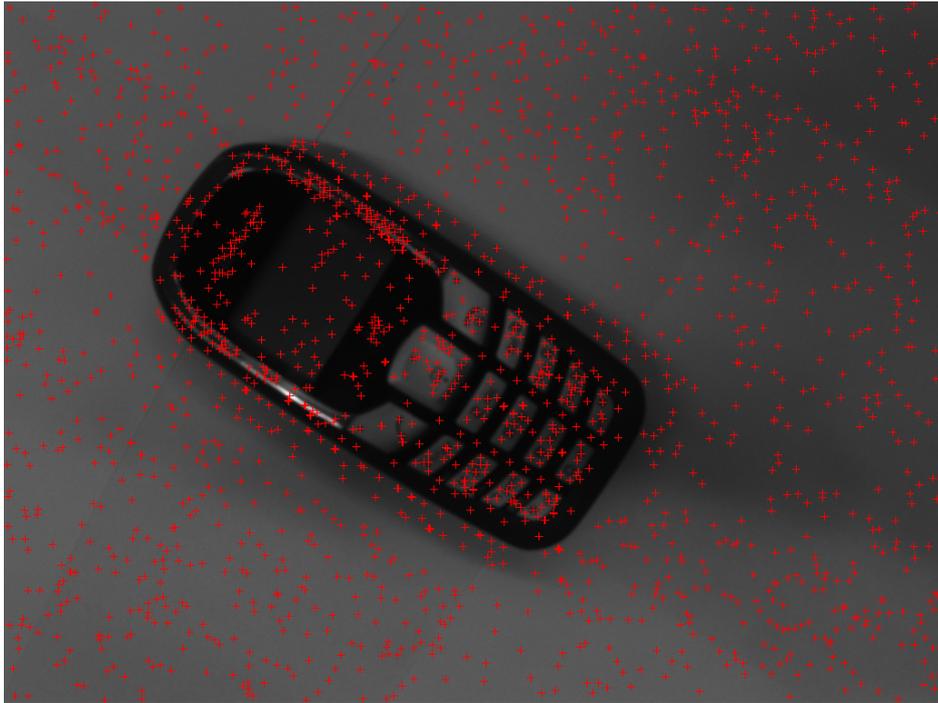
Es ist zu erkennen, dass es sich bei einigen der Punktpaare um Ausreißer handelt, bei denen Verbindungen zwischen nicht korrespondierenden Bildpunkten bestehen. Mittels des vorgestellten RANSAC-Algorithmus zur robusten Schätzung der Homography-Matrix \mathbf{H} konnten die tatsächlichen Übereinstimmungen aus Grafik 10.4 detektiert werden. Da sich die tatsächlichen Punktpaare dabei vollständig auf das Handy beziehen, konnte dieses in der zweiten Ansicht B wiedererkannt werden. Dies ist nur dadurch möglich, da sich das Handy in den Abbildungen 10.1 und 10.2 auf unterschiedlichen Hintergründen (Papier in B^* und Gegenstände auf grauem Untergrund in B) dargestellt wird.

Wäre dies nicht der Fall, dann würden in der Grafik 10.4 weitere Punktpaare vorhanden sein, welche sich nicht auf den Gegenstand beziehen. Eine gegenstandsbasierte Wiedererkennung wäre dann nicht möglich. Um während eines Greifvorgangs nicht zwangsläufig einen anderen Hintergrund als in der Referenzansicht B^* verwenden zu müssen, wird das folgende Vorgehen genutzt.

Unter Gliederungspunkt 3.1 wurde erläutert, dass zum Greifen eines Gegenstands eine Teilrekonstruktion von diesem benötigt wird. Diese soll dabei *offline* anhand von zwei perspektivisch unterschiedlichen Ansichten erstellt werden, welche die gleiche Szene zeigen. Beide Ansichten werden hierzu vorab in einem *Teach-In* Schritt aufgenommen. Eine dieser beiden Ansichten stellt das Zielbild B^* dar, welches während eines Greifvorgangs auch zur bildbasierten Regelung genutzt wird. Die zweite Ansicht wird mit B bezeichnet, in der sich der Gegenstand auf einem anderen Hintergrund als in B^* befindet.

Für beide Ansichten wird anschließend die hier vorgestellte Korrespondenzanalyse ausgeführt. Anschließend sind wie im gezeigten Beispiel nur die Punktpaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ bezüglich des Gegenstands vorhanden. Diese Punktpaare werden daraufhin zur Erstellung einer Teilrekonstruktion des Gegenstands benötigt. Die dafür notwendigen Berechnungen werden später in Kapitel 11 erläutert.

Anhand der zur Rekonstruktion verwendeten Punktpaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ kann *online* während eines Greifprozesses auch eine gegenstandsbasierte Wiedererkennung durchgeführt werden, indem für die Wiedererkennung nur die Referenz-Bildpunkte $\mathbf{p}_i^{n^*}$ mit ihren zugehörigen SIFT-Deskriptoren genutzt werden. Dementsprechend darf sich ein wiederzuerkennender Gegenstand *online* auf einem beliebigen Hintergrund befinden.

Abbildung 10.1: SIFT-Punkte (rot) aus der Referenzansicht B^* Abbildung 10.2: SIFT-Punkte (grün) aus einer zweiten Ansicht B

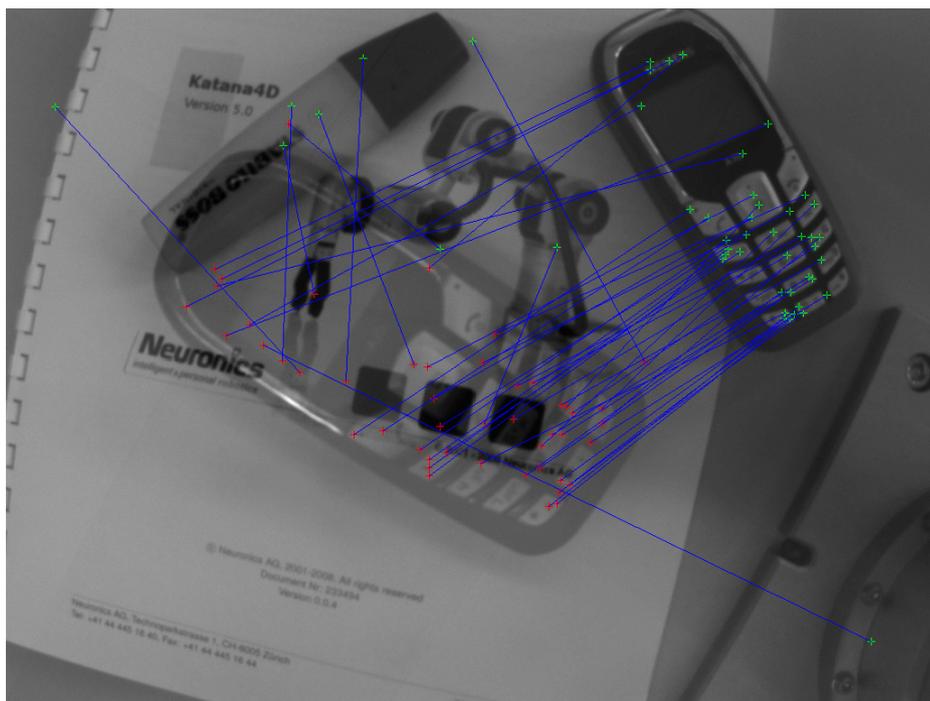


Abbildung 10.3: Vermeintliche Übereinstimmungen bezüglich der überlagerten Bilder 10.1 und 10.2

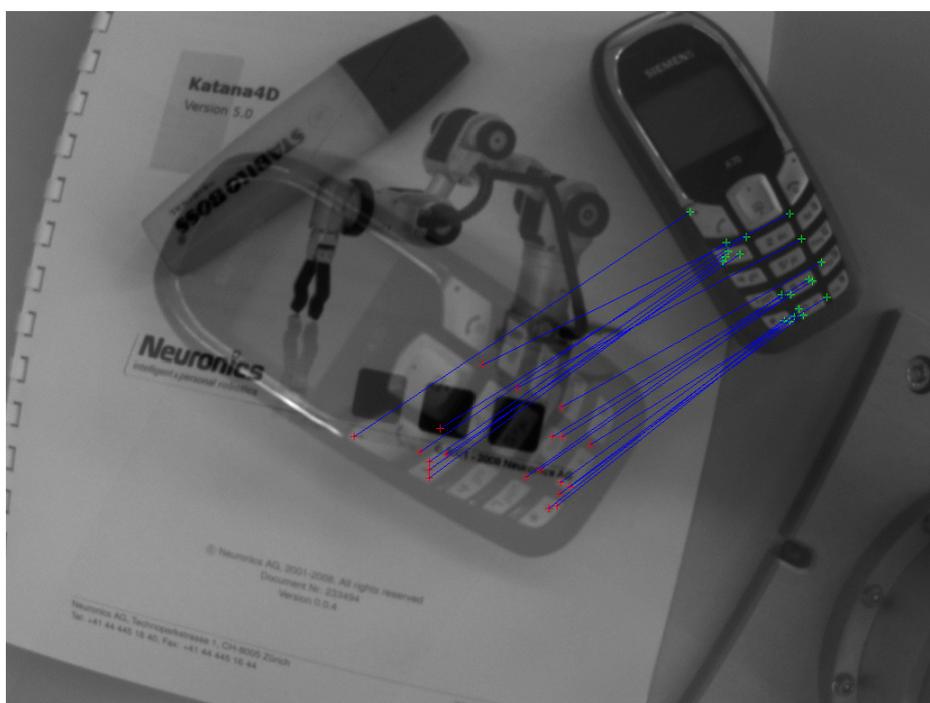


Abbildung 10.4: Tatsächliche Übereinstimmungen bezüglich der überlagerten Bilder 10.1 und 10.2

Um einen Gegenstand auf die beschriebene Weise wiedererkennen zu können, muss zusätzlich die Anzahl n der durch RANSAC detektierten Punktepaare $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ erfasst und mit einem Schwellwert t_{id} verglichen werden. Wenn die Bedingung $n > t_{id}$ erfüllt ist, dann wird davon ausgegangen, dass sich der Gegenstand aus der Referenzansicht B^* in der aktuellen Ansicht B

befindet.

Die Verwendung des Schwellwerts t_{id} ist erforderlich, da auf Grund des *Nearest-Neighbor*-Algorithmus aus Abschnitt 10.1 auch dann vermeintliche Punktpaare gefunden werden können, wenn sich ein wiederzuerkennender Gegenstand aus der Referenzansicht B^* nicht im aktuellen Kamerabild B befindet. Wenn in diesem Fall genügend vermeintliche Punktpaare vorhanden sind, dann wird die durch RANSAC geschätzte Homography-Matrix H in der Regel für vier Punktpaare gültig sein. Der Grund dafür ist, dass RANSAC in jedem Iterationsschritt eine minimale Lösung für H auf der Basis von vier Punktpaaren erzeugt. In Ausnahmefällen kann es vorkommen, dass hierbei minimal mehr als vier Punktpaare als gültig gekennzeichnet werden. Aus diesem Grund sollte der Schwellwert t_{id} auf einen ausreichend hohen Wert gesetzt werden.

Als nächstes soll ein Eindruck über die Qualität der beschriebenen Wiedererkennung von Gegenständen gegeben werden. Da die Anzahl der Punktpaare in Grafik 10.4 hoch genug ist, konnte das Handy in der Ansicht B wiedererkannt werden. Hierbei wird auch deutlich, dass die Wiedererkennung trotz eines hohen Auftretens von anderen Gegenständen robust durchgeführt werden konnte.

In einem weiteren Test wurde das Handy aus einer dritten Perspektive aufgenommen (vgl. Abbildung 10.5). Hierbei wurde die Kamera stärker gekippt als im Vergleich zu Grafik 10.2. Mittels der entwickelten Korrespondenzanalyse konnten die Punktpaare aus Grafik 10.6 bezüglich der Referenz- und der dritten Perspektive detektiert werden.

Obwohl sich der Gegenstand in Abbildung 10.5 im Kamerabild befindet, können nicht genügend Punktpaare für eine zuverlässige Wiedererkennung ermittelt werden. Des Weiteren ist zu erkennen, dass es sich bei einigen der Punktpaare um Ausreißer handelt. Der Grund dafür liegt darin, dass sich die Perspektive gegenüber der Referenzansicht aus Abbildung 10.1 stark geändert hat. Dies gilt vor allem für den Neigungswinkel.

Da SIFT-Merkmalvektoren nur eine partielle Invarianz gegenüber perspektivischer Änderungen besitzen, werden von vornherein nur wenig korrekte Übereinstimmungen gefunden. Somit ergibt sich eine ähnliche Situation wie in dem bereits dargelegten Fall, in dem sich ein wiederzuerkennender Gegenstand aus der Referenzansicht B^* nicht im aktuellen Kamerabild B befindet.

Aus dem letzten Beispiel folgt also, dass eine auf dem SIFT-Verfahren basierende Wiedererkennung perspektivisch gesehen nur in einer bestimmten Nachbarschaft bezüglich der Referenzansicht möglich ist. Wenn sich also ein gesuchter Gegenstand bereits im Kamerabild befindet, dann kann es vorkommen, dass der Gegenstand nicht wiedererkannt wird. Die Suche nach einem Gegenstand muss in diesem Fall fortgesetzt werden, bis eine Perspektive erreicht wurde in der der Gegenstand wiedererkannt werden kann.

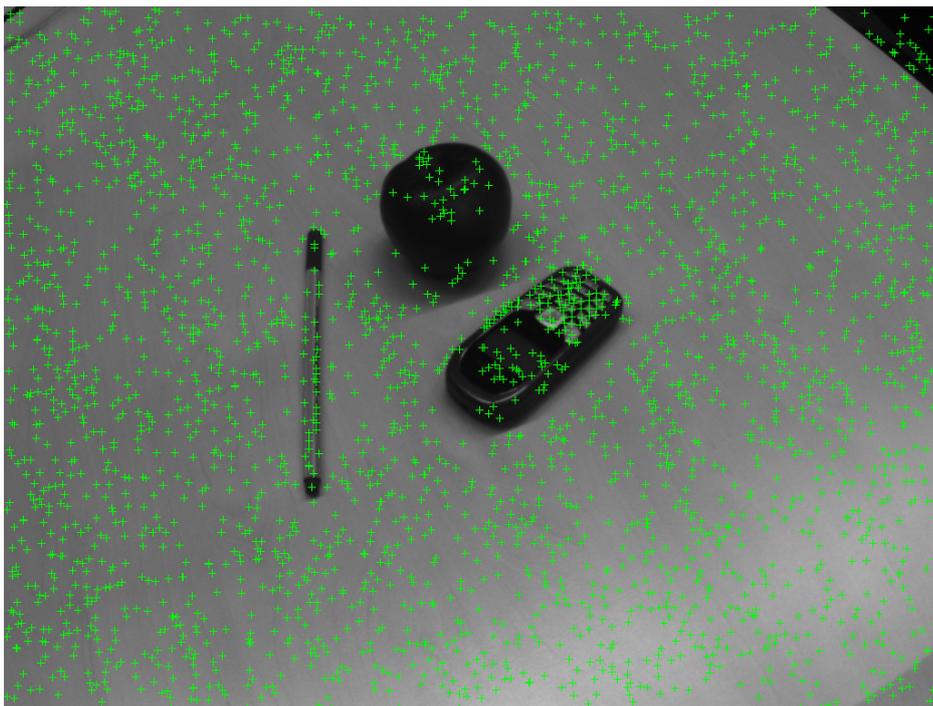


Abbildung 10.5: SIFT-Punkte (grün) aus einer dritten Ansicht B



Abbildung 10.6: Übereinstimmungen bezüglich der überlagerten Bilder 10.1 und 10.5

11 Rekonstruktion von Raumpunkten

Dieses Kapitel beschäftigt sich mit der Rekonstruktion von Raumpunkten zur Generierung eines partiellen Gegenstandsmodells. Bei Letzterem handelt es sich hierbei um eine räumliche Punktwolke, die durch die Rückprojektion von einzelnen korrespondierenden Bildpunkten aus zwei Ansichten entsteht. Es geht hierbei also nicht um die Erstellung einer vollständigen Teilrekonstruktion eines Objekts. Die durch die Rückprojektion errechneten Raumpunkte werden mit \mathbf{p}_i^w bezeichnet und definieren somit die unter Gliederungspunkt 7.1.2 eingeführten Werkstückpunkte, welche zur Bestimmung der Werkstück-Lage (${}^c\mathbf{R}_w, {}^c\mathbf{o}_w$) benötigt werden. Diese Lageinformation wird wiederum, wie unter Abschnitt 8 erläutert, für die Endeffektorpositionierung während der direkten Steuerung gebraucht.

Die Erstellung einer Teilrekonstruktion eines Gegenstands soll *offline* anhand von zwei perspektivisch unterschiedlichen Bildern erfolgen, welche die gleiche Szene zeigen. Beide Ansichten werden hierbei vorab in einem *Teach-In* Schritt aufgenommen. Eine der Ansichten stellt dabei das Zielbild B^* dar, welches ebenfalls für die bildpunkt-basierte Regelung verwendet wird. Die zweite Ansicht wird mit B bezeichnet, in der sich der Gegenstand aus den in Kapitel 10.3 erwähnten Gründen auf einem anderen Hintergrund als in B^* befindet.

Für beide Ansichten wird anschließend die entwickelte Korrespondenzanalyse ausgeführt. Anschließend stehen, wie in Grafik 10.4 exemplarisch dargestellt, Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ bezüglich des Gegenstands zur Verfügung. Diese Punktepaare stellen die Eingangsgrößen zur Rekonstruktion der Werkstückpunkte \mathbf{p}_i^w dar, so dass anschließend die Übereinstimmungen $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^w$ und $\mathbf{p}_i^n \leftrightarrow \mathbf{p}_i^w$ gelten.

Um eine Rekonstruktion der Werkstückpunkte \mathbf{p}_i^w auf Basis der Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ durchführen zu können, muss die Beziehung zwischen Bild- und Werkstückpunkten bekannt sein. Dieser Zusammenhang ist durch die Projektionsmatrix \mathbf{P} aus Gleichung 6.6 erfüllt. Es muss allerdings berücksichtigt werden, dass es sich bei $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ um normierte Bildpunkte handelt. Die Projektionsmatrix \mathbf{P} vereinfacht sich somit zu:

$$\begin{aligned} \tilde{\mathbf{p}}^n &= \underbrace{\mathbf{K} \begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{o}_w \end{bmatrix}}_{\mathbf{P}} \tilde{\mathbf{p}}^w \\ &= \underbrace{\mathbf{I}_{3 \times 3} \begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{o}_w \end{bmatrix}}_{\mathbf{P}} \tilde{\mathbf{p}}^w \\ &= \underbrace{\begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{o}_w \end{bmatrix}}_{\mathbf{P}} \tilde{\mathbf{p}}^w. \end{aligned} \quad (11.1)$$

Die Kalibriermatrix \mathbf{K} wird hierbei durch die (3×3) Einheitsmatrix $\mathbf{I}_{3 \times 3}$ ersetzt, da die Bildweite f bei der Berechnung von normierten Bildpunkten 1 entspricht (vgl. Formel 6.8). Somit bleiben in \mathbf{P} lediglich die extrinsischen Kameraparameter in Form der Werkstück-Lage (${}^c\mathbf{R}_w, {}^c\mathbf{o}_w$) übrig. \mathbf{P} ermöglicht hierbei die Abbildung eines Werkstückpunktes \mathbf{p}^w auf einen Bildpunkt \mathbf{p}^n bezüglich der Ansicht B . In diesem Zusammenhang wird \mathbf{P} auch als normalisierte Projektionsmatrix bezeichnet (vgl. [HZ03], S. 257).

Die Projektionsmatrix welche \mathbf{p}^w auf einen Bildpunkt \mathbf{p}^{n^*} bezüglich der Zielansicht B^* abbildet, wird nachstehend durch \mathbf{P}^* gekennzeichnet. Um die Rekonstruktion eines Werkstückpunktes \mathbf{p}^w umsetzen zu können, müssen neben den Übereinstimmungen $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ auch

die Projektionsmatrizen P und P^* bekannt sein. Die Rekonstruktion eines Raumpunktes p^w entspricht dabei der Berechnung des Schnittpunkts der Projektionsstrahlen die ausgehend von p^w auf die korrespondierenden Bildpunkte $p^{n^*} \leftrightarrow p^n$ treffen (vgl. Grafik 2.7).

Bisher sind lediglich die Punktepaare $p_i^{n^*} \leftrightarrow p_i^n$ gegeben. Im Folgenden werden zunächst die Rechenschritte vorgestellt, welche eine Bestimmung der zur Rekonstruktion notwendigen Projektionsmatrizen P und P^* ermöglicht. Hierbei wird die sogenannte Epipolargeometrie hinzugezogen, welche eine Schätzung der Lage $({}^cR_{c^*}, {}^cO_{c^*})$ der Kamera c^* im Ziel bezüglich der zweiten Kamera c ermöglicht.

Auf dieser Grundlage ist es möglich eine Rekonstruktion der Werkstückpunkte p_i^w vorzunehmen. Die resultierenden Raumpunkte müssen in anschließenden Berechnungen verbessert werden, um eine wahrheitsgetreue Teilrekonstruktion eines Gegenstandes zu erhalten.

11.1 Epipolargeometrie

Die Epipolargeometrie beschreibt die Projektionsgeometrie zwischen zwei Ansichten, welche die gleiche Szene zeigen. Dieser Aspekt wird durch Grafik 11.1 verdeutlicht. Die dadurch entstehenden Beziehungen werden nachstehend erläutert (vgl. [HZ03], S. 239-241). Wenn ein Raumpunkt p^w auf den Bildpunkt p^* in der Ansicht B^* abgebildet wird, dann befindet sich der zu p^* korrespondierende Bildpunkt p auf der sogenannten Epipolarlinie l in der Ansicht B . Die Epipolarlinie l stellt dabei die Abbildung des von p^w nach p^* führenden Projektionsstrahls dar. Die Bezeichnung p^* respektive p gibt hierbei an, dass es sich um beliebige Bildpunkte hinsichtlich der unterschiedlichen Koordinatensysteme des zentralen Projektionsmodells aus Kapitel 6.1 handelt.

Aus der Sicht der Aufnahme B^* gilt, dass sich der Bildpunkt p^* auf der Epipolarlinie l^* befindet. Diese definiert wiederum die Abbildung des von p^w nach p führenden Projektionsstrahls. Gilt es nun zum Beispiel den zu p^* korrespondierenden Bildpunkt p zu finden, dann können die beschriebenen Zusammenhänge wie folgt genutzt werden. Unter der Bedingung dass die Epipolarlinie l bekannt ist, kann die Suche nach p auf l eingeschränkt werden.

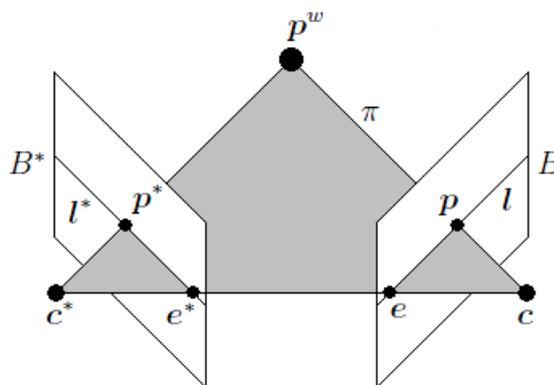


Abbildung 11.1: Epipolargeometrie

Eine weitere Größe ist durch den Epipol e gegeben, welcher das in die Ansicht B projizierte Kamerazentrum c^* darstellt. Der Epipol e ist dabei der Bildpunkt, in dem sich alle Epipolarlinien der Aufnahme B schneiden. Der Epipol e^* definiert dementsprechend die Abbildung des

Kamerazentrums c . Die in Grafik 11.1 gezeigten Punkte $\mathbf{p}^w, \mathbf{p}^*, \mathbf{p}, \mathbf{e}^*, \mathbf{e}, \mathbf{c}^*$ und c liegen dabei alle auf der Epipolarebene π . Die Epipolarlinien \mathbf{l}^*, \mathbf{l} werden durch die Schnittlinien von π in den Bildebenen B^* und B erzeugt.

Raumpunkte die sich unter- oder oberhalb von \mathbf{p}^w befinden, führen zu weiteren Epipolarlinien und Epipolarebenen. Letztere erzeugen ein Ebenenbüschel und besitzen ihre gemeinsame Gerade in Form der sogenannten Basislinie, welche von \mathbf{c}^* nach c führt. Die mathematische Beschreibung der Epipolargeometrie ist durch die Fundamentalmatrix \mathbf{F} gegeben.

11.1.1 Fundamentalmatrix

Im Folgenden wird die Fundamentalmatrix \mathbf{F} und die damit verbundenen Abbildungsvorschriften erläutert. Auf dieser Grundlage wird eine Ausgleichungsmethode zur Berechnung von \mathbf{F} vorgestellt.

11.1.1.1 Übersicht

Die Fundamentalmatrix \mathbf{F} ist eine homogene (3×3) Matrix vom Rang $r = 2$ und bildet einen homogenen Bildpunkt $\tilde{\mathbf{p}}^{n*}$ auf die Epipolarlinie $\mathbf{l} = [a \ b \ c]^T$ bezüglich der Ansicht B wie folgt ab (vgl. [HZ03], S. 243-246):

$$\mathbf{l} = \mathbf{F}\tilde{\mathbf{p}}^{n*} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \tilde{\mathbf{p}}^{n*}. \quad (11.2)$$

Es sei angemerkt, dass Gleichung 11.2 auch für beliebige Bildpunkte bezüglich der unterschiedlichen Koordinatensysteme des zentralen Projektionsmodells aus Kapitel 6.1 definiert werden kann. Zur Vereinfachung wird die Abbildung hier allerdings nur für normierte Bildpunkte eingeführt. Die Berechnung der Epipolarlinie \mathbf{l}^* bezüglich der Ansicht B^* ist durch

$$\mathbf{l}^* = \mathbf{F}^T \tilde{\mathbf{p}}^n \quad (11.3)$$

gegeben. Mit Hilfe der Epipolarlinie \mathbf{l} kann die homogene Geradengleichung

$$0 = (\tilde{\mathbf{p}}^n)^T \mathbf{l} \quad (11.4)$$

angegeben werden, die durch den zu $\tilde{\mathbf{p}}^{n*}$ korrespondierenden Bildpunkt $\tilde{\mathbf{p}}^n$ erfüllt wird (vgl. Grafik 11.1). Wird \mathbf{l} hierbei durch die rechte Seite aus Formel 11.2 ersetzt, dann ergibt sich:

$$0 = (\tilde{\mathbf{p}}^n)^T \mathbf{F}\tilde{\mathbf{p}}^{n*}. \quad (11.5)$$

Somit besteht ein direkter Zusammenhang zwischen der Fundamentalmatrix \mathbf{F} und korrespondierenden Bildpunkten $\mathbf{p}^{n*} \leftrightarrow \mathbf{p}^n$. Diese Beziehung wird nachfolgend genutzt, um \mathbf{F} zu bestimmen.

11.1.1.2 Ausgleichung

Zu Beginn dieses Kapitels wurde erläutert, dass die Punktepaare $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ die Eingangsgrößen zur Rekonstruktion der Werkstückpunkte \mathbf{p}_i^w darstellen. Um \mathbf{F} anhand von $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ ermitteln zu können, wird Formel 11.5 zunächst folgendermaßen umgeformt:

$$\begin{aligned}
0 &= \mathbf{A} \underbrace{[f_1 \cdots f_9]^T}_{\mathbf{f}} \\
&= \begin{bmatrix} x_n x_{n^*} & x_n y_{n^*} & x_n & y_n x_{n^*} & y_n y_{n^*} & y_n & x_{n^*} & y_{n^*} & 1 \end{bmatrix} \mathbf{f}.
\end{aligned} \tag{11.6}$$

Die dritte Spalte von $\tilde{\mathbf{p}}^{n^*}$ und $\tilde{\mathbf{p}}^n$ wurde hierbei der Einfachheit halber auf den Wert 1 festgelegt, so dass nur noch die euklidischen Komponenten x_n, y_n, x_{n^*} und y_{n^*} vorhanden sind. Bei einer Verwendung von n Punktepaaren ergibt sich das lineare Gleichungssystem:

$$\mathbf{0} = \underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{bmatrix}}_{\mathbf{A}_{ges}} \mathbf{f}. \tag{11.7}$$

\mathbf{A}_{ges} wird im Folgenden der Einfachheit halber mit \mathbf{A} bezeichnet. Da es sich bei der gesuchten Größe \mathbf{F} um eine homogene (3×3) Matrix vom Rang $r = 2$ handelt, besitzt diese sieben FHG. Dies ermöglicht die Berechnung von \mathbf{F} anhand von $n = 7$ Punktepaaren, welche in ([HZ03], S. 281) erläutert wird. Hierbei besteht allerdings die Einschränkung, dass die Berechnung nur für den Minimalfall von $n = 7$ übereinstimmenden Bildpunkten ausgelegt ist.

In der vorliegenden Arbeit sollen mehr als sieben Punktepaare zur Berechnung von \mathbf{F} genutzt werden, da die Übereinstimmungen $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ mit Bildrauschen behaftet sind. Die Nutzung von möglichst vielen Punktepaaren bietet dabei die Möglichkeit die Genauigkeit einer Ausgleichslösung zu erhöhen. Aus diesem Grund soll der nachstehend vorgestellte 8-Punkte Algorithmus verwendet werden ([HZ03], S. 279-281).

Bei diesem Vorgehen muss die Bedingung $n \geq 8$ bezüglich der Anzahl der Punktepaare erfüllt sein. \mathbf{A} besitzt dann immer den Rang $r = 8$, unter der Voraussetzung dass die verwendeten n Punktepaare nicht fehlerbehaftet sind. In diesem Fall verfügt \mathbf{A} über einen 1-dimensionalen Kern, da die zu berechnende Größe \mathbf{f} neun Elemente enthält. Demzufolge kann für \mathbf{f} eine vom Nullvektor verschiedene Lösung bestimmt werden, welche jedoch nur bis auf eine Skalierung $s \neq 0$ gegeben ist. Da die Fundamentalmatrix \mathbf{F} im Allgemeinen sowieso mit einem Skalierungsfaktor $s \neq 0$ multipliziert werden kann, ist die Lösung für \mathbf{f} somit exakt.

In dem Fall, dass das lineare Gleichungssystem aus Formel 11.7 überbestimmt ($n > 8$) und die verwendeten Punktepaare fehlerbehaftet sind, kann es passieren das für \mathbf{f} nur noch die triviale Lösung $\mathbf{0}_{9 \times 1}$ vorliegt. Diese Situation tritt ein, wenn die Matrix \mathbf{A} den Rang $r = 9$ besitzt. Unter diesen Umständen muss eine fehlerminimierte Schätzung für \mathbf{f} ausgeführt werden. Wenn jedoch $n = 8$ fehlerbehaftete Punktepaare verwendet, dann ist für \mathbf{f} eine eindeutige Lösung bis auf eine Skalierung $s \neq 0$ vorhanden. Auf Grund der abweichenden Punkte ist diese Lösung allerdings nicht exakt.

Nachfolgend wird eine allgemeine Methode zur Bestimmung von \mathbf{f} beschrieben, welche im Fall von $n \geq 8$ Punktepaaren eingesetzt werden kann. Hierbei ist es unerheblich ob die Punktepaare exakt oder fehlerbehaftet vorliegen. Der Lösungsweg stimmt mit der Vorgehensweise zur Schätzung der Homography-Matrix \mathbf{H} aus Abschnitt 10.2.2 überein. Es wird ein Minimierungsproblem mit einer Nebenbedingung auf der Grundlage von Gleichung 11.7 formuliert:

$$\begin{aligned} \min_{\mathbf{f}} \|\mathbf{A}\mathbf{f}\|_2^2, \\ \text{so dass } \|\mathbf{f}\|_2^2 = 1. \end{aligned} \quad (11.8)$$

Zur Lösung dieses Problems kommt die Lagrange'sche Multiplikator-Technik zum Einsatz. Auf diese Weise wird die Nebenbedingung in die zu minimierende Funktion integriert und nach einigen Schritten folgt hieraus das Eigenwertproblem

$$\mathbf{A}^T \mathbf{A} \mathbf{f} = \lambda \mathbf{f}. \quad (11.9)$$

Die gesuchte Lösung für \mathbf{f} ist dabei durch den Eigenvektor zum kleinsten Eigenwert definiert. Äquivalent dazu kann eine Singulärwertzerlegung für \mathbf{A} durchgeführt und die gesuchte Lösung für \mathbf{f} anhand des Eigenvektors zum kleinsten Singulärwert ermittelt werden. Die durch den hier vorgestellte 8-Punkte Algorithmus berechnete Lösung für die Fundamentalmatrix \mathbf{F} wird den vollen Rang $r = 3$ besitzen. Da es sich bei \mathbf{F} , wie unter Gliederungspunkt 11.1.1.1 erläutert, um eine Matrix vom Rang $r = 2$ handelt, muss diese Bedingung nachträglich erfüllt werden.

Hierzu wird \mathbf{F} mittels der Singulärwertzerlegung dekomponiert und der kleinste Singulärwert auf den Wert 0 gesetzt. \mathbf{F} wird anschließend wieder zusammengesetzt und besitzt dann den Rang $r = 2$.

Um die Genauigkeit der numerisch berechneten Lösung für \mathbf{F} zu erhöhen wird vorab, wie im Fall der Homography-Matrix \mathbf{H} , eine Normalisierung der zur Schätzung von \mathbf{F} genutzten Bildpunkte ausgeführt. Dieses Vorgehen wird laut ([HZ03], S. 281-282) als der normalisierte 8-Punkte Algorithmus bezeichnet. Zunächst müssen die Punktepaare $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ mittels der unter Abschnitt B.3 beschriebenen Methode in normalisierte Bildpunkte $\mathbf{p}_i^{r*} \leftrightarrow \mathbf{p}_i^r$ transformiert werden.

Daraufhin kann die Bestimmung der Fundamentalmatrix \mathbf{F}^r bezüglich der Punktepaare $\mathbf{p}_i^{r*} \leftrightarrow \mathbf{p}_i^r$ durchgeführt werden. Dazu wird jede Koeffizientenmatrix \mathbf{A}_i aus Gleichung 11.6 mit den normalisierten Punktepaaren $\mathbf{p}_i^{r*} \leftrightarrow \mathbf{p}_i^r$ parametrisiert. Nach der Erfassung von \mathbf{F}^r , kann eine Repräsentation der Fundamentalmatrix bezüglich der ursprünglichen Punktepaare $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ anhand der Denormalisierung

$$\mathbf{F} = ({}^r\mathbf{T}_n)^T \mathbf{F}^r {}^{r*}\mathbf{T}_{n*} \quad (11.10)$$

berechnet werden. Die Transformationsmatrizen ${}^r\mathbf{T}_n$ und ${}^{r*}\mathbf{T}_{n*}$ sind auf Grund des vorgestellten Normalisierungsvorgangs hinsichtlich der Bildpunkte gegeben (sh. Kapitel B.3).

Da die zur Berechnung von \mathbf{F} verwendeten Punktepaare $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ mit Bildrauschen behaftet sind, wird für \mathbf{F} lediglich eine Schätzung ermittelt werden können. Dies wird wiederum Auswirkungen auf die Genauigkeit der angestrebten Rekonstruktion haben. In [HZ03] werden weiterführende Methoden vorgestellt, um die Genauigkeit von \mathbf{F} auf Basis der hier beschriebenen Vorgehensweise zu verbessern. Dies wird in der vorliegenden Arbeit jedoch nicht weiter berücksichtigt.

11.1.1.3 Degeneration

Bei der vorgestellten Berechnung der Fundamentalmatrix \mathbf{F} kann es laut ([HZ03], S. 296-297)

zu degenerierten Konfigurationen kommen. Hierbei müssen die folgenden zwei Fälle unterschieden werden.

- **Strukturdegeneration:** Wenn alle nach $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ projizierten Werkstückpunkte \mathbf{p}_i^w koplanar sind, dann existieren mehrere Lösungen für \mathbf{F} .
- **Bewegungsdegeneration:** In dem Fall, dass die Kamerazentren \mathbf{c}^* und \mathbf{c} aus Grafik 11.1 koinzidieren, besteht keine Translation zwischen den Kameras. Unter dieser Bedingung existieren unabhängig von der vorhandenen Rotation mehrere Lösungen für \mathbf{F} .

Weiterhin besteht die Problematik, dass die Schätzung für \mathbf{F} numerisch schlecht konditioniert ist, wenn es sich um eine Konfiguration handelt die bezüglich der beschriebenen Fälle beinahe degeneriert ist. Unter diesen Umständen besteht eine hohe Anfälligkeit der Lösung für \mathbf{F} gegenüber den Abweichungen der Eingangsdaten.

Um eine Schätzung der Fundamentalmatrix \mathbf{F} zu ermöglichen, sollen die degenerierten Konfigurationen wie folgt berücksichtigt werden. Wie zu Beginn dieses Kapitel erläutert, soll die Erstellung einer Teilrekonstruktion eines Gegenstands *offline* anhand von zwei perspektivisch unterschiedlichen Bildern B^* und B vorgenommen werden. Beide Ansichten zeigen dabei die gleiche Szene. Zusätzlich zu diesen Bildern werden zwei weitere Bilder \hat{B}^* und \hat{B} aufgenommen, die allerdings nicht mehr den zu rekonstruierenden Gegenstand zeigen. Stattdessen wird das Gittermuster aus Grafik 11.2 aufgenommen.

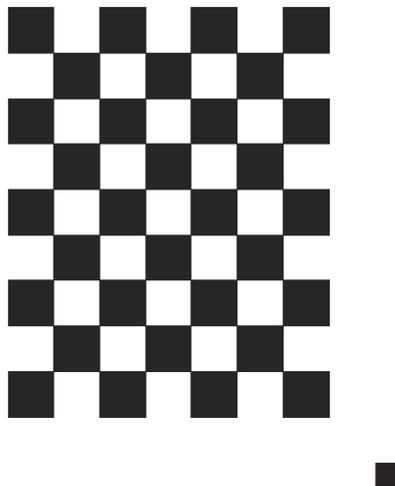


Abbildung 11.2: Gittermuster zur Rekonstruktion

Das Gittermuster soll hierbei als Hilfsobjekt dienen, um die beschriebene Strukturdegeneration vermeiden zu können. Wenn zum Beispiel eine Schachtel mit sechs ebenen Seiten teilrekonstruiert werden soll, dann kann es passieren, dass alle nach $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ abgebildeten Werkstückpunkte \mathbf{p}_i^w koplanar sind. Mit Hilfe des Gittermusters soll ermöglicht werden, dass zur Schätzung von \mathbf{F} zusätzliche Punktepaare $\hat{\mathbf{p}}_i^{n*} \leftrightarrow \hat{\mathbf{p}}_i^n$ hinzugezogen werden können. Letztere stellen dabei die Abbildungen der Werkstückpunkte $\hat{\mathbf{p}}_i^w$ dar, welche nicht koplanar sind.

Um dies zu gewährleisten wird das Gittermuster aus Grafik 11.2 auf einer gewölbten Oberfläche angebracht. Bei den zusätzlichen Übereinstimmungen $\hat{\mathbf{p}}_i^{n*} \leftrightarrow \hat{\mathbf{p}}_i^n$ handelt es sich um bestimmte Eckpunkte des Gittermusters, welche pro Ansicht wie folgt ermittelt werden. Zunächst werden alle Ecken des Gittermusters durch das Verfahren *Harris corner detector* (vgl.

[HS88]) bestimmt. Die Position der Eckpunkte liegt dabei allerdings nur grob vor und somit wird anschließend eine Verbesserung durch die subpixelgenaue Berechnung der Eckpunkte vorgenommen. Das Resultat wird exemplarisch in Grafik 11.3 gezeigt.

Um die Eckpunkte der Aufnahmen \hat{B}^* und \hat{B} einander zuordnen zu können, werden die Ecken pro Ansicht automatisch durchnummeriert. Der Ausgangspunkt ist hierbei durch das schwarze Quadrat in der rechten unteren Ecke aus Grafik 11.2 definiert. Für die Nummerierung der Eckpunkte werden diese als Vektoren betrachtet. Auf Grundlage der Richtungen und Längen aller Differenzvektoren, können die Vektoren in eine vordefinierte Reihenfolge gebracht werden. Der erwähnte Ausgangspunkt kann erfasst werden, da die kürzesten Differenzvektoren bezüglich des schwarze Quadrats in der rechten unteren Ecke aus Grafik 11.2 gegeben sind.

Nach der Nummerierung der Punkte befinden sich diese in einer geordneten Reihenfolge und können einander bezüglich der Ansichten \hat{B}^* und \hat{B} zugeordnet werden. Bei der weiteren Verwendung wird hier jedoch nur eine Untermenge betrachtet. Dabei handelt es sich um 48 geordnete Ecken pro Ansicht, welche für eine Ansicht exemplarisch in Grafik 11.4 dargestellt werden. Der Grund für die Einschränkung auf die 48 Eckpunkte ist durch die Genauigkeit der Position der einzelnen Eckpunkte gegeben. Die Position einer jeden Ecken wurde zwar subpixelgenau bestimmt, allerdings ist die Genauigkeit der Position der 48 inneren Eckpunkte höher als die der äußeren Ecken. Dies liegt daran, dass sich die inneren Ecken durch eine Überlappung aus jeweils zwei Quadraten ergeben.

Anhand der erläuterten 48 Eckpunkte aus den Ansichten \hat{B}^* und \hat{B} sind die Übereinstimmungen $\hat{\mathbf{p}}_i^{b^*} \leftrightarrow \hat{\mathbf{p}}_i^b$ gegeben. Letztere müssen zur weiteren Verwendung mittels der unter Gliederungs-punkt 6.3 beschriebenen Vorgehensweise entzerrt werden, woraus die Punktepaare $\hat{\mathbf{p}}_i^{n^*} \leftrightarrow \hat{\mathbf{p}}_i^n$ resultieren. Um Letztere zusammen mit den gegenstandsbezogenen Punktepaaren $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ zur Schätzung von \mathbf{F} verwenden zu können, müssen die folgenden Bedingungen eingehalten werden.

1. Die Lage (${}^c\mathbf{R}_{c^*}$, ${}^c\mathbf{o}_{c^*}$) der Kamera c^* relativ zur zweiten Kamera c darf sich bei der Aufnahme der Bilder B^* und \hat{B}^* nicht ändern. Das Gleiche gilt aus Sicht der zweiten Kamera c bezüglich c^* bei der Aufnahme der Bilder B und \hat{B} .
2. Weiterhin darf der zu rekonstruierende Gegenstand zwischen der Aufnahme von B^* und B nicht bewegt werden. Dies gilt ebenso für das Gittermuster während der Aufnahme der Bilder \hat{B}^* und \hat{B} .

Die Einhaltung dieser Bedingungen ist notwendig, da die Epipolargeometrie aus Abbildung 11.1 lediglich für zwei Ansichten definiert ist. Um die Bedingungen berücksichtigen zu können, werden die Aufnahmen von B^* , B , \hat{B}^* und \hat{B} mittels der Kamera durchgeführt, die sich am Endeffektor des Katana-Roboterarms befindet. Auf diese Weise kann für beide Aufnahme-positionen jeweils eine feste Endeffektor-Lage ${}^0\mathbf{T}_e$ definiert werden. Zunächst werden die Ansichten B^* und B von einem Gegenstand und die entsprechenden zwei Endeffektor-Lagen gesichert. Anschließend wird der Gegenstand durch das beschriebene Gittermuster ausgetauscht und zur Aufnahme der Bilder \hat{B}^* und \hat{B} werden entsprechend die vordefinierten Endeffektor-Lagen eingestellt.

Auf Grund des Aspekts das die gemeinsam verwendeten Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ und $\hat{\mathbf{p}}_i^{n^*} \leftrightarrow \hat{\mathbf{p}}_i^n$ anhand von insgesamt vier Ansichten ermittelt wurden, kann es vorkommen, dass sich einzelne

Punkte in den gesamten Übereinstimmungen doppelt vorkommen. Da sich dieser Sachverhalt negativ auf die Schätzung der Fundamentalmatrix F auswirken kann, werden vorab alle doppelten Punkte entfernt. Dabei muss darauf geachtet werden, dass beim Entfernen eines Punktes auch sein übereinstimmender Punkt beseitigt wird.

Abschließend muss die zu Beginn erläuterte Bewegungsdegeneration berücksichtigt werden. Um eine Bewegungsdegeneration zu vermeiden, muss darauf geachtet werden, dass sich bei der Aufnahme der Bilder B^* , B , \hat{B}^* und \hat{B} eine Translation ${}^c o_{c^*} \neq 0$ bezüglich der Lage $({}^c R_{c^*}, {}^c o_{c^*})$ der Kamera c^* relativ zur zweiten Kamera c vorhanden ist. Die Translation sollte dabei nicht zu gering sein, da es sich sonst um eine beinahe degenerierte Konfiguration handelt. Unter diesen Umständen wäre eine Schätzung für F numerisch schlecht konditioniert.

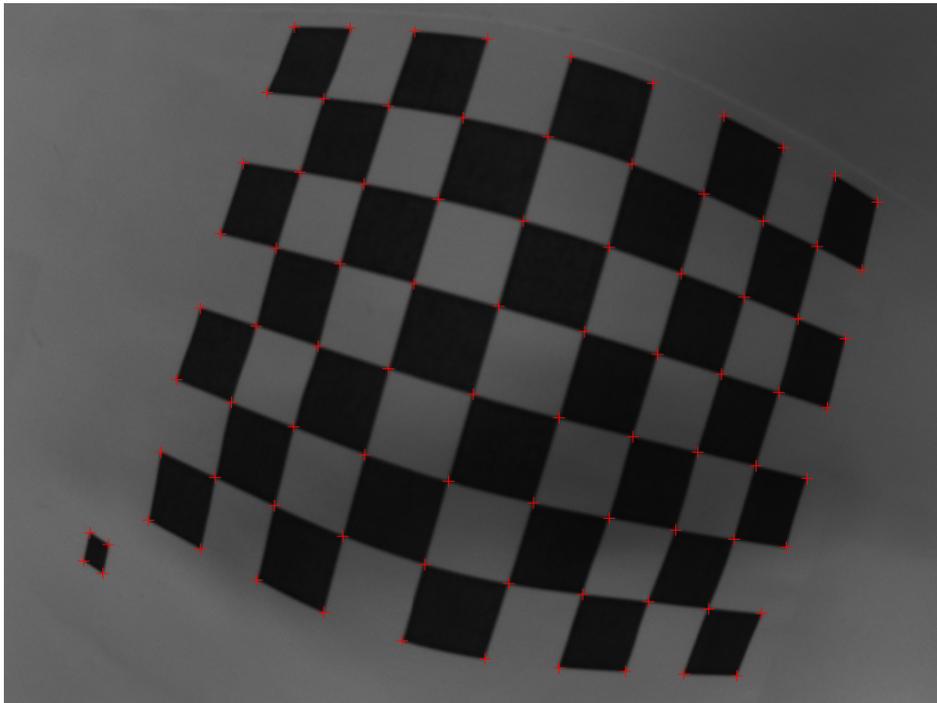


Abbildung 11.3: Eckpunkte des Gittermusters

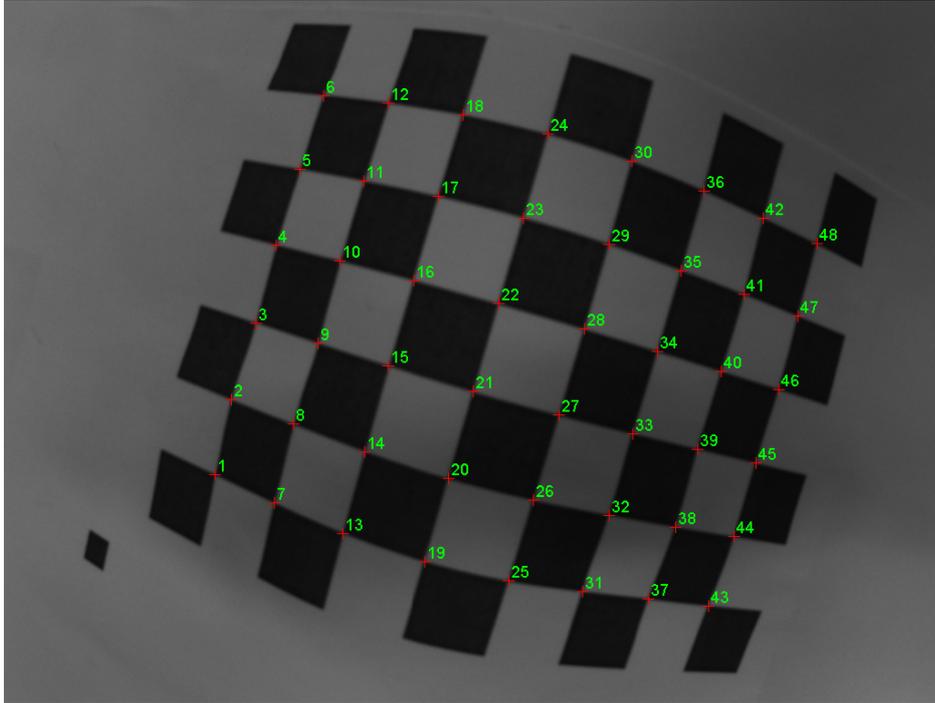


Abbildung 11.4: Ausgewählte Eckpunkte des Gittermusters

11.1.2 Essentialmatrix

Auf Basis der geschätzten Fundamentalmatrix F könnte bereits eine Rekonstruktion der zu $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ korrespondierenden Raumpunkte durchgeführt werden. Laut ([HZ03], S. 266-267) würde dies allerdings zu keiner sogenannten metrischen Rekonstruktion führen. Somit würden die Winkel zwischen Linien und Längenverhältnisse im Modell nicht erhalten bleiben. Um dies jedoch zu erreichen, müssen neben F weitere Informationen vorliegen. Eine Möglichkeit besteht darin, die Kalibriermatrix K aus Gleichung 6.7 zu berücksichtigen. Dies führt zu der im Folgenden erläuterten Essentialmatrix.

11.1.2.1 Beschreibung

Die Essentialmatrix E wird anhand der Kalibriermatrizen K^* und K anhand von

$$E = K^T F K^* \quad (11.11)$$

ermittelt (vgl. [HZ03], S. 257). Da F in Abschnitt 11.1.1.2 bereits mittels normierter Bildpunkte $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ berechnet wurde, können K^* und K jeweils durch die (3×3) Einheitsmatrix $I_{3 \times 3}$ ersetzt werden:

$$E = I_{3 \times 3} F I_{3 \times 3} = F. \quad (11.12)$$

Somit ist F in der vorliegenden Arbeit äquivalent zu E . Die Essentialmatrix kann ebenfalls anhand der erläuterten Lage $({}^c R_{c^*}, {}^c o_{c^*})$ der Kamera c^* relativ zur zweiten Kamera c berechnet werden (zur Herleitung sh. [HZ03] S. 243-244):

$$E = S({}^c o_{c^*}) {}^c R_{c^*}. \quad (11.13)$$

Hierbei steht $S(\cdot)$ für die schiefsymmetrische Matrix zu einem gegebenen Vektor $\in \mathfrak{R}^3$. Das Ziel

ist es nun die gegebene Essentialmatrix E zu zerlegen, um die Lage (${}^cR_{c^*}$, ${}^c\mathbf{o}_{c^*}$) bestimmen zu können.

11.1.2.2 Zerlegung

Zu Beginn dieses Kapitels wurde erläutert, dass für eine Rekonstruktion zunächst die Projektionsmatrizen P und P^* berechnet werden müssen. Die Projektionsmatrix P^* wird wie folgt definiert (vgl. [HZ03] S. 257):

$$P^* = \begin{bmatrix} {}^cR_w & {}^c\mathbf{o}_w \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & \mathbf{0}_{3 \times 1} \end{bmatrix}. \quad (11.14)$$

Durch die Festlegung der Werkstück-Lage (cR_w , ${}^c\mathbf{o}_w$) anhand von $(I_{3 \times 3}, \mathbf{0}_{3 \times 1})$ koinzidiert das Kamera-Koordinatensystem $(KS)_{c^*}$ mit dem Werkstück-Koordinatensystem $(KS)_w$. Die Projektionsmatrix P kann demnach mittels der Lage (${}^cR_{c^*}$, ${}^c\mathbf{o}_{c^*}$) definiert werden:

$$P = \begin{bmatrix} {}^cR_{c^*} & {}^c\mathbf{o}_{c^*} \end{bmatrix}. \quad (11.15)$$

Im Folgenden wird beschrieben wie die Lage (${}^cR_{c^*}$, ${}^c\mathbf{o}_{c^*}$) durch eine Zerlegung der Essentialmatrix E bestimmt werden kann. Laut ([HZ03], S. 257-258) handelt es sich bei einer gegebenen Matrix E nur dann um eine Essentialmatrix wenn die ersten beiden Singulärwerte von E äquivalent sind und der dritte den Wert 0 besitzt. Da zur Schätzung von E fehlerbehaftete Bildpunkte verwendet werden, wird diese Bedingung für E nicht erfüllbar sein. Aus diesem Grund muss die Bedingung nachträglich erfüllt werden. Dazu wird eine Singulärwertzerlegung für E durchgeführt, wodurch die Matrizen U , Σ und V bestimmt werden:

$$E = U\Sigma V^T. \quad (11.16)$$

Bei Σ handelt es sich um eine (3×3) Diagonalmatrix, welche die Singulärwerte von E enthält. Um die Singulärwert-Bedingung für E zu erfüllen wird Σ durch die Diagonalmatrix $\text{diag}(1, 1, 0)$ ersetzt und anschließend wird E wieder zusammengesetzt:

$$E = U\text{diag}(1, 1, 0)V^T. \quad (11.17)$$

Für eine Zerlegung von E anhand von Gleichung 11.13 wird in ([HZ03], S. 258-259) wieder eine Singulärwertzerlegung für E durchgeführt. Man könnte annehmen das eine Zusammensetzung von E in Formel 11.17 somit redundant ist und die Matrizen U , $\text{diag}(1, 1, 0)$ und V direkt verwendet werden können. Allerdings wirkt sich die Komposition wegen des Auswechselns von Σ durch $\text{diag}(1, 1, 0)$ bei einer erneuten Zerlegung von E auf die Matrizen U und V aus. Anhand der Matrizen U , V und

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11.18)$$

können für den Rotationsanteil zwei mögliche Faktorisierungen

$$\begin{aligned} R_1 &= UWV^T, \\ R_2 &= UW^T V^T \end{aligned} \quad (11.19)$$

bestimmt werden. Da das Vorzeichen von E bei dessen Schätzung nicht ermittelt werden kann, existieren durch die Negierung von R_1 und R_2 zwei weitere mögliche Faktorisierungen für den Rotationsanteil. Bei einer Rotationsmatrix R handelt es sich im Allgemeinen um eine orthogo-

nale Matrix mit der Determinante $\det(\mathbf{R}) = 1$. Somit kann das richtige Vorzeichen für \mathbf{R}_1 und \mathbf{R}_2 anhand der Signumfunktion $\text{sgn}(x)$ wie folgt bestimmt werden:

$$\begin{aligned}\mathbf{R}_1 &= \mathbf{R}_1 \text{sgn}(\det(\mathbf{R}_1)), \\ \mathbf{R}_2 &= \mathbf{R}_2 \text{sgn}(\det(\mathbf{R}_2)).\end{aligned}\tag{11.20}$$

Für den Translationsanteil \mathbf{o} können mittels

$$\begin{aligned}\mathbf{o}_1 &= \mathbf{U}[0 \ 0 \ 1]^T, \\ \mathbf{o}_2 &= -\mathbf{U}[0 \ 0 \ 1]^T\end{aligned}\tag{11.21}$$

ebenfalls zwei mögliche Zerlegungen vorgenommen werden. Daraus folgt, dass für die Projektionsmatrix \mathbf{P} vier mögliche Lösungen konstruiert werden können:

$$\begin{aligned}\mathbf{P}_1 &= \begin{bmatrix} \mathbf{R}_1 & \mathbf{o}_1 \end{bmatrix}, \\ \mathbf{P}_2 &= \begin{bmatrix} \mathbf{R}_1 & \mathbf{o}_2 \end{bmatrix}, \\ \mathbf{P}_3 &= \begin{bmatrix} \mathbf{R}_2 & \mathbf{o}_1 \end{bmatrix}, \\ \mathbf{P}_4 &= \begin{bmatrix} \mathbf{R}_2 & \mathbf{o}_2 \end{bmatrix}.\end{aligned}\tag{11.22}$$

Um die richtige Lösung für \mathbf{P} zu ermitteln kann die Rekonstruktion eines Werkstückpunktes \mathbf{p}^w für alle vier Lösungen durchgeführt werden. Dabei ist folgender Sachverhalt entscheidend. In ([HZ03], S. 259-260) wird beschrieben, dass sich ein rekonstruierter Raumpunkt \mathbf{p}^w nur in einem der vier Fälle korrekterweise vor beiden Kameras c^* und c befindet. In den übrigen Fällen gilt dies nicht. Somit kann die richtige Lösung ermittelt werden in dem der Tiefenabstand eines rekonstruierten Raumpunktes \mathbf{p}^w bezüglich beider Kameras c^* und c berechnet wird. Dies kann mit Hilfe der Transformationsmatrizen ${}^{c^*}\mathbf{T}_w$ und ${}^c\mathbf{T}_w$ erreicht werden (sh. Formel 5.7). Letztere sind anhand der Rotations- und Translationsbestandteile der Projektionsmatrizen \mathbf{P}^* und \mathbf{P} gegeben.

Da es sich bei der zerlegten Essentialmatrix \mathbf{E} lediglich um eine Schätzung handelt, werden die vier möglichen Lösungen für \mathbf{P} ebenso fehlerbehaftet sein. Aus diesem Grund soll die erläuterte Vorgehensweise zur Bestimmung der korrekten Lösung auf Basis der Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ für alle hierzu korrespondierenden Werkstückpunkte \mathbf{p}_i^w vorgenommen werden. Dementsprechend wird für \mathbf{P} die Lösung ausgewählt, welche alle Raumpunkte \mathbf{p}_i^w vor beide Kameras rückprojiziert. Die bezüglich der korrekten Projektionsmatrix rekonstruierten Raumpunkte \mathbf{p}_i^w stellen gleichzeitig die angestrebte Teilrekonstruktion eines Gegenstandes dar.

Nachstehend wird erläutert wie die Rekonstruktion von \mathbf{p}_i^w auf Basis der Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ und der Projektionsmatrizen \mathbf{P}^* und \mathbf{P} durchgeführt werden kann. Hierfür ist anzumerken, dass die rotatorischen und translatorischen Bestandteile von \mathbf{P} durch die Zerlegung von \mathbf{E} nur bis auf eine Skalierung bestimmt werden können. Dies liegt daran, dass es sich bei \mathbf{E} um eine homogene Matrix mit lediglich fünf FHG handelt. Eine Rotationsmatrix \mathbf{R} und ein Translationsvektor \mathbf{o} besitzen jedoch jeweils drei FHG.

Auf Grund dieser Tatsache werden die rekonstruierten Raumpunkte \mathbf{p}_i^w von ihrer korrekten Position abweichen. Allerdings wird auf die erläuterte Weise trotzdem eine metrische Rekonstruktion erreicht und somit bleiben Winkel zwischen Linien und Längenverhältnisse im Modell wahrheitsgemäß erhalten.

11.2 Triangulation

Die Rekonstruktion eines Raumpunkts \mathbf{p}^w entspricht einer Triangulation mittels des zu \mathbf{p}^w korrespondierenden Punktepaars $\mathbf{p}^{n*} \leftrightarrow \mathbf{p}^n$ und der Projektionsmatrizen \mathbf{P}^* und \mathbf{P} . Als Ausgangspunkt hierfür dient Gleichung 11.1, welche den Zusammenhang zwischen einem Bildpunkt \mathbf{p}^n und seinem korrespondierenden Werkstückpunkt \mathbf{p}^w auf Basis der Projektionsmatrix \mathbf{P} herstellt.

Zur Berechnung eines Raumpunkts \mathbf{p}^w wird nachfolgend das vorgestellte DLT-Verfahren aus Kapitel 10.2.2 eingesetzt. Dementsprechend können die Zusammenhänge aus Formel 11.1 mit Hilfe des Kreuzproduktes ausgedrückt werden:

$$\mathbf{0}_{3 \times 1} = \tilde{\mathbf{p}}^n \times \mathbf{P} \tilde{\mathbf{p}}^w. \quad (11.23)$$

Unter Berücksichtigung der Übereinstimmung $\mathbf{p}^{n*} \leftrightarrow \mathbf{p}^n$ und der Projektionsmatrizen \mathbf{P}^* und \mathbf{P} folgt hieraus:

$$\mathbf{0}_{4 \times 1} = \underbrace{\begin{bmatrix} x_{n^*} \mathbf{p}_3 - \mathbf{p}_1 \\ y_{n^*} \mathbf{p}_3 - \mathbf{p}_2 \\ x_n \mathbf{p}_3^* - \mathbf{p}_1^* \\ y_n \mathbf{p}_3^* - \mathbf{p}_2^* \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T}_{\tilde{\mathbf{p}}^w}. \quad (11.24)$$

Dabei gilt

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} \quad (11.25)$$

für \mathbf{P} und entsprechend auch für \mathbf{P}^* . Die dritte Spalte von $\tilde{\mathbf{p}}^{n*}$ und $\tilde{\mathbf{p}}^n$ wurden in Formel 11.24 zur Vereinfachung auf den Wert 1 festgelegt, so dass nur noch die euklidischen Komponenten x_n, y_n, x_{n^*} und y_{n^*} übrig bleiben.

Die gesuchte Lösung für $\tilde{\mathbf{p}}^w$ kann wie im Fall der Berechnung der Fundamentalmatrix \mathbf{F} anhand einer Singulärwertzerlegung für \mathbf{A} ermittelt werden. Der Raumpunkt $\tilde{\mathbf{p}}^w$ wird dann anhand des Eigenvektors zum kleinsten Singulärwert ermittelt. Ein rückprojizierter Raumpunkt \mathbf{p}^w kann auf diese Weise allerdings nur bis auf eine unbekannte Skalierung s bestimmt werden. Führt man eine Rekonstruktion für alle Punktepaare $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ durch, dann ist s jedoch für alle resultierenden Werkstückpunkte \mathbf{p}_i^w gleich.

11.3 Bündelblockausgleich

Da die Triangulation auf der Basis von fehlerbehafteten Größen durchgeführt wird, werden auch die daraus hervorgehenden Werkstückpunkte \mathbf{p}_i^w mit Abweichungen versehen sein. Mit Hilfe eines sogenannten Bündelblockausgleichs kann die Genauigkeit der Teilrekonstruktion verbessert werden. Hierbei wird das nicht-lineare Optimierungsproblem

$$\min_{\mathbf{P}^*, \mathbf{P}, \mathbf{p}_i^w} \sum_{i=1}^n \|f(\mathbf{P}^*, \mathbf{p}_i^w) - \mathbf{p}_i^{n*}\|_2^2 + \sum_{i=1}^n \|f(\mathbf{P}, \mathbf{p}_i^w) - \mathbf{p}_i^n\|_2^2 \quad (11.26)$$

definiert, welches auf dem zentralen Projektionsmodell aus Kapitel 6.1 basiert. Neben der Teilrekonstruktion \mathbf{p}_i^w werden hier auch die Projektionsmatrizen \mathbf{P}^* und \mathbf{P} eingepasst. Letztere enthalten dabei lediglich die rotatorischen und translatorischen Bestandteile, da es sich um normalisierte Projektionsmatrizen handelt. Als Startwert dienen hierbei die bisher ermittelten Größen, welche aus der Zerlegung der Essentialmatrix \mathbf{E} und der Triangulation von Raumpunkten hervorgehen. Zur Durchführung der Optimierung kann zum Beispiel der *Levenberg-Marquardt-Algorithmus* genutzt werden.

11.4 Skalierung der Rekonstruktion

Mit den bisher erläuterten Methoden kann eine metrische Teilrekonstruktion eines Gegenstands erstellt werden. Ein solches Modell stellt eine wahrheitsgemäße Abbildung dar, die aus den erläuterten Gründen allerdings lediglich bis auf eine unbekannte Skalierung s gegeben ist. Für die unter Abschnitt 7.1.2 vorgestellte Kalibrierung der extrinsischen Kameraparameter müssen die rückprojizierten Raumpunkte \mathbf{p}_i^w allerdings korrekt skaliert vorliegen. Ein korrekt skaliertes Modell wird auch als euklidische Rekonstruktion bezeichnet.

Um die bisher erlangte metrische Teilrekonstruktion in ein euklidisches Modell transformieren zu können, wird nachstehend erläutert wie der Skalierungsfaktor s ermittelt werden kann. Zunächst soll noch einmal das Gittermuster aus Kapitel 11.1.1.3 eingesetzt werden. Es wurde erläutert, dass die zusätzlichen Punktepaare $\hat{\mathbf{p}}_i^{n*} \leftrightarrow \hat{\mathbf{p}}_i^n$ bezüglich des Gittermusters berechnet werden können. Führt man die vorgestellte Triangulation ebenfalls für $\hat{\mathbf{p}}_i^{n*} \leftrightarrow \hat{\mathbf{p}}_i^n$ durch, so entstehen die zusätzlichen Raumpunkte $\hat{\mathbf{p}}_i^w$. Dieses Prozedur kann für zwei beliebige Raumpunkte $\hat{\mathbf{p}}_1^w$ und $\hat{\mathbf{p}}_2^w$ vorgenommen werden, um den Abstand r_{ist} zu ermitteln:

$$r_{ist} = \|\hat{\mathbf{p}}_2^w - \hat{\mathbf{p}}_1^w\|_2. \quad (11.27)$$

Anhand einer manuellen Messung des tatsächlichen Abstands r_{soll} der zu $\hat{\mathbf{p}}_1^w$ und $\hat{\mathbf{p}}_2^w$ korrespondierenden Punkte auf dem Gittermuster, kann der Skalierungsfaktor s durch

$$s = \frac{r_{soll}}{r_{ist}} \quad (11.28)$$

bestimmt werden. Zur Erstellung eines euklidischen Modells werden die Werkstückpunkte \mathbf{p}_i^w jeweils mit s multipliziert.

11.5 Auswertung

Die vorgestellte Methode zur Rekonstruktion von Raumpunkten basiert auf der Zerlegung der Essentialmatrix \mathbf{E} . Auf diese Weise wird die Projektionsmatrix \mathbf{P} bis auf eine Skalierung bestimmt. Alternativ könnte \mathbf{P} anhand des Vorgehens aus Abschnitt 11.1.1.3 direkt bestimmt werden, da bei der Aufnahme der Bilder B^* und B bereits jeweils die Lage des Endeffektors relativ zum Umwelt-Koordinatensystem bekannt ist. Anhand der durchgeführten Hand-Auge-Kalibrierung ist ebenfalls die Kamerapose bezüglich des Endeffektors gegeben. Allerdings würden in \mathbf{P} somit Kalibrierfehler einfließen und folglich würde sich die Genauigkeit der Rekonstruktion im Vergleich zum vorgestellten Verfahren verschlechtern.

Abschließend wird die Erstellung einer Teilrekonstruktion von einem Gegenstand exemplarisch durchgeführt. Hierfür wurden von einer Tee-Schachtel zwei Ansichten B^* und B aufgenommen. Aus den in Kapitel 10.3 erwähnten Gründen befindet sich die Tee-Schachtel in den beiden

Ansichten jeweils auf einem anderen Hintergrund. Anhand der entwickelten Korrespondenzanalyse wurden für B^* und B die Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ berechnet (vgl. Grafik 11.5).

Anschließend wurde die hier dargelegte Rekonstruktion von Raumpunkten durchgeführt. Das Resultat ist anhand der zu $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ korrespondierenden Werkstückpunkte \mathbf{p}_i^w gegeben, welche in den Abbildungen 11.6 und 11.7 gezeigt werden. In Grafik 11.5 ist zu erkennen, dass die Übereinstimmungen $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ bezüglich zweier Ebenen der Tee-Schachtel gegeben sind. Aus diesem Grund sind die rekonstruierten Raumpunkte \mathbf{p}_i^w ebenfalls in zwei Ebenen unterteilt.

In Grafik 11.6 wird eine Sicht in beide Ebenen der Teilrekonstruktion gezeigt. Es wird deutlich, dass es sich wie bei der realen Tee-Schachtel um orthogonale Ebenen handelt. Weiterhin ist erkennbar, dass sich einige Punkte außerhalb der Ebenen befinden. Hierbei handelt es sich um Ausreißer in der Teilrekonstruktion.

Neben den rückprojizierten Raumpunkten werden auch die Kameras c^* (rot) und c (grün) mittels ihrer Lage bezüglich des Werkstücks dargestellt. Diese Lageinformationen wurden anhand des Algorithmus 7.1 auf Basis der Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ und der hierzu korrespondierenden rekonstruierten Werkstückpunkte \mathbf{p}_i^w ermittelt.



Abbildung 11.5: Punktepaare bezüglich der überlagerten Ansichten B^* und B

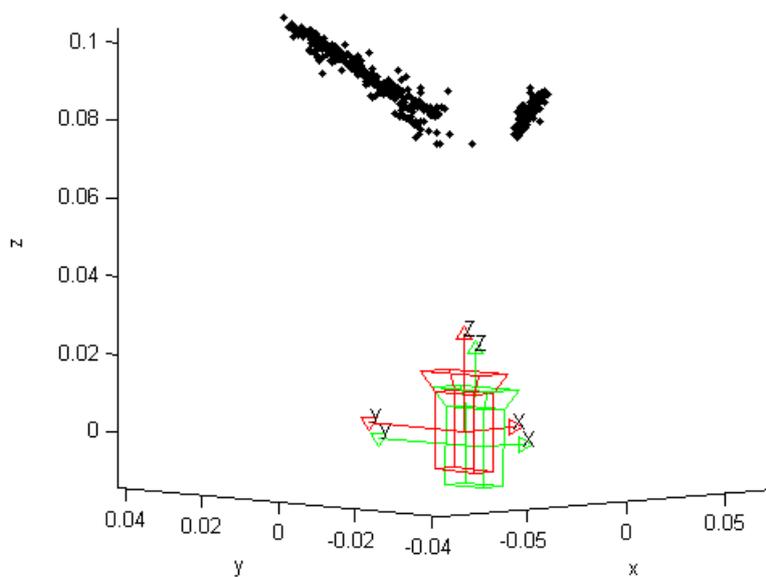


Abbildung 11.6: Sicht in die Ebenen der rekonstruierten Teebox

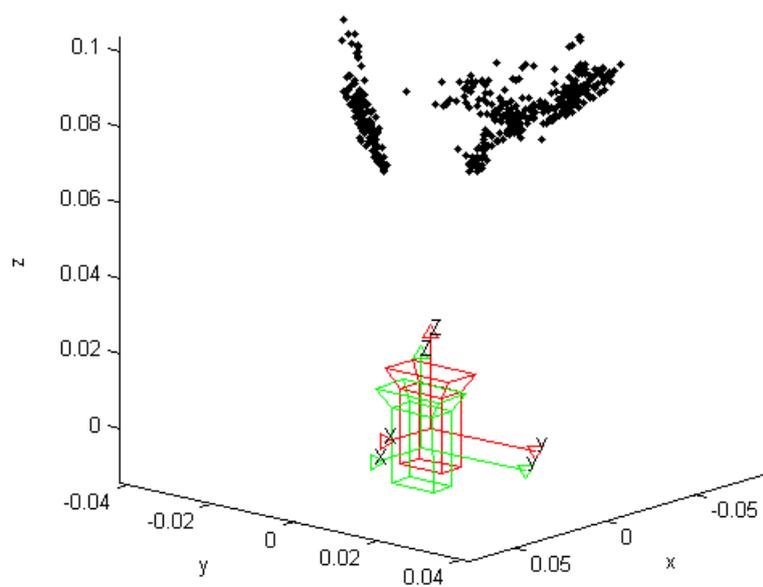


Abbildung 11.7: Sicht auf die Ebenen der rekonstruierten Teebox

12 Synthese

In den vorangegangenen Kapiteln wurden die einzelnen Systemkomponenten zur Realisierung des Greifkonzepts aus Kapitel 3 modelliert. Zur Beschreibung des Gesamt-Systems sollen die Module nachfolgend miteinander kombiniert werden. Zunächst wird erläutert, welche Vorbereitungen für einen Greifvorgang getroffen werden müssen. Hierbei geht es um *Teach-In* Schritte, Vorabberechnungen und die Erfassung von Greifparametern. Daraufhin wird die Durchführung eines Greifvorgangs dargelegt und es wird erläutert mit welchen software-technischen Mitteln das Gesamt-System realisiert wurde.

12.1 Vorbereitung eines Greifvorgangs

Der erste Schritt bei der Vorbereitung eines Greifvorgangs beläuft sich auf die Generierung einer Teilrekonstruktion für den zu greifenden Gegenstand. Hierfür werden in einem *Teach-In* Schritt zunächst zwei perspektivisch unterschiedliche Bilder B^* und B von dem Gegenstand aufgenommen, welche die gleiche Szene zeigen. B^* stellt dabei auch die Referenzansicht hinsichtlich der bildpunktbasierten Regelung dar. Bei der Aufnahme der beiden Ansichten ist darauf zu achten, dass sich der Gegenstand in B^* auf einem anderen Hintergrund als in B befindet. Auf diese Weise wird in der anschließenden Korrespondenzanalyse gewährleistet, dass sich die detektierten korrespondierenden Bildpunkte $p_i^{n^*} \leftrightarrow p_i^n$ auf den Gegenstand beziehen (vgl. Kapitel 10).

Mit Hilfe der Punktepaare $p_i^{n^*} \leftrightarrow p_i^n$ wird daraufhin eine Rekonstruktion der zu $p_i^{n^*} \leftrightarrow p_i^n$ übereinstimmenden Werkstückpunkte p_i^w vorgenommen (sh. Abbildung 12.1). Die entstehende Punktwolke stellt eine partielle Rekonstruktion des Gegenstands dar.

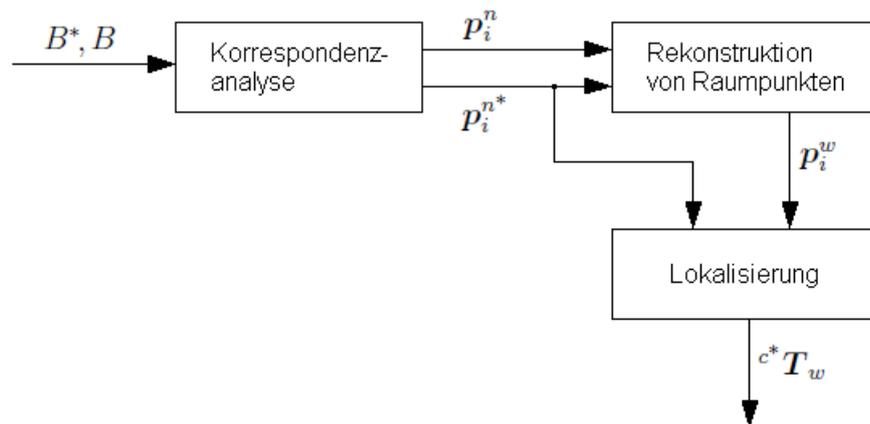


Abbildung 12.1: Erfassung von Greifparametern und Rekonstruktionsdaten

Anhand der korrespondierenden Bild- und Werkstückpunkte $p_i^{n^*} \leftrightarrow p_i^w$ kann die Werkstück-Lage ${}^{c^*}T_w$ bezüglich der Kamera im Ziel berechnet werden. Hierfür wird der auf RANSAC basierende Algorithmus 7.1 verwendet, um mit Ausreißern bezüglich der rückprojizierten Werkstückpunkte umgehen zu können. ${}^{c^*}T_w$ repräsentiert dabei die vorgegebene Greiflage, welche *online* während der direkten Steuerung zur groben Positionierung des Endeffektors benötigt wird (vgl. Abschnitt 8).

Bis zu diesem Punkt wurden alle Daten ermittelt, welche für die Suche nach einem Gegenstand und die anschließende Positionierung der Roboterhand in eine Greiflage bekannt sein müssen. Letztere setzt sich, wie unter Gliederungspunkte 3 beschrieben, aus einer Kopplung des Prinzips der direkten Steuerung mit der bildpunktbasierter Regelung zusammen. Bei den benötigten Daten handelt es sich zunächst um die detektierten Bildpunkte p_i^{n*} bezüglich der Referenzansicht B^* , welche zu den Bildpunkten p_i^n aus der Ansicht B korrespondieren. Die Bildpunkte p_i^n wurden nur für die Rekonstruktion benötigt, so dass im Folgenden nur noch die Referenz-Bildpunkte p_i^{n*} betrachtet werden. Neben den Referenz-Bildpunkten werden auch die zugehörigen SIFT-Merkmalvektoren d_i^* für den weiteren Gebrauch gesichert. Letztere entstammen dabei aus der vorangegangenen Korrespondenzanalyse (sh. auch Kapitel 10).

Weiterhin wird die Teilrekonstruktion in Form der rückprojizierten Werkstückpunkte p_i^w zur weiteren Verwendung gespeichert. Dies gilt auch für die vorgegebene Greiflage ${}^c T_w$. Alle bisher erläuterten Daten werden nochmal in der Tabelle 12.1 zusammengefasst.

In Kapitel 2.1.3 wurde erläutert, dass der Roboterarm nach der bildpunktbasierter Regelung unter Umständen weitere Bewegungen zum Greifen eines Objekts durchführen muss. Dies liegt daran, dass sich der Gegenstand für den gesamten Zeitraum der Regelung im Blickfeld der Kamera befinden muss. Die zusätzlichen Bewegungen werden wie folgt festgelegt. Da die durch B^* repräsentierte Greiflage anhand der bildpunktbasierter Regelung exakt eingestellt werden kann, können ausgehend von der erreichten Greiflage fest vorgegebene Posen bezüglich des Objekts angefahren werden. Anschließend befindet sich der Endeffektor in einer Lage, in der die Hand zum Greifen eines Gegenstandes lediglich geschlossen werden muss.

Die Berechnung der anzufahrenden Posen geschieht dabei auf Grundlage der Pose des Endeffektors bezüglich des Umwelt-Koordinatensystems $(KS)_0$. Nachfolgend wird hierfür ein Beispiel gegeben. In einem weiteren *Teach-In* Schritt werden die Endeffektor-Lagen ${}^0 T_{e^*}$ und ${}^0 T_{e1}$ gespeichert. ${}^0 T_{e^*}$ definiert dabei die Lage des Endeffektors, wenn die Kamera das zur bildpunktbasierter Regelung benötigte Zielbild B^* aufnimmt. In der Lage ${}^0 T_{e1}$, muss sich die Roboterhand zum Greifen lediglich schließen. Während der Erfassung der Lagen ${}^0 T_{e^*}$ und ${}^0 T_{e1}$ muss darauf geachtet werden, dass der Gegenstand nicht verschoben wird. Mittels der erfassten Lageinformationen kann die Pose ${}^{e^*} T_{e1}$ berechnet werden:

$$\begin{aligned} {}^{e^*} T_{e1} &= ({}^0 T_{e^*})^{-1} {}^0 T_{e1} \\ &= {}^{e^*} T_0 {}^0 T_{e1}. \end{aligned} \quad (12.1)$$

Letztere gibt dabei die Lage des Endeffektors bezüglich der Greiflage an, welche zur Laufzeit nach der bildbasierten Regelung eingestellt werden kann. Um Beschädigungen am Roboter auf Grund von unvorhersehbaren Trajektorien zu vermeiden, kann dieses Vorgehen spezifisch auf n Stationen erweitert werden. Eine zusätzliche Station bezieht sich dabei immer auf die vorherige. Für den beschriebenen Fall würde sich eine weitere Station durch die Lage ${}^{e^*} T_{e2}$ ergeben.

Nachdem die endgültige Lage des Endeffektors eingestellt wurde, muss die Roboterhand zum Greifen eines Objekts nur noch geschlossen werden. Wie weit sich die Hand schließen muss, wird ebenfalls vorab in einem *Teach-In* Schritt ermittelt. Bei der Roboterhand handelt es sich in der vorliegenden Arbeit um einen Winkelgreifer mit zwei Fingern. Aus diesem Grund muss lediglich ein Gelenkwinkel $q_{grripper}$ bestimmt werden. Die hier dargelegten Parameter für die zusätzlichen Greifbewegungen werden in Tabelle 12.1 zusammengefasst.

Symbol	Bedeutung	Verwendung
$\mathbf{p}_i^{n^*}$	Referenz-Bildpunkte	Suche nach einem Gegenstand und
\mathbf{d}_i^*	SIFT-Merkmalvektoren bezüglich $\mathbf{p}_i^{n^*}$	
\mathbf{p}_i^w	Rekonstruierte Werkstückpunkte. Es gilt $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^w$	Positionierung des Endeffektors
${}^{c^*}\mathbf{T}_w$	Vorgegebene Greiflage	
$\{{}^{c^*}\mathbf{T}_{e^1}, \dots, {}^{c^*}\mathbf{T}_{e^n}\}$	Folgelagen bezüglich der Greiflage	Abschließende Greifbewegungen
$q_{gripper}$	Gelenkwinkel der Roboterhand beim Greifen	

Tabelle 12.1: Rekonstruktionsdaten und Greifparameter

12.2 Durchführung eines Greifvorgangs

Im Folgenden wird erläutert wie die einzelnen Schritte eines Greifvorgangs auf Basis der vorab ermittelten Parameter aus Tabelle 12.1 ausgeführt werden. Ein Greifvorgang wurde in Kapitel 3.2 in vier Schritte aufgeteilt. Hierbei handelt es sich um die initiale Suche nach dem Gegenstand, die direkte Steuerung, die bildpunkt-basierte Regelung und das abschließende Greifen des Gegenstandes. Die direkte Steuerung und die anschließende bildpunkt-basierte Regelung ergeben zusammen das Positionier-System. Dieses hat die Aufgabe den Endeffektor in eine vorgegebene Greiflage bezüglich des Gegenstands zu positionieren.

12.2.1 Suche nach dem Gegenstand

Für die Positionierung wird vorausgesetzt, dass sich der zu greifende Gegenstand im Blickfeld der Kamera befindet. Um dies zu ermöglichen, sucht der Roboter seinen Arbeitsraum anhand eines bestimmten Musters ab, bis der Gegenstand wiedererkannt wird. Hierfür wird in jedem Schritt das aktuelle Kamerabild B ausgewertet. Wenn sich der gesuchte Gegenstand nicht in B befindet, dann verändert der Roboter seine Position geringfügig und es wird ein neues Bild aufgenommen.

Die Wiedererkennung eines Objekts wird dabei anhand der Korrespondenzanalyse aus Abschnitt 10 durchgeführt. Als Eingangsdaten für die Korrespondenzanalyse dienen die *offline* erfasste Zielansicht B^* und das aktuelle Bild B . Allerdings wird B^* im vorliegenden Fall nicht vollständig verarbeitet. Stattdessen werden lediglich die bereits vorhandenen Referenz-Bildpunkte $\mathbf{p}_i^{n^*}$ und deren zugehörige SIFT-Merkmalvektoren \mathbf{d}_i^* verwendet (vgl. Tabelle 12.1).

Durch die Korrespondenzanalyse werden die Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ ermittelt. Bei den Bildpunkten $\mathbf{p}_i^{n^*}$ handelt es sich hierbei um eine Untermenge der eingangs verwendeten Referenz-Bildpunkte. Das gesuchte Objekt wird wiedererkannt, wenn die Anzahl der detektierten Punktepaare $\mathbf{p}_i^{n^*} \leftrightarrow \mathbf{p}_i^n$ größer als ein zuvor festgelegter Schwellwert t_{id} ist (vgl. Abschnitt 10.3).

12.2.2 Direkte Steuerung

Sobald der zu greifende Gegenstand wiedererkannt wurde, wird der Endeffektor mit Hilfe des Positionier-Systems in die vorgegebene Greiflage bezüglich des Gegenstandes gebracht. Im ersten Schritt der Positionierung wird die direkte Steuerung aus Kapitel 8 eingesetzt. Die direkte Steuerung erfordert, dass zunächst die aktuelle Lage cT_w des Gegenstands bezüglich der Kamera berechnet wird. Hierfür wird der Algorithmus 7.1 auf Basis der übereinstimmenden Bild- und Werkstückpunkte $p_i^n \leftrightarrow p_i^w$ genutzt.

Die Beziehung $p_i^n \leftrightarrow p_i^w$ ist dabei anhand der zuvor ausgeführten Korrespondenzanalyse und der vorab definierten Beziehung $p_i^{n*} \leftrightarrow p_i^w$ (vgl. Tabelle 12.1) gegeben. Die aktuelle Lage cT_w sowie die vorgegebene Greiflage ${}^{c*}T_w$ (sh. Tabelle 12.1) werden dazu genutzt, um mittels Gleichung 8.1 die Pose ${}^0T_{e^*}$ zu berechnen. Letztere stellt hierbei die einzustellende Greiflage des Endeffektors bezüglich des Umwelt-Koordinatensystems dar. ${}^0T_{e^*}$ wird mit Hilfe der inversen Kinematik aus Abschnitt 5.8.1 in Gelenkstellungen q konvertiert, um den Roboter zu positionieren. Der beschriebene Ablauf wird noch einmal in Abbildung 12.2 zusammengefasst.

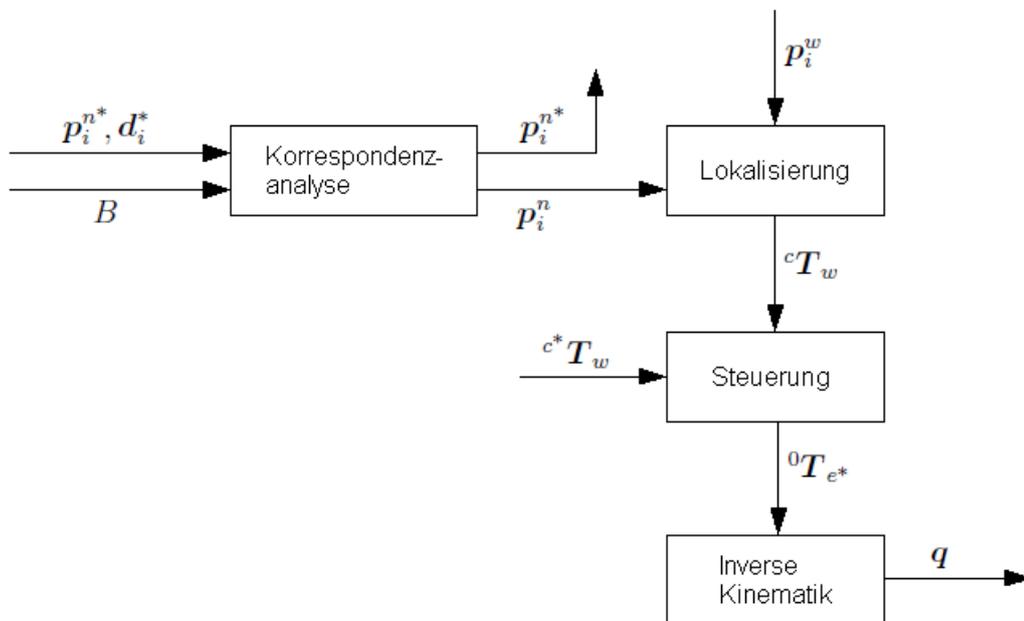


Abbildung 12.2: Direkte Steuerung

12.2.3 Bildpunkt-basierte Regelung

Aus den in Kapitel 2.1.1 erwähnten Gründen wird die durch die direkte Steuerung eingestellte Greiflage von der vorgegebenen Greiflage abweichen. Um eine genaue Einstellung der Greiflage zu erreichen, wird deshalb im Anschluss eine bildpunkt-basierte Regelung ausgeführt. Die Greiflage bezüglich des Objekts wird hierbei durch das vorab aufgenommene Zielbild B^* repräsentiert. Durch die Überführung des aktuellen Bildes B in das Zielbild B^* , befindet sich der Endeffektor anschließend in der vordefinierten Greiflage.

Zu Beginn eines jeden Regelschrittes muss zunächst eine Identifikation des zu greifenden Objekts in der aktuellen Ansicht B erfolgen. Dazu wird, wie im Fall der initialen Suche nach dem Objekt, eine Korrespondenzanalyse bezüglich der Ansichten B^* und B ausgeführt. Hierbei werden wieder statt B^* lediglich die bereits vorhandenen Referenz-Bildpunkte p_i^{n*} und

deren zugehörige SIFT-Merkmalvektoren \mathbf{d}_i^* verarbeitet. Anhand der Anzahl der detektierten Punktepaare $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ und des erwähnten Schwellwertes t_{id} wird abermals entschieden ob sich der zu greifende Gegenstand im aktuellen Kamerabild B befindet. Ist dies nicht der Fall, so muss der Regelvorgang abgebrochen werden.

Wenn der Gegenstand in B wiedererkannt wurde, dann wird der Regelfehler e (vgl. Gleichung 9.4) anhand der erfassten Punktepaare $\mathbf{p}_i^{n*} \leftrightarrow \mathbf{p}_i^n$ berechnet. Da die bildpunkt-basierte Regelung aus Kapitel 9 allgemein für Bildpunkte \mathbf{p}^s bezüglich des Sensor-Koordinatensystems $(KS)_s$ definiert wurde, erfolgt an dieser Stelle eine spezifische Definition bezüglich der hier verwendeten normierten Bildpunkte \mathbf{p}^n . Im weiteren Verlauf ist die Soll-Größe des Regelkreises für n Bildpunkte durch

$$\mathbf{s}^* = [(\mathbf{p}_1^{n*})^T \ \dots \ (\mathbf{p}_n^{n*})^T]^T \quad (12.2)$$

gegeben. Die Ist-Größe des Regelkreises ist dementsprechend anhand von

$$\mathbf{s} = [(\mathbf{p}_1^n)^T \ \dots \ (\mathbf{p}_n^n)^T]^T \quad (12.3)$$

definiert. Da die Bildweite f bei der Berechnung von normierten Bildpunkten \mathbf{p}^n den Wert 1 zugewiesen bekommt, vereinfacht sich die Bildjacobimatrix \mathbf{J}_i aus Formel 9.1 zu:

$$\dot{\mathbf{p}}^n = \underbrace{\begin{bmatrix} -\frac{1}{z_c} & 0 & \frac{x_n}{z_c} & x_n y_n & -x_n^2 + 1 & y_n \\ 0 & -\frac{1}{z_c} & \frac{y_n}{z_c} & y_n^2 + 1 & -x_n y_n & -x_n \end{bmatrix}}_{\mathbf{J}_i} \mathbf{v}_c. \quad (12.4)$$

\mathbf{J}_i enthält die Tiefeninformation z_c des Objektpunktes $\mathbf{p}^c = [x_c \ y_c \ z_c]^T$ bezüglich der Kamera, welcher auf den aktuellen Bildpunkt \mathbf{p}^n projiziert wird. Um z_c berechnen zu können, wird wieder die aktuelle Lage ${}^c\mathbf{T}_w$ des Gegenstands bezüglich der Kamera mittels der übereinstimmenden Bild- und Werkstückpunkte $\mathbf{p}_i^n \leftrightarrow \mathbf{p}_i^w$ berechnet. Die Zusammenhänge sind dabei äquivalent zu denen aus Kapitel 12.2.2. Die ermittelte Lage ${}^c\mathbf{T}_w$ ermöglicht den Wechsel des Bezugs-Koordinatensystems für die gegebenen Werkstückpunkte von \mathbf{p}_i^w nach \mathbf{p}_i^c (sh. Gleichung 5.7). Somit ist auch die erforderliche Tiefeninformation z_c bekannt.

Anhand des erfassten Regelfehlers e und der Bildjacobimatrix \mathbf{J}_i kann die Stellgröße des Regelkreises in Form der Kamerageschwindigkeit ${}^c\mathbf{v}_c$ berechnet werden. Dies geschieht mit Hilfe des Regelgesetzes aus Formel 9.5. Mittels der Kamerageschwindigkeit ${}^c\mathbf{v}_c$ kann eine Bewegung des Endeffektors in Richtung der durch B^* repräsentierten Greiflage ermöglicht werden.

Um dies zu erreichen kann ${}^c\mathbf{v}_c$ in Geschwindigkeiten der Gelenkwinkel $\dot{\mathbf{q}}$ des Roboters transformiert werden. Die Ansteuerung des Roboters über die Geschwindigkeiten $\dot{\mathbf{q}}$ würde bedeuten, dass sich der Roboter kontinuierlich bewegt. Um die Stellgröße rechtzeitig aktualisieren zu können, müssten die Berechnungen bezüglich der Bildverarbeitung sehr schnell erfolgen. Die Realisierung eines solchen Vorgehens wird auch als *Dynamic look-and-move system* bezeichnet (vgl. Abschnitt 2.1.2.1).

Da die Algorithmen in der vorliegenden Arbeit nicht auf Schnelligkeit ausgelegt sind, soll diese Vorgehensweise nicht umgesetzt werden. Stattdessen soll ein sogenanntes *Direct visual servo system* realisiert werden bei dem die Gelenke des Roboters direkt eingestellt werden. Weiterhin der intensiven Berechnungen soll die Auswertung eines Bildes nur dann erfolgen, wenn sich

der Roboter nicht bewegt. Diese Methode wird auch als *look-then-move* bezeichnet.

Zur Umsetzung dieses Vorgehens ist es zunächst notwendig das Bezugs-Koordinatensystem der Kamerageschwindigkeit von ${}^c\mathbf{v}_c$ nach ${}^0\mathbf{v}_c$ zu wechseln. Auf diese Weise ist die Geschwindigkeit der Kamera bezüglich des Umwelt-Koordinatensystems $(KS)_0$ gegeben. Hierfür wird Formel 5.11 verwendet, welche die Berücksichtigung des rotatorischen Anteils ${}^0\mathbf{R}_c$ der aktuellen Pose ${}^0\mathbf{T}_c$ der Kamera bezüglich $(KS)_0$ erfordert. Letztere kann über die aktuelle Endeffektor-Pose ${}^0\mathbf{T}_e$ (vgl. Abschnitt 5.6) und die vorab kalibrierte Kamerapose ${}^e\mathbf{T}_c$ relativ zum Endeffektor bestimmt werden:

$${}^0\mathbf{T}_c = {}^0\mathbf{T}_e {}^e\mathbf{T}_c. \quad (12.5)$$

${}^e\mathbf{T}_c$ stellt dabei das Ergebnis der Hand-Auge-Kalibrierung aus Kapitel 7.2 dar. Nach der Erfassung der Kamerageschwindigkeit ${}^0\mathbf{v}_c$ kann diese integriert werden, um eine Bewegung des Endeffektors in Richtung der durch B^* repräsentierten Greiflage zu erzeugen. Der Integrations-schritt wird dabei folgendermaßen durchgeführt.

Die Kamerageschwindigkeit ${}^0\mathbf{v}_c$ enthält laut Gleichung 5.10 die Winkelgeschwindigkeiten ${}^0\mathbf{w}_c = [w_x \ w_y \ w_z]^T$. Letztere werden anhand von Formel A.4 in die erste Ableitung ${}^0\dot{\mathbf{R}}_c$ der Rotationsmatrix transformiert, wobei wieder der rotatorische Anteil ${}^0\mathbf{R}_c$ der aktuellen Pose ${}^0\mathbf{T}_c$ der Kamera hinzugezogen werden muss. Kamerageschwindigkeit ${}^0\mathbf{v}_c$ enthält neben den Winkelgeschwindigkeiten ${}^0\mathbf{w}_c$ ebenfalls die translatorischen Geschwindigkeiten ${}^0\dot{\mathbf{o}}_c = [\dot{x} \ \dot{y} \ \dot{z}]^T$. Auf Basis von ${}^0\dot{\mathbf{o}}_c$ und der ersten Ableitung ${}^0\dot{\mathbf{R}}_c$ der Rotationsmatrix erfolgt eine Integration der Geschwindigkeiten durch

$$\begin{aligned} {}^0\mathbf{o}_c(k+1) &= {}^0\mathbf{o}_c(k) + {}^0\dot{\mathbf{o}}_c, \\ {}^0\mathbf{R}_c(k+1) &= {}^0\mathbf{R}_c(k) + {}^0\dot{\mathbf{R}}_c. \end{aligned} \quad (12.6)$$

Hierbei steht k für den aktuellen Regelschritt. Auf die hier erläuterte Art kann somit die Lage ${}^0\mathbf{T}_c(k)$ zu ${}^0\mathbf{T}_c(k+1)$ aktualisiert werden. Zur Aktualisierung der Gelenkwinkel \mathbf{q} des Roboters muss ${}^0\mathbf{T}_c(k+1)$ in die Lage ${}^0\mathbf{T}_e(k+1)$ des Endeffektors bezüglich $(KS)_0$ transformiert werden:

$${}^0\mathbf{T}_e(k+1) = {}^0\mathbf{T}_c(k+1) {}^c\mathbf{T}_e. \quad (12.7)$$

Die berechnete Lage ${}^0\mathbf{T}_e(k+1)$ wird anschließend mit Hilfe der inversen Kinematik aus Abschnitt 5.8.1 in Gelenkstellungen \mathbf{q} konvertiert, um den Roboter in Richtung der durch B^* repräsentierten Greiflage zu bewegen. Sobald die Regelabweichung gegen $\mathbf{0}$ konvergiert, stimmt B mit B^* überein. In diesem Fall befindet sich die Hand des Roboters in der vorgegebenen Greiflage relativ zum Gegenstand. Die bildpunkt-basierte Regelung wird noch einmal in Grafik 12.3 zusammengefasst.

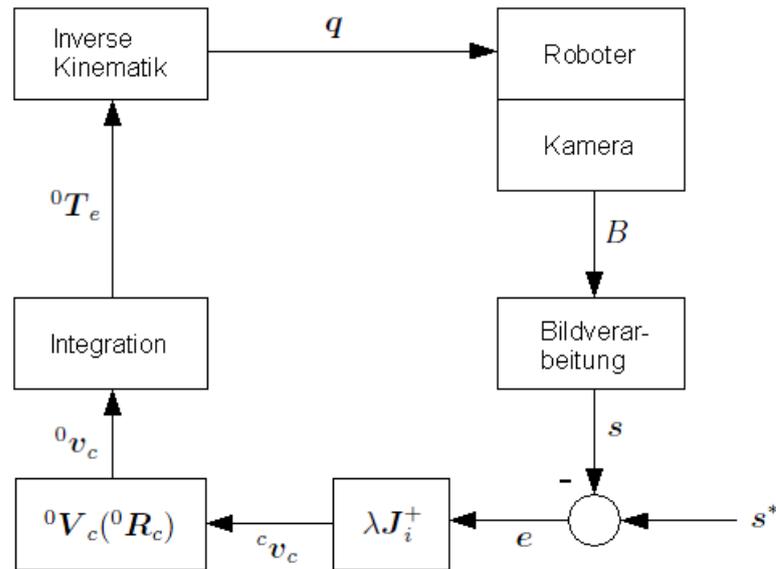


Abbildung 12.3: Bildpunktbasierte Regelung

12.2.4 Abschließende Greifbewegungen

Sobald sich der Endeffektor in der vorgegebenen Greiflage befindet, können die abschließenden Greifbewegungen ausgeführt werden. Hierbei werden zunächst vordefinierte Endeffektor-Posen sukzessive bezüglich der erreichten Greiflage eingestellt. Letztere sind anhand der Folgelagen $\{e^*T_{e1}, \dots, e^*T_{en}\}$ aus Tabelle 12.1 gegeben.

Um Beschädigungen am Roboter auf Grund von unvorhersehbaren Trajektorien zu vermeiden, können die Folgelagen schrittweise angefahren werden. Die Berechnung der ersten Endeffektor-Lage bezüglich $(KS)_0$ erfolgt durch:

$${}^0T_{e1} = {}^0T_{e^*} e^*T_{e1}. \quad (12.8)$$

${}^0T_{e^*}$ bezeichnet dabei die Endeffektor-Lage bezüglich $(KS)_0$ in der Zielsituation nach der bildpunktbasierten Regelung. Das Ergebnis ${}^0T_{e1}$ wird anschließend mit Hilfe der inversen Kinematik in Gelenkstellungen q transformiert. Alle weiteren Folgelagen werden nach dem gleichen Prinzip berechnet.

Nachdem die endgültige Lage des Endeffektors eingestellt wurde, muss die Hand des Roboters zum Greifen des Gegenstandes lediglich geschlossen werden. Im vorliegenden Fall bestimmt der vorab festgelegte Parameter $q_{gripper}$, wie weit sich die Hand schließen muss.

12.3 Realisierung

Das vorgestellte Greifsystem wurde mit Matlab implementiert, wobei die Bibliotheken aus Tabelle 12.2 genutzt wurden. Im Folgenden wird das Greifsystem simulativ sowie experimentell auf Basis des Katana-Roboters verifiziert.

Toolbox	Verwendungszweck
<i>Camera Calibration Toolbox</i> [BOU]	Kalibrierung der intrinsischen und extrinsischen Kameraparameter
<i>Camera Calibration and Hand to Eye Calibration Toolbox</i> [WEN]	Hand-Auge-Kalibrierung
<ul style="list-style-type: none"> • <i>VLFeat</i> [VF] • <i>MATLAB and Octave Functions for Computer Vision and Image Processing</i> [KOV] • <i>MATLAB Functions for Multiple View Geometry</i> [CFK⁺] 	Korrespondenzanalyse
<i>Fundamental Matrix Estimation toolbox</i> [AS03]	Schätzung der Fundamentalmatrix
<i>Structure and Motion Toolkit in Matlab</i> [TOR]	Zerlegung der Essentialmatrix und Triangulation
<i>Robotics Toolbox</i> [COR96]	Inverse Kinematik und Simulationsumgebung für den Roboter
<i>Epipolar Geometry Toolbox</i> [MP05]	Darstellung
<i>Katana Native Interface</i>	Ansteuerung des Katana-Roboterarms

Tabelle 12.2: Verwendete Bibliotheken

13 Simulation

Dieses Kapitel umfasst eine simulative Verifikation des Positionierungs-Verfahrens auf Basis des Katana-Roboters. Zunächst soll jedoch der unter Gliederungspunkt 2.1.2.2 erläuterte *camera retreat* anhand einer Simulation zur bildpunktbasierten Regelung untersucht werden. Auf diese Weise soll herausgefunden werden, ab welcher Drehung bezüglich der optischen Achse ein *camera retreat* auftritt.

13.1 Camera retreat

Da sich die Kamera im Fall eines *camera retreat* erst vom Ziel weg und später wieder darauf zubewegt, kann dies zum Abbruch der Regelung führen. Der Grund dafür ist, dass die geforderte Bewegung wegen der Gelenkgrenzen des Roboterarms unter Umständen nicht mehr ausgeführt werden kann. Je höher die Verdrehung um die optische Achse ist, umso stärker ist der beschriebene Effekt.

Die direkte Steuerung sorgt für eine grobe Positionierung des Endeffektors in Richtung der vorgegebenen Greiflage. Somit kann es vorkommen, dass die bildpunktbasierte Regelung im Anschluss eine Drehung um die optische Achse kompensieren muss. Es ist zu erwarten, dass die besagte Drehung nicht hoch sein wird. Auf Grund von Kalibrierfehlern und Bildrauschen kann die erforderliche Rotationsbewegung allerdings variieren. Aus diesem Grund wird nachstehend untersucht, ab welcher Drehung ein *camera retreat* auftritt. Somit besteht die Möglichkeit zur Abschätzung, wie genau die Positionierung mittels der direkten Steuerung sein muss, damit eine anschließende Regelung erfolgreich sein kann.

Zur Durchführung wird die bildpunktbasierte Regelung unabhängig von der direkten Steuerung untersucht. Weiterhin wird die Bewegung der Kamera direkt erzeugt, wodurch der Roboterarm in diesem Fall nicht benötigt wird. In Abbildung 13.1 wird zunächst die Simulationsumgebung vorgestellt. Die grün dargestellte Kamera gibt die Lage der aktuellen Kamera c an und bildet die vier Werkstückpunkte (blau) in die aktuelle Ansicht B ab, welche in Grafik 13.2 gezeigt wird.

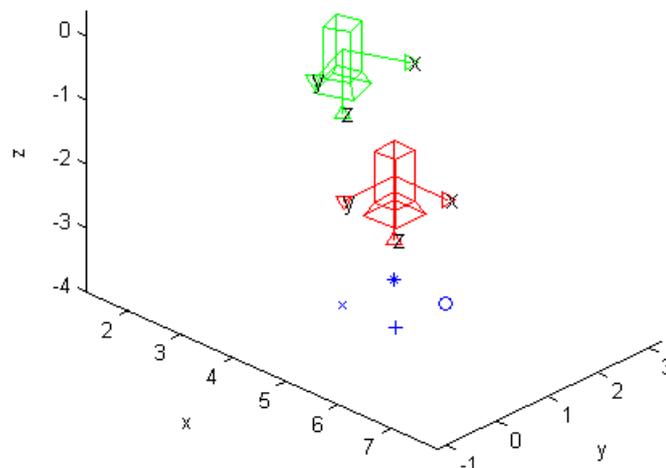


Abbildung 13.1: Simulationsumgebung

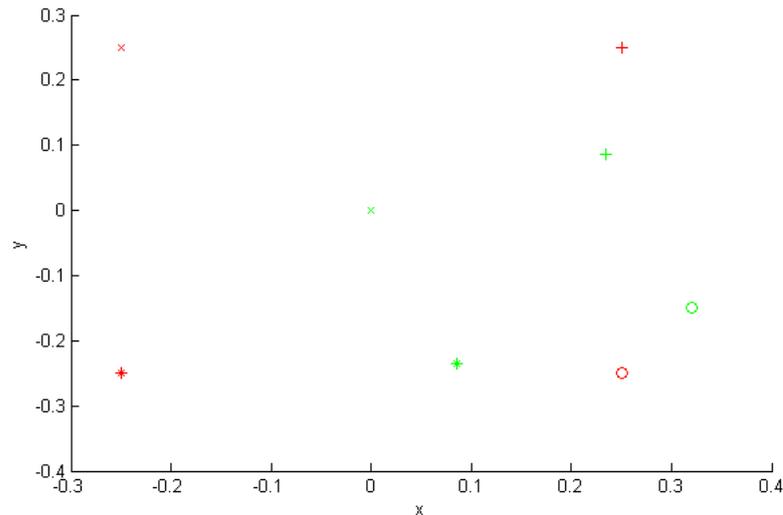


Abbildung 13.2: Überlagerte Bildpunkte bezüglich der Referenzansicht B^* (rot) und der aktuellen Ansicht B (grün)

Die rot dargestellte Kamera visualisiert die Lage der Ziel-Kamera c^* , welche die vier Werkstückpunkte in die Referenzansicht B^* abbildet. Die aktuelle Kamera c weist hierbei relativ zu c^* eine Verdrehung von 20 Grad um die optische Achse auf. Anhand der projizierten Bildpunkte ist der Regelfehler e gegeben, welcher anhand des Regelgesetzes auf eine Bewegung der Kamera c in Richtung c^* abgebildet wird. Der Verlauf des Regelfehlers e wird in Grafik 13.3 dargestellt, wobei hier der Betrag des Vektors e gezeigt wird. Dabei ist das typische exponentielle Abklingen des Regelfehlers zu sehen, welcher gegen 0 konvergiert.

Die entsprechende Bahn der Bildpunkte wird in Abbildung 13.4 visualisiert. Der lineare Verlauf der Bildpunkte kommt zu Stande, da in der Simulation alle Größen ideal und somit ohne Abweichungen vorliegen. Ein weiterer Grund für diesen Verlauf ist durch die Wahl der Regelverstärkung λ gegeben, welche im vorliegenden Fall 0,01 beträgt. Da es sich bei der im Arbeitspunkt berechneten Bildjacobimatrix J_i um ein linearisiertes Modell handelt, ist dessen Gültigkeit bezüglich der Kamerageschwindigkeit 0v_c nur auf eine kleine Umgebung beschränkt. Je höher der Wert für λ gewählt wird, umso größer ist die Ungenauigkeit bezüglich 0v_c . Dies würde unter Umständen zu einem Verlust der Bildpunkte aus dem Blickfeld der Kamera führen. Um dies zu vermeiden, wird für λ prinzipiell ein kleiner Wert gewählt. Dies hat allerdings auch zur Folge, dass die Regelung länger dauert.

In Grafik 13.5 wird die Kamerageschwindigkeit 0v_c für jeden Regelschritt dargestellt. Hierbei ist zu berücksichtigen, dass sich die Geschwindigkeiten w_x und w_y überlagern. Anhand des Verlaufs der translatorischen Geschwindigkeiten \dot{x} , \dot{y} und \dot{z} ist zu erkennen, dass die Kamera ihre Richtung nie geändert hat. In dem hier dargelegten Fall ist also kein *camera retreat* aufgetreten. Am Ende der Regelung koinzidiert die aktuelle Kamera c mit der Kamera c^* im Ziel.

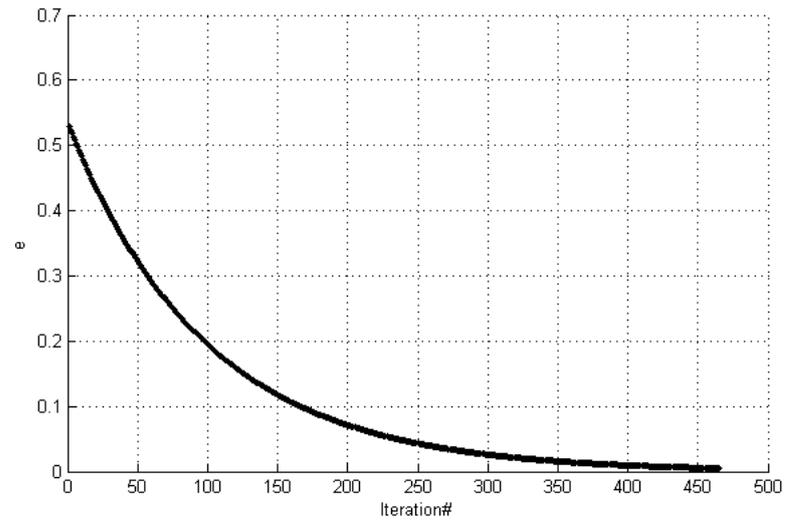
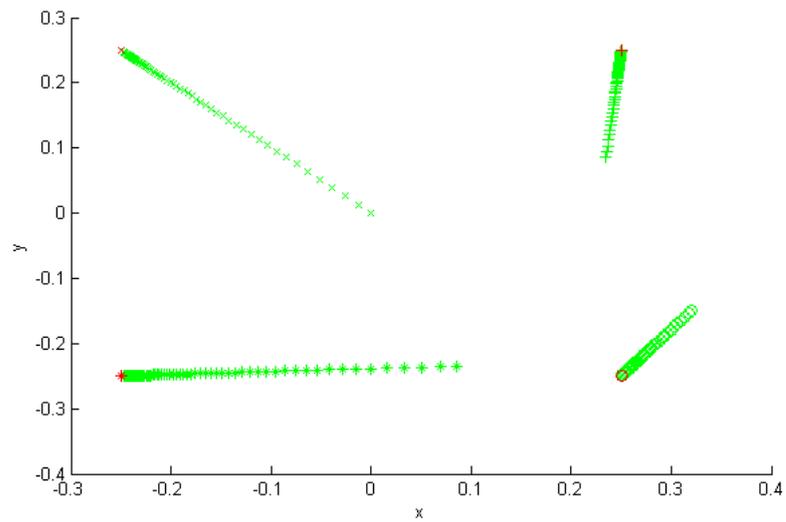
Abbildung 13.3: Regelabweichung e 

Abbildung 13.4: Verlauf der Bildpunkte

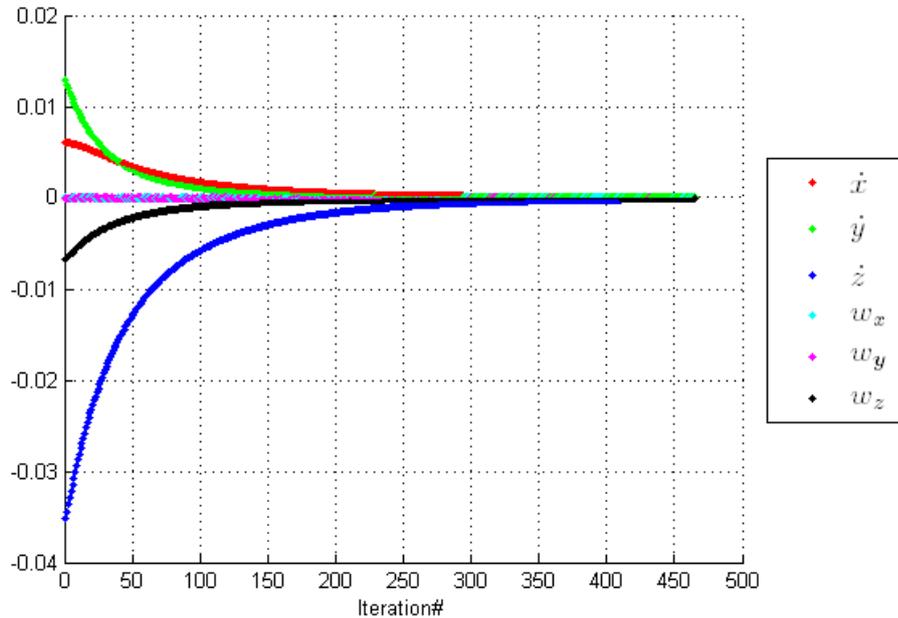


Abbildung 13.5: Geschwindigkeit 0v_c der Kamera bezüglich des Umwelt-Koordinatensystems

Nachstehend wird eine Konfiguration erläutert, bei der ein *camera retreat* in Erscheinung tritt. Die aktuelle Kamera c aus Abbildung 13.6 weist hierbei relativ zu c^* eine Verdrehung von 100 Grad um die optische Achse auf. Die Bahn der Bildpunkte wird in Abbildung 13.7 visualisiert, wobei hier die Kompensation der notwendigen Drehung um die optische Achse zu sehen ist.

Anhand von Grafik 13.8 ist zu sehen, dass sich die Kamera hinsichtlich der translatorischen Geschwindigkeiten \dot{x} , und \dot{z} zunächst vom Ziel weg bewegt hat. Anschließend wurde die Richtung für \dot{x} und \dot{z} gewechselt und die Kamera hat sich auf das Ziel zu bewegt. In dem geschilderten Fall ist es also zu einem *camera retreat* gekommen.

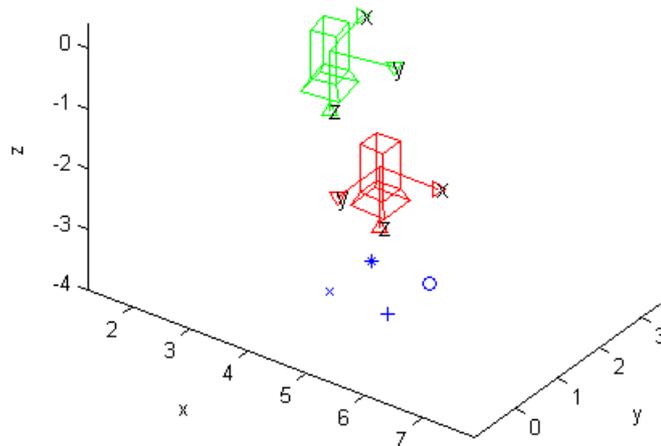


Abbildung 13.6: Lage der Kamera c bei einer Drehung von 100 Grad bezüglich der optischen Achse

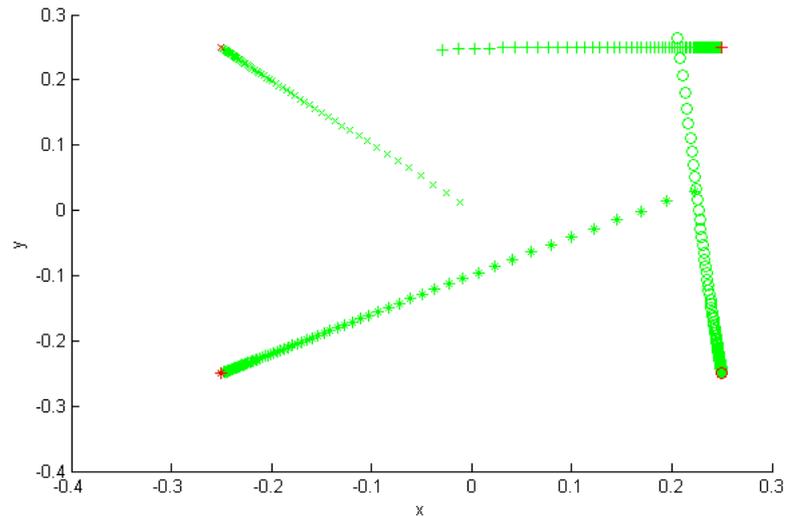
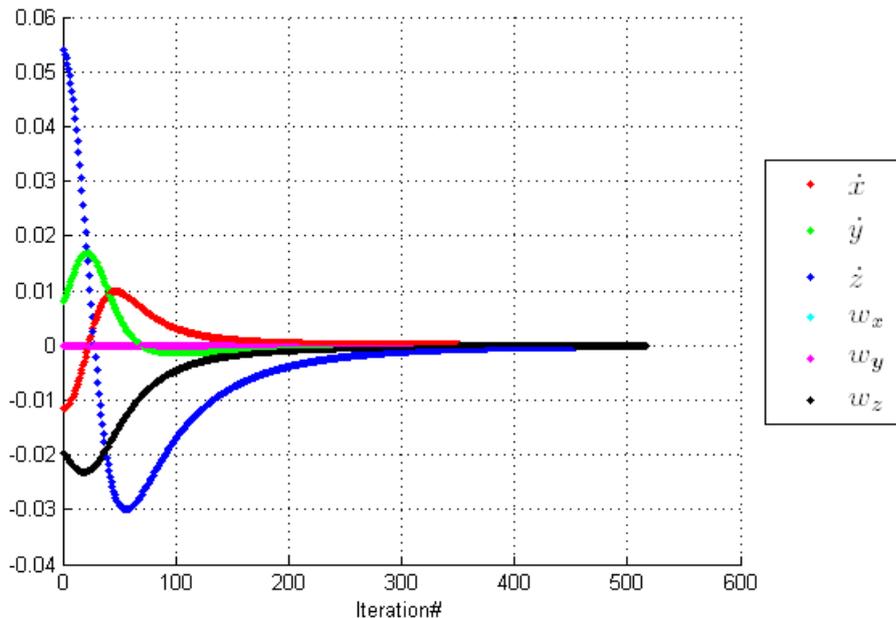


Abbildung 13.7: Verlauf der Bildpunkte

Abbildung 13.8: Kamera-Geschwindigkeit 0v_c

Zur Untersuchung des *camera retreat* wurden weitere Durchläufe mit unterschiedlichen Verdrehungen bezüglich der optischen Achse durchgeführt. Dabei hat sich herausgestellt, dass die beschriebene Problematik erst ab einer Drehung von 40 Grad auftritt. Hinsichtlich des Gesamtsystems können daraus folgende Schlüsse gezogen werden. Im Anschluss an die direkte Steuerung ist davon auszugehen, dass die Verdrehung bezüglich der optischen Achse deutlich kleiner als 40 Grad sein wird. Folglich wird während der Regelung kein *camera retreat* auftreten.

13.2 Positionierung des Endeffektors

Als nächstes wird eine simulative Verifikation des Positionierungs-Verfahrens auf Basis des Katana-Roboters durchgeführt. Es wird zunächst die direkte Steuerung zur groben Positionierung des Endeffektors genutzt. Anschließend erfolgt die bildpunkt-basierte Regelung zur

genauen Einstellung der gewünschten Endeffektor-Lage.

In Abbildung 13.9 wird die Simulationsumgebung gezeigt, welche im Vergleich zur vorigen Simulation um ein Modell des Katana-Greifarms erweitert wurde. Die Lage der aktuellen Kamera c bezüglich des Endeffektors ist dabei anhand der Ergebnisse der Hand-Auge-Kalibrierung aus Kapitel 7.2 definiert. Weiterhin wurde die Anzahl der Werkstückpunkte (blau) erhöht, welche in die Ansichten B^* und B projiziert werden. Die somit entstehenden Bildpunkte werden zusätzlich mit gaußverteilterm Rauschen versehen, um die Effekte beobachten zu können, welche im realen System durch fehlerbehaftete Größen entstehen.

Bei der in Grafik 13.9 gezeigten Start-Konfiguration besteht die in Tabelle 13.1 aufgeführte Lage ${}^{c^*}p_c$ der Kamera c bezüglich der Kamera c^* . Es handelt sich hierbei also um die Lage welche mittels der Positionierung kompensiert werden soll. Die Notation ${}^{c^*}p_c$ entspricht dabei der kompakten Darstellung einer Lage durch die RPY-Winkelkonvention. Die translatorischen Anteile werden hier in Millimeter und die Winkel in Radiant angegeben.

Auf Basis einer Rekonstruktion der Werkstückpunkte wurde ausgehend von der Start-Konfiguration zunächst die direkte Steuerung ausgeführt, welche zu dem in Abbildung 13.10 gezeigten Ergebnis führt. Anhand des Versatzes zwischen den Kameras c^* und c wird deutlich, dass nur eine grobe Positionierung des Endeffektors in Richtung der Ziellage erfolgte (vgl. Tabelle 13.1). Dies liegt daran, dass die Bildpunkte zuvor mit Abweichungen versehen wurden. Letztere wirken sich auf die Genauigkeit der rekonstruierten Raumpunkte und die Schätzung der Kameralagen bezüglich des Werkstück-Koordinatensystems aus.

Wenn die Bildpunkte hingegen ohne Abweichungen vorliegen, dann reicht die direkte Steuerung zur genauen Positionierung des Endeffektors aus. Dieser Umstand kann allerdings nur während der Simulation erzeugt werden. Im realen System sind Bildpunkte immer fehlerbehaftet und somit ist eine anschließende Regelung zur genauen Positionierung gerechtfertigt.

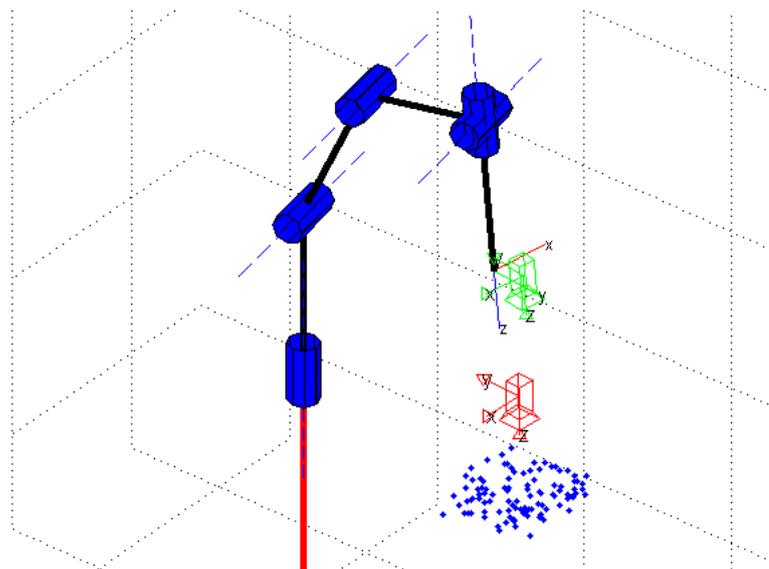
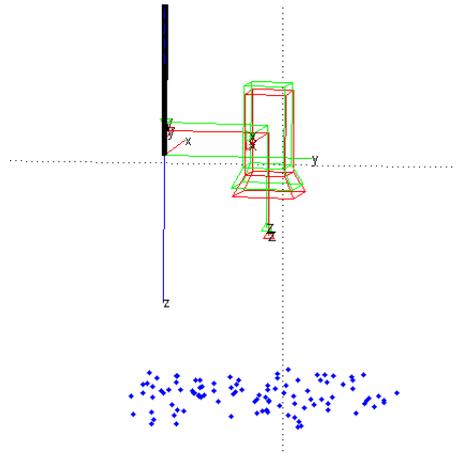


Abbildung 13.9: Simulationsumgebung

Situation	Kameralage ${}^{c^*}p_c$
Start	$[-30,725 \quad 22,379 \quad -129,669 \quad 0,081 \quad -0,035 \quad 0,17]^T$
Nach der direkten Steuerung	$[3,672 \quad -0,057 \quad -4,321 \quad -0,003 \quad 0,001 \quad -0,003]^T$
Nach der Regelung	$[0,003 \quad -0,037 \quad -0,119 \quad -0,003 \quad 0,001 \quad 0]^T$

Tabelle 13.1: Lage der Kamera c bezüglich der Kamera c^* Abbildung 13.10: Lage der Kamera c nach der direkten Steuerung

Der Versatz zwischen den Kameras aus Grafik 13.10 konnte mittels der bildpunktbasierter Regelung nahezu vollständig ausgeglichen werden (vgl. Abbildung 13.11). Der restliche Versatz zwischen den Kameras wird in Tabelle 13.1 angegeben. Der entsprechende Verlauf der Regelabweichung e und die Bahn der Bildpunkte werden in Abbildung 13.12 respektive 13.13 gezeigt, wobei hier die Auswirkungen des Bildrauschens anhand der Schwankungen zu sehen sind. Außerdem konvergiert e nicht mehr gegen 0. Die Regelung terminiert aus diesem Grund erfolgreich, sobald die Bedingung

$$\|e\|_1 < t \quad (13.1)$$

erfüllt wird. Die Regelung wird somit beendet, sobald die Betragssummennorm der Regelabweichung einen Schwellwert t unterschreitet. Letzterer muss experimentell ermittelt werden und sollte möglichst auf einen niedrigen Wert gesetzt werden. Der Verlauf der Bildpunkte wird hier lediglich exemplarisch für vier Bildpunkte angegeben, während die Regelung allerdings mit allen korrespondierenden Bildpunkten ausgeführt wird. Schwingungen in der Trajektorie der Bildpunkte können unter Umständen dazu führen, dass die verfolgten Punkte aus dem Blickfeld der Kamera geraten. Wenn die festgelegte Anzahl der benötigten Bildpunkte ($n \geq 4$) hierdurch unterschritten wird, dann muss die Regelung abgebrochen werden.

Im vorliegenden Fall wurde allerdings das Kriterium für die lokale asymptotische Stabilität aus Gleichung 9.8 in jedem Regelschritt erfüllt. Hieraus folgt, dass ein Verlust der Bildpunkte allgemein nur im Fall hoher Parameterunsicherheiten zu erwarten ist. Unabhängig von fehlerbehafteten Bildpunkten wird die Stabilität der Regelung im realen System jedoch durch weitere Störungen beeinflusst, welche nicht durch das Kriterium aus Formel 9.8 erfasst werden. Hierbei handelt es sich zum Beispiel um Kalibrierfehler bezüglich der Kinematik des Roboters sowie

der Hand-Auge-Kalibrierung. Wie sich die Abweichungen gemeinsam auf die Stabilität der Regelung auswirken, soll im Rahmen der experimentellen Untersuchungen ermittelt werden.

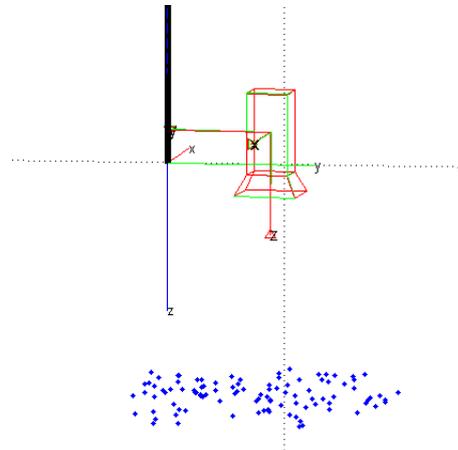


Abbildung 13.11: Lage der Kamera c nach der bildpunktbasierten Regelung

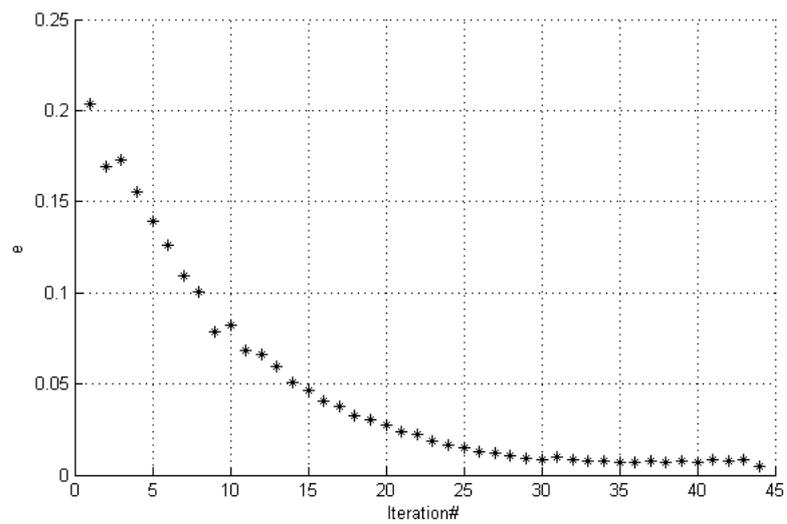


Abbildung 13.12: Regelabweichung e

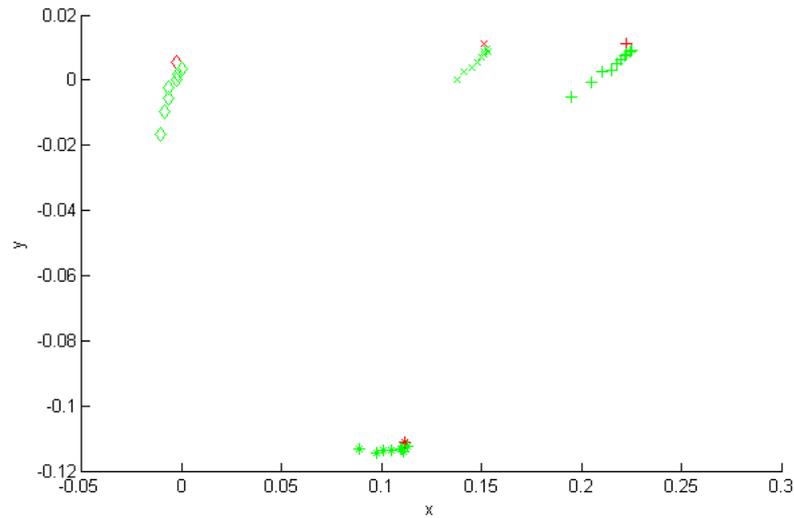
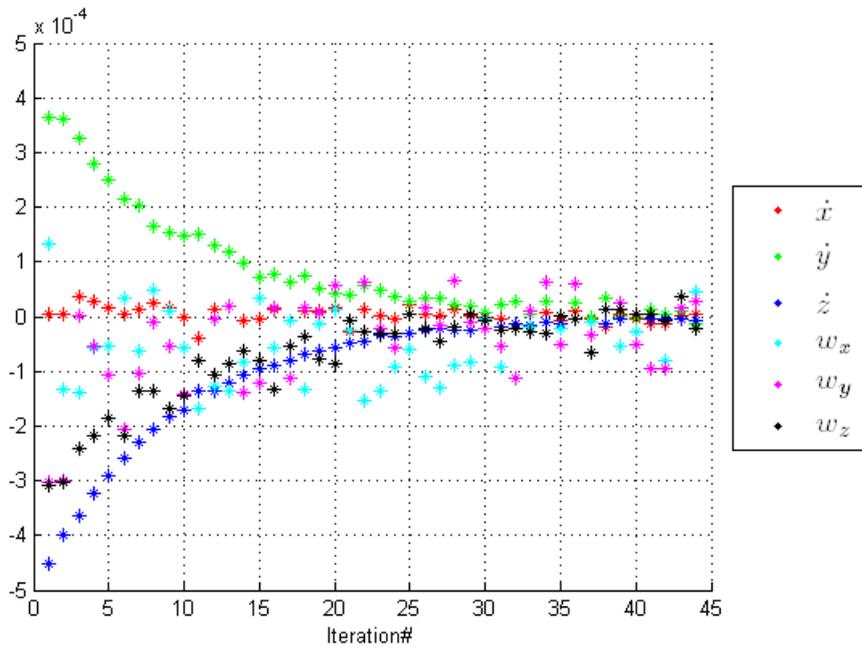


Abbildung 13.13: Verlauf der Bildpunkte

In Grafik 13.14 wird die Kamerageschwindigkeit 0v_c für jeden Regelschritt dargestellt. Hierbei sind natürlich ebenfalls die Auswirkungen des Bildrauschens anhand der Schwingungen zu sehen. Entsprechend zum Regelfehler e konvergieren die einzelnen Geschwindigkeiten ebenfalls nicht gegen 0, sondern schwanken lediglich um diesen Wert. Die hier vorgestellte Simulation wurde für weitere Konfigurationen durchgeführt, welche zu ähnlichen Ergebnissen führten.

Abbildung 13.14: Kamera-Geschwindigkeit 0v_c

14 Experimentelle Ergebnisse

Im Folgenden wird das Greifsystem experimentell auf Basis des Katana-Roboters verifiziert. Weiterhin wird erläutert, welche Einschränkungen bei der Ausführung von Greifvorgängen bezüglich des Katana-Greifarms vorhanden sind.

14.1 Gesamt-System

Nachstehend werden die Ergebnisse für einen vollständigen Greifvorgang erläutert. Bei dem zu greifenden Gegenstand handelt es sich um die Schachtel mit der Aufschrift ACC aus Grafik 14.1. In der gezeigten Situation wurde die Suche nach dem Gegenstand bereits erfolgreich abgeschlossen, indem der Gegenstand im aktuellen Kamerabild B wiedererkannt wurde. Letzteres wird anhand der detektierten Punktepaare deutlich, welche in Abbildung 14.2 dargestellt werden.



Abbildung 14.1: Katana-Greifarm nach der Identifikation der Schachtel im aktuellen Kamerabild



Abbildung 14.2: Punktpaare bezüglich der überlagerten Ansichten B^* und B nach der Wiedererkennung

Mittels der direkten Steuerung wurde eine Positionierung des Endeffektors in Richtung der vorab spezifizierten Greiflage vorgenommen. Der Versatz der anschließend erfassten korrespondierenden Bildpunkte (vgl. Abbildung 14.3) zeigt, dass es sich hierbei lediglich um eine grobe Positionierung gehandelt hat. Ein Grund hierfür ist durch das Bildrauschen und die Abweichungen bezüglich der rekonstruierten Raumpunkte gegeben, welche sich auf die Genauigkeit der geschätzten Lage des Gegenstands relativ zur Kamera auswirken. Eine weitere Ursache ist anhand der Ungenauigkeiten hinsichtlich der kalibrierten Lage zwischen der Kamera und dem Endeffektor gegeben.

Außerdem wird die Genauigkeit der direkten Steuerung durch Abweichungen bezüglich der Roboter-Kinematik beeinflusst. Es hat sich herausgestellt, dass die mittels Gleichung 5.14 berechnete Pose ${}^0T_e(\mathbf{q})$ des Endeffektors bezüglich des Umwelt-Koordinatensystems von der wahrhaftigen Pose geringfügig abweicht. Daraus lässt sich schließen, dass die DH-Parameter aus Tabelle 5.2 lediglich eine Näherung darstellen.

In Anlehnung an die Ergebnisse zur Simulation des *camera retreat* aus Abschnitt 13.1, hat sich folgendes ergeben. Die Verdrehung um die optische Achse war nach der direkten Steuerung im Rahmen aller ausgeführten Testdurchläufe deutlich niedriger als 40 Grad. Ein *camera retreat* ist hier somit nie aufgetreten.

Um eine genaue Einstellung der Greiflage zu erreichen wurde die bildpunkt-basierte Regelung im Anschluss an die direkte Steuerung ausgeführt. Dadurch konnte der Restfehler bezüglich der detektierten Punktpaare nahezu vollständig ausgeglichen werden (sh. Grafik 14.4).

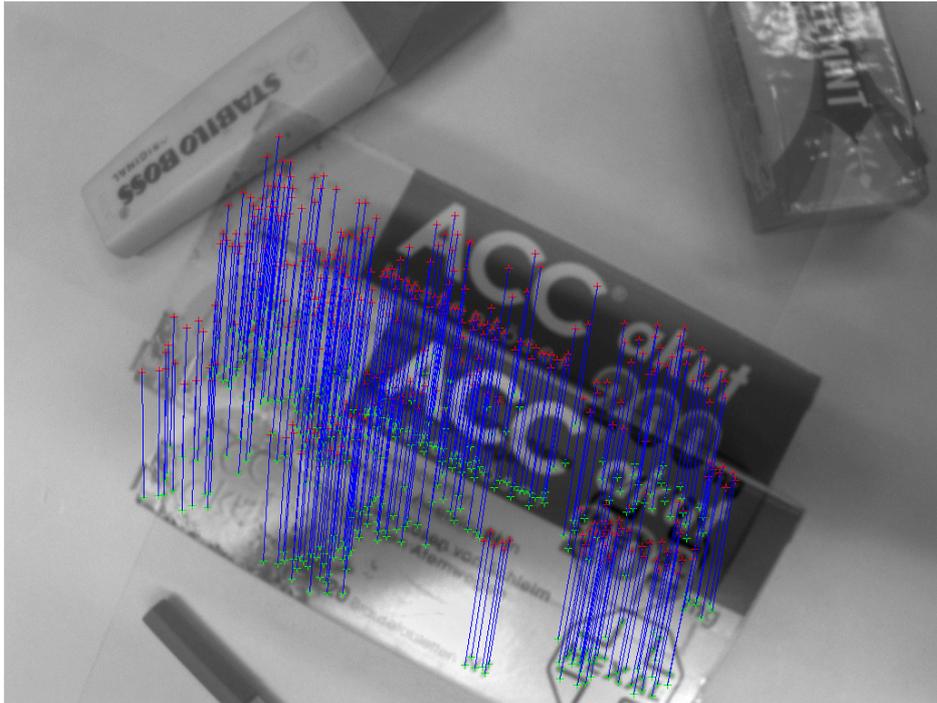


Abbildung 14.3: Punktpaare bezüglich der überlagerten Ansichten B^* und B nach der direkten Steuerung



Abbildung 14.4: Punktpaare bezüglich der überlagerten Ansichten B^* und B nach der Regelung

Der entsprechende Verlauf der Regelabweichung e und der Bildpunkte wird in Abbildung 14.5 respektive 14.6 gezeigt. Der Einfluss des Bildrauschens und der Kalibrierfehler wird besonders anhand der starken Schwingungen in der Bildpunkt-Bahn deutlich. Es sei angemerkt, dass der Verlauf der Bildpunkte hier lediglich exemplarisch für vier Bildpunkte angegeben wird,

während die Regelung mit allen detektierten Punktepaaren ausgeführt wurde. Weiterhin lässt sich erkennen, dass hin und wieder Lücken in der Trajektorie der Bildpunkte auftreten. Dies liegt daran, dass manche der Bildpunkte nicht in jedem Regelschritt verfolgt werden konnten.

Schwingungen im Verlauf der Bildpunkte können unter Umständen dazu führen, dass die verfolgten Punkte aus dem Blickfeld der Kamera geraten. Im Rahmen aller ausgeführten Testdurchläufe wurde jedoch das Kriterium für die lokale asymptotische Stabilität aus Gleichung 9.8 stets erfüllt. Dieses Kriterium erfasst dabei nur Abweichungen die hinsichtlich der Bildpunkte vorliegen. Es hat sich allerdings gezeigt, dass auch die zusätzlichen Störungen in Form der Kalibrierfehler nicht zu einem Verlust der verfolgten Bildpunkte führen. Im Fall von erhöhten Parameterunsicherheiten kann diese Gefahr hingegen nicht ausgeschlossen werden.

Abbildung 14.7 zeigt die Kamerageschwindigkeit 0v_c für jeden Regelschritt. Die Schwingungen sind hierbei deutlich geringer als im Vergleich zum Verlauf der Bildpunkte aus Abbildung 14.6. Dies liegt daran, dass bei der Berechnung von 0v_c hauptsächlich die Abweichungen bezüglich der Bildpunkte einfließen.

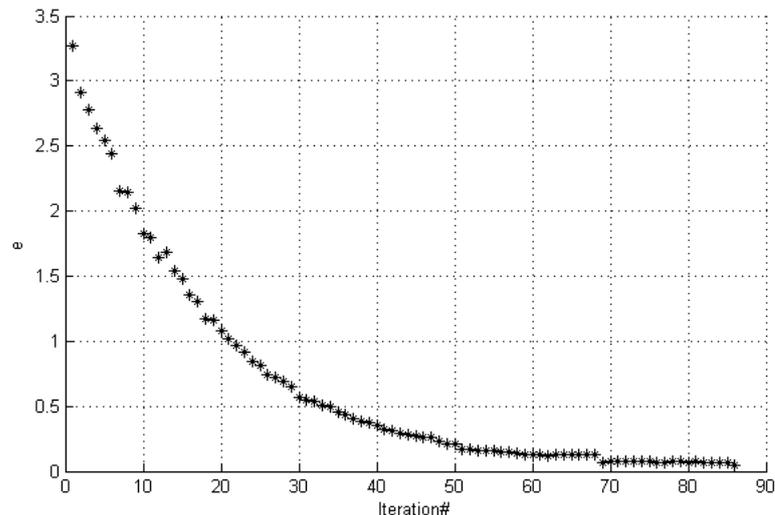


Abbildung 14.5: Regelabweichung e

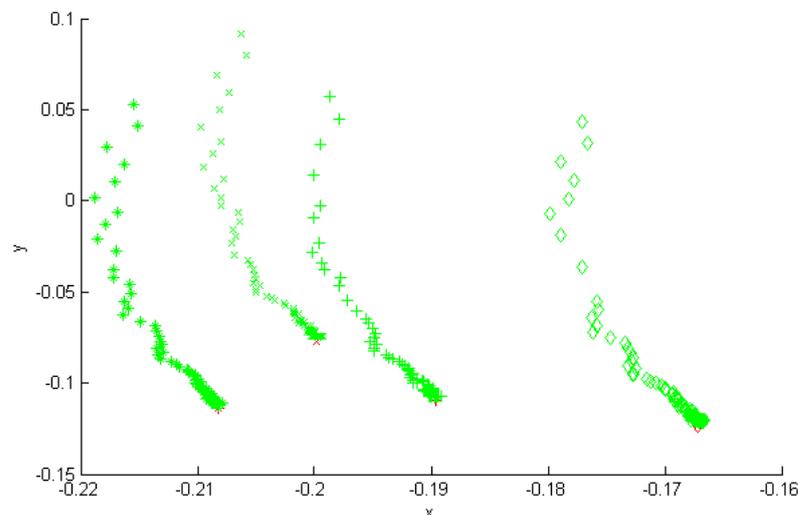


Abbildung 14.6: Verlauf der Bildpunkte

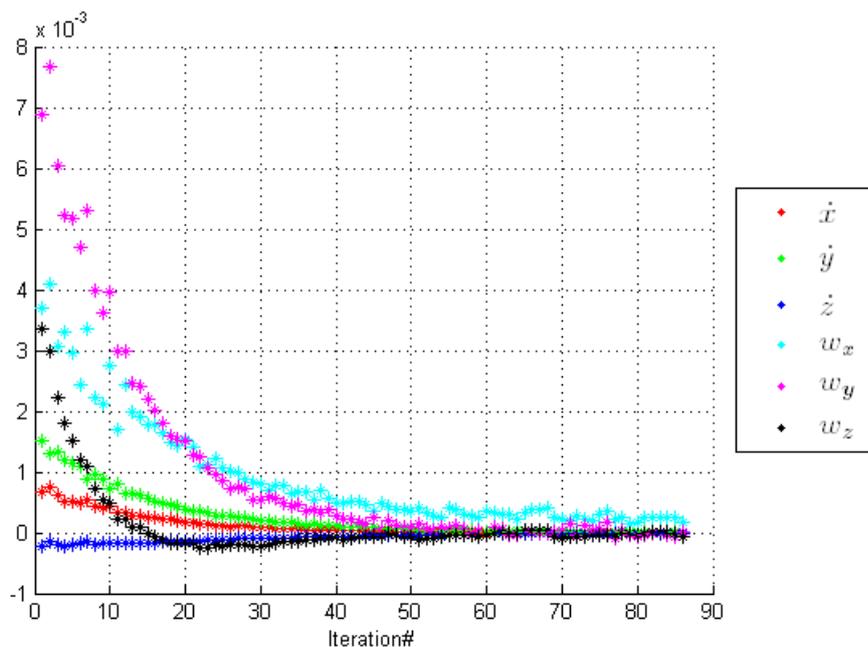


Abbildung 14.7: Geschwindigkeit 0v_c der Kamera bezüglich des Umwelt-Koordinatensystems

In Grafik 14.8 wird der Katana-Greifarm nach der Terminierung der Regelung gezeigt. Es ist zu erkennen, dass zum Greifen der Schachtel mit der Aufschrift ACC weitere Bewegungen notwendig sind. Dafür wurden zunächst zwei vorab definierte Lagen des Endeffektors bezüglich der eingestellten Greiflage sukzessive angefahren. Anschließend musste die Hand des Roboters zum Greifen des Gegenstandes lediglich geschlossen werden (vgl. Abbildung 14.9).

Der hier gezeigte Ablauf wurde für verschiedene Konfigurationen wiederholt, bei denen sich der zu greifende Gegenstand immer an einer anderen Stelle im Arbeitsraum des Roboters befand. Ebenso wurden in den Testdurchläufen unterschiedliche Objekte erfolgreich gegriffen. Hinsichtlich der Ausführung von Greifvorgängen auf der Basis des Katana-Roboters haben sich einige Einschränkungen ergeben, welche im Folgenden beschrieben werden.



Abbildung 14.8: Katana-Greifarm nach der Regelung



Abbildung 14.9: Katana-Greifarm nach dem Greifen der Schachtel

14.2 Einschränkungen bezüglich des Katana-Roboters

Die Gelenkstruktur des Katana-Greifarms bringt es mit sich, dass Greifvorgänge nur eingeschränkt durchgeführt werden können (vgl. Abschnitt 4.1.3). Aus diesem Grund wurden die zu greifenden Gegenstände in den Test-Situationen so hingestellt, dass sie für den Katana-Manipulator erreichbar waren. Trotzdem konnte der Greifvorgang in manchen Fällen nicht erfolgreich ausgeführt werden. Der Grund hierfür wird nachfolgend anhand eines Beispiels erläutert.

In Abbildung 14.10 werden die detektierten korrespondierenden Bildpunkte bezüglich eines Gegenstandes während der bildpunktbasierten Regelung gezeigt. Weiterhin wird der Verlauf der Kamerageschwindigkeit ${}^0\mathbf{v}_c$ in Grafik 14.11 dargestellt. Hierbei ist zu erkennen, dass sich die einzelnen Geschwindigkeiten von Beginn der Regelung an kaum verändert haben. Dementsprechend hat sich der Roboter während der Regelung nur geringfügig bewegt und somit hat sich auch der Versatz zwischen den Punktepaaren aus Abbildung 14.10 kaum geändert. Folglich liegt hier eine Situation vor, in der die Regelung nicht durchgeführt werden kann.

Die Ursache hierfür ist die inverse Kinematik aus Kapitel 5.8.1. Um das Minimierungsproblem aus Gleichung 5.24 lösen zu können, muss eine der sechs Komponenten des Geschwindigkeitsvektors ${}^0\mathbf{v}_e(\mathbf{q})$ auf den Wert 0 gesetzt werden. Dies ist notwendig, da der Katana-Greifarm lediglich fünf Gelenke besitzt. Im vorliegenden Fall wurde hierfür die Komponente w_x gewählt, welche äquivalent zu der Größe w_x aus Grafik 14.11 ist. Dementsprechend wird während der gesamten Regelung keine nennenswerte Änderung bezüglich der aktuellen Gelenkwinkel erzeugt.

In dem beschriebenen Fall hätte die Regelung erfolgreich durchgeführt werden können, wenn zum Beispiel für die Komponente w_y der Wert 0 angenommen worden wäre. Allerdings kann dieses Vorgehen in einer anderen Situation, analog zum dargelegten Fall, ebenfalls zum Scheitern der Regelung führen.

Eine der Komponenten des Vektors ${}^0\mathbf{v}_e(\mathbf{q})$ auf den Wert 0 zu setzen, bedeutet dabei allerdings nicht, dass die Regelung zwangsläufig fehlschlägt, sobald die manipulierte Komponente nicht von Beginn an den Wert 0 aufweist. Dies wird anhand von Grafik 14.7 verdeutlicht, wobei hier alle Geschwindigkeiten von Beginn an einen von 0 verschiedenen Wert besitzen. Obwohl für w_x der Wert 0 angenommen wird, kann die Regelung trotzdem erfolgreich durchgeführt werden. Dies liegt daran, dass die übrigen Geschwindigkeiten ausreichen, um den Regelfehler zu minimieren.

Ein Scheitern der Regelung kann demzufolge nur dann zu Stande kommen, sobald nur noch die manipulierte Geschwindigkeits-Komponente für eine Änderung sorgen kann. Als Konsequenz hieraus folgt, dass die verwendete inverse Kinematik nur für Roboter geeignet ist, welche sechs oder mehr FHG bezüglich des Gelenkraums besitzen. Um das entwickelte Greif-Verfahren uneingeschränkt für den Katana-Roboter nutzen zu können, müsste hierfür eine andere inverse Kinematik genutzt werden.

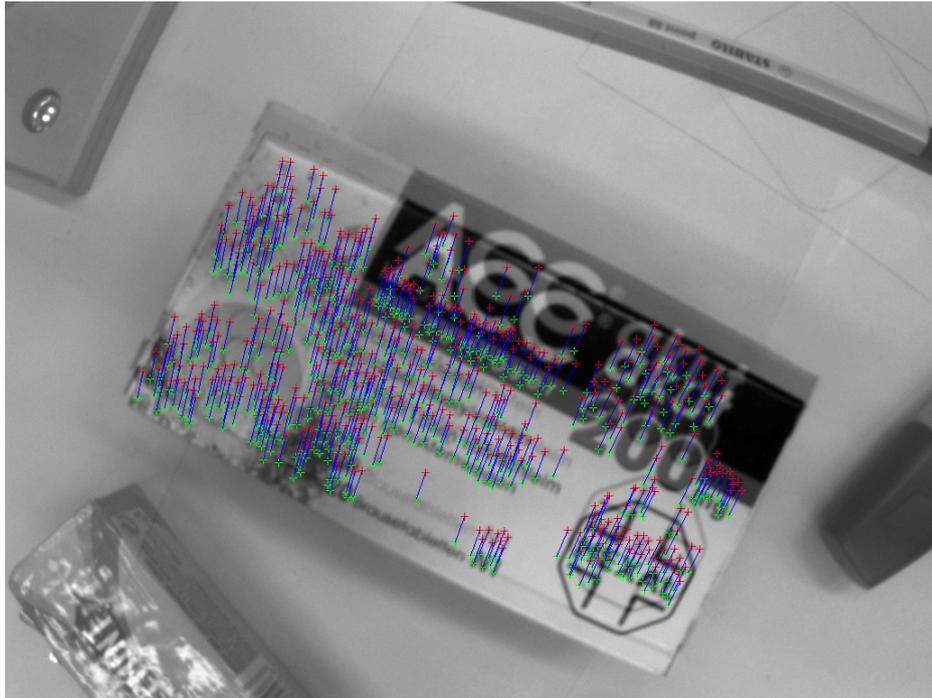


Abbildung 14.10: Punktepaare bezüglich der überlagerten Ansichten B^* und B während der Regelung

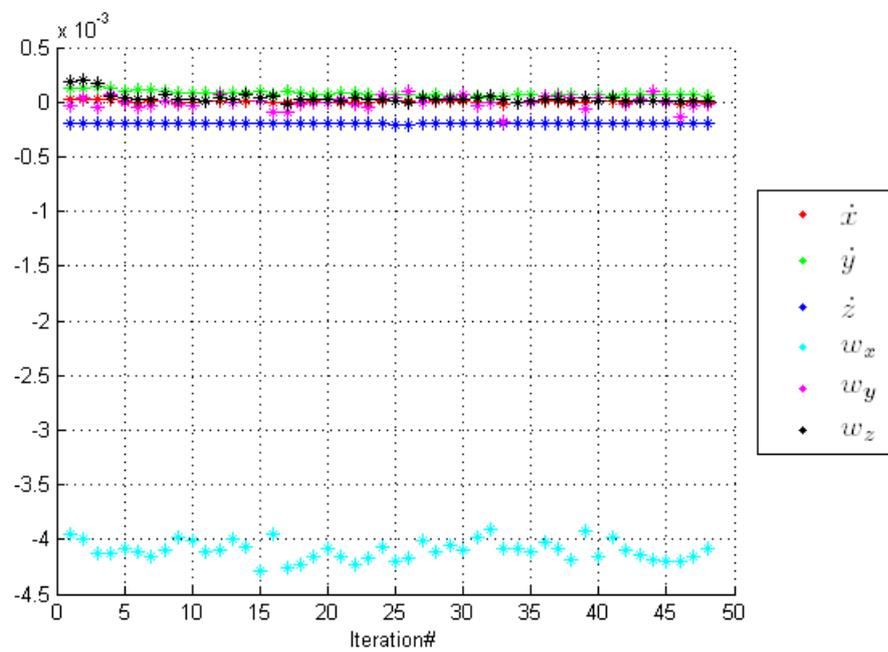


Abbildung 14.11: Kamera-Geschwindigkeit 0v_c

15 Zusammenfassung und Ausblick

In dieser Masterarbeit wurde ein generisches Verfahren zum kamerabasierten Greifen ungeordneter Gegenstände für stationäre Roboterarme entwickelt. Das Verfahren wurde simulativ sowie experimentell auf Basis des Katana-Roboterarms vom Typ 450-6M180 verifiziert. Die Resultate dieser Arbeit sollen möglicherweise zukünftig für einen mobilen Greifarm im Kontext von Assistenzaufgaben im Haushalt verwertet werden.

Im Rahmen der Untersuchung von vergleichbaren Arbeiten wurde ein Greifvorgang zunächst in die folgenden zwei Schritte unterteilt. Zu Beginn wird eine kamerabasierte Positionierung des Endeffektors in eine Greiflage bezüglich eines Gegenstands ausgeführt. Daraufhin erfolgen abschließende Greifbewegungen. Zuerst wurden verschiedene Vorgehensweisen für den Positionierungs-Schritt erläutert. Hierbei wurden zwei Ansätze verfolgt, wobei es sich um eine direkte Steuerung (*open-loop approach*) und ein regelbasiertes Vorgehen (*closed-loop approach*) handelt. Letzteres wird auch als *Visual Servoing* (VS) bezeichnet.

Es hat sich gezeigt, dass die direkte Steuerung eine ungenaue Methode zur Positionierung darstellt. Im Kontrast hierzu ermöglicht das VS auf Grund des regelbasierten Vorgehens eine genauere aber auch langsamere Positionierung des Endeffektors. Weiterhin hat sich ergeben, dass die meisten der vorgestellten VS-Verfahren lediglich unter bestimmten Voraussetzungen einwandfrei funktionieren. Um den Restriktionen der einzelnen Verfahrensweisen entgegenwirken zu können, wurde in der vorliegenden Arbeit ein hybrides Positionierungs-System entwickelt. Hierbei kommt es zu einer Kopplung der direkten Steuerung mit einer anschließenden bildpunkt-basierten Regelung, welche robust gegenüber Kalibrierungsfehlern und Bildrauschen ist.

Dieses Vorgehen bietet insgesamt eine genaue Einstellung der Greiflage. Weiterhin wird so eine geringere Prozesszeit als im Fall einer rein regelbasierten Methode ermöglicht. Die Greiflage wird im Fall bildpunkt-basierten Regelung durch eine Referenzansicht bezüglich des zu greifenden Gegenstands repräsentiert, welche vorab in einem *Teach-In* Schritt erfasst wird.

Der zweite Schritt eines Greifvorgangs besteht aus abschließenden Greifbewegungen, welche in dieser Arbeit anhand einer direkten Steuerung umgesetzt wurden. Da sich ein Gegenstand während der gesamten Zeit der Regelung im Blickfeld der Kamera befinden muss, kann es vorkommen, dass die Hand des Roboters den Gegenstand nach der Regelung noch nicht greifen kann. In diesem Fall werden sukzessive Folgelagen des Endeffektors relativ zur Greiflage eingestellt, wobei letztere durch die Regelung umgesetzt wird.

Diese zusätzlichen Lageinformationen werden, in Anlehnung an die Aufnahme der Referenzansicht, ebenfalls *offline* anhand eines *Teach-In* Vorgangs bestimmt. Nach der Einstellung der Folgelagen muss die Hand des Roboters zum Greifen des Objekts lediglich geschlossen werden. Wie weit sich die Hand schließen muss wird hier ebenfalls vorab erfasst.

Zur Realisierung des Greif-Verfahrens wurden die einzelnen Systemkomponenten identifiziert und schrittweise modelliert. Hierbei handelt es sich zunächst um die kinematischen Zusammenhänge zwischen den Gelenkwinkeln des Roboterarms und der Pose des Endeffektors bezüglich eines Umwelt-Koordinatensystems. Weiterhin wurde das Projektionsverhalten der verwendeten Kamera modelliert, um eine Kalibrierung der intrinsischen und extrinsischen Kameraparameter

zu ermöglichen.

Die intrinsischen Kameraparameter ermöglichen dabei die Korrektur der Linsenverzeichnung bezüglich der Bildpunkte. In der vorliegenden Arbeit wird die Verfolgung von Bildpunkten in perspektivisch unterschiedlichen Bildern benötigt. Dieses Vorgehen wird auch als Korrespondenzanalyse bezeichnet und wurde im ersten Schritt mittels des Verfahrens *Scale Invariant Feature Transform* (SIFT) sowie anhand des *Nearest-Neighbor*-Algorithmus realisiert. Diese Vorgehensweise führt dazu, dass lediglich vermeintlich übereinstimmende Bildpunkte gefunden werden. Um die wahren Punktepaare zu ermitteln wird eine robuste Schätzung der sogenannten Homography-Matrix mittels des *Random Sample Consensus* (RANSAC)-Algorithmus auf Basis der vermeintlich korrespondierenden Bildpunkte vorgenommen.

Die Detektion von übereinstimmenden Bildpunkten ermöglicht die Wiedererkennung eines zu greifenden Gegenstands, welcher im System anhand der vorab aufgenommenen Referenzansicht bekannt ist. Korrespondierende Bildpunkte bieten außerdem die Möglichkeit zur Rekonstruktion von Raumpunkten auf der Grundlage der Schätzung der Fundamentalmatrix. Dies dient in der vorliegenden Arbeit zur vorab durchgeführten Generierung eines partiellen Modells von einem Gegenstand, der gegriffen werden soll. Anhand einer solchen Teilrekonstruktion wird die Erfassung der Pose eines Gegenstands bezüglich der Kamera vorgenommen.

Letzteres ermöglicht die Ausführung der direkten Steuerung, welche den ersten Schritt des hybriden Positionierungs-Systems definiert. Auf Grundlage der Detektion von korrespondierenden Bildpunkten wird anschließend die bildpunkt-basierte Regelung zur genauen Einstellung der Greiflage ausgeführt.

Im Rahmen einer simulativen Verifikation wurde zunächst die bildpunkt-basierte Regelung analysiert, um zu ermitteln ab welcher Drehung bezüglich der optischen Achse ein *camera retreat* auftritt. Da sich die Kamera im Fall eines *camera retreat* erst vom Ziel weg und später wieder darauf zubewegt, kann dies auf Grund der Gelenkgrenzen des Roboters zum Abbruch der Regelung führen. Es hat sich herausgestellt, dass die beschriebene Problematik in der Simulation erst ab einer Drehung von 40 Grad um die optische Achse auftritt. Experimentelle Ergebnisse bezüglich des Gesamt-Systems haben gezeigt, dass die Drehung um die optische Achse nach der direkten Steuerung deutlich niedriger als 40 Grad ist. Ein *camera retreat* kann folglich ausgeschlossen werden.

Durch Bildrauschen und Kalibrierfehler kann es während der bildpunkt-basierten Regelung vorkommen, dass ein zu greifender Gegenstand aus dem Blickfeld der Kamera gerät. Dies würde zum Abbruch der Regelung führen. Es hat sich jedoch herausgestellt, dass die Regelung während der gesamten Testdurchläufe trotz vorhandener Störeinflüsse stabil war. Greifvorgänge konnten für unterschiedliche Objekte erfolgreich durchgeführt werden.

Hierbei haben sich jedoch Einschränkungen bezüglich der Ausführung auf der Basis des Katana-Roboters geäußert. Manche Greifvorgänge konnten nicht erfolgreich durchgeführt werden, da die verwendete inverse Kinematik nicht vollständig für den Katana-Roboter geeignet ist. Um das entwickelte Greif-Verfahren uneingeschränkt für den Katana-Roboter nutzen zu können, müsste eine andere inverse Kinematik eingesetzt werden.

In dieser Masterarbeit wurde eine robuste Lösung zur Realisierung von Greifvorgängen an-

gestrebt, um zunächst zu einer funktionierenden Lösung zu gelangen. Die Geschwindigkeit während der Ausführung lag hier hingegen nicht im Vordergrund und soll jedoch in weiterführenden Arbeiten berücksichtigt werden. Eine Verkürzung der Prozesszeit kann zum Beispiel durch die nachstehenden Optimierungen erreicht werden.

Im Rahmen der Korrespondenzanalyse könnte das Verfahren *Speeded-Up Robust Features* (SURF) eingesetzt werden, welches im Vergleich zu SIFT wesentlich weniger Rechenzeit benötigt. Gleichzeitig werden auf diese Weise jedoch deutlich weniger korrekt übereinstimmende Bildpunkte detektiert. Außerdem kann hierbei auch der *Nearest-Neighbor*-Algorithmus ersetzt werden, welcher die trivialste und gleichzeitig aufwendigste Möglichkeit zur Erfassung von vermeintlich übereinstimmenden Bildpunkten darstellt.

Während der bildpunktbasierter Regelung kann der optische Fluss zur Detektion von korrespondierenden Bildpunkten berechnet werden. Dies ist möglich, da während der Regelung nur geringe Positionsänderungen stattfinden. Somit könnte die zeitaufwendigere Erfassung von markanten Bildpunkten ab dem zweiten Regelschritt vollständig eingespart werden.

Die robuste Schätzung der Lage des Werkstücks bezüglich der Kamera wird momentan auf Basis einer Methode durchgeführt, welche dem RANSAC-Algorithmus ähnelt. Hierbei wird in jedem Iterationsschritt eine nicht-lineare Optimierung durchgeführt, welche unter Umständen durch eine linearisierende Vorgehensweise substituiert werden könnte.

Ausblickend kann zusätzlich die Robustheit des Greif-Systems erhöht werden. Dies betrifft zunächst die Wiedererkennung von Objekten, welche in der vorliegenden Arbeit mit Hilfe von SIFT durchgeführt wird. Es hat sich gezeigt, dass die Möglichkeit zur Wiedererkennung direkt von der Perspektive abhängt. Wenn sich die Perspektive gegenüber der vorab gespeicherten Referenzansicht stark ändert, dann ist eine Identifikation eines Gegenstandes im aktuellen Kamerabild nicht möglich. Als Konsequenz folgt hieraus, dass die Suche nach einem Gegenstand erheblich länger dauern kann.

Um dieses Problem zu beseitigen, könnten vorab mehrere Ansichten von einem Gegenstand aufgenommen werden, welche dabei die gleiche Szene wie die Referenzansicht zeigen. Zur Identifikation im aktuellen Bild müsste die Korrespondenzanalyse für jede der vorab gesicherten Ansichten vorgenommen werden.

Eine Einschränkung des entwickelten Greif-Verfahrens ist dadurch gegeben, dass ein Gegenstand nur bezüglich der vorab aufgenommenen Referenzansicht wiedererkannt und gegriffen werden kann. Zur Laufzeit kann es vorkommen, dass die in der Referenzansicht gezeigte Gegenstandsseite verdeckt ist. Das Greifen eines Objekts ist in diesem Fall somit nicht möglich. Zur Beseitigung dieser Problematik kann das Greif-Verfahren nach dem vorgestellten Prinzip auf mehrere Zielansichten und Teilrekonstruktionen rundum einen Gegenstand erweitert werden.

A Mathematische Herleitungen

A.1 Homogene Koordinaten

Die Konvertierung eines euklidischen Raumpunktes $\mathbf{p} = [x \ y \ z]^T$ in einen homogenen Raumpunkt $\tilde{\mathbf{p}} = [x_1 \ x_2 \ x_3 \ x_4]^T$ erfolgt mit (sh. ([SSVO09], S. 56-57):

$$\tilde{\mathbf{p}} = [x \ y \ z \ 1]^T. \quad (\text{A.1})$$

Eine Umkehrung wird anhand von

$$\mathbf{p} = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_4 & x_4 \end{bmatrix}^T \quad (\text{A.2})$$

durchgeführt.

A.2 Erste Ableitung der Rotationsmatrix und Winkelgeschwindigkeiten

Die Größe ${}^j\mathbf{w}_i$ definiert die voneinander unabhängigen Winkelgeschwindigkeiten um die jeweiligen Achsen x , y und z (vgl. Abschnitt 5.4):

$${}^j\mathbf{w}_i = [w_x \ w_y \ w_z]^T. \quad (\text{A.3})$$

Die Winkelgeschwindigkeiten ${}^j\mathbf{w}_i$ können laut ([SSVO09], S. 107) durch

$${}^j\dot{\mathbf{R}}_i = S({}^j\mathbf{w}_i) {}^j\mathbf{R}_i \quad (\text{A.4})$$

auf die erste Ableitung der Rotationsmatrix abgebildet werden. Hierbei steht $S(\cdot)$ für die schiefsymmetrische Matrix zu einem gegebenen Vektor $\in \mathfrak{R}^3$. Die Umkehrung kann folgendermaßen erreicht werden.

$$S({}^j\mathbf{w}_i) = {}^j\dot{\mathbf{R}}_i {}^i\mathbf{R}_j \quad (\text{A.5})$$

$$\begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}$$

Die Winkelgeschwindigkeiten können demnach wie folgt mit Hilfe des arithmetischen Mittels berechnet werden.

$$\begin{aligned} w_x &= \frac{-s_{23} + s_{32}}{2}, \\ w_y &= \frac{s_{13} - s_{31}}{2}, \\ w_z &= \frac{-s_{12} + s_{21}}{2}. \end{aligned} \quad (\text{A.6})$$

A.3 Moore-Penrose-Inverse

Die Moore-Penrose-Inverse oder auch Pseudoinverse \mathbf{A}^+ einer Matrix \mathbf{A} ergibt sich aus (vgl. [SSVO09], S. 575-576):

$$\begin{aligned} \mathbf{b} &= \mathbf{A}\mathbf{x} \\ \mathbf{A}^T\mathbf{b} &= \mathbf{A}^T\mathbf{A}\mathbf{x} \\ (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} &= (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{A})\mathbf{x} \\ \underbrace{(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T}_{\mathbf{A}^+}\mathbf{b} &= \mathbf{x}. \end{aligned} \tag{A.7}$$

Dabei muss die Bedingung erfüllt werden, dass der Rang der Matrix \mathbf{A} äquivalent zur Anzahl der Spalten von \mathbf{A} ist. Andernfalls existiert \mathbf{A}^+ nicht.

B Algorithmen

B.1 RANSAC

Im Folgenden wird der *Random Sample Consensus* (RANSAC)-Algorithmus aus ([FB81], S. 3-5) erläutert. RANSAC ermöglicht die robuste Schätzung von Modellparametern unter Berücksichtigung von fehlerbehafteten Messdaten.

Ziel:

Robuste Schätzung eines Modells M bezüglich der Punktmenge P , welche fehlerbehaftete Messungen enthält.

Vorbedingung:

Wenn zur Schätzung von M mindestens n Punkte benötigt werden, dann gilt $|P| \geq n$.

Algorithmus:

(1): Wähle zufällig n Punkte aus der Menge P aus und schätze das Modell M_i .

(2): Bestimme aus P die Untermenge P_i deren Elemente innerhalb eines Distanzschwellwertes t bezüglich M_i liegen. P_i wird hierbei als die Konsensmenge der aktuellen Abtastung bezeichnet.

(3.a): Wenn die Anzahl der Punkte $|P_i|$ der Konsensmenge größer als ein bestimmter Schwellwert T ist, dann schätze M_i unter Berücksichtigung aller Punkte in P_i erneut und terminiere. T hängt dabei von der vermuteten Anzahl der fehlerbehafteten Punkte in P ab.

(3.b): Gilt hingegen $|P_i| < T$, dann wiederhole den Vorgang ab Schritt 1. Nach N Wiederholungen verwende die größte Konsensmenge P_i hinsichtlich aller Abtastungen zur erneuten Schätzung von M_i oder terminiere mit einem Fehler.

Algorithmus B.1: RANSAC

Im Fall der erfolgreichen Terminierung liegt eine robuste Schätzung für M vor und bei den Elementen der zugehörigen Konsensmenge P_i handelt es sich um näherungsweise korrekte Punkte. Der vorgestellte Algorithmus ermöglicht es also auch die fehlerbehafteten Messungen aus der Punktmenge P herauszufiltern.

Um RANSAC einsetzen zu können müssen die Größen N , T und t bekannt sein. Zur Ermittlung dieser Parameter wird in [HZ03] auf den Seiten 118-120 das nachstehende Vorgehen beschrieben. Der Wert für N wird so gewählt, dass garantiert wenigstens eine der zufälligen Abtastungen mit n Punkten bei einer Wahrscheinlichkeit von p frei von Ausreißern ist. Auf diese Weise kann vermieden werden, dass im schlechtesten Fall alle möglichen Abtastungen berechnet werden müssen. Unter der Berücksichtigung eines relativen Ausreißeranteils ϵ in der Punktmenge P wird für die Berechnung von N folgende Gleichung angegeben:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^n)}. \quad (\text{B.1})$$

Die Größe n definiert dabei die minimale Anzahl der Punkte, die zur Schätzung des Modells M notwendig sind (vgl. Schritt 1 des Algorithmus B.1). Die Wahrscheinlichkeit p wird hierbei üblicherweise auf den Wert 0.99 gesetzt, um die Chancen einer erfolgreichen Terminierung so hoch wie möglich zu halten. Ein Nachteil der sich abzeichnet ist, dass der relative Ausreißeranteil ϵ gegeben sein muss. In der vorliegenden Arbeit werden hierfür empirische Werte herangezogen. Dieses Vorgehen wird auch für die Ermittlung des Distanzschwellwertes t durchgeführt, welcher im zweiten Schritt des RANSAC-Algorithmus benötigt wird.

Anhand der Gleichung B.1 gilt für die maximale Wiederholungsanzahl N folgendes Verhalten. Je größer p , n und ϵ sind, desto höher ist auch N . Mit steigender Genauigkeit erhöht sich also auch der Rechenaufwand.

Der Schwellwert T aus Schritt 3.a des RANSAC-Algorithmus gibt an wie viele korrekte Messungen in der Punktmenge P vermutet werden. Demnach kann T anhand des vermuteten Ausreißeranteils ϵ wie folgt ermittelt werden:

$$T = (1 - \epsilon)|P|. \quad (\text{B.2})$$

B.2 RANSAC mit adaptiver Parameterbestimmung

Bisher wurde vorausgesetzt, dass der relative Ausreißeranteil ϵ gegeben sein muss. Da diese Bedingung jedoch nicht immer erfüllbar ist, kann RANSAC auch durch eine adaptive Berechnung von ϵ wie folgt erweitert werden (sh. [HZ03], S. 120-121).

Ziel:

Robuste Schätzung eines Modells M bezüglich der Punktmenge P , welche fehlerbehaftete Messungen enthält. Der relative Ausreißeranteil ϵ ist dabei nicht bekannt.

Vorbedingung:

Wenn zur Schätzung von M mindestens n Punkte benötigt werden, dann gilt $|P| \geq n$.

Initialisierung:

- Initialisiere $\epsilon = 1$. Es wird somit zunächst vom schlechtesten Fall, einem Ausreißeranteil von 100 %, ausgegangen.
- Berechne N durch Formel B.1.
- Initialisiere $best_score = 0$.

Algorithmus:

(1): Wähle zufällig n Punkte aus der Menge P aus und schätze das Modell M_i .

(2): Bestimme aus P die Untermenge P_i deren Elemente innerhalb eines Distanzschwelleswertes t bezüglich M_i liegen.

(3): Im Fall $|P_i| > best_score$ aktualisiere ϵ anhand von Gleichung B.3 und anschließend N mittels Formel B.1. Speichere die aktuelle Konsensmenge $P_{best} = P_i$ und setze $best_score = |P_i|$. Wiederhole den Vorgang in jedem Fall ab Schritt 1.

(4): Nach N Wiederholungen: Schätze M_i unter Berücksichtigung aller Punkte in P_{best} erneut und terminiere.

Algorithmus B.2: RANSAC mit adaptiver Bestimmung von ϵ

$$\epsilon = 1 - \frac{|P_i|}{|P|}. \quad (\text{B.3})$$

B.3 Normalisierung von Bildpunkten

Um die Genauigkeit des Ergebnisses bei der numerischen Berechnung von Modellparametern auf Basis von Bildpunkten zu erhöhen, kann eine Normalisierung der verwendeten Bildpunkte als sogenannte Vorkonditionierung ausgeführt werden. Dies ist in besonderem Maße dann notwendig, wenn eine Ausgleichslösung auf Grund von fehlerbehafteten Bildpunkten berechnet werden muss (vgl. [HZ03], S. 107-108).

Eine geeignete Normalisierung für n homogene Bildpunkte \tilde{p}_i kann durch die nachstehende affine Transformation erreicht werden.

- Durch eine Verschiebung wird der Schwerpunkt der Bildpunkte p_i als neuer Ursprung festgelegt.

- Weiterhin wird eine isotropische Skalierung vorgenommen. Hierbei werden die Bildpunkte p_i so skaliert, dass deren mittlere Distanz vom neuen Ursprung $\sqrt{2}$ beträgt.

Zur Realisierung dieser Transformation muss zunächst der Schwerpunkt $\bar{p} = [\bar{x} \ \bar{y}]^T$ der n Bildpunkte p_i anhand des arithmetischen Mittels berechnet werden. Anschließend kann die Bestimmung der mittleren Distanz der einzelnen Bildpunkte p_i bezüglich ihres Schwerpunktes \bar{p} durch

$$\overline{dist} = \frac{1}{n} \sum_{i=1}^n \| \bar{p} - p_i \|_2 \quad (\text{B.4})$$

erfolgen. Der erläuterte isotropische Skalierungsfaktor kann somit per

$$s = \sqrt{2} / \overline{dist} \quad (\text{B.5})$$

definiert werden. Auf der Grundlage der berechneten Größen kann die beschriebene Normalisierung eines homogenen Bildpunktes \tilde{p} mit Hilfe der homogenen Transformationsmatrix rT durchgeführt werden:

$$\tilde{p}^r = {}^rT\tilde{p} = \begin{bmatrix} s & 0 & -\bar{x}s \\ 0 & s & -\bar{y}s \\ 0 & 0 & 1 \end{bmatrix} \tilde{p}. \quad (\text{B.6})$$

C Inhalt der CD-ROM

Die beiliegende CD-ROM beinhaltet die folgende Verzeichnis-Struktur:

```
kni_lib/  
pattern/  
pdf/  
projekt_katana/  
toolboxes/
```

Im Ordner *kni_lib/* befindet sich die Open-Source C++ Software-Bibliothek *Katana Native Interface* in der Version 4.0.0. Zur Nutzung der Steuerfunktionen unter Matlab wurde die Bibliothek modifiziert. Nähere Erläuterungen hierzu und zur Nutzung der Bibliothek befinden sich unter *kni_lib/readme.txt*.

Das Verzeichnis *pattern/* beinhaltet das Gittermuster aus Grafik 11.2, welches als Hilfestellung bei der Rekonstruktion von Raumpunkten genutzt wird. Eine Kopie der vorliegenden Arbeit liegt im Verzeichnis *pdf/*.

Im Ordner *projekt_katana/* befindet sich der Source-Code, welcher im Rahmen dieser Arbeit entstanden ist. Eine detaillierte Beschreibung der einzelnen Matlab-Module befindet sich unter *projekt_katana/readme.txt*. Die externen Matlab-Toolboxes, welche zur Realisierung des vorgestellten Greifsystems genutzt wurden (vgl. Tabelle 12.2), befinden sich im Verzeichnis *toolboxes/*.

Für die nachfolgend aufgeführten Größen werden SI-Einheiten verwendet. Aus diesem Grund wird nicht jeweils auf die Einheit einer Größe eingegangen.

Symbolverzeichnis

a_i, d_i	DH-Parameter eines Gelenks j_{i+1} bezüglich eines Gelenks j_i
B	Aktuelles Bild
B^*	Referenzansicht
d	SIFT-Merkmalvektor
e	Regelabweichung
E	Essentialmatrix
f	Bildweite
F	Fundamentalmatrix
H	Homography-Matrix
I	Einheitsmatrix
J_i	Bildjacobimatrix
J_G	Geometrische Jacobimatrix der Kinematik
J_R	Rotatorischer Anteil der geometrischen Jacobimatrix der Kinematik
J_T	Translatorischer Anteil der geometrischen Jacobimatrix der Kinematik
K	Kalibriermatrix
l	Epipolarlinie
m_x, m_y	Skalierung in x - und y -Richtung
${}^j \mathbf{o}_i$	Translationsvektor von $(KS)_i$ bezüglich $(KS)_j$
${}^j \dot{\mathbf{o}}_j$	Translatorischer Bestandteil der Geschwindigkeit eines Koordinatensystems
${}^0 \mathbf{p}_e$	Endeffektor-Pose relativ zu $(KS)_0$
\mathbf{p}^i	Euklidischer Punkt bezüglich $(KS)_i$
$\tilde{\mathbf{p}}^i$	Homogener Punkt bezüglich $(KS)_i$
$\dot{\mathbf{p}}^i$	Änderung eines Punktes bezüglich $(KS)_i$
P	Projektionsmatrix
q	Gelenkwinkel
\dot{q}	Gelenkgeschwindigkeiten
${}^j \mathbf{R}_i$	Rotationsmatrix von $(KS)_i$ bezüglich $(KS)_j$
${}^j \dot{\mathbf{R}}_i$	Erste Ableitung der Rotationsmatrix
s	Ist-Größe des Regelkreises: Die zu s^* korrespondierenden Bildpunkte
s^*	Soll-Größe des Regelkreises: Die zu s korrespondierenden Bildpunkte
s_k	Scherungsfaktor
${}^j \mathbf{T}_i$	Transformationsmatrix von $(KS)_i$ relativ zu $(KS)_j$

$[u_0 \ v_0]^T$	Skalierte Verschiebung des Bildhauptpunkts
${}^i \mathbf{v}_j$	Geschwindigkeit eines Koordinatensystems $(KS)_j$ relativ zu $(KS)_i$
${}^j \mathbf{v}_j$	Geschwindigkeit eines Koordinatensystems $(KS)_j$
${}^i \mathbf{V}_j$	Transformationsmatrix zum Wechsel des Bezugskordinatensystems der Geschwindigkeit eines Koordinatensystems von $(KS)_j$ nach $(KS)_i$
${}^j \mathbf{w}_j$	Rotatorischer Bestandteil der Geschwindigkeit eines Koordinatensystems
$[x_0 \ y_0]^T$	Verschiebung des Bildhauptpunkts
α_i, θ_i	DH-Parameter eines Gelenks j_{i+1} bezüglich eines Gelenks j_i
α_x, α_y	Skalierte Bildweite in x - und y -Richtung
δ_{dec}	Dezentrierende Verzeichnung
δ_{rad}	Radiale Verzeichnung
η	RPY-Winkel
λ	Regelverstärkung

Glossar

Teach-In Beschreibt hier einen Vorgang bei dem der Endeffektor manuell in eine Lage bezüglich eines Gegenstands geführt wird, um anschließend Lage- oder Bildinformationen zu speichern.

Tool Center Point Fiktiver Punkt am Endeffektor

camera retreat Die Kamera bewegt sich bei der bildpunktbasierter Regelung auf Grund von hohen Verdrehungen um die optische Achse erst vom Ziel weg und später wieder darauf zu.

direct visual servo system Die Gelenke des Roboters werden direkt anhand von Gelenkwinkeln eingestellt und eine Regelung der Gelenkgeschwindigkeiten entfällt somit.

dynamic look-and-move system Kaskadierte Roboterregelung bei der die Ansteuerung des Roboters über einen Regler für Gelenkgeschwindigkeiten erfolgt.

eye-in-hand Es werden eine oder mehrere Kameras in der Nähe des Endeffektors montiert.

eye-to-hand Eine oder mehrere Kameras sind statisch an einer festen Position angebracht und beobachten den Endeffektor und dessen Zielposition.

Endeffektor Letztes Glied innerhalb einer kinematischen Kette. In dieser Arbeit handelt es sich um eine Greifvorrichtung.

Abkürzungen

CCD	<i>Charge-coupled Device</i>
DH	<i>Denavit-Hartenberg</i>
DLT	<i>Direct Linear Transformation</i>
EIH	<i>eye-in-hand</i>
ETH	<i>eye-to-hand</i>
FHG	Freiheitsgrad
GLOH	<i>Gradient Location and Orientation Histogram</i>
IFR	<i>International Federation of Robotics</i>
KNI	<i>Katana Native Interface</i>
KS	Koordinatensystem
MKS	Mehrkörpersystem
PbD	<i>Programming by Demonstration</i>
RANSAC	<i>Random Sample Consensus</i>
RPY	<i>Roll-Pitch-Yaw</i>
SIFT	<i>Scale Invariant Feature Transform</i>
SURF	<i>Speeded-Up Robust Features</i>
TCP	<i>Tool Center Point</i>
VS	<i>Visual Servoing</i>

Abbildungsverzeichnis

1.1	Katana-Greifarm vom Typ 450-6M180	2
2.1	Einstellung der Greiflage	3
2.2	Direkte Steuerung	4
2.3	Objektpose bezüglich der Kamera in der Start- und Zielsituation	4
2.4	Regelkreis	6
2.5	Hand-Auge-Konfiguration (Bildquelle: [SRV06])	7
2.6	Bildpunktbasieretes VS	8
2.7	Korrespondierende Bildpunkte	9
2.8	3-D VS	10
2.9	Pose zwischen der Kamera in der Start- und Zielsituation	11
2.10	Folgeregelung	13
2.11	Greifvorgang	15
2.12	Box-Approximation (Bildquelle: [HK])	17
2.13	Ermittelte Greifpunkte (Bildquelle: [SDKN])	18
2.14	Schachtel aus der ersten Perspektive	20
2.15	Schachtel aus der zweiten Perspektive	20
2.16	Übereinstimmende Bildpunkte bezüglich der überlagerten Bilder 2.14 und 2.15	21
3.1	Ablauf eines Greifvorgangs	26
3.2	Direkte Steuerung	27
4.1	Drehachsen des Katana-Greifarms	29
4.2	Arbeitsraum des Katana-Greifarms (Bildquelle: [NEU08])	30
4.3	Koordinatensystem des Roboters (Bildquelle: [NEU08])	31
4.4	Katana-Roboter von oben	31
5.1	Pose eines starren Körpers	34
5.2	Elementare Rotationen (Bildquelle: [HEI])	35
5.3	Rotationen laut RPY-Winkelkonvention (Bildquelle: [HEI])	35
5.4	DH-Parameter	38
5.5	Abmessungen des Katana-Greifarms 6M180 (Bildquelle: [NEU08])	42
5.6	Kinematische Kette	43
6.1	Zentralprojektion	45
6.2	Sensor- und Bildkoordinatensystem	45
6.3	Arten der radialen Verzeichnung (Bildquelle: [GRA07])	47
6.4	Tangentiale Verzeichnung von Linien	47
6.5	Radiale und tangentiale Verzeichnung eines Punktes	47
7.1	Kalibriermuster	50
7.2	Bilder des Kalibrierusters	51
7.3	Eckpunkte des Kalibrierusters	51
7.4	Subpixelgenaue Eckpunkte des Kalibrierusters	52
7.5	Aufbau zur Kalibrierung	56
7.6	Modell für die Hand-Auge-Kalibrierung	58
8.1	Verwendete Posen für die direkte Steuerung	60
9.1	Bildpunktbasierete Regelung	62
9.2	Bahn der verfolgten Bildpunkte s (blau) in Richtung der Ziel-Bildpunkte s^* (rosa). (Bildquelle: [CHA98])	64
10.1	SIFT-Punkte (rot) aus der Referenzansicht B^*	72
10.2	SIFT-Punkte (grün) aus einer zweiten Ansicht B	72

10.3	Vermeintliche Übereinstimmungen bezüglich der überlagerten Bilder 10.1 und 10.2	73
10.4	Tatsächliche Übereinstimmungen bezüglich der überlagerten Bilder 10.1 und 10.2	73
10.5	SIFT-Punkte (grün) aus einer dritten Ansicht B	75
10.6	Übereinstimmungen bezüglich der überlagerten Bilder 10.1 und 10.5	75
11.1	Epipolarometrie	77
11.2	Gittermuster zur Rekonstruktion	81
11.3	Eckpunkte des Gittermusters	83
11.4	Ausgewählte Eckpunkte des Gittermusters	84
11.5	Punktepaare bezüglich der überlagerten Ansichten B^* und B	89
11.6	Sicht in die Ebenen der rekonstruierten Teebox	90
11.7	Sicht auf die Ebenen der rekonstruierten Teebox	90
12.1	Erfassung von Greifparametern und Rekonstruktionsdaten	91
12.2	Direkte Steuerung	94
12.3	Bildpunktbasierte Regelung	97
13.1	Simulationsumgebung	99
13.2	Überlagerte Bildpunkte bezüglich der Referenzansicht B^* (rot) und der aktuellen Ansicht B (grün)	100
13.3	Regelabweichung e	101
13.4	Verlauf der Bildpunkte	101
13.5	Geschwindigkeit 0v_c der Kamera bezüglich des Umwelt-Koordinatensystems	102
13.6	Lage der Kamera c bei einer Drehung von 100 Grad bezüglich der optischen Achse	102
13.7	Verlauf der Bildpunkte	103
13.8	Kamera-Geschwindigkeit 0v_c	103
13.9	Simulationsumgebung	104
13.10	Lage der Kamera c nach der direkten Steuerung	105
13.11	Lage der Kamera c nach der bildpunktbasierten Regelung	106
13.12	Regelabweichung e	106
13.13	Verlauf der Bildpunkte	107
13.14	Kamera-Geschwindigkeit 0v_c	107
14.1	Katana-Greifarm nach der Identifikation der Schachtel im aktuellen Kamerabild	108
14.2	Punktepaare bezüglich der überlagerten Ansichten B^* und B nach der Wiedererkennung	109
14.3	Punktepaare bezüglich der überlagerten Ansichten B^* und B nach der direkten Steuerung	110
14.4	Punktepaare bezüglich der überlagerten Ansichten B^* und B nach der Regelung	110
14.5	Regelabweichung e	111
14.6	Verlauf der Bildpunkte	111
14.7	Geschwindigkeit 0v_c der Kamera bezüglich des Umwelt-Koordinatensystems	112
14.8	Katana-Greifarm nach der Regelung	113
14.9	Katana-Greifarm nach dem Greifen der Schachtel	113
14.10	Punktepaare bezüglich der überlagerten Ansichten B^* und B während der Regelung	115
14.11	Kamera-Geschwindigkeit 0v_c	115

Tabellenverzeichnis

3.1	Eigenschaften des Greifkonzepts	25
4.1	Eigenschaften der Kamera mit Objektiv	32
5.1	Länge der Armglieder	42
5.2	DH-Parameter des Katana-Greifarms	43
7.1	Kalibrierergebnisse für die Guppy-Kamera	52
7.2	Mittelwert und Standardabweichung der Reprojktionsfehler	53
7.3	Ergebnisse für die Hand-Auge-Kalibrierung	57
12.1	Rekonstruktionsdaten und Greifparameter	93
12.2	Verwendete Bibliotheken	98
13.1	Lage der Kamera c bezüglich der Kamera c^*	105

Algorithmenverzeichnis

7.1 Robuste Schätzung der Werkstück-Lage (${}^c\mathbf{R}_w, {}^c\mathbf{o}_w$)	55
B.1 RANSAC	121
B.2 RANSAC mit adaptiver Bestimmung von ϵ	123

Quellenverzeichnis

Literaturquellen

- [AS03] ARMANGUÉ, X. ; SALVI, J.: Overall View Regarding Fundamental Matrix Estimation. In: *Image and Vision Computing* 21 (2003), Februar, Nr. 2, S. 205–220
- [BC07] BOURQUARDEZ, O. ; CHAUMETTE, F.: Visual servoing of an airplane for auto-landing. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007), November, S. 1314–1319
- [BETVG08] BAY, H. ; ESS, A. ; TUYTELAARS, T. ; VAN GOOL, L.: SURF: Speeded Up Robust Features. In: *Computer Vision and Image Understanding* 110 (2008), Nr. 3, S. 346–359
- [BGH⁺08] BOURQUARDEZ, O. ; GUENARD, N. ; HAMEL, T. ; CHAUMETTE, F. ; MAHONY, R. ; ECK, L.: Kinematic visual servo controls of an X4-flyer: practical study. In: *2008 Mediterranean Conf. on Intelligent Systems and Automation* (2008), Juni
- [BK10] BOHG, J. ; KRAGIC, D.: Learning grasping points with shape context. In: *Robotics and Autonomous Systems* 58 (2010), April, Nr. 4, S. 362–377
- [CCO08] CHERUBINI, A. ; CHAUMETTE, F. ; ORIOLO, G.: An image-based visual servoing scheme for following paths with nonholonomic mobile robots. In: *10th International Conference on Control, Automation, Robotics and Vision* (2008), Dezember, S. 108–113
- [CH01] CORKE, P. I. ; HUTCHINSON, S. A.: A new partitioned approach to image-based visual servo control. In: *IEEE Transactions on Robotics and Automation* 17 (2001), August, Nr. 4, S. 507–515
- [CH06] CHAUMETTE, F. ; HUTCHINSON, S.: Visual servo control, Part I: Basic approaches. In: *IEEE Robotics and Automation Magazine* 13 (2006), Dezember, Nr. 4, S. 82–90
- [CH07] CHAUMETTE, F. ; HUTCHINSON, S.: Visual servo control, Part II: Advanced approaches. In: *IEEE Robotics and Automation Magazine* 14 (2007), März, Nr. 1, S. 109–118
- [CMC08] COLLEWET, C. ; MARCHAND, E. ; CHAUMETTE, F.: Visual servoing set free from image processing. In: *2008 IEEE International Conference on Robotics and Automation* (2008), Mai, S. 81–86
- [Cor96] CORKE, P. I.: A Robotics Toolbox for MATLAB. In: *IEEE Robotics and Automation Magazine* 3 (1996), März, Nr. 1, S. 24–32
- [EK05] EKVAL, S. ; KRAGIC, D.: Integrating object and grasp recognition for dynamic scene interpretation. In: *Proceedings of the 12th International Conference on Advanced Robotics* (2005), Juli, S. 331–336
- [FB81] FISCHLER, M. A. ; BOLLES, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: *Communications of the ACM* 24 (1981), Nr. 6, S. 381–395

- [Fit01] FITZGIBBON, A. W.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1* (2001), S. 125–132
- [GH03] GANS, N. R. ; HUTCHINSON, S. A.: An experimental study of hybrid switched system approaches to visual servoing. In: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation 3* (2003), September, S. 3061–3068
- [GH07] GANS, N. R. ; HUTCHINSON, S. A.: Stable visual servoing through hybrid switched-system control. In: *IEEE Transactions on Robotics* 23 (2007), Juni, Nr. 3, S. 530–540
- [HD95] HORAUD, R. ; DORNAIKA, F.: Hand-Eye Calibration. In: *The International Journal of Robotics Research* 14 (1995), Juni, Nr. 3, S. 195–210
- [HHC96] HUTCHINSON, S. ; HAGER, G. D. ; CORKE, P. I.: A tutorial on visual servo control. In: *IEEE Transactions on Robotics and Automation* 12 (1996), Oktober, Nr. 5, S. 651–670
- [HK08] HUEBNER, K. ; KRAGIC, D.: Selection of robot pre-grasps using box-based shape approximation. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2008), September, S. 1765–1770
- [HNSR06] HOFFMANN, F. ; NIEROBISCH, T. ; SEYFFARTH, T. ; RUDOLPH, G.: Visual Servoing with Moments of SIFT Features. In: *2006 IEEE International Conference on Systems, Man and Cybernetics* (2006), Oktober, S. 4262–4267
- [Hor07] HORNING, O.: *Bildbasierte Bahnplanung und Regelung von Robotern für Handhabungsaufgaben*. VDI Verlag GmbH, 2007 (Mess-, Steuerungs- und Regelungstechnik). – ISBN 978-3-18-511608-7
- [HS88] HARRIS, C. ; STEPHENS, M.: A Combined Corner and Edge Detector. In: *Proceedings of the Fourth Alvey Vision Conference* (1988), S. 147–151
- [HS97a] HARTLEY, R. I. ; STURM, P.: Triangulation. In: *Computer Vision and Image Understanding* 68 (1997), November, Nr. 2, S. 146–157
- [HS97b] HEIKKILA, J. ; SILVEN, O.: A four-step camera calibration procedure with implicit image correction. In: *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1997), S. 1106–1112
- [HZ03] HARTLEY, R. ; ZISSERMAN, A.: *Multiple View Geometry in Computer Vision*. 2. Auflage. Cambridge University Press, 2003. – ISBN 0-521-54051-8
- [KBCE05] KRAGIC, D. ; BJÖRKMAN, M. ; CHRISTENSEN, H. I. ; EKLUNDH, J.-O.: Vision for robotic object manipulation in domestic settings. In: *Robotics and Autonomous Systems* 52 (2005), Juli, Nr. 1, S. 85–100
- [KC03] KRAGIC, D. ; CHRISTENSEN, H. I.: A Framework for Visual Servoing. In: *Lecture Notes in Computer Science* 2626 (2003), S. 345–354

- [LJK03] LEE, P.-M. ; JEON, B.-H. ; KIM, S.-M.: Visual servoing for underwater docking of an autonomous underwater vehicle with one camera. In: *OCEANS 2003. Proceedings 2* (2003), September, S. 677–682
- [Low04] LOWE, D. G.: Distinctive image features from scale-invariant keypoints. In: *International Journal of Computer Vision* 60 (2004), November, Nr. 2, S. 91–110
- [MA04] MILLER, A. T. ; ALLEN, P. K.: Graspit! A versatile simulator for robotic grasping. In: *IEEE Robotics and Automation Magazine* 11 (2004), Dezember, Nr. 4, S. 110–122
- [MAA⁺06] MORALES, A. ; ASFOUR, T. ; AZAD, P. ; KNOOP, S. ; DILLMANN, R.: Integrated Grasp Planning and Visual Object Localization For a Humanoid Robot with Five-Fingered Hands. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006), Oktober, S. 5663–5668
- [MC00] MALIS, E. ; CHAUMETTE, F.: 2 1/2 D Visual Servoing with Respect to Unknown Objects Through a New Estimation Scheme of Camera Displacement. In: *International Journal of Computer Vision* 37 (2000), Nr. 1, S. 79–97
- [MC08] MAREY, M. ; CHAUMETTE, F.: Analysis of classical and new visual servoing control laws. In: *2008 IEEE International Conference on Robotics and Automation* (2008), Mai, S. 3244–3249
- [MCB99] MALIS, E. ; CHAUMETTE, F. ; BOUDET, S.: 2-1/2-D visual servoing. In: *IEEE Transactions on Robotics and Automation* 15 (1999), April, Nr. 2, S. 238–250
- [MP05] MARIOTTINI, G. L. ; PRATTICHIZZO, D.: EGT: a Toolbox for Multiple View Geometry and Visual Servoing. In: *IEEE Robotics and Automation Magazine* 3 (2005), Dezember, Nr. 12
- [MS05] MIKOLAJCZYK, K. ; SCHMID, C.: A performance evaluation of local descriptors. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005), Nr. 10, S. 1615–1630
- [PCCS06] PAGES, J. ; COLLEWET, C. ; CHAUMETTE, F. ; SALVI, J.: Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light. In: *IEEE Transactions on Robotics* 22 (2006), Oktober, Nr. 5, S. 1000–1010
- [PKB⁺10] POPOVIĆ, M. ; KRAFT, D. ; BODENHAGEN, L. ; BAŞESKI, E. ; PUGEAULT, N. ; KRAGIC, D. ; ASFOUR, T. ; KRÜGER, N.: A strategy for grasping unknown objects based on co-planarity and colour information. In: *Robotics and Autonomous Systems* 58 (2010), Mai, Nr. 5, S. 551–565
- [RLSP06] ROTHGANGER, F. ; LAZEBNIK, S. ; SCHMID, C. ; PONCE, J.: Object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. In: *International Journal of Computer Vision* 66 (2006), Nr. 3
- [SDN08] SAXENA, A. ; DRIEMEYER, J. ; NG, A. Y.: Robotic grasping of novel objects using vision. In: *The International Journal of Robotics Research* 27 (2008), Februar, Nr. 2, S. 157–173

- [SM07] SILVEIRA, G. ; MALIS, E.: Direct visual servoing with respect to rigid objects. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007), Oktober/November, S. 1963–1968
- [SSVO09] SICILIANO, B. ; SCIAVICCO, L. ; VILLANI, L. ; ORIOLO, G.: *Robotics. Modelling, planning and control*. Springer London, 2009 (Advanced Textbooks in Control and Signal Processing). – ISBN 978-1-84628-641-4
- [VMO05] VITRANI, M.-A. ; MOREL, G. ; ORTMAIER, T.: Automatic Guidance of a Surgical Instrument with Ultrasound Based Visual Servoing. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005), April, S. 508–513
- [WCH92] WENG, J. ; COHEN, P. ; HERNIOU, M.: Camera calibration with distortion models and accuracy evaluation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Oktober, Nr. 10, S. 965–980

Sonstige Quellen

- [All] ALLIED VISION TECHNOLOGIES: *AVT Cameras - Guppy F-080B / F-080C*. [HTTP://WWW.ALLIEDVISIONTEC.COM/FILEADMIN/CONTENT/PDF/PRODUCTS/DATA_SHEET/CAMERAS/GUPPY/GUPPY_DATASHEET_F080BC_V3.0.0_EN.PDF](http://www.alliedvisiontec.com/fileadmin/content/pdf/products/data_sheet/cameras/guppy/guppy_datasheet_f080bc_v3.0.0_en.pdf). – Zugriffsdatum: 7. Oktober 2010
- [Bou] BOUGUET, J.-Y.: *Camera Calibration Toolbox for Matlab*. [HTTP://WWW.VISION.CALTECH.EDU/BOUGUETJ/CALIB_DOC/](http://www.vision.caltech.edu/bouguetj/calib_doc/). – Stand: 27. Juli 2010. Zugriffsdatum: 2. Februar 2011
- [BSP07] BAUER, J. ; SÜNDERHAUF, N. ; PROTZEL, P.: *Comparing Several Implementations of Two Recently Published Feature Detectors*. International Conference on Intelligent and Autonomous Systems (2007). [HTTP://WWW.TU-CHEMNITZ.DE/ETIT/PROAUT/RSRC/IAV07-SURF.PDF](http://www.tu-chemnitz.de/etit/proaut/rsrc/IAV07-SURF.pdf). Version: 2007. – Zugriffsdatum: 2. Februar 2011
- [CFK⁺] CAPEL, D. ; FITZGIBBON, A. ; KOVESI, P. ; WERNER, T. ; WEXLER, Y. ; ZISSERMAN, A.: *MATLAB Functions for Multiple View Geometry*. [HTTP://WWW.ROBOTS.OX.AC.UK/~VGG/HZBOOK/CODE/](http://www.robots.ox.ac.uk/~vgg/hzbook/code/). – Zugriffsdatum: 2. Februar 2011
- [Cha98] CHAUMETTE, F.: Potential problems of stability and convergence in image-based and position-based visual servoing. In: *The Confluence of Vision and Control LNCIS Series 237* (1998), 66-78. [HTTP://WWW.IRISA.FR/LAGADIC/PDF/1998_LNCIS_CHAUMETTE.PDF](http://www.irisa.fr/LAGADIC/PDF/1998_LNCIS_CHAUMETTE.PDF). – Zugriffsdatum: 2. Februar 2011
- [Gra07] GRAF, S.: *Kamerakalibrierung mit radialer Verzeichnung - die radiale essentielle Matrix*, Universität Passau, Dissertation, September 2007. [HTTP://WWW.OPUS-BAYERN.DE/UNI-PASSAU/VOLLTEXTE/2008/1271/PDF/GRAF_SIMONE.PDF](http://www.opus-bayern.de/uni-passau/volltexte/2008/1271/pdf/GRAF_SIMONE.PDF). – Zugriffsdatum: 2. Februar 2011
- [Hei] HEIMANN, B.: *Robotik I*. Skript. [HTTP://WWW.IMES.UNI-HANNOVER.DE/UPLOAD/ROBOTIK/ROBOTIK_SKRIPT_WS_08-09.PDF](http://www.imes.uni-hannover.de/upload/robotik/robotik_skript_ws_08-09.pdf). – Zugriffsdatum: 17. Januar 2010

- [HK] HUEBNER, K. ; KRAGIC, D.: *Selection of Robot Pre-Grasps using Box-Based Shape Approximation*. [HTTP://WWW.NADA.KTH.SE/~DANIK/PAPERS/IROS08_KH.PDF](http://www.nada.kth.se/~danik/papers/iros08_kh.pdf). – Zugriffsdatum: 2. Februar 2011
- [ino] INOS AUTOMATIONSSOFTWARE GMBH: *inos Visual Servoing*. [HTTP://WWW.INOS-AUTOMATION.DE/DE/VISUAL_SERVOING.HTML](http://www.inos-automation.de/de/visual_servoing.html). – Zugriffsdatum: 2. Februar 2011
- [Inta] INTERNATIONAL FEDERATION OF ROBOTICS: *Service Robots - Products*. [HTTP://WWW.IFR.ORG/SERVICE-ROBOTS/PRODUCTS/](http://www.ifr.org/service-robots/products/). – Zugriffsdatum: 2. Februar 2011
- [Intb] INTERNATIONAL FEDERATION OF ROBOTICS: *World Robotics 2010 - Executive Summary*. [HTTP://WWW.IFR.ORG/UPLOADS/MEDIA/2010_EXECUTIVE_SUMMARY_REV_01.PDF](http://www.ifr.org/uploads/media/2010_executive_summary_rev_01.pdf). – Zugriffsdatum: 2. Februar 2011
- [KC02] KRAGIC, D. ; CHRISTENSEN, H. I.: *Survey on visual servoing for manipulation*. [FTP://FTP.NADA.KTH.SE/CVAP/REPORTS/CVAP259.PS.GZ](ftp://ftp.nada.kth.se/cvap/reports/cvap259.ps.gz). Version: 2002. – Zugriffsdatum: 2. Februar 2011
- [Kov] KOVESI, P. D.: *MATLAB and Octave Functions for Computer Vision and Image Processing*. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. [HTTP://WWW.CSSE.UWA.EDU.AU/~PK/RESEARCH/MATLABFNS/](http://www.csse.uwa.edu.au/~pk/research/matlabfns/). – Zugriffsdatum: 2. Februar 2011
- [Neu08] NEURONICS AG (Hrsg.): *Katana 450 Benutzerhandbuch*. Neuronics AG, 2008. – Version 2.0.3
- [SDKN] SAXENA, A. ; DRIEMEYER, J. ; KEARNS, J. ; NG, A. Y.: *Robotic Grasping of Novel Objects*. [HTTP://WWW.CS.CORNELL.EDU/~ASAXENA/LEARNINGGRASP/NIPS_LEARNINGGRASP.PDF](http://www.cs.cornell.edu/~asaxena/learninggrasp/nips_learninggrasp.pdf). – Zugriffsdatum: 2. Februar 2011
- [srv06] *Visual Servoing*. Skript. [HTTP://WWW.RST.E-TECHNIK.UNI-DORTMUND.DE/CMS/MEDIENPOOL/DOWNLOADS/LEHRE/VORLESUNGEN/ROBOTICS_THEORY/VISUALSERVOING_4.PDF](http://www.rst.e-technik.uni-dortmund.de/cms/medienpool/downloads/lehre/vorlesungen/robotics_theory/visualservoing_4.pdf). Version: 2006. – Zugriffsdatum: 2. Februar 2011
- [Tor] TORR, P. H. S.: *Structure and Motion Toolkit in Matlab*. [HTTP://CMS.BROOKES.AC.UK/STAFF/PHILIPTORR/](http://cms.brookes.ac.uk/staff/philipTorr/). – Zugriffsdatum: 2. Februar 2011
- [VF] VEDALDI, A. ; FULKERSON, B.: *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. [HTTP://WWW.VLFEAT.ORG/](http://www.vlfeat.org/). – Zugriffsdatum: 2. Februar 2011
- [Wen] WENGERT, C.: *Camera Calibration and Hand to Eye Calibration Toolbox*. [HTTP://WWW.VISION.EE.ETHZ.CH/SOFTWARE/CALIBRATION_TOOLBOX//CALIBRATION_TOOLBOX.PHP](http://www.vision.ee.ethz.ch/software/calibration_toolbox/calibration_toolbox.php). – Zugriffsdatum: 2. Februar 2011

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 11. Februar 2011

Ort, Datum

Unterschrift