



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Peter Oltmann

3D-Lagebestimmung von Objekten für Assistenzroboter

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Peter Oltmann

3D-Lagebestimmung von Objekten für Assistenzroboter

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing.
Zweitgutachter: Prof. Dr. Wolfgang Fohl

Eingereicht am: 30. August 2012

Peter Oltmann

Thema der Arbeit

3D-Lagebestimmung von Objekten für Assistenzroboter

Stichworte

3D-Lagebestimmung, 3D-Rekonstruktion, Identifikation, Objekterkennung, Bildmerkmale, SIFT, SURF, ROS

Kurzzusammenfassung

Diese Arbeit behandelt ein Verfahren, das der Identifikation und 3D-Lagebestimmung von Objekten dient. Aus perspektivisch unterschiedlichen Kameraaufnahmen eines Objekts werden dessen dreidimensionale Oberflächenpunkte rekonstruiert. Diesen Punkten werden eindeutige Bildmerkmale zugeordnet, um eine Identifikation zu ermöglichen. Durch die Implementierung dieses Verfahrens ist es möglich, ein 3D-Modell eines Objekts zu erstellen, sodass die dreidimensionale Lage bestimmt werden kann.

Peter Oltmann

Title of the paper

3D positioning of objects for assisting robots

Keywords

3D positioning, 3D reconstruction, identification, image features, object detection, SIFT, SURF, ROS

Abstract

This thesis examines a procedure that conduces to the identification and 3D positioning of objects. From different perspective camera views of an object its three-dimensional surface points are reconstructed. Distinct image features are assigned to those points to permit the identification. In consequence of implementing this procedure it is possible to create an 3D model of an object so the three-dimensional position can be determined.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	2
1.3	Gliederung	4
2	Technischer Aufbau und Entwicklungsumgebung	5
2.1	Kamera	5
2.2	ROS – Robot Operating System	6
2.3	OpenCV	8
3	Identifikation von Objekten	9
3.1	Bildmerkmale	9
3.1.1	SIFT	10
3.1.2	SURF	11
3.1.3	Weitere Merkmalstypen	13
3.2	Korrespondenzsuche	14
3.2.1	Brute Force Search	14
3.2.2	FLANN – Fast Approximate Nearest Neighbors	14
3.2.3	Filterung	15
4	3D-Rekonstruktion	17
4.1	Kamerakalibrierung	17
4.1.1	Kameramodell	17
4.1.2	Intrinsische Kameraparameter und Linsenverzeichnung	19
4.1.3	Kalibriervorgang	20
4.1.4	Entfernung der Linsenverzeichnung	21
4.2	Bestimmung der 3D-Weltpunkte	22
4.2.1	Fundamentalmatrix	22
4.2.2	Projektionsmatrizen	25
4.2.3	Bestimmung der Projektionsmatrizen	26
4.2.4	Triangulation der Weltpunkte	30
4.2.5	Verifikation der Weltpunkte	31
4.2.6	Ermittlung des Skalierungsfaktors	34
4.2.7	Erweiterung der 3D-Weltpunktmenge	36

5	Implementierung	40
5.1	Architektur	40
5.1.1	Hauptprogramm	42
5.1.2	Merkmalsextraktion	44
5.1.3	3D-Rekonstruktion	45
5.1.4	Datenverwaltung und -speicherung	46
6	Ergebnisse	47
6.1	Identifikation von Objekten	47
6.2	3D-Rekonstruktion	49
6.2.1	Initiale Bestimmung einer 3D-Welpunktmenge	49
6.2.2	Erweiterung einer 3D-Welpunktmenge	51
6.3	Möglichkeiten und Grenzen des Verfahrens	55
7	Fazit	57
	Abbildungsverzeichnis	58
	Tabellenverzeichnis	60
	Glossar	61
	Formelzeichen	62
	Literaturverzeichnis	64

1 Einleitung

1.1 Motivation

Roboter erhalten in einem immer größeren Ausmaß Einzug in unser Leben. Vor allem autonome Assistenzroboter sind alte Zukunftsvisionen, deren Verwirklichung in vollem Gang ist.

Im Forschungsprojekt Robot Vision aus dem Departement der Hochschule für Angewandte Wissenschaften Hamburg wird mit Assistenzrobotern gearbeitet und an ihnen geforscht. Hierbei handelt es sich um Roboter, die autonom agieren sollen und können. Im Gegensatz zu Robotern, die einem fest einprogrammiertem Arbeitsablauf folgen, müssen diese Roboter sich in unbekanntem Umgebungen selbständig zurechtfinden. Diese Roboter werden sukzessive mit neu entwickelten Fähigkeiten ausgestattet.

Neben Laserscannern, Messsystemen und Spracherkennungs-Sensorik befindet sich im Robot Vision Labor auch der Katana Greifarm der Firma Neuronics¹. Dieser ist ein ortsgebundener Knickarm-Roboter mit einer über dem Greifer montierten Kamera. Der Katana Greifarm ist in Abbildung 1.1a dargestellt.

Weiterhin befindet sich dort auch die Roboterplattform „SCITOS G5“ der Firma MetraLabs² mit einem darauf montierten Greifarm der Firma Schunk³. Die Roboterplattform ist frei beweglich und u.a. mit 24 Ultraschallsensoren und einem 2D-Laserscanner ausgestattet. Der Greifarm hat fünf Freiheitsgrade (ohne Greifer) und hat eine maximale Nutzlast von 5 kg. Die SCITOS G5 Plattform mit Greifarm ist in Abbildung 1.1b dargestellt. Der Roboter wird mit dem ROS-Framework⁴ betrieben und weiterentwickelt.

¹<http://www.neuronics.ch>

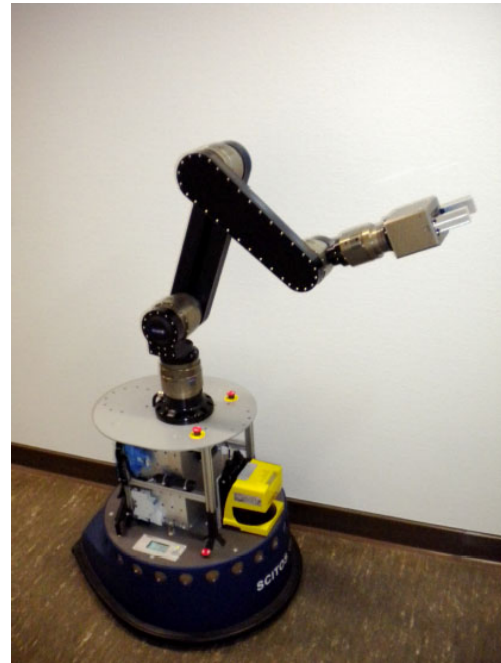
²MetraLabs GmbH: mobile robots - <http://www.metralabs.com>

³Schunk GmbH & Co. KG: Modular Robotics - <http://www.schunk-modular-robotics.com>

⁴Robot Operating System - <http://www.ros.org>



(a) Katana-Greifarm



(b) SCITOS G5 mit Greifarm

Abbildung 1.1: Roboter des Robot Vision-Projekts

1.2 Zielsetzung

Ziel der Arbeit ist die Untersuchung und Implementation eines Verfahrens zur kamerabasierten 3D-Lagebestimmung von Objekten. Die Ausarbeitungen zu diesem Thema von Carsten Fries in seiner Masterthesis (Fri11) sollen hierfür die Grundlage bilden.

Im Rahmen der vorliegenden Arbeit wird eine Software entwickelt, die mit dem ROS-Framework betrieben werden soll. Für die Bildverarbeitung wird die in ROS integrierte Open Source Library OpenCV⁵ verwendet.

Ausgehend von Kameraaufnahmen eines Objekts sollen dessen dreidimensionale Oberflächenpunkte rekonstruiert werden, sodass eine 3D-Lagebestimmung möglich ist. Um das zu erreichen, werden 2D-Korrespondenzen von Bildpunkten aus perspektivisch unterschiedlichen Aufnahmen eines Objekts benötigt. In diesem Zusammenhang soll die Möglichkeit geschaffen werden, die rekonstruierten Punkte in weiteren Kameraaufnahmen wiederzuerkennen und zu erweitern.

⁵Open Source Computer Vision - <http://opencv.org>

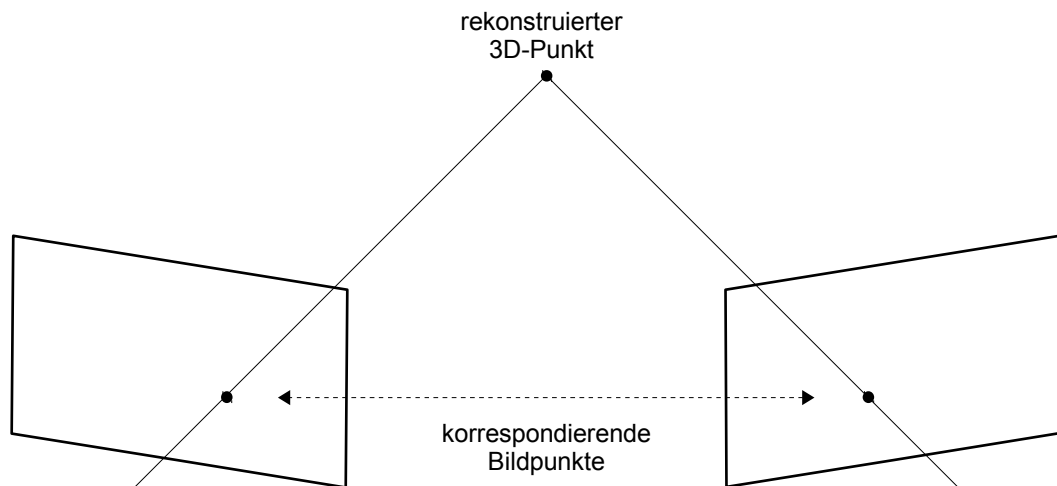


Abbildung 1.2: 3D-Rekonstruktion von korrespondierenden Bildpunkten

Zur Ermittlung der korrespondierenden Bildpunkte sollen Bildmerkmale dienen. Hierfür muss ein Objekt aus mehreren Perspektiven aufgenommen werden und es müssen Merkmale in diesen Aufnahmen wiedererkannt und einander zugeordnet werden können. Wichtig für die darauf aufbauende 3D-Rekonstruktion ist es, hierbei Fehlzuordnungen auszuschließen oder zu erkennen. Die Robustheit und Eignung der Bildmerkmale kann von Fall zu Fall variieren, weshalb es ermöglicht werden soll, verschiedene Merkmalstypen zu verwenden. Hier gibt es bereits Algorithmen zur Erkennung und Beschreibung verschiedener Arten von Merkmalen, auf die zurückgegriffen werden kann.

Basierend auf den in verschiedenen Aufnahmen wiedererkannten und einander zugeordneten Bildmerkmalen soll eine 3D-Rekonstruktion der Punkte vollzogen werden. Dies soll in einer Menge dreidimensionaler Weltpunkte resultieren. Durch die Zuordnung der Bildmerkmale zu den rekonstruierten Punkten wird versucht die Wiedererkennbarkeit der Weltpunkte zu erreichen. Wird ein Teil der Weltpunkte in weiteren Kameraaufnahmen wiedererkannt, soll die Menge durch die aus den neuen Aufnahmen rekonstruierten Punkte erweitert werden.

Ziel der zu entwickelnden Software ist die Erstellung eines aus dreidimensionalen Punkten bestehenden Modells, das der Beschreibung eines Objekts dienen soll. Als Ausgangspunkt dienen lediglich perspektivisch verschiedene Kameraaufnahmen des verwendeten Objekts. Um die 3D-Weltpunkte eines Objekts in einem bereits erstellten Modell wiedererkennen und erweitern zu können, werden den Weltpunkten die Bildmerkmale der 2D-Korrespondenzen zugeordnet, aus denen sie rekonstruiert wurden.

1.3 Gliederung

Kapitel 1 beschreibt die Motivation und die Zielsetzung der Arbeit.

Kapitel 2 gibt einen Überblick über den technischen Aufbau und die eingesetzten Entwicklungswerkzeuge in dieser Arbeit.

Kapitel 3 erläutert die Identifikation von Objekten über Bildmerkmale und die Suche nach Merkmalskorrespondenzen in verschiedenen Kameraaufnahmen.

Kapitel 4 beschreibt das Verfahren der 3D-Rekonstruktion. Neben der Kamerakalibrierung wird die Berechnung dreidimensionaler Weltpunkte aus korrespondierenden Bildpunkten in perspektivisch unterschiedlichen Kameraaufnahmen behandelt. Es wird der Weg erläutert, wie aus Kameraaufnahmen eines Objekts eine das Objekt beschreibende Weltpunktmenge erzeugt und erweitert werden kann.

Kapitel 5 beschreibt die in dieser Arbeit entwickelte Software.

Kapitel 6 stellt Ergebnisse dieser Arbeit zusammen und analysiert diese.

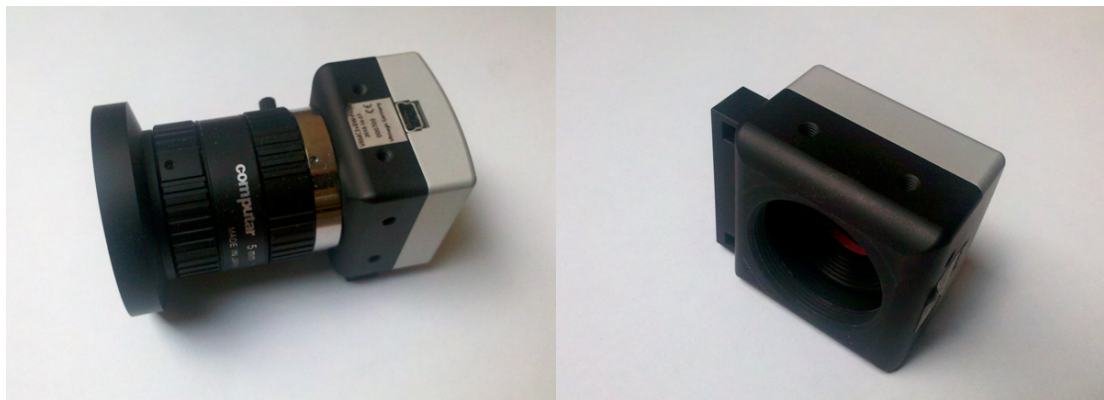
Kapitel 7 fasst die Resultate dieser Arbeit zusammen und es wird ein Fazit gezogen.

2 Technischer Aufbau und Entwicklungsumgebung

Dieses Kapitel bietet einen kurzen Überblick über die in dieser Arbeit eingesetzten Werkzeuge und Umgebungen, deren wichtigste Eigenschaften im Folgenden erläutert und zusammengefasst sind.

2.1 Kamera

Für alle in dieser Arbeit benötigten Kameraaufnahmen wurde die monochrome Kamera *VRmC3+PRO-BW* der Firma *VRmagic*¹ verwendet (Abbildung 2.1).



(a) Mit Objektiv

(b) Ohne Objektiv

Abbildung 2.1: VRmC3+PRO-BW

Auf Grund ihrer Größe bietet es sich an, die Kamera auf den Greifarm eines Roboters zu montieren. Durch den Anschluss und die Stromversorgung über USB 2.0 ist dies auf den

¹VRmagic GmbH: <http://www.vrmagic.com/>

meisten Plattformen ohne Weiteres möglich. Der Hersteller liefert Treiber und Programmierschnittstellen sowohl für Linux als auch für Windows. Allerdings muss ein Treiberprogramm für ROS (vgl. nächster Abschnitt 2.2) geschrieben werden, sodass die Kamera auch dort genutzt werden kann.

In Tabelle 2.1 sind die technischen Daten der Kamera aufgelistet.

Spezifikationen	VRmC3+PRO-USB2.0-BW
Kamera-Klasse	Flächenkamera, monochrom
Sensor-Klasse	Global Shutter-CMOS
Auflösung (effektiv)	754 x 482 (H x V)
Bildfrequenz (Standard)	69 Hz
Bit pro Pixel	8 bit
Digitales Interface	USB 2.0
Versorgungsspannung	via USB 2.0
Anschlüsse	Standard Mini-USB-B
Objektivanschluss	C-Mount
Abmessungen	36 x 36 x 27 (B x H x T in mm)
Gewicht	70 g

Tabelle 2.1: Technische Daten der VRmC3+PRO-USB2.0-BW

2.2 ROS – Robot Operating System

Das *Robot Operating System*² bietet eine Plattform, um eine einfache Entwicklung von Applikationen in der Robotik zu gewährleisten. Es ist jedoch weniger, wie der Name vermuten lässt, ein Betriebssystem als vielmehr ein Framework. ROS enthält bereits viele Applikationen, Programmbibliotheken und Gerätetreiber, bietet jedoch auch einfach zugängliche Schnittstellen, um diese zu ergänzen. Als Mittel der Interprozesskommunikation dient das Konzept des Message Passing.

Die Software dieser Arbeit wird für ROS entwickelt und kann somit auf allen mit ROS ausgestatteten Computern und Robotern betrieben werden. ROS kann im Grunde auf jedem Computer installiert werden. Ubuntu ist hierbei als Betriebssystem unterstützt und zu empfehlen. Da es nativ in C/C++ programmiert ist, ist eine hohe Performance gewährleistet. Es werden allerdings auch Python-Schnittstellen geboten, um die Plattformunabhängigkeit zu erhöhen.

²ROS: <http://www.ros.org/>

Die Architektur bietet dem Entwickler einfache Möglichkeiten, seine Applikationen in ROS zu integrieren und mit den bereits vorhandenen zu interagieren. Die einzelnen Module in ROS werden als *Packages* bezeichnet. Ein Package kann mehrere Komponenten enthalten. Dies können etwa ROS-unabhängige Bibliotheken, Konfigurationsdateien oder sogenannte *Knoten* sein.

Ein Knoten stellt die eigentliche Applikation dar und wird als Prozess im Betriebssystem gestartet. Über ROS kann ein Knoten mit anderen Knoten über *Messages* kommunizieren. Hierbei gibt es auf der einen Seite die Möglichkeit der asynchronen Kommunikation. Diese läuft über sogenannte *Topics* ab. Ein Knoten abonniert ein Topic (*Subscriber*), um alle von anderen Knoten an das Topic gesendeten Nachrichten (*Publisher*) zu erhalten. Auf der anderen Seite besteht die Möglichkeit der synchronen Kommunikation über *Services*. Diese stellen das Pendant zu den Topics dar. Knoten können eine Anfrage (*Request*) an einen Service stellen und blockieren, bis ein anderer Knoten darauf geantwortet hat (*Reply*).

Um Anwendungen (Knoten) ausführen zu können, muss zunächst der Namensdienst von ROS, der *Master*, gestartet werden. Dieser dient als zentrale Stelle zur Organisation der Kommunikation. Ein Knoten muss sich nach dem Start des Prozesses beim Master anmelden, damit eine Kommunikation mit anderen Knoten gewährleistet werden kann.

In Abbildung 2.2 ist zur Verdeutlichung ein beispielhafter, sogenannter ROS *Computation Graph* dargestellt. Dieser beschreibt alle im Master registrierten Knoten sowie die Topics und ggf. Services, über die kommuniziert wird. In diesem Beispiel sind drei Knoten aktiv. Der Kameratreiber (*vrmagic_mono_driver*) veröffentlicht die von der Kamera produzierten Bilder über das Topic *image_raw*. Jeder Knoten, der dieses Topic abonniert hat, enthält die Bilder. In diesem Fall ist das nur der Knoten *image_record*, der der Aufnahme einzelner Bilder dient, um diese über das Topic *record* zu veröffentlichen. Der Knoten *model3d*, der die in dieser Arbeit entwickelte Software in ROS integriert, nimmt diese Kameraaufnahmen entgegen. Der Knoten *image_record* steht repräsentativ für alle Knoten, die verwertbare Kameraaufnahmen erzeugen können. Dies kann z. B. ein Roboter sein, der verschiedene Aufnahmen von einem Gegenstand macht.

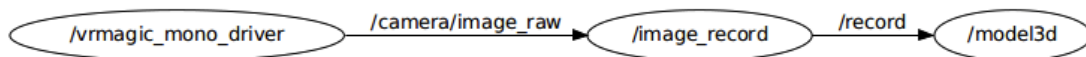


Abbildung 2.2: ROS *Computation Graph*

2.3 OpenCV

OpenCV³ (Open Source Computer Vision Library) ist eine Open Source Softwarebibliothek für Bildverarbeitung und maschinelles Sehen. Es ist viel Wert auf die Effizienz der implementierten Algorithmen gelegt worden, weshalb OpenCV auch in C/C++ programmiert ist.

Die Nutzung von OpenCV in dieser Arbeit bietet sich aus mehreren Gründen an. Ein großer Vorteil ist, dass OpenCV ein fester Bestandteil von ROS ist. Außerdem bietet es eine große Anzahl an implementierten Algorithmen, die auch in dieser Arbeit Anwendung finden. Weiterhin ist die Speicherverwaltung von großen Objekten, wie zum Beispiel von Bildern, effizient implementiert. Dies gilt ebenso für die in dieser Arbeit häufig benötigte Matrixarithmetik.

So können in dieser Arbeit die in OpenCV vorhandenen Algorithmen zur Detektion und Extraktion verschiedener Typen von Bildmerkmalen und zur Suche nach Punktkorrespondenzen in perspektivisch unterschiedlichen Kameraaufnahmen (Kapitel 3) verwendet werden. Außerdem bietet es Routinen zur Kamerakalibrierung und einige unterstützende Funktionen für die 3D-Rekonstruktion (Kapitel 4).

³OpenCV: <http://www.opencv.org/>

3 Identifikation von Objekten

Dieses Kapitel behandelt die Identifikation von Objekten durch die Wiedererkennung korrespondierender Bildpunkte in perspektivisch unterschiedlichen Kameraaufnahmen. Dies geschieht durch die Erkennung und Extraktion sogenannter Bildmerkmale. Ziel ist das Aufstellen einer Korrespondenzmenge von Objektpunkten in diesen Kameraaufnahmen, sodass deren dreidimensionale Lage rekonstruiert werden kann (Kapitel 4). Durch die Bildmerkmale ist die Möglichkeit gegeben, diese Punkte zu identifizieren.

3.1 Bildmerkmale

Bildmerkmale dienen der Wiedererkennung von Punkten in unterschiedlichen Bildern. So können Objekte in unterschiedlichen Kameraaufnahmen identifiziert werden, ohne dass auf dem Objekt etwaige Markierungen vorgenommen werden müssen.

Ein Bildmerkmal wird je nach Typ auf verschiedene Art und Weisen beschrieben. Dies muss jedoch so geschehen, dass ein Vergleich der Merkmale durchführbar ist. Gleiche Objektpunkte in unterschiedlichen Kameraaufnahmen sollen durch nahezu identische Bildmerkmale repräsentiert werden. Meist wird die eindeutige Beschreibung eines Merkmals durch die Verwendung eines hochdimensionalen Vektors, dem *Deskriptor*, erreicht.

Damit die Wiedererkennung eines Merkmals möglich ist, muss neben der Eindeutigkeit auch die Invarianz des Merkmals gegeben sein. Zentral notwendige Anforderungen sind hierbei, dass der Deskriptor sowohl invariant gegenüber Rotationen in der Bildebene als auch gegenüber unterschiedlicher Skalierung ist. Weiterhin ist es von großem Vorteil, wenn die Merkmale bis zu einem gewissen Grad robust gegenüber der bei der räumlichen Rotation eines Objekts auftretenden perspektivischen Verzeichnung sowie Störung und Beleuchtungsänderungen sind.

Es gibt verschiedene Merkmalstypen und Algorithmen zur Merkmalsextraktion. Die Eigenschaften und zentralen Berechnungsschemen einiger Merkmalstypen sind im Folgenden

aufgeführt und zusammengefasst. Die Algorithmen werden hier jedoch nicht erneut erläutert. Für genauere Informationen sei an dieser Stelle auf die entsprechenden Veröffentlichungen verwiesen, wie etwa (Low99), (Low04) und (BETG08).

3.1.1 SIFT

SIFT (*Scale-Invariant Feature Transform*) (Low04) extrahiert Bildmerkmale aus Kameraaufnahmen. SIFT-Merkmale sind skalierungs- und rotationsinvariant. Auch sind sie robust gegenüber einem geringen Grad an perspektivischer Verzerrung, Störung und Beleuchtungsänderungen.

Ein SIFT-Merkmal besteht aus zwei Komponenten. Dies ist zum einen der sogenannte *Keypoint*. Ein Keypoint enthält die Informationen über den Ort sowie die Region eines Merkmals. Der Ort wird durch eine Bildkoordinate $(x, y)^T$ beschrieben. Die Region wird definiert durch einen Winkel, der die sogenannte Hauptorientierung eines Merkmals darstellt. Die Hauptorientierung wird während der Extraktion ermittelt. Über sie wird versucht die Rotationsinvarianz zu erreichen. Außerdem ist die Größe der Region durch einen Wert beschrieben. Zum anderen besteht ein Merkmal aus einem das Merkmal beschreibenden Vektor, dem *Deskriptor*. Über den Deskriptor kann ein Merkmal in unterschiedlichen Aufnahmen identifiziert und wiedererkannt werden.

Als Grundlage für die Extraktion der Keypoints dienen sogenannte DoG-Pyramiden (*Difference of Gaussian*). Dabei werden in der Skalierung variierende Versionen eines Bildes mit einem Gaußfilter gefaltet und voneinander subtrahiert. Daraufhin wird in diesen Pyramiden nach Maxima gesucht. Hierbei wird ein Pixel mit seinen 26 Nachbarn verglichen. Ein gefundenes Maximum stellt einen potentiellen Keypoint dar.

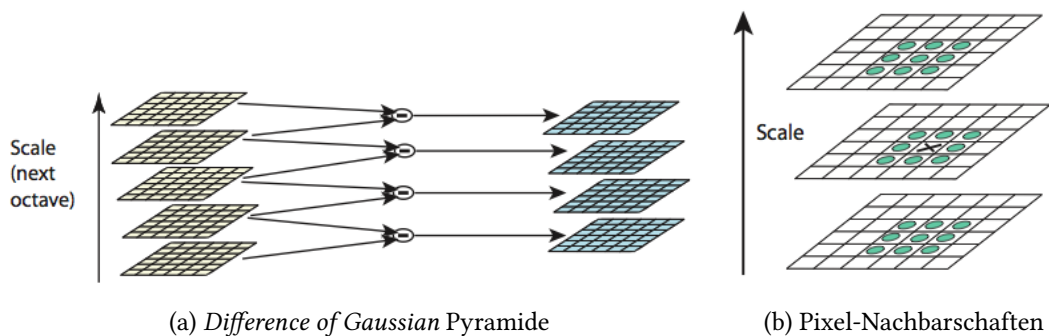


Abbildung 3.1: DoG-Pyramide (Low04)

Die Menge der potentiellen Keypoints muss gefiltert werden, denn ein Keypoint muss gewisse Stabilitätskriterien aufweisen. An dieser Stelle findet eine Untersuchung des Kontrastes im Vergleich zu seinen Nachbarn sowie eine Detektion auf Ecken, Kanten und homogene Regionen statt. Abschließend wird die Hauptorientierung eines Merkmals durch die Betrachtung der Gradienten der benachbarten Bildpunkte bestimmt.

Der SIFT-Deskriptor ist ein 128-dimensionaler Vektor. Er beschreibt die Merkmalsumgebung relativ zu ihrer Hauptorientierung, wodurch die Rotationsinvarianz erreicht wird. Hierbei wird die Merkmalsregion in kleinere Bereiche aufgeteilt. Die Orientierungen dieser Bereiche sind, quantisiert nach Wertebereichen, im Merkmalsvektor enthalten. Dieser Zusammenhang ist in Abbildung 3.2 dargestellt.

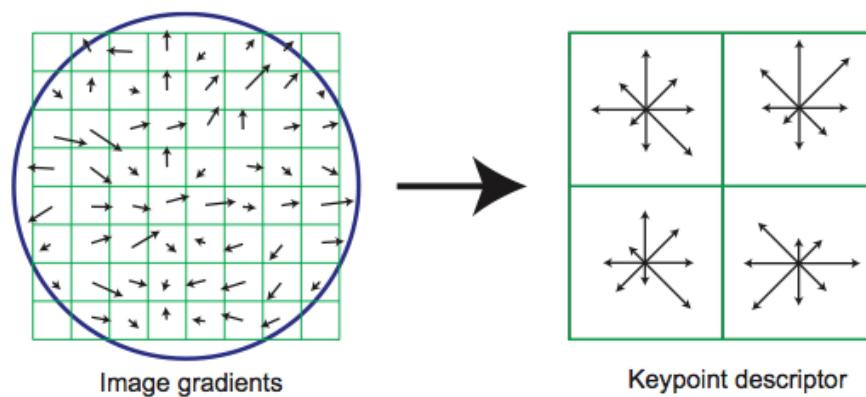


Abbildung 3.2: SIFT-Deskriptor (Low04)

3.1.2 SURF

SURF (*Speeded Up Robust Features*) (BETG08) ähnelt dem als Vorbild dienenden SIFT-Verfahren. Ziel der Entwicklung war eine Steigerung der Geschwindigkeit bei der Extraktion der Merkmale. Erreicht wurde dies durch Approximation oder Ersetzung der benutzten Berechnungsschemen. Die Geschwindigkeitserhöhung verringert allerdings die Anzahl der extrahierten Merkmale. Die Entwickler beschreiben den Verlust jedoch als akzeptabel.

So basiert die Detektion von Merkmalen nicht auf DoG-Pyramiden, sondern auf der Hesse-Matrix der Faltung der 2. Ableitung der Gaußfunktion. Maxima werden durch den Vergleich der Determinanten bestimmt.

$$\mathbf{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \quad (3.1)$$

Hierbei stellt $\mathbf{x} = (x, y)^T$ den betrachteten Bildpunkt dar, σ den Skalierungsfaktor und L die Faltung durch die zweite Ableitung der Gaußfunktion. Die Gaußfilter werden im SURF-Verfahren durch Mittelwertfilter (*box filters*) approximiert (Abbildung 3.3).

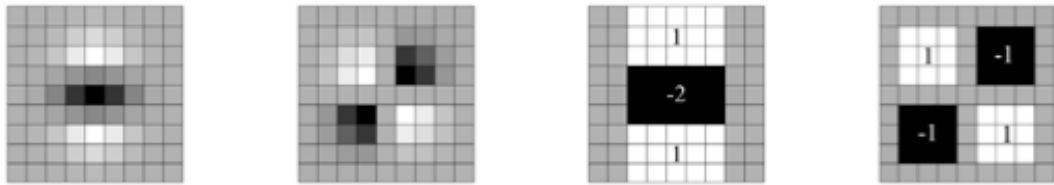


Abbildung 3.3: L_{xx} und L_{xy} bzw. Approximation durch Mittelwertfilter (BETG08)

Durch die Verwendung von Integralbildern kann die Anzahl der Berechnungen bei der Faltung enorm verringert werden. Ein Integralbild enthält die Summe aller Pixel eines Bildes zwischen dem betrachteten Punkt $\mathbf{x} = (x, y)^T$ und dem Bildursprung $(0, 0)^T$.

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (3.2)$$

Ist ein Integralbild aufgestellt, lassen sich Pixelsummen in rechteckigen Bildausschnitten durch die Addition der Werte der vier Eckpunkte berechnen (vgl. Abbildung 3.4). Die Anzahl der Additionen ist demnach unabhängig von der Größe des Ausschnitts. Dies bringt große Vorteile mit sich, da im SURF-Verfahren große Filtergrößen verwendet werden.

Der SURF-Deskriptor weist ähnliche Eigenschaften wie der SIFT-Deskriptor auf. Zur Berechnung der Hauptorientierung sowie der Bereichsorientierungen werden sogenannte Haar-Wavelets¹ verwendet. Die Bereiche sind, anders als beim SIFT-Verfahren, so aufgeteilt, dass der SURF-Deskriptor als 64-dimensionaler Vektor definiert ist.

¹Funktion zur Zeit-Frequenz-Transformation

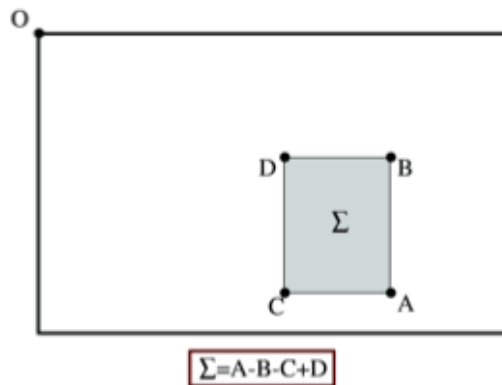


Abbildung 3.4: Berechnung von Pixelsummen durch Integralbilder (BETG08)

3.1.3 Weitere Merkmalstypen

Neben SIFT und SURF existieren noch weitere Merkmalstypen. Da in dieser Arbeit die Softwarebibliothek OpenCV (vgl. Abschnitt 2.3) verwendet wird, stehen alle dort implementierten Algorithmen zur Merkmalsextraktion zur Verfügung. Einige der vorhandenen Verfahren sind im Folgenden aufgelistet.

FAST ist ein Algorithmus zur Detektion von Ecken mit geringer Rechenintensität und Komplexität, entwickelt für Echtzeitanwendungen (RD06).

MSER *Maximally Stable Extremal Region Extractor* ist ein Verfahren zur sogenannten *blob detection*. Es dient der Extraktion von zusammenhängenden Regionen in Bildern (MCUP02).

ORB ist ein Verfahren, das von seinen Entwicklern als effiziente Alternative zu SIFT oder SURF bezeichnet wird. Den Entwicklern nach habe sich im Vergleich zu SIFT die Geschwindigkeit gesteigert, während die Präzision meist erhalten bleibe (RRKB11).

FREAK Mit dem *Fast Retina Keypoint* ist ein Keypoint-Deskriptor, inspiriert durch die menschliche Netzhaut (Retina), entwickelt worden. Den Entwicklern nach sei die Berechnungszeit und Speicherauslastung geringer als beim SIFT- oder SURF-Verfahren. Außerdem wiesen die FREAK-Deskriptoren einen höheren Grad an Robustheit auf (AOV12).

3.2 Korrespondenzsuche

Durch eine Suche nach Merkmalskorrespondenzen kann eine Menge korrespondierender Bildpunkte zwischen zwei Kameraaufnahmen aufgestellt werden. Die Suche findet über einen Vergleich der Deskriptoren der Bildmerkmale statt. Gesucht ist der einem Deskriptor nächste Nachbar (*Nearest Neighbors Search*).

3.2.1 Brute Force Search

Die sogenannte *Brute Force Search* basiert auf dem euklidischen Abstand d zweier Deskriptoren \mathbf{D}_1 und \mathbf{D}_2 der Dimension n .

$$d(\mathbf{D}_1, \mathbf{D}_2) = \sqrt{\sum_{i=1}^n (\mathbf{D}_{1i} - \mathbf{D}_{2i})^2} \quad (3.3)$$

Gegeben seien zwei Mengen an Deskriptoren $\{\mathbf{D}\}$ und $\{\mathbf{D}'\}$, die jeweils aus einer Kameraaufnahme extrahiert wurden. Bei der Brute Force Search wird nun jeder Deskriptor der ersten Menge mit jedem Deskriptor der zweiten Menge verglichen. Der Vektor \mathbf{D}'_j mit dem geringsten euklidischen Abstand zu einem Vektor \mathbf{D}_i ist dessen korrespondierender Deskriptor. Dadurch ist ein Paar an korrespondierenden Bildpunkten gefunden.

3.2.2 FLANN – Fast Approximate Nearest Neighbors

„Fast Approximate Nearest Neighbors With Automatic Algorithm Configuration“ (ML09) ist eine Methode zur Ermittlung des besten Algorithmus zur Korrespondenzsuche. Die Wahl des Algorithmus ist abhängig von Art, Anzahl und Größe der gegebenen Vektoren. Neben der Brute Force Search gibt es jedoch oftmals keinen schnelleren Algorithmus, der ebenso exakte Ergebnisse liefert. Approximierte Algorithmen hingegen liefern häufig enorme Geschwindigkeitssteigerungen bei nur geringem Präzisionsverlust. FLANN analysiert die vorliegenden Daten und wählt dann den passenden approximierten Algorithmus sowie dessen optimale Parameter aus.

In (ML09) wurde experimentell ermittelt, dass, abhängig von den vorliegenden Daten, entweder die Nutzung *hierarchischer k-mean-trees* oder *multiple randomized kd-trees* die beste Alternative darstellt. Die Funktionsweise dieser Algorithmen wird an dieser Stelle jedoch nicht erläutert.

Für weitere Informationen sei auf (ML09) bzw. die dort referenzierten Veröffentlichungen verwiesen. In OpenCV wurde FLANN implementiert. In der Dokumentation (Ope12) finden sich weitere Informationen, wie etwa über dessen Parametrisierung.

3.2.3 Filterung

Würden die Verfahren zur Korrespondenzsuche ohne weitere Filterung angewendet werden, so hätte jedes extrahierte Merkmal der ersten Menge ein korrespondierendes Merkmal in der zweiten Menge. Hier würden mit großer Wahrscheinlichkeit viele Fehlzusordnungen auftreten, weil zum einen die Möglichkeit besteht, dass der geringste Abstand sehr groß ist, zum anderen können die beiden geringsten Abstände sehr nahe beieinander liegen. Demnach wären die Zuordnungen nicht eindeutig. Dieser Sachverhalt ist in Abbildung 3.5 veranschaulicht. Dabei stellt d_1 den geringsten und d_2 den zweitgeringsten Abstand dar. In 3.5a sind beide Abstände sehr gering, jedoch sehr nahe beieinander. Die Zuordnung ist also nicht eindeutig. In 3.5b ist der Abstand d_1 sehr groß. Setzt man d_2 dazu ins Verhältnis, so ist auch diese Zuordnung nicht eindeutig. In 3.5c hingegen ist d_1 im Vergleich zu d_2 sehr viel kleiner. Hier ist die Zuordnung eindeutig.

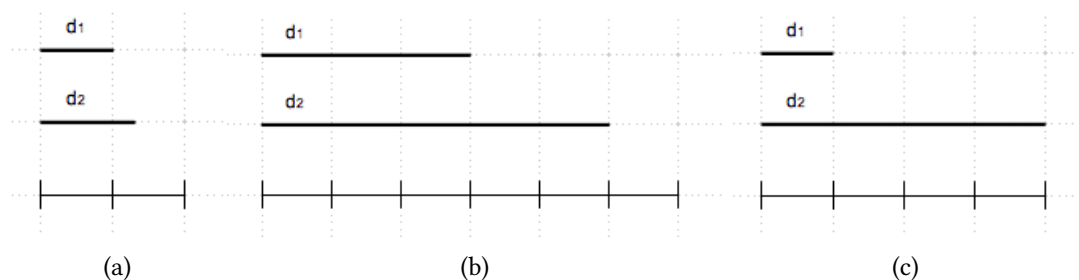


Abbildung 3.5: Vergleich der beiden kleinsten Abstände zwischen zwei Deskriptoren

Aus diesem Grund wird einem Bildpunkt nur dann eine Korrespondenz zugeordnet, wenn das Verhältnis der beiden geringsten Abstände zwischen den betrachteten Deskriptoren einen bestimmten Schwellwert (*threshold*) nicht übersteigt.

$$d_1 < \textit{threshold} \cdot d_2 \quad \text{mit } 0 < \textit{threshold} < 1 \quad (3.4)$$

Weiterhin besteht die Möglichkeit, nur die Merkmalskorrespondenzen zu übernehmen, die einen sehr hohen Grad an Robustheit aufweisen. Dies kann dadurch erreicht werden, dass eine dritte Kameraaufnahme des Objekts herangezogen wird. Es werden dann nur die Merkmale

3 Identifikation von Objekten

übernommen, zu denen sowohl auf der zweiten als auch auf der dritten Kameraaufnahme Korrespondenzen zu finden sind. Das kann ebenfalls dazu dienen, Bildmerkmale in der Umgebung eines betrachteten Objekts auszusortieren.

4 3D-Rekonstruktion

Dieses Kapitel behandelt ein Verfahren zur Bestimmung dreidimensionaler Punkte aus perspektivisch unterschiedlichen Kameraaufnahmen. Dafür muss eine Menge korrespondierender Bildpunkte in diesen Aufnahmen bekannt sein. Das Ergebnis ist eine Menge daraus rekonstruierter 3D-Weltpunkte. Es wird sowohl die initiale Bestimmung der Weltpunkte behandelt als auch die Erweiterung einer bereits bekannten Weltpunktmenge. Damit eine Erweiterung oder eine spätere Identifikation möglich ist, müssen die den 2D-Korrespondenzen zugeordneten Deskriptoren bekannt sein (vgl. Kapitel 3). Bevor die Rekonstruktion der Punkte durchgeführt werden kann, muss die für die Aufnahmen verwendete Kamera kalibriert werden.

4.1 Kamerakalibrierung

Dieser Abschnitt beschreibt den mathematischen Zusammenhang der Geometrie einer Kamera und 3D-Weltpunkten sowie die Ermittlung der Parameter dieser Geometrie durch einen automatisierten Kalibriervorgang.

4.1.1 Kameramodell

Das Aufnehmen von Bildern durch Kameras geschieht durch die Projektion von Licht auf einen Bildsensor. Das Licht trifft dabei auf eine Bildebene. Vor der Bildebene befindet sich eine Linse, die für die Bündelung der eintreffenden Lichtstrahlen sorgt. Das Kamerabild auf der Bildebene ist dabei auf den Kopf gestellt.

Mathematisch kann das Modell allerdings einfacher beschrieben werden. Eine Kamera projiziert einen 3D-Weltpunkt $\mathbf{X} = (X, Y, Z)^T$ auf einen 2D-Bildpunkt $\mathbf{x} = (x, y)^T$. Das sogenannte *Lochkameramodell* zur Beschreibung dieser Transformation ist in Abbildung 4.1 dargestellt.

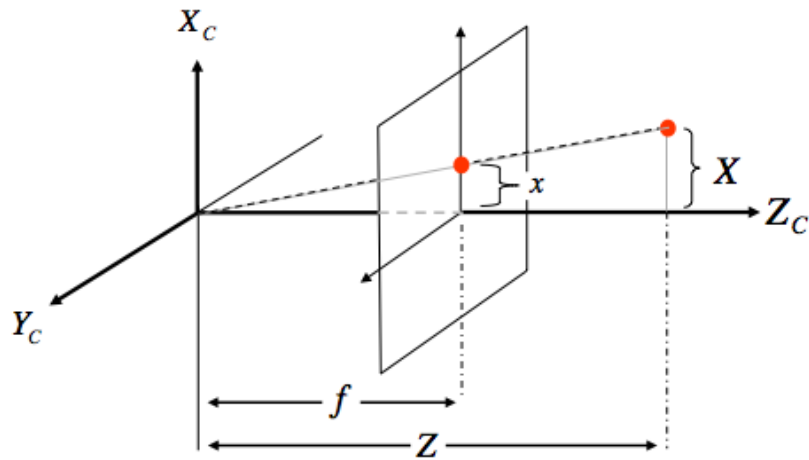


Abbildung 4.1: Lochkameramodell (Mei08)

Die in diesem Modell beschriebene Transformation kann in zwei Schritten betrachtet werden. Zunächst wird ein in Weltkoordinaten angegebener Raumpunkt \mathbf{X} durch eine Rotation \mathbf{R} und eine Translation \mathbf{t} in Koordinaten transformiert, die ihren Ursprung im Kamerazentrum haben. \mathbf{R} und \mathbf{t} beschreiben also eine Transformation in das Kamerakoordinatensystem und werden als *extrinsische Kameraparameter* bezeichnet.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{R} \begin{bmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{Z} \end{bmatrix} + \mathbf{t} \quad (4.1)$$

Als zweiter Schritt wird die Projektion von Kamerakoordinaten in Bildkoordinaten betrachtet. Parametrisiert wird diese durch die *intrinsischen* Kameraparameter sowie die Linsenverzeichnungskoeffizienten. Dies wird im folgenden Abschnitt 4.1.2 erläutert.

Das Kameramodell kann durch eine Projektionsmatrix, welche die sogenannten *intrinsischen* und *extrinsischen* Kameraparameter enthält, beschrieben werden. Dies wird im späteren Abschnitt 4.2.2 aufgegriffen und ausführlich erläutert.

4.1.2 Intrinsische Kameraparameter und Linsenverzeichnung

Die intrinsischen Kameraparameter beschreiben die innere Geometrie einer Kamera. Weiterhin gibt es die Linsenverzeichnungskoeffizienten. Diese Parameter sind unabhängig von der beobachteten Szene.

1. Bildweite $\mathbf{f} = (f_x, f_y)^T$
2. Bildhauptpunkt $\mathbf{h} = (h_x, h_y)^T$
3. Verzeichnungskoeffizienten $\mathbf{d} = (k_1, k_2, p_1, p_2, k_3)^T$

Die Bildweite beschreibt den Abstand des Kamerazentrums zur Bildebene in Pixeln (vgl. Abbildung 4.1). Die Kalibrierung kann unterschiedliche, jedoch nahezu identische Werte für die Projektion der X - bzw. Y -Koordinate liefern. Sie wird demnach durch die Gleichung 4.2 beschrieben.

$$\begin{aligned}\tilde{x} &= f_x \frac{X}{Z} \\ \tilde{y} &= f_y \frac{Y}{Z}\end{aligned}\tag{4.2}$$

Der Bildhauptpunkt beschreibt den Durchstoßpunkt der Hauptachse (Z -Achse) durch die Bildebene. Dieser Punkt liegt nicht immer exakt im Zentrum der Bildebene, sondern ist meist um einige Pixel verschoben.

$$\begin{aligned}x &= \tilde{x} + h_x \\ y &= \tilde{y} + h_y\end{aligned}\tag{4.3}$$

Die Abbildung von Kamerakoordinaten in Bildkoordinaten kann auch in homogenen Koordinaten beschrieben werden. Die 3×3 Kalibriermatrix \mathbf{K} enthält die intrinsischen Kameraparameter.

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix}\tag{4.4}$$

Die Verzeichnungskoeffizienten k_1 , k_2 und k_3 beschreiben die radiale Linsenverzeichnung. Die Koeffizienten p_1 und p_2 hingegen beschreiben die tangentiale Verzeichnung. Die Linsenverzeichnung wird durch die Gleichung 4.5 beschrieben, wobei $r^2 = x^2 + y^2$ ist.

$$\begin{aligned} x_d &= x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x^2) \\ y_d &= \underbrace{y(1 + k_1r^2 + k_2r^4 + k_3r^6)}_{\text{radial}} + \underbrace{2p_1(r^2 + 2y^2) + 2p_2xy}_{\text{tangential}} \end{aligned} \quad (4.5)$$

4.1.3 Kalibriervorgang

In dieser Arbeit kommt das in OpenCV implementierte Verfahren zur automatischen Ermittlung der intrinsischen Kameraparameter und Verzeichnungskoeffizienten zum Einsatz. Der Algorithmus richtet sich nach (Zha98) und stellt die OpenCV-Variante der *Camera Calibration Toolbox for Matlab* (Bou08) dar.

Für die Ermittlung der Parameter muss ein Kalibriermuster verwendet werden, dessen Geometrie bekannt ist und das leicht wiederzuerkennende Bildmerkmale aufweist. Typischerweise wird ein ebenes Schachbrettmuster eingesetzt, dessen Ecken leicht wiedererkennbar sind (Abbildung 4.2). Die Abstände zwischen den Schachbrettecken sind hierbei identisch. In dieser Arbeit wurde ein 6 x 8 Schachbrettmuster verwendet. Der Abstand der Ecken beträgt 3 cm.

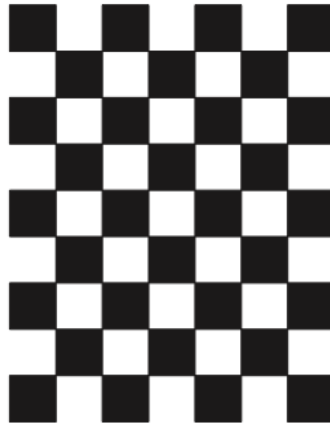


Abbildung 4.2: Schachbrettmuster (6 x 8 innere Ecken)

Für das Verfahren werden mehrere Aufnahmen des Kalibriermusters aus unterschiedlichen Perspektiven benötigt, so dass die Kameraparameter und Verzeichnungskoeffizienten optimal

geschätzt werden können. Theoretisch sind nur zwei Aufnahmen notwendig, jedoch werden durch eine Anzahl von 15-20 Aufnahmen logischerweise bessere Kalibrierergebnisse erzielt.

Die ermittelten extrinsischen Parameter transformieren die Schachbrettecken aus ihrem Koordinatensystem, das den Ursprung in der ersten erkannten Ecke hat, in das Kamerakoordinatensystem und sind demnach an dieser Stelle unbedeutend. Allerdings kann der Rückprojektionsfehler Aufschluss über die Genauigkeit der Kalibrierung geben.

4.1.4 Entfernung der Linsenverzeichnung

Für die 3D-Rekonstruktion müssen die verwendeten Punkte transformiert werden, um die zuvor ermittelte Linsenverzeichnung zu kompensieren. Dies kann zum Beispiel durch ein iteratives Verfahren, wie es in (Fri11, S. 29) Anwendung findet, geschehen.

Um die Funktionsweise der Verzeichnungskompensation zu verdeutlichen, wird im Folgenden kurz das sogenannte *Target-to-Source Mapping* erläutert. Damit kann eine komplette Kameraaufnahme von Störung befreit werden. Da für die 3D-Rekonstruktion jedoch nur einzelne Bildpunkte verwendet werden, würde eine Verwendung dieses Verfahrens einen zu hohen Rechenaufwand mit sich bringen.

4.1.4.1 Target-to-Source Mapping

Target-to-Source Mapping bedeutet, dass zu jedem Pixel in einem Zielbild der korrespondierende Grauwert im Quellbild berechnet wird (Abbildung 4.3).

Hierbei wird das korrigierte, nicht verzeichnete Bild als Zielbild angesehen. Zu jedem Punkt kann durch die Gleichung 4.5 die Quellbildkoordinate berechnet werden. Der Grauwert dieser Koordinate entspricht dem gesuchten Grauwert des Zielbildes. Da die berechnete Quellkoordinate meist einen Subpixel-Wert aufweist, wird der Zielgrauwert durch *bilineare Interpolation* aus den vier Nachbarkoordinaten bestimmt.

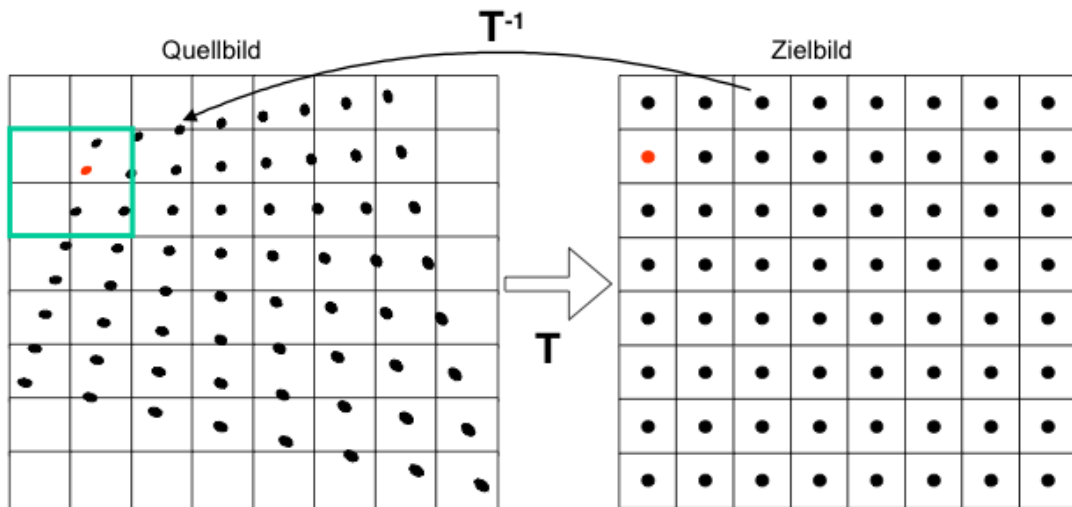


Abbildung 4.3: Target-to-Source Mapping (Mei12)

4.2 Bestimmung der 3D-Weltpunkte

Als Ausgangspunkt für die Bestimmung der dreidimensionalen Weltpunkte eines Objekts dient eine korrespondierende Menge zweidimensionaler Punkte aus zwei perspektivisch unterschiedlichen Kameraaufnahmen. Diese Menge wird im Folgenden durch $\{x_i \leftrightarrow x'_i\}$ bezeichnet, wobei x_i einen Bildpunkt aus der ersten Kameraaufnahme und x'_i den korrespondierenden Bildpunkt aus der zweiten Kameraaufnahme kennzeichnet. Ein projizierter bzw. rekonstruierter 3D-Punkt wird mit X beschrieben. Wird ein einzelnes Paar dieser zusammengehörigen Punkte behandelt, kann das „i“ in der Bezeichnung entfallen.

Das Verfahren zur Extraktion von Punktkorrespondenzen wurde in Kapitel 3 beschrieben. Weiterhin sind die intrinsischen Parameter der Kameraaufnahmen bekannt. Die Linsenverzeichnung der Punkte sollte entfernt worden sein (vgl. Abschnitt 4.1).

Die folgenden Unterkapitel beschreiben die nacheinander auszuführenden Schritte zur Bestimmung der Weltpunkte.

4.2.1 Fundamentalmatrix

Der Zusammenhang zwischen zwei Aufnahmen aus unterschiedlichen Perspektiven wird durch die sogenannte Epipolargeometrie beschrieben (Abbildung 4.4). C und C' stellen die

Kamerazentren dar, \mathbf{x} und \mathbf{x}' die korrespondierenden Punkte und \mathbf{X} den 3D-Punkt, der auf die Bildebenen der Kameras projiziert wurde. \mathbf{C} , \mathbf{C}' und \mathbf{X} spannen die Epipolarebene auf, die die Bildebenen jeweils in ihren Epipolarlinien l und l' schneidet. Die Basislinie ist die Linie durch beide Kamerazentren. Sie schneidet die beiden Bildebenen in ihren Epipolen e und e' .

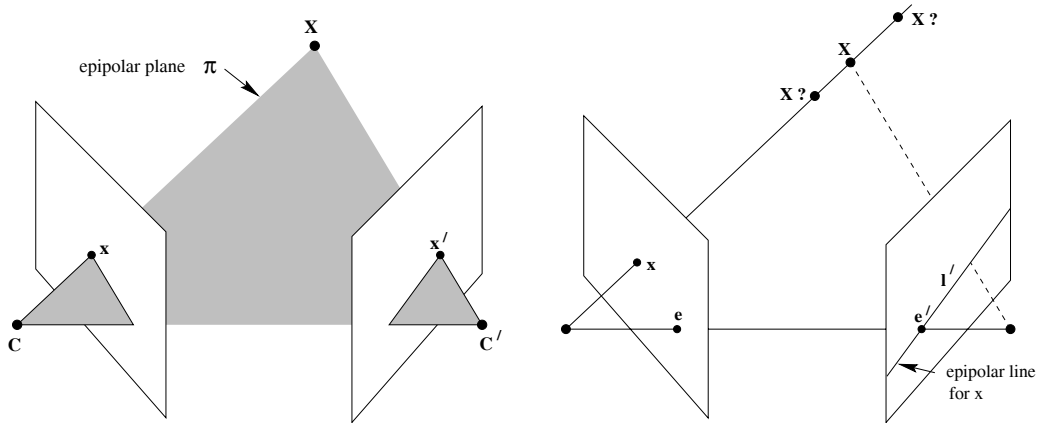


Abbildung 4.4: Epipolargeometrie (HZ04)

Die Fundamentalmatrix \mathbf{F} enthält die Parameter der Epipolargeometrie und ist eine homogene 3×3 Matrix vom Rang 2. Sie ist eine Abbildung $\mathbf{x} \mapsto l'$ und hat die beiden in Gleichung 4.6 und 4.7 dargestellten Eigenschaften.

$$l' = \mathbf{F}\mathbf{x} \quad (4.6)$$

$$l = \mathbf{F}^T \mathbf{x}' \quad (4.7)$$

So kann mit \mathbf{F} zu jedem Bildpunkt \mathbf{x} der ersten Aufnahme eine Linie l' berechnet werden, auf der der korrespondierende Bildpunkt \mathbf{x}' der zweiten Aufnahme liegt.

Da sich der Punkt \mathbf{x}' auf der Epipolarlinie l' befindet, gilt die Gleichung 4.8.

$$\mathbf{x}'^T l' = 0 \quad (4.8)$$

Durch Einsetzen der Gleichung 4.6 in 4.8 folgt die Gleichung 4.9. Diese wird als *Epipolarbedingung* bezeichnet und gilt für alle Punktkorrespondenzen $\mathbf{x} \leftrightarrow \mathbf{x}'$.

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (4.9)$$

4.2.1.1 Berechnung der Fundamentalmatrix

Mit Hilfe der Epipolarbedingung (Gleichung 4.9) kann ein lineares Gleichungssystem zur Berechnung der Fundamentalmatrix aufgestellt werden. Mit $\mathbf{x} = [x, y, 1]$ und $\mathbf{x}' = [x', y', 1]$ erhält man:

$$[x'x, x'y, x', y'x, y'y, x, y, 1] \cdot \mathbf{f} = 0 \quad (4.10)$$

mit

$$\mathbf{f} = [f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}] \quad (4.11)$$

Mit n Punktkorrespondenzen kann das lineare Gleichungssystem aufgestellt werden:

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & x_1 & y_1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & x_n & y_n & 1 \end{bmatrix} \cdot \mathbf{f} = 0 \quad (4.12)$$

Nach (HZ04) kann eine Lösung für \mathbf{F} bis auf eine Skalierung genau bestimmt werden. Dafür muss n mindestens 8 sein, d.h. es müssen mindestens 8 Punktkorrespondenzen vorliegen.

4.2.1.2 RANSAC

RANSAC (Random Sample Consensus) ist ein Verfahren, dass zur automatischen Berechnung der Fundamentalmatrix herangezogen wird. Hierbei wird das Ergebnis robust bestimmt. Außerdem werden mit einer hohen Wahrscheinlichkeit alle Fehlzuordnungen von Korrespondenzen, sogenannte *Outlier*, entdeckt. Das Verfahren ist in (HZ04, Kap. 11) erläutert und wurde ausführlich in (Fri11) behandelt.

4.2.2 Projektionsmatrizen

Die Projektion eines 3D-Welpunktes \mathbf{X} auf einen Punkt \mathbf{x} einer Kameraaufnahme (vgl. Abschnitt 4.1.1) kann durch eine homogene 3×4 Projektionsmatrix \mathbf{P} beschrieben werden. Dabei gilt die Gleichung 4.13.

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (4.13)$$

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \end{bmatrix}}_{\text{Spaltenvektoren}} = \left. \begin{bmatrix} \mathbf{P}^{1T} \\ \mathbf{P}^{2T} \\ \mathbf{P}^{3T} \end{bmatrix} \right\} \text{Reihenvektoren} \quad (4.14)$$

Die Projektionsmatrix \mathbf{P} kann auch anders dargestellt werden, denn sie besteht aus einem Rotationsanteil \mathbf{R} und einem Translationsanteil \mathbf{t} (Gleichung 4.15). \mathbf{R} und \mathbf{t} werden auch als die extrinsischen Kameraparameter bezeichnet. Weiterhin enthält die Projektionsmatrix die intrinsischen Kameraparameter \mathbf{K} (vgl. Abschnitt 4.1).

$$\mathbf{P} = \mathbf{K} \cdot \begin{bmatrix} \underbrace{\mathbf{R}}_{3 \times 3} & \underbrace{\mathbf{t}}_{3 \times 1} \end{bmatrix} \quad (4.15)$$

Eine Projektionsmatrix $\hat{\mathbf{P}}$ einer normalisierten Kamera enthält nicht die instrinsischen Kameraparameter, sondern lediglich den Rotations- und Translationsanteil.

$$\hat{\mathbf{P}} = \mathbf{K}^{-1} \cdot \mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \quad (4.16)$$

Wird ein 3D-Welpunkt \mathbf{X} mit $\hat{\mathbf{P}}$ projiziert, erhält man die normalisierten Koordinaten eines Bildpunktes.

$$\hat{\mathbf{x}} = \hat{\mathbf{P}}\mathbf{X} \quad (4.17)$$

Unter der Berücksichtigung von Gleichung 4.16 kann das Verhältnis zwischen originalen und normalisierten Bildpunkten beschrieben werden.

$$\mathbf{x} = \mathbf{K}\hat{\mathbf{x}} \quad (4.18)$$

Weiterhin sind im Folgenden einige Begriffe aufgelistet und erläutert, die durch eine Projektionsmatrix \mathbf{P} beschrieben werden.

Achsebenen sind beschrieben durch \mathbf{P}^1 und \mathbf{P}^2 und führen durch das Kamerazentrum und die x-Achse bzw. y-Achse der Bildebene. (Abbildung 4.5).

Hauptebene (engl. „principal plane“) ist beschrieben durch \mathbf{P}^3 . Die Hauptebene ist parallel zur Bildebene und führt durch das Kamerazentrum (Abbildung 4.5).

Hauptachse ist die Achse, die durch das Kamerazentrum führt. Der Vektor $\mathbf{v} = \det(\mathbf{R}) \cdot \mathbf{R}^3$ zeigt in Richtung der Vorderseite der Kamera, wobei \mathbf{R}^{3T} die dritte Reihe von \mathbf{R} ist.

Hauptpunkt ist der Punkt, an dem die Hauptachse die Bildebene durchstößt ($\mathbf{x}_0 = \mathbf{R} \cdot \mathbf{R}^3$).

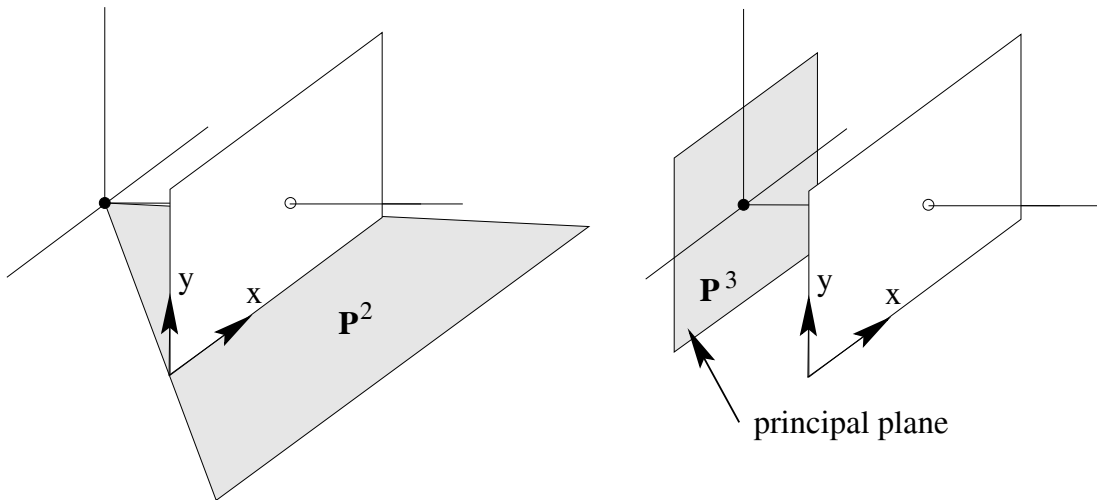


Abbildung 4.5: Kameraebenen - Reihenvektoren einer Projektionsmatrix \mathbf{P} (HZ04)

4.2.3 Bestimmung der Projektionsmatrizen

Um die dreidimensionalen Weltkoordinaten zu den korrespondierenden zweidimensionalen Punkten $\mathbf{x} \leftrightarrow \mathbf{x}'$ zu bestimmen, werden die Projektionsmatrizen der ersten und zweiten Kame-

raufnahme benötigt. Diese können ausgehend von der Fundamentalmatrix (Abschnitt 4.2.1) bestimmt werden.

An dieser Stelle wird der Ursprung des dreidimensionalen Weltkoordinatensystems im Zentrum der ersten Kamera festgelegt. Es entspricht also dem Kamerakoordinatensystem. Demnach ist die Ermittlung der extrinsischen Kameraparameter trivial, denn bei der Transformation von Welt- in Kamerakoordinaten findet weder eine Rotation noch eine Translation statt. Für die normalisierte Projektionsmatrix $\hat{\mathbf{P}}_1$ der ersten Kamera ergibt sich somit die Gleichung 4.19.

$$\hat{\mathbf{P}}_1 = \begin{bmatrix} \underbrace{\mathbf{I}}_{\mathbf{R}} & \underbrace{\mathbf{0}}_{\mathbf{t}} \\ \hline & \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.19)$$

Genau genommen muss also nur die normalisierte Projektionsmatrix $\hat{\mathbf{P}}_2$ der zweiten Kamera bestimmt werden. Diese kann ausgehend von der Fundamentalmatrix \mathbf{F} bestimmt werden. Die Schritte hierfür sind im Folgenden beschrieben.

Die Epipolarbedingung aus Gleichung 4.9 kann auch für normalisierte Koordinaten formuliert werden. Diese sogenannte *Essentialmatrix* \mathbf{E} ist die Fundamentalmatrix der normalisierten Kameras.

$$\hat{\mathbf{x}}'^T \mathbf{E} \hat{\mathbf{x}} = 0 \quad (4.20)$$

Die Essentialmatrix kann aus der Fundamentalmatrix berechnet werden (Gleichung 4.21). Herleitung:

$$\begin{aligned} \hat{\mathbf{x}}'^T \mathbf{E} \hat{\mathbf{x}} &= 0 \\ \mathbf{x}'^T \cdot \underbrace{(\mathbf{K}^{-1})^T \cdot \mathbf{E} \cdot \mathbf{K}^{-1}}_{\mathbf{F}} \cdot \mathbf{x} &= 0 \\ (\mathbf{K}^{-1})^T \cdot \mathbf{E} \cdot \mathbf{K}^{-1} &= \mathbf{F} \\ \mathbf{E} &= \mathbf{K}^T \cdot \mathbf{F} \cdot \mathbf{K} \end{aligned} \quad (4.21)$$

Weiterhin kann die Essentialmatrix \mathbf{E} in einen Rotationsanteil \mathbf{R} und einen Translationsanteil \mathbf{t} zerlegt werden. Hierfür wird eine Singulärwertzerlegung auf \mathbf{E} angewendet. Als Ergebnis

erhält man die drei Matrizen \mathbf{U} , \mathbf{S} und \mathbf{V} . Nach (HZ04) existieren unter der Zuhilfenahme der Matrix \mathbf{W} je zwei Lösungen für \mathbf{R} und \mathbf{t} .

$$\text{SVD}(\mathbf{E}) = \mathbf{USV}^T \quad (4.22)$$

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.23)$$

Der dritte Reihenvektor \mathbf{u}_3 von \mathbf{U} ist die Lösung für den Translationsanteil. Allerdings ist die Richtung des Vektors noch unklar. Auch für den Rotationsanteil gibt es zwei Lösungen. Diese unterscheiden sich in der Transponierten der Hilfsmatrix \mathbf{W} .

Damit existieren vier mathematische Lösungen für die normalisierte Projektionsmatrix $\hat{\mathbf{P}}_2$.

$$\hat{\mathbf{P}}_2 = \begin{bmatrix} \mathbf{U}\mathbf{W}\mathbf{V}^T & +\mathbf{u}_3 \end{bmatrix} \quad \text{oder} \quad (4.24)$$

$$\hat{\mathbf{P}}_2 = \begin{bmatrix} \mathbf{U}\mathbf{W}\mathbf{V}^T & -\mathbf{u}_3 \end{bmatrix} \quad \text{oder} \quad (4.25)$$

$$\hat{\mathbf{P}}_2 = \begin{bmatrix} \mathbf{U}\mathbf{W}^T\mathbf{V}^T & +\mathbf{u}_3 \end{bmatrix} \quad \text{oder} \quad (4.26)$$

$$\hat{\mathbf{P}}_2 = \underbrace{\begin{bmatrix} \mathbf{U}\mathbf{W}^T\mathbf{V}^T & -\mathbf{u}_3 \end{bmatrix}}_{\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}} \quad (4.27)$$

Es existiert jedoch nur eine plausible Lösung, denn nur in einem der vier Fälle befindet sich der projizierte 3D-Weltpunkt vor den Zentren beider Kameras. Dieser Sachverhalt ist in Abbildung 4.6 dargestellt. Hierbei stellt \mathbf{A} das Zentrum der ersten und \mathbf{B} das Zentrum der zweiten Kamera dar. Lediglich im Fall (a) liegt der projizierte Weltpunkt vor beiden Kamerazentren.

Um mathematisch zu bestimmen, welche dieser vier Lösungen für $\hat{\mathbf{P}}_2$ die einzig plausible ist, muss ein 3D-Weltpunkt \mathbf{X} bekannt sein. Damit kann die Tiefe des Punktes \mathbf{X} vor der Hauptebene der Kamera bestimmt werden, sodass entschieden werden kann, ob ein Punkt vor dem Kamerazentrum liegt. Mit dem 3D-Punkt $\mathbf{X} = [X, Y, Z, T]$, dem projiziertem 2D-Punkt

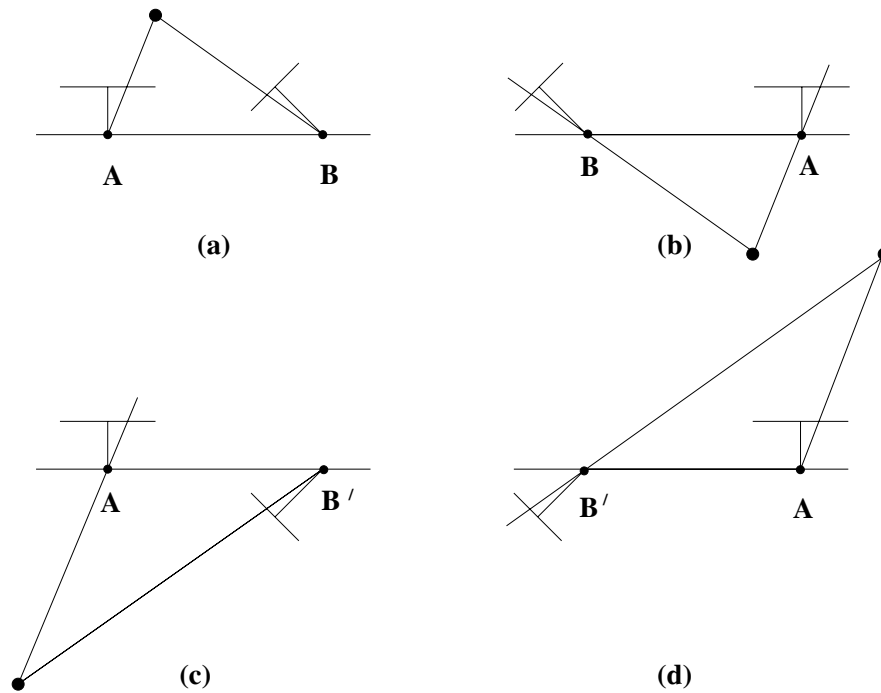


Abbildung 4.6: Vier Möglichkeiten für $\hat{\mathbf{P}}_2$ (HZ04)

$\mathbf{x} = [x, y, w]$ in homogenen Koordinaten und der Gleichung $\mathbf{x} = \mathbf{P}\mathbf{X}$, gilt für die Tiefe nach (HZ04) die Gleichung 4.28.

$$depth(\mathbf{X}; \mathbf{P}) = \frac{sign(det(\mathbf{R}))w}{T\|\mathbf{R}^3\|} \quad (4.28)$$

Für normalisierte Kameras ist $det(\mathbf{R}) > 0$ und $\|\mathbf{R}^3\| = 1$. Damit gilt für die Tiefe ebenfalls die Gleichung 4.29.

$$depth(\mathbf{X}; \hat{\mathbf{P}}) = \frac{w}{T} \quad (4.29)$$

In nur einem der vier Fälle ist sowohl $depth(\mathbf{X}; \hat{\mathbf{P}}_1) > 0$ als auch $depth(\mathbf{X}; \hat{\mathbf{P}}_2) > 0$, sodass ist die richtige normalisierte Projektionsmatrix $\hat{\mathbf{P}}_2$ gefunden werden kann.

Der Weltpunkt \mathbf{X} kann durch Triangulation (Abschnitt 4.2.4) bestimmt werden. Da es vorkommen kann, dass einige Weltpunkte falsch trianguliert werden oder Fehlzuordnungen bei den 2D-Korrespondenzen existieren, sollte die Probe mit allen Weltpunkten durchgeführt werden.

4.2.4 Triangulation der Weltpunkte

An diesem Punkt sind sowohl die korrespondierende Menge zweidimensionaler Punkte $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ als auch die Projektionsmatrizen \mathbf{P}_1 und \mathbf{P}_2 (vier Möglichkeiten, vgl. Abschnitt 4.2.3) bekannt.

Mittels der linearen Triangulation kann der 3D-Weltpunkt \mathbf{X} zu den korrespondierenden Bildpunkten $\mathbf{x} = (x, y)^T$ und $\mathbf{x}' = (x', y')^T$ bestimmt werden. Hierfür wird ein lineares Gleichungssystem $\mathbf{A}\mathbf{X} = 0$ aufgestellt. Dabei sind \mathbf{P}^{iT} die Reihen von \mathbf{P} (vgl. Gleichung 4.14, S. 25).

$$\mathbf{A} = \begin{bmatrix} x\mathbf{P}_1^{3T} & -\mathbf{P}_1^{1T} \\ y\mathbf{P}_1^{3T} & -\mathbf{P}_1^{2T} \\ x'\mathbf{P}_2^{3T} & -\mathbf{P}_2^{1T} \\ y'\mathbf{P}_2^{3T} & -\mathbf{P}_2^{2T} \end{bmatrix} \quad (4.30)$$

Die Lösung erhält man mittels einer Singulärwertzerlegung auf \mathbf{A} .

$$\text{SVD}(\mathbf{A}) = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (4.31)$$

Der 3D-Weltpunkt $\mathbf{X} = (X, Y, Z, W)$ in homogenen Koordinaten ist der letzte Spaltenvektor von \mathbf{V} .

Da allerdings die gemessenen Punkte der 2D-Korrespondenzmenge fehlerbehaftet sind, erfüllt kein 3D-Punkt \mathbf{X} exakt die Gleichungen $\mathbf{x} = \mathbf{P}_1\mathbf{X}$ und $\mathbf{x}' = \mathbf{P}_2\mathbf{X}$. Dies bedeutet, dass die Strahlen durch die Kamerazentren \mathbf{C} bzw. \mathbf{C}' und den rückprojizierten Punkt \mathbf{x} bzw. \mathbf{x}' versetzt verlaufen. Sie schneiden sich nicht in dem eigentlich gemeinsamen Punkt \mathbf{X} (Abbildung 4.7).

Dies ist gleichbedeutend damit, dass die Bildpunkte die Epipolarbedingung $\mathbf{x}^T\mathbf{F}\mathbf{x} = 0$ nicht exakt erfüllen. Deshalb muss die Menge der Punktkorrespondenzen $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ in Abhängigkeit von der Epipolarbedingung korrigiert werden. Dies geschieht durch eine Minimierung des geometrischen Fehlers (Gleichung 4.32).

$$d(\mathbf{x}_i, \tilde{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \tilde{\mathbf{x}}'_i)^2 \quad (4.32)$$

$d(\mathbf{a}, \mathbf{b})$ stellt den euklidischen Abstand zwischen zwei Punkten dar. Die korrigierten Korrespondenzen $\{\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}'_i\}$ erfüllen hierbei die Epipolarbedingung nahezu exakt. In (HZ04, S. 318)

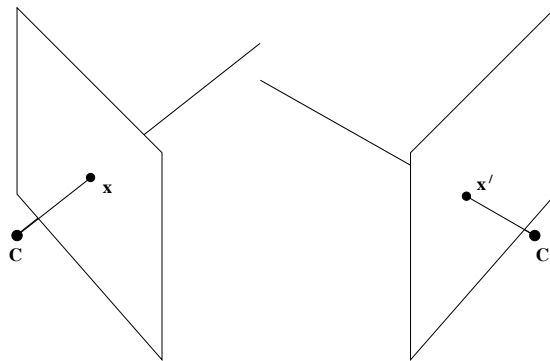


Abbildung 4.7: Strahlen durch rückprojizierte Punkte verlaufen versetzt (HZ04)

ist der Algorithmus für die sogenannte „optimale Methode“ zur Korrektur der Korrespondenzen aufgeführt, der auch in dieser Arbeit Anwendung findet.

Die Triangulation kann nun mit den korrigierten Korrespondenzen und je einer Möglichkeit für \mathbf{P}_2 vollzogen werden. Als Resultat erhält man vier Mengen an triangulierten Punkten. Mit diesen Punkten kann nun festgestellt werden, welches die richtige Projektionsmatrix \mathbf{P}_2 und damit die richtige Menge an 3D-Weltpunkten ist (vgl. Abschnitt 4.2.3).

4.2.5 Verifikation der Weltpunkte

An diesem Punkt ist die Menge der triangulierten 3D-Weltpunkte $\{\mathbf{X}_i\}$ bekannt. Diese Weltpunkte müssen nun verifiziert werden. Dabei müssen die Punkte aussortiert werden, die bestimmte Kriterien nicht erfüllen. Auszusortierende Weltpunkte werden im Folgenden auch als *Outlier* bezeichnet. Dieser Abschnitt behandelt die verwendeten Verifikations- und Filterungsmaßnahmen.

4.2.5.1 Untersuchung bezüglich eines Gültigkeitsbereichs

Zunächst werden die Weltpunkte bezüglich eines Gültigkeitsbereichs untersucht. Hier werden falsch triangulierte Ausreißer aussortiert, die ungewöhnlich weit entfernt von den restlichen Punkten liegen.

Der Gültigkeitsbereich wird definiert durch den maximal erlaubten Abstand d_{valid} eines Weltpunktes zum Schwerpunkt der Menge.

$$d_{valid} = \alpha \cdot d_{arith} = \alpha \cdot \frac{1}{n} \cdot \sum_{i=1}^n d(\mathbf{X}_i^{\text{sort}}, \tilde{\mathbf{X}}) \quad (4.33)$$

Hierbei dient der Median $\tilde{\mathbf{X}}$ der sortierten Weltpunktmenge $\{\mathbf{X}_i^{\text{sort}}\}$ als Schwerpunkt. Weiterhin wird der durchschnittliche euklidische Abstand (arithmetisches Mittel) d_{arith} der Weltpunkte $\mathbf{X}_i^{\text{sort}}$ zum Schwerpunkt berechnet. Um einen Faktor α verstärkt, beschreibt dieser den maximal erlaubten Abstand d_{valid} .

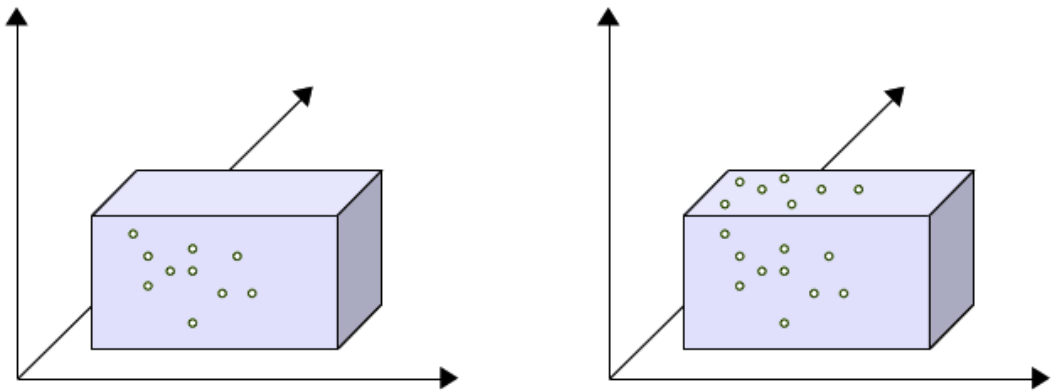
Befindet sich einer der triangulierten Weltpunkte nicht innerhalb des Gültigkeitsbereichs, wird er als *Outlier* gekennzeichnet. Übersteigt die Anzahl der *Outlier* einen prozentualen Schwellwert, wurden zu viele Punkte falsch trianguliert und die Weltpunktmenge muss verworfen werden.

4.2.5.2 Erkennung einer koplanaren Weltpunktmenge

Es gibt zwei Situationen, in denen die Fundamentalmatrix (Abschnitt 4.2.1) nicht eindeutig berechnet werden kann. Die Epipolargeometrie ist für diese Fälle nicht definiert, sodass die 3D-Punkte nicht rekonstruiert werden können, da keine eindeutige Information über die Tiefe der Punkte vorliegt. Diese Situationen werden auch als *degeneriert* bezeichnet.

Die erste Situation entsteht, wenn zwischen der ersten und der zweiten Kameraaufnahme lediglich eine Rotation um das Kamerazentrum, d.h. keine Translation, stattgefunden hat. Durch eine Translation $\neq 0$ bei der Aufnahme der Bilder kann diese sogenannte *degenerierte Bewegung* ausgeschlossen werden. Die zweite Situation entsteht, wenn die triangulierte 3D-Weltpunktmenge koplanar ist. Koplanar bedeutet, dass alle Weltpunkte auf einer Ebene liegen, also linear abhängig sind. Dieser Umstand wird auch als *degenerierte Struktur* bezeichnet und in Abbildung 4.8 veranschaulicht.

Das in (Fri11) vorgestellte Verfahren erkennt eine näherungsweise koplanare Weltpunktmenge zuverlässig und ist im Folgenden aufgelistet. Die Weltpunkte liegen in euklidischen Koordinaten vor ($\mathbf{X}_i = (X_i, Y_i, Z_i)^T$).



(a) Alle Welpunkte auf derselben Ebene (b) Welpunkte auf unterschiedlichen Ebenen

Abbildung 4.8: Koplanare (a) und nicht koplanare Welpunkte (b)

1. Der arithmetische Mittelwert \mathbf{X}_{arith} (3 x 1 Vektor) der Welpunktmenge $\{\mathbf{X}_i\}$ wird berechnet.

$$\mathbf{X}_{arith} = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i \quad (4.34)$$

2. Die Varianz σ^2 (3 x 3 Matrix) von $\{\mathbf{X}_i\}$ wird berechnet.

$$\begin{aligned} \sigma &= \mathbf{X}_{3 \times n} - \mathbf{X}_{arith} \cdot [\mathbf{1}, \dots, \mathbf{1}]_{1 \times n} \\ \sigma^2 &= \sigma \cdot \sigma^T \end{aligned} \quad (4.35)$$

3. Auf die Varianz wird eine Singulärwertzerlegung angewendet.

$$\text{SVD}(\sigma^2) = \mathbf{U} \mathbf{S} \mathbf{V}^T = \mathbf{U} \cdot \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \cdot \mathbf{V}^T \quad (4.36)$$

4. Ist der Singulärwert d_3 im Verhältnis zum Singulärwert d_2 sehr klein, ist die Weltpunktmenge näherungsweise koplanar.

$$d_3 \ll d_2 \Rightarrow \{\mathbf{X}_i\} \text{ ist näherungsweise koplanar} \quad (4.37)$$

Ist die Weltpunktmenge näherungsweise koplanar, muss sie verworfen werden.

4.2.5.3 Filterung bezüglich des Rückprojektionsfehlers

Als letzter Schritt der Verifikation werden die triangulierten Weltpunkte rückprojiziert. Für den Rückprojektionsfehler r_{err} gilt der euklidische Abstand zwischen den ursprünglichen Bildpunkten und den rückprojizierten Punkten.

$$\begin{aligned} r_{err} &= d(\mathbf{x}, \mathbf{P}_1 \mathbf{X}) \\ r'_{err} &= d(\mathbf{x}', \mathbf{P}_2 \mathbf{X}) \end{aligned} \quad (4.38)$$

Es wird der Rückprojektionsfehler jedes Weltpunktes für die Projektionsmatrizen beider Kameras \mathbf{P}_1 und \mathbf{P}_2 berechnet. Weist ein Weltpunkt \mathbf{X} einen zu großen Rückprojektionsfehler auf, wird er aussortiert bzw. als *Outlier* markiert.

4.2.6 Ermittlung des Skalierungsfaktors

Da die Fundamentalmatrix (Abschnitt 4.2.1) nur bis auf eine Skalierung genau bestimmt werden kann, gilt dies auch für die Projektionsmatrizen und damit auch für die triangulierten 3D-Weltpunkte. Deshalb muss die tatsächliche Skalierung ermittelt und korrigiert werden.

An diesem Punkt trennen sich die Wege der initialen Bestimmung und der Erweiterung einer 3D-Weltpunktmenge. Sind schon Weltpunkte bekannt, so kann eine Transformation zwischen den Koordinatensystemen der bereits bekannten und der neuen Weltpunkte ermittelt werden, durch die die neuen Weltpunkte auch richtig skaliert werden. Die Berechnung einer solchen Transformation wird im nächsten Abschnitt 4.2.7 (S. 36) beschrieben. Dieser Abschnitt hingegen behandelt die Ermittlung des Skalierungsfaktors bei der initialen Bestimmung der 3D-Weltpunkte.

Der Skalierungsfaktor kann bestimmt werden, wenn der tatsächliche Abstand d_{known} zwischen zwei verschiedenen Weltpunkten \mathbf{X}_i und \mathbf{X}_j bekannt ist. Das Verhältnis dieses Abstandes zu dem Abstand der entsprechenden triangulierten Weltpunkte d_{tria} ergibt den gesuchten Skalierungsfaktor s .

$$s = \frac{d_{known}}{d_{tria}} \quad (4.39)$$

Sind n tatsächliche Abstände bekannt, kann der Skalierungsfaktor durch das arithmetische Mittel der Abstandsverhältnisse bestimmt werden.

$$s = \frac{1}{n} \sum_{i=1}^n \frac{d_{known_i}}{d_{tria_i}} \quad (4.40)$$

Um den Skalierungsfaktor berechnen zu können, müssen also tatsächliche Abstände bekannt sein. Das kann durch die Verwendung eines Musters erreicht werden, auf dem koplanare Merkmale leicht und robust wiederzuerkennen sind. Die Abstände zwischen diesen Merkmalen müssen bekannt sein. In dieser Arbeit wird das Schachbrettmuster aus Abschnitt 4.1 verwendet, da es sehr einfach ist, die Schachbrettecken auf einer Kameraaufnahme zu erkennen.

Weiterhin müssen die erkannten Korrespondenzen der Schachbrettecken $\{\mathbf{x}_{ci} \leftrightarrow \mathbf{x}'_{ci}\}$ den Prozess der Triangulation und Verifikation zusammen mit den Merkmalskorrespondenzen des Objekts $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ durchlaufen. Mit dem euklidischen Abstand d_{tria} der triangulierten Schachbrettecken

$$d_{tria} = d(\mathbf{X}_{ci}, \mathbf{X}_{cj}) \quad (4.41)$$

kann der Skalierungsfaktor s nach der Gleichung 4.40 berechnet werden. Damit können zum einen die triangulierte Weltpunktmenge $\{\mathbf{X}_i\}$ und zum anderen die Projektionsmatrizen \mathbf{P}_1 und \mathbf{P}_2 richtig skaliert werden. Bei den Projektionsmatrizen muss lediglich der vierte Spaltenvektor, der Translationsanteil, skaliert werden.

$$\mathbf{X}'_i = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{X}_i$$
$$\mathbf{P}' = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & s \cdot \mathbf{p}_4 \end{bmatrix} \quad (4.42)$$

4.2.7 Erweiterung der 3D-Welpunktmenge

Ist bereits eine 3D-Welpunktmenge $\{\mathbf{X}_i\}$ eines Objekts bekannt, so kann sie durch Welpunkte aus weiteren Kameraaufnahmen des Objekts erweitert werden. Hierfür wird eine neue Menge an korrespondierenden 2D-Punkten $\{y_i \leftrightarrow y'_i\}$ aus zwei perspektivisch unterschiedlichen Kameraaufnahmen benötigt. Diese muss alle bis hierhin beschriebenen Schritte, mit Ausnahme der Ermittlung des Skalierungsfaktors, durchlaufen, so dass eine neue Welpunktmenge $\{\mathbf{Y}_i\}$ vorliegt.

Das Koordinatensystem der neuen Welpunktmenge hat seinen Ursprung im Kamerazentrum der ersten für die Ermittlung verwendeten Aufnahme. Da die neuen Welpunkte vollkommen unabhängig von den bereits existierenden bestimmt worden sind, muss nun eine Transformation gefunden werden, die jeden Welpunkt \mathbf{Y}_i in das Koordinatensystem der existierenden Welpunktmenge $\{\mathbf{X}_i\}$ transformiert. Das schließt die Skalierung der Welpunkte mit ein. Ist zumindest eine Teilmenge der neu rekonstruierten Welpunkte bereits in $\{\mathbf{X}_i\}$ vorhanden, kann eine solche Transformation berechnet werden. Da zu jedem Welpunkt \mathbf{X}_i und \mathbf{Y}_i die Merkmalsdeskriptoren bekannt sind, ist die Aufstellung der für die Berechnung notwendigen Korrespondenzmenge $\{\mathbf{X}_i \leftrightarrow \mathbf{Y}_i\}$ möglich.

4.2.7.1 Bestimmung einer Transformation aus 3D-Punktkorrespondenzen

Gegeben ist eine Menge korrespondierender 3D-Welpunkte $\{\mathbf{X}_i \leftrightarrow \mathbf{Y}_i\}$. Es soll eine Transformation gefunden werden, die jeden Welpunkt \mathbf{Y}_i in das Koordinatensystem der Welpunktmenge $\{\mathbf{X}_i\}$ transformiert.

$$\mathbf{X}_i = s \cdot \mathbf{R} \cdot \mathbf{Y}_i + \mathbf{t} \quad (4.43)$$

\mathbf{R} beschreibt dabei eine Rotation, \mathbf{t} eine Translation und s eine Skalierung. Die Transformation kann auch in homogenen Koordinaten beschrieben werden.

$$\mathbf{T} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \quad (4.44)$$

In (HHN88) und (ELF97) wird beschrieben, wie aus einer Menge korrespondierender Punkte eine optimale Transformation ermittelt werden kann. Die einzelnen Berechnungsschritte des Verfahrens sind im Folgenden aufgelistet.

1. Berechnung des arithmetischen Mittels $\mathbf{X}_{\text{arith}}$ und $\mathbf{Y}_{\text{arith}}$. Die Mittelwerte bestimmen die Schwerpunkte der jeweiligen Weltpunktmenge.

$$\begin{aligned} \mathbf{X}_{\text{arith}} &= \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i \\ \mathbf{Y}_{\text{arith}} &= \frac{1}{n} \sum_{i=1}^n \mathbf{Y}_i \end{aligned} \quad (4.45)$$

2. Normalisierung der Weltpunktmenge bezüglich ihres Schwerpunkts.

$$\begin{aligned} \hat{\mathbf{X}}_i &= \mathbf{X}_i - \mathbf{X}_{\text{arith}} \\ \hat{\mathbf{Y}}_i &= \mathbf{Y}_i - \mathbf{Y}_{\text{arith}} \end{aligned} \quad (4.46)$$

3. Die Skalierung s kann berechnet werden.

$$s = \sqrt{\frac{\sum_{i=1}^n \|\hat{\mathbf{X}}_i\|^2}{\sum_{i=1}^n \|\hat{\mathbf{Y}}_i\|^2}} \quad (4.47)$$

4. Die normalisierten Korrespondenzen werden miteinander multipliziert und aufsummiert. Das resultiert in einer 3 x 3 Matrix \mathbf{M} .

$$\mathbf{M} = \sum_{i=1}^n \hat{\mathbf{X}}_i \cdot \hat{\mathbf{Y}}_i^T \quad (4.48)$$

5. Auf die Matrix \mathbf{M} wird eine Singulärwertzerlegung angewendet.

$$\text{SVD}(\mathbf{M}) = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (4.49)$$

6. Die 3 x 3 Rotationsmatrix \mathbf{R} der Transformation kann aus dem Ergebnis der Singulärwertzerlegung mit $\mathbf{U}\mathbf{V}^T$ berechnet werden. Die Determinante $\det(\mathbf{U}\mathbf{V}^T)$ korrigiert ggf. das Vorzeichen der Rotation.

$$\mathbf{R} = \mathbf{U} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U}\mathbf{V}^T) \end{pmatrix} \mathbf{V}^T \quad (4.50)$$

7. Durch eine Umstellung der Transformationsgleichung kann der 3 x 1 Translationsvektor \mathbf{t} bestimmt werden. Als Weltpunkte werden hier die Schwerpunkte der jeweiligen Menge eingesetzt.

$$\mathbf{t} = \mathbf{X}_{\text{arith}} - s \cdot \mathbf{R} \cdot \mathbf{Y}_{\text{arith}} \quad (4.51)$$

4.2.7.2 Integration der neuen Weltpunkte

Die neue Weltpunktmenge $\{\mathbf{Y}_i\}$ kann jetzt mit der errechneten homogenen Transformationsmatrix \mathbf{T} aus Gleichung 4.44 (S. 37) in die bestehende Weltpunktmenge $\{\mathbf{X}_i\}$ integriert werden. Dafür wird jeder Punkt der neuen Weltpunktmenge in das Koordinatensystem der bestehenden Weltpunktmenge transformiert.

$$\tilde{\mathbf{Y}}_i = \mathbf{T}\mathbf{Y}_i \quad (4.52)$$

Daraufhin wird der durchschnittliche euklidische Abstand der zuvor aufgestellten korrespondierenden Weltpunktmenge $\{\mathbf{X}_i \leftrightarrow \tilde{\mathbf{Y}}_i\}$ bestimmt.

$$d_{\text{arith}} = \frac{1}{m} \sum_{i=1}^m d(\mathbf{X}_i, \tilde{\mathbf{Y}}_i) \quad (4.53)$$

Übersteigt d_{arith} einen bestimmten Schwellwert, ist die Transformation nicht genau genug berechnet worden und die neue Weltpunktmenge muss verworfen werden. Andernfalls können die neuen, transformierten Weltpunkte $\tilde{\mathbf{Y}}_i$ in die bereits bekannte Menge $\{\mathbf{X}_i\}$ integriert werden. Den in der Korrespondenzmenge $\{\mathbf{X}_i \leftrightarrow \tilde{\mathbf{Y}}_i\}$ enthaltenen Punkten werden lediglich die neuen Deskriptoren zugeordnet. Alle anderen Punkte ergänzen die Menge gemeinsam mit ihren beiden Deskriptoren.

5 Implementierung

Dieses Kapitel beschreibt den Aufbau und die Umsetzung der in dieser Arbeit entwickelten Software. Ziel der Software ist die Erstellung eines Modells von 3D-Welpunkten, das die dreidimensionale Lage eines Objekts beschreibt. Ausgehend von zwei Kameraaufnahmen eines Objekts werden durch Bildmerkmale 2D-Korrespondenzen in diesen Aufnahmen ermittelt. Daraus werden dann die zugehörigen 3D-Welpunkte rekonstruiert. Durch weitere Kameraaufnahmen desselben Objekts kann die Menge der Welpunkte erweitert werden. Über die Deskriptoren der Bildmerkmale, die den rekonstruierten Punkten zugeordnet werden, ist eine Identifikation der Welpunkte und somit des Objekts möglich.

Die Software ist für ROS (vgl. Abschnitt 2.2) programmiert, sodass der Betrieb auf allen mit ROS ausgestatteten Robotern oder Computern möglich ist.

5.1 Architektur

Die Software gliedert sich in vier Komponenten (Abbildung 5.1). In der folgenden Auflistung ist ein einleitender Überblick über diese gegeben. Die darauf folgenden Unterabschnitte beschreiben die Komponenten detailliert.

Model3D ist das Hauptprogramm und der zu startende ROS-Knoten. Dieser dient der Integration in das ROS-Framework und der Ablaufsteuerung des implementierten Verfahrens. Er abonniert ein bestimmtes ROS-Topic, über das Kameraaufnahmen eines Objekts veröffentlicht werden. Dabei wird auf eingehende Bilder gewartet, um diese zu verarbeiten. Sind genügend Bilder empfangen worden, werden die Verfahren zur Merkmalsextraktion und 3D-Rekonstruktion gestartet.

FeatureExtractor ist die für die Merkmalsextraktion zuständige Komponente. Sie bietet die Nutzung der in Kapitel 3 behandelten Verfahren zur Aufstellung einer Korrespondenzmenge von 2D-Bildpunkten in perspektivisch unterschiedlichen Kameraaufnahmen. Für die meisten Teilfunktionen kann auf die Softwarebibliothek OpenCV zurückgegriffen

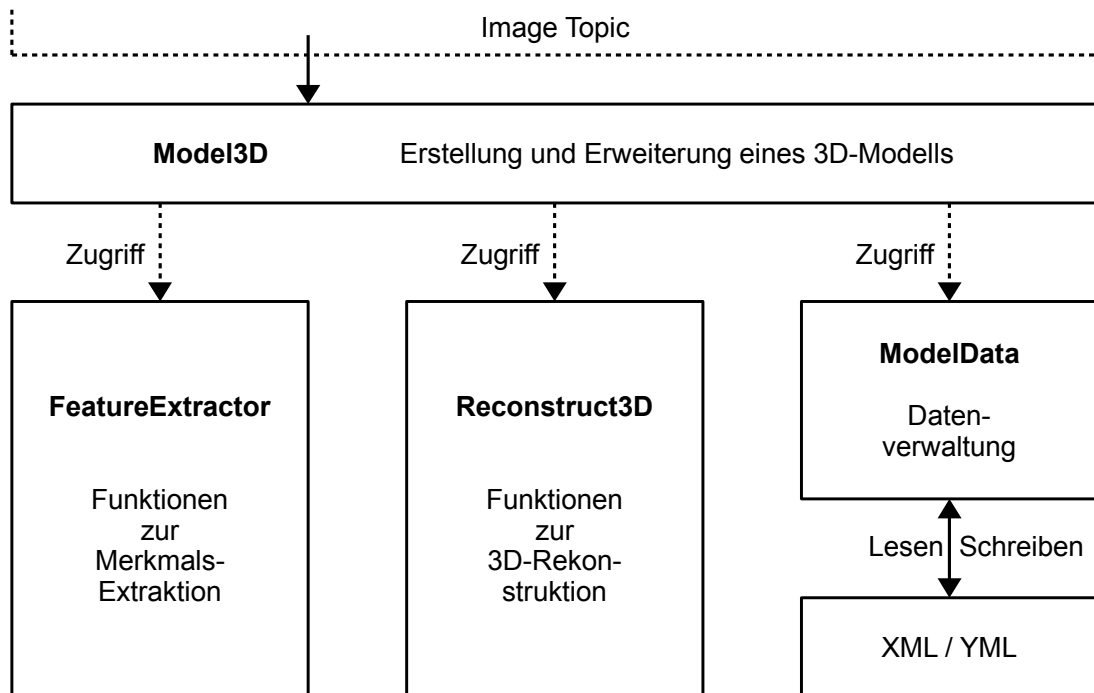


Abbildung 5.1: Überblick über die Software-Komponenten

werden. Es wird eine Funktion zur Ausführung des gesamten Prozesses zur Verfügung gestellt.

Reconstruct3D ist die für die 3D-Rekonstruktion zuständige Komponente. Sie implementiert das in Kapitel 4 behandelte Verfahren zur 3D-Rekonstruktion von korrespondierenden Bildpunkten. Es wird eine Funktion zur Ausführung des gesamten Prozesses zur Verfügung gestellt. Die einzelnen Teilfunktionen sind ebenfalls verfügbar.

ModelData dient der Verwaltung der einem Objekt zugehörigen Weltpunkte und Deskriptoren. Zur Speicherung der Daten dient wahlweise eine .xml- oder .yml-Datei.

Neben diesen Komponenten ist weiterhin ein Treiberprogramm für die verwendete Kamera (Abschnitt 2.1) notwendig, das die vom Hersteller bereitgestellten Programmierschnittstellen nutzt, um die Bilder der Kamera an einem ROS-Topic zu veröffentlichen.

5.1.1 Hauptprogramm

Dieser Abschnitt beschreibt den Aufbau und den Ablauf des Hauptprogramms der Software (Model3D). Es ist der ROS-Knoten, der gestartet werden muss, damit die Erstellung oder Erweiterung einer Weltpunktmenge eines Objekts möglich ist. Der gesamte Ablauf des Verfahrens und der dafür nötigen Funktions- und Datenzugriffe wird durch diese Komponente gesteuert.

Bei Programmstart meldet sich der Knoten an einem Topic an, über das die zu verwendenden Kameraaufnahmen empfangen werden. So kann beispielsweise ein Roboter, auf dem eine Kamera montiert ist, verschiedene Aufnahmen von einem Gegenstand machen und über das Topic veröffentlichen. Weiterhin wird der Name des Objekts, dessen Weltpunkte bestimmt werden sollen, angegeben. Die Datei zur Speicherung der einem Objekt zugehörigen Modelldaten hat den gleichen Namen. Die Daten werden, soweit vorhanden, aus der Datei gelesen und in einem ModelData-Objekt verwaltet. Ist keine Datei vorhanden, wird diese erstellt.

An dieser Stelle beginnt der Ablauf des Verfahrens zur Erstellung oder Erweiterung eines 3D-Modells. Die Schritte sind im Folgenden aufgeführt. Außerdem wird die Arbeitsweise in [Abbildung 5.2](#) auf [Seite 43](#) veranschaulicht.

1. Zunächst wird entschieden, ob der Prozess der initialen Bestimmung der Weltpunkte abgearbeitet oder eine bereits vorhandene Weltpunktmenge erweitert werden soll. Da der Zugriff auf die Modelldaten gewährleistet ist, kann einfach überprüft werden, ob bereits 3D-Weltpunkte zu dem betrachteten Objekt vorhanden sind.
2. Soll eine initiale Bestimmung der Weltpunkte eines Objekts durchgeführt werden, so werden fünf Kameraaufnahmen erwartet. Neben drei Aufnahmen von einem Gegenstand werden auch Aufnahmen des verwendeten Musters (Schachbrettmuster) erwartet. Dies ist notwendig zur Bestimmung des Skalierungsfaktors der Weltpunkte (vgl. [Abschnitt 4.2.6](#)). Da die Aufnahmen des Musters jeweils aus derselben Perspektive wie das betrachtete Objekt aufgenommen werden müssen, werden die Bilder in der folgenden Reihenfolge erwartet:
 - a) Schachbrettmuster aus Perspektive 1
 - b) Objekt aus Perspektive 1
 - c) Objekt aus Perspektive 2
 - d) Schachbrettmuster aus Perspektive 2
 - e) Objekt aus Perspektive 3

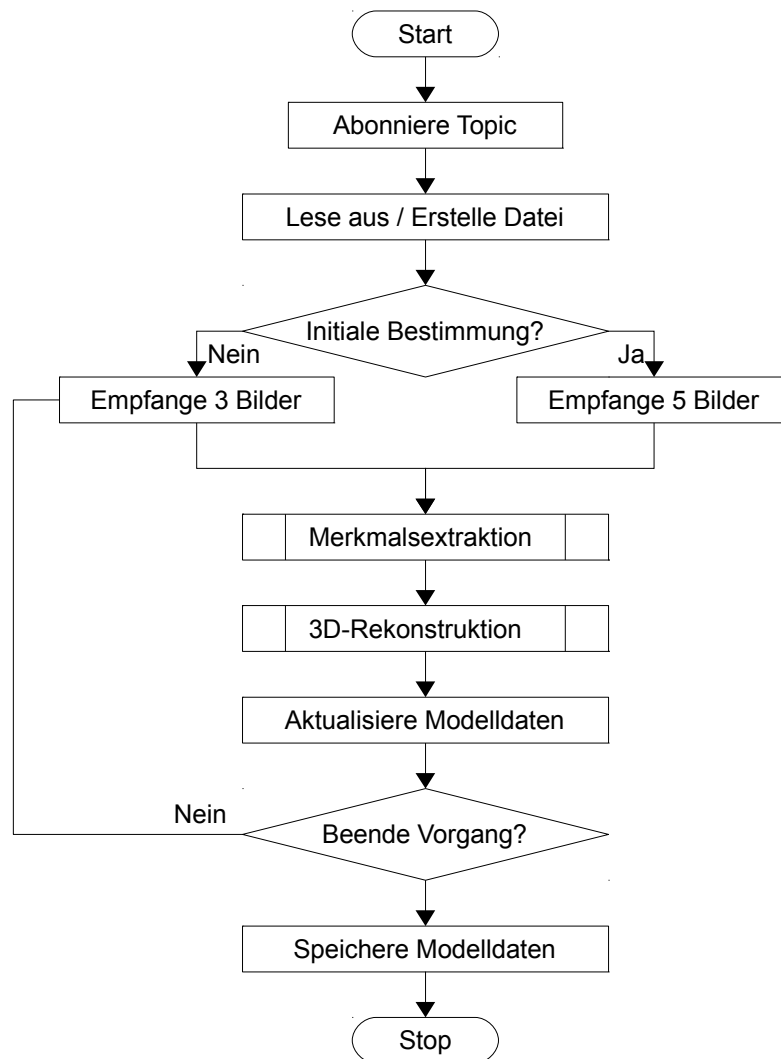


Abbildung 5.2: Ablauf des Hauptprogramms

Sind bereits Weltpunkte des betrachteten Objekts bekannt, werden nur drei Kameraaufnahmen erwartet. Die Aufnahme aus der dritten Perspektive wird benötigt, um bei der Merkmalsextraktion die weniger robusten Merkmale bzw. die Merkmale der Objektumgebung auszusortieren (vgl. Abschnitt 3.2.3).

3. Bei Empfang der dritten bzw. fünften Kameraaufnahme wird der Prozess der Merkmalsextraktion gestartet. Hierfür werden die Kameraaufnahmen des betrachteten Objekts einem `FeatureExtractor`-Objekt übergeben, damit eine Merkmalsextraktion durchgeführt werden kann. Bei der initialen Bestimmung müssen zusätzlich die Schachbrettecken detektiert werden.
4. Ist die Merkmalsextraktion erfolgreich, erhält man als Ergebnis eine Korrespondenzmenge von 2D-Bildpunkten sowie deren Merkmalsdeskriptoren. Weiterhin sind, je nachdem, ob eine initiale Bestimmung oder eine Erweiterung der 3D-Weltpunkte durchgeführt wird, die Punktkorrespondenzen des Kalibrierusters oder die bereits vorhandene Weltpunktmenge und ihre Deskriptoren bekannt. Die vorliegenden Daten werden an ein `Reconstruct3D`-Objekt übergeben, sodass eine 3D-Rekonstruktion gestartet werden kann.
5. Das Ergebnis der 3D-Rekonstruktion ist eine 3D-Weltpunktmenge. Diese kann durch Zugriff auf die Modelldaten (`ModelData`) gespeichert werden.
6. Ist das Verfahren der Erstellung oder Erweiterung des 3D-Modells abgeschlossen, so wird an dieser Stelle auf weitere Aufnahmen des betrachteten Objekts gewartet, um die Menge der Weltpunkte zu erweitern. Dies gilt ebenso für den Fall, dass die Bestimmung der Punktkorrespondenzen oder der Weltpunkte nicht erfolgreich war.

5.1.2 Merkmalsextraktion

Dieser Abschnitt beschreibt die zur Merkmalsextraktion verwendete Komponente der Software (`FeatureExtractor`). Sie nutzt und implementiert die in Kapitel 3 beschriebenen Mechanismen zur Merkmalsextraktion und Korrespondenzsuche.

Es wird die Auswahl verschiedener Merkmalstypen angeboten. Alle auch in OpenCV vorhandenen Algorithmen können hier ausgewählt werden. Es ist prinzipiell möglich, den Prozess der Merkmalsextraktion mit jeder beliebigen Anzahl an Bildern auszuführen, jedoch wird empfohlen, den Prozess mit zwei oder drei Kameraaufnahmen zu starten. Jede weitere Aufnahme nach der zweiten wird dazu verwendet, durch das Aufstellen einer Schnittmenge bei

der Korrespondenzsuche die Robustheit der Merkmalskorrespondenzen zu erhöhen bzw. die Merkmale in der Objektumgebung auszusortieren.

Weiterhin kann der Algorithmus zur Korrespondenzsuche ausgewählt werden. Hier steht auf der einen Seite natürlich die Brute Force Search zur Verfügung. Auf der anderen Seite können das in OpenCV implementierte FLANN-Verfahren und dessen Algorithmen genutzt werden. Hier wird neben der automatischen auch die manuelle Auswahl des Algorithmus angeboten, der vom FLANN-Verfahren verwendet werden soll.

Die Filterung der Punktkorrespondenzen wird wie in Abschnitt 3.2.3 beschrieben durchgeführt. Der hierfür benötigte Schwellwert (*threshold*) kann ebenfalls spezifiziert werden.

Neben dem Zugriff auf die Keypoints, Deskriptoren und Ergebnisse der Korrespondenzsuche stehen auch Funktionen zum Zeichnen der Punktkorrespondenzen auf die verwendeten Bilder zur Verfügung.

5.1.3 3D-Rekonstruktion

Dieser Abschnitt beschreibt die zur 3D-Rekonstruktion verwendete Komponente der Software (Reconstruct3D). Sie implementiert die in Abschnitt 4.2 beschriebenen Schritte zur Bestimmung der 3D-Weltpunkte eines Objekts.

Die Komponente bietet die Möglichkeit einer 3D-Rekonstruktion ausgehend von einer Menge von 2D-Punktkorrespondenzen. Je nachdem, ob eine initiale Bestimmung durchgeführt werden soll oder nicht, werden entweder die Punktkorrespondenzen eines Schachbrettmusters oder die bereits bekannte Weltpunktemenge und deren Deskriptoren benötigt.

Neben der Möglichkeit den gesamten Prozess der 3D-Rekonstruktion auszuführen, sind auch die einzelnen Teilfunktionen vorhanden. Die verfügbaren Funktionen sind im Folgenden aufgelistet und erläutert. Die Begrifflichkeiten, der mathematische Hintergrund sowie das genaue Vorgehen wurden in Kapitel 4, insbesondere in Abschnitt 4.2, ausführlich behandelt.

Fundamentalmatrix Bestimmung der Fundamentalmatrix aus 2D-Punktkorrespondenzen mit Hilfe des RANSAC-Algorithmus (vgl. Abschnitt 4.2.1.2). Alle hierbei detektierten *Outlier* werden im Anschluss aussortiert.

Vier Projektionsmatrizen Ausgehend von der Essentialmatrix werden die vier Möglichkeiten der Projektionsmatrix der zweiten Kamera bestimmt.

Triangulation Ausgehend von den bekannten 2D-Punktkorrespondenzen, der Projektionsmatrix der ersten Kamera und den vier möglichen Projektionsmatrizen der zweiten Kamera werden vier mögliche Mengen an 3D-Weltpunkten trianguliert. Bei der initialen Bestimmung wird die Triangulation gemeinsam mit den 2D-Korrespondenzen der auf dem Schachbrettmuster detektierten Punkte durchgeführt.

Richtige Weltpunktmenge Mit Hilfe der den vier möglichen Projektionsmatrizen zugeordneten triangulierten Weltpunkte kann bestimmt werden, welche die einzig plausible Weltpunktmenge ist. Nur eine der vier Mengen liegt geometrisch vor beiden Kamerazentren.

Gültigkeitsbereich Die triangulierte Weltpunktmenge wird bezüglich der Abstände zu dem Schwerpunkt der Menge untersucht.

Koplanare Weltpunkte Die triangulierte Weltpunktmenge wird auf eine *degenerierte Struktur* untersucht. Diese liegt vor, wenn alle Weltpunkte koplanar sind.

Rückprojektionsfehler Weltpunkte, die einen zu großen Rückprojektionsfehler aufweisen, werden aussortiert.

Skalierungskorrektur Bei der initialen Bestimmung wird der Skalierungsfaktor der Weltpunkte aus einem Kalibrieremuster zugehörigen Punkten ermittelt.

Korrespondenzsuche Bei der Erweiterung der Weltpunktmenge wird eine Korrespondenzsuche zwischen den neuen und den bereits vorhandenen Weltpunkten durchgeführt. Da die Deskriptoren der Bildpunkte bekannt sind, aus denen die Weltpunkte rekonstruiert wurden, ist das an dieser Stelle möglich.

Transformation Die Transformation, die die neuen Weltpunkte in das Koordinatensystem der bereits vorhandenen Weltpunkte transformiert, wird berechnet.

Weiterhin steht eine Funktion zum Zeichnen der rückprojizierten Punkte auf die verwendeten Bilder zur Verfügung.

5.1.4 Datenverwaltung und -speicherung

Die Komponente `ModelData` verwaltet und speichert die generierten Weltpunkte und deren Deskriptoren. Sie kapselt die Funktionalität zum Hinzufügen neuer Weltpunkte und Deskriptoren sowie die Lese- und Schreibzugriffe, um die Daten eines Objekts in einer Datei festzuhalten. Dies kann wahlweise in einer XML- oder YML-Format geschehen.

6 Ergebnisse

Dieses Kapitel beschreibt die in dieser Arbeit erzielten Ergebnisse. Anhand eines beispielhaften Objekts werden die Resultate des verwendeten und implementierten Verfahrens zur Identifikation und 3D-Rekonstruktion von Objekten aus verschiedenen Kameraaufnahmen veranschaulicht. Weiterhin werden die Möglichkeiten und Grenzen des Verfahrens analysiert.

6.1 Identifikation von Objekten

Dieser Abschnitt stellt die Ergebnisse zur Identifikation von Objekten in perspektivisch unterschiedlichen Kameraaufnahmen zusammen. Es wird die Wiedererkennung von Bildmerkmalen über deren Deskriptoren veranschaulicht.

Das Resultat einer Korrespondenzsuche von Bildmerkmalen am Beispiel eines Gegenstandes ist in Abbildung 6.1 bis 6.3 auf Seite 48 dargestellt. Hierbei wurden zwei Kameraaufnahmen des Gegenstandes gemacht und einer SURF-Merkmalsextraktion unterzogen. Aus einer Korrespondenzsuche durch einen Vergleich der Merkmalsdeskriptoren resultierte eine Menge einander zugeordneter Bildpunkte. Die korrespondierenden Punkte sind mit verschiedenfarbigen Linien verbunden. Um die Eindeutigkeit der Merkmalskorrespondenzen zu erhöhen, wurden bei der Korrespondenzsuche nur die Bildpunkte übernommen, bei denen das Verhältnis der beiden geringsten Abstände der verglichenen Deskriptoren einen bestimmten Schwellwert nicht überstieg.

Durch einen Schwellwert zwischen 0.3 und 0.6 werden erfahrungsgemäß sehr robuste Korrespondenzmengen ohne bzw. mit sehr wenigen Fehlzuordnungen detektiert, wobei deren Anzahl noch nicht zu gering für eine darauf basierende 3D-Rekonstruktion ist. Ein größerer Schwellwert resultiert zwar in einer sehr großen Korrespondenzmenge, jedoch auch in vielen Fehlzuordnungen.



Abbildung 6.1: SURF-Korrespondenzen – Schwellwert: 0.1

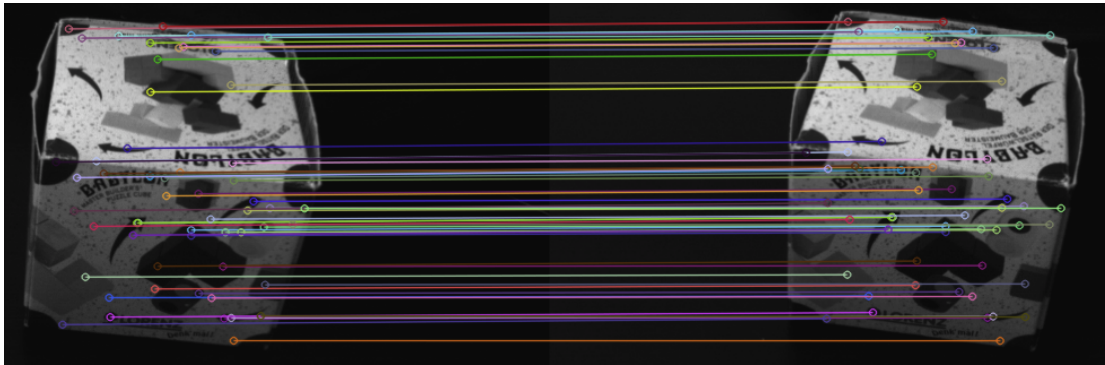


Abbildung 6.2: SURF-Korrespondenzen – Schwellwert: 0.3



Abbildung 6.3: SURF-Korrespondenzen – Schwellwert: 0.5

6.2 3D-Rekonstruktion

Dieser Abschnitt stellt die Ergebnisse zur 3D-Rekonstruktion aus 2D-Bildpunkten von Objekten zusammen. Der Rekonstruktionsprozess und die Erstellung eines 3D-Modells des Objekts wird veranschaulicht.

6.2.1 Initiale Bestimmung einer 3D-Weltpunktmenge

Für die initiale Bestimmung von 3D-Weltpunkten eines Objekts wurden fünf Kameraaufnahmen verwendet. Neben zwei Aufnahmen des Objekts zur Bestimmung der korrespondierenden Bildpunkte wurden zwei Aufnahmen eines Schachbrettmusters gemacht, um die Skalierung der zu rekonstruierenden 3D-Weltpunkte zu ermitteln. Die Schachbrettmuster mussten dafür aus derselben Perspektive wie die erste bzw. zweite Aufnahme des Objekts aufgenommen werden. Eine dritte Aufnahme des Objekts diente der Aussortierung weniger robuster Bildmerkmale sowie der in der Umgebung des Objekts erkannten Merkmalskorrespondenzen.

In Abbildung 6.4 sind die ermittelten korrespondierenden Bildpunkte dargestellt, auf deren Basis die 3D-Weltpunkte des Objekts rekonstruiert wurden.

Nachdem die Menge der 2D-Korrespondenzen aufgestellt war, konnten die Bildpunkte den Prozess der 3D-Rekonstruktion gemeinsam mit den erkannten Schachbrettecken durchlaufen. Zunächst wurde aus den korrespondierenden Bildpunkten die Fundamentalmatrix berechnet, auf deren Grundlage die Projektionsmatrizen berechnet werden konnten. Mit Hilfe der Projektionsmatrizen konnten aus den 2D-Korrespondenzen die 3D-Weltpunkte bis auf eine Skalierung genau trianguliert werden. Durch die verschiedenen Schritte der Verifikation der Weltpunkte wurden sowohl triangulierte Objektpunkte als auch Schachbrettecken aussortiert. Durch die bekannten realen Abstände der Schachbrettecken konnte der Skalierungsfaktor ausfindig gemacht und somit die Weltpunkte richtig skaliert werden.

In Abbildung 6.5 sind die rückprojizierten 3D-Weltpunkte dargestellt. Hier sind die beiden zugrunde liegenden Kameraaufnahmen jeweils doppelt abgebildet. Die linke Seite enthält die Punktkorrespondenzen und Schachbrettecken vor der Triangulation, während die rechte Seite die rückprojizierten 3D-Punkte beinhaltet. Hierbei sind die Punkte des Objekts grün und die Schachbrettecken grau dargestellt.

Eine dreidimensionale Ansicht der initial bestimmten Weltpunkte ist in Abbildung 6.6 dargestellt. Neben den Weltpunkten (blau) sind hier auch die triangulierten Schachbrettecken

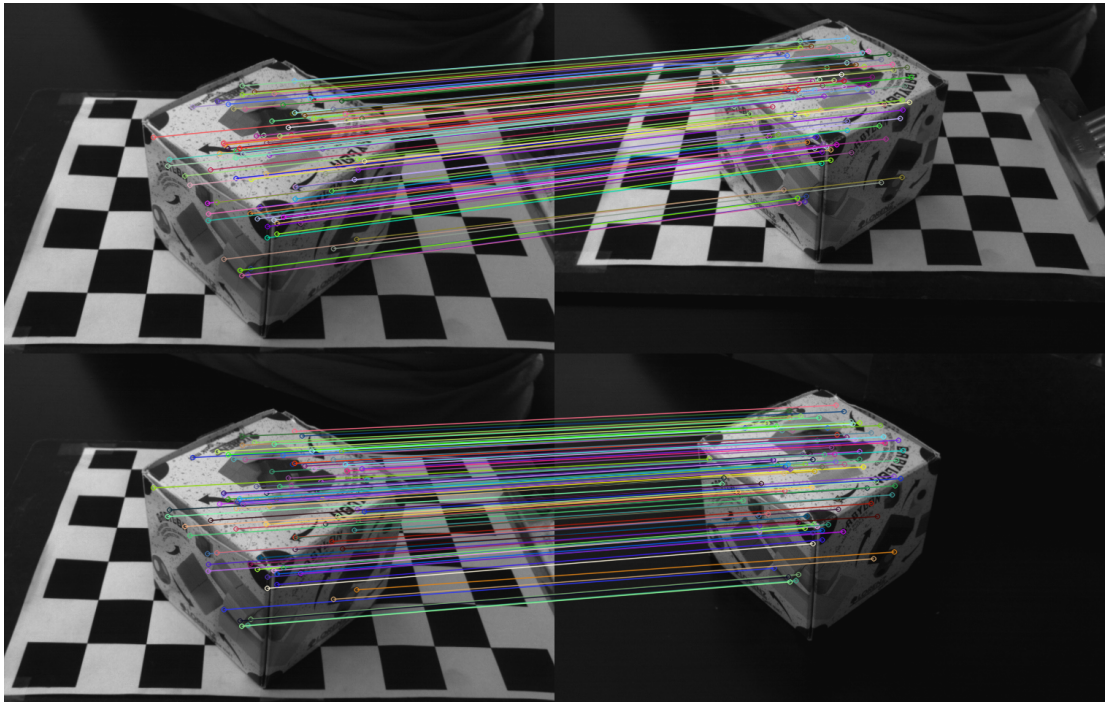


Abbildung 6.4: Ermittelte 2D-Korrespondenzen

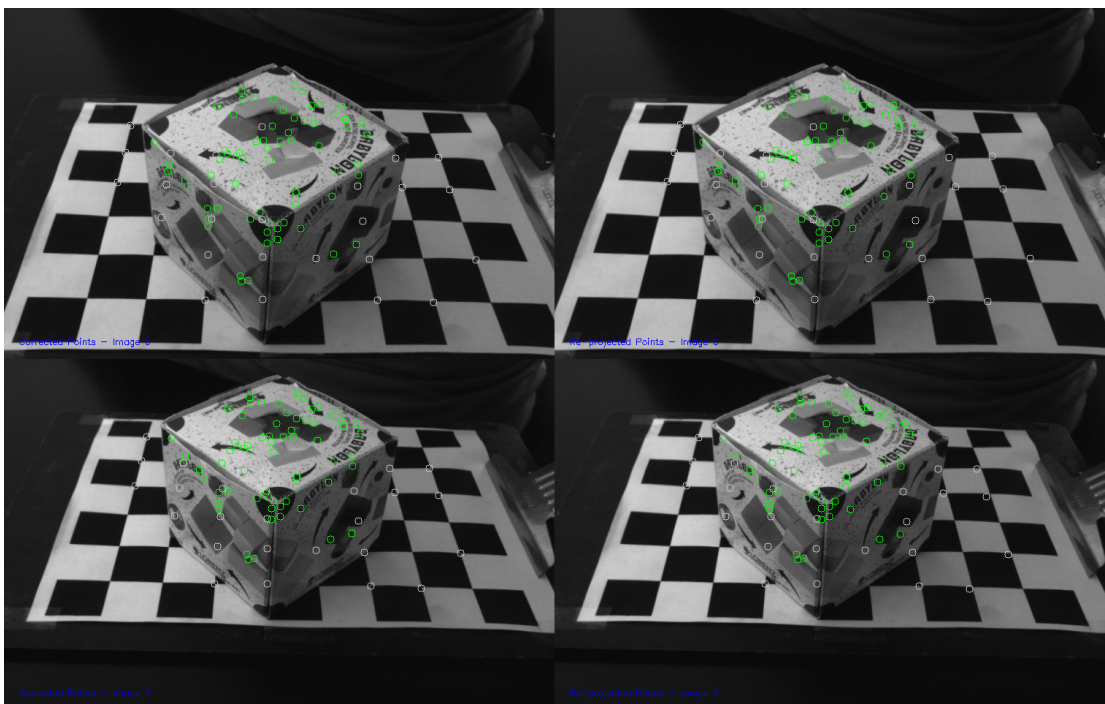


Abbildung 6.5: Rückprojizierte 3D-Weltpunkte

(schwarz) abgebildet. Vergleicht man die Lage der Objektpunkte mit der Lage der Schachbrettecken, ist in den verschiedenen Ansichten deutlich zu erkennen, dass die Weltpunkte erfolgreich rekonstruiert wurden.

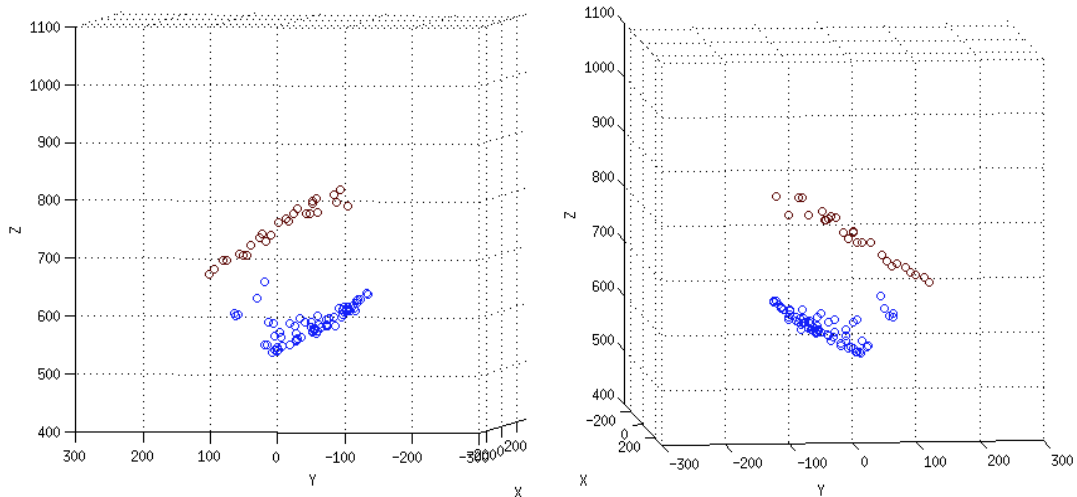


Abbildung 6.6: Dreidimensionale Ansicht der rekonstruierten Weltpunkte

In Tabelle 6.1 sind die den Vorgang und das Ergebnis beschreibenden Daten dieser 3D-Rekonstruktion enthalten.

Initiale Bestimmung der Weltpunkte	
Merkmalstyp	SURF
Schwellwert der Korrespondenzsuche	0.6
Anzahl der 2D-Korrespondenzen	96
Gefilterte Weltpunkte	68

Tabelle 6.1: Initiale Bestimmung der Weltpunkte

6.2.2 Erweiterung einer 3D-Weltpunktmenge

Die initial bestimmte Weltpunktmenge wurde im Anschluss durch neue Kameraaufnahmen des Objekts erweitert. Hierfür wurden drei Aufnahmen verwendet, wobei die dritte auch hier wieder der Aussortierung weniger robuster Merkmale diente.

Der Prozess der 3D-Rekonstruktion unterscheidet sich bei der Erweiterung der Weltpunkte nach der Triangulation und Verifikation von dem Vorgehen bei der initialen Bestimmung,

denn die neuen Weltpunkte müssen in die bereits bekannte Weltpunktmenge integriert werden. Hierfür muss eine Transformation berechnet werden, die die neuen Weltpunkte in das Koordinatensystem der bereits bekannten Weltpunktmenge transformiert und richtig skaliert.

Dafür wurde zunächst eine Korrespondenzmenge zwischen den neu ermittelten und den bereits bekannten Weltpunkten aufgestellt. Dies geschah über eine Korrespondenzsuche über die den Weltpunkten zugeordneten Merkmalsdeskriptoren. Auf Grundlage dieser Korrespondenzen konnte die Transformation berechnet werden.

In Abbildung 6.7 sind die rückprojizierten Punkte der beiden Aufnahmen zusammen mit ihren ursprünglichen Punkten dargestellt, wobei mit der bekannten Menge korrespondierende Punkte orange und neue Punkte grün dargestellt werden. Die rot gefärbten Punkte wurden auf Grund ihres zu großen Rückprojektionsfehlers aussortiert.

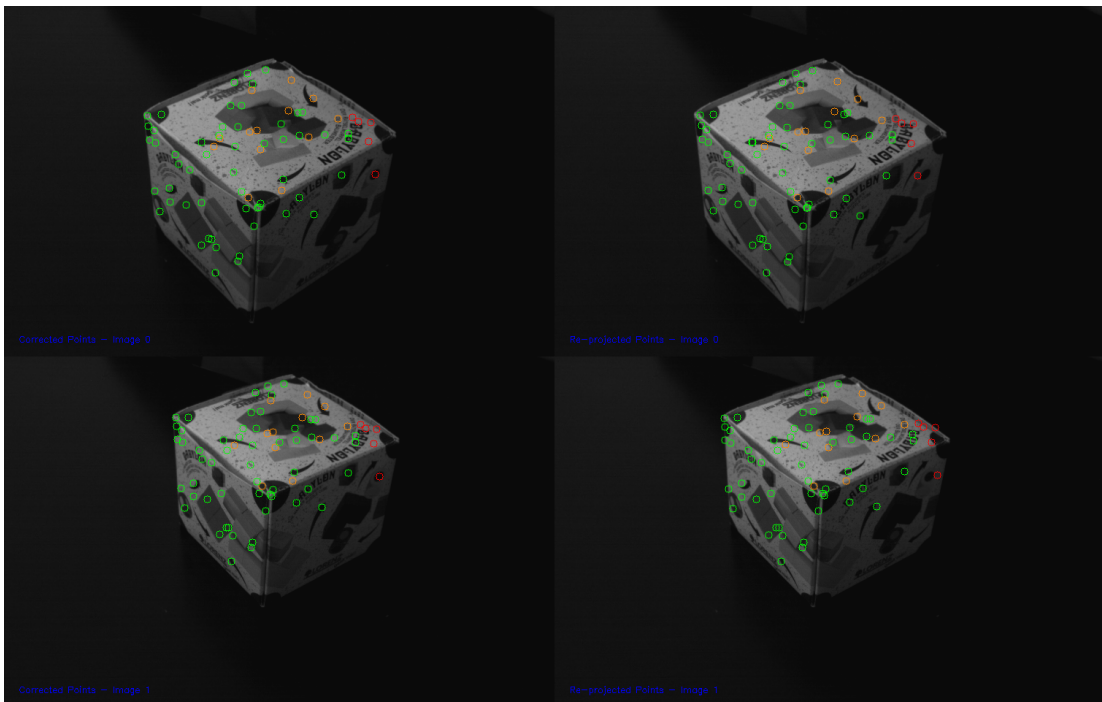


Abbildung 6.7: Rückprojizierte neue 3D-Weltpunkte

In Abbildung 6.8 sind die 3D-Korrespondenzen nach Anwendung der Transformation dargestellt, wobei Punkte aus der bereits bekannten Weltpunktmenge schwarz und die korrespondierenden neu ermittelten Weltpunkte blau gefärbt sind. Hier ist zu erkennen, dass sich diese nur geringfügig voneinander unterscheiden. Die Berechnung der Transformation war demnach erfolgreich.

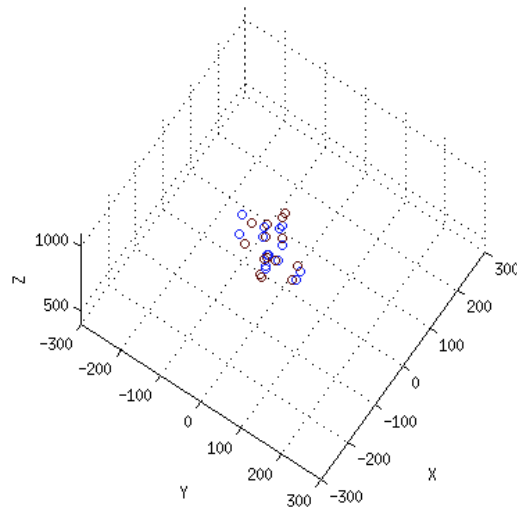


Abbildung 6.8: Transformierte 3D-Korrespondenzen

Alle neu ermittelten Weltpunkte, die sich nicht in der ermittelten Korrespondenzmenge befanden, wurden daraufhin ebenfalls transformiert und damit in die bestehende Weltpunktmenge integriert. Das Resultat ist in verschiedenen Ansichten in Abbildung 6.9 dargestellt. Die bereits bekannten Weltpunkte sind hierbei schwarz, neue Weltpunkte blau gefärbt. Die den Vorgang und das Ergebnis beschreibende Daten sind in Tabelle 6.2 enthalten.

Erste Erweiterung der Weltpunktmenge	
Merkmalstyp	SURF
Schwellwert der Korrespondenzsuche	0.6
Anzahl der 2D-Korrespondenzen	104
Gefilterte Weltpunkte	69
3D-Korrespondenzen in bekannter Menge	14

Tabelle 6.2: Erste Erweiterung der Weltpunktmenge

Die Weltpunktmenge wurde im Anschluss ein zweites Mal erweitert. Das Ergebnis ist in verschiedenen Ansichten in Abbildung 6.10 sowie in Tabelle 6.3 dargestellt.

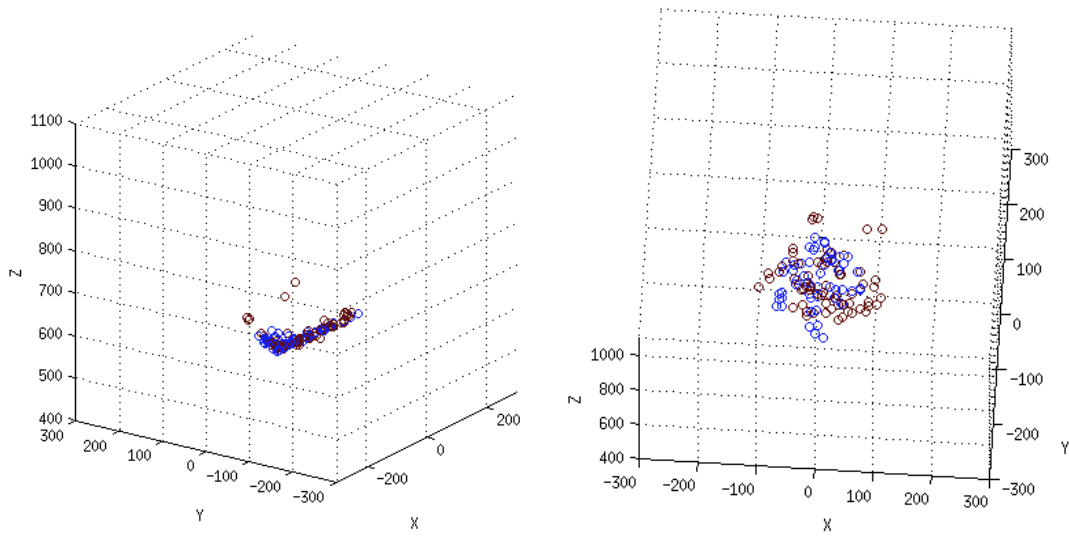


Abbildung 6.9: Erste erweiterte Weltpunktmenge

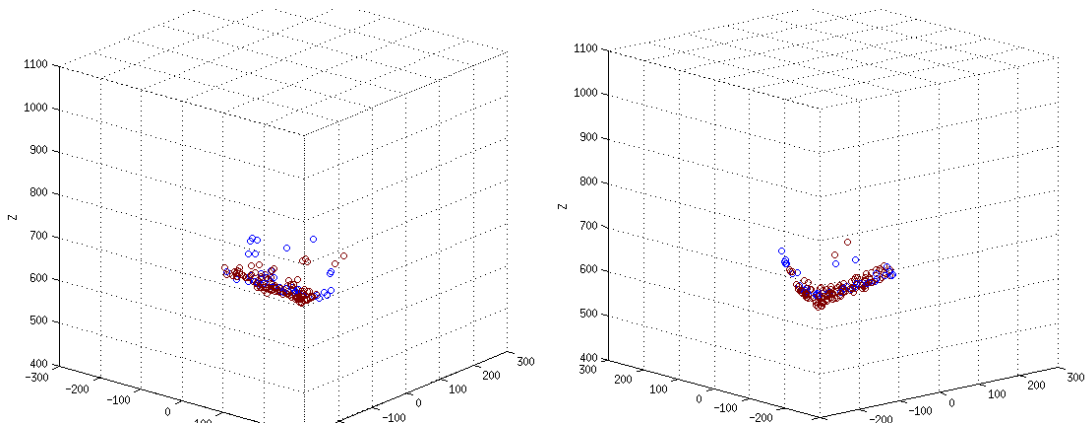


Abbildung 6.10: Zweite erweiterte Weltpunktmenge

Zweite Erweiterung der Weltpunktmenge	
Merkmalstyp	SURF
Schwellwert der Korrespondenzsuche	0.6
Anzahl der 2D-Korrespondenzen	85
Gefilterte Weltpunkte	52
3D-Korrespondenzen in bekannter Menge	11

Tabelle 6.3: Zweite Erweiterung der Weltpunktmenge

6.3 Möglichkeiten und Grenzen des Verfahrens

In den vorangegangenen Abschnitten dieses Kapitels wurde gezeigt, dass die Identifikation und 3D-Rekonstruktion von Objekten mit den in dieser Arbeit behandelten Verfahren möglich und anwendbar ist. Ausgangspunkt für das Verfahren sind lediglich perspektivisch unterschiedliche Kameraaufnahmen eines Objekts. Die dazu entwickelte Software bietet einen automatisierten Prozess zur Erstellung eines 3D-Modells für ein Objekt, das es möglich macht, das Objekt zu identifizieren und die dreidimensionale Lage im Raum zu bestimmen.

Die zu Beginn des Verfahrens der Modellerstellung ermittelten Bildmerkmale bilden hierbei die Grundlage für eine 3D-Rekonstruktion eines Objekts. Weiterhin kann das Objekt über diese Merkmale identifiziert werden, da sie den ermittelten 3D-Weltpunkten des Objekts zugeordnet sind. Zum einen wird während der Modellerstellung davon Gebrauch gemacht, zum anderen kann damit aber auch ein 3D-Modell identifiziert werden, dessen Erstellung bereits abgeschlossen ist. So kann eine 3D-Lagebestimmung eines Objekts erzielt werden.

Allerdings weist das Verfahren auch Grenzen auf. Da Bildmerkmale die Grundlage bilden, muss es möglich sein, genügend Bildmerkmale aus den Kameraaufnahmen des Objekts zu extrahieren und durch eine Korrespondenzsuche einander zuzuordnen. Werden sehr wenige 2D-Punktkorrespondenzen gefunden, so hat der Prozess der 3D-Rekonstruktion eine sehr schwache Grundlage und schlägt mit hoher Wahrscheinlichkeit fehl.

Weiterhin müssen die Bildmerkmale einen sehr hohen Grad an Translations- und Rotationsinvarianz sowie auch perspektivischer Verzeichnung aufweisen, damit es möglich ist, die Weltpunkte eines 3D-Modells bei dessen Erweiterung wiederzuerkennen. Werden hierbei nämlich nicht genügend oder falsche 3D-Korrespondenzen ermittelt, so kann die benötigte Transformation in das Koordinatensystem der bestehenden Menge nicht berechnet und somit die neuen Weltpunkte nicht ins Modell integriert werden. Auch hier würde das Verfahren fehlschlagen.

Ein drittes Problem des Verfahrens sind koplanare Punktmengen. Es wird zwar versucht, diese zu erkennen und auszuschließen, jedoch kann dies nicht garantiert werden. So ist eine aus koplanaren Bildpunkten ermittelte Weltpunktemenge auf Grund fehlender Tiefeninformation nicht rekonstruierbar. Allerdings müssen die Punkte zur Überprüfung trianguliert werden. Dies resultiert oftmals nicht in einer koplanaren, sondern schlichtweg in einer falsch triangulierten Weltpunktemenge.

Ein Beispiel hierfür ist in Abbildung 6.11 dargestellt. Die 2D-Korrespondenzen liegen für den Menschen deutlich erkennbar auf einer Ebene, während die in Abbildung 6.12 dargestellten triangulierten Weltpunkte eindeutig nicht koplanar sind.

Diese Grenzen sollten bei der Anwendung des Verfahrens eingehalten werden. Erfüllen die Eingangsgrößen der 3D-Rekonstruktion, also die 2D-Punktkorrespondenzen in verschiedenen Kameraaufnahmen, die geforderten Kriterien, so kann eine 3D-Rekonstruktion erfolgreich durchgeführt werden.

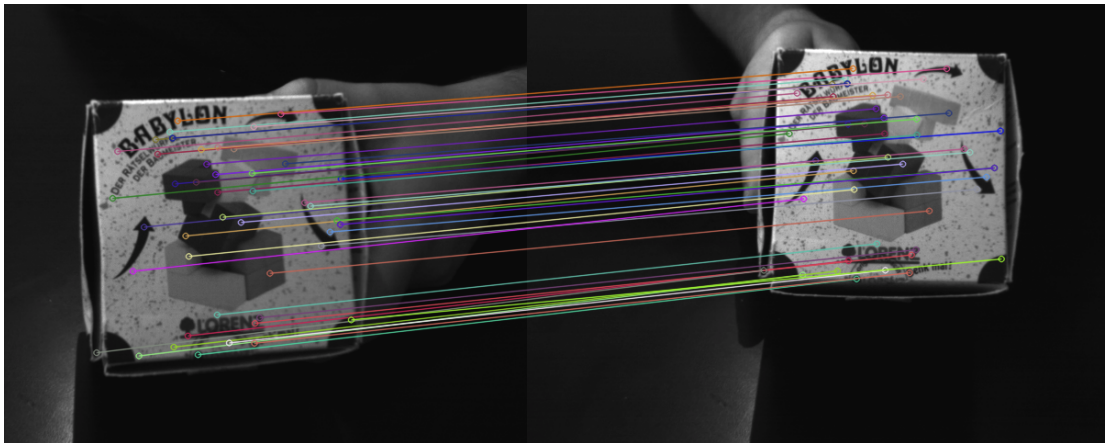


Abbildung 6.11: Koplanare 2D-Punktkorrespondenzen

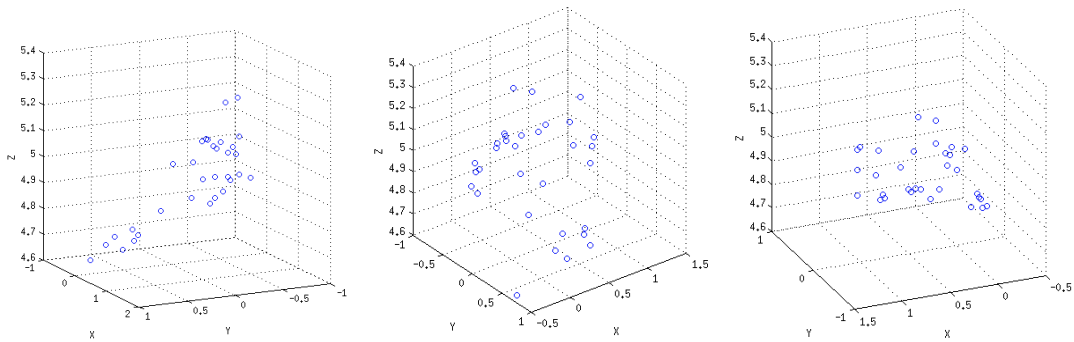


Abbildung 6.12: Falsch triangulierte, eigentlich koplanare Weltpunkte

7 Fazit

In dieser Arbeit wurde ein Verfahren beschrieben und untersucht, mit dem es durch einfache Kameraaufnahmen erreicht werden kann, einen Gegenstand zu identifizieren, sowie dessen dreidimensionale Lage zu rekonstruieren. Die gesamte Durchführung basiert lediglich auf den Aufnahmen einer Kamera. Das Verfahren wurde für das ROS-Framework implementiert, sodass alle mit ROS ausgestatteten Computer und Roboter für den Betrieb geeignet sind.

Durch korrespondierende Bildpunkte in perspektivisch unterschiedlichen Kameraaufnahmen eines Objekts ist es mit diesem Verfahren möglich, die zugehörigen dreidimensionalen Punkte zu rekonstruieren. Die benötigten Korrespondenzen können durch eine Extraktion von Bildmerkmalen aus den verwendeten Kameraaufnahmen ermittelt werden.

Da diese 2D-Korrespondenzen die Grundlage für den Rekonstruktionsprozess bilden, müssen sie bestimmte Kriterien erfüllen. Für die Möglichkeit der Identifikation ist die erste Anforderung die Eindeutigkeit eines Merkmals. Weiterhin ist eine erhebliche Anzahl an Punktkorrespondenzen notwendig, weshalb nur Gegenstände verwendet werden können, auf denen genügend Merkmale zu erkennen sind. Eine letzte Anforderung an die Korrespondenzen ist, dass diese nicht koplanar sein dürfen. Sind diese Eingangsbedingungen erfüllt, kann eine erfolgreiche 3D-Rekonstruktion durchgeführt werden.

Durch die Implementierung dieses Verfahrens kann ein Modell erzeugt werden, das neben den 3D-Punkten eines Objekts auch die zugehörigen Bildmerkmale enthält. Damit ist es sowohl möglich ein Objekt über Kameraaufnahmen zu identifizieren als auch dessen Lage im Raum zu bestimmen.

Abbildungsverzeichnis

1.1	Roboter des Robot Vision-Projekts	2
1.2	3D-Rekonstruktion von korrespondierenden Bildpunkten	3
2.1	VRmC3+PRO-BW	5
2.2	ROS <i>Computation Graph</i>	7
3.1	DoG-Pyramide (Low04)	10
3.2	SIFT-Deskriptor (Low04)	11
3.3	L_{xx} und L_{xy} bzw. Approximation durch Mittelwertfilter (BETG08)	12
3.4	Berechnung von Pixelsummen durch Integralbilder (BETG08)	13
3.5	Vergleich der beiden kleinsten Abstände zwischen zwei Deskriptoren	15
4.1	Lochkameramodell (Mei08)	18
4.2	Schachbrettmuster (6 x 8 innere Ecken)	20
4.3	Target-to-Source Mapping (Mei12)	22
4.4	Epipolargeometrie (HZ04)	23
4.5	Kameraebenen - Reihenvektoren einer Projektionsmatrix \mathbf{P} (HZ04)	26
4.6	Vier Möglichkeiten für $\hat{\mathbf{P}}_2$ (HZ04)	29
4.7	Strahlen durch rückprojizierte Punkte verlaufen versetzt (HZ04)	31
4.8	Koplanare (a) und nicht koplanare Weltpunkte (b)	33
5.1	Überblick über die Software-Komponenten	41
5.2	Ablauf des Hauptprogramms	43
6.1	SURF-Korrespondenzen – Schwellwert: 0.1	48
6.2	SURF-Korrespondenzen – Schwellwert: 0.3	48
6.3	SURF-Korrespondenzen – Schwellwert: 0.5	48
6.4	Ermittelte 2D-Korrespondenzen	50
6.5	Rückprojizierte 3D-Weltpunkte	50

6.6	Dreidimensionale Ansicht der rekonstruierten Weltpunkte	51
6.7	Rückprojizierte neue 3D-Weltpunkte	52
6.8	Transformierte 3D-Korrespondenzen	53
6.9	Erste erweiterte Weltpunktmenge	54
6.10	Zweite erweiterte Weltpunktmenge	54
6.11	Koplanare 2D-Punktkorrespondenzen	56
6.12	Falsch triangulierte, eigentlich koplanare Weltpunkte	56

Tabellenverzeichnis

2.1	Technische Daten der VRmC3+PRO-USB2.0-BW	6
6.1	Initiale Bestimmung der Weltpunkte	51
6.2	Erste Erweiterung der Weltpunktmenge	53
6.3	Zweite Erweiterung der Weltpunktmenge	54

Glossar

Deskriptor	Ein hochdimensionaler Vektor, der der Beschreibung von Bildmerkmalen dient.
Framework	Eine Programmstruktur, in die ein unter einem vorgegebenen Rahmen (engl. Frame) entwickeltes Programm integriert werden kann.
Knoten	Ein in das ROS-Framework integriertes, ausführbares Programm (siehe Abschnitt 2.2, S. 6).
OpenCV	Eine Softwarebibliothek für Bildverarbeitung und maschinelles Sehen (siehe Abschnitt 2.3, S. 8).
Outlier	Ausreißer bzw. bestimmte Kriterien nicht erfüllender Punkt.
ROS	Robot Operating System (siehe Abschnitt 2.2, S. 6).
SIFT	Scale-Invariant Feature Transform (siehe Abschnitt 3.1.1, S. 10).
SURF	Speeded Up Robust Features (siehe Abschnitt 3.1.2, S. 11).
Topic	Ermöglicht die asynchrone Kommunikation im ROS-Framework (siehe Abschnitt 2.2, S. 6).

Formelzeichen

'	Korrespondenz von Punkten / Linien / Vektoren.
$I(x, y)$	Wert eines Integralbildes an dem Punkt $(x, y)^T$.
I_Σ	Integralbild.
L	Zweite Ableitung der Gaußfunktion.
C	Kamerazentrum.
D	Deskriptor.
E	Homogene 3 x 3 Essentialmatrix.
F	Homogene 3 x 3 Fundamentalmatrix.
H	Hesse-Matrix.
K	Kalibriermatrix.
Pⁱ	Reihenvektor einer Projektionsmatrix.
P	Homogene 3 x 4 Projektionsmatrix.
R	3 x 3 Rotationsmatrix.
T	Homogene 4 x 4 Transformationsmatrix.
U, S, V	Drei Ergebnismatrizen einer Singulärwertzerlegung.
X	Dreidimensionaler Weltpunkt in euklidischen $(X, Y, Z)^T$ oder homogenen Koordinaten $(X, Y, Z, W)^T$.
α	Verstärkungsfaktor.
\hat{P}	Homogene 3 x 4 Projektionsmatrix einer normalisierten Kamera.
\hat{x}	Bildpunkt einer normalisierten Kamera.
σ	Skalierungsfaktor.
d	Verzeichnungskoeffizienten $(k_1, k_2, p_1, p_2, k_3)^T$.
e	Epipol.

Formelzeichen

\mathbf{f}	Bildweite $(f_x, f_y)^T$.
\mathbf{h}	Bildhauptpunkt $(h_x, h_y)^T$.
\mathbf{l}	Epipolarlinie.
\mathbf{p}_i	Spaltenvektor einer Projektionsmatrix.
\mathbf{t}	3 x 1 Translationsvektor.
\mathbf{x}	Zweidimensionaler Bildpunkt in euklidischen $(x, y)^T$ oder homogenen Koordinaten $(x, y, w)^T$.
$\text{SVD}(\mathbf{A})$	Singulärwertzerlegung einer Matrix \mathbf{A} .
T	Transponierte eines Vektors / einer Matrix.
d	Euklidischer Abstand.
$d(\mathbf{x}, \mathbf{y})$	Euklidischer Abstand zwischen zwei Vektoren \mathbf{x} und \mathbf{y} .
$\text{depth}(\mathbf{X}, \mathbf{P})$	Tiefe eines Punktes \mathbf{X} vor einer Kamera.
$\det(\mathbf{A})$	Determinante einer Matrix \mathbf{A} .
r_{err}	Rückprojektionsfehler.
s	Skalierungsfaktor.
$\text{sign}(a)$	Vorzeichen einer Zahl a .
threshold	Schwellwert.

Literaturverzeichnis

- [AOV12] ALAHI, A. ; ORTIZ, R. ; VANDERGHEYNST, P.: FREAK: Fast Retina Keypoint. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2012
- [BETG08] BAY, Herbert ; Ess, Andreas ; TUYTELAARS, Tinne ; GOOL, Luc V.: SURF: Speeded Up Robust Features. In: *Computer Vision and Image Understanding (CVIU)* (2008), Nr. 3, S. 346–359
- [Bou08] BOUGUET, J. Y.: *Camera Calibration Toolbox for Matlab*. 2008. – http://www.vision.caltech.edu/bouguetj/calib_doc/; Zugriffsdatum: 06.07.2012
- [ELF97] EGGERT, D.W. ; LORUSSO, A. ; FISHER, R.B.: Estimating 3-D rigid body transformations: a comparison of four major algorithms. In: *Machine Vision and Applications* 9 (1997), 272-290. <http://dx.doi.org/10.1007/s001380050048>. – ISSN 0932–8092. – 10.1007/s001380050048
- [Fri11] FRIES, Carsten: *Kamerabasierte Identifizierung und Lokalisierung von Gegenständen für flexible Roboter*, Hochschule für Angewandte Wissenschaften Hamburg, Masterarbeit, 2011
- [HHN88] HORN, Berthold K. P. ; HILDEN, H.M. ; NEGAHDARIPOUR, Shariar: Closed-Form Solution of Absolute Orientation using Orthonormal Matrices. In: *JOURNAL OF THE OPTICAL SOCIETY AMERICA* 5 (1988), Nr. 7, S. 1127–1135
- [HZ04] HARTLEY, R. I. ; ZISSERMAN, A.: *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004
- [Lag11] LAGANIÈRE, Robert: *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, ISBN: 1849513244, 2011
- [Low99] LOWE, David G.: Object Recognition from Local Scale-Invariant Features. In: *International Conference on Computer Vision*, 1999

- [Low04] LOWE, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* 60 (2004), Nr. 2, S. 91–110
- [MCUP02] MATAS, J. ; CHUM, O. ; URBAN, M. ; PAJDLA, T.: Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In: *British Machine Vision Conference*, 2002, S. 384–393
- [Mei08] MEISEL, Andreas: *Anwendungen 2: 3D-Bildverarbeitung, Vorlesungsskript*. 2008. – <http://www.informatik.haw-hamburg.de/1708.html>; Zugriffsdatum: 30.06.2012
- [Mei12] MEISEL, Andreas: *Robot Vision, Vorlesungsskript*. 2012. – http://www.informatik.haw-hamburg.de/wp_robot_vision.html; Zugriffsdatum: 30.06.2012
- [ML09] MUJA, Marius ; LOWE, David G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: *International Conference on Computer Vision Theory and Applications*, 2009, S. 331–340
- [Ope12] *OpenCV Dokumentation*. 2012. – <http://docs.opencv.org/>; Zugriffsdatum: 06.07.2012
- [RD06] ROSTEN, Edward ; DRUMMOND, Tom: Machine learning for high-speed corner detection. In: *European Conference on Computer Vision* Bd. 1, 2006, S. 430–443
- [ROS12] *ROS Dokumentation*. 2012. – <http://www.ros.org/>; Zugriffsdatum: 06.07.2012
- [RRKB11] RUBLEE, Ethan ; RABAUD, Vincent ; KONOLIGE, Kurt ; BRADSKI, Gary: ORB: An Efficient Alternative to SIFT or SURF. In: *International Conference on Computer Vision (ICCV)*, 2011, S. 2564–2571
- [Zha98] ZHANG, Zhengyou: A Flexible New Technique for Camera Calibration. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (1998), S. 1330–1334

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 30. August 2012

 Peter Oltmann