

# **Modellbasierte Objekterkennung für einen Industrieroboter**

Max Schwenzer

20.10.2015

# Aufgabenstellung

## Modellbasierte Objekterkennung für einen Industrieroboter

**Inv.-Nr.:** 2015/23288

**Verfasser:** Max Schwenzler

**Ausgabe:** 20.04.2015

**Betreuer:** Gregor Lux

**Abgabe:** 20.10.2015

### **Ausgangssituation:**

Roboter werden in großer Zahl in der Industrie eingesetzt. Es wird erwartet, dass durch die Entwicklung hin zur „Industrie 4.0“ die Verbreitung von Industrierobotern weiter zunimmt. Aktuell wird besonderes Augenmerk auf die Steigerung der Flexibilität und Autonomie der Roboter gelegt. So kann die Zusammenarbeit von Mensch und Maschine verbessert werden, und die Produktion schnell umgestaltet oder geändert werden.

Heutige Industrieroboter werden auf festgelegten Bahnen bewegt. Aufgrund mangelnder Sensorik werden dabei restriktive Vorgaben für die Umgebung getroffen. So werden bspw. Position und Orientierung des Werkstücks als vorgegeben angenommen. Dies lässt sich nur mit aufwendigen Positionierungsvorrichtungen und teuren Messsystemen gewährleisten. Um dies zu vermeiden und die Mensch-Maschine-Interaktion zu vereinfachen, werden Sensoren verwendet. Diese ermöglichen dem Roboter seine Umwelt wahrzunehmen und den Arbeitsablauf entsprechend anzupassen. Die Arbeit ist Teil eines Forschungsprojekts in dem ein Objekt mit einem robotergeführten Laserscanner zur Qualitätssicherung vermessen wird. Für eine akkurate Bahnplanung wird die Position des Objekts momentan manuell vermessen.

### **Zielsetzung:**

In dieser Arbeit soll ein bestehender Industrieroboter befähigt werden, eigenständig sein Werkstück zu erkennen und dessen Position zu bestimmen.

Der Roboter ist mit einem kommerziellen Laserscanner ausgestattet und wird in der Qualitätssicherung eingesetzt werden. Dazu muss er an einem Werkstück bestimmte

Punkte abfahren und vermessen. Um aufwändiges Einspannen des Werkstücks zu vermeiden, soll das Werkstück selbständig anhand seines CAD-Modells erkannt und seine Position relativ zum Roboter bestimmt werden. Anschließend wird – nicht Teil dieser Arbeit – die optimale Bahn zum Abtasten der Messpunkte geplant.

### **Vorgehensweise und Arbeitsmethodik:**

- Überblick über Methoden und Möglichkeiten der Objekterkennung und Stereovision.
- Begründete Auswahl eines Konzepts zur Erkennung des Werkstücks und zu dessen Positionsbestimmung.
- Umsetzung des Konzepts mit Open Source Mitteln und mit der gegebenen Hardware.
- Validierung des Systems und Aufzeigen seiner Grenzen.

### **Vereinbarung:**

Mit der Betreuung von Herrn cand.-M.Sc. Max Schwenzer durch Herrn Dipl.-Phys. Gregor Lux fließt geistiges Eigentum des *iwb* in diese Arbeit ein. Eine Veröffentlichung der Arbeit oder eine Weitergabe an Dritte bedarf der Genehmigung durch den Lehrstuhlinhaber. Der Archivierung der Arbeit in der *iwb*-eigenen und nur für *iwb*-Mitarbeiter zugänglichen Bibliothek als Bestand und in der digitalen Studienarbeitsdatenbank des *iwb* als PDF-Dokument stimme ich zu.

Garching, den 20.10.2015

Prof. Dr.-Ing.  
Gunther Reinhart

Dipl.-Phys.  
Gregor Lux

cand.-M.Sc.  
Max Schwenzer

# Inhalt

<b>Aufgabenstellung</b>	<b>I</b>
<b>Abstract</b>	<b>IX</b>
<b>Abkürzungsverzeichnis</b>	<b>XI</b>
<b>Verzeichnis der Formelzeichen</b>	<b>XII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Handlungsbedarf und Zielsetzung . . . . .	2
1.2.1 Arbeitsbausteine . . . . .	3
1.2.2 Randbedingungen . . . . .	4
1.3 Aufbau der Arbeit . . . . .	4
<b>2 Stand der Technik</b>	<b>6</b>
2.1 Bilderkennung . . . . .	6
2.1.1 Objekterkennung . . . . .	6
2.1.2 Robot Vision . . . . .	8
2.2 Stereovision . . . . .	9
2.3 Anwendungen in der Praxis . . . . .	10
<b>3 Grundlagen der Bildverarbeitung</b>	<b>12</b>
3.1 Bildrepräsentation und Farbräume . . . . .	12
3.2 Einfache Bildverarbeitungsmethoden . . . . .	13
3.2.1 Histogrammanpassung . . . . .	13
3.2.2 Filter . . . . .	17
3.2.3 Skalierung . . . . .	20
3.3 Canny Kantendetektor . . . . .	22
3.4 Houghtransformation . . . . .	26
3.5 Geometrische Kamerakalibrierung . . . . .	27

<b>4 Objekterkennung</b>	<b>29</b>
4.1 Generischer Aufbau einer Objekterkennung . . . . .	29
4.2 Objekterkennung mit Konturen . . . . .	30
4.2.1 Konturapproximation . . . . .	30
4.2.2 Charakteristiken von Konturen . . . . .	31
4.2.3 Bildmomente . . . . .	34
4.3 Objekterkennung mit markanten Punkten . . . . .	36
4.3.1 FAST Merkmal-Extraktor . . . . .	37
4.3.2 BRIEF Merkmal-Deskriptor . . . . .	38
4.3.3 ORB Algorithmus: Oriented FAST and Rotated BRIEF . . . . .	39
4.4 Zusammenfassung und Bezug zum Projekt . . . . .	40
<b>5 Stereovision</b>	<b>41</b>
5.1 Grundlagen der Stereovision . . . . .	41
5.2 Rektifikation . . . . .	42
5.2.1 Epipolargeometrie . . . . .	43
5.2.2 Fundamentalmatrix . . . . .	45
5.2.3 Bestimmen der Fundamentalmatrix . . . . .	47
5.3 Korrespondenzanalyse und Triangulation . . . . .	48
5.4 Zusammenfassung und Bezug zum Projekt . . . . .	51
<b>6 Praktische Umsetzung</b>	<b>52</b>
6.1 System . . . . .	52
6.2 Vorverarbeitung . . . . .	54
6.3 Objekterkennung: Konturen . . . . .	56
6.3.1 Beschreiben von Konturen . . . . .	57
6.3.2 Zuordnung von Konturen . . . . .	58
6.4 Stereovision . . . . .	62
6.4.1 Triangulation von Konturen . . . . .	62
6.4.2 Erzeugen einer Tiefenkarte . . . . .	64
<b>7 Ergebnisse</b>	<b>69</b>
7.1 Objekterkennung . . . . .	69
7.1.1 Geometrische Korrespondenzanalyse . . . . .	69
7.1.2 Quellenbasierte Korrespondenzanalyse . . . . .	70
7.1.3 Modellbasierte Objekterkennung . . . . .	71
7.2 Stereovision . . . . .	78
7.2.1 Geometrische Kamerakalibrierung . . . . .	78

7.2.2 Entfernungsmessung . . . . .	81
<b>8 Zusammenfassung und Ausblick</b>	<b>84</b>
8.1 Zusammenfassung . . . . .	84
8.1.1 Objekterkennung . . . . .	84
8.1.2 Stereovision . . . . .	86
8.2 Ausblick . . . . .	86
8.2.1 Weiterführende Forschung . . . . .	87
8.2.2 Vision . . . . .	88
<b>Literaturverzeichnis</b>	<b>90</b>
<b>Anhang</b>	<b>102</b>
A.1 Teil 1: Ergänzende Abbildungen . . . . .	102
A.2 Teil 2: Details Boxplots . . . . .	105
A.3 Teil 3: Linien Zuordnung . . . . .	107
A.4 Teil 4: Erstelltes Programm . . . . .	110

## Abbildungsverzeichnis

3.1	Bildkoordinatensysteme . . . . .	13
3.2	Histogrammanpassung Sensor . . . . .	15
3.3	Resultate verschiedener Filter . . . . .	18
3.4	Prinzip Gauß-Filter . . . . .	19
3.5	Bildverkleinerung . . . . .	21
3.6	Skalierung Beispiel . . . . .	21
3.7	Sobel-Filter Sensor . . . . .	23
3.8	Canny Kantendetektor . . . . .	25
3.9	Bildentzerrung . . . . .	27
3.10	Sub-Pixel-Ecke . . . . .	28
4.1	Kontursegmentierung . . . . .	31
4.2	Konturapproximation durch Polygone . . . . .	32
4.3	Konturparameter . . . . .	33
4.4	Funktionsprinzip FAST . . . . .	38
4.5	MusterBinärer-Deskriptoren Beispiel . . . . .	39
5.1	Standard-Stereo Aufbau . . . . .	43
5.2	Epipolargeometrie . . . . .	44
5.3	Rotation und Translation . . . . .	46
5.4	Prinzip RANSAC-Algorithmus . . . . .	48
5.5	Triangulation . . . . .	49
6.1	Modellbild . . . . .	54
6.2	Vorverarbeitungsschritte . . . . .	55
6.3	Ablauf der Objekterkennung mit Konturen. . . . .	61
6.4	Automatische Kamerakalibrierung . . . . .	63
6.5	Rektifizierung Nikon D40 . . . . .	64
6.6	Schätzen der Fundamentalmatrix (externe Kamera) . . . . .	66
6.7	Schätzen der Fundamentalmatrix (Sensor) . . . . .	67
6.8	Tiefenkarte . . . . .	68

7.1	Ergebnisse geometrische Zuordnung . . . . .	70
7.2	Ergebnisse relationale Überprüfung . . . . .	71
7.3	Beispiel geometrische Zuordnung . . . . .	72
7.4	Beispiele quellbasierte Korrespondenzanalyse . . . . .	73
7.5	Beispiel geometrische Konturqualität . . . . .	74
7.6	Ergebnisse automatische modell- und quellenbasierte Zuordnung . . . . .	74
7.7	Ergebnisse modellbasierte Stereo-Zuordnung . . . . .	75
7.8	Erkennungsrate . . . . .	76
7.9	Modellbasierte Objekterkennung Beispiel . . . . .	77
7.10	Geometrische Kamerakalibrierung Beispiel . . . . .	79
7.11	Ungenau detektiertes Kalibrierungsmuster . . . . .	79
7.12	Definition Boxplot . . . . .	80
7.13	Boxplot intrinsische Parameter . . . . .	81
7.14	Boxplot Entfernungsmessung . . . . .	82
7.15	Qualität der Entfernungsmessung . . . . .	83
A.1	Sobel-Filter externe Kamera . . . . .	102
A.2	Histogrammanpassung externe Kamera . . . . .	103
A.3	Resultate verschiedener Filter . . . . .	104
A.4	Rektifizierung Sensor Beispiel . . . . .	104
A.5	Linien Zuordnung Beispiel . . . . .	108
A.6	Ablauf Linienzuordnung . . . . .	109
A.7	Programme . . . . .	111

## Tabellenverzeichnis

4.1	Hu Momente . . . . .	36
4.2	Laufzeitvergleich von ORB, SIFT und SURF . . . . .	40
6.1	Konturparameter . . . . .	57
A.1	Streuung intrinsische Kameraparameter . . . . .	105
A.2	Abweichung der Entfernungsmessung . . . . .	105
A.3	Relative Abweichung der Entfernungsmessung . . . . .	105
A.4	Absolute Abweichung der Entfernungsmessung . . . . .	106

## Abstract

### English:

This work describes a mono-stereo-system for an industrial robot to recognize his workpiece and determine its relative distance. It is a pure software update for a passive, model based object recognition.

This work is part of a research project in automotive quality assurance. An industrial robot with a laser scanner is being used. The object recognition is based on the detection of contours which are matched by their geometric and relational characteristics. A new matching algorithm is explained which automatically selects a seed contour and matches the surrounding contours with higher robustness. This method reaches a recognition rate up to 99 % for model based object recognition. In model based stereo recognition 85 % are reached.

A mono-stereo-system based on contour recognition is created. The object's distance is estimated with an average deviation of +20 %. This is being caused by the inaccurate geometric camera calibration of the sensor's camera chip. The value of the focal length has a variance of  $\pm 10\%$ . Nonetheless the capability of a near field measuring system to perform a rough long distance measurement is demonstrated. The object can be approached reliably if the offset in the distance measuring is considered.

**Deutsch:**

In dieser Arbeit wird ein passives Mono-Stereo-System zur Objekt- und Positionserkennung eines Werkstücks entwickelt. Die Komponenten des Systems sind ein Industrieroboter und ein Laserscanner. Es wird in der Qualitätssicherung eingesetzt werden. Die Arbeit beschreibt ein reines Softwareupdate.

Die Objekterkennung basiert auf der Detektion von Konturen, deren geometrische und relationalen Charakteristika verglichen werden. Es wird ein neuer Algorithmus für die Zuordnung von Konturen beschrieben. Er bestimmt eine Quellkontur, von welcher ausgehend benachbarten Konturen mit hoher Robustheit zugeordnet werden. In der modellbasierten Objekterkennung erreicht diese Methode eine Erkennungsrate von 99 %, in der modellbasierten Stereo-Objekterkennung 85 %.

Das Mono-Stereo-System nutzt die Objekterkennung mit Konturen, um die Distanz zum Objekt zu schätzen. Dabei wird gezeigt, wie mit einem für den Nahbereich konzipiertes Messsystem Distanzmessungen durchgeführt werden. Die berechneten Entfernungswerte weichen durchschnittlich um +20 % von der realen Entfernung ab. Das liegt an der schwierigen geometrischen Kamerakalibrierung des Sensors und kann herausgerechnet werden. Die Brennweite unterliegt einer Streuung von  $\pm 10\%$ .

## Abkürzungsverzeichnis

- AHE** Adaptive Histogram Equalization  
**BIRSK** Binary Robust Invariant Scalable Keypoints  
**BRIEF** Binary Robust Independent Elementary Features  
**CAD** Computer Aided Design  
**CLAHE** Contrast Limited Histogram Equalization  
**DSLR** Digital Single Lens Reflex  
**FAST** Features from Accelerated Segment Test  
**GPS** Global Positioning System  
**IFF** Fraunhofer Institut für Fabrikbetrieb und -automatisierung  
**IPA** Fraunhofer Institut für Produktionstechnik und Automatisierung  
**iwb** Institut für Werkzeugmaschinen und Betriebswissenschaften  
**KMG** Koordinatenmessgerät  
**ORB** Oriented FAST and Rotated BRIEF  
**PTP** Point-To-Point  
**QR** Quick Response Code  
**RANSAC** RANdom SAmple Consensus  
**RoHoQ** Roboterbasierte, hochgenaue Qualitätssicherungssysteme  
**SBGM** Semi-Global Block Matching  
**SIFT** Scale-Invariant Feature Transform  
**SLAM** Simultaneous Localization And Mapping  
**SURF** Speeded-Up Robust Features  
**SVM** Support Vector Maschine  
**UAV** Unmanned Air Vehicle  
**VDMA** Verband Deutscher Maschinen- und Anlagenbau  
**VR** Virtual Reality

## Verzeichnis der Formelzeichen

$I$	[ ]	Intensitätswert
$G()$	[ ]	Grauwert
$G_{\sigma}()$	[ ]	Gauß-Filter
$\sigma$	[ ]	Standardabweichung
$BF()$	[ ]	Bilateral-Filter
$f()$	[ ]	Filter ( allgemein )
$f_x, f_y$	[px]	Brennweite
$c_x, c_y$	[px]	Koordinate Optisches Zentrum
$O$	[px]	Optisches Zentrum
$x, y$	[px]	Koordinate Bildpunkt
$X, Y, Z$	[mm]	Physikalische Koordinaten
$p, q$	[mm] / [px]	Punkt/Pixel
$l, r$	[ ]	Indexbezeichner links, rechts
$T$	[diverse] / [mm]	Schwellenwert/Basisabstand
$m_{i,j}$	[ ]	Spatiales Moment
$\mu_{i,j}$	[ ]	Zentrales Moment
$\nu_{i,j}$	[ ]	Zentrales, normalisiertes Moment
$Hu$	[ ]	Hu Moment
$\eta$	[ ] / [px]	Exzentrizität/Schwellenwert
$\kappa$	[ ]	Rundheit
$r$	[mm]	Radius
$A$	[mm <sup>2</sup> ]	Fläche
$e$	[px]	Epipole
$F$	[px]	Fundamentalmatrix
$E$	[px]	Essentialmatrix
$\delta$	[mm] / [%]	Abweichung

# 1 Einleitung

## 1.1 Motivation

Automatisierung und Industrie 4.0 sind zurzeit die alles beherrschenden Schlagworte im Maschinenbau. Die Produktion soll flexibler werden, Maschinen sollen Daten austauschen und autonom agieren (CIUPEK 2015a). Vor dem Hintergrund des sich abzeichnenden Fachkräftemangels und der hohen Arbeitskosten in Deutschland nehmen Automatisierung und Flexibilisierung von Maschinen zu (SCHNEIDER & STEIGER 2015). Nach dem Vorsitzenden der VDMA-Fachabteilung Industrielle Bildverarbeitung, Olaf Munkelt, setzen dabei immer mehr Branchen auf Bildverarbeitung als Teil des Fertigungsprozesses (CIUPEK 2014). Die Bildverarbeitungsindustrie wächst<sup>1</sup> 2014 und 2015 im zweistelligen Prozentbereich (WENDEL 2014, 2015). Werden Bildverarbeitungssysteme momentan hauptsächlich zur Qualitäts- und Vollständigkeitsprüfung eingesetzt, so wandeln sie sich zunehmend zu einem integralen Bestandteil der Fertigungsstrategie (CIUPEK 2014). Als Ergänzung zu einem Industrieroboter wird dieser befähigt, flexibel auf seine Umwelt zu reagieren (VDMA 2014, S. 4). So kann bspw. auf einer Produktionsstrecke für verschiedene Produkttypen der Roboter je nach erkanntem Bauteil den richtigen Programmablauf abrufen (VDMA 2014, S. 4). Der Verband Deutscher Maschinen- und Anlagenbau (VDMA) zieht als Indikator für den Automatisierungsgrad und die Entwicklung der Industriellen Bildverarbeitung die Anzahl der verkauften Industrieroboter heran: Diese wird zwischen 2015 und 2017 allein in China um schätzungsweise 25 % wachsen (VDMA 2014).<sup>2</sup> Viele davon werden mit einem „Robot Vision“ System ausgestattet sein, einer Verbindung aus Roboter und Bildverarbeitungssystem (VDMA 2014, S. 4; 21).

---

<sup>1</sup>Prognose

<sup>2</sup>Vor den aktuellen Turbulenzen an der chinesischen Börse und den gedämpften Wirtschaftszahlen für 2015.

Diese Arbeit ist Teil des Projekts Roboterbasierte, hochgenaue Qualitätssicherungssysteme (RoHoQ) am Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb) der Technischen Universität München. Ziel des Projektes ist es „die Genauigkeit von roboterbasierten Messsystemen zur Qualitätssicherung ohne aktives externes Messsystem auf  $\pm 0,1$  mm“ zu steigern und den Messvorgang zu automatisieren (ULRICH 2013). Projektpartner sind die Perceptron GmbH und die Audi AG. Konkreter Anlass ist das Vermessen von Karosserieteilen zur Qualitätssicherung. Dies wird zurzeit taktil mit einem Koordinatenmessgerät (KMG) realisiert und benötigt Stunden (WECKENMANN 2012, S. 77). Für eine Bohrung müssen mehrere Kontrollpunkte erfasst werden. Ein Tastsensor fährt jeden Punkte einzeln an. Schneller als solche taktilen Messungen sind optische Verfahren, bei denen bspw. mit einem Laser viele Punkte gleichzeitig erfasst werden (SCHWARZ 2015; WECKENMANN 2012, 129f).

In dem Projekt wird untersucht wie dies schneller und kostengünstiger mit einem Industrieroboter und einem Laserscanner realisiert werden kann. Damit der Roboter die Messpunkte anfahren kann, muss er wissen, wo sich das Werkstück – in diesem Fall eine Autotür – befindet. Dafür muss die Position des Werkstücks exakt vermessen werden. Das erfolgt bspw. mit einem externen Laser-Tracker und ist aufwändig und zeitintensiv.

In dieser Arbeit wird ein Bildverarbeitungssystem entwickelt, das das Werkstück selbstständig identifiziert und dem Roboter ermöglicht dieses anzufahren. Der Aufwand für das Einmessen der Tür bzw. deren exakte Positionierung entfällt somit gänzlich. Dies führt zu einer Kosten- und Zeitersparnis im gesamten Messvorgang.

Es wird ein konkretes Beispiel aus der Praxis verwendet. Die hier beschriebenen Techniken und der erstellte Code sind jedoch allgemeingültig. Die Autotür steht repräsentativ für ein industrielles Werkstück. Es wird nicht die Tür als solche erkannt, sondern ausschließlich ihre markanten Bohrungen und Aussparungen. Diese sind in nahezu jedem industriellen Werkstück, insbesondere aber in (Blech-)Halbzeugen, vorhanden und deshalb universelle Merkmale.

### 1.2 Handlungsbedarf und Zielsetzung

Die konkrete Aufgabe folgt aus dem Projekt und dem Stand der Technik. Dieser wird im nächsten Kapitel ausführlich beschrieben. Die Aufgabenbeschreibung als Motivation für diese Arbeit wird hier vorweggenommen. Der Handlungsbedarf ergibt sich aus der aufwändigen Positionierung bzw. dem zeitintensiven Einmessen des Werkstücks durch einen Werker. Der Nutzen sind die Zeitersparnis, den ein autonomes System ermöglicht,

sowie den entfallenden Kosten für ein externes Messsystem ( z. B. einen Laser-Tracker ). Das Projekt RoHoQ entwickelt ein Messsystem, das sich gegenüber einem kartesischen Koordinatenmessgerät durch eine signifikante Zeitersparnis und Kostenreduktion auszeichnet. Diese soll durch ein automatisches Anfahren an das Werkstück weiter verbessert werden.

### 1.2.1 Arbeitsbausteine

Um dem Roboter ein automatisches Anfahren des Werkstücks zu ermöglichen, muss es zunächst erkannt und dessen Position relativ zum Roboter bestimmt werden. Die Aufgabe kann grob in diese beiden Bereiche unterteilt werden, welche sich in weitere Subbereiche gliedern:

- Objekterkennung
  - Bilderzeugung
  - Vorverarbeitung
  - Detektion von Merkmalen in Kameraaufnahme und CAD-Modell
  - Vergleich dieser Merkmale
- Positionsbestimmung ( durch Stereovision )
  - Erzeugen einer Stereoaufnahme
  - Vorverarbeitung
  - Bestimmen korrespondierender Merkmale zwischen den Kamerabildern
  - Bestimmung der Kameraparameter ( Kalibrierung )
  - Triangulation

Die einzelnen Schritte überschneiden sich teilweise. So wird bei der Detektion korrespondierender Merkmale in einem Stereobildpaar auf Algorithmen aus der Objekterkennung zurückgegriffen.

Die hier beschriebene grobe Gliederung wird in der Arbeit beibehalten. Es wird jedoch immer wieder auf das Zusammenspiel mit den anderen Komponenten Bezug genommen.

### 1.2.2 Randbedingungen

Die Randbedingungen für diese Arbeit sind fest vorgegeben und führen zu einer reinen Software-Erweiterung, durch die der Roboter befähigt wird, seine Umgebung wahrzunehmen.

1. keine Veränderung der Zelle
  - a) keine zusätzliche Beleuchtung
  - b) kein spezieller Hintergrund
  - c) Beachten der Zellengröße
2. keine Veränderung des Roboters
  - a) Verwenden der integrierten Kamera des Laserscanners
  - b) Beachten der Reichweite des Roboters

Auf diese Bedingungen wird in der gesamten Arbeit verwiesen und an geeigneter Stelle erklärt, welche Konsequenzen diese haben.

Insbesondere die sehr restriktiven Randbedingungen zur Umgebung, unterscheidet das System von der heutigen Praxis, in der die Umgebung auf das Bildverarbeitungssystem angepasst wird (vgl. Abschnitt 2.3). Das Verwenden nur einer Kamera zur Erzeugung einer Stereoaufnahme ist in der Praxis ebenfalls nicht üblich (SUK & BHANDARKAR 1992a).

Die Festlegung auf eine nicht primär zur Objekterkennung gedachten Kamera bereitet Schwierigkeiten, die besondere Vorverarbeitungsschritte erfordern.

### 1.3 Aufbau der Arbeit

Die Gliederung dieser Arbeit ist an die Arbeitsbausteine aus Abschnitt 1.2.1 angelehnt. Zunächst wird in Kapitel 2 der aktuelle Stand der Technik erörtert. Dabei wird auf die Gebiete Bilderkennung und Stereovision allgemein, sowie speziell in Verbindung mit einem Roboter in der Industrie eingegangen. Die Kapitel 3, 4 und 5 behandeln ausführlich Methoden der Bildverarbeitung. Dabei werden nur Verfahren erläutert, die in dieser Arbeit verwendet werden. Der Umfang der Theorie sind dem fehlenden Vorwissen zur Bildverarbeitung im Maschinenbau geschuldet. Die Implementierung der Methoden wird anschließend in Kapitel 6 erklärt. Dabei wird auf Besonderheiten des Systems und dessen Randbedingungen eingegangen. Die Ergebnisse werden in Kapitel 7 diskutiert.

In Kapitel 8 werden die Ergebnisse zusammengefasst und es wird ein Ausblick über die weitere Entwicklung von Bildverarbeitungssystemen für Industrieroboter gegeben und auf weitere Forschungsfelder eingegangen.

Anhang A.1 und A.2 enthalten ergänzende Abbildungen und Tabellen auf die in der Arbeit verwiesen wird. Anhang A.3 beschreibt das Konzept eines neuen Algorithmus' für die Zuordnung von Linien. Anhang A.4 skizziert das erstellte Programm.

## **2 Stand der Technik**

In diesem Kapitel wird ein Überblick über verschiedene Systeme und speziell aktuelle Entwicklungen im Bereich der „Robot Vision“ gegeben.

### **2.1 Bildererkennung**

Aus Gründen der Übersichtlichkeit wird dieser Abschnitt in zwei Unterabschnitte geteilt: Objekterkennung und Robot Vision. Ersteres beschreibt die allgemeine Anwendung von Bildverarbeitung zur Extraktion von Information aus Bildern. Letzteres berücksichtigt dabei besonders den Bezug zur Hardware – unabhängig ob Fahrzeug, Flugobjekt oder Roboter.

#### **2.1.1 Objekterkennung**

Bildererkennung ist in unserem heutigen Alltag allgegenwärtig. Quick Response Codes (QRs) finden sich als physikalischen Links auf vielen Plakaten und Objekten (BELUSSI & HIRATA 2011, S. 1). Die Analyse dieser 2D Barcodes erfolgt in zwei Schritten: zuerst Detektion des Codes und anschließend Extraktion der Information. Die Information kann mit Schwellenwertoperationen (GU et al. 2011) oder Konturerkennungsalgorithmen erfolgen (BHARATH 2014). Diese Methoden werden auch in der vorliegenden Arbeit verwendet und in Kapitel 3 und Abschnitt 4.2 erklärt. BELUSSI & HIRATA beschreiben ein effizientes, viel zitiertes Verfahren zur Detektion von QRs in beliebigen Aufnahmen. Dabei werden Blockfilter verwendet, um den Informationsgehalt im Bild zu reduzieren (BELUSSI & HIRATA 2011, S. 3). Filter werden in Abschnitt 3.2.2 erklärt.

Die Gesichtserkennung ist ein aktuelles Forschungsthema der Bildverarbeitung. Internetkonzerne wie Google (Alphabet Inc.) oder facebook (Facebook Inc.) benutzen diese Technologie bereits in ihren Sozialen Netzwerken (O'TOOLE 2014). Dazu

werden Bildverarbeitungsmethoden mit Algorithmen aus dem maschinellen Lernen kombiniert (TAIGMAN et al. 2014, 1f). TAIGMAN et al. haben ein System vorgestellt, das genauso zuverlässig funktioniert wie bei Menschen. Dafür werden die Bilder in ein 3D Modell umgerechnet und anschließend mit einem „Deep Neural Network“ analysiert (TAIGMAN et al. 2014). Der Algorithmus wird mit Bildern aus dem Sozialen Netzwerk facebook trainiert und an Standard-Datenbanken zur Gesichtserkennung getestet (TAIGMAN et al. 2014, 4f). Die Zuverlässigkeit liegt bei 97,35 %, Menschen schaffen 97,53 % (TAIGMAN et al. 2014; O'TOOLE 2014).

Mit Hilfe von Bilderkennungsalgorithmen werden (Hand-)Schriften erkannt und digitalisiert (OLIVEIRA JR. et al. 2009). Dazu können unterschiedliche Merkmale analysiert werden. Eine einfache Methode zur Erkennung von Blockschrift wird in Abschnitt 4.2.3 beschrieben (BRADSKI & KAEHLER 2008, S. 255).

Zur eindeutigen Identifikation von Fingerabdrücken werden Merkmale wie Schleifen oder Ecken im Abdruck der Papillarleisten detektiert und verglichen (FADZILAH & DZULKIFLI 2009). Mittlerweile verwenden auch Endverbrauchergeräte wie das Apple iPhone (ab Version 5s) einen Fingerabdrucksensor, der jedoch keinen hohen Zugangsschutz darstellt (STARBURG 2014).

Bildverarbeitungsmethoden werden auch zum Erkennen von Bildern der Iris verwendet. CAMPOS et al. stellten auf der *14th Iberoamerican Conference on Pattern Recognition* ein Verfahren vor, das 99,867 % Genauigkeit erreicht.

Die Erkennung von Objekten geschieht häufig mit Hilfe neuronaler Netze (bspw. (YUAN 2004)). Die künstliche Intelligenz ist ein aktiver Forschungsbereich in der Bildverarbeitung mit Bezug zu „Big Data“ (HOF 2013). Sie wird in erster Linie von IT-Firmen wie Google oder Facebook betrieben, soll aber in Zukunft auch Anwendung in der Industrie bspw. bei der Inspektion finden (HOF 2013)

Die Automobilindustrie setzt verstärkt auf Kamera basierte Fahrerassistenzsysteme. Die ersten Systeme wurden zur Erkennung von Verkehrsschildern entwickelt. Mobileye stellte 2008 das erste kommerzielle System zum Erkennen von Geschwindigkeitsbegrenzungen und Überholverbotszonen vor (MOBILEYE 2014). Mittlerweile bietet Bosch (Robert Bosch GmbH) dies auch als App für mobile Endgeräte an (BOSCH 2014). Die Detektion von Verkehrsschildern beruht üblicherweise auf Farbe und Geometrie der Verkehrszeichen (EICHNER & BRECKON 2007). EICHNER & BRECKON benutzen den RANSAC Algorithmus, der in Abschnitt 5.2.3 erklärt wird, zur Detektion von Formen.

Da es nur eine begrenzte Anzahl definierter Verkehrsschilder gibt, ist das Erkennen dieser ein Klassifizierungsproblem (SEO et al. 2015). Dies geschieht meist mit Methoden aus dem maschinellen Lernen, Neuronalen Netzen oder „Support Vector Mashines (SVMs)“ (EICHNER & BRECKON 2007; GREENHALGH & MIRMEHDI 2012). GREENHALGH & MIRMEHDI verwenden eine SVM, nachdem sie die Aufnahmen mit Schwellenwertoperation und dem Sobel-Filter (s. Kapitel 3) bearbeitet haben. Aktuell wird an der Weiterentwicklung dieser Systeme geforscht, sodass bspw. auch Baustellenbegrenzungen und Fahrbahnverengungen erkannt werden können (SEO et al. 2015).

Ein weiteres Anwendungsfeld, in dem Objekterkennung genutzt wird, ist die Virtuelle Realität (VR). Diese wird durch eine Brille mit integriertem Bildschirm zugänglich. Dabei werden reale Objekte erkannt und in die virtuelle Umgebung integriert (AZUMA 1997, S. 28). Methoden zur Objekterkennung werden auch zum Nachverfolgen der Kopfposition des Trägers verwendet (AZUMA 1997, 28f).

### 2.1.2 Robot Vision

Ein großer Bereich der Forschung für autonome Roboter ist Simultaneous Localization And Mapping (SLAM). Mobile Roboter erstellen automatisch Karten zur eigenständigen Navigation. Dazu wird die Umgebung z. B. mit einem 3D Laserscanner abgetastet und die Daten bspw. mit dem RANSAC Algorithmus (Abschnitt 5.2.3) ausgewertet (NÜCHTER & HERTZBERG 2008, S. 919). Die videobasierte Navigation mobiler Roboter, insbesondere mit einfacher Hardware, ist ebenfalls ein aktiver Forschungsbereich (SHOJAEIPOUR et al. 2009; TAYLOR & KRIEGMAN 1998). In Bildern werden Kanten detektiert, über die Hindernisse erkannt werden, denen der Roboter automatisch ausweicht (SHOJAEIPOUR et al. 2009). Mobile Serviceroboter werden in Museen, wie dem Louvre, eingesetzt und müssen dort in einer sich schnell verändernden Umgebung navigieren können (TAYLOR & KRIEGMAN 1998).

Eine Alternative zur Navigation über GPS ist die Zuhilfenahme von Orientierungspunkten (englisch „landmark“). Diese müssen zuverlässig erkannt werden, um die relative Position zu diesen zu bestimmen (ADÁN et al. 2009, S. 21). ADÁN et al. stellen ein System vor zur Navigation von Fußgängern in virtuellen Umgebungen. Visuelles SLAM wird auch für selbst fahrende Autos (HAMME et al. 2015) und Unmanned Air Vehicles (UAVs) untersucht (ZHANG et al. 2015). So stellen HAMME et al. ein auf einer

einzigem Kamera basierendes System vor, das die Position eines Autos relativ zu einem vorherigen Punkt bestimmt. ZHANG et al. lassen einen Quadrokopter eine definierte Trajektorie nachfliegen.

Seit 1996 findet der „RoboCup“ statt, bei dem mobile Roboter gegeneinander Fußball spielen. Seit 2003 gibt es eine Liga für humanoide Roboter (ROBOCUP 2015). Die Regularien legen insbesondere die farbliche Markierung des Spielfeldes und der Gegner fest, sodass die Detektion der einzelnen Objekte (Tor, (Mit-)Spieler, Ball) durch eine sehr einfache Farberkennung geschehen kann (ROBOCUP 2015).

### 2.2 Stereovision

Seit 1922 werden Filme mit Stereokameras in 3D produziert (Titel: „The Power of Love“). Die 3D Illusion wird mit verschiedenen Techniken (bspw. anaglyphische -, polarisierte -, shutter Brillen) dem Zuschauer vermittelt (SCHEDEEN 2010, 1f). Die Aufnahme der Bilder beruht auf der Stereovision (SCHEDEEN 2010, S. 4).

Die 3D Wahrnehmung ist eine entscheidende Aufgabe im autonomen Fahren und Stereokameras sind eine kostengünstige Lösung dafür (GEHRIG et al. 2009, S. 134). Mit dem Stereosystem können sogenannte *Tiefenkarten* erzeugt werden. Diese stellen die räumliche Information grafisch dar (vgl. (GEHRIG et al. 2009, S. 292; BRADSKI & KAEHLER 2008, S. 454)). Bei sicherheitskritischen Systemen kommt es besonders auf die Zuverlässigkeit und Laufzeit der Algorithmen an, welche STEINGRUBE et al. eingehend untersucht haben.

Zur 3D Rekonstruktion werden nicht immer Stereobilder benötigt. Dasselbe Objekt muss aus unterschiedlichen Perspektiven aufgenommen sein (BAY et al. 2007, S. 356). Die 3D Rekonstruktion von Gebäuden aus Luft- oder Satellitenbildern wird bspw. von Google Earth oder Herstellern von Navigationssystemen betrieben (BEHLEY & STEINHAGE 2009, S. 164). Dabei wird die Höhe von Objekten anhand der Verschiebung markanter Punkte – wie Ecken – in unterschiedlichen Aufnahmen bestimmt.

### 2.3 Anwendungen in der Praxis

Das klassische Beispiel in der Robot Vision und dem Maschinellen Sehen ist der „Griff in die Kiste“. Ungeordnet in einer Kiste liegende Werkstücke müssen erkannt und eine Greifstrategie für diese entwickelt werden. Immer wieder werden Kamera-Roboter-Systeme für solche chaotischen „pick-and-place“ Aufgaben eingesetzt (STADLMEYER 2011; BATCHELOR 2012, S. 637). Der „Griff in die Kiste“ beschreibt ein Problem der Bildverarbeitung, das „pick-and-place“ ist eine Herausforderung für den Roboter. Das Problem ist auch heute noch ein aktuelles Forschungsthema, mit dem sich bspw. das Fraunhofer Fraunhofer Institut für Produktionstechnik und Automatisierung (IPA) beschäftigt (SPENRATH 2015).

Viele in der Industrie eingesetzten Bildverarbeitungssysteme dienen der Qualitätssicherung und ersetzen die Sichtprüfung durch den Menschen (CIUPEK 2014; TREIBER 2010, S. 3). So hat das Fraunhofer Institut für Fabrikbetrieb und -automatisierung (IFF) ein Robot Vision System entwickelt, das Nietverbindungen bei Flugzeugrumpfschalen inspiziert und den Zeitaufwand dafür verglichen mit einem Werker<sup>3</sup> um bis zu 70% reduziert (BÖTTGER 2014). Aktuell wandelt sich der Einsatz von Bildverarbeitung in der Fertigung vom reinen Inspektionssystem hin zu einem integralen Bestandteil der Steuerung und Regelung (VDMA 2014, S. 5). Trends, z. B. sich schleichend verändernde Maße, werden während der Produktion erkannt und direkt in der Regelung berücksichtigt. So optimiert die Bildverarbeitung den Prozess (WENDEL 2014). Olaf Munkelt vom VDMA beschreibt eine breite Anwendbarkeit von Bildverarbeitung und sieht in Zukunft Industrieroboter nur noch mit integrierter Kamera (MUNKELT & BÖTTGER 2015).

Im Bereich der automatischen Prüftechnik sinken die Zykluszeiten (KOTHE 2009). Optische Verfahren können den Messvorgang beschleunigen (SCHWARZ 2015). Dabei kommen zunehmend Multi-Sensor-Geräte zum Einsatz, bei denen häufig Bildverarbeitungssensoren integriert sind (KOLL 2015). Die Gesellschaft für Optische Messtechnik mbH (GOM) bietet eine Roboter-Messzelle zur vollautomatischen Inspektion beliebiger Bauteile mit einem 3D-Kamerasystem an (GOM 2013). Alle bereits heute kommerziell angebotenen Systemen gemein ist die kontrollierte Umgebung, in der sie arbeiten. Hintergrund und Beleuchtung werden optimal auf die Bildverarbeitung eingestellt.

Industrielle Systeme verwenden häufig Auflichtverfahren, bei denen das Werkstück durch die Helligkeit vom Hintergrund abgegrenzt wird (TREIBER 2010, 83ff; BATCHELOR 2012, S. 12). So können Umrisse/Formen gut erkannt werden – insbesondere, wenn

---

<sup>3</sup>In dieser Arbeit wird die männliche Wortform stellvertretend für alle Geschlechter verwendet.

die Werkstücke vereinzelt sind. Sie dienen der geometrischen 100%-Kontrolle auch bei hohen Stückzahlen und Geschwindigkeiten (DRUNK 2015).

Mit Stereokamerasystemen können 3D-Informationen aus 2D Bildern gewonnen werden. Auch in der Fertigung finden diese Anwendung, z. B. zur 3D-Lagebestimmung für einen Greifer (BATCHELOR 2012, S. 637). Oft handelt es sich dabei um fest installierte Systeme mit bekanntem Abstand zum Objekt, mit denen die 3D-Koordinaten berechnet werden (DRUNK 2015). Stereo-Systeme an einem Industrieroboter finden auch bereits Anwendung (GOM 2013). Heute sind die meisten Verfahren für Bildverarbeitung und Robotik für den 2D Bereich entwickelt (FILI 2008). Die 3D-Bildverarbeitung ist sehr anspruchsvoll (CIUPEK 2014). Enis Ersü, Vorstandsvorsitzender der Isra Vision AG, bescheinigt ihr jedoch hohes Potential im Bereich der 3D-Roboterführung (FILI 2008).

Sogenannte Mono-Stereo-Systeme – auch „Motion Stereo“ genannt, bei denen eine Kamera zwischen den Aufnahmen für das Stereobild verfahren wird, finden in der Praxis kaum Anwendung (SUK & BHANDARKAR 1992a, S. 28). Auch heute noch gibt es keine kommerziellen Beispiele für ein solches System.

## 3 Grundlagen der Bildverarbeitung

### 3.1 Bildrepräsentation und Farbräume

Ein Bild ist die 2D Projektion einer 3D *Szene*. Der Sensor einer Kamera speichert die (Licht-)Information in einem Bildpunkt, einem *Pixel*. Die Summe aller Pixel ist die *Auflösung* des Sensors. Die Pixel werden von der linken oberen Bild- bzw. Sensorecke aus gezählt (siehe Abbildung 3.1).

Ein Pixel speichert die Helligkeit oder *Intensität* zu einer räumlichen Position auf dem Kamerasensor. Die Intensität  $I$  bezieht sich dabei immer auf eine „Grundfarbe“. So erhält man ein klassisches Graustufenbild als 2D Matrix von Intensitätswerten der Farbe Weiß (100%: Weiß, 0%: Schwarz). Intensitäten werden üblicherweise als 8-bit oder 16-bit Zahl kodiert (BARTHEL 2003, S. 249). Werden Intensitätsmatrizen unterschiedlicher „Grundfarben“ überlagert, kann ein Farbbild erzeugt werden. Der Sensor einer Farbkamera besteht genau genommen aus mehreren Sensoren unterschiedlicher Empfindlichkeit für bestimmte Wellenlängen des Lichts. Üblicherweise werden Farben in einem Kamerasensor und am Computer als Überlagerung von Rot, Grün und Blau dargestellt<sup>4</sup>, dem RGB-Farbraum (BARTHEL 2003, S. 248). Das Farbbild wird als 3D Matrix mit unterschiedlichen *Farbkanälen* dargestellt:  $(u, v, c) \rightarrow I; c \in [R, G, B]$ .

Ein anderer bekannte Form der Farbkodierung ist CMYK (englisch „cyan, Magenta, Yellow, Black“). Dieser Farbraum wird zum Drucken verwendet. Darüber hinaus wird in der Bildverarbeitung häufig der HSV (englisch: „Hue, Saturation, Value“) eingesetzt, in dem eine Farbe über den Farbwert, die Sättigung und die Helligkeit definiert wird.

Die Matrizen können vollständig oder komprimiert gespeichert werden. Ein bekanntes Kompressionsformat ist JPEG (englisch „Joint Photographic Experts Group“, Datei-Endung: jpg). Pixel ähnlicher Farben werden blockweise zusammengefasst (BARTHEL 2003, S. 254). Das TIFF (englisch „Tagged Image File Format“, Datei-Endung: tif) ist ein

---

<sup>4</sup>Das liegt daran, dass auch das menschliche Auge Sinneszellen für diese drei Farben hat (BARTHEL 2003, S. 248).

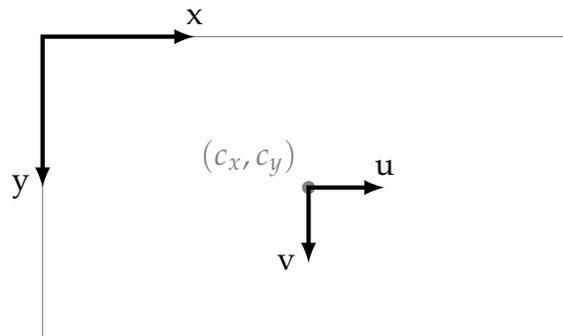


Abbildung 3.1: Verschiedene Bildkoordinatensysteme.  $(c_x, c_y)$  optischer Mittelpunkt.

gebräuchliches verlustloses Dateiformat (VAN DAMME 2015).

Der verwendete Perceptron-Sensor liefert ein 8-bit Graustufenbild mit  $968 \times 1296$  Pixeln im TIFF Format. Die Intensitätswerte – in diesem Fall gleichbedeutend mit Grauwerten – liegen also im Intervall  $[1, 256]^5$ .

## 3.2 Einfache Bildverarbeitungsmethoden

### 3.2.1 Histogrammanpassung

Der Kontrastumfang eines Bildes – auch *Dynamik* genannt – hängt stark von der Beleuchtung der Szene bzw. dem Licht ab, das auf den Kamerasensor fällt (BRADSKI & KAEHLER 2008, S. 187). Das menschliche Auge hat Schwierigkeiten, niedrige Intensitätsunterschiede (geringe Kontraste) zu erkennen (MOESLUND 2012, S. 44). Viele Algorithmen zur Objekterkennung basieren auf dem Bildgradienten und damit auf einem (lokal) hohen Kontrast (vgl. Abschnitt 3.3 und 4.3).

Mit Histogrammen kann man die Verteilung von Intensitätswerten (in einem Graustufenbild die Werte  $[1, 256]$ ) messen. Ein Histogramm ist allgemein die grafische Darstellung der Häufigkeit von Ereignissen (MOESLUND 2012, S. 50). Ein Charakteristikum wird anhand seines Werts in definierte Körbe (englisch „bins“) unterteilt (BRADSKI & KAEHLER 2008, S. 193). Das Intensitätshistogramm zeigt die Häufigkeit einzelner Grauwerte. Ein ideales Histogramm ist über den ganzen Intensitätsbereich gleich verteilt (BRADSKI & KAEHLER 2008, S. 188).

<sup>5</sup>Wie bei der Indizierung von Arrays wird beim Programmieren das Intervall  $[0, 255]$  verwendet.

Abbildung 3.2 (a) zeigt eine Aufnahme des Sensors am Roboter und das dazugehörige Histogramm. Das Histogramm ist sehr schmal (niedrige Dynamik) mit einzelnen Spitzen. Die Dynamik kann nachträglich angepasst werden, indem der Intensitätsbereich gestreckt wird (BRADSKI & KAEHLER 2008, S. 188; OPENCV 2015b). Auf diese Weise wird das Bild robuster gegenüber Änderungen in der Beleuchtung (BRADSKI & KAEHLER 2008, 188f).

Die Histogrammäquilisation oder Histogrammanpassung wird dazu am häufigsten verwendet (BRADSKI & KAEHLER 2008, S. 188). Dabei wird das ursprüngliche Histogramm mit Hilfe einer nichtlinearen Abbildungsfunktion auf eine größere, einheitlichere Verteilung skaliert (BRADSKI & KAEHLER 2008, 188f; MOESLUND 2012, S. 53). Die Abbildungsfunktion  $C$  ist die laufende Summe der Originalverteilung – also die kumulierten Grauwerte im Histogramm  $H$  bis zum betrachteten Grauwert  $j$ :

$$C[j] = \sum_{i=0}^j H[i] \quad i, j \in [0, 255] \quad (3.1)$$

$$G_{eq}[x, y] = C_{norm} [G[x, y]] \quad (3.2)$$

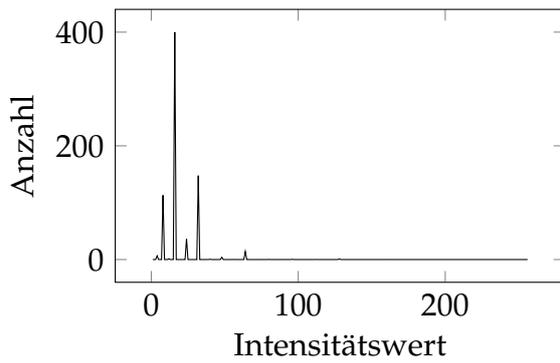
Die Grauwerte  $G$  des ursprünglichen Bildes werden über die normierte *kumulierte Dichte*  $C_{norm}$  auf das angepasste Bild  $G_{eq}$  abgebildet. Häufige Grauwerte werden auseinander gezogen, Grauwertbereiche geringer Häufigkeit auf ein (weniger großes) Intervall gestaucht. Die Histogrammäquilisation ist dadurch verlustbehaftet. Die Information vieler Pixel wird deutlicher auf Kosten der Unterscheidbarkeit weniger Pixel.

Abbildung 3.2 (b) zeigt das angepasste Bild mit Histogramm. Man sieht gut, dass das Histogramm nun den gesamten Grauwertbereich abdeckt und der Abstand der diskreten Grauwerte nun größer ist.<sup>6</sup> Der Kontrast wird erhöht und bei dem verwendeten Kamerasensor das Bild in einen für das menschliche Auge unterscheidbaren Graustufenbereich transformiert.

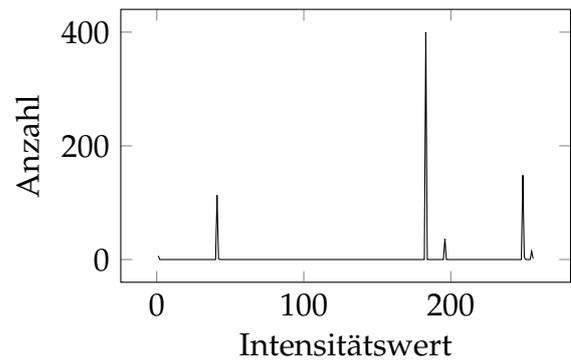
Das Bild ist immer noch empfindlich gegenüber Beleuchtungsänderungen. Sehr helle und sehr dunkle Bildbereiche – wie sie durch direkte Sonnenstrahlen oder Schatten entstehen können – beeinflussen die Histogrammanpassung stark, indem sie die

---

<sup>6</sup>Bei einer kontinuierlichen Verteilung der Intensitätswerte im Ausgangsbild ist das Ergebnis ein exakt gleich verteiltes Histogramm im angepassten Bild (BRADSKI & KAEHLER 2008, S. 189).



(a) Histogramm des Ursprungsbilds



(b) angepasstes Histogramm



(a)



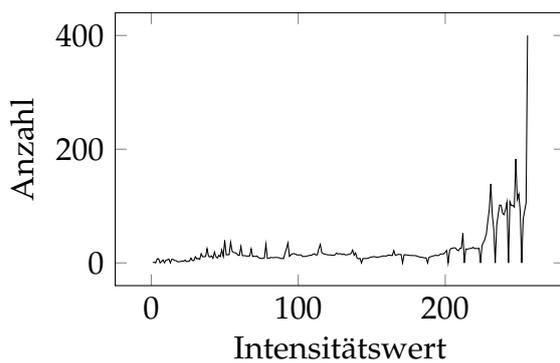
(b)



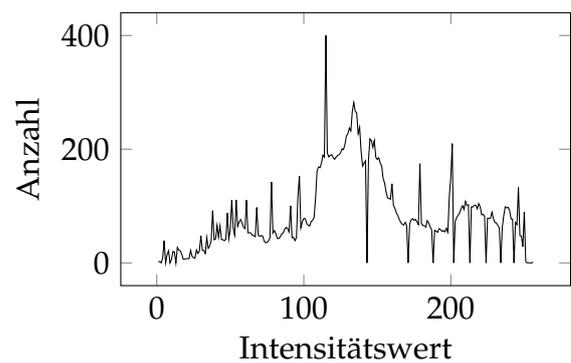
(c)



(d)



(c) Adaptive Histogram Equalization



(d) Contrast Limited Histogram Equalization

Abbildung 3.2: Histogrammanpassung. Die Histogramme zeigen wie häufig ein bestimmter Intensitätswert  $[1, 256]$  im Bild vorkommt. (a) Originalbild; (b) Bild mit angepasstem Histogramm; (c) Bild mit AHE auf  $6 \times 6$  Bildregionen angewendet; (d) Bild mit CLAHE auf  $6 \times 6$  Bildregionen und einem Limit von 100.

kumulierte Dichtefunktion beeinflussen. Dem kann mit der *adaptiven Histogrammanpassung* (englisch: „Adaptive Histogram Equalization (AHE)“) entgegengewirkt werden (HECKBERT 1994, S. 474). Bei der adaptiven Histogrammanpassung wird das Bild in mehrere Regionen unterteilt und für jedes Teilbild eine Histogrammanpassung durchgeführt. Üblicherweise wird ein Bild in  $8 \times 8 = 64$  Unterregionen geteilt (HECKBERT 1994, S. 475). Diese Methode orientiert sich am menschlichen Auge, das sich an die lokalen Kontrastverhältnisse anpasst (HECKBERT 1994, S. 475).

Die adaptive Histogrammanpassung reagiert – genau wie die (normale) Histogrammanpassung – empfindlich auf Bildrauschen. Ausreißer verfälschen den Kontrastumfang. Dieser wird aber bei der Histogrammanpassung, abhängig von der Häufigkeit der Intensitätswerte, auseinandergedrückt (BRADSKI & KAEHLER 2008, 188f). Die Verstärkung von Bildrauschen ist der größte Nachteil der adaptiven Histogrammanpassung (HECKBERT 1994, S. 475). Diesem kann mit Filtern (siehe Abschnitt 3.2.2) begegnet werden oder direkt mit dem CLAHE Algorithmus, welcher die Kontrastverbesserung in homogenen Bereichen begrenzt (HECKBERT 1994, S. 476).

Die *Contrast Limited Histogram Equalization (CLAHE)* verteilt die Teile des Histogramms, die über einem festgelegten Schwellwert liegen, gleichmäßig auf das ganze Histogramm um (HECKBERT 1994, S. 476).<sup>7</sup> Dadurch wird der Kontrast begrenzt und Bildrauschen effektiv unterdrückt (vgl. Abbildung 3.2 (d)). Der Schwellwert ist ein Kompromiss zwischen dem Unterdrücken von Bildrauschen und einem hohen Kontrast zum Erkennen von Objekten. Abbildung A.2 im Anhang zeigt die angepassten Bilder und die dazugehörigen Histogramme von einer hochwertigeren Kamera.

Es ist zu beachten, dass zwischen den Intensitätswerten des Originalbildes und des angepassten Bildes keine eindeutige Beziehung mehr besteht. Deshalb sind CLAHE Bilder ungeeignet für quantitative Messungen, bei denen die Histogramme von Bildern verglichen werden (HECKBERT 1994, S. 477).

CLAHE wird in dieser Arbeit benutzt, um den Kontrast der Sensorbilder zu erhöhen – erst so werden die Sensoraufnahmen zugänglich für das menschliche Auge. Gleichzeitig werden sie unempfindlicher gegenüber Änderungen in der Beleuchtung. Dies ist wichtig, um auch bei wechselnden Lichtverhältnissen zuverlässig Objektmerkmale zu erkennen.

---

<sup>7</sup>Es handelt sich dabei um die Histogramme der adaptiven Histogrammanpassung. Die Häufigkeitswerte werden also in jeder Bildregion umverteilt.

### 3.2.2 Filter

Filter geben ein zu den Eingangswerten proportionales Signal zurück (BRAHMBHATT 2013, S. 41). In der Bildverarbeitung sind die Eingangswerte die Farbinsintensitäten der Pixel. Filter sind die Basis aller Operationen des Maschinellen Sehens (BRAHMBHATT 2013, S. 65). Sie werden zum Glätten<sup>8</sup> verwendet, um bspw. Bildrauschen zu unterdrücken, oder Formen im Bild herauszuarbeiten (BRADSKI & KAEHLER 2008, S. 109; PARIS et al. 2007). Stellvertretend für eine Vielzahl unterschiedlicher Filter werden im Folgenden drei in dieser Arbeit verwendete Filter erklärt.

Ein sehr effektiver Filter zur Unterdrückung von sogenanntem Salz & Pfeffer-Rauschen ist der Median-Filter (LAGANIERE 2014, S. 178). Der gefilterte Grauwert eines Pixels ist der Median<sup>9</sup> aller Grauwerte seiner Umgebung. So werden extreme Grauwerte wie Weiß („Salz“) und Schwarz („Pfeffer“) unterdrückt. Die Umgebung ist ein Quadrat mit einstellbarer Kantenlänge um den betrachteten Pixel herum. Abbildung 3.3 (b) zeigt das Ergebnis eines Median-Filters über eine  $5 \times 5$  großen Umgebung.

Damit bereits gefilterte Pixel nicht die nächsten Pixel beeinflussen, werden Filter  $f(x, y)$  in umgekehrter Reihenfolge angewendet: vom Zielbild  $dst(x, y)$  auf das Originalbild  $src(x, y)$ .

$$dst(x, y) = src( f_x(x, y), f_y(x, y) ) \quad (3.3)$$

Anstatt den Median der umgebenden Grauwerte zu verwenden, kann auch der gewichtete Mittelwert der gesamten Umgebung gebildet werden. Besonders beliebt ist der Gauß-Filter (BRADSKI & KAEHLER 2008, S. 112), dessen Gewichte wie eine 2D Gauß-Kurve vom Zentralpixel ausgehend abfallen.

Der Ausgang des Gauß-Filters  $GB()$  angewandt auf den Pixel  $p$  mit dem Intensitätswert  $I_p$  ist der Wert  $I_{p,GB}$ . Dies ist die Summe der Intensitätswerte  $I_q$  aus der Umgebung  $S$ , gewichtet mit der entfernungsabhängigen Gauß-Funktion  $G_\sigma(x)$  (PARIS et al. 2007,

<sup>8</sup>Glätten bedeutet die Intensitäten oder Grauwerte angrenzender Pixel ähnlicher zu machen (BRAHMBHATT 2013, S. 46; PARIS et al. 2007, S. 2).

<sup>9</sup>Der Median ist der mittlere Wert einer sortierten Reihe. Beispielsweise ist der Median der Reihe [2, 9, 8] die 8.

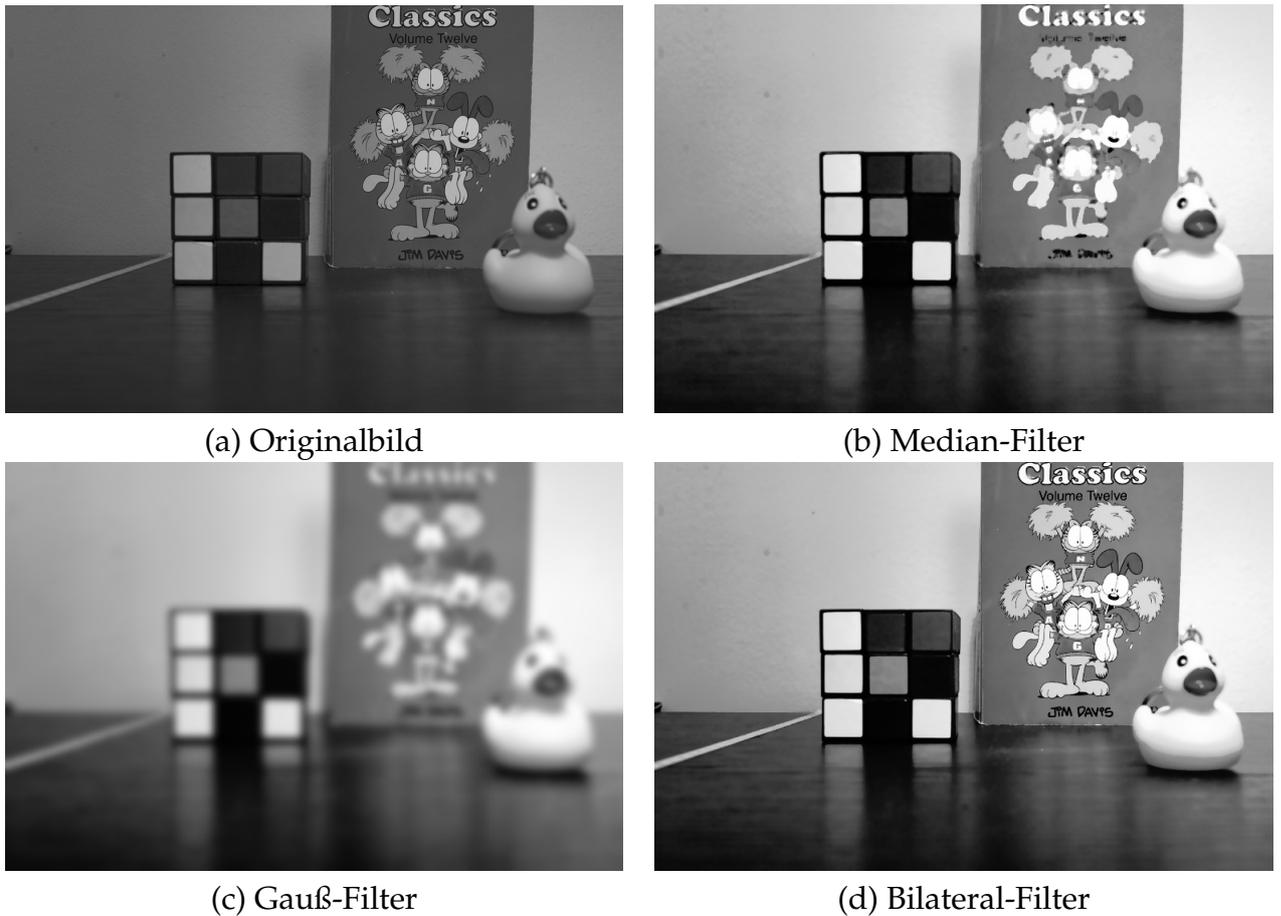


Abbildung 3.3: (a) ungefilterte Originalaufnahme; (b) Median-Filter Kernelgröße  $k = 5$ ; (c) Gauß-Filter  $\sigma = 5$ ; (d) Bilateral-Filter Kerlengröße  $d = 5$ ,  $\sigma_{color} = 13$ ,  $\sigma_{space} = 3,5$ . Die Details sind der Vollständigkeit halber hier angegeben. Für ihre genaue Bedeutung sei auf die OpenCV Dokumentation verwiesen: (OPENCV 2015c).

2-13f):

$$I_{p,GB} = GB(p) = \sum_{q \in S} G_{\sigma}(\|p - q\|) I_q \quad (3.4)$$

$$G_{\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (3.5)$$

Die Gauß-Funktion ist die Kernel-Funktion einer Faltung (BRADSKI & KAEHLER 2008, S. 112). Ein großes  $\sigma$  führt zu einer flachen Gauß-Kurve und einer starken Glättung

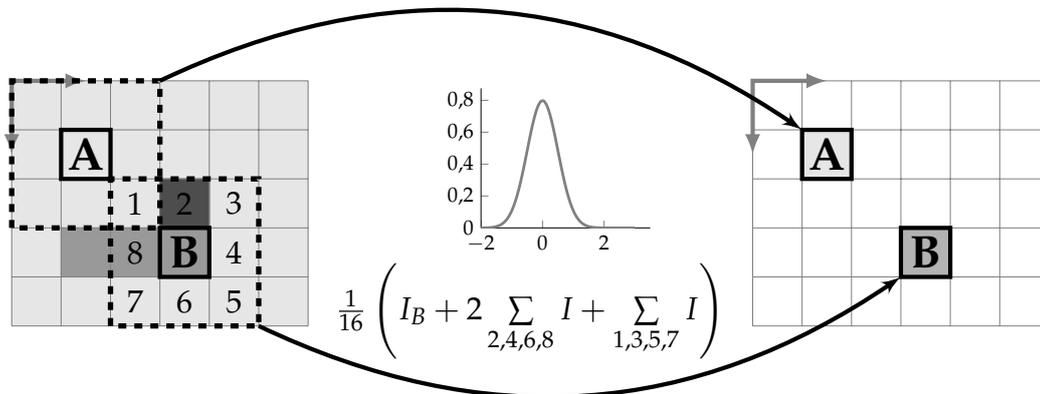


Abbildung 3.4: Gauß-Filter. Die nächsten Nachbarn in einer  $3 \times 3$  Umgebung werden zur Bestimmung des neuen Intensitätswertes hinzugezogen. Ihr Einfluss sinkt mit zunehmendem Abstand zum zentralen Pixel. Hier wird der Intensitätswert des zentralen Pixels A oder B mit  $3/4$  und die Nachbarn mit  $1/4$  gewichtet. Die Operation wird für jeden Pixel mit den Intensitätswerten aus dem Originalbild (links) durchgeführt und in einem neuen Bild (rechts) gespeichert.

(PARIS et al. 2007, S. 2-15). Ein Beispiel für den Kernel eines Gauß-Filters ist

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} .$$

Der Intensitätswert des mittleren Pixels wird am stärksten gewichtet, seine Umgebung wird mit einer Gauß-Kurve abfallend gewichtet. Die Gewichte sind in Summe auf 1 normiert. Das Prinzip wird in Abbildung 3.4 dargestellt, das Resultat in Abbildung 3.3 (c).

Mit dem Median- und dem Gauß-Filter werden die Intensitätswerte anhand des räumlichen Abstandes zu ihren Nachbarn geglättet (PARIS et al. 2007, S. 3-18). Bildmerkmale wie Kanten oder Ecken zeichnen sich durch einen hohen *Bildgradienten* – einen großen Intensitätsunterschied benachbarter Pixel – aus (mehr dazu in Abschnitt 3.3). Um Ecken oder Kanten zu erhalten, müssen die Intensitätswerte  $I$  der betrachteten Pixel berücksichtigt werden. Der Bilateral-Filter  $BF$  ist eine Erweiterung des Gauß-Filters um einen Intensitätsterm:

$$BF(I_p) = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q \quad (3.6)$$

$G_{\sigma_s}$  ist die Gauß-Funktion, die den Abstand ( $s$ : „spatial weight“) des mittleren Pixels

$p$  berücksichtigt.  $G_{\sigma_r}$  gewichtet den Intensitätsunterschied ( $r$ : „range weight“) zwischen dem mittleren Pixel und dem betrachteten Pixel  $q$  aus der Umgebung  $S$ .<sup>10</sup>  $W_p$  normalisiert die Summe auf das Intervall  $[0, 255]$ , den normalen Intensitätsbereich. Abbildung 3.3 (d) zeigt das Ergebnis eines Bilateral-Filters. Abbildung A.3 in Anhang A.1 zeigt das gefilterte Sensorbild.

Durch die Berücksichtigung der Intensität wird Bildrauschen wie starke Kanten behandelt. Beide weisen einen hohen Intensitätsunterschied zu ihrer Umgebung auf. Das Bild muss daher vorher z. B. mit dem Median-Filter geglättet werden (PARIS et al. 2007, S. 4-9). Der Bilateral-Filter ist aufwendig zu berechnen, da Gewichte aus zwei Gauß-Funktionen bestimmt werden müssen (PARIS et al. 2007, S. 3-30). Die Form des Kernels hängt vom Bildinhalt ab und nicht nur von der Größe der Umgebung (PARIS et al. 2007, S. 3-3).

Filter berechnen aus der Umgebung eines Pixels ein bestimmtes Mittel (BRAHMBHATT 2013, S. 46). Alleinstehend dienen sie i. d. R. als Vorverarbeitungsschritt. Die Sensorkamera ist für sehr helles Licht ausgelegt. Im normalen Lichtbereich ist die Kamera vergleichsweise unempfindlich und produziert deshalb ein hohes Bildrauschen. Dieses wird mit einer Kombination aus Gauß- und Bilateral-Filter unterdrückt (vgl. Abschnitt 6.2).

#### 3.2.3 Skalierung

Die Skalierung eines Bildes hat maßgeblichen Einfluss auf die Geschwindigkeit vieler Bildverarbeitungsoperationen. Die Anzahl der Pixeloperationen ist direkt proportional zur Auflösung (Anzahl der Pixel in einem Bild) und wächst somit quadratisch mit der Skalierung (bei gleichbleibendem Seitenverhältnis).

Es gibt verschiedene Methoden zur Skalierung von Bildern (BRADSKI & KAEHLER 2008, S. 130). Der einfachste Weg ein Bild auf die Hälfte zu verkleinern ist es, jeden zweiten Pixel zu löschen. Bei einer Vergrößerung werden die Pixel auseinandergezogen und die Lücken mit dem gleichen Wert des linken (Original-)Pixels aufgefüllt. Dies ist die Methode des *nächsten Nachbarn* (englisch *Nearest-Neighbor*) (BRADSKI & KAEHLER 2008, S. 130). Die Funktionsweise ist in Abbildung 3.5 dargestellt. Sie führt zu sehr harten („verpixelten“) Kanten und funktioniert nur bei Vielfachen des Ausgangsbildes. Abbildung 3.6 unten zeigt den Effekt an einem Bild. Bessere, glattere Ergebnisse werden durch Interpolation der Farb- bzw. Grauwerte erzielt. Bei der *inter-area Interpolation* ist der Grauwert des neuen Pixels die Summe der gewichteten Grauwerten des Umfelds

---

<sup>10</sup>Für  $\sigma_r = \infty$  erhält man  $G_{\sigma_r} = 1$  und damit den Gauß-Filter.

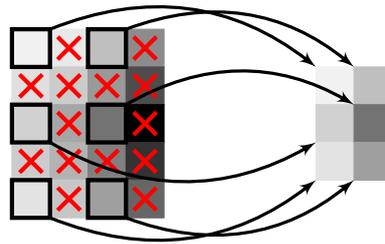


Abbildung 3.5: Die einfachste Art, ein Bild zu verkleinern.

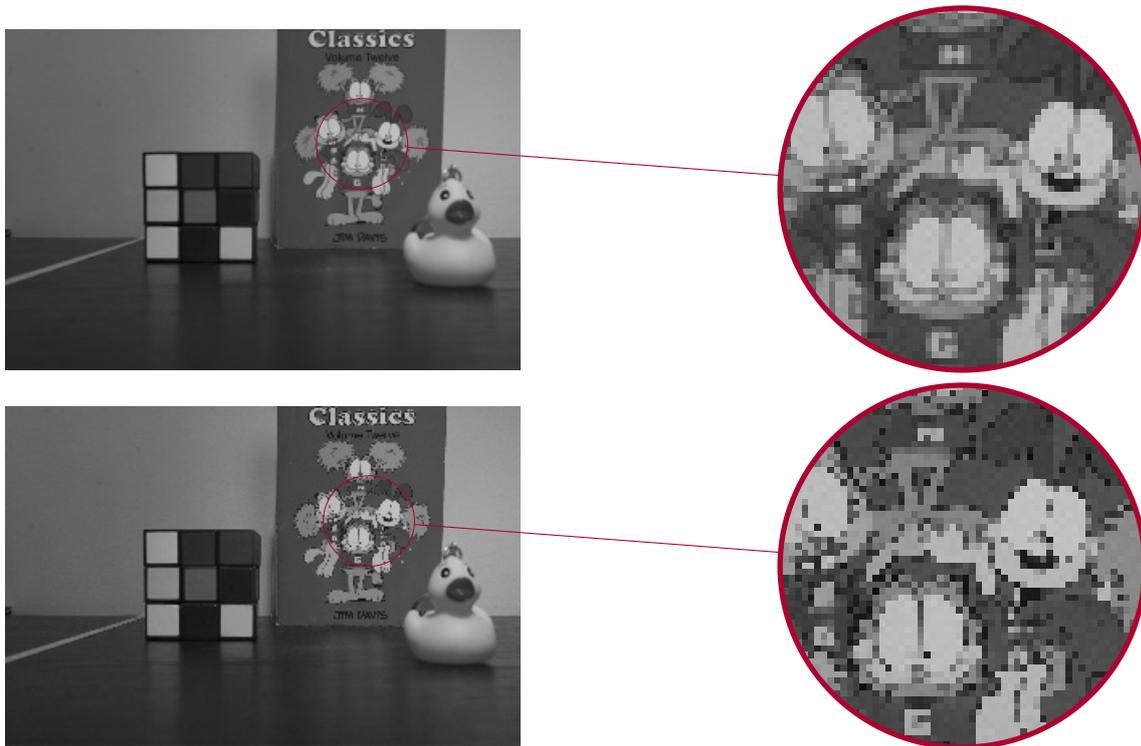


Abbildung 3.6: Skalierung auf 25%: inter-area Interpolation (oben), nächster Nachbar (unten).

des Originalpixels (BRADSKI & KAEHLER 2008, S. 130). Der Gauß-Filter funktioniert genauso. Die Methoden zur Skalierung sind Filter, die das Bild geometrisch manipulieren (BRADSKI & KAEHLER 2008, S. 163). Abbildung 3.6 oben illustriert die inter-area Interpolation. Ziel ist es, das Originalbild möglichst glatt auf das skalierte Zielbild abzubilden (BRADSKI & KAEHLER 2008, S. 130). Zur Verkleinerung eines Bildes eignet sich die inter-area Interpolation am besten (OPENCV 2015a).

Zur Verbesserung der Laufzeit und zur vollständigen Darstellung der Sensoraufnahmen am Bildschirm werden die Aufnahmen auf 60% verkleinert. Dies führt zu einer Laufzeitverkürzung auf  $(\frac{60\%}{100\%})^2 = 36\%$ , ohne die Objekterkennung negativ zu beeinflussen.

### 3.3 Canny Kantendetektor

Dieser Algorithmus zur Detektion von Kanten wurde 1986 von John Canny vorgestellt (CANNY 1986, 678ff). Er ist eine robuste Methode zum Detektieren von 1 px (Pixel) dicken Kanten in Bildern (MOESLUND 2012, S. 88) und kombiniert mehrere einfache Bildverarbeitungsmethoden zu einem komplexen Algorithmus.

Eine *Kante* in einem Bild beschreibt einen Bereich, über dem sich die Intensität stark ändert (BRAHMBHATT 2013, S. 51; LAMMERS et al. 2003, S. 9). Der *Bildgradient* – kurz Gradient – gibt die Änderung der Intensität an (BRAHMBHATT 2013, S. 51). In einem Graustufenbild ist ein hoher Bildgradient dort, wo helle (weiße) Bereiche an dunkle (schwarze) grenzen.

Kanten sind interessante Informationen in Bildern. Sie begrenzen Objekte (beliebiger Form) oder bilden charakteristische Muster. Der Canny Kantendetektor findet Kanten und extrahiert so Informationen aus einem Bild. Er läuft in folgenden Schritten ab (MOESLUND 2012, S. 88):

1. Gauß-Glättung
2. Sobel-Filter
3. nicht-maximale Unterdrückung (Verdünnung)
4. Schwellenwertoperation

Das Bild wird zunächst mit dem Gauß-Filter geglättet. Das macht den Algorithmus unempfindlich gegenüber Bildrauschen (CANNY 1986, S. 691), aber auch sehr dünnen (Pixelbreite) Kanten. Danach werden die Bildpixel aus Bild  $B$  um die beiden Sobel-Kernel  $S_x, S_y$  mittig gefalzen:

$$G_x = S_x * B = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * B; \quad G_y = S_y * B = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * B; \quad (3.7)$$

Die Ergebnisse dieser Operation sind in Abbildung 3.7 für ein Sensorbild und Abbildung A.1 im Anhang für das Bild einer DSLR-Kamera dargestellt. Es wird deutlich, dass mit dem Sobel-Operator  $S_x$  ( $S_y$ ) besonders gut horizontale (vertikale) Kanten entdeckt werden.

Davon ausgehend kann die Richtung des Bildgradienten mit der Formel

$$\Theta = \text{atan2}(G_x, G_y) \quad (3.8)$$

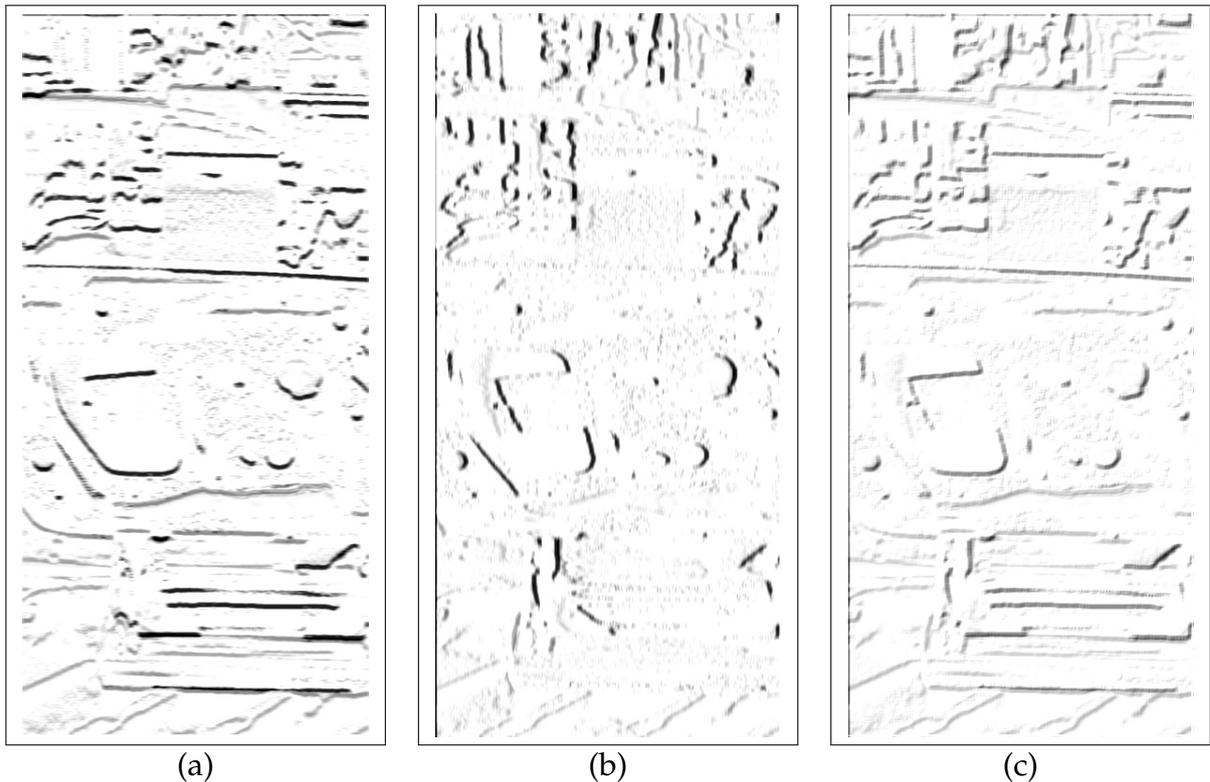


Abbildung 3.7: Sobel-Filter: (a) horizontaler Kernel; (b) vertikaler Kernel; (c) Überlagerung des horizontalen und des vertikalen Kernels.  
Zur besseren Darstellung sind die Farben invertiert.

berechnet werden. Für jeden Pixel wird der Betrag des Bildgradienten mit seinen zwei nächsten Nachbarn in Richtung des Gradienten verglichen und nur der Pixel mit dem größten Bildgradienten wird weiter untersucht (MOESLUND 2012, S. 88). Dieser Vorgang wird *nicht-maximale Unterdrückung*, der ganze Schritt *Verdünnung* genannt. Durch diese Operation ist sichergestellt, dass die Kanten genau 1 px breit sind (MOESLUND 2012, S. 88).

Der letzte Schritt ist eine Schwellenwertoperation (englisch „Thresholding“) in Form einer Hysterese. Pixel mit einem Bildgradientenwert  $g(x, y)$  größer als der obere Schwellwert  $T_{oben}$  gehören – per Definition – zu einer Kante. Pixel mit einem Gradientenbetrag kleiner als der untere Schwellwert  $T_{unten}$  werden gelöscht. Diejenigen Pixel, deren Werte zwischen diesen beiden Schwellwerten liegen werden nur dann berücksichtigt, wenn sie mit einem Pixel verbunden sind, dessen Bildgradientenwert größer ist als der obere Schwellwert.<sup>11</sup> Die Schwellwerte müssen *a priori* manuell vorgegeben werden. Die Fallunterscheidungen sind in Gleichung 3.9 zusammengefasst (BRADSKI & KAEHLER 2008,

<sup>11</sup>auch über mehrere „schwache“ Pixel hinweg.

S. 152).

$$g(x, y) = \begin{cases} > T_{oben} & \rightarrow \text{Kantenpixel} \\ < T_{unten} & \rightarrow \text{kein Kantenpixel} \\ \in [T_{unten}, T_{oben}] & \begin{cases} \cup \text{Kantenpixel} & \rightarrow \text{Kantenpixel} \\ \cap \text{Kantenpixel} & \rightarrow \text{kein Kantenpixel} \end{cases} \end{cases} \quad (3.9)$$

Der Algorithmus erzeugt ein Binärbild, in dem nur die Kanten verzeichnet sind. In Abbildung 3.8 ist das Resultat der Kantendetektion anhand zweier Beispielbilder und unterschiedlicher Schwellwerte dargestellt.

Der Canny Kantendetektor ist ein robuster Algorithmus zur Detektion von Kanten. Er funktioniert sowohl bei Farbbildern als auch bei Graustufenbildern. Er wird bspw. bei der Kamerakalibrierung zur Bestimmung von *Ecken* – Schnittpunkte von Kanten – verwendet (HARTLEY & ZISSERMAN 2003, S. 182). Die geometrische Kamerakalibrierung wird in Abschnitt 3.5 behandelt. In dieser Arbeit wird der Canny-Algorithmus verwendet, um Konturen von Objektmerkmalen zu erkennen.

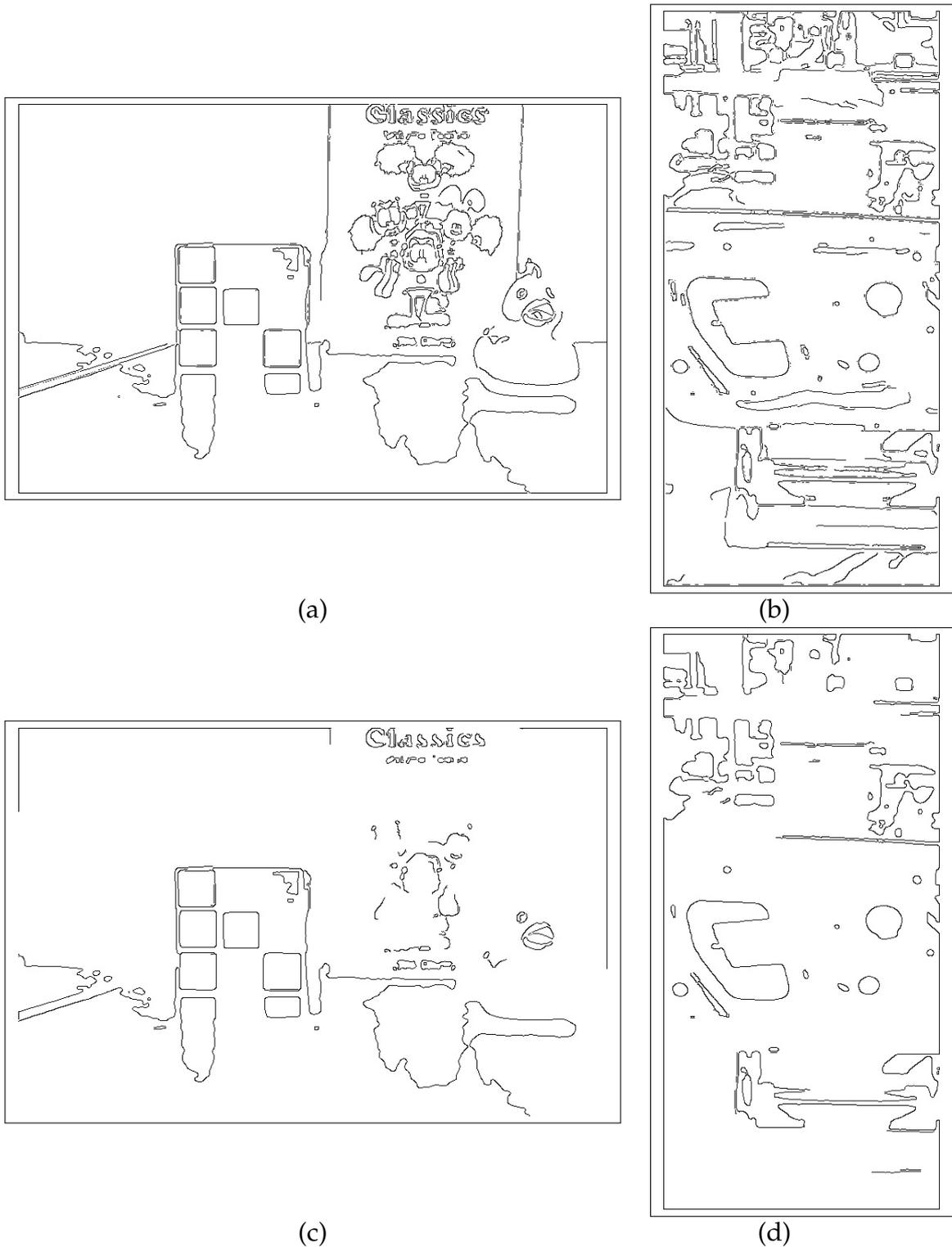


Abbildung 3.8: Binärbilder des Canny Kantendetektors. Bei den Bildern (a) und (b) sind die Schwellwerte 50 und 150; bei (c) und (d) 200:400. Die Farben der Binärbilder sind invertiert.

### 3.4 Houghtransformation

Die Houghtransformation ist eine robuste Methode zur Erkennung von mathematisch beschreibbaren Formen in Binärbildern (BATCHELOR 2012, S. 613). Ein solches Binärbild kann bspw. durch den Canny Kantendetektor erzeugt werden. Paul Hough stellte das Verfahren bereits 1962 vor (OTTERBACH 1995, S. 67) – ursprünglich nur zur Detektion von Geraden (BRADSKI & KAEHLER 2008, S. 152). Das Prinzip der Houghtransformation ist für alle Formen gleich (SUK & BHANDARKAR 1992a, 111f) und wird der Einfachheit halber anhand der Houghtransformation für Geraden erklärt.

Geraden lassen sich z. B. durch ihre Steigung und ihren y-Achsenabschnitt vollständig beschreiben. Eine andere Darstellung ist die Hessische Normalform, die durch Winkel  $\rho$  und Abstand zum Ursprung  $\eta$  charakterisiert ist. Für die Houghtransformation werden aus allen Bildpunkten<sup>12</sup> Geraden berechnet und im sogenannten Parameter- oder Hough-Raum gezählt (OTTERBACH 1995, 67f). Im Fall von Geraden ist der Parameterraum der zweidimensionale  $\rho$ - $\eta$ -Raum. Jeder Punkt im Raum beschreibt eine bestimmte Gerade. Jede Gerade „votiert“ (+1) im Parameterraum für eine Parameterkombination (OTTERBACH 1995, 67f). Aus Punkten, die auf derselben Geraden liegen, werden die gleichen Parameter gebildet. Es muss folglich nur noch nach Häufungen im Parameterraum (auch *Akkumulatorraum*) gesucht werden (OTTERBACH 1995, 67f). Durch dieses Votieren für eine bestimmte Parameterkombination ist der Algorithmus sehr robust gegenüber Verdeckungen (*Okklusion*) und Bildrauschen (TREIBER 2010, 44, 183f).

Das Prinzip kann auf alle mathematisch beschreibbare Formen ausgeweitet werden – mit einem entsprechend höherdimensionalen Parameterraum: Kreise (Mittelpunkt und Radius  $(x, y, r)$ ) Ellipsen (Brennpunkt und Halbachsen  $(x, y, a, b)$ ) (BRADSKI & KAEHLER 2008, 161f). Mit der *verallgemeinerten Houghtransformation* können beliebige Formen (ohne mathematische Beschreibung) detektiert werden (LAMMERS et al. 2003, S. 12). Die verallgemeinerte Houghtransformation funktioniert nach dem gleichen Prinzip wie die Standard Houghtransformation (SUK & BHANDARKAR 1992a, 111f).

Die Houghtransformation ist eine robuste (TREIBER 2010, 183f) und schnelle (BRADSKI & KAEHLER 2008, S. 152) Methode zur Detektion von Formen. Sie wird in dieser Arbeit verwendet, um *gerade Kanten*<sup>13</sup> in Bildern zu detektieren. Bei industriell verarbeiteten

<sup>12</sup>In einem Binärbild werden alle weißen Punkte (Wert 1) als Bildpunkte betrachtet.

<sup>13</sup>(allgemeine) Kanten können eine beliebige Form haben. *Gerade Kanten* sind Geraden aus detektieren Kanten z. B. mit Hilfe des Canny Algorithmus' (Abschnitt 3.3).

Blechen und Blechhalbzeugen sind gerade Kanten ein besonders deutliches und häufiges Merkmal.

### 3.5 Geometrische Kamerakalibrierung

Um von einem 2D Bild ausgehend eine Aussage über die 3D Welt treffen zu können, wird eine Abbildungsfunktion von den Bildkoordinaten  $(u, v)$  zu den realen Koordinaten  $(X, Y, Z)$  benötigt. Die geometrische Kamerakalibrierung liefert die Funktion  $(u, v) \rightarrow (X, Y)$ . Dazu müssen die Brennweiten  $f_x, f_y$ , sowie die Lage des Bildmittelpunkts  $(c_x, c_y)$  bestimmt werden (BRAHMBHATT 2013, S. 175). Die Werte werden in der *intrinsischen Kameramatrix* zusammengefasst:

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

In der Praxis bildet keine Linse perfekt ab (BRADSKI & KAEHLER 2008, S. 396). Bei der Fertigung der Linse gibt es Formtoleranzen und beim Zusammenbau der Kamera kann der Sensor nicht immer perfekt montiert werden (BRADSKI & KAEHLER 2008, S. 396). Dadurch wird die Projektion der 3D Szene verzerrt. Diese Verzerrung kann in 5 Koeffizienten gemessen werden und nachträglich aus der Aufnahme herausgerechnet werden. Die Verzerrungskoeffizienten werden einmalig beim Kalibrieren der Kamera bestimmt und sind skalierungsinvariant (OPENCV 2015a). Sie gehören ebenfalls zu den *intrinsischen Parametern* der Kamera (BRADSKI & KAEHLER 2008, S. 371). Die (radiale) Verzerrung wird in Abbildung 3.9 dargestellt.

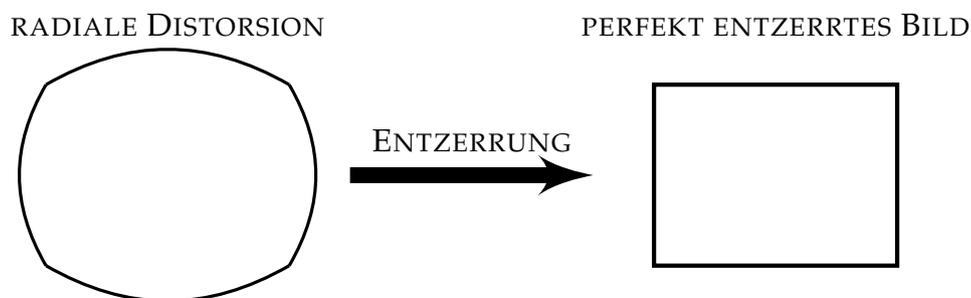


Abbildung 3.9: (Radial) verzerrtes und entzerrtes Bild.  
Nach (HARTLEY & ZISSERMAN 2003, S. 190)

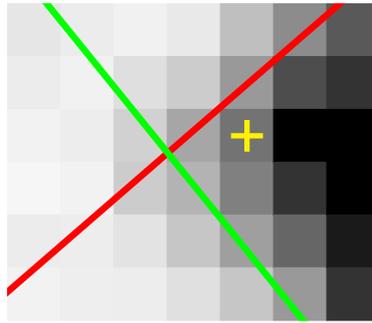


Abbildung 3.10: Detektion einer Ecke im Sub-Pixel-Bereich. Das gelb markierte Pixel ist die Ecke im Pixel-Bereich. Der Schnittpunkt der roten und grünen Linie, welche jeweils das Maximum eines Bildgradienten darstellen, gibt die Position der Ecke im Sub-Pixel-Bereich an.

Nach (STOCK et al. 2002, S. 194).

Um die intrinsischen Parameter zu bestimmen, wird eine Aufnahme genau bekannter Punkte benötigt. Das kann prinzipiell jedes vermessene Objekt sein (BRADSKI & KAEHLER 2008, S. 381). Es existieren auch Ansätze zur Selbstkalibrierung der Kamera durch mehrerer Aufnahmen einer beliebigen Szene (FRAGA 2009). In der Praxis ist ein Schachbrettmuster der einfachste Weg, eine Kamera zu kalibrieren (BRADSKI & KAEHLER 2008, S. 393). Zum Bestimmen der intrinsischen Matrix (4 Parameter) und der 5 Verzerrungskoeffizienten werden mindestens 10 Aufnahmen eines  $7 \times 8$  Schachbrettmusters aus unterschiedlichen Perspektiven benötigt (BRADSKI & KAEHLER 2008, 388f).<sup>14</sup>

Die Kantenlänge des Schachbretts gibt die Einheit der Parameter vor: Bei einer (realen) Kantenlänge von 250 mm entspricht die Länge einer Kante im Bild  $1 \text{ mm}/250$ .

Der Maßstab zwischen Pixel-Koordinaten und realen Koordinaten wird über die Länge der Kanten in Pixel bestimmt. Dazu wird die Position der Ecken im Bild im Subpixelbereich genau bestimmt (BRADSKI & KAEHLER 2008, S. 383). Abbildung 3.10 erklärt das Vorgehen. Es können Genauigkeiten höher als  $1/10$  Pixel erreicht werden (HARTLEY & ZISSERMAN 2003, S. 182).

Spricht man von einer „kalibrierten Kamera“, so ist nicht die Kamera selbst perfekt, sondern ihre Aufnahmen können mit den intrinsischen Parametern nachträglich perfekt gemacht werden. Mit der intrinsischen Matrix können Punkte eindeutig zwischen Bild- und Weltkoordinaten abgebildet werden (BRADSKI & KAEHLER 2008, S. 405; HARTLEY & ZISSERMAN 2003, S. 16).

<sup>14</sup>Für eine detaillierte Begründung sei der Leser an (BRADSKI & KAEHLER 2008, S. 388) verwiesen. Die Verwendung eines nicht-quadratischen Musters ist intuitiv ersichtlich, da sich das Muster so erst nach Drehung der Kamera um  $180^\circ$  wiederholt und nicht schon nach  $90^\circ$ .

## 4 Objekterkennung

### 4.1 Generischer Aufbau einer Objekterkennung

Algorithmen zur Objekterkennung sind allgemein aus drei Schritten aufgebaut (LOWE 2004; BAY et al. 2007; ABBASI et al. 1999; LEVI 2013a):

1. finden – Detektion/Extraktion
2. beschreiben – Deskription
3. vergleichen – Zuordnung

Die meiste Aufmerksamkeit erfährt die Detektion von Merkmalen (englisch „Features“). Das können markante Punkte, bestimmte Formen wie Linien und Kreise oder beliebige Konturen sein. In der Regel werden Punkte oder Konturen verwendet (LOWE 2004, S. 472). Dabei wird die große Informationsmenge des Bildes auf einige wenige repräsentative Pixel herunter gebrochen (Schritt 1).

Im zweiten Schritt werden die gefunden Bildmerkmale möglichst eindeutig beschrieben, um sie danach mit Merkmalen aus anderen Bildern vergleichen zu können. Dazu wird die Informationsmenge wieder erhöht: Bei markanten Punkten wird ihre Umgebung betrachtet und bei Konturen deren geometrische Parameter (vgl. Abschnitt 4.3.1 und 4.2.2). Der sogenannte *Deskriptor* kodiert die Eigenschaften eines Merkmals in einem hochdimensionalen Vektor (BAY et al. 2007, S. 346).

Im dritten Schritt werden die Eigenschafts-Vektoren (Deskriptoren) miteinander verglichen. Entscheidend hierbei ist das Vergleichsmaß, z. B. der Hamming-Abstand oder die Summe der quadrierten Differenzen, und die Schwelle ab dem ein Objekt als „erkannt“ gilt. Bei Deskriptoren aus der Literatur wird i. d. R. eine Vergleichsmethode empfohlen. Die Erkennungsschwelle ist jedoch stark abhängig von Bildqualität und der Fähigkeit des Detektors, die genau gleichen Merkmale in zwei Bildern zu finden (TREIBER 2010, S. 155). Sie wird deshalb empirisch bestimmt.

Algorithmen zur Objekterkennung sind – unabhängig von den Merkmalen auf denen sie basieren (z. B. Punkte oder Konturen) – modular aufgebaut. Jeder der drei Schritte Detektion, Deskription und Zuordnung lässt sich nach unterschiedlichen Gesichtspunkten optimieren. Die Auswahl hängt von der zu erfüllenden Aufgabe und Situation ab.

Im Weiteren wird speziell auf die Objekterkennung mit Konturen (Abschnitt 4.2) und markanten Punkten (Abschnitt 4.3) eingegangen. Konturen werden zur Erkennung der Tür und zur groben Bestimmung ihrer Lage verwendet. Mit markanten Bildpunkten wird eine Tiefenkarte berechnet, um die Entfernungsmessung zu überprüfen und gegebenenfalls zu verbessern. Um die Theorie in Grenzen zu halten, werden nur Algorithmen und Methoden erklärt, die auch in der Arbeit Anwendung finden. Der interessierte Leser sei auf die genannten Quellen und das Literaturverzeichnis verwiesen.

### 4.2 Objekterkennung mit Konturen

*Konturen* sind Linien, die ein Objekt begrenzen (BRAHMBHATT 2013, S. 67). Sie werden in einem Nachbearbeitungsschritt der Kantendetektion bestimmt. Das Binärbild mit den Kanten (siehe Abbildung 4.1 (b) - (c)) wird in sinnvolle Bestandteile zerlegt – es wird *segmentiert* (BÖHM 2004, S. 38). Konturen in einem Bild sind zusammenhängende Punkte einer Kante. Kanten wiederum sind Bereiche starker Änderungen der Grauwerte des Bildes (LAMMERS et al. 2003, 9f). Konturen in Bildern sind also nicht zwangsläufig Grenzlinien von Objekten, sondern nur Kandidaten interessanter Bildbereiche (BRADSKI & KAEHLER 2008, S. 244).

#### 4.2.1 Konturapproximation

Bei der Analyse von Formen ist es üblich, die Kontur durch Polygone anzunähern (BRADSKI & KAEHLER 2008, S. 245). Diese Approximation ist nichts anderes als eine Kurvenglättung und macht die Kontur robuster gegenüber falsch erkannten Konturpunkten. Sie liefert auch eine kompaktere Darstellung der Kontur (LAGANIERE 2014, S. 244). In dieser Arbeit wird dafür der weit verbreitete Douglas-Peucker-Algorithmus verwendet (DOUGLAS & PEUCKER 1973). Er läuft in folgenden drei Schritten ab (BRADSKI & KAEHLER 2008, 245f):

1. Bilde eine Linie zwischen den zwei äußersten Punkten.
2. Bestimme den Punkt mit dem größten Abstand  $d$  zur Kurve.

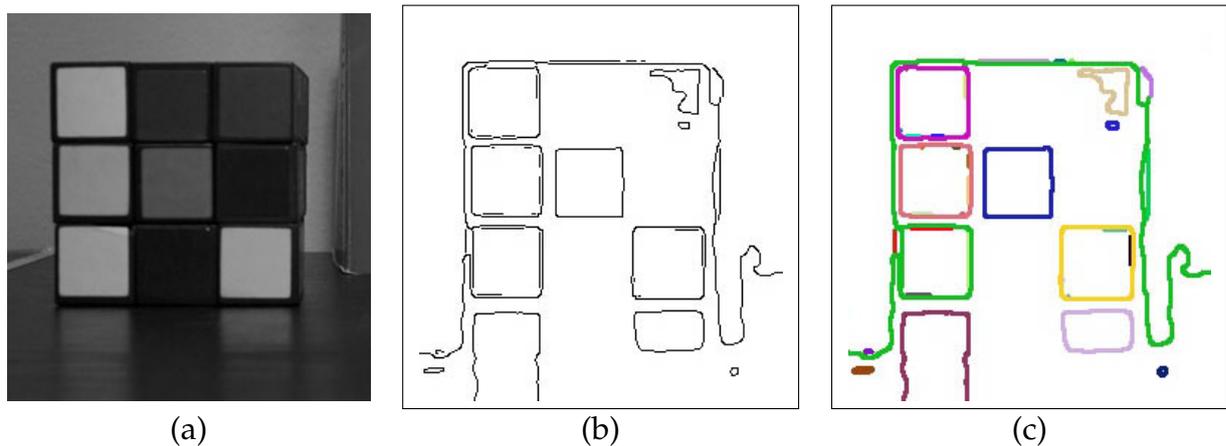


Abbildung 4.1: Detektion von Konturen. (a) Originalbild; (b) Resultat des Canny Kantendetektor (Schwellwerte 180:360); (c) segmentierte Konturen.

3. Falls  $d > d_{max}$  : füge den Punkt zur Kurve hinzu und wiederhole Schritte 2-3.

Abbildung 4.2 stellt den Ablauf grafisch dar. Über den maximalen Abstand  $d_{max}$ , den das genäherte Polygon zur ursprünglichen Kontur haben darf, kann die Genauigkeit der Approximation eingestellt werden. Ist die der Wert zu hoch, gehen wichtige Informationen über die Form der Kontur verloren; ist der Wert zu klein, so wird kein Gewinn an Robustheit erreicht. Gute Werte für den maximalen Abstand sind Anteile der Konturgröße, wie bspw. die Länge (BRADSKI & KAEHLER 2008, S. 246). Bei dem vorliegenden System wird die Anzahl der Pixel  $\#px$  und der Umfang bzw. die Länge  $l$  berücksichtigt:

$$d_{max} = \frac{l \cdot \#px}{2} / 100 \quad (4.1)$$

#### 4.2.2 Charakteristiken von Konturen

Konturen können mit einem einfachen geometrischen Filter verglichen werden: Das Bild wird nach der exakt gleichen (Größe, Orientierung, etc.) Kontur durchsucht. Diese Methode ist nicht zielführend, denn sie ist anfällig für Verdrehung, Verschiebung und Größenänderungen (TREIBER 2010, 74f). Um Konturen vergleichen zu können, werden ihre unterschiedlichen Eigenschaften möglichst eindeutig beschrieben. Diese Parameter werden in einem Deskriptor zusammengefasst. Es kann zwischen globalen/lokalen Parametern und verschiedenen Detaillierungsgraden unterschieden werden.

Globale Parameter beschreiben die Kontur im Ganzen (ABBASI et al. 1999, S. 471). Dazu gehören beispielsweise:

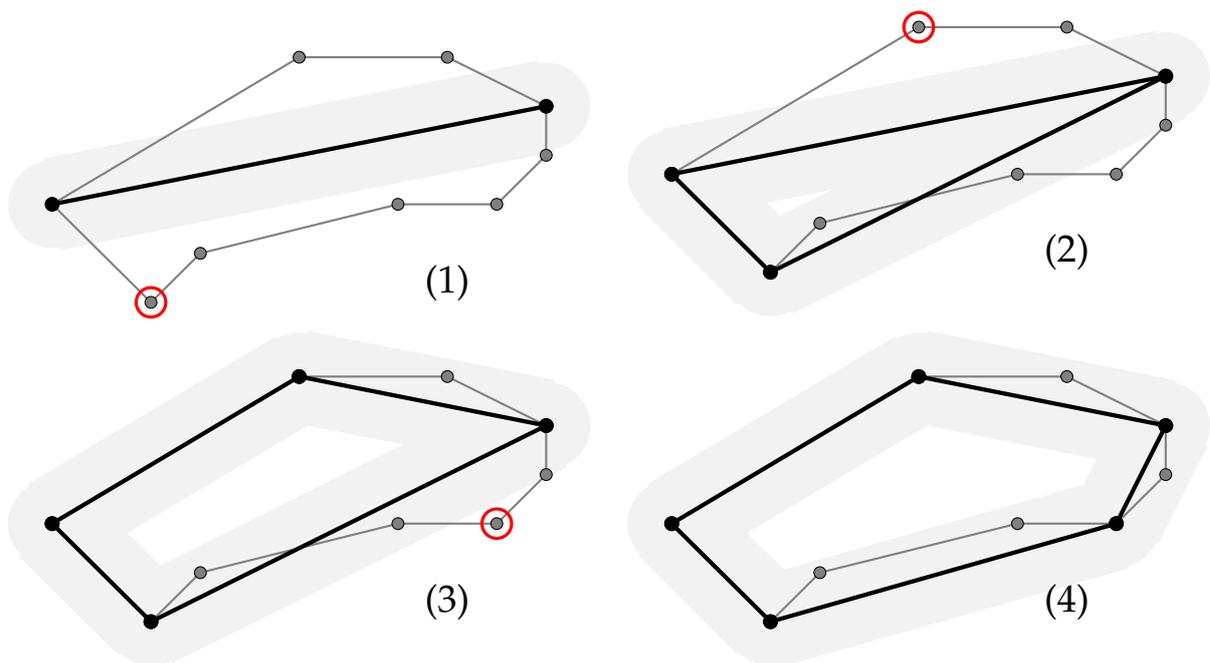


Abbildung 4.2: Polygonapproximation einer Kontur nach dem Douglas-Peucker Algorithmus. Ausgehend von einer geschlossenen Linie zwischen den am weitesten auseinanderliegenden Punkten wird sukzessive die neue Konturlinie um den Punkt erweitert, der am weitesten von ihr entfernt liegt. Die Punkte innerhalb des Toleranzbereichs (hellgrau) sind genähert.

- Länge/Umfang
- Fläche
- umrahmende Formen: Rechteck, Kreis, Ellipse
- Exzentrizität, Rundheit
- Bildmomente

Abbildung 4.3 zeigt das flächenmäßig kleinste Rechteck und den kleinsten Kreis, die die Kontur umschließen, sowie die in auf die Kontur eingepasste (englisch „fitted“) Ellipse. Parameter wie Exzentrizität  $\varepsilon$  und Rundheit  $\kappa$  sind ebenfalls globale Parameter (ABBASI et al. 1999, S. 473). Sie beschreiben wie sich die Form zu einem Kreis verhält. Die Exzentrizität liegt zwischen 0 (Kreis) und 1 (Linie) und ist invariant gegenüber Skalierung und Rotation (KILIAN 2001, S. 7). Wegen des festen Wertintervalls lässt

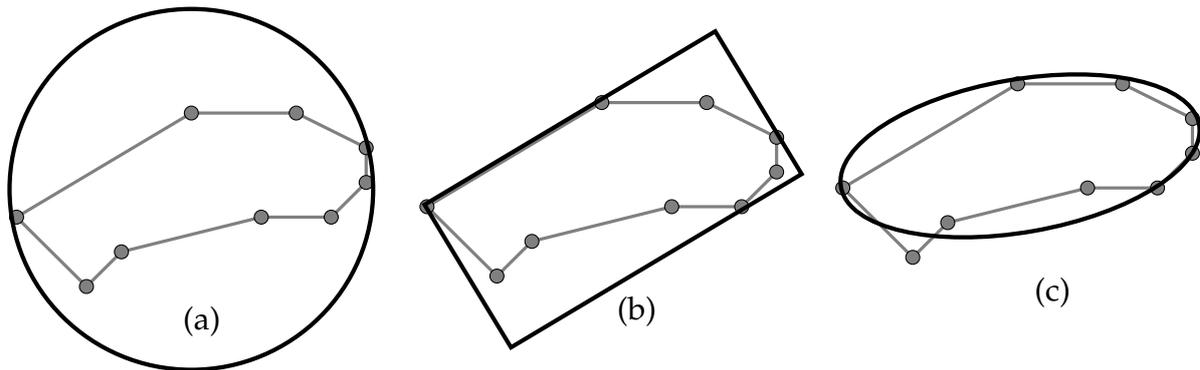


Abbildung 4.3: Flächenmäßig kleinster Kreis (a) und kleinstes Rechteck (b), die die Kontur umschließen. Die Ellipse (c) wird mit der „least-squared“ Methodik eingepasst.

sich die Exzentrizität  $\varepsilon$  besser vergleichen und ist der Rundheit  $\kappa$  deshalb vorzuziehen (KILIAN 2001, S. 8).

$$\varepsilon = \sqrt{1 - \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}} \quad (4.2)$$

$$\text{mit } \lambda_1 = \mu_{20} + \mu_{02}; \quad \lambda_2 = \sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2} \quad (4.3)$$

$$\kappa = \frac{4\pi A}{r^2} \quad (4.4)$$

$A$  Fläche der Kontur;  $r$  Radius des kleinsten, die Form einschließenden, Kreises.  $\lambda$  ist eine Hilfsgröße.  $\mu_{i,j}$  sind die zentralisierten Bildmomente. Bildmomente sind mächtige Parameter zur kompakten Beschreibung einer Kontur (LAGANIERE 2014, 223ff). Sie werden im nächsten Kapitel detailliert betrachtet.

Lokale Parameter beschreiben Details der Kontur (LOWE 2004, S. 472). So zählt der größte/kleinste Abstand zweier Konturpunkte zu den lokalen Parametern. Einen interessanten Ansatz verfolgen OLIVEIRA JR. et al. Sie zählen die konvexen und konkaven Halbkreise zum Erkennen von Schreibrift (OLIVEIRA JR. et al. 2009, S. 374). ABBASI et al. zählen die Anzahl der Wechsel einer Kontur von konvex nach konkav und andersherum (TREIBER 2010, 135ff).<sup>15</sup> Der Algorithmus ist auf markante Formänderungen angewiesen und daher nicht geeignet zur Beschreibung von Bohrungen wie sie in technischen Bauteilen häufig vorkommen.

<sup>15</sup> ABBASI et al. benutzten dies 1999 zum Erkennen unterschiedlicher Fische. Ihre markanten Flossen lassen sich auch nach mehreren Glättungsschritten noch durch die Änderung der Konturkrümmung unterscheiden. Der Algorithmus ist vergleichsweise langsam.

Viele dieser Parameter sind zwar rotationsinvariant, aber stark abhängig bspw. von der Bildskalierung und können deshalb nicht ohne weiteres verwendet werden, um Konturen zwischen Bildern zu vergleichen. In Abschnitt 6.3.1 wird erläutert welche Parameter für das vorliegende System verwendet werden.

Eine etwas untergeordnete Rolle spielen *relationale* Charakteristiken in der Literatur. Unter diesem Begriff werden Parameter zusammengefasst, die die Entfernung und Orientierung von Merkmalen zueinander beschreiben (LOWE 2004, S. 472). Relationale Parameter sind sehr nützlich, wenn Objekte aus verschiedenen Merkmalen bestehen. Dies ist bspw. bei großen Blechteilen mit Bohrungen und Löchern der Fall. Hier lassen sich häufig die markanten Löcher zuverlässiger detektieren, als der Umriss des gesamten Bauteils (s. Abschnitt 6.3.2).

### 4.2.3 Bildmomente

*Bildmomente* oder kurz *Momente* sind Charakteristiken von Bildsegmenten, die aus dem Integral über alle Pixel abgeleitet werden (BRADSKI & KAEHLER 2008, S. 252). Sie beschreiben grob die Form einer Kontur (BRADSKI & KAEHLER 2008, S. 247). Bildmomente sind eine der einfachsten Arten, um Konturen zu vergleichen (BRADSKI & KAEHLER 2008, S. 252). Die allgemeine Definition eines Moments ( $m_{pq}$ ) wird bei digitalen Bildern zu einer Summe über alle diskreten Pixel ( $m_{ij}$ ):

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (4.5)$$

$$m_{ij} = \sum_x \sum_y x^i y^j g(x, y) \quad (4.6)$$

Mit den Grauwerten  $g(x, y)$  über alle  $(x, y)$ -Kombinationen innerhalb des Segments. Bei binären Bildern entfallen diese und die Bildmomente werden nur aus der (geometrischen) Form berechnet.<sup>16</sup> Bildmomente haben keine physikalische Bedeutung; mit ihrer Hilfe ist jedoch eine geometrische Rekonstruktion des Objekts möglich

<sup>16</sup>In Abschnitt 3.2.1 wurde darauf hingewiesen, dass durch die adaptive Histogrammanpassung keine eindeutige Beziehung zwischen den Intensitätswerten der Originalaufnahme und dem angepassten Bild besteht. Deshalb werden die Bildmomente in dieser Arbeit nur für Binärbilder bestimmt und beschreiben somit ausschließlich die Geometrie.

(KILIAN 2001, S. 5). So ist das Moment  $m_{00} = \sum \sum g(x, y)$  der Flächeninhalt des Segments bzw. der Kontur (Gleichung 4.6 mit  $i = 0$  und  $j = 0$ ). Der Flächenschwerpunkt kann aus den Momenten  $m_{00}$ ,  $m_{01}$  und  $m_{10}$  nach Gleichung 4.7 berechnet werden:

$$(x_{SP}, y_{SP}) = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (4.7)$$

Da diese Momente von ihrer Position im Bild abhängen, spricht man von *spatialen Momenten* (KILIAN 2001, S. 3). Durch Verschiebung des Koordinatenursprungs in den Flächenmittelpunkt und Normalisierung durch die Fläche werden die Momente translations- und skalierungsinvariant: „*central normalized moments*“  $v_{ij}$  (KILIAN 2001, S. 3).<sup>17</sup>

$$\mu_{ij} = \sum_x \sum_y (x - x_{SP})^i (y - y_{SP})^j g(x, y) \quad (4.8)$$

$$v_{ij} = \frac{\mu_{ij}}{\mu_{ij}^{1 + \frac{i+j}{2}}} \quad (4.9)$$

Die „*central normalized moments*“ können linear kombiniert werden, sodass sie invariant gegenüber Rotation und Spiegelung werden. Die so entstehenden Momente heißen *Hu Momente* (BRADSKI & KAEHLER 2008, S. 254). Die ersten sieben Momente stammen aus dem Originalartikel von HU, das achte Moment  $Hu_8$  wurde von FLUSSER ergänzt. Die vollständigen Hu Momente sind in Tabelle 4.1 aufgelistet. Sie werden in dieser Arbeit zur Beschreibung von Konturen verwendet (vgl. Abschnitt 6.3.1).

Je komplexer ein Objekt, desto höher sollte die Ordnung  $i + j$  der Momente sein (KILIAN 2001, S. 5). Gleichzeitig sinkt der Betrag der Momente mit steigender Ordnung (BRADSKI & KAEHLER 2008, S. 254). Dies muss beim Vergleich von Hu Momenten berücksichtigt werden, z. B. durch eine logarithmische Skalierung zur Basis 10 (OPENCV 2015d). So ergibt sich die Differenz  $D(A, B)$  der Momente zweier Konturen  $A$  und  $B$  zu:

$$D(A, B) = \sum_{i=1}^7 \left| \frac{1}{\text{sign}(Hu_i^A) \cdot \lg(Hu_i^A)} - \frac{1}{\text{sign}(Hu_i^B) \cdot \lg(Hu_i^B)} \right| \quad (4.10)$$

<sup>17</sup>Die Momente zweiten Grades ( $i + j = 2$ ) können wie Hauptträgheitsachsen bzw. dem Trägheitstensor aus der Mechanik interpretiert werden (KILIAN 2001, S. 6).  $J = \begin{bmatrix} \mu_{20} & -\mu_{11} \\ -\mu_{11} & \mu_{02} \end{bmatrix}$ . Die Einträge lassen sich nach dem Prinzip  $f(x, y) \cdot (\text{Abstand zum Schwerpunkt})^2$  bilden (KILIAN 2001, S. 6).

Tabelle 4.1: Hu Momente nach (HU 1962; FLUSSER 1999)

$$\begin{aligned}
 Hu_1 &= \eta_{20} + \eta_{02} \\
 Hu_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
 Hu_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
 Hu_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
 Hu_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[ (\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + \\
 &\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[ 3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \\
 Hu_6 &= (\eta_{20} - \eta_{02}) \left[ (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
 Hu_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[ (\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] - \\
 &\quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \left[ 3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \\
 Hu_8 &= \eta_{11} \left[ (\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2 \right] - (\eta_{20} - \eta_{02})(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21})
 \end{aligned}$$

Der Aufwand zur Berechnung der Momente wird durch das Theorem nach GREEN deutlich reduziert. Es beschreibt den Zusammenhang zwischen dem Doppelintegral über eine Fläche und dem Linienintegral über ihre Kontur (RIEMANN 1867, 8f). In digitalen Binärbildern muss nur noch die Summe über die Punkte  $(x, y)$  der Kontur  $C$  berechnet werden (BRADSKI & KAEHLER 2008, S. 252):

$$m_{ij} = \sum_x \sum_y x^i y^j g(x, y) = \sum_{(x,y) \in C} x^i y^j g(x, y) \quad (4.11)$$

Momente bieten eine einfache Möglichkeit, Konturen in Bildern zu vergleichen (BRADSKI & KAEHLER 2008, S. 252). In dieser Arbeit werden unter anderem Bildmomente verwendet. Sie sind leider nicht einzigartig genug, um allein durch sie Konturen einander zuzuordnen (englisch „matching“). Dies wird in Abschnitt 6.3.1 näher erläutert.

### 4.3 Objekterkennung mit markanten Punkten

*Markante Punkte* (englisch „keypoints“) sind Pixel, die sich genügend deutlich von ihrer Umgebung<sup>18</sup> unterscheiden (LOWE 2004, S. 5; BAY et al. 2007, S. 350). Das Prinzip ähnelt dem des Canny Kantendetektors (vgl. Abschnitt 3.3), jedoch werden jetzt einzelne Punkte und keine Kanten gesucht. Die verschiedenen Algorithmen zur Extraktion von markanten Punkten unterscheiden sich in Form und Größe der

<sup>18</sup>Die *Umgebung* ist ein festgelegter Bereich um den den betrachteten Pixel herum. Dabei können auch unterschiedlich stark geglättete Bilder des Originalbildes als „Ebenen“ mit einbezogen werden (LOWE 2004, 6f).

Umgebung sowie in der Wahl von Filtern als Vorverarbeitungsschritte. Die Algorithmen bestehen i. d. R. aus Detektor und Deskriptor (vgl. Abschnitt 4.1 Schritt 1 und 2).

Der mit Abstand bekannteste und am weitesten verbreitete Algorithmus zur Detektion von markanten Punkten ist Scale-Invariant Feature Transform (SIFT) (BRAHMBHATT 2013, S. 121). SIFT wurde 2004 von LOWE im *International Journal of Computer Vision* vorgestellt. Der Algorithmus ist robust gegenüber Bildrauschen und Verzerrungen und zeichnet sich durch seine eindeutigen Deskriptoren aus, die es ihm ermöglichen, auch sehr ähnliche Objekte zu unterscheiden (LOWE 2004, S. 4, 13, 25). Eine Weiterentwicklung von SIFT ist Speeded-Up Robust Features (SURF) (BAY et al. 2007). Er ist ähnlich robust und zuverlässig wie SIFT, bei halber Berechnungszeit (BRAHMBHATT 2013, S. 134; BAY et al. 2007, S. 352). Beide Algorithmen sind durch US-Patente geschützt und müssen für eine kommerzielle Anwendung lizenziert werden (BRADSKI & KAEHLER 2008, S. 121, 134). Wegen ihrer weiten Verbreitung werden sie jedoch häufig als Vergleichsmaßstab für neue Algorithmen herangezogen (TREIBER 2010, S. 154).

In dieser Arbeit wird der Oriented FAST and Rotated BRIEF (ORB) Algorithmus verwendet. Er wurde 2011 von RUBLEE et al. auf der 13<sup>th</sup> *International Conferenc on Computer Vision* als effiziente Alternative zu SIFT und SURF vorgestellt. Er bedient sich Teilen zweier anderer Algorithmen: dem Detektor von FAST und dem Deskriptor von BRIEF. Im Folgenden wird die Funktionsweise von ORB anhand seiner Teile beschrieben.

### 4.3.1 FAST Merkmal-Extraktor

FAST ist ein Akronym für *Features from Accelerated Segment Test* und wurde von ROSTEN & DRUMMOND entwickelt. Beim FAST Algorithmus wird ein Kreis aus  $N$  Pixeln Umfang um den zentralen Pixel  $C$  untersucht. Abbildung 4.4 illustriert dies. Gibt es z. B.  $N = 12$  zusammenhängende Pixel, die mehr als einen bestimmten Schwellwert  $t$  von der Intensität  $I_C$  des zentralen Pixels  $C$  abweichen, so ist  $C$  ein markanter Punkt (ROSTEN & DRUMMOND 2005, S. 6). Dieser Test kann beschleunigt werden, indem man zuerst die Pixel 1 und 7, sowie 4 und 10 testet:  $I_{1,7,5,10} < I_C \pm t$  (ROSTEN & DRUMMOND 2005, S. 6). So müssen nicht immer alle 12 Pixel getestet werden und die Pixeloperation wird deutlich beschleunigt.<sup>19</sup>

---

<sup>19</sup>In einer Video-Testsequenz von ROSTEN & DRUMMOND mussten im Durchschnitt nur 3,8 Umgebungspixel betrachtet werden (bei  $N = 16$ ).

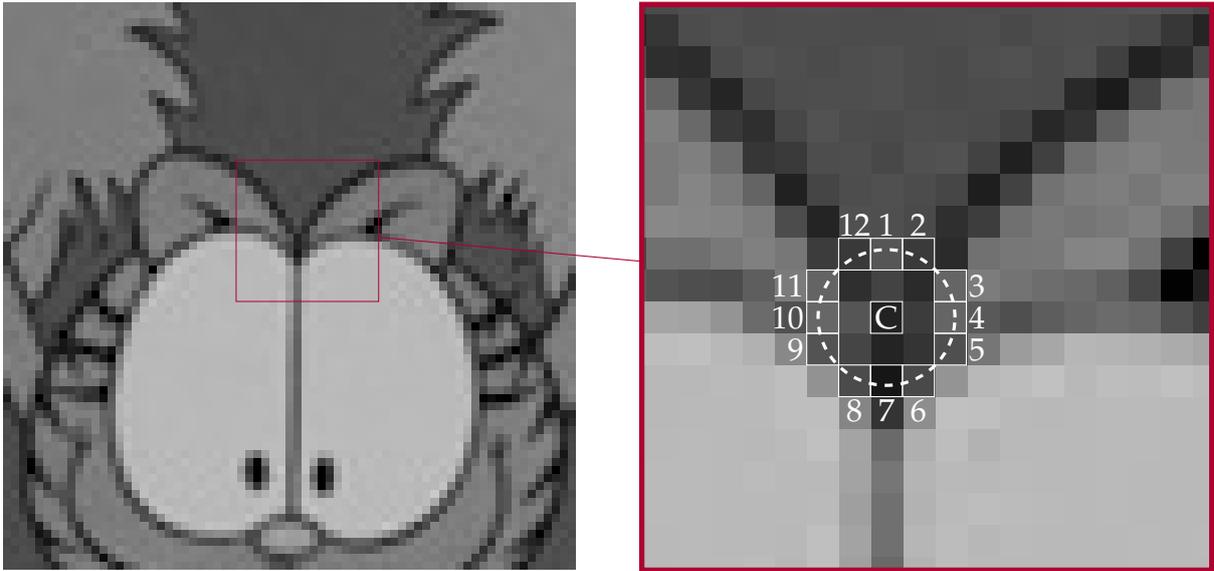


Abbildung 4.4: Funktionsprinzip FAST. Es werden die Nachbarn in einem  $N = 12$  px großen Kreis um den zentralen Pixel C herum betrachtet.  
 Nach (ROSTEN & DRUMMOND 2005, S. 6)

### 4.3.2 BRIEF Merkmal-Deskriptor

*Binary Robust Independent Elementary Features (BRIEF)* Deskriptor ist ein Vektor mit binären Intensitätstests (RUBLEE et al. 2011, S. 3).

Bei binären Deskriptoren werden durch ein bestimmtes Muster Pixelpaare  $(p_i, p_j)$  aus der Umgebung des markanten Punkts erzeugt und ihre Intensitäten verglichen (LEVI 2013b):

$$b = \begin{cases} 1, & I_j > I_i \\ 0, & \text{sonst} \end{cases} \quad (4.12)$$

Der so entstehende binäre Deskriptor-Vektor kann sehr schnell mit dem XOR Befehl verglichen werden (LEVI 2013b).<sup>20</sup> Der *Abstand* oder Unterschied zweier Deskriptoren ist die Summe der XOR Operationen: die Hamming-Distanz. Um einen genügend eindeutigen Deskriptor zu bekommen, müssen viele Pixelpaare betrachtet werden. Die genaue Anzahl wird im nächsten Abschnitt genauer betrachtet.

Der BRIEF Deskriptor benutzt ein zufällig generiertes Muster (LEVI 2013b). Das Muster wird einmal generiert und dann auf alle zu vergleichenden Bilder bzw. Merkmale angewendet. Abbildung 4.5 zeigt beispielhaft ein solches zufälliges Muster und das

<sup>20</sup>XOR ist eine elementare Rechneroperation, die auf Assemblerebene stattfindet. Daher die Schnelligkeit von Vergleichen zwischen binären Deskriptoren.

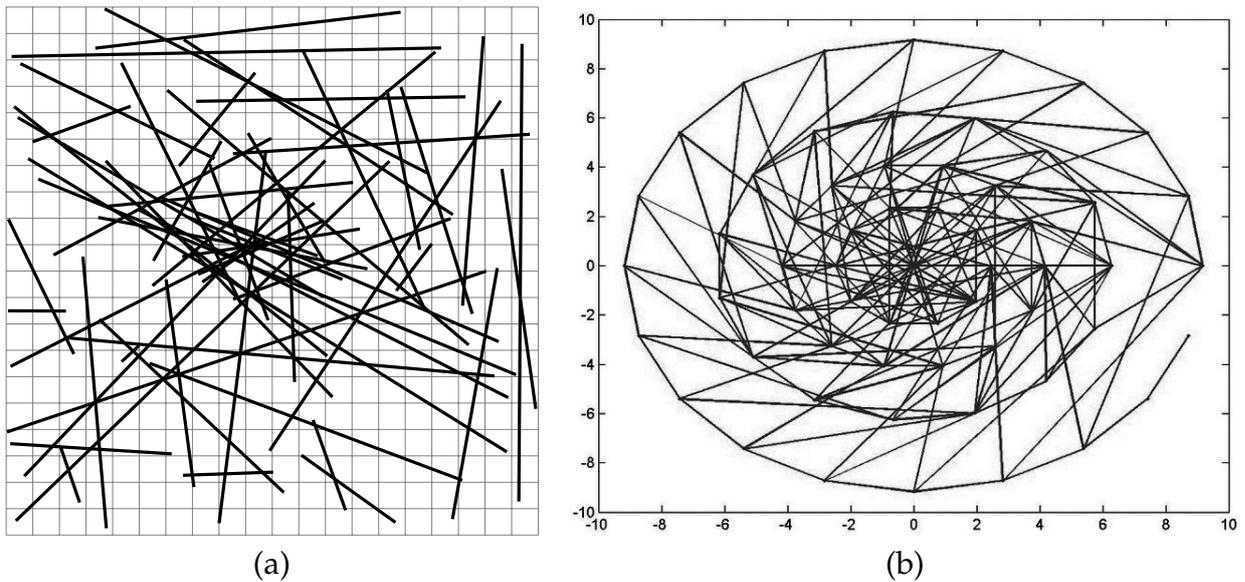


Abbildung 4.5: (a): Zufällig generiertes Muster, wie es für BRIEF benutzt wird. (b) Deskriptormuster von BIRSK  
Quelle (b): (LEVI 2013a)

eines anderen binären Deskriptors. Binäre Deskriptoren vergleichen einzelne Pixel miteinander. Das macht sie sehr anfällig für Bildrauschen. Das Bild bzw. die Bildregion sollte deshalb vorher geglättet werden (LEVI 2013b).

Der ORB Algorithmus erweitert den BRIEF Deskriptor um eine Kompensation der Rotation.

### 4.3.3 ORB Algorithmus: Oriented FAST and Rotated BRIEF

ORB benutzt den FAST Algorithmus (mit  $N = 12$ ) zu Detektion markanter Punkte. Darüber hinaus wird mit Hilfe von Bildmomenten die Orientierung dieser markanten Punkte bestimmt (RUBLEE et al. 2011, S. 2). Dazu wird nach Gleichung 4.7 der Flächenschwerpunkt  $C_{SP}$  der Umgebung bestimmt. Der Verbindungsvektor  $\overrightarrow{O_P C_{SP}}$  vom zentralen Pixel  $P$  zum Intensitäts-Schwerpunkt schließt mit der x-Achse den Winkel  $\theta$  ein:

$$\theta = \arccos \left( \frac{\overrightarrow{O_P C_{SP}} \cdot \vec{x}}{|\overrightarrow{O_P C_{SP}}| \cdot 1} \right) = \text{atan2}(m_{01}, m_{10}) \quad (4.13)$$

$m_{01}$  und  $m_{10}$  sind die spatialen Bildmomente,  $\vec{x} = [1 \ 0]^T$  der Einheitsvektor in Richtung der x-Achse.

Mit dem Orientierungswinkel  $\theta$  wird das Muster des BRIEF Deskriptors gedreht und so der Deskriptor invariant gegenüber Rotation gemacht.

Um den BRIEF Deskriptor weiter zu beschleunigen, werden von ihm nur die 256 aussagekräftigsten Pixelpaare benutzt (RUBLEE et al. 2011, S. 4).<sup>21</sup> Der Deskriptor ist dadurch genauso eindeutig wie der SIFT Deskriptor, aber um zwei Größenordnungen schneller (RUBLEE et al. 2011, S. 1).

Der ORB Algorithmus ist sehr schnell, ohne dabei – verglichen mit SIFT – an Genauigkeit einzubüßen. Er ist unempfindlich gegenüber Skalierung und Rotation und sehr robust bei Bildrauschen und Verzerrungen (BRAHMBHATT 2013, S. 136). In Tabelle 4.2 ist ein Laufzeitvergleich von ORB, SIFT und SURF von RUBLEE et al. zusammengefasst.

*Tabelle 4.2: Laufzeitvergleich von ORB, SIFT und SURF. Die Zeiten wurden als Durchschnitt aus 2686 Bildern ermittelt.*

*Quelle: RUBLEE et al. 2011, S. 7*

Detektor	ORB	SURF	SIFT
Zeit (ms)	15	217	5229

### 4.4 Zusammenfassung und Bezug zum Projekt

Für die modellbasierte Objekterkennung in dieser Arbeit werden Konturen verwendet. Industrielle Werkstücke besitzen i. d. R. markante Hinterschneidungen und Bohrungen, die gut anhand ihrer Kontur beschrieben werden können. Zudem werden Konturen sowohl im Modellbild, als auch in der Sensoraufnahme robust detektiert. Bei der Zuordnung korrespondierender Konturen werden ihre geometrischen und relationalen Charakteristiken betrachtet.

Der Roboter wird damit befähigt sein Werkstück zu erkennen.

Bei der Berechnung einer Tiefenkarte in der Stereovision (Kapitel 5) werden korrespondierende Pixel zwischen zwei Sensoraufnahmen benötigt. Dazu eignen sich Algorithmen, die markante Punkte detektieren. Hier wird der ORB Algorithmus verwendet, der sich durch Schnelligkeit und hohe Eindeutigkeit seines Deskriptors auszeichnet.

---

<sup>21</sup>Die „aussagekräftigsten“ Pixelpaare werden erlernt, indem aus einer Reihe von Testbildern die Deskriptoren der markanten Punkte ausgewertet werden. Dazu wird ein  $31 \times 31$  Pixel großer Bereich um den markanten Punkt in  $5 \times 5$  große Unterbereiche unterteilt. Alle möglichen, nicht überlappenden Kombinationen dieser Unterbereiche werden binär nach Gleichung 4.12 verglichen. Das sind 205590 Tests. Die 256 Bereichspaare, die über alle Testbilder hinweg den größten Hamming-Abstand haben, ergeben das „erlernte“ Deskriptormuster (genauer: ihre Mittelpunkte bilden das Muster). RUBLEE et al. nennen diesen Deskriptor *rBRIEF*.

## 5 Stereovision

In der Stereovision werden Merkmale von zwei oder mehr Aufnahmen verglichen und aus *korrespondierenden* (passenden) Merkmalen Tiefeninformationen gewonnen (BRADSKI & KAEHLER 2008, S. 405). In der klassischen Stereovision werden die Aufnahmen mit fest installierten, synchronisierten Kameras erzeugt (BRADSKI & KAEHLER 2008, S. 405). Durch die relative Verschiebung der Merkmale – dies können markante Pixel, aber auch Konturen sein – in den verschiedenen Aufnahmen kann die Entfernung trianguliert werden.

Die Information über die Tiefe steckt in der relativen Verschiebung gleicher Merkmale in unterschiedlichen Aufnahmen. Bei statischen Objekten können diese Aufnahmen auch durch Verfahren einer einzelnen Kamera generiert werden (BRADSKI & KAEHLER 2008, S. 453). Man spricht dann von *Mono-Stereovision* oder „*motion stereo*“ (SUK & BHANDARKAR 1992a, S. 27). Es gibt kaum kommerzielle Mono-Stereosysteme (SUK & BHANDARKAR 1992a, S. 28).

Im weiteren Verlauf des Kapitels werden Ablauf und Funktionsweise der Stereovision erklärt. Es wird insbesondere auf die Schwierigkeiten der Mono-Stereovision eingegangen. Ohne Verlust der Allgemeingültigkeit werden im Folgenden die Prinzipien der Stereovision anhand von zwei Aufnahmen erläutert. Sämtliche Erklärungen treffen auch für mehr Aufnahmen zu. Der Einfachheit halber wird der Begriff „Pixeln“ synonym für „Merkmal“ verwendet. Das liegt daran, dass häufig Algorithmen zur Detektion markanter Punkte (s. Abschnitt 4.3) verwendet werden.

### 5.1 Grundlagen der Stereovision

Die Stereovision unterteilt sich im Wesentlichen in vier Schritte (BRADSKI & KAEHLER 2008, S. 415):

1. Entzerrung (englisch „undistortion“)

2. Rektifikation
3. Korrespondenzanalyse
4. Triangulation

Im ersten Schritt werden die Aufnahmen von kameraspezifischen Verzerrungen und Störungen befreit. Danach werden sie so ausgerichtet, dass gleiche Pixel in den Aufnahmen in der gleichen horizontalen Ebene (Linie) liegen. Diese Ausrichtung erleichtert die anschließende Korrespondenzanalyse erheblich (BRADSKI & KAEHLER 2008, S. 419). *Korrespondierende Punkte* sind Bildpunkte in beiden Bildern, die den gleichen realen (3D) Punkt abbilden. Mit ihnen – genauer: ihrer relativen Verschiebung – wird im letzten Schritt (Triangulation) die Entfernung der Pixel in 3D berechnet.

Die radiale und tangentielle Verzerrung wird bei der geometrischen Kamerakalibrierung in fünf Koeffizienten gemessen (vgl. Abschnitt 3.5). Die Algorithmen zur Bildentzerrung sind Standard und werden hier nicht behandelt.

## 5.2 Rektifikation

Sind zwei Kameras zur Erzeugung eines Stereo-Bildpaares ideal ausgerichtet, so verlaufen ihre optischen Achsen parallel zu einander (NAWRATH 2013). Eine solche Ausrichtung wird auch als *Standard-Stereo-* oder *achsparallele* („frontal parallel“) Konfiguration bezeichnet (NAWRATH 2013). Sie ist in Abbildung 5.1 dargestellt. Der reale Punkt  $P = (X_P, Y_P, Z_P)$  wird auf die Bildpunkte  $p_l = (x_l^p, y_l^p)$  und  $p_r = (x_r^p, y_r^p)$  projiziert. Bei angenommener perfekt horizontaler Montage der Kameras ( $y_{p,l} - y_{p,r} = 0$ ) kann über die Differenz der Pixelkoordinaten  $x_{p,l} - x_{p,r}$  der Abstand  $Z_P$  berechnet werden.<sup>22</sup> Dies wird in Abschnitt 5.3 behandelt.

Montageungenauigkeiten bei den Kameras selbst und bei dem Aufbau der Stereokonfiguration führen dazu, dass eine achsparallele Konfiguration in der Realität so gut wie nie vorkommt (BRADSKI & KAEHLER 2008, S. 417). Die Bilder müssen nachträglich in diese überführt werden. Das geschieht in zwei Schritten. Mit der Fundamentalmatrix (s. Abschnitt 5.2.2) wird die Ausrichtung der Kameras berichtigt. Anschließend werden

<sup>22</sup>Der Abstand  $T$  der optischen Achsen und die Brennweite  $f$  werden dazu ebenfalls benötigt. Dies sind i. d. R. feste Werte, die bei der Montage bzw. der Kalibrierung bestimmt werden.

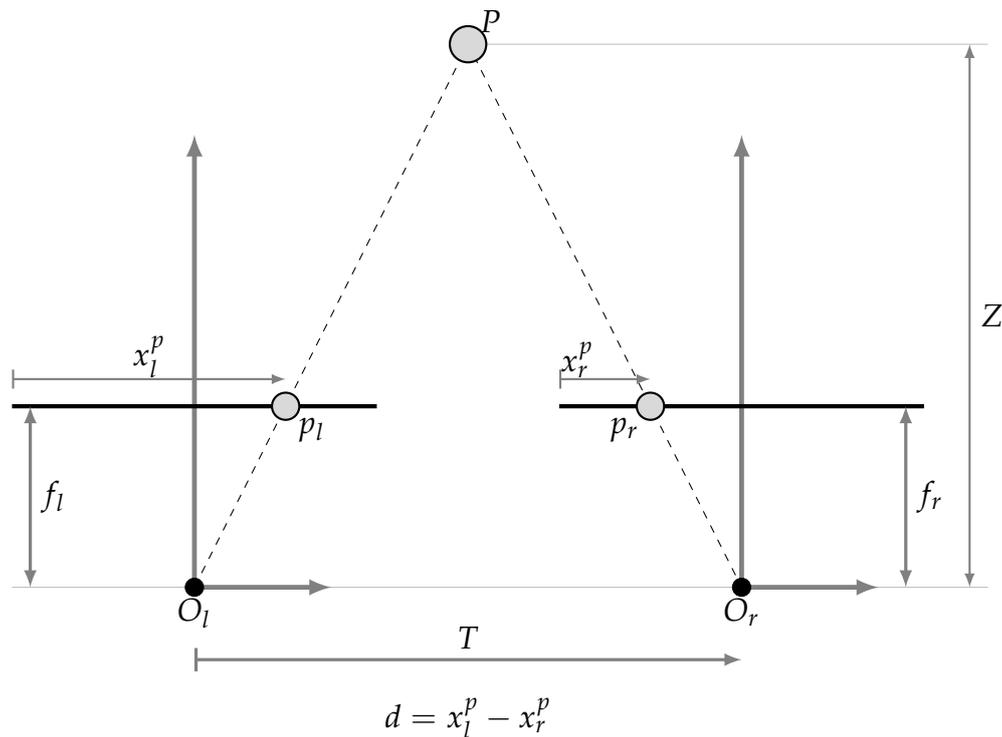


Abbildung 5.1: Standard-Stereo Aufbau. Die optischen Achsen der Kameras verlaufen parallel. Aus den korrespondierenden Bildpunkten  $x_l^p, x_r^p$  kann mittels Triangulation über die Brennweite  $f$  und den Abstand  $T$  der optischen Achsen bzw. der Projektionsmittelpunkte  $O_l, O_r$  bestimmt werden. Die Bildebenen sind zu einer kompakteren Darstellung nach vorne geklappt. Nach (BRADSKI & KAEHLER 2008, S. 416).

die Bilder zeilenweise ausgerichtet, sodass – wie im obigen Beispiel angenommen – gilt:  $y_{p,l} - y_{p,r} = 0$ .

Dieser Ablauf heißt *Rektifikation*. Es ist der Prozess, Bilder so zu „korrigieren“, dass Punkte eines realen Objekts in beiden Bildern in dieselbe Bildzeile abgebildet werden (BRADSKI & KAEHLER 2008, S. 427). Obwohl die Bilder nachträglich in die achsparallele Konfiguration umgerechnet werden, ist es wichtig, die Kameras möglichst horizontal und parallel auszurichten, da es sonst zu erheblichen Verzerrungen bei der Rektifikation kommen kann (BRADSKI & KAEHLER 2008, S. 418).

### 5.2.1 Epipolargeometrie

*Epipolarlinien* beschreiben die Geometrie eines Zwei-Ansichten-Systems (LAGANIERE 2014, S. 294). Mit ihnen kann die Korrespondenzanalyse, die Suche nach einem Bild-

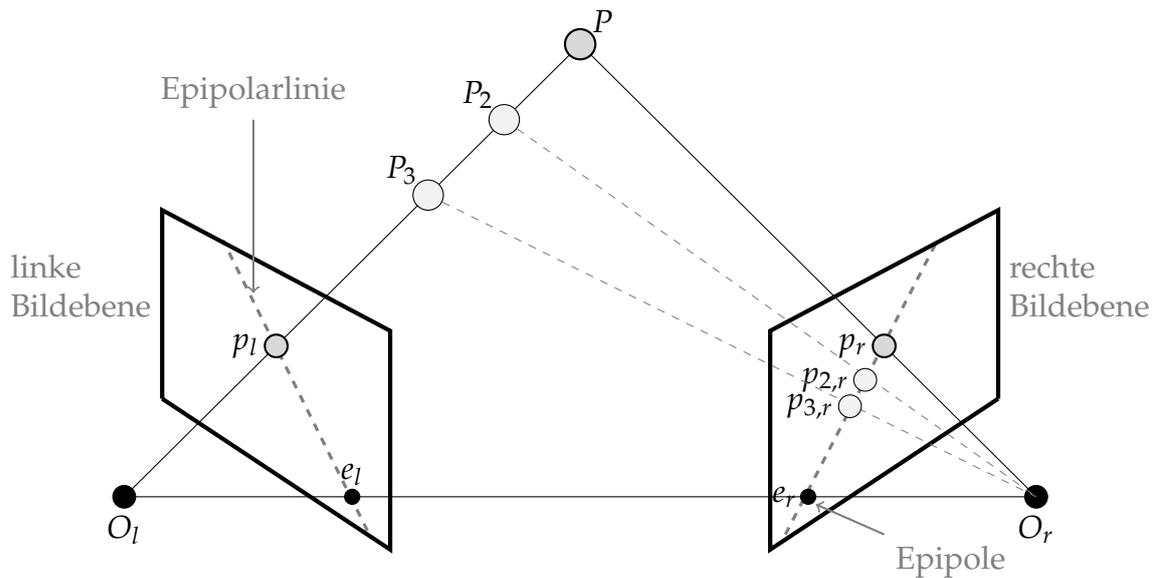


Abbildung 5.2: Epipolargeometrie. Die Epipolarebene ist durch die Punkte  $O_l$ ,  $P$ ,  $O_r$  vollständig beschrieben. Die Epipole  $e_l$ ,  $e_r$  liegen an den Schnittpunkten der Verbindung zwischen den Projektionsmittelpunkten ( $O_l$ ,  $O_r$ ) mit der jeweiligen Bildebene. Die Epipolarlinie geht durch den Epipol und die Projektion des Punktes  $P$  auf der Bildebene.

Nach (BRADSKI & KAEHLER 2008, S. 420).

punkt im linken und rechten Bild, der den gleichen realen Punkt beschreibt, deutlich vereinfacht werden (BRADSKI & KAEHLER 2008, S. 419; AHLVERS & ZÖLZER 2005, S. 4).

In Abbildung 5.2 sind die Elemente der Epipolargeometrie eingezeichnet. Die Epipolarebene wird durch den realen Punkt  $P$  und den Optischen Zentren der Kameras  $O_l$  und  $O_r$  bestimmt. Die Schnittlinie der Epipolarebene mit den Bildebene der jeweiligen Kamera heißt Epipolarlinie. Auf ihr liegt die Projektion des realen Punktes  $P \rightarrow p_l, p_r$ . Alle Punkte  $P, P_2, P_3$  auf der Verbindungslinie  $\overline{O_l P}$  werden im linken Bild auf den Punkt  $p_l$  abgebildet. Im rechten Bild degenerieren die realen Punkte nicht, sondern ihre Bildpunkte liegen auf der gleichen Epipolarebene.<sup>23</sup> Diese Eigenschaft der Epipolargeometrie bzw. der Epipolarlinien führt dazu, dass die Suche nach dem korrespondierenden Pixel im anderen Bild auf ein 1D Problem reduziert wird (AHLVERS & ZÖLZER 2005, S. 4).

Epipolarlinien sind Eigenschaften zweier Bilder. Alle Epipolarlinien eines Bildes schnei-

<sup>23</sup>Die Transformation zwischen den Bildpunkten  $(p_l, p_r)$  wird durch die *Homographiematrix* beschrieben. Sie ist eine reine Abbildungsvorschrift zwischen den Ebenen, ohne Kompensation etwaiger Verzerrungen oder Störungen (BRADSKI & KAEHLER 2008, S. 384). Sie ist in der im nächsten Abschnitt vorgestellten Fundamentalmatrix implizit enthalten und wird hier nur der Vollständigkeit halber erwähnt werden.

den sich in einem Punkt, dem *Epipole*  $e_l, e_r$  (LAGANIERE 2014, S. 295). Anschaulich beschrieben ist dieser Punkt die Projektion des Standpunkts der rechten Kamera im Bild der linken Kamera und vice versa (BRADSKI & KAEHLER 2008, S. 419). Bei achsparalleler Ausrichtung der Kameras liegen die Epipole im Unendlichen und die Epipolarlinien verlaufen parallel zu einander (AHLVERS & ZÖLZER 2005, S. 2). Die Epipolarlinien eines Stereo-Bildpaars sind in Abbildung 6.6 in Abschnitt 6.4.2 dargestellt.

Rektifizierung ist der Prozess, der die Bilder so transformiert, dass ihre Epipolarlinien-Paare kollinear und parallel zur x-Achse werden (AHLVERS & ZÖLZER 2005, S. 14). So muss bei der Korrespondenzanalyse nur die gleiche Zeile in beiden Bildern nach korrespondierenden Punkten durchsucht werden.

### 5.2.2 Fundamentalmatrix

Die Rotation  $R$  und Translation  $T$  zwischen zwei der Projektion  $(p_l, p_r)$  eines (3D) Punktes kann wie folgt berechnet werden:  $p_r = R(p_l - T)$ . Abbildung 5.3 illustriert dies. Unter Verwendung der Epipolarebene in der Hesseschen Normalform lässt sich der Zusammenhang umformulieren:<sup>24</sup>

$$(R^T p_r)^T (T \times p_l) = 0 \quad (5.1)$$

$$\Rightarrow R(p_r)^T S p_l = 0 \quad (5.2)$$

$$\Rightarrow (p_r)^T E p_l = 0 \quad (5.3)$$

Die Matrix  $S$  entspricht dem als Matrix-Vektor-Produkt umgeschriebenen Kreuzprodukt von  $T \times p_l$ .<sup>25</sup>  $E$  ist die *Essentialmatrix* (englisch „essential matrix“). Es gilt  $\text{rang}(E) = 2$ , die Matrix ist also nicht eindeutig (BRADSKI & KAEHLER 2008, S. 422). Die Essentialmatrix enthält rein geometrische Informationen und kann deshalb Punkte in physikalischen (realen) Koordinaten oder im Kamerakoordinatensystem abbilden, nicht jedoch in

<sup>24</sup>Die neue Ebene  $(p_l - T)^T (T \times p_l) = 0$  mit  $p_r = R(p_l - T)$  wird zu Gleichung 5.2.

<sup>25</sup>allgemein:

$$\vec{a} \times \vec{b} = \Omega \vec{b} \quad \text{mit} \quad \Omega = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

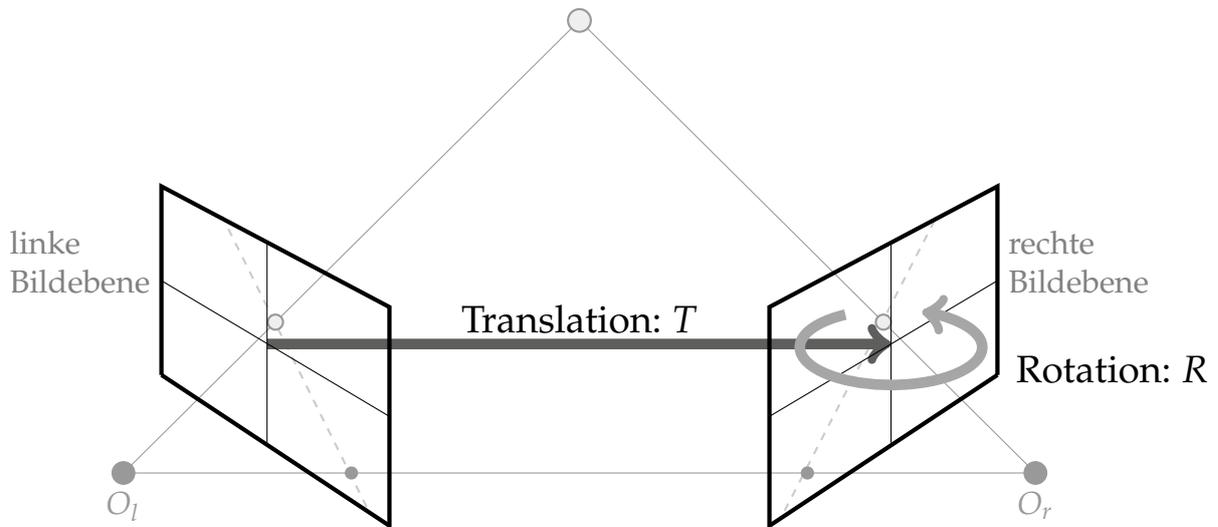


Abbildung 5.3: Die Essentialmatrix  $E$  beschreibt die Rotation  $R$  und Translation  $T$  zwischen der Bildebene der linken Kamera und der Bildebene der rechten Kamera.  
Nach (BRADSKI & KAEHLER 2008, S. 421).

Pixel-Koordinaten (BRADSKI & KAEHLER 2008, 422f). Dazu müssen die intrinsischen Kameraparameter (vgl. Abschnitt 3.5) berücksichtigt werden.

$$q_r^T (M_r^{-1})^T E M_l^{-1} q_l = 0 \quad (5.4)$$

$$\Rightarrow q_r^T F q_l = 0 \quad \text{mit } F = (M_r^{-1})^T E M_l^{-1} \quad (5.5)$$

$(q_l, q_r)$  sind die Projektionen des Punktes  $P \rightarrow (p_l, p_r)$  in Pixel-Koordinaten;  $M$  ist die intrinsische Kameramatrix (s. Gleichung 3.10). Die Matrix  $F$  heißt *Fundamentalmatrix* und beschreibt die Transformation zwischen der linken und der rechten Kamera in Pixelkoordinaten (BRADSKI & KAEHLER 2008, S. 412). Mit ihr kann man das rechte Bild so umrechnen, als wäre es von der Position der linken Kamera gemacht worden.

Die Fundamentalmatrix ist eine  $3 \times 3$  Matrix. Es müssen also 9 Einträge bestimmt werden. Wird eine Skalierung von 1 verwendet und berücksichtigt, dass alle Epipolarlinien durch einen Punkt gehen ( $\det(F) = 0$ ), so werden nur 7 korrespondierende Punkte benötigt (LAGANIERE 2014, S. 298). Diese können bspw. durch ein Muster mit markanten Ecken erzeugt werden. Dies wird im nächsten Kapitel behandelt. Abschnitt 5.2.3 erklärt, wie die Fundamentalmatrix ohne ein festes Muster geschätzt werden kann.

### 5.2.3 Bestimmen der Fundamentalmatrix mit und ohne Stereo-Kalibrierung

Ziel der Stereo-Kalibrierung ist es, die Fundamentalmatrix zu bestimmen, mit der die Position der rechten Kamera in die der linken umgerechnet werden kann (BRADSKI & KAEHLER 2008, S. 428). Die Stereo-Kalibrierung erfolgt – wie bei der geometrischen Kalibrierung – mit einem (Schachbrett-)Muster. Aus den definierten (korrespondierenden) Punkten kann die Fundamentalmatrix bestimmt werden.

Die minimale Anzahl korrespondierender Punkte ist sieben (vgl. Abschnitt 5.2.2). Das Ergebnis kann jedoch instabil sein, weshalb mehr Punkte verwendet werden sollten (LAGANIERE 2014, S. 297). Die Epipolarbedingung wird i. d. R. durch eine niedrige Toleranz  $\varepsilon$  relativiert:

$$p_r^T F p_l = 0 \quad \Rightarrow \quad p_r^T F p_l < \varepsilon \quad (5.6)$$

Nun werden 8 korrespondierende Punkte zur Berechnung von  $F$  benötigt. Dafür reicht ein  $5 \times 3$  Schachbrettmuster mit 8 innen liegenden Punkten. Stehen mehr Punkte zur Verfügung, z. B. durch ein größeres Schachbrettmuster, kann bspw. der RANSAC Algorithmus verwendet werden.

*RANdom SAmples Consensus (RANSAC)* ist ein sehr robuster Algorithmus zur Parameterbestimmung einer mathematischen Funktion. Das Funktionsprinzip ist simpel und in Abbildung 5.4 dargestellt. Es wird am Beispiel einer Ausgleichsgeraden erklärt. Es werden zwei beliebige Punkte verwendet, um eine Gerade zu bestimmen. Anschließend wird überprüft, wie viele Punkte diese Gerade „unterstützen“, d. h. innerhalb eines Toleranzbandes der Breite  $\varepsilon$  liegen. Die Toleranz wird meist empirisch gewählt (HARTLEY & ZISSERMAN 2003, S. 118). Die Gerade, die von den meisten Punkten unterstützt wird, ist das Resultat. Im Falle der Fundamentalmatrix wird die Matrix gesucht, mit der die meisten korrespondierenden Punkte Gleichung 5.6 erfüllen.

RANSAC liefert auch bei vielen Ausreißern bzw. falschen Punkten noch gute Ergebnisse (HARTLEY & ZISSERMAN 2003, S. 117; NÜCHTER & HERTZBERG 2008, S. 3). Da die Parameter immer aus der besten Teilmenge der zur Verfügung stehenden Punkten bestimmt werden, werden sie nicht von Ausreißern beeinflusst.

Wird – wie im vorliegenden Fall – keine fester Stereo-Aufbau verwendet, oder ist aus einem anderen Grund eine Stereo-Kalibrierung der Kamera nicht möglich, so kann mit dem RANSAC Algorithmus die Fundamentalmatrix geschätzt werden. Das Ergebnis

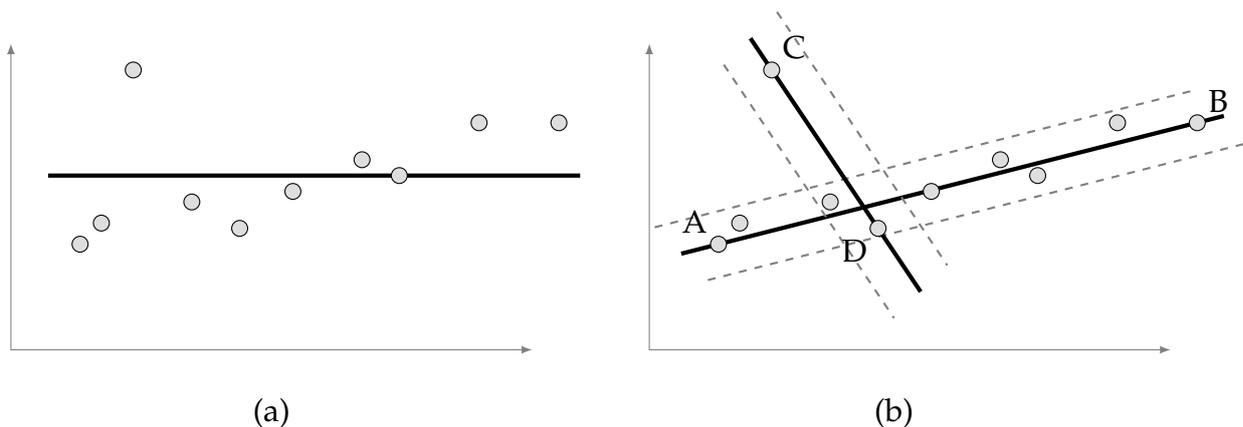


Abbildung 5.4: Funktionsweise des RANSAC Algorithmus am Beispiel einer Geradenanpassung. (a) zeigt eine klassische Ausgleichsgerade nach der Summe der kleinsten quadrierten Abweichungen („least-squares“). Diese wird stark von dem Ausreißern beeinflusst. (b): RANSAC hingegen verwendet jeweils zwei Punkte zum Beschreiben einer Geraden und überprüft, wieviele Punkte innerhalb eines Toleranzbereichs um diese Gerade liegen. Die Gerade  $\overline{AB}$  wird bspw. von 9 Punkten unterstützt, die Gerade  $\overline{CD}$  hingegen nur von 3. RANdom SAmple Consensus wählt die Gerade mit der höchsten Unterstützung aus.  
Nach (HARTLEY & ZISSERMAN 2003, S. 117).

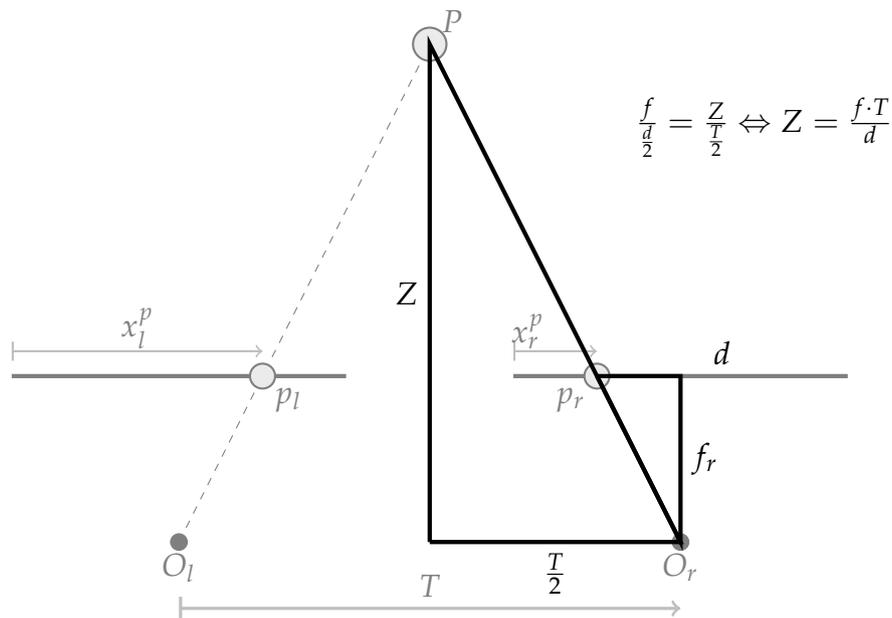
hängt von der Qualität der zuvor – z. B. mit dem ORB Algorithmus – gefundenen korrespondierenden Punkten ab. Im Allgemeinen wird jedoch von einem solchen Vorgehen abgeraten (AHLVERS & ZÖLZER 2005, S. 16).

Bei einem Mono-Stereosystem lässt sich dies jedoch nicht vermeiden. Durch Wiederholungsgenauigkeiten beim Verfahren des Roboters lässt sich das System nicht stereo-kalibrieren.<sup>26</sup>

### 5.3 Korrespondenzanalyse und Triangulation

In der Korrespondenzanalyse wird für jedes (markante) Pixel im linken Bild das korrespondierende Pixel im rechten Bild gesucht (BRADSKI & KAEHLER 2008, S. 440). Nach der Rektifizierung brauchen dafür nur noch gleiche Zeilen untersucht zu werden. Dies geschieht üblicherweise mit einem „block matching“ Algorithmus (STEINGRUBE et al.

<sup>26</sup>Mit einem Muster, das auf dem Werkstück oder gut sichtbar an der Zelle des Roboters installiert ist, ließe sich das System bei jeder Stereoaufnahme neu stereo-kalibrieren. Ein solcher „Marker“ am Werkstück führt jedoch die Objekterkennung ad absurdum. Ein fest installiertes Muster, z. B. am Boden, ist eine praktikable Möglichkeit, verletzt aber die Randbedingung der Arbeit, dass die Roboterzelle nicht verändert werden darf (vgl. Abschnitt 1.2.2).

Abbildung 5.5: Triangulation der Tiefe  $Z$  von Punkt  $P$ .

2009, S. 292).<sup>27</sup> Dabei wird entlang mehrere 1D Pfade die „Energie“ eines Blocks minimiert (GEHRIG et al. 2009, S. 134). Die Energie setzt sich aus den Intensitätswerten im rechten Bild zusammen, die angepasst werden müssen, um den betrachteten Intensitätswerten aus dem linken Bild zu entsprechen.

Mit der Korrespondenzanalyse wird eine Karte des Bildes erstellt, die die Verschiebungen (*Disparitäten*) zwischen den korrespondierenden Punkten enthält: die sogenannte *Disparity-Map* (BRADSKI & KAEHLER 2008, S. 421). Aus ihr wird die Tiefenkarte (englisch „Depth Map“) bestimmt. Abbildung 5.1 zeigt beispielhaft die Triangulation der Entfernung eines Punktes anhand des Unterschieds in den Bildern einer Stereo-Kamera. Bildet man nun die Dreiecke, wie sie in Abbildung 5.5 eingezeichnet sind, so lässt sich aus der Disparität  $d = x_{p,l} - x_{p,r}$  der Abstand  $Z$  berechnen:

$$\tan(\alpha) = \frac{Z}{T/2} = \frac{f}{d/2} \quad (5.7)$$

$$\Rightarrow Z = \frac{f T}{d} \quad (5.8)$$

Macht man dies für alle korrespondierenden Punkte und füllt die dazwischen liegenden Pixel mit Schätzwerten, so erhält man eine Tiefenkarte, wie sie in Abbildung 6.8 dargestellt ist. Die Tiefenkarte ist eine Visualisierung der 3D-Punktewolke, die mit der

<sup>27</sup>Auch OpenCV (2.4.11) verwendet ausschließlich Block-Matching-Algorithmen.

Korrespondenzanalyse und Triangulation erzeugt wurde.

Die Genauigkeit, mit der der Abstand  $Z$  bestimmt werden kann, sinkt mit der Entfernung zur Kamera und steigt mit dem „Augenabstand“ bzw. dem *Basisabstand*  $T$  (BRADSKI & KAEHLER 2008, S. 417).

$$\Delta Z = \frac{Z^2}{f T} d \quad (5.9)$$

Der Basisabstand ist ein Kompromiss aus der Genauigkeit der Entfernungsmessung und der Überlappung der Stereo-Bilder und damit des Sichtbereichs der Stereo-Kamera. Der Abstand  $Z$  des Objekts ist meist durch externe Randbedingungen vorgegeben. Hier ist er vom Sichtbereich der Kamera und der Zellengröße beschränkt.<sup>28</sup>

Aus der so bestimmten 3D-Punktwolke kann Position und Lage eines Objekts bestimmt werden. Theoretisch kann auch das Objekt anhand der Punktwolke erkannt werden. Insbesondere Letzteres setzt viele Punkte hoher Qualität voraus. Diese Anforderungen sind häufig nicht ohne Weiteres zu erfüllen, wie in Abschnitt 6.4 beschrieben wird.

Zum Bestimmen der relativen Objektposition in Richtung der  $x$ - und  $y$ -Achse gilt (BRADSKI & KAEHLER 2008, S. 373):

$$x = f_x \frac{X}{Z} + c_x \quad (5.10)$$

$$\Leftrightarrow X = \frac{x - c_x}{f_x} Z \quad (5.11)$$

$$(5.12)$$

Große Koordinaten bezeichnen die reale Position in physikalischen Einheiten.  $c_x$  ist die  $x$ -Koordinate des Optischen Zentrums und  $f_x$  die Brennweite. Mit Gleichung 5.8 und der Annahme, dass gilt  $f_x \approx f_y \approx f$  ergibt sich:

$$X = (x - c_x) \frac{T}{d} \quad (5.13)$$

<sup>28</sup>Folglich sind zur Extraktion von Tiefeninformationen eine Weitwinkelobjektiv und eine nahe Positionierung der Kamera am Objekt wünschenswert. Bei Weitwinkelobjektiven muss die Bildverzerrung zu den Rändern hin beachtet werden (s. a. Abschnitt 3.5). In dieser Arbeit ist die Kamera durch den Sensor vorgegeben.

Die y-Koordinate kann analog berechnet werden. Die relative Position in Richtung der x- und y-Achse ist unabhängig von der Brennweite.

#### 5.4 Zusammenfassung und Bezug zum Projekt

Zum Erstellen einer Tiefenkarte müssen die Stereoaufnahmen komplanar und achsparallel ausgerichtet werden. Bei einem Mono-Stereosystem muss dazu die Fundamentalmatrix geschätzt werden. Dies wird anhand korrespondierender Pixel in der linken und rechten Sensoraufnahme gemacht. Zur Detektion dieser Pixel wird der ORB Algorithmus aus Abschnitt 4.3.3 verwendet.

Die Tiefeninformation einzelner (korrespondierender) Konturen kann auch mittels Triangulation extrahiert werden. Dazu wird der Unterschied (Disparität) zwischen den Konturmittelpunkten bestimmt und nach Gleichung 5.8 daraus ihr Abstand relativ zum Roboter berechnet:

$$Z = \frac{f T}{d} \quad (5.8)$$

Der Roboter kann so die Position seines Werkstücks bestimmen und dieses selbstständig anfahren.

## 6 Praktische Umsetzung der modellbasierten Objekterkennung

Dieses Kapitel beschreibt detailliert das Vorgehen bei der Implementierung der Algorithmen und ihre Überführung in die Praxis. Dabei wird speziell auf Besonderheiten des Systems eingegangen.

Zunächst wird die verwendete Soft- und Hardware beschrieben. Anschließend wird auf die Umsetzung der Objekterkennung und der Stereovision eingegangen.

### 6.1 System

Das beschriebene Vorgehen in dieser Arbeit ist prinzipiell allgemeingültig und unabhängig von den Eigenschaften des Systems. Es bestehen system- und hardwarebedingte Einschränkungen, die bereits in der Beschreibung der Methodik erwähnt wurden, und auf die im Folgenden detaillierter eingegangen wird.

Die erzielten Ergebnisse sind speziell auf das System bezogen, werden aber nach Möglichkeit verallgemeinert und für universelle Werkstücke angepasst. Dieser Abschnitt beschreibt die verwendete Infrastruktur.

Der Industrieroboter ist ein „KR 90 R3100 HA extra“ Roboter von KUKA (KUKA 2013). Zur Steuerung wird die handelsübliche KUKA-Steuerung verwendet. Der Roboter führt Point-To-Point (PTP) Bewegungen aus. Ein externer PC gibt die Position der Achswinkel vor, die mit einer lehrstuhleigenen Software berechnet werden. Die externe Steuerung des Roboters ist in C# geschrieben. Die Roboterzelle ist  $2,8\text{ m} \times 3,3\text{ m}$  groß.

Der Sensor ist ein Helix Laserscanner von Perceptron. Er projiziert eine Laserlinie wird auf das zu messende Objekt und über dessen Reflektion auf einen Kamerasensor trianguliert wie eine Punktwolke. Das Aufnahmevolumen des Laserscanners beträgt  $20\text{ cm} \times 20\text{ cm} \times 20\text{ cm}$  und ist daher ungeeignet zum Erkennen großer Objekte in der

Roboterzelle. Der Kamerasensor ist für die Lichtintensität der Laserreflektion ausgelegt, das Graustufenbild daher sehr dunkel (vgl. Abbildung 3.2(a)). Die Aufnahme wird nachträglich ausgelesen und hat eine Auflösung von  $968 \times 1296$  Pixeln. Der Aufnahmebereich des Kamerasensors ist jedoch eingeschränkt, weshalb effektiv nur ein Bild von  $615 \times 1296$  Pixeln zur Verfügung steht. Von dem Kamerasensor liegt kein Datenblatt vor.

Zur Bildanalyse wird die freie Softwarebibliothek *OpenCV* in der Version 2.4.11 verwendet.<sup>29</sup> *OpenCV* ist eine umfassende Bibliothek zur Bildverarbeitung, die viele Standardalgorithmen aber auch neueste Veröffentlichungen implementiert hat. *OpenCV* ist in den Programmiersprachen C, C++ und Python verfügbar. In dieser Arbeit wurde die C++ Implementierung verwendet.

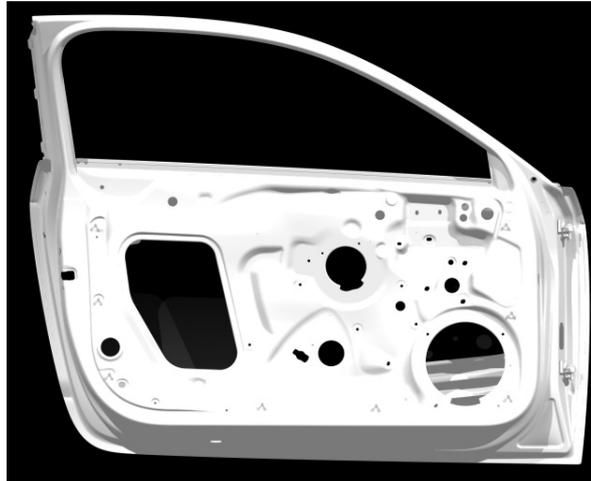
Die Berechnungen für diese Arbeit wurden in einer Virtuellen Maschine (Oracles „Virtual Box“ 4.3.3) mit 2048 MB Arbeitsspeicher und 90% CPU Begrenzung durchgeführt. Als Gastbetriebssystem wurde Windows 7 (32-bit) verwendet. Der reale Computer ist ein MacBook 13 von 2010 mit dem Betriebssystem OS X 10.9.5, 4GB Arbeitsspeicher und einem 2.4 GHz Intel Core 2 Duo Prozessor.

Das Modell ist eine CAD Datei, die mit CATIA V5 erstellt wurde. Das virtuelle Objekt (die Tür) wird so ausgerichtet, dass es annähernd der Ausrichtung des realen Objekts entspricht. Dann wird ein Rendering<sup>30</sup> erzeugt, bei dem das Objekt einfarbig ist und einen hohen Kontrast zum Hintergrund aufweist (z. B. schwarzer Hintergrund, weißes Objekt). Bei Bohrungen und Hinterschneidungen wurde der hintere Objektteil auf transparent gesetzt. Der Hintergrund ist auf diffus leuchtend eingestellt und auf zusätzliche virtuelle Lampen wurde verzichtet. Mit dieser Konfiguration werden Schatten und Spiegelung verhindert. Als Auflösung sind 150 dpi oder 300 dpi ausreichend. Abbildung 6.1 zeigt das CAD Objekt als aufbereitetes Rendering.

---

<sup>29</sup>Zum Startzeitpunkt der Arbeit war bereits die Nachfolgeversion 3.0.0 freigegeben. Wegen der lückenhaften Dokumentation wurde dennoch die ältere Version gewählt.

<sup>30</sup>Als „rendern“ wird in der Computergrafik der Vorgang bezeichnet, bei dem eine Vektorgrafik in eine 2D Pixelgrafik umgerechnet wird.



*Abbildung 6.1: Gerendertes Modellbild für die Objekterkennung.  
Diffus leuchtender Hintergrund, matt farbenes Objekt, transparente Objektrückseite, 150 dpi.*

### 6.2 Vorverarbeitung

In Kapitel 3 wurden die Grundlagen der Bildverarbeitung erklärt. Hier wird nun einen Überblick über die verwendeten Methoden gegeben.

Das Sensorbild wird auf 60 % seiner Größe skaliert, um die nachfolgenden Verarbeitungsschritte zu beschleunigen und damit es vollständig auf einem normalen Bildschirm (24 ") dargestellt werden kann. Das Bild ist sehr dunkel<sup>31</sup> und muss mittels einer Histogrammanpassung in den für Menschen sichtbaren Kontrastbereich transformiert werden. Ein erhöhter Kontrast unterstützt auch höhere Bildverarbeitungsalgorithmen wie den Canny Kantendetektor oder das Erkennen markanter Punkte (vgl. Kapitel 4). Um Beleuchtungsänderungen in der Roboterzelle – bspw. durch (Dach)Fenster – vorzubeugen, wird der CLAHE Algorithmus verwendet. Die Aufnahme wird in  $6 \times 6$  Regionen unterteilt und der Kontrastschwellwert liegt bei  $40 \cdot \text{Mittelwert}$ . Das Ergebnis ist in Abbildung 6.2 (b) dargestellt.

In einem nachfolgenden Schritt wird das angepasste Bild für die Kantendetektion vorbereitet. Sollen markante Punkte detektiert werden, so müssen die folgenden Schritte nicht durchgeführt werden. Die Kantendetektion reagiert sehr empfindlich auf Bildrauschen. Trotz der Rauschen unterdrückenden Wirkung der CLAHE muss das Bild geglättet

---

<sup>31</sup>Der Sensor ist für die hohe Intensität des reflektierten Laserstrahls ausgelegt.

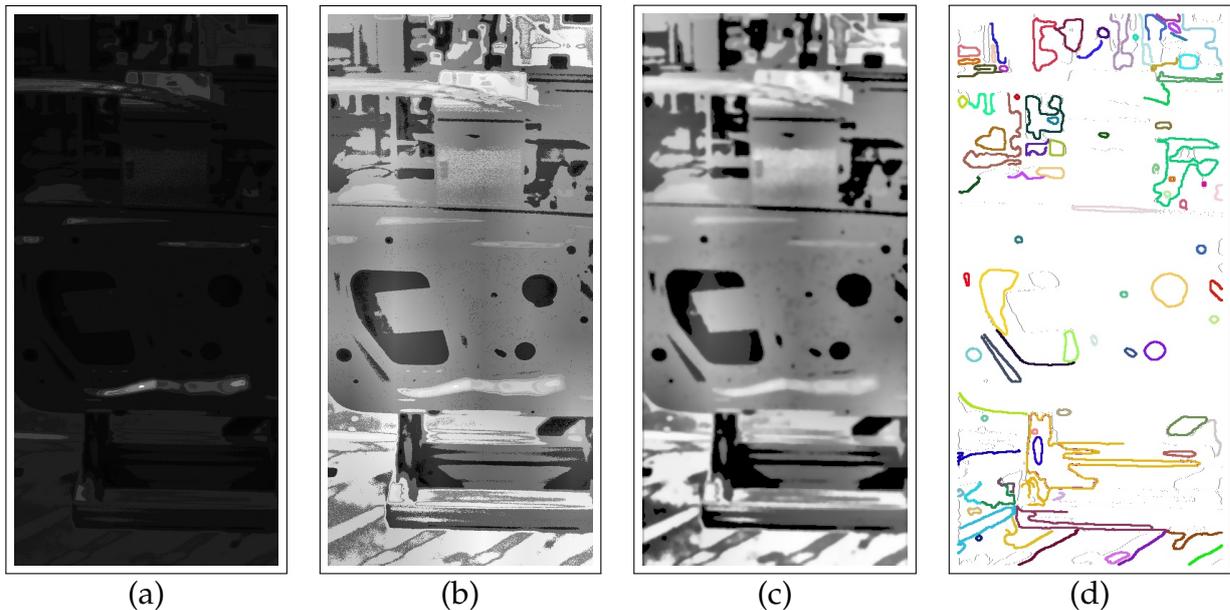


Abbildung 6.2: Bildverarbeitungsschritte bis zur Konturdetektion. (a) Originalbild; (b) Bild mit angepasstem Histogramm (CLAHE); (c) Gauß-, Bilateral-Filter und Schwellwertoperation; (d) segmentierte Kanten = Konturen.

werden. Hierfür wird ein Gauß-Filter mit einer  $5 \times 5$  großen Umgebung verwendet.<sup>32</sup> Anschließend wird ein Bilateralfilter benutzt, um Kanten stärker zu definieren. Bilaterale Filter sind aufwendig zu berechnen (PARIS et al. 2007, S. 3-30). Ein recht kleiner Filterausschnitt von  $7 \times 7$  Pixeln genügt hier. Dieser lässt sich in etwa 1,5 Sekunden berechnen. Abbildung 6.2(c) zeigt das Ergebnis.

Bohrungen und Hinterschneidungen sind die markanten Objekte, an denen das vorliegende System die Karosserie erkennt. Sie sind i. d. R. dunkler als ihre Umgebung<sup>33</sup>. Damit diese Löcher immer vollständig erkannt werden, wird eine Schwellwertoperation („Thresholding“) mit Relation zum Mittelwert der Intensitäten  $\bar{I}$  durchgeführt. So können kleinere Änderungen der Grauwerte innerhalb der Löcher unterdrückt werden.

$$I(x, y) \begin{cases} < \frac{3}{7} \bar{I} & \Rightarrow I(x, y) = 0 \\ \text{sonst} & - \end{cases}$$

Erst danach wird mit dem Canny Kantendetektor ein binäres Kantenbild erzeugt. Dieses ist in Abbildung 6.2 (d) dargestellt. Als oberen und unteren Schwellwert wurden für

<sup>32</sup>Bei einem Gauß-Filter kann entweder die Varianz  $\sigma$  und damit die „Steilheit“ der Gauß-Funktion angegeben werden oder die Größe, bis zu der diese (für diskrete Werte) abgeklungen sein soll.

<sup>33</sup>Sofern das Werkstück nicht durchleuchtet wird.

die Sensorbilder 120 : 260 und für das gerenderte Bild vom Modell 200 : 400 gewählt. Sie liegen innerhalb des empfohlenen Bereichs zwischen 1 : 2 und 1 : 3 (BRADSKI & KAEHLER 2008, S. 152). Die Werte sind empirisch bestimmt. Allgemein gilt: je höher die Schwellwerte, desto stärkere Kanten werden erkannt. Die Anzahl der Kanten sinkt jedoch im Allgemeinen mit steigenden Werten. Ob eine Kante in ihrer ganzen Länge erkannt wird, hängt vom unteren Schwellwert ab; ob sie überhaupt erkannt wird, vom oberen (vgl. Abschnitt 3.3). Durch eine Segmentierung des binären Kantenbildes erhält man einzelne Konturen. Diese sind farbig markiert in Abbildung 6.2 (e) zu sehen.

Bei dem aus der CAD Datei erzeugten Bild entfallen die Vorverarbeitungsschritte zur Histogrammanpassung, die Filter und die Schwellwertoperation. Es können direkt Kanten detektiert werden.

Die nächsten Kapitel erläutern, wie die Konturen aus verschiedenen Bildern einander zugeordnet werden.

### 6.3 Objekterkennung: Konturen

Bei der *modellbasierten* Objekterkennung soll ein bspw. als CAD Datei vorliegendes Objekt in einem Kamerabild erkannt werden (BÖHM 2004, S. 64). Prinzipbedingt kann diese Aufgabe nicht mit markanten Punkten gelöst werden. Da beim eindeutigen Beschreiben der markanten Punkte (Deskription – vgl. Abschnitt 4.3) die umgebenden Punkte kodiert werden, müssen die Bilder ähnliche Farb- bzw. Intensitätswerte aufweisen. Dies ist beim Vergleich eines CAD Modells mit einer Kameraaufnahme nicht gegeben. Abbildung 7.9 (c) zeigt bspw. Modell- und Sensorbild.

Rein geometrische Merkmale, wie Konturen, sind für eine solche Zuordnung besser geeignet. Geometrische Merkmale – so sie erkannt werden – sind unabhängig von Beleuchtung und Hintergrund. Es ist auch möglich, Objekte anhand von Linien zu erkennen. Insbesondere industriell gefertigte Werkstücke haben viele Linien und Kanten. Ein Konzept für die Zuordnung von Linien („Line-Matcher“) ist in Anhang A.3 beschrieben. Es wurde im Rahmen dieser Arbeit entwickelt, seine Untersuchung wurde jedoch wegen des begrenzten zeitlichen Umfangs dieser Arbeit eingestellt.

Konturen sind zusammenhängende Punkte, die mit hoher Wahrscheinlichkeit an einer Objektgrenze liegen (BRAHMBHATT 2013, S. 67). Es sind geometrische Merkmale, die mittels Kantendetektor – hier wird der Canny Kantendetektor (Abschnitt 3.3) verwendet

– und einer anschließenden Segmentierung erkannt werden. Sie sind robust gegenüber Bildrauschen und unabhängig von den Intensitätswerten ihrer Umgebung.<sup>34</sup> Konturen sind folglich gut geeignet zur Erkennung von Objekten mit definierten Kanten und werden deshalb in dieser Arbeit verwendet.

### 6.3.1 Beschreiben von Konturen

Konturen können durch unterschiedliche Charakteristiken beschrieben werden (LAMMERS et al. 2003, S. 9). Abschnitt 4.2.2 enthält einige Beispiele. Dieser Abschnitt beschreibt die in dieser Arbeit verwendeten geometrischen Parameter. Es wird besonders auf Skalierungs- und Rotationsinvarianz Wert gelegt. In Tabelle 6.1 stehen die verwendeten Parameter zur geometrische Beschreibung der Kontur.

Tabelle 6.1: Verwendete Parameter zur Charakterisierung von Konturen.

Beschreibung	Abkürzung	Berechnung
Exzentrizität	$\varepsilon$	nach Gleichung 4.3
Seitenverhältnis des kleinsten Rechtecks*	$a$	$\frac{h}{w}$
Umfang der Kontur zu Radius des kleinsten Kreises*	$R_1$	$\frac{l}{r}$
	$R_2$	
Höhe des Rechtecks* zu Radius des Kreises*	$hr$	$\frac{h}{r}$
Breite des Rechtecks* zu Radius des Kreises*	$wr$	$\frac{w}{r}$
Längste Strecke innerhalb der Kontur zur durchschnittlichen Streckenlänge	$\overline{max}$	$\frac{S_{max}}{\bar{s}}$
Kürzeste Strecke innerhalb der Kontur zur durchschnittlichen Streckenlänge	$\overline{min}$	$\frac{S_{min}}{\bar{s}}$
Median der Strecken innerhalb der Kontur zur durchschnittlichen Streckenlänge	$\overline{median}$	$\frac{S_{median}}{\bar{s}}$
Wertebereich der Strecken innerhalb der Kontur zur durchschnittlichen Streckenlänge	$\overline{range}$	$\frac{S_{range}}{\bar{s}}$
Summe der Hu Momente	-	nach Gleichung 4.10

\*Rechteck/Kreis kleinster Fläche, das/der die Kontur umrandet.

Gleiche Konturen aus unterschiedlichen Bildern bilden ein Konturpaar. Ein solches Konturpaar besitzt Eigenschaften, die über die Eigenschaften einer einzelnen Kontur

<sup>34</sup>Die Detektion von Kanten beruht auf der Änderung des Bildgradienten der benachbarten Pixel und damit den Intensitätswerten (vgl. Abschnitt 3.3). Allerdings kann der Kantendetektor über seine Schwellenwerte auf den jeweiligen Bildtyp – Kamerasensor oder CAD Modell – angepasst werden.

hinausgehen. So kann die Skalierung zwischen den beiden Bildern über das Flächenverhältnis (oder den Umfang) der zugeordneten Konturen bestimmt werden<sup>35</sup>:

$$S_{1,2} = A_1 / A_2 \quad (6.1)$$

Diese Eigenschaft beschreibt eine Relation zwischen zugeordneten Konturen.

Mehre Konturpaare haben darüber hinaus weitere Eigenschaften. Ihre relative Lage zueinander im Bild ist ein wichtiges Charakteristikum. Das Verhältnis der Entfernung zwischen den Flächenschwerpunkten  $A$  und  $B$  in verschiedenen Bildern liegt in der gleichen Größenordnung wie Gleichung 6.1:

$$S_{A,B} = \overline{AB}_1 / \overline{AB}_2 \quad (6.2)$$

### 6.3.2 Zuordnung von Konturen

Das Zuordnen (englisch „matching“) von Konturen aus unterschiedlichen Bildern, basiert im Wesentlichen auf dem Vergleich der geometrischen Merkmale. Die Qualität, aber auch die Anzahl der Zuordnungen hängt von den gewählten Toleranzen ab. Die gefundenen Zuordnungen können anhand ihrer Verteilung auf dem Bild, ihren relationalen Eigenschaften, überprüft werden.

Von einem richtig zugeordneten Paar ausgehend, können die relationalen Eigenschaften eine geometrische Zuordnung unterstützen, indem sie die zu überprüfenden Konturen auf eine kleinere Region des Bildes eingrenzen.

#### Geometrische Zuordnung und relationale Überprüfung

Die Zuordnung beschreibt den Prozess, gleiche Konturen in unterschiedlichen Bildern zu finden. Bei der geometrischen Zuordnung werden dafür nur geometrische Merkmale betrachtet.

Ein Merkmal zweier Konturen wird bis zu einer robusten Toleranz als „gleich“ betrachtet. Dieser Toleranzwert wird empirisch für jedes Merkmal festgelegt. Um den Vorgang

---

<sup>35</sup>Vorausgesetzt, die Bilder sind nicht unsymmetrisch verzerrt.

intuitiver zu gestalten, schlagen ABBASI et al. eine Skalierung der Abweichung auf den Bereich  $[0, 1]$  vor:

$$\alpha_\lambda = \frac{|\lambda_1 - \lambda_2|}{\max(\lambda_1, \lambda_2)} \quad (6.3)$$

So kann ein maximaler Unterschied zwischen den Parametern  $\lambda_1, \lambda_2$  von 30% mit  $\alpha \leq 0,3$  angegeben werden.

Mittels solcher Schwellenwerte können schlechte Zuordnungen schnell ausgeschlossen werden.<sup>36</sup> Aus den Kandidaten, deren Parameter unterhalb der Schwellenwerte liegen, wird das Minimum der Summe der quadrierten Abweichungen gesucht.

$$SSE = \sum \alpha^2 \quad (6.4)$$

Durch das Minimum wird sichergestellt, dass einer bestimmten Kontur aus einem Bild genau eine Kontur aus dem anderen Bild zugeordnet wird. Dies ist eine sehr restriktive Bedingung und ist anfällig für Fehlzuordnungen. Dem kann begegnet werden, indem die  $k$  besten (nach Gleichung 6.4) Konturen ermittelt werden und anschließend anhand ihrer relationalen Eigenschaften überprüft werden.

Die Relation der gefundenen Konturen kann mit den in Abschnitt 6.3.1 eingeführten Faktoren überprüft werden. Von jedem Konturpaar wird die Entfernung zu allen anderen Konturpaaren gebildet (nach Gleichung 6.2). Aus dieser Menge und der Menge der Skalierungsfaktoren wird robust<sup>37</sup> der häufigste Wert ermittelt. Diesem Vorgehen liegt die Annahme zugrunde, dass die relationalen Faktoren bei falscher Zuordnung beliebige Werte annehmen, jedoch bei richtiger Zuordnung einen bestimmten Wert. Die Methode funktioniert umso besser, je höher der Anteil der richtigen Zuordnungen an der Gesamtmenge aller Konturpaare ist. Abbildung 6.3 zeigt den Ablauf als Flussdiagramm.

### Quellenbasierte Zuordnung

Mit einem richtig zugeordneten Konturpaar als Ausgangspunkt bzw. *Quelle* können die relationalen Parameter stärker berücksichtigt werden. Ihre „Integrität“ ist durch die Quelle gewährleistet und der richtige Skalierungsfaktor muss nicht umständlich

<sup>36</sup>Diese sogenannten Kurzschluss-Operationen haben positiven Einfluss auf die Rechenzeit.

<sup>37</sup>Die numerische Genauigkeit des Rechners ist hierbei nicht relevant. Es müssen vielmehr die Unsicherheiten bei der Konturerkennung und daraus resultierend bei der Bestimmung der Fläche und des Flächenschwerpunktes berücksichtigt werden. Die Faktoren werden auf die erste Dezimalstelle gerundet.

bestimmt zu werden.

Die Schwierigkeit besteht darin, das *Konturpaar*, das als Quelle dient, zu bestimmen. Dafür gibt es drei Möglichkeiten:

1. Offline-Vorgabe einer Kontur am Modell.

Am CAD Modell wird einmalig eine Kontur vorgegeben, die bei jedem Programmaufruf zunächst in Modell- und Sensorbild gesucht wird. Diese Kontur muss „einzigartig“ sein. Das bedeutet, dass sie sich von allen anderen Konturen unterscheiden muss, sodass sie rein durch die geometrischen Parameter zuverlässig zugeordnet wird. Runde Bohrungen sind folglich schlecht geeignet.

2. Benutzereingabe.

Über eine Benutzereingabe wird ein *Konturpaar* einander zugeordnet. Dies muss für jedes aufgenommene Bild neu erfolgen.

3. Verallgemeinerung.

Die Konturen werden zunächst grob zugeordnet. Diese Paare werden anschließend der Reihe nach als Quelle benutzt. Die Quelle, mit der die meisten *Konturpaare* bestimmt werden können, wird als richtig erachtet.

Der zuverlässigste Weg ist die Benutzereingabe (1). Menschen können Objekte mühelos und instantan erkennen (JAIN et al. 1995, S. 459). Es ist gleichzeitig die Möglichkeit mit dem geringsten Grad an Autonomie/Automatisierung.

Eine einzigartige Kontur oder Kante zu bestimmen (2) erfordert ein grundlegendes Verständnis vom Ablauf und den Grenzen der geometrischen Zuordnung von Konturen. Dies muss jedoch nur einmal bei der Erzeugung des CAD Bildes erfolgen.

Die praktikabelste Lösung ist die Verallgemeinerung der quellenbasierten Zuordnung (3). Sie hat den höchsten Grad an Autonomie und funktioniert zuverlässig. Schwierigkeiten bereiten nur symmetrische Objekte, die darüber hinaus symmetrisch angeordnet sind. Der Ablauf ist schrittweise in Abbildung 6.3 dargestellt.

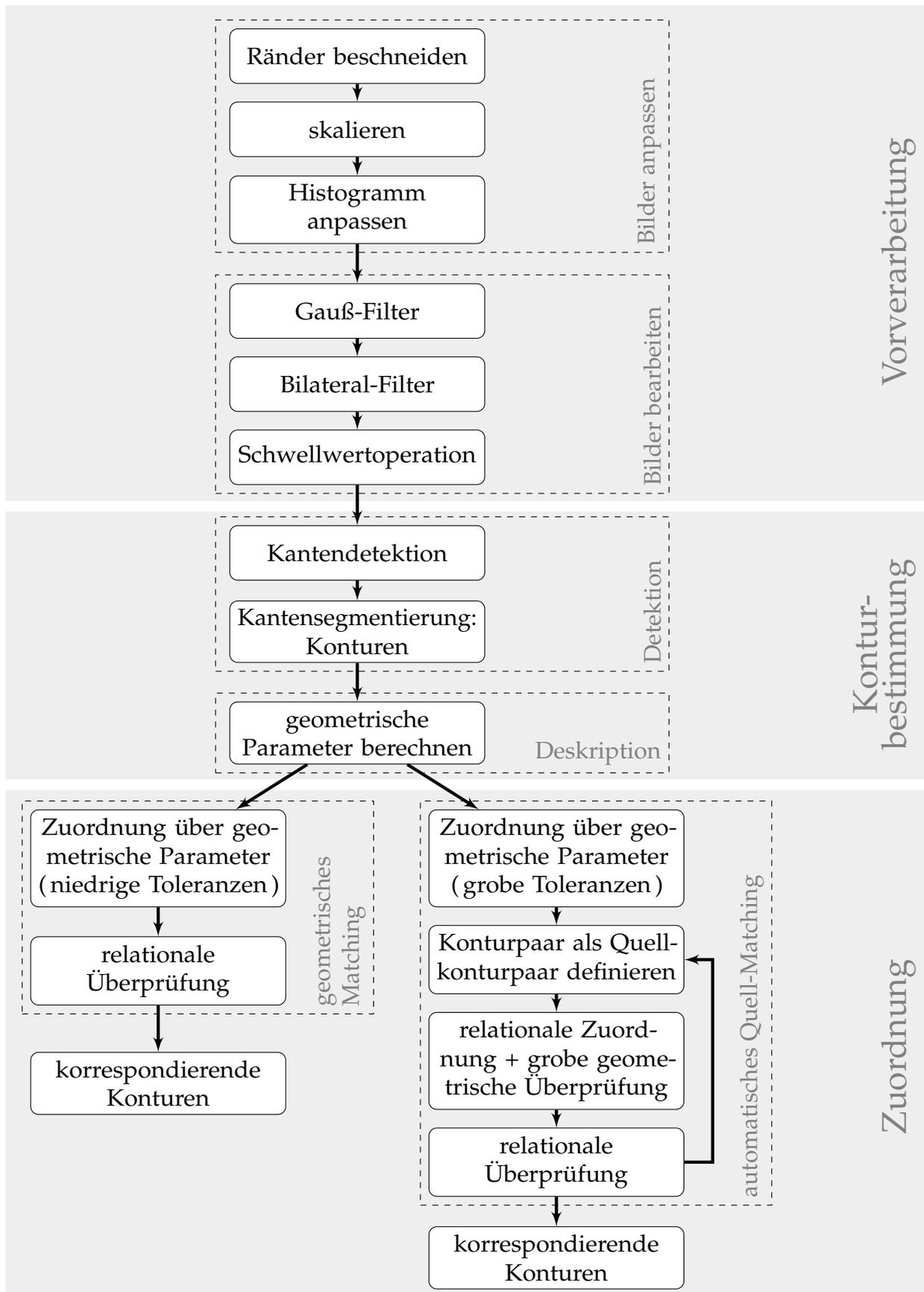


Abbildung 6.3: Ablauf der Objekterkennung mit Konturen.

### 6.4 Stereovision

In der Stereovision werden Merkmale aus zwei oder mehr Bildern einander zugeordnet und aus ihrer bildlichen Verschiebung (Disparität) Informationen über die Entfernung des realen Punktes zur Kamera gewonnen (BRADSKI & KAEHLER 2008, S. 405).

Diese Information kann als Tiefenkarte oder als Abstand einzelner Punkte/Objekte zur Kamera berechnet werden. Beide Wege wurden im Rahmen dieser Arbeit untersucht. Das Erstellen einer Tiefenkarte hat sich als äußerst schwierig herausgestellt. Der Ansatz wurde im Verlauf der Arbeit eingestellt. Deutlich praktikabler ist die Synthese der Tiefeninformation anhand ausgewählter Punkte.

#### 6.4.1 Triangulation von Konturen

Für eine Triangulation müssen gleiche Konturen in zwei Sensorbildern gefunden werden. Dies funktioniert gut mit einem Vergleich der geometrischen Parameter. Die Konturen in den Sensorbildern haben die gleiche Qualität bezüglich Bildrauschen und äußerer Einflüsse. Deshalb funktioniert hier eine geometrische Zuordnung deutlich besser als beim Vergleich mit den Idealkonturen aus dem Modell. Abbildung 7.3 und Abbildung 7.4 zeigen beispielhaft Stereobildpaare mit zugeordneten Konturen. Es ist gut zu erkennen, dass die Bohrungen auf den Sensorbildern nicht ideal kreisrund sind. Das liegt nicht an der Güte der realen Bohrungen, sondern ist dem Kamerasensor und den umfangreichen Vorverarbeitungsschritten geschuldet.

Zur Triangulation werden einzelne Bildpunkte benötigt. Dafür werden die Flächenschwerpunkte nach Gleichung 4.7 berechnet. Die Brennweite wird über die geometrische Kamerakalibrierung mit einem  $7 \times 8$  Schachbrettmuster bestimmt. Die Aufnahme geeigneter Bilder erfolgt automatisch. Dazu wird die Kamera zu den in Abbildung 6.4 dargestellten Punkten verfahren. Die Roboterpositionen für die Kalibrierung liegen auf der Oberfläche einer Kugel. Die Größe dieser Kugel wird von der Roboterzelle begrenzt, in der der Roboter frei verfahren kann. Für eine erfolgreiche Kalibrierung müssen sich die Aufnahmen genügend unterscheiden (BRADSKI & KAEHLER 2008, S. 388). Deshalb wird die gesamte Größe der Zelle verwendet. Der Sensor wird dabei ungefähr auf das Kalibrierungsobjekt – hier ein Schachbrettmuster – ausgerichtet. Die Ergebnisse sind in Abschnitt 7.2.1 dargestellt.

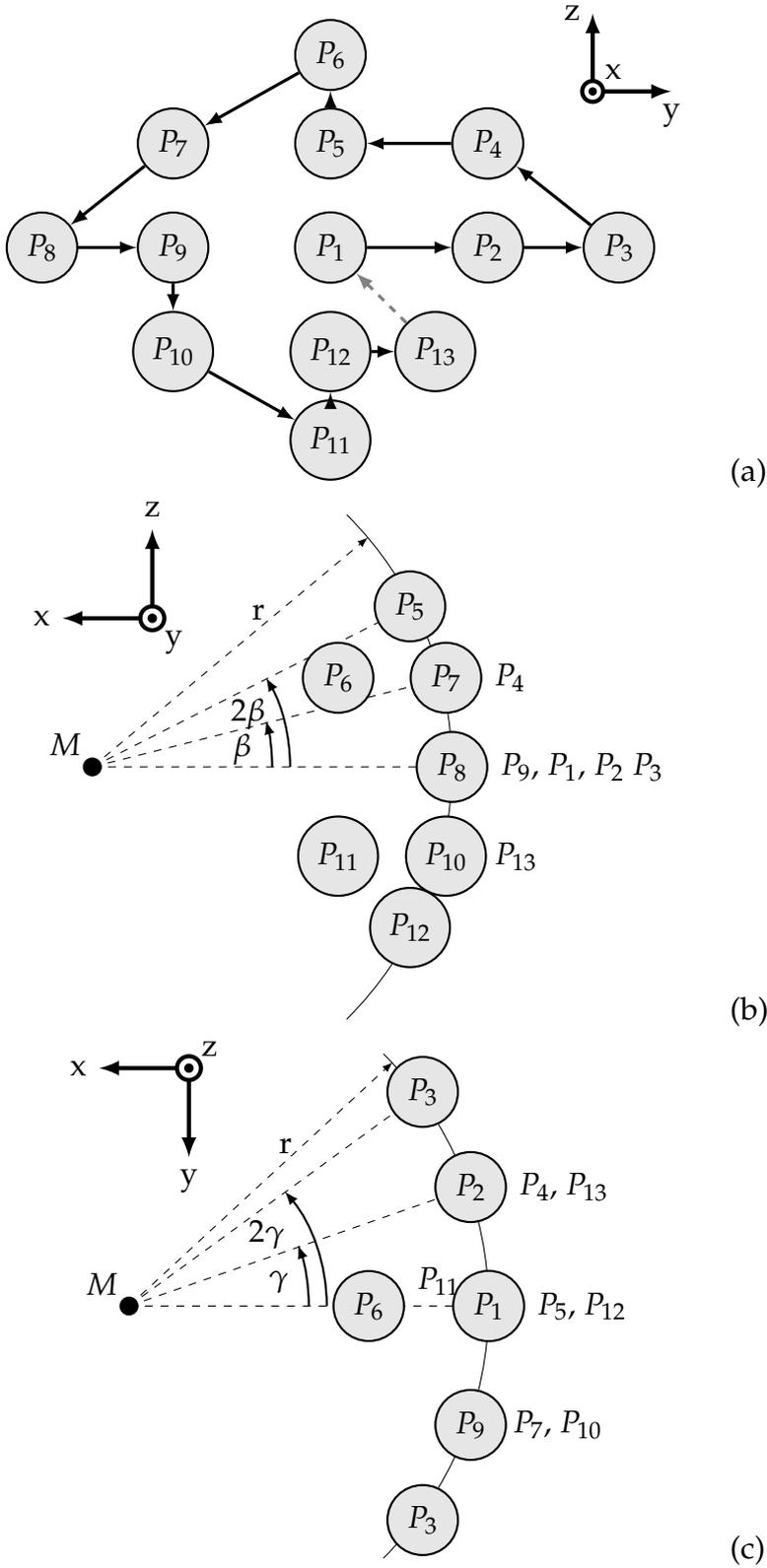


Abbildung 6.4: Punkte im Raum, die zur automatischen Kamerakalibrierung angefahren werden. Die Form der Sphäre wird über den Radius  $r$  und die Winkel  $\beta$ ,  $\gamma$  eingestellt

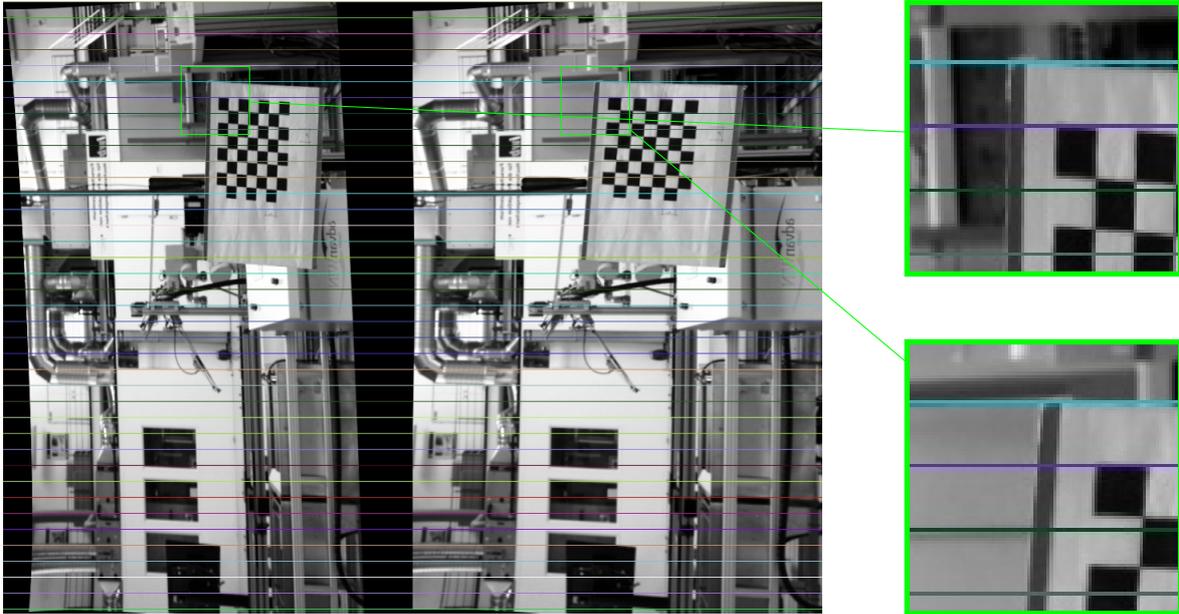


Abbildung 6.5: Rektifizierte Stereoaufnahmen ohne Stereo-Kalibrierung. (Das Kalibrierungsmuster wurde als Objekt verwendet, um definierte Punkte zu generieren.) Mit Hilfe der geschätzten Fundamentalmatrix werden die Bilder so ausgerichtet, dass Bildpunkte des gleichen realen Objekts in derselben Zeile liegen.  
Kamera: Nikon D40.

### 6.4.2 Erzeugen einer Tiefenkarte

Um eine Tiefenkarte zu berechnen, wird wie in Abschnitt 5.1 beschrieben vorgegangen:

1. Entzerrung
2. Rektifikation
3. Korrespondenzanalyse
4. Triangulation

Bei der geometrischen Kalibrierung der Sensorkamera werden auch die Verzerrungskoeffizienten bestimmt. Diese sind unabhängig von der Skalierung des Bildes (OPENCV 2015a) und müssen einmalig für eine Kamera bestimmt werden. Mit ihnen kann die Aufnahme in ihre entzerrte Form umgerechnet werden. Die entzerrte Stereoaufnahme ist in Abbildung 6.5 dargestellt. Entlang der horizontalen Linien finden sich gleiche Punkte in den Bildern. Abbildung A.4 in Anhang A.1 zeigt das gleiche Stereobild als Aufnahme mit dem Helix-Sensor von Perceptron.

Durch die Rektifikation werden zwei Aufnahmen so ausgerichtet, dass sie kollinear und komplanar sind (BRADSKI & KAEHLER 2008, S. 427). Dazu wird i. d. R. ein Stereo-

kamerasystem stereo-kalibriert (BRADSKI & KAEHLER 2008; BRAHMBHATT 2013). Dies ist im vorliegenden Fall nicht möglich. Da es sich um ein Mono-Stereovision-System handelt, müsste bei jeder Stereoaufnahme das System neu stereo-kalibriert werden. Dies erfordert das Anbringen eines Musters zur Kalibrierung in der Zelle oder am Werkstück und wird von den Randbedingungen ausgeschlossen. Aus diesem Grund muss die Fundamentalmatrix (s. Abschnitt 5.2.2) nach dem im Kapitel Abschnitt 5.2.3 beschriebenen Verfahren geschätzt werden.

Dazu werden mit dem ORB Algorithmus korrespondierende Punkte zwischen den Bildern einer Stereoaufnahme gesucht. Jedem markanten Punkt aus dem linken Bild wird der Punkt zugeordnet, dessen Deskriptor den geringsten Abstand zum Deskriptor des linken Punktes hat – unabhängig vom Betrag des Abstands. Über einen Schwellenwert  $\kappa$  werden „schlechten“ Zuordnungen aussortiert. Der ORB Deskriptor umfasst 256 binäre Einträge, der Abstand zwischen zwei Deskriptoren wird mit der Hamming-Distanz bestimmt (s. a. Abschnitt 4.3.3). Der Abstand liegt folglich im Intervall  $[0, 256]$ . Als Schwellenwert werden etwa 10% des maximal möglichen Abstandes verwendet:  $\kappa = 20$ . Die Richtung der Verschiebung gibt die Translation des Sensors bzw. des Roboters vor. Damit lassen sich die zugeordneten markanten Punkte in ihrer Richtung überprüfen. Abbildung 6.6 zeigt alle zugeordneten Punkte in (a). Diejenigen, deren Diskriptorabstand unterhalb des Schwellenwertes  $\kappa$  liegen in (b) und in (c) werden die Punkte gezeigt, deren Verbindungsvektor der Translationsrichtung des Sensors entspricht. Anschließend werden in Punkte, die zu nah beieinander liegen, aussortiert (d). Denn zur Berechnung der Fundamentalmatrix sollen die Punkte gleichmäßig über das Bild verteilt sein. Aus diesen Punktepaares wird mit dem RANSAC Algorithmus die Fundamentalmatrix bestimmt. (e) zeigt die Punktepaares, die die berechnete Fundamentalmatrix unterstützen. Die Epipolarlinien sind in (f) dargestellt. Es ist gut zu erkennen, dass die Epipolarlinien parallel und horizontal verlaufen, wie in Abschnitt 5.2.1 erwartet wurde.

Abbildung 6.7 zeigt den gleichen Vorgang für eine Stereoaufnahme des Sensors. Hier kann die Fundamentalmatrix nicht so gut geschätzt werden. Die Epipolarlinien verlaufen nur annähernd horizontal. Zur besseren Vergleichbarkeit zwischen Sensoraufnahmen und Aufnahmen mit einer anderen Kamera wurden die gleichen Stereobilder mit einer extern am Roboter befestigten Spiegelreflexkamera Nikon D40 aufgenommen.

Zwischen den ausgerichteten (rektifizierten) Bildern wird nun eine Korrespondenzanalyse durchgeführt. Hierbei werden – wie für die Schätzung der Fundamentalmatrix – korrespondierende Punkte gesucht. Da durch die nun kollinearen Aufnahmen der Suchbereich für einen korrespondierenden Punkt nur noch auf eine Zeile begrenzt

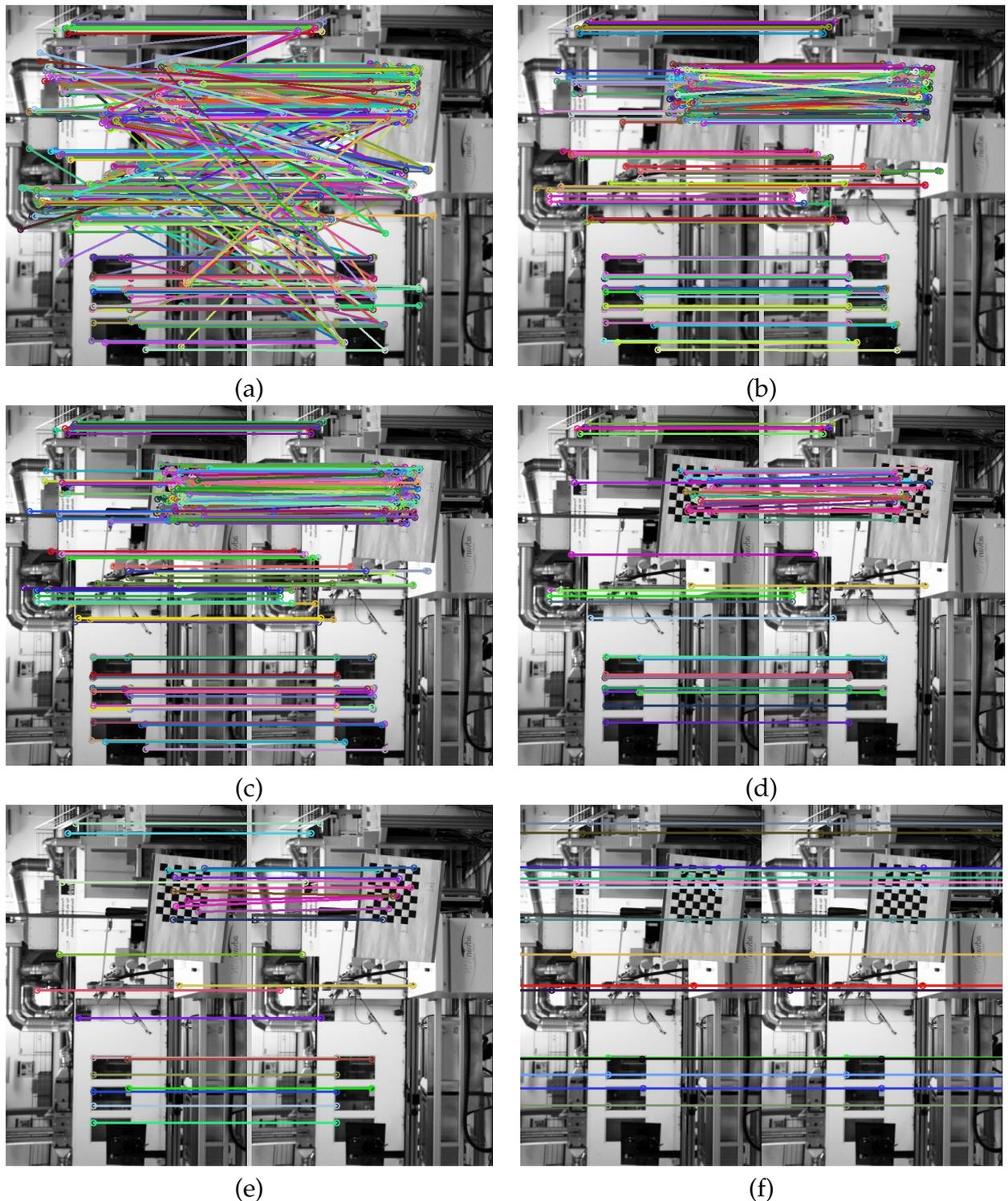


Abbildung 6.6: Bestimmung der Punktepaare zum Schätzen der Fundamentalmatrix.  
 (a) Alle Zuordnungen des ORB Algorithmus. (b) Zuordnungen, deren Deskriptorabstand kleiner als 20 (bit) ist. (c) Punktepaare mit gleicher Richtung, wie die Translation des Roboters. (d) Gleichmäßig über das Bild verteilte Punktepaare. (e) Unterstützende Punktepaare beim RANSAC Algorithmus. Mit diesem wird die Fundamentalmatrix berechnet. (f) Epipolarlinien.  
 Kamera: Nikon D40

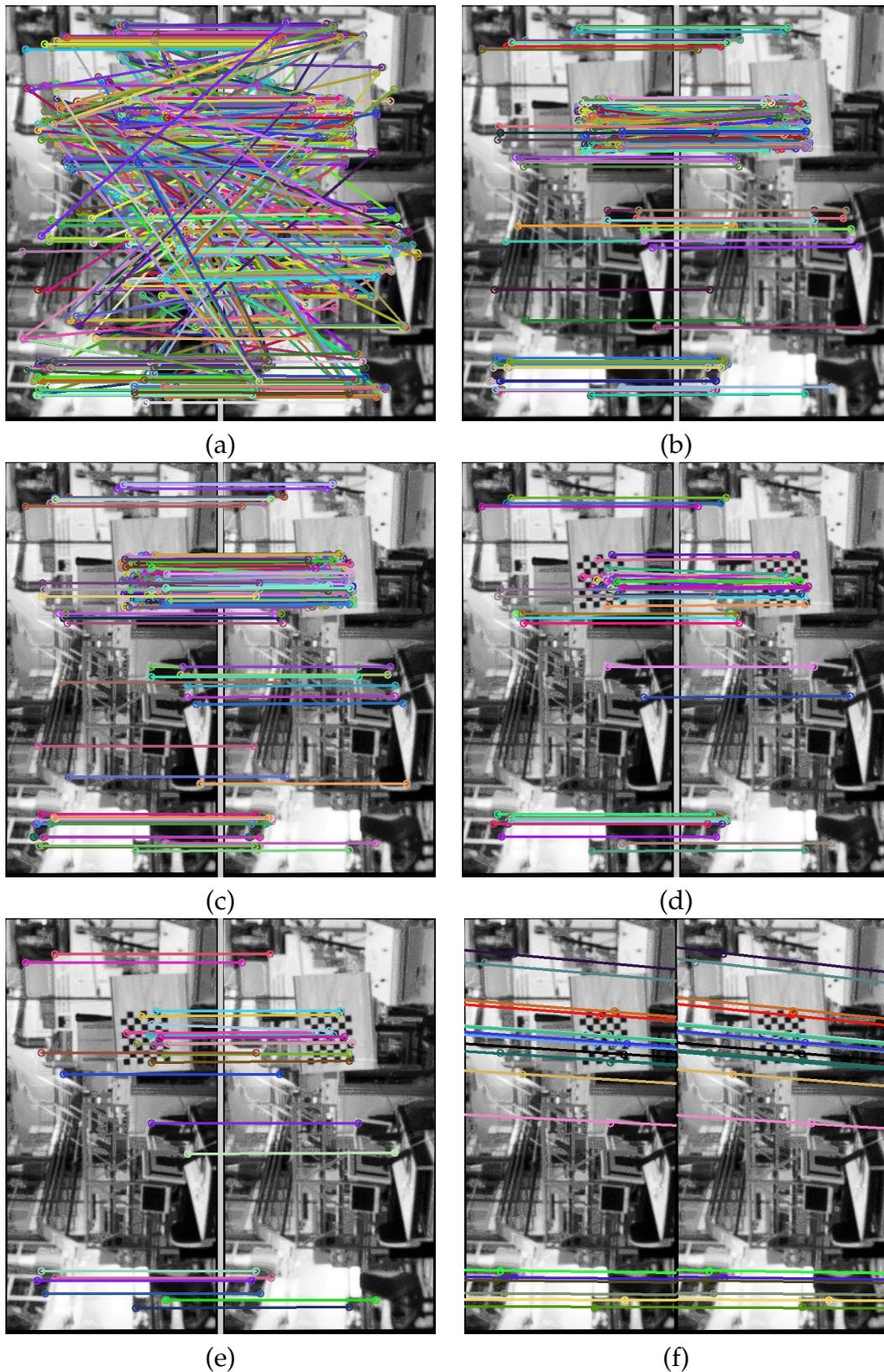


Abbildung 6.7: Bestimmung der Punktpaare zum Schätzen der Fundamentalmatrix.  
 (a) Alle Zuordnungen. (b) Deskriptorabstand  $< 20$  (bit). (c) In Translationsrichtung. (d) Gleichmäßig verteilt. (e) Unterstützen RANSAC. (f) Epipolarlinien.

Kamera: Helix-Sensor

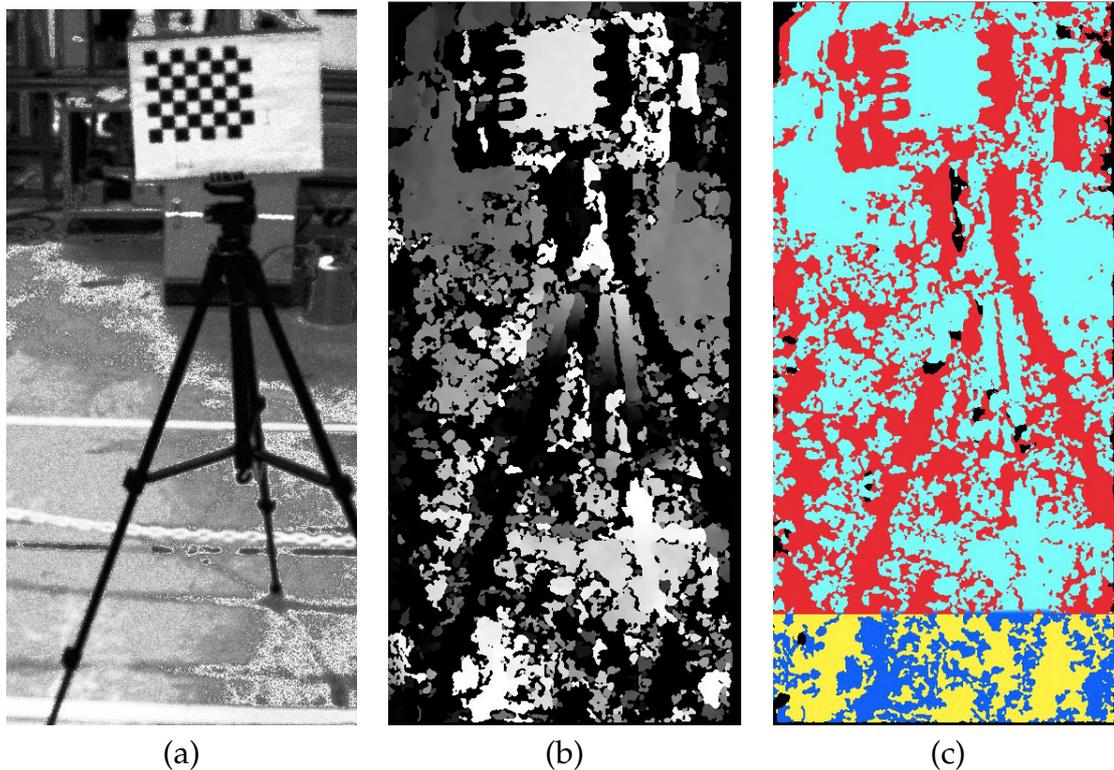


Abbildung 6.8: Berechnung einer Tiefenkarte ohne stereo-kalibrierte Kamera mit dem SBGM Algorithmus. (a) angepasste Aufnahme der linken Kamera. (b) „disparity map“ grafische Darstellung der Disparitäten ( Unterschiede ) zwischen den Stereobildern; (c) grafische Darstellung der Tiefenkarte.

ist, müssen diese Punkte nur bezogen auf die Zeile „markant“ sein. Dadurch kann eine deutlich höhere Anzahl an Korrespondenzen ermittelt werden. Das Funktionsprinzip der Korrespondenzanalyse ist in Abschnitt 5.3 beschrieben. Das Ergebnis der Korrespondenzanalyse ist eine Karte bzw. ein Bild, das die relative Verschiebung der korrespondierenden Pixel kodiert. Diese sogenannte „Disparity-Map“ wird mittels Gleichung 5.8 in eine Tiefenkarte überführt. Diese kann mit unterschiedlichen Farben bildlich dargestellt werden. Abbildung 6.8 zeigt eine Disparitätskarte (b) und die daraus berechnete Tiefenkarte (c).

Das Erstellen einer Tiefenkarte, die auf den im Laufe dieser Arbeit erzeugten Stereoaufnahmen basiert, ist sehr schwierig. Die Entzerrung der Bilder funktioniert nur unbefriedigend und die anschließende Rektifikation scheitert gänzlich. Der begrenzte zeitliche Umfang dieser Arbeit ließ eine weitere Untersuchung nicht zu. Detaillierte Untersuchungen dazu und eine verbesserte Implementierung scheinen ein lohnenswerter Bereich für weitere Arbeiten zu sein.

## 7 Ergebnisse

Dieses Kapitel wertet das beschriebene System aus und diskutiert die Ergebnisse. In Kapitel 8 wird anschließend ein Fazit gezogen.

### 7.1 Objekterkennung

In diesem Abschnitt werden die verschiedenen, im vorherigen Kapitel beschriebenen Methoden zur Zuordnung von Konturen verglichen und bewertet. Um ein Objekt in einer Aufnahme zu „erkennen“, muss ein Vergleichsbild vorhanden sein. Dies wird in Abschnitt 7.1.3 betrachtet.

Zur besseren Vergleichbarkeit wurden die Ergebnisse immer an den 18 gleichen Stereoaufnahmen ermittelt. Die Unterscheidung in „richtige“ und „falsche“ Zuordnungen erfolgte manuell.

#### 7.1.1 Geometrische Korrespondenzanalyse

In diesem Abschnitt werden korrespondierende Konturen zwischen zwei Bildern gesucht. Diese sind nicht limitiert auf Merkmale eines bestimmten Objekts (z. B. einer Autotür). Korrespondenzen werden deshalb nur „detektiert“, aber nicht „erkannt“.

Die rein geometrische Zuordnung von Konturen ist unzuverlässig. Im Durchschnitt werden 5,2 korrespondierende Konturen detektiert. Davon sind aber fast 30 % falsche Zuordnungen. Abbildung 7.1 zeigt dies aufgelöst nach den einzelnen Stereoaufnahmen.

Es gibt keine Möglichkeit, über die Konturform die Zuordnungen zu verifizieren. Insbesondere können keine falschen Korrespondenzen herausgefiltert werden und beeinflussen deshalb massiv die Entfernungsmessung.

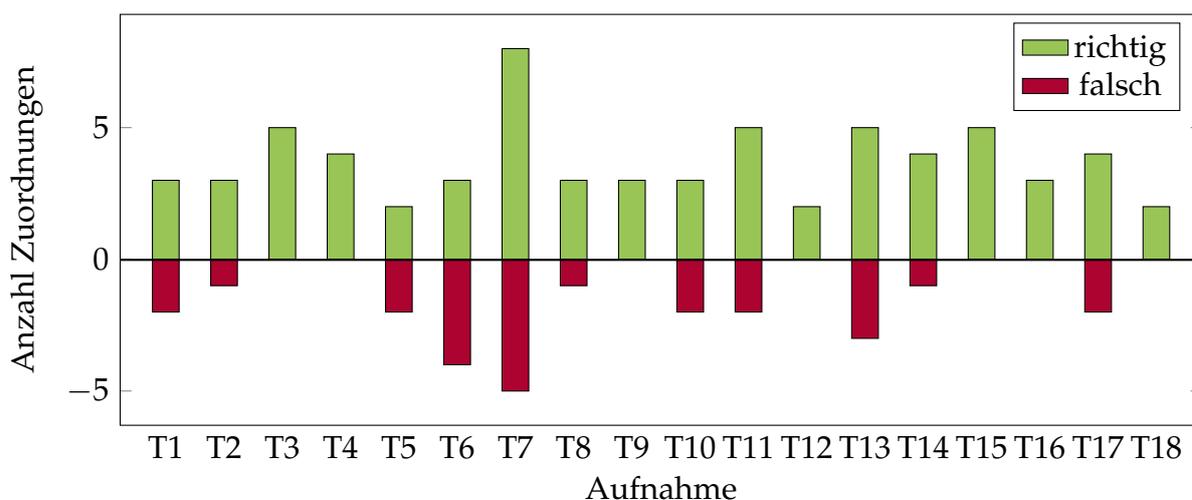


Abbildung 7.1: Ergebnisse der geometrischen Zuordnung. Es wurden Konturpaare in Stereoaufnahmen gesucht.

Wird anschließend die Lage der Konturen zueinander in beiden Bildern überprüft, so können falsche Zuordnungen herausgefiltert werden. Dieses Vorgehen wurde in Abschnitt 6.3.2 „relationale Überprüfung“ genannt. Ihr Potential wird in Abbildung 7.3 an einem Beispiel gezeigt. Dadurch werden zu 97 % richtige Korrespondenzen in den Stereoaufnahmen detektiert. Die durchschnittliche Anzahl richtig zugeordneter Konturen sinkt von 3,72 (rein geometrische Zuordnung) nur wenig auf 3,5 (geometrische Zuordnung mit relationaler Überprüfung). In Abbildung 7.2 sind die Anzahl der richtigen und falschen Zuordnungen nach Testbildern aufgeschlüsselt dargestellt.

Lediglich in Stereoaufnahme T10 werden mit der relationalen Überprüfung falsche Korrespondenzen detektiert. Falsche Zuordnungen verfälschen eine weitere Verarbeitung besonders stark. Durch sie wird die Entfernungsschätzung falsch und kann im schlimmsten Fall zu einer Kollision des Roboters mit einem Objekt führen. Falsche Zuordnungen sind deshalb ein nicht zu tolerierendes Sicherheitsrisiko für das System.

### 7.1.2 Quellenbasierte Korrespondenzanalyse

Der Startpunkt der quellenbasierten Zuordnung von Konturen ist ein korrespondierendes Konturpaar (die Quelle). Anhand dieser werden Skalierung und Orientierung des rechten Bildes relativ zum linken Bild bestimmt. Die anderen, durch die geometrische Korrespondenzanalyse grob zugeordneten Konturpaare werden nun anhand ihrer Lage (Entfernung und Richtung) zur Quellkontur überprüft. Dieser Prozess kann automa-

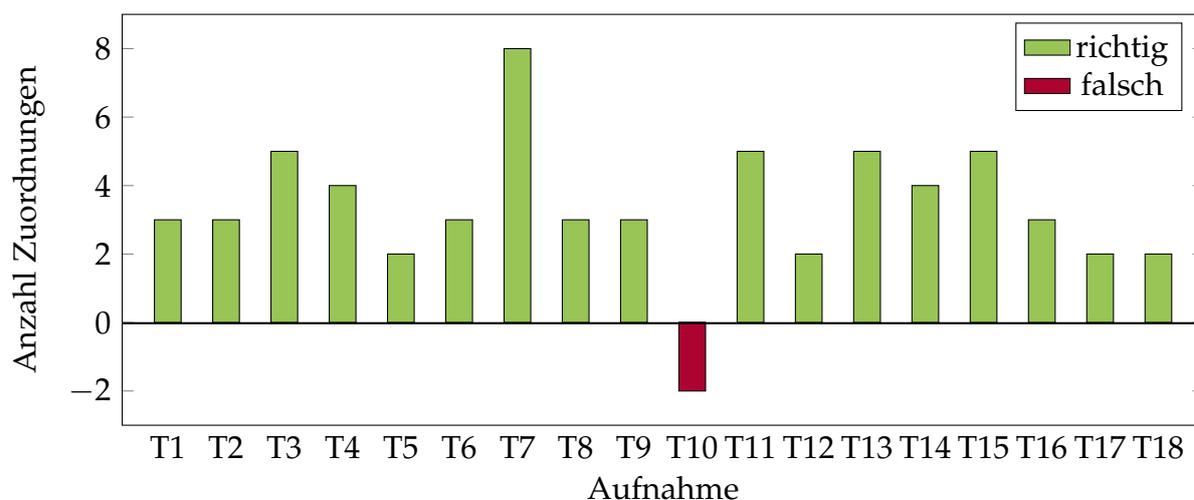


Abbildung 7.2: Ergebnisse der geometrischen Zuordnung mit anschließender relationaler Überprüfung. Es wurden Konturpaare in Stereoaufnahmen gesucht.

tisiert werden, sodass auch die Quelle selbstständig bestimmt wird. Abschnitt 6.3.2 erläutert dies detailliert.

Abbildung 7.4 zeigt vier Beispiele für die quellenbasierte Zuordnung. In (c) (Testbild T20) kann man gut erkennen, dass auch Konturen, die sich bis zu einem gewissen Grad in ihrer Geometrie unterscheiden, einander zugeordnet werden. Durch die Orientierung und Skalierung des Quellkonturpaares kann der Suchbereich für die anderen Konturpaare auf kleine Bildregionen eingegrenzt werden. In diesem Suchbereich können die geometrischen Parameter verhältnismäßig grob verglichen werden.

In Abbildung 7.4 (d) können nur zwei Konturpaare detektiert werden. Dies ist kritisch für ein eindeutiges „Erkennen“ des Objekts. Es kann nur eine grobe relationale Überprüfung durchgeführt werden. Die Größe der detektierten Konturen relativiert diese Einschränkung jedoch wieder.

### 7.1.3 Modellbasierte Objekterkennung

Bei der modellbasierten Objekterkennung wird versucht, Merkmale eines Objekts (Autotür), das als Modell (CAD-Rendering) gegeben ist, in einer Kameraaufnahme zu detektieren. Kann eine gewisse Anzahl Merkmale detektiert werden, so gilt das Objekt als erkannt.

Über die geometrische Korrespondenzanalyse werden Kandidaten für die quellenbasierte Zuordnung bestimmt. Wird nur ein Konturpaar bzw. eine Quelle erkannt, so kann nicht mit Sicherheit gesagt werden, ob es eine richtige oder falsche Zuordnung ist. Das

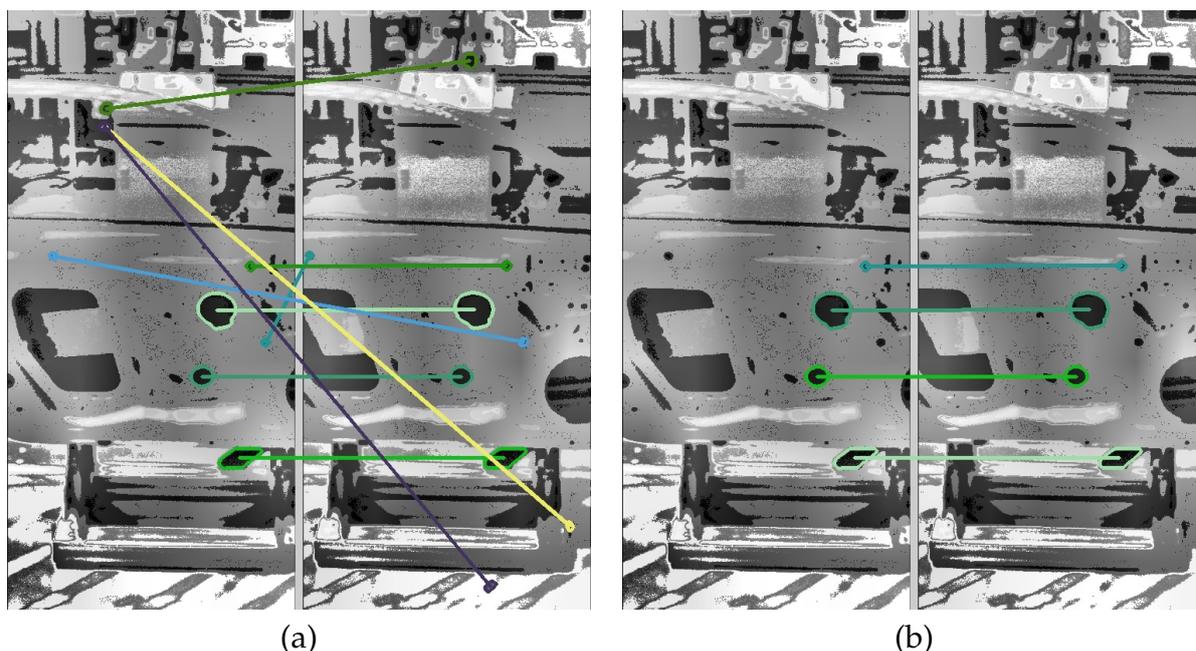


Abbildung 7.3: Geometrische Zuordnung des Stereobildes T6. (a) rein geometrische Zuordnung; (b) geometrische Zuordnung mit relationaler Überprüfung.

Objekt gilt als nicht erkannt. Werden zwei Konturpaare detektiert, so konnten diese relational überprüft werden. Die Wahrscheinlichkeit, dass dies Konturen des Objekts sind, ist deutlich höher. Insbesondere bei kleinen Konturen<sup>38</sup> können Zuordnungen auch falsch sein. Ab drei zugeordneten Konturpaaren gilt das Objekt als „erkannt“.

Die modellbasierte Objekterkennung funktioniert ähnlich wie die quellenbasierte Korrespondenzanalyse. Anstatt zweier Kameraaufnahmen wird ein Modellbild mit einer Kameraaufnahme verglichen. Die in Abschnitt 6.3.1 beschriebenen Toleranzen müssen dabei großzügiger gewählt werden als beim Vergleich zweier Sensorbilder. Die Bohrungen im Modellbild sind perfekt rund; die im Sensorbild hingegen, durch die starke Glättung und die Aufnahmequalität des Sensors, leicht „ausgefranst“. Das zeigt Abbildung 7.5 an einem Beispiel.

<sup>38</sup> „klein“ ist hier auf die Fläche und dem Umfang der Kontur bezogen. Bei kleinen Konturen sind geometrische Parameter, wie „die kürzeste Strecke zwischen zwei Punkten bezogen auf den mittleren Abstand aller Punkte einer Kontur“, oder „der kleinste, die Kontur einschließende Kreis“, nicht so aussagekräftig wie bei größeren Konturen. Kleinere Konturen haben tendenziell weniger „Beulen“ oder Knicke und sie werden vom Kamerasensor schlechter aufgelöst. Vorverarbeitungsschritte wie Glätten durch den Gauß- und Bilateral-Filter haben ebenfalls höheren Einfluss auf kleinere Konturen, da sie besonders den Rand (die Kanten) von Konturen Glätten und dies erheblichen Einfluss auf bspw. Umfang und Fläche der Kontur hat – bezogen auf die ursprünglichen Werte.

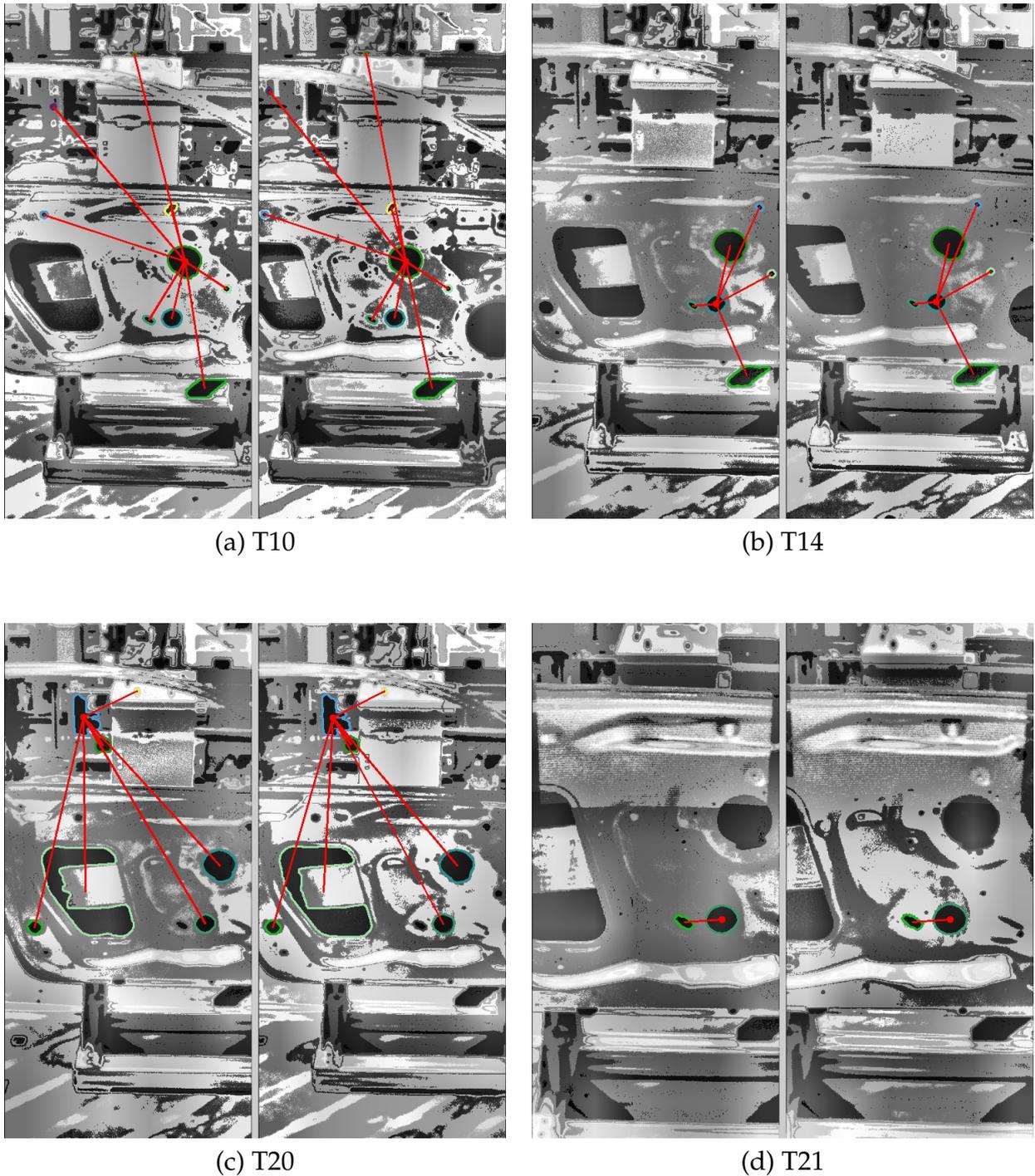


Abbildung 7.4: Quellenbasierte Zuordnungen in einer Stereoaufnahme. Die roten Linien verbinden die automatisch bestimmte Quellkontur mit den anderen Konturen im linken/rechten Bild. Je nach Aufnahme werden unterschiedliche Konturpaare detektiert. Die detektierten Konturpaare sind nicht auf das Objekt (die Autoteile) beschränkt. Es werden auch Konturpaare detektiert, deren Geometrie nicht exakt übereinstimmen, die jedoch – relativ zur Quelle – richtig liegen. Dies macht das Verfahren auch bei Aufnahmen von schlechter Qualität robust.

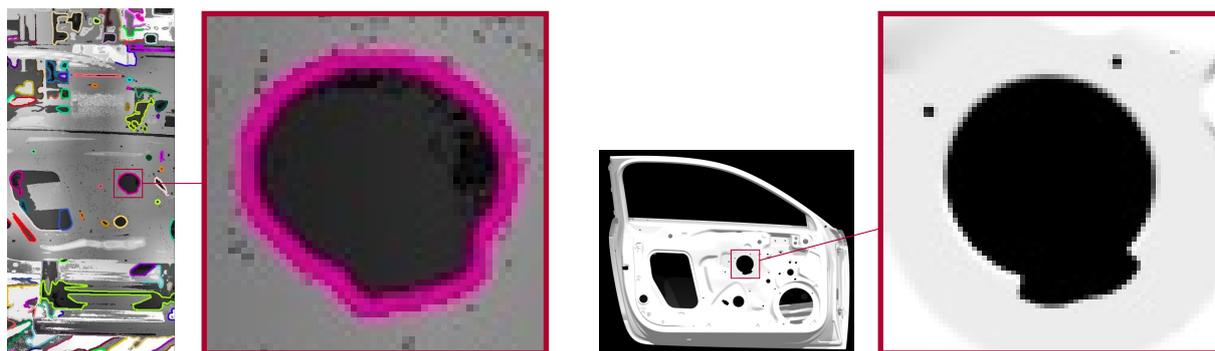


Abbildung 7.5: Die Qualität der geometrischen Form einer Kontur: Unterschied zwischen regradem Modell und detektierter Kontur in der Sensoraufnahme.

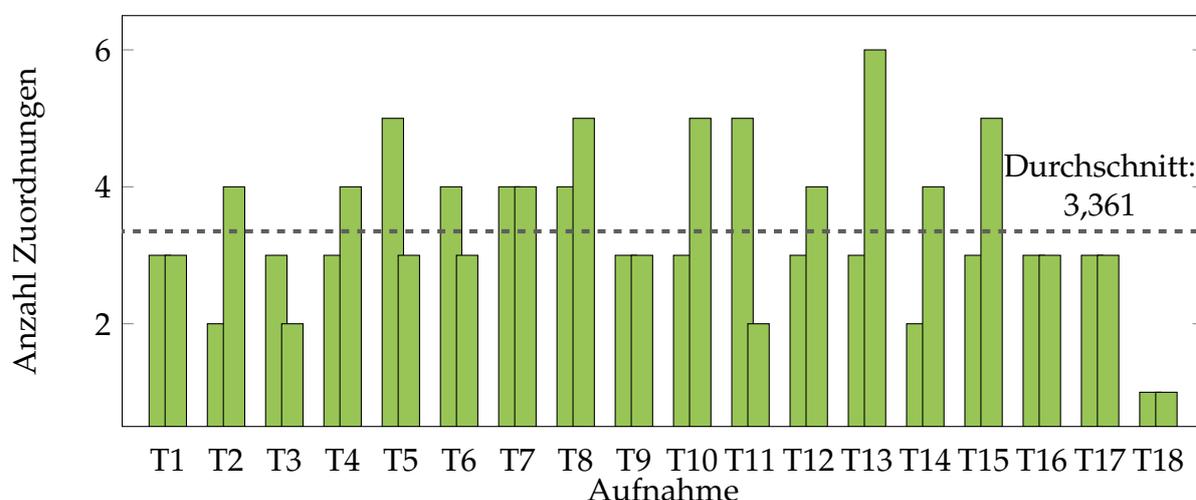


Abbildung 7.6: Ergebnisse der modellbasierten Zuordnung mit automatisch bestimmter Quellkontur. Die gruppierten Balken zeigen jeweils das Ergebnis für die linke und rechte Stereoaufnahme der Testbilder T1 bis T18.

Abbildung 7.6 zeigt die Ergebnisse der automatischen quellenbasierten Korrespondenzanalyse zwischen Modellbild und Sensorbild. Dabei wurden alle Einzelbilder der 18 Stereoaufnahmen analysiert. Im Durchschnitt werden 3,36 Konturen detektiert. Alle Konturen werden zwar korrekt zugeordnet, jedoch entspricht die Anzahl der Zuordnungen nicht immer dem eben beschriebenen Gütekriterium. Insbesondere die Aufnahmen aus T18 sind kritische, da in ihnen jeweils nur eine Kontur detektiert wird. Die Aufnahme wurde aus sehr kurzer Distanz zum Objekt gemacht. Deshalb sind nur wenige Konturen vorhanden. Bei kurzen Distanzen ist der Basisabstand (die Strecke, um die der Roboter zwischen den Aufnahmen verfahren wird) besonders wichtig, da er den Überlappungsbereich der Aufnahmen maßgeblich beeinflusst.

Die Aufnahmen T2<sub>l</sub>, T3<sub>r</sub>, T11<sub>r</sub> und T14<sub>l</sub> wurden korrekt zugeordnet, ihre Zuverlässigkeit ist mit zwei erkannten Konturen jedoch nicht hoch. Hier muss die Größe der Konturen

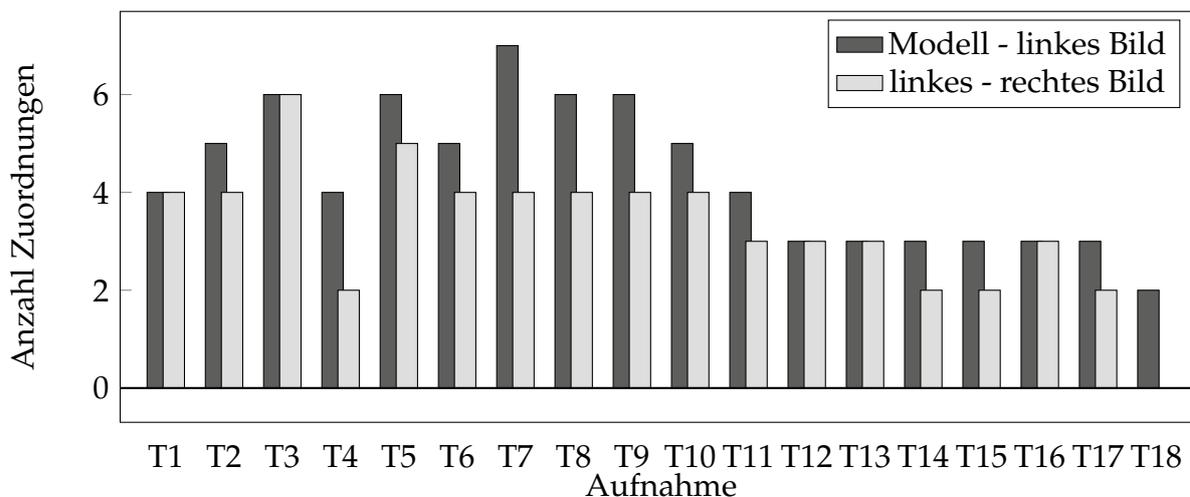


Abbildung 7.7: Ergebnisse der Modell basierten Zuordnung von Stereobildern. Grau: Zuordnungen zwischen Modell und linker Stereoaufnahme; Grün: Zuordnungen zwischen den Zuordnungen aus Modell und linkem Bild und der rechten Stereoaufnahme.

für das „Erkennen“ berücksichtigt werden oder eine externe Überprüfung durch einen Werker vorgenommen werden.

Die modellbasierte Objekterkennung für Stereobilder erfolgt in zwei Schritten:

1. Modellbasierte Objekterkennung in der linken Aufnahme des Stereobildes.
2. Quellenbasierte Korrespondenzanalyse mit den Modell-Konturen von der linken Aufnahme in der rechten Aufnahme.

Im rechten Bild werden nur die zuvor im linken Bild gefundenen Konturen des Modells gesucht. Dies stellt sicher, dass im nachfolgenden Schritt, der Triangulation (s. a. Abschnitt 7.2.2), nur für Merkmale des Objekts die Entfernung bestimmt wird. Es verringert auch die Möglichkeit, falsche Korrespondenzen im Stereobild zu erkennen, durch die Eingrenzung auf wenige, zuverlässige Konturen. Abbildung 7.7 zeigt die Ergebnisse der Testreihe für die modellbasierte Stereo-Zuordnung. Bis auf T18 werden immer richtige Objekt-Konturpaare in den Stereobildern gefunden. Obwohl zwei Konturpaare zwischen Modell und linkem Stereobild T18, detektiert werden, werden keine Korrespondenzen zwischen diesen und dem rechten Stereobild gefunden. Positiv zu bewerten ist, dass keine falschen Konturen detektiert werden. Das Objekt wird in der rechten Aufnahme schlicht nicht „erkannt“.

Abbildung 7.9 zeigt die einzelnen Schritte der modellbasierten Korrespondenzanalyse anhand der Stereo-Testaufnahme T9. Eine der Korrespondenzen zwischen Modell und

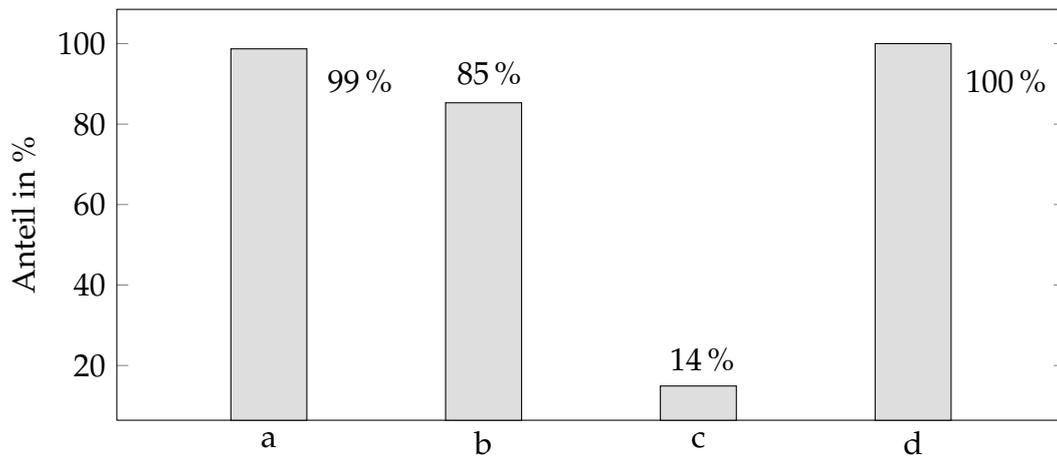


Abbildung 7.8: Jeweils Anteil an der Gesamtmenge der Stereoaufnahmen. a: Erkanntes Modell im linken Bild. b: Modellkonturen aus dem linken Bild im rechten Bild erkannt. c: Falsch zugeordnete Konturen. d: Richtig nicht erkanntes Objekt (in leeren Bildern)

linker Stereoaufnahme kann in der rechten Stereoaufnahme nicht gefunden werden. Die Kontur ist schlicht durch das Verfahren der Kamera nicht mehr im Aufnahmebereich.

Abbildung 7.8 stellt die Erkennungsrate aus 77 Stereoaufnahmen aus unterschiedlichen Entfernungen<sup>39</sup> grafisch dar. Es wurden auch mehrere Aufnahmen ohne Objekt gemacht. In allen Fällen wurde auch kein Objekt erkannt. Es werden also keine falschen Objekte, bspw. aus dem Hintergrund, erkannt. Dies ist wichtig, um die Zuverlässigkeit des Algorithmus' zu bewerten. In 1 % der Fälle wird ein vorhandenes Objekt nicht erkannt. Allerdings werden in 14 % der Fälle falsche Modellkonturen dem linken Bild zugeordnet. Daher die Diskrepanz zwischen Abbildung 7.8 (a) und (b).

<sup>39</sup>Die Entfernung lag zwischen 1,4 m und 1,7 m. Die Qualität dieser Messungen ist in Abbildung 7.15 dargestellt.

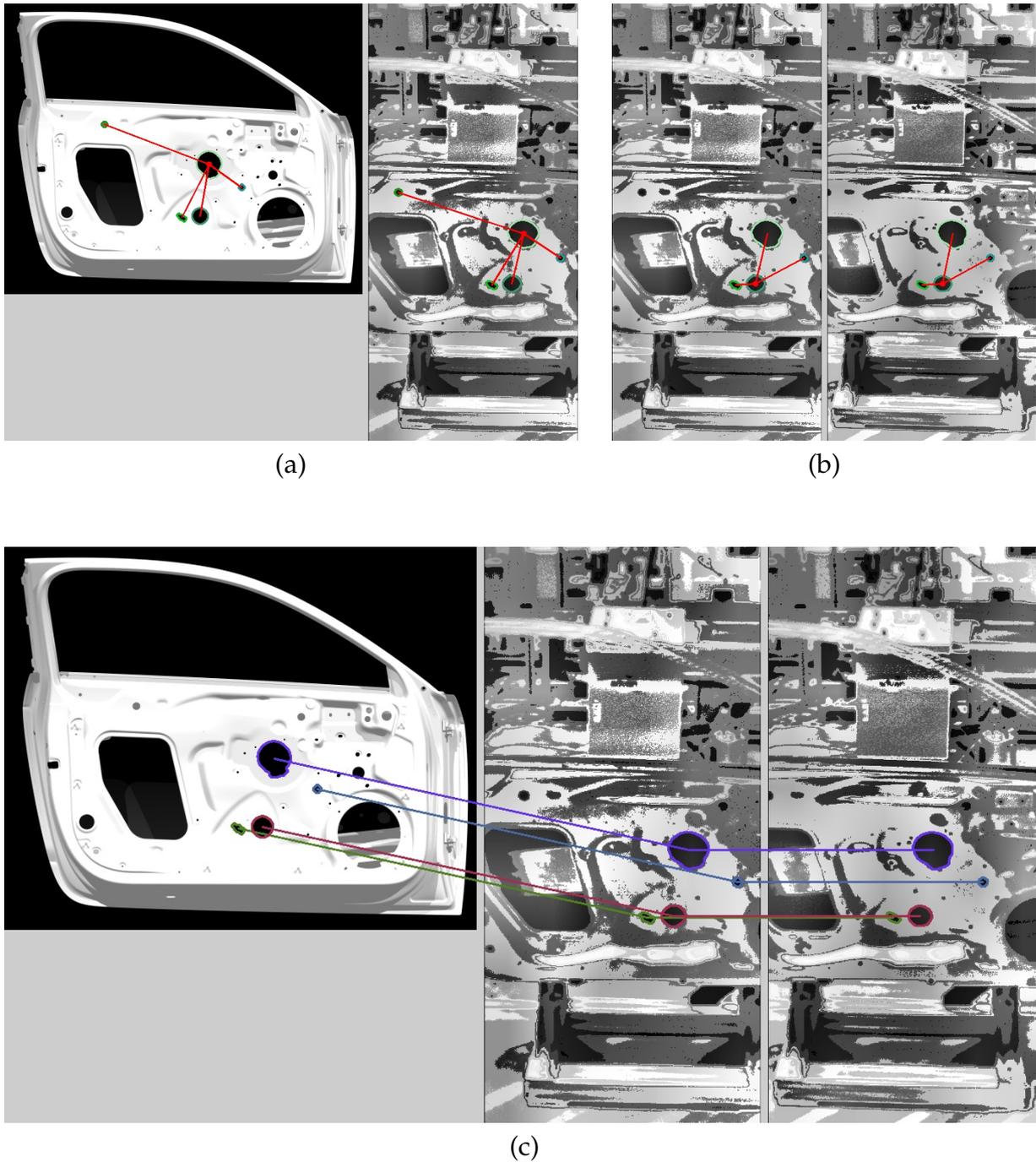


Abbildung 7.9: Modellbasierte Objekterkennung am Testbild T9. (a) Zuordnung von Konturen zwischen Modellbild und linkem Stereobild. (b) Zuordnung von Konturen zwischen den im linken Stereobild gefundenen Modellkonturen und den Konturen des rechten Stereobildes. (c) Markierung der (b) detektierten Konturen in allen Bildern.

## 7.2 Stereovision

Zur Bestimmung der Entfernung eines realen Punktes aus einer Stereoaufnahme sind nach Gleichung 5.8 die Brennweite  $f$ , der Basisabstand  $T$  und die Disparität  $d$  wichtig:

$$Z = \frac{f T}{d} \quad (5.8)$$

Der Basisabstand ist aus dem Fahrweg des Roboters bekannt. Die Disparität wird über die Korrespondenzanalyse bestimmt. Die Brennweite wird über die geometrische Kamerakalibrierung (vgl. Abschnitt 3.5) bestimmt.

Im nächsten Abschnitt werden zunächst die Ergebnisse der Kalibrierung beschrieben und anschließend in Abschnitt 7.2.2 die Entfernungsmessung analysiert.

### 7.2.1 Geometrische Kamerakalibrierung

Zur Bestimmung der intrinsischen Kameraparameter wird ein Schachbrettmuster verwendet. Dieses wird aus den 13 in Abschnitt 6.4.1 beschriebenen Positionen fotografiert und anschließend analysiert. Die daraus bestimmte intrinsische Kameramatrix (vgl. Abschnitt 3.5) ergibt sich zu:

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1096,0 \pm 84 & 0 & 180,7 \pm 25 \\ 0 & 1099,4 \pm 103 & 386,7 \pm 14 \\ 0 & 0 & 1 \end{bmatrix}$$

Die Einheit der Brennweite ( $f_x, f_y$ ) und des Optischen Zentrums ( $c_x, c_y$ ) sind Pixel (BRADSKI & KAEHLER 2008, S. 373).<sup>40</sup> Die Streuung bei den Brennweiten ist bei 18 Messungen mit 10 % hoch. Dies ist der Aufnahmequalität geschuldet. Die Ecken des Kalibrierungsmusters werden nicht immer an der richtigen Stelle erkannt. Wenn die Ecken nicht auf einer Geraden liegen, wie es in Abbildung 7.11 der Fall ist, ist dies gut zu

<sup>40</sup>Die physikalische Brennweite  $F$  kann mit einem Skalierungsfaktor  $s$  in der Einheit  $1 \text{ px/mm}$  bestimmt werden:  $f = F \cdot s$ . Aus den Abmessungen des Kamerasensors und seiner Auflösung kann  $s$  berechnet werden. Es ist jedoch nicht möglich  $F$ , oder  $s$  durch ein Kalibrierungsverfahren direkt zu bestimmen, sondern nur die Kombination  $F \cdot s = f$  (BRADSKI & KAEHLER 2008, S. 373).

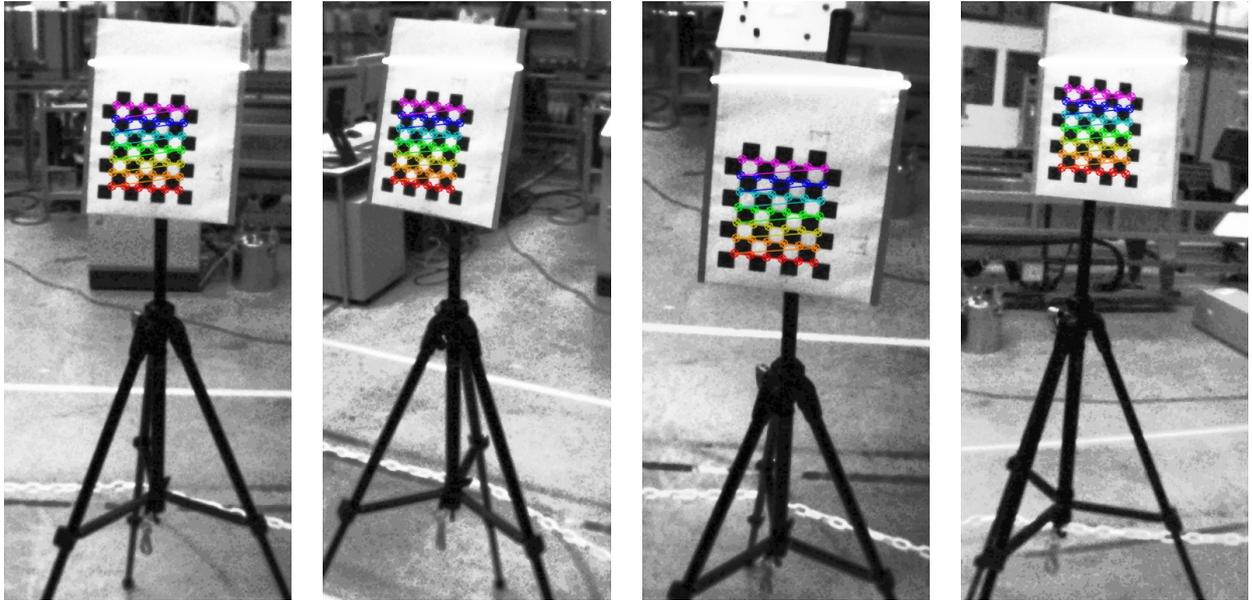


Abbildung 7.10: Aufnahmen der geometrischen Kamerakalibrierung mit erkanntem Schachbrettmuster bzw. dessen Ecken. Die Bilder sind aus den Posen 1, 2, 4, 8 aufgenommen worden (vgl. Abbildung 6.4), bei einer Entfernung von 1,2 m

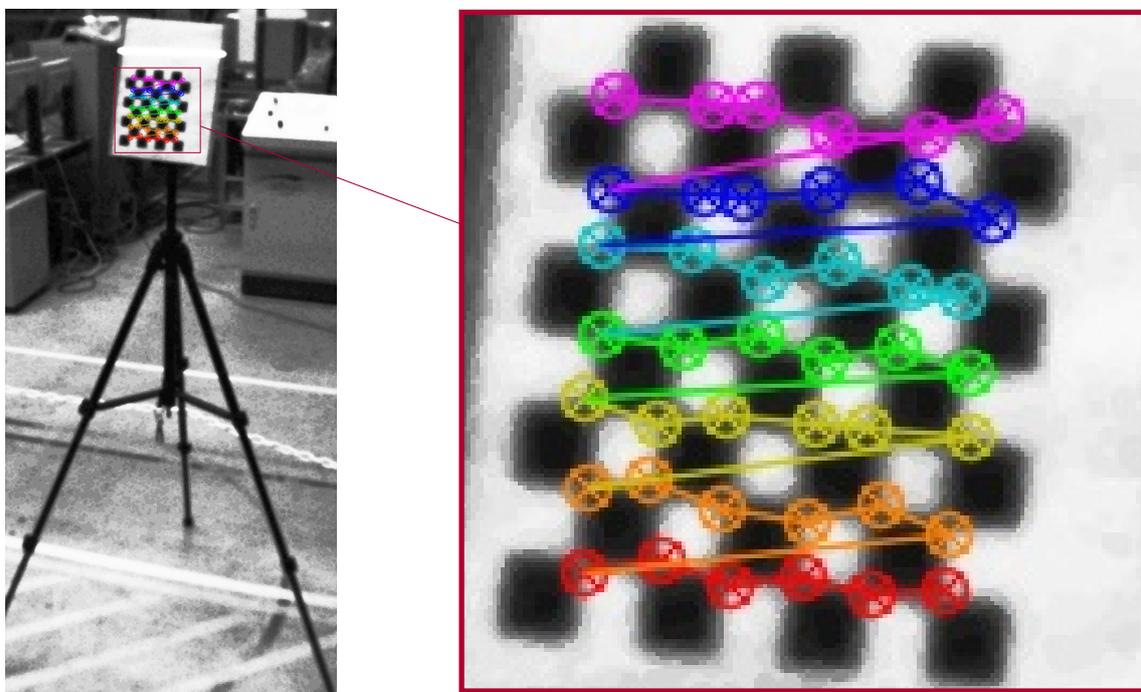


Abbildung 7.11: Ungenau detektiertes Kalibrierungsmuster.

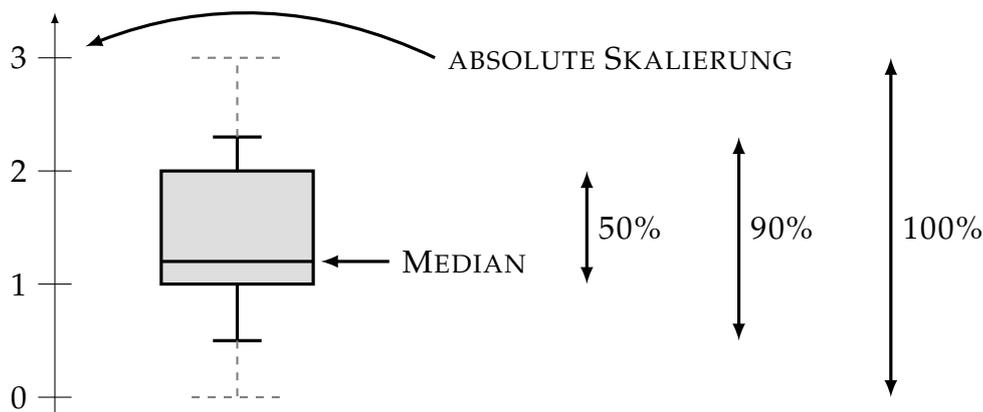


Abbildung 7.12: Definition eines Boxplots. Innerhalb der Box befinden sich 50 % aller Werte, innerhalb der „Wisker“ (durch die schwarzen Linien begrenzter Bereich) 90 % aller Werte. Die Extrema sind durch graue, gestrichelte Wisker gekennzeichnet. Der Median wird durch einen langen, waagerechten Strich angezeigt. Sein Wert ist genau die Mitte aller Messwerte.

erkennen. Die intrinsischen Parameter sind in Abbildung 7.13 als Boxplot<sup>41</sup> dargestellt. Die genauen Werte dazu sind in Tabelle A.1 zusammengefasst. Abbildung 7.10 zeigt vier Aufnahmen mit detektiertem Kalibrierungsmuster.

Die Brennweite  $f_x$  streut um +9 % und -11 % um den Mittelwert. Die Brennweite  $f_y$  um +11 % und -12 %. Eine solch hohe Streuung beeinflusst die Qualität der Entfernungsmessung massiv. Ein entsprechend großer Sicherheitsfaktor muss auf eine Entfernungsmessung aufgeschlagen werden.

Die große Streuung liegt an der Bildqualität. Wie in Abbildung 7.11 zu sehen, liegen die Verbindungslinien der Ecken des Schachbrettmusters liegen nicht immer auf einer Geraden. Die Sensoraufnahmen sind teilweise von so schlechter Qualität, dass der Algorithmus das Muster nicht erkennt. Allgemein sind die Aufnahmen aus kürzerer Entfernung schärfer und das Muster wird bis 1,2 m Abstand zuverlässig erkannt. Der Algorithmus wertet das Kalibrierungsmuster auf Subpixel-Genauigkeit aus (s. a. Abschnitt 3.5) (BRADSKI & KAEHLER 2008, S. 383). Abweichungen im detektierten Kalibrierungsmuster, die mit bloßem Auge zu erkennen sind, beeinflussen daher die Qualität des Ergebnisses stark.

<sup>41</sup>Ein Boxplot stellt die Streuung eines Wertes dar. Die Boxen stellen den Wertebereich, in dem ein bestimmter Prozentsatz an Werten liegt dar. Die Quantile  $Q_x\%$  geben an, unterhalb welches Werts  $x\%$  der Messwerte liegen. Die waagerechten Linien in einem Boxplot geben diese Quantile an. Hier 5 %, 25 %, 50 %, 75 % und 95 %. Das 50 % Quantil  $Q_{50\%}$  entspricht dem Median. Abbildung 7.12 erklärt das Prinzip eines Boxplots und definiert gleichzeitig die genauen Bereiche der hier verwendeten Boxplots. In Anhang A.2 werden die exakten Werte der einzelnen Boxplots angegeben.

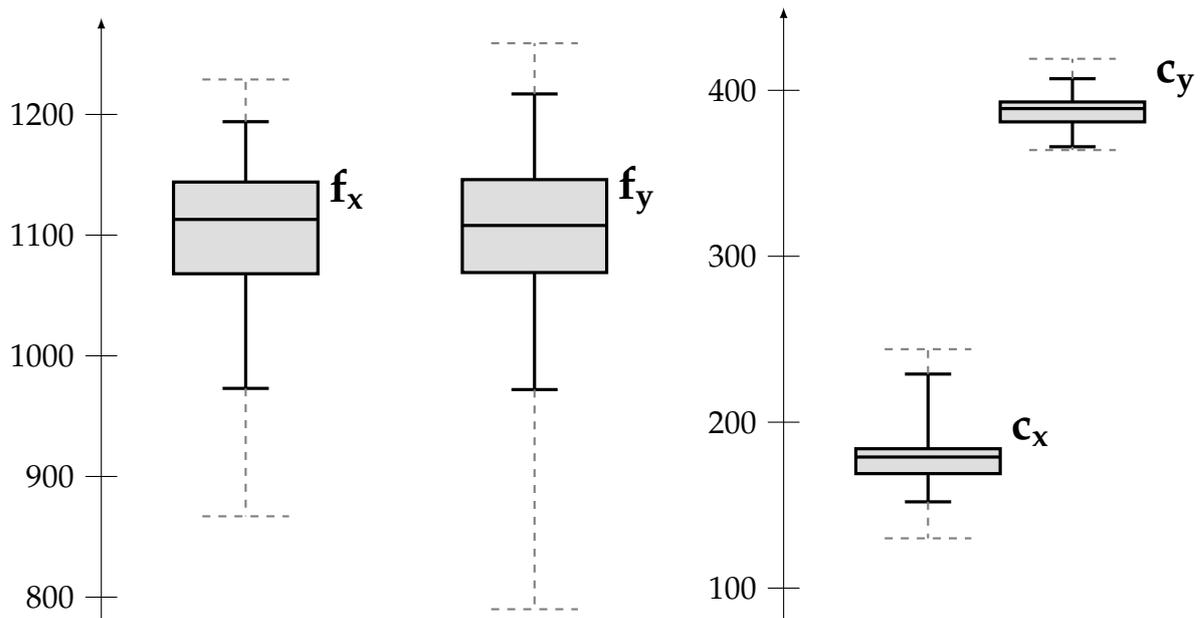


Abbildung 7.13: Streuung der intrinsischen Kameraparameter.  $f_x$ ,  $f_y$  Brennweite in Pixeln;  $c_x$ ,  $c_y$  Koordinaten des Optischen Zentrums in Pixeln.

Die Kalibrierung kann durch ein genaues Datenblatt des Sensors umgangen werden. Leider liegt dieses nicht vom Hersteller vor. Es kann deshalb auch keine Aussage über die Richtigkeit der gemessenen intrinsischen Parameter getroffen werden.

## 7.2.2 Entfernungsmessung

In den Bildern einer Stereoaufnahme können nur einzelne Punkte trianguliert werden. Dazu werden die Flächenschwerpunkt der Kontur herangezogen (Gleichung 4.7).

Die Abweichung  $\delta$  des Messwertes  $Z_{\text{Messung}}$  von der realen Entfernung  $Z_{\text{real}}$  wird nach Gleichung 7.2 und Gleichung 7.3 berechnet.

$$\delta_{\text{absolut}} = Z_{\text{Messung}} - Z_{\text{real}} \quad (7.1)$$

$$\delta_{\text{relativ}} = \frac{Z_{\text{Messung}} - Z_{\text{real}}}{Z_{\text{real}}} \quad (7.2)$$

$$(7.3)$$

In Abbildung 7.14 ist die relative (a) und absolute (b) Abweichung zwischen gemessener und realer Entfernung dargestellt. Die Abweichung vom Mittelwert<sup>42</sup> liegt zwischen

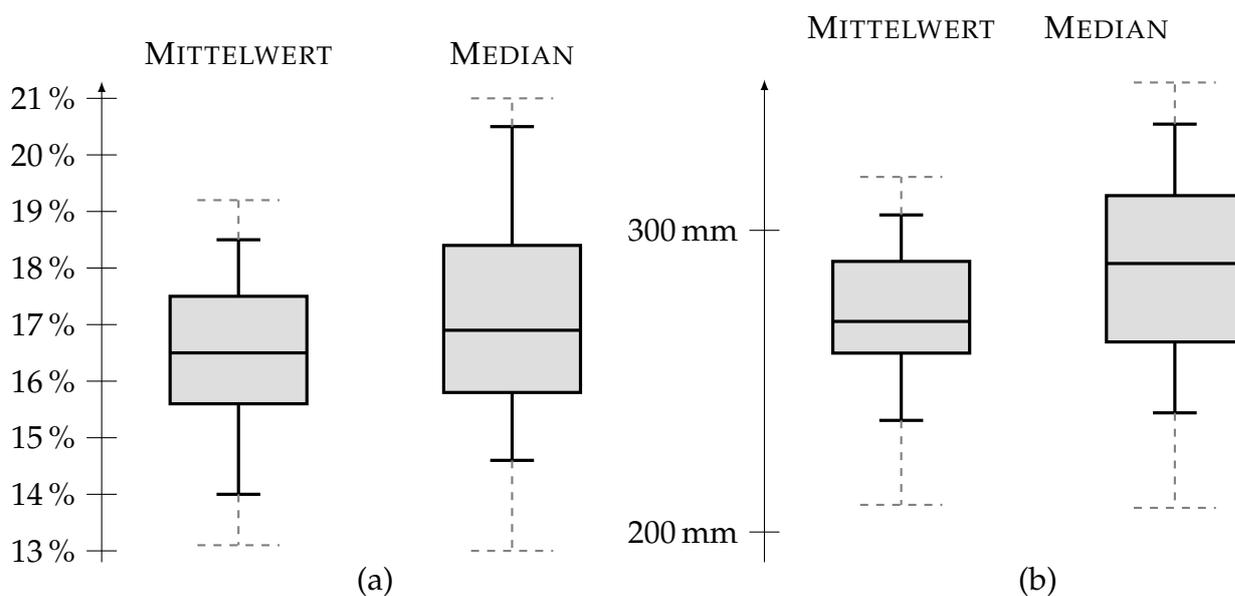


Abbildung 7.14: Abweichung der gemessenen Entfernung. (a) Relative Abweichung nach Gleichung 7.3. (b) absolute Abweichung in mm. Die Messungen wurden in verschiedenen Entfernungen und mit unterschiedlichen Basisabständen durchgeführt.

14 % und 18 %. Absolut weicht die Entfernung im Mittel um 270 mm ab. Die Abweichung vom Median unterliegt einer höheren Streuung.

Bemerkenswert ist, dass die Entfernung ausschließlich überschätzt wird. Dies legt den Schluss nahe, dass die Brennweite ebenfalls konstant überschätzt wird (s. a. Abbildung 7.13).<sup>43</sup>

Werden die Messwerte nach der Entfernung aufgeschlüsselt, so ergibt sich Abbildung 7.15. Die Streuung wächst mit sinkender Entfernung erheblich. Für Distanzen unterhalb 1,4 m konnten keine statistisch verwendbaren Ergebnisse erzielt werden.

<sup>42</sup>Hier wurden Ausreißer nicht für die Berechnung des Mittelwerts berücksichtigt. Ein Ausreißer liegt dann vor, wenn der Wert um mehr als 40 % vom Median abweicht.

<sup>43</sup>Wird die Brennweite  $f_x = 810$  px definiert, so weicht die Entfernung im Mittel nur noch um  $-0,2$  % ab (Median:  $+0,2$  %). An der Streuung ändert dies nichts. Diese bewegt sich – bei Entfernungen größer 1,5 m – im einstelligen Prozentbereich.

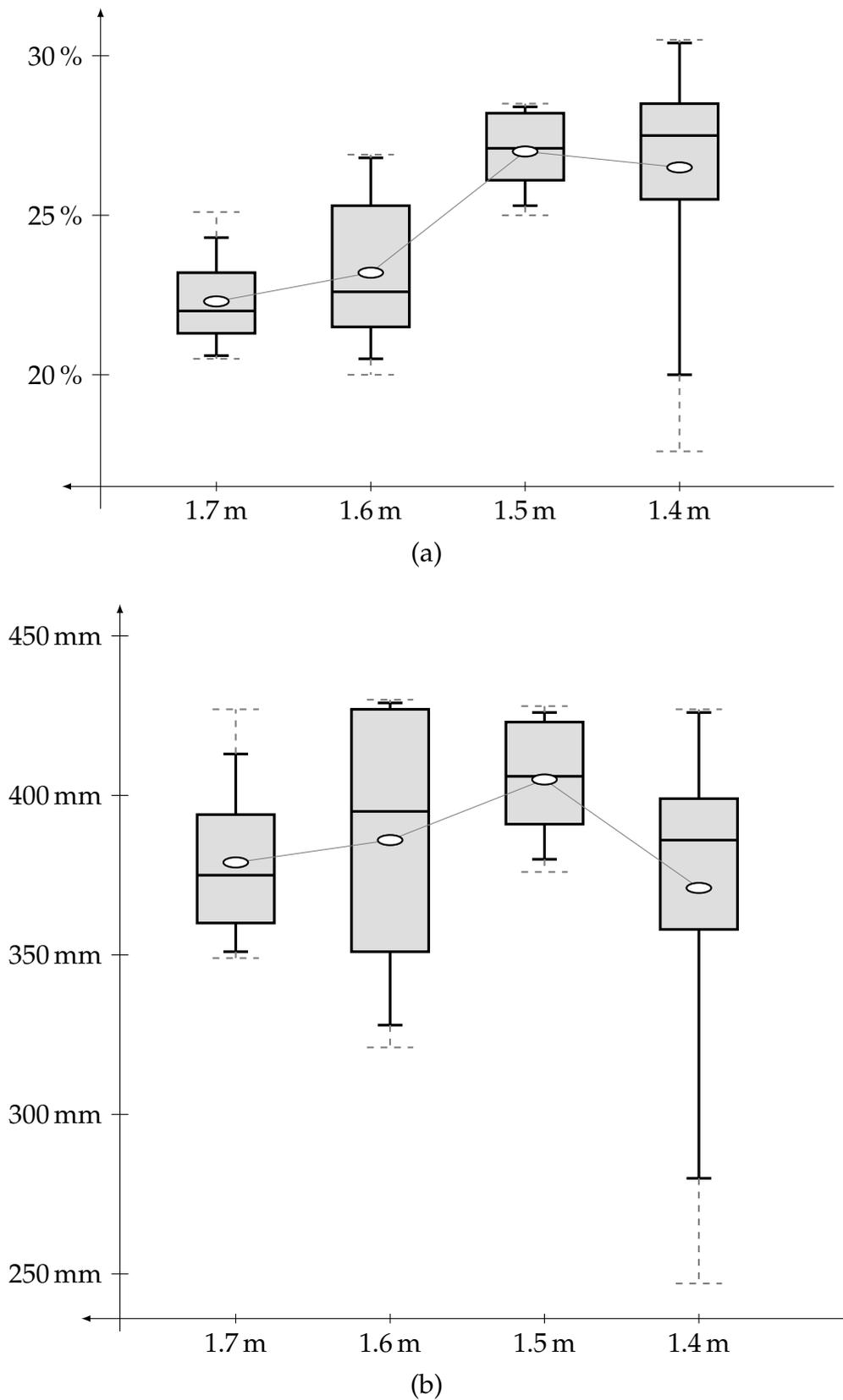


Abbildung 7.15: In unterschiedlichen Abständen zum Objekt wurde die Entfernung geschätzt. Streuung und Mittelwert (weißer Punkt) der Abweichung von der realen Entfernung bleiben annähernd konstant. (a) relative Abweichung; (b) absolute Abweichung.

## **8 Zusammenfassung und Ausblick**

Nachdem im vorherigen Kapitel die Ergebnisse dargestellt und erörtert wurden, wird im nächsten Abschnitt ein Fazit gezogen. Es wird auf die Möglichkeiten und Grenzen der Robot Vision eingegangen und anschließend ein Überblick über weiterführende Forschungsbereiche gegeben (Abschnitt 8.2). Dort wird auch eine Prognose zum zukünftigen Einsatz der Bildverarbeitung in der Industrie allgemein und in der Robotik im Speziellen gegeben.

### **8.1 Zusammenfassung**

#### **8.1.1 Objekterkennung**

Das Erkennen von industriellen Objekten in Bildern kann auf unterschiedliche Arten erfolgen. In dieser Arbeit wurde auf die Detektion von Konturen und der Vergleich ihrer geometrischen und relationalen Parameter zurückgegriffen. Die einzelnen Verfahren sind in Abschnitt 4.2 und 6.3 beschrieben. Die Ergebnisse sind in Abschnitt 7.1 dargestellt.

Gleiche Konturen in unterschiedlichen Kameraaufnahmen können gut mit den geometrischen Charakteristiken der Konturen und ihrer relativen Lage in den Bildern einander zugeordnet werden (vgl. Abbildung 7.2). Die geometrischen Charakteristiken sind abhängig von der Bildqualität. Einen größeren Fokus auf die relationalen Eigenschaften von Konturen legt die (automatische) quellenbasierte Zuordnung (s. Abschnitt 7.1.2). Dabei können die sehr engen Toleranzen der geometrischen Zuordnung gelockert werden, da der Suchbereich des Bildes von vornherein eingegrenzt wird. Dies ist für die modellbasierte Objekterkennung besonders wichtig.

Der Vergleich eines gerenderten Modellbildes mit einer Kameraaufnahme kann nur schwierig mit den geometrischen Parametern erfolgen. Daher wird die quellenbasierte Zuordnung verwendet. Sie liefert beeindruckende Ergebnisse: In 94 % - 99 % der Fälle wird das Objekt richtig erkannt (vgl. Abbildungen 7.6 und 7.8). Die Methode hat jedoch

Probleme, wenn zu wenig Konturpaare erkannt werden. Dies ist bei Nahaufnahmen der Fall (s. a. Abschnitt 7.1.2).

Die modellbasierte Stereo-Objekterkennung gehört thematisch zum Abschnitt „Objekterkennung“. Praktisch ist sie jedoch nur in der Stereovision relevant. Ihre Resultate werden hier erörtert.

Die modellbasierte Stereo-Objekterkennung läuft in zwei Schritten ab: Erst werden die Modellkonturen mit Hilfe der (automatischen) quellenbasierten Zuordnung in der linken Stereoaufnahme erkannt. Anschließend werden diese Konturen der linken Stereoaufnahme im rechten Stereobild gesucht. Dabei werden – je nach Bildausschnitt – nicht immer alle Konturpaare aus dem ersten Schritt auch im zweiten Schritt gefunden. In den 18 Stereo-Testbildern wird das Objekt in 94 % der Fälle erkannt (Abbildung 7.7).<sup>44</sup> In einer größeren Menge an Stereoaufnahmen werden in 85 % der Fälle Korrespondenzen zwischen Modell und den beiden Stereobildern gefunden (Abschnitt 7.1.2).

Es ist schwierig, die Erkennungsrate bzw. die Zuverlässigkeit unterschiedlicher Systeme zu vergleichen. Dies liegt an den zu berücksichtigenden Randbedingungen und dem konkreten Anwendungsfall. Die Gesichtserkennung von O'TOOLE erkennt bspw. in 97 % ein Gesicht. Die Bilder müssen dafür jedoch segmentiert vorliegen, was störenden Hintergrund nahezu vollständig eliminiert.

Auch in Beispielen mit sehr wechselnder Umgebung werden hohe Erkennungsraten erreicht. Bei der Erkennung von Verkehrsschildern werden ebenfalls über 90 % erreicht (EICHNER & BRECKON 2007, 1ff). Allerdings wird hier auch die Farbe genutzt (insbesondere der rote Rand von Verkehrsschildern). RUBLEE et al. geben in ihrer Beschreibung des ORB Algorithmus' ein Beispiel zur Detektion der Lage eines bestimmten Objekts in einer Stereoaufnahme. Hier werden in 61 % der Fälle Objekt und Pose richtig erkannt (RUBLEE et al. 2011, S. 7).

Unter den Randbedingungen einer offenen Umgebung, den damit verbundenen Beleuchtungsschwankungen, dem unruhigen Hintergrund, sind die mit dem in dieser Arbeit beschriebenen System erzielten Resultate sehr gut. Es konnte kein kommerzielles Beispiel für ein solch offenes System in der industriellen Fertigung gefunden werden.

---

<sup>44</sup>Soll für ein sicheres „Erkennen“ mehr als zwei Konturpaare detektiert werden, so fällt dieser Wert auf 72 %

### 8.1.2 Stereovision

Die Resultate der Stereovision sehr kritisch zu betrachten.

Bemerkenswert ist, dass die Entfernung konstant überschätzt wird. Dies legt den Schluss nahe, dass die Brennweite, welche durch die Kamerakalibrierung bestimmt wird, zu groß ist. Es liegt kein Datenblatt des Sensors vor, der diesen Verdacht entkräften konnte. Die Streuung der Messwerte für die Brennweite(n) ist so hoch (Abbildung 7.13), dass die Vermutung weder bestätigt noch widerlegt werden kann.

Die gemessene Entfernung weicht von der realen im Mittel um +20 % bis +25 % ab (s. a. Abbildung 7.15). Wird diese Überschätzung herausgerechnet, so kann das Werkstück zuverlässig angefahren werden. Für eine Validierung, soll eine Kontur auf 20 cm Entfernung angefahren werden, um anschließend ihre genaue Position mit einem Laserscan zu überprüfen. Letzteres ließ der begrenzte zeitliche Umfang der Arbeit nicht mehr zu. Auch musste von einer detaillierten Untersuchung der intrinsischen Parameter  $c_x, c_y$ , mit denen die reale Position  $X, Y$  einer Kontur bestimmt werden kann, abgesehen werden. Hier wird ebenfalls vermutet, dass die Werte der geometrischen Kalibrierung fehlerhaft sind.

Für die Triangulation werden die Flächenschwerpunkte (Gleichung 4.7) der korrespondierenden Konturen verwendet. Durch Abweichungen in der Konturgeometrie wird die Lage des Flächenschwerpunktes beeinflusst. Dadurch wird das Messergebnis negativ beeinflusst. Hier muss untersucht werden, ob es geeignetere Wege gibt, die Konturpaare auf ein Punktepaar zu reduzieren. Auch könnte aus einer Kontur mehr als ein Messwert für die Tiefe gewonnen werden. Darauf wird in Abschnitt 8.2.1 ausführlicher eingegangen.

### 8.2 Ausblick

Dieser Abschnitt gibt einen Überblick zu den konkreten Forschungsbereichen für eine weitere Untersuchung. In Abschnitt 8.2.2 wird auf die zukünftige Entwicklung der Bildverarbeitung in Industrie und Robotik eingegangen.

### 8.2.1 Weiterführende Forschung

Das in dieser Arbeit erstellte Robot Vision System liefert vielversprechende Ergebnisse in der Objekterkennung und beschreibt ein Mono-Stereosystem mit Potential zur weiteren Flexibilisierung und Automatisierung der industriellen Produktion. Dieser Abschnitt soll lohnenswerte Gebiete für eine weitere Untersuchung aufzeigen.

Die Arbeit hat zum Ziel eine Objekt- und Lageerkennung aufzubauen. Es gibt noch viele Möglichkeiten zur Verbesserung. In einem Bildverarbeitungssystem gibt es sehr viele Stellschrauben: Die Schwellenwerte für den Canny Kantendetektor, Auswahl und Größe der Filter, Approximation der Konturen, weitere geometrische Parameter und deren Toleranzen, eine Beurteilung der Qualität einer erkannten Kontur ( Größe, geometrische Übereinstimmung ), etc.

Die Untersuchung anderer geometrischer Parameter erscheint ein besonders lohnenswertes Forschungsfeld zu sein. Es konnte gezeigt werden, dass bereits mit einfachen lokalen und globalen Parametern verhältnismäßig gute Resultate erzielt werden können (vgl. Abschnitt 7.1.1). Der nur am Rande in Abschnitt 4.2.2 angesprochene *Curvature Scale Space (CSS)* Algorithmus, den ABBASI et al. vorgestellt haben, beschreibt einen guten Ansatz für eine Erweiterung der lokalen Konturparameter.

Eine Möglichkeit zur Verbesserung der Zuordnung zwischen Stereoaufnahmen ist der Vergleich der Histogramme von Konturen. So können die Intensitätswerte innerhalb von Konturen verglichen werden. Ähnlich funktionieren Bildmomente (s. Abschnitt 4.2.3), wenn sie nicht auf Binärbilder angewendet werden. Gerade bei der Detektion von Bohrungen und Hinterschneidungen – wie in dieser Arbeit – erscheinen diese in Bildern sehr dunkel. Ihre Histogramme unterscheiden sich daher kaum von dem eines Binärbildes. Zur Detektion anderer Bildmerkmale eignen sie sich jedoch gut (BRADSKI & KAEHLER 2008, 193ff).

Es wurde ein reines Softwareupdate erstellt, mit dem der Roboter befähigt wird, seine Umgebung wahrzunehmen. Es steckt viel Potential in der Verwendung geeigneterer Komponenten. So können mit einer für die Bildverarbeitung konzipierten Kamera hellere und weniger verrauschte Aufnahmen gemacht werden. Ein einfarbiger Hintergrund erleichtert die Segmentierung von Objekten und eine externe Beleuchtung macht nicht nur eine Histogrammanpassung (vgl. Abschnitt 3.2.1) überflüssig,

sondern ermöglicht auch neue Verfahren für die Detektion ( Durchleuchtungsverfahren ).

Die Mono-Stereovision ist ein interessanter Forschungsbereich. Bislang konnten sich keine kommerziellen Anwendungen durchsetzen (SUK & BHANDARKAR 1992a, S. 28). Durch den Preisverfall bei hochwertigen (Stereo-)Kameras scheinen solche Systeme für viele Anwendungen nicht mehr nötig. Das hier beschriebene System kombiniert zwei Messverfahren für verschiedene Entfernungsbereiche. Es kann universell auf bestehende Sensoren übertragen werden und vereint somit niedrige Hürden bei der Integration mit hohem Potential in der Automatisierung.

Die Entfernungsmessung kann über eine sogenannte Tiefenkarte erfolgen (vgl. Abschnitt 5.3). Dadurch können 3D Punktwolken erzeugt werden, die auch zur Objekterkennung verwendet werden können. In einem dichten Punktefeld können Objekte anhand ihrer Entfernung relativ zur Kamera segmentiert werden und über ihre 3D Form erkannt werden.

Die Tiefe kann auch über den Skalierungsunterschied zwischen Modellkontur und der im Bild erkannten korrespondierenden Kontur bestimmt werden. Im Modell kann das Verhältnis aus realer Konturgröße und der Kontur im Modellbild berechnet werden. Über die Skalierung zwischen Modell und Kameraaufnahme kann die reale Konturgröße geschätzt werden:

$$s = \frac{A_{real}}{A_{Modell}} \cdot A_{Kamerabild}$$

Danach kann über die Konturverzerrung auf die Entfernung geschlossen werden. Diese Methode ist für das vorliegende System noch zu untersuchen.

Algorithmen, die die Lage eines Objekts anhand ihrer Verzerrung in einem Bild schätzen sind interessant zu untersuchen. Besonders der POSIT ( englisch „Pose from Orthography and Scaling with Iteration“ ) ist ein nützlicher Algorithmus, der die 3D Lage eines Objekts schätzen kann (BRADSKI & KAEHLER 2008, S. 412).

### 8.2.2 Vision

Die Digitalisierung des klassischen Maschinenbaus ist unter dem Stichwort „Industrie 4.0“ in vollem Gange. Automatisierung und Flexibilisierung gewinnen zunehmend

an Bedeutung (SCHNEIDER & STEIGER 2015). Gerade die Hersteller von Robotern profitieren davon (VDMA 2014). Aktuell sind kollaborierende Roboter, die ohne Sicherheitszäune direkt mit dem Werker arbeiten, in der Entwicklung. Noch hat bspw. der Roboter „YuMi“ von ABB (ABB Ltd.) kein (Stereo-)Kamerasystem zu Umfelderkennung (CIUPEK 2015b). Es ist jedoch leicht vorstellbar, dass bereits in naher Zukunft Roboter auch über Bilderkennung mit dem Menschen interagieren. Der VDMA schätzt sogar, dass bald kein (Industrie-)Roboter mehr ohne „Robot Vision“-System auskommt (MUNKELT & BÖTTGER 2015).

Die industrielle Bildverarbeitung wuchs in den letzten Jahren so stark wie nie (WENDEL 2015, 2014). Dabei erlebt sie einen Wandel weg von reinen Inspektionsaufgaben, hin zum integralen Bestandteil einer Automatisierung (CIUPEK 2014). Durch Objekterkennung können Stillstandszeiten ganzer Produktionsanlagen gesenkt werden, wenn diese selbstständig erkennt, um welches Werkstück es sich handelt und automatisch den richtigen Bearbeitungsablauf auswählt (CIUPEK 2014). Dabei ist „die Zukunft der Robot Vision mehrdimensional“ (FILI 2008). Die Positions- und Lageerkennung durch Stereovision ist eine anspruchsvolle Aufgabe (CIUPEK 2014; FILI 2008). Gerade bei nicht kontrollierter Umgebung – wie es bei kollaborierenden Robotern der Fall ist, wo z. B. der Werker dem Roboter ein Objekt ungeordnet in dessen Arbeitsbereich legt – ist diese Fähigkeit sehr nützlich.

Diese Arbeit reiht sich ein in die Forschung nach einer hochflexiblen, integrativen Fertigung. Es wird sich zeigen, inwieweit das Wachstum im Bereich der Robot Vision anhält. Bisher scheint das Wachstum ungebrochen (VDMA 2014).

## Literaturverzeichnis

ABBASI et al. 1999

Abbasi, S.; F. Mokhtarian; J. Kittler: Curvature scale space image in shape similarity retrieval. English. In: *Multimedia Systems 7*. Centre for Vision Speech and Signal Processing, Department of Electronic & Electrical Engineering, University of Surrey. Springer-Verlag. 1999, S. 467–476.

ADÁN et al. 2009

Adán, A.; A. Martín; E. Valero; P. Merchán: Landmark Real-Time Recognition and Positioning for Pedestrian Navigation. English. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Guadalajara, Mexiko: Springer-Verlag. 2009, S. 21–28. ISBN: 978-3-642-10267-7.

AHLVERS & ZÖLZER 2005

Ahlvers, U.; U. Zölzer: 3D-Auswertung von Stereobildern. Deutsch. *Uni Forschung [Forschungsmagazin der Helmut-Schmidt-Universität - Universität der Bundeswehr Hamburg]* (2005) 15. Fachbereich Elektrotechnik.

AZUMA 1997

Azuma, R.: A survey of augmented reality. *Presence: Teleoperators and Virtual Environments* 6 (1997) 4. cited By 1654, S. 355–385. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-0039560342&partnerID=40&md5=dd05a75c8e426c66deef694752614a7e>.

BARTHEL 2003

Barthel, K. U.: Verlustlose Bildkompression. Deutsch. In: *it - Information Technology*. 45. FHTW Berlin. Oldenbourg Verlag. 2003. URL: [http://www.f4.fhtw-berlin.de/people/barthel/paper/barthel\\_it.pdf](http://www.f4.fhtw-berlin.de/people/barthel/paper/barthel_it.pdf) (besucht am 26.08.2015).

BATCHELOR 2012

Batchelor, B. G.: *Machine Vision Handbook*. English. 1st ed. Bd. 1. New York: Springer. 2012. ISBN: 978-1-84996-168-4.

BAY et al. 2007

Bay, H.; A. Ess; T. Tuytelaars; L. Van Gool: Speeded-Up Robust Features (SURF). English. *Computer Vision and Image Understanding* 110 (2007), S. 346–359.

BEHLEY & STEINHAGE 2009

Behley, J.; V. Steinhage: Generation of 3D City Models Using Domain-Specific Information Fusion. English. In: *Computer Vision Systems*. Liège, Belgium: Springer-Verlag. 2009, S. 164–173. ISBN: 978-3-642-04666-7.

BELUSSI & HIRATA 2011

Belussi, L.; N. Hirata: Fast QR code detection in arbitrarily acquired images. English. In: *Proceedings - 24th SIBGRAPI Conference on Graphics, Patterns and Images*. cited By 11. 2011, S. 281–288. DOI: 10.1109/SIBGRAPI.2011.16. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84857173890&partnerID=40&md5=08c43159c4581e9bd13563776bf157f2>.

BHARATH 2014

Bharath, P.: OpenCV: QR Code detection and extraction | DsynFLO. English. URL: <http://dsynflo.blogspot.ca/2014/10/opencv-qr-code-detection-and-extraction.html> (besucht am 11.10.2015).

BÖHM 2004

Böhm, J.: Modellbasierte Segmentierung und Objekterkennung aus Distanzbildern. Deutsch. ISSN 0065-5325. Diss. Stuttgart: Universität Stuttgart. 2004.

BOSCH 2014

Bosch: MyDriveAssist | Bosch - MyDriveAssist Fahrerassistenz App. Deutsch. URL: <http://iphone.bosch.com/mydriveassist/> (besucht am 12.10.2015).

BÖTTGER 2014

Böttger, U.: Prüfsystem übernimmt Sisyphusarbeit an Flugzeugrumpfteilen - Roboter inspiziert 40 000 Niete - Konradin Verlag. URL: <http://www.industrieanzeiger.de/qualitaetssicherung/-/article/32571342/40077092/Roboter-inspiziert-40-000-Nieten> (besucht am 27.09.2015).

BRADSKI & KAEHLER 2008

Bradski, G.; A. Kaehler: *Learning OpenCV: [computer vision with the OpenCV library]*. English. 1. ed. Software that sees. Beijing: O'Reilly Media Inc. 2008. ISBN: 978-0-596-51613-0.

BRAHMBHATT 2013

Brahmbhatt, S.: *Practical OpenCV: [hands on project for computer vision on the Windows, Linux and Raspberry Pi platforms]*. English. TIA, technology in action. New York: Apress. 2013. ISBN: 978-1-4302-6079-0.

CAMPOS et al. 2009

Campos, S.; R. Salas; H. Allende; C. Castro: Multimodal Algorithm for Iris Recognition with Local Topological Descriptors. English. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Guadalajara, Mexiko: Springer-Verlag. 2009, S. 766–773. ISBN: 978-3-642-10267-7.

CANNY 1986

Canny, J.: A Computational Approach to Edge Detection. English. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 6. IEEE Computer Society. 1986, S. 679–698. URL: [https://perso.limsi.fr/vezien/PAPIERS\\_ACS/canny1986.pdf](https://perso.limsi.fr/vezien/PAPIERS_ACS/canny1986.pdf).

CIUPEK 2014

Ciupek, M.: Automation braucht Durchblick. Deutsch. *VDI nachrichten* (2014) 47. URL: <http://www.vdi-nachrichten.com/Technik-Wirtschaft/Automation-Durchblick>.

CIUPEK 2015a

Ciupek, M.: Grenzen für autonome Roboter werden ausgelotet. Deutsch. *VDI nachrichten* (2015) 27. URL: <http://www.vdi-nachrichten.com/Technik-Wirtschaft/Grenzen-fuer-autonome-Roboter-ausgelotet>.

CIUPEK 2015b

Ciupek, M.: Kollaborierende Roboter erfordern Umdenken. Deutsch. *VDI nachrichten* (2015) 17.

DOUGLAS & PEUCKER 1973

Douglas, D.; T. Peucker: Algorithms for the reduction of the number of points required to represent a digitized line or is caricature. English. In: *The Canadian Carographer*. 2. Aufl. Bd. 10. Toronto: Ontario Institute of Chartered Cartographers. 1973, S. 112–122. URL: <http://catalog.loc.gov/vwebv/holdingsInfo?searchId=19690&recCount=25&recPointer=0&bibId=11278679>.

DRUNK 2015

Drunk, G.: Optische Messtechnik mit Bildverarbeitung. URL: <http://www.>

xpertgate . de / produkte / Optische - Messtechnik - Bildverarbeitung .  
html (besucht am 28. 09. 2015).

EICHNER & BRECKON 2007

Eichner, M. L.; T. P. Breckon: Integrated Speed Limit Detection and Recognition from Real-Time Video. URL: [http://breckon.eu/toby/publications/papers/eichner08speedlimit\\_a.pdf](http://breckon.eu/toby/publications/papers/eichner08speedlimit_a.pdf).

FADZILAH & DZULKIFLI 2009

Fadzilah, A.; M. Dzulkifli: Fingerprint Classification Based on Analysis of Singularities and Image Quality. English. In: *Visual Informatics: Bridging Research and Practice*. Kuala Lumpur, Malaysia: Springer-Verlag. 2009, S. 551–560. ISBN: 978-3-642-05035-0.

FILI 2008

Fili, W.: Die Zukunft der Robot-Vision ist mehrdimensional. Deutsch. URL: <http://www.maschinenmarkt.vogel.de/themenkanale/produktion/verbindungstechnik/articles/109516/> (besucht am 27. 09. 2015).

FLUSSER 1999

Flusser, J.: On the independence of rotation moment invariants. English. In: *Pattern Recognition [The Journal of the Pattern Recognition Society]*. Bd. 33. Elsevier Science Ltd. 1999, S. 1405–1410. URL: <http://library.utia.cas.cz/prace/20000033.pdf>.

FRAGA 2009

Fraga, L. G. de la: Self-calibration from Planes Using Differential Evolution. English. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Guadalajara, Mexiko: Springer-Verlag. 2009, S. 724–731. ISBN: 978-3-642-10267-7.

GEHRIG et al. 2009

Gehrig, S. K.; F. Eberli; T. Meyer: A Real-Time Low-Poper Stereo Vision Engine Using Semi-Global Matching. English. In: *Computer Vision Systems*. Liège, Belgium: Springer-Verlag. 2009, S. 134–143. ISBN: 978-3-642-04666-7.

GOM 2013

GOM: Digitalisierung und Inspektion von großen Bauteilen - Schlüsselfertige Messzelle - Konradin Verlag. Deutsch. URL: <http://www.industrieanzeiger.de/qualitaetssicherung/-/article/32571342/38561754/Schl%C3%83%C2%BCselfertige-Messzelle> (besucht am 27. 09. 2015).

GREEN 1828

Green, G.: An Essay on the Application of Mathematical Analysis to the Theories of Electricity and Magnetism. English. URL: <https://books.google.ae/books?id=GwYXAAAAYAAJ&pg=PA10&hl=ar#v=onepage&q&f=false>.

GREENHALGH & MIRMEHDI 2012

Greenhalgh, J.; M. Mirmehdi: Real-time detection and recognition of road traffic signs. *IEEE Transactions on Intelligent Transportation Systems* 13 (2012) 4. cited By 0, S. 1498–1506. DOI: 10.1109/TITS.2012.2208909. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84870511714&partnerID=40&md5=5d24c0847b8b6b79e8ed2e409f34454f>.

GU et al. 2011

Gu, Z.; M. Scott; G. Chen; J. Tien: „Qr code detection“. US20110290882 A1. US-Klassifikation 235/462.11; Internationale Klassifikation G06K7/10; Unternehmensklassifikation G06K7/1456. URL: <https://www.google.com/patents/US20110290882>.

HAMME et al. 2015

Hamme, D.; W. Goeman; P. Veelaert; W. Philips: Robust monocular visual odometry for road vehicles using uncertain perspective projection. *Eurasip Journal on Image and Video Processing* 2015 (2015) 1. cited By 0. DOI: 10.1186/s13640-015-0065-6. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84929000869&partnerID=40&md5=e45056b6ac3f65cafe7612caacb537f1>.

HARTLEY & ZISSERMAN 2003

Hartley, R.; A. Zisserman: *Multiple view geometry in computer vision*. 2nd ed. Cambridge, UK ; New York: Cambridge University Press. 2003. ISBN: 0521540518.

HECKBERT 1994

Heckbert, P. S., Hrsg. (1994): *Graphics gems IV*. The Graphics gems series. Boston: AP Professional. 1994. ISBN: 0123361567 0123361559.

HOF 2013

Hof, R. D.: New Techniques from Google and Ray Kurzweil Are Taking Artificial Intelligence to Another Level | MIT Technology Review. English. URL: <http://www.technologyreview.com/featuredstory/513696/deep-learning/> (besucht am 28.09.2015).

HU 1962

Hu, M.-K.: Visual Pattern Recognition by Moment Invariants. In: *IRE Transactions on Information Theory*. Downloaded on February 4, 2010 at 13:21 from IEEE Xplore. 1962, S. 179–187.

JAIN et al. 1995

Jain, R.; R. Kasturi; B. G. Schunck: Object Recognition. In: *Machine Vision*. McGraw-Hill series in computer science. New York: McGraw-Hill. 1995, S. 459–491. ISBN: 0-07-032018-7. URL: [Index%20of%20/~rlk/MachineVisionBook/MachineVision.files](http://index%20of%20/~rlk/MachineVisionBook/MachineVision.files).

KILIAN 2001

Kilian, J.: Simple Image Analysis By Moments Version 0.2. English. URL: <http://breckon.eu/toby/teaching/dip/opencv/SimpleImageAnalysisbyMoments.pdf> (besucht am 21.08.2015).

KOLL 2015

Koll, S.: Control 2015: Trends bei Koordinatenmessgeräten - Multisensorik statt rein taktiler Methoden - Konradin Verlag. Deutsch. URL: <http://www.industrieanzeiger.de/qualitaetssicherung/-/article/32571342/40639915/Multisensorik-statt-rein-taktiler-Methoden> (besucht am 27.09.2015).

KOTHE 2009

Kothe, N.: Automatisiertes Messen liegt im Trend. URL: <http://www.elektroniknet.de/messen-testen/sonstiges/artikel/20878/> (besucht am 27.09.2015).

KUKA 2013

KUKA, R. G., Hrsg. (März 2013): KR QUANTEC extra HA [Spezifikation]. Deutsch. Version: Spez KR QUANTEC extra HA V2 de (PDF).

LAGANIERE 2014

Laganier, R.: *OpenCV Computer Vision Application Programming Cookbook*. English. 2. Aufl. Birmingham: Packt Publishing. 2014. ISBN: 978-1-78216-148-6.

LAMMERS et al. 2003

Lammers, D.; S. Wachenfeld; C. Lordemann; M. Lambers: „Objekterkennung in Bilddaten“. Deutsch. Seminar: „Unterstützung von Landminendetektion durch Bildauswertungsverfahren und Robotereinsatz“ (WS 03/04). Münster.

URL: <http://wwwmath.uni-muenster.de/u/lammers/EDU/ws03/Landminen/Abgaben/Gruppe9/Thema09-ObjekterkennungInBilddaten-ChristianGrosseLordemann-MartinLammers.pdf> (besucht am 14.04.2015).

LEVI 2013a

Levi, G.: A Short introduction to descriptors. URL: <https://gilscvblog.wordpress.com/2013/08/18/a-short-introduction-to-descriptors/> (besucht am 20.09.2015).

LEVI 2013b

Levi, G.: A tutorial on binary descriptors – part 2 – The BRIEF descriptor. URL: <https://gilscvblog.wordpress.com/2013/09/19/a-tutorial-on-binary-descriptors-part-2-the-brief-descriptor/> (besucht am 20.09.2015).

LOWE 2004

Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoint. English. *International Journal of Computer Vision* (2004).

MOBILEYE 2014

Mobileye: Traffic Sign Detection. English. URL: <http://mobileye.com/en/en/technology/applications/traffic-sign-detection/> (besucht am 11.10.2015).

MOESLUND 2012

Moeslund, T. B.: *Introduction to video and image processing [building real systems and applications]*. 1st ed. Undergraduate topics in computer science. New York: Springer. 2012. ISBN: 978-1-4471-2502-0.

MUNKELT & BÖTTGER 2015

Munkelt, O.; U. Böttger: Dr. Olaf Munkelt über die Rolle der Bildverarbeitung in der Automobilindustrie - „Roboter ohne Kamera, das geht gar nicht - Konradin Verlag. Deutsch. URL: <http://www.industrieanzeiger.de/automation/-/article/32571342/41106914> (besucht am 27.09.2015).

NAWRATH 2013

Nawrath, A.: 3D-Geometrieerfassung mit Stereovision. Deutsch. Wiki. ISSN 2195-9986. URL: [https://wiki.zimt.uni-siegen.de/fertigungsautomatisierung/index.php/3D-Geometrieerfassung\\_mit\\_Stereovision](https://wiki.zimt.uni-siegen.de/fertigungsautomatisierung/index.php/3D-Geometrieerfassung_mit_Stereovision) (besucht am 15.04.2015).

NÜCHTER & HERTZBERG 2008

Nüchter, A.; J. Hertzberg: Towards semantic maps for mobile robots. *Robotics and Autonomous Systems* 56 (2008) 11. cited By 147, S. 915–926. DOI: 10.1016/j.robot.2008.08.001. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-53849127134&partnerID=40&md5=e7a003324338d5222382b8297c83e7ac>.

OLIVEIRA JR. et al. 2009

Oliveira Jr., J. J. de; C. O. de A. Freitas; J. M. de Carvalho; R. Sabourin: Handwritten Word Recognition Using Multi-view Analysis. English. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Guadalajara, Mexiko: Springer-Verlag. 2009, S. 371–378. ISBN: 978-3-642-10267-7.

OPENCV 2015a

OpenCV, d. t.: Geometric Image Transformations — OpenCV 2.4.11.0 documentation. URL: [http://docs.opencv.org/modules/imgproc/doc/geometric\\_transformations.html](http://docs.opencv.org/modules/imgproc/doc/geometric_transformations.html) (besucht am 15.06.2015).

OPENCV 2015b

OpenCV, d. t.: Histogram Equalization — OpenCV 2.4.11.0 documentation. URL: [http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram\\_equalization/histogram\\_equalization.html](http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html) (besucht am 21.05.2015).

OPENCV 2015c

OpenCV, d. t.: Image Filtering — OpenCV 2.4.11.0 documentation. URL: <http://docs.opencv.org/modules/imgproc/doc/filtering.html> (besucht am 15.10.2015).

OPENCV 2015d

OpenCV, d. t.: Structural Analysis and Shape Descriptors — OpenCV 2.4.11.0 documentation. English. URL: [http://docs.opencv.org/modules/imgproc/doc/structural\\_analysis\\_and\\_shape\\_descriptors.html](http://docs.opencv.org/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html) (besucht am 07.10.2015).

O'TOOLE 2014

O'Toole, J.: Facebook's new face recognition knows you from the side. URL: <http://money.cnn.com/2014/04/04/technology/innovation/facebook-facial-recognition/index.html> (besucht am 12.10.2015).

OTTERBACH 1995

Otterbach, R.: *Robuste 3D-Objekterkennung und Lagebestimmung durch Auswertung von 2D-*

*Bildfolgen*. Deutsch. Bd. 8. 8: Meß-, Steuerungs- und Regelungstechnik 481. Düsseldorf: VDI Verlag. 1995. ISBN: 3-18-348101-1.

PARIS et al. 2007

Paris, S.; P. Kornprobst; J. Tumblin; F. Durand: A Gentle Introduction to Bilateral Filtering and its Applications. English. URL: [https://people.csail.mit.edu/sparis/bf\\_course/](https://people.csail.mit.edu/sparis/bf_course/) (besucht am 17.09.2015).

RIEMANN 1867

Riemann, B.: Grundlagen für eine allgemeine Theorie der Functionen einer veränderlichen complexen Grösse. Deutsch. Diss. Göttingen: Georg-August-Universität Göttingen. 1867.

ROBOCUP 2015

RoboCup: Rules | RoboCup Humanoid League. English. URL: <https://www.robocuphumanoid.org/materials/rules/> (besucht am 12.10.2015).

ROSTEN & DRUMMOND 2005

Rosten, E.; T. Drummond: Fusing Points and Lines for High Performance Tracking. English. In: *IEEE International Conference on Computer Vision*. 2005, S. 1508–1511. DOI: 10.1109/ICCV.2005.104. URL: [http://www.coxphysics.com/work/rosten\\_2005\\_tracking.pdf](http://www.coxphysics.com/work/rosten_2005_tracking.pdf).

RUBLEE et al. 2011

Rublee, E.; V. Rabaud; K. Konolige; G. Bradski: ORB: an efficient alternative to SIFT or SURF. English. In: Willow Garage, Menlo Park, California. Barcelona. 2011.

SCHEDDEEN 2010

Schedeen, B. J.: The History of 3D Movie Tech. URL: <http://www.ign.com/articles/2010/04/23/the-history-of-3d-movie-tech> (besucht am 11.10.2015).

SCHNEIDER & STEIGER 2015

Schneider, L.; H. Steiger: Welle der Automatisierung rollt. Deutsch. *VDI nachrichten* (2015) 7. URL: <http://www.vdi-nachrichten.com/Technik-Gesellschaft/Die-Welle-Automatisierung-rollt>.

SCHWARZ 2015

Schwarz, J.: Carl Zeiss in Oberkochen beschleunigt die Qualitätssicherung - Messtechnik darf nicht zum Nadelöhr werden - Konradin Verlag. Deutsch. URL: <http://www.>

industrieanzeiger.de/home/-/article/12503/32913598/Messtechnik-darf-nicht-zum-Nadel (besucht am 27.09.2015).

SEO et al. 2015

Seo, Y.-W.; J. Lee; W. Zhang; D. Wettergreen: Recognition of Highway Workzones for Reliable Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems* 16 (2015) 2. cited By 0, S. 708–718. DOI: 10.1109/TITS.2014.2335535. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84927514621&partnerID=40&md5=a168ae4d7b3014fd9cae230673af8f91>.

SHOJAEIPOUR et al. 2009

Shojaeipour, S.; S. M. Haris; M. I. Khairir: Vision-Based Mobile Robot Navigation Using Image Processing and Cell Decomposition. English. In: *Visual Informatics: Bridging Research and Practice*. Kuala Lumpur, Malaysia: Springer-Verlag. 2009. ISBN: 978-3-642-05035-0.

SPENRATH 2015

Spenrath, F.: Bin picking 3d / Griff-in-die-Kiste mit Zweiarm-Roboter. Fraunhofer IPA. URL: <http://www.youtube.com/watch?v=xhTkgaJg8wQ> (besucht am 27.09.2015).

STADLMEYER 2011

Stadlmeyer, R.-P.: *Installation eines Visionsystems zur Objekterkennung an einem Industrieroboter*. Deutsch. Saarbrücken: VDM Verlag Dr. Müller. 2011. ISBN: 978-3-8364-6501-4.

STARBURG 2014

Starburg: hacking touchID. Deutsch. Gulaschprogrammiersnacht 14. Münchsteinach. URL: [https://media.ccc.de/browse/conferences/gpn/gpn14/gpn14\\_-\\_5835\\_-\\_de\\_-\\_medientheater\\_-\\_201406211730\\_-\\_hacking\\_touchid\\_-\\_starbug.html#video](https://media.ccc.de/browse/conferences/gpn/gpn14/gpn14_-_5835_-_de_-_medientheater_-_201406211730_-_hacking_touchid_-_starbug.html#video) (besucht am 12.10.2015).

STEINGRUBE et al. 2009

Steingrube, P.; S. K. Gehrig; U. Franke: Performance Evaluation of Stereo Algorithms for Automotive Applications. English. In: *Computer Vision Systems*. Liège, Belgium: Springer-Verlag. 2009, S. 285–294. ISBN: 978-3-642-04666-7.

STOCK et al. 2002

Stock, C.; U. Mühlmann; M. Krishna; A. Pinz: Subpixel Corner Detection for Tracking Applications using CMOS Camera Technology. English. In: Bd. 160. Graz. 2002, S. 191–

199. URL: <http://graphics.ucsd.edu/~manu/pdf/aapr2002.pdf> (besucht am 03.10.2015).

SUK & BHANDARKAR 1992a

Suk, M.; S. M. Bhandarkar: *Three-Dimensional Object Recognition from Range Images*. English. 1. Aufl. Computer Science Workbench. Tokyo; New York: Springer-Verlag. 1992. ISBN: 4-431-70107-9.

SUK & BHANDARKAR 1992b

Suk, M.; S. M. Bhandarkar: *Three-Dimensional Object Recognition from Range Images*. English. 1. Aufl. Computer Science Workbench. Tokyo; New York: Springer-Verlag. 1992. ISBN: 4-431-70107-9.

TAIGMAN et al. 2014

Taigman, Y.; M. Yang; M. A. Ranzato; L. Wolf: DeepFace: Closing the Gap to Human-Level Performance in Face Verification. English. In: Columbus, Ohio, USA. 2014.

TAYLOR & KRIEGMAN 1998

Taylor, C.; D. Kriegman: Vision-based motion planning and exploration algorithms for mobile robots. *IEEE Transactions on Robotics and Automation* 14 (1998) 3. cited By 59, S. 417–426. DOI: 10.1109/70.678451. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-0032099186&partnerID=40&md5=4743d5d74db6bcb5391517e0370358e4>.

TREIBER 2010

Treiber, M.: *An introduction to object recognition [selected algorithms for a wide variety of applications]*. Advances in Pattern Recognition. London; New York: Springer. 2010. ISBN: 978-1-84996-234-6.

ULRICH 2013

Ulrich, M.: iwB - Roboterbasierte, hochgenaue Qualitätssicherungssysteme (RoHoQ). URL: <http://www.iwb.tum.de/RoHoQ-path-1,1001998,955290,1024.html> (besucht am 27.09.2015).

VAN DAMME 2015

Van Damme, J.: TIFF File Format FAQ. URL: <http://www.awaresystems.be/imaging/tiff/faq.html> (besucht am 26.08.2015).

VDMA 2014

VDMA, F. I. B., Hrsg. (2014): VDMA Branchenführer Industrielle Bildverarbeitung

2015/16. Deutsch. URL: [http://ibv.vdma.org/documents/256550/0/IBV\\_2014\\_D\\_LR.pdf/212a74a6-f6e7-4448-94bd-c185a47aa851](http://ibv.vdma.org/documents/256550/0/IBV_2014_D_LR.pdf/212a74a6-f6e7-4448-94bd-c185a47aa851) (besucht am 27.09.2015).

WECKENMANN 2012

Weckenmann, A., Hrsg. (2012): *Koordinatenmesstechnik [Flexible Strategien für funktions- und fertigungsgerechtes Prüfen]*. Deutsch. 2. Aufl. München: Carl Hanser Verlag. 2012. ISBN: 978-3-446-40739-8.

WENDEL 2014

Wendel, A.: VDMA: Industrielle Bildverarbeitung so stark wie nie. Deutsch. id:5936188. URL: <http://ibv.vdma.org/article/-/articleview/5936188> (besucht am 27.09.2015).

WENDEL 2015

Wendel, A.: VDMA: Industrielle Bildverarbeitung bricht alle Rekorde. Deutsch. 16.07.2015 | id:8966181. URL: <http://ibv.vdma.org/article/-/articleview/8966181;jsessionid=301B6EE570BECC6DEBD82E4D038B8E39> (besucht am 27.09.2015).

YUAN 2004

Yuan, C.: *Artificial Neural Networks for Object Recognition and Localization*. English. Fraunhofer Series in Information and Communication Technology 9/2004. Aachen: Shaker. 2004. ISBN: 3-8322-3293-1.

ZHANG et al. 2015

Zhang, X.; B. Xian; B. Zhao; Y. Zhang: Autonomous Flight Control of a Nano Quadrotor Helicopter in a GPS-Denied Environment Using On-Board Vision. *IEEE Transactions on Industrial Electronics* 62 (2015) 10. cited By 0, S. 6392–6403. DOI: 10.1109/TIE.2015.2420036. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84941591748&partnerID=40&md5=e432183e406b5c48d2376e51bb7a0c26>.

# Anhang

## A.1 Teil 1: Ergänzende Abbildungen

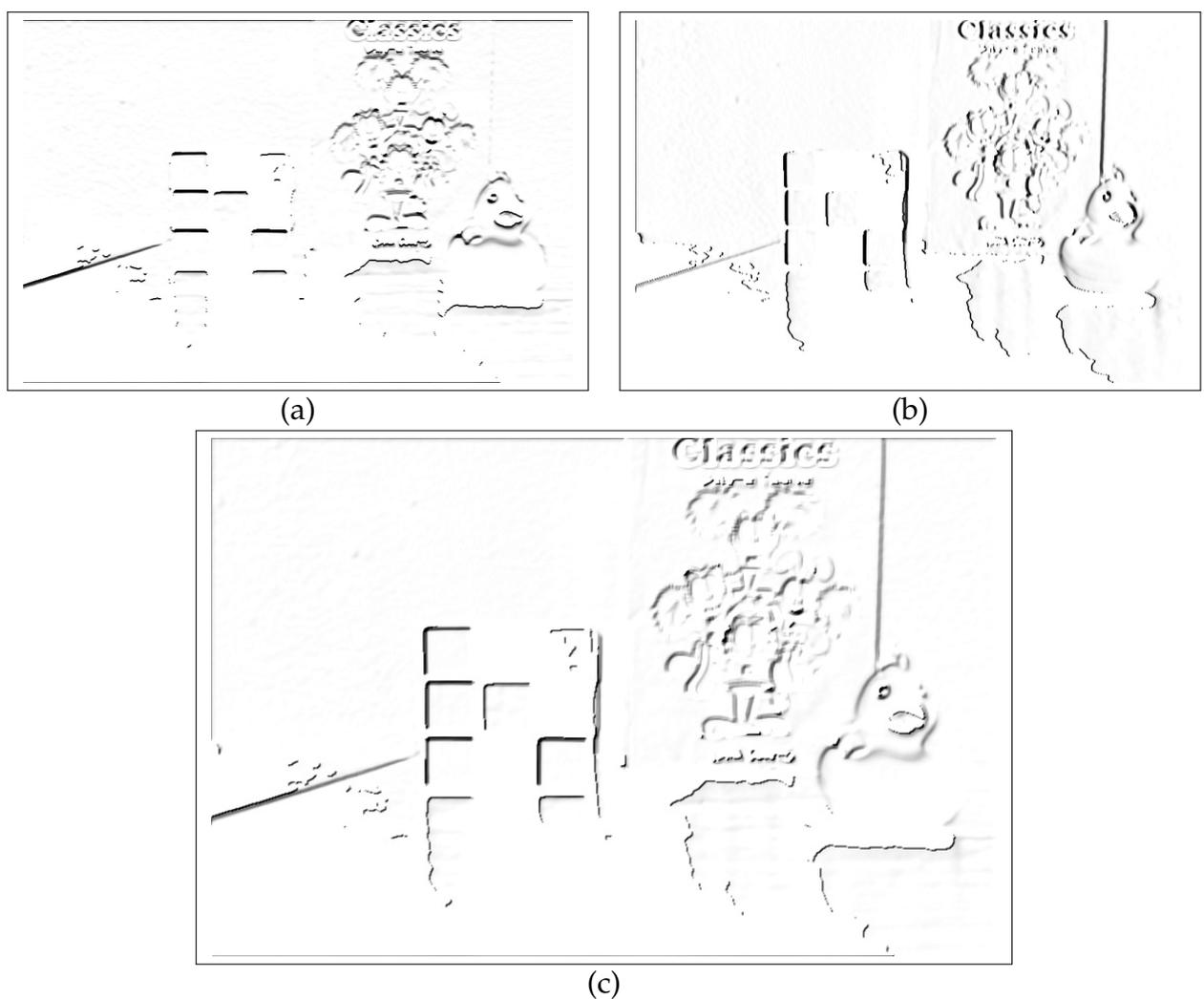


Abbildung A.1: Sobel-Filter: (a) horizontaler Kernel; (b) vertikaler Kernel; (c) Überlagerung des horizontalen und des vertikalen Kernels.  
Zur besseren Darstellung sind die Farben invertiert.

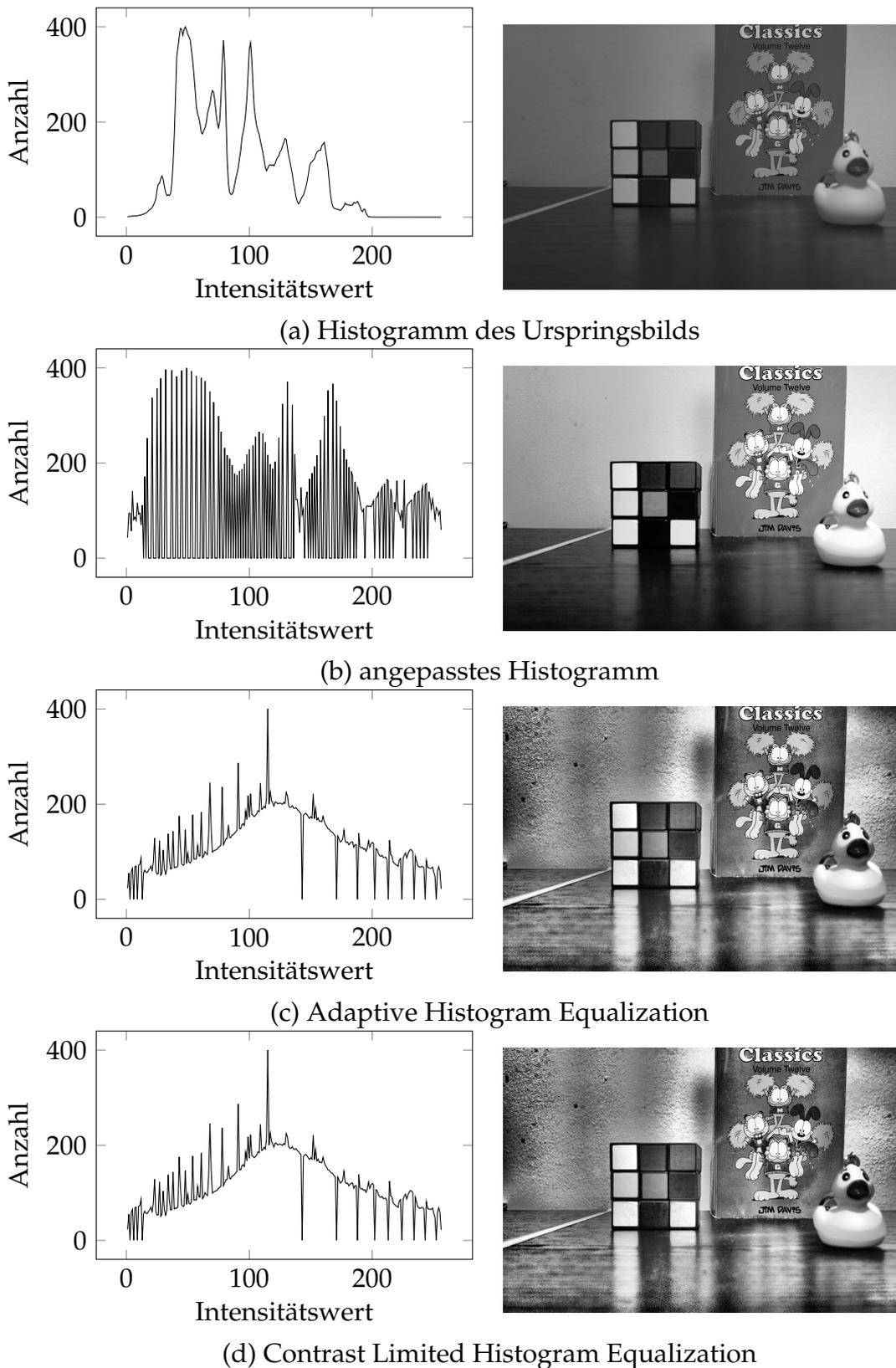


Abbildung A.2: Histogrammanpassung. Die Histogramme zeigen wie häufig ein bestimmter Intensitätswert  $[1, 256]$  im Bild vorkommt. (a) Originalbild; (b) Bild mit angepasstem Histogramm; (c) Bild mit AHE auf  $6 \times 6$  Bildregionen angewendet; (d) Bild mit CLAHE auf  $6 \times 6$  Bildregionen und einem Limit von 100.

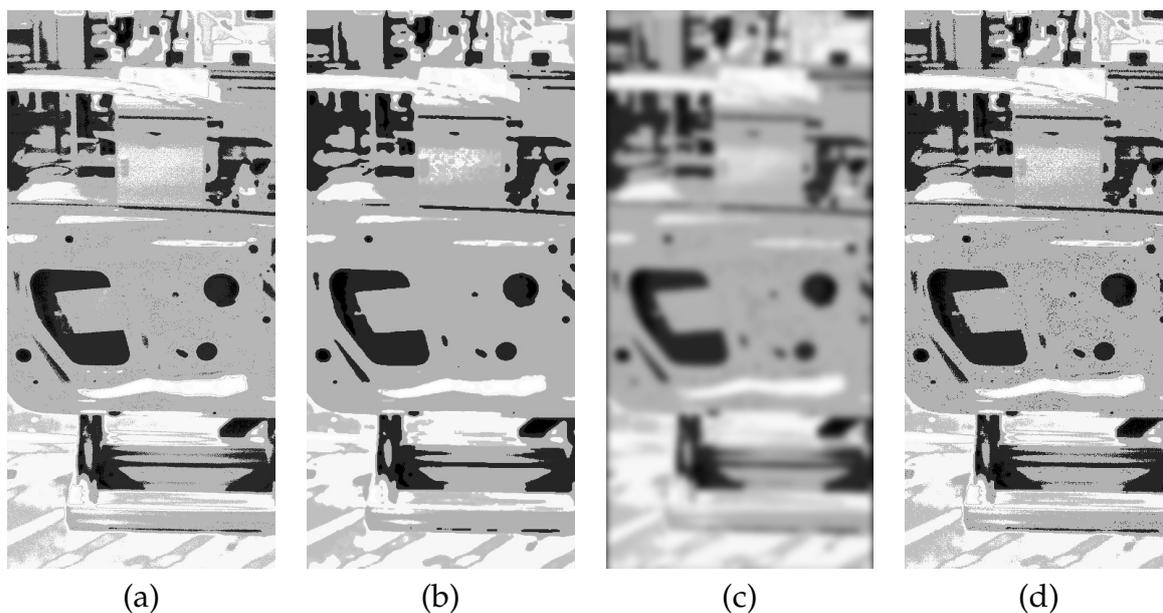


Abbildung A.3: (a) angepasste Originalaufnahme; (b) Median-Filter Kernelgröße  $k = 5$ ; (c) Gauß-Filter  $\sigma = 5$ ; (d) Bilateral-Filter Kerlengröße  $d = 5$ ,  $\sigma_{color} = 13$ ,  $\sigma_{space} = 3,5$ . Die Details sind der Vollständigkeit halber hier angegeben. Für ihre genaue Bedeutung sei auf die OpenCV Dokumentation verwiesen: (OPENCV 2015c).

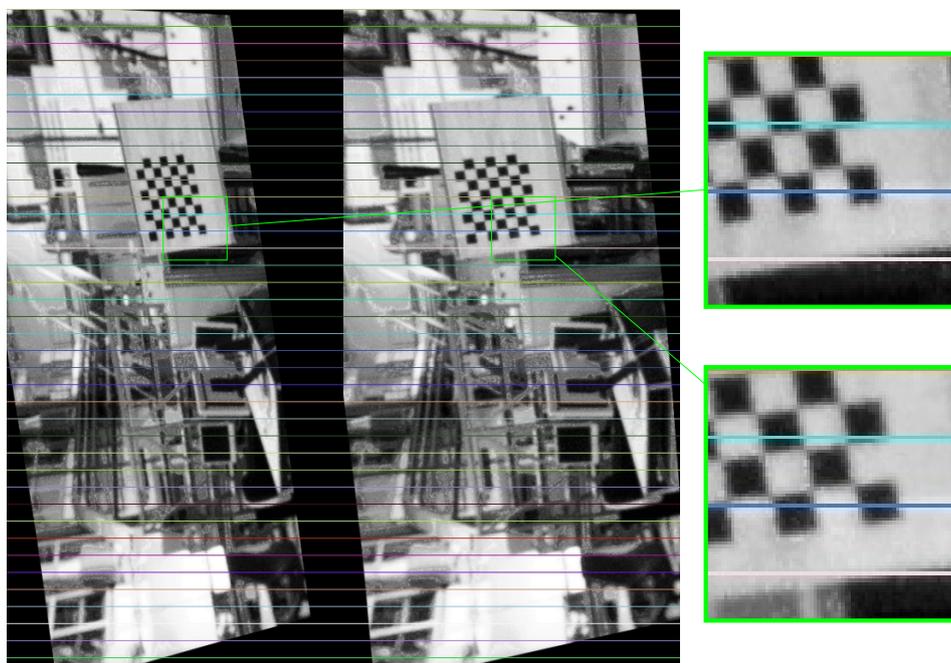


Abbildung A.4: Rektifizierte Stereoaufnahmen ohne Stereo-Kalibrierung. ( Das Kalibrierungsmuster wurde als Objekt verwendet um definierte Punkte zu generieren. ) Mit Hilfe der geschätzten Fundamentalmatrix werden die Bilder so ausgerichtet, dass Bildpunkte des gleichen realen Objekts in derselben Zeile liegen.  
Kamera: Helix-Sensor.

## A.2 Teil 2: Details Boxplots

Tabelle A.1: Streuung intrinsische Kameraparameter

	$f_x$	$f_y$	$c_x$	$c_y$
<b>Mittelwert</b>	1096,0	1099,4	180,7	386,7
<b>Minimum</b>	867,1	790,4	129,9	363,7
<b>Q<sub>5%</sub></b>	972,9	971,6	151,9	366,3
<b>Q<sub>25%</sub></b>	1068,1	1068,6	169,1	380,7
<b>Median</b>	1112,9	1108,0	179,1	388,7
<b>Q<sub>75%</sub></b>	1143,5	1146,0	183,7	393,4
<b>Q<sub>95%</sub></b>	1194,3	1216,7	228,7	406,7
<b>Maximum</b>	1229,0	1258,8	243,9	419,4

Tabelle A.2: Abweichung der Entfernungsmessung. Details zu Abbildung 7.14.

	Relative Abweichung vom Mittelwert	Absolute Abweichung vom Mittelwert	Relative Abweichung vom Median	Absolute Abweichung vom Median
<b>Mittelwert</b>	16,4 %	272,8 mm	17,2 %	285,9 mm
<b>Minimum</b>	13,1 %	209,0 mm	13,0 %	208,0 mm
<b>Q<sub>5%</sub></b>	14,0 %	233,7 mm	14,6 %	239,5 mm
<b>Q<sub>25%</sub></b>	15,6 %	259,3 mm	15,8 %	263,0 mm
<b>Median</b>	16,5 %	269,8 mm	16,9 %	289,9 mm
<b>Q<sub>75%</sub></b>	17,5 %	289,7 mm	18,4 %	311,5 mm
<b>Q<sub>95%</sub></b>	18,5 %	305,1 mm	20,5 %	335,2 mm
<b>Maximum</b>	19,2 %	317,7 mm	21,0 %	349,0 mm

Tabelle A.3: Relative Abweichung der Entfernungsmessung. Details zu Abbildung 7.15(a).

	1,7 m	1,6 m	1,5 m	1,4 m
<b>Mittelwert</b>	22.3 %	23.2 %	27.0 %	26.5 %
<b>Minimum</b>	20.5 %	20.0 %	25.0 %	17.6 %
<b>Q<sub>5%</sub></b>	20.6 %	20.5 %	25.3 %	20.0 %
<b>Q<sub>25%</sub></b>	21.2 %	21.5 %	26.1 %	25.5 %
<b>Median</b>	22.0 %	22.6 %	27.1 %	27.5 %
<b>Q<sub>75%</sub></b>	23.2 %	25.3 %	28.2 %	28.5 %
<b>Q<sub>95%</sub></b>	24.3 %	26.8 %	28.4 %	30.4 %
<b>Maximum</b>	25.1 %	26.9 %	28.5 %	30.5 %

Tabelle A.4: Absolute Abweichung der Entfernungsmessung. Details zu Abbildung 7.15(b).

	1,7 m	1,6 m	1,5 m	1,4 m
<b>Mittelwert</b>	378,5 mm	386,2 mm	405,1 mm	371,0 mm
<b>Minimum</b>	349,9 mm	320,7 mm	375,5 mm	246,5 mm
<b>Q<sub>5%</sub></b>	351,0 mm	327,8 mm	379,8 mm	280,3 mm
<b>Q<sub>25%</sub></b>	360,4 mm	351,3 mm	391,4 mm	357,5 mm
<b>Median</b>	374,6 mm	395,1 mm	405,8 mm	385,5 mm
<b>Q<sub>75%</sub></b>	394,3 mm	427,1 mm	423,0 mm	399,9 mm
<b>Q<sub>95%</sub></b>	412,8 mm	429,3 mm	426,2 mm	425,6 mm
<b>Maximum</b>	426,5 mm	430,3 mm	428,9 mm	426,7 mm

### A.3 Teil 3: Linien Zuordnung

Industrielle Halbzeuge und Bauteile zeichnen sich häufig durch gerade Kanten aus. Diese können mit der Hough Transformation (vgl. Abschnitt 3.4) detektiert werden. Für diese Arbeit wurde ein „Line-Matcher“ entwickelt, der Geraden zwischen zwei Bildern vergleicht und darüber Objekte erkennt. Aufgrund der langen Laufzeit des Algorithmus' und des begrenzten zeitlichen Umfangs der Arbeit wurde seine weitere Entwicklung eingestellt. Das Prinzip wird hier skizziert und anhand eines einfachen Beispiels illustriert.

Der Algorithmus läuft nach folgenden Schritten ab:

1. Detektion von Kanten
2. Detektion von Geraden bspw. durch die Hough Transformation
3. Bestimmen der Geradenschnittpunkte
4. Berechnung der Winkel zwischen den Geraden in ihren Schnittpunkten
5. Grobe Selektion über die Winkel
6. Bilden von Dreiecken aus allen Schnittpunkten
7. Vergleich der Dreiecke über ihre Winkel
8. Vergleich der Flächenverhältnisse aller Dreiecke

Nach der Detektion von Geraden werden ihre Schnittpunkte und Winkel zueinander bestimmt. Anhand der Winkel werden erste Schnittpunkte aussortiert. Diese Selektion ist recht grob und dient der Laufzeitreduzierung. Anschließend werden zwischen den Schnittpunkten alle möglichen Dreiecke gebildet. Die Anzahl der Dreiecke steigt sehr schnell mit der Anzahl  $n$  der Schnittpunkte bzw. der detektierten Geraden:

$$\#Dreiecke = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n 1$$

Bei 3 Schnittpunkten ist es ein Dreieck, bei 4 Schnittpunkten sind es drei Dreiecke und bei 5 Schnittpunkten sind es sechs Dreiecke, u.s.w.

Die Anzahl der Schnittpunkte ergibt sich aus:  $\#Linien \cdot (\#Linien - 1)$ .

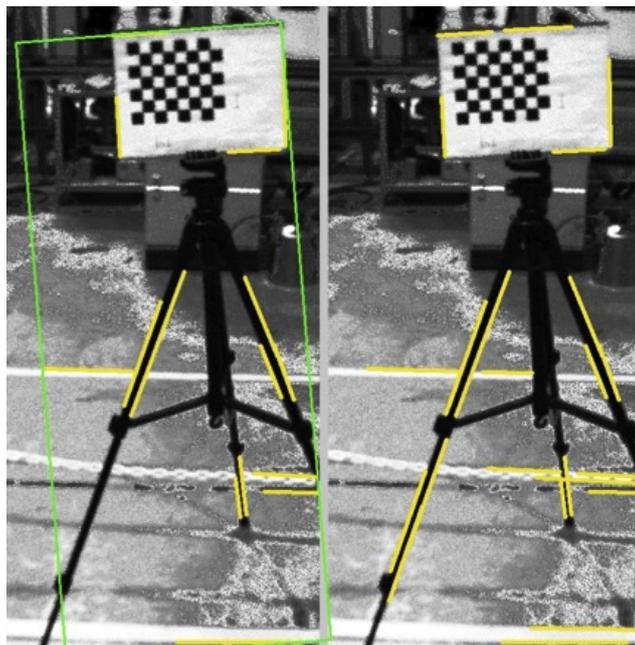


Abbildung A.5: Resultat der Linienzuordnung. Im linken Bild wurde das erkannte Objekt grün eingerahmt. Die erkannten Linien sind in beiden Bildern gelb markiert. Das rechte Bild zeigt das gesuchte Objekt. In diesem Fall die identische Aufnahme.

Die gebildeten Dreiecke werden nun über die Winkel in beiden Bildern verglichen. Wird kein Dreieck mit (annähernd) gleichen Winkeln gefunden, so wird dieses aussortiert.<sup>45</sup> Anschließend werden – ähnlich wie bei der relationalen Überprüfung aus Abschnitt 6.3.2 – die Flächenverhältnisse der zugeordneten Dreiecke gebildet und davon der Median bestimmt. Dreieckspaare, deren Flächenverhältnis zu stark vom Median abweicht, werden aussortiert. Die übrigen Dreiecke bzw. die dazugehörigen Geraden gelten als erkannt. Das Vorgehen ist in Abbildung A.6 dargestellt. Das Resultat eines Beispielbildes zeigt Abbildung A.5.

Die Zahl der zu vergleichenden Dreiecke kann reduziert werden und mit ihnen die Programmlaufzeit, wenn man von einem richtig zugeordneten Dreieck ausgehend die Dreiecke bildet. Zu zwei richtigen Punkten wird immer ein neuer Punkt hinzugenommen um ein Dreieck zu bilden. So reduziert sich die Anzahl aller möglichen Dreiecke auf eine normale Summe (im Gegensatz zu einer dreifachen Summe). Dieses Vorgehen ist ähnlich der quellenbasierten Zuordnung. Die Schwierigkeit besteht darin, die Quelle zu bestimmen.

<sup>45</sup>Dabei sind nur zwei Winkel unabhängig von einander (es gilt:  $\alpha + \beta + \gamma = 180$ ). Benutzt man Geometrien, die aus mehr Punkten gebildet werden können – Viereck, Oktagon, ... – so kann hier eine eindeutigere Zuordnung erreicht werden.

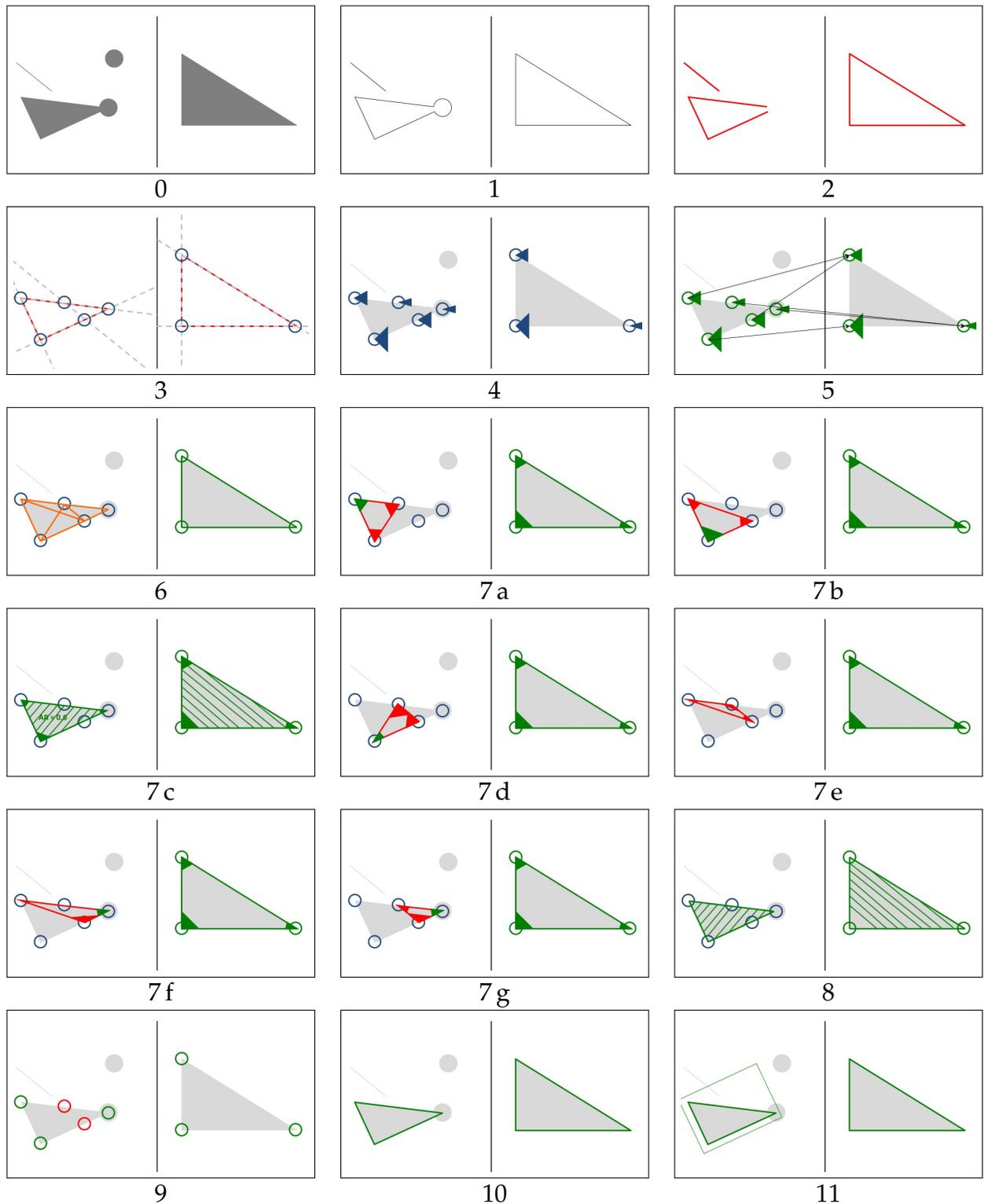


Abbildung A.6: Schrittweiser Ablauf der Zuordnung von Linien. 0 Ausgangsbilder: Kameraaufnahme (links) und zu erkennendes Objekt (rechts); 1 Kantendetektion; 2 Liniendetektion; 3 Schnittpunktbestimmung; 4 Bestimmen der Winkel; 5 Selektion über die Winkel; 6 Bilden aller Dreiecke; 7 Einzelne Schritte beim Vergleich der Dreiecke in beiden Bildern über die Winkel; 8 Vergleich der Flächenverhältnisse; 9 (nicht) zugeordnete Schnittpunkte (rot) grün; 10 erkannte Kanten; 11 erkanntes Objekt (eingerahmt).

## A.4 Teil 4: Erstelltes Programm

In diesem Teil des Anhangs wird das in der Arbeit erstellte Software-Programm anhand seiner Unterprogramme skizziert. Abbildung A.7 zeigt das Zusammenspiel der Teilprogramme.

Das Hauptprogramm ist *iRobot\_3DVision\_addon*. Es ist in C# geschrieben und steuert den Roboter mit Hilfe der DLL (englisch „Dynamic Linked Library“), die von *iwb* geschrieben wurde. Die Analyseprogramme sind mit der freien Softwarebibliothek OpenCV (Version 2.4.11) in C++ geschrieben. Sie werden vom Hauptprogramm über den Standard-Eingangs-String (englisch „standard input string“ (stdin)) als Kommandozeilenprogramm aufgerufen. Dies kann in der GUI des Hauptprogramms unter *Settings > Robot* ausgeschaltet werden.

Die Benutzeroberfläche ist in drei Bereiche aufgeteilt: Einstellungen („Settings“), Messungen („Take Pictures“) und die Bewegungsüberwachung („Movement Control“). In diesem letzten Bereich kann der Roboter über Vorgabe kartesischer Koordinaten in einer PTP-Bewegung verfahren werden. Hier wird auch die aktuelle Position (nach einer Bewegung) angezeigt. Die Definition der einzelnen Achsen ist in kleinen Piktogrammen dargestellt.

Im Bereich *Settings*, kann über den Reiter *Connection* die Verbindung zum Roboter spezifiziert und aufgebaut werden. Unter *Robot* wird die Achsengeschwindigkeit und -beschleunigung angepasst. Standardmäßig sind niedrige Werte eingestellt. Unter *Image* kann zwischen den verschiedenen Verfahren zur Histogrammanpassung gewechselt werden. Dies betrifft nur die Unterprogramme *RV\_StereoVision.exe* und *RV\_ModelBasedStereoVision.exe*. Hier kann auch eingestellt werden, ob Stereobilder (modellbasiert) analysiert werden sollen. Über *Search* kann die automatische Werkstücksuche eingestellt werden (Sicherheitsfaktoren, welche Koordinatenrichtungen angefahren werden sollen, etc.). Dabei wird immer das Programm *RV\_[ModelBased]StereoVision.exe* aufgerufen, unabhängig von der Benutzerauswahl. Ein Objekt gilt als erkannt, wenn die Unterprogramme Entfernungswerte in eine CSV-Datei geschrieben haben. Diese Schaltfläche ist das schlussendliche Ziel der Arbeit. Dieses Programm ist keine Messung im eigentlichen Sinne und wurde deshalb im Bereich *Settings* angeordnet.

Der Bereich *Take Pictures* wird standardmäßig mit dem Reiter *Stereo* geöffnet. Hier können Stereobilder erzeugt und angezeigt werden. Die angezeigten Bilder sind immer

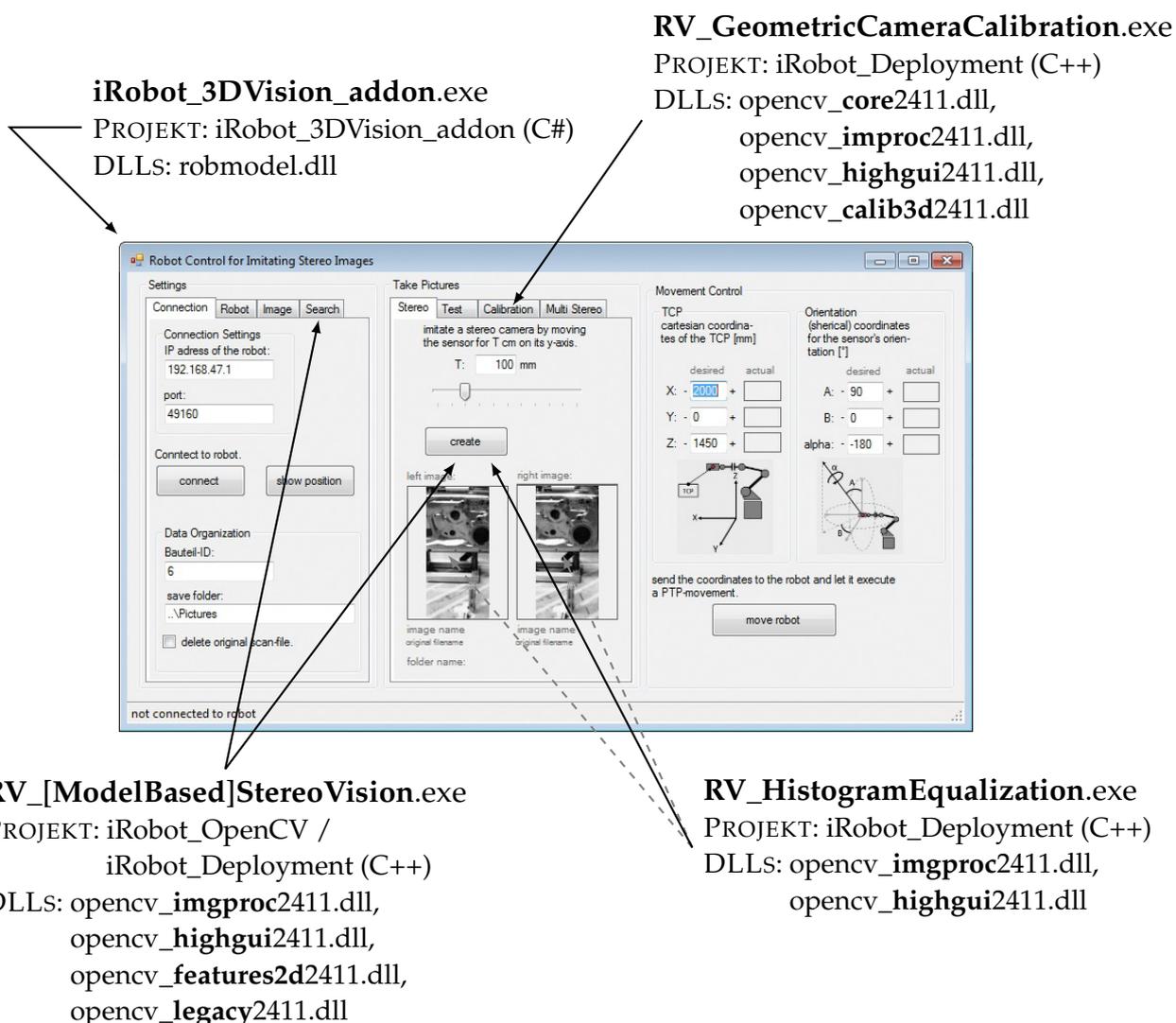


Abbildung A.7: Zusammenarbeit der einzelnen Teilprogramme.

über die Histogrammäquivalisation (HE) angepasste Bilder. Unter *Test* kann man einzelne Bilder aufnehmen um die Ausrichtung des Roboters zu überprüfen. *Calibration* stellt das Kalibrierungsmuster ein und spezifiziert den Pfad, wo die intrinsische Matrix gespeichert ist/wird.<sup>46</sup> Mit dem Reiter *Multi Stereo* können mehrere Stereoaufnahmen erzeugt werden, deren kombinierte Analyse eine genauere Entfernungsmessung ermöglicht. Er dient rein experimentellen Zwecken und wurde nicht weiterentwickelt.

<sup>46</sup>Diese ist als OpenCV  $3 \times 3$  Matrix (Typ „Mat“) mit dem Namen „cameraMatrix“ in einer XML-Datei mit dem Namen „Data.xml“ gespeichert.

## **Eidesstattliche Erklärung**

Ich erkläre hiermit eidesstattlich, dass ich die vorliegende Arbeit selbstständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche gekennzeichnet.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Garching, den 20.10.2015

(Max Schwenzler)