

# Erkennen und Lokalisieren von Objekten in komplexer Umgebung anhand von Videobildern

Diplomarbeit

am

Forschungsinstitut für  
anwendungsorientierte Wissensverarbeitung

vorgelegt  
von

David Oliver Horn  
Fachhochschule Augsburg  
Fachbereich Informatik

Erstprüfer: Prof. Georg Stark  
Betreuer: Dr. Erwin Praßler

Sommersemester 1996



# Erklärung

Hiermit erkläre ich, David Oliver Horn, daß diese Arbeit selbständig verfaßt und noch nicht anderweitig für Prüfungszwecke vorgelegt wurde. Alle Quellen und Hilfsmittel sind im Literaturverzeichnis angegeben. Wörtliche und sinngemäße Zitate sind als solche gekennzeichnet.

Augsburg, 19. Oktober 2002

.....  
(David Oliver Horn)



# Vorwort

Die hier vorgestellte Diplomarbeit „Erkennen und Lokalisieren von Objekten in komplexer Umgebung anhand von Videobildern“ wurde am Forschungsinstitut für anwendungsorientierte Wissensverarbeitung, kurz FAW, im Projekt DAVID verfaßt.

Das in Ulm ansässige Institut wurde 1987 mit dem Ziel gegründet, vorwettbewerbliche Forschung und Auftragsforschung durchzuführen. Dabei geht es nur zum Teil darum, neue Technologien zu entwickeln. Ein Großteil der Forschung orientiert sich an der praktischen Umsetzung von theoretischen Erkenntnissen. Dabei finanziert sich das Institut zum überwiegenden Teil selbst, wird jedoch auch durch das Land Baden-Württemberg und namhafte Industriepartner<sup>1</sup> gefördert. Die Hauptbetätigungsfelder am FAW sind Geschäftsprozeßmanagement, Telematikdienste, Umweltinformationssysteme, Verkehrsinformatik, Softwaretechnik und Kommunikationssysteme sowie Autonome Systeme.

Das Projekt DAVID ist in den Bereich Autonome Systeme eingegliedert. Ziel ist es, ein autonomes System zu entwerfen bzw. zu entwickeln, das sich in einer Büroumgebung zurechtfindet und gestellte Aufgaben selbständig bearbeitet. Beispielhaft dafür ist das Einsammeln von Papierkörben, die in der Arbeitsumgebung an beliebiger Stelle abgestellt sind. Für die praktische Ausführung steht dem Projekt DAVID eine Experimentierplattform der amerikanischen Firma Nomadic Technologies, Inc. zur Verfügung (siehe Abb. 1).

Für die Realisierung werden hauptsächlich bestehende theoretische Ansätze verwendet, die für eine praktische Anwendung in der realen Welt jedoch meist nicht ohne tiefgreifende Änderungen und Erweiterungen übernommen werden können. Für die sichere und zuverlässige Ausführung der gestellten Aufgabe wirken viele Einzelkomponenten zusammen. Die einzelnen Teilbereiche sind dabei die Positionsbestimmung vor und während der Fahrt, die Bestimmung von optimalen Beobachterstandpunkten für die Objektsuche, die Wegeplanung mittels Landkarten, das interaktive Abfahren der geplanten Strecke, um auf dynamische Veränderungen der Umgebung zu reagieren, die Kartierung der aktuellen Umgebung, die Objekterkennung zusammen mit der Positionsbestimmung für erkannte Objekte sowie das Andocken an ein gefundenes Objekt. Weiterhin notwendig sind Verfahren, die jeweiligen Aktivitäten des Roboters zu koordinieren, so daß die erfolgreiche Ausführung der gestellten Aufgabe sichergestellt ist.

---

<sup>1</sup>Daimler-Benz AG, Digital Equipment GmbH, Hewlett Packard GmbH, IBM Deutschland GmbH, Robert Bosch GmbH, Siemens-Nixdorf Informationssysteme AG und strässle Informationssysteme GmbH

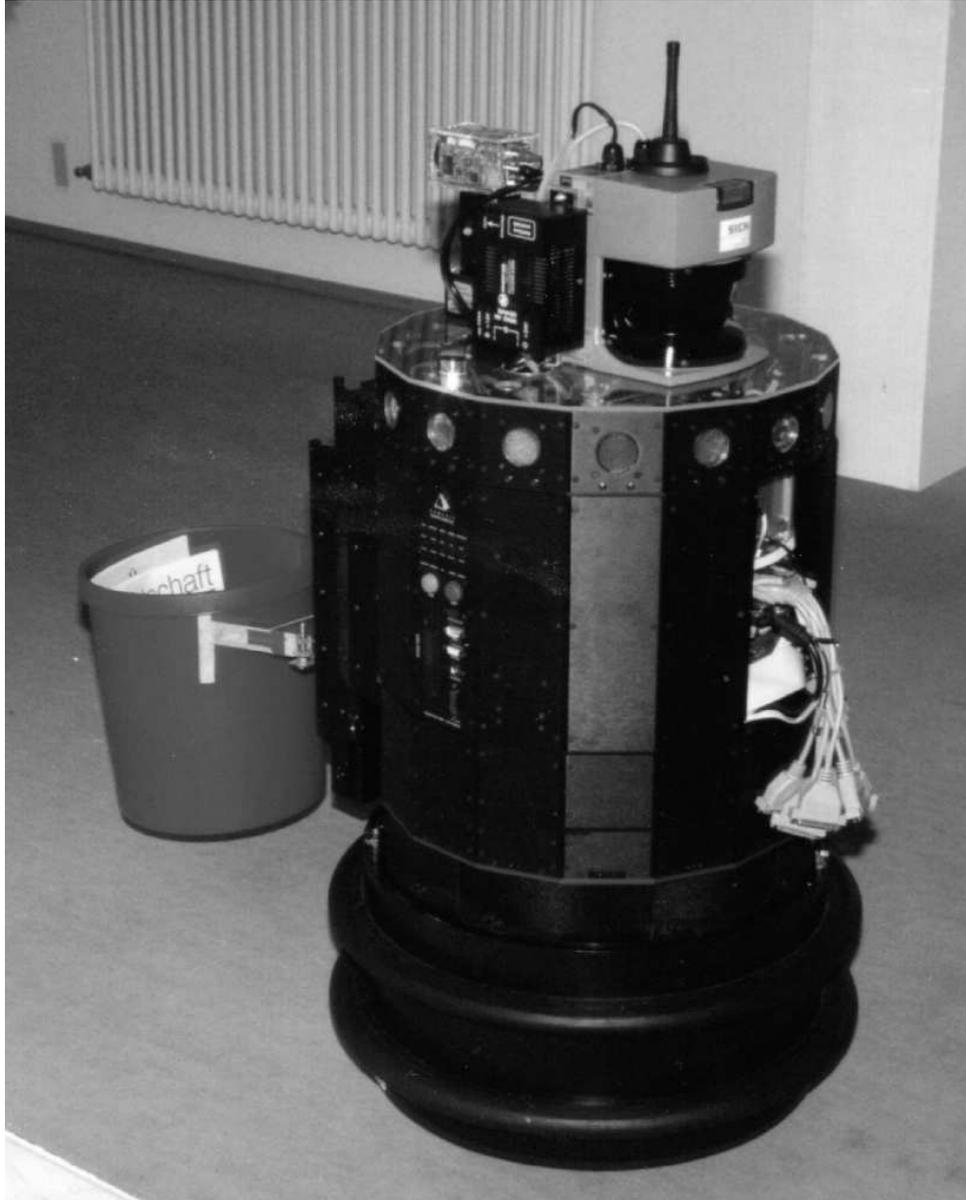


Abbildung 1: Das autonome System NOMAD

# Inhaltsverzeichnis

<b>Erklärung</b>	<b>iii</b>
<b>Vorwort</b>	<b>v</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Was bedeutet es zu sehen? . . . . .	1
1.2 Die Probleme für einen Computer . . . . .	1
1.3 Klassifizierung von Bildverarbeitung . . . . .	2
1.3.1 Digitale Bildverarbeitung . . . . .	2
1.3.2 Mustererkennung . . . . .	2
1.3.3 Objekterkennung . . . . .	2
1.4 Erläuterung der Aufgabenstellung . . . . .	3
1.4.1 Kontext der Aufgabe . . . . .	3
1.4.2 Konkrete Aufgabenstellung . . . . .	3
1.5 Überblick . . . . .	3
<b>2 Digitale Bildverarbeitung</b>	<b>5</b>
2.1 Einleitung . . . . .	5
2.2 Grundlagen . . . . .	6
2.2.1 Darstellungsformen von Bildern . . . . .	6
2.2.2 Definition von Farbe . . . . .	7
2.3 Klassifizierung von Bildoperatoren . . . . .	10
2.3.1 Punktoperatoren . . . . .	10
2.3.2 Lokale Operatoren . . . . .	10
2.3.3 Globale Operatoren . . . . .	10
2.3.4 Objektorientierte Operatoren . . . . .	11
2.4 Bildglättung . . . . .	11
2.4.1 Mittelwertfilter . . . . .	11
2.4.2 Medianfilter . . . . .	12
2.4.3 Gaußsches Glättungsfilter . . . . .	12
2.4.4 Größenfilter . . . . .	13
2.5 Segmentierung durch Klassifizierung . . . . .	13
2.5.1 Fest dimensionierte überwachte Klassifizierung . . . . .	13
2.5.2 Fest dimensionierte unüberwachte Klassifizierung . . . . .	14
2.5.3 Lernende Klassifizierung . . . . .	14
2.6 Kantenextraktion . . . . .	14
2.6.1 Numerische Approximation . . . . .	15
2.6.2 Sobel-Operator . . . . .	15
2.6.3 Laplace-Operator . . . . .	16
2.7 Binärbilderzeugung . . . . .	16
2.7.1 Fester Schwellwert . . . . .	17
2.7.2 Adaptiver Schwellwert . . . . .	17

2.7.3	Variabler Schwellwert . . . . .	17
2.8	Distanztransformation . . . . .	17
2.8.1	Definition von Distanz . . . . .	17
2.8.2	Berechnung der Distanztransformation . . . . .	19
2.8.3	Voronoi-Diagramm . . . . .	19
<b>3</b>	<b>Objekterkennung</b> . . . . .	<b>21</b>
3.1	Einführung . . . . .	21
3.2	Einstufung der gestellten Aufgabe . . . . .	22
3.2.1	Komplexität der Bildinformation . . . . .	22
3.2.2	Komplexität der Modellinformation . . . . .	22
3.2.3	Komplexitätsklassifizierung . . . . .	22
3.2.4	Wertung . . . . .	23
3.3	Grundlagen . . . . .	23
3.3.1	Modelldatenbank . . . . .	23
3.3.2	Merkmalsextraktion . . . . .	25
3.3.3	Aufstellen einer Objekthypothese . . . . .	25
3.3.4	Überprüfung der Objekthypothese . . . . .	26
3.4	Gängige Objekterkennungsstrategien . . . . .	27
3.4.1	Interpretationsbäume . . . . .	27
3.4.2	Ausrichtung . . . . .	28
3.4.3	Hough-Transformation . . . . .	28
3.4.4	Geometrisches Hashing . . . . .	29
3.4.5	Berechenbare Geometrie . . . . .	30
3.4.6	Aufteilung des Transformationsraumes . . . . .	30
<b>4</b>	<b>Hausdorff-Distanz</b> . . . . .	<b>33</b>
4.1	Einführung . . . . .	33
4.2	Definition . . . . .	33
4.2.1	Gerichtete Hausdorff-Distanz . . . . .	33
4.2.2	Ungerichtete Hausdorff-Distanz . . . . .	34
4.3	Metrische Eigenschaften . . . . .	34
4.4	Erweiterung der Hausdorff-Distanz . . . . .	35
4.4.1	Partielle Hausdorff-Distanz . . . . .	35
4.4.2	Beschränkte Hausdorff-Distanz . . . . .	36
4.5	Transformation der Modellpunkte . . . . .	37
4.5.1	Translation . . . . .	38
4.5.2	Skalierung . . . . .	38
4.5.3	Affine Transformation . . . . .	38
4.6	Diskretisierung . . . . .	39
4.6.1	Punktkoordinaten . . . . .	39
4.6.2	Transformationsraum . . . . .	39
4.7	Vorgehensweise bei der Objektsuche . . . . .	40
4.7.1	Bestimmung der Objekthypothese . . . . .	41
4.7.2	Überprüfung der Objekthypothese . . . . .	41
4.8	Optimierung der Suche . . . . .	41
4.8.1	Vorzeitiges Verwerfen . . . . .	41
4.8.2	Überspringen von Transformationen . . . . .	42
4.8.3	Ausschließen von Kugelräumen . . . . .	42
4.8.4	Suchraumaufteilung . . . . .	42
4.9	Spezialisierte Suche . . . . .	43
4.9.1	Einzelnes Objekt . . . . .	43
4.9.2	Bestes Objekt . . . . .	43

<b>5</b>	<b>Auswahl der Verfahren</b>	<b>45</b>
5.1	Objekterkennungsstrategie . . . . .	45
5.2	Modelldatenbank . . . . .	46
5.3	Merkmalsextraktion . . . . .	46
5.3.1	Bildglättung . . . . .	47
5.3.2	Farbmodell . . . . .	47
5.3.3	Farbklassifizierung . . . . .	47
5.3.4	Kantenextraktion . . . . .	48
5.3.5	Kombination . . . . .	48
5.3.6	Distanztransformation . . . . .	49
5.4	Objektsuche . . . . .	50
<b>6</b>	<b>Beispielszenarios</b>	<b>59</b>
6.1	Szenario II . . . . .	59
6.2	Szenario III . . . . .	64
6.3	Szenario IV . . . . .	69
6.4	Szenario V . . . . .	74
<b>7</b>	<b>Entwurf und Implementierung</b>	<b>79</b>
7.1	Entwicklungsplattform . . . . .	79
7.2	Anforderungen . . . . .	79
7.3	Datenbeschreibung . . . . .	79
7.3.1	Struktur der Bilddaten . . . . .	80
7.3.2	Struktur der Modelldaten . . . . .	80
7.4	Datenablaufdiagramm . . . . .	81
7.5	Aspekte der Implementierung . . . . .	81
7.5.1	Aufteilung der Funktionalität . . . . .	81
7.5.2	Kodierung der zentralen Objekterkennungsroutine . . . . .	84
<b>8</b>	<b>Positionsbestimmung</b>	<b>85</b>
8.1	Bildgewinnung . . . . .	85
8.2	Rückgewinnung der Tiefeninformation . . . . .	86
8.3	Homogene Koordinaten . . . . .	86
8.3.1	Motivation . . . . .	86
8.3.2	Einführung . . . . .	87
8.3.3	Translation . . . . .	87
8.3.4	Skalierung . . . . .	87
8.3.5	Rotation . . . . .	87
8.3.6	Scherung . . . . .	88
8.3.7	Komposition von Transformationen . . . . .	88
8.3.8	Änderung des Koordinatensystems . . . . .	89
8.4	Kamerakalibrierung . . . . .	89
8.5	Objektreferenzpunkt . . . . .	90
8.5.1	Definition des Referenzpunktes . . . . .	90
8.5.2	Bestimmung des Referenzpunktes . . . . .	90
8.6	Berechnung der Position . . . . .	91
8.7	Genauigkeit des Verfahrens . . . . .	92
<b>9</b>	<b>Entwurf und Implementierung</b>	<b>93</b>
9.1	Entwicklungsplattform . . . . .	93
9.2	Anforderungen . . . . .	93
9.3	Ablaufszenario . . . . .	93
9.4	Realisierungskonzept . . . . .	94
9.4.1	Kommunikationsschema . . . . .	94

9.4.2	Iterationskreislauf . . . . .	95
9.5	Ablaufdiagramm der Positionsbestimmung . . . . .	95
9.6	Aufteilung der Funktionalität . . . . .	96
<b>10</b>	<b>Ausblick</b>	<b>99</b>
	<b>Literaturverzeichnis</b>	<b>101</b>

# Abbildungsverzeichnis

1	Das autonome System NOMAD . . . . .	vi
2.1	Der RGB-Farbraum . . . . .	8
2.2	Der HSB-Farbraum . . . . .	9
2.3	Draufsicht eines HSB-Farbraums . . . . .	9
2.4	Kantenausprägungen . . . . .	15
5.1	Die Modelldatenbank . . . . .	52
5.2	Szenario I – Originalaufnahme . . . . .	53
5.3	Der gesamte Prozeß der Merkmalsextraktion im Überblick . . . . .	54
5.4	Szenario I – Ergebnis der Farbklassifizierung . . . . .	55
5.5	Szenario I – Distanztransformation der Farbklassifizierung . . . . .	55
5.6	Szenario I – Kantenextraktion der Originalaufnahme . . . . .	56
5.7	Szenario I – Binarisiertes Kantenbild . . . . .	56
5.8	Szenario I – Verknüpfung von Farbklassifizierung und Kantenbild . . . . .	57
5.9	Szenario I – Distanztransformation des binarisierten Kantenbildes . . . . .	57
5.10	Szenario I – Originalaufnahme mit eingeblendetem Suchergebnis . . . . .	58
6.1	Szenario II – Originalaufnahme . . . . .	60
6.2	Szenario II – Ergebnis der Farbklassifizierung . . . . .	60
6.3	Szenario II – Distanztransformation der Farbklassifizierung . . . . .	61
6.4	Szenario II – Kantenextraktion der Originalaufnahme . . . . .	61
6.5	Szenario II – Binarisiertes Kantenbild . . . . .	62
6.6	Szenario II – Verknüpfung von Farbklassifizierung und Kantenbild . . . . .	62
6.7	Szenario II – Distanztransformation des binarisierten Kantenbildes . . . . .	63
6.8	Szenario III – Originalaufnahme . . . . .	65
6.9	Szenario III – Ergebnis der Farbklassifizierung . . . . .	65
6.10	Szenario III – Distanztransformation der Farbklassifizierung . . . . .	66
6.11	Szenario III – Kantenextraktion der Originalaufnahme . . . . .	66
6.12	Szenario III – Binarisiertes Kantenbild . . . . .	67
6.13	Szenario III – Verknüpfung von Farbklassifizierung und Kantenbild . . . . .	67
6.14	Szenario III – Distanztransformation des binarisierten Kantenbildes . . . . .	68
6.15	Szenario III – Originalaufnahme mit eingeblendetem Suchergebnis . . . . .	68
6.16	Szenario IV – Originalaufnahme . . . . .	70
6.17	Szenario IV – Ergebnis der Farbklassifizierung . . . . .	70
6.18	Szenario IV – Distanztransformation der Farbklassifizierung . . . . .	71
6.19	Szenario IV – Kantenextraktion der Originalaufnahme . . . . .	71
6.20	Szenario IV – Binarisiertes Kantenbild . . . . .	72
6.21	Szenario IV – Verknüpfung von Farbklassifizierung und Kantenbild . . . . .	72
6.22	Szenario IV – Distanztransformation des binarisierten Kantenbildes . . . . .	73
6.23	Szenario V – Originalaufnahme . . . . .	75
6.24	Szenario V – Ergebnis der Farbklassifizierung . . . . .	75

6.25	Szenario V – Distanztransformation der Farbklassifizierung . . . . .	76
6.26	Szenario V – Kantenextraktion der Originalaufnahme . . . . .	76
6.27	Szenario V – Binarisiertes Kantenbild . . . . .	77
6.28	Szenario V – Verknüpfung von Farbklassifizierung und Kantenbild . . . . .	77
6.29	Szenario V – Distanztransformation des binarisierten Kantenbildes . . . . .	78
7.1	Datenverlauf der gesamten Objekterkennung . . . . .	82
7.2	Modulstruktur des Objekterkennungsprozesses . . . . .	83
8.1	Das Modell der Lochkamera . . . . .	85
9.1	Iterationskreislauf . . . . .	96
9.2	Ablaufdiagramm der Positionsbestimmung . . . . .	97
9.3	Modulstruktur der Positionsbestimmung . . . . .	98

# Tabellenverzeichnis

8.1	Winkelauflösungen für den Weitwinkel- und Telebereich in $^{\circ}/Pixel$	90
-----	---	----



# Kapitel 1

## Einführung

In diesem Kapitel wird der Leser an das Thema Bildverarbeitung herangeführt. Es werden Gedanken vermittelt, die den Einstieg in diese Thematik erleichtern sollen. Außerdem werden die grundsätzlichen Probleme dargestellt, die sich mit der Verarbeitung von Abbildern aus der realen Welt ergeben.

Eine genaue Definition der Aufgabenstellung liefert die Motivation für diese Arbeit, deren Aufbau zur Orientierung in einem Überblick dargestellt ist.

### 1.1 Was bedeutet es zu sehen?

Sehen bedeutet, seine Umgebung visuell aufzunehmen, um daraus Rückschlüsse auf die Gegebenheiten in der Umwelt zu ziehen. Es vermittelt Wissen darüber, was sich in der aufgenommenen Szene befindet und wo es sich befindet.

Sehen ist dabei ein sehr komplexer Vorgang. Das Wissen über den Kontext einer Szene und die eigene Erfahrung helfen, das Abbild der Szene, beispielsweise in Vordergrund und Hintergrund, aufzuteilen. Diese Bildaufteilung ermöglicht es, die immense Informationsflut durch Abstraktion zu minimieren und das Wesentliche aus einem Bild zu extrahieren. Dabei spielt auch die Fähigkeit des Menschen, Regelmäßigkeiten, verdeckte Kanten, Gruppierungen von Objekten und vieles mehr zu erkennen, eine große Rolle. Auch die Konsistenzprüfung der gewonnenen Information trägt dazu bei, eine Szene korrekt interpretieren zu können.

Die Frage, wo sich in unserer drei-dimensionalen Welt etwas befindet, das wir gerade sehen, wird weitestgehend durch Stereoskopie gelöst. Stereoskopie bedeutet die Aufnahme einer Szene aus zwei unterschiedlichen Ansichten. Dabei ist es für den Menschen ein leichtes, zueinandergehörende Teile aus beiden Bildern miteinander zu verknüpfen und dadurch einen räumlichen Eindruck der gesamten Szene zu gewinnen.

### 1.2 Die Probleme für einen Computer

Sehen, ein für Menschen so selbstverständlicher Prozeß, ist für Computer eine anspruchsvolle Aufgabe, die viele Probleme mit sich bringt.

Um eine Szene in eine für den Computer zu verarbeitende Form zu bringen, wird ein perspektivisches Abbild angefertigt. In diesem Abbild ist jedoch keine Tiefeninformation enthalten. Es ist daher nicht möglich, aus diesem Bild die reale Szene zu rekonstruieren. Um eine im Bild festgehaltene Szene korrekt interpretieren zu können, ist entweder zusätzliche Information notwendig, oder es müssen im Vorfeld geeignete Annahmen getroffen werden.

Ein weiterer Aspekt sind die zahlreichen Einflüsse, die die Erscheinung einer Szene prägen. Die Oberflächenbeschaffenheit der einzelnen Objekte, der Einfallswinkel der direkten Lichteinstrahlung und die damit verbundene Schattenbildung, die Intensität des diffusen Lichtanteils und die Güte der Kamera sind nur einige Parameter, die das Abbild einer Szene beeinflussen. Dabei wirken alle Faktoren gleichzeitig auf jeden Punkt des Abbildes. Im nachhinein ist es nicht mehr möglich zu differenzieren, welche Parameter diesen Punkt des Abbildes maßgeblich beeinflusst haben.

Störungen bei der Abbildung einer Szene, wie sie durch Diskretisierung bei der Abtastung entstehen, können dazu führen, daß einzelne Teile der Szene nicht bzw. nur schwach abgebildet werden. Sind Elemente davon betroffen, die für die korrekte Erkennung eines Objekts zwingend notwendig sind, so schlägt die Erkennung fehl, obwohl das Objekt in der Szene vorhanden ist.

Ebenfalls nicht zu vernachlässigen ist, daß Bilder über eine enorme Informationsdichte verfügen. Die Bearbeitung solch großer Datenmengen bedarf leistungsfähiger Hardware.

## 1.3 Klassifizierung von Bildverarbeitung

Der Begriff Bildverarbeitung findet in der Informatik in vielfältiger Weise Verwendung. Dabei sind die einzelnen Teilgebiete in ihrer Methodik und Komplexität sehr verschieden. Eine feinere Klassifizierung, die auf die Unterschiede eingeht, ist daher angebracht.

### 1.3.1 Digitale Bildverarbeitung

Digitale Bildverarbeitung ist die Umformung von Bildern in Bilder. Dabei ist der Inhalt dieser Bilder für die Bearbeitung oft ohne Belang. Sie wird verwendet, um die Qualität von Bildern zu steigern, besondere Eigenschaften in Bildern hervorzuheben oder die Darstellungsform der Bilddaten neuen Bedürfnissen anzupassen. Beispiele dafür sind die Bildglättung zur Rauschminderung, die Bildsegmentierung, die Kantenextraktion sowie die Fouriertransformation. Methoden der digitalen Bildverarbeitung werden in Kapitel 2 beschrieben.

### 1.3.2 Mustererkennung

Mustererkennung ordnet Bilder vordefinierten Klassen zu. Es werden relevante Teile aus einem Bild extrahiert und einer statistischen Untersuchung unterzogen. Darauf basierend wird entschieden, ob dieses Bild einer vorgegebenen Klasse angehört oder nicht. Die Entwicklung von optimalen Entscheidungsregeln ist dabei ein zentrales Thema. Ein Beispiel hierfür ist die Schrifterkennung. Das Teilgebiet Mustererkennung wird in dieser Abhandlung nicht weiter vertieft.

### 1.3.3 Objekterkennung

Im Gegensatz zur Mustererkennung arbeitet die Objekterkennung im drei-dimensionalen Raum. Für die Objekterkennung ist es hilfreich, im Vorfeld Aussagen über die zu erwartende Szene machen zu können. Zusätzlich ist Kontextinformation in impliziter oder expliziter Form unbedingt notwendig, um den Inhalt einer Szene korrekt interpretieren zu können. Das Bild, ein perspektivisches Abbild der realen Szene, wird vorher in der Regel einer aufwendigen Bildaufbereitung unterzogen, denn störende Einflüsse, wie Rauschen oder unausgeglichene Lichtverhältnisse, können sich ungünstig auf die Analyse der Szene auswirken. In Kapitel 3 wird auf

die Objekterkennung im allgemeinen, in Kapitel 4 speziell auf die Verwendung der Hausdorff-Distanz zur Objekterkennung eingegangen.

## 1.4 Erläuterung der Aufgabenstellung

### 1.4.1 Kontext der Aufgabe

Das globale Ziel des Projektes DAVID ist es, in Koordination mit dem Projekt AMOS eine vollautomatische Entleerung aller im Bürogebäude befindlicher Papierkörbe zu realisieren. Dazu wurden die Aufgabe folgendermaßen aufgeteilt.

Das autonome System NOMAD durchsucht selbständig jeden Büroraum nach Papierkörben, die beliebig auf dem Fußboden aufgestellt sein können, und befördert diese an eine vordefinierte Position auf dem Gang. Der Serviceroboter AMOS übernimmt den Papierkorb, entleert diesen und stellt ihn anschließend in den Gang zurück. Von dort wird der Papierkorb durch das NOMAD-System an die Stelle im Büro zurückgebracht, an der er ursprünglich aufgegriffen wurde.

Die Verteilung der Aufgabe auf zwei Robotersysteme beruht auf den unterschiedlichen Eigenschaften von NOMAD und AMOS. Das autonome System NOMAD ist kompakt gebaut und verfügt über einen Antrieb, der eine hohe Wendigkeit und Manövrierfähigkeit erlaubt. Es ist daher für den Einsatz in engen und verwinkelten Räumen besonders geeignet. Seine Zuladung ist jedoch begrenzt. Der Serviceroboter AMOS hingegen baut auf einer gängigen Industriepattform auf. Diese ermöglicht eine hohe Ladekapazität sowie die Bewältigung langer Strecken mit hohem Tempo. Durch seine Antriebsart und die ausladenden Maße ist die Bewegung in engen und verwinkelten Räumen jedoch schwierig oder sogar unmöglich.

### 1.4.2 Konkrete Aufgabenstellung

Ziel dieser Arbeit ist es, für das NOMAD-System die Objekterkennung zu entwickeln, mit der die Büroumgebung nach Papierkörben durchsucht wird. Dabei ist Form, Farbe und Lage der Papierkörbe festgelegt, die von dem System erkannt werden sollen. Abbildung 5.2 auf Seite 53 zeigt einen solchen Papierkorb; in einem Farbbild wäre dieser rot dargestellt. Da vom System nur aufrecht stehende Papierkörbe gegriffen werden können, ist auch die Erkennung nur dieser ausreichend.

Ist ein Papierkorb gefunden, so muß seine Position im Raum bestimmt werden, damit dieser vom Roboter angefahren und gegriffen werden kann. Auch dafür präsentiert diese Arbeit einen Lösungsweg.

Zur Bildgewinnung stellt das System eine CCD-Farbkamera mit automatischer Scharfstellung zur Verfügung. Sie ist auf ein fernsteuerbares Stativ montiert, daß ein Neigen und Schwenken erlaubt. Eine Framegrabberkarte ermöglicht das Auslesen und Speichern des Kamerabildes.

## 1.5 Überblick

Kapitel 2 erklärt Grundlegendes zum Thema Digitale Bildverarbeitung und stellt für die hier vorgestellte Arbeit relevante Bildverarbeitungsoperatoren vor. Für jede Kategorie wird eine Auswahl an Verfahren erläutert, so daß ein Vergleich möglich wird. In Kapitel 3 werden Grundlagen zum Thema Objekterkennung vermittelt und einige gängige Verfahren skizziert. Daurch soll die getroffene Entscheidung zur Auswahl der Objekterkennungsstrategie nachvollziehbar werden. Die Objektsuche mit Hilfe der Hausdorff-Distanz wird in Kapitel 4 detailliert vorgestellt. Dieses Verfahren beruht auf einer anderen Sichtweise des Problems Objekterkennung und bietet

wesentliche Vorteile gegenüber den in Kapitel 3 beschriebenen Verfahren. Der entwickelte Objekterkennungsprozeß wird in Kapitel 5 anhand eines Beispiels vorgestellt. Die Auswahl der einzelnen Werkzeuge wird erläutert. Kapitel 6 zeigt weitere Beispiele und wertet die Teilergebnisse. Kapitel 7 stellt das Design und den Aufbau der Objekterkennung dar. In Kapitel 8 sind Grundlagen und Definitionen, sowie die notwendigen Schritte zur Bestimmung der gesuchten Position zusammengefaßt. In Kapitel 9 sind die internen Mechanismen sowie die Programmstruktur der Positionsbestimmung dargestellt. Kapitel 10 schließlich gewährt einen Ausblick auf Ansätze zur Verbesserung der in dieser Arbeit gezeigten Problemlösung.

## Kapitel 2

# Digitale Bildverarbeitung

In diesem Kapitel werden die Grundlagen der digitalen Bildverarbeitung erörtert und die gängigsten Bildverarbeitungsstrategien im einzelnen vorgestellt. Für jede Kategorie ist eine Auswahl an Verfahren dargestellt. Dadurch bietet sich eine Vergleichsmöglichkeit über Vor- und Nachteile der einzelnen Operatoren. Die Entscheidung, welche Verfahren für die in Kapitel 5 dargestellte Objekterkennung verwendet wurden, wird somit nachvollziehbar.

### 2.1 Einleitung

Die digitale Bildverarbeitung ist jene Teildisziplin auf dem Gebiet der Bildbearbeitung, die sich mit der Umwandlung von Bilddaten beschäftigt. Ziel ist das Erstellen von Histogrammen, die Verbesserung der Bildqualität, die Hervorhebung von Bildmerkmalen sowie beispielsweise die Segmentierung eines Bildes.

Die digitale Bildverarbeitung gehört dabei zu den am besten erforschten Gebieten im Bereich der Bildbearbeitung. Es existieren zahlreiche Standardverfahren, die im Fortgang dieses Kapitels näher erläutert werden.

Digitale Bildverarbeitung wird angewendet, wenn Bilder vom Menschen interpretiert werden. Dabei läßt die vorherige Bearbeitung der Bilder mit Computern eine veränderte Darstellung zu, die neue Betrachtungsweisen ermöglicht und Rückschlüsse zuläßt. Typisches Beispiel ist die Fouriertransformation. Mit ihrer Hilfe läßt sich ein Frequenzspektrum von einer Darstellungsweise in eine andere transformieren. Wird das Frequenzspektrum als Summe aller Frequenzamplituden an einer Zeitachse angetragen, so zeigen sich Auffälligkeiten, die bei einer Aufschlüsselung nach Wellenlängen nicht zu erkennen sind. Tatsächlich stellt ein digitalisiertes Bild auch ein Frequenzspektrum dar. Die Fouriertransformierte eines Bildes wird oft verwendet, um eine Klassifizierung der Eigenschaften von Musterungen im Bild vorzunehmen.

Ein zweites großes Anwendungsgebiet der digitalen Bildverarbeitung ist die automatisierte Bildanalyse, die von Computern durchgeführt wird. Dabei ist es oft erforderlich, das Eingabebild für die Bearbeitung durch den Computer aufzubereiten. Zum einen zählen dazu qualitative Aspekte, wie Rauschminderung und das Hervorheben von wesentlichen Strukturen. Zum anderen muß das Eingabebild so verändert werden, daß es den Anforderungen für einen Analysealgorithmus genügt. Beispiele hierfür sind die Bildsegmentierung und die Kantenextraktion.

## 2.2 Grundlagen

### 2.2.1 Darstellungsformen von Bildern

#### Allgemeine Anmerkung

Um den Inhalt einer Szene in eine für Bildverarbeitungsalgorithmen verarbeitbare Form zu bringen, wird durch Projektion ein Abbild der Szene erstellt. Dieses Abbild enthält kontinuierliche Farbwerte. Die Information, die eine Abbildung enthält, sind eben diese Farbwerte mit ihrer jeweiligen Position im Bild. Um dieses Bild von einem Computer analysieren zu lassen, müssen die Farbwerte und ihre Position diskretisiert werden, d. h. ein Raster für Farbe und Position wird eingeführt. Die Anzahl der zur Verfügung stehenden Farbabstufungen nennt man Quantisierungsstufen, die Positionsrauerung wird als Auflösung bezeichnet. In der digitalen Bildverarbeitung ist die Abstufung in 256 Graustufen üblich. Es sind jedoch auch Abstufungen mit 32, 64, 128 oder 512 Grauwerten gängig. Die Quantisierungsstufen sind dabei in der Regel Zweierpotenzen, angepaßt an die binäre Arbeitsweise von Computern. Eine besondere Farbrasterung stellt das binäre Bild dar, das lediglich zwei Farbstufen, Schwarz und Weiß, zur Verfügung stellt. Diese Sonderform hat historische Gründe. Die Bearbeitung von großen Datenmengen, wie sie in der Bildverarbeitung anfallen, erforderte damals wie heute vernünftiges Haushalten mit den zur Verfügung stehenden Ressourcen, wie Rechenleistung und Speicherkapazität. Ein weiterer Grund, der auch heute noch die Nutzung von Binärbildern rechtfertigt, ist die Fähigkeit des Menschen, Strichzeichnungen, Silhouetten und andere schwarz-weiße Abbildungen zu interpretieren. Außerdem ist die Bearbeitung von Binärbildern heutzutage ein gut erforschtes Gebiet.

#### Mathematisches Modell

Digitalisierte, also diskretisierte Bilder sind zweidimensionale Objekte, bei denen sich der Helligkeitswert von Punkt zu Punkt ändert. Mathematisch ausgedrückt ist  $f(x, y)$  der Grauwert eines Bildes im Punkt  $(x, y)$ . Dabei sind die Ortskoordinaten  $x$  und  $y$  beschränkt und üblicherweise nicht negativ. In der Regel besitzt die Funktion  $f(x, y)$  eine Fouriertransformierte und läßt sich integrieren. Dies sind nicht alle, jedoch in der digitalen Bildverarbeitung die wichtigsten Eigenschaften für ein zu bearbeitendes Bild. Modell (2.1) beschreibt die Struktur eines diskretisierten Bildes.

$$\begin{aligned}
 G &= \{0, 1, 2, \dots, \text{Graustufen} - 1\} \\
 X &= \{0, 1, 2, \dots, \text{Spalten} - 1\} \\
 Y &= \{0, 1, 2, \dots, \text{Zeilen} - 1\} \\
 f(x, y) &= g \\
 &\quad \text{mit } x \in X, y \in Y \text{ und } g \in G \\
 F &= (f(x, y)) \tag{2.1}
 \end{aligned}$$

Dabei ist  $G$  die Menge der Graustufen,  $X$  und  $Y$  der gültige Koordinatenraum und  $F$  das gesamte Bild. Für  $n$ -kanalige Bilder, bei Farbbildern üblicherweise  $n = 3$ , wird Modell (2.1) mit (2.2) erweitert.

$$F = \begin{pmatrix} f_0(x, y) \\ f_1(x, y) \\ \vdots \\ f_{n-1}(x, y) \end{pmatrix} \tag{2.2}$$

### 2.2.2 Definition von Farbe

Große Teile der digitalen Bildverarbeitung bauen auf der Bearbeitung monochromer Bildinformation auf. Dabei ist die Unterscheidung verschiedener Farben für den Menschen ein wichtiges Werkzeug bei der Interpretation von Bildern. Die zusätzliche Farbinformation kann also auch bei der Bildverarbeitung ein entscheidender Aspekt sein.

Drei Grundfarben sind ausreichend, um durch Mischen jede beliebige Farbe herzustellen. Die Auswahl der drei Grundfarben ist beliebig, sofern keine davon durch Mischen der beiden anderen hergestellt werden kann. Diese Erkenntnis kann dahingehend erweitert werden, daß grundsätzlich drei Parameter ausreichend sind, um eine Farbe eindeutig zu identifizieren. In der Vergangenheit wurden daher zahlreiche Modelle zur Farbdarstellung entwickelt, die unterschiedliche Betrachtungsweisen bezüglich der Farbe zulassen. Im folgenden werden hier vier der wichtigsten Modelle vorgestellt.

#### Das RGB-Farbmodell

Das RGB-Modell basiert auf der additiven Farbmischung der drei Grundfarben Rot, Grün und Blau, die dem Modell seinen Namen geben. Additive Farbmischung bedeutet, die Summe aller Beiträge einer jeden Grundfarbe ergibt die Mischfarbe. Technisch ausgedrückt werden zum Frequenzspektrum der Farbe Schwarz<sup>1</sup> die Spektralanteile der Farben Rot, Grün und Blau addiert. Es ergibt sich die gewünschte Mischfarbe.

Dargestellt wird der RGB-Farbraum durch einen sogenannten Farbwürfel, wie ihn Abbildung 2.1 zeigt. Dabei wird jeder der drei Dimensionen eine Grundfarbe zugeordnet. Angetragen in einem Wertebereich von 0 bis 1 ergeben sich Punkte im Raum, die die darzustellende Farbe repräsentieren. Sämtliche Grauwerte liegen auf der Diagonale des Würfels zwischen Schwarz (0, 0, 0) und Weiß (1, 1, 1).

Die Braunsche Röhre verwendet diese Darstellung zur Erzeugung von Farbbildern.

#### Das CMY-Farbmodell

Das CMY-Farbmodell ist das exakte Gegenteil des RGB-Farbraums. Es basiert auf der subtraktiven Farbmischung von Cyan, Magenta und Gelb (engl. Yellow), den Komplementärfarben zu Rot, Grün und Blau. Aus dem vollen Farbspektrum der Farbe Weiß werden die Farbanteile von Cyan, Magenta und Gelb herausgefiltert. Es bleibt die gewünschte Mischfarbe. Die Überführung des RGB-Farbmodells in das CMY-Farbmodell und umgekehrt ist einfach.

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (2.3)$$

Das subtraktive Farbmodell hat in der digitalen Bildverarbeitung kaum Bedeutung. Vielfache Anwendung findet es in der Drucktechnik. Tintenstrahldrucker verwenden Cyan, Magenta und Gelb, um nichtbenötigte Farbanteile aus der weißen Farbe des Papiers zu filtern.

#### Das YUV-Farbmodell

Ein weiteres Farbmodell stellt das YUV-Modell dar. Es gliedert die Farbe in Luminanz (Y) und Chrominanz (U und V) auf. Dabei stellt die Luminanz nur die Helligkeitsinformation des Bildes zur Verfügung. Die Chrominanz, verantwortlich für die

<sup>1</sup>Für alle Frequenzen im Frequenzspektrum der Farbe Schwarz ist die Amplitude gleich 0.

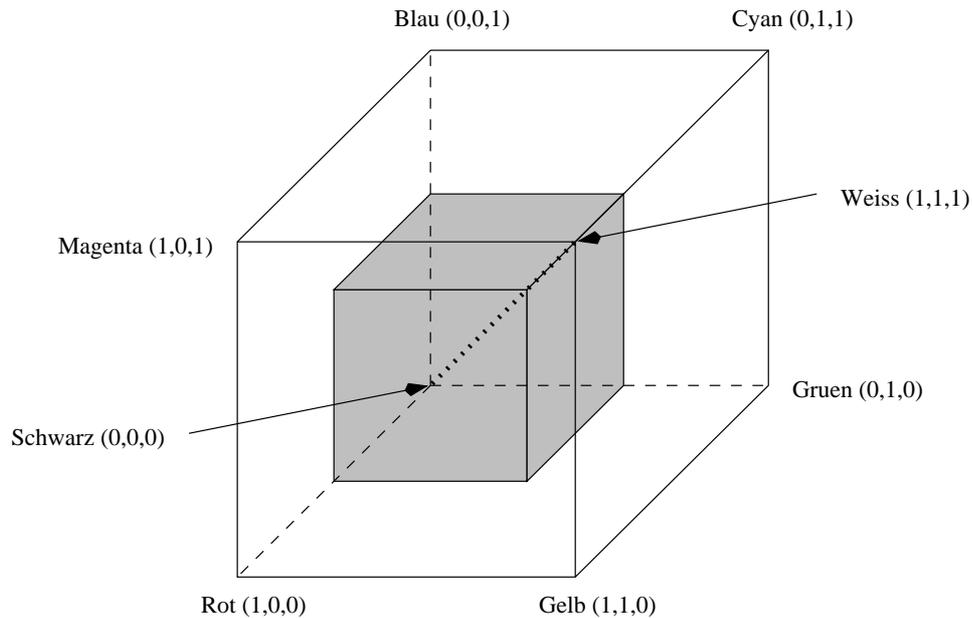


Abbildung 2.1: Der RGB-Farbraum mit eingeblenndem RGB-Subraum

Farbinformation, ist aufgeteilt in zwei Kanäle. Der U-Kanal liefert die Rot-Grün-Information, der V-Kanal die Blau-Gelb-Information für das Bild. Werden U-Kanal und V-Kanal nicht ausgewertet, so erhält man ein Grauwertbild, vergleichbar dem Schwarz-Weiß-Fernsehen.

In der Fernsehtechnik hat das YUV-Farbmodell auch eines seiner Hauptanwendungsgebiete. Sowohl die amerikanische Fernsehnorm NTSC (National Television System Comittee), wie auch die westeuropäische Norm PAL (Phase Alternating Line) bauen auf diesem Prinzip auf. Hiermit ist es möglich, sowohl Grauwertbilder als auch Farbbilder mit einer einzigen Norm zu übertragen. Auch Framegrabber-Karten, die das von der Kamera aufgenommene Bild für den Computer zugänglich machen, arbeiten häufig ausschließlich mit dem YUV-Format.

Ein Hauptmotiv für die Entwicklung dieses Farbmodells lag in der Anpassung der Farbdarstellung an das menschliche Sehen. Das Auge hat seperate Rezeptoren für Helligkeit und Farbe. Es kann Helligkeitsunterschiede wesentlich feiner auflösen als Änderungen in der Farbe. Das oft gebräuchliche 4:2:2-Format trägt dieser Beobachtung Rechnung. Für die Speicherung von vier Bildpunkten werden alle vier Helligkeitswerte, jedoch nur jeweils zwei Farbwerte verwendet. Dies bedeutet eine Platzersparnis von 66% ohne nennenswerten Qualitätsverlust.

Die Umrechnung von RGB-Daten in das YUV-Format beschreibt Gleichung (2.4).

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (2.4)$$

### Das HSB-Farbmodell

Im Gegensatz zu den bisher beschriebenen Farbmodellen, bei deren Definition der Bezug zur Hardware im Vordergrund stand, orientiert sich das HSB-Modell an den menschlichen Gewohnheiten. Der Mensch pflegt Farbe in Farbton, Sättigung und Intensität (engl. Hue, Saturation und Brightness) zu klassifizieren.

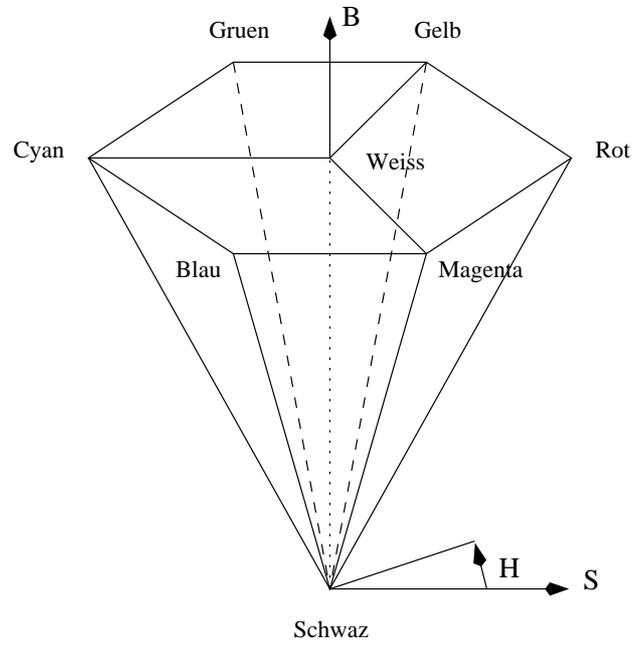


Abbildung 2.2: Der HSB-Farbraum

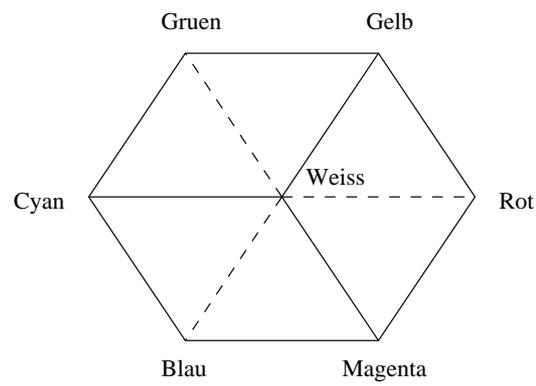


Abbildung 2.3: Draufsicht eines HSB-Farbraums

Der HSB-Farbraum (siehe Abb. 2.2) ist durch ein drei-dimensionales Polarkoordinatensystem definiert. Der gültige Wertebereich beschreibt dabei eine sechsseitige Pyramide. Der Farbton wird durch den Winkel um die Längsachse  $B$  angetragen, die Sättigung als Abstand von der Längsachse nach außen und die Intensität entlang der Längsachse. Auf der Längsachse selbst liegen alle verfügbaren Grauwerte, wobei Schwarz bei  $B = 0$  und Weiß bei  $B = 1$  liegen. Für  $B = 0$  ist  $H$  und  $S$  undefiniert.

Die Draufsicht, wie sie in Abbildung 2.3 dargestellt ist, ist die Projektion des RGB-Farbraums (siehe Abb. 2.1). Entlang der Längsachse entspricht jeder Querschnitt der Ansicht eines RGB-Subraums.

## 2.3 Klassifizierung von Bildoperatoren

Ein Bildoperator formt ein Eingabebild in ein Ausgabebild um. Dabei kann die Anwendung eines Operators auf eine Teilmenge der Bilder beschränkt sein. Solche Teilmengen sind Binärbilder, Grauwertbilder oder auch Mehrkanalbilder. Alle Algorithmen in der digitalen Bildverarbeitung gehören einer der im folgenden dargestellten Klassen an.

### 2.3.1 Punktoperatoren

Punktoperatoren betrachten jeden Punkt des Eingabebildes isoliert, d. h. der entsprechende Bildpunkt im Ausgabebild ist ausschließlich von  $(x, y, f(x, y))$  des Eingabebildes abhängig. Ein typischer Vertreter der Punktoperatoren ist die Schwellwertbildung, auch Thresholding genannt.

### 2.3.2 Lokale Operatoren

Ein lokaler Bildoperator erzeugt ein Ausgabebild, dessen Bildpunkte durch benachbarte Punkte des entsprechenden Eingabebildpunktes beeinflusst wurden. Dies wird häufig dadurch erreicht, daß eine Matrix über das Eingabebild bewegt wird und alle Punkte innerhalb der Matrix entsprechend bewertet werden. Gängige Größen für solche Matrizen sind  $3 \times 3$  und  $5 \times 5$ . Ein einfaches Beispiel hierfür ist die  $3 \times 3$ -Matrix des Prewitt-Operators zur Kantenextraktion.

$$\mathcal{M} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \implies$$

$$f'(x, y) = \begin{pmatrix} -f(x-1, y-1) & + & 0 & + & f(x+1, y-1) & + \\ -f(x-1, y) & & + & 0 & + & f(x+1, y) & + \\ -f(x-1, y+1) & + & 0 & + & f(x+1, y+1) & \end{pmatrix} \quad (2.5)$$

Problematisch bei der Bearbeitung eines Bildes mit lokalen Operatoren sind die Randpunkte. Diese verfügen über weniger Nachbarn als Punkte im Inneren des Bildes. Eine Lösung ist, die Randpunkte unberücksichtigt zu lassen, was zur Folge hat, daß das resultierende Bild gegenüber dem Eingabebild kleiner ist. Eine zweite Möglichkeit besteht darin, die Randpunkte über die Bildgrenzen hinaus zu spiegeln, um eine Kontinuität der Bilddaten zu erreichen. Dieser aufwendigere Ansatz produziert ein Ausgabebild identischer Größe. Gerade bei wiederholter Bearbeitung des Bildes mit lokalen Operatoren oder bei kleinen Bildern kann es von Vorteil sein, die Bildgröße zu erhalten.

### 2.3.3 Globale Operatoren

Die Arbeitsweise globaler Operatoren bezieht für die Berechnung eines Ausgabebildpunktes beliebig viele oder alle Punkte des Eingabebildes ein. Im Vorfeld kann oft

keine Angabe darüber gemacht werden, welche Punkte der Eingabe für einen Punkt der Ausgabe verantwortlich sind. Der Extremfall ist: jeder Punkt beeinflusst jeden. Die Bearbeitung von Bildern mittels globaler Operatoren ist rechenzeitaufwendig und speicherintensiv. Oft existieren besondere Prozessoren, die spezielle Aufgaben besonders effizient ausführen. Beispiele für globale Operatoren sind die Fouriertransformation und die Berechnung von Voronoi-Diagrammen und Histogrammen.

### 2.3.4 Objektorientierte Operatoren

Objektorientierte Operatoren sind die komplexeste Form von Bildoperatoren. Sie stellen das stärkste Bindeglied zur Objekterkennung dar. Die Auswertung eines Bildes erfolgt bereits objektorientiert. Oft ist es notwendig, Aussagen über die Charakteristik nur der Bildpunkte zu machen, die einem spezifischen Objekt angehören. Andererseits ist aber die Kenntnis über die Charakteristik der Bildpunkte eines einzelnen Objekts nötig, um diese zuverlässig von den übrigen Bildpunkten abzugrenzen (*hier beißt sich die Katze oft in den Schwanz*).

Ein verwandtes Gebiet ist die Bildsegmentierung, mit deren Hilfe Bilder in Vorder- und Hintergrund oder in mehrere elementfremde Objekte aufgeteilt werden. Die mittels objektorientierter Operatoren gewonnenen Informationen, wie Größe oder mittlere Helligkeit, bilden oft die Grundlage für eine Objekterkennung. Auch können Merkmale dieser Objekte verwendet werden, um Fehler im Bild zu erkennen, ohne deren Kenntnis eine korrekte Erkennung des gesuchten Objekts schwer oder gar unmöglich wäre.

## 2.4 Bildglättung

Abbilder einer zu analysierenden Szene enthalten oft Störungen. Der Mensch besitzt die Fähigkeit, wesentliche Aspekte des Bildes von solchen Störungen zu differenzieren. Algorithmen der digitalen Bildverarbeitung reagieren jedoch empfindlich auf solche Einflüsse, da sie keine Kenntnis über die Bedeutung des Bildinhaltes haben. Das Bild wird nach einer Vorschrift, Punkt für Punkt, abgearbeitet. Besonders kantenextrahierende Operationen auf verrauschte Daten erzeugen oft Ergebnisse, die für eine weitere Bearbeitung ungeeignet sind. Eine vorherige Bildglättung ist hier erforderlich. Dies wird durch spezielle Filter, in der Regel Tiefpaßfilter, erreicht, von denen einige im folgenden kurz vorgestellt werden.

### 2.4.1 Mittelwertfilter

Eines der einfachsten Glättungsfilter ist das Mittelwertfilter, wie es in Gleichung (2.6) beispielhaft für ein  $3 \times 3$ -Filter dargestellt ist. Dem aktuellen Bildpunkt wird dabei der Mittelwert aus seinem Grauwert und denen seiner Nachbarn zugeordnet (siehe Gleichung (2.7)). Der Wirkungsbereich hängt allein von der Größe der Filtermatrix ab. Je größer die Matrix, desto unschärfer das Ergebnis, d. h. feine Details verschwinden mehr und mehr. Das Mittelwertfilter hat die Eigenschaft, Kanten im Bild zu verschmieren und erzeugt durch die Mittelung Grauwerte, die unter Umständen im Bild vorher nicht vorhanden waren.

$$\mathcal{F} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.6)$$

$$f'(x, y) = \frac{1}{9} \sum_{x'=x-1}^{x+1} \sum_{y'=y-1}^{y+1} f(x', y') \quad (2.7)$$

Eine Erweiterung des Mittelwertfilters kann durch verschiedene Gewichtungen erfolgen, wie Gleichung (2.8) beispielhaft zeigt. Bei der Verteilung der Gewichtungen sollte jedoch auf die Symmetrie des Filters, sowohl in horizontaler als auch in vertikaler Richtung, geachtet werden.

$$\mathcal{F} = \begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix} \quad (2.8)$$

### 2.4.2 Medianfilter

Ein Hauptproblem des Mittelwertfilters ist, daß er das Bild zwar glättet, jedoch zusätzlich auch verschleiert. Das Medianfilter besitzt in dieser Beziehung eine wesentlich günstigere Charakteristik. Aus den Grauwerten der benachbarten Punkte wird nicht der Mittelwert gebildet, sondern die Grauwerte werden der Größe nach sortiert und der mittlere, der sogenannte Median, als neuer Wert dem Punkt zugewiesen. Wesentliche Eigenschaften des Medianfilters sind:

- Eine monoton ansteigende Rampe im Eingabebild wird unverfälscht in das Ausgabebild übernommen.
- Eine Kante zwischen zwei Bereichen konstanter Grauwerte wird nicht verschmiert.
- Einzelne gestörte Bildpunkte werden eliminiert, ohne die umliegenden Punkte wesentlich zu verfälschen.
- Bereiche gleicher Graufärbung sowie Kanten gelten als konstante Bereiche und bleiben bei der Medianfilterung bestehen.

Besonders gut ist das Medianfilter geeignet, fehlerhafte Punkte eines Bildes zu entfernen.

$$\begin{array}{ccccccc} & & & & 19 & & \\ & & & & 21 & & \\ & \vdots & \vdots & \vdots & 22 & \vdots & \vdots & \vdots \\ \dots & 22 & 27 & 24 & \dots & \dots & 22 & 27 & 24 & \dots \\ \dots & 28 & \boxed{56} & 19 & \dots & \Rightarrow & \boxed{24} & \Rightarrow & \dots & 28 & \boxed{24} & 19 & \dots \\ \dots & 23 & 25 & 21 & \dots & & 25 & & \dots & 23 & 25 & 21 & \dots \\ & \vdots & \vdots & \vdots & 27 & & \vdots & & \vdots & \vdots & \vdots \\ & & & & 28 & & & & & & & & \\ & & & & 56 & & & & & & & & \end{array} \quad (2.9)$$

### 2.4.3 Gaußsches Glättungsfilter

Beim Gaußfilter sind die Gewichtungen für die Glättungsmatrix von der Gaußschen Glockenkurve abgeleitet. Das Gaußfilter ist gut geeignet, um normalverteiltes Rauschen in Bilddaten zu reduzieren. Die Gleichung für eine ein-dimensionale Gaußsche Glockenkurve lautet wie folgt:

$$g(x) = e^{-\frac{x^2}{2\sigma^2}} \quad (2.10)$$

Dabei bestimmt  $\sigma$  die Breite der Glockenkurve. Für die Bearbeitung von Bilddaten wird jedoch die zwei-dimensionale Glockenkurve verwendet, die sich folgendermaßen berechnet:

$$g(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.11)$$

Mit Hilfe des Parameters  $\sigma$  wird der Wirkungsgrad des Gaußfilters bestimmt. Ein großes  $\sigma$  erzeugt eine breitere Glockenkurve und damit eine stärkere Glättung des Bildes, ein verkleinertes  $\sigma$  analog eine schmalere Glockenkurve und somit eine geringere Glättung.

#### 2.4.4 Größenfilter

Bildglättung bei Grauwertbildern wird benutzt, um störende Einflüsse zu reduzieren, die die Weiterverarbeitung dieser Daten erschweren. Bei diesen Störungen handelt es sich in der Regel um Rauschen verschiedener Art. Solch störende Einflüsse können jedoch auch andere Ursachen haben. Bei der Kantenextraktion mit anschließendem Thresholding enthält das entstandene Binärbild oft Segmente, die im Originalbild keiner Kante zugeordnet werden können, sondern eher durch Verfälschungen im Eingabebild entstehen. Diese Segmente sind üblicherweise im Verhältnis zu den korrekten Kantensegmenten klein. Für eine robuste Objekterkennung ist es hilfreich, solche Segmente zu eliminieren. Ziel ist es, die kleinen Störungen aus dem Bild ‚herauszusieben‘. Dazu wird die Größe der Kantensegmente als Kriterium verwendet. Ein Schwellwert wird festgelegt und alle Kantensegmente, deren Größe diesen Schwellwert unterschreiten, werden aus dem Bild entfernt. Entstanden ist ein minder gestörtes und somit ‚geglättetes‘ Bild.

## 2.5 Segmentierung durch Klassifizierung

Die Klassifizierung von Bilddaten ist eine Form der Bildsegmentierung. Sie stellt dabei ein Bindeglied zwischen digitaler Bildverarbeitung und der Mustererkennung dar. Mehrdimensionale Bilder, z. B. Farbbilder, sind für die Bildklassifizierung besonders gut geeignet. Die Segmentierung eines Bildes mittels Klassifizierung beruht auf der Existenz von Merkmalräumen. Bildpunkte werden aufgrund dieser Merkmale sogenannten Musterklassen zugeordnet. Für die Bestimmung der Merkmalräume existieren im wesentlichen vier unterschiedliche Verfahren.

### 2.5.1 Fest dimensionierte überwachte Klassifizierung

Ist es a priori möglich, durch eine repräsentative Lernstichprobe, beispielsweise mittels Testszenarien, den Merkmalraum zu definieren, so spricht man von einer überwachten Klassifizierung.

Bei der überwachten Klassifizierung kann der Merkmalraum in der Regel jedoch nur näherungsweise bestimmt werden. Eine Fehlinterpretation der Lernstichprobe und eine ungenügend große Stichprobe sind zwei Hauptfehlerquellen.

#### Fehler erster Art

Als Fehler erster Art wird bezeichnet, wenn der durch die Stichprobe definierte Merkmalraum gegenüber dem tatsächlichen Merkmalraum des Objekts eingeschränkt ist. Dies hat zur Folge, daß einige Bildpunkte des Objekts als diesem nicht zugehörig klassifiziert werden.

#### Fehler zweiter Art

Ist der durch die Stichprobe definierte Merkmalraum dem tatsächlichen Merkmalraum des Objekts gegenüber erweitert, so spricht man von einem Fehler zweiter Art. Objektfremde Bildpunkte werden als dem Objekt zugehörig klassifiziert.

### 2.5.2 Fest dimensionierte unüberwachte Klassifizierung

Ist – anders als bei der überwachten Klassifizierung – im Voraus nicht bekannt, welche Klassen das Bild enthält, so muß eine zur Bearbeitungszeit dynamische Definition der Merkmalsräume erfolgen.

Ausgehend von einem ersten Bildpunkt werden, meist iterativ, Merkmalsräume geschaffen und gleichzeitig Bildpunkte durch bereits bestehende Merkmalsräume verschiedenen Klassen zugeordnet. Sind alle Bildpunkte klassifiziert, so werden zu kleine Punkthäufungen mit benachbarten verschmolzen und zu große weiter unterteilt.

Nachteil dieser Klassifizierungsstrategie ist, daß die Bearbeitungsreihenfolge das Endergebnis beeinflußt. Ein bereits bearbeiteter Bildpunkt würde nach der Definition eines neuen Merkmalsraumes einer anderen Klasse zugeordnet, als das ursprünglich der Fall war.

### 2.5.3 Lernende Klassifizierung

Bei der fest dimensionierten Klassifizierung werden bestehende Merkmalsräume nach ihrer Definition nicht mehr geändert. Motiviert durch eine wiederkehrende Bearbeitung von Bilddaten aus einer sich stetig ändernden Umgebung paßt die lernende Klassifizierung existierende Merkmalsräume den äußeren Gegebenheiten an. Die Unterteilung in überwachte und unüberwachte Klassifizierung ist auch hier zutreffend, da unterschiedliche Verfahren für die Gewinnung der initialen Merkmalsräume verwendet werden.

## 2.6 Kantenextraktion

Die Kantenextraktion ist für die Objekterkennung ein wichtiger Prozeß. Kanten in Bildern sind abrupte Änderungen der Intensität. Sie entstehen an Grenzen von Objekten oder – allgemeiner Ausgedrückt – an Grenzen zweier benachbarter Flächen und sind somit wichtige Merkmale für die Bildanalyse.

Ideale Ausprägungen von Kanten sind Stufen und Linien (siehe Abb. 2.4), diese kommen in der Realität jedoch kaum vor. Häufiger sind dagegen Rampen und Dächer, die, im Gegensatz zu den steilen Flanken von Stufen und Linien, eine abgeschwächte, kontinuierliche Änderung der Intensität darstellen. Ursache dafür sind in der Regel eine vorher durchgeführte Bildglättung, um die Detektion von falschen Kanten zu minimieren oder Ungenauigkeiten bei der Abbildung der Szene, die durch die Optik oder die Rasterung bei der Digitalisierung entstehen.

Die bei der Kantenextraktion eines Bildes entstehenden Kanten lassen sich in zwei Gruppen aufteilen. Kanten, die sowohl im Originalbild als auch im kantenextrahierten Bild vorkommen, gelten als korrekte Kanten. Kanten im Ergebnisbild, die keiner der Kanten im Originalbild zugeordnet werden können, werden als falsche Kanten bezeichnet. Eine dritte Gruppe sind die fehlenden Kanten. Diese Kanten existieren zwar im Eingabebild, jedoch fehlt das Gegenstück im kantenextrahierten Bild.

Die Berechnung der Kanten erfolgt im allgemeinen durch die erste Ableitung des Eingabebildes. Dabei werden die Daten des Bildes als die Diskretisierung einer kontinuierlichen Funktion interpretiert. Starke Änderungen der Intensität verursachen lokale Maxima in der Ableitungsfunktion. Einige Algorithmen zur Kantenerkennung analysieren die zweite Ableitung des Eingabebildes. Nullstellen in dieser zweiten Ableitung sind, wie leicht nachzuvollziehen ist, ebenfalls ein Indikator für Kanten im Eingabebild.

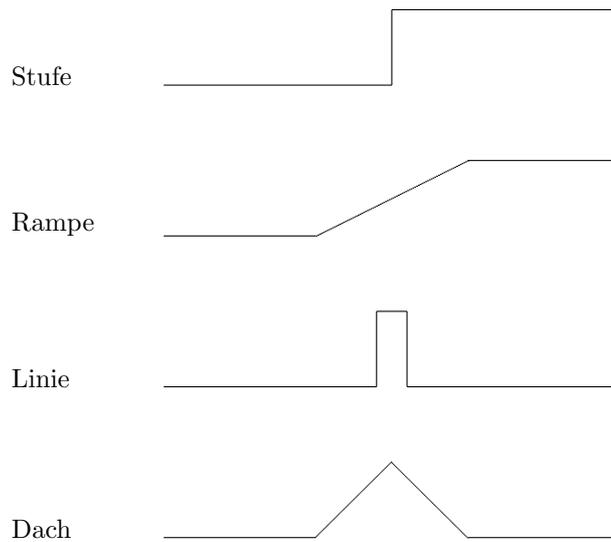


Abbildung 2.4: Kantenausprägungen

### 2.6.1 Numerische Approximation

Die numerische Approximation ist das einfachste Verfahren zur Kantenextraktion. Die Auswertung des Eingabebildes erfolgt in zwei Bearbeitungsschritten, einen für die horizontalen und einen für die vertikalen Kanten. Jeder Schritt entspricht einer Differentiation in der zur Kante senkrechten Richtung. Allein zwei nebeneinander liegende Bildpunkte werden miteinander verglichen, um daraus Rückschlüsse auf die Existenz einer Kante zu ziehen. Gleichung (2.12) zeigt die Operationsmatrizen für diesen einfachen Fall.

$$\mathcal{M}_x = \begin{pmatrix} -1 & 1 \end{pmatrix} \quad \mathcal{M}_y = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad (2.12)$$

Zu beachten ist dabei, daß der berechnete Kantenwert nicht für den Bildpunkt  $(x, y)$  gilt, sondern bei  $\mathcal{M}_x$  für den interpolierten Punkt  $(x, y + \frac{1}{2})$  und bei  $\mathcal{M}_y$  entsprechend für  $(x + \frac{1}{2}, y)$ . Die gültige Position des Ergebnisses unterscheidet sich also für einen konstanten Punkt  $(x, y)$  bei der horizontalen und vertikalen Bearbeitung. Abhilfe schafft die Erweiterung der Bearbeitungsmatrix der Größe  $1 \times 2$  bzw.  $2 \times 1$  zu zwei  $2 \times 2$ -Matrizen, dargestellt in Gleichung (2.13).

$$\mathcal{M}_x = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \quad \mathcal{M}_y = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \quad (2.13)$$

Durch diese Erweiterung ist der resultierende Bildpunkt der Berechnung von  $(x, y)$  sowohl im horizontalen als auch im vertikalen Fall stets  $(x + \frac{1}{2}, y + \frac{1}{2})$ . Eine solche Erkenntnis drängt den Gedanken auf, diese Bearbeitungsmatrizen auf  $3 \times 3$ -Matrizen zu erweitern. Der Vorteil: Eingabebildpunkt und Ausgabebildpunkt sind identisch. Diesem Gedanken wird im folgenden Abschnitt Rechnung getragen.

### 2.6.2 Sobel-Operator

Der Sobel-Operator stellt einen der meist benutzten Kantenoperatoren dar. Zusätzlich zur Kantenextraktion führt der Sobel-Operator eine gewichtete Glättung in Kantenrichtung zum Zweck der Rauschunterdrückung durch. Gleichung (2.14) zeigt

die beiden Bearbeitungsmatrizen.

$$\mathcal{S}_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad \mathcal{S}_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (2.14)$$

Der Grauwert des aktuell bearbeiteten Bildpunktes ist der Gradientenbetrag der horizontalen und vertikalen Ableitung an dieser Stelle. Wahlweise kann auch das Maximum der beiden Ableitungen als neuer Grauwert angenommen werden.

$$f(x, y) = \sqrt{s_x^2 + s_y^2} \quad (2.15)$$

$$f(x, y) = \max(s_x, s_y) \quad (2.16)$$

Die Zugehörigkeit des Sobel-Operators zu den Verfahren, die mit Hilfe der ersten Ableitung Kantenpunkte aus dem Eingabebild extrahieren, bedeutet, daß Kanten nicht nur auf ihre lokalen Maxima abgebildet werden. Vielmehr steigt die Intensität der Kantenpunkte zur Kantenmitte stetig an. Eine anschließende Schwellwertbildung führt zu einer verbreiterten Darstellung der Kantenlinie.

### 2.6.3 Laplace-Operator

Der Laplace-Operator basiert im Gegensatz zu den bisher vorgestellten Verfahren auf der Verarbeitung der zweiten Ableitung eines Eingabebildes. Ein weiterer Unterschied ist, daß der Laplace-Operator sowohl horizontale als auch vertikale Kanten in einem Durchgang bearbeitet. Der Laplace-Operator erreicht, verglichen mit dem Sobel-Operator, die doppelte Geschwindigkeit. Die mathematische Grundlage für den Laplace-Operator ist in Gleichung (2.17) dargestellt.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.17)$$

Dies führt zu einer Operatormatrix, wie in Gleichung (2.18) links abgebildet. Die Matrix in Gleichung (2.18) rechts zeigt ebenfalls einen Laplace-Operator, jedoch mit einer größeren Gewichtung der Bildpunkte im Zentrum.

$$\nabla^2 \approx \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{bzw.} \quad \nabla^2 \approx \begin{pmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{pmatrix} \quad (2.18)$$

Die Verwendung der zweiten Ableitung für die Kantenextraktion hat zwar den Vorteil, jedes Kantensegment mit einer Breite von nur einem Bildpunkt abzubilden, zugleich wird der Kantenfindungsprozeß aber durch Rauschen und andere Bildfehler empfindlich gestört. Jedes lokale Maximum in der ersten Ableitung erzeugt einen Nulldurchgang in der zweiten Ableitung. Selbst schwächste Kanten, die durch Schwellwertbildung bei der einfachen Ableitung eliminiert werden können, werden in das Ausgabebild übernommen und erzeugen so eine Fülle von falschen Kanten. Deshalb werden Kantenoperatoren, basierend auf der Verarbeitung der zweiten Ableitung, nur selten verwendet.

## 2.7 Binärbilderzeugung

Viele Bildoperatoren verarbeiten Grauwertbilder. Dies ist jedoch nicht die Regel. Einige Verfahren setzen ein binarisiertes Bild, also ein Bild mit lediglich zwei Helligkeitsstufen, vorzugsweise 0 und 1, voraus. Um ein Grauwertbild mit einem solchen Verfahren bearbeiten zu können, muß dieses also umgewandelt werden. Einen solchen Prozeß nennt man Thresholding oder Schwellwertbildung.

### 2.7.1 Fester Schwellwert

Das einfachste Verfahren zur Erzeugung eines Zweipegelbildes ist das Setzen einer fixen Grenze. Alle Grauwerte, die unterhalb dieser Grenze liegen, werden im Ausgabebild mit 0, üblicherweise dem Hintergrund, belegt. Grauwerte oberhalb werden analog mit 1 belegt.

$$f'(x, y) = \begin{cases} 1, & \text{falls } f(x, y) > c \\ 0, & \text{sonst} \end{cases} \quad (2.19)$$

Dabei ist  $c$  der Schwellwert. Dieser Schwellwert kann a priori festgelegt sein und ist somit für jedes zu bearbeitende Bild gleich. Ein solcher Schwellwert wird statischer Schwellwert genannt.

Eine zweite Möglichkeit der Schwellwertbestimmung ist die Analyse des Eingabebildes durch ein Histogramm, d. h. eine Analyse der Grauwerte und deren Häufigkeit. Bei einem hellen Objekt vor einem dunklen Hintergrund ergibt sich ein sogenanntes bimodales Histogramm, ein Histogramm mit zwei Maxima. Der Schwellwert wird dann mit dem lokalen Minimum zwischen den beiden Maxima belegt.

Der Ausdruck Semischwellwertbildung wird verwendet, wenn die Grauwerte unterhalb der Schwelle zwar auf 0 gesetzt, die Grauwerte über dem Schwellwert jedoch unverändert in das Ausgabebild übernommen werden. Die Semischwellwertbildung ist sinnvoll, um Teile des Bildes von vornherein von einer weiteren Bearbeitung auszuschließen.

### 2.7.2 Adaptiver Schwellwert

Sind Bilder in ihrer Ausleuchtung ungleichmäßig, führt eine Binarisierung mit festem Schwellwert zu keinem befriedigenden Ergebnis. Vielmehr muß auf die unterschiedlichen Lichtverhältnisse eingegangen werden. Dazu wird das Eingabebild in Regionen, vorzugsweise Rechtecke oder Quadrate, aufgeteilt. Für jede dieser Regionen wird anschließend eine individuelle Schwellwertbestimmung, beispielsweise durch ein Histogramm, vorgenommen und die Region für sich binarisiert.

### 2.7.3 Variabler Schwellwert

Eine weitere Möglichkeit, in der Ausleuchtung variierende Bilder zu verarbeiten, ist der variable Schwellwert. Hierzu wird, vorzugsweise anhand des Hintergrundes – und deshalb auch manchmal Hintergrundnormalisierung genannt – die Ausleuchtung analysiert und daraus eine mathematische Funktion ermittelt, die die Beleuchtungsverhältnisse im Bild möglichst genau wiedergibt. Solche Funktionen können simple Ebenen, aber auch komplexe Flächen sein. Anhand dieser mathematischen Funktion wird der Schwellwert bei der Bearbeitung des Eingabebildes abhängig von der aktuellen Position für jeden Bildpunkt dynamisch festgelegt.

## 2.8 Distanztransformation

Die Distanztransformation eines binarisierten Bildes mit Hintergrund 0 und Vordergrund 1 gibt für jeden Punkt Auskunft über dessen Entfernung zum nächstgelegenen Vordergrundbildpunkt. Vordergrundbildpunkte selbst haben die Entfernung 0. Entfernung ist dabei kein eindeutiger Begriff und bedarf daher einer genaueren Definition.

### 2.8.1 Definition von Distanz

Die Distanz zwischen zwei Punkten einer Ebene kann abhängig vom Anwendungsgebiet und den gestellten Anforderungen unterschiedlich definiert werden. Da die



**Schachbrett-Distanz**

Die Schachbrett-Distanz ist das Maximum der Entfernungen in  $x$ - und  $y$ -Richtung zwischen Punkt  $a$  und  $b$ .

$$d((x_a, y_a), (x_b, y_b)) = \max(|x_a - x_b|, |y_a - y_b|) \tag{2.28}$$

$$\begin{array}{cccccc}
 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
 3 & 2 & 2 & 2 & 2 & 2 & 3 \\
 3 & 2 & 1 & 1 & 1 & 2 & 3 \\
 3 & 2 & 1 & 0 & 1 & 2 & 3 \\
 3 & 2 & 1 & 1 & 1 & 2 & 3 \\
 3 & 2 & 2 & 2 & 2 & 2 & 3 \\
 3 & 3 & 3 & 3 & 3 & 3 & 3
 \end{array} \tag{2.29}$$

**2.8.2 Berechnung der Distanztransformation**

Zur Berechnung der Distanztransformation muß das binäre Bild  $[0, 1]$  so umgeformt werden, daß den Hintergrundbildpunkten 0 der Wert  $\infty$  und den Vordergrundbildpunkten 1 der Wert 0 zugewiesen wird. Dadurch sind alle Hintergrundbildpunkte unendlich weit vom nächstgelegenen Vordergrundbildpunkt entfernt, wohingegen die Vordergrundbildpunkte, entsprechend der Definition, eine Entfernung gleich 0 haben. Die Bildpunkte ungleich 0 müssen nun korrigiert werden. Dies geschieht, abhängig von der gewählten Definition für Distanz, mit jeweils vier Matrizen. Beispielfhaft sind in (2.30) die Bearbeitungsmatrizen für die Euklidische Distanz dargestellt.

$$\begin{array}{l}
 \mathcal{M}_{\searrow} = \begin{pmatrix} \infty & \sqrt{5} & \infty \\ \sqrt{5} & \sqrt{2} & 1 \\ \infty & 1 & 0 \end{pmatrix} \quad \mathcal{M}_{\nearrow} = \begin{pmatrix} \infty & 1 & 0 \\ \sqrt{5} & \sqrt{2} & 1 \\ \infty & \sqrt{5} & \infty \end{pmatrix} \\
 \mathcal{M}_{\swarrow} = \begin{pmatrix} 0 & 1 & \infty \\ 1 & \sqrt{2} & \sqrt{5} \\ \infty & \sqrt{5} & \infty \end{pmatrix} \quad \mathcal{M}_{\nwarrow} = \begin{pmatrix} \infty & \sqrt{5} & \infty \\ 1 & \sqrt{2} & \sqrt{5} \\ 0 & 1 & \infty \end{pmatrix}
 \end{array} \tag{2.30}$$

Nacheinander werden diese Matrizen in ihrer durch  $\searrow$ ,  $\nearrow$ ,  $\swarrow$  und  $\nwarrow$  vorgegebenen Richtung über das Bild bewegt. Anders als bei den bisher vorgestellten Bearbeitungsmatrizen ist der zu manipulierende Bildpunkt nicht das Zentrum der Matrix, sondern der mit 0 belegte Randpunkt. Diesem Punkt wird das Minimum der Summen aus jeweils einem Punktwert und dem entsprechenden Matrixelement zugewiesen. Gleichung (2.31) zeigt die Berechnung von  $f(x, y)$  mit  $\mathcal{M}_{\searrow}$ .

$$f(x, y) = \min \begin{pmatrix} f(x-2, y-1) + \sqrt{5}, & f(x-1, y-2) + \sqrt{5}, & \\ f(x-1, y-1) + \sqrt{2}, & f(x-1, y-1) + \sqrt{2}, & f(x, y-1) + 1, \\ f(x-1, y) + 1, & & f(x, y) + 0 \end{pmatrix} \tag{2.31}$$

**2.8.3 Voronoi-Diagramm**

Werden die Bildebene als Grundebene und die Werte der Bildpunkte als Höhe im Raum betrachtet, so wird die entstehende Fläche Voronoi-Fläche genannt. Für jeden Vordergrundbildpunkt ergibt sich, abhängig von der verwendeten Distanzdefinition,

---

<sup>2</sup> $\searrow$  entspricht der Leserichtung, also der Abarbeitung von rechts nach links für jede Zeile und zeilenweise von oben nach unten. Die Abarbeitung für  $\nearrow$ ,  $\swarrow$  und  $\nwarrow$  ist analog.

der Teil eines auf der Spitze stehenden Kegels bzw. einer Pyramide. Werden die Schnittkanten der einzelnen Körper auf die Grundebene projiziert, so entsteht ein sogenanntes Voronoi-Diagramm. Die sich aus der Projektion ergebenden Linien im Voronoi-Diagramm sind äquidistante Bereiche zweier benachbarter Punkte zu denen kein dritter Punkt näher liegt.

## Kapitel 3

# Objekterkennung

Dieses Kapitel dient nur zum Teil der Einführung von Grundlagen, die notwendig sind, um den in Kapitel 5 vorgestellten Objekterkennungsprozeß zu verstehen. Teil 1 dieses Kapitels leitet in das Thema ein und nimmt eine Bewertung der gestellten Aufgabe vor. Teil 2 stellt Werkzeuge und Arbeitsweise etablierter Objekterkennungsstrategien dar, um den Unterschied zu dem in Kapitel 4 beschriebenen Verfahren zu verdeutlichen. Dabei soll weniger detailliertes Wissen als vielmehr ein grundsätzliches Verständnis über die Vorzüge und Probleme bestehender Ansätze vermittelt werden.

### 3.1 Einführung

Im Gegensatz zu den in Kapitel 2 vorgestellten Bildverarbeitungsalgorithmen befaßt sich die Objekterkennung mit der Interpretation von Bildern. Folglich ist die produzierte Ausgabe kein neues Bild, sondern vielmehr eine Symbolik, die Aussagen über den Inhalt einer abgebildeten Szene macht. Die Existenz eines gesuchten Objekts oder dessen Position sind typische Ergebnisse der Objekterkennung.

Die Algorithmen zur Objekterkennung unterscheiden sich durch die Art von Wissen, das benötigt wird, um eine Szene zu analysieren, und die Art der Bildinformation, die verarbeitet wird.

Wissen ist notwendig, um die Semantik einer Szene zu verstehen. Es gibt jedoch zwei grundsätzlich verschiedene Ansätze für die Darstellung dieses Wissens. Zum einen ist da das Modellwissen. Das System hat genaue Kenntnis über das zu suchende Objekt. Beschrieben wird das Objekt durch besondere Merkmale, Kanten, Flächen, Löcher oder auch primitive Körper. Entscheidend ist dabei auch die Anordnung dieser Merkmale zueinander. Verwaltet wird solche Information in sogenannten Modelldatenbanken. Eine ganz andere Möglichkeit ist, den Kontext zu kennen, in dem das zu suchende Objekt vorkommt. Selbst die Kenntnis über die Funktionalität eines Objekts kann verwendet werden, um eine Szene auszuwerten.

Die zur Verfügung stehende Bildinformation kann unterschiedlich dargestellt sein. Häufig wird zwischen zwei-dimensional, zweieinhalb-dimensional und drei-dimensional unterschieden. Zwei-dimensional sind herkömmliche, photometrische Abbildungen, die die Intensität einer Szene Punkt für Punkt wiedergeben. Zweieinhalb-dimensional wird jene Bilddarstellung bezeichnet, die durch Entfernungsmessung entstanden ist. Jedem Bildpunkt ist die Entfernung zwischen dem Beobachterpunkt und dem ersten Hindernis, auf das der Sehstrahl trifft, zugeordnet. Drei-dimensional schließlich ist die komplexeste, aber auch informationsreichste Darstellung einer Szene. Wie beim menschlichen Sehen wird eine Szene aus zwei unterschiedlichen Blickwinkeln aufgenommen und zueinander gehörende Teile aus beiden Bildern ver-

knüpft.

Aufgrund der technischen Voraussetzungen beschränkt sich diese Arbeit auf die Verwendung von zwei-dimensionalen Intensitätsbildern des sichtbaren Lichts. Als Wissensbasis wird, angepaßt an die Problemstellung, eine Modelldatenbank verwendet.

## 3.2 Einstufung der gestellten Aufgabe

Für eine Einstufung der gestellten Aufgabe innerhalb des Gebietes der Objekterkennung werden Bildinformation und Modellinformation nach Suetens, Fua und Hansons [SFH92] klassifiziert.

### 3.2.1 Komplexität der Bildinformation

Bilddaten werden in zwei Kategorien eingeteilt, einfache Daten und komplexe Daten. Die Klassifizierung orientiert sich dabei hauptsächlich an den Störeinflüssen, die das Bild enthält. Dazu zählen neben dem Rauschanteil des Bildes ebenso die Vorhersagbarkeit des Inhaltes. Bilder mit sterilem, stets gleichbleibendem Hintergrund sind wesentlich einfacher zu bearbeiten als Abbilder einer natürlichen Umgebung, die nicht ohne weiteres in Vorder- und Hintergrund aufgeteilt werden können. Eine solche Umgebung ist in Abbildung 5.2 auf Seite 53 zu sehen.

### 3.2.2 Komplexität der Modellinformation

Die Zuordnung von Modellen in die Klassen einfache Modelle und komplexe Modelle erfolgt durch die Bewertung der Modellgeometrie. Wenn die Darstellung des Modells eine hierarchische Anordnung mehrerer Teilmodelle erforderlich macht, so spricht man von einem komplexen Modell. Ein hierarchischer Aufbau kann notwendig sein, wenn nur durch die Kombination von elementaren Bausteinen das Modell ausreichend genau zu beschreiben ist, oder wenn das Modell in sich beweglich gestaltet werden soll. Modelle, die aus nur einem Grundkörper bestehen, werden demzufolge als einfache Modelle bezeichnet.

### 3.2.3 Komplexitätsklassifizierung

Die Komplexität des gesamten Objekterkennungsprozesses läßt sich mit dieser Klassifizierung in vier Stufen einteilen:

1. **Einfache Daten und ein einfaches Modell** lassen sich durch Segmentierung mittels Klassifizierung (siehe 2.5) zuverlässig bearbeiten. Für einen homogenen Hintergrund und ein konstantes Objekt ist es möglich, hinreichend genaue Merkmalräume zu definieren. Zusätzliche Kriterien, wie Form und Größe, erlauben eine weitere Einschränkung.
2. **Einfache Daten und ein komplexes Modell** sind verhältnismäßig leicht zu bearbeiten, wenn wichtige symbolische Strukturen zuverlässig aus dem Eingabebild gewonnen werden können. Auf die Beschreibung des komplexen Modells passende Bildteile werden verwendet, um den zu bearbeitenden Suchraum einzuschränken und schließlich eine Objekthypothese aufzustellen.
3. **Komplexe Daten und ein einfaches Modell** werden durch die Aufbereitung des Eingabebildes und einer anschließenden Suche nach übereinstimmenden Merkmalen zwischen Modell und Bild bearbeitet. Starre, parametrisierte Modelle sind ebenso üblich wie flexible Modelle. Eine gute Bildvorverarbeitung ist die Voraussetzung für eine robuste Objekterkennung.

4. **Komplexe Daten und ein komplexes Modell** erfordern eine Kombination von Objekterkennungsstrategien. Ohne Zweifel gehört diese Kategorie zu den schwierigsten Aufgaben im Bereich der Objekterkennung. Zwei typische Beispiele für die Bearbeitung solcher Probleme sind 3DPO [BH86] und die MDL-Methode (minimal description length method) [FH93].

### 3.2.4 Wertung

Das gestellte Problem, die Suche des Mülleimers in einer Büroumgebung, gehört zu Klasse 3. Die folgende Diskussion verschiedener Aspekte der Objekterkennung orientiert sich an dieser Einstufung.

## 3.3 Grundlagen

Die Objektorientierung läßt sich in zwei Bereiche aufteilen, die *Low-level*-Objekterkennung und die *High-level*-Objekterkennung.

Die *Low-level*-Objekterkennung beschränkt sich auf die Verarbeitung von Bilddaten und die Suche von Objekten mit verhältnismäßig einfachen Modellinformationen. Beispiele dafür sind eine Farbbeschreibung des Objekts, dessen Hauptachsen, die Angabe der Größe von interessanten Bildbereichen und die Fouriertransformation der Bilddaten. Die Einsatzmöglichkeiten für dieses Verfahrens sind wegen seiner Einfachheit sehr beschränkt.

Die *High-level*-Objekterkennung macht sich detaillierte Modellinformationen bei der Objektsuche zunutze. Dabei werden üblicherweise anhand einer Modelldatenbank Objekthypothesen aufgestellt. Wird eine Objekthypothese durch die anschließende Verifikation bestätigt, so ist ein Objekt gefunden. Dieses Verfahren ist – verglichen mit der *Low-level*-Objekterkennung – wesentlich leistungsfähiger, aber in der Regel auch besonders rechenintensiv.

Zur Erkennung von Objekten durch genaue Kontextinformation haben sich zwei Ansätze etabliert, die kombinatorischen und die auf Korrelation basierenden. Die Arbeitsweise der kombinatorischen Ansätze baut auf der Verknüpfung von Modell- und Bildelementen auf. Die auf Korrelation basierenden Ansätze interpretieren Modell- und Bildinformation als Funktionen. Bei der Objektsuche ist die Korrelation zwischen diesen Funktionen das Entscheidungskriterium.

Ein Großteil der Objekterkennungsstrategien basiert auf der Kombination von Modell- und Bildelementen. Daher wird im folgenden ein kurzer Überblick über die grundlegenden Techniken gegeben, die bei der Objekterkennung Anwendung finden oder als Basis für neue Verfahren dienen.

### 3.3.1 Modelldatenbank

Entscheidend bei der Objekterkennung ist zu wissen, wonach gesucht werden soll. Diese Information liefert die Modelldatenbank. In einer Modelldatenbank sind alle Objekte abgelegt, die von der Objekterkennung erfaßt werden sollen. Dabei kann jedes Objekt einmal oder für verschiedene Ansichten wiederholt in einer Datenbank enthalten sein. Abhängig von der Objekterkennungsstrategie sind unterschiedliche Daten über das Modell gespeichert. Grundlegende Unterschiede bestehen zwischen der Darstellung des Objekts durch Grundbausteine und der Beschreibung des Objekts durch seine Eigenschaften und Merkmale.

#### Primitive Körper

Eine Darstellungsmethode für Modelle ist das Kombinieren von primitiven Körpern. Als primitive Körper gelten Quader, Kegel, Zylinder und Kugel. Diese werden durch

Vereinigung, Durchschnitt und Differenz zu einem komplexen Objekt verknüpft. Die mit dieser Methode dargestellten Objekte sind in der Regel relativ einfach. Stühle und Tische sind Objekte, die leicht aus primitiven Körpern konstruiert werden können. Für Objekte wie etwa Schuhe ist eine Modellierung aus Primitiven oft zu aufwendig.

### Voxel-Repräsentation

Analog zu Bildern, die durch einzelne Pixel<sup>1</sup> dargestellt werden, können dreidimensionale Körper aus sogenannten Voxeln<sup>2</sup> aufgebaut werden. Abhängig von der Größe dieser Voxel ist die Auflösung, mit der Details des Objekts dargestellt werden können. Die einfachste Form eines Voxels ist der Würfel, aber auch komplexere Formen wie Oktaeder und Tetraeder sind vorstellbar. Die direkte Verarbeitung von durch Voxel beschriebenen Modellen ist schwierig, daher werden vor der Verarbeitung anhand der Voxel-Repräsentation Merkmale des Modells bestimmt, die zur Objekterkennung verwendet werden können.

### Oberflächenbeschreibung

Ein Objekt kann durch die Definition einer oder mehrerer Flächen beschrieben werden, die eine Näherung der Objektoberfläche darstellen. Verfahren aus der graphischen Datenverarbeitung finden hier ihre Anwendung. Dabei reicht das Spektrum von der Annäherung der Fläche durch die Anordnung von Dreiecken bis hin zur Verwendung von *nonuniform rational B-splines*, abgekürzt NURBS.

### Globale Merkmale

Als globale Merkmale werden Eigenschaften bestimmter Bereiche bezeichnet. Dazu gehören unter anderem die Größe, der Umfang, die Hauptachsen, festgelegt durch die Eigenvektoren, und die Fouriertransformierte. Die Auswertung globaler Merkmale ist kritisch, wenn eine teilweise Verdeckung des gesuchten Objekts zulässig ist. Ist dies der Fall, so wird die Auswertung der Bilddaten Werte liefern, die sich mit den Anforderungen aus der Modelldatenbank nicht decken lassen. Die Folge ist: das Objekt wird nicht erkannt.

### Lokale Merkmale

Lokale Merkmale sind Besonderheiten des Objekts, die aus dem Kontext eines kleinen Ausschnittes heraus ersichtlich sind. Dabei werden hauptsächlich Eigenschaften betrachtet, die sich an den Grenzen des Objekts erkennen lassen. Typische lokale Merkmale sind Kanten, deren konkave oder konvexe Krümmung und, besonders wichtig, Ecken als Extreme der Krümmung.

### Relative Position

Die Lage von lokalen bzw. globalen Merkmalen zueinander wird durch deren relative Position festgelegt. Sie stellt die Verknüpfung zwischen den ansonsten nur in ihrer Art definierten Merkmalen dar. Die relative Position bestimmt auch die Lage verschiedener Teile des Objekts zueinander, wenn das Objekt als eine Kombination aus mehreren Einzelteilen dargestellt wird. Gerade die Definition der relativen Position macht eine Objektbeschreibung aus, denn die gleichen Merkmale in verschiedenen Konstellationen können ganz unterschiedliche Objekte beschreiben.

<sup>1</sup>Pixel ist ein Kunstwort, gebildet aus den beiden englischen Wörtern *picture* und *element*.

<sup>2</sup>Voxel ist das Analogon zu Pixel und ergibt sich aus der Kombination von *volume* und *element*.

### Mehrfachansichten

Soll ein drei-dimensionales Objekt in mehreren Orientierungen erkannt werden, so müssen verschiedene Ansichten des Objekts in der Modelldatenbank gespeichert werden. Jede dieser Ansichten wird folglich als das gesuchte Objekt erkannt. Da die Speicherung aller Ansichten eines Objekts nicht realisierbar ist, werden üblicherweise typische Ansichten in einer gewissen Rasterung der Freiheitsgrade in die Datenbank aufgenommen. Zum einen sind kleine Abweichungen in der Orientierung bei der Objekterkennung unproblematisch, solange sich die charakteristische Ansicht nicht wesentlich ändert, zum anderen können durch eine vernünftige Einschränkung der möglichen Orientierungen sowohl die Bearbeitungsdauer verkürzt als auch die Suche auf sinnvoll erscheinende Instanzen beschränkt werden. Ein auf der Seite oder gar auf dem Dach liegendes Auto kann für viele Anwendungen als nicht relevant eingestuft und daher ausgeschlossen werden. Eine weitere Einschränkung stellen rotationssymmetrische Körper dar. Da eine Drehung des Objekts um die Rotationsachse auf das Erscheinungsbild keinen Einfluß hat, kann dieser Freiheitsgrad unberücksichtigt bleiben.

Eine besondere Form, verschiedene Ansichten eines Objekts zu speichern, stellt der Aspektgraph dar. In einem Aspektgraphen werden alle sogenannten stabilen Ansichten eines Objekts hinterlegt. Aus der Anordnung im Aspektgraphen können zu jeder Ansicht alle Ansichten gefunden werden, die sich aus einer ähnlichen Orientierung des Objekts ergeben.

### 3.3.2 Merkmalsextraktion

Aus dem auszuwertenden Bild müssen die Merkmale extrahiert werden, die zur Objektbeschreibung verwendet wurden. Daraus ergibt sich, daß eine Bearbeitung der Bilddaten abhängig von der geplanten Objekterkennungsstrategie erfolgen muß. Da Kanten bei der Objekterkennung und daher auch bei der Objektbeschreibung eine große Rolle spielen, ist das Ziel der Merkmalsextraktion oft die Kantenextrahierte des Eingabebildes oder deren Distanztransformation. Zur Herleitung solcher Informationen aus dem Originalbild ist eine Kombination verschiedener Grundverfahren aus der digitalen Bildverarbeitung notwendig. Voraussetzungen und Wirkungsweise solcher Verfahren wurden in Kapitel 2 bereits diskutiert.

### 3.3.3 Aufstellen einer Objekthypothese

Wurden die Merkmale aus dem Bild extrahiert, so muß nun versucht werden, diese mit den in der Modelldatenbank abgelegten Merkmalen in Verbindung zu bringen. Gelingt dies, so ist eine Objekthypothese gefunden. Im folgenden werden einige Verfahren vorgestellt, die als Grundlage bei der Objekterkennung dienen.

### Klassifizierung

Ein Modell für ein gesuchtes Objekt besteht aus einer Menge von Merkmalen. Jedem dieser Merkmale ist im Merkmalsraum eine feste Position zugeordnet. Wird ein Modell durch  $n$  Merkmale beschrieben, so definieren  $n$  Punkte im Merkmalsraum dieses Modell. In diesem Merkmalsraum können beliebig viele verschiedene Objekte definiert sein. Zur Beschreibung jedes Objekts können mehrere Modelle verwendet werden. Als Prototyp für dieses Objekt wird der Mittelpunkt bzw. der Mittelwert aus allen Modellmerkmalen für diese Objekt festgelegt.

Soll ein unbekanntes Objekt im Bild einem der vorgegebenen Modelle zugeordnet werden, so wird die Klassifizierung anhand der Merkmalspunktswolken vorgenommen. Die Entfernung zwischen den Merkmalen des unbekanntes Objekts und denen der definierten Modelle wird berechnet. Das unbekanntes Objekt wird der Klasse,

also dem bekannten Objekt, zugeordnet, zu dem es im Merkmalsraum am nächsten liegt.

Neuronale Netze gehören ebenfalls zur den Klassifizierern. Sie zeichnen sich durch die Fähigkeit aus, den Merkmalsraum selbst in Klassen aufzuteilen. Dies geschieht während der Trainingsphase anhand der Objekte, die später erkannt werden sollen. Die Besonderheit neuronaler Netze liegt darin, daß nichtlineare Grenzen verwendet werden, um den Merkmalsraum in Klassenräume aufzuteilen. Ist das Training abgeschlossen, arbeitet das neuronale Netz wie jeder andere Klassifizierer.

### Kombination

Anders als bei der Klassifizierung werden bei der Kombination die Klassen sequentiell bearbeitet. Jede Klasse wird für sich mit den Merkmalen aus dem Eingabebild verglichen. Dabei wird versucht, die Merkmale der Klasse mit denen des Bildes in Übereinstimmung zu bringen.

Bei der **Merkmalskombination** wird jedem Klassenmerkmal eine Priorität abhängig davon zugewiesen, wie entscheidend die Existenz dieses Merkmals im Bild für die Korrektheit der Objekthypothese ist. Zusammen mit der Güte für die Kombination eines Klassenmerkmals mit einem Bildmerkmal ergibt sich eine Maßzahl für jedes Merkmal der Klasse. Die Summe dieser Maßzahlen ist das Kriterium für die Auswahl der Objekthypothese. Nachdem alle Klassen bewertet wurden, wird die als Objekthypothese angenommen, die die höchste Wertung erhalten hat. Zusätzlich kann für diese Wertung ein Schwellwert eingeführt werden, der übertroffen werden muß, um eine Klasse als gültige Objekthypothese anzuerkennen.

Die **Symbolische Kombination** macht sich, verglichen mit der Merkmalskombination, zusätzlich die Information über die relative Position von Merkmalen zueinander zunutze. Durch die Erweiterung um die geographische Abhängigkeit lassen sich sowohl die Merkmale der Klassen als auch die des Bildes jeweils als Graph darstellen. Ist das Objekt im Bild vollständig sichtbar, so reduziert sich die Problematik auf den Vergleich zweier Graphen. Ist das Objekt jedoch teilweise verdeckt, so ergibt sich der Vergleich eines Graphen mit einem Subgraphen und damit ein Problem in  $\mathcal{NP}$ , für dessen effiziente Lösung wenig Hoffnung besteht.

Die **Merkmalsindizierung** optimiert die Suche nach einer Objekthypothese dahingehend, daß im voraus bereits die Klassen von der Überprüfung ausgeschlossen werden, die anhand der vorkommenden Bildmerkmale für eine Objekthypothese nicht in Frage kommen.

Der Index ist Teil der Modelldatenbank und gibt für ausgesuchte Merkmale an, in welchen Klassen diese vorkommen und in welchen nicht. Das Bild wird nach Merkmalen aus dem Index durchsucht. Enthält das Bild ein solches Merkmal, können all die Klassen ausgeschlossen werden, die bei der Kombination von Klasse und Bild für dieses Merkmal kein passendes Gegenstück zur Verfügung stellen.

Nach der Reduzierung auf die möglichen Objektklassen erfolgt die Bestimmung der Objekthypothese wie gewohnt.

### 3.3.4 Überprüfung der Objekthypothese

Bei der Aufstellung der Objekthypothese wurde besonders darauf geachtet, zu jedem Klassenmerkmal ein passendes Bildmerkmal zu finden. Bei der Verifizierung der Objekthypothese wird überprüft, ob auch im umgekehrten Fall Bildmerkmale und Klassenmerkmale zusammenpassen. Insbesondere bei der symbolischen Kombination kann die Lage der Merkmale zueinander verifiziert werden. Es ergibt sich somit

ein aussagekräftiges Kriterium für die Korrektheit der Hypothese. Liegen Bildmerkmale in Bereichen, in denen kein entsprechendes Modellmerkmal angeordnet ist, so ist dies ein Indiz für eine falsche Hypothese. Kanten oder Ecken in Bereichen, in denen der gesuchte Körper große glatte Flächen aufweist, lassen Zweifel an der Korrektheit der aufgestellten Objekthypothese aufkommen. Befinden sich hingegen Bildmerkmale und Modellmerkmale in genügend dichtem Abstand zueinander und sind diese in ihrer Charakteristik vergleichbar, so wird die Hypothese als bestätigt angesehen. Das Objekt gilt als gefunden.

## 3.4 Gängige Objekterkennungsstrategien

Dieser Abschnitt stellt Objekterkennungsstrategien vor, die auf einem kombinatorischen Ansatz beruhen. Kombinatorik bedeutet, daß einzelne Merkmale von Modell und Bild verknüpft werden und auf die Existenz, die Position und die Lage des gesuchten Objekts im Bild geschlossen wird. Auf die Klasse der Strategien, die anhand der Güte von Korrelation zwischen Modell und Bild eine Entscheidung über die Existenz des Objektes im Bild treffen, wurde hier verzichtet. Dafür wird in Kapitel 4 ein solches Verfahren detailliert vorgestellt.

Eine Liste weiterführender bzw. relevanter Literatur zu jedem der vorgestellten Verfahren ist im Anschluß an jeden Abschnitt aufgeführt.

### 3.4.1 Interpretationsbäume

Interpretationsbäume werden verwendet, um zwei-dimensionale Modelle in zwei-dimensionale Bilder einzupassen. Als Merkmale für die Definition des Modells dienen Liniensegmente, die zur Suche durch Translation und Rotation mit den aus dem Bild extrahierten Liniensegmenten in Deckung gebracht werden. In der Erweiterung auf drei-dimensionale Modelle und Bilder werden anstatt der Liniensegmente polygonale Flächen verwendet.

Im Interpretationsbaum stehen alle Knoten einer Tiefe für ein Bildmerkmal. Jeder Knoten im Baum besitzt für alle Modellmerkmale einen Nachfolger und einen zusätzlichen, der ausdrückt, daß zu diesem Bildmerkmal kein passendes Modellmerkmal gefunden wurde. Die Blätter des Baumes stellen somit eine komplette Verknüpfung von Modell- und Bildmerkmalen dar.

Bei der Suche nach dem Objekt im Bild wird versucht, jedes Bildmerkmal mit einem Modellmerkmal zu verknüpfen. Alle passenden Paarungen werden mit den gefundenen Paarungen für bereits bearbeitete Bildmerkmale verglichen. Dabei wird paarweise überprüft, ob eine Transformation ausreichend ist, um beide Modellmerkmale mit den jeweils verknüpften Bildmerkmalen in Deckung zu bringen. Paßt keines der Modellmerkmale zum Bildmerkmal, so wird dies ebenfalls im Baum vermerkt. Wurde bei der Paarung von Bild- und Modellmerkmalen ein sogenanntes Blatt im Suchbaum erreicht, so sind alle Bildmerkmale mit Modellmerkmalen verknüpft. Nun muß überprüft werden, ob die Kombination der Merkmale konsistent ist, d. h. ob die Anordnung der einzelnen Merkmale so erfolgt ist, daß sich daraus das gesuchte Objekt ergibt. Ist dies der Fall, so ist das Objekt im Bild gefunden. Ist dies nicht der Fall, so wird mittels *backtracking* die gefundene Kombination so lange modifiziert, bis entweder eine gültige Paarung aller Merkmale gefunden ist oder keine weitere Möglichkeit der Paarung besteht.

Große Probleme dieses Verfahrens sind zum einen die exponentielle Laufzeit im *worst case* und zum anderen die zuverlässige Extraktion der Bildmerkmale. Durch Störungen im Bild können Kanten unterbrochen werden. Die Folge ist eine Schar von Liniensegmenten, die nicht mehr eindeutig zu Modellmerkmalen zugeordnet werden

können. Auch Kreissegmente sind problematisch, da deren Darstellung durch Liniensegmente nicht konsistent ist und durch Fehler im Bild oft stark variiert.

*Literaturliste:* [GLP87] und [Gri90b]

### 3.4.2 Ausrichtung

Bei der Ausrichtungsmethode wird mittels einer Untermenge der Modellmerkmale und einer Untermenge der Bildmerkmale eine Hypothese über Ort und Lage des Objekts aufgestellt, die anschließend anhand der Gesamt mengen verifiziert wird.

Zur Aufstellung der Hypothese werden aus der Menge der Modellmerkmale und der Menge der Bildmerkmale jeweils gleichgroße Teilmengen gebildet. Diese ausgewählten Merkmale werden zu Paarungen kombiniert, anhand derer die mögliche Position des gesuchten Objekts und dessen Lage ermittelt wird. Eine gültige Hypothese entspricht einer Transformation der Modellmerkmale so, daß diese mit den ausgewählten Bildmerkmalen in ausreichend genaue Deckung gebracht werden.

Überprüft wird die aufgestellte Hypothese durch die Betrachtung aller Modellmerkmale im Vergleich zu den vorhandenen Bildmerkmalen. Aufgrund der durch die Hypothese festgelegten Position und Lage werden nach und nach alle Modellmerkmale im Bild platziert und überprüft, ob dort vergleichbare Bildmerkmale gefunden werden können. Ist dies genügend oft und ausreichend gut der Fall, so wird die Hypothese als bestätigt angesehen und das Objekt ist gefunden. Weichen die Bildmerkmale von den transformierten Modellmerkmalen in unzulässigem Maße ab oder sind an den geforderten Positionen keine Bildmerkmale vorhanden, so muß die Hypothese als falsch angesehen und somit zurückgewiesen werden.

Hauptproblem bei der Ausrichtungsmethode ist der Fehler bei der Bestimmung der Objektlage durch eine Teilmenge der bekannten Merkmale. Aufgrund von kleinen Fehlern bei der Extraktion der Bildmerkmale kann die Voraussage der Objektposition und dessen Lage von der tatsächlichen abweichen. Die Verifikation hätte die Zurückweisung dieser Hypothese zur Folge, wenn die Position der restlichen eingblendeten Modellmerkmale durch den anfänglichen Fehler so stark verfälscht wurde, daß eine Korrelation zwischen Modell- und Bildmerkmalen nicht mehr nachvollziehbar ist. Die Einführung von Fehlerräumen für die Positionierung weiterer Modellmerkmale und die Korrektur der geschätzten Position des Objekts können dem entgegenwirken.

*Literaturliste:* [HU90], [Ols94], [GHJ92], [Jac91], [AJ94], [AF86] und [Low87]

### 3.4.3 Hough-Transformation

Werden die Freiheitsgrade einer Modelltransformation parametrisiert, so ist das Ziel der Objekterkennung die Parameter so zu wählen, daß die Modellmerkmale und die Bildmerkmale in ausreichende und möglichst gute Deckung gebracht werden. Mit Hilfe der Hough-Transformation werden die Parameter für die Transformation so bestimmt, daß die daraus resultierende Lage für das Objekt als Anhaltspunkt für die Suche nach der exakten Lage und Position des Objekts dient.

Zur Ermittlung der Transformationsparameter werden möglichst viele Paarungen zwischen Modell- und Bildmerkmalsteilmengen gebildet. Daraus werden alle Parameter abgeleitet, die die Modellmerkmale durch eine einzige Transformation mit den Bildmerkmalen in Übereinstimmung bringt. Korrekte Transformationen

werden in einem Histogramm eingetragen. Die Transformation, die aus unterschiedlichen Paarungen als gültige Transformation erkannt wurde, ist Anhaltspunkt für die korrekte Lage des Objekts und dient somit als Ausgangspunkt für die Suche nach der exakten Transformation.

Im Gegensatz zur Ausrichtungsmethode, bei der es das Ziel ist, einige wenige Transformationen der Modellmerkmale zur ermitteln, die zu einer Deckung mit den vorhandenen Bildmerkmalen führen, können bei der Hough-Transformation auch ganze Teilräume im Transformationsraum das Ergebnis der Verknüpfung von Modell- und Bildmerkmalen sein. Durch die Auswertung des Histogramms werden aus der beträchtlichen Menge der gefundenen Transformationen einige wenige als interessant eingestuft und dann weiterverarbeitet.

*Literaturliste:* [SHD86], [TM87], [GH90b] und [GH91]

### 3.4.4 Geometrisches Hashing

Eine Ebene im Raum wird durch drei Punkte eindeutig festgelegt, sofern die resultierenden Basisvektoren nicht kollinear sind. Eine durch die beliebige Rotation dieser Ebene definierte affine Abbildung ist durch die drei transformierten Punkte ebenfalls eindeutig festgelegt. Diese Beobachtung findet bei der Erkennung von zweidimensionalen Objekten in zwei-dimensionalen Bildern Anwendung. Das gesuchte Objekt kann dabei jede Lage im Raum einnehmen. Einzige Ausnahme ist: der Normalenvektor der Ebene darf auf keinem Sichtstrahl der perspektivischen Ansicht senkrecht stehen.

Aus den nichttransformierten Modellpunktmerkmalen werden jeweils drei zusammen dazu verwendet, um eine Basis herzuleiten. Diese Basis definiert einen Ursprung und zwei Basisvektoren, also ein Koordinatensystem. In diesem haben alle übrigen Punktmerkmale des Modells eine eindeutige Position. Werden alle Modellpunktmerkmale einer affinen Transformation unterzogen und anhand der drei ursprünglich verwendeten Punkte die Basis neu bestimmt, so ändert sich die relative Position der übrigen Punkte nicht.

Für jede Kombination aus drei Punkten des Modells und der daraus resultierenden Basis werden vor der eigentlichen Objekterkennung die Koordinaten aller übrigen Modellpunkte ermittelt und zusammen mit der zugrundeliegenden Basis in einer Tabelle, der sogenannten Hash-Tabelle, abgelegt.

Zur Objekterkennung werden aus dem zu untersuchenden Bild die Punktmerkmale extrahiert und drei davon beliebig ausgewählt. Die daraus hergeleitete Basis wird verwendet, um die Position der restlichen Bildpunkte zu interpretieren. Die erhaltenen Koordinaten werden mit den Eintragungen in der Hash-Tabelle verglichen und bei allen Übereinstimmungen die zugrundeliegenden Basen extrahiert. Ist aus der Hash-Tabelle eine Basis übermäßig oft ausgelesen worden, so läßt sich daraus schließen, daß die drei ausgewählten Bildpunkte Teil des gesuchten Objekts sind. Das Objekt gilt als gefunden, die Position ist durch die auffällige Basis eindeutig definiert. Wurde jedoch keine der Basen verstärkt extrahiert, so muß angenommen werden, daß nicht alle der drei Punkte Elemente des gesuchten Objekts waren. Für einen neuen Versuch wird ein neues Punktetrio ausgewählt und die gesamte Auswertung wiederholt.

Sind für das Modell mehr als circa 30 Punktmerkmale notwendig, so ist eine effektive Suche nach dem Objekt in der Regel nicht mehr möglich. Außerdem ist, wie in allen bisher vorgestellten Verfahren, auch hier die Lagebestimmung anhand einzelner Bildmerkmale kritisch. Störungen im Bild können dazu führen, daß anstatt der korrekten Lage eine leicht fehlerbehaftete Lage berechnet wird, die zur Folge hat,

daß die Einblendung weiterer Modellmerkmale nicht korrekt erfolgt und das Objekt nicht, oder nicht richtig lokalisiert wird. Desweiteren ist es möglich, daß beliebige unabhängige Punkte im Bild zufällig dazu führen, daß eine Basis bevorzugt aus der Hash-Tabelle extrahiert wird. Folge ist eine vermeindlich korrekte Erkennung eines Objekts im Bild.

*Literaturliste:* [LW88], [GH90a] und [GHJ92]

### 3.4.5 Berechenbare Geometrie

Algorithmen dieser Klasse konstruieren eine Anordnung von Subräumen im Transformationsraum. Die Anordnung dieser wird durchsucht, um herauszufinden, wo im Raum sich gute Transformationen befinden.

Jede Paarung zwischen zwei einzelnen oder zwei Teilmengen von Modell- und Bildmerkmalen ergibt im Transformationsraum einen zusammenhängenden Bereich, den sogenannten Subraum, innerhalb dessen alle Transformationen zu einer genügend guten Übereinstimmung der berücksichtigten Merkmale führen. In der Regel sind diese Bereiche kugelförmig oder typische Vielecke. Ziel ist es, den Bereich im Transformationsraum zu finden, in dem sich möglichst viele solcher Subräume überschneiden.

Aus diesem Bereich wird anhand eines Modellmerkmals eine Transformation so gewählt, daß dieses ausgewählte Modellmerkmal optimal zum entsprechenden Bildmerkmal angeordnet ist. Diese Transformation wird verwendet, um alle weiteren Modellmerkmale im Bild zu positionieren und somit mit Bildmerkmalen zu paaren.

Dabei tritt auch hier das typische Problem von kombinatorischen Ansätzen, nämlich ein eventueller Positionsversatz bzw. eine nicht einwandfreie Lage des gefundenen Objekts auf.

*Literaturliste:* [Cas90], [HU90], [Cas93] und [Hop95]

### 3.4.6 Aufteilung des Transformationsraumes

Durch Aufteilung des Transformationsraumes werden Bereiche, die offensichtlich keine interessanten Transformationen enthalten, frühzeitig von der weiteren Betrachtung ausgeschlossen. Die Suche konzentriert sich dann auf die Teile des Transformationsraumes, die darauf schließen lassen, darin Transformationen zu finden, die zu einer genügend guten Paarung von Modell- und Bildmerkmalen führen. Entscheidend beim Aufteilen des Transformationsraumes ist, daß keine Transformation durch vorzeitiges Ausschließen von Suchräumen übersehen wird. Jede genügend gute Transformation muß bei der Durchsuchung des Transformationsraumes gefunden werden.

Vergleichbar zu dem in 3.4.5 vorgestellten Verfahren werden Subräume, also zusammenhängende Bereiche gültiger Transformationen, im Suchraum bestimmt. Der Transformationsraum wird abhängig von der Anzahl der Dimensionen in Rechtecke, Quader, o. ä. aufgeteilt. Für jeden Quadranten wird überprüft, wieviele Subräume diesen schneiden und wieviele verschiedene Modellmerkmale zur Erzeugung der einzelnen Subräume geführt haben. Sind die ermittelten Werte für diesen Quadranten ausreichend hoch, so wird dieser weiter unterteilt und jeder der entstandenen Teilräume weiter untersucht. Die Aufteilung erfolgt solange, bis ein genügend kleiner Raum spezifiziert wurde, in dem dann nach herkömmlicher Weise weitergesucht

wird, oder die Unterteilung zu einem Quadranten führt, der nur noch eine Transformation enthält. Ergibt die Überprüfung des Quadranten, daß keine gültige Transformation in diesem Teilraum enthalten sein kann, so wird der komplette Quadrant von der weiteren Bearbeitung ausgeschlossen.

Obwohl diese Methode eine *branch-and-bound*-Technik realisiert und somit nur interessante Teile des Suchbaumes weiterverfolgt werden, arbeitet dieses Verfahren nicht effizient, wenn das Modell aus zahlreichen Merkmalen konstruiert werden muß oder das zu durchsuchende Bild eine große Anzahl von Merkmalen enthält. Der Aufwand für die Überprüfung jedes einzelnen Quadranten in der Größenordnung von  $O(mn)$  mit  $m$  gleich der Anzahl der Modellmerkmale und  $n$  gleich der Anzahl der Bildmerkmale steigt für große  $m$  bzw.  $n$  relativ schnell an.

*Literaturliste:* [Bru91]



# Kapitel 4

## Hausdorff-Distanz

Dieses Kapitel befaßt sich mit der theoretischen Grundlage der Hausdorff-Distanz. Die Hausdorff-Distanz und ihre Abwandlungen werden definiert und Verfahren zur Berechnung vorgestellt. Weiterhin wird gezeigt, welche Schritte nötig sind, um mit Hilfe der Hausdorff-Distanz eine Objekterkennung zu realisieren.

### 4.1 Einführung

Eine Objekterkennung, die auf der Auswertung der Hausdorff-Distanz beruht, zählt zu der Klasse der auf Korrelation basierenden Verfahren. Zur Bewertung der Korrelation zwischen Modell und Bild wird eine mathematische Funktion definiert, die von Modell- und Bildpunktmenge abhängt. Im Unterschied zu den kombinatorischen Verfahren, bei denen jedes Merkmal von Modell und Bild für sich betrachtet wird und aus der Menge der möglichen Paarungen das gesuchte Objekt *zusammengepuzzelt* wird, bewerten auf Korrelation basierende Verfahren Modell und Bild als ganzes bzw. beim Bild den als relevant erkannten Teil.

### 4.2 Definition

Die Hausdorff-Distanz  $H$  in ihrer Grundform definiert den Abstand zwischen zwei Punktmenge  $A$  und  $B$ .

$$H(A, B) = \max(h(A, B), h(B, A)) \quad (4.1)$$

$$h(A, B) = \max_{a \in A} \left( \min_{b \in B} (d(a, b)) \right) \quad (4.2)$$

Für die Verwendung der Hausdorff-Distanz im Bereich der Objekterkennung stellt  $A$  die Menge der Bildpunkte,  $B$  die Menge der Modellpunkte und  $d(a, b) = \sqrt{\Delta x^2 + \Delta y^2}$  mit  $\Delta x = x_a - x_b$  und  $\Delta y = y_a - y_b$  die Distanz zwischen einem Bild- und einem Modellpunkt dar.

#### 4.2.1 Gerichtete Hausdorff-Distanz

$h(A, B)$  wird die gerichtete Hausdorff-Distanz genannt. Ihr Wert entspricht der größten vorkommenden Entfernung eines Punktes  $a \in A$  zu dessen nächstgelegenen Punkt  $b \in B$ . Analog entspricht  $h(B, A)$  der größten vorkommenden Entfernung eines Punktes  $b \in B$  zu dessen nächstgelegenen Punkt  $a \in A$ .  $h(A, B)$  und  $h(B, A)$  sind dabei nicht notwendigerweise identisch. Angenommen, jeder Punkt in  $A$  liegt nahe eines Punktes aus  $B$ , aber es existieren Punkte in  $B$ , die vom nächsten Punkt

aus  $A$  weit entfernt sind. Bei  $h(A, B)$  werden alle Punkte der Menge  $A$  und ihre Lage zu  $B$  ausgewertet, es ergibt sich dadurch ein relativ kleiner Wert; die ‚Ausreißer‘ aus Menge  $B$  bleiben hier unberücksichtigt. Im umgekehrten Fall wird für  $h(B, A)$  der Abstand jedes Punktes aus  $B$  zu seinem nächstgelegenen Nachbarpunkt aus  $A$  bestimmt. Für die abseits von  $A$  liegenden Punkte  $b \in B$  ergibt sich somit ein relativ großer Wert, der schließlich auch zum größeren Wert von  $h(B, A)$  gegenüber  $h(A, B)$  führt.

### Hypothesendistanz

Für die Anwendung der Hausdorff-Distanz in der Objekterkennung wird die gerichtete Distanz  $h(B, A)$  zur Suche nach einer Objekthypothese im Bild verwendet.  $h(B, A)$  wird daher im folgenden als Hypothesendistanz<sup>1</sup> bezeichnet.

### Verifikationsdistanz

Eine aufgestellte Objekthypothese wird durch die gerichtete Distanz  $h(A, B)$  verifiziert. Für eine „sieht aus wie“-Beziehung zwischen dem Modell und dem Teil des Bildes müssen nicht nur die Modellpunkte nahe an Bildpunkten, sondern auch umgekehrt, Bildpunkte nahe an Modellpunkten liegen.  $h(A, B)$  wird demzufolge im weiteren Verifikationsdistanz<sup>2</sup> genannt.

## 4.2.2 Ungerichtete Hausdorff-Distanz

Die Hausdorff-Distanz  $H(A, B)$  ergibt sich aus dem Maximum der gerichteten Distanzen  $h(A, B)$  und  $h(B, A)$ . Sie wird oft als ungerichtete Distanz bezeichnet.

Ist  $H(A, B) = d$ , so liegt jeder Punkt  $a \in A$  höchstens  $d$  Einheiten vom nächsten Punkt  $b \in B$  entfernt, und alle Punkte in  $B$  liegen in einem Umkreis von  $d$  Einheiten um einen Punkt in  $A$ . Zusätzlich gilt, daß mindestens ein Punkt aus  $A$  relativ zu  $B$  oder umgekehrt genau  $d$  Einheiten vom nächsten Punkt aus  $B$  bzw.  $A$  entfernt liegt.

## 4.3 Metrische Eigenschaften

Die ungerichtete Hausdorff-Distanz ist eine Metrik, da sie die notwendigen Eigenschaften

1.  $\bigwedge_{A,B} H(A, B) \geq 0$
2.  $H(A, B) = 0$  genau dann, wenn  $A = B$
3.  $H(A, B) = H(B, A)$
4.  $H(A, C) \leq H(A, B) + H(B, C)$

erfüllt.

Diese Eigenschaften sind im Bereich der Objekterkennung bedeutend, denn ein Objekt ist mit sich selbst identisch (2.), die Reihenfolge bei einem Vergleich zweier Objekte ist unerheblich (3.) und zwei unterschiedlich aussehende Objekte können nicht gleichzeitig Ähnlichkeit mit einem dritten haben (4.).

<sup>1</sup>In englischsprachigen Veröffentlichungen, wie [HKR93], [HR] und [Ruc], wird hierfür der Begriff *forward distance* verwendet.

<sup>2</sup>Analog zu *forward distance* wird in der englischsprachigen Literatur von *reverse distance* gesprochen.

## 4.4 Erweiterung der Hausdorff-Distanz

Die Hausdorff-Distanz in ihrer ursprünglichen Definition, wie sie im Abschnitt 4.2 beschrieben wurde, eignet sich nur bedingt für die Objekterkennung. Geringe Unterschiede zwischen Bild und Modell führen zu einer kleinen Hausdorff-Distanz. Gegenüber teilweiser Verdeckung oder Störung durch Rauschen reagiert die Hausdorff-Distanz empfindlich. Ein einzelner gestörter Bildpunkt abseits der Modellpunkte wird bei der Bestimmung der Hausdorff-Distanz zum entscheidenden Element. Die Position der ungestörten Punkte bleibt dabei unberücksichtigt, gerade diese wäre aber für die Objekterkennung interessant.

### 4.4.1 Partielle Hausdorff-Distanz

Die Einführung der partiellen Hausdorff-Distanz relativiert dieses Problem. Anstatt alle Punkte aus  $A$  mit allen Punkten aus  $B$  zu vergleichen, wird nur ein Teil der jeweiligen Punktmenge mit der anderen verglichen. Wird dieser Teil günstig ausgewählt, so bleiben Störungen unberücksichtigt. Die Auswahl von  $A' \subseteq A$  bzw.  $B' \subseteq B$  soll so erfolgen, daß die am besten passende Teilmenge zum Vergleich herangezogen wird.

Da die Punkte selbst, die zur Berechnung der partiellen Hausdorff-Distanz verwendet werden, nicht von Bedeutung sind, sondern nur das daraus resultierende Ergebnis, kann die Suche nach einer Teilmenge dahingehend vereinfacht werden, daß lediglich eine vorher festgelegte Anzahl von Punkten beim Vergleich unberücksichtigt bleibt. Dazu werden alle Entfernungen von Punkten einer Menge zu ihrem nächstgelegenen Nachbarn der anderen Menge der Größe nach sortiert. Als Distanz zwischen den beiden Punktmenge wird die  $l$ -kleinste Entfernung angenommen.

$$h_l(A, B) = l\text{-tes min}(d(a, b)) \quad (4.3)$$

mit  $1 \leq l \leq |A|$

Die partielle ungerichtete Hausdorff-Distanz berechnet sich folglich als

$$H_{lk}(A, B) = \max(h_l(A, B), h_k(B, A)). \quad (4.4)$$

Diese Abschwächung der Hausdorff-Distanz erfüllt die metrischen Eigenschaften nicht. Jedoch bleiben die intuitiven Eigenschaften für den Bereich der Objekterkennung, die in 4.3 kurz dargestellt wurden, weitestgehend erhalten, wie im folgenden erläutert wird:

**Theorem 4.1** *Ist  $H_{lk}(A, B) = d$ , dann existiert  $A_l \subseteq A$  und  $B_k \subseteq B$ , so daß  $H(A_l, B_k) \leq d$ . Die Mächtigkeit von  $A_l$  und  $B_k$  ist genau  $\min(k, l)$ .*

*Beweis:* Sei  $A'$  die Menge der Punkte aus  $A$ , für die gilt:

$$A' = \left\{ a \in A \mid \bigvee_{b \in B} d(a, b) \leq d \right\}$$

Analog sei  $B' \subseteq B$ . Weiterhin muß gelten  $|A'| \geq l$  und  $|B'| \geq k$ , da nach  $h_l(A, B) \leq d$  mindestens  $l$  Punkte aus  $A$  näher als oder genau  $d$  Einheiten an einem Punkt aus  $B$  liegen. Gleiches gilt für  $h_k(B, A) \leq d$  analog. Die Symmetrieeigenschaft von  $d(a, b)$  sagt aus, daß die Nachbarpunkte für alle  $a \in A'$  Elemente der Menge  $B'$  sind und umgekehrt.

Das Problem reduziert sich auf die Suche nach  $A_l \subseteq A'$  und  $B_k \subseteq B'$  mit  $|A_l| = |B_k| = \min(k, l)$ , so daß  $H(A_l, B_k) \leq d$  gilt.

**Induktionsanfang:** Ein beliebiger Punkt  $a \in A'$  sei das erste Element in  $A_l$ . Wähle das erste Element  $b \in B'$  für  $B_k$  so, daß  $d(a, b) \leq d$  gilt. Damit ergibt sich  $H(A_l, B_k) = d(a, b) \leq d$ .

**Induktionsschritt:** Die Mengen  $A_l$  und  $B_k$  besitzen jeweils  $1 < n < \min(k, l)$  Elemente. Für einen Induktionsschritt ist eine Fallunterscheidung notwendig:

1. Existiert ein  $a \in A' \cap A_l$  und ein  $b \in B' \cap B_k$  mit  $d(a, b) \leq d$ , so wird  $A_l = A_l \cup a$  und  $B_k = B_k \cup b$ .
2. Existiert kein  $a \in A' \cap A_l$  und  $b \in B' \cap B_k$  mit  $d(a, b) \leq d$ , so wird ein beliebiges  $a \in A' \cap A_l$  gewählt. Der Nachbarpunkt aus  $B'$  befindet sich bereits in  $B_k$ . Analog kann ein beliebiger Punkt  $b \in B' \cap B_k$  gewählt werden. Auch dessen Nachbarpunkt befindet sich bereits in  $A_l$ . Für  $A_l = A_l \cup a$  und  $B_k = B_k \cup b$  gilt weiterhin  $H(A_l, B_k) \leq d$ .

Die Induktionsschritte lassen sich bis zu einer Mächtigkeit von  $A_l$  und  $B_k$  gleich  $\min(k, l)$  fortführen, da  $A'$   $l$  Elemente und  $B'$   $k$  Elemente besitzt.

Diese Induktion baut  $A_l$  und  $B_k$  schrittweise auf. Dabei ist sichergestellt, daß sich nach der Erweiterung von  $A_l$  und  $B_k$  um die Punkte  $a$  und  $b$  deren Nachbarn ebenfalls in  $B_k$  und  $A_l$  befinden. Die Bedingung  $H(A_l, B_k) \leq d$  bleibt also erhalten.  $\square$

**Theorem 4.2** *Ist  $H_{lk}(A, B) = d$ , dann existiert  $A'_l \subseteq A$  und  $B'_k \subseteq B$ , so daß  $H(A'_l, B'_k) = d$  mit  $l \leq |A'_l| \leq \max(k, l)$  und  $k \leq |B'_k| \leq \max(k, l)$ .*

*Beweis:*  $A'$ ,  $A_l$ ,  $B'$  und  $B_k$  beziehen sich jeweils auf ihre Definition in Beweis zu Theorem 4.1.

Ohne Beschränkung der Allgemeinheit wird  $l \leq k$  angenommen.  $k - l$  beliebige Punkte aus  $B' \cap B_k$  vereint mit  $B_k$  ergeben  $B'_k$ .  $A'_l$  ist die Vereinigung von  $A_l$  und den Nachbarpunkten  $a \in A'$  für alle Punkte  $b$  aus  $B_k \cap B'_k$ , sofern diese nicht schon in  $A_l$  enthalten sind. Da jeder Punkt aus  $A'_l$  einen Nachbarn im Umkreis von  $d$  Einheiten in  $B'_k$  und umgekehrt besitzt, gilt  $H(A'_l, B'_k) \leq d$ . Weiterhin muß  $H(A'_l, B'_k) = d$  gelten. Wäre  $H(A'_l, B'_k) < d$ , so würde implizit auch  $H_{lk}(A, B) < d$  gelten, da  $A'_l$  und  $B'_k$  somit minimierende Teilmengen der Größe  $l$  bzw.  $k$  wären.  $\square$

Theorem 4.2 zeigt, daß die Identität und die Symmetrie als Eigenschaften einer Metrik für  $A'_l$  und  $B'_k$  weiterhin gelten. Die Dreiecksungleichung hingegen ist hier nicht allgemeingültig anwendbar, da sich beim Vergleich der Punktmenge  $A$  und  $B$  andere minimierende Teilmengen ergeben können als bei einem Vergleich zwischen  $B$  und  $C$ . Gilt einschränkend, daß  $B'_k \subseteq B$  sowohl minimierende Teilmenge für  $H_{lk}(A, B)$  als auch für  $H_{km}(B, C)$  ist, so ist die Dreiecksungleichung

$$H(A'_l, C'_m) \leq H(A'_l, B'_k) + H(B'_k, C'_m) \quad (\text{vgl. 4.3})$$

für diesen Sonderfall weiterhin gültig. Für die Objekterkennung bedeutet dies, wenn ein Modell zu zwei verschiedenen Bildausschnitten paßt, so ist der Inhalt der beiden Ausschnitte zueinander ähnlich.

#### 4.4.2 Beschränkte Hausdorff-Distanz

In der Objekterkennung werden eine das Modell repräsentierende Punktmenge mit einer das Bild repräsentierenden Punktmenge verglichen. Oft stellt das aufgenommene Bild nicht nur ein einzelnes Objekt, sondern eine Reihe verschiedener Objekte dar. Ein Großteil der Bildpunkte hat also keinen Bezug zu dem gesuchten Objekt. Es

ist daher auch nicht praktikabel, das zur Verfügung stehende Modell mit dem kompletten Bild zu vergleichen: das Ergebnis wäre wenig aussagekräftig. Vielmehr ist es notwendig, den Bereich der relevanten Bildpunkte so einzugrenzen, daß ‚objekt-fremde‘ Punkte von der Bewertung ausgeschlossen bleiben. Hierfür wurde eine Erweiterung der gerichteten Hausdorff-Distanz zur beschränkten Hausdorff-Distanz<sup>3</sup> vorgenommen. Diese sieht vor, nur die innerhalb eines vorher bestimmten Rahmens liegenden Bildpunkte zur Auswertung heranzuziehen. Die Ausmaße des Rahmens  $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$  ergeben sich aus den Extrema der Modellkoordinaten.

$$h_{\square}(A, B) = \max_{\substack{\{a \in A \mid x_{\min} \leq x_a \leq x_{\max} \wedge \\ y_{\min} \leq y_a \leq y_{\max}\}}} \left( \min_{b \in B} (d(a, b)) \right) \quad (4.5)$$

Die beschränkte Hausdorff-Distanz läßt sich mit der partiellen Hausdorff-Distanz zur partiellen beschränkten Hausdorff-Distanz kombinieren. Dadurch werden nicht alle innerhalb des Rahmens liegenden Bildpunkte, sondern nur die am besten passenden mit denen des Modells verglichen (siehe 4.4.1).

Die partielle beschränkte Hausdorff-Distanz wird in der Objekterkennung ausschließlich als Verifikationsdistanz verwendet, also zur Bestimmung des Abstandes der Bildpunktmenge zur Modellpunktmenge. Im umgekehrten Fall, der Berechnung der Hypothesendistanz, besteht für die Verwendung der beschränkten Hausdorff-Distanz keine Notwendigkeit.

## 4.5 Transformation der Modellpunkte

Die Hausdorff-Distanz, soweit sie bis jetzt vorgestellt wurde, ermöglicht den Vergleich zweier Punktmenge in verschiedenen Variationen. Allerdings sind die Positionen der Punkte in beiden Mengen fix. Interessant wäre jedoch, die Punktmenge gegeneinander verschieben zu können. Schließlich ist nicht zu erwarten, daß die Bildpunkte und die Modellpunkte schon deckungsgleich übereinander liegen. In der Regel ist das gesuchte Objekt nur ein Teil des Bildes und kann sich dort an beliebiger Stelle befinden. Die Modellpunkte müssen also vorher einer Transformation unterzogen werden, um für den Vergleich richtig positioniert zu sein. Im wesentlichen sind die Translation  $T$ , die Skalierung  $S$  bzw. deren Kombination  $ST$  und die affinen Abbildungen  $A$  als Transformationen für die Objektsuche mittels der Hausdorff-Distanz relevant.

Ist  $G$  die Menge von zulässigen Transformationen, so wird die Hausdorff-Distanz unter Transformation folgendermaßen definiert:

$$D_G(g) = H(A, g(B)) \quad (4.6)$$

$$d_G(g) = h(g(B), A) \quad (4.7)$$

$$d'_G(g) = h(A, g(B)) \quad (4.8)$$

$D_G(g)$  ist die ungerichtete Hausdorff-Distanz zwischen Bildpunktmenge  $A$  und der mit  $g \in G$  transformierten Modellpunktmenge  $B$ .  $d_G(g)$  stellt die Hypothesendistanz und  $d'_G(g)$  die Verifikationsdistanz (vgl. 4.2.1) zwischen dem Bild und dem transformierten Modell dar.

Ziel ist es, die Transformation  $g \in G$  zu finden, so daß Bildpunkte und Modellpunkte möglichst dicht zusammen liegen.

$$H_G(A, B) = \min_{g \in G} (D_G(g)) \quad (4.9)$$

$$h_G(B, A) = \min_{g \in G} (d_G(g)) \quad (4.10)$$

<sup>3</sup>Die englischsprachige Literatur verwendet hierfür den Ausdruck *box-reverse Hausdorff distance*.

Die Hausdorff-Distanz unter Transformation als Minimum aller  $d'_G(g)$  ist für die Objekterkennung ohne Belang, da eine mit  $h_G(B, A)$  durch die Transformation  $g$  bestimmte Objekthypothese mit Hilfe der Verifikationsdistanz lediglich bestätigt oder verworfen wird.

### 4.5.1 Translation

Die Translation stellt die einfachste Klasse der Transformationen dar. Im zweidimensionalen Raum, passend zur Bearbeitung von Bilddaten, bedeutet die Translation eine Verschiebung der Punktkoordinaten in  $x$ - und  $y$ -Richtung um  $t_x$  bzw.  $t_y$  Einheiten.

$$P(x, y) \xrightarrow{T} P'(x + t_x, y + t_y) \quad (4.11)$$

Wird eine ganze Punktmenge einer Translation unterzogen, so wird jeder einzelne Punkt der Menge für sich verschoben. Die neuen Punkte  $P'$  repräsentieren die verschobene Punktmenge. Die Definition für die Hausdorff-Distanz lautet wie folgt:

$$H_T(A, B) = \min_{t_x, t_y} (H(A, B \oplus (t_x, t_y))) \quad (4.12)$$

$$B \oplus (t_x, t_y) = \{(b_x + t_x, b_y + t_y) \mid (b_x, b_y) \in B\} \quad (4.13)$$

### 4.5.2 Skalierung

Durch die Skalierung ändert sich die Punktkoordinate abhängig von Koordinatenursprung und den Skalierungsfaktoren  $s_x$  und  $s_y$ .

$$P(x, y) \xrightarrow{S} P'(s_x x, s_y y) \quad (4.14)$$

Dabei ist zu beachten, daß keiner der beiden Skalierungsfaktoren 0 ist. Ein Kollabieren der Punktkoordinaten auf die Nulllinie oder sogar den Nullpunkt wäre die Folge und eine Umkehrung der Transformation damit nicht möglich.

$$H_S(A, B) = \min_{s_x, s_y} (H(A, B \otimes (s_x, s_y))) \quad (4.15)$$

$$B \otimes (s_x, s_y) = \{(s_x b_x, s_y b_y) \mid (b_x, b_y) \in B\} \quad (4.16)$$

Die Skalierung wird oft zusammen mit der Translation verwendet. Diese Kombination der beiden Transformationen wird durch ein 4-Tupel  $(s_x, s_y, t_x, t_y)$  eindeutig festgelegt.

### 4.5.3 Affine Transformation

Die affine Abbildung ermöglicht alle Transformationen, die Punkte in einer Ebene im Raum erfahren können. Eine Rotation, eine Translation und eine einheitliche Skalierung lassen die Ebene aus jedem Blickwinkel im Raum erscheinen. Sie ist daher für planare Modelle besonders geeignet.

$$P(x, y) \xrightarrow{A} P'(m_{00}x + m_{01}y + t_x, m_{10}x + m_{11}y + t_y) \quad (4.17)$$

$$\text{mit } \mathcal{M} = \begin{pmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{pmatrix}$$

$\mathcal{M}$  muß dabei so gewählt werden, daß die Determinante  $(m_{00}m_{11} - m_{10}m_{01})$  größer 0 ist. Eine Determinante gleich 0 hat die Degeneration der Ebene in eine Linie zur Folge und ist somit nicht umkehrbar.

$$H_A(A, B) = \min_{\substack{m_{00}, m_{01}, \\ m_{10}, m_{11}, \\ t_x, t_y}} (H(A, \mathcal{M}(B) \oplus (t_x, t_y))) \quad (4.18)$$

$$\mathcal{M}(B) \oplus (t_x, t_y) = \{(m_{00}b_x + m_{01}b_y + t_x, m_{10}b_x + m_{11}b_y + t_y) \mid (b_x, b_y) \in B\} \quad (4.19)$$

Die affine Transformation wird durch ein 6-Tupel  $(m_{00}, m_{01}, m_{10}, m_{11}, t_x, t_y)$  eindeutig festgelegt.

## 4.6 Diskretisierung

### 4.6.1 Punktkoordinaten

Die Koordinaten der Bildpunkte sind aufgrund der Bildauflösung bereits diskretisiert. Da es sich um eine endliche Menge von möglichen Positionen in  $x$ - und  $y$ -Richtung handelt, liegt es nahe, die  $x$ - und  $y$ -Koordinaten mit Hilfe der natürlichen Zahlen durchnummerieren. Es ergibt sich ein Wertebereich in  $\mathbb{N}^2$ .

Für die Punktkoordinaten des untransformierten Modells gilt eine entsprechende Einteilung. Die Koordinaten der transformierten Modellpunkte können nicht notwendigerweise in  $\mathbb{N}^2$  angenommen werden, da Teile der Transformationen in  $\mathbb{N}^2$  nicht definiert sind. Zur Einhaltung des festgelegten Rasters werden die einzelnen Modellkoordinaten nach der Transformation auf die jeweils nächstgelegenen Punkte in  $\mathbb{N}^2$  gerundet.

### 4.6.2 Transformationsraum

Die in 4.5 beschriebenen Transformationen werden in mehrdimensionalen Räumen eindeutig festgelegt. So ist die Menge der Translationen mit  $(t_x, t_y)$  in  $\mathbb{R}^2$ , die Menge der Kombination aus Skalierung und Translation mit  $(s_x, s_y, t_x, t_y)$  in  $\mathbb{R}^4$  und die Menge der affinen Abbildungen mit  $(m_{00}, m_{01}, m_{10}, m_{11}, t_x, t_y)$  in  $\mathbb{R}^6$  definiert. In Räumen aus  $\mathbb{R}^k$  mit  $k \in \mathbb{N}$  existieren grundsätzlich unendlich viele Punkte. Dabei repräsentiert jeder dieser Punkte in den oben beschriebenen Transformationsräumen  $\mathbb{R}^2$ ,  $\mathbb{R}^4$  und  $\mathbb{R}^6$  eine gültige Transformation. Einzelne Beschränkungen auf einen Skalierungsfaktor größer 0 oder eine positive Determinante haben auf die Mächtigkeit von  $\mathbb{R}^n$  mit  $n \in \{2, 4, 6\}$  keinen Einfluß.

Ist keine Abhängigkeit zwischen den Koordinaten  $(x_1, \dots, x_n)$  bekannt, wie sie beispielsweise durch eine stetige Funktion in  $\mathbb{R}^n$  definiert wäre, so ist eine exakte Bestimmung der Extrema unmöglich. Die Einschränkung der zulässigen Transformationen auf eine abzählbare Menge ist nötig, um den Raum nach Minimum bzw. Maximum durchsuchen zu können.

Wird zusätzlich eine geschickte Einteilung innerhalb der zulässigen Werte gewählt, kann die zu erwartende Änderung der Hausdorff-Distanz nach einer Modifikation der Transformationsparameter in gewissen Grenzen vorausgesagt werden. Die Suche nach der optimalen Transformation läßt sich dadurch auf unterschiedliche Weise gegenüber der erschöpfenden Suche, auch *brute force* genannt, optimieren.

### Grundidee

Wird ein einzelner Transformationsparameter um genau eine Einheit im Transformationsraum geändert, so soll die neue Position aller Modellpunkte höchstens um eine Einheit von der alten abweichen. Dadurch wird erreicht, daß benachbarte Transformationen im Raum nur geringfügig unterschiedliche Modellpunktkoordinaten liefern. Ist die Hausdorff-Distanz für eine Transformation der Modellpunkte bekannt, so können Rückschlüsse auf die Ergebnisse der umliegenden Transformationen gezogen werden.

Eine Änderung des  $n$ -Tupels einer Transformation um genau  $d$  Schritte (beliebig aufgeteilt auf alle Dimensionen) hat eine um höchstens  $d$  verkleinerte Hausdorff-Distanz des neuen Modells zu den Bildpunkten zur Folge. Analog gilt dies für die maximale Vergrößerung der Hausdorff-Distanz, allerdings ist diese Beobachtung für die Suche nach der minimalen Distanz zwischen  $A$  und  $B$  ohne Belang.

### Einteilung

Optimal wird die Einteilung der einzelnen Dimensionen, wenn die Änderung in einer Dimension um einen Schritt alle Modellpunkte um höchstens eine Einheit, jedoch mindestens einen um genau eine Einheit verschiebt.

**Translation** Da sowohl Bildpunkte als auch die untransformierten Modellpunkte jeweils in  $\mathbb{N}^2$  liegen und eine Translation der Modellpunkte auf Koordinaten der Bildpunkte hinreichend ist, ist eine Beschränkung der Translationsparameter  $t_x$  und  $t_y$  auf die Elemente der natürlichen Zahlen sinnvoll. Wird als Schrittweite

$$\Delta t_x = 1 \quad \text{und} \quad \Delta t_y = 1$$

angenommen, so verschieben sich alle Modellpunkte bei jedem Schritt um genau eine Einheit.

**Skalierung** Im Gegensatz zur Transformation hängt die Schrittweite für die Skalierung von der Modellgröße ab. Für  $\Delta s_x$  und  $\Delta s_y$  sind dabei die vom Koordinatenursprung in  $x$ - bzw.  $y$ -Richtung am weitesten entfernt liegenden Modellpunkte entscheidend. Sie werden durch den Skalierungsfaktor am stärksten beeinflusst und erfahren somit die größte Positionsänderung.

Um den für  $\Delta s_x$  relevanten Punkt um genau eine Einheit zu verschieben, muß  $s_x$  um  $\frac{1}{x_{\max}}$  geändert werden. Für die Skalierung ergibt sich damit eine Rasterung mit

$$\Delta s_x = \frac{1}{x_{\max}} \quad \text{und} \quad \Delta s_y = \frac{1}{y_{\max}}.$$

Bei einer Änderung der Skalierung ausgehend von  $s_x = \frac{1}{3}$  verschiebt sich die Position von  $x_{\max}$  nur noch um circa  $\frac{1}{3}$  Einheit. Nach der Rundung auf den nächstgelegenen Punkt in  $\mathbb{N}^2$  bedeutet das nicht für jeden Schritt in der Skalierung eine Änderung der Punktcoordinate.

**Affine Transformation** Ausgehend von den Erkenntnissen aus den Paragraphen Translation und Skalierung können aus der Abbildungsvorschrift für affine Transformation

$$P(x, y) \xrightarrow{A} P'(m_{00}x + m_{01}y + t_x, m_{10}x + m_{11}y + t_y)$$

die Schrittweiten für jede der sechs Dimensionen leicht abgeleitet werden. Die Herleitung für den translativen Teil der Abbildung entspricht der der reinen Translation. Die Argumentation für die Parameter  $m_{00}, \dots, m_{11}$  ist analog zur Skalierung. Damit ergeben sich folgende Einteilungen für die einzelnen Dimensionen des 6-Tupels  $(m_{00}, m_{01}, m_{10}, m_{11}, t_x, t_y)$ :

$$\begin{aligned} \Delta m_{00} &= \frac{1}{x_{\max}}, & \Delta m_{01} &= \frac{1}{y_{\max}}, \\ \Delta m_{10} &= \frac{1}{x_{\max}}, & \Delta m_{11} &= \frac{1}{y_{\max}}, \\ \Delta t_x &= 1 & \text{und} & \Delta t_y = 1 \end{aligned}$$

## 4.7 Vorgehensweise bei der Objektsuche

In Abschnitt 4.2.1 wurden die Hypothesendistanz und die Verifikationsdistanz vorgestellt, die in den Abschnitten 4.4.1 und 4.4.2 zur partiellen Hypothesendistanz

und zur partiellen beschränkten Verifikationsdistanz erweitert worden sind. Motivation für diese Erweiterungen war die robuste Anwendung der Hausdorff-Distanz auf gestörte<sup>4</sup> Bilddaten.

### 4.7.1 Bestimmung der Objekthypothese

Zur Suche nach Objekthypothesen werden zwei Parameter  $f_h$  und  $\tau_h$  eingeführt. Dabei ist  $\tau_h$  der Schwellwert für die partielle Hypothesendistanz und  $f_h$  der Anteil der Modellpunkte, die nach der Transformation höchstens  $\tau_h$  Einheiten von ihrem nächstgelegenen Bildpunkt entfernt liegen dürfen. Aus der Menge aller zulässigen Transformationen werden die ausgewählt, für die

$$h_{f_h \cdot |B|}(g(B), A) \leq \tau_h \quad (4.20)$$

erfüllt ist. Die damit transformierten Modelle legen die Menge der Objekthypothesen fest.

### 4.7.2 Überprüfung der Objekthypothese

Zur Überprüfung der Objekthypothese wird die partielle beschränkte Hausdorff-Distanz verwendet. Analog zur Bestimmung der Objekthypothese werden auch für die Verifikation zwei Parameter  $f_v$  und  $\tau_v$  definiert.  $\tau_v$  ist der Schwellwert für die Verifikationsdistanz,  $f_v$  der Anteil der Bildpunkte, die zum einen innerhalb der vorgegebenen Koordinatengrenzen und zum anderen nicht weiter als  $\tau_v$  Einheiten vom nächstgelegenen transformierten Modellpunkt liegen. Gilt

$$h_{\square f_v \cdot |\{a \in A | x_{\min} \leq x_a \leq x_{\max} \wedge y_{\min} \leq y_a \leq y_{\max}\}|}(A, g(B)) \leq \tau_v, \quad (4.21)$$

so wird die Objekthypothese als bestätigt angesehen. Eine Instanz des Objekts im Bild ist gefunden. Ist (4.21) jedoch nicht erfüllt, so wird die Objekthypothese verworfen.

## 4.8 Optimierung der Suche

Eine erschöpfende Suche im kompletten Suchraum kann extrem zeitaufwendig sein. Die in Abschnitt 4.6 vorgestellte Diskretisierung ermöglicht es, die Hausdorff-Distanz in gewissen Grenzen vorauszusagen. Dies erlaubt, die Suche auf Teilbereiche einzuschränken und somit die Bearbeitungszeit zu verkürzen. Dabei muß aber sichergestellt sein, daß alle Transformationen, die die Kriterien für eine Objekthypothese erfüllen, gefunden werden. Vier Strategien zur Suchoptimierung werden im folgenden kurz vorgestellt.

### 4.8.1 Vorzeitiges Verwerfen

Sowohl zur Suche als auch zur Verifizierung gefundener Objekthypothesen wird die partielle Hausdorff-Distanz verwendet. Generell ist gefordert, daß ein vorher festgelegter Anteil von Modellpunkten innerhalb der Distanz  $\tau$  einen Nachbarn aus der Menge der Bildpunkte hat; analog gilt das gleiche zwischen Bild- und Modellpunkten. Bei der Bestimmung der partiellen Hausdorff-Distanz werden alle Modellpunkte einzeln begutachtet. Ist vorgeschrieben, daß mindestens  $f \cdot |B|$  Punkte innerhalb von  $\tau$  um einen Punkt aus  $A$  liegen, kann die Auswertung für die Hausdorff-Distanz abgebrochen werden, wenn bereits  $(1 - f) \cdot |B|$  Punkte außerhalb von  $\tau$  gefunden wurden. Die geforderte Anzahl von passenden Modellpunkten kann nicht mehr erreicht

<sup>4</sup>Verrauschte Bilddaten, falsch detektierte Kanten und die teilweise Verdeckung von Objekten sind die häufigsten Ursachen für eine fehlgeschlagene Objekterkennung.

werden, also führt diese Transformation zu keiner gültigen Objekthypothese. Bei der Bestimmung der Verifikationsdistanz kann diese Erkenntnis analog angewendet und eine Objekthypothese somit vorzeitig verworfen werden.

### 4.8.2 Überspringen von Transformationen

Ist für eine gültige Transformation im Suchraum die Hypothesendistanz  $h_l(g(B), A)$  berechnet und ist diese größer  $\tau$ , so ist sicher, daß die nächsten  $h_l(g(B), A) - \tau$  benachbarten Transformationen kein Ergebnis kleiner  $\tau$  liefern können, da sich per Definition zu einer benachbarten Transformation alle Modellpunkte höchstens um eine Position verschieben. Die Hausdorff-Distanz kann sich deshalb pro Schritt nur um 1 vergrößern bzw. verkleinern. Ist die Hausdorff-Distanz um  $n$  größer als  $\tau$  und wirken im optimalen Fall alle Schritte verkleinernd, so sind mindestens  $n$  Schritte notwendig, um die Hausdorff-Distanz auf  $\tau$  zu reduzieren. Die Ermittlung der Hausdorff-Distanz für diese Transformationen erübrigt sich, da sie keine gültigen Objekthypothesen ergeben. Sie werden überspringen und die Berechnung wird bei der darauf folgenden Transformation fortgesetzt.

### 4.8.3 Ausschließen von Kugelräumen

Die Erkenntnisse aus 4.8.2 lassen den Schluß zu, daß um eine Transformation  $g$  mit  $h_l(g(B), A) > \tau$  eine Hyperkugel<sup>5</sup> mit Radius  $h_l(g(B), A) - \tau$  existiert, in der alle Transformationen  $g \in G$  eine Hausdorff-Distanz  $h_l(g(B), A) > \tau$  zwischen Bildpunkten und dem transformierten Modell zur Folge haben. Die Auswertung der darin enthaltenen Transformationen wird somit überflüssig und kann unterbleiben.

Wie schon beim Überspringen von Transformationen (siehe 4.8.2) gilt dies jedoch nur für die Hypothesendistanz, da im umgekehrten Fall bei der beschränkten Verifikationsdistanz abhängig von den transformierten Modellpunkten und der daraus resultierenden Verschiebung der Koordinatengrenzen Bildpunkte in Betracht gezogen werden müssen, die vorher für die Berechnung der Hausdorff-Distanz unerheblich waren.

### 4.8.4 Suchraumaufteilung

Das Verfahren der Suchraumaufteilung basiert auf der Existenz von Kugelräumen. Dabei werden die Kugelräume durch Hyperwürfel angenähert. Das Aufteilen des Suchraums in sich nicht überlappende Hyperwürfel ist wesentlich einfacher als die Darstellung des Suchraums durch Hyperkugeln. Auch die weitere Aufteilung der entstehenden Hyperkugeln würde sich schwierig gestalten.

Der Suchraum  $(x_1, x_2, \dots, x_n)$  wird durch einen Hyperwürfel  $H$  dargestellt. Dabei ist  $g_c$  die Transformation im Zentrum des Hyperwürfels und  $d_c$  der Abstand von  $g_c$  zu den Eckpunkten von  $H$ .

$$H = [x_{1\min}, x_{1\max}] \times [x_{2\min}, x_{2\max}] \times \dots \times [x_{n\min}, x_{n\max}] \quad (4.22)$$

$$g_c = \left( \frac{x_{1\min} - x_{1\max}}{2}, \frac{x_{2\min} - x_{2\max}}{2}, \dots, \frac{x_{n\min} - x_{n\max}}{2} \right) \quad (4.23)$$

$$d_c = d \left( \frac{x_{1\min} - x_{1\max}}{2}, \frac{x_{2\min} - x_{2\max}}{2}, \dots, \frac{x_{n\min} - x_{n\max}}{2} \right) \quad (4.24)$$

Wird nun die Hausdorff-Distanz  $h_l(g_c(B), A)$  berechnet und ist diese größer  $\tau + d_c$ , so kann dieser Hyperwürfel keine zulässige Transformation enthalten. In diesem Fall wird der Hyperwürfel aus dem Suchraum ausgeschlossen. Deutet  $h_l(g_c(B), A) \leq$

<sup>5</sup>Eine Hyperkugel ist eine Kugel in einem höherdimensionalen Raum

$\tau + d_c$  an, daß in diesem Hyperwürfel möglicherweise eine gültige Transformation existiert, so wird dieser unterteilt. Dazu wird jede Dimension des Hyperwürfels in  $r$  Teile aufgeteilt, es ergeben sich  $r^d$  Subwürfel mit  $d = \text{Anzahl der Dimensionen}$ . Für jeden dieser Subwürfel wird  $g_c$  und  $d_c$  neu bestimmt, die Hausdorff-Distanz berechnet und mit  $\tau + d_c$  verglichen.

Sind alle Kanten des Hyperwürfels etwa gleich lang, so fällt  $d_c$  nicht unnötig groß aus. Da  $d_c$  der Radius der umschreibenden Hyperkugel ist, kann es vorkommen, daß ein Hyperwürfel als interessant eingestuft und weiter unterteilt wird, obwohl die  $h_l(g_c(B), A) \leq \tau + d_c$  erfüllende Transformation nicht in  $H$  selbst, sondern nur in der umspannenden Hyperkugel existiert. Dies läßt sich zwar auch im Idealfall, einem Hyperraum mit gleicher Länge für alle Kanten, nicht vermeiden, jedoch treten solche Fehlentscheidungen bei der Suchraumaufteilung weitaus seltener auf als bei einem Hyperwürfel mit ungleicher Kantenlänge.

Die Reihenfolge, mit der die entstehenden Teilräume untersucht werden, ist für die Suche nach gültigen Transformationen ohne Belang. Zwei gängige Strategien sind *breadth first search* und *depth first search*, die die Teilräume entweder vorrangig in der Breite<sup>6</sup> oder in der Tiefe<sup>7</sup> durchsuchen.

## 4.9 Spezialisierte Suche

An die Suche nach Objekten können verschiedene Anforderungen gestellt werden. Eine Spezialisierung der Suche ist daher möglich. Üblicherweise wird die Suche spezialisiert, um die Bearbeitungsgeschwindigkeit oder die Güte des Resultats zu optimieren. Im folgenden sind einige davon vorgestellt.

### 4.9.1 Einzelnes Objekt

Für die Objektsuche kann es ausreichend sein, die Existenz mindestens eines Objekts im Bild nachzuweisen. Diese Form der Spezialisierung kann die gesamte Bearbeitungszeit verkürzen, da sich die weitere Suche erübrigt, sobald ein Objekt im Bild gefunden wurde.

Konkret gilt also bei der Anwendung der Hausdorff-Distanz: ist eine Objekthypothese gefunden und wird diese durch die Verifikationsdistanz innerhalb der zulässigen Grenzen bestätigt, so wird die Suche abgebrochen und die ermittelte Transformation für die Modellpunkte ausgegeben.

### 4.9.2 Bestes Objekt

Eine Erweiterung der in 4.9.1 dargestellten Spezialisierung stellt die Suche nach dem Objekt dar, das der Beschreibung des gesuchten Objekts am besten entspricht.

Die Hausdorff-Distanz mißt die Übereinstimmung der Lage zwischen den Bildpunkten des gefundenen Objekts und den Modellpunkten der Objektbeschreibung anhand des Anteils der Modell- bzw. Bildpunkte, der die geforderte Höchststanz einhält. Es ergeben sich daher zwei Möglichkeiten, die Güte der Übereinstimmung zu werten.

#### Größter Anteil

Die erste Möglichkeit besteht darin, die zu erfüllende Distanz  $\tau$  zwischen Bild- und Modellpunkten konstant zu halten und das Suchergebnis dahingehend zu optimie-

<sup>6</sup>Breitensuche bedeutet, daß in einem Suchbaum alle Knoten einer Tiefe untersucht werden, bevor ein Knoten der nächsten Tiefe bearbeitet wird.

<sup>7</sup>Tiefensuche bedeutet, daß in einem Suchbaum alle Nachfolger eines Knotens untersucht werden, bevor ein weiterer Knoten derselben Tiefe bearbeitet wird.

ren, daß möglichst viele Punkte, festgelegt durch  $f$ , innerhalb der tolerierten Distanz liegen.

Am Anfang der Suche werden also  $\tau_v$  und  $f_v$  für die Verifikation, sowie  $\tau_h$  und ein Startwert für  $f_h$  festgelegt.  $\tau_h$  und  $f_h$  sind die Mindestanforderungen für eine erste Objekthypothese. Wurde eine Hypothese gefunden und durch die Verifikation bestätigt, so wird die verwendete Transformation der Modellpunkte als zur Zeit optimale Lösung gespeichert. Ist  $f'_h$  der Anteil der passenden Modellpunkte zu dem gefundenen Objekt, so wird  $f_h > f'_h$ , üblicherweise  $f_h = f'_h + 1E$  mit  $1E$  gleich ein Schritt in der Rasterung, gewählt. Damit wird erreicht, daß Objekthypothesen nur noch dann als gültig angenommen werden, wenn diese besser sind als die der bereits gefundenen Lösung.

Entspricht der Anteil der passenden Modellpunkte 100%, so kann wahlweise die Hypothesendistanz nach und nach verringert (siehe folgenden Abschnitt) oder die Suche abgebrochen werden.

### **Geringste Distanz**

Wird die Objektsuche nach der geringsten Distanz zwischen Bild- und Modellpunkten optimiert, so wird ein zur Suche nach dem größten Anteil analoges Verfahren angewendet. Hierbei wird allerdings  $f_h$  als fixer Wert festgelegt und für  $\tau_h$  ein Startwert definiert, der während der Objektsuche sukzessiv reduziert wird.

Ist die Distanz zwischen den zu berücksichtigenden Punkten gleich 0, so kann entweder die Suche abgebrochen oder eine Erhöhung des passenden Anteils angestrebt werden.

# Kapitel 5

## Auswahl der Verfahren

In diesem Kapitel werden sämtliche Schritte des Objekterkennungsprozesses anhand eines Beispielszenarios vorgestellt und genau erläutert. Die Auswahl der einzelnen Werkzeuge wird erörtert. Alle Werte für Parameter, die die Wirkungsweise der Werkzeuge oder deren Empfindlichkeit beeinflussen und zur Abstimmung des gesamten Prozesses verwendet wurden, sind angegeben. Sie wurden experimentell bestimmt und können leicht an unterschiedliche Gegebenheiten, wie sie etwa durch den Wechsel der Kamera oder des Framegrabbers auftreten, angepaßt werden.

### 5.1 Objekterkennungsstrategie

In Abschnitt 3.4 und in Kapitel 4 wurden einige Verfahren zur Objekterkennung vorgestellt. Größtenteils waren dies auf Kombinatorik basierende Ansätze. Die meisten dieser kombinatorischen Methoden haben jedoch einige entscheidende Nachteile. Enthält ein Bild eine große Menge an Merkmalen oder bzw. und ist das Modell des gesuchten Objekts aus zahlreichen Merkmalen aufgebaut, so ist die Verknüpfung dieser Merkmale ein rechenaufwendiger Prozeß. Schnell ist ein Maß erreicht, das eine Objekterkennung in einem vernünftigen Rahmen unmöglich macht. Ein zweiter großer Nachteil kombinatorischer Verfahren ist, daß sie auf Fehler bei der Extraktion der Bildmerkmale empfindlich reagieren. Die Folge ist oft, daß kein Objekt im Bild gefunden wird oder die ermittelte Lage nicht korrekt ist. Solange also keine zuverlässige Merkmalsextraktion möglich ist, die korrekte und konstante Ergebnisse liefert, ist eine robuste Objekterkennung mit kombinatorischen Verfahren schwierig.

Wesentlich unempfindlicher gegen Störungen bei der Merkmalsextraktion sind Verfahren, die die Korrelation von Modell und Bild auswerten. Gerade die Hausdorff-Distanz, deren Anwendung auf dem Gebiet der Objekterkennung noch verhältnismäßig jung ist, zeigt sich außerordentlich robust gegen Fehler bei der Merkmalsextraktion sowie auch gegenüber teilweiser Verdeckung des gesuchten Objekts. Auch sind Korrelationsverfahren in der Regel leistungsfähiger, was die Menge der zu verarbeitenden Daten angeht. Eine große Anzahl von Merkmalen sowohl im Bild als auch für das Modell ist für das Hausdorff-Distanz-Verfahren keine Herausforderung.

Für den hier vorgestellten Objekterkennungsprozeß wurde die Hausdorff-Distanz verwendet, um aus einem Bild extrahierte Merkmale nach der Existenz eines in der Modelldatenbank gespeicherten Objekts zu durchsuchen. Die Modellierung der Datenbank und die Extraktion der Bildmerkmale wurde auf die Verwendung der Hausdorff-Distanz abgestimmt und wird im folgenden genau erläutert.

## 5.2 Modelldatenbank

Für die Modelldarstellung bei der Objektsuche durch die Minimierung der Hausdorff-Distanz wird ein Punktearray, also ein binäres Bild, verwendet. Dabei gehört jeder mit 1 belegte Punkt in diesem Bild zu einer Kante des gesuchten Objekts. Alle mit 0 belegten Punkte sind für die Objekterkennung nicht relevante Bereiche.

Größenänderungen des gesuchten Objekts im Bild werden durch die proportionale Skalierung des Modells in  $x$ - und  $y$ -Richtung kompensiert. Da sich das gesuchte Objekt jedoch nicht in einer zur Kamera konstanten Entfernung befindet, sondern vielmehr an jedem Punkt des in der Regel schief im Kamerakoordinatenraum liegenden Fußbodens aufgestellt sein kann, muß auch die Änderung der Objektansicht berücksichtigt werden. Der Kamera nahe Objekte werden eher unter einem steilen Winkel, also von oben abgebildet. Weiter entfernte Objekte jedoch bildet die Kamera unter einem flachen Winkel und somit eher von vorne ab. Die Hausdorff-Distanz ist durch ihre Definition in der Lage, eine gewisse Abweichung zwischen der tatsächlichen Objektdarstellung im Bild und der in der Modelldatenbank abgelegten Ansicht zu tolerieren.

Die zu erfüllende Gesamtaufgabe fordert das Erkennen und Lokalisieren des gesuchten Objekts und ein anschließendes Andocken an das gefundene Objekt. Da die Andockvorrichtung nur aufrecht stehende Papierkörbe greifen kann, erübrigt sich die Erkennung von Papierkörben in einer nicht aufrechten Lage. Diese Einschränkung minimiert die Menge der möglichen Ansichten auf einige wenige.

Da das Objekt für einen festen Kamerawinkel durch eine begrenzte Zahl von Ansichten repräsentiert werden kann, ist es praktikabel, in der Modelldatenbank für jedes Modell die Kamerawinkel zu vermerken, unter denen das Objekt so erscheint. Anhand dieser Indizierung können für die aktuelle Konfiguration des Roboters alle Modelle ausgeschlossen werden, die keine aus diesem Blickwinkel gültige Ansicht darstellen.

Für die Modelldatenbank, die zur Demonstration der hier diskutierten Problemlösung verwendet wird, wurde auf diese Indizierung verzichtet, da von den hier verwendeten Kameraaufnahmen keine Daten über den Aufnahmewinkel zur Verfügung standen.

Abbildung 5.1 zeigt die Modelldatenbank für den hier dargestellten Fall.

## 5.3 Merkmalsextraktion

Das gesuchte Objekt wird durch die Minimierung der Hausdorff-Distanz zwischen der Bildpunktmenge und der Modellpunktmenge im Bild lokalisiert. Dazu muß das Originalbild so modifiziert werden, daß entscheidende Merkmale aus dem Bild extrahiert und diese für die Berechnung der Hausdorff-Distanz aufbereitet sind. Ziel der Merkmalsextraktion wird daher sein, ein Kantenbild zu konstruieren, daß nach Möglichkeit nur die Kanten enthält, deren Zugehörigkeit zum gesuchten Objekt wahrscheinlich ist. Da bei der Berechnung der Hausdorff-Distanz ausschließlich der Abstand einzelner Punkte zu deren nächstgelegener Kante benötigt wird, wird das Endergebnis der Merkmalsextraktion die Distanztransformierte des reduzierten Kantenbildes sein.

Anhand des in Abbildung 5.2 dargestellten Szenarios werden alle Schritte der Merkmalsextraktion im einzelnen vorgestellt und gegebenenfalls deren Motivation erläutert. Einen Überblick über den gesamten Prozeß und die interne Abhängigkeit der einzelnen Schritte zeigt Abbildung 5.3.

Auf die Abbildung des geglätteten Originalbildes wird hier verzichtet, da zwar der Einfluß der Glättung auf einen kantenextrahierenden Bildoperator beträchtlich

ist, jedoch der Unterschied zum Originalbild vom menschlichen Auge kaum wahrgenommen wird.

### 5.3.1 Bildglättung

Zur Bildglättung wurde ein  $3 \times 3$ -Medianfilter verwendet, dessen Arbeitsweise in Abschnitt 2.4.2 bereits vorgestellt wurde. Dieses häufig verwendete Filter arbeitet kantenerhaltend und glättet verrauschte Bereiche konstanter Färbung. Starkes Rauschen in konstanten Bereichen des Bildes führt bei der Bearbeitung mit einem Kantenoperator zur Erzeugung falscher Kanten, die eine Objekterkennung extrem stören können. Außerdem werden bei der Medianfilterung im Gegensatz zu anderen Glättungsoperatoren existierende Kanten im Bild nicht verschmiert.

### 5.3.2 Farbmodell

Als Modell für die Farbdarstellung im Bild wurde das in 2.2.2 vorgestellte RGB-Farbmodell verwendet. Der verwendete Framegrabber liefert das Bild bereits im RGB-Format, eine Konvertierung der gesamten Bilddaten ist nicht notwendig. Ein weiteres Argument für den Vorzug des RGB-Farbmodells gegenüber den übrigen ergab sich bei der Betrachtung der räumlichen Gegebenheiten des geplanten Einsatzbereiches. Aufgabe ist es, einen roten Papierkorb, der sich in aufrechter Lage auf einem einheitlich grünen Fußboden befindet, zu erkennen und zu lokalisieren. Da Grün die Komplementärfarbe zu Rot ist, stellt der Papierkorb zum Fußboden einen starken Kontrast dar. Im Rot-Kanal des Farbbildes ist der Papierkorb hell, der Fußboden dunkel abgebildet, analog ist im Grün-Kanal des Farbbildes der Papierkorb dunkel und der Fußboden hell dargestellt. Wird anstatt des Grauwertbildes nun der Rot- bzw. Grün-Kanal des Farbbildes als Eingabebild zur Kantenextraktion verwendet, so erscheinen die Kanten des Papierkorbes im Bereich des Fußbodens besonders intensiv.

### 5.3.3 Farbklassifizierung

Durch eine Farbklassifizierung können im Bild Bereiche bestimmt werden, in denen die Existenz des gesuchten Objekts wahrscheinlich ist. Da a priori die Farbe des gesuchten Objekts bekannt ist, kann anhand dieser Information eine Auswertung des Bildes erfolgen.

Folgende Bewertung der Farbwerte  $R$ ,  $G$  und  $B$  hat sich als zuverlässiges Kriterium für die Bestimmung von interessanten Bereichen bewährt:

$$1.2 \cdot \frac{G + B}{2} < R \quad (5.1)$$

$$|G - B| < 15 \quad (5.2)$$

Die Auswertung von Intensitätsverhältnissen zwischen verschiedenen Farbkanalkombinationen in (5.1) führt zu einer helligkeitsunabhängigen Bewertung der dargestellten Farbe. Die zulässige maximale Differenz zwischen  $G$  und  $B$ , die in (5.2) überprüft wird, bleibt über den gesamten Helligkeitsbereich konstant.

Abbildung 5.4 zeigt die Bereiche, die anhand der Farbklassifizierung als interessant eingestuft wurden. Uninteressante Bereiche sind durch eine Schwellwertbildung schwarz dargestellt.

Kanten, die innerhalb dieser interessanten Bereiche liegen, werden als Bildmerkmale für die Suche nach dem Objekt verwendet. Um zu vermeiden, daß bei einer nicht optimalen Farbklassifizierung Kanten unberücksichtigt bleiben, obwohl

sie Teil des gesuchten Objektes sind, wird die Eliminierung der Kanten so eingeschränkt, daß Kantenpixel, deren euklidische Distanz zum nächstgelegenen Farbsegment höchstens  $n$  beträgt, gültige Bildmerkmale repräsentieren. Eine Distanztransformation der Farbklassifizierung (siehe Abb. 5.5) erleichtert deren Kombination mit der Kantenextraktion des Originalbildes.

### 5.3.4 Kantenextraktion

Als Bildmerkmale bei der Objektsuche durch das Hausdorff-Distanz-Verfahren werden die Kanten aus dem Eingabebild verwendet. Verarbeitet werden dabei nicht ganze Kantensegmente, sondern eine Menge von einzelnen Bildpunkten, die Teil der Kanten im Bild sind.

Verfahren zur Kantenextraktion wurden in Abschnitt 2.6 bereits diskutiert. Zur Merkmalsextraktion wurde hier der Sobel-Operator eingesetzt. Dieser relativ einfache, jedoch oft verwendete Kantenextraktor liefert ein Bild, wie es Abbildung 5.6 zeigt. Die Kanten werden abhängig von ihrer Härte in unterschiedlicher Intensität dargestellt.

Das aus Graustufen aufgebaute Kantenbild muß für die weitere Verarbeitung in ein binäres Bild umgewandelt werden. Dazu werden die Graustufen auf zwei Helligkeitsstufen, vorzugsweise 0 und 1, reduziert.

Da auch in für das menschliche Auge leeren Bereichen des Kantenbildes Helligkeitsunterschiede mit lokalen Maxima existieren, ist eine Binarisierung durch einen adaptiven Schwellwert (siehe 2.7.2) wenig praktikabel. In leeren Bereichen würden diese relativ kleinen lokalen Maxima fälschlicherweise zur Darstellung von Kantenpunkten führen.

Auch die Berechnung eines variablen Schwellwertes (siehe 2.7.3) aus dem Kantenbild ist nicht sinnvoll, da der Hintergrund, sofern er als solcher bezeichnet werden kann, grundsätzlich monoton schwarz ist und keinen verwertbaren Helligkeitsunterschied aufweist. Die Herleitung des variablen Schwellwertes aus dem Originalbild ist ebenfalls nicht sinnvoll. So können in über- bzw. unterbelichteten Bereichen des Bildes ebenso starke Kanten auftreten wie in korrekt belichteten Teilen. Diese werden im Kantenbild mit gleicher Intensität abgebildet. Ein von der Helligkeitsverteilung im Originalbild abhängiger Schwellwert würde Kanten in hellen Bereichen nur schwach, Kanten in dunklen Bereichen übermäßig detailliert darstellen.

Die Verwendung eines festen Schwellwertes (siehe 2.7.1) auf das normalisierte Kantenbild hat sich als ausreichend gut erwiesen. Die Binarisierung des Kantenbildes (Abb. 5.6) mit einem Schwellwert  $c$  gleich 45 zeigt Abbildung 5.7.

### 5.3.5 Kombination

Um das binarisierte Kantenbild so zu reduzieren, daß nur die Merkmale erhalten bleiben, für die ein Zusammenhang mit dem gesuchten Objekt wahrscheinlich ist, werden anhand der aus der Farbklassifizierung gewonnenen Information Teile des Kantenbildes ausgeblendet.

Wie schon in Abschnitt 5.3.3 erläutert, sollen die Teile des Kantenbildes, die ausreichend nah an gültigen Farbsegmenten liegen, bei der Objektsuche berücksichtigt werden. Zur Bestimmung der Distanz eines Bildpunktes zum nächstgelegenen Farbsegment wurde daher die Distanztransformierte der Farbklassifizierung berechnet. Die Reduktion des Kantenbildes, die als Schwellwertbildung im Kantenbild anhand der berechneten Distanztransformation der Farbsegmentierung angesehen werden kann, wird mathematisch wie folgt ausgedrückt:

$$f'(x, y) = \begin{cases} f(x, y), & \text{falls } g(x, y) \leq n \\ 0, & \text{sonst} \end{cases} \quad (5.3)$$

Dabei ist  $f(x, y)$  das ursprüngliche Kantenbild,  $f'(x, y)$  das reduzierte Kantenbild und  $g(x, y)$  die Distanztransformierte der Farbklassifizierung.  $n$  ist der maximal zulässige Abstand eines Kantenpunktes zum nächsten Farbsegment. Abbildung 5.8 zeigt das reduzierte Kantenbild für  $n = 14$ .

Zur Bemessung von  $n$  ist anzumerken, daß ein zu groß gewählter Schwellwert zusätzliche Kantenpunkte in das reduzierte Bild einbringt: ein leicht erhöhter Zeitaufwand bei der Objektsuche ist die Folge. Ein zu gering gewählter Wert für  $n$  hingegen kann dazu führen, daß Teile des Kantenbildes, die für eine korrekte Erkennung wichtig wären, bei der Reduktion ausgeblendet werden. Die Zuverlässigkeit des gesamten Objekterkennungsprozesses wäre durch ein zu klein gewähltes  $n$  gefährdet. Daher wurde hier eine großzügige, jedoch dennoch verhältnismäßige Bemessung für  $n$  vorgenommen.

### 5.3.6 Distanztransformation

Zur Bestimmung der Hausdorff-Distanz zwischen zwei Punktmenge ist es hilfreich, auf die Distanztransformierten der beiden Punktmenge zurückzugreifen. Bei der Objektsuche werden für verschiedene Transformationen des Modells Hypothesendistanz und Verifikationsdistanz zwischen Modell- und Bildpunktmenge berechnet. Da die Anordnung der Modellpunkte zueinander von den Transformationsparametern abhängt, ist a priori eine Berechnung der Distanztransformation der Modellpunktmenge nicht ohne weiteres möglich. Für die Bildpunktmenge hingegen ist eine einmalige Berechnung der Distanztransformation im voraus leicht und für den Durchsatz der Suche durchaus förderlich, da die Bildpunkte im Rahmen der Objektsuche keine Transformation erfahren.

Jeder Bildpunkt der Distanztransformation erhält den Wert der euklidischen Distanz zum nächstgelegenen Kantenpunkt, Kantenpunkte selbst erhalten den Wert 0. Um Werte  $d \notin \mathbb{N}$  in  $\mathbb{N}$  ausreichend genau darzustellen, wird für jeden Punkt nicht  $d$  selbst, sondern  $100d$  gespeichert. Damit sind die Entfernungen auf zwei Stellen nach dem Komma genau für eine Berechnung der Hausdorff-Distanz verfügbar. Wichtig bei der Rundung ist, daß die gespeicherte Entfernung grundsätzlich kleiner ist als die tatsächlich berechnete, andernfalls könnten bei einigen Verfahren zur Bestimmung der optimalen Transformation gültige Lösungen vorzeitig verworfen werden, da zwar die genäherte Distanz größer der maximal zulässigen ist, jedoch die tatsächliche noch innerhalb der Toleranzgrenzen liegen würde.

Zur Berechnung der Distanztransformation wurde, angelehnt an die Ausführungen in Abschnitt 2.8.2, folgende Belegung für die Matrizen verwendet:

$$\begin{aligned} \mathcal{M}_{\setminus} &= \begin{pmatrix} -1 & 223 & -1 \\ 223 & 141 & 100 \\ -1 & 100 & 0 \end{pmatrix} & \mathcal{M}_{/} &= \begin{pmatrix} -1 & 100 & 0 \\ 223 & 141 & 100 \\ -1 & 223 & -1 \end{pmatrix} \\ \mathcal{M}_{\setminus} &= \begin{pmatrix} 0 & 100 & -1 \\ 100 & 141 & 223 \\ -1 & 223 & -1 \end{pmatrix} & \mathcal{M}_{/} &= \begin{pmatrix} -1 & 223 & -1 \\ 100 & 141 & 223 \\ 0 & 100 & -1 \end{pmatrix} \end{aligned} \quad (5.4)$$

Abbildung 5.9 zeigt die Distanztransformation des reduzierten Kantenbildes (siehe Abb. 5.8). Zum Zweck der Darstellbarkeit wurde die tatsächliche Distanztransformation auf einen Wertebereich von  $[0, 255]_{\in \mathbb{N}}$  normalisiert. Regulär übersteigen die Werte einer Distanztransformation für ein Bild dieser Größe einen solchen Wertebereich um ein Vielfaches. Bei der Darstellung der ursprünglichen Distanztransformation würden alle Werte in einem Restesystem modulo 256 abgebildet und das resultierende Bild wäre wenig aussagekräftig.

Je heller ein Punkt in der abgebildeten Distanztransformation erscheint, umso weiter entfernt liegt dieser vom nächsten Kantenpunkt.

## 5.4 Objektsuche

Bei der Objektsuche wird nach der größtmöglichen Korrelation zwischen den Kantenpunkten des Bildes und dem Modell gesucht. Das Maß für die Güte der Korrelation ist die Hausdorff-Distanz. Durch Translation und Skalierung des Modells soll der Abstand zwischen Modellpunkten und Kantenpunkten im Bild minimiert werden. Dies führt zu einem vier-dimensionalen Suchraum, jeweils eine Dimension für die Translation in  $x$ - und  $y$ -Richtung und jeweils eine für die Skalierung von  $x$  und  $y$ .

Für jede Transformation im Suchraum wird die partielle gerichtete Hausdorff-Distanz von Modell zu Bild, die Hypothesendistanz berechnet. Liegt diese innerhalb der Toleranzgrenzen, so bedeutet dies, das Modell hat in dieser Transformation eine ausreichende Übereinstimmung mit dem Bild. Um aber eine Ähnlichkeit zwischen Modell und Bild sicherzustellen, muß umgekehrt auch überprüft werden, ob das Bild im durch die Transformation festgelegten Bereich mit dem Modell eine ausreichend gute Übereinstimmung aufzeigt. Dazu wird die partiell beschränkte gerichtete Hausdorff-Distanz von Bild zu Modell, die Verifikationsdistanz bestimmt und das Ergebnis mit den zulässigen Grenzwerten verglichen. Ergibt dieser Test, daß eine Übereinstimmung gegeben ist, wird der Grad der Ähnlichkeit zusammen mit der verwendeten Transformation als vorläufiges Ergebnis gespeichert.

Da für die Erfüllung der gestellten Aufgabe, dem Andocken an einen Papierkorb, die Kenntnis über die Existenz eines einzelnen Objektes im Bild ausreichend ist und ein zweites Objekt ohnehin nicht zur gleichen Zeit bearbeitet werden könnte, wurde die Suche so konstruiert, daß als Endergebnis die Transformation ausgegeben wird, die zur besten Übereinstimmung zwischen Bild und Modell geführt hat.

Konkret wurden als Mindestanforderung für die Hypothesendistanz  $f_h = 0.8$  und  $\tau_h = 200$  gewählt. Für die Verifikationsdistanz gelten während der gesamten Objektsuche  $f_v = 0.8$  und  $\tau_v = 200$  als fixe Werte. Wurde eine Transformation gefunden, die diese Anforderungen erfüllt, so wird  $\tau_h$  sukzessive, aus *performance*-Gründen jedoch höchstens zehnmals, reduziert und die Suche unter verschärften Bedingungen fortgesetzt, bis keine bessere Transformation mehr gefunden werden kann oder eine Transformation gefunden wurde, für die  $\tau_h$  bereits 0 ist.

Einschränkend gilt bei der Suche, daß eine Skalierung des Modells auf weniger als 30% seiner ursprünglichen Größe nicht zulässig ist. In der Realität variiert das gesuchte Objekt in einer festgelegten Ansicht auch nur zu einem gewissen Grad und schrumpft in seiner Größe nicht gegen 0. Weiterhin beschränken sich die Transformationen im Suchraum, die als Suchergebnis akzeptiert werden, auf solche, bei denen das Verhältnis  $v$  der Skalierungsfaktoren  $s_x$  und  $s_y$  mindestens 0.95 beträgt.  $v$  berechnet sich dabei wie folgt:

$$v = \frac{\min(s_x, s_y)}{\max(s_x, s_y)} \quad (5.5)$$

Die Suche nach der optimalen Transformation erfolgt nach dem Verfahren der Suchraumaufteilung wie in Abschnitt 4.8.4 beschrieben. Enthält die Modelldatenbank mehr als ein Modell, so ist anfangs in der Zelliste für jedes Modell ein eigener Suchraum initialisiert. Werden bei der Suche durch eine gültige Transformation die Suchkriterien verschärft, so werden alle Suchräume in der Zelliste entfernt, die diesen neuen Bedingungen nicht genügen. Da die Suche nicht für jedes Modell separat, sondern für alle Modelle gleichzeitig erfolgt, bedeutet dies eine Verkürzung der gesamten Rechenzeit gegenüber einer sequentiellen Bearbeitung der einzelnen Modelle.

Die Berechnung der Hypothesen- und Verifikationsdistanz wird durch vorzeitiges Verwerfen einer Transformation beschleunigt, wenn feststeht, daß diese die geforderten Kriterien nicht erfüllt (siehe 4.8.1).

Abbildung 5.10 zeigt die Originalszene, in die das transformierte Modell eingebildet wurde.

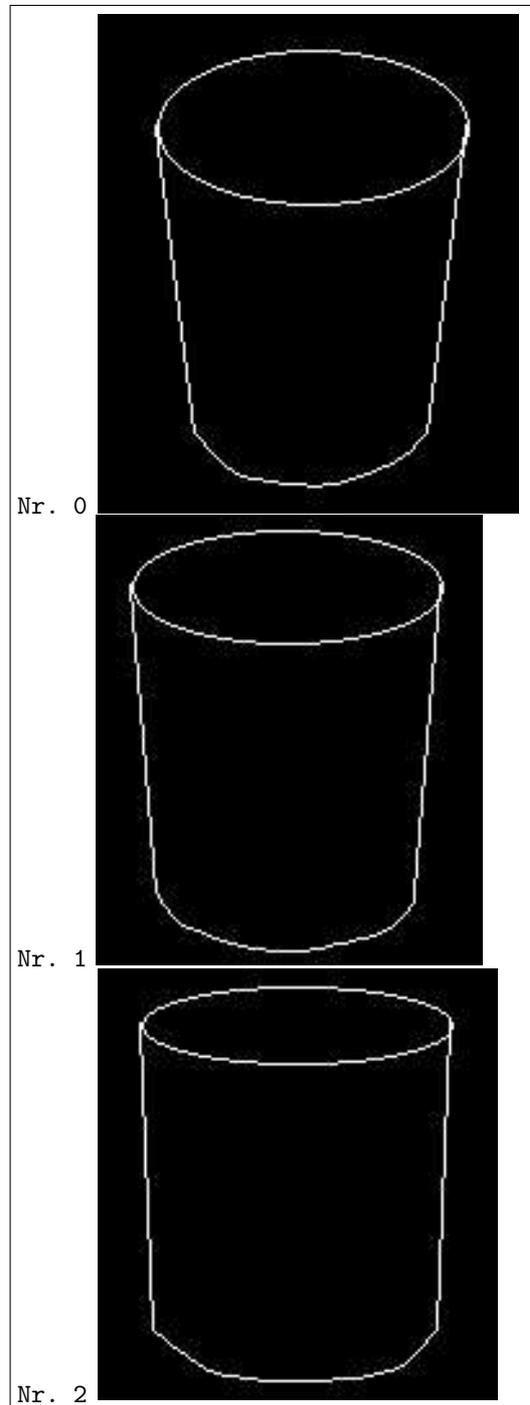


Abbildung 5.1: Die Modelldatenbank



Abbildung 5.2: Szenario I – Originalaufnahme

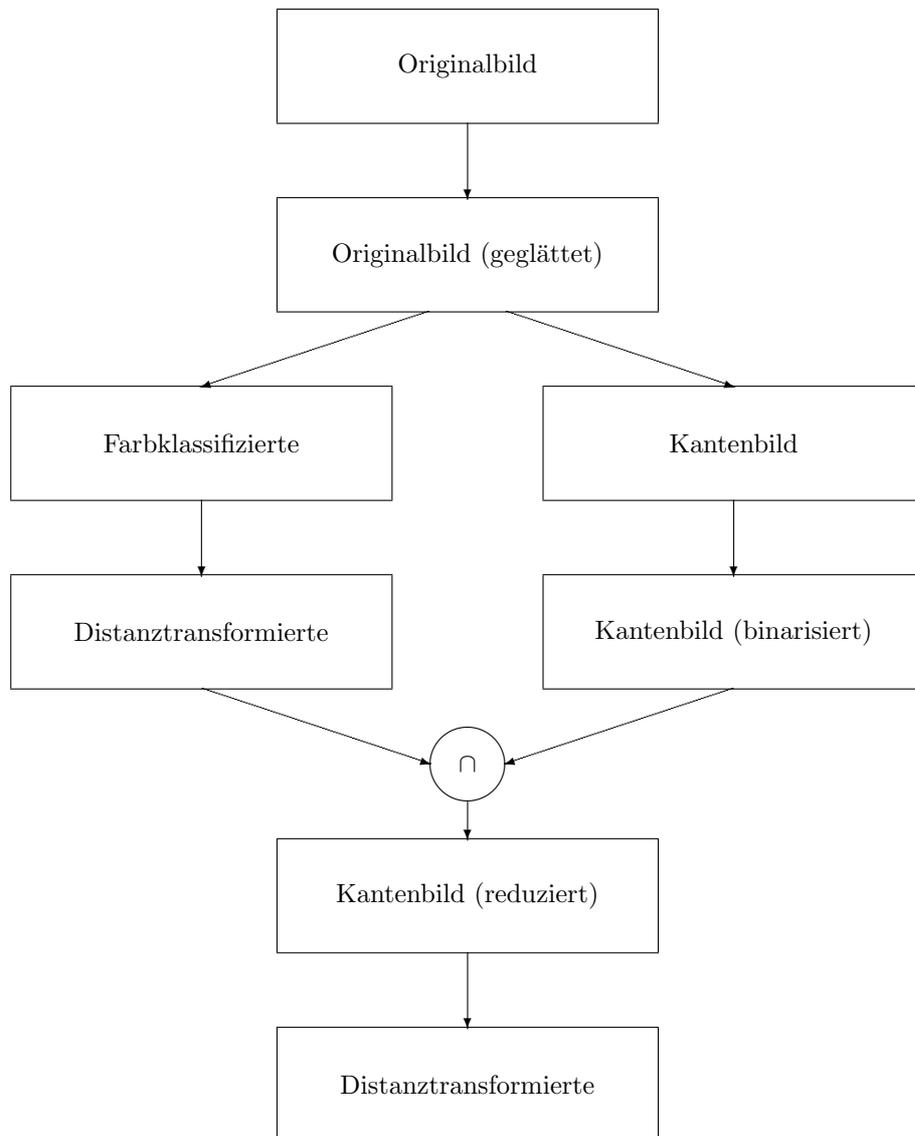


Abbildung 5.3: Der gesamte Prozeß der Merkmalsextraktion im Überblick

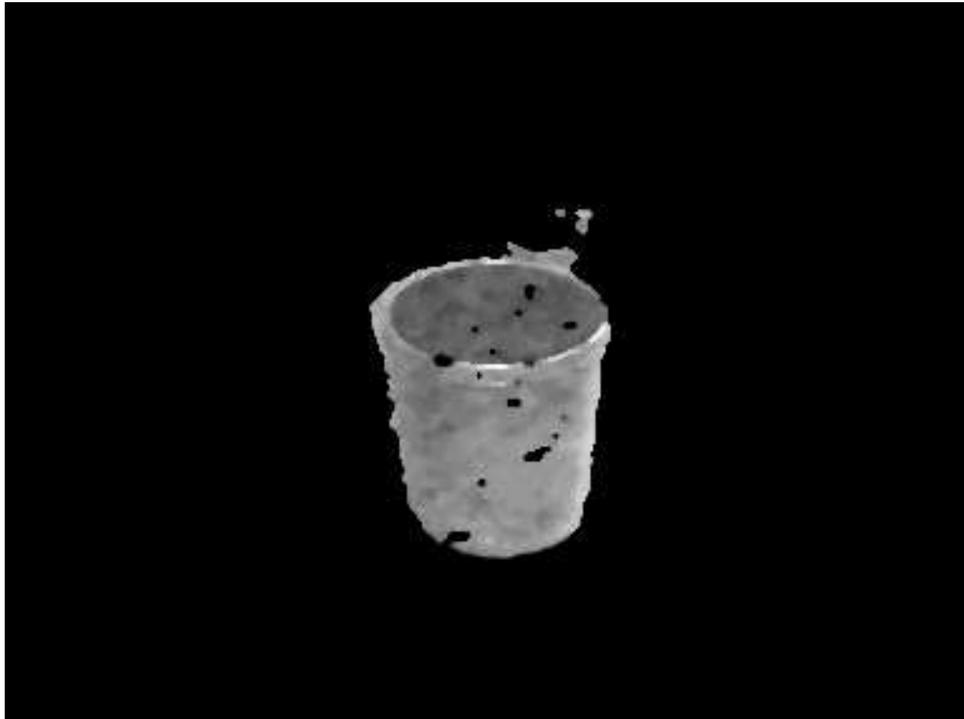


Abbildung 5.4: Szenario I – Ergebnis der Farbklassifizierung

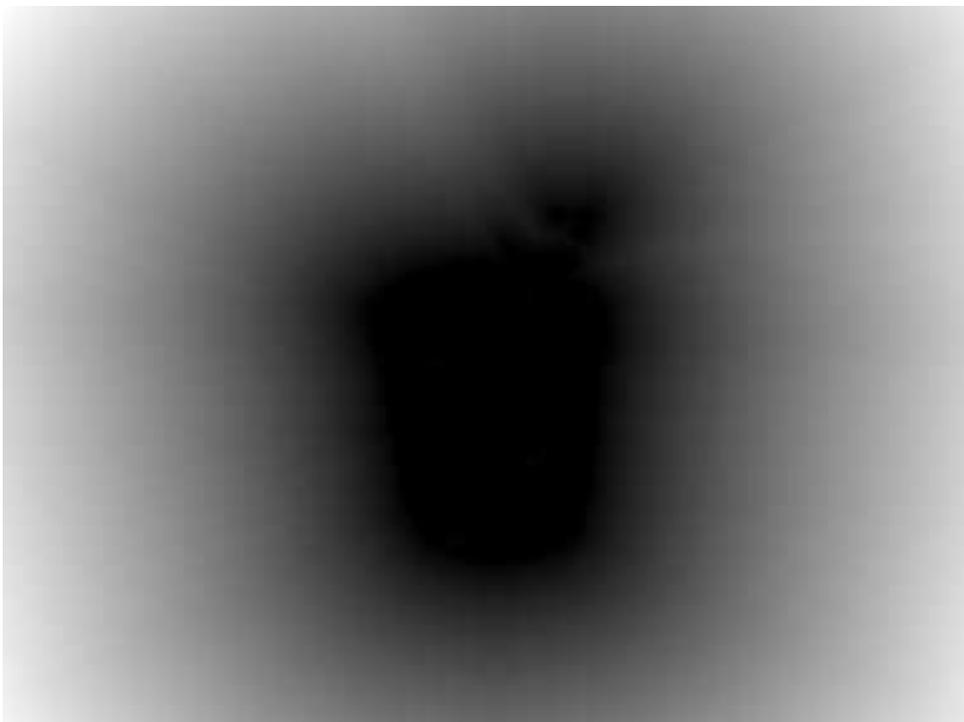


Abbildung 5.5: Szenario I – Distanztransformation der Farbklassifizierung

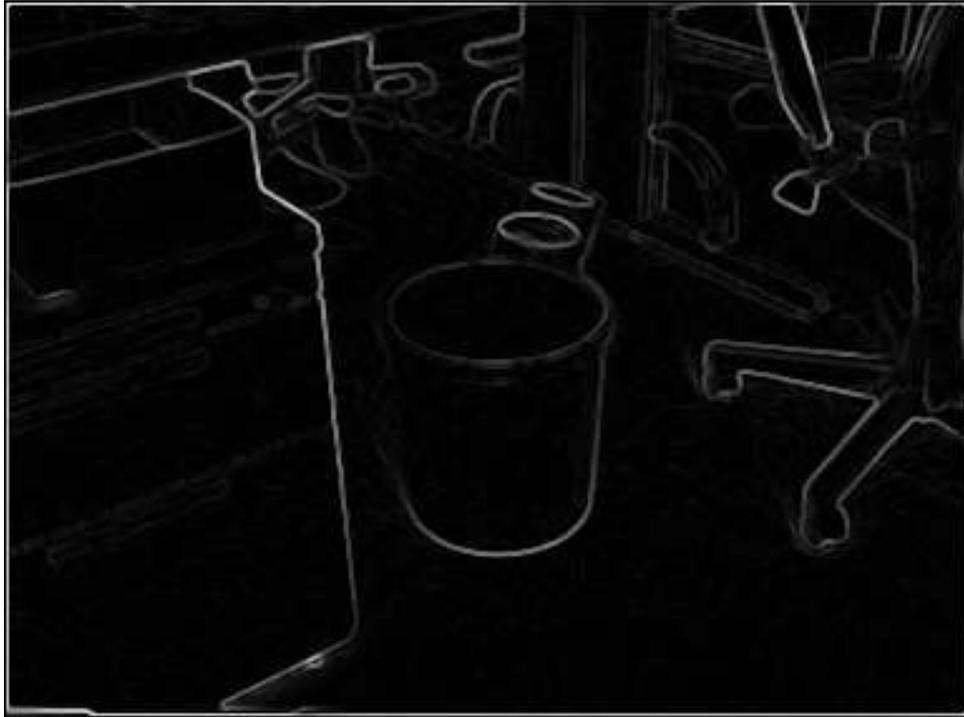


Abbildung 5.6: Szenario I – Kantenextraktion der Originalaufnahme



Abbildung 5.7: Szenario I – Binarisiertes Kantenbild

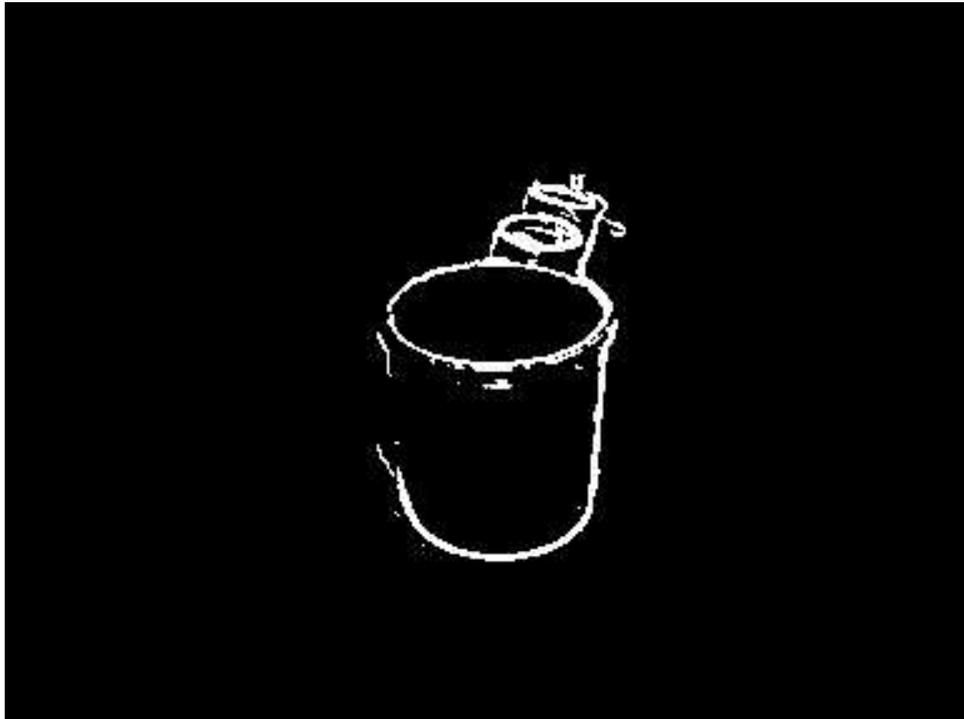


Abbildung 5.8: Szenario I – Verknüpfung von Farbklassifizierung und Kantenbild

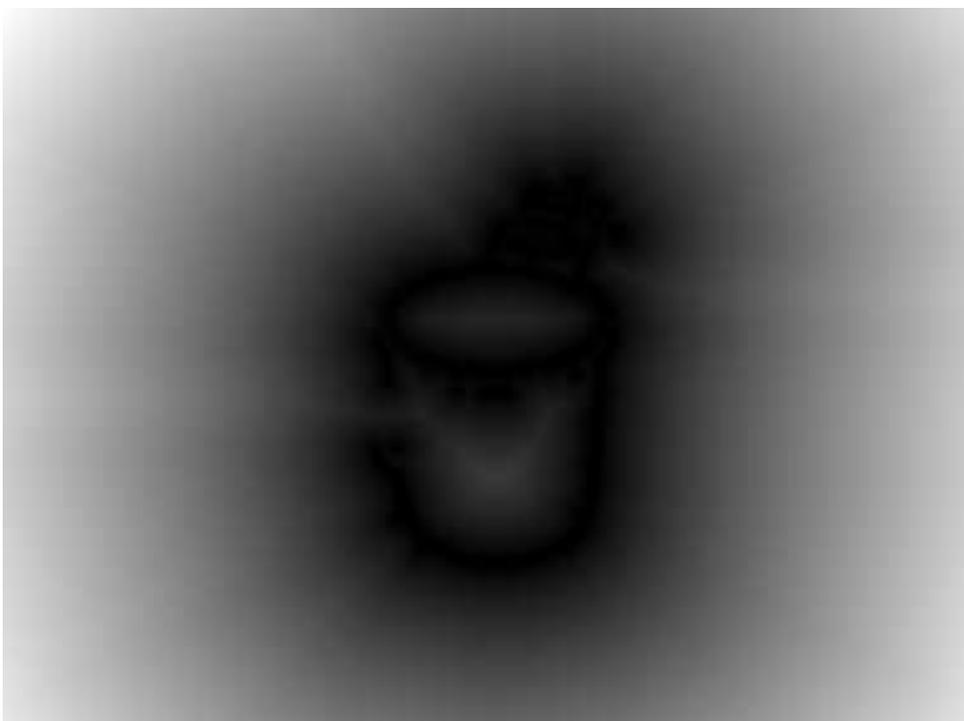


Abbildung 5.9: Szenario I – Distanztransformation des binarisierten Kantenbildes



Abbildung 5.10: Szenario I – Originalaufnahme mit eingeblendetem Suchergebnis

# Kapitel 6

## Beispielszenarios

Dieses Kapitel zeigt ergänzend zu dem in Kapitel 5 vorgestellten Szenario einige weitere und erläutert jeweils das Resultat der Suche.

### 6.1 Szenario II

Bild 6.1 zeigt das gesuchte Objekt zur Hälfte verdeckt hinter einem Schubkasten.

Die Farbklassifizierung (siehe Abb. 6.2) hat auf den Papierkorb gut angesprochen und das Objekt komplett erfaßt. Im linken oberen Teil des Bildes sind kleine Bereiche zu sehen, die von der Farbklassifizierung ebenfalls erfaßt wurden. Dieser sogenannte Fehler zweiter Art bei der Farbklassifizierung ist für den Prozeß unbedenklich. Befinden sich im Bereich dieser objektfremden Segmente Kanten, so werden diese bei der Objekterkennung ebenfalls ausgewertet, bleiben aber ohne Wirkung für die korrekte Bearbeitung des Bildes.

Die Kantenextraktion, in Abbildung 6.4 zu sehen, zeigt die Kanten des gesuchten Objekts fast vollständig. Lediglich ein kleiner Bereich im rechten oberen Teil des Objekts wurde nicht korrekt abgebildet. Im Originalbild ist zu sehen, daß in diesem Bereich in der Tat kein sichtbarer Übergang im Bild besteht. Für die Hausdorff-Distanz sind solche Fehler unproblematisch. Gerade die Fähigkeit, gegenüber einer fehlerhaften Merkmalsextraktion robust zu funktionieren, zeichnet sie aus. Kombinatorische Verfahren reagieren auf solche Fehler wesentlich empfindlicher. Die Binarisierung des Kantenbildes (siehe Abb. 6.5) hat die detektierten Kanten des Papierkorbes komplett übernommen.

Die Kombination von Farbklassifizierung und Kantenextraktion (Abb. 6.6) zeigt wie zu erwarten alle dem gesuchten Objekt zugehörigen Merkmale. Die Abbildung zusätzlicher Merkmale im Bild ist unbedenklich, lediglich die Verarbeitungsdauer kann dadurch etwas erhöht werden.

Das gesuchte Objekt wurde im Bild nicht erkannt. Die Voraussetzung, 80% der Modellmerkmale mit Bildmerkmalen in Korrelation zu bringen, konnte nicht erfüllt werden. Nachdem nur etwas mehr als die Hälfte des Papierkorbes im Bild zu sehen war, ist das negative Ergebnis der Suche als korrekt zu bewerten. Wichtig in dem Zusammenhang ist auch, daß an keiner anderen Stelle des Bildes die Existenz des gesuchten Objekts bestätigt wurde. Dies wäre als eine klare Fehlentscheidung der Objekterkennung zu werten gewesen. Das Ergebnis ist befriedigend.



Abbildung 6.1: Szenario II – Originalaufnahme

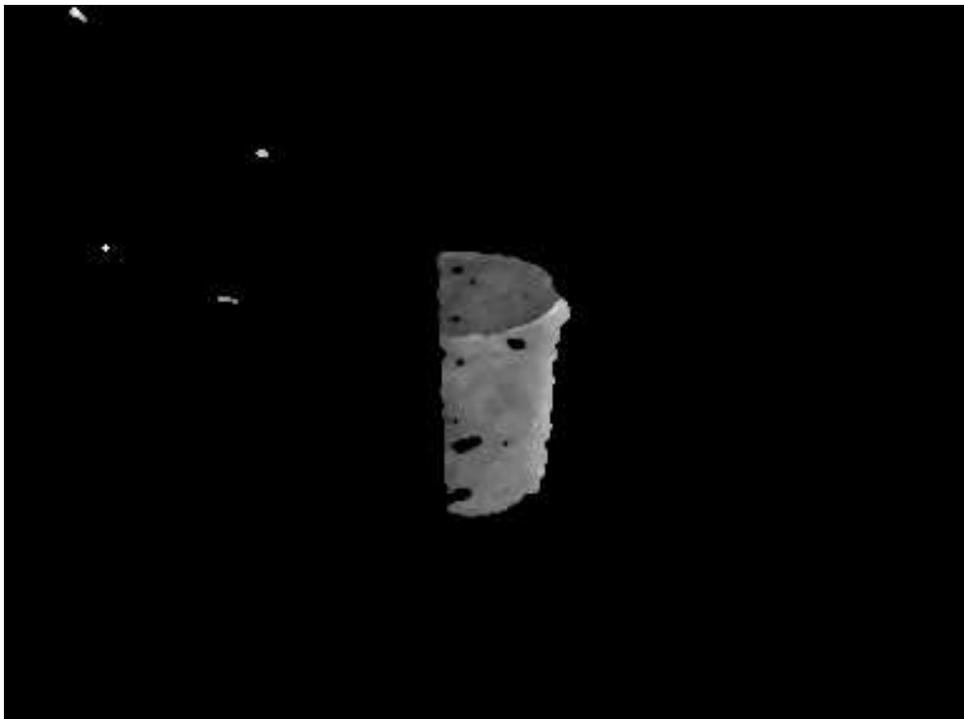


Abbildung 6.2: Szenario II – Ergebnis der Farbklassifizierung

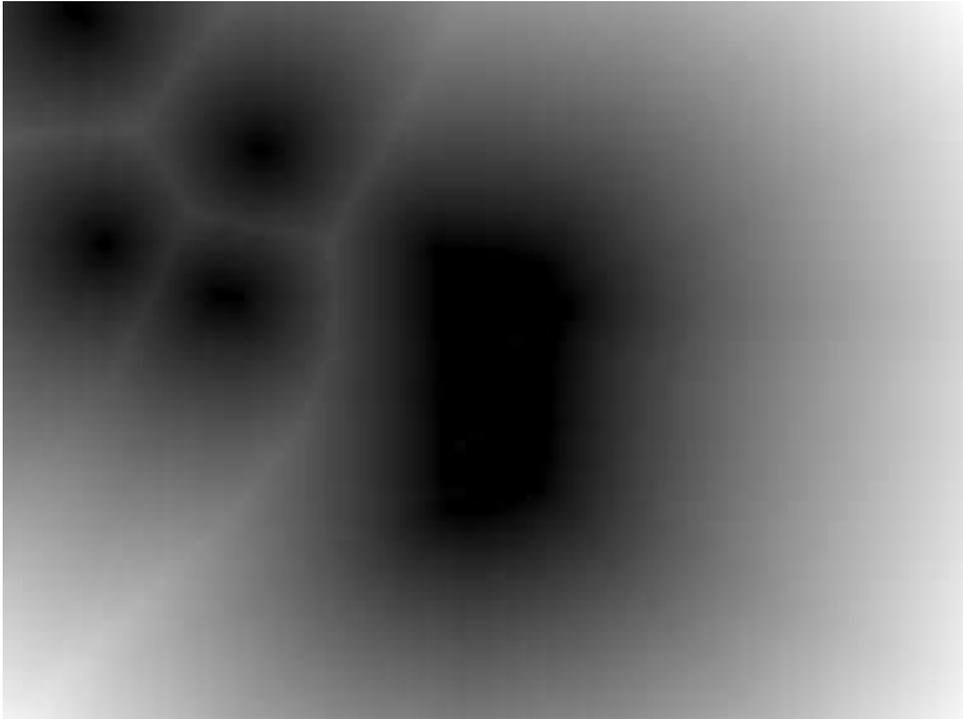


Abbildung 6.3: Szenario II – Distanztransformation der Farbklassifizierung

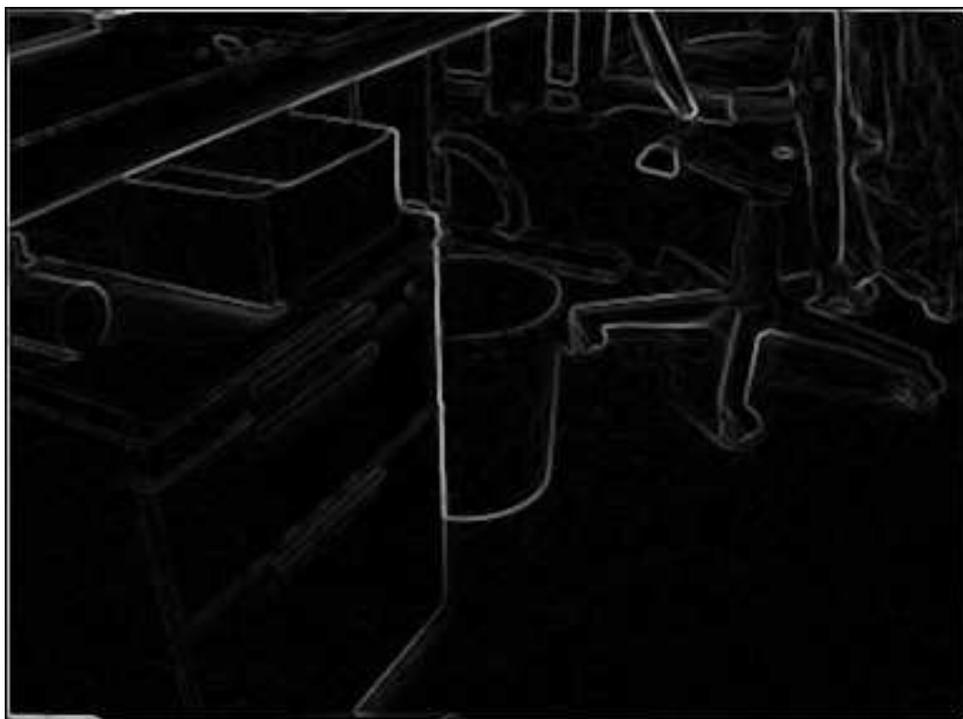


Abbildung 6.4: Szenario II – Kantenextraktion der Originalaufnahme

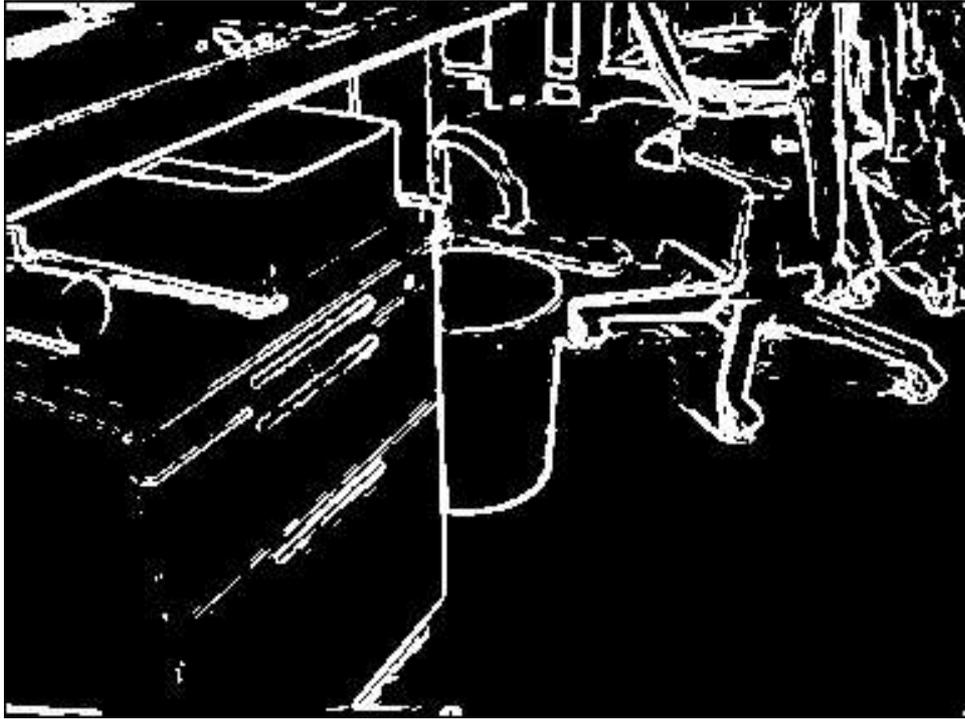


Abbildung 6.5: Szenario II – Binarisiertes Kantenbild

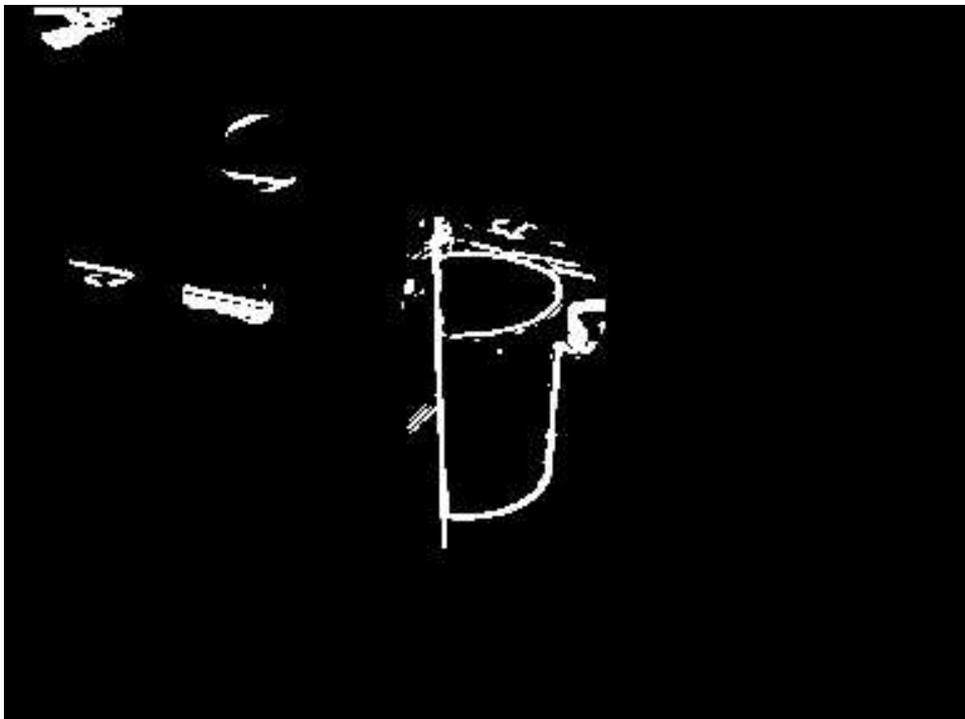


Abbildung 6.6: Szenario II – Verknüpfung von Farbklassifizierung und Kantenbild



Abbildung 6.7: Szenario II – Distanztransformation des binarisierten Kantenbildes

## 6.2 Szenario III

Bild 6.8 zeigt die Originalaufnahme mit dem gesuchten Objekt unter dem Schreibtisch stehend. Eine über den Stuhl gehängte Jacke verdeckt dabei einen Teil des Papierkorbes.

Die Farbklassifizierung (Abb. 6.9) hat das gesuchte Objekt komplett erfaßt. Zusätzlich wurden jedoch ebenfalls für auf dem Fensterbrett aufgestellte Blumentöpfe und einen neben dem Papierkorb stehenden Ordner Farbsegmente erzeugt. Dies ist nicht optimal, jedoch für den Objekterkennungsprozeß unbedenklich. Auf eine Größenfilterung oder die Auswertung der Form der vorhandenen Segmente wurde verzichtet, um zu vermeiden, daß kleine, abgespaltete Segmente, die dennoch zu dem gesuchten Objekt gehören, fälschlicherweise eliminiert werden.

Nicht alle Kanten des gesuchten Objekts wurden bei der Erzeugung des Kantenbildes (siehe Abb. 6.11) korrekt abgebildet. Die rechte Seitenlinie des Papierkorbes fehlt komplett, ein Teil der Öffnung wurde nur schwach dargestellt. Bei der Bildglättung wurden der Mülleimer und der dahinter befindliche Fußboden als ein konstanter Bereich angesehen, der eine Störung enthält. Diese wurde herausgeglättet und bei der Kantenextraktion konnte auf der rechten Seite keine Kante mehr detektiert werden. Die Binarisierung des Kantenbildes zeigt Abbildung 6.12. Sämtliche Kanten des gesuchten Objekts wurden in das Binärbild übernommen. Auch die nur schwach abgebildete Oberkante des Papierkorbes ist zum Großteil enthalten.

Die Verknüpfung von Farbklassifizierung und Kantenbild in Abbildung 6.12 zeigt sich wie erwartet. Durch die nicht optimale Farbklassifizierung bleiben verhältnismäßig viele Kanten erhalten. Es ist zu erwarten, daß die Bearbeitungsdauer dadurch etwas ansteigt.

Abbildung 6.15 zeigt die Originalaufnahme, in die das Modell mit der ermittelten Transformation eingeblendet wurde. Trotz teilweise fehlender Kanten wurde das Objekt korrekt erkannt. Der obere Rand des Papierkorbes wurde präzise lokalisiert, die Seiten des Papierkorbes sind trotz ungenügender Kanteninformation gut plaziert. Selbst die linke Kante des rechts vom Papierkorb stehenden Ordners führte zu keiner falschen Positionierung des Modells. Lediglich der untere Rand des Papierkorbes liegt leicht versetzt von den Zügen des Modells. Grund dafür können die starren Dimensionen des Modells oder ein vorzeitiger Abbruch der Suche aufgrund eines genügend guten Ergebnisses sein. Das Ergebnis ist befriedigend.



Abbildung 6.8: Szenario III – Originalaufnahme



Abbildung 6.9: Szenario III – Ergebnis der Farbklassifizierung

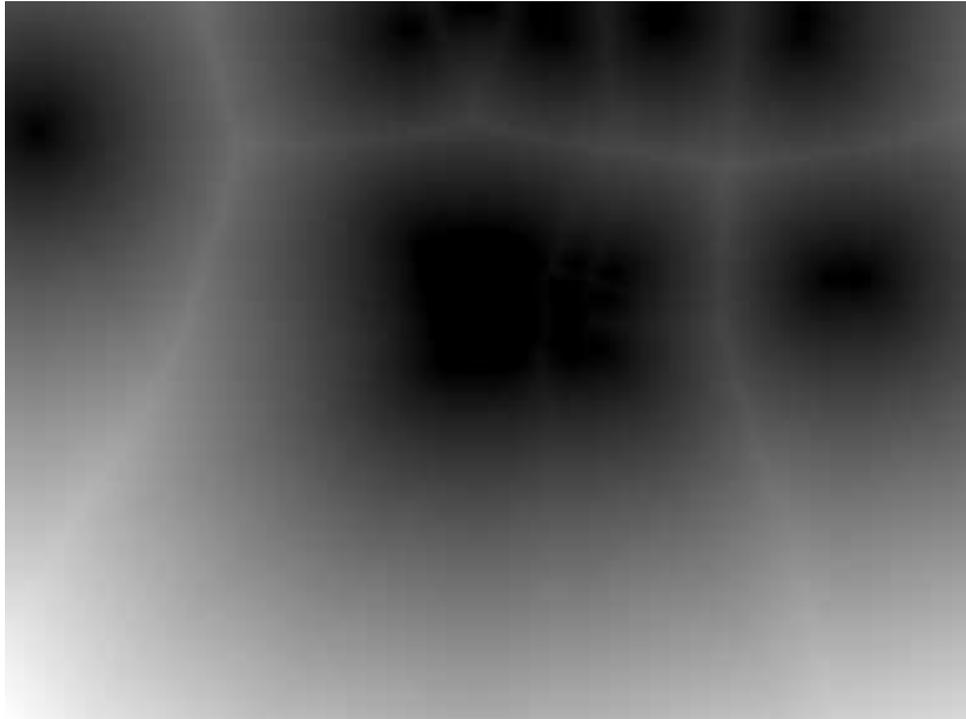


Abbildung 6.10: Szenario III – Distanztransformation der Farbklassifizierung



Abbildung 6.11: Szenario III – Kantenextraktion der Originalaufnahme



Abbildung 6.12: Szenario III – Binarisiertes Kantenbild

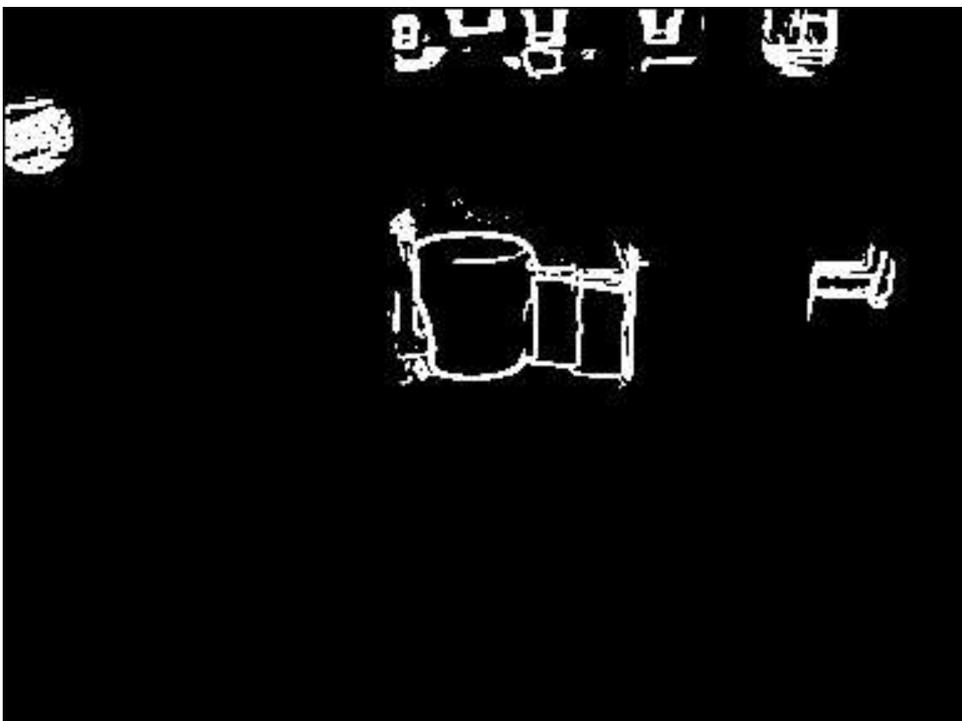


Abbildung 6.13: Szenario III – Verknüpfung von Farbklassifizierung und Kantenbild



Abbildung 6.14: Szenario III – Distanztransformation des binarisierten Kantenbildes



Abbildung 6.15: Szenario III – Originalaufnahme mit eingeblendetem Suchergebnis

## 6.3 Szenario IV

Die Originalaufnahme für Szenario IV zeigt Abbildung 6.16. Das gesuchte Objekt steht frei ohne jede Verdeckung unter einem Schreibtisch. Die rechte Seite des Objekts ist dabei stark schattiert, allgemein hebt sich das gesuchte Objekt nur schwach vom Hintergrund und dem Rest der Szene ab.

Die Farbklassifizierung, in Abbildung 6.17 dargestellt, weist einen großen Fehler erster Art auf – ein für die Zuverlässigkeit des gesamten Prozesses wenig förderliches Ergebnis. Um die Folgen eines solchen Fehlers zu lindern, wird generell zur Reduzierung des binarisierten Kantenbildes nicht das Ausmaß des Farbsegmentes selbst, sondern zusätzlich ein Bereich um das Farbsegment herum als interessanter Bereich deklariert. Zur Definition dieses umgebenden Bereiches wird die Distanztransformierte der Farbklassifizierung verwendet, sie ist in Abbildung 6.18 zu sehen.

Durch den schwachen Kontrast des gesuchten Objekts zur restlichen Szene sind bei der Kantenextraktion große Teile der Kanten nicht oder nur schwach abgebildet. Dies äußert sich auch bei der Binarisierung des Kantenbildes (siehe Abb. 6.20). Große Teile der zur Erkennung benötigten Bildmerkmale sind nicht vorhanden. Die Erkennung ist gefährdet.

Bei der Kombination von Farbklassifizierung und Kantenbild (Abb. 6.21) sind alle im Binärbild der Kantenextraktion vorkommenden Merkmale des Papierkorbes enthalten. Der Fehler erster Art bei der Farbklassifizierung wurde durch die Verwendung ihrer Distanztransformation voll kompensiert und hat sich nicht ausgewirkt. Die fehlenden Merkmale im Kantenbild hingegen sind jedoch weiterhin problematisch.

Wie bereits befürchtet, wurde das gesuchte und im Bild existente Objekt nicht gefunden. Schuld daran waren die schlechten Lichtverhältnisse im Schatten der Schreibtischoberfläche. Eine zusätzliche Beleuchtung des zu durchsuchenden Bereiches während der Aufnahme oder eine partielle Aufhellung des Originalbildes wären Ansätze zur Lösung dieses Problems. Das Ergebnis der Suche ist nicht befriedigend.



Abbildung 6.16: Szenario IV – Originalaufnahme



Abbildung 6.17: Szenario IV – Ergebnis der Farbklassifizierung

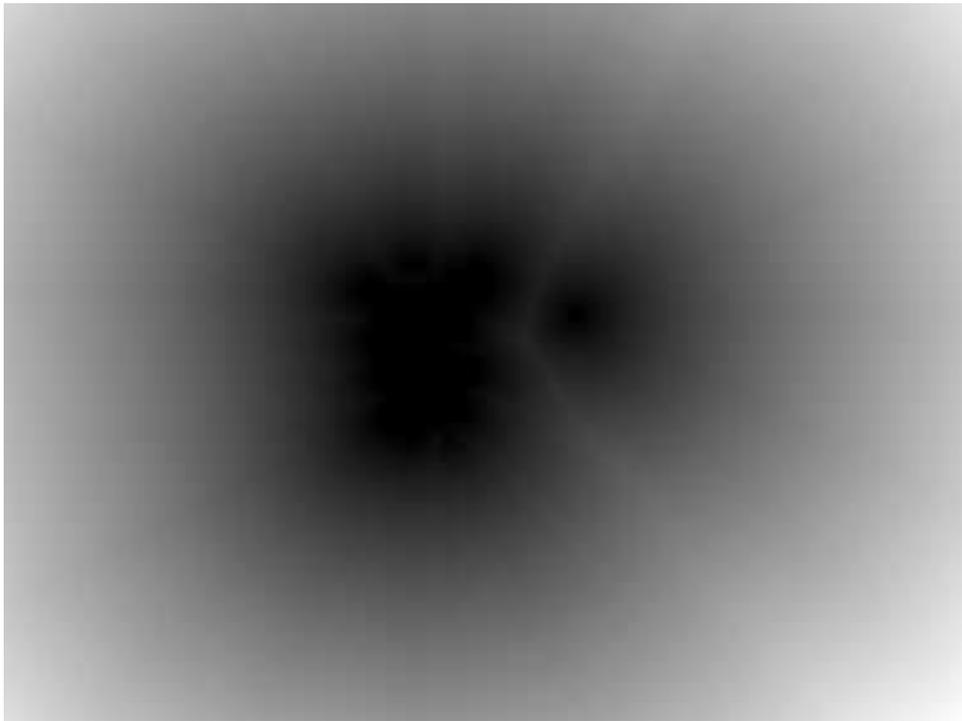


Abbildung 6.18: Szenario IV – Distanztransformation der Farbklassifizierung

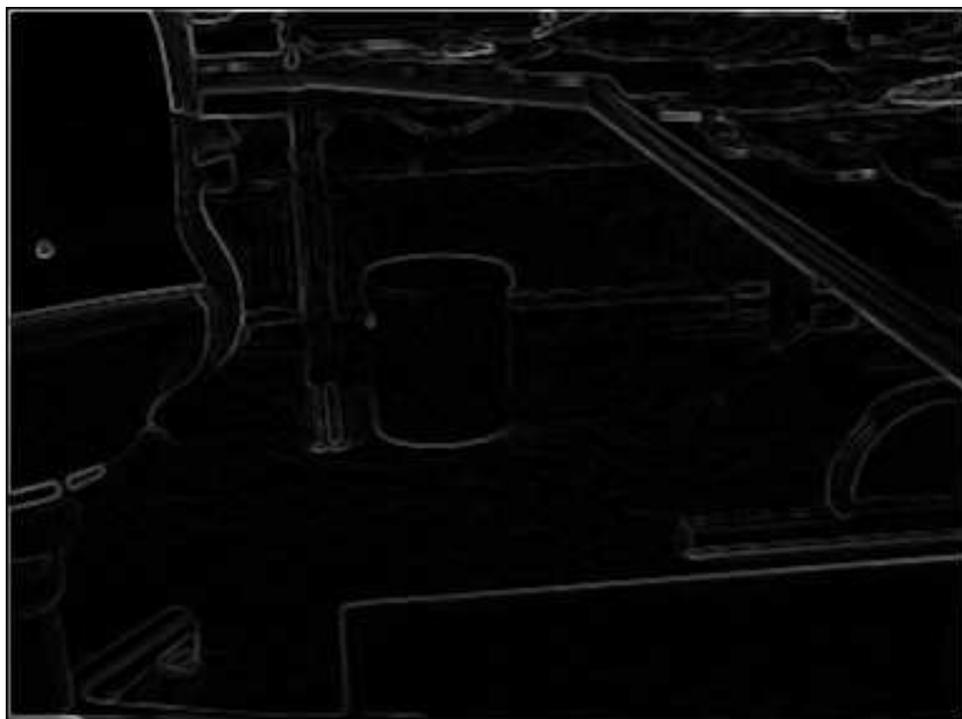


Abbildung 6.19: Szenario IV – Kantenextraktion der Originalaufnahme

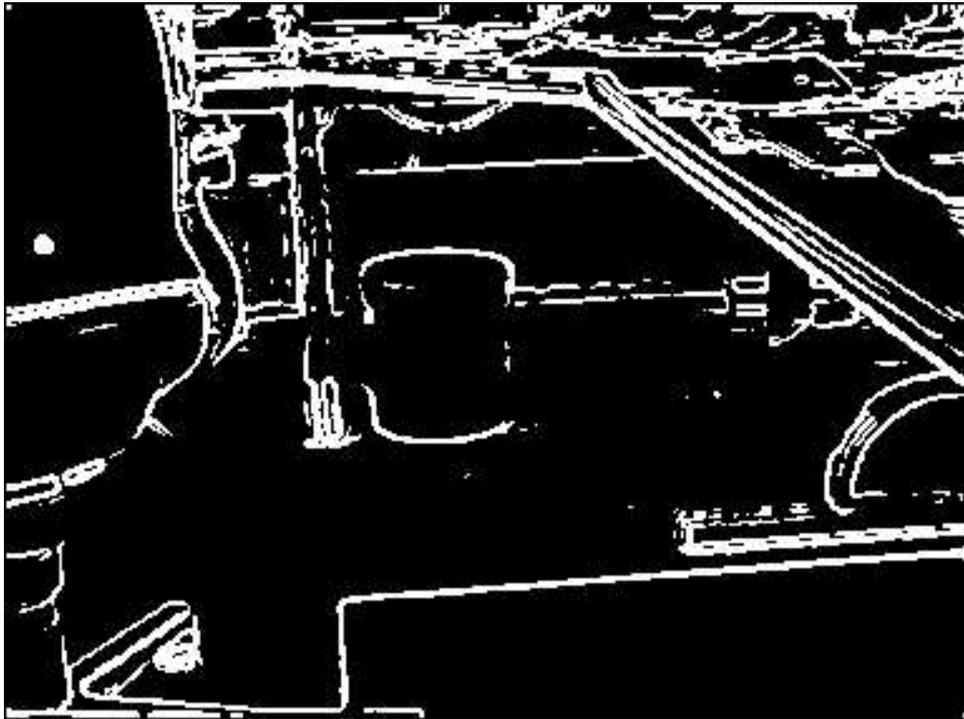


Abbildung 6.20: Szenario IV – Binarisiertes Kantenbild

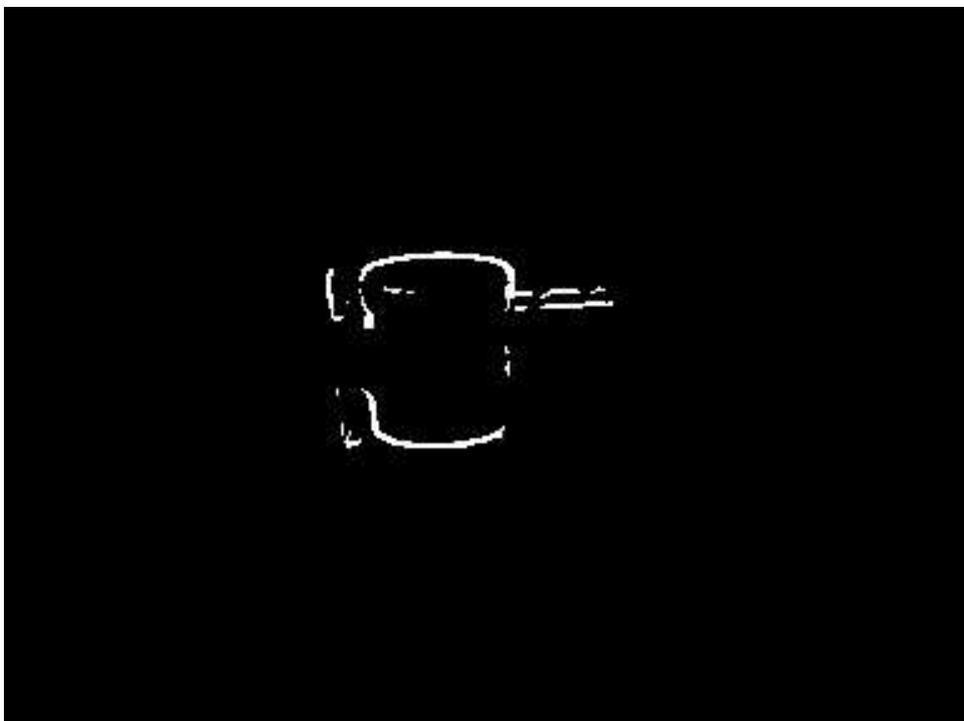


Abbildung 6.21: Szenario IV – Verknüpfung von Farbklassifizierung und Kantenbild

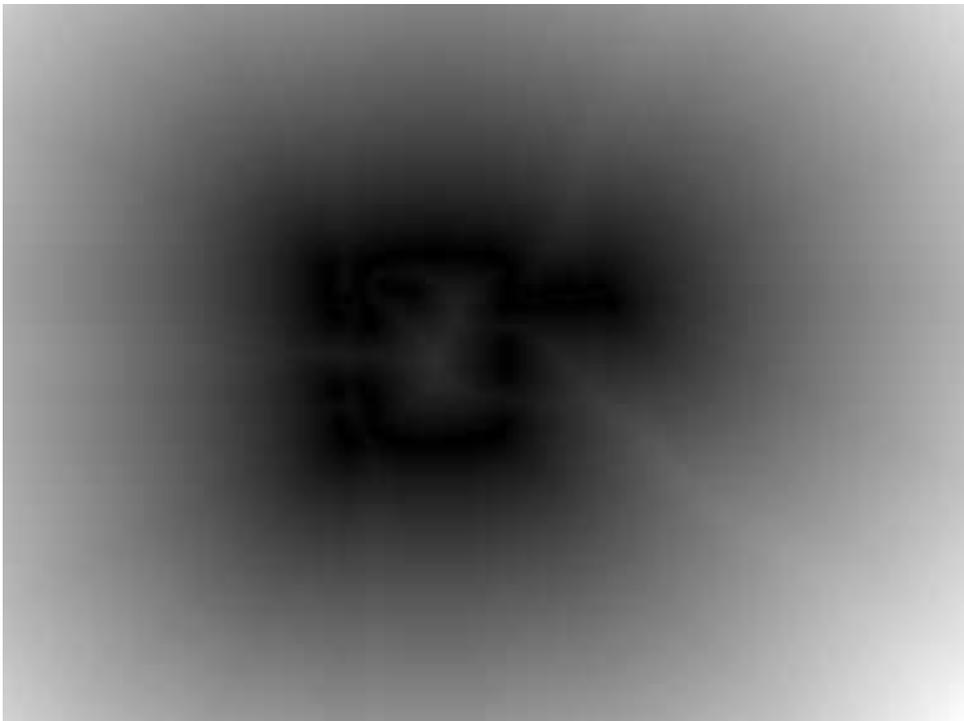


Abbildung 6.22: Szenario IV – Distanztransformation des binarisierten Kantenbildes

## 6.4 Szenario V

Abbildung 6.23 zeigt eine Szene, in der kein roter Papierkorb enthalten ist. Stattdessen ist ein formgleicher weißer Papierkorb unter dem Schreibtisch zu sehen. Dieser Papierkorb erfüllt jedoch nicht die geforderte Spezifikation und gehört somit auch nicht zur Klasse der gesuchten Objekte. Korrektes Ergebnis der Objekterkennung wäre also die Bestätigung, daß sich in der untersuchten Szene kein gültiges Objekt befindet.

Wie erwartet schlägt die Farbklassifizierung (siehe Abb. 6.24) nicht auf den existierenden Papierkorb an. Blumentöpfe auf dem Fensterbrett, die am oberen Bildrand sichtbar sind und zwei Aktenordner unter dem Schreibtisch erfüllen das klassifizierte Farbschema. Dieser Fehler zweiter Art ist für die Objekterkennung unkritisch, solange nicht fälschlicherweise der weiße Papierkorb ebenfalls als gültiger Bereich erkannt wird. Da der weiße Papierkorb zu dem gesuchten roten formgleich ist, würde die Formüberprüfung positiv ausfallen. Zusammen mit der fehlerhaften Farbklassifizierung wäre die Bestätigung der Existenz eines gültigen Objekts im Bild die Folge.

Die Kantenextraktion (Abb. 6.26) hat alle relevanten Kanten aus dem Originalbild sauber übernommen. Auch die Binarisierung des Kantenbildes, die in Abbildung 6.27 darstellt ist, zeigt keine Schwächen. Ein Formvergleich zwischen den Referenzmodellen und diesem Kantenbild würde ohne Zweifel zu einem positiven Ergebnis führen.

Durch die Kombination aus Farbklassifizierung und Kantenbild (siehe Abb. 6.28) werden jedoch große Teile des ursprünglichen Bildes ausgeblendet. Auch die Kanten des weißen Papierkorbes sind weitestgehend eliminiert. Eine Erkennung des Papierkorbes nach diesem Schritt ist unwahrscheinlich.

In der Tat wird die Suche beendet, ohne die Existenz eines interessanten Objekts anzuzeigen. Die Entscheidung der Objekterkennung ist korrekt, das Ergebnis zufriedenstellend.



Abbildung 6.23: Szenario V – Originalaufnahme



Abbildung 6.24: Szenario V – Ergebnis der Farbklassifizierung

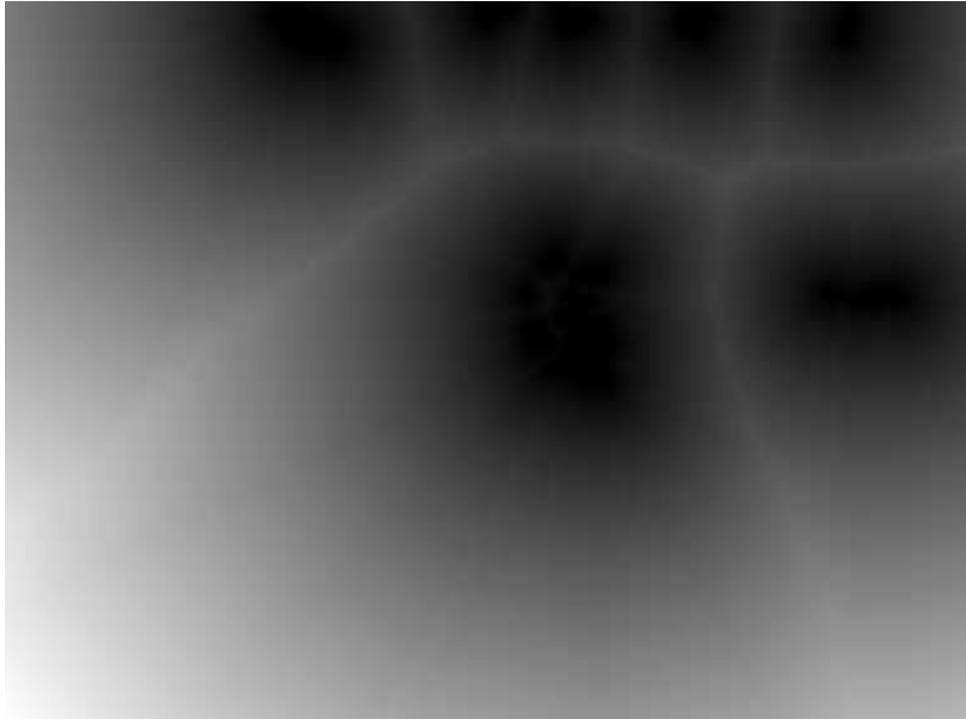


Abbildung 6.25: Szenario V – Distanztransformation der Farbklassifizierung



Abbildung 6.26: Szenario V – Kantenextraktion der Originalaufnahme

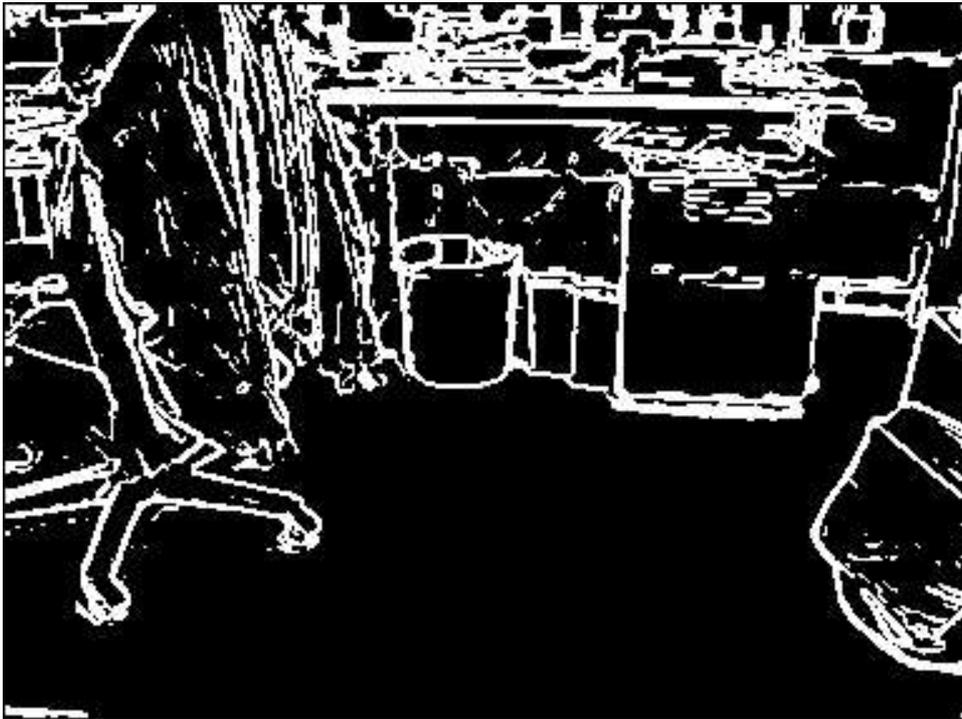


Abbildung 6.27: Szenario V – Binarisiertes Kantenbild

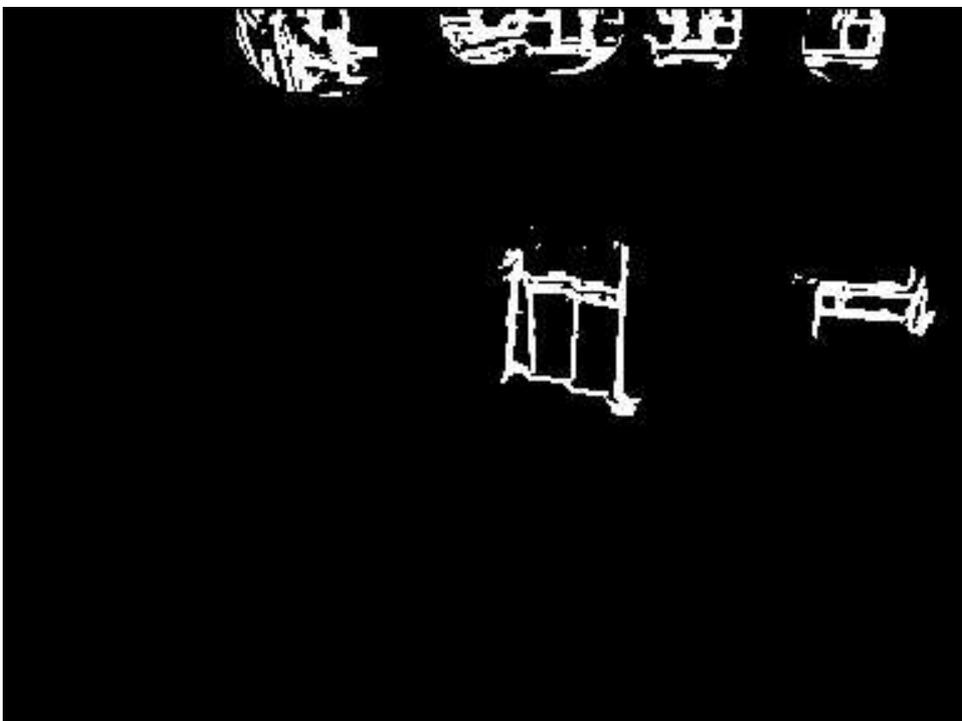


Abbildung 6.28: Szenario V – Verknüpfung von Farbklassifizierung und Kantenbild

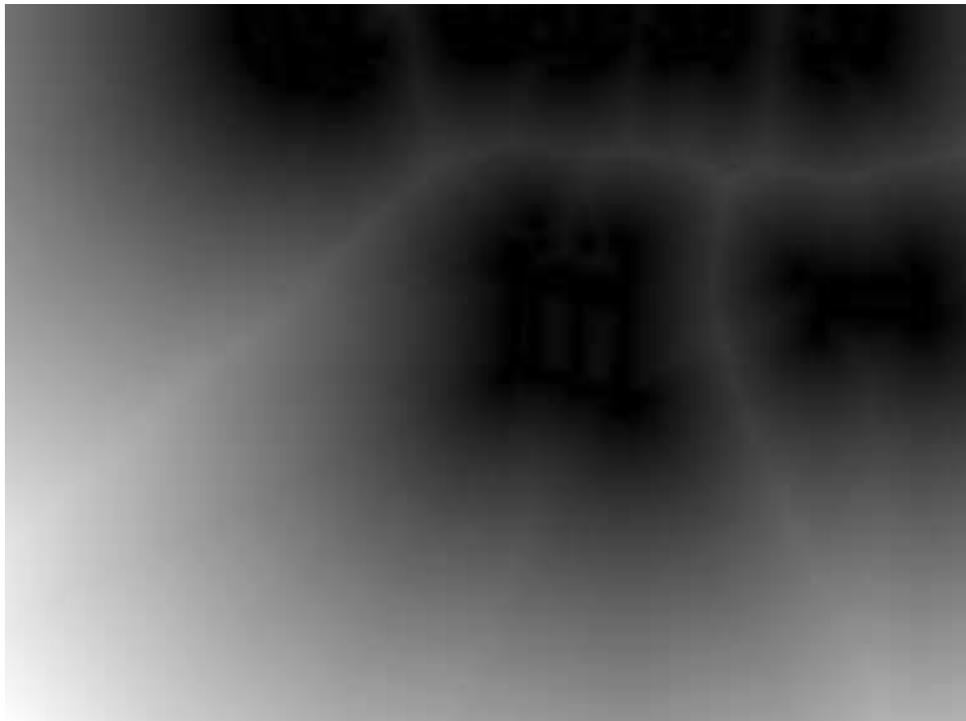


Abbildung 6.29: Szenario V – Distanztransformation des binarisierten Kantenbildes

## Kapitel 7

# Entwurf und Implementierung

Dieses Kapitel beschreibt die Voraussetzungen, die Anforderungen und den formalen Aufbau der Objekterkennung. Die wesentlichen Datenstrukturen und der Datenfluß des Prozesses sind beschrieben. Die bei der Implementierung gewählte Aufteilung der Gesamtfunktionalität in einzelne Module ist dokumentiert. Abschließend ist die Grundstruktur der zentralen Objekterkennungsroutine in Pseudocode dargestellt.

### 7.1 Entwicklungsplattform

Als Entwicklungsplattform stand eine Silicon Graphics Indigo<sup>2</sup> UNIX-Workstation und ein ANSI C-Compiler zur Verfügung. Produktiv wird die Objekterkennung nach Abschluß der Entwicklungsphase auf einem autonomen System von Nomadic Technologies Inc. eingesetzt. NOMAD basiert auf einer PC-Architektur mit Intel Pentium Prozessor und einer PCI-Busarchitektur. Als Betriebssystem wird Linux verwendet.

### 7.2 Anforderungen

Die Aufnahme der zu analysierenden Szene erfolgt im Stand. Nach Abschluß der Objekterkennung und der Übermittlung der Ergebnisse an das autonome System fährt dieses mit der weiteren Bearbeitung der gestellten Aufgabe fort.

Da es sich bei dem autonomen System um einen Prototypen handelt, bei dem die Überprüfung der grundsätzlichen Tauglichkeit verschiedener theoretischer Verfahren sowie deren Anpassung für die Realisierung eines Gesamtsystems im Vordergrund steht, werden weder an das Betriebssystem noch an die implementierte Steuerung Echtzeitanforderungen gestellt. Daraus ergibt sich, daß die Einhaltung von Echtzeitanforderungen auch für den Objekterkennungsprozeß nicht relevant ist. Bei der Entwicklung der Objekterkennung handelt es sich um eine Studie für das Gesamtkonzept.

### 7.3 Datenbeschreibung

Die Bildinformation und die Modellinformation stellen bei der Objekterkennung die zentralen Datenelemente dar. Im folgenden sind die verwendeten Strukturen für die Datenhaltung beschrieben, die in der Headerdatei `typedef.h` definiert sind.

### 7.3.1 Struktur der Bilddaten

Die Definition der Datenstruktur für die Speicherung der Bilddaten sieht in C-Notation folgendermaßen aus:

```
typedef struct
{
    int x, y;
    int bands;
    int *depth;
    int ***data;
} image_t;
```

Dabei stellen `x` und `y` die horizontale bzw. vertikale Größe des Bildes dar. Das Strukturelement `bands` gibt Auskunft über die Anzahl der Kanäle des Bildes. Für RGB-Bilder ist dieser Wert 3, für Monochrombilder 1. Der eindimensionale Array `depth` der Länge `bands` gibt für jeden Bildkanal die maximal zulässige Helligkeitsstufe an. Bei einem *true color* RGB-Farbbild ist `depth` demzufolge `{255, 255, 255}`, bei einem binären Schwarz-Weiß-Bild idealerweise `{1}`. Das Element `data` ist für die Speicherung der Bildinformation verantwortlich. Für ein  $1000 \times 1000$  Bildpunkte großes Bild mit drei Kanälen wäre `***data` als `data[3][1000][1000]` zu interpretieren.

### 7.3.2 Struktur der Modelldaten

Für die Darstellung der Modellinformation, die aus einer endlichen Menge von Punktkoordinaten besteht, wurde eine komplexe Datenstruktur aus mehreren Unterstrukturen entworfen. In C-Notation sind die Modelldaten wie folgt definiert:

```
typedef struct
{
    int x, y;
} point_t;

typedef struct _plist_t
{
    point_t p;
    struct _plist_t *next;
};

typedef struct
{
    plist_t *plist;
    int size;
    point_t max;
} model_t;
```

Die Struktur `point_t` ist eine einfache Hilfsstruktur zur Darstellung von Punktkoordinaten in  $N^2$ . `plist_t` definiert die Struktur für eine einfach verkettete Liste der einzelnen Punktkoordinaten. `model_t` schließlich definiert die Datenstruktur zur Speicherung der Modelldaten für ein einzelnes Modell. Erstes Element ist die Menge der Modellpunkte als `plist_t`, `size` gibt die Anzahl der Punkte in `plist` an und `max` liefert das Ausmaß der Streuung der Modellpunktswolke.

## 7.4 Datenablaufdiagramm

In Abbildung 7.1 ist der Verlauf der Bilddaten und der Modelldaten durch den gesamten Objekterkennungsprozeß dargestellt. Als Eingabe für den Objekterkennungsprozeß dienen das Originalbild sowie die gegebene Modellinformation. Ausgabe des Prozesses ist eine Symbolik für das Suchergebnis. Diese enthält die Transformationsparameter und einen Verweis auf das verwendete Modell.

## 7.5 Aspekte der Implementierung

Der gesamte Ablauf der Objekterkennung, einschließlich der Bildvorverarbeitung, ist im System in einem Prozeß implementiert. Da bei der Bildvorverarbeitung bis auf eine Ausnahme stets das Ergebnis der vorangegangenen Operation als Eingabe für den nächsten Schritt benötigt wird, ist eine Aufteilung auf mehrere Prozesse hier nicht sinnvoll – der nur quasiparallele Ablauf bringt keinen Geschwindigkeitsvorteil. Ebenso verhält es sich bei der eigentlichen Objekterkennung. Das Durchsuchen des gesamten Suchraumes könnte leicht auf mehrere Prozesse aufgeteilt werden. Auf die Effizienz (Ordnung) des Algorithmus hat dies jedoch keinen Einfluß. Eine nur quasiparallele Verarbeitung wirkt sich durch den zusätzlichen Overhead sogar nachteilig auf die Effizienz aus. Auf einer echten Multiprozessormaschine könnte der gesamte Suchraum gleichmäßig auf die verschiedenen Prozessoren aufgeteilt und eine tatsächlich parallel ablaufende Suche durchgeführt werden. Da bei der Suchraumaufteilung auf  $n$  Prozessoren zwar Overhead entsteht, der jedoch vernachlässigbar klein ist, könnte für die reine Objekterkennung eine Verkürzung der Bearbeitungsdauer auf fast  $\frac{1}{n}$  der ursprünglichen Zeit erreicht werden. Die Aufteilung auf mehrere Prozessoren ist bei der vorgegebenen Hardware jedoch nicht möglich.

### 7.5.1 Aufteilung der Funktionalität

Die Implementierung der benötigten Funktionalität wurde auf verschiedene Module aufgeteilt. Dabei wurden die Funktionen nach Funktionsgruppen zusammengefaßt.

Abbildung 7.2 zeigt die Modulstruktur der Implementierung.

#### Modul `recognizer`

Das Modul `recognizer` stellt die *mainline* des gesamten Objekterkennungsprozesses dar. Die Information aus der Modelldatenbank wird geladen und das Originalbild in den Hauptspeicher gelesen. Anschließend werden die einzelnen Bearbeitungsschritte der Bildvorverarbeitung mit eventuell benötigten Parametern aufgerufen. Das Ergebnis der Bildvorverarbeitung, die Modellinformation und die Qualitätsanforderungen an das Suchergebnis werden der Objekterkennungsroutine übergeben. War die Suche erfolgreich, so wird das Ergebnis dargestellt, andernfalls wird eine Meldung ausgegeben, daß das gesuchte Objekt im Bild nicht gefunden wurde.

Zsätzlich ist die Funktion `Evaluate()` zur Farbauswertung in diesem Modul definiert. Sie wird als Übergabeparameter der Farbklassifizierung übergeben. So sind die Eigenschaften der Farbsegmentierung extern definiert und nicht in die eigentliche Implementierung fest einprogrammiert.

#### Modul `image`

Im Modul `image` sind alle Funktionen und Hilfsfunktionen vereint, die zur Bildvorverarbeitung notwendig sind. Hier sind beispielsweise der Kantenoperator `Sobel()`,

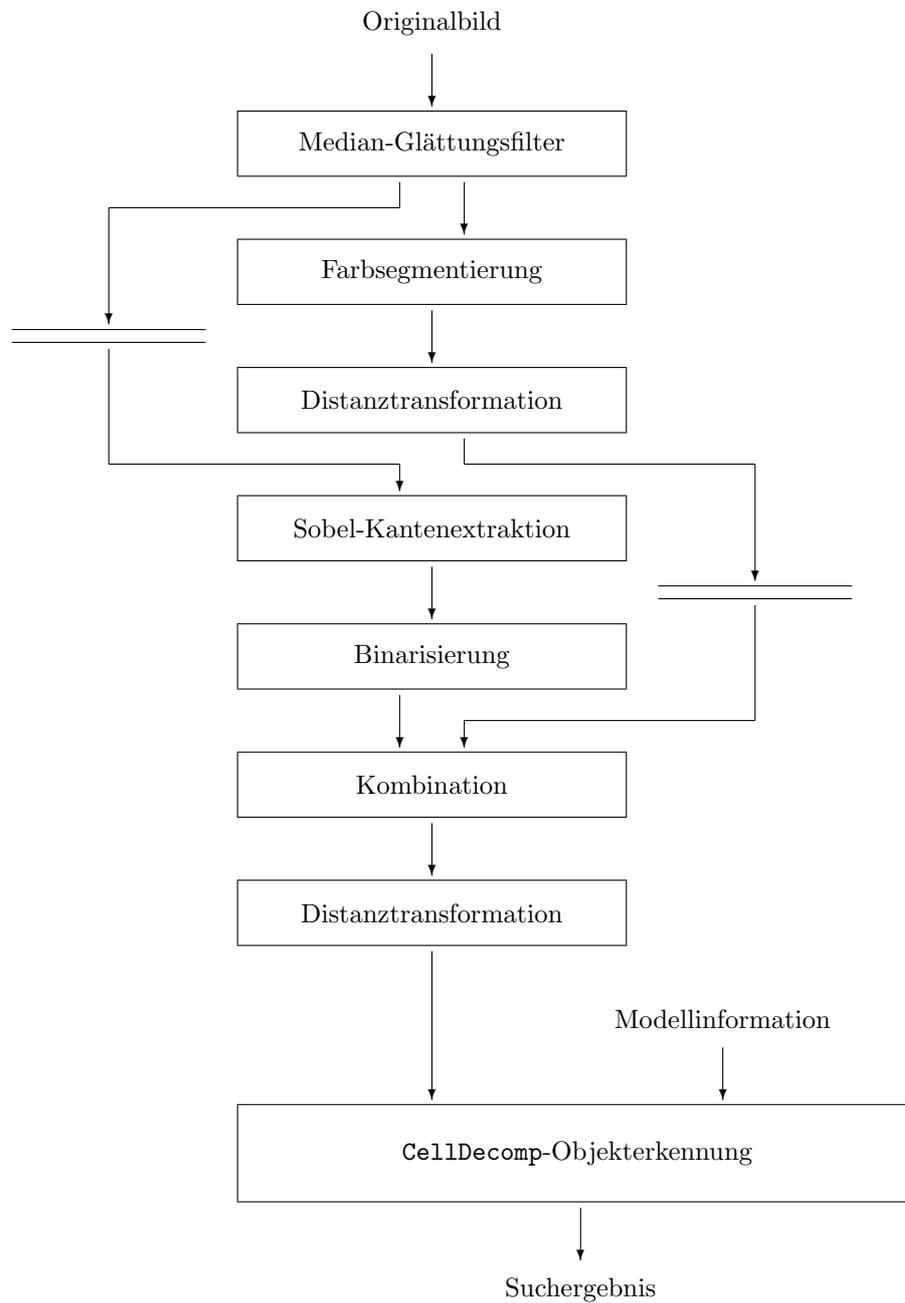


Abbildung 7.1: Datenverlauf der gesamten Objekterkennung

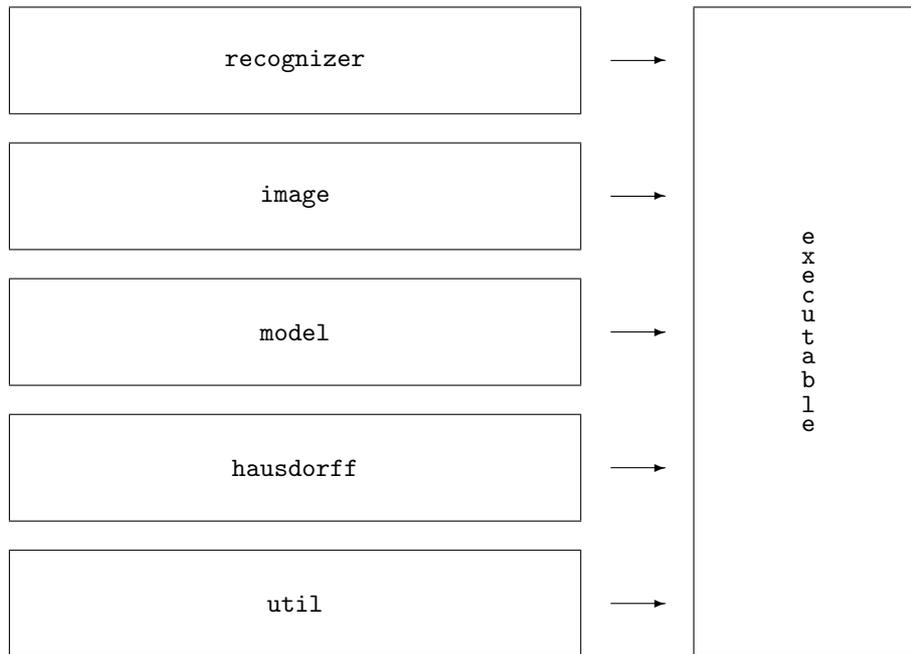


Abbildung 7.2: Modulstruktur des Objekterkennungsprozesses

die Farbklassifizierung `Segmentate()` und die Kombination `Connect()` implementiert. Unter Hilfsfunktionen verstehen sich Funktionen für das Laden, Speichern und Kopieren von Bilddaten.

#### Modul `model`

Das Modul `model` realisiert das Laden der Modelldaten aus der Modelldatenbank in den Hauptspeicher durch `LoadModelList()` und die Hilfsfunktionen zum Anlegen und Freigeben einer Modellliste im Hauptspeicher mit `AllocateModelList()` und `FreeModelList()`.

#### Modul `hausdorff`

Sämtliche Funktionen, die zur Berechnung der Hausdorff-Distanz und zur Administration des Verfahrens benötigt werden, sind im Modul `hausdorff` enthalten.

Hauptfunktion des Moduls ist `CellDecomp()`. Diese realisiert den Kern der Objekterkennung nach der Methode der Suchraumaufteilung (siehe Abschnitt 4.8.4). Ihr werden die Bilddaten, die Modelldaten und die Qualitätsansprüche für das Suchergebnis übergeben. Als Resultat liefert sie ein 6-Tupel zusammen mit der Modellkennung. Ist das 6-Tupel gleich 0, so wurde keine gültige Transformation gefunden – die Modellkennung ist nicht definiert. Ist das 6-Tupel ungleich 0, so enthält es die Translations- und Skalierungsparameter jeweils für  $x$  und  $y$  und die Basis der Skalierung ebenfalls für  $x$  und  $y$ . Die Modellkennung benennt das Modell, für das die zurückgegebene Transformation gültig ist.

Weiterhin sind Funktionen wie `InitList()`, `CreateCell()`, `CheckCell()`, `SubDivideCell()` und `DeleteCell()` zur Zellistenverwaltung, `Translate()`, `Scale()` und `Transform()` zur Punkttransformation sowie `FwdHK()` und `RevHK()` zur Berechnung der partiellen bzw. partiellen beschränkten Hausdorff-Distanz enthalten.

**Modul util**

Im Modul `util` sind lediglich Hilfsfunktionen implementiert, die allgemein anwendbar sind. Konkret enthält dieses Modul die Vergleichsfunktionen `CompInt()` und `CompRevInt()`, die einen Integervergleich analog zur C-Libraryfunktion `strcmp()` realisieren. Sie werden bei der Anwendung der C-Libraryfunktion `qsort()` verwendet, die den Quicksort-Algorithmus effizient implementiert.

**7.5.2 Kodierung der zentralen Objekterkennungsroutine**

Die Kodierung der zentralen Objekterkennungsroutine `CellDecomp()` ist im folgenden als Pseudocode dargestellt.

```

CellDecomp ← Bildinformation, Modellinformation,  $f_h, \tau_h, f_v, \tau_v$ 
Lösung ←  $\emptyset$ 
# Liste der zu durchsuchenden Transformationsräume aufbauen
Liste ←  $\emptyset$ 
für (alle Modelle)
    Neu ← Hyperwürfel über den gesamten Transformationsraum des Modells
    Neu.Distanz ← Hypothesendistanz ...
        ... für zentrale Transformation des Hyperwürfels
    Liste ← Liste + Neu
# Transformationsraum durchsuchen
solange (Verbesserung möglich  $\wedge$  Liste  $\neq \emptyset$ )
    Auswahl ← Hyperwürfel mit geringster Distanz in Liste
    Liste ← Liste - Auswahl
    Subliste ← Aufteilung der Auswahl in gleichgroße Subwürfel
    für (alle Subwürfel in Subliste)
        Subwürfel.Distanz ← Hypothesendistanz ...
            ... für zentrale Transformation des Subwürfels
        wenn (Subwürfel.Distanz > Subwürfel.Diagonale/2 +  $\tau_h$ )
            Subliste ← Subliste - Subwürfel
    sonst
        wenn (Inhalt von Subwürfel nur eine Transformation)
            Hypothese ← Transformation im Subwürfel
            wenn (Verifikationsdistanz der Hypothese <  $\tau_v$ )
                Lösung ← Modellnummer und Transformation
                Subliste ← Subliste - Subwürfel
                 $\tau_h$  ← Subwürfel.Distanz - 1
    Liste ← Liste + Subliste
# Ergebnis zurückgeben
← Lösung

```

# Kapitel 8

## Positionsbestimmung

Dieses Kapitel vermittelt Grundlagen, die für das prinzipielle Verständnis der Positionsbestimmung notwendig sind. Ferner werden das Verfahren zur Positionsbestimmung und die dazu notwendigen Definitionen beschrieben.

### 8.1 Bildgewinnung

In der Bildverarbeitung, speziell bei der Objekterkennung, wird die zu analysierende Szene durch perspektivische Projektion auf eine Bildebene abgebildet. Häufig geschieht dies nach dem Prinzip der Lochkamera, wie es in Abbildung 8.1 schematisch dargestellt ist.

$x$ ,  $y$  und  $z$  sind die Raumkoordinaten der Szene im Kamerakoordinatensystem, d. h. aus der Sicht der Kamera.  $f$  ist die sogenannte Brennweite, also der Abstand der Bildebene zum Projektionszentrum  $P$ , der eine scharfe Abbildung erzeugt. Somit ergibt sich eine Abbildungsgleichung vom drei-dimensionalen Raum der Szene in die zwei-dimensionale Ebene der Projektion.

$$(x, y, z) \mapsto (x', y') = \left( \frac{fx}{z}, \frac{fy}{z} \right) \quad (8.1)$$

Bei der Zentralprojektion werden die Weltkoordinaten parallel zur Bildebene beim Übergang zu den Bildkoordinaten mit dem Skalierungsfaktor  $\frac{f}{z}$  multipliziert. Die Bildkoordinaten  $x'$  und  $y'$  enthalten daher keine absolute Information über die

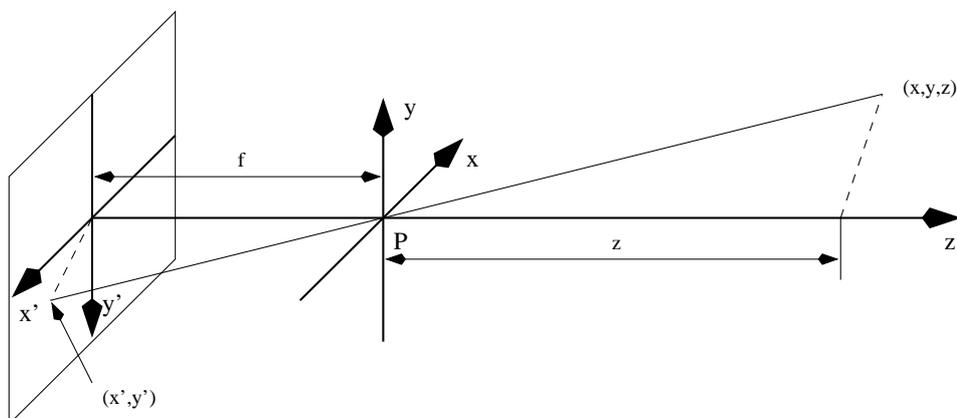


Abbildung 8.1: Das Modell der Lochkamera

Weltkoordinaten. Die Entfernung eines Objekts ist aus den Bildkoordinaten ebenso wenig bestimmbar wie dessen wahre Größe.

## 8.2 Rückgewinnung der Tiefeninformation

Der Punkt auf der Bildebene, auf den ein Punkt der Szene projiziert wird, ergibt sich aus dem Schnittpunkt von Bildebene und dem sogenannten Sichtstrahl. Der Sichtstrahl ist durch den abzubildenden Punkt  $(x, y, z)$  der Szene und das Projektionszentrum  $P$  im Raum eindeutig festgelegt.

Für den umgekehrten Vorgang kann der Sichtstrahl aus den Bildkoordinaten  $(x', y')$  und dem Projektionszentrum  $P$  berechnet werden, der Punkt  $(x, y, z)$  auf dem Sichtstrahl allerdings kann ohne zusätzliche Information nicht zurückgewonnen werden. Wird der Bereich, in dem sich relevante Punkte der Szene ausschließlich befinden dürfen, ausreichend eingeschränkt, so ist eine Berechnung der ursprünglichen Position möglich. Eine hinreichende Einschränkung ist die Definition einer Ebene  $E(a, b, c, d)$  mit bekannter Lage, wenn  $P \notin E$  ist.

Verwendet wurde hier der Fußboden, dessen relative Lage zur Kamera stets bekannt ist.

## 8.3 Homogene Koordinaten

Wurde die Position innerhalb des Kamerakoordinatensystems bestimmt und ist das Kamerakoordinatensystem und das Weltkoordinatensystem nicht identisch, so wird eine Koordinatentransformation notwendig. Im einfachen Fall geschieht dies durch eine Folge von Drehungen, um die Koordinatenachsen beider Systeme gegeneinander auszurichten, gefolgt von einer Verschiebung, die den Koordinatenursprung des Kamerasystems mit dem des Weltkoordinatensystems zusammenlegt.

### 8.3.1 Motivation

Eine Folge verschiedener Transformationen ist notwendig, um das Ursprungskoordinatensystem in das Zielkoordinatensystem abzubilden. Zwei Arten von Transformationen sind im folgenden beispielhaft kurz angeführt, um den Sachverhalt zu verdeutlichen. Beide werden aber später im Kapitel genauer behandelt.

Eine Translation (8.2) kann als eine Addition (8.3), eine Skalierung (8.4) als eine Multiplikation (8.5) dargestellt werden.

$$x' = x + d_x, \quad y' = y + d_y \quad \text{und} \quad z' = z + d_z \quad (8.2)$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} \quad \text{oder} \quad P' = P + T \quad (8.3)$$

$$x' = s_x \cdot x, \quad y' = s_y \cdot y \quad \text{und} \quad z' = s_z \cdot z \quad (8.4)$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \text{oder} \quad P' = S \cdot P \quad (8.5)$$

Es ist leicht zu erkennen, daß die Behandlung von Translation und Skalierung unterschiedlich ist. Wünschenswert wäre jedoch eine einheitliche Form, so daß durch eine einzige Rechenoperation, allein abhängig vom Operator, jede beliebige Transformation durchgeführt werden kann. Nach einer Erweiterung der drei-dimensionalen Koordinaten auf homogene Koordinaten werden alle Transformationen durch Multiplikation berechnet.

### 8.3.2 Einführung

Im homogenen Koordinatensystem wird jeder Punkt  $(x, y, z)$  durch ein 4-Tupel  $(x', y', z', W)$  dargestellt, in dem mindestens eine Koordinate ungleich 0 ist. Der Punkt  $(0, 0, 0, 0)$  ist nicht definiert. Zwei homogene Koordinaten  $(x, y, z, W)$  und  $(x', y', z', W')$  repräsentieren den gleichen Punkt, wenn sie linear abhängig sind. Punkte mit  $W = 0$  liegen im Unendlichen. Punkte mit  $W \neq 0$  werden in der Regel normiert, so daß  $W = 1$  ist.  $\frac{x}{W}$ ,  $\frac{y}{W}$  und  $\frac{z}{W}$  werden als die kartesischen Koordinaten eines homogenen Punktes bezeichnet.

### 8.3.3 Translation

Eine Translation ist die Verschiebung eines Punktes um  $d_x$  Einheiten entlang der  $x$ -Achse,  $d_y$  Einheiten entlang der  $y$ -Achse und  $d_z$  Einheiten entlang der  $z$ -Achse. Die Gleichung im homogenen Koordinatensystem ist

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (8.6)$$

### 8.3.4 Skalierung

Eine Skalierung ist die Streckung der Punktkoordinaten um den Streckungsfaktor relativ zum Koordinatenursprung. Die Verschiebung von  $P(x, y, z)$  in Richtung  $\vec{OP}$  ist umso größer, je weiter  $P$  vom Ursprung  $O(0, 0, 0)$  entfernt ist. Die Gleichung im homogenen Koordinatensystem ist

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (8.7)$$

### 8.3.5 Rotation

Eine Rotation ist die Drehung eines Punktes um eine der Achsen im Koordinatensystem. Ein positiver Drehwinkel  $\varphi$  bedeutet eine Drehung gegen den Uhrzeigersinn in einem rechten Koordinatensystem<sup>1</sup> und eine Drehung im Uhrzeigersinn in einem linken Koordinatensystem<sup>2</sup>.

#### $x$ -Achse als Rotationsachse

Eine Rotation um die  $x$ -Achse mit positivem Drehwinkel  $\varphi$  bedeutet eine Drehung von  $y$  nach  $z$ . Die Gleichung im homogenen Koordinatensystem ist

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi & 0 \\ 0 & \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (8.8)$$

<sup>1</sup>Werden positive  $x$  nach rechts angetragen und positive  $y$  nach oben, so liegen positive  $z$  auf der dem Leser zugewandten Seite des Koordinatensystems.

<sup>2</sup>Werden positive  $x$  nach rechts angetragen und positive  $y$  nach oben, so liegen positive  $z$  auf der dem Leser abgewandten Seite des Koordinatensystems.

### $y$ -Achse als Rotationsachse

Eine Rotation um die  $y$ -Achse mit positivem Drehwinkel  $\varphi$  bedeutet eine Drehung von  $z$  nach  $x$ . Die Gleichung im homogenen Koordinatensystem ist

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \varphi & 0 & \sin \varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \varphi & 0 & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (8.9)$$

### $z$ -Achse als Rotationsachse

Eine Rotation um die  $z$ -Achse mit positivem Drehwinkel  $\varphi$  bedeutet eine Drehung von  $x$  nach  $y$ . Die Gleichung im homogenen Koordinatensystem ist

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 & 0 \\ \sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (8.10)$$

## 8.3.6 Scherung

Die Scherung verändert zwei Koordinaten eines Punktes abhängig von der dritten. Dabei erhöht sich der Wert der Koordinate um ihren Betrag skaliert mit dem Wert der dritten Koordinate.

Die Scherung bedarf, ähnlich der Rotation, einer weiteren Unterteilung. Da die Scherung für diese Arbeit aber ohne Belang ist, wird beispielhaft die Scherung der  $x$ - und  $y$ -Koordinate dargestellt. Die Scherung für andere Koordinaten ist analog.

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (8.11)$$

## 8.3.7 Komposition von Transformationen

Die in Abschnitt 8.3 beschriebenen Transformationen lassen sich beliebig verknüpfen. Sind mehrere Schritte notwendig, um eine Koordinatentransformation durchzuführen, so werden die Umformungen der Reihe nach auf den Ausgangspunkt angewendet. Ein Beispiel soll dies verdeutlichen.

Ein beliebiges Objekt im Raum mit Mittelpunkt  $M(10, 5, 7)$  soll auf die Hälfte verkleinert werden. Außerdem ist es um die  $x$ -Achse mit einem Winkel von  $\varphi = 45^\circ$  und um die  $z$ -Achse mit einem Winkel von  $\vartheta = 90^\circ$  zu drehen. Folgende Schritte sind dazu im einzelnen nötig:

1. Translation des Objekts in den Koordinatenursprung, dem Referenzpunkt für eine Skalierung, mit  $T(-10, -5, -7)$
2. Skalierung des Objekts mit  $S(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$
3. Rotation des Objekts um die  $x$ -Achse mit  $\varphi = 45^\circ$  durch  $R_x(45^\circ)$
4. Rotation des Objekts um die  $z$ -Achse mit  $\vartheta = 90^\circ$  durch  $R_z(90^\circ)$
5. Translation des Objekts zurück an die ursprüngliche Position mit  $T(10, 5, 7)$

Die komplette Koordinatentransformation für das Objekt im Raum wird durch eine Gleichung erreicht. Die einzelnen Umformungen werden in umgekehrter Reihenfolge auf jeden Punkt des Objekts angewendet.

$$P' = \underbrace{T(10, 5, 7) \cdot R_z(90^\circ) \cdot R_x(45^\circ) \cdot S\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right) \cdot T(-10, -5, -7)}_{M_{TRRST}} \cdot P \quad (8.12)$$

$$P' = M_{TRRST} \cdot P \quad (8.13)$$

Da die Multiplikation von Matrizen assoziativ ist, können Teile der Transformation, aber auch die komplette Transformation, wie oben angedeutet, im voraus berechnet werden. Die Transformation eines zweiten Objektpunktes erfolgt durch die Multiplikation mit einer einzigen Matrix.

### 8.3.8 Änderung des Koordinatensystems

Die Transformation eines Punktes, wie sie bisher beschrieben wurde, ändert die Koordinaten des Punktes im Koordinatensystem. Eine andere Sichtweise ist, daß der Punkt im Raum unverändert bleibt, das Koordinatensystem aber transformiert wird. Dies ist besonders nützlich, wenn ein fixer Punkt im Raum gegeben ist, dessen Position in einer veränderten Sicht weiterverarbeitet wird. Beispiel dafür ist die Kinematik.

Ist  $P_i$  die Darstellung von  $P$  im Koordinatensystem  $i$  und  $M_{i \leftarrow j}$  die Transformation vom Koordinatensystem  $j$  in das Koordinatensystem  $i$ , so gilt:

$$P_2 = M_{2 \leftarrow 1} \cdot P_1 \quad (8.14)$$

Diese Transformation ist transitiv, daher gilt weiter, wenn

$$P_3 = M_{3 \leftarrow 2} \cdot P_2 \quad \text{und} \quad P_2 = M_{2 \leftarrow 1} \cdot P_1, \quad (8.15)$$

dann ist

$$P_3 = \underbrace{M_{3 \leftarrow 2} \cdot M_{2 \leftarrow 1}}_{M_{3 \leftarrow 1}} \cdot P_1 \quad (8.16)$$

$$P_3 = M_{3 \leftarrow 1} \cdot P_1. \quad (8.17)$$

Eine einfache, aber nützliche Koordinatensystemtransformation ist die Umformung der Koordinaten von einem rechten Koordinatensystem in ein linkes und umgekehrt mit

$$M_{R \leftarrow L} = M_{L \leftarrow R} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (8.18)$$

Konkret wird in dieser Arbeit mit Hilfe der Koordinatentransformation das Kamerakoordinatensystem in das Roboterkoordinatensystem bzw. das Weltkoordinatensystem transformiert. Die Position eines mit der Kamera georteten Objekts kann so leicht in eine für das Robotersystem relevante Darstellung umgeformt werden.

## 8.4 Kamerakalibrierung

Die Kamerakalibrierung ist ein wichtiges Gebiet für die Vermessung anhand von Videobildern, auch Photogrammetrie oder Videometrie genannt. Es ist ein ebenso komplexes wie umfangreiches Gebiet. Für diese Arbeit waren die Ergebnisse der Kamerakalibrierung sehr wertvoll, wurden aber nicht selbst hergeleitet. Daher wird auf diese Thematik hier nicht näher eingegangen. Die Grundlagen der Kamerakalibrierung werden in [JKS95] Kapitel 12 erläutert, eine detaillierte Abhandlung über diese Thematik findet sich in [Tsa87]. Die Ergebnisse und ihre Relevanz für die Positionsbestimmung werden hier jedoch kurz dargestellt.

### Eliminierung der Verzerrung erster Ordnung

Die lineare Verzerrung in einem Bild mit Mittelpunkt  $M(x_0, y_0)$  und einem Verzerrungsfaktor  $\kappa$  wird für  $x' = f(x, y)$  und  $y' = g(x, y)$  wie folgt eliminiert:

$$f(x, y) = (x - x_0) \left( 1 + \kappa \left( (x - x_0)^2 + (y - y_0)^2 \right) \right) + x_0 \quad (8.19)$$

$$g(x, y) = (y - y_0) \left( 1 + \kappa \left( (x - x_0)^2 + (y - y_0)^2 \right) \right) + y_0 \quad (8.20)$$

Ein entzerrtes Bild ist notwendig, um die Richtung des Sichtstrahls korrekt zu ermitteln. Ungenauigkeiten durch die Linsenverzerrung hätten eine Verfälschung der Positionsschätzung zur Folge.

Konkret war  $M = (184.0, 140.0)$  und  $\kappa = -1.374 \cdot 10^{-6}$ .

### Winkelauflösung in $x$ - und $y$ -Richtung

Die Winkelauflösung im Kamerabild wird zur Bestimmung des Richtungsvektors für die Sichtlinie benötigt, die zu einem Bildpunkt gehört (vgl. Abschnitt 8.1 und 8.2, sowie Abb. 8.1). Der Sichtstrahl für den Bildmittelpunkt hat in  $x$ - und  $y$ -Richtung die Auslenkung von jeweils  $0^\circ$ . Der Sichtstrahl für direkt benachbarte Bildpunkte erfährt eine Auslenkung in  $x$ -Richtung um  $\pm\varphi_w$  bzw. in  $y$ -Richtung um  $\pm\vartheta_w$  bei einer Weitwinkelaufnahme und analog um  $\pm\varphi_t$  bzw.  $\pm\vartheta_t$  bei einer Teleaufnahme. Die Auslenkung ist ein Vielfaches der Winkelauflösung für weiter entfernte Bildpunkte. Siehe Tabelle 8.1 für die berechneten Winkelauflösungen.

$$\begin{aligned} \varphi_w &= 1.27 \cdot 10^{-1} \\ \vartheta_w &= 1.24 \cdot 10^{-1} \\ \varphi_t &= 1.58 \cdot 10^{-2} \\ \vartheta_t &= 1.56 \cdot 10^{-2} \end{aligned}$$

Tabelle 8.1: Winkelauflösungen für den Weitwinkel- und Telebereich in  $^\circ/\text{Pixel}$

## 8.5 Objektreferenzpunkt

### 8.5.1 Definition des Referenzpunktes

Die Einschränkungen für die Position eines Referenzpunktes wurden in Abschnitt 8.2 bereits erläutert. Konkret wurde festgelegt, daß der Referenzpunkt  $R$  ein Element der Ebene  $E$  sein muß, die den Fußboden repräsentiert. Zusätzlich soll der Referenzpunkt Teil des gesuchten Objekts sein. Damit ergibt sich als mögliche Position die Unterkante des stehenden Mülleimers. Da es sich bei dem zu suchenden Mülleimer um ein rotationssymmetrisches Objekt handelt, wird der Referenzpunkt folgendermaßen eindeutig definiert:

1. Der Referenzpunkt liegt auf der Kante, die der stehende Mülleimer mit dem Fußboden bildet.
2. Der Referenzpunkt ist der Punkt der Kante, der der Kamera am nächsten ist.

### 8.5.2 Bestimmung des Referenzpunktes

Ist das gesuchte Objekt im Objekterkennungsprozeß gefunden worden, so kann der Referenzpunkt im Bild leicht bestimmt werden. Leider ist die Objekterkennung ein aufwendiger und rechenintensiver Prozeß, der, abhängig von den Eingabedaten,

zwischen einigen Sekunden und mehreren Minuten dauern kann. Für einen flüssigen Ablauf beim Andocken ist dies nicht praktikabel, wenn mehrere Positionsbestimmungen von unterschiedlichen Orten aus erfolgen sollen. Nach dem Objekterkennungsprozeß ist bekannt, daß sich das gesuchte Objekt in der untersuchten Szene befindet. Es kann vorausgesetzt werden, daß während des Andockvorgangs das Objekt an dieser Position verbleibt. Es wäre also überflüssig, bei einer Positionsbestimmung das Objekt wieder als Mülleimer zu identifizieren. Eine einfache Heuristik mit einer Antwortzeit von wenigen Sekunden, die lediglich den Referenzpunkt im Bild bestimmt, ist hier vollkommen ausreichend.

Erreicht wird dies durch eine Klassifizierung der Bilddaten, wie in 2.5.1 beschrieben. Durch ein Größenfilter (siehe 2.4.4) bleiben zu kleine, aber auch zu große Punkthäufungen unberücksichtigt. Die größte verbleibende Punkthäufung wird als Objekthypothese angenommen. Ist die Mitte der Unterkante dieser Hypothese bestimmt, wird dort der Referenzpunkt festgelegt. Ergibt die Positionsbestimmung eine zulässige Position, so wird die Hypothese als bestätigt angesehen und die Position ausgegeben. Ist die Position unzulässig, so wird die Punktmenge dieser Objekthypothese von der weiteren Betrachtung ausgeschlossen und die nächstkleinere Punkthäufung als Objekthypothese angenommen. Dieser Vorgang wird solange fortgesetzt, bis entweder eine Hypothese bestätigt wurde oder keine weitere Hypothese aufgestellt werden kann. Im letzteren Fall bricht die Positionsbestimmung erfolglos ab.

## 8.6 Berechnung der Position

Ist der Referenzpunkt im Bild für eine Punkthäufung bestimmt, wird zuerst eine Koordinatenkorrektur anhand der Gleichungen (8.19) und (8.20) für diesen Punkt im Bild vorgenommen, die den Fehler der Verzerrung bei der Aufnahme durch die Linsenoptik ausgleicht. Für diesen korrigierten Punkt wird anhand von Tabelle 8.1 die Auslenkung des Sichtstrahles in  $x$ - und  $y$ -Richtung bestimmt, auf dem sich der Referenzpunkt befindet.

Aus der durch den Sichtstrahl definierten Gerade und der den Fußboden repräsentierenden Ebene, die durch die Geometrie des Roboters und den Neigungswinkel des Kamerastativs definiert ist, ergibt sich ein Schnittpunkt im Raum, der den Referenzpunkt des Mülleimers aus der Sicht der Kamera darstellt.

Dieser Punkt muß nun in eine für das Robotersystem bekannte Darstellung transformiert werden. Dazu wird der berechnete Punkt im Raum mit der Gleichung (8.21) vom Kamerakoordinatensystem in das Roboterkoordinatensystem übertragen.

$$P_R = T(s_x, s_y, s_z) \cdot R_z(\varphi) \cdot R_y(\vartheta) \cdot T(c_x, c_y, c_z) \cdot P_K \quad (8.21)$$

Dabei ist  $(s_x, s_y, s_z)$  der Vektor vom Roboterursprung zum Drehpunkt des Kamerastativs und  $(c_x, c_y, c_z)$  der Vektor von diesem zum Brennpunkt der Videokamera.  $\varphi$  ist der aktuelle Schwenkwinkel und  $\vartheta$  der aktuelle Neigungswinkel des Stativs relativ zum Roboterkoordinatensystem, unter dem die Kamera die Szene aufgenommen hat.  $P_K$  ist der Originalpunkt im Kamerakoordinatensystem,  $P_R$  derselbe Punkt im Roboterkoordinatensystem dargestellt.

Nachdem die Position des Referenzpunktes im Roboterkoordinatensystem bestimmt ist, muß nun auf die tatsächliche Position des Mülleimers mit dessen Koordinatenursprung, dem Zentrum seiner Grundfläche, geschlossen werden. Da der Mülleimer ein rotationssymmetrisches Objekt ist und der festgelegte Referenzpunkt auf einer Kreisbahn um dessen Zentrum auf dem zum Roboter nächstgelegenen Punkt liegt, wird zur berechneten Position des Referenzpunktes ein Vektor der Länge  $r$  mit  $r$  gleich dem Radius der Mülleimergrundfläche und der Richtung gleich der des Einheitsvektors, der vom Roboterkoordinatenursprung in Richtung Referenzpunkt

zeigt, addiert. Ergebnis ist die Position des Mülleimers, festgelegt anhand seines definierten Nullpunktes, dem Zentrum seiner Grundfläche.

## 8.7 Genauigkeit des Verfahrens

Die Güte der berechneten Position kann unter zwei Gesichtspunkten, der Richtungs-  
genauigkeit und der Entfernungsgenauigkeit, betrachtet werden. Prinzipiell sinkt die  
Genauigkeit des Ergebnisses mit steigender Entfernung zum Zielobjekt.

Bei der Entfernungsschätzung unter einem flachen Kamerawinkel machen sich  
kleine Ungenauigkeiten bei der Bestimmung des Sichtstrahls besonders bemerkbar.  
Ursache dafür ist der kleine Winkel, den Sichtstrahl und Referenzebene (Fußboden)  
aufspannen. Schon minimale Schwankungen dieses Winkels verursachen eine signifi-  
kante Verschiebung des Schnittpunktes von Strahl und Ebene. Bei der Bestimmung  
der Richtung sind die Auswirkungen kleiner Meß- oder Kalibrierungenungenauigkei-  
ten unkritisch.

Praktisch hat sich herausgestellt, daß eine Positionsschätzung für Objekte im  
Umkreis von bis zu drei Metern hinreichend genau ist, um das Ziel zuverlässig  
anzufahren. Ergibt eine erste Messung eine Entfernung von mehr als drei Metern  
zum Objekt, sollte eine zweite Messung erfolgen, wenn sich der Roboter bis auf zwei  
Meter dem Zielobjekt genähert hat. Für die Arbeit in engen Büroräumen ist eine  
zweite Messung jedoch selten erforderlich.

## Kapitel 9

# Entwurf und Implementierung

Dieses Kapitel beschreibt die Voraussetzungen, die Anforderungen und den formalen Aufbau der Positionsbestimmung. Die Interaktion der beteiligten Teilprozesse und der Ablauf der Positionsbestimmung sind beschrieben. Die bei der Implementierung gewählte Aufteilung der Gesamtfunktionalität in einzelne Module ist dokumentiert.

### 9.1 Entwicklungsplattform

Als Entwicklungsplattform stand das autonome System NOMAD, wie in 7.1 bereits erläutert, zur Verfügung. Dieses ist mit einer CCD-Farbkamera, einem fernsteuerbaren Stativ, einer sogenannten *Pan-Tilt Unit*, und einer Framegrabberkarte ausgestattet.

### 9.2 Anforderungen

Die Positionsbestimmung soll aus verschiedenen Ansichten den Standort eines bekannten Objekts ermitteln können. Insbesondere ist hierbei von Interesse, die Messung während der Fahrt vorzunehmen. Dazu ist eine Realisierung notwendig, die die parallele Ausführung der Steuerung ermöglicht.

### 9.3 Ablaufszenario

Nach der positiven Objekterkennung ist die Kamera so eingestellt, daß das zu lokalisierende Objekt im Bild ist. Im Stand wird dann von der Positionsbestimmung eine erste Positionsschätzung vorgenommen. Das Ergebnis wird sowohl der Steuerung übergeben als auch intern gespeichert.

Das Steuerungssystem berechnet einen Weg zur angegebenen Position und legt einen Zwischenpunkt fest, bei dem eine erneute Überprüfung der Objektposition vorgenommen werden soll. Das System fährt an und liefert der Positionsbestimmung die nächste Position, von der aus eine Schätzung erfolgen soll. Anhand der intern gespeicherten Objektposition und der neuen Beobachtungsposition wird die Kamerastellung so berechnet, daß am Zielort das gefundene Objekt exakt in der Bildmitte zu erwarten ist. Die berechnete Stellung wird durch das fernsteuerbare Stativ eingestellt. Ist der Schwenkvorgang beendet, so meldet die Positionsbestimmung die Bereitschaft an das Steuerungssystem. Wird die Position durchfahren, an

der die nächste Bewertung stattfinden soll, meldet dies die Steuerung der Positionsbestimmung, die daraufhin das aktuelle Kamerabild speichert und auswertet. Das Ergebnis der Positionsbestimmung wird der Steuerung zur Verfügung gestellt. Diese berechnet daraus die neue Bahn und der gesamte Zyklus beginnt mit einer neuen Iteration.

Sämtliche Aktionen der Iteration laufen während der Fahrt ab. Einzige Ausnahme ist: das autonome System erreicht die geplante Zwischenstation für eine neue Messung, bevor die Positionsbestimmung ihre Bereitschaft gemeldet hat. In diesem Fall bleibt das System an der geplanten Position stehen und wartet auf die Bereitschaft der Positionsbestimmung. Wird diese signalisiert, so meldet die Steuerung das Eintreffen an der geplanten Zwischenstation und das System fährt weiter.

## 9.4 Realisierungskonzept

Aus den oben aufgeführten Anforderungen und dem Ablaufszenario ergibt sich, daß für die Positionsbestimmung eine eigene Prozeßumgebung geschaffen werden muß. Die Kommunikation zwischen der Steuerung und der Positionsbestimmung muß so realisiert werden, daß sowohl eine unterbrechungsfreie Abarbeitung der Regelschleife seitens der Steuerung als auch eine ressourcensparende Abfrage der Kommunikationsschnittstelle bei Leerlauf seitens der Positionsbestimmung sichergestellt ist.

### 9.4.1 Kommunikationsschema

Die Kommunikation zwischen den beiden beteiligten Prozessen wird mittels *IPC-shared memory* und UNIX-Signalen realisiert. Die *shared memory*-Bereiche dienen zur Datenübermittlung und zur Synchronisation für den Steuerungsprozeß, die Signale werden zur Synchronisation für die Positionsbestimmung verwendet.

#### shared memory-Bereiche

Zur Kommunikation werden zwei Datenstrukturen, eine Kommandostruktur und eine Ergebnisstruktur, verwendet, deren Definition in C-Notation im folgenden dargestellt ist.

```
typedef struct
{
    double x, y;
    double theta;
    char mode;
} shmcmd;
```

```
typedef struct
{
    double x, y;
    char flag;
} shmres;
```

`shmcmd` stellt die Kommandostruktur dar. Die Elemente `x`, `y` und `theta` geben die Koordinate und die Orientierung der nächsten Zwischenposition an. Das Feld `mode` gibt an, wie die gegebene Position zu interpretieren ist und wie weiter verfahren werden soll. Dazu sind die Bitmasken `TERM`, `ABS`, `REL` und `PRE` definiert. `ABS` und `REL` schließen sich gegenseitig aus.

Die Bedeutung der Flags im einzelnen ist:

- `TERM` führt zur Beendigung der Positionsbestimmung (absolute Priorität).

- **ABS** Die angegebene Position im Raum ist absolut, ausgehend vom Weltkoordinatensystem.
- **REL** Die angegebene Position im Raum ist relativ zur letzten bekannten Position.
- **PRE** Vor der ersten Iteration wird die Kamera so eingestellt, daß ein fiktives Objekt an der durch  $x$  und  $y$  angegebenen Position in der Mitte des Kamerabildes erscheint.

`shmres` definiert die Ergebnisstruktur. Das Feld `flag` dient zur Synchronisation der Steuerung mit der Positionsbestimmung. Gültige Werte sind `READY`, `VALID` und `INVALID`. Durch `READY` signalisiert die Positionsbestimmung der Steuerung, daß die Kameraeinstellung abgeschlossen ist und eine weitere Iteration der Positionsbestimmung durchgeführt werden kann. `VALID` gibt an, daß die Positionsbestimmung abgeschlossen und die durch  $x$  und  $y$  angegebene Position gültig ist.  $x$  und  $y$  geben dabei die Position in dem Modus an, der in der Kommandostruktur vereinbart wurde. `INVALID` wird im Fehlerfall verwendet, um der Steuerung die Ergreifung geeigneter Maßnahme zu ermöglichen.

### Signale

Um keine Rechenzeit für das Polling der Kommandostruktur zu verschwenden, geht die Positionsbestimmung in Leerlaufzeiten in einen passiven Wartemodus. Durch den Aufruf von `pause()` gibt der Prozeß die CPU frei und erhält diese erst wieder, wenn ein Signal an den Prozeß geschickt wird.

Hat die Steuerung das Kommandofeld mit einem neuen Auftrag versorgt, wird ein Signal an die Positionsbestimmung gesendet. Diese holt daraufhin den Auftrag ab oder nimmt kontextbezogen eine neue Positionsschätzung vor.

### 9.4.2 Iterationskreislauf

Ist die Objekterkennung positiv abgeschlossen worden, so fordert die Steuerung den *shared memory* für die Kommando- und Ergebnisstruktur an, initialisiert diese und übergibt deren Zugriffsschlüssel beim Aufruf der Positionsbestimmung.

Die Positionsbestimmung wertet die Kommandostruktur sofort aus und führt anschließend eine erste Positionsschätzung durch. Das Ergebnis wird in die Ergebnisstruktur eingetragen und das Feld `flag` auf `VALID` gesetzt. Die Steuerung pollt unterdessen bei jedem Durchgang der Regelschleife die Ergebnisstruktur auf das `VALID-flag`. Sobald dieses gesetzt ist, berechnet die Steuerung einen Weg zum Ziel und trägt die Zwischenposition in das Kommandofeld ein. Durch das Senden eines Signals an die Positionsbestimmung wird diese aufgefordert, die Kamera auf die zu erwartende Position einzurichten. Ist die Positionierung abgeschlossen, wird von der Positionsbestimmung das `flag` in der Ergebnisstruktur auf `READY` gesetzt. Ist der Roboter an der geplanten Zwischenposition angekommen, so sendet die Steuerung ein Signal an die Positionsbestimmung, wenn das `READY-flag` gesetzt ist. Ist dies nicht der Fall, bleibt der Roboter an der Zwischenposition stehen und wartet auf die Bestätigung der Positionsbestimmung. Das Signal wird gesendet, sobald diese eingegangen ist.

Abbildung 9.1 zeigt den Iterationskreislauf seitens der Positionsbestimmung.

## 9.5 Ablaufdiagramm der Positionsbestimmung

In Abbildung 9.2 ist das Ablaufdiagramm für die Positionsbestimmung dargestellt. In den Kästchen sind die einzelnen Funktionen eingetragen, die Vektoren symboli-

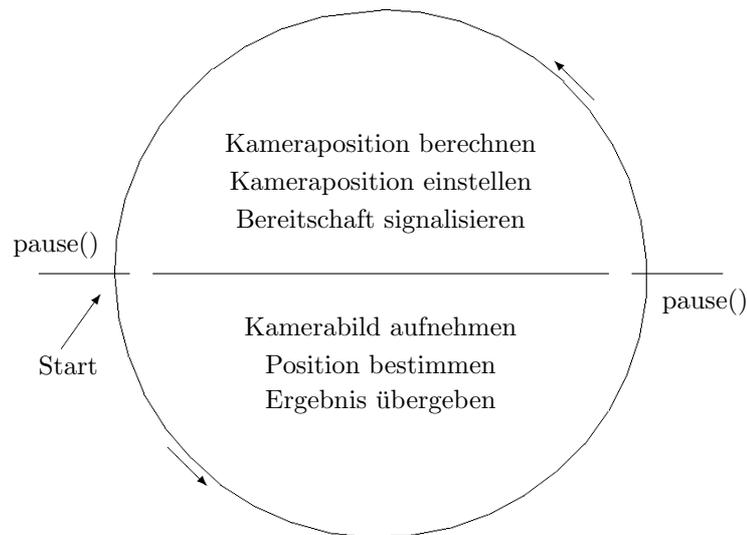


Abbildung 9.1: Iterationskreislauf

sieren den Programmfluß. Die Beschriftung neben den Vektoren zeigt das jeweilige Zwischenergebnis an.

## 9.6 Aufteilung der Funktionalität

Die Implementierung der benötigten Funktionalität wurde auf verschiedene Module aufgeteilt. Dabei wurden die Funktionen nach Funktionsgruppen zusammengefaßt.

Abbildung 9.3 zeigt die Modulstruktur der Implementierung.

### Modul `localizer`

Das Modul `localizer` stellt die *mainline* der gesamten Positionsbestimmung dar. Nachdem administrative Tätigkeiten, wie das *attachen* an die *shared memory*-Bereiche oder das Impfen gegen Signale, erledigt sind, wird die Iteration eingeleitet. Die Kommunikation zum Steuerungssystem ist ebenfalls hier implementiert.

### Modul `visual`

Im Modul `visual` sind alle Funktionen und Hilfsfunktionen vereint, die zur Bildverarbeitung notwendig sind. Dazu gehören die Farbsegmentierung mit `findpoi()`, die Größenbestimmung der einzelnen Farbsegmente mit `findacc()` sowie die Eliminierung der Bildverzerrung mit `erasedisortion()`.

### Modul `geometry`

Das Modul `geometry` realisiert die komplexen Koordinatenberechnungen und Koordinatentransformationen. Beispiele dafür sind `target()` zur Bestimmung der Position aus dem Referenzpunkt, `cam2rbot()` zur Berechnung der Transformation vom Kamera- in das Roboterkoordinatensystem und `aim()` zur Ausrichtung der Kamera auf einen vorgegebenen Punkt.

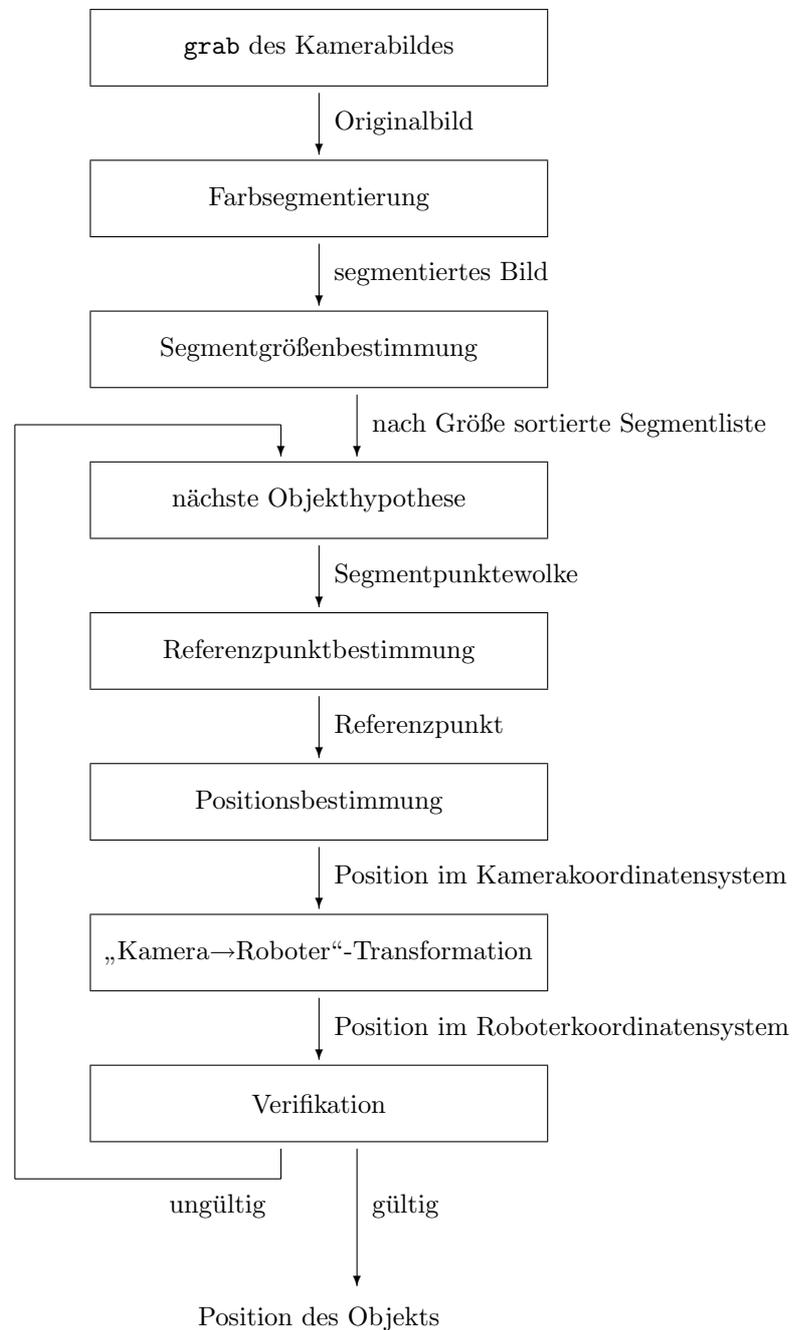


Abbildung 9.2: Ablaufdiagramm der Positionsbestimmung

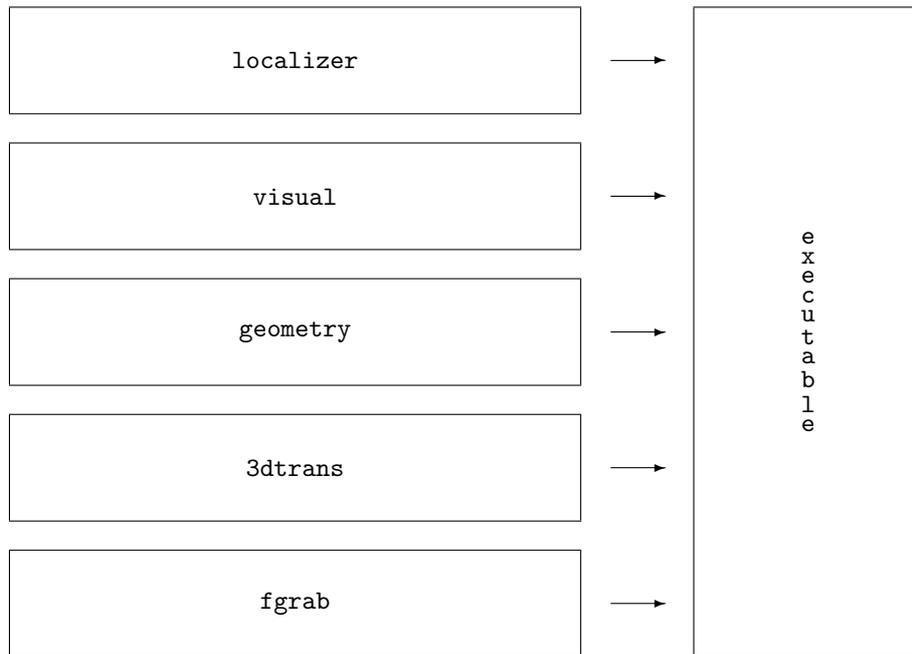


Abbildung 9.3: Modulstruktur der Positionsbestimmung

### Modul 3dtrans

Alle Funktionen und Hilfsfunktionen für die Definition einfacher Koordinatentransformationen sind im Modul `3dtrans` implementiert. Die Erstellung von Translationsmatrizen mit `translation()` und Rotationsmatrizen mit `drotation()` gehört ebenso zur Ausstattung wie die Realisierung von Matrix-Multiplikation und Matrix-Vektor-Multiplikation durch `mm_multiply()` und `mv_multiply()`. Die Transformation von Grad in Bogenmaß und umgekehrt ist ebenfalls in diesem Modul implementiert.

### Modul fgrab

Das Modul `fgrab` stellt eine Abstraktionsebene zwischen der C-API der Framegrabberkarte und der Implementierung der Positionsbestimmung dar. Sie übernimmt automatisch die Initialisierung der Karte sowie die Konvertierung der Bilddaten in das geforderte Format.

# Kapitel 10

## Ausblick

Als Erweiterung der hier vorgestellten Lösung sind zwei Ansätze denkbar.

Eine unterschiedliche Gewichtung verschiedener Teile des Modells könnte verwendet werden, um in bestimmten Bereichen eine besonders gute Korrelation von Modell- und Bildpunkten zu fordern – oder umgekehrt, Bereiche festzulegen, in denen eine geringe Korrelation zwischen Modell und Bild toleriert wird. So könnte verhindert werden, daß unvorhersehbare, jedoch zulässige Veränderungen im Bereich des gesuchten Objekts eine korrekte Erkennung beeinträchtigen. Beispiel dafür ist der Abfall in einem Papierkorb. Ist dieser aus der Kameraperspektive zu sehen, so enthält das Bild zusätzliche Kanten beliebiger Form im Bereich des gesuchten Objekts, die besonders die Verifikationsdistanz negativ beeinflussen. Eine verminderte Sensitivität der Verifikationsdistanz in diesem Teil des Modells wäre für die Robustheit des gesamten Objekterkennungsprozesses durchaus förderlich.

Die Greifvorrichtung des Roboters ist so konzipiert, daß diese das Objekt am oberen Rand faßt. Eine Erweiterung dieser Greifvorrichtung zu einem Greifarm ist bereits in Planung. Dadurch ist es dem Roboter möglich, um ein Hindernis herum zu greifen. Ein seitlich verdecktes Objekt stellt somit kein Problem dar. Ist allerdings der obere Rand des Objektes verdeckt, so hat der Roboter keine Möglichkeit das Objekt zu fassen. Durch eine Aufteilung des Modells in mehrere logische Einheiten, beispielsweise *Oberkante links/Mitte/rechts*, *Unterkante links/Mitte/rechts*, *Linke Seite oben/Mitte/unten* und *Rechte Seite oben/Mitte/unten*, wäre leicht feststellbar, ob unkorrelierte Modellpunkte auf Störungen bei der Bildvorverarbeitung zurückzuführen sind oder ob eine teilweise Verdeckung des gefundenen Objekts dazu geführt hat – und wenn ja, welche Teile davon betroffen sind.



# Literaturverzeichnis

- [AF86] N. Ayache and O. Faugeras. HYPER: A new Approach for the Recognition and Positioning of Two-dimensional Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44–54, January 1986.
- [AJ94] T. D. Alter and D. W. Jacobs. Error Propagation in Full 3D-from-2D Object Recognition. *Proceedings on Computer Vision and Pattern Recognition*, pages 892–898, 1994.
- [BH86] R. C. Bolles and P. Horaud. 3DPO: A Three-dimensional Part Orientation System. *International Journal of Robotics Research*, 5(3):3–26, 1986.
- [BO92] G. Borgefors and H. Olsson. Localizing and Identifying Objects: A Method for Distinguishing Noise, Occlusion and Other Disturbances. *Proceedings on Second Nordic Workshop on Industrial Machine Vision*, March 1992.
- [Bru91] T. M. Bruel. Fast Recognition Using Adaptive Subdivision of Transformation Space. Technical Report 1313, M.I.T: AI Lab, February 1991.
- [BTBW77] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric Correspondence and Champfer Matching: Two New Techniques for Image Matching. *Proceedings on Fifth International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.
- [Cas90] T. A. Cass. Feature Matching for Object Localization in the Presence of Uncertainty. *Proceedings on Third International Conference on Computer Vision*, pages 360–364, 1990.
- [Cas93] T. A. Cass. *Polynomial-Time Geometric Matching for Object Recognition*. PhD thesis, Massachusetts Institute of Technology, 1993.
- [CCH94] Zen Chen, Tsorng-Lin Chia, and Shinn-Ying Ho. Measuring 3-D Location and Shape Parameters of Cylinders by a Spatial Encoding Technique. *IEEE Transactions on Robotics and Automation*, 10(5):632–647, October 1994.
- [CF82] Paul R. Cohen and Edward A. Feigenbaum, editors. *The Handbook of Artificial Intelligence*, volume III, chapter XIII: Vision. Addison-Wesley Publishing Company, Inc., Reading MA., Workingham (England), Amsterdam, Sydney, Tokyo, 1982.
- [FH93] P. Fua and A. J. Hanson. An Optimization Framework for Feature Extraction. *Machine Vision Applications*, 4:59–87, 1993.

- [FKPS] R. James Firby, Roger E. Kahn, Peter N. Prokopowicz, and Michael J. Swain. An Architecture for Vision and Action. Technical report, Department of Computer Science, University of Chicago, 1100 E. 58th St., Chicago, IL 60637.
- [FvDF<sup>+</sup>94] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, and Richard L. Phillips. *Introduction to Computer Graphics*. Addison-Wesley Publishing Company, Inc., New York, Amsterdam, Bonn, Sydney, Singapore, Tokyo, Madrid, San Juan, Milan, Paris, 1994.
- [GH90a] W. Eric L. Grimson and Daniel P. Huttenlocher. On the Sensitivity of Geometric Hashing. *Proceedings on Third International Conference on Computer Vision*, pages 334–338, 1990.
- [GH90b] W. Eric L. Grimson and Daniel P. Huttenlocher. On the Sensitivity of the Hough Transform for Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):255–274, March 1990.
- [GH91] W. Eric L. Grimson and Daniel P. Huttenlocher. On the Verification of Hypothesized Matches in Model-Based Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(12):1201–1213, December 1991.
- [GHJ92] W. Eric L. Grimson, Daniel P. Huttenlocher, and D. W. Jacobs. A Study of Affine matching with Bounded Sensor Error. *Proceedings on Second European Conference on Computer Vision*, pages 291–306, May 1992.
- [GLP87] W. Eric L. Grimson and T. Lozano-Pérez. Localizing Overlapping Parts by Searching the Interpretation Tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, July 1987.
- [Gri90a] W. Eric L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. Artificial Intelligence. The MIT Press, Cambridge MA., London (England), 1990.
- [Gri90b] W. Eric L. Grimson. The Combinatorics of Object Recognition in Cluttered Environments Using Constrained Search. *Artificial Intelligence*, 44:121–165, 1990.
- [Hab87] Peter Haberäcker. *Digitale Bildverarbeitung – Grundlagen und Anwendungen*. Carl Hanser Verlag, München, Wien, 1987.
- [HKR93] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.
- [Hop95] M. J. Hopcroft. *A Geometrical Approach to Model Based Recognition*. PhD thesis, Cornell University, January 1995.
- [HR] Daniel P. Huttenlocher and William J. Rucklidge. A Multi-Resolution Technique for Comparing Images Using the Hausdorff Distance. Technical report, Department of Computer Science, Cornell University, Ithaca, NY 14853.
- [HU90] Daniel P. Huttenlocher and S. Ullman. Recognizing Solid Objects by Alignment with an Image. *International Journal of Computer Vision*, 5(2):195–212, 1990.

- [Hua93] Thomas S. Huang. Introduction to the Special Section on 3-D Modeling in Image Analysis and Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):529–530, June 1993.
- [Jäh89] Bernd Jähne. *Digitale Bildverarbeitung*. Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1989.
- [Jac91] D. W. Jacobs. Optimal Matching of Planar Models in 3D Scenes. *Proceedings on Computer Vision and Pattern Recognition*, pages 269–274, 1991.
- [JKS95] Ramesh Jain, Rangachar Kasturi, and Brian G. Schunk. *Machine Vision*. Artificial Intelligence. McGraw-Hill, Inc., New York, San Francisco, London, Madrid, Sydney, Tokyo, 1995.
- [KZ92] Reinhard Klette and Piero Zamperoni. *Handbuch der Operatoren für die Bildbearbeitung*. Friedrich Vieweg & Sohn, Braunschweig, Wiesbaden, 1992.
- [LMZ] Stefan Lanser, Olaf Munkelt, and Christoph Zierl. Robuste videobasierte Identifizierung von Hindernissen und Werkstücken sowie die Bestimmung ihrer räumliche Lage. Technical report, TU München, Institut für Informatik, Orleanstr. 34, D-81667 München.
- [Low87] David G. Lowe. Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, 31:355–395, 1987.
- [Low91] David G. Lowe. Fitting Parameterized Three-Dimensional Models to Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.
- [LW88] Y. Lamdan and H. J. Wolfson. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. *Proceedings on Second International Conference on Computer Vision*, pages 238–249, 1988.
- [Mar82] David Marr. *Vision – A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman and Company, New York, 1982.
- [Nib86] Wayne Niblack. *An Introduction to Digital Image Processing*. Prentice Hall International, London, Sydney, Toronto, Tokyo, Singapore, Rio de Janeiro, 1986.
- [Ols94] C. F. Olson. Time and Space Efficient Pose Clustering. *Proceedings on Computer Vision and Pattern Recognition*, pages 251–258, 1994.
- [Pag92] D. W. Paglieroni. Distance Transforms: Properties and Machine Vision Applications. *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, 54(1):56–74, 1992.
- [Pop94] R. Pope. Model-Based Object Recognition – A Survey of Recent Research. Technical Report 94–04, Univeristy of British Columbia, January 1994.
- [PSSK96] Erwin Praßler, Eleni Stroulia, Matthias Strobel, and Thomas Kämpke. Mobile Robots in Office Logistics. Technical report, Research Institute of Applied Knowledge Processing (FAW), P.O. Box 2060, D-89010 Ulm, Germany, October 1996.

- [Ruc] William J. Rucklidge. Efficient Computation of the Minimum Hausdorff Distance for Vision Recognition. Technical report, Department of Computer Science, Cornell University, Ithaca, NY 14853.
- [SFH92] Paul Suetens, Pascal Fua, and Andrew J. Hanson. Computational Strategies for Object Recognition. *ACM Computing Surveys*, 24(1), March 1992.
- [SHD86] T. M. Silberberg, D. A. Harwood, and L. S. Davis. Object Recognition Using Oriented Model Points. *Computer Vision, Graphics and Image Processing*, 35:47–71, 1986.
- [SP94] Stan Sclaroff and Alex P. Pentland. Object Recognition and Categorization Using Modal Matching. Technical Report 267, M.I.T. Media Laboratory Perceptual Computing Section, February 1994.
- [TM87] D. W. Thompson and J. L. Mundy. Three-dimensional Model Matching from an Unconstrained Viewpoint. *Proceedings on International Conference on Robotics and Automation*, pages 208–220, 1987.
- [Tsa87] Roger Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, August 1987.
- [Zam89] Piero Zamperoni. *Methoden der digitalen Bildsignalverarbeitung*. Friedrich Vieweg & Sohn, Braunschweig, Wiesbaden, 1989.
- [ZSB93] Shujun Zhang, Geoff D. Sullivan, and Keith D. Baker. The Automatic Construction of a View-Independent Relational Model for 3-D Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):531–544, June 1993.