

# Masterarbeit

## Konturbasierte Objekterkennung aus Tiefenbildern eines 3D-Laserscanners

vorgelegt von:

Stefan Stiene

Universität Osnabrück  
Fachbereich Informatik

Erstgutachter: Prof. Dr. Joachim Hertzberg  
Zweitgutachter: Prof. Dr. Martin Riedmiller



## Zusammenfassung

Objekterkennung spielt in verschiedensten Industrie- und Forschungsbereichen eine zentrale Rolle. Einem mobilen Roboter soll sie beispielsweise ermöglichen, seine Umwelt nicht nur aufzunehmen und sich in dieser bewegen zu können, sondern diese zu interpretieren und mit ihr zu interagieren. Insbesondere ist konturbasierte Objekterkennung ein schon seit Mitte des letzten Jahrhunderts intensiv betriebenes Forschungsgebiet. Die dadurch entstandene Vielzahl der zur Verfügung stehenden Verfahren sowie die einfache Extraktion der Objektkonturen aus Tiefenbildern motivieren diese Arbeit.

Die Tiefenbilder werden mit Hilfe eines 3D-Laserscanners, der auf dem mobilen Roboter Kurt3D installiert ist, aufgenommen. Die Segmentierung der Tiefenbilder mittels adaptivem Schwellwertverfahren wird durch eine semantische Betrachtung der Messwerte erleichtert. Diese entfernt Bodenpunkte im Tiefenbild und markiert Tiefensprünge. Die aus dem segmentierten Tiefenbild extrahierten Konturen werden durch verschiedene Merkmale beschrieben und mittels einer Support-Vector-Machine klassifiziert. Es werden geometrische Merkmale, Hu- und Zernike-Momente, die Curvature-Scale-Space-Repräsentation, das Border-Signature-Verfahren sowie die Angular-Radial-Transformation eingesetzt. Die Performanz der Klassifizierer wird mit Hilfe der Receiver-Operating-Characteristics-Analyse untersucht.

## Abstract

Object detection plays an important role in different industrial and research domains. For example it ought to enable a mobile robot not only to map its environment and maneuver in it, but also to interpret and interact with it. In particular, contour-based object detection is a research domain since mid last century. The resulting multiplicity of available methods and the simple contour extraction from range images motivate this work.

The range images are obtained by a 3D laser scanner, that is mounted on the mobile robot Kurt3D. The range image segmentation by adaptive thresholding is simplified by preprend interpretation of the measured data. The interpretation removes the ground points in the range image and marks range steps. The segmented range images extracted from contours, are described using several features and classified by a Support Vector Machine. Geometric features, Hu and Zernike moments, the Curvature Scale Space representation, the Border Signature method as well as the Angular Radial Transformation are used. The performances of the classifiers are tested with the Receiver Operating Characteristics Analysis.



## Danksagung

Zunächst möchte ich Professor Joachim Hertzberg danken, der mich während der gesamten Arbeit mit konstruktiver Kritik und hilfreichen Beiträgen unterstützt hat. Des Weiteren möchte ich Andreas Nüchter und Kai Lingemann für die Betreuung während der gesamten Arbeit danken. Ihre programmiertechnische und fachliche Erfahrung hat diese Arbeit erst ermöglicht.

Dank gilt auch Annika Schomaker sowie allen anderen Korrekturlesern. Abschließend möchte ich noch meinen Eltern danken, die mir durch ihre finanzielle Unterstützung dieses Studium ermöglicht haben.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Stufen eines Bildverarbeitungssystems . . . . .	2
1.2	Aufbau der Arbeit . . . . .	2
1.3	Wissenschaftlicher Beitrag . . . . .	3
<b>2</b>	<b>Stand der Technik</b>	<b>5</b>
2.1	Tiefenbild Segmentierungstechniken . . . . .	5
2.2	Objekterkennung . . . . .	7
2.2.1	Template Matching . . . . .	8
2.2.2	Principal-Component-Analysis (PCA) . . . . .	8
2.2.3	Klassifikation von Merkmalen . . . . .	9
<b>3</b>	<b>Tiefenbilder</b>	<b>11</b>
3.1	Erzeugung von Tiefenbildern . . . . .	11
3.2	Eigenschaften von Tiefenbildern . . . . .	13
<b>4</b>	<b>Konturextraktion</b>	<b>15</b>
4.1	Bildvorverarbeitung . . . . .	16
4.1.1	Kontrastverbesserung . . . . .	16
4.1.2	Glättung durch lokale Operatoren . . . . .	16
4.1.3	Glättung durch morphologische Operatoren . . . . .	19
4.2	Segmentierungsverfahren . . . . .	22
4.2.1	Canny Kantendetektion . . . . .	23
4.2.2	$k$ -means Clustering . . . . .	25
4.2.3	Schwellwertverfahren . . . . .	26
4.3	Konturextraktion . . . . .	29
4.4	Verbesserungen am Eingabebild . . . . .	29

4.4.1	Parallele Projektion . . . . .	30
4.4.2	Entfernung der Bodenpunkte im Tiefenbild . . . . .	30
4.4.3	Manipulation der Grauwertinterpolation . . . . .	31
4.5	Endfassung des Konturextraktionsalgorithmus . . . . .	32
<b>5</b>	<b>Konturrepräsentation und Merkmalsextraktion</b>	<b>35</b>
5.1	Anforderungen an die Konturrepräsentationen . . . . .	36
5.2	Anforderung an die Merkmale . . . . .	37
5.3	Einfache geometrische Merkmale . . . . .	38
5.4	Merkmalsextraktion mittels Momenten . . . . .	41
5.4.1	Hu-Momente . . . . .	42
5.4.2	Zernike-Momente . . . . .	42
5.5	Repräsentation durch den Curvature-Scale-Space (CSS) . . . . .	44
5.5.1	Krümmung einer ebenen Kurve . . . . .	44
5.5.2	Eigenschaften der CSS-Repräsentation . . . . .	46
5.5.3	Merkmalsextraktion mit der Eigen-CSS Methode . . . . .	46
5.5.4	CSS-Repräsentation zur Beschreibung von teilweise verdeckten Objekten . . . . .	50
5.6	Angular-Radial-Transformation (ART) . . . . .	52
5.7	Repräsentation durch Fourier-Deskriptoren . . . . .	53
5.8	Border-Signature . . . . .	54
<b>6</b>	<b>Klassifikation</b>	<b>55</b>
6.1	Merkmalsauswahl . . . . .	55
6.1.1	plus $l$ -take away $r$ Algorithmus . . . . .	56
6.1.2	Principal-Component-Analysis (PCA) . . . . .	56
6.2	Klassifikationsstrategien . . . . .	56
6.3	Klassifikation mittels Support-Vector-Machines (SVM) . . . . .	57
6.3.1	Primäre Form . . . . .	57
6.3.2	Duales Optimierungsproblem . . . . .	59
6.3.3	SVM zur Klassifikation von mehreren Objekten . . . . .	60
6.4	Viola Jones AdaBoost Klassifikator . . . . .	62

<b>7</b>	<b>Performanzevaluation</b>	<b>65</b>
7.1	Receiver-Operating-Characteristics (ROC)-Analyse . . . . .	65
7.1.1	ROC-Kurven . . . . .	67
7.1.2	Fläche unter der ROC-Kurve (AUC) . . . . .	67
7.1.3	Multi-Klassen ROC-Analyse . . . . .	67
7.1.4	Konvexe Hülle der ROC-Kurve und kombinierte Klassifikatoren . . . . .	68
7.1.5	Linien gleicher Exaktheit und der optimale Schwellwert . . . . .	69
7.2	Ergebnis der Performanz-Analyse . . . . .	70
7.2.1	Vergleich mit dem Viola Jones AdaBoost Klassifikator . . . . .	78
7.2.2	Klassifikation von verdeckten Konturen . . . . .	79
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>81</b>
<b>A</b>	<b>Freeman-Kettencode</b>	<b>83</b>
<b>B</b>	<b>Curvature-Scale-Space-Matching-Verfahren nach Mokhtarian</b>	<b>85</b>



# Abbildungsverzeichnis

1.1	Mobiler Roboter Kurt3D . . . . .	2
2.1	Parameteroptimierung für Segmentierungsverfahren . . . . .	8
3.1	Verwendter 3D-Laserscanner mit Kippvorrichtung und zugehöriges Koordinatensystem	12
3.2	Seitenansicht eines Laserscans und Tiefenbild . . . . .	13
4.1	Unterschied zwischen Bildpunkt- und lokalen Operationen . . . . .	15
4.2	Look-Up-Table-Kennlinien einer Grauwertdehnung . . . . .	17
4.3	Kontrasterhöhung des Tiefenbildes . . . . .	17
4.4	Glätten eines Tiefenbildes . . . . .	18
4.5	Verschiedene zweidimensionale Strukturelemente . . . . .	19
4.6	Wirkungsweise der Erosionsoperation . . . . .	20
4.7	Wirkungsweise der Dilatationsoperation . . . . .	20
4.8	Wirkungsweise der Öffnungsoperation . . . . .	21
4.9	Wirkungsweise der Schließungsoperation . . . . .	22
4.10	Orientierungsdiskretisierung der Gradientenvektoren bei der Canny-Kantendetektion	24
4.11	Experimentelles Ergebnis der Konturextraktion durch Canny-Kantendetektion . . .	24
4.12	Segmentierung mit $k$ -Means Verfahren . . . . .	25
4.13	Segmentierung mit globalem Schwellwertverfahren . . . . .	26
4.14	Grauwertbild und zugehöriges Histogramm . . . . .	27
4.15	Wirkungsweise eines kombinierten Verfahrens aus adaptivem und lokalem Schwellwert	28
4.16	Segmentierung mit adaptivem und lokalem Schwellwert . . . . .	28
4.17	Perspektivische und parallele Projektion der 3D-Datenpunkte . . . . .	30
4.18	Entfernung der Bodenpunkte aus dem Tiefenbild und Markierung von Tiefensprüngen	32
4.19	Schematischer Ablauf der Konturextraktion . . . . .	33
5.1	Vergleich von regionenorientierten und konturorientierten Formmerkmalen . . . . .	36

5.2	Taxonomie der Formrepräsentationstechniken . . . . .	37
5.3	Einfache geometrische Konturmerkmale . . . . .	40
5.4	Entstehungsweise der CSS-Repräsentation aus einer Kontur . . . . .	46
5.5	CSS $\mathbf{r}$ -Vektoren vor der Phasenkorrelation . . . . .	48
5.6	CSS $\tilde{\mathbf{r}}$ -Vektoren nach der Phasenkorrelation . . . . .	49
5.7	Erstellung des Merkmalsvektors einer Kontur mit dem Eigen-CSS Verfahren . . . . .	49
5.8	CSS-Darstellung der Kontur eines stehenden Menschen . . . . .	51
5.9	CSS-Darstellung von verschiedenen Teilkonturen eines Menschen . . . . .	51
5.10	Darstellung der ersten 36 Basisfunktionen der ART-Repräsentation . . . . .	52
5.11	Beschreibung einer Kontur durch Fourierelipsen . . . . .	53
5.12	Diskretisierung des zweidimensionalen Raums beim Border-Signature-Verfahren . . . . .	54
6.1	Darstellung der von der SVM erstellten optimalen Hyperebene . . . . .	58
6.2	Transformation linear nicht trennbarer Daten in einen höherdimensionalen Raum . . . . .	60
6.3	Multiklassen SVM mit gerichteten azyklischen Graphen . . . . .	61
6.4	Einige Viola Jones Merkmale . . . . .	63
7.1	ROC-Graph für fünf diskrete Klassifikatoren . . . . .	66
7.2	ROC-Kurven für zwei Klassifikatoren und für den aus diesen kombinierten Klassifikator . . . . .	68
7.3	Exaktheit Isometriken im ROC-Graphen . . . . .	69
7.4	AUC-Wert in Abhängigkeit zur Anzahl der verwendeten Basisvektoren . . . . .	71
7.5	ROC-Kurven der Klassifikatoren (K1-K7) . . . . .	73
7.6	ROC-Kurven der Klassifikatoren (K8-K14) . . . . .	75
7.7	ROC-Kurven der Klassifikatoren (K15-K20) sowie (K22) . . . . .	77
7.8	Vergleich der Robustheit der Klassifikatoren gegenüber Verdeckung . . . . .	79
A.1	Richtungsschema zur Erzeugung des Freeman Kettencodes . . . . .	83

# Tabellenverzeichnis

6.1	Häufig verwendete Kernel für Support-Vector-Machines . . . . .	60
7.1	Vergleich der Klassifikatoren für das Objekt „Mensch 1“ . . . . .	72
7.2	Vergleich der Klassifikatoren für das Objekt „Mensch 2“ . . . . .	74
7.3	Vergleich der Klassifikatoren für das Objekt „Kurt 2D“ . . . . .	76
7.4	Vergleich von Viola Jones und konturorientierten Klassifikatoren . . . . .	78



# Abkürzungsverzeichnis

3D	dreidimensional
ART	Angular-Radial-Transformation
AUC	Area under Curve
CSS	Curvature-Scale-Space
KURT3D	?
LUT	Look-Up-Table
MPEG-7	Multimedia Content Description Interface
PCA	Principal-Component-Analysis
ROC	Receiver-Operating-Characteristics
SBS	sequentielle Rückwärtsselektion
SFS	sequentielle Vorwärtsselektion
SVD	Single-Value-Decomposition
SVM	Support-Vector-Machine
UB	Universität Bern
UE	University of Edinburgh
USF	University of South Florida
WSU	Washington State University



# Symbolverzeichnis

$A(x, y)$	verarbeitetes Grauwertbild
$B(x, y)$	Grauwertbild
$\varepsilon_B(X)$	Erosion einer Menge X mit Strukturelement B
$\delta_B(X)$	Dilatation einer Menge X mit Strukturelement B
$\gamma_B(X)$	Öffnung einer Menge X mit Strukturelement B
$\phi_B(X)$	Schließung einer Menge X mit Strukturelement B
$m_{p,q}$	geometrisches Moment
$\mu_{p,q}$	zentrales Moment
$\eta_{p,q}$	normiertes zentrales Momente
$h_i$	Hu-Moment
$A_{nm}$	Zernike-Momente
$tp$	true positive
$fp$	false positive
$tn$	true negative
$fn$	false negative



# Kapitel 1

## Einleitung

Damit ein mobiler Roboter autonom arbeiten kann, muss er in der Lage sein, seine Umgebung über verschiedene Sensoren aufnehmen und analysieren zu können. Um sich in seiner Umwelt zu bewegen, muss der Roboter im Stande sein, sich zu lokalisieren und Hindernisse zu erkennen. Sobald der Roboter mit seiner Umwelt interagieren möchte, muss er Objekte wahrnehmen, sie lokalisieren und falls bereits bekannt, wiedererkennen können. Diese autonome Objekterkennung ist ein interdisziplinäres und seit Jahrzehnten intensiv betriebenes Forschungsgebiet. Sie wird unter anderem in der Medizin zur Auswertung von Röntgenbildern, zur militärischen und geowissenschaftlichen Auswertung von Satellitendaten oder zur Qualitätsprüfung der Produktion in der Industrie eingesetzt.

In dieser Arbeit wird die Objekterkennung über Tiefenbilder eines Laserscanners realisiert, der auf dem mobilen Roboter Kurt3D installiert ist (siehe Abb. 1.1). Objekterkennung soll es diesem ermöglichen, die vor ihm befindliche Szene zu erkennen und zu interpretieren. Die Objekte werden mit aus dem Tiefenbild extrahierten Konturen erkannt. Dieses Verfahren wird angewendet, da die konturorientierte Objekterkennung als eine der ersten Objekterkennungsstrategien entwickelt wurde. Aus diesem Grund gibt es eine Vielzahl von Konturextraktionsverfahren sowie viele konturorientierte Merkmale, die eine Objekterkennung ermöglichen. Ein weiterer Grund konturorientierte Verfahren zu verwenden, ist die relativ einfache Extraktion der Konturen aus einem Tiefenbild. So ermöglicht die Eigenschaft der Tiefenbilder, die Information über die Entfernung der Objekte in der Szene zu speichern, eine beleuchtungsunabhängige Konturextraktion. Ferner besitzen Punkte einer relativ planaren Objektfläche die selbe Entfernung zum Scanner und somit einen relativ gleichbleibenden Grauwert im Tiefenbild. An Objekträndern entsteht ein leicht identifizierbarer Tiefensprung.

Objekterkennung wird üblicherweise in die beiden Verarbeitungsschritte Hypothese und Test aufgeteilt. In der ersten Phase werden anhand einer Objektdatenbank Hypothesen aufgestellt, welches Objekt sich wo in einem Bild befindet. Dazu werden Merkmale der Datenbank mit aus dem Bild gewonnenen Merkmalen verglichen. Diese Merkmale werden auf unterschiedlichste Weise extrahiert. In dieser Arbeit werden ein Teil der kontur- bzw. formorientierten Merkmale verwendet. In der Testphase wird ein Modell des erkannten Objektes in das Bild projiziert und mittels Vergleichsverfahren entschieden, in wie weit das Objekt im Bild und das projizierte Objekt übereinstimmen. Ist die Übereinstimmung über einem definierten Schwellwert, wird die Klassifikationshypothese akzeptiert [15]. In dieser Arbeit wird ausschließlich die Hypothesenphase bearbeitet.



**Abbildung 1.1:** Mobiler Roboter Kurt3D

## 1.1 Stufen eines Bildverarbeitungssystems

Dieser Abschnitt beschreibt das klassische Bildverarbeitungssystem. Es wird hier kurz erwähnt, da in dieser Arbeit jeder der fünf Verarbeitungsschritte bearbeitet wird. Es soll dem Leser eine Zuordnung der verschiedenen Kapitel zu den Stufen des Bildverarbeitungssystems ermöglichen. Die einzelnen Schritte werden in den jeweiligen Kapiteln ausführlich beschrieben. Die fünf Verarbeitungsschritte des Bildverarbeitungssystems lauten [1]:

1. Bilderfassung
2. Bildvorverarbeitung
3. Segmentierung
4. Merkmalsextraktion
5. Klassifikation

In einigen Bildverarbeitungssystemen kommt noch eine sechste Phase, die Interpretationsphase hinzu. In dieser Phase wird die betrachtete Szene anhand der erkannten Objekte interpretiert.

## 1.2 Aufbau der Arbeit

**Kapitel 1:** Im ersten (diesem) Kapitel wird der generelle Aufbau eines Bildverarbeitungssystems sowie der Aufbau dieser Arbeit beschrieben. Des Weiteren wird auf den wissenschaftlichen Beitrag eingegangen, den diese Arbeit liefert.

**Kapitel 2:** In diesem Kapitel wird ein Überblick über den Stand der Forschung gegeben. Es werden unterschiedliche Techniken beschrieben, um ein Tiefenbild zu segmentieren. Ferner werden unterschiedliche Herangehensweisen bei der Objekterkennung aufgezeigt und für die hier verwendete Objekterkennungsstrategie werden weitere in dieser Arbeit nicht verwendete Verfahren dargestellt.

- Kapitel 3: Im dritten Kapitel werden Tiefenbilder als Ausgangsdaten der Objekterkennung eingeführt. Dabei wird insbesondere auf deren Erzeugung mit Hilfe eines 3D-Laserscanners eingegangen und darauf, welche Eigenschaften die erzeugten Bilder besitzen.
- Kapitel 4: Kapitel 4 beschreibt den Bereich der Konturextraktion, der aus Bildvorverarbeitung und Bildsegmentierung besteht. Es werden verschiedene Techniken vorgestellt, um Objektkonturen aus dem Tiefenbild zu extrahieren. Außerdem werden der Konturextraktion vorgeschaltete Interpretationen des Tiefenbildes beschrieben, die es ermöglichen, den Boden und Grauwertsprünge im Tiefenbild zu erkennen.
- Kapitel 5: Im fünften Kapitel werden verschiedene Kontur-Repräsentationstechniken und konturbeschreibende Merkmale vorgestellt. Es werden einfache geometrische Merkmale, Hu- und Zernike-Momente, die Curvature-Scale-Space-Repräsentation, die Angular-Radial-Transformation, Fourier-Deskriptoren und das Border-Signature-Verfahren beschrieben.
- Kapitel 6: Die in Kapitel 5 ermittelten Merkmale werden zu Vektoren zusammengefasst. In diesem Kapitel wird beschrieben, welche Verfahren die Klassifikation dieser Merkmalsvektoren ermöglicht. Es wird auf die Reduktion des Merkmalsvektorraums eingegangen. Die Klassifikationsstrategien der überwachten und unüberwachten Klassifikation werden erläutert. Insbesondere wird auf das in dieser Arbeit verwendete überwachte Klassifikationsverfahren der Support-Vektor-Maschine (SVM) eingegangen.
- Kapitel 7: Dieses Kapitel befasst sich mit der Performanz-Analyse des Klassifikators. Es werden die verschiedenen Merkmalsextraktionsmethoden miteinander verglichen und bestimmt mit welchem Kernel die SVMs die jeweils beste Performanz liefern. Dazu wird die Receiver-Operating-Characteristics-Analyse eingesetzt.
- Kapitel 8: Das letzte Kapitel liefert die Zusammenfassung der Arbeit. Es wird beschrieben, welche Methoden in wie weit in der Lage sind, Objekterkennung zu ermöglichen, wo Grenzen und Probleme des verwendeten Ansatzes liegen und welche Verbesserungen in weiterführenden Arbeiten vorgenommen werden können.

### 1.3 Wissenschaftlicher Beitrag

Objekterkennung spielt im Bereich der autonomen mobilen Robotik eine große Rolle. Der Roboter muss für abstraktere Aufgaben wie Interpretation der vor ihm liegenden Szene und darauf basierende Handlungsplanung wissen, welche Objekte sich in der Szene befinden. In dieser Arbeit wird die Objekterkennung über eine konturorientierte Klassifikation der Tiefenbilder des auf dem Roboter installierten Laserscanners realisiert. Der wissenschaftliche Beitrag dieser Arbeit liegt in dem Gesamtpaket, das von Tiefenbildgenerierung über Segmentierung bis zur Klassifikation und Performanzevaluation reicht. Es werden verschiedene konturbeschreibende Merkmale zur Klassifikation eingesetzt. Unter anderem werden Merkmale mit Hilfe des Eigenraums der Curvature-Scale-Space-Repräsentation einer Kontur erstellt. Dieses Verfahren wurde bis jetzt nur auf künstlich generierten Daten angewendet. In dieser Arbeit wird es auf reale Daten angewendet und ein Vergleich mit anderen formbeschreibenden Merkmalen gegeben.



# Kapitel 2

## Stand der Technik

Da in dieser Arbeit das gesamte Bildverarbeitungssystem von der Bildgewinnung bis zur Objektklassifikation bearbeitet wird, teilt sich der dieser Abschnitt ebenso in verschiedene Bereiche auf.

### 2.1 Tiefenbild Segmentierungstechniken

Es gibt eine Vielzahl von Tiefenbild-Segmentierungstechniken, die teilweise für Grauwertbilder entwickelt wurden und dann für Tiefenbilder übernommen worden sind. Es existieren aber auch Segmentierungsverfahren, die direkt für Tiefenbilder entwickelt wurden. Sie kombinieren unter anderem das Tiefenbild mit dem Intensitätsbild, das auch mit dem Laserscanner aufgenommen wird.

Hier sollen kurz einige der in dieser Arbeit nicht verwendeten Verfahren zur Tiefenbildsegmentierung vorgestellt werden:

**Aktive Konturen:** Aktive Konturen (auch Schlangen engl.: *snakes*) ist ein 1987 von M. Kass, A. Witkin und D. Terzopoulos entwickeltes Verfahren zur Extraktion von Konturen aus einem Grauwertbild [28]. In den folgenden Jahren wurden viele auf dieser Technik basierende Verfahren entwickelt, die es ermöglichen, Konturen auch aus Farbbildern und dreidimensionalen Daten zu extrahieren [60, 23, 36]. Das Verfahren minimiert iterativ, ausgehend von einem definiertem interessanten Bildausschnitt (engl.: *region of interest*) eine Energiefunktion. Diese ist so definiert, dass die Kontur sich in jedem Iterationsschritt immer mehr dem zu segmentierenden Objekt annähert. Sie ist zusammengesetzt aus äusserer und innerer Energie. Die äussere Energie wird minimiert, wenn die Kontur sich an Kanten im Bild anschmiegt, anstatt diese zu schneiden. Die innere Energie wird in Dehnungs- und Bindungsenergie unterteilt. Die Dehnungsenergie wird minimiert, indem sich die Kontur zusammenzieht, und die Bindungsenergie wird umso geringer, je glatter die Kontur ist. In jedem Iterationsschritt wird für jede Energieform ein Vektor bestimmt, der in die energieminimierende Richtung zeigt. Ein Problem stellt nun dar, diese unterschiedlichen Richtungsvektoren so zu gewichten, dass die Gesamtbewegungsrichtung die Kontur dichter an das Objekt heranzführt.

**Wasserscheiden-Transformation:** Die Wasserscheiden-Transformation arbeitet nicht direkt auf dem Grauwertbild, sondern auf dem Gradientenbild des Grauwertbildes. Dieses kann als Relief des

Grauwertbildes verstanden werden, wenn die Stärke des Gradienten als Höhenwert interpretiert wird. Je stärker der Grauwertgradient im Originalbild ist, desto höher ist die Erhebung im Relief. In dieses wird nun bildlich gesehen in jedes lokale Minimum ein Loch gebohrt. Die Segmentierung des Bildes erfolgt so, als ob das Bild in einen mit Wasser gefüllten Behälter getaucht wird und das Wasser langsam durch die Löcher hineinströmt. Dies führt dazu, dass sich im Bild überflutete Bereiche ausbilden (Staubecken), die durch das Relief voneinander getrennt sind (Wasserscheiden). Ist der Wasserstand höher als die Wasserscheide zwischen zwei Staubecken, verschmelzen diese zu einem größeren Staubecken. Somit wird das Bild, in Abhängigkeit von der Wasserhöhe, in voneinander getrennte Bereiche (Staubecken) segmentiert.

Es gibt spezielle Erweiterungen des Verfahrens, um Tiefenbilder zu segmentieren. So wird in [71] das Relief aus dem Gradientenbild des Tiefenbildes, dem Gradientenbild der Oberflächennormalen und dem Gradientenbild des Intensitätsbildes kombiniert. In [2] wird das Verfahren mit den in Abschnitt 4.1.3 beschriebenen morphologischen Operatoren kombiniert.

USF: Dieser und die folgenden drei Segmentierungsalgorithmen sind 1996 im Rahmen eines Projektes zum Vergleich von Tiefenbild-Segmentierungsalgorithmen von vier Universitäten entwickelt worden [19]. Es wird zum ersten Mal eine Vergleichsmetrik für Segmentierungsverfahren entwickelt und auf die vier Algorithmen angewendet, so dass diese auch bei aktuellen Segmentierungsalgorithmen immer noch als Vergleichsstandart dienen [25, 65].

Der USF-Algorithmus (University of South Florida) basiert auf einem Regionen-Wachstumsalgorithmus (engl.: *region merging*) ähnlich der oben beschriebenen Wasserscheiden-Transformation. Zunächst wird an jedes Pixel des Tiefenbildes unter Berücksichtigung einer definierten Anzahl von Umgebungspixeln eine Fläche angepasst. Dabei werden Ausreißer, deren Entfernungen sich stark von denen ihrer Umgebungspixel unterscheiden, ignoriert. Anschließend werden Pixel, die eine ähnliche Flächenanpassung besitzen, vereinigt.

WSU: Der WSU-Algorithmus (Washington State University) bestimmt zunächst Kanten im Tiefenbild, indem die Tiefenkoordinate des aktuellen Pixels mit der seiner Umgebungspixel verglichen wird. Anschließend wird für jedes Pixel, in dessen Umgebung definierter Größe keine Kantenpixel liegen, dessen Oberflächennormale berechnet. Die Richtungskomponenten der Normalen zusammen mit den Koordinaten des Pixels bilden einen Vektor im 6-dimensionalen Raum. Dort wird ein Clustering-Algorithmus angewendet, der die Vektoren in eine vorgegebene Anzahl von Cluster aufteilt. Das Tiefenbild wird segmentiert, indem jedes Pixel dem nächstgelegenen Clustermittelpunkt zugeordnet wird. Anschließend werden Segmente zusammengefasst, deren gemeinsame Grenzpixel Oberflächennormalen besitzen, deren Differenz unter einem definierten Schwellwert liegt. Des Weiteren darf der Tiefenunterschied einen definierten Schwellwert nicht überschreiten. Zum Schluss werden die als Kantenpixel markierten Pixel demjenigen an sie angrenzenden Segment zugeordnet, mit dem die Kantenpixelparameter am besten übereinstimmen.

UB: Der UB-Algorithmus (Universität Bern) basiert darauf, dass ebene Flächen in der gescannten Szene gerade Linien in einer Scanebene bilden. Aus diesem Grund werden die Scanebenen (Zeilen des Tiefenbildes) zerteilt und durch gerade Linien approximiert. Anschließend wird ein Regionen-Wachstumsalgorithmus verwendet, der Liniensegmente zu Flächen gruppiert. Als Ausgangspunkte des Verfahrens werden solche verwendet, an denen sich mindestens drei benachbarte Linien in der selben Tiefe befinden und sich zu einem definierten Prozentsatz

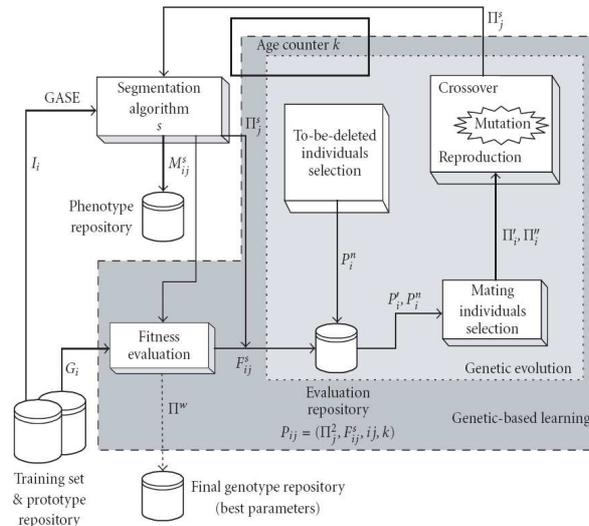
überlappen. Die längste der drei Linien wird als Startpunkt gewählt. Diese wird solange um ihre benachbarten Linien erweitert, bis diese eine definierte, von der Größe der Region abhängige, Bedingung nicht mehr erfüllen. Danach wird zur nächst besten Startlinie übergegangen.

UE: Der UE-Algorithmus (University of Edinburgh) geht zunächst analog zum USF-Algorithmus vor und erstellt für jedes Pixel seine Flächennormale. Es werden Tiefen- und Normalendiscontinuitäten einzelner Pixel ermittelt und durch Filter ausgeglichen. Anschließend wird für jedes Pixel die gaußsche und die mittlere Krümmung bestimmt. Abhängig von diesen beiden Werten wird jedem Pixel ein Flächentyp (eben, elliptisch,...) zugeordnet. Das Pixel kann auch mit „Unbekannter Flächentyp“ markiert werden. Besitzen mindestens zwei Pixel 8er Nachbarschaft und den selben Flächentyp, werden sie als Startpunkt für den folgenden Regionen-Wachstumsalgorithmus verwendet. Lücken, die durch unbekannte Flächentypen entstehen, werden mit Hilfe der in Abschnitt 4.1.3 beschriebenen morphologischen Operatoren geschlossen. Für jede Region um einen Startpunkt wird wieder eine Flächenanpassung ausgeführt. Der Regionen-Wachstumsprozess wird ausgehend von den Startpunkten durch iterative Anwendung von Ausdehnung, neuer Flächenanpassung der Region und Zusammenziehen erreicht. Die Übereinstimmung der neuen Flächenanpassung mit dem ursprünglichen Flächentyp liefert Kriterien, anhand derer entschieden wird, ob Nachbarpixel zur wachsenden Region im Ausdehnungsprozeß hinzugefügt bzw. von dieser beim Zusammenziehen entfernt werden. Zum Schluss werden die Regionengrenzen des erhaltenen segmentierten Tiefenbildes noch einmal überarbeitet, indem der Regionen-Wachstumsalgorithmus mit anderen Kriterien für das Ausdehnen und Zusammenziehen wiederholt wird.

Da jedes Segmentierungsverfahren auf vorgegebene Parameter angewiesen ist, existieren verschiedene Verfahren, diese Parameter zu optimieren. So werden z.B. im GASE-Projekt (genetic algorithm segmentation environment) genetische Algorithmen eingesetzt, um die Parameter unterschiedlicher Tiefenbild-Segmentierungstechniken zu optimieren [48]. Zum Vergleich der unterschiedlichen Segmentierungsergebnisse wird die in [19] vorgeschlagene Performanz-Metrik verwendet. Abbildung 2.1 zeigt den schematischen Verlauf der Parameteroptimierung.

## 2.2 Objekterkennung

Im Folgenden werden verschiedene Ansätze erläutert, Objekte in zweidimensionalen Bildern zu finden. Obwohl über die Tiefenbilder eine dreidimensionale Information der zu klassifizierenden Objekte vorliegt, verfolgt diese Arbeit den Ansatz, über die ansichtsvariante zweidimensionale Kontur ein dreidimensionales Objekt zu erkennen. Alternative dreidimensionale Herangehensweisen verwenden zum Beispiel ein dreidimensionales Modell des zu klassifizierenden Objektes in einer objektzentrischen Betrachtungsweise und versuchen, dieses Modell in den Tiefendaten wiederzufinden [27]. Dieser dreidimensionale Ansatz verfolgt die Philosophie, ein Objekt möglichst ansichtsinvariant zu beschreiben, wohingegen diese Arbeit durch die Verwendung von zwei Dimensionen eine ansichtsbasierte Objekterkennung verfolgt. Dabei wird jede Ansicht eines Objektes als selbständiges Unterobjekt betrachtet und kann einzeln erkannt werden. Diese Vorgehensweise hat gegenüber dem ansichtsinvarianten Ansatz den Vorteil, dass sich für einzelne Ansichten eines dreidimensionalen Objektes einfacher Merkmale finden lassen, die diese Ansicht robust beschreiben und gegen andere Objekte abgrenzen.



**Abbildung 2.1:** Schematische Darstellung der Parameteroptimierung für Segmentierungsverfahren mittels genetischer Algorithmen [48]

### 2.2.1 Template Matching

Ein alternativer Ansatz zum Vorgehen in dieser Arbeit ist das Template Matching. Es verfolgt die Strategie, ein bekanntes Objekt im Bild zu suchen und nicht, wie es in dieser Arbeit der Fall ist, das Bild zu segmentieren und die erhaltenen Objekte Klassen zuzuordnen.

Beim Template Matching wird eine Schablone (engl.: *template*) des zu erkennenden Objekts über das Bild geschoben, und es wird bestimmt, in wie weit das unter der Schablone befindliche Gebiet mit dem zu suchenden Objekt übereinstimmt. Dies wird so lange wiederholt, bis die Schablone über allen Positionen des Bildes in allen möglichen Orientierungen gelegen hat. Da dieses Verfahren aufwendig ist, kann es nur für eine geringe Anzahl von zu erkennenden Objekten durchgeführt werden [56]. Das Verfahren wird vor allem zur Materialprüfung eingesetzt, da dort Bedingungen geschaffen werden können, die eine effiziente Objektklassifikation über Template Matching ermöglichen.

### 2.2.2 Principal-Component-Analysis (PCA)

Die PCA ist ein multivariates Verfahren der Statistik zur Analyse von Datensätzen. Sie dient dazu, aus einem Datensatz die relevante Information zu extrahieren und durch einen kleineren Datensatz darzustellen. Dazu werden korrelierte Variablen des ursprünglichen Datensatzes durch nichtkorrelierte neue Variablen ersetzt. Dies geschieht, indem der Datensatz auf neue, an den Datensatz angepasste Basisvektoren (Eigenvektoren der Kovarianzmatrix) abgebildet wird. Die neuen Basisvektoren spannen einen an den jeweilige Datensatz angepassten Eigenraum auf. Dennoch geht durch das Reduzieren der Variablen immer Information verloren. Wie groß die Information einer Variable ist, wird durch die Varianz beschrieben. Die PCA verkleinert den Datensatz so, dass die Varianz des neuen Datensatzes möglichst groß wird. Zur Objekterkennung werden die Bilder in den Eigenraum projiziert. Der Abstand der projizierten Bilder im Eigenraum macht eine Aussage über die Ähnlichkeit der Bilder [16].

### 2.2.3 Klassifikation von Merkmalen

Der in dieser Arbeit verwendete Ansatz ist die Objekterkennung durch Vergleich von Merkmalen, die von den einzelnen Objekten extrahiert werden. Diese Merkmale können je nach verwendetem Sensor aus der Form, Farbe oder Textur der Objekte gewonnen werden. In dieser Arbeit werden eine Auswahl von Formmerkmalen verwendet, da in einem Tiefenbild die Farbe (bzw. Grauwert) keine Information über ein Objekt enthält und die Textur der Objekte nicht erfasst wird. Da die Objekterkennung über die Form der Objekte schon seit Mitte des letzten Jahrhunderts ein intensives Forschungsgebiet ist, existieren eine Vielzahl von Möglichkeiten, die Form der Objekte zu repräsentieren und Objekterkennung ermöglichende Merkmale zu extrahieren. Hier sollen kurz einige, der in dieser Arbeit nicht verwendeten Verfahren, vorgestellt werden [3]:

**Shape-Context.** Das Shape-Kontext-Verfahren geht davon aus, dass das Objekt in Form von Randpunkten vorliegt. Über diese Randpunkte wird ein logarithmisch polares Gitter gelegt. In diesem Gitter wird bestimmt, wie viele Randpunkte in welcher Zelle liegen. Der Anzahl der Randpunkte wird ein Grauwert zugewiesen (viele Punkte pro Zelle = dunkel, wenig Punkte = hell). Die Grauwerte werden in ein  $\Theta - \log r$  Histogramm eingetragen. Objekte werden nun klassifiziert, indem der Abstand zwischen verschiedenen Histogrammen mit einem zweiteiligen Vergleichsverfahren (engl.: *bipartite matching*) bestimmt wird [4].

Zu diesem Verfahren existieren verschiedene verwandte Verfahren wie „Grid-method“ und „Shape-Matrix“. „Grid-method“ verwendet im Gegensatz zum „Shape-Context“ ein rechteckiges Gitter und weist den einzelnen Zellen binäre Werte zu. Die „Shape-Matrix“ verwendet als festen Ursprung des Gitters den Schwerpunkt des Objekts.

**1D-Konturrepräsentationen von 2D-Konturen.** Es gibt verschiedene Möglichkeiten, die zweidimensionale Kontur auf eine reale oder komplexe eindimensionale Funktion abzubilden. Eine eindimensionale Möglichkeit, die Kontur zu repräsentieren, ist es, die Tangente der Kontur in jedem Punkt zu berechnen und diese über der Bogenlänge der Kontur aufzutragen (engl.: *angle vs. arc length function*). Analog wird beim Ansatz der intrinsischen Gleichung (engl.: *intrinsic equation*) anstatt der Tangente die Krümmung jedes Konturpunktes über der Bogenlänge aufzutragen. Eine weitere Möglichkeit besteht darin, den Schwerpunkt der Kontur zu berechnen und jeden Punkt der Kontur durch den Abstand zu repräsentieren, den dieser zum Schwerpunkt besitzt (engl.: *centroid-to-boundary distance approach*). Ein weiterer, mit der Schwerpunktmethode verwandter Ansatz bestimmt nicht den Abstand jedes Konturpunktes zum Schwerpunkt, sondern nur den von ausgezeichneten Merkmalspunkten, die sich durch eine starke Krümmung der Kontur auszeichnen [30].

Des Weiteren wird in [33] ein merkmalsbasierter Ansatz zur Objekterkennung in bereits segmentierten Tiefenbildern vorgestellt. In jedem Tiefenbild ist jeweils nur ein Objekt vor einem schwarzen Hintergrund zu sehen. Es werden Histogramme der Intensitäten, Oberflächennormalen und Oberflächenkrümmungen erstellt und als Merkmale zur Objekterkennung eingesetzt. Dieser Ansatz lässt sich in dieser Arbeit nicht realisieren, da keine segmentierten Tiefenbilder vorliegen. Ferner wird die Segmentierung des Tiefenbildes in dieser Arbeit über die Intensität der Pixel gesteuert, so dass die einzelnen Segmente mehr oder weniger die selbe Intensität besitzen und ein Intensitätshistogramm nicht sehr aussagekräftig wäre.



# Kapitel 3

## Tiefenbilder

In dieser Arbeit werden ausschließlich Tiefenbilder zur Objekterkennung eingesetzt. Die Tiefenbilder werden mit einem Laserscanner aufgenommen. Dieser befindet sich auf dem mobilen Roboter Kurt3D. Der verwendete Laser ist vom Typ Sick LMS 200 und besitzt eine maximale Auflösung von 10 mm und eine Reichweite von 80 m.

### 3.1 Erzeugung von Tiefenbildern

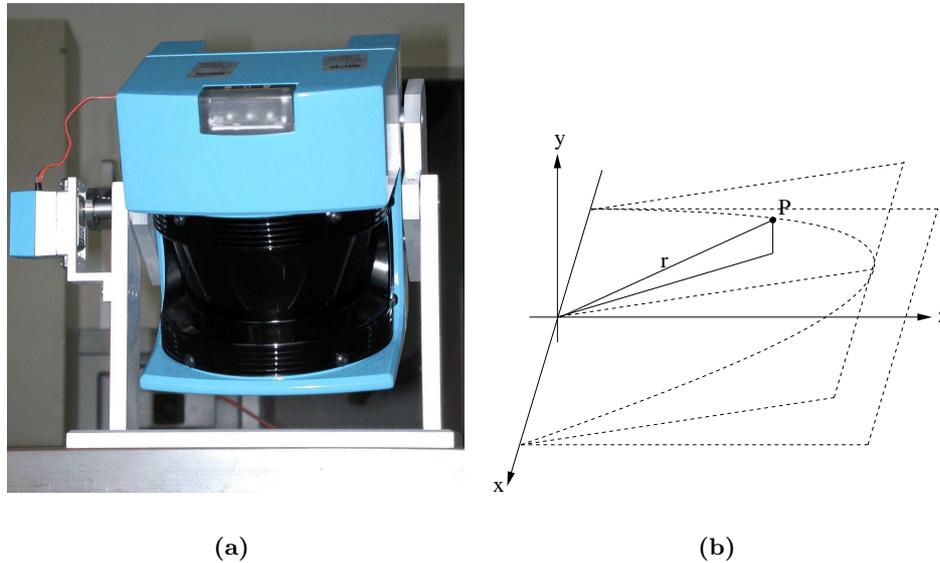
Der verwendete Scanner arbeitet nach dem Prinzip der Pulslaufzeitmessung. Dabei sendet der Laserscanner einen gepulsten Laserstrahl aus. Dieser trifft auf einen Gegenstand in der gescannten Szene und wird von diesem reflektiert. Der Laserstrahl gelangt zurück zum Scanner und wird registriert. Über die Zeit, die vom Aussenden des Signals bis zum Registrieren verstrichen ist, lässt sich über Gleichung 3.1 direkt die Entfernung  $R$  des Objektes zum Scanner berechnen.

$$R = \frac{vt}{2} \quad (3.1)$$

Da der Laserstrahl eine Geschwindigkeit von  $v \approx 3 \cdot 10^8 \frac{m}{s}$  besitzt, sind an die Zeitauflösung hohe Ansprüche gestellt. Eine Tiefenauflösung von einem Zentimeter benötigt eine Zeitauflösung von 67 Pikosekunden. Diese Genauigkeit wird durch Amplitudenmodulation erreicht. Dabei wird die Phasenverschiebung des Detektorsignals relativ zum emittierten Signal ermittelt. Da die Phasenverschiebung nur modulo  $2\pi$  bestimmt werden kann, wird das emittierte Signal mit einer Modulationsfrequenz überlagert, um die Entfernung eindeutig bestimmen zu können.

Das Tiefenbild einer dreidimensionalen Szene entsteht dadurch, dass der Laserscanner mittels eines im Scanner befindlichen Drehspiegels den Laserstrahl ablenkt und so einzelne Punkte in der zu scannenden Szene in einer horizontalen Ebene fächerförmig abtastet. Danach wird der Scanner um eine von der Auflösung abhängige Gradzahl gekippt, und ein weiterer fächerförmiger Scan wird aufgenommen. Durch Wiederholen dieser Prozedur wird erreicht, dass über die gesamte zu scannende Szene ein Raster von Scanpunkten gelegt wird. Der verwendete 2D-Laserscanner mit Kippvorrichtung ist in Abbildung 3.1a zu sehen.

Die so ermittelten Punkte im dreidimensionalen Raum müssen nun als Grauwertbild abgebildet werden. Dazu werden den Punkten in Abhängigkeit ihrer Entfernung zum Scanner Grauwerte zugeordnet. Das Koordinatensystem in das die 3D-Datenpunkte abgebildet werden ist in Abbildung 3.1b



**Abbildung 3.1:** (a) 2D-Laserscanner mit Kippvorrichtung, um 3D-Scans aufzunehmen (b) Koordinatensystem des 3D-Laserscanners

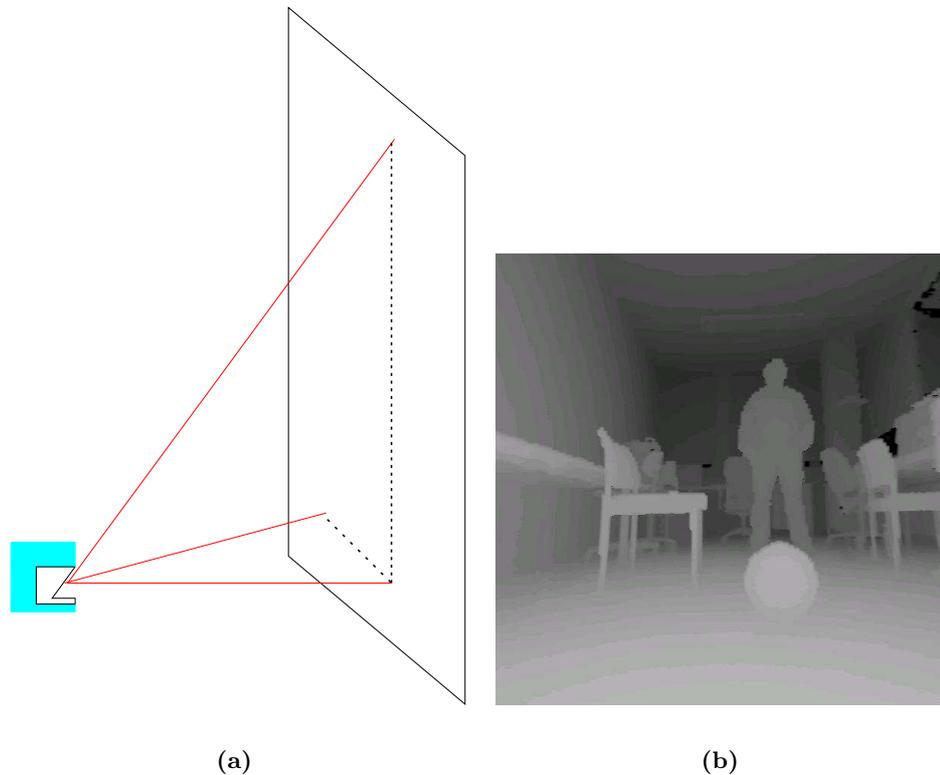
zu sehen. Folgende Gleichung berechnet den Abstand eines 3D-Datenpunktes zum Scanner und bildet ihn auf das Intervall  $[0, 1]$  ab:

$$d = 1 - \sqrt{\frac{\sqrt{x^2 + y^2 + z^2}}{\text{maximale Distanz}}} \quad (3.2)$$

Die Variable „maximale Distanz“ wird verwendet, um den Kontrast des Tiefenbildes zu steuern (vgl. 4.1.1). In dieser Gleichung wird der euklidische Abstand des Objektes zum Scanner verwendet. Dies ist jedoch nicht optimal, da dadurch ein großes Objekt (auch wenn es eine planare Fläche besitzt und senkrecht zum Scanner steht) unterschiedliche Grauwerte erhält, da der Objektpunkt in Scannerhöhe eine geringere Entfernung besitzt, als ein Objektpunkt der sich oberhalb oder seitlich vom Scanner befindet. Abbildung 3.2a macht diesen Zusammenhang deutlich. Da in dieser Arbeit Objekte als zusammenhängende Pixel mit einem nahezu identischen Grauwert definiert sind und die Segmentierungsverfahren darauf hin ausgerichtet sind (vgl. 4.2), muss Gleichung 3.2 dahingehend verändert werden, dass nur die Tiefenkoordinate berücksichtigt wird:

$$d = 1 - \sqrt{\frac{z}{\text{maximale Distanz}}} \cdot$$

Da die Auflösung des Tiefenbildes oft größer ist, als die Anzahl der 3D-Datenpunkte, wird zwischen den Grauwerten der einzelnen Datenpunkte interpoliert, um das gesamte Tiefenbild zu erhalten. Diese Eigenschaft wird in Abschnitt 4.4.3 zur Segmentierung des Tiefenbildes ausgenutzt. Abbildung 3.2b zeigt das auf diese Weise erhaltene Tiefenbild.



**Abbildung 3.2:** (a) Seitenansicht eines Laserscans. (b) Durch Mapping der 3D-Datenpunkte auf Grauwerte und der Interpolation zwischen diesen entstandenes Tiefenbild.

## 3.2 Eigenschaften von Tiefenbildern

Im Gegensatz zu herkömmlichen Kamerabildern, in denen die Lichtintensität gespeichert ist, repräsentieren die in Tiefenbildern enthaltenen Grauwerte den Abstand von Objektoberflächen zum Scanner. Durch diese Eigenschaft ist das Tiefenbild unempfindlich gegenüber Beleuchtungsänderungen, Schattenwurf und Verschmutzung von Objektoberflächen. Tiefendaten bieten über die bekannte Entfernung von Objekten zum Scanner auch die Möglichkeit die reale Objektgröße zu bestimmen. Diese kann als Verifikation bei der Objekterkennung dienen. Da die Kontur eines Objektes, das sich von seinem Hintergrund abhebt im Tiefenbild einen Grauwertsprung verursacht, ist die Segmentierung einfacher als in normalen Grauwertbildern. Normale Grauwertbilder verwenden die Intensität des vom Objekt reflektierten Lichtes und segmentieren somit unterschiedlich reflektierende Bereiche des Objektes.

Ein Nachteil der Tiefenbilder ist der, dass die Extraktion der Merkmale, die zur Objekterkennung dienen, nur mit der Form der Objekte arbeiten kann, da Eigenschaften wie Textur und Farbe der Objekte in Tiefenbildern nicht gespeichert sind. Der Grauwert selbst besitzt auch keine Information über das Objekt, da es sich in beliebiger Entfernung zum Scanner befinden und somit einen beliebigen Grauwert annehmen kann (vgl. Kapitel 5).



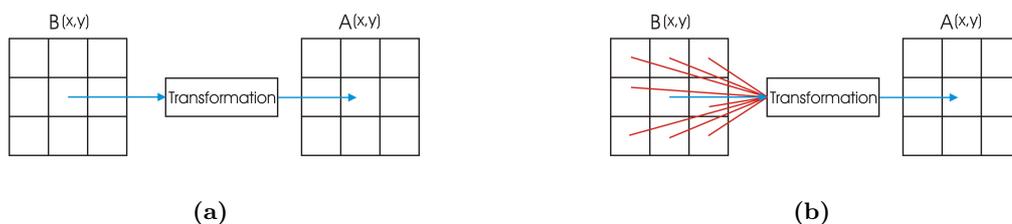
# Kapitel 4

## Konturextraktion

Die Kontur eines Objektes bezeichnet dessen Grenze zu anderen Objekten oder zum Hintergrund. Sie variiert mit dem Blickwinkel unter dem das Objekt gesehen wird. Damit ist die Objektklassifikation über Konturen eine ansichtsbasierte Objekterkennung. Im Gegensatz zur ansichtsinvarianten Objekterkennung wird hier ein dreidimensionales Objekt durch verschiedene Ansichten repräsentiert. Die Kontur bzw. Form gehört neben der Farbe und der Textur eines Objektes zu den Standardinformationsträgern im Bereich des maschinellen Sehens. In diesem Kapitel wird beschrieben, wie die Konturen aus einem Tiefenbild gewonnen werden.

Die Konturextraktion teilt sich in eine Bildvorverarbeitungsphase und eine Segmentierungsphase auf. Beide Verarbeitungsschritte sind in dieser Arbeit in einem Kapitel zusammengefasst, da der Übergang zwischen ihnen fließend ist. So dienen die in 4.1.3 beschriebenen Verfahren sowohl zur Glättung als auch zur Segmentierung des Tiefenbildes.

Bei den zur Bildverarbeitung eingesetzten Operationen wird zwischen Bildpunktoperationen und lokalen Operationen unterschieden. Abbildung 4.1 macht den Unterschied zwischen beiden Kategorien deutlich. Bei einer Bildpunktoperation geht in die Berechnung jedes Bildpixels des verarbeiteten Bildes  $A(x, y)$  jeweils nur das entsprechende Pixel des Originalbildes  $B(x, y)$  ein. Bei lokalen Operationen werden zusätzlich noch Umgebungspixel des gerade verarbeiteten Pixels betrachtet. Werden lokale Operationen iterativ auf jedes Pixel angewendet, spricht man von Filtern. Diese werden überwiegend zur Glättung und Kantenextraktion eingesetzt. Neben Bildpunkt- und lokalen Operationen findet noch eine Unterteilung in lineare und nichtlineare Operationen statt.



**Abbildung 4.1:** Unterschied zwischen einer (a) Bildpunkt- und einer (b) lokalen Operationen.

## 4.1 Bildvorverarbeitung

Vor der Segmentierung des Bildes wird dieses oft mehrfach vorverarbeitet. Mit dem Ziel, das Segmentieren des Bildes zu erleichtern, ohne dabei relevante Informationen des Bildes zu verändern. Es gibt keine Methoden, die für die Bildvorverarbeitung generell optimal sind. Alle Maßnahmen müssen entsprechend den Anforderungen an das verarbeitete Bild und den experimentellen Gegebenheiten zur Bildgewinnung gewählt werden. Die hier vorgenommenen Bildvorverarbeitungsmaßnahmen sind Kontrastverbesserung und Rauschunterdrückung bzw. Ausgleichen von Digitalisierungsfehlern durch Glättung des Tiefenbildes.

### 4.1.1 Kontrastverbesserung

Kontrast bezeichnet den Übergang von einem Grauwert zu einem anderen in einem begrenzten Bereich des Bildes. Je schneller die Grauwertänderung erfolgt, desto größer ist der Kontrast.

Die Kontrastverbesserung wird eingesetzt, damit sich Objekte im Bild stärker voneinander abheben und somit leichter zu segmentieren sind.

Es gibt verschiedene Ansätze, den Kontrast eines Bildes zu erhöhen. In dieser Arbeit wird das Verfahren der linearen Grauwertdehnung angewendet. Es gehört zur Gruppe der linearen Bildpunktoperationen. Diese modifizieren mittels Look-Up-Tables (LUT) die Grauwertskala eines Bildes.

Der Kontrast eines Grauwertbildes  $B(x, y)$  kann mit diesem Verfahren verbessert werden, wenn die im Bild vorkommenden minimalen und maximalen Grauwerte  $B_{min}$  und  $B_{max}$  nicht den gesamten zur Verfügung stehenden Grauwertbereich  $A_{min}$  bis  $A_{max}$  abdecken. In diesem Fall lässt sich nach Gleichung 4.1 das Bild  $B(x, y)$  in ein Bild  $A(x, y)$  mit höherem Kontrast transformieren [1]:

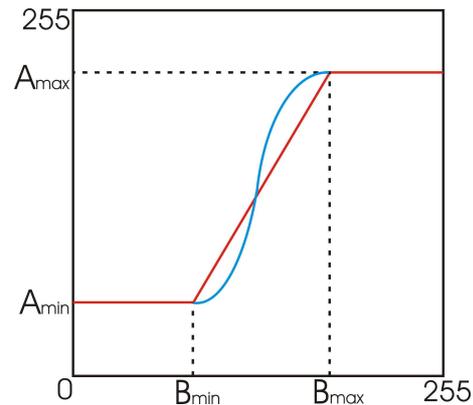
$$A(x, y) = \frac{A_{max} - A_{min}}{B_{max} - B_{min}} \cdot (B(x, y) - B_{min}) + A_{min} . \quad (4.1)$$

Die Transformation von  $B(x, y)$  nach  $A(x, y)$  ist eine lineare, ortsunabhängige und globale Grauwerttransformation [56]. Es sind jedoch auch nichtlineare Transformationen vorstellbar. Eine nichtlineare Transformation ermöglicht es, den Kontrast zwischen Objekten, die sich zu nah am Scanner befinden oder zu weit entfernt sind, herabzusenken. Dadurch steht im relevanten Entfernungsbereich ein größerer Grauwertbereich zur Verfügung, so dass der Kontrast dort noch stärker erhöht werden kann. Abbildung 4.2 stellt die LUT-Kennlinien einer linearen bzw. nichtlinearen Transformation dar.

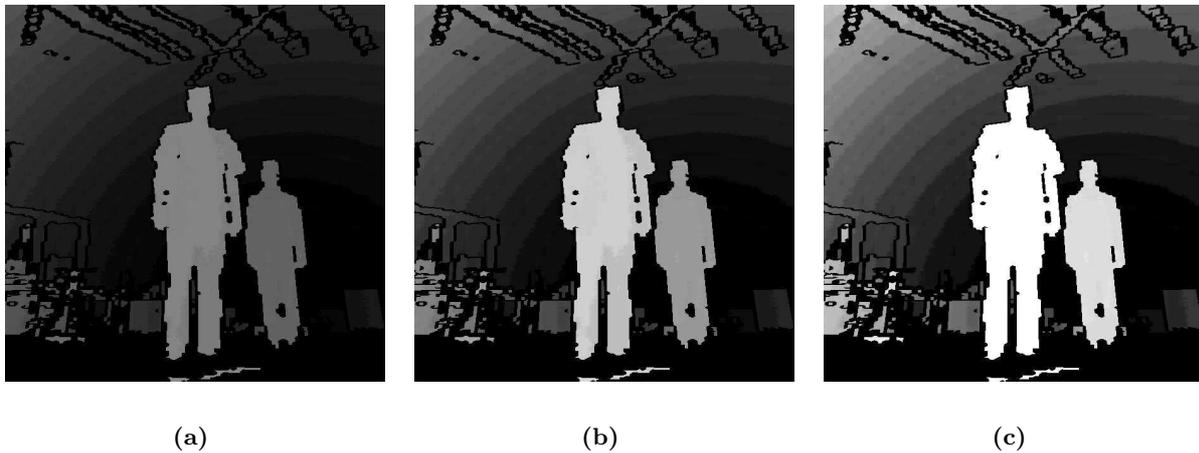
**Clipping:** Das Clipping ist neben den nicht linearen Kennlinien ein weiteres Verfahren, um den Kontrast im relevanten Bereich zu verstärken. Dabei wird anstatt des Intervalls  $[B_{min}, B_{max}]$  ein kleineres Intervall  $[B'_{min}, B'_{max}]$  mit  $B_{min} \leq B'_{min} \leq B'_{max} \leq B_{max}$  abgebildet [1]. Abbildung 4.3 macht deutlich, dass insbesondere für Tiefenbilder nach der Entfernung von Bodenpunkten (deren Erzeugung in 4.4.2 beschrieben ist) das Abschneiden der hellsten Grauwerte durch Clipping Sinn macht, da diese meist nur von wenigen falsch klassifizierten Bodenpunkten besetzt sind.

### 4.1.2 Glättung durch lokale Operatoren

Wie Abbildung 4.4a zeigt, kann das Tiefenbild durch Digitalisierungsfehler einzelne fehlende Pixel oder fehlende horizontale Linien enthalten. Neben diesen Digitalisierungsfehlern sind Tiefenbilder



**Abbildung 4.2:** Look-Up-Table-Kennlinien einer linearen bzw. nichtlinearen Grauwertdehnung.



(a)

(b)

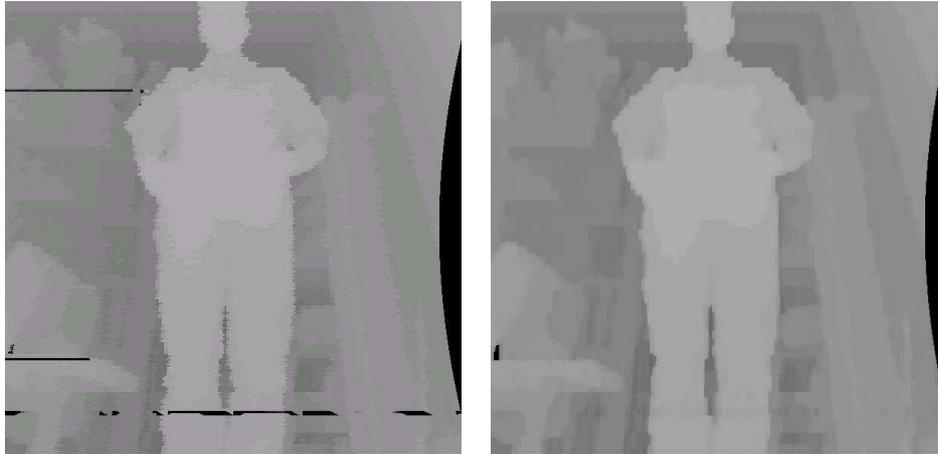
(c)

**Abbildung 4.3:** Kontrasterhöhung des Tiefenbildes: (a) Tiefenbild ohne Kontrasterhöhung, (b) Kontrasterhöhung durch Grauwertdehnung, (c) Kontrasterhöhung durch Grauwertdehnung mit Clipping

realer Szenen im allgemeinen mit Rauschen behaftet [25]. Um diese Fehler vor der Segmentierungsphase auszugleichen, muss das Bild geglättet werden. An die Glättungsoperation ist die Anforderung gestellt, dass sie die erwähnten Störungen unterdrückt ohne die Kanten im Bild zu beeinflussen. Dazu müssen lokale Operatoren (Filter) eingesetzt werden, da ein Bildpunktoperator, der keine Information über seine Umgebung besitzt, keine glättende Funktion übernehmen kann.

### Lineare Filter

Lineare Filter sind lineare homogene Funktionen und werden auch als Faltungen bezeichnet. Jeder lineare Filter besitzt eine inverse Transformation und benötigt wenig Rechenaufwand. Sie werden vor allem zur Glättung und Kantendetektion eingesetzt. Zu den linearen Glättungsfilttern gehören der Mittelwert- und der Gaußfilter.



**Abbildung 4.4:** Originaltiefenbild (a) mit Digitalisierungsfehlern und ein mit einer „Öffnen-Schließen“ Kombination geglättetes Tiefenbild (b), in dem nahezu alle Fehler entfernt sind

**Mittelwert-Filter:** Der Mittelwert-Filter ersetzt jedes Pixel durch den Mittelwert seiner Umgebungspixel. Dabei wird jedes Umgebungspixel gleich gewichtet. Dies hat zur Folge, dass der Mittelwert-Filter nicht nur Rauschen glättet, sondern auch Kanten im Bild verwischt.

**Gauss-Filter:** Der Gauss-Filter ersetzt jedes Pixel, indem die Umgebungspixel mit einer Gaussfunktion gewichtet werden. Dadurch verwischen Kanten im Tiefenbild nicht so stark wie beim Mittelwert-Filter aber immer noch stärker als bei den im folgenden beschriebenen nichtlinearen Filtern.

### Nichtlineare Filter

Es existieren verschiedene Arten von nichtlinearen Filtern. Hier sollen ausschließlich Filter aus der Gruppe der Rangordnungsoperatoren besprochen werden. Nichtlineare Filter besitzen keine inverse Operation wie die linearen Filter und benötigen mehr Rechenzeit als diese. Der bekannteste Rangordnungsoperator ist der Median-Filter. Neben diesem Operator werden die morphologischen Rangordnungsoperatoren Erosion und Dilatation besprochen. Sie erhalten aufgrund ihrer Komplexität ein eigenes Unterkapitel.

**Median-Filter:** Der Medianfilter wurde Anfang der 70er Jahre zur nichtlinearen Glättung von Signalen vorgeschlagen [56]. Bei der Medianfilterung wird der Grauwert eines Pixels durch den Median aller Grauwerte der Umgebungspixel ersetzt. Dieser ergibt sich, indem alle Pixel der herangezogenen Umgebung in eine sortierte Folge gebracht werden und das aktuelle Pixel durch den mittleren Wert der Folge (den Median) ersetzt wird. Der Medianfilter unterdrückt Rauschen und ist gleichzeitig ein kantenerhaltender Operator.

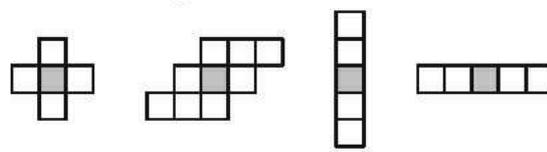


Abbildung 4.5: Verschiedene zweidimensionale Strukturelemente [21]

### 4.1.3 Glättung durch morphologische Operatoren

Zu den morphologischen Operatoren gehören Dilatation, Erosion und Skeletierung, wobei Skeletierung eine spezielle Art der Erosion ist [56]. Morphologische Verfahren können auf Binär- und Grauwertbilder (als Rangordnungsoperatoren) angewendet werden. Binäre morphologische Operatoren arbeiten mit der Form von Objekten, die sich in einem Bild befinden und verändern diese. Angewendet auf ein Grauwertbild führen morphologische Operatoren zu einer Aufhellung bzw. Verdunkelung und können zum Glätten des Bildes eingesetzt werden. In dieser Arbeit werden morphologische Operatoren sowohl auf Grauwertbilder (zum Glätten) als auch auf Binärbilder (zur Segmentierung) angewendet.

Morphologische Operationen gehören zur Klasse der nichtlinearen Filter [21] und dienen unter anderem zur Rauschunterdrückung, zum Trennen bzw. Zusammenführen von Regionen, zur Merkmalsextraktion, zur Kontrasterhöhung und zur Randerkennung.

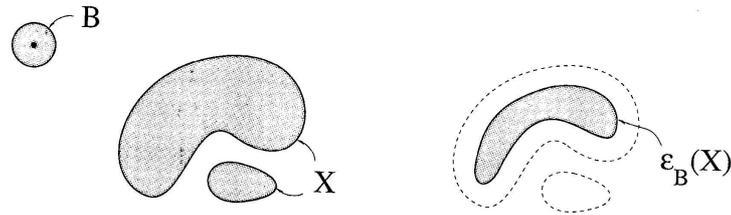
Die Operationen falten das Bild mit einem meist quadratischen Strukturelement (auch Kern genannt). Abbildung 4.5 zeigt häufig verwendete nicht quadratische Strukturelemente, die jeweils einen definierten Ankerpunkt besitzen. Die Strukturelemente besitzen unterschiedliche Formen, um eine Richtungsabhängigkeit der Operationen zu erreichen.

#### Erosion

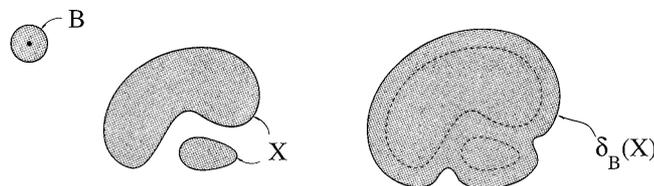
Erosion wird verwendet, um ein Objekt zu verkleinern. Bei einer Erosionsoperation wird der Ankerpunkt des Strukturelementes iterativ über jedes Pixel des zu verarbeiteten Bildes gelegt. Für jedes Pixel wird bestimmt, wieviele Pixel des Strukturelementes über Objektpixeln des zu verarbeiteten Bildes liegen. Nur wenn die Anzahl der Pixel, für die diese Bedingung gilt, einen definierten Schwellwert überschreitet, wird der Ankerpunkt als Objektpixel des verarbeiteten Bildes übernommen. Meist wird der Schwellwert auf die Gesamtgröße des Strukturelementes gesetzt, so dass alle Pixel des Strukturelementes Objektpixel sein müssen, damit der Ankerpunkt als Objektpixel markiert wird. Die Wirkungsweise der Erosionsoperation ist in Abbildung 4.6 zu sehen.

Formal lässt sich die Erosion einer Menge  $X$  mit einem Strukturelement  $B$  als Schnitt negativer Translationen des Strukturelementes darstellen [68]:

$$\varepsilon_B(X) = \bigcap_{b \in B} X_{-b} \quad \text{mit Translation} \quad X_{-b} = \{x + (-b) \mid x \in X\}.$$



**Abbildung 4.6:** Wirkungsweise der Erosionsoperation: es ist zu sehen, dass die Erosion die kleinere Komponente der Menge  $X$  entfernt, da das Strukturelement  $B$  nicht komplett in die Komponente hineinpasst [54].



**Abbildung 4.7:** Wirkungsweise der Dilatationsoperation: es ist zu sehen, dass die Dilatation die Lücke zwischen den Komponenten der Menge  $X$  schließt [54].

## Dilatation

Dilatation führt zu einem Wachstums- bzw. Ausdehnungsvorgang der Bildobjekte. Es ist möglich, verschiedene Objekte zu vereinigen, Löcher zu füllen oder Risse zu schließen. Dilatation entspricht der Erosionsoperation mit einem geringen Schwellwert. Meist wird dieser auf eins gesetzt, so dass der aktuelle Ankerpunkt des Strukturelementes als Objektpixel markiert wird, sobald ein Pixel unter dem Strukturelement ein Objektpixel im Originalbild ist:

$$\delta_B(X) = \{x \mid B_x \cap X \neq \emptyset\}.$$

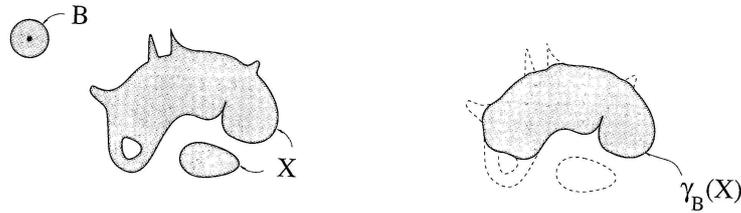
Es kann gezeigt werden, dass dies der Vereinigung von Translationen des Originalbildes mit Bezug auf das Strukturelement entspricht:

$$\delta_B(X) = \bigcup_{b \in B} X_{-b}.$$

Die Wirkungsweise der Dilatationsoperation ist in Abbildung 4.7 zu sehen. Dilatation und Erosion sind einander entgegengesetzt aber nicht invers, da es sich um keine linearen Filter handelt.

## Öffnung

Die Öffnungsoperation ist nach Gleichung 4.2 eine Kombination aus Erosion und anschließender Dilatation mit punktgespiegeltem Strukturelement  $B'$  [54]. Sie hat das Ziel, die meisten Strukturen,



**Abbildung 4.8:** Wirkungsweise der Öffnungsoperation [54].

die durch die Erosion verloren gegangen sind, wiederherzustellen (siehe Abbildung 4.8). Es kann aber immer nur ein Teil der Strukturen wiederhergestellt werden, da Objekte, die kleiner als das Strukturelement, sind bei der Erosion komplett zerstört werden. Somit besitzt die Öffnungsoperation eine von Art und Größe des Strukturelementes abhängige Filtereigenschaft:

$$\gamma_B(X) = \delta_{B'} [\varepsilon_B(X)] . \quad (4.2)$$

Die Öffnungsoperation kann ebenso wie Erosion und Dilatation geometrisch interpretiert werden. Nur wenn das Strukturelement ganz über Objektpunkten liegt, werden im Gegensatz zur Erosion nicht nur der Ankerpunkt, sondern alle Punkte des Strukturelementes übernommen. Dies führt zur formalen Beschreibung einer geöffneten Menge, die sich als Vereinigung aller Strukturelemente, die Teilmenge der Originalmenge sind, ergibt:

$$\gamma_B(X) = \bigcup \{B \mid B \subseteq X\} .$$

Die geöffnete Menge ist somit immer Teilmenge der Originalmenge (Anti-Extensivitätseigenschaft) und mehrfache Anwendung der Öffnungsoperation führt zu keiner Veränderung der bereits geöffneten Menge (Idempotenzeigenschaft) [54].

### Schließung

Der duale Operator zur morphologischen Öffnung ist die morphologische Schließung. Sie ist analog zur Öffnung definiert, nur werden Reihenfolge von Erosion und Dilatation vertauscht:

$$\phi_B(X) = \varepsilon_{B'} [\delta_B(X)] .$$

Die veränderte Reihenfolge führt dazu, dass die Operation das Objekt nicht verkleinert, sondern kleine Lücken im und zwischen Objekten schließt. Die algebraische Formulierung lautet: Passt das Strukturelement ganz in den Hintergrund  $X^c$  der zu schließenden Menge, gehören alle Punkte des Strukturelementes zum Komplement der Schließung der Menge [54]:

$$\phi_B(X) = \left[ \bigcup \{B \mid B \subseteq X^c\} \right]^c .$$

Die Schließungsoperation besitzt ebenso wie die Öffnungsoperation die Idempotenzeigenschaft. Sie erzeugt aber im Gegensatz zur Öffnungsoperation keine Untermenge der Originalmenge, sondern eine Obermenge (Extensivitätseigenschaft). Die Wirkungsweise der Schließungsoperation ist in Abbildung 4.9 zu sehen.

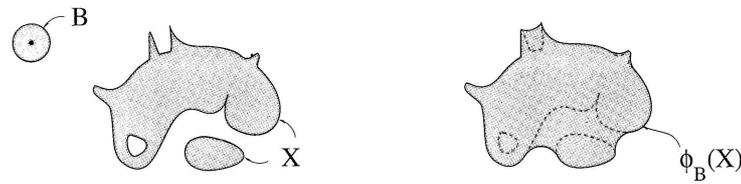


Abbildung 4.9: Wirkungsweise der Schließungsoperation[54].

### Morphologische Operatoren und Grauwertbilder

Werden Dilatation und Erosion auf Grauwertbildern angewendet, arbeiten sie als Rangordnungsoperatoren. Die Erosionsoperation ersetzt jedes Pixel durch den geringsten Grauwert, der sich im Originalbild unter dem Strukturelement befindet. Dadurch führt die Erosion zu einer Verdunkelung des Grauwertbildes [54]. Die Grauwert-Dilatation ersetzt jedes Pixel durch den größten Grauwert, der sich unter dem Strukturelement befindet, und führt dem entsprechend zu einer Aufhellung der Bildes.

Grauwert-Öffnung und -Schließung arbeiten analog zur binären Öffnung und Schließung mit dem Unterschied, dass sie Grauwert-Erosion und -Dilatation verwenden. Beide Operationen führen zu einer Glättung des Grauwertbildes. Dabei filtert die Öffnungs- bzw. Schließungsoperation Störungen oberhalb bzw. unterhalb des eigentlichen Signals heraus.

Die glättende Wirkung der morphologischen Operationen wird durch eine Kombination aus Öffnen und Schließen erreicht [45]. Dabei sind eine Hintereinanderschaltung von Öffnen und Schließen bzw. Schließen und Öffnen möglich [25]. Abbildung 4.4b zeigt ein mit einer Öffnen-Schließen Kombination geglättetes Bild. Dabei wird ein vertikales Strukturelement der Größe  $1 \times 5$  eingesetzt, da dies die meist horizontalen Digitalisierungsfehler am effektivsten geglättet. Im Vergleich mit Abbildung 4.4a ist zu sehen, dass die Glättung Fehler im Tiefenbild nahezu vollständig entfernt, ohne die Kanten des Bildes zu verwischen.

## 4.2 Segmentierungsverfahren

Die Bildsegmentierung ist der Übergang von der rein numerischen zu einer symbolischen aber nicht interpretierten Darstellung des Bildes. Sie ist somit klar von der Bildvorverarbeitung und der Mustererkennung abzugrenzen, obwohl die Segmentierung teilweise die selben Operationen wie diese verwendet [56].

Mittels Segmentierungsverfahren wird ein Bild in einzelne voneinander getrennte, für eine gewisse Eigenschaft homogene Bereiche unterteilt. Segmentierung ist ein anwendungsspezifisches Verfahren, das stark von verwendeten Sensoren und Aufnahmeprozessen abhängt. Zur Segmentierung wird in der Regel kein oder nur wenig Objektwissen eingesetzt [56]. In dieser Applikation wird das Wissen verwendet, dass die Objekte einen relativ gleichbleibenden Grauwert besitzen.

Jedes Bildauswertungsverfahren benötigt direkt oder indirekt diesen Schritt der sinnvollen Gruppierung der Bildpixel. Jedem segmentierten Objekt können im Gegensatz zu einzelnen Bildpixeln komplexere Eigenschaften wie Form, Textur usw. zugeschrieben werden. Viele High-Level-Bildverarbeitungssysteme arbeiten mit diesen Eigenschaften. Wie in Kapitel 5 zu sehen ist, wird

in dieser Arbeit nur die Eigenschaft „Form“ als Informationsträger für ein Objekt verwendet. Eine zuverlässige Segmentierung ist von zentraler Bedeutung, da sich Fehler in der Segmentierungsphase auf alle folgenden Verarbeitungsschritte auswirken. Selbst ein optimaler Klassifizierer ist nicht in der Lage, ein falsch segmentiertes Bild richtig zu klassifizieren. Ein gutes Segmentierungsverfahren besitzt die Eigenschaft, dass es möglichst große, disjunkte Segmente von zusammenhängenden Bereichen erzeugt, deren Vereinigung wieder das gesamte Bild ergibt.

Segmentierungsalgorithmen lassen sich in folgende vier Klassen unterteilen:

**punktorientierte Verfahren:** Die punktorientierten Verfahren (auch elementare Verfahren) verwenden zur Entscheidung, ob ein Pixel zu einem Segment gehört, nur die Eigenschaften des Pixels wie dessen Grauwert. Zu den punktorientierten Verfahren gehört das in 4.2.3 beschriebene Schwellwertverfahren.

**kantenorientierte Verfahren:** Kantenorientierte Verfahren versuchen zunächst mittels lokaler Operationen, Kanten im Bild zu finden. Anschließend extrahiert ein Konturverfolgungsalgorithmus (vgl. Abschnitt 4.3) die Konturen aus dem Kantenbild. Zu den am häufigsten verwendeten Kantendetektionsoperatoren gehören der Sobel-, Laplace- und der Canny-Operator. Bei Tiefenbildern wird zwischen sogenannten „Sprungkanten“ (engl.: *jump edges*) und „Faltenkanten“ (engl.: *crease edges*) unterschieden [26]. Sprungkanten zeichnen sich durch eine Diskontinuität der Tiefenkoordinate aus. Während Faltenkanten entstehen, wenn zwei Flächen unter einem Winkel aufeinander treffen. Oben genannte Kantendetektionsoperatoren sind nur in der Lage Sprungkanten im Tiefenbild zu entdecken, da bei Faltenkanten ein fließender Grauwertübergang gegeben ist. Faltenkanten werden durch den Vergleich der Oberflächennormalen der aneinanderstoßenden Flächen detektiert. Überschreitet der Winkel zwischen den Normalen einen definierten Wert, muss zwischen den Flächen eine Faltenkante liegen.

**regionenbasierte Verfahren:** Regionenbasierte Verfahren verwenden im Gegensatz zu punktorientierten Verfahren auch Umgebungspixel um zu entscheiden, ob ein Pixel zu einem Segment gehört oder nicht. Diese Verfahren werden in regionenwachstums- und baumorientierte Verfahren unterteilt. Regionenwachstumsverfahren setzen sukzessiv Regionenkeime in das Bild und weiten die Region auf die Nachbarpixel der Regionenkeime aus, wenn diese ein bestimmtes Homogenitätskriterium erfüllen. Baumorientierte Verfahren repräsentieren das Bild durch eine Baumstruktur (Pyramiden, Quad Trees), wobei ein Knoten im Baum einen homogenen Bereich repräsentiert.

**regelbasierte Verfahren:** Regelbasierte Verfahren (auch Segmentierung durch Klassifikation) besitzen im Gegensatz zu den bisher vorgestellten Verfahren einen mehrdimensionalen Merkmalsraum. Es ist somit möglich, neben der Grauwertskala weitere Merkmale wie Textureigenschaft oder Grauwertgradient zu verwenden. Mit Hilfe von Lernverfahren werden diese Merkmalsvektoren der Pixel geclustert und somit das Bild segmentiert.

### 4.2.1 Canny Kantendetektion

Da die Objekte im Tiefenbild sich von ihrem Hintergrund durch einen Grauwertsprung abheben, wird das Canny-Kantendetektionsverfahren [8] eingesetzt, um diese Objektkanten zu ermitteln. Der Canny-Operator ist eine Kombination aus verschiedenen lokalen Operationen. Zunächst wird ein

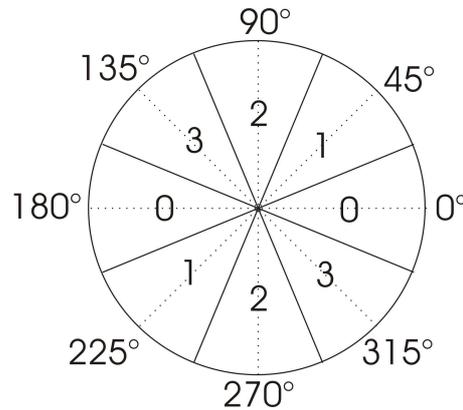


Abbildung 4.10: Schema zur Diskretisierung der Orientierung der Gradientenvektoren nach [22].

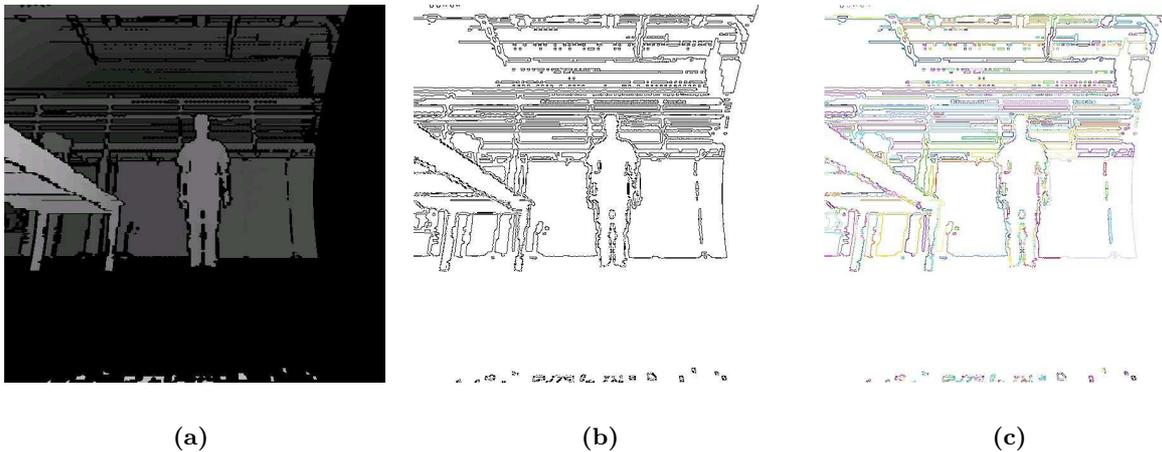
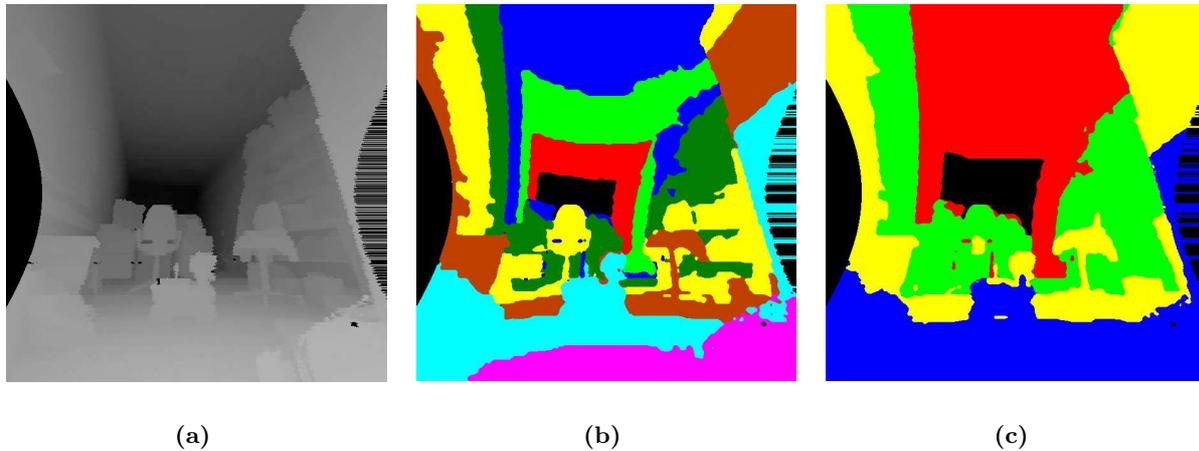


Abbildung 4.11: Experimentelles Ergebnis der Konturextraktion mittels des Canny-Operator. Abbildung (a) zeigt das Tiefenbild, auf das der Canny-Operator angewendet wird. Dadurch entsteht das Kantenbild (b). In (c) sind die aus dem Kantenbild extrahierten Konturen zu sehen, wobei zusammenhängende Konturpunkte mit der selben Farbe markiert sind.

Gauss-Filter angewendet, um Rauschen im Bild zu eliminieren, da die durch Rauschen entstandenen Pixel einen Helligkeitsunterschied zu ihren Nachbarpixeln besitzen und somit als Kante erkannt werden. Das geglättete Tiefenbild wird mit dem Sobel-Operator gefaltet. Der Sobel-Operator ist ein Kantendetektor erster Ordnung und wird in  $x$ - und  $y$ -Richtung auf das Bild angewendet, um horizontale bzw. vertikale Kanten zu extrahieren. Danach werden Größe und Orientierung der Gradientenvektoren bestimmt. Die Größe der Gradienten wird durch das geometrische Mittel der  $x$ - und  $y$ -Kantenbilder bestimmt. Die Orientierung wird nach Gleichung 4.3 ermittelt und durch das in Abbildung 4.10 dargestellte Schema diskretisiert.

$$\text{Orientierung}(x, y) = \arctan \left( \frac{y - \text{Kantenbild}(x, y)}{x - \text{Kantenbild}(x, y)} \right) \quad (4.3)$$



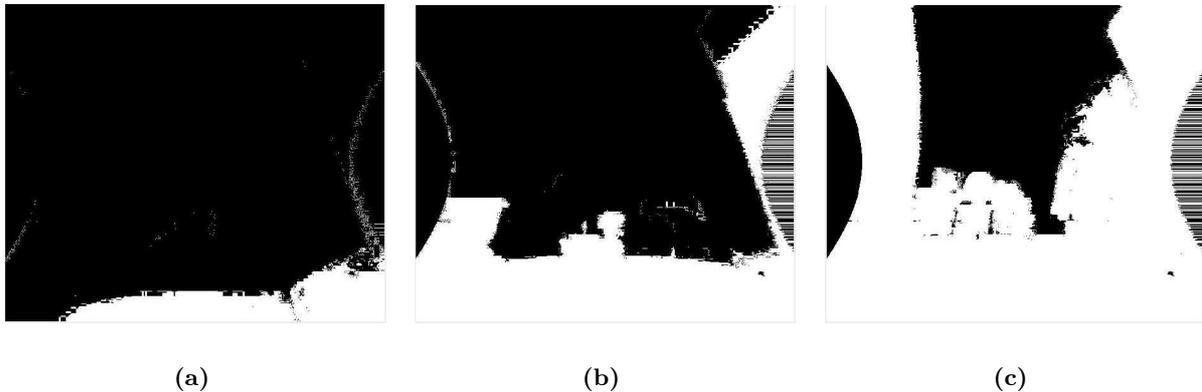
**Abbildung 4.12:** (b) stellt das mittels  $k$ -Means Verfahren mit neun Clustern segmentiertes Tiefenbild aus (a) dar. Abbildung (c) zeigt die Segmentierung des selben Tiefenbildes mit fünf Clustern.

Zur Kantendetektion wird jedes Pixel entlang der durch die Orientierung vorgegebene Richtung mit seinen Nachbarn verglichen. Besitzt das Pixel einen größeren Gradienten als seine Nachbarn, wird dieses als Kantenpixel markiert, andernfalls wird es unterdrückt (Non-Maxima Suppression). Da benachbarte Kantenpixel einen relativ gleichbleibenden Gradienten besitzen, würde dieses Verfahren zu vielen Unterbrechungen der Kanten führen. Aus diesem Grund wird ein Hysterese-Schwelwert verwendet. Pixel werden sofort als Kantenpixel akzeptiert, wenn ihr Gradienten einen definierten Schwellwert überschreitet und sofort verworfen, sobald ihr Gradienten unter einem zweiten definierten Schwellwert liegt. Befindet sich der Gradienten zwischen den beiden Schwellwerten, wird das Pixel dann als Kantenpixel markiert, wenn es mit Pixeln verbunden ist, die einen großen Gradienten besitzen.

Abbildung 4.11 zeigt die Wirkungsweise des Canny-Operators. Das Eingangsbild 4.11a ist ein nach Abschnitt 4.4.2 erstelltes Tiefenbild. Das vom Canny-Operator erzeugte Kantenbild ist in Abbildung 4.11b zu sehen. Abbildung 4.11c zeigt die mit dem in Abschnitt 4.3 beschriebenen Verfahren ermittelten Konturen. Es ist zu sehen, dass trotz des Hysterese-Schwelwertes die Objektkonturen durch Unterbrechungen in viele Unterabschnitte aufgeteilt sind. Obwohl Verfahren existieren, um diese Konturlücken zu schließen [24], werden im Folgenden kantenorientierte Verfahren nicht weiter verfolgt.

### 4.2.2 $k$ -means Clustering

Das  $k$ -means Verfahren ist eine häufig verwendete Technik zum clustern von Datenpunkten. Das Verfahren benötigt eine vorgegebene Anzahl  $k$  für die zu findenden Cluster. Zunächst werden die Daten beliebig auf die Cluster aufgeteilt. Nach der Aufteilung wird der Mittelwert der Datenpunkte für jeden Cluster berechnet. Diesen Mittelwerten werden die einzelnen Datenpunkte so zugeordnet, dass jeder Datenpunkt seinem naheliegenden Mittelwert angehört. Dadurch entsteht eine neue Partitionierung der Daten. Es werden neue Mittelwerte berechnet und die Daten werden den neuen Mittelwerten zugeordnet. Diese Prozedur wird so lange angewendet, bis die Summe der Distan-



**Abbildung 4.13:** Ein globales Schwellwertverfahren erzeugt definierte Schnitte durch das Tiefenbild bei variierendem Schwellwert.

zen der Datenpunkte zu ihren Mittelwerten sich nicht mehr maßgeblich ändert. Die Summe der Distanzen ist eine mit den Iterationsschritten monoton fallende Funktion.

In dieser Arbeit wird das  $k$ -means Verfahren eingesetzt, um die Tiefenbilder zu segmentieren. Dabei werden die Pixel des Tiefenbildes anhand ihres Grauwertes geclustert. Abbildung 4.12 zeigt das Ergebnis eines mit fünf bzw. neun Clustern segmentierten Tiefenbildes. Es ist zu sehen, dass das Problem der in den Boden übergehenden Konturen auch mit einer größeren Anzahl von Clustern (bis zu 52 Cluster wurden getestet) nicht gelöst werden kann. Darüber hinaus zeigt der Vergleich von Abbildung 4.12 (b) und (c), dass Objekte zufällig richtig segmentiert werden. Dies liegt an der unbekanntenen Anzahl der benötigten Cluster, die von Tiefenbild zu Tiefenbild variiert. Eine Möglichkeit wäre, jedes Tiefenbild mehrmals mit unterschiedlicher Clusterzahl zu segmentieren. Dies ist aber eine ineffiziente Methode von zufälligem Erfolg und wird in dieser Arbeit nicht weiter verfolgt.

### 4.2.3 Schwellwertverfahren

Mit Hilfe von Schwellwertverfahren kann das Tiefenbild in ein binäres Bild umgewandelt werden. Da die Pixel eines binären Bildes nur zwei unterschiedliche Werte annehmen, ist es einfach zusammenhängende Bereiche in einem binären Bild zu extrahieren und das Bild somit zu segmentieren. Das Erfassen von zusammenhängenden Bereichen kann zum Beispiel mit dem in Abschnitt 4.3 beschriebenen Konturverfolgungsalgorithmus erreicht werden.

#### Konstanter globaler Schwellwert

Das einfachste Verfahren zur Erzeugung eines binären Bildes aus einem Grauwertbild ist der pixelweise Vergleich mit einer konstanten Grauwertschwelle. Liegt der Grauwert des Pixels über der Schwelle, wird er auf Eins gesetzt, andernfalls wird er auf Null gesetzt.

Dieses einfache Verfahren kann aufgrund der Tatsache, dass Objekte durch unterschiedliche Entfernung zum Scanner verschiedene Grauwerte besitzen, nicht angewendet werden. Wie in Abbildung 4.13 zu sehen ist, kann mit einem konstanten Schwellwert nur eine diskrete Entfernung gefiltert

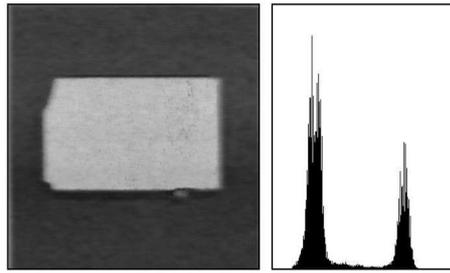


Abbildung 4.14: Grauwertbild und zugehöriges Histogramm [17].

werden. Befindet sich das Objekt hinter der vom Schwellwert definierten Entfernung, ist der Grauwert zu niedrig, und die Objektpixel sowie dessen Umgebung werden auf Null gesetzt, so dass das Objekt nicht mehr zu erkennen ist. Das gleiche gilt für den Fall, dass das Objekt sich zu dicht am Scanner befindet, nur dass diesmal alle Pixel auf eins gesetzt werden. Die Kontur des Objektes ist nur dann zu erkennen, wenn es sich in der vom Schwellwert definierten Entfernung befindet, so dass die Objektpixel mit Eins und die Umgebung des Objektes mit Null belegt werden. Mit diesem Verfahren können keine geschlossenen Konturen von Objekten extrahiert werden, die sich auf dem Boden befinden, denn der Boden auf dem das Objekt steht, besitzt immer einen Grauwert, der mindestens genauso groß ist wie der Grauwert des Objektes selbst.

### Grauwert-Histogramm und optimaler Schwellwert

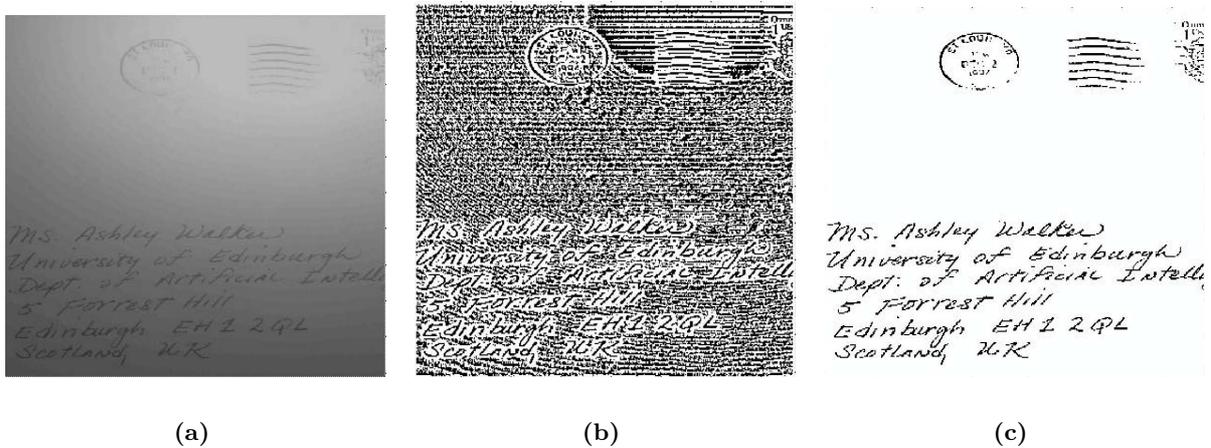
Mit Hilfe von Grauwert-Histogrammen lässt sich ein optimaler globaler Schwellwert berechnen, der das Bild in Vorder- und Hintergrund aufteilt.

In  $x$ -Richtung sind in einem Grauwert-Histogramm die Graustufen von 0 bis 255 aufgetragen und in  $y$ -Richtung die Anzahl der Pixel, die den jeweiligen Grauwert besitzen. Das Histogramm enthält somit die Information, wieviele Pixel welchen Grauwert besitzen. Abbildung 4.14 zeigt ein Grauwertbild und das dazugehörige Histogramm. Es ist zu sehen, dass sich hier die Grauwerte in zwei diskrete Maxima aufteilen. Wird der Schwellwert zwischen die beiden Maxima gelegt, entsteht ein binäres Bild, in dem das Objekt und der Hintergrund richtig markiert sind. Das Verfahren des optimalen Schwellwertes ist nur für Grauwertbilder geeignet, bei denen eine klare Trennung zwischen Objekten und Hintergrund gegeben ist. Eine mögliche Anwendung wäre die Segmentierung von hell reflektierenden Gegenständen auf einem nicht reflektierenden Lieferband.

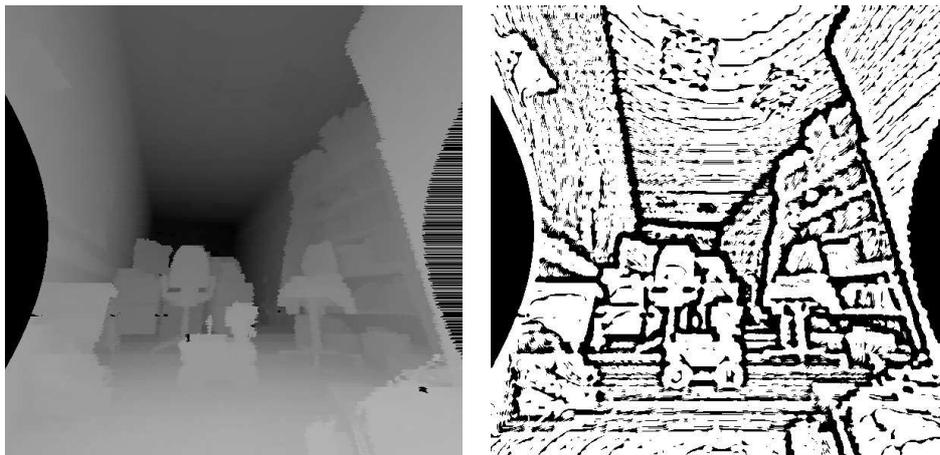
### Adaptiver Schwellwert

Bei dem adaptiven Schwellwertverfahren gibt es keinen globalen Schwellwert, sondern viele lokale Schwellwerte. Es werden hauptsächlich zwei unterschiedliche Strategien verwendet: der Chow und Kanenko-Ansatz und der lokale Schwellwerte-Ansatz.

Der Ansatz nach Chow und Kanenko unterteilt das Bild in sich überlappende Unterbilder. Für jedes dieser Unterbilder wird ein Grauwert-Histogramm erstellt, mit dessen Hilfe ein optimaler Schwellwert berechnet wird. Die Schwellwerte der Unterbilder werden interpoliert, so dass man für jedes Pixel einen lokalen Schwellwert erhält. Dieser Ansatz liefert zwar gute Resultate, ist für Echtzeitanwendungen aber zu rechenintensiv [38].



**Abbildung 4.15:** Wirkungsweise eines kombinierten Verfahrens aus adaptivem und lokalem Schwellwert. In (a) ist das Originalbild zu sehen, dass nach einem adaptiven Schwellwert in homogenen Regionen zu Rauschen führt (b). Kombiniert mit einem globalen Schwellwert wird dieses Rauschen eliminiert [38].



**Abbildung 4.16:** Original Tiefenbild (a), das mit einem kombinierten Verfahren aus adaptivem und globalem Schwellwert binarisiert wurde (b)

Der Ansatz der lokalen Schwellwerte betrachtet jedes Pixel einzeln und berechnet dessen Schwellwert in Abhängigkeit zu den Grauwerten in seiner Umgebung. Dieser Ansatz liefert zwei Probleme:

1. Wie sollen die Grauwerte der Umgebungspixel gewichtet werden?
2. Wie groß ist die zur Schwellwertberechnung herangezogene Umgebung zu wählen?

Zum ersten Problem ist zu sagen, dass es keine einheitlich richtige Methode gibt, sondern die Wahl der richtigen Gewichte von der Natur des Bildes abhängt. Häufig verwendete Gewichtungen sind:

Der durchschnittliche Grauwert, der Median oder eine Gaußverteilung.

Bei der Wahl der Umgebungsgröße ist zu beachten, dass eine große Umgebung zu einem Ergebnis führt, dass gröbere Segmente erzeugt und kleinere Regionen mit zur Umgebung verschiedenem Grauwert unterdrückt. Dies kann bei der Segmentierung von relativ großen Objekten ein durchaus gewünschter Effekt sein. Ein Nachteil großer Umgebungen ist die höhere Rechenintensität im Vergleich zu kleinen Umgebungen. Wird die Umgebung zu klein gewählt, erhält eventuelles Rauschen einen zu starken Einfluß.

Die Auswirkung eines adaptiven Schwellwertes auf ein Bild, das einen Grauwertgradienten enthält, ist in Abbildung 4.15 zu sehen. Es fällt auf, dass die Schrift gut separiert werden kann. Dies liegt daran, dass im Bereich der Schrift stark unterschiedliche Grauwerte in der Umgebung eines Pixels liegen, so dass der ermittelte Schwellwert zwischen Hintergrund und Schrift liegt. Im Hintergrundbereich besitzen die Umgebungspixel einen mit dem aktuell untersuchten Pixel vergleichbaren Grauwert. Dies führt dazu, dass der Schwellwert in etwa dem Grauwert des Pixels entspricht und dieser fast zufällig auf eins oder Null gesetzt wird, so dass der Hintergrund nicht als Hintergrund separiert wird.

Um dieses Problem zu lösen, wird das adaptive Schwellwertverfahren mit einem globalen Schwellwertverfahren kombiniert. Bei der Berechnung des Schwellwertes für einen Pixel wird vom Pixelwert noch ein globaler Schwellwert abgezogen, so dass Pixel in einer homogenen Umgebung auf Null gesetzt werden. Das Ergebnis des kombinierten Verfahrens ist in Abbildung 4.15c zu sehen.

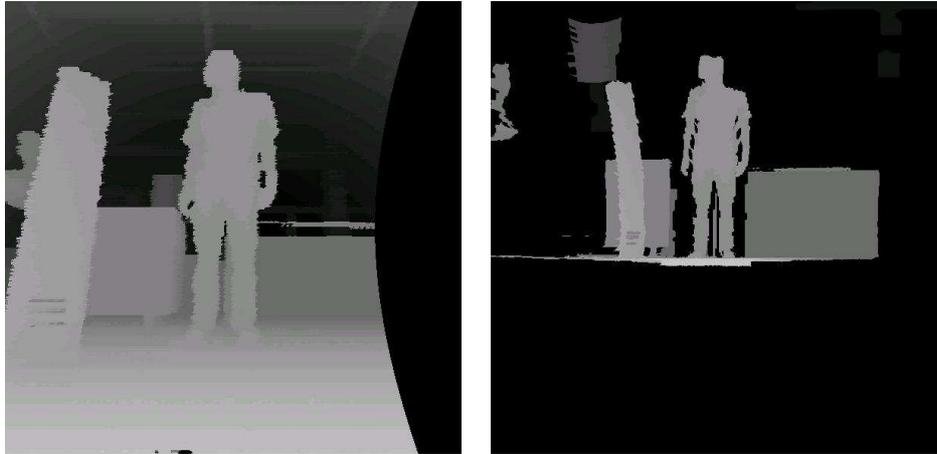
Da die zu verarbeitenden Tiefenbilder mit Abbildung 4.15a vergleichbar sind, weil sie durch den entfernungsabhängigen Grauwert alle einen Grauwertgradienten besitzen, wird das kombinierte Verfahren zur Binarisierung eingesetzt. Abbildung 4.16 zeigt das Ergebnis eines auf diese Weise binarisierten Tiefenbildes.

### 4.3 Konturextraktion

Aus dem binarisierten Bild werden die Konturen mit einem Konturverfolgungsalgorithmus nach [58] extrahiert. Dieser geht das Bild zeilenweise durch und beginnt eine Konturverfolgungsprozedur, sobald er auf einen noch nicht markierten Konturpunkt trifft. Die Konturverfolgungsprozedur verfolgt, wie der Name schon sagt, die Kontur und markiert jeden Konturpunkt dabei eindeutig. Der Algorithmus geht das Bild so lange durch, bis keine neuen Konturpunkte gefunden werden.

### 4.4 Verbesserungen am Eingabebild

Da das Problem der in den Boden übergehenden Konturen durch die vorgestellten Bildvorverarbeitungs- bzw. Segmentierungsverfahren nicht gelöst wird, erläutert dieser Abschnitt verschiedene Ansätze, die ein modifiziertes Tiefenbild erzeugen. Dieser Abschnitt ist damit klar von der Bildvorverarbeitung und der Segmentierung abzugrenzen, da es sich nicht um Algorithmen handelt, die auf das Tiefenbild angewendet werden, sondern um unterschiedliche Verfahren, aus den 3D-Datenpunkten Tiefenbilder zu generieren.



**Abbildung 4.17:** Perspektivische Projektion (a) und parallele Projektion (b) der 3D-Datenpunkte.

#### 4.4.1 Parallele Projektion

Der erste Ansatz ist der, das Kameramodel zu ändern mit dem das Tiefenbild aufgenommen wird. Der virtuelle Beobachter, der sich in den 3D-Daten befindet, hat bis jetzt immer Tiefenbilder mit einem Lochkameramodell erzeugt. Wird das Kameramodel in eine Kamera mit Parallelprojektion geändert, bleiben die  $x$ - und  $y$ -Achse erhalten und die  $z$ -Achse wird direkt abgebildet. Das Projektionszentrum der Lochkamera wandert ins Unendliche, so dass eine Projektionsrichtung und kein Projektionszentrum erzeugt wird. Dies hat zur Folge, dass das Tiefenbild keine Perspektive mehr besitzt, da ein Objekt im Bildvordergrund genauso groß abgebildet wird, als wenn es sich im Bildhintergrund befinden würde.

In einem Tiefenbild eines 3D-Scans, der exakt parallel zum Boden aufgenommen wurde, ist der Boden aufgrund obiger Erläuterungen nur als Linie zu sehen. Diese Linie besitzt einen vom Objekt verschiedenen Grauwert, da sich die ersten Bodenpunkte sehr dicht am Scanner befinden und dementsprechend einen hohen Grauwert zugewiesen bekommen. Diese Eigenschaft ermöglicht es, ein Objekt, das sich auf dem Boden befindet, von diesem zu separieren. Abbildung 4.17 zeigt ein Tiefenbild aus perspektivischer und paralleler Sicht. Im parallel projizierten Bild ist zu sehen, dass der Boden nicht exakt zu einer Linie wird und es immer noch einen fließenden Grauwertübergang gibt. Die exakte Parallelität des 3D-Scans zum Boden ist also experimentell nicht gegeben. Aus diesem Grund wird das Verfahren nicht verwendet. Statt dessen kommt das im folgenden Abschnitt beschriebene Verfahren zum Einsatz

#### 4.4.2 Entfernung der Bodenpunkte im Tiefenbild

Bei diesem Ansatz wird der Abbildung der 3D-Punkte auf einen Grauwertbereich eine Interpretation derselben vorangestellt. Diese ermöglicht es, die Menge der 3D-Punkte in Boden-, Decken- und Objektpunkte aufzuteilen. Das Verfahren berechnet den Gradienten zwischen benachbarten Punkten und teilt sie anhand diesem einer der drei Gruppen zu [47][67]. Dazu werden durch die Punktwolke von der Scannerposition ausgehend vertikale Schnitte gelegt. In einem Zylinderkoordi-

natensystem besitzt die Punktvolke die Darstellung  $p_{i,j} = (\Phi_i, r_{i,j}, z_{i,j})^T$ , wobei  $\Phi_i$  der Drehwinkel der jeweiligen vertikalen Schnittebene ist und  $z_{i,j}$  bzw.  $r_{i,j}$  die Koordinaten eines der sich in der vertikalen Ebene  $i$  befindlichen Punktes  $j$  sind. Der Gradient  $\alpha_{i,j}$  zwischen zwei benachbarten Punkten lässt sich mittels Gleichung 4.4 bestimmen.

$$\alpha_{i,j} = \arctan\left(\frac{z_{i,j} - z_{i,j-1}}{r_{i,j} - r_{i,j-1}}\right) \quad \text{mit} \quad -\frac{1}{2}\pi \leq \alpha_{i,j} < \frac{3}{2}\pi \quad (4.4)$$

Der Vergleich von  $\alpha_{i,j}$  mit einem festen Schwellwert ermöglicht eine Interpretation des Punktes:

1.  $\alpha_{i,j} < \tau$ : Punkt  $j$  ist ein Bodenpunkt
2.  $\tau \leq \alpha_{i,j} \leq \pi - \tau$ : Punkt  $j$  ist ein Objektpunkt
3.  $\pi - \tau < \alpha_{i,j}$ : Punkt  $j$  ist ein Deckenpunkt

Um dieses Verfahren unempfindlich gegenüber Rauschen zu machen, wird nicht der direkte Nachbar eines Punktes betrachtet, sondern ein weiter entfernter Punkt mit dem Index-Offset  $k$ :

$$\tan \alpha_{i,j} = \frac{z_{i,j} - z_{i,j-k}}{r_{i,j} - r_{i,j-k}}$$

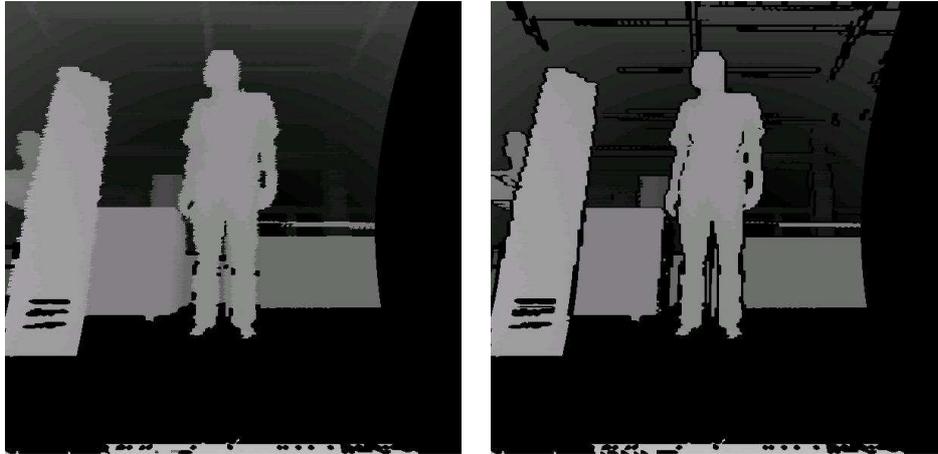
Wie groß der Abstand der beiden Punkte sein muss, lässt sich mit Formel (4.5) berechnen. Bei dieser Formel wird die Tiefenaufösung des Scanners  $d_{min}$  mit berücksichtigt. Der Punkt  $k$ , der zur Gradientenberechnung herangezogen wird, muss die Bedingung erfüllen, dass seine Entfernung zum aktuellen Punkt größer ist als die Tiefenaufösung, da der Punkt  $k$  ansonsten durch Rauschen entstanden sein muss:

$$\sqrt{(r_{i,j} - r_{i,j-k})^2 + (z_{i,j} - z_{i,j-k})^2} > d_{min} \quad (4.5)$$

Ein weiteres Problem dieses Verfahrens sind Sprünge zwischen Kanten, wie sie beispielsweise durch das Scannen von offenen Türen entstehen. Ein Bodenpunkt wird nicht als solcher erkannt, wenn er mit einem Punkt verglichen wird, der sich auf einer anderen Kante befindet. Um dieses Problem zu lösen, wird ein maximaler Entfernungswert definiert, den zwei Punkte besitzen dürfen, die zur Gradientenberechnung herangezogen werden.

### 4.4.3 Manipulation der Grauwertinterpolation

Da die gemessenen 3D-Punkte nicht der Anzahl der Pixel im Tiefenbild entsprechen, muss zwischen den Grauwerten der 3D-Punkte interpoliert werden, um jedes Pixel einen Grauwert zuweisen zu können. Diese Tatsache bietet die Ansatzmöglichkeit für eine weitere Vereinfachung, indem nicht immer interpoliert wird, sondern nur wenn die Entfernung zwischen den 3D-Punkten nicht zu groß ist. Bei großen Entfernungen zwischen zwei 3D-Punkten wird nicht interpoliert, so dass die Pixel auf Null gesetzt bleiben. Kombiniert mit den in Abschnitt 4.4.2 beschriebenen Verfahren zur Entfernung der Bodenpunkte ergibt sich ein Tiefenbild, wie es in Abbildung 4.18b zu sehen ist. Bei diesem Tiefenbild beträgt die maximale Entfernung zwischen zwei 3D-Punkten, bei der noch interpoliert wird, 10 cm. Dadurch sind alle Objekte, die eine größere Entfernung zu hinter sich befindlichen Objekten besitzen, mit einer schwarzen Kontur umgeben.

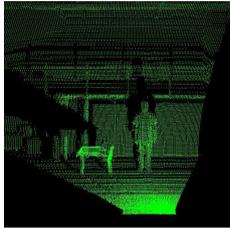


**Abbildung 4.18:** (a) Tiefenbild, das aus 3D-Daten mit entfernten Bodenpunkten generiert wurde. (b) Tiefenbild mit entfernter Grauwertinterpolation an Objektkanten und entfernten Bodenpunkten.

Ein Nachteil dieses Verfahrens ist, dass die in Abschnitt 4.1.2 beschriebenen Verfahren zum Entfernen von Digitalisierungsfehlern nicht weiter angewendet werden können, da die durch Digitalisierungsfehler entstandenen horizontalen Linien den selben Charakter besitzen wie die durch dieses Verfahren entstandenen schwarzen Konturen. Eine Glättung würde ebenso die schwarzen Ränder der Objekte entfernen. Da die Digitalisierungsfehler nur selten vorkommen, kann dieser Nachteil in Kauf genommen werden, insbesondere deshalb, weil durch dieses Verfahren die Segmentierung des Bildes schon fast gegeben ist.

## 4.5 Endfassung des Konturextraktionsalgorithmus

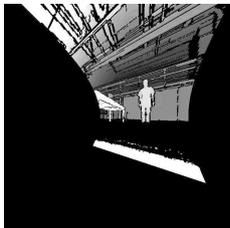
Da in diesem Kapitel unterschiedliche, sich teilweise gegenseitig ausschließende Verfahren vorgestellt wurden, wird in diesem Abschnitt die in dieser Applikation verwendete Endfassung beschrieben. Den Ausgangspunkt der Konturextraktion bilden die 3D-Datenpunkte, die mit dem Laserscanner aufgenommen werden. Auf diese Datenpunkte wird ein semantischer Klassifizierer angewendet, der die Bodenpunkte entfernt 4.4.2. Zusätzlich wird jedes Objekt, das sich von seinem Hintergrund abhebt, nach 4.4.3 mit einem schwarzen Rand umgeben. Danach wird mit dem adaptivem Schwellwertverfahren 4.2.3 aus dem Grauwertbild ein Binärbild generiert. Da in diesem Binärbild verschiedene Segmente oft durch einzelne Pixel verbunden sind (vgl. Abbildung 4.16b), wird ein binäres Öffnen nach 4.1.3 angewendet, um die Segmente voneinander zu trennen. Um aus dem geöffneten Binärbild die Konturen zu extrahieren, wird der Konturverfolgungsalgorithmus nach 4.3 angewendet. Abbildung 4.19 zeigt diesen Ablauf und das Tiefenbild in den einzelnen Verarbeitungsschritten.



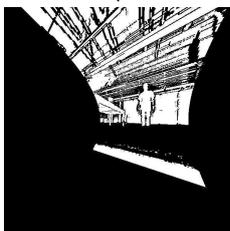
Ausgangspunkt der Konturextraktion sind die 3D-Datenpunkte.



Aus den Datenpunkten wird ein Tiefenbild mit entfernten Bodenpunkten und schwarz umrandeten Objekten generiert.



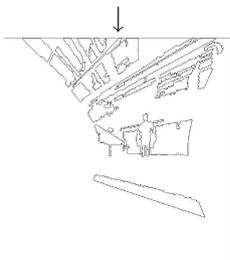
Der Kontrast im Tiefenbild wird mittels Grauwertdehnung mit Clipping erhöht. Dabei wird das Clipping nur auf die hohen Grauwerte im Bild angewendet.



Mittels adaptivem Schwellwert-Verfahren wird das Bild binarisiert. Zum Berechnen der lokalen Schwellwerte wird eine große Umgebung von  $101 \times 101$  Pixeln herangezogen.



Die Objekte im binarisierten Bild werden durch die morphologische Öffnungsoperation besser voneinander separiert. Kleine Strukturen werden komplett unterdrückt.



Auf das geöffnete Bild wird der Konturverfolgungsalgorithmus angewendet.

**Abbildung 4.19:** Schematischer Ablauf der Konturextraktion



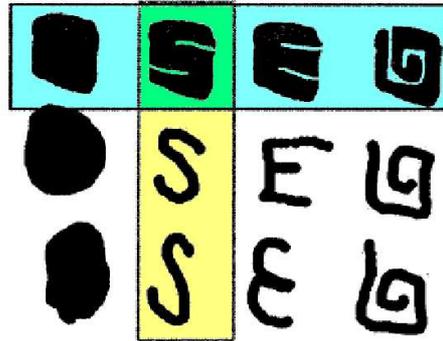
## Kapitel 5

# Konturrepräsentation und Merkmalsextraktion

In diesem Kapitel werden zunächst die Begriffe Repräsentation und Merkmal einer Kontur geklärt. Anschließend werden alle in dieser Arbeit zur Klassifikation einer Kontur verwendeten Repräsentationen und Merkmale vorgestellt.

Es gibt zwei verschiedene Herangehensweisen, die Form von Objekten zu repräsentieren: kontur- und regionorientiert. Die erste Kategorie repräsentiert die Form, wie der Name schon sagt, durch ihre Kontur, während die zweite Kategorie die Form als Komposition von zweidimensionalen Regionen auffasst. Abbildung 5.1 zeigt, welche Objekte einander in den unterschiedlichen Beschreibungstechniken ähneln. Techniken in beiden Kategorien werden in räumliche (spatial domain) bzw. transformierte Domänen unterteilt. Die räumliche Domäne verwendet numerische, im Ortsraum erzeugte Merkmale, um die Form zu beschreiben. Techniken der transformierten Domäne erzeugen Deskriptoren mit deren Hilfe die Form in einen anderen Raum abgebildet wird, so dass die formbeschreibenden Merkmale in diesem Raum (z.B. Frequenzraum) erzeugt werden [70]. Die Techniken der transformierten Domäne werden in „singlescale“- (z.B. Fourier Deskriptoren) bzw. „multiscale“- Repräsentationen (z.B. Curvature-Scale-Space-Repräsentation) unterteilt [10]. Eine weitere wichtige Unterteilung der Repräsentationstechniken unterscheidet zwischen globalen und lokalen (auch strukturellen) Techniken. Globale Techniken verwenden zur Repräsentation die Form als Ganzes, während lokale Techniken Teile der Form verwenden, um diese zu beschreiben. Abbildung 5.2 zeigt einen Überblick der Techniken. Eine Zusammenfassung aktueller Techniken wird in [70] gegeben.

Durch die Merkmalsextraktion sollen die in den zu klassifizierenden Daten vorhandenen relevanten Informationen auf einen niedrigdimensionaleren Merkmalsraum abgebildet werden. Dabei sollen redundante und unwichtige Signale unterdrückt werden, so dass die extrahierten Merkmale die gesamte relevante Information des Objektes besitzen. Es gibt einfache, skalare, geometrische Merkmale (siehe Abschnitt 5.3), die keine explizite Repräsentation der Kontur benötigen, sondern sich direkt aus der Kontur berechnen lassen. Zu ihnen gehören Umfang, Seitenverhältnis des umschließenden Rechtecks, Kreisförmigkeit, Exzentrizität, Füllungsgrad usw. Komplexere Merkmale werden erzeugt, indem zunächst eine Repräsentation der Kontur bestimmt wird, aus der dann die Merkmale extrahiert werden. Die Repräsentationen sind zumeist eindimensionale Funktionen, die aus den Punkten der zweidimensionalen Kontur gewonnen werden [10]. Hierbei können aus einer Repräsentationstechnik verschiedene Merkmale bestimmt werden. Welche Repräsentationstechnik



**Abbildung 5.1:** Vergleich von regionorientierten und konturorientierten Formmerkmalen: Regionorientierte Verfahren werden die blau hinterlegten Objekte als ähnlich erachten, während konturorientierte Verfahren die vertikalen, gelb hinterlegten Objekte als ähnlich erachten werden [6].

bzw. welches Merkmal dieser Repräsentation verwendet wird, hängt von der Anwendung ab, und es gibt keine Theorie, die besagt, welche Merkmale bei einer bestimmten Anwendung extrahiert werden müssen [1]. Somit ist die allgemeine Vorgehensweise die, dass erst viele Merkmale extrahiert werden und danach mittels statistischer Verfahren geprüft wird, wie stark die einzelnen Merkmale korreliert sind.

## 5.1 Anforderungen an die Konturrepräsentationen

Damit eine Konturrepräsentation Objekterkennung ermöglichen kann, muss sie eine Anzahl von Kriterien erfüllen [42]. In den folgenden Definitionen werden zwei Konturen als gleich bezeichnet, wenn die eine Kontur durch eine Kombination aus Rotation, Translation und Skalierung auf die zweite abgebildet werden kann.

**Invarianz:** Wenn zwei Konturen die gleiche Form besitzen, sollten die Repräsentationen identisch sein.

**Eindeutigkeit:** Wenn zwei Konturen unterschiedliche Formen besitzen, sollten die Repräsentationen unterschiedlich sein.

**Stabilität:** Wenn zwei Konturen sich nur geringfügig unterscheiden, sollte das auch für ihre Repräsentationen gelten. Ebenso sollten zwei einander ähnelnde Repräsentationen zu sich ähnelnden Konturen gehören.

Neben diesen notwendigen Bedingungen gibt es eine Reihe von weiteren Bedingungen, die beim Einsatz einer Konturrepräsentation in einer praktischen Anwendung ebenfalls gewährleistet sein sollten.

**Lokale Information:** Eine Konturrepräsentation sollte lokale Informationen einer Kontur verwenden, so dass es möglich ist, die Repräsentationen von Teilen einer Kontur zu vergleichen.

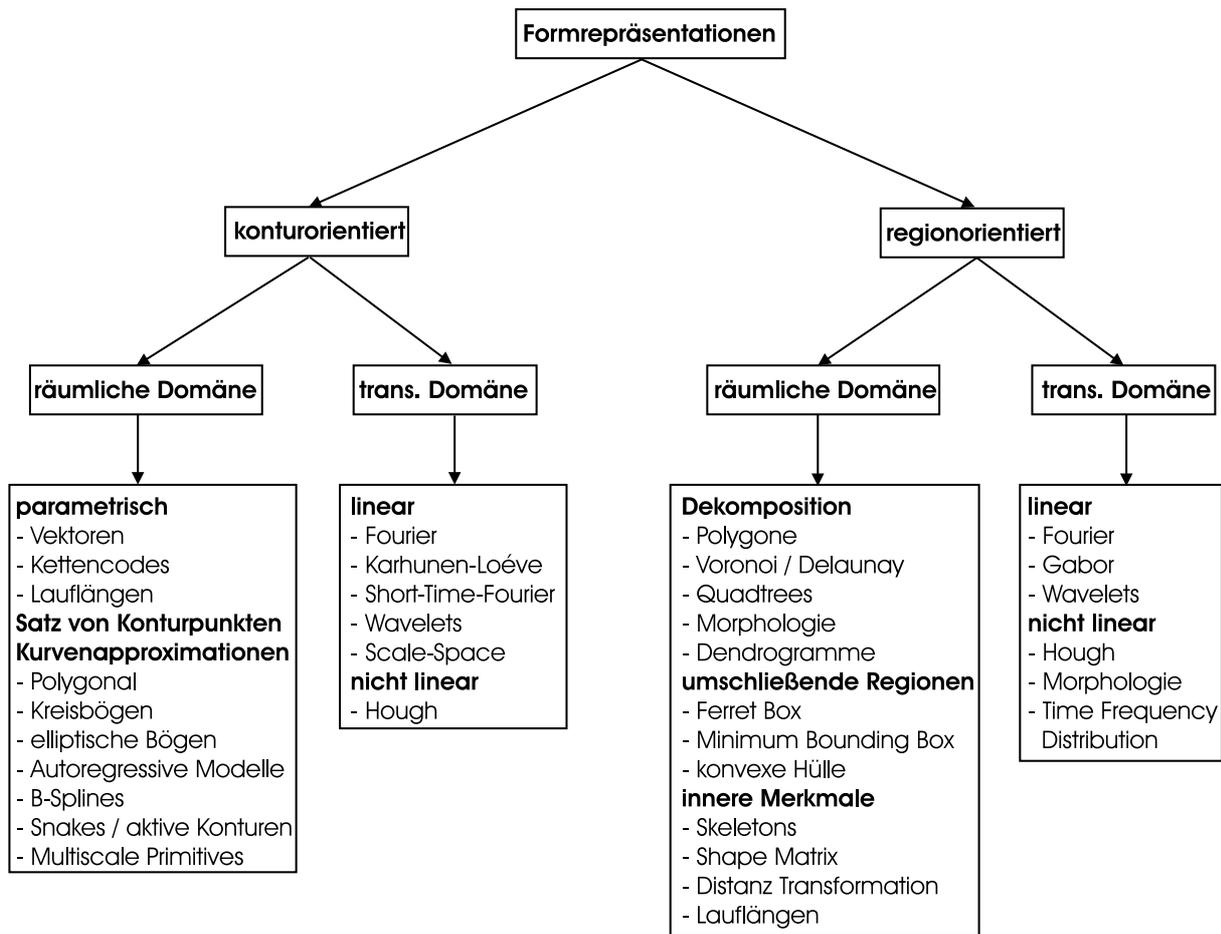


Abbildung 5.2: Taxonomie der Formrepräsentationstechniken nach [52]

Effektivität: Die Repräsentation sollte effizient zu berechnen und zu speichern sein, da Objekterkennung oft in Echtzeit betrieben wird.

Formeigenschaften: Falls die Kontur symmetrisch ist, sollte auch die Repräsentation symmetrisch sein (Symmetrie-Kriterium). Wenn die Kontur Teil einer anderen zweiten Kontur ist, sollte sich auch in der Repräsentation der zweiten Kontur die Repräsentation der ersten Kontur wieder finden lassen (Teil-Ganzes-Kriterium).

## 5.2 Anforderung an die Merkmale

An die aus einer Konturrepräsentation extrahierten Merkmale werden, wie an die Repräsentation selbst, gewisse Anforderungen gestellt [7].

Robustheit: Die Merkmale müssen robust gegenüber Rauschen und Rechenungenauigkeiten sein.

Rechenkomplexität: Es muss die Möglichkeit gegeben sein, die Merkmale effizient zu berechnen.

**Diskriminanz:** Die zu untersuchenden Konturen müssen sich durch die Merkmale beschreiben, klar trennen und klassifizieren lassen.

**Generalität:** Die Datenbank der zu klassifizierenden Objekte muss für einen Merkmalsatz erweiterbar sein, ohne dass die Diskriminanzbedingung aufgehoben wird.

**Flexibilität:** Das System muss sich einfach an sich ändernde Randbedingungen anpassen, ohne einen neuen Merkmalsatz extrahieren zu müssen.

## 5.3 Einfache geometrische Merkmale

### Objektfläche und Umfang

Die Objektfläche ist eines der einfachsten geometrischen Merkmale. In einem binären Bild  $B(x, y)$  besitzen Objektpixel den Wert 1 und Hintergrundpixel den Wert 0. Der Flächeninhalt  $A$  ergibt sich als Summe aller Objektpixel des Bildes,

$$A = \sum_{x=1}^M \sum_{y=1}^N B(x, y).$$

wobei  $M$  die Spalten- und  $N$  die Zeilenanzahl ist. Die Objektfläche ist ein translations- und rotationsinvariantes Merkmal. Gegenüber Skalierung ist die Objektfläche trivialerweise nicht invariant. Der Objektumfang lässt sich aus einer modifizierten Bildmatrix  $C(x, y)$  berechnen, in der alle inneren Objektpixel ebenfalls auf 0 gesetzt werden. In einem solchen Bild ergibt sich der Umfang  $U$  analog zur Objektfläche als Summe aller Pixel:

$$U = \sum_{x=1}^M \sum_{y=1}^N C(x, y).$$

Ebenso wie die Objektfläche handelt es sich beim Objektumfang um ein translations- und rotationsinvariantes Merkmal, das jedoch variant gegenüber Skalierung ist.

Ein Nachteil dieser Berechnungsweise ist der, dass nicht zwischen Konturpixeln, die diagonal miteinander verbunden sind, und Konturpixeln, die 4er Nachbarschaft besitzen, unterschieden wird. Eine exaktere Berechnung des Umfangs wird erreicht, indem die Kontur durch den Freeman-Kettencode aus Abschnitt A beschrieben wird. Auf diese Weise ist es möglich, diagonal verbundene Pixel mit  $\sqrt{2}$  zu gewichten, was zu einem besseren Ergebnis führt.

### Kreisförmigkeit

Die Kreisförmigkeit  $C$  einer Kontur (auch Dünneheitsverhältnis) ist ein rotations-, skalierungs- und translationsinvariantes Merkmal [32]. Es stellt ein Verhältnis aus Fläche  $A$  und Umfang  $U$  einer Kontur dar.

$$C = \frac{4\pi A}{U^2}$$

Die Kreisförmigkeit reicht von 0 für eine Linie, bis 1 für einen Kreis. Es gibt keine größeren Werte, da der Kreis die maximale Fläche bei gegebenem Umfang besitzt.

### Seitenverhältnis

Das Merkmal „Seitenverhältnis“ (engl.: *aspect ratio*) beschreibt das Seitenverhältnis eines die Kontur umschreibenden Rechtecks. Es gibt zwei verschiedene Rechtecke: das kleinste, umschreibende, achsenparallele Rechteck (engl.: *Ferret box*). Dieses überdeckt die Kontur komplett, besitzt minimalen Flächeninhalt und ist parallel zu den Koordinatenachsen ausgerichtet. Das Verhältnis der Seitenlängen ist skalierungs- und translationsinvariant. Die Rotationsinvarianz lässt sich erreichen, indem die Bedingung der Parallelität des Rechtecks zu den Koordinatenachsen aufgehoben wird. Dieses Rechteck wird umschreibendes Rechteck minimaler Fläche (engl.: *minimum bounding rectangle-MBR*) genannt. Bei diesem Rechteck wird das Seitenverhältnis der längeren zur kürzeren Seite berechnet. Der Winkel, um den das achsenparallele Rechteck gedreht werden muss, um das umschreibende Rechteck minimaler Fläche zu erhalten, lässt sich durch das Merkmal Orientierung bestimmen. Meist wird das Verhältnis der Seitenlängen logarithmisch angegeben, da gilt

$$\left| \log \frac{H}{B} \right| = \left| \log \frac{B}{H} \right|,$$

wobei  $H$  die Höhe und  $B$  die Breite des Rechtecks ist.

### Füllungsgrad

Das Merkmal „Füllungsgrad“ (auch relative Fläche) wird aus dem Verhältnis der Objektfläche zu einer definierten Vergleichsfläche gebildet und berechnet sich nach Gleichung 5.1. Als Vergleichsfläche wird häufig die Fläche eines umschließenden Rechtecks minimaler Fläche verwendet. Bei dieser Vergleichsfläche ist der Füllungsgrad ein translations-, rotations- und skalierungsinvariantes Merkmal. Wird das kleinste umschreibende, achsenparallele Rechteck verwendet, geht die Invarianz des Merkmals gegenüber Rotation verloren.

$$F_G = \frac{A}{H \cdot B} \quad (5.1)$$

### Schwerpunkt

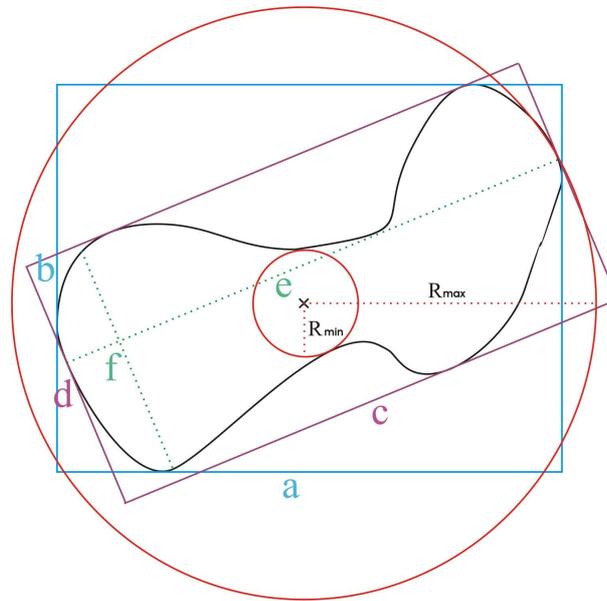
Die Koordinaten des geometrischen Schwerpunktes für ein Objekt werden gemäß folgender Gleichung berechnet:

$$\bar{x} = \frac{1}{A} \sum_{i=1}^A x_i \quad \text{und} \quad \bar{y} = \frac{1}{A} \sum_{i=1}^A y_i.$$

Der Schwerpunkt ist invariant gegenüber Rotation und homogener Skalierung, aber variant gegenüber Translation.

### Radienverhältnis

Das Merkmal „Radienverhältnis“ ist translations-, rotations-, und skalierungsinvariant. Es wird aus dem Verhältnis der Radien des kleinsten umschreibenden Kreises mit dem des größten Kreises, der vollständig im Objekt liegt, gebildet. Beide Kreise besitzen als Mittelpunkt den Schwerpunkt des Objektes. Das Merkmal „Radienverhältnis“ beschreibt zusammen mit den Merkmalen „Seitenverhältnis“ und „Kreisförmigkeit“ die Kompaktheit des Objektes.



**Abbildung 5.3:** Einfache geometrische Merkmale: umschließendes achsenparalleles Rechteck minimaler Fläche (blau), umschließendes Rechteck minimaler Fläche (lila) Hauptachse  $e$  und Nebenachse  $f$  des Objektes (grün) und minimaler umschreibender Kreis bzw. maximaler innerer Kreis (rot)

### Exzentrizität

Die Exzentrizität sowie die im Folgenden beschriebene Orientierung verwenden zur Berechnung die in Abschnitt 5.4 beschriebenen Momente. Sie sind trotzdem an dieser Stelle aufgeführt, da sie eine geometrische Interpretation besitzen.

Die Exzentrizität (auch Elongation) ist translations-, rotations-, und skalierungsinvariantes Merkmal und beschreibt das Verhältnis der Hauptachse zur Nebenachse eines Objekts. Die Hauptachse ist die längste Sehne durch das Objekt und die Nebenachse die längste Sehne, die senkrecht zu dieser verläuft. Abbildung 5.3 macht dies deutlich. Gleichung 5.2 zeigt, wie Exzentrizität aus den zentralen Momenten berechnet wird.

$$e = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{(\mu_{20} + \mu_{02})^2} \quad (5.2)$$

### Orientierung

Das Merkmal Orientierung beschreibt die Hauptrichtung des Objektes. Es handelt sich um ein translations- und skalierungsinvariantes Merkmal.

$$\Theta = \frac{1}{2} \arctan \left( \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (5.3)$$

## 5.4 Merkmalsextraktion mittels Momenten

Momente lassen sich für allgemeine Grauwertbilder berechnen. Der Grauwert eines Objektes macht in dieser Arbeit aber keine Aussage über das Objekt selbst, da sich das Objekt in beliebiger Entfernung zum Scanner befinden und somit einen beliebigen Grauwert annehmen kann. Aus diesem Grund werden die Momente für binarisierte Bilder berechnet. Momente entnehmen dem Bild globale Informationen und benötigen keine geschlossenen Konturen wie es zum Beispiel bei Fourier Descriptoren der Fall ist [29].

Diskrete geometrische Momente (auch reguläre Momente) einer zweidimensionalen Bildfunktion  $B(x, y)$ , lassen sich nach Gleichung 5.4 berechnen. Dabei bezeichnet  $N$  die Zeilen- und  $M$  die Spaltenanzahl. Das Moment besitzt die Ordnung  $j = (p + q)$ , und zu jeder Ordnung  $j$  gibt es  $j + 1$  Momente.

$$m_{p,q} = \sum_{x=1}^M \sum_{y=1}^N B(x, y) x^p y^q \quad \text{mit } p, q = 0, 1, 2, \dots \quad (5.4)$$

Da diese Momente nicht translationsinvariant sind, werden sie zu zentralen Momenten weiterverarbeitet. Zur Berechnung der zentralen Momente wird der Schwerpunkt der Kontur benötigt. Dieser berechnet sich aus den geometrischen Momenten nach Gleichung 5.5.

$$\bar{x} = \frac{m_{1,0}}{m_{0,0}} \quad \bar{y} = \frac{m_{0,1}}{m_{0,0}} \quad (5.5)$$

Hierbei bezeichnet  $m_{0,0}$  das Moment 0. Ordnung. Dieses gibt die Summe aller Pixelwerte eines Bildes an und entspricht bei einer binären Bildfunktion dem Flächeninhalt.  $m_{1,0}$  und  $m_{0,1}$  werden als Zeilenmoment bzw. Spaltenmoment erster Ordnung bezeichnet. Mit Hilfe der Schwerpunktskoordinaten können die zentralen Momente definiert werden.

$$\mu_{p,q} = \sum_{x=1}^M \sum_{y=1}^N B(x, y) (x - \bar{x})^p (y - \bar{y})^q$$

Durch die Verschiebung um  $(\bar{x}, \bar{y})$  in ein dem Objekt angepasstes Koordinatensystem sind die zentralen Momente translationsinvariant aber nicht skalierungsinvariant. Um die Skalierungsinvarianz zu erreichen, müssen die Momente normiert werden. Der Normierungsfaktor wird ermittelt, indem die Momente einer Bildfunktion  $B(x, y)$  mit denen einer um  $\alpha$  skalierten Bildfunktion  $B'(x', y')$  verglichen werden [7]. Es gilt:  $B(x, y) = B'(\alpha x, \alpha y) = B'(x', y')$ .

$$\begin{aligned} \int \int B'(x', y') x'^p y'^q dx' dy' &= \int \int B(x, y) (\alpha x)^p (\alpha y)^q \alpha dx \alpha dy \\ &= \alpha^{p+q+2} \int \int B(x, y) x^p y^q dx dy \end{aligned}$$

Wird zusätzlich berücksichtigt, dass das geometrische Moment  $m_{0,0}$  dem Flächeninhalt entspricht, legt dies eine Normierung der zentralen Momente nach Gleichung 5.6 nahe.

$$\eta_{p,q} = \frac{\mu_{p,q}}{(m_{0,0})^\alpha} \quad \text{mit } \alpha = \frac{p+q}{2} + 1 \quad (5.6)$$

Diese normierten Momente  $\eta_{p,q}$  werden normalisierte Momente genannt und sind translations- und skalierungsinvariant. Die fehlende Rotationsinvarianz wird erst durch geeignete Kombination der normalisierten Momente erreicht.

### 5.4.1 Hu-Momente

Hu verwendet in seiner Arbeit von 1962 [20] normalisierte Momente, um daraus rotations-, translations- und skalierungsinvariante Merkmale zu extrahieren.

Durch eine nichtlineare Kombination normalisierter Momente bis zur 3. Ordnung konnte Hu folgende sieben invariante Merkmale berechnen.

$$\begin{aligned}
h_1 &= \eta_{2,0} + \eta_{0,2} \\
h_2 &= [\eta_{2,0} - \eta_{0,2}]^2 + 4[\eta_{1,1}]^2 \\
h_3 &= [\eta_{3,0} - 3\eta_{1,2}]^2 + [\eta_{0,3} - 3\eta_{2,1}]^2 \\
h_4 &= [\eta_{3,0} + \eta_{1,2}]^2 + [\eta_{0,3} + 2\eta_{2,1}]^2 \\
h_5 &= [\eta_{3,0} - 3\eta_{1,2}][\eta_{3,0} + \eta_{1,2}]\{[\eta_{3,0} + \eta_{1,2}]^2 - 3[\eta_{0,3} + \eta_{2,1}]^2\} \\
&\quad + [3\eta_{2,1} - \eta_{0,3}][\eta_{0,3} + \eta_{2,1}]\{3[\eta_{3,0} + \eta_{1,2}]^2 - [\eta_{0,3} + \eta_{2,1}]^2\} \\
h_6 &= [\eta_{2,0} + \eta_{0,2}]\{[\eta_{3,0} + \eta_{1,2}]^2 - [\eta_{0,3} + \eta_{2,1}]^2\} \\
&\quad + 4\eta_{1,1}[\eta_{3,0} + \eta_{1,2}][\eta_{0,3} + \eta_{2,1}] \\
h_7 &= [3\eta_{2,1} - \eta_{0,3}][\eta_{3,0} + \eta_{1,2}]\{[\eta_{3,0} + \eta_{1,2}]^2 - 3[\eta_{0,3} + \eta_{2,1}]^2\} \\
&\quad + [3\eta_{1,2} - \eta_{3,0}][\eta_{0,3} + \eta_{2,1}]\{3[\eta_{3,0} + \eta_{1,2}]^2 - [\eta_{0,3} + \eta_{2,1}]^2\}
\end{aligned}$$

Das siebte Moment ist jedoch nicht invariant gegenüber Spiegelung. Da der numerische Wert der Hu-Momente stark mit der Ordnung abfällt, wird in Applikationen oft ein logarithmierter Wert berechnet.

$$h_{logi} = \text{sign}(h_i) \log(|h_i|)$$

### 5.4.2 Zernike-Momente

Zernike-Polynome wurden in der theoretischen Optik bereits in den 30er Jahren des letzten Jahrhunderts verwendet [35] [69]. Aber erst 1980 wurden von Teague auf diesen Polynomen basierende orthogonale komplexe Momente entwickelt [59]. Zernike-Polynome besitzen die Form [29]:

$$V_{nm}(x, y) = R_{nm}(\varrho)e^{im\Theta},$$

wobei  $n$  die radiale und  $m$  die azimutale Ordnung des Polynoms beschreibt.  $\varrho$  bestimmt die Länge des Vektors vom Ursprung bis zum Pixel und  $\Theta$  den Winkel zwischen diesem und der  $x$ -Achse. Zernike Polynome lassen sich in einen radialen ( $R_{nm}(\varrho)$ ) und einen winkelabhängigen Teil ( $e^{im\Theta}$ ) separieren. Der radiale Anteil besitzt die Form:

$$R_{nm}(\varrho) = \sum_{s=0}^{n-|m|/2} (-1)^s \cdot \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \varrho^{n-2s}.$$

Es sind komplexe, orthogonale Polynome, die senkrecht auf dem Einheitskreis  $x^2 + y^2 \leq 1$  verlaufen. Die Zernike-Momente entstehen durch die Projektion der Bildfunktion  $B(x, y)$  auf diese orthogonalen Basisfunktionen. Dies geschieht analog zur Konstruktion der geometrischen Momente (vgl. 5.4) nur das die Bildfunktion nicht auf die nicht orthogonale Basis  $x^p y^q$  projiziert wird, sondern auf die

orthogonalen Zernike-Polynome. Mit Hilfe dieser Basisfunktionen ergeben sich folgende Zernike-Momente:

$$A_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2 \leq 1} B(x, y) V_{nm}^*(\varrho, \Theta) dx dy.$$

In Polarkoordinaten werden diese zu:

$$A_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 B(\varrho, \Theta) R_{nm}(\varrho) e^{-im\Theta} \varrho d\varrho d\Theta.$$

Zernike-Momente besitzen die Eigenschaft, dass sie ähnlich der in Abschnitt 5.5 beschriebene Curvature-Scale-Space-Repräsentation unter Rotation des Objektes einen Phasenshift vollführen. In [29] wird die Rotationsinvarianz der Merkmale erreicht, indem nur jeweils der Betrag der Momente genommen wird, da dieser invariant gegenüber einer Phasenverschiebung ist (vgl. 5.5.3). Eine Invarianz gegenüber Skalierung und Translation wird erreicht, indem das Bild vor der Berechnung der Momente mit Hilfe seiner geometrischen Momente normiert wird.

Einen alternativen Ansatz, aus den Zernike-Momenten einen kompletten Satz invarianter Merkmale zu erzeugen, wird in [64] vorgestellt. Dieser Ansatz beruht auf den beiden folgenden Aussagen:

1. Aus jedem Paar komplexer Momente müssen mindestens zwei Invarianten bestimmt werden. Dadurch wird es möglich, die Magnitude und die Phase zurückzugewinnen.
2. Für eine gerade Ordnung  $n$  wird das 0.Moment direkt verwendet, da es keine Phase besitzt und somit rotationsinvariant ist. Es darf nicht der Betrag des Momentes genommen werden, da das Vorzeichen erhalten bleiben muss.

Auf diesen Grundaussagen aufbauend werden drei Regeln definiert, die folgende Invarianten liefern:

$$\begin{aligned} 0. \text{ Ordnung} \quad I_{00} &= A_{00} \\ 1. \text{ Ordnung} \quad I_{10} &= |A_{11}|^2 \\ &I_{01} = |A_{11}|^2 = I_{10} \\ 2. \text{ Ordnung} \quad I_{20} &= |A_{22}|^2 \\ &I_{21} = A_{20} \\ &I_{22} = |A_{22}|^2 = I_{20} \\ 3. \text{ Ordnung} \quad I_{30} &= |A_{33}|^2 \\ &I_{31} = |A_{31}|^2 \\ &I_{32} = |A_{31}|^2 |A_{22}| \cos(2\varphi_{31} - \varphi_{22}) \\ &P_{32} = |A_{31}|^2 |A_{22}| \sin(2\varphi_{31} - \varphi_{22}) \\ &I_{33} = |A_{33}| |A_{31}|^3 \cos(2\varphi_{33} - 3\varphi_{31}) \\ &P_{33} = |A_{33}| |A_{31}|^3 \sin(2\varphi_{33} - 3\varphi_{31}). \end{aligned}$$

## 5.5 Repräsentation durch den Curvature-Scale-Space (CSS)

Die Curvature-Scale-Space Theorie wurde von F. Mokhtarian und A. K. Mackworth als Repräsentationstechnik für planare Kurven eingeführt [43]. Es ist, wie der Name schon sagt, ein krümmungsbasiertes, auf die „scale space“ Theorie aufsetzendes Verfahren.

Die Repräsentation wird durch Faltung einer pfadparametrisierten Kurve mit einer Gaußfunktion erzeugt [42]. Durch das Falten wird die Kurve geglättet. Nach dem Glätten der parametrisierten Kurve wird bestimmt, an welchen Punkten die Krümmung gleich Null ist. Diese Prozedur wird mit einer Gaußfunktion mit monoton wachsender Standardabweichung  $\sigma$  wiederholt, bis die Kurve keine Wendestellen mehr besitzt.

### 5.5.1 Krümmung einer ebenen Kurve

Zwei Punkte einer ebenen Kurve, die durch eine Bogenlänge  $\Delta s$  voneinander getrennt sind, besitzen Tangenten mit im Allgemeinen unterschiedlichem Richtungswinkel. Die Differenz der Richtungswinkel der Tangenten beträgt  $\Delta\alpha$ . Je größer die Änderung der Richtungstangenten ist, desto größer muss die Krümmung der Kurve sein. Aus diesem Grund gibt der Quotient  $\frac{\Delta\alpha}{\Delta s}$  die mittlere Krümmung der Kurve für den Bereich  $\Delta s$  an [34]. Um die Krümmung der Kurve in einem Punkt zu bestimmen, muss nach Gleichung 5.7 der Grenzübergang  $\Delta s \rightarrow 0$  erfolgen:

$$\kappa = \lim_{\Delta s \rightarrow 0} \frac{\Delta\alpha}{\Delta s} = \frac{d\alpha}{ds}. \quad (5.7)$$

Da  $\tan \alpha = y'$  bzw.  $\alpha = \arctan y'$  gilt, folgt:

$$d\alpha = \frac{y''}{1 + y'^2} dx.$$

Mit der Definition der Bogenlänge

$$\begin{aligned} ds^2 &= dx^2 + y^2 \\ &= \left(1 + \left(\frac{dy}{dx}\right)^2\right) dx^2 \\ \Rightarrow ds &= \sqrt{1 + y'^2} dx \end{aligned}$$

folgt für die Krümmung in einem Punkt:

$$\kappa = \frac{y''}{(1 + y'^2)^{\frac{3}{2}}}. \quad (5.8)$$

Da die Konturen als Kurven in Parameterdarstellung vorliegen, wird obiger Ausdruck für die Krümmung dahingehend umgeschrieben.

Zweidimensionale, mit dem Parameter  $u$  parametrisierte Kurven besitzen folgende Form:

$$x = x(u) \quad y = y(u).$$

Die in Gleichung 5.8 benötigte erste und zweite Ableitung der Kurve erhält man durch Anwenden der Kettenregel für parametrisierte Funktionen. Für die Differentiation nach dem Parameter  $u$  gilt:

$$\frac{dx}{du} = \dot{x} \quad \text{und} \quad \frac{dy}{du} = \dot{y}.$$

Somit ergibt sich die erste Ableitung zu:

$$y' = \frac{dy}{dx} = \frac{\dot{x}}{\dot{y}} \quad (5.9)$$

Die zweite Ableitung ergibt sich, indem  $y'$  nach  $u$  und  $u$  nach  $x$  differenziert wird.

$$y'' = \frac{d(\frac{\dot{y}}{\dot{x}})}{du} \cdot \frac{du}{dx} = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\dot{x}^2} \cdot \frac{1}{\dot{x}} = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\dot{x}^3} \quad (5.10)$$

Einsetzen von Gleichung 5.9 und 5.10 in Gleichung 5.8 ergibt die Krümmung einer parametrisierten Kurve im Punkt  $u$ .

$$\begin{aligned} \kappa(u) &= \frac{\dot{x}(u)\ddot{y}(u) - \dot{y}(u)\ddot{x}(u)}{(1 + (\frac{\dot{y}(u)}{\dot{x}(u)})^2)^{\frac{3}{2}}\dot{x}^3} \\ &= \frac{\dot{x}(u)\ddot{y}(u) - \dot{y}(u)\ddot{x}(u)}{(\dot{x}(u)^2 + \dot{y}(u)^2)^{\frac{3}{2}}}. \end{aligned}$$

Wird anstatt der Parametrisierung  $u$  der normalisierte Bogenlängenparameter  $w$  verwendet, vereinfacht sich die Krümmung zu:

$$\kappa(w) = \dot{x}(w)\ddot{y}(w) - \dot{y}(w)\ddot{x}(w).$$

Da eine CSS-Repräsentation durch Falten der parametrisierten Kurve mit einer Gaussfunktion entsteht, muss eine Darstellung für die Krümmung in Abhängigkeit der Parametrisierung  $u$  und der Standardabweichung der Gaußkurve  $\sigma$  gefunden werden. Dazu wird eine Kurve  $\Gamma$  mit der normalisierten Bogenlänge  $w$  parametrisiert:

$$\Gamma = (x(w), y(w)).$$

Nach der Faltung mit der Gaussfunktion besitzt die Kurve die Form:

$$\Gamma_\sigma = ((X(u, \sigma), Y(u, \sigma)) \quad \text{mit} \quad \begin{aligned} X(u, \sigma) &= x(u) \otimes g(u, \sigma) \\ Y(u, \sigma) &= y(u) \otimes g(u, \sigma) \end{aligned}$$

wobei  $\otimes g(u, \sigma)$  die Faltung mit einer Gaussfunktion mit Standardabweichung  $\sigma$  beschreibt. Analog zur Herleitung für die parametrisierte Kurve ergibt sich für die Krümmung der gefalteten Kurve:

$$\kappa(u, \sigma) = \frac{X_u(u, \sigma)Y_{uu}(u, \sigma) - X_{uu}(u, \sigma)Y_u(u, \sigma)}{(X_u(u, \sigma)^2 + Y_u(u, \sigma)^2)^{1.5}}$$

wobei

$$X_u(u, \sigma) = \frac{\partial}{\partial u}(x(u) \otimes g(u, \sigma)) = x(u) \otimes g_u(u, \sigma)$$

$$X_{uu}(u, \sigma) = \frac{\partial^2}{\partial u^2}(x(u) \otimes g(u, \sigma)) = x(u) \otimes g_{uu}(u, \sigma)$$

und entsprechend

$$Y_u(u, \sigma) = y(u) \otimes g_u(u, \sigma)$$

$$Y_{uu}(u, \sigma) = y(u) \otimes g_{uu}(u, \sigma)$$

Die CSS-Repräsentation ergibt sich, indem die Lösung der Gleichung  $\kappa(u, \sigma) = 0$  mit  $u$  als Ordinate und  $\sigma$  als Abszisse aufgetragen wird. Abbildung 5.4 macht die Entstehungsweise eines CSS-Bildes deutlich.

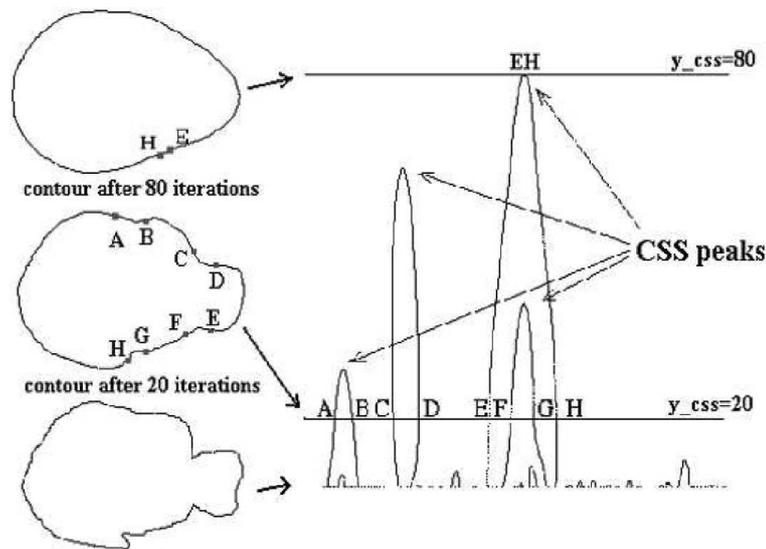


Abbildung 5.4: Entstehungsweise der CSS-Repräsentation aus einer Kontur [44].

### 5.5.2 Eigenschaften der CSS-Repräsentation

Die CSS-Repräsentation erfüllt das Kriterium der Invarianz (vgl. 5.1), da eine uniforme Skalierung der Kontur nur eine uniforme Skalierung der Repräsentation bewirkt. Bei einer geschlossenen Kontur ist es möglich, die CSS-Repräsentationen zu normalisieren und somit eine Invarianz bzgl. uniformer Skalierung zu erreichen. Ferner erzeugt eine Rotation der Kontur einen horizontalen Shift der Repräsentation. Diese Translation kann mittels geeigneter Matching-Verfahren ausgeglichen werden (vgl. 5.5.3).

Die CSS-Repräsentation ist eindeutig, da es möglich ist, eine Kontur aus ihrer CSS-Repräsentation zurückzugewinnen. Des Weiteren ist eine CSS-Repräsentation stabil, da sie unter kleinen Veränderungen der Kontur sich ebenfalls nur geringfügig verändert. Dies resultiert daraus, dass geringfügige Änderungen der Kontur schon durch einen Gaussfilter mit geringem  $\sigma$  geglättet werden. Deshalb entstehen durch Rauschen nur niedrige Peaks in der CSS-Repräsentation. Werden beim Matching-Verfahren nur Maxima verwendet, die über einem definierten Schwellwert liegen, wird die Repräsentation damit stabil gegenüber Rauschen.

Zur Erzeugung der CSS-Repräsentation werden nur endliche Gauss-Filter und endliche Nachbarschaften des jeweils geglätteten Punktes  $u$  der Kurve betrachtet. Aus diesem Grund kann die CSS-Repräsentation auch für geöffnete Konturen berechnet werden. Dadurch ist es möglich, Konturen zu erkennen, die Teil einer anderen Kontur sind. Ebenfalls ist es möglich, Konturen zu erkennen, die teilweise verdeckt sind. Somit erfüllt die CSS-Repräsentation das Kriterium der lokalen Information.

### 5.5.3 Merkmalsextraktion mit der Eigen-CSS Methode

Die Eigenschaft der CSS-Repräsentation, dass aus einer Rotationsbewegung eine Translationsbewegung der Repräsentation folgt, stellt eine technische Herausforderung an das Matching-Verfahren dar. Drew, Lee und Rova schlagen ein Verfahren vor, dieses Translationsproblem durch eine Kombination aus drei einfachen, im Bereich des maschinellen Sehens (engl.: *computer vision*) etablierten

Methoden zu erreichen: Marginale-Summen, Phasenkorrelation und Single-Value-Decomposition [11]. Durch Marginale-Summen und Phasenkorrelation wird das Translationsproblem gelöst und Single-Value-Decomposition überführt die rotationsinvariante Darstellung in ihren Eigenraum. Drew, Lee und Rova definieren diese Repräsentation als Eigen-CSS Darstellung. Diese Eigenraumdarstellung kann direkt als Merkmalsvektor genutzt werden.

### Marginale-Summe

Da die Translationsbewegung der CSS-Bilder parallel zur Parametrisierung  $u$  gerichtet ist, wird das Bild durch Anwendung von Marginalen-Summen in eine rotationsunabhängige Zeilendarstellung  $\mathbf{c}$  und eine rotationsabhängige Spaltendarstellung  $\mathbf{r}$  aufgeteilt [11].

Die Komponenten dieser Spalten- und Zeilenvektoren entstehen durch Aufsummieren der gesetzten Pixel in der jeweiligen Zeile bzw. Spalte. Gleichungen 5.11 und 5.12 verdeutlichen das Vorgehen, wobei  $C(i, j)$  das Pixel in der  $i$ -ten Zeile und der  $j$ -ten Spalte bezeichnet.

$$\mathbf{x} = [\mathbf{r} \ \mathbf{c}]^T$$

$$\mathbf{r} = \left[ \sum_i C(i, 1), \sum_i C(i, 2), \dots, \sum_i C(i, c) \right]^T \quad (5.11)$$

$$\mathbf{c} = \left[ \sum_j C(1, j), \sum_j C(2, j), \dots, \sum_j C(r, j) \right]^T \quad (5.12)$$

### Phasenkorrelation

Das Phasenkorrelations-Verfahren wurde bereits Anfang der 70er Jahre eingesetzt, um gegeneinander translatierte Bilder miteinander zu vergleichen. Das Verfahren beruht auf dem Verschiebungssatz der Fouriertransformation, der aussagt, dass eine Verschiebung im Orts- bzw. Frequenzraum sich im korrespondierenden Bereich nur durch einen Phasenfaktor äußert [1].

Gegeben seien zwei Funktionen im Ortsraum  $f_1(x, y)$  und  $f_2(x, y)$ . Da sich die Funktionen nur durch eine Translation unterscheiden sollen, gilt:

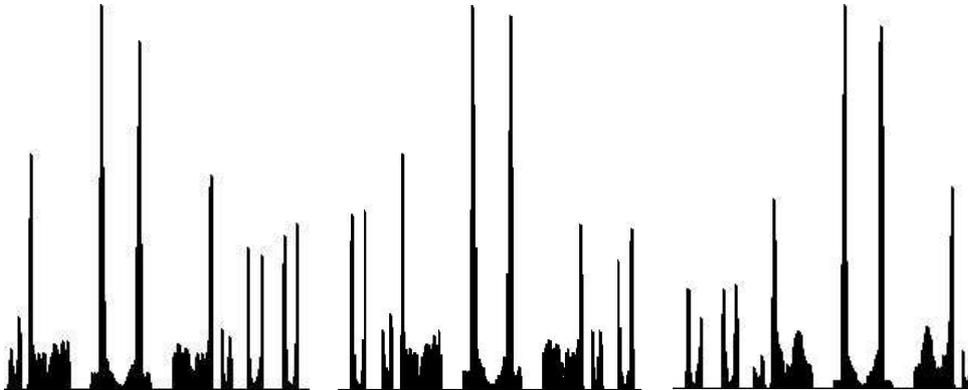
$$f_2(x, y) = f_1(x + x_0, y + y_0).$$

Mit dem Verschiebungssatz folgt:

$$F_2(u, v) = F_1(u, v)e^{i(ux_0 + vy_0)},$$

wobei  $F_2(u, v)$  und  $F_1(u, v)$  die Fouriertransformierten von  $f_1(x, y)$  bzw.  $f_2(x, y)$  sind. Wird hieraus das Cross power Spektrum gebildet, wird die Phase isoliert.

$$\frac{F_2(u, v)F_1(u, v)^*}{|F_2(u, v)F_1(u, v)^*|} = e^{i(ux_0 + vy_0)}$$



**Abbildung 5.5:**  $\mathbf{r}$ -Vektoren einer (a) um 90 Grad nach links, (b) nicht rotierten und (c) 90 Grad nach rechts rotierten Kontur eines Menschen

Die Phasenkorrelationsfunktion ist als inverse Fouriertransformation der Kreuz-Korrelation definiert:

$$\begin{aligned} f_{PCF}(f_1(x, y), f_2(x, y)) &= F^{-1} \left( \frac{F_2(u, v) F_1(u, v)^*}{|F_2(u, v) F_1(u, v)^*|} \right) \\ &= F^{-1}(e^{i(ux_0 + vy_0)}) \\ &= \delta(x_0, y_0). \end{aligned}$$

Somit entsteht bei der Korrelation zweier Bilder ein Korrelationsspektrum mit nur einem Peak, dessen Koordinaten der Verschiebung des Bildes in  $x$ - bzw.  $y$ -Richtung entsprechen. Es ist möglich, durch Erweiterungen dieses Ansatzes neben der Translation auch Rotation und Skalierung zu bestimmen, die zwei Bilder ineinander überführt.

In dieser Applikation wird jedoch nur die Translation in  $x$ -Richtung benötigt, und es sind nicht zwei Bildfunktionen  $f_1$  und  $f_2$  zu korrelieren, sondern nur der Spaltenvektor  $\mathbf{r}$ . Die beschriebene Phasenkorrelation kann aus dem Grund nicht angewendet werden. Um den gewünschten Effekt der Rotationsinvarianz zu erzielen, reicht es aus, das Verschiebungstheorem der Fouriertransformation anzuwenden. Dieses besagt, dass die Magnitudenspektren zweier translaterter Bildfunktionen identisch sind. Somit ergibt sich folgende Vorgehensweise:

1. Berechnen der diskreten Fouriertransformation des  $\mathbf{r}$ -Vektors
2. Berechnen des Magnitudenspektrums
3. Rücktransformation des Magnitudenspektrums

$$\begin{aligned} \tilde{\mathbf{r}} &= |F^{-1}(|F(\mathbf{r})|)| \\ \mathbf{x} &= [\tilde{\mathbf{r}}\mathbf{c}]^T \end{aligned}$$

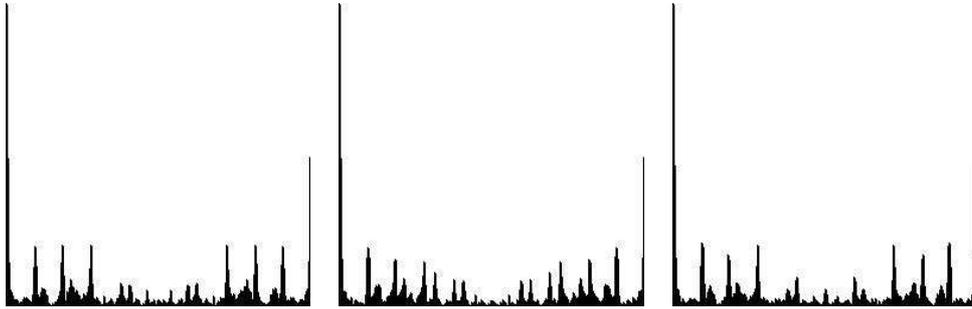


Abbildung 5.6: Mittels Phasenkorrelation erstellte  $\tilde{r}$ -Vektoren der  $r$ -Vektoren aus Abbildung 5.5(a-c)

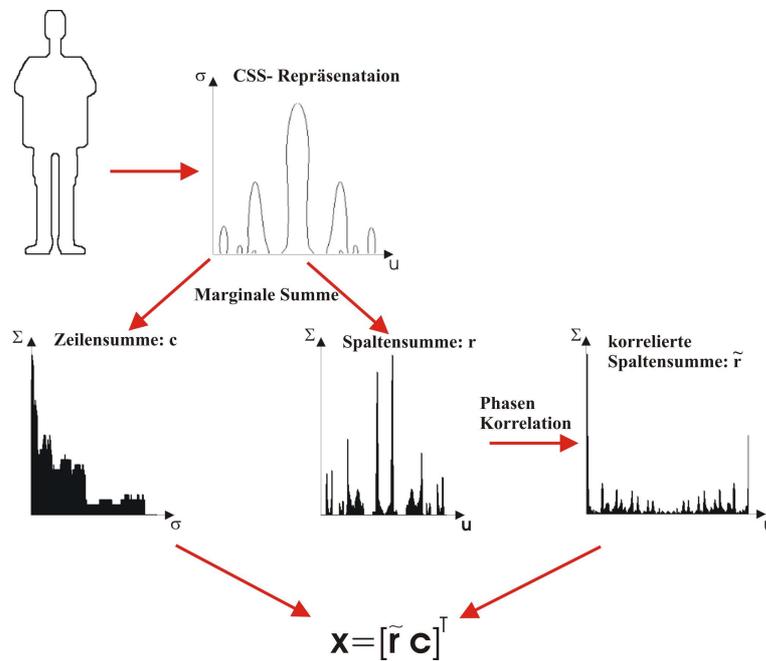


Abbildung 5.7: Erstellung des Merkmalsvektors einer Kontur mit dem Eigen-CSS Verfahren

### Single-Value-Decomposition

Um den aus der CSS-Repräsentation erstellten  $\mathbf{x}$ -Vektor in einen Eigenraum abzubilden, muss eine neue Basis aus orthonormalen Basisvektoren gefunden werden.

Die algebraisch formulierte Problemstellung lautet:

Es gibt  $N$  Vektoren in einem  $M$ -dimensionalen Vektorraum mit  $N \leq M$ . Nun soll ein orthonormaler Satz von  $N$  Vektoren gefunden werden, der den selben Vektorraum aufspannt wie die ursprünglichen  $N$  Vektoren. Dieses Problem kann mit Hilfe der Single-Value-Decomposition (SVD) gelöst werden. Dazu wird eine  $N \times M$  dimensionale Matrix  $\mathbf{X}$  aus den  $N$  Vektoren erstellt. Diese Matrix lässt sich als Multiplikation einer spaltenorthonormalen  $N \times M$  Matrix  $\mathbf{U}$ , einer  $N \times N$  Diagonalmatrix  $\mathbf{W}$  (Singular Values) und der Transponierten einer  $N \times N$  orthogonalen Matrix  $\mathbf{V}$

schreiben [49].

Wenn der Rang der  $\mathbf{X}$ -Matrix nicht ihrer Dimensionalität entspricht, wird der durch  $\mathbf{X}$  aufgespannte Vektorraum in zwei Unterräume geteilt: In Nullspace und Range. Der Nullspace bildet einen Vektor immer auf Null ab. Range besitzt die Dimension des Ranges von  $\mathbf{X}$  und beschreibt den durch Multiplikation mit  $\mathbf{X}$  erreichbaren, von Null verschiedenen, Vektorraum. Mit Hilfe der SVD-Operation kann eine orthonormale Basis für beide Vektorräume gefunden werden. Dazu werden die Einträge  $w_j$  der  $\mathbf{W}$ -Matrix betrachtet. Die Orthonormalbasis des Range bilden die Spalten  $j$  der  $\mathbf{U}$ -Matrix, deren  $w_j$  von Null verschieden sind. Die Orthonormalbasis des Nullraums bilden die Spalten  $j$  der  $\mathbf{V}$ -Matrix, deren  $w_j$  gleich Null sind.

$$\mathbf{X} = \mathbf{U} \begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{pmatrix} \mathbf{V}^T$$

Folgende Aufzählung beschreibt die erforderlichen Schritte, um die von einem Objekt erstellten  $\mathbf{x}$ -Vektoren in ihren Eigenraum abzubilden.

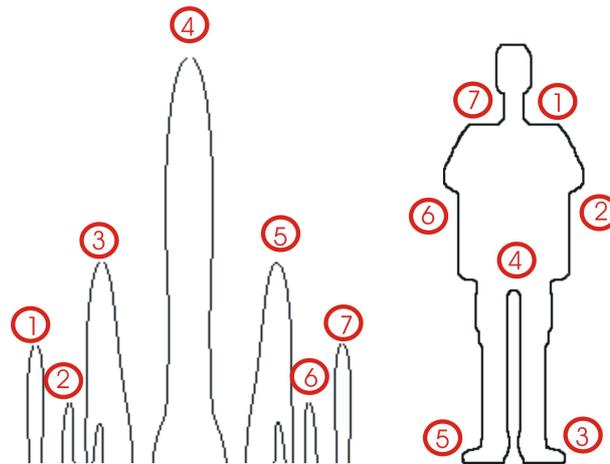
1. Bestimme für eine feste Anzahl von Beispielen (von jedem zu klassifizierenden Objekt) die  $\mathbf{x}$ -Vektoren.
2. Subtrahiere von jedem  $\mathbf{x}$ -Vektor seinen Durchschnittswert.
3. Erstelle die  $\mathbf{X}$ -Matrix aus den subtrahierten  $\mathbf{x}$ -Vektoren nach Gleichung 5.13. Jeder  $\mathbf{x}$ -Vektor entspricht einer Spalte der  $\mathbf{X}$ -Matrix.
4. Durchführung der SVD
5. Reduziere die  $\mathbf{U}$ -Matrix auf die Spalten  $j$ , deren  $w_j$  ungleich Null sind.
6. Projiziere die  $\mathbf{x}$ -Vektoren in ihren Eigenraum durch Multiplikation mit der transponierten  $\mathbf{U}$ -Matrix nach Gleichung 5.14.

$$\mathbf{X} = [\mathbf{x}_1 - \bar{\mathbf{x}}_1, \mathbf{x}_2 - \bar{\mathbf{x}}_2, \dots, \mathbf{x}_n - \bar{\mathbf{x}}] \quad (5.13)$$

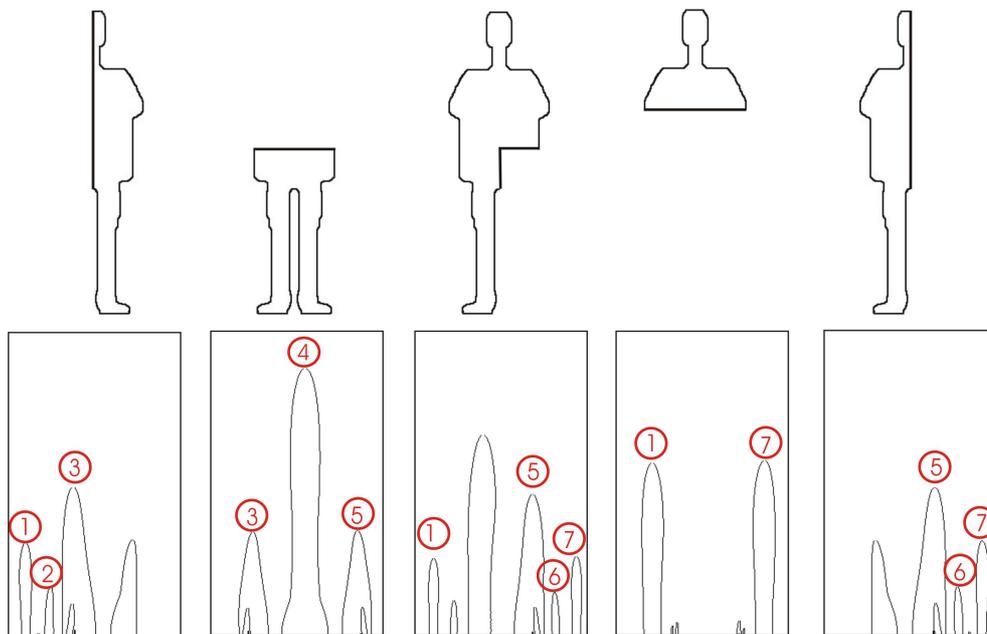
$$\mathbf{u} = \mathbf{U}^T \mathbf{x} \quad (5.14)$$

#### 5.5.4 CSS-Repräsentation zur Beschreibung von teilweise verdeckten Objekten

Obwohl in dieser Arbeit das CSS-Verfahren nicht zur Erkennung von teilweise verdeckten Objekten verwendet wird, soll in diesem Abschnitt dessen Potential zur Repräsentation eben solcher Konturen beschrieben werden. Durch die Verwendung von lokalen Informationen einer Kontur, ist die CSS-Repräsentation geeignet, um teilweise verdeckte Objekte erkennen zu können. Abbildung 5.8 zeigt die Kontur eines Menschen mit zugehöriger CSS-Repräsentation. Das Bild macht deutlich, dass jedes Maximum der CSS-Repräsentation eindeutig einem bestimmten Teil der Kontur zugeordnet werden kann. Werden Teile der Kontur überdeckt, verschwinden nur die Maxima, die aus Bereichen der verdeckten Kontur entstanden sind. Die restlichen Maxima bleiben erhalten. Dies ist in Abbildung 5.9 verdeutlicht. Im Vergleich mit Abbildung 5.8 ist zu sehen, dass eine Teilkontur die selben Maxima erzeugt, wie die entsprechenden Teilbereiche der nicht verdeckten Originalkontur. Diese Maxima können durch ein geeignetes Matching-Verfahren mit der CSS-Repräsentation



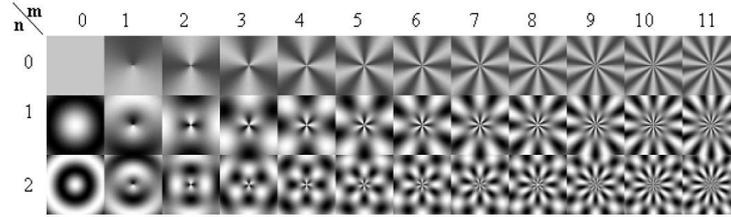
**Abbildung 5.8:** CSS-Darstellung der Kontur eines stehenden Menschen



**Abbildung 5.9:** CSS-Darstellung von verschiedenen Teilkonturen eines Menschen

der gesamten Kontur verglichen werden. Es ist somit möglich, die verdeckte Kontur als Teil der Originalkontur zu klassifizieren.

Das Matching-Verfahren für CSS-Repräsentationen verdeckter Objekte kann nicht den einfachen Ansatz verfolgen, die Maxima der Größe nach jeweils paarweise zu vergleichen, wie es bei nicht verdeckten Objekten möglich ist. Diese Vorgehensweise ist nicht möglich, da zum Beispiel das größte Maximum der nicht verdeckten Kontur in der verdeckten Kontur evtl. nicht vorhanden ist. Nach dem Matching-Verfahren für nicht verdeckte Objekte würde das größte Maximum der verdeckten Kontur mit dem größten Maximum der nicht verdeckten Kontur verglichen. Entsprechendes gilt



**Abbildung 5.10:** Darstellung der ersten 36 Basisfunktionen der ART-Repräsentation [51].

für die folgenden kleineren Maxima. Es ergibt sich somit ein großer Matchingfehler, obwohl die Maxima exakt übereinstimmen, sobald das größte Maximum der nicht verdeckten Kontur ignoriert wird.

Mokhtarian schlägt zum Matchen von verdeckten Konturen ein Verfahren vor, das jede Kontur in Teilkonturen zerschneidet, die jeweils von Nulldurchgängen der Krümmung begrenzt sind [39]. Diese Teilkonturen werden einzeln verglichen.

## 5.6 Angular-Radial-Transformation (ART)

Die ART-Repräsentation ist eine regionbasierte Formrepräsentation und bildet mit der konturorientierten CSS-Repräsentationstechnik die beiden Formbeschreibungstechniken des MPEG-7 Standards [6]. Die ART-Repräsentation beschreibt die Form eines Objektes kompakt und effizient. Außerdem ist sie invariant gegenüber Rotation und Verzerrung der Objektform.

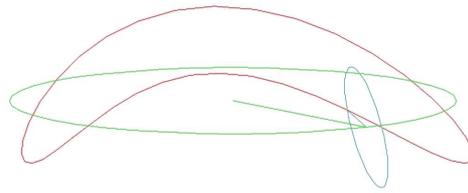
Die ART-Repräsentation eines Objektes besteht aus Koeffizienten  $F_{nm}$ , die sich gemäß Gleichung 5.15 berechnen lassen. Diese Koeffizienten entsprechen den in Abschnitt 5.4.2 beschriebenen Zernike-Momenten. Der Unterschied zwischen Zernike-Momenten und der ART-Repräsentation ist die Wahl der Basisfunktionen, die zur Berechnung eingesetzt werden.

Analog zu den Zernike-Momenten bezeichnen  $V_{nm}(\varrho, \Theta)$  die Basisfunktionen der ART-Repräsentation. Sie besitzen die Ordnung  $n$  und  $m$  und lassen sich, wie in Gleichung 5.16 zu sehen ist, in einen radialen und einen winkelabhängigen Teil separieren. Die Variable  $f(\varrho, \Theta)$  ist ein binäres Bild in Polarkoordinaten, welches das zu untersuchende Objekt (oder auch mehrere) enthält. Die ART-Repräsentation lässt sich in abgewandelter Form auch auf Graustufen und Farbbilder anwenden [18][51].

$$\begin{aligned} F_{nm} &= \langle V_{nm}(\varrho, \Theta), f(\varrho, \Theta) \rangle \\ &= \int_0^{2\pi} \int_0^1 V_{nm}^*(\varrho, \Theta) f(\varrho, \Theta) \varrho d\varrho d\Theta \end{aligned} \quad (5.15)$$

$$V_{nm}(\varrho, \Theta) = \frac{1}{2\pi} e^{im\Theta} R_n(\varrho) \quad (5.16)$$

$$R_n(\varrho) = \begin{cases} 1, & n = 0 \\ 2 \cos(\pi n \varrho), & n \neq 0 \end{cases} \quad (5.17)$$



**Abbildung 5.11:** Beschreibung einer Kontur (rot) durch eine Fourierellipse erster (grün) und zweiter Ordnung (blau).

<http://lmb.informatik.uni-freiburg.de/lectures/mustererkennung/WS0405/FourierDemo/fourdem.html>

Im MPEG-7 Standard wird empfohlen, die Magnitude der ersten 35 Koeffizienten zur Beschreibung der Objektform heranzuziehen, wobei der erste Koeffizient nicht berücksichtigt werden soll, da dieser zum Normalisieren der restlichen Koeffizienten eingesetzt wird. Es werden drei radiale und zwölf winkelabhängige Funktionen eingesetzt ( $n \leq 3$  und  $m \leq 12$ ), um die Koeffizienten zu bestimmen. Abbildung 5.10 zeigt diese Funktionen.

## 5.7 Repräsentation durch Fourier-Deskriptoren

Die Repräsentation von Konturen durch Fourier-Deskriptoren ist eine häufig verwendete globale Repräsentationstechnik.

Geschlossene Konturen bilden in der komplexen Ebene periodische Funktionen mit der Periode  $T$ . Für parametrisierte Kurven kann diese Beziehung durch  $x(u) = x(u + kT)$  ausgedrückt werden. Periodische Funktionen lassen sich durch Fourierreihen darstellen. Diese werden nach Gleichung 5.18 berechnet und überführen die Kontur vom Orts- in den Frequenzraum.

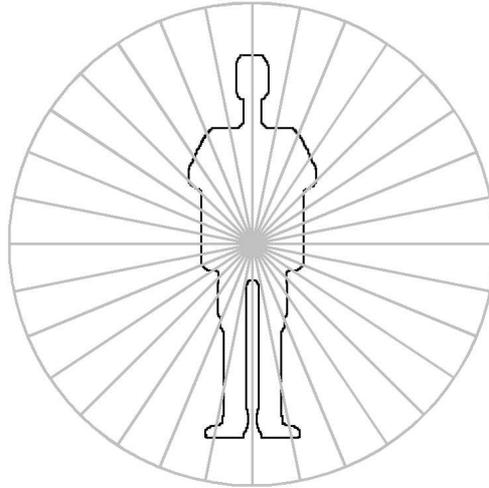
$$x(u) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega u} \quad \text{mit} \quad \omega = \frac{2\pi}{T} \quad \text{und} \quad c_n = \frac{1}{T} \int_0^T x(u) e^{-jn\omega u} du \quad (5.18)$$

Die Kontur kann durch die Summe der Fourierkoeffizienten  $c_n$  beschrieben werden. Da die Repräsentation globalen Charakter besitzt, verfügt jeder Fourierkoeffizient über Informationen der gesamten Kontur. Geschlossenen Konturen sind stetig und beschränkt. Die Fourierkoeffizienten einer solchen Kontur streben  $\sim \frac{1}{n^2}$  gegen Null. Zur Repräsentation der Kontur müssen daher keine unendlichen Summen ausgeführt werden, da sich die Kontur schon durch wenige (niederfrequente) Koeffizienten approximieren lässt.

Eigenschaften der Fourierkoeffizienten bezüglich Konturtransformationen:

**Rotation:** Eine rotierte Kontur im Ortsraum führt nur zu einer Phasenveränderung im Frequenzraum (vgl. 5.5.3). Das Magnitudenspektrum bleibt konstant.

**Translation:** Eine translatierte Kontur besitzt andere Fourierkoeffizienten als die Originalkontur. Aus diesem Grund wird der Ursprung der Komplexen Ebene häufig in den Schwerpunkt der Kontur gelegt bzw. die Kontur wird mit dem ersten Fourierkoeffizienten normiert, der die Lage des Schwerpunktes angibt.



**Abbildung 5.12:** Diskretisierung des zweidimensionalen Raums in radiale Segmente beim Border-Signature-Verfahren.

Skalierung: Eine Skalierung der Kontur im Ortsraum führt zu einer Skalierung der Fourierkoeffizienten im Frequenzraum mit dem selben Skalierungsfaktor. Durch eine Normierung der Fourierkoeffizienten wird die Kontur somit ebenfalls auf eine Einheitsgröße normiert.

Die normierten Fourierkoeffizienten werden Fourier-Deskriptoren genannt. Die Wirkungsweise der Fourier-Deskriptoren kann anschaulich durch Fourierelipsen beschrieben werden. Eine Fourierreihe mit einem Koeffizienten beschreibt einen Kreis. Zusammen mit dem entsprechendem negativem Koeffizienten ergibt sich eine Ellipse. Werden weitere Koeffizientenpaare verwendet, entstehen weitere Ellipsen, die als Ursprung den Zeiger der Ellipse besitzen, die die nächst kleinere Ordnung besitzt. Abbildung 5.11 zeigt, wie die Kontur (rot) durch zwei Fourierelipsen beschrieben wird.

## 5.8 Border-Signature

Das Border-Signature-Verfahren arbeitet ähnlich wie das in 2.2.3 beschriebene Shape-Context-Verfahren. Der zweidimensionale Raum in dem sich die zu beschreibende Kontur befindetet wird in radiale Segmente gleicher Größe aufgeteilt. Wie in Abbildung 5.12 zu sehen ist, besitzen alle Segmente einen gemeinsamen Ursprung der in dieser Arbeit auf den Schwerpunkt der Kontur gesetzt wird. Die konturbeschreibenden Merkmale können nun auf zwei verschiedene Arten aus dieser Darstellung gewonnen werden. Die erste Methode durchläuft ausgehend von einem definierten Startpunkt jedes Segment im Uhrzeigersinn und berechnet den Flächenanteil, den die Kontur in jedem Segment einnimmt. Alle Flächenanteile werden anschließend über die Gesamtfläche normiert. Die zweite Methode betrachtet die Entfernungen, die die Randpunkte der Kontur im jeweiligen Segment besitzen. Als Merkmal wird die mittlere Entfernung aller Punkte pro Segment berechnet. Bei beiden Methoden entspricht die Anzahl der Segmente der Anzahl der extrahierten Merkmale. In dieser Arbeit werden 32 Segmente und die Merkmalsextraktion verwendet, die die Entfernungen der Randpunkte betrachtet.

# Kapitel 6

## Klassifikation

Ziel der Klassifikation ist die Zuordnung der aus dem Tiefenbild extrahierten Objekte zu verschiedenen Klassen. Die Basis der Klassifikation bilden die Merkmalsvektoren, welche die Objekte beschreiben. Diese Merkmalsvektoren werden aus den extrahierten Merkmalen gebildet und bilden die Objekte in einen Merkmalsvektorraum ab, dessen Dimensionalität der Anzahl der extrahierten Merkmale entspricht. Die extrahierten Merkmale sollten für Objekte der selben Klasse ähnlich sein (vgl. Kapitel 5). Somit kann die Annahme gemacht werden, dass Objekte der gleichen Klasse Cluster im Merkmalsvektorraum bilden. Das Problem der Objektklassifikation ist somit ein Clusteringproblem der Merkmalsvektoren im Merkmalsvektorraum.

### 6.1 Merkmalsauswahl

Der erste Schritt der Klassifikation ist eine Dimensionsreduktion des Merkmalsvektorraums. Die in Kapitel 5 vorgestellten Merkmale bilden nur eine kleine Auswahl der Formmerkmale. Da nicht bekannt ist, welche Merkmale gut für ein bestimmtes Problem geeignet sind, werden zunächst viele Merkmale extrahiert. Entgegengesetzt zur intuitiven Annahme muss das Klassifikationsergebnis in einem hochdimensionalen Merkmalsvektorraum nicht zwingend besser als in einem Merkmalsvektorraum sein, der nur von wenigen Merkmalen aufgespannt wird. Aus diesem Grund muss nach der Merkmalsextraktion eine Selektion der relevanten Merkmale erfolgen. Diese Selektion bestimmt durch statistische Verfahren, wie stark die Merkmale der Merkmalsvektoren korreliert sind. Die erhaltene Auswahl der Merkmale sollte optimaler Weise weit voneinander entfernte Clustermittelpunkte für die einzelnen Klassen bilden, um die der Klasse zugehörigen Merkmalsvektoren nur eine geringe Streuung besitzen.

Eine Dimensionsreduktion des hochdimensionalen Vektorraums besitzt ferner den Vorteil, dass der Rechenaufwand bei der Merkmalsextraktion und Klassifikation im reduzierten Merkmalsvektorraum geringer ist. Da es oft Echtzeitanwendungen sind, in deren Rahmen klassifiziert werden muss, ist dies ein nicht zu unterschätzender Faktor.

Zu beachten ist, dass die Dimensionalität des Merkmalsvektorraums und die Anzahl der Testbeispiele nicht die selbe Größenordnung besitzen darf, da dadurch eine Separation der unterschiedlichen Klassen trivial wird. Das würde dazu führen, dass die Testbeispiele „auswendig“ gelernt werden und der Klassifikator seine Generalisierungsfähigkeit verliert [1]. Dieser Effekt wird „Overfitting“ genannt. Derselbe Effekt tritt auf, wenn kein repräsentativer Trainingssatz ausgewählt wird. Der

Klassifikator orientiert sich beim Lernen an Eigenheiten des Trainingsatzes, die nicht repräsentativ für die gesamte Datenmenge sind und liefert in der Klassifikationsphase schlechtere Ergebnisse [13]. Ein Selektionsverfahren muss ein Gütekriterium für die extrahierten Merkmale und ein Auswahlverfahren definieren, um einen optimalen Merkmalsatz zu finden [1]. Im Folgenden werden zwei Auswahlverfahren vorgestellt.

### 6.1.1 plus $l$ -take away $r$ Algorithmus

Der „plus  $l$ -take away  $r$ “ Algorithmus ist eine Kombination aus sequentieller Vorwärtsselektion (SFS) und sequentieller Rückwärtsselektion (SBS). Die SFS-Operation startet zunächst mit einem leeren Merkmalsvektor und fügt in jeder Iteration den jeweils signifikantesten Vektor hinzu, bis eine definierte Anzahl von Merkmalen erreicht ist. Die SBS-Operation startet mit einem aus allen Merkmalen bestehenden Merkmalsvektor und entfernt in jeder Iteration das jeweils am wenigsten signifikante Merkmal. Wie signifikant die Merkmale sind, wird durch Vergleichsprüfung (engl.: *cross validation*) entschieden. Da die Operationen nicht die Möglichkeit besitzen, die Entscheidung über die Signifikanz der Merkmale in späteren Iterationsschritten eventuell zu korrigieren, ist das Ergebnis oft suboptimal [50]. Der „plus  $l$ -take away  $r$ “ Algorithmus wendet in einem Iterationsschritt  $l$ -fach die SFS-Operation und  $r$ -fach die SBS-Operation an. Somit ist durch die Wahl von  $l$  und  $r$  eine Möglichkeit des „Zurücknehmens“ (engl.: *backtracking*) gegeben, so dass keine suboptimalen Merkmale in den resultierenden Merkmalsvektor gelangen.

Eine Erweiterung des „plus  $l$ -take away  $r$ “ Algorithmus ist der „sequential floating forward search“ (SFFS)-Algorithmus, der die Möglichkeit des automatisierten Zurücknehmens von Merkmalen bietet. Der SFFS-Algorithmus lässt einer SFS-Operation so lange SBS-Operationen folgen, wie das erhaltene Ergebnis besser ist als vorherige. Es gibt also keine Rückwärtsschritte, falls die Performance des Klassifikators nicht erhöht werden kann. Somit wird das Zurücknehmen der Merkmale dynamisch gesteuert [50].

### 6.1.2 Principal-Component-Analysis (PCA)

Da der reduzierte Merkmalsatz die größtmögliche Varianz des hochdimensionalen Merkmalsatzes erfassen soll, ist PCA ein geeignetes Mittel, den Merkmalsvektorraum zu verkleinern. Die PCA erstellt signifikantere Merkmale als Linearkombination der ursprünglichen Merkmale. Dazu wird ein Satz von Datenpunkten genommen und ein niedrig dimensionaler Unterraum der Merkmalsvektoren konstruiert, der die Variation dieser Datenpunkte um ihren Mittelwert am besten erklärt [15]. Diese Vorgehensweise wird auch Karhunen-Loève-Transformation genannt.

## 6.2 Klassifikationsstrategien

Es gibt zwei unterschiedliche Herangehensweisen, einen Klassifikator zu trainieren.

Unüberwachtes Lernen: Unüberwachtes Lernen bezeichnet die Klassifikation von Daten ohne im voraus bekannte Zielwerte. Zu den unüberwachten Lernverfahren gehören Clustering-Verfahren. Bei unüberwachtem Lernen werden keine Aussagen darüber gemacht, zu welcher Klasse ein Merkmalsvektor gehört, sondern nur, welche Merkmalsvektoren zur selben Klasse gehören.

Ein Beispiel für ein Clustering-Verfahren ist die  $k$ -Nächste-Nachbarn (engl.: *k-Nearest-Neighbor*) Technik.

Überwachtes Lernen: Bei überwachtem Lernen sind für eine definierte Menge von Merkmalsvektoren (Trainingsmenge) die Klasse bekannt, zu der die durch die Merkmalsvektoren repräsentierten Objekte gehören. Mit diesen Merkmalsvektoren wird der Klassifikator trainiert. Bei der Auswahl der Testbeispiele der Trainingsmenge ist darauf zu achten, dass diese homogen über die Klasse verteilt sind und diese gut repräsentieren. Nach dem Training soll der Klassifikator in der Lage sein, unbekannte Eingaben anhand der Merkmalsvektoren richtig zu klassifizieren. Neben der Trainingsmenge gibt es eine Testmenge, anhand der die Performanz des Klassifikators getestet wird. Mit dieser soll „Overfitting“ verhindert werden. Zu diesem Zweck wird häufig der Punkt, bei dem der Fehler auf der Testmenge nicht mehr sinkt, sondern wieder größer wird, als Abbruchpunkt des Trainingsvorgangs genommen. Dadurch wird die Testmenge in die Trainingsphase involviert, so dass eine dritte Validierungsmenge eingeführt wird, die den Abbruch des Lernvorgangs bestimmt, damit die Testmenge unabhängig vom Lernprozess die Performanz des Klassifikators bestimmen kann.

Ein Beispiel für überwachte Lernverfahren sind Neuronale Netze. Sie bestehen aus einzelnen Schaltern den Neuronen und Verbindungen zwischen den Neuronen, den Synapsen [5]. In einer Netzstruktur sind die Neuronen in Schichten organisiert. Jedes Neuron besitzt beliebig viele, gewichtete Eingaben. Durch den Vergleich der gewichteten Eingaben mit einem Schwellwert bestimmen die Neuronen ihren Ausgabewert. Die Ausgabe des Neurons kann als Eingabe weiterer Neuronen dienen, so dass ein Netz entsteht. Ein neuronales Netz wird trainiert, indem die Gewichte und Schwellwerte der einzelnen Neuronen im Netz so angepasst werden, dass sie für die Trainingsbeispiele die richtige Ausgabe liefern. Natürlich muss auch hier das Problem des „Overfitting“ beachtet werden.

## 6.3 Klassifikation mittels Support-Vector-Machines (SVM)

### 6.3.1 Primäre Form

Das Prinzip der Support-Vector-Machines ist ein von Vladimir Vapnik entwickeltes überwachtes Lernverfahren zum Lösen von Klassifikationsproblemen [61]. Es sucht eine optimale Hyperebene, um zwei Klassen mit positiven und negativen Beispielen  $\mathbf{x}$  voneinander zu trennen. Die Hyperebene ist über einen optimalen Gewichtsvektor  $\mathbf{w}$  und einen Schwellwert  $b$  definiert. Sie besitzt die Form:

$$\mathbf{w}^T \mathbf{x} + b = 0.$$

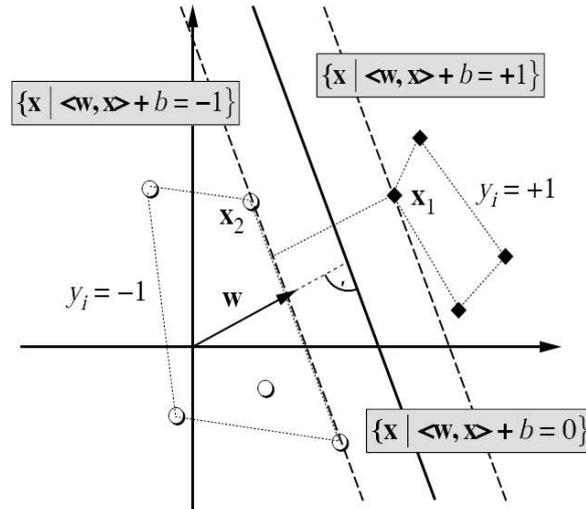
Damit lautet die Entscheidungsfunktion, die besagt, zu welcher der durch die Hyperebene getrennten Klassen ein Punkt  $\mathbf{x}$  gehört:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b). \quad (6.1)$$

Der Abstand eines Punktes zur Hyperebene ist gegeben durch:

$$r = \frac{\mathbf{w}^T \mathbf{x} + b}{|\mathbf{w}|}.$$

Der minimale Abstand definiert die Trenngüte der SVM. Die maximale Trenngüte wiederum definiert die optimale Hyperebene. Die Punkte der beiden Klassen, die jeweils am dichtesten zur



**Abbildung 6.1:** Darstellung der von der SVM erstellten optimalen Hyperebene zur Trennung von zwei Datensätzen [53].

Hyperebene liegen, werden „Support-Vektoren“ genannt. Bei einer optimalen Hyperebene gilt für diese Punkte:

$$\mathbf{w}^T \mathbf{x} + b = \begin{cases} 1 & \text{für positives Beispiel} \\ -1 & \text{für negatives Beispiel} \end{cases} \quad (6.2)$$

Somit beträgt die Breite des Korridors um die optimale Hyperebene  $d = \frac{2}{|\mathbf{w}|}$ . Um ein optimal trennende Klassifizierung zu erhalten, muss der Korridor möglichst breit und somit  $|\mathbf{w}|$  möglichst klein werden. Es handelt sich um ein Optimierungsproblem unter der Nebenbedingung, dass die einzelnen Punkte richtig klassifiziert werden.

$$\text{minimiere } \mathbf{w}^T \mathbf{w}$$

$$\text{unter der Nebenbedingung } -(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1) \leq 0.$$

$y_i$  gibt an, ob der jeweilige Punkt zu der positiven ( $y_i = 1$ ) bzw. negativen ( $y_i = -1$ ) Klasse gehört. Der Index  $i$  läuft über alle Beispiele der Trainingsmenge. Die zu dem Optimierungsproblem gehörige Lagrangefunktion besitzt die Form:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} |\mathbf{w}|^2 - \sum_{i=1}^p \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1). \quad (6.3)$$

In dieser Gleichung wird die zu minimierende Funktion mit  $\frac{1}{2}$  erweitert. Dies geschieht aus rechen-technischen Gründen und ändert nichts an dem zu lösenden Optimierungsproblem. Differenzieren der Lagrangefunktion ergibt den optimalen Gewichtsvektor  $\mathbf{w}^*$  und eine Nebenbedingung für die Lagrangemultiplikatoren  $\alpha_i$ .

$$\mathbf{w}^* = \sum_{i=1}^p \alpha_i^* y_i \mathbf{x}_i \quad (6.4)$$

$$0 = \sum_{i=1}^p \alpha_i y_i \quad (6.5)$$

Zusätzlich gilt die Karush-Kuhn-Tucker-Nebenbedingung:

$$0 = \alpha_i^*(y_i(\mathbf{w}^{*T} \mathbf{x}_i + b^*) - 1) \quad \text{für } 1 \leq i \leq p.$$

Der Vergleich mit obiger Definition der Support-Vektoren (Gleichung 6.2) zeigt, dass alle Lagrange-Faktoren bis auf die der Support-Vektoren Null werden. Somit müssen nur die Terme der Support-Vektoren in Gleichung 6.4 zur Berechnung der optimalen Gewichte berücksichtigt werden. Die optimalen Biaswerte  $b^*$  werden ebenfalls mit Gleichung 6.2 bestimmt. Unter Berücksichtigung von  $y_i$  lässt sie sich umformen zu:

$$y_s(\mathbf{w}^{*T} \mathbf{x}_s + b^*) = 1.$$

Daraus folgt:

$$\begin{aligned} b^* &= y_s - \mathbf{w}^{*T} \mathbf{x}_s \\ &= y_s - \sum_{i=1}^p \alpha_i^* y_i \mathbf{x}_i^T \mathbf{x}_s. \end{aligned}$$

Die Support-Vektoren komprimieren die Information über die optimale Hyperebene.

### 6.3.2 Duales Optimierungsproblem

Das Einsetzen des optimalen Gewichtsvektors 6.4 in die Lagrange-Gleichung 6.3 unter Berücksichtigung der Nebenbedingung 6.5 liefert das duale Optimierungsproblem.

$$Q(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i,j=1}^p \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^p \alpha_i$$

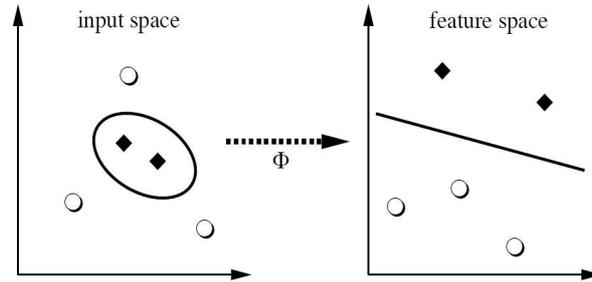
mit Nebenbedingungen

$$\begin{aligned} \sum_{i=1}^p \alpha_i^* y_i &= 0 \quad 1 \leq i \leq p \\ \alpha_i^* &\geq 0. \end{aligned}$$

Das duale Optimierungsproblem hängt nur noch von den Skalarprodukten von  $\mathbf{x}$  und nicht mehr direkt von  $\mathbf{x}$  ab. Diese Darstellung führt zum Kerngedanken der Support-Vector-Machines (kernel trick). Linear nicht trennbare Datensätze können mit einer Funktion  $\phi(\mathbf{x})$  in einen hochdimensionalen Raum abgebildet werden. Im höherdimensionalen Raum ist es wahrscheinlicher, die Daten linear trennen zu können. Dies wird realisiert, indem das Skalarprodukt im dualen Optimierungsproblem durch das Skalarprodukt der Abbildungsfunktionen ersetzt wird.

$$\mathbf{x}_i^T \mathbf{x}_j \Rightarrow \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

Da das Skalarprodukt der Abbildungsfunktionen schwer zu berechnen ist, wenn die Dimension des Abbildungsraums entsprechend hoch ist, wird eine Kernelfunktion  $K_\phi(\mathbf{x}_i, \mathbf{x}_j)$  eingeführt, die sich wie das Skalarprodukt verhält. Häufig verwendete Kernel werden in Tabelle 6.1 aufgeführt [66].



**Abbildung 6.2:** Linear nicht trennbare Daten werden in einen höherdimensionalen Raum abgebildet und dort linear getrennt. Dies führt zu einer nichtlinearen Trennung der Daten im Originalraum [53].

Kernelfunktion	Definition	Parameter
Linear	$\mathbf{x}_i^T \cdot \mathbf{x}_j$	
Polynomial	$(\kappa(\mathbf{x}_i^T \cdot \mathbf{x}_j) + \Theta)^d$	Ordnung $d$ (positiver Integer), $\kappa$ $\Theta$
Radial	$\exp\left(-\frac{ \mathbf{x}_i - \mathbf{x}_j ^2}{2\sigma}\right)$	$\sigma$
Sigmoidal	$\tanh(\kappa(\mathbf{x}_i^T \cdot \mathbf{x}_j) + \Theta)$	$\kappa$ und $\Theta$

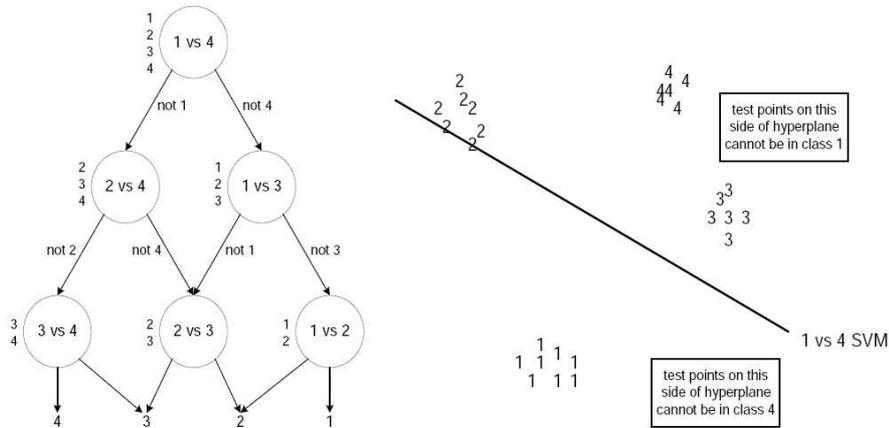
**Tabelle 6.1:** Häufig verwendete Kernel für Support-Vector-Machines

Die Entscheidungsfunktion 6.1 lässt sich mit Gleichung 6.4 und unter Berücksichtigung der Kernelfunktion folgendermaßen umformen.

$$\begin{aligned}
 f(\mathbf{x}) &= \text{sign} \left( \sum_{i=1}^p y_i \alpha_i \mathbf{x}_i^T \mathbf{x} + b \right) \\
 &= \text{sign} \left( \sum_{i=1}^p y_i \alpha_i \phi(\mathbf{x}_i^T) \phi(\mathbf{x}) + b \right) \\
 &= \text{sign} \left( \sum_{i=1}^p y_i \alpha_i K_{\phi}(\mathbf{x}_i^T, \mathbf{x}) + b \right)
 \end{aligned}$$

### 6.3.3 SVM zur Klassifikation von mehreren Objekten

Da die SVM als binärer Klassifikator entwickelt wurde, müssen für eine Klassifikation von mehreren Klassen Strategien entwickelt werden, um das Prinzip der SVM zu erweitern. Dazu werden in [53] verschiedene Techniken vorgestellt. Hier werden die drei am häufigsten verwendeten Methoden beschrieben [66]:



**Abbildung 6.3:** Links: gerichteter azyklischer Graph um zwischen vier Klassen zu unterscheiden. Rechts: Hyperebene des 1 vs. 4 Klassifikators im Merkmalsvektorraum [55].

**Einer gegen den Rest Klassifikation:** Für jedes der zu klassifizierenden Objekte wird eine binäre SVM trainiert [37]. Alle Objekte der jeweils anderen Klassen werden in der Trainingsphase als negative Beispiele betrachtet. In der Klassifikationsphase wird ein unbekanntes Objekt der Klasse zugeordnet, dessen SVM die höchste Ausgabe erzeugt (winner-takes-all). Dies ist auch der größte Kritikpunkt an diesem Verfahren, da es sich um unterschiedliche binäre Klassifikationsprobleme handelt und nicht sichergestellt ist, dass die Ausgaben der SVMs vergleichbar sind.

**Paarweise Klassifikation:** Bei der paarweisen Klassifikation werden bei  $M$  Klassen für jede mögliche Paarung zweier Klassen eine binäre SVM trainiert. Es entstehen somit  $(M-1)M/2$  SVMs und nicht  $M$ , wie es beim „Einer gegen den Rest Klassifikation“ der Fall ist [53]. In der Klassifikationsphase wird ein unbekanntes Objekt der Klasse zugeordnet, für die die Anzahl der SVMs, die das Objekt dieser Klasse zuordnen, am größten ist. Fälle, bei denen ein Objekt von der gleichen Anzahl von SVMs verschiedenen Klassen zugeordnet wird, müssen getrennt betrachtet werden.

Ein Nachteil dieser Vorgehensweise ist die große Anzahl von Klassifikatoren, die trainiert werden müssen, wenn die Zahl  $M$  der unterschiedlichen Klassen steigt. Die Klassifikation wird langsam, da der Merkmalsvektor von sehr vielen SVMs klassifiziert werden muss. Aus diesem Grund wird die „Paarweise Klassifikation“ oft mit gerichteten, azyklischen Graphen (engl.: *Directed Acyclic Graph (DAG)*) kombiniert. Diese verhindern die Ausführung von redundanten SVMs der „Paarweisen Klassifikation“.

Wird ein Objekt von den ersten  $N$  SVMs weder der Klasse  $M_i$  noch der Klasse  $M_j$  zugeordnet, ist die SVM, die zwischen diesen beiden Klassen entscheidet, redundant, da das Objekt höchstwahrscheinlich keiner der beiden Klassen angehört.

Um das Ausführen von redundanten SVMs zu vermeiden, werden die SVMs in einem DAG hierarchisch geordnet. Abbildung 6.3 zeigt einen solchen Graphen für ein Vier-Klassen-Problem. Da jede SVM eine Klasse ausschließen kann, müssen nur die SVMs entlang eines Pfades im Graphen ausgeführt werden. Das erreichte Blatt des Graphen gibt die Klasse an, der das Objekt zugeordnet wird.

Vielklassen-Zielfunktion (engl.: *Multi-Class Objective Functions*;) Im Gegensatz zu den beiden ersten Verfahren, werden in diesem Ansatz nicht mehrere binäre SVMs kombiniert, sondern die Entscheidungsfunktion der SVM wird so modifiziert, dass sie mit mehreren Klassen arbeiten kann.

In dieser Arbeit wird die „Einer gegen den Rest Klassifikation“ mit einer Art „Voting“ der SVMs kombiniert. Da für jede Klasse mehrere SVMs mit unterschiedlichen Merkmalen trainiert werden, wird folgenderweise vorgegangen:

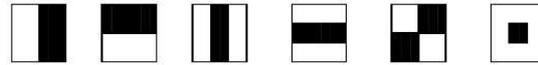
1. Die zu klassifizierende Kontur wird mit jeder SVM klassifiziert.
2. Jede Kontur besitzt einen Ergebnis-Vektor, der besagt, in wie weit die Kontur welcher Klasse angehört.
3. Wenn die Kontur von einer SVM erkannt wird, addiert diese den Prozentsatz, zu dem sie die Kontur erkannt hat, dem Wert der Klasse im Ergebnis-Vektor der Kontur hinzu, mit der sie trainiert wurde. Falls die SVM die Kontur als negatives Beispiel klassifiziert, wird der Wert subtrahiert.
4. Nachdem alle SVMs ausgewertet wurden, wird die Kontur der Klasse mit dem höchsten Wert im Ergebnis-Vektor zugewiesen. Es sei den, dieser Wert ist geringer als ein definierter optimaler Schwellwert. In diesem Fall wird die Kontur als negatives Beispiel klassifiziert.

Die Vorgehensweise hat den Vorteil, dass Fehler einer SVM durch SVMs der selben Klasse, die andere Merkmale verwenden aufgehoben werden. Konturen werden nur dann einer bestimmten Klasse zugewiesen, wenn die Mehrheit der SVMs dieser Klasse die Kontur als Objekt erkennen bzw. wenige SVMs die Kontur zu einem großen Prozentsatz erkennen. Der Nachteil dieser Vorgehensweise ist der, dass SVMs, deren Merkmale als Einzige ermöglichen die Kontur zu erkennen, unterdrückt werden.

## 6.4 Viola Jones AdaBoost Klassifikator

In der bestehenden Software-Umgebung ist bereits ein Klassifikator vorhanden, der nach dem Viola Jones AdaBoost Prinzip arbeitet. Da im folgenden Kapitel ein Performanz-Vergleich der in dieser Arbeit entwickelten Klassifikatoren mit dem Viola Jones Klassifikator stattfindet, wird hier kurz die Theorie des Viola Jones AdaBoost Klassifikators erläutert.

Die Grundidee von Viola und Jones ist es, sehr viele einfache, schnelle und suboptimale Klassifikatoren (werden als Merkmale bezeichnet) zu einem guten Klassifikator zu kombinieren [63]. Dabei besitzen die Merkmale typischerweise Klassifikationswerte, die nicht weit von 0,5 entfernt sind (schwache Klassifikatoren) [31]. Die Merkmale sind in Abbildung 6.4 zu sehen. Sie detektieren Helligkeitsunterschiede im Bild und lassen sich mit Hilfe von Integralbildern in  $O(1)$  berechnen. Sie werden für einen Trainingsdatensatz in ein Detektionsfenster eingepasst. Da die verwendeten Bildausschnitte sehr klein sind, können mehrere hunderttausend Merkmale für ein Bild entstehen [46]. Aus dieser großen Zahl von Merkmalen werden mit dem AdaBoost-Verfahren diejenigen Merkmale herausgearbeitet, die die Trainingsdaten am besten beschreiben. Diese Merkmale werden über gewichtete Summen zu starken Klassifikatoren kombiniert. Welche Merkmale in diese Summe genommen werden und wie diese gewichtet sind wird mit dem AdaBoost-Verfahren bestimmt. Das



**Abbildung 6.4:** Einige Viola Jones Merkmale [46].

Gesamtsystem verwendet eine Kaskade von  $N$  starken Klassifikatoren. In der Klassifikationsphase wird das Detektionsfenster über das zu klassifizierende Bild geschoben. Der jeweilige Bildausschnitt wird mit dem ersten starken Klassifikator klassifiziert und falls positiv an den nächsten starken Klassifikator in der Kaskade weitergereicht. Sobald ein starker Klassifikator in der Kaskade den Bildausschnitt negativ klassifiziert wird der Bildausschnitt verworfen. Da die einzelnen starken Klassifikatoren bereits ein Klassifikationsergebnis Nahe bei 1 bzw. 0 liefern wird die Fehlerrate durch die Kaskadierung dieser starken Klassifikator exponentiell verringert.

Da ein aus wenigen Merkmalen bestehender starker Klassifikator schneller ist, als ein aus vielen Merkmalen zusammengesetzter, aber im Allgemeinen eine schlechtere Performanz besitzt, wird die Kaskade mit ersteren begonnen. Dies sorgt dafür, dass ein schneller starker Klassifikator zunächst die Detektionsfenster grob aussortiert, auf die im Folgenden komplexere, aus mehr Merkmalen bestehende starke Klassifikatoren angewendet werden.



# Kapitel 7

## Performanzevaluation

### 7.1 Receiver-Operating-Characteristics (ROC)-Analyse

Die ROC-Analyse wurde im 2. Weltkrieg zur Rauschunterdrückung in der Radartechnik entwickelt. Sie ermöglicht es, das Verhalten von Diagnose-Systemen zu analysieren und zu visualisieren. Das Verfahren wurde bereits in den 60er Jahren des letzten Jahrhunderts in der Medizin eingesetzt und etablierte sich dort als Analyse-Verfahren in der Krankheitsdiagnostik. In den letzten Jahren hat sich das Verfahren auch im Bereich des maschinellen Lernens durchgesetzt [62].

Um die Performance eines Klassifikators beurteilen zu können, muss zunächst betrachtet werden, welche Ergebnisse ein Klassifikator liefern kann. Ein binärer Klassifikator erzeugt für eine Testinstanz nur vier diskrete Ausgaben:

True Positiv ( $tp$ ): die Testinstanz ist ein positives Beispiel und wird auch als solches klassifiziert.

False Positiv ( $fp$ ): die Testinstanz ist ein positives Beispiel und wird als negatives Beispiel klassifiziert.

True Negativ ( $tn$ ): die Testinstanz ist ein negatives Beispiel und wird als solches klassifiziert.

False Negative ( $fn$ ): die Testinstanz ist ein negatives Beispiel und wird als positives klassifiziert.

Aus diesen Größen können Performanz Metriken hergeleitet werden. Zur Erstellung der ROC-Analyse werden die Metriken  $fp$ -Rate und  $tp$ -Rate wie folgt definiert:

$$fp\text{-Rate} = \frac{fp}{tn - fp} \quad \text{und} \quad tp\text{-Rate} = \frac{tp}{tp + fn}.$$

Die  $tp$ -Rate gibt somit das Verhältnis von korrekt positiv klassifizierten Beispielen zur Gesamtzahl der positiven Beispiele an. Entsprechendes gilt für die  $fp$ -Rate und negative Beispiele. Beide Werte besitzen somit den Wertebereich von  $[0,1]$ . Oft werden im Zusammenhang mit der ROC-Analyse auch die Begriffe *sensitivity* und *specificity* verwendet, wobei *sensitivity* ein Synonym für die  $tp$ -Rate ist und  $specificity = 1 - fp\text{-Rate}$  gilt:

$$sensitivity = \frac{tp}{tp + fn}$$

$$\text{specificity} = \frac{tn}{tn + fp}.$$

Visualisiert wird die Performanz im ROC-Raum. Dieser entsteht, indem die  $tp$ -Rate eines Klassifikators über seiner  $fp$ -Rate aufgetragen wird. Wie Abbildung 7.1 zeigt, wird ein diskreter Klassifikator

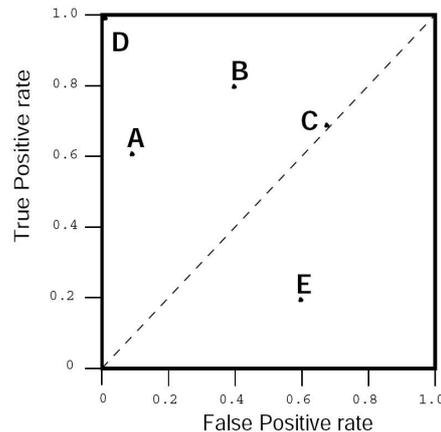


Abbildung 7.1: ROC-Graph für fünf diskrete Klassifikatoren [12].

durch einen Punkt im ROC-Graphen dargestellt. Da ein diskreter Klassifikator nur die Klasse für die jeweiligen Beispiele ausgibt, kann er nur eine diskrete  $tp$ -Rate bzw.  $fp$ -Rate besitzen.

Die Lage des Punktes im ROC-Graphen lässt auf die Performanz des Klassifikators schließen. Befindet sich ein Klassifikator auf der Diagonalen zwischen den Punkten (0,0) und (1,1), entspricht die Performanz einfachem Raten der Klassen, da der Anteil der richtig klassifizierten Beispiele gleich dem Anteil der falsch klassifizierten Beispiele ist. Befindet sich der Klassifikator in der unteren rechten Hälfte, ist seine Performanz schlechter als Raten. Die linke obere Hälfte ist dementsprechend besser. Somit ist die rechte untere Hälfte des ROC-Graphen nicht relevant, da ein Klassifikator aus diesem Bereich durch Invertieren seiner Ausgaben in einen Klassifikator umgewandelt werden kann, der besser als Raten ist.

Ein Vorteil der ROC-Graphen ist der, dass die Beurteilung eines Klassifikators unabhängig vom Verhältnis der Testklasseninstanzen ist. Verändert sich die Anzahl von positiven bzw. negativen Instanzen im Testdatensatz, verändert sich nur das Verhältnis von linker zur rechter Spalte der ?confusion? Matrix. Da sowohl  $tp$ -Rate als auch  $fp$ -Rate nur ein Verhältnis in einer Spalte bilden, bleiben diese Metriken konstant. Metriken wie Exaktheit (engl.: *accuracy*), Präzision (engl.: *precision*), lift, F-score,... sind hingegen sensitiv gegenüber Veränderungen des Verhältnisses der Testklasseninstanzen, da diese Metriken durch Werte aus unterschiedlichen Spalten gebildet werden [12], wie in folgenden Gleichungen ersichtlich ist:

$$\begin{aligned} \text{Exaktheit} &= \frac{tp + tn}{tp + tn + fp + fn} \\ \text{Präzision} &= \frac{tp}{tp + fp} . \end{aligned}$$

### 7.1.1 ROC-Kurven

ROC-Kurven können mit nicht diskreten Klassifikatoren erstellt werden. Sie beruhen darauf, dass der Klassifikator einer Testinstanz nicht nur ein Klassenlabel zuordnet, sondern auch einen Wert, wie wahrscheinlich die Testinstanz der Klasse entspricht. Dadurch ist es möglich, die  $tp$ - bzw.  $fp$ -Rate in Abhängigkeit eines Schwellwertes  $\Theta$  zu definieren. Dazu werden in dieser Arbeit die Klassen  $p$  und  $n$  auf das Intervall  $[0,1]$  abgebildet.  $100\%n$  entspricht  $0$ ,  $0\%n$  ebenso wie  $0\%p$  entspricht  $0,5$  und  $100\%p$  wird auf  $1$  abgebildet. In der ROC-Analyse wird eine Testinstanz nur dann als positiv gezählt, wenn der Klassenzugehörigkeitswert über dem Schwellwert liegt. Jeder Schwellwert  $\Theta$  erzeugt somit einen unterschiedlichen Punkt im ROC-Graphen. Der Schwellwert wird von  $-0.1$  bis  $1.1$  hochgezählt. Bei einem Schwellwert von  $-0.1$  werden alle Testinstanzen als positiv klassifiziert. Dadurch wird die  $tp$ - sowie die  $fp$ -Rate gleich  $1$ , da alle positiven Beispiele richtigerweise als positiv erkannt werden und alle negativen Beispiele falsch klassifiziert werden. Durch Erhöhung des Schwellwertes werden immer mehr Testinstanzen als negativ klassifiziert. Dadurch sinkt die  $fp$ -Rate und bei einem nicht optimalen Klassifikator sinkt die  $tp$ -Rate ebenfalls, da nun positive Beispiele fälschlicherweise als negativ klassifiziert werden. Erreicht  $\Theta$   $1.1$ , werden alle Testinstanzen als negativ klassifiziert, so dass sowohl  $tp$ - als auch  $fp$ -Rate  $0$  werden. Die ROC-Kurve entsteht, indem die durch die unterschiedlichen Schwellwerte erzeugten Punkte im ROC-Graphen verbunden werden. Die ROC-Kurve besitzt aus oben genannten Gründen die Endpunkte  $(0,0)$  bzw.  $(1,1)$ .

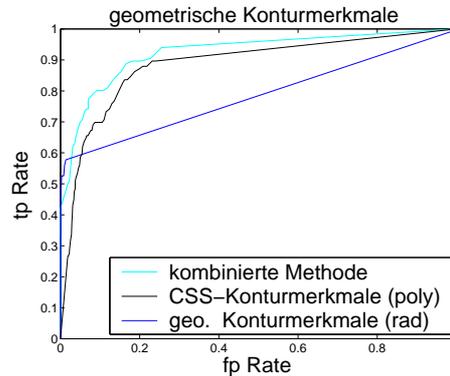
### 7.1.2 Fläche unter der ROC-Kurve (AUC)

Um die Performanz eines Klassifikators auf einen Wert abzubilden, wird die Fläche unter der ROC-Kurve berechnet. Diese wird als AUC-Wert (AUC = engl.: *area under curve*) der ROC-Kurve bezeichnet. Der AUC-Wert besitzt einen Wertebereich von  $[0,1]$ . Da eine Fläche von  $0,5$  einfachem Raten entspricht, sollte der AUC-Wert eines Klassifikators (analog zur Erläuterung über die Lage diskreter Klassifikatoren) einen größeren Wert besitzen. Der AUC-Wert sagt etwas über die durchschnittliche Performanz eines Klassifikators aus. Ein Klassifikator mit einem hohen AUC-Wert kann somit für einen gewissen Schwellwertbereich eine schlechtere Performanz besitzen als ein Klassifikator mit einem niedrigen AUC-Wert. Der AUC-Wert entspricht der Wahrscheinlichkeit, dass der Klassifikator einer zufällig gewählten positiven Instanz einen höheren Wert zuordnet als einer zufällig gewählten negativen Instanz [12].

### 7.1.3 Multi-Klassen ROC-Analyse

Die bisherigen Erläuterungen gehen von einem Klassifikator aus, der zwischen zwei Klassen entscheidet. Diese Situation ist zwar in den meisten ursprünglichen Anwendungen (Signalverarbeitung, medizinische Anwendungen) der ROC-Analyse gegeben. Im Bereich des maschinellen Sehens (engl.: *computer vision*) sollen aber oft mehrere Objekte erkannt werden. Somit muss die ROC-Analyse auf  $n$  Klassen erweitert werden. Die  $n$  Klassen ?confusion? Matrix besitzt neben den  $n$  richtig klassifizierten Elementen der Hauptdiagonalen  $n^2 - n$  verschiedene Falschklassifikationen.

Eine Möglichkeit, dieses Problem zu lösen, ist, für  $n$  Klassen  $n$  ROC-Analysen durchzuführen, wobei sich jede Analyse wieder auf zwei Klassen bezieht. Dabei werden alle von der aktuellen Klasse verschiedenen Klassen als negative Instanzen gezählt. Durch diese Vorgehensweise verliert die



**Abbildung 7.2:** ROC-Kurven für zwei Klassifikatoren  $A$  und  $B$  und für den aus diesen kombinierten Klassifikator.

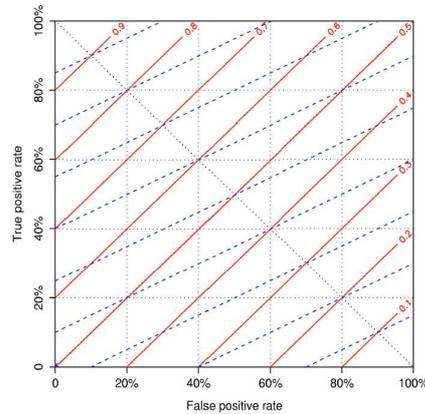
ROC-Analyse jedoch ihre Insensitivität gegenüber Veränderungen des Klassenverhältnisses im Testdatensatz, da sich die negativen Beispiele aus  $n - 1$  Klassen bzw. Spalten der ?confusion? Matrix zusammensetzen.

Durch  $n$  ROC-Graphen werden für einen Klassifikator auch  $n$  AUC-Werte berechnet, die zu einem AUC-Wert kombiniert werden müssen.

#### 7.1.4 Konvexe Hülle der ROC-Kurve und kombinierte Klassifikatoren

Um verschiedene Klassifikatoren zu vergleichen, gibt es neben der AUC-Metrik die konvexe Hülle eines ROC-Graphen. Letztere berücksichtigt den Fall, dass ein Klassifikator mit einem geringeren AUC-Wert durchaus für gewisse Schwellwerte bessere Ergebnisse liefern kann als ein Klassifikator mit einem höheren AUC-Wert.

Befindet sich die ROC-Kurve eines Klassifikators  $A$  für einen Teilbereich dichter an der oberen linken Ecke des ROC-Graphen als ein Klassifikator  $B$ , wird letzterer von  $A$  für diesen Teilbereich dominiert. In Abbildung 7.7b ist zu sehen, dass die mit polynomialen Kernel trainierte SVM die SVM mit linearen bzw. radialen Kernel über den gesamten Bereich dominiert. Abbildung 7.2 zeigt hingegen, dass nicht eine Kurve dominiert, sondern der Klassifikator  $A$  für einen Bereich dominiert und der Klassifikator  $B$  für einen anderen. Diejenige Kurve, die alle Kurven verschiedener Klassifikatoren dominiert, wird konvexe Hülle (ROCCH) des ROC-Graphen genannt. In dieser Arbeit wird die konvexe Hülle dazu verwendet, die SVMs eines Objektes zu bestimmen, die kombiniert einen besseren Klassifikator ergeben würden (vgl. Abschnitt 6.3.3). In Abbildung 7.2 sind die ROC-Kurven für drei Klassifikatoren zu sehen. Es fällt auf, dass es sich bei dem durch die blaue ROC-Kurve repräsentierten Klassifikator um einen „konservativen“ Klassifikator handelt. Diese klassifizieren verhältnismäßig wenig Objekte als positiv. Es werden nur solche erkannt, die eine starke Klassenzugehörigkeit besitzen. Somit erzeugt diese Kategorie von Klassifikatoren sehr wenig  $fp$ -Klassifikationen, und die Kurve liegt dicht an der  $tp$ -Rate-Achse. Konservative Klassifikatoren besitzen aus erwähnten Gründen typischerweise keine hohe  $tp$ -Rate. Die blaue ROC-Kurve gehört hingegen zu einem „liberalen“ Klassifikator. Diese Sorte von Klassifikatoren erkennt sehr viele der positiven Beispiele, macht aber im Vergleich zu einem konservativen Klassifikator mehr Fehler bei den negativen Beispielen. Der dritte, aus den beiden anderen Klassifikatoren kombinierte Klassifikator dominiert über einen großen Bereich des ROC-Graphen und ist sogar besser als die



**Abbildung 7.3:** Exaktheit Isometrien für eine Klassenzusammensetzung von  $c = 1$  (rote Linien) bzw. einer Klassenzusammensetzung von  $c = 0,5$  (blaue Linien) [14].

konvexe Hülle. Für einen kleinen Bereich dominiert jedoch der Klassifikator  $A$  den kombinierten Klassifikator.

In dem vorgestellten Fall werden die Klassifikationsergebnisse der Einzelklassifikatoren jeweils gleich gewichtet, um den kombinierten Klassifikator zu erhalten. Es ist jedoch durchaus denkbar, dass für eine definierte Anzahl von Klassifikatoren ein bestimmter Performanz Punkt im ROC-Graphen erreicht werden soll. Hierzu wird in [12] ein Verfahren vorgestellt, das anhand der ROC-Kurven der einzelnen Klassifikatoren deren Gewichtung bestimmt, damit der kombinierte Klassifikator den gewünschten Performanzpunkt erreicht.

### 7.1.5 Linien gleicher Exaktheit und der optimale Schwellwert

ROC-Kurven lassen sich auf andere Performanz Metriken wie Exaktheit oder Präzision abbilden. Dabei werden Punkte im ROC-Graphen, die für eine bestimmte Metrik den selben Wert besitzen, als Isometrik bezeichnet. In dieser Arbeit wird die ROC-Kurve auf das Performanz Maß Exaktheit abgebildet, um zu bestimmen, welcher Schwellwert optimalerweise gewählt werden sollte:

$$\text{Exaktheit} = \frac{tp\text{-Rate} + c \cdot (1 - fp\text{-Rate})}{1 + c}$$

wobei  $c$  die Klassenzusammensetzung des Testdatensatzes beschreibt:

$$c = \frac{\text{Anzahl der negativen Beispiele}}{\text{Anzahl der positiven Beispiele}}.$$

In Abbildung 7.3 sind die Isometrien für Exaktheit mit  $c = 1$  und  $c = 0,5$  abgebildet. Es ist zu sehen, dass die Isometrien für eine Klassenzusammensetzung von 0,5 flacher verlaufen als die einer Klassenzusammensetzung, welche aus gleich vielen positiven wie negativen Beispielen besteht. Da in der Objekterkennung typischerweise die Anzahl der negativen Beispiele sehr viel größer ist als die Anzahl der positiven Beispiele, verlaufen die Isometrien für Exaktheit nahezu vertikal. Auf der Diagonalen von Punkt (0,1) zu Punkt (1,0) ist unabhängig von der Klassenzusammensetzung die Exaktheit gleich der  $tp$ -Rate. Dieser Sachverhalt wird in Applikationen ausgenutzt, in denen die

Klassenzusammensetzung nicht bekannt ist.

Der optimale Schwellwert der SVM ergibt sich als der, bei dem die Exaktheit am größten ist.

## 7.2 Ergebnis der Performanz-Analyse

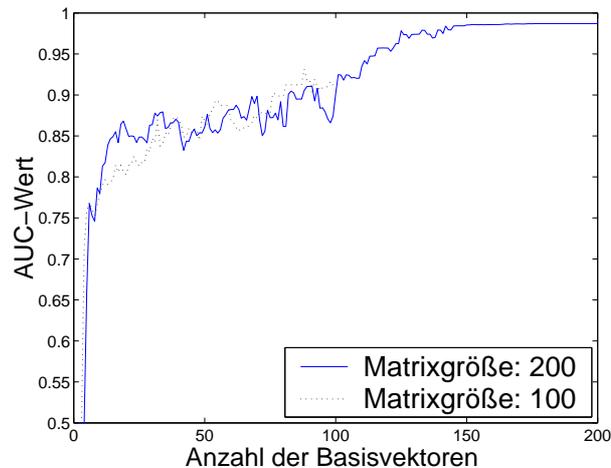
Es stellt sich heraus, dass die Angular-Radial-Transformation sich am besten eignet, um die extrahierten Konturen zu beschreiben. So wird der Testdatensatz des „Mensch 1“-Objektes von der ART-Repräsentation mit orthogonalen, radialen Basisfunktionen nahezu perfekt klassifiziert. Es werden 1887 der 1893 negativen und 101 der 104 positiven Beispiele beim optimalen Schwellwert richtig klassifiziert (vgl. Tabelle 7.1, K5). Die ART-Repräsentation mit orthogonalen radialen Basisvektoren besitzt für alle trainierte Objekte eine bessere Performanz als die ART-Repräsentation ohne orthogonale Basisvektoren, wie sie im MPEG-7 Standard vorgeschlagen wird.

Die zweitbeste Performanz liefert das Border-Signature-Verfahren. Es zeigt zwar für das „Kurt 2D“-Objekt ein schlechteres Ergebnis als die Zernike-Momente, ist für die anderen Objekte jedoch besser geeignet. Außerdem benötigt das Verfahren die geringste Rechenzeit von nur 70,4 ms pro Kontur. Die Zernike-Momente sind sehr rechenintensiv. So werden pro Kontur ca. 200 ms benötigt (Pentium4 3 GHz 1 GB Arbeitsspeicher). Bei durchschnittlich 20 extrahierten Konturen pro Tiefenbild wird eine Klassifikationszeit von etwa 4 s erreicht.

Das zweit schnellste und nächstbeste Verfahren mit 4 ms pro Kontur ist das CSS-Verfahren. Dieses Verfahren ist interessant, da die in dieser Arbeit verwendete Eigenraum-Darstellung der CSS-Repräsentation noch nie auf realen Daten angewendet wurde. Am schlechtesten schneiden die geometrischen Merkmale bei der Beschreibung der Kontur ab. Natürlich hängt die Performanz der einzelnen Methoden von der zu klassifizierenden Objektklasse ab. So ist die CSS-Methode besser geeignet, einen breitbeinig stehenden Menschen (aufgrund der signifikanteren Krümmung) zu erkennen als einen Menschen, der mit geschlossenen Beinen vor dem Scanner steht. Die CSS-Methode besitzt für dieses Objekt sogar eine bessere Performanz als die Zernike-Momente.

Ferner ist an den Graphen der verschiedenen Klassifikatoren zu sehen, dass bei der Repräsentation durch geometrische Merkmale, der CSS-Repräsentation und dem Border-Signature-Verfahren die Performanz signifikant von dem mit der SVM verwendeten Kernel abhängt. Die SVM der CSS-Repräsentation mit radialem Kernel klassifiziert alle Beispiele als negativ, so dass die ROC-Kurve nur aus den Punkten (0,0) und (1,1) besteht. Das Border-Signature-Verfahren besitzt eine klare Performanzreihenfolge in den verwendeten Kernels. Der radiale Kernel besitzt die beste Performanz, gefolgt vom polynomialen und linearen Kernel. Bei SVMs, die mit geometrischen Konturmerkmalen trainiert wurden, ist zu sehen, dass die klare Performanzreihenfolge der Kernel nicht gegeben ist. So ist die SVM mit polynomialen Kernel für das „Mensch 2“-Objekt ein liberaler Klassifikator und die SVM mit linearem Kernel ein konservativer Klassifikator, wohin beim „Kurt 2D“-Objekt die Charakteristik der Kernel genau andersherum verläuft. Für das „Mensch 1“-Objekt hingegen erzeugen alle drei Kernel einen konservativen Klassifikator. Im Allgemeinen lässt sich sagen, dass die Performanz der SVMs, die mit geometrischen- oder Border-Signature-Merkmalen trainiert wurden im Vergleich zu den anderen Verfahren nicht nur mit der Wahl des Kernels sondern auch von Objekt zu Objekt stark variiert.

Zu den kombinierten SVMs lässt sich sagen, dass die ortho. ART-Repräsentation für das „Mensch 1“-Objekt so gut ist, dass die Kombination mit anderen Verfahren nur eine sehr geringe Performanzsteigerung erzielt. Wohingegen die Kombination aller Verfahren beim „Mensch 2“ und „Kurt 2D“-Objekt zu einer deutlicheren Performanzsteigerung führt. Auffällig ist, dass der Schwellwert für die



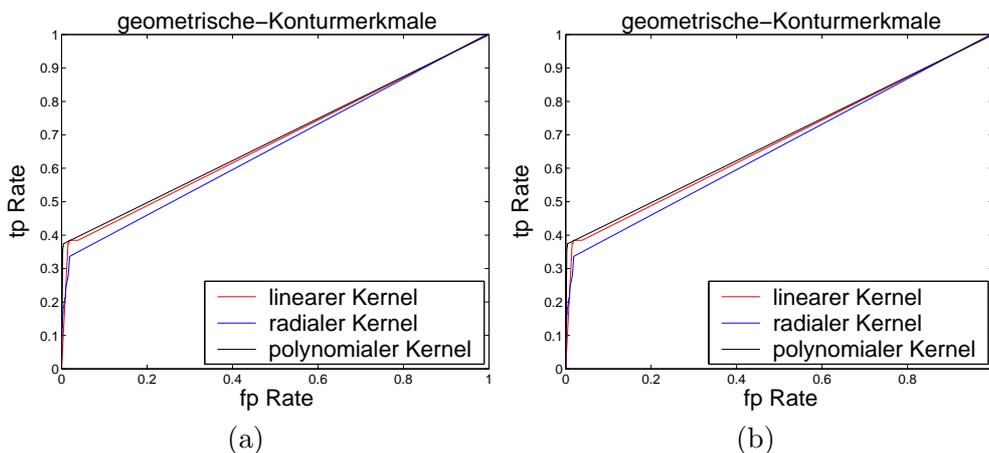
**Abbildung 7.4:** AUC-Wert in Abhängigkeit zur Anzahl der verwendeten Basisvektoren bei der CSS-Repräsentation. Die Gesamtanzahl der verwendeten Beispiele beträgt für die Klassifikatoren der beiden Kurven jeweils 200. Die Anzahl der zur Abbildungsmatrix herangezogenen Beispiele beträgt für die schwarze (gepunktete) Kurve 100 und für die blaue Kurve 200.

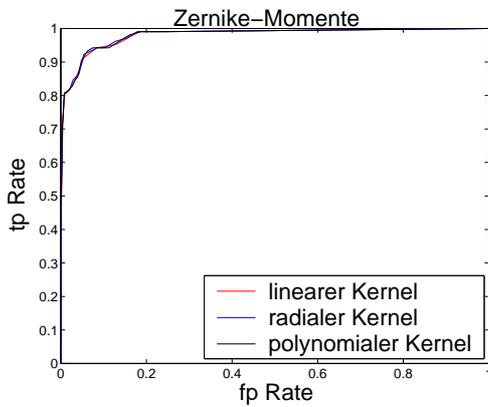
höchste Exaktheit der Einzelklassifikatoren zwischen 0 und 0,20 oder 0,75 und 1 liegt, wohingegen die Kombination der Klassifikatoren zu einem Schwellwert in die Nähe von 0,5 führt.

Neben der Performanz der Klassifikatoren wird mit der ROC-Analyse in dieser Arbeit ebenfalls untersucht, wie sich die Performanz der CSS-Konturrepräsentation in Abhängigkeit zur Anzahl der verwendeten Basisvektoren verhält. Die Anzahl der Beispiele wird innerhalb dieser Untersuchung konstant gehalten. Die Basisvektoren werden reduziert, indem nicht die gesamte  $\mathbf{U}$ -Matrix verwendet wird, sondern diese Spaltenweise reduziert wird. Durch Transponieren der reduzierten  $\mathbf{U}$ -Matrix und Multiplizieren mit dem CSS-Merkmalvektor  $\mathbf{x}$  entsteht ein reduzierter Eigenraum-Merkmalvektor (vgl. 5.5.3). Abbildung 7.4 zeigt den AUC-Wert des jeweiligen Klassifikators über der Anzahl der verwendeten Basisvektoren. Der Unterschied zwischen den Klassifikatoren der blauen und schwarzen (gepunkteten) Kurve ist der, dass bei der schwarzen (gepunkteten) Kurve nicht alle positiven Beispiele zur Erzeugung der  $\mathbf{U}$ -Matrix herangezogen werden. Diese Messungen werden unternommen, da so in der Trainingsphase positive Beispielkonturen trainiert wird, die nicht zur Erzeugung des Objekt-Eigenraums verwendet werden. Dadurch wird das Verfahren auf unbekannte positive Konturen in der Online-Phase vorbereitet. Die maximale Anzahl der Basisvektoren ist aufgrund der kleineren  $\mathbf{U}$ -Matrix geringer als bei den Klassifikatoren der blauen Linie, bei denen alle Beispiele zur Erzeugung des Eigenraums verwendet werden. Es ist zu sehen, dass die Klassifikatoren der schwarzen (gepunkteten) Linie im Mittel keinen höheren AUC-Wert besitzen als die der blauen Linie. Daher scheint es nicht wichtig zu sein, zum Training des Klassifikators Merkmalsvektoren einzusetzen, die nicht zur Erzeugung des Eigenraums beigetragen haben. Die blaue Linie besitzt ferner bei Klassifikatoren, die zwischen 100 und 150 Basisvektoren verwenden, eine deutliche Performanzsteigerung, so dass der Klassifikator mit der besten Performanz aus der blauen Linie kommt. Zur Frage der notwendigen Basisvektoren ist zu sagen, dass Klassifikatoren der blauen Linie ab 150 Basisvektoren keine Performanzsteigerung mehr besitzen und der Klassifikator der schwarzen (gepunkteten) Linie mit der besten Performanz etwa 90 Basisvektoren verwendet. Außerdem ist die Verwendung von vielen Basisvektoren nicht kritisch, da ein signifikanter Unterschied in der Rechenzeit, im Vergleich zu wenigen Basisvektoren, nur in der Trainingsphase des Klassifikators entsteht.

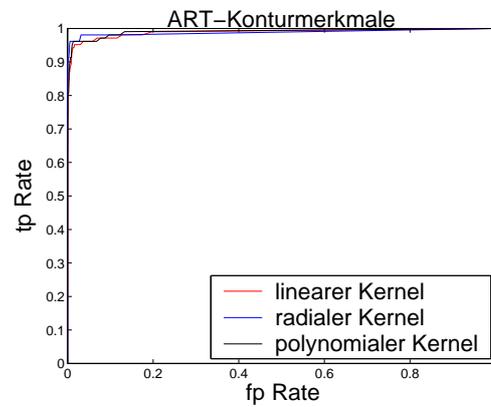
	K1	K2	K3	K4	K5	K6	K7
Mensch 1	200	200	200	200	200	200	200
Mensch 2	-	-	-	-	-	-	-
Kurt 2D	-	-	-	-	-	-	-
neg. Beispiele	700	700	700	700	700	700	700
Kreisförmigkeit	<b>x</b>	-	-	-	-	-	<b>x</b>
Seitenverhältnis	<b>x</b>	-	-	-	-	-	<b>x</b>
Füllungsgrad	<b>x</b>	-	-	-	-	-	<b>x</b>
Exzentrizität	<b>x</b>	-	-	-	-	-	<b>x</b>
Orientierung	<b>x</b>	-	-	-	-	-	<b>x</b>
Radienverhältnis	<b>x</b>	-	-	-	-	-	<b>x</b>
Hu-Momente	<b>x</b>	-	-	-	-	-	<b>x</b>
Zernike-Momente	-	-	<b>x</b>	-	-	-	<b>x</b>
Anzahl	7	-	42	-	-	-	7 bzw. 42
CSS	-	<b>x</b>	-	-	-	-	<b>x</b>
U-Matrix Beispiele	-	200	-	-	-	-	200
Anzahl der Basisvektoren	-	200	-	-	-	-	200
ART	-	-	-	<b>x</b>	-	-	<b>x</b>
ortho. ART	-	-	-	-	<b>x</b>	-	<b>x</b>
Border-Signature	-	-	-	-	-	<b>x</b>	<b>x</b>
Rechenzeit [ms]	7	4	187	32	32	0,4	
Kernel	poly.	lin.	poly.	poly.	rad.	rad.	-
optimaler Schwellwert	0,98	0,76	0,15	0,77	0,78	0,97	0,40
Exaktheit	0,964	0,989	0,982	0,990	0,995	0,963	0,996
AUC	0,685	0,986	0,980	0,991	0,999	0,990	0,999

**Tabelle 7.1:** Vergleich der Klassifikatoren für das Objekt „Mensch 1“: zunächst wird die Anzahl der Trainingsbeispiele der jeweiligen Objekte aufgeführt. Darunter sind die Merkmale markiert, die jeweils extrahiert werden, gefolgt von der Rechenzeit in ms, die zur Berechnung der jeweiligen Merkmale benötigt wird. Es wird über 1997 Konturen gemittelt und ein Pentium4 3 GHz Rechner mit 1 GB Arbeitsspeicher verwendet. Unten in der Tabelle ist der Kernel mit der besten Performanz für die jeweilige Methode, der optimale Schwellwert für diese SVM, die Exaktheit (bestimmt den optimalen Schwellwert) und der jeweilige AUC-Wert aufgeführt (bestimmt den optimalen Kernel).

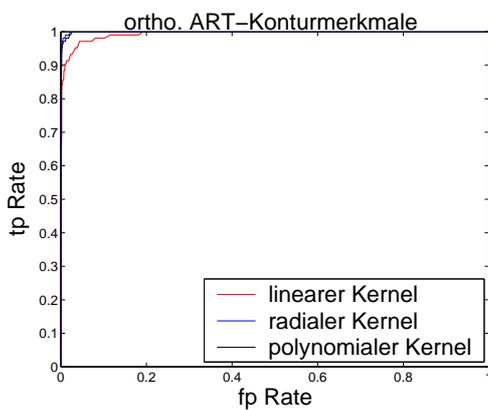




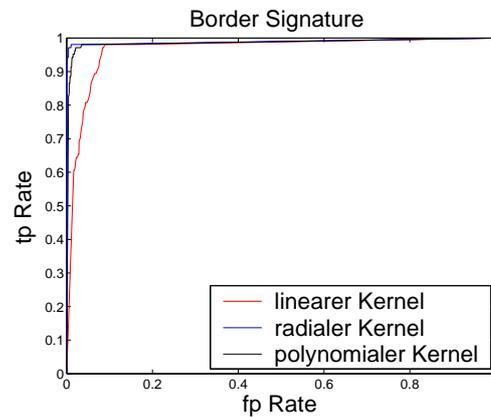
(c)



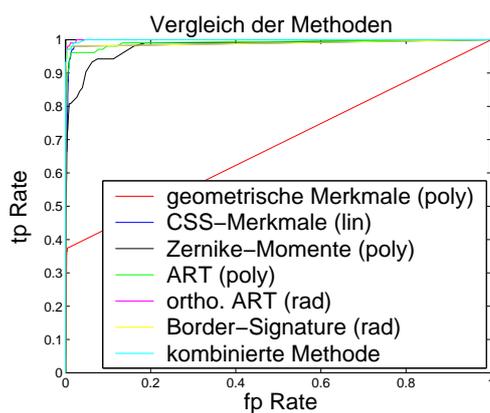
(d)



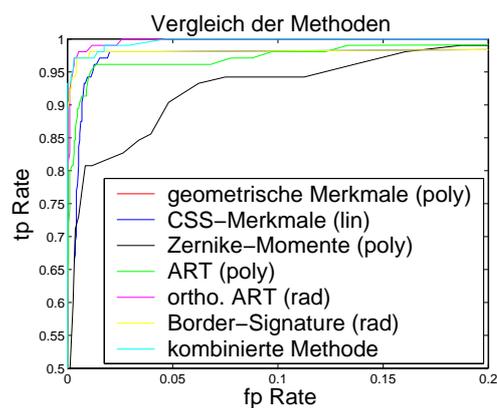
(e)



(f)



(g)

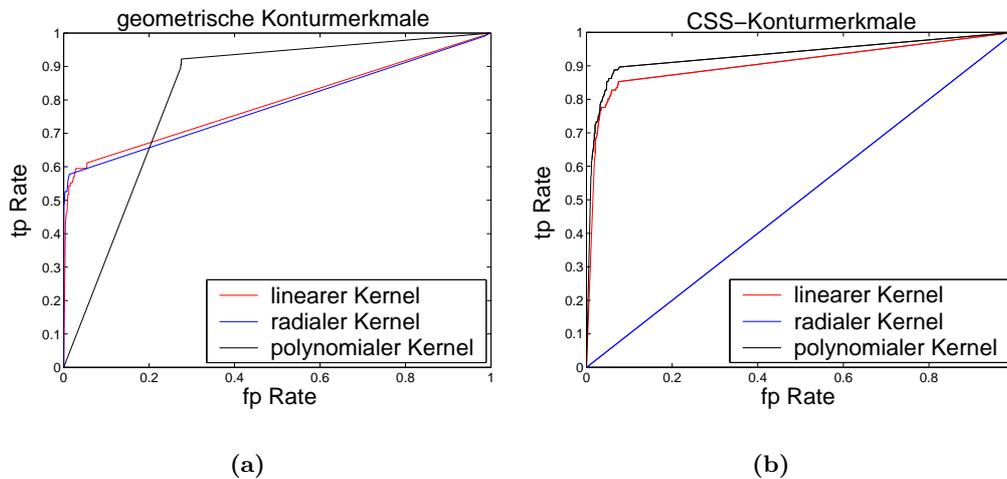


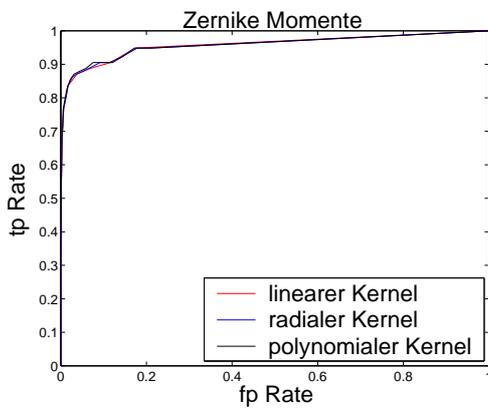
(h)

**Abbildung 7.5:** (a-f): ROC-Kurven der mit den verschiedenen Methoden trainierten SVMs (K1-K6) und der Vergleich (g)(h) dieser SVMs mit der aus ihnen kombinierten SVM (K7).

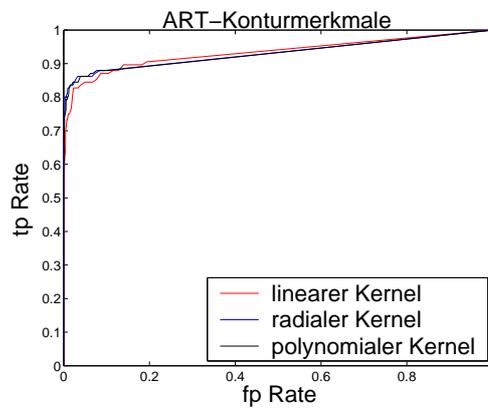
	K8	K9	K10	K11	K12	K13	K14
Mensch 1	-	-	-	-	-	-	-
Mensch 2	200	200	200	200	200	200	200
Kurt 2D	-	-	-	-	-	-	-
neg. Beispiele	700	700	700	700	700	700	700
Kreisförmigkeit	<b>x</b>	-	-	-	-	-	<b>x</b>
Seitenverhältnis	<b>x</b>	-	-	-	-	-	<b>x</b>
Füllungsgrad	<b>x</b>	-	-	-	-	-	<b>x</b>
Exzentrizität	<b>x</b>	-	-	-	-	-	<b>x</b>
Orientierung	<b>x</b>	-	-	-	-	-	<b>x</b>
Radienverhältnis	<b>x</b>	-	-	-	-	-	<b>x</b>
Hu-Momente	<b>x</b>	-	-	-	-	-	<b>x</b>
Zernike-Momente	-	-	<b>x</b>	-	-	-	<b>x</b>
Anzahl	7	-	42	-	-	-	7 bzw. 42
CSS	-	<b>x</b>	-	-	-	-	<b>x</b>
U-Matrix Beispiele	-	200	-	-	-	-	200
Anzahl der Basisvektoren	-	200	-	-	-	-	200
ART	-	-	-	<b>x</b>	-	-	<b>x</b>
ortho. ART	-	-	-	-	<b>x</b>	-	<b>x</b>
Border-Signature	-	-	-	-	-	<b>x</b>	<b>x</b>
Rechenzeit [ms]	7	4	187	32	32	0,4	?227?
Kernel	rad.	poly.	lin.	rad.	poly.	rad.	-
optimaler Schwellwert	0,18	0,93	0,06	0,76	0,81	0,86	0,46
Exaktheit	0,969	0,961	0,980	0,985	0,983	0,988	0,986
AUC	0,784	0,932	0,958	0,931	0,933	0,957	0,972

**Tabelle 7.2:** Fortsetzung von Tabelle 7.1 für das Objekt „Mensch 2“.

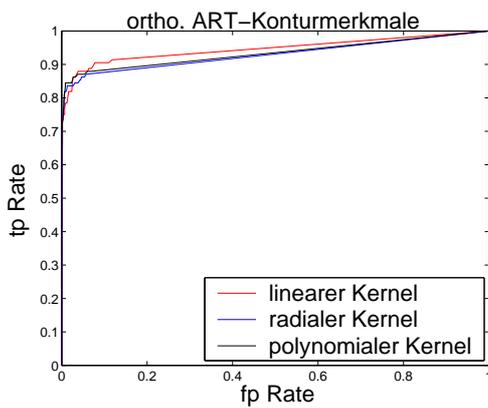




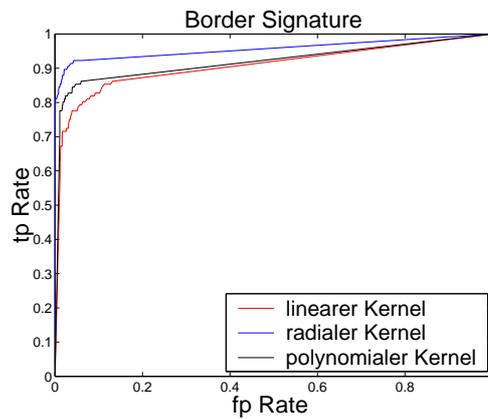
(c)



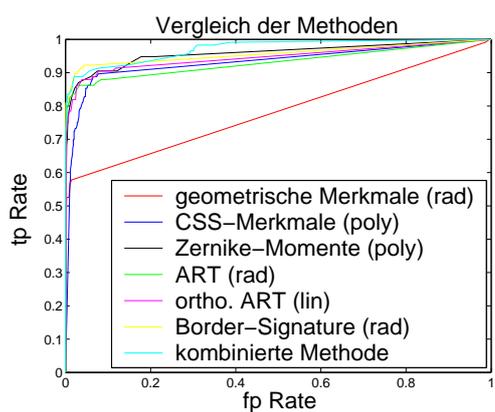
(d)



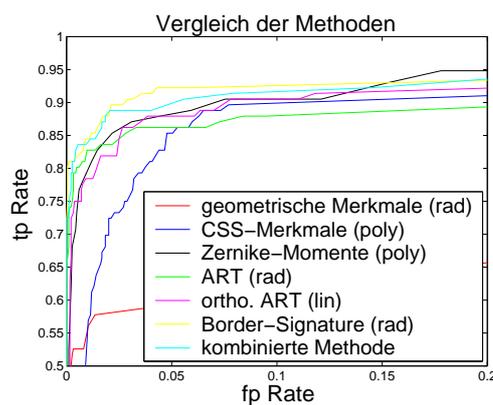
(e)



(f)



(g)

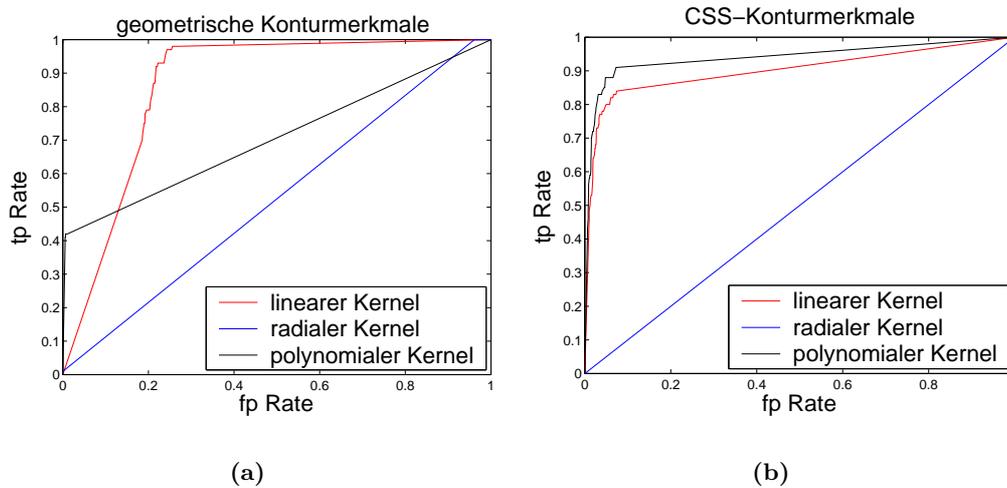


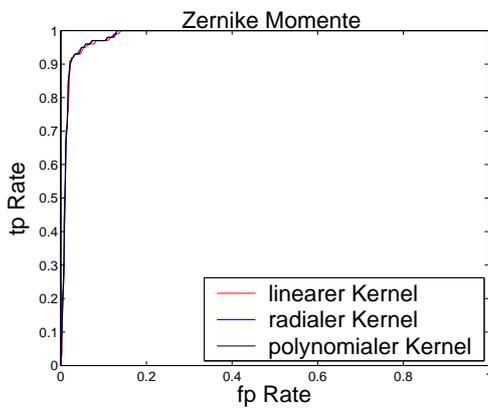
(h)

**Abbildung 7.6:** (a-f): ROC-Kurven der mit den verschiedenen Methoden trainierten SVMs (K8-K13) und der Vergleich (g)(h) dieser SVMs mit der aus ihnen kombinierten SVM (K14).

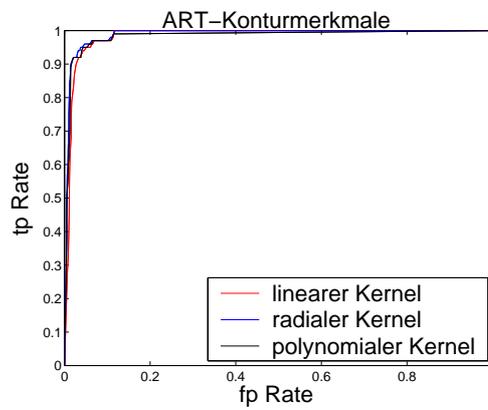
	K15	K16	K17	K18	K19	K20	K21	K22
Mensch 1	-	-	-	-	-	-	-	-
Mensch 2	-	-	-	-	-	-	-	-
Kurt 2D	200	200	200	200	200	200	200	200
neg. Beispiele	700	700	700	700	700	700	700	700
Kreisförmigkeit	x	-	-	-	-	-	x	x
Seitenverhältnis	x	-	-	-	-	-	x	x
Füllungsgrad	x	-	-	-	-	-	x	x
Exzentrizität	x	-	-	-	-	-	x	x
Orientierung	x	-	-	-	-	-	x	x
Radienverhältnis	x	-	-	-	-	-	x	x
Hu-Momente	x	-	-	-	-	-	x	x
Zernike-Momente	-	-	x	-	-	-	-	x
Anzahl	7	-	42	-	-	-	7	7 bzw. 42
CSS	-	x	-	-	-	-	-	x
U-Matrix Beispiele	-	200	-	-	-	-	-	200
Anzahl der Basisvektoren	-	200	-	-	-	-	-	200
ART	-	-	-	x	-	-	-	x
ortho. ART	-	-	-	-	x	-	-	x
Border-Signature	-	-	-	-	-	x	-	x
Rechenzeit [ms]	7	4	187	32	32	0,4		
Kernel	poly.	poly.	poly.	poly.	rad.	rad.	lin. und poly.	-
optimaler Schwellwert	0,93	0,95	0,89	0,85	0,85	0,91	0,76	0,75
Exaktheit	0,967	0,966	0,976	0,985	0,979	0,972	0,965	0,986
AUC	0,71	0,940	0,985	0,983	0,984	0,973	0,926	0,995

Tabelle 7.3: Fortsetzung von Tabelle 7.1 für das Objekt „Kurt 2D“.

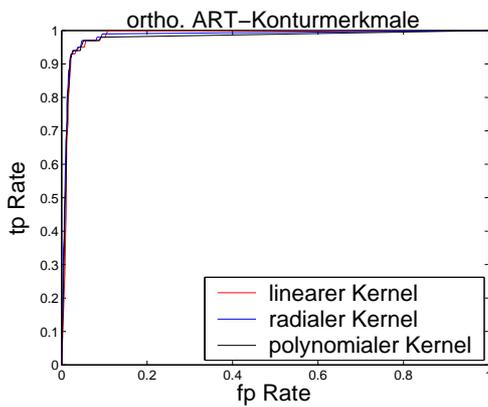




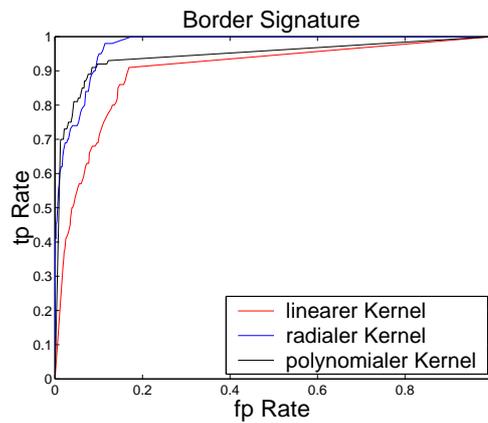
(c)



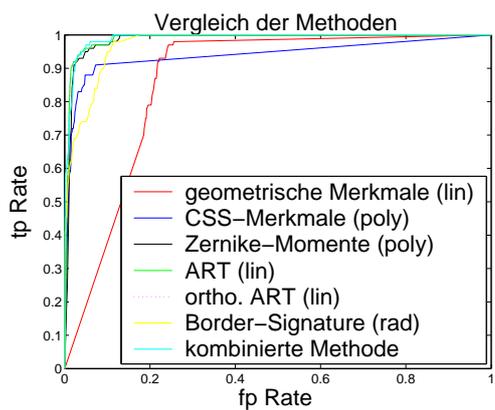
(d)



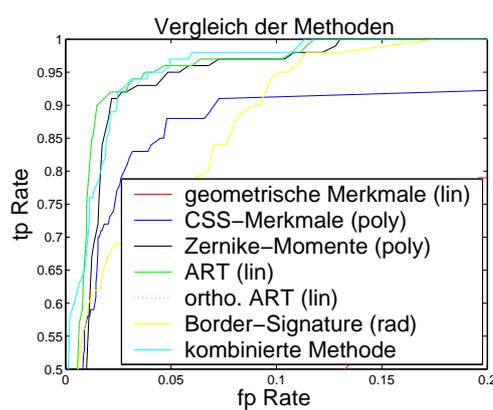
(e)



(f)



(g)



(h)

**Abbildung 7.7:** (a-f): ROC-Kurven der mit den verschiedenen Methoden trainierten SVMs (K15-K20) und der Vergleich (g)(h) dieser SVMs mit der aus ihnen kombinierten SVM (K22).

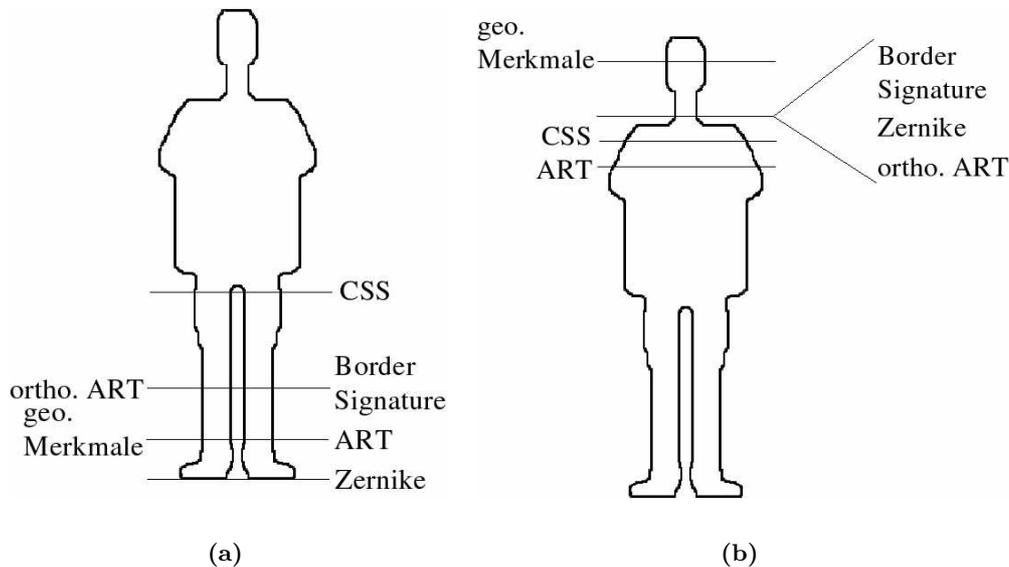
	<i>tp</i>	<i>fp</i>
geometrische Merkmale	22	1
CSS	27	11
Zernike-Momente	23	2
ART	28	3
ortho. ART	29	0
Border-Signature	26	1
Viola Jones kombiniert	22	1
Viola Jones Reflektionsbild	23	56
Viola Jones Tiefenbild	30	1

**Tabelle 7.4:** Vergleich des Viola Jones Klassifikators mit den in dieser Arbeit entwickelten Klassifikatoren. Die klassifizierten Bilder enthalten insgesamt 30 positive Beispiele.

### 7.2.1 Vergleich mit dem Viola Jones AdaBoost Klassifikator

In diesem Abschnitt wird die Performanz der in dieser Arbeit entwickelten Klassifikatoren mit dem in der bestehenden Softwarearchitektur vorhandenen Viola Jones AdaBoost Klassifikator verglichen. Dazu wurden zehn Testscans aufgenommen aus denen jeweils drei Ansichten klassifiziert werden. Da der Viola Jones AdaBoost Klassifikator als diskreter Klassifikator implementiert ist, kann für ihn keine ROC-Kurve erstellt werden. Statt dessen werden die *tp* und *fp* Klassifikationsergebnisse gezählt und für alle Klassifikatoren aufgetragen. Bei den verschiedenen Ansichten wird darauf geachtet, dass der Mensch vollständig zu sehen ist und das Bild richtig segmentiert wird. Der Testdatensatz besteht aus insgesamt 30 Menschen von denen 9 mit geschlossenen Beinen gescannt wurden. Die in dieser Arbeit entwickelten Klassifikatoren werden mit den in Abschnitt 7.2 ermittelten optimalen Schwellwerten betrieben. Das Ergebnis des Vergleichs ist in Tabelle 6.4 aufgetragen. Es ist zu sehen, dass die im letzten Abschnitt ermittelte Performanzreihenfolge der Klassifikatoren sich auch in diesem Experiment widerspiegelt. Ausnahme bilden die Zernike-Momente, die eine schlechtere Performanz liefern. Dies kann an dem relativ kleinen Testdatensatz oder der Wahl des optimalen Schwellwertes liegen. Im Vergleich mit dem Viola Jones Klassifikator stellt sich heraus, dass der aus Tiefen- und Reflektionsbild kombinierte Klassifikator ein konservativer Klassifikator ist, der nur eine *fp*-Klassifikation erzeugt. Der Viola Jones Klassifikator der ausschließlich das Reflektionsbild verwendet ist ein liberaler Klassifikator, der etwa doppelt so viel *fp*-Klassifikationen wie *tp*-Klassifikationen erzeugt. Der Viola Jones Klassifikator der ausschließlich das Tiefenbild verwendet ist für diesen Testdatensatz ein nahezu optimaler Klassifikator, der alle positiven Beispiele erkennt und dabei nur eine *fp*-Klassifikationen erzeugt. Bei den konturorientierten Verfahren schneidet der Klassifikator der die ART-Repräsentation mit orthogonalen, radialen Basisvektoren verwendet am besten ab. Er erkennt 29 der 30 positiven Beispiele und erzeugt keine *fp*-Klassifikation. Somit ist die Performanz mit dem Viola-Jones Tiefenbild Klassifikator vergleichbar. Beim CSS-Verfahren fällt auf, dass es mit Abstand am meisten *fp*-Klassifikationen unter den konturorientierten Verfahren erzeugt. Dies kann an dem relativ geringen optimalen Schwellwert von 0,76 liegen. Zu den *fp*-Klassifikationen der ART-Repräsentation ist zu sagen, dass diese ausschließlich durch nahezu kreisrunde Konturen erzeugt werden.

Zum Auswertungsverfahren muss betont werden, dass die zur Klassifikation herangezogenen Ansichten unter der Voraussetzung der richtigen Segmentierung ausgewählt werden. Wird diese Voraussetzung nicht berücksichtigt besitzt der Viola Jones Klassifikator ein deutlich besseres Ergebnis



**Abbildung 7.8:** Vergleich der Robustheit der Klassifikatoren gegenüber Verdeckung von unten (a) bzw. von oben (b).

als die in dieser Arbeit entwickelten Klassifikatoren. So erkennt der Viola Jones Klassifikator sehr kleine weit entfernte Objekte, deren feine Strukturen durch den Segmentierungsalgorithmus der konturorientierten Klassifikatoren zerstört wird. Ein weiterer Vorteil des Viola Jones Klassifikators der ebenfalls durch die fehlende Segmentierung entsteht, sind die geringeren Anforderungen an die zu lernenden Objekte. Sie müssen keine relativ große, planare Seitenansicht besitzen, wie es bei den konturorientierten Verfahren der Fall ist.

Während des Vergleichs ist außerdem aufgefallen, dass das Ergebnis des Viola Jones Klassifikators stärker unter verschiedenen Ansichten des selben Scans variiert, als es für konturorientierten Verfahren der Fall ist. Ferner sind einige konturorientierte Verfahren robuster gegenüber Verdeckung (vgl. folgender Abschnitt). Ein Nachteil in der praktischen Anwendung des Viola Jones Klassifikators liegt darin, dass sowohl das Tiefenbild als auch das Reflektionsbild benötigt werden und die Aufnahme des Reflektionsbildes zeitintensiv ist.

### 7.2.2 Klassifikation von verdeckten Konturen

Obwohl die Klassifikatoren in dieser Arbeit nicht unter dem Aspekt entwickelt wurden auch verdeckte Konturen zu erkennen, hat sich während der Untersuchungen im letzten Abschnitt gezeigt, dass einige der Verfahren relativ robust gegenüber Verdeckung sind. In Abbildung 7.8 sind die Ergebnisse aller Verfahren zu sehen. Der zu der jeweiligen Methode gehörige Strich markiert die maximale Abschnittskante, bei der das Restobjekt noch erkannt wird. Es ist zu sehen, dass die CSS-Repräsentation bei der Kontur eines Menschen dessen Beine verdeckt sind deutlich robuster ist als alle anderen Verfahren. Bei Verdeckung von oben schneidet das ART-Verfahren jedoch besser als das CSS-Verfahren ab.



## Kapitel 8

# Zusammenfassung und Ausblick

Damit Roboter mit ihrer Umwelt interagieren können, müssen sie wissen, welche Objekte sich in dieser Umwelt befinden. Diese Arbeit stellt einen konturbasierten Ansatz vor, der mit Hilfe von Tiefenbildern eines auf dem Roboter installierten Laserscanners Objekte in dessen Umgebung erkennt. In dieser Arbeit wurde ein vollständiges Bildverarbeitungssystem für Tiefenbilder entwickelt. Zunächst wird eine Interpretation des Tiefenbildes vorgenommen, die 3D-Datenpunkte entfernt, die zum Boden der gescannten Szene gehören. Danach wird das Tiefenbild aus den aufgenommenen 3D-Datenpunkten generiert, indem die Entfernung jedes Punktes zum Scanner berechnet und einem Grauwert zugewiesen wird. Da die Anzahl der Datenpunkte nicht der Anzahl der Pixel im Tiefenbild entspricht, muss zwischen den einzelnen Grauwerten der 3D-Datenpunkte interpoliert werden, um jedem Pixel einen Grauwert zuweisen zu können. Um die Segmentierung des Tiefenbildes zu erleichtern, wird diese Interpolation an Entfernungssprüngen nicht durchgeführt. Dadurch werden Objekte, die sich von ihrem Hintergrund abheben, mit einem schwarzen Rand umgeben. Die Konturen des Tiefenbildes werden extrahiert, indem ein adaptives Schwellwertverfahren das Bild binarisiert und die Konturen anschließend mit einem Konturverfolgungsalgorithmus erfasst werden.

Zur Beschreibung der Konturen werden geometrische Merkmale, Hu- und Zernike-Momente, die Curvature-Scale-Space-Repräsentation, Border-Signature sowie die Angular-Radial-Transformation herangezogen. Diese Verfahren bilden die Kontur in einen Merkmalsraum ab, in dem optimalerweise für jede Objektklasse ein Cluster entsteht, der sich mit Clustern anderer Objektklassen nicht überlappt. Um zwischen diesen Clustern Trennungshyperebenen in den Merkmalsraum zu legen, wird in dieser Arbeit eine Support-Vector-Machine eingesetzt.

Um die Performanz zu bestimmen, die ein Klassifizierer besitzt, wird für jedes Objekt ein Testdatensatz erstellt und mittels Receiver-Operating-Characteristics-Analyse bestimmt, wie gut der jeweilige Klassifizierer auf dem Testdatensatz ist. Anhand der ROC-Kurven wird bestimmt, welche Klassifizierer kombiniert werden können, um ein besseres Klassifikationsergebnis zu erhalten.

Grundsätzlich liefert diese Arbeit das Ergebnis, dass das Hauptproblem nicht in der Beschreibung und Klassifikation der Konturen, sondern in deren Extraktion liegt. So ist die Entfernung der Bodenpunkte nicht fehlerfrei, so dass das Objekt nicht immer von diesem segmentiert werden kann. Außerdem muss der Abstand, den ein Objekt zu seinem Hintergrund besitzt, groß genug sein, damit keine Grauwertinterpolation vorgenommen wird oder zumindest ein Grauwertsprung erkennbar ist. Weiterführende Arbeiten sollten deshalb im Bereich der Tiefenbildsegmentierung unternommen werden, da so ein größerer Performanzgewinn des Gesamtsystems erzielt werden kann, als wenn wei-

tere konturbeschreibende Merkmale entwickelt werden.

Weiterführende Arbeiten zur Segmentierung können das oft verwendete Wasserscheiden-Verfahren implementieren. Es kann ferner das Intensitätsbild des Laserscanners zur Hilfe genommen werden, wie es in [71] beschrieben ist. Dort wird ein fusioniertes Kantenbild aus Tiefenbild und Intensitätsbild erstellt, auf das das Wasserscheiden-Verfahren angewandt wird. Die Verwendung von Intensitätsbildern hätte jedoch den Nachteil, dass deren Erzeugung im Vergleich zur Tiefenbildgenerierung zeitaufwendiger ist.

Ein weiterer Ansatz könnte versuchen, die Segmentierung schon vor der Tiefenbildgenerierung durchzuführen, wie es in Abschnitt 4.4.2 schon in Ansätzen angewandt wird. Da viele Verfahren zur Segmentierung von 3D-Punktwolken bestehen, könnten diese vor der Tiefenbildgenerierung angewandt werden oder mit Segmentierungsverfahren des Tiefenbildes kombiniert werden.

Des Weiteren kann die Objekterkennung auf teilweise verdeckte und unterschiedliche Ansichten der Objekte ausgedehnt werden. Für teilweise verdeckte Objekte gibt es zwei unterschiedliche Ansätze. Der erste Ansatz versucht, Verdeckungen im Tiefenbild auszugleichen [57, 9]. Der zweite Ansatz verwendet Merkmale, die lokale Informationen des Objektes verwenden (z.B. Curvature-Scale-Space) und somit die Klassifikation von verdeckten Objekten ermöglichen. Um verschiedene Ansichten eines Objektes zu klassifizieren, wird in [40] mit Hilfe der CSS-Repräsentation bestimmt, wieviele und welche Ansichten notwendig sind, um die Kontur des Objektes, das sich in einer beliebigen Pose befindet, klassifizieren zu können.

## Anhang A

# Freeman-Kettencode

Eine einfache, eindimensionale Konturrepräsentation ist der Freeman-Kettencode (auch Chain-Code). Er verwendet zur Repräsentation der Kontur die Richtungsvektoren zwischen den Randpixeln der Kontur. Hierbei wird jeder Richtung nach dem Schema in Abbildung A.1 ein Wert zwischen Null und Sieben zugeordnet. Der Kettencode entsteht, indem von einem definierten Anfangspunkt aus die Kontur Pixel für Pixel durchlaufen und die entsprechende Richtungszahl an den Kettencode angehängt wird.

Neben der Datenreduktion besitzt der Kettencode den Vorteil, dass einfache Merkmale der Form wie Konkavität bzw. Konvexität, Fläche und Umfang leicht bestimmbar sind [32].

Der Kettencode besitzt den Nachteil, dass er sensitiv gegenüber Rauschen, Skalierung und Rotation ist. Aus diesem Grund findet er in der hier beschriebenen einfachen Grundform keine Anwendung in der Objekterkennung [10].

3	2	1
4		0
5	6	7

**Abbildung A.1:** Richtungsschema zur Erzeugung des Freeman Kettencodes



## Anhang B

# Curvature-Scale-Space-Matching-Verfahren nach Mokhtarian

Zusammen mit der CSS-Repräsentationstechnik wird von Mokhtarian ein mögliches Matching-Verfahren beschrieben, das Konturen anhand ihrer CSS-Repräsentation vergleicht [41]. Die Information einer Kontur wird in der CSS-Repräsentation in den Koordinaten der Maxima des CSS-Bildes komprimiert. Folgendes Verfahren ermöglicht es, die Koordinaten verschiedener Konturen miteinander zu vergleichen, und somit Ähnlichkeiten zwischen Konturen zu bestimmen [42].

1. Zunächst werden die Koordinaten der Maxima aus dem CSS-Bild der Kontur extrahiert. Sie werden bezüglich der Parametrisierung  $u$  auf das Intervall  $[0, 1]$  normalisiert und bezüglich  $\sigma$  sortiert.
2. Das größte Maximum der zu untersuchenden Kontur wird mit jedem größten Maximum der Konturen der Objektdatenbank verglichen. Falls die Differenz der  $\sigma$ -Koordinaten unter einem definierten Wert liegt, wird für die entsprechenden Konturen ein Verschiebeparameter  $\alpha$  gemäß Gleichung B.1 berechnet.

$$\alpha = u_m - u_i \tag{B.1}$$

Der jeweilige Verschiebeparameter ergibt sich, indem die Differenz der  $x$ -Koordinaten des jeweils größten Maximums der Datenbankkontur ( $u_m$ ) mit der aktuellen Kontur ( $u_i$ ) gebildet wird. Mit Hilfe des Verschiebeparameters wird die CSS-Repräsentation der aktuellen Kontur verschoben, so dass die größten Maxima der jeweiligen Konturen beim selben Parameter  $u$  liegen. Dieser Schritt ist wichtig, da eine rotierte Kontur eine bezüglich  $u$  translatierte CSS-Repräsentation besitzt.

3. Im nächsten Schritt werden für jeden Vergleich von zwei Maxima jeweils zwei Listen angelegt. Die erste Liste enthält alle CSS-Maxima der aktuellen Kontur, die bereits mit dem aktuellen Maximum verglichen wurden. Die zweite Liste enthält die entsprechenden Maxima der Datenbankkonturen. Das erste Element der Listen ist das jeweils größte Maximum der beiden Konturen.

4. Zum Matchen wird das nächst größte Maximum der zu klassifizierenden Kontur betrachtet, das sich nicht in der ersten Liste befindet. Dieses wird mit den Maxima der Datenbankkontur verglichen, die sich nicht in der zweiten Liste befinden. Auf diese Weise werden alle Maxima der aktuellen Kontur nacheinander einem Maximum einer Datenbankkontur zugewiesen.
5. Die Kosten eines Matches setzen sich aus zwei Anteilen zusammen. Der erste Anteil betrachtet die Differenz der einander zugeordneten Maxima. Dieser Kostenanteil berechnet sich als Summe aller euklidischen Abstände der einander zugeordneten Maxima. Der zweite Kostenanteil berücksichtigt die Möglichkeit, dass sowohl die zu klassifizierende Kontur als auch die Datenbankkontur weitere Maxima besitzen kann, die keinem Maximum zugeordnet werden. Diese Kosten berechnen sich durch die Höhe bzw.  $\sigma$ -Koordinate des größten, nicht berücksichtigten Maximums.

# Literaturverzeichnis

- [1] W. Abmayr. *Einführung in die digitale Bildverarbeitung*. B.G. Teubner Stuttgart, 1994.
- [2] M. Baccar, L.A. Gee, R.C. Gonzalez, and M.A. Abidi. Segmentation of Range Images via Data Fusion and Morphological Watersheds. *Pattern Recognition*, 29(10):1673 – 1687, 1996.
- [3] W. Bailer, F. Höller, A. Messina, D. Airola, P. Schallauer, and M. Hausenblas. State of the Art of Content Analysis Tools for Video, Audio and Speech. Technical report, Presto Space, 3 2005.
- [4] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509 – 522, 2002.
- [5] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.
- [6] M. Bober. MPEG-7 Visual Shape Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716 – 719, 2001.
- [7] M. Brocke. *Statistische Ereignisdetektion in Bildfolgen*. PhD, Ruprecht-Karls-Universität Heidelberg, 2002.
- [8] J. Canny. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Machine Intelligence*, 8:679 – 698, 1986.
- [9] U. Castellani, S. Livatino, and R.B. Fisher. Improving Environment Modelling by Occlusion Surface Completion. Technical report, University of Verona, Aalborg University and University of Edinburgh, 2002.
- [10] F. A. Cheikh. *MUVIS: A System for Content-Based Image Retrieval*. PhD, Tempere University of Technology, 2004.
- [11] M.S. Drew, T.K. Lee, and A. Rova. Shape Retrieval with Eigen-CSS Search. Technical report, School of Computing Science, Simon Fraser University, Vancouver, B.C., Canada V5A 1S6, 2 2005.
- [12] T. Fawcett. Roc Graphs: Notes and Practical Considerations for Researchers. Technical report, HP Laboratories, MS 1143, 1501 Page Mill Road, Palo Alto, CA 94304, 3 2004.
- [13] R. Ferber. *Information Retrieval Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. dpunkt Verlag, 2003.

- 
- [14] P.A. Flach. The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics. Technical report, Department of Computer Science, University of Bristol, Woodland Road, Bristol BS8 1UB, United Kingdom, 2003.
- [15] D. Forsyth and J. Ponce. *Computer Vision. A Modern Approach*. Prentice Hall PTR, 2002.
- [16] S. Frintrop. Robuste Roboterlokalisierung mit omnidirektionaler Bildsensorik. Master's thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2001.
- [17] T. Hanning. *Vektorielle Mehrniveaupassung Anwendungen in der Bildsegmentierung*. PhD, Fakultät für Mathematik und Informatik der Universität Passau, 2001.
- [18] D. He, Q. Sun, and Q. Tian. A Secure and Robust Object-Based Video Authentication System. *EURASIP Journal on Applied Signal Processing*, 14:2185 – 2200, 2004.
- [19] A. Hoover, G.J.-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D.B. Goldgof, K. Bowyer, D.W. Eggert, A. Fitzgibbon, and R.B. Fisher. An Experimental Comparison of Range Image Segmentation Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):675 – 689, 1996.
- [20] M.K. Hu. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, IT-8:179 – 187, 1962.
- [21] M. Hüppi and D. Kuhn. Chromatische Steinklassifizierung. Master's thesis, Hochschule Rapperswil HSR, 2000.
- [22] Intel Corporation. *Open Source Computer Vision Library*, original issue edition, 12 2000.
- [23] A.C. Jalba, M.H.F. Wilkinson, and J.B.T.M. Roerdink. CPM: A Deformable Model for Shape Recovery and Segmentation Based on Charged Particles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1 – 16, 2004.
- [24] X. Jiang. A Adaptive Contour Closure Algorithm and Its Experimental Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1252 – 1264, 2000.
- [25] X. Jiang and H. Bunke. *Dreidimensionales Computersehen Gewinnung und Analyse von Tiefenbildern*. Springer Verlag, Berlin Heidelberg, 1997.
- [26] X. Jiang and H. Bunke. Edge Detection in Range Images Based on Scan Line Approximation. *Computer Vision and Image Understanding*, 73(2):183 – 199, 1999.
- [27] E. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD, Carnegie Mellon University, 1997.
- [28] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, pages 321 – 331, 1988.
- [29] A. Khotanzad and Y.H. Hong. Invariant Image Recognition by Zernike Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489 – 497, 1990.
- [30] V. Kindratenko. *Development and Application of Image Analysis Techniques for Identification and Classification of Microscopic Particles*. PhD thesis, University of Antwerp, Belgium, 1997.

- 
- [31] S. Lang. *Multimodale Aufmerksamkeitssteuerung für einen mobilen Roboter*. PhD, Technische Fakultät Universität Bielefeld, 2005.
- [32] A. D. Lattner. Automatische Generierung von Klassifikationsregeln aus Farb-, Textur- und Konturmerkmalen durch Induktives Lernen. Master's thesis, Universität Bremen, 2000.
- [33] B. Leibe, G. Hetzel, and P. Levi. Local Feature Histograms for Object Recognition from Range Images. Technical report, Universität Stuttgart, Breitenwiesenstr. 20-22, D-70565 Stuttgart.
- [34] W. Leupold, R. Conrad, S. Völkel, G. Große, Prof. R. Fucke, H. Nickel, and H. Mende. *Analysis für Ingenieure*. Verlag Harri Deutsch, Frankfurt/M. u. Zürich, 1972.
- [35] S.X. Liao. *Image Analysis by Moments*. PhD, The Department of Electrical and Computer Engineering The University of Manitoba, 1993.
- [36] M. Martin-Fernández and C. Alberola-López. An approach for contour detection of human kidneys from ultrasound images using Markov random fields and active contours. *Medical Image Analysis*, 9:1 – 23, 2005.
- [37] J. Milgram, M. Cheriet, and R. Sabourin. Estimating accurate multi-class probabilities with support vector machines. Technical report, Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle, École de Technologie Supérieure, Université du Québec 1100, rue Notre-Dame Ouest, Montréal, Canada, 2005.
- [38] N. Milstein. Image Segmentation by adaptive Thresholding. Technical report, Technion Israel Institute of Technology, The Faculty for Computer Sciences, 1998.
- [39] F. Mokhtarian. Silhouette-Based Occluded Object Recognition Through Curvature Scale Space. *Machine Vision and Applications*, 10(3):87 – 97, 1997.
- [40] F. Mokhtarian and S. Abbasi. Automatic Selection of Optimal Views in Multi-view Object Recognition. In *BMVC*, 2000.
- [41] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and Efficient Shape Indexing through Curvature scale space. In *BMVC*, 1996.
- [42] F. Mokhtarian and M. Bober. *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization*. Kluwer Academic Publishers, Dordrecht / Boston / London, 2003.
- [43] F. Mokhtarian and A.K. Mackworth. Scale-Based Description and Recognition of Planar Curves and Two Dimensional Shapes. *IEEE Trans Pattern Analysis and Machine Intelligence*, 8(1):34 – 43, 1986.
- [44] F. Mokhtarian and R. Suomela. Robust Image Corner Detection Through Curvature Scale Space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1378 – 1381, 1998.
- [45] Morphological Filtering for Image Enhancement and Feature Extraction. P. maragos. Chapter 3.3 for the Book: *The Image and Video Processing Handbook*, 2nd edition, Edited by A. C. Bovik, to be published by Academic Press, 2005.

- 
- [46] A. Nüchter, H. Surmann, and J. Hertzberg. Automatic Classification of Objects in 3D Laser Range Scans. In *Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS '04)*, pages 963 – 970, Amsterdam, The Netherlands, March 2004.
- [47] A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann. 3D Mapping with Semantic Knowledge. Technical report, Universität Osnabrück, Universität Hannover, Fraunhofer Institut für autonome Systeme (AIS), Albrechtstraße 28, D-49069 Osnabrück, 2005.
- [48] G. Pignalberi. Tuning Range Image Segmentation by Genetic Algorithm. *EURASIP Journal on Applied Signal Processing*, 3(8):780 – 790, 2003.
- [49] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.
- [50] P. Pudil, F.J. Ferri, J. Novovicova, and J. Kittler. Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions. In *Proceedings of the IEEE Intl. Conf. on Pattern Recognition*, pages 279 – 283, 1994.
- [51] J. Ricard, D. Coeurjolly, and A. Baskurt. Generalization of Angular Radial Transform. Technical report, LIRIS, Laboratoire d'Informatique en Image et Systèmes d'information, FRE 2672 CNRS, INSA Lyon, UCB Lyon 1, EC Lyon, Univ. Lyon 2, 2004.
- [52] M. Safar, C. Shahabi, and X. Sun. Image Retrieval By Shape: A Comparative Study. *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, 4(1):141 – 144, 2000.
- [53] B. Schölkopf and A.J. Smola. *Learning with Kernels Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2001.
- [54] P. Soille. *Morphologische Bildverarbeitung*. Springer Verlag, 1998.
- [55] S.A. Solla, T.K. Leen, and K.-R. Müller. Large Margin DAGs for Multiclass Classification. *MIT Press*, pages 547 – 553, 2000.
- [56] R. Steinbrecher. *Bildverarbeitung in der Praxis*. Oldenbourg Verlag, München Wien Oldenbourg, 1993.
- [57] F. Stulp, F. Dell'Acqua, and R. Fisher. Reconstruction of surfaces behind occlusions in range images. Technical report, Division of Informatics University of Edingburgh, 2000.
- [58] S. Suzuki and K. Abe. Topological Structural Analysis of Digitized Binary Images by Border Following. *Computer Vision, Graphics, and Image Processing*, 30(1):32 – 46, April 1985.
- [59] M.R. Teague. Image Analysis via the General Theory of Moments. *J. Optical Soc. Am.*, 70(8):920 – 930, 1980.
- [60] L. Thomas, F.B. Sachse C.D. Werner, and O. Dössel. Topologisch veränderbare dreidimensionale aktive konturen in der medizinischen bildverarbeitung. Technical report, Institut für Biomedizinische Technik, Universität Kaslsruhe, 5 Forrest Hill, Edingburgh EH1 2QL, 2001.
- [61] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

- 
- [62] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the Sensitivity of Support Vector Machines. Technical report, Department of Engineering Mathematics, Bristol University, United Kingdom, 1999.
- [63] P. Viola and M.J. Jones. Image Analysis via the General Theory of Moments. *International Journal of Computer Vision*, 57(2):137 – 154, 2004.
- [64] A. Wallin and O. Kübler. Complete Sets of Complex Zernike Moment Invariants and the Role of the Pseudoinvariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1106 – 1110, 1995.
- [65] H. Wang and D. Suter. A Model-Based Range Image Segmentation Algorithm Using a Novel Robust Estimator. In *3rd International Workshop on Statistical and Computational Theories of Vision*, Nice, France, October 2003.
- [66] P. Watanachaturaporn, P.K. Varshney, and M.K. Arora. Evaluation of Factors Affecting Support Vector Machines for Hyperspectral Classification. Technical report, Electrical Engineering and Computer Science Department, Syracuse University, New York, 2004.
- [67] O. Wulf, K. O. Arras, H. I. Christensen, and B. A. Wagner. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, pages 4204 – 4209, New Orleans, USA, April 2004.
- [68] H. G. Yu. Morphological Image Segmentation for Co-Aligned Multiple Images using Watersheds Transformation. Master's thesis, The Florida State University, 2004.
- [69] F. Zernike. Beugungstheorie des Schneidverfahrens und seiner Verbesserten Form der Phasenkontrastmethode. *Physica*, 1:689 – 704, 1934.
- [70] D. Zhang and G. Lu. Review of Shape Representation and Description Techniques. *Pattern Recognition*, 37:1 – 19, 2004.
- [71] Y. Zhang, Y. Sun, H. Sari-Sarraf, and M. Abidi. Impact of Intensity Edge Map on Segmentation of Noisy Range Images. *Proc. of SPIE Conf. on Three-Dimensional Image Capture and Applications III*, 3958:260 – 269, 2000.

