

---

**Forschungszentrum Karlsruhe**  
Technik und Umwelt

---

**Wissenschaftliche Berichte**  
FZKA 5987

**Parallelisierbare Erkennung  
trihedraler, starrer und  
statischer Polyeder in  
einzelnen hochauflösenden  
Abstandsbildern**

**V. G. Vogelgesang**  
Institut für Angewandte Informatik

Oktober 1997

---



Forschungszentrum Karlsruhe

Technik und Umwelt

Wissenschaftliche Berichte

FZKA 5987

Parallelisierbare Erkennung  
trihedraler, starrer und statischer Polyeder  
in einzelnen hochauflösenden Abstandsbildern \*

Volker G. Vogelgesang

Institut für Angewandte Informatik

\* Von der Fakultät Informatik der Technischen Universität Dresden genehmigte  
Dissertation

Forschungszentrum Karlsruhe GmbH, Karlsruhe  
1997

**Als Manuskript gedruckt**  
**Für diesen Bericht behalten wir uns alle Rechte vor**  
**Forschungszentrum Karlsruhe GmbH**  
**Postfach 3640, 76021 Karlsruhe**  
**Mitglied der Hermann von Helmholtz-Gemeinschaft**  
**Deutscher Forschungszentren (HGF)**  
**ISSN 0947-8620**

## Kurzfassung

Ein Problem bei der autonomen Navigation mobiler Roboter besteht darin, die aktuelle Position eines Roboters innerhalb bekannt strukturierter Umgebung zu bestimmen. Die Positionsbestimmung aufgrund interner Sensoren wie Kreisel oder wegmessender Systeme weist allerdings eine mit der Zeit wachsende Drift auf. Interne Sensoren müssen daher von Zeit zu Zeit mit Hilfe externer Sensoren an die Umgebung angepaßt werden. Andere Verfahren zur externen Positionsbestimmung können durch Erkennung dreidimensionaler Objekte anhand ihrer Form sinnvoll ergänzt werden. Da bei der Navigation auch die Positionsbestimmung Echtzeitanforderungen erfüllen muß, sind insbesondere gut parallelisierbare Verfahren von großer Bedeutung.

Zur Erfassung der Form von Szenen sind vor allem 3D-Laserscanner geeignet. Aufgrund bestimmter Vorteile bei der Bildaufnahme werden hier hochauflösende Abstandsbilder von 3D-Laserradarscannern verwendet.

Unter Verwendung des a-priori Wissens, daß Szenen aus starren und statischen Polyedern bestehen, in deren Polyederecken sich genau drei Ebenen schneiden, wird eine kantenbasierte Szenenbeschreibung des Abstandsbildes generiert. Zu erkennende Objekte werden als charakteristische Ansichten repräsentiert werden, wobei charakteristische Ansichten und Szenenbeschreibung gleiche Struktur besitzen.

Zunächst werden mittels eines speziellen Krümmungsmaßes potentielle Kantenpunkte verschiedenen Typs extrahiert. Die Position der Kantenpunkte wird in einer Lokalisierungsphase so verfeinert, daß bestimmte Kantenpunkte trotz eines für das Radarmessverfahren typischen Sensoreffekts extrahiert werden. Aus den Kantenpunkten wird eine Szenenbeschreibung aus 3D-Kanten aufgebaut, die sich vor allem durch geeignet gewählte Attribute für Polyederecken auszeichnet.

Die Erkennung von Objekten anhand der Szenenbeschreibung besteht darin, Korrespondenzen zwischen 3D-Objekt- und 3D-Szenenkanten zu finden. Zur Lösung des Korrespondenzproblems werden künstliche neuronale Netze und verwandte Ansätze vorgeschlagen und innerhalb eines einheitlichen Rahmens gegenübergestellt. Qualitative Erkennungsergebnisse werden für künstliche neuronale Netze bestimmten Typs aufgezeigt. Anhand eines Benchmarks wird ein Vergleich mit einem verwandten Verfahren durchgeführt.

## Abstract

### **Parallelizable recognition of trihedral, rigid and static polyhedrons in single high-resolution range images**

A key problem in the field of autonomous robot navigation is to determine the current position within a known environment. But the position determined by internal sensors is affected by an over the time increasing inaccuracy. Internal sensors have to be adapted from time to time by external sensors with respect to the environment. Other methods of external position determination can be supported by recognizing three-dimensional objects by shape. Because the position determination has to meet some real-time requirements, well parallelizable methods are of great importance.

Especially 3D laser scanners are suitable to grasp the shape of scene surfaces. Because of an advantage over other types of laser scanners, high-resolution range images of 3D time-of-flight laser range scanners are used.

By using a-priori knowledge, that scenes consist of rigid and static polyhedrons, where exactly three planes are intersecting each other in every vertex, edge-based descriptions of range images are generated. The objects to be recognized are represented by characteristic views, whereas characteristic views and the scene description have the same structure.

First, by a special curvature measure potential edge points of different type are extracted. Within a localisation phase the edge point position is so refined, that certain edge points are extracted in spite of a typical property of laser radar range images. From the edge points a scene description consisting of 3D-edges is generated, which distinguishes in suitably chosen attributes for polyhedron vertices.

The recognition of objects in the scene description consists in finding correspondences between 3D-object and 3D-scene edges. To solve the correspondence problem artificial neural networks and related methods are proposed and opposed to each other within an uniform framework. Qualitative recognition results are presented for artificial neural networks of certain type. The neural network is compared with a related method by using a benchmark.

# Inhaltsverzeichnis

<b>1</b>	<b>Abkürzungen und Symbolik</b>	<b>xv</b>
<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Relative externe Positionsbestimmung . . . . .	2
1.2	Modellbasierte Objekterkennung . . . . .	4
1.2.1	Paradigma der modellbasierten Objekterkennung . . . . .	4
1.2.2	Modelle . . . . .	5
1.2.3	Erkennungsverfahren . . . . .	9
1.2.4	Parallelisierbarkeit von Erkennungsverfahren . . . . .	12
1.3	Problemstellung . . . . .	13
1.4	Beschreibung des Ansatzes . . . . .	14
1.5	Kapitelübersicht . . . . .	15
<b>2</b>	<b>Bildaufnahme und Szenenwissen</b>	<b>17</b>
2.1	Bildgebende Laserscanner . . . . .	17
2.1.1	Triangulationslaserscanner . . . . .	17
2.1.2	Laserradarscanner . . . . .	18
2.2	Meßfehler von 3D-Laserradarscannern . . . . .	18
2.2.1	Meßfehler in jedem Abtastpunkt . . . . .	20
2.2.2	Meßfehler in bestimmten Abtastpunkten . . . . .	20
2.3	Sensormodell . . . . .	22
2.4	Wissen über Szenen . . . . .	23
2.4.1	Modell von Sugihara . . . . .	28
2.4.2	Linienverbindungen als Vertexmerkmale . . . . .	29
2.5	Reale Szenen und Abstandsbilder . . . . .	32
2.6	Zusammenfassung . . . . .	33

<b>3</b>	<b>Bildbereichshinweise</b>	<b>37</b>
3.1	Übersicht Kantendetektion . . . . .	37
3.2	Angepaßte Mittlere Krümmung . . . . .	41
3.3	Lokalisierung von Bildkantenpunkten . . . . .	45
3.3.1	Lokalisierungsstrategie . . . . .	45
3.3.2	Lokalisierungsverfahren . . . . .	48
3.4	Verkettung von Bildkantenpunkten . . . . .	55
3.5	Zusammenfassung . . . . .	57
<b>4</b>	<b>Kantenbasierte Szenenbereichshinweise</b>	<b>61</b>
4.1	Übersicht über kantenbasierte Szenenbereichshinweise . . . . .	61
4.2	Generieren der Szenenkantenstruktur . . . . .	64
4.2.1	Approximation von Kanten . . . . .	64
4.2.2	Gruppierung von Kanten . . . . .	66
4.2.3	Ausrichtung verdeckender Kanten . . . . .	70
4.3	Attributierung von Vertizes . . . . .	73
4.4	Diskussion der Genauigkeit der Szenenkantenstruktur . . . . .	77
4.5	Repräsentation von Szenenbereichshinweisen und Modellen . . . . .	81
4.6	Zusammenfassung . . . . .	85
<b>5</b>	<b>Parallelisierbare Objekterkennung</b>	<b>89</b>
5.1	Optimierungsproblem . . . . .	89
5.1.1	Kodierung von Korrespondenzen . . . . .	89
5.1.2	Konstruktion einer Kostenfunktion . . . . .	90
5.1.3	Problemformulierung . . . . .	97
5.2	Optimierungsverfahren der iterativen Verbesserung . . . . .	100
5.2.1	Iterative Verbesserung . . . . .	100
5.2.2	Übersicht . . . . .	101
5.2.3	Künstliche rückgekoppelte neuronale Netze . . . . .	104
5.2.4	Mean Field Annealing . . . . .	107
5.2.5	Relaxationsmarkierung . . . . .	112
5.2.6	Gegenüberstellung der Verfahren . . . . .	116
5.3	Experimentelle Ergebnisse . . . . .	121
5.3.1	Parameterbestimmung . . . . .	121
5.3.2	Resultate für das BSB-Netz . . . . .	123
5.3.3	Diskussion der Ergebnisse . . . . .	130
5.3.4	Vergleich BSB-MFA und Diskussion . . . . .	131
5.4	Zusammenfassung . . . . .	135
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>137</b>

---

A Herleitung der Linienverbindungen nach Sugihara	140
B Schätzen partieller Ableitungen digitaler Flächen	144
C Werkzeuge	149



# Abbildungsverzeichnis

1.1	Interpretationszyklus der modellbasierten Objekterkennung nach Kanade . . . . .	4
1.2	Verarbeitungsschritte des Objekterkennungsansatzes . . . . .	14
2.1	Abstandsmessung bei dem Radarverfahren . . . . .	18
2.2	Verwendete Laserradarscanner mit verschiedener Abtasteinheit . . . . .	19
2.3	Technische Daten der verwendeten Laserradarscanner . . . . .	19
2.4	Entstehung des Mixed-Point Effektes . . . . .	21
2.5	Größe des Laserstrahlfußabdrucks . . . . .	22
2.6	Abtastung von Szenenflächen bezogen auf das Kugelkoordinatensystem des Sensors und Repräsentation als Abstandsbildmatrix . . . . .	23
2.7	Auswirkung der ungleichförmigen Abtastung von Szenenkanten . . . . .	24
2.8	Perspektivische Projektion einer synthetischen Testszene, Abstandsbild der synthetischen Testszene in Grauwertkodierung und perspektivische Projektion der abgetasteten, dreidimensionalen Meßpunkte . . . . .	25
2.9	Kantentypen . . . . .	26
2.10	Verdeckende, verdeckte und teilweise sichtbare Linien in Strichzeichnungen . . . . .	27
2.11	Zwischen Polyedern erlaubte Kontaktarten . . . . .	29
2.12	Angepaßte und ergänzte Linienverbindungen von Sugihara als Vertextypen . . . . .	30
2.13	Perspektivische Projektion zur Bestimmung der Vertextypen . . . . .	31
2.14	Photographie und Abstandsbild einer realen Testszene . . . . .	32
2.15	Versuchsaufbau des Laserradarscanners . . . . .	33
2.16	Perspektivische Projektion der dreidimensionalen Abtastpunkte der realen Testszene und Wirkung des Mixed-Point Effektes . . . . .	34
3.1	Mittlere und angepaßte Mittlere Krümmung . . . . .	43
3.2	Krümmungsbilder des Abstandsbildes der ersten Testszene . . . . .	45
3.3	Strategie zur Lokalisierung verdeckender Punkte anhand des Krümmungsvorzeichenwechsels . . . . .	46
3.4	Positionierungsfehler von lokalisierten verdeckenden Sprungkantenpunkten . . . . .	47

3.5	Wirkungsweise von Nachbarschaftsoperatoren . . . . .	48
3.6	Löschen von Nulldurchgängen in Krümmungsvorzeichenbildern . . .	50
3.7	Lokalisierung verdeckender Sprungkantenpunkte . . . . .	52
3.8	Unterscheidung konkaver und verdeckter Bildkantenpunkte . . . . .	54
3.9	Extrahierte Bildkantenpunkte verschiedenen Typs . . . . .	55
3.10	Wirkung der Kantenpunktlokalisierung und der Medianfilterung auf Szenenpunkte . . . . .	58
4.1	Funktionsweise der iterativen Endpunktanpassung . . . . .	65
4.2	Maximales Verhältnis von Kanten- zu Kettenlänge als Kriterium zur Akzeptanz approximierter Teilketten . . . . .	66
4.3	Bildpunktketten mit Splitpunkten und zugehörige 3D-Szenenpunkte	67
4.4	Approximierte Kanten der Testszene . . . . .	68
4.5	Gruppierungsmodell eines kollinearen Kantenpaares . . . . .	69
4.6	Gruppierungsmodell eines nicht-kollinearen Kantenpaares . . . . .	69
4.7	Gruppierte Kanten der Testszene . . . . .	71
4.8	Flächentest benachbarter Kanten und Winkeltest benachbarter Linien	72
4.9	Durchführung des Flächentests nur zu geeigneten Nachbarkanten . .	73
4.10	Ausgerichtete verdeckende Kanten der Testszene . . . . .	74
4.11	Klassifikation von Linienverbindungen mit drei Linien anhand der eingeschlossenen Winkel . . . . .	75
4.12	Problemfall ähnlicher 2D-Winkel, aber unterschiedlicher 3D-Konstellationen bei der Ermittlung des Indexes von Linienverbindungen . . .	75
4.13	Vertizes der Testszene mit Vertextypen . . . . .	76
4.14	Photographie, Abstandsbild und markierte Strichzeichnung der zweiten realen Testszene . . . . .	79
4.15	Beispielhafte Abbildung einer Szenenkantenstruktur in einen Szenen- graphen . . . . .	82
4.16	Berechnung des Winkelattributes mit Hilfe von Linienverbindungen .	84
4.17	Markierte Strichzeichnung einer Szenenkantenstruktur und Strich- zeichnung einer Objektkantenstruktur mit erwarteter Markierung von Linien und Linienverbindungen . . . . .	86
4.18	Markierte Strichzeichnungen der Objektkantenstrukturen verschiede- ner charakteristischer Ansichten . . . . .	87
5.1	Bezeichnungen zur Definition der Ähnlichkeit zwischen einem Ob- jektgraphknoten und einem Szenengraphknoten . . . . .	92
5.2	Ähnlichkeitsfunktion zum Vergleich numerischer Attribute . . . . .	93
5.3	Bezeichnungen zur Definition der Ähnlichkeit zweier Knotenpaare . .	95
5.4	Strafffunktion für die Einhaltung der Zeilen- und Spaltensummenbe- dingung . . . . .	98
5.5	Strafffunktion zur Bevorzugung binärer Endzustände . . . . .	98

5.6	Modell eines formalen Neurons . . . . .	104
5.7	Stückweise lineare Ausgangsfunktion des Brain-state-in-a-box Neurons	106
5.8	Nicht-vollständige Vernetzung eines Neurons . . . . .	107
5.9	Sigmoide Funktionen zur Berechnung von Spinerwartungswerten für das Mean Field Annealing . . . . .	110
5.10	Korrespondenzmatrix und Visualisierung resultierender Korrespon- denzen . . . . .	127
5.11	Korrespondenzen zwischen Szenengraph Szene1 und Objektgraph Tisch-links-oben . . . . .	128
5.12	Korrespondenzen zwischen Szenengraph Szene1 und Objektgraph Treppe-links-oben . . . . .	128
5.13	Korrespondenzen zwischen Szenengraph Szene1 und Objektgraphen Stuhl-links-oben und Stuhl-vorne . . . . .	129
5.14	Korrespondenzen zwischen Szenengraph Szene2 und Objektgraph Stuhl-rechts-oben . . . . .	130
5.15	Synthetisches Abstandsbild und markierte Strichzeichnung der Szene des Benchmarks . . . . .	132
5.16	Verschiedene Abkühlfunktionen . . . . .	134
5.17	Temperaturabnahme einer Abkühlfunktion . . . . .	134
A.1	Klassifizierung von isolierten Vertizes . . . . .	141
A.2	Zulässige Kontaktarten zweier Polyeder . . . . .	141
A.3	Linienverbindungen nach Sugihara und polyedrische Beispielszenen .	143
B.1	Partielle Ableitungen des synthetischen Abstandsbildes der Testszene	147
C.1	Systemarchitektur und Schnittstellen . . . . .	150
C.2	Sensorsimulator LASI zur Berechnung idealer synthetischer Abstands- bilder . . . . .	151
C.3	Oberfläche von Khoros zur visuellen Programmierung . . . . .	152
C.4	Veränderung der Korrespondenzstärken der Kanten als Bildfolge . .	154
C.5	Veränderung der Korrespondenzstärken der Kanten als Bildfolge (Fort- setzung) . . . . .	155
C.6	Oberfläche von AVS zur visuellen Programmierung . . . . .	156



# Tabellenverzeichnis

3.1	Klassifikation der Verfahren zur Kantendetektion in Abstandsbildern	38
4.1	Übersicht der Ansätze zur Extraktion kantenbasierter Szenenbereichs- hinweise . . . . .	62
5.1	Schema der Iterativen Verbesserung . . . . .	101
5.2	Übersicht von Anwendungen in der Objekterkennung unter Einsatz von Verfahren der iterativen Verbesserung . . . . .	102
5.3	Brain-state-in-a-box Algorithmus . . . . .	108
5.4	Mean Field Annealing Algorithmus . . . . .	111
5.5	Algorithmus der angepaßten nicht-linearen Relaxationsmarkierung .	116
5.6	Gegenüberstellung der Optimierungsverfahren . . . . .	118



# Abkürzungen und Symbolik

Abkürzung	Begriff
BSB-Netz	Brain-state-in-a-box Netz
MFA	Mean Field Annealing
NLR-Markierung	nicht-lineare Relaxationsmarkierung
HT-Netz	Hopfield/Tank-Netz
<i>Funktionen</i>	
$r$	Abstandsbildfunktion
$k$	Umwandlung kartesische $\mapsto$ Kugel-Koordinaten
$x$	Umwandlung Kugel- $\mapsto$ kartesische Koordinaten
<i>round</i>	Rundungsfunktion
<i>Vektoren</i>	
$x$	Vektor
$x_i$	$i$ -te Komponente in $x$
<i>Matrizen</i>	
$A$	Matrix
$a_{i,k}$	Matrizelement
<i>iterative Verbesserung</i>	
$x_{i,k}$	Stärke einer Korrespondenz
$\Delta$	Schrittweite
$\pi$	Projektionsfunktion
$X$	Korrespondenzmatrix
$K$	Kostenfunktion
<i>Neuronale Netze</i>	
$n_{i,k}$	Ausgang eines Neurons
$N$	Neuronenmatrix
$w_{i,k,j,l}$	Verbindungsgewicht
$\Theta_{i,k}$	Schwelle
$\Sigma_{i,k}$	Eingangsfunktion
$m_{i,k}$	Aktivierung
$\Phi_{i,k}$	Ausgangsfunktion
$f_{lin}, f_{sig}$	lineare, sigmoide Ausgangsfunktion
$E$	Netzenergie

<i>Spinsysteme</i>	
$\langle s_{i,k} \rangle$	Spinerwartungswert
$\mathbf{S}$	Spinmatrix
$\Phi_{i,k}$	mean field eines Spins
$\beta_{i,k,j,l}$	Spininteraktion
$h_{i,k}$	externe Feldkomponente
$H$	Hamilton-Funktion
$T$	Temperatur
<i>Relaxationsmarkierung</i>	
$a_i$	zu markierendes Objekt
$\lambda_k$	Marke
$\theta_i$	Variable für zu markierendes Objekt
$P(\theta_i = \lambda_k), P_{i,k}$	Wahrscheinlichkeit einer Zuordnung $a_i \mapsto \lambda_k$
$\mathbf{P}$	Wahrscheinlichkeitsmatrix
$\tau(\theta_i = \lambda_k, \theta_j = \lambda_l), \tau_{i,k,j,l}$	Kompatibilitätskoeffizient
$Q(\theta_i = \lambda_k), Q_{i,k}$	Gesamtkompatibilität einer Zuordnung $a_i \mapsto \lambda_k$
$F$	Inkompatibilitätsfunktion
<i>Symbole in Algorithmen</i>	
$\otimes$	diskreter Faltungsoperator
$\leftarrow$	Wertzuweisung
$\llbracket \langle \text{Anweisung} \rangle \rrbracket$	Nachbarschaftsoperator

# Kapitel 1

## Einleitung

Bereits seit langem werden fahrerlose Transportsysteme in industriellen Fertigungsumgebungen erfolgreich zum Materialtransport eingesetzt. Damit zu jedem Zeitpunkt der Zustand aller Fahrzeuge des Systems definiert und bekannt ist, muß die Einsatzumgebung von vornherein durch künstliche Markierungen, sog. Landmarken, präpariert werden. Mit Hilfe spezialisierter Sensoren ist eine absolute Positionsbestimmung bezogen auf die Landmarken als Referenzpunkte möglich, die zum Beispiel in Form einer Spurführung vorliegen. Die Umgebung muß vor dem Einsatz mit teilweise hohem Aufwand durch Landmarken künstlich strukturiert werden.

Allerdings ist der Einsatz von Fahrzeugen oder mobilen Robotern auch in solchen Umgebungen erforderlich, in denen eine künstliche Strukturierung von vornherein nicht vorgesehen war und nachträglich nicht mehr möglich ist. Ein typisches Beispiel ist das Gebäudeinnere einer großtechnischen Anlage wie die eines Kernkraftwerks, wo Arbeiten in einer potentiell gefährlichen, zum Beispiel radioaktiven Umgebung durchgeführt werden müssen [Vogelgesang *et al.* 92].

In solchen Umgebungen müssen mobile Roboter daher von einem menschlichen Bediener ferngesteuert werden [Sheridan 87]. Um den Bediener bei sich oft wiederholenden Abläufen unterstützen zu können, wäre eine teilweise Autonomie mobiler Roboter in bestimmten Aufgabenbereichen wünschenswert. Mögliche Einsatzgebiete reichen von der routinemäßigen periodischen Inspektion und Überwachung von Anlagenteilen bis hin zu einfachen Wartungs- und Reparaturaufgaben. Ein Szenario für den Einsatz eines mobilen Roboters in einem Druckwasserreaktor-Kraftwerk wird zum Beispiel in [Valavanis & Saridis 92] gezeichnet.

Ein wesentlicher Fortschritt wäre bereits die spurfreie Navigation eines mobilen Roboters innerhalb einer bekannten Umgebung bestimmter Struktur. Unter Navigation ist dabei die Bestimmung von Standort und Kurs sowie dessen Einhaltung zu verstehen. Ein mobiler Roboter wird in Bezug auf die Navigation unter anderem erst dann als autonom bezeichnet, wenn er ohne externe Unterstützungsmaßnahmen jede erreichbare Position innerhalb einer bekannten Umgebung anfahren kann. Dazu ist es notwendig, die aktuelle Position und Orientierung bezogen auf ein inertiales Weltkoordinatensystem in der vorgegebenen Umweltkonstellation zu bestimmen [Knieriemen 91].

Die heute verfügbaren Sensorsysteme können in verschiedene Klassen eingeteilt werden, die sich zum Beispiel durch die Anwendungsfunktion, das Meßprinzip und die Dimensionszahl der Meßwerte unterscheiden. Sensoren für Roboter werden allgemein zwei Hauptgruppen zugeordnet [Knieriemen 91]: interne Sensoren und externe Sensoren. Interne Sensoren erfassen die roboterinternen Systemzustände wie zum

Beispiel die Geschwindigkeit der Räder. Externe Sensoren hingegen erfassen Systemzustände des Roboters in Bezug auf die externe Umgebung, wie zum Beispiel den relativen Abstand zwischen Roboter und Oberflächen der Umgebung.

Interne Sensoren wie Inertialsysteme und wegmessende Systeme haben eine mit der Zeit wachsende Drift und müssen in gewissen Zeitabständen neu an die externe Umgebung angepaßt werden. Die Anpassung erfordert daher eine externe Positionsbestimmung, die sich auf Daten externer Sensoren stützen muß. Eine Alternative oder Ergänzung zu der absoluten Positionsbestimmung ist die *relative externe Positionsbestimmung*, die sich nicht auf künstliche Landmarken bezieht.

## 1.1 Relative externe Positionsbestimmung

Die relative externe Positionsbestimmung ist sehr gut in dem Buch von Knieriemmen [Knieriemmen 91] beschrieben. Im Falle der relativen externen Positionsbestimmung wählt der Roboter selbst Objekte aus, die er als seine eigenen Ortsreferenzen betrachtet. Geeignet sind gut sichtbare, charakteristische Objekte wie Säulen, Schächte, Stufen oder Raumecken. Die ortsfesten Umweltmerkmale werden in einer vorgegebenen Landkarte vermerkt. Als Ortsreferenzen in großtechnischen Umgebungen könnten beispielsweise polyedrische Objekte wie Schächte oder Säulen im Wand- oder Deckenbereich dienen. Die relative Methode ist zwar ohne Präparierung der Umgebung durch Landmarken einsetzbar, jedoch müssen für das automatische Generieren von Referenzen eine komplette Weltmodellierung mit speziellen Verfahren zur Merkmalsextraktion geleistet werden. Bezüglich dieser Ortsreferenzen läßt sich dann die relative Position des Roboters bestimmen. Fortschrittliche Konzepte zur Positionsbestimmung kombinieren die interne mit der externen Positionsbestimmung, wobei per Koppelnavigation eine grobe Positionsschätzung erfolgt, die durch ein relatives Verfahren überprüft und eventuell korrigiert wird.

Um sich in seiner Umgebung bewegen und die dafür erforderlichen Befehle korrekt ausführen zu können, muß ein autonomer mobiler Roboter eine interne Darstellung oder eine Karte dieser Umwelt besitzen. Eine Karte kann von außen vorgegeben sein oder während des Durchfahrens der Umgebung automatisch aufgebaut werden. In beiden Fällen hängt die Information, die in die Karte aufzunehmen ist, von dem jeweiligen Einsatzzweck und von der Umgebungsstruktur ab. Eine einfach strukturierte Umgebung wird die Anforderungen an die Positionsbestimmung reduzieren.

Damit die externe relative Positionsbestimmung durchgeführt werden kann, muß das Weltmodell oder die Karte ein bestimmtes Abstraktionsniveau besitzen. Man kann geometrische, topologische und semantische Modelle unterscheiden. In der Regel wird das geometrische Modell direkt aus den Sensordaten abgeleitet. Die geometrischen Grundelemente dieser Darstellungsstufe unterscheiden sich je nach Einsatzumgebung, Sensorik und gewählter Repräsentationsform.

Das geometrische Modell wird vorwiegend für Navigationsaufgaben wie die lokale Kursplanung und lokale Positionskorrektur benötigt. Das topologische Modell repräsentiert die Einsatzumgebung durch die Lage und Anordnung von geometrischen Strukturformen. Dies können beispielsweise Kanten von Hindernissen oder konvexe Freiräume darstellen. Topologische Beziehungen sind als Graph- oder Baumstruktur darstellbar und unterstützen die globale Wegplanung. In dem semantischen Modell werden die geometrischen und topologischen Darstellungen auf ihre Bedeutungen hin untersucht und nach Möglichkeit identifiziert. Das Ergebnis dieser Abstraktion sollte allgemeinverständliche Informationen über Struktur und Objekte der Einsatzumgebung liefern. Die semantische Information ist nicht nur in der Aufgabenplanung verwertbar, sondern auch zur externen Positionsbestimmung. Sind Objekte

oder Objektteile identifiziert, lassen sie sich als Ortsreferenzen verwenden und die Position des Roboters läßt sich in Bezug auf diese Ortsreferenz ermitteln.

Ein wichtiges Merkmal zur Charakterisierung geometrischer Weltmodelle ist die Dimensionalität. Je nach Aufgabenstellung und Zielsetzung können für die Planungsaufgaben eines mobilen Roboters 2D-Darstellungen ausreichen, oder es ist eine 3D-Repräsentation erforderlich. Einige Funktionen wie die kollisionsüberwachende Fahrt oder die Kollisionsvermeidung durch Ausweichen können auf projizierte 2D-Darstellungen der Einsatzumgebung abgestützt werden. Detaillierte Kenntnisse über die Geometrie der dreidimensionalen Umwelt kann nicht nur dann erforderlich sein, wenn manipulierende Aufträge ausgeführt werden sollen, sondern auch, wenn eine externe relative Positionsbestimmung durchgeführt werden soll. Um eine externe relative Positionsbestimmung an dreidimensionalen Objekten durchführen zu können, muß man Objekte erkennen und Modellwissen über die Form der Umgebung einsetzen. Der Vorgang wird daher auch als modellbasierte Objekterkennung bezeichnet.

In der Arbeit von [Christensen *et al.* 94] wird beispielsweise ein geometrisches kantenbasiertes Weltmodell zur Vorhersage von Liniensegmenten in Videobildern und zur relativen externen Positionsbestimmung in geschlossenen Räumen verwendet. In [Lee *et al.* 94] wird ebenfalls ein modellbasiertes Verfahren für die Positionsbestimmung eingesetzt. Dabei dienen Bilder charakteristischer Szenen als Modellbilder. Durch Korrespondenzen zwischen Masken in Modellbildern mit Teilbereichen in einem aufgenommenen Bild wird die dreidimensionale Position eines mobilen Roboters in Bezug auf signifikante Szenenteile ermittelt.

Speziell zur Erkennung von Objekten anhand ihrer Form sind Abstandsbilder von Vorteil. Während die Helligkeit eines Videobildpunktes von Lichtquellen, Schattenbildung und Oberflächenbeschaffenheit abhangig ist, ist ein Bildpunkt eines Abstandsbildes nur von der Form der Umgebung abhangig. Die Erkennung von Objekten anhand ihrer Form bzw. Geometrie wird dadurch vereinfacht. Die Vorteile beider Sensordaten lassen sich aber auch erfolgreich zur Objekterkennung kombinieren [Wei 89]. Zur Aufnahme von hochauflosenden Abstandsbildern sind vor allem 3D-Laserscanner geeignet.

3D-Laserscanner lassen sich nach dem Prinzip der Abstandsmessung in das Laufzeit- oder Radarverfahren und das Triangulationsverfahren einteilen. Da das Triangulationsverfahren bestimmte Umgebungsbereiche aufgrund des Abschattungseffektes nur unvollstandig erfassen kann, wurden in diesem Ansatz Abstandsbilder eines 3D-Laserradarscanners verwendet. Zur externen Positionsbestimmung haben sich Laserradarscanner bereits bewahrt. Zur Korrektur der aktuellen Fahrzeuglage werden in [Azarm *et al.* 93] vertikale, ebene Flachen als naturliche Landmarken aus dem Abstandsbild eines 3D-Laserradarscanners extrahiert und anschlieend mit erwarteten Flachen aus einem Umgebungsmodell verglichen. bereinstimmende Daten aus dem Abstandsbild und dem Umgebungsmodell werden dann zur Lagekorrektur eingesetzt. In [Knieriemen 91] werden vertikal versetzte Abtastprofile zur Positionsbestimmung und zur modellbasierten Erkennung einfacher Objekte eingesetzt. Dabei wird jedes Profil durch Kanten approximiert, die unter bestimmten Annahmen zur Erkennung einfacher Objekte und als in die Bodenebene projizierte Linien zur Positionsbestimmung dienen.

Im Kontext der Navigation eines mobilen Roboters werden insbesondere auch an die externe Positionsbestimmung und damit an die modellbasierte Objekterkennung Echtzeitanforderungen gestellt. Gefordert sind daher solche Merkmale, die schnell aus Abstandsbildern zu extrahieren sind, und gut parallelisierbare Erkennungsverfahren, da sie das Potential besitzen, die Erkennungsgeschwindigkeit durch Einsatz

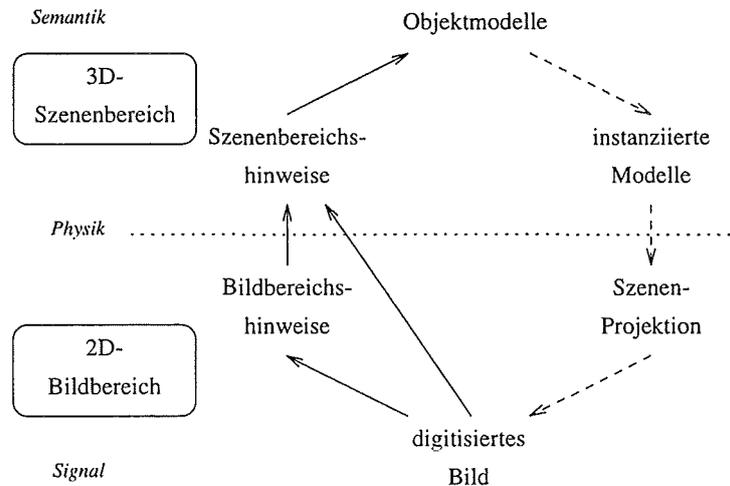


Abbildung 1.1: Interpretationszyklus der modellbasierten Objekterkennung nach Kanade (Erklärung siehe Abschnitt 1.2.1)

von Parallelrechnern zu erhöhen. Die modellbasierte Objekterkennung wird gesondert im folgenden Abschnitt erläutert.

## 1.2 Modellbasierte Objekterkennung

Einen guten Überblick über die modellbasierte Objekterkennung aus Abstandsbildern bieten die Artikel von [Besl & Jain 85b], [Chin & Dyer 86], [Arman & Aggarwal 93] und das Buch von [Suk & Bhandarkar 93]. Um Objekterkennungsprobleme in einem eindeutigen begrifflichen Rahmen beschreiben zu können eignet sich das Paradigma von [Kanade 78].

### 1.2.1 Paradigma der modellbasierten Objekterkennung

Das Paradigma von [Kanade 78] und die Modifikation nach [Nagel 79] eignen sich durch die Unterscheidung zwischen Bild- und Szenenbereich als terminologischer Rahmen (Abbildung 1.1).

In einem Bildaufnahmevorgang werden Signale eines Sensors, die beispielsweise Informationen über Abstände oder Helligkeiten besitzen, in ein digitisiertes Bild einer Szene umgesetzt. Ein Teilaspekt der Bildaufnahme ist der Projektionsvorgang. Da aus einer dreidimensionalen Szene ein zweidimensionales Bild aufgenommen wird, gehen viele Informationen über die Szene verloren. Um in der Bildmatrix nicht nur genau festgelegte Muster, sondern Objekte unter verschiedenen Ansichten erkennen zu können, ist eine abstrakte symbolische Beschreibung der Szene notwendig.

Um an diese Beschreibung zu gelangen, ist eine Umkehrung des Projektionsvorgangs nicht möglich, da die Zahlenmatrix allein hochgradig mehrdeutig ist. Eine teilweise Rekonstruktion der Szene ist daher nur dann möglich, wenn die Bildauswertung implizites und explizites Wissen über die zu erkennenden Objekte und die Einsatzumgebung verwendet.

Die Bildauswertung kann grob in die ikonische und die symbolische Bildauswertung eingeteilt werden. Die ikonische Bildauswertung arbeitet ausschließlich in dem zweidimensionalen Bildbereich, die symbolische Bildauswertung in dem dreidimensionalen Szenenbereich.

Unter Einsatz von Wissen über den Bildaufnahme-prozeß und die physikalische Beschaffenheit von Szenen werden in der Phase der ikonischen Bildauswertung Bildbereichshinweise wie Gradienten oder Regionen aus dem digitisierten Bild extrahiert. Bildbereichshinweise geben Hinweise auf die physikalische Beschaffenheit der Szene. Bildbereichshinweise lassen sich unterteilen in Bildbereichselemente (Gradienten, Krümmungen) und Bildbereichsstrukturen (Punktekette(n), Regionen), die Bildbereichselemente wiederum zueinander in Beziehung (2D Nachbarschaft) setzen.

In der Phase der symbolischen Bildauswertung werden die Bildbereichshinweise zusammen mit Kontextwissen über die physikalische Beschaffenheit von Szenen zur Extraktion dreidimensionaler Szenenbereichshinweise genutzt. Szenenbereichshinweise lassen sich ebenfalls unterteilen in Szenenbereichselemente (3D-Kanten, 3D-Flächenelemente) und Szenenbereichsstrukturen (benachbarte 3D-Kanten und 3D-Flächenelemente), die Szenenbereichselemente zueinander in Beziehung (3D-Nähe) setzen.

Szenenbereichshinweise werden durch einen Interpretationsprozeß in eine hypothetische Ausprägung eines oder mehrerer Objektmodelle umgesetzt. Solche Modelle enthalten nicht die gesamte, die das Objekt beschreibende Information, sondern nur die für eine robuste Erkennung signifikanten Komponenten. Erkennungsmodelle können konstruiert oder aus Bildern automatisch erzeugt werden.

Da ein Modell im allgemeinen freie Parameter wie Position und Orientierung enthält, wird es auch als generische Beschreibung bezeichnet. Durch geeignete Festlegung dieser Parameter erhält man eine Ausprägung dieser Beschreibung.

Durch Projektion der hypothetischen Ausprägung in den Bildbereich, die sog. Szenenprojektion, läßt sich eine Szenenskizze, die beispielsweise nur Kantenelemente enthält, oder ein synthetisches Bild erzeugen. Szenenskizze und synthetisches Bild können potentiell zum Verifizieren, Verfeinern oder zum Suchen von Bildbereichshinweisen dienen. Durch Extraktion weiterer Bildbereichshinweise schließt sich die Verarbeitungskette, hypothetische Ausprägungen können in weiteren Interpretationszyklen verworfen oder bestätigt werden.

Es wurde schon eine Vielzahl von Objekterkennungssystemen realisiert, die Abstandsbilder verwenden [Vogelgesang 91a]. Einen kurzen und prägnanten Überblick bietet [Fan 90]. Die ihnen gemeinsame Grundstruktur wird in den Überblickartikeln von [Besl & Jain 85b], [Arman & Aggarwal 93] und in [Fan 90] unter anderem nach verwendeten Modellen und nach verwendeten Verfahren zur Interpretation klassifiziert.

### 1.2.2 Modelle

Nach der räumlichen Ausdehnung von Modellmerkmalen unterscheidet man folgende Modelle:

- unstetigkeitsbasierte Modelle
- flächenbasierte Modelle
- volumetrische Modelle

Volumetrische Modelle haben zwar den Vorteil, dreidimensionale Objekte effizient zu beschreiben, jedoch können sie nicht direkt für die Erkennung benutzt werden [Arman & Aggarwal 93]. Sie werden daher in diesem Zusammenhang nicht näher beschrieben.

Ein anderes Unterscheidungsmerkmal von Modellen besteht darin, ob Merkmale globale oder lokale Objekteigenschaften darstellen. Während globale Objekteigenschaften wie der Flächeninhalt von der ganzen sichtbaren Objektfläche abhängen, sind lokale Objekteigenschaften wie die Existenz von Kanten oder Ecken mit räumlich begrenzter Ausdehnung. Ein Beispiel eines globalen Flächenmodells ist das Modell der Gauß'schen Kugel. Im Modell der Gauß'schen Kugel werden Orientierungen der Oberflächennormalen in Bildpunkten auf entsprechende Punkte einer Einheitskugel abgebildet. Das resultierende Orientierungshistogramm eines Szenenobjektes kann als Merkmal mit Histogrammen von Objektmodellen verglichen werden. Allerdings funktioniert das Verfahren nur für konvexe Objekte, durch die Abbildung auf die Gauß'sche Kugel gehen Informationen über Abmessungen verloren, und Szenen müssen bereits in hypothetische Szenenobjekte segmentiert sein. Wie bei allen Modellen, die globale Merkmale repräsentieren, ist auch die Gaußkugel sehr empfindlich gegenüber teilweiser Verdeckung. Daher werden globale Modelle hier nicht näher betrachtet.

### Unstetigkeitsbasierte Modelle

Unstetigkeitsbasierte Modelle repräsentieren diejenigen Flächenpunkte, in denen sich eine Flächeneigenschaft wie zum Beispiel die lokale Krümmung ändert. Solche Punkte lassen sich im allgemeinen durch dreidimensionale Kurvenstücke wie Strecken oder parametrisch polynomielle Kurven beschreiben. Die Repräsentationen lassen sich unterteilen in die frühe Flächenskizze und die Unstetigkeitsmarkierung [Arman & Aggarwal 93].

In der *frühen Flächenskizze* werden vier verschiedene Typen an Unstetigkeiten repräsentiert: Sprungkanten, an denen die Abstandsbildfunktion unstetig ist, Dachkanten, an denen die Flächennormale unstetig ist, glatte Verbindungen, wo die Krümmung unstetig ist, und Schultern, die aus der Kombination zweier Dachkanten entstehen. Die Unstetigkeiten verschiedenen Typs werden als Schnittkurven von glatten Flächen interpretiert, für die eine Flächenbeschreibung vorgeschlagen wird. Die *frühe Flächenskizze* ist vor allem für die Repräsentation glatter Flächen geeignet.

Die Technik der *Unstetigkeitsmarkierung* repräsentiert Unstetigkeiten in einem Kanten-Linienverbindungsgraphen. Eine Kante ist ein aus einem Abstandsbild extrahiertes dreidimensionales Kurven- oder Geradenstück und eine Linie ist das Abbild einer Kante im Bild. Jede Linie ist durch einen Linientyp markiert, der die lokale räumlich-geometrische Ausprägung einer Kante charakterisiert. Die Stellen, in denen Linien mit gemeinsamem Endpunkt zusammentreffen, werden als Linienverbindungen und alle Linien zusammen als Strichzeichnung bezeichnet. Der Kanten-Linienverbindungsgraph stellt einen attributierten, ungerichteten Graphen dar. Für jede Linienverbindung existiert ein Knoten und für jede Linie ein Bogen zwischen Knoten in dem Graphen. Da Linien durch Merkmale der entsprechenden dreidimensionalen Kante attributiert sind, wird der Graph als Kanten-Linienverbindungsgraph bezeichnet. Erfüllt eine Szene bestimmte Bedingungen, treten Linienverbindungen auf, die sich in nur wenige Klassen einteilen lassen. Eine Linienverbindung stellt daher ein Merkmal mit hohem Informationsgehalt dar, um die Objektansichten zu charakterisieren.

Die einzige dem Autor bekannte Arbeit, in der ein Kanten-Linienverbindungsgraph aus Abstandsbildern automatisch erzeugt wurde, ist die Arbeit in [Godin & Levine 89]. Die Repräsentation wird hier näher beschrieben, da sich der hier vorgestellte Ansatz daran anlehnt. In der Arbeit von Godin und Levine wird der Kanten-Linienverbindungsgraph als Szenenbereichsstruktur für Abstandsbilder von Objekten eingesetzt, die sich durch gerade und gekrümmte Kanten auszeichnen. Die entsprechenden Linien bilden Linienverbindungen, in denen idealerweise bis zu drei Linien zusammentreffen. Jede Linie ist durch einen von vier möglichen Linientypen markiert. Markierte Linienverbindungen werden nochmals grob in fünf verschiedene Klassen eingeteilt. Die Klassen dienen neben der 2D-Bildkoordinate, der 3D-Szenenkoordinate und der Anzahl der zusammentreffenden Linien als Attribute der Linienverbindung. Die den Linien entsprechenden Kanten werden durch einen von vier Kantentypen und durch die 3D-Koordinaten der Kantenendpunkte beschrieben.

### Flächenbasierte Modelle

Flächenmodelle lassen sich sinnvoll einteilen in globale und lokale Flächenmodelle [Fan 90]. Das globale Flächenmodell der Gauß'schen Kugel wurde bereits erwähnt. Globale Flächenmodelle werden hier jedoch nicht berücksichtigt.

*Lokale Flächenmodelle* definieren ein Objekt durch die das Objekt begrenzenden dreidimensionalen Flächenelemente und durch strukturelle Beziehungen zwischen den Flächenelementen. Die Flächenelemente können als Ebenen oder als parametrisch polynomielle Flächen beschrieben werden [Besl & Jain 85b]. Man unterscheidet weiterhin segmentierte und approximierende Flächenbeschreibungen.

*Approximierende Flächenmodelle* nähern die Oberfläche von Objekten durch einfache Flächenelemente an, ohne physikalische Merkmale wie Objektkanten zu berücksichtigen. An solchen Stellen werden die Flächenelemente solange in feinere unterteilt, bis die Approximation genau genug ist. Die Anzahl der Flächenelemente ist meist sehr hoch. Die Beziehungen zwischen Flächenelementen und die Flächenelemente selbst besitzen keine physikalische Signifikanz. Die Flächenapproximation ist meist blickpunktabhängig. Approximierte Flächenmodelle sind vor allem in der rechnergestützten Konstruktion und in der Computergraphik von Bedeutung, da eine detailgetreue Repräsentation im Vordergrund steht.

*Segmentierte Flächenmodelle* hingegen nutzen charakteristische Merkmale von Objekten, um Abstandsbilder in Regionen zu segmentieren und Flächenelemente an Regionen anzupassen. Die strukturellen Beziehungen zwischen Flächenelementen orientieren sich ebenfalls an physikalisch markanten Merkmalen. Die Segmentierung in Flächen ist daher blickpunktunabhängig. Ein Beispiel für ein segmentiertes Flächenmodell findet sich in [Fan 90]. Das Flächenmodell repräsentiert Objekte, die durch Kanten begrenzte Flächen bis zu zweiter Ordnung charakterisiert sind. Jede Objektansicht wird durch einen attributierten Graphen repräsentiert. Jeder Knoten enthält geometrische Informationen über ein Flächenstück und jeder Bogen zwischen Knoten repräsentiert Beziehungen zwischen Flächenstücken. Als Knotenattribute werden die folgenden Flächenmerkmale verwendet: Flächeninhalt, Flächennormale, durchschnittliche Flächenkrümmung, geschätzter Grad an Verdeckung und der Flächenmittelpunkt. Als Bogenattribute dient die Wahrscheinlichkeit, daß zwei Flächenstücke verbunden sind, und entlang welchem Kantentyp zwei Flächen verbunden sind bzw. eine Fläche von einer anderen verdeckt wird.

Eine weitere Klassifikation unterscheidet zwischen Ansichtsklassenmodellen und objektzentrierten Modellen. Objektzentrierte Modelle beschreiben die volle dreidimensionale Struktur von Objekten unabhängig von potentiellen Blickpunkten bezo-

gen auf ein Objektkoordinatensystem. Allerdings vernachlässigen sie das Problem, daß von einem Blickpunkt betrachtete dreidimensionale Objekte in einem zweidimensionalen Bild erkannt werden müssen [Haralick & Shapiro 93b]. In der Technik der Ansichtklassenrepräsentation wird dieses Problem stärker berücksichtigt.

### Ansichtklassenmodelle

In Ansichtklassenmodellen wird die Menge der möglichen Blickpunkte in eine endliche Menge von Ansichtsklassen unterteilt. Jede Ansichtsklasse stellt eine Menge von Blickpunkten mit einer gemeinsamen Eigenschaft dar. Eine solche Eigenschaft ist beispielsweise, daß unter allen Blickpunkten einer Klasse die gleichen Linien im Bild sichtbar sind oder daß sich die Strichzeichnung als Menge aller Linien zwischen Linienverbindungsstellen topologisch nicht ändert. Die Einführung des Konzeptes der Ansichtsklassen und insbesondere des Konzeptes des Aspektgraphen wird [Koenderink & Van Dorn 79] zugeschrieben.

Jeder Knoten eines Aspektgraphen repräsentiert eine Ansicht oder einen Aspekt zum Beispiel in Form einer Strichzeichnung eines dreidimensionalen Objektes. Dabei ist die Ansicht jedes Knotens von der Ansicht der Nachbarknoten topologisch verschieden. Jeder Bogen repräsentiert einen Übergang von einer Klasse von Blickpunkten in eine andere. Ein Blickpunkt wird als stabil oder „allgemein“ (general) bezeichnet, wenn die Ansicht von diesem Blickpunkt und die Ansichten von benachbarten Blickpunkten aus identisch sind. Eine Blickrichtung gilt als „gestört“ (accidental), wenn sie nicht identisch sind. Für die Identität wird in [Charkavaty & Freeman 82] folgendes Beispiel angegeben. Die perspektivische Projektion einer sichtbaren Polyederecke im dreidimensionalen Raum wird dabei als Linienverbindung und die perspektivische Projektion einer sichtbaren dreidimensionalen Kante als Linie bezeichnet. Unter einer Linienverbindungs-Identität versteht man, daß in beiden Ansichten einer Blickpunktklasse die gleichen Linienverbindungen sichtbar sind und Linienverbindungen in gleicher Art und Weise durch Linien verbunden sind, während sich die Längen korrespondierender Linien unterscheiden können. Im Gegensatz zu einer gestörten Blickrichtung haben kleine Änderungen in einem allgemeinen Blickpunkt nur kleine Auswirkung auf die resultierende Ansicht. Übergänge zwischen den Regionen stellen wiederum gestörte Blickrichtungen dar.

Unter der Annahme, daß in Bildern nur ganz bestimmte Objektansichten auftreten, kann man sich auf Ansichten beschränken, die in wenige Ansichtsklassen eingeteilt werden können. In der *Repräsentation charakteristischer Ansichten* wird ein Objekt durch eine Menge repräsentativer Ansichten beschrieben [Charkavaty & Freeman 82]. Eine jede solche Ansicht ist Repräsentant einer Ansichtsklasse. Charakteristische Ansichten sind vor allem dann geeignet, wenn Wissen darüber bereitsteht, daß Objekte nur aus bestimmten Blickrichtungen betrachtet werden. Ist dies der Fall, kann man Objekte durch Vergleich aller charakteristischer Ansichten mit einer Bildbeschreibung erkennen. Kann man die Zahl charakteristischer Ansichten klein halten, ist diese Vorgehensweise relativ effizient.

Unter der Annahme einer orthographischen Projektion kann man die möglichen Blickrichtungen durch die Repräsentation der *Blickrichtungskugel* beschreiben. In dieser Beschreibungsform befindet sich das Objekt unendlich klein im Zentrum der Kugel. Jedem Punkt auf der Kugeloberfläche entspricht eine mögliche Blickrichtung in Richtung des Kugelzentrums. Die Blickrichtungskugel ist so in zusammenhängende Blickpunktclassen unterteilt, daß alle Blickpunkte einer Klasse isomorphe Ansichten erzeugen. Dazu wird die Kugel mosaikartig unterteilt und für den Blickpunkt jeder Facette die entsprechende Objektansicht berechnet. Danach werden Blickpunkte jeder Facette, die isomorphe Ansichten erzeugen, zusammengefaßt. Die

Unterteilung ist zwar einfach durchführbar, jedoch ist es schwierig, den richtigen Grad der Unterteilung zu finden. Wählt man die Unterteilung zu fein, sind die Ansichten in benachbarten Facetten gleich. Wählt man die Unterteilung dagegen zu grob, können für die Erkennung benötigte Ansichten verlorengehen.

Um lokale Modelle einer gegebenen Repräsentation anhand einer extrahierten Bildbeschreibung erkennen zu können, existieren verschiedene Erkennungsverfahren.

### 1.2.3 Erkennungsverfahren

Das Ziel eines Objekterkennungssystems kann wie folgt definiert werden. Gegeben sei eine Menge von Objektmodellen, die die Form oder die Geometrie der zu erkennenden starren Objekte beschreiben. Weiterhin sei eine symbolische Bildbeschreibung gegeben. Mit diesen Eingaben soll ein Objekterkennungssystem im Idealfall eine Menge aller konsistenten Interpretationen der Szenenbereichselemente in Bezug auf die Objektmodelle liefern [Grimson 90].

Eine Interpretation bedeutet dabei, daß sowohl Korrespondenzen zwischen Szenenbereichselementen und Modellelementen, als auch eine Schätzung einer Koordinatensystemtransformation gefunden werden, um ein Objektmodell bezogen auf ein objektzentriertes Koordinatensystem in das Koordinatensystem des Sensors zu transformieren. Unter "konsistent" wird hier verstanden, daß nach Anwenden der Koordinatensystemtransformation auf das Modell ein Modellelement in dem Sensorkoordinatensystem an der Position eines solchen Szenenbereichsstrukturelementes erscheint, für das eine Korrespondenz mit dem Modellelement identifiziert wurde. Die Vorgehensweise verwendet explizit die Zwangsbedingung der Blickpunktconsistenz. Sie besagt, daß die Positionen aller projizierten Modellelemente in das Bild mit der Projektion von einem einzigen Blickpunkt aus konsistent sein müssen. Da für mehr als ein Objektmodell oder für ein einziges Objektmodell in mehreren Lagen mehrere mögliche Interpretation existieren können, sollte im Idealfall die *Menge* aller möglichen Interpretationen ermittelt werden. Modellbasierte Objekterkennungsverfahren lassen sich nach dem Suchraum in drei Ansätze klassifizieren: Suchen im Korrespondenzraum, Suchen im Positions- und Orientierungsraum und Suchen in einer Kombinationen aus beiden Suchräumen.

Der Korrespondenzraum besteht aus allen Zuordnungen von Bildelementen zu Modellelementen. Ein Punkt in diesem diskreten Raum stellt eine Interpretation aller Bildelemente dar. Liegen  $m$  Bildelemente und  $n$  Modellelemente vor, so ist der resultierende Korrespondenzraum  $m$ -dimensional. Das Suchproblem besteht darin, konsistente Interpretationen in dem Suchraum aufzufinden. Da der Korrespondenzraum mit der Anzahl der Modell- und Bildelemente i.a. exponentiell wächst und auch inkonsistente Interpretationen enthält, sind Methoden erforderlich, um Korrespondenzen auszuschließen, ohne sie alle testen zu müssen. In dem Suchprozeß benötigt man dazu Zwangsbedingungen. Zwangsbedingungen können den Suchraum um viele, jedoch nicht um alle unmögliche Interpretationen verkleinern. Da Objekte hier anhand ihrer Form erkannt werden sollen, kommen in erster Linie geometrische Zwangsbedingungen in Frage. Um solche Zwangsbedingungen zu erhalten, muß ermittelt werden, unter welchen Bedingungen Attribute von Bild- und Modellelementen in Übereinstimmung sind. Die restlichen Punkte des Suchraums, die den Zwangsbedingungen genügen, müssen durch Transformation in das Bild überprüft werden. Eine Methode, den Suchraum zu durchsuchen, ist das Suchen in Interpretationsbäumen. Die Methode wird im nächsten Abschnitt näher erläutert.

Der Positions- und Orientierungsraum hingegen besteht aus dem Raum aller Transformationen der Modellelemente in Bildelemente. Im zweidimensionalen Fall

kann der Positions- und Orientierungsraum drei Parameter besitzen, nämlich die  $x$ - und die  $y$ -Koordinate und einen Rotationswinkel. Jeder Punkt in dem Positions- und Orientierungsraum stellt eine Transformation von Modell- in Bildelemente dar. Das Suchproblem besteht darin, Transformationen so zu bestimmen, daß die transformierten Modellelemente zu Bildelementen passen. Eine einfache Methode besteht darin, den Positions- und Orientierungsraum vollständig äquidistant abzutasten und die an jedem Punkt resultierende Transformation zu verwenden, um zu überprüfen, ob die Interpretation von Bildelementen als Instanz des Modells möglich ist und wenn ja, die Interpretation durch ein Gütemaß zu bewerten. Der Punkt des Suchraums mit der höchsten Güte gilt als beste Interpretation. Ein Problem dieses Ansatzes besteht darin, daß der Positions- und Orientierungsraum sehr fein abgetastet werden muß, um eine hohe Lagegenauigkeit zu erzielen.

Eine Kombination aus Korrespondenzraum und Positions- und Orientierungsraum wird in der Technik „Aufstellen und Testen von Hypothesen“ verwendet. Die Idee besteht darin, daß immer nur so viele Korrespondenzen ermittelt werden, daß eine vollständige Transformation eindeutig berechnet werden kann. Zunächst wird eine begrenzte Anzahl an initialen Korrespondenzen zwischen Modell- und Bildelementen gesucht. Diese initialen Korrespondenzen dienen dazu, die Hypothese über eine Transformationsmatrix aufzustellen, die die Position und die Orientierung der Modellinstanz in Bezug auf den Sensor angibt. Danach wird die Hypothese überprüft, indem eine Modellausprägung gemäß der Transformation in das Bild projiziert wird. Die Übereinstimmung der Modellausprägung mit dem Bild wird anhand einer Gütefunktion bewertet. Das heißt, daß die initialen Korrespondenzen bereits einen bestimmten Blickpunkt festlegen, so daß andere Bildelemente unter den gleichen Lageparametern zu dem Objektmodell passen müssen. Sowohl Erkennungsverfahren, die Hypothesen aufstellen und testen, als auch Erkennungsverfahren, die mögliche Interpretationen im Korrespondenzraum suchen, können eine Baumsuche in Interpretationsbäumen durchführen.

### Baumsuche in Interpretationsbäumen

Eine geeignete Repräsentation für die Suche im Korrespondenzraum ist der Interpretationsbaum. Außer in der Baumwurzel steht in jedem Knoten eine Korrespondenz zwischen einem Bildelement und einem Modellelement. Knoten auf der gleichen Baumtiefe repräsentieren Korrespondenzen von einem Bildelement zu allen Modellelementen. Existieren  $m$  Modellelemente, so besitzt jeder Knoten  $m$  Nachfolger. Ein Knoten auf der Tiefe  $n$  beschreibt eine unvollständige Interpretation, in der die Knoten auf dem Pfad zu der Wurzel eine Zuordnung von Modellelementen zu den ersten  $n$  Bildelementen darstellen. Jedes Blatt des Baumes stellt also eine Interpretation aller Bildelemente dar.

Eine triviale Methode besteht darin, alle Blätter des Interpretationsbaumes darauf zu überprüfen, ob die entsprechende Interpretation konsistent ist. Dies ist allerdings zu aufwendig, da der Korrespondenzraum vollständig durchsucht werden müsste und die Größe des Korrespondenzraums exponentiell mit der Anzahl der Modell- und Bildelemente steigt. Ist der Korrespondenzraum jedoch durch Zwangsbedingungen eingeschränkt, fallen ganze Teilbäume des Interpretationsbaumes weg. Eine Zwangsbedingung kann nicht die Konsistenz einer Interpretation garantieren, sondern nur inkonsistente Interpretationen ausschließen.

Durch Zwangsbedingungen verkleinerte Bäume können durch die bekannten Baumsuchverfahren wie Tiefensuche oder Breitensuche durchsucht werden. Andere Baumsuchverfahren sind in [Grimson 90] aufgeführt. Baumsuchverfahren können um Heuristiken ergänzt werden.

Mit Heuristiken kann man festlegen in welcher Reihenfolge Bildelemente verglichen werden. Zum Beispiel können Elemente in Bild und Modell mit großer Ausdehnung oder mit geringem Abstand zuerst verglichen werden. Eine andere Heuristik bricht einen Suchvorgang vorzeitig ab, wenn eine geeignete Kostenfunktion, die eine Interpretation bewertet, einen bestimmten Schwellwert unterschreitet.

Kostenfunktionen können aber auch den Suchprozeß selbst steuern, indem einzelne Baumknoten bewertet werden. Eine Kostenfunktion dient dabei als Maß für die Wahrscheinlichkeit, daß der zu bewertende Knoten auf einem Pfad liegt, der zu einer Lösung führt. Um eine Interpretation mit minimalen Kosten zu suchen, kann man statt einer Baumsuche in einem Interpretationsbaum auch Optimierungsverfahren anwenden, die eine optimale Belegung von Korrespondenzvariablen suchen. Solche Verfahren arbeiten nach dem Schema der iterativen Verbesserung anhand einer Kostenfunktion.

### Iterative Verbesserung anhand einer Kostenfunktion

Der Begriff der iterativen Verbesserung (engl. iterative improvement) wurde in Anlehnung an [Kirkpatrick *et al.* 83] gewählt. Während bei der Baumsuche in Interpretationsbäumen jede Korrespondenz in einem Knoten kodiert wird, wird in Verfahren der iterativen Verbesserung eine Korrespondenz durch eine Optimierungs- oder Korrespondenzvariable repräsentiert. Die Variable kann binär oder kontinuierlich sein. Die binäre Korrespondenzvariable kodiert nur, ob eine oder ob keine Zuordnung von einem Bild- zu einem Modellelement besteht. Eine kontinuierliche Korrespondenzvariable gibt an, wie stark eine Korrespondenz ist. Existieren  $m$  Bildelemente und  $n$  Modellelemente, lassen sich die Korrespondenzvariablen zu einer  $m \times n$ -Matrix anordnen, die als Korrespondenzmatrix bezeichnet wird. Mit jeder Belegung der Matrix ist eine Interpretation der Bildelemente und damit ein Punkt in dem Suchraum gegeben. Der Suchraum besitzt eine Dimension pro Korrespondenzvariable und ist daher  $n \cdot m$ -dimensional.

Da nicht alle Matrixbelegungen erlaubte Korrespondenzen liefern, werden Konsistenzbedingungen an die Korrespondenzmatrix gestellt. Zum Beispiel kann eine Konsistenzbedingung ausdrücken, daß die Korrespondenz eines Bildelementes mit zwei oder mehreren Modellelementen nicht erlaubt ist. Geometrische Zwangsbedingungen werden so in einer Kostenfunktion kodiert, daß eine Korrespondenzmatrixbelegung, die global minimale Kosten besitzt, eine maximale Ähnlichkeit zwischen Attributen korrespondierender Modell- und Bildelemente darstellt. Damit liegt ein Optimierungsproblem vor, in dem eine Matrixbelegung zu suchen ist, die unter Einhaltung der Konsistenzbedingungen möglichst minimale Kosten aufweisen soll. Da die Kostenfunktion Ähnlichkeiten von relationalen Strukturen beschreibt, wird für eine solche Kostenfunktion in der Literatur auch der Begriff des relationalen Abstandes verwendet [Haralick & Shapiro 93b].

Die Korrespondenzmatrix wird initialisiert und für die Matrixelemente werden iterativ neue Werte gemäß der Kostenfunktion berechnet, bis eine Terminierungsbedingung erfüllt ist. Die Verbesserungsschritte der Optimierungsvariablen können alle gleichzeitig erfolgen. Das Verfahren ist vor allem dann geeignet, wenn genau eine wahrscheinliche Interpretation der Bildelemente gesucht ist und nicht alle möglichen Interpretationen gesucht sind. Als Konsistenztests können zwar keine globalen Tests im Sinne eines Modelltests eingesetzt werden, jedoch können die in der Kostenfunktion kodierten Zwangsbedingungen den Suchraum so einschränken, daß konsistente Interpretationen mit hoher Wahrscheinlichkeit gefunden werden.

Besonders unter dem Gesichtspunkt der Parallelisierung sind Erkennungsverfahren der iterativen Verbesserung interessant.

### 1.2.4 Parallelisierbarkeit von Erkennungsverfahren

Die Anforderungen an Rechenzeit und die Einhaltung von Echtzeitbedingungen in Verbindung mit Anwendungen im Maschinensehen haben zu vielen Anstrengungen geführt, parallele Verfahren zu untersuchen [Reeves 91]. Die Arbeitsweise von Objekterkennungssystemen kann meist auf verschiedenen Ebenen beschrieben werden. Auf der unteren Ebene werden aus Bilddaten Merkmale zum Beispiel Kanten detektiert. Auf hoher Ebene werden daraus abstrakte Beschreibungen von Objektteilen oder Objekten generiert, die zur Erkennung und Lagebestimmung von Objekten dienen. Auf jeder Ebene besteht ein hohes Potential an Parallelismus [Weems *et al.* 91]. In dieser Arbeit ist nur die Parallelisierbarkeit von Verfahren zur Objekterkennung auf der hohen Ebene von Interesse.

Nach [Waldschmidt 95] werden die in einer Anwendung vorhandenen datenumabhängigen Operationen als *Parallelität der Anwendung* bezeichnet. Die Verarbeitungseinheiten, die die Berechnung eines Prozesses ausführen können, werden auch als *Prozessorelemente* oder *Prozessoren* bezeichnet. Die Prozessoren sind über ein *Verbindungsnetzwerk* zusammengeschaltet. Der Datenaustausch zwischen Prozessoren über das Verbindungsnetzwerk wird als *Kommunikation* bezeichnet. Die *Synchronisation* ist eine Operation, bei der zwei oder mehr Prozessoren kommunizieren, um die zeitliche Folge von Ereignissen festzulegen.

Die Parallelisierbarkeit von Erkennungsverfahren wird hier grob anhand der Klassifikation nach synchroner Parallelität und asynchroner Parallelität bewertet. Die Klassifikation und Bewertung stützt sich auf das Buch von Bräunl [Bräunl 93].

- Bei der klassischen asynchronen Parallelität wird die zu lösende Aufgabe in Teilaufgaben zerlegt, die in Form von Prozessen auf eine Gruppe von selbständigen, unabhängigen Prozessoren verteilt werden, d.h. ein asynchron arbeitendes Programm besteht aus mehreren Kontrollflüssen. Sind die Teilaufgaben nicht völlig unabhängig voneinander, so müssen die Prozesse untereinander Daten austauschen und sich dazu gegenseitig synchronisieren. Die Prozesse führen meist größere Teilaufgaben aus, denn das Aufteilen in kleinere Aufgaben, wie etwa das Auswerten arithmetischer Ausdrücke, würde im Vergleich zum Parallelitätsgewinn zu hohe Synchronisationskosten erfordern.
- Bei der synchronen Parallelität können die für eine Aufgabe eingesetzten Prozessoren von einem zentralen Programm mit gleichem Takt gesteuert werden und sind nicht mehr unabhängig voneinander. Ein synchron paralleles Programm besitzt nur einen einzigen Kontrollfluß. Dieses vereinfachte Berechnungsmodell stellt zwar eine Einschränkung dar, jedoch können die Prozessorelemente wegen ihres einfachen Aufbaus höher integriert werden. Dadurch können Rechner mit erheblich mehr Prozessoren gebaut werden, als dies bei asynchroner Parallelität möglich ist. Die Synchronisation zwischen Prozessoren erfolgt implizit bei jedem Schritt und ist nicht mehr Aufgabe des Programmierers. Die Prozessoren arbeiten an kleinen Verarbeitungseinheiten wie der Auswertung von Vektorausdrücken. Wenn alle Prozessoren im Gleichtakt rechnen, also in jedem Schritt synchronisiert werden, entfällt die Notwendigkeit für aufwendige und fehleranfällige Synchronisationsmechanismen. Es kann ein kollektiver Datenaustausch zwischen allen Prozessoren oder innerhalb einer Gruppe von Prozessoren stattfinden.

Mit den Arten der Parallelität lassen sich die Verfahren zur iterativen Verbesserung und zur Suche in Interpretationsbäumen hinsichtlich der Parallelisierbarkeit wie folgt bewerten:

- Die Parallelisierung von Verfahren zur Suche in Interpretationsbäumen besteht im Wesentlichen darin, bekannte Baumsuchverfahren zu parallelisieren. Dazu wird beispielsweise eine Tiefensuche in Teilprozesse für einzelne Teilbäume zerlegt, die unabhängig und parallel abgearbeitet werden können. Durch ungeschicktes Verteilen von Teilprozessen auf Prozessoren können jedoch erhebliche Effizienzverluste auftreten. Im Extremfall kann die Leistung eines Parallelrechners auf die eines einzigen Prozessors sinken, obwohl eine weitere Parallelverarbeitung möglich wäre. Um solche Ineffizienzen zu vermeiden, wurden verschiedene Methoden zur Lastbalancierung entwickelt, die einen großen Problemkreis der asynchronen Programmierung darstellen. Verfahren zur Lastbalancierung zeichnen sich durch einen hohen allgemeinen Verwaltungsaufwand aus und setzen im allgemeinen zu spät ein, nämlich erst dann, wenn das Lastgleichgewicht bereits erheblich gestört ist. Weiterhin ist bei voller paralleler Systemlast jede Lastbalancierung zwecklos, und der allgemeine Verwaltungsaufwand für die Lastbalancierung verursacht dann einen unnötigen Mehraufwand an Rechenzeit.
- Optimierungsverfahren der iterativen Verbesserung anhand einer Kostenfunktion können als synchron-parallele Algorithmen formuliert werden. Dabei entspricht jede Optimierungsvariable einem Prozessor. Alle Optimierungsvariablen können im gleichen Takt auf einen neuen Wert angepaßt oder verbessert werden. Es existiert daher nur ein Kontrollfluß, die Synchronisation der Prozessoren ist implizit. Jeder Prozessor benötigt nur einfache arithmetische Operationen, um einen neuen Wert der Optimierungsvariable zu bestimmen, mit dem die Kostenfunktion mit hoher Wahrscheinlichkeit fällt. Systembedingt kann und muß keine Lastbalancierung vorgenommen werden.
- Modelle künstlicher neuronaler Netze können zu Verfahren der iterativen Verbesserung gehören, wenn alle Neuronen zum gleichen Zeitpunkt schalten. Im Modell künstlicher neuronaler Netze werden Neuronen zu informationsverarbeitenden Einheiten abstrahiert, die aus einem Prozessorelement mit lokalem Speicher bestehen. Das künstliche neuronale Netz besteht dann aus einer Vielzahl von sehr einfachen, aber stets aktiven Elementen, die parallel identische Verarbeitungsroutinen durchführen.

Unter dem Aspekt der Parallelisierbarkeit werden hier keine Baumsuchverfahren in Interpretationsbäumen, sondern Verfahren der iterativen Verbesserung anhand einer Kostenfunktion als Erkennungsverfahren untersucht.

## 1.3 Problemstellung

Das Problem der Objekterkennung kann in vier Teilprobleme strukturiert werden.

- Zunächst sollen aus einem Abstandsbild kantenbasierte Szenenbereichshinweise extrahiert werden, die auf der Technik der Unstetigkeitsmarkierung beruhen. Dabei müssen typische Meßfehler von 3D-Laserradarscannern beachtet werden.
- Für die zu erkennenden Objekte soll eine geeignete Repräsentation gewählt werden, die unter Verwendung geometrischer Zwangsbedingungen einen Vergleich mit der symbolischen Bildbeschreibung erlaubt.
- Das Korrespondenzproblem soll unter Verwendung von Konsistenzbedingungen als Optimierungsproblem formuliert werden, so daß auf das Problem Verfahren der iterativen Verbesserung anwendbar sind.

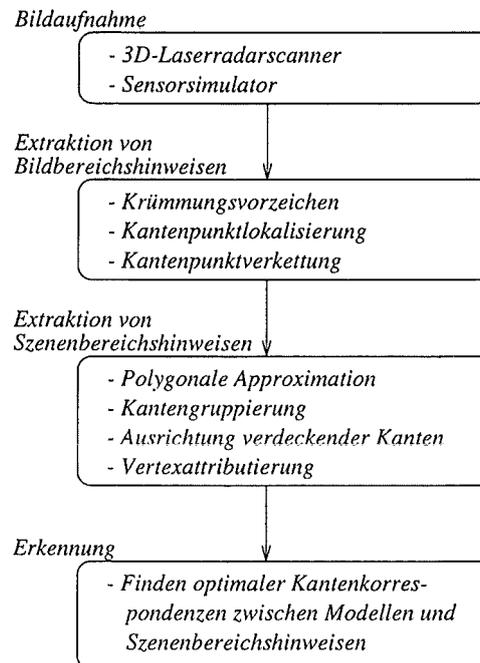


Abbildung 1.2: Verarbeitungsschritte des Objekterkennungsansatzes

- Es soll schließlich untersucht werden, ob die geometrischen Zwangsbedingungen den Suchraum so einschränken, daß Verfahren der iterativen Verbesserung zufriedenstellende Lösungen des Korrespondenzproblems liefern.

## 1.4 Beschreibung des Ansatzes

Mit dem Wissen über das Einsatzgebiet und die Einsatzumgebung soll der hier vorgeschlagene Ansatz anhand der Terminologie von [Kanade 78] beschrieben werden. Abbildung 1.1 veranschaulicht die hier beschriebenen Teilprozesse in Form durchgezogener Pfeile. Die einzelnen Verarbeitungsschritte sind in Abb. 1.2 dargestellt.

Über eine Szene ist bekannt, daß die Objektkonstellationen aus trihedralen, starren und statischen Polyedern bestehen. Es wird vorausgesetzt, daß jedes zu erkennende Objekt in einer Szene nicht mehrmals, sondern maximal einmal auftritt. Damit ist eine Orientierung oder eine externe Positionsbestimmung anhand zu erkennender Objekte möglich. Von einer derart strukturierten Szene möge zum Zwecke der externen Positionsbestimmung ein einzelnes dichtbesetztes digitalisiertes Abstandsbild vorliegen.

Da Abstandsbilder von polyedrischen Szenen aufgenommen sind, werden Szenenbereichshinweise, die auf Polyederkanten hindeuten, durch die kantenbasierte Technik der Unstetigkeitsmarkierung repräsentiert. Die Unstetigkeitsmarkierung kombiniert die Beschreibung dreidimensionaler Kanten mit Merkmalen ihrer zweidimensionalen Projektion.

In den Knoten eines attributierten Graphen werden dreidimensionale Kanten, in den Bögen die Nachbarschaft von Kanten repräsentiert. Als Knotenattribute

dienen die Kantenlänge und physikalische Kanteneigenschaften. Als Bogenattribute dienen sog. Linienverbindungen, die das Zusammentreffen von Kanten verschiedenen Typs in einem gemeinsamen Endpunkt anhand ihrer perspektivischen Projektion charakterisieren. Linienverbindungen können in wenige Klassen eingeteilt werden und dienen in Form von Indizes als Bogenattribute.

Unter der Annahme, daß Objekte nur aus eingeschränkten Blickrichtungen betrachtet werden, wird jedes Objekt durch eine geringe Anzahl charakteristischer Ansichten beschrieben. Jede Ansicht wiederum wird als attributierter Graph repräsentiert. Die Extraktion dreidimensionaler Kanten erfolgt in zwei Schritten, nach Bildbereichshinweisen und nach Szenenbereichshinweisen.

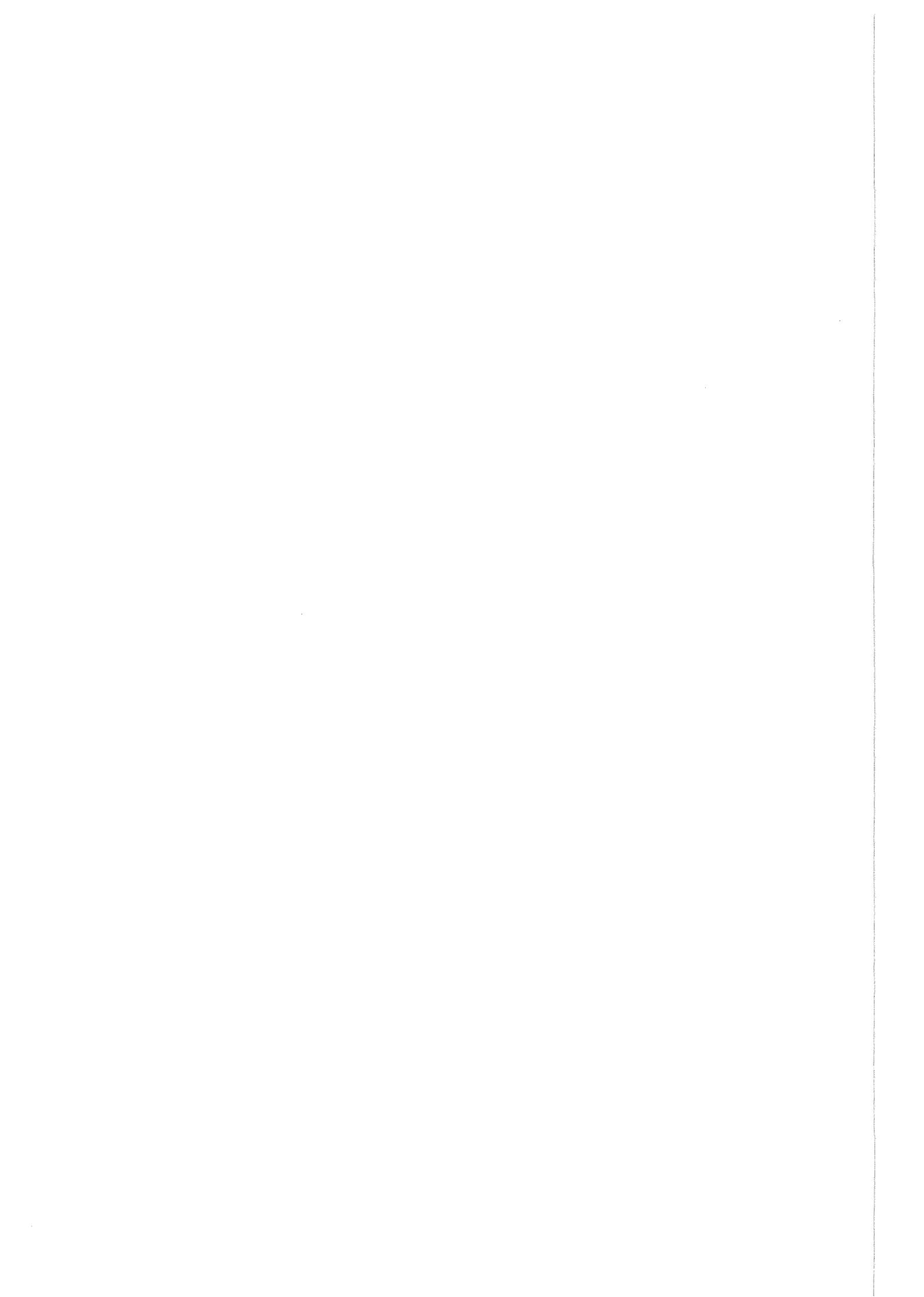
Um Hinweise auf Bildkantenpunkte bestimmten Typs zu erhalten wird ein spezielles Krümmungsmaß verwendet. Kanten mit Abstandsunstetigkeiten werden dabei als benachbarte Bildpunkte mit entgegengesetztem Krümmungsvorzeichen detektiert. Diese Nachbarschaft wird verwendet, um Kantenpunkte trotz eines für das Radarverfahren typischen Sensoreffekts möglichst exakt zu lokalisieren. Die Bildpunkte jeden Typs werden danach zu 2D-Bildpunktketten zusammengefaßt.

Unter Erhaltung ihres Typs werden die 2D-Punktketten in 3D-Punktketten transformiert. Dies ist möglich, da die Transformation von Bild- in Szenekoordinaten in Form des Abtast- und Meßverfahrens des 3D-Laserradars explizit bekannt ist. Die Transformation bildet den Übergang von Hinweisen aus dem Bildbereich zu Hinweisen im Szenenbereich. Durch polygonale Approximation der 3D-Punktketten erhält man 3D-Kanten. Daraufhin werden 3D-Kanten nach Gruppierungskriterien gemäß des Abstands von 3D-Kantenendpunkten und gemäß der Richtung von 3D-Kanten zu einer Struktur benachbarter Kanten geeignet gruppiert. Damit liegt eine Szenenkantenstruktur vor, die aus 3D-Kanten und räumlichen Beziehungen besteht. Nachfolgend werden Attribute berechnet, die von der Ansicht einer Szene, also von Blickpunkt und -richtung abhängig sind. Dazu werden bestimmte Kanten mit einer Orientierung versehen, die eine Aussage über die Lage angrenzender Flächen macht.

Um Korrespondenzen zwischen Knoten des attributierten Graphen der Szene und der attributierten Graphen charakteristischer Objektansichten zu bewerten, wird systematisch eine Kostenfunktion konstruiert. Mit Hilfe der Kostenfunktion wird ein Optimierungsproblem formuliert. Zur Lösung des Problems werden künstliche neuronale Netze nach dem Brain-state-in-the-box Modell verwendet. Anhand eines Benchmarks werden die verwendeten neuronalen Netze mit dem Ansatz des Mean Field Annealing verglichen.

## 1.5 Kapitelübersicht

Im nächsten Kapitel wird die Funktionsweise von 3D-Laserradarscannern, die einhergehenden Meßfehler und das Wissen über die betrachteten Szenen erläutert. Im dritten Kapitel wird beschrieben, wie geeignete Krümmungen berechnet, Kantenpunkte verschiedenen Typs trotz bestimmter Meßfehler lokalisiert und danach Kantenpunktketten generiert werden. Wie aus Punktketten eine attributierte Kantenstruktur erzeugt wird, beschreibt Kapitel vier. Im fünften Kapitel wird zunächst das Korrespondenzproblem definiert. Dann werden Verfahren zur Lösung des Optimierungsproblems in einem einheitlichen Rahmen gegenübergestellt sowie die Ergebnisse eines Verfahrens und ein Vergleich mit einem anderen Verfahren anhand eines Benchmarks vorgestellt. Das letzte Kapitel schließlich faßt die Ergebnisse zusammen und gibt einen kurzen Ausblick.



## Kapitel 2

# Bildaufnahme und Szenenwissen

### 2.1 Bildgebende Laserscanner

Laserscanner gehören zu der Klasse bildgebender Abstandssensoren. Solche Sensoren können aktiv oder passiv sein. Ein aktiver Sensor strahlt Energie in die Umgebung ab, der passive Sensor nimmt Energie wie zum Beispiel Licht aus der Umgebung auf [Besl 88a]. Aktive Sensoren hingegen strahlen beispielsweise Schallwellen wie Ultraschall oder elektromagnetische Wellen wie Laserlicht ab. Methoden zur Abstandsbestimmung können weiterhin nach bildbasierten und direkten Verfahren unterteilt werden [Jarvis 93]. Bildbasierte Methoden versuchen, Abstandsbilder durch Analyse von Videobildern zu bestimmen. Direkte Methoden hingegen bestimmen den Abstand ohne eine nachgeschaltete Bildanalyse. Eine gute Übersicht über bildbasierte Methoden bietet Jarvis [Jarvis 83, Jarvis93]. Speziell unter den Laserlicht abstrahlenden bildgebenden Abstandssensoren lassen sich Triangulationslaserscanner und Laserradarscanner unterscheiden.

#### 2.1.1 Triangulationslaserscanner

Ein spezielles Verfahren von Triangulationsscannern projiziert eine Lichtebene in die Szene. Das reflektierte Licht wird von einer Videokamera aufgenommen, die einen bekannten Basisabstand zur Laserquelle und eine bekannte Ausrichtung besitzt. In dem Kamerabild erscheint ein möglicherweise unterbrochener Lichtstreifen, der um einen bestimmten Betrag proportional zu dem zu messenden Abstand versetzt ist. Durch einfache Triangulationsrechnung kann zu einem Bildpunkt des Lichtstreifens der entsprechende Abstand zu der Oberfläche der Szene berechnet werden. Da zur Abstandsberechnung ein Videobild ausgewertet werden muß, liegt eine bildbasierte Methode vor. Eine wichtige Eigenschaft des Triangulationslaserscanners als bildbasierte Methode besteht darin, daß das auszuwertende Videobild eine perspektivische Projektion der Szene darstellt. Die perspektivische Projektion wiederum bewirkt, daß gerade Kanten in der Szene auf gerade Linien in dem Abstandsbild abgebildet werden.

Ein Nachteil des Verfahrens ist der Abschattungseffekt. Er besteht darin, daß Szenenpunkte, die ausgeleuchtet werden von der Kamera nicht sichtbar sind, und umgekehrt, daß Szenenpunkte, die von der Kamera sichtbar sind, nicht ausgeleuchtet werden können. Für solche Punkte kann daher kein Abstand gemessen werden.

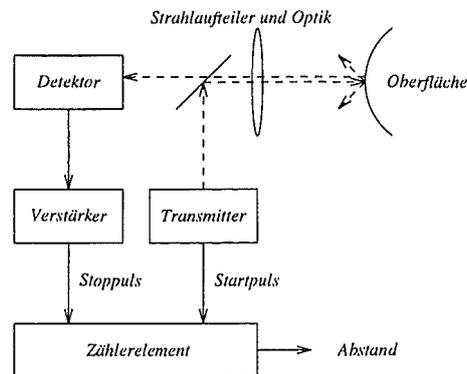


Abbildung 2.1: Abstandsmessung bei dem Radarverfahren: Der Transmitter startet das Zählerelement. Das von der Objektfläche reflektierte Licht wird von einem Detektor empfangen, verstärkt und stoppt schließlich das Zählerelement. Aus dem Zählerstand kann dann ein Abstand ermittelt werden. (Gestrichelte Linien repräsentieren Laserlicht, durchgezogene Linien elektrische Signale.)

### 2.1.2 Laserradarscanner

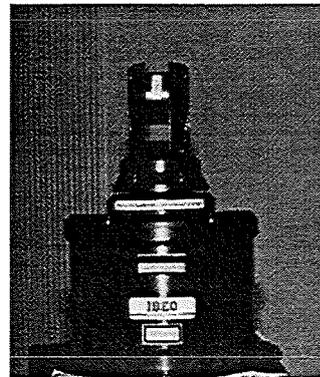
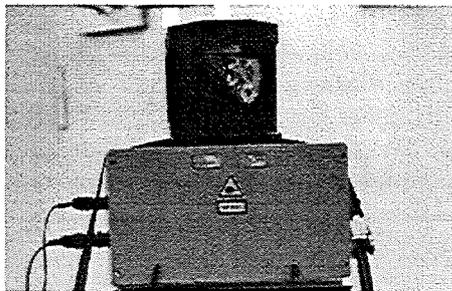
Beim allgemeinen Radarprinzip wird die Signallaufzeit zwischen Aussende-, Reflektions- und Detektionsvorgang gemessen. Beim Laserradar wird der Abstand zu Oberflächen der Umgebung aus der Laufzeit des Laserlichts berechnet. Da im Gegensatz zu Triangulationslaserscannern Laserquelle und Detektor fast koaxial angeordnet sind, können zu nahezu allen Szenenpunkten, die ausgeleuchtet werden, auch Abstände gemessen werden. Abb. 2.1 verdeutlicht die Funktionsweise des Laserradarverfahrens.

Die Richtungen, entlang derer Abstände gemessen werden, steuert eine Abtasteinheit, auch Scanner genannt. Je nach Art und Weise der Abtastung lassen sich die Laserstrahlrichtungen mathematisch durch Koordinatensysteme beschreiben. Die hier verwendeten Laserradarscanner steuern die Laserstrahlrichtung in Kugelkoordinaten, wobei sie sich in der Realisierung der Abtasteinheit unterscheiden (siehe Abb. 2.2).

Indem zeilenweise von oben nach unten Abtast- und Meßvorgänge durchgeführt werden, entsteht sequentiell ein Abstandsbild. Die Genauigkeit einer bzgl. der Winkel äquidistanten Abtastung wird durch ein horizontales und vertikales Winkelinkrement charakterisiert. Je kleiner dieses Winkelinkrement, desto höher ist die Abtastauflösung. Die Daten der eingesetzten Laserradarscanner werden in Tabelle 2.3 zusammengefaßt.

## 2.2 Meßfehler von 3D-Laserradarscannern

Der Bildaufnahmeprozess bestehend aus Abtastung und Abstandsmessung ist durch eine Vielzahl von Störgrößen bestimmt [Besl 88a]. Die folgenden Sensoreffekte speziell von 3D-Laserradarscannern werden in [Hebert & Krotkov 92] und [Koprek 92] diskutiert. Man kann zwei Arten von Meßfehlern unterscheiden: Meßfehler, die in jedem Abtastpunkt auftreten, und Meßfehler, die nur in bestimmten Abtastpunkten wirksam sind.



(a) Laserscanner 1 mit Polygonspiegel (b) Laserscanner 2 mit Schwenk-Neige-Kopf

Abbildung 2.2: Verwendete Laserradarscanner mit verschiedener Abtasteinheit: beide Laserradarscanner tasten in einem Kugelkoordinatensystem ab. Laserscanner 1 lenkt den Laserstrahl über einen zweifach axial-gelagerten, rotierenden Polygonspiegel ab [Orlov 93], [Ibeo 90a], [Ibeo 90b]. Im Gegensatz dazu sind im Laserscanner 2 Laserquelle und Detektor in einem Schwenk-Neige-Kopf integriert, so daß auch Aufnahmen mit einem 360° Rundumblick möglich sind.

Meßbereichstiefe	6 m
Abstandsgenauigkeit	max. $\pm 1$ cm
minimaler Abstand	0.5 m
vertikale Blickfeldbegrenzung	0° – 135°
horizontale Blickfeldbegrenzung	Scanner 1: 0° – 180° Scanner 2: 0° – 360°
minimales vertikales Winkelinkrement	0.1°
minimales horizontales Winkelinkrement	< 0.1°
Meßwertzugriff	punkt-, zeilen und bildweise
horizontale Geschwindigkeit des Sensorkopfes	0.1-7Hz
Auflösung	programmierbar
Abstandskodierung eines Meßwertes	2 Byte (in mm)
Abtastreihenfolge	von rechts nach links, von oben nach unten
Scanner	Scanner 1: Polygonspiegel Scanner 2: Schwenk-Neige-Kopf

Abbildung 2.3: Technische Daten der verwendeten Laserradarscanner

### 2.2.1 Meßfehler in jedem Abtastpunkt

Meßfehler, die in jedem Abtastpunkt auftreten, sind nur von den Eigenschaften des Sensors abhängig. Die Abstandsmessung ist von der Größe des Laserstrahl-Fußabdrucks und der Wahl des Triggerzeitpunktes abhängig, während die Abtastung durch eine nicht-exakte Synchronisation verfälscht wird. Die Größe des Laserstrahlfußabdrucks wirkt sich auf die Abstandsmessung wie folgt aus. Der Lichtstrahl eines Lasers ist kein exakter Zylinder, sondern vielmehr ein Kegel mit einem spitzen Öffnungswinkel. Der Entfernungswert ist daher der Mittelwert aller Werte des zweidimensionalen Oberflächenabschnittes, dem sog. Fußabdruck, der durch den Strahlkegel ausgeleuchtet wird. Da die Amplitude des empfangenen Impulses umgekehrt proportional abnimmt, jedoch gleichzeitig das Rauschen proportional mit dem Quadrat des Abstandes zunimmt, sind vor allem Meßwerte, die in der Nähe des Meßbereichsendes liegen, mit extremer Unsicherheit behaftet. In diesem Bereich erreicht das Rauschen daher die Größenordnung des Signales selbst. Der Meßbereich zuverlässiger Abstandswerte ist daher durch eine maximale Reichweite beschränkt.

Die Wahl des Triggerzeitpunktes beeinflusst ebenfalls die Genauigkeit, mit der ein Abstand gemessen werden kann. Diese hängt bei dem Radarverfahren davon ab, wie genau der Eintreffzeitpunkt des Empfangssignals bestimmt werden kann. Der Eintreffzeitpunkt wird bei dem verwendeten Sensor mit Hilfe des Schwerpunktes des Empfangssignals ermittelt. Allerdings kann der Schwerpunkt mittels Integration erst nachträglich aus dem Gesamtsignal ermittelt werden. Da das Empfangssignal in der Regel verrauscht ist, wird auch der Schwerpunkt verschoben und damit die Messung verfälscht.

Ein weiterer Meßfehler, der jede Messung beeinflusst, besteht in einer nicht-exakten Synchronisation. Idealerweise sollen Abstände nur entlang von Strahlrichtungen gemessen werden, die Vielfachen von konstanten Winkelinkrementen entsprechen. Bei den meisten Laserscannern ist die exakte Winkelposition nicht bekannt. Diese wird vielmehr indirekt aus der Winkelgeschwindigkeit bzgl. der beiden Spiegelachsen und einer Abtastfrequenz berechnet. Da Abtastung und Winkelbewegung nicht exakt synchronisiert sind, werden Abstände in einer leicht versetzten Richtung gemessen.

Stimmt die berechnete nicht mit der tatsächlichen Strahlrichtung überein, können sich gemessene Abstände von tatsächlichen Abständen gerade in der Nähe von Sprungkanten beträchtlich unterscheiden. Infolgedessen erscheinen Sprungkanten im Abstandsbild zerfranst.

In [Koprek 92] werden weitere Rauschquellen erwähnt wie z.B. statistische Rauschquellen. Sie zeichnen sich dadurch aus, daß sich der resultierende Fehler durch Mittelung gemäß einer angenommenen Wahrscheinlichkeitsverteilung aus mehreren Abstandsmessungen reduzieren läßt. Eine solche Rauschquelle liegt beispielsweise in statistischen Schwankungen der Laserimpulsform.

### 2.2.2 Meßfehler in bestimmten Abtastpunkten

Meßfehler, die nur in bestimmten Abtastpunkten auftreten, hängen vor allem von Eigenschaften der Szene ab. Man kann die Meßfehler nach der Abhängigkeit von der Geometrie oder von den Oberflächeneigenschaften der Szene unterscheiden.

Die Oberflächen der Umgebung müssen bzgl. Reflektion, Absorption und Transmission bestimmte Eigenschaften aufweisen. Die Oberflächen dürfen nicht absorbieren, da in diesem Fall kein Licht reflektiert wird und kein Abstand gemessen werden kann. Schwierigkeiten treten z.B. bei schwarzen und matten Oberflächen auf. Die

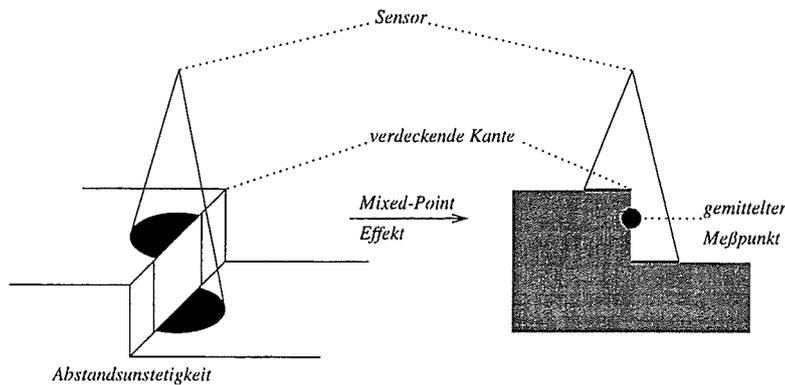


Abbildung 2.4: Entstehung des Mixed-Point Effektes: Da ein Laserstrahl eine bestimmte Strahlaufweitung besitzt, werden beim Auftreffen des Strahls auf eine Abstandsunstetigkeit zwei versetzt liegende Teilflächen ausgeleuchtet. Der gemessene Abstand liegt zwischen der Teilfläche im Vordergrund und der Teilfläche im Hintergrund. Die genaue Detektion von verdeckenden Kanten wird dadurch erschwert.

Oberflächen dürfen auch nicht durchsichtig sein, da der gemessene Abstand nicht dem Abstand zu der durchsichtigen Fläche entspricht. Die Oberflächen dürfen weiterhin nicht spiegeln, sondern müssen diffus streuen, damit der Auftreffpunkt aus der Richtung des Sensors detektiert werden kann. Aufgrund verschiedener Oberflächeneigenschaften können weiterhin unterschiedliche Abstände zu Oberflächen gemessen werden, obwohl diese räumlich den gleichen Abstand besitzen. Beispielsweise können an Übergängen von Flächen verschiedener Oberflächeneigenschaften im Abstandsbild leichte Kanten auftreten, obwohl physikalisch keine Kante vorhanden ist. Ein Meßfehler, der auch bei einheitlichen Oberflächeneigenschaften auftritt, ist der Mixed-Point Effekt.

### Mixed-Point Effekt

Der Mixed-Point Effekt ist in erster Linie von der Geometrie der Szene abhängig. Er wird stets wirksam, wenn der Fußabdruck auf eine Oberfläche mit hohen Abstandsunstetigkeiten trifft. Dadurch wird der Fußabdruck in räumlich versetzt liegende Teilflächen aufgespaltet, die folglich verschiedene Laufzeiten besitzen. Da der Sensor kurz aufeinanderfolgende Signale mittelt, liegt das Meßergebnis dann irgendwo zwischen den Abstandswerten der einzelnen Teilflächen (siehe Abb. 2.4). Der gemessene Abstand entlang einer Richtung ist der Durchschnitt der Abstandswerte über der zweidimensionalen Fläche des Fußabdrucks.

Das Auftreten des Mixed-Point Effektes ist daher stark abhängig von der Größe des Fußabdrucks (siehe Abb. 2.5). Der Flächeninhalt ist proportional zu dem Quadrat des Abstandes zum Zentrum des Fußabdrucks. Der Flächeninhalt des Fußabdrucks hängt auch von dem Winkel  $\gamma$  zwischen der Oberflächennormale und dem Strahl ab. Die Größe des Fußabdrucks ist näherungsweise proportional zu dem Winkel  $\gamma$ , wenn man annimmt daß der Fußabdruck klein genug ist und  $\delta$  nahezu konstant ist bzw. nur innerhalb eines kleinen Bereiches schwankt. Die folgende Tabelle verdeutlicht die Abhängigkeit des Fußabdruckdurchmessers von dem Abstand für den Fall, daß der Laserstrahlkegel senkrecht ( $\gamma = 0$ ) auf eine Ebene auftritt und der Öffnungswinkel  $\delta = 0.138^\circ$  beträgt. Der Durchmesser wird nach der Formel:  $R/D = \tan(\delta)$  berechnet:

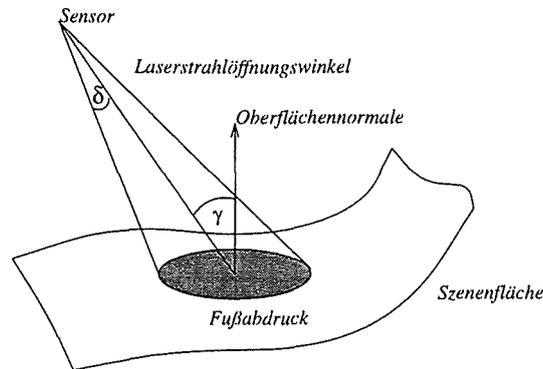
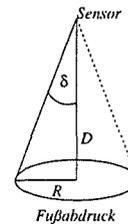


Abbildung 2.5: Größe des Laserstrahlfußabdrucks: die Größe des Fußabdrucks ist abhängig von der Strahlaufweitung  $2\delta$  und dem Eintreffwinkel  $\gamma$

Abstand $D$	Fußabdruckdurchmesser $2R$
1 m	2.41 mm
3 m	7.23 mm
5 m	12.04 mm
10 m	24.09 mm



Die genaue Detektion verdeckender Kanten in diesen Meßpunkten wird also erschwert [Kluge *et al.* 90, Hebert & Krotkov 92]. Da der Mixed-Point Effekt auch bei einheitlichen Oberflächeneigenschaften auftritt, muß er bei der Kantendetektion auf jeden Fall gesondert behandelt werden.

### 2.3 Sensormodell

Zur eindeutigen Begriffsbestimmung wird die Abtastung und die Abstandsmessung des 3D-Laserradars in einem Sensormodell beschrieben. Ohne Beschränkung der Allgemeinheit liege im Blickpunkt des Sensors ein kartesisches, rechtshändiges Koordinatensystem. Die Bildaufnahme kann im vorliegenden Fall mit Hilfe von Kugelkoordinaten  $(r, \phi_u, \theta_v)$  bezogen auf dieses Koordinatensystem beschrieben werden. Das Resultat des Bildaufnahmevorgangs ist die Abstandsbildfunktion  $r(u, v)$ . Über die Bildaufnahme sei weiterhin bekannt, daß alle Abstände einen maximalen Betrag nicht überschreiten. Der Bildaufnahmevorgang wird in Abb. 2.6 verdeutlicht.

**Definition 2.1 (digitale Abstandsbildfunktion)** Die digitale Abstandsbildfunktion  $r$  ordnet jeder Bildkoordinate  $(u, v)$  den durch die Funktion  $q$  quantisierten Wert des Abstandes  $d$  zu einem Punkt  $x$  der Umgebung in Richtung  $(\phi_u, \theta_v)$  zu:

$$\begin{aligned}
 r(u, v) &= q \circ d(\phi_u, \theta_v) && \text{mit} \\
 \phi_u &= \phi_{min} + u\Delta\phi && u = 1, \dots, u_{max} \\
 \theta_v &= \theta_{min} + v\Delta\theta && v = 1, \dots, v_{max} \\
 d &\leq d_{max}
 \end{aligned}$$

$$\mathbf{x}(d, \phi_u, \theta_v) = \begin{pmatrix} d \cos \phi_u \sin \theta_v \\ d \sin \phi_u \sin \theta_v \\ d \cos \phi_u \end{pmatrix}$$

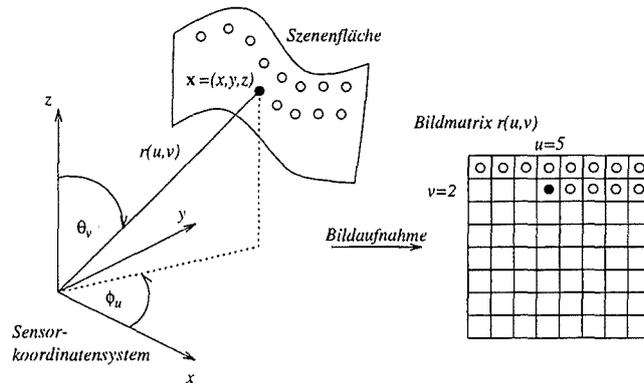


Abbildung 2.6: Abtastung von Szenenflächen bezogen auf das Kugelkoordinatensystem des Sensors und Repräsentation als Abstandsbildmatrix: die Lage des Sensors ist hier durch das  $(x, y, z)$ -Koordinatensystem repräsentiert. Szenenflächen werden in Kugelkoordinaten abgetastet. Die Laserstrahlrichtung wird durch  $\phi_u$  und  $\theta_v$  festgelegt.  $\phi_u$  sowie  $\theta_v$  stellen äquidistante Winkelwerte dar. Ein Abstandswert  $r$  in Richtung  $(\phi_u, \theta_v)$  wird in der Bildmatrix an der Stelle  $(u, v)$  eingetragen.

Alle Abstände liegen unterhalb einer Schwelle  $d_{max}$ . Die Bildaufnahme ist bestimmt durch die Winkelinkremente  $\Delta\phi$  und  $\Delta\theta$ , die Startwinkel  $\phi_a$  und  $\theta_a$ , die horizontale bzw. vertikale Bildauflösung  $u_{max}$  bzw.  $v_{max}$  in Richtung von  $\phi$  bzw.  $\theta$ .

Da die Transformationsvorschrift zwischen Bild- und Szenenbereich explizit in Form der Funktion  $x$  vorliegt, ist es möglich in demjenigen Bereich zu operieren, der für eine bestimmte Aufgabe am besten geeignet ist. Abgetastete Punkte im Bild- und Szenenbereich werden in Abb. 2.8 verdeutlicht. Die synthetische Testszene in Abb. 2.8a wurde mit Hilfe eines Sensorsimulators [Cramer 91] abgetastet und die abgetasteten Punkte im Bildbereich veranschaulicht (siehe Abb. 2.8b).

Im Abstandsbild in Abb. 2.8b erscheinen gerade Kanten der Szene in Abb. 2.8a als gekrümmte Linien. Der Effekt der Verzerrung wird in Abb. 2.7 erklärt. Diese Verzerrung hat zur Folge, daß bestimmte Operationen in der nachgelagerten Bildauswertung nicht direkt im Abstandsbild durchgeführt werden dürfen. Beispielsweise könnte aufgrund dieser Verzerrung ein Winkel zwischen Linien falsch berechnet werden.

## 2.4 Wissen über Szenen

Das Wissen über die Struktur der hier betrachteten Szenen soll hier genauer spezifiziert werden. Szenen bestehen aus trihedralen, starren und statischen Polyedern, die sich berühren können. Starre und statische Polyeder sind dreidimensionale Körper, die durch eine endliche Anzahl von Flächen begrenzt werden und deren Form und Lage sich zeitlich nicht verändert. Trihedrale Polyeder wiederum sind solche Polyeder, deren räumliche Eckpunkte durch den Schnitt genau dreier ebener Flächen entstehen. Zwei sich schneidende Flächen bilden unter dieser Annahme gerade Kanten.

In diesem Zusammenhang sollen zunächst einige Begriffe definiert werden: (1) Eine *Polyederkante* ist die nicht-leere Schnittmenge zweier Polyederflächen. Analog

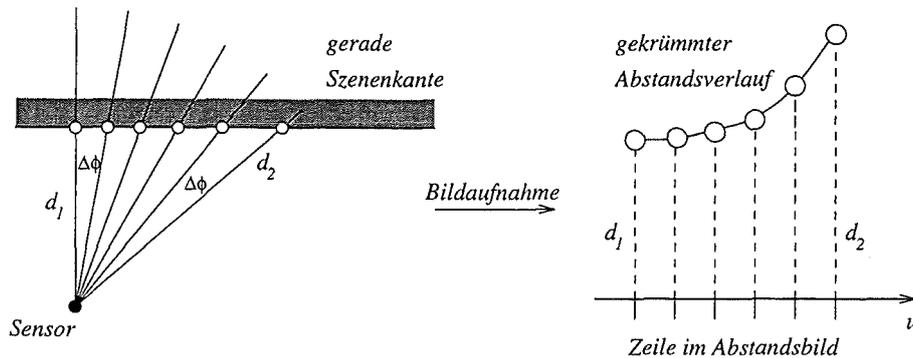
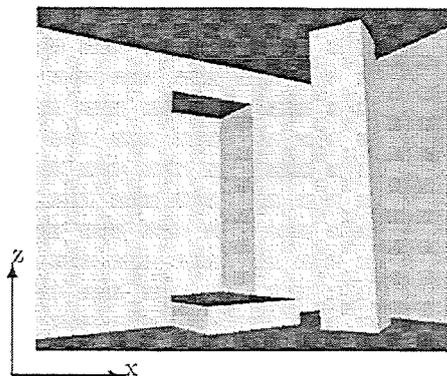


Abbildung 2.7: Auswirkung der ungleichförmigen Abtastung von Szenenkanten: bezogen auf die Aufnahme von geraden Szenenkanten besitzen im Abtastvorgang aufeinanderfolgende Strahlen gleichen Winkelabstand  $\Delta\phi$ , jedoch stehen die euklidischen Abstände  $d_1$  und  $d_2$  trotz der Abtastung einer geraden Kante in einem nicht-linearen Verhältnis (linke Teilabb.). Dies wird dadurch deutlich, daß gerade Szenenkanten im Abstandsbild einen gekrümmten Abstandsverlauf (rechte Teilabb.) und gekrümmte Linien hervorrufen.

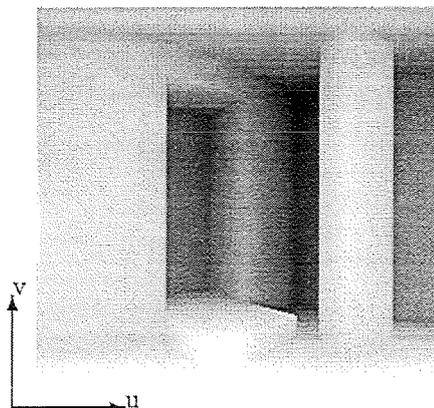
ist die *Polyederecke* die nicht-leere Schnittmenge dreier Polyederflächen. Die Polyederkante und der Polyedereckpunkt sind also blickpunktunabhängig. (2) Eine *Kante* ist eine von einem Blickpunkt aus sichtbare, zusammenhängende Teilmenge einer Polyederkante. (3) Eine *Linie* ist die perspektivische Projektion einer Kante in die Bildebene, die senkrecht zur Blickrichtung des Sensors steht. (4) *Szenenkantentypenpunkte* sind dreidimensionale Abtastpunkte der Szene, die auf Kanten liegen. (5) *Bildkantentypenpunkte* sind Punkte im Bild, die Szenenkantentypenpunkten entsprechen, d.h. Bildkantentypenpunkte liegen auf Linien. (6) Ein *Vertex* ist ein Szenenkantentypenpunkt, in dem sich drei Kanten treffen. (7) Eine *Linienverbindung* ist ein Bildkantentypenpunkt, der mit den Endpunkten mehrerer Linien identisch ist. (8) Die Menge aller Linien und Linienverbindungen wird als *Strichzeichnung* bezeichnet. Falls eine erste Kante eine zweite Kante so verdeckt, daß die zweite nur noch teilweise sichtbar ist, liegt aufgrund der obigen Definitionen keine Linienverbindung vor. In der Strichzeichnung können sich die beiden Linien zwar berühren, jedoch stellt dieser Berührungspunkt zumindest für die erste Linie kein Linienendpunkt dar.

Nach der physikalisch-geometrischen Ausprägung von Szenenkanten in polyedrischen Szenen lassen sich Sprungkanten und Dachkanten unterscheiden. Dachkanten zeichnen sich durch eine Unstetigkeit in der Änderung der Flächennormale aus, während sich Sprungkanten durch eine Unstetigkeit in Abständen auszeichnen. Dachkanten unterteilen sich weiterhin in konvexe und konkave Kanten, Sprungkanten in verdeckende und verdeckte Kanten (siehe Abb. 2.9). Die verdeckende Kante liegt dabei im Vordergrund und erzeugt die verdeckte Kante als Übergang zwischen abgeschattetem und sichtbarem Teil der Fläche im Hintergrund. Konvex (+), konkav (-), verdeckend (→) und verdeckt (→, jedoch unterbrochen gezeichnet) werden im folgenden als Kantentypen, die in Klammern aufgeführten Symbole als Marken bezeichnet. Verdeckende Linien werden nach Konvention durch Angabe einer Richtung so orientiert, daß sich die Fläche *im Vordergrund* rechts ihrer Richtung befindet. Eine verdeckte Linie wird so orientiert, daß sich die Fläche *im Hintergrund* rechts ihrer Richtung befindet. Eine markierte Strichzeichnung einer polyedrischen Szene ist in Abb. 2.10 gezeigt.

Da jede Linie durch perspektivische Projektion einer Kante bestimmten Typs

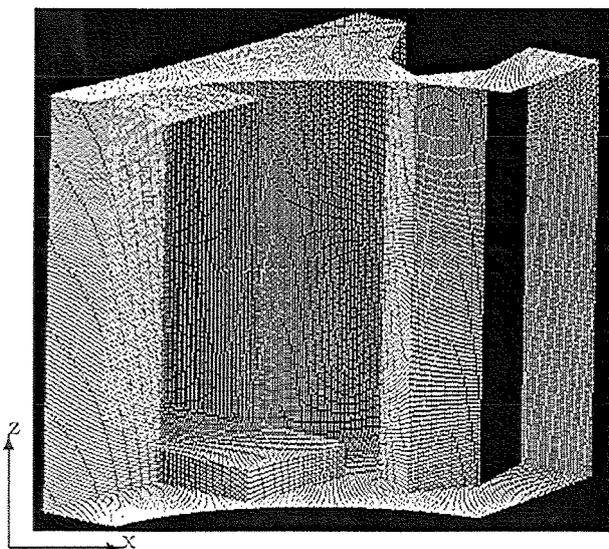


(a) Perspektivische Projektion einer synthetischen Testszene bestehend aus Boden, Decke, Wänden, einer Säule und einer Stufe in einer Wandaussparung



(b) Abstandsbild  $r(u, v)$  der synthetischen Testszene in Grauwertkodierung: helle Bildpunkte kodieren geringe Abstände, dunkle Bildpunkte kodieren hohe Abstände.

Zu beachten ist, daß z.B. die gerade Kante in Teilabbildung (a), wo sich Decke und die linke Wand treffen, im Abstandsbild als gekrümmte Linie abgebildet wird.



(c) Perspektivische Projektion der abgetasteten, dreidimensionalen Meßpunkte: Der Blickpunkt, der für die perspektivische Projektion verwendet wurde, wurde gegenüber der ursprünglichen Lage des Sensorkoordinatensystems verschoben, um die dreidimensionale Anordnung der abgetasteten Meßpunkte zu veranschaulichen.

Zwischen Meßpunkten am rechten Rand der Säule und den Meßpunkten auf der dahinterliegenden Wand besteht ein Abstandssprung. In realen Abstandsbildern liegen aufgrund des Mixed-Point-Effektes jedoch einzelne Meßpunkte in diesem Abstandsbereich.

Abbildung 2.8: Perspektivische Projektion einer synthetischen Testszene (a), Abstandsbild der synthetischen Testszene in Grauwertkodierung (b) und perspektivische Projektion der abgetasteten, dreidimensionalen Meßpunkte (c)

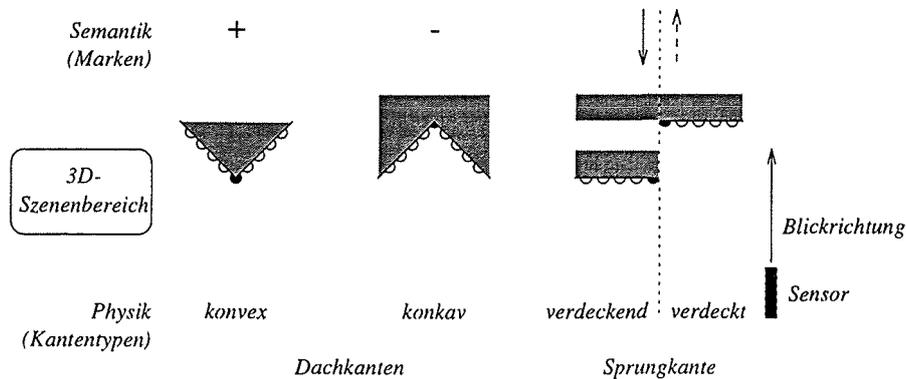


Abbildung 2.9: Kantentypen: Die geometrische Gestalt der Szenenkanten ist grau gezeichnet. Von der Sensorblickrichtung aus sind Abtastpunkte eingezeichnet. Jeder Kantentyp wird durch eine symbolische Marke repräsentiert.

entsteht, läßt sich dieser Typ bzw. die entsprechende Marke auch der zugehörigen Linie zuordnen. Man spricht dann von markierten Linien, markierten Linienverbindungen und markierten Strichzeichnungen. Linienverbindungen wurden bisher vor allem für folgende Zwecke eingesetzt:

- Interpretation nicht-markierter Strichzeichnungen
- Vorhersage und Verifikation fehlender Linien einer Strichzeichnung
- Linienverbindungen als Vertexmerkmale zu erkennender Objekte

Die Interpretation nicht-markierter Strichzeichnungen spielt vor allem bei der Interpretation von Videobildern eine bedeutende Rolle. Mit Bildverarbeitungsverfahren sind zwar Kanten als Helligkeitsunterschiede detektierbar, jedoch ist eine Aussage über die physikalisch-geometrische Ausprägung von Linien im Sinne einer markierten Strichzeichnung nicht direkt ableitbar. Huffman und Clowes [Huffman 71, Clowes 71] erklärten etwa zur gleichen Zeit die räumliche Interpretierbarkeit von Strichzeichnungen am Beispiel trihedral polyedrischer Szenen systematisch. Unter bestimmten Annahmen haben sie alle möglichen Linienverbindungen klassifiziert. Der Markierungsvorgang, der in der Literatur als Waltz-Filterung bekannt ist [Waltz 75], nutzt die Linienverbindungen, um jeder Linie einer Strichzeichnung eine Marke zuzuordnen.

Statt zur Interpretation von Strichzeichnungen wurde eine Aufzählung von Linienverbindungen in [Sugihara 79] zur Kantenextraktion aus Abstandsbildern trihedral polyedrischer Szenen verwendet. Abstandsbilder zeichnen sich dadurch aus, daß die Information über die geometrische Ausprägung von Kanten explizit enthalten ist und durch Bildverarbeitungsoperationen extrahiert werden kann. Da sich die Kantentypen aus Abstandsbildern algorithmisch ermitteln lassen, ist keine Deutung von Strichzeichnungen notwendig. Sugihara verwendete die Zwangsbedingungen, die durch Linienverbindungen einer Strichzeichnung auferlegt werden, zur Vorhersage fehlender Linien und zur Verifikation bereits extrahierter Linien. Im Gegensatz zu den Linienverbindungen von Huffman und Clowes besitzen die Linienverbindungen zwei Besonderheiten. Erstens werden auch bestimmte Berührungen von Polyedern erlaubt. Zweitens existieren auch Linienverbindungen mit verdeckten Linien, so daß auch Linienverbindungen mit vier Linien auftreten. Da verdeckte Linien in einer

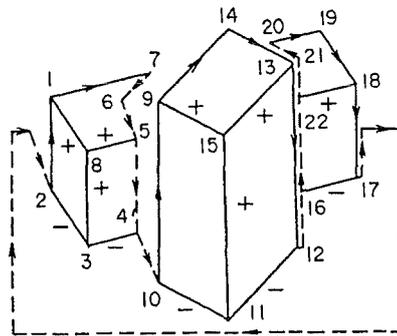


Abbildung 2.10: Verdeckende, verdeckte und teilweise sichtbare Linien in Strichzeichnungen: *verdeckende Linien* sind nicht-unterbrochen gezeichnet und mit einem Pfeil gekennzeichnet. Die Linie zwischen Linienverbindungen 10 und 9 ist beispielsweise verdeckend. *Verdeckte Linien* sind unterbrochen gezeichnet. Die Linie zwischen Linienverbindungen 5 und 4 stellt beispielsweise eine verdeckte Linie dar. Da die abgebildete Strichzeichnung von einem Abstandsbild eines Triangulationslaserscanners stammt, sind verdeckende und zugehörige verdeckte Linien räumlich versetzt abgebildet. *Teilweise sichtbare Linien* besitzen mindestens einen Endpunkt, an dem keine weiteren nicht-verdeckten Linien zusammentreffen. Die Linie zwischen den Linienverbindungen 5 und 8 sind beispielsweise teilweise sichtbar. (Abbildung aus [Sugihara 79])

geometrischen Beziehung zu verdeckenden Linien stehen, kann diese Beziehung zur Aufstellung weiterer Zwangsbedingungen herangezogen werden.

Godin und Levine [Godin & Levine 89] verwenden Linienverbindungen weder zur Interpretation von Strichzeichnungen noch zur Kantenextraktion, sondern als charakteristische Merkmale von Vertizes. Da eine Linienverbindung die Ansicht von dreidimensionalen Kanten mit gemeinsamem Vertex charakterisiert, kann die Linienverbindung als blickpunktabhängiges Merkmal dieses Vertex dienen. Godin und Levine betrachteten Abstandsbilder von Szenen, die nicht nur aus ebenen, sondern auch aus gekrümmten Flächen bestehen, die ihrerseits durch Kanten begrenzt sind. Polyedrische Szenen sind also ebenfalls erlaubt. Als Vertexmerkmale wurden die Linienverbindungen von [Malik 87] herangezogen. Da in der Arbeit von Malik Linienverbindungen für Objekte mit gekrümmten Flächen untersucht werden, ist die Berührung von gekrümmten Flächen zweier Objekte von untergeordneter Bedeutung. Die Flächen müßten dazu genau eine komplementäre Form besitzen, um einen flächigen Kontakt auszubilden.

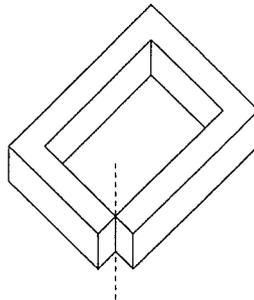
In dieser Arbeit sollen Linienverbindungen ebenfalls als Merkmale für Vertizes herangezogen werden, um eine möglichst informationsreiche symbolische Beschreibung von Abstandsbildern polyedrischer Szenen erzeugen zu können. Im Gegensatz zu Godin und Levine bestehen die Szenen hier aber nicht aus gekrümmten Objekten sondern aus Polyedern, die sich auch flächig berühren dürfen. Da die Linienverbindungen in [Sugihara 79] gerade auch sich berührende trihedrale Polyeder erlauben, werden sie hier verwendet. Da die Herleitung der Linienverbindungen bestimmte Einschränkungen voraussetzt, werden diese im nächsten Unterabschnitt erläutert. Im darauffolgenden Unterabschnitt wird grob die Strategie erläutert, wie Linienverbindungen speziell aus den hier vorliegenden Laserradarabstandsbildern gewonnen werden.

### 2.4.1 Modell von Sugihara

Zusätzlich zur trihedral polyedrischen Szenenstruktur werden in dem Modell von Sugihara je eine Restriktion bezüglich Szene und Blickpunkt eingeführt. Zusammenfassend lassen sich dann die Restriktionen bezüglich der Szene und des Blickpunktes durch folgende Bedingungen ausdrücken:

**Bedingung 2.1 (Szene)** Die Szene ist eine Menge polyedrischer Körper. Genau drei Flächen treffen an jedem Vertex eines Polyeders zusammen. Folgende Anordnungen treten in der Szene nicht auf: Weder zwei verschiedene Kanten, noch zwei verschiedene Vertizes teilen sich eine gemeinsame Position. Kein Vertex liegt auf einer Kante eines anderen Körpers.

Die Szene wie in folgender Abbildung ist also nicht erlaubt, da sich in der unterbrochen gezeichneten Linie 4 Polyederflächen treffen.



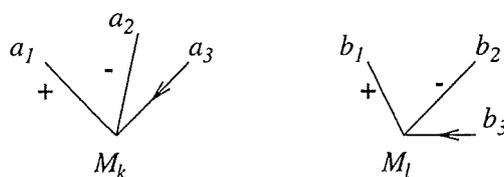
**Bedingung 2.2 (Blickpunkt)** Die Lage des Blickpunktes schließt aus, daß ein Sehstrahl in einer Polyederfläche liegt.

Eine Linienverbindung heißt nun *möglich* genau dann, wenn die Linienverbindung unter Einhaltung der Bedingungen 2.1 und 2.2 in der Strichzeichnung des Bildes einer Szene erscheinen kann. Polyeder der Szenen dürfen sich weiterhin nur in bestimmter Art und Weise berühren. Die erlaubten Berührungs- bzw. Kontaktarten kommen durch folgende Bedingung zum Ausdruck.

**Bedingung 2.3 (Kontaktarten)** Zwischen Polyedern sind die folgenden Kontaktarten erlaubt: (a) Fläche mit Fläche (b) Vertex mit Fläche (c) Kante mit Fläche und (d) Kantenpunkt mit Kantenpunkt, wobei die Kantenpunkte keine Vertizes sein dürfen.

Die Kontaktarten werden in Abbildung 2.11 veranschaulicht.

Unter der Annahme vier verschiedener Arten, wie sich Polyeder berühren dürfen, hat Sugihara die möglichen Linienverbindungen klassifiziert und die einzelnen Klassen systematisch aufgezählt. Zwei Linienverbindungen  $M_k, M_l$  werden dann als zur gleichen Klasse zugehörig angesehen, wenn die Linien die folgenden Bedingungen erfüllen:



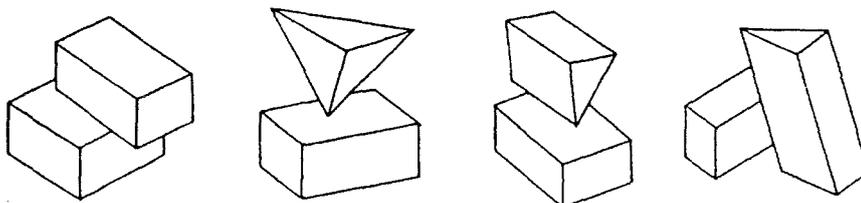


Abbildung 2.11: Zwischen Polyedern erlaubte Kontaktarten (siehe Bedingung 2.3)

**Bedingung 2.4 (Linienanzahl)** Die Linienverbindungen  $M_k$  und  $M_l$  besitzen die gleiche Anzahl  $n$  an Linien und die Linien  $a_1, \dots, a_n$  bzw.  $b_1, \dots, b_n$  von  $M_k$  bzw.  $M_l$  sind im Uhrzeigersinn angeordnet.

**Bedingung 2.5 (Linientypen)** Der Linientyp von  $a_i$  ist gleich dem Linientyp von  $b_i$  für alle  $i = 1, \dots, n$ .

**Bedingung 2.6 (Winkel zwischen Linien)** Die Winkel zwischen  $a_i$  und  $a_j$  und zwischen  $b_i$  und  $b_j$  sind beide entweder kleiner  $180^\circ$ , gleich  $180^\circ$  oder größer als  $180^\circ$  für alle  $i, j = 1, \dots, n$ .

Die Winkelbedingung drückt aus, daß eine kleine Blickpunktänderung nur in einer kleinen Änderung korrespondierender Winkel zwischen Kantenelementen resultiert. Die quantitative Größe des Winkels wird daher nicht benötigt.

Sugihara hat alle möglichen Linienverbindungen in insgesamt 20 Klassen klassifiziert. Die systematische Aufzählung der Repräsentanten der Linienverbindungsklassen wird in Anhang A kurz skizziert. Die Repräsentanten der Linienverbindungsklassen werden hier auch kurz als Linienverbindungen bezeichnet.

## 2.4.2 Linienverbindungen als Vertexmerkmale

Um Linienverbindungen als Vertexattribute verwenden zu können, müssen zunächst die Linienverbindungen für den Verwendungszweck der Objekterkennung angepaßt werden.

Sugihara hat zur Vorhersage und Verifikation fehlender Linien auch solche Linienverbindungen benutzt, die auch verdeckte Linien besitzen. Diese Linienverbindungen sind in Abb. A.3 ersichtlich. Dadurch wird es ermöglicht, eine räumliche Beziehung zwischen verdeckten und verdeckenden Linien einer Linienverbindung zu verwenden. Trifft nämlich eine verdeckte mit einer verdeckenden Linie in einer Linienverbindung zusammen, ist die entsprechende verdeckte Kante durch Abschattung aus der verdeckenden Kante entstanden. Diese Beziehung kann zwar zur Aufstellung von Zwangsbedingungen herangezogen werden, jedoch liefern verdeckte Kanten keine geometrische Information über zu erkennende Objekte, da verdeckte Kanten durch Abschattung während des Aufnahmevorgangs entstehen. Die Linienverbindungen von Sugihara werden daher dahingehend angepaßt, daß verdeckte Linien aus allen Linienverbindungen entfernt werden. Aus den 20 Linienverbindungen von Sugihara entstehen dadurch direkt die hier verwendeten Linienverbindungen, die in dem oberen Teil von Abb. 2.12 mit Indizes von 1 bis 20 durchnummeriert sind.

Eine zweite Anpassung wird notwendig, weil auch Vertizes teilweise sichtbarer Kanten charakterisiert werden sollen. Ein Beispiel einer teilweise sichtbaren Kante

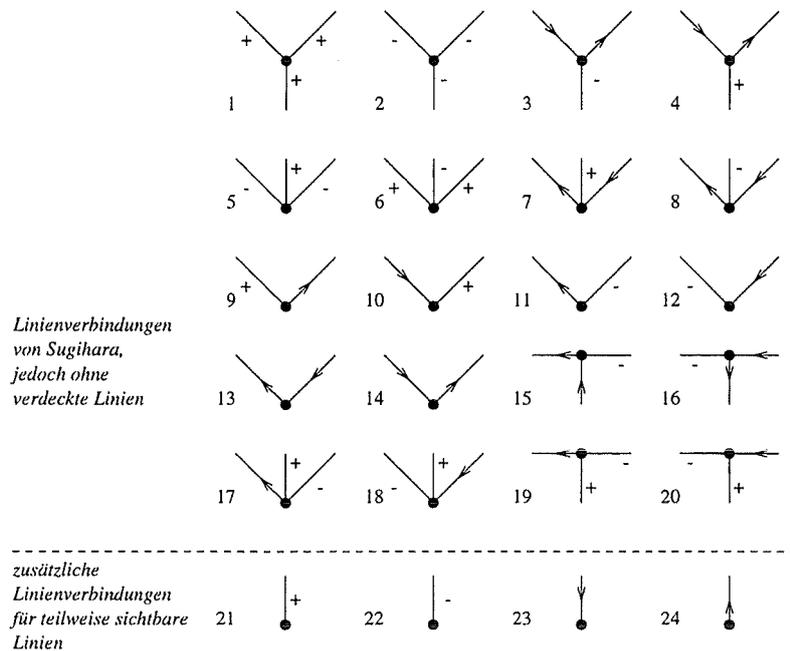


Abbildung 2.12: Angepaßte und ergänzte Liniverbindungen von Sugihara als Vertextypen: die Liniverbindungen 1 bis 20 von Sugihara wurden leicht verändert (oberer Teil). Die Liniverbindungen 21 bis 24 wurden ergänzt (unterer Teil). Die Indizes der Liniverbindungen dienen als Vertextypen.

bzw. Linie ist in Abb. 2.10 als Linie zwischen Linienvbindung 5 und Linienvbindung 8 ersichtlich. Teilweise sichtbare Kanten können durch Verdeckung oder aufgrund der Blickfeldbegrenzung entstehen. Hier werden teilweise sichtbare Kanten und verdeckte Kanten begrifflich streng getrennt. In Abb. 2.10 ist die Linie zwischen den Linienvbindungen 5 und 8 teilweise sichtbar, während die Linie zwischen den Linienvbindungen 4 und 5 verdeckt ist. Eine teilweise sichtbare Linie zeichnet sich dadurch aus, daß mindestens ein Liniende einer Linienvbindung angehört, die keine weiteren nicht-verdeckte Linien besitzt. Durch eine zweite Anpassung der Linienvbindungen von Sugihara, nämlich durch das Entfernen der verdeckten Linien, entstehen die Pseudo-Linienvbindungen, die in dem unteren Teil der Abb. 2.12 mit den Indizes 21 bis 24 durchnummeriert sind. Die zweite Anpassung besteht also in einer Erweiterung.

Nun wird geklärt, wie die angepaßten Linienvbindungen speziell für den Fall der hier verwendeten Abstandsbilder korrekt bestimmt werden können. Sugihara konnte die Linienvbindungen direkt im Abstandsbild bestimmen, da er Bilder eines Triangulationslaserscannern verwendet hat. Wie bereits erwähnt, zeichnen sich Abstandsbilder von Triangulationslaserscannern dadurch aus, daß gerade Kanten auch als gerade Linien im Abstandsbild erscheinen. Aufgrund der ungleichförmigen Abtastung des hier verwendeten Laserradarscannern, werden gerade Kanten allerdings auf i.a. gekrümmte Linien abgebildet. Gerade Linien sind aber für die Berechnung der zwischen Linien eingeschlossenen Winkel erforderlich, um Linienvbindungen mit gleicher Linienzahl und Linienmarkierung korrekt unterscheiden zu können.

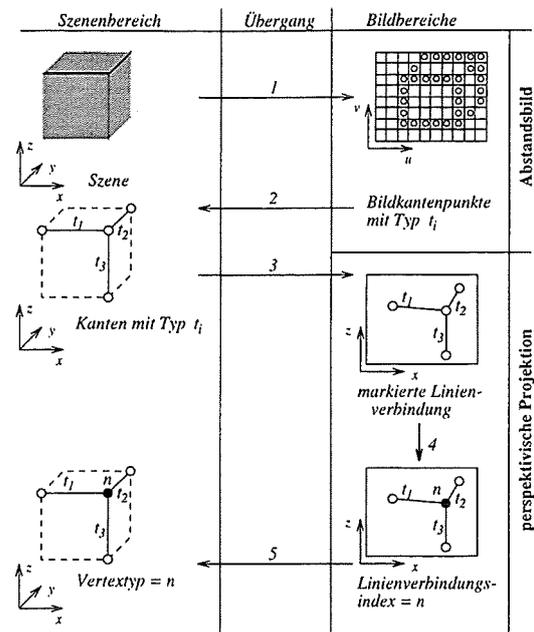
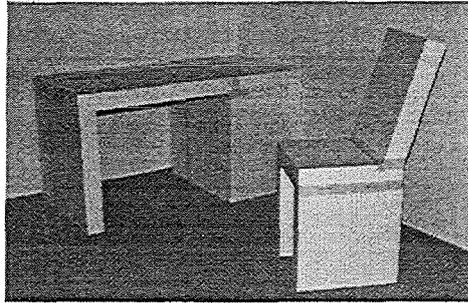


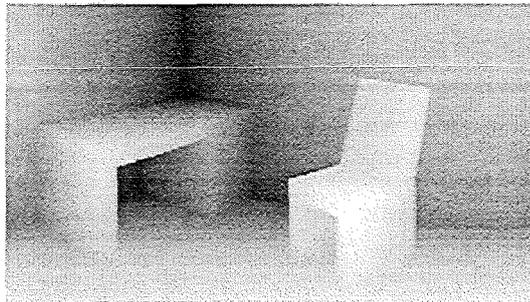
Abbildung 2.13: Perspektivische Projektion zur Bestimmung der Vertextypen: für die hier verwendeten Laserradarbilder können Linienverbindungen nicht im Abstands-bild direkt ermittelt werden. Durch den Umweg über die Bildebene bzgl. der perspektivischen Projektion erhält man neben dem Abstands-bild einen weiteren Bildbereich.

Das Modell von Sugihara ist dennoch anwendbar, wenn es gelingt, zunächst aus dem Abstands-bild dreidimensionale Kanten zu gewinnen, die in eine geeignete Bildebene perspektivisch projiziert werden. Geeignete Bildebenen stehen senkrecht zur  $y$ -Achse des Sensorkoordinatensystems. Die resultierenden Linien bilden dann zum Teil Linienverbindungen aus, die unter anderem durch den zweidimensionalen Winkel zwischen Linien charakterisiert werden können.

Diese Vorgehensweise bedeutet, daß zwei Bildbereiche verwendet werden: das Abstands-bild und die Bildebene der perspektivischen Projektion. Der Ablauf der Bildauswertung bezogen auf den Szenen- und die beiden Bildbereiche verdeutlicht Abbildung 2.13 in fünf Schritten. Im ersten Schritt wird von der dreidimensionalen Szene ein zweidimensionales Abstands-bild aufgenommen, in dem Bildkantenpunkte mit Kantentyp extrahiert werden können. Im zweiten Schritt werden aus diesen Kantenpunkten des zweidimensionalen Bildes dreidimensionale Kanten generiert. Der Typ  $t_i$  einer Kante entspricht dabei dem Typ der entsprechenden Kantenpunkte. Die nächsten Schritte dienen nur der Ableitung von Linienverbindungen. Zur Unterscheidung von Linienverbindungen werden in einem dritten Schritt die Kanten auf Linien in der Bildebene bezüglich der perspektivischen Projektion abgebildet. Der Typ einer Linie entspricht dabei dem Typ der entsprechenden Kante. Im vierten Schritt können anhand der markierten Linien und der eingeschlossenen Winkel die Linienverbindungen ermittelt werden. Im letzten Schritt wird der Index  $n$  der Linienverbindung als Typ eines dreidimensionalen Vertex verwendet.



(a) Photographie einer realen Testszene



(b) Abstandsbild der Testszene

Abbildung 2.14: Photographie und Abstandsbild einer realen Testszene

## 2.5 Reale Szenen und Abstandsbilder

Die Bildauswertung sowie die Erkennung von realen Abstandsbildern muß sowohl Eigenschaften der aufgenommenen Szenen als auch den Bildaufnahmevorgang durch den Laserradarscanner berücksichtigen. Bei Szeneneigenschaften kann man zwischen Oberflächeneigenschaften und der Szenengeometrie unterscheiden.

Die Geometrie der aufgenommenen Testszenen kann wie folgt charakterisiert werden. Die Szenen bestehen aus einfachen, konstruierten Objekten, die in der Ecke eines Raumes aufgestellt sind. Für die Objekte wurde nur aus Gründen der Prägnanz eine möbelähnliche Gestalt (eines Stuhls, eines Bürotisches und zweier Stufen) gewählt. Selbst einfache Möbelstücke sind geometrisch komplizierter aufgebaut. Die Objekte bestehen aus 10 cm dicken Styroporplatten, die durch Metalldrähte und Klebeband zusammengehalten werden. Die Styroporobjekte erfüllen die Forderung nach polyedrischen Objekten sehr gut, da die Kanten sehr scharf sind und die Oberflächen der Platten planare Flächen gut annähern. Eine weitere Eigenschaft der Testszenen besteht darin, daß viele verschiedene Kantenlängen auftreten, jedoch wenige verschiedene Winkel. Diese Eigenschaft dürfte auch eine Gemeinsamkeit mit Objekten wie Schächten und Säulen in großtechnischen Umgebungen darstellen.

Die in den Testszenen auftretenden Oberflächen können wie folgt charakterisiert werden. Die Oberflächen von Teppichboden, Rohfasertapete und Styropor sind nicht transparent, absorbieren sehr wenig und reflektieren das Laserlicht gut. In Experimenten konnten Abstände zu schwarzen und matten Flächen aufgrund von Absorption nicht erfaßt werden. Ähnliche Probleme bereiten spiegelnde Oberflächen wie Chromrohre von Stühlen.



Abbildung 2.15: Versuchsaufbau des Laserradarscanners: der Laserradarscanner ist über eine Interfacekarte mit einem Personalcomputer verbunden. Der Sensor wird über ein eigenes externes Netzteil versorgt. Scanner und PC sind auf einem fahrbaren Tisch aufgestellt. Im Falle der aufgenommenen Testszenen liegt der Sensorblickpunkt etwa 1,60 m über der Bodenebene. Im Hintergrund ist die zweite Testszene aufgebaut.

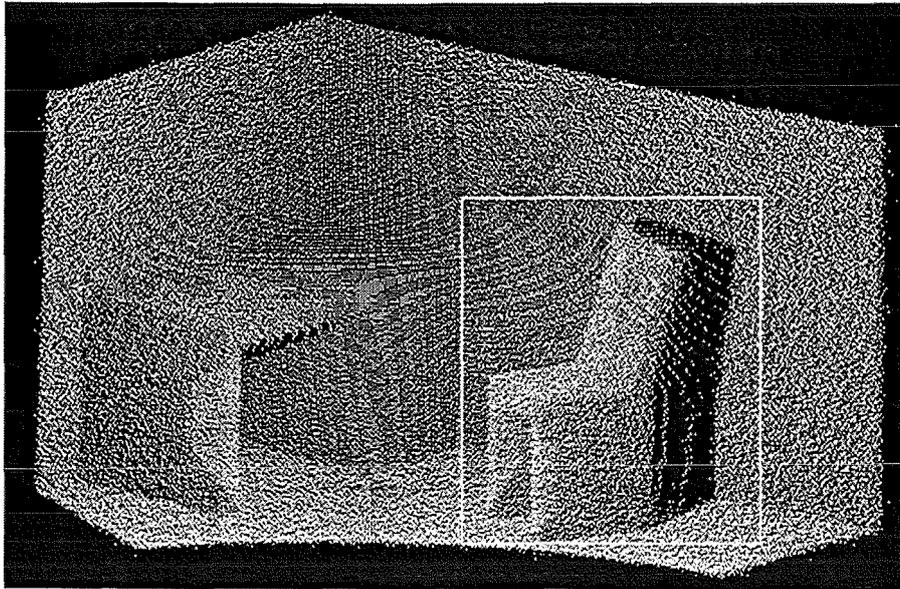
Unter anderem wurden Abstandsbilder zweier Testszenen aufgenommen. Die Bilder besitzen eine Auflösung von  $u_{max} = 400$  und  $v_{max} = 230$  Punkten. Das horizontale Blickfeld beträgt  $\phi_e - \phi_a = 70^\circ$ , das vertikale Blickfeld beträgt  $\theta_e - \theta_a = 40^\circ$ . Horizontales sowie vertikales Winkelinkrement sind mit  $\Delta\phi = \Delta\theta = 0.176^\circ$  gleich groß. Alle Abstandswerte liegen zwischen 2,5 m und 4 m. Die Aufnahmezeit für ein solches Abstandsbild liegt bei 7 bis 8 min. Photographie und Abstandsbild der ersten Testszene sind in Abb. 2.14 abgebildet. Um den Mixed-Point Effekt in realen Abstandsbildern zu verdeutlichen, wurden die abgetasteten Szenenpunkte in Abb. 2.16a visualisiert. In Abb. 2.16b ist der Ausschnitt des stuhlförmigen Styroporobjektes vergrößert dargestellt. Auf der Stuhlrückseite sind zwischen Vorder- und Hintergrund die durch den Mixed-Point Effekt verfälschten Meßpunkte zu sehen. Der schwarze Bereich liegt im Schatten des Stuhles.

## 2.6 Zusammenfassung

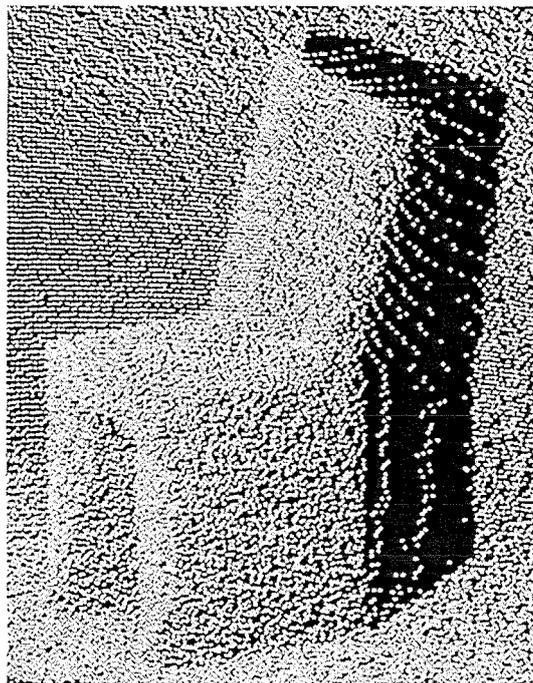
Zusätzlich zu einem gewissen Rauschanteil in jedem Bildpunkt tritt in Bildpunkten mit hohen Abstandsdifferenzen der Mixed-Point Effekt auf, der bei der Extraktion verdeckender Kanten gesondert berücksichtigt werden muß. Die Extraktion verdeckender Kanten stellt ein zentrales Problem dar, um aus Abstandsbildern von Laserradarscannern eine Szenenkantenstruktur zu generieren.

Durch Einbindung der perspektivischen Projektion lassen sich Winkel zwischen Linien und damit auch Linienverbindungen trotz der ungleichförmigen Abtastung des Laserradarscanners korrekt unterscheiden. Es werden daher in dieser Arbeit zwei Bildbereiche verwendet: das Abstandsbild und die perspektivische Projektion bereits extrahierter dreidimensionaler Kanten.

Linienverbindungen werden in dieser Arbeit nicht zur Markierung von Strichzeichnungen oder zur Verifikation von Linien, sondern als Vertexattribute verwen-



(a) Visualisierung der dreidimensionalen Meßpunkte der realen Testszene in der perspektivischen Projektion. Der Ausschnitt des Stuhles ist weiß gekennzeichnet.



(b) Auf der Stuhlrückseite sind zwischen Vorder- und Hintergrund die durch den Mixed-Point Effekt verfälschten Meßpunkte zu sehen. Der schwarze Bereich liegt im Schatten des Stuhles.

Abbildung 2.16: Perspektivische Projektion der dreidimensionalen Abtastpunkte der realen Testszene (a) und Wirkung des Mixed-Point Effektes (b).

det. Da Linienverbindungen von der Markierung aller inzidenten Linien abhängig sind, stellen sie Merkmale mit hohem Informationsgehalt dar. Die Linienverbindungen von Sugihara werden hier für Abstandsbilder von Laserradarscannern angepaßt und zum Zweck der Objekterkennung um Pseudolinienverbindungen mit teilweise sichtbaren Linien erweitert.



## Kapitel 3

# Bildbereichshinweise

Bildbereichshinweise sind Merkmale eines Bildes, die auf charakteristische Merkmale der aufgenommenen Szene hindeuten. Zur Extraktion von Bildbereichshinweisen aus Abstandsbildern bestehen zwei verschiedene Vorgehensweisen: die Detektion von Kanten und die Detektion von homogenen Regionen. Während homogene Regionen auf Polyederflächen hindeuten, weisen Kantenpunkte auf Polyederkanten hin. Allerdings ist der Aufwand zur Detektion homogener Regionen vergleichsweise hoch, da das Verfahren in mehreren Iterationen arbeitet und aufwendige Operationen benötigt. Der Aufwand lohnt sich insbesondere dann, wenn Merkmale von gekrümmten Flächen oder von Flächen mit Löchern detektiert werden sollen.

Der Vorteil der Kantendetektion dagegen liegt vor allem darin, daß die Operationen bezogen auf Schnelligkeit und Datenhaltung weniger Aufwand erfordern, was im Hinblick auf Echtzeitanforderungen wichtig ist. Insbesondere wenn bekannt ist, daß wenige, beispielsweise maximal drei Kanten zwischen planaren Flächen zusammentreffen, ist die Kantendetektion geeignet. Die Kantendetektion in Abstandsbildern wird außerdem mit der Verfügbarkeit verbesserter bildgebender Abstandssensoren weiter an Bedeutung gewinnen.

Die Kantendetektion soll verdeckende, konvexe und konkave Bildkantenpunkte unterscheiden können. Bei der Detektion verdeckender Kanten soll speziell der Mixed-Point Effekt berücksichtigt werden. Außerdem sollen Bildkantenpunkte einer Kante benachbart sein, so daß benachbarte Bildkantenpunkte verkettet werden können.

### 3.1 Übersicht Kantendetektion

Das Ziel der Kantendetektion besteht in der Unterscheidung von Bildkantenpunkten, um auf entsprechende Szenenkanten schließen zu können. Konvexe und konkave Kanten werden auch als Dachkanten, verdeckende und verdeckte Kanten als Sprungkanten bezeichnet. Sprungkanten zeichnen sich durch eine Unstetigkeit in Abständen, Dachkanten durch eine Unstetigkeit in der Änderung der Oberflächennormale aus.

Die Kantendetektion in Abstandsbildern kann man getrennt nach Sprungkanten- und Dachkantendetektion betrachten. Die Detektion von Dachkanten in Abstandsbildern ist schwieriger, da keine Abstandsunstetigkeit, sondern eine Unstetigkeit bzgl. der Änderung der Oberflächennormalen detektiert werden muß. Ein Hinweis auf Sprungkanten sind hohe Abstandssprünge innerhalb einer lokalen Umgebung.

Technik	Referenz
Anpassen von 2-dim. Funktionen $m$ -ter Ordnung	[Mitiche & Aggarwal 83]: $m = 1$ [Haralick <i>et al.</i> 83]: $m = 3$ [Langridge 84]: $m = 1$ [Besl & Jain 85a]: $m = 2$ [Fan 90]: $m = 2$
Ringoperatoren	[Inokuchi <i>et al.</i> 82]
morphologische Operatoren	[Gupta & Krishnapuram 91]
3D-Momente	[Ghosal & Mehrotra 92]
Auswertung von Bildpunkten auf Höhenlinien	[Wani & Batchelor 94]

Tabelle 3.1: Klassifikation der Verfahren zur Kantendetektion in Abstandsbildern

Da Kanten in Videobildern die gleiche Eigenschaft besitzen, können Sprungkanten in Abstandsbildern wie in [Swietlik 94] durch einfache Gradientenoperatoren aus der Videobildauswertung wie der Roberts- oder Sobeloperator detektiert werden. Schwieriger ist allerdings die Unterscheidung zwischen verdeckenden und verdeckten Kanten. Die einfachste Unterscheidung von verdeckenden und verdeckten Kantenpunkten erfolgt dadurch, daß der Sprungkantenpunkt mit dem geringsten Abstand als verdeckend markiert wird. Diese Methode funktioniert allerdings nur für Abstandsbilder, die nicht geglättet sind und nicht durch den Mixed-Point Effekt verfälscht sind.

Im Hinblick auf das vorhandene Wissen über Szenen und die Bildaufnahme (siehe Abs. 2.3 und 2.4) sollen die bisherigen Verfahren zur Kantendetektion in Abstandsbildern dargestellt werden. Geeignete Ansatzpunkte sind die Übersichtsartikel von [Besl & Jain 85b], [Besl 88b], [Arman & Aggarwal 93] und das Buch von [Suk & Bhandarkar 93]. Die Ansätze können nach den Techniken in Tabelle 3.1 eingeteilt werden.

In den meisten Arbeiten zur Kantendetektion werden lokal ein- oder zweidimensionale Funktionen verschiedener Ordnung an Szenen- oder Bildpunkte angepaßt, um aus den Eigenschaften der Funktionen Rückschlüsse über Kantenpunkte zu ziehen. Verfahren, die nur zeilen- und spaltenweise eindimensionale Funktionen anpassen, sind rauschempfindlich. In dem Überblick werden daher nur solche Verfahren berücksichtigt, die zweidimensionale Funktionen an alle Punkte innerhalb eines Bildfensters anpassen.

Mitiche und Aggarwal [Mitiche & Aggarwal 83] entwickelten basierend auf einem probabilistischen Modell einen rauschunempfindlichen Detektor für Dachkanten. Für jede von zwei diskreten Bildrichtungen wird in einem Bildpunkt auf eine Dachkante getestet. Unter dieser Annahme werden pro Richtung je zwei Ebenen an die Pixel in einer Umgebung angepaßt. Wenn die eingeschlossenen Winkel nicht größer als ein bestimmter Schwellwert sind, wird kein Kantenpunkt angenommen. Im positiven Falle wird eine Richtung nach der Bayes'schen Wahrscheinlichkeit und einem weiteren Schwellwert ausgewählt. Problematisch ist vor allem das Finden geeigneter Schwellwerte und, daß Dachkanten nur in zwei diskreten Richtungen detektiert werden.

Haralick *et al.* [Haralick *et al.* 83] klassifizieren die Topographie digitaler Flächen, insbesondere von Videobildern. Dazu wird die digitale Fläche lokal mit einer bikubischen polynomiellen Fläche approximiert. Aus den ersten, zweiten und dritten partiellen Ableitungen der Polynomfläche und weiteren Größen lassen sich Bild-

punkte als einen von zehn verschiedenen Typen klassifizieren. Manche Flächentypen können auch Kanten in Abstandsbildern charakterisieren. Die meisten Flächentypen sind jedoch nicht invariant gegen Blickpunktveränderung, so daß einfachere Verfahren geeigneter sind.

Langridge [Langridge 84] detektiert und lokalisiert Unstetigkeiten in der ersten Ableitung von Flächen, die durch über einer Ebene irregulär angeordnete Abtastpunkte gegeben sind. Durch das Vorzeichen eines speziellen Operators können konvexe Kanten mit positivem, konkave Kanten mit negativem und Sprungkanten als Vorzeichenwechsel detektiert werden. Da irregulär angeordnete Daten erlaubt sind, müssen allerdings Nachbarschaftsberechnungen durchgeführt werden.

Besl und Jain [Besl & Jain 85a] nutzen die Methode von [Beaudet 78], um lokal die ersten und zweiten partiellen Ableitungen durch Anpassen eines biquadratischen Polynoms über beliebig großen Bildfenstern zu schätzen. Die Ableitungen werden zur Berechnung von Sprungkantendetektoren und von verschiedenen Krümmungen herangezogen, die die Unterscheidung von Dachkanten erlauben. Die vorgeschlagenen Sprungkantendetektoren erlauben keine direkte Unterscheidung von verdeckten und verdeckenden Kantenpunkten.

Fan et. al. [Fan et al. 87], [Fan 90] schlagen drei Verfahren zur Kantendetektion anhand von Krümmungsunstetigkeiten vor. Die einfachste Methode glättet das Bild zunächst mit einem einzigen Gaußfilter. Für jeden Bildpunkt werden Krümmungen in vier diskreten Richtungen bestimmt. Dachkanten werden als Krümmungsextrema und Sprungkanten als Nulldurchgänge bezogen auf die vier Richtungen in jedem Punkt detektiert. Ein Nachteil besteht darin, daß Krümmungen in nur vier diskreten Richtungen berechnet werden. Die Methode besitzt allerdings zwei Vorteile. Erstens können Sprungkanten durch einen Vorzeichenwechsel detektiert werden, so daß auch geglättete oder verschmierte Sprungkanten genau lokalisiert werden können. Zweitens ist aufgrund des Vorzeichenwechsels eine Unterscheidung zwischen verdeckender und verdeckter Seite einer Sprungkante möglich. Da sich der Vorzeichenwechsel zur Lokalisierung verdeckender Kantenpunkte heranziehen läßt, wird dieses Verfahren weiter unten genauer beschrieben. Die nächste Methode wendet die erste Methode auf eine Hierarchie verschieden stark geglätteter Bilder an, benötigt jedoch einen höheren Rechenaufwand. Die dritte Methode kombiniert eine adaptive Krümmungsberechnung mit einem Canny-Kantendetektor zur Lokalisierung von Sprungkanten und ist daher ebenfalls aufwendig. Weitere Ansätze zur Kantendetektion, die nicht durch Anpassen von Funktionen arbeiten, sind: Ringoperatoren, morphologische Operatoren und Momente.

Inokuchi [Inokuchi et al. 82] schlägt einen Ringoperator zur Kantendetektion vor. Der Ringoperator extrahiert eine eindimensionale Funktion, die aus Abstandswerten auf einem Ring rund um einen bestimmten Bildpunkt besteht. Mit Hilfe von Fourierkoeffizienten können Kantenpunkte verschiedenen Typs unterschieden werden. Allerdings muß für jeden Bildpunkt eine schnelle Fouriertransformation durchgeführt werden, was sehr aufwendig ist. Außerdem werden Abstände innerhalb des Rings nicht berücksichtigt, wodurch das Resultat rauschempfindlich ist.

Gupta und Krishnapuram [Gupta & Krishnapuram 91] haben untersucht, konvexe, konkave und Sprungkanten in Abstandsbildern mit Hilfe morphologischer Operatoren zu detektieren. Das Abstandsbild wird dazu als Gebirge betrachtet, das erodiert oder dilatiert werden kann. Allerdings können nicht alle, sondern nur spezielle konvexe und konkave Kantenpunkte extrahiert werden.

Ghosal und Mehrotra [Ghosal & Mehrotra 92] haben Momente zur Kantendetektion in Abstandsbildern verwendet. Die Momente betrachten das Abstandsbild als Masseverteilung innerhalb einer kreisförmigen Umgebung. Aus den Momenten

lassen sich fünf Parameter eines Kantenmodells berechnen. Mit Hilfe geeigneter Schwellwerte lassen sich Kanten verschiedenen Typs unterscheiden, jedoch kann es schwierig sein, Schwellwerte zu finden, die für alle Bildbereiche zufriedenstellende Ergebnisse liefern.

Wani und Batchelor [Wani & Batchelor 94] detektieren Kantenpunkte durch Auswertung von Bildpunkten gleichen Abstandes, die als Höhenlinien interpretiert werden können. Das Abstandsbild wird zunächst in Bildpunkte gleichen Abstandes unterteilt. Pro Höhenlinie werden sog. kritische Punkte ermittelt. Ausgehend von kritischen Punkten werden lokal begrenzt weitere Kantenpunkte verschiedenen Typs detektiert, indem spezielle Bedingungen auf Bildelementen eines lokalen Fensters geprüft werden. Die Bedingungen können an den Rauschanteil in Abstandsbildern angepaßt werden. Allerdings geht der Ansatz davon aus, daß zwischen den versetzt liegenden Flächen einer Sprungkante keine weiteren Meßpunkte liegen. So werden verdeckende Bildkantenpunkte als Endpunkte nicht-geschlossener Höhenlinien angenommen. Diese Endpunkte sind hier aber gerade von dem Mixed-Point Effekt betroffen.

### Unterscheidung verdeckender und verdeckter Kantenpunkte

Mit der Extraktion verdeckender Sprungkantenpunkte, an denen der Mixed-Point Effekt auftritt, haben sich bisher nur wenige Arbeiten beschäftigt. Hebert und Krotkov [Hebert & Krotkov 92] schlagen zwei Lösungsmöglichkeiten vor: (1) Anwenden von Medianfiltern auf 2D-Bildpunkte und (2) Entfernen isolierter 3D-Szenenpunkte. Medianfilter sortieren Abstandswerte innerhalb eines Bildfensters zum Beispiel aufsteigend und entnehmen den mittleren Wert der sortierten Folge als geglätteten Wert. Da an Sprungkanten der mittlere Wert aber gerade ein durch den Mixed-Point Effekt verfälschter Abtastpunkt repräsentiert, kann eine Medianfilterung keine Verbesserung bringen. In der zweiten Technik sollen isoliert auftretende 3D-Szenenpunkte entfernt werden. Isoliert auftretende 3D-Punkte müssen durch Abstandsdifferenzen zu Nachbarpunkten bestimmt werden, die einen gegebenen Schwellwert überschreiten. Verdeckende Punkte einer Kante, die nun mit Hilfe eines Standardverfahrens bestimmt werden können, müssen aber im Bild nicht mehr zusammenhängend sein, so daß eine Kantenpunktverkettung erschwert wird.

Dem Autor sind keine Arbeiten bekannt, die verdeckende Sprungkanten in Abstandsbildern von Laserradarscannern so detektieren, daß eine Kantenpunktverkettung möglich wäre. Die Schwierigkeit besteht darin, Bildpunkte auf verdeckenden Kanten von solchen zu unterscheiden, die durch den Mixed-Point Effekt verfälscht sind. In dieser Arbeit wird zur Unterscheidung verdeckender von verdeckten Kanten und zur Lokalisierung verdeckender Kanten ein Krümmungsvorzeichenwechsel herangezogen. Da in der Kantendetektion in [Fan 90] ebenfalls Krümmungsvorzeichenwechsel verwendet werden und sich der hier verfolgte Ansatz daran anlehnt, wird das Verfahren von Fan hier genauer dargestellt.

Die Vorgehensweise des einfachsten der drei Verfahren erfolgt in drei Schritten: (1) Gauß'sche Glättung des Abstandsbildes, (2) Ableitungs- und Krümmungsberechnung und (3) Markierung von Extrema und Nulldurchgängen. Die Berechnung benötigt zweite Ableitungen der Bildfunktion. Da die zweite Ableitung sehr rauschempfindlich ist, wird das Bild vor der Krümmungsberechnung geglättet. Die Glättung wird durch eine Faltung des Bildes mit einer zweidimensionalen Gaußfunktion bestimmter Parametrisierung erreicht. Im zweiten Schritt werden pro Bildpunkt die ersten und zweiten Ableitungen in vier diskreten Richtungen von  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  und  $135^\circ$  geschätzt. Aus der ersten und zweiten Ableitung jeder diskreten Richtung wird dann die Krümmung für die entsprechende Richtung berechnet. Im drit-

ten Schritt werden wieder jeweils für jede diskrete Richtung getrennt Dachkanten als Krümmungsextrema und Sprungkanten als Nulldurchgänge detektiert. Extrema sind Krümmungswerte oberhalb eines bestimmten Schwellwertes. Ein Bildpunkt wird hingegen als Nulldurchgang markiert, wenn seine Krümmung nahezu Null ist, die Nachbarpunkte in der gegebenen Richtung ungleich Null sind und entgegengesetztes Vorzeichen besitzen.

Da als Merkmal einer Sprungkante ein Nulldurchgang bzw. Vorzeichenwechsel dient, können auch verschmierte und geglättete Sprungkantenpunkte relativ genau lokalisiert werden. Diese Eigenschaft ist bedeutend, da auch durch den Mixed-Point Effekt verfälschte Abtastpunkte genau wie geglättete Sprungkantenpunkte zwischen Vorder- und Hintergrund einer Sprungkante liegen. Es ist daher möglich, von benachbarten Punkten positiven bzw. negativen Krümmungsvorzeichens auf verdeckende bzw. verdeckte Sprungkantenpunkte zu schließen. Verdeckende und verdeckte Kantenpunkte sind also in Anwesenheit des Mixed-Point Effekts unterscheidbar.

In dem hier vorgeschlagenen Ansatz [Vogelgesang 94] soll daher der Vorzeichenwechsel ähnlich wie in [Fan 90] genutzt werden, um Sprungkantenpunkte zu detektieren. Anhand des Vorzeichenwechsels sollen darüberhinaus speziell verdeckende Sprungkantenpunkte derart lokalisiert werden, daß sie mit hoher Wahrscheinlichkeit nicht durch den Mixed-Point Effekt verfälscht sind. Dieses Verfahren wird in dem Abschnitt der Kantenpunktlokalisierung erläutert. Im Gegensatz zu [Fan 90] sollen Krümmungen allerdings nicht in diskreten Richtungen berechnet werden. Vielmehr sollen richtungsunabhängige zweidimensionale Krümmungen herangezogen werden. Auf diese Problematik geht der folgende Abschnitt ein.

## 3.2 Angepaßte Mittlere Krümmung

Fan verwendet Krümmungen der Bildfunktion in vier diskreten Richtungen. Hier sollen richtungsunabhängige Krümmungen untersucht werden. Dies vermeidet zum einen die getrennte Behandlung richtungsabhängiger Krümmungen und kann im Gegensatz zu wenigen Richtungen eine höhere Lokalisierungsgenauigkeit erzielen. Wie in [Fan 90] wird das Abstandsbild zunächst geglättet, damit im Verlauf der Krümmungsberechnung hochfrequente Anteile nicht übermäßig verstärkt werden.

Die Glättung besteht in einer Mittelung von Abstandswerten, die durch eine Faltung mit einer zweidimensionalen Gaußfunktion bewirkt werden kann. Die Gaußfunktion  $G(u, v, \sigma)$  mit der Varianz  $\sigma$  ist gegeben mit:

$$G(u, v, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Unter den zweidimensionalen Krümmungsmaßen besitzen insbesondere die Gauß'sche und die Mittlere Krümmung die erwünschte Eigenschaft der Blickpunktinvarianz. Die Blickpunktinvarianz bedeutet, daß für Punkte auf einer Fläche, die unter einer Blickpunktänderung sichtbar bleiben, sich die Krümmung nicht ändert. Speziell zur Detektion von Kanten in Abstandsbildern hat sich die Mittlere Krümmung als rauschunempfindlicher als die Gauß'sche Krümmung erwiesen [Besl 88b].

Die Mittlere Krümmung in einem Bildpunkt benötigt die ersten und zweiten partiellen Ableitungen in dem Bildpunkt. Jede Ableitung kann durch eine Faltung der Bildfunktion mit einer Faltungsmaske  $D_x, x \in \{u, v, uu, vv, uv\}$  bestimmter Größe berechnet werden. Die Berechnung der Faltungsmasken wird in Anhang B

näher erläutert.

$$\begin{aligned}r_u &= D_u \otimes r(u, v) \\r_v &= D_v \otimes r(u, v) \\r_{uu} &= D_{uu} \otimes r(u, v) \\r_{vv} &= D_{vv} \otimes r(u, v) \\r_{uv} &= D_{uv} \otimes r(u, v)\end{aligned}$$

Mit den fünf Ableitungen berechnet sich die Mittlere Krümmung  $H$  zu:

$$H = \frac{1}{2} \frac{(1 + r_v^2)r_{uu} + (1 + r_u^2)r_{vv} - 2r_u r_v r_{uv}}{(1 + r_u^2 + r_v^2)^{\frac{3}{2}}}$$

Krümmungen mit hohem Betrag werden anhand der Funktion  $sgn(H, \epsilon_+, \epsilon_-)$  ermittelt, wobei  $\epsilon_+$  und  $\epsilon_-$  Schwellwerte darstellen:

$$sgn(H, \epsilon_+, \epsilon_-) = \begin{cases} +, & H > \epsilon_+ \\ 0, & \epsilon_- \leq H \leq \epsilon_+ \\ -, & H < \epsilon_- \end{cases}$$

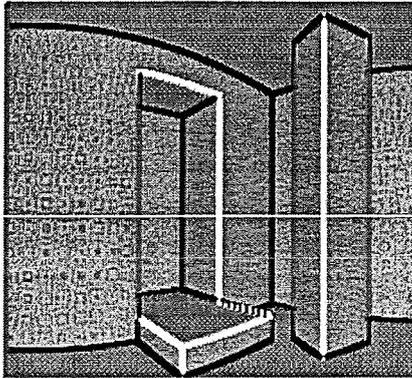
Die Vorzeichen als Werte der Funktion werden in Krümmungsvorzeichenbildern nach folgender Grauwertkodierung abgebildet:

$sgn(H, \epsilon_+, \epsilon_-)$	Grauwert
0	grau
+	weiß
-	schwarz

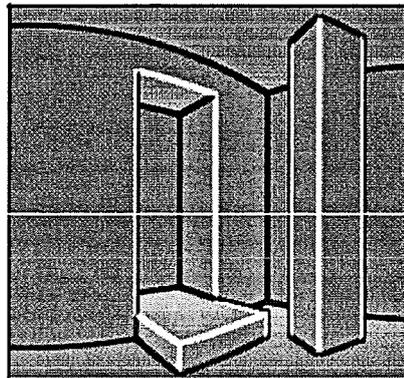
Zur Krümmungsberechnung hat sich folgende Heuristik bewährt. Werden die Ableitungen vor der Krümmungsberechnung alle um einen Faktor  $c > 0$  skaliert, läßt sich die Position der Krümmungsextrema verschieben, ohne die Qualität der Krümmungsfunktion selbst signifikant zu verändern. Durch Skalierung mit  $c_1 = 0.001$  werden die Krümmungsextrema so verschoben, daß sie in das Abtastraster fallen, während sie durch den größeren Faktor  $c_2 = 0.1$  nicht in das Abtastraster fallen. Durch den Faktor läßt sich also steuern, ob Sprungkantenpunkte detektiert oder nicht detektiert werden sollen. Die Ableitungen werden also vor der Krümmungsberechnung mit dem Faktor  $c$  multipliziert:

$$\begin{aligned}r_u &\leftarrow c \cdot r_u \\r_v &\leftarrow c \cdot r_v \\r_{uu} &\leftarrow c \cdot r_{uu} \\r_{vv} &\leftarrow c \cdot r_{vv} \\r_{uv} &\leftarrow c \cdot r_{uv}\end{aligned}$$

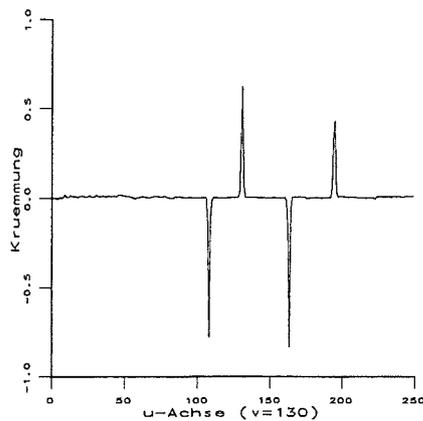
Eine quantitative analytische Untersuchung des Zusammenhangs zwischen Abstandskodierung, Rasterung und Krümmungsnulldurchgang wäre wünschenswert, wurde hier aber nicht durchgeführt. Die angepaßte Mittlere Krümmung mit  $c = 0.1$  und  $c = 0.001$ , angewendet auf das synthetische Abstandsbild zeigt Abb. 3.1. Die Krümmungen des nicht-geglätteten Abstandsbildes wurden anhand von  $5 \times 5$ -Faltungsmasken berechnet.



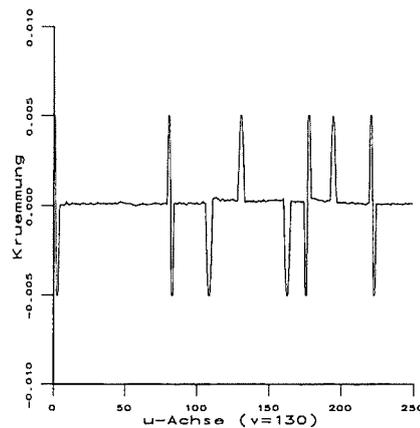
(a) Angepaßte Mittlere Krümmung mit  $c = 0.1$ . Entlang der eingeblendeten horizontalen Linie ist der Krümmungsverlauf in Teilabb. 3.1c abgebildet.



(b) Angepaßte Mittlere Krümmung mit  $c = 0.001$ . Entlang der eingeblendeten horizontalen Linie ist der Krümmungsverlauf in Teilabb. 3.1d abgebildet.



(c) Angepaßte Mittlere Krümmung entlang der horizontalen Linie in Teilabb. 3.1a mit Faktor  $c = 0.1$ , Sprungkanten werden *nicht* detektiert



(d) Angepaßte Mittlere Krümmung entlang der horizontalen Linie in Teilabb. 3.1b mit Faktor  $c = 0.001$ , Sprungkanten werden als *Vorzeichenwechsel* detektiert.

Abbildung 3.1: Mittlere und Angepaßte Mittlere Krümmung: die Anpassung der Mittleren Krümmung besteht darin, daß die Ableitungen vor der Krümmungsbeziehung durch einem Faktor skaliert werden.

Das Ziel der Krümmungsberechnung besteht darin, Sprungkanten in Krümmungsbildern durch einen Krümmungsvorzeichenwechsel zu detektieren. Bei positiver Krümmung kann sowohl ein konvexer als auch ein verdeckender Kantenpunkt, und bei negativer Krümmung sowohl ein konkaver als auch ein verdeckter Kantenpunkt vorliegen. Um diese Unterscheidung dennoch treffen zu können hat sich folgende Strategie als erfolgreich erwiesen. Konvexe Kantenpunkte werden durch geeignete Wahl des Parameters  $c$  und der Schwellwerte  $\epsilon_+$ ,  $\epsilon_-$  getrennt detektiert, so daß nur noch verdeckte und konkave Kanten unterschieden werden müssen. Zu diesem Zweck kann auch der Faktor  $c$  herangezogen werden.

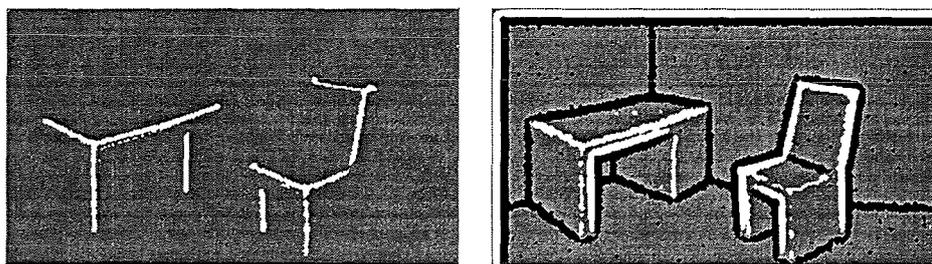
Für beide Bilder müssen also jeweils folgende Parameter festgelegt werden: (1) der Parameter  $\sigma$  der Gaußfunktion und die Größe  $n_G$  der Gaußfaltungsmaske, (2) die Fenstergröße  $n_H$  der Faltungsmasken für die Ableitungsberechnung, (3) der Faktor  $c$  für die Ableitungsskalierung und (4) die Schwellwerte  $\epsilon_+$ ,  $\epsilon_-$  für die Funktion  $\text{sgn}$ . Bei realen Abstandsbildern haben sich für die beiden Krümmungsbilder folgende Parameter experimentell als erfolgreich erwiesen:

Parameter	Krümmungsvorzeichenbild 1 für <i>konvexe</i> Kantenpunkte	Krümmungsvorzeichenbild 2 für <i>verdeckende und konkave</i> Kantenpunkte
$n_H$	11	9
$n_G$	3	11
$\sigma$	1.0	3.0
$c$	0.1	0.001
$\epsilon_+$	0.10	0.0015
$\epsilon_-$	$-\infty$	-0.0006

Die ermittelten Parameter werden im folgenden interpretiert und bzgl. ihrer Empfindlichkeit bewertet. Für die Krümmungsberechnung konvexer Dachkanten (Krümmungsbild 1) ist eine höhere Fenstergröße erforderlich, da planare Szenenflächen in der Abstandsbildfunktion aufgrund der ungleichförmigen Abtastung leicht gekrümmt sind. Da die Krümmungsberechnung innerhalb eines relativ großen  $11 \times 11$ -Fensters selbst glättende Wirkung besitzt, ist auch eine nur relativ geringe Gaußsche Glättung erforderlich. Damit konvexe Krümmungen, die von verdeckenden Kanten herrühren, nicht detektiert werden, wird  $c$  mit  $c = 0.1$  relativ groß gewählt.

Um den Krümmungsvorzeichenwechsel und konkave Kantenpunkte zu detektieren (Krümmungsbild 2), waren für die realen Abstandsbilder der Testszene  $9 \times 9$ -Fenster mit relativ kleinen Werten für  $c$  mit  $c = 0.001$  ausreichend. Die Krümmungsvorzeichenbilder 1 und 2 der ersten Testszene sind in Abb. 3.2a und 3.2b abgebildet. Die Parameter  $n_H$ ,  $n_G$ ,  $\sigma$ ,  $c$  sind in beiden Bildern relativ unempfindlich, auch leicht abweichende Werte liefern zufriedenstellende Ergebnisse. Die Schwellwerte  $\epsilon_+$ ,  $\epsilon_-$  sind empfindlicher. Ist  $\epsilon_+$  zu hoch eingestellt, werden in Krümmungsbild 1 nicht alle konvexen Kantenpunkte ermittelt oder die resultierenden Kantenpunkte bilden keine zusammenhängenden Bildpunktgruppen aus, so daß eine spätere Bildpunktverkettung nicht direkt möglich ist.

Im Falle von Krümmungsbild 2 ist eine 8-Nachbarschaft zwischen Bildpunkten positiver und negativer Krümmung erwünscht, um mittels lokaler Operatoren einen Vorzeichenwechsel lokalisieren zu können. Ist nun  $\epsilon_+$  zu groß oder  $\epsilon_-$  zu klein, läßt sich diese Nachbarschaftsbeziehung nicht direkt herstellen. Eine bessere Wahl der Schwellwerte könnte ein dynamisches Schwellwertverfahren ermöglichen, das anhand von Informationen über die Abstandskodierung und die Rasterung den Schwellwert von Bildpunkt zu Bildpunkt geeignet anpaßt.



(a) Krümmungsbild 1 für die Extraktion konvexer Kantenpunkte

(b) Krümmungsbild 2 für die Extraktion konkaver und verdeckender Kantenpunkte

Abbildung 3.2: Krümmungsbilder des Abstandsbildes der ersten Testszene: aus Krümmungsbild 1 werden konvexe Kantenpunkte direkt durch eine Schwellwertoperation ermittelt. Aus dem Krümmungsbild 2 werden konkave und verdeckende Kanten in der Phase der Kantenpunktlokalisierung unterschieden.

### 3.3 Lokalisierung von Bildkantenpunkten

#### 3.3.1 Lokalisierungsstrategie

Die Erläuterung der weiteren Bildauswertung erfordert die Definition folgender Begriffe. (1) Eine *Region* stellt eine zusammenhängende Teilmenge der Bildpunkte des digitisierten Bildes dar. (2) Repräsentiert man jeden Bildpunkt als Quadrat innerhalb eines quadratischen Gitters, besitzt jedes quadratische Bildelement einen gemeinsamen Rand mit vier benachbarten Bildelementen und teilt eine Ecke des Quadrates mit vier weiteren Bildelementen. Zwei Bildelemente heißen nun *4-Nachbarn*, wenn sie einen gemeinsamen Rand bzw. zumindest eine gemeinsame Ecke besitzen. (3) Ein *8-Pfad* zwischen dem Bildelement an der Stelle  $[u_0, v_0]$  und dem Bildelement in  $[u_n, v_n]$  ist eine Folge von Bildelementindizes  $([u_0, v_0], \dots, [u_n, v_n])$ , so daß jedes Bildelement in  $[u_k, v_k]$  ein 8-Nachbar von Bildelement in  $[u_{k+1}, v_{k+1}]$  für alle  $0 \leq k \leq n - 1$  ist. (4) Eine *Region*  $R$  heißt nun *8-zusammenhängend*, wenn zwischen allen Bildelementen ein 8-Pfad existiert. (5) Der *Rand* der Region  $R$  ist die Menge aller Bildelemente von  $R$ , die 4-Nachbarn in  $\bar{R}$  besitzen. (6) Eine *Verdünnungsoperation* löscht alle Bildelemente auf dem Rand einer ohne Beschränkung der Allgemeinheit zusammenhängenden Region, die den 8-Zusammenhang der Region nicht auflöst. (7) Die schrittweise Anwendung der Verdünnungsoperation reduziert die Region auf das *Skelett* der Region.

Da Krümmungen von bereits geglätteten Bildern berechnet werden und die Krümmungsvorzeichen durch eine Schwellwertoperation bestimmt werden, bilden die Krümmungsvorzeichen breite Regionen im Bild aus. Die Kantenpunktregionen sollen nun so lokalisiert werden, daß sie folgenden Zielformulierungen genügen. In nachfolgender Zielformulierung 3.1 kommt zum Ausdruck, daß die entsprechenden Bildregionen Strichbreite aufweisen sollen, damit eine exakte Position der Kanten ermittelt und eine eindeutige Verkettung von Bildkantenpunkten ermöglicht wird.

**Zielformulierung 3.1 (Bildkantenpunktregionen)** *Konvexe, konkave und verdeckende Kantenpunkte sollen so lokalisiert werden, daß sie unter der Annahme einer 8-Nachbarschaft zu Punktketten verknüpft werden können.*

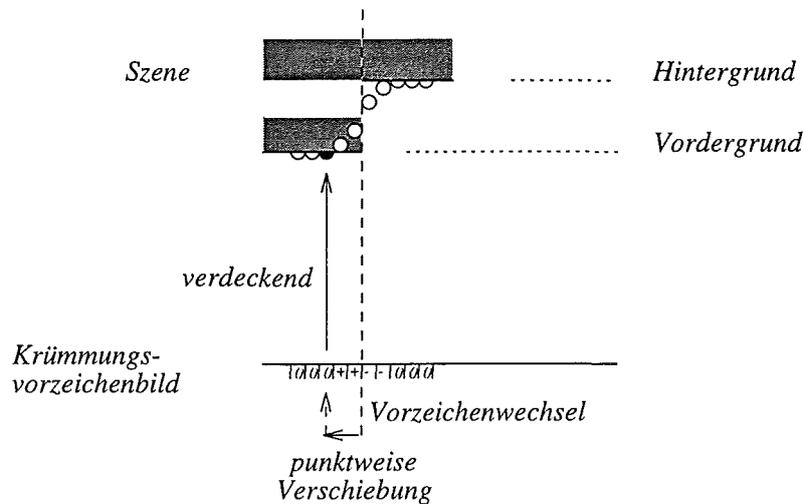


Abbildung 3.3: Strategie zur Lokalisierung verdeckender Punkte anhand des Krümmungsvorzeichenwechsels: Die Position des Krümmungsvorzeichenwechsels lokalisiert den Ort der Sprungkante. Verschiebt man diesen Ort ausgehend von dem Ort des Vorzeichenwechsels in Richtung der positiven Vorzeichen, erhält man einen Bildpunkt, dessen entsprechender Szenenpunkt mit hoher Wahrscheinlichkeit nicht durch den Mixed-Point Effekt verfälscht ist.

**Zielformulierung 3.2 (Verdeckende Bildkantenpunkte)** *Verdeckende Bildkantenpunkte sollen so lokalisiert werden, daß sie mit hoher Wahrscheinlichkeit nicht durch den Mixed-Point Effekt verfälscht sind.*

Verdeckende Punkte in Zielformulierung 3.2 sind solche Punkte, die an dem Vorzeichenwechsel beteiligt sind und selbst positives Vorzeichen besitzen. Verdeckende Punkte sind allerdings mit hoher Wahrscheinlichkeit durch dem Mixed-Point Effekt verfälscht. Das heißt, sie liegen nicht auf Objektflächen im Vordergrund, sondern haben einen nicht-definierten Abstand zwischen Vorder- und Hintergrund. Durch den Mixed-Point Effekt verfälschte Bildpunkte sollen daher nicht als verdeckende Bildkantenpunkte in Betracht kommen.

Verdeckende Punkte sollen neben Zielformulierung 3.2 auch Zielformulierung 3.1 genügen, das heißt, daß sie als zusammenhängende Bildpunktmenge eine Kantenpunktverkettung erlauben. Nach der Idee in [Hebert & Krotkov 92], verdeckende Kantenpunkte durch Entfernen isolierter 3D-Szenenpunkte zu identifizieren, ist hingegen nicht garantiert, daß verdeckende Kantenpunkte als 2D-Bildpunkte verkettet werden können. Ausgangspunkt für eine Strategie zur Lokalisierung verdeckender Bildkantenpunkte ist folgende Überlegung. Da bekannt ist, daß eine Sprungkante Teil einer polyedrischen Fläche ist, liegen verdeckende Punkte, die im Vorzeichenbild einen bestimmten Betrag von der Position des Vorzeichenwechsels entfernt liegen, ebenfalls auf der Polyederfläche und nahe der betrachteten Sprungkante. Wie nahe verdeckende Punkte an einem Vorzeichenwechsel liegen können, ohne bereits durch den Mixed-Point Effekt beeinflusst zu werden, hängt vor allem von der Größe des Laserstrahlfußabdrucks und der Feinheit der gewählten Abtastung ab. Die in dieser Arbeit verfolgte Strategie besteht darin, ausgehend von der Position des Vorzeichenwechsels verdeckende Bildkantenpunkte durch eine punktweise Verschiebung zu lokalisieren (siehe Abb. 3.3). Im Zusammenhang mit der Abtastung treten bei dieser Strategie allerdings zwei Probleme auf.

Die lokalisierten Kantenpunkte liegen zwar nahe der Sprungkante. Jedoch weist

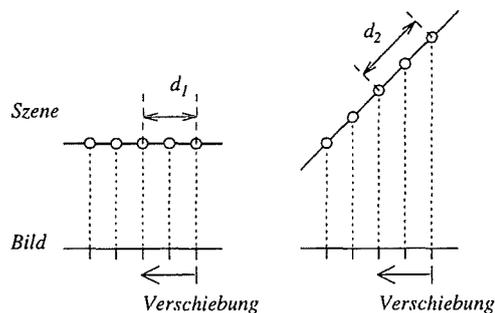


Abbildung 3.4: Positionierungsfehler von lokalisierten verdeckenden Sprungkantenpunkten: eine Verschiebung um  $n$  Bildpunkte (hier:  $n = 2$ ), erzeugt bei schräger Aufsicht einen größeren Fehler in Szenenkoordinaten (im Sinne des euklidischen Abstandes).

die lokalisierte 3D-Position gegenüber der tatsächlichen 3D-Position der verdeckenden Kante einen Fehler auf, der nicht nur von dem Betrag der Verschiebung im Bild abhängig ist. Der Fehler ist auch von der Neigung der Fläche in Bezug zur Blickrichtung des Sensors abhängig. Je spitzer die Aufsicht auf eine Szenenfläche ist, desto weniger Abtastpunkte liegen auf einer Flächeneinheit. Im Gegensatz zu einem Positionierungsfehler in dem Abstandsbild einer senkrechten Flächenaufsicht bewirkt ein Positionierungsfehler im Abstandsbild einer schrägen Flächenaufsicht einen größeren Positionierungsfehler in Szenenkoordinaten (siehe Abb. 3.4). Folglich müssen bei der Erkennung von Objekten anhand der Länge von 3D-Szenenkanten bestimmte Toleranzgrenzen zugelassen sein.

Das zweite Problem besteht darin, daß die lokalisierten verdeckenden Kantenpunkte nur dann auf der Polyederfläche der angrenzenden Szenenkante liegen, wenn eine Mindestanzahl an Abtastpunkten in Richtung der Verschiebung auf der Polyederfläche liegen. Ist dies nicht der Fall, liegen lokalisierte Punkte dann auf einer benachbarten Polyederfläche. Wenn zwischen dem Vorzeichenwechsel zweier Sprungkanten weniger als eine Mindestanzahl an Bildpunkten liegen, ist eine Lokalisierung verdeckender Sprungkanten nach dieser Strategie nicht möglich. In diesem Fall können regionenbasierte Verfahren helfen, um mit dem Regionenrand die erwünschten Kanten zu erhalten. Folgender Algorithmus genügt den Zielformulierungen 3.1 und 3.2:

**Algorithmus 3.1 (Lokalisierung von Bildkantenpunkten)** Eingabe sind die Krümmungsvorzeichenbilder 1 und 2. Ausgabe sind die Binärbilder  $e_1(u, v)$  für lokalisierte verdeckende Bildkantenpunkte,  $e_2(u, v)$  für lokalisierte konkave Bildkantenpunkte und  $e_3(u, v)$  für lokalisierte konvexe Bildkantenpunkte. Führe der Reihe nach folgende Anweisungen aus:

1. Eingabe: Krümmungsbild  $1 = a(u, v)$  (siehe Abb. 3.2a)  
Ausgabe: Krümmungsvorzeichenbild  $b(u, v)$  ohne Nulldurchgänge  
Anweisung: Führe Algorithmus in Abschnitt „Löschen von Nulldurchgängen“ aus.
2. Eingabe: Bild  $b(u, v)$ , Anzahl  $max$  an punktwisen Verschiebungen  
Ausgabe: Krümmungsvorzeichenbild  $c_{max}(u, v)$ , Binärbild  $e_1(u, v)$   
Anweisung: Führe Algorithmus in Abschnitt „Lokalisierung verdeckender Bildkantenpunkte“ aus.

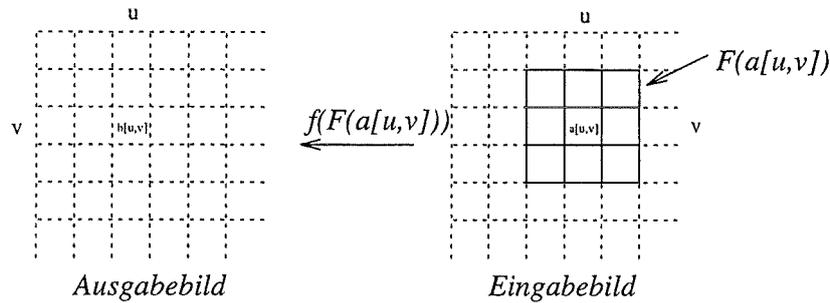


Abbildung 3.5: Wirkungsweise von Nachbarschaftsoperatoren: Ein Nachbarschaftsoperator überführt ein Eingabebild  $a$  in ein Ausgabebild  $b$  gleicher Auflösung. Dazu wird ein Bildfenster bestimmter Größe spalten- und zeilenweise über das Eingabebild geschoben. Alle Werte innerhalb des Fensters mit dem Zentrum  $a[u,v]$  werden herangezogen, um gemäß der Funktion  $f$  einen Ausgabewert zu berechnen. Der Ausgabewert  $b[u,v]$  wird an der gleichen Stelle  $[u,v]$  des Ausgabebildes  $b$  eingetragen.

3. *Eingabe:* Bild  $c_{max}(u,v)$   
*Ausgabe:* Binärbild  $e_2(u,v)$   
*Anweisung:* Führe Algorithmus in Abschnitt „Lokalisierung konkaver Bildkantenpunkte“ aus.
4. *Eingabe:* Krümmungsbild 1 (siehe Abb. 3.2b)  
*Ausgabe:* Binärbild  $e_3(u,v)$   
*Anweisung:* Führe Algorithmus in Abschnitt „Lokalisierung konvexer Bildkantenpunkte“ aus.
5. Gebe folgende Binärbilder aus:  $e_1(u,v), e_2(u,v), e_3(u,v)$ .

### 3.3.2 Lokalisierungsverfahren

Zur Lokalisierung der Kantenpunkte wird die Technik der Nachbarschaftsoperatoren [Haralick & Shapiro 93a] verwendet. Die Wirkung eines Nachbarschaftsoperators wird in Abb. 3.5 verdeutlicht. Da auf den betrachteten Bildern keine numerischen Berechnungen, sondern symbolische Operationen durchgeführt werden, liegen symbolische Nachbarschaftsoperatoren vor. Die zeilen- und spaltenweise Verschiebung des Fensterzentrums über das Eingabebild kann man durch zwei geschachtelte Schleifen beschreiben. Die Berechnung des Ausgabewertes  $b$  als Funktion  $f$  der Bildpunkte des Fensters  $F(a[u,v])$  an der Position  $[u,v]$  des Eingabebildes  $a$  wird im Schleifenrumpf vorgenommen. Zur kürzeren Notation von Nachbarschaftsoperatoren wird eine Notation in Anlehnung an [Davies 90] verwendet.

$$[[b[u,v] \leftarrow f(F(a[u,v]))]]_{u,v} \equiv \left[ \begin{array}{l} \text{for } u \leftarrow 1 \text{ to } u_{max} \text{ do} \\ \quad \text{for } v \leftarrow 1 \text{ to } v_{max} \text{ do} \\ \quad \quad b[u,v] \leftarrow f(F(a[u,v])); \\ \quad \text{od;} \\ \text{od;} \end{array} \right]$$

Die symbolischen Nachbarschaftsoperatoren arbeiten auf mehrwertigen Bildern, wobei hier jeder Bildpunkt eines von sechs Symbolen als Wert besitzen kann. Die Symbole selbst sind in Bildern durch verschiedene Grauwerte kodiert. Insbesondere die „ $\pm_i$ “-Symbole dienen zur Lokalisierung verdeckender Kantenpunkte. In Bildern werden sie einheitlich als hellgrau markiert.

Wert	Bedeutung	Grauwert
0	Krümmung gleich 0	grau
+	positive Krümmung	weiß
-	negative Krümmung	schwarz
$\pm_0$	positive Krümmung mit „-“-Nachbar	hellgrau
$\pm_1$	„0“-Punkt oder „-“-Punkt mit „ $\pm_0$ “-Nachbar	hellgrau
$\pm_2$	„0“-Punkt oder „-“-Punkt mit „ $\pm_1$ “-Nachbar	hellgrau

Mit Hilfe von Nachbarschaftsoperatoren werden im folgenden die vier oben genannten Schritte beschrieben.

### Löschen von Nulldurchgängen

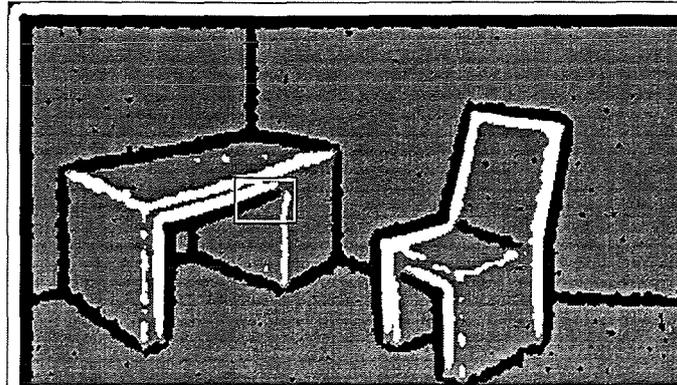
Da Krümmungen von geglätteten Bildern berechnet werden, ist der Vorzeichenwechsel entlang von Sprungkanten nicht ideal. Vielmehr befinden sich zwischen Bildpunkten entgegengesetzten Vorzeichens einzelne Punkte der Krümmung Null. Da Sprungkanten durch die Gauß'sche Glättung abgeflacht werden, wird die Bildfunktion in den entsprechenden Bildpunkten als lokal flach, d.h. mit Krümmung 0, klassifiziert. Um in einem mehrwertigen Bild einen perfekten Vorzeichenwechsel herzustellen, werden solche Punkte mit dem Wert „0“ auf „-“ gesetzt, die innerhalb eines  $3 \times 3$ -Fensters von einer gleich hohen Anzahl von „-“ und „+“-Punkten umgeben sind. Zur Notation eines Nachbarschaftsoperators sei die Funktion  $Anzahl(F(a[u, v]), w)$  gegeben, die die Anzahl der Bildpunkte  $F(a[u, v])$  innerhalb eines  $3 \times 3$ -Fensters von Bild  $a$  angibt, die den Wert  $w$  besitzen. Damit lautet der Nachbarschaftsoperator:

```
[ (* Löschen von Nulldurchgängen *)
  if a[u, v] = 0 then
    p ← Anzahl(F(a[u, v]), +)
    n ← Anzahl(F(a[u, v]), -)
    if p ≥ 3 and n ≥ 3 and p + n ≥ 7 then
      b[u, v] ← -
    else
      b[u, v] ← a[u, v]
  fi;
else
  b[u, v] ← a[u, v]
fi;
]u,v
```

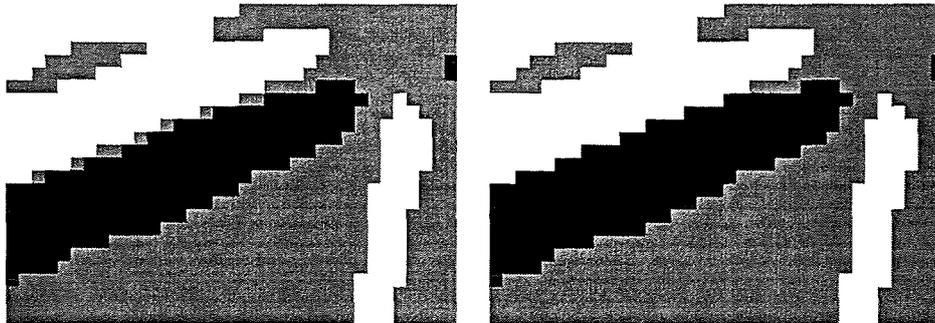
Die Wirkung des Nachbarschaftsoperators wird in Abb. 3.6 veranschaulicht. Von Krümmungsbild 2 wird ein Teil der unteren Tischkante vergrößert, die eine Sprungkante darstellt. Zwischen weißen und schwarzen Punkten, die verdeckende und verdeckte Sprungkanten darstellen, sind einzelne graue bzw. lokal flache Punkte eingeschlossen. Nach Anwendung des Operators ist ein Großteil der eingeschlossenen Bildpunkte auf „-“ gesetzt.

### Lokalisierung verdeckender Bildkantenpunkte

Durch das Löschen von Nulldurchgängen sind Punkte entgegengesetzten Vorzeichens direkt benachbart. Die direkte Nachbarschaft kann jetzt genutzt werden, um verdeckende Kantenpunkte zu lokalisieren. Die Lokalisierung verdeckender Kantenpunkte läuft in zwei Schritten ab. In einem ersten Schritt werden alle „+“-Punkte,



(a) Krümmungsbild 2 mit markiertem Bildausschnitt



(b) Nulldurchgänge erscheinen als lokal flache (graue) Punkte an der Stelle des Vorzeichenwechsels

(c) Fast alle Nulldurchgänge sind durch den Nachbarschaftsoperator gelöscht.

Abbildung 3.6: Löschen von Nulldurchgängen in Krümmungsvorzeichenbildern: der Vorzeichenwechsel an Sprungkanten ist durch Nulldurchgänge gestört (Teilabb. (b)). Durch den Nachbarschaftsoperator werden die Nulldurchgänge zu Punkten negativer Krümmung aufgefüllt (Teilabb. (c)).

die einen „-“-Nachbar besitzen, als „ $\pm_0$ “ markiert. Das heißt, „ $\pm_0$ “-Punkte sind solche Punkte, die an dem Vorzeichenwechsel beteiligt sind und positive Krümmung besitzen. In Abwesenheit eines Mixed-Point Effektes können „ $\pm_0$ “-Punkte bereits als verdeckende Kantenpunkte interpretiert werden. Die „ $\pm_0$ “-Punkte werden durch folgenden Operator bestimmt:

```
[ (* Vorzeichenwechsel *)
  if  $b[u, v] = +$  and  $Anzahl(F(b[u, v]), -) \geq 1$  then
     $c_0[u, v] \leftarrow \pm_0$ 
  else
     $c_0[u, v] \leftarrow b[u, v]$ 
  fi;
]u, v
```

Ausgehend von den „ $\pm_0$ “-Punkten werden schrittweise „ $\pm_i$ “-Punkte bestimmt, die von dem Vorzeichenwechsel um den Betrag von  $i$  Bildpunkten entfernt liegen. Dies wird erreicht, indem schrittweise „ $\pm_i$ “-Punkte bestimmt werden, die „ $\pm_{(i-1)}$ “-Punkte als Nachbarn besitzen. „ $\pm_{max}$ “-Punkte werden als endgültige verdeckende Kantenpunkte in Form eines Binärbildes  $e_1(u, v)$  verwendet. Der entsprechende Nachbarschaftsoperator hat folgende Gestalt:

```

[ (* i-te Verschiebung, i>=1 *)
  if  $c_{i-1}[u, v] \in \{0, -\}$  and  $Anzahl(F(c_{i-1}[u, v]), \pm_{i-1}) \geq 1$  then
     $c_i[u, v] \leftarrow \pm_i$ ;
    if  $i = max$  then  $e_1[u, v] \leftarrow 1$ ;
  else
     $c_i[u, v] \leftarrow c_{i-1}[u, v]$ ;
    if  $i = max$  then  $e_1[u, v] \leftarrow 0$ ;
  fi;
]u,v

```

Das Ergebnis der Anwendung des Operators für  $i = 1$  und nachfolgende Anwendung für  $i = 2$  zeigt Abb. 3.7.

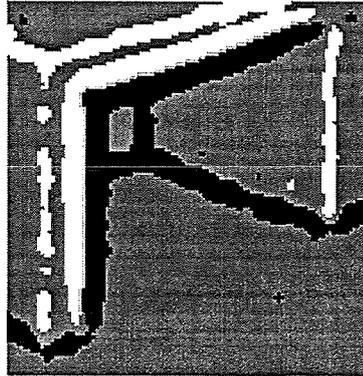
### Lokalisierung konkaver Bildkantenpunkte

In dem Krümmungsvorzeichenbild 2 besitzen sowohl verdeckte als auch konkave Kantenpunkte gleiches Krümmungsvorzeichen und sind daher beide als „-“-Punkte repräsentiert. Das einzige Unterscheidungsmerkmal der beiden Kantentypen besteht darin, daß verdeckte Punkte zu verdeckenden Punkten direkt benachbart sind. Da die Krümmungsvorzeichen von geglätteten Bildern berechnet sind, bilden verdeckte Kantenpunkte flächige Regionen aus. Um die direkte Nachbarschaft als Unterscheidungsmerkmal für konkave und verdeckte Kantenpunkte heranziehen zu können, werden alle „-“-Punkte durch ein Verdünnungsverfahren nur dann auf „0“ gesetzt, falls keine direkte Nachbarschaft zu „ $\pm_0$ “-Punkten besteht. Nach dieser bedingten Verdünnung liegt folgendes Ergebnis vor (siehe Teilabb. 3.8b):

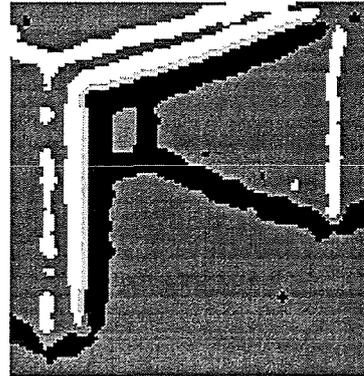
- Alle „-“-Punkte, die keinen „ $\pm_0$ “-Nachbar besitzen, können als konkav interpretiert werden.
- Flächige Kantengebiete, die von konkaven Kanten herrühren, sind auf Regionen mit Strichbreite verdünnt.

Durch die mehrfache Durchführung einer Verdünnungsoperation werden die „-“-Gebiete in Skelette überführt. Skelettpunkte zeichnen sich dadurch aus, daß sie einen maximalen Abstand zu den Rändern der ursprünglichen Region besitzen. Unter der Annahme, daß Punkte maximaler Krümmung in der Regionenmitte liegen, liefert die Verdünnung die korrekten konkaven Kantenpunkte. Zur Verdünnung wird hier das Verfahren von [Chen & Hsu 88] verwendet.

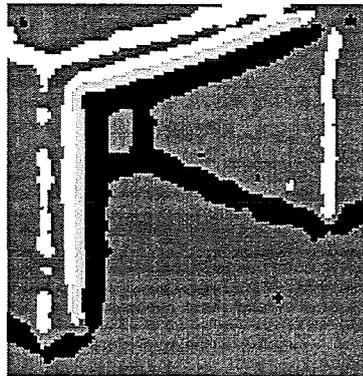
Dieses Verfahren bietet unter Berücksichtigung des vorhandenen Szenenwissens folgende Vorteile: (1) Skelettlinien werden gegenüber der ursprünglichen Region von ihrem Endpunkt her im Gegensatz zu anderen Verdünnungsverfahren nur wenig verkürzt. Dies bedeutet, daß Endpunkte in dem Binärskelett mit hoher Wahrscheinlichkeit nahe an Vertizes liegen. (2) Wird der Rand der ursprünglichen Region um einzelne Punkte erweitert, bleibt die Position der Skelettpunkte stabil. Dienen solche Skelettpunkte in nachgelagerten Verarbeitungsschritten zur Approximation



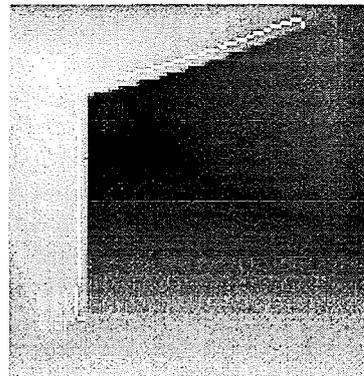
(a) Lokalisierung des Vorzeichenwechsels als hellgrau markierte „ $\pm_0$ “-Punkte



(b) Zusätzlich einmal verschobene „ $\pm_1$ “-Punkte



(c) Zusätzlich zweimal verschobene „ $\pm_2$ “-Punkte



(d) Lokalisierte verdeckende Sprungkantenpunkte im Abstandsbild

Abbildung 3.7: Lokalisierung verdeckender Sprungkantenpunkte: durch Anwendung von Nachbarschaftsoperatoren werden schrittweise verdeckende Punkte lokalisiert, die einen bestimmten Betrag von der Position des Vorzeichenwechsels und damit von dem Rand der verdeckenden Kante entfernt liegen.

von Kanten, so ist auch die Lage der approximierten Kanten entsprechend stabil. (3) Repräsentiert das ursprüngliche Gebiet zwei sich kreuzende Linien, resultiert daraus im Skelett ein einziger Kreuzungspunkt und nicht wie bei anderen Verfahren zwei verschiedene Kreuzungspunkte, die das Skelett an dieser Stelle verzerren. Dies bedeutet, daß Kreuzungspunkte im Binärskelett Hinweise auf Vertizes geben, an denen mehr als zwei Kanten zusammentreffen.

In einem zweiten Schritt ist es nun möglich, alle Sprungkantenpunkte, also alle verdeckten und verdeckenden Kantenpunkte abzuziehen, so daß die verbleibenden „-“-Punkte physikalisch als konkave Kantenpunkte interpretiert werden können. Sie werden im folgenden als Binärbild repräsentiert. Der zugehörige Nachbarschaftsoperator läßt sich wie folgt schreiben:

```

[  d' ← c2[u, v];
   (* verdeckende Punkte abziehen *)
   if c2[u, v] ∈ {±0, ..., ±max} then d' ← 0
   (* verdeckte Punkte abziehen *)
   if c2[u, v] = - and ±0 ∈ F(c2[u, v]) then d' ← 0
   (* konkave Kantenpunkte in Binärbild e2 setzen *)
   if d' = - then
     e2[u, v] ← 1;
   else
     e2[u, v] ← 0;
   fi;
   (* Restbild d ohne konkave Kantenpunkte *)
   d[u, v] ← d';
]u,v

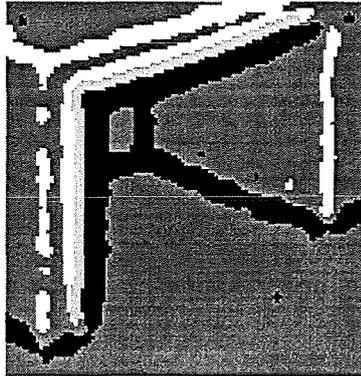
```

In Abb. 3.8c ist die Wirkung des Nachbarschaftsoperators anhand des Restbildes  $d(u, v)$  verdeutlicht. In Teilabb. (d) sind die verbleibenden konkaven Kantenpunkte des Binärbildes  $e_2(u, v)$  in dem Abstandsbild überlagert. Danach werden isoliert auftretende Punkte mittels eines Nachbarschaftsoperators wie in [Davies 90, p. 29f] entfernt. Um zu überprüfen, ob die extrahierten verdeckenden Punkte auf Objektflächen liegen und nicht aufgrund des Mixed-Point Effektes zwischen Vorder- und Hintergrund liegen, wurden die 2D-Bildpunkte in 3D-Szenenpunkte umgerechnet und in ihrer perspektivischen Projektion visualisiert (Abb. 3.10).

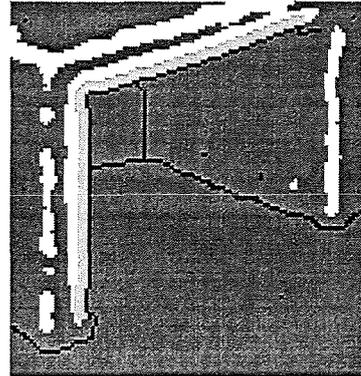
### Lokalisierung konvexer Bildkantenpunkte

Zur Lokalisierung konvexer Kantenpunkte wird das Krümmungsvorzeichenbild 1 direkt in ein Binärbild überführt, indem für jeden Bildpunkt des Krümmungsvorzeichenbildes 1 (siehe Abb. 3.2(a)) mit der Marke „0“ eine 0 und für jeden Bildpunkt des Krümmungsvorzeichenbildes 1 mit der Marke „+“ eine 1 im Binärbild  $e_3(u, v)$  gesetzt wird. Da konvexe Kantenpunkte noch nicht verdünnt sind, wird das betreffende Binärbild in einem zweiten Schritt ebenfalls nach dem Verfahren von [Chen & Hsu 88] verdünnt, wiederum unter der Annahme, daß Punkte maximaler Krümmung in der Regionenmitte liegen.

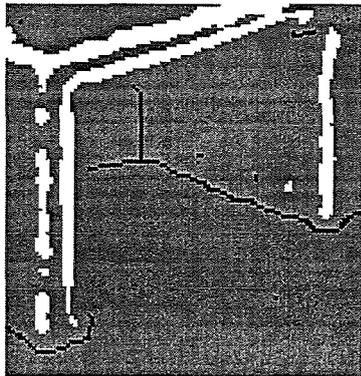
Verdünnungsverfahren können Bildregionen, in denen mehrere Kanten in spitzem Winkel zusammentreffen, stark verformen und damit den idealen Treffpunkt nicht exakt lokalisieren. Daher ist es wichtig, daß konvexe Kantenpunkte in einem eigenen Binärbild in Abwesenheit der übrigen Kantentypen verdünnt werden. Die Binärskelette für konvexe, konkave und verdeckende Kantenpunkte zeigt Abb. 3.9.



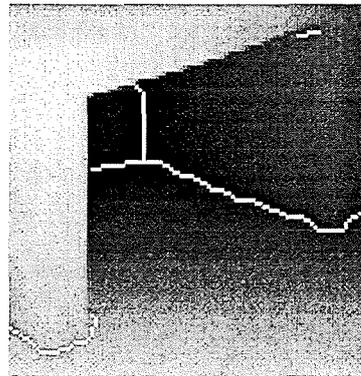
(a) Bild nach Ermittlung der „ $\pm 2$ “-  
Punkte



(b) Bild nach der bedingten Verdün-  
nung



(c) Bild aus Teilabb. 3.8b ohne ver-  
deckende und verdeckte Punkte



(d) lokalisierte konkave Punkte im Ab-  
standsbild

Abbildung 3.8: Unterscheidung konkaver und verdeckter Bildkantenpunkte: konkave und verdeckte Bildkantenpunkte besitzen gleiches Krümmungsvorzeichen. Verdeckte Punkte zeichnen sich aber durch die Nachbarschaft zu verdeckenden Punkten aus. Durch die bedingte Verdünnung (Teilabb. (b)) und das Entfernen von verdeckten und verdeckenden Punkten (Teilabb. (c)) wird die Extraktion konkaver Kantenpunkte ermöglicht.

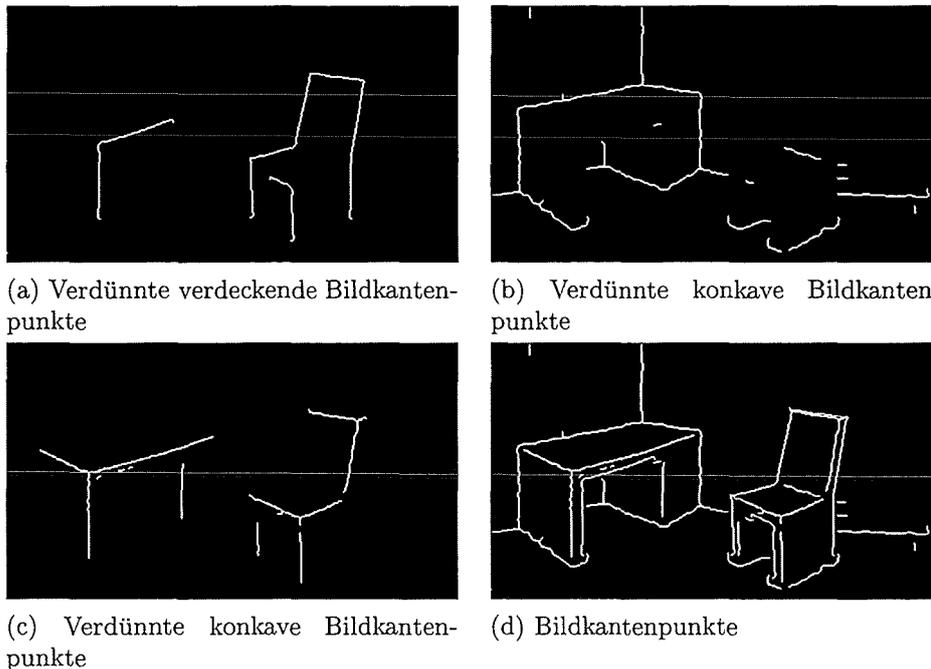


Abbildung 3.9: Extrahierte Bildkantenpunkte verschiedenen Typs

Die Symbole „+“, „-“ bzw. „ $\pm_{max}$ “ stehen im folgenden nicht mehr für Krümmungsvorzeichen und -wechsel, sondern werden als physikalisch als konvexe, konkave bzw. verdeckende Kanten interpretiert.

### 3.4 Verkettung von Bildkantenpunkten

Eine wichtige Eigenschaft der lokalisierten verdeckenden Bildkantenpunkte ist ihre gegenseitige Nachbarschaft im Bild, so daß eine Verkettung benachbarter Punkte möglich ist. Nach der Idee in [Hebert & Krotkov 92], verdeckende Kantenpunkte durch Entfernen isolierter 3D-Szenenpunkte zu identifizieren, ist hingegen nicht garantiert, daß verdeckende Kantenpunkte als 2D-Bildpunkte verkettet werden können. Das Ziel der Kantenpunktverkettung besteht darin, in jedem der drei Binärskelette der Kantenpunktlokalisierung benachbarte Bildpunkte zwischen markanten Skelettpunkten zusammenzufassen. Aus jedem 2D-Bildpunkt einer solchen Kette kann direkt die entsprechende Kugelkoordinate und damit auch die entsprechende kartesische Szenenkoordinate berechnet werden. Die Bildpunktkette ist daher die Voraussetzung, um Szenenpunkte mit dreidimensionalen Kanten approximieren zu können.

Die Verkettung von Skelettpunkten wird erreicht, indem (1) charakteristische Skelettpunkte markiert werden und danach (2) Punkte zwischen charakteristischen Skelettpunkten verkettet werden. Charakteristische Skelettpunkte wie Verzweigungs- oder Endpunkte können Hinweise auf Vertices von Polyedern darstellen. Da ein Skelett Kantenpunkte eines einzigen Kantenpunkttyps repräsentiert, deutet ein Skelettverzweigungspunkt auf einen Vertex hin, in dem drei Kanten gleichen Typs zusammentreffen, während ein Skelettendpunkt auf einen Vertex hindeutet, in dem eine Kante mit Kanten anderen Typs zusammentrifft.

Als Merkmal von Punkten in einem Binärskelett wird ein Konnektivitätsmaß in

Anlehnung an die Konnektivitätszahl  $k$  von Rutovitz herangezogen [Haralick & Shapiro 93a]. Sie zählt die Anzahl von 0-1 Übergängen gegen den Uhrzeigersinn innerhalb eines  $n \times n$ -Fensters um einen Punkt, falls dieser gleich 1 ist. Da Binärskelette vorliegen, werden Endpunkte durch einen, Punkte auf Kanten durch zwei und Verzweigungen durch drei 0-1 Übergänge detektiert. Idealerweise können keine höheren Werte für  $k$  entstehen, da sich in einem Bild trihedraler Polyeder maximal drei Kanten treffen können. Da sich bei der Fenstergröße  $n = 3$  Bildpunkte benachbarter Kanten aber berühren können und somit zu wenige 0-1 Übergänge ermittelt werden, haben sich Fenster der Größe 5 bewährt. Mit der folgenden Numerierung von Bildpunkten um den Punkt  $e_i[u_0, v_0] = P_0$  lässt sich die Konnektivitätszahl  $k$  als Nachbarschaftsoperator schreiben:

$P_7$	$P_6$	$P_5$	$P_4$	$P_3$
$P_8$				$P_2$
$P_9$		$e_i[u_0, v_0] = P_0$		$P_1$
$P_{10}$				$P_{16}$
$P_{11}$	$P_{12}$	$P_{13}$	$P_{14}$	$P_{15}$

Der Nachbarschaftsoperator liefert für jedes Eingabebild  $e_i(u, v)$ , das ein Binärskelett repräsentiert, ein zugehöriges Bild  $f_i(u, v)$  mit dem Wertebereich  $\{0, 1, 2, 3\}$ . Dabei stellt jedes Element des Wertebereiches eine Konnektivitätszahl an einer Stelle im entsprechenden Binärbild dar. Der Nachbarschaftsoperator lautet wie folgt:

```

[[ Binärbild  $e_i$  ist Eingabe
  if  $P_0 = 1$  then
    (* Konnektivitätszahl *)
     $n_{01} \leftarrow 0$ ;
    for  $i \leftarrow 1$  to 16 do
      if  $P_i = 0$  and  $P_{i+1 \bmod 16} = 1$  then  $n_{01} \leftarrow n_{01} + 1$ ;
       $f_i[u, v] \leftarrow n_{01}$ ;
    od;
  else
     $f_i[u, v] \leftarrow 0$ 
  fi;
]]_{u,v}
```

Mit Hilfe des Nachbarschaftsoperators kann man den Algorithmus zur Verkettung von Bildpunkten wie folgt schreiben.

**Algorithmus 3.2 (Bildkantenpunktverkettung)** *Eingabe sind die Binärbilder  $e_1(u, v)$ ,  $e_2(u, v)$ ,  $e_3(u, v)$  verdeckender, konvexer und konkaver Bildpunkte. Ausgabe sind Listen von Bildpunktekettens verdeckender, konvexer und konkaver Bildpunkte. Führe folgende Anweisungen getrennt für jedes einzelne Binärbild aus:*

1. Erzeuge mit dem obigen Nachbarschaftsoperator aus dem Binärbild  $e_i(u, v)$  das Bild  $f_i(u, v)$ .
2. Füge die Bildkoordinaten  $(u, v)$  mit  $f_i(u, v) \in \{1, 3\}$  in eine Liste  $\mathcal{L}$  ein.
3. Für jeden Bildpunkt  $p$  aus  $\mathcal{L}$  mit der Konnektivitätszahl 1:
  - (a) Verkette jeden innerhalb eines  $3 \times 3$ -Fensters benachbarten Skelettpunkt beginnend mit  $p$  zu einer Bildpunktkette  $K$ , bis ein Bildpunkt aus  $\mathcal{L}$  mit der Konnektivitätszahl  $k \in \{1, 3\}$  erreicht ist und lösche die Bildpunkte der Liste im bearbeiteten Binärbild.

- (b) Gebe die Bildpunktkette  $\mathcal{K}$  aus und vermindere die Konnektivitätszahl des Start- und Endpunktes der Bildpunktkette in der Liste  $\mathcal{L}$  um 1.
4. Wiederhole Schritt 2, bis die Konnektivitätszahlen aller Listenelemente in  $\mathcal{L}$  gleich 0 sind.

Um Rauschanteile bzgl. der Abstandswerte in den entsprechenden Bildpunkten des Abstandsbildes zu vermindern, werden die den Bildpunkten der Kette entsprechenden Abstände gefiltert. Da an die Bildpunktketten dreidimensionale Kanten angepaßt werden sollen, sollen Eckpunkte möglichst gut erhalten bleiben. Diese Eigenschaft besitzt der Medianfilter. Die Medianfilterung einer Liste von Abständen beschreibt der folgende Algorithmus.

**Algorithmus 3.3 (Medianfilterung von Abstandsketten)** *Eingabe ist eine Bildpunktkette  $\mathcal{K}$ , das originale Abstandsbild  $r(u, v)$  und die Fenstergröße  $n$ . Ausgabe ist eine Kette  $\mathcal{A}_{\text{median}}$  von gefilterten Abständen für entsprechende Bildpunkte der Bildpunktkette  $\mathcal{K}$ .*

1. Generiere aus der Liste  $\mathcal{K}$  von Bildpunkten eine Liste  $\mathcal{A}$  von Abständen, die an der entsprechenden Stelle im originalen Abstandsbild auftreten. Initialisiere eine Liste  $\mathcal{A}_{\text{median}}$ .
2. Schiebe ein Fenster über die Liste  $\mathcal{A}$ , das  $n$  aufeinanderfolgende Listeneinträge lang ist. An jeder Position:
  - (a) Sortiere die  $n$  Abstandswerte innerhalb des Fensters aufsteigend.
  - (b) Hänge den Abstandswert der Position  $n \text{ div } 2$  der sortierten Folge an die Liste  $\mathcal{A}_{\text{median}}$  an.
3. Gebe die Liste  $\mathcal{A}_{\text{median}}$  aus.

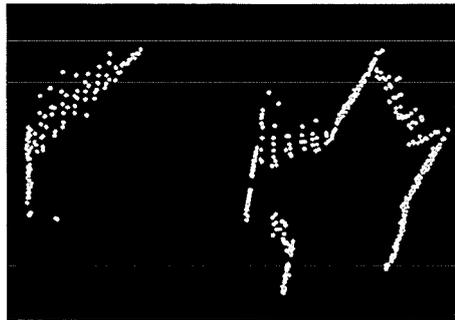
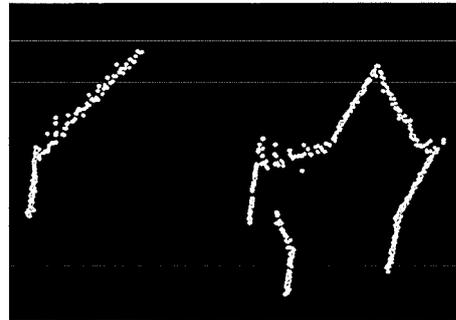
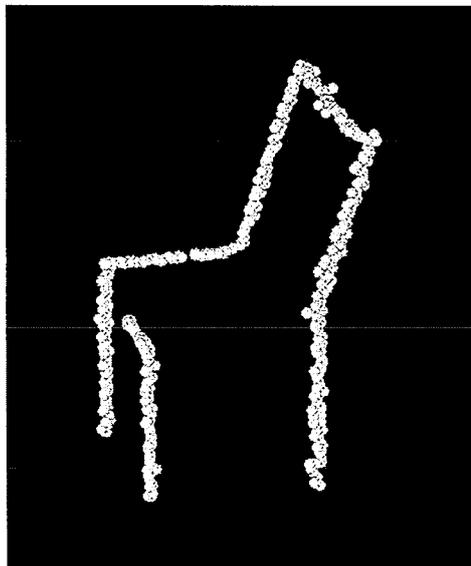
In Abbildung 3.10 (e) und (f) sind die lokalisierten verdeckenden 3D-Szenenpunkte des stuhlförmigen Objektes der Testszene abgebildet. In Teilabb. 3.10e sind die lokalisierten verdeckenden Kantenpunkte ohne Medianfilterung und in Teilabb. 3.10f mit Medianfilterung ersichtlich. Rauschanteile entlang der Punktkette wurden deutlich vermindert.

Zusammenfassend wird der Effekt der Lokalisierung verdeckender Kantenpunkte und der Medianfilterung von Abstandsketten in den Teilabb. 3.10 (a)-(d) verdeutlicht. Um die Wirkungsweise zu visualisieren, wurde eine Ansicht der Szenenpunkte von schräg oben gewählt. Deutlich sichtbar sind Szenenpunkte, die durch den Mixed-Point Effekt verfälscht wurden. Nach schrittweiser Anwendung des hier vorgestellten Verfahrens liegen alle verdeckenden Punkte auf Objektflächen im Vordergrund.

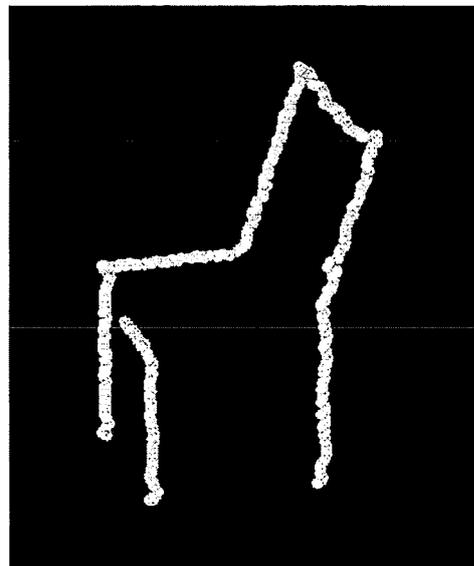
## 3.5 Zusammenfassung

Als Bildbereichshinweise dienen Bildpunktketten aus konvexen, konkaven und verdeckenden Bildkantenpunkten. Das Ziel wird in drei Schritten erreicht: Krümmungsberechnung, Kantenpunktlokalisierung und Bildpunktverkettung.

Die Berechnung von Mittleren Krümmungen wurde so angepaßt, daß der Krümmungsvorzeichenwechsel zur Unterscheidung von verdeckenden und verdeckten Kantenpunkten und zur Lokalisierung verdeckender Kantenpunkte herangezogen werden kann. Die angepaßte Mittlere Krümmung ermöglicht es dadurch einer nachgelagerten Lokalisierungsphase, Bildpunkte auf verdeckenden Kanten von solchen zu

(a) „ $\pm_0$ “-Bildpunkte als Szenenpunkte(b) „ $\pm_1$ “-Bildpunkte als Szenenpunkte(c) „ $\pm_2$ “-Bildpunkte in Szenenpunkte(d) Median-gefilterte „ $\pm_2$ “-Bildpunkt-ketten

(e) Vergrößerter Ausschnitt aus Teilabbildung (c)



(f) Median-gefilterte Bildpunktkette als Szenenpunkte

Abbildung 3.10: Wirkung der Kantenpunktlokalisierung und der Medianfilterung auf Szenenpunkte: Die „ $\pm_i$ “-Punkte der Testszene wurden in Szenenkoordinaten transformiert, um die Wirkung der Kantenpunktlokalisierung zu verifizieren (Teilabb. (a)-(d)). Eine weitere Verbesserung wird durch die Medianfilterung der „ $\pm_2$ “-Bildpunktkette erreicht (Teilabb. (e)-(f)).

unterscheiden, die durch den Mixed-Point Effekt verfälscht sind. Da als Merkmal einer Sprungkante ein Vorzeichenwechsel dient, können außerdem auch verschmierte und geglättete Sprungkantenpunkte relativ genau lokalisiert werden. Im Gegensatz zu ähnlichen Arbeiten werden Krümmungen nicht in diskreten Richtungen, sondern richtungsunabhängig berechnet, wodurch die getrennte Behandlung richtungsabhängiger Krümmungen vermieden und eine höhere Lokalisierungsgenauigkeit erzielt wird.

Die Kantenpunktlokalisierung verfeinert unter Berücksichtigung des vorhandenen Szenenwissens insbesondere verdeckende Kantenpunkte so, daß sie mit hoher Wahrscheinlichkeit nicht durch den Mixed-Point Effekt verfälscht sind und sie im Gegensatz zu anderen Verfahren auch zu Bildpunktketten zusammengefaßt werden können. Dem Autor sind keine Arbeiten bekannt, die verdeckende Sprungkanten in Abstandsbildern von Laserradarscannern so detektieren und lokalisieren, daß eine Kantenpunktverkettung möglich wäre.

Die Bildpunktverkettung stellt jede Bildpunktkette mit zugehöriger Abstandskette bereit, die beide zusammen die Transformation in dreidimensionale Szenenkoordinaten erlauben. Die Abstandsketten sind so aufbereitet, daß Rauscheinflüsse reduziert sind und gleichzeitig die geometrische Information über Vertizes erhalten bleibt.



# Kapitel 4

## Kantenbasierte Szenenbereichshinweise

Szenenbereichshinweise stellen charakteristische Merkmale von Objekten der Szene dar. Kantenbasierte Szenenbereichsstrukturen bestehen aus dreidimensionalen Kanten und ihren wechselseitigen Beziehungen.

### 4.1 Übersicht über kantenbasierte Szenenbereichshinweise

Alle Ansätze zur Generierung von kantenbasierten Szenenbereichsstrukturen aus Abstandsbildern zum Zwecke der Objekterkennung arbeiten zumindest teilweise nach folgendem Ablauf bzw. mit folgenden Techniken:

1. Repräsentation der Szenenbereichsstruktur als attributierter Graph
2. Kantendetektion
3. Kantenpunktverkettung
4. Approximation mit geometrischen Primitiven
5. Kantengruppierung
6. Schlußfolgern neuer und Verfeinern vorhandener Attribute

Eine kantenbasierte Szenenbereichsstruktur wird bzw. kann als attributierter Graph aufgefaßt werden. Knoten des Graphen repräsentieren Vertices und Bögen zwischen Knoten repräsentieren Kanten zwischen Vertices. Bogenattribute sind beispielsweise die Kantentypen konvex, konkav und verdeckend. Knotenattribute sind beispielsweise die kartesischen 3D-Koordinaten (Punkt 1). Um Bildkantenpunkte zu erhalten, die mit hoher Wahrscheinlichkeit einer gemeinsamen Kante entsprechen, werden zunächst benachbarte Kantenpunkte verschiedenen Typs detektiert (Punkt 2) und verkettet (Punkt 3). Die Punktketten werden dann nach 2D- bzw. 3D-Merkmalen durch geometrische 2D- bzw. 3D-Primitive approximiert (Punkt 4). Kollineare Kanten sowie Kanten mit nahen Endpunkten werden dann nach bestimmten Regeln gruppiert (Punkt 5). Im letzten Schritt werden gegebenenfalls weitere aufgabenspezifische Attribute wie der eingeschlossene Winkel abgeleitet (Punkt 6). Alle dem

Referenz	Merkmale	Szenen
[Shirai & Suwa 71]	Kanten	Polyeder
[Sugihara 79]	Kanten	trihedrale Polyeder
[Bolles & Horaud 83]	gerade und kreisförmige Kanten	Zylinder
[Hebert & Kanade 85]	Konturkanten	Polyeder
[Smith & Kanade 85]	gekrümmte Konturkanten	konische und zylindrische Flächen
[Herman 85]	Kanten	Polyeder
[Fan <i>et al.</i> 87], [Fan 90]	Kantentypen	durch Kanten begrenzte gekrümmte Flächen
[Godin & Levine 89]	Kantenendpunkte, Linienverbindungen	gerade und gekrümmte Kanten

Tabelle 4.1: Übersicht der Ansätze zur Extraktion kantenbasierter Szenenbereichshinweise

Autor bekannten Verfahren, die solche Kantenbeschreibungen aufbauen, arbeiten mit Abstandsbildern von Triangulationslaserscannern. In Abstandsbildern solcher Sensoren erscheinen Kanten wie in ihrer perspektivischen Projektion. Das heißt, gerade Kanten werden auf gerade Linien im Bild abgebildet. Obwohl gerade Kanten in den hier verwendeten Laserradarbildern als leicht gekrümmte Linien erscheinen, sind viele Methoden übertragbar.

## Übersicht

Unterschiede der einzelnen Ansätze ergeben sich aus der Struktur der Szene und zu erkennender Objekte. Daraus ergibt sich die Wahl geeigneter Attribute des Graphen und geeigneter Methoden zur Realisierung der einzelnen Verarbeitungsschritte. Während alle Ansätze die Kantenpunktdetektion und die Kantenpunktverkettung im Bildbereich durchführen, arbeiten die nächsten Verarbeitungsschritte mit einer Kombination von Merkmalen in Bild- und Szenenbereich.

Sugihara [Sugihara 79] generiert Strichzeichnungen aus Abstandsbildern trihedral polyedrischer Szenen. Alle Verarbeitungsschritte arbeiten im Bildbereich. Ein Lexikon möglicher Linienverbindungen dient dazu, aufgrund bereits extrahierter Linien fehlende Linien vorherzusagen. Linienhypothesen werden aufgrund bereits vorhandener Linienverbindungen überprüft. Die Aufspaltung von Bildpunktketten an Punkten hoher 2D-Kurvenkrümmung kann problematisch sein, da die Krümmung aufgrund der Bildprojektion auch gering ausfallen kann. Ein ähnliches Problem stellt die Verfeinerung der 2D-Position einer Linienverbindung dar. Denn trotz kleiner Abweichung zwischen aktueller und korrigierter 2D-Position kann die Abweichung der 3D-Vertexpositionen groß sein. Die resultierenden Linien werden nicht in Kanten umgerechnet.

Bolles und Horaud [Bolles & Horaud 83] betrachten Szenen, die aus mehreren Instanzen eines einzigen Objektes bestehen, zum Zwecke der Erkennung und Lagebestimmung. Objekte sind durch Kanten und kreisrunde Konturen charakterisiert. In den ersten Verarbeitungsschritten werden Bildpunktketten verschiedenen Typs generiert. Die Bildpunktketten werden dann aber in Szenenpunktketten umgerechnet. Da sich Objekte durch Kantenmerkmale auszeichnen, die jeweils in einer gemeinsamen Ebene liegen, werden 3D-Szenenpunktketten in Teilketten zerlegt, die diese

Eigenschaft erfüllen. Teilketten werden dann aber wieder in diese Ebene projiziert, so daß die Kantenanpassung wieder in 2D-Koordinaten bzgl. dieser Ebene erfolgt. Kanten werden jedoch nicht als alleinige Merkmale zur Objekterkennung, sondern als Hypothesen für die Lage von Objektmerkmalen höherer Ordnung herangezogen. Auf der Ebene von Kanten wird daher keine Gruppierung durchgeführt.

Herman [Herman 85] extrahiert Kantenbeschreibungen auch nicht-trihedral polyedrischer Szenen. Mit Hilfe einer 2D-Hough-Transformation werden 2D-Linien an Bildkantenpunktketten verschiedenen Typs angepaßt. Verdeckende Kanten werden einheitlich nach der Lage der angrenzenden Fläche ausgerichtet, indem die durchschnittlichen Abstände auf jeder Seite der verdeckenden Kante verglichen werden. 2D-Linien werden dann gemäß einer Fallunterscheidung zu Linienverbindungen gruppiert. Allerdings entstehen auch Linienverbindungen, die keine Entsprechung in Form eines Polyedereckpunktes besitzen. Erst im letzten Schritt werden Linienverbindungen in Vertizes und Linien in Kanten umgerechnet. Ähnlich arbeitet der Ansatz von [Hebert & Kanade 85] zu Erkennung polyedrischer Objekte. Jedoch verwenden sie nur verdeckende Kanten als Silhouetten, die charakteristisch für zu erkennende Ansichten polyedrischer Objekte sind.

Smith und Kanade [Smith & Kanade 85] extrahieren keine Dachkanten, sondern nur Konturkanten aus Abstandsbildern konischer und zylindrischer Objekte. Aus Merkmalen wie Konturkanten werden schrittweise konische und zylindrische Flächenelemente generiert, deren wechselseitige Beziehungen in einem Graphen repräsentiert werden. Konturkanten wie gerade Kanten, kreisförmige Kurven, nicht-kreisförmige Kurven innerhalb einer Ebene und nicht-planare Kurven werden in acht Typen klassifiziert, indem geometrische Beziehungen bei der Bildaufnahme durch einen Triangulationslaserscanner analysiert werden. Entlang verketteter Konturpunkte werden mittels lokaler Krümmungen Krümmungsextrema identifiziert. Zwischen Krümmungsextrema werden dreidimensionale Kurven der acht Kurventypen angepaßt. Das Verfahren eignet sich allerdings nur für Objekte mit gekrümmten Flächen, da keine Dachkanten detektiert werden.

Fan et. al. [Fan et al. 87], [Fan 90] extrahieren Kanten-Flächenbeschreibungen für Szenen, die aus gekrümmten Flächen und dazwischenliegenden Unstetigkeiten bestehen. Allerdings werden Kantenpunkte verschiedenen Typs nur dazu verwendet, um geschlossene Bildpunktketten zu extrahieren. Die eingeschlossenen Regionen dienen dann zum Anpassen von Flächen. Als Information von Kanten wird lediglich der Kantentyp verwendet.

Godin und Levine [Godin & Levine 89] generieren aus Abstandsbildern von Szenen, die durch gerade und gekrümmte Kanten charakterisiert sind, sog. Kanten-Linienverbindungs-Graphen. Ein Kanten-Linienverbindungs-Graph repräsentiert in einem Knoten die Linienverbindung einer Strichzeichnung und in einem Bogen eine dreidimensionale Kante. Kantenpunkte verschiedenen Typs werden verkettet und bereits im nächsten Schritt in Szenenpunktketten umgerechnet. Da die Szenenpunktketten durch ein lokales Verfahren in zu approximierende Teilketten zerlegt werden, ist die Approximation rauschempfindlich. Zuletzt werden kollineare Kanten gruppiert, danach trihedrale Linienverbindungen und zuletzt Verbindungen krummliniger Linien hergestellt. Linienverbindungen werden nicht wie in [Sugihara 79] zur Steuerung der Kantendetektion, sondern als Merkmale von Vertizes verwendet. Im Gegensatz zu [Sugihara 79] werden aber nicht alle Linienverbindungen verwendet, die durch Berührung zwischen Polyedern entstehen können.

In dem hier vorgeschlagenen Ansatz wird die Unstetigkeitsmarkierung ähnlich der Repräsentation von [Godin & Levine 89] verwendet. Allerdings sollen die Linienverbindungen von [Sugihara 79] verwendet werden, die auch die Berührung von Polyedern berücksichtigen.

## 4.2 Generieren der Szenenkantenstruktur

### 4.2.1 Approximation von Kanten

Ausgangspunkt zur Extraktion dreidimensionaler Kanten sind Bildbereichshinweise in Form von Bildpunktketten und zugehöriger mediangefilterter Abstandsketten, die Kantenpunkte verschiedenen Typs repräsentieren. In den Arbeiten von [Sugihara 79] und [Herman 85] wurden 2D-Bildpunktketten direkt im Bild durch Linien approximiert, um erst später die Umrechnung in 3D-Kanten durchzuführen. Diese Vorgehensweise ist nur deshalb möglich, da in Abstandsbildern von Triangulationslaserscannern gerade Kanten aufgrund der perspektivischen Projektion auch als gerade Linien erscheinen.

Diese Vorgehensweise ist im Falle der vorliegenden Laserradarbilder nicht möglich, da Szenenkanten aufgrund ungleichförmiger Abtastung im Bild als leicht gekrümmte Linien erscheinen. Deshalb werden hier die 2D-Bildpunktketten zusammen mit den zugehörigen Abstandsketten wie in [Godin & Levine 89] gleich in 3D-Szenenpunktketten umgerechnet, um 3D-Kanten an geeignete Teilketten anzupassen. Da Szenenpunktketten in [Godin & Levine 89] auch von gekrümmten Kanten herrühren können, werden die Szenenpunktketten dort durch ein lokal arbeitendes Verfahren aufgespalten. Dabei wird schrittweise ein Fenster über die Szenenpunktkette geschoben. An die Szenenpunktkette innerhalb eines Fensters werden zwei kurze Geradensegmente angepaßt. Anhand des eingeschlossenen Winkels wird dann entschieden, ob lokal eine eventuell gekrümmte Kante oder eine Ecke vorliegt. Da das Verfahren nur wenige Bildpunkte innerhalb eines kleinen Fensters verwendet, ist es relativ rauschempfindlich. Weil in den hier betrachteten Szenen nur gerade Kanten auftreten, wird das robustere Verfahren der iterativen Endpunktanpassung verwendet [Duda & Hart 73]. Das Verfahren benutzt zum Aufspalten einer Punktkette nicht nur Punkte innerhalb einer lokalen Umgebung, sondern alle Punkte einer Punktkette, wodurch es rauschunempfindlicher ist.

Die Approximation von Kanten an Szenenpunktketten besteht aus zwei Teilen:

- Aufspalten in Teilketten
- Akzeptanz von Teilketten

Das Aufspalten einer Szenenpunktkette in Teilketten durch das Verfahren der iterativen Endpunktanpassung kann man wie folgt beschreiben. Das Verfahren fordert stetige, also Punktketten ohne Abstandssprünge. Die Forderung ist jeweils für Ketten konvexer, konkaver und verdeckender 3D-Szenenpunktketten erfüllt. Die Funktionsweise wird in Abb. 4.1 veranschaulicht. Durch Anfangs- und Endpunkt einer Szenenpunktkette wird eine 3D-Gerade gelegt. Danach wird der maximale Abstand  $d_{max}$  aller Kettenpunkte zu der 3D-Gerade ermittelt. Falls der maximale Abstand unterhalb eines vorher festgelegten Schwellwertes liegt, wird eine potentielle 3D-Kante zwischen Anfangs- und Endpunkt ausgegeben. Andernfalls wird die Szenenpunktkette an dem Punkt maximalen Abstands in Teilketten aufgespalten und das Verfahren rekursiv auf die Teilketten angewendet. Das Verfahren liefert also eine Menge von Teilketten und zugehörigen Kanten, die den Kopf- und den Endpunkt jeder Kette als Kantenendpunkte besitzen.

Teilketten sollen aber nur dann akzeptiert werden, wenn sie mit hoher Wahrscheinlichkeit Kanten der Szene repräsentieren. Die Teilketten sollen daher die folgenden heuristischen Bedingungen erfüllen.

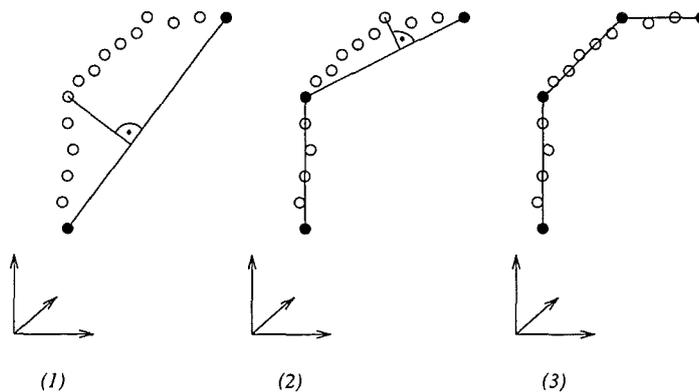


Abbildung 4.1: Funktionsweise der iterativen Endpunktanpassung: Eine Szenenkette wird solange in Teilketten zerlegt, bis der resultierende Polygongzug die Szenenkette genügend genau approximiert.

**Bedingung 4.1 (Teilkettenlänge, Kantenlänge)** Die Teilkettenlänge ist die Anzahl der Szenenpunkte einer Teilkette, die Kantenlänge ist der euklidische Abstand zwischen den Endpunkten einer Kante. Sowohl jede Teilkette, als auch jede Kante soll eine Mindestlänge aufweisen.

**Bedingung 4.2 (Kantenlänge-Kettenlänge-Verhältnis)** Das Verhältnis  $\frac{d}{n}$  aus Kantenlänge  $d$  und Teilkettenlänge  $n$  soll eine bestimmte Schwelle  $\chi_{max}$  nicht überschreiten. Die Schwelle  $\chi_{max}$  wird mit der Funktion  $f_{n_0}$  in Abhängigkeit eines weiteren Schwellwertes  $n_0$  bzgl. der Kettenlänge  $n$  gesetzt:

$$\chi_{max} = f_{n_0}(n) = \begin{cases} \chi_{max1} & \text{für } n \geq n_0 \\ \chi_{max2} & \text{für } n < n_0 \end{cases}$$

Die Notwendigkeit der zweiten Bedingung ergibt sich daraus, daß Szenenkanten gleicher Länge bei senkrechter Aufsicht viele, bei schräger Aufsicht aber nur wenige Kantenpunkte abgetastet werden können. Es sollte nicht erlaubt sein, daß approximierten Kanten nur aufgrund weniger Abtastpunkte akzeptiert werden, die eventuell durch gewisse Rauschquellen verursacht sein könnten. Die zweite Bedingung formuliert daher, daß zwischen der euklidischen Länge einer Kante und der Anzahl der approximierten Kantenpunkte ein bestimmtes maximales Verhältnis nicht überschritten werden darf. Dieses Verhältnis wird durch den Schwellwert  $\chi_{max}$  festgelegt.

Allerdings ergibt sich das Problem, für Teilketten verschiedener Länge geeignete Schwellwerte zu finden. Lange Teilketten besitzen aufgrund der hohen Anzahl an Abtastpunkten auch eine hohe Wahrscheinlichkeit, auf einer Szenenkante zu liegen. Das obige Verhältnis darf also relativ hoch ausfallen. Im Falle von kurzen Teilketten stellen die Abtastpunkte nur dann mit hoher Wahrscheinlichkeit Kantenpunkte dar, wenn auch die entsprechenden Kanten kurz und damit das Verhältnis relativ gering ist. Experimentell hat sich als ausreichend erwiesen, lange und kurze Punktketten anhand eines weiteren Schwellwertes  $n_0$  zu unterscheiden, wobei für lange Punktketten ein hohes maximales Verhältnis  $\chi_{max1}$ , für kurze Punktketten ein kleineres maximales Verhältnis  $\chi_{max2}$  erlaubt ist (siehe Abb. 4.2). Folgender Algorithmus wurde für die Kantenapproximation verwendet:

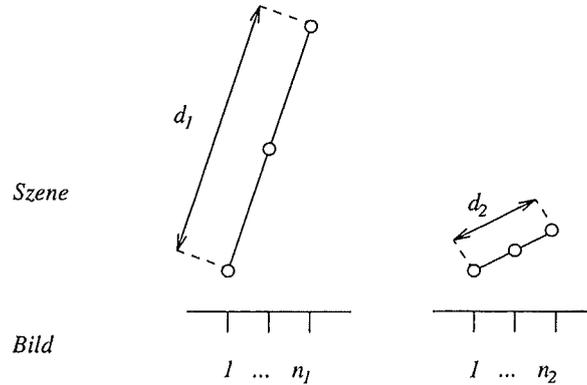


Abbildung 4.2: Maximales Verhältnis von Kanten- zu Kettenlänge als Kriterium zur Akzeptanz approximierter Teilketten: nur die Teilkette im rechten Teil der Abbildung wird akzeptiert, da das Verhältnis  $\frac{d_2}{n_2}$  kleiner als die Schwelle  $\chi_{max}$  ist.

**Algorithmus 4.1 (Kantenapproximation)** *Eingabe ist eine Bildpunktkette aus konvexen, konkaven oder verdeckenden Kantenpunkten. Jedem Kettenpunkt ist der medianefilterte Abstand zugeordnet. Ausgabe ist eine Menge von Kanten, die durch die kartesischen Koordinaten der beiden Endpunkte festgelegt sind.*

1. Berechne aus der Bildpunktkette eine Szenenpunktkette. Dazu wird jeder Kettenpunkt  $(u, v)$  zusammen mit dem zugehörigen Abstand  $r_{median}(u, v)$  nach folgender Formel in den korrespondierenden Szenenpunkt  $(x, y, z)^T$  überführt:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r_{med}(u, v) \cos \phi_u \sin \theta_v \\ r_{med}(u, v) \sin \phi_u \sin \theta_v \\ r_{med}(u, v) \cos \phi_u \end{pmatrix}$$

mit:

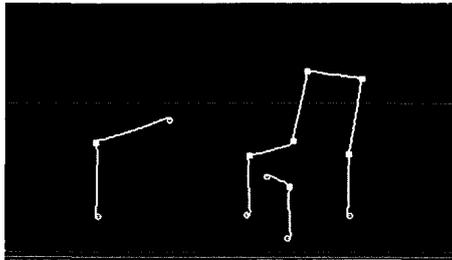
$$\begin{aligned} \phi_u &= \phi_{min} + u \Delta \phi \\ \theta_v &= \theta_{min} + v \Delta \theta \end{aligned}$$

2. Wende das Verfahren der iterativen Endpunktanpassung in [Duda & Hart 73] mit dem Schwellwert  $d_{max}$  auf die Szenenpunktkette an.
3. Akzeptiere nur solche Teilketten aus Schritt 2, die die Bedingungen 4.1 und 4.2 erfüllen.

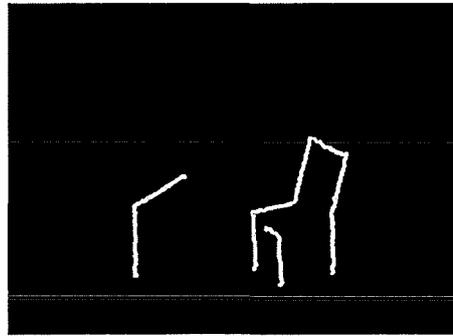
In Abb. 4.3 sind Splitpunkte von Punktekettens verschiedenen Typs als Quadrate, Anfangs- bzw. Endpunkte von Punktekettens als Kreise markiert. Die Kettenpunkte sind sowohl als Bild- als auch als Szenenpunkt visualisiert. Das Resultat besteht aus einer Menge von Kanten gleichen Typs. Das Ergebnis für die Punktekettens verschiedenen Typs zeigt Abb. 4.4.

#### 4.2.2 Gruppierung von Kanten

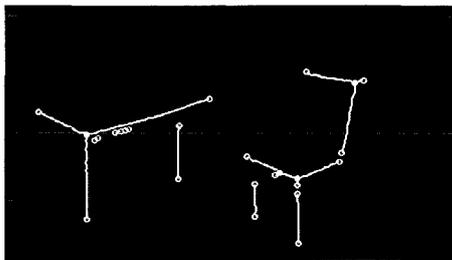
Für die Erkennung von Objekten anhand von Kanten, sind auch Beziehungen zwischen einzelnen Kanten von Bedeutung. Approximierte Kanten können durch zweierlei Maßnahmen in eine Kantenstruktur überführt werden, die solche Beziehungen



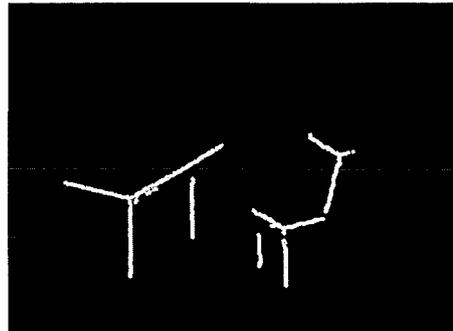
(a) 2D ±



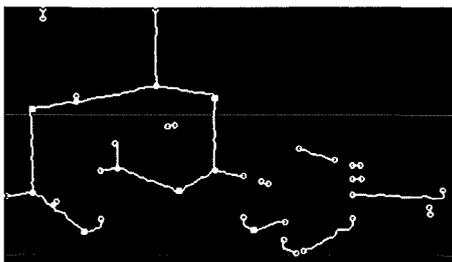
(b) 3D ±



(c) 2D +



(d) 3D +



(e) 2D -



(f) 3D -

Abbildung 4.3: 2D-Bildpunktketten mit Splitpunkten (links) und Lage der 3D-Szenenpunkte (rechts): In Teilabb. (a), (c) und (e) sind in 2D-Bildpunktketten Splitpunkte als Quadrate und Endpunkte als Kreise markiert. In Teilabb. (b), (d) und (f) ist die perspektivische Projektion der jeweiligen dreidimensionalen Kantenpunkte abgebildet. Jeder Szenenpunkt ist durch eine kleine schattierte Kugel dargestellt.

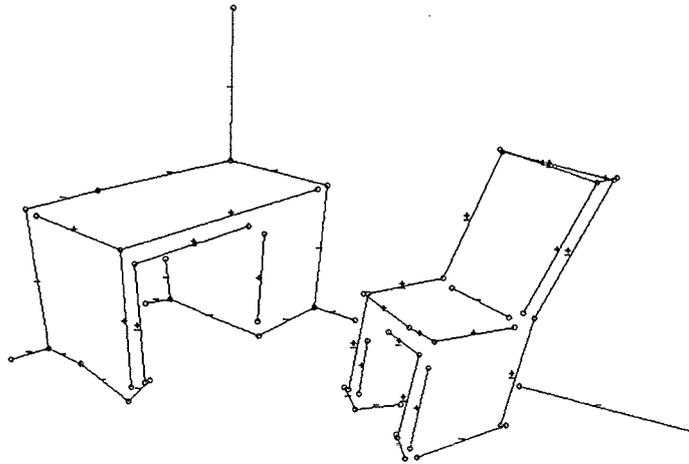


Abbildung 4.4: Approximierte Kanten der Testszene: Konvexe Kanten sind durch ein „+“, konkave Kanten durch ein „-“ und verdeckende Kanten durch ein „±“ markiert.

repräsentiert. Die erste Maßnahme betrifft die Tatsache, daß für eine Szenekante mehrere approximierte Kanten gleichen Typs extrahiert sein können. Da sich polyedrische Szenen durch gerade Szenekanten auszeichnen, können nahezu kollineare Kanten zu einer einzigen zusammengefaßt werden.

Die zweite Maßnahme betrifft approximierte Kanten, die mit hoher Wahrscheinlichkeit auf Szenekanten liegen, die ihrerseits in einer gemeinsamen Polyederecke zusammentreffen. Da die betrachteten polyedrischen Szenen trihedral sind, können maximal drei verschiedene Kantenendpunkte zu einem einzigen Vertex gruppiert werden, den alle drei Kanten als gemeinsamen Endpunkt besitzen. Bei beiden Operationen kommt es nicht auf die exakte Rekonstruktion der Position und Lage von Kanten und Vertices an. Vielmehr soll eine Kantenstruktur generiert werden, die zur Identifizierung zu erkennender Objektteile geeignet ist. Die Gruppierung von Kanten besteht also in der Gruppierung von kollinearen Kanten gleichen Kantenyps und der Gruppierung von nicht-kollinearen Kanten.

Zur Gruppierung von Paaren nicht-kollinear Linien hat [Herman 85] fünf Fälle unterschieden, je nachdem, ob sich der Schnittpunkt zweier Linien, auf bzw. außerhalb einer oder beider Linien befindet und je nach dem, wie stark eine Linie oder beide Linien verlängert oder verkürzt werden müssen, um eine gemeinsame Linienverbindung zu bilden. Die verschiedenen Fälle zur Liniengruppierung werden hier verallgemeinert und auf 3D-Kanten übertragen. In beiden Fällen müssen von zwei zu gruppierenden Kanten jeweils diejenigen Kantenendpunkte ermittelt werden, die nahe zueinander liegen. Infolge der ersten Gruppierungsmaßnahme werden nahe Endpunkte zweier kollinear Kanten entfernt, während infolge der zweiten Gruppierungsmaßnahme nahe Endpunkte nicht-kollinear Kanten zur Berechnung eines neuen Vertex verwendet werden.

Nahe Endpunkte werden wie folgt berechnet. Da zwei Kanten  $g$  und  $h$  aufgrund von Ungenauigkeiten im allgemeinen windschief sind, werden nahe Endpunkte anhand der Lotfußpunkte  $g_l$  und  $h_l$  aus Abb. 4.6 ermittelt. Wenn der Lotfußpunkt  $g_l$  der Kante  $g$  näher am Endpunkt  $g_e$  als am Anfangspunkt  $g_a$  liegt, werden Anfangs-

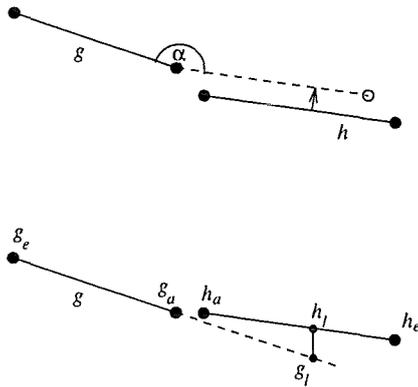


Abbildung 4.5: Gruppierungsmodell eines kollinearen Kantenpaares: Berechnung des Winkels  $\alpha$  (oben) und Lage mindestens eines Lotfußpunktes außerhalb einer Kante (unten).

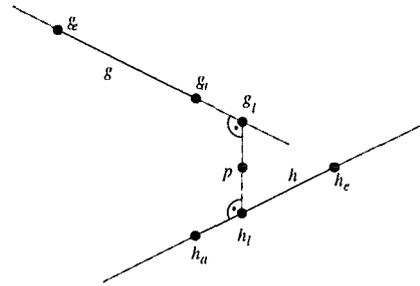


Abbildung 4.6: Gruppierungsmodell eines nicht-kollinearen Kantenpaares: Die Längenänderung von Kante  $g$  beträgt  $|g_a - g_l|$ .

und Endpunkt vertauscht. Wird der Vorgang analog für Kante  $h$  durchgeführt, stellen  $g_a$  und  $h_a$  die nahen Kantenendpunkte dar.

Um zu entscheiden, ob zwei Kanten kollinear sind, muß überprüft werden, (1) ob die entsprechenden Geraden nahezu parallel sind und, (2) ob die Kanten nicht nebeneinander liegen. Die erste Bedingung kann durch den Winkel  $\alpha$  in Abb. 4.5 überprüft werden. Die zweite Bedingung wird anhand der Parameterdarstellung der beiden Lotfußpunkte überprüft:

$$\begin{aligned} g_l &= g_a + \lambda_g(g_e - g_a) \\ h_l &= h_a + \lambda_h(h_e - h_a) \end{aligned}$$

Wenn beide Lotfußpunkte nicht in dem mittleren Bereich der Kante liegen, werden sie als nicht nebeneinander gelegen angesehen. Dieser Sachverhalt wird formal durch folgende Bedingung bzgl. der Parameter  $\lambda_g$  und  $\lambda_h$  ausgedrückt, wobei  $0 < \epsilon < 0.5$ :

$$\lambda_g \notin [0.5 - \epsilon, 0.5 + \epsilon] \wedge \lambda_h \notin [0.5 - \epsilon, 0.5 + \epsilon]$$

**Regel 4.1 (Kollineare Kanten gleichen Typs)** Zwei kollineare Kanten  $g$  und  $h$  gleichen Typs mit nahen Endpunkten  $g_a$  und  $h_a$  sollen durch die Kante  $(g_e, h_e)$  ersetzt werden, wenn der Winkel  $180^\circ - \alpha$  in Abb. 4.5 kleiner als ein gegebener Schwellwert ist und die Lage der Lotfußpunkte in Abb. 4.5 die Bedingung der Lotfußpunkte  $g_l$  und  $h_l$  bzgl.  $\lambda_g$  und  $\lambda_h$  erfüllt.

**Algorithmus 4.2 (Gruppierung kollinearere Kanten gleichen Typs)** Eingabe sind approximierter konvexe, konkave und verdeckende Kanten. (1) Durchlaufe alle Paare von Kanten gleichen Typs. Wende auf jedes Kantenpaar Regel 4.1 an. (2) Wiederhole Schritt 1, bis keine Gruppierungen mehr möglich sind.

Bei dem obigen Gruppierungsvorgang ist die Reihenfolge der Gruppierung nicht kritisch. Bei der Gruppierung nicht-kollinearere Kanten hingegen muß die Reihenfolge beachtet werden, da in Vertizes trihedral polyedrischer Szenen maximal drei Kanten zusammentreffen dürfen. Daher müssen Kanten mit höherer Wahrscheinlichkeit, einen gemeinsamen Kantenendpunkt zu besitzen, zuerst gruppiert werden.

Um festlegen zu können, welche Kantenpaare zuerst gruppiert werden, kann man diskrete und kontinuierliche Parameter heranziehen. Zunächst sollen Kanten bevorzugt gruppiert werden, die in nahen Endpunkten keine Nachbarkanten besitzen. Die Anzahl der Nachbarn stellt dabei ein diskreter Parameter dar. Unter Kantenpaaren mit dieser Eigenschaft sollen wiederum solche zuerst gruppiert werden, deren i.a. windschiefe Kanten einen geringen Abstand besitzen und deren Kantenlängenänderungen klein sind. Dabei stellen Abstand und Kantenlängenänderung kontinuierliche Parameter dar. Der Abstand und die Kantenlängenänderungen werden in Abb. 4.6 veranschaulicht.

Experimentell hat sich eine zweistufige Gruppierung als ausreichend erwiesen. In jeder Stufe dürfen die Schwellwerte für den maximalen Kantenabstand  $d_{max}$  und für die maximalen Verlängerungen  $lh_{max}$  und  $lg_{max}$  nicht überschritten werden. In der zweiten Stufe werden die Schwellwerte im Vergleich zur ersten Stufe höher gewählt.

Erst danach sollen Kantenpaare bearbeitet werden, wobei sich in dem nahen Endpunkt einer Kante (Kante  $g$ ) *keine* Nachbarkante, in dem nahen Endpunkt der anderen Kante (Kante  $h$ ) *genau eine* Nachbarkante befindet. Auch für diesen Fall wird eine zweistufige Gruppierung durchgeführt, wobei die Schwellwerte erhöht werden. Jedoch wird als Kantenlängenänderung der Kante  $h$  nur ein Wert zugelassen, der kleiner als derjenige der Kante  $g$  ist. Dies ist daher notwendig, da Kante  $h$  bereits eine Nachbarkante besitzt und daher ein große Längenänderung unwahrscheinlich ist. Der Vorteil dieser Vorgehensweise liegt darin, daß i.G. zu [Godin & Levine 89] nicht mehr als drei Kanten pro Vertex möglich sind. Der Algorithmus zur Gruppierung nicht-kollinearere Kanten kann man wie folgt beschreiben:

**Algorithmus 4.3 (Gruppierung nicht-kollinearere Kanten)** *Eingabe sind approximierete konvexe, konkave und verdeckende Kanten, wobei keine kollinearen Kanten gleichen Typs existieren.*

1. Für alle Kanten  $g$  und  $h$  ohne Nachbarkante in nahen Endpunkten:
  - (a) Setze die Schwellwerte  $d_{max}$ ,  $lg_{max}$  und  $lh_{max}$
  - (b) Wenn die Schwellwerte nicht überschritten werden, dann berechne den gruppierten Punkt mit  $\mathbf{p} = \mathbf{g}_a + 1/2(\mathbf{h}_a + \mathbf{g}_a)$  und ersetze die nahen Endpunkte durch  $\mathbf{p}$ :  $\mathbf{g}_a \leftarrow \mathbf{p}$ ,  $\mathbf{h}_a \leftarrow \mathbf{p}$ .
2. Für alle Kanten  $g$  und  $h$ , wobei  $g$  keine Nachbarkante in  $g_a$  und  $h$  genau eine Nachbarkante in  $h_a$  besitzt: Wiederhole Schritte 1a und 1b, jedoch unter der Bedingung, daß die Längenänderung  $lh_{max}$  kleiner als die Längenänderung  $lg_{max}$  ist.

Die resultierenden gruppierten konvexen, konkaven und verdeckenden Kanten der Testszene zeigt Abb. 4.7. Alle Kanten, die nach der Gruppierungsphase in einem Endpunkt keine Nachbarkante besitzen, stellen teilweise sichtbare Kanten dar. Beispielsweise sind Kanten in Abb. 4.7, die sich im Hintergrund des stuhlförmigen Objektes befinden, nur teilweise sichtbar. Eine Kante kann teilweise sichtbar sein, wenn sie von Objekten im Vordergrund teilweise verdeckt ist oder nur teilweise im Blickfeld liegt. Freie Endpunkte können dazu verwendet werden, um Objekte trotz teilweiser Verdeckung korrekt zu erkennen.

### 4.2.3 Ausrichtung verdeckender Kanten

Während in der perspektivischen Projektion tridredal polyedrischer Szenen auf beiden Seiten einer konvexen oder konkaven Linie die Region einer Polyederfläche liegt,

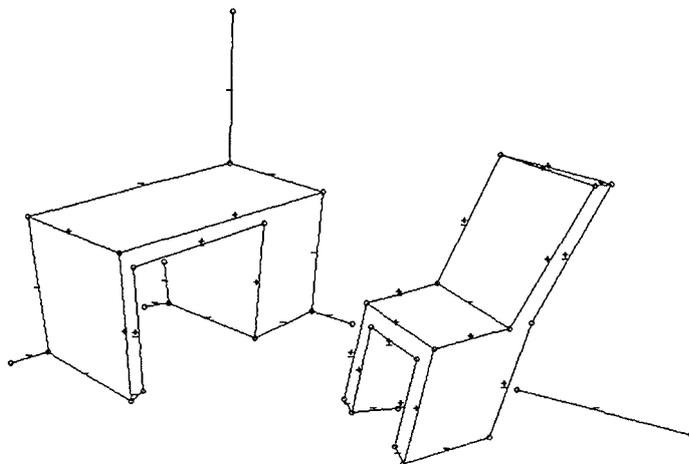


Abbildung 4.7: Gruppierete Kanten der Testszene: Kollineare Kanten des Gruppierungsmodells sind zu einer einzigen Kante gruppiert, nicht-kollineare Kanten des Gruppierungsmodells sind zu Kanten mit gemeinsamen Endpunkten gruppiert.

liegt im Falle einer verdeckenden Linie nur auf einer Seite die Region einer Polyederfläche. Auf der anderen Seite der entsprechenden Szenenkante liegt in der Szene durch einen Abstandssprung getrennt, eine dahinterliegende Polyederfläche. Durch Orientierung oder Ausrichtung verdeckender Linien läßt sich die Seite der angrenzenden Fläche spezifizieren. Die Lageinformation wird später implizit in Form markierter Linienverbindungen benutzt, die ihrerseits zur Attributierung von Vertices verwendet werden.

Per Konvention soll die Linie so ausgerichtet sein, daß die Bildregion einer Fläche rechts der Richtung einer verdeckenden Linie zu liegen kommt. Die Ausrichtung ist also blickpunktabhängig.

In [Herman 85] wird die Seite der Fläche durch Vergleich der Abstände von Punkten bestimmt, die auf beiden Seiten einer verdeckenden Kante direkt benachbart sind. In [Fan *et al.* 87] wird die Möglichkeit vorgeschlagen, die Seite der Fläche durch einen Krümmungsvorzeichenwechsel an der verdeckenden Kante zu identifizieren. Weil die verdeckenden Kantenpunkte aufgrund des Mixed-Point Effektes in der Phase der Kantenpunktlokalisierung verschoben sind, sind beide Möglichkeiten problematisch. Direkt benachbarte Punkte auf beiden Seiten von lokalisierten verdeckenden Kanten können alle auf der Polyederfläche liegen und müssen daher weder unterschiedliche Abstände (Verfahren von Herman) noch entgegengesetztes Krümmungsvorzeichen (Idee von Fan) aufweisen. Eine weitere Möglichkeit besteht in der Prüflinienmethode in [Wei 89]. Dabei wird die Seite des Vordergrunds durch eine Prüflinie senkrecht zu dem Verlauf der verdeckenden Linie ermittelt. Im Falle kurzer verdeckender Linien, die eine kurze Nachbarlinie in spitzem Winkel besitzen, kann die zugehörige Prüflinie auch die Nachbarlinie überlappen. Aus diesem Grund wird die Ausrichtung verdeckender Linien hier in Bezug auf eine Nachbarlinie wie folgt verifiziert.

Ist durch  $\pi$  die perspektivische Projektion, durch  $g$  eine verdeckende Kante und durch  $h$  eine benachbarte Kante bezeichnet, so stellen  $\pi(g)$  und  $\pi(h)$  die entsprechenden Linien dar. Dabei wird auf eine zur Blickrichtung senkrecht stehende

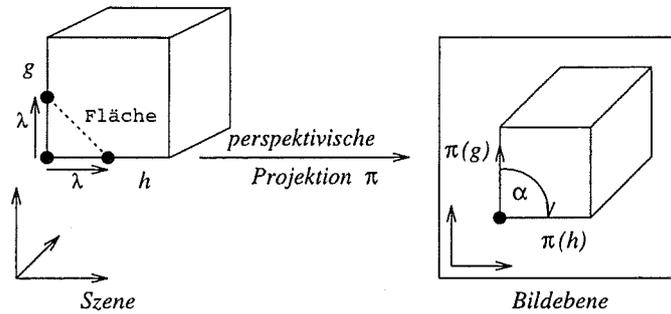


Abbildung 4.8: Flächentest benachbarter Kanten und Winkeltest benachbarter Linien: Zur Überprüfung der korrekten Ausrichtung verdeckender Kanten zu geeigneten Nachbarkanten wird ein Flächentest bzgl. dreidimensionaler Szenenkanten (linker Teil der Abb.) mit einem Winkeltest bzgl. zweidimensionaler Bildkanten kombiniert (rechter Teil der Abb.).

Bildebene projiziert. Eine zufällige Ausrichtung einer verdeckenden Linie  $\pi(g)$  wird in Bezug auf eine geeignete Nachbarlinie  $\pi(h)$  überprüft. Die Ausrichtung von  $\pi(g)$  wird gemäß Abb. 4.8 nach folgender Regel ermittelt.

**Regel 4.2 (Ausrichtung verdeckender Linien)** *Vertausche die zufällige Ausrichtung einer verdeckenden Linie  $\pi(g)$ , wenn folgende Bedingungen nach Abb. 4.8 nicht erfüllt sind:*

1. *Zwischen der 3D-Kante  $g$  und der Nachbarkante  $h$  existiert eine planare Fläche.*
2. *Die Nachbarlinie  $\pi(h)$  liegt in der Bildebene der perspektivischen Projektion rechts der verdeckenden Linie  $\pi(g)$ .*

Nach der Regel muß also ein Flächentest in Bezug auf eine geeignete Nachbarkante und ein Winkeltest in Bezug auf die entsprechende Nachbarlinie durchgeführt werden. Wenn die Nachbarlinie ebenfalls verdeckend ist, und in dem gemeinsamen Endpunkt drei Linien zusammentreffen, so ist die Nachbarlinie nicht geeignet. In diesem Fall fällt der Flächentest auf jeden Fall negativ aus. Der Zusammenhang wird in Abb. 4.9 verdeutlicht. Sind keine geeigneten Nachbarlinien vorhanden, kann das Verfahren speziell für diesen Fall um das Prüflinienverfahren von Wei [Wei 89] ergänzt werden.

Der Flächentest zwischen der verdeckenden Kante und einer geeigneten benachbarten Kante wird entlang einer 3D-Strecke durchgeführt. Die 3D-Prüfstrecke liegt zwischen Punkten auf den beiden Kanten, die einen gewissen Betrag  $\lambda$  von dem gemeinsamen Punkt entfernt liegen (siehe Abb. 4.8). Der Mindestbetrag  $\lambda$  ist dabei so groß wie die Mindestlänge approximierter Kanten gewählt. Dadurch ist gewährleistet, daß der Flächentest auf allen Kanten in diesem Stadium durchführbar ist. Der Flächentest arbeitet wie folgt:

**Algorithmus 4.4 (Flächentest)** *Eine Fläche ist vorhanden, wenn die Abstände zwischen äquidistanten Prüfstreckenpunkten  $\mathbf{p}$  und den Szenenkoordinaten  $\mathbf{p}'$  der korrespondierenden Bildpunkte im Durchschnitt unter einem Schwellwert liegen. Der durchschnittliche Abstand wird nach folgendem Algorithmus berechnet:*

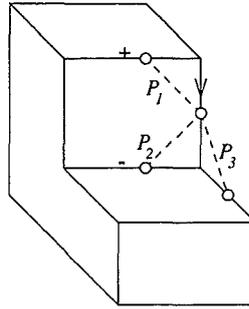


Abbildung 4.9: Durchführung des Flächentests nur zu geeigneten Nachbarkanten: Nur die 3D-Prüfstrecken  $P_1$  und  $P_2$  sind für den Flächentest geeignet.

```

begin
  durchschnittl_Abstand  $\leftarrow$  0;
   $\delta \leftarrow \frac{1}{max}$ 
  for  $i \leftarrow 0$  to  $max$  do
    (* nächster 3D-Prüfpunkt entlang der 3D-Prüfstrecke *)
     $\mathbf{p} \leftarrow \mathbf{p}_h + i \cdot \delta(\mathbf{p}_g - \mathbf{p}_h)$ ;
    (* zu 3D-Prüfpunkt im Abstandsbild korrespondierende *)
    (* Kugelkoordinate ermitteln *)
     $(u, v) \leftarrow Bildkoordinate(\mathbf{p})$ ;
     $(\phi_u, \theta_v, r(\phi_u, \theta_v)) \leftarrow Kugelkoordinate(u, v)$ ;
    (* Kugelkoordinate in Szenenkoordinate umrechnen *)
     $\mathbf{p}' \leftarrow Szenenkoordinate(\phi_u, \theta_v, r)$ ;
    (* durchschnittlichen Abstand fortschreiben *)
     $durchschnittl\_Abstand \leftarrow durchschnittl\_Abstand + \delta \|\mathbf{p} - \mathbf{p}'\|$ ;
  od;
end

```

Die Nachbarlinie liegt dann rechts der verdeckenden Linie, wenn der Winkel  $\alpha$  zwischen g- und h-Linie im Uhrzeigersinn kleiner als  $180^\circ$  ist. Falls eine verdeckende Kante keine Nachbarkanten besitzt, also isoliert im Raum liegt, kann ihre Ausrichtung vernachlässigt werden, da die Ausrichtung hier nur zur Ermittlung von Linienverbindungen benötigt wird.

Wird jeder Vertex durch einen Index identifiziert, dann sei eine Kante per Konvention durch ein Indexpaar so beschrieben, daß der kleinere Index an erster, der größere Index an zweiter Stelle steht. Jede verdeckende Kante besitzt dann statt des Attributes  $\pm$  je nach Ausrichtung eines der beiden Kantenattribute  $\leftarrow$  oder  $\rightarrow$ . Die ausgerichteten Kanten der Testszene sind in Abb. 4.10 gezeigt.

### 4.3 Attributierung von Vertizes

Gruppierte und ausgerichtete Kanten, die in einem gemeinsamen Vertex zusammentreffen, lassen sich durch die in der perspektivischen Projektion entsprechenden Linienverbindungen charakterisieren. In [Sugihara 79] werden Linienverbindungen nach der Linienanzahl, der Linienmarkierung und den zwischen Linien eingeschlossenen Winkeln charakterisiert. Linienverbindungen mit drei Linien können anhand

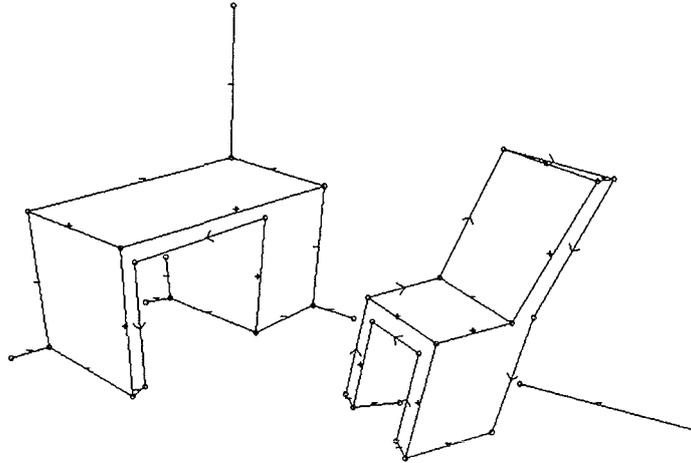


Abbildung 4.10: Ausgerichtete verdeckende Kanten der Testszene

der drei eingeschlossenen Winkel gemäß Abb. 4.11 in verschiedene Klassen eingeteilt werden. Im Gegensatz dazu werden in [Godin & Levine 89] Linienverbindungen vollständig im dreidimensionalen Raum ermittelt. Das Verfahren beruht auf der Idee, das dreidimensionale Modell, das zur vollständigen Aufzählung von Linienverbindungen dient, direkt auf 3D-Kanten anzuwenden, um zu bestimmen, welche Linienverbindungen die entsprechenden Kanten in der Bildebene bilden.

Die Aufzählung von Linienverbindungen geht grob so vor, daß der Raum entsprechend den Polyederflächen eines trihedralen Vertex in drei Ebenen aufgeteilt wird. Dann werden der Reihe nach 1, 3, 5 und 7 Oktanten mit Material aufgefüllt und von jedem freien Oktanten aus betrachtet, jedoch nicht von einem Blickpunkt innerhalb der Ebenen. Durch das systematische Betrachten verschieden gefüllter Oktanten von verschiedenen nicht-gefüllten Oktanten kann man alle Linienverbindungen aufzählen. Godin und Levine verwenden die dreidimensionale Struktur des Oktantenmodells direkt, um anhand von 3D-Winkeln zwischen Kanten und der Lage von Kanten in Bezug auf Oktanten die korrekte Linienverbindung abzuleiten. Vorteilhaft ist, daß Objekte nicht in allgemeiner Lage betrachtet werden müssen. Das verwendete Oktantenmodell berücksichtigt allerdings keine Polyederberührungen wie das Modell von Sugihara.

Da die hier betrachteten Szenen aus trihedralen Polyedern bestehen, die sich berühren dürfen, und zusätzlich die Bedingungen aus Abschnitt 2.4 als erfüllt angenommen werden, kann jeder Linienverbindung einer von 24 Indizes zugeordnet werden, die sich durch die Anpassung und Erweiterung des Modells von Sugihara nach Abschnitt 2.4.2 ergeben. Da eine Linienverbindung von der Ansicht der Polyederflächen in einem Vertex abhängig ist, wird der Linienverbindungsindex als Vertextyp verwendet.

Die verschiedenen Linienverbindungen wie in Abb. 2.12 werden anhand der Linienanzahl, der Linientypen und der Winkel zwischen Linien nach den Bedingungen 2.4, 2.5 und 2.6 unterschieden. Da gerade Kanten in den hier betrachteten Laser-radarbildern i.a. nicht als gerade Linien abgebildet werden, können die 2D-Winkel nicht direkt in Abstandsbildern berechnet werden. Zur 2D-Winkelbestimmung werden die 3D-Kanten daher durch perspektivische Projektion in eine Strichzeichnung

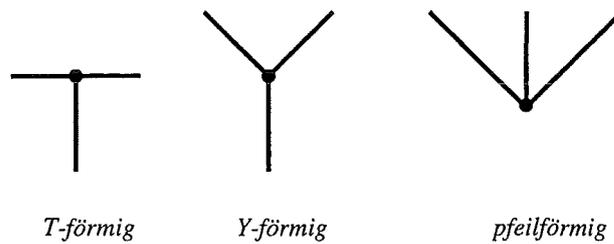


Abbildung 4.11: Klassifikation von Linienverbindungen mit drei Linien anhand der drei eingeschlossenen Winkel: Bei „T“-Verbindungen ist genau ein Winkel gleich  $180^\circ$ , bei „Y“-Verbindungen sind alle drei Winkel kleiner als  $180^\circ$  und bei pfeilförmigen Verbindungen ist ein Winkel größer und zwei Winkel sind kleiner als  $180^\circ$ .

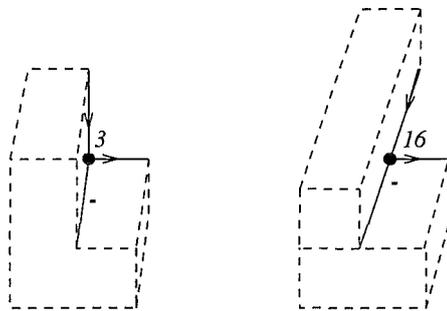


Abbildung 4.12: Problemfall ähnlicher 2D-Winkel, aber unterschiedlicher 3D-Konstellationen bei der Ermittlung des Indexes einer Linienverbindung: In beiden abgebildeten Linienverbindungen ist der 2D-Winkel zwischen der einlaufenden verdeckenden Linie und der konkaven Linie nahe  $180^\circ$ . Es ist aufgrund des 2D-Winkels zwischen Linien nicht zuverlässig zu entscheiden, ob die entsprechenden 3D-Kanten ebenfalls nahezu einen  $180^\circ$ -Winkel einschließen (Linienverbindung Nr. 16) oder, ob der 2D-Winkel lediglich infolge der Projektion entstanden ist (Linienverbindung Nr. 3). Daher wird der 2D-Winkeltest auf  $180^\circ$  zwischen Linien durch einen 3D-Winkeltest auf  $180^\circ$  zwischen 3D-Kanten ersetzt.

überführt. Wie im Falle der Ausrichtung verdeckender Linien dient die Bildebene der perspektivischen Projektion neben dem Abstandsbild als zweiter Bildbereich.

Um „T“-förmige Linienverbindungen robuster von anderen unterscheiden zu können, wird der Test eines Winkels auf  $180^\circ$  nicht zwischen Linien, sondern zwischen Kanten durchgeführt. Die Notwendigkeit dieses Winkeltests kommt in Abb. 4.12 zum Ausdruck. Damit ist der Ablauf zur Bestimmung von Linienverbindungen wie folgt.

**Algorithmus 4.5 (Index von Linienverbindungen)** *Eingabe sind die Linienanzahl  $n$ ,  $n$  Marken und  $n$  Linien, die nach ein- und auslaufenden Linien unterschieden werden. Ausgabe ist der Index der Linienverbindung nach Abb. 2.12 aus Kapitel 2.*

*Prüfe nur anhand der Linienanzahl und der zugehörigen Marken, in welche der folgenden Klassen die eingegebene Linienverbindung fällt, und führe die der Klasse zugeordnete Anweisung durch. Fällt die Linienverbindung in keine der folgenden*

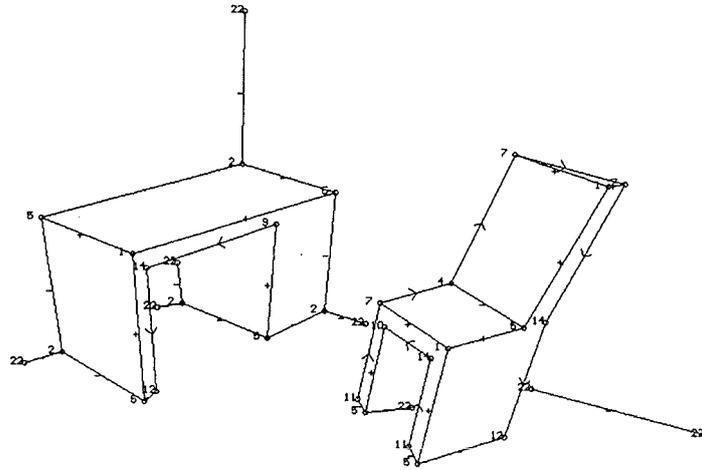


Abbildung 4.13: Vertices der Testszene mit Vertextypen: In der perspektivischen Projektion der Szenenkantenstruktur ist jeder Vertex mit dem ermittelten Vertextyp markiert.

*Klassen, gebe den Index 25 zurück, um eine unmögliche Linienverbindung zu kennzeichnen.*

- {1, 2, 5, 6, 9, 10, 11, 12, 21, 22, 23, 24} : Ermittle den Linienverbindungsindex anhand der eindeutigen Kombination aus Linienmarkierung und Linienanzahl.*
- {13, 14} : Ist der 2D-Winkel der auslaufenden bzgl. der einlaufenden Linie entgegen dem Uhrzeigersinn größer als  $180^\circ$ , gebe 13 zurück, sonst 14.*
- {4, 7} : Ist das Kriterium für „Y“-Linienverbindungen nach Abb. 4.11 erfüllt, dann gebe 4, sonst 7 zurück.*
- {3, 8, 15, 16} : Bildet die einlaufende verdeckende Kante mit der konkaven Kante einen 3D-Winkel von  $180^\circ$  dann gebe 16, sonst 15 zurück. Sonst prüfe folgende Bedingung: Ist der 2D-Winkel der auslaufenden bzgl. der einlaufenden Linie entgegen dem Uhrzeigersinn größer als  $180^\circ$ , gebe 8 zurück, sonst 3.*
- {17, 18, 19, 20} : Ist der 3D-Winkel zwischen einer verdeckenden und konkaven Kante gleich  $180^\circ$ , so unterscheide anhand der Linienmarkierung zwischen Linienverbindungen 19 und 20, sonst unterscheide anhand der Linienmarkierung zwischen Linienverbindungen 17 und 18.*

Die resultierende Strichzeichnung der Testszene mit korrekt ausgerichteten Kanten und durch Indizes gekennzeichnete Linienverbindungen zeigt Abb. 4.13.

In dem Modell von Sugihara wird jedoch eine Einschränkung (Bedingung 2.2) bei der Wahl des Blickpunktes gemacht, die im Kontext der Navigation als restriktiv anzusehen ist. Eine Lösung besteht darin, Linienverbindungen in Anlehnung an Godin und Levine ebenfalls direkt in dem dreidimensionalen Oktandenmodell zu bestimmen, wodurch die Bedingung bzgl. der allgemeinen Lage abgeschwächt wird.

## 4.4 Diskussion der Genauigkeit der Szenenkantenstruktur

Das beschriebene Verfahren zur Extraktion der Szenenkantenstruktur erhebt nicht den Anspruch, eine Szene anhand eines Abstandsbildes möglichst exakt zu rekonstruieren. Die Kantenstruktur soll vielmehr eine Struktur darstellen, die es erlaubt, Objektmodelle anhand von Kanten und anhand von Beziehungen zwischen Kanten zu erkennen. Da Kantenlängen und Winkel zwischen Kanten wichtige Merkmale zur Erkennung von Objekten darstellen, ist dennoch wichtig, in welcher Genauigkeit diese Merkmale extrahiert werden. Wünschenswert wäre eine quantitative Analyse, die die Genauigkeit der Merkmale in Abhängigkeit von anderen Einflußgrößen analytisch beschreibt. Mögliche Einflußgrößen sind beispielsweise der Abstand zu Oberflächen, die Neigung von Oberflächen, der Feinheitsgrad der Abtastung, Fenstergröße der Glättungsoperation, die Fenstergröße der Krümmungsberechnungen und die Auswirkung der Abtastung in Kugelkoordinaten auf die Genauigkeit in Szenenkoordinaten. Bevor solche quantitativen Abhängigkeiten untersucht werden, müssen jedoch erst qualitative Eigenschaften extrahierter Szenengraphen charakterisiert werden. Erst dann lassen sich Aussagen über die quantitative Genauigkeit einordnen. Diese qualitativen Eigenschaften lassen sich meist nicht allein auf die Extraktion von Bildbereichshinweisen *oder* Szenenbereichshinweisen, sondern auf Bild- *und* Szenenbereichshinweise zurückführen. Probleme, die die Genauigkeit von Szenengraphen beeinflussen, lassen sich grob in fünf Klassen einteilen:

- ungenaue Positionen von Kantenendpunkten
- unvollständige verdeckende Kanten
- unechte Kanten, fehlende Kanten und unmögliche Linienverbindungen
- Verkürzte und verlängerte Kanten
- verschobene Position verdeckender Kanten

### Ungenaue Positionen von Kantenendpunkten

Ausgangspunkt für die Kantendetektion in einem Abstandsbild ist die Berechnung zweier Krümmungsbilder, die durch Schwellwertbildung in entsprechende Krümmungsvorzeichenbilder umgesetzt werden. Das zweite der beiden Krümmungsvorzeichenbilder dient der Extraktion konkaver und verdeckender Bildkantenpunkte. Eine Glättungsoperation führt allerdings dazu, daß insbesondere konkave Szenenkanten im Krümmungsvorzeichenbild als breite Gebiete erscheinen. Konkave Kantenpunkte werden aus diesen Kantengebieten durch Verdünnung bestimmt. Unter der Annahme, daß die Bildpunkte mit höchstem Krümmungsbetrag in der Regionenmitte liegen, liefert die Verdünnung eine gute Schätzung der Position konkaver Kantenpunkte. Berühren sich aber die Kantengebiete mehrerer konkaver Kanten, entstehen flächig ausgedehnte Kantengebiete. Da die Verdünnung keine Krümmungsinformation, sondern nur das dreiwertige Vorzeichenbild nutzt, können die ermittelten konkaven Bildkantenpunkte von den Bildpunkten maximaler negativer Krümmung stark abweichen. Dies führt zu Punkteketten und damit zu Kanten, deren Enden verfälschte Positionen aufweisen. Dies wiederum wirkt sich so aus, daß der Winkel zwischen Kanten verfälscht sein kann und im Extremfall zum Beispiel eine „Y“-förmige als pfeilförmige Linienverbindung klassifiziert wird.

### Unvollständige verdeckende Kanten

Die unvollständige Detektion verdeckender Kanten führt ebenfalls zu verfälschten Kantenendpunkten, jedoch liegt eine andere Entstehungsweise vor. In dem in Kapitel 3 beschriebenen Verfahren werden bereits detektierte verdeckende Bildpunkte zur korrekten Lokalisierung konkaver Bildkantenpunkte benötigt. Durch die unvollständige Detektion verdeckender Bildkantenpunkte werden daher komplementär andere Kantenpunkte fälschlicherweise als konkav interpretiert.

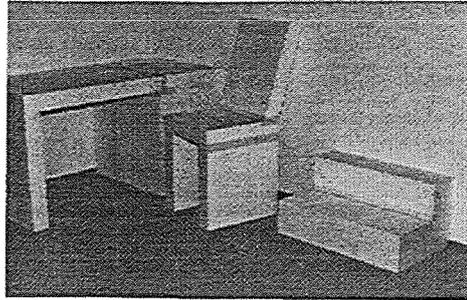
Solche als konkav klassifizierte Kantenpunkte entstehen an den Enden von verdeckenden Kanten, wo Abstandsunterschiede zu Nachbarflächen immer geringer werden. Solche Punkte mit negativem Vorzeichen repräsentieren aber *verdeckte* Bildpunkte, d.h. Punkte einer Sprungkante, die auf der Seite des Hintergrundes liegen. Da diese verdeckten Punkte in Punktketten nur kurze Kettenstücke beitragen, liegen entsprechende Endpunkte auch nicht weit im Hintergrund. In der Kantenapproximation wird das Kettenstück aufgrund der geringen Länge nicht aufgetrennt. Folglich liegt ein Kettenendpunkt und damit ein Kantenendpunkt im Hintergrund, wodurch die Lage der Kante verfälscht ist. Um eine solche Kante gruppieren zu können, benötigt man hohe Schwellwerte für den Gruppierungsabstand zwischen Kanten, so daß aber auch der den gruppierten Kanten gemeinsame Endpunkt eine hohe Ungenauigkeit besitzt. Eine mögliche Verbesserung besteht darin, in der Approximation einer Kante zunächst eine Gerade durch lineare Regression zu bestimmen und erst dann die Kantenendpunkte durch Fällen des Lotes der Kettenendpunkte auf die Gerade zu bestimmen.

Um den Gruppierungsfehler zu minimieren, ist es zumindest bei der Gruppierung nicht-kollinearere Kanten möglich, Flächeninformation miteinzubeziehen. Anstatt den Punkt der gruppierten Kanten anhand zweier Kanten zu berechnen, kann der Punkt durch Schnitt einer Kante mit der Ebene berechnet werden, die durch zwei bereits gruppierte Kanten aufgespannt wird. Zur Verifikation der Lage von Kanten wäre auch die Überprüfung direkt im Abstandsbild möglich, indem die einer 3D-Kante entsprechenden 2D-Bildpunkte bzgl. des Abstandes mit den korrespondierenden Punkten des Abstandsbildes verglichen werden. Da aus gruppierten Kanten in der späteren Verarbeitung Linienverbindungen als Merkmale ermittelt werden, sind gruppierte Kanten mit leicht verschobenen Kantenendpunkten wichtiger als nicht-gruppierte Kanten mit genaueren Endpunktpositionen.

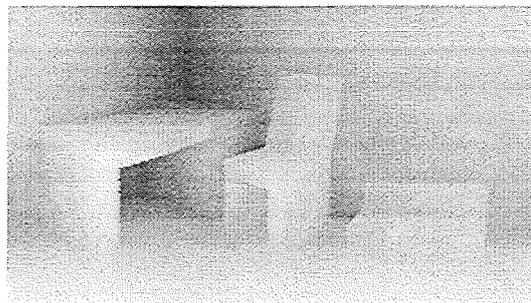
### Unechte Kanten, fehlende Kanten und unmögliche Linienverbindungen

Je spitzer die Aufsicht auf Oberflächen, desto weniger Bildkantenpunkte werden abgetastet. Bildkantenpunkte resultieren dann in kurzen Bild- und Szenenpunktketten, die ein zu großes Verhältnis der Kanten- zur Kettenlänge aufweisen können. Ist dieses Verhältnis zu groß, wird die Punktkette nicht für die Kantenapproximation akzeptiert. Das heißt, in dem Szenengraphen fehlt diese Kante. Ein Beispiel ist der Szenengraph der zweiten Testszene in Abb. 4.14.

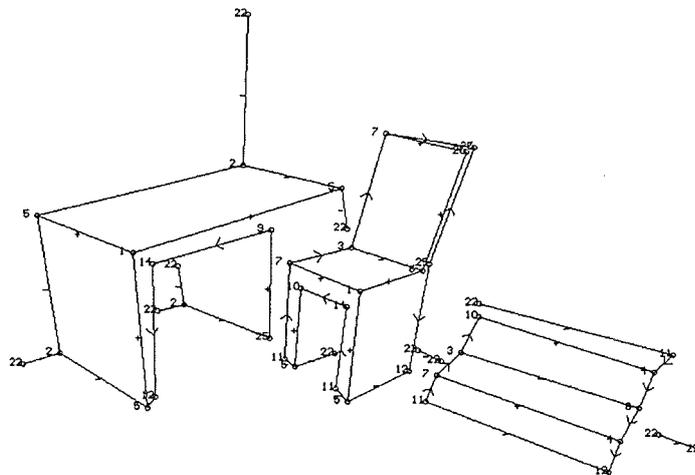
Eine Szenenkante des stufenförmigen Objektes fehlt, wodurch die mögliche, aber nicht korrekte Linienverbindung 10 entsteht. Eine Behebung des Problems besteht in Verfahren der Flächenrekonstruktion, die die Kanteninformation aus der Flächenberandung gewinnen, und in einer anschließenden Prüfung auf fehlende, die Fläche berandende Kanten. Weiter unten sei kurz das Verfahren von [Herman 85] zur kantenbasierten Flächenrekonstruktion skizziert. Das Verfahren liefert aus Strichzeichnungen Polygonzüge, die jeweils zu dem Rand einer einzigen Fläche gehören. Aus einem Polygonzug lassen sich direkt die notwendigen Flächenparameter ableiten. Ist der Polygonzug nicht geschlossen, kann mit Hilfe der Flächeninformation und



(a) Photographie einer zweiten Testszene



(b) Abstandsbild der zweiten realen Testszene



(c) Markierte Strichzeichnung

Abbildung 4.14: Photographie, Abstandsbild und markierte Strichzeichnung der zweiten realen Testszene: Eine Szenenkante des stufenförmigen Objektes fehlt, wodurch die mögliche, aber nicht korrekte Linienverbindung 10 entsteht. Eine andere Kante fehlt an der rechten oberen Seite der Stuhllehne. Weil dadurch eine passende Bezugskante fehlt, ist eine verdeckende Kante auf der rechten, hinteren Seite der Stuhllehne falsch ausgerichtet. Folglich entstehen zwei unmögliche Linienverbindungen, die daher mit dem Index 25 markiert sind.

der Linienverbindungen der freien Enden des Polygonzugs eine Hypothese über die noch fehlende Kante und den Kantentyp erstellt werden. Die Hypothese kann dann gezielt im Abstandsbild zum Beispiel mittels einer Prüflinie überprüft werden.

Eine andere Kante fehlt an der rechten oberen Seite der Stuhllehne, wodurch zwei unmögliche Linienverbindungen entstehen, die daher mit dem Index 25 markiert sind. Der entgegengesetzte Fall fehlender Kanten ist der Fall unechter Kanten, die durch die Bildauswertung entstehen. Wenn Bildpunkte konvexer und konkaver Kanten direkt benachbart sind, können aufgrund des entstehenden Vorzeichenwechsels fälschlicherweise auch längere verdeckende Punktketten detektiert werden. Dies ist beispielsweise bei einer konvexen und konkaven Kante der Fall, die parallel angeordnet sind und sehr eng zusammenliegen. Eine Behebung des Problems liegt darin, daß der Vorzeichenwechsel nur dann als Merkmal für verdeckende Kantentypen gilt, wenn auch eine Abstandsunstetigkeit vorliegt, die zum Beispiel mit einem Sobeloperator ermittelt werden kann.

### *Kantenbasierte Flächenrekonstruktion*

Herman [Herman 85] verwendet markierte Strichzeichnungen, die aus Abstandsbildern polyedrischer, auch nicht-trihedralearer Szenen generiert sind. Obwohl bereits dreidimensionale Kanten extrahiert sind, werden Linien im Bildbereich verwendet, da durch die Markierung von Linien die relative Lage von Flächen implizit kodiert ist.

Eine Liste von Linien, die eine Fläche beranden, wird durch ein bestimmtes Durchlaufen der Linien der Strichzeichnung ausgehend von einer Startlinie erzeugt. Startlinien und die nächste zu durchlaufende Linie sind abhängig von dem Linientyp und von dem Winkel zwischen benachbarten Linien.

Da das Verfahren von Herman für allgemeine polyedrische Szenen arbeitet, ist es hier auch für die Rekonstruktion von Polyederflächen geeignet. Als Eingabe dient eine zweidimensionale Strichzeichnung einer polyedrischen Szene, wobei Linien mit dem Linientyp und Linienverbindungen mit dem Linienverbindungsindex markiert sind.

Zu einer Fläche gehörende Kanten werden durch Durchlaufen der Strichzeichnung ermittelt. Betrachtet man die Strichzeichnung als Graphen mit den Linien als Knoten und den Linienverbindungen als Bögen, so lassen sich Linien einer Fläche durch Durchlaufen eines Pfades in dem Graphen beschreiben, wobei jede durchlaufene Linie eine begrenzende Linie einer Region darstellt.

Verdeckende Linien werden dabei nur einmal durchlaufen, da nur auf einer Seite der Linie eine angrenzende Fläche abgebildet ist. Konvexe und konkave Linien werden hingegen genau zweimal und in entgegengesetzter Richtung durchlaufen: je einmal pro angrenzende Fläche. Man kann dabei vollständig und teilweise sichtbare Flächen unterscheiden. Vollständig sichtbare Flächen werden durch vollständiges Durchlaufen von Linien gefunden, so daß der Vorgang mit jeder Linie begonnen werden kann. Der Algorithmus besteht aus drei Komponenten: (1) Auswählen von Startlinien, (2) Durchlaufen der Strichzeichnung und (3) Prüfen einer Terminierungsbedingung.

Im Falle teilweise sichtbarer Flächen werden Startkanten in einer bestimmten Reihenfolge gewählt, so daß auf jeden Fall alle sichtbaren Linien einer Fläche durchlaufen werden. Ausgehend von Startlinien wird die nächste Linie durchlaufen, die in Laufrichtung benachbart ist und entgegen dem Uhrzeigersinn den kleinsten Winkel besitzt. Dieser Vorgang wird wiederholt, bis bestimmte Terminierungsbedingungen erfüllt sind, die auf eine andere Fläche hindeuten.

Das Verfahren von Herman besitzt allerdings den Nachteil, daß mehrere Polygonketten für eine einzige Fläche ausgegeben werden, ohne zu wissen, ob sie zu dieser Fläche gehören. Daher muß in einem nachgeschalteten Verarbeitungsschritt für jeden Polygonzug die Zugehörigkeit durch einen Test gegen bereits extrahierte Ebenen ermittelt werden.

#### Verkürzte und verlängerte Kanten

Durch die Lokalisierung verdeckender Kantenpunkte werden diese in Richtung des konvexen Anteils des Vorzeichenwechsels einer Sprungkante verschoben. Dadurch werden Kanten in konvexen Ecken verkürzt und in konkaven Ecken verlängert. Der Grad der Verkürzung und der Verlängerung ist dabei von der Neigung der entsprechenden Oberfläche bezogen auf den Sensor abhängig. Wenn von der Szenenfläche in Verschiebungsrichtung weniger als eine Mindestanzahl von Punkten abgebildet ist, werden verdeckende Kantenpunkte auf Nachbarflächen lokalisiert. Der Fehler wäre vor allem dann zu groß, wenn diese Fläche entlang einer benachbarten verdeckenden Kante im Hintergrund liegt. Das heißt, zwischen zwei verdeckenden Kanten muß eine Mindestanzahl von Abtastpunkten liegen, damit das beschriebene Verfahren korrekt arbeitet.

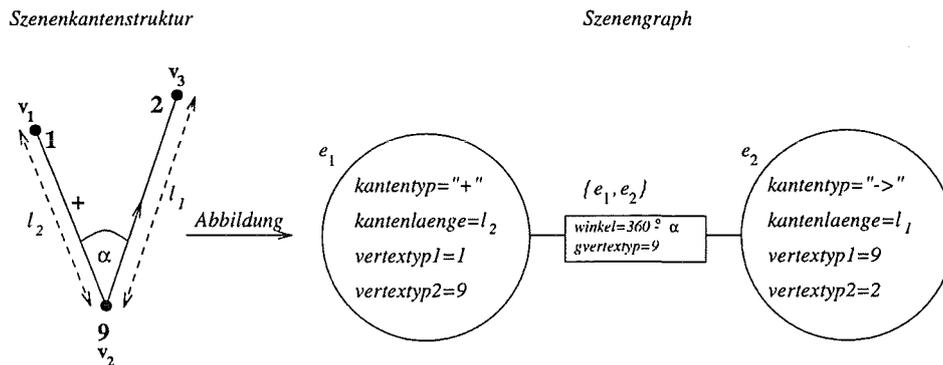
#### Position verdeckender Kanten

Eine weitere Folge der Lokalisierung verdeckender Kantenpunkte ist eine Verschiebung verdeckender Kanten um einen bestimmten Betrag senkrecht zu der Ausrichtung der verdeckenden Kante. Der Betrag beträgt zwar eine konstante Anzahl von Bildpunkten, der jedoch in Abhängigkeit von der Neigung von Szenenflächen unterschiedliche räumliche Verschiebungen verursachen kann. Bei schräger Aufsicht von Oberflächen ist die Länge der entsprechenden räumlichen Verschiebung hoch, bei senkrechter Aufsicht hingegen am geringsten.

## 4.5 Repräsentation von Szenenbereichshinweisen und Modellen

Um Korrespondenzen zwischen Szenenbereichshinweisen und Modellen zu finden, muß die Repräsentation beider Strukturen geklärt werden. Dazu werden zunächst folgende Begriffe gegeneinander abgegrenzt: (1) Die *Szenenkantenstruktur* stellt die aus dem Abstandsbild generierte Kantenstruktur dar, die als perspektivische Projektion auch als *Strichzeichnung* bezeichnet werden kann. (2) Der *Szenengraph* ist eine geeignete Repräsentation der Szenenkantenstruktur zum Zwecke der Objekterkennung. Charakteristisch für den Szenengraphen ist, daß Kanten als Knoten repräsentiert werden und die Linienverbindung des gemeinsamen Endpunktes zweier Kanten als Bogenattribut repräsentiert wird. Sowohl Szenenkantenstruktur als auch Szenengraph stellen im Sinne von Kanade [Kanade 78] Szenenbereichshinweise dar. (3) Eine *Objektkantenstruktur* ist hier eine zu einer Objektansicht manuell erstellte Kantenstruktur, die ein Hilfsmittel zur Konstruktion, Visualisierung und Kontrolle darstellt. (4) Der *Objektgraph* ist eine Repräsentation einer Objektkantenstruktur für den Zweck der Objekterkennung. (5) Ein *Modell* oder *Objektmodell* besteht aus allen Objektgraphen eines Objektes. Zunächst werden Szenengraphen genauer spezifiziert.

Da der Szenengraph zur Erkennung benötigt wird, muß er aus der Szenenkantenstruktur generiert werden. Dieser Prozeß wird in Abbildung 4.15 verdeutlicht. Im



Elemente der Szenenkantenstruktur:

Vertizes

$$\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\} \subset \mathbb{R}^3$$

Kanten

$$\{(\mathbf{v}_1, \mathbf{v}_2), (\mathbf{v}_2, \mathbf{v}_3)\}$$

benachbarte Kanten

$$\{ \{(\mathbf{v}_1, \mathbf{v}_2), (\mathbf{v}_2, \mathbf{v}_3)\} \}$$

Elemente des Szenengraphen:

Knoten

$$\{e_1, e_2\} = \{(1, 2), (2, 3)\}$$

Bögen

$$\{\{e_1, e_2\}\}$$

Knotenattribute

$$\text{z.B. } kantentyp(e_1) = +$$

$$\text{z.B. } kantenlaenge(e_1) = l_2$$

$$\text{z.B. } vertextyp1(e_1) = 1$$

$$winkel(\{e_1, e_2\}) = 360^\circ - \alpha$$

Bogenattribute

$$gvertextyp(\{e_1, e_2\}) = 9$$

Abbildung 4.15: Beispielhafte Abbildung einer Szenenkantenstruktur in einen Szenengraphen: Im linken, oberen Teil der Abbildung ist eine Szenenkantenstruktur, im rechten oberen Teil ist der korrespondierende Szenengraph ersichtlich. Im unteren Teil der Abbildung sind die Bezeichner verwendeter Elemente genauer aufgeschlüsselt (Erklärung siehe Text).

linken, oberen Teil der Abbildung sind zwei Kanten einer Szenenkantenstruktur abgebildet. Die dreidimensionalen Koordinaten der Endpunkte werden mit  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ , die Kantenlängen mit  $l_1, l_2$  und der eingeschlossene Winkel mit  $\alpha$  bezeichnet. Die Vertextypen sind durch die fett gedruckten Zahlen **1,2,17** gekennzeichnet. Im rechten, oberen Teil der Abbildung ist der korrespondierende Szenengraph abgebildet. Jeder Knoten des Graphen repräsentiert eine Szenenkante, die durch ein Indexpaar  $(i, j)$  ihrer Endpunkte  $\mathbf{v}_i, \mathbf{v}_j$  beschrieben wird. Jeder Knoten besitzt vier Attribute (innerhalb des Ovals). Die hier verwendeten Kantenattribute sind der Kantentyp, die Kantenlänge und die Vertextypen der beiden Kantenendpunkte. Da eine Linienverbindung angibt, wie sich Polyederflächen in einem Vertex schneiden, wird der entsprechende Linienverbindungsindex im folgenden auch als Vertextyp bezeichnet.

Jedes Knotenpaar ist weiterhin über einen Bogen verbunden, wenn die korrespondierenden Szenenkanten einen gemeinsamen Endpunkt besitzen. Der Bogen besitzt zwei Attribute, die innerhalb des Rechtecks eingetragen sind. Bogenattribute sind der Winkel zwischen benachbarten Kanten und der Vertextyp des gemeinsamen Endpunktes. Die Knoten, Bögen und Attribute zusammen bilden einen attributierten Graphen [Barrow & Popplestone 71]. Attribute lassen sich nach diskreten oder symbolischen und kontinuierlichen oder numerischen unterscheiden. Kantenlänge und Winkel sind numerische, Kanten- und Vertextyp sind symbolische Attribute. Der Szenengraph ist damit wie folgt definiert:

#### Definition 4.1 (Szenengraph)

- Sei  $V \subset \mathbb{R}^3$  eine endliche Menge von Vertizes  $\{\mathbf{v}_1, \dots, \mathbf{v}_{n1}\}$  einer Szenenkantenstruktur und sei die Menge  $\{(\mathbf{v}_p, \mathbf{v}_q) \mid \mathbf{v}_p \text{ und } \mathbf{v}_q \text{ sind die beiden Endpunkte einer Szenenkante, } (\mathbf{v}_p, \mathbf{v}_q) \text{ ist ein geordnetes 2-Tupel, } p < q\}$  die Menge aller Szenenkanten.
- Dann heißt die Menge  $E = \{e_1, \dots, e_{n2}\} = \{(p, q) \mid (\mathbf{v}_p, \mathbf{v}_q) \text{ ist Szenenkante}\}$  die Knotenmenge des Szenengraphen.
- Ist weiterhin das Prädikat  $\text{nachbar}(\{e_i, e_j\}) \equiv [\exists p \in \mathbb{N} : p \in e_i \wedge p \in e_j]$  gegeben, so heißt  $P = \{\{e_i, e_j\} \mid e_i, e_j \in E, i \neq j, \text{nachbar}(\{e_i, e_j\})\}$  die Bogenmenge des Szenengraphen.
- Weiterhin seien die folgenden Attributierungsfunktionen gegeben, die Knoten und Bögen Attribute zuordnen:

Knotenattributierung	kantentyp : $E$	$\mapsto$	$\{+, -, \rightarrow, \leftarrow\}$
	laenge : $E$	$\mapsto$	$\mathbb{R}^+$
	vertextyp1 : $E$	$\mapsto$	$\{1, \dots, 24\}$
	vertextyp2 : $E$	$\mapsto$	$\{1, \dots, 24\}$
Bogenattributierung	gvertextyp : $P$	$\mapsto$	$\{1, \dots, 24\}$
	winkel : $P$	$\mapsto$	$[0^\circ, 360^\circ]$

Das Tupel  $(E_S, P_S, A_S)$  mit  $A = \{ \text{kantentyp, laenge, vertextyp1, vertextyp2, gvertextyp, winkel} \}$  heißt Szenengraph.

Die Attributierungsfunktion  $laenge(e_i)$  liefert trivialerweise den Abstand  $|\mathbf{v}_p - \mathbf{v}_q|$  Endpunkte  $\mathbf{v}_p$  und  $\mathbf{v}_q$  der dem Knoten  $e_i$  entsprechenden Szenenkante. Die Funktion  $gvertextyp(\{e_i, e_j\})$  liefert als Wert den Linienverbindungsindex des gemeinsamen Endpunktes derjenigen Szenenkanten, deren zugehörige Knoten über einen Bogen verbunden sind.

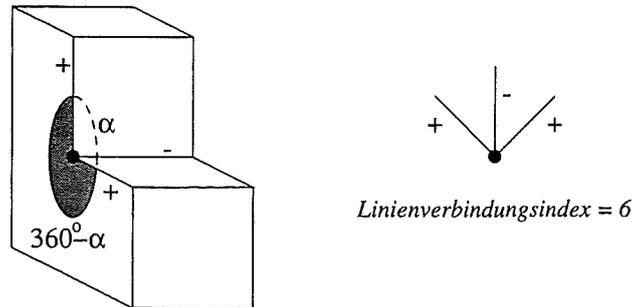


Abbildung 4.16: Berechnung des Winkelattributes mit Hilfe von Linienverbindungen: Aufgrund des Linienverbindungsindex Nr. 6 ist bekannt, daß zwischen den beiden Kanten kein Material liegt. Daher wird der zu dem eingeschlossenen Winkel  $\alpha$  zugehörige komplementäre Winkel  $360^\circ - \alpha$  verwendet.

Die Attributierungsfunktion  $winkel(\{e_i, e_j\})$  liefert einen speziellen 3D-Winkel. Der zwischen zwei 3D-Kanten  $e_i, e_j$  eingeschlossene Winkel  $\alpha(\{e_i, e_j\})$  ist stets kleiner als  $180^\circ$ . Mit Hilfe von Linienverbindungen läßt sich jedoch entscheiden, ob sich zwischen den entsprechenden Kanten „Material“ oder kein „Material“ befindet. Im letzten Fall wird der eingeschlossene Winkel von  $360^\circ$  abgezogen, so daß über das erweiterte Winkelintervall auch Information über die Flächenkonstellation in einem Vertex kodiert ist. Die Berechnung wird in Abb. 4.16 verdeutlicht. Ob zwischen zwei Kanten der komplementäre Winkel verwendet werden soll, kann man mit Hilfe des Typs des den Kanten gemeinsamen Vertex entscheiden. In Abhängigkeit von den beiden Kantentypen und dem Vertextyp lassen sich drei Fälle für die komplementäre Winkelberechnung unterscheiden, die sich formal durch Bedingungen ausdrücken lassen. Die Bedingungen können leicht anhand der Linienverbindungen in Abb. 2.12 überprüft werden. Die Attributierungsfunktion  $winkel(\{e_i, e_j\})$  wird demgemäß wie folgt aus dem eingeschlossenen Winkel  $\alpha(\{e_i, e_j\})$  berechnet, wenn  $t = gvertexyp(\{e_i, e_j\})$  der Vertextyp des gemeinsamen Endpunktes des Kantenpaares darstellt.

$$winkel(\{e_i, e_j\}) = \left\{ \begin{array}{l} 360^\circ - \alpha(\{e_i, e_j\}) \\ \text{für:} \\ \quad (* \text{ Bedingung 1 } *) \\ \quad t \in \{9, 10, 14\} \\ \quad \vee \\ \quad (* \text{ Bedingung 2 } *) \\ \quad t \in \{19, 20\} \wedge \\ \quad (kantentyp(e_i) \in \{\leftarrow, \rightarrow\} \wedge \\ \quad kantentyp(e_j) = + \vee \\ \quad kantentyp(e_j) \in \{\leftarrow, \rightarrow\} \wedge \\ \quad kantentyp(e_i) = +) \\ \quad \vee \\ \quad (* \text{ Bedingung 3 } *) \\ \quad t \in \{3, 6, 8, 15, 16\} \wedge \\ \quad kantentyp(e_i) = kantentyp(e_j) \\ \alpha(\{e_i, e_j\}), \quad \text{sonst} \end{array} \right.$$

Zur Repräsentation von Modellen wird die Technik der charakteristischen Ansichten

[Charkavaty & Freeman 82] verwendet. Charakteristische Ansichten werden durch eine Menge von Blickpunkten erzeugt, die topologisch identische Strichzeichnungen erzeugen. Charkavaty und Freeman geben ein hier geeignetes Beispiel für die sog. Linienverbindungsidentität an. Unter dieser Identität wird verstanden, daß in beiden topologisch identischen Strichzeichnungen die gleichen Linienverbindungen sichtbar sind und über gleich markierte Linien verbunden sind, während sich aber die Längen korrespondierender Linien unterscheiden können.

Die Repräsentation von Modellen als charakteristische Ansichten erscheint aus folgenden Gründen als geeignet. Da angenommen wird, daß Objekte in der Regel nicht von allen Seiten aus zugänglich sind, können sie nur von Blickpunkten eines eingeschränkten räumlichen Bereiches aus betrachtet werden. Es wird daher angenommen, daß Objekte nur in wenigen charakteristischen Ansichten in Abstandsbildern erscheinen können. Folglich kann jedes Objekt durch eine Menge seiner charakteristischen Ansichten repräsentiert werden.

Vorteilhaft für das Erkennungsverfahren ist weiterhin, daß einzelne charakteristische Ansichten als Objektgraphen repräsentiert werden können, die nahezu die gleiche Struktur wie Szenengraphen aufweisen. Da die Länge vollständig sichtbarer 3D-Kanten und der Winkel zwischen 3D-Kanten invariant bzgl. Transformationen der euklidischen Bewegungsgruppe [Lenz 90] ist, sind die numerischen Attribute invariant bzgl. Blickpunktänderungen. Die symbolischen Attribute Kanten- und Vertextyp sind nur bzgl. Transformationen zwischen Blickpunkten derselben Blickpunktklasse invariant.

Objektkantenstrukturen unterscheiden sich von Szenenkantenstrukturen in einem Punkt. Sowohl Vertextypen als auch Kantentypen der Objektkantenstruktur und damit auch des Objektgraphen werden danach festgelegt, wie zu erwarten ist, daß entsprechende Kanten als Objektinstanz in der Szene mit anderen Szenenflächen in Berührung stehen. Der Sachverhalt wird in Abb. 4.17 anhand der markierten Strichzeichnung einer Szenenkantenstruktur und einer Objektkantenstruktur verdeutlicht. Kanten in Objektkantenstrukturen, die in der Szene als teilweise sichtbar erwartet werden, werden im korrespondierenden Objektgraphen in dem verdeckten Endpunkt statt mit den Vertextypen 21-24 mit dem Vertextyp 0 versehen. Dadurch wird gekennzeichnet, daß zwar eine teilweise Verdeckung in der Szene erwartet wird, gleichzeitig aber die Originallänge einer Kante kenntlich gemacht wird. Damit ergibt sich folgende Definition für Objektgraphen in Anlehnung an die Definition des Szenengraphen.

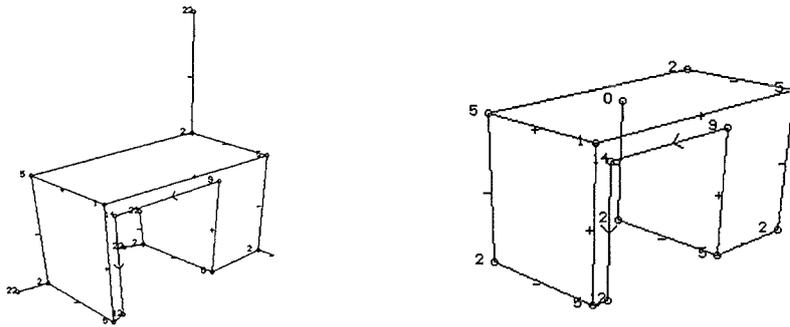
**Definition 4.2 (Objektgraph)** *Ein Objektgraph ist ein Graph  $G_O = (E_O, P_O, A_O)$ , wobei die Wertemenge für Linienverbindungsindizes um das Element 0 erweitert ist.*

Die hier verwendeten Objektkantenstrukturen in Form markierter Strichzeichnungen zeigt Abb. 4.18.

## 4.6 Zusammenfassung

Zusammenfassend werden Szenenbereichshinweise in fünf Schritten aus Bildbereichshinweisen extrahiert: Approximation von Kanten, Gruppierung von Kanten, Ausrichtung von Kanten, Attributierung von Vertizes und Aufbau des Szenengraphen.

In diesem Ansatz werden Linienverbindungen als Merkmale von Vertizes in einer bestimmten Ansicht verwendet. Da Linienverbindungen von der Markierung der inzidenten Linien abhängig sind, stellen sie Merkmale mit hohem Informationsgehalt



(a) Ausschnitt der markierten Strichzeichnung der Testszene: freie Endpunkte teilweise sichtbarer Linien sind mit dem Linienverbindungsindex 22 markiert.

(b) Strichzeichnung einer Objektkantenstruktur mit *erwarteten* Linienmarkierungen und Linienverbindungsindizes: Die teilweise sichtbare Linie besitzt einen Endpunkt, der mit dem Linienverbindungsindex 0 statt mit dem Index 22 markiert ist.

Abbildung 4.17: Markierte Strichzeichnung einer Szenenkantenstruktur und Strichzeichnung einer Objektkantenstruktur mit erwarteter Markierung von Linien und Linienverbindungen

dar. Die einzige dem Autor bekannte Repräsentation, die ebenfalls Linienverbindungen als Merkmale verwendet, ist der Kanten-Linienverbindungsgraph von [Godin & Levine 89]. Allerdings werden in diesem Ansatz auch Linienverbindungen berücksichtigt, die durch die Berührung von Polyedern entstehen. Im Gegensatz zu der Arbeit von Godin und Levine werden hier keine Gruppierungen zugelassen, die mehr als drei Kanten zusammenfassen. Hier werden nur diejenigen Kanten mittels einer geeigneten Gruppierungsreihenfolge gruppiert, die am wahrscheinlichsten einen Vertex bilden. Die Bildung valider Linienverbindungen hat höchste Priorität. Da Linienverbindungen aufgrund der ungleichförmigen Abtastung des Laserradarscanners nicht direkt im Abstandsbild ermittelt werden können, wird neben dem Abstandsbild ein zweiter Bildbereich verwendet. Durch die Einbindung der perspektivischen Projektion extrahierter Szenenkanten als zweiten Bildbereich wird es trotz der ungleichförmigen Abtastung ermöglicht, Winkel zwischen Linien und damit Linienverbindungen korrekt zu ermitteln. Dadurch können Linienverbindungen auch bei Verwendung von Laserradarscannern korrekt ermittelt werden, die nicht in Kugelkoordinaten abtasten, sondern eine andere ungleichförmige Abtastung aufweisen.

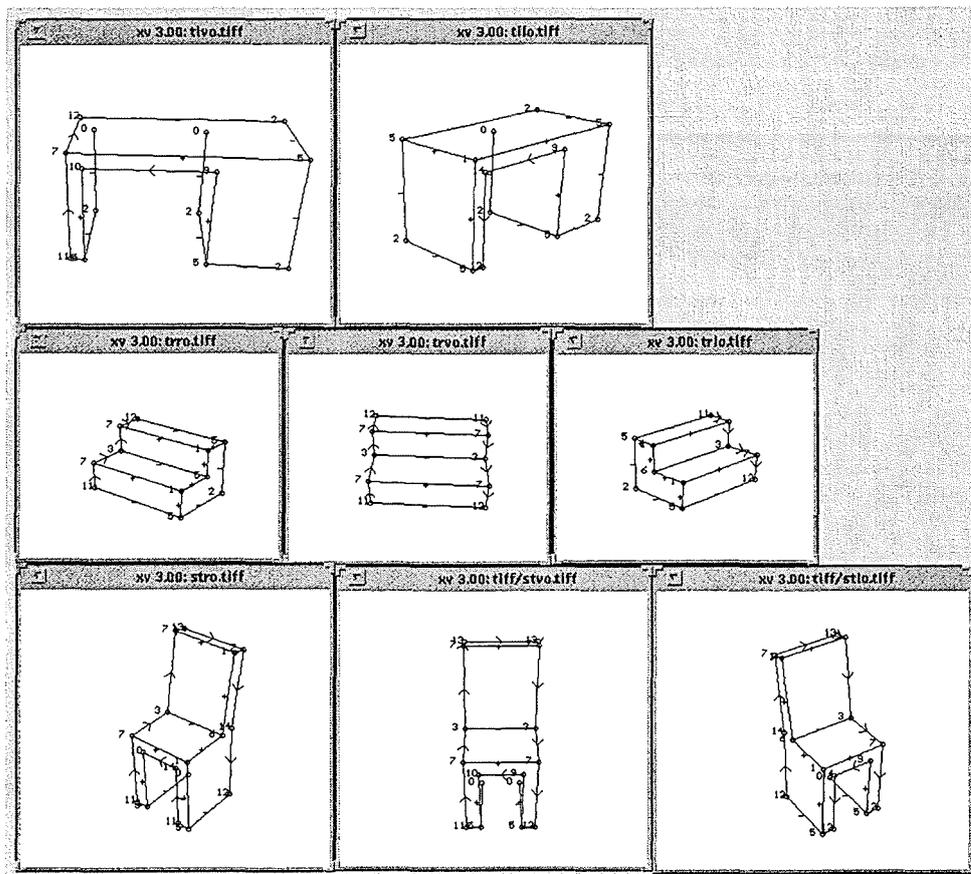


Abbildung 4.18: Markierte Strichzeichnungen der Objektkantenstrukturen verschiedener charakteristischer Ansichten



## Kapitel 5

# Parallelisierbare Objekterkennung

Um in Szenenbereichsstrukturen Objekte erkennen zu können, müssen Objekte in Form von Objektmodellen einer geeigneten Repräsentation vorliegen. Szenenbereichsstrukturen liegen hier als Szenengraph vor. Jedes Objektmodell besteht hier aus einer Menge charakteristischer Ansichten, wobei jede Ansicht einen Objektgraph darstellt. Um einen Objektgraphen in einem Szenengraphen zu erkennen, müssen Knoten des Objektgraphen korrespondierenden Knoten des Szenengraphen zugeordnet werden. Wenn aufgrund der ermittelten Korrespondenzen zu allen Knoten des Szenengraphen bekannt ist, von welchem Objektgraphen sie jeweils eine Ausprägung darstellen, liegt damit eine Interpretation des Szenengraphen vor. Wird jede Interpretation durch ein Gütemaß bewertet, gilt derjenige Objektgraph als erkannt, dessen Vergleich die beste Güte erzielt.

In dieser Arbeit sollen solche Erkennungsverfahren verwendet werden, die im Sinne von Abschnitt 1.2.4 Vorteile hinsichtlich der Parallelisierbarkeit besitzen. Das Korrespondenzproblem wird im folgenden Abschnitt als Optimierungsproblem formuliert.

### 5.1 Optimierungsproblem

Um das Korrespondenzproblem als Optimierungsproblem formulieren zu können, wird im folgenden geklärt, wie Korrespondenzen repräsentiert oder kodiert werden sollen. Der Begriff des Korrespondenzproblems wird auch in der Literatur zur Stereobildanalyse verwendet. In diesem Zusammenhang ist aber stets das Zuordnungsproblem zwischen Knoten des Objekt- und Knoten des Szenengraphen gemeint. Im folgenden wird ein Knoten im Objekt- bzw. Szenengraphen als *Objektknoten* bzw. *Szeneknoten* bezeichnet.

#### 5.1.1 Kodierung von Korrespondenzen

Die Korrespondenz zwischen einem Objektknoten  $e_i$  und einem Szeneknoten  $e_k$  wird durch eine Binär- oder Entscheidungsvariable  $b_{ik} \in \{0, 1\}$  repräsentiert. Werden für alle Paarungen von Objekt- und Szeneknoten entsprechende Variable ein-

geführt, entsteht die folgende Binärmatrix  $\mathbf{B}$ :

$$\mathbf{B} = \begin{pmatrix} b_{1,1} & \dots & b_{i,1} & \dots & b_{j,1} & \dots & b_{N,1} \\ \vdots & & \vdots & & & & \vdots \\ b_{1,k} & \dots & b_{i,k} & & & & b_{N,k} \\ \vdots & & & & & & \vdots \\ b_{1,l} & & & & b_{i,j} & \dots & b_{N,l} \\ \vdots & & & & \vdots & & \vdots \\ b_{1,M} & \dots & b_{i,M} & \dots & b_{j,M} & \dots & b_{N,M} \end{pmatrix}$$

Da angenommen wird, daß jeweils kollineare Szenenkanten bereits zu einer einzigen Kante gruppiert worden sind und eine Objektansicht nicht mehrmals in der Szene instanziiert ist, sollen einem Objektknoten nie mehrere Szenenknoten zugeordnet werden. Es sind also nur solche Korrespondenzen erwünscht, bei denen einer Objektkante entweder genau eine oder keine Szenenkante zugeordnet wird.

Geht man davon aus, daß die Matrix  $\mathbf{B}$  eine Abbildung kodiert, so kann die gesuchte Abbildung nach zwei Fällen charakterisiert werden. (1) Wenn alle Objektknoten entsprechende Instanzen in der Szene besitzen, können alle Objektknoten zugeordnet werden. Das bedeutet, daß auch Szenenkanten existieren, die keine Instanz eines zu erkennenden Objektmodells darstellen. Es existiert also ein Subgraph des Szenengraphen, der isomorph zu dem Objektgraph ist. Die Abbildung  $G_O \mapsto G_S$  des Objektgraphen auf den Szenengraphen wird daher auch als Subgraphisomorphismus [Barrow & Burstall 76] oder als Monomorphismus bezeichnet [Vosselman 92]. Ein Monomorphismus ist eine 1-zu-1 Abbildung, die allen Elementen des Definitionsbereiches (Objektknoten) jeweils genau ein Element des Wertebereiches (Szenenknoten) zuordnet. (2) Objektkanten können aber auch keine korrespondierenden Szenenkanten besitzen, wenn eine Szenenkante aufgrund von Verdeckung oder aufgrund der Lage außerhalb des Blickfeldes nicht sichtbar ist. Da zugeordnete Objektknoten einen Subgraphen des Objektgraphen  $G_O$  bilden und zugeordnete Szenenknoten einen Subgraphen des Szenengraphen  $G_S$  bilden, ist dann ein gemeinsamer Subgraph gesucht. Die entsprechende Abbildung kann analog als partieller Monomorphismus bezeichnet werden, da der Definitionsbereich der Objektknoten nur partiell zugeordnet werden kann.

Die Binärmatrix  $\mathbf{B}$  kodiert nur dann einen partiellen Monomorphismus  $\omega$ , falls in jeder Zeile und Spalte der Matrix maximal eine Eins steht. Diese Forderung läßt sich formal als Bedingung bzgl. Zeilen- und Spaltensumme der Binärmatrix schreiben:

$$\forall i : \sum_{p=1}^N b_{i,p} \leq 1 \quad (\text{Spaltensummenbedingung}) \quad (5.1)$$

$$\forall k : \sum_{q=1}^M b_{q,k} \leq 1 \quad (\text{Zeilensummenbedingung})$$

### 5.1.2 Konstruktion einer Kostenfunktion

Da Szenenbereichshinweise und damit die Attribute des Szenengraphen immer mit Ungenauigkeiten behaftet sind, kommt es nicht darauf an, einen exakt identischen gemeinsamen Subgraphen zu finden. Vielmehr ist ein Subgraph des Objektgraphen gesucht, der einem Subgraphen des Szenengraphen bezogen auf Attribute möglichst

ähnlich ist. Daher soll eine Kostenfunktion konstruiert werden, die sowohl Zwangsbedingungen bzgl. der Monomorphiebedingungen 5.1 als auch bzgl. der Ähnlichkeiten der Subgraphen berücksichtigt.

Die Ähnlichkeit kann durch eine Zielfunktion bewertet werden, die entweder eine zu maximierende Gütefunktion oder zu minimierende Kostenfunktion darstellt. Da das Finden des Maximums einer Gütefunktion genau dem Finden des Minimums einer Kostenfunktion entspricht, wird im folgenden nur noch eine Kostenfunktion verwendet. Sie muß folgende Eigenschaften besitzen: (1) Da zwischen Objekt- und Szenengraph unter Umständen mehrere gemeinsame Subgraphen existieren, soll der größte gemeinsame Subgraph gefunden werden. Daher sollen die Kosten desto geringer ausfallen, je mehr Korrespondenzen gefunden werden. (2) Damit möglichst ähnliche Subgraphen gefunden werden, sollen die Kosten desto geringer ausfallen, je ähnlicher die Attribute der Graphen sind.

Um die Ähnlichkeit zweier Graphen anhand von Kosten zu bewerten, wird eine Kostenfunktion in drei Schritten konstruiert:

1. Berechnen von Teilkosten für Knotenvergleiche anhand von Knotenähnlichkeiten
2. Berechnen von Teilkosten für Knotenpaarvergleiche anhand von Knotenpaarähnlichkeiten
3. Berechnen von Gesamtkosten anhand der beiden obigen Teilkosten

Die beiden Teilkosten werden im folgenden aus Ähnlichkeitsfunktionen aufgebaut. Unterscheidet man nach Attributarten und nach zu vergleichenden Daten, erhält man vier Ähnlichkeitsfunktionen: Ähnlichkeitsfunktionen bzgl. symbolischer Attribute von 3D-Kanten, bzgl. numerischer Attribute von 3D-Kanten und Ähnlichkeitsfunktionen bzgl. symbolischer Attribute von 3D-Kantenpaaren und bzgl. numerischer Attribute von 3D-Kantenpaaren. Die drei obigen Schritte werden jeweils in den nächsten Unterabschnitten erläutert.

### Teilkosten für Knotenvergleich

Die Teilkosten  $K_K$  für den Vergleich von Knoten werden aus allen Ähnlichkeiten  $k$  einzelner Knoten berechnet. Die Ähnlichkeit zweier einzelner Knoten wiederum wird aus der symbolischen Ähnlichkeit  $ks$  und der numerischen Ähnlichkeit  $kn$  zusammengesetzt.

Um die Funktion  $k$  aus  $ks$  und  $kn$  zu berechnen, wird folgende Beobachtung verwendet. Symbolische Attribute wie Kanten- und Vertextyp sind relativ robust, da sie im Verlauf der Bildauswertung aus einer großen Anzahl an Bildpunkten ermittelt worden sind. Numerische Attribute wie Kantenlängen unterliegen einer großen Anzahl von Meßfehlern des Sensors und Ungenauigkeiten in der Bildauswertung. Daher werden Ähnlichkeiten bzgl. numerischer Attribute erst dann ausgewertet, wenn solche bzgl. symbolischer Attribute verträglich sind.

Die symbolischen Attribute einer Objektkante  $e_i$  und einer Szenenkante  $e_k$  sind daher nur dann ähnlich, wenn folgende Bedingungen erfüllt sind:

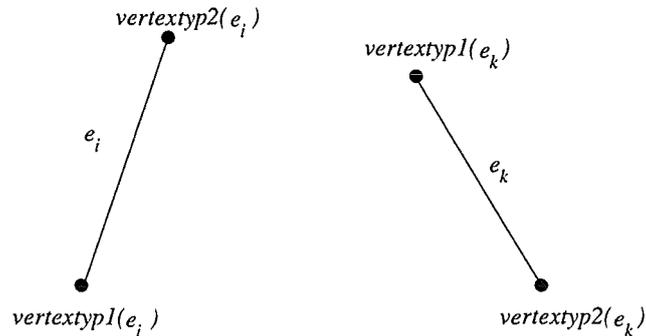


Abbildung 5.1: Bezeichnungen zur Definition der Ähnlichkeit zwischen einem Objektgraphknoten und einem Szenengraphknoten: Die von Knoten repräsentierten Kanten sind mit den zugehörigen Attributbezeichnern beschriftet.

1. Da angenommen wird, daß Szenenkanten von Objektinstanzen in verschiedenen Szenen ihren Kantentyp zum Beispiel aufgrund von Berührung nicht ändern, sollen die Kantentypen identisch sein.
2. Wenn die dem Objektgraph entsprechende Objektansicht in der Szene ausgeprägt ist, ist die Strichzeichnung der Objektansicht mit einem Teil der Strichzeichnung der Szene ähnlich. Das bedeutet, daß die einer Objektkante entsprechende Linie und die der Szenekante entsprechende Linie in der jeweiligen Strichzeichnung identische Linienverbindungen aufweisen müssen. Da Linienverbindungen gemäß Abb. 2.12 durch Indizes identifiziert sind und diese Indizes als Vertextypen dienen, sollen auch die Vertextypen identisch sein. Das heißt, es muß eine Zuordnung von Objektkantenendpunkten zu Szenekantenendpunkten existieren, so daß die Vertextypen identisch sind.

Eine Abschwächung dieser Bedingung auf nicht-identische Vertextypen wäre dann wünschenswert, wenn z.B. aufgrund von Störungen bei der Bildaufnahme einzelne Kanten nicht extrahiert werden können. Eine fehlende Kante kann zu unmöglichen oder nicht-korrekten Linienverbindungen und damit zu nicht-ähnlichen Knoten führen. Um eine abgeschwächte Bedingung zu formulieren, die einen gewissen Grad an Ähnlichkeit trotz nicht-übereinstimmender Vertexattribute erlaubt, benötigt man jedoch zusätzlich Wissen über die Ähnlichkeit von nicht-identischen Linienverbindungen.

3. Die Länge zweier Kanten ist erst dann ein zuverlässiges Merkmal einer Kante, wenn die Kante vollständig sichtbar ist. Ist sie beispielsweise verdeckt, ist die ursprüngliche Länge verkürzt. Wie oben erwähnt, sollen Kantenlängen nur dann verglichen werden, wenn die symbolischen Attribute ähnlich sind. Daher dürfen Kanten bzgl. symbolischer Attribute nur dann ähnlich sein, wenn sie in keinem Endpunkt verdeckt sind. Solche Vertextypen besitzen nach Abb. 2.12 Werte größer oder gleich 21.

Die Ähnlichkeitsfunktionen bzgl. symbolischer Attribute besitzen per Konvention den diskreten Wertebereich  $\{0, 1\}$ , wobei der Wert 0 als unähnlich und der Wert 1 als ähnlich interpretiert wird. Die drei Bedingungen lassen sich mit den Bezeichnungen wie in Abb. 5.1 und in den Definitionen des Szenen- und Objektgraphen formal wie folgt beschreiben:

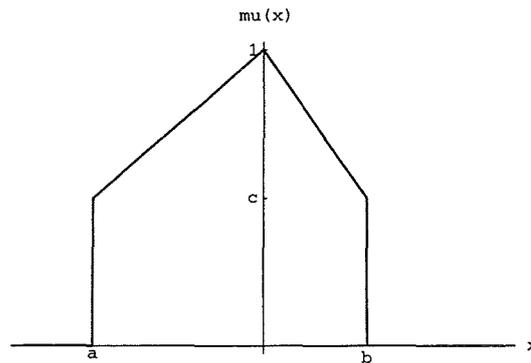


Abbildung 5.2: Ähnlichkeitsfunktion zum Vergleich numerischer Attribute

$$ks \stackrel{\text{def}}{=} \begin{cases} E_O \times E_S \mapsto \{0, 1\} \\ (e_i, e_k) \mapsto ks(e_i, e_k) \end{cases}$$

mit:

$$ks(e_i, e_k) = \begin{cases} 1, & \begin{array}{l} (* \text{ Bedingung 1 } *) \\ (kantentyp(e_i) = kantentyp(e_k)) \\ \wedge \\ (* \text{ Bedingung 2 } *) \\ (vertextyp1(e_k) = vertextyp1(e_i) \wedge \\ vertextyp2(e_k) = vertextyp2(e_i) \vee \\ vertextyp2(e_k) = vertextyp1(e_i) \wedge \\ vertextyp1(e_k) = vertextyp2(e_i)) \\ \wedge \\ (* \text{ Bedingung 3 } *) \\ (vertextyp1(e_k) < 21 \wedge vertextyp2(e_k) < 21) \end{array} \\ 0, & \text{sonst} \end{cases}$$

Erst wenn die drei Bedingungen erfüllt sind, also  $ks = 1$  ist, wird die Kantenlänge als Merkmal von Kanten betrachtet. Die Ähnlichkeit von Kanten wird anhand der Differenz der Kantenlängen durch die Hilfsfunktion  $\mu$  bewertet:

$$\mu_{a,b,c}(x) = \begin{cases} \frac{1-c}{a} \cdot x + 1, & a < x \leq 0 \\ -\frac{1-c}{b} \cdot x + 1, & b > x > 0 \\ 0, & \text{sonst} \end{cases}$$

Die Funktion  $\mu_{a,b,c}(x)$  ist in Abb. 5.2 dargestellt. Sie legt mit den Parametern  $a < 0$  bzw.  $b > 0$  fest, wie stark eine Szenenkante bezogen auf eine Objektkante verkürzt bzw. verlängert sein darf, damit die Ähnlichkeit größer als 0 ist. Der Parameter  $c > 0$  legt den kleinsten Ähnlichkeitswert ungleich 0 fest. Wird  $\mu = 0$  als unähnlich interpretiert, so kann  $\mu = c$  als geringste Ähnlichkeit interpretiert werden.  $c$  ist deshalb größer als Null, damit ähnliche Kanten immer eine höhere Ähnlichkeit besitzen als unähnliche Kanten. Ähnlichkeitsfunktionen bzgl. numerischer Attribute

besitzen per Konvention den kontinuierlichen Wertebereich  $[0, 1]$ , wobei der Wert 0 als unähnlich und Werte aus dem Intervall  $[c, 1]$  als verschieden ähnlich interpretiert werden. Die Ähnlichkeitsfunktion  $kn(e_i, e_k)$  bzgl. numerischer Attribute des Objektknotens  $e_i$  und des Szenenknotens  $e_k$  ist damit:

$$kn \quad \text{def} \quad \begin{cases} E_O \times E_S \mapsto [0, 1] \\ (e_i, e_k) \mapsto kn(e_i, e_k) \end{cases}$$

mit:

$$kn(e_i, e_k) = \mu_{a,b,c}(laenge(e_i) - laenge(e_k))$$

Die Parameter  $a, b$  und  $c$  müssen spezifisch für eine Klasse von Probleminstanzen angepaßt werden. Um auszudrücken, daß die Ähnlichkeitsfunktion  $kn$  nur dann ausgewertet wird, wenn  $ks = 1$  ist, läßt sich die Ähnlichkeit  $k(e_i, e_k)$  zweier Kanten  $e_i, e_k$  nun als Produkt schreiben:

$$k(e_i, e_k) = ks(e_i, e_k) \cdot kn(e_i, e_k)$$

Die Knotenähnlichkeit wird nun dazu verwendet, die Teilkosten von Knotenzuordnungen, die durch eine gegebene Belegung der Binärmatrix  $\mathbf{B}$  festgelegt sind, zu berechnen. Die Teilkosten  $E_K$  berechnen sich als durch die Ähnlichkeiten gewichtete Summe der einzelnen Binärvariablen  $b_{i,k}$  in  $\mathbf{B}$ . Da sich Ähnlichkeiten und Kosten gerade komplementär verhalten, wird die Summe mit -1 multipliziert.

$$K_K(\mathbf{B}) = - \sum_{i,k} k(e_i, e_k) b_{i,k}$$

### Teilkosten für Knotenpaarvergleich

Analog werden die Teilkosten bzgl. der Knotenpaare berechnet. Die Teilkosten  $K_P$  für den Vergleich von Knotenpaaren werden aus allen Ähnlichkeiten  $p$  einzelner Knotenpaare berechnet. Die Ähnlichkeit zweier einzelner Knotenpaare wiederum wird aus der symbolischen Ähnlichkeit  $ps$  und der numerischen Ähnlichkeit  $pn$  zusammengesetzt.

Um die Funktion  $p$  aus  $ps$  und  $pn$  zu berechnen, wird wiederum die Beobachtung verwendet, daß symbolische Attribute wie Kanten- und Vertextyp relativ robust sind, während numerische Attribute wie der Winkel eine bestimmte Ungenauigkeit aufweisen. Daher werden Ähnlichkeiten bzgl. numerischer Attribute wiederum erst dann ausgewertet, wenn solche bzgl. symbolischer Attribute verträglich sind.

Die symbolischen Attribute eines Objektknotenpaares  $(e_i, e_j)$  und eines Szenenknotenpaares  $(e_k, e_l)$  sind daher nur dann ähnlich, wenn folgende Bedingungen erfüllt sind:

1. Der Winkel eines Kantenpaares ist erst dann ein zuverlässiges Merkmal des Kantenpaares, wenn die beiden Kanten einen gemeinsamen Endpunkt besitzen. Wie oben erwähnt, sollen Winkel als numerische Attribute nur dann verglichen werden, wenn die symbolischen Attribute ähnlich sind. Daher dürfen Knotenpaare bzgl. symbolischer Attribute nur dann ähnlich sein, wenn zwischen den entsprechenden Knoten jedes Knotenpaares einen Bogen existiert

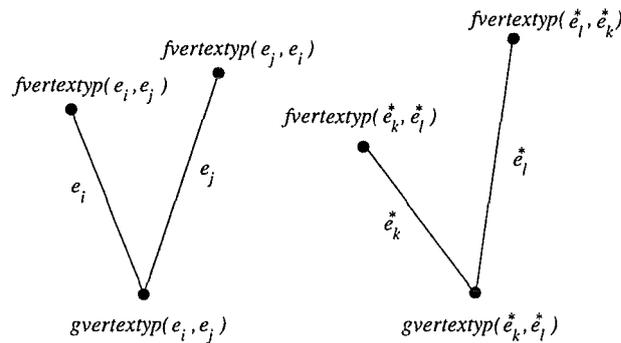


Abbildung 5.3: Bezeichnungen zur Definition der Ähnlichkeit zweier Knotenpaare. Die von Knoten repräsentierte Kanten sind mit den zugehörigen Attributbezeichnungen beschriftet.

und das Bogenattribut *gvertex* des Bogens im Objektgraph mit demjenigen im Szenengraph übereinstimmt.

2. Da Kantenpaare unabhängig von ihrer Orientierung im Raum verglichen werden sollen, kann eine Objektkante potentiell mit einer von *beiden* Kanten eines Szenenkantenpaares korrespondieren. Zwei Knotenpaare sind daher ähnlich, wenn es eine bijektive Abbildung zwischen Knoten der beiden Paare gibt, so daß:

- (a) die gemäß der bijektiven Abbildung zugeordneten Knoten das gleiche Attribut *kantentyp* besitzen.
- (b) die Vertextypen der freien Endpunkte der durch Knoten repräsentierte Kanten unter der bijektiven Abbildung zwischen Objekt- und Szeneknoten identisch und nicht verdeckt sind. Dazu existiert die Ausnahme, daß der Vertextyp maximal einer Szenenkante auch verdeckt sein darf. Dadurch ist es möglich, eine teilweise sichtbare Szenenkante zuzuordnen, obwohl sie zu allen *einzelnen* Objektkanten unähnlich ist.

Für die obige bijektive Abbildung  $f : \{e_i, e_j\} \mapsto \{e_k, e_l\}$  sei folgende Schreibweise gegeben:  $f(e_i) = e_k^*$  und  $f(e_j) = e_l^*$ . Sind zwei Knoten über einen Bogen verbunden, sei außerdem folgende Hilfsfunktion  $fvertextyp(e_i, e_k)$  gegeben, die den Vertextyp des freien Endpunktes der Kante des Knotens  $e_i$  liefert:

$$fvertextyp(e_i, e_j) = \begin{cases} vertextyp1(e_i), & \text{falls } vertextyp2(e_i) = gvertextyp(e_i, e_j) \\ vertextyp2(e_i), & \text{sonst} \end{cases}$$

Die Bedingungen lassen sich mit den Bezeichnungen wie in Abb. 5.3 formal wie folgt beschreiben:

$$ps \quad \text{:def} \quad \begin{cases} E_O^2 \times E_S^2 \mapsto [0, 1] \\ (e_i, e_j, e_k, e_l) \mapsto ps(e_i, e_j, e_k, e_l) \end{cases}$$

mit:

$$ps(e_i, e_j, e_k, e_l) = \begin{cases} 1, & \begin{array}{l} (* \text{ Bedingung 1 } *) \\ (gvertexyp(e_i, e_j) \neq -1 \wedge \\ gvertexyp(e_i, e_j) = gvertexyp(e_k, e_l)) \\ \wedge \\ [ \\ (* \text{ Bedingung 2a } *) \\ (kantentyp(e_i) = kantentyp(e_k^*) \wedge \\ kantentyp(e_j) = kantentyp(e_l^*)) \\ \wedge \\ (* \text{ Bedingung 2b } *) \\ ((fvertexyp(e_i, e_j) = fvertexyp(e_k^*, e_l^*) \wedge \\ fvertexyp(e_k^*, e_l^*) < 21) \wedge \\ (fvertexyp(e_j, e_i) = fvertexyp(e_l^*, e_k^*) \vee \\ fvertexyp(e_l^*, e_k^*) > 20)) \\ ] \end{array} \\ 0, & \text{sonst} \end{cases}$$

Erst wenn die zwei Bedingungen erfüllt sind, also  $ps = 1$  ist, wird der Winkel als Merkmal von Kantenpaaren betrachtet. Die Ähnlichkeit der numerischen Attribute  $pn$  von Knotenpaaren wird analog zu der Knotenähnlichkeit  $kn$  anhand der Differenz der Winkel durch die Hilfsfunktion  $\mu$  bewertet:

$$pn \stackrel{\text{def}}{=} \begin{cases} E_O^2 \times E_S^2 \mapsto [0, 1] \\ (e_i, e_j, e_k, e_l) \mapsto pn(e_i, e_j, e_k, e_l) \end{cases}$$

mit:

$$pn(e_i, e_j, e_k, e_l) = \mu_{a,b,c}(winkel(e_i, e_j) - winkel(e_k, e_l))$$

Die Parameter  $a, b$  und  $c$  müssen wiederum spezifisch für eine Klasse von Probleminstanzen angepasst werden. Die Ähnlichkeit  $p(e_i, e_j, e_k, e_l)$  zweier Knotenpaare  $(e_i, e_j)$  und  $(e_k, e_l)$  läßt sich nun ebenfalls als Produkt schreiben:

$$p(e_i, e_j, e_k, e_l) = ps(e_i, e_j, e_k, e_l) \cdot pn(e_i, e_j, e_k, e_l)$$

Die Knotenpaarähnlichkeit wird nun dazu verwendet, die Teilkosten von Zuordnungspaaren zu berechnen. Die Teilkosten  $K_P$  berechnen sich daher als durch die Ähnlichkeiten gewichtete Summe von quadratischen Produkten einzelner Korrespondenzvariable. Da sich Ähnlichkeiten und Kosten komplementär verhalten, wird die Summe wiederum mit -1 multipliziert:

$$K_P(\mathbf{B}) = - \sum_{i,j,k,l} p(e_i, e_j, e_k, e_l) b_{i,k} b_{j,l}$$

### Gesamtkosten

Eine vorläufige Gesamtkostenfunktion  $K'(\mathbf{B})$ , die die Kosten der Zuordnungen der Binärmatrix  $\mathbf{B}$  bewertet, ist als gewichtete Summe der Teilkosten  $K_K$  und  $K_P$  aufgebaut [Vosselman 92]:

$$K'(\mathbf{B}) = C_K K_K(\mathbf{B}) + C_P K_P(\mathbf{B})$$

Durch die Gewichtungparameter ist es möglich, die Wichtigkeit bestimmter Zwangsbedingungen, die in Form der entsprechenden Teilkosten gegeben sind, genau zu justieren. Die Parameter  $C_K$  und  $C_P$  müssen also ebenso wie die Parameter der Hilfsfunktion  $\mu$  an eine Klasse von Probleminstanzen angepasst werden.

### 5.1.3 Problemformulierung

Mit Hilfe der Kostenfunktion und den Monomorphiebedingungen läßt sich das Korrespondenzproblem nun als Optimierungsproblem formulieren, für dessen Lösung sich die in dem Überblick dargestellten Verfahren verwenden lassen. Als Resultat erhält man eine möglichst optimale Lösung des Problems und damit Zuordnungen von Objekt- zu Szenenknoten.

Ein Optimierungsproblem besteht aus einem Suchraum  $\mathcal{S}$ , einer Kostenfunktion  $K(\mathbf{B})$  abhängig von Variablen  $\mathbf{B} \in \mathcal{S}$  und den Nebenbedingungen  $g_i(\mathbf{B}) \leq 0$ , die das Suchgebiet  $\mathcal{G} = \{\mathbf{B} | \forall i : g_i(\mathbf{B}) \leq 0\} \subset \mathcal{S}$  definieren. Ein Optimierungsproblem heißt diskret bzw. kontinuierlich, wenn die Elemente des Suchraums diskrete bzw. reelle Werte darstellen. Da die Binärmatrizen diskrete Variable darstellen, liegt ein diskretes Optimierungsproblem vor.

Man kann zwei Arten von Beschränkungen unterscheiden: Soll-, bedingte, weiche oder admissive Beschränkungen, die Richtlinien, Zielkriterien, Absichten oder Präferenzen zum Ausdruck bringen, und Muß-, unbedingte, harte oder imperative Beschränkungen, die Vorschriften, Rahmenbedingungen oder Mindestanforderungen vorgeben. Imperative Beschränkungen legen den Raum zulässiger Korrespondenzen fest, während admissive Beschränkungen angeben, welche Korrespondenzen unter den zulässigen bevorzugt werden sollen.

Im Falle des diskreten Optimierungsproblems stellen die Monomorphiebedingungen imperative Beschränkungen und die Kostenfunktion admissive Beschränkungen dar. Das Suchgebiet  $\mathcal{G}$  ist die Menge aller partieller Monomorphismen. Das diskrete Optimierungsproblem für die binäre  $N \times M$ -Matrix  $\mathbf{B}$  mit  $N = |E_O|$  Objektkanten und  $M = |E_S|$  Szenenkanten stellt sich dann formal wie folgt:

#### Problem 5.1 (Diskretes Optimierungsproblem)

$$\begin{array}{ll} \text{Sei} & \mathbf{B} \in \{0, 1\}^{N \cdot M} \\ \text{Minimiere} & K'(\mathbf{B}) = C_R K_K(\mathbf{B}) + C_P K_P(\mathbf{B}) \\ \text{bzgl.} & \forall i : \sum_{p=1}^N b_{p,i} - 1 \leq 0 \\ & \forall j : \sum_{q=1}^M b_{k,q} - 1 \leq 0 \end{array}$$

Gesucht ist möglichst das globale Minimum der quadratischen Kostenfunktion unter Einhaltung der Nebenbedingungen. Eine triviale Lösung zum Ermitteln der besten Korrespondenzen besteht in der erschöpfenden Suche. Das heißt, alle kombinatorisch legalen Zuordnungen werden systematisch aufgezählt und die Kostenfunktion wird ausgewertet. Danach werden alle minimalen Lösungen ausgegeben. Allerdings ergeben sich dadurch exponentiell viele Zuordnungen in Abhängigkeit von der Problemeingabe. Das hier vorliegende Problem ist als Spezialfall des Problems, einen größten gemeinsamen Subgraph zweier Graphen zu finden, *NP*-vollständig [Herauld & Niez 91], [Garey & Johnson 79]. Aufgrund der kombinatorischen Explosion können Probleme dieser Klasse praktisch nur für kleine Problemeingaben in vertretbarer Zeit gelöst werden. Interessant sind daher Algorithmen, die anstatt global optimaler vielmehr qualitativ hochwertige, suboptimale Näherungslösungen mit Hilfe von Heuristiken liefern. Hier sollen Probleminstanzen getestet werden, die sich durch folgende Eigenschaften auszeichnen: Der Knotenrang der Szenen- und Objektgraphen ist aufgrund trihedral polyedrischer Szenen maximal vom Grad drei. Die Szenengraphen sind außerdem i.a. nicht vollständig vernetzt.

Diskrete Korrespondenzvariable sind zwar ausreichend, um Korrespondenzen qualitativ zu kodieren. Kontinuierliche Korrespondenzwerte wären allerdings wünschenswert, um Korrespondenzen auch quantitativ bewerten zu können. Kontinuierliche Variable sagen nicht nur aus, ob Knoten korrespondieren oder nicht, sondern

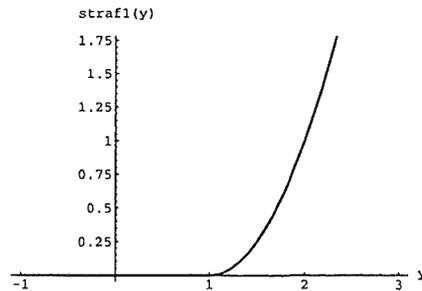


Abbildung 5.4: Straffunktion  $straf_1$ : Die Straffunktion dient zur Einhaltung der Zeilen- und Spaltensummenbedingung

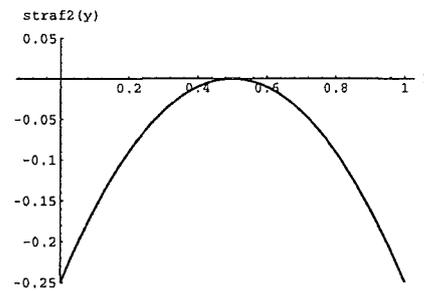


Abbildung 5.5: Straffunktion  $straf_2$ : Die Straffunktion dient zur Bevorzugung binärer Endzustände

auch wie gut bzw. wie schlecht sie korrespondieren. Korrespondenzwerte zwischen Knoten können verwendet werden, um genauer zu entscheiden, wann eine Objektansicht als erkannt gilt. Das diskrete Optimierungsproblem wird daher in ein kontinuierliches Ersatzproblem überführt. Dies erfolgt in vier Schritten:

1. Ersetzen binärer Entscheidungsvariable durch kontinuierliche Variable
2. Entfernen der Nebenbedingungen und Erweitern der Kostenfunktion
3. Einführen einer neuen Nebenbedingung
4. Bevorzugen binärer Endzustände durch Erweitern der Kostenfunktion

Zunächst wird statt der Binärmatrix  $\mathbf{B}$  eine Matrix  $\mathbf{X}$  bestehend aus reellen Matrixelementen  $x_{ik}$  verwendet. Im zweiten Schritt werden die linearen Nebenbedingungen bzgl. der Zeilen- und Spaltensumme als neue Teilkosten  $E_{SZ}$  berücksichtigt. Zur Berechnung von Teilkosten kann die Methode der Straffunktionen oder die Methode der Barrierefunktionen verwendet werden. Im Falle von Straffunktionen entstehen erst dann Kosten, wenn die Zeilen- oder Spaltensumme Eins überschreitet. Im Falle von Barrierefunktionen entstehen je mehr Kosten, desto näher die Zeilen- oder Spaltensumme den Wert von Eins annähert. Da sich die Methode der Straffunktionen nicht nur auf Nebenbedingungen in Form von Gleichungen, sondern auch in Form von Ungleichungen anwenden läßt, werden hier Straffunktionen verwendet. Eine weitere Methode, um Nebenbedingungen in Form von Ungleichungen einzuhalten, besteht im Einführen weiterer Optimierungsvariable [Tagliarini *et al.* 91]. Durch die Straffunktion  $straf_1(y)$  entstehen quadratische Teilkosten, falls die Zeilen- oder Spaltensumme größer als 1 ist. Den Funktionsgraphen von  $straf_1(y)$  zeigt Abb. 5.4.

$$K_{SZ}(\mathbf{X}) = \sum_i straf_1\left(\sum_p x_{p,i}\right) + \sum_k straf_1\left(\sum_q x_{k,q}\right)$$

$$straf_1(y) = \begin{cases} (y-1)^2, & y > 1 \\ 0, & \text{sonst} \end{cases}$$

Da die Matrixelemente  $x_{i,k}$  reellwertig sind, jedoch nur Werte aus dem Einheitsintervall  $[0, 1]$  zulässig sind, wird in dem dritten Schritt eine neue lineare Nebenbedingung bzgl. der Matrixelemente  $x_{i,k}$  eingeführt:

$$\forall i, k : 0 \leq x_{i,k} \leq 1$$

Die kontinuierlichen Variablen sollten außerdem möglichst nahe bei 0 oder 1 liegen, um eine Entscheidung für oder gegen eine Korrespondenz fällen zu können. Durch eine weitere Straffunktion werden daher Werte von  $x_{i,k}$  in Form weiterer Teilkosten  $E_{01}$  je mehr bestraft, je weiter sie von 0 oder 1 entfernt liegen. Den Funktionsgraph von  $\text{straf}_2(y)$  zeigt Abb. 5.5:

$$K_{01}(\mathbf{x}) = - \sum_{i,k} \text{straf}_2(x_{i,k})$$

$$\text{straf}_2(y) = -(y - 1/2)^2$$

Nun kann das kontinuierliche Ersatzproblem mit einer neuen Nebenbedingung und einer um zwei Teilkostenterme erweiterten Kostenfunktion formuliert werden:

#### Problem 5.2 (Kontinuierliches Ersatzproblem)

$$\begin{array}{ll} \text{Sei} & \mathbf{X} \in \mathbb{R}^{N \cdot M} \\ \text{Minimiere} & K(\mathbf{X}) = C_R K_K(\mathbf{X}) + C_P K_P(\mathbf{X}) + C_{SZ} K_{SZ}(\mathbf{X}) + C_{01} K_{01}(\mathbf{X}) \\ \text{bzgl.} & \forall i, k : 0 \leq x_{i,k} \leq 1 \end{array}$$

Da die Kostenfunktion  $K$  aus quadratischen Polynomen zusammengesetzt ist, stellt sie selbst wieder ein quadratisches Polynom dar und läßt sich auch in folgender Form mit den Koeffizienten  $a_{i,k,j,l}$  quadratischer und den Koeffizienten  $b_{i,k}$  linearer Teilterme schreiben:

$$K(\mathbf{X}) = \sum_{i,k,j,l} a_{i,k,j,l} x_{i,k} x_{j,l} + \sum_{i,k} b_{i,k} x_{i,k}$$

## 5.2 Optimierungsverfahren der iterativen Verbesserung

Objekte sollen hier anhand von Korrespondenzen zwischen Objekt- und Szenengraphknoten erkannt werden. Dazu werden alle Objektgraphen unabhängig voneinander mit einem Szenengraphen verglichen. Für jeden Vergleich sollen optimale Korrespondenzen zwischen Objekt- und Szenenknoten ermittelt werden. Die Erkennung stellt ein Korrespondenzproblem dar, das hier als Optimierungsproblem in Form des Ersatzproblems 5.2 gestellt wurde. Da hier Optimierungsverfahren der iterativen Verbesserung eingesetzt werden, wird diese Klasse von Verfahren im folgenden erläutert.

### 5.2.1 Iterative Verbesserung

Verfahren der iterativen Verbesserung benötigen kontinuierliche Optimierungsvariable  $\mathbf{X}$ , eine Zielfunktion, ohne Beschränkung der Allgemeinheit eine Kostenfunktion  $K(\mathbf{X})$ , eine Schätzung des Startwertes  $\mathbf{X}^0$ , eine Iterationsvorschrift  $\mathbf{X}^{t+1} \leftarrow f(\mathbf{X}^t)$ , wie man aus einem aktuellen Wert  $\mathbf{X}^t$  zum Zeitpunkt  $t$  einen wahrscheinlich besseren Wert  $\mathbf{X}^{t+1}$  erhält, und eine Terminierungsbedingung  $\mathcal{B}$ . Die hier untersuchten Verfahren verwenden als Iterationsvorschrift  $f(\mathbf{X}^t) = \mathbf{X}^t + \Delta \cdot \mathbf{g}(\mathbf{X}^t)$ , wobei  $\mathbf{g}$  die Richtung darstellt, in der mit hoher Wahrscheinlichkeit bessere Werte zu finden sind. Die Schrittweite  $\Delta$  gibt an, wie weit man entlang der Richtung gehen muß, um den Wert  $\mathbf{X}^{t+1}$  zu erhalten.

Der schematische Ablauf besteht aus Initialisierung, iterativer Verbesserung bis eine Terminierungsbedingung erfüllt ist und einer Interpretation  $\mathcal{I}$ . In der Initialisierung wird die Optimierungsvariable  $\mathbf{X}$  auf einen Startwert  $\mathbf{X}^0$  gesetzt. In jedem Iterationsschritt werden alle Optimierungsvariable  $x_{i,k}$  verbessert und eine lineare Nebenbedingung bzgl. der Variable  $x_{i,k}$  durch die Projektionsfunktion  $\pi$  eingehalten. Eine einfache Projektionsfunktion stellt eine Schwellwertfunktion dar, die bei Unter- oder Überschreitung von Schwellwerten konstante Werte und sonst Werte gemäß einer linearen Funktion liefert. Der Iterationsschritt wird solange wiederholt, bis eine näher zu spezifizierende Terminierungsbedingung erfüllt ist. Schließlich werden die kontinuierlichen Variable  $x_{i,k}$  der Matrix  $\mathbf{X}$  durch die Funktion  $\mathcal{I}(\mathbf{X})$  als Entscheidungs-, Korrespondenz- oder Binärvariable interpretiert.

Nach der Initialisierung und der Iterationsvorschrift lassen sich deterministische und stochastische Verfahren der iterativen Verbesserung unterscheiden. Deterministische Verfahren liefern im Gegensatz zu stochastischen Verfahren für gleiche Eingaben immer gleiche Ausgaben. Ist die Initialisierung der Optimierungsvariable oder die Iterationsvorschrift von einer Wahrscheinlichkeitsverteilung abhängig, liegt ein stochastisches Verfahren vor. Da aufgrund von Echtzeitanforderungen nicht genügend Zeit für mehrere, eventuell zeitraubende Erkennungsversuche zur Verfügung steht und die Güte eines Resultates nicht vom Zufall abhängig sein sollte, sollen hier nur deterministische Verfahren betrachtet werden. Das heißt, sowohl Verbesserungsschritt als auch Initialisierung sollen deterministisch sein. Der Aufbau der iterativen Verbesserung kann schematisch als Algorithmus in Tab. 5.1 dargestellt werden.

In der folgenden Übersicht wird nun dargestellt, wie Verfahren aus diesen Bereichen bisher für Korrespondenzprobleme in der Objekterkennung eingesetzt worden sind.

**Algorithmus 5.1 (Schema Iterative Verbesserung)**

<b>Eingabe.</b>	Startwert $\mathbf{X}^0$ , Kostenfunktion $K(\mathbf{X})$ , Terminierungsbedingung $\mathcal{B}$
<b>Ausgabe.</b>	Binärmatrix $\mathbf{B}$
<b>Algorithmus.</b>	<pre> (* Initialisierung *) <math>\forall i, k : x_{i,k} \leftarrow x_{i,k}^0</math>; <math>t \leftarrow 0</math>; (* iterative Verbesserung *) while <math>\neg \mathcal{B}</math> do (* Terminierungsbedingung *)   (* Iterationsschritt *)   for <math>(i, k) \in \{1, \dots, N\} \times \{1, \dots, M\}</math> do     (* Richtung *)     <math>g_{i,k}(K(\mathbf{X}), t) \leftarrow \frac{\partial K(\mathbf{X})}{\partial x_{i,k}}</math>;     (* Verbesserung *)     <math>x_{i,k}(t+1) \leftarrow x_{i,k}(t) - \Delta \cdot g_{i,k}(K(\mathbf{X}), t)</math>;     (* Projektion *)     <math>x_{i,k}(t+1) \leftarrow \pi(x_{i,k}(t+1))</math>;   od;   <math>t \leftarrow t + 1</math>; od; (* Interpretation *) <math>\mathbf{B} \leftarrow \mathcal{I}(\mathbf{X})</math>; return <math>\mathbf{B}</math> </pre>

Tabelle 5.1: Schema der Iterativen Verbesserung

**5.2.2 Übersicht**

Im folgenden wird in einer kurzen Übersicht dargestellt, wie die Verfahren der iterativen Verbesserung zur Lösung von Korrespondenzproblemen im Kontext der Objekterkennung eingesetzt worden sind. Die Verfahren besitzen in ihrer Anwendung auf Objekterkennungsprobleme die Gemeinsamkeit, daß Modelle sowie Szenenbereichshinweise nur Attribute von und nur binäre Beziehungen zwischen Modell- und Szenenbereichselementen repräsentieren. Beziehungen zwischen drei oder mehr Elementen sind also nicht erlaubt. Gemeinsam ist ebenfalls, daß in der Zielfunktion entsprechend unäre bzw. binäre Zwangsbedingungen bezogen auf Attribute bzw. Beziehungen kodiert sind, die einen diskreten oder kontinuierlich Wertebereich besitzen.

Unterschiede der dargestellten Ansätze liegen in der Anwendung, und folglich in den Szenenbereichshinweisen, den Objektmodellen, in dem Aufbau der Zielfunktion und dem gewählten Optimierungsverfahren. In dieser Übersicht werden nur Ansätze untersucht, die auf folgenden Optimierungsverfahren basieren:

- Rückgekoppelte künstliche neuronale Netze nach [Hopfield & Tank 85]
- Mean Field Annealing [Peterson & Anderson 87]
- Relaxationsmarkierung [Rosenfeld *et al.* 76]

Eine tabellarische Übersicht über Anwendungen der Verfahren in der Objekterkennung oder verwandten Problemstellungen zeigt Tabelle 5.2.

Referenz	Ansatz	Szenenbereichshinweise
[Kitchen 80]	Relaxationsmarkierung	Landkarten
[Nevatia & Price 82]	Relaxationsmarkierung	Strichzeichnungen aus Luftaufnahmen
[Kuner 87], [Kuner & Ueberreiter 88]	Relaxationsmarkierung, Hopfield/Tank-Netz	technische Zeichnungen
[Parvin & Medioni 91]	Hopfield/Tank-Netz	Flächenelemente aus Abstandsbildern
[Lin <i>et al.</i> 91]	Hopfield-Netz, Hopfield/Tank-Netz	Strichzeichnung
[Nasrabadi & Li 91]	Hopfield-Netz	2D-Konturrecken
[Tresp 91b]	Mean Field Annealing	3D-Kanten aus Stereobildern
[Chen & Lin 91]	Mean Field Annealing	CSG-Graphen aus Abstandsbildern

Tabelle 5.2: Übersicht von Anwendungen in der Objekterkennung unter Einsatz von Verfahren der iterativen Verbesserung

Kitchen [Kitchen 80] schlägt eine Erweiterung des Ansatzes in [Rosenfeld *et al.* 76] vor, um Korrespondenzen auch in quantitativen relationalen Strukturen finden zu können. Experimentelle Resultate werden nur anhand politischer Landkarten als relationale Strukturen demonstriert. Dabei dienen die Einwohnerzahl eines Landes und Entfernungen zwischen Städten als quantitative Attribute, Ländergrenzen als wechselseitige Beziehungen.

Nevatia und Price [Nevatia & Price 82] haben sich mit der Interpretation von Luftaufnahmen befaßt. Bilder werden durch Linien, die Brücken repräsentieren, und durch Regionen beschrieben, die Städte, Flugplätze, Wald- oder Wasserflächen repräsentieren. Räumliche Beziehungen sind Relationen wie links-von, Teil-von oder benachbart-mit. Linien und Regionen werden durch Attribute wie Größe, Intensität, Position, Orientierung und dem Verhältnis von Länge zu Breite beschrieben. Die Bildbeschreibung wird durch die eine Variante der Relaxationsmarkierung von [Rosenfeld *et al.* 76] mit einer Modellbeschreibung des aufgenommenen Landgebietes verglichen.

Kuner und Überreiter [Kuner 87], [Kuner & Ueberreiter 88] untersuchen die Interpretation von technischen Zeichnungen elektrischer Schaltungen. Ein Schaltbild und Modelle von Schaltsymbolen bestehen aus Linien. Beziehungen zwischen Linien beschreiben zwei Winkel und zwei Skalierungsparameter, die zusammen angeben, wie eine Linie transformiert werden muß, um mit einer anderen zu koinzidieren. Die Kostenfunktion bewertet Korrespondenzen aufgrund der vier Parameter. Die Optimierung wird durch die nicht-lineare probabilistische Variante der Relaxationsmarkierung und durch eine Modifikation des Hopfield/Tank-Netzes durchgeführt.

Tresp [Tresp 91b] verwendet Korrespondenzen zwischen 3D-Kanten, extrahiert aus Stereo-Bildern, und 3D-Kanten in objektzentrierten Modellen zur Objekterkennung. In den Knoten eines attributierten Graphen werden Kanten, in den Bögen werden die Winkel zwischen sowie das Längenverhältnis von benachbarten Kanten repräsentiert. Korrespondenzen zwischen Szenen- und Modell-Linien werden durch vier Kostenterme bewertet: zwei Terme für Zwangsbedingungen bzgl. Knoten und Knotenpaaren und zwei Terme für Zwangsbedingungen bzgl. erlaubter Korrespondenzen.

Parvin und Medioni [Parvin & Medioni 91] verwenden Korrespondenzen zwischen segmentierten Flächenstücken und Vertizes von 3D-Objektansichten in Abstandsbildern, um aus einzelnen Ansichten 3D-Objekte zu rekonstruieren. Korrespondenzen zwischen Flächen dienen in einem ersten Schritt zum Bilden von Hypothesen, die in einem zweiten Schritt anhand von Vertexkorrespondenzen verfeinert und verifiziert werden. Attributierte Graphen repräsentieren in den Knoten einerseits Flächenmerkmale wie Flächeninhalt, Flächentyp, durchschnittlichen Hauptkrümmungen und Flächenrichtung, andererseits Vertexmerkmale wie die Vertexposition und den Winkel der Kantentangente. In den Graphbögen wird der Typ der zwischen angrenzenden Flächen liegenden Kante bzw. der Kantentyp zwischen benachbarten Vertizes repräsentiert. Die Kostenfunktion berücksichtigt in drei Teilfunktionen lokale Zwangsbedingungen bzgl. der Flächenattribute, Adjazenzzwangsbedingungen zwischen benachbarten Flächen und globale Zwangsbedingungen bzgl. der geometrischen Transformation einzelner Ansichten. Als Optimierungsverfahren wird das kontinuierliche Hopfield/Tank-Netz verwendet.

Nasrabadi und Li [Nasrabadi & Li 91] verwenden Korrespondenzen zwischen Ecken auf Konturlinien von 2D-Objekten und Ecken von Modellen zur Objektidentifikation und Objektlokalisierung. Objekte können sich in der Szene teilweise überlappen und berühren. Die Bögen repräsentieren benachbarte Ecken, die Knoten des attributierten Graphen repräsentieren Winkel und Abstand zwischen benachbarten Ecken. Die Kostenfunktion besteht aus einem Term zur Bewertung korrespondierender Ecken und einem Term bzgl. erlaubter Korrespondenzen. Als Optimierungsverfahren dient ein Hopfield-Netz, das Neuronen mit binären Ausgängen verwendet. Es wird eine Technik vorgestellt, wie mit Hilfe eines globalen Objektgraphen Korrespondenzen mit allen einzelnen Objektgraphen parallel gefunden werden können.

Chen und Lin [Chen & Lin 91] verwenden Korrespondenzen zwischen CSG-Vorranggraphen zur Erkennung von 3D-Objekten in Abstandsbildern. Solche Graphen können unter der Annahme von Szenen, die aus Quadrikflächen bestehen, aus Abstandsbildern extrahiert werden. Die Knoten eines attributierten Graphen repräsentieren die geometrischen Primitive, die Bögen repräsentieren Verknüpfungsoperationen oder die Beziehungen „Verdeckung“ und „Angrenzung“. Die Kostenfunktion gewährleistet valide Zuordnungen, die Ähnlichkeit korrespondierender Primitive, die Erhaltung der Vorranggraph-Topologie und der geometrischen Struktur. Als Optimierungsverfahren wird das Mean Field Annealing gewählt.

Lin et. al. [Lin et al. 91] verwenden Korrespondenzen zwischen Linien in einer Strichzeichnung und Linien in charakteristischen Ansichten zur Objekterkennung. Durch Flächenkorrespondenzen werden zunächst Hypothesen gebildet, die danach durch Vertexkorrespondenzen verifiziert werden. Die Knoten eines attributierten Graphen repräsentieren Regionenmerkmale (wie Mittelpunkt, Flächeninhalt) bzw. Vertexmerkmale (wie Winkel zwischen Linien). Die Bögen repräsentieren Beziehungen zwischen Regionen (wie Abstände zu Nachbarregionen) bzw. zwischen Vertizes (wie die sog. Formzahl). Als Optimierungsverfahren werden das binäre Hopfield-Netz und das kontinuierliche Hopfield/Tank-Netz verwendet.

Li [Li 92] verwendet Korrespondenzen zwischen kleinen gekrümmten Flächenelementen in einem Abstandsbild und Flächenelementen charakteristischer Ansichten zur Objekterkennung. In den Knoten eines attributierten Graphen werden Flächentyp und Flächeninhalt repräsentiert, in den Bögen der Abstand zu benachbarten Flächenelementen. Ein linearer Teilterm der Kostenfunktion berücksichtigt die Zwangsbedingung bzgl. der Flächen, ein quadratischer Teilterm die Zwangsbedingung bzgl. benachbarter Flächen.

Entsprechend der in der Übersicht behandelten Klassen an Optimierungsverfahren

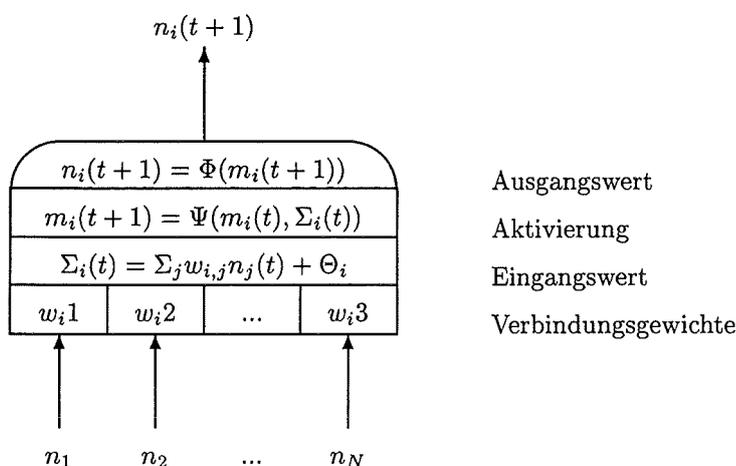


Abbildung 5.6: Modell eines formalen Neurons

ren werden hier drei Vertreter aus je einer Klasse ausgewählt, einander gegenübergestellt und verglichen. Als Optimierungsverfahren werden das Mean Field Annealing (kurz: MFA) als Thermodynamisches Modell, die nicht-lineare probabilistische Variante der Relaxationsmarkierung (kurz: NLR-Markierung) und eine Vereinfachung des Hopfield/Tank-Netzes (kurz: HT-Netz) behandelt. Das vereinfachte Hopfield/Tank-Netz entspricht dem Brain-State-in-a-box Netz (kurz: BSB-Netz). Zunächst wird aber erläutert, wie im Falle des Optimierungsproblems 5.2 geeignete rückgekoppelte neuronale Netze dem obigen Schema entsprechen.

### 5.2.3 Künstliche rückgekoppelte neuronale Netze

Künstliche neuronale Netze, hier kurz neuronale Netze genannt, bestehen aus (1) formalen Neuronen, die gemäß einer bestimmten (2) Vernetzungsstruktur über gewichtete Verbindungen vernetzt sind, (3) einer Gewichtbestimmung und (4) einer Schaltdynamik, die angibt, in welcher Reihenfolge Neuronen schalten.

Jedes Neuron  $i$  verrechnet mehrere Ausgänge von anderen Neuronen zum Zeitpunkt  $t$  und gibt ein Resultat an genau einem Ausgang  $n_i(t+1)$  aus.

Die Verrechnung erfolgt im allgemeinen Fall in drei Schritten (Abb. 5.6). Aus den Eingängen anderer Neuronen, den Gewichten der Verbindungen zu diesen Neuronen und einem Schwellwert  $\Theta_i$  wird zunächst ein Eingangswert  $\Sigma_i(t)$  berechnet. Im nächsten Schritt wird aus dem Eingangswert und einer momentanen Aktivierung  $m_i(t)$  eine neue Aktivierung  $m_i(t+1)$  gebildet:  $m_i(t+1) = \Psi(m_i(t), \Sigma_i(t))$ . Schließlich ergibt sich der Ausgangswert  $n_i(t+1)$  eines Neurons aus der Aktivierung  $m_i(t+1)$  über eine i.a. nicht-lineare Ausgangsfunktion  $\Phi$ :  $n_i(t+1) = \Phi(m_i(t+1))$ .

Bei der Schaltdynamik eines Neurons unterscheidet man zwischen diskret/kontinuierlich und synchron/asynchron. Im zeitkontinuierlichen Falle ergibt sich das Neuronenverhalten als System gekoppelter Differentialgleichungen, im zeitdiskreten Falle als System gekoppelter Differenzgleichungen. Da die Systeme auf Digitalrechnern simuliert werden, wird die zeitkontinuierliche Schaltdynamik durch Anwendung numerischer Integrationsverfahren diskret simuliert. Bei synchroner Dynamik

schalten pro Zeitpunkt alle Neuronen gleichzeitig, bei asynchroner Schaltdynamik ändern die Neuronen ihre Werte zu verschiedenen Zeitpunkten.

Die Vernetzungsstrukturen lassen sich anhand von Graphen klassifizieren, indem man die Neuronen mit Knoten eines Graphen und die Verbindungen mit Kanten des Graphen identifiziert. Die Vernetzungsstruktur eines neuronalen Netzes läßt sich dann als nicht-rückgekoppelt bzw. rückgekoppelt klassifizieren, falls der entsprechende Graph zyklen-frei bzw. zyklisch ist. Speziell symmetrisch rückgekoppelte Vernetzungsstrukturen sind bei dem Einsatz von neuronalen Netzen als Optimierungsverfahren bedeutend. Bei symmetrisch rückgekoppelter Vernetzung besitzen Verbindungsgewichte die Eigenschaft, daß das Verbindungsgewicht  $w_{i,j}$  von Neuron  $j$  zu Neuron  $i$  gleich dem Gewicht  $w_{j,i}$  von Neuron  $i$  zu Neuron  $j$  ist.

In dem Vorgang der Gewichtbestimmung können die Verbindungsgewichte entweder durch eine Lernregel iterativ angepaßt oder aufgrund einer Problemspezifikation direkt berechnet werden. Im Falle von Optimierungsproblemen können die Gewichte aufgrund einer Problemspezifikation direkt berechnet werden. Gewichte und Vernetzung symmetrischer rückgekoppelter neuronaler Netze sind durch eine Energiefunktion bestimmt. Nach einer bestimmten Anzahl von Schaltvorgängen besitzt das Netz einen Zustand minimaler Energie, der von der Energiefunktion abhängig ist.

### Brain-State-in-a-Box Netz

Hopfield und Tank [Hopfield & Tank 85] haben ein quadratisches Optimierungsproblem, das sog. Handlungsreisenden Problem, so in ein analoges Hopfield Netzwerk kodiert, daß die Netzenergie beim Einschwingvorgang näherungsweise optimiert wird. Das nach den Autoren benannte Hopfield/Tank-Netz oder kurz HT-Netz fällt in die allgemeine Klasse der Gradientenverfahren der nicht-linearen Optimierung mit Randbedingungen. Im Gegensatz zu konventionellen Methoden besitzt das HT-Netz jedoch ein größeres Potential der Parallelisierbarkeit [Bouzerdoum & Pattison 93].

Braun [Braun 90] verbesserte das Hopfield/Tank-Netz durch Eliminierung redundanter Parameter und durch Vereinfachungen. Das vereinfachte Modell entspricht im Wesentlichen dem Brain-state-in-a-box Netz, das in [Anderson *et al.* 77] untersucht wurde. Der Name rührt daher, daß der Zustand aller Neuronen innerhalb eines positiven Einheitswürfels liegt. In [Poteryaiko 91] wurde das Brain-State-in-a-Box Netz oder kurz BSB-Netz speziell für Graph-Matching Probleme vorgeschlagen.

In Neuronen des BSB-Netzes wird der Eingangswert  $\Sigma$  durch gewichtete Summation der Eingänge und Subtraktion eines Schwellwertes berechnet:

$$\Sigma_i(t) = \sum_j w_{i,j} n_j(t) + \Theta_i$$

Die neue Aktivierung ergibt sich aus der alten Aktivierung durch:

$$m_i(t+1) = m_i(t) + \Delta \Sigma_i(t)$$

Aus der Aktivierung wird durch die stückweise lineare Ausgangsfunktion  $f_{lin}$  der Ausgangswert  $n_i(t+1)$  berechnet:

$$n_i(t+1) = f_{lin}(m_i(t+1))$$

Der Graph der Ausgangsfunktion ist in Abb. 5.7 gezeigt, sie ist wie folgt definiert:

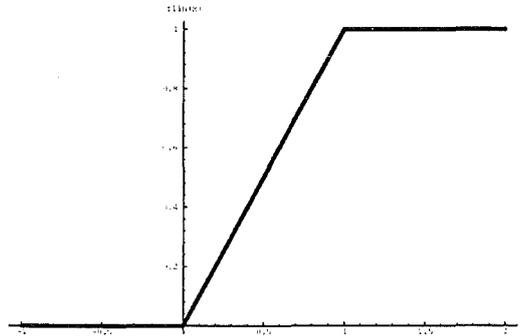


Abbildung 5.7: Stückweise lineare Ausgangsfunktion  $f_{in}$  des BSB-Neurons

$$f_{in}(x) = \begin{cases} 1, & x > 1 \\ x, & 0 \leq x \leq 1 \\ 0, & x < 0 \end{cases}$$

Wie im Hopfield-Netz lautet die Netzenergiefunktion:

$$E(\mathbf{n}) = \sum_{i,j} w_{i,j} n_i n_j + \sum_i \Theta_i n_i$$

Wie im folgenden gezeigt wird, arbeitet das Neuronale Netz nach dem Schema der iterativen Verbesserung. Dazu werden der Reihe nach das BSB-Neuron, die Gewichtbestimmung, die Vernetzung und die Schaltdynamik in Anlehnung an das Schema der iterativen Verbesserung erläutert.

Für die Anwendung auf das Korrespondenzproblem identifiziert man zunächst die Neuronenaktivierung  $m_i$  über eine geeignete Indextransformation mit den Optimierungsvariablen  $x_{i,k}$ , die Neuronenausgaben  $n_i$  mit den projizierten Optimierungsvariablen  $\pi(x_{i,k})$  und die Schwellwerte  $\Theta_i$  mit den Koeffizienten  $b_{i,k}$  der Kostenfunktion. Eine Neuronenausgabe gibt dann die aktuelle Stärke einer Korrespondenz an. Zweitens wird die Energiefunktion  $E(\mathbf{n})$  des neuronalen Netzes ebenfalls über die Indextransformation mit der Kostenfunktion  $K(\mathbf{X})$  des Optimierungsproblems identifiziert, so daß das Netz im Endzustand geringer Netzenergie gerade gute Korrespondenzen widerspiegelt.

Die Kostenfunktion legt erstens die Gewichte und zweitens auch die Vernetzung des BSB-Netzes fest. Die Koeffizienten von Teiltermen zweiter Ordnung (quadratische Teilterme) der Kostenfunktion  $K$  entsprechen den Gewichten zwischen jeweils zwei Neuronen. Die Koeffizienten von Teiltermen erster Ordnung (lineare Teilterme) entsprechen den jeweiligen Schwellen von Neuronen.

Die Vernetzung kommt wie folgt zustande. Kodiert man die Vernetzung durch Gewichte ungleich Null, so sind die Neuronen  $(i, k)$  und  $(j, l)$  genau dann verbunden, wenn das Gewicht  $w_{i,k,j,l} \neq 0$  ist. Verbindungen zwischen verschiedenen Neuronen können also nur aufgrund der quadratischen Teilkosten  $K_P, K_{SZ}$  entstehen. Aufgrund der Zeilen- und Spaltensummenbedingung ist ein Neuron nicht mit allen, sondern nur mit Neuronen der gleichen Spalte und Zeile verbunden. Aufgrund der Knotenpaarvergleiche sind Neuronen nur dann verbunden, wenn sie ähnliche Knotenpaare repräsentieren. Das Netz ist also für praktisch auftretende Probleminstanzen nicht vollständig vernetzt.

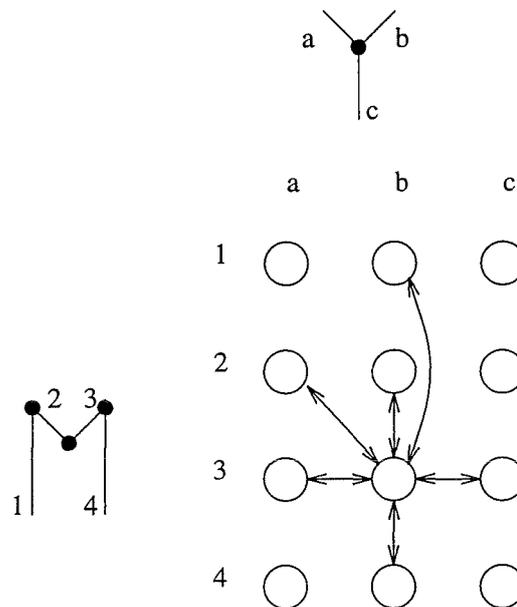


Abbildung 5.8: Nicht-vollständige Vernetzung von Neuron  $b_3$  (Erläuterung siehe Text)

Diese unvollständige Vernetzung hat den Vorteil, daß zur Berechnung des Eingangswertes eines Neurons nicht alle möglichen Verbindungen, sondern nur die notwendigen Verbindungen gewichtet und aufsummiert werden müssen. Bei  $N$  Objektknoten und  $M$  Szeneknoten beträgt zum Beispiel die Anzahl der Verbindungen zur Berechnung der Zeilen- und Spaltensumme nur  $N + M$  und nicht  $N \cdot M$ . Zur Verdeutlichung des Sachverhalts ist in Abb. 5.8 die Korrespondenz- bzw. Neuronenmatrix für eine Probleminstanz mit den Linien des Buchstabens „Y“ als Knoten des Objektgraphen und den Linien des Buchstabens „M“ als Knoten des Szenengraphen abgebildet. Die drei Objektlinien sind alphabetisch, die vier Szenenlinien sind numerisch identifiziert. Die resultierende Neuronenmatrix  $N$  besitzt also  $3 \cdot 4 = 12$  Neuronen, die als Kreise gezeichnet sind. Gezeigt ist nur die lokale Vernetzung in dem Neuron  $n_{b_3}$ . Von den sechs Verbindungen dienen drei der Spaltensummenbedingung, zwei der Zeilensummenbedingung und eine dem Kantenpaarvergleich. Dabei wird angenommen, daß für den Kantenpaarvergleich nur der eingeschlossene Winkel herangezogen wird.

Da bei allen hier betrachteten Verfahren eine diskrete und synchrone Schaltdynamik verwendet wird, ergibt sich der Algorithmus für das BSB-Netz wie in Tab. 5.3.

### 5.2.4 Mean Field Annealing

Ein weiteres Optimierungsverfahren der iterativen Verbesserung ist das Mean Field Annealing, das kurz als MFA bezeichnet wird. In [Tresp 91a] und [Chen & Lin 91] wurde das MFA erfolgreich zur Lösung von Korrespondenzproblemen in der Objekterkennung verwendet. Hier soll untersucht werden, ob dieser Ansatz für das vorliegende Problem im Vergleich zu dem BSB-Netz Vorteile bringt.

Das Mean Field Annealing ist eine Vereinfachung des Simulated Annealing [Kirk-

## Algorithmus 5.2 (Brain-state-in-a-box)

<b>Eingabe.</b>	Kostenfunktion $K(\mathbf{X})$ , Terminierungsbedingung $\mathcal{B}$ , Schrittweite $\Delta$ , Startwerte $\mathbf{X}^0$
<b>Ausgabe.</b>	Binärmatrix $\mathbf{B}$
<b>Algorithmus.</b>	<pre> (* Initialisierung *) <math>\forall i, k : x_{i,k} \leftarrow x_{i,k}^0</math>; <math>t \leftarrow 0</math>; (* Ganzes Netz schalten *) while <math>\neg \mathcal{B}</math> do (* Terminierungsbedingung *)   (* Alle Neuronen schalten *)   for <math>(i, k) \in \{1, \dots, N\} \times \{1, \dots, M\}</math> do     (* Eingangswert *)     <math>g_{i,k}(K(\mathbf{X}), t) \leftarrow 2 \sum_{j,l} a_{i,k,j,l} x_{j,l}(t) + b_{i,k}</math>;     (* Aktivierung *)     <math>x_{i,k}(t+1) \leftarrow x_{i,k}(t) - \Delta \cdot g_{i,k}(K(\mathbf{X}), t)</math>;     (* Ausgangswert *)     <math>x_{i,k}(t+1) \leftarrow \text{lin}(x_{i,k}(t+1))</math>;   od;   <math>t \leftarrow t + 1</math>; od; (* Interpretation *) <math>\forall i, k : b_{i,k} \leftarrow \text{round}(x_{i,k})</math>; return <math>\mathbf{B}</math>; </pre>

Tabelle 5.3: Brain-state-in-a-box Algorithmus

patrick *et al.* 83], das sich als Optimierungsverfahren für kombinatorische Optimierungsprobleme bewährt hat. Das vorliegende Subgraph-Isomorphie Problem ist als kombinatorisches Optimierungsproblem durch eine große Zahl untereinander abhängiger binärer Optimierungsvariable gekennzeichnet.

Das Simulated Annealing modelliert die Optimierungsvariable eines Problems als Freiheitsgrade einer Ansammlung von Atomen, die langsam in einen Grundzustand abgekühlt werden, der einer optimalen Lösung entspricht. Die Idee wird durch folgende physikalische Analogie verdeutlicht. Werden bestimmte Atome in ihrer Schmelze langsam abgekühlt, entsteht ein kristalliner Festkörper, der einen sehr regelmäßigen und damit energie-armen Zustand darstellt. Ist der Abkühlungsvorgang abrupt, so entsteht ein Glas, in dem Atome unregelmäßig und daher mit hohem Energiegehalt angeordnet sind. Genau wie bei Optimierungsverfahren eine optimale Lösung gesucht ist, ist beim Simulierten Abkühlen ein energieärmerer Zustand gesucht.

Das Simulated Annealing gehört zu der Klasse der Gradientenverfahren [Van Den Bout & Müller 90]. Das einfachste Gradientenverfahren ist das Verfahren des steilsten Abstiegs. Nachteilig an diesem Verfahren ist, daß eine aktuelle Lösung in einem lokalen Optimum nicht weiter verbessert werden kann. Das Simulated Annealing hat den Vorteil, daß das Verfahren in lokalen Minima nicht stecken bleiben muß.

Das Simulated Annealing kombiniert dazu einen Gradientenabstieg mit einem Zufallsprozeß. Um aus einem lokalen Minimum entkommen zu können, sind unter bestimmten Bedingungen auch solche Veränderungen der Optimierungsvariable erlaubt, die die Energie erhöhen. Veränderungen der Variable, die die Energie ver-

mindern, werden immer angenommen. Veränderungen, die eine Energieerhöhung hervorrufen, werden nur mit einer bestimmten Akzeptanzwahrscheinlichkeit angenommen. Das Simulated Annealing ist also ein stochastisches Verfahren.

In dem Prozeß des Simulated Annealing wird die Temperatur, von der die Akzeptanzwahrscheinlichkeit direkt abhängt, so vermindert, daß die Wahrscheinlichkeit der Akzeptanz von Energieerhöhungen schrittweise vermindert wird. Durch den Abkühlvorgang werden Umordnungen, die große Energieänderungen bewirken, auf hoher Temperaturstufe, und solche, die kleine Energieänderungen bewirken auf niedriger Temperaturstufe vorgenommen. Der Zustand im Nullpunkt stellt letztendlich eine suboptimale Lösung dar, die im allgemeinen nicht global optimal ist.

Die Physik befaßt sich ebenfalls mit Systemen, die eine große Anzahl interagierender Freiheitsgrade besitzen. Um Probleme solcher Systeme zu vereinfachen wird die Mean Field Approximation benutzt. Sie stellt eine analytische Approximation des Systemverhaltens von Teilchen oder Spins in thermischem Gleichgewicht dar. Das Mean Field Annealing ist durch die Mean Field Approximation aus dem Simulated Annealing abgeleitet. Die Approximation dient dazu, die Vorteile des Simulated Annealing mit einer schnellen deterministischen Berechnung zu verbinden. Dazu wird im folgenden die Mean Field Theorie kurz erläutert.

### Mean Field Theorie

Da die Mean Field Theorie das Verhalten eines Spinsystems auf einer Temperaturstufe beschreibt, stellt sie die Grundlage des MFA-Algorithmus dar. Die Mean Field Theorie wird daher hier kurz erläutert. Die Darstellung lehnt sich an die Bezeichnungen in [Bilbro *et al.* 89] und in [Van Den Bout & Miller 90] an.

Der Energiegehalt einer Spin-Konfiguration  $s$  wird durch die Hamilton-Funktion  $H(s)$  beschrieben:

$$H(s) = \sum_i \sum_j \beta_{ij} s_i s_j + \sum_i h_i s_i$$

Dabei wird angenommen, daß die Spin Interaktion  $\beta_{ij}$  symmetrisch, also  $\beta_{ij} = \beta_{ji}$  ist und keine Selbstinteraktion besitzt, das heißt  $\beta_{ii} = 0$ . Der Zustandsraum jedes Spins ist:

$$s_i \in \{0, 1\}$$

Im Simulated Annealing bewegen zufällige Umordnungen das System bei der momentanen Temperatur in Richtung seines thermischen Gleichgewichts. Unter der Annahme, daß sich alle Spins im thermischen Gleichgewicht befinden und alle anderen Spins  $s_j (i \neq j)$  konstant gehalten werden, kann man den Spinerwartungswert  $\langle s_i \rangle$  im Gleichgewichtszustand durch die Boltzmann-Verteilung und die Änderung der durchschnittlichen Systemenergie durch einen Wechsel des Spins  $i$  von  $s_i = 0$  auf  $s_i = 1$  bestimmen. Sei mit  $H_0$  bzw.  $H_1$  der Erwartungswert der Energie  $\langle H(s) \rangle |_{s_i=0}$  bzw.  $\langle H(s) \rangle |_{s_i=1}$  gegeben, wenn der Spin  $s_i = 0$  bzw. der Spin  $s_i = 1$  ist. Da das System Boltzmann-verteilt ist, wird der Wert von  $\langle s_i \rangle$  wie folgt berechnet:

$$\begin{aligned} \langle s_i \rangle &= 0 \cdot P(s_i = 0) + 1 \cdot P(s_i = 1) \\ &= \frac{\exp(-H_1/T)}{\exp(-H_0/T) + \exp(-H_1/T)} \\ &= \frac{1}{1 + \exp(-\Phi_i/T)} \quad \text{mit: } \Phi_i := H_0 - H_1 \end{aligned} \quad (5.2)$$

Der Graph der Funktion besitzt einen sigmoiden Verlauf und ist in Abb. 5.9 mit verschiedenen Werten für  $T$  abgebildet. Die Größe  $\Phi_i$  wird auch als Mean Field

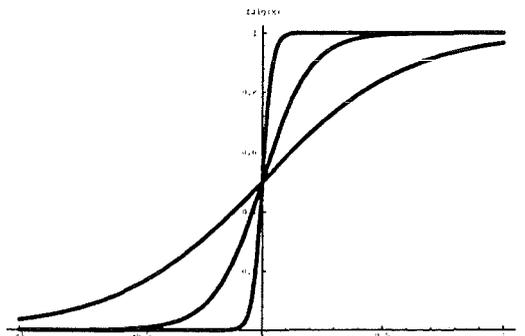


Abbildung 5.9: Sigmoidale Funktionen mit unterschiedlichen Werten für den Parameter  $T$ : Die Funktion dient zur Berechnung von Spinnerwartungswerten für das Mean Field Annealing. Werte für  $T > 0$  nahe bei 0 erzeugen einen steilen Kurvenverlauf, größere Werte  $T > 0$  erzeugen einen abgeflachten Kurvenverlauf.

bezeichnet, das den Spin  $i$  beeinflusst. Allerdings ist es im allgemeinen schwierig,  $\Phi_i$  für eine große Anzahl  $N$  an Spins zu berechnen:

$$\begin{aligned} \langle H(\mathbf{s}) \rangle &= \left\langle \sum_{i=1}^N \sum_{j=1}^N \beta_{ij} s_i s_j + \sum_{i=1}^N h_i s_i \right\rangle \\ &= \sum_{i=1}^N \sum_{j=1}^N \beta_{ij} \langle s_i s_j \rangle + \sum_{i=1}^N h_i \langle s_i \rangle \end{aligned}$$

Da  $s_i$  und  $s_j$  nicht unabhängig sind, sind die Erwartungswerte in  $\langle s_i s_j \rangle$  nicht separierbar. Unter der Annahme, daß die Zahl interagierender Spins groß genug ist und der Beitrag eines einzelnen Spins zu anderen Spins im Vergleich zu dem Gesamtfeld sehr klein ist, kann aber die folgende Mean Field Approximation benutzt werden:

$$\langle s_i s_j \rangle = \langle s_i \rangle \langle s_j \rangle$$

Das Mean Field  $\Phi_i$  ergibt sich nun als negative partielle Ableitung von  $\langle H(\mathbf{s}) \rangle$  nach  $\langle s \rangle_i$ :

$$\begin{aligned} \Phi_i &= H_0 - H_1 \\ &= -2 \sum_j \beta_{ij} \langle s \rangle_j - h_i \\ &= - \frac{\partial \langle H(\mathbf{s}) \rangle}{\partial \langle s \rangle_i} \end{aligned} \tag{5.3}$$

Nun sind bei jeder Temperatur die Spinnerwartungswerte  $\langle \mathbf{s} \rangle$  gesucht, so daß für jeden Spin die Gleichungen 5.3 und 5.2 erfüllt sind.

Die Lösung oder der Fixpunkt läßt sich ermitteln, indem ein Startvektor der Spinnerwartungswerte benutzt wird, um die Mean Field Komponenten zu berechnen. Der Mean Field Vektor kann dann benutzt werden, um die Spinnerwartungswerte wieder neu zu berechnen. Draufhin wird der Prozess wiederholt, bis sich die Erwartungswerte für die momentane Temperatur stabilisiert haben. Dieser Prozess

**Algorithmus 5.3 (Mean Field Annealing)**

<b>Eingabe.</b>	Kostenfunktion $K(\mathbf{X})$ , Schrittweite $\Delta$ , Startwerte $\mathbf{X}^0$ , Abkühlplan $(T_a, a(T), n(T))$ , Terminierungsbed. $\mathcal{B}$
<b>Ausgabe.</b>	Binärmatrix $\mathbf{B}$
<b>Algorithmus.</b>	<pre> (* Initialisierung *) <math>\forall i, k : x_{i,k} \leftarrow x_{i,k}^0</math>; <math>t \leftarrow 0</math>; <math>T \leftarrow T_a</math>; (* iterative Verbesserung *) while <math>\neg \mathcal{B}</math> do (* Terminierungsbedingung *)   (* Abkühlen auf Temperaturstufe *)   for <math>p \leftarrow 1</math> to <math>n(T)</math> do     (* Zustandsänderung *)     for <math>(i, k) \in \{1, \dots, N\} \times \{1, \dots, M\}</math> do       (* Gradient *)       <math>g_{i,k}(K(\mathbf{X}), t) \leftarrow 2\sum_{j,l} a_{i,k,j,l} x_{j,l}(t) + b_{i,k}</math>;       (* Mean Field *)       <math>x_{i,k}(t+1) \leftarrow x_{i,k}(t) -</math>         <math>\Delta(g_{i,k}(K(\mathbf{X}), t) + x_{i,k}(t));</math>       (* Spin-Erwartungswert *)       <math>x_{i,k}(t+1) \leftarrow (1 + \exp(-x_{i,k}(t+1)/T))^{-1}</math>;     od;   od;   <math>T \leftarrow a(T)</math>; od; (* Interpretation *) <math>\forall i, k : b_{i,k} \leftarrow \text{round}(x_{i,k})</math> return <math>\mathbf{B}</math>; </pre>

Tabelle 5.4: Mean Field Annealing Algorithmus

ist allerdings sequentiell: ein Spin wird selektiert, neu berechnet, worauf der neue Wert andere beeinflusst.

Man erhält eine parallele Variante, indem man die neuen Mean Field Werte parallel berechnet und ausgehend von dem momentanen Wert nur einen bestimmten Betrag in Richtung des neuen Mean Field Wertes rückt. Die Spinerwartungswerte werden dann aufgrund der dazwischenliegenden Werte berechnet und der Prozeß wird iteriert. Die parallele Variante ist durch  $N$  Differenzgleichungen charakterisiert:

$$\Phi_i(t+1) = \Phi_i(t) + \Delta \left( -2 \sum_j \beta_{ij} \langle s_j(t) \rangle - h_i - \Phi_i(t) \right)$$

$$\langle s_i(t+1) \rangle = \frac{1}{1 + \exp(-\Phi_i(t+1)/T)}$$

Vermindert man nun den Temperaturparameter iterativ in bestimmter Art und Weise und führt pro Temperaturstufe die geschilderte Iteration durch, erhält man den MFA-Algorithmus.

## Der MFA-Algorithmus

Um den MFA-Algorithmus auf das vorliegende Korrespondenzproblem anwenden zu können, müssen die Bestandteile der iterativen Verbesserung und damit des zu lösenden Optimierungsproblems auf den MFA-Algorithmus abgebildet werden.

Der MFA-Algorithmus besitzt hohe Ähnlichkeit mit neuronalen Netzen. Der Spinnerwartungswert läßt sich als Neuronenausgabe betrachten, der interne Zustand als Neuronenaktivierung, das Mean Field als Neuroneneingangswert und die Hamilton-Funktion als Netzenergiefunktion. Alle problem-spezifischen Parameter des BSB-Netzes sowie Initialisierung und Interpretation der iterativen Verbesserung können übernommen werden, da sich nicht das Problem, sondern das Optimierungsverfahren ändert.

Neben der Schrittweite muß also zusätzlich zu diesen Parametern ein geeigneter Abkühlplan festgelegt werden, der die Temperaturabnahme beschreibt. Theoretisch kann man nachweisen, daß dabei mit Wahrscheinlichkeit 1 das globale Minimum erreicht wird, wenn man äußerst langsam abkühlt:  $T(t) = T_a / \ln(t)$  [Herault & Niez 91]. In der Praxis kühlt man aus Zeitgründen jedoch weitaus schneller ab, zum Beispiel mit  $T(n) = T_a / q^n$ ,  $q = 1,06$  [Braun 90]. Da auf jeder Temperaturstufe mehrere Zustandsänderungen vorgenommen werden sollen, bis der Systemzustand stabil ist [Van Den Bout & Miller 90], wird ein stufenweiser Abkühlplan verwendet. Dieser ist gekennzeichnet durch eine Anfangstemperatur  $T_a$ , eine kontinuierliche Temperaturabnahmefunktion  $a(T)$  und der Anzahl  $n(T)$  von Zustandsänderungen pro Temperatur. Ein Abkühlplan ist also ein Tripel  $(T_a, a(T), n(T))$ .

Um den MFA-Algorithmus für das Korrespondenzproblem besser formulieren zu können, werden die Spinnerwartungswerte  $\langle s_i \rangle$  durch eine Indextransformation in die Matrixelemente  $x_{i,k}$  der Spinmatrix  $\mathbf{X}$  überführt. Damit kann der Algorithmus wie in Tab. 5.4 notiert werden.

### 5.2.5 Relaxationsmarkierung

Die Zuordnung von Knoten eines Objektgraphen zu Knoten eines Szenengraphen stellt ein ähnliches Problem dar, das in [Rosenfeld *et al.* 76] untersucht wurde. Ausgangspunkt ist eine Menge von Objekten in einer Szene, deren Identität mehrdeutig ist. Anhand von Beziehungen zwischen den Objekten werden durch iterierte parallele Operationen, die als Relaxationsoperationen bezeichnet werden, Mehrdeutigkeiten bzgl. der Identität von Objekten reduziert oder gar eliminiert. Interpretiert man die Objekte als Objektknoten und die Marken als Szeneknoten, so ist es möglich, durch parallele Operationen schrittweise mehrdeutige Zuordnungen von Objekt- zu Szeneknoten so zu eliminieren, daß im idealen Fall die Zuordnungen eindeutig sind.

Den Ursprung hat die Relaxationsmarkierung in der Interpretation von Strichzeichnungen aus Videobildern. Von Rosenfeld wurde der Ansatz auf die parallele Ausführung von kontinuierlichen Relaxationsoperationen verallgemeinert. Hier wird das klassische nicht-lineare probabilistische Modell untersucht, das auch in vielen Korrespondenzproblemen in der Mustererkennung verwendet wurde. Einen umfassenden Überblick über Anwendungen und Erweiterungen der Relaxationsmarkierung bietet der Artikel von [Kittler & Illingworth 85].

Die oft genannten Nachteile der nicht-linearen probabilistischen Relaxationsmarkierung bestehen in folgenden fünf Punkten [Lloyd 83]: (1) Im allgemeinen ist keine Konvergenz garantiert. (2) Das Resultat ist stark von den Startwahrscheinlichkeiten abhängig. (3) Konvergiert die Iteration, so kann das Resultat unerwünscht sein. (4) Die Relaxationsmarkierung besitzt eine kleine theoretische Basis. (5) Nur in

der frühen Phase der Relaxation werden Mehrdeutigkeiten entfernt, später werden Lösungen oft schlechter. Dennoch sind folgende Fragen von Interesse:

- Welche Unterschiede und Gemeinsamkeiten besitzt die NLR-Markierung im Vergleich mit dem BSB-Netz und dem MFA-Algorithmus ?
- Wie müßte die NLR-Markierung an das hier vorliegende Problem angepaßt werden ?

Dazu wird im folgenden die nicht-lineare probabilistische Variante der Relaxationsmarkierung kurz erläutert.

### Nicht-lineare probabilistische Relaxationsmarkierung

Gegeben sei die Objektmenge  $A$  bestehend aus Objekten  $a_i$  und die Menge  $\Lambda$  bestehend aus Marken  $\lambda_j$ :

$$\begin{aligned} A &=_{\text{def}} \{a_1, \dots, a_i, \dots, a_N\} \\ \Lambda &=_{\text{def}} \{\lambda_1, \dots, \lambda_k, \dots, \lambda_M\} \end{aligned}$$

Die Variable  $\theta_i$  hat die Menge aller Marken als Wertebereich:

$$\theta_i \in \{\lambda_1, \dots, \lambda_M\}$$

Die Zuordnung einer Marke zu einem Objekt kann auch als Interpretation betrachtet werden. Sollen auch mehrdeutige Interpretationen zugelassen werden, kann man die Wahrscheinlichkeit betrachten, daß dem Objekt  $a_i$  die Marke  $\lambda_k$  zugeordnet ist:

$$P(\theta_i = \lambda_k) \text{ oder kurz: } P_{i,k}$$

Aufgrund der Deutung mit Wahrscheinlichkeiten, ergeben sich die folgenden Bedingungen:

$$\begin{aligned} \forall i : P_{i,k} &\in [0, 1] \\ \forall i : \sum_q P_{i,q} &= 1 \end{aligned} \quad (5.4)$$

Die Koeffizienten  $r$  beschreiben den Grad an Kompatibilität (Konsistenz, Verträglichkeit) und an Widerspruch zwischen zwei Objekt-Marke Zuordnungen bzw. zwischen zwei Interpretationen:

$$r(\theta_i = \lambda_k, \theta_j = \lambda_l) \text{ oder kurz: } r_{i,k,j,l}$$

Die Koeffizienten haben für ihren Wertebereich die folgende Bedeutung:

$$\begin{aligned} r_{i,k,j,l} &\in [-1, 1] \\ r_{i,k,j,l} &= \begin{cases} < 0, & a_i \mapsto \lambda_k \text{ mit } a_j \mapsto \lambda_l \text{ inkompatibel mit Grad } |r_{i,k,j,l}| \\ = 0, & a_i \mapsto \lambda_k \text{ von } a_j \mapsto \lambda_l \text{ unabhängig} \\ > 0, & a_i \mapsto \lambda_k \text{ ist mit } a_j \mapsto \lambda_l \text{ kompatibel mit Grad } r_{i,k,j,l} \end{cases} \end{aligned}$$

Das Ziel besteht nun darin, ausgehend von Startwerten für die Wahrscheinlichkeiten parallel und iterativ Mehrdeutigkeiten mit Hilfe einer Iterationsvorschrift zu reduzieren, und damit möglicherweise zu einer eindeutigen Objekt-Marke Zuordnung zu gelangen.

Die Wahrscheinlichkeiten im Iterationsschritt  $t$  werden von nun an als  $P_{i,k}^{(t)}$  geschrieben. Die Gesamtkompatibilität  $Q^{(t)}(\theta_i = \lambda_k)$ , oder kurz  $Q_{i,k}^{(t)}$  der Zuordnung  $a_i \mapsto \lambda_k$  unter Kenntnis der Kompatibilitäten aller Zuordnungen ist definiert als:

$$Q_{i,k}^{(t)} = \sum_j C_{i,j} \sum_l r_{i,k,j,l} P_{j,l}^{(t)}$$

mit:

$$\sum_j C_{i,j} = 1$$

Durch die Gewichtskoeffizienten  $C_{i,j}$  kann die relative Wichtigkeit einzelner Objektpaare ausgedrückt werden. Die Iterationsvorschrift der NLR-Markierung für die Wahrscheinlichkeit einer Objekt-Marke Zuordnung ist:

$$P_{i,k}^{(t+1)} = P_{i,k}^{(t)} \cdot \frac{Q_{i,k}^{(t)}}{\sum_r P_{i,r}^{(t)} Q_{i,r}^{(t)}}$$

wobei der Nenner einen Normalisierungsfaktor darstellt, der die Einhaltung der Wahrscheinlichkeitsbedingung 5.4 garantiert. Sind die  $P_{i,k}$  in einer Matrix  $\mathbf{P}$  angeordnet, erzwingt die Normalisierung also, daß die Summe in jeder Spalte der Matrix  $\mathbf{P}$  gleich Eins ist.

Um die Iteration starten zu können, benötigt man noch Startwerte  $P_i^{(0)}(\lambda_k)$  für die Wahrscheinlichkeiten. Die Startwerte können als Wahrscheinlichkeiten ohne Kontext interpretiert werden.

### Anpassung des Modells

Um die NLR-Markierung auf das vorliegende Korrespondenzproblem anwenden zu können, muß die NLR-Markierung durch erlaubte Transformationen auf die Bestandteile der iterativen Verbesserung abgebildet und damit auf das zu lösende Optimierungsproblem angepaßt werden.

Mit der Interpretation der Objekte  $a_i$  als Objektknoten und der Interpretation der Marken  $\lambda_k$  als Szenenknotten werden vier zulässige und plausible Anpassungen vorgenommen:

- symmetrische Kompatibilitätskoeffizienten
- Kompatibilitätskoeffizienten aus dem Einheitsintervall
- gleiche, nicht-normierte Wichtungsfaktoren  $C_{ij}$
- Entfernen der Normierung der Spaltensumme auf 1

Da die Kompatibilität der Zuordnung eines Objektknotenpaares  $a_i, a_j$  zu einem Szenenknottenpaar  $\lambda_k, \lambda_l$  der Kompatibilität der umgekehrten Zuordnung entsprechen soll, sind die Kompatibilitäten symmetrisch, d.h.  $r_{i,k,j,l} = r_{j,l,i,k}$ .

Zweitens können die Kompatibilitätskoeffizienten durch die folgende lineare Transformation von dem Intervall  $[-1, 1]$  in das Intervall  $[0, 1]$  abgebildet werden:

$$r_{i,j,k,l} \leftarrow \frac{1}{2}(r_{i,j,k,l} + 1)$$

Dabei sind zwei Zuordnungen inkompatibel (unabhängig, kompatibel), wenn der entsprechende Koeffizient  $r_{i,k,j,l} = 0 (= 1/2, = 1)$  ist.

Drittens können alle  $C_{i,j}$  auf einen gemeinsamen Wert gesetzt werden, da Objektknoten keine unterschiedlichen Wichtigkeiten besitzen. Wie in [Lloyd 83] wird daher angenommen, daß die Bedingung  $\sum_i C_{i,j} = 1$  nicht erfüllt sein muß, sondern vielmehr alle  $C_{i,j} = 1$  gesetzt werden. Mit diesen Änderungen stellt die NLR-Markierung nach [Lloyd 83] ein Minimierungsverfahren für die folgende Funktion  $F$  dar, die als Inkompatibilitätsfunktion bezeichnet werden kann. Dabei können die Wahrscheinlichkeiten  $P_{i,k}$  zu einer Matrix  $\mathbf{P}$  zusammengefaßt werden:

$$\begin{aligned} F(\mathbf{P}) &= \frac{1}{2} \sum_{i,k} \sum_{j,l} P_{i,k} Q_{i,k} \\ &= \frac{1}{2} \sum_{i,k,j,l} r_{i,k,j,l} P_{i,k} P_{j,l} \end{aligned}$$

Durch die NLR-Markierung wird die Inkompatibilitätsfunktion allerdings unter der Nebenbedingung minimiert, daß die Summe aller Zuordnungswahrscheinlichkeiten eines Objektes gleich 1 sein muß. Dies wird durch die Normierung in Gleichung 5.4 erreicht. Das heißt, es wird nicht erlaubt, daß einem Objektknoten *kein* Szeneknoten zugeordnet wird. Diese Bedingung hätte die Wirkung, daß der Objektgraph auf jeden Fall vollständig in dem Szenengraphen enthalten sein muß. Anders ausgedrückt, sind keine partiellen Subgraphisomorphismen zwischen Objekt- und Szenengraphen zugelassen. Aufgrund teilweiser Sichtbarkeit von Objekten darf man das Einhalten der Bedingung nicht fordern. Die Normierung der Spaltensumme auf 1 wird daher entfernt.

Wird weiterhin nach jeder Kompatibilitätsverrechnung mit Hilfe der Funktion  $f_{lin}$  die folgende Projektion in das Einheitsintervall  $[0, 1]$  durchgeführt:

$$P_{i,k} \leftarrow f_{lin}(P_{i,k})$$

erhält man ein Optimierungsverfahren, um das folgende quadratische Optimierungsproblem zu lösen.

### Problem 5.3 (Angepaßtes Markierungsproblem)

$$\begin{array}{l} \text{Sei} \quad \mathbf{P} \in \mathbb{R}^{N \cdot M} \\ \text{Minimiere} \quad F(\mathbf{P}) = \frac{1}{2} \sum_{i,k,j,l} r_{i,k,j,l} P_{i,k} P_{j,l} \\ \text{bzgl.} \quad \forall i, k : 0 \leq P_{i,k} \leq 1 \end{array}$$

Dieses Problem ist identisch mit dem kontinuierlichen Ersatzproblem 5.2. Die Wahrscheinlichkeitsmatrix  $\mathbf{P}$  entspricht der Matrix  $\mathbf{X}$  der Optimierungsvariable und die Kostenfunktion  $K$  entspricht der Inkompatibilitätsfunktion  $F$ . In der Funktion  $F$  sind zwar keine linearen Terme vorhanden, jedoch können sie durch quadratische Produkte mit einem konstanten Faktor realisiert werden.

Um die NLR-Markierung besser in das Schema der iterativen Verbesserung einordnen zu können, kann die Iterationsvorschrift der NLR-Markierung:

$$P_{i,k}^{(t+1)} = P_{i,k}^{(t)} Q_{i,k}^{(t)}$$

**Algorithmus 5.4 (Angepaßte nicht-lineare Relaxationsmarkierung)**

<b>Eingabe.</b>	Kostenfunktion $K(\mathbf{X})$ , Terminierungsbedingung $\mathcal{B}$ , Startwerte $\mathbf{X}^0$
<b>Ausgabe.</b>	Binärmatrix $\mathbf{B}$
<b>Algorithmus.</b>	<pre> (* Initialisierung *) <math>\forall i, k : x_{i,k} \leftarrow x_{i,k}^0</math>; <math>t \leftarrow 0</math>; <math>\Delta \leftarrow 1</math>; (* iterative Verbesserung *) while <math>\neg \mathcal{B}</math> do (* Terminierungsbedingung *)   (* Mehrdeutigkeiten eliminieren *)   for <math>(i, k) \in \{1, \dots, N\} \times \{1, \dots, M\}</math> do     (* Richtung *)     <math>g_{i,k}(K(\mathbf{X}), t) \leftarrow \frac{1}{2} x_{i,k}(t) (\sum_{j,l} a_{i,k,j,l} x_{j,l}(t) - 1)</math>;     (* Wahrscheinlichkeit einer Zuordnung *)     <math>x_{i,k}(t+1) \leftarrow x_{i,k}(t) + \Delta \cdot g_{i,k}(K(\mathbf{X}), t)</math>;     (* Projektion der Wahrscheinlichkeit *)     <math>x_{i,k}(t+1) \leftarrow \text{flin}(x_{i,k}(t+1))</math>;   od;   <math>t \leftarrow t + 1</math>; od; (* Interpretation *) <math>\forall i, k : b_{i,k} \leftarrow \text{round}(x_{i,k})</math>; return <math>\mathbf{B}</math>; </pre>

Tabelle 5.5: Algorithmus der angepaßten nicht-linearen Relaxationsmarkierung

nach [Lloyd 83] auch wie folgt umgeformt werden:

$$\begin{aligned}
 P_{i,k}^{(t+1)} &= P_{i,k}^{(t)} + \Delta \cdot g_{i,k}^{(t)}(\mathbf{P}^{(t)}) \\
 \text{mit:} \\
 \Delta &= 1 \\
 g_{i,k}^{(t)}(\mathbf{P}^{(t)}) &= P_{i,k}^{(t)}(Q_{i,k}^{(t)} - 1) \\
 &= P_{i,k} \left( \sum_{j,l} r_{i,k,j,l} P_{j,l} - 1 \right)
 \end{aligned}$$

Man erhält eine Modifikation der ursprünglichen probabilistischen NLR-Markierung, die verwandt mit dem BSB-Netz ist. Unterschiede liegen in der Verbesserungsrichtung und der sehr großen, nicht-beliebigen Schrittweite  $\Delta = 1$ . Die modifizierte NLR-Markierung kann nicht mehr als probabilistisch bezeichnet werden, da auf die Wahrscheinlichkeitsbedingung, d.h. auf die Normierung auf 1, verzichtet wurde. Der Algorithmus kann nach dem Schema der iterativen Verbesserung in Tab. 5.5 notiert werden.

**5.2.6 Gegenüberstellung der Verfahren**

BSB-Netz, Mean Field Annealing und die angepaßte NLR-Markierung arbeiten als Gradientenverfahren nach dem Schema der iterativen Verbesserung. Aufgrund ver-

schiedener Schwerpunkte wurden Variablenbezeichner und Fachbegriffe in den Abschnitten 5.2.1 bis 5.2.4 unterschiedlich gewählt:

- Die theoretischen Grundlagen der drei Verfahren und des Schemas der iterativen Verbesserung werden jeweils mit den Bezeichnern in Anlehnung an die jeweilige Literatur beschrieben.
- Die entsprechenden Algorithmen 5.1 bis 5.4 sind jeweils in der Notation der iterativen Verbesserung beschrieben, um die Unterschiede und Gemeinsamkeiten im algorithmischen Ablauf für das vorliegende Problem zu verdeutlichen.
- In der anschließenden Tabelle stellen die Variablenbezeichner und Fachbegriffe eine Kombination der vorangegangenen Schreibweisen dar. Die Variablenbezeichner und Fachbegriffe entsprechen denjenigen aus der Literatur, während die Variablenindizes aufgrund des vorliegenden Problems zweidimensional gewählt sind.

Eine genaue Gegenüberstellung der Bestandteile bzgl. der iterativen Verbesserung zeigt Tabelle 5.6. Unterschiede der drei Verfahren liegen in der Richtung des Verbesserungsschritts (Zeile 3), in der Schrittweite (Zeile 4) und in der Variablenprojektion (Zeile 6). Die übrigen Bestandteile (Zeilen 1, 2, 5, 7, 8) besitzen in verschiedenen Verfahren lediglich verschiedene Begriffe und Variablenbezeichner.

Die Verbesserungsrichtung im BSB-Netz entspricht genau dem Gradienten der Kostenfunktion. Zusammen mit der Schwellwertfunktion  $\pi$  realisiert das BSB-Netz einen projizierten Gradientenabstieg. Die Verbesserungsrichtung des MFA-Algorithmus verwendet aufgrund der Variablenprojektion zur Berechnung von  $\langle s_{i,k} \rangle$  erst bei  $T = 1$  den Gradienten der Kostenfunktion. Die Verbesserungsrichtung der NLR-Markierung entspricht nicht den Gradienten der Kostenfunktion, so daß das Verfahren lokal nicht in Richtung der geringsten Kosten geht. Auf die Verbesserungsrichtung und die feste, nicht-beliebige Schrittweite sind die auf Seite 112 genannten Nachteile der NLR-Markierung zurückzuführen.

### Arten der Parallelität

An dieser Stelle sollen keine speziellen Hardware-Architekturen und keine Abbildungen auf Hardware-Architekturen behandelt werden, die zur Simulation neuronaler Netze geeignet sind. Solche Hardware-Architekturen werden in [Zell 94, p. 452] ausführlich anhand von Beispielen beschrieben. Vielmehr sollen die Arten der Parallelität aufgezeigt werden, die den in dieser Arbeit vorgestellten Verfahren der iterativen Verbesserung gemeinsam ist.

Da viele Arbeiten über die hier vorgestellte Klasse von Algorithmen im Zusammenhang mit neuronalen Netzen publiziert werden, werden im folgenden auch die Fachbegriffe aus dem Bereich der neuronalen Netze verwendet.

In [Zell 94] und [d’Acierno & Vaccaro 94] werden verschiedene Arten der Parallelität von neuronalen Netzen erläutert. Da die Gewichte der hier vorliegenden Netze nicht trainiert, sondern errechnet werden, ist die Art der Parallelität nur für die Arbeitsphase relevant. Folgende Arten der Parallelität lassen sich unterscheiden:

**Neuronenparallelität:** Bei der Neuronenparallelität werden die Aktivierungen aller Neuronen parallel berechnet. Jedes Neuron wird durch einen eigenen Prozeß berechnet.

Nr.	Schema „Iterative Verbesserung“	BSB-Netz	MFA-Algorithmus	angepaßte NLR-Markierung
1	Optimierungsvariable $x_{i,k}(t)$	Aktivierung $m_{i,k}(t)$	Mean Field $\Phi_{i,k}(t)$	„Wahrscheinlichkeit“ einer Objekt-Marke Zuordnung $P_{i,k}(t)$
2	Kostenfunktion $K(\mathbf{X}) = \sum_{i,k,j,l} a_{i,k,j,l} x_{i,k} x_{j,l} + \sum_{i,k} b_{i,k} x_{i,k}$	Energiefunktion $E(\mathbf{N}) = \sum_{i,k,j,l} w_{i,k,j,l} n_{i,k} n_{j,l} + \sum_{i,k} \Theta_{i,k} n_{i,k}$	Hamilton-Funktion $H(\mathbf{S}) = \sum_{i,k,j,l} \beta_{i,k,j,l} \langle s_{i,k} \rangle \langle s_{j,l} \rangle + \sum_{i,k} h_{i,k} \langle s_{i,k} \rangle$	Inkonsistenzfunktion $F(\mathbf{P}) = \frac{1}{2} \sum_{i,k,j,l} r_{i,k,j,l} P_{i,k} P_{j,l}$
3	Richtung $g_{i,k}(K(\mathbf{X}), t) = 2 \sum_{j,l} a_{i,k,j,l} x_{j,l}(t) + b_{i,k}$	Richtung $g_{i,k}(E(\mathbf{N}), t) = 2 \sum_{j,l} w_{i,k,j,l} n_{j,l}(t) + \Theta_{i,k}$	Richtung $g_{i,k}(H(\mathbf{S}), t) = 2 \sum_{j,l} \beta_{i,k,j,l} \langle s_{j,l} \rangle (t) + h_{i,k}$	Richtung $g_{i,k}(F(\mathbf{P}), t) = P_{i,k}^{(t)} (\sum_{j,l} r_{i,k,j,l} P_{j,l}^{(t)} - 1)$
4	Schrittweite $\Delta$	beliebig fest	beliebig fest	$\Delta = 1$
5	Verbesserungsschritt $x_{i,k}(t+1) \leftarrow x_{i,k}(t) - \Delta \cdot g_{i,k}(K(\mathbf{X}), t)$	Schalten eines Neurons $m_{i,k}(t+1) \leftarrow m_{i,k}(t) - \Delta \cdot g_{i,k}(E(\mathbf{N}), t)$	mikroskopische Umordnung $\Phi_{i,k}(t+1) \leftarrow \Phi_{i,k}(t) - \Delta (g_{i,k}(H(\mathbf{S}), t) + \Phi_{i,k}(t))$	Kompatibilitätsverrechnung $P_{i,k}^{(t+1)} \leftarrow P_{i,k}^{(t)} + \Delta \cdot g_{i,k}(F(\mathbf{P}), t)$
6	projizierte Variable $\pi(x_{i,k})$	Neuronenausgang $n_{i,k} = \text{flin}(m_{i,k})$	Spinerwartungswert $\langle s_{i,k} \rangle = [1 + \exp(-\Phi_{i,k}/T)]^{-1}$	projizierte Wahrscheinlichkeit $\text{flin}(P_{i,k})$
7	Koeffizient $a_{i,k,j,l}$ von Produkten in $K$	Verbindungsgewicht $w_{i,k,j,l}$	paarweise Spin-Interaktion $\beta_{i,k,j,l}$	paarweise Inkompatibilität $r_{i,k,j,l}$
8	Koeffizient $b_{i,k}$ linearer Teilterme in $K$	Schwelle $\Theta_{i,k}$	externe Feldkomponente $h_{i,k}$	-

Tabelle 5.6: Gegenüberstellung der Optimierungsverfahren im einheitlichen Rahmen der iterativen Verbesserung (siehe Text).

**Synapsenparallelität:** Bei der Synapsenparallelität wird jede Eingabe eines Neurons mit dem entsprechenden Gewicht multipliziert. Dieser Prozeß kann parallel für alle Eingaben eines Neurons durchgeführt werden.

**Netztopologie-Parallelität:** Bei der Netztopologieparallelität werden mehrere neuronale Netze parallel simuliert. Jedes neuronale Netz wird durch einen eigenen Prozeß simuliert. Netze, die sich z.B. in der Topologie unterscheiden, können gleichzeitig arbeiten.

Die Netztopologie-Parallelität kann in der hier vorliegenden Anwendung der Objekterkennung in zwei Stufen ablaufen. In der ersten Stufe werden alle Objektgraphen parallel mit dem Szenengraphen verglichen, in der zweiten Stufe wird für alle Objektmodelle parallel entschieden, welcher Objektgraph des Objektmodells die beste Erkennungsgüte besitzt. Diese Art der Parallelität ist aber prinzipiell auch realisierbar, indem Iterationen im batch-Betrieb auf vernetzten Workstations gestartet werden, und ist auch auf nicht-neuronale Verfahren, beispielweise Baumsuchverfahren übertragbar.

Im Zusammenhang mit der Parallelität ist auch die Schaltdynamik von Bedeutung. Die Schaltdynamik gibt an, in welcher Reihenfolge die Neuronen ihre Eingaben erhalten und in welcher Reihenfolge sie ihre Ausgaben berechnen. Im Falle der synchronen Schaltdynamik ändern alle Neuronen ihre Werte gleichzeitig, d.h. alle Neuronen berechnen im ersten Schritt die neue Aktivierung ihrer Zellen, danach berechnen sie im zweiten Schritt die Ausgaben und geben sie weiter. In einer sequentiellen Implementierung müssen alle Neuronen ihre neue Aktivierung erst intern berechnen und dürfen sie erst dann, wenn die Berechnung der neuen Aktivierung aller Neuronen abgeschlossen ist, nach außen weitergeben. Im Falle der asynchronen Schaltdynamik ändern einzelne Neuronen ihre Werte zu verschiedenen Zeitpunkten oder in einer bestimmten festgelegten oder zufälligen Reihenfolge.

Alle bekannten Parallelisierungsansätze für rekurrente neuronale Netze basieren auf dem Skalarprodukt zweier  $n$ -dimensionaler Vektoren. Das Skalarprodukt enthält  $n$  skalare Multiplikationen und eine Summenbildung über  $n$  Zahlen. Die Schaltdynamik definiert die Zahl der Skalarprodukte, die parallel ausgewertet werden können. Der Grad der Parallelität, d.h. die maximale Anzahl der parallel ausführbaren Operationen, ist begrenzt durch die Schaltdynamik. Prinzipiell erwünscht ist also eine möglichst synchrone Schaltdynamik. Während vorwärtsgerichtete neuronale Netze bisher am besten auf eine parallele Implementierung untersucht wurden [Zell 94], haben sich bisher nur wenige Arbeiten mit der Parallelisierung rekurrenter neuronaler Netze befaßt. Diese werden im folgenden dargestellt.

Barbosa und Lima [Barbosa & Lima 90] bilden den Zustand eines Neurons und eine Zeile der Verbindungsmatrix auf einen Prozeß ab. Das heißt, das Skalarprodukt und die Aktivierung eines Neurons werden von einem Prozeß berechnet. Ein zusätzlicher, übergeordneter Masterprozeß initialisiert die Neuronen, sammelt die Neuronenwerte, prüft ob die Terminierungsbedingung erfüllt ist oder sendet ein Terminierungssignal. Das Verfahren kann in einer asynchron arbeitenden Variante verwendet werden, wenn die Verbindungsmatrix dünn besetzt ist. Das Verfahren ist also direkt von den Nullelementen der Verbindungsmatrix abhängig. Gerade in dem hier behandelten Korrespondenzproblem ist die Verbindungsmatrix dünn besetzt, da in der Matrix nur dann ein Wert ungleich Null steht, wenn ein Objektknotenpaar einem Szenenknotenpaar ähnlich ist. Experimentelle Ergebnisse werden für die Abbildung der Prozesse auf ein Transputernetzwerk vorgestellt.

In der Arbeit von [d'Acierno & Vaccaro 94] wird ein Netzwerk in Teilnetzwerke unterteilt. Dabei wird eine Mischung aus synchroner und asynchroner Schaltdynamik verwendet, die als blocksequentielle Schaltdynamik bezeichnet wird. Dabei wird

der  $n$ -dimensionale Zustandsvektor aller Neuronen in  $K$  Teilvektoren unterteilt, die asynchron berechnet werden. Die Elemente innerhalb der Teilvektoren hingegen werden synchron berechnet. Es handelt sich also um eine Neuronenparallelität. Die blocksequentielle Schaltdynamik ist mit  $K = n$  eine Verallgemeinerung der asynchronen und mit  $K = 1$  eine Verallgemeinerung der synchronen Schaltdynamik. Es wird ein Array aus Prozessen vorgestellt, in dem in jeder Zeile ein Skalarprodukt berechnet wird. Da nicht nur die Berechnung der Neuronenausgänge, sondern auch die Gewichtung der Neuroneneingänge parallel durchgeführt wird, handelt es sich um Neuronen- und Synapsenparallelität. Verschiedene Zeilen werden zu Blöcken zusammengefaßt, die mit der blocksequentuellen Schaltdynamik geschaltet werden. Das Netz kann auf verschiedene Parallelrechnerarchitekturen abgebildet werden. Besonders eignen sich grob-körnige Rechnerarchitekturen, wobei ein Teilnetz durch ein Prozessor berechnet werden kann. Experimentelle Ergebnisse werden für eine Implementierung auf einem Transputer-Cluster vorgestellt. Andere Arbeiten berücksichtigen spezielle Rechnerarchitekturen [Kung & Hwang 89] oder erlauben nur asynchrone Schaltdynamik [Eun *et al.* 91].

## 5.3 Experimentelle Ergebnisse

### 5.3.1 Parameterbestimmung

Der BSB-Algorithmus enthält freie Parameter und Prozedurabschnitte der iterativen Verbesserung, die geeignet festgelegt werden müssen. Die freien Parameter lassen sich klassifizieren je nach dem, ob sie für das Optimierungsverfahren oder das Optimierungsproblem spezifisch sind:

- Komponenten der iterativen Verbesserung:
  - Initialisierungswerte  $\mathbf{X}^0$
  - Terminierungsbedingung  $\mathcal{B}$
  - Interpretation  $\mathcal{I}$
- Parameter, spezifisch für Optimierungsverfahren: Schrittweite  $\Delta$
- Parameter, spezifisch für Optimierungsproblem:
  - Wichtungparameter  $C_P, C_K, C_{SZ}, C_{01}$  von Teilkosten
  - Parameter  $a_L, b_L, c_L$  bzw.  $a_W, b_W, c_W$  in numerischen Ähnlichkeitsfunktionen bezogen auf Länge bzw. Winkel

#### Initialisierung und Schrittweite

Komponenten der iterativen Verbesserung sind die Terminierungsbedingung und die Initialisierung. In der Literatur wie zum Beispiel in [Van Den Bout & Miller 90] werden alle Neuronen mit dem Wert  $1/2 \pm \epsilon$  initialisiert, wobei  $\epsilon$  eine Zufallsvariable darstellt. Da im nicht-trivialen Fall mehr als zwei Szenenkanten vorhanden sind, ist die Spaltensumme in dem ersten Iterationsschritt weit größer als 1. Da die Wichtung der Spaltensumme relativ groß ist, dienen die ersten Iterationsschritte also in erster Linie dazu, alle Neuronen auf einen gleich niedrigen Wert zu bringen. Erst danach erfolgt eine Entwicklung, die Knoten- und Knotenpaarvergleiche berücksichtigt.

Daher wird die Konvergenzgeschwindigkeit hier wie folgt erhöht. Im Falle des Benchmarks sind beispielsweise 63 Szenenkanten vorhanden. Da die anfängliche Spaltensumme daher  $63 \cdot 1/2 = 31.5$  beträgt, werden die Neuronen in den ersten Iterationsschritten auf Werte von etwa  $1/63 = 0.016$  reduziert. Um die ersten Schritte der Iteration zu sparen, werden alle Neuronen daher gleich mit dem Wert  $1/\max\{M, N\}$  initialisiert. Die Neuronen sind also alle gleich und nicht leicht unterschiedlich initialisiert.

Die Schwierigkeit bei der Bestimmung einer Schrittweite besteht darin, daß größere Schrittweiten zwar eine schnellere Konvergenz bewirken, jedoch lokale Minima entlang der Verbesserungsrichtung übersprungen werden können. Die Schrittweite wurde hier mit  $\Delta = 0.1$  so groß gewählt, daß entlang der Kante des Hyperinheitswürfels als Suchraum 10 Schritte durchgeführt werden können. Bei noch größeren Schrittweiten wurde die Kostenfunktion im Falle der Testszenen wieder erhöht.

#### Problemspezifische Parameter

Das Optimierungsproblem ist unter anderem in der Energiefunktion kodiert. Da die Energiefunktion aus gewichteten Teiltermen besteht, müssen die vier Gewichtungparameter als problemspezifische Parameter bestimmt werden. Die vier optimalen

Gewichtungsparameter für eine Klasse von Probleminstanzen zu finden, stellt wieder ein eigenes Optimierungsproblem dar. Hier soll keine optimale, sondern lediglich eine günstige Parameterkombination gefunden werden. Dazu wurde folgende Strategie gewählt. Die Wichtigkeit der Zwangsbedingungen zwischen Knoten, die durch die Teilkosten  $K_K$  zum Ausdruck kommen, wird als etwa gleich groß betrachtet wie die Wichtigkeit der Zwangsbedingungen zwischen Knotenpaaren, die durch die Teilkosten  $K_P$  zum Ausdruck kommen. Die entsprechenden Gewichtungparameter wurden daher zunächst auf  $C_K = C_P = 1.0$  gesetzt. Da die Bedingung an die Zeilen- und Spaltensumme eine imperative Zwangsbedingung darstellt, muß die Gewichtung der Teilkosten  $K_{SZ}$  vielfach größer sein als die der admissiven Zwangsbedingungen. Zunächst wurde  $C_{SZ} = 5$  gesetzt. Die Gewichtung der Teilkosten  $K_{01}$  kann klein gewählt werden, da sich erst im späteren Verlauf der Iteration eine Konvergenz einer Korrespondenzvariable gegen 0 oder 1 erfolgen soll. Die entsprechende Gewichtung wurde daher mit  $C_{01} = 1/5$  vielfach kleiner als  $C_K$  und  $C_P$  gewählt. Danach wurden die einzelnen Parameter durch Versuch-und-Irrtum variiert [Weidner 94], so daß sich für die getesteten Probleminstanzen plausible Resultate ergeben haben. Experimentell wurden die folgenden Parameter ermittelt:

$$\begin{aligned} C_K &= 1.0 \\ C_P &= 0.9 \\ C_{SZ} &= 6.4 \\ C_{01} &= 0.1 \end{aligned}$$

Ebenfalls problemspezifische Parameter sind die drei Parameter  $a, b$  und  $c$  der Hilfsfunktion  $\mu_{a,b,c}$ , die zur Beschreibung numerischer Ähnlichkeitsfunktionen benutzt werden. Als maximal tolerierte Verkürzung  $a_L$  bzw. Verlängerung  $b_L$  einer Szenen- gegenüber einer Objektkante wurden im Falle der Testszenen Werte von  $a_L = 75mm$  bzw.  $b_L = 56mm$  zugelassen. Da Kanten umso stärker verkürzt sein können, je schräger die Aufsicht ist, in der sie betrachtet werden (siehe Kapitel 3), kann eine extrahierte Szenenkante bezogen auf die entsprechende Objektkante eine Verkürzung bzw. Verlängerung aufweisen, die diese Schwellwerte immer übersteigt. In diesem Fall können Zuordnungen nicht aufgrund der Ähnlichkeit zwischen einzelnen Knoten, sondern nur zwischen Knotenpaaren erfolgen.

Da die Ähnlichkeiten direkt in die Gewichte eingehen und die Gewichte wiederum die Größe des Gradienten bestimmen, sind die Mindestähnlichkeiten  $c_L$  und  $c_W$  zum einen so groß zu wählen, daß der resultierende Gradient groß genug und damit die Konvergenz nicht zu langsam ist. Zum anderen soll aber auch die Unterscheidung zwischen maximaler Ähnlichkeit ( $=1$ ) und Mindestähnlichkeit ( $=c$ ), möglichst groß sein. Daher wurden die Parameter  $c_L$  bzw.  $c_W$  für die Funktion  $\mu_{a,b,c}$  zur Bewertung von Längen- und Winkelabweichungen auf 0.5 gesetzt. Da in den Testszenen wenige unterschiedliche Winkel zwischen Kanten auftreten, wurde die erlaubte Winkeltoleranz mit  $\pm 5.0^\circ$  relativ großzügig gewählt. Zusammenfassend erhält man damit folgende problemspezifischen Parameter:

$$\begin{aligned} a_L &= -75mm \\ b_L &= 56mm \\ c_L &= 0.5 \\ a_W &= -5.0^\circ \\ b_W &= 5.0^\circ \\ c_W &= 0.5 \end{aligned}$$

### Terminierungsbedingung und Interpretation

Ist die Iteration gestartet, stellt sich die Frage, wann sie terminiert werden soll. Die Terminierungsbedingung kann von den Werten der Neuronenmatrix, von der Energie und von der Schrittzahl abhängen. Eine mögliche Terminierungsbedingung bricht die Iteration ab, falls die Energie nach mehreren Iterationen nicht weiter abnimmt. In diesem Fall wäre ein Minimum erreicht.

Es hat sich allerdings experimentell herausgestellt, daß eine Interpretation der Neuronenmatrix durch Rundung bereits korrekte Resultate liefert, bevor ein Minimum erreicht ist. Die Iteration wird daher nicht in Abhängigkeit von der Energieänderung terminiert, sondern wenn die Zeilen- und Spaltensummenbedingung bzgl. der gerundeten Neuronenausgänge erfüllt ist.

Ein Problem besteht allerdings darin, daß Szenenkanten, die teilweise sichtbar sind, nicht aufgrund von Ähnlichkeiten zwischen einzelnen Knoten, sondern allein aufgrund von Ähnlichkeiten zwischen einzelnen Knotenpaaren zugeordnet werden können. Die Erhöhung des Korrespondenzwertes eines Objektknotens und eines Szenenknotens, der eine teilweise sichtbare Szenenkante repräsentiert, erfolgt daher langsamer. Damit auch teilweise sichtbare Szenenkanten zugeordnet werden können, muß daher eine Mindestanzahl  $t_{min}$  an Iterationen durchgeführt werden.

In dem Fall, daß nach der Mindestanzahl  $t_{min}$  an Iterationen die Zeilen- und Spaltensummenbedingung bezogen auf die gerundeten Korrespondenzwerte nicht erfüllt wird, soll die Iteration nach einer maximalen Anzahl  $t_{max}$  an Iterationen terminiert werden. Damit ergibt sich die folgende Terminierungsbedingung:

$$\mathcal{B}(\mathbf{X}, t) \equiv [E_{SZ}(\text{round}(\mathbf{X})) = 0 \wedge t \geq t_{min} \vee t \geq t_{max}]$$

Dabei bezeichnet  $\text{round}(\mathbf{X})$  die Rundung der einzelnen Matrixelemente  $x_{i,k}$ . Als Interpretation werden die kontinuierlichen Korrespondenzvariable  $x_{i,k}$  auf binäre Korrespondenzvariable  $b_{i,k}$  gerundet. Die Interpretation  $\mathcal{I}$  besteht also in der Rundung der Matrixelemente von  $\mathbf{X}$ :

$$[\mathbf{B} = \mathcal{I}(\mathbf{X})] \equiv [\forall i, k : b_{i,k} = \text{round}(x_{i,k})]$$

#### 5.3.2 Resultate für das BSB-Netz

Um das BSB-Netz mit den ermittelten Parametern zu testen, werden die Objektgraphen der Objektkantenstrukturen in Abb. 4.18 mit den Szenengraphen der Szenenkantenstrukturen aus Abb. 4.13 und Abb. 4.14 verglichen. Die folgende Tabelle zeigt die Vertex- und Kantenanzahl der Objektkantenstrukturen:

Objektkantenstruktur	Vertexanzahl	Kantenanzahl
<i>Stuhl-links-oben</i>	17	21
<i>Stuhl-vorne-oben</i>	16	19
<i>Stuhl-rechts-oben</i>	17	21
<i>Tisch-links-oben</i>	13	15
<i>Tisch-vorne-oben</i>	14	15
<i>Stufen-links-oben</i>	11	15
<i>Stufen-vorne-oben</i>	10	13
<i>Stufen-rechts-oben</i>	11	15

Vergleiche der entsprechenden Objekt- mit Szenengraphen wurden nach folgenden Fällen untersucht:

- Der Objektgraph repräsentiert die Ansicht eines Objektes, das in der Szene instanziiert ist. Die Objektinstanz ist in dieser Objektansicht abgebildet.
- Der Objektgraph repräsentiert die Ansicht eines Objektes, das in der Szene instanziiert ist. Die Objektinstanz ist in dieser Objektansicht *nicht* abgebildet.
- Der Objektgraph repräsentiert die Ansicht eines Objektes, das in der Szene *nicht* instanziiert ist.

Von besonderem Interesse ist dabei, welche Erkennungsergebnisse das BSB-Netz bei fehlenden Szenenkanten und falschen oder unmöglichen Linienverbindungen liefert. Da die Erkennungsergebnisse interpretiert werden sollen, wird im folgenden auch von Korrespondenzen zwischen *Kanten* gesprochen.

### Objektansicht in Szene vorhanden

Im ersten untersuchten Fall repräsentiert der zu erkennende Objektgraph die Ansicht eines Objektes, das in der Szene instanziiert ist. Die Objektansicht ist in dieser Ansicht abgebildet. Um diesen Fall zu untersuchen, wird der Objektgraph *Stuhl-rechts-oben* mit dem Szenengraphen *Szene1* verglichen (siehe Abb. 5.10). Zunächst werden die verwendeten Objekt- und Szenenkantenstrukturen näher erläutert. Linien in der Strichzeichnung der Szenen- bzw. Objektkantenstruktur werden kurz als Szenen- bzw. Objektlinien bezeichnet.

In der Objektkantenstruktur in dem unteren Teil von Abb. 5.10 existiert eine Kante (Kante 20) am linken unteren „Stuhlbein“, die in den repräsentierten Ansichten nur als teilweise sichtbar angenommen wird. Die Szenenkantenstruktur zeichnet sich dadurch aus, daß an der linken Oberseite der Stuhllehne eine kurze verdeckende Kante fehlt. Die entsprechende Linie müßte an einem Linienendpunkt eine Linienverbindung mit Szenenlinie 8 bilden. Durch das Fehlen der kurzen Kante ist fälschlicherweise die Linienverbindung der Szenenlinien 7, 11 und 8 entstanden. Da die Szenenlinie 8 gleiche Ausrichtung besitzt wie die fehlende verdeckende Linie, wird die Linienverbindung und damit der entsprechende Vertextyp der Szenenlinien 7 und 11 nicht verändert. Wäre die fehlende Linie vorhanden, würde Szenenlinie 8 mit dieser Linie eine Linienverbindung mit dem Index 13 bilden.

Nach Start und anschließender Terminierung der Iteration besitzt die Korrespondenzmatrix  $X$  aus kontinuierlichen Korrespondenzwerten die Belegung wie im oberen Teil von Abb. 5.10. Rundet man jedes Matrixelement der kontinuierlichen Korrespondenzmatrix, erhält man die binäre Korrespondenzmatrix. Die Binärmatrix kann visualisiert werden, indem man in einer Strichzeichnung der Objektkantenstruktur jede Linie fett zeichnet, in deren *Spalte* eine Eins auftritt, und ebenso in der Strichzeichnung der Szenenkantenstruktur jede Linie fett zeichnet, in deren *Zeile* eine Eins auftritt. Damit wird prägnant dargestellt, welche Linien überhaupt korrespondierende Linien besitzen. Allerdings wird nicht dargestellt, welche Objektlinie mit welcher Szenenlinie korrespondiert. Daher wird jede Linie in beiden Strichzeichnungen mit dem Kantenindex markiert und jedes Matrixelement  $b_{i,k} = 1$  der Binärmatrix wird textuell als Zuordnung  $i \mapsto k$  ausgegeben. Die beiden Strichzeichnungen und die Zuordnungen sind in dem unteren Teil von Abb. 5.10 dargestellt. Die Ergebnisse für diesen Fall können wie folgt zusammengefaßt werden.

Nahezu alle Objektkanten werden in korrekter Weise den entsprechenden Szenenkanten zugeordnet. Auch die als teilweise sichtbar erwartete Objektkante wird korrekt der teilweise sichtbaren Szenenkante 30 zugeordnet. Die Zuordnung erfolgt nur durch die Knotenpaarvergleiche, da nur dann Kanten mit nicht-identischen Vertextypen zu einer Reduzierung der Kostenfunktion beitragen können.

Da in den Spalten der Objektkanten 4 und 9 alle Werte unter 0.5 liegen, wurden diesen Objektkanten keine Szenenkanten zugeordnet. Dieses Ergebnis ist korrekt, da die der Objektkante 4 entsprechende Szenenkante fehlt, und die der Objektkante 9 entsprechende Szenenkante 8 in einem Endpunkt einen falschen Vertextyp besitzt. Da bei Knotenvergleichen die Übereinstimmung von Vertextypen eine Mußbedingung darstellt, findet keine Zuordnung statt.

Ein weiteres Beispiel zeigt Abb. 5.11. Der Objektgraph *Tisch-links-oben* wird dabei mit dem Szenengraph *Szene1* verglichen. Die Objektkantenstruktur besitzt eine Kante (Kante 15), die in der charakteristischen Ansicht als teilweise sichtbar angenommen wird. Da in der Szenenkantenstruktur jedoch mehrere Kanten gleichen Typs existieren, die in den Endpunkten die gleichen Vertextypen und auch Nachbarkanten mit entsprechenden Kantentypen besitzen, besteht bei der Zuordnung der Objektkante 15 eine Konkurrenz mehrerer Szenenkanten, wobei keine Korrespondenzvariable den Schwellwert von 0.5 überschreiten kann. Daher kann die Objektkante 15 keiner der beiden Szenenkanten zugeordnet werden.

### Objekt nicht in Szene vorhanden

Im zweiten untersuchten Fall repräsentiert der Objektgraph die Ansicht eines Objektes, das nicht in der Szene instanziiert ist. Um diesen Fall zu untersuchen, wurde der Objektgraph *Treppe-links-oben* mit dem Szenengraph *Szene1* verglichen. Das Ergebnis zeigt Abb. 5.12. In diesem Fall wird der größte gemeinsame Subgraph gefunden, der in genau einem Szenen- und Objektknoten besteht. Die entsprechenden Kanten weisen zufälligerweise ähnliche Attribute auf. Sowohl Objektkante 14 als auch Szenenkante 36 besitzen den gleichen Kantentyp „-“, gleiche Vertextypen 2 und 5, haben zwar verschiedene Kantenlängen, deren Längenunterschiede jedoch innerhalb der erlaubten Abweichung liegen. Definiert man eine Gütefunktion, die eine Mindestanzahl korrespondierender Kanten fordert, werden solche Erkennungsergebnisse als nicht-erkannt interpretiert.

### Objekt mit anderer Ansicht in Szene vorhanden

Im dritten untersuchten Fall repräsentiert der zu erkennende Objektgraph die Ansicht eines Objektes, das in der Szene zwar instanziiert ist, dessen Ansicht in der Szene aber nicht abgebildet ist. Um diesen Fall zu untersuchen, werden die Objektgraphen *Stuhl-links-oben* und *Stuhl-vorne-oben* mit dem Szenengraphen *Szene1* verglichen (siehe Abb. 5.13). Die Ergebnisse können wie folgt zusammengefaßt werden.

Auch in diesem Fall werden nahezu alle Kanten bis auf folgende Ausnahmen korrekt zugeordnet. Der Objektgraph *Stuhl-links-oben* repräsentiert zwar im Vergleich zu dem Objektgraphen *Stuhl-rechts-oben* eine andere Ansicht, jedoch bestehen zwischen beiden Objektkantenstrukturen aufgrund von Symmetrien entlang einer senkrecht zur „Stuhlsitzfläche“ stehenden Spiegelebene mehrere gleich attributierte Kanten. Da die wechselseitige Lage von Kanten durch Winkel kodiert ist, sind bestimmte Kanten um andere drehbar, die ihrerseits als Drehachse dienen. Bei gleichen Kantenwinkeln sind also mehrere geometrische Kantenkonfigurationen möglich. Die Geometrie eines Objektes wird also nicht notwendigerweise eindeutig repräsentiert und eine spiegelverkehrte Zuordnung von Kanten ist möglich. Unterschiede zwischen den Objektgraphen beider Ansichten bestehen aber dennoch in den Vertextypen, die durch die teilweise entgegengesetzte Ausrichtung verdeckender Kanten entstehen. Die entgegengesetzte Ausrichtung in verschiedenen Ansichten

kommt dadurch zustande, daß einmal die Region der angrenzenden Fläche beispielsweise links der verdeckenden Linie und im anderen Fall die Region der angrenzenden Fläche rechts der spiegelsymmetrischen verdeckenden Linie zu liegen kommt. Diese Eigenschaft bedeutet, daß auch bzgl. ihrer Geometrie symmetrische Objektansichten *eines* Objektmodells aufgrund der Kanten- und Vertextypen verschieden sind. Die ermittelten Korrespondenzen sind aus dem oberen Teil der Abb. 5.13 ersichtlich.

Da die Spalten- und Zeilensummenbedingung zwar Spalten- und Zeilensummen größer als 1 bestrafen, jedoch Werte unter 1 nicht garantieren, können die Objektkanten 14 und 15 auch gleichzeitig einer einzigen Szenenkante (Szenenkante 31) zugeordnet werden. Dies kommt dadurch zustande, daß beide Objektkanten genau identische Attribute und auch fast gleich attributierte Nachbarkanten besitzen. Die Interpretation der Korrespondenzmatrix sollte daher so ergänzt werden, daß alle Korrespondenzen  $i \mapsto k$  und  $j \mapsto k, i \neq j$  und alle Korrespondenzen  $i \mapsto k$  und  $i \mapsto l, k \neq l$  gelöscht werden. Eine weitere Möglichkeit besteht darin, den Gewichtungsparemeter  $C_{01}$  der Teilkosten  $K_{01}$  kurz vor der Terminierung zu erhöhen, um Korrespondenzwerte nahe bei 0 oder 1 zu erzwingen. Ein ähnlicher Fall liegt bei dem Vergleich des Objektgraphen *Stuhl-vorne-oben* mit dem Szenengraphen *Szene1* vor. In dem unteren Teil von Abb. 5.13 ist ersichtlich, daß Objektkante 3 und 7 auf eine einzige Szenenkante (Kante 7) abgebildet werden.

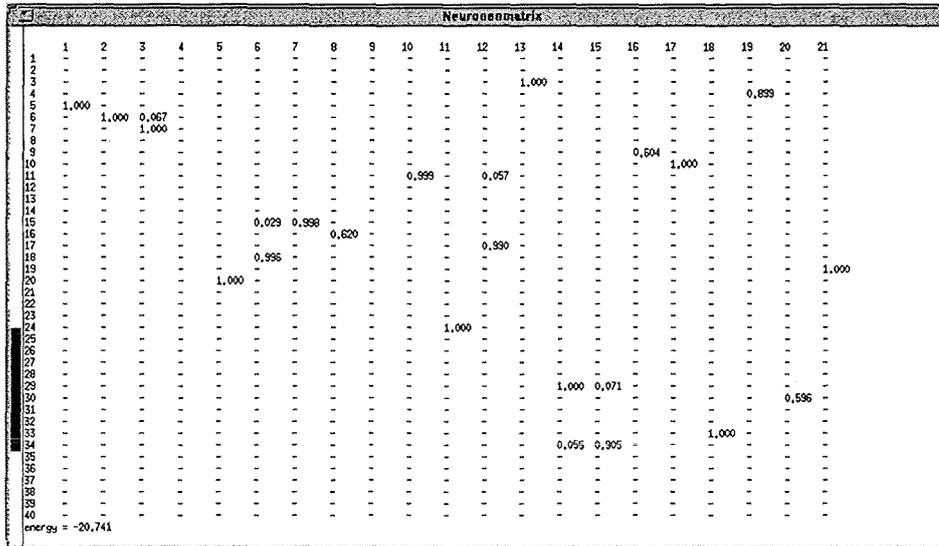
Aufgrund zugeordneter, fett gezeichneter Objekt- und Szenenlinien ist außerdem ersichtlich, daß der gefundene gemeinsame Subgraph nicht zusammenhängend ist. Dies kommt dadurch zustande, daß in der Kostenfunktion lediglich Vergleiche zwischen einzelnen Knoten oder einzelnen Knotenpaaren, jedoch nicht zwischen ganzen Knotengruppen eingehen. Um eine Zwangsbedingung für zusammenhängende Subgraphen in der Kostenfunktion einzubeziehen, müßte die Kostenfunktion von höherer als quadratischer Ordnung sein. Für solche Kostenfunktionen existieren aber nur aufwendigere Optimierungsverfahren, außerdem würde auch der Vernetzungsgrad bzw. Vernetzungsaufwand des neuronalen Netzes höher ausfallen. Wenn allerdings a priori bekannt ist, daß auch Verdeckungen auftreten können, die ein Objekt in mehrere Regionen aufteilen, dann kann die Eigenschaft, auch nicht-zusammenhängende Subgraphen erkennen zu können, gerade erwünscht sein.

### Unmögliche Linienverbindungen

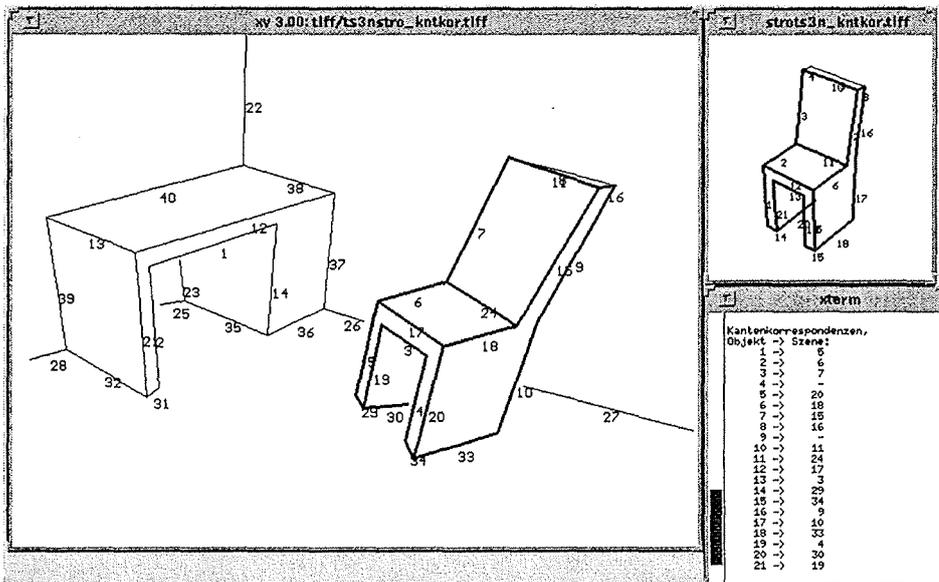
Der Linienverbindungsindex besitzt einen sehr hohen Informationsgehalt, da mehrere Linien, ihre Markierung und implizit der Blickpunkt eingehen. Da Knoten und Knotenpaare nur dann als ähnlich angesehen werden, wenn die Vertextypen übereinstimmen, müssen sie auch robust zu extrahieren sein. Mögliche Fehler sind fehlende Linien, falsche Orientierung verdeckender Linien oder verzerrte 2D-Winkel zwischen Linien.

Um zu sehen, welche Korrespondenzen resultieren, wenn in der Szene nicht erlaubte Linienverbindungen auftreten, wird der Objektgraph *Stuhl-rechts-oben* mit dem Szenengraphen *Szene2* verglichen. In der Strichzeichnung der Szene fehlt unter anderem an dem rechten oberen Eck der Stuhllehne eine konvexe Kante. Dadurch konnte Szenenkante 11 in der Phase der Ausrichtung verdeckender Kanten (Kapitel 4.2.3) nicht korrekt ausgerichtet werden.

Die resultierenden Korrespondenzen sind in Abb. 5.14 dargestellt. Objektkante 8 kann korrekterweise wegen fehlender Entsprechung in der Szene nicht zugeordnet werden. Objektkante 16 hingegen kann deshalb nicht zugeordnet werden, weil die entsprechende Szenenkante falsch ausgerichtet ist und als Folge davon an den Linienendpunkten unmögliche Linienverbindungen entstehen. Die restlichen Objektkanten werden korrekt zugeordnet.



Resultierende Korrespondenzmatrix: In der Matrix sind waagrecht die Indizes der Objektkanten und senkrecht die Indizes der Szenenkanten aufgetragen. Die jeweiligen Korrespondenzwerte liegen zwischen 0 und 1. Werte über 0.5 können als Korrespondenz interpretiert werden. Da in den Spalten der Objektkanten 4 und 9 alle Werte unter 0.5 liegen, wurden diesen Objektkanten keine Szenenkanten zugeordnet.



Korrespondenzen zwischen Szenengraph Szene1 und Objektgraph Stuhl-rechts-oben: Abgebildet sind die perspektivische Projektion der Szenenkanten (links) und der Objektkanten (rechts oben), die mit dem jeweiligen Kantenindex versehen sind. Zugeordnete Kanten sind fett gezeichnet, die Korrespondenzen selbst sind in einem eigenen Fenster abgebildet (rechts unten). Objektkanten 4 und 9 können keine Szenenkanten zugeordnet werden (siehe Text).

Abbildung 5.10: Korrespondenzmatrix (oben) und Visualisierung resultierender Korrespondenzen (unten)

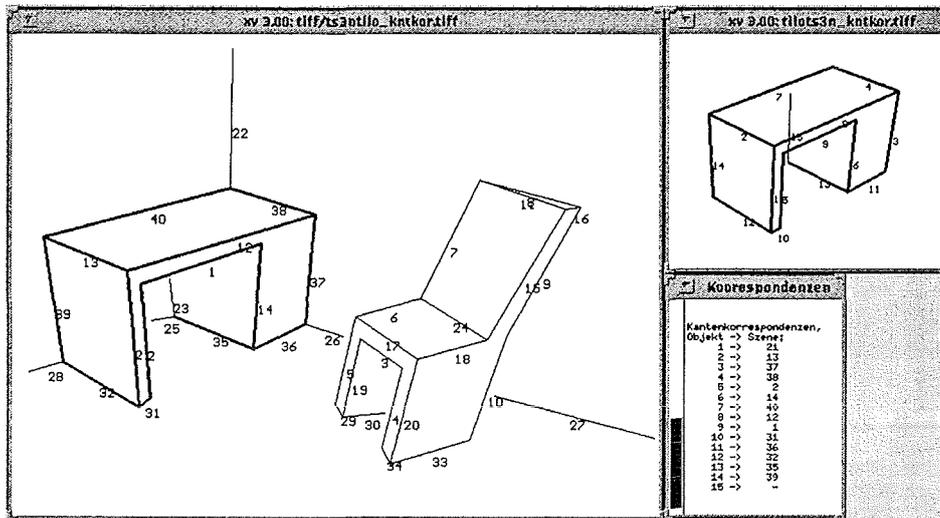


Abbildung 5.11: Korrespondenzen zwischen Szenengraph Szene1 und Objektgraph Tisch-links-oben: Die Objektansicht ist in der Szene vorhanden, jedoch kann die als teilweise sichtbar erwartete Objektkante 15 der Szenenkante 23 nicht zugeordnet werden (siehe Text).

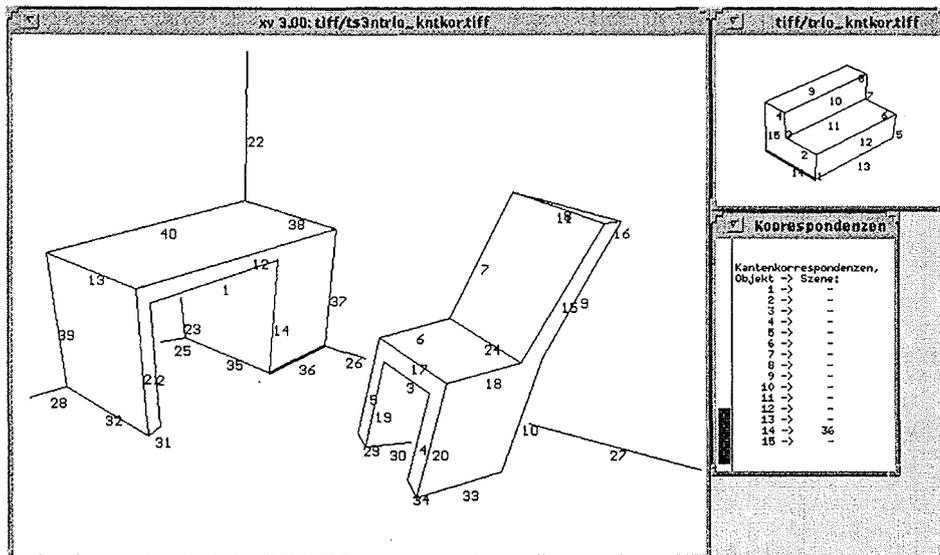
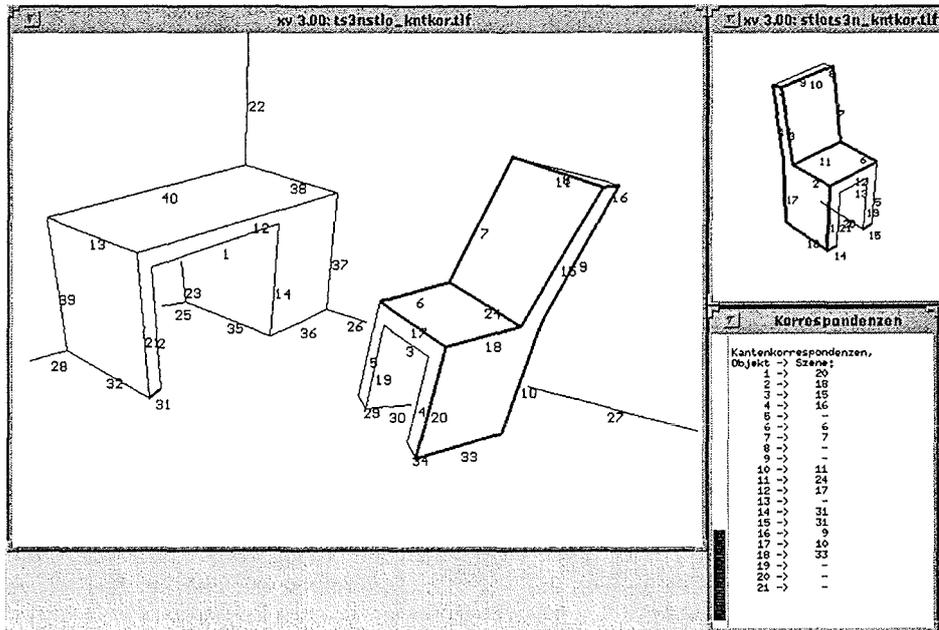
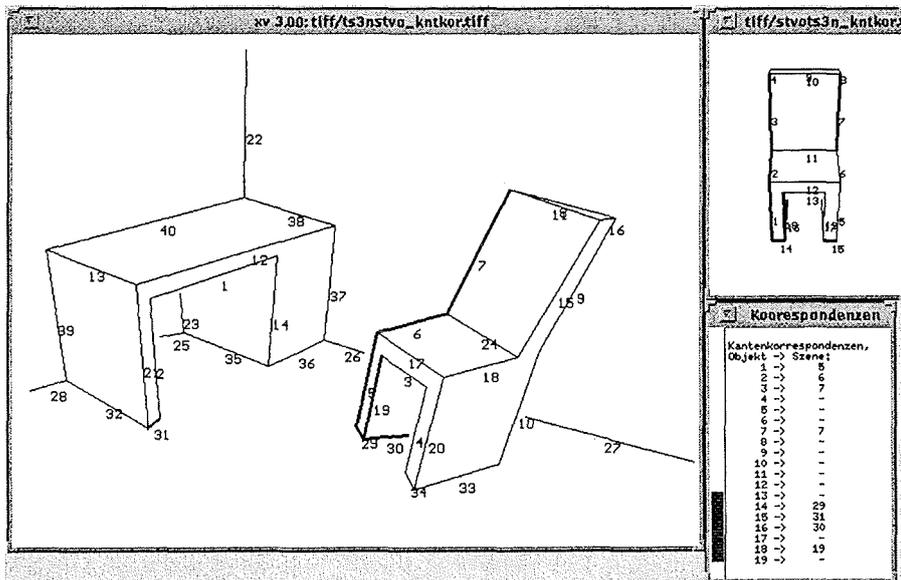


Abbildung 5.12: Korrespondenzen zwischen Szenengraph Szene1 und Objektgraph Treppe-links-oben: Die Szenenkante und die Objektkante der einzigen Korrespondenz weisen beide zufälligerweise ähnliche Attribute auf.



Korrespondenzen zwischen Szenengraph Szene1 und Objektgraph Stuhl-links-oben: Kanten werden spiegelverkehrt zugeordnet, der gefundene Subgraph ist nicht zusammenhängend. Objektkanten 14 und 15 werden beide der Szenenkante 31 zugeordnet.



Korrespondenzen zwischen Szenengraph Szene1 und Objektgraph Stuhl-vorne-oben: Der gefundene Subgraph ist aufgrund der Zuordnung von Objektkante 15 zu Szenenkante 31 nicht zusammenhängend.

Abbildung 5.13: Korrespondenzen zwischen Szenengraph Szene1 und Objektgraphen Stuhl-links-oben und Stuhl-vorne: das Objekt ist in der Szene zwar enthalten, jedoch nicht in der zu erkennenden Ansicht.

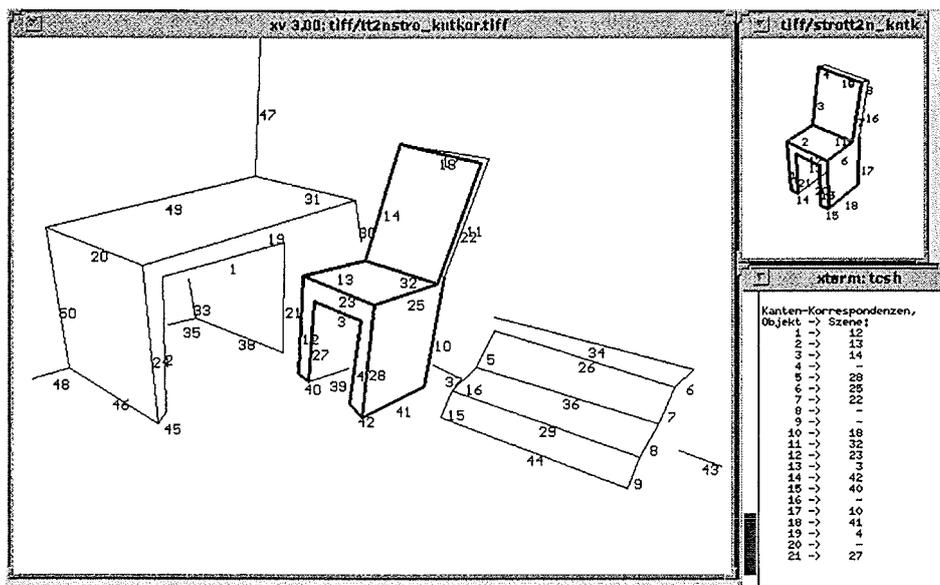


Abbildung 5.14: Korrespondenzen zwischen Szenengraph Szene2 und Objektgraph Stuhl-rechts-oben: Die verdeckende Szenenkante 11 ist falsch ausgerichtet (siehe Abb. 4.14). Dadurch kann auch die Nachbarkante infolge verfälschter Kantentypen nicht zugeordnet werden.

### 5.3.3 Diskussion der Ergebnisse

Die Ergebnisse können wie folgt zusammengefaßt werden:

- Der Ansatz setzt voraus, daß Objektansichten in der aufgenommenen Szene nicht mehrfach vorhanden sind. Die zu erkennenden Objektansichten müssen zur Orientierung geeignete Landmarken darstellen, im Falle mehrerer instanzierter Objektansichten wäre eine eindeutige Position nicht mehr bestimmbar.
- Obwohl Szenenkantenstrukturen zu einem bestimmten Grad verzerrt und einzelne Szenenkanten teilweise sichtbar sind, können Korrespondenzen korrekt ermittelt werden.
- Gefundene Subgraphen müssen nicht notwendigerweise zusammenhängend sein. Resultate nicht-zusammenhängender Subgraphen treten vor allem dann auf, wenn zu erkennende Objekte nicht oder nicht in der zu erkennenden Ansicht in der Szene instanziiert sind.

Wenn ein Objekt in einer anderen Ansicht instanziiert ist, läßt sich aber anhand der Anzahl korrespondierender Kanten und der kontinuierlichen Korrespondenzstärken entscheiden, welche Ansicht als erkannt interpretiert werden soll. Außerdem ist die Forderung nach zusammenhängenden Subgraphen zu einschränkend, wenn auch Verdeckungen auftreten können, die ein Objekt im Bild in mehrere Regionen aufteilen.

- Insbesondere wenn die zu erkennende charakteristische Ansicht zu der charakteristischen Ansicht, in der das Objekt abgebildet ist, symmetrisch ist, können mehrdeutige Zuordnungen entstehen. Dann sollte folgende Forderung erfüllt sein: Sind die zu erkennende und abgebildete Ansicht identisch, sollen mit hoher Wahrscheinlichkeit mehr oder zumindest gleich viele Korrespondenzen ermittelt werden, als wenn sie symmetrisch sind. Dies ist aber gewährleistet, da zueinander symmetrische verdeckende Kanten entgegengesetzte Ausrichtung und folglich unterschiedliche Vertextypen an ihren Endpunkten besitzen.

Das heißt, in einem weiteren Verarbeitungsschritt werden die Korrespondenzergebnisse aller Ansichten eines einzigen Modells miteinander verglichen. Dabei wird nur das beste Resultat als mögliche Interpretation vorgesehen, da vorausgesetzt wird, daß jedes Objekt und daher auch nur eine Ansicht dieses Objektes maximal einmal instanziiert ist.

- Der Ansatz um Verfahren ergänzt werden, die in einem zweiten zeitlich weniger kritischen Schritt bestimmte Konsistenzbedingungen berücksichtigen und das Erkennungsergebnis verfeinern.
  - Hinzufügen von Korrespondenzen, die aufgrund von Konkurrenzbeziehungen zwischen ähnlichen Szenenkanten um die Zuordnung zu einer Objektkante nicht ermittelt werden konnten.
  - Löschen von Korrespondenzen, bei denen aufgrund exakt identischer Attributierung mehreren Objektkanten eine Szenenkante zugeordnet wird, und die auch nicht durch Verschärfung von Parametern des Optimierungsproblems unterdrückt werden können.
  - Hinzufügen und Löschen von Korrespondenzen, um einen zusammenhängenden gemeinsamen Subgraphen herzustellen, falls a-priori bekannt ist, daß ein Objekt im Bild nicht in mehreren Regionen abgebildet sein kann.

Im Gegensatz zu Baumsuchverfahren für Interpretationsbäume, kann die globale Konsistenz der ermittelten Korrespondenzen im Sinne von Abs. 1.2.3 nicht garantiert werden. Es ist jedoch die Auffassung des Autors, daß durch Verwendung genügend vieler, einschränkender geometrischer Zwangsbedingungen wie beispielsweise Vertexattribute im Durchschnitt konsistente Lösungen gefunden werden.

### 5.3.4 Vergleich BSB-MFA und Diskussion

Zur Parameterbestimmung und zum Vergleich des Mean Field Annealing Algorithmus mit dem BSB-Netz dient ein Benchmark. Da der Szenengraph des Benchmarks mehr Kanten als die realen Testszene aufweisen soll und die Konstruktion von Testszene nur für diesen Fall zu aufwendig wäre, wurde ein synthetisches Abstandsbild einer trihedral polyedrischen Szene errechnet, mit dem in den Kapiteln 3 und 4 beschriebenen Verfahren eine Szenenkantenstruktur generiert und daraus im Anschluß ein Szenengraph erzeugt (Abb. 5.15). Der Benchmark besteht aus dem Szenengraphen und mehreren Objektgraphen, die in der Szene instanziierte Objektansichten repräsentieren. Verdeckende Kantenpunkte wurden mittels des Verfahrens der Kantenpunktlokalisierung aus Kapitel 3 um den Betrag von nur einem Bildpunkt verschoben. Im Gegensatz zu den realen Testszene beträgt die minimale Länge von Kanten der Szene nicht 10 cm sondern 5 cm. Die Objektkantenstrukturen wurden wiederum manuell konstruiert. Die Anzahl der Vertizes und Kanten zeigt folgende Tabelle:

Objektkantenstruktur	Vertexanzahl	Kantenanzahl
<i>Tür-links</i>	8	8
<i>Tisch-rechts-oben</i>	15	15
<i>Stuhl-links-oben</i>	16	19
<i>Treppe-links-oben</i>	12	14

Alle problemspezifischen Parameter des BSB-Netzes sowie Initialisierung und Interpretation der iterativen Verbesserung können übernommen werden, da sich nicht das Problem, sondern das Optimierungsverfahren ändert. Neben der Schrittweite

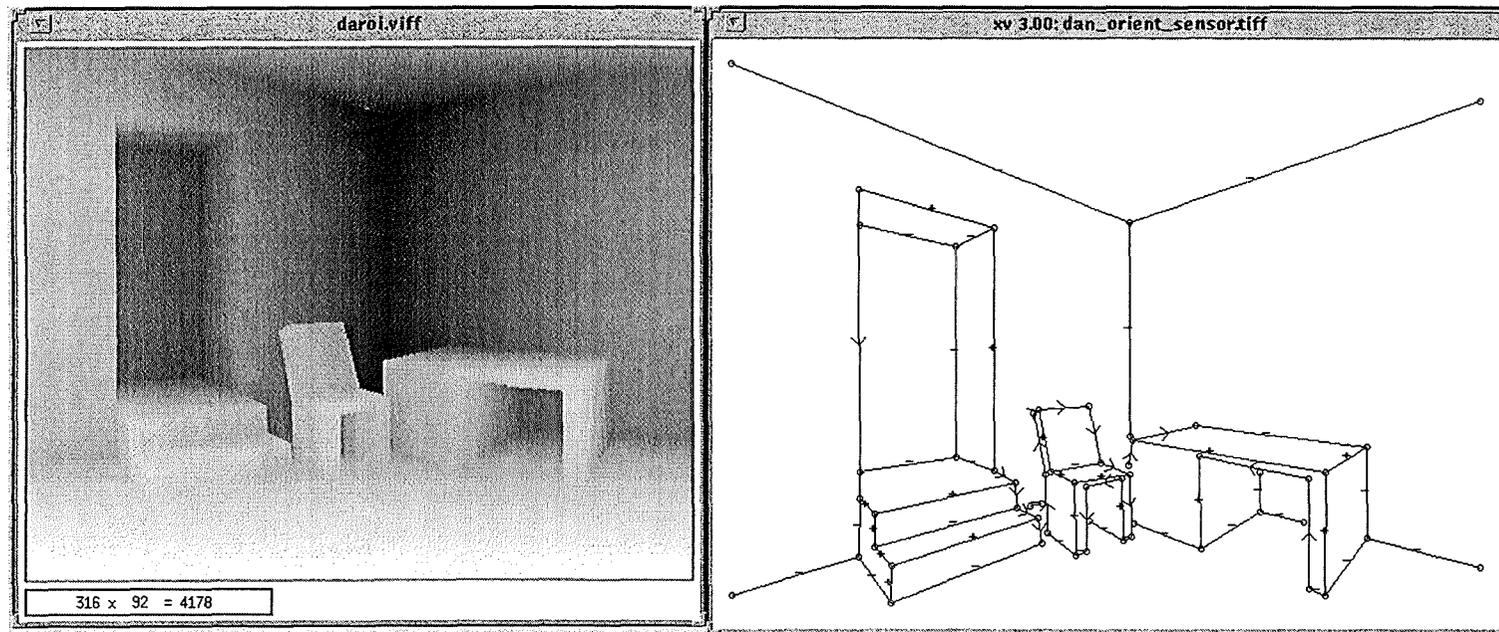


Abbildung 5.15: Synthetisches Abstandsbild (links) und markierte Strichzeichnung der Szene des Benchmarks (rechts): Das synthetische Abstandsbild besitzt eine Auflösung von  $500 \times 400$  Punkten, mit einem maximalen Abstand von 5 Meter innerhalb eines Blickfeldes von  $70^\circ \times 70^\circ$  Grad. In dem Abstandsbild ist der Ausschnitt eines Raumes abgebildet, in dessen Raumecke sich eine Tür, eine Treppe, ein Stuhl und ein Tisch befinden. Die Szene ist nur aufgrund einer hohen Prägnanz gewählt, reale Zimmer oder Räume sind weitaus komplexer.

muß also zusätzlich zu diesen Parametern ein geeigneter Abkühlplan festgelegt werden, der die Temperaturabnahme beschreibt. Beide Parameter wurden in [Weidner 94] experimentell ermittelt.

Die Schrittweite wurde zunächst kleiner als die des BSB-Netzes gewählt, um genügend Verbesserungsschritte innerhalb des Suchraums des hochdimensionalen Einheitswürfels durchführen zu können. Zunächst wurde  $\Delta_{MFA} = 0.05$  gesetzt.

Um einen Abkühlplan festzulegen, sind die Anfangstemperatur  $T_a$ , die Temperaturabnahme  $T(t)$  und die Anzahl der Iterationen  $n(T)$  auf jeder Temperaturstufe festzulegen. Die beiden Parameter und die Funktion für eine Klasse von Probleminstanzen zu finden, stellt ein eigenes Optimierungsproblem dar. Hier soll keine optimale, sondern lediglich eine günstige Kombination gefunden werden. Dazu wurde die Strategie gewählt, daß zuerst eine günstige Anfangstemperatur ermittelt wird. Danach wird eine günstige Iterationsanzahl der ersten Temperaturstufe ermittelt und für alle weiteren Temperaturstufen festgehalten. Dann erst wird eine geeignete Temperaturabnahme eingestellt.

Die Schwierigkeit bei der Bestimmung der Anfangstemperatur besteht darin, daß zu hohe Temperaturen eine im Vergleich zum Rechenaufwand nur geringe Verbesserung bringen, während zu tiefe Temperaturen das System zu schnell abkühlen, so daß falsche Zuordnungen erzielt werden. Gesucht ist daher die Temperatur, bei der sich jeder Spin signifikant gegen 0 oder 1 zu bewegen beginnt. Diese Temperatur wird deshalb auch als kritische Temperatur  $T_c$  bezeichnet. In [Van Den Bout & Miller 90] wird für ein ebenfalls quadratisches kombinatorisches Optimierungsproblem eine analytische Schätzung der kritischen Temperatur vorgeschlagen. Die Schätzung ist aber aufgrund einer Eigenwertberechnung sehr aufwendig.

Hier wurde eine praktikable Vorgehensweise gewählt. Eine günstige Anfangstemperatur wird experimentell dadurch ermittelt, daß bei voreingestellter Schrittweite und fester Anzahl  $n(T_a)$  an Iterationen mehrere Durchläufe bei verschiedenen Temperaturen von  $T_a = 1.0$  fallend bis  $T_a = 0.1$  durchgeführt wurden. Für jede Anfangstemperatur wurde nach  $n(T_a)$  Iterationen visuell entschieden, ob sich die Spinwerte signifikant von ihrem Anfangszustand verändern. Die höchste Temperatur, bei der im Falle des Benchmarks signifikante Änderungen zu beobachten waren, war  $T_a = 0.4$ .

Für die Anfangstemperatur  $T_a = 0.4$  wurde der Vorgang für verschiedene  $n = 1 \dots 10$  durchgeführt. Signifikante Änderungen waren im Falle des Benchmarks erst ab 4 Iterationen zu beobachten. Die Anzahl von Konfigurationsänderungen pro Temperaturstufe wurde auf konstant  $n(T) = 4$  gesetzt.

Für die Temperaturabnahme wurde keine logarithmische Funktion, sondern aus Zeitgründen die Abkühlfunktion  $T(t) = T_a/q^t$  verwendet, die durch den Parameter  $q$  gekennzeichnet ist. Drei Abkühlfunktionen wurden für wenig unterschiedliche Werte  $q \in \{1.06, 1.09, 1.12\}$  getestet. Die Graphen der drei Abkühlfunktionen sind in Abb. 5.17 dargestellt. In der folgenden Tabelle ist die Anzahl der Iterationen für unterschiedliche  $q$ -Werte mit  $\Delta = 0.05, T_a = 0.4, n(T) = 4$  aufgelistet. Die Zahlen in Klammern geben an, wie viele Iterationen für die korrekte Zuordnung von teilweise sichtbaren Kanten benötigt wurde. Bei den mit \* gekennzeichneten Iterationen war ein Springen der Werte von Korrespondenzvariablen zwischen Werten weit über 0.5 und weit unter 0.5 zu beobachten, was auf ein zu schnelles Abkühlen hindeutet.

	$q = 1.06$	$q = 1.09$	$q = 1.12$
Treppe-links-oben	32	28	28
Stuhl-links-oben	32(40)	32(40)	28(*)
Tisch-rechts-oben	36(44)	32(*)	32(*)
Tür-links	40(48)	40(48)	40(48)

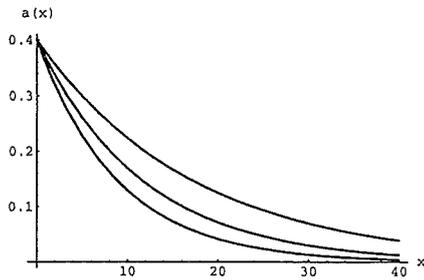


Abbildung 5.16: Verschiedene Abkühlfunktionen für verschiedene  $q$ -Werte

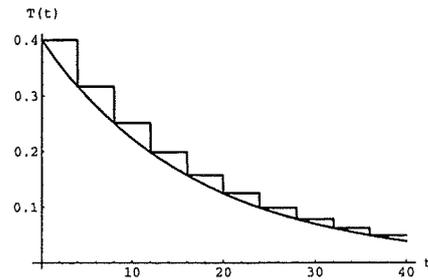


Abbildung 5.17: Temperaturabnahme einer Abkühlfunktion

Abkühlfunktionen mit verschiedenen Werten für  $q$  sind in Abb. 5.16 ersichtlich. Das schnellere Abkühlen mit  $q = 1.09$  und  $q = 1.12$  bringt zwar eine leichte Konvergenzbeschleunigung, allerdings werden nicht alle Korrespondenzen korrekt gefunden. Durch das mögliche Springen der Werte von Korrespondenzvariablen wird außerdem das Festlegen einer geeigneten Terminierungsbedingung erschwert.

Da nur durch Abkühlen mit dem Parameter  $q = 1.06$  die Knoten aller Objektgraphen zugeordnet werden können, wird  $q = 1.06$  gesetzt. Die obigen Parameter legen also den Abkühlplan  $(0.4, 0.4/1.06^t, 4)$  fest, der eine stufenweise Temperaturabnahme wie in Abb. 5.17 beschreibt. Im Vergleich dazu sind die notwendigen Iterationen des BSB-Netzes für  $\Delta = 0.05$  und für  $\Delta = 0.1$  in folgender Tabelle gegenübergestellt:

	BSB $\Delta = 0.1$	BSB $\Delta = 0.05$	MFA $\Delta = 0.05$
Treppe-links-oben	10(-)	19(-)	32(-)
Stuhl-links-oben	12(16)	22(32)	32(40)
Tisch-rechts-oben	14(17)	26(35)	36(44)
Tür-links	11(11)	21(21)	40 (48)

Aus der Tabelle ist ersichtlich, daß das BSB-Netz bei gleicher Schrittweite etwa um den Faktor 1.6 schneller, d.h. mit weniger Iterationsschritten zu gleichen Korrespondenzen kommt. Die Schrittweite des MFA-Algorithmus kann nicht erhöht werden, ohne die Ergebnisse zu verschlechtern. Im Gegensatz dazu kommt das BSB-Netz durch Verdoppelung der Schrittweite auf nahezu doppelte Erkennungsgeschwindigkeit.

Die Ergebnisse des MFA-Algorithmus können für den Benchmark wie folgt zusammengefaßt werden. Das Mean Field Annealing erfordert im Vergleich zu dem Brain-State-in-a-Box Netz die experimentelle Bestimmung von drei zusätzlichen Parametern, um den Abkühlplan zu spezifizieren. Die Parameter sind außerdem sensibel, da eine geringfügige Abänderung des  $q$ -Faktors große Auswirkungen auf die gefundenen Korrespondenzen hat. Weiterhin muß die Schrittweite  $\Delta$  im Vergleich zu dem BSB-Netz halbiert werden. Das Mean Field Annealing war dadurch um Faktor 1.6 langsamer als das BSB-Netz mit halbiertes Schrittweite.

Die Ergebnisse lassen folgende Schlußfolgerungen zu. Die Kantenattribute und ihre Repräsentation in Szenen- und Objektgraphen bilden Zwangsbedingungen, die

nur wenige Minima der Kostenfunktion erzeugen. Um eine korrekte Lösung zu finden, reicht das einfache BSB-Netz bereits aus. Die Eigenschaft des Mean Field Annealing, aus lokalen Minima entkommen zu können, zahlt sich vor allem bei solchen kombinatorischen Optimierungsproblemen aus, die sehr viele, nahezu gleich gute, suboptimale Lösungen besitzen, wobei es darauf ankommt, lediglich eine von vielen ähnlich guten Lösungen zu finden.

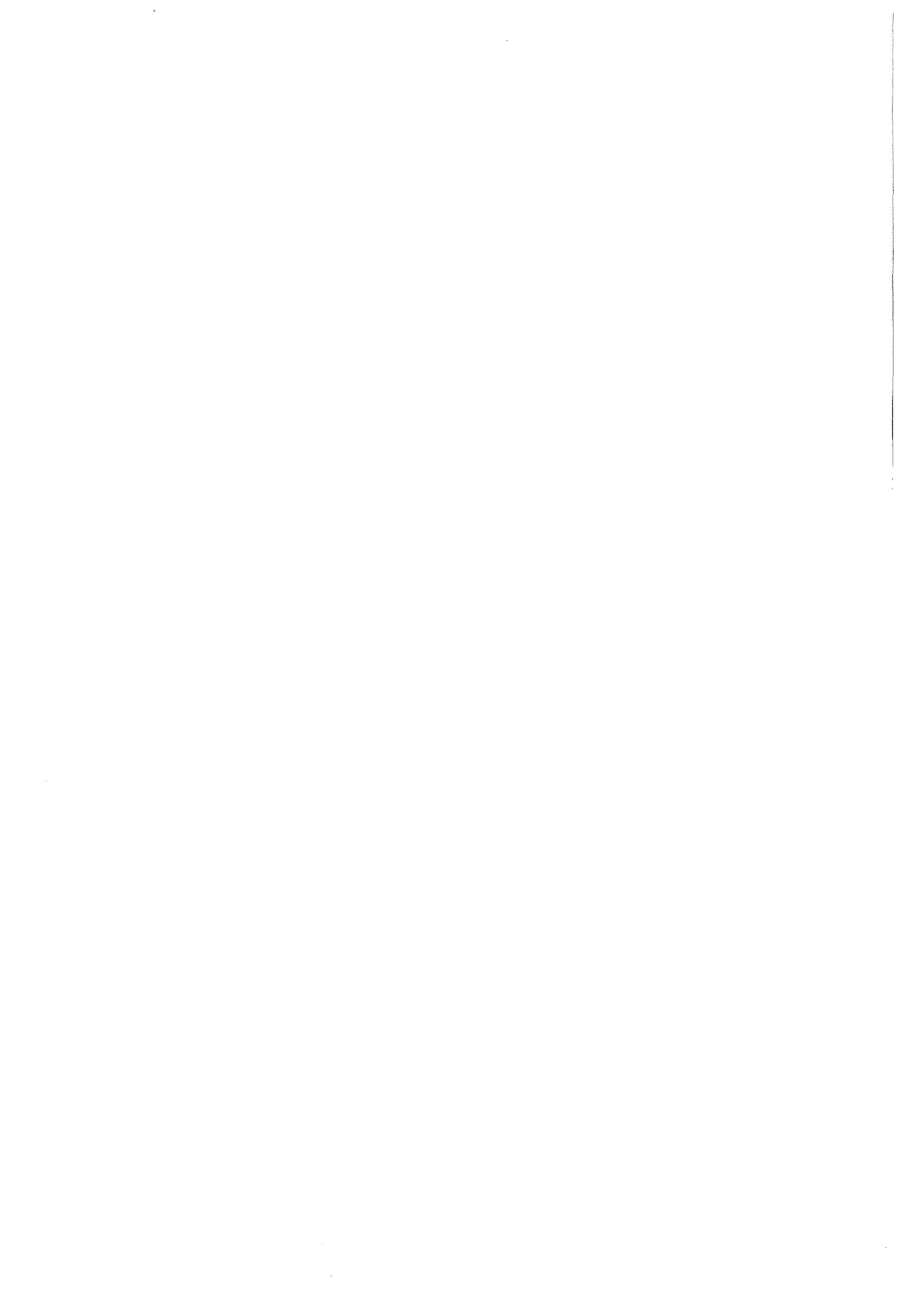
## 5.4 Zusammenfassung

Zur Bewertung optimaler Korrespondenzen zwischen Objekt- und Szenenkanten wurde eine Kostenfunktion systematisch konstruiert, so daß eine Ergänzung um weitere Merkmale einfach möglich ist. Die Kostenfunktion läßt Korrespondenzen teilweise sichtbarer Kanten zu und berücksichtigt bis zu einem bestimmten Grad verzerrte Szenenkantenstrukturen.

Als gut parallelisierbare Optimierungsverfahren im Sinne des Abschnitts 1.2.4 wurden drei Ansätze in einem einheitlichen Rahmen gegenübergestellt. Als einheitlicher Rahmen dient das Schema der iterativen Verbesserung. Für das Brain-state-in-a-box Netz, den Mean-Field Algorithmus und eine Anpassung der nicht-linearen Relaxationsmarkierung wurden anhand des Schemas der iterativen Verbesserung und anhand des behandelten Objekterkennungsproblems verdeutlicht, worin sich die Verfahren unterscheiden, und daß teilweise lediglich verschiedene Begriffe verwendet werden, die aber Gemeinsamkeiten darstellen. Die nicht-lineare Relaxationsmarkierung mußte dazu so angepaßt werden, daß sie erst dadurch auf das vorliegende Problem anwendbar ist.

Experimente wurden für verschiedene Spezialfälle durchgeführt. Die Erkennungsergebnisse wurden auf der Grundlage des verwendeten Erkennungsverfahrens diskutiert. Der Ansatz eignet sich vor allem dann, wenn sehr schnell erste Initialkorrespondenzen erforderlich sind, die in einem zweiten zeitlich weniger kritischen Schritt im Hinblick auf globale Konsistenzkriterien verfeinert werden können.

Das Mean Field Annealing erfordert im Vergleich zu dem Brain-State-in-a-Box Netz die experimentelle Bestimmung zusätzlicher und auch sensibler Parameter. Das Mean Field Annealing war um Faktor 1.6 langsamer als das BSB-Netz mit halbiertes Schrittweite. Größere Schrittweiten waren für das Mean Field Annealing nicht möglich.



## Kapitel 6

# Zusammenfassung und Ausblick

In dem hier vorgestellten Ansatz zur Objekterkennung aus Abstandsbildern werden Abstandsbilder von Laserradarscannern verwendet, da sie Szenen gegenüber den oft eingesetzten Triangulationslaserscannern vollständiger aufnehmen können. Da die Objekterkennung im Kontext der Navigation bestimmten Echtzeitanforderungen genügen muß, sollten Verfahren eingesetzt werden, die eine hohe Erkennungsgeschwindigkeit ermöglichen. Diese Anforderung hatte zwei Auswirkungen. Als Bildmerkmale von Polyedern wurden Kanten gewählt, da Verfahren, die Polyederflächen als Bildmerkmale heranziehen, einen um ein Vielfaches höheren Zeit- und Speicheraufwand besitzen. Zum zweiten wurden gut parallelisierbare Verfahren zur Erkennung von Objekten verwendet, da sie das Potential bieten, die Erkennungsgeschwindigkeit durch Einsatz von Parallelrechnern zu erhöhen.

Die vorliegende Arbeit leistet Beiträge bezogen auf Bildbereichshinweise, auf Szenenbereichshinweise und auf die Objekterkennung.

- Die Extraktion von verdeckenden Kanten als Bildbereichshinweise wird trotz eines für das Radarmeßverfahrens typischen Meßfehlers ermöglicht. Die Berechnung von Mittleren Krümmungen wurde dazu so angepaßt, daß der Krümmungsvorzeichenwechsel zur Unterscheidung von verdeckenden und verdeckten Kantenpunkten und zur Lokalisierung verdeckender Kantenpunkte herangezogen werden kann. Die angepaßte Mittlere Krümmung ermöglicht es dadurch einer nachgelagerten Lokalisierungsphase, Bildpunkte auf verdeckenden Kanten von solchen zu unterscheiden, die durch den Mixed-Point Effekt verfälscht sind.
- Die hier vorgestellte Kantenpunktlokalisierung verfeinert unter Berücksichtigung des vorhandenen Szenenwissens insbesondere verdeckende Kantenpunkte so, daß sie mit hoher Wahrscheinlichkeit nicht durch den Mixed-Point Effekt verfälscht sind und sie im Gegensatz zu anderen Verfahren auch zu Bildpunktketten zusammengefaßt werden können. Dem Autor sind keine Arbeiten bekannt, die verdeckende Sprungkanten in Abstandsbildern von Laserradarscannern so detektieren und lokalisieren, daß eine Kantenpunktverkettung möglich wäre.
- In diesem Ansatz werden Linienverbindungen nicht zur Markierung von Strichzeichnungen, sondern als Merkmale von Vertizes in einer bestimmten Ansicht

verwendet. Da Linienverbindungen von der Markierung aller inzidenten Linien abhängig sind, stellen sie Merkmale mit hohem Informationsgehalt dar. Die einzige dem Autor bekannte Repräsentation, die ebenfalls Linienverbindungen als Merkmale verwendet, ist der Kanten-Linienverbindungsgraph von [Godin & Levine 89]. Allerdings werden hier auch alle Linienverbindungen berücksichtigt, die durch die flächige Berührung von Polyedern entstehen können.

- Da Linienverbindungen aufgrund der ungleichförmigen Abtastung des Laser-radarscanners nicht direkt im Abstandsbild ermittelt werden können, wird neben dem Abstandsbild ein zweiter Bildbereich verwendet. Durch die Einbindung der perspektivischen Projektion extrahierter Szenenkanten als zweiten Bildbereich wird es verschiedensten Laserradarscannern mit unterschiedlichen Arten ungleichförmiger Abtastung ermöglicht, Linienverbindungen korrekt zu ermitteln.
- Zur Bewertung optimaler Korrespondenzen zwischen Objekt- und Szenenkanten wurde eine Kostenfunktion systematisch konstruiert, so daß sie auch um die Verwendung weiterer beispielsweise flächenbasierter Merkmale einfach erweitert werden kann. Die Kostenfunktion läßt Korrespondenzen teilweise sichtbarer Kanten zu und berücksichtigt bis zu einem bestimmten Grad verzerrte Szenenkantenstrukturen.
- Als gut parallelisierbare Optimierungsverfahren im Sinne des Abschnitts 1.2.4 wurden drei Ansätze in einem einheitlichen Rahmen gegenübergestellt. Für das Brain-state-in-a-box Netz, den Mean-Field Algorithmus und eine Anpassung der nicht-linearen Relaxationsmarkierung wurden Unterschiede der Verfahren und unterschiedliche Bezeichnungen, die aber Gemeinsamkeiten darstellen, transparent gemacht. Die nicht-lineare Relaxationsmarkierung wurde dazu so angepaßt, daß sie erst dadurch auf das vorliegende Problem anwendbar ist.
- Aufgrund der experimentellen Ergebnisse (siehe Abschnitt 5.3.5) eignet sich der hier vorgestellte Ansatz vor allem dann, wenn (1) in aufgenommenen Szenen zu erkennende Objektansichten nicht mehrfach in der Szene vorhanden sind, wenn (2) auch Objektansichten erkannt werden sollen, die beispielsweise durch Verdeckung in getrennten Bildbereichen erscheinen und wenn (3) sehr schnell erste Initialkorrespondenzen erforderlich sind. Die Grenzen des Verfahrens wurden ebenfalls anhand der experimentellen Ergebnisse aufgezeigt.
- Der MFA-Algorithmus wurde mit dem BSB-Netz anhand eines Benchmarks in Bezug auf die Erkennungsgeschwindigkeit verglichen. Das Mean Field Annealing erfordert im Vergleich zu dem Brain-State-in-a-Box Netz die experimentelle Bestimmung von zusätzlichen und sensiblen Parametern. Das Mean Field Annealing war etwa um den Faktor 1.6 langsamer als das BSB-Netz mit halbiertes Schrittweite. Größere Schrittweiten und damit höhere Erkennungsgeschwindigkeiten waren für das Mean Field Annealing nicht möglich.
- Die Ergebnisse lassen folgende Schlußfolgerungen zu. Die Kantenattribute und ihre Repräsentation in Szenen- und Objektgraphen bilden Zwangsbedingungen, die nur wenige Minima der Kostenfunktion erzeugen. Um eine korrekte Lösung zu finden, reicht das einfache BSB-Netz bereits aus. Die Eigenschaft des Mean Field Annealing, aus lokalen Minima entkommen zu können, zählt sich vor allem bei solchen kombinatorischen Optimierungsproblemen aus, die sehr viele, nahezu gleich gute, suboptimale Lösungen besitzen, wobei es darauf ankommt, lediglich eine von vielen ähnlich guten Lösungen zu finden.

Da diese Arbeit besonderes Augenmerk auf bestimmte Teilprobleme legt, konnten nicht alle interessanten Problemstellungen mit der gleichen Intensität bearbeitet werden. Dadurch ergeben sich Ansatzpunkte zu Verbesserungen und Erweiterungen.

Wünschenswert wäre auch die Erkennung nicht nur trihedraler, sondern allgemeiner Polyeder. Dies erfordert Vertixattribute, die mehr als drei inzidente Kanten einbeziehen. Dabei sollen die Vertextypen nur als charakteristisches Merkmal von Vertizes dienen und müssen nicht unbedingt zur Markierung von Strichzeichnungen geeignet sein.

Weiterhin kann die Genauigkeit der Kantenapproximation, der Kantengruppierung und der Bestimmung der Vertextypen erhöht werden. Kanten können genauer approximiert werden, wenn Kanten durch eine lineare Regression an die zu approximierenden Abtastpunkte angepaßt werden. Kanten können außerdem genauer gruppiert werden, indem die Gruppierung nicht nur Kantenpaare, sondern alle Kanten in einem Vertex berücksichtigt, so daß durch topologische Zwangsbedingungen eine höhere Genauigkeit erzielt werden kann. Die Vertextypen und damit die Linienverbindungen können eventuell robuster ermittelt werden, indem sie immer durch Winkel zwischen Kanten ermittelt werden. Mit dem auf Seite 77 angegebenen Verfahren können zu einer Fläche gehörende Kanten aggregiert werden, wodurch die Extraktion flächenbasierter Merkmale ermöglicht wird. Flächenbasierte Merkmale können dazu verwendet werden, Mehrdeutigkeiten weiter zu reduzieren oder zu eliminieren.

Eine Ergänzung des Verfahrens wäre wünschenswert, in der ein Erkennungsergebnis unter Berücksichtigung bestimmter Konsistenzbedingungen verfeinert wird. Vorschläge zur Verfeinerung von Korrespondenzen sind in Abschnitt 5.3.3 aufgeführt. Ein weiterer Verarbeitungsschritt ist erforderlich, der aufgrund der für jede Objektansicht gefundenen Korrespondenzen entscheidet, welches Objektmodell als erkannt gelten soll. Dazu können Verfahren eingesetzt werden, in denen eine Gütefunktion einen bestimmten Schwellwert übersteigen muß. Die Gütefunktion könnte eine Mindestanzahl korrespondierender Kanten, die Längen- und Winkelabweichungen korrespondierender Kanten, die Anzahl korrespondierender teilweise sichtbarer Kanten und die Stärken der Korrespondenzvariable berücksichtigen.

Ein weitere sehr interessante Fragestellung besteht darin, wie die betrachteten Verfahren parallelisiert werden können. Dazu sollte allerdings die Architektur des Parallelrechners bekannt sein, auf dem die Erkennung effizient ablaufen soll. Aufgrund der Vielfalt von Rechnerarchitekturen würde die Behandlung dieser Frage den Rahmen sprengen.

Schließlich bleibt anzumerken, daß die Technologie bildgebender Laserscanner dauernd weiterentwickelt wird. Mit abnehmendem Rauschanteil wird die Kantenextraktion im Gegensatz zu regionenbasierten Verfahren in Zukunft an Bedeutung gewinnen. Richtungsweisend sind die Integration von laserbasierten Abstandssensoren in VLSI-Schaltungen [Kanade *et al.* 90] oder Kameras für Abstandsbildfolgen in Videobildrate [Yamamoto *et al.* 93]. Neue Scannersysteme, die nicht in Kugelkoordinaten, sondern in kartesischen Koordinaten abtasten [Elektronik 92], vermeiden außerdem Bildverzerrungen aufgrund ungleichförmiger Abtastung.

## Anhang A

# Herleitung der Linienverbindungen nach Sugihara

Die Aufzählung der möglichen Linienverbindungen wird systematisch in der folgenden Art und Weise durchgeführt.

Aufgrund der Annahme trihedral polyedrischer Szenen nach Bedingung 2.1 besitzt ein isolierter Vertex im dreidimensionalen Raum genau drei Kanten und jede Kante ist entweder konvex oder konkav. Vertizes können daher in vier Klassen unterteilt werden. (1) Vertizes mit drei konvexen Kanten, (2) Vertizes mit zwei konvexen und einer konkaven Kante, (3) Vertizes mit zwei konkaven und einer konvexen Kante und (4) Vertizes mit drei konkaven Kanten (siehe Abb. A.1).

Legt man durch die drei Flächen eines Vertex drei Ebenen, unterteilen diese den umgebenden Raum in acht Oktanten. Einige der Oktanten werden von Material ausgefüllt, die anderen sind leer. Die Anzahl der gefüllten Oktanten ist dabei 1, 3, 5, 7, wenn die Zahl der konvexen Kanten jeweils 3, 2, 1 oder 0 ist. Wenn man den Blickpunkt in einen der Oktanten plaziert, erhält man aufgrund der perspektivischen Projektion eine Ansicht des Vertex und zugleich eine Linienverbindung. Dabei ändert sich die Klasse einer Linienverbindung nicht, wenn sich der Blickpunkt in demselben Oktanten bewegt. Über alle Blickpunkte in freien Oktanten in Kombination mit den vier isolierten Vertizes kann man eine vollständige Liste aller möglichen markierten Linienverbindungen aufstellen.

Aufgrund der Annahme über die Kontaktarten von Polyedern nach Bedingung 2.3 sind zulässige Kontaktarten zweier Körper (siehe Abb. A.2): (a) Fläche mit Fläche, (b) Punkt mit Fläche, (c) Kante mit Fläche und (d) Kantenpunkt mit Kantenpunkt. An jedem Vertex wie zum Beispiel  $q$  oder  $r$ , oder an jedem Pseudovertex wie zum Beispiel  $p$ ,  $s$  oder  $t$ , teilen die Ebenen, die in den umgebenden Flächen liegen, den Raum in eine endliche Anzahl von Zellen. Jede Zelle ist wiederum entweder mit Material gefüllt oder leer. Wiederum kann man über alle Anordnungen von Blickpunkten in freien Oktanten eine komplette Liste aller möglichen markierten Linienverbindungen aufstellen. Der Vertex  $q$  und der Pseudo-Vertex  $t$  erzeugen Linienverbindungen, die in einfachere zerlegt werden können, und sind daher in der unten aufgeführten Liste nicht enthalten.

Mit der Annahme über Szenen und den drei Konventionen in der Aufzählung zeigt der obere Teil in Abb. A.3 die Liste aller möglichen Linienverbindungsklas-

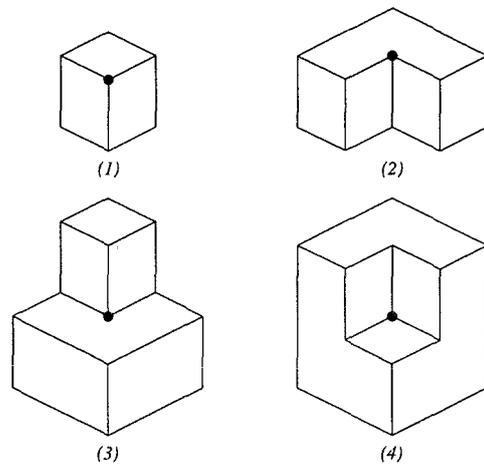


Abbildung A.1: Klassifizierung von isolierten Vertizes

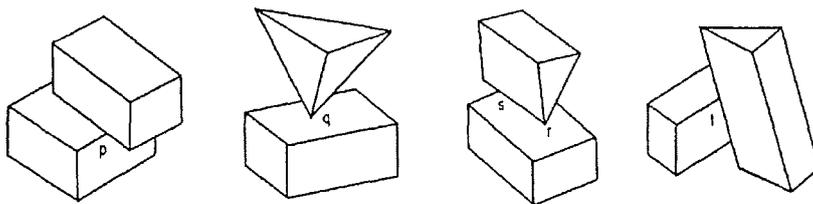
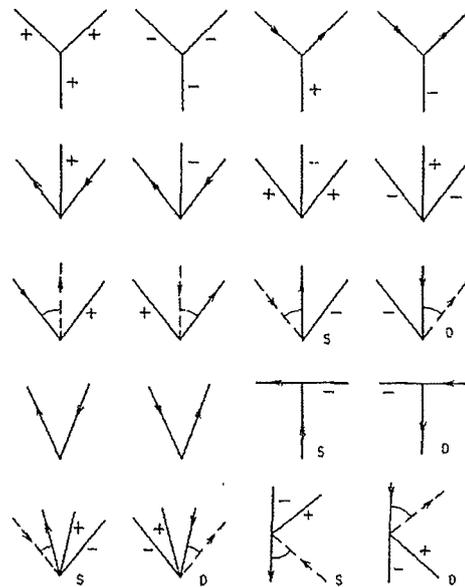
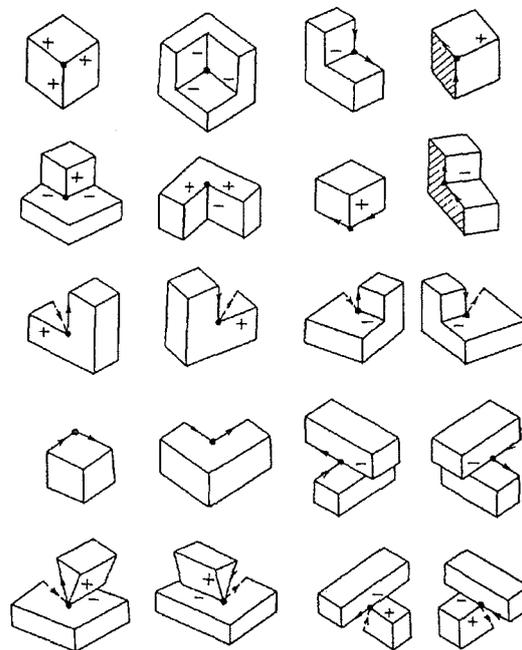


Abbildung A.2: Zulässige Kontaktarten zweier Polyeder (v.l.n.r.): Fläche mit Fläche, Punkt mit Fläche, Kante mit Fläche und Kantenpunkt mit Kantenpunkt (siehe Text).

sen. In dem unteren Teil der Abbildung A.3 sind Beispielszenen für die jeweiligen Linienverbindungen aufgeführt. In der Liste der Linienverbindungen ist eine Beziehung zwischen verdeckten und verdeckenden Linien durch einen verbindenden Bogen gekennzeichnet. Alle Y-, L- und pfeilförmigen Linienverbindungen stammen von isolierten Polyedern. Die restlichen T-förmigen Linienverbindungen und solche mit vier Linien stammen von solchen Punkten, in denen sich zwei Polyeder berühren.



Linienverbindungen von Sugihara: aufgrund verdeckter Linien (unterbrochen gezeichnet) besitzen manche Linienverbindungen auch vier Linien. Verdeckende und zugehörige verdeckte Linien sind über einen Bogen verbunden.



Polyedrische Beispielszenen für die 20 Linienverbindungen von Sugihara

Abbildung A.3: Linienverbindungen nach Sugihara und polyedrische Beispielszenen

## Anhang B

# Schätzen partieller Ableitungen digitaler Flächen

Die ersten und zweiten partiellen Ableitungen in einem Punkt des Abstandsbildes werden nach dem Verfahren von [Beaudet 78], [Besl & Jain 85a] geschätzt [Vogelgesang 91b]. Die Technik beruht auf der Anpassung einer Polynomfläche zweiten Grades mit diskreten orthogonalen Basispolynomen an eine lokale Umgebung innerhalb eines  $n \times n$ -Fensters mit ungerader Fenstergröße  $n$ .

Fensterkoordinaten seien mit  $(u, v) \in U \times U$  mit  $U = \{-(n-1)/2, \dots, -1, 0, 1, \dots, (n-1)/2\}$  gegeben. Zur Berechnung der zweiten Ableitungen müssen die polynomiellen Basisfunktionen mindestens vom Grad zwei sein.

$$\begin{aligned}\phi_0(u) &= 1 \\ \phi_1(u) &= u \\ \phi_2(u) &= u^2 - m(m+1)/3 \text{ mit: } m = (n-1)/2\end{aligned}$$

Die Polynombasis ist orthogonal, das heißt:  $\sum_{u \in U} \phi_i(u)\phi_j(u) = \delta_{ij}$ . Die Basisfunktionen  $\phi_i$  werden durch die für gegebene Fenstergröße  $n$  einmal zu berechnenden Faktoren  $P_i$  normiert:

$$\begin{aligned}P_0 &= n \\ P_1 &= \frac{2}{3}m^3 + m^2 + \frac{1}{3}m \\ P_2 &= \frac{8}{45}m^5 + \frac{4}{9}m^4 + \frac{2}{9}m^3 - \frac{1}{9}m^2 - \frac{1}{15}m\end{aligned}$$

Gesucht ist die Funktion  $\hat{r}$ , die die digitale Funktion  $r$  für gegebene Polynombasis möglichst gut annähert:

$$\hat{r}(u, v) = \sum_{i=0 \wedge i+j \leq 2}^2 a_{ij} \phi_i(u) \phi_j(v)$$

Die Lösung für die einzelnen Polynomkoeffizienten erhält man dann mit:

$$a_{ij} = \sum_{(u,v) \in U^2} r(u, v) b_i(u) b_j(v)$$

Das Polynom hat dann die folgende Gestalt:

$$r(u, v) = a_{00} + a_{01}v + a_{10}u + a_{11}uv + a_{02}v^2 + a_{20}u^2$$

Die ersten Ableitungen lassen sich analytisch ausrechnen mit:

$$\begin{aligned} r_u &= a_{10} + a_{11}v + 2a_{20}u \\ r_v &= a_{01} + a_{11}u + 2a_{02}v \end{aligned}$$

In [Besl & Jain 86] werden die Ableitungen allerdings abgeschätzt mit  $r_u = a_{10}$  und  $r_v = a_{01}$ . Die zweiten partiellen Ableitungen ergeben sich zu:

$$\begin{aligned} r_{uu} &= 2a_{20} \\ r_{vv} &= 2a_{02} \\ r_{uv} &= r_{vu} = a_{11} \end{aligned}$$

Jede Ableitungsberechnung ist eine Faltung der Bildfunktion mit einer Faltungsmaske  $D_{ij}$ :

$$\begin{aligned} a_{ij} &= \sum_{(u,v) \in U^2} r(u, v) b_i(u) b_j(v) \\ &= \sum_{u=-m}^m \sum_{v=-m}^m r(u, v) D_{ij}(u, v) \\ &= D_{ij} \otimes r(u, v) \end{aligned}$$

Für jede erste und zweite partielle Ableitung berechnet sich analog die entsprechende Faltungsmaske:

$$\begin{aligned} D_{01} &= D_u = \mathbf{d}_0 \mathbf{d}_1^T \\ D_{10} &= D_v = \mathbf{d}_1 \mathbf{d}_0^T \\ D_{02} &= D_{uu} = \mathbf{d}_0 \mathbf{d}_2^T \\ D_{20} &= D_{vv} = \mathbf{d}_2 \mathbf{d}_0^T \\ D_{11} &= D_{uv} = \mathbf{d}_1 \mathbf{d}_1^T \end{aligned}$$

wobei sich jeder Vektor  $\mathbf{d}_i$  aus dem Vektor der Basisfunktionen an den Abtastpunkten zusammensetzt:

$$\mathbf{d}_i = (b_j(-m), \dots, b_j(m))^T$$

wobei:

$$\begin{aligned} \mathbf{d}_0 &= \frac{1}{P_0} \underbrace{[1, \dots, 1]}_n^T \\ \mathbf{d}_1 &= \frac{1}{P_1} [-m, -(m-1), \dots, 0, \dots, (m-1), m]^T \\ \mathbf{d}_2 &= \frac{1}{P_2} [m^2 - m(m+1)/3, (m-1)^2 - m(m+1)/3, \dots, -m(m+1)/3, \dots, \\ &\quad (m-1)^2 - m(m+1)/3, m^2 - m(m+1)/3]^T \end{aligned}$$

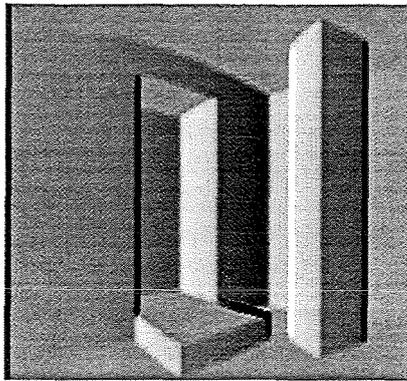
Für  $n = 7, m = 3$  ergeben sich folgende Vektoren:

$$\begin{aligned} \mathbf{d}_0 &= \frac{1}{7} [1, 1, 1, 1, 1, 1, 1]^T \\ \mathbf{d}_1 &= \frac{1}{28} [-3, -2, -1, 0, 1, 2, 3]^T \\ \mathbf{d}_2 &= \frac{1}{84} [5, 0, -3, -4, -3, 0, 5]^T \end{aligned}$$

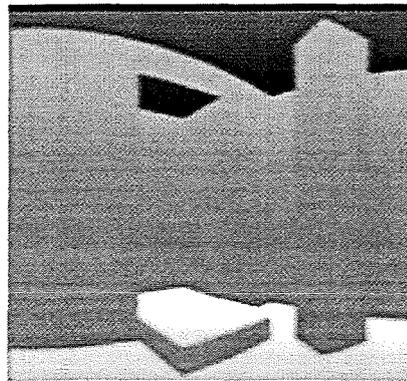
Mit den fünf  $n \times n$  Matrizen als Faltungsmasken läßt sich jede erste und zweite partielle Ableitung durch Faltung der Bildfunktion mit der entsprechenden Faltungsmaske berechnen. Die Ableitungen erhält man dann mit:

$$\begin{aligned}r_u &= D_u \otimes r(u, v) \\r_v &= D_v \otimes r(u, v) \\r_{uu} &= D_{uu} \otimes r(u, v) \\r_{vv} &= D_{vv} \otimes r(u, v) \\r_{uv} &= D_{uv} \otimes r(u, v)\end{aligned}$$

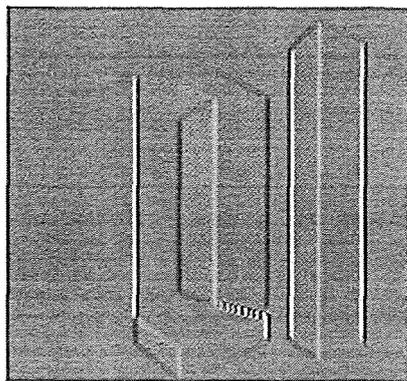
In Abb. B.1 sind die fünf Gradientenbilder des synthetischen Abstandsbildes der Testszene abgebildet.



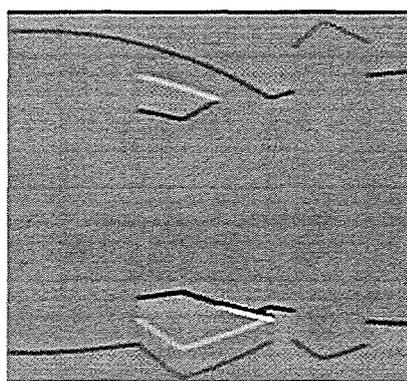
$T_u$



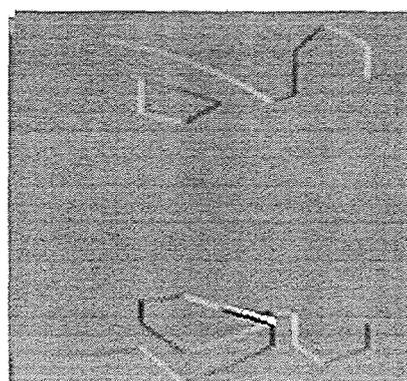
$T_v$



$T_{uu}$

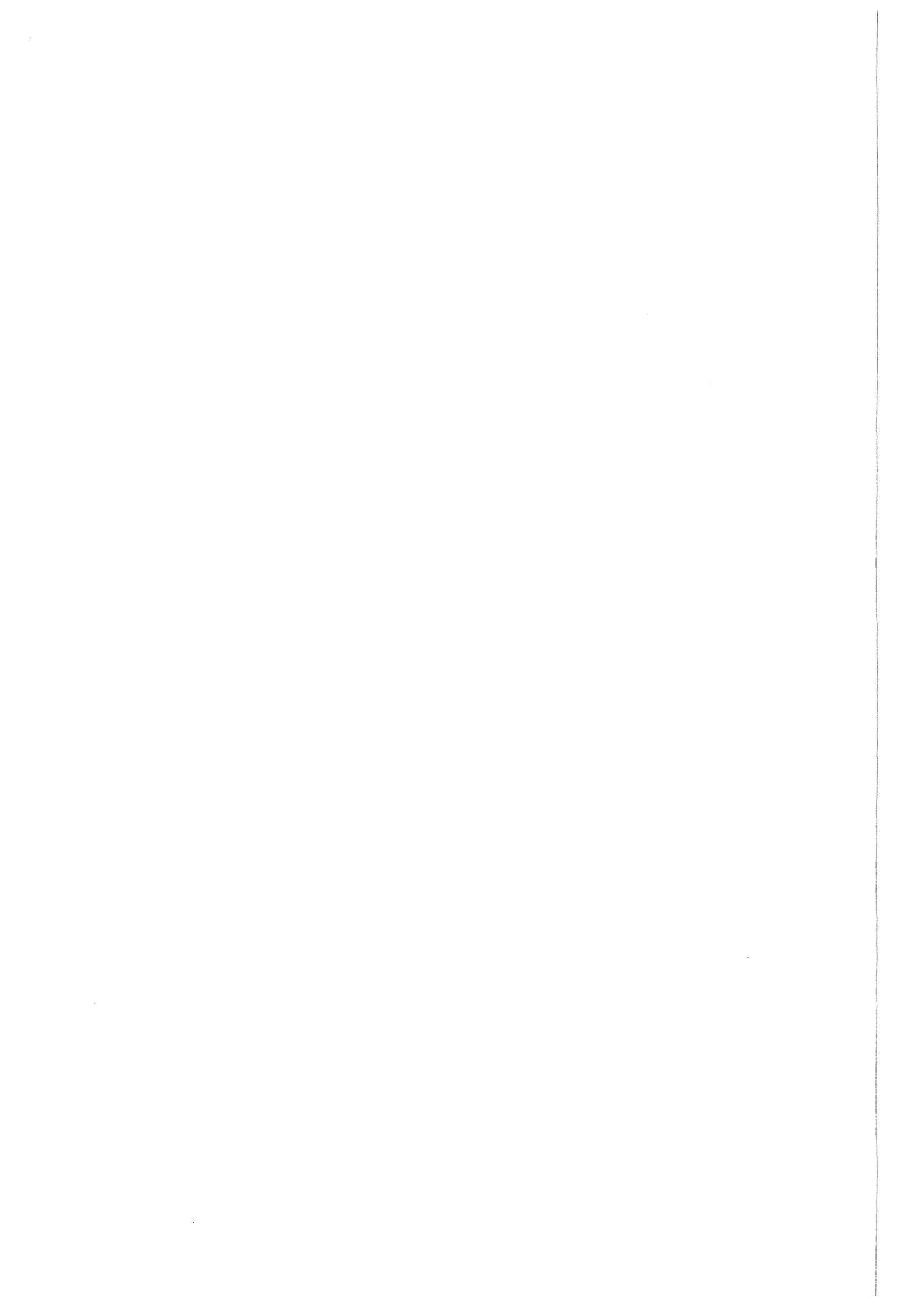


$T_{vv}$



$T_{uv}$

Abbildung B.1: Partielle Ableitungen des synthetischen Abstandsbildes der Testszene in Grauwertkodierung (weiß=positiv, schwarz=negativ, grau= $\pm 0$ )



# Anhang C

## Werkzeuge

Alle Verfahren der Bildauswertung wurden auf einem PC-AT i486 unter den Betriebssystemen MS-DOS und OS/2 in der Sprache Pascal implementiert. Um die Erkennungsverfahren schneller testen zu können, wurden sie in der Sprache C auf einer Sun SparcStation 2 implementiert. Zur Validierung der Resultate wurden zusätzlich Visualisierungswerkzeuge auf Sun Unix-Workstations verwendet.

Der Verarbeitungsablauf untergliedert sich in vier Teile (siehe Abb. C.1). Zur Erzeugung von Abstandsbildern existieren zwei Möglichkeiten: Entweder nimmt ein 3D-Laserradar reale Abstandsbilder der Umgebung auf oder ein Sensorsimulator berechnet ideale, synthetische Abstandsbilder anhand geometrischer Szenenbeschreibungen. Der Sensorsimulator stellt dabei eine Implementierung des Sensormodells aus Abschnitt 2.3 dar. Ideale Bilder liegen als Datei in einem lesbaren Format vor, das zur weiteren Bearbeitung in das Binärformat der realen Abstandsbilder umgewandelt wird.

Von Prozeduren der ikonischen Bildauswertung werden aus einem Abstandsbild Bildbereichshinweise in Form von Bildpunktketten erzeugt. Die Prozeduren dienen zur Krümmungsberechnung, zur Verdünnung von Binärbildern, zur Anwendung von Nachbarschaftsoperatoren, zur Bildarithmetik und zur Bildpunktverkettung. Die Bildpunktketten werden durch Routinen der symbolischen Bildauswertung schrittweise zu einem Szenengraphen umgewandelt. Dabei lesen und schreiben die Routinen zur Kantenapproximation, Gruppierung, Ausrichtung und Vertexattributierung Textdateien eines festgelegten Formates.

Die Textdatei der Szenenkantenstruktur zusammen mit einer Textdatei einer der Objektkantenstrukturen bilden die Eingabe des Erkennungsprogramms. Die erkannten Korrespondenzen liegen als Zuordnungen zwischen Indizes von Szenen- und Objektkanten vor. Die Indexzuordnungen mit Korrespondenzstärken können dann von anderen nachgelagerten Routinen benutzt werden.

Zur Visualisierung und Validierung von Bildbereichshinweisen wurde das Bildverarbeitungssystem KHOROS [Khoros Group 93], zur Visualisierung und Validierung von Szenenbereichshinweisen vor allem das System AVS der Firma Advanced Visual Systems Inc. [AVS 92] verwendet. Um Bildpunkte aus Abstandsbildern und Quaderszenen mit AVS visualisieren zu können, wurden Konvertierungsroutinen implementiert, die lesbare Textdateien im AVS-Format erzeugen. Zusammenfassend existieren die folgenden Schnittstellen in Form von Dateien: Quaderszenenbeschreibungen für LASI, Abstandsbilder von LASI und von 3D-Ladar, Textdateien für Objektkantenstrukturen und für verschiedene Stadien der Szenenkantenstrukturu-

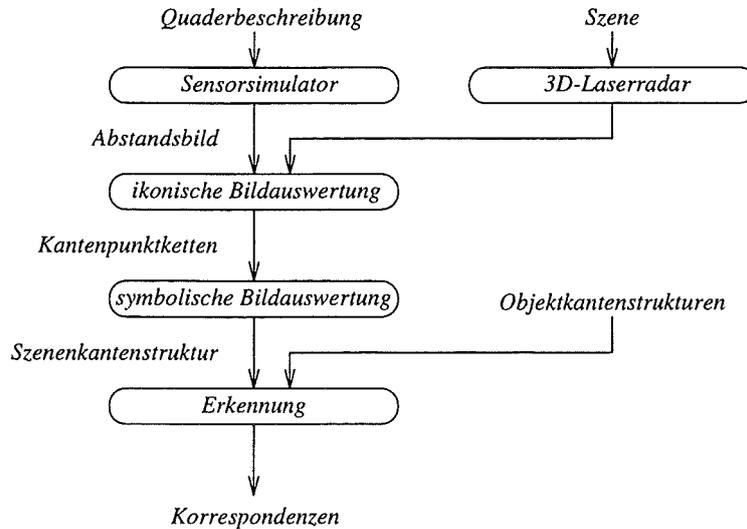


Abbildung C.1: Systemarchitektur und Schnittstellen: an den Pfeilen sind die Daten vermerkt, die zwischen den Modulen fließen.

ren, Korrespondenzen als Textdatei pro Objekt- und Szenekantenstruktur, Khoros-Bildformat, AVS-Dateiformate.

### Sensorsimulator LASI

Um den Ansatz zunächst für ideale Abstandsbilder validieren zu können, wurde der Sensorsimulator LASI [Cramer 91] implementiert (siehe Abb. C.2). Synthetische Abstandsbilder wurden verwendet, um zunächst die Korrektheit der Kantenextraktion für den idealen Fall zu überprüfen. Vorteilhaft ist außerdem, daß gezielt solche Unsicherheiten untersucht werden können, die nur infolge von Bildauswerteverfahren erzeugt werden.

Die Eingabedatei des Sensorsimulators ist in lesbarem Format. Sie besteht aus den sechs Sensorparametern, die das Blickfeld in Bezug auf das Sensorkoordinatensystem festlegen: die Scanwinkel  $\phi_a, \theta_a$ , die Winkelinkremente  $\Delta\phi, \Delta\theta$  und die Anzahl der horizontalen und vertikalen Abtastungen  $\phi_{max}, \theta_{max}$ . Weiterhin sind die Quader durch einen Index und die entsprechende Geometrie in Form der Breite, Höhe und Tiefe spezifiziert. Die Instanz eines Quaders liegt in Form des Quaderindex, der Position in kartesischen Koordinaten in Bezug auf das Umweltkoordinatensystem und die Orientierung in Form dreier Eulerwinkel vor. Ebenso sind Position und Orientierung des Sensors bezogen auf das Sensorkoordinatensystem angegeben.

Der Simulator liest die Datei ein, zeichnet die Szene als perspektivische oder orthographische Projektion der Quader und visualisiert schrittweise jeden Abtastvorgang in der Quaderszene durch Einzeichnen des Abtastpunktes. Dadurch wird insbesondere erleichtert, trotz der ungleichförmigen Abtastung ein Blickfeld festzulegen, das einen gewünschten Szenenausschnitt enthält. Die Ausgabedatei ist ebenfalls lesbar und enthält wiederum die Sensorparameter und pro Abstandswert eine Fließkommazahl in Metern.

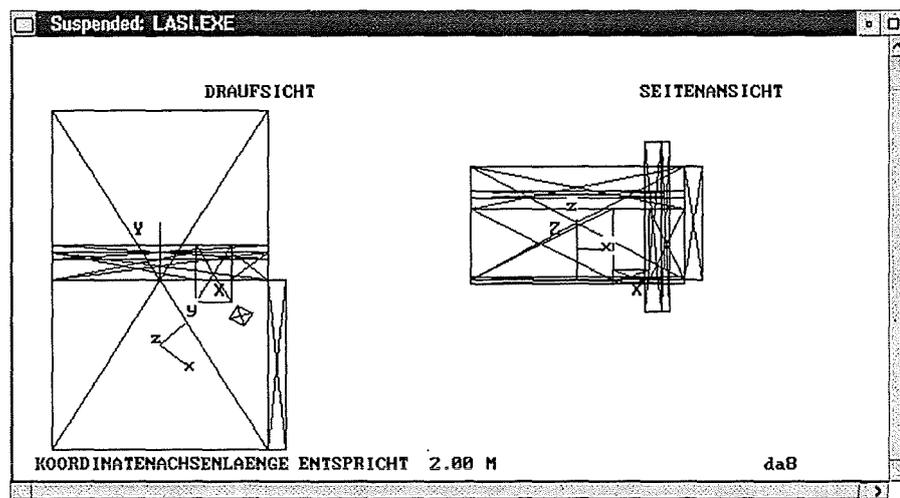


Abbildung C.2: Sensorsimulator LASI zur Berechnung idealer synthetischer Abstandsbilder: Innerhalb des Fensters ist die Szene aus Abb. 2.8a in verschiedenen Ansichten abgebildet. Alle Bestandteile sind aus einzelnen Quadrern zusammengesetzt.

## KHOROS

KHOROS ist eine Software-Umgebung, die als Werkzeug zur wissenschaftlichen Visualisierung dient [Khoros Group 93]. Khoros besteht aus mehreren Schichten interagierender Subsysteme. Neben Komponenten zum Entwurf von graphischen Benutzeroberflächen besteht die anwendungsspezifische Schicht aus einer Vielzahl von Routinen, die auf dem einheitlichen Khoros-Bildformat arbeiten. Die insgesamt 260 Routinen stammen aus den Bereichen Bildverarbeitung, Signalverarbeitung, Mustererkennung, Fernerkundung, Maschinen Sehen und Geometrische Informationssysteme.

Um die Möglichkeiten von Khoros nutzen zu können, wurden Rohdaten von binären, von mehrwertigen und von Abstandsbildern mit Hilfe von Khoros-Konvertierungsroutinen in das einheitliche *viff*-Bildformat von Khoros umgewandelt. Routinen können auf zwei Arten verwendet werden. Zum einen können Routinen als Kommandos mit verschiedenen Optionen ausgeführt werden. Zum anderen erlaubt eine graphische Programmieroberfläche, interaktiv ganze Operatorketten zu erstellen. Dabei stellt die Ausgabe einer Routine die Eingabe für eine andere dar. Die Oberfläche hat sich vor allem bei der Validierung von Bildverarbeitungsoperationen bewährt. Abb. C zeigt beispielsweise, wie die angepasste Mittlere Krümmung aus den fünf Gradientenbildern berechnet wird, ohne einen neuen Operator in einer Programmiersprache schreiben zu müssen. Verwendung haben vor allem Routinen aus den Khoros-Teilbereichen der Bildkonversion, Visualisierung und Animation, Bildverarbeitung und Bildarithmetik gefunden.

Beispielsweise kann die iterative Verbesserung von Korrespondenzwerten als Bildfolge von Strichzeichnungen der Szenenkantenstruktur und als Bildfolge von Strichzeichnungen einer Objektkantenstruktur visualisiert werden. Dabei wird eine Linie in jeder Strichzeichnung der Szenenkantenstruktur umso fetter gezeichnet, je größer die maximale Korrespondenzvariable in der gleichen Zeile der Korrespondenzmatrix ist. Analog wird eine Linie in jeder Strichzeichnung einer Objektkanten-

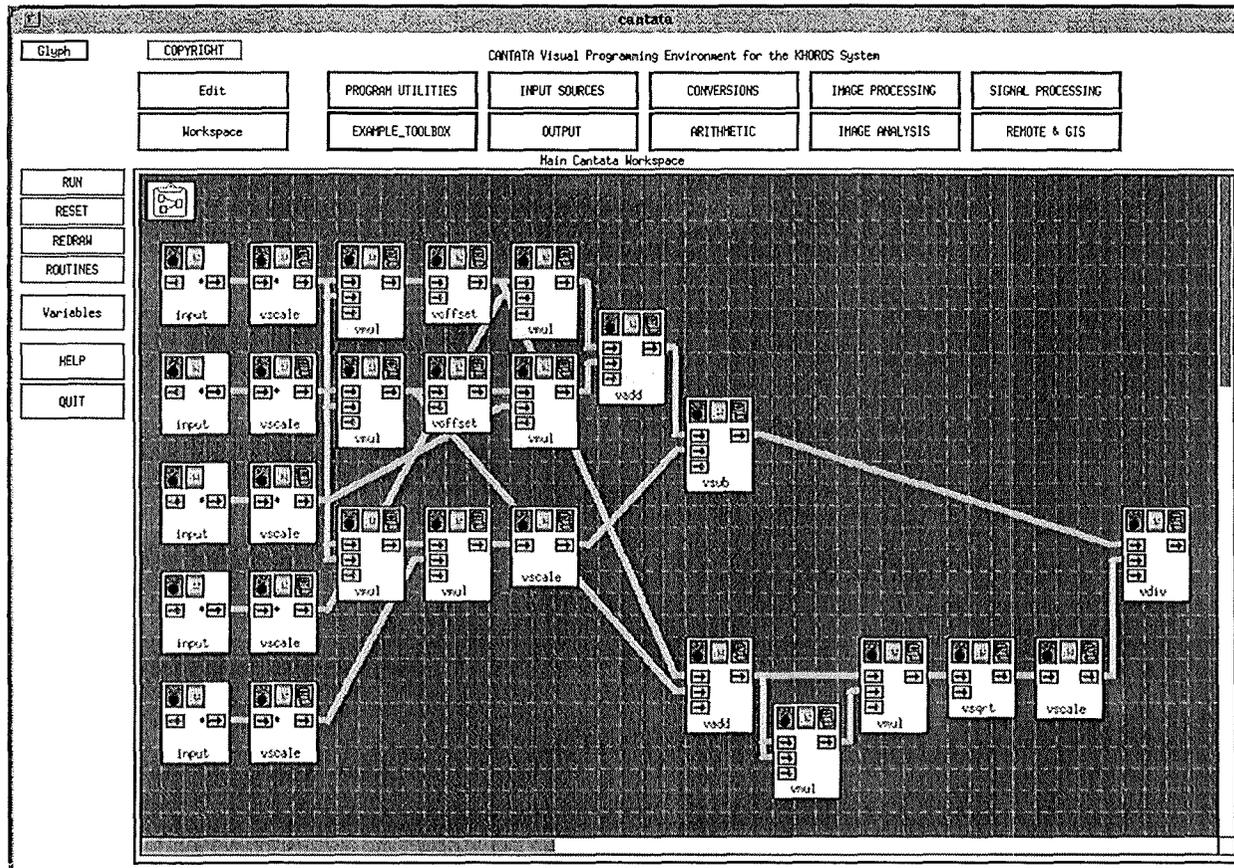


Abbildung C.3: Oberfläche von KHOROS zur visuellen Programmierung: Berechnung der angepassten Mittleren Krümmung aus den fünf Gradientenbildern mit Hilfe arithmetischer Operatoren, ohne einen neuen Operator in einer Programmiersprache schreiben zu müssen.

struktur umso fetter gezeichnet, je größer die maximale Korrespondenzvariable in der gleichen Spalte der Korrespondenzmatrix ist. Da von allen Korrespondenzvariablen einer Zeile bzw. Spalte nur die größte in der Strichdicke kodiert ist, geht daraus nicht hervor, welche Objekt- mit welcher Szenenkante korrespondiert. Man gewinnt vor allem einen Eindruck von der zeitlichen Reihenfolge von Zuordnungen. Strichzeichnungen wurden als Rasterbilder abgespeichert und mit dem Khoros-Werkzeug zur Animation von Rasterbildern animiert. Abbildungen C.4 und C.5 zeigen die Strichzeichnung der Szene des Benchmarks und einer charakteristischen Objektsansicht zu verschiedenen Zeitpunkten der iterativen Verbesserung.

### Application Visualization System

Das Application Visualization System, oder kurz AVS, ist ein Werkzeug zur graphischen Visualisierung großer mehrdimensionaler wissenschaftlicher Daten. Ähnlich wie in KHOROS können Applikationen erstellt werden, ohne in einer Programmiersprache wie C oder Pascal arbeiten zu müssen. AVS besteht aus Modulen, die zu Flußnetzwerken verbunden werden können. Module werden in drei Klassen eingeteilt. Filtermodule erlauben, aus gegebenen Daten nur die gewünschten herauszufiltern. Abbildungsmodule bilden gefilterte Daten auf andere Daten ab, die zum Beispiel für die graphische Darstellung geeigneter sind. Schließlich erzeugen Darstellungsmodule Graphiken und Bilder von abgebildeten Daten.

AVS wurde hier zur Visualisierung von Bildpunkt mengen, von Bildpunkt netzen und von Quaderszenen verwendet. Dazu wurden zum einen Konvertierungsroutinen implementiert, die Bildpunkte in eine Textdatei aus Szenenpunkten im AVS-Format umwandeln. Zum anderen wurde ein AVS-Abbildungsmodul implementiert, das eine Quaderszenenbeschreibung in eine AVS-Geometrie umwandelt, die durch vorhandene Darstellungsmodule schattiert dargestellt werden kann.

Szenenpunkte werden durch ein 4-Tupel aus den drei Komponenten der kartesischen Koordinate und dem Abstand beschrieben. Szenenpunktmengen wurden als Menge von 3D-Kugeln, als benachbarte 3D-Kugeln auf einem Netz, als Netzlinien oder als schattierte Fläche visualisiert.

AVS hat sich vor allem bei der Analyse der räumlichen Lage von verdeckenden Kantenpunkten bewährt. Abb. C.6 zeigt die Szenenkantenpunkte der Testszene aus Abb. 2.8 in ihrer perspektivischen Projektion. Dabei ist der Blickpunkt der perspektivischen Projektion verschieden von dem Sensorblickpunkt. Die Abstandswerte der einzelnen Kantenpunkte wurden zusätzlich durch Farbnuancen einzelner Kugeln kodiert.

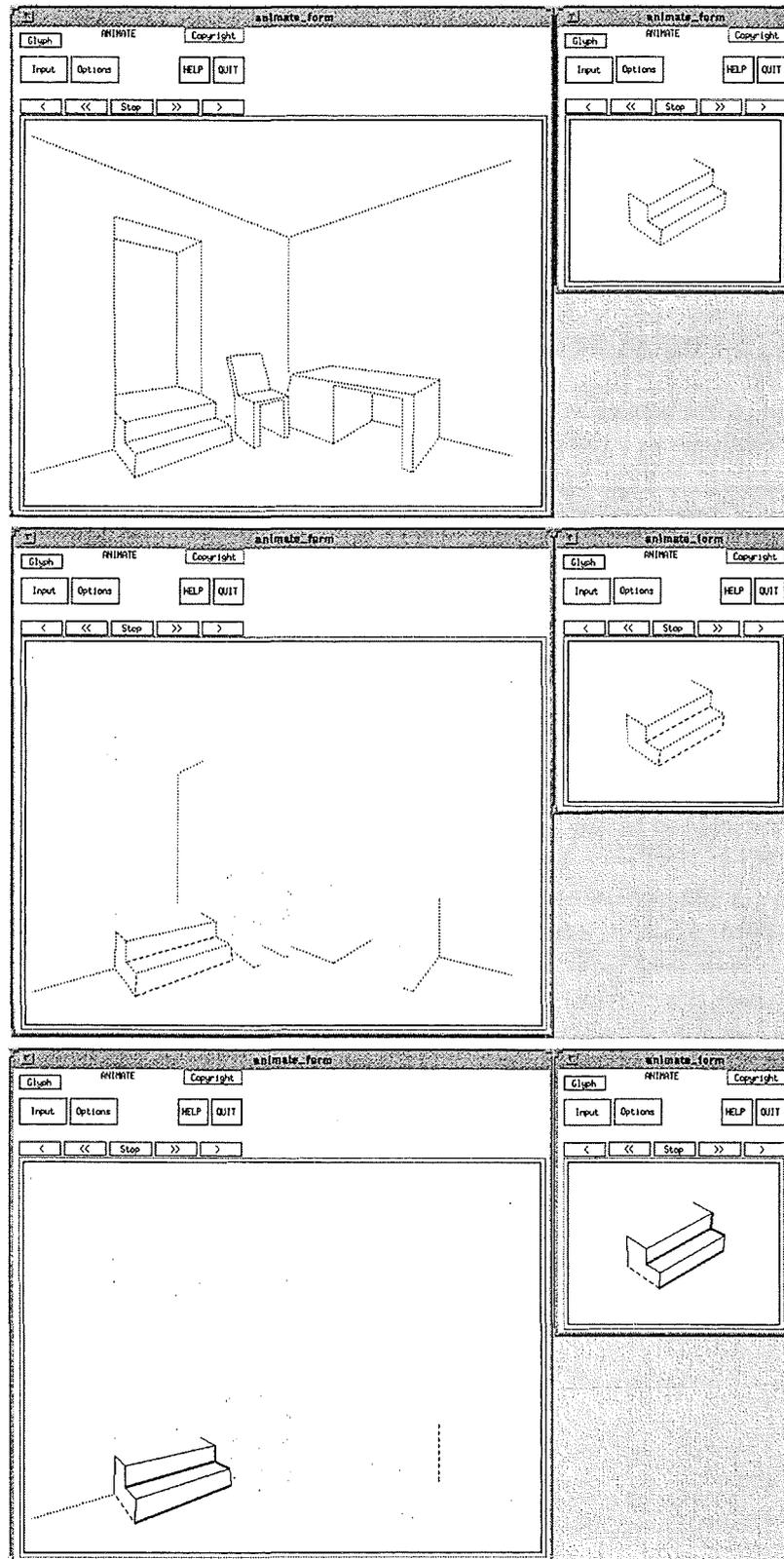


Abbildung C.4: Veränderung der Korrespondenzstärken der Kanten als Bildfolge: Die Veränderung der Korrespondenzwerte kann als Bildfolge visualisiert werden, indem in der Szenen- und Objektkantenstruktur Korrespondenzwerte einer Kante durch die Strichstärke dargestellt werden.

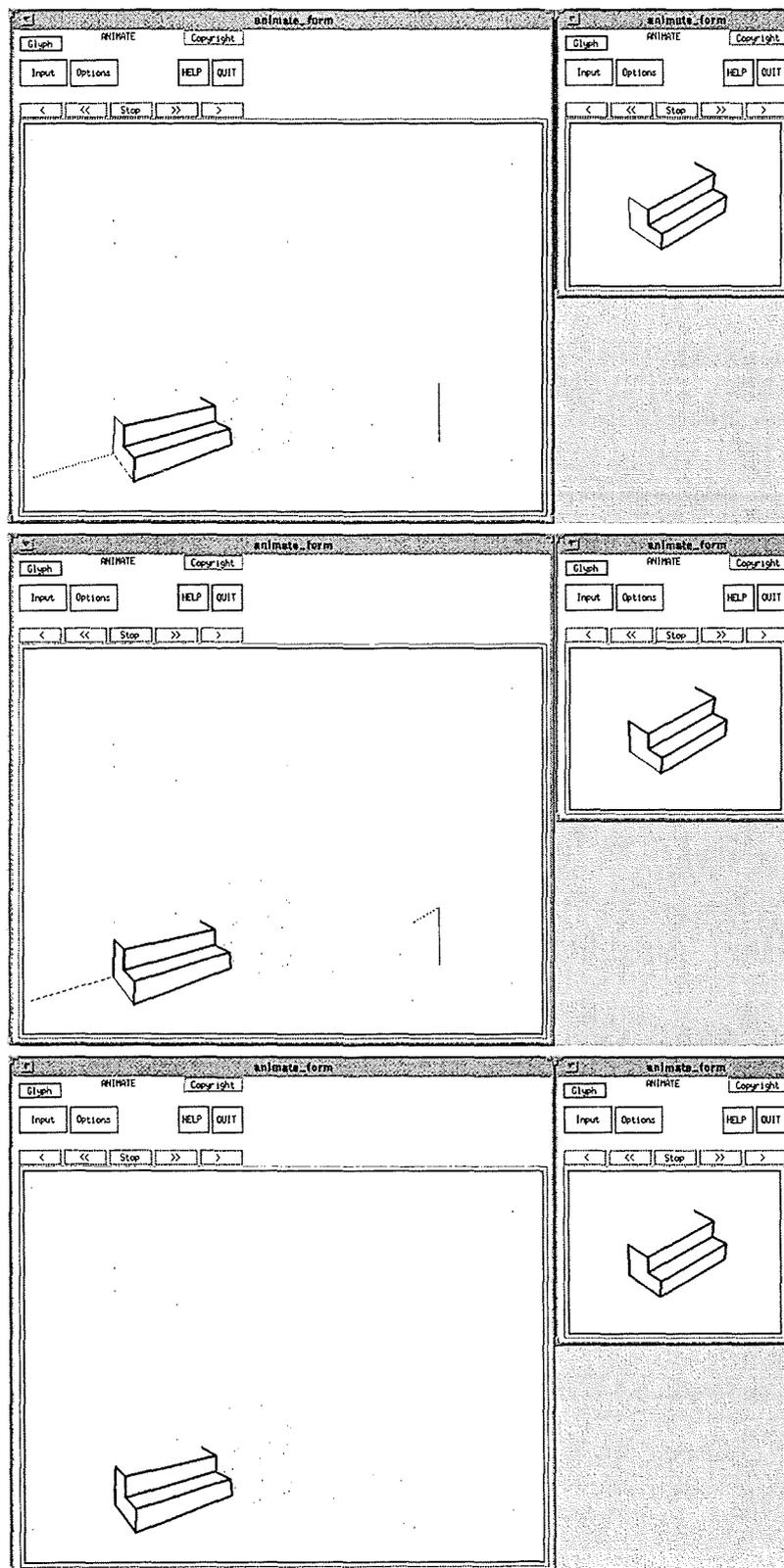


Abbildung C.5: Veränderung der Korrespondenzstärken der Kanten als Bildfolge (Fortsetzung): je höher die Korrespondenzwerte, desto dicker sind die Striche gezeichnet (keine Linie - gepunktet - gestrichelt - normal - fett). Im ersten der sechs Bilder sind alle Korrespondenzwerte gleich initialisiert.

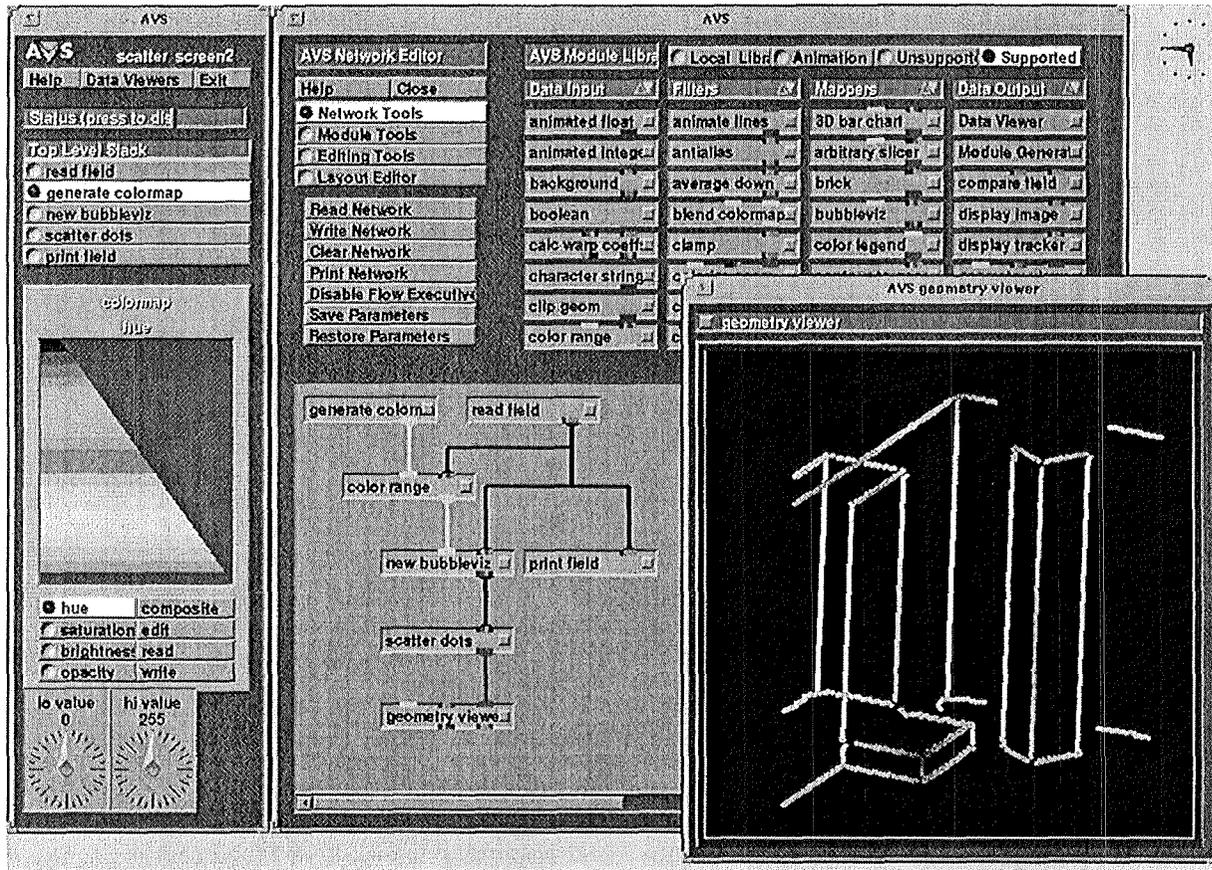


Abbildung C.6: Oberfläche von AVS zur visuellen Programmierung: Die Kantenpunkte der Szene in Abb. 2.8 sind in einem Fenster als schattierte Kugeln dargestellt.

# Literaturverzeichnis

- [Anderson *et al.* 77] J. A. ANDERSON, J. W. SILVERSTEIN, S. A. RITZ und R. S. JONES. Distinctive features, categorical perception, and propability learning: some applications of a neural network model. *Psychological Review* (1977).
- [Arman & Aggarwal 93] F. ARMAN und J. K. AGGARWAL. Model-based object recognition in dense-range images - a review. *ACM Computing Surveys* 25(1), 5-43 (März 1993).
- [AVS 92] AVS. „AVS User's Guide“. Advanced Visual Systems Inc., Waltham, MA, release 4 Auflage (Mai 1992).
- [Azarm *et al.* 93] K. AZARM, W. BOTT, F. FREYBERGER, D. GLÜER, J. HORN und G. SCHMIDT. Autonomiebausteine des mobilen roboters MACROBE. In G. SCHMIDT (Herausgeber), „9. Fachgespräch Autonome Mobile Systeme“, Seiten 81-94, München (Oktober 1993).
- [Barbosa & Lima 90] V. C. BARBOSA und P. M. V. LIMA. On the distributed parallel simulation of Hopfield's neural networks. *Software-Practice and Experience* 20(10), 967-983 (Oktober 1990).
- [Barrow & Burstall 76] H. G. BARROW und R. M. BURSTALL. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters* 4(4), 83-84 (Januar 1976).
- [Barrow & Popplestone 71] H. G. BARROW und R. J. POPPLESTONE. „Machine Intelligence“, Band 6, Kapitel Relational Descriptions in Picture Processing, Seiten 377-396. Elsevier, New York (1971).
- [Beaudet 78] P. R. BEAUDET. Rotationally invariant image operators. In „Proceedings of 4th International Conference on Pattern Recognition“, Seiten 579-583, Kyoto, Japan (November 1978).
- [Besl 88a] P. J. BESL. Active optical range imaging sensors. In „Advances in Machine Vision: Architectures and Applications“, Kapitel 1, Seiten 1-63. Springer (1988).
- [Besl 88b] P. J. BESL. „Surfaces in range image understanding“. Springer Series in Perception Engineering. Springer-Verlag, New York (1988).
- [Besl & Jain 86] P. J. BESL und R. A. JAIN. Invariant surface characteristics for 3D object recognition in range images. *Computer Vision, Graphics and Image Processing* 33, 33-80 (1986).
- [Besl & Jain 85a] P. J. BESL und R. C. JAIN. Intrinsic and extrinsic surface characteristics. In „Proceedings of the IEEE: Computer Vision and Pattern Recognition“, Seiten 226-233, San Francisco, CA (Juni 1985).

- [Besl & Jain 85b] P. J. BESL und R. C. JAIN. Three-dimensional object recognition. *ACM Computing Surveys* 17(1), 75–145 (1985).
- [Bilbro *et al.* 89] G. BILBRO, R. MANN, T. K. MILLER, W. E. SNYDER, D. E. VAN DEN BOUT und M. WHITE. Optimization by mean field annealing. In D. S. TOURETZKY (Herausgeber), „Advances in Neural Information Processing Systems“, Band 1, Seiten 91–95. Unbekannt (1989).
- [Bolles & Horaud 83] R.C. BOLLES und P. HORAUD. 3DPO: a three-dimensional part orientation system. *International Journal of Robotics Research* 5(3), 3–26 (1983).
- [Bouzerdoum & Pattison 93] A. BOUZERDOUM und T. R. PATTISON. Neural network for quadratic optimization with bound constraints. *IEEE Transactions on Neural Networks* 4(2), 293–304 (1993).
- [Braun 90] H. BRAUN. „Massiv parallele Algorithmen für kombinatorische Optimierungsprobleme und ihre Implementierung auf einem Parallelrechner“. Dissertation, Universität Karlsruhe, Karlsruhe (1990).
- [Bräunl 93] T. BRÄUNL. „Parallele Programmierung: eine Einführung“. Vieweg, Braunschweig (1993).
- [Charkavaty & Freeman 82] I. CHARKAVATRY und H. FREEMAN. Characteristic views as a basis for three-dimensional object recognition. In „Proceedings of SPIE“, Band 336, Seiten 37–45, Arlington, VA (1982).
- [Chen & Lin 91] T.-W. CHEN und W.-C. LIN. A neural network approach to CSG-based 3-D object recognition. In „Proceedings Conference on Computer Vision and Pattern Recognition“, Seite 718, Lahaina, Maui, Hawaii (Juni 1991).
- [Chen & Hsu 88] Y.-S. CHEN und W.-H. HSU. A modified fast parallel algorithm for thinning digital patterns. *Pattern Recognition Letters* 7, 99–106 (Februar 1988).
- [Chin & Dyer 86] R. T. CHIN und C. R. DYER. Model-based recognition in robot vision. *ACM Computing Surveys* 18(1), 67–108 (März 1986).
- [Christensen *et al.* 94] H.I. CHRISTENSEN, N.O. KIRKEBY, S. KRISTENSEN, L. KNUDSEN und E. GRANUM. Model-driven vision for in-door navigation. *Robotics and Autonomous Systems* 12, 199–207 (1994).
- [Clowes 71] M. B. CLOWES. On seeing things. *Artificial Intelligence* 2, 79–116 (1971).
- [Cramer 91] M. CRAMER. Lasi - Laserscanner-Simulator. Ire-bericht, Kernforschungszentrum Karlsruhe (1991).
- [d’Acierno & Vaccaro 94] A. D’ACIERNO und R. VACCARO. On parallelizing recursive neural networks on coarse-grained parallel computers. *Parallel Computing* 20, 245–256 (1994).
- [Davies 90] E. R. DAVIES. „Machine Vision: Theory, Algorithms, Practicalities“. Academic Press, London (1990).
- [Duda & Hart 73] R. O. DUDA und P. E. HART. „Pattern Classification and Scene Analysis“. Wiley, New York (1973).

- [Elektronik 92] ELEKTRONIK. Vier unter einem Dach. *Elektronik* 9, 32–33 (September 1992).
- [Eun et al. 91] S. B. EUN, S. R. MAENG und H. YOON. Parallel simulation of Hopfield neural networks on distributed-memory computers. In „Proceedings of International Joint Conference on Neural Networks“, Seiten 2199–2203, Singapore (1991).
- [Fan 90] T. J. FAN. „Describing and recognizing 3-D objects using surface properties“. Springer Series in Perception Engineering. Springer-Verlag, New York (1990).
- [Fan et al. 87] T.-J. FAN, G. MEDIONI und R. NEVATIA. Segmented descriptions of 3-D surfaces. *IEEE Journal of Robotics and Automation* 3(6), 527–538 (Dezember 1987).
- [Garey & Johnson 79] M. R. GAREY und D. S. JOHNSON. „Computers and Intractability: a guide to the theory of NP-completeness“. Books in the Mathematical Sciences. Freeman and Company, New York (1979).
- [Ghosal & Mehrotra 92] S. GHOSAL und R. MEHROTRA. Range image segmentation using Zernike moment-based generalized edge detector. In „Proceedings of IEEE International Conference on Robotics and Automation“, Seiten 1584–1589, Nice, Frankreich (Mai 1992).
- [Godin & Levine 89] G. D. GODIN und M. D. LEVINE. Structured edge map of curved objects in a range image. In „Proceedings Conference on Computer Vision and Pattern Recognition“, Seiten 276–281 (Juni 1989).
- [Grimson 90] W.E.L. GRIMSON. „Object recognition by computer: the role of geometric constraints“. MIT-Press, Cambridge, MA (1990).
- [Gupta & Krishnapuram 91] S. GUPTA und R. KRISHNAPURAM. Morphologic edge detection in range images. In „Image Algebra and Morphological Image Processing“, Band 2, Seiten 335–346 (Juli 1991).
- [Haralick & Shapiro 93a] R. M. HARALICK und L. G. SHAPIRO. „Computer and Robot Vision“, Band 1. Addison-Wesley, Reading, MA (1993).
- [Haralick & Shapiro 93b] R. M. HARALICK und L. G. SHAPIRO. „Computer and Robot Vision“, Band 2. Addison Wesley, Reading, MA (1993).
- [Haralick et al. 83] R. M. HARALICK, L. T. WATSON und T. J. LAFFEY. The topographic primal sketch. *International Journal of Robotics* 2(1), 50–72 (1983).
- [Hebert & Kanade 85] M. HEBERT und T. KANADE. The 3D-profile method for object recognition. In „Proceedings International Conference on Computer Vision and Pattern Recognition“, Seiten 458–463 (1985).
- [Hebert & Krotkov 92] M. HEBERT und E. KROTKOV. 3D measurements from imaging laser radars: how good are they? *Image and Vision Computing* 10(3), 170–178 (1992).
- [Herault & Niez 91] L. HERAULT und J. J. NIEZ. Neural networks and combinatorial optimization: a study of NP-complete graph problems. In E. GELLENBE (Herausgeber), „Neural networks: advances and applications“. North-Holland, Amsterdam (1991).

- [Herman 85] M. HERMAN. Generating detailed scene descriptions from range images. In „International Conference on Robotics and Automation“, Seiten 426–431, St. Louis (März 1985).
- [Hopfield & Tank 85] J. J. HOPFIELD und D. W. TANK. Neural computation of decisions in optimization problems. *Biological Cybernetics* **52**, 141–152 (1985).
- [Huffman 71] D. A. HUFFMAN. Impossible objects as nonsense sentences. In R. MELTZER und D. MICHIE (Herausgeber), „Machine Intelligence“, Band 6, Seiten 295–323. Edinburgh University Press (1971).
- [Ibeo 90a] IBEO. „Gerätebeschreibung LADAR 3D-Scanner“. IBEO GmbH Lasertechnik, Hamburg (September 1990).
- [Ibeo 90b] IBEO. „Software Beschreibung 3D Scanner - Anhang B: Feinspezifikation“. IBEO GmbH Lasertechnik, Hamburg (Februar 1990).
- [Inokuchi *et al.* 82] S. INOKUCHI, T. NITA, F. MATSUDA und Y. SAKURAI. A three-dimensional edge-region operator for range pictures. In „Proceedings Sixth International Conference on Pattern Recognition“, Seiten 918–920 (1982).
- [Jarvis 93] R. A. JARVIS. Range sensing for computer vision. In A. K. JAIN und P. J. FLYNN (Herausgeber), „Three-dimensional object recognition systems“, Band 1 aus „Advances in Image Communication“, Seiten 17–56. Elsevier (1993).
- [Kanade 78] T. KANADE. Region segmentation: signal vs. semantics. In „Proceedings of International Joint Conference on Pattern Recognition“, Seiten 95–105 (1978).
- [Kanade *et al.* 90] T. KANADE, A. GRUSS und L. R. CARLEY. A VLSI sensor based rangefinding system. In H. MIURA und S. ARIMOTO (Herausgeber), „Fifth International Symposium on Robotics Research“, Seiten 49–56 (August 1990).
- [Khoros Group 93] KHOROS GROUP. „Khoros Manual Volume I“. Departement of Electrical and Computer Engineering, University of New Mexico, Albuquerque (1993).
- [Kirkpatrick *et al.* 83] S. KIRKPATRICK, C. D. GELATT und M. P. VECCHI. Optimization by simulated annealing. *Science* **220**, 671–680 (Mai 1983).
- [Kitchen 80] L. KITCHEN. Relaxation applied to matching quantitative relational structures. *IEEE Transactions on Systems, Man, and Cybernetics* **10**(2), 96–101 (Februar 1980).
- [Kittler & Illingworth 85] J. KITTLER und J. ILLINGWORTH. Relaxation labelling algorithms. *Image and Vision Computing* **3**(4), 206–216 (November 1985).
- [Kluge *et al.* 90] K. KLUGE, T. KANADE und H. KUGA. Car recognition by the CMU navlab. In T. KANADE (Herausgeber), „Vision and Navigation: The Carnegie Mellon NavLab“, Kapitel 6, Seiten 95–96. Elsevier (1990).

- [Knieriemen 91] T. KNIERIEMEN. „Autonome mobile Roboter: Sensordateninterpretation und Weltmodellierung zur Navigation in unbekannter Umgebung“, Band 80 aus „Reihe Informatik“. BI Wissenschaftsverlag (1991).
- [Koenderink & Van Dorn 79] J. J. KOENDERINK und A. J. VAN DORN. The internal representation of solid shape with respect to vision. *Biological Cybernetics* 32, 211–216 (Januar 1979).
- [Koprek 92] J. KOPREK. Vorverarbeitung und Merkmalsextraktion von Entfernungsbildern mit Transputern. Diplomarbeit, Universität Karlsruhe (TH), Institut für Hochfrequenztechnik und Elektronik (Oktober 1992).
- [Kuner 87] P. KUNER. Reducing relational graph matching task to integer programming problems, which are solved by a consistent labeling approach. In J.-O. EKLUNDH (Herausgeber), „Proceedings Fifth Scandinavian Conference on Image Analysis“, Seiten 127–134 (Juni 1987).
- [Kuner & Ueberreiter 88] P. KUNER und B. UEBERREITER. Pattern recognition by graph matching - combinatorial versus continuous optimization. *International Journal of Pattern Recognition and Artificial Intelligence* 2(3), 527–542 (1988).
- [Kung & Hwang 89] S. Y. KUNG und J. N. HWANG. A unified systolic architecture for artificial neural networks. *Journal of Parallel Distributed Computing* 6, 358–387 (1989).
- [Langridge 84] D. J. LANGRIDGE. Detection of discontinuities in the first derivatives of surfaces. *Computer Vision, Graphics and Image Processing* 27, 291–308 (1984).
- [Lee et al. 94] P.-S. LEE, Y.-E. SHEN und L.-L. WANG. Model-based location of automated guided vehicles in the navigation sessions by 3D computer vision. *Journal of Robotic Systems* 11(3), 181–195 (1994).
- [Lenz 90] R. LENZ. „Group theoretical methods in image processing“, Band 413 aus „Lecture Notes in Computer Science“. Springer, Berlin (1990).
- [Li 92] S. Z. LI. Toward 3D-vision from range images: an optimization framework and parallel networks. *CVGIP: Image Understanding* 55(3), 231–260 (Mai 1992).
- [Lin et al. 91] W.-C. LIN, F.-Y. LIAO, C.-K. TSAO und T. LINGUTLA. A hierarchical multiple-view approach to three-dimensional object recognition. *Neural Networks* 2(1), 84–92 (Januar 1991).
- [Lloyd 83] S. A. LLOYD. An optimization approach to relaxation labelling algorithms. *Image and Vision Computing* 1(2), 85–91 (Mai 1983).
- [Malik 87] J. MALIK. Interpreting line drawings of curved objects. *International Journal of Computer Vision* 1(1), 73–103 (1987).
- [Mitiche & Aggarwal 83] A. MITICHE und J. K. AGGARWAL. Detection of edges using range information. *IEEE Transactions on Pattern Recognition and Machine Intelligence* 5(2), 174–178 (März 1983).

- [Nagel 79] H.-H. NAGEL. Über die Repräsentation von Wissen zur Auswertung von Bildern. In FOITH (Herausgeber), „Angewandte Szenenanalyse“, Nummer 20 in Informatik Fachberichte, Seiten 3–21. Springer-Verlag (Oktober 1979).
- [Nasrabadi & Li 91] N. M. NASRABADI und W. LI. Object recognition by a Hopfield neural network. *IEEE Transactions on Systems, Man and Cybernetics* 21(6), 1523–1535 (November 1991).
- [Nevatia & Price 82] NEVATIA und PRICE. Relaxations operations. *IEEE PAMI* 9(9), 99–99 (1982).
- [Orlov 93] A. A. ORLOV. Mirror beam-control systems for locators. *Soviet Journal of Optical Technology* 60(1), 35–38 (Januar 1993).
- [Parvin & Medioni 91] B. PARVIN und G. MEDIONI. A layered network for the correspondence of 3D objects. In „Proceedings of IEEE International Conference on Robotics and Automation“, Seiten 1808–1813, Sacramento, CA (April 1991).
- [Peterson & Anderson 87] C. PETERSON und J. R. ANDERSON. A mean field theory learning algorithm for neural networks. *Complex Systems* 1, 995–1019 (1987).
- [Reeves 91] REEVES. Parallel computing for computer vision. *IEEE Software* 8(6), 51–59 (1991).
- [Rosenfeld *et al.* 76] A. ROSENFELD, R. A. HUMMEL und S. W. ZUCKER. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics* 6(6), 420–433 (Januar 1976).
- [Sheridan 87] T. B. SHERIDAN. Telerobotics. In „10th Triennial World Congress“, Seiten 67–81, München (1987). IFAC.
- [Shirai & Suwa 71] Y. SHIRAI und M. SUWA. Recognition of polyhedrons with a range finder. In „Proceedings 2nd International Joint Conference on Artificial Intelligence“, Seiten 80–87, London, England (1971).
- [Smith & Kanade 85] D.R. SMITH und T. KANADE. Autonomous scene description with range imagery. *Computer Vision, Graphics and Image Processing* 31(3), 322–334 (September 1985).
- [Sugihara 79] K. SUGIHARA. Range-data analysis guided by a junction dictionary. *Artificial Intelligence* 12(1), 41–69 (1979).
- [Suk & Bhandarkar 93] M. SUK und S.M. BHANDARKAR. „Three-dimensional object recognition from range images“. Computer Science Workbench. Springer (1993).
- [Swietlik 94] A. SWIETLIK. „Geometrische Rekonstruktion von dreidimensionalen Szenen aus Abstandsbildern“. Dissertation, Universität Karlsruhe (TH), Fakultät für Informatik, Karlsruhe (Februar 1994).
- [Tagliarini *et al.* 91] G. A. TAGLIARINI, J. F. CHRIST und E. W. PAGE. Optimization using neural networks. *IEEE Transactions on Computers* 40(12), 1347–1358 (Dezember 1991).
- [Tresp 91a] V. TRESP. A neural architecture for 2-D and 3-D vision. In B. RADIG (Herausgeber), „Mustererkennung 1991“, Seiten 437–445. Springer (Oktober 1991).

- [Tresp 91b] V. TRESP. A neural network approach for three-dimensional object recognition. In R. LIPPMANN (Herausgeber), „Advances in Neural Information Processing Systems“, Band 3. Morgan Kaufman, San Mateo, CA (1991).
- [Valavanis & Saridis 92] K. P. VALAVANIS und G. N. SARIDIS. „An intelligent robot operating system for emergency situations in a nuclear power plant“, Band Intelligent Robotic Systems: Theory, Design and Applications aus „Kluwer International Series in Engineering and Computer Science“, Kapitel 6.2, Seiten 156–185. Kluwer Academic Publishers (1992).
- [Van Den Bout & Miller 90] D. E. VAN DEN BOUT und T. K. MILLER. Graph partitioning using annealed neural networks. *IEEE Transactions on Neural Networks* 1(2), 192–203 (Juni 1990).
- [Vogelgesang 91a] V. G. VOGELGESANG. unveröffentlichter Bericht, (1991).
- [Vogelgesang 91b] V. G. VOGELGESANG. unveröffentlichter Bericht, (1991).
- [Vogelgesang 94] V. G. VOGELGESANG. Attributed graph extraction from high resolution range images. In „Symposium on Signal Processing, Robotics and Neural Networks“, Lille, France (April 1994).
- [Vogelgesang et al. 92] V. G. VOGELGESANG, W. JAKOB und E. STRATMANN. Teilsysteme für autonome Operationen in halbstrukturierter Umgebung. In W. GÖRKE, H. RININSLAND und M. SYRBE (Herausgeber), „Information als Produktionsfaktor“, Seiten 337–343. Gesellschaft für Informatik, Springer (Oktober 1992).
- [Vosselman 92] G. VOSSELMAN. „Relational Matching“, Band 628 aus „Lecture Notes in Computer Science“. Springer (1992).
- [Waldschmidt 95] K. WALDSCHMIDT (Herausgeber). „Parallelrechner: Architekturen - Systeme - Werkzeuge“. Leitfäden der Informatik. B. G. Teubner, Stuttgart (1995).
- [Waltz 75] D. WALTZ. „The Psychology of Computer Vision“, Kapitel Understanding of line drawings of scenes with shadows, Seiten 19–91. McGraw-Hill, New York, NY (1975).
- [Wani & Batchelor 94] M. A. WANI und B. G. BATCHELOR. Edge-region-based segmentation of range images. *IEEE Transactions on Pattern Recognition and Machine Intelligence* 16(3), 314–319 (März 1994).
- [Weems et al. 91] C.C. WEEMS, C. BROWN, J.A. WEBB, T. POGGIO und J.R. KENDER. Parallel programming in the DARPA strategic computing vision program. *IEEE Expert* Seiten 22–38 (Oktober 1991).
- [Wei 89] J. WEI. „Modell-gesteuerte Szenen-Interpretation durch Fusion von Intensitäts- und Abstandsbildern“. Dissertation, Universität Karlsruhe (TH), Fakultät für Informatik, Karlsruhe (Dezember 1989).
- [Weidner 94] A. WEIDNER. Extraktion partieller Objektgraphen in Szenengraphen mittels Neuronaler Netze. Diplomarbeit, Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme, Karlsruhe (April 1994).

- [Yamamoto *et al.* 93] M. YAMAMOTO, P. BOULANGER, J. A. BERARDIN und M. RIOUX. Direct estimation of range flow on deformable shape from a video rate range camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(1), 82-89 (Januar 1993).
- [Zell 94] A. ZELL. „Simulation neuronaler Netze“. Addison-Wesley, Bonn (1994).