

KfK 5167  
Februar 1993

# **Oberflächendarstellung zur Repräsentation und Erkennung von Objekten aus Abstandsbildern**

P. Kohlhepp  
Institut für Angewandte Informatik

**Kernforschungszentrum Karlsruhe**



**KERNFORSCHUNGSZENTRUM KARLSRUHE**  
**Institut für Angewandte Informatik**

**KfK 5167**

**Oberflächendarstellung zur Repräsentation  
und Erkennung von Objekten aus Abstandsbildern**

**P. Kohlhepp**

**Kernforschungszentrum Karlsruhe, GmbH, Karlsruhe**

Als Manuskript gedruckt  
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH  
Postfach 3640, 7500 Karlsruhe 1

**ISSN 0303-4003**

# Boundary Representations for the Reconstruction and Recognition of Objects from Range Images

## Abstract

This report documents the current state and progress in a range image processing project at the Nuclear Research Centre Karlsruhe (KfK). A conceptual basis is laid for the reconstruction and recognition of 3D objects from range images produced by a laser radar sensor (2D LADAR).

First, the project SOMBRERO (surface oriented model building for the reconstruction and recognition of objects), its terminology, and prerequisites are defined. The 2D LADAR testbed, a transputer network for preprocessing, and some preliminary results on data acquisition and curvature extraction are presented. Several future applications of range image processing are proposed.

Next, a uniform augmented boundary representation is defined for both object models and scenes, forming the interface between preprocessing, reconstruction, and recognition. The recognition algorithm uses a set of location and occlusion invariant constraints for hypothesis generation, and location dependent constraints for hypothesis testing.

The report also covers organizational aspects like work packages, milestones, and a time schedule. Within a joint effort, part of the development will be delegated to the Forschungszentrum Informatik (FZI) at Karlsruhe.

## Zusammenfassung

Dieser Bericht dokumentiert den aktuellen Stand der Arbeiten zur Entfernungsbildverarbeitung mit dem 2D LADAR, und legt die konzeptionellen Grundlagen für die geplanten Arbeiten zur Rekonstruktion und Wiedererkennung von Objekten aus Entfernungsbildern.

Die Problemstellung, ihre Terminologie sowie die konkreten Randbedingungen für das Projekt SOMBRERO werden definiert. Der 2D LADAR - Teststand und erste Ergebnisse zur Datenerfassung und Merkmalsextraktion werden beschrieben und zukünftige Anwendungen der Entfernungsbildverarbeitung aufgezeigt.

Es wird ferner eine Objektrepräsentation für Szenen und Modellobjekte auf der Basis von Oberflächen definiert. Sie bildet die Schnittstelle zwischen den Verfahren der Vorverarbeitung und der Wiedererkennung. Ein Konzept für die Wiedererkennung mit teils lage- und verdeckungsunabhängigen, teils lageabhängigen Beschränkungen wird entworfen.

Der Bericht enthält auch Vorschläge zur organisatorischen Abwicklung des Projektes in Form von Arbeitspaketen, Zeitplan und einem Entwicklungsauftrag an das Forschungszentrum Informatik (FZI) mit Meilensteinen.

# Inhaltsverzeichnis

<b>1.</b>	<b>Projektbeschreibung</b>	<b>3</b>
1.1	Problemstellung	3
1.2	Anwendungen	8
1.2.1	Teststand zur Entfernungsdatenaufnahme	8
1.2.2	Längerfristige Anwendungen	14
1.3	Randbedingungen	15
1.3.1	Einsatzumgebung	15
1.3.2	Objekte	16
1.3.3	Sensordatenaufnahme	17
1.3.4	Ausgangsdaten	19
1.4	Prinzipien und Forschungsziele	20
1.4.1	Das Geometriemodell zwischen Rekonstruktion und Wiedererkennung	20
1.4.2	Explizites Anwendungswissen statt impliziter Heuristik	23
1.5	Systemarchitektur und Komponenten	27
1.5.1	Überblick	27
1.5.2	Vorverarbeitung	34
1.5.3	Implementierung auf Parallelrechnern	42
<b>2.</b>	<b>Grobkonzept zur Objektwiedererkennung</b>	<b>46</b>
2.1	Objektrepräsentation	46
2.1.1	Geometrische Grundelemente	46
2.1.1.1	Attribute	46
2.1.1.2	Relationen	48
2.1.2	Szenenrepräsentation	50
2.1.3	Modellobjekte	51
2.2	Objekterkennung als konsistentes Zuordnungsproblem	51
2.3	Invariante Beschränkungen	53
2.4	Propagierung von Beschränkungen	61
2.4.1	Diskrete Relaxation	61
2.4.2	Generieren und Testen von Hypothesen	64
2.4.3	Vergleich der beiden Verfahren	67
2.4.4	Kombiniertes Verfahren	69
2.4.5	Strategische Überlegungen	73
<b>3.</b>	<b>Projektentwicklung</b>	<b>76</b>
3.1	Arbeitspakete und Zeitplan	76
3.2	Objektwiedererkennung	80
3.2.1	Entwicklungsauftrag	80
3.2.1.1	Grob- und Feinspezifikation	80
3.2.1.2	Auswahl der Entwicklungsumgebung	81
3.2.1.3	Synthetische Testdaten	82
3.2.1.4	Revision, Erweiterung und Erprobung	82
3.2.1.5	Parallelisierung der Erkennung	84
3.2.2	Meilensteine	84
<b>4.</b>	<b>Zusammenfassung und Ausblick</b>	<b>86</b>
<b>5.</b>	<b>Literatur</b>	<b>88</b>

# 1. Projektbeschreibung

## 1.1 Problemstellung

Industrieroboter unterstützen den Menschen in der Fertigung, beim Transport oder bei Inspektions- und Wartungsarbeiten in unzugänglichen oder gefährlichen Arbeitsumgebungen, und in vielen anderen Bereichen. Um sie aber effektiver einzusetzen, muß ihre Flexibilität erhöht und der Programmieraufwand drastisch verringert werden /13/. Ein wichtiger Entwicklungsansatz in diese Richtung, am dem sich das KfK beteiligt, sind (teil-)autonome Roboter und Transportfahrzeuge.

Autonomie bedeutet Selbständigkeit in der Durchführung von Aufgaben, also die Fähigkeit, sich an verändernde oder nicht exakt vorherzuplanende Umgebungsbedingungen anzupassen. Dazu muß ein teil-autonomes System den Zustand seiner Arbeitsumgebung erkennen (**Perzeption**) und daraus zielgerichtete Handlungsfolgen ableiten können (**Planung**). Informationsquellen zur Erkennung der Umgebung sind optische, taktile, akustische oder thermische **Sensoren**.

Wir beschränken uns in diesem Teilvorhaben auf den Teilbereich der Perzeption, der die **räumlichen** Eigenschaften der Umwelt mit **bildgebenden**, optischen Sensoren, insbesondere **Abstandssensoren** erkennt (**Maschinensehen, Objekterkennung**). Ein **Objekterkennungssystem** beantwortet, gestützt auf Vorwissen über seine Umgebung Fragen wie

- befinden sich im Blickbereich Gegenstände (Objekte) eines bestimmten Typs?
- welches sind ihre Positionen, Orientierungen und ggf. genauen Abmessungen?
- in welchen räumlichen oder (bei Bildfolgen) zeitlichen Beziehungen stehen die Objekte ?

Ein solches System sollte, wie das menschliche Sehen, mit realistischen Randbedingungen wie komplexen Formen, unbekannter Objektlage, Verdeckung, fehlerhafter oder unvollständiger Bildinformation zurechtkommen.

Wir geben zuerst einige Literaturhinweise zum Stand der Forschung und Entwicklung, stellen unsere Testumgebung zur Entfernungsdatenverarbeitung vor und skizzieren einige zukünftige Anwendungen (1.2). Abschnitt 1.3 schreibt die konkreten Randbedingungen dieses Vorhabens für einen überschaubaren Zeitraum von ca. 2 Jahren fest. Abschnitt 1.4 zeigt grundlegende Prinzipien und Forschungsperspektiven auf. Abschnitt 1.5 stellt die Grobarchitektur des zu entwickelnden Systems vor, darunter auch in knapper Form die Komponenten zur Datenvorverarbeitung und die bereits erzielten Ergebnisse.

Den Schwerpunkt dieses Berichtes bilden jedoch die Objektrepräsentation und die Verfahren zur Wiedererkennung (Kap. 2). Auch auf die organisatorische Projektabwicklung gehen wir ein (Kap. 3).

Der Bericht enthält aus Umfangs- und Zeitgründen keine ausführliche Darstellung, Klassifizierung und Bewertung der bekannten Arbeiten zur Objekterkennung. Vielmehr wird bei konkreten Einzelproblemen auf die Literatur eingegangen. Folgende Arbeiten - zumeist auf Entfernungsbildern basierend - wurden aber für die Arbeit herangezogen und untersucht:

- das (Intensitätsbild-)Erkennungssystem ACRONYM von Brooks /8/, das verallgemeinerte Zylinder als Objektbeschreibung und 'ribbons' (ellipsenförmige Bänder) als Merkmale einsetzt,
- das 3DPO-System von Bolles /7/,
- der EGI ('Extended Gaussian Image') Ansatz von Horn und Ikeuchi /27//28/ zur Charakterisierung konvexer, nichtverdeckter Einzelobjekte,
- das Objekterkennungssystem von Oshima und Shirai /53/ auf der Basis einer Oberflächendarstellung,
- das ebenfalls facettierte Oberflächendarstellungen verwendende System von Bhanu /5/,
- das System von Grimson und Lozano-Perez /20//21/ für polyedrische Objekte,
- der Ansatz von Faugeras und Hebert /16/, der sich auf Punkte, Linien und Ebenen als Erkennungsmerkmale beschränkt,
- das auf einer ansichten-orientierten Objektdarstellung beruhende System von Ikeuchi /32/ für das Spezialproblem "Griff in die Kiste",
- die Arbeit von Fan /15/, die als eine von wenigen ein vollständiges Objekterkennungssystem auf der Basis von Oberflächen einschließlich der Vorverarbeitung und der Gruppierung von Flächen zu Objektinstanzen beschreibt,

- das wissensbasierte System 3D-Form von Walker, Hermann und Kanade /65/ zur Interpretation von Umrisszeichnungen (Drahtmodellen) anhand von Objektklassenhierarchien,
- die Arbeiten der Bildverarbeitungsgruppe am Robot Vision Laboratory der Purdue University unter A.C.Kak, darunter das wissensbasierte System PSEIKI von Andress und Kak /1/, der Ansatz von C.A.Kak et. al. /36/, der besonders die einheitliche, CAD-generierbare Objekt- und Szenenbeschreibung betont, und das Erkennungssystem von Chen und Kak /9/, das die Komplexitäts- und Effizienzprobleme der Wiedererkennung in den Mittelpunkt stellt und effiziente Datenstrukturen zur Merkmalsrepräsentation vorschlägt,
- die Arbeiten von Flynn und Jain /17/ zu CAD-basierenden, relationalen Objektrepräsentationen,
- verschiedene Arbeiten zu den verwandten Problemen der Kartierung und Ortsbestimmung bei autonomen Fahrzeugen, u.a Shaffer et.al. /57/, Onoguchi et. al. /52/, das Oxford Autonomous Guided Vehicle (AGV) von Reid et.al. /54/, die Kartierungs- und Ortsbestimmungsansätze von Lang u. Wong /43/ sowie die von Hoppen, Knieriemer und v.Puttkamer an der Universität Kaiserslautern /26/.

Die Problemstellung sowie einen guten Überblick über den Stand der Forschung und Entwicklung zur Entfernungsbildverarbeitung geben die Übersichtsartikel von Besl und Jain /4/, Chin und Dyer /10/, das Lehrbuch von R.C.Jain und A.K.Jain /35/, darin insbesondere der Artikel von G.Stockman zur Objekterkennung /58/, die Dissertation von T.J.Fan /15/ und die ebenfalls auf diesen Quellen beruhende Recherche von Vogelgesang /63//64/.

**Neuronale Netze (NNe)**, die für bestimmte, meist zweidimensionale Erkennungsaufgaben wie Handschriften-Symbole, Kfz-Kennzeichen oder Gesichter /42/ mit Erfolg eingesetzt werden, werden in diesem Vorhaben nicht berücksichtigt. Der interessierte Leser sei hier z.B. auf Vogelgesang /64/ verwiesen. NNe im gut trainierten Zustand haben den Vorteil kurzer und sogar annähernd deterministischer Antwortzeiten (Echtzeitfähigkeit). Sie werden mit lokalen Abbildungsfehlern wie Verschiebung, Verzerrung oder leichter Blickwinkeländerung im allgemeinen gut fertig. Kompliziertere Abbildungen, wie beliebige Anordnung mehrerer Objekte in einer Szene, Verdrehung oder partielle Verdeckung - wie in unseren Randbedingungen in 1.3 - wurden in den NN-Anwendungen bisher kaum betrachtet. Auch mangelt es noch an einer konstruktiven, ingenieurmäßigen Methode, um NNe für schwierige Anwendungen erfolgreich zu planen, oder um die erforderlichen Änderungen der Netzstruktur schon bei vergleichsweise geringfügigen Änderungen der Problemstellung vorherzusagen.

Zunächst klären wir einige im folgenden immer wieder vorkommenden Begriffe.

### *Definition 1.1 (Objektyp, Objektinstanz, Szene)*

- Gegeben sei eine endliche Menge  $\mathbf{O} := \{O_i\}_{i \in I}$  von **Objektypen** über einer Indexmenge  $I$  von Bezeichnern.
- Zu jedem  $O_i$  gehört eine räumliche Verteilung von Massepunkten  $Z_i \subseteq \mathbb{R}^3$ , bezogen auf ein beliebiges **objektspezifisches kartesisches Koordinatensystem**  $KO_i$  ( $i \in I$ ).
- Eine Punktmenge  $A_i \subseteq \mathbb{R}^3$  bezüglich eines beliebigen kartesischen Koordinatensystems  $KS$  heißt **Instanz** des Objekttyps  $O_i \Leftrightarrow$  es existieren Translations-, Rotations- und Skalierungsabbildungen  $T, R, D : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ , so daß

$$A_i = T \cdot R \cdot D (Z_i)$$

$T \in \mathbf{T}$ , der Menge der Translationsabbildungen ( $\mathbf{T} \cong \mathbb{R}^3$ )  
 $R \in \mathbf{R}$ , der Menge der Rotationsabbildungen ( $\mathbf{R} \cong$  orthogonale Matrizen  $\in \mathbb{R}^{3 \times 3}$ )  
 $D \in \mathbf{D}$ , der Menge der Skalierungsabbildungen ( $\mathbf{D} \cong$  Diagonalmatrizen  $\in \mathbb{R}^{3 \times 3}$ )

- Eine Anordnung  $\bigcup_{i \in I} A_i$  disjunkter Objektinstanzen  $A_i$  mit  $A_i \cap A_j = \emptyset$ , bezogen auf ein gemeinsames kartesisches Koordinatensystem  $KS$  ("Weltkoordinaten") heißt **Szene**.

**Bemerkungen:**

- Ein Objekttyp wird nur der kompakten Schreibweise wegen als räumliche Punkteverteilung  $Z_i$  beschrieben. Für die Erkennung ist diese Darstellung ungeeignet. In Kapitel 2 wird eine Repräsentation von Objekten durch ihre sichtbaren Oberflächeneigenschaften angegeben.
- Eine Objektinstanz besitzt eine unbekannte Lage relativ zur Punkteverteilung  $Z_i$  ihres Objekttyps, beschrieben durch hintereinandergeschaltete Koordinatentransformationen  $T, R, D$ . Sind echte Formtoleranzen erlaubt ( $D \neq \text{Einheitsmatrix}$ ), so spricht man auch von "generischen" Objekten. In einer Szene können beliebig viele Instanzen desselben Objekttyps enthalten sein.

**Definition 1.2 (Sensorabbild einer Szene)**

Sei  $B$  Menge der Oberflächen  $B \subseteq \mathbb{R}^3$  bezüglich eines kartesischen Koordinatensystems  $KS$ . Jede Oberfläche  $B$  ist darstellbar als Funktion zweier skalarer Variabler  $(u, v)$  über einem Gebiet  $G \subseteq \mathbb{R}^2$ :

$$B := \{ \mathbf{x}(u, v) = (x, y, z)^T \in \mathbb{R}^3 : \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix} \quad (u, v) \in G \}$$

Sei  $S$  die Menge der Szenen bezüglich Koordinatensystem  $KS$  (Def. 1.1).

Ein **optischer Abstandssensor**, auch **geometrischer Sensor** genannt, wird durch eine Abbildung

$$\pi : S \rightarrow B$$

beschrieben, die jeder Szene  $S$  eine sichtbare Oberfläche  $B = \pi(S)$ , ihr **Sensorabbild**, zuordnet.

Die Abbildungsvorschrift  $\pi$  hängt von den geometrischen und physikalischen Eigenschaften des Sensors ab: Projektion, Reflexion an der Materialoberfläche, aber auch Meßfehler wie Rauschen.

**Beispiele zum Meßprinzip, der Dimension und Koordinatendarstellung**

- Bei einem 3D-Laserscanner /29/ stehen  $u$  und  $v$  für zwei unabhängig einstellbare Ablenkwinkel des Laserstrahls, welcher Abstände  $h(u, v)$  mißt.  $u$  und  $v$  legen eine Richtung fest, und  $\mathbf{x}(u, v)$  sind die kartesischen Koordinaten des Oberflächenpunktes im Abstand  $h(u, v)$  vom Sensor.  $B$  beinhaltet eine Transformation von Kugelkoordinaten in kartesische Koordinaten.
- Beim 2D-Laserscanner /29/ existiert nur ein Ablenkwinkel  $u$ . Bei fester Sensororientierung liegen alle Meßpunkte  $(u, h(u))$  in einer Ebene, der Scanebene. Die "fehlende" dritte Dimension wird durch die Bewegung eines Roboters oder Fahrzeugs erzeugt, an dem der Sensor montiert ist. Die Positionen und Orientierungen  $\mathbf{x}_R := (X, Y, Z, A, B, C)[j]^T$  des Roboters werden als Funktion des Bahnparameters bzw. Punktindex  $j$  des Roboterprogramms modelliert. Insgesamt ergibt sich der Oberflächenpunkt  $\mathbf{x}(u, j)$  durch Hintereinanderschaltung
  - einer Transformation der Polarkoordinaten  $(u, h(u))^T$  in kartesische 2D- Sensorkoordinaten,
  - einer weiteren, von  $j$  abhängigen Transformation (Translation, Rotation, Einbettung) zwischen Sensorkoordinaten und Weltkoordinaten.
- Schließlich können auch 1D-Abstandssensoren, die an einem Roboter oder Fahrzeug montiert sind, spezielle Oberflächen aufnehmen.
- Das physikalische Meßprinzip zur Bestimmung der Entfernung  $h$  (Triangulationsprinzip, Laufzeitmessung, Phasendifferenzmessung des modulierten Lichtes, taktil, Ultraschall, Infrarot,...)

spielt für uns generell keine Rolle, solange es sich um geometrische Signale handelt. Nicht unter Def. 1.2 fallen dagegen Grauwertbilder, die für jeden Bildpunkt  $(u,v)$  einen Helligkeitswert  $J(u,v)$ , aber keine kartesischen 3D-Koordinaten  $\underline{x}(u,v)$  liefern.

- Da die **Dimension** von Bildverarbeitungsproblemen in der Literatur nicht einheitlich festgelegt wird, sei für den vorliegenden Bericht folgendes (definitivisch) klargestellt:

Die Dimension richtet sich grundsätzlich nach den Dimensionen des Objekt- und Bildpunktraums als **Vektorraum**, nicht als **topologischer Raum**. Wir betreiben also 3D-Objekterkennung in 3D-Entfernungsbildern, weil das Bild  $B$  nicht in einem 2-dimensionalen **Untervektorraum** (Ebene) des  $\mathbb{R}^3$  enthalten ist. Topologisch gesehen bildet eine Oberfläche  $B$  aber eine "zweidimensionale Mannigfaltigkeit", ist also isomorph einem zweidimensionalen Unterraum des  $\mathbb{R}^3$ .

Die Dimensionalität eines optischen Sensors wird genauso definiert: ein **1D-Abstandssensor** mißt nur Oberflächenpunkte auf einer Scan-Geraden (1D-Raum), ein **2D-Sensor** Punkte innerhalb einer Scan-Ebene, u.s.w.

**2½D-Darstellung** einer Oberfläche  $B$  bedeutet, daß ein kartesisches Koordinatensystem existiert, bezüglich dessen zu jedem Paar  $(x,y)$  höchstens ein  $z$ -Wert mit  $(x,y,z) \in B$  existiert. Solche Oberflächen sind durch eine eindeutige Abbildung  $z=f(x,y)$  darstellbar. Ein einfaches Gegenbeispiel ist eine Kugeloberfläche. Unsere Sensorabbilder der zu erkennenden Objekte und Szenen besitzen im allgemeinen die 2½D-Darstellung **nicht**.

### Definition 1.3 (Szeneninterpretation)

Eine **Szeneninterpretation** ist eine Folge  $SI := ((O_1, T_1, R_1, D_1), \dots, (O_n, T_n, R_n, D_n))$ , die eine kollisionsfreie Szene  $S = s(SI)$  von Objektinstanzen  $A_i := T_i \cdot R_i \cdot D_i$  ( $Z_k(i)$ ) mit  $A_i \cap A_j = \emptyset \quad \forall i \neq j$  benennt. Dies wird durch eine Abbildung beschrieben:

$$s: \quad SI \quad \rightarrow \quad S \quad \quad SI \subseteq \mathcal{A}(O \times T \times R \times D), \text{ den Folgen über } O \times T \times R \times D$$

$$s(SI) := \bigcup_{i=1}^n T_i \cdot R_i \cdot D_i(Z_k(i))$$

Die Lagetransformation  $T_i \cdot R_i \cdot D_i$  wird im folgenden zu  $T_i$  abgekürzt.

Umgekehrt heißt jede Folge  $SI \in s^{-1}(S)$  **zulässige Interpretation** der Szene  $S$ .  $SI$  zählt auf, welche bekannten Objekte unter welcher Lagetransformation in Szene  $S$  vorkommen. Die Interpretation ist oft mehrdeutig, so etwa bei rotationssymmetrischen Objekten.

Als Objekterkennung bezeichnen wir das Auffinden der zulässigen Interpretationen einer Szene  $S$ , von der das Sensorabbild  $\pi(S)$  unter einem Entfernungssensor gegeben ist.

**Er-kennen** kann ein maschinelles System, wie auch der Mensch, streng genommen nur bereits **Bekanntes**. Zu jedem Objekttyp  $O_k$  führen wir eine **Objektrepräsentation**  $OR_k$  ein, die dieses Vorwissen in expliziter Form beschreibt.

Die Wiedererkennung anhand von Repräsentationen bekannter Objekte heißt **modellbasiert**.

### Definition 1.4 (Objektwiedererkennungsproblem)

**Gegeben:** - das Sensorabbild  $B \in \mathcal{B}$  einer Szene

- das **Vorwissen**: Repräsentationen  $\{OR_i\}$  bekannter Typen von Objekten  $O_i$  ( $i \in I$ ), im folgenden auch **Modellobjekte** genannt.

**Gesucht:** die Menge  $s^{-1}(\pi^{-1}(B)) \subseteq SI$  der **zulässigen Interpretationen** der durch  $B$ , oder durch ein maximales Teilbild  $B' \subseteq B$  mit  $s^{-1}(\pi^{-1}(B')) \neq \emptyset$ , abgebildeten Szene, der Teilszene also, die die **bekanntesten Objekte** enthält. Diese sind durch Typ und Lage zu benennen, und mehrdeutige Interpretationen sind anzugeben.

Der duale Vorgang, um dieses Vorwissen zu erwerben, heißt Objektrekonstruktion. Alltagssprachlich bedeutet "rekonstruieren", einen Gegenstand aus seinen verschiedenen Erscheinungsformen (Facetten) zu einer Ganzheit zusammenzufügen. Da die Objekterkennung anhand von Sensorabbildern erfolgt, liegt es nahe, die Objektrepräsentationen aus Sensorabbildern unter verschiedenen Blickwinkeln, den Ansichten, abzuleiten:

$$OR_i = f (\{\pi (Tr_j(O_i))\}_{j \in J})$$

was nicht bedeutet, daß  $OR_i$  identisch mit der Menge dieser Ansichten oder einer endlichen Teilmenge davon ist. Eine stärker verdichtete Repräsentation ohne explizite Angabe des Blickwinkels, also einer bestimmten Lagetransformation, ist erwünscht. Anschaulich kann man sich darunter die automatische Erstellung eines Geometric-(CAD-)Modells, das markante Attribute für die Erkennung enthält, aus Sensordaten vorstellen.

### **Definition 1.5 (Objektrekonstruktionsproblem)**

- Gegeben:**
- ein Objekttyp  $O_i$
  - Sensorabbilder  $\{\pi (Tr_j(O_i))\}_{j \in J}$ ,  $Tr_j$  geeignete Lagetransformationen (Teilansichten)
- Gesucht:**
- die Objektrepräsentation  $OR_i$ , die  $O_i$  vollständig und redundanzfrei für die Wiedererkennung in beliebigen Lagen kennzeichnet.

Objektwiedererkennung und -rekonstruktion stehen komplementär zueinander: das bei der Erkennung benutzte Vorwissen wird bei der Rekonstruktion erzeugt. Wegen ihrer verschiedenen Ein- und Ausgabe lassen sie sich nicht sinnvoll in einer Problemhierarchie (Verallgemeinerungs-/Spezialisierungs- oder Schwierigkeitshierarchie) anordnen, wie zum Teil in der Literatur /63/ üblich. Wohl haben beide viele Teilprobleme gemeinsam, z.B. das Erzeugen einer Repräsentation durch Vorverarbeitung der Entfernungsdaten, oder das Zusammensetzen einer Gesamtrepräsentation für eine Szene oder ein Objekt aus Teilansichten.

Folgende bekannten Probleme bei autonomen Fahrzeugen sind Spezialfälle der Grundprobleme Rekonstruktion und Wiedererkennung.

### **Hinderniserkennung ( $\subseteq$ Wiedererkennung)**

Die Umgebung eines Fahrzeugs ist zu klassifizieren nach: freiem Fahrweg, Hindernissen, und Hintergrund. Die Hindernis-Objekte sind durch einfache globale Merkmale wie Breite, Höhe charakterisiert.

Meist ist keine Erkennung der Gestalt oder der Orientierung des Hindernisses notwendig.

### **Griff in die Kiste ( $\subseteq$ Wiedererkennung)**

Das oberste unter mehreren gleichartigen Werkstücken in einem Behälter ist zu erkennen, um es zu greifen.

Der Objekttyp liegt oft a priori fest; Position/Orientierung sind aber in der Regel zu bestimmen, um eine stabile Greiferposition zu finden.

### **Kartierung, Exploration ( $\subseteq$ Rekonstruktion)**

Ein größerer, zusammenhängender Bereich (Werkshalle, Bergwerkstollen o.ä.) ist abzufahren, um von verschiedenen Positionen aus Abstandsbilder aufzunehmen und zu einem Geometriemodell, der Landkarte, zu fügen.

Oft genügt es, waagerechte 2D-Profile der Umgebung in einer gegebenen Höhe zu vermessen. Da die Lage der Meßpositionen mit den odometrischen Fahrzeugsensoren nur ungenau meßbar ist, stellt das Fusionsproblem, die überlappenden Teilbilder in einem gemeinsamen Weltkoordinatensystem zur Deckung zu bringen, ziemlich harte Anforderungen.

## Ortsbestimmung ( $\subseteq$ Wiedererkennung)

Die unbekannte oder nur grob bekannte Fahrzeugposition ist anhand charakteristischer, aus den Entfernungsdaten ableitbarer Merkmale, z.B. Landmarken, durch Vergleich mit der Landkarte präzise zu bestimmen.

Die gesamte Fahrzeugumgebung wird dabei als ein in sich starres Komplexobjekt gesehen und anhand charakteristischer Merkmale, der Landmarken, wiedererkannt, um die unbekannte Lage des Fahrzeugs (Sensors) relativ zu diesem Objekt zu bestimmen. Bei der Werkstückerkennung verhält es sich genau umgekehrt: hier ist die Lage des Sensors gegeben und die des erkannten Werkstückes zu bestimmen.

Die Werkstückrekonstruktion verhält sich zur Werkstückerkennung also wie die Kartierung zur Ortsbestimmung bei autonomen Fahrzeugen. Während autonome Fahrzeuge für die Navigation meist nur 2D-Modelle erfordern und die Lage der wiederzuerkennenden Merkmale schon grob bekannt ist, erfordert das Greifen von Objekten in unbekannter Lage meist 3D-Modelle.

Eine Kombination oder Iteration der Grundbausteine Rekonstruktion und Erkennung ist in komplexen Wahrnehmungsaufgaben natürlich denkbar. Z.B. könnte ein System jede neue Szene mit den bereits bekannten Objekten vergleichen, also wiedererkennen, und dabei die Bildteile isolieren, die keinem bekannten Objekt entsprechen. Diese Gegenstände bzw. Bereiche werden im nächsten Schritt gezielt, vielleicht von mehreren Seiten vermessen und als neue, zukünftig bekannte Objekte rekonstruiert. Dieser Zyklus wiederholt sich, bis keine unbekannt Objekte mehr auftreten.

## 1.2 Anwendungen

### 1.2.1 Teststand zur Entfernungsdatenaufnahme

Die ersten Erfahrungen mit Entfernungssensoren wurden im COMETOS-Projekt mit einem nach dem Triangulationsprinzip arbeitenden Laserscanner der Firma Oldelft (60x60 mm Fenster) gewonnen. Damit wurde die Oberfläche von Gußwerkstücken vermessen, um die Bearbeitungsstellen (Gußgrate) zu erkennen und aus der Menge der Bearbeitungsstellen einen Werkstückumriß zu rekonstruieren. Wegen der geringen Reichweite des an der Roboterhand montierten Sensors war es nötig, diesen sehr dicht am Werkstück entlangzuführen.

Für zukünftige Arbeiten steht ein Abstandssensor größerer Reichweite (0.5-10m) zur Verfügung /29/, welcher die Laufzeit des Laserlichtes mißt und von einem festen Punkt aus Höhenprofile durch Ablenken des Laserstrahls in der Abtastebene mittels eines Schwingspiegels liefert (2D LADAR, Laser Radar Prinzip). Die wichtigsten Parameter sind in Tabelle 1.1 aufgelistet.

Reichweite der Entfernungsmessung	0,5-10m
max. Öffnungswinkel des Blickfeldes	220°
Genauigkeit der Entfernungsmessung	$\pm 20\text{mm}$ ( $\sigma$ ) bei 1-2m
Winkelauflösung in Bit	14
min. Winkelschrittweite	ca. 0,6°
Entfernungsauflösung in Bit	16
Rate der 2D-Profile (Spiegelfrequenz)	8 Hz
Rate der Entfernungsmessungen	ca. 4600 Hz

**Tabelle 1.1:** Die wichtigsten Parameter des 2D LADAR

Um die 3. Dimension zu erhalten, muß der Sensor mit einem Roboter verfahren werden. Aus Platzgründen war es nicht möglich, den Sensor direkt an der Roboterhand zu montieren und so die volle Orientierung zu nutzen. Stattdessen ist der Sensor an der Roboter-Z-Achse ca. 1.5m oberhalb der Hand montiert. Die Standardorientierung der Scanebene ist parallel zur  $y_w/z_w$ -Ebene des Weltkoordinatensystems, wobei die Szene senkrecht unterhalb des Sensors aufgenommen wird (Abb. 1.1). Das Blickfeld des Sensors läßt sich aber in  $5^\circ$ -Schritten um die  $x_w$ -Achse (Winkel  $\Delta\theta$ ) drehen und um die  $y_w$ -Achse (Winkel  $\Delta\phi$ ) neigen. Drehung und Neigung der Scanebene sind aber zur Zeit nur mechanisch möglich, und die Winkel sind nicht rechnerseitig lesbar.

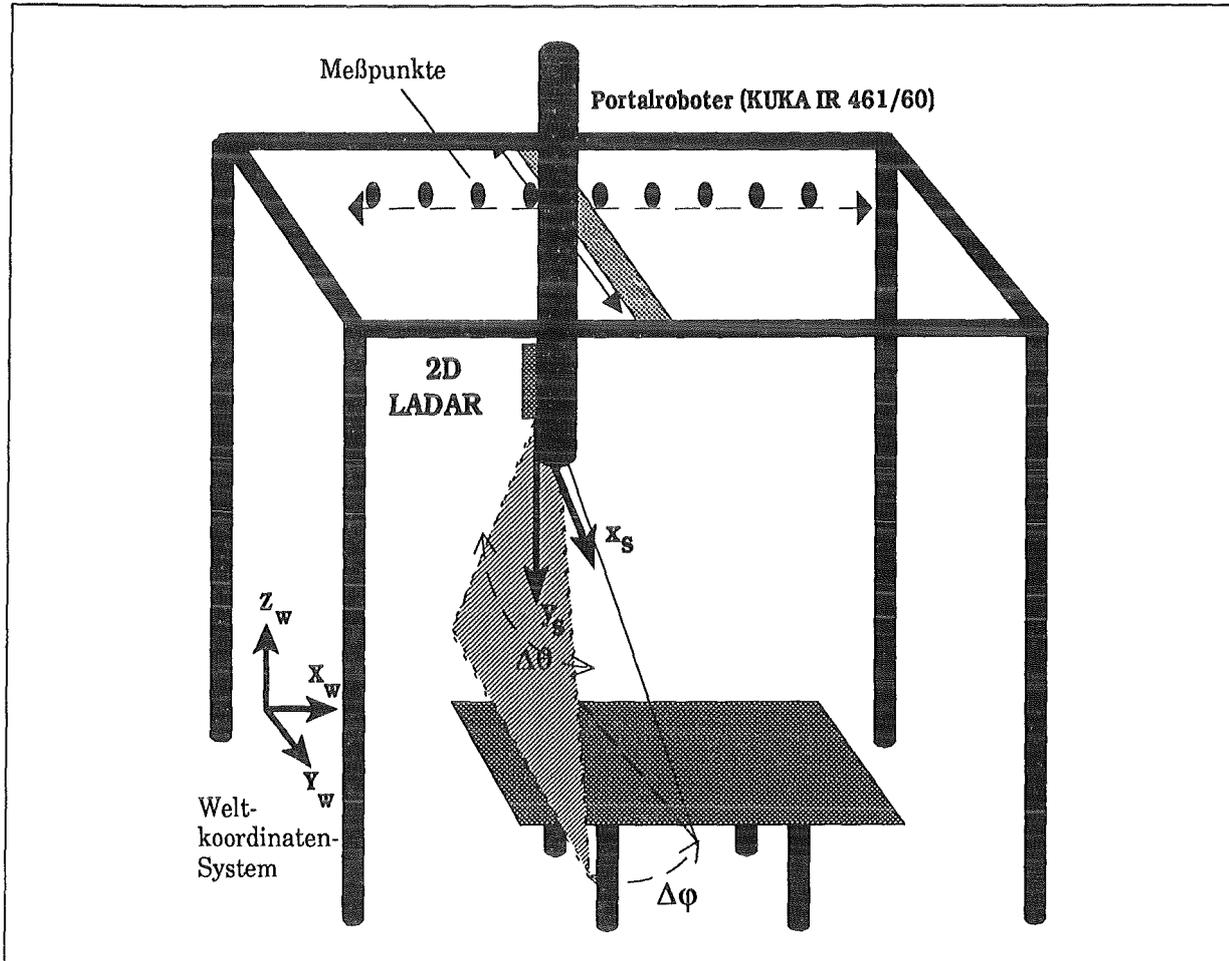


Abb. 1.1 2D LADAR - Teststand

Um eine komplette Szene zu vermessen, wird ein Meßprogramm für den Meßroboter erzeugt, entweder konventionell mit dem Programmierhandgerät der Robotersteuerung (r500 der Fa. AEG, Programmiersprache DOROB), oder durch direktes Führen des Meßroboters mit Hilfe der COMETOS Bedieneroberfläche /34/. Roboterfahrt und Höhenprofilmessung wechseln im 'stop and go'-Verkehr ab. Der Roboter wartet an jedem Meßpunkt, bis der LADAR ein oder mehrere gemittelte 2D-Höhenprofile erfaßt hat, und fährt dann zum nächsten Meßpunkt. So wird eine einfache und fehlerfreie Zuordnung der Sensordaten und Roboterkoordinaten gewährleistet.

Zur Steuerung der Entfernungsdaterfassung wird ein IBM PC (AT386) mit einem Transputer-Motherboard MB1T9S /48/ (kompatibel zu INMOS B008) als Einsteckkarte eingesetzt. Auf dem Motherboard sind ein 32 Bit Root-Transputer (T805) und ein 16 Bit Transputer (T222) fest installiert. Sie sind mit dem Host-PC, den Winkel- und Abstandskanälen des LADAR sowie mit der DNC-Schnittstelle (RS232) der Robotersteuerung über Transputer-Links und Schnittstellenumsetzer gekoppelt (Abb. 1.2).

Die SW stellt eine Erweiterung des Spiegelscanner-Programmsystems spisca der Firma IBEO zur Erfassung von Datenbeständen mit 2D und 3D Laserscannern dar /30/. Mit der Erweiterung können komplette Roboterprogramme abgefahren und die anfallenden Sensor- und Roboterkoordinaten zu dreidimensionalen kartesischen Koordinatenbeständen verknüpft werden. Die Aufgaben sind folgendermaßen verteilt:

Host-PC (Hauptprogramm spisca10.exe, Programmiersprache C, Betriebssystem MS-DOS):

- Menugesteuerte Oberfläche zur Realisierung der einzelnen Betriebsarten (2D-Horizontalscan bzw. Dauerscan, 3D-Scan, Roboterprogramm-Scan)
- Parametrierung des Sensors
- Visualisierung der Sensordaten
- Abspeichern der Sensordaten als kartesische Koordinatensätze
- "Serverfunktion" für das Transputersystem, d.h. Unterstützung des Kommunikationsprotokolls zwischen PC und Transputer, Laden und Booten des Transputer-Programms

[NB: die Bezeichnung "Server" ist mißverständlich, PC und Transputer kommunizieren zwar nach dem client-server-Modell, dabei fungiert aber der PC als **Klient** und fordert Daten und Funktionen an, die der Transputer als **Server** bereitstellt. Die Bezeichnung rührt daher, daß das PC-Hauptprogramm auch den Standard-PC-Server des Toolset-Entwicklungssystems für OCCAM 2 /33/, den ISERVER, ersetzt.]

Transputer T805 (Prozeß spisca.occ, Programmiersprache OCCAM 2)

- Kommunikation mit dem Host-PC (Empfangen von Funktions- und Datenanforderungen, Senden von Koordinaten, vgl. dazu die Software-Dokumentation /11//30/)
- Erfassung der Winkel und Entfernungswerte des 2D LADAR
- Kommunikation mit dem Meßroboter über zwei Schnittstellendienste (Roboterkoordinaten erfragen, Fahren zum nächsten Meßpunkt)
- Mittelung redundanter LADAR-Abstandsmessungen; Umrechnung der Sensordaten (Polarkoordinaten) und der Roboterkoordinaten in kartesische Koordinaten

Transputer T222 (Prozeß dnc.occ, Programmiersprache OCCAM 2)

- Implementierung der Schnittstellendienste zum Meßroboter mit Hilfe von DNC-Telegrammen der Robotersteuerung r500 (Anfordern und Empfangen der Roboterkoordinaten, Synchronisation mit dem Meßroboter im Stop-and-Go-Verkehr)
- Implementierung des Kommunikationsprotokolls der DNC-Schnittstelle der r500

Die kartesischen Datenbestände liegen in ASCII-Format zur Weiterverarbeitung auf beliebigen Wirtsrechnern vor, z.B. mit dem punkt-orientierten Modellierungssystem POMOS auf der Silicon Graphics PC IRIS /44/. In Abb. 1.3 ist eine einfache 3D-Szene mit 2 Schachteln auf einem Tisch unter POMOS als Gitternetz perspektivisch dargestellt. Der Meßroboter fährt in diesem Beispiel eine gerade Linie parallel zur  $x_w$ -Achse, die Sensororientierung ist konstant.

Man erkennt deutlich die Abschattung im linken Bildbereich, wo der Tisch nahtlos in die Paletten-Spannvorrichtung von COMETOS überzugehen scheint. Solche nur scheinbaren, durch die Abschattung entstehenden und parallel zur Blickrichtung des Sensors liegenden Körperflächen werden als **Phantomflächen** bezeichnet. Sie entstehen keineswegs nur durch die SW-Interpolation der Meßpunkte bei der Erzeugung des Gitternetzes, sondern sind bereits in den Sensorrohdaten enthalten. Wenn der Laserlichtkegel mit seinem nicht vernachlässigbaren Öffnungswinkel (**Footprint**,  $0.14^\circ$ ) eine Objektkante trifft, liefert die Mittelung der reflektierten Lichtanteile zufällige Zwischenwerte der Entfernung ('**mixed-point-Effekt**' /40/).

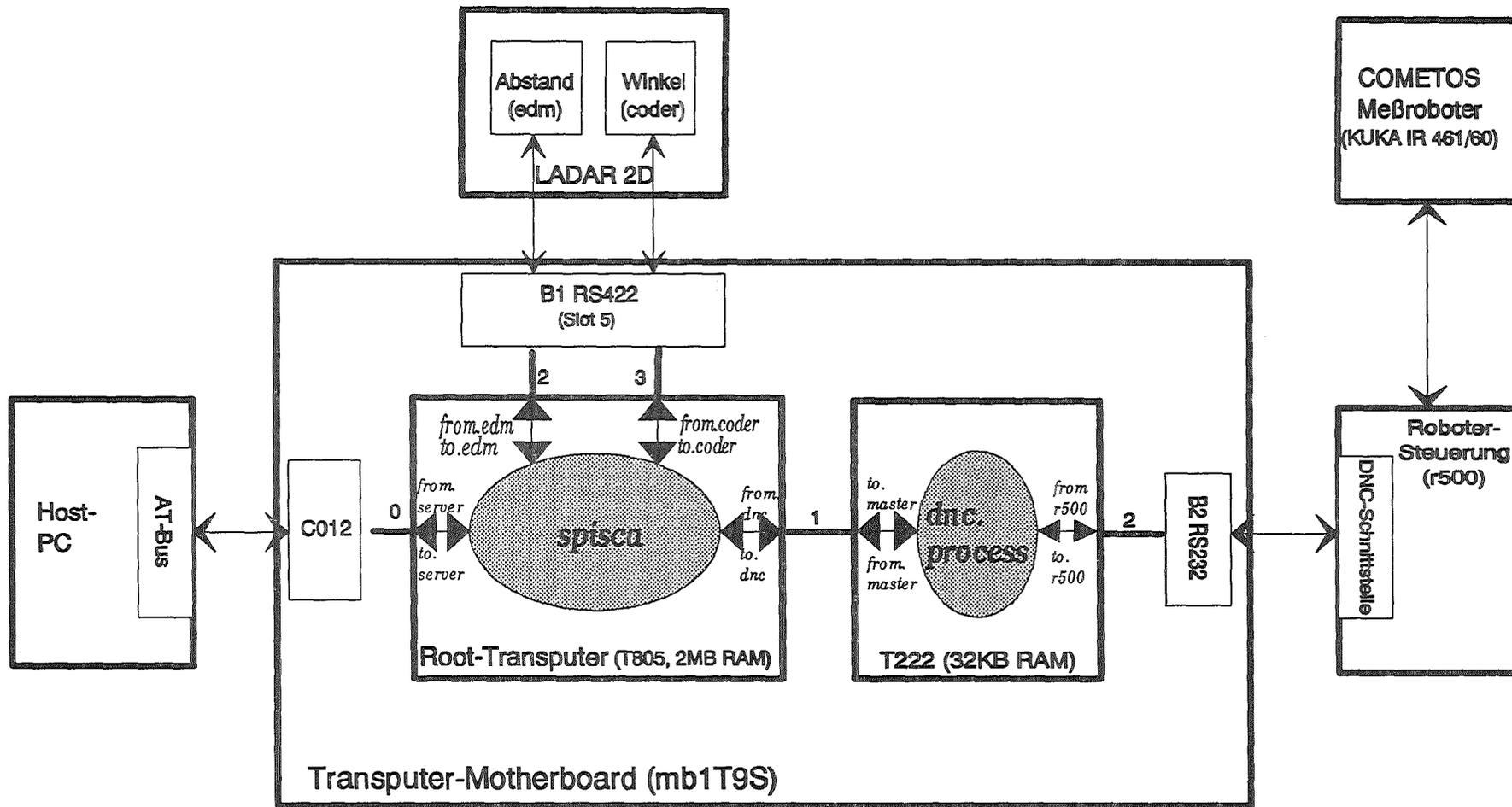


Abb. 1.2 Konfiguration des Transputersystems für den 2D LADAR Teststand

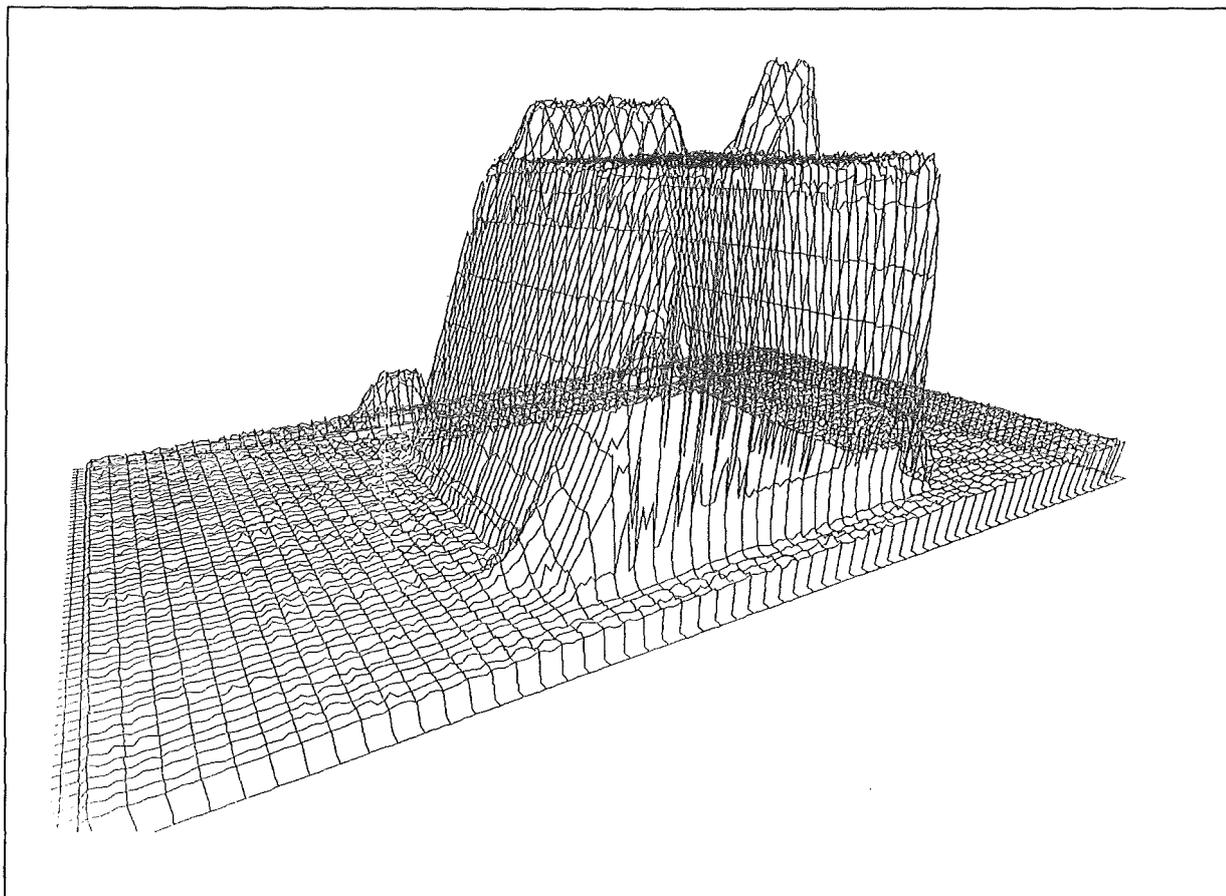


Abb. 1.3 LADAR-Szene (2 Schachteln auf Tisch) als Gitternetz

Zu den Meßfehlern in den Sensordaten tragen auch die Fehler der Laufzeitmessung des Laserlichtes und andere, in ihren Ursachen noch ungeklärte impulsartige Störungen bei. Die Standardabweichung der Entfernungswerte  $\sigma$  liegt bei  $\pm 20\text{mm}$ , doch treten immer wieder Impulse bis  $100\text{mm}$  auf. Die Fehler nehmen mit der Entfernung zu.

Abb. 1.4a zeigt die Daten einer zweiten Szene ("Kiste und Tonne"), Abb. 1.4b die durch einen  $5 \times 5$ -Medianfilter gefilterten Daten.

Die Winkelschrittweite  $\Delta w$  des 2D LADAR, und damit die eigentliche Horizontalauflösung der Messung ist durch

$$\Delta w \geq \omega_{\min} \Delta t_{\min}$$

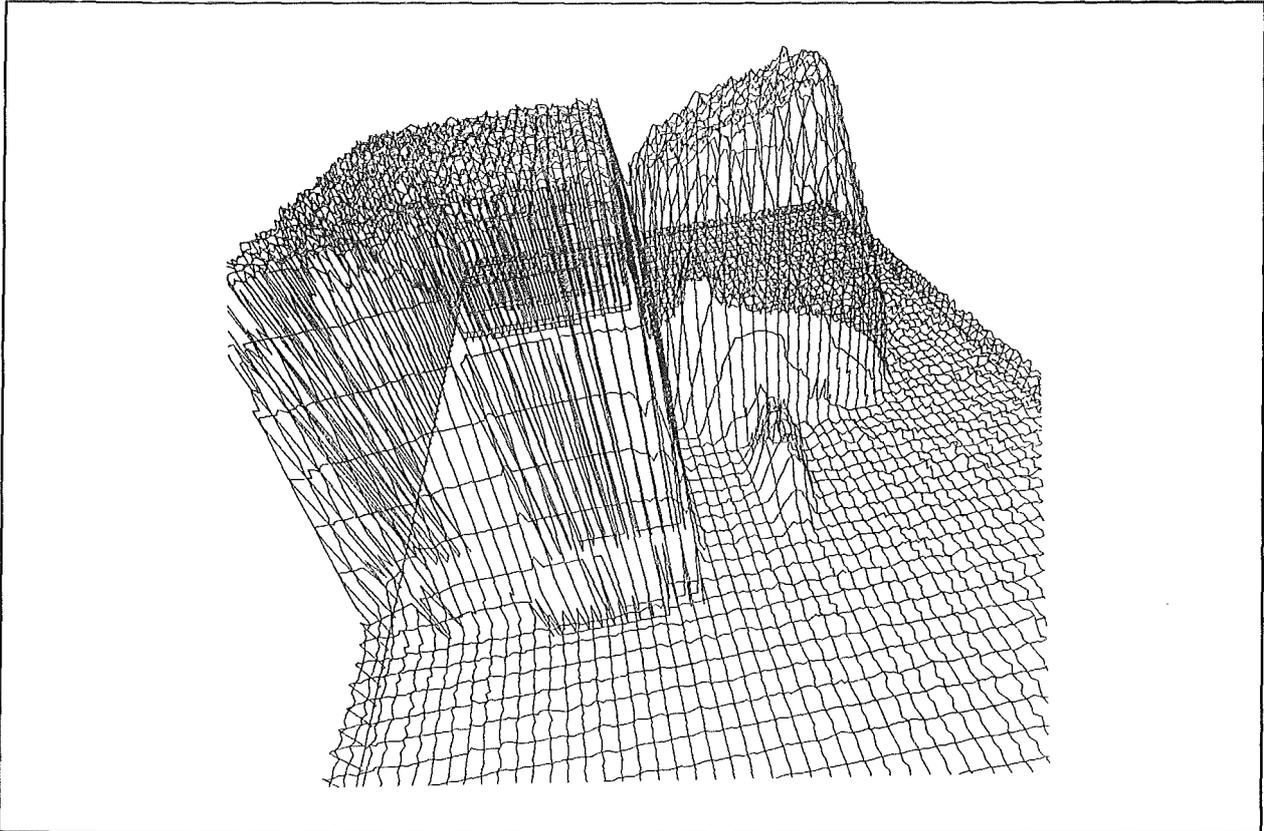
$\omega_{\min}$ : kleinste zulässige Winkelgeschwindigkeit des Schwingspiegels

$\Delta t_{\min}$ : Meßzeit für einen Abstandswert (ca.  $200\mu\text{s}$ )

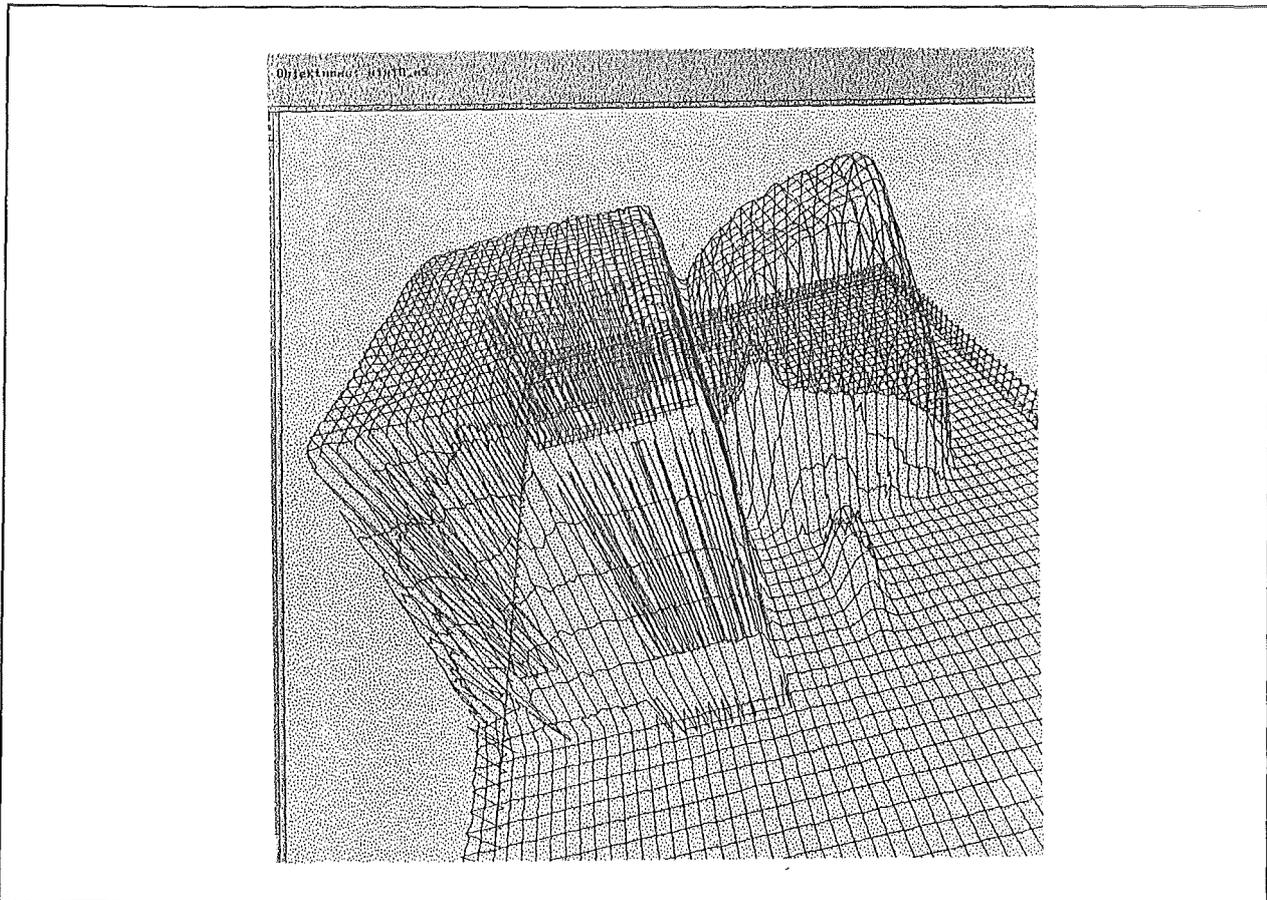
nach unten auf ca.  $0,6^\circ$  begrenzt. Dies entspricht einem Abstand der Meßpunkte von ca.  $3\text{cm}$  bei  $2\text{m}$  Entfernung zum Meßobjekt und läßt nur die Erkennung grobstrukturierter Objekte zu ( $0,5$  bis  $2\text{m}$   $\varnothing$ ).

Eine Fehlerquelle der Messung liegt darin, daß Abstandswerte und Winkel nicht exakt zeitgleich sind: während die Winkelwerte rein rechnerisch im äquidistanten Raster inkrementiert werden, werden die Abstände in jedem Meßzyklus neu angefordert, und zwar im Mittel in dem entsprechenden Zeitraster von ca.  $200\mu\text{s}$ , aber mit geringen stochastischen Zeitschwankungen.

Die einzelnen Fehlerquellen der Entfernungsmessung sowie Maßnahmen zur Rauschunterdrückung werden in der Diplomarbeit /40/ ausführlich diskutiert.



**Abb. 1.4a** LADAR-Szene (Kiste und Tonne), als Gitternetz



**Abb. 1.4b** Szendaten gefiltert durch 5x5 - Medianfilter

Mit dem Versuchsaufbau lassen sich aber schon heute Algorithmen zur Objektrekonstruktion und -erkennung an realen Sensordaten testen. Zukünftige Arbeiten im Bereich der eigentlichen Datenerfassung zielen auf

- Verbesserung der Winkelauflösung des 2D LADAR und der Korrelation der Winkel- und Abstandswerte
- Berücksichtigung des Intensitätssignals, das der 2D LADAR neben dem Entfernungssignal liefert; quantitative Analyse der Meßfehler in Abhängigkeit von Oberflächeneigenschaften
- Erweiterungen der Kommunikation mit der Robotersteuerung (Programmstart etc.)
- Verbesserung und Erweiterung der Oberfläche der Tranputer-SW, bei Bedarf eigenständige Visualisierungskomponente
- Integration weiterer Abstandssensoren, z.B. 3D LADAR
- Verbesserung und Erweiterung der Positioniereinheit des Sensors (Handmontage), falls notwendig.

## 1.2.2 Längerfristige Anwendungen

Die Objektrekonstruktion und -wiedererkennung mit Entfernungssensoren besitzt eine Reihe wichtiger Anwendungen im Arbeitsprogramm des KfK.

- a) Die Vermessung von Werkstücken zur nachträglichen, automatischen oder halbautomatischen, Erstellung von CAD-Modellen.

Diese zur Zeit unter dem graphischen Punktmodellierungssystem POMOS laufenden und dem Bereich Rekonstruktion zuzuordnenden Arbeiten stellen eine Weiterentwicklung der **Koordinatenmeßtechnik** dar. Mit heutigen Präzisions-Koordinatenmeßmaschinen auf der Basis von taktilen Sensoren oder Triangulations-Laserscannern erreicht man Entfernungsaufösungen von ca. 50-100 $\mu$ m.

Eine noch feinere Auflösung (1-10 $\mu$ m) erfordert die **3D-Modellierung von Mikrostrukturen**, z.B. mikromechanischer Elemente oder beschichteter Liga-Strukturen.

Optische Meßsysteme, die eine solche Auflösung und entsprechende Genauigkeit erreichen (z.B. Bildebenen-Lokalisierungssysteme, holografische oder interferometrische Verfahren) existieren vielfach erst als Labormuster, werden aber bald kommerziell verfügbar sein /60/. Da sie die räumliche Tiefeninformation mehr oder weniger direkt liefern, könnten sie die Intensitätsbildverarbeitung mit Rasterelektronenmikroskopen sinnvoll ergänzen.

- b) Benutzung von CAD-Werkstückmodellen zur off-line oder on-line-Programmierung von **Handhabungsaufgaben in wenig geordneten Fertigungsumgebungen**, beim Transport, der Bestückung, Bearbeitung oder Montage.

Ein konkretes Beispiel ist die automatische Erkennung und Beladung von Containern in Häfen mittels schienen- oder radgestützter Portalkräne. Diese Aufgabenstellung wird an anderer Stelle beschrieben /31//39/.

- c) Unterstützung (teil)autonomer Fahrzeuge bei **Inspektions- und Handhabungsaufgaben in unzugänglichen Bereichen** /52//61/.

Neben der unmittelbaren Aufgabe, ein Fahrzeug sicher durch eine solche Umgebung zu steuern (Hinderniserkennung, Kartierung, Ortsbestimmung, die nach Abschnitt 1.1 auch mit den Verfahren zur Rekonstruktion und Wiedererkennung gelöst werden können), ist hier vor allem an punktuelle Arbeiten am Einsatzort gedacht, wie Ablesen von Ventil- und Schalterstellungen und analoger Anzeigeinstrumente, Betätigen von Hebeln, Schiebern, Schützen, das Hantieren mit Flanschen und Rohrleitungen u.a.

Solche Arbeiten fallen bei der Reparatur oder Demontage kerntechnischer Anlagen an, sind also der **Energietechnik** zuzuordnen. Zur Handhabung wird räumliche Tiefeninformation benötigt, wobei für optimale Beleuchtungsverhältnisse oft nicht gesorgt werden kann. Unter diesen Randbedingungen stellen Entfernungssensoren eine wertvolle Alternative und Ergänzung zu Videokameras dar.

Weitere denkbare Anwendungen liegen in der **Umweltechnik**, z.B. der Inspektion und Sanierung von Abwasserrohren mit Robotern (Kanalsanierung). Die Inspektion der Schäden (Risse) erfolgt bisher allerdings allein mit Grauwertbildern, meist durch den Bediener, und es bleibt zu prüfen, ob Entfernungsbilder Vorteile bringen können.

Eine weitere Anwendung mit Zukunft könnte die robotergestützte Erkennung und Sortierung von Abfallprodukten zur stofflichen Wiederverwertung (Recycling) sein. Diese schmutzige, belastende und sehr zeitraubende Arbeit erfolgt heute manuell.

Selbstverständlich stellt jede dieser Anwendungen ihre speziellen Anforderungen bzgl. Reichweite, Auflösung, Genauigkeit, Datenaufnahmeverfahren, und Einsatzumgebung der Sensoren. Auch erfordert jedes Anwendungssystem eine individuelle ingenieurtechnische Gesamtlösung.

Ein allen Anwendungen gemeinsames Bindeglied ist die Informationstechnik: die Methoden und Verfahren zur Datenaufbereitung und -filterung, Merkmalsextraktion, Segmentierung, Fusion von Teilansichten, Objektrepräsentation, und die Wiedererkennungsverfahren. Ausreichend robuste, flexible, effiziente, insgesamt praxistaugliche Verfahren fehlen gegenwärtig. Dies bestätigen insbesondere die im Bereich Autonome Roboter und Entfernungssensoren tätigen Firmen. Jedes neue, leistungsfähige Verfahren kann in vielen Anwendungen eingesetzt werden und besitzt damit auch ein großes Marktpotential.

### 1.3 Randbedingungen

Ein System zur Objekterkennung aus Entfernungsbildern läßt sich nach folgenden Kriterien klassifizieren:

1. der **Einsatzumgebung**
2. der Beschaffenheit der **Objekte** (Form, Größe, Lage, Anzahl,...)
3. der **Sensordatenaufnahme** bzw. den **Eingangsdaten**
4. den **Leistungsanforderungen** bzw. **Ausgangsdaten**.

#### 1.3.1 Einsatzumgebung

Hier kann man unterscheiden (vgl. /64/)

- |    |                     |  |
|----|---------------------|--|
| a) | <i>natürliche</i>   | Umgebung (freies Gelände, Straßen, Bäume,...)                  |
|    | <i>künstliche</i>   | Umgebung (Fabrik, Lagerhalle, Hafen, Bahnhof, Kanalsystem,...) |
| b) | <i>vorgefundene</i> | Umgebung   |
|    | <i>optimierte</i>   | Umgebung   |

Nur bei einer in ihren wesentlichen Teilen ingenieurmäßig konstruierten, künstlichen Welt hat der modellbasierte Ansatz eine Chance. Das nötige Vorwissen über die Umwelt kann hier effektiv erworben und auf ähnliche, reproduzierbar wiederkehrende Objekte bzw. Situationen angewandt werden. Die wichtigsten Objekte sind mit relativ wenigen Merkmalen unterscheidbar. Dies gilt bei der Vielfalt natürlicher Formen i.a. nicht. Dies ist methodisch gesehen der wesentliche Unterschied zwischen natürlicher und künstlicher Umgebung.

Bei den meisten Anwendungen in 1.2.2 handelt es sich um künstliche Umgebungen.

Natürliche Umgebungen spielen bei autonomen Fahrzeugen im Straßenverkehr oder bei militärischen Anwendungen eine Rolle. Anstelle einer vollständigen Gestalterkennung reicht oft eine grobe Klassifizierung der Bildobjekte als Fahrweg, Hindernis, Szenenhintergrund oder Operationsziel aus. Weit schwieriger wird die Aufgabe dann, wenn ein autonomes Fahrzeug sich an natürlichen Landmarken im freien Gelände orientieren soll.

Optimierte Umgebung unter b) bedeutet, daß spezielle Bedürfnisse oder Annahmen des Bildverarbeitungssystems oder autonomen Fahrzeugs, z.B. die Gestaltung des Arbeitsraums, die Beleuchtung, die Form und Anordnung der Objekte (vgl. 1.3.2) betreffend, erfüllt sein müssen. Ein solches System kann also nicht ohne weiteres in eine existierende Umgebung nachträglich eingebracht werden, anders als eines, das mit einer vorgefundenen Umgebung zurechtkommt.

Strenggenommen widerspricht eine optimierte Umgebung dem Anspruch auf Autonomie. Ein System mit solch harten Anforderungen an seine Umwelt setzt sich leicht dem Vorwurf aus, mit Kanonen auf Spatzen zu schießen. Wenn nur ein einziges Werkstück in wenigen Vorzugslagen erwartet wird, wird man versuchen, die Werkstücke durch einfache organisatorische und mechanische Maßnahmen so auszurichten, daß ein Toleranzausgleich ganz entfällt oder so einfach wird, daß ein konventionelles und damit billigeres Automatisierungssystem einsetzbar wird.

So haben z.B. Objekterkennungssysteme für den "Griff in die Kiste", so wichtig sie zur Demonstration grundlegender Prinzipien des Bildverstehens waren, bis heute kaum Eingang in die Fertigungspraxis gefunden.

Zusammenfassend ist im vorliegenden Projekt von **künstlichen, aber vorgefundenen** Umgebungen auszugehen. Diese Randbedingungen werden in der Literatur zum Teil mit dem unscharfen Begriff "halb-strukturierte" oder "quasi-strukturierte" Umgebung umrissen.

### 1.3.2 Objekte

#### a) Form

Viele Objekterkennungssysteme machen starke Einschränkungen der folgenden Art:

- **Blockwelten** (primitive Grundkörper wie Quader, Zylinder, Kugel, Kegel) /7/ /51/
- bekanntes **Konstruktionsgesetz**: jedes Objekt ist explizit als Vereinigung oder Differenz einfacher Grundkörper wie Quader, Zylinder, Kegel oder Kugel gegeben (Beispiele: CSG-Prinzip oder sweep-Repräsentationen) /8/ /36/
- **ebene** Oberflächen (polyedrische Körper) /5/ /16/ /20/ /32/
- **konvexe** Oberflächen /27/ /28/.

Wir lassen hingegen auch nichtkonvexe, gekrümmte Körper mit unbekanntem "Formungsgesetz" zu. Richtschnur könnte etwa das VDAFS-Freifformflächenmodell sein. Rechnerisch sind aber Oberflächenapproximationen niedriger Ordnung, z.B. biquadratische Polynome oder Quadriken, notwendig. Sollten weitere Einschränkungen notwendig werden, ist es wichtig, sich klarzumachen:

- Wofür werden sie konkret benötigt (sind sie unerlässlich für das Funktionieren einer Methode, oder vereinfachen sie nur gewisse Rechnungen oder den Funktionsnachweis)?
- Wie stark leidet darunter der praktische Nutzen?
- Zu welchem Preis, z.B. Algorithmenkomplexität, lassen sie sich aufheben?

#### b) Lage, Anzahl

Mehrere Instanzen beliebiger Objekttypen sollen in beliebiger Lage in einer Szene vorkommen dürfen. Objekte können Teile von sich selbst oder sich gegenseitig teilweise verdecken oder bündig aneinander oder aufeinander liegen.

Keine bestimmte Beschaffenheit des Bildhintergrundes (z.B. eine ebene Arbeitsfläche oder Fußboden) soll a priori vorausgesetzt werden.

Klarerweise wird in der Praxis die Zahl, Größe und Formkomplexität der Objekte durch die Reichweite, die Genauigkeit und das Auflösungsvermögen des Sensors und die für die Bildauswertung verfügbare Zeit und Rechenleistung begrenzt. Verdeckung und symmetriebedingte Mehrdeutigkeiten erschweren auch dem Menschen die Objekt- und Lageerkennung, wenn ihm zusätzliche Unterscheidungsmerkmale wie Farbe oder Beschriftung fehlen.

Dies alles sind aber durch die Praxis sich erst ergebende, zu **evaluierende Leistungsgrenzen** eines Systems, und gedanklich zu unterscheiden von **a priori Annahmen** und Einschränkungen, die zum festen Bestandteil des Bilderkennungssystem werden.

### c) Oberflächenbeschaffenheit

Entfernungssensoren wie Triangulations-Laser oder Laser-Radar verwenden das Prinzip der **diffusen Lichtreflexion**. Vollständig lichtabsorbierende, spiegelnde oder lichtdurchlässige Oberflächen führen zu Fehlmessungen. Nun lassen sich solche Einflüsse in einer vorgefundenen industriellen Umgebung nicht völlig ausschließen. Deshalb soll in Vorversuchen der Einfluß der Oberflächenbeschaffenheit bei verschiedenen Auftreffwinkeln auf die Güte der Meßdaten experimentell untersucht werden, z.B. mit

- metallischen Oberflächen (rauh/poliert)
- Holz, Pappe
- Kunststoffe
- Glas, Plexiglas
- hellen/dunklen Farben.

Von Bedeutung sind rechnerisch einfach auswertbare Kriterien (z.B. statistische Kenngrößen wie Varianz und Verteilung der Meßfehler, Rauschmaße, Ergebnisse der Intensitätsmessung), anhand derer "pathologische" Umgebungsbedingungen bereits bei der Datenvorverarbeitung zuverlässig erkannt und notfalls ganze Bildbereiche markiert und ausgesondert werden können.

### d) Generische Objekte

Typisches Beispiel für ein generisches Objekt ist die Menge aller Brief- oder Paketsendungen der Bundespost oder eines Versandhauses. Sie alle sind Instanzen des generischen Objekttyps "Quader" mit variabler Länge, Breite oder Höhe /62/.

Prinzipiell bereitet die **Wiedererkennung** und Vermessung generischer Objekte mit invarianten Beschränkungen (Kap. 2) keine besonderen Schwierigkeiten; es fallen nur die Größenrelationen und -prädikate weg, während Formbeschränkungen erhalten bleiben. Ungleich schwieriger ist ihre (automatische) **Rekonstruktion**, da das System von der koordinatenorientierten Darstellung der Rohdaten zu einer koordinatenfreien Objektrepräsentation selbständig **abstrahieren** müßte, wie beim Lernen von Konzepten aus Beispielen.

Die Vorteile generischer Objekte sind die kompakte Darstellung und unbegrenzte Maßabweichungen. Bei vielen Anwendungen lassen sich generische Objekte durch eine diskrete, überschaubare Menge von Einzelobjekten mit genormten Abmessungen ersetzen. Gewisse Toleranzen müssen bei der Objektwiedererkennung natürlich immer, schon wegen der Meßfehler zugelassen werden, doch sind diese **beschränkt**.

Insgesamt sehen wir die Behandlung generischer Objekte nicht als vorrangiges Ziel an.

### e) Statische/starre/dynamische Objekte

Dynamische Objekte, z.B. im Sensorblickfeld sich bewegende Personen, Fahrzeuge oder Roboter, verändern ihre Lage und i.a. auch ihre Form mit der Zeit. Statische Objekte ändern nicht ihre Lage, starre Objekte nicht ihre Form. Dynamische Objekte, die in einem Zeitintervall kontinuierlich verfolgt werden sollen, erfordern Methoden der Bildfolgenanalyse (z.B. optischer Fluß /50/).

Bei den Anwendungen in 1.2.2 geht es im wesentlichen um die Analyse **statischer Szenen** mit **starrten Einzelobjekten**. Falls dynamische Objekte vorkommen, werden sie wie unbekannte Objekte behandelt. Bei starren, auf Transporteinrichtungen (Band, Drehtisch, Palette) bewegten Werkstücken sind zumindest Transportrichtung und -geschwindigkeit a priori bekannt, also entsprechende Verschiebungsvektoren berechenbar.

Daher können wir uns auf die Analyse statischer Szenen und Objekte beschränken.

## 1.3.3 Sensordatenaufnahme

### a) Eingabedaten (Rohdaten)

Die Sensordaten zu einer Szene bilden eine digitalisierte Oberfläche und liegen in kartesischen Koordinaten, im ASCII-Format, vor (vgl. Def. 1.2). Sie können von beliebigen **geometrischen Sensoren** (1D,2D oder 3D) stammen:

- Triangulations-Laserscanner
- Laser Radar
- taktile, Ultraschall- oder Infrarot-Entfernungssensoren

Zur Gewinnung dieser Daten werden im allgemeinen zusätzlich Fahrzeug- oder Roboterkoordinaten benötigt. Sensoren, die als primäre Signale keine Geometriedaten liefern, z.B. Videokameras oder Kraft-Momenten-Sensoren, werden nicht betrachtet. Entsprechende Merkmale wie Farbe, Textur, Beschriftung, Härte o.ä. stehen also für die Objekterkennung nicht zur Verfügung.

#### b) Teilszenen

Wie in 1.1 (Def. 1.2) bereits festgestellt, erfüllt die Oberfläche im allgemeinen nicht die  $2\frac{1}{2}D$ -Eigenschaft, die viele Verfahren der Entfernungsbildverarbeitung benötigen.

Jedoch ist die gesamte Oberfläche als Vereinigung endlich vieler Teilansichten mit jeweils konstanter Sensororientierung darstellbar, von denen jede einzelne die  $2\frac{1}{2}D$ -Eigenschaft besitzt. Wird dies ausgenutzt, muß das Problem der Fusionierung dieser Teilansichten gelöst werden (1.5.1).

#### c) Meßstrategie

Mehrere Forschungsarbeiten behandeln das Problem, den Blickwinkel für die Sensordatenerfassung so zu wählen, daß eine optimale Abbildung erreicht wird ('active sensing' /45/). Teilweise wird sogar an eine on-line Rückkoppelung des Meßprozesses durch Auswertungsergebnisse gedacht.

Bei der Werkstückerkennung oder der Ortsbestimmung autonomer Fahrzeuge sind solche on-line-Optimierungsmöglichkeiten durch die Anordnung der Sensoren, den Manövrierraum des Fahrzeugs, die verfügbare Zeit und andere Randbedingungen stark eingeschränkt. Eher sind sie noch bei der Planung des Arbeitsraums oder Sensorsystems zu berücksichtigen. Deshalb sehen wir den Prozeß der Datengewinnung für die Wiedererkennung als gegeben und nicht beeinflussbar an.

Anders bei der **Rekonstruktion**: hier ist es sehr wohl von Interesse, ein einzelnes Objekt mit möglichst wenigen Ansichten aus Abstandsdaten zu rekonstruieren. Soll ein vollständiges CAD-Modell erzeugt werden, so sind in der Regel alle Hinterschneidungen und Abschattungen auszuleuchten. Ist das Ziel nur die Wiedererkennung, so reicht eine teilweise Vermessung aus. Objektsymmetrien können ausgenutzt, und bei der Erkennung nicht vorkommende Blickrichtungen ausgelassen werden.

#### d) Sensormodelle

Die Auflösung und Genauigkeit der Sensordaten, die Qualität, Zuverlässigkeit und Verfügbarkeit daraus abgeleiteter Merkmale, der Zeitaufwand zu ihrer Gewinnung (Messung, Vorverarbeitung) sind Faktoren, die den Prozeß der Wiedererkennung entscheidend beeinflussen und daher auch berücksichtigt werden sollten.

Im Prinzip können solche Informationen durch Sensormodelle beschrieben werden /14/ /55/. Diese sollten nicht nur die Sensor-Hardware, sondern auch die Vorverarbeitung und Segmentierung beschreiben. Es wird jedoch während der Projektlaufzeit nicht möglich sein, ein auch nur annähernd vollständiges Modell für die Sensoren und Vorverarbeitungsschritte zu erarbeiten. Die Objekterkennung kann sich hier nur auf punktuelle Erfahrungswerte oder Herstellerangaben stützen.

#### e) Aktivierung

Wir entwickeln und erproben zuerst die Verfahren zur Entfernungsbildverarbeitung als solche, bevor wir berücksichtigen, wie diese in die Steuerungs-SW der Anwendung eingebettet werden, z.B. wann, wie oft und in welcher Form die Bildauswertung aktiviert wird oder Resultate abliefern. Jede Szene wird als in sich abgeschlossener Auftrag angesehen und komplett ausgewertet, bevor die nächste Szene bearbeitet wird (**Einzelbildauswertung**).

Wichtigster Anwendungsfall ist die Szenenerkennung von einem stationären (Portal-)Roboter oder Kran aus. Beim Einsatz in autonomen Fahrzeugen wird man in der Praxis anders vorgehen: die Wahrnehmungskomponente als unabhängig arbeitendes Teilsystem, das den visuellen Eingabestrom kontinuierlich verarbeitet und asynchron auf Aufträge anderer Teilsysteme, nach bestimmten Objekten zu suchen, reagiert. Anstelle der

Einzelbildauswertung wird eine **inkrementelle Bildfolgenanalyse** angewandt: jede neue Sensorinformation wird gezielt auf die Unterschiede zum bisherigen Sensorabbild abgesucht und nur diese Differenzmenge genauer analysiert. Algorithmisch ähnelt dies der Fusion von Teilansichten.

### 1.3.4 Ausgangsdaten

#### a) Ausgabe

##### *Fall 1 (eindeutig erkanntes Objekt)*

Für jede Objektinstanz, deren Typ und Lage das System eindeutig identifiziert, ist auszugeben:

- Identität der Instanz und ihres Modellobjekttyps
- Koordinatentransformation der Instanz gegenüber dem objektspezifischen Koordinatensystem des Modellobjektes
  - Rotation R
  - Translation T
  - [bei generischen Objekten: Skalierung]

##### *Fall 2 (mehrdeutiges Objekt oder mehrdeutige Lage)*

Wenn mehrere Interpretationen der Objektidentität und/oder Lage - wegen Verdeckung, Symmetrien oder ungünstigem Blickwinkel - in Frage kommen, sollte das System im Prinzip **alle** Interpretationen im Format von Fall 1 liefern, wie es ja auch Definition 1.4 in 1.1 entspricht, statt nur **eine** Interpretation anzubieten, von der es "glaubt", daß sie die richtige sei. Es kann nämlich notwendig sein, zusätzliche Merkmale aus anderen Informationsquellen heranzuziehen, um die richtige Interpretation auszuwählen. Zumindest soll das System melden, daß die Interpretation mehrdeutig ist.

##### *Fall 3 (unerkanntes Objekt)*

Wurden Merkmale gefunden, die zu keinem der bekannten Objekte passen, sollte ausgegeben werden:

- alle zu dem unbekanntem Objekt gehörenden, benachbarten Teile bzw. Merkmale, falls mehrere solche Merkmale als zum selben Objekt gehörig erkannt werden können; andernfalls bildet jede nicht zuordbare Oberfläche ein eigenes Pseudo-Objekt
- ungefähre Lage und Ausdehnung des unbekanntem Objektes (Umhüllende, Schwerpunkt)

##### **Zusätzliche Informationen:**

- bei erkannten Objekten: Zahl und Prozentanteil der in der Szene wiedererkannten Merkmale, ggf. gewichtet nach Signifikanz
- Falls Angaben über die Genauigkeit/Zuverlässigkeit der Sensordaten existieren, sollte daraus eine entsprechende Genauigkeit für die Objektlagebestimmung errechnet werden ("Propagierung der Unschärfe")

#### b) Trefferquote

Folgende Leistungsmaße sind durch Vergleich mit der realen Szene zu bestimmen:

- Zahl der Objekte, für die das System **qualitativ** falsch, d.h. gemäß der Fälle 1-3 anders entscheidet, als aufgrund der vorliegenden Szenedaten und des Vorwissens zu erwarten wäre.
- Genauigkeit der errechneten Lage für die korrekt erkannten Objekte (**quantitativer Fehler**).

#### c) Antwortzeit

Zu bestimmen ist die Zeit vom Vorliegen des gesamten Entfernungsbildes bis zum Ende der Auswertung.

## 1.4 Prinzipien und Forschungsziele

### 1.4.1 Das Geometriemodell zwischen Rekonstruktion und Wiedererkennung

Ein Objekterkennungssystem benötigt Vorwissen über die zu erwartenden Objekte. Diese, und die zu erkennende Szene sollten durch Geometriemodelle kompakt und **einheitlich** beschrieben sein, um nicht bei der Merkmalszuordnung ständig unterschiedliche Darstellungen ineinander transformieren zu müssen. Der Vorteil der gleichen Repräsentation wird heute überwiegend akzeptiert und genutzt.

Welche Repräsentation nun am geeignetsten ist, hängt davon ab, auf welche Weise die Modellobjekte erstellt werden. Bei den meisten Laborsystemen zur Entfernungsbildverarbeitung werden diese vom Benutzer, der oft zugleich der Systementwickler ist, selbst konstruiert. CAD-Systeme werden hierfür in zunehmendem Maße eingesetzt /17/ /36/ /47/. Da CAD-Modelle nicht alle für die Objekterkennung wichtigen Merkmale explizit enthalten, müssen diese durch nachgeschaltete, anwendungsspezifische SW aus der internen Darstellung errechnet werden.

Wir gehen hier aber einen Schritt weiter: Objektmodelle sollen nicht (nur) manuell konstruiert, sondern vom System selbständig aus Entfernungssensordaten rekonstruiert, in einer Modellobjekt-Bibliothek abgelegt und auf unbekannte Szenen angewandt werden können. Dafür gibt es folgende Gründe:

- (1) die nachträgliche Erstellung von CAD-Modellen existierender Werkstücke ist selbst ein wichtiges Teilziel, weil für viele Werkstücke in der Praxis keine solchen Modelle zur Verfügung stehen.
- (2) Bei autonomen Fahrzeuganwendungen muß sich das Fahrzeug in jedem Fall mit seinen Sensoren das Umweltmodell selbst schrittweise aufbauen und kann nicht davon ausgehen, daß ihm CAD-Modelle aller in seiner Umgebung vorkommenden Objekte zur Verfügung stehen.

Das Geometriemodell muß also alle für die Wiedererkennung wichtige Information liefern, aber diese Information muß das System auch selbständig für neue, unbekannte Objekte aus Sensordaten erwerben können. Beim Entwurf des Geometriemodells sollten Rekonstruktion und Wiedererkennung als **Einheit** betrachtet werden, statt die Bürde einseitig zu Lasten eines Problems zu verteilen.

Einige Sonderfälle von Geometriemodellen erweisen sich dabei von vornherein als ungeeignet.

#### *Darstellung aus Ansichten* (engl. multi-view representations, aspect graphs)

Jedes Objekt wird durch eine Menge von Ansichten repräsentiert. Eine Ansicht ist die **Projektion** der von einem bestimmten Blickwinkel aus sichtbaren  $2\frac{1}{2}$ D-Oberfläche des Objektes auf eine **2D-Bildebene**. Darstellungen aus Ansichten sind bei Grauwert-Bildverarbeitungssystemen beliebt, bedeuten aber bei Entfernungsbildern den **Verlust der Tiefeninformation**. Merkmale aus Ansichten sind **nicht lageinvariant**. Bei unbekannter Objektlage müssen alle möglichen, zumindest alle wesentlich verschiedenen Ansichten repräsentiert und durchsucht werden. Dazu wird eine parzellierte Einheitssphäre von Blickrichtungen definiert /32/ /51/ /53/. Der große Nachteil dieser diskretisierten Darstellung besteht darin, daß die Genauigkeit der Lagebestimmung zwangsläufig mit dem Speicher- und Rechenaufwand gekoppelt wird.

#### *Kanten- bzw. Drahtmodelle* (engl. wire-frame)

Bei dieser Darstellung werden nur die Ecken des Objektes und die sie verbindenden Kanten als Graph dargestellt. Es entsteht eine Linien- bzw. Umrißzeichnung, bei der aber der **Materialrichtungsvektor** und wesentliche Oberflächeneigenschaften, wie z.B. **Krümmungen**, verlorengehen. Dies führt zu mehrdeutigen Interpretationen von Objektdarstellungen.

Es sollte aber erwähnt werden, daß einige frühe Objekterkennungssysteme in der Literatur solche Linienzeichnungen zur Objektrepräsentation verwendeten, z.B. Waltz /66/.

**Volumenmodelle** besitzen einen **höheren Informationsgehalt** als Drahtmodelle oder Ansichten, und reduzieren dadurch die **Mehrdeutigkeiten** der Interpretation. In CAD-Systemen haben sich 3 Hauptklassen von Volumenmodellen herauskristallisiert. Man vgl dazu auch die Arbeit von Seiler /56/ und Abb. 1.5.

**Volumenbelegungsmodelle, Zellmodelle** (engl. spatial occupancy, octree)

Hierbei wird der euklidische Raum in Oktanten bzgl. eines beliebigen Objektkoordinatensystems eingeteilt. Oktanten werden als belegt gekennzeichnet, wenn sie einen Teil des Objektes enthalten, andernfalls als frei. Diese Zerlegung wird rekursiv auf alle belegten Oktanten bis zu einer gewünschten Auflösung fortgesetzt. Die Darstellung ist also eine Baumstruktur.

**Oberflächenmodelle** (engl. boundary representation, B-rep)

Jedes Objekt wird durch seine sichtbaren Oberflächen charakterisiert. Eine Oberfläche besitzt genau eine äußere Berandung, zusätzlich unter Umständen innere Berandungen ("Löcher"), einen Materialrichtungsvektor und eine Approximationsfunktion, die die Form der Oberfläche charakterisiert. Es gibt verschiedene Arten von Oberflächenmodellen, deren Komplexität von eben-facettierter Darstellung über biquadratische Approximationen oder Quadriken bis zu den sog. Freiform-Flächenmodellen oder gar allgemein analytischen Approximationen reicht.

**Generative Modelle, Verknüpfungsmodelle** (z.B. CSG=constructive solid geometry, sweep-Modelle)

Bei der CSG-Darstellung ist eine Menge parametrisierbarer Grundkörper gegeben, meist Quader, Zylinder, Kugel und Kegel, sowie eine Menge von Verknüpfungsoperatoren wie Vereinigung und Differenz. Jedes Objekt entsteht durch endlichmalige Verknüpfung endlich vieler Grundkörper.

Bei der sweep-Repräsentation werden verallgemeinerte Zylinder durch Translation, Rotation oder Größenskalierung einer Fläche bezogen auf die Punkte einer stetigen Raumkurve (Achse) gebildet. Auch hierbei handelt es sich um generative Modelle: aus Grundprimitiven entstehen komplexere Körper durch Anwendung eines Formungsgesetzes.

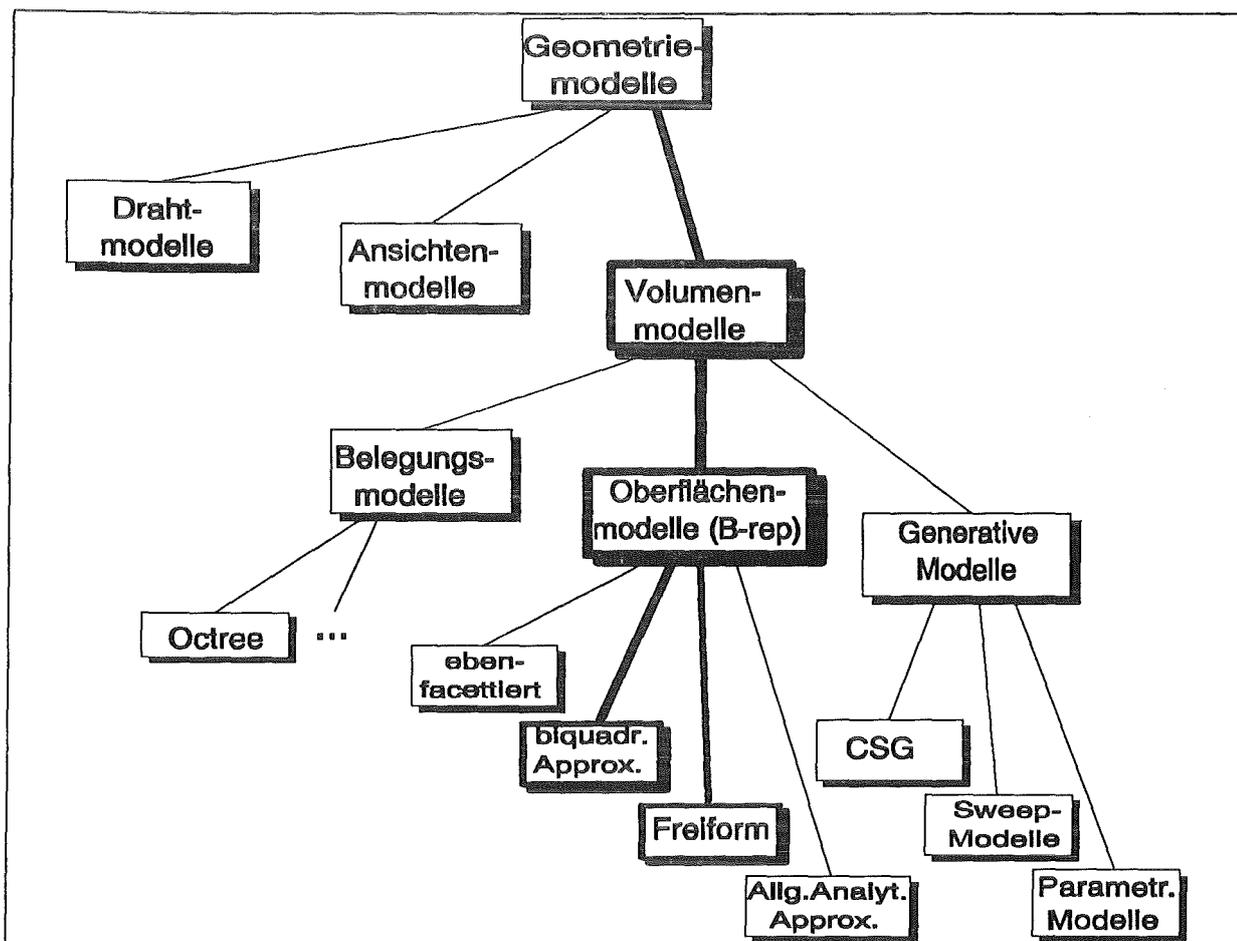


Abb. 1.5

Taxonomie der Geometriemodelle

## Vergleich der Volumenmodelle

Die *Volumenbelegung* eines Objektes läßt sich aus Entfernungssensordaten algorithmisch relativ leicht *rekonstruieren*. Die Volumenbelegung eignet sich dagegen schlecht zur *Wiedererkennung*, weil sie das **Objektinnere** charakterisiert, aber **nicht die markanten äußeren Merkmale**. Zudem gibt es keine signifikanten, unter Translation, Rotation, Skalierung und insbesondere Verdeckung eines Objektes invarianten Eigenschaften der Volumenbelegung.

Demgegenüber beschreiben *Oberflächenmodelle* die sichtbaren, äußeren Objektmerkmale, die aus **Attributen von Oberflächen**, z.B. Krümmungen, und **Relationen**, z.B. Nachbarschaft, Schnittwinkel, Eckenrelation, abgeleitet werden. Sie bilden also eine gute Ausgangsbasis für die *Wiedererkennung*. Andererseits ergibt sich die oberflächenorientierte Darstellung natürlicherweise aus der **Segmentierung** der Sensordaten in Flächen, eignet sich also auch für eine bottom-up *Rekonstruktion* unbekannter Objekte.

Zur *Wiedererkennung* eignen sich auch *Verknüpfungsmodelle*. Sie besitzen einen noch höheren Informationsgehalt als Oberflächenmodelle, weil sie nur **topologisch sinnvolle** Objekte beschreiben, und diese **eindeutig**. Durch das Mitführen der Entstehungsgeschichte ist diese Darstellung besonders änderungsfreundlich und daher im CAD-Konstruktionsprozeß vorteilhaft. Außerdem ist sie kompakter als B-rep. Viele CAD-Systeme verwenden beide Darstellungen gleichzeitig, letztere für die Grafik. Aus dem Verknüpfungsmodell ist die Oberflächendarstellung jederzeit ableitbar. Die Umkehrung gilt dagegen i.a. nicht: es gibt **kein allgemeines Verfahren**, um aus einer beliebigen Oberflächendarstellung das unbekannte Formungsgesetz nachträglich zu erschließen. Für die automatische *Objektrekonstruktion* aus Entfernungsbildern stellen Verknüpfungsmodelle eine kaum überwindbare Hürde dar.

Die **Oberflächendarstellung (B-rep)** ist also der Kompromiß, der die Anforderungen von automatischer Rekonstruktion und Wiedererkennung am besten vereint.

Folgende Entwurfsentscheidungen sind nun noch zu treffen:

### 1. Zerlegungskriterien

Nach welchen Kriterien soll die Zerlegung des Entfernungsbildes in einzelne Flächen erfolgen? Auch hier ist im Grunde ein Kompromiß zwischen Rekonstruktion und Wiedererkennung gefragt. Ein CAD-Modell erfordert genaue Maßangaben. Diesem Ziel ist gedient, wenn alle Punkte derselben Fläche z.B. durch eine Polynomfunktion niedriger Ordnung bei einem möglichst kleinem Ausgleichsfehler approximierbar sind. Dies führt zu einer Segmentierung in **viele kleine Flächen(stücke)**, die aber wenig Anhaltspunkte für die Erkennung bieten. Besonders augenfällig wird dies, wenn gekrümmte Oberflächen durch ebene Facetten approximiert werden.

**Wenige große**, nach topologischen Kriterien gebildete Flächen sind demgegenüber numerisch schwer approximierbar, bilden aber die groben visuellen Konturen für die Erkennung besser ab. Mathematisch bedeutet Segmentierung die **Partition** einer Oberfläche  $B$  in Regionen  $R_i$  /2/:

$$B = \bigcup_{i=1}^k R_i \text{ mit } R_i \cap R_j = \emptyset \text{ für } i \neq j \text{ so daß}$$

- $P(R_i) \quad \forall i \quad P$  Prädikat über Punktemengen  $R \subseteq B$
- $\neg P(R_i \cup R_j) \quad \forall i, j$

Die Regionen sind also **maximale** zusammengehörige Punktemengen, wobei die Zusammengehörigkeit durch das "Uniformitätsprädikat"  $P$  spezifiziert wird. Zum Beispiel sollen je zwei Punkte derselben Region  $R$  auf einem Weg innerhalb  $R$  ohne sprungförmige Übergänge des Entfernungsbildes und des Normalenvektors, aber auch - wichtig für gekrümmte Flächen ohne scharfe Kantenübergänge - ohne prinzipielle Änderungen des Krümmungsverhaltens **erreichbar** sein. (Übergänge dieser Art können natürlich **lokal** innerhalb einer zusammenhängenden Fläche dennoch auftreten). Durch solche Eigenschaften lassen sich die groben visuellen Konturen mathematisch fassen.

## 2. Objektrepräsentation und CAD-Schnittstelle

Wie bereits erwähnt, erfordert die Objekterkennung weitere Flächenattribute und -relationen mit zugehörigen Berechnungsvorschriften, als die Oberflächenrepräsentation existierender CAD-Systeme normalerweise bereitstellt. Diese werden in Abschnitt 2.1 genauer beschrieben. Es ist zu klären, ob solche Erweiterungen programmtechnisch leicht in vorhandene CAD-Systeme integriert werden können, und welche CAD-Schnittstelle als Ausgangspunkt sich am besten zur Implementation der Objektrepräsentation eignet.

### 1.4.2 Explizites Anwendungswissen statt impliziter Heuristik

Während die Vorverarbeitung der Entfernungsbilder, also Filterung, Merkmalsextraktion und Segmentierung, bei den meisten Objekterkennungssystemen zwar abhängig vom Sensor, aber unabhängig von der Anwendung ist, wird die eigentliche Wiedererkennung durch Vorwissen aus der Anwendung, also Erwartungen gesteuert, welche Modellobjekte auftreten können und in welchen Beziehungen sie zueinander stehen /36/. Es handelt sich dabei um eine Art der Wissensverarbeitung, die auf überwiegend geometrischem, unter Umständen auch auf physikalischem Grundwissen, z.B. den Gesetzen der Schwerkraft, operiert.

Wir fordern nun, daß alles Wissen über die Anwendungsumgebung **explizit** in den **Modellobjekten** und ihren **Beziehungen** zu halten und von den **Verfahren** zur Wiedererkennung selbst **strikt zu trennen** ist (**wissensbasierte Darstellung**). Das sollte nicht nur für die Geometrieigenschaften der Modellobjekte, sondern möglichst auch für daraus abgeleitetes Metawissen, wie Reihenfolgestrategien zum Testen bestimmter Hypothesen über Objekttyp und -lage, durchgehalten werden.

Das Vorwissen darf also nicht in Form impliziter Annahmen über die Anwendungsumgebung in den Algorithmen versteckt und damit "zementiert" werden. Eine solche Vorgehensweise führt zu einem System, das versagt, wenn nur eine seiner wesentlichen Annahmen fehlschlägt, und das in seiner Fehlfunktion auch nicht mehr durchschaubar und diagnostizierbar ist. Ein komplexes System benötigt in der Regel viel Wissen dieser Art an verschiedenen Stellen. Es ist nicht ohne umfangreiche Neuprogrammierung an auch nur leicht geänderte Umgebungsbedingungen anpaßbar.

Ein System, in dem das Wissen explizit repräsentiert ist, ist demgegenüber leichter zu testen und experimentell zu evaluieren. Der Kern, sein Erkennungsalgorithmus, ist wiederverwendbar. Die Benutzung speziellen Anwendungswissens beschleunigt den Erkennungs- und Eingrenzungsprozeß erheblich. Wenn die Annahmen nicht erfüllt sind, muß auf dieses Wissen verzichtet werden. Dies führt aber nur zu einem Effizienzverlust, nicht zu einer Fehlfunktion.

Der Unterschied zwischen den Vorgehensweisen soll an 2 einfachen Beispielen erläutert werden; die konkreten Verfahren zur Objekterkennung werden in Kap. 2.4 besprochen.

#### **Beispiel 1: Trennung des Bildhintergrundes und Ausnutzung der Vorzugslage**

Eine wichtige Information in einer Anwendung ist z.B. die Kenntnis, daß der Sensor von oben Werkstücke auf einer ebenen Arbeitsfläche (Fußboden o.ä.) sieht. Diese Arbeitsfläche bilde den Bildhintergrund, innerhalb dessen jedes einzelne Werkstück Platz habe. Es gebe zylinderförmige Objekte, die mit einer ihrer kreisförmigen Flächen auf der Arbeitsfläche stehen, sowie quaderförmige Objekte, die nicht hochkant, d.h. nicht mit ihrer kleinsten Fläche auf der Arbeitsfläche stehen. Dies sind Beispiele von Vorzugslagen und stabilen Orientierungen von Objekten.

#### **Ansatz A (inkorrekt!)**

Um die interessierenden Werkstücke vom Bildhintergrund zu trennen und zu klassifizieren, könnte man folgende sehr einfache Heuristik entwerfen (die Vorverarbeitung des Entfernungsbildes liefere zu jeder Fläche  $F$  eine Abschätzung des Schwerpunktes  $F_s$ , ferner eine Aussage, ob ihre Berandung kreisförmig ist).

```

for all F do
  if F.s[3] < threshold then      {...[3] sei die z-Koordinate)
    {F gehört zum Bildhintergrund)
    ...
  else
    if kreisfoermig (F) then
      {F ist zylinderförmiges Objekt)
      ...
    else
      {F ist quaderförmiges Objekt)
      ...

```

Ein eher technisches Problem des Schwellwertkriteriums ist, daß es nicht lageinvariant ist, sondern nur bei exakt horizontalem Fußboden ( $z=const$ ) funktioniert, wobei die z-Achse zum Sensor hin zeigt. Bereits ein Fehler in der Sensorkalibrierung (Schräglage des Hintergrundes) kann zu einer Fehlentscheidung führen, wo der Hintergrund und wo die Objekte sind. Bei leicht geänderter Aufgabenstellung, wenn der Sensor die gleiche Sorte von Werkstücken z.B. vor dem Hintergrund einer Werkshalle oder eines Regals sieht, bricht die Grundstrategie zusammen.

Auch die weiteren ad-hoc-Kriterien für die Objekttypen sind gelinde gesagt voreilig. Zwar liefern zylinderförmige Werkstücke unter den Lageannahmen in der Draufsicht kreisförmige Flächen, doch der Umkehrschluß ist ohne genaueres Vorwissen über die Werkstücke unzulässig. Es könnte sich beispielsweise auch um quaderförmige Werkstücke mit einer kreisförmigen Aussparung handeln. Annahmen dieser Art sind viel zu speziell, um sie für jede Situation neu "in Anweisungen zu gießen".

## Ansatz B

Das spezielle Anwendungswissen läßt sich genauso effizient benutzen, aber so in der Objektrepräsentation ablegen, daß die Schlußfolgerungsmechanismen (Erkennungsalgorithmen) unabhängig von seiner Gültigkeit sind.

1. Der Bildhintergrund ist ein **spezielles Modellobjekt**  $M_0$ , das nur aus einer einzigen ebenen Fläche  $F_0$  besteht (CAD-Experten mögen diesen Lapsus verzeihen), deren Ausdehnung dem gesamten Arbeitsraum entspricht.
2. Zylinderförmige Objekte sind durch 3 Flächen  $F_{oben}$ ,  $F_{unten}$ ,  $F_{mantel}$  und Relationen zwischen diesen Flächen repräsentiert (die Syntax der Darstellung ist unerheblich):

*cylinder 1*

```

[surface [Foben, eben [...], rand [..., kreisförmig,...], ausdehnung [......]],
[surface [Funten, eben [...], rand [..., kreisförmig,...], ausdehnung [......]],
[surface [Fmantel, zylindrisch, ...]],
[relation [...]]
...

```

Vorzugslagen sind spezielle Relationen zwischen Flächen verschiedener Objekte, z.B.

$$(R1) \quad \forall F: \text{kreisförmig}(F) \wedge \text{eben}(F) \wedge F \text{ Zylinderoberfläche} \Rightarrow \text{parallel}(F, F_0)$$

$$(R2) \quad \forall F: \text{eben}(F) \wedge F \text{ kleinste Quaderoberfläche} \Rightarrow \text{senkrecht}(F, F_0)$$

Solches Wissen läßt sich ähnlich effizient anwenden wie in ad-hoc-Strategie A. Wenn z.B. in einer Szene eine ebene Fläche  $FS_0$  existiert, die aufgrund ihrer Ausdehnung in keine Modellfläche "hineinpaßt", ist diese sofort und unabhängig von ihrer Raumlage als Hintergrund erkannt. Alle zu  $FS_0$  parallelen, kreisförmigen Szeneflächen  $FS_i$  sind mit (R1) entweder sofort als Boden und Deckel eines Zylinders erkannt - damit wäre auch die Lage bestimmt - oder die Hypothese "Objekttyp ist Zylinder" ist falsch. Letztere Möglichkeit müssen wir - anders als Ansatz A - zulassen, es sei denn, nur zylinderförmige Modellobjekte in unserer Wissensbasis besitzen kreisförmige ebene Flächen. Wenn nur Zylinder

kreisförmige Flächen haben, kann das System ferner alle kreisförmigen, nicht zu  $FS_0$  parallelen Flächen als unbekannt, da (R1) widersprechende, Objekte identifizieren.

In einer Anwendung ohne ebene Arbeitsfläche oder ohne bekannten Vorzugslagen fehlen lediglich die entsprechenden Modellobjekte und Relationen. Aus ihnen können dann keine Folgerungen über die Objektidentität oder -lage gezogen werden, und die Interpretationen des Objekterkennungssystems bleiben mehrdeutig, wenn die Eindeutigkeit nicht durch andere Beschränkungen gegeben ist. Gleiches gilt z.B. für eine Anwendung, für die zwar ein ebener Arbeitsbereich existiert, dieser aber so mit Objekten zugestellt ist, daß sein Rand nicht sichtbar ist, also über das Attribut Ausdehnung keine Erkennung der Arbeitsfläche möglich ist.

### Beispiel 2: Objektgruppierung durch Auswertung der Flächennachbarschaftsbeziehung

Die Ausnutzung vieler geometrischer Beziehungen zwischen Flächen setzt voraus, daß diese alle zum selben, starren Objekt gehören. So schneiden sich zum Beispiel Flächen derselben Objektinstanz unabhängig von der Raumlage und von partiellen Verdeckungen unter einem definierten Winkel. Es ist daher wichtig zu wissen, welche Flächen in einer Szene zur selben und welche zu verschiedenen Objektinstanzen gehören. Dieses Problem bezeichnen wir als **Objektgruppierung**.

Wichtige Hinweise hierzu liefert die Analyse der Kanten auf dem gemeinsamen Rand benachbarter Szeneflächen:

- **Sprungkanten** (engl. step edges, jumps) sind Unstetigkeitsstellen des Entfernungsbildes

Spezialfälle:  $F1$  verdeckt  $F2$   $\Leftrightarrow$   $F1, F2$  durch Sprungkante verbunden,  $F1$  näher am Sensor

$F1$  verdeckt durch  $F2$   $\Leftrightarrow$   $F1, F2$  durch Sprungkante verbunden,  $F1$  weiter vom Sensor entfernt

- **Dachkanten** (engl. roof edges, creases) sind stetige Übergänge des Entfernungsbildes, aber Unstetigkeitsstellen des Normalenvektorfeldes

Spezialfälle:  $F1$  und  $F2$  bilden einen *konvexen* Übergang  
 $F1$  und  $F2$  bilden einen *konkaven* Übergang

#### Ansatz A

Ob zwei Flächen  $F1, F2$  zur selben oder zu verschiedenen Objektinstanzen gehören, wird durch eine Heuristik in Form einer a priori Wahrscheinlichkeit  $p$  festgelegt:

$p=1$   $\Leftrightarrow$   $F1, F2$  sind durch eine konvexe Dachkante verbunden

$p=0.75$   $\Leftrightarrow$   $F1, F2$  sind durch eine konkave Dachkante verbunden

$p=0.25$   $\Leftrightarrow$   $F1, F2$  sind durch eine Sprungkante verbunden

Es wird anschließend ein Graph konstruiert, dessen Knoten die Flächen repräsentieren und dessen Kanten diejenigen Flächennachbarn verbinden, zwischen denen eine gemeinsame Dachkante mit der Wahrscheinlichkeit  $p > 0.5$  existiert. Jede Zusammenhangskomponente dieses Graphen bildet eine Objektinstanz. Ein solches Verfahren wird z.B. von T.J.Fan /15/ verwendet.

#### Ansatz B

Es werden **keine a priori** Wahrscheinlichkeiten für die Zugehörigkeit von Flächen zur selben Objektinstanz festgelegt. Vielmehr hängt die Gruppierung von den **konkreten Beschränkungen** ab, die die Modellobjekte auferlegen. Dabei sind i.a. mehrere, baumartig strukturierte Hypothesen für die Objektgruppierung zu untersuchen. Aus den Häufigkeiten, mit der bestimmte Kantenübergänge in den Modellobjekten und in der Szene vorkommen, kann eine anwendungsspezifische Reihenfolgestrategie abgeleitet werden, welche die wahrscheinlichsten Hypothesen zuerst testet. Dies beeinflußt aber nur das Zeitverhalten der Erkennung, nicht ihr Ergebnis, weil auch die weniger wahrscheinlichen Hypothesen später eine Chance erhalten, falls die wahrscheinlicheren sich als falsch erwiesen haben.

Wenn die Flächen jedes Modellobjektes rechte oder stumpfe Winkel bilden, können zwei verschiedene Objektinstanzen in der Tat nicht so angeordnet werden, daß sie in einer **konvexen Dachkante** bündig aneinanderstoßen. Bei 2 Objekten mit spitzen Ecken kann aber genau dieser Fall auftreten (nach der Heuristik müßten alle Flächen, die eine konvexe Dachkante bilden, mit Wahrscheinlichkeit 1 zum **selben** Objekt gehören).

Sind alle Modellobjekte konvex, können sie unter **keiner** Ansicht Teile von sich selbst verdecken. Sich verdeckende, also durch Sprungkanten verbundene Flächen gehören dann **sicher** zu verschiedenen Objektinstanzen, und nicht nur mit Wahrscheinlichkeit  $1-p = 0.75$  wie im Fall 3 oben. Bei nichtkonvexen Objekten, wie z.B. Robotern, Büromöbeln und auch vielen industriellen Werkstücken treten dagegen Selbstverdeckungen häufig auf, und es wäre voreilig, die Objektgruppierung allein mit Hilfe der verdeckenden Sprungkanten vorzunehmen.

Die Möglichkeiten, also die auftretenden Flächennachbarschaften in Abhängigkeit von den Modellobjekteigenschaften, sind aus folgender Tabelle ersichtlich.

Modellobjekte Flächennachbarschaften	konvexe Modellobjekte, "stumpfwinklig"	konvexe Modellobjekte, "spitzwinklig"	nichtkonvexe Modellobjekte
konvexe Dachkante	- dasselbe Objekt	- dasselbe Objekt, konvexe Ecke - verschiedene Objekte, mit spitzen Ecken bündig anliegend	- dasselbe Objekt, konvexe Ecke - verschiedene Objekte, mit spitzen Ecken bündig anliegend
konkave Dachkante	- verschiedene Objekte, bündig anliegend	- verschiedene Objekte, bündig anliegend	- dasselbe Objekt, konkave Ecke - verschiedene Objekte, bündig anliegend
Sprungkante	- verschiedene Objekte, Verdeckung	- verschiedene Objekte, Verdeckung	- verschiedene Objekte, Verdeckung - gegenseitig verdeckende Flächen desselben Objektes

Tab. 1.2 Arten von Flächennachbarschaften in Abhängigkeit von den Modellobjekten

Während also Ansatz A die Zerlegung in disjunkte Instanzen auf implizites, nicht näher spezifiziertes Vorwissen gründet, gehen bei B die Eigenschaften der Anwendung explizit ein.

Anders als in Beispiel 1 läßt sich der Ansatz A nicht grundsätzlich als falsch verwerfen; solche Heuristiken wurden unter anderem im Objekterkennungssystem von T.J.Fan /15/ eingesetzt, um relativ komplexe Szenen mit Spielzeugobjekten zu interpretieren. Auf die Objektgruppierung folgt hier die eigentliche Erkennung: zuerst werden die in Frage kommenden Modellobjekte ausgewählt ('screening'), dann eine detaillierte Zuordnung von Szene- zu Modellobjektflächen durchgeführt ('matching'). Die Zuordnung wird durch Berechnung der Lagetransformation verifiziert. Beide Schritte stützen sich, wie die Gruppierung, auf Heuristiken und Erfahrungswerte ab und liefern als Endergebnis eine Wahrscheinlichkeit  $w_e$  für die Glaubwürdigkeit der Erkennung.

Fällt  $w_e$  zu niedrig aus, folgt als letzter Schritt eine Reparaturmaßnahme: die Partition in disjunkte Objektinstanzen wird vergrößert oder verfeinert und die vorhergehenden Schritte wiederholt. Viele Fragen bleiben dabei offen, so zum Beispiel:

1. Kann eine **falsche** Objektgruppierung doch letztlich zu einem befriedigenden Wert für  $w_e$  führen, sodaß das System, durch seine Erkennungs-Heuristik getäuscht, sich mit der inkorrekten Lösung begnügt?
2. Kann eine **korrekte** Objektgruppierung zu einem schlechten Wert für  $w_e$  führen, wobei aber das System nicht seine Erkennung oder die vielleicht schlechte Bildqualität, sondern die Gruppierung verantwortlich macht?
3. Kann die heuristische Gruppierung in Objektinstanzen derart falsch sein, daß sie auch durch Teilen und Vereinigen nicht mehr repariert werden kann?

## 1.5 Systemarchitektur und Komponenten

Der folgende Abschnitt gibt einen groben Überblick über das Gesamtsystem zur Rekonstruktion und Wiedererkennung dreidimensionaler Objekte aus Entfernungsbildern. Der Hauptschwerpunkt in diesem Abschnitt liegt auf der Erzeugung von Oberflächendarstellungen aus Rohdaten, also Filterung, Merkmalsextraktion, Segmentierung und Fusion.

### 1.5.1 Überblick

In Abb. 1.6 sind die groben Datenflüsse dargestellt.

1. **Quelle des Datenflusses bilden ein oder mehrere Entfernungssensoren**, die auf einem Roboter oder autonomen Fahrzeug als Trägersystem montiert sind und in verschiedenen Freiheitsgraden positionierbar/orientierbar sind. Konkret arbeiten wir mit dem in 1.2.1 beschriebenen 2D LADAR Teststand.

2. Jeder Entfernungssensor liefert ein Teilbild, das eine digitalisierte Oberfläche als Folge von Punkten in kartesischen Koordinaten darstellt. Das zugehörige Weltkoordinatensystem (WKS) ist - im Gegensatz zum Koordinatensystem des Entfernungssensors - ortsfest. Eine Ordnungsreihenfolge der Punkte nach Koordinatenwerten, ein äquidistanter Abstand der Punkte, oder eine Indexierung entsprechend der Scanrichtungen, kann, muß jedoch nicht explizit gegeben sein. Dieses Format der Rohdaten ist, wie bereits in 1.3.3 festgestellt, neutral, das heißt unabhängig vom konkreten Sensor. Entfernungssensoren verschiedener Dimensionalität (1D,2D,3D) sowie verschiedener physikalischer Meßprinzipien, Auflösung, Reichweite etc. kommen in Frage.

3. **Erzeugung von Oberflächendarstellungen (Vorverarbeitung der Rohdaten)**

Die Vorverarbeitung eines Teilbildes umfaßt vier Hauptschritte, die in 1.5.2 näher erläutert werden: die Glättung durch Filteralgorithmen, die Berechnung differentialgeometrischer Krümmungsmerkmale, die Zerlegung des Bildes in berandete Flächen (Segmentierung), und die Berechnung der für die Oberflächendarstellung benötigten Flächenattribute und -relationen.

4. **Verschmelzung von Oberflächendarstellungen (Fusionierung von Teilansichten)**

Oft werden Werkstücke, Szenen oder Arbeitsbereiche aus verschiedenen Blickwinkeln oder mit verschiedenen Entfernungssensoren aufgenommen. Das Problem besteht darin, überlappende Teilansichten mit oft unterschiedlicher Auflösung oder unterschiedlich gearteten Meßfehlern zu einer konsistenten, redundanzfreien und ansichtenunabhängigen Gesamtdarstellung zu fügen.

Eine wichtige Entwurfsentscheidung ist nun, auf welcher **Ebene** die Fusion der Teilbilder stattfindet:

- a) auf **unterster** Ebene: da alle Teilansichten in kartesischen Weltkoordinaten vorliegen, ist es möglich, die Meßpunkte - mit ihren kartesischen Koordinatenwerten als Ordnungskriterium - zuerst in eine **gemeinsame Punktwolke** oder ein Gitternetz zu **sortieren**. Aus dieser einen Punktwolke wird dann eine Oberflächen-Gesamtdarstellung nach den Verfahren unter 3. berechnet (Abb. 1.7a).
- b) auf **höchster** Ebene: alle Teilansichten werden getrennt vorverarbeitet und segmentiert; die **Oberflächen(teil)darstellungen** werden erst am Ende **vereinigt** (Abb. 1.7b).

Auch wenn Methode a) auf den ersten Blick näher liegt als b) (man vgl. hierzu die Arbeiten zur Objektrekonstruktion mit Laserscannern von S.Müller /49/ sowie die Aufgabenbeschreibung der Fa. AR zur 3D Entfernungsbildverarbeitung /31/), sprechen triftige Gründe gegen eine solche Vorgehensweise:

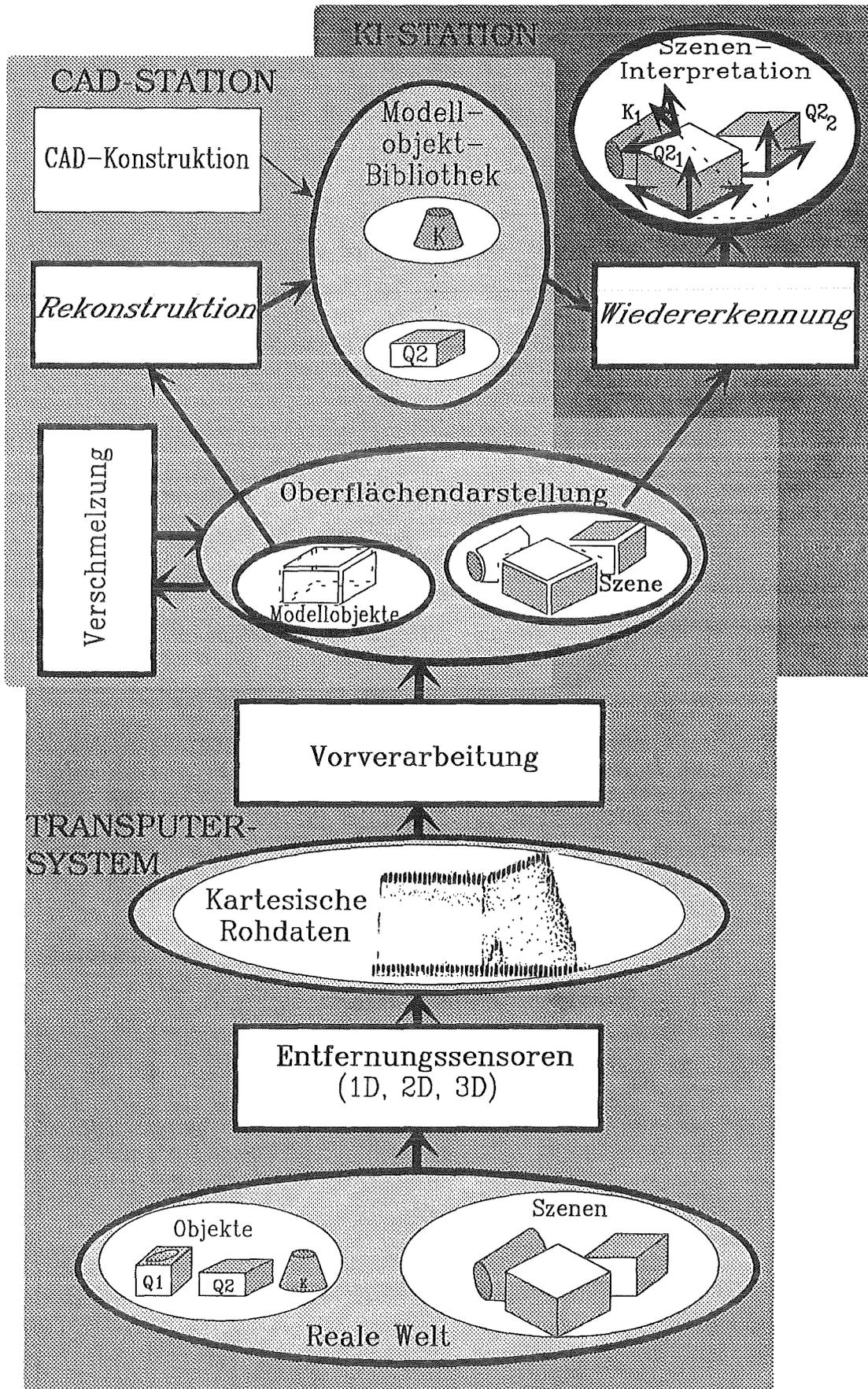


Abb. 1.6

Gesamtarchitektur und Datenflüsse zur Objekterkennung und -rekonstruktion

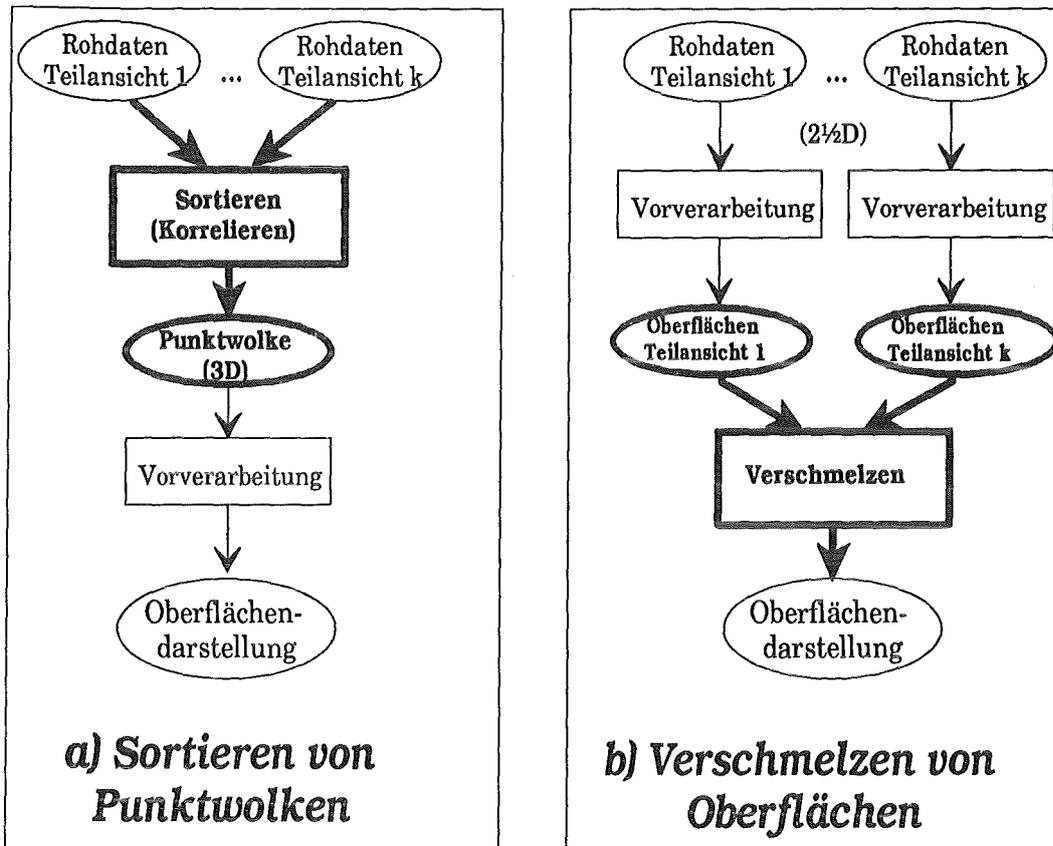


Abb. 1.7 Zwei Ansätze zur Fusionierung von Entfernungsbildern

- Selten in der Praxis sind die Parameter der Transformationen zwischen Sensor- und Weltkoordinaten so exakt bestimmt, daß die aus verschiedenen Blickwinkeln oder mit verschiedenen Sensoren aufgenommenen Punkte richtig korreliert werden können. Die genaue Bestimmung des Ursprungs und der Lage des Sensor-KS im WKS erfordert aufwendige **Kalibrierungsmessungen**. Bei beweglichen Fahrzeugen liegt das Problem in der ungenauen Positionsbestimmung durch die Koppelnavigation (Odometrie).
- Die meisten Verfahren zur Krümmungsberechnung und Segmentierung setzen die  $2\frac{1}{2}D$ -Eigenschaft der Oberfläche voraus, die verlorengeht, wenn komplexe Objekte "rundum" abgetastet und aus den Daten eine einzige, dreidimensionale Punktwolke erstellt wird.
- Durch **Abschattung** entstehen künstliche Meßpunkte ('mixed points', vgl. 1.2.1), die in manchen Teilansichten sichtbar sind und in anderen fehlen. Die Vereinigung von Teilansichten führt hier zu inkonsistenten und unsinnigen Objektdarstellungen. Einzelnen Rohdaten in der Punktwolke ist diese Eigenschaft schwer anzusehen, vor allem wenn die Blickrichtung nicht mehr bekannt ist. Schlimmer noch: oft werden solche Punkte durch **Interpolation** erst erzeugt, z.B. weil Vorverarbeitungsschritte (Filter- und Ableitungsmasken) ein äquidistantes Gitternetz erfordern.
- Die Punktwolke als zentrale Datenstruktur behindert die **Parallelverarbeitung** (1.5.3).

In den meisten Fällen führt der Weg b) eher zum Ziel. Ausgenommen sind Teilbilder, die aus so wenigen Meßpunkten bestehen, daß eine separate Vorverarbeitung und Segmentierung nicht sinnvoll ist.

Die  $2\frac{1}{2}D$ -Eigenschaft ist für Teilansichten leicht zu gewährleisten. Beim 2D LADAR etwa kann die Sensororientierung für jede Teilansicht annähernd konstant gewählt werden. Große Meßprogramme werden leicht in entsprechende Bahnelemente zerlegt. Jede Teilansicht besteht dann aus parallel

liegenden Horizontalscans ("Streifen"). Beim 3D LADAR besitzt jeder von einem festen Punkt aus aufgenommene 3D-Scan die 2½D-Eigenschaft.

Phantomflächen parallel zur bekannten Blickrichtung werden aus jeder segmentierten Teilansicht **am Ende entfernt** (1.5.2). Es verbleiben nur echte und gut sichtbare, wenn auch teilweise verdeckte Körperoberflächen. Die übrigen Oberflächen sind unter einem anderen Blickwinkel sicher besser zu erkennen.

Auch das Korrespondenzproblem unter Kalibrierungsfehlern ist für Flächen besser lösbar als für Punkte: als **Wiedererkennung von Flächen** - anstelle ganzer Objekte wie in Def. 1.4 - bei **nahezu bekannter Lage**. Es wird eine Lagetransformation  $Tr_{ij}$  (Translation und Rotation) berechnet, die als relativer Kalibrierungsfehler zwischen Teilansichten  $i$  und  $j$  interpretiert werden kann und die, auf eine von beiden angewandt, die Teilansichten zur Deckung bringt.

Insgesamt funktioniert die Verschmelzung von Oberflächendarstellungen deshalb besser als das Sortieren von Punkten, weil Teilansichten unter einem definiertem Blickwinkel, die schon in Flächen gegliedert und komprimiert sind, viel mehr **Struktur** enthalten als verrauschte Punktwolken von Rohdaten. Die Schritte dieses Verfahrens sind etwas detaillierter in /38/ erläutert.

Abb. 1.8 zeigt ein einfaches Beispiel für die Verschmelzung von Flächen. Dort werden 2 einfach zusammenhängende, unter den Ansichten A1 bzw. A2 nur teilweise sichtbare Flächen F1 bzw. F2 zu einer einzigen vollständig sichtbaren, unverdeckten Fläche F mit Loch verschmolzen. F1 und F2 besitzen je nur eine einzige (äußere) Berandung, F dagegen eine äußere und eine innere Berandung. Die Fusion erfordert eine genaue Analyse der Berandungen, insbesondere der verdeckenden Sprungkanten.

Da F1, F2 nun als Ansichten derselben Fläche F identifiziert sind, können weitere Flächen, die nur in A1 zusammen mit F1, oder nur in A2 zusammen mit F2 sichtbar sind, über das Bindeglied F in direkte Beziehung gesetzt werden (transitive Hüllenbildung).

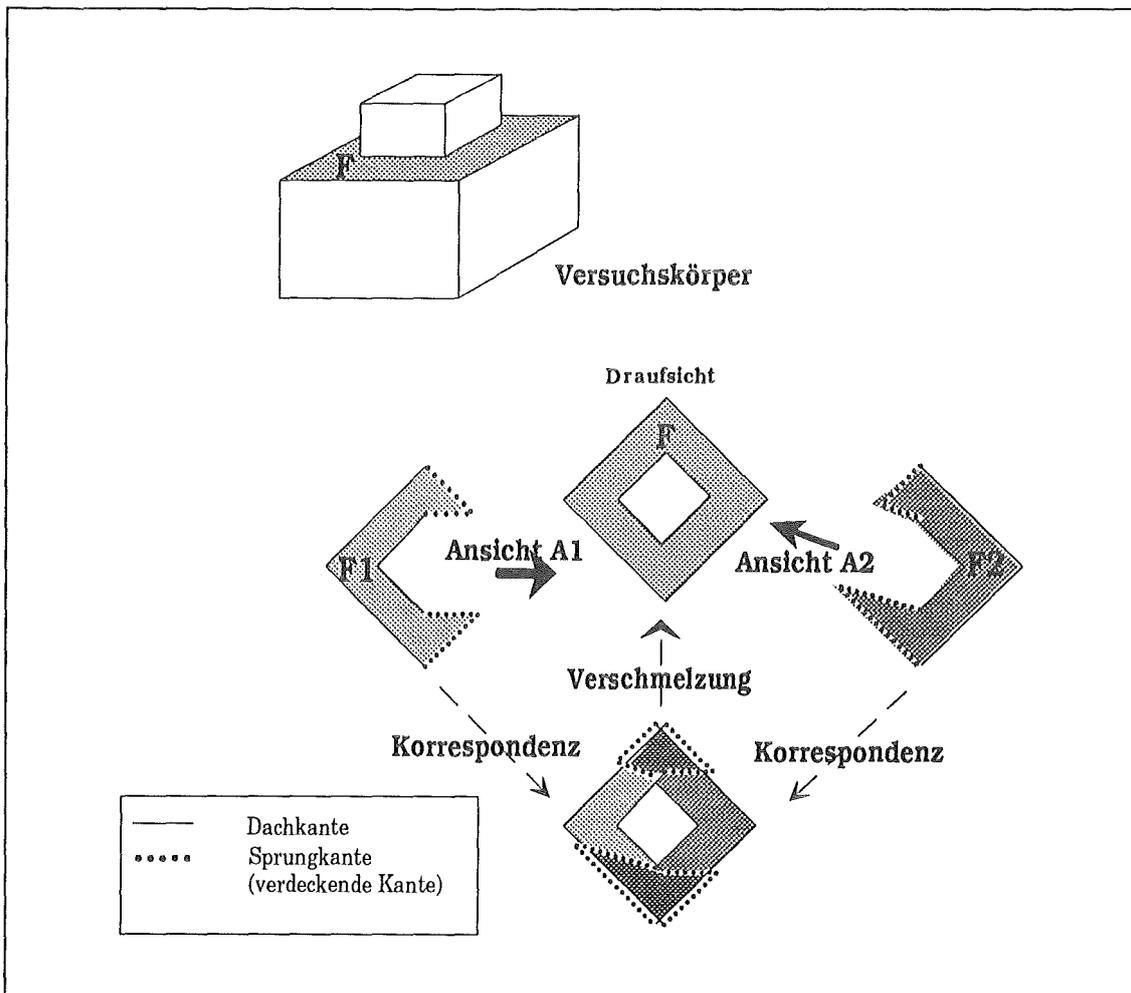


Abb. 1.8 Zur Fusionierung segmentierter Teilansichten

In der Literatur wird unter Fusion übrigens meist die Verknüpfung qualitativ verschiedener Signale, wie Intensitäts- oder Kraftsignale mit Geometriedaten verstanden /67/. Bereits die Fusion von Geometriedaten allein gehört zu den anspruchsvollsten Problemen der Vorverarbeitung. Überzeugende, allgemeingültige Lösungen haben nicht nur für das Maschinensehen große Bedeutung, sondern auch für die Weiterentwicklung von CAD-Systemen, in der Koordinatenmeßtechnik u.a.

## 5. Rekonstruktion der Modellobjekte

Nachdem die Oberflächendarstellung für einen neuen Gegenstand oder eine unbekannte Arbeitsumgebung aus Teilansichten in den Schritten 1.-4. erstellt wurde, besteht die *Rekonstruktion* eigentlich nur mehr darin, dafür einen Begriff einzuführen und die Objektrepräsentation in einer Modellobjektbibliothek abzulegen. Geometrische Beziehungen, die zwischen diesem neuen und den übrigen Objekten gelten sollen, wie z.B. die in 1.4.2 erwähnten stabilen Lagen, oder sonstiges, auch nichtgeometrisches Wissen, sind ebenfalls hier einzubringen. Dazu müssen Grundfunktionen für Einfügen, Löschen, Ändern und Auslesen von Modellobjekten und Relationen zur Verfügung stehen (Bibliotheks- und Datenbankfunktionen, vgl. Abb. 1.9).

Die Erstellung von CAD-Modellen von Werkstücken aus Sensordaten, welche den Anforderungen der offline-Programmierung genügen, gelingt jedoch selten vollautomatisch, sondern erfordert auch die Kontrolle und Eingriffe durch den Bediener, die bis auf die Ebene der Rohdaten hinunterreichen können /44/, z.B.

- Abgrenzen und Abschneiden nicht zum Werkstück gehöriger Daten, wenn versehentlich ein Teil der Umgebung des Werkstücks mit vermessen wurde.
- Prüfung des Geometriemodells auf **Konsistenz, Vollständigkeit und Genauigkeit**, ggf. Nachbesserung. Zu prüfen ist, ob keine verdeckten Flächen mehr vorhanden sind, ob alle Flächenberandungen mit anderen Flächen desselben Objektes verbunden sind und ob die Auflösung der Meßpunkte und die Approximationsgüte der Oberflächen ausreicht.
- Nachbessern von Fehlern, die bei der automatischen Segmentierung und Fusion gemacht wurden. Grundlage dafür ist die 3D-Visualisierung der Zwischenergebnisse.

Ist das Ziel der Rekonstruktion dagegen die Umweltmodellierung (Kartierung) durch ein autonomes Fahrzeug, so sind solche Bedieneringriffe weder erwünscht noch notwendig. Die Segmentierung und Fusionierung müssen jedoch einwandfrei funktionieren. Die Genauigkeitsanforderungen sind, schon wegen der großen Reichweite der Sensoren, i.a. niedrig. Rekonstruktion und Wiedererkennung sind so miteinander verzahnt, daß jede neue Sensorinformation mit dem bestehenden Umweltmodell verglichen wird, wobei der unerkannte Ausschnitt neu rekonstruiert wird. Dies geschieht solange, bis nichts Neues mehr angetroffen wird. Das Problem der Vollständigkeit und der Abgrenzung wie bei CAD-Modellen einzelner Werkstücke stellt sich hier nicht.

Neben der automatischen Rekonstruktion aus Sensordaten sollen Modellobjekte auch aus CAD-Konstruktionszeichnungen erstellt werden können (Abb. 1.9 oben). Um die gleiche Repräsentation zu verwenden, müssen

- die interne CAD-Darstellung bei der Konstruktion (z.B. CSG) in eine Oberflächendarstellung (B-rep) umgewandelt, und
- die Oberflächendarstellung um zusätzliche Attribute und Relationen erweitert werden.

## 6. Wiedererkennung der Modellobjekte in Szenen

Aufgabe der Instanz *Wiedererkennung* ist es, in einer Szene Objekte anhand einer vorgegebenen Auswahl von Modellobjekten zu identifizieren und lokalisieren, wobei die Ausgabe nach 1.3.4 zu erzeugen ist. Die Erkennung von Modellobjekten in Szenen, also die **Zuordnung** von Szenen- zu Modellobjekten wird zurückgeführt auf eine Zuordnung der **Oberflächen** unter Beachtung der

zwischen den Flächen geltenden **geometrischen Beziehungen**. Dieser Vorgang ist in Abb. 1.10 in Teilschritte zerlegt:

- Objektgruppierung: Aufteilung der Menge aller Szene-Oberflächen in disjunkte Objektinstanzen
- Bestimmung des Objekttyps jeder Instanz
- Bestimmung der Objektlage, also der Transformation zwischen Welt-KS und Modellobjekt-KS. Dazu werden Paarungen von Modellobjekt- und Szenenoberflächen herangezogen. Mit dieser Lagetransformation werden bereits getroffene Zuordnungen erhärtet oder widerlegt, und neue Zuordnungen aufgestellt.
- Planung der Suchstrategie, das heißt: Finden einer günstigen Reihenfolge für das Aufstellen der Paarungen und für die Auswertung der Attribute und Relationen. Maßgebend dafür sind die Verteilung und die Häufigkeit gewisser Merkmale in den Modellobjekten und der Szene, die ihren Informationswert bestimmen.

Diese Punkte bedürfen weiterer Klärung (Kap. 2). Offen ist z.B., ob Objektgruppierung, Typ- und Lagebestimmung wirklich getrennte Schritte bilden sollen, oder vielmehr interdependent sind.

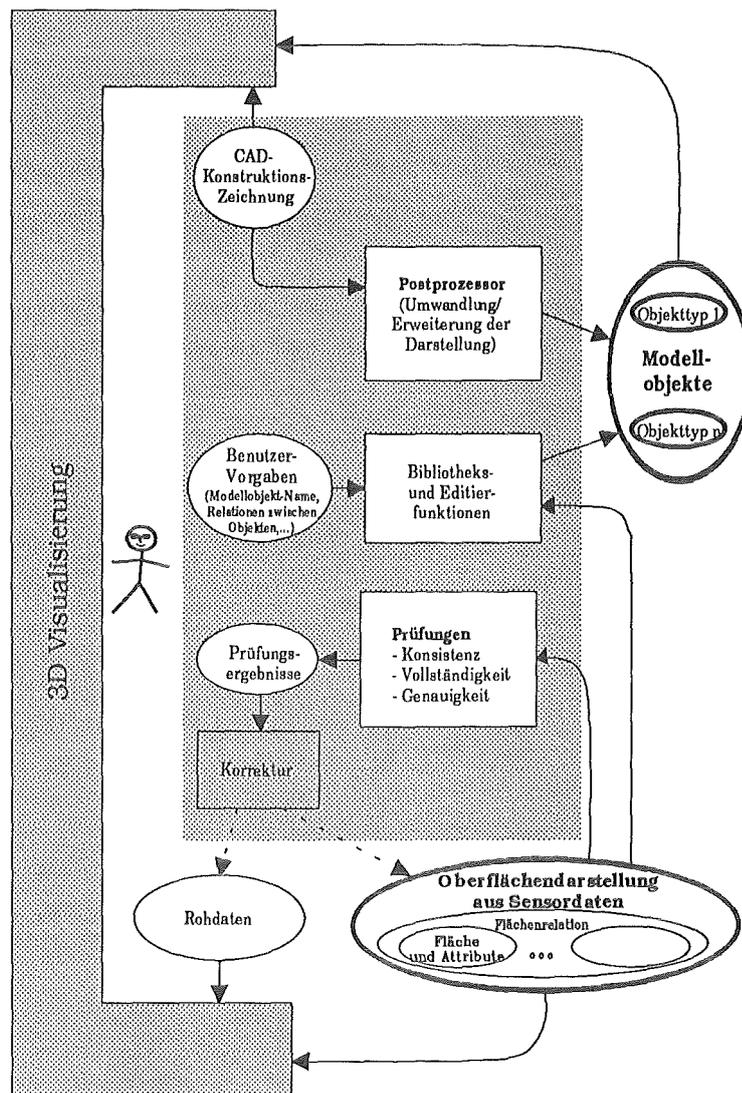


Abb. 1.9 Rekonstruktion der Modellobjekte (Verfeinerung der Instanzen Rekonstruktion und CAD-Konstruktion in Abb. 1.6)

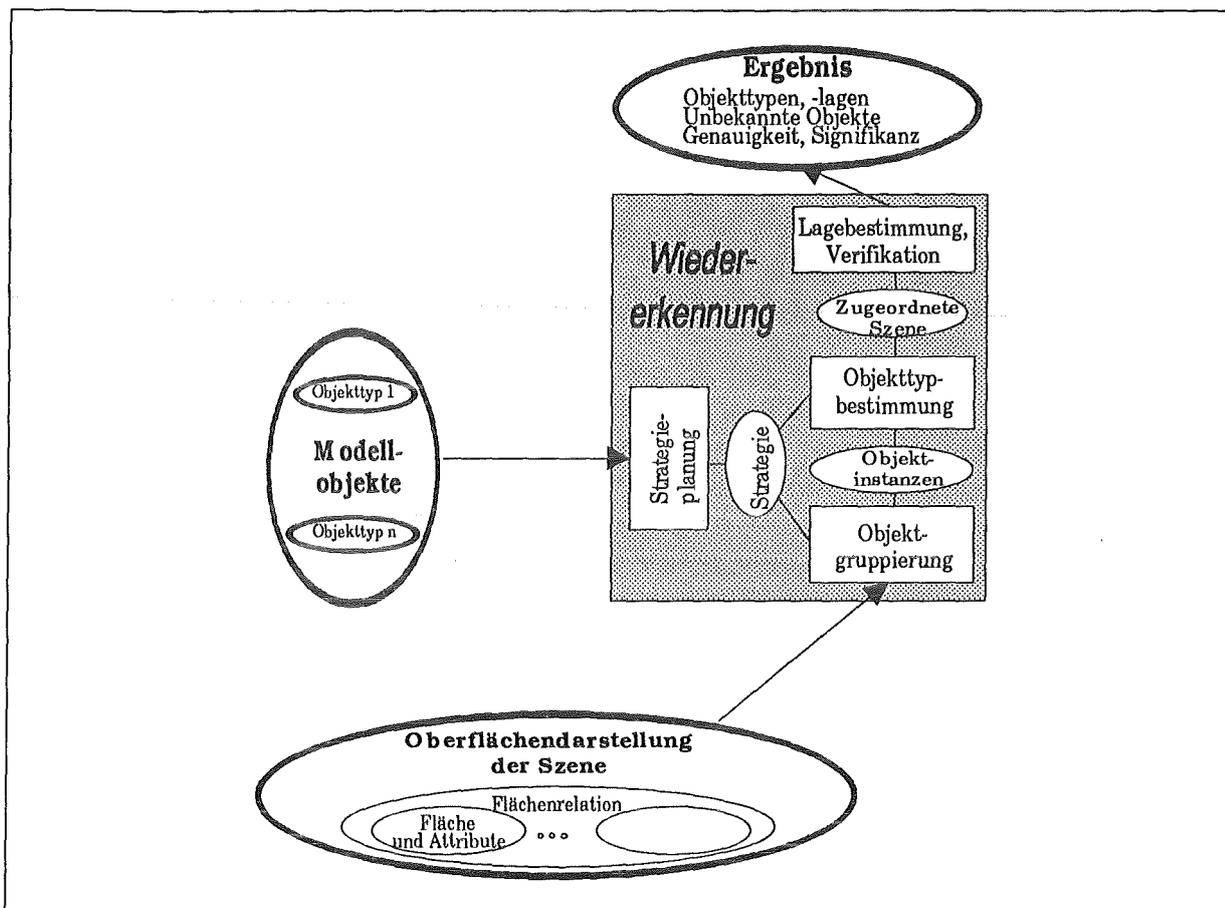


Abb. 1.10 Verfeinerung der Instanz Wiedererkennung in Abb. 1.6

Die Zuordnung der Aufgabenbereiche zu Rechnerkonfigurationen ist im Übersichtsdiagramm Abb. 1.6 durch unterschiedlich schattierte Bereiche angedeutet.

Bei der **Wiedererkennung** handelt es sich um ein Wissensverarbeitungs-, speziell um ein Instantiierungsproblem: Variablen (Szeneoberflächen) werden Werte (Modellobjektoberflächen) zugewiesen, wobei Beschränkungen durch die geometrischen Attribute und Relationen zu beachten sind. Die Anforderungen liegen hier auf flexibler Wissensrepräsentation und leistungsfähigen Verfahren zur Propagierung von Beschränkungen. Als Entwicklungsmaschine bietet sich eine KI-Arbeitsstation mit den zugehörigen Sprachen, Entwicklungswerkzeugen und Laufzeitumgebung an. Die Wahl ist aber noch offen (vgl. Kap. 3).

Im Bereich **Rekonstruktion** liegt der Schwerpunkt auf der Verwaltung und der Transformation vorwiegend koordinatenorientierter und speicherintensiver Geometriemodelle. Ein weiterer Schwerpunkt ist die 3D-Visualisierung von Zwischenresultaten aller Art. Hierfür eignet sich am besten eine CAD-Arbeitsstation. Zur Zeit laufen die Entwicklungen zur Rekonstruktion im Rahmen des Programmsystems POMOS /44/ auf Silicon Graphics PC IRIS Arbeitsstationen.

Der gesamte Bereich der **Datenvorverarbeitung** schließlich ist, neben der direkten Sensorik-Ankoppelung, vor allem gekennzeichnet durch numerische Verarbeitung sehr großer Datenmengen, wobei der Rechenaufwand den Kommunikationsaufwand dominiert. Ein Transputersystem scheint hier die richtige Wahl. Dieses dient zugleich als Versuchsfeld für hochparallele und echtzeitfähige Algorithmen zur Entfernungsbildverarbeitung, was längerfristig auch die Wiedererkennung unter anwendungsspezifischen Echtzeitbedingungen einschließen soll.

### 1.5.2 Vorverarbeitung

In Abb. 1.11 sind die Instanzen der Vorverarbeitung und der Attribut- und Relationenrechnung verfeinert dargestellt.

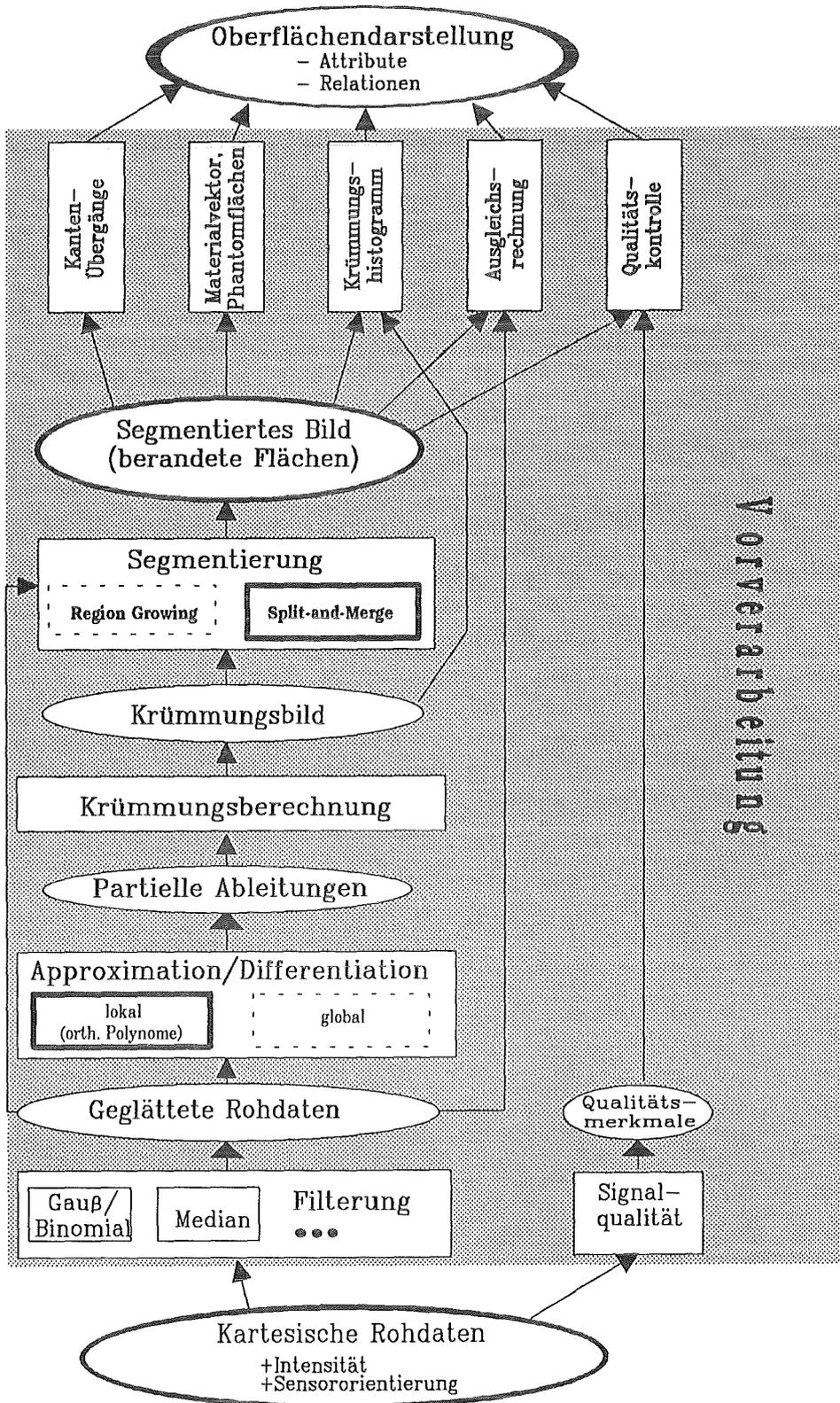


Abb. 1.11 Datenfluß zur Vorverarbeitung der Entfernungsbilder

## 1. Filterung

Im ersten Schritt werden die kartesischen Rohdaten geglättet und die Signalqualität geprüft.

Klassische Filteroperatoren im Frequenzbereich wie Tiefpaß und Bandpaß sind für geometrische Signale ungeeignet, weil sich das Frequenzspektrum eines verrauschten geometrischen Signals von dem des rauschfreien Signals nicht wesentlich unterscheidet /3/.

Bisher wurden zwei Standardfilter implementiert:

- Binomialfilter zur Unterdrückung Gauß-verteilter Störsignale
- Kantenerhaltender Medianfilter.

Der Medianfilter mit Maskengröße 5 oder 7 erwies sich dabei als wesentlich effektiver zur Unterdrückung insbesondere impulsartiger Störungen als der Binomialfilter bei gleicher Maskengröße /40/. Geometrische Feinstrukturen unterhalb der halben Maskenbreite werden durch den Medianfilter jedoch eliminiert. Noch bessere Ergebnisse könnten durch den Einsatz adaptiver Filter oder eines Medianfilters mit adaptiv eingestellter Maskenbreite erzielt werden. Ihr optimaler Einsatz erfordert jedoch, daß die Größe der kleinsten geometrischen Feinstrukturen in Relation zu den Impulsstörungen bekannt ist. Solche anwendungsabhängigen Kenntnisse über den Bildinhalt sind bei 'low-level'-Bildverarbeitungsalgorithmen unerwünscht /2/.

Die Prüfung der Signalqualität soll dazu dienen, 'pathologische', z.B. spiegelnde, durchscheinende oder vollständig lichtabsorbierende Materialoberflächen möglichst frühzeitig zu erkennen und auszusondern. Neben dem Winkel- und Entfernungswert liefert der 2D LADAR ein Intensitätsmaß für das reflektierte Licht. Dieses soll zusammen mit den kartesischen Koordinaten übertragen und ausgewertet werden, um so lichtabsorbierende Oberflächen zu erkennen. Vorversuche mit einer **bekanntem, ebenen Referenzoberfläche**, aber verschiedenen Oberflächenwerkstoffen sollen über die Verteilung der quadratischen Abstandsfehler Rückschlüsse auf die Reflexionseigenschaften der Oberfläche geben. Ein weiteres zu untersuchendes Qualitätsmerkmal ist die Varianz mehrerer gemittelter Punktmessungen: es wird erwartet, daß die Meßergebnisse für 'mixed points' oder Spiegelreflexionen weit stärker streuen als für normale Oberflächenpunkte.

## 2. Lokale Approximation und Krümmungsberechnung

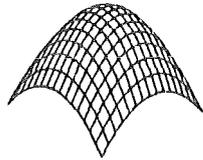
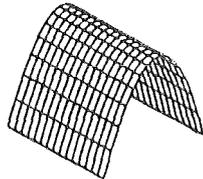
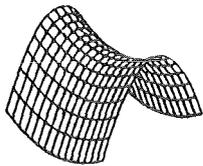
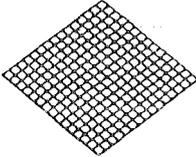
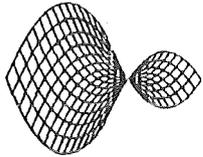
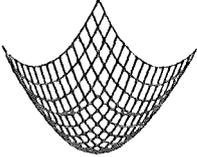
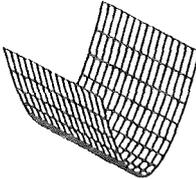
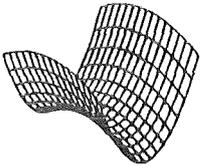
Aus den geglätteten Daten werden Krümmungsmerkmale errechnet. Sie erfüllen zwei wichtige Aufgaben:

1. Kriterien für die **Bildaufteilung** entlang der wesentlichen Konturen zu liefern (vgl. die in 1.4 genannten Kriterien für die Segmentierung, insbesondere das Uniformitätsprädikat);
2. Teiloberflächen lageinvariant mit Hilfe der Verteilung ihrer Krümmungen zu **charakterisieren** und somit wiederzuerkennen. Dazu werden in der Objektrepräsentation Krümmungshistogramme vorgesehen (2.1).

Die Gestalt allgemeiner, dreidimensionaler und stetig differenzierbarer Oberflächen läßt sich durch Krümmungsfunktionen (die 6 sog. Fundamentalformen) differentialgeometrisch eindeutig charakterisieren, welche invariant unter Translation und Rotation sind. Die **Gauß'sche Krümmung  $K(u,v)$**  und die **mittlere Krümmung  $H(u,v)$**  sind daraus abgeleitete, in der Entfernungsbildverarbeitung gekrümmter Oberflächen weit verbreitete Krümmungsmaße. Sie erlauben zwar keine eindeutige Charakterisierung mehr, sind aber dafür sogar invariant unter Änderungen der Darstellung (Parametrierung) der Oberfläche als Funktion der zwei unabhängigen Variablen  $u,v$  in Def. 1.2 /2/.

Aus den 9 möglichen Vorzeichenkombinationen von  $H$  und  $K$  lassen sich lokal für jeden Punkt 8 mögliche **diskrete Fächentypen** berechnen, deren geometrische Bedeutung in folgender Tabelle 1.3 veranschaulicht wird (die Kombination  $H=0$  und  $K>0$  kommt nicht vor):

Da die mathematischen Grundlagen und Algorithmen der differentialgeometrischen Krümmungsmerkmale ausführlich dokumentiert sind, etwa bei Besl /2/, Kap.3.2-3.5, /3/ oder Vogelgesang /64/, soll an dieser Stelle darauf verzichtet werden.

	$K > \epsilon_K$	$K = 0$	$K < -\epsilon_K$
$H < -\epsilon_H$	Spitze 	Kamm 	Sattelkamm 
$H \approx 0$	-	Ebene 	Minimum 
$H > \epsilon_H$	Grube 	Tal 	Satteltal 

Tab. 1.3 Lokale diskrete Flächentypen ('labels'), nach /3/

Speziell für eine 2½D-Oberfläche  $\underline{x}(x,y) := [x, y, z=f(x,y)]^T$  ergeben sich die Krümmungsfunktionen  $H(x,y)$ ,  $K(x,y)$  aus den Formeln

$$H = \frac{1(1+f_y^2)f_{xx} + (1+f_x^2)f_{yy} - 2f_x f_y f_{xy}}{2(1+f_x^2+f_y^2)^{3/2}}$$

und

$$K = \frac{f_{xx}f_{yy} - f_{xy}^2}{(1+f_x^2+f_y^2)^2}$$

Darin bedeuten

$$\begin{aligned} f_x &:= \frac{\partial f}{\partial x} & f_y &:= \frac{\partial f}{\partial y} \\ f_{xy} &:= \frac{\partial^2 f}{\partial x \partial y} & f_{xx} &:= \frac{\partial^2 f}{\partial x^2} & f_{yy} &:= \frac{\partial^2 f}{\partial y^2} \end{aligned}$$

die partiellen Ableitungen von  $f(x,y)$ .

Da die Differentiation diskretisierter Oberflächendaten trotz Glättung sehr rauschempfindlich ist, wird die Oberfläche zunächst **durch Polynomfunktionen approximiert** und deren Ableitungen anstelle der Originaldaten verwendet. Für die Approximation kommen in Betracht

- die lokale Approximation jedes Punktes in einer  $N \times N$ -Nachbarschaft durch orthogonale Polynome (/3/, Kap. 3.7.1)
- die globale Approximation der Oberfläche durch B-Spline-Funktionen (/3/, Kap. 3.7.2)
- die globale Approximation durch Minimierung von Energiefunktionen, die Unstetigkeiten erhält /46/.

Da bei b) Unstetigkeitsstellen im Abstandsbild durch glatte Polynomfunktionen verschliffen werden, sind Methoden a) oder c) vorzuziehen. Nach Untersuchungen von Besl liefert Methode a) einen optimalen Kompromiß zwischen Rechenaufwand und Approximationsgüte.

Orthogonale Polynome haben gegenüber allgemeinen Ausgleichsfunktionen den Vorteil, daß rechnerisch die Approximation und Ableitungsberechnung durch **Faltung** jedes Bildpunktes mit einer **konstanten, also nur einmal zu berechnenden Ableitungsmaske** bewältigt werden. Dabei wird vorausgesetzt, daß die Punkte in jeder Nachbarschaft **äquidistant** in  $x$  und  $y$  sind. Die Rohdaten sind nach der Umrechnung in kartesische Koordinaten  $(x,y,z(x,y))^T$  dagegen im allgemeinen **nicht mehr äquidistant** in  $x$  und  $y$ . Kann man weder diese, noch die Fehler einer **Interpolation** zur Berechnung eines äquidistanten Gitternetzes tolerieren, sollten **Filterung, Approximation und Krümmungsberechnung vor der Transformation in kartesische Koordinaten**, also - im Fall 2D LADAR, vgl. Anmerkungen zu Def. 1.2 - bereits bei den **Entfernungsdaten**  $(u, j, h(u,j))^T$  ansetzen. Hier besteht **Äquidistanz** bezüglich des **Ablenkwinkels**  $u$  und des **ganzahligen Meßpunktparameters**  $j$ . Da letztlich dieselbe Oberfläche dargestellt wird, folgt aus der Invarianz gegenüber Parametrierung, daß die  $K$ - und  $H$ -Werte unverändert bleiben müßten. Einen Nachteil hat dieses Vorgehen dennoch: es ist abhängig vom Entfernungssensor und der Datenaufnahme, im Gegensatz zum sensorneutralen kartesischen Datenformat.

Tests an realen Szenen ergaben, daß  $K$  und  $H$  trotz vorangegangener Median-Filterung und lokaler Approximation sehr rauschempfindlich sind /40/. Die Darstellung der zu einem einzigen skalaren Merkmal  $\Psi(x,y):=a_1K^2+a_2H^2$  kombinierten Krümmung für die Szene "Kiste und Tonne" in Abb. 1.12a zeigt dies ebenso deutlich wie die Darstellung des diskreten Flächentyps "Kamm" in Abb. 1.12b. Verhältnismäßig große Krümmungswerte und parallel verlaufende Kämmen auf der eigentlich ebenen, aber noch Rauschteile enthaltenden Kisten- bzw. Tischoberfläche stehen wider Erwarten kleine Krümmungen und durchbrochene Kämmen an hohen Körperkanten (Quantisierungssprüngen) gegenüber.  $K$  und  $H$  eignen sich also nicht ohne weiteres für eine Bildsegmentierung oder Kantenextraktion. Schuld daran ist sowohl die Definition von  $K$  und  $H$  als solche, insbesondere das Verhalten der Nennerfunktionen für große Ableitungen  $f_x, f_y$ , als auch die Polynomapproximation vom Grad 2. Dies, so haben vergleichende Analysen für verschiedene synthetische Szenen ergeben /40/, scheint der Preis für die Invarianzeigenschaften dieser Merkmale zu sein. Andere Merkmale, wie das Ableitungsmerkmal

$$\sqrt{g(x,y)}:=\sqrt{1+f_x^2+f_y^2} \quad (\text{metrische Determinante, in Abb. 1.13a für dieselbe Szene dargestellt}),$$

oder das Krümmungsmerkmal

$$Q(x,y):=f_{xx}^2+2f_{xy}^2+f_{yy}^2 \quad (\text{quadratische Variation, Abb. 1.13b})$$

sind robuster gegenüber Rauschen und geeigneter zur Konturerkennung. Neben der Tisch- und Zylinderoberfläche erkennt man in Abb. 1.13b auch deutlich die Begrenzungen der Abschattungsflächen (Phantomflächen). Die Merkmale  $Q$  und  $\sqrt{g}$  eignen sich aber nicht ohne weiteres zur **Charakterisierung** gekrümmter Oberflächen, da ihre Werte sich mit dem Blickwinkel ändern.

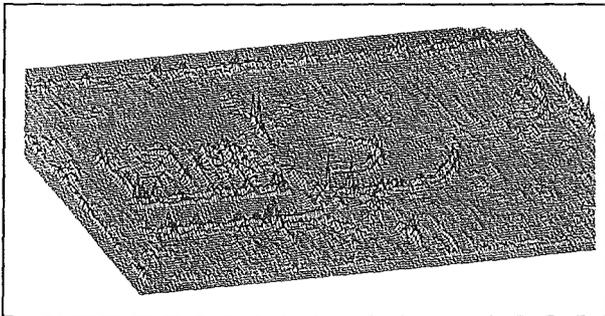


Abb. 1.12a: Krümmungsmerkmal  $\Psi$  für Szene "Kiste und Tonne"

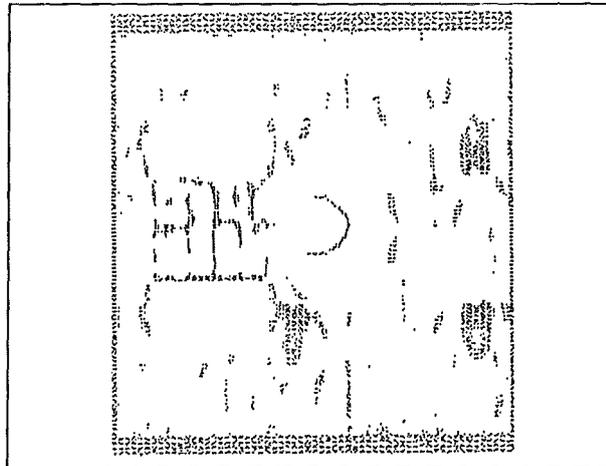


Abb. 1.12b: Flächentyp Kamm

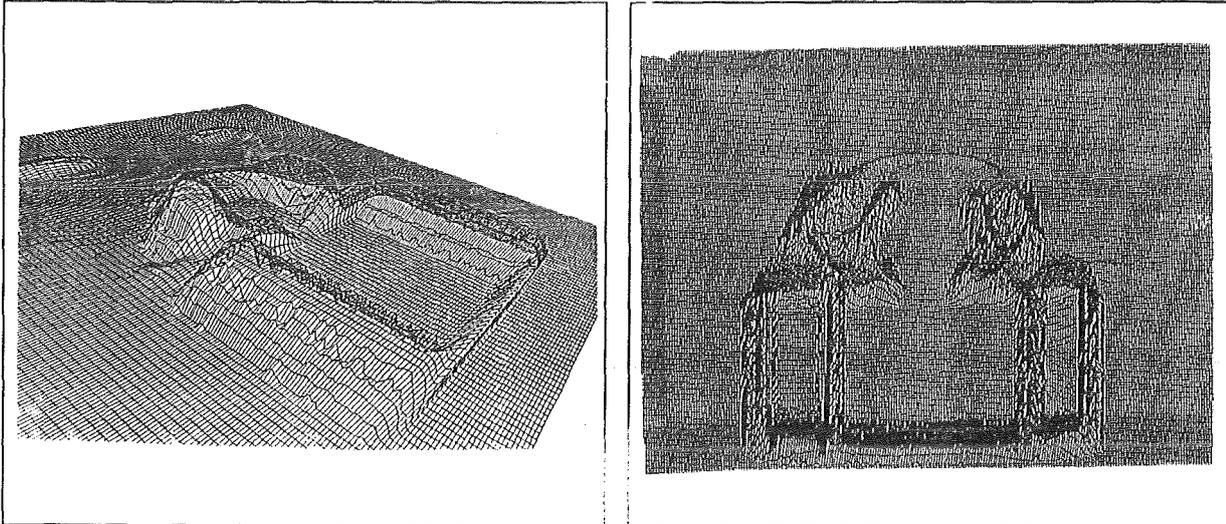


Abb. 1.13a: Ableitungsmerkmal  $\sqrt{g}$  für Szene "Kiste und Tonne"

Abb. 1.13b: Krümmungsmerkmal  $Q$

### 3. Segmentierung

Aufgabe der Segmentierung ist die Zerlegung des Entfernungsbildes in disjunkte, berandete Regionen, die die Oberflächen bilden. Unter dem Ergebnis, dem **segmentierten Teilbild**, hat man sich im wesentlichen vorzustellen

- eine **Liste von Regionen**, jede charakterisiert durch
  - Nummer, Bezeichner o.ä.
  - eine oder mehrere geschlossene Kurven, die Berandung(en).

Zur Information einer Berandung gehören die Arten der Übergänge, die Namen der angrenzenden Regionen und die Geometriedaten der Berandung.

- die i.w. um die **Zugehörigkeit** jedes Punktes zu einer Region erweiterten Rohdaten.

Für die Segmentierung sind 2 Klassen von Verfahren bekannt. Es muß in jedem Fall ein Kriterium definiert werden, wann Bildpunkte zur selben und wann zu verschiedenen Fläche gehören (Uniformitätsprädikat  $P$  in 1.4).

#### (a) Region Growing (Flächenwachstum)

*Minimale* Flächen (sog. Saat-Regionen) werden solange *ausgedehnt*, wie  $P$  erfüllt bleibt [2] [68].

Besl beschreibt in seiner Arbeit [2] ein region growing Verfahren, bei dem kleine, im Innern größerer, zusammenhängender Bereiche einheitlichen Flächentyps liegende Punktemengen die Saatregionen bilden. Durch die Punkte jeder Saatregion wird eine Ausgleichsfläche variabler Ordnung ( $<5$ ) gelegt. Die Saatregion wächst in einem iterativen Prozeß solange, bis selbst die Approximation höchster Ordnung keinen tolerierbaren Ausgleichsfehler mehr besitzt. Die Vorzeichen der Ausgleichsfehler und ihre Verteilung dienen als Entscheidungskriterium, wann die Ordnung der approximierenden Funktion erhöht wird.

Sobald das Wachstum einer Fläche beendet ist, folgt ein abschließender Akzeptanztest. Angrenzende oder überlappende Flächen werden in einem Nachbearbeitungsschritt vereinigt, wenn sie hinreichend glatte Übergänge bilden.

Das Verfahren erfordert die 2½D-Eigenschaft. Insbesondere sind als approximierende Funktionen nur Polynome max. 4. Grades der Form  $z=g(x,y)$  zugelassen, und nicht z.B. Quadriken  $g(x,y,z)=0$ , die auch Kugel- oder Ellipsoidoberflächen modellieren können.

Das Verfahren ist - vor allem wegen der wiederholten Ausgleichsrechnung mit Polynomen variabler Ordnung - algorithmisch sehr aufwendig und auch nicht ohne weiteres parallelisierbar; zumindest ist keine parallele Version bekannt.

Für viele, darunter auch unregelmäßig gekrümmte und konkave Werkstücke gibt Besl optisch perfekte Segmentierungsergebnisse an. Dabei spielt die richtige Einstellung vieler Verfahrensparameter (Wahl der Saatregionen, Ausgleichsfehler, Terminierungskriterien, Vereinigungskriterien etc.) eine entscheidende Rolle. Die Wahl dieser Parameter erfolgt empirisch, zum Teil sogar anwendungsabhängig, und setzt genaue Kenntnis der Meßfehler in den Rohdaten voraus. Eine ungünstige Wahl der Parameter oder der approximierenden Funktionen liefert, vor allem bei stark verrauschten Meßdaten, kleine, unregelmäßig berandete und optisch wenig aussagekräftige Flächenstücke. Auch die Vereinigung kann dann daran nichts Grundsätzliches mehr ändern.

#### (b) Split-and-Merge (Teilen und Vereinigen)

Zuerst wird das Bild solange unterteilt, bis alle Teilflächen  $P$  erfüllen.

Dann werden angrenzende Teilflächen wiedervereinigt, solange  $P$  erfüllt bleibt /6/ /59/.

Der Teilungsschritt ist eine hierarchische Tetraederdekomposition. Sie setzt ebenfalls eine 2½D-Oberfläche voraus. Man wählt 3 Eckpunkte  $P_1, P_2, P_3$  auf der Oberfläche, die nicht auf einer Geraden liegen, und untersucht die Abstände aller durch  $P_1, P_2$ , und  $P_3$  begrenzten Bildpunkte von der durch diese Punkte führenden Ebene  $E$ . Liegen die Punkte alle innerhalb eines Toleranzbandes  $\pm \epsilon$  um  $E$ , so liegt eine einzige, dreieckförmige, annähernd ebene Fläche vor, die  $P$  erfüllt. Andernfalls wird der Bereich innerhalb  $P_1, P_2, P_3$  an dem Punkt  $P$  unterteilt, der den maximalen Abstand von  $E$  besitzt, und der Algorithmus rekursiv auf die Unterbereiche

$$\langle P_1, P_2, P \rangle \quad \langle P_2, P_3, P \rangle \quad \langle P_1, P_3, P \rangle$$

angewandt. Ergebnis dieser Aufrufe sind Flächen mit dreieckförmigem, allgemeiner: polygonalem Rand.

Angenommen, 2 dieser Flächen aus verschiedenen Bereichen,  $F_0$  und  $F_1$ , mit

$$\text{Rand}(F_0) = \overline{R_{01} \dots R_{0i} \overline{RR'} R_{0i+1} \dots R_{0n}}$$

$$\text{Rand}(F_1) = \overline{R_{11} \dots R_{1j} \overline{RR'} R_{1j+1} \dots R_{1m}}$$

grenzen entlang  $\overline{P_1P}$ ,  $\overline{P_2P}$  oder  $\overline{P_3P}$  aneinander und haben den Kantenzug  $\overline{RR'}$  gemeinsam, so werden sie zu einer neuen, bereichsübergreifenden Fläche  $F$  mit

$$\text{Rand}(F) := \overline{R_{01} \dots R_{0i} \overline{RR'} R_{1j} \dots R_{11} R_{1m} \dots R_{1j+1} \overline{R'} R_{0i+1} \dots R_{0n}}$$

vereinigt, wenn folgende Bedingungen erfüllt sind:

- der Kantenzug  $\overline{RR'}$  ist weder Sprung-, noch Dachkante mit markantem Knickwinkel
- er trennt keine Gebiete von unterschiedlichem Krümmungstyp (z.B. "Spitze" und "Sattel")
- der Krümmungsbetrag ändert sich auch nicht über ein bestimmtes Maß hinaus.

Gekrümmte Oberflächen, zum Beispiel eine Kugeloberfläche, zerlegt die Tetraederdekomposition so erst in viele kleine, ebene Facetten, um sie anschließend wiederzuvereinigen. Um Teilflächen mit homogenem Krümmungsverhalten als Ganzes zu erhalten, kann man die Dekomposition, statt auf das Entfernungsbild  $(x, y, f(x,y))$ , auf das Krümmungsbild  $(x, y, k(x,y))$  ansetzen, wobei  $k(x,y)$  einer der unter 2. definierten Krümmungswerte ist. Für alle ebenen Flächen sind die Krümmungswerte nahe 0, liegen also von Anfang an innerhalb des Toleranzbandes um die Ebene  $z=0$ . An Sprungkanten, sowie an konkaven oder konvexen Dachkanten besitzt das Krümmungsbild lokale Extrema, welche Flächenteilungspunkte bilden. Die Arbeitsweise des Algorithmus wird in Abb. 1.14 an einer sehr einfachen Szene veranschaulicht. In der Praxis

liegt das Hauptproblem der Dekomposition im Krümmungsraum, wie gesagt, in der hohen Rauschempfindlichkeit der bisher von uns untersuchten Krümmungsmerkmale.

Das split-and-merge-Verfahren ist sehr effizient (hierarchisch) parallelisierbar, vgl. 1.5.3. Seine konzeptionelle **Einfachheit und leichte Parallelisierbarkeit** bilden den Hauptvorteil gegenüber dem region growing.

Als weiteren Vorteil sehen wir, daß die Flächen - als Vereinigung dreiecksförmiger Elementarflächen - einfache, **regelmäßige Berandungen** (Polygonzüge) haben. Mit der top-down-Dekomposition in Flächen wird automatisch auch deren Berandung nur so fein approximiert, wie das Toleranzbandkriterium es erfordert, im Gegensatz zu der bottom-up-Vorgehensweise beim Flächenwachstum, die eher zu unregelmäßigen, diffusen Berandungen neigt, weil die Regionen 'pixelweise' in mehrere Richtungen wachsen.

#### 4. Oberflächendarstellung (Attribut- und Relationenrechnung)

Im letzten Schritt wird aus dem segmentierten Abstandsbild die endgültige Oberflächendarstellung berechnet:

- Zunächst werden Flächen mit einem signifikanten Anteil von Punkten schlechter **Signalqualität**, z.B. hoher Streuung oder geringer Intensität, **ausgesondert**.
- Alle annähernd **parallel zur Blickrichtung** des Sensors liegenden Flächen werden **ausgesondert**. Oft handelt es sich hierbei um Abschattungsflächen (**Phantomflächen**, engl. limbs), welche gar nicht real existieren (vgl. Abb.1.3 und Abb. 1.14). In jedem Fall sind Flächen unter Streiflicht nicht gut erkennbar und ihre Punkte stark verrauscht.  
Für alle anderen Flächen wird ein von der Fläche wegweisender **Materialvektor** berechnet, z.B. der negative Blickrichtungsvektor eines Punktes im Innern oder - für ebene Flächen - der Normalenvektor.
- Die **Nachbarschaftsrelationen** zwischen den Flächen werden bestimmt. Zwei Flächen sind benachbart, wenn sie einen Teil ihrer Berandung gemeinsam haben. Die **Kantenübergänge** zwischen diesen Flächen werden nach Sprungkanten, konvexe/konkave Dachkanten und sonstigen Übergängen klassifiziert.
- Die Punkte jeder Fläche werden durch eine **analytische Funktion** approximiert. Hier können ähnliche Techniken wie bei POMOS verwendet werden, z.B.
  - Ausgleichsrechnung mit impliziten Polynomfunktionen vom Grad 2 (Quadriken) oder 4
  - Approximation durch nicht-uniforme rationale B-Splines (NURBS /22/).

Die Rohdaten sind damit durch einen einzigen Parametersatz beschrieben. Die Ausgleichsfehler können groß werden, da das Kriterium für die Flächenbildung nicht die Approximierbarkeit, sondern geometrische Kriterien sind. Eine weitere, interne Zerlegung in Teilflächen kann notwendig sein, um die Approximationsgüte zu verbessern.

- Die Verteilung der Krümmungswerte auf jeder Fläche wird durch **Histogramme** dargestellt. Im Gegensatz zur Parameterdarstellung der Approximationsfunktion ist das Krümmungshistogramm invariant unter Translation und Rotation. Es kann zur effektiven Charakterisierung und Wiedererkennung auch verdeckter, gekrümmter Oberflächen verwendet werden, selbst wenn diese mathematisch schlecht approximierbar sind.  
Für die Histogramme sollten die Punkte auf oder in unmittelbarer Nähe des Flächenrandes ausgelassen werden, insbesondere bei unstetigen Übergängen, weil hier - durch die lokale Approximation bedingt - die Werte und Vorzeichen der Krümmungen wenig aussagekräftig sind.

Aus diesen Grundgrößen werden weitere Flächenattribute und weitere Flächenrelationen berechnet (vgl. Kap. 2).

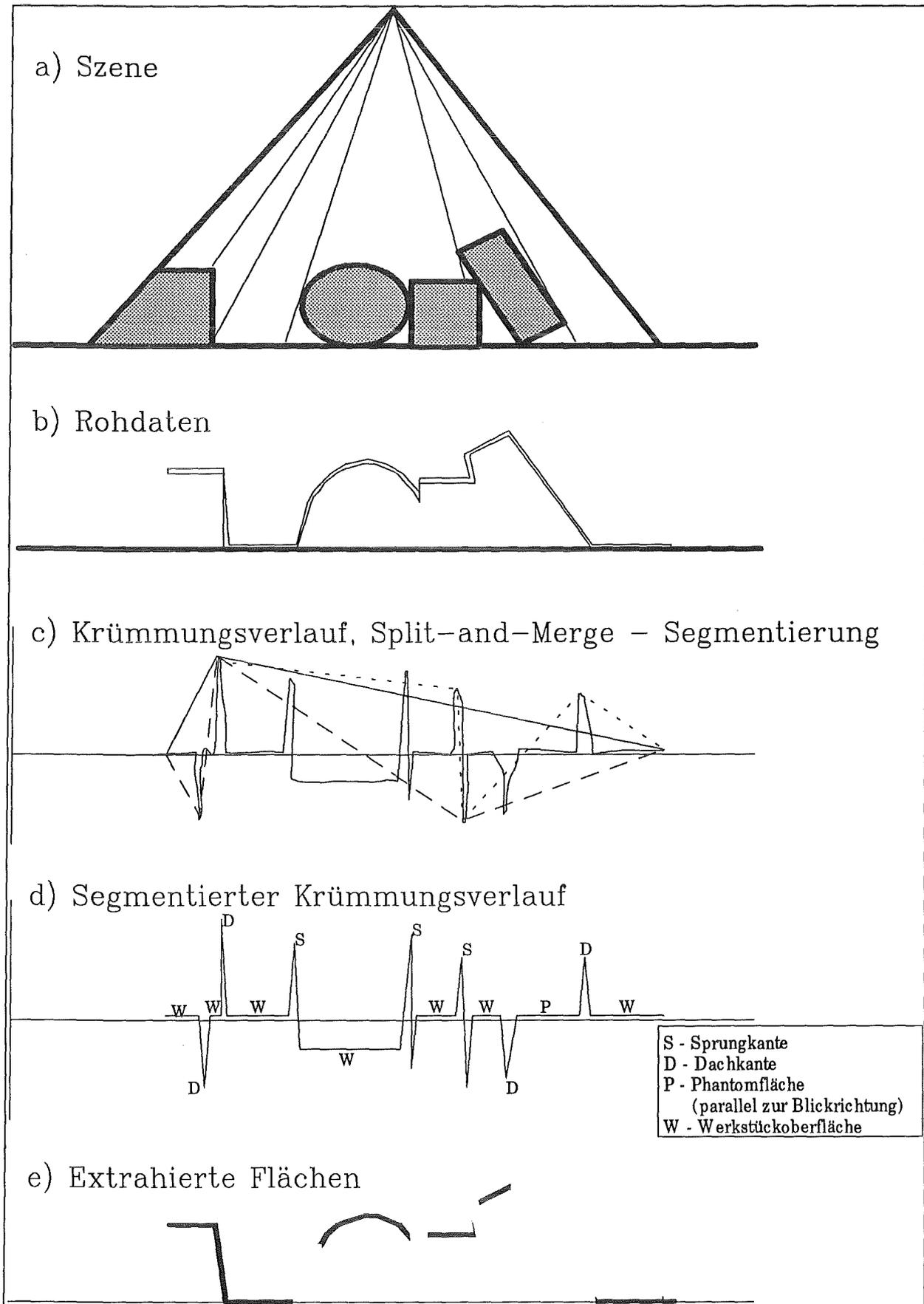


Abb. 1.14 Prinzipielle Arbeitsweise der Vorverarbeitung

### 1.5.3 Implementierung auf Parallelrechnern

Längerfristig sollen die Verfahren zur Entfernungsbildverarbeitung in einer übergeordneten Fahrzeug- oder Robotersteuerung in Echtzeit ablaufen. Eine notwendige, allein natürlich nicht hinreichende Voraussetzung ( $\rightarrow$ zeitlicher Determinismus!) ist der Einsatz massiv paralleler Rechensysteme. Da es sich um rechenintensive, numerische Anwendungen mit begrenzter Kommunikation handelt, bilden Transputer eine gute Wahl.

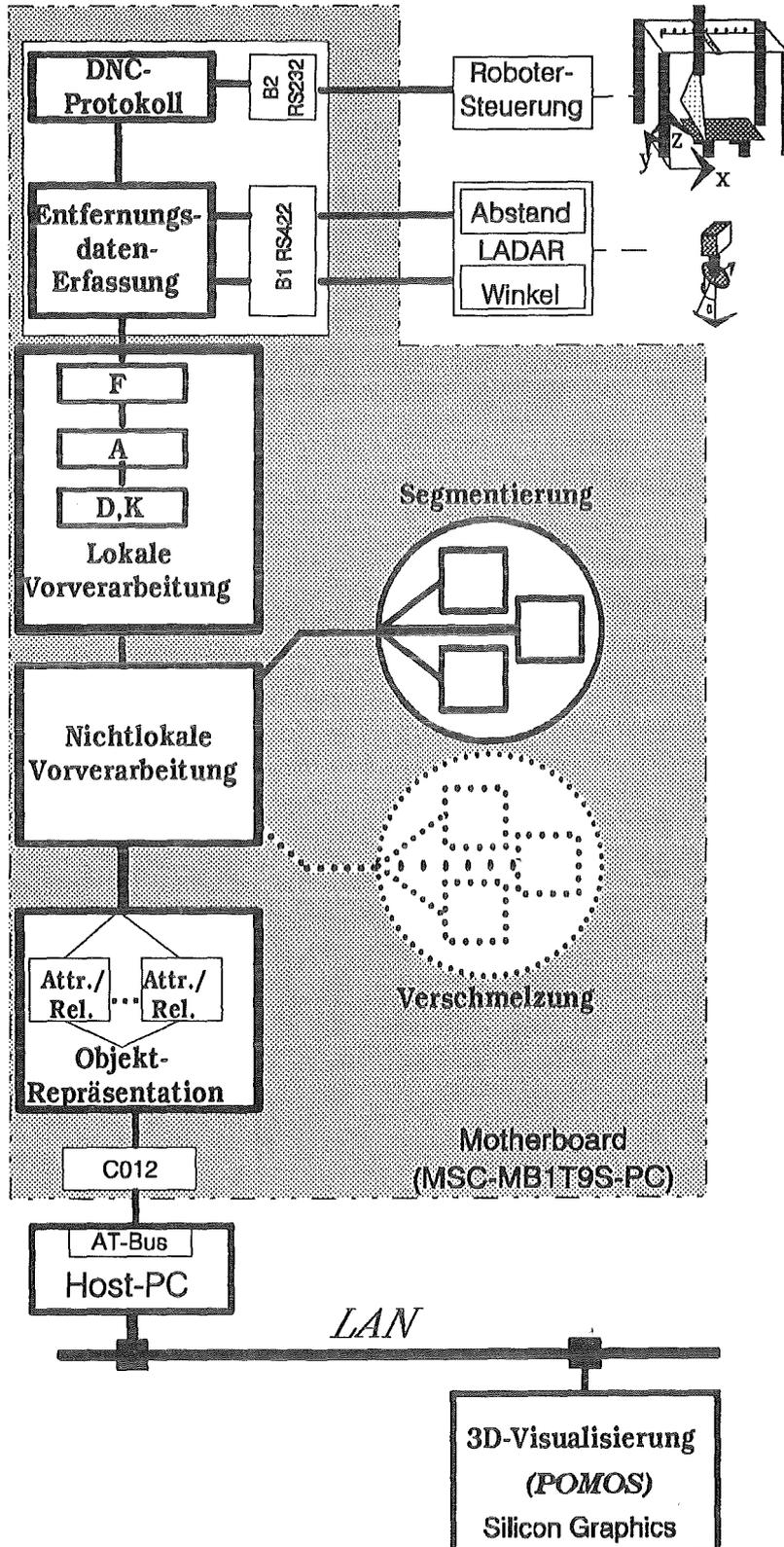


Abb. 1.15

Transputernetz zur Vorverarbeitung

Als Hardware-Plattform fungieren zur Zeit PC-Motherboards (PC-Einsteckkarten MB1T9S, kompatibel zu INMOS B008), die mit 9 Transputermodulen (TRAM's, bei uns vorwiegend T805 mit bis zu 4MB RAM) bestückt werden können. Die Netztopologie ist mit Hilfe der Module Motherboard Software (S708) konfigurierbar. Als Programmiersprache wird derzeit OCCAM 2 /33/ ohne Netzwerk-Betriebssystem eingesetzt. In Zukunft sollen T9000-basierende Systeme im 19" Einschub zum Einsatz kommen, welche neben den Transputer Links über einen schnellen I/O-Bus zur Versorgung der Transputer mit Initialisierungsdaten verfügen (broadcast-Verfahren).

Der optimale Einsatz einer prozeßorientierten Hochsprache erfordert, daß Datenstrukturen und Algorithmen bereits im Entwurf für die Parallelverarbeitung ausgelegt werden.

Die 2 Transputer in Abb. 1.15 oben, die die Entfernungs- und die Roboterkoordinaten erfassen und LADAR und Roboter synchronisieren, wurden in 1.2.1 (Abb. 1.2) bereits erläutert.

Die nachfolgenden Vorverarbeitungsschritte - Filterung, Approximation, Ableitungs- und Krümmungsberechnung - operieren alle auf lokalen Nachbarschaften der Maskenbreite  $N \times N$  (**lokale Vorverarbeitung**). Eine parallele Bearbeitung disjunkter Teilbilder (geografische Parallelisierung) ist daher immer möglich. Da die Rohdaten **zeilenweise** und relativ langsam mit max. 8 Zeilen (Höhenprofilen) pro sec anfallen, können sie sogar **on-line** verarbeitet werden, wobei die Prozesse der Verarbeitungskette vorteilhaft im **Pipeline-Betrieb** arbeiten, indem jeder Prozeß über einen Ringpuffer der  $N$  letzten Höhenprofile verfügt.

Weder eine on-line Verarbeitung noch eine disjunkte Teilbildzerlegung funktionieren im allgemeinen bei der **Segmentierung**. Der in 1.5.2 beschriebene split-and-merge-Algorithmus setzt das **gesamte Bild**, also **globale Information** bereits voraus, um den ersten Teilungspunkt  $P$  zu bestimmen. Einmal gefunden, können die Unterbereiche  $\langle P_1, P_2, P \rangle$ ,  $\langle P_2, P_3, P \rangle$ ,  $\langle P_1, P_3, P \rangle$  aber sehr wohl parallel weiterzerlegt werden, und dies rekursiv für deren Unterbereiche u.s.w. ("Verteilen und Herrschen"). Die baumartig anwachsende Menge von Zerlegungsaufträgen wird nach dem Farmer-Worker-Prinzip in einer Warteschlange verwaltet und auf freie Prozessoren verteilt, deren Ergebnisse, die Teilungspunkte, zu neuen Aufträgen führen.

Dieses Grundprinzip besitzt jedoch einige Nachteile:

1. Zu Beginn sind wenige Prozessoren mit großen Aufträgen ausgelastet.
2. Gegen Ende bearbeiten viele Prozessor kleine Aufträge, deren hohe Ankunftsrate an Rückmeldungen den Farmer selbst zum Engpaß macht.

Problem 1 vermeiden wir, indem wir den Zerlegungsauftrag, also im wesentlichen die Abstandsmaximum-Bestimmung, selbst parallelisieren und dazu einen großen Bereich in mehrere gleichgroße Unterbereiche aufspalten. Für Problem 2 führen wir einen Schwellwert, die **Paketgröße  $p$**  ein:

- Bereiche mit mehr als  $p$  Punkten werden wie bisher als Zerlegungsaufträge bearbeitet, die als Ergebnis nur einen Teilungspunkt zurückmelden, der den Farmer veranlaßt, neue Unteraufträge zu verteilen
- Bereiche mit weniger als  $p$  Punkten werden vollständig segmentiert; der Auftragnehmer (Worker) meldet die fertigen Flächen zurück. Dies belastet den Farmer nicht mehr.

Abb. 1.16 zeigt Antwortzeitmessungen für verschiedene Transputerkonfigurationen. Erst ab **2000** Punkten (Koordinatensätzen) bringt die Parallelisierung überhaupt einen Effizienzgewinn; unsere 2D LADAR Entfernungsbilder enthalten zwischen 20000 und 60000 Punkten. Für 20000 Punkte ergeben sich **Beschleunigungen der Antwortzeit von ca. 3** gegenüber einem sequentiellen split-and-merge-Algorithmus ohne jeden Verwaltungsaufwand auf einem einzigen T805 Transputer gleicher Leistung (25MHz). Das Parallelisierungspotential wird aber erst für **Paketgrößen ab 500** ausgeschöpft. Ab **4 Arbeitsprozessoren pro Farmer** tritt bereits eine **Sättigung** ein.

Eine weitere Beschleunigung ist durch eine hierarchische Struktur zu erzielen, in der die Farmer selbst (bis zu 4) Farmer beschäftigen (Abb. 1.17).

Dies vermeidet die Überlastung des einzelnen Farmers durch Rückmeldungen zu vieler Worker und reduziert indirekt den Aufwand für Routing und Multiplexing der Nachrichten. Andererseits produziert diese Lösung, wie jede organisatorische Hierarchie, selbst einen gewissen Mehraufwand an Verwaltung.

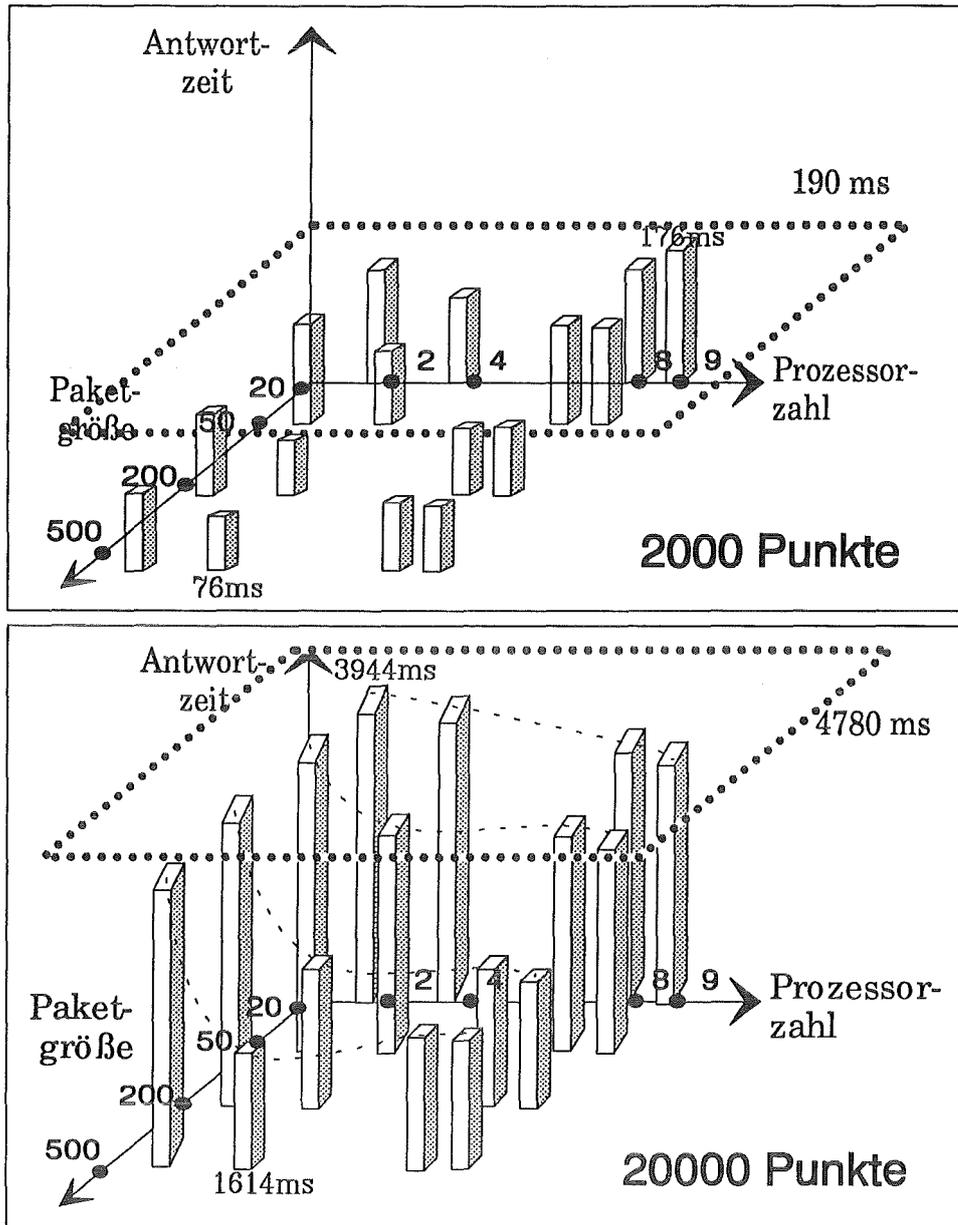


Abb.1.16 Antwortzeiten für die Flächenzerlegung bei verschiedenen Transputer-Konfigurationen und Paketgrößen

Im Gegensatz zur globalen, bildüberspannenden Segmentierung ist das letzte Glied der Verarbeitungskette, die **Attribut- und Relationenberechnung**, wiederum problemlos zu parallelisieren: Das Entfernungsbild ist bereits in disjunkte Flächen zerlegt (-> geografische Parallelisierung), und auf jeder Fläche können verschiedene Verfahren aus identischen Eingangsdaten disjunkte Attribute und Relationen als Ergebnisse errechnen (-> funktionelle Parallelisierung). Die Beschleunigung wird hier eher durch die Prozessoranzahl als durch das Problem begrenzt.

Die **Verschmelzung** mehrerer Teilansichten zu einer Gesamt-Oberflächendarstellungen ist ebenfalls sehr gut parallelisierbar:

- alle Teil-Darstellungen werden parallel aus ihren jeweiligen Sensordaten berechnet
- das Ergebnis der Verschmelzung von Flächenpaaren ist **unabhängig von der Reihenfolge der einzelnen Verschmelzungsoperationen**. Diese Operationen können daher ebenfalls parallel ausgeführt werden. Das Abbruchkriterium - keine Redundanz mehr in den Flächendarstellungen vorhanden - muß global und verteilt ausgewertet werden.

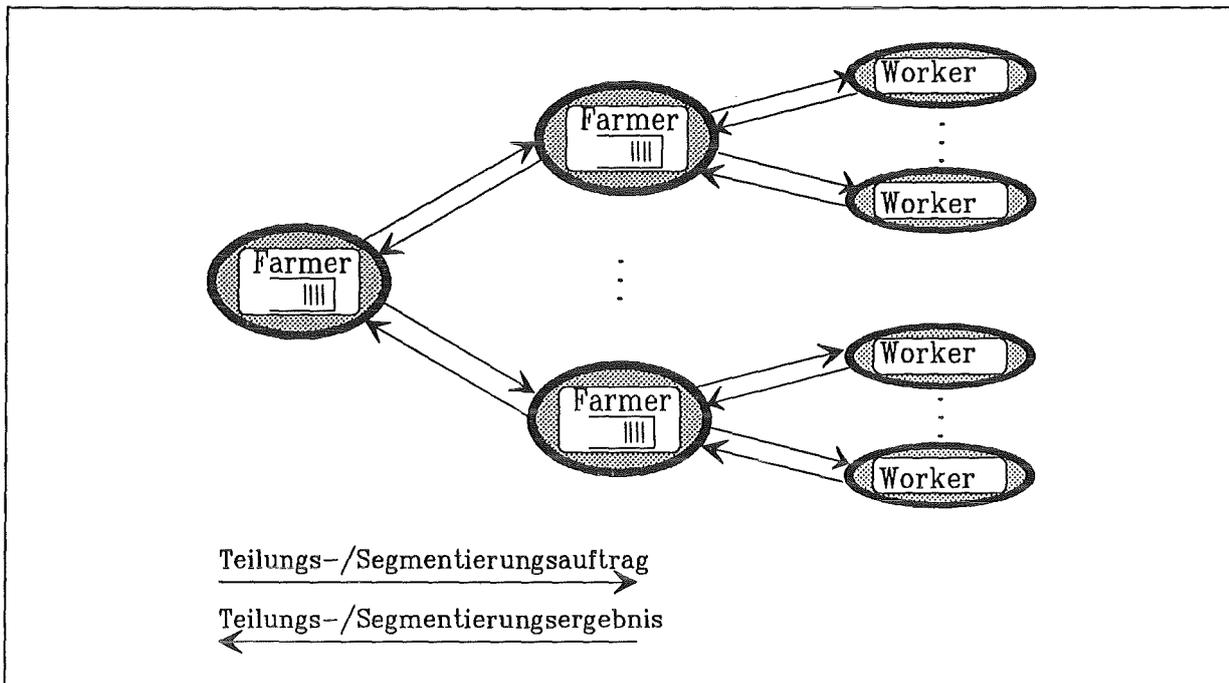


Abb. 1.17 Hierarchisches Farmer-Worker-Konzept

Zusammenfassend werden also folgende Grundmuster der Parallelverarbeitung immer wieder benutzt:

- geografische Parallelisierung (gleiche Funktion auf disjunkten Bildbereichen)  
(lokale  $(N \times N)$ -Vorverarbeitung, Flächenattribute, Teilansichten)
- Pipeline-Parallelität  
(zeilenweise Datenerfassung und Vorverarbeitung)
- Hierarchische Parallelisierung  
(Segmentierung)
- Funktionelle Parallelisierung (disjunkte Ergebnisse aus identischen Eingangsdaten)  
(Attribut- und Relationenrechnung)
- Reihenfolgeunabhängige Elementaroperationen auf gemeinsamem Datennetz  
(Verschmelzung von Teilansichten)

Oft läßt sich eine Operation in mehrere gleichartige Teiloperationen kleineren Umfangs zerlegen. Doch sind diese Teilprobleme im allgemeinen weder logisch unabhängig noch im Datenumfang gleichgroß. Dies ist nur bei den Operationen auf lokalen Nachbarschaften wie Filterung oder Krümmungsberechnung der Fall. Der zeitliche Ablauf läßt sich daher selten so organisieren, daß die gleiche Funktion auf allen Prozessoren auch **zur gleichen Zeit** ablaufen kann. Ein asynchrones, grobkörnigeres Parallelisierungsmodell (MIMD, **multiple instruction, multiple data**) ist hier universeller einsetzbar als ein getaktetes und feinkörniges wie SIMD (**single instruction, multiple data**).

## 2 Grobkonzept zur Objektwiedererkennung

In diesem Kapitel wird der aktuelle Stand der Überlegungen zur Objektwiedererkennung grob dargelegt, und die wichtigsten Datenstrukturen und Verfahren werden beschrieben.

### 2.1 Objektrepräsentation

Gesucht ist eine einheitliche oberflächenorientierte Darstellung für die zu interpretierenden Szenen und die Modellobjekte.

Alle Szenen und Modellobjekte bestehen aus **geometrischen Grundelementen**, den **Oberflächen**, und deren **Beziehungen (Relationen)**. Kanten und Ecken sind spezielle Relationen zwischen Oberflächen.

#### 2.1.1 Geometrische Grundelemente

##### 2.1.1.1 Attribute

Jede Oberfläche  $F$  ist gegeben durch

- **Benennung**  $F.id$
- **Definierende Attribute**
  - (höchstens) eine **äußere Berandung**  $F.bdo$ .

$F.bdo$  ist durch Stützstellen  $P_1, \dots, P_n \in \mathbb{R}^3$  gegeben, die einen geschlossenen Polygonzug  $P_1 \rightarrow \dots \rightarrow P_n \rightarrow P_1$  bilden. Die Segmentierung liefert immer eine diskrete Menge von Stützstellen als Flächenberandung; die Darstellung sollte es aber erlauben, den Rand auch durch stückweise glatte Kurven, z.B. Spline-Funktionen oder Kreisbögen, zu approximieren.

- $k \geq 0$  **innere Berandungen**  $F.bdi [j]$  ( $j=1..k$ )

die ebenfalls geschlossene Polygonzüge bilden.

Die Unterscheidung zwischen äußerer und innerer Berandung ist topologisch zu verstehen: die äußere Berandung umschließt die gesamte Fläche, eine innere Berandung ist ein Loch in der Fläche, das keinen Punkt der Fläche enthält.

- **Approximierende Funktion**  $F.funct$ , **Approximationsfehler**  $F.err$ .

Für  $F.funct$  kommen Polynomfunktion niederen Grades in Frage:

- implizite Form  $g(x, y, z) := \sum_{i+j+k \leq n} a_{ijk} x^i y^j z^k = 0$   $n=2$  (Quadrik) oder  $n=4$

- Darstellung durch B-Spline-Funktionen

$$\begin{aligned}
 x &= x(u, v) := \sum_{i \leq 3} \sum_{j \leq 3} a_{xij} u^i v^j \\
 y &= y(u, v) := \sum_{i \leq 3} \sum_{j \leq 3} a_{yij} u^i v^j \\
 z &= z(u, v) := \sum_{i \leq 3} \sum_{j \leq 3} a_{zij} u^i v^j \quad (u, v \text{ aus geeigneten reellwertigen Intervallen})
 \end{aligned}$$

Als Approximationsfehler  $F.err$  wird der **mittlere Abstand** aller Oberflächenpunkte von der analytischen Oberflächenapproximation angegeben. Ferner wird vermerkt, ob die Approximation als Ganzes **brauchbar** ist oder nicht ( $F.valid$ ). Ist die Ordnung oder

Funktionsklasse der approximierenden Funktion unzureichend zur Modellierung der Fläche, so ist dies meist an der Verteilung und den Vorzeichen der Punktabstände von der approximierenden Fläche erkennbar.

- **Materialvektor  $F_m$** ,  
welcher vom Material weg und zum Sensor hin zeigt.

• **Weitere Attribute zur Erkennung**

- **Histogramme der wichtigsten Krümmungsmerkmale**, z.B. der Gauß'schen und mittleren Krümmung  $F_{h_H}$ ,  $F_{h_K}$ , und ihre Träger  $tr(h_H)$ ,  $tr(h_K)$ .  
Für eine Funktion  $f: \mathbb{R} \rightarrow \mathbb{R}$  sei der Träger  
 $tr(f) := \{x : |f(x)| > \varepsilon\} \subseteq \mathbb{R}$  mit einer Toleranzschwelle  $\varepsilon > 0$  definiert.  
Der Träger der Krümmungshistogramme ist also die Menge der Krümmungswerte, die mit nennenswerter Häufigkeit vorkommen.

Die Krümmungshistogramme und ihre Träger charakterisieren gekrümmte Flächen lageinvariant. Für folgende einfache Grundkörper ergibt sich (vgl. Abb. 2.1):

Ebene:	$tr(h_K) := [0]$ ,	$tr(h_H) := [0]$	
Zylinder:	$tr(h_K) := [0]$ ,	$tr(h_H) := [-1/r]$	$r > 0$
Kegel(stumpf):	$tr(h_K) := [0]$ ,	$tr(h_H) := [-1/r_1, -1/r_2]$	$0 < r_1 < r_2$
Ellipsoid:	$tr(h_K) := [a, b]$ ,	$tr(h_H) := [c, d]$	$0 < a < b, c < d < 0$

Da wir mit den Krümmungsmerkmalen zur Zeit experimentieren (vgl. 1.5.2), ist es möglich, daß die endgültigen Merkmale andere als K und H sein werden. An den Histogrammen und der Darstellung als solcher wird sich aber nichts Prinzipielles ändern.

- **Hauptrichtungsvektor  $F_\Phi$  (normiert)**

Die Richtung des Vektors  $\Phi$  hängt vom Typ der Oberfläche ab und ist wie bei Chen und Kak /9/ definiert:

Ebene:	Richtung der Oberflächennormalen
Zylinder,	
Kegel(stumpf):	Richtung der Zylinder- bzw. Kegel(stumpf)-Achse
Sphäre:	Richtung des Ortsvektors (!) des Kugelmittelpunktes
Quadrik:	Richtung des dominanten Eigenvektors

Beim Hauptrichtungsvektor handelt es sich um eine Verallgemeinerung des Extended Gaussian Image - Konzeptes /27/. Ein starrer, konvexer Körper bekannter Lage wird durch die Verteilung seiner Hauptrichtungsvektoren auf der Einheitskugel eindeutig charakterisiert. Die Hauptrichtungsvektoren eignen sich zur effizienten Überprüfung von Hypothesen zur Identität und Lage von Objekten, vgl. /9/, weniger zur Generierung der Hypothesen.

- **Schwerpunkt  $F_s$  ( $:= \frac{1}{n} \sum_{x_i \in F(i=1..n)} x_i$ )**

- **Größenparameter**

- Abschätzung des **Flächeninhaltes  $F_f$**
- maximale und minimale **Ausdehnung** der Fläche  $F_{dmin}$ ,  $F_{dmax}$ , z.B.  
 $dmin$ ,  $dmax$ : minimaler und maximaler Abstand der äußeren Randpunkte vom Schwerpunkt  $F_s$ , oder  
 $dmax$ : Länge der Diagonale der Umhüllenden (bounding box) der Oberfläche

- **Formparameter**, die die Form der Oberflächenberandung auf einfache Weise charakterisieren.

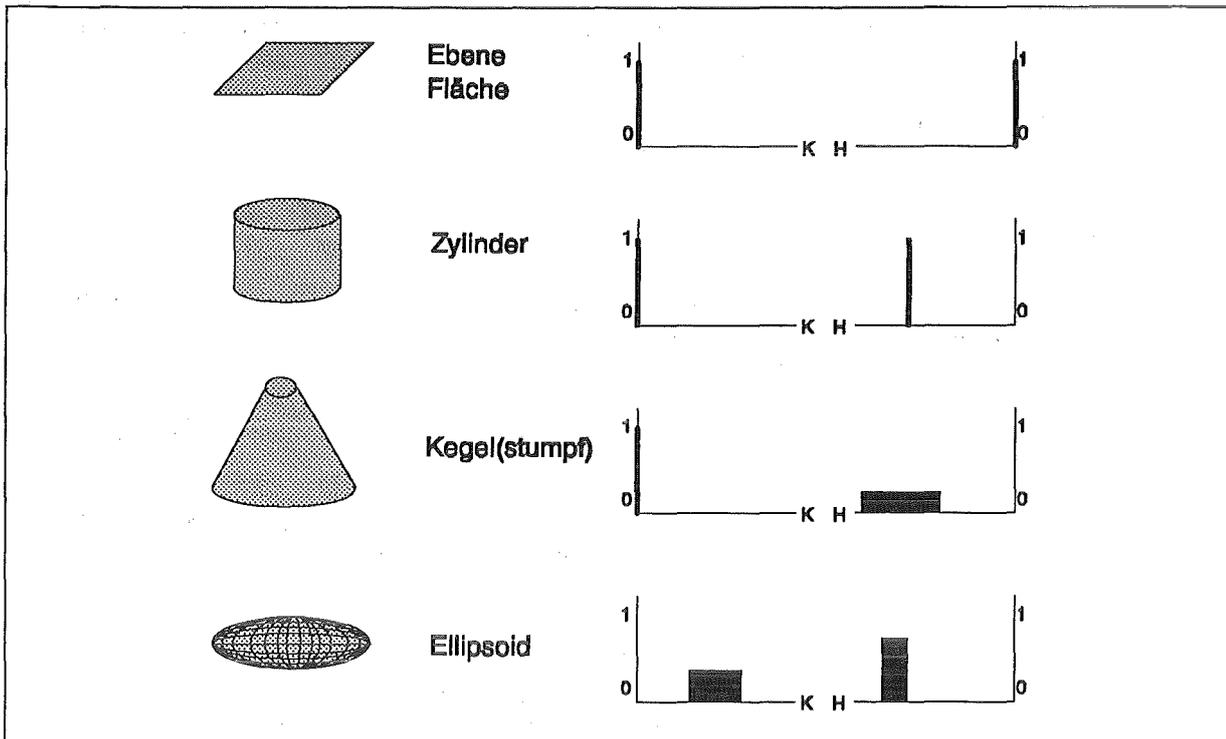


Abb. 2.1 Krümmungshistogramme einiger Grundkörper

#### Bemerkungen:

- Manche Oberflächenattribute, wie die approximierende Funktion, sind für ein maßgenaues CAD-Modell wichtig, leisten aber wenig für die Wiedererkennung. Andere, z.B. Krümmungshistogramm oder Hauptrichtungsvektoren dienen vor allem der Erkennung, werden aber in CAD-Systemen üblicherweise nicht unterstützt.
- Einzelne Attributwerte können, je nach Typ der Oberfläche und Qualität der Sensordaten oder der Rechenergebnisse der Vorverarbeitung, undefiniert oder unbrauchbar sein.
- Bei der Auswahl der Attribute insgesamt ist der Kompromiß zwischen Aufwand für die Vorverarbeitung und Nutzen für die Wiedererkennung noch im einzelnen auszuhandeln und erst durch Erfahrung zu festigen. Die Implementation der Objektrepräsentation ist daher offen und leicht änderbar und erweiterbar zu gestalten!

### 2.1.1.2 Relationen

#### Relationen 1. Ordnung zwischen Oberflächen

Grundlegende Relationen zwischen Flächen sind die **Nachbarschaftsbeziehungen**, also die Existenz und die Art einer gemeinsamen Berandung.

$nachbar(F_1, F_2, bdy) \Leftrightarrow$  die Oberflächen  $F_1, F_2$  haben einen Teil  $bdy$  ihrer inneren oder äußeren Berandung gemeinsam;  $bdy$  ist offener oder geschlossener Kantenzug.

Die gemeinsame Berandung  $bdy$  heißt **Sprungkante**, wenn überall dort ein unstetiger Übergang des Abstandssignals  $h$  vorliegt. Man spricht auch von verdeckender Kante, weil diese Eigenschaft vom Sensorstandort abhängt.

bdy heißt Dachkante, wenn dort das Entfernungsbild stetig ist, aber die Oberflächennormale einen unstetigen Übergang besitzt. Das ist genau dann der Fall, wenn die Tangentialebenen  $TE_1(y)$  von  $F_1$  bzw.  $TE_2(y)$  von  $F_2$  sich in allen Punkten  $y$  des Randes bdy schneiden. Der eingeschlossene Winkel

$\angle (F_1, F_2, y)$  von  $F_1, F_2$  im Punkt  $y$

wird als der im Material gemessene Winkel zwischen den Tangentialebenen  $TE_1(y)$  und  $TE_2(y)$  definiert. Ist dieser Winkel in jedem Punkt  $y$  gleich, so wird er als Flächenwinkel  $\angle (F_1, F_2)$  bezeichnet. Genauso ist der Flächenwinkel auch für nicht benachbarte, ebene Flächen definiert.

Gilt  $\angle (F_1, F_2) < 180^\circ$  (alternativ: liegt ein negatives Extremum der Richtungskrümmung senkrecht zur Kante bdy vor) handelt es sich um eine konvexe Dachkante.

Andernfalls - also  $\angle (F_1, F_2) > 180^\circ$  bzw. positives Extremum der Krümmung - liegt eine konkave Dachkante vor.

Damit läßt sich definieren:

$verbunden (F_1, F_2, roof, \alpha)$	$\Leftrightarrow$	$\exists bdy: \text{nachbar}(F_1, F_2, bdy) \wedge$ bdy ist Dachkante der Art roof (konvex oder konkav) $\wedge$ $\forall y \in bdy: \angle (F_1, F_2, y) = \alpha$
--------------------------------------	-------------------	---

Ein Sonderfall von *verbunden* ist die nichtsymmetrische Relation

$umschließt (F_1, F_2)$	$\Leftrightarrow$	$\exists bdy: \text{nachbar}(F_1, F_2, bdy) \wedge$ die Dachkante bdy ist geschlossene, innere Berandung von $F_1$
-------------------------	-------------------	---

Eine Verallgemeinerung ist die  $k$ -stellige Relation ( $k \geq 3$ )

$ecke (F_1, \dots, F_k, P, \alpha_1, \dots, \alpha_k)$	$\Leftrightarrow$	$\bigwedge_{i=1}^k (\text{verbunden}(F_i, F_{i+1 \bmod k}, \dots, \alpha_i) \wedge$ $F_1, \dots, F_k \text{ schneiden sich im Punkt } P$
--	-------------------	---

Bei der Existenz von Sprungkanten zwischen Flächen interessiert, welche der Flächen näher am Sensor aus der jeweiligen Blickrichtung liegt.  $F_1.m$  und  $F_2.m$  seien die Materialvektoren der Flächen nahe der gemeinsamen Sprungkante.

$näher (F_1, F_2)$	$\Leftrightarrow$	$F_1$ liegt von $F_2$ aus in Richtung $F_2.m \wedge$ $F_2$ liegt von $F_1$ aus in Richtung $-F_1.m$
--------------------	-------------------	--

Dies führt auf die, zu *verbunden* komplementäre, nichtsymmetrische Relation

$verdeckt (F_1, F_2)$	$\Leftrightarrow$	$\exists bdy: \text{nachbar}(F_1, F_2, bdy) \wedge$ bdy ist Sprungkante $\wedge$ $näher (F_1, F_2)$
-----------------------	-------------------	---

Daraus läßt sich die in einer Szene besonders vorteilhafte Eigenschaft ableiten, daß eine Fläche frei, also unverdeckt ist:

$frei (F)$	$\Leftrightarrow$	$\forall F': \text{nachbar}(F, F', \dots) \Rightarrow \text{verbunden}(F, F', \dots) \vee \text{verdeckt}(F, F')$
------------	-------------------	---

Wenn auf der gemeinsamen Berandung zweier gekrümmter Flächen verschiedenartige Übergänge vorliegen, können diese Flächen sowohl in der *verbunden*- als auch in der *verdeckt* - Relationen stehen, aber auch in keiner von beiden.

Folgende **Winkelbeziehungen** zwischen annähernd ebenen Flächen, die **nicht benachbart** sein müssen, werden festgehalten:

$schnitt (F_1, F_2, \alpha)$	$\Leftrightarrow$	die ebenen Flächen $F_1, F_2$ schneiden sich $\wedge$ $\angle (F_1, F_2) = \alpha$
$parallel (F_1, F_2, d, f_{\cap})$	$\Leftrightarrow$	die ebenen Flächen $F_1, F_2$ sind parallel bzw. koplanar $\wedge$ ihr euklidischer Abstand ist $d \wedge$ und ihr Überlappungsbereich hat Flächeninhalt $f_{\cap}$

Zwischen beliebigen, ebenen oder gekrümmtem, nicht benachbarten Oberflächen werden folgende **Abstandsbeziehungen** definiert:

$min.abstand (F_1, F_2, d)$	$\Leftrightarrow$	$\forall p_1 \in F_1, \forall p_2 \in F_2 : d(p_1, p_2) \geq d$  ( $d(p_1, p_2)$ euklidischer Abstand von $p_1$ und $p_2$ )
$max.abstand (F_1, F_2, d)$	$\Leftrightarrow$	$\forall p_1 \in F_1, \forall p_2 \in F_2 : d(p_1, p_2) \leq d$

### Relationen höherer Ordnung zwischen Oberflächen

Die bisher betrachteten Relationen bestehen aus Tupeln von geometrischen Grundelementen (Oberflächen). Über solchen Tupelmengen können nun selbst wieder Relationen höherer Ordnung gebildet werden. Die wichtigste Anwendung bilden die sogenannten **Fluchtlinien**. Das sind imaginäre Schnittlinien zweier annähernd ebener, nicht benachbarter und nicht paralleler oder koplanarer Oberflächen, also Tupel der oben definierten Lagebeziehung *schnitt*.

Faßt man Fluchtlinien  $L_1, L_2$ , als Geraden im  $\mathbf{R}^3$  auf, so gelten die üblichen Beziehungen zwischen Geraden:

$g.parallel (L_1, L_2, d)$	$\Leftrightarrow$	$L_1, L_2$ sind (annähernd) parallel und haben den euklidischen Abstand $d$
$g.windschief (L_1, L_2, \alpha, w)$	$\Leftrightarrow$	$L_1, L_2$ liegen windschief zueinander, haben den eingeschlossenen Winkel $\alpha$ und den Lotabstand $w$
$g.schnitt (L_1, L_2, \alpha, s)$	$\Leftrightarrow$	$L_1, L_2$ schneiden sich im Punkt $s$ und haben den eingeschlossenen Winkel $\alpha$

Tupel der Relation 2. Ordnung *g.schnitt* heißen auch **Fluchtpunkte** oder **Fluchtecken**. Eine Relation 3. Ordnung ist z.B. der euklidische Abstand zwischen Fluchtecken.

Ebenso ist der **minimale kollisionsfreie Abstand** zweier starrer Objekte eine Relation 2. Ordnung zwischen all den Flächen, die diese Objekte bilden.

### 2.1.2 Szenenrepräsentation

Die Repräsentation der von dem Objekterkennungssystem zu interpretierende Szene besteht aus

- einer Menge  $FS := \{F_1, \dots, F_s\}$  von Oberflächen, deren Attribute wie in 2.1.1 definiert sind
- einer Menge von Relationen erster oder höherer Ordnung über der Grundmenge  $FS$ , wie in 2.1.1 definiert.

### 2.1.3 Modellobjekte (Eingabe)

Das Vorwissen des Objekterkennungssystems über die zu erwartenden Objekte besteht aus

- a) einer Menge von **Modellobjekttypen**  $O_1, \dots, O_m$ .

Jedes Modellobjekt  $O_i$  ist repräsentiert durch

- eine Benennung  $oid_i$
- eine Menge  $FM_i := \{FM_{i,1}, \dots, FM_{i,m(i)}\}$  von Oberflächen, deren Attribute wie in 2.1.1 definiert sind
- eine Menge von Relationen erster oder höherer Ordnung über der Grundmenge  $FM_i$ , wie in 2.1.1 definiert.

Anders als Szenenansichten sind Modellobjekte ansichtenunabhängig und sollen keine verdeckenden Sprungkanten enthalten. Aus Sensordaten rekonstruierte 3D-Modellobjekte sollten folgende Vollständigkeits- und Konsistenzbedingung erfüllen:

Jede Fläche ist auf ihrer gesamten Berandung mit lauter Flächen desselben Objektes *verbunden*. Insbesondere ist jede Fläche eines Modellobjektes frei!

- b) **Relationen zwischen verschiedenen Objekten**

Relationen zwischen Objekten können meist auf Relationen höherer Ordnung zwischen deren Oberflächen zurückgeführt werden. Dabei handelt es sich vor allem um

- **Parallelitäts- oder Schnittrelationen**, die zum Beispiel stabile Lagen zwischen einem Objekt und einem anderen Objekt, etwa der Arbeitsfläche, beschreiben
- **Abstandsbeziehungen**, die den minimal möglichen kollisionsfreien Abstand zwischen verschiedenen starren Objekten wiedergeben und aus einer Kollisionsanalyse bzw. Konfigurationsraumanalyse abgeleitet werden können.

## 2.2 Objekterkennung als konsistentes Zuordnungsproblem

Das Wiedererkennungsproblem nach Def. 1.4 besteht im Kern aus einer Zuordnung:

$z$ : Szeneflächen  $\rightarrow$  Modellobjektflächen  $\times$  Instanznummern

$$FS \rightarrow \left( \bigcup_{j=1}^m FM_j \right) \times N$$

$$z(F) = \begin{pmatrix} zf(F) \\ zi(F) \end{pmatrix}$$

$zf(F)$  ist die  $F$  zugeordnete Modellobjektfläche, die zugleich den Typ des Modellobjektes eindeutig kennzeichnet,  $zi(F)$  die zugeordnete Instanznummer. Verschiedene Objekte haben verschiedene Nummern. Bei einer **korrekten** Zuordnung muß jede Fläche  $F$  einer Instanz  $k$  eines starren Objektes in einer Szene auf ihrer zugeordneten, instanzspezifisch transformierten Modellobjektfläche liegen:

$$F \subseteq Tr_k(zf(F)) \quad (\text{lageabhängige Beschränkung})$$

Umgekehrt kann in vielen Fällen die unbekannte Lagetransformation  $Tr_k$  (Translation  $T$  und Rotation  $R$ ) aus Paaren zugeordneter, zur selben Instanz  $k$  gehöriger Flächen

$$(F, zf(F)), F \in z^{-1}(FM \times \{k\})$$

berechnet und damit die gewünschte Lokalisierungsinformation zum Greifen des Objektes erhalten werden.

Zum Beispiel:

Für 3 ebene, sich in einem Punkt  $P_s$  schneidende Szeneflächen  $FS_1, FS_2, FS_3$ , und ihre zugeordneten, sich im Punkt  $P_m$  schneidenden Modellobjektflächen  $FM_1, FM_2, FM_3$  ergibt sich:

- Die Translation  $T \in \mathbb{R}^3$  führt die Schnittpunkte ineinander über  $\Rightarrow T = P_s - P_m$
- Die Rotation  $R \in \mathbb{R}^{3 \times 3}$  führt die linear unabhängigen Normalenvektoren ineinander über. Mit

$$\begin{aligned} \underline{NS} &= [\underline{ns}_1, \underline{ns}_2, \underline{ns}_3] \\ \underline{NM} &= [\underline{nm}_1, \underline{nm}_2, \underline{nm}_3] \text{ gilt } R \cdot \underline{NS} = \underline{NM} \quad \Rightarrow \quad R = \underline{NM} \cdot \underline{NS}^{-1} \end{aligned}$$

Bei mehr als 3 Flächenpaaren ist die Lagetransformation überbestimmt, also durch eine Ausgleichsrechnung zu errechnen. Andere Randbedingungen, z.B. nur 2 Flächenpaare und ein Punkt, linear abhängige Flächen oder gekrümmte Flächen liefern im allgemeinen keine eindeutige Transformation, man vgl. /9/ /36/ /43/.

Andererseits reicht unter bestimmten Bedingungen schon eine einzige zugeordnete Fläche, um eine Lagetransformation für das ganze Objekt zu berechnen, wenn deren Raumlage eindeutig ist. Es müssen hier mindestens 4 korrespondierende Punktepaare in Szene und Modell gefunden werden. Dazu muß die Szenefläche gut sichtbar, möglichst unverdeckt sein und darf keine Symmetrien enthalten, so daß markante Eckpunkte eindeutig wiedererkannt werden können.

Algorithmen, um Lagetransformationen in möglichst vielseitigen Situationen auffinden oder zumindest eingrenzen zu können, sind noch genauer zu untersuchen!

Jede **Zuordnung**  $z$  liefert also zusammen mit der Lagetransformation auch eine **Szeneninterpretation**, wie sie für die Objekterkennung nach Definition 1.3 und 1.4. verlangt wird:

$$SI := ((O\tilde{z}(1), T_1, R_1), \dots, (O\tilde{z}(n), T_n, R_n))$$

wobei  $\tilde{z}()$  die Zuordnung von Instanzen zu Modellobjekt-Typen bezeichnet.

### Definition 2.1 (korrekte Zuordnung)

$z$  heißt **korrekte Zuordnung**, wenn die aus  $z$  abgeleitete Interpretation  $SI$  des Sensorabbildes  $B$  bzw. seiner Repräsentation  $FS$  zulässig und korrekt im Sinne der Definition 1.4 ist.

Die Korrektheit zu prüfen erfordert

1. (Theoretisch) Das Objekterkennungssystem prüft die Bedingungen von 1.3 und 1.4 direkt nach:
  - $SI$  ist eine zulässige Interpretation einer Szene, beschreibt insbesondere eine kollisionsfreie Anordnung von starren Objekten.
  - Es gilt  $\pi(s(SI)) \subseteq B$ , d.h. die Szene  $s(SI)$  wird unter dem Sensor tatsächlich auf das Bild  $B$ , bzw. seine Repräsentation  $FS$  abgebildet.

Dies erfordert entweder, die Sensorabbildung  $\pi$  algorithmisch zu berechnen oder die errechnete Szene real nachzubauen, abzuscannen und zu vergleichen. Dieser sehr aufwendige Weg kommt höchstens in der Testphase (Funktionstest) eines Objekterkennungssystems mit simulierten Sensordaten in Frage.

2. Das Objekterkennungssystem prüft, ob alle bekannten **lageabhängigen Beschränkungen** erfüllt sind, also für jede Szenefläche  $F$  und jede Instanz  $k$ :

$$F \subseteq Tr_k(zf(F))$$

Diese Bedingung ist notwendig, im allgemeinen aber nicht hinreichend für die Korrektheit.

Lageabhängige Beschränkungen dienen dazu, eine **gefundene** Zuordnung  $z$  zu **testen**. Sie setzen die Kenntnis der Lagetransformation  $Tr$ , und damit eine zumindest teilweise bekannte Zuordnung voraus. Um  $z$  zu **finden**, kann  $Tr$  **nicht vorausgesetzt** werden. Hierfür sind sogenannte **invariante Beschränkungen** auszunutzen. Das sind  $k \geq 1$  stellige Prädikate  $P(F_1, \dots, F_k)$  über Szene- und Modellobjektflächen derart, daß für jede Zuordnung  $z$ , die eine korrekte Szeneninterpretation liefern soll, gelten muß:

$$P(F_1, \dots, F_k) \quad \Rightarrow \quad P(zf(F_1), \dots, zf(F_k)) \quad (2.1)$$

### Definition 2.2 (Invariante Beschränkung)

Ein Prädikat  $P$  über einer Menge von Oberflächen, die sowohl **FS** als auch **FM** enthält, liefert eine **invariante Beschränkung**, wenn für alle Szeneflächen  $F_1, \dots, F_k$ , und jede korrekte Zuordnung  $z$  (2.1) gilt.

Invarianz bedeutet Unabhängigkeit von

- der Lagetransformation zwischen Szene und Modellobjekten
- der partiellen Verdeckung von Objekten unter einer bestimmten Blickrichtung des Sensors
- den Meßfehlern und numerischen Ungenauigkeiten durch die Entfernungsmessung, Vorverarbeitung und Segmentierung, die zu Deformationen einzelner Merkmale führen können.

Die Idee besteht nun darin, gezielt solche Zuordnungen zu **suchen**, die invariante Beschränkungen erfüllen.

### Definition 2.3 (Global konsistente Zuordnung)

Eine Zuordnung  $z$  heißt (**global**) **konsistent** bezüglich einer Menge  $P$  von invarianten Beschränkungen, wenn  $z$  bzw.  $zf$  Bedingung (2.1) für alle  $P \in P$  erfüllt.

Man kann **FS** und **FM** mit der Menge  $P$  auch als relationale Struktur auffassen; in diesem Kontext wird eine global konsistente Zuordnung als **Monomorphismus** /37/ bezeichnet. Invariante Beschränkungen führen die Erkennung komplexer Objekte zurück auf die Zuordnung einfacherer Komponenten (Flächen) unter Ausnutzung der zwischen ihnen geltenden Beziehungen. Eine konsistente Zuordnung findet man durch **Propagieren invarianter Beschränkungen** ('constraint propagation' /41/).

Eine korrekte Zuordnung ist trivialerweise konsistent bzgl. **jeder** Menge invarianter Beschränkungen, aber **nicht umgekehrt**. Die  $P(F_1, \dots, F_k)$  stellen notwendige, im allgemeinen keine hinreichenden Bedingungen für eine korrekte Zuordnung dar. Wenn die Beschränkungen  $P$  nur **Winkelbeziehungen** zwischen Szene- und Modellobjektflächen betrachten, ignorieren die bzgl.  $P$  konsistenten Zuordnungen **Abmessungsfehler** bei der Zuordnung. Je vielseitiger und umfassender die Menge der invarianten Beschränkungen, umso näher kommt die Konsistenz der Korrektheit. Ob und welche die Korrektheit **garantierenden** invarianten Beschränkungen in der Entfernungsbildverarbeitung existieren, ist nicht bekannt.

Bevor wir Algorithmen zum Propagieren von Beschränkungen diskutieren, geben wir zunächst Beispiele und Gegenbeispiele für die Invarianz an.

## 2.3 Invariante Beschränkungen

Jeder  $n$ -stellige Relation ( $n \geq 2$ )  $R$  über Flächen und jedem Attribut  $A$  einer Fläche entsprechen Prädikate

$$P_R(F_1, \dots, F_k) \quad \Leftrightarrow \quad (F_1, \dots, F_k) \in R$$

$$P_A(F) \quad \Leftrightarrow \quad P(F.A) \quad \text{wobei } P \text{ irgendein Prädikat über der Wertemenge des Attributes } A, \\ \text{z.B. arithmetische Relation wie "kleiner" oder "größer".}$$

Es folgt eine Auswahl von Invarianzeigenschaften, die zum systematischen Auffinden konsistenter Zuordnungen geeignet scheinen. Diese Eigenschaften sind aber in jedem Fall erst durch Experimente zu verifizieren. Vor allem müssen geeignete Toleranzschwellen oder Maße der Unschärfe eingeführt werden, um Fehler und Ungenauigkeiten in der Erfassung und der Vorverarbeitung zu tolerieren. Oft sind zusätzliche Bedingungen an die Zuordnung  $z$  geknüpft, etwa daß die beteiligten Flächen zur selben oder zu verschiedenen Objektinstanz(en) gehören, um Beschränkungen anwenden zu können.

### 1) Inklusion der Krümmungshistogramm-Träger

**Gauß'sche und mittlere Krümmung  $K$  und  $H$**  als lokale Eigenschaften in jedem Punkt einer Oberfläche  $F$  sind bekanntlich invariant gegenüber Translation und Rotation. Folglich gilt dies auch für die Verteilungen der Krümmungswerte auf einer Fläche, die in 2.1.1 durch die Histogramme  $h_H, h_K$  eingeführt wurden. Dies gilt übrigens auch für die aus  $K$  und  $H$  berechenbaren **Hauptkrümmungen**  $k_1, k_2$  und die Histogramme  $h_{k_1}, h_{k_2}$ . Verdeckt man nun einen Teil der Fläche, entfällt auch ein Teil der Krümmungswerte, und damit kann sich die Verteilung der Krümmungswerte insgesamt verändern. Die Krümmungshistogramme als solche sind also nicht invariant gegenüber Verdeckung. Durch teilweise Abdeckung einer Szenefläche kommen aber keine neuen Krümmungswerte hinzu, außer vielleicht am Rand, der bei der Krümmungsverteilung ausgenommen wird. Also gilt für die Träger (vgl. 2.1.1) folgende wichtige Invarianzeigenschaft:

$\text{tr}(F.h_H)$	$\subseteq$	$\text{tr}(z(F).h_H)$
$\text{tr}(F.h_K)$	$\subseteq$	$\text{tr}(z(F).h_K)$
$\text{tr}(F.h_{k_1})$	$\subseteq$	$\text{tr}(z(F).h_{k_1})$
$\text{tr}(F.h_{k_2})$	$\subseteq$	$\text{tr}(z(F).h_{k_2})$

Die Inklusion ist äquivalent zur Implikation (2.1). Sie erlaubt komplexe, gekrümmte Flächen, auf die keine invarianten Eigenschaften ebener Flächen, wie Ecken, Schnittwinkel oder Abstände, anwendbar sind, trotz teilweiser Verdeckung zu klassifizieren. "Einfache" Oberflächen wie Zylinder, Kegel, Ellipsoid oder Torus sind Spezialfälle. Abb. 2.2 veranschaulicht die Anwendung der Krümmungshistogramm-Träger zur Erkennung einer teilweise verdeckten elliptischen Oberfläche, für die zwei Modellobjekte zur Auswahl stehen. Die Methode ist aber keineswegs nur auf quadratische Gleichungen oder ähnliche Funktionsklassen beschränkt.

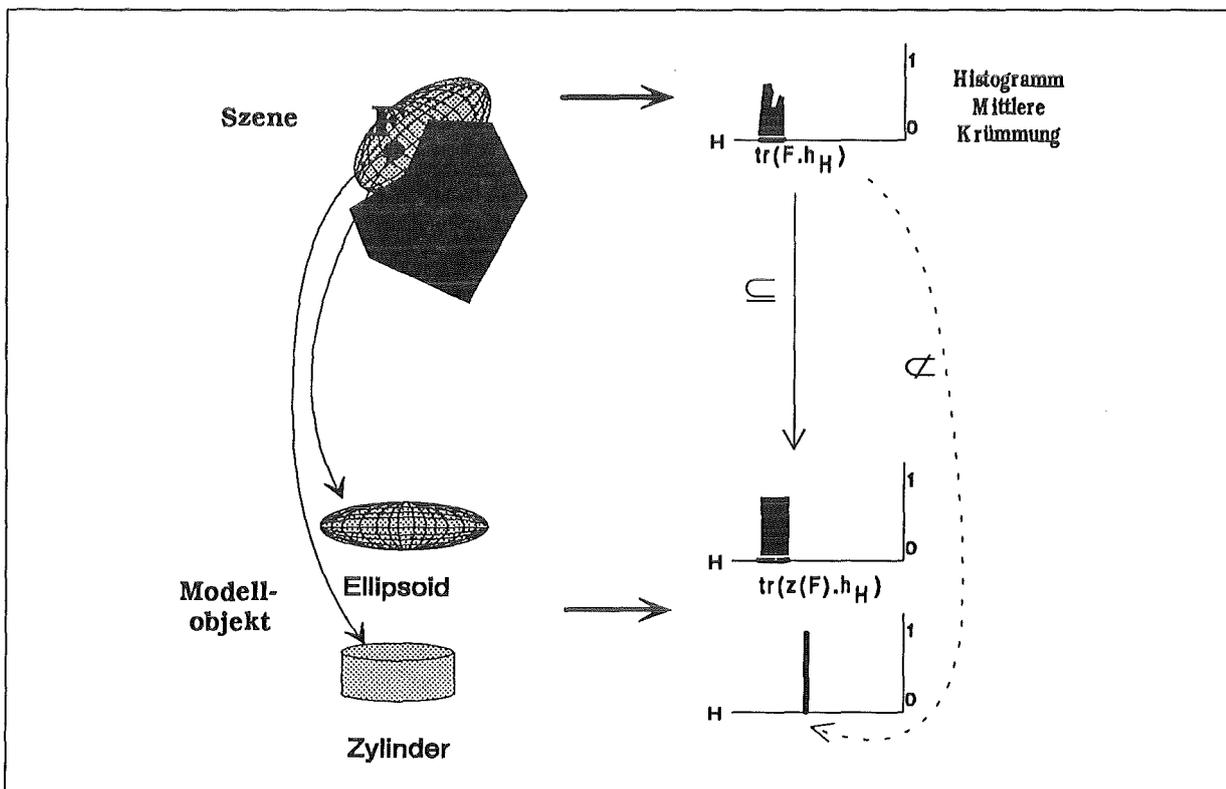


Abb. 2.2 Charakterisierung von Oberflächen durch die Träger der Krümmungshistogramme

Doch auch ein Wort zur Vorsicht: in der Praxis (vgl. 1.5.2) können Rauschanteile in den Sensordaten oder unterschiedliche Auflösung der Meßpunkte in Szene und Modellobjekten die Merkmale verändern. Hier ist eine gute Filterung der Meßpunkte, eine kritische Auswahl der Krümmungsmerkmale, aber auch die richtige Definition der Histogramme, eine "ungefähr-enthalten"-Relation  $\tilde{\subseteq}$  ihrer Träger etc. gefordert.

Die Inklusion der Histogramm-Träger gilt übrigens nicht bei Größenskalierung gekrümmter Flächen (generische Objekte)!

## 2. Nachbarschafts- und Winkelbeschränkungen

Für ebene Flächen  $F_1, F_2$ , die zur selben Instanz eines starren Objektes gehören, gilt

$$\text{verbunden}(F_1, F_2, \text{roof}, \alpha) \Rightarrow \text{verbunden}(z(F_1), z(F_2), \text{roof}, \alpha \pm \varepsilon)$$

( $\varepsilon$  geeignete Toleranzschwelle)

Entsprechend gilt für die k-stellige Eckenrelation

$$\text{ecke}(F_1, \dots, F_k, \mathbf{P}, \alpha_1, \dots, \alpha_k) \Rightarrow \text{ecke}(z(F_1), \dots, z(F_k), \text{Tr}(\mathbf{P}), \alpha_1 \pm \varepsilon, \dots, \alpha_k \pm \varepsilon)$$

Der Schnittpunkt  $\mathbf{P}$  ändert sich durch die Lagetransformation  $\text{Tr}$  und wird nicht benutzt. Aber die eingeschlossenen Winkel, unter der benachbarte, zum selben Objekt gehörende Szeneflächen aneinanderstoßen, und die Art der Dachkante (konvex oder konkav) sind invariant. Die Winkelinvarianz ist auch auf gekrümmte Flächen verwendbar, wenn die Flächenwinkel auf der gemeinsamen Berandung definiert sind, wie z.B. der 90°-Winkel zwischen Zylindermantel und -oberfläche.

Gehören dagegen  $F_1$  und  $F_2$  zu verschiedenen, bündig aneinanderliegenden Objektinstanzen, kann im allgemeinen nichts über deren Identität und Raumorientierung geschlossen werden, es sei denn, spezielle Beziehungen wie stabile Lagen oder Vorzugslagen erlauben dies.

Nachbarschaftsbeschränkungen, die in vielen Objekterkennungssystemen verwendet werden, können die Zuordnungen wirksam beschneiden, sind aber auch empfindlich gegenüber Segmentierungsfehlern:

In Abb. 2.3 verweisen die in der Szene benachbarten, verbundenen Flächen  $F_1, F_2$  auf zwei Modellflächen  $FM_1, FM_2$ , die in Wirklichkeit nicht direkt benachbart, sondern durch eine schmale angeschrägte Fläche  $FM_3$  ("Fase") getrennt sind. Dieses Detail wurde bei der Rekonstruktion oder CAD-Zeichnung erfaßt, ging aber in der Szenenbeschreibung verloren, weil die Auflösung der Meßpunkte zu grob war und die Segmentierung die Punkte von  $FM_3$  auf die Flächen  $F_1$  und  $F_2$  aufgeteilt hat, sodaß  $F_1$  und  $F_2$  nun direkt benachbart sind. Die Anwendung der Nachbarschafts-Beschränkung würde bedeuten, daß die eigentlich korrekte Zuordnung ausgeschlossen wird!

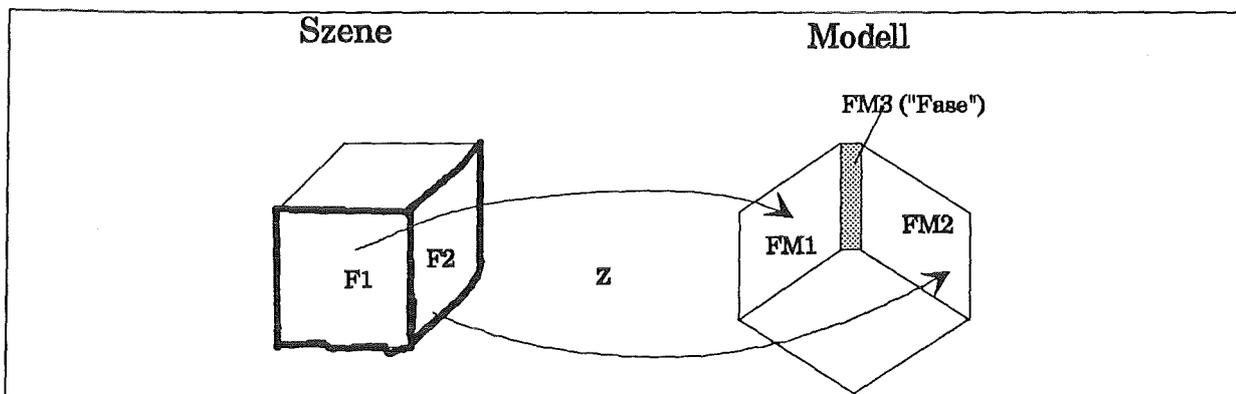


Abb. 2.3 Beispiel, in dem die *verbunden*-Relation nicht invariant gegenüber Segmentierungsfehlern ist

Die Winkelbeziehung als solche ist dagegen relativ robust gegenüber Meß- und Segmentierungsfehlern. Die folgende Schnittrelation berücksichtigt nur die eingeschlossenen Winkel von Flächen desselben Objektes, egal ob diese benachbart sind:

$$\boxed{\text{schnitt}(F_1, F_2, \alpha) \quad \Rightarrow \quad \text{schnitt}(z(F_1), z(F_2), \alpha \pm \epsilon)}$$

Entsprechend gilt für die Parallelitätsrelation:

$$\boxed{\text{parallel}(F_1, F_2, d, f_{\cap}) \quad \Rightarrow \quad \text{parallel}(z(F_1), z(F_2), d \pm \epsilon, )}$$

Über die Größe des Überlappungsbereiches  $f_{\cap}^*$  der zugeordneten Flächen läßt sich nichts aussagen.

Wir schlagen also vor, anstelle der Nachbarschaftsrelation vorgegebene Flächenwinkel zu suchen und dabei

- auch **nicht benachbarte Modellobjektflächen** zu berücksichtigen, weil wir uns auf die Invarianz der Nachbarschaft **nicht verlassen können**,
- auch **nicht benachbarte Szeneflächen** derselben Objektinstanz zu berücksichtigen, um Situationen abzudecken, in denen die Nachbarschaft gar **nicht anwendbar** wäre, z.B. infolge von Verdeckung in der Szene,
- die Information, die uns die **binäre Nachbarschaftsrelation** liefert, möglichst durch **euklidische Abstandsmaße** zwischen Flächen (vgl. 4.) zu ersetzen.

Um Winkel- und Nachbarschaftsbeschränkungen anzuwenden, muß bekannt sein, daß die beteiligten Flächen zur selben Objektinstanz gehören. Gegebenenfalls sind mehrere Hypothesen bzgl. Zugehörigkeit zur selben/zu verschiedenen Objektinstanzen getrennt zu untersuchen. Daß 2 **benachbarte** Flächen durch eine konvexe Dachkante verbunden sind, aber zu verschiedenen Objekten gehören, schränkt die in Frage kommenden Objekte und ihre möglichen Lagen ebenfalls sehr stark ein (vgl. 2.4.4).

Flächennachbarschafts- und Winkelbeziehungen sind auch invariant unter Skalierung, also auf generische Objekte anwendbar.

### 3. Fluchtlinien- und Fluchteckenbeschränkungen

Alle Eigenschaften von Fluchtlinien und Fluchtecken **ebener** Flächen in 2.1.1 sind invariant gegenüber Translation, Rotation und Verdeckung. Dazu zählen die *Schnitt*-, die *Parallelitäts*- und die *Windschief*-Relation zwischen Fluchtlinien und die *Abstände* zwischen Fluchtecken.

Das liegt daran, daß Fluchtlinien als **gedachte** Schnittlinien ebener Flächen kaum davon abhängen, wieviel von ihnen **sichtbar** ist. Große numerische Fehler treten erst auf, wenn die Flächen fast parallel oder koplanar, oder die Fluchtlinien bzw. -ecken von den Körperoberflächen weit entfernt sind.

Stellvertretend für invariante Beschränkungen greifen wir die Parallelitätsrelation für 2 Fluchtlinien heraus:

$$\boxed{\forall \quad F_{11}, F_{12}, F_{21}, F_{22}: \\ g.\text{parallel}(f_{\text{linie}}(F_{11}, F_{12}), \quad f_{\text{linie}}(F_{21}, F_{22}), \quad d) \quad \Rightarrow \\ g.\text{parallel}(f_{\text{linie}}(z(F_{11}), z(F_{12})), \quad f_{\text{linie}}(z(F_{21}), z(F_{22})), \quad d \pm \epsilon)}$$

Abb. 2.4 illustriert die Invarianz; als Beispiel wird der eingeschlossene Winkel  $\alpha$  der zwei Fluchtlinien l1 und l2, und der Abstand d der Fluchtecken P1 und P2 eines partiell verdeckten Polyeders betrachtet. Bei Fluchtecken, die aus ebenen Szeneflächen gebildet werden, kann es sich um

- wirkliche und sichtbare Körperecken
- wirkliche, aber verdeckte Körperecken (P2 in Abb. 2.4)
- imaginäre Körperecken (Ecken des konvexen Hüllkörpers, P1 in Abb. 2.4)

handeln. Sie sind also in allen 3 Fällen anwendbar.

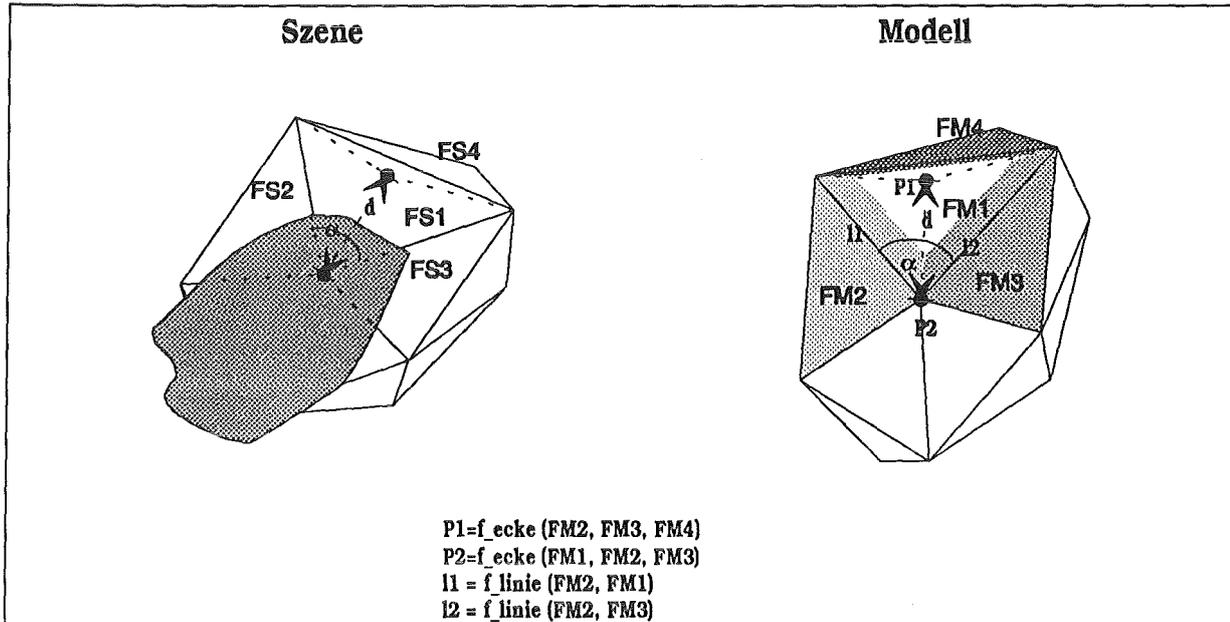


Abb. 2.4 Fluchtlinien- und Fluchteckenbeschränkungen bei polyedrischen Körpern

#### 4. Abstands- und Größenbeschränkungen

Die maximale Ausdehnung einer einzelnen Fläche bleibt weitgehend invariant unter Verschiebung und auch Drehung, solange die Fläche aus dem Blickwinkel des Abstandssensors noch gut erkennbar ist, und nimmt ab unter Verdeckung. Also

$$\forall F: \quad F.d\_max \leq z(F).d\_max \pm \epsilon$$

Diese Beschränkung eignet sich zur Identifizierung besonders großer Szenefflächen, etwa eines ebenen Bildhintergrundes.

Ebenso sind der maximale und der minimale (lichte) Abstand zwischen zwei beliebigen Flächen derselben, starren Objektinstanz lageinvariant und verdeckungs-monoton:

$$\forall F_1, F_2: \quad \begin{aligned} \max.abstand (F_1, F_2) &\leq \max.abstand (z(F_1), z(F_2)) \pm \epsilon \\ \min.abstand (F_1, F_2) &\geq \min.abstand (z(F_1), z(F_2)) \pm \epsilon \end{aligned}$$

Der minimale Flächenabstand hat eine ähnliche Funktion wie die Flächennachbarschaft. Besonders große maximale, oder besonders kleine minimale Abstände zwischen Flächenpaaren in einer Szene lassen also die wenigsten Alternativen. Bei leerer Alternativenmenge handelt es sich um **verschiedene Objektinstanzen**.

Sind andererseits **untere Schranken** für den Abstand zwischen Flächen bekannt, die zu **verschiedenen** Modellobjektinstanzen gehören, so gelten diese auch für jede kollisionsfrei angeordnete Szene mit verdeckten Objekten. Wenn zwei Objekte aneinanderstoßen, haben die sich direkt gegenüberliegenden Flächen immer einen minimalen Abstand nahe 0; der Nutzen des Flächenabstands ist dann gering. Der **minimale kollisionsfreie Abstand** von **Volumina** (Objekte), statt von Flächen, leistet wesentlich mehr. Der Abstand der Volumina kann als Abstand ihrer Schwerpunkte definiert werden. Um den Schwerpunkt auch für partiell verdeckte Objekte berechnen zu können, müssen in der Szene zumindest einzelne unverdeckte und lokalisierbare Flächen zu diesen Objekten existieren.

Sei	$\text{obj.abstand}(A_i, A_j) := \ A_i \cdot \underline{s} - A_j \cdot \underline{s}\ $	( $\underline{s}$ Schwerpunkt einer Objektinstanz $A_i$ bzw. $A_j$ )
	$\text{koll.abstand}(O_i, O_j) := \min_{A_i, A_j} \left\{ \text{obj.abstand}(A_i, A_j) \mid A_i/j \text{ Instanz von } O_i/j \right\}$	
$\forall F_1, F_2:$	$z_i(F_1)=i \wedge z_i(F_2)=j \wedge i \neq j \Rightarrow$	
	$\text{obj.abstand}(A_i, A_j) \geq$	$\text{koll.abstand}(O_{\tilde{z}(i)}, O_{\tilde{z}(j)}) \pm \epsilon$

koll.abstand kann für Paare von Modellobjekten z.B. aus einer Kollisions- bzw. Konfigurationsraumanalyse berechnet werden /14/.

Besonders kleine zugehörige Objektabstände zu Szeneflächen lassen also wenige Alternativen für die Vermutung, daß die Flächen auf verschiedenen Objekten liegen. Bei leerer Alternativenmenge handelt es sich um Flächen derselben Objektinstanz.

Auch Größen- und Abstandseigenschaften von Flächen haben eine wichtige implizite Voraussetzung, die falsch sein kann, nämlich daß jede Szenefläche zu genau einer Modellfläche gehört.

In Abb. 2.5 liegen 3 quaderförmige Objekte bündig aufeinander. Die Front- und Seitenflächen mehrerer Objekte verschmelzen zu einer einzigen ebenen Szenefläche. Die Szeneflächen F und F1 können durch die Abstandsbild-Segmentierung nicht sauber getrennt werden (in einem Grauwertbild wären die Kanten der einzelnen Quader noch eher erkennbar). Die maximale Ausdehnung  $d_{\max}$  von F übertrifft die aller Flächen der 3 Modellquader; die Anwendung der Größenbeschränkung würde zu dem falschen Schluß führen, daß F und F1 zu **keinem** der bekannten Modellobjekte gehören!

Läßt sich mit unseren bisherigen Mitteln überhaupt eine korrekte Zuordnung zu Objektinstanzen herleiten?

Die Antwort lautet ja, es reichen sogar elementare Kanten- und Winkelbeziehungen aus (konvexe und konkave Ecken, Parallelität). In Abb. 2.5 rechts sind die relevanten Beziehungen zwischen den Szeneflächen durch unterschiedliche Arten von Kanten in einem Graphen dargestellt. Die Lösung setzt voraus:

- die Zuordnung einer Szenefläche in 2.2 muß erweitert werden: mehrere Modellobjektflächen und Instanzen sind möglich.
- Kanten- und Winkelbeziehungen müssen auch **nicht benachbarte** Szeneflächen berücksichtigen.

Wir nehmen an, daß alle Modellobjekte polyedrisch, insbesondere konvex sind und ihre Flächen stumpfe oder rechte Winkel bilden. Dies schließt Quader aller Abmessungen ein. Flächen verschiedener Polyeder können nicht in einer konvexen Dachkante aneinanderstoßen, Flächen gleicher Polyeder sind weder parallel noch bilden sie konkave Übergänge. Also:

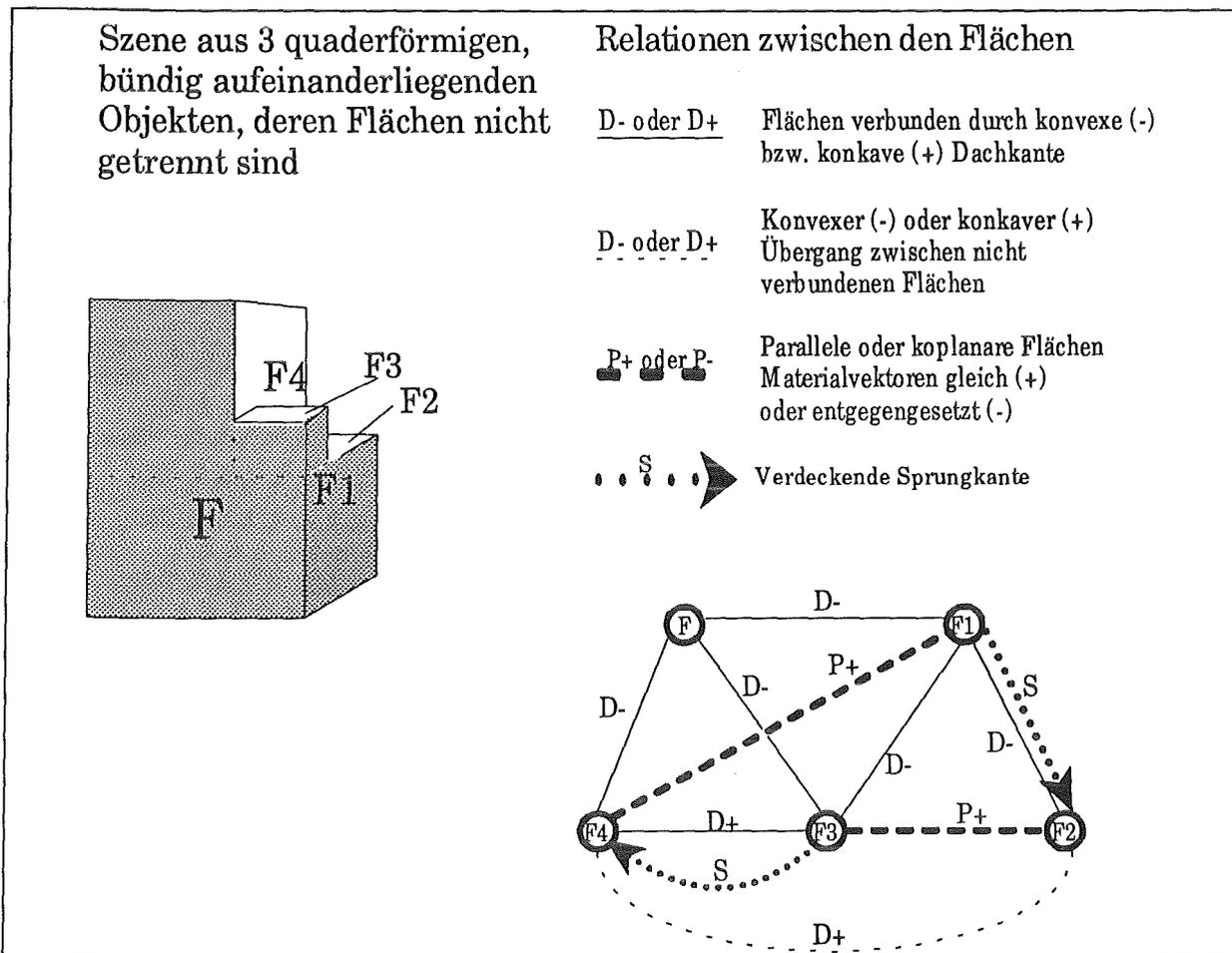
- |     |        |  |   |
|-----|--------|--|---|
| (1) | F, F4  | gehören zur selben Objektinstanz         | (da verbunden durch konvexe Dachkante)    |
| (2) | F, F3  | gehören zur selben Objektinstanz         | (da verbunden durch konvexe Dachkante)    |
| (3) | F, F1  | gehören zur selben Objektinstanz         | (da verbunden durch konvexe Dachkante)    |
| (4) | F1, F2 | gehören zur selben Objektinstanz         | (da verbunden durch konvexe Dachkante)    |
| (5) | F1, F3 | gehören zur selben Objektinstanz         | (da verbunden durch konvexe Dachkante)    |
| (6) | F1, F4 | gehören zu verschiedenen Objektinstanzen | (da parallel und gleicher Materialvektor) |
| (7) | F2, F3 | gehören zu verschiedenen Objektinstanzen | (da parallel und gleicher Materialvektor) |
| (8) | F2, F4 | gehören zu verschiedenen Objektinstanzen | (da konkaver Übergang ("Fluchtkante"))    |
| (9) | F3, F4 | gehören zu verschiedenen Objektinstanzen | (da verbunden durch konkave Dachkante)    |

$$\begin{aligned}
 (7),(8),(9) \quad \Rightarrow \quad & z_i(F2) \cap z_i(F3) = \emptyset \\
 & z_i(F2) \cap z_i(F4) = \emptyset \\
 & z_i(F3) \cap z_i(F4) = \emptyset \quad \Rightarrow \quad \text{mindestens 3 Objektinstanzen}
 \end{aligned}$$

$$\begin{aligned} (4),(5) &\Rightarrow z_i(F1) \supseteq z_i(F2) \cup z_i(F3) \\ (6) &\Rightarrow z_i(F1) \cap z_i(F4) = \emptyset \Rightarrow F1 \text{ Vereinigung von 2 Instanzenflächen} \end{aligned}$$

$$\begin{aligned} (1),(2),(3) &\Rightarrow z_i(F) \supseteq z_i(F1) \cup z_i(F3) \cup z_i(F4) \\ &\supseteq z_i(F2) \cup z_i(F3) \cup z_i(F4) \Rightarrow F \text{ Vereinigung von 3 Instanzenflächen} \end{aligned}$$

Mit dieser Information kann man Objekttyp und Lage der freien, nur einer einzigen Instanz angehörenden Fläche F3 bestimmen. Unter Benutzung der Lagetransformation ist es im Prinzip möglich, die Flächen F1 und schließlich F so zu zerlegen, daß auch die anderen beiden Objekte lokalisiert werden können.



**Abb. 2.5** Zuordnung von Szeneflächen, die zu mehreren Objektinstanzen gehören, mit Hilfe von Kantenübergängen und Winkelbeziehungen

### 5. Beschränkungen freier Flächen

Unverdeckte Szeneflächen erlauben uns, alle Beschränkungen uneingeschränkt anzuwenden, die eigentlich nur unter Translation, Rotation und Vorverarbeitungsfehlern invariant sind. In allgemeiner Form:

$$\forall F_1, \dots, F_k, \quad \forall P: P \text{ invariant unter Transformation und Vorverarbeitung}$$

$$\bigwedge_{i=1}^k \text{frei}(F_i) \quad \wedge \quad P(F_1, \dots, F_k) \quad \Rightarrow \quad P(z(F_1), \dots, z(F_k))$$

Konkret bedeutet dies zum Beispiel:

- der Flächeninhalt bleibt ungefähr gleich
- minimale und maximale Ausdehnungen bleiben ungefähr gleich
- die Form der Berandung bleibt gleich, usw.

## 6. Gegenbeispiele

- Die *verdeckt*-Relation (komplementär zur *verbunden*-Relation) ist trivialerweise nicht lage- und nicht verdeckungsvariant.
- Topologische Zusammenhangseigenschaften, z.B. Zahl der Löcher, sind lage-, aber nicht verdeckungsvariant. Sie können sich infolge Verdeckung sowohl vergrößern als auch verkleinern.
- Formeigenschaften wie Exzentrizität, Kompaktheit etc. sind sehr empfindlich gegenüber Verdeckung
- Der Flächeninhalt ist nicht verdeckungsvariant.
- Flächenschwerpunkte sind weder lage- noch verdeckungsvariant. Aus dem Schwerpunkt abgeleitete Maße (z.B. Radien, Abstände zwischen Flächen als Abstände ihrer Schwerpunkte definiert) sind nicht verdeckungsvariant, können sich bei Verdeckung also verkleinern, aber auch vergrößern.
- Trägheitsmomente sind empfindlich gegenüber Verdeckung und gegenüber der Vorverarbeitung (Punktauflösung).
- Hauptrichtungsvektoren und verwandte globale Objektmerkmale wie EGI sind weder rotations- noch (bei allgemeinen gekrümmten Oberflächen) verdeckungsvariant.
- Die Parameter der approximierenden Funktion einer Fläche (z.B. Quadrik, B-spline) sind nicht translations- und rotationsinvariant, aber vergleichsweise robust gegenüber Verdeckung.

## Resumee

- Die Anwendungsvoraussetzungen für die Invarianz von Beschränkungen sind genau zu prüfen. Winkelbeziehungen setzen voraus, daß mehrere Szeneflächen zur selben Objektinstanz gehören, Größenbeziehungen, daß eine einzige Szenefläche sich nicht über mehrere Objektinstanzen erstreckt, und Nachbarschaftsbeziehungen sind generell mit Vorsicht zu gebrauchen.

*Nicht invariante, oder in den Modellobjekten nicht repräsentierte Beschränkungen anzuwenden bedeutet, korrekte Szeneninterpretationen auszuschließen.*

- Starke, nur unter sehr speziellen Voraussetzungen gültige Beschränkungen sollten immer dann angewendet werden, wenn ihre Gültigkeit nachgeprüft wurde (z.B. freie Flächen).

*Invariante Beschränkungen der Szene nicht anzuwenden bedeutet, inkorrekte Szeneninterpretationen mitzuführen.*

- Wenn eine starke Beschränkung nicht anwendbar ist, leistet die Kombination mehrerer schwacher Beschränkungen vielleicht das gleiche. Dazu muß aber das System eine breite Palette invarianter Attribute und Relationen verarbeiten können, mehr als nur Nachbarschaft und Kantenübergänge, wie viele Objekterkennungssysteme. Die in diesem Abschnitt gegebene Auswahl erhebt keinen Anspruch auf Vollständigkeit und soll vor allem Anstoß zu weiterem Nachdenken sein.

- Es sollte deutlich geworden sein, daß invariante Beschränkungen zum Kern der Objektwiedererkennung gehören und daß eine effiziente Systemunterstützung zur Repräsentation, Berechnung, Propagierung, und Auswahl der Beschränkungen entscheidend für dessen Erfolg ist.

## 2.4 Propagierung von Beschränkungen

Zur Lösung von Zuordnungsproblemen mit Beschränkungen (dabei muß es sich nicht um Objekterkennung handeln, siehe /41/) sind aus der Künstlichen Intelligenz 2 Klassen von Verfahren bekannt: diskrete Relaxationsverfahren, sowie Generieren und Testen von Hypothesen.

### 2.4.1 Diskrete Relaxation

Statt eine Funktionszuordnung

$$z: \quad \text{FS} \quad \rightarrow \quad \text{FM}$$

zwischen Szene- und Modellelementen, oder bei mehrdeutigen Lösungen mehrere Funktionszuordnungen aufzustellen, geht die diskrete Relaxation von einer Relation

$$zr \subseteq \text{FS} \quad \times \quad \text{FM}$$

oder -äquivalent- von einer mehrdeutigen Zuordnung aus:

$$zm: \quad \text{FS} \quad \rightarrow \quad 2^{\text{FM}}, \quad zm(F) := \{F' \in \text{FM} \mid (F, F') \in zr\} \quad \forall F \in \text{FS}.$$

Die Menge  $zm(F)$  enthält also alle möglichen Zuordnungen (Modellflächen) der Szenefläche  $F$ . Ziel ist es, von  $\text{FM}$  ausgehend, die Mengen  $zm(F)$  durch wiederholte Anwendung invarianter Beschränkungen systematisch einzuschränken.

Für jedes Tupel von Szeneflächen  $(F_1, \dots, F_k)$ , das eine invariante Beschränkung  $P(F_1, \dots, F_k)$  erfüllt, lassen sich die Mengen  $zm(F_1), \dots, zm(F_k)$  folgendermaßen eingrenzen:

$$\forall j (j=1..k) : \quad zm(F_j)^{(n+1)} \subseteq \text{proj}_j \left( \prod_{i=1}^k zm(F_i)^{(n)} \cap R_P \right) \quad (2.2)$$

wobei der Index  $(n)$  bedeutet: nach der  $n$ -ten Anwendung,

$$R_P := \{(FM_1, \dots, FM_k) \mid P(FM_1, \dots, FM_k)\} \subseteq \text{FM}^k$$

die Menge aller Tupel von Modellflächen, die auch  $P$  erfüllen, und

$$\text{proj}_j(R) := \{FM \in \text{FM} \mid \exists (FM_1, \dots, FM_k) \in R \text{ mit } FM = FM_j\}$$

die  $j$ -te Projektion der  $k$ -stelligen Relation  $R \subseteq \text{FM}^k$  auf  $\text{FM}$ ,  $j=1..k$ .

In Worten: jeder Szenefläche  $F_j$ , für die sich eine invariante Beziehung  $P(F_1, \dots, F_j, \dots, F_k)$  aufstellen läßt, sind nur solche Modellflächen  $FM_j$  zuordbar, für die ihrerseits  $P$  erfüllende ( $\subseteq R_P$ ) und nach bisheriger Kenntnis mögliche Tupel  $(FM_1, \dots, FM_j, \dots, FM_k) \in \prod zm(F_i)^{(n)}$  existieren, mit  $FM_j$  als  $j$ -tem Element ( $\subseteq \text{proj}_j(\dots)$ ).

Änderungen bei einer Menge  $zm(F_j)$  pflanzen sich über benachbarte, mit  $F_j$  durch lokale Beschränkungen verknüpfte Elemente netzartig fort. Dasselbe Tupel kann dabei mehrmals an die Reihe kommen, wenn sich die Menge  $zm(F)$  für eine seiner Komponenten zwischenzeitlich verringert hat. Welche Tupel wann anwendbar sind, wird im folgenden Algorithmus durch die Menge *activated* ausgedrückt. Sind keine Tupel mehr aktivierbar, so terminiert der Algorithmus. Algorithmen in ähnlicher Form sind bei Kitchen und Rosenfeld /37/ und bei Kumar /41/ zu finden.

Sequentieller diskreter Relaxationsalgorithmus (D\_Relaxation)

**Eingabe:** Grundmengen  $FS, FM$  von Szenen- und Modellelementen  
 Menge  $P$  invarianter  $k$ -stelliger Beschränkungen mit Tupeln  $\in \bigcup_{k \geq 1} FS^k \cup \bigcup_{k \geq 1} FM^k$

**Ausgabe:** Mehrdeutige Zuordnung  $zm: FS \rightarrow 2^{FM}$

```

for all  $F \in FS$  do  $zm[F] := FM;$  {Initialisierung}
activated :=  $\{(\underline{t}, R) \mid R \in P, \underline{t} \in R\};$ 

repeat
  select  $(\underline{t} = (F_1, \dots, F_k), R) \in activated$  {Auswahl einer Beschränkung
  und eines Tupels; Strategiefunktion}

  for  $j := 1$  to  $k$  do
     $zm\_new[F_j] := proj_j \left( \prod_{i=1}^k zm[F_i] \cap R \right)$  {Anwendung von (2.2)}
    if  $zm\_new[F_j] \neq zm[F_j]$  then {geänderte Mengen merken}
       $changed\_FS := changed\_FS \cup \{F_j\};$ 
       $zm[F_j] := zm\_new[F_j];$ 

   $activated := activated \setminus \{(\underline{t}, R)\};$ 

  if  $changed\_FS \neq \emptyset$  then
    for all  $R' \subseteq FS^m$  ( $m \geq 2$ ),  $\underline{t}' = (F'_1, \dots, F'_k) \in R'$  do
      if  $(\underline{t}', R') \neq (\underline{t}, R)$  and  $(\{F'_1, \dots, F'_k\} \cap changed\_FS) \neq \emptyset$  then
         $activated := activated \cup \{(\underline{t}', R')\};$  {aktivierbare Tupel suchen,
        deren Element-Mengen sich
        geändert haben}

   $changed\_FS := \emptyset;$ 
until  $activated = \emptyset;$ 

```

**Resultate:**

$\exists F \in FS:$	$ zm(F)  = 0$	$\Rightarrow$	<i>es existiert keine konsistente Zuordnung</i>
$\forall F \in FS:$	$ zm(F)  = 1$	$\Rightarrow$	<i>eindeutige, konsistente Zuordnung</i>
$\forall F \in FS:$	$ zm(F)  \geq 1$	$\Rightarrow$	<i>mehrdeutige, konsistente Zuordnung</i>
$\wedge \exists F \in FS:$	$ zm(F)  > 1$	$\Rightarrow$	<i>mehrdeutige, konsistente Zuordnung</i>

Diskrete Relaxationsalgorithmen besitzen eine Reihe theoretisch wie praktisch wichtiger Eigenschaften; man vergleiche dazu auch Kitchen und Rosenfeld /37/.

1. Die **Terminierung** ist garantiert: in jeder Iteration nimmt entweder die Menge  $activated$ , oder mindestens eine der Mengen  $zm[F_i]$  ab. Beides ist nur endlich oft möglich.
2. Das Ergebnis reicht als **Lösung** im allgemeinen nicht aus. Es ist nämlich eine **lokal konsistente Zuordnung**  $zm: FS \rightarrow 2^{FM}$ , und zwar die **maximale**, jede andere lokal konsistente Zuordnung umfassende.

Eine mehrdeutige Zuordnung  $zm$  heißt **lokal konsistent**, wenn für jedes  $F \in FS$  und  $FM \in zm(F)$  eine die Beschränkungen erfüllende Funktionszuordnung  $z$  mit  $z(F)=FM$  in der Nachbarschaft von  $F$  existiert. Die Nachbarschaft sind diejenigen Szenenelemente  $F'$ , die mit  $F$  durch Beschränkungen (Tupel) verknüpft sind.

Es existiert aber nicht notwendig eine Funktionszuordnung auf der ganzen Menge  $FS$ , die alle Beschränkungen erfüllt, also nicht notwendig eine **global konsistente Zuordnung** wie in Def 2.3 verlangt.

$zm$  enthält aber als maximale Zuordnung alle global konsistenten Funktionszuordnungen  $z: FS \rightarrow FM$ , also alle möglichen mehrdeutigen Interpretationen von Objekttypen und -lagen, ohne diese explizit aufzuzählen. Gilt  $|zm(F)|=1$  für alle  $F$ , so hat man Glück:  $zm$  ist dann zugleich die einzige global konsistente Zuordnung, also die Lösung des Objekterkennungsproblems.

$zm$  kann auch global inkonsistente Funktionszuordnungen, also inkorrekte Lösungen enthalten. Es ist sogar möglich, daß der Algorithmus gar keine Reduktion der  $|FM| \cdot |FS|$  Zuordnungsmöglichkeiten erreicht, obwohl alle inkonsistent sind!

Die diskrete Relaxation stellt also eher ein Mittel zur **Eingrenzung** der möglichen Lösungen dar - oft aber ein sehr effektives Mittel - dem ein Suchprozeß folgen muß, um die tatsächlichen Lösungen herauszufinden.

3. Unterschiedliche **Reihenfolgen** bei der Auswahl der Beschränkungen (select (...)) beeinflussen die Geschwindigkeit der Terminierung, also die Effizienz. Das Endergebnis ist nach 2. aber immer gleich, die maximal lokal konsistente Zuordnung. Die Unabhängigkeit des Ergebnisses von der Reihenfolge der einzelnen Relaxationsoperationen (2.2) liefert zugleich den Schlüssel, um den Algorithmus zu **parallelisieren**, z.B. wie folgt:

- jedem Szenenelement  $F \in FS$  wird ein **eigener Prozeß**  $P[F]$  zugeordnet, der die Menge  $zm(F)$  verwaltet.
- Bei der Initialisierung wertet  $P[F]$  alle **einstelligen** Beschränkungen aus und bestimmt daraus  $zm(F)$ . Einstellige Beschränkungen sollen nur **einmal**, zu Beginn ausgewertet werden!
- Immer wenn  $P[F]$  von einem seiner Nachbarprozesse  $P[F']$  dessen aktualisierte Menge  $zm(F')$  empfängt, wertet  $P[F]$  die Tupel aus, in denen  $F$  zusammen mit  $F'$  vorkommt, und aktualisiert seine eigene Menge  $zm(F)$  nach Formel (2.2).
- Wenn  $zm[F]$  dadurch kleiner wurde, sendet  $P[F]$  die neue Menge an alle Nachbarprozesse  $P[F']$ , die gemeinsame Beschränkungen (Tupel) mit  $P[F]$  haben.

Der verteilte Algorithmus terminiert zu einem Zeitpunkt, zu dem jeder Prozeß alle empfangenen Nachrichten verarbeitet hat, ohne eine weitere Reduktion zu erreichen, und auch keine Nachrichten mehr unterwegs sind. Das Terminierungskriterium ist durch übliche Rundrufverfahren verteilt auszuwerten.

4. Die **Komplexität** des Algorithmus ist

$O(|FM| \cdot |FS| \cdot \# \text{ der Tupel (Beschränkungen)})$ ,

in aller Regel dramatisch weniger als

$O(|FM| \cdot |FS|)$ , die Zahl der möglichen Zuordnungen.

Denn maximal  $|FM| \cdot |FS|$ -mal kann eine der Mengen  $zm(F)$  verkleinert werden. Bei jedem solchen Ereignis werden alle Szenen-Tupel  $(F_1, \dots, F_k)$  (über  $FS$ ) bzgl. Aktivierbarkeit durchsucht, und alle Modell-Tupel (über  $FM$ ) zur Mengenaktualisierung.

In jeder Iteration, in der alle  $zm(F)$  gleichbleiben, vermindert sich  $|activated|$  um 1. Die Komplexität dieser erfolglosen Auswertungen ist ebenfalls durch die Zahl der Tupel begrenzt, weil mit  $activated=\emptyset$  der Algorithmus terminiert.

In der Praxis ist auch die effiziente Auswertung der einzelnen Projektionen in (2.2) wichtig; ein für beliebige  $k$ -stellige und vor allem nichtsymmetrische Relationen vom Rechenaufwand her gesehen nicht triviales relationales Datenbankzugriffsproblem.

## 5. Literatur, Erfahrungen

Waltz /66/ ist eine der ersten Arbeiten, die Relaxationsansätze zur Interpretation von Linienzeichnungen untersucht haben.

Bei Kitchen und Rosenfeld /37/ werden die mathematischen Grundlagen der diskreten Relaxation, insbesondere die lokale Konsistenz von Zuordnungen eingeführt. Als Anwendung werden Kontinente (Afrika, Nordamerika) aufgrund von Attributen und Relationen ihrer Länder (z.B. gemeinsame Grenzen) und ihrer Hauptstädte (z.B. Anfangsbuchstaben) bestimmt.

Bei Haralick und Shapiro /24/ werden Varianten des Relaxationsansatzes für das Subgraph-Isomorphieproblem vorgestellt.

Der Übersichtsartikel von Kumar /41/ zeigt, daß Relaxationsansätze noch immer ein aktives Forschungsgebiet darstellen und bringt Verallgemeinerungen der lokalen Konsistenz in Richtung globaler Konsistenz ("k-Konsistenz"). Kumar bezweifelt allerdings, ob die auf diesen Konzepten basierenden Verfahren im allgemeinen effizienter sind als diejenigen, die global konsistente Zuordnungen direkt, durch Suchen und Probieren mit Zurücksetzen, zu finden versuchen (siehe 2.4.2).

Praktische Erfahrungen mit einem sehr speziellen diskreten Relaxationsalgorithmus haben wir selbst der Erkennung von 2D-Umrissen von Druckmaschinenseitenwänden im Rahmen des Gußputzprojektes COMETOS /23/ gesammelt. Die Szenenelemente waren Liniensegmente, die aus der Gratfußvermessung geradliniger Bahnelemente stammten. Ihnen waren Bearbeitungsbahnen zuzuordnen, die am Bearbeitungsroboter für ein Musterwerkstück eingelernt worden waren. Die Koordinatentransformation zwischen Meß- und Bearbeitungsraum war zu bestimmen. Eine Art "Verdeckungsproblematik" ergab sich durch die zufällige Lage der gemessenen (meist kurzen) Gratfußelemente auf den langen Bearbeitungselementen. Als invariante Relationen wurden die eingeschlossenen Winkel der Linienelemente, die Abstände paralleler Linienelemente und die Abstände von Fluchtecken benutzt.

In diesem einfachen Spezialfall war der Relaxationsalgorithmus in der Lage, die eindeutige, auch global konsistente Zuordnung zu finden.

### 2.4.2 Generieren und Testen von Hypothesen

Im Gegensatz zur Relaxation arbeiten die folgenden Verfahren immer mit partiellen Funktionszuordnungen, die in einem Suchbaum organisiert werden.

Man beginnt damit, einem Szenenelement  $FS_1$  ein Modellelement  $FM_1 = z(FS_1)$  nach irgendeiner Auswahlstrategie zuzuordnen, und erweitert die Zuordnung schrittweise um neue Paare so, daß alle invarianten Beschränkungen erfüllt sind. Ansonsten besteht freie Wahl beim Probieren der Szenen- und Modellelemente. Man generiert also eine **hypothetische Zuordnung**.

Existieren genügend viele Paarungen von Szenen- und Modellelemente, die derselben Objektinstanz angehören, so kann man für diese Instanz eine einzige **Lagetransformation** berechnen (vgl. 2.2). Dies leistet zweierlei:

- Durch Anwendung der Transformation läßt sich rückwärtsrechnen, wo gewisse, noch nicht zugeordnete Modellobjektelemente in der Szene liegen müßten. Damit ergeben sich zwangsweise neue Zuordnungen für Szenenelemente, die bisher nicht zugeordnet waren.
- Existierende Zuordnungen lassen sich durch die zusätzliche Lageinformation erhärten oder widerlegen. Mit dem Übergang von lageunabhängigem zu lageabhängigem Wissen geht ein Übergang vom **Generieren** zum **Testen** der Hypothesen einher.

Der Prozeß endet entweder damit, daß die partielle Zuordnung eine der Beschränkungen verletzt, also verworfen wird, oder daß eine vollständige und korrekte Lösung gefunden wurde. In beiden Fällen setzt das

Verfahren bis zum letzten Verzweigungspunkt mit noch unerforschter Zuordnungsalternative zurück (backtracking), ordnet das Szenenelement neu zu und begibt sich von neuem in die Hypothesengenerierung.

Dieses Grundschema, nach dem die Mehrzahl heutiger Objekterkennungssysteme arbeitet, läßt sich durch folgenden Algorithmus beschreiben. Die wichtigste Datenstruktur ist der pulsierende Keller *zp* der partiellen Zuordnungen Szenelemente→Modellelemente:

```

type instantiation is record
    FS:          scene_element; {Variable}
    FM:          model_element; {zugeordneter Wert}
    enforced:    boolean;      {Zuordnung erzwungen?}
end;

```

```

type assignment is stack of instantiation;
var zp: assignment;

```

Auf dem speziellen Kellertyp *assignment* werden neben den Standardoperationen *push*, *pop*, *top*, *empty* etc. folgende Hilfsoperationen benötigt (Abkürzung *IB* bedeute: invariante Beschränkung)

<i>consistent</i> ( <i>zp</i> : <i>assignment</i> ): <i>boolean</i> ;	<i>true</i> ⇔ die partielle Zuordnung <i>zp</i> ist konsistent bezüglich der <i>IB</i> (induktiv über die Länge von <i>zp</i> definiert: jedes neue Paar in <i>zp</i> wird mit allen vorhergehenden Elementen auf Einhaltung der <i>IB</i> geprüft)
<i>TR_testable</i> ( <i>zp</i> : <i>assignment</i> ): <i>boolean</i> ;	<i>true</i> ⇔ die partielle Zuordnung <i>zp</i> enthält genügend viele Paare von Szene-/Modellelementen, die zur selben Objektinstanz gehören, um daraus eine instanzspezifische Lagetransformation <i>Tr</i> zu berechnen.
<i>TR_transform</i> ( <i>in out zp</i> : <i>assignment</i> );	Berechnet zu einer <i>TR_testable</i> , partiellen Zuordnung die Lagetransformation(en) <i>Tr</i> . In 2.2 wurde ein einfaches Rechenbeispiel dafür angegeben. Die Transformation <i>Tr</i> impliziert neue Paare ( <i>FS</i> , <i>FM</i> ), die zur selben Objektinstanz gehören. <i>TR_transform</i> erweitert den Keller <i>zp</i> um diese Paare und setzt für diese <i>enforced=true</i> .
<i>TR_correct</i> ( <i>zp</i> : <i>assignment</i> ): <i>boolean</i> ;	<i>true</i> ⇔ die durch <i>zp</i> definierte partielle Zuordnung ist konsistent bezüglich der <i>IB</i> und erfüllt auch alle bekannten lageabhängigen Beschränkungen.
<i>next_scene_el</i> ( <i>zp</i> : <i>assignment</i> ): <i>scene_element</i> ;	Liefert abhängig von den in <i>zp</i> vorhandenen Paaren die nächste zuzuordnende Variable ( <i>scene_element</i> ), oder <i>nil</i> , wenn die Zuordnung vollständig ist (Strategiefunktion).
<i>first_model_el</i> ( <i>zp</i> : <i>assignment</i> ; <i>FS</i> : <i>scene_element</i> ): <i>model_element</i> ;	Liefert abhängig von den in <i>zp</i> vorhandenen Paaren den ersten möglichen Wert für die Variable <i>FS</i> ( <i>scene_element</i> ) als Ergebnis, oder <i>nil</i> , falls keiner existiert (Strategiefunktion).
<i>next_model_el</i> ( <i>zp</i> : <i>assignment</i> ; <i>FS</i> : <i>scene_element</i> ; <i>FM</i> : <i>model_element</i> ): <i>model_element</i> ;	Liefert abhängig von den in <i>zp</i> vorhandenen Paaren den nächsten, auf <i>FM</i> <u>folgenden</u> Wert für die Variable <i>FS</i> als Ergebnis, oder <i>nil</i> , falls keiner mehr existiert (Strategiefunktion).

backtrack (in out zp: assignment): instantiation; Entfernt von einem nichtleeren Keller zp solange das oberste Paar, bis eine Zuordnung mit Wahlmöglichkeit (enforced=false) erscheint, oder nil. Liefert dieses neue, oberste Paar als Ergebnis.

```
backtrack (zp:assignment) =
    if not empty (zp) and not top (zp).enforced then
        top (zp)
    else
        backtrack (pop (zp, ..));
```

### Sequentieller Algorithmus: Generieren und Testen von Hypothesen (G&T)

**Eingabe:** Grundmengen FS, FM von Szenen- und Modellelementen  
Menge *P* invarianter Beschränkungen

**Ausgabe:** Menge von global konsistenten Funktionszuordnungen  
zp: FS → FM,  
die auch alle errechenbaren lageabhängigen Beschränkungen erfüllen

```
zp: assignment      := ();                               {Initialisierungen}
S: scene_element   := next_scene_el (zp);
M: model_element   := first_model_el (zp, S);

while M ≠ nil do

    lookahead_mode := true;
    while lookahead_mode and (S ≠ nil) do           {Erweiterungs-Modus}

        push (zp, (S,M, false));                   {enforced=false}
        if consistent (zp) then
            if TR_testable (zp) then
                lookahead_mode := TR_correct (TR_transform (zp))
            else
                lookahead_mode := false;

        if lookahead_mode then
            S := next_scene_el (zp);
            if S = nil then
                output (zp);
                lookahead_mode := false;
                {Eine gefundene, korrekte Szenenterpretation wird ausgegeben; nach
                weiteren wird gesucht}
            else
                M := first_model_el (zp, S);

    backtrack_mode := true;                         {Rücksetz-Modus}

    while not empty (zp) and backtrack_mode do

        (S,M) := backtrack (zp);
        M := next_model_el (zp, S, M);
        backtrack_mode := (M = nil);
```

Eine weniger detaillierte Darstellung dieses Algorithmus findet sich bei Stockman /58/. Objekterkennungssysteme für Entfernungsbilder, die solche oder ähnliche Verfahren benutzen, wurden von Kak et al /36/, Chen und Kak /9/ sowie von Fan /15/ beschrieben. Auch die in vielen Literaturstellen zitierten graph matching-Verfahren sind Spezialfälle. In Wirklichkeit ist bei der Objekterkennung aber ein Zuordnungsproblem für eine Menge von Hypergraphen - für jede k-stellige ( $k > 1$ ) invariante Relationen ein eigener Graph oder Hypergraph - und nicht für einen einzigen Graphen, also eine einzige 2-stellige Nachbarschaftsrelation, zu lösen.

### 2.4.3 Vergleich der beiden Verfahren

Auch das Generieren und Testen von Hypothesen muß wie die Relaxation auch invariante Beschränkungen ausnutzen, um inkonsistente Zuordnungen frühzeitig zu kappen. Die Zuordnungen blind durchzuprobieren und sich nur auf die Lagetransformation zu verlassen, scheidet in aller Regel an der schieren Größe des Suchraums.

Ansonsten unterscheiden sich die beiden Verfahren grundlegend in ihren Eigenschaften, siehe Tabelle 2.1.

Tab. 2.1 Vergleich von Diskreter Relaxation und Hypothesengenerierung

Kriterium	Diskrete Relaxation (DR)	Generieren und Testen (G&T)
Ergebnis	lokal konsistente, mehrdeutige Zuordnung	global konsistente Funktionszuordnung(en)
Aufzählen von Lösungen?	nein	ja
Mehrdeutigkeiten erhaltend?	ja	in der Praxis nein (zu aufwendig)
Komplexität	$O( FM  *  FS  * \# \text{Beschränkungen})$	$O( FM   FS )$
Zurücksetzen (Backtracking)?	nein	ja
Ausnutzen lageabhängiger Beschränkungen?	nein (bei Mehrdeutigkeit)	ja
Strategische Entscheidungen (Reihenfolgen)	Reihenfolge der Beschränkungen	Reihenfolge der Zuordnungen
Bedeutung der Strategien?	beeinflußt Effizienz	erfolgsentscheidend
Ansatz zur Parallelisierung	Unabhängigkeit aller Elemente und Zuordnungen	Unabhängigkeit der Alternativen zur selben Zuordnung

Die Vorteile und die Notwendigkeit von G&T liegen auf der Hand: nur diese Methode kann im allgemeinen global konsistente Funktionszuordnungen, und damit Objekttyp und -lage eindeutig bestimmen. Die DR muß sich oft mit einer mehrdeutigen Zuordnung begnügen, auch wenn eine **eindeutige** oder **gar keine** Lösung existiert. Wenn aber stets mehrere Zuordnungsmöglichkeiten bleiben, gibt es auch mehrere mögliche Lagebeziehungen. Daraus folgt, daß DR lageabhängigen Beschränkungen im allgemeinen nicht nutzen kann, um gefundene Lösungen zu erhärten oder zu widerlegen.

Dem stehen eine Reihe, letztlich alle in der Komplexität des Suchproblems  $O(|FM| |FS|)$  begründete und vor allem mit der Zahl der Szenenelemente wachsende **Probleme** der G&T-Verfahren gegenüber. Gute, anwendungsbezogene Suchstrategien helfen zwar, doch ist das Verfahren im Prinzip blind dafür, in welchen Teilen des Suchbaums korrekte Lösungen liegen und wo Verletzungen von Beschränkungen auftreten. Zurücksetzen bildet daher einen integralen Bestandteil. Dabei wird viel redundante Arbeit geleistet: dieselben Zuordnungen scheitern in verschiedenen Ästen des Suchbaums immer wieder an denselben lokalen Beschränkungen. Dieser Effekt wird in /41/ als 'thrashing' bezeichnet. "Intelligentere" Ansätze existieren zwar, doch macht der Aufwand, um Buch zu führen, wo und weshalb Zuordnungen gescheitert sind, den eingesparten Suchaufwand oft mehr als wett.

Selbst wenn durch die Beschränkungen, durch konsequentes Ausnutzen der Lagetransformation, durch geschickte Reihenfolgestrategien **eine erste** Lösung mit weit geringerem als exponentiellem Aufwand gefunden wird (z.B. polynomialem Aufwand niederen Grades in  $|FM|, |FS|$ ), gilt dies nicht mehr, wenn der Suchbaum

nach **allen** Lösungen, also **mehrdeutigen Interpretationen** zu durchsuchen ist. Bezeichnenderweise ist nach unserer Kenntnis keines der Objekterkennungssysteme dazu in der Lage. Wie bereits in 1.3.4 festgestellt wurde, erfassen Entfernungssensoren nur einen Teil der Umwelt. Ein Entfernungsbild-Erkennungssystem sollte keine Allwissenheit vortäuschen, sondern mit anderen Wissensquellen sinnvoll kooperieren können, die vielleicht den entscheidenden Hinweis, etwa Farbe oder Beschriftung von Objekten, liefern, welche der nach rein geometrischen Gesichtspunkten gleichwertigen Szeneninterpretationen die richtige ist. Dazu müssen aber alle diese Lösungen auch angeboten werden.

Ein Vorteil der mengenorientierten Darstellung der DR ist auch, daß mehrdeutige Interpretationen in kompakter Form erhalten bleiben, ohne alle Kombinationen explizit **generieren** zu müssen. In einer Szene, die  $k$  Objektinstanzen, zum Beispiel Polyeder, mit stark mehrdeutiger Lage enthält, von denen nur wenige ( $m < n$ ) der  $n$  Körperflächen sichtbar sind, ist eine Mengendarstellung der Zuordnung in der Form

$$\{FS_{i1}, \dots, FS_{im}\} \rightarrow \{FM_{i1}, \dots, FM_{in}\}_{1 \leq i \leq k} \quad (m: \text{Zahl der sichtbaren Szeneflächen von Objekt } i, m < n)$$

aussagekräftiger als die explizite Aufzählung aller möglicher Kombinationen der Objektlagen von  $k$  Körpern.

Ein weiterer Gesichtspunkt ist die Rolle der Strategien. Weil nur ein polynomialer Aufwand praktisch zu bewältigen ist, dienen Heuristiken bei G&T dazu, große Teile des Suchbaums von der Suche **auszuschließen**, und nicht nur eine effiziente **Suchreihenfolge** zu finden, also die "wahrscheinlichste" Lösung zuerst zu probieren. Nun können die ausgeschlossenen Teile gerade die korrekten Lösungen enthalten. Heuristiken in G&T-Verfahren berühren also nicht nur die **Effizienz**, sondern immer auch das Ergebnis, also die **Korrektheit** der Lösung.

Bei der DR hingegen wird eine mehrdeutige Zuordnung schrittweise reduziert, ohne dabei jemals korrekte Lösungen auszuschließen. Also muß auch nie zurückgesetzt werden. Die Strategie zur Auswahl der Beschränkungen beeinflußt die Konvergenzgeschwindigkeit (Effizienz), aber nicht das Ergebnis.

Ein wichtiges Kriterium, um die Objekterkennung auf ein verteiltes Transputer-System portieren zu können, ist die Parallelisierbarkeit der Verfahren. Beide lassen sich prinzipiell parallelisieren, und unterscheiden sich darin doch grundlegend.

Bei der DR operieren alle Szenen- und Modellelemente unabhängig und gleichzeitig auf ihren lokalen Beschränkungen. Das G&T-Verfahren arbeitet eher hierarchisch-sequentiell: zu jedem Zeitpunkt existiert eine partielle Zuordnung, von der verschiedene Zuordnungsalternativen abzweigen, die parallel verfolgt werden können. Selbst wenn jederzeit  $|FM|$  Prozessoren mit paralleler Alternativensuche gleichmäßig ausgelastet werden können, wird damit die Gesamttiefe  $|FS|$  des Suchbaums nur um 1 vermindert.

## Zusammenfassung

Diskrete Relaxation löst nicht das Gesamtproblem, bildet aber einen wichtigen Baustein, um den Suchraum einzuschränken, besitzt attraktive Eigenschaften und vermeidet viele Probleme der zurücksetzenden Verfahren. Die beiden Verfahren ergänzen sich in vieler Hinsicht. Wir schlagen daher vor, DR in G&T geeignet einzubetten, zum Beispiel beide Methoden abwechselnd anzuwenden, je nachdem welche gerade die größte Effektivität verspricht (vgl. 2.4.4, 2.4.5). Dabei sollten auch Anforderungen aus der Anwendung einfließen, etwa ob mehrdeutige Interpretationen auszugeben sind. Das kombinierte Verfahren ist nach unserem Wissen für die Objekterkennung neu, aber in ähnlicher Form auf andere Suchprobleme mit Beschränkungen bereits erfolgreich eingesetzt worden /41/.

Strategiefragen werden hierbei eine große Rolle spielen. Bevor wir Strategien sinnvoll diskutieren können, ist zuerst die Struktur des kombinierten Verfahrens festzulegen. Die Schnittstellen des Relaxationsalgorithmus und des G&T-Algorithmus sind so zu erweitern, daß beide sinnvoll kooperieren können.

## 2.4.4 Kombiniertes Verfahren

Das Verfahren operiert auf folgender gemeinsamer Schnittstelle:

- einer **partiellen, eindeutigen** Zuordnung auf einer Teilmenge der Szenenelemente  $\text{dom}(zp) \subseteq \text{FS}$  (zu Beginn gilt  $\text{dom}(zp) = \emptyset$ )  $zp: \text{dom}(zp) \rightarrow \text{FM}$
- einer **mehrdeutigen** Zuordnung auf den restlichen Szenenelementen (zu Beginn gilt  $zm(F) = \text{FM}$  für alle  $F \in \text{FS}$ ).  $zm: \text{FS} \setminus \text{dom}(zp) \rightarrow 2^{\text{FM}}$

Während G&T die partielle Zuordnung  $zp$  auszudehnen versucht, wobei  $zm(F)$  den Suchraum für die restlichen, noch zuzuordnenden Elemente  $F \in \text{FS}$  absteckt, faßt die diskrete Relaxation  $(zp, zm)$  als eine gesamte, mehrdeutige Zuordnung  $z: \text{FS} \rightarrow 2^{\text{FM}}$  auf und reduziert die Mehrdeutigkeit dieser Zuordnung, schränkt also den Suchraum ein.

**Algorithmus:** D\_Relaxation (für kombiniertes Verfahren)

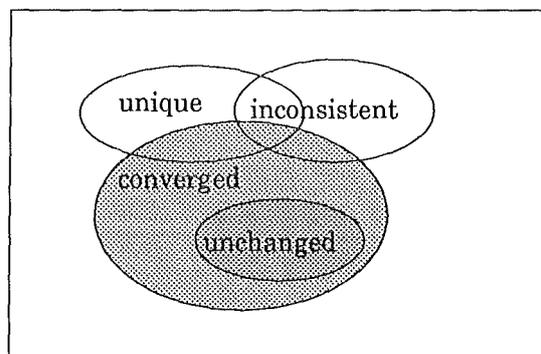
**Eingabe:** Grundmengen  $\text{FS}$ ,  $\text{FM}$  von Szenen- und Modellelementen  
Zuordnungen  $zp$ ,  $zm$  (siehe oben)  
Menge  $P$  von invarianten Beschränkungen.

**Ausgabe:** Zuordnungen  $(zp', zm')$  mit  
 $\text{dom}(zp') \supseteq \text{dom}(zp)$ ,  $zp' = zp \upharpoonright_{\text{dom}(zp)}$  (bisherige Zuordnung  $zp$  bleibt unverändert!)  
 $zm'(F) \subseteq zm(F) \forall F \in \text{dom}(zm)$

Weitere boolesche Resultate (Ergebnis-Flags):

- unique**  $\Leftrightarrow$  Die Zahl der eindeutig zugeordneten Elemente hat sich erhöht  
( $\text{dom}(zp) \neq \text{dom}(zp')$ )
- inconsistent**  $\Leftrightarrow$  Die Zuordnung ist widersprüchlich  
( $\exists F \in \text{FS}: |zm(F)| = 0$ )
- unchanged**  $\Leftrightarrow$  Die Zuordnung ist unverändert  
( $zp = zp'$  und  $zm = zm'$ )
- converged**  $\Leftrightarrow$  Die Zuordnung wurde stationär, D\_Relaxation hat konvergiert.

Die möglichen booleschen Ergebnisse sind in Abb. 2.6 als Mengendiagramm dargestellt. Anders als in dem früheren Algorithmus (2.4.1) kann jetzt die Relaxation verlassen werden, bevor sie konvergiert hat (**converged=false**). Prinzipiell könnte sie an diesem Punkt noch weiterarbeiten, aber wir ziehen es vor, mit der Hypothesengenerierung fortzufahren.



**Abb. 2.6** Mögliche Ergebnisse von D\_Relaxation

G&T reagiert im wesentlichen wie folgt:

- Fall 1 Gilt  $zm(F)=\emptyset$ , wurde also das Ergebnisflag **inconsistent** durch die vorangegangene Relaxation gesetzt, so ist die aktuelle Zuordnung  $zp$  zurückzusetzen.
- Fall 2 Wurden weitere Mengen  $zm(F)$  einelementig (**unique**), so versucht G&T zuerst, lageabhängige Beschränkungen zu generieren und prüfen.
- Fall 3 Andernfalls generiert G&T eine Hypothese, erweitert also  $zp$  um ein neues Paar (S,M).

Die Relaxation kann wieder aufgenommen werden, um den Suchraum weiter einzugrenzen. Dabei sind folgende Informationen relevant: welches Ergebnis brachte die letzte Relaxation (**unchanged, converged**), wurde seitdem schon eine neue Hypothese aufgestellt (Fall 3), gibt es eine gute Wahl für das nächste Szenenlement, und wie groß ist der Suchraum (Resultat `G&T_selection` der Funktion `next_scene_el`)?

Vorsicht ist beim Zurücksetzen einer partiellen Zuordnung  $zp$  geboten: die Mengen  $zm(F)$ , also die verbleibenden Möglichkeiten für die noch nicht zugeordnete Elemente, **ändern** sich, wenn  $zp$  zurückgesetzt und ein Teil der Hypothesen somit verworfen wird. Im folgenden Algorithmus werden die zukünftigen Zuordnungsmöglichkeiten, also die gesamte Zuordnung  $zm$ , an **jedem Verzweigungspunkt**, also für jedes Paar (S,M) im Keller  $zp$ , mitgeführt. Wird  $zp$  zurückgesetzt, so wird auch  $zm$  restauriert. Die Schnittstellen der Hilfsoperationen in 2.4.2 sind entsprechend zu modifizieren.

Die entfernt an PASCAL angelehnte Darstellung soll nur die Funktionsweise eines solchen Zuordnungsalgorithmus im Groben veranschaulichen und läßt keine Schlüsse auf die konkrete Implementation zu. Bei einer deklarativen Sprache wie PROLOG etwa ist Zurücksetzen in die Sprache integriert. Sowohl die aus der Anwendung kommenden Beschränkungen, als auch die Kontrollstruktur des Algorithmus, z.B. die Strategien, die die Wechsel zwischen Relaxation und Hypothesengenerierung steuern, sind durch Prädikate (Horn-Klauseln) repräsentiert.

Eine imperative Sprache wie PASCAL oder C sehen wir als ungeeignet zur Implementation des Prototyps eines solchen Zuordnungsalgorithmus an, weil sie unsere Forderung nach expliziter Repräsentation von Anwendungswissen (Kap. 1.4), insbesondere den flexiblen und einfachen Austausch von Attributen, Relationen und Beschränkungen, und von Strategien ihrer Anwendung nicht unterstützt. Dies gilt mit Einschränkungen auch für eine funktionale Sprache wie LISP.

Die Anwendung des kombinierten Verfahrens wird an 2 einfachen Beispielen erläutert.

### Beispiel 1: Anwendung der Lagetransformation

In Abb. 2.7 seien 5 Szeneflächen  $FS_1, \dots, FS_5$  zuzuordnen. Als Ergebnis von DR werden  $zm(FS_1)$  und  $zm(FS_2)$  auf je ein Element und  $zm(FS_5)$  auf 2 Elemente reduziert. Für die übrigen Szenenelemente  $FS_3, FS_4, FS_5$  bleibt die Zuordnung  $zm$  mehrdeutig. Diese Situation signalisiert G&T, die Regie zu übernehmen und Hypothesen für die Szenefläche mit den wenigsten Alternativen zu generieren, also  $FS_5$  mit seinen beiden Alternativen  $FM_{51}, FM_{52}$ . Die Alternative 1 (linker Zweig) führt auf eine Transformation  $T_1$  ( $FS_1, FM_1, FS_2, FM_2, FS_5, FM_{51}$ ), bei deren Anwendung die Fläche  $FS_3$  auf keiner der Modellobjektflächen in  $zm(FS_3)$  liegen würde. Also muß die Zuordnung  $FS_5 \rightarrow FM_{51}$  inkorrekt sein.

Folglich wird die Zuordnung  $zp$  zurückgesetzt und statt  $FS_5 \rightarrow FM_{51}$  die Alternative  $FS_5 \rightarrow FM_{52}$  versucht (rechter Zweig). Damit ergibt sich eine neue Lagetransformation  $T_2$  ( $FS_1, FM_1, FS_2, FM_2, FS_5, FM_{52}$ ). Da mit  $T_2$  in unserem Beispiel nur die Fläche  $FM_3 \in zm(FS_3)$  zu  $FS_3$  paßt, kann  $(FS_3, FM_3)$  im Gefolge mit  $(FS_5, FM_{52})$  in die eindeutige Zuordnung  $zp$  aufgenommen werden.

Mit Hilfe der neuen Information über  $FS_3$  und  $FS_5$  und der Relationen zwischen  $FS_4$  und  $FS_3$  bzw.  $FS_5$  könnte dann wieder die Relaxation ansetzen und die Alternativen für  $FS_4$  noch weiter reduzieren.

Im Grunde ähnelt das Verfahren sehr der Problemlösungsstrategie beim Lösen symbolischer Zahlenrätsel "durch Nachdenken, Probieren und Rechnen", bei der eine Zuordnung Ziffernsymbole  $\rightarrow$  Ziffern gesucht ist. "Nachdenken", also das systematische Eingrenzen möglicher Ziffern mit Hilfe der symbolischen Gleichungen und der Gesetze der Arithmetik (Beschränkungen!) entspricht unserer Relaxation. "Probieren", also Raten einer Ziffer, entspricht der Hypothesengenerierung in unserem Fall. "Rechnen", also Einsetzen einer gefundenen oder erratenen Ziffer in die Rechenaufgabe, um Vermutungen zu konkretisieren oder widerlegen, wird bei uns durch Anwendung der berechneten Lagetransformation erreicht.

Sequentieller kombinierter Algorithmus: G&T mit Relaxation

**Eingabe:** Grundmengen FS, FM von Szenen- und Modellelementen  
Menge  $P$  invarianter Beschränkungen

**Ausgabe:** Menge von global konsistenten Funktionszuordnungen  
 $zp: FS \rightarrow FM$ ,  
die auch alle bekannten lageabhängigen Beschränkungen erfüllen

```

zp: assignment      := ();                                {Initialisierungen}
for all F $\in$ FS do zm[F] := FM;
S: scene_element   := next_scene_el ((zp, zm), ...);
M: model_element;
search_exhausted  : boolean := false;

(zp, zm) := D_Relaxation ((zp, zm), P,
                        unchanged, converged, unique, inconsistent);

while not search_exhausted do

    lookahead_mode := true;
    while lookahead_mode and (S  $\neq$  nil) do      {Erweiterungs-Modus}

        if inconsistent then                    {Ergebnis von D_Relaxation:
                                                Rückkehr in Backtrack-Modus}

            lookahead_mode := false
        else
            if unique then                      {Neue eindeutige Zuordnungen}
                if TR_testable (zp) then
                    lookahead_mode := TR_correct
                    (TR_transform ((zp, zm)), unique)
                    {vermerken, ob neue einelementige
                     Mengen}

            if lookahead_mode then              {Zuordnung erweitern}

                S := next_scene_el ((zp, zm), G&T_selection);
                if S = nil then
                    output (zp);
                    lookahead_mode := false;
                    {Eine gefundene, korrekte Szenenterpretation wird ausgegeben; nach
                     weiteren wird gesucht}
                else
                    if resume_G&T              {Strategische Entscheidung:}
                        (unchanged, converged, unique, G&T_selection)
                    then                        {Hypothese generieren}
                        M := first_model_el ((zp, zm), S);
                        push (zp, (S, M, zm)); {Zuordnung zm retten!}
                        lookahead_mode := consistent (zp);
                    else                          {Suchraum weiter eingrenzen}
                        (zp, zm) := D_Relaxation ((zp, zm), P,
                                                unchanged, converged,
                                                unique, inconsistent);

            while not search_exhausted do      {Rücksetz-Modus}

                (S, M) := backtrack ((zp, zm)); {Zuordnung zm restaurieren}
                M := next_model_el (zp, zm, S, M);
                search_exhausted := (M = nil) or (S = nil);

```



vorzunehmen. Um den Suchraum effizient zu strukturieren, schlagen wir vor, überall dort, wo verschiedene **Gruppierungsentscheidungen** für Flächen  $FS_i, FS_j$  möglich sind, diese als **Hypothesen** durch den G&T-Algorithmus zu generieren:

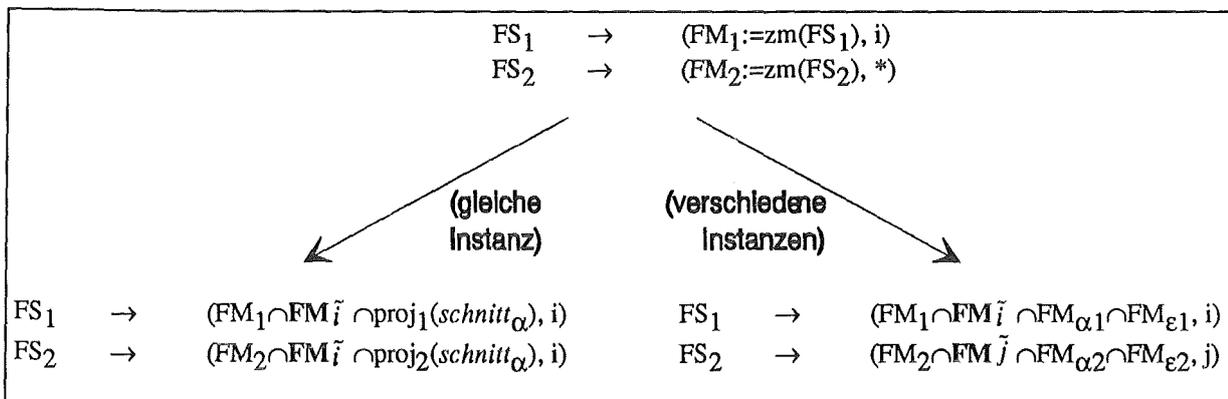
- (H1)  $FS_i$  gehört zu selben Instanz wie Fläche  $FS_j$   
 (H2)  $FS_i$  gehört zu einer anderen Instanz als  $FS_j$ ,

und dann (H1) bzw. (H2) mit ihren jeweils eigenen Mengen von Beschränkungen weiter zu reduzieren.

Um die Grundidee zu erläutern, nehmen wir 2 beliebige Flächen  $F_1, F_2$ , die

- durch eine konvexe Dachkante mit eingeschlossenem Winkel  $\alpha$  verbunden sind
- einen Abstand von  $\epsilon$  besitzen.

$F_1$  sei bereits eine Objektinstanznummer  $i$  zugeordnet,  $F_2$  noch nicht. Dann ergeben sich folgende Alternativen:



wobei

- $FM_{\tilde{i}}, FM_{\tilde{j}}$  alle Modellflächen, die mit der Instanznr.  $i$  bzw.  $j$  möglich sind  
 $schnitt_\alpha$  alle Paare von Modellobjektflächen, die sich im Winkel  $\alpha$  schneiden  
 $FM_{\alpha 1/2}$  alle Modellflächen in  $FM_1 \cap FM_{\tilde{i}}$  bzw.  $FM_2 \cap FM_{\tilde{j}}$ , die in konvexen Ecken ihrer jeweiligen Modellobjekte mit eingeschlossenen Winkeln vorkommen, deren Summe  $< \alpha$  ist  
 $FM_{\epsilon 1/2}$  alle Modellflächen in  $FM_1 \cap FM_{\tilde{i}}$  bzw.  $FM_2 \cap FM_{\tilde{j}}$ , für die ein minimaler gegenseitiger Abstand  $< \epsilon$  möglich ist, wenn sie zu verschiedenen Objektinstanzen gehören.

### 2.4.5 Strategische Überlegungen

Bei dem kombinierten Algorithmus in 2.4.4 sind geeignete Strategien für die Auswahl von Elementen und Verfahren zu treffen:

1. innerhalb der Relaxation: in welcher Reihenfolge werden die Beschränkungen ausgewählt?  
(select-Anweisung des Algorithmus in 2.4.1)
2. bei der Hypothesengenerierung: wie werden die Szenen- und Modellelemente ausgewählt?  
(Funktionen `next_scene_el`, `first_model_el`, `next_model_el` des Algorithmus in 2.4.4)
3. Nach welchen Kriterien erfolgen die Übergänge zwischen den Verfahren
  - von der Relaxation zur Hypothesengenerierung
  - von der Hypothesengenerierung zur Relaxation ?

### Zu 1 (Relaxation)

- Alle 1-stelligen, nur von einem einzigen Szene- oder Modellelement abhängigen Beschränkungen (z.B. Krümmungshistogramm), sind am Anfang, vor allen mehrstelligen Beschränkungen anzuwenden. Mehrstellige Beschränkungen können nämlich von den 1-stelligen profitieren, aber nicht umgekehrt.
- Für Szenentupel mit Beschränkungen, die unter den Modellelementen selten vorkommen, verbleiben nur wenige Möglichkeiten. Solche Tupel sind bevorzugt auszuwählen und bilden eine Art Keimzelle für die Reduktion benachbarter Elemente.

Werden z.B. folgende Merkmale betrachtet

- a) die eingeschlossenen Winkel zweier Flächen
- b) die Abstände paralleler Flächen
- c) die maximale Ausdehnung einzelner Flächen

und die Modellobjekte aus Werkstücken verschiedener Größen und Formen mit überwiegend rechten Winkeln bestehen, dann sollte man die Abstandsbeziehung paralleler Flächen (b) bevorzugen und nicht die Winkel a), weil fast alle Winkel  $0^\circ$  oder  $90^\circ$  betragen. Treten dagegen die Winkelwerte breit gestreut auf, so besitzt die Winkelbeziehung a) eine bessere Unterscheidungsfähigkeit. Wenn Szenenelemente existieren, deren Ausdehnung (c) die fast aller Modellelemente übertrifft, sollten diese aus demselben Grund frühzeitig identifiziert werden.

Die Reihenfolgenfindung wird erleichtert, indem man die Modellelemente nach ihren Merkmalswerten vorsortiert (vgl. /32/ /55/).

- Offen ist, ob Beschränkungen, die sich als besonders leistungsfähig erwiesen haben, in aggressiver Weise immer wieder feuern, solange sie aktivierbar sind, oder ob unterschiedlich strukturierte Beschränkungen und Elemente nach einer Fairneßregelung reihum aktiviert werden.

### Zu 2. (G&T)

Eine speziell für die Objekterkennung nützliche Strategie besteht darin, solche Szenen- und Modellelemente zuerst zuzuordnen, die die **Berechnung einer Lagetransformation** erlauben. Wenn eine Objektinstanz mit zwei zugeordneten Szeneflächen bereits existiert, sollte eine dritte ebene, nicht parallele Fläche derselben Instanz gesucht und zugeordnet werden, um daraus direkt eine Translation und eine Rotation berechnen zu können.

Nun ist die Zuordnung auch ein Spezialfall des **Instantiierungsproblems**, bei dem **Variablen** (Szenenelementen) **Werte** (Modellelemente) aus einem gegebenen Wertebereich zugeordnet werden. Für dieses allgemeinere Problem sind viele Strategien bekannt /41/. Wir erwähnen hier einige, deren Kosten und Nutzen noch zu untersuchen bleibt:

- a) Variablen mit dem **kleinsten Wertebereich** zuerst instantiieren ('search rearrangement method')
- b) Variablen, die in den **meisten Beschränkungen** (Tupeln) vorkommen, **zuerst** instantiieren, siehe Fox, Sadeh und Baykan /18/ oder Freuder /19/.  
Daran knüpft sich die Hoffnung, fehlerhafte Zuordnungen für diese Variablen frühzeitig zu entdecken.  
Eine Variante besteht darin, Mengen von Variablen zu finden, zwischen denen **keine Beschränkungen** existieren ('stable sets'), und diese **zuletzt** zuzuweisen.  
Dies kann die Komplexität des Instantiierungsproblems sogar drastisch, von exponentiell auf linear, reduzieren. Leider ist das Problem, stabile Variablenmengen zu finden, selbst NP-vollständig /41/.
- c) Variablenmengen finden, nach deren Wegnahme der Graph der **Beschränkungen** **zyklenfrei** wird ('cycle cutsets'), und diese Variablen **zuerst** instantiieren. Damit können die verbleibenden Variablen ohne Zurücksetzen zugewiesen werden /12/.

Diese Strategien zur Auswahl der **Variablen** verfolgen alle das Ziel, falsche Zuordnungen möglichst früh zu erkennen, also möglichst nah an die Wurzel des Suchbaums zu bringen.

Eine weitere Strategie betrifft die Auswahl der **Werte** und verfolgt das Ziel, die korrekten Lösungen möglichst weit links im Suchbaum zu plazieren, wo sie zuerst gefunden werden:

- d) Bei der Zuordnung solche Werte bevorzugen, die für die noch zu instantiierenden Variablen möglichst viele Möglichkeiten offenlassen, siehe Haralick und Elliot /25/.

### Zu 3. (Übergänge)

Die Relaxation muß auf jeden Fall beendet werden, wenn die Zuordnung  $z_m$  stationär geworden ist. Es empfiehlt sich aber, schon vorher die Hypothesengenerierung zu aktivieren, wenn

- die mittlere "Reduktionsrate" der Mengen  $z_m(F)$  pro Beschränkung zu gering geworden ist, oder
- einzelne Mengen  $z_m(F)$  im Vergleich zu anderen sehr klein oder einelementig geworden sind. G&T muß für diese Elemente nur noch wenige Alternativen testen und kann vielleicht eine Lagetransformation berechnen.

Umgekehrt kann es sinnvoll sein, von der Hypothesengenerierung zur Relaxation zurückzukehren, wenn

- G&T bei den Szenen- und Modellelementen keine bevorzugte Wahl findet, **und**
- seit der letzten Anwendung der Relaxation bereits Fortschritte erzielt wurden, also die partielle Zuordnung erweitert wurde.

### 3. Projektentwicklung

#### 3.1 Arbeitspakete und Zeitplan

Im folgenden wird ein Vorschlag zur Abwicklung des Projektes SOMBRERO ('surface oriented model building for the reconstruction and recognition of objects') gemacht. Die Gesamtaufgabe wird in Arbeitspakete eingeteilt, die sich aus der inhaltlichen Darstellung der Kapitel 1 und 2 ergeben und daher nur stichpunktartig aufgeführt werden.

Anschließend wird ein Zeitplan erstellt (Abb. 3.1), der von folgenden personellen und organisatorischen Randbedingungen ausgeht:

- Die Pilotentwicklung des Zuordnungsalgorithmus (Kap. 2.4) wird als Entwicklungsauftrag an ein Forschungsinstitut vergeben. Kernpunkte der Entwicklung sind die flexible Objektrepräsentation und effiziente, zum Teil neuartige Mechanismen zur Propagierung invarianter Beschränkungen. Diese Punkte sollen in einer geeigneten KI-Entwicklungsumgebung bearbeitet und zunächst an einfachen, simulierten Szene- und Modellobjektdateien getestet werden. Wir hoffen, mit dieser Aufgabenteilung vorhandenes grundlagenorientiertes know-how für unsere Anwendung nutzbar machen zu können, statt es bei uns selbst zu entwickeln.  
Die Spezifikation der Objektrepräsentation und der Verfahren erfolgt zusammen mit dem Auftraggeber (KfK-IAI). Die Verfahren sollen später auf ein verteiltes Transputer-Zielsystem portiert werden. Der Gesichtspunkt der Parallelisierbarkeit ist daher schon im Entwurf zu berücksichtigen. Hierbei, und später bei der Erprobung des Zuordnungsalgorithmus an realen 2D-LADAR-Szenen ist eine enge Zusammenarbeit zwischen Auftraggeber und Auftragnehmer notwendig.
- Für Entwicklungs- und Implementierungsarbeiten und Experimente in den Bereichen Datenerfassung, Vorverarbeitung, Segmentierung, Merkmalsberechnung, Fusionierung stehen weitere Mitarbeiter/Hilfskräfte (Praktikanten, Diplomanden, evtl. Doktoranden) zur Verfügung.  
Die Software zur Vorverarbeitung wird von Anfang an für das Parallelrechner-Zielsystem (Transputernetz, OCCAM 2-, eventuell auch C-Toolset) entwickelt.
- Längerfristige Arbeiten wie die Portierung der eigentlichen Wiedererkennung auf ein T9000-basierendes Transputernetz, die Schließung des Kreislaufs von Objektrekonstruktion und Wiedererkennung, die Integration und Erprobung neuer Sensoren etc. sind im Zeitplan aufgeführt, werden aber erst bearbeitet, wenn sich eine Anwendung im Bereich Umwelttechnik, Mikrosystemtechnik oder Energietechnik konkretisiert. Hierfür wird dann weiteres Fachpersonal benötigt.  
Das TT-Vorhaben zur Kranautomatisierung ist als relativ eigenständiges und unabhängiges Projekt zu sehen, in dem zwar dieselben Entfernungssensoren und Transputer-Hardware, aber aufgrund der spezifischen Echtzeitanforderungen in weiten Teilen maßgeschneiderte Verfahren verwendet werden.

Arbeitspakete (*kursiv* in Klammern: kausale Abhängigkeiten von direkten Vorgängern):

**(P1) Erfassung kartesischer Datenbestände**

Sensor- und Roboterkoordinaten, DNC-Schnittstelle, Weltkoordinaten, Transputer-Konfiguration mb1T9S

**(P2) Verbesserungen und Erweiterungen der Datenerfassung (P1)**

Winkelauflösung, Korrelation der Winkel- und Entfernungswerte, Sensorkalibrierung, DNC-Steuerung, SPISCA-Bedieneroberfläche

**(P3) Datenaufbereitung (P1)**

Filterung, Sensor-Fehlermodelle, Signalanalyse, Experimente mit verschiedenen Oberflächen

- (P4) **Krümmungsextraktion (P3)**  
lokale Approximation, partielle Ableitungen, Krümmungsmerkmale, diskrete Flächentypen
- (P5) **Grob- und Feinspezifikation der Wiedererkennung**  
Primärbericht,  
Details zur Objekt- und Szenenrepräsentation und zur Zuordnung,  
Abschlußbericht Konzept (Veröffentlichung)
- (P6) **Segmentierung (P4)**  
Ausgangsdaten: Entfernungs- oder Krümmungsbild? Split-Kriterium, Merge-Kriterium,  
Randberechnung, parallele Transputer-Konfiguration, Experimente mit LADAR-Szenen
- (P7) **Objekt- und Szenenrepräsentation (P5)(P6)**  
Materialvektor/Phantomflächen,  
Errechnung der benötigten Flächenattribute (Krümmungshistogramm, Flächeninhalt, Schwerpunkt,  
Hauptorientierung) und Flächenrelationen (Kantenübergänge, Schnittwinkel, ...)  
Transputer-Konfiguration, Experimente mit LADAR-Szenen
- (P8) **Fusion von Objektansichten und Teilszenen (P6)**  
Flächenkongruenzkriterium, Randanalyse, Flächenvereinigung, transitive Hüllenbildung,  
Transputer-Konfiguration, Experimente mit Teilansichten von Werkstücken und Szenen
- (P9) **Auswahl der Entwicklungsumgebung für die Wiedererkennung (P5)**  
LISP, PROLOG-Umgebungen, XPS-Schalen, Constraint-Propagierung, CAD-Schnittstellen,  
Beschaffung und Installation
- (P10) **Prototypimplementierung der Wiedererkennung in der Entwicklungsumgebung (P5)(P9)**  
interne Objektrepräsentation, Beschränkungen, Propagierung der Beschränkungen, Lagetransformation,  
Backtracking, Reihenfolgekriterien, Ausgabe
- (P11) **Erzeugung synthetischer Testdaten für die Wiedererkennung (P5)**  
einfache CAD-Modellobjekte, Erzeugung bzw. Umwandlung der Oberflächenrepräsentation,  
segmentierte Ansichten einfacher CAD-Musterszenen, einfaches Sensormodell
- (P12) **Funktionstest der Wiedererkennung (P10)(P11)**  
Erkennungsverfahren in der Entwicklungsumgebung implementiert (P10),  
Test mit synthetischen Daten (P11)
- (P13) **Revision und Erweiterung der Wiedererkennung (P6)(P7)(P12)**  
Anpassung der Objektrepräsentation gestützt auf Erfahrungen mit der Merkmalsberechnung (P6),(P7)  
Verbesserung des Erkennungsalgorithmus gestützt auf synthetische Daten (P12)  
Erweiterung der Ausgabe (Mehrdeutigkeit, Genauigkeit, Signifikanz, Erklärung, Diagnose)  
Bedienoberfläche (Editieren von Modellobjekten, -relationen,...)

- (P14) Erprobung und Tauglichkeitsnachweis der Wiedererkennung (P8)(P13)**  
 an realen, segmentierten 2D LADAR Szenen,  
 Evaluierung der Objektrepräsentation und der Leistungsdaten des Erkennungssystems  
 Experimente parallel im KfK-HTL und beim Auftragnehmer
- (P15) Konzept zur Parallelisierung der Wiedererkennung (P10)**  
 MIMD-Architektur, nachrichtengekoppelt, CSP/OCCAM-Programmiermodell,  
 mittel- bis grobkörniger Parallelismus
- (P16) Übergabe und Abschluß der Wiedererkennung (P14)(P15)**  
 Abschlußbericht, Veröffentlichung(en), Bedienerhandbuch, SW-Übergabe, Vorführungen
- (P17) Planung, Beschaffung, Aufbau und Inbetriebnahme des Transputer-Zielsystems (P14)(P20)**  
 19"-System mit T9000, C104, Fast Data Distribution Bus (z.B. MC-3)
- (P18) Portierung der Objektrepräsentation/Erkennung (P15)(P17)(P20)**  
 Schnittstellen für die Modellobjekt- und Szene-Repräsentation im Zielsystem,  
 Implementierung einer Spezialversion der Wiedererkennung
- (P19) CAD-Integration, geschlossener Kreislauf Rekonstruktion<->Wiedererkennung (P14)(P20)**  
 Programmierschnittstellen für die Objektrepräsentation (P7) im CAD-System,  
 Konvertierungsfunktionen für CSG/B-rep, Erweiterungen der Repräsentation (Attribut- und  
 Relationenrechnung)  
 Verwaltung der Modellobjekte, Bedieneroberfläche (Editierfunktionen)
- (P20) Spezialisierung auf Anwendung(en) (P7)(P12)**  
 Mögliche Anwendungen:  
 Kranautomatisierung für Containerhafen, Kanalinspektion, Sortierung von Abfallprodukten, 3D-  
 Modellierung von Mikrostrukturen, Inspektion/Wartung in kerntechnischem Bereich  
 Anforderungsanalyse:  
 Sensorik (Reichweite, Genauigkeit, Auflösung, Datenrate, Meßprinzip und phys. Umwelt)  
 Randbedingungen der Erkennung (Umwelt, Objekte, Zweck und Umfang, Genauigkeit,  
 Zeitbedingungen)  
 Integration in die Gesamtarchitektur und -steuerung, z.B. autonomes Fahrzeug
- (P21) Einbindung neuer Entfernungssensoren (P20)**  
 z.B. 3D LADAR, Sensoren im µm-Bereich, ...

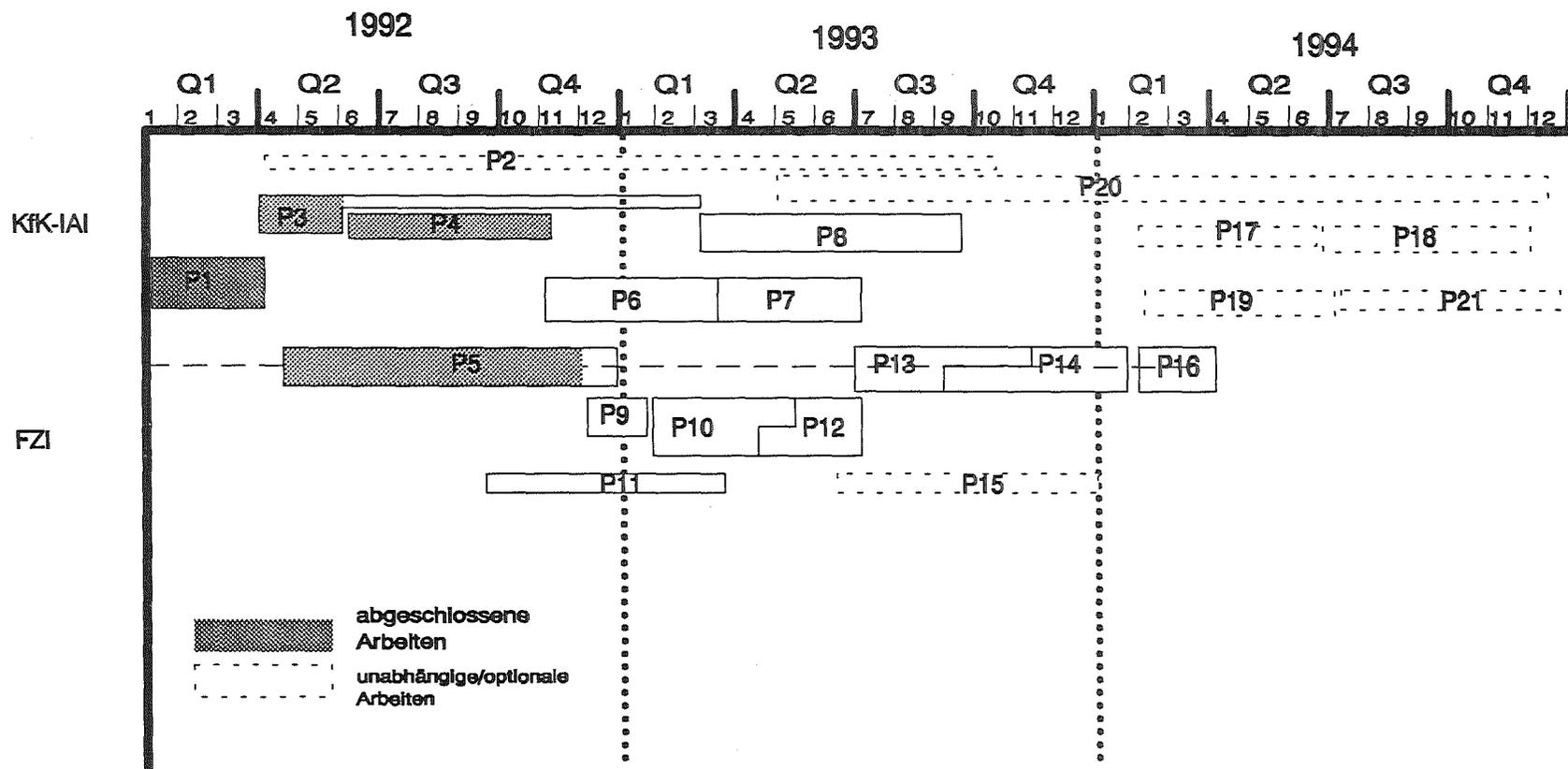


Abb. 3.1: Projektzeiplan SOMBRERO

## 3.2 Objektwiedererkennung

### 3.2.1 Entwicklungsauftrag

Zu entwickeln ist ein Algorithmus für die Wiedererkennung starrer dreidimensionaler Objekte beliebiger Lage in 3D Szenen anhand von Geometriemodellen. Die Objekte können unter der Sensoransicht teilweise verdeckt sein oder bündig anliegen. Auch mit unbekanntem Objekten ist zu rechnen. Sowohl die Szene als auch die Modelle sind durch dieselbe Objektrepräsentation gegeben, die aus Oberflächen mit bestimmten Attributen bzw. Merkmalen und aus Relationen besteht. Ausgabe des Algorithmus sind der Typ, die Position und die Orientierung für jedes in der Szene erkannte Objekt; mehrdeutige Interpretationen sind anzugeben.

Die Arbeitspakete von 3.1, die direkt in den Bereich des Entwicklungsauftrags fallen, werden erläutert, soweit sie nicht selbsterklärend sind.

#### 3.2.1.1 Grob- und Feinspezifikation (Arbeitspaket (P5))

##### a) Objektrepräsentation (Szene und Modellobjekte)

Auf der Grundlage dieses Berichtes sind im Detail festzulegen:

- die Attribute der Oberflächen, ihre Relationen, sowie Relationen zwischen verschiedenen Objekten
- die Operationen auf der Objektrepräsentation (z.B. Einfügen und Entfernen von Modellobjekten, Zugriffsfunktionen für Attributwerte und Relationen, Sortierfunktionen, Editierfunktionen,...)
- die konkrete Syntax (Daten- und Dateiformate), in der der Erkennungsalgorithmus die Szenen- und ModellobjektDarstellungen erwartet und in der diese für den Test an realen Szenen bereitzustellen sind.

Die Objektrepräsentation als solche muß leicht änderbar und erweiterbar sein. Attribute und Relationen können entfallen, wenn sich herausstellt, daß sie

- nicht oder sehr schwer aus den Sensordaten ableitbar
- nicht invariant, d.h. nicht hinreichend robust gegenüber Lageänderung, Verdeckung, Meß- und Vorverarbeitungsfehlern, oder
- wenig hilfreich für die Erkennung sind, weil ihr Rechenaufwand zu hoch oder ihr Nutzen als Unterscheidungsmerkmal zu gering ist.

Dann muß es möglich sein, andere an ihre Stelle zu setzen. Einzelne Attributwerte von Szeneflächen können undefiniert sein, wenn die Sensordaten ungenügend sind.

##### b) Erkennungsalgorithmus

Auf der Grundlage von 2.2, 2.3 ist das Erkennungsverfahren zu spezifizieren. Dies umfaßt insbesondere

- den Lösungs- bzw. Suchraum für die Zuordnung
- die invarianten Beschränkungen
- die Methoden zur effizienten Propagierung der Beschränkungen
- die Berechnung von Lage-Transformationen und Auswertung lageabhängiger Beschränkungen
- die interne Modularisierung
- Strategien und Reihenfolgen
- die Ein- und Ausgabekomponenten einschließlich Bedieneroberfläche.

Das kombinierte Verfahren aus diskreter Relaxation und Hypothesengenerierung in Abschnitt 2.4 ist ein Vorschlag von KfK-IAI, der eine Ausgangsbasis bildet, aber nicht bindend für den zu entwickelnden Algorithmus ist.

Zu erfüllen sind in jedem Fall die in den Abschnitten 1.3, 1.4 und Kapitel 2 dargelegten Anforderungen an die Lösung.

Der Punkt a) wird in enger Abstimmung mit dem Auftraggeber (KfK-IAI) bearbeitet.

### 3.2.1.2 Auswahl der Entwicklungsumgebung (Arbeitspaket (P9))

Eine Entwicklungsmaschine, die sich zur Implementierung der Objektrepräsentation und des Erkennungsalgorithmus eignet, und die Sprachen, das Laufzeitsystem, sowie Werkzeuge für Entwurf, Analyse, Test und Projektverwaltung umfaßt, ist auszuwählen und gegebenenfalls zu beschaffen. Prinzipiell kann man in 2 Richtungen denken:

#### a) CAD-Arbeitsstation (z.B. Silicon Graphics Personal IRIS, CAD-System ROBCAD)

Diese Ausrichtung käme einem langfristigen Projektziel entgegen: der funktionellen Erweiterung heutiger CAD-Systeme um die automatische Rekonstruktion von Geometriemodellen aus Sensordaten, und die Wiedererkennung für die off-line oder on-line-Programmierung. Diese Wahl bietet auch einige praktische Vorteile:

- (+) Eine Oberflächenrepräsentation (B-rep) existiert in vielen CAD-Systemen bereits.
- (+) Modellobjekte lassen sich zu Testzwecken grafisch konstruieren, und alle Arbeitsschritte dreidimensional visualisieren.
- (+) Umfangreiche Bibliotheken für numerische und geometrische Berechnungen existieren.

Doch überwiegen die Einschränkungen und Nachteile:

- (-) Existierende Oberflächendarstellungen (B-rep's) haben oft herstellerspezifische Einschränkungen, semantischer und syntaktischer Art.
- (-) In jedem Fall sind Erweiterungen zu implementieren (Attribute, Relationen). Daher stellt sich das Problem, ob geeignete Programmierschnittstellen oder der Quellcode verfügbar sind?
- (--) Es existiert keine Unterstützung für die wissensbasierte Programmierung (z.B. Propagierung von Beschränkungen)!

#### b) KI-Arbeitsstation

Diese Wahl ist für die Pilotentwicklung vorzuziehen. Das Geometriemodell muß aber selbst implementiert werden. Sollen Testkörper grafisch konstruiert und Erkennungsergebnisse grafisch visualisiert werden, sind geeignete Datenschnittstellen zu einem CAD-System über Prä- bzw. Postprozessoren zu schaffen. Wesentliche Auswahlkriterien sind die Wissensrepräsentation und Mechanismen für 'constraint propagation'.

Einige Alternativen sind

- **Expertensystem- oder Blackboard-Schalen** (z.B. KEE, Knowledge Craft, General Blackboard Architecture -GBB)  
Diese Schalen bieten vorgefertigte Mechanismen für Wissensrepräsentation, Inferenzmechanismen, Propagierung, Objekthierarchien, Relationen etc. 'oberhalb' der Entwicklungssprachen, aber sind sie flexibel und leistungsfähig genug für unseren Zweck?
- **funktionale Sprache (LISP)**  
Hier würde man die Objektrepräsentation und die Zuordnung komplett in LISP entwickeln. Vorteil wäre ein sehr flexibles, offenes System mit Optimierungsmöglichkeiten, Nachteil ein hoher, prozeduraler oder imperativer Sprachen wie Turbo PASCAL oder C vergleichbarer Implementierungsaufwand.
- **deklarative Sprache (PROLOG 'mit constraints')**  
Die Vorteile sind das hohe Abstraktionsniveau, das logische Fundament der Sprache und die eingebauten backtracking-Möglichkeiten. Sie versprechen den geringsten Implementierungsaufwand unter den genannten Alternativen. Allerdings sollte die Propagierung von Beschränkungen unterstützt werden (PROLOG-Erweiterung). Gibt es ausgereifte Produkte? Produktentwicklungen laufen z.B. bei SIEMENS-NIXDORF München.

Um die Klärung dieser Fragen geht es also im wesentlichen bei diesem Arbeitspaket. Abstimmung mit dem Auftraggeber ist erforderlich, da die Umgebung zur Weiterentwicklung des Prototyps auch im KfK-IAI genutzt werden soll.

### 3.2.1.3 Synthetische Testdaten (Arbeitspaket (P11))

Das Erkennungsverfahren soll zuerst an synthetischen Testdaten getestet werden, da

- segmentierte LADAR-Szenen mit den Merkmalen, wie sie von der Objektrepräsentation verlangt werden, noch nicht zur Verfügung stehen,
- spezielle Probleme, etwa die Robustheit einzelner Merkmale gegenüber Meßfehlern und Rechenungenauigkeiten zugunsten eines klaren, wissensbasierten Gesamtkonzeptes zunächst ausgeklammert werden sollen.

Bereitzustellen sind synthetische Oberflächendarstellungen von

- einfachen **Modellobjekten** (z.B. Quader, Zylinder, Kegel) als Testkörper,
- **Musterszenen** angeordneter Modellobjekte,

und zwar in dem Datenformat (3.2.1.1), in dem später auch die LADAR-Szenen vorliegen.

Folgende Strategie scheint zur Gewinnung der Testdaten gangbar (vgl. Abb. 3.2).

Die Modellobjekte werden zunächst an einem CAD-System konstruiert, welches (auch) über eine B-rep-Darstellung verfügt. Ihre Oberflächendarstellung (surfaces) wird ausgegeben, mit Hilfe eines Postprozessors um die benötigten Attribute und -Relationen erweitert und in das Datenformat der Objektrepräsentation transformiert. Das Ergebnis ist für jeden Testkörper ein ansichteninvariantes 3D-Modell bezüglich seines objektzentrierten Koordinatensystems.

Die Testkörper können im CAD-System individuell verschoben/verdreht und ggf. parametrisiert zu einer kollisionsfreien Szene angeordnet werden. Es sollte gelingen, die Oberflächendarstellung für die gesamte Szene, wie für jeden Testkörper, zu erhalten. Diese ansichteninvariante Darstellung berücksichtigt aber noch nicht, welche Flächen vom Blickpunkt des Entfernungssensors aus sichtbar sind. Um eine oberflächenorientierte **Sensoransicht** der Szene zu generieren, bieten sich die in Abb. 3.2 skizzierten 2 Wege an:

**Weg 1** Mit einem **simulierten Entfernungssensor** wird die CAD-Szene **pixelweise** abgetastet. Die Sensordaten werden wie beim realen Entfernungssensor vorverarbeitet und segmentiert. Der Segmentierungsalgorithmus muß also, wie für reale LADAR-Szenen, bereits zur Verfügung stehen. Dies wäre bei folgendem Weg unnötig:

**Weg 2** Aus der Oberflächendarstellung der Szene wird durch eine Art hidden-surface-Algorithmus **direkt** eine oberflächenorientierte Sensoransicht erzeugt, ohne den Umweg der pixelweisen Abtastung und erneuten Erzeugung von Flächen mittels Segmentierung zu gehen. Dafür sollte ein sehr einfaches Sensormodell - im wesentlichen nur eine (perspektivische?, Parallel-?) Projektion - zunächst ausreichen.

### 3.2.1.4 Revision, Erweiterung und Erprobung (Arbeitspakete (P13), (P14))

Nach dem Funktionstest der Wiedererkennung soll der Prototyp überarbeitet werden. Dabei sind nicht nur die erkannten Fehler und Mängel zu beseitigen, sondern auch folgende Punkte aufzunehmen:

- **Revision der Objektrepräsentation:**  
haben sich die Attribute, Merkmale und Relationen als effektiv für die Wiedererkennung erwiesen, sind sie aus den LADAR-Daten ausreichend genau und mit vertretbarem Rechenaufwand berechenbar? Hier fließen die Erfahrungen aus dem Funktionstest (P12) und aus der Vorverarbeitung (P6),(P7) ein.
- **Verbesserung und Erweiterung des Erkennungsalgorithmus:**  
Spätestens jetzt soll die **Ausgabe** der Erkennung im vollen Funktionsumfang (1.3.4) realisiert werden, d.h. mehrdeutige Interpretationen für Objektidentität und -lage, Signifikanzmaße für die Erkennung, Genauigkeit der Lagetransformation bei gegebener Genauigkeit der Eingabedaten, Erklärungs- und Diagnoseinformationen ausgegeben werden.

### Erweiterung der Bedieneroberfläche:

Hier ist vor allem eine bequemere Handhabung der Modellobjekt-Datenbasis anzustreben. Wie spezifiziert der Bediener Relationen zwischen verschiedenen Modellobjekten, z.B. stabile Lagen, Vorzugslagen?

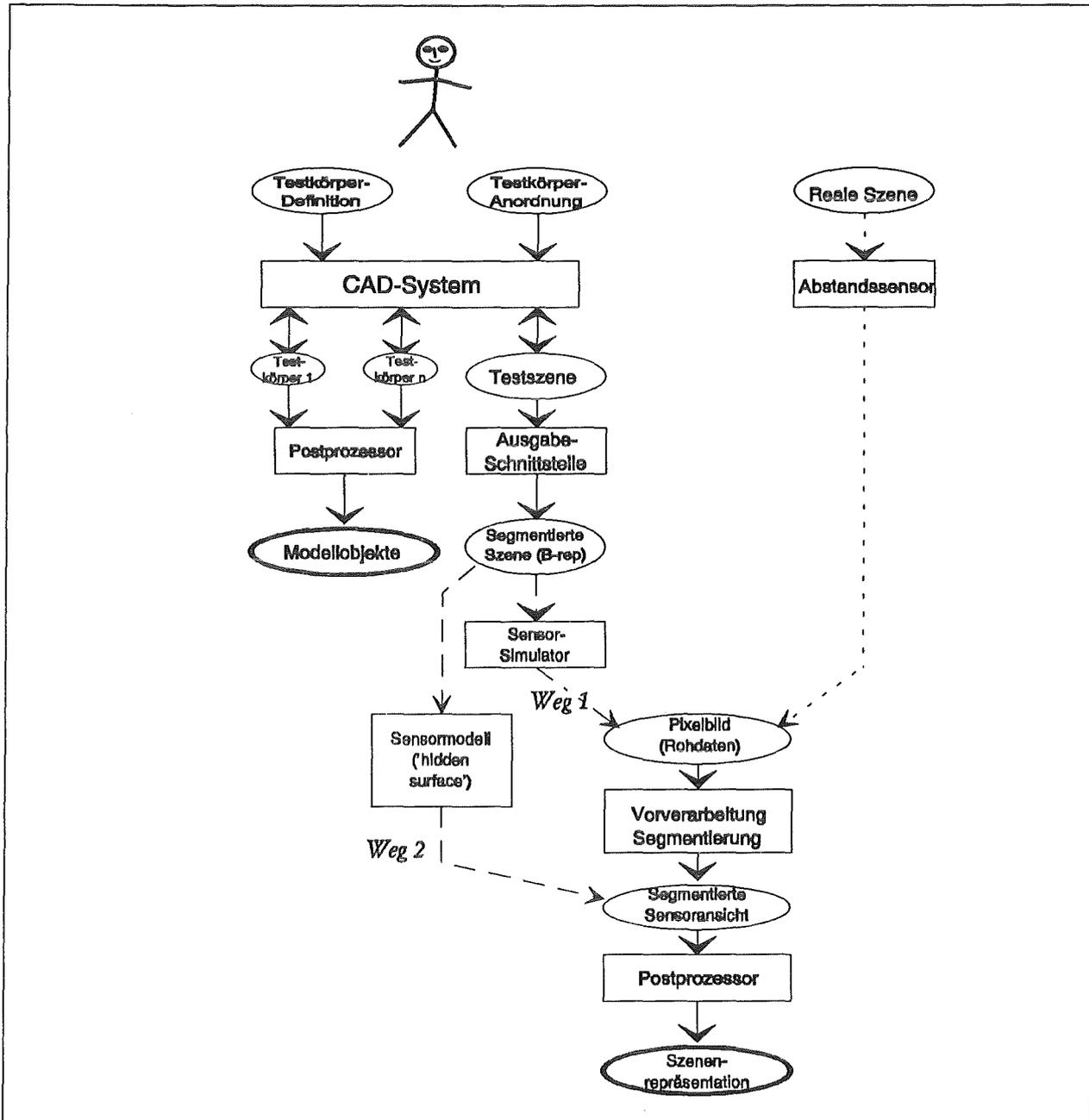


Abb. 3.2 Strategien zur Testdatenerzeugung

Der überarbeitete Prototyp wird dann an realen LADAR-Szenen im KfK erprobt. Gegenüber dem Funktionstest an simulierten Daten kommen folgende Aspekte hinzu:

- die Komplexität realer Werkstücke und Szenen, unbekannte Objekte,
- die Grenzen der Auflösung des Entfernungssensors,
- Ungenauigkeiten und Fehler durch die Meßdatenerfassung, Vorverarbeitung, Segmentierung und Attributberechnung. Diese können entweder nur in der Szenedarstellung auftreten, oder auch in den Modellobjekten, wenn diese selbst aus Entfernungsbildern rekonstruiert werden. Diese Diskrepanzen

werden im günstigsten Fall nur höhere Toleranzschwellen für die invarianten Beschränkungen erfordern, im schlimmsten Fall andere Merkmale und andere Beschränkungen.

Ferner sollen die Leistungsdaten (Trefferquote, Lagegenauigkeit, Antwortzeit,...) quantitativ ausgewertet werden.

### 3.2.1.5 Parallelisierung der Wiedererkennung (Arbeitspaket (P15))

Längerfristig soll die Objekterkennung mit einem übergeordneten fertigungstechnischen Prozeß oder einer autonomen Fahrzeugsteuerung on-line gekoppelt werden. Neben den für jede Anwendung individuell zu berücksichtigenden Antwortzeitbedingungen muß es ein generelles Ziel sein, die heute oft im Minutenbereich liegenden Erkennungszeiten einfacher, für den Menschen "auf einen Blick" zu erfassender Szenen drastisch zu reduzieren. Dies erfordert - neben der Entwicklung neuer Datenstrukturen und Algorithmen - den Einsatz massiv paralleler Systeme. Einige Ideen für parallele Algorithmen zur Segmentierung und Erkennung wurden in 1.5.3, 2.4.1 und 2.4.3 erwähnt. Wir haben dabei folgende Zielrechnerarchitekturen im Auge:

- MIMD-Rechner, Parallelismus mittel- bis grobkörnig,
- nachrichtengekoppeltes Programmiermodell mit synchroner 1:1 oder m:n-Kommunikation (CSP, OCCAM2),
- T9000-basierendes Transputersystem.

Als Vorstufe zur eigentlichen Zielsystementwicklung (P17)(P18) erwarten wir im Rahmen des Entwicklungsauftrags ein detailliertes Konzept zur Parallelisierung des in (P10) entwickelten Erkennungsalgorithmus. Das Gesamtsystem besteht dabei aus 2 virtuellen Maschinen:

1. der Entwicklungsmaschine inclusive Laufzeitsystem
2. der Anwendung, also dem eigentlichen, mit den Konzepten von 1. entwickelten Erkennungsalgorithmus.

Demzufolge gibt es 2 prinzipiell verschiedene Wege zur Parallelisierung:

- (A) Neuimplementierung des **Gesamtsystems** auf der Basis der verteilten Zielmaschine (Transputer-Netz)
- (B) Verteilte Realisierung der **Entwicklungsmaschine**, z.B. Verwendung einer parallelen PROLOG-Maschine PARLOG statt einer sequentiellen. Die Anwendung selbst könnte bei dieser Lösung unverändert fortbestehen und würde nur parallel abgearbeitet.

Dem höheren Entwicklungsaufwand steht ein flexibleres, offenes System bei (A) gegenüber. Bei (B) ist man an die Kommunikationsschnittstellen der Entwicklungsmaschine gebunden. Die Koppelung mit der Datenvorverarbeitung könnte sich als Engpaß erweisen.

### 3.2.2 Meilensteine

Der Entwicklungsauftrag wird in 2 Phasen unterteilt:

#### Phase 1

- |                  |  |
|------------------|--|
| <b>Laufzeit:</b> | August '92 bis Ende Juni '93   |
| <b>Ergebnis:</b> | - Erkennungsalgorithmus in Entwicklungsumgebung                          |
|                  | - prinzipieller Funktionsnachweis an einfachen, synthetischen Testszenen |

#### Phase 2 (Folgauftrag)

- |                  |  |
|------------------|--|
| <b>Laufzeit:</b> | September '93 bis Ende März '94  |
| <b>Ergebnis:</b> | - Erprobung und Tauglichkeitsnachweis der Objekterkennung an realen, segmentierten LADAR-Szenen im KfK |
|                  | - Konzept zur Parallelisierung des Erkennungsalgorithmus   |

Die Aufteilung macht Phase 1 unabhängig von der Verfügbarkeit realer, segmentierter Testszenen (kritisch sind hierbei die Arbeitspakete (P6), (P7) und (P8)) und gibt uns die Möglichkeit, den Folgauftrag von den erzielten Ergebnissen und ggf. neuen, anwendungsbezogenen Anforderungen abhängig zu machen.

Für die beiden Phasen und ihre Arbeitspakete werden folgende Meilensteine definiert:

*Phase 1:*

<i>Gegenstand (Arbeitspaket)</i>	<i>Art des Abschlusses</i>	<i>Aufwand (ca.)</i>	<i>Termin</i>
<u>1.Grob- und Feinspezifikation (P5)</u> - Objektrepräsentation: Szene, Modellobjekte - Zuordnungsalgorithmus  (gemeinsam mit KfK)	Abschlußbericht Konzept (Veröffentlichung) (KfK-IAI/FZI)	4 MM	Dez. 92
<u>2.Auswahl und Beschaffung der Entwicklungsumgebung (P9)</u>	ggf. Beschaffung Vorführung	1 MM	Jan. 93
<u>3.Prototyp-Implementierung (P10)</u> (Zuordnungsalgorithmus)	Dokumentation	4 MM	April 93
<u>4.Vorbereitung der synthetischen Testdaten (P11)</u>	-	3 MM	März 93
<u>5.Funktionsnachweis (P12)</u> (Test an synthetischen Testdaten)	- Demo (FZI) - Übergabe - Ergebnisbericht	1.5 MM	Juni 93

*Phase 2 (Zeiten für Beginn Sept. 93, tatsächlicher Arbeitsbeginn noch offen) :*

<i>Gegenstand (Arbeitspaket)</i>	<i>Art des Abschlusses</i>	<i>Aufwand (ca.)</i>	<i>Termin</i>
<u>1.Revision und Erweiterung der Objektrepräsentation und des Zuordnungsalgorithmus (P13)</u>	Dokumentation	3 MM	Okt. 93
<u>2.Test an realen segmentierten Szenen (P14)</u>  (KfK und FZI gemeinsam)	- Versuchsprotokoll - Bericht - Vorführungen	3.5 MM	Jan. 94
<u>3.Parallelisierungskonzept (P15)</u>	- Bericht	3 MM	Jan. 94
<u>4.Übergabe und Abschluß (P16)</u>	- Abschlußbericht/ Veröffentlichung (KfK-IAI/FZI) - Bedienerhandbuch - SW-Übergabe	2.5 MM	März 94

## 4. Zusammenfassung und Ausblick

Die wesentlichen Komponenten zur Rekonstruktion und Wiedererkennung von Objekten aus 2D LADAR Entfernungsbildern sind in Arbeit oder im Entwurf. Endgültige Lösungen existieren aber noch nicht. Der momentane Erkenntnisstand und das weitere Vorgehen lassen sich wie folgt zusammenfassen:

### 1. Datenerfassung

Der 2D LADAR zusammen mit dem Portalroboter reicht aus, um komplexe 3D-Szenen in kartesischen Koordinaten aufnehmen. Die Qualität der Rohdaten wurde durch die Scan-Mittelung und insbesondere durch Median-Filterung wesentlich verbessert.

Doch muß, neben einer Glättung, die Qualität der Rohdaten schon an der Quelle verbessert werden, um wirklich industrielle Werkstücke erkennen zu können. Zumindest müssen die Ursachen der Meßfehler genauer bekannt sein. Die prinzipiellen, physikalisch-technisch bedingten Ungenauigkeiten der Laufzeitmessung, Fehler in der zeitlichen Korrelation zwischen Winkel und Abstand, auch ein möglicher Gerätedefekt sind dabei zu untersuchen oder auszuschließen. Ferner sollte die Horizontalauflösung der Meßpunkte (Winkelinkrement  $0,6^\circ$ ) verbessert werden.

Die detaillierte experimentelle Untersuchung der **Oberflächeneinflüsse** steht ebenfalls noch aus.

In Zukunft soll mit den kartesischen Meßpunkten auch das **Intensitätssignal** sowie eine **Sensororientierung** (pro Höhenprofil) übertragen werden, um die Signalgüte, Abschattungen etc. besser auswerten zu können.

Weitere Arbeiten, wie die Kalibrierung der Transformation oder die Handmontage mit vollen Orientierungsfreiheitsgraden sind ebenfalls wünschenswert, aber nicht so dringlich.

### 2. Krümmungsberechnung

Die **Gauß'sche** und die **mittlere Oberflächenkrümmung** und die daraus abgeleiteten diskreten Flächentypen bilden die direkte Vorstufe zur Segmentierung und zur Charakterisierung gekrümmter Oberflächen durch Histogramme. Diese Merkmale wurden auf Transputer in OCCAM 2 implementiert, an synthetischen Daten überprüft und auf einfache reale Szenen angewandt. 2 Probleme traten dabei zutage:

- a) die hohe Empfindlichkeit gegenüber **Rauschen**
- b) die geringe Aussagekraft, um Konturen und insbesondere Sprungkanten im Entfernungsbild zu erkennen, bedingt u.a. durch die lokale **Approximation** durch orthogonale Polynome.

Eine verbesserte Qualität der Rohdaten (siehe 1.) hilft, Problem a) zu entschärfen. Weitere Krümmungsmerkmale, zum Beispiel die Hauptkrümmungen, sind zu untersuchen, um eine bessere Konturerkennung und bessere Invarianzeigenschaften zu erreichen.

Notfalls müssen andere Wege zur Krümmungsberechnung als die lokale Approximation versucht werden, etwa die in /46/ vorgeschlagene, wesentlich aufwendigere Approximation durch globale Minimierung von Energiefunktionen.

### 3. Objektrepräsentation

Wir haben eine einheitliche **Volumenrepräsentation** von Objekten und Szenen durch die sichtbaren **Oberflächen** und ihre **Attribute** und **Relationen** definiert. Eine solche Darstellung ist für die Wiedererkennung aussagekräftiger als z.B. Kantenmodelle oder ansichtenorientierte Modelle. Sie ist CAD-gerecht, vereint im Vergleich zu CSG- oder octree-Modellen am besten die Anforderungen nach Wiedererkennbarkeit und automatischer Rekonstruierbarkeit, und kann geometrisches Wissen über die Anwendung wie die Form der Objekte oder die Vorzugslagen in expliziter Form beschreiben. Den zu erwartenden Schwierigkeiten zum Trotz halten wir an dem Ziel, aus verrauschten Entfernungsbildern Flächen zu extrahieren, fest.

Details der Darstellung, z.B. Art und Ordnung der Approximation, Definition von Größen- und Formattributen von Oberflächen sind noch festzulegen, ferner die Syntax und die benötigten Hilfsoperationen auf Szenen- und Modellobjektrepräsentationen. Die Repräsentation bildet die gemeinsame Schnittstelle zwischen Vorverarbeitung und Wiedererkennung, an der getrennte Teams (KfK-IAI und FZI) arbeiten.

#### 4. Vorverarbeitung

Trotz umfangreicher Literatur sowie Vorarbeiten unter POMOS bleibt als wichtigstes Ziel in diesem Bereich, neue und leistungsfähige **Segmentierungsalgorithmen** für Abstandsbilder zu entwickeln und zu erproben. Diese müssen vollautomatisch und ohne manuelles Nacheditieren, robust unter verrauschten Entfernungswerten und Krümmungswerten (siehe 1. und 2.), und als globale, bildüberspannende, rechenaufwendige Operation massiv parallel ausgelegt sein. 3D Tetraederdekomposition im Krümmungsraum ('split and merge') ist der derzeit aussichtsreichste Kandidat. Erste Messungen zur hierarchischen Parallelisierung von Zerlegungsalgorithmen ergaben eine Beschleunigung von ca. 3 bei 4 Arbeitsprozessoren.

Da Krümmungsextraktion und Segmentierung die  $2\frac{1}{2}$ D-Eigenschaft erfordern, die komplexe Werkstückoberflächen selten besitzen, besteht das Problem, mehrere  $2\frac{1}{2}$ D-Teilansichten zu einer Gesamtoberfläche zu fügen. Die Idee besteht hier darin, mehrere Teilansichten unabhängig voneinander zu verarbeiten und erst am Ende die **Oberflächendarstellungen** zu **vereinen**, statt verrauschte **Rohdaten** zu **sortieren** und eine gemeinsame Punktwolke zu erstellen. Der Algorithmus für dieses Fusionierungsproblem ist aber noch im Detail zu entwickeln und zu erproben.

#### 5. Wiedererkennung

Wiedererkennung einer Szene bedeutet **Zuordnung** ihrer Oberflächen zu Modellobjektoberflächen und -Instanzen unter Beachtung der **geometrischen Beschränkungen**.

Dabei benutzen wir ein Instrumentarium an **invarianten**, also ohne Kenntnis der Objektlage anwendbaren Beschränkungen, wie Krümmungshistogramm-Inklusion, Nachbarschafts-, Winkel-, Größen- und Flucht(linien, ecken)-Beschränkungen. Da eine einzige Beschränkung möglicherweise nicht robust genug oder in einer speziellen Anwendung nicht anwendbar ist, verfolgen wir anders als die meisten Objekterkennungssysteme einen diversitären Ansatz.

Daneben sind **lageabhängige** Beschränkungen anzuwenden, sobald eine Lagetransformation berechenbar wird. Zu untersuchen ist, unter welchen Bedingungen und wie dies möglich ist.

Die gesamte Zuordnung führt zu einem Suchbaumverfahren mit exponentiellem Aufwand. Grobgestrickte skalare Bewertungsheuristiken oder Gütefunktionen mit dem Ziel, große Teile des Suchbaums von vornherein ausschließen, führen zu schwer nachvollziehbaren und in ihrer Leistung schwer beurteilbaren Lösungen. Wir versuchen vielmehr, die Komplexität des Suchraums durch eine Kombination von Relaxation und Hypothesengenerierung sowie durch Parallelverarbeitung zu beherrschen. Auch off-line Sortieren der Modellobjekte und ihrer Beschränkungen kann dazu beitragen.

Die Frage, ob die Zerlegung der Szene in disjunkte Objektinstanzen in den Suchbaum integriert oder besser separat behandelt wird, kann noch nicht endgültig beantwortet werden.

## 5. Literatur

- /1/ Andress K.M., Kak A.C.  
Evidence Accumulation and Flow of Control in a Hierarchical Spatial Reasoning System  
AI Magazine, Vol. 9 (2), 1988, pp.75-94
- /2/ Besl P.J.  
Surfaces in range image understanding  
Springer Series in Perception Engineering, Ed. R.C.Jain, Springer Verlag, New York, 1988
- /3/ Besl P.J.  
Geometric Signal Processing  
In: Analysis and Interpretation of Range Images, Jain R.C., Jain A.K. (ed.)  
Springer Series on Perception Engineering, New York, 1990, pp.141-205
- /4/ Besl P.J., Jain R.C.  
Three-dimensional object recognition  
ACM Computing Surveys Vol. 17 (1), März 1985, pp. 75-145
- /5/ Bhanu B.  
Representation and Shape Matching of 3-D Objects  
Transactions on Pattern Analysis and Machine Intelligence, PAMI Vol. 6 (3), Mai 1984, pp. 340-350
- /6/ Boissonat J.D.  
Representation of object triangulating points in 3-D-space.  
Proc. of the 6th International Conference on Pattern Recognition, München, 19.-22.10.1982, pp.830-832
- /7/ Bolles R.C., Horaud P.  
3DPO: A Three-Dimensional Part Orientation System  
The International Journal of Robotics Research, Vol. 5(3), 1986, pp. 3-26
- /8/ Brooks R.A.  
Model-Based Three-Dimensional Interpretations of Two-Dimensional Images  
IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 5(2), März 1983, pp. 140-150
- /9/ Chen C.H., Kak A.C.  
A Robot Vision System for Recognizing 3-D Objects in Low-Order Polynomial Time  
IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19 (6), Nov./Dez. 1989, pp.1535-1563
- /10/ Chin R.T., Dyer C.R.  
Model-based recognition in robot vision  
ACM Computing Surveys Vol. 18 (1), März 1986, pp. 67-108
- /11/ Coy M., Ogley A.M.  
Persönliche Mitteilung, Kernforschungszentrum Karlsruhe, Institut für Angewandte Informatik, 1992
- /12/ Dechter R.  
Learning While Searching in Constraint-Satisfaction Problems.  
Proc. of the 5th National Conference on Artificial Intelligence, Menlo Park, Calif., 1986, pp. 178-183
- /13/ Demel P.  
Studie über "Autonome Mobile Systeme" - Abschlußbericht für das Kernforschungszentrum Karlsruhe.  
Institut für Steuerungstechnik, Universität Stuttgart, 1992
- /14/ Dillmann R., Huck M.  
Informationsverarbeitung in der Robotik  
Springer Verlag, 1991

- /15/ Fan T.J.  
Describing and recognizing 3-D objects using surface properties  
Springer Verlag, New York, 1990
- /16/ Faugeras O.D., Hebert M.  
The Representation, Recognition and Locating of 3-D Objects  
The International Journal of Robotics Research, Vol. 5(3), 1986, pp. 27-52
- /17/ Flynn P.J., Jain A.K.  
CAD-Based Computer Vision: From CAD Models to Relational Graphs  
Transactions on Pattern Analysis and Machine Intelligence, PAMI Vol. 13 (2), Feb. 1991, pp. 114-132
- /18/ Fox M.S., Sadeh M., Baykan C.  
Constrained Heuristic Search  
Proc. of the 11th International Joint Conference on Artificial Intelligence, Menlo Park, Ca., 1989, pp. 309-315
- /19/ Freuder E., Quinn M.  
Taking advantage of Stable Sets of Variables in Constraint-Satisfaction Problems  
Proc. of the 9th International Joint Conference on Artificial Intelligence, Menlo Park, Ca., 1985, pp. 1076-1078
- /20/ Grimson W.E.L, Lozano-Perez T.  
Model-Based Recognition and Localization from Sparse Range or Tactile Data  
The International Journal of Robotics Research, Vol. 3(3), 1984, pp. 3-35
- /21/ Grimson W.E.L, Lozano-Perez T.  
Localizing Overlapping Parts by Searching the Interpretation Tree  
Transactions on Pattern Analysis and Machine Intelligence, PAMI Vol. 9(4), Juli 1984
- /22/ Häfele K.H., Hansjosten E., Kummert H., Müller S., Roesner C.  
Persönliche Mitteilung, Kernforschungszentrum Karlsruhe, Institut für Angewandte Informatik, Feb. 1992
- /23/ Häfele K.H., Kohlhepp P.  
Unveröffentlichter Bericht, Kernforschungszentrum Karlsruhe, Institut für Angewandte Informatik, Feb. 1991
- /24/ Haralick R.M., Shapiro L.G.  
The Consistent Labeling Problem: Part I  
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 1 (2), 1979, pp. 173-183
- /25/ Haralick R., Elliot G.  
Increasing Tree Search Efficiency for Constraint-Satisfaction Problems  
Artificial Intelligence 14 (3), 1980, pp. 263-313
- /26/ Hoppen P., Knieriemen T., von Puttkamer E.  
Laser-Radar based Mapping and Navigation for an Autonomous Mobile Robot  
IEEE Int. Conference on Robotics and Automation, Cincinnati, Ohio, 13.-18.5.1990, pp. 948-953
- /27/ Horn B.K.P  
Extended Gaussian Images  
Proc. of the IEEE, vol. 72, Dez. 1984, pp. 1656-1678
- /28/ Horn B.K.P, Ikeuchi K.  
The mechanical manipulation of randomly oriented parts  
Science America, vol. 251 (2), Aug. 1984, pp. 100-111

- /29/ IBEO Lasertechnik  
LADAR 2D, 2D LINEAR, 3D  
Interne Firmenschrift, Hamburg, 1990
- /30/ IBEO Lasertechnik  
3D-Scanner (Polarscanner) Software-Beschreibung  
Anhang A: Kurzbeschreibung des Bedienprogramms POLSCA10.EXE,  
Anhang B: Software Feinspezifikation  
Interne Dokumentation, Hamburg, 1991
- /31/ IBEO Lasertechnik  
Aufgabenbeschreibung zur 3D-Entfernungsbildverarbeitung  
Schreiben vom 30.8.1991, Hamburg
- /32/ Ikeuchi K.  
Precompiling a Geometrical Model into an Interpretation Tree for Object Recognition in Bin-picking tasks, Proc. of the DARPA Image Understanding Workshop, Feb. 1987, pp. 321-339
- /33/ INMOS  
occam 2 Reference Manual  
Prentice Hall International Series in Computer Science, New York, 1988
- /34/ Isele J.  
Realisierung eines Fertigungssystems zum Gußputzen mit aufgabenorientierter Programmierung  
Dissertation, Universität Karlsruhe, Institut für Rechneranwendung in Planung und Konstruktion, 1992
- /35/ Jain R.C., Jain A.K. (ed.)  
Analysis and Interpretation of Range Images  
Springer Series on Perception Engineering, New York, 1990
- /36/ Kak A.C., Vayda A.J., Cromwell R.L., Kim W.Y., Chen C.H.  
Knowledge-based robotics  
Int. J. Prod. Research, Vol. 26 (5), 1988, pp. 707-734
- /37/ Kitchen L., Rosenfeld A.  
Discrete Relaxation for Matching Relational Structures  
IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9 (12), Dez. 1979, pp.869-874
- /38/ Kohlhepp P., Haffner H.  
Rekonstruktion und Wiedererkennung dreidimensionaler Gegenstandsmodelle mit Entfernungssensoren  
8. Fachgespräch über Autonome Mobile Systeme, Karlsruhe, November 1992
- /39/ Kohlhepp P.  
Unveröffentlichte Notiz, Kernforschungszentrum Karlsruhe, Institut für Angewandte Informatik,  
September 1992
- /40/ Koprek J.  
Vorverarbeitung und Merkmalsextraktion von Entfernungsbildern mit Transputern  
Diplomarbeit an der Universität Karlsruhe, Institut für Hochfrequenztechnik und Elektronik, 1992
- /41/ Kumar V.  
Algorithms for Constraint Satisfaction Problems: A Survey  
AI Magazine, Frühjahr 1992, pp. 32-44
- /42/ Lades M., Vorbrüggen J.C., Buhmann J. et al.  
Distortion Invariant Object Recognition in the Dynamic Link Architecture  
Angenommen für IEEE Transactions on Computers, 1992
- /43/ Lang S.Y.T., Wong A.K.C.  
Map Making for an Autonomous Mobile Robot  
Quelle unbekannt, 1992

- /44/ Lawo M., Häfele K.H., Müller S.  
3-D-Punktmessung mit Industrierobotern zur Modellierung von Freiformflächenmodellen  
Fachtagung Industrieroboter messen und prüfen, Köln, 12.-13.11.1991, VDI Bericht 921, pp. 205-214
- /45/ Lee S., Hahn H.  
Recognition and Localization of 3-D Natural Quadric Objects based on Active Sensing  
IEEE International Conference on Robotics and Automation, Sacramento, Ca., 9.-11.4.1991, pp.156-161
- /46/ Li S.Z.  
Invariant Surface Segmentation Through Energy Minimization with Discontinuities  
International Journal of Computer Vision, Vol. 5 (2), 1990, pp. 161-194
- /47/ Lu H., Shapiro L.G., Camps O.I.  
A Relational Pyramid Approach to View Class Determination  
Workshop on Interpretation of 3D-Scenes, Austin, Tx., 27.-29.11.1989, pp. 176-183
- /48/ MSC (Microcomputers Systems Components Vertriebs GmbH)  
Transputer PC-Board MB1T9S-PC  
Produkt-Dokumentation, Stutensee, 1992
- /49/ Müller S.  
Erzeugung eines Freiformflächenmodells auf der Basis von 3D-Meßpunkten  
Studienarbeit, Universität Karlsruhe, Institut für Rechneranwendung in Planung und Konstruktion, 1991
- /50/ Nagel H.H., Enkelmann W.  
An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 8 (5), 1986
- /51/ Nevatia R., Binford T.O.  
Description and Recognition of Complex-Curved-Objects  
Artificial Intelligence, Vol. 8, 1977, pp. 77-98
- /52/ Onoguchi K., Watanabe M., Okamoto Y., Kuno Y., Asada H.  
A Visual Navigation System Using a Multi-Information Local Map  
IEEE Int. Conference on Robotics and Automation, Cincinnati, Ohio, 13.-18.5.1990, pp. 767-774
- /53/ Oshima M., Shirai Y.  
Object recognition using three-dimensional information  
Transactions on Pattern Analysis and Machine Intelligence, PAMI Vol. 3 (4), Juli 1983, pp. 353-361
- /54/ Reid I. et. al.  
Range vision and the Oxford AGV  
Conference on Image Processing, London, 10.-12.10.1989, pp.51-72
- /55/ Rillo A.H.  
Grouping-based recognition system  
Quelle unbekannt, 1992
- /56/ Seiler W.  
Technische Modellierungs- und Kommunikationsverfahren für das Konzipieren und Gestalten auf der Basis der Modell-Integration  
Fortschritt-Berichte VDI Nr. 49, Reihe 10:Angewandte Informatik, VDI-Verlag, Düsseldorf, 1985
- /57/ Shaffer G.K., Stentz A., Whittaker W.L., Fitzpatrick K.W.  
Position Estimator for Underground Mine Equipment  
Proc. Int. Workshop on Information Processing in Autonomous Mobile Robots, 6.-8.3.1991, München, pp. 1-10
- /58/ Stockman G.  
Object Recognition  
In: Analysis and Interpretation of Range Images, Jain R.C., Jain A.K. (eds.)

Springer Series on Perception Engineering, New York, 1990, pp. 207-253

- /59/ Taylor W.T., Savini M., Reeves A.P.  
Fast Segmentation of Range Imagery into Planar Regions  
Computer Vision, Graphics, and Image Processing, Vol. 45, 1989, pp.42-60
- /60/ Tiziani H.  
Optische Verfahren zur Abstands- und Topografiebestimmung  
Informationstechnik it 33 (1), 1991, pp. 5-14
- /61/ Trivedi M.M., Abidi M.A., Eason R.O., Gonzalez R.C.  
Object Recognition and Pose Determination in Multisensor Robotic Systems  
IEEE Conf. on systems, man, and cybernetics, Cambridge, Mass., 14.-17.11.1989, Vol. 1, pp.186-193
- /62/ Vayda A.J., Kak A.C.  
Geometric Reasoning for Pose and Size Estimation of Generic Shaped Objects  
IEEE Int. Conference on Robotics and Automation, Cincinnati, Ohio, 13.-18.5.1990, pp. 782-789
- /63/ Vogelgesang V.G.  
Persönliche Mitteilung, Kernforschungszentrum Karlsruhe, Institut für Reaktorentwicklung, Aug. 1991
- /64/ Vogelgesang V.G.  
Persönliche Mitteilung, Kernforschungszentrum Karlsruhe, Institut für Reaktorentwicklung, Nov. 1991
- /65/ Walker E.L., Herman M., Kanade T.  
A Framework for Representing and Reasoning about Three-Dimensional Objects for Vision  
AI Magazine, Vol. 9 (2), 1988, pp.47-58
- /66/ Waltz D.  
Understanding Line Drawings of Scenes with Shadows  
In: The Psychology of Computer Vision, ed. P.H.Winston, McGraw Hill, 1975, pp. 19-91
- /67/ Wei J.  
Modell-gesteuerte Szenen-Interpretation durch Fusion von Intensitäts- und Abstandsbildern  
Dissertation, Universität Karlsruhe, Fakultät für Informatik, Dez. 1989
- /68/ Zucker S.W.  
Region Growing: childhood and adolescence  
Computer Vision, Graphics, Image Processing, Vol.5, 1976, pp.382-399