

# Zusatzmaterialien zur Vorlesung Robotik

R. Der  
Universität Leipzig  
Institut für Informatik

June 19, 2003

## 1 Sensoren für autonome Roboter

Autonome Roboter sind als reaktive Systeme in weit stärkerem Maße als die Industrieroboter auf die Sensorkomponente angewiesen. Als Verbindung des AR zur Welt liefert sie die erforderlichen Informationen, mit denen der AR seine Überlebensfähigkeit absichern und sinnvolle Handlungen erfolgreich ausführen kann. Allgemein können die Sensoren eines AR zunächst in interne und externe Sensoren unterschieden werden: Interne Sensoren messen den internen Zustand des Roboters, z. B. die Radgeschwindigkeiten, die Gelenkstellungen eines Manipulators aber auch seinen Energiestatus. Externe Sensoren identifizieren die Stellung des Roboters und seiner Teile relativ zur Welt und liefern Informationen über die ihn umgebenden Objekte.

### 1.1 Klassifikation der Sensoren

Eine allgemein akzeptiertes Klassifikationsschema ergibt sich nach folgenden Gesichtspunkten (auszugsweise aus [Dillmann, 1991])

---

Klassifikation	Meßgröße	Meßverfahren
Interner S.	Gelenkposition Kräfte und Momente	Potentiometer Piezokristall Kapazität
Externe taktile S.	Druck Scherkraft Sicherheit	Fiberoptik/k. Haut Piezokristall Induktivität
Extern, nichtvisuell	Ex. von Objekt Entfernung Annäherung	Ultraschall Laser (s. 2.4) Induktivität/ Fotodiode
Extern, visuell	Konturerkennung Bewegung von Objekten Kante/Ecke/Fläche	Fernsehkamera Lineare Kamera Laserscanner

Für die AR sind optische Sensoren und die Entfernungsmesser mit Ultraschall bzw. Laserlicht besonders wichtig. Für die Realisierung von Sensoren wird eine Vielzahl von physikalischen Effekten herangezogen. Diese sollen aber nicht hier sondern getrennt im Abschnitt ??? besprochen werden.

## 2 Vorverarbeitung der Sensordaten

Der physikalische Output der Sensoren stellt die sensorischen Rohdaten oder die sensorische Primärinformation dar. Diese ist im allgemeinen nicht besonders aufschlußreich. Die Sensorkomponente umfaßt deshalb in den meisten Architekturen auch die Primärverarbeitung der Rohdaten, d. h. sie leistet die Vorverarbeitung der Sensordaten für die nachfolgenden Funktionseinheiten in der jeweiligen Architektur. Diese Verarbeitungsleistungen reichen von Entrauschung, Filterung und Reduktion der Rohdaten über primäre Merkmalsextraktion, Klassifikation und Mustererkennung bis hin zu einfachen Formen der Bildverarbeitung. Bei hierarchischen Architekturen ist auch die Sensordatenverarbeitung meist ein hierarchisch organisierter Prozeß, bei dem auf jeder Stufe der Hierarchie Filterungen, Merkmalsextraktions-, Klassifizierungs- oder Mustererkennungsprozeduren vorgenommen werden, deren Resultat den Input in die nächste Hierarchieebene darstellt.

Diese Operationen können mathematisch durch eine Funktion modelliert werden:

Sei  $\vec{x} = (x_1, \dots, x_n)$ ,  $x \in \mathcal{X}$  der Vektor der primären Sensordaten (oder Rohdaten), d. h. der Information, die unmittelbar durch die Sensoren (oder von der vorhergehenden Verarbeitungsstufe) geliefert wird. Dann ist die Verarbeitung durch die Abbildung  $G : \mathcal{X} \rightarrow \mathcal{Q}$

$$\vec{q} = G(\vec{x}), \quad \vec{q} \in \mathcal{Q} \quad (1)$$

definiert.  $\vec{q}$  stellt dabei einen verallgemeinerten Merkmalsvektor dar, dessen Komponenten numerische oder symbolische Variable (Namen von Objekten) sein können. Diese Verarbeitung erfordert i. a. zusätzlich Kontextinformation, die von der Wissenskomponente bereitgestellt wird

$$\vec{q} = G(\vec{x} \mid \vec{\kappa}), \quad \vec{q} \in \mathcal{Q}, \quad \kappa \in \mathcal{K} \quad (2)$$

wobei  $\mathcal{K}$  die Menge der Kontextvariablen ist.

**Beispiele:** (1)  $\kappa_i$  ist ein abgespeichertes Prototypmuster (s.u.), z. B. ein Baum, der Vergleich  $\vec{x}$  mit  $\vec{\kappa}_i$  liefert  $q_{Baum} = 1$ . (2) Der Laserentfernungsmesser des AR mißt die Entfernung zum nächsten Hindernissen unter zwei Winkeln  $\pm\alpha$  relativ zur Fahrtrichtung. Der Vergleich mit den in der Wissensbasis (Weltmodell) abgespeicherten objektbegrenzenden Geraden liefert für die Variable  $q_j = 1$  oder 0 je nachdem, ob die zwei Punkte zu einer der abgespeicherten Geraden gehören oder nicht, wobei  $q_j = GeradeVoraus$

Im folgenden sollen einige Beispiel für die hier allgemein benannten Prozeduren besprochen werden.

## 2.1 Odometrie

Ein wichtiger **interner** Sensor mobiler Roboter ist der Meßfühler zur Bestimmung der aktuellen Umdrehungsgeschwindigkeit des jeweiligen Rades und damit der theoretisch pro (infinitesimaler) Zeiteinheit zurückgelegten Strecke pro Rad. Diese stehen unmittelbar als Rohdaten zur Verfügung und dienen – im Sinne einer Primärauswertung – zur Bestimmung der Stellungsänderung des AR bzw. durch Aufsummation zur Bestimmung der absoluten Position bzw. Stellung des AR. Von Fehlerquellen wie Rutschen, Reifenabnutzung oder Bodenunebenheiten abgesehen, sind diese Daten oft die wichtigste Quelle zur Bestimmung der Stellung des AR im physikalischen Raum.

Für einen zweirädrigen AR sei die Position durch Angabe des Ortsvektors  $\vec{r} = (x, y)$  zum Achsmittelpunkt (AMP) festgelegt, zur Definition der Orientierung im Raum führen wir noch den Achsvektor  $\vec{a} = (a_1, a_2)$  (Einheitsvektor,

d. h.  $||\vec{a}|| = 1$ ) ein, der vom AMP zum linken Rad des AR zeigt, s. (??). Die Orientierung des AR ist dann durch den Winkel  $\alpha$  von  $\vec{a}$  mit der  $x$ -Achse gegeben, d. h.  $\cos \alpha = \vec{a} \cdot \vec{e}_x$  wenn  $\vec{e}_x$  Einheitsvektor längs der  $x$ -Achse ist<sup>1</sup>. Explizit ist

$$\vec{a} = \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} \quad (3)$$

Der Einheitsvektor  $\vec{g}$  soll senkrecht auf der Achse stehen ( $\vec{a} \cdot \vec{g} = 0$ ) und in Richtung zum Bug des AR zeigen, folglich

$$\vec{g} = \begin{pmatrix} \sin \alpha \\ -\cos \alpha \end{pmatrix} \quad (4)$$

Ist  $R$  der Achsradius ( $2R$ =Abstand zwischen den Rädern), so ist der Ortsvektor zum linken bzw. rechten Rad

$$\vec{r}_l = \vec{r} + R\vec{a} \quad (5)$$

$$\vec{r}_r = \vec{r} - R\vec{a}$$

Nehmen wir nun an, daß durch Motoransteuerung über einen Zeitraum, in dem wir die Geschwindigkeiten der Räder als konstant annehmen können, die Räder die Strecken  $s_l$  und  $s_r$  zurücklegen<sup>2</sup>. Sind diese gleich, d.h.  $s_l = s_r = s$  so gibt es keine Richtungsänderungen, der AR legt in Richtung von  $\vec{g}$  die Strecke  $s$  zurück. Damit ist die Verschiebung  $\Delta\vec{r}$  des AMP

$$\Delta\vec{r} = s\vec{g} \quad (6)$$

Bei ungleicher Drehgeschwindigkeit der Räder gibt es eine Richtungsänderung  $\Delta\alpha$ . Diese kann elementar bestimmt werden, indem man sich etwa mit dem linken Radmittelpunkt mitbewegt denkt. Man beobachtet dann eine Achsdrehung um den (vorzeichenbehafteten) Winkel<sup>3</sup>

$$\Delta\alpha = -\frac{s_l - s_r}{2R} \quad (7)$$

<sup>1</sup> $\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2$  ist das Skalarprodukt der beiden Vektoren.

<sup>2</sup>Die Strecken sind vorzeichenbehaftet, Rückwärtsfahrt eines Rades heißt negative Strecke.

<sup>3</sup>Wir messen die Winkel im Bogenmaß, d.h.  $\alpha = 2\pi$  ist der Vollkreis. Ist z. B.  $s_r = 0$  und  $s_l = 4\pi R$  so hat das linke Rad einen Vollkreis beschrieben, so daß  $\Delta\alpha = 2\pi$  in Übereinstimmung mit 7.

Formel 7 gilt solange die Geschwindigkeit der Räder gleich (und die Drehung weniger als ein Vollkreis) ist. Zur Bestimmung der Verschiebung des AMP im allgemeinen Fall betrachten wir wieder einen Zeitraum, in dem wir die Geschwindigkeiten der Räder als konstant annehmen können. Der AMP des AR beschreibt dann ein Kreissegment mit Radius

$$\rho = \frac{s}{\Delta\alpha}, \quad s = \frac{s_l + s_r}{2} \quad (8)$$

wobei  $s$  die vom AMP zurückgelegte Strecke ist. Der Radius ist vorzeichenbehaftet, d. h. er ist negativ, falls der Kreismittelpunkt vom AMP aus gesehen in der zu  $\vec{a}$  entgegengesetzten Richtung liegt. Die Verschiebung des AMP ist damit

$$\Delta\vec{r} = R \frac{s_l + s_r}{s_l - s_r} \left[ \begin{pmatrix} \cos(\alpha + \Delta\alpha) \\ \sin(\alpha + \Delta\alpha) \end{pmatrix} - \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \right] \quad (9)$$

Die Formel 7 bestimmt die Änderung der Position  $\vec{r}$  und der Orientierung  $\alpha$  des AR vollständig. Sie ist anwendbar solange nur die Geschwindigkeiten der Räder konstant sind und die tatsächliche Winkeländerung kleiner als  $2\pi$  bleibt. Oft ist es günstiger mit dem Winkel  $\phi = \alpha - \pi/2$  zu arbeiten, den der Vektor  $\vec{g}$  (Blickrichtung des Roboters) mit der  $x$ -Achse einschließt. Gehen wir zur Komponentenschreibweise über, so gilt jetzt anstelle von (9)

$$\begin{aligned} \Delta x &= -R \frac{s_l + s_r}{s_l - s_r} (\sin(\phi + \Delta\phi) - \sin\phi) \\ \Delta y &= R \frac{s_l + s_r}{s_l - s_r} (\cos(\phi + \Delta\phi) - \cos\phi) \end{aligned} \quad (10)$$

Wenn sich die Geschwindigkeiten der Räder kontinuierlich ändern, dann gilt mit  $\Delta\phi = \Delta\alpha$  folgende Differentialgleichung für die Komponenten  $x$  und  $y$  des Ortsvektors  $\vec{r}$  und des Winkels  $\phi$

$$\begin{aligned} \dot{x} &= \frac{v_l + v_r}{2} \cos\phi \\ \dot{y} &= -\frac{v_l + v_r}{2} \sin\phi \\ \dot{\phi} &= -\frac{(v_l - v_r)}{2R} \end{aligned} \quad (11)$$

wobei die Radgeschwindigkeiten durch den Controller vorgegeben zu denken sind. Die Differentialgleichung kann elementar oder durch den Grenzübergang  $\Delta\phi \rightarrow 0$  aus (10) gewonnen werden.

### Odometriefehler:

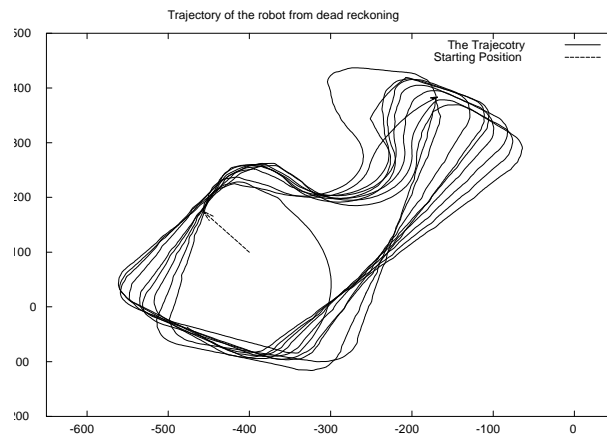


Figure 1: Odometriefehler. Im Laufe von 10 Runden, die ein Khepera-Roboter in einem einfachen Labyrinth zurücklegt, ergibt sich durch die (systematischen) Odometriefehler eine scheinbare Winkeldrift von mehr als  $45^\circ$ . Der Fehler folgt wahrscheinlich aus den nicht genau bekannten Raddurchmessern, die für die Odometrie als gleich angenommen wurden.

- Die Größen  $s_l$  und  $s_r$  sind die tatsächlich zurückgelegten Strecken. Diese weichen von den aus den Radumdrehungen bestimmten theoretischen Strecken in der Praxis durch das Rutschen der Räder (Schlupf) oft erheblich ab. Die odometrischen Daten müssen folglich immer mit – aus der Position von Landmarken, Hinernissen o. ä. ermittelten – absoluten Stellungsdaten abgeglichen werden.
- Der Rutscheffekt macht leider die bei vielen Robotern recht genaue Ansteuerung der Räder teilweise wieder zunichte. Beispiel: Der Khepera Miniaturroboter realisiert als minimalen Schritt Strecken  $s_l, s_r = 0,08$  mm, die tatsächlich zurückgelegten Strecken weichen davon um bis zu  $\pm 10\%$  ab, vgl. Abb. ??.

### Bemerkung:

Nachteil von Formel (9) bzw. (10) ist das Auftreten von  $\Delta\alpha$  im Nenner (vgl. (7)), das ja beliebig klein und bei gleicher Geschwindigkeit der Räder auch Null werden kann. Abhilfe

schaft für  $\Delta\alpha \ll 1$  die für alle  $\alpha$  schnell konvergierende Taylorentwicklung

$$\cos(\alpha + \Delta\alpha) = \cos\alpha - \Delta\alpha \sin\alpha - \frac{(\Delta\alpha)^2}{2} \cos\alpha + \dots$$

$$\sin(\alpha + \Delta\alpha) = \sin\alpha + \Delta\alpha \cos\alpha - \frac{(\Delta\alpha)^2}{2} \sin\alpha + \dots$$

Abbruch nach dem zweiten Glied und Einsetzen in 9 liefert

$$\Delta\vec{r} = s \left[ \begin{pmatrix} \sin\alpha \\ -\cos\alpha \end{pmatrix} + \frac{\Delta\alpha}{2} \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \right] = s\vec{g} + s\frac{\Delta\alpha}{2}\vec{a}, \quad \Delta\alpha \ll 1 \quad (12)$$

oder

$$\Delta\vec{r} = \frac{s_l + s_r}{2}\vec{g} - \frac{s_l^2 - s_r^2}{4R}\vec{a}, \quad \frac{s_l - s_r}{2R} \ll 1 \quad (13)$$

Der erste Term beschreibt offensichtlich die geradlinige, tangential gerichtete Verschiebung längs  $\vec{g}$  um die Strecke  $s$ , der zweite Term die achsial gerichtete Verschiebung um die Strecke  $\pi \frac{s_l^2 - s_r^2}{4R}$ . Dieser Korrekturterm schiebt den AMP zurück auf den Kreisbogen. Die Formel 12 bzw. 13 ist für alle endlichen  $\Delta\alpha$  anwendbar, allerdings nur solange  $\Delta\alpha \ll 1$ .

## 2.2 Merkmalsextraktion aus Entfernungsdaten

Für die Messung der Abstände des AR zu den umgebenden Objekten stehen eine Reihe von Verfahren wie Infrarot- und , Ultraschallsensoren, Laserscanner usw. zur Verfügung. In der Sensorkomponente wird gemeinhin schon eine elementare geometrische Identifikation der Objektwelt vorgenommen, standardmäßig etwa die Identifikation von objektbegrenzenden Geraden Diese Verfahren setzen den aktiven Einsatz der Sensoren voraus, was ihre Integration in die Sensorkomponente begründet.

## 2.3 Abstandsmessung mit Ultraschall (Sonar)

Die Entfernung wird ähnlich wie beim Laserscanner (s. u.) aus der Laufzeit des reflektierten Schalls ermittelt. Vorteil dieser Sensoren ist ihr geringer Preis, ihre Robustheit und einfache Realisierung. Nachteile sind (1) die geringe Ortsauflösung wegen der relativ großen Keulenbreite von 10 – 30 Grad und die relativ stark ausgeprägten Nebenmaxima in der Strahlungskeule und (2) die geringe Abtastrate. Diese ergibt sich aus der Schallgeschwindigkeit von ca. 300 m/s, d. h. Laufzeit für 1 m ist ca. 3,3 ms. Täuschungen durch Mehrfachreflexionen.

Ultraschallsensoren werden aufgrund dieser Eigenschaften meist als "flächendeckender" Sensor für die Kollisionsvermeidung eingesetzt. Die Keulenbreite läßt sich außerdem elektronisch in gewissem Umfang steuern. Wichtig für Kurvenfahrten.

## 2.4 Abstandsmessung mit Laserscanner

Die Abstandsmessung kann sowohl durch (1) Messung der Flugzeit, (2) der Phasenverschiebung oder (3) durch Lasertriangulation erfolgen. Die Flugzeitmessung erfolgt wie beim Ultraschallradar nach folgendem Schema:

**Abb.:** Messung des Abstandes nach der Laufzeitmethode.

Die Messung der Laufzeit wird durch einen Zeit—Amplitudenkonverter Konv. realisiert, deren Auflösung gegenwärtig bei  $\leq 10$  Picosekunden (1 Picosekunde =  $10^{-12}s$ ) liegt. Zum Vergleich: Das Licht legt in 3,3 Picosekunden ca. 1 mm zurück. Der Pulsgenerator (Pulser) startet über den Trigger den Konverter und den Laser, der einen Strahl auf das Objekt emittiert. Der Oberflächen—Sperrschichtdetektor detektiert das ankommende reflektierte Signal und gibt es praktisch ohne Zeitverzögerung an den Verstärker weiter. (An diesem kann zusätzlich noch ein Intensitätssignal abgegriffen werden) Über einen Diskriminator und eine Verzögerung erreicht es den Konverter und stoppt diesen. Der Konverter liefert ein zur Zeitdifferenz proportionales Analogsignal, das über einen AD—Wandler direkt als Abstand ausgegeben wird.

Beim Phasenverfahren dient statt der Laufzeit die durch Interferenz bestimmte Phasenverschiebung zwischen dem reflektierten und dem direkten Strahl des Lasers als Maß für die durchlaufene Strecke. Das Laser—Triangulationsverfahren registriert den reflektierten Laserstrahl in einer Zeile bzw. in einem Feld von Fotoelementen (Linearkamera bzw. Fotoarray, Matrixbildsensoren).

## 2.5 Merkmalsextraktion aus Abstandsdaten: Identifikation von Geraden

Für einen Roboter, der sich in einer komplexen Umgebung bewegt, ist eines der primären Merkmale die geometrische Grundstruktur der ihn umgebenden Hindernisse bzw. Wände. Die den aktuellen Horizont des AR begrenzenden Geraden können auf einfache Weise mit einem Winkelhistogramm identifiziert werden. Dazu wird der Abstand zum nächsten Objekt unter zwei benachbarten Winkeln  $\alpha, \alpha'$  relativ zu einer Bezugsachse vermessen. Unter der Hypothese, daß die zwei Aufpunkte auf einer Geraden liegen, kann der Winkel  $\gamma$  dieser Geraden mit der Bezugsachse leicht bestimmt werden. Die Häufigkeit der Ergebniswinkel  $\gamma$  dieser vielfach unter immer neuen Winkelkombinationen



---

$\alpha, \alpha'$  vorgenommenen Vermessung liefern, über  $\gamma$  abgetragen, ein Histogramm. Die Winkel, die zu real vorhandenen Geraden gehören, zeichnen sich durch eine vermehrte Häufigkeit aus und sind so im Histogramm leicht auszumachen und somit automatisch zu detektieren.

### 3 Mustererkennung und Klassifikation zur Datenreduktion

Die Klassifikation der Sensordaten nach vorgegebenen Mustern ist ein wichtiges Hilfsmittel zur Extraktion der relevanten Information aus den Sensordaten. Die sensorischen Muster können z. B. aus prototypischen Stellungen des AR relativ zu bestimmten Hindernissen gewonnen werden. Jede dieser  $i = 1, \dots, P$  ausgewählten Stellungen erzeugt einen sensorischen Output  $\vec{x} = \vec{w}_i$ , der als sensorischer Prototyp oder Repräsentant des Musters  $i$  in der Wissensbasis abgelegt wird. Diese Zuordnung ist eine Abbildung  $\mathcal{M}$  der abstrakten Muster  $i = 1, \dots, P$  in den Raum der Sensordaten  $\mathcal{X}$ . Die Abbildung  $\mathcal{M}$  ordnet also jedem der  $P$  Muster einen Repräsentanten oder Codebuchvektor (CV)  $\vec{w}_i \in \mathcal{X}$ ,  $i = 1, \dots, P$  zu, der das Muster in  $\mathcal{X}$  repräsentiert.

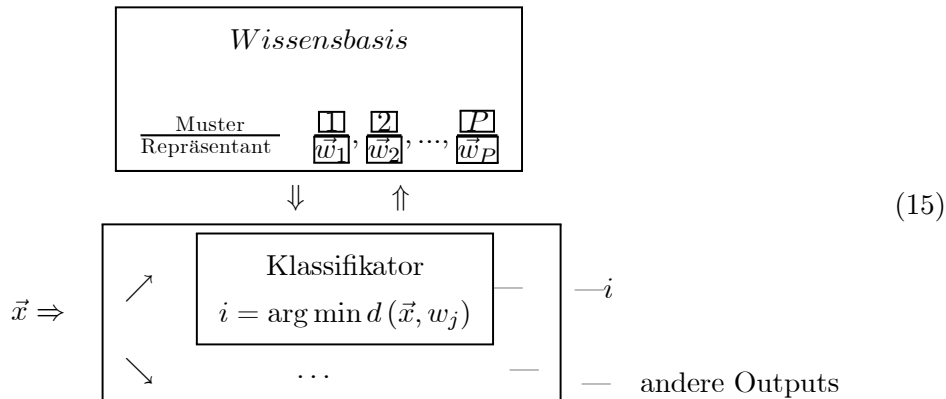
**Beispiel:** Wir wollen drei typische Stellungen eines Khepera Roboters vor einem Hindernis (Wand) klassifizieren. Für jede dieser drei prototypischen Stellungen  $k = 1, \dots, 3$  seien die Outputs  $x_i^{(k)}$ ,  $i = 1, \dots, 8$  der acht Infrarotsensoren ermittelt. Die drei achtdimensionalen Vektoren  $\vec{w}_k = (x_1^{(k)}, x_2^{(k)}, \dots, x_8^{(k)})$  sind die gesuchten Repräsentanten der drei Muster. Gewinnung sensorischer Prototypen

Die Mustererkennung reduziert sich dann auf die Aufgabe, zu einem gegebenen sensorischen Input in das Verarbeitungssystem  $\vec{x}$  das zugehörige Muster  $i$  zu finden. Die Mustererkennung entspricht mathematisch also einer Funktion

$$i = m(\vec{x}) \tag{14}$$

die jedem  $\vec{x}$  genau ein  $i$  zuordnet. Das erfolgt durch einfachen Vergleich mit

den in der Wissensbasis gespeicherten Repräsentanten:



Die Übereinstimmung (*match*) des durch  $\vec{w}_i$  repräsentierten Musters  $i$  mit  $\vec{x}$  sei durch die Abstandsfunktion  $d(\vec{x}, \vec{w}_i)$  quantifizierbar. Ist  $i$  das Muster mit der besten Übereinstimmung, dann gilt für dieses

$$d(\vec{x}, \vec{w}_i) \leq d(\vec{x}, \vec{w}_j) \quad \forall j \tag{16}$$

Somit ist

$$m(\vec{x}) = \arg \min_j d(\vec{x}, \vec{w}_j) \tag{17}$$

**Bemerkung:** Ist speziell die Abstandsfunktion durch die Euklidische Metrik gegeben,

$$d(\vec{x}, \vec{w}) = \sqrt{\sum_{i=1}^n (x_i - w_i)^2} \tag{18}$$

Voronoi Par-  
kettierung des  
Sensordaten-  
raumes

dann definiert 16 die Voronoi-Parzellierung (oder -Parzellierung) des Sensordatenraumes  $\mathcal{X}$ . Jedes Muster  $i$  belegt eine Zelle  $\mathcal{U}_i$  in  $\mathcal{X}$ , die durch die Menge der  $\vec{x}$  gegeben ist, die auf  $i$  abgebildet werden, d. h. die dem Repräsentanten  $\vec{w}_i$  des Musters am nächsten liegen

$$\mathcal{U}_i = \{\vec{x} \mid d(\vec{x}, \vec{w}_i) \leq d(\vec{x}, \vec{w}_j) \quad \forall j\} \tag{19}$$

bzw.

$$\mathcal{U}_i = \{\vec{x} \mid i = m(\vec{x})\} \tag{20}$$

und es gilt

$$\mathcal{X} = \cup_{i=1}^n \mathcal{U}_i \tag{21}$$

d. h. daß die Zellenzerlegung vollständig ist.

Die Hinzunahme von Kontextinformationen kann (1) die Zahl der aktuellen Vergleichsmuster reduzieren (bestimmte Muster können unter gewissen Umständen nicht auftreten) und damit die Klassifikationssicherheit erhöhen und

---

(2) neue Dimensionen in's Spiel bringen wodurch die Abstandsbestimmung diskriminativer werden kann. Wir fassen die Sensorinformation  $\vec{x}$  und die Kontextinformation (die allgemein eine Funktion von  $\vec{x}$  ist)  $\vec{k} = (k_1, \dots, k_m)$  zu einem Vektor zusammen

$$\vec{X} = (\vec{x}, \vec{k})$$

und entsprechend

$$\vec{W}_i = (\vec{w}_i, \vec{k}_i)$$

wobei  $\vec{k}_i$  die prototypische Kontextinformation zu Muster  $i$  ist. Dann erlaubt die neue Abstandsfunktion  $d(\vec{X}, \vec{W}_i)$  i. a. eine genauere Zuordnung. Im obigen Beispiel der Stellungen eines Khepera Roboters vor einem Hindernis wären Kontextinformationen beispielweise Reflexionseigenschaften oder Gestalt (Wandkrümmung) bzw. Objekttyp des Hindernisses.

**Beispiel:** So kann für die Sensordaten  $\vec{x}, \vec{x}'$  zweier verschiedener Stellungen  $d(\vec{x}, \vec{w}_i) \neq d(\vec{x}', \vec{w}_i)$  gelten (Zuordnung zu verschiedenen Mustern), unter Berücksichtigung der Kontextinformation werden diese aber als gleich erkannt, d.h.  $d(\vec{X}, \vec{W}_i) = d(\vec{X}', \vec{W}_i)$ . Das könnte z. B. bei gleichen Stellungen vor Wänden mit unterschiedlichen Reflexionseigenschaften der Fall sein.

## 4 Optische Sensoren

Optische Sensoren (Bildwandler) in Verbindung mit Bildverarbeitungssystemen stellen die wichtigste Grundlage für die intelligente Reaktion eines Roboters auf die Umgebung dar. Sie involvieren aber damit auch die gesamte Komplexität, die sich mit dem maschinellen Sehen in der KI verbindet. Typische Aufgabenstellungen in der Robotik sind:

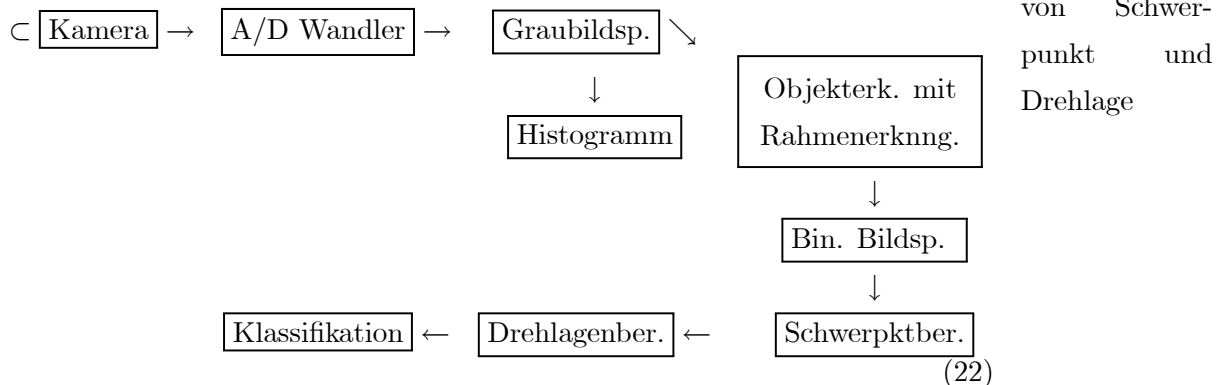
1. Erkennen und Klassifizieren von Kollisionshindernissen.
2. Bestimmung von Lage und Orientierung von Objekten relativ zum AR.
3. Objekterkennung allgemein.
4. Analyse von Szenen und Bildfolgen.
5. Interpretation bewegter Szenen.

Die wichtigsten Bildwandlersysteme lassen sich nach Dillmann [Dillmann, 1991] in folgende vier Klassen gliedern:

- 
1. Binäre fotoelektrische Zellen (Fotoelemente): Output je nach Helligkeit des auffallenden Lichtes gleich 0 oder 1.
  2. Linearkamera: Reihe von Fotozellen in linearer Anordnung. Meist in CCD-Technik (charge-coupled-devices = lichtempfindliche CMOS Kondensatoren). Diese bauen eine elektrische Ladung proportional zur Stärke des einfallenden Lichtes auf. Serielles Auslesen über analoge Schieberegister.
  3. Matrixkamera (Array von Fotoelementen): Bis zu  $1024 \times 1024$  Fotoelemente in zweidimensionaler Anordnung.
  4. CCD-Fernsehkameras: Zusätzlich Erfassung der Grauwerte. Rel. geringer Preis bei hoher Auflösung.

Für die sichere Erkennung eines Objektes ist *a priori* Wissen erforderlich, z. B. Kenntnisse über die Geometrie, Oberflächen, Kanten, Ecken, ... der in Frage kommenden Objekte. Dieses Wissen kann in einer vorgeschalteten Lernphase anhand von Beispielen erworben werden. Mit Kenntnissen über Licht, Auflösung, Rauschpegel und die abgespeicherten Merkmalen der Objekte kann nach Ausführung der erforderlichen Faltungsoperationen, Korrelationsberechnungen und Vergleich mit Referenzobjekten die Objektklassifikation mit hoher Sicherheit vorgenommen werden.

Das folgende binäre optische Bildererkennungssystem realisiert die Klassifikation eines planaren Objektes und extrahiert den Schwerpunkt und die Drehlage aus den mit einer CCD-Kamera aufgenommenen Bilddaten:



Das Histogramm speichert die Häufigkeit  $h(g)$  des Auftretens eines bestimmten Grauwertes  $g$ . Es dient der allgemeinen statistischen Charakterisierung des Bildes und stellt Hintergrundwissen für die Objektidentifikation

---

oder die Objekt-Hintergrund-Trennung zur Verfügung. Mathematisch ist es das gewichtete Mittel der Kroneckerfunktion  $\delta(a, b) = 1$  für  $a = b$  und  $\delta(a, b) = 0$  sonst:

$$h(g) = \frac{1}{N} \sum_{(i,j)} \delta(f(i, j), g) \quad (23)$$

Histogramm

wobei  $f(i, j)$  der Grauwert des Pixels  $(i, j)$  und  $N = \sum_{(i,j)} 1$  die Zahl der Pixel ist.

Berechnung des Flächenschwerpunktes: Sei  $\chi_{ij}$  die Indikatorfunktion mit

$$\chi_{ij} = \begin{cases} 1 & \text{falls Pixel } (i, j) \in \text{Objekt} \\ 0 & \text{sonst} \end{cases} \quad (24)$$

Die Koordinaten des Schwerpunktes sind dann

$$i_S = \frac{1}{F} \sum_{i,j} i \chi_{ij} \quad (25)$$

$$j_S = \frac{1}{F} \sum_{i,j} j \chi_{ij}$$

wobei  $F = \sum_{i,j} \chi_{ij}$  die Fläche des Objektes ist. Die Bestimmung der Drehlage des Objektes erfolgt nach der Methode der polaren Kodierung: Lege Kreis  $K_0$  mit Radius  $R$  um den Schwerpunkt des Objektes. Lege  $m$  weitere Kreise  $K_k$ ,  $k = 1, \dots, m$  mit Radius  $R$  deren Mittelpunkte gleichmäßig auf Kreis  $K_0$  verteilt liegen. Bilde für alle Kreise  $K_k$

$$C(k) = \sum_{(i,j) \in K_k} \chi_{ji} \quad (26)$$

Das zählt die Objektpixel, die in Kreis  $K_k$  liegen. Zur Bestimmung der gesuchten Phasenverschiebung (Objektdrehung) dient die Referenzfunktion  $C_R(k)$  für dieses Objekt, die mittels (26) mit dem Objekt in definierter Lage ermittelt wurde. Ausgabe ist

$$k_s = \arg \max_{k_s} \left( \sum_{k=1}^m (C(k) - C_R(k - k_s))^2 \right) \quad (27)$$

wobei  $C_R(m + \kappa) = C_R(\kappa)$ . Das ist der Index des Kreises, in dem der Bezugspunkt des gedrehten Objektes zu suchen ist.

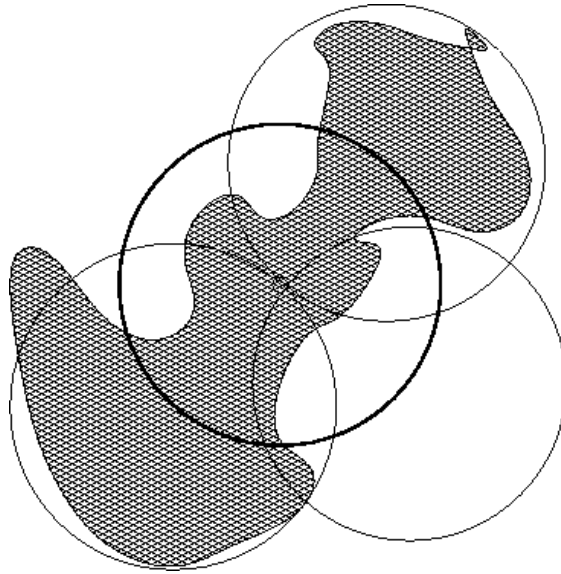


Figure 2: Die Methode der polaren Kodierung. Eingezeichnet ist der zentrale Kreis und drei der Kreise deren Überlappungen mit dem Objekt die Komponenten des Vektors  $C$  bestimmen.

## 5 Filterungen

Filter dienen der gezielten Hervorhebung erwünschter bzw. der Unterdrückung unerwünschter Komponenten in einem Signal. Prototypen eines Filters sind elektrische Schwingkreise, die durch das ankommende Signal in ihrer Eigenfrequenz erregt werden. Das gefilterte Signal enthält dann wesentlich nur diese Frequenz. Gewöhnlich sind Filterungen Operationen im Frequenzbereich signalverarbeitender Systeme, die dann in eine Integraloperation (Faltung) auf der Funktion selbst rückübersetzt werden.

Allgemein können wir die Filterung mit Übertragungsfunktion  $K$  des Filters

$$f \quad \boxed{\text{Filter } K} \quad \tilde{f}$$

einer einstelligen Funktion  $f(x)$  durch eine solche Faltungsoperation mit einem Kern  $K(x')$  definieren

Definition der  
Filterung

$$\tilde{f}(x) = \int dx' K(x') f(x - x') \quad (28)$$

Für eine zweistellige Funktion  $f(x, y)$  wird die Filterung mit einem Kern  $K(x', y')$  durch

$$\tilde{f}(x, y) = \int dx' dy' K(x', y') f(x - x', y - y') \quad (29)$$

geleistet<sup>4</sup>. Die diskrete Variante von 29 braucht man etwa für die Filterung eines Matrixbildes mit Pixelwert (Grauwert)  $f(i, j)$  für Pixel  $(i, j)$ . Die gefilterte Funktion ist

$$\tilde{f}(i, j) = \sum_{i', j'} K(i', j') f(i - i', j - j') \quad (30)$$

Die diskrete Filterung 30 findet vor allem in der Vorverarbeitung von Bild-daten Verwendung. Die Matrix  $K(i', j')$  wird hier als eine **Faltungsmaske** interpretiert, die mit ihrem Zentrum auf das betrachtete Pixel  $(i, j)$  aufsetzt und deren Matrixelemente außerhalb einer Umgebung  $\mathbf{U}$  um das Zentrum<sup>5</sup> gleich Null sind,  $K(i', j') = 0 \quad \forall (i', j') \notin \mathbf{U}$ . Die Faltung 30 ist damit eine Nachbarschaftsoperation

$$\tilde{f}(i, j) = \sum_{(i', j') \in \mathbf{U}} K(i', j') f(i - i', j - j') \quad (31)$$

bei der sich der neue Pixelwert  $\tilde{f}(i, j)$  durch die gewichtet Aufsummation der Pixelwerte  $f(i - i', j - j')$  aus der Nachbarschaft von  $(i, j)$  (die den Pixel selbst mit einschließt) ergibt. Die Maske wird sukzessive über alle Pixel des Bildes geschoben, im Ergebnis erhält man das gefilterte Bild.

Die wichtigsten Beispiele für die Anwendung von Filterungen sind:

1. Gleitende Mittel zur Rauschunterdrückung: Besonders bei Sensordaten in Form von Zeitreihen, d.h. der Output eines Sensors sei  $x(t)$ ,  $t = 0, 1, 2, \dots$  bieten sich **gewichtete gleitende Mittel** (GGM) über die Vergangenheit an. Die Verarbeitungseinheit sieht dabei zum aktuellen Zeitpunkt  $t$  den ausgegebenen Sensorwert  $x(t)$  und gibt unmittelbar das zugehörige gewichtete Mittel  $\tilde{x}$  aus —

$$x(t) \text{ — } \boxed{\text{GGM}} \text{ — } \tilde{x} \quad (32)$$

wobei

$$\tilde{x} = (1 - \gamma) \sum_{k=0}^t \gamma^k x(t - k) \quad (33)$$

und  $0 < \gamma < 1$  ein Parameter ist, der den Zeithorizont (s.u.) bestimmt. Der Ansatz 33 entspricht der Wahl  $K(k) = (1 - \gamma) \gamma^k$  in der eindimensionalen Version von 31. Der Vorfaktor ist so gewählt, daß für hinreichend

<sup>4</sup>Wir betrachten hier nur homogene (verschiebungsinvariante) Filter, der allgemeine Fall entspricht einer Transformation  $\tilde{f}(x, y) = \int dx' dy' K(x, x'; y, y') f(x', y')$ .

<sup>5</sup>Die Umgebung schließt das Zentrum selbst mit ein.

große  $t$  und konstantes  $x(t) = x$  sich gerade  $\tilde{x} = x$  ergibt (Beweis mit geometrischer Reihe). Die Glättung 33 entspricht einer gewichteten Aufsummation über die Werte der Vergangenheit, wobei die Gewichte  $\gamma^k$  der zeitlich weiter zurückliegenden Meßwerte  $x(t - k)$  exponentiell abnehmen.

Bei der Realisierung 32 von 33 müssen die vergangenen Werte im Speicher gehalten werden. Der folgende Update-Algorithmus erlaubt, die geglättete Größe *on-line* zu aktualisieren. Dabei wird nur  $\tilde{x}$  intern in der Verarbeitungseinheit GGM gehalten und für jeden neu einkommenden Sensorwert  $x(t)$  mit

$$\Delta\tilde{x}(t) = \frac{1}{T} (x(t) - \tilde{x}(t)) \quad (34)$$

inkrementiert<sup>6</sup>, wobei  $T = \frac{\gamma}{1-\gamma}$ . (Zur Kontrolle: Für konstantes  $x(t) = x$  kommt der Algorithmus zum Stehen, sobald  $\tilde{x}(t) = x$ ). Der frei wählbare Parameter  $T$  ist hier unmittelbarer als Zeitkonstante bzw. als Ausdehnung des Zeithorizontes erkenntlich (Für  $T \rightarrow \infty$  ist 34 äquivalent zur Differentialgleichung  $d\tilde{x}/dt = \frac{1}{T} (x(t) - \tilde{x}(t))$ ). Die explizite Lösung z. B. für  $x = const$  ist  $\tilde{x}(t) = x + e^{-t/T} (\tilde{x}(0) - x)$ .

2. Reduktion des Rauschens, z. B. durch lokale Glättungsoperationen über Matrixbildern: Das entspricht einer Tiefpaßfilterung – im Frequenzbild Tiefpaß ergibt sich die Glättung durch Unterdrückung höherer Frequenzanteile.

Bei einer einfachen  $3 \times 3$  Umgebung  $\mathbf{U}$  sind die wichtigsten Masken

$$K = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \text{ die mittlere Maske}$$

$$K = \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \text{ und } K = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}. \text{ Die einfache Mit-}$$

telung beseitigt unerwünschte schnelle Änderungen der Bildintensität und fungiert damit als Rauschverringerng. Im Ergebnis erzeugt sie leicht verschwommene Bilder, glättet aber auch die Kanten. Diesen Effekt vermei-

<sup>6</sup>Zum Beweis bedenken wir, daß  $\tilde{x}(t)$  der **neue** Wert der gemittelten Größe ist (d.h. mit Berücksichtigung von  $x(t)$ ), das Inkrement ist also als  $\Delta\tilde{x}(t) = \tilde{x}(t) - \tilde{x}(t-1)$  zu verstehen. Dann folgt mit  $\tilde{x}(t-1) = (1-\gamma) \sum_{k=0}^{t-1} \gamma^k x(t-1-k)$   
 $= (1-\gamma) \sum_{k=1}^t \gamma^{k-1} x(t-k) = \frac{1}{\gamma} \tilde{x}(t) - \frac{1-\gamma}{\gamma} x(t)$  leicht die Update-Regel 34.



---

det der *Medianfilter*, der das mittlere Bildelement durch den Medianwert<sup>7</sup> der Pixel aus der Umgebungsmenge ersetzt.

Hochpaß

3. Unterstützung der Merkmalsextraktion durch z. B. Kontrastverstärkung, insbesondere Kantenverstärkung: Das entspricht einem Hoch- oder Bandpaßfilter bzw. mathematisch einer Differentiation. Typisches Beispiel einer Maske für die Hochpaßfilterung ist die Maske  $K_H = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$ , diese führt im wesentlichen eine Differentiation aus (s.u.), erhält aber die Information über die Pixelwerte. Diese Maske läßt somit Gebiete gleicher Pixelwerte unbeeinflusst, hebt aber die Übergänge zwischen solchen Gebieten hervor.

4. Merkmalsdetektoren, speziell Kantendetektoren: Der o. g. Hochpaßfilter **verstärkt** vorhandene Kanten in einem Pixelbild. Es lassen sich auch Masken angeben die als allgemeine Merkmalsdetektoren fungieren, d. h. die nur bei Vorhandensein dieses Merkmals einen von Null verschiedenen Output liefern. Die erste und zweite Ableitung des Bildes in Richtung der  $x$ - Achse oder der  $y$ - Achse kann z. B. zur Detektion von Kanten dienen. Die Ableitung ist in Gebieten homogener Pixelwerte gleich Null und nur merklich ungleich Null in Gebieten rascher Veränderung der Pixelwerte. Man fährt zur Bildung der Ableitung entweder mit einer eindimensionalen Maske die Zeilen und Spalten des Pixelbildes ab, die Maske für die zweite Ableitung in  $x$ -Richtung ist im einfachsten Fall

$$K = \begin{pmatrix} -1 & 2 & -1 \end{pmatrix} \quad (35)$$

oder verwendet eine der diskreten Darstellungen des Laplaceschen Differentialoperators  $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ , realisiert z. B. durch die Maske

$$K_{\Delta} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}. \text{ Wie man unmittelbar sieht, gibt der Filter in}$$

Regionen gleicher Pixelwerte in der Tat Null zurück. Der in 3 aufgeführte

---

<sup>7</sup>Der Medianwert  $x_M$  einer Menge  $S$  von Werten ist so gewählt, daß die Hälfte der Werte aus  $S$  größer und die andere Hälfte kleiner als  $x_M$  ist. Beachte, daß die Zahl der Werte in einer Pixelumgebung i. a. ungeradzahlig ist.

---

Hochpaßfilter  $K_H$  ist die Summe  $K_H = K_\Delta + I$ , wobei  $I = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

den Pixelwert identisch erhält.

5. Kantenverfolgung: Die Kantendetektoren liefern nur Informationen über das lokale Vorhandensein von Kanten. Das Verfolgen der Kanten kann ebenfalls mithilfe der Faltungsoperation 31 mit  $K$  als sog. Schablonenmaske erreicht werden. Die Bestimmung des Verlaufs einer diagonalen Kante mit Orientierung  $45^\circ$  beispielweise kann mit der Maske  $K = \begin{pmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{pmatrix}$  erfolgen. Kantenverfolgung

6. Elimination störender Information, z. B. Hintergrund in Szenen.

**Bemerkungen:**

- Die in den obigen Punkten 2 und 3 genannten Masken sind  $3 \times 3$  Matrizen, sie beziehen damit nur eine sehr kleine Umgebung des betrachteten Pixels ein. Eingesetzt werden auch  $5 \times 5$  oder  $9 \times 9$  Matrizen allerdings um den Preis eines wesentlich höheren Verarbeitungsaufwandes was für autonome Roboter schnell zum Verlust der Echtzeitfähigkeit führt.
- Die Methode 5 der Kantenverfolgung mit Schablonenmasken kann auch bei autonomen Robotern zur Identifikation der geometrischen Struktur eines Hindernisses aus Positionsdaten beim Umfahren desselben verwendet werden.

## 6 Sensorvalidierung und –verifikation

Besonders bei autonomen Robotern (AR) müssen (1) die Sensoren ständig in ihrer Funktionsfähigkeit überprüft und bei Ausfall eines Sensor geeignete Maßnahmen ergriffen werden = Validierung des Sensors. Dazu gehören Kalibrierungsoperationen bzw. Vergleich mit prototypischen Sensorcharakteristiken bzw. der Korrelationen mit anderen Sensoren. Außerdem besteht (2) immer die Gefahr, daß die Sensoren "getäuscht" werden. Beispiele sind Glastüren, die von Laserentfernungsmessern nicht wahrgenommen werden, Mehrfachreflexionen, die bei Ultraschallsensoren die Objekte in größerer Entfernung erscheinen

lassen usw. Wichtigste Gegenmaßnahme: Verifikation der im Ergebnis der Sensordatenverarbeitung erhaltenen Informationen wie Objektklassen, Lage und Entfernung von Objekten usw. durch

1. Verwendung von Kontextinformation aus der Wissensbasis
2. Sensorfusion, d. h. gemeinsame Verarbeitung der Informationen über den gleichen Gegenstand, die von verschiedenen Sensoren geliefert werden.  
**Beispiel:** Lösung des Glastürproblems durch Fusion Laserscanner mit Ultraschall—Entfernungsmesser.

Eine wichtige Voraussetzung für die Sensorfusion ist die Verwendung kompatibler Datenformate, was aber aufgrund der unterschiedlichen physikalischen Wirkprinzipien der Sensoren schwierig ist. Ein Ausweg ist die logische Sensorspezifikation. Dadurch kann die Spezifikation des Sensors von seiner physikalischen Implementierung unabhängig gestaltet werden. Ein Multisensorsystem wird dadurch portabel und deckt eine Vielzahl von Anwendungsfällen ab, vgl. [Dillmann, 1991].

## References

[Dillmann, 1991] R. Dillmann. *Informationsverarbeitung in der Robotik*. Springer, 1991.