

Kapitel 5

Umweltmodellierung

5.1.	Geometrische Karten	141
5.1.1.	Linienbasierte Karten	141
5.1.1.1.	Segmentierung	141
5.1.1.2.	Liniengenerierung	142
5.1.1.3.	Ausgleichsgerade	142
5.1.1.4.	Fusionieren von Linien	143
5.1.2.	Polygondarstellung	144
5.1.3.	Rasterkarten	145
5.1.3.1.	Tessellation	145
5.1.3.2.	Quadtree-Darstellung	146
5.1.3.3.	Aufnahme einer Rasterradarkarte	147
5.1.3.4.	Wahrscheinlichkeiten	148
5.1.3.5.	Fusion von Messdaten verschiedener Messsysteme	149
5.2.	Modellierung der Umwelt durch Szenen	149
5.2.1.	Verdichtung von Bildern	149
5.2.2.	Scenenklassifikation	150
5.2.2.1.	Vorverarbeitete Pixeldaten, nicht interpretiert	150
5.2.2.2.	Klassifikation auf den Rohdaten	151
5.2.3.	Vektorklassifizierer	151
5.2.4.	Objekterkennung aus Sensordaten	153
5.2.4.1.	Anwendung von Bilderkennungsalgorithmen	153
5.2.4.2.	Objekterkennung aus Radarkarten	153
5.2.5.	Landkarten	154
5.2.5.1.	Aufbau einer Karte (Edlinger und Weiß)	154
5.2.5.2.	Voraussetzungen	154
5.2.5.3.	Referenzpunkte	155
5.2.5.4.	Explorationsstrategie	156
5.3.	Topologische Karten	159
5.3.1.	Definition	159

5.3.2.	Umwandlung geometrische in topologische Karte	160
5.3.2.1.	Konvexe Polygone	161
5.3.2.2.	Zellgraph	162
5.3.2.3.	Erzeugung eines Dekompositionsbaumes	162
5.3.3.	ALICE: Erzeugung eines topologischen Graphen aus Rohdaten (U. Zimmer) ...	163
5.3.3.1.	Der AMR ALICE	164
5.3.3.2.	Konnektionistischer Ansatz	164
5.3.3.3.	Aufbau des Graphen aus Sensorsituationen	165
5.3.3.4.	Einschränkung des Suchraumes für die bmu	167
5.3.3.5.	Driftkorrektur	167
5.3.3.6.	Ergebnisse	168

5.1. Geometrische Karten

Modellierung der Umwelt durch Karten mit metrischen Angaben

- Begrenzungslinien von Hindernissen
- Polygone als Hindernisse / Freiräume
- Gebiete mit Befahrbarkeitseigenschaften

5.1.1. Linienbasierte Karten

Einteilung der Umwelt in Freiraum, Hindernis und/oder unbekannt

Ausgangspunkt: Messwerte entfernungsgebender Sensoren

Darstellung der Hindernisse als Punkthaufen (Cluster) oder durch Begrenzungslinien

5.1.1.1. Segmentierung

Vorr. gegeben ein Radarscan $\{ r_i, \varphi_i \}$ $i = 1, \dots, n$ von Position $Q = (x_0, y_0)$ aus.

Sei ein Radarpunkt P_i gegeben durch (r_i, φ_i)

Gesucht werden Cluster C_j und Segmente S_k und Hilfscluster H_m

Init: $C_1 := \{P_1\}$; $i := 1$; $j := 1$; $k := 1$; $m := 1$; $S_k := \{\emptyset\}$; $H_m := \{\emptyset\}$;

while $i < n - 1$ do

if $|P_{i+1} - P_i| < d$ $\implies H_m := H_m \cup P_{i+1}$; $i := i+1$;

else if $|P_{i+2} - P_i| < d$ $\implies H_m := H_m \cup P_{i+2}$; $C_j := \{P_{i+1}\}$; $j := j+1$; $i := i+1$;

else if $|P_{i+2} - P_{i+1}| < d$ $\implies m := m+1$; $H_m := \{P_{i+1}, P_{i+2}\}$; $i := i+2$;

else $C_j := \{P_{i+1}\}$; $j := j+1$; $m := m+1$; $H_m = \{P_{i+2}\}$; $i := i+2$;

endwhile

while $m > 0$ do

if $\#(H_m) \geq c \implies C_j := H_m$; $j := j+1$; $m := m-1$; $\{^* c \text{ ist max. Anz. in Cluster } ^*\}$

else $S_k := H_m$; $k := k+1$; $m := m-1$

endwhile

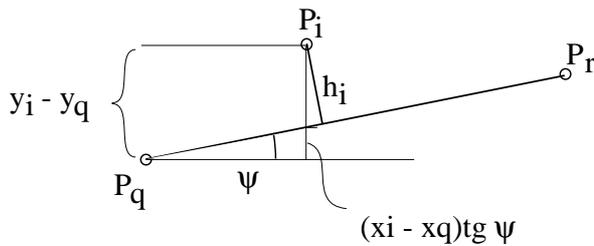
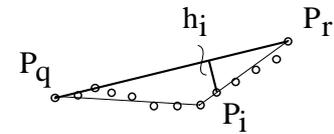
Es werden k Segmente und j Cluster gefunden.

5.1.1.2. Liniengenerierung

iterative endpoint fit (Duda)

betrachte Segment S_k mit Punkten P_q, \dots, P_r

bilde Abstände h_{r+1}, \dots, h_{q-1}



$$P_i = (x_i, y_i)$$

$$\text{tg } \psi = \frac{y_r - y_q}{x_r - x_q}$$

Betrachte Linie (P_q, P_r)

sei $h_j = \max_{r < i < q} (h_i) > \epsilon$ { * Unschärfe * }

==> bilde neue Linien (P_r, P_j) und (P_j, P_q)

sonst : Linie (P_q, P_r) etabliert

Aufwand: # Linien $\cdot (q-r)$

$$h_i = (y_i - y_q) \cos \psi - (x_i - x_q) \sin \psi$$

5.1.1.3. Ausgleichsgerade

Finde eine Gerade durch die Punkte P_q, \dots, P_r so, dass die Summe der Quadrate der Abstände minimal wird.

$$h_i = (y_i - y_q) \cos \psi - (x_i - x_q) \sin \psi \quad (\text{Gerade durch } P_q \text{ und } P_r)$$

$$h_i = (y_i - b) \cos \gamma - (x_i) \sin \gamma \quad (\text{Gerade durch die N Punkte und } (0, b))$$

$$\sum_i h_i^2 = \sum_i \{ (y_i - b)^2 \cos^2 \gamma - 2(y_i - b)(x_i) \sin \gamma \cos \gamma + (x_i)^2 \sin^2 \gamma \}$$

$$\frac{\partial (\Sigma(\dots))}{\partial b} = \sum_i \{ -2(y_i - b) \cos^2 \gamma + 2(x_i) \sin \gamma \cos \gamma \} \stackrel{!}{=} 0$$

$$0 = \sum_i \{ -(y_i - b) \cos \gamma + (x_i) \sin \gamma \} = \sum_i b - \sum_i y_i + \sum_i x_i \text{tg } \gamma$$

$$N b = \sum_i y_i - \text{tg } \gamma \sum_i x_i \implies y_s = \text{tg } \gamma x_s + b \quad (\text{Gerade durch den Schwerpunkt})$$

Die Gerade durch den Schwerpunkt (x_s, y_s) minimiert die Quadrate der Abstände.

$$\sum_{i=1}^N h_i^2 = \sum_{i=1}^N \{ (y_i - y_s)^2 \cos^2 \gamma - 2(y_i - y_s)(x_i - x_s) \sin \gamma \cos \gamma + (x_i - x_s)^2 \sin^2 \gamma \}$$

$$\frac{\partial \Sigma(\dots)}{\partial \gamma} = \sum_{i=1}^N \{ -2(y_i - y_s)^2 \cos \gamma \sin \gamma - 2(y_i - y_s)(x_i - x_s)(-\sin^2 \gamma + \cos^2 \gamma) + 2(x_i - x_s)^2 \sin \gamma \cos \gamma \} \stackrel{!}{=} 0$$

$$\sum_{i=1}^N \{ (y_i - y_s)^2 \operatorname{tg} \gamma - (y_i - y_s)(x_i - x_s) (1 - \operatorname{tg}^2 \gamma) + (x_i - x_s)^2 \operatorname{tg} \gamma \} = 0$$

$$\sum_{i=1}^N (y_i - y_s)(x_i - x_s) \operatorname{tg}^2 \gamma + \sum_{i=1}^N \{ -(y_i - y_s)^2 + (x_i - x_s)^2 \} \operatorname{tg} \gamma - \sum_{i=1}^N (y_i - y_s)(x_i - x_s) = 0$$

$$a \cdot z^2 + b \cdot z - a = 0$$

$$z_{1,2} = -\frac{b}{2a} \pm \sqrt{1 + \frac{b^2}{4a^2}}$$

$$\operatorname{tg} \gamma = z$$

$$a = \sum_{i=1}^N (y_i - y_s)(x_i - x_s)$$

$$b = \sum_{i=1}^N \{ -(y_i - y_s)^2 + (x_i - x_s)^2 \}$$

Verschieben der Linienendpunkte

Sei $\psi \approx \gamma$

$$h_s = (y_s - y_q) \cos \psi - (x_s - x_q) \sin \psi$$

$$\Sigma (h_i - h_s) = \Sigma \{ (y_i - y_s) \cos \psi - (x_i - x_s) \sin \psi \}$$

$$= \cos \psi (\Sigma y_i - N y_s) - \sin \psi (\Sigma x_i - N x_s) = 0$$

==> Verschiebung der Linienendpunkte um h_s :

$$P_q \rightarrow Q = (x_q + \Delta x, y_q + \Delta y) \quad \Delta x = h_s \sin \psi$$

$$P_r \rightarrow R = (x_r + \Delta x, y_r + \Delta y) \quad \Delta y = h_s \cos \psi$$

Durch die Punkte P und Q und den Schwerpunkt (x_s, y_s) läuft die gefundene Ausgleichsgerade.

5.1.1.4. Fusionieren von Linien

Zusammenfassen von Linien aus verschiedenen Scans und Verschmelzen zu einer Linie.
Damit erhalten die Linien sich gegenseitig stützend Gewicht.

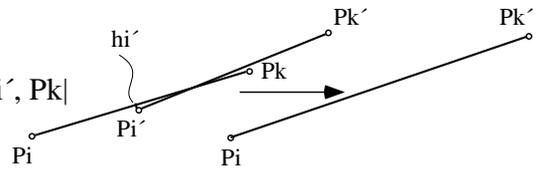
Seien R_1 und R_2 zwei Radaraufnahmen von verschiedenen Positionen P_0 und P_0' aus

sei (P_i, P_k) eine Linie aus R_1 und (P_i', P_k') eine Linie aus R_2 mit $\psi' = \psi + \epsilon$

a) sei $h_i' < \delta$ für P_i' in Bezug auf (P_i, P_k)

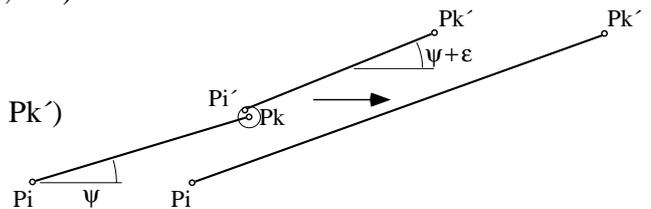
und P_i' zwischen P_i und P_k und $|P_i', P_k'| > |P_i', P_k|$

\implies Fusionieren zu einer Linie (P_i, P_k')



b) sei $|P_k, P_i'| < d$

\implies Fusionieren zu einer Linie (P_i, P_k')

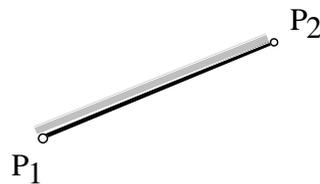


5.1.2. Polyondarstellung

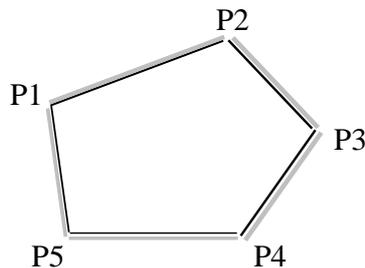
Die gefundenen Linien trennen Hindernis und Freiraum. Im Moment ihrer Aufnahme ist klar, wo Hindernisse sind. Diese Information wird in die Darstellung der Linien übernommen durch eine Konvention:

Orientierung der Linien

links von (P_1, P_2) ist Hindernis

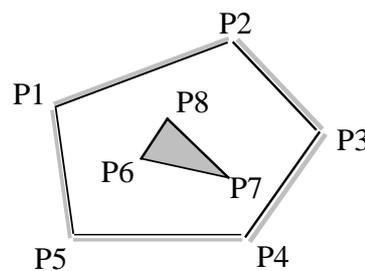


Beschreiben der Umgebung durch Polygone, die Freiraum kennzeichnen



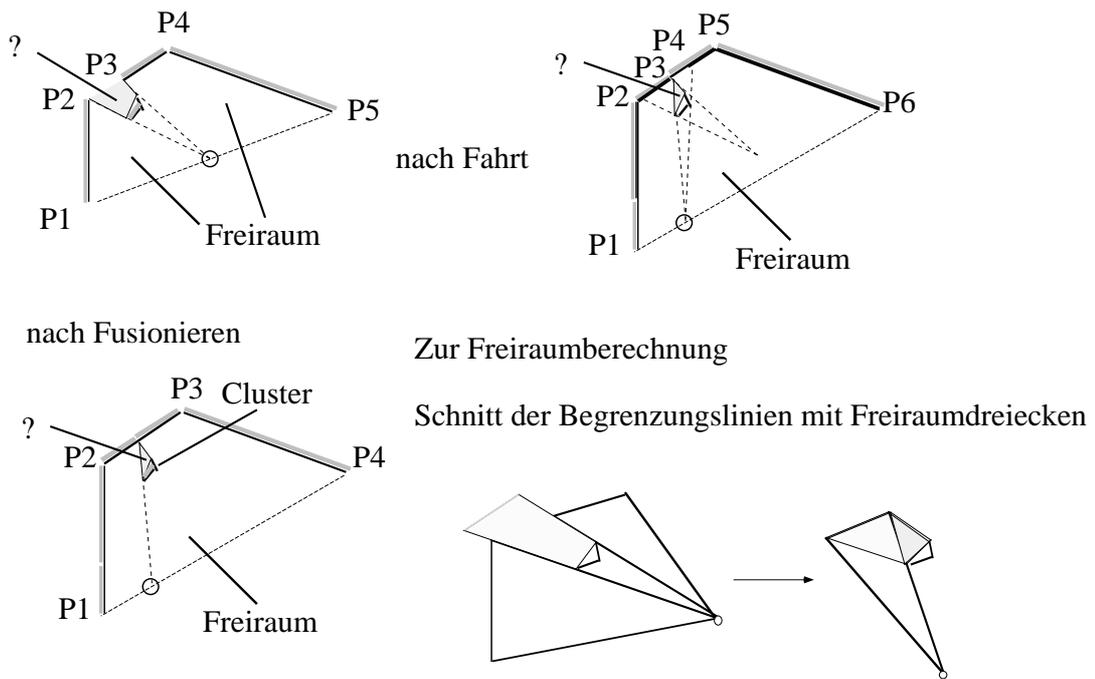
Polygon $(P_1, P_2, P_3, P_4, P_5)$

Polygone können isolierte Hindernisse sein:



Erfassen von Freiraum

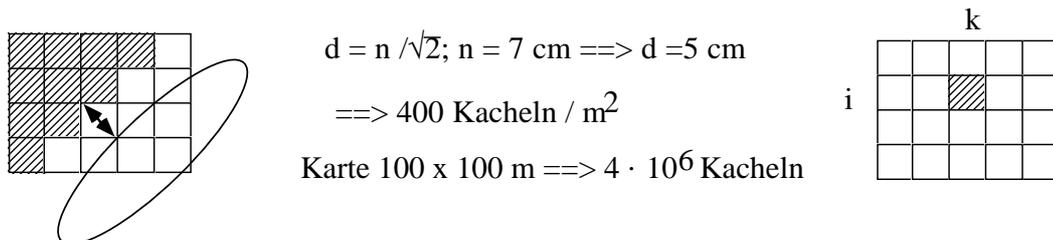
Zum Zeitpunkt einer Radaraufnahme ist aus den Hindernislinien und den sich daraus ergebenden Abschattungen klar, was Freiraum ist und wo nicht bekanntes Gelände liegt. Nach der Fahrt zu einer anderen Stelle ergibt die Fusionierung ggf. einen Freiraum mit nur noch wenigen unbekanntem Flecken, die schließlich durch Schnitt der Freiraumflächen auch noch bestimmt werden.



5.1.3. Rasterkarten

5.1.3.1. Tesselation

Üblich ist eine Tesselation der Ebene durch ein quadratisches Raster mit Kacheln der Seitenlänge d . Jede Kachel hat einen Index (i, k) . Die Kachelgröße ist an die Fahraufgabe angepasst: wenn der Roboter Kacheln im freien Raum ausfüllt, soll noch n cm Raum zu einem Hindernis sein...

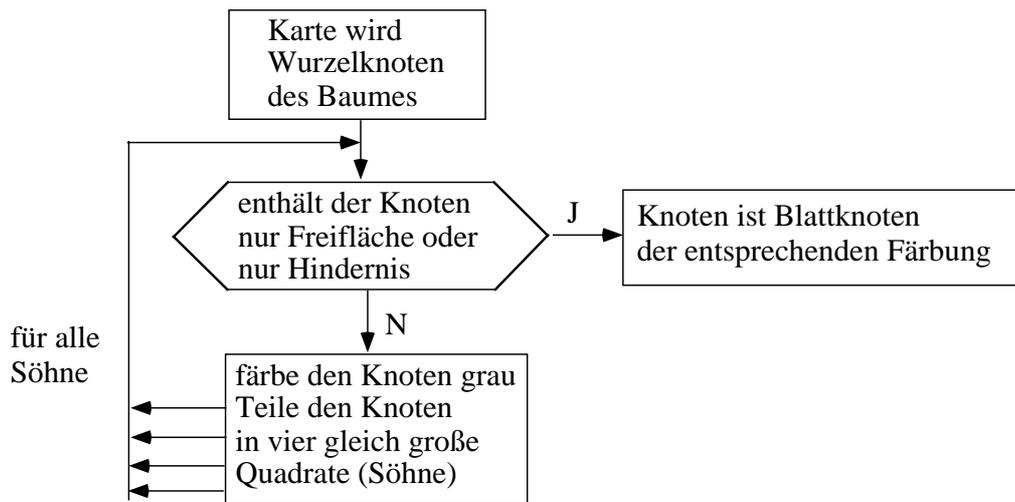


Die Belegung der Kacheln ist: " Freiraum", "Hindernis", "unbekannt", "Roboter", ...
 Es reichen 1 Byte pro Kachel zur Kennzeichnung der Rasterkarte.

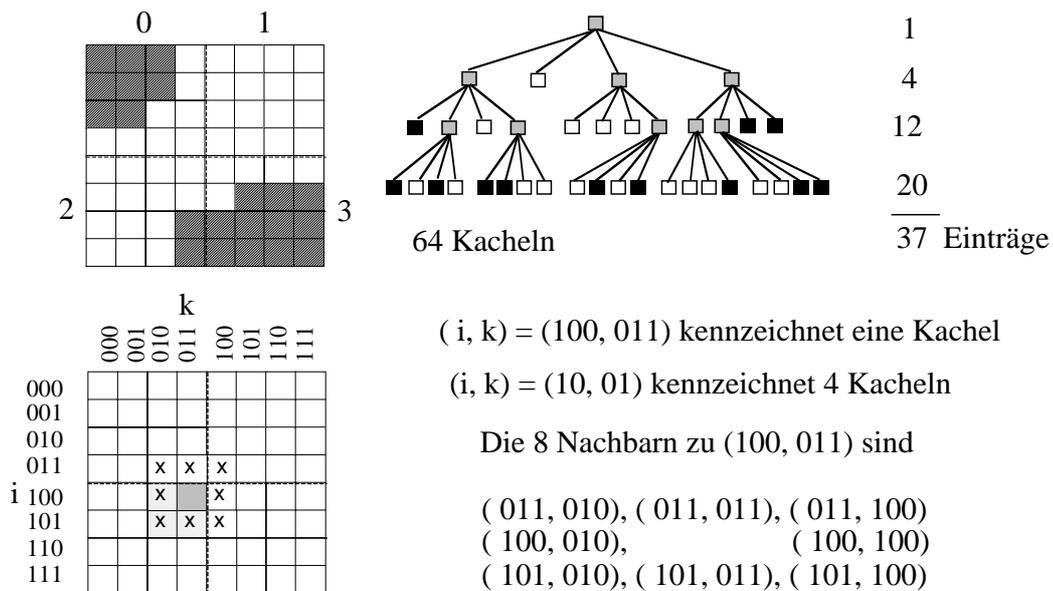
5.1.3.2. **Quadtree-Darstellung**

Eine Speicherplatz sparende Darstellung bei wenigen kompakten Hindernissen im Freiraum ist die Darstellung als Quadtree: die Karte wird gerastert und rekursiv geviertelt. Komplette freie oder belegte Viertel sind "weiße" oder "schwarze" Blattknoten des Quadtree; Viertel, die Freiraum und Hindernisse enthalten, sind grau gefärbte Knoten. Das ihnen zugeordnete Viertel wird weiter geviertelt, bis man ggf. auf der Ebene der Einzelkacheln landet.

Das Flußdiagramm zeigt das Vorgehen.



Hier ist ein Beispiel, das zugleich die Nummerierung der Kachel zeigt:

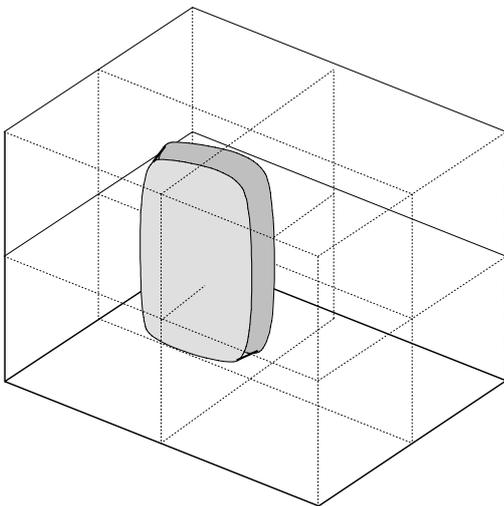


Quadtree-Definitionen

- Quadtree
rekursive Zerlegung eines Rasterbildes in einheitlich gefärbte $2^i \times 2^i$ große Blöcke
- Knoten
repräsentiert einen $2^i \times 2^i$ großen quadratischen Ausschnitt des Rasters
- freier Knoten
repräsentiert eine Fläche freien Raumes
- Hindernisknoten
repräsentiert ein Hindernisgebiet
- grauer Knoten
repräsentiert eine Mischung aus Freiflächen und Hindernis
- Blattknoten
Darstellung einer nicht weiter zerlegbaren Fläche

Octree (3D-Quadtree)

Er entsteht durch rekursive Aufteilung eines $2^n \times 2^n \times 2^n$ Voxelraumes in 8 Kuben der Größe $2^{n-1} \times 2^{n-1} \times 2^{n-1}$. In der gleichen Weise wie beim Quadtree werden den 8 Kuben Knoten eines Graphen zugeordnet und Knoten, die grau gefärbt sind, expandiert durch Aufteilung in 8 weitere Kuben bis zur Ebene der minimalen Voxel.



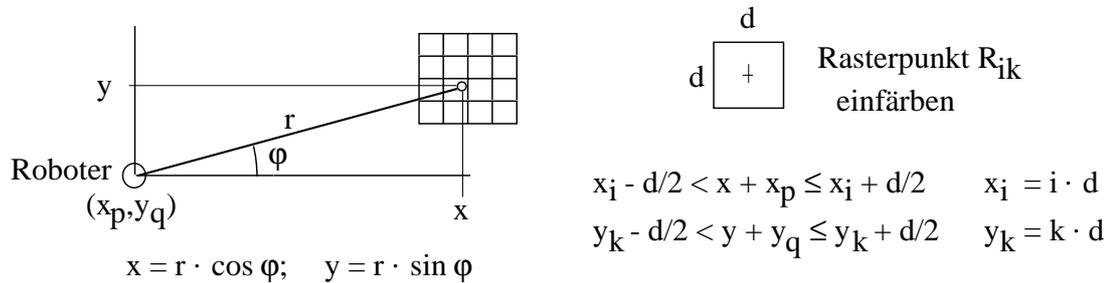
5.1.3.3. Aufnahme einer Rasterradarkarte

Die Kachelgröße werde hier so gewählt, dass sie übereinstimmt mit der Genauigkeit der Radarkarte. Sei $\Delta r \approx \pm 2,5$ cm und die Radarkeule $0,5^\circ$

$$\implies s = 5 \text{ cm in } r = s \cdot 360 / (2\pi \cdot 0,5) = 570 \text{ cm Entfernung.}$$

Bis in diese Entfernung belegt der Radarpunkt eine Kachel und die Keule Freiraum.

Sei der Roboter an der Stelle (x_p, y_q) und misst ein Hindernis bei (r, φ) . Dann ist der Rasterpunkt (x_i, y_k) einzufärben, auf den der Strahl bei (r, φ) trifft:



Ist auf dem Weg von (x_p, y_q) nach (x_i, y_k) ein Hindernis eingetragen, so ist es zu löschen oder zu altern: seine Wahrscheinlichkeit nimmt ab $p_H := p_H - \Delta$

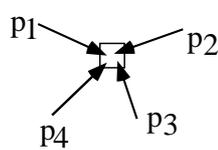
Welche Kacheln dabei berücksichtigt werden müssen, werden durch den Bresenham-Algorithmus bestimmt.

Das Einfügen oder Löschen von belegten Kacheln kann eine Umorganisation eines Quadrees zur Folge haben. Günstiger ist dann, temporär eine Rasterdarstellung zu wählen und erst am Ende einer Radaraufnahme die Umwandlung der Rasterdarstellung in einen Quadtree vorzunehmen.

5.1.3.4. Wahrscheinlichkeiten

Eine Rasterkarte eignet sich gut, um unscharfe Werte für Freiraum und Hindernisse durch Eintrag einer Hinderniswahrscheinlichkeit bei den Rasterpunkten zu modellieren, besonders sinnvoll bei Ultraschallsensoren, die von sich aus unscharfe Werte liefern.

Bei mehreren unabhängigen Messungen überlagern sich die Wahrscheinlichkeiten (Mittelwertbildung bei Festhalten des Gewichts)



P_n = Wahrscheinlichkeit, ein Hindernis zu enthalten, gestützt durch n Messungen

p_i = Hinderniswahrscheinlichkeit bei Messung i

$$P_3 = (p_1 + p_2 + p_3) / 3 ; 3$$

$$P_4 = [\{ (p_1 + p_2 + p_3) / 3 \} \cdot 3 + p_4] / 4 ; 4$$

$$= (p_1 + p_2 + p_3 + p_4) / 4 ; 4$$

5.1.3.5. Fusion von Messdaten verschiedener Messsysteme

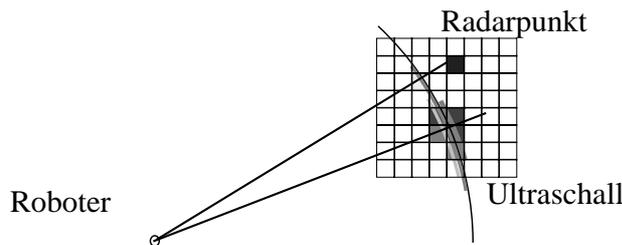
Im gleichen Rasterpunkt Eintragungen von verschiedenen Messsystemen

==> teilweises Auflösen von Widersprüchen:

Hindernis vs. Freiraum ==> Hindernis (Glastür: Ultraschall vs. Laser)

Hindernis möglich vs. Freiraum etabliert

==> Freiraum (Laser in Keule Ultraschall)



Gewichtung mit Wahrscheinlichkeiten

5.2. Modellierung der Umwelt durch Szenen

Beschreibung von Szenen durch vorverarbeitete Sensoreindrücke und Ortsangaben

Scene = $\{(M_1, M_2, \dots, M_k), (x, y, \varphi)\}$;

M_i = Merkmal i aus Sensoreindruck am Ort (x, y, φ) .

Aus den Rohdaten eines Sensoreindrucks sind für die Scene relevante Merkmale zu extrahieren. Die Extraktion hängt ab von den verwendeten Sensoren.

5.2.1. Verdichtung von Bildern

Sensor: Kamera mit Pixelbild $\sim 400 \times 500$ Pixel; s/w-, Grau- oder Farbbild

Interessant: Kontrastkanten - Übergänge hell <--> dunkel oder Farbe 1 <--> Farbe 2

Schwierigkeit: Beleuchtung i.allg. nicht beeinflussbar

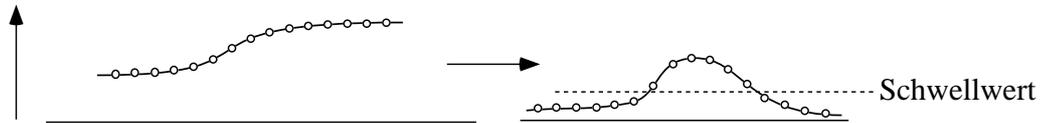
==> Kontrastkanten im Bild stammen von

- unterschiedlicher Beleuchtung --> Artefakte (Schatten, Lichtkringel, ...)
- unterschiedlicher Reflexion --> inhärente Eigenschaft der Scene

Ohne weitere Interpretation sind Kontrastkanten für sich nicht sehr aussagekräftig, sind andererseits neben der Farbe wesentliche Merkmale in Bildern und sollten extrahiert werden. Die Interpretation führt zu Fragen der Objekterkennung, die in Kapitel 7 kurz angesprochen werden. Die Extraktion von Kontrastkanten wird durch gängige Bildverarbeitungssoftware geleistet. Oft geht ein Verfahren zweistufig vor: zunächst werden Kontraste im Bild verschärft und dann mit Schwellwerten Kontrastkanten gefunden.

- Kontrastverschärfung in einem Bild : Änderungen werden herausgehoben

Anwenden eines Punktfilters auf alle Pixel $P(u,v) \rightarrow x_{i,k}$; $Q(u,v) = \left| \frac{\partial P}{\partial u} \right| + \left| \frac{\partial P}{\partial v} \right|$



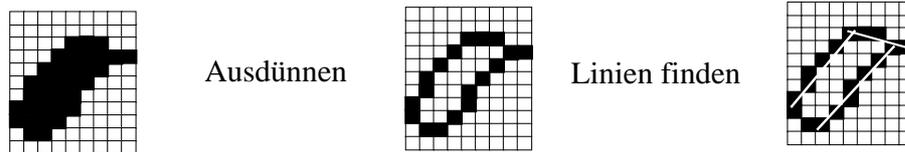
$$y_{i,k} = |(x_{i,k+1} - x_{i,k}) + (x_{i,k} - x_{i,k-1})| + |(x_{i+1,k} - x_{i,k}) + (x_{i,k} - x_{i-1,k})|$$

$$y_{i,k} = |x_{i,k+1} - x_{i,k-1}| + |x_{i+1,k} - x_{i-1,k}| \quad \{ * \text{differenzieren nach } u \text{ und } v * \}$$

- Ermitteln von Kontrastkanten

Feststellen der Richtung eines Kantenverlaufs über viele Pixel hinweg

Binarisierung des kontrastverschärften Bildes durch Schwellwert



5.2.2. Szenenklassifikation

Die Klassifikation einer Szene kann auf vorverarbeiteten (Pixel)daten oder Rohdaten erfolgen.

5.2.2.1. Vorverarbeitete Pixeldaten, nicht interpretiert

Betrachte Kanten über mehr als 10 Pixel; kurze Kanten beschreiben feine Details.

Anzahl nach Längen und Richtungen der Kanten in relativ grober Aufteilung geordnet.

Richtungen: senkrecht, waagrecht, 23° , 45° , 67° , -23° , -45° , -67° jeweils $\pm 11^\circ$

Längen: 10 - 20, 20 - 40, 40 - 80, 80 - 160, > 160 Pixel
 z. B. # waagerechte Linien über 20 - 40 Pixel
 # senkrechte Linien über 80 - 160 Pixel

\Rightarrow Vektor V mit 40 Komponenten $V = (m_1, m_2, \dots, m_{40})$

Eingang in einen Vektorklassifizierer; Ausgang Nr. eines typ. Vektors S_j .

Beschreibung der Szene durch (j, x, y, ϕ) : Am Ort (x, y, ϕ) wurde eine Szene aufgenommen, die ähnlicher zu S_j ist als zu allen anderen typischen Szenen.

5.2.2.2. **Klassifikation auf den Rohdaten**

Beispiel: ALICE (U. Zimmer)
 Robot ca. 30 cm Ø mit Synchrodrive
 Sensorsituation beschrieben durch
 24 Lichtwerte I_0, \dots, I_{23} zu 8 Bit
 24 Tastwerte T_0, \dots, T_{23} aus 0/1
 nach Messung aller I_i und T_i Anpassung

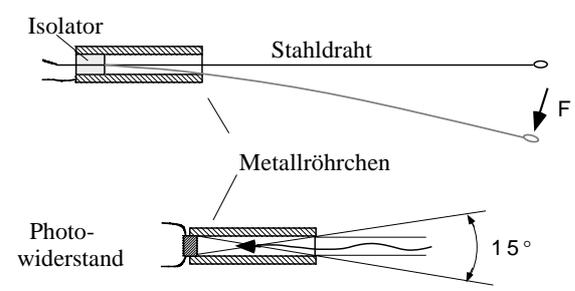
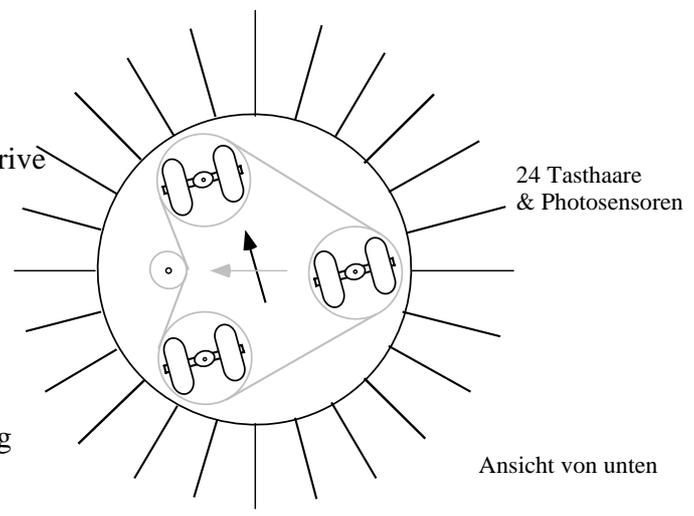
- der Lichtwerte auf Maximum

$\implies I_i := I_i / I_{max} \in [0,1]$

- der Tastwerte ($i \bmod 24$) wg. Unsicherheit

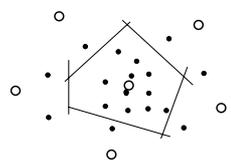
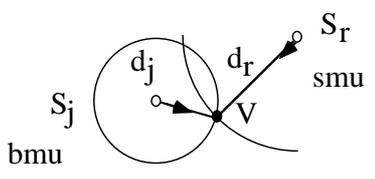
$$T_i := (2 T_i + T_{i-1} + T_{i+1}) / 4$$

\implies Vektor mit 48 Komponenten



5.2.3. **Vektorklassifizierer**

Sei der Eingabevektor $V = (m_1, \dots, m_q)$.
 Seien S_i typische Vektoren mit $i = 1, \dots, n$ und $S_i = (s_{i1}, \dots, s_{iq})$.
 Es werden euklidische Abstände d_i gerechnet; sei d_0 ein Minimalabstand (problemabh.).
 Zu jedem typ. Vektor S_i gibt es einen Matchingzähler z_i mit $1 \leq z_i \leq Z$.
 z_i wird inkrementiert, wenn S_i der nächstgelegene Vektor ist (best matching unit, bmu)
 Wird $z_i > Z$ ist S_i zu häufig typ. Vektor, seine Umgebung ist unterrepräsentiert.
 Ist S_j bmu, wird S_j in Richtung V geschoben (Faktor ϵ).
 Ebenso die zweitnächste Vektor S_r (smu - second best matching unit) mit Faktor ϵ_s .



Als Beispiel eines Vektorklassifizierers soll das **Neural Gas Net nach Fritzke** betrachtet werden:

Initialisierung:

$n := 1$; $S_n := V$ (der erste Vektor wird als typisch angesehen)

$z_n := 1$ (Matchingzähler bei S_n)

begin

für jeden Eingabevektor V wird gerechnet

- für alle i bestimme $d_i = |V - S_i|$;
- sei $d_j = \min_i \{d_i, i = 1, \dots, n\}$ und $d_j \leq d_0 \implies$ Ausgabe von j

S_j ist "best matching unit" (bmu)

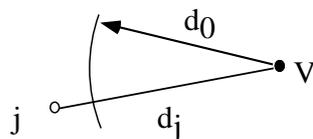
Anpassen der Komponenten in Richtung V : $s_{kj} := s_{kj} + \varepsilon \cdot m_k$ für alle $k = 1, \dots, q$
 ε kleiner Faktor 0,01 ... 0,1 beschreibt Einfluß von V auf bmu

- sei $d_r \leq d_i$ für alle $i \neq j, r \implies S_r$ ist "second best matching unit" (smu)
 Anpassen: mit $\varepsilon_s < \varepsilon$ wird $s_{rk} := s_{rk} + \varepsilon_s \cdot m_k$ für alle Komponenten von S_r .

- inkrementieren des Matchingzählers $z_j := z_j + 1$
 if $z_j \geq Z$ (Umgebung von S_j ist unterrepräsentiert)
 $\implies z_j := 1$; $n := n+1$;
 $S_n := V$; $z_n := 1$ (V ist neuer typ. Vektor)

- sei $d_j > d_0$ (unähnlich allen anderen Szenen) $\implies n := n+1$; $S_n := V$
 V wird ein neuer typischer Vektor

end.



Die Anzahl der Repräsentanten passt sich an das Problem an.

Minimalabstand d_0 und Anpassungsfaktoren ε und ε_s sind problemabhängig.

5.2.4. Objekterkennung aus Sensordaten

5.2.4.1. Anwendung von Bilderkennungsalgorithmen

Die intendierte Anwendung bestimmt die interessierenden Kantenlängen

Beispiel:

entfernter Türrahmen nimmt in Bild ca. die Hälfte ein ==> Linie über ca. 100 Pixel
 in der Nähe Namensschild ca. 30% des Bildes ==> Linie über ca. 60 Pixel
 Bild stellt Flur dar ==> Kanten von 10% Bildgröße ==> Linien über ca. 20 Pixel

zusätzliche Hinzunahme von Daten entfernungsgebender Sensoren

==> Entfernungssprünge kennzeichnen Objektkanten

Farbsegmentierung erlaubt weitere Klassifikation wie Türen in Rahmen

bei Vorwissen über die Lage von Objekten weitere Klassifikation möglich

z. B. Raumnummern erfassen aus Türschildern

Türgriffe erkennen

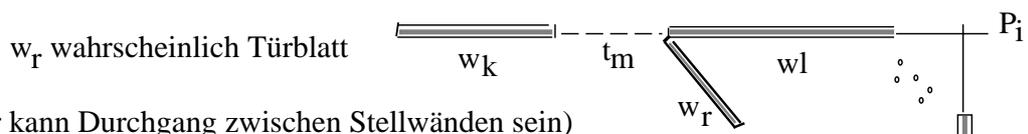
5.2.4.2. Objekterkennung aus Radarkarten

Extraktion von Linien als Hindernisbegrenzungen

lange Linien in rechten Winkeln zueinander : wahrscheinlich Wände

parallele Linien von zwei Seiten gesehen --> Wandstück w_k 

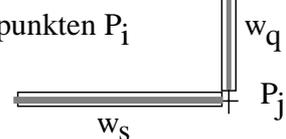
Wandstücke in einer Linie mit befahrbarer Lücke dazwischen --> Wand mit Tür t_m



(Tür kann Durchgang zwischen Stellwänden sein)

Verlängerung von Wandstücke hinter Hindernissen zu Schnittpunkten P_i

Schnittpunkte von Wandstücken P_j



geschlossenes Polygon aus Wandstücken mit Türen und Wänden

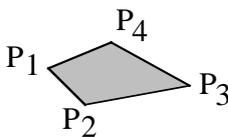
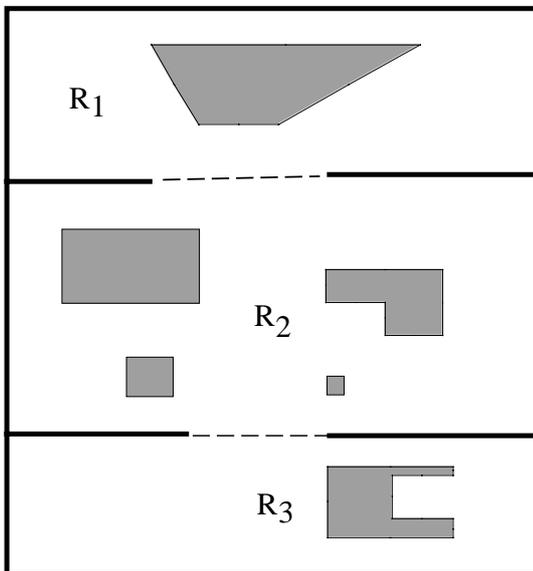
mit Schnittpunkten P_1, \dots, P_n ==> ein Raum R_n

==> Konzepte "Wand", "Tür", "Raum"

Mit diesen Konzepten kann dann eine geometrische Karte aufgebaut werden, in der der AMR navigieren kann.

Geschlossenes Polygon aus Hindernislinien mit Hindernis im Innenraum

==> geschlossenes Hindernis

==> geometrische Karte
aus interpretierten
Hindernislinien

Wände —————
Türen - - - - -
geschlossene Hindernisse 
Räumen R₁, R₂, R₃

5.2.5. Landkarten

Geometrische Karten mit **interpretierten** Symbolen und Längenangaben

- maßstäbliche Abbildungen realer Umgebungen in verschiedenem Detaillierungsgrad
- viel mehr als befahrbarer Raum und Hindernisse
- Gebäudepläne mit Räumen, Fenstern, Türen, WC's, Steckdosen, ...
- Stadtpläne mit Straßen, Plätzen, Telefonzellen, Schulen, Kirchen, ...
- Geländekarten mit Wegen, Wald, Wiese, Sumpf, Dorf und Stadt, ...
- Landkarten mit Straßen, Autobahnen, Eisenbahntrassen, Orten, Gebirgen, ...
- Spezialkarten, die bestimmte Teile herausstellen:
Elektroleitungen, Abwassersystem, ... in Innenräumen
Drainagen, Hochspannungsleitungen, ... draußen

5.2.5.1. Aufbau einer Karte (Edlinger und Weiß)

5.2.5.2. Voraussetzungen

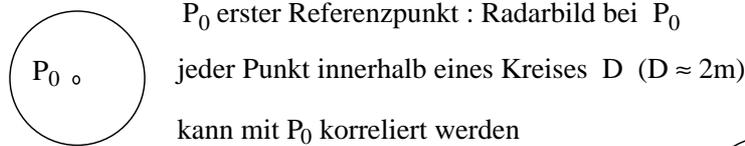
- keine a priori Kenntnis der Umgebung
- Verwendung einer groben "current sensor map" CSM
(Fusion von Radarkarte und grobauflösendem Hindernisradar mit 30 Sektoren in 180°
==> 60 Sektoren von 6°)

Bedingungen

- von einer Startposition P_0 aus beginne mit der Exploration der Umgebung
- jedes bei der Exploration gefundene Hindernis wird als permanent angesehen
- Aufbau einer Grobkarte für die Navigation
- wissen, wo Freiraum zum Navigieren ist
- wissen, wann die Umgebung exploriert ist

5.2.5.3. Referenzpunkte

Bei der Exploration werden Referenzpunkte angelegt



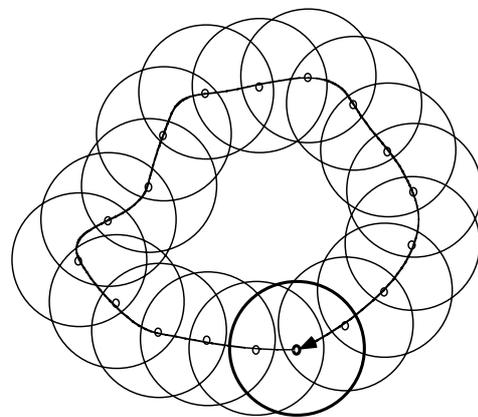
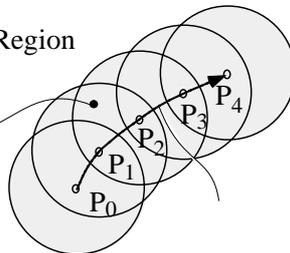
wähle den nächsten Referenzpunkt P_{k+1}

(Koordinaten & Radarbild)

im Kreis um P_k

==> die Menge $\{ P_k \}$ definiert
 eine referenzierbare Region

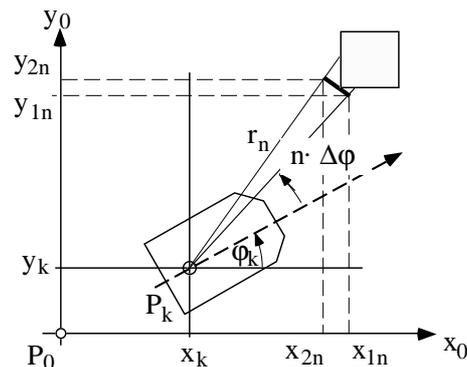
jeder Punkt im Kreis
 korreliert zu P_0



closed path
 $P_0 = P_K ?$

Information beim Referenzpunkt P_k

- Position und Orientierung in Bezug auf P_0
 $P_k = (x_k, y_k, \phi_k)$
- Radarkarte $\{ r_i \} i = 0, \dots, 719 ; \phi_i = i \cdot \delta\phi ; \delta\phi = 0,5^\circ$
- "current sensor map" $\{(x_{1n}, y_{1n}), (x_{2n}, y_{2n})\} n = 0, \dots, 59;$



mit $x_{1n} = x_k + r_n \cdot \sin(\phi_k + n \cdot \Delta\phi)$

$y_{1n} = y_k + r_n \cdot \cos(\phi_k + n \cdot \Delta\phi)$

$x_{2n} = x_k + r_n \cdot \sin(\phi_k + (n+1) \cdot \Delta\phi)$

und $y_{2n} = y_k + r_n \cdot \cos(\phi_k + (n+1) \cdot \Delta\phi)$

$\Delta\phi = 6^\circ$

5.2.5.4. Explorationsstrategie

(*) nimm eine grobe "current sensor map" CSM auf
(60 Sektoren über 360°; nur in Vorwärtsrichtung)
akkumuliere sie in die Karte

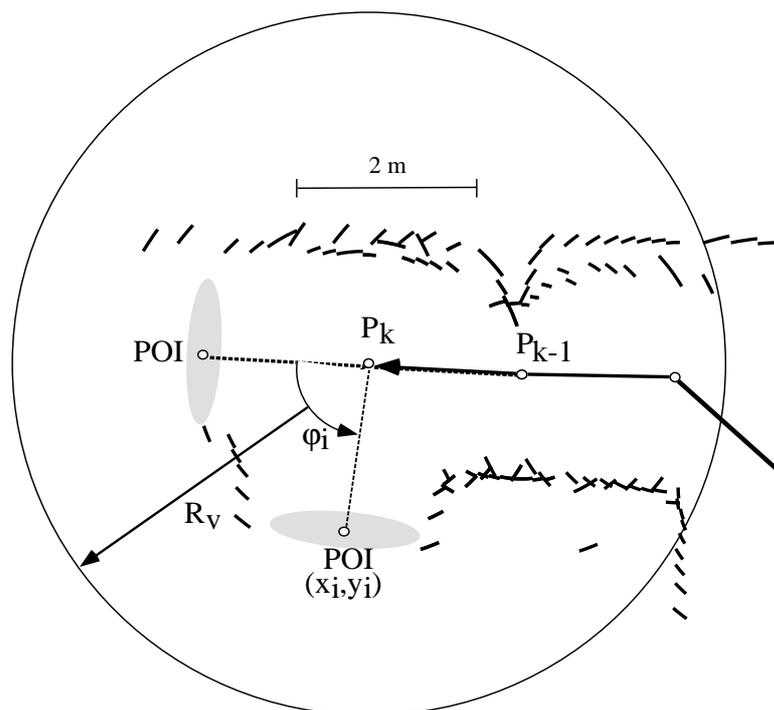
suche Durchfahrten

wenn es Durchfahrten gibt markiere Anfahrpunkte (point of interest; POI)
wähle einen aus, lege die anderen auf einen Stack
fahre zu diesem POI mache bei (*) weiter

wenn es keine Durchfahrt gibt und der Stack nicht leer ist
nimm ein POI vom Stack und fahre dorthin
weiter bei (*)

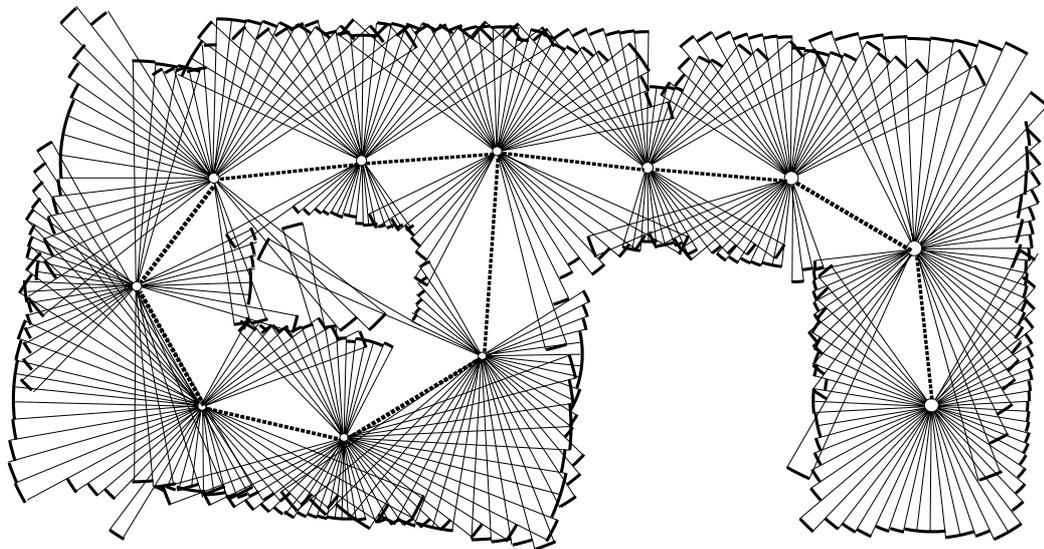
Wenn der Stack leer ist, ist die Exploration zu Ende und eine Karte der Umgebung erstellt.◇

Beispiel für Anfahrpunkte (point of interest, POI)

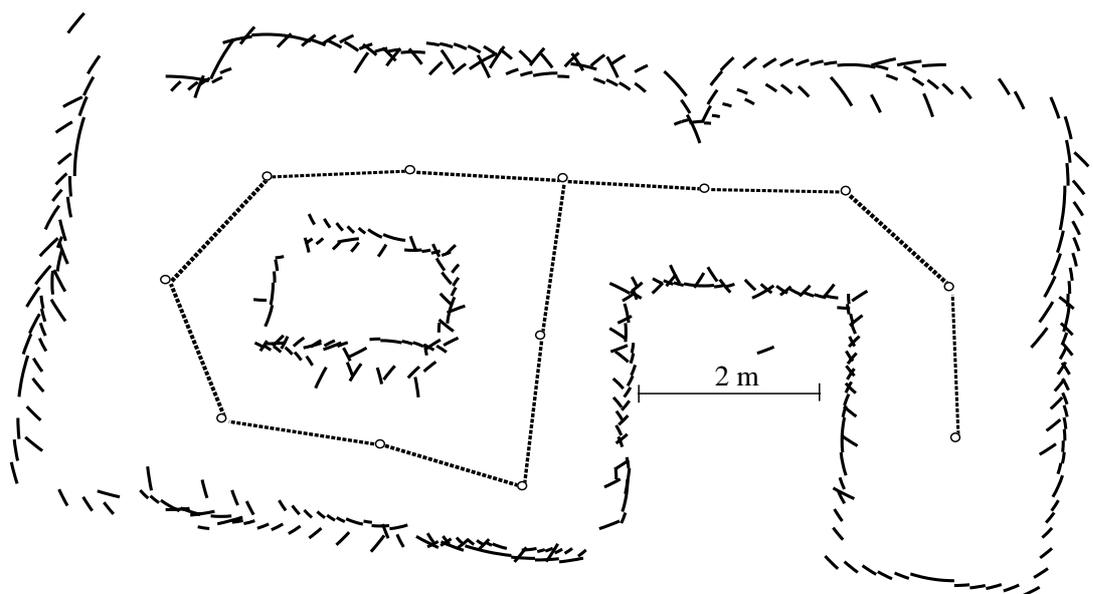


Die Durchfahrtpunkte liegen so im Freiraum, daß der AMR sie vom gegenwärtigen Punkt auf gerader Linie anfahren kann und sie sind nur soweit entfernt, dass bis dahin die Orientierung und Position gehalten werden kann. Sie sollen auch nicht am Rande der Sensorreichweite liegen, um beim Anfahren eines POI nicht plötzlich vor einer Wand zu stehen.

Das nächste Bild zeigt eine Testumgebung mit Referenzpunkten, den Aufnahmepunkten für die "current sensor maps", hier mit einer Auflösung von $\Delta\phi = 6^\circ$:

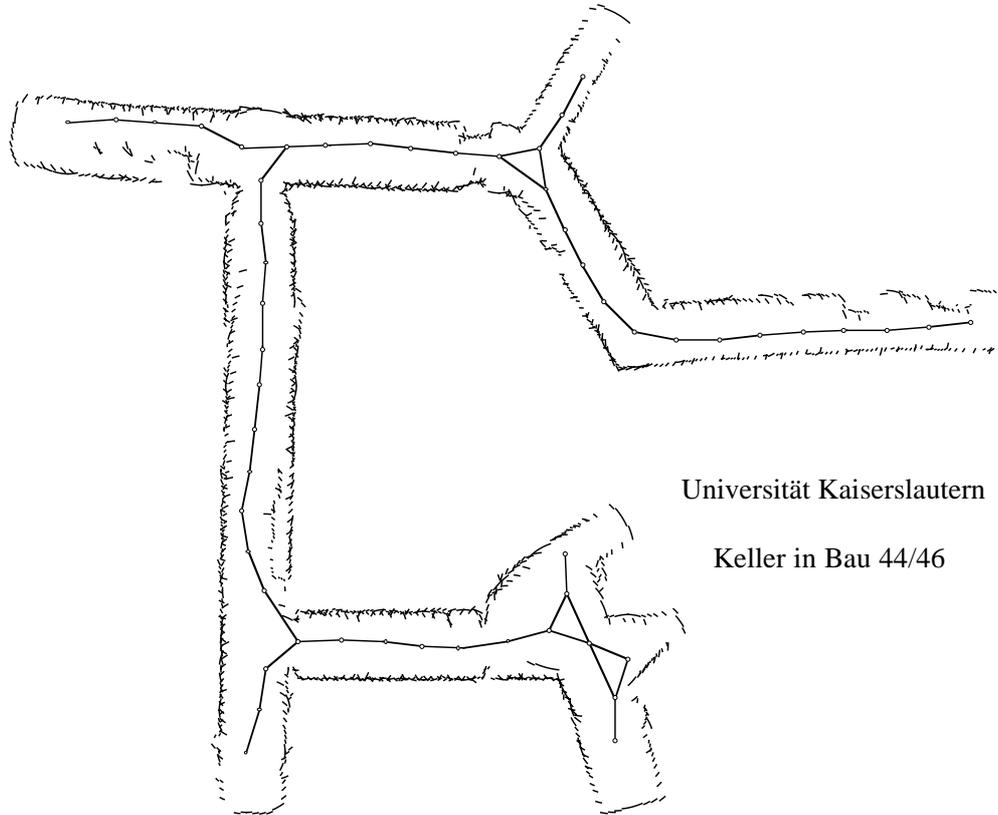


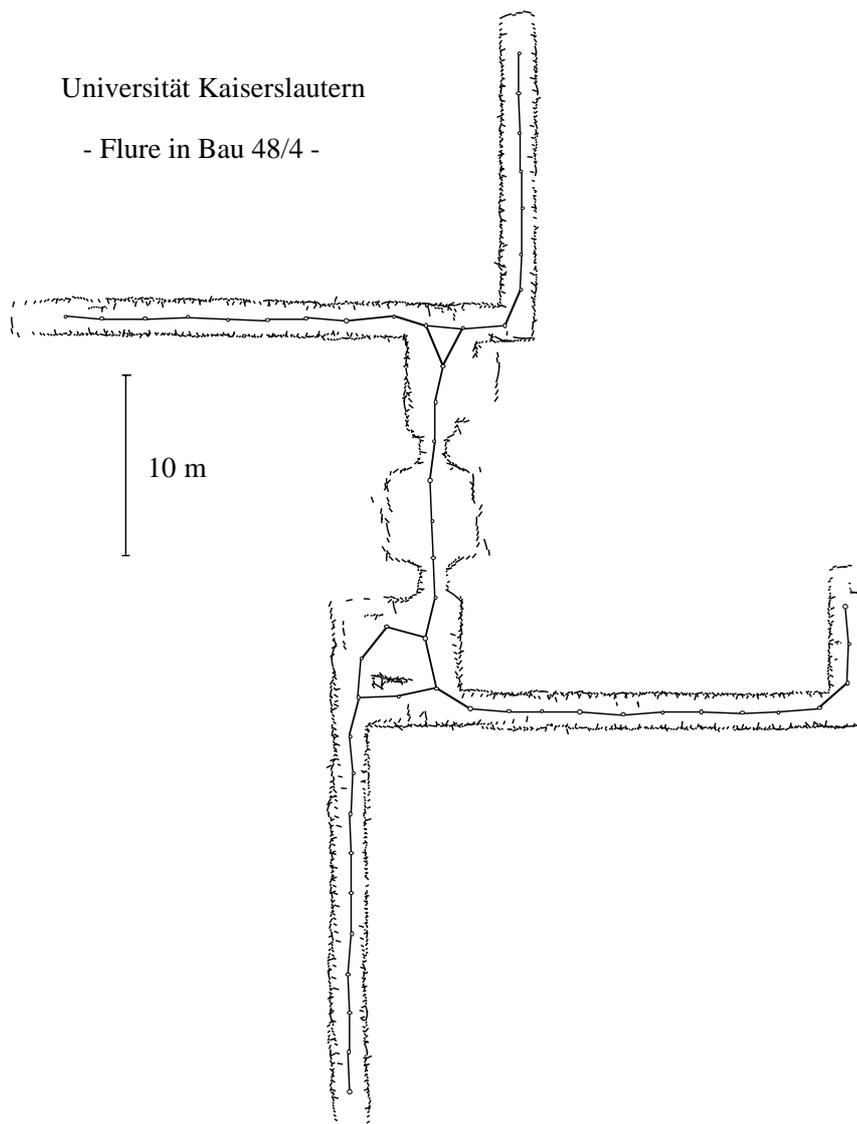
Die Überlagerung der einzelnen Aufnahmen ergibt eine grobe geometrische Karte der Testumgebung, die für Zwecke der Navigation auf den Verbindungslinien zwischen Referenzpunkten ausreicht.



Die nächsten beiden Bilder zeigen den Keller von Bau 46 und den 4. Stock von Bau 48. Der AMR hat diese Umgebungen selbständig exploriert und sich die Karten aufgebaut.

Eingezeichnet sind die Referenzpunkte und die Verbindungen dazwischen. Sie spannen einen topologischen Graphen auf, auf dem der AMR navigiert.





5.3. Topologische Karten

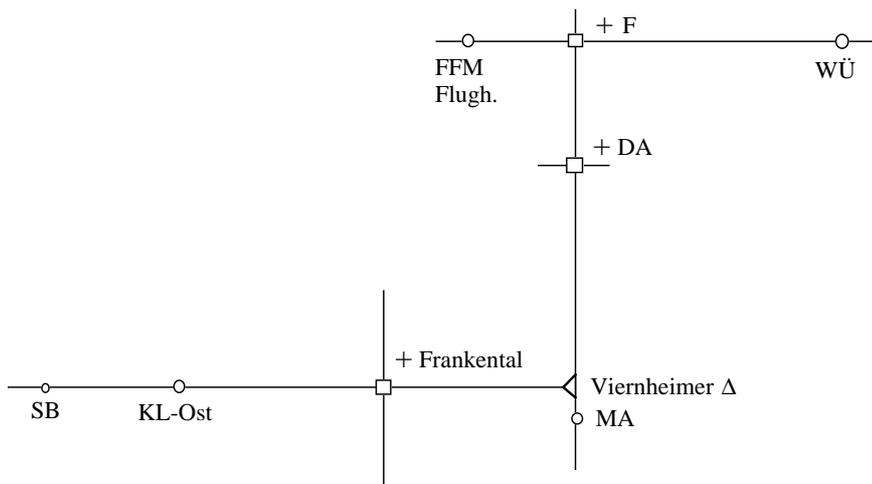
5.3.1. Definition

Topologische Karten sind interpretierte Graphen, nicht notwendig maßstabgerechte Abbildungen der Umgebung.

Knoten: Plätze, attributierte Objekte

Kanten: Verbindungen dazwischen, ggf. mit Attributen

Beispiele: Elektroleitungen und Verbraucher in einem Gebäude
Ethernet einer Uni
Haltestellen und Buslinien eines städtischen Netzes
Netzplan der Eisenbahn (Bahnhöfe und Strecken)
Autobahnnetz (Kreuzungen, Raststätten, Ausfahrten)



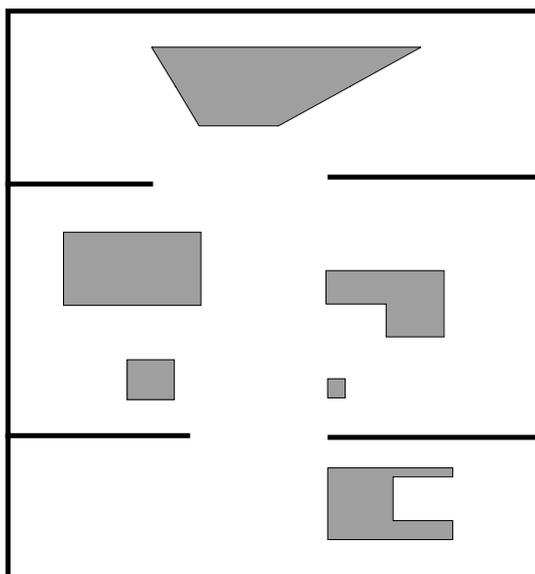
Hier gezeigt:

Graph der Autobahnverbindung Flughafen Frankfurt/Main <-> Kaiserslautern-Ost.

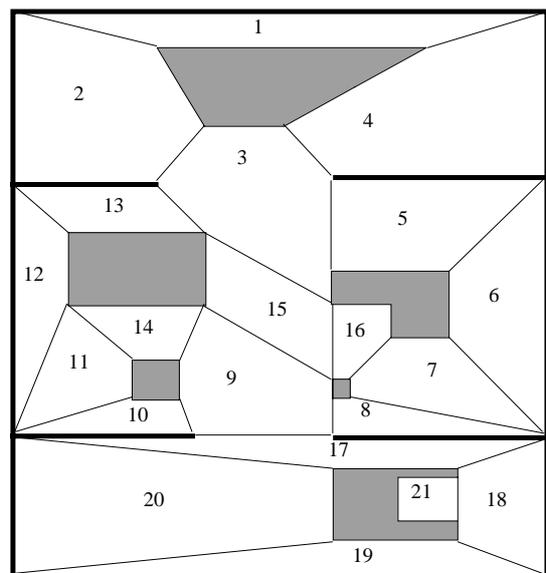
Die Knoten des Graphen sind die Abzweigungen, an denen ggf. die Autobahn zu wechseln ist, und die Städte, die auf den Richtungsschildern zu finden sind.

5.3.2. Umwandlung geometrische in topologische Karte

Hier dargestellt am Beispiel der Umwandlung einer geometrischen Karte aus nicht interpretierten Hindernislinien in eine topologische Karte.



geometrische Karte



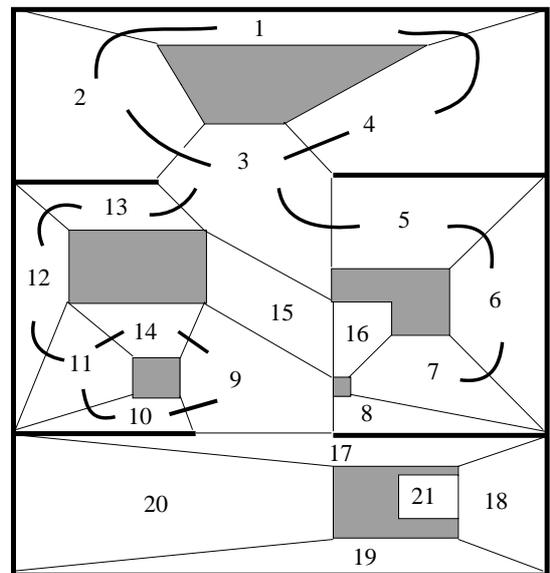
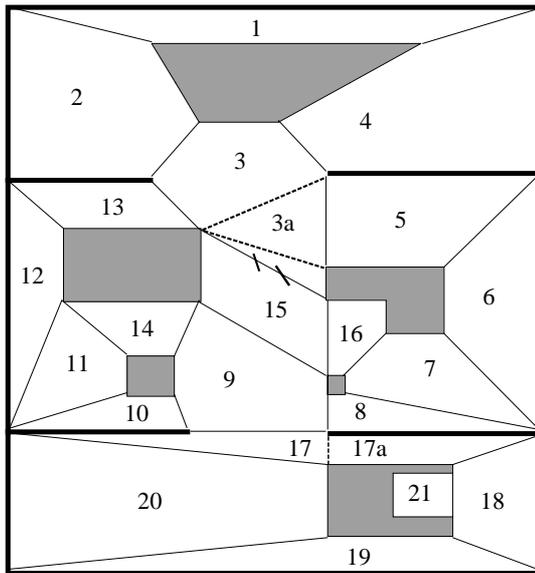
Verbindung der Eckpunkte
und Nummerierung der Gebiete

5.3.2.1. Konvexe Polygone

In einem ersten Schritt wird der Freiraum in der geometrischen Karte in konvexe Polygone aufgeteilt. Dazu werden die konkaven Eckpunkte der Wände mit den nächstgelegenen konvexen Eckpunkten freistehender Hindernisse oder Wände verbunden. Die Polygone werden durchnummeriert. Die Aufteilung ist nicht eindeutig und spiegelt die Unbestimmtheit in der Zuordnung von Räumen wider, die aus dem Alltagsleben bekannt ist.

kürzeste Verbindungen von Ecken
(Grenzlinien)

befahrbare Wege über Grenzlinien
zwischen Zellen

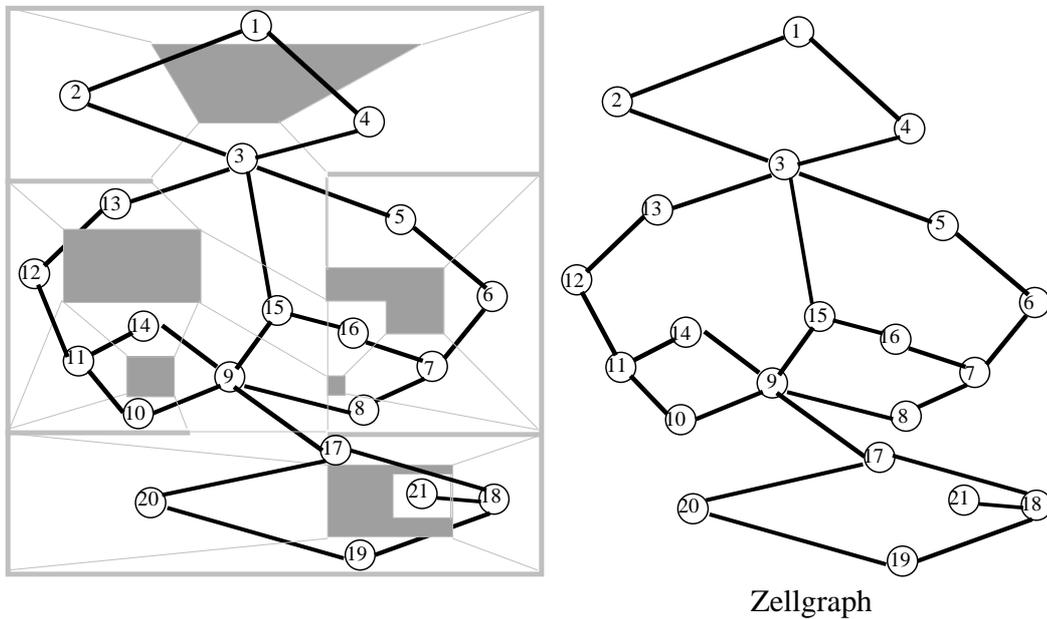


nicht eindeutige Zuordnung
zu Zellen

Die Trennlinien zwischen benachbarten Freiraumpolygonen sind die Zelllinien.

5.3.2.2. Zellgraph

Der Aufbau eines Graphen (Zellgraph) geschieht dann durch Verbindung der Zellen über die Zelllinien.



Hindernisse implizit durch Zyklen dargestellt

Die weitere Analyse auf Räume in der geometrischen Karte kann durch die Zusammenfassung von Teilgraphen erfolgen.

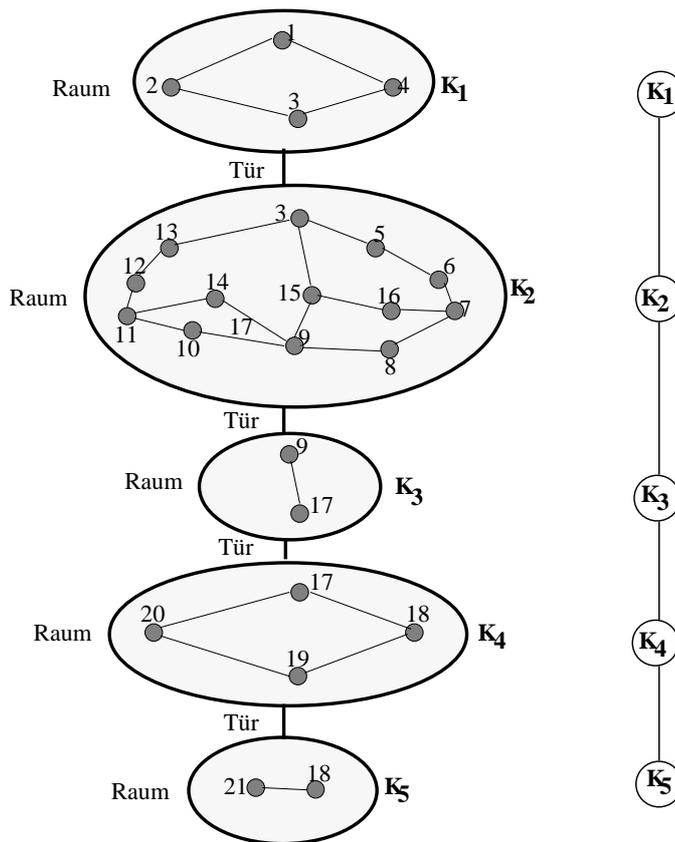
5.3.2.3. Erzeugung eines Dekompositionsbaumes

Definitionen

- Ein Graph mit mindestens $k+2$ Knoten heißt **k-zusammenhängend** genau dann, wenn er nach Entfernung von $k-1$ Knoten immer noch zusammenhängt.
- Eine **Verbindungsmenge** ist eine Menge von Knoten, deren Entfernung zu einem nicht zusammenhängenden Graphen führt.

Zerlegung des Zellgraphen in Zusammenhangskomponenten

- Betrachte die 2-zusammenhängenden Komponenten.
- Fasse Zusammenhangskomponenten und Verbindungs Mengen zu einem **Dekompositionsbaum** zusammen.



Analyse des Dekompositionsbaumes

Verschiedene Interpretationen sind zulässig.

Beispiele für Heuristiken:

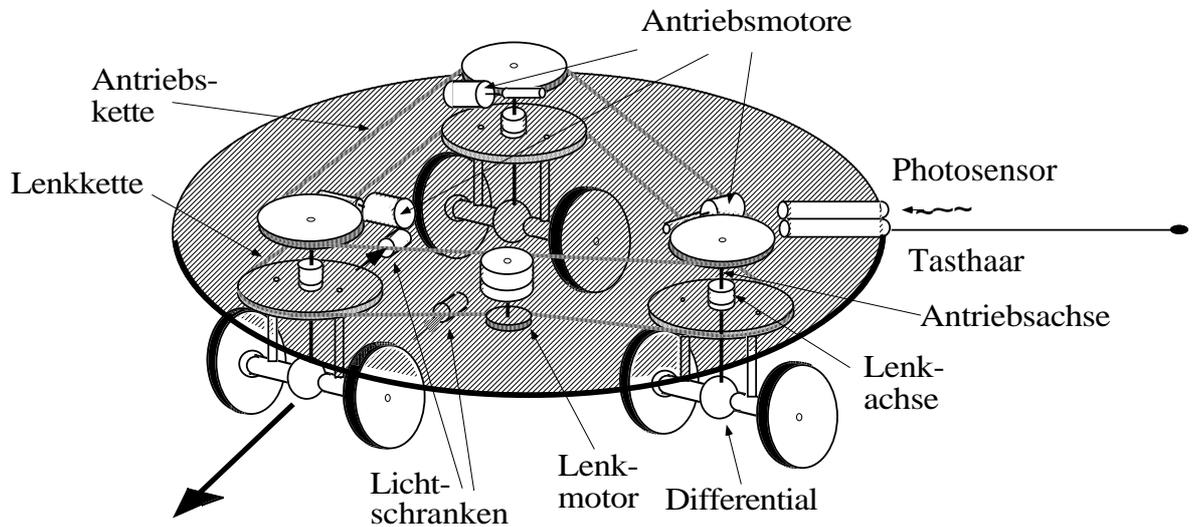
- Eine Kante, die zwei Komponenten verbindet, ist "Tür" oder "Korridor".
- Zwei benachbarte 2-zusammenhängende Komponenten, die keine Kante sind, werden als zwei verschiedene Räume markiert, falls ihr gemeinsamer Verbindungsknoten eine Zelle mit kleiner Fläche ist.
- Zwei benachbarte 2-zusammenhängende Komponenten, die eine Kante bilden, werden als ein Raum aufgefasst, falls ihr gemeinsamer Verbindungsknoten eine Zelle mit großer Oberfläche ist.

5.3.3. ALICE: Erzeugung eines topologischen Graphen aus Rohdaten (U. Zimmer)

Mit einem sehr einfachen AMR wird aus den Rohdaten einfacher Sensoren ein topologischer Graph erzeugt, der die Umgebung repräsentiert und auf dem der AMR navigieren kann.

5.3.3.1. Der AMR ALICE

Die Kinematik der ALICE ist die eines Synchrodrive.



Es wird eine gewollt unpräzise Mechanik mit Losen - ähnlich wie bei Lebewesen - verwendet; der Aufbau der Karte soll davon kaum berührt werden. Es gibt großes Driften in Lenkwinkeln und Wegen und eine primitive Odometrie mit Lichtschranken an der Antriebs- und Lenkkette für v und ω .

5.3.3.2. Konnektionistischer Ansatz

Ziel: Darstellung der Umwelt durch ein Netz **typischer** Sensorsituationen

Definition:

Sensorsituation:

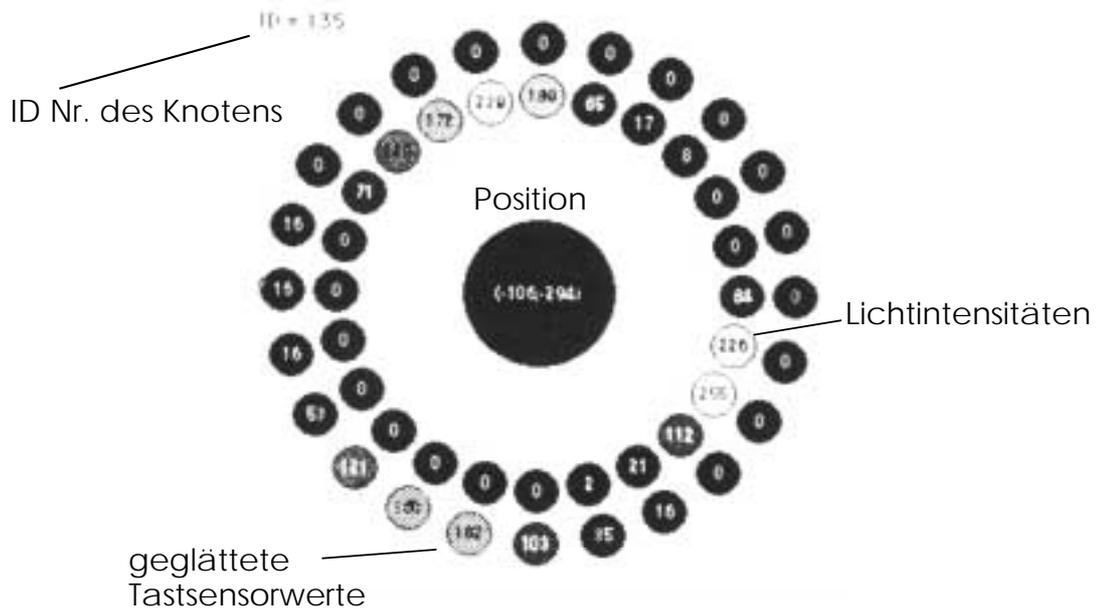
Vektor mit 50 Komponenten

- 24 Lichtwerte der Photosensoren, normiert auf Umgebungshelligkeit
- 24 Tastsensorwerte, geglättet mit einer Gaussfunktion
- 2 Positionswerte (x, y) in cm aus den Quadraturzählern der Lichtschranken.

Durch Gewichtungsfaktoren werden die Komponenten einander angeglichen.

Alle 250 ms wird eine neue Situation erfasst.

Eine Sensorsituation zeigt das folgende Bild (einer der Knoten des Graphen):



5.3.3.3. Aufbau des Graphen aus Sensorsituationen

Es wird ein Neuronales Netz aufgebaut (Fritzke Gas Net).

Zu jeder neuen Sensorsituation wird die **typische** Sensorsituation T_i ermittelt:
 für eine Sensorsituation S ist T_i am ähnlichsten
 $\iff T_i$ hat den kleinsten Abstand zu S

$$\text{Abstand} \quad \| S - T_i \| = \left(\sum_{k=1}^{50} (t_{ik} - s_k)^2 \right)^{1/2} \quad 0 \leq s_k < 256$$

T_i ist "best matching unit" **bmu** zu S

in der Umkehrung: T_i ist Attraktor für alle S , deren bmu T_i ist

$\implies T_i$ bilden die Knoten des Netzes

wenn T_i bmu wird \implies ein Knotenzähler Z_{k_i} wird inkrementiert

- Initialisierung

die erste Sensorsituation S wird zur typischen Situation T_0
 ein Besuchszähler Z_{k_0} wird auf Eins gesetzt

- Einfügeoperationen

- Knoten des Netzes

sei $\| S - T_i \| > d_{\text{ins-err}}$ oder der Knotenzähler $Z_{k_i} > n_{\text{dis}}$

$\implies S$ wird neuer typischer Knoten T_{neu}

Knotenzähler Z_{k_i} und $Z_{k_{\text{neu}}}$ werden auf eins zurückgesetzt.

{* Die Sensorsituation ist entweder völlig verschieden von allen bisher erlebten Situationen oder T_i war zu häufig bmu; dann ist die Umgebung allein durch T_i unterrepräsentiert *}

- Einfügen von Kanten

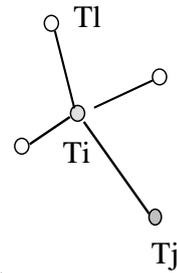
sein T_i und dann T_j nacheinander bmu

\implies zwischen T_i und T_j wird eine Kante gezogen

ein Besuchszähler $Z(i,j) = v_{max}$ gesetzt

es habe T_i k weitere Nachbarn T_l mit $l = 1 \dots k$

\implies die Besuchszähler $Z(i,l)$ werden um $\frac{V_{red}}{k+1}$ reduziert



{* Die Kanten entsprechen tatsächlich gefahrenen Wegen. Bis auf dynamische Hindernisse kann der AMR auf diesen Wegen auch später erneut fahren. *}

- Löschen von Knoten und Kanten

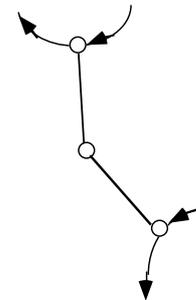
sei $Z(i,l) < v_{min}$ und T_l einer der $(k+1)$ Nachbarn von T_i

\implies die Kante zwischen T_i und T_l wird gelöscht

es habe T_i keine Nachbarn mehr

\implies der Knoten T_i wird gelöscht

\implies Elimination von Fehlmessungen

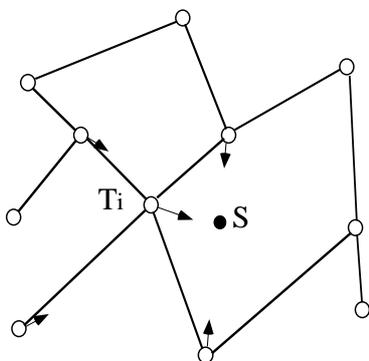


- Einfluß von S auf die Situationen um S herum

sei T_i bmu zu S und T_j die Nachbarn von T_i (es gibt eine Kante von T_i nach T_j)

\implies Anpassung: $T_i := T_i + \epsilon_{bmu} (S - T_i)$

$T_j := T_j + \epsilon_{nachbar} (S - T_j)$

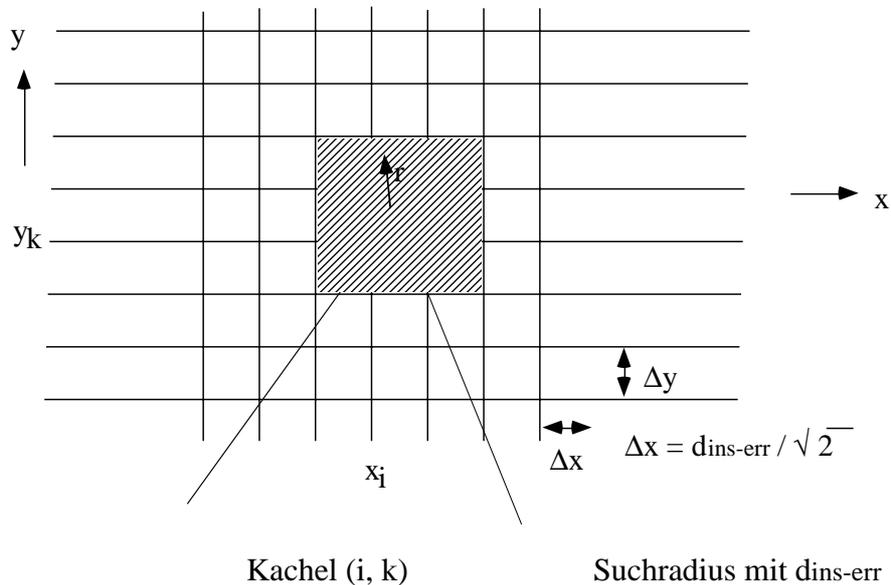


Punkte im 50-dim. Raum

5.3.3.4. Einschränkung des Suchraumes für die bmu

Für eine effektive Implementierung sollte die Anzahl der typischen Sensorsituationen, gegen die eine gegebene Sensorsituation abzugleichen ist, beschränkt bleiben. Das wird einmal erreicht durch die Hinzunahme der Position zur Sensorsituation: zwei Situationen, die räumlich weit auseinanderliegen, haben auch einen Abstand $d > d_{ins-err}$.

Zum anderen werden die T_j geeignet geclustert: die Ebene wird tesselliert in Quadrate der Seitenlänge $\Delta x = \Delta y = d_{ins-err} / \sqrt{2}$. Eine Rasterkachel (i, k) enthält dann alle T_j mit Positionen (x, y) mit $x_i \leq x \leq x_{i+1}$ und $y_k \leq y \leq y_{k+1}$. Eine Sensorsituation S am Ort (x, y) muss dann nur abgeglichen werden mit den typischen Sensorsituationen T_j in den Kacheln (i, k) und den 8 Nachbarkacheln. Dort ist, wenn überhaupt, die bmu zu finden; oder S wird zur neuen typischen Sensorsituation, da der AMR noch nie in der Nähe von (x, y) war.



5.3.3.5. Driftkorrektur

Verschärft durch die bewußt "schlabbrige" Mechanik besteht eine absolute Notwendigkeit zu einer Driftkorrektur um eine langfristige Orientierung in der Umwelt zu ermöglichen.

Zur Ermittlung der bmu zu einer Situation S muss der Robot ungefähr seine Position kennen.

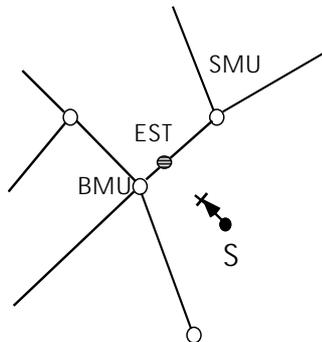
Die Drift hat zwei Ursachen:

- eine langsame Drehung der Plattform durch Ungenauigkeiten des Synchrodrives:
 - ==> Drehung des internen Koordinatensystems um Winkel α
 - Ermittlung des Korrekturwinkels α :
 - bilde Winkeldifferenz Schwerpunkt Lichtverteilung in S gegen Schwerpunkt Lichtverteilung in T_{next} , (räumlich nächste typische Situation)

- ein Driften der Position durch die ungenaue Odometrie:

Abhilfe:

Verschieben der Position von S in Richtung einer vermuteten Position EST zwischen der bmu und der nächstentfernten typischen Situation, der "second best matching unit" smu. Die Verschiebung von S ist $d_{faktor} \cdot (S. pos - EST. pos)$



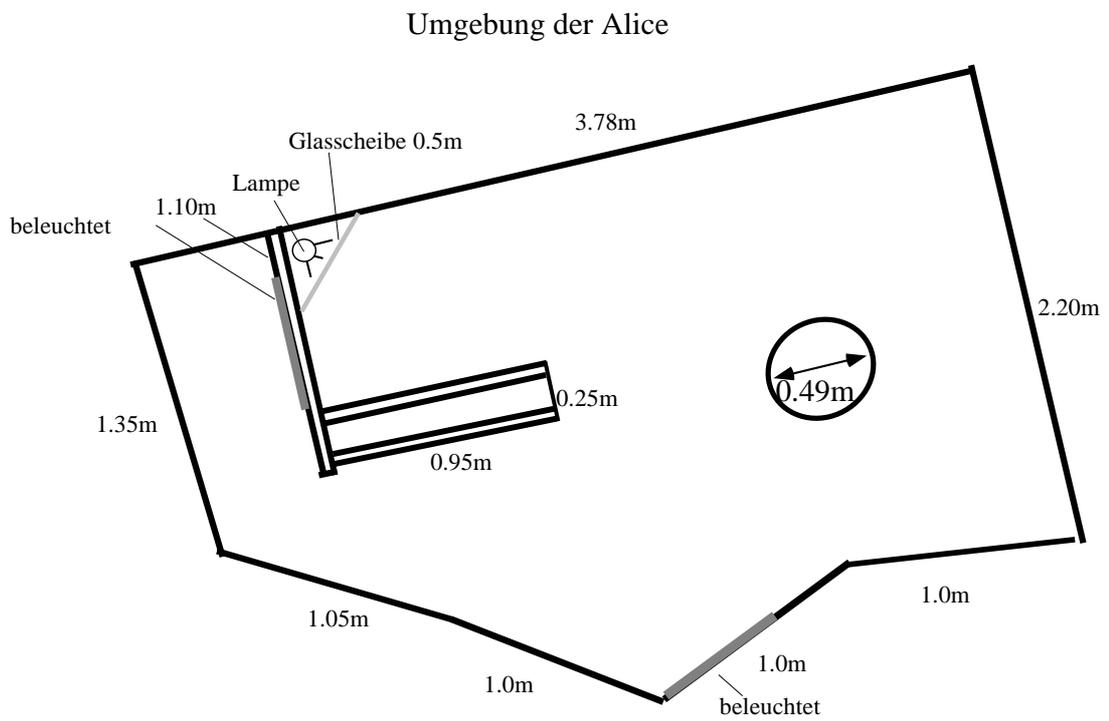
Die Korrektur ist statistisch: im Mittel wird der Fehler klein bleiben.

5.3.3.6. Ergebnisse

Die Ergebnisse hängen ab von den gewählten Kennwerten für das Netz:

Verschiebung von T_i in Richtung S:	$\epsilon_{bmu} = 0.03$
Verschiebung der Nachbarn --> S:	$\epsilon_{nachbar} = 0.003$
max. Distanz zum Einfügen:	$d_{ins-err} = 300$
max. Wert des Besuchszählers:	$ndis = 40$
Dekrement des Kantenbesuchszählers:	$vred = 1$
Minimalwert des Kantenbesuchszählers:	$vmin = 0.3$
Sensorgewichte:	
Licht	1 (auf 255 normiert)
Tasthaare:	0.7 (auf 255 normiert)
Position:	1 (in cm angegeben)

Mit den angegebenen Kennwerten wurde der AMR in einer Testumgebung losgelassen:



Nach einiger Zeit hatte ALICE dieses Netz aufgespannt, das die Umgebung recht gut approximiert. Jeder Knoten repräsentiert eine typische Sensorsituation.



Kennwerte für das Netz

$$\epsilon_{bmu} = 0.03$$

$$\epsilon_{nachbar} = 0.003$$

$$d_{ins-err} = 300$$

$$n_{dis} = 40$$

$$v_{max} = 30$$

$$v_{red} = 1$$

$$v_{min} = 0.3$$

Sensorgewichte

Licht: 1 (auf 255 normiert)

Tasthaare: 0.7 (auf 255 normiert)

Position: 1 (in cm angegeben)

