Kap. 6: Registrierung kinematischer Systeme

- Trajektorien
- Kalmanfilter
- **Partikelfilter**
  - **Modellannahmen und Lösungsansatz**
  - Globale Lokalisierung durch Partikelfilter
  - Vergleich Partikelfilter- und Kalmanfilter
- SLAM

[Corke]
[Thrun]
[Wendel]

Modellannahmen und Lösungsansatz

Modellannahmen Kalmanfilter (Wdh.)

– Sensoren haben gaussverteilte Wahrscheinlichkeitsdichtefunktion

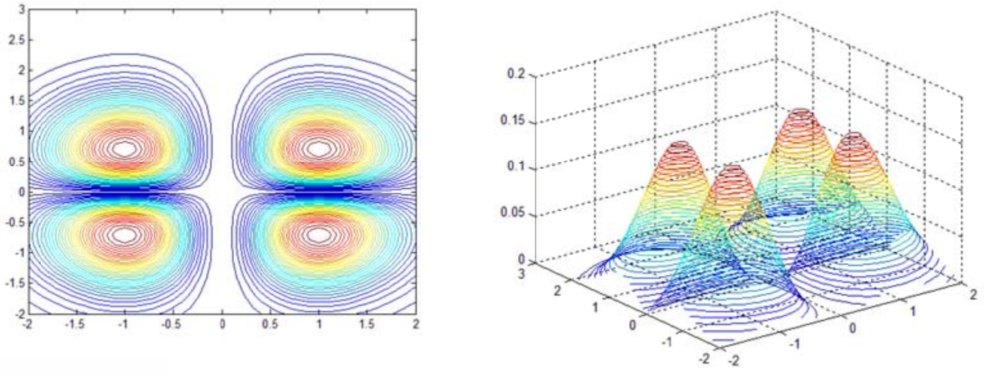
– Bewegungsmodelle und Messmodelle sind linear (bzw. werden linearisiert)

Modellannahmen Partikelfilter

– Sensoren können nicht gaussverteilte WDFs haben

– WDF kann multimodal, d.h. mit mehreren Peaks sein

– Messmodell und Bewegungsmodell können hochgradig nichtlinear sein



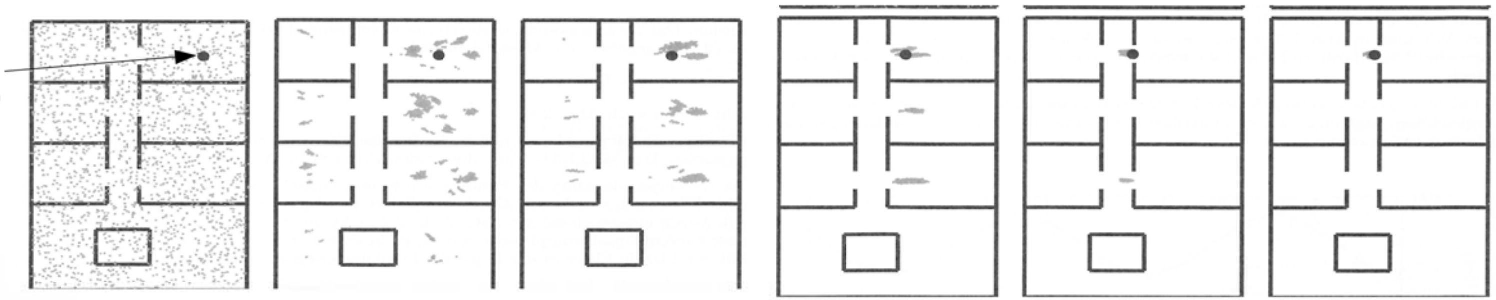
(Multimodale Wahrscheinlichkeitsdichtefunktion)

Lösungsansatz

– Speichern vieler unterschiedlicher Zustandsvektoren

– Bei neuer Messung: Bewertung aller dieser Zustandsvektoren und daraus Berechnung einer neuen Generation

Standpunkt des Roboters (unbekannt)



Aus [*Wendel*]

Begriff (Generation und Partikel)

Sei $X_k = \left(x_k^{(0)}, \ldots, x_k^{(N-1)}\right)$ eine Folge von N *posen* . Dann heißt X_k **Generation zum Zeitpunkt *k*.**

Eine pose $x_k^{(i)} = \left(u_k^{(i)}, v_k^{(i)}, \theta_k^{(i)}\right)$ heißt **Partikel** der Generation X_k

.

==> Jede Generation approximiert die Wahrscheinlichkeits-dichtefunktion (WDF) anhand des momentan verfügbaren Wissens

Globale Lokalisierung durch Partikelfilter

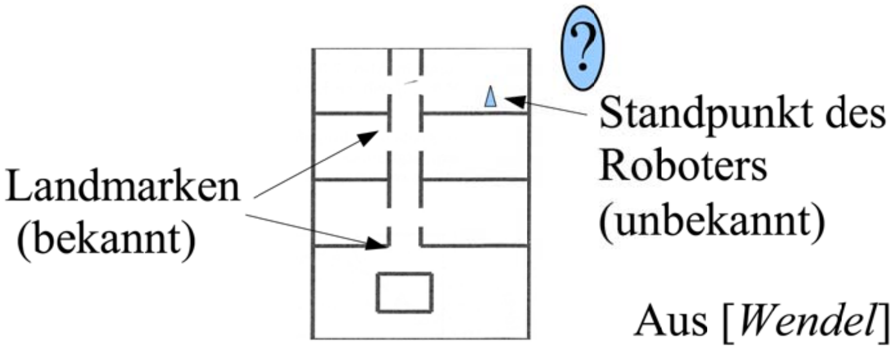
Gegeben

– Bekannte Welt (in Form von Landmarken)

– Ein mit einem „range-bearing-Sensor“ ausgestatteter Roboter wird in dieser Welt „ausgesetzt“

Gesucht

Lokalisation des des Roboters



Aus [*Wendel*]

Algorithmus

`Partikel Filter` (Übersicht)

(1) Initialisierung durch Gleichverteilung von Partikeln
(2) Bewegung des Roboters
(3) Messungen durchführen
(4) Propagation aller Partikel durch Bewegungsmodell
(5) Bewertung der Messungen für jedes Partikel
(6) Neuverteilung der Partikel (resampling)

k-te Interation

Schritt 1:

Initialisierung durch Gleichverteilung von Partikeln

wähle *N* zufällig über den Arbeitsraum ausgewählte Partikel

$$X_0 = \left(x_0^{(0)}, \ldots, x_0^{(N-1)}\right) \qquad = (x,y,\Theta)$$

Und ordne jedem ein initiales Gewicht zu:

$$W_0 = \left(w_0^{(0)}, \ldots, w_0^{(N-1)}\right)$$

mit

$$w_0^{(i)} = 1/N$$

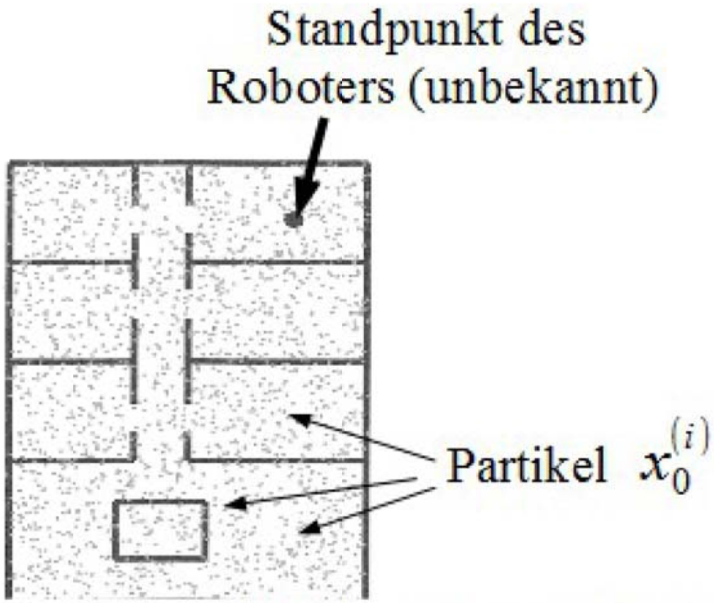


Abb. 1: Approximation der WDF zum Zeitpunkt t=0

Schritt 2: Bewegen des Roboters (zum Zeitpunkt k-1)

Der Roboter werde bewegt. Die Bewegung werde beschrieben durch

$$\delta_{k-1} := \left(\delta_{k-1}^{(d)}, \delta_{k-1}^{(\alpha)}\right)$$

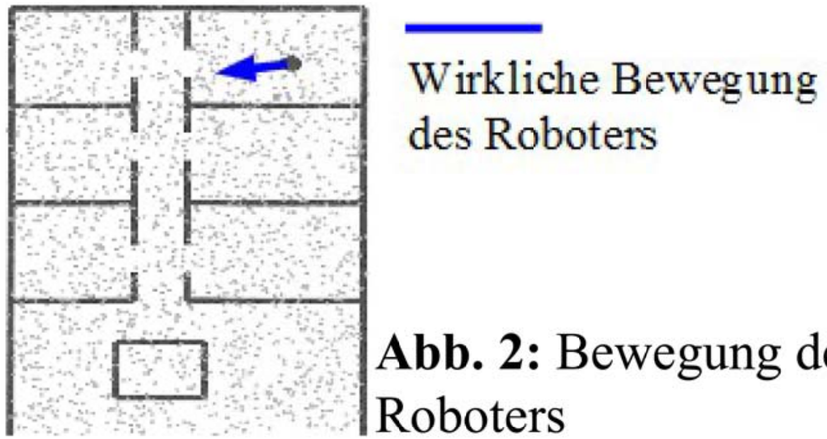
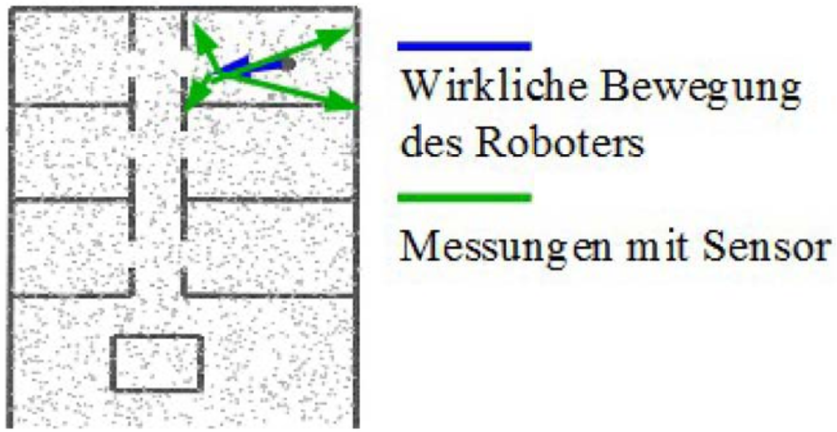


Abb. 2: Bewegung des Roboters

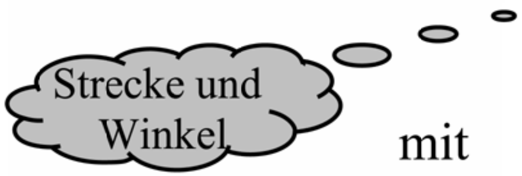
Schritt 3: Messungen durchführen (zum Zeitpunkt k)

Abb. 3: Messungen des Roboters



Der Roboter führt Messungen mit seinem „Range-Bearing-Sensor“ durch

$$Z_k = \left(z_k^{(0)}, .., z_k^{(P-1)}\right)$$



mit

$$z_k^{(p)} = \left(rg_k^{(p)}, \beta_k^{(p)}\right)$$

Schritt 4: Propagation aller Partikel durch Bewegungsmodell (zum Zeitpunkt k)

Sei f das Bewegungsmodell des Roboters und sei $\delta_{k-1} := (\delta_{k-1}^{(d)}, \delta_{k-1}^{(\alpha)})$ die Bewegungsänderung von $t=k$ nach $t=k+1$. Wende f auf alle Partikel der Generation X_{k-1} an:

$$x_k^{-(i)} = f(x_{k-1}^{(i)}, \delta_{k-1}) + q$$

q bezeichnet hierbei ein (oft gaussverteiltes) Rauschen, also:

$$q = \sim N(0, Q) \text{ mit } Q = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\beta^2 \end{pmatrix}$$

Achtung: Gerade nach der Initialisierung haben diese Partikel noch völlig unterschiedliche Orientierungen (siehe Abb. 4)

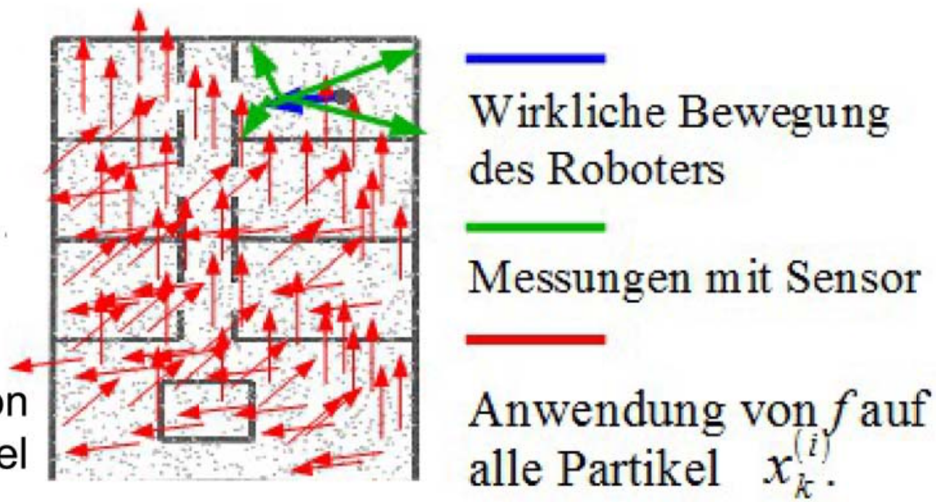


Abb. 4: Propagation der Bewegung auf jeden Partikel

Schritt 5: Bewertung der Messungen für jedes Partikel

Sei h das Messmodell des Roboters und seien $Z_k := (z_k^{(0)}, .., z_k^{(P-1)})$ die in Schritt 3 durchgeführten Messungen zum Zeitpunkt k. Bezeichne mit

$$M := (m^{(0)}, .., m^{(J-1)})$$

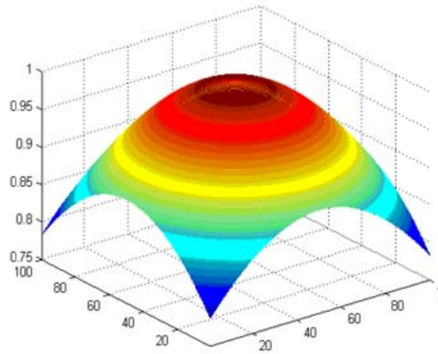
die **Koordinaten** der bekannten Landmarken. Berechne die **Innovation** zwischen jedem propagierten Partikel, jedem Messwert und jeder Landmarke durch

$$v_k^{(i,j,p)} = h(x_k^{-(i)}, m^{(j)}) - z_k^{(p)}$$

WDF m.d.E:
$v_k^{(i,j,p)} \rightarrow 0 \Rightarrow \tilde{w}_k^{(i,j,p)} \rightarrow 1$

und **gewichte** diese durch eine Funktion, z.B.

$$\tilde{w}_k^{(i,j,p)} = \exp\left(-0.5\left(v_k^{(i,j,p)}\right)^t L^{-1}\left(v_k^{(i,j,p)}\right)\right) + \tilde{w}_0$$



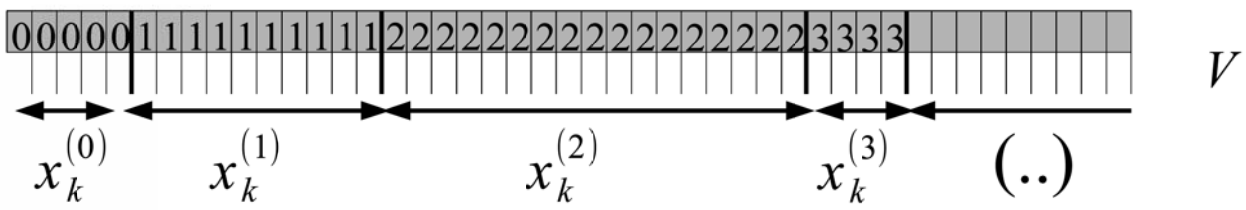
Mit einer Kovarianzmatrix L. Der Term $\tilde{w}_0 > 0$ modelliert eine verbliebene Restwahrscheinlichkeit für jeden Vektor im Zustandsraum.

Setze schließlich für jeden Partikel $\quad w_k^{(i)} = \sum_{j=0}^{J-1} \sum_{p=0}^{P-1} \tilde{w}_k^{(i,j,p)}$

Schritt 6: Neuverteilung der Partikel (resampling)

Idee: Je höher die Gewichtung eines Partikels, desto häufiger soll er in der nächsten Generation wieder vorkommen.

Vervielfältige Partikel je nach Gewicht und schreibe sie in einen Vektor V der Länge $\sum_{j=0}^{N-1} w_k^{(j)}$



Und berechne daraus die neue Generation X_{k+1}

For i=0 : N-1 do

- Generiere Zufallszahl $r \in [0, |V|-1]$
- Setze $x_{k+1}^{(i)} := x_k^{(r)}$ und $w_{k+1}^{(i)} := 1/N$

end