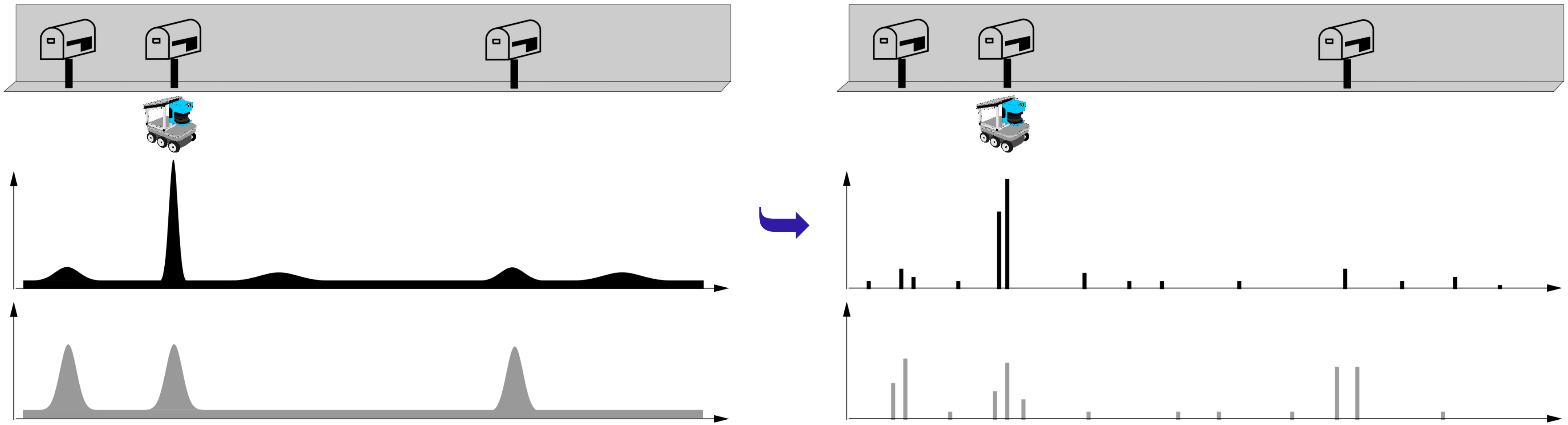
5.3.6 Monte-Carlo-Lokalisierung

Monte-Carlo-Algorithmen approximieren aufwändige exakte Lösungen durch effizientes Berechnen von (relativ wenigen) Stichproben



MCL-Algorithmus zur Roboterlokalisierung: **Partikelfilter**

Algorithmus MCL (Buch Algo. 5.2)

Eingabe: Partikelmenge $\mathcal{X}_t = \{\langle \boldsymbol{x}_t^i, w_t^i \rangle\}_{i=1,\ldots,N}$, Aktion $\boldsymbol{u}_{t+1}$, Messung $\boldsymbol{z}_{t+1}$.

Gewicht des Partikels

Ausgabe: Neue Menge von Partikeln $\mathcal{X}_{t+1}$

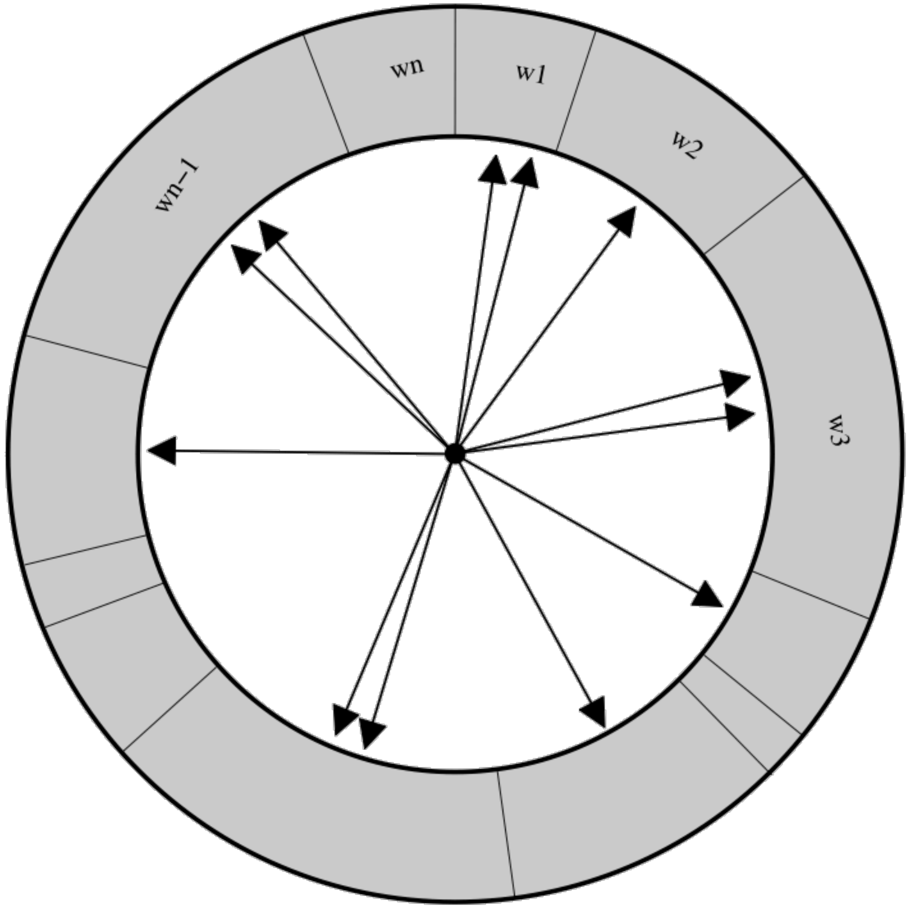
Wie zieht man gemäß Verteilung?

$\mathcal{X}_{t+1} = \emptyset$
$\eta = 0$
for $i = 1 \ldots N$ **do**
  Ziehe ein j entsprechend der Verteilung über w_t *// $Bel(\boldsymbol{x}_t)$*
  Ziehe ein $\boldsymbol{x}_{t+1}^i$ entsprechend $P(\boldsymbol{x}_{t+1} \mid \boldsymbol{u}_{t+1}, \boldsymbol{x}_t^j)$ *// Transition*
  $w_{t+1}^i = P(\boldsymbol{z}_{t+1} \mid \boldsymbol{x}_{t+1}^i)$ *// Sensormodell*
  $\eta = \eta + w_{t+1}^i$ *// Update des Normierungsfaktors*
  $\mathcal{X}_{t+1} = \mathcal{X}_{t+1} \cup \{\langle \boldsymbol{x}_{t+1}^i, w_{t+1}^i \rangle\}$
end for
for $i = 1 \ldots N$ **do** *// Normierung der Gewichte in $\mathcal{X}_{t+1}$*
  $\boldsymbol{w}_{t+1}^i = \eta^{-1} w_{t+1}^i$
end for
return $\mathcal{X}_{t+1}$

Wie definiere W'keit konkreter Messung?

Ziehen gemäß Verteilung

… hatten wir abstrakt in Folie 157f. Hier konkret für Partikel-Verteilung, durch Gewichte w_i gegeben, z.B. **Rouletterad-Auswahl**



- Voraussetzung: Ziehen nach Gleichverteilung (approx.: `random(0,`Σw_i`)`)
- Für gezogene Zahl r ist dasjenige Partikel n gezogen, für das gilt:

$$\sum_{i=1}^{n-1} w_i < r \leq \sum_{i=1}^{n} w_i$$

Vergleich realer vs. erwarteter Scan

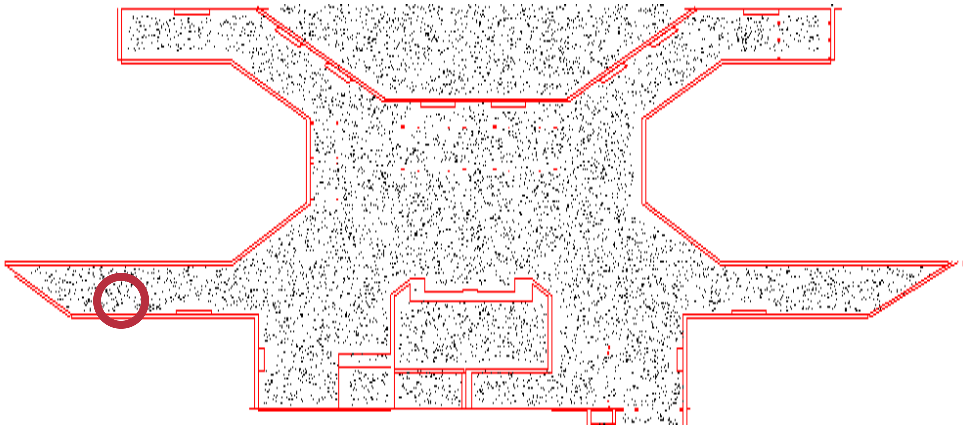
… realisiert/approximiert Zeile 6 MCL: $w^i_{t+1} = P(z_{t+1} \,|\, x^i_{t+1})$

Definiere $\xi(\mathbf{X},\mathbf{Z})$:
Maß für Übereinstimmung von realem Scan $\mathbf{Z}$ und Scan, der in Zustand (Partikel) $\mathbf{X}$ gemäß Rasterkarte erwartet wäre:

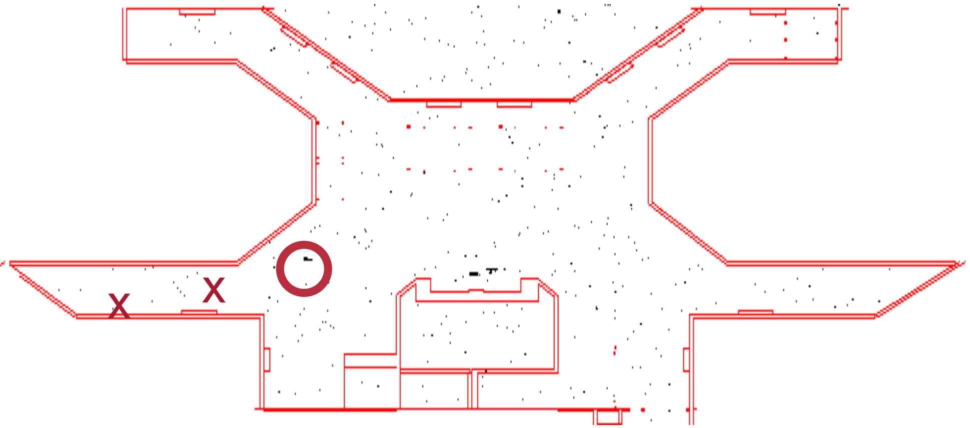
- „Simuliere“ in Rasterkarte Scan von Pose von $\mathbf{X}$ (Berechne einzelne Messwerte in den diskreten Scanwinkeln des realen Scanners)
- Ermittle Kreuzkorrelation (Folie 187, Winkelhistogramme) zwischen realem und simuliertem Scan; oder …
- … alternativ: Summiere Messwertdifferenzen, dabei wichte klein reale Messwerte, die deutlich „zu kurz“ sind (nicht kartierte Möbel, Personen, …), wichte groß „zu lange“ reale Messwerte

Beispiel MCL

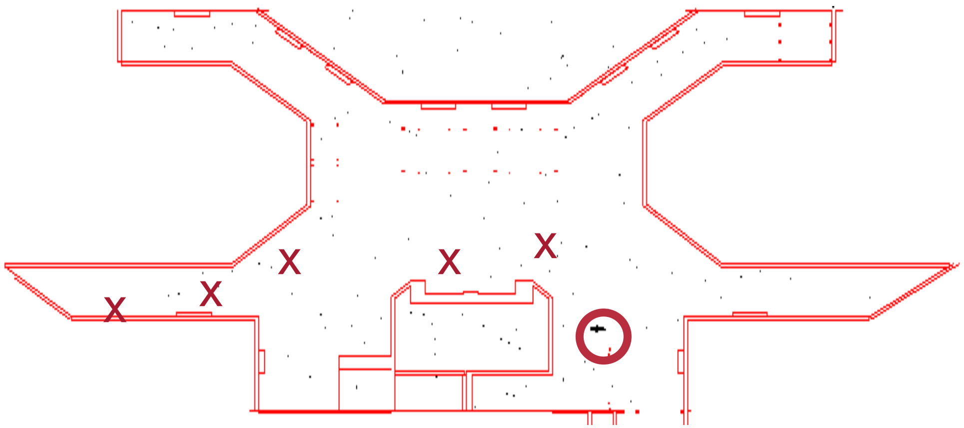
Aktuelle Position markiert mit Kreis; Wegpunkte markiert mit x



Start: Gleichverteilung im Freiraum



Dritte Messung: wenige Maxima



Sechste Messung: Ein Maximum

„Absterbende" Partikel (Untergewicht, unmögliche Pose in der Wand, …) ersetze:

- zufällig in Freiraum und/oder
- gewichtet nach aktueller Verteilung

Meint der Roboter, dass er weiß, wo er ist?

Erinnerung: **Entropie** einer Verteilung $\mathbf{P}(X)$ über Var.n X:

$$H(X) := -\sum_{i=1}^{|X|} P(x_i) \log_2 P(x_i)$$

… ist maximal bei Gleichverteilung, nämlich

$$H_{\max}(X) = -\sum_{i=1}^{|X|} \tfrac{1}{|X|} \log_2 \tfrac{1}{|X|} = -|X| \cdot \tfrac{1}{|X|} \left(\log_2 1 - \log_2 |X| \right) = \log_2 |X|$$

(Lokalisation: Posen gleichverteilt, Roboter hat keine Idee, wo er ist)

… ist minimal bei sicherer Information, nämlich 0 (wegen $\log(1)=0$

$$H(\mathcal{X}) := -\sum_{\mathbf{x} \in \mathcal{X}} Bel(\mathbf{x}) \frac{\log_2 Bel(\mathbf{x})}{\log_2 |\mathcal{X}|} = -\sum_{\mathbf{x} \in \mathcal{X}} Bel(\mathbf{x}) \log_{|\mathcal{X}|} Bel(\mathbf{x})$$

…normiert auf $[0,1]$ für Ü-Zustand bezüglich Partikelmenge $\mathcal{X}$

Unsicherheitsmaß d. aktuellen Überzeugungszust. $Bel(\mathcal{X})$

($H(\mathcal{X})=0$ ➥ genaue Kenntnis der Pose; $H(\mathcal{X})=1$ ➥ völlige Unkenntnis)

Wieviele Stichproben braucht man?

Kommt drauf an! …

Mehr Info zu Partikelfiltern:
Dieter Fox, U. Washington
www.cs.washington.edu/ai/Mobile_Robotics/mcl/

… wie groß relevante Posemenge:
klein $\hookrightarrow$ N klein; groß $\hookrightarrow$ N groß

… wie häufig Landmarken in der Umgebung:
viele $\hookrightarrow$ N klein; wenige $\hookrightarrow$ N groß (s. nächste Folie!)

… wie gut Pose aktuell bekannt (<u>variabel</u>!):
$H(\mathcal{X})$ nahe 0 $\hookrightarrow$ N klein; $H(\mathcal{X})$ nahe 1 $\hookrightarrow$ N groß

- Für Gebäudenavigation reicht N«10.000
- Für kleine Areale (RoboCup!) reicht N«1.000
- **KLD-Sampling** adaptiert N

“Face it, Fred—you’re lost!”

Fazit Lokalisierung in Karten

- Entsprechend Varianten von Karten und Varianten von Sensoren gibt es viele Varianten von Lokalisierung in Karten
- Markow-Lokalisierung (MCL) dominiert derzeit Literatur
  - Mathematisch gut verstanden; effizient implementierbar
  - Integration mit anderer probabilistischer Information
  - Integration mit Kartenbau (SLAM, s.u.)
- Fast alle Karten/Lokalisierung sind für 2D-Roboterbewegung (x,z,θ)
- Verwenden fast nur Geometrie, keine „Semantik“
- Klappen nur bei relativ langsamer Fahrt (Ausnahme HAYAI)

⤷ Es gibt noch viel zu entdecken!