

Konzeption und Softwareimplementierung des Laserscanner-basierten Indoornavigationssystems für einen Fahrroboter

Aufgabe dieser Thesis war die Implementierung des Partikelfilters auf dem Volksbot, um diesen in Gebäuden lokalisieren zu können. Auf dem Volksbot ist ein Laserscanner installiert welcher Entfernungsmessungen z liefert. Zusätzlich stehen die Odometriedaten u und eine Karte m des Gebäudes zur Verfügung.

Aus diesen Daten wird der *belief* $bel(x_t)$ berechnet, die Schätzung des Zustandsraums x . Dieser beinhaltet die Pose des Roboters (x, y, θ) .

```
Eingabe: Partikelset  $X_{t-1}$ , repräsentiert den alten belief  $Bel(x_{t-1})$ ; Steuerdaten  $u_t$ ; Messungen  $z_t$ 
1  $X_t := 0$ ;
2  $\hat{x}_t := 0$ ;
3  $sum := 0$  // Summe der Gewichte
4 for  $m:=1$  bis  $M$  do
5   // Vorhersageschritt:
6    $x_t^m := \text{Bewegungsmodell}(x_{t-1}^m, u_t)$ ;
7   // Korrekturschritt:
8    $w_t^m := \text{Beobachtungsmodell}(z_t, x_t^m)$ ;
9    $sum := sum + w_t^m$ ;
10 end
11 // Gewichte normieren:
12 for  $m:=1$  bis  $M$  do
13    $w_t^m := w_t^m / sum$ ;
14 end
15 // Zustandsvektor schätzen:
16 for  $m:=1$  bis  $M$  do
17    $\hat{x}_t := \hat{x}_t + x_t^m * w_t^m$ ;
18 end
19 // Resampling:
20  $\text{Resample}(X_t)$ ;
Ausgabe: Partikelset  $X_t$ , welches den neuen belief  $Bel(x_t)$  repräsentiert;
geschätzter Zustandsvektor  $\hat{x}_t$ 
```

Abb. 1: Der Partikelfilter Algorithmus

Der Partikelfilter basiert auf dem Bayes Filter, welcher den $bel(x_t)$ rekursiv aus dem vorhergehenden $bel(x_{t-1})$ berechnet. Er nähert den $bel(x_t)$ durch eine Anzahl M von Stichproben, den Partikeln, an.

Zu Beginn werden die Partikel gemäß der Gleichverteilung über die ganze Karte initialisiert. Im Vorhersageschritt wird anhand der Odometriedaten und der vorhergehenden Pose die neue Pose der Partikel berechnet.

Hochschule Karlsruhe – Technik und Wirtschaft

Fakultät IMM • Studiengang Geodäsie und Navigation (B.Sc.)

www.hs-karlsruhe.de

Bearbeiter: Pascal Kaiser

E-Mail-Adresse: kapa1012@hs-karlsruhe.de

Referent: Prof. Dr.-Ing. Reiner Jäger

Korreferent: Prof. Dr.-Ing. Tilman Müller

Danach wird im Korrekturschritt die Wahrscheinlichkeit bestimmt, mit der die aktuelle Messung an dieser Pose aufgenommen wurde. Anschließend erfolgt das Resampling. Dabei werden die Partikel proportional zu ihren Gewichten neu gezogen, die Gewichte werden anschließend zurückgesetzt. Danach sind die Partikel gemäß dem $bel(x_t)$ verteilt.

Der Algorithmus ist in C++ geschrieben. Er läuft unter dem Robotic Operation System (ROS).

Die Tests des Algorithmus fanden im B-Bau statt. Während mehrerer Testläufe wurde die optimale Konfiguration des Programms ermittelt. Es mussten unter anderem die Standardabweichungen der Odometrie und der Entfernungsmessungen angepasst werden. Sowohl das Bewegungsmodell als auch das Beobachtungsmodell wurden modifiziert.

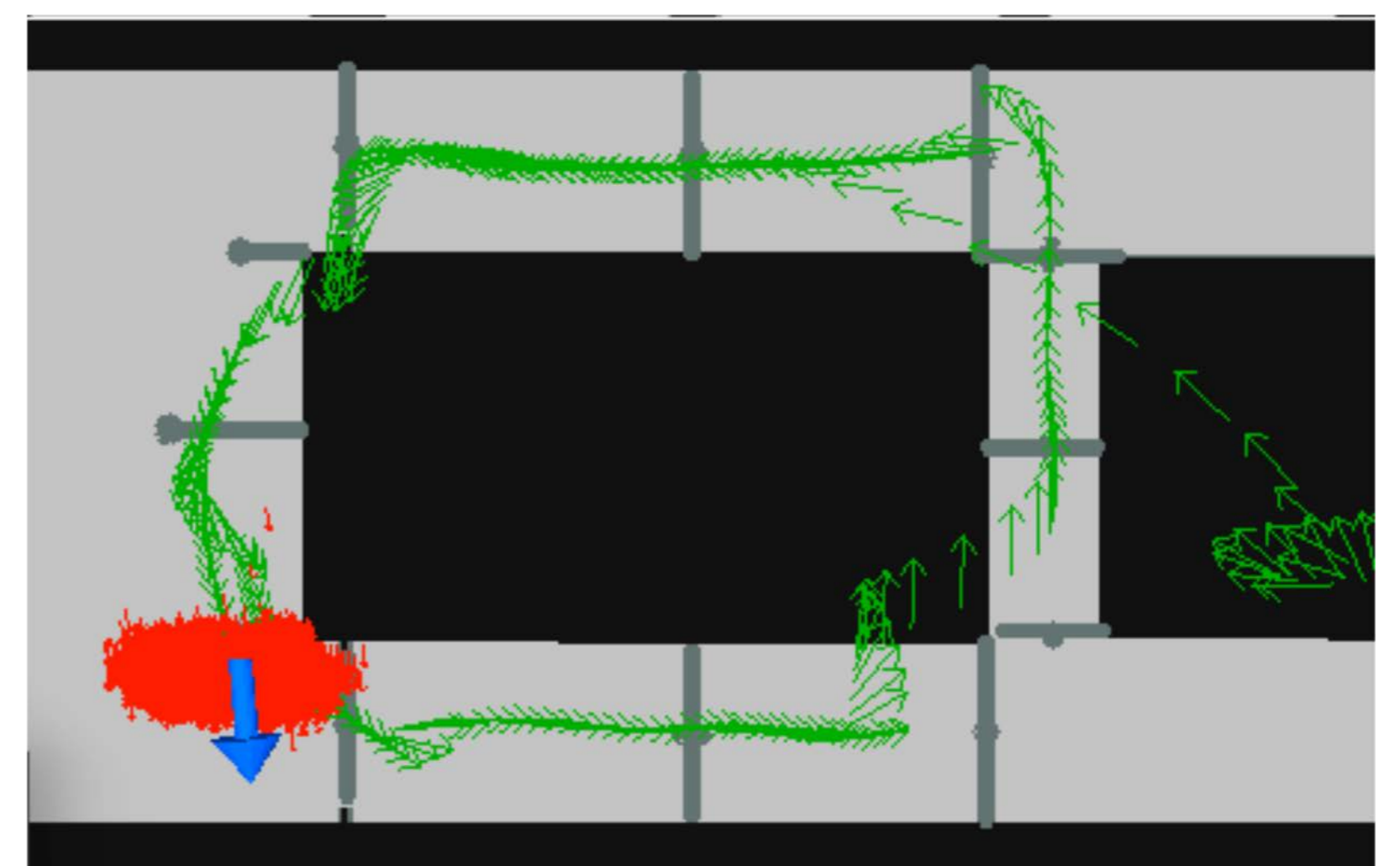


Abb. 2: Der Verlauf der geschätzten Pose und die aktuelle Partikelwolke

Am Ende erfolgte die Bestimmung der Genauigkeit indem mehrere Messpunkte angefahren und die Differenz zwischen geschätzter und wahrer Pose ermittelt wird. Dabei wurde eine durchschnittliche Abweichung von 1,7m in der Strecke und 7° in der Orientierung erzielt. Insbesondere die Umgebung hat einen großen Einfluss auf das Ergebnis.