



FAKULTÄT FÜR
INFORMATIK

Umsetzung eines Online-SLAM-Verfahrens auf der Roboterplattform VolksBot-Lab

Bachelorarbeit

Frank Engelhardt
Institut für Verteilte Systeme

Gliederung

- Motivation
- Problemexposition
- Lösungsverfahren
- Zeitverhalten
- Ergebnis

Motivation

SLAM – Simultaneous Localization and Mapping

- Mobiler Roboter operiert in unbekannter Umgebung
- Ziele:
 - Kartierung der Umgebung (zweidimensional)
 - Bestimmung der eigenen Position

Online–Umsetzung

- D.h. Ausgaben bereits während der Fahrt erzeugen

Verwendbar z.B. für Pfadplanung oder autonome Exploration

Rahmenbedingungen

Verfügbare Hardware

- Differentialantrieb
- Odometriesensoren
- Laserscanner
- Kein GPS o.Ä.
- Geringe Rechenleistung
(533MHz CPU, 256 MB RAM)

Umgebung

- Indoor
- Eben
(keine Höhenunterschiede)

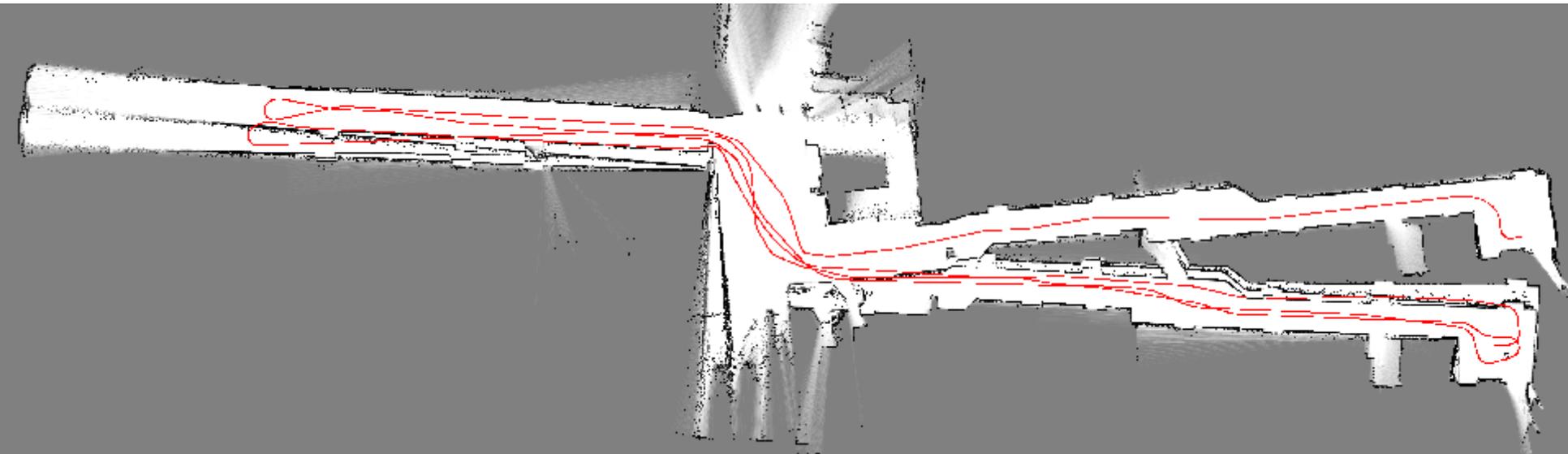


VolksBot-Lab [1]

Problemexposition I (SLAM-Problem)

Inkrementelle Positionsbestimmung

- Positionsbestimmung allein durch Odometrie führt zu Mehrdeutigkeiten in der Karte
- Problem: Fehleraufsummierung



Problemexposition II (Echtzeitproblem)

Größe der erkundeten Umgebung hat Einfluss auf die Laufzeit

- Je weiter die Erkundung fortschreitet, desto höher ist (im Allgemeinen) die Laufzeit von SLAM-Algorithmen

Der Roboter ist allerdings ständig in Bewegung

- Neue Sensordaten treffen regelmäßig ein
- Bei steigendem Rechenaufwand müssen ggf. Sensordaten verworfen werden
- Konsequenz: Verschlechterung der Ergebnisqualität (Position und Karte) im Laufe der Zeit

Echtzeitverhalten erforderlich

- Mindestqualität der Ergebnisse erfordert Einschränkung der Laufzeit

Bekannte Lösungsverfahren für das SLAM-Problem

Verschiedene Algorithmen (Auswahl)

SEIF-SLAM [2]	+ konstante Laufzeit - Rechenaufwand - Umsetzung kompliziert
Graph-SLAM [3]	- Nicht onlinefähig
FastSLAM [4]	- Zu hoher Speicherverbrauch - Laufzeit logarithmisch in der Kartengröße
TinySLAM [5]	+ Sehr effizient - unbestimmte Laufzeit

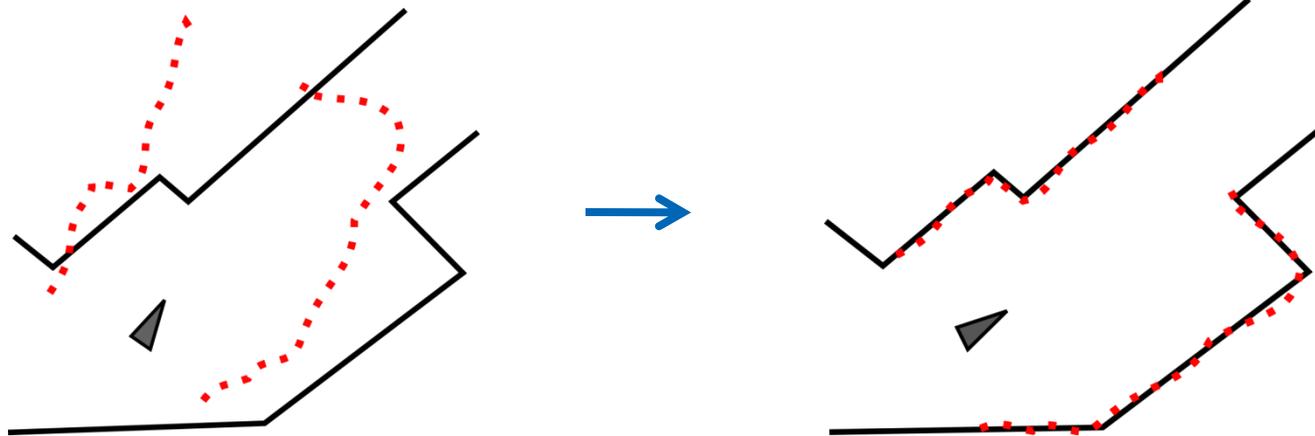
Wahl fiel auf TinySLAM

- Alle anderen Algorithmen zu Rechenaufwändig
- Echtzeitfähigkeit muss noch sichergestellt werden

TinySLAM: Funktionsweise

Lokalisierung

- Zunächst grobe Lokalisierung per Odometrie
- Korrektur der Position durch Vergleich der Scannerdaten mit den vorhandenen Kartendaten



Suche nach Monte-Carlo-Prinzip

- Lokalisierungsdauer ist variabel (Anytime-Algorithmus)
- Abbruchkriterium für Original-TinySLAM: Lokalisierungsqualität
-> Laufzeit kann nicht abgeschätzt werden

Zeitverhalten

Lokalisierungsdauer ist bei TinySLAM der zeitliche Flaschenhals

- Alle anderen Operationen haben konstante Laufzeit
- Begrenzung der Lokalisierungsdauer löst Echtzeitproblem

Antwortzeit ist nicht fest vorgegeben

- Lokalisierungsdauer kann so gewählt werden, dass möglichst hohe Lokalisierungsqualität erreicht wird
- Lokalisierungsqualität ist entscheidender Faktor für SLAM

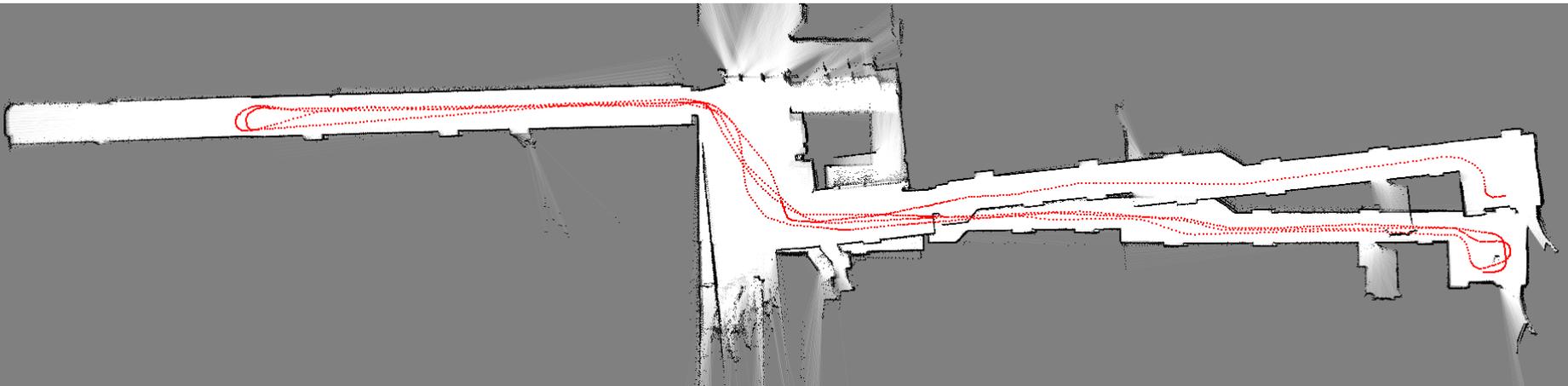
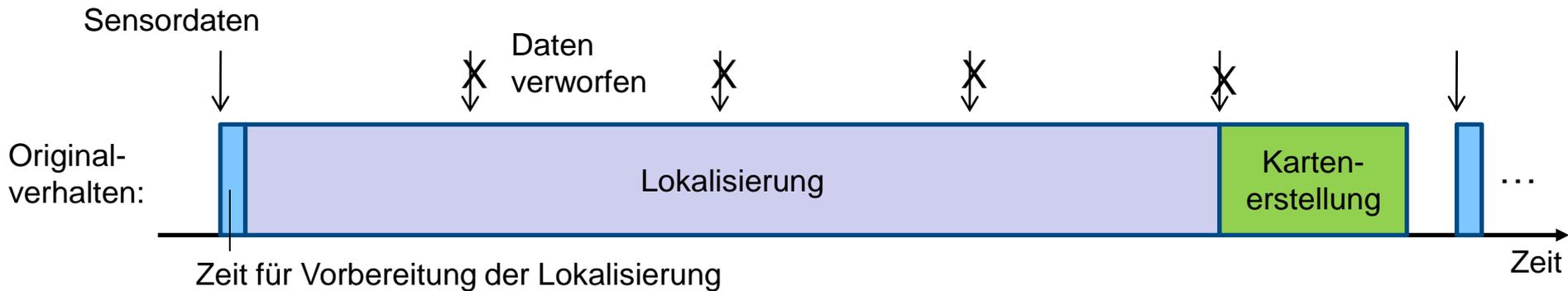
Optimierung der Lokalisierungsdauer

- Durch Offline-Simulationen mit aufgezeichneten Daten
- Ergebnis: etwa 50 Durchläufe der Monte-Carlo-Suche erforderlich

Zeitverhalten

Problem: Suchzeit auf dem Zielsystem zu lang

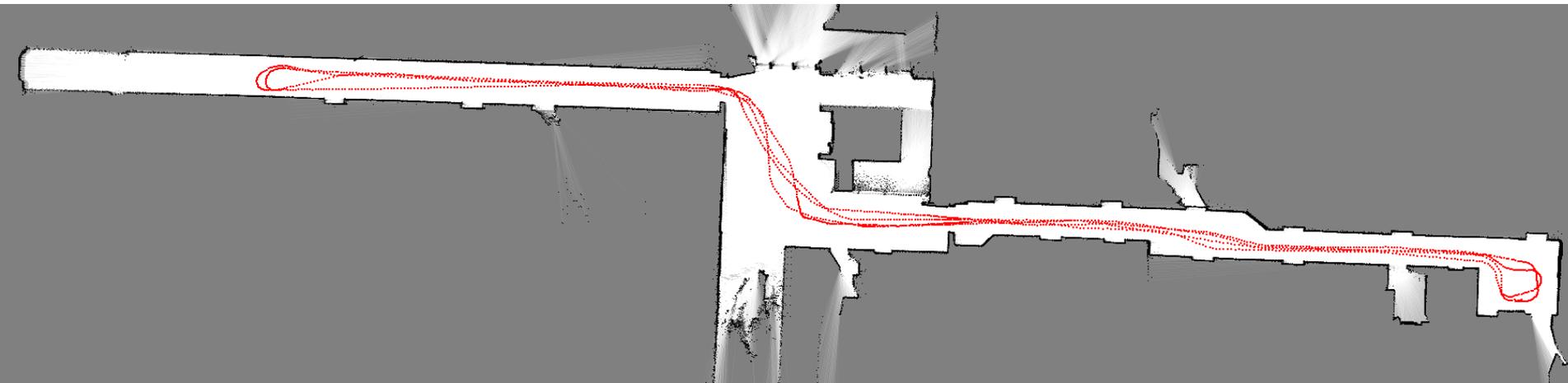
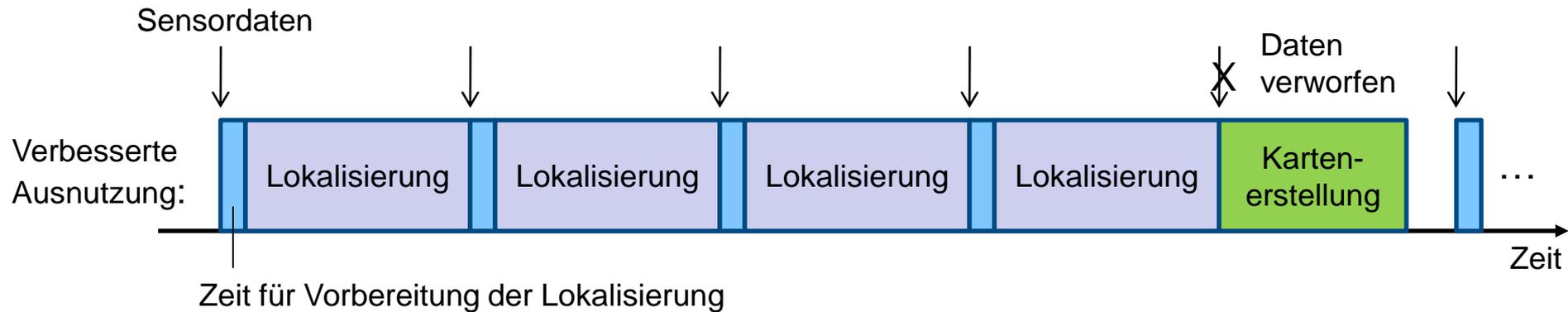
- Lokalisierung würde etwa 200ms dauern
- In der Zeit würden etwa 10 Laserscans verlorengehen



Zeitverhalten

Lösung: Unterbrechung der Lokalisierung

- ‚Neustart‘ der Lokalisierung bei jedem Eintreffen von Sensordaten
- Erhöht Informationsausbeute



Ergebnis

Durch Unterbrechung der Lokalisierung konnte gewünschte Performanz erreicht werden

- Kartenerstellung und Lokalisierung erfolgen weitestgehend konsistent, d.h. ohne Mehrdeutigkeiten

Onlinefähigkeit durch Echtzeitverarbeitung erreicht

- Konstante Laufzeit des Verfahrens
- Zielsystem erlaubt nur weiche Echtzeit
 - Antwortzeit für die Positionsbestimmung ca. 80 ms

Quellen

- [1] www.volksbot-lab.de
- [2] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hough Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7–8):693–716, 2004.
- [3] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 9 2005.
- [4] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2002.
- [5] Bruno Steux and Oussama El Hamzaoui. tinySLAM : a SLAM Algorithm in less than 200 lines of C code. *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2010.
- Demovideo: <http://euk.cs.ovgu.de/de/online-slam-volksbot-lab>