

Institut für Roboterforschung Abteilung Informationstechnik





Modellierung

Lokalisierung (Part 1) und verteilte Modellierung der dynamischen Umgebungselemente (Part 2)

Stefan Czarnetzki





Part 1 – Multiple-Hypotheses Kalman Filter

- Anforderungen an Lokalisierung im RoboCup
 - Schnelle Relokalisation, multimodale Aufenthaltswahrscheinlichkeit
 →Partikelfilter
 - Glatte, wenig schwankende/verrauschte Positionsverfolgung
 →Kalman-Filter
- Ansatz: Multiple-Hypotheses Kalman Filter (aka Gaussian Mixture, Sum of Gaussians, usw.)
 - Theorie ist "nett", aber wachsende Komplexität mit Anzahl Terme
 - Trade-off: Laufzeiteffizienz vs. "was Gaussian Mixture Filtern eigentlich toll macht":-/
- Alternative Umsetzung eines Multiple-Hypotheses Kalman Filter
 - Alles super! ;-)





- Bayes Filter (bla bla, kennt ihr alles)
- Belief ist gewichtete Summe von Gaussians
 - $bel(x_t) = \sum \alpha_i N(\mu_i, P_i)$
 - Gewichtung α_i entspricht grob der Likelyhood des Means μ_i
- Mehrdeutige Situation → Aufsplitten der Gaussians
 - Diskrete Mehrdeutigkeit: Beobachtungs-Korrespondenzproblem
 - Hohe Bewegungsunsicherheit ≈ "Kontinuierliche Mehrdeutigkeit"
 - Gute Behandlung der Nicht-Linearitäten möglich (theoretisch)



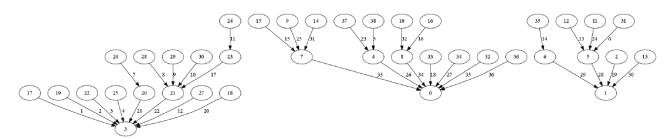


- Problem: Exponentielles Wachstum der Terme
- Klassische Lösung: Pruning
 - Theoretisch korrekt: Berechnung neuer Summe mit gewünschter (geringer) Anzahl Terme, die die alte Summe optimal approximiert
 - Praktisch: Einfaches Verschmelzen und Löschen

Michael J. Quinlan and Richard H. Middleton



(a) Split when models are uncertain

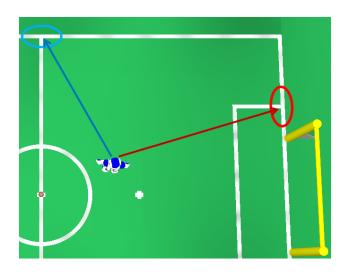


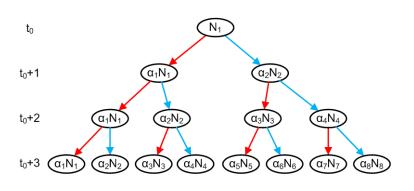
(b) Merge after uncertain split





- Problem: Exponentielles Wachstum der Terme
- Klassische Lösung: Pruning
 - Theoretisch korrekt: Berechnung neuer Summe mit gewünschter (geringer) Anzahl Terme, die die alte Summe optimal approximiert
 - Praktisch: Einfaches Verschmelzen und Löschen









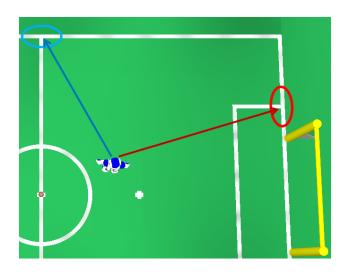
- Gaussian Mixture Performance Dilemma
 - Temporär unwahrscheinliche Bereiche also doch nicht abgedeckt
 - Cluster bleiben unimodal durch aggressives Pruning
- Warum dann überhaupt noch GMs???

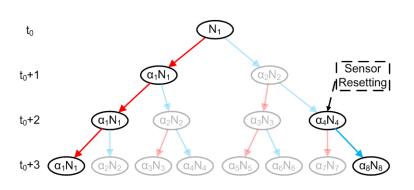




Alternativ: Maximum Likelyhood-Choices

- Ausschließlich Updates mit Maximum Likelyhood Entscheidungen
 - Jeder Term verhält sich wie Unimodaler Kalman Filter
- Wichtig: Max. Likelyhood kann auch "False Positive" sein!
 - Robust gegen systematische "False Positives" im Publikum
- Weiterhin Pruning durch einfaches Verschmelzen und Löschen



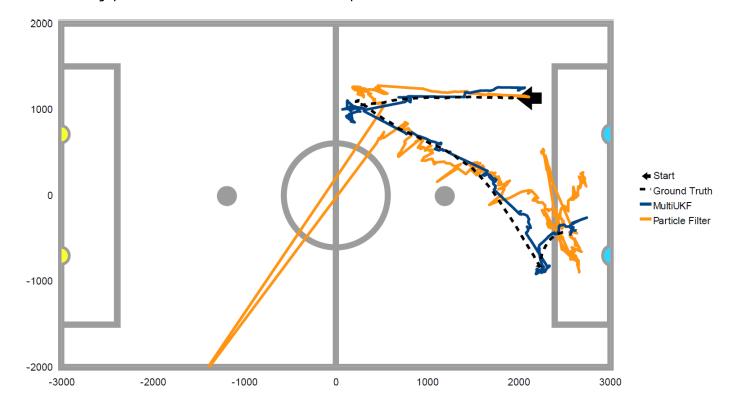






Evaluation

- Realer Roboter, Ground-Truth aus Deckenkamera
- Multi-Hypotheses UKF läuft parallel zu altem Partikelfilter

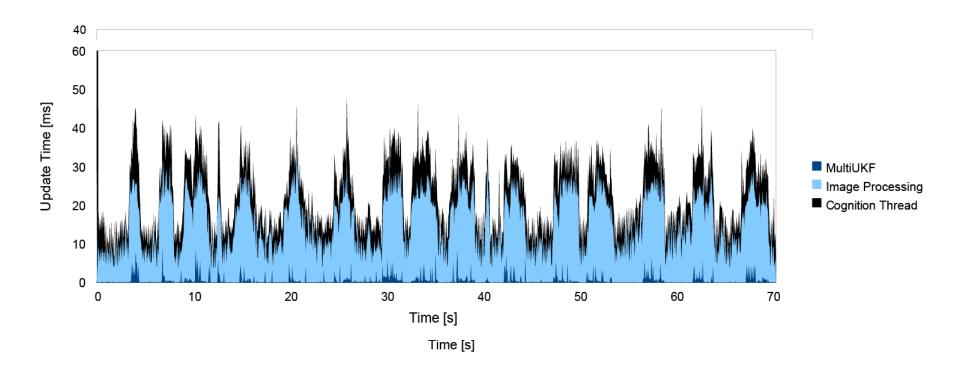






Evaluation

Bessere Lokalisierungsqualität bei weit niedrigerer Laufzeit







Part 1 – Multiple-Hypotheses Kalman Filter

Fragen soweit? :-)







Part 2 – Kooperative Weltmodellierung

- "Wissen, wo alles ist, auch wenn man selbst grad nicht hinguckt."
 - Eigene Wahrnehmung oft unsicher: Ball noch ok, Roboter sehr unzuverlässig
 - Viele potentielle Verdeckungen bei 4 vs 4
- Weltmodellierung als SLAM-Problem
- Lösung 1: Basierend auf FastSLAM, nur schneller und verteilt (implementiert, funktioniert, nicht ganz schnell genug)
- Lösung 2: Basierend auf MultiUKF-Lokalisierung und EKF-SLAM (noch nicht komplett implementiert, "sollte" klappen)
- Lösung 3: Basierend auf MultiUKF-Lokalisierung, ohne "S" in SLAM (wird seit den GermanOpen2011 benutzt)





Weltmodellierung als SLAM-Problem

- In Literatur: Entweder "nur" Lokalisation, oder "komplettes" SLAM
- Realistische Szenarien im Allgemeinen: "Irgendwo dazwischen"
 - A-priori Information oft vorhanden (Karten, Luftbilder, Grundrisse, ...)
 - Nie vollständiges Wissen: Elemente zu Mappen/Tracken
 - Messungen und Modelle mit sehr unterschiedlichen Unsicherheiten
 - Vorher bekannt ⇔ vorher unbekannt
 - Eindeutig ⇔ mehrdeutig
 - Statisch ⇔ dynamisch
 - Lokal ⇔ verteilt
- Kontext SPL: Tore, Linienkreuzungen, Mittelkreis, Ball, Roboter, ...





Weltmodellierung als SLAM-Problem

- Warum so kompliziert?!?
 - Lokalisierungsproblem? Gelöst.
 - Tracking üblicherweise einfach in lokalen Koordinaten
 - Ist effizienter
 - Ist "auserforscht": Stationärer Beobachter trackt dynamisches Objekt
- Modellierung des vollständigen Zustandes hat Vorteile
 - Dynamische Elemente potentiell nützlich für Lokalisation
 - Geteilte Information kann mehrdeutige Lokalisierung auflösen
 - Odometriefehler ist in Positionsschätzung schon kompensiert und summiert sich nicht auf





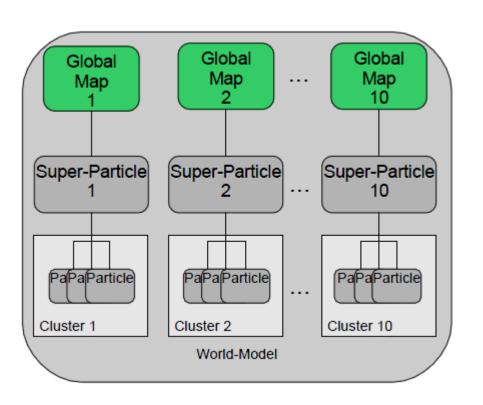
Lösung 1: Basierend auf FastSLAM

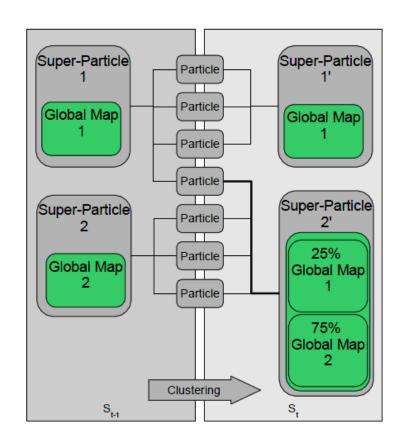
- FastSLAM: Partikelfilter für Lokalisierungsschätzung
 - Torpfosten, Linienkreuzungen, Mittelkreis, usw. brauchen nicht gemappt werden, da im Vorhinein bekannt und unveränderlich
 - Je Partikel eigene Karte mit voneinander unabhängigen Features
- Beobachtung: Funktioniert, ist aber zu langsam und nutzt die Unterschiede der Unsicherheiten in Beobachtungen und Modellen nicht aus
 - Modellunsicherheit überwiegt lokale Lokalisierungsunsicherheit
 - Viele Partikel mit selber Karte → Ineffizient!





Lösung 1: Anpassung "Super-Partikel"









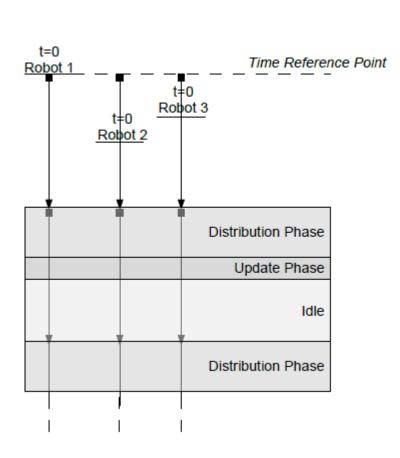
Lösung 1: Anpassung "Percept-Buffer"

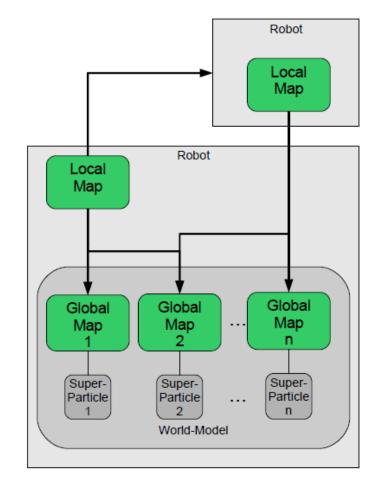
- Akkumulieren von Beobachtungen zur Unsicherheits-Reduktion
 - Lokaler, roboter-zentrischer Percept-Buffer (Kalman Filter)
 - Nur ein Update pro Zeitschritt statt Update pro Partikel
 - Zeitliche Begrenzung (500ms) verhindert Integration von Odometriefehlern
 - Ergebnis wird genutzt, um periodisch die Partikel zu updaten





Lösung 1: Multi-Robot Sensor Fusion

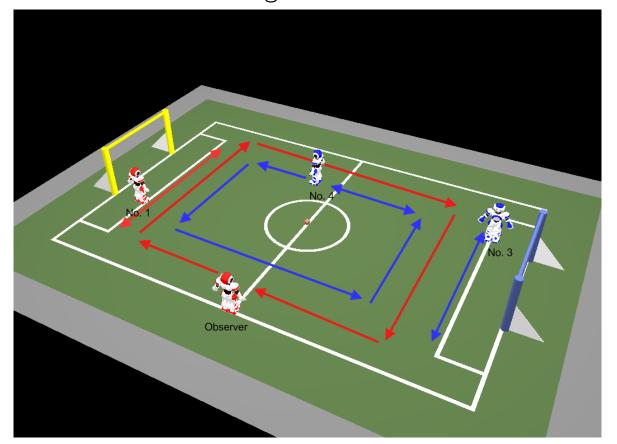








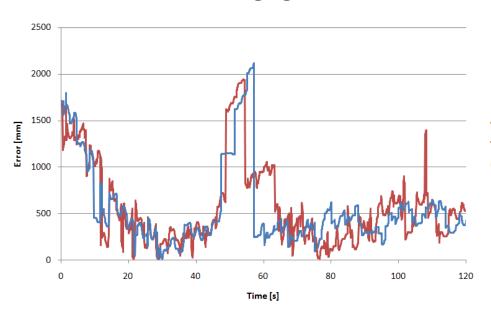
Odometrie und Wahrnehmungsfehler wie beim realen Roboter

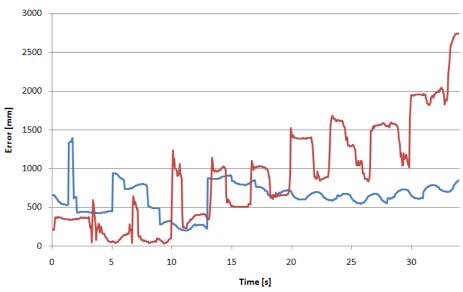






 Vorteil von Modellierung des vollen Zustands gegenüber separater Modellierung





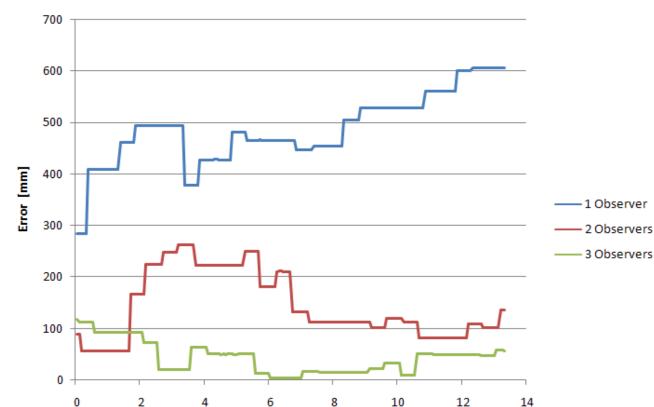
Frequently observed robot

Infrequently observed robot





Genauigkeitsgewinn durch Multi-Robot Sensor Fusion



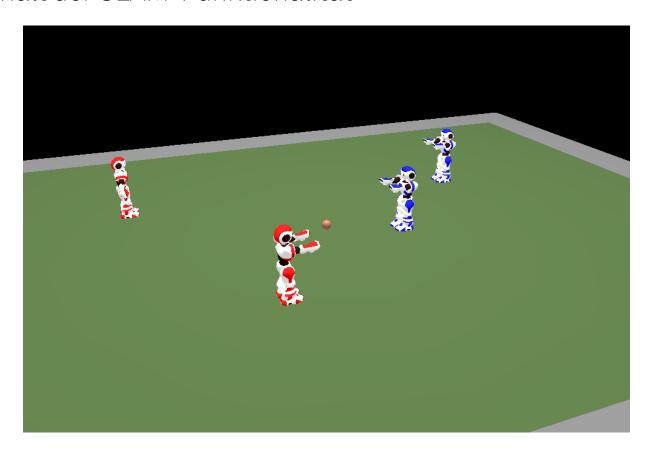
Time [s]

- Vollständigkeit
 - Missing:1.2 → 0.3
 - False positive:
 0.16 → 0.25





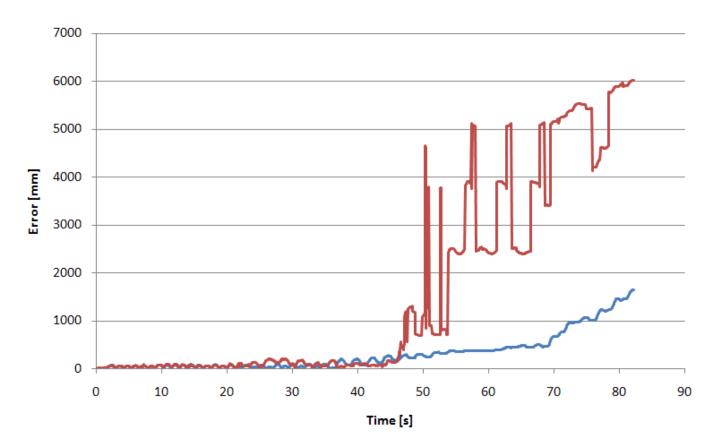
Erhalt der SLAM-Funktionalität







Erhalt der SLAM-Funktionalität







Lösung 1: Fazit

- Basiert auf FastSLAM, nur schneller und verteilt
 - Implementiert
 - Funktioniert
 - Nicht ganz schnell genug
 - Außerdem Umstieg von Partikelfilter auf MultiUKF-Lokalisierung





Lösung 2: Basierend auf MultiUKF und EKF-SLAM

- Basierend auf MultiUKF-Lokalisierung
- Percept-Buffern und Verteilung der temporären, lokalen Modelle wie in Lösung 1
- Grundidee: MultiUKF-Lokalisierung ist auch nur Percept-Buffer, nicht endgültiges Lokalisierungsergebnis
 - Verletzt die Bayes-Annahme voneinander unabhängiger Beobachtungen, da die Lokalisierung nicht resettet wird
- Je Lokalisierungs-Hypothese (oder auch nur für Haupthypothese) ein EKF über alle Roboterpositionen und die Ballposition
 - Lokale gepufferte sowie verteilte Percepte ergeben periodische Updates mit entsprechenden Kovarianzen zwischen Lokalisierung und Roboterwahrnehmungen





Lösung 3: Basierend auf MultiUKF, ohne "S" in SLAM

- Basierend auf MultiUKF-Lokalisierung
- Percept-Buffern und Verteilung der temporären, lokalen Modelle wie in Lösung 1&2
- Grundidee: Eigenes MultiUKF-Lokalisierungsergebnis ist korrekt
 - Theoretische Konsequenz: Einzelne Modelle sind unabhängig
 - Praktische Konsequenz:
 - Niedrige Berechnungskomplexität :-)
 - Fremde Beobachtungen beeinflussen eigene Lokalisierung nicht, wohl aber das modellierte Objekt
 - Fehler in relativen Abständen, was scheiße ist für Ballannäherung und Ausweichen :-/





Lösung 3: Fazit

- Basiert auf MultiUKF-Lokalisierung, ohne "S" in SLAM
- Ist umgesetzt und wird seit den GermanOpen2011 benutzt
 - Funktioniert (aber noch keine quantitative Analyse)
 - Schnell genug
 - In manchen Situationen trotzdem noch lokales Modell nötig
 - Ballannäherung auf mittlere bis kurze Distanz, Ball ständig im Blick: vertretbar, siehe Auswertung zu Lösung 1
 - Ausweichen: eigentlich nicht gewünscht, da die Robotermodellierung am meisten von gemeinsamer Modellierung profitiert
 - Taktische Entscheidungen auf globalem Gegnermodell möglich





Part 2 – Kooperative Weltmodellierung

Noch Fragen? :-)

