

Selbstlernende merkmalsbasierte Karte unter Verwendung von Random-Finite-Sets

Fabian Gies, Manuel Stübler, Stephan Reuter und Klaus Dietmayer*

Zusammenfassung: Digitale Karten sind in der Entwicklung des autonomen Fahrens von großer Bedeutung. Die Schwierigkeit besteht dabei aufgrund ständiger Änderungen in der Umwelt eine Karte aktuell zu halten. Der Einsatz von hochgenauen Referenzfahrzeugen zur Kartierung ist mit einem enormen Aufwand verbunden, weswegen in dieser Arbeit eine selbstlernende Karte vorgestellt wird. Hier erfolgt unter Einsatz vieler einfacher Agenten die Kartierung mit dem Labeled-Multi-Bernoulli-Filter und die gleichzeitige Monte-Carlo-Lokalisierung, indem aufgenommene Messungen und der derzeitige Kartenausschnitt als Random-Finite-Set modelliert werden. Erste simulative Ergebnisse zeigen bereits Verbesserungen der Karte infolge der Informationsfusion einzelner Agenten. Das vorgestellte Verfahren wird zusätzlich mit Realdaten evaluiert.

Schlüsselwörter: Kartierung, Lokalisierung, Random-Finite-Set, selbstlernende Karte

1 Einleitung

Für das autonome Fahren ist eine exakte Lokalisierung des Eigenfahrzeugs in der Umwelt unerlässlich. Hierfür wurden in der Robotik [1] Verfahren erforscht, die ohne den Einsatz des globalen Navigationssatellitensystem (GNSS) eine Positionsschätzung relativ zu einer gegebenen Karte durchführen. Die Markov-Lokalisierung, das Multi-Hypothesen-Tracking mit einem Kalmanfilter, sowie die Monte-Carlo-Lokalisierung (MCL) mit einem Partikelfilter sind hier zu nennen. Für diese Verfahren ist es essenziell, eine exakte Repräsentation der Umwelt und deren Unsicherheit mittels einer Karte darzustellen. Aufgrund der ständigen Änderungen in der Umwelt muss zur Instandhaltung einer vollständigen und aktuellen Karte der Kartierungsprozess mit einem hochgenauen Referenzfahrzeug wiederholt durchgeführt werden, was mit einem enormen Aufwand verbunden ist. Bei Verwendung einer merkmalsbasierten Karte werden üblicherweise markante statische Objekte, sogenannte Landmarken, mithilfe von Umfellsensoren detektiert. Die relativen Beobachtungen der detektierten Landmarken ermöglichen es, nach der Transformation in ein globales Koordinatensystem, die Bestimmung der globalen Eigenfahrzeugpose mittels der vorhandenen Karte. Andere Verfahren führen eine Lokalisierung anhand der relativen Beobachtungen, zur gleichzeitigen Kartierung in einer unbekanntem Umgebung durch. Dies wird in der Literatur als Simultaneous-Localization-and-Mapping (SLAM) bezeichnet [2,3]. Eine der größten Herausforderungen beim SLAM stellt die kontinuierliche Fehlerfortpflanzung

*Die Autoren sind am Institut für Mess-, Regel- und Mikrotechnik, Universität Ulm, 89081 Ulm, Deutschland. E-Mail: {vorname.nachname} at uni-ulm.de

der Unsicherheiten dar. Hierdurch wird eine korrekte Assoziation zwischen den Beobachtungen und Landmarken in der Karte erschwert und resultierende Fehler im SLAM-Algorithmus fortgeführt. Zur Minimierung des Fehlers kann beim Wiedereintritt in einen bereits bekannten Bereich eine Korrektur der geschätzten Trajektorie sowie der erstellten Karte durchgeführt werden, was hier als Gap-Closing bezeichnet wird.

Die vorgestellte Methode zur Erstellung einer selbstlernenden merkmalsbasierten Karte soll Änderungen in der Umwelt erkennen und mit möglichst geringen Voraussetzungen an die als Agenten eingesetzten Fahrzeuge eine vorhandene Karte aktualisieren. Diese Fahrzeuge sind lediglich mit einem Sensor zur Messung der Eigenbewegung, sowie der Möglichkeit zur Detektion von Landmarken ausgestattet. Ein (Differential-) Global-Positioning-System ((D)GPS) wird nicht verwendet, da diese vor allem in urbanen Gebieten unzuverlässig sind und zusätzliche Kosten entstehen. Die Lokalisierung wird auf einer bereits bekannten, jedoch unvollständigen und teilweise inkorrekten Karte mittels der MCL [4] durchgeführt. Für die Karte repräsentiert ein Random-Finite-Set (RFS) die Kardinalität und Position der Landmarken, sowie deren Unsicherheiten. Anhand der Beobachtungen in der Fahrzeugumgebung und der gegebenen Karte, detektiert das Verfahren, ob eine Lokalisierung möglich ist. Befindet sich das Fahrzeug in einem unbekanntem Teilabschnitt wird die ermittelte Trajektorie nach einem Gap-Closing korrigiert und anschließend die gefahrene Strecke mit dem Labeled-Multi-Bernoulli-Filter (LMB-Filter) [5] kartiert. Somit generiert jeder Agent eine eigenständige Karte, welche durch Informationsfusion der aufgenommenen Daten aller Agenten die Instandhaltung einer zentralen Karte gewährleistet. Im Gegensatz zum allgemeinen SLAM-Verfahren entsteht im vorgestellten Ansatz die resultierende Karte aus mehreren unabhängigen Fahrten der einzelnen Agenten.

Der restliche Teil dieses Beitrags gliedert sich in folgende 4 Teile. In Kapitel 2 wird das eingesetzte Verfahren zur Lokalisierung auf einer unvollständigen Karte und die Kartierung unter Verwendung von Random-Finite-Sets vorgestellt. Die Informationsfusion und Methodik der selbstlernenden Karte wird in Kapitel 3 beschrieben, was in Kapitel 4 anhand von Simulations- und Realdaten ausgewertet wird. Mit einer Zusammenfassung in Kapitel 5 wird das Manuskript geschlossen.

2 Modellierung der Karte als Random-Finite-Set

Die Modellierung der merkmalsbasierten Karte $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n\}$ durch ein RFS ermöglicht das gleichzeitige Schätzen der Landmarkenposition $\mathbf{m}_i = [x_i, y_i]^T$, sowie der Anzahl an Landmarken n . Hierbei sind die Elemente $\mathbf{m}_i \in \mathcal{M}$ nach einer Wahrscheinlichkeitsdichtefunktion $p(\mathbf{m}_i)$ (engl. Probability-Density-Function (PDF)) verteilt und eine Kardinalitätsverteilung repräsentiert die Anzahl an Elementen $n = |\mathcal{M}|$. Diese Modellierung für die merkmalsbasierte Karte berücksichtigt die unbekannte Anzahl von Landmarken, deren Objektzustand und die Unsicherheit einer Datenassoziation [6]. In den Finite-Set-Statistics (FISST) [7] wird die Verwendung von RFSs in einem Bayes-Framework beschrieben und in [8] ein Multi-Objekt-Bayesfilter zur Lokalisierung und simultanen Kartierung erfolgreich realisiert.

2.1 Lokalisierung

Um die Lokalisierung durchzuführen wird der momentane Kartenausschnitt $\mathcal{M}' = \mathcal{M} \cap \text{FoV}$ in jedem Zeitschritt extrahiert, der die Landmarken beinhaltet, welche sich innerhalb des potenziellen Erfassungsbereichs (engl. Field of View (FoV)) der Sensoren zur Landmarkendetektion befinden. Für die Schätzung der globalen Fahrzeugpose $\hat{\mathbf{x}} = [\hat{x}, \hat{y}, \hat{\psi}]^T$, werden die ebenso als RFS modellierten Landmarkenmessungen $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_m\}$ von lokalen in globale Koordinaten transformiert. Das Bayesfilter zur Lokalisierung unter Verwendung eines Partikelfilters ist als MCL bekannt [4]. Dabei besteht die Partikelmenge

$$\mathcal{X} := \{(\hat{\mathbf{x}}^{(i)}, w^{(i)})\}_{i=1, \dots, N} \quad (1)$$

aus N -Partikeln mit den Partikelzuständen $\hat{\mathbf{x}}^{(i)} = [\hat{x}^{(i)}, \hat{y}^{(i)}, \hat{\psi}^{(i)}]^T$ und einem zugehörigen Gewicht $w^{(i)}$. Nach einem Gewichtsupdate

$$w_k^{(i)} = w_{k-1}^{(i)} \cdot g(\mathcal{Z} | \mathcal{X}, \mathcal{M}') \quad (2)$$

der Partikel mithilfe der Multi-Objekt-Likelihood [7]

$$g(\mathcal{Z} | \mathcal{X}, \mathcal{M}') = \pi_C(\mathcal{Z}) \pi(\emptyset | \mathcal{M}') \sum_{\theta} \prod_{i: \theta(i) > 0} \frac{p_D(\mathbf{m}_i) \cdot g(\mathbf{z}_{(\theta(i))} | \mathbf{m}_i)}{(1 - p_D(\mathbf{m}_i)) \cdot \lambda_c c(\mathbf{z}_{(\theta(i))})} \quad (3)$$

errechnet sich die geschätzte Fahrzeugpose aus dem Mittelwert der gewichteten Partikel. In (3) wird die Summe über alle Assoziation $\theta : \{1, \dots, n\} \rightarrow \{0, 1, \dots, m\}$ von n Landmarken zu m Messungen berechnet. Wird einer Landmarke keine Messung zugewiesen erhält die Assoziation den Wert $\theta(i) = 0$, ansonsten ist $\theta(i) > 0$. Zusätzlich liefert der Vorfaktor $\pi_C(\mathcal{Z})$ die Wahrscheinlichkeit, dass alle Messungen $\mathbf{z}_i \in \mathcal{Z}$ Falschalarme sind, sowie der Term $\pi(\emptyset | \mathcal{M}')$ die Wahrscheinlichkeit angibt, dass keine der Landmarken $\mathbf{m}_i \in \mathcal{M}'$ detektiert wird. Des Weiteren ist $g(\cdot)$ die Einzel-Objekt-Likelihood, $p_D(\mathbf{m}_i)$ die zustands- und sensorabhängige Detektionswahrscheinlichkeit einer Landmarke und λ_c die erwartete Anzahl auftretender Falschalarme mit der räumlichen Verteilung $c(\cdot)$ innerhalb des FoV des Sensors. Ist die Karte veraltet, sodass einige Landmarken in der Umwelt nicht mehr existieren oder neu entstanden sind und zusätzlich abschnittsweise nicht ausreichend Landmarken vorkommen, kann die Lokalisierung durch falsche Gewichtsupdates fehlschlagen. Daher ist eine Bewertung der aktuellen Kartengüte sowie eine erneute Kartierung dieser Abschnitte notwendig.

2.2 Konfidenz der Lokalisierung

Um zur Laufzeit der MCL die Unsicherheit und Zuverlässigkeit der Posenschätzung [9] auszuwerten, wird in dieser Arbeit die logarithmische relative Multi-Objekt-Likelihood

$$\gamma(\mathcal{Z} | \mathcal{X}, \mathcal{M}') = \log \left(\sum_{\theta} \prod_{i: \theta(i) > 0} \frac{p_D \cdot g(\mathbf{z}_{(\theta(i))} | \mathbf{m}_i)}{(1 - p_D) \cdot \lambda_c c(\mathbf{z}_{(\theta(i))})} \right), \quad (4)$$

unter Vernachlässigung der Vorfaktoren aus (3), berechnet. Die Modellierung der Detektionswahrscheinlichkeit $p_D(\mathbf{m}_i) = p_D$ des Sensors wird zustandsunabhängig angenommen. Jede erfolgreiche Assoziation $\theta(i) > 0$ zwischen Landmarken und Messungen trägt zur

Konfidenz der Lokalisierung bei. Somit wird in jedem Zeitschritt das Lokalisierungsergebnis anhand der Landmarken in unmittelbarer Fahrzeugumgebung aus der vorhandenen Karte sowie den aufgenommenen Messungen der detektierten Landmarken bewertet. Unterschreitet dieses Konfidenzmaß einen unteren Schwellwert, so ist eine genaue Lokalisierung durch die MCL nicht gewährleistet. Der Schwellwert wird bestimmt durch die relative Betrachtung des Wertes bei der Assoziation, dass alle Messungen Falschalarme sind und keine der Landmarken detektiert wurde.

2.3 Gap-Closing

Befindet sich das Fahrzeug in einem unbekanntem Abschnitt, so wird die minimal geforderte Konfidenz der Lokalisierung über einen längeren Zeitraum nicht erfüllt. In diesem Fall findet die Prädiktion der Partikelmenge \mathcal{X} anhand der gemessenen Eigenbewegung $\mathbf{u} = [v, \omega]^T$, sprich Geschwindigkeit und Gierrate, angewendet auf die Koppelnavigation statt [1]. Zusätzlich wird die Messung $\tilde{\mathbf{u}}^{(i)} = \mathbf{u} + \delta_{v, \omega}^{(i)}$ für jedes Partikel $i \in N$ additiv verrauscht. Die geschätzte Trajektorie für diesen Abschnitt wird beim Wiedereintritt in einen bekannten Bereich der Karte mithilfe einer Zustandsoptimierung durch den Graph-SLAM-Algorithmus [10] korrigiert, was schematisch in Abb. 1 dargestellt ist. Die Korrektur wird, unter Berücksichtigung der Eigenbewegungsmessung, ausgehend vom Zeitpunkt des Verlassens eines bekannten Bereichs bis zum Wiedereintritt berechnet. Anschließend wird mit der korrigierten Trajektorie eine Kartierung durchgeführt.



Abbildung 1: Schematisch dargestelltes Gap-Closing zur Korrektur der geschätzten Trajektorie (rot), mit der wahren (schwarzen) und korrigierten (blauen) Trajektorie, sowie den bekannten (grün) und unbekanntem (schwarz) Landmarken.

2.4 Kartierung mit dem Labeled-Multi-Bernoulli-Filter

Zur Erstellung einer merkmalsbasierten Karte wird das in [5] vorgestellte LMB-Filter eingesetzt. Dieses Multi-Objekt-Filter berücksichtigt die Unsicherheit der Datenassoziation unter Verwendung von RFSs. In dieser Arbeit wird das LMB-Filter verwendet um die Objekte, hier Landmarken, in Fahrzeugumgebung zeitlich zu verfolgen (engl. “tracking”) und anschließend aus den gewonnenen Daten eine lokale Karte zu erstellen. Diese merkmalsbasierte Karte benötigt globale Koordinaten der Landmarken, weshalb die aufgenommenen Landmarkenmessungen \mathcal{Z} von Fahrzeugkoordinaten in globale Koordinaten transformiert werden. Die Transformation wird in jedem Zeitschritt anhand der geschätzten Fahrzeugpose $\hat{\mathbf{x}}$ berechnet, wodurch eine Korrelation zwischen dem Fehler der ermittelten Trajektorie und der resultierenden Karte entsteht. Das LMB-Filter verwendet für die Landmarken ein

statisches Prozessmodell und schätzt den Objektzustand $\mathbf{m} = [x, y]^T$ mit einer räumlichen Unsicherheit \mathbf{P} . Zusätzlich berechnet das LMB-Filter eine Existenzwahrscheinlichkeit r für jedes Objekt. Mithilfe der Multi-Objekt-Markovdichte aus [7] wird das Auftreten oder Verschwinden von Objekten modelliert, was in diesem Fall beim Betreten oder Verlassen des FoVs auftritt. Befindet sich eine Landmarke im FoV, korrigieren die generierten Messungen auf Basis der Multi-Objekt-Likelihood aus (3) im Updateschritt des Filters den geschätzten Objektzustand. Eine erfolgreiche Assoziation zwischen Messung und Objekt erhöht dessen Existenzwahrscheinlichkeit und beim Überschreiten des oberen Schwellenwerts ϑ_{r+} wird das Objekt mit den zugehörigen Koordinaten und einer eindeutigen ID als Landmarke in der lokalen Karte gespeichert.

Zusammenfassend erstellt jeder Agent mithilfe der vorgestellten Verfahren, auch bei unvollständigen Karten und unbekanntem Bereichen, eine eigenständige Karte für den befahrenen Bereich. Es wird die Trajektorie anhand der MCL und der Korrektur nach einem Gap-Closing geschätzt. Die Karte wird mit dem LMB-Filter auf Basis der geschätzten Trajektorie erstellt. Bei der Realisierung einer selbstlernenden Karte dient die lokale Momentaufnahme der Karte zur Aktualisierung der zentralen Karte.

3 Selbstlernende Karte

Ziel der selbstlernenden Karte ist es, durch eine Vielzahl von Agenten die zentrale Karte \mathcal{M} dauerhaft auf dem aktuellen Stand zu halten. Die Agenten erstellen in jeder Fahrt eine lokale Karte für den befahrenen Bereich, was in dieser Arbeit als Short-Term-LMB-Filter bezeichnet wird. Ein Long-Term-LMB-Filter verwendet diese lokalen Karten zur Informationsfusion mit der bereits vorhandenen zentralen Karte und reagiert so auf Änderungen in der Umwelt.

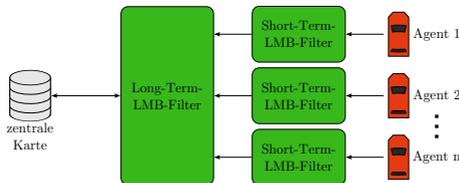


Abbildung 2: Schema der selbstlernenden Karte mit dem Short- und Long-Term-LMB-Filter.

Zum Aktualisieren der vorhandenen zentralen Karte \mathcal{M} wird ebenso ein LMB-Filter als übergeordneter Long-Term-LMB-Filter eingesetzt. Dessen Aufgabe ist die zentrale Karte langfristig in einer Datenbank zu verwalten und befahrene Abschnitte des Short-Term-LMB-Filters der Agenten in die zentrale Karte zu fusionieren, was in Abbildung 2 dargestellt wird. Vor dem Kartenupdate wird der Kartenausschnitt extrahiert, der die Landmarken beinhaltet, welche der Agent entlang seiner gefahrenen Trajektorie in seinem potenziellen FoV detektieren konnte. Nur die extrahierten Landmarken werden beim Kartenupdate berücksichtigt. Das Long-Term-LMB-Filter modelliert die ermittelten Landmarken der lokalen Karte des Short-Term-LMB-Filters als Messungs-RFS und verwendet

diese für den Updateschritt. Beim Update bekannter Landmarken wird lediglich die Existenzwahrscheinlichkeit angepasst, da die Koordinaten der Landmarken in der zentralen Karte als korrekt angenommen werden. Die Kartierung findet somit relativ zu den bekannten Landmarken der zentralen Karte statt. Existiert eine Landmarke in der zentralen Karte, jedoch nicht in der geschätzten lokalen Karte des Short-Term-LMB-Filters, bewirkt dies aufgrund keiner möglichen Assoziation zur Abnahme der Existenzwahrscheinlichkeit. Unterschreitet diese einen unteren Schwellwert ϑ_{r-} wird die Landmarke aus der zentralen Karte entfernt. Im Gegensatz dazu, werden neu entstandene Landmarken anhand des adaptiven Geburtenmodells [5] als Objekte aufgesetzt und bei Überschreiten der Existenzwahrscheinlichkeit ϑ_{r+} in der zentralen Karte hinzugefügt.

Das Zusammenspiel zwischen Short-Term-LMB-Filter und Long-Term-LMB-Filter ist die Grundidee zur Realisierung der selbstlernenden Karte, bei der unter Verwendung des LMB-Filters zu jeder Landmarke eine eindeutige ID, deren globalen Koordinaten, die räumliche Unsicherheit sowie eine Existenzwahrscheinlichkeit geschätzt werden. Mittels der Fusion aller unabhängigen Fahrten der einzelnen Agenten entsteht im Long-Term-LMB-Filter ein iterativer Prozess zur Instandhaltung der Karte. Die Fusion kompensiert die entstandenen Fehler der geschätzten Trajektorien und der Kartenerstellung jedes Agenten. Abhängig von der Existenzwahrscheinlichkeit werden alte Landmarken aus der zentralen Karte entfernt und neue hinzugefügt. Nachfolgende Agenten, können somit auf einer aktualisierten zentralen Karte arbeiten. Die Funktionsweise wird im folgenden Kapitel in einer Evaluation mit Simulations- und Realdaten gezeigt.

4 Evaluation

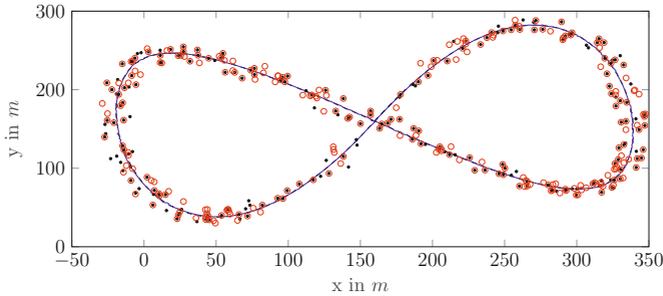
Die Auswertung des Kartenupdates und des Fehlers der merkmalsbasierten Karte wird mit der in [11] vorgestellten Metrik, dem Cardinalized-Optimal-Linear-Assignment (COLA), berechnet. Bei dieser Metrik wird neben dem Distanzfehler zwischen den wahren und geschätzten Landmarken auch die Kardinalität der beiden Mengen berücksichtigt. Bei der COLA-Distanz handelt es sich um eine dimensionslose Größe. Ist die wahre Karte gegeben durch $\mathcal{M} = \{\mathbf{m}_1, \dots, \mathbf{m}_n\}$ und die geschätzte Karte $\hat{\mathcal{M}} = \{\hat{\mathbf{m}}_1, \dots, \hat{\mathbf{m}}_{\hat{n}}\}$, so wird die COLA-Distanz bei $\hat{n} > n$ berechnet mit:

$$d_{\text{COLA}}^p(\mathcal{M}, \hat{\mathcal{M}}) = \left(\min_{\theta} \sum_{i=1}^{\hat{n}} \left(\frac{d^{(c)}(\mathbf{m}_i, \hat{\mathbf{m}}_{\theta(i)})}{c} \right)^p \right)^{1/p}. \quad (5)$$

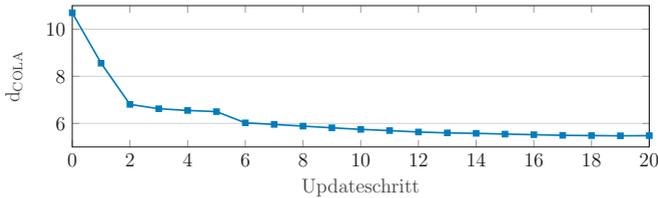
Die Assoziation θ zwischen den Elementen der Mengen wird so bestimmt, dass die Distanz minimal wird, weswegen die Ungarische Methode [12] (engl. Hungarian-Algorithmus) die optimale Assoziation zwischen den wahren und geschätzten Landmarken berechnet. Der Wert p bestimmt die Ordnung der Metrik. Die maximal zulässige Distanz der Assoziation von zwei Landmarken (engl. "cut-off distance") definiert der Parameter c .

4.1 Simulation

In der Simulation ist eine Teststrecke entworfen bei der unter definierten Systemparametern das Verfahren getestet wird. Das Szenario beinhaltet eine zufällig erzeugte wahre Karte $\mathcal{M} = \{\mathbf{m}_1, \dots, \mathbf{m}_n\}$ mit $n = 230$ Landmarken. In der verwendeten initialen Karte



(a) Exemplarische Simulationsfahrt: Die schwarze und blaue Linie sind die wahre und geschätzte Trajektorie. Schwarze Punkte sind die wahren Landmarken und rote Kreise die initial gegebenen Landmarken.



(b) Mittlere COLA-Distanz nach Kartenupdate bei 50 Monte-Carlo-Simulationen ($c = 1,5$ m, $p = 2$).

Abbildung 3: Simulationsframework mit einer exemplarischen Fahrt und dem Verlauf des Kartenfehlers aller Monte-Carlo-Simulation.

sind 70% der wahren Landmarken bekannt und 115 falsche Landmarken zufällig entlang der Strecke gestreut. Die Fahrzeugpose $\mathbf{x} = [x, y, \psi]^T$ ist in jedem Zeitschritt mit der Abtastzeit $T = 0,15$ s durch die globale Position auf der Karte und den zugehörigen Gierwinkel definiert. Der modellierte Sensor detektiert Landmarken in einem 360° -ringförmigen FoV mit einem inneren und äußeren Radius $r_{\min} = 1$ m und $r_{\max} = 20$ m. Jede Landmarke generiert einen mittelwertfreien normalverteilten Messwert in Fahrzeugkoordinaten mit einer räumlichen Unsicherheit bei einer Standardabweichung von $\sigma_{x,y} = 0,1$ m. Landmarken werden mit der Detektionswahrscheinlichkeit $p_D = 0,95$ aufgenommen. Falschalarme innerhalb des FoVs sind Poissonverteilt mit einer erwarteten Anzahl $\lambda_c = 1$. Die Messung der Eigenbewegung ist durch eine mittelwertfreie Normalverteilung mit der Standardabweichung $\sigma_v = 1$ m/s und $\sigma_\omega = 0,1$ rad/s additiv verrauscht.

Im Simulationsframework wird in jedem Zeitschritt eine Menge an Landmarkenmessungen $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_m\}$ inklusive der Falschalarme, sowie die Messung der Eigenbewegung $\mathbf{u} = [v, \omega]^T$ generiert. Die Lokalisierung mit der MCL ist, wie in [4] beschrieben, mit $N = 200$ Partikeln implementiert. Zur Bewertung des Lokalisierungsergebnis wird das vorgestellte Konfidenzmaß aus Gleichung (4) in jedem Zeitschritt berechnet. Nach

absolvieren der Teststrecke in einem Durchlauf erstellt der Short-Term-LMB-Filter auf Grundlage der geschätzten Trajektorie eine lokale Karte. In Abbildung 3a ist das Simulationsframework mit den wahren und initialen Landmarken, sowie der wahren und geschätzten Trajektorie exemplarisch an einem Durchlauf dargestellt.

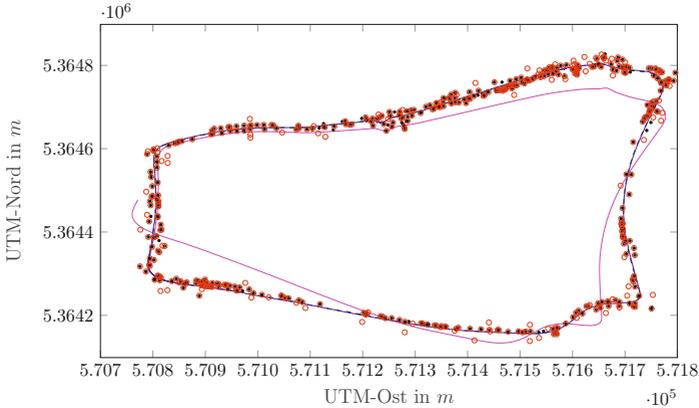
Zur Evaluation sind 50 Monte-Carlo-Simulationen mit jeweils 20 unabhängigen Durchläufen absolviert, bei denen der Short-Term-LMB-Filter eine lokale Karte erstellt. Die 20 lokalen Karten werden sequenziell dem Long-Term-LMB-Filter übergeben und ein Update der zentralen Karte relativ zur initialen Karte durchgeführt. Nach dem Kartenupdate wird die COLA-Distanz aus Gleichung (5) berechnet. In Abbildung 3b ist der Mittelwert der COLA-Distanz von 50 Monte-Carlo-Simulationen dargestellt. Es ist zu erkennen, dass der Fehler, von $d_{COLA} = 10,7$ in Schritt 0, der initialen Karte mit jedem Updateschritt des Long-Term-LMB-Filters abnimmt, bis zu einem Wert von etwa $d_{COLA} = 5,47$. Bis zum Ende der 20 Durchläufe sind die 115 falschen Landmarken aus der Karte entfernt, jedoch sind nicht alle 69 fehlenden Landmarken hinzugefügt. Eine Begründung hierfür ist, dass der Short-Term-LMB-Filter in einigen Abschnitten diese Landmarken nicht erfolgreich trackt und die Existenzwahrscheinlichkeit der Landmarke den geforderten Schwellwert $\vartheta_{r+} > 0,9$ zur Aufnahme in die lokale Karte nicht erreicht. Dadurch bekommt der Long-Term-LMB-Filter nicht ausreichend Messungen für diese Landmarke wodurch auch hier die Existenzwahrscheinlichkeit unter dem Schwellwert $\vartheta_{r+} > 0,8$ bleibt.

Bei der simulativen Auswertungen, in Abb. 3b dargestellt, konnte gezeigt werden, dass durch die Fusion mehrerer unabhängiger Fahrten die COLA-Distanz abnimmt und somit der Kardinalitätsfehler und Distanzfehler bis zu einem unteren Wert konvergiert. Das Ziel zur Verbesserung der initialen Karte unter Verwendung von mehreren unabhängigen Agenten ist in der Simulation nachgewiesen und soll folgend anhand von Realdaten geprüft werden.

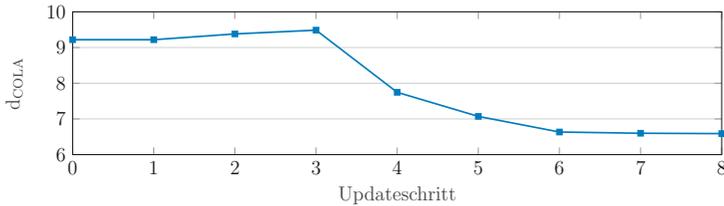
4.2 Realdatenauswertung

Für die Generierung eines Datensatzes unter realen Bedingungen wird der Versuchsträger des Instituts für Mess-, Regel- und Mikrotechnik der Universität Ulm [13] im Stadtgebiet Ulm-Lehr eingesetzt. Die zurückgelegte Distanz der Teststrecke beträgt etwa 2,9 km. Zur Detektion der Landmarken wird ein LiDAR-Sensor der Firma Ibeo verwendet. Die Messungen werden in einem nach vorne gerichteten Kreisbogen mit dem Radius 50 m und einer Winkelauflösung von 85° aufgenommen. Zur robusten Detektion von Landmarken wird der Algorithmus des Density-Based-Spatial-Clustering-of-Applications-with-Noise (DBSCAN) [14] für die Clusterbildung der einzelnen Messpunkte verwendet. Die Clustermitelpunkte haben eine räumliche Standardabweichung $\sigma_{x,y} = 0,15$ m bei einer Detektionswahrscheinlichkeit $p_D = 0,3$ und der erwarteten Anzahl $\lambda_c = 0,75$ von Falschalarmen. Für die Erstellung der wahren Karte wurden insgesamt 6 Fahrten entlang der Teststrecke mit einem DGPS als hochgenauem Referenzsystem absolviert und die Landmarken mit einem LMB-Tracker geschätzt. In der wahren Karte befinden sich insgesamt 427 Landmarken die während einer Fahrt mit dem Lidarsensor und dem DBSCAN-Algorithmus detektiert werden sollten. In der initialen Karte wurden 43 Landmarken entfernt und 86 falsche Landmarken künstlich hinzugefügt. Die Messung der Eigenbewegung wird mit dem internen ESP-Steuergerät gemessen mit einer Standardabweichung $\sigma_v = 0,06$ m/s und $\sigma_\omega = 0,0023$ rad/s im Vergleich zum Referenzsystem. In Abbildung 4a ist neben den wahren und initialen Landmarken exemplarisch eine der 6 Fahrten mit der aufgenommenen

Referenztrajektorie, der Trajektorie der Koppelnavigation auf Basis der ESP-Messungen, sowie die durch das vorgestellte Verfahren geschätzte Trajektorie dargestellt.



(a) Teststrecke in Ulm-Lehr: Die schwarz-, blau- und magentafarbenen Linien sind die wahre, geschätzte und ESP Trajektorie. Schwarze Punkte sind die wahren Landmarken und rote Kreise die initial gegebenen Landmarken.



(b) COLA-Distanz nach Kartenupdate der 8 Fahrten ($c = 1,5 \text{ m}$, $p = 2$).

Abbildung 4: Realdatensatz mit einer Fahrt durch Ulm-Lehr und den Verlauf des Kartenfehlers bei jedem Updateschritt

Auf Basis der gegebenen initialen Karte, ist es nicht möglich eine Lokalisierung mittels MCL entlang der gesamten Teststrecke durchzuführen, was abschnittsweise durch die geringe Anzahl an bekannten wahren Landmarken verursacht wird. Die Verwendung des Multi-Objekt-Konfidenzmaßes aus Abschnitt 2.2 ermöglicht eine durchgängige Schätzung der Trajektorie, sowie der Korrektur nach einem Gap-Closing. Aufgrund von langen Abschnitten, die mit dem Gap-Closing korrigiert werden, kann die Trajektorie in diesen Bereichen höhere Abweichungen aufweisen, was in der nachgelagerten lokalen Kartierung mit dem Short-Term-LMB-Filter zu erhöhten Fehlern führt. Der Long-Term-LMB-Filter verwendet zum Ausgleich ein erhöhtes Messrauschen $\sigma_z = 4 \text{ m}$ und eine niedrige Detektionswahrscheinlichkeit von $p_D = 0,6$. Dies hat zur Folge, dass das Filter mehrere

Updateschritte benötigt um falsche Landmarken aus der Karte zu entfernen und neue hinzuzufügen. In Abbildung 4b ist die COLA-Distanz nach jedem Updateschritt für 8 durchgeführte Fahrten in Ulm-Lehr abgebildet. Nach einem leichten Anstieg ist die Reduzierung des Fehlers auf eine COLA-Distanz $d_{COLA} = 6,59$ zu erkennen. Die Ursache für den Anstieg entsteht durch die erhöhte räumliche Unsicherheiten des Short-Term-LMB-Filters, wodurch auch korrekte Landmarken im Long-Term-LMB-Filter nicht assoziiert werden. Dadurch sinkt die Existenzwahrscheinlichkeit und diese Landmarken werden zunächst aus der zentralen Karte entfernt. Zum Ausgleich dieses Fehler ist die Verwendung von weiteren Fahrten oder einer verbesserten Schätzung der Trajektorie notwendig. Zusätzlich sind die aufgenommenen Fahrten über einen längeren Zeitraum von mehreren Monaten entstanden, weswegen die wahre Karte möglicherweise die Realität nicht korrekt abbildet.

5 Zusammenfassung

Das vorgestellte Verfahren beschreibt die Möglichkeit zur Erstellung einer selbstlernenden merkmalsbasierten Karte bei der unter Einsatz einer Vielzahl von Agenten die Instandhaltung realisiert ist. Das LMB-Filter bietet als Multi-Objekt-Tracker eine geeignete Anwendung für den Kartierungsprozess. Die Genauigkeit der erstellten lokalen Karte hängt stark von dem Fehler der geschätzten Pose des Eigenfahrzeugs ab. Unter Einsatz des Multi-Objekt-Konfidenzmaßes und der Korrektur der Trajektorie nach einem Gap-Closing wird dieser Fehler reduziert. Zusätzlich verbessert der übergeordnete Long-Term-LMB-Filter weitere Ungenauigkeiten und ermöglicht unter Betrachtung der Existenzwahrscheinlichkeit das Hinzufügen und Entfernen von Landmarken der zentralen Karte. Die Funktionsweise des vorgestellten Verfahrens unter Einsatz vieler unabhängiger Agenten konnte in dieser Arbeit anhand von Simulations- und Realdaten nachgewiesen werden.

Literatur

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Intelligent robotics and autonomous agents, MIT Press, 2005.
- [2] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 229–241, Jun 2001.
- [3] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part 1,” *IEEE Robotics Automation Magazine*, vol. 13, pp. 99–110, June 2006.
- [4] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte-Carlo-Localization for mobile robots,” in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2, pp. 1322–1328, IEEE, 1999.
- [5] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, “The labeled multi-Bernoulli filter,” *IEEE Trans. Signal Process.*, vol. 62, Jun 2014.
- [6] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo, *Random Finite Sets for Robot Mapping and SLAM*. Springer Berlin Heidelberg, 2011.

- [7] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., 2007.
- [8] H. Deusch, S. Reuter, and K. Dietmayer, “The labeled multi-Bernoulli SLAM filter,” *IEEE Signal Process. Lett.*, vol. 22, Oct 2015.
- [9] M. Stübler, J. Wiest, S. Reuter, and K. Dietmayer, “A confidence measure for feature-based localization using random finite sets,” in *10. Uni DAS e.V. Workshop Fahrerassistenzsysteme in Walting (Altmühltal)*, 2015.
- [10] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, “A tutorial on graph based SLAM,” *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, pp. 31–43, winter 2010.
- [11] P. Barrios, G. Naqvi, M. Adams, K. Leung, and F. Inostroza, “The cardinalized optimal linear assignment (COLA) metric for multi-object error evaluation,” in *Information Fusion (Fusion), 2015 18th International Conference on*, pp. 271–279, July 2015.
- [12] H. W. Kuhn, *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, ch. The Hungarian Method for the Assignment Problem, pp. 29–47. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- [13] F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschneider, A. Scheel, M. Stuebler, M. Bach, P. Hatzelmann, C. Wild, and K. Dietmayer, “Autonomous driving at ulm university: A modular, robust, and sensor-independent fusion approach,” in *Intelligent Vehicles Symposium*, pp. 666–673, 2015.
- [14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Kdd*, vol. 96, pp. 226–231, 1996.