

Aufgabe 3

Gruppe 4

Jonas Eckhoff Anton Jabs Florian Brach Felix Kieckhäuser

29. Januar 2019

Zum Abschluss des Kurses soll ein frei gewähltes Projekt ausgearbeitet und vorgestellt werden. Im unserem Abschlussprojekt soll das Auto auf einer unbekannten Strecke fahren und eine Karte der Umgebung erstellen. Außerdem haben wir einen Spurhalteregler entwickelt, der einen Rundkurs möglichst schnell durchfahren soll.

1 Kartenerstellung

Die ursprüngliche Idee unseres Projektes war folgende: Das Auto sollte einen unbekannten Rundkurs durchfahren und eine Karte der Strecke erstellen. Zur Unterstützung der Positionsbestimmung werden Marker plaziert, deren Position oder Abstand entlang der Strecke dem Auto bekannt oder unbekannt sind. Nach der Durchfahrt könnte das Auto anhand der Karte eine Trajektorie zur Durchfahrt des Rundkurses berechnen, bei dem das Auto den Kurs in minimaler Zeit durchfährt und sich dabei gegebenenfalls noch an Verkehrsregeln wie zum Beispiel Geschwindigkeitsbegrenzungen hält, welche als Schilder von der Kamera des Autos wargenommen wurden. Abschließend soll das Auto die berechnete Trajektorie abfahren.

Da der Umfang des Projektes ziemlich groß ist, haben wir uns fürs erste nur auf die Kartenerstellung eines unbekannten Rundkurses beschränkt und dabei verschiedene Verfahren getestet:

1.1 Meilenstein 1 - Computersimulation

Für unseren ersten Meilenstein wurde ein Kreis als unbekannte Strecke betrachtet. Mithilfe von Matlab und der Robotics Toolbox von Peter Corke sollte die Testfahrt simuliert werden und die Karte während der Durchfahrt durch den Extended Kalman Filter berechnet werden. Dabei wurden für das Auto folgende Annahmen getroffen:

- Das Auto kennt die genaue Position der Marker
- Bei jedem Berechnungsschritt stehen dem Auto fehlerbehaftete Odometriedaten (Geschwindigkeit und Lenkwinkel) zur Verfügung, wobei der Fehler normalverteilt ist und die Varianz bekannt ist
- ist ein Marker im Sichtfeld der Kamera des Autos, so kann das Auto Abstand und Winkel zum Marker bestimmen. Diese Daten sind auch mit einem normalverteilten Fehler behaftet, dessen Varianz bekannt ist.

Für die Simulation wurden einige Klassen der Robotics Toolbox modifiziert, um

1.2 Implementation

Um eine Karte von einem Gebiet $\Omega \subset \mathbb{R}^2$ zu erstellen, fährt das Auto auf einer Strecke mit konstanter, vorher definierter Geschwindigkeit v . Dabei kann das Auto mittels einer implementierten Spurverfolgung autonom oder manuell durch eine externe Lenkung fahren. Während der Fahrt werden die Bilder der Kamera aufgenommen sowie die Winkelbeschleunigung um die z -Achse. Am Ende der Fahrt wird iterativ eine geschätzte Position des Autos in der Welt zu jedem Zeitpunkt berechnet.

Die Position und Orientierung des Autos lassen sich mit den Weltkoordinaten x, y und einem Winkel ϕ beschreiben. Die Startwerte $x(0), y(0)$ und $\phi(0)$ können beliebig gewählt werden. Um nun die nächste Orientierung zu berechnen, wird zuerst die Winkelgeschwindigkeit ω und die konstante Geschwindigkeit v verwendet.

$$\begin{aligned}\phi(i+1) &= \phi(i) + \omega(i) \\ x(i+1) &= x(i) + \cos(\phi(i)) \cdot v \\ y(i+1) &= y(i) + \sin(\phi(i)) \cdot v\end{aligned}$$



(a) Kamerabild



(b) Transformation in
Vogelperspektive

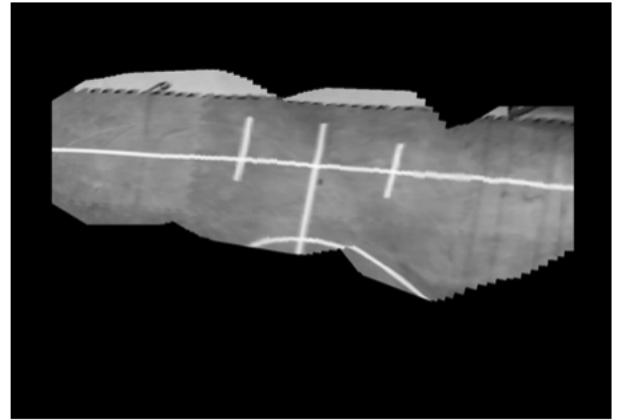
Dies ist eine gute erste Schätzung für die tatsächliche Position. Mithilfe der Videodaten soll diese Schätzung nun verbessert werden. Im ersten Schritt wird dazu jedes Bild des Videos in die Vogelperspektive transformiert. Dies hat den Vorteil, dass eine Draufsicht erzeugt wird, in der Abstände in der realen Welt linear vom Pixelabstand abhängen, solange sie in der Ebene liegen. Sei nun $\Omega_i^* := \bigcup_{j=1}^i \Omega_j$ das bereits kartografierte Gebiet, $K_i : \Omega_i^* \rightarrow \mathbb{R}$ die Karte, die bis zu Zeitpunkt i erstellt wurde und $V_{i+1} : \Omega_{i+1} \rightarrow \mathbb{R}$ das Bild der Vogelperspektive zum Zeitpunkt $i + 1$. Nun soll das Bild V_{i+1} bestmöglich in die Karte K_i integriert werden. Dafür betrachten wir das folgende Variationsproblem mit $(x, y, \phi) \in \Omega \times [-\pi, \pi]$ und der Annahme $|\Omega_{i+1} \cap \Omega_i^*| > \frac{|\Omega_{i+1}|}{2}$:

$$E_{i+1}(x, y, \phi) := \frac{1}{|\Omega_{i+1} \cap \Omega_i^*|} \sum_{(u, v) \in \Omega_{i+1} \cap \Omega_i^*} |K_i(u, v) - V_{i+1}(u + x \cos(\phi), v + y \sin(\phi))|^2 \rightarrow \min$$

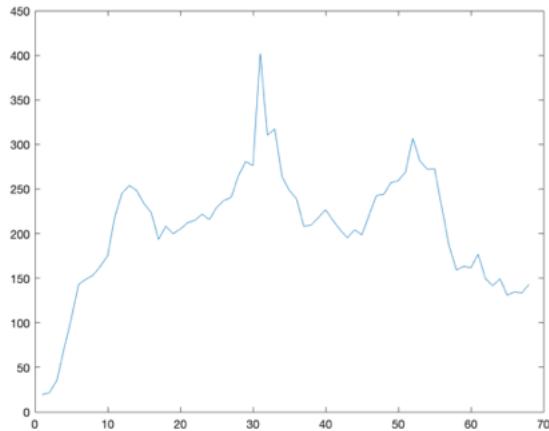
Dieses Problem lösen wir nun durch Diskretisierung bei der wir die zuvor geschätzten Werte x, y und ϕ als Iterationsstart benutzen können, um schneller zu einer Lösung zu kommen. Hierdurch erhalten wir die Transformationsparameter um das Bild V_{i+1} mit der Karte K_i zu kombinieren, sodass wir die Karte K_{i+1} erhalten. Außerdem lassen sich die zuvor geschätzten Werte aktualisieren.



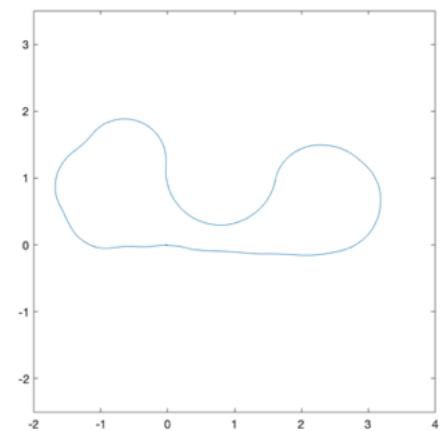
(c) Kamerabild



(d) errechnetes „Satellitenbild“



(e) Variationsfehler



(f) fertige Karte

1.3 Ausblick

Für die Zukunft kann eine Fahrt mit gut sichtbaren Makern realisiert werden. Mit den bekannten Markerpositionen kann die vorher bestimmte geschätzte Autoposition verglichen werden. Durch einen solchen Vergleich kann ein zuvor entstandener aufsummierter Fehler minimiert werden.

Des Weiteren kann durch einen Plattformwechsel (Python oder Robot Operating System (ROS)) die allgemeine Performance verbessert werden. So führt eine höhere Kameraauflösung zu einer genaueren Kartenerstellung. Außerdem könnte über eine drahtlose Verbindung zu einem PC eine Karte in Echtzeit generiert werden.

Mit einer verbesserten Kartengenauigkeit kann das Auto nach einer ersten erfolgreich gefahrenen Runde die gesammelten Daten für die eigene Navigation nutzen. Dafür muss die aktuelle geschätzte Position mit der bestehenden Karte abgeglichen und ggf. korrigiert werden.

2 Spurregler

Um den Spurhalter zu verbessern, haben wir den bereits existierenden Regler optimiert.

Die Spurerkennung lassen wir nun auf einem variablen Streifen des Bildes durchführen. Während die Breite des Streifens

konstant ist, passt sich die Position an dem zuletzt gefundenen Straßenabschnitt an. Das Auto wird also dort nach der Straße suchen, wo vorher die Straße war. Außerdem lassen wir dunkle Grautöne in der Erkennung unberücksichtigt, sodass nur noch Kanten zu hellen Objekten gefunden werden.

Zusätzlich haben wir den maximal möglichen Lenkwinkel erhöht und berücksichtigen ihn, um die Geschwindigkeit des Autos zu verändern. Bei einem großen Winkel wird die Geschwindigkeit gedrosselt, während die Geschwindigkeit erhöht wird, wenn der Winkel gering ist. Dies resultiert in einer höheren Geschwindigkeit auf geraden Strecken.



(a) Kamerabild



(b) Transformation in die Vogelperspektive