

Exploiting Sparse Semantic HD Maps for Self-Driving Vehicle Localization

Seminar Mobile Robotik

Max Dunger, Yannik Motzett

January 22, 2021

HTWG Konstanz – Fakultät Informatik

Table of contents

1. Einleitung
2. Related Work
3. Lightweight HD Mapping
4. Localization as Bayes Inference with Deep Semantics
5. Experiments
6. Conclusion

Einleitung

Probleme aktueller Lokalisierungstechniken

Gängige Lokalisierung mit GPS und IMU

- Triangulation verschiedener Satelliten
- Zu große Ungenauigkeit für Autonomes Fahren

Genauere Ortserkennungstechniken mussten entwickelt werden

- Speicherung der Umwelt in Form von Geometrie, visueller Erscheinung & Semantik
- Auch Problembehaftet

Funktionsweise der Lokalisierung

- HD-Karten enthalten Wissen über die Umwelt
- Autonome Autos müssen sich in Bezug auf die Karte lokalisieren
- Sehr strenge Anforderungen
- Vielzahl neuer, verschiedener Systementwicklungen

Thema dieser Arbeit

- Vorstellung einer Lightweight Lokalisierungsmethode
- Sie nutzt Fahrzeugdynamik und semantische Karte für die Lokalisierung
- Das Ziel ist eine zentimetergenaue Lokalisierung ohne der Nachteile bestehender Ortserkennungstechniken

Related Work

Lightweight Localization

- Steigendes Interesse an kostengünstigen Lokalisierungsverfahren
- Lediglich kartografische Karten notwendig

Vorteil:

- Deutlich weniger Speicher notwendig

Nachteile:

- Begrenzte Genauigkeit durch Leistung der Odometrie
- Sehr große Komplexität bei sich wiederholenden Mustern



Figure 1: Kartographische Karten [4]

Lightweight Localization

Beispiele

- Lokalisierung durch Schätzung der Anfangsposition und Egotrajektorie innerhalb eines kleinen Bereichs [4]
- Mehrere Erweiterungen mit Nutzung des Sonnenstandes und des Straßentyps [8]

High-precision Map-based Localization

- Hochauflösende 3D-Karten: Wurde aus Offline-SLAM mit hochpräzisem GNSS-System erstellt.
Eigenschaften: Offline, A-priori
- Im Online Modus wird ein Abgleich zwischen Sensormesswerten und a-priori Karte durchgeführt. Hierfür wird meist ein CNN verwendet.

Vorteil:

- Sehr genaue Lokalisierungsmöglichkeiten

Nachteil:

- Sehr großer Speicherbedarf, selbst bei kleinen Teststrecken

Beispiel:

- CNN zur Lokalisierung mit Hilfe der LiDAR-Intensität [2]

Place Recognition

- einer der am weitesten verbreiteten Ansätze in der Selbstlokalisierung
- im Vorhinein wird von Umgebung eine 3D-Karte erstellt (z.B. Punktewolke)
- Abrufproblem: die Pose entspricht der ähnlichsten Szene

Nachteil: 3D-Karte muss aktuell gehalten werden, da Sensordaten stark mit Karte korrelieren)

Erweiterung: bildbasierter Ansatz ist robuster gegen Veränderungen, PoseNet kann direkt Kamerapose ermitteln [6]

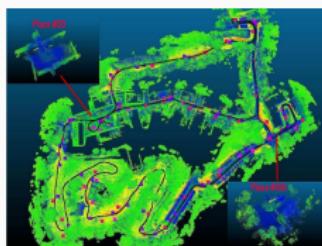


Figure 2: Place Recognition mit LiDAR-Punktewolke [5]

Gemoetry-based Localization

Lösen von Perspective-n-Point (PnP)-Problem: Übereinstimmung von Bild mit voraufgenommenem 3D-Bild

Algorithmen: random forests, SIFT, Branch-and-Bound

Nachteile: 3D-Modell muss im Vorhinein berechnet werden, funktioniert nicht mit wiederholenden geometrischen Strukturen

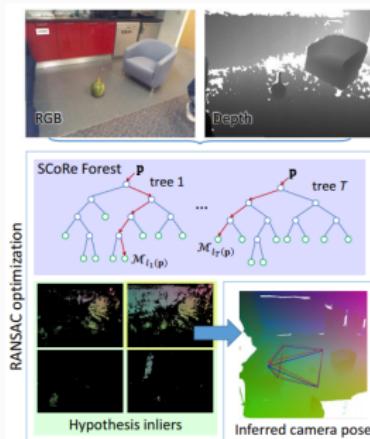


Figure 3: Regression Forests [10]

Simultaneous Localization and Mapping (SLAM)

aus der Folge von Sensordaten (Bilder, Punktwolken, Tiefenbildern) wird eine 3D-Karte der Umgebung erstellt und die Kameraposition geschätzt
Nachteil: während sich der Roboter bewegt, summieren sich Fehler auf

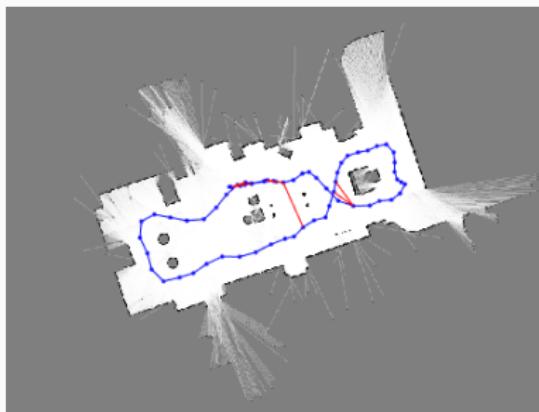


Figure 4: LiDAR Indoor SLAM [9]

Lightweight HD Mapping

Anforderungen an neue HD-Map

- einfache bzw. automatische Erstellung und Instandhaltung
- Skalierbarkeit
- Änderungen sollen in Echtzeit an Flotte bereitgestellt werden
- geringer Speicherbedarf

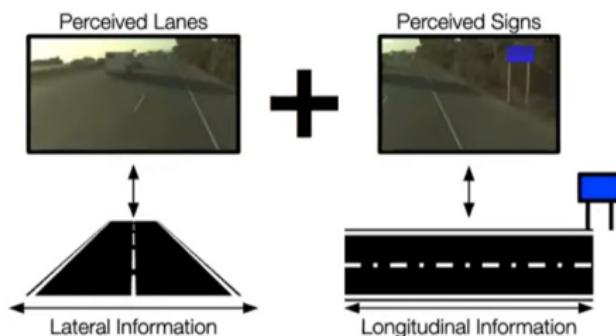


Figure 5: Sparse HD-Maps [11]

Lane Graph

Lane Graph \mathcal{L}

- Ist in der realen Welt meistens vorhanden
- Zeigen die erwartete Trajektorie der Fahrzeuge auf
- Strukturierte Darstellung des Straßennetzes aus einer Menge polygonaler Ketten (Polylinien)
- Liefert Informationen über die seitliche Position und den Kurs des Fahrzeuges

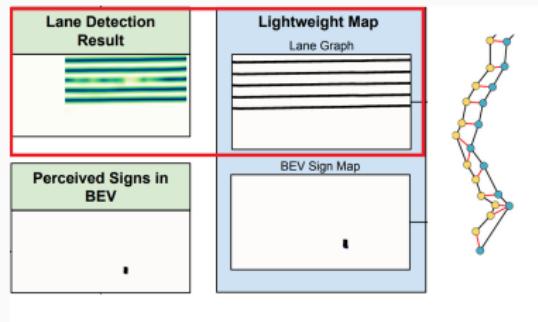


Figure 6: Lane Graph [7]

Traffic Signs

- Verkehrsschilder geben Hinweise für Lokalisierung in Längsrichtung
- es wird automatisch Map erstellt, die Informationen über Schilder enthalten

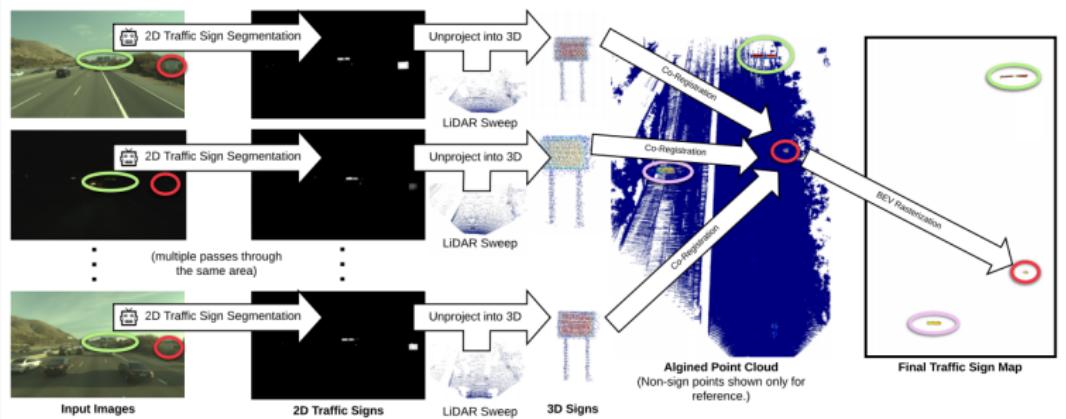


Figure 7: Erstellungsprozess Sign Map [7]

Localization as Bayes Inference with Deep Semantics

Probabilistic Pose Filter Formulation

Aufbau



Figure 8: Zusammensetzung der Wahrscheinlichkeitsformel

Probabilistic Pose Filter Formulation

Wahrscheinlichkeitsformel für die Positionsbestimmung

$$\text{Bel}_t(\mathbf{x}) = \eta \cdot P_{\text{LANE}}(\mathcal{S}_t | \mathbf{x}, \mathcal{L}; \mathbf{w}_{\text{LANE}}) P_{\text{SIGN}}(\mathcal{S}_t | \mathbf{x}, \mathcal{T}; \mathbf{w}_{\text{SIGN}})$$
$$P_{\text{GPS}}(\mathcal{G}_t | \mathbf{x}) \text{Bel}_{t|t-1}(\mathbf{x} | \mathcal{X}_t),$$

Probabilistic Pose Filter Formulation

Lane Observation Model

$$\text{Bel}_t(\mathbf{x}) = \eta [P_{\text{LANE}}(\mathcal{S}_t | \mathbf{x}, \mathcal{L}; \mathbf{w}_{\text{LANE}}) P_{\text{SIGN}}(\mathcal{S}_t | \mathbf{x}, \mathcal{T}; \mathbf{w}_{\text{SIGN}}) \\ P_{\text{GPS}}(\mathcal{G}_t | \mathbf{x}) \text{Bel}_{t|t-1}(\mathbf{x} | \mathcal{X}_t),$$

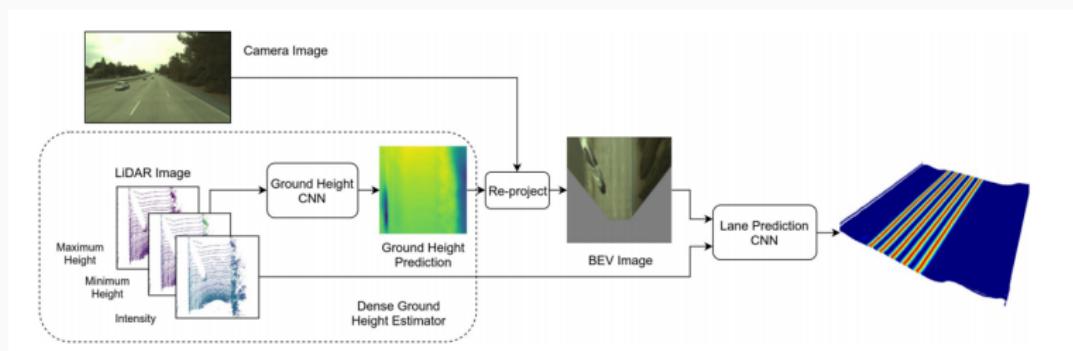


Figure 9: Aufbau des Lane Observation CNNs [1]

Probabilistic Pose Filter Formulation

Lane Observation Model

$$\text{Bel}_t(\mathbf{x}) = \eta [P_{\text{LANE}}(\mathcal{S}_t | \mathbf{x}, \mathcal{L}; \mathbf{w}_{\text{LANE}}) P_{\text{SIGN}}(\mathcal{S}_t | \mathbf{x}, \mathcal{T}; \mathbf{w}_{\text{SIGN}}) \\ P_{\text{GPS}}(\mathcal{G}_t | \mathbf{x}) \text{Bel}_{t|t-1}(\mathbf{x} | \mathcal{X}_t),$$

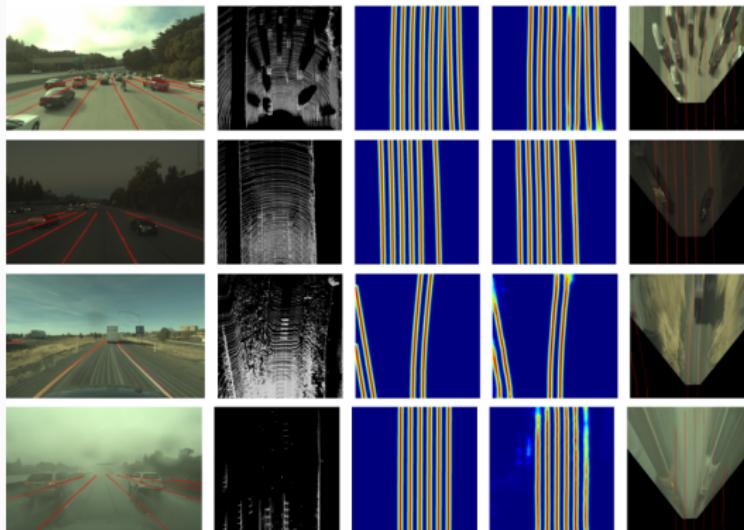


Figure 10: Beispiele der Fahrspurenerkennung [1]

Probabilistic Pose Filter Formulation

Lane Observation Model

$$\text{Bel}_t(\mathbf{x}) = \eta [P_{\text{LANE}}(\mathcal{S}_t | \mathbf{x}, \mathcal{L}; \mathbf{w}_{\text{LANE}}) P_{\text{SIGN}}(\mathcal{S}_t | \mathbf{x}, \mathcal{T}; \mathbf{w}_{\text{SIGN}}) \\ P_{\text{GPS}}(\mathcal{G}_t | \mathbf{x}) \text{Bel}_{t|t-1}(\mathbf{x} | \mathcal{X}_t),$$

Übereinstimmung zwischen Fahrspurbeobachtung und Karte

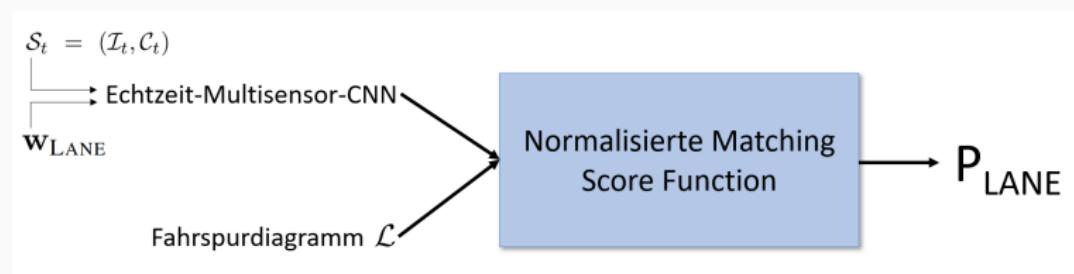


Figure 11: Matching Score Function

Probabilistic Pose Filter Formulation

Traffic Sign Observation Model

$$\text{Bel}_t(\mathbf{x}) = \eta \cdot P_{\text{LANE}}(\mathcal{S}_t | \mathbf{x}, \mathcal{L}; \mathbf{w}_{\text{LANE}}) P_{\text{SIGN}}(\mathcal{S}_t | \mathbf{x}, \mathcal{T}; \mathbf{w}_{\text{SIGN}})$$
$$P_{\text{GPS}}(\mathcal{G}_t | \mathbf{x}) \text{Bel}_{t|t-1}(\mathbf{x} | \mathcal{X}_t),$$

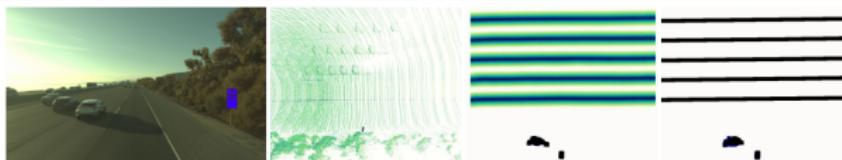


Figure 12: Traffic Sign Model [7]

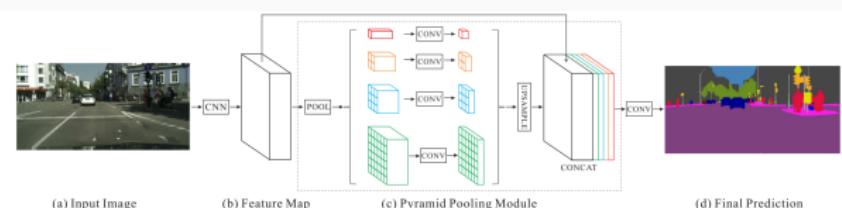


Figure 13: PSPnet [12]

Probabilistic Pose Filter Formulation

GPS Observation Model

$$\text{Bel}_t(\mathbf{x}) = \eta \cdot P_{\text{LANE}}(\mathcal{S}_t | \mathbf{x}, \mathcal{L}; \mathbf{w}_{\text{LANE}}) P_{\text{SIGN}}(\mathcal{S}_t | \mathbf{x}, \mathcal{T}; \mathbf{w}_{\text{SIGN}})$$
$$P_{\text{GPS}}(\mathcal{G}_t | \mathbf{x}) \text{Bel}_{t|t-1}(\mathbf{x} | \mathcal{X}_t),$$

- GPS-Messung \mathcal{G}_t liefert Wahrscheinlichkeit $P_{\text{GPS}}(\mathcal{G}_t | \mathbf{x})$
- Fahrzeug kann auf wenige Meter genau lokalisiert werden
- mit UTM-Koordinaten (Universal Transverse Mercator) wird Fahrzeug auf HD-Karte lokalisiert

Probabilistic Pose Filter Formulation

Dynamics Model

$$\text{Bel}_t(\mathbf{x}) = \eta \cdot P_{\text{LANE}}(\mathcal{S}_t | \mathbf{x}, \mathcal{L}; \mathbf{w}_{\text{LANE}}) P_{\text{SIGN}}(\mathcal{S}_t | \mathbf{x}, \mathcal{T}; \mathbf{w}_{\text{SIGN}}) \\ P_{\text{GPS}}(\mathcal{G}_t | \mathbf{x}) \boxed{\text{Bel}_{t|t-1}(\mathbf{x} | \mathcal{X}_t)}$$

aus vorausgegangener Pose wird aus Fahrzeuggbewegung mit Extended Kalman Filter die aktuelle Pose geschätzt

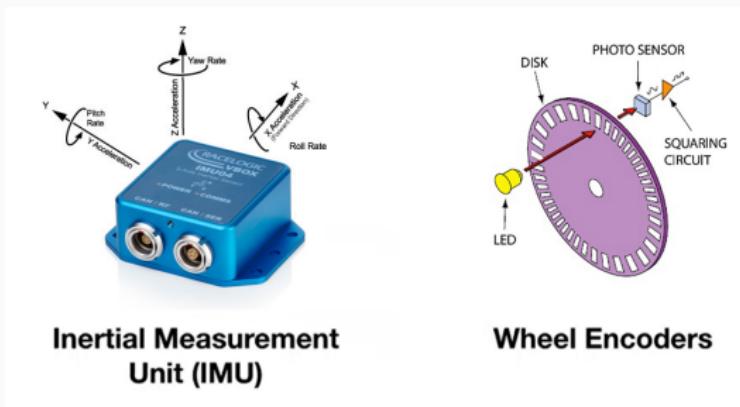


Figure 14: Sensoren für Dynamik Model [11]

Systemarchitektur

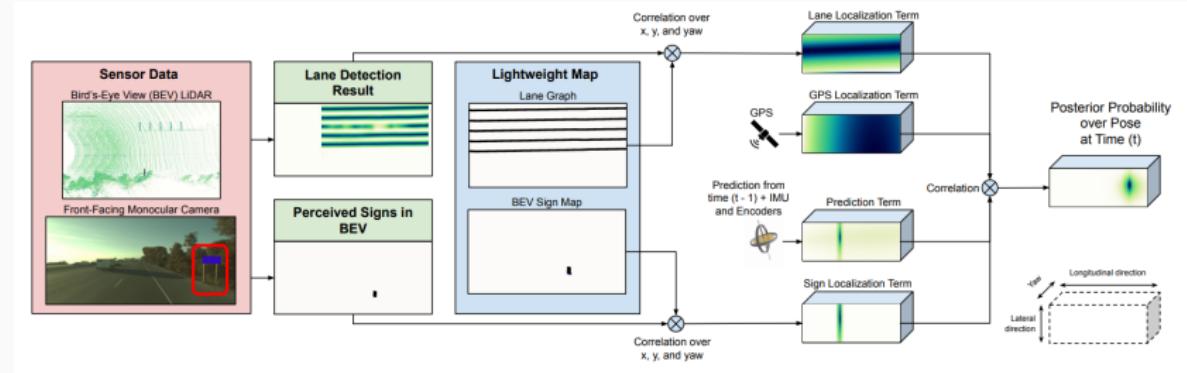


Figure 15: Systemarchitektur [7]

Experiments

Sensors

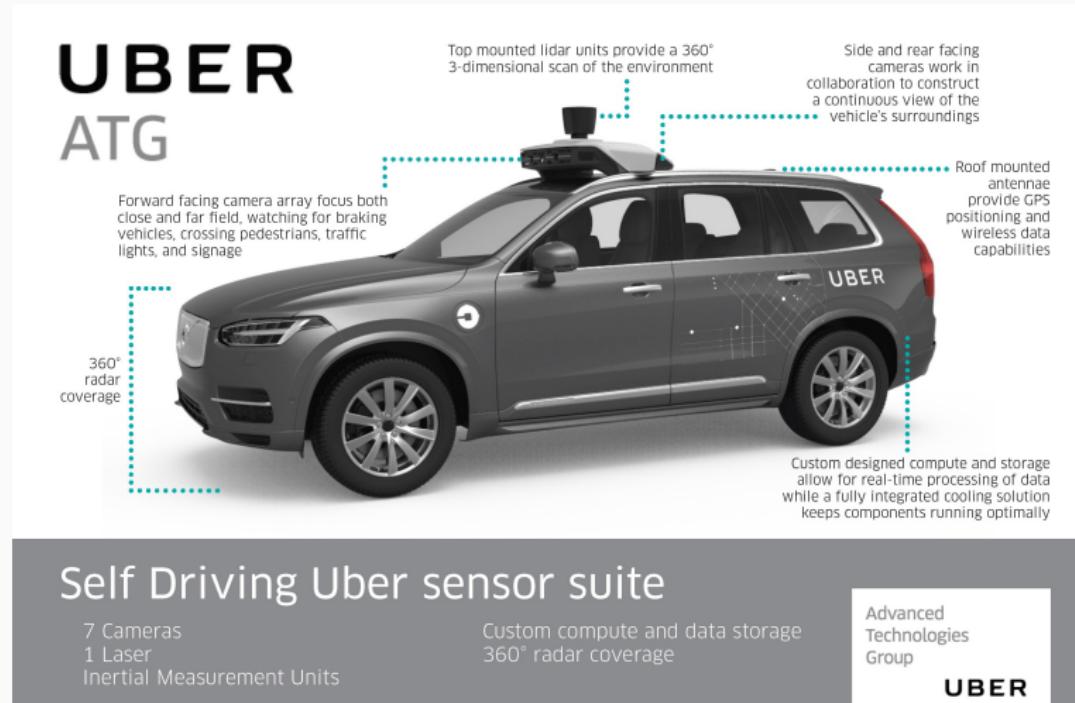


Figure 16: Uber Sensor Set – Volvo XC90 [3]

Datenset

- 312 km nordamerikanische Autobahn
- Aufteilung in 2 km Datensatzschnipsel
- Ground-Truth-3D-Lokalisierung: Geschätzt durch hochpräzises IPC-basiertes Offline-Graphen-SLAM
- Aufteilung der Daten in Trainings-, Validierungs- und Testdaten

Fahrspurerkennungs-CNN

- 50k Frames auf Trainingsregion
- Mini-Batch-Größe: 16
- Lernrate: 10^{-4}
- Gaußsche Initialisierung
- Konvergenz nach 10 Epochen

Verkehrszeichen-CNN

- Separat auf 4 GPUs
- Gesamt-Mini-Batch-Größe: 8
- Lernrate: 10^{-4}
- Training des Netzwerkes von Grund auf

Performance Analyse

- Laufzeitanalyse: 7 Hz
- Speicheranalyse: 0,55 MiB pro Quadratmeter

Methods	Longitudinal Error (m)			Lateral Error (m)		
	Median	95%	99%	Median	95%	99%
Dynamics	24.85	128.21	310.50	114.46	779.33	784.22
GPS	1.16	5.78	6.76	1.25	8.56	9.44
INS	1.59	6.89	13.62	2.34	11.02	42.34
Ours	1.12	3.55	5.92	0.05	0.18	0.23

Figure 17: Genauigkeit [7]

Method	Properties			Travelling Dist = 2km					
	Lane	GPS	Sign	Longitudinal Error (m)			Lateral Error (m)		
				Median	95%	99%	Median	95%	99%
Lane	yes	no	no	13.45	37.86	51.59	0.20	1.08	1.59
Lane+GPS	yes	yes	no	1.53	5.95	6.27	0.06	0.24	0.43
Lane+Sign	yes	no	yes	6.23	31.98	51.70	0.10	0.85	1.41
All	yes	yes	yes	1.12	3.55	5.92	0.05	0.18	0.23

Figure 18: Vergleich Anteil Komponenten [7]

Conclusion

Conclusion

- Karte benötigt 3-mal weniger Speicher als traditionelle Methoden
- Potential: Skalierbarkeit und geringere Kosten bei Ausrollen von Updates
- laterale Lokalisierung durch Spurmarkierungen, in Längsrichtung mit Verkehrsschilder
- Algorithmus wird durch GPS, IMU und Radsensoren unterstützt
- System läuft in Echtzeit mit 7 Hz
- getestet auf 300 km Autobahnabschnitt

Questions?

References i

- [1] M. Bai, G. Mátyus, N. Homayounfar, S. Wang, S. K. Lakshmikanth, and R. Urtasun.
Deep multi-sensor lane detection.
CoRR, abs/1905.01555, 2019.
- [2] I. A. Barsan, S. Wang, A. Pokrovsky, and R. Urtasun.
Learning to localize using a lidar intensity map.
In A. Billard, A. Dragan, J. Peters, and J. Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 605–616. PMLR, 29–31 Oct 2018.

References ii

- [3] D. Etherington.
Uber's self-driving cars start picking up passengers in san francisco.
<https://techcrunch.com>.
Eingesehen am 30.12.2020.
- [4] G. Floros, B. Van der Zander, and B. Leibe.
Openstreet slam: Global vehicle localization using openstreetmaps.
pages 1054–1059, 05 2013.
- [5] J. Guo, P. V. K. Borges, C. Park, and A. Gawel.
Local descriptor for robust place recognition using lidar intensity.
IEEE Robotics and Automation Letters, 4(2):1470–1477, 2019.

References iii

- [6] A. Kendall, M. Grimes, and R. Cipolla.
Convolutional networks for real-time 6-dof camera relocalization.
CoRR, abs/1505.07427, 2015.
- [7] W. Ma, I. Tartavull, I. A. Bârsan, S. Wang, M. Bai, G. Mátyus, N. Homayounfar, S. K. Lakshmikanth, A. Pokrovsky, and R. Urtasun.
Exploiting sparse semantic HD maps for self-driving vehicle localization.
CoRR, abs/1908.03274, 2019.
- [8] W.-C. Ma, S. Wang, M. Brubaker, S. Fidler, and R. Urtasun.
Find your way by observing the sun and other semantic cues.
06 2016.

References iv

- [9] MathWorks.
Implement simultaneous localization and mapping (slam) with lidar scans.
<https://mathworks.com>.
Eingesehen am 28.12.2020.
- [10] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. W. Fitzgibbon.
Scene coordinate regression forests for camera relocalization in rgb-d images.
In *CVPR*, pages 2930–2937. IEEE Computer Society, 2013.
- [11] R. Urtasun.
Localization for self-driving.
https://www.youtube.com/watch?v=V8JMwE_L5s0.
Eingesehen am 29.12.2020.

- [12] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia.

Pyramid scene parsing network.

In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017.