

Autonomes Fahren

Niclas zum Felde

Hochschule für angewandte Wissenschaften Hamburg
Dept. Informatik
Berliner Tor 7
20099 Hamburg, Deutschland
Niclas.zumFelde@haw-hamburg.de

Zusammenfassung

Diese Ausarbeitung stellt relevante Themen aus dem Forschungsgebiet der autonomen Autos vor und erläutert diese. Es werden Paper behandelt, die sich mit maschinellem Lernen, Simultaneous Localization and Mapping (SLAM) und autonomen Fahren auseinandersetzen.

Schlüsselwörter: Autonomes Fahren, neuronale Netze, SLAM, Automatisierung, Deep Learning, Reinforced Learning, Klassifizierung, Echtzeit

Autonomes Fahren

1. Einführung

Die erste Zielsetzung für autonome Fahrzeuge wurde von John McCarthy in seinem Essay „Computer Controlled Cars“ beschrieben, das erstmals 1969 veröffentlicht wurde. Kernziele sind dabei höhere Sicherheit, die Nutzer müssen nicht fahren können. Das System ist ohne Anpassung der Straßen zu gebrauchen, und es muss nur ein Zielpunkt im System eingegeben werden, zu dem das Fahrzeug den Nutzer fährt. [1]

In der heutigen Zeit ist autonomes Fahren eine der aufkommenden Technologien. Im Gartner Hype Cycle des Jahres 2018 findet sich autonomes Fahren zweimal auf der Kurve. [2]

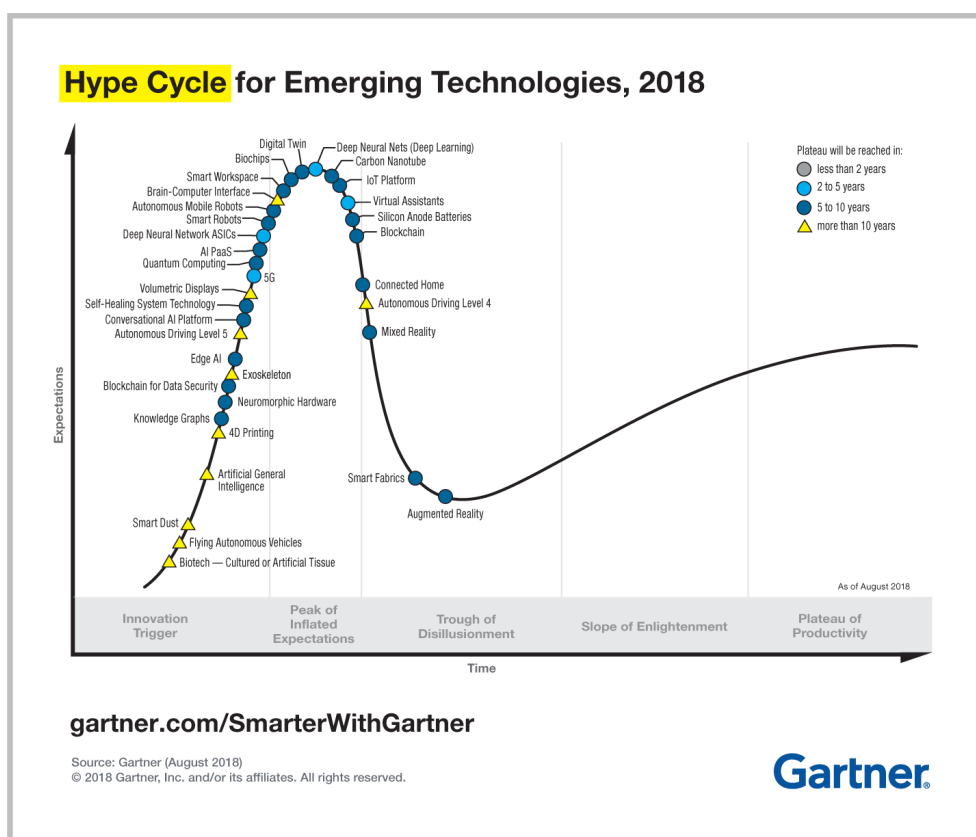


Abbildung 1 Hype Cycle for Emerging Technologies, 2018 [2]

Zum einen taucht autonomes Fahren als „Autonomous Driving Level 4“ und zum anderen als „Autonomous Driving Level 5“ in **Abbildung 1** auf. Diese Trennung basiert auf dem von der SAE International herausgegeben J3016 Standard. Dieser erstmals 2014 herausgegebene Standard definiert sechs unterschiedliche Klassen von Automatisierung in einem Automobil. Bei den Klassen „Level 0“ bis „Level 2“ handelt es sich um Systeme, bei denen der fahrende Mensch selbständig seine Fahrumgebung überwacht und nur Teilaspekte der Fahraufgabe automatisiert ablaufen.

Autonomes Fahren

KLASSE	NAME	BESCHREIBUNG
LEVEL 0	Keine Automatisierung	Alle Aspekte des Fahrens werden von einem menschlichen Fahrer übernommen. Warn- oder Eingriffssysteme können vorhanden sein.
LEVEL 1	Fahrassistenz	Lenkung des Fahrzeugs oder Beschleunigung/Bremsen werden von einer Fahrassistenz für einige Fahr-Modi übernommen. Erwartung, dass der Mensch alle anderen Aspekte des Fahrens übernimmt.
LEVEL 2	Teilweise Automatisierung	Lenkung des Fahrzeugs und Beschleunigung/Bremsen werden von einer Fahrassistenz für einige Fahr-Modi übernommen. Erwartung, dass der Mensch alle anderen Aspekte des Fahrens übernimmt.
LEVEL 3	Bedingte Automatisierung	Ein automatisiertes Fahrsystem übernimmt alle Aspekte einiger Fahr-Modi mit der Erwartung das ein Mensch auf Warnungen reagiert.
LEVEL 4	Hohe Automatisierung	Ein automatisiertes Fahrsystem übernimmt alle Aspekte einiger Fahr-Modi, selbst wenn ein Mensch nicht angemessen auf Warnungen reagiert.
LEVEL 5	Volle Automatisierung	Ein automatisiertes Fahrsystem übernimmt alle Aspekte aller Fahr-Modi in allen Umwelt- und Straßenbedingungen, die auch von einem menschlichen Fahrer gemeistert werden.

Tabelle 1 Automatisierung eines Fahrzeugs nach J3016 [3]

Die Klassen von „Level 3“ bis „Level 5“ definieren Systeme, bei denen die Fahrumgebung von einem autonomen System überwacht wird und bei denen der menschliche Fahrer nur noch Teilaspekte der Fahraufgabe übernimmt oder beim Ausfall des Systems die Aufgabe übernehmen muss. [3]

Für Realisierung solcher Systeme ist der Einsatz einer Vielzahl an unterschiedlicher Sensorik notwendig. Diese umfasst Light detection and ranging Sensoren (LiDAR), Radio detection and ranging Sensoren (RADAR), inertielle Messeinheiten (IMU), Global Positioning System-Sensoren (GPS) und Kameras. Insbesondere auf LiDAR-Sensorik wird noch einmal im Abschnitt 3 eingegangen.

2. Neuronale Netze

Künstliche neuronale Netze (im Folgenden: neuronale Netze) werden erstmal 1943 von Warren McCulloch und Walter Pitts beschrieben. [4] Neuronale Netze bilden die Funktion biologischer Neuronen ab. Ein neuronales Netz besteht aus einzelnen McCulloch-Pitts-Neuronen oder ähnlichen Abwandlungen dieser.

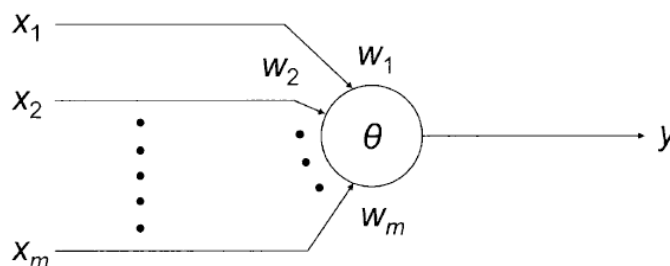


Abbildung 2 McCulloch-Pitts-Neuron [5]

Autonomes Fahren

Das Neuron (**Abbildung 3**) verfügt über $x_1 \dots x_m$ Inputlinien und über eine Outputlinie y . Die gesendeten Impulse auf der i -ten Inputline geben der Inputline ein Gewicht w_i , welches größer als Null ist. Der Output, der über die Outputlinie y zum Zeitpunkt $n + 1$ gesendet wird, ist bestimmt durch die zum Zeitpunkt n auf den Inputlinien empfangenen Impulse. [5]

Ein neuronales Netz wird dann durch das Verbinden der Input- und Outputlinien der einzelnen Neuronen gebildet. Solche Netze können in unterschiedlichen Topologien aufgebaut werden und können auch Rückkopplungen enthalten. Neuronale Netze können gut zur Lösung von Problemen eingesetzt werden über deren Lösung es wenig systematisches Wissen gibt. Dazu muss ein neuronales Netz mit Daten trainiert werden.

Schon sehr frühe Versuche zum autonomen Fahren setzten auf neuronale Netze. Das Navlab 5 ein 1995 von der Carnegie Mellon University in Pittsburgh aufgebautes Versuchsfahrzeug zum autonomen Fahren verwendete ein neuronales Netz, dieses kam im ALVINN-System des Fahrzeugs zum Einsatz. Dort wurden die Bilder einer in Fahrtrichtung montierten Kamera über das neuronale Netz auf den Lenkeinschlag des Fahrzeuges umgerechnet. Trainiert wurde das System, in dem es einen menschlichen Fahrer während der Fahrt beobachtet hat. [6]

Dies ist auch heute noch der typische Ansatz von Automobilherstellern, welche versuchen autonomes Fahren zu realisieren. Diese veröffentlichen regelmäßig die Pressemitteilungen über die von ihren Fahrzeugen autonom zurückgelegten Strecken [7] und sammeln auch Daten von Fahrzeugen mit autonomen Fähigkeiten während diese deaktiviert sind. [8]

Simuliertes Training

Ein interessanter neuer Ansatz ist es, neuronale Netzwerke nicht durch Beobachtung zu trainieren. Das Aufzeichnen des menschlichen Fahrverhaltens und das Klassifizieren der Daten ist sehr zeit- und kostenaufwendig. Stattdessen sollen autonome Fahrsysteme in simulierten Umgebungen trainiert werden. [9]

Dazu wird statt eines traditionellen neuronalen Netzwerks, welches überwacht lernt, ein sogenanntes Deep Reinforcement Learning (DLR) Netzwerk eingesetzt. Typische neuronale Netze bestehen aus einem Input-Layer, einem Hidden-Layer und einem Output-Layer. Beim Deep Learning wird ein Netzwerk mit einer tieferen Struktur verwendet, typischerweise ein Input-Layer, drei Hidden-Layer und ein Output-Layer. Hochdimensionale Funktionen lassen sich einfacher in solchen tiefen neuronalen Netzen ausdrücken. [10]

DLR kombiniert Deep Learning mit Reinforcement Learning. Mittels Reinforcement Learning kann ein Agent trainiert werden, die richtige Aktion zur Beeinflussung seiner Umwelt zu treffen. In der Trainingssimulation für ein autonomes Fahrsystem wurde hier zu Deep Q Learning eingesetzt, dabei verfügt der Agent (autonomes Fahrzeug) über Zustände und dazugehörige Aktionen. Zu jedem Zeitschritt wählt der Agent eine Aktion und ändert dadurch seinen Zustand. Diese Aktion wird mit einer Belohnung oder Bestrafung verbunden. Bei dem Verfahren Deep Q Learning bestimmt ein tiefes neuronales Netz den sogenannten Q-Wert, welcher die Belohnung für den Agenten darstellt. Deshalb wird es Deep Q Network (DQN) genannt. [9]

Autonomes Fahren

Das simulierte Training geschieht dann in einer in Unity geschaffenen Umgebung. In dieser Umwelt muss der Agent eine gerade Straße abfahren und dabei unterschiedlichen Hindernissen ausweichen. Diese Umwelt ist in **Abbildung 3** dargestellt.

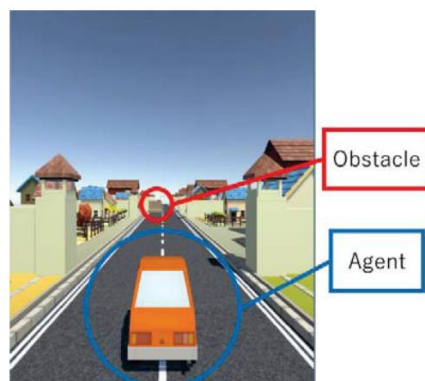


Abbildung 3 Simulierte Umwelt für Deep Reinforcement Learning [9]

Zur Vereinfachung verfügt der Agent in der Simulation nur über die Aktionen geradeaus Fahren und 10 Grad nach links oder rechts steuern. Jede Szene der Straße wird durch das DQN analysiert und die Belohnungen (Q-Werte) für die Aktionen, die der Agent ausführen kann, bestimmt. [9]

Das Lernen erfolgt dann in einem Zyklus, bei dem der Agent für jedes erfolgreich passierte Hindernis eine Belohnung erhält. Kollidiert der Agent mit einem Hindernis wird zurückgesetzt und ein neuer Lauf beginnt. Bestimmender Parameter dafür, wie gut der Agent lernen kann, war in den durchgeführten Experimenten die Geschwindigkeit, in der sich der Agent bewegt. Im nächsten Schritt soll ein Miniatur-Fahrzeug, mit dem in der Simulation trainierten Modell, einen Hindernis-Parkour abfahren. [9]

3. Simultaneous Localization and Mapping

Autonome Fahrzeuge sind stark davon abhängig ihre eigene Position bestimmen zu können, um sicheres Fahren gewährleisten zu können. Eine Technologie zur Positionsbestimmung ist das GPS. Dieses gewährleistet aber nicht in allen Fahrsituationen eine präzise Positionsbestimmung und auch bei idealen Bedingungen können Abweichungen auftreten.

GPS erlaubt ebenfalls nicht die Bestimmung der Position von anderen Verkehrsteilnehmern, die sich mit dem autonomen Fahrzeug im Straßenverkehr bewegen. Auch die Wahrnehmung von anderen Hindernissen ist über GPS nicht möglich.

Simultaneous Localization and Mapping ist ein Problem aus der Robotik, bei dem ein Roboter seine Position nicht über GPS abschätzen kann. Der Roboter muss dabei über Sensorik wie Kameras, RADAR oder LiDAR zeitgleich seine Position abschätzen und eine Karte seiner Umgebung zur Navigation erstellen.

Autonomes Fahren

Es gibt unterschiedliche Algorithmen, die das SLAM-Problem lösen: Extended Kalman Filter SLAM (EFK SLAM), SLAM mit Partikelfiltern (FastSLAM) oder Graph-basierte Lösungen (GraphSLAM). Es ist möglich SLAM durch Sensor Fusion mit in der Umwelt verteilter Sensorik zu unterstützen, um präzisere Karten zu erstellen. [11] Die Anpassung der Umwelt widerspricht allerdings den von John McCarthy aufgestellten Zielen für autonome Fahrzeuge. [1]

SLAM in Offroad-Umgebungen

Für den Einsatz in urbanen Regionen gibt es bereits viele sehr geeignete SLAM-Algorithmen, die in Echtzeit eine Positionsbestimmung durchführen können. [12] In Outdoor Umgebungen treten allerdings zusätzliche Herausforderungen für SLAM-Algorithmen auf. Die Extraction von Features ist schwierig, da die offenen Flächen sehr wenige Features bieten. Gleichzeitig bietet die Vegetation sehr viele spezifische Features an. [13] Zudem kann Staub, der aufgewirbelt wird, Probleme bei den Messungen der Sensoren verursachen.

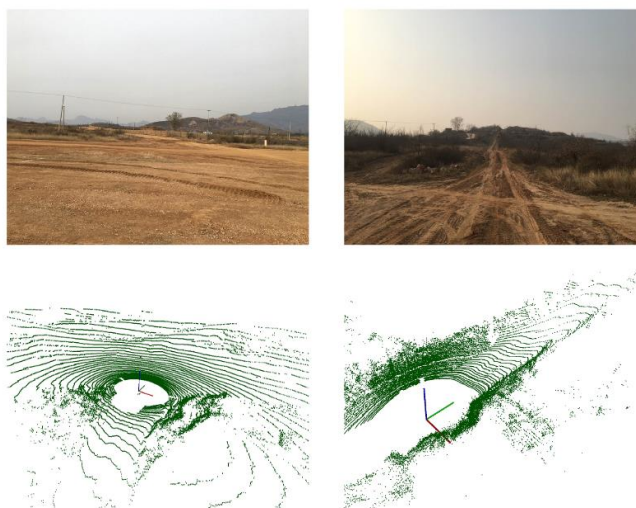


Abbildung 4 Offenes Gelände und LiDAR-Punktwolken des Geländes [13]

In **Abbildung 4** sind Aufnahmen von typischen Offroad Umgebungen zu sehen. Das offene Gelände lässt sich nicht gut auf die LiDAR-Punktwolke abbilden.

Dieses Problem lösen Algorithmen wie der Closet Probability and Feature Grid (CPFPG)-SLAM Algorithmus. Dieser ist 3D-LiDAR basiert und kombiniert Feature Extraction aus den Punktwolken, die vom LiDAR aufgenommen werden, mit einem Wahrscheinlichkeitsgitter. Damit werden die Punktwolken in einer Gitterkarte registriert, zur Schätzung Ihrer Position. Der Ablauf des Algorithmus ist in **Abbildung 5** dargestellt.

Autonomes Fahren

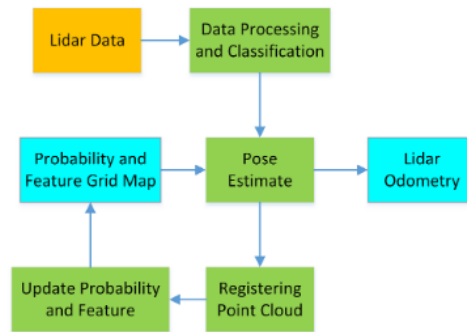


Abbildung 5 Flussdiagramm CPFG-SLAM [13]

Im Vergleich mit anderen Echtzeit-SLAM Algorithmen wie LOAM [12] oder NDT [14] liefert CPFG-SLAM einen geringeren durchschnittlichen Rotations- und Translations-Fehler. [13] CPFG-SLAM ist allerdings nur auf Offroad-Umgebungen optimiert, und die Auswirkung dynamischer Objekte (andere Fahrzeuge) auf die Positionsbestimmung ist ungetestet. In den nächsten Schritten sollen GPS-Daten über Sensorfusion in CPFG-SLAM hinzugezogen werden, um die Positionsbestimmung weiter zu verbessern. [13]

4. Klassifikation von Objekten

Autonome Fahrzeuge sollen den Straßenverkehr sicherer machen. Dazu müssen autonome Fahrsysteme auch andere Verkehrsteilnehmer erkennen können und diesen Klassen zuweisen, um mögliche Bewegungsmuster abzuschätzen. So kann ein Modell der Umwelt des Fahrzeugs aufgebaut werden. Als Sensorik wird in diesem Bereich auf 3D-LiDAR gesetzt.

Der Ablauf des allgemeinen Verfahrens ist dabei das Aufzeichnen von Daten über einen 3D-LiDAR-Sensor. In **Abbildung 6** wird der Ablauf an Hand eines 3D-Lidar Bildes dargestellt. Die aufgezeichneten Punktwolken werden daraufhin in Segmente aufgeteilt. Beispielsweise über die Projektion der 3D-Punktwolken in eine 2D-Occupancy-Grid. Überschreitet die Differenz zwischen minimaler und maximaler Höhe der Punkte in einer Zelle einen Grenzwert, wird die Zelle als Objekt segmentiert. [15]

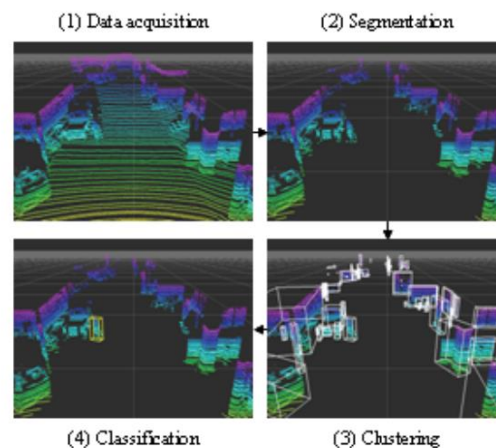


Abbildung 6 Ablauf von Objekterkennung [15]

Autonomes Fahren

Nach der Segmentierung der einzelnen Objekte erfolgt ein Clustering auf den Objekten, die in dem 2D-Occupancy-Grid enthalten sind. Ein solches Clustering kann beispielsweise mittels eines Distanz-basierten Algorithmus erfolgen. [15] Nachdem das Clustering erfolgt ist, werden dann mittels eines Klassifikationsalgorithmus die Cluster den Klassen wie „Auto“, „Fußgänger“, „Radfahrer“ oder „Hintergrund“ zu geordnet. [16]

Als Klassifikator werden oft sogenannte Boosting-Klassifikatoren verwendet. Boosting Klassifikatoren zeichnen sich dadurch aus, dass sie viele schwache Klassifikatoren für Features wie die Größe oder Geschwindigkeit eines Clusters zu einem starken Klassifikator kombinieren. [16] [17]

Wichtig bei der Klassifikation ist, dass die Klassen gewisse Toleranzen aufweisen. Damit auch Fußgänger, die dicht beieinander laufen oder Koffer ziehen, korrekt erkannt werden, wenn ein Feature wie die Größe der Cluster zur Klassifikation genutzt wird. [17] Dies gilt auch für andere Features wie Geschwindigkeit, denn Motorrad- und Fahrradfahrer weisen eine ähnliche Größe auf, bewegen sich allerdings mit unterschiedlicher Geschwindigkeit. [16]

Eine Anforderung an die Objekterkennung in einem autonomen Fahrzeug ist, dass die Erkennung eines Objektes schon in weiter Entfernung (größer 30 Meter) erfolgt, um frühzeitig Bewegungspfade der Objekte tracken zu können und Kollisionen zu vermeiden. [16] Neue Experimente mit Algorithmen, die auf den 3D-LiDAR Punktwolken Feature Extraction betreiben und Klassifikation mittels AdaBoost-Algorithmus durchführen, sind in der Lage mit 90-prozentiger Genauigkeit Objekte in bis zu 50 Meter Entfernung zu erkennen. [16]

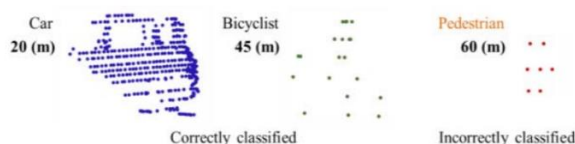


Abbildung 7 Klassifikation von Punktwolken [16]

In **Abbildung 7** sind von einem 3D-LiDAR aufgezeichnete Punktwolken zu sehen, die bereits geclustert wurden. Es ist zu sehen, dass die Cluster im Nahbereich (20m-45m) korrekt klassifiziert wurden. Das 60m entfernte Cluster wurde inkorrekt klassifiziert. Dies liegt an den wenigen Punkten die das Cluster bietet. Somit kommt es bei der Feature Extraction zu Ungenauigkeiten.

5. Herausforderungen des autonomen Fahrens

Neben den technischen Problemen und Lösungen treten natürlich auch andere Herausforderungen im Themenkomplex des autonomen Fahrens auf. Diese sind gesetzlicher, gesellschaftlicher und ethischer Natur. Einige dieser Herausforderungen sollen hier beleuchtet werden.

Gesetzliche Probleme treten beim Machine Learning auf. Aktuelle Sicherheitsstandards für Software in Automobilen orientieren sich sehr stark an dem Quellcode und der Information, die in diesem liegt. Beim Machine Learning liegt die Information jedoch nicht im Code, sondern

Autonomes Fahren

in dem errechneten Modell. Dieses ist aber nicht explizit durch den Code abgebildet. Die für z.B. Deep Learning miteinander verbundenen Layer von Neuronen machen auch das Interpretieren der Modelle für Sicherheitsanalysen der Software schwierig. [18] In diesem Gebiet lohnt es sich Ansätze wie Interpretable Machine Learning zu verfolgen.

Weitere Herausforderungen ergeben sich bei den Datensets, mit denen die Machine Learning Algorithmen für autonome Fahrzeuge trainiert werden. In **Absatz 2** wurde bereits das Training eines Modells durch eine Simulation besprochen. Die heute zum Training verwendeten Datensets setzen vor allem auf Daten, die durch das reale Fahren von Menschen oder anderen autonomen Fahrsystemen gewonnen wurden. [7] [8] Diese Datensets müssen aber nicht unbedingt vollständig sein. Im regulären Straßenverkehr treten selten Anomalien auf, diese müssen dennoch in den zum Training verwendeten Daten vorhanden sein um zu gewährleisten, dass ein autonomes Fahrzeug mit solchen Anomalien umgehen kann. [18] Hier müssen Lösungen gefunden werden, wie z.B. die Mischung von realen und simulierten Daten zur Vorbereitung der Modelle auf Anomalien.

Auch die Übertragbarkeit von Modellen ist eine Herausforderung. Wenn ein Modell nur mit Daten trainiert wird, die in einem Land aufgezeichnet wurden, aber global eingesetzt werden soll, muss das Modell möglicherweise für andere Länder neu abgestimmt werden, z.B. von Rechtsverkehr auf Linksverkehr oder Unterschiede in den Straßenlinien. Was hier untersucht werden muss, ist wie viele zusätzliche Daten dafür benötigt werden. [18]

Auch ethische Fragen ergeben sich. Ist beispielsweise eine Kollision unvermeidlich muss das autonome Fahrsystem unter Umständen entscheiden, ob es das Leben der Passagiere oder das von Passanten priorisiert. [19]

6. Fazit und Ausblick

Die Forschung im Themengebiet autonomes Fahren hat in den letzten Jahren signifikante Fortschritte gemacht. Vor allem im Bereich der Echtzeit-Erkennung von Objekten und SLAM, in der jüngeren Vergangenheit wurden dazu viele Algorithmen entworfen. Eine der treibenden Technologien im Bereich des autonomen Fahrens ist das Machine Learning.

Dennoch sind viele Forschungsfragen in diesem Bereich noch offen. Diese betreffen die Simulation von Trainingsdaten, die Interpretierbarkeit von Machine Learning und den Transfer von trainierten Modellen in andere Umgebungen, sowie das abstimmen solcher Modelle auf die neue Umgebung mittels neuer Daten.

Im weiteren Verlauf meines Masterstudiums, möchte ich mich mit Machine Learning befassen. Insbesondere mit dem Training von Modellen mittels einem hybriden Datenset, welches aus simulierten und real aufgezeichneten Daten besteht.

Abbildungsverzeichnis

Abbildung 1 Hype Cycle for Emerging Technologies, 2018 [2]	2
Abbildung 2 McCulloch-Pitts-Neuron [5]	3
Abbildung 3 Simulierte Umwelt für Deep Reinforcement Learning [9]	5
Abbildung 4 Offenes Gelände und LiDAR-Punktwolken des Geländes [13]	6
Abbildung 5 Flussdiagramm CPFG-SLAM [13]	7
Abbildung 6 Ablauf von Objekterkennung [15]	7
Abbildung 7 Klassifikation von Punktwolken [16]	8

Referenzen

- [1] J. McCarthy, „Computer Controlled Cars,“ 1996. [Online]. Available: <http://jmc.stanford.edu/commentary/progress/cars.pdf>. [Zugriff am 18 Februar 2019].
- [2] K. Panetta, „5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018,“ Gartner, 16 08 2018. [Online]. Available: <https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/>. [Zugriff am 13 Februar 2019].
- [3] On-Road Automated Driving (ORAD) Commitee, *J3016*, SAE International, 2018.
- [4] W. S. McCulloch und W. Pitts, „A logical calculus of the ideas immanent in nervous activity,“ *Bulletin of Mathematical Biophysics*, pp. 115-133, 1943.
- [5] K. Mainzer, „7.2. Neuronale Netze und Lernalgorithmen,“ in *Künstliche Intelligenz - Wann übernehmen die Maschinen?*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2019, pp. 104-110.
- [6] T. Jochem, D. Pomerleau, B. Kumar und J. Armstrong, „PANS: A Portable Navigation Platform,“ 1995. [Online]. Available: https://www.cs.cmu.edu/~tjochem/nhaa/navlab5_details.html. [Zugriff am 14 Februar 2019].
- [7] D. Silver, „Waymo Has The Most Autonomous Miles, By A Lot,“ *Forbes*, 26 Juli 2018. [Online]. Available: <https://www.forbes.com/sites/davidsilver/2018/07/26/waymo-has-the-most-autonomous-miles-by-a-lot/#63d7d38d7ee5>. [Zugriff am 20 Februar 2019].
- [8] F. Lambert, „Tesla’s fleet has accumulated over 1.2 billion miles on Autopilot and even more in ‘shadow mode’, report says,“ *Electrek*, 17 Juli 2018. [Online]. Available: <https://electrek.co/2018/07/17/tesla-autopilot-miles-shadow-mode-report/>. [Zugriff am 20 Februar 2019].

Autonomes Fahren

- [9] T. Okuyama, T. Gonsalves und J. Upadhay, „Autonomous Driving System based on Deep Q Learnig,“ *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*, pp. 201-205, 2018.
- [10] P. Kuang, W.-N. Cao und Q. Wu, „Preview on structures and algorithms of deep learning,“ *2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP)*, pp. 176-179, 2014.
- [11] K. Morioka und F. Hashikawa, „Mobile robot navigation based on interactive SLAM with an intelligent space,“ *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 788-789, 2011.
- [12] J. Zhang und S. Singh, „LOAM: Lidar Odometry and Mapping in Real-time,“ *Robotics: Science and Systems*, Bd. 2, 2014.
- [13] K. Ji, H. Chen, J. Gong, G. Xiong, J. Qi und T. Yi, „CPFG-SLAM:A Robust Simultaneous Localization and Mapping based on LIDAR in Off-Road Environment,“ *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 650-655, 2018.
- [14] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss und W. Burgard, „OctoMap: an efficient probabilistic 3D mapping framework based on octrees,“ *Autonomous Robots*, Bd. 34, Nr. 3, pp. 189-206, 2013.
- [15] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito und J. Miura, „Pedestrian Recognition Using High-definition LIDAR,“ *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 405-410, 2011.
- [16] M. Yoshioka, N. Suganuma, K. Yoneda und M. Aldibaja, „Real-time object classification for autonomous vehicle using LIDAR,“ *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, pp. 210-211, 2017.
- [17] S. Thurn, J. Levinson und A. Teichmann, „Towards 3D object recognition via classification of arbitrary object tracks,“ *IEEE International Conference on Robotics and Automation*, pp. 4034-4041, 2011.
- [18] Q. Rao und J. Frtunikj, „Deep Learning for Self-Driving Cars: Chances and Challenges,“ *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, pp. 35-38, 2018.
- [19] „Whom should self-driving cars protect in an accident?,“ *The Economist*, 27 Oktober 2018. [Online]. Available: <https://www.economist.com/science-and-technology/2018/10/27/whom-should-self-driving-cars-protect-in-an-accident>. [Zugriff am 23 Februar 2019].