

Robuste Ampeldetektion und Haltlinienfreigabe durch Kartenassoziation in automatisierten Fahrzeugen

Robust traffic light detection and stopline release by map association for automated driving

Richard Fehler¹, Kevin Rösch¹, Fabian Immel¹ und Christoph Stiller^{1,2}

¹ FZI Forschungszentrum Informatik, Intelligent Systems and Production
Engineering

Haid-und-Neu-Straße 10-14, 76131 Karlsruhe

² KIT, Institut für Mess- und Regelungstechnik
Engler-Bunte-Ring 21, 76131 Karlsruhe

Zusammenfassung In dieser Arbeit präsentieren wir eine in Deutschland auf öffentlichen Straßen erfolgreich erprobte Systemarchitektur, um robust Haltlinien zugeordnete Ampeln wahrzunehmen und den resultierenden Freigabezustand zu filtern. Ampeldetektionen werden mit Ampeln aus einer HD-Karte [1] assoziiert, um sie der Haltlinie zuzuordnen die das Fahrzeug betrifft. Somit kann die in der Karte hinterlegten Beziehung zwischen Ampeln und Haltlinie genutzt werden, ohne auf eine fehleranfällige Rückprojektion angewiesen zu sein. Wir evaluieren das Gesamtsystem anhand ausgewählter Szenarien in Karlsruhe und Sindelfingen und zeigen damit die Einsatzbereitschaft in realen automatisierten Fahrzeugen.

Schlüsselwörter Deep learning, autonomes Fahren, Ampeln, Detektion, Assoziation

Abstract In this work, we present a system architecture that has been successfully tested on public roads in Germany to robustly perceive traffic lights assigned to stop lines and filter the resulting release state. Traffic light detections are associated with traffic lights from an HD map [1] to assign them to the stop line relevant to the vehicle. This allows the use of the relationship be-

tween traffic lights and stop lines stored in the map, without relying on an error-prone back-projection. We evaluate the overall system using selected scenarios in Karlsruhe and Sindelfingen, demonstrating its readiness for deployment in real automated vehicles.

Keywords Deep learning, autonomous driving, traffic lights, detection, association

1 Einleitung

1.1 Stand der Technik

Ermöglicht durch die hochgenaue 3D Kartierung und Lokalisierung haben automatisierte Fahrzeuge den für Sie relevanten Ampelzustand bereits vor der Entwicklung von Deep-Learning-basierter Objektdetektoren erfolgreich durch die Rückprojektion und anschließender Klassifizierung geschätzt [2]. Moderne Ansätze verwenden Deep-Learning-Dektoren, trainiert auf speziellen Ampeldatensätzen [3]. Arbeiten die das gesamte System und nicht nur die Wahrnehmung im Rahmen des hochautomatisierten kartenbasierten [1] Fahrens entwerfen und evaluieren gibt es in deutlich geringerem Umfang [2].

1.2 Ziel der Arbeit

Die Rückprojektion von in 3D kartierten Ampeln in das Kamerakoordinatensystem und anschließende Klassifikation des Ampelzustandes ist sehr anfällig für Lokalisierungs-, Kalibrierungs- oder Kartierungsfehler. Der Beitrag dieser Arbeit ist der Entwurf und die Umsetzung eines robusten Systems zur Freigabe von durch Ampeln geregelten Haltelinien.

2 Methode

Die Systemarchitektur wird in die Module *Detektion*, *Klassifikation*, *3D Erweiterung*, *Assoziation*, und *Filterung* der Sektionen 1-5 unterteilt. Der

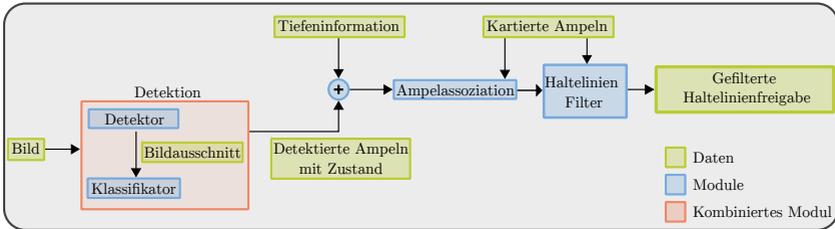


Abbildung 1: Die Systemarchitektur eines robusten Freigabesystems für Ampelhaltlinien.

Datenfluss von Eingabe-Bildern bis zu den gefilterten Haltlinienfreigaben der HD-Karte wird in Abb. 1 in Relation zu den Funktionsmodulen gesetzt um eine Meta Architektur zu bilden.

2.1 Ampeldetektion

Um die instabile Rückprojektion von kartierten Ampeln zur Bestimmung des Ampelzustands zu umgehen ist eine Ampeldetektor mit ausreichend hoher Genauigkeit und Trefferquote notwendig. Wir verwenden einen Faster-RCNN [4] Detektor mit einem ResNet50 [5] Rückgrad, trainiert auf dem Microsoft COCO Datensatz [6], welcher Ampeln in ausreichend hoher Anzahl annotiert hat. Dieser zweistufige Box Detektor liefert gute Detektionsergebnisse auch für kleine Objekte bei Bildern mit Auflösungen von über 3000 Pixel Bildbreite. Die Inferenzzeit beträgt 70 ms bei Nutzung von PyTorch [7] und einer RTX 6000 Ada GPU. Die Boxen der Ampeldetektionen werden an den feingranularen Typ- und Zustandsklassifikator innerhalb des kombinierten Moduls *Detektion* im selben Prozess überreicht.

2.2 Klassifikation

Wir generieren Bildausschnitte für jede detektierte Ampel basierend auf den Detektionen. Diese Bildausschnitte werden durch ein doppelköpfigen Klassifikator in Zustand und Typ eingeordnet, entsprechend der *pictogram* und *state* Attribut-Klassen des DriveU Traffic Light Dataset (DTLD) [8]. Der Klassifikator basiert auf einem modifizierten



Abbildung 2: Die in Typ und Zustand klassifizierte Ampeldetektion. Die Farbe der Box entspricht dem Ampelzustand. Unbekannte Zustände sind in pinker Farbe gekennzeichnet. Am besten digital zu betrachten.

EfficientNet [9] Modell welches auf den Bildausschnitten der annotierten Boxen des DTLD Datensatzes trainiert wurde.

Tabelle 1: Die durch den doppelköpfigen Klassifikator eingeordneten Zustand- und Typklassen.

Ampelzustand	Ampeltyp
off	circle
red	arrow_left
yellow	arrow_right
red_yellow	arrow_straight
green	arrow_straight_left
unknown	arrow_straight_right
	tram
	bicycle
	pedestrian
	pedestrian_bicycle
	unknown

2.3 Erweiterung in den 3D Raum

Die Boxen der Ampeldetektionen werden durch Tiefeninformationen in den 3D Raum erweitert. Die Tiefeninformationen können dabei dynamisch von verschiedenen Sensorquellen und Tiefenschätzungsmethoden gewonnen werden. Wenn genügend Lidar-

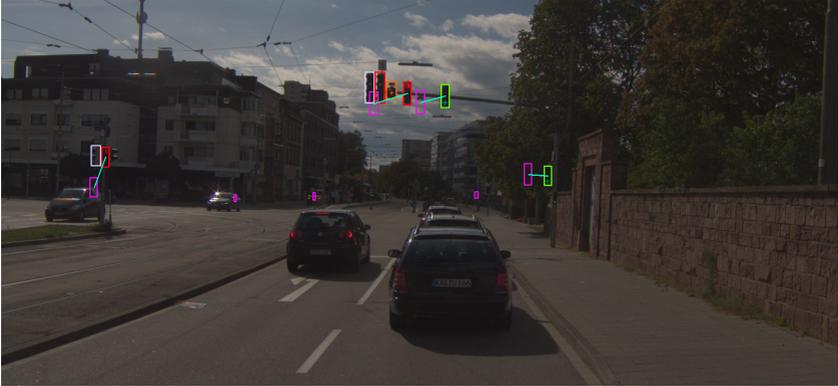


Abbildung 3: die Ampeln der HD-Karte werden mit den in den 3D Raum erweiterten Detektionen assoziiert (türkise Verbindungslinie). Die Ampeln der HD-Karte werden zur Visualisierung des Lokalisierungsfehlers in das Kamerabild rückprojiziert (Pink).

punkte in die Box der Detektion fallen, werden aus der Menge der Lidarpunkte durch Gewichtung und Clustering ein skalarer Tiefenwert berechnet. Wenn die Box ausserhalb des Lidarstrahlbereichs liegt, kann eine dichte Tiefenkarte geschätzt werden, basierend auf Stereokameras oder Monokularen Kameras um den skalaren Tiefenwert der Box zu bestimmen. Durch Ausnutzung der normierten Ampeldimensionen haben sich heuristische Methoden zur Tiefenschätzung der Ampel als möglich erwiesen. Wenn fehlende Lidardaten und eine geringe Rechenleistung das System einschränken, kann eine objektbedingte Tiefenschätzung direkt durch einen 2D Detektor gelernt werden [10]. In dem evaluierten System kommt ein 128 Zeilen Lidar zum Einsatz mit einer heuristischen Tiefenschätzung für den vom Lidar nicht abgedeckten extremen Nahbereich.

2.4 Assoziation

Die nun in den 3D Raum erweiterten Detektion werden den kartierten Ampeln in einem bipartiten Graph zur Assoziation gegenübergestellt. Wir formulieren das Assoziationsproblem als Minimum Cost Flow Problem [11, 12]. Gegenüber der weiter verbreiteten Lösung mit dem

Hungarian Algorithmus für optimale 1 zu 1 Assoziation, stellt das Minimum Cost Flow Problem eine Verallgemeinerung dar. Es ist hiermit auch die Assoziation von mehreren Detektionen zu einem Kartenelement oder umgekehrt möglich. Dies erlaubt es Doppeldetektionen robust abzufangen, welche die Ergebnisse bei herkömmlicher 1 zu 1 Assoziation stark beeinträchtigen. Weitere Vorteile sind die intuitive Einführung eines Assoziationskostenmaximums und die Möglichkeit, beispielsweise die global besten 5 Assoziationen zu liefern, formuliert in einem einzigen Optimierungsproblem.

Die Kostenfunktion berücksichtigt den Ampeltyp, die Dimensionen der Ampel und ihre 3D Position. Der bipartite Graph wird, wie bei dem Minimum Cost Flow Problem üblich, als Flussgraph einer Quelle zu einer Senke dargestellt, der dabei die bipartiten Knoten der detektierten und kartierten Ampeln passieren muss. Der Quellfluss kann als die Anzahl der kartierten Ampeln in einem Suchradius, die Anzahl der Detektionen oder das geringere von beiden gewählt werden, wobei wir in unseren Versuchen letztere Option verwenden. Ein Kostenmaximum filtert Assoziation mit zu hohen Kosten. Die Typen *bicycle*, *pedestrian* aus Tabelle 1 und deren Kombination werden nicht mit Fahrzeugampeln der Karte assoziiert. Als *tram* klassifizierte Typen werden weiterhin verwendet, aber mit höheren Assoziationskosten verbunden, da abgeschaltete Fahrzeugampeln häufig fälschlicherweise dem Typ *tram* zugeordnet werden.

2.5 Filter

Für jede kartierte *Haltelinie* werden die assoziierten Zustände *aller* der Linie zugehörigen Ampeln durch einen gleitenden Fensterfilter geglättet. Dies sorgt für eine robuste Haltelinienfreigabe auch bei Verdeckungen oder instabilen Detektionen und Klassifizierungen durch schlechte Sichtbedingungen. Wenn nicht genügend aktuelle *green* oder *off* Zustände im Fenster vorhanden sind wird die Haltelinie geschlossen. Durch den robusten Entwurf können auch instabile Wahrnehmungsergebnisse zu korrekten Haltelinienfreigaben führen, indem bei unbekanntem und somit geschlossenem Freigabezustand das Fahrzeug verzögernd in bessere Sichtverhältnisse rückt.

3 Ergebnisse

Das System wurde im realen Strassenverkehr in Karlsruhe und Sindelfingen im automatisierten Testbetrieb eines Versuchsfahrzeugs evaluiert. Die präsentierte Versuchsreihe zeigt ein korrekte Freigabe in über 96% der Haltelinienüberfahrungen der Erprobungsstrecken, auch bei Erprobungen in Regen oder der Dämmerung.

3.1 Klassifikationsergebnisse

Der gewählte Detektor erzielt auf dem COCO Validierungsdatensatz eine mAP von 46.7. Diese Detektionsperformance ist die Basis für alle weiteren Teile des Systems. Der auf Boxendetektionen aufbauende doppeltköpfige Klassifikator wurde auf dem DTLT [8] Datensatz trainiert. Die initialen Gewichte für das Modell wurden auf dem ImageNet-1k Datensatz [13] vortrainiert. Die Hyperparameter des Modells und des Trainingregimes sind in Tabelle 2 aufgeführt.

Parameters	Value
Model	Efficient Net [9]
Optimierer	ADAM [14]
Kostenfunktion	Kreuzentropie
Lernrate	$1e^{-3}$
Epochen	60
Bildgröße	224^2
Batch size	64

Tabelle 2: Die Hyperparameter des Klassifikators.

Mit diesen Parametern erreicht das Netzwerk eine Zustandsklassifikator eine Genauigkeit von 96% auf dem DTLT Datensatz. Das Netzwerk wurde besonders wegen der schnellen Trainingszeit und der effiziente Inferenzzeit von unter 30ms ausgewählt.

3.2 Systemevaluation im Strassenverkehr

Wir befahren in einem automatisierten Versuchsfahrzeug eine wie in Abb. 4 und Abb. 5 zu sehende kartierte Route in Karlsruhe und Sindelfingen innerhalb von vier Monaten mehrfach ab um die System-

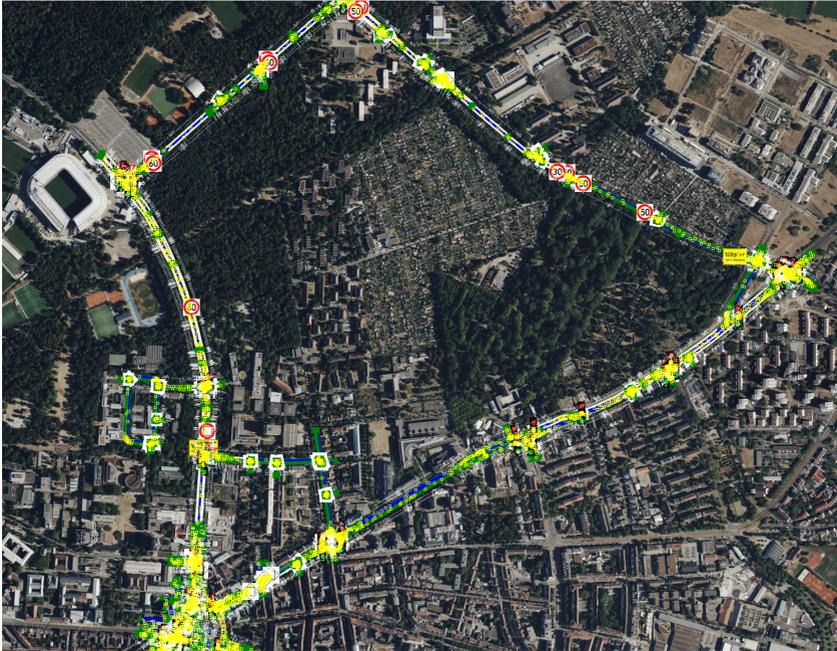


Abbildung 4: Die Evaluierte Route in Karlsruhe mit einer Länge von über 7 km. Luftbilder von Bing Maps © 2024 Microsoft Corporation

funktion zu bewerten. Dabei zählen wir die korrekten Ampelhaltelinienzustände bei An- und Überfahrung der Haltelinie und die entsprechend inkorrekten Ampelhaltelinienzustände. Die Befahrungen haben an sonnigen Tagen, an bewölkten Tagen und auch bei Dämmerung und Starkregen stattgefunden. folgende Probleme haben dabei zu Systemfehlern geführt:

- Regen und schlechte Sicht sorgen für fehlende Detektionen von insbesondere abgeschalteten (*off*) Ampeln und damit zu einem unbekanntem Zustand der Ampeln oder falschen Assoziationen mit leuchtenden Ampeln.
- hoch montierte, rot leuchtende Ampeln die teilweise wie in Abb. 6 abgeschnitten werden, werden detektiert, jedoch als *off*

klassifiziert.

- Ampeln die Aufgrund der Fahrzeugausrichtung nicht in das Frontkamerabild fallen, können durch diese nicht beobachtet werden.

Route	korrekt	inkorrekt	# Haltelinien	Präzision (%)
KA	155	5	160	96.88
SiFi	49	2	51	96.08
Gesamt	204	7	211	96.68

Tabelle 3: Anzahl korrekter und inkorrekt Haltelinienzustände entlang der Routen in Karlsruhe (KA) und Sindelfingen (SiFi).

Eine falsche Ausrichtungen des Fahrzeugs kann durch Erweiterung der Detektion und Assoziation auf das 360 Grad Ringkamera System des Fahrzeugs ausgeglichen werden. Die Problematik abgeschnittener Ampeln kann durch einen angepassten Sensoraufbau, wie vertikale Orientierung der vorderen Kameras oder durch eine angepasste Halte- distanz zu Ampeln und deren Haltelinien durch die Trajektorienplanung gelöst werden. Den Einfluss der Trainingsdaten auf die Detektionsrate bei schlechten Sichtbedingungen muss weiter untersucht werden. Eine aktive Lernstrategie kann hier angewandt werden um fehlende Bedingungen in den Trainingsdatensätzen auszugleichen.

4 Zusammenfassung

Der vorgeschlagene Entwurf und dessen Umsetzung eines robusten Systems zur Freigabe von durch Ampeln geregelten Haltelinien konnte den Haltelinienzustand in 96% der 211 evaluierten Haltelinienüberfahrungen in Karlsruhe und Sindelfingen schätzen und hat den automatisierten Betrieb eines Versuchsfahrzeugs ermöglicht. Das System zeigt ein robustes Verhalten gegenüber Lokalisierungs-, Kalibrierungs- oder Kartierungsfehlern und die verbleibenden beobachteten systematischen Schwachstellen des Systems wurden erörtert.

Literatur

1. F. Poggenschans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, "Lanelet2: A high-definition map framework for the future of automated driving," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2018.
2. N. Fairfield and C. Urmson, "Traffic light mapping and detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
3. K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
4. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.
5. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
6. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.
7. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
8. A. Fregin, J. Muller, U. Krebel, and K. Dietmayer, "The driveu traffic light dataset: Introduction and comparison with existing datasets," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
9. M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2019.
10. J.-H. Pauls, R. Fehler, M. Lauer, and C. Stiller, "Combining 2d and 3d datasets with object-conditioned depth estimation," in *IEEE Intelligent Vehicles Symposium (IV)*, 2022.
11. A. V. Goldberg and R. E. Tarjan, "Finding minimum-cost circulations by successive approximation," *Mathematics of Operations Research*, 1990.
12. A. V. Goldberg and M. Kharitonov, *On implementing scaling push-relabel algorithms for the minimum-cost flow problem*, 1992.

R. Fehler et al.

13. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.
14. D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.