

Simulationsumgebung zur Ermittlung der lichttechnischen Einflüsse auf kamerabasierte Objekterkennung

Sundermeier, Max Caspar; Glück, Tobias; Dai, Zhuoqun; Hamlaoui, Rayen; Lachmayer, Roland

*Institut für Produktentwicklung und Gerätebau der Leibniz Universität Hannover
Deutschland, Garbsen, An der Universität 1*

Zusammenfassung

Zur Ermittlung der lichttechnischen Einflüsse auf kamerabasierte Objekterkennung im Straßenverkehr wird in diesem Beitrag eine Simulationsumgebung ausgewählt und vorgestellt, in der eine reale Versuchsumgebung abgebildet wird. Dabei sollen mithilfe von Ray-Tracing in der Simulationsumgebung fotorealistische Abbildungen, welche zur Auswertung durch Objekterkennungsalgorithmen verwendet werden können, erzeugt werden. Der Fokus liegt auf der Übertragung von gemessenen Lichtverteilungen in die Simulationsumgebung sowie die Validierung der simulierten photometrischen Größen. Dabei wird eine algorithmische Vorgehensweise zur automatisierten Bestimmung der simulierten Beleuchtungsstärke vorgestellt sowie die Abweichungen der Simulation von den Messwerten erörtert. Durch eine Versuchsreihe wird die simulative Abbildung der Lichtverteilungen validiert.

1. Einleitung

Zur Validierung von Algorithmen und technischen Systemen in Fahrzeuganwendungen werden herkömmlicherweise Testfahrten durchgeführt, um die korrekte Funktionsweise nachzuweisen. Im Rahmen der Testfahrten werden die unterschiedlichen Einsatzszenarien und Extremfälle des Einsatzes erprobt. Auch die Funktionsweise von Scheinwerfersystemen sowie kamerabasierte Objekterkennungsalgorithmen (OEA) werden durch den Einsatz auf Testfahrten validiert. Bisher werden physiologische Studien zur Auslegung von Fahrzeugfunktionen herangezogen [1]. Auf Basis von physiologischen Studien werden im Anschluss Lichtverteilungen bestimmt, die durch Fahrzeugscheinwerfer in den Verkehrsraum abgebildet werden [2]. Zur Reduktion des Versuchsaufwandes sowie der Generierung von replizierbaren Szenarien wird in der Forschung die simulative Abbildung von Lichtverteilungen erörtert [3]. Aus vorangegangener Arbeit kann für die Auslegung von Scheinwerfern der Bedarf nach physikalisch korrekten Abbildungen von Lichtverteilungen in Simulationsumgebungen abgeleitet werden, die die Simulation von kamerabasierter Objekterkennung ermöglichen [4], [5].

Eine Herausforderung bei der Abbildung der kamerabasierten Objekterkennung für Fahrzeuge ist die Abbildung der Scheinwerferlichtverteilung. Um den Unterschied zwischen Realwelt und Simulation zu minimieren, werden in dieser Arbeit Herausforderungen und Lösungsansätze zur Abbildung von Scheinwerferlichtverteilungen in einer ausgewählten Simulationsumgebung vorgestellt.

Zunächst werden mögliche Simulationsumgebungen vorgestellt und hinsichtlich der Verwendbarkeit für die Simulation der kamerabasierten Objekterkennung bewertet sowie eine

Simulationsumgebung ausgewählt. Daran anschließend wird die Simulation von gemessenen Scheinwerferlichtverteilungen vorgestellt sowie die digitale Vermessung simulierter Scheinwerferlichtverteilungen. Abschließend erfolgt eine Validierung der Simulation der Scheinwerferlichtverteilung.

2. Auswahl einer Simulationssoftware

Um die kamerabasierte Objekterkennung im Fahrzeugumfeld simulativ abzubilden, werden im Folgenden Anforderungen an die Simulationssoftware vorgestellt:

Zur photorealistischen Darstellung der Lichtpropagation im Raum sowieso der Streuung, Reflexion und Transmission ist die Möglichkeit des Ray-Tracings der Simulationssoftware erforderlich. Darüber hinaus sind in die Simulationssoftware frei konstruierbare Objekte einzufügen, weshalb die Simulationssoftware eine 3D-Modellierung unterstützen muss. Um diese Objekte in die Simulationssoftware einzubringen und zu bewegen ist eine Automatisierungsschnittstelle erforderlich, die insbesondere auch die Trajektorienplanung von automatisierten Fahrzeugen ermöglicht. Neben der Trajektorienplanung von Fahrzeugen, müssen auch andere Bestandteile der Simulationssoftware, wie Wetter oder Umgebungslicht, automatisiert beeinflusst werden können, weshalb die Fahrzeugtrajektorienplanung und die Automatisierungsschnittstelle zwei voneinander getrennte Anforderungen darstellen. Um die kamerabasierte Objekterkennung abbilden zu können, ist darüber hinaus eine Darstellbarkeit von Detektionsobjekten erforderlich. Zu der Darstellbarkeit gehören unter anderem die Einbindung von Scheinwerferlichtverteilungen und die Anpassung von Reflexionseigenschaften von Oberflächen. In Tabelle 1 sind die Anforderungen an die Simulationssoftware zusammengefasst und die Erfüllung der Anforderungen durch unterschiedliche Softwares angegeben.

Tab. 1: Vergleich von Simulationsumgebungen für die kamerabasierte Objekterkennung (X = erfüllt)

Anforderungen	Blender [6]	Unreal Engine 5 [7]	Carla (Unreal Engine 4) [8]	CarMaker [9]	SVL Simulator [10]
Ray-Tracing	X	X	O	O	O
3D-Modellierung	X	O	O	O	O
Automatisierungsschnittstelle	O	X	X	X	O
Fahrzeugtrajektorienplanung	O	X	X	X	X
Darstellbarkeit von Detektionsobjekten	X	X	O	O	O
Summe	3	4	2	2	1

Aufgrund der Übereinstimmung mit den Anforderungen wird für die Simulation der kamerabasierten Objekterkennung Unreal Engine 5 (UE5) (Version 5.0.3) gewählt und für die wei-

teren Versuche verwendet. Um die 3D-Modellierung trotz der Verwendung von UE5 zu ermöglichen, wird Blender als Modellierungssoftware verwendet und die modellierten Objekte in die Umgebung von UE5 portiert.

3. Simulation von Scheinwerferlichtverteilungen

Um Fahrsituationen bei Nacht realistisch simulieren zu können, ist eine Abbildung von Scheinwerferlichtverteilungen in der Simulationsumgebung UE5 erforderlich. Dazu werden gemessene Lichtverteilungen in eine Punktlichtquelle in UE5 portiert. Scheinwerferlichtverteilungen werden üblicherweise mit Fernfeldgoniophotometern gemessen und die Dateien im .ies-Format gespeichert. Bei .ies-Dateien handelt es sich um ASCII-Dateien, welche neben einigen Metadaten die Winkelauflösung der Messung sowie die Messwerte enthalten. Im Folgenden werden zwei Methoden vorgestellt, bei denen sowohl die .ies-Datei direkt in die Simulationsumgebung importiert wird, als auch als *Light Function Material* dargestellt wird.

In Abbildung 1 ist der Vergleich zwischen einer beispielhaften gemessenen Lichtverteilung sowie ihrer direkten Abbildung in UE5 im .ies-Dateiformat gezeigt.

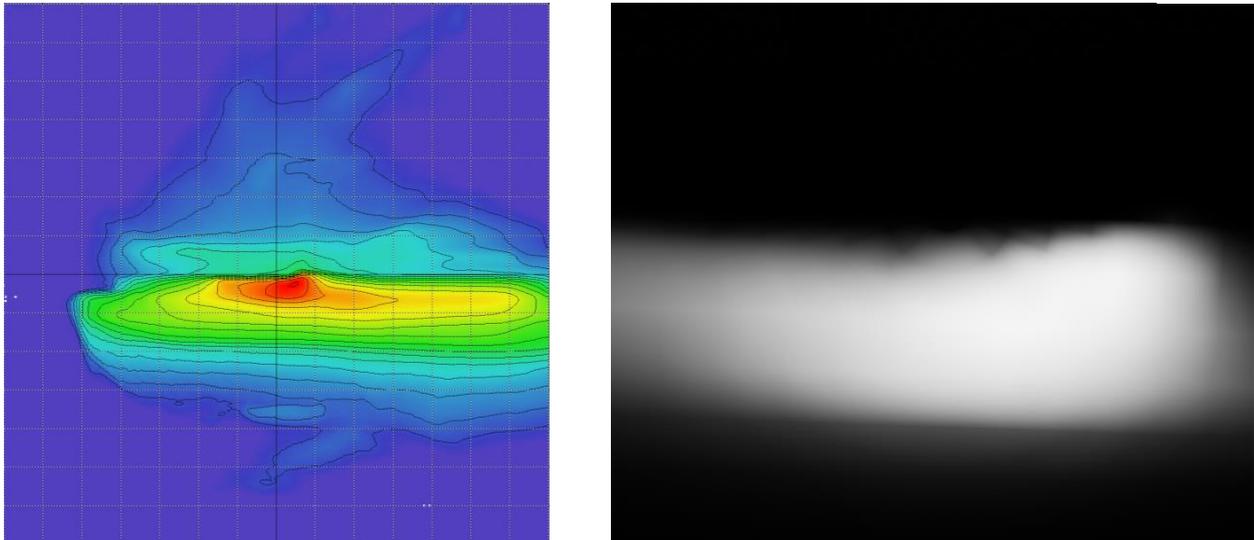


Abb. 1: Vergleich gemessene Lichtverteilung (links) vs. in UE5 als .ies (rechts)

Der in Abbildung 1 gezeigte Vergleich zwischen gemessener Lichtverteilung und simulierter Lichtverteilung zeigt insbesondere im Übergangsbereich der Hell-Dunkel-Grenze (HDG) Abweichungen. Die Abweichungen sind auf eine von der Simulationsumgebung durchgeführte Drehung der simulierten Lichtquelle um die optische Achse zurückzuführen, weshalb die simulative Darstellung einer .ies-Datei für nicht-rotationssymmetrische Lichtverteilungen Nachteile birgt.

Als Alternative wird die Lichtverteilung aus der .ies-Datei in eine .png-Datei überführt, die unter Berücksichtigung eines Leistungsfaktors l , jedem gemessenen Helligkeitswert I_i einen Relativwert

$$R_i = 255 \cdot \frac{I_i - I_{min}}{I_{max} - I_{min}} \cdot l \quad (1)$$

zwischen 0 und 255 zuordnet.

Anhand des in (1) bestimmten Relativwertes kann eine .png-Datei erstellt werden, die als *Light Function Material* einer Punktlichtquelle in Unreal Engine zugeordnet werden kann. Der Algorithmus zur Erstellung einer .png-Datei aus einer .ies-Datei ist öffentlich verfügbar [11]. In Abbildung 2 ist entsprechend Abbildung 1 der Vergleich zwischen der gemessenen Lichtverteilung und der simulierten Lichtverteilung auf Basis des *Light Function Materials* gezeigt.

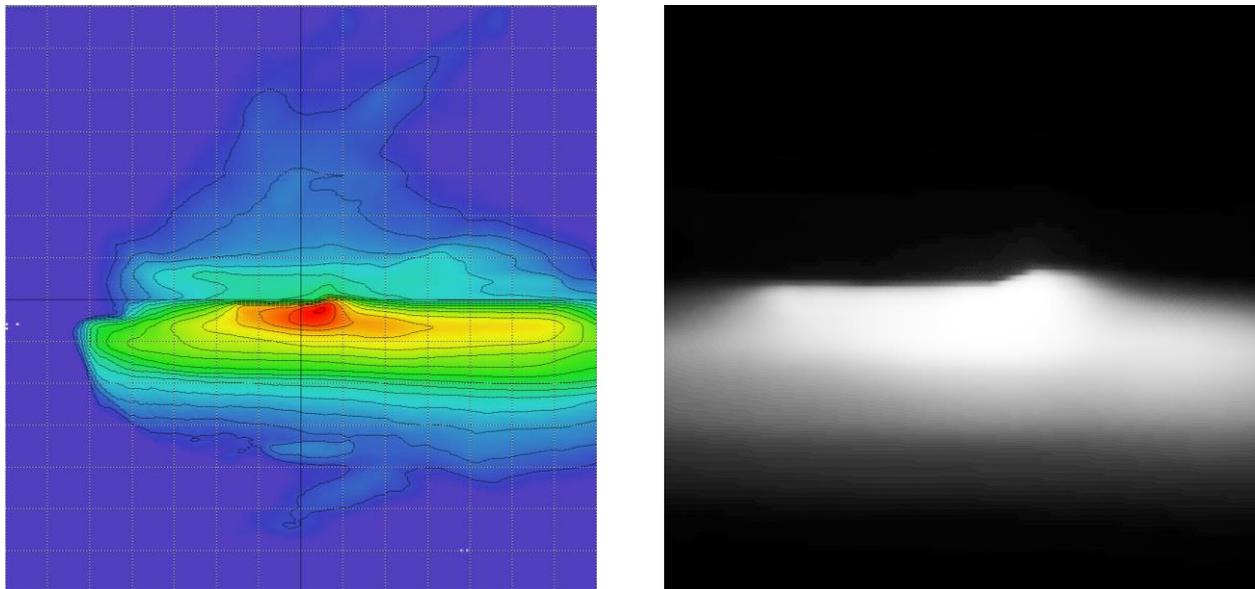


Abb. 2: Vergleich gemessene Lichtverteilung (links) vs. In UE5 als *Light Function Material* (rechts)

In der in Abbildung 2 rechts gezeigten Lichtverteilung ist eine klarere HDG zu erkennen, als in Abbildung 1 rechts. Der Unterschied zwischen der Abweichung der beiden Methoden von der gemessenen Lichtverteilung lässt sich allerdings bisher nicht quantifizieren.

4. Digitale Vermessung und Validierung

Um die zuvor generierten Lichtverteilungen zu validieren, ist eine digitale Vermessung der Scheinwerferlichtverteilung erforderlich, worauf im folgenden Kapitel eingegangen wird. Zwar bietet UE5 keine native Möglichkeit zur Auswertung von gesamten Lichtverteilungen, allerdings können Werte der Beleuchtungsstärke an diskreten Punkten im Blickfeld ermittelt werden. Anhand der ermittelten Werte kann eine digitale Vermessung der Scheinwerferlichtverteilung erfolgen, was eine Validierung der simulativen Abbildung ermöglicht.

Die Funktionsweise eines realen Fernfeldgoniophotometers wird in UE5 nachgebildet. Die in Abbildung 3 dargestellte virtuelle Messumgebung besteht aus drei Komponenten: Eine Punktlichtquelle mit dem *Light Function Material*, eine Kamera und eine Projektionswand.

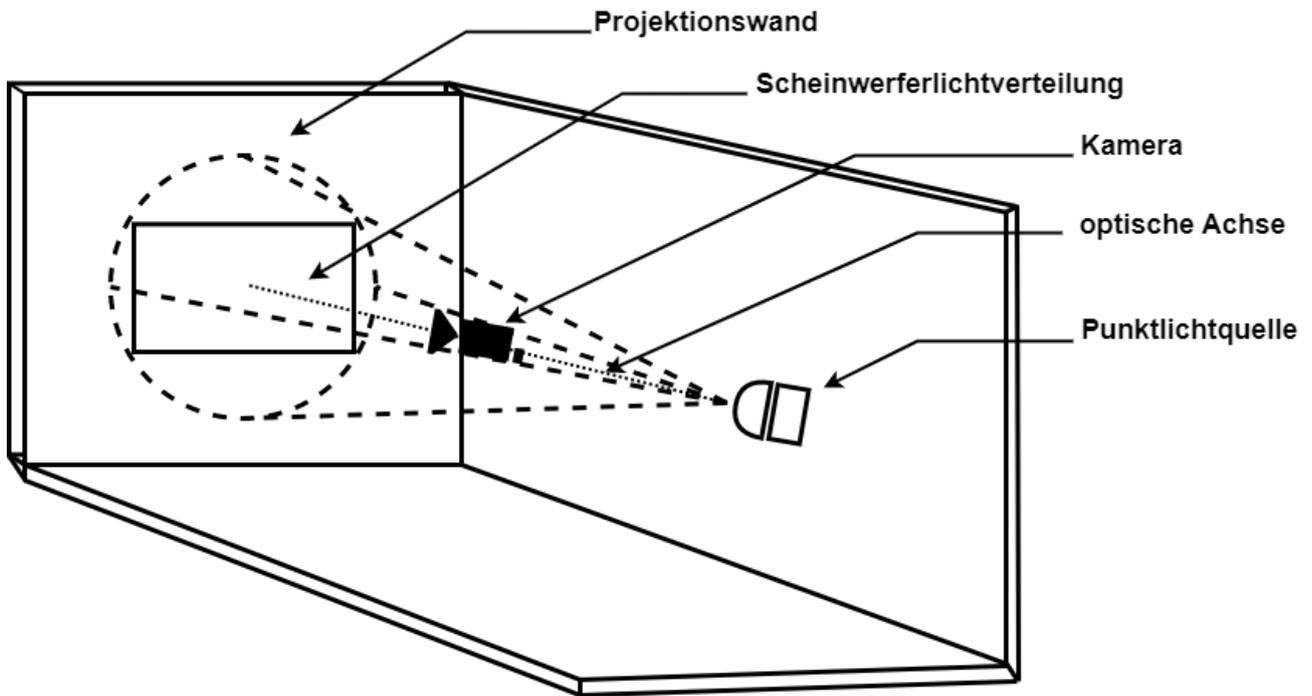


Abb. 3: Virtuelle Lichtmessstrecke

Ähnlich vergleichbar zur Vermessung von Scheinwerfern in 25 m Messentfernung, wird eine virtuelle Wand in 25 m Entfernung von der Punktlichtquelle platziert. Die virtuelle Umgebung wird mittels der in die Punktlichtquelle portierten Scheinwerferlichtverteilung beleuchtet und die Lichtverteilung wird auf der Wand abgebildet. Anschließend können die Messwerte für die Beleuchtungsstärke in diskreten Punkten der Simulationsumgebung ausgelesen werden.

Nach der Einrichtung der Messumgebung erfolgt die Vermessung der Scheinwerferlichtverteilung indem die Position der Kamera geändert wird. Der Startpunkt entspricht der oberen linken Ecke der Scheinwerferlichtverteilung, danach wird die Kamera in einer der Auflösung entsprechenden Schrittweite horizontal (y) bzw. vertikal (z) verschoben, bis in der unteren rechten Ecke das Ende des Messbereiches erreicht ist.

Die Kamera wird in die in Abbildung 4 dargestellten Startposition aufgestellt. Die Bestimmung der Ausgangsposition (y_{start}, z_{start}) hängt von der initialen horizontalen (φ_s) und vertikalen (ϑ_s) Neigung der gemessenen Lichtverteilung ab.

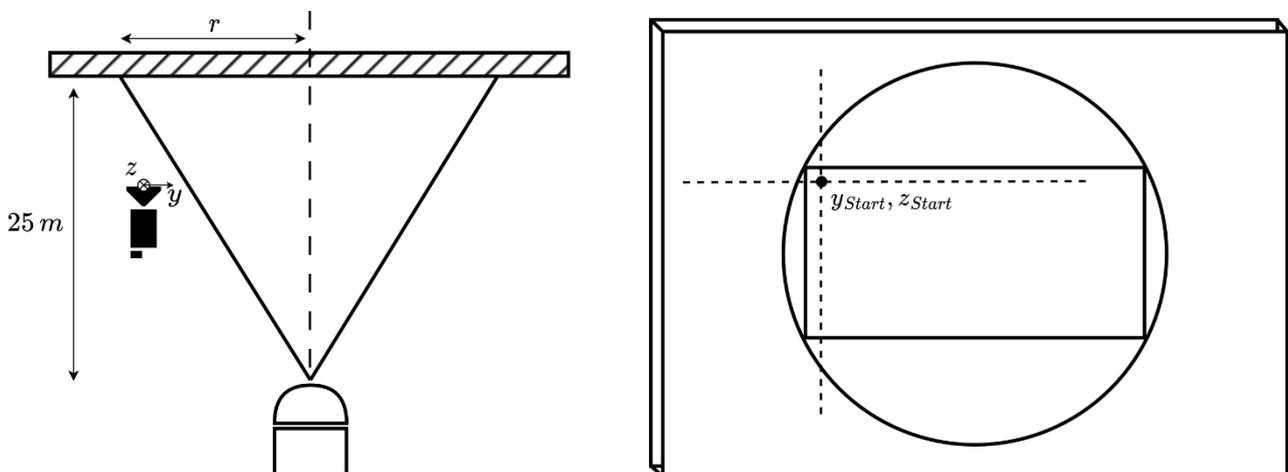


Abb. 4: Links: Überblick über die Messumgebung und die verwendeten Größen. Rechts: Abbildung des Startpunktes für die virtuelle Vermessung

Die Startkoordinaten der Messung werden berechnet zu

$$y_{Start} = 25 \text{ m} \cdot \tan(\varphi_s) \quad (2)$$

und

$$z_{Start} = 25 \text{ m} \cdot \tan(\vartheta_s). \quad (3)$$

Um den Vergleich der realen und simulierten Messwerte zu ermöglichen, wird der Fehlerfaktor

$$\beta = \frac{E_{v,real}}{E_{v,simulation}} \quad (4)$$

definiert. In Abbildung 5 ist der Verlauf des mittleren horizontalen Fehlerfaktors dargestellt

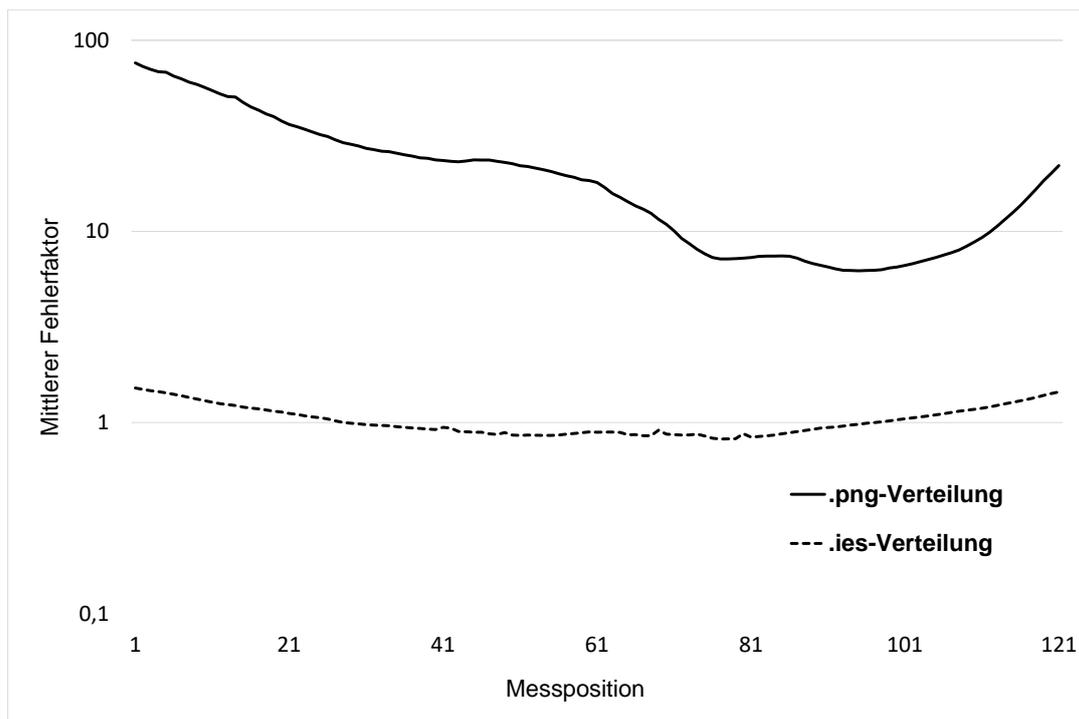


Abb. 5: Vergleich des horizontalen mittleren Fehlerfaktors für die .ies- und die .png-Lichtverteilung (logarithmische Skala)

Der in Abbildung 5 gezeigte Vergleich des horizontalen mittleren Fehlerfaktors quantifiziert den Unterschied zwischen der Abweichung von Simulation zu Realität. Während für die menschliche Wahrnehmung die .png-Lichtverteilung eine höhere Übereinstimmung zu der gemessenen realen Lichtverteilung aufweist, zeigt die Messung eine deutliche Abweichung sowohl des mittleren Fehlerfaktors als auch der Einzelwerte. Zur Erweiterung der Aussagekraft ist in Abbildung 6 ist der mittlere vertikale Fehlerfaktor dargestellt.

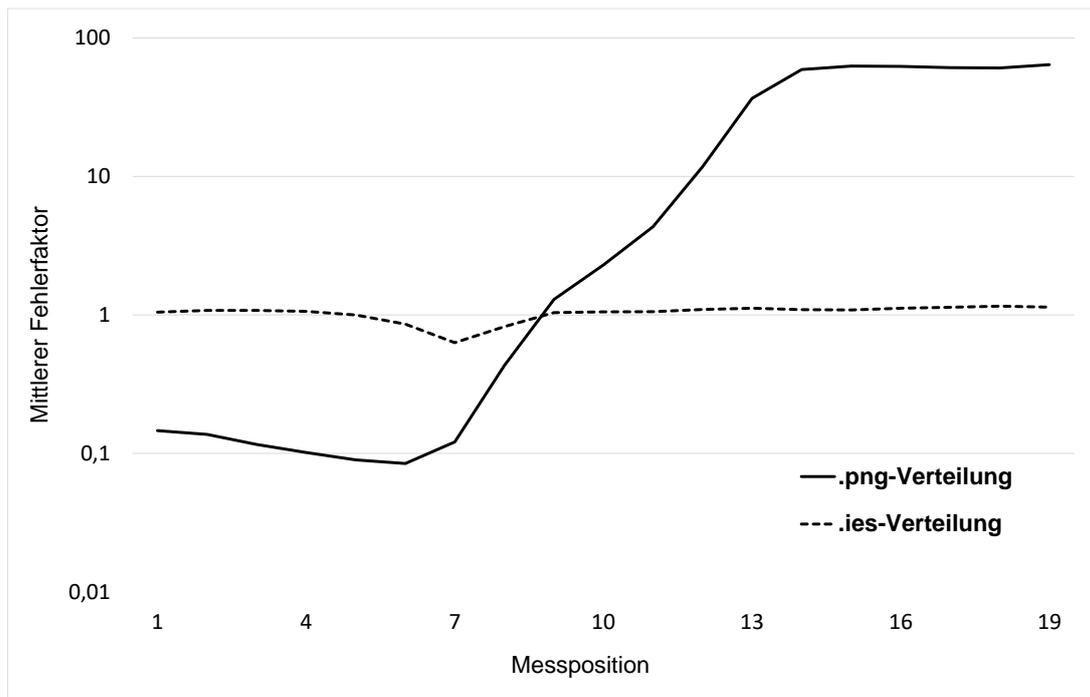


Abb. 6: Vergleich des vertikalen Fehlerfaktors für die .ies- und die .png-Lichtverteilung (logarithmische Skala)

Aus dem in Abbildung 5 und 6 gezeigten Vergleich zwischen der simulativen Abbildung von .png- und .ies-Lichtverteilung kann abgeleitet werden, dass die Verwendung von .ies-Lichtverteilungen gegenüber der .png-Lichtverteilung zu bevorzugen ist. Nichtsdestotrotz sind bei der Betrachtung des mittleren horizontalen Fehlerfaktors auch für die .ies-Verteilung Abbildungsfehler von bis zu 60% im Randbereich der Lichtverteilung festzustellen. Die größere Abweichung kann sowohl auf den Einfluss der vergrößerten Distanz zwischen Quelle und Detektor als auch auf die in Gleichung (4) gezeigte Berechnungsweise zurückgeführt werden. Da die absoluten Werte der Beleuchtungsstärke im Randbereich deutlich geringer sind als entlang der optischen Achse, hat der Dividend einen überproportionalen Einfluss auf den Fehlerfaktor, was durch die Berücksichtigung der Absolutwerte reduziert werden kann. Der vertikale mittlere Fehlerfaktor der .ies-Verteilung fällt lediglich im Bereich von Messposition 7 auf unter 1 ab, was auf mögliche Ungenauigkeiten in der Positionierung des simulierten Scheinwerfers zurückzuführen sein kann. Eine entsprechende Justage kann die lokale Verschlechterung des Fehlerfaktors reduzieren. Darüber hinaus ist die Anpassung der .ies-Lichtverteilung basierend auf den Messungen in der Simulationsumgebung denkbar.

5. Zusammenfassung und Ausblick

In der vorgestellten Forschungsarbeit wird eine geeignete Simulationsumgebung zur Abbildung von lichttechnischen Einflüssen auf kamerabasierte Objekterkennung ausgewählt. Um die lichttechnischen Einflüsse modellieren zu können, werden unterschiedliche Verfahren des Imports von gemessenen Lichtverteilungen vorgestellt und bewertet. Eine Methode zum Messen von simulierten Lichtverteilungen wird vorgestellt und die zuvor importierten Lichtverteilungen unter Zuhilfenahme der Messmethode mit den realen Messwerten verglichen. Dabei wird die Einbindung von .ies-Lichtverteilungen als optimale Methode festgestellt.

In weiterer Forschung ist der Unterschied zwischen Simulation und Realität zu reduzieren, was durch die Einführung einer iterativen Anpassung der Lichtverteilung auf Basis der er-

mittelten Fehlerwerte realisiert werden kann. Darüber hinaus können Validität und die Grenzen des Messverfahrens durch die Vermessung weiterer Lichtverteilungen erweitert werden.

Literaturverzeichnis

- [1] T. Glück u. a., „Distraction Potential of Vehicle-Based On-Road Projection“, *Applied Sciences*, Bd. 11, Nr. 24, Dez. 2021, doi: 10.3390/app112412030.
- [2] P.-P. Ley, M. Knöchelmann, G. Kloppenburg, und R. Lachmayer, „Development Methodology for Optomechatronic Systems Using the Example of a High-Resolution Projection Module“, *Proceedings of the Design Society: International Conference on Engineering Design*, Bd. 1, Nr. 1, S. 2547–2556, Juli 2019, doi: 10.1017/dsi.2019.261.
- [3] M. Waldner und T. Bertram, „Optimal Real-time Digitization of Matrix-Headlights“, in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Juli 2021, S. 31–36. doi: 10.1109/AIM46487.2021.9517704.
- [4] M. C. Sundermeier, H. Dierend, P.-P. Ley, A. Wolf, und R. Lachmayer, „Active NIR illumination for improved camera view in automated driving application“, März 2022, S. 25. doi: 10.1117/12.2608162.
- [5] M. C. Sundermeier, H. Dierend, und R. Lachmayer, „Verbesserung der kameragestützten Objekterkennung im Straßenverkehr durch nahinfrarote Zusatzbeleuchtung“, Juni 2022.
- [6] B. Foundation, „About“, *blender.org*. <https://www.blender.org/about/> (zugegriffen 30. Januar 2023).
- [7] „Unreal Engine | Features“, *Unreal Engine*. <https://www.unrealengine.com/en-US/features> (zugegriffen 30. Januar 2023).
- [8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, und V. Koltun, „CARLA: An Open Urban Driving Simulator“. arXiv, 10. November 2017. Zugegriffen: 30. Januar 2023. [Online]. Verfügbar unter: <http://arxiv.org/abs/1711.03938>
- [9] „CarMaker | IPG Automotive“. <https://ipg-automotive.com/en/products-solutions/software/carmaker/> (zugegriffen 30. Januar 2023).
- [10] „An end-to-end simulation platform“. <https://www.svl simulator.com/product/simulation/> (zugegriffen 30. Januar 2023).
- [11] https://github.com/ma-sunde/LMT_Lichverteilung-Graustufenbilder