Bildbasierte Situationsanalyse zur intuitiven Mensch-Roboter-Interaktion in dynamischen Umgebungen

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

von **M. Sc. Thorsten Hempel**

geb. am 18.10.1993 in Pinneberg

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik
der Otto-von-Guericke-Universität Magdeburg

Gutachter:

Prof. Dr.-Ing. habil. Ayoub Al-Hamadi

Prof. Dr.-Ing. Andreas Nürnberger

Prof. Dr.-Ing. Sebastian von Enzberg

Promotionskolloquium am 24. März 2025

Inhaltsverzeichnis

Kurzfassung

Mobile, intelligente Roboter helfen, die Produktivität, Präzision und Effizienz in der Industrie zu steigern, Arbeitsunfälle und Kosten zu reduzieren und tragen damit gleichzeitig zu einer umweltfreundlichen Ressourcenschonung bei. Zusätzlich birgt ihr Einsatz in medizinischen und sozialen Bereichen erhebliche Potenziale. Sie können die Zusammenarbeit von Hilfsbedürftigen und Helfenden unterstützen und so zur Steigerung der Lebensqualität beitragen. Für die Realisierung dieser Potenziale muss jedoch die intelligente Erfassung des semantischen Aktionsraums und der darin befindlichen menschlichen Interaktionspartner verbessert werden, um eine kontextbezogene und intuitive Mensch-Roboter-Interaktionen zu ermöglichen.

Die vorliegende Arbeit befasst sich mit der Entwicklung, Implementierung und Evaluierung bildbasierter Deep Learning-Methoden, die die soziale Autonomie mobiler Roboter verbessern und den Informationsgehalt zur Bestimmung adäquater Verhaltensstrategien erhöhen. Sie ist in mehrere wissenschaftliche Beiträge unterteilt, die sich auf die räumlich-semantische Umgebungsanalyse und die Analyse menschlicher Interaktionspartner konzentrieren.

Der erste wissenschaftliche Beitrag befasst sich mit der Orientierung mobiler Roboter in komplexen, dynamischen Umgebungen. Hierfür wird visueller SLAM (Simultaneous Localization and Mapping) mittels eines Deep Learning-basierten Szenen-Flows erweitert, wodurch eine pixelgenaue Erfassung dynamischer Bildelemente erzielt und eine signifikante Reduzierung des Trajektoriefehlers erreicht werden kann. Als Nächstes wird eine neue Methode zur semantischen Kartierung vorgestellt, bei der rein geometrische Umgebungskarten durch semantische Objekte erweitert werden. Dies verbessert das kontextuelle Verständnis der Umgebung und ermöglicht das Greifen und Transportieren von Objekten, während die kartierten Objekte gleichzeitig für die Optimierung der Trajektoriebestimmung einbezogen werden können.

Zur Analyse von Interaktionspartnern wird eine neue Methode zur Kopfposeschätzung vorgestellt, welche den gesamten Rotationsbereich abschätzen kann und in Robustheit und Genauigkeit den Stand der Technik übertrifft. Diese Methode wird im Anschluss mittels eines Multi-Task-Ansatzes mit einer Blickrichtungsschätzung kombiniert, um Synergien beider Aufgaben auszuschöpfen, welche zu einer Verbesserung der Generalisierungsfähigkeit des Modells, insbesondere für die Blickrichtungsschätzung, führt. Mithilfe eines zusätzlichen Modells wird sich der Detektion von Blickkontakt aus der Egoperspektive angenommen. Für diesen noch weitgehend unerforschten Bereich wird eine umfangreiche Datenbank erzeugt, mit deren Hilfe akkurate und robuste Prädiktionsmodelle erzeugt werden können, welche neben Kopfpose und Blickrichtung nonverbale Interaktionen mit menschlichen Kooperationspartnern verbessern.

Insgesamt trägt diese Arbeit zur Verbesserung der mobilen Mensch-Roboter-Interaktion bei, indem sie Lokalisierungsfehler in dynamischen Umgebungen reduziert, semantische Informationen in die Umgebungserfassung einbettet und Methoden zur Erfassung und Verarbeitung menschlicher Interaktionspartner entwickelt. Jede der vorgestellten Methoden ist dabei modular gestaltet, sodass sie sowohl isoliert als auch in anderen Applikationsbereichen eingesetzt werden können.

Abstract

Mobile, intelligent robots can enhance productivity and efficiency in industry, reduce workplace accidents and costs, and thereby contribute to environmentally friendly resource conservation. Additionally, their use in medical and social fields holds significant potential to support collaboration between those in need and caregivers, thus contributing to an improved quality of life.

This work focuses on the development, implementation, and evaluation of image-based deep learning methods aimed at improving the social autonomy of mobile robots and enhancing their information content for determining appropriate behavioral strategies. It is divided into several scientific contributions that concentrate on spatial-semantic environment perception and the analysis of human interaction partners.

The first contribution addresses the orientation of mobile robots in dynamic environments by extending visual SLAM (Simultaneous Localization and Mapping) with deep learning-generated optical flow into a scene flow. This enables fine, pixel-based capture of dynamic image elements and significantly reduces trajectory error. Next, a new method for semantic mapping is presented, where purely geometric environment maps are augmented with semantic objects. This enhances the understanding of the environment and enables the grasping and transporting of objects.

For the analysis of interaction partners, a new method for head pose estimation is introduced, which can analyze the entire range of rotation and surpasses the state of the art in robustness and accuracy. This method is subsequently combined with gaze estimation using a multi-task approach to exploit synergies between both tasks, leading to an improvement in the model's generalization ability, especially for gaze estimation. An additional model addresses gaze contact detection from an ego perspective. For this largely unexplored area, an extensive database is created, enabling the development of accurate and robust prediction models that improve non-verbal interactions with human cooperation partners by incorporating head pose and gaze direction.

Overall, this work contributes to the enhancement of human-robot interaction (HRI) by reducing localization errors in dynamic environments, embedding semantic information into environment perception, and developing methods for capturing and processing human interaction partners. Each of the presented methods is modular in design, allowing them to be used both in isolation and in other application areas.

Cobots (collaborative robots) are robots capable of interacting directly and safely with humans. Unlike conventional industrial robots, which often work in enclosed areas, cobots can be used in close proximity to humans. They are increasingly used in industry to automate physically demanding or monotonous tasks, thus increasing productivity, and also offer the possibility for use in other areas such as healthcare and even private use as personal assistants.

To fully exploit the potential of cobots, their abilities for autonomous navigation and interaction must be further improved. Special challenges lie in environment sensing and in the registration of nonverbal communication signals to enable efficient human-robot interactions without misunderstandings. This dissertations presents a series of new methods that optimize human-robot interaction (HRI) through image-based techniques. These include algorithms for reducing localization errors of mobile cobots

in dynamic environments, embedding semantic information into their environment sensing, and various methods for sensing and processing human interaction partners to enable more efficient and intuitive collaborations.

Abbildungsverzeichnis

Tabellenverzeichnis

Veröffentlichungen

1. Thorsten Hempel, Ahmed A. Abdelrahman, Ayoub Al-Hamadi. “Towards Robust and Unconstrained Full Range of Rotation Head Pose Estimation”. In: *IEEE Transactions on Image Processing*, März 2024,
   Impact Factor: 10,6

2. Thorsten Hempel, Ayoub Al-Hamadi. “An online semantic mapping system for extending and enhancing visual SLAM”. In: *Engineering Applications of Artificial Intelligence*, Volume 111, 2022, 104830, ISSN 0952-1976,
   Impact Factor: 8,635

3. Thorsten Hempel, Ayoub Al-Hamadi. “Pixel-Wise Motion Segmentation for SLAM in Dynamic Environments”. In: *IEEE Access 08:164521 - 164528*, September 2020,
   Impact Factor: 3,9

4. Thorsten Hempel, Magnus Jung, Ahmed A. Abdelrahman, Ayoub Al-Hamadi. “NITEC: Versatile Hand-Annotated Eye Contact Dataset for Ego-Vision Interaction”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 4437-4446.

5. Thorsten Hempel, Laslo Dinges, Ayoub Al-Hamadi. “Sentiment-Based Engagement Strategies for Intuitive Human-Robot Interaction”. In: *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP*, ISBN 978-989-758-634-7; ISSN 2184-4321, pages 680-686, 2023

6. Thorsten Hempel, Ahmed A Abdelrahman, Ayoub Al-Hamadi. ”6D Rotation Representation For Unconstrained Head Pose Estimation”. In: *IEEE International Conference on Image Processing (ICIP)*, Bordeaux, France, 2022, pp. 2496-2500.

7. Thorsten Hempel, Ayoub Al-Hamadi, “On contextual perception of workers in complex production environments”. In: *Engineering for a Changing World: Proceedings; 60th ISC, Ilmenau Scientific Colloquium, Technische Universit´at Ilmenau, September 04-08, 2023*, DOI: 10.22032/dbt.58931.

8. Thorsten Hempel, Ayoub Al-Hamadi. “SLAM-Based Multistate Tracking System for Mobile Human-Robot Interaction”. In: *Image Analysis and Recognition*, 368-376, Springer International Publishing, 2020

9. Thorsten Hempel, Marc-André Fiedler, Aly Khalifa, Ayoub Al-Hamadi, Laslo Dinges. “Semantic-Aware Environment Perception for Mobile Human-Robot Interaction”. In: *12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2021, pp. 200–203.

10. Laslo Dinges, Marc-André Fiedler, Ayoub Al-Hamadi, Thorsten Hempel, Ahmed Abdelrahman, Joachim Weimann, Dmitri Bershadskyy. “Automated Deception Detection from Videos: Using End-to-End Learning Based High-Level Features and Classification Approaches”, In: *Neural Computing and Applications*, 2024,
Impact Factor: 6

11. Basheer Al-Tawil, Thorsten Hempel, Ahmed Abdelrahman, Ayoub Al-Hamadi. “A review of visual SLAM for robotics: evolution, properties, and future applications”. In: *Frontiers in Robotics and AI*. 2024; 10.3389/frobt.2024.1347985,
Impact Factor: 3.4

12. Ahmed A Abdelrahman, Dominykas Strazdas, Aly Khalifa, Jan Hintz, Thorsten Hempel, Ayoub Al-Hamadi. “Multimodal Engagement Prediction in Multiperson Human–Robot Interaction”. In: *IEEE Access*, vol. 10, pp. 61980-61991, 2022,
Impact Factor: 3,9

13. Dominykas Strazdas, Jan Hintz, Aly Khalifa, Ahmed A. Abdelrahman, Thorsten Hempel, Ayoub Al-Hamadi. “Robot System Assistant (RoSA): Towards Intuitive Multi-Modal and Multi-Device Human-Robot Interaction”. In: *Sensors*. 2022; 22(3):923,
Impact Factor: 3.9

14. Aly Khalifa, Ahmed A. Abdelrahman, Thorsten Hempel, Ayoub Al-Hamadi. “Towards efficient and robust face recognition through attention-integrated multi-level CNN”. In: *Multimedia Tools and Applications*. 2024, doi: 10.1007/s11042-024-19521-0.
Impact Factor: 3,6

15. Aly Khalifa, Ahmed A Abdelrahman, Dominykas Strazdas, Jan Hintz, Thorsten Hempel, Ayoub Al-Hamadi. “Face Recognition and Tracking Framework for Human–Robot Interaction”. In: *Applied Sciences*. 2022; 12(11):5568.,
Impact Factor: 2,7

16. Laslo Dinges, Ayoub Al-Hamadi, Thorsten Hempel, Z Al Aghbari. “Using Facial Action Recognition to Evaluate User Perception in Aggravated HRC Scenarios”. In: *12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2022.

17. Ahmed Abdelrahman, Thorsten Hempel, Aly Khalifa, Ayoub Al-Hamadi, Laslo Dinges. “L2CS-Net : Fine-Grained Gaze Estimation in Unconstrained Environments”, In: *International Conference on Frontiers of Signal Processing (ICFSP)*, 2023.

18. Ahmed Abdelrahman, Thorsten Hempel, Ayoub Al-Hamadi. “MTGH: Multi-task Gaze and Head Pose Estimation in-the-Wild”, In: *Neural Computing and Applications*, (Im Peer-Review),
    Impact Factor: 6

19. Ahmed Abdelrahman, Thorsten Hempel, Aly Khalifa, Ayoub Al-Hamadi. “Fine-grained gaze estimation based on the combination of regression and classification losses”, In: *Applied Intelligence*, (Im Peer-Review),
    Impact Factor: 5,3

20. Arman Ahmed Khan, Thorsten Hempel, Ayoub Al-Hamadi. “Dynamic Hand Gesture Recognition for Human-Robot Interaction on Mobile Platforms”. In: *Frontiers in Robotics and AI*. (Im Peer-Review),
    Impact Factor: 3,4

21. Magnus Jung, Ahmed Abdelrahman, Thorsten Hempel, Basheer Al-Tawil, Qiaoyue Yang, Sven Wachsmuth, Ayoub Al-Hamadi. “Eye Contact Based Engagement Prediction for Efficient Human-Robot Interaction”. In: *International Journal of Social Robotics*, (Im Peer-Review),
    Impact Factor: 4,7

KAPITEL 1

Einleitung

Bei einer Cocktailparty im Jahr 1956 traf Joseph Engelberger den Erfinder George Devol. Die beiden kamen ins Gespräch über Georges neueste Erfindung – die *Programmed Article Transfer* Vorrichtung – einen programmierbaren, maschinellen Arm. „Das klingt für mich nach einem Roboter“, soll Engelbergers Entgegnung gewesen sein, der durch seine Liebe zu den Science-Fiction-Geschichten des Schriftstellers Isaac Asimov eine tiefe Faszination für Roboter hatte. Drei Jahre später, im Jahre 1959, installierten Engelberger und Devol in Kooperation mit General Motors den ersten Prototyp, welcher bereits 1961 in der Fertigungsstraße bei General Motors eingesetzt wurde. Der Prototyp trug den Namen »Unimate«, führte Schweißarbeiten an Karosserien durch und gilt heute als der weltweit erste industrielle Roboter.

In den darauffolgenden Jahren wurden auch in Japan und Deutschland erste Roboter in der Automobilproduktion eingesetzt. Mit ihnen trat eine bis heute andauernde technische Revolution in der Automatisierung von Prozessen ein. Ihre Präzision, Geschwindigkeit und Kontinuität in der Ausführung ermöglichen die Steigerung der Produktivität, Qualität und Effizienz von Fertigungsprozessen. Dabei ist die Automobilbranche schon lange nicht mehr alleiniger Profiteur automatisierter Systeme. Weitere Anwendungsbereiche umfassen die Elektronikbranche zur Bestückung von Leiterplatten, zur Inspektion und zur Handhabung empfindlicher Komponenten, die Lebensmittel- und Getränkeindustrie bei der Lebensmittelverarbeitung, Verpackung und Palettierung und dem Gesundheitswesen, wie der Roboter-assistierten Chirurgie, Rehabilitation und Pflege. Weiterhin werden Roboter auch in der Chemie- und Pharmaindustrie, Bauindustrie sowie Luft- und Raumfahrt eingesetzt. Aufgrund der kontinuierlichen technischen Weiterentwicklung können die Systeme ihre bestehenden Fähigkeiten stetig ausbauen und optimieren und mit zugewonnenen Kompetenzen neue Anwendungsbereiche erobern.

Seit den 2010er Jahren manifestiert sich eine neue Art von Robotern. Zuvor war der Einsatz und infolgedessen die Eigenschaften von Robotern überwiegend auf repetitive Produktionsprozesse

am Fließband spezialisiert, für die leistungsstarke, stationäre Roboter bestens geeignet sind. Die Zusammenarbeit von Mensch und Roboter in gemeinsamen Arbeitsprozessen spielte hierbei eine untergeordnete Rolle, welche sich durch sicherheitsbedingte strikt getrennten Arbeitsräumen ausprägte. Der Einzug von Robotern in neue Anwendungsbereiche führte jedoch zu einem Wandel in ihrem Anforderungsprofil. Der Einsatz beispielsweise im Bereich der Spezialanfertigung erfordert mehr Flexibilität, vor allem für jene Prozesse, in denen individuelle Entscheidungen, Flexibilität und Mobilität benötigt werden. Für diese Arbeiten werden kollaborative Roboter (eng. *collaborative robots – Cobots* -) benötigt.

Cobots sind darauf ausgerichtet, mit Menschen sicher in einem gemeinsamen Arbeitsraum zu agieren und zusammenzuarbeiten. Ziel ist es, den Menschen als Kontrollinstanz und Dirigent zurück ins Zentrum der Prozessabläufe zu bringen, welcher die Cobots gezielt und bedarfsweise steuern, koordinieren und kontrollieren. Dies erhöht die Flexibilisierung von Arbeitsprozessen und stärkt die symbiotischen Fähigkeiten von Mensch und Maschine. Durch ihre geringe Größe sind Cobots zudem günstiger als größere Industrieroboter und können auf mobilen Plattformen installiert werden. Obwohl der Markt für kollaborative Roboter immer noch einen kleinen Teil (7,5%) des gesamten Industrierobotikmarktes ausmacht, erfährt er ein ungleich höheres Wachstum [1]. Im Jahre 2023 beträgt die Marktgröße von Cobots bereits 1,2 Milliarden USD. Bis 2029 wird ein Marktumsatz bei einer jährlichen Wachstumsrate von knapp 35% von 6,8 Milliarden USD erwartet [2]. Bis dahin müssen jedoch noch elementare Fähigkeiten zur Automatisierung und Autonomie der maschinisierten Assistenten entwickelt und verbessert werden, um langfristig eine produktive, flexible und intuitive gemeinsame Arbeitswelt von Mensch und Roboter zu schaffen.

1.1 Motivation

Die gewünschten Eigenschaften mobiler Cobots bringen eine Vielzahl neuer Herausforderungen an die technische Umsetzung mit sich. Die Mobilität ermöglicht es, den Cobots ihren Einsatzort flexibel zu wechseln, sodass Auslastung und Applikabilität gesteigert und die Anzahl an Einsatzprofilen erweitert werden. Hierbei muss die autonome Navigation eine inhärente Fähigkeit des Cobots sein, welche unabhängig des Einsatzortes und dessen Rahmenbedingungen erfolgen kann. Dazu müssen die notwendigen technischen Fertigkeiten für mobile Systeme geschaffen werden, um sich eigenständig in komplexen Umgebungen zu orientieren und Zielpositionen kollisionsfrei anzufahren.

Bei der autonomen Navigation sind Cobots mit zusätzlichen, herausfordernden Umgebungsbedingungen konfrontiert. Während herkömmliche Industrieroboter üblicherweise in eigens für sie vorbehaltenen Schutzzonen operieren, teilen mobile Cobots ihren Aktionsraum gemeinsam mit Menschen und anderen Robotern. Sie müssen deshalb mit zusätzlichen Sicherheitsmechanismen ausgestattet werden, um die eigene und die Unversehrtheit der Umgebung zu gewährleisten. Gleichzeitig führt der geteilte Arbeitsraum zu einer dynamischen, ständig wechselnden Einsatzumgebung. Diese hat nicht nur Einfluss auf die Genauigkeit der Orientierung und die Navigation des Roboters, sondern wirkt sich auch erschwerend auf die gesamtheitliche Erfassung und Verarbeitung seiner Umwelt aus. Ohne abgeschlossenen Arbeitsraum können Arbeitsgeräte, Werkzeuge und andere Objekte durch Dritte bewegt werden. Somit muss der Cobot zusätzlich in der Lage sein, seine

Umgebung kontinuierlich zu erfassen, zu kategorisieren und ein konsistentes Verständnis seiner Umwelt aufzubauen, um in ihr und mit ihr interagieren zu können.

Eine weitere Herausforderung liegt in der Mensch-Roboter-Kollaboration selbst. Hierfür muss der Roboter mit den nötigen Fähigkeiten ausgestattet sein, den Menschen selbstständig wahrzunehmen und mit ihm zu kommunizieren. Die Art der Kommunikation soll hierbei auf menschliche Kommunikationsmechanismen zugeschnitten werden, um dem Menschen als Zentrum der Interaktion einen möglichst intuitiven Zugang zu gewähren. Eine zentrale Herausforderung liegt hierbei in der Erfassung von nonverbalen Kommunikationssignalen, welche implizit durch Körperpose und Verhalten in zwischenmenschlichen Interaktionen einen wichtigen Teil des Informationsaustausches übernehmen, und für eine effiziente und präzisere Kommunikation sorgen [3]. Die visuelle Erfassung dieser Signale ist damit eine essenzielle Aufgabe für Roboter, um die Zusammenarbeit mit Menschen langfristig intuitiver und produktiver zu gestalten [4, 5].

1.2 Herausforderungen und Ziele der Arbeit

Diese Dissertation befasst sich mit der Verbesserung der bildbasierten Wahrnehmung für mobile Roboter, um ihren Einsatz in einer dynamischen und menschenzentrierten Arbeitsumgebung langfristig intuitiver und flexibler zu gestalten. Hierzu werden Methoden, insbesondere aus dem Bereich Deep Learning, entwickelt und evaluiert, die es dem Roboter ermöglichen, elementare Informationen seiner Umgebung und menschlichen Interaktionspartnern zu erfassen und produktivere Kollaborationsprozesse zu schaffen.

Die Ausgangslage bildet die Herausforderungen einer Einsatzumgebung, wie sie unter realen Bedingungen wiederzufinden sind. Dazu gehören Fertigungs- und Produktionshallen, Werkstätten sowie Großraumkomplexe wie Krankenhäuser. Sie alle verbinden drei essenzielle Herausforderungen:

- Dynamische Umgebungen (z.B. Personen, Geräte),
- die Arbeit mit Objekten (z.B. Werkzeug, Alltagsgegenstände),
- und die Interaktion mit Menschen.

Abbildung 1.1 illustriert ein solches realitätsnahes Szenario, bei dem ein Roboter gleichzeitig mit all diesen Herausforderungen konfrontiert wird. Zum einen umfasst die Umgebung Objekte, welche zum Erreichen von Arbeitszielen als Werkzeug oder Arbeitsmittel eingesetzt werden müssen. Gleichzeitig sind der Austausch und die Interaktion mit Menschen wesentlicher Bestandteil, um Aufgaben entgegenzunehmen, Informationen auszutauschen oder gemeinsame Arbeitsprozesse durchzuführen. Dabei sollten für die Interaktion möglichst natürliche und für den Menschen intuitive Kommunikationsstrategien eingesetzt werden. Alle Vorgänge finden in einer dynamischen Umgebung statt, in der z. B. Personen den Arbeitsbereich durchlaufen oder andere räumliche Veränderungen stattfinden, welche die Orientierung und Autonomie des Roboters nicht beeinträchtigen dürfen.

Das Ziel dieser Arbeit liegt in der Entwicklung und Evaluierung von bildverarbeitenden Methoden, um eine intuitive und kontextsensitive mobile Mensch-Roboter-Interaktion zu ermöglichen und so

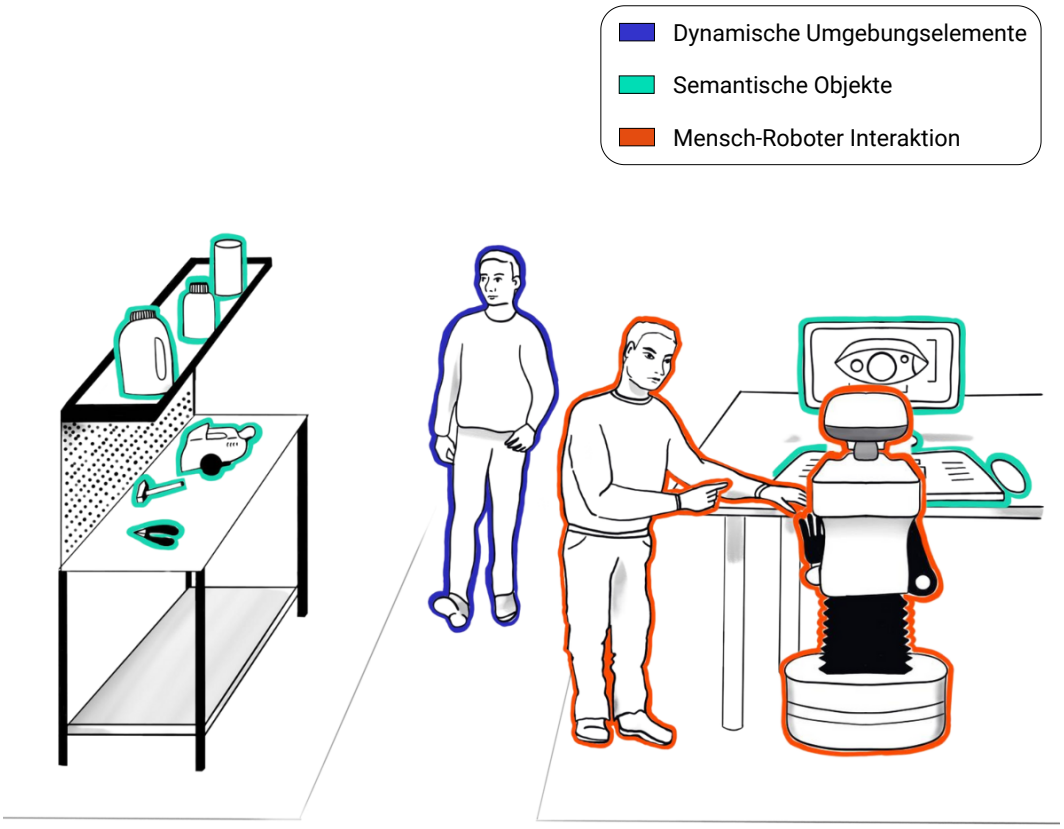


Abbildung 1.1: Beispielszenario eines Roboters in einer interaktiven Einsatzumgebung. Die Schlüssel-Herausforderungen bestehen in dynamischen Umgebungselementen, dem Einbezug von Objekten in den Einsatzkontext und der intuitiven Interaktion mit Menschen.

den produktiven Einzug intelligenter Roboter in ein solches Einsatzumfeld zu ebnen. Strategisch liegt hierbei der Fokus auf der Adaption der Fähigkeiten des Roboters auf die Bedürfnisse und Verhaltensweisen des Menschen, insbesondere der Erfassung menschlicher Kommunikationssignale, um dadurch Potenziale für intuitive und effiziente Zusammenarbeit zu generieren.

Hierfür ist diese Arbeit in Form eines *Bottom-Up*-Ansatzes strukturiert, welcher in Abbildung 1.2 illustriert ist. Grundlegendes Ziel ist die Ermöglichung intuitiver und kontextsensitiver Interaktionen für mobile Roboter. Hierfür benötigt es Methoden zur robusten Orientierung mobiler Roboter, die selbst in komplexen, dynamischen Umgebungen einsatzfähig sind. Als Nächstes müssen die Umgebungskarten in ihrer reinen räumlichen Ausprägung mit einer semantischen Ebene erweitert werden, in der Objekte und ihre Bedeutung erkannt und räumlich eingeordnet werden können. Die robuste räumlich-semantische Umgebungserfassung ermöglicht bereits eine über die reine Navigation hinaus gehende Interaktion mit der Umwelt, durch das Suchen, Greifen und Bewegen von Objekten. Die Methoden hierfür werden als “Umgebungsanalyse” zusammengefasst.

Für die Interaktion mit Menschen müssen darüber hinaus jedoch weitere Methoden eingesetzt werden, welche sich in der Kategorie “Personenanalyse” zusammenfassen. Sie beinhaltet Ansätze, mit deren

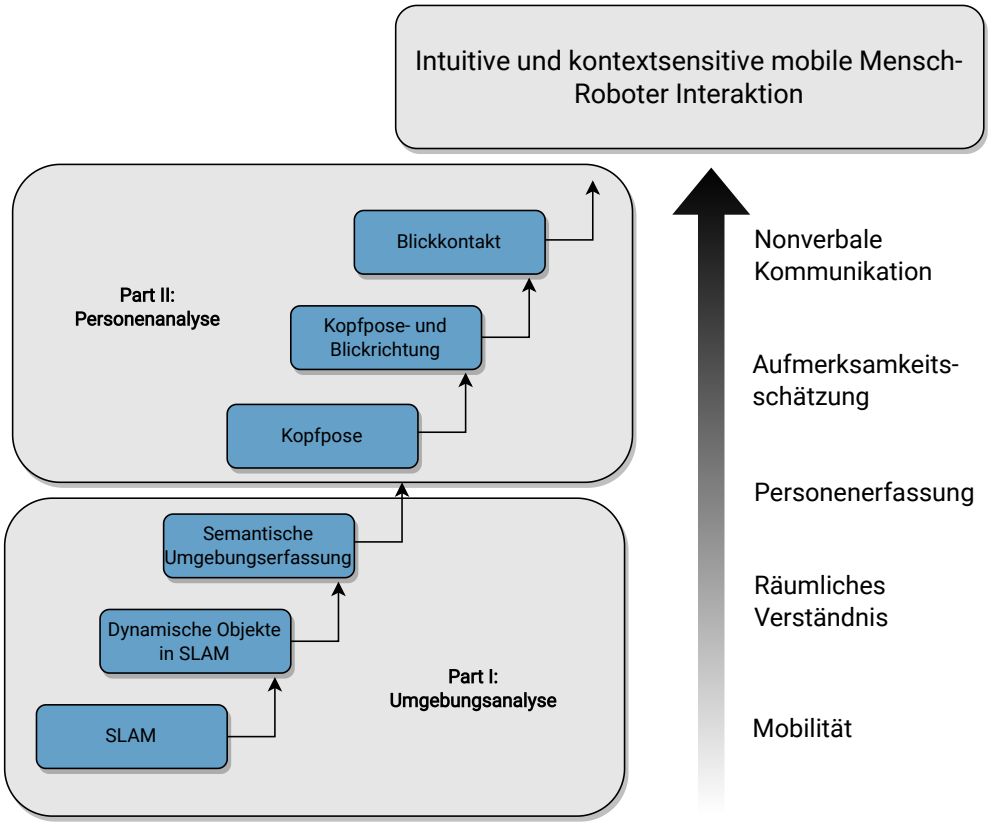


Abbildung 1.2: Übersicht der Bottom-Up-Strategie zur intuitiven, kontextsensitiven und mobilen Mensch-Roboter Interaktion.

Hilfe die potenziellen Interaktionspartner erfasst und ihre Aufmerksamkeit eingeschätzt werden kann.

Die vorgeschlagenen Ansätze werden dabei modular entwickelt, sodass sie nicht nur als Gesamtsystem, sondern losgelöst in isolierter Form für unterschiedlichste Anwendungszwecke einsetzbar sind. Dies stärkt den wissenschaftlichen Beitrag dieser Arbeit und erweitert die Anwendbarkeit für ein breiteres Spektrum an Applikationen.

1.3 Gliederung und wissenschaftlicher Beitrag der Arbeit

Die vorliegende Arbeit untergliedert sich in mehrere Teile und Kapitel, über die im Folgenden eine kurze Übersicht gegeben wird.

Kapitel 2 Das Kapitel beschreibt die Grundlagen neuronaler Netze zur lernbasierten Bildverarbeitung und zeigt die wichtigsten Architekturen für Deep Learning basierte Computer Vision.

Teil I: Umgebungserfassung

Die Kapitel, die sich in diesem Teil einordnen, befassen sich mit der robusten autonomen Orientierung und der kontextsensitiven Erfassung der Umgebung eines mobilen Roboters.

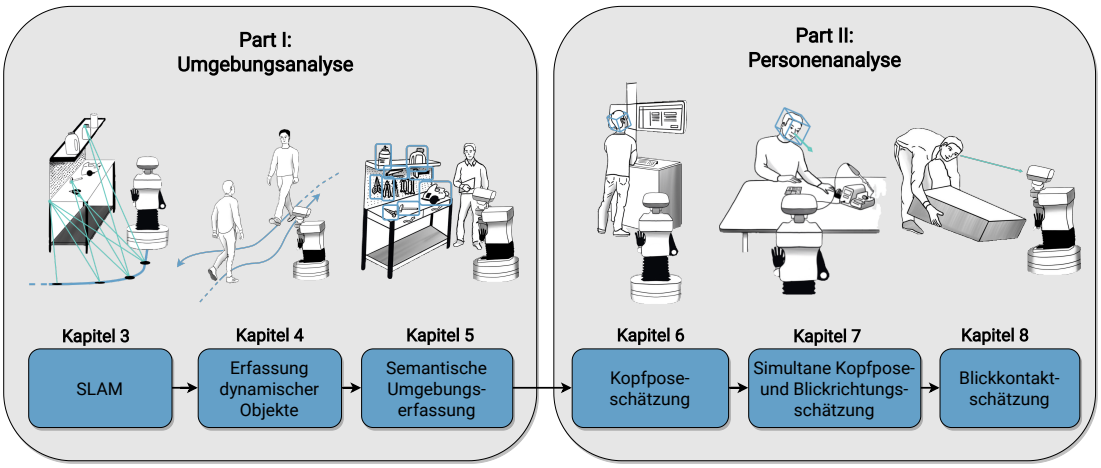


Abbildung 1.3: Gliederung der wissenschaftlichen Beiträge dieser Arbeit.

Kapitel 3 In diesem Kapitel werden die Grundlagen, Komponenten und der methodische Ablauf eines typischen Ansatzes zur simultanen Lokalisierung und Kartierung (SLAM) erläutert, welche ein mobiler Roboter nutzt, um sich in unbekannten Umgebungen zu orientieren und eine Karte des Umfelds aufzubauen. Zudem werden die Herausforderungen und Limitationen diskutiert, welche den größten Einfluss auf die Genauigkeit und die Robustheit von SLAM haben.

Kapitel 4 Dieses Kapitel befasst sich mit der Reduzierung von Lokalisierungsfehlern mobiler Roboter in Umgebungen mit dynamischen Störobjekten. Es wird eine Methode vorgeschlagen, die auf einer Deep Learning gestützten Erfassung eines Szenenflusses basiert, welche im Anschluss auf auffällige Bewegungsvektoren untersucht wird. Dadurch lassen sich dynamische Bildbereiche pixelgenau und bildspezifisch segmentieren und gleichzeitig einen maximal großen Referenzbereich erhalten. Dadurch verringert sich der Lokalisierungsfehler in Testsequenzen um bis zu 98%.

Kapitel 5 Das letzte Kapitel zur Umgebungsanalyse führt eine neue Methode zur semantischen Erweiterung von Kartierungsmethoden ein, die in Echtzeit Objektinstanzen in die Umgebungskarte registriert. Hierzu wird ein bildbasierter Objektdetektor verwendet, dessen Prädiktionen in den Raum projiziert werden, um Gegenstände (semantische Objekte) in die Umgebungskarte zu lokalisieren und einzubetten. Durch eine effiziente Datenassoziation ist der Ansatz besonders ressourcenschonend und für den Einsatz auf mobilen Plattformen geeignet.

Teil II: Personenanalyse

In diesem Teil befassen sich die Kapitel mit Methoden zur visuellen Erfassung und Interpretation von menschlichen Interaktionspartnern.

Kapitel 6 In diesem Kapitel wird eine neue, effiziente Methode zur bildbasierten Kopfposeschätzung entwickelt und evaluiert, um den Fokus der Aufmerksamkeit von Interaktionspartnern ermitteln zu können. Der entwickelte Ansatz zielt auf die verbesserte

Lernfähigkeit von Rotationen für neuronale Netze ab, wodurch besonders effiziente und robuste Modelle zur Kopfposeprädiktion erzeugt werden können und eine signifikante Verbesserung gegenüber dem Stand der Technik erreicht werden kann. Die Methode ist zudem beim Lernen im Rotationsbereich unbegrenzt, sodass mithilfe eines kombinierten Datensatzes erstmalig die Prädiktion des gesamten Rotationsbereichs des Kopfes ermöglicht wird.

Der in diesem Kapitel erzeugte Quellcode wurde zur vereinfachten Nutzung und zur besseren Reproduzierbarkeit der Öffentlichkeit zugänglich gemacht.[1][2]

Kapitel 7 Das anschließende Kapitel befasst sich mit der Erweiterung der Kopfposeschätzung hin zu einem Modell zur simultanen Prädiktion von Kopf- und Blickrichtung. Dabei werden die synergetischen Abhängigkeiten beider Aufgaben in einem neuronalen Netz vereint, um die Generalisierungsfähigkeit des Modells insbesondere für die Blickrichtungsschätzung zu verbessern und gleichzeitig den Rechenaufwand für mobile Anwendungen zu optimieren.

Kapitel 8 Dieses Kapitel befasst sich mit dem Sonderfall, bei dem der visuelle Fokus der Aufmerksamkeit direkt einem interagierenden Roboter oder einer Person gewidmet ist. Dieser Fall des Blickkontakts (bzw. Augenkontakts) ist ein elementarer Baustein nonverbaler Interaktion, der in der Forschung bisher jedoch nur wenig Beachtung erfahren hat. Zum Aufbau einer wissenschaftlichen Grundlage wird deshalb ein Datensatz zur Blickkontaktdetektion aus der Ego-Perspektive aufgebaut und evaluiert. Anschließend werden auf Basis des Datensatzes Modelle zur Blickkontaktdetektion erzeugt, deren Genauigkeit und Robustheit den Stand der Technik übertreffen und den herausfordernden Rahmenbedingungen einer praxisorientierten Probandenstudie standhalten.

Der Datensatz und die verwendeten Modelle wurden zur vereinfachten Nutzung in zukünftigen Applikationen und Forschungsvorhaben der Öffentlichkeit zugänglich gemacht.[3]

Kapitel 9 Das letzte Kapitel gibt eine Zusammenfassung der gesamten Arbeit und ihrer wissenschaftlichen Beiträge und zeigt einen perspektivischen Ausblick auf zukünftige Ansätze.

Anmerkung: *Diese Arbeit basiert auf den wissenschaftlichen Beiträgen mehrerer eigener Publikationen (insbesondere [6], [7], [8], [9], [10], [11], [12], [13], [14]). Die Inhalte dieser Arbeiten wurden in modifizierter Form in den Hauptkapiteln dieser Dissertation aufgenommen und mit zusätzlichen Informationen zum Stand der Technik, Methodik und Evaluation erweitert. Die wissenschaftlichen Arbeiten sind primär im Rahmen von zwei Projekten des Bundesministeriums für Bildung und Forschung (BMBF) (RoboAssist 03ZZ0448G-L, AutoKoWAT 13N16336 [4]) entstanden.*

[1] https://github.com/thohemp/6DRepNet
[2] https://github.com/thohemp/6DRepNet360
[3] https://github.com/thohemp/nitec
[4] https://autokowat.github.io/

KAPITEL 2

Grundlagen

Künstliche neuronale Netze (KNN) haben in den vergangenen Jahren erhebliche Fortschritte erzielt und erreichen im Bereich des maschinellen Sehens Menschen-ebenbürtige Erkennungsraten [15, 16]. Daher fokussiert sich diese Arbeit auf die Verbesserung und Entwicklung neuer, auf KNN basierender Methoden, um ihre Potenziale den Fähigkeiten der Cobots zugänglich zu machen. Als Grundlage gibt dieses Kapitel eine kurze Zusammenfassung der wichtigsten Konzepte von KNN.

2.1 Künstliche neuronale Netze

Das grundlegende Ziel von KNN ist die Zuordnung von Eingangsvariablen (z. B. ein Bild) zu Ausgangsvariablen (z. B. eine Objektklasse). Dieses Prinzip kann durch eine simple mathematische Gleichung beschrieben werden:

$$y = mx + c. \tag{2.1}$$

Das x beschreibt die Eingangsvariable, y die Ausgangsvariable, m und c sind Parameter, welche x in Abhängigkeit der Zielvorgabe des Modells interpretieren. In vielen Aufgaben wird y jedoch nicht nur durch einen einzigen Input definiert, sondern durch mehrere. Dient das Modell beispielsweise zur Klassifizierung einer Frucht, müssen mehrere Attribute zur Klassifizierung herangezogen werden (z. B. Form und Farbe), um eine Orange von einem Apfel unterscheiden zu können. Hierfür muss die obige Gleichung zu

$$y = \sum_{i=1}^{n} (x_i m_i) + c \tag{2.2}$$

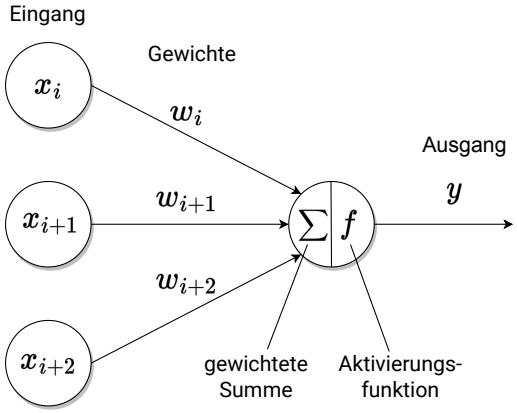


Abbildung 2.1: Aufbau eines künstlichen Neurons.

erweitert werden, in der n unterschiedliche Eingangsvariablen individuell gewichtet und dann aufsummiert werden. Diese Gleichung lässt sich noch weiter vereinfachen, wenn angenommen wird, dass es eine weitere Eingangsvariable mit dem Wert 1 gibt. Die zugehörige Gewichtung entspricht dabei dem y-Achsenabschnitt c. Dadurch ergibt sich die Gleichung

$$y = \sum_{i=0}^{n} (x_i m_i), \tag{2.3}$$

in welcher der Index i aufgrund des zusätzlichen Inputs nun bei 0 beginnt.

In künstlichen neuronalen Netzen wird dieses Modell als künstliches Neuron [17] modelliert, das im Aufbau der menschlichen Nervenzelle nachempfunden ist und den Prinzipien aus der Gleichung 2.3 folgt. Abbildung 2.1 zeigt eine grafische Darstellung eines solchen künstlichen Neurons, welches über mehrere Eingänge (Inputs) verfügt, die unter Berücksichtigung der Gewichtungen $w_{i:n}$ aufsummiert werden, um den Output y zu definieren. Auch hier beginnen die Inputs mit dem Index $i = 0$, bei dem der Input $x_0 = 1$ entspricht. Die Gewichtung w_0 wird bei künstlichen Neuronen auch *Bias* genannt.

In der Abbildung des künstlichen Neurons wird die gewichtete Summe zusätzlich durch die Funktion f verarbeitet. Diese Funktion wird Aktivierungsfunktion genannt und hat die Aufgabe, komplexere Ausgangssignale modellieren zu können. Im Vergleich zur menschlichen Nervenzelle entscheidet dieser Teil der Zelle darüber, ob ein Feuersignal weitergegeben werden soll oder nicht. Die Aktivierungsfunktion eines künstlichen Neurons kann darüber hinaus festlegen, in welchem Wertbereich die Ausgangsvariable y liegen darf. Welche unterschiedlichen Arten von Aktivierungsfunktionen es gibt und welche Eigenschaften sie aufweisen, wird in Abschnitt 2.1.2 genauer beschrieben.

Ein künstliches Neuron, wie in Abbildung 2.1, entspricht der einfachsten Form eines neuronalen Netzes und wird Perzeptron genannt. Mittels eines einzelnen Perzeptrons lassen sich elementarische Funktionen wie ein AND-, OR- oder NOT-Gatter implementieren. Komplexere Zusammenhänge wie das XOR-Gatter hingegen entsprechen einem nicht linearen Zusammenhang und können nicht durch ein einzelnes Perzeptron modelliert werden. Für komplexere Lösungsansätze müssen mehrere

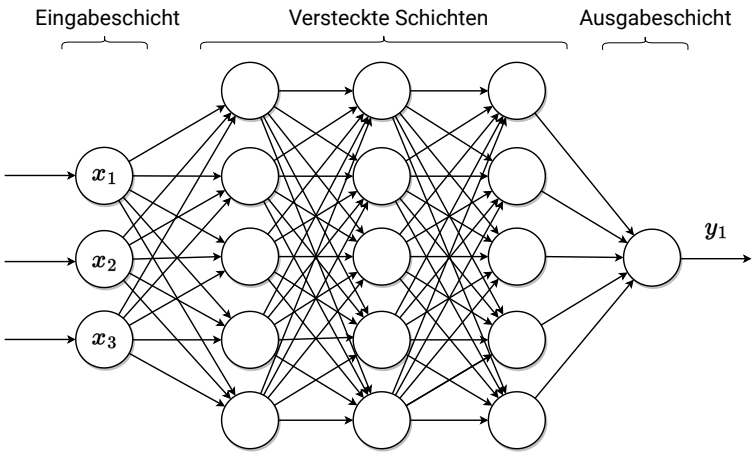


Abbildung 2.2: Aufbau eines künstlichen neuronalen Netzes.

Perzeptrons miteinander verknüpft werden. Hierfür wird der Output eines Neurons zum Input eines nachfolgenden Neurons. Es entsteht ein Netz von Neuronen, von dem die neuronalen Netze auch ihren Namen haben. Abbildung 2.2 zeigt ein Beispiel eines künstlichen neuronalen Netzes aus mehreren Neuronen, welches auch als Multi-Layer Perzeptron bezeichnet wird. Jeder Knoten entspricht einem Neuron. Das Netz besteht aus einer Eingabeschicht mit drei Neuronen, drei verstecken Schichten (engl. *hidden layers*) mit jeweils fünf Neuronen und einer Ausgabeschicht mit einem Neuron. Jedes Neuron ist mit allen Neuronen der vorherigen und allen Neuronen der nachfolgenden Schicht verbunden. Daher werden solche Schichten auch als *Fully Connected Layer* (auch *Dense Layer*) bezeichnet.

2.1.1 Parametrisierung durch Backpropagation

Damit ein neuronales Netz abhängig von den Eingangsdaten einen sinnvollen Ausgangswert generieren kann, müssen die Parameter des Modells, die Gewichte (und Biases) der Neuronen, sinnvoll konfiguriert werden. Die Parametrisierung wird bei künstlichen neuronalen Netzen durch ein iteratives Lernverfahren ermittelt, welches durch das Prinzip der Fehlerrückführung einen zufällig initialisierten Zustand hin zur gesuchten Parametrisierung optimiert. Grundlage für die Ermittlung der Gewichtskorrekturen ist die Kostenfunktion (engl. *loss function*), die die Distanz zwischen prädiziertem Ausgangswert und dem zugrunde liegenden Wahrheitswert (engl. *ground truth*) berechnet. Durch sie lässt sich ableiten, wie weit die Parametrisierung von der Erzeugung gewünschter Ausgabewerte entfernt ist und wie stark die Parameter korrigiert werden müssen. Über das Gradientenabstiegsverfahren oder ähnliche Optimierierungsverfahren werden damit die Gewichte und Biases iterativ angepasst, um die Kostenfunktion zu minimieren.

Die Gleichung 2.4 beschreibt, wie die Gewichte $\theta = \{w_0, w_1, w_2, ..., w_n\}$ durch das Gradientenabstiegsverfahren bestimmt werden. Dabei wird der Gradient berechnet, der abhängig von der Lernrate α die Größe und Richtung des Schrittes vorgibt, der in Richtung Minimum unternommen wird. Dieser Prozess erfolgte stufenweise von der Ausgangsschicht startend zurück zur Eingabeschicht und wird auch Backpropagation genannt.

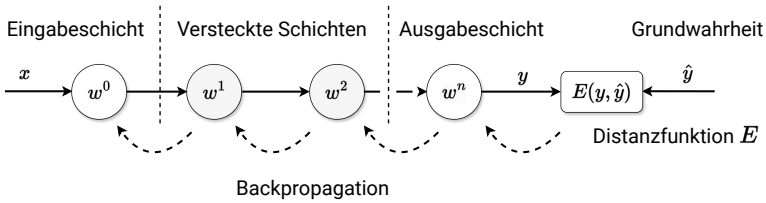


Abbildung 2.3: Prinzip des Forward- und Backward-Passes eines neuronalen Netzes.

$$\theta = \theta - \alpha \Delta E(y, g) \tag{2.4}$$

Abbildung 2.3 veranschaulicht dieses Prinzip, bei dem zunächst eine Prädiktion durchgeführt wird (engl. *Forward-Pass*) und anschließend auf Basis der Kostenfunktion über den Backward-Pass das Gewichtsupdate vorgenommen wird. Der sich iterierende Prozess aus der Verarbeitung von Eingangsdaten zur Erzeugung der Ausgangsvariablen, der Berechnung der Kosten und der Korrektur der Gewichte wird als *Training* bezeichnet und wird so lange wiederholt, bis eine Halte-Bedingung erfüllt ist.

2.1.2 Aktivierungsfunktionen

Eine Aktivierungsfunktion bestimmt, ob ein Neuron aktiviert (abgefeuert) werden soll oder nicht, abhängig von seiner Relevanz für die Zielfunktion des Modells. Zudem lassen sich durch nichtlineare Aktivierungsfunktionen komplexe Prädiktionsaufgaben modellieren. Im Folgenden wird eine kurze Übersicht über die gängigsten Aktivierungsfunktionen für neuronale Netze gegeben.

Sigmoid: Die Sigmoidfunktion modelliert die Ausgabewerte im Bereich zwischen [0,1] und wird deshalb oft genutzt, um die Wahrscheinlichkeit von unabhängigen Zuständen (z. B. Binär- und Multi-Label-Klassifikation) zu prädizieren. Aufgrund des Sättigungseffektes im stark negativen und stark positiven Bereich birgt die Sigmoidfunktion die Gefahr von *"vanishing gradients"*, das sich negativ auf den Lernprozess auswirkt.

$$a(x) = \frac{1}{1 + e^{-x}} \tag{2.5}$$

Softmax: Auch die Softmaxfunktion prädiziert Ausgaben im Wertbereich von [0,1]. Da sich die Gesamtwahrscheinlichkeit auf alle Ausgangswerte aufteilt, wird diese Funktion insbesondere für voneinander abhängige Mehrklassenprobleme eingesetzt.

$$a(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.6}$$

ReLU: Die Rectified Linear Unit (ReLU) setzt alle negativen Eingangswerte auf 0, während alle positiven Werte linear auf die Ausgangswerte übertragen werden. Diese Funktion wird insbesondere zwischen versteckten Schichten eingesetzt, um overfitting Prozesse zu regulieren. Da

positive Werte durch die lineare Übertragung nicht sättigen, besteht bei der ReLU-Funktion auch keine Gefahr von vanishing gradients.

$$a(x) = max(x, 0) \tag{2.7}$$

2.2 Convolutional Neural Network (CNN)

Die in Abschnitt 2.1 vorgestellte neuronale Netzarchitektur aus Fully Connected Layern bildet das klassische Beispiel eines KNN ab. Für die Verarbeitung von Bildern sind sie jedoch suboptimal, da sie aufgrund der großen Menge an Eingangsneuronen schlecht skalieren. Bei einem $200x200$ Pixel großen Bild, bestehend aus drei Kanälen (rot, grün, blau), hätte die Eingangsschicht bereits mehr als 120.000 Gewichte zum Trainieren. Mit wachsender Anzahl an Schichten würde so eine Architektur rasch unvorteilhafte Größen annehmen. In der klassischen Bildverarbeitung werden deshalb markante Bildinformationen in einem vorgelagerten Prozess durch statistische Verfahren aus dem Bild extrahiert. Dadurch erfolgt eine Reduktion der Eingangsdaten und somit der Komplexität des Netzes. Es muss jedoch bekannt sein, welche Merkmale für die jeweilige Aufgabe relevant und gleichzeitig auch in den Bilddaten vorhanden sind. Diese Kombination kann sich je nach Applikationsumgebung ändern und bietet deshalb keine robusten, generalisierbar zuverlässigen Ergebnisse.

Convolutional Neural Networks (CNN) sind eine spezielle Art von neuronalen Netzen, die besondere Vorteile für die Verarbeitung von Bildern mit sich bringen [18]. Sie ermöglichen es, die Erkennung sowie die Extraktion der relevanten Merkmale in den Lernprozess zu inkludieren. Im Gegensatz zu normalen neuronalen Netzen sind die Schichten von CNN volumetrisch aufgebaut und besitzen, angelehnt am technischen Aufbau eines RGB-Bildes, die drei Dimensionen Breite, Höhe und Tiefe. Die Neuronen zwischen den Schichten sind dabei nicht vollständig miteinander verbunden. Stattdessen fassen Neuronen ganze Regionen aus vorherigen Schichten zusammen. Dabei kommen typischerweise drei Arten von Schichten mit unterschiedlichen Wirkmechanismen zum Einsatz: *Convolutional Layer*, *Pooling Layer* und *Fully Connected Layer*.

Convolutional und Pooling Layer werden abwechselnd zum Extrahieren wichtiger Bildinformationen und zum Zusammenfassen von Merkmalsregionen eingesetzt, während ein oder mehrere Fully Connected Layer am Ende des Netzes die extrahierten Merkmale im Sinne der vorgegebenen Aufgabe interpretieren. Abbildung 2.4 illustriert den strukturellen Aufbau eines CNN. Zwischen dem letzten Pooling Layer und dem ersten Fully Connected Layer wird außerdem ein Flatten Layer eingesetzt. Diese Schicht ist lediglich eine Umstrukturierung des 3D-Tensors des letzten Pooling Layers, bestehend aus mehreren aufeinander gestapelten 2D Merkmalsschichten, zu einem 1D-Tensor, um die Verbindung mit dem nachfolgenden Fully Connected Layer herstellen zu können.

Im Folgenden wird ein kurzer Überblick über den Aufbau und die Wirkungsweise der für CNN charakteristischen Convolution und Pooling Layer gegeben.

Convolutional Layer: Im Convolutional Layer werden die Kernoperationen eines CNN durchgeführt, welche den größten Rechenaufwand beanspruchen. In ihnen wird die Faltung (engl.

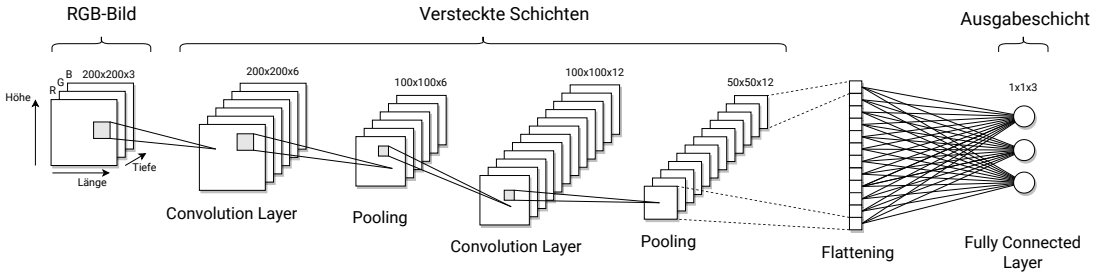
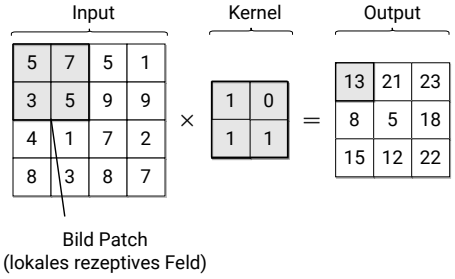


Abbildung 2.4: Exemplarischer Aufbau eines Convolutional Neural Networks.

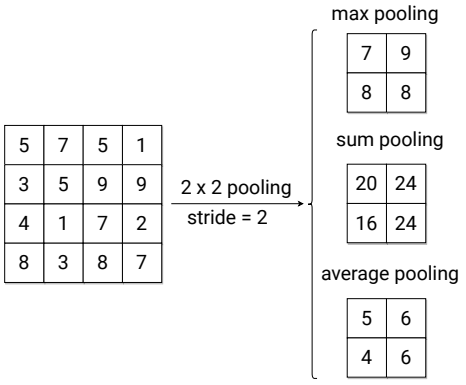
convolution) von Eingangsdaten auf Basis von Filtermatrizen (engl. *filter kernel*) unterschiedlicher Größe (z. B. 3×3 oder 5×5) durchgeführt. Die Filtermatrix wandert per *Sliding Window* Prinzip über die Pixel-Matrix des Inputs und berechnet die gewichtete Summe aus den Eingangsdaten und den Gewichten aus der Filtermatrix. Mittels *Padding* wird festgelegt, wie sich die Filtermatrix an Randregionen der Eingangsdaten verhält. Als Resultat wird eine neue Pixelmatrix in Form einer Merkmalskarte (engl. *feature map*) generiert, die in der Lage ist, regionale Zusammenhänge aus den Eingangsdaten zu erfassen. Dabei basiert jede Merkmalskarte auf den Gewichten eines einzelnen Filters. Dieses Prinzip der *shared weights* stärkt die translative Invarianz, sodass markante Merkmale unabhängig von ihrer Position im Bild gefunden werden können. Gleichzeitig sind die Neuronen eines Convolution Layers mit nur einem kleinen Bereich der vorherigen Schicht verbunden, sodass diese nur auf die Reize aus dem entsprechenden Bereich reagieren können. Dieses Prinzip entspricht dem biologischen Vorbild des rezeptiven Feldes und definiert den Netzhautbereich des Auges, mit dem ein einzelner Rezeptor verbunden ist.

Üblicherweise wird jede Eingangsschicht durch mehrere Filtermatrizen verarbeitet. So besteht die erste Convolution Ebene eines CNN typischerweise aus 16 oder 32 Merkmalskarten. Die Gewichte einer jeden Filtermatrix werden als Teil der *back propagation* mit trainiert, sodass jede Merkmalskarte individuelle Merkmale aus den Eingangsdaten extrahieren kann. Da die Filter auf räumlich benachbarte Informationen der Pixel-Matrizen angewendet werden, werden spatiale Informationen beibehalten. Auf diese Weise reduzieren Convolution Layer nicht nur die Anzahl der lernbaren Gewichte, sondern ermöglichen die Identifikation räumlicher Strukturen wie Kanten oder Objekte durch das Training der entsprechenden Filter. In klassischen Architekturen werden üblicherweise zwei Convolution Layer hintereinander geschaltet. Dies sorgt dafür, dass durch die doppelte Faltung auch spatiale Zusammenhänge aus weiter entfernten Pixelbereichen aufgebaut werden können.

Pooling Layer: Pooling Layer haben die Aufgabe, die Anzahl der Neuronen zu reduzieren und damit den Rechenaufwand zu senken, indem sie die Auflösung von Merkmalskarten verkleinern. Gleichzeitig wird die translative Invarianz der Merkmale gestärkt und das rezeptive Feld vergrößert. Dies erfolgt über sogenannte Pooling Operationen, mittels derer lokale Pixel-Gruppierungen durch eine Filtermaske in einem Wert zusammengefasst werden (*Downsampling*). Auch die Pooling-Operation ist eine Faltungs-Operation. Sie unterscheidet sich vom Convolution Layer jedoch dadurch, dass die Gewichte der Filtermatrizen nicht trainierbar,



(a) Faltung von Daten mittels eines Convolution Kernels mit *padding=0* und *stride=1*.



(b) Beispiel unterschiedlicher Pooling Operationen bei einer Filtermatrix von 2×2 und einem *stride* von 2.

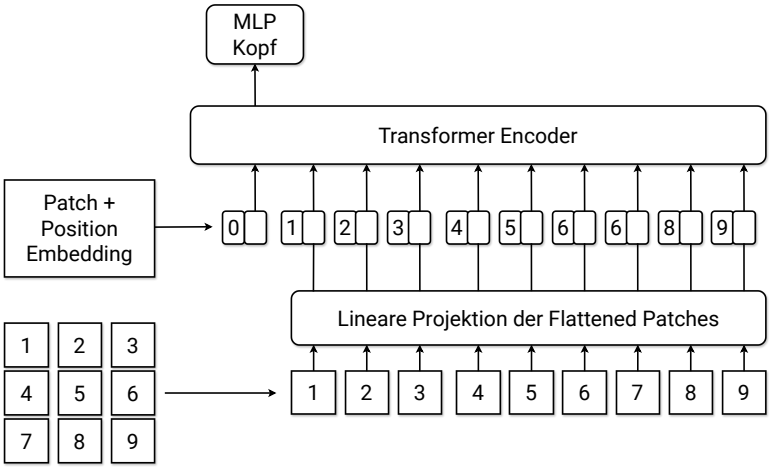


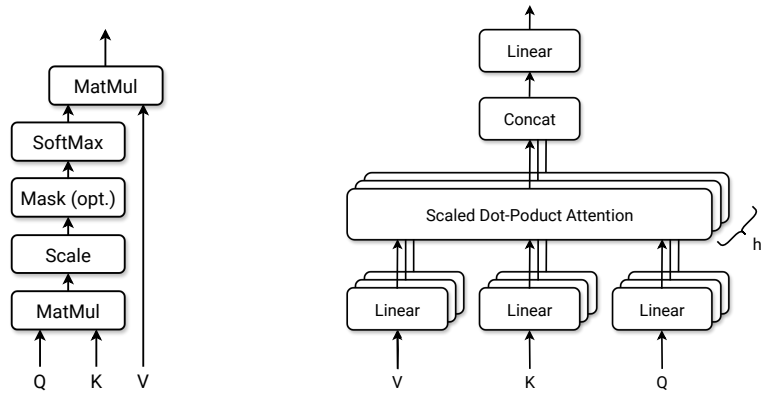
Abbildung 2.6: Aufbau eines Vision-Transformers.

sondern fest definiert sind. Außerdem wird eine größere Schrittweite (engl. *stride*) beim Ablaufen der Filtermaske über die Eingangsdaten verwendet. Der *stride* definiert dabei den Reduktionsfaktor für die resultierende Merkmalskarte und die Dimension der Filtermatrix bestimmt die Größe der Fläche, die in der Faltung berücksichtigt werden soll. Eine Filtermaske von 2×2 fasst vier Pixel zu einem zusammen. Bei einem *stride=2* erhält die resultierende Merkmalskarte nur ein Viertel der Auflösung. Typische Pooling Operationen sind Max Pooling, Average Pooling und Summen Pooling.

2.3 Vision-Transformer

Vision-Transformer (ViT) sind künstliche neuronale Netze, die auf der Transformer-Architektur beruhen und ursprünglich für die Entwicklung von Modellen zur Sprachverarbeitung verwendet [19] entwickelt wurden. Mittels einiger Änderungen der Architektur konnten Transformer jedoch auch für die Verarbeitung von Bildern zugänglich [20] gemacht und für Klassifikation [21, 22], Detektion [23] und Segmentierungsaufgaben [24] eingesetzt werden.

Transformer für Sprachmodelle verarbeiten Sätze, die als Sequenzen von Wörtern verarbeitet werden. In Vision-Transformern werden äquivalent dazu die Eingangsbilder in gleich große Bildausschnitte aufgeteilt und als Sequenz geordnet. Die einzelnen 2D-Bildausschnitte werden daraufhin in einen 1D-Vektor umstrukturiert (*flattening*), gleich dem Flattening in einem CNN. Der Transformer verwendet eine konstante, latente Vektorgröße in allen seinen Schichten, weshalb die 1D-Vektoren mittels einer trainierbaren linearen Projektion zusätzlich auf die passende Vektorlänge reduziert werden. Die resultierenden Vektoren der Bildauschnitte dienen zusammen mit jeweils einem Positions-Embedding als Input für den Transformer Encoder, dem Herzstück des Vision-Transformers. Der Transformer Encoder ist gleich dem eines Transformers für Sprachverarbeitung und basiert auf dem Attention-Prinzip [25, 26]. Der Aufbau einer typischen Vision-Transformer-Architektur ist in Abbildung 2.6 illustriert.



(a) Scaled dot-product attention

(b) Multi-Head Attention

Abbildung 2.7: Self-attention bei Vision-Transformern.

Für jede Inputsequenz und Positions-Embedding wird mittels Gewichtsmatrix ein Query-Vektor, ein Key-Vektor und ein Value-Vektor erzeugt, welche nun als Input dienen. Für den Query- und Key-Vektor wird als Nächstes der Attention-Score berechnet, skaliert, optional maskiert (sofern die Sequenzlänge aufgefüllt werden muss) und anschließend mittels Softmax-Funktion normalisiert. Anschließend erfolgt der Einbezug des Value-Vektors über das Skalar-Produkt. Dieser Prozess wird *scaled dot-product attention* genannt und ist in Abbildung 2.7a dargestellt. Er ist eingebettet in der *Multi-Head Attention* (Abbildung 2.7b), in der alle resultierenden Skalar-Produkte linear transformiert dem Multi-Layer Perzeptron übergeben werden. Für eine tiefergehende Erläuterung der Transformer-Architektur und des Attention-Prinzips sei auf [27, 28] verwiesen.

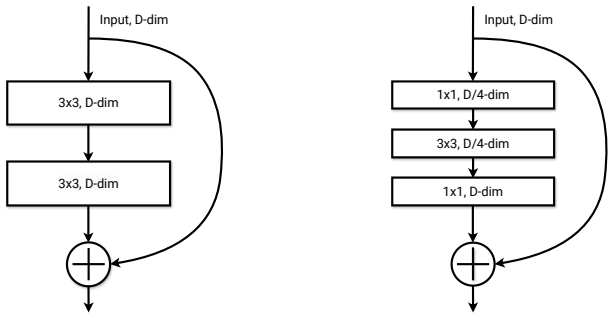
2.4 Etablierte Deep Learning Architekturen

In den vergangenen Jahren haben sich einige Deep-Learning-Architekturen zur lernbasierten Bildverarbeitung als besonders effektiv erwiesen. Im Folgenden werden zwei dieser Ansätze vorgestellt, die besonders populär sind und u. A. im weiteren Verlauf dieser Arbeit eingesetzt werden.

2.4.1 ResNet

He *et al*. [29] adressiert das Problem der verschwindenden Gradienten mit *residual blocks*, welche mit *identity skip connection* ausgestattet sind. Skip Connections, auch *residual connections* genannt, sind Direktverbindungen zwischen zwei Convolution Layers, die nicht durch Filter verarbeitet werden. Statt die Abbildung $x \rightarrow F(x)$ zu lernen, lernt ein Residual Block $x \rightarrow F(x) + x$, wobei $+x$ der Direktverbindung entspricht. Dieses Prinzip ermöglicht das Trainieren besonders tiefer Netze mit über 100 Layern.

Das ResNet stellt hierfür zwei Arten von residualen Blöcken vor: der *Basic Block* und der *Bottleneck Block*. Der Basic Block besteht aus zwei sequenziellen 3×3 Convolutional Layern mit einer Skip Connection, bei der Ein- und Ausgangsdimensionen identisch sind. Der Bottleneck Block besteht aus drei sequenziellen Convolutional Layern, wobei der erste und dritte ein 1×1 Layer ist und der zweite ein 3×3 Layer. Hierbei reduziert die erste 1×1 Faltungsschicht die Eingangsdimensionen auf ein Viertel, die zweite Schicht behält die reduzierte Anzahl an Dimensionen und die dritte Schicht erhöht die Anzahl an Merkmalskarten wieder auf den ursprünglichen Wert. Der Bottleneck Block ist Basis für die ResNet50, Resnet101 und die ResNet152-Architektur mit jeweils 50, 101 und 152 Schichten. ResNet18 und Resnet34 nutzen Basic Blocks.



(a) Basic Block **(b)** Bottleneck Block

Abbildung 2.8: Blocktypen der ResNet-Architektur.

2.4.2 MobileNet

Das MobileNet verfolgt das Ziel, die Anzahl der Parameter eines neuronalen Netzes zu reduzieren, bei gleichzeitigem Erhalt der Performanz. Die erste MobileNet Version wurde 2017 veröffentlicht [30], gefolgt von zwei weiteren verbesserten Versionen [31, 32]. Die zentrale Charakteristik zur Verschlankung des Netzes ist die *Depthwise Separable Convolution*-Technik, welche sich aus der *Depthwise Convolution* und der *Pointwise Convolution* zusammensetzt.

Bei der regulären Convolution (siehe Abbildung 2.9) entspricht die Tiefe M des Kernels der Anzahl an Merkmalsschichten. Bei einer Eingangsschicht mit 100 Featuremaps und einer Kernelgröße von 3×3, entsprechen die Dimensionen des Kernels somit $3 \times 3 \times 100$ mit $M = 100$.

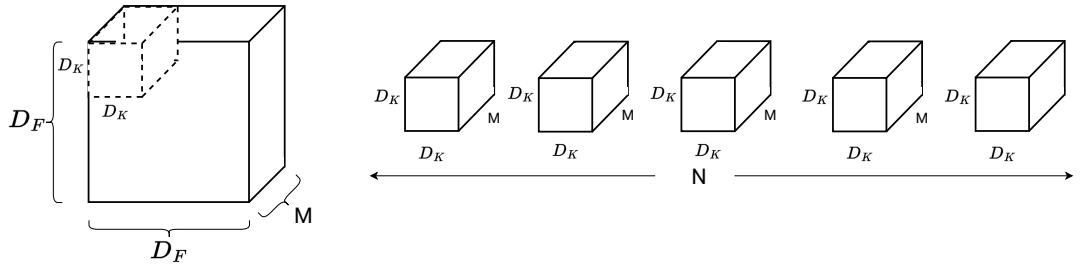


Abbildung 2.9: Prinzip der Standard Convolution.

Bei der Depthwise Convolution (siehe Abbildung 2.10) wird diese Verbindung aufgehoben und die Kernel wird mit einer festen Tiefe $M = 1$ versehen. Dadurch verarbeitet jeder Filter nur eine Merkmalsebene des Eingangs und benötigt N Filter, um jede Ebene einmal separat zu filtern.

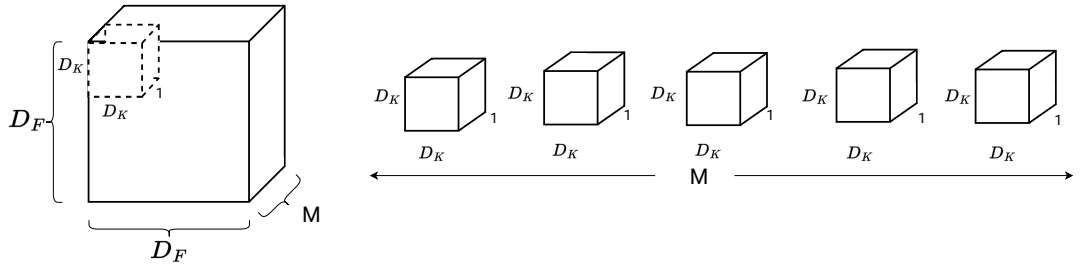


Abbildung 2.10: Prinzip der Depthwise Convolution.

Die Depthwise Convolution wird anschließend mit der Pointwise Convolution (siehe Abbildung 2.11) kombiniert. Bei ihr hat jeder Kernel die Größe 1×1, besitzt jedoch die Tiefe M analog zur Anzahl der Merkmalsebenen des Einganges.

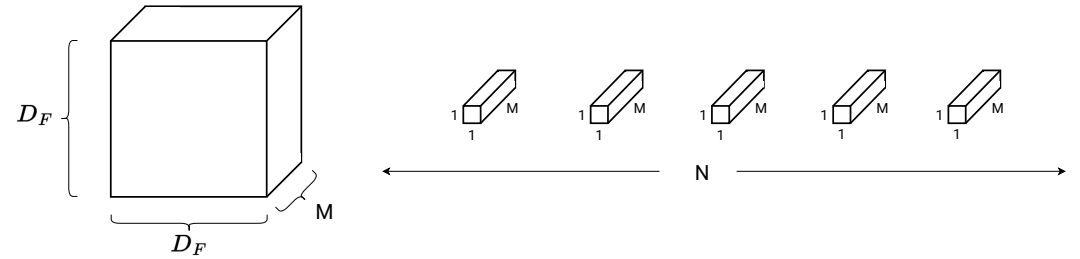


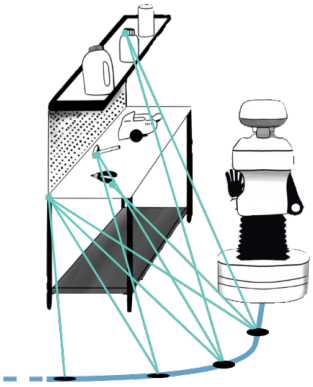
Abbildung 2.11: Prinzip der Pointwise Convolution.

Für die Anzahl an Parametern für N Kernel der regulären Convolution liegt bei $N \times M \times D_K^2$. Bei der Depthwise Separable Convolution beträgt die Anzahl der Parameter für die Depthwise Convolution $M \times D_K^2$ und für die Pointwise Convolution $N \times M$. Zusammen ergibt sich daraus eine Anzahl an Parametern von $M(D_K^2 + N)$ und somit weniger als für die reguläre Convolution benötigt werden.

Mit der dritten Version [32] wurde zudem eine Kombination mit dem *Squeeze-and-Excitation*-Prinzip [33] vorgeschlagen, um die Vernetzung zwischen den Channels zu verstärken und somit die Lernfähigkeit des Netzes zu verbessern.

Teil I

Umgebungsanalyse



KAPITEL 3

Simultane Lokalisierung und Kartierung

Eine präzise Positionsbestimmung und eine konsistente Umgebungskarte sind die Grundlagen für die zuverlässige Orientierung und sichere Navigation mobiler Roboter. Im Außenbereich können Roboter ihre eigene Position und eine detaillierte Karte ihrer Umgebung durch externe Informationsgeber, wie z. B. GPS, abrufen [34]. Dieses Prinzip ist jedoch oftmals nicht erwünscht, da hierbei die Funktionsfähigkeit des Roboters von einer intakten Verbindung zur externen Informationsquelle abhängig ist und somit die Sicherheit und Zuverlässigkeit beschränkt ist. In unbekannten oder abgeschirmten Umgebungen, z. B. in Gebäuden, Unterwasser oder in Höhlen, steht die Option der externen Informationsbeschaffung nicht zur Verfügung. In diesem Fall muss der Roboter selbstständig in der Lage sein, sich zu orientieren und sich einen Eindruck seiner Umgebung verschaffen zu können.

SLAM-Methoden (Simultane Lokalisierung und Kartierung) befassen sich mit Lösungsansätzen, um die Funktionalität der Positionsbestimmung und Kartenerstellung der Umgebung vollständig auf den Roboter zu übertragen. Hierfür muss der Roboter eine konsistente Karte erstellen und gleichzeitig seine Position innerhalb der erzeugten Karte lokalisieren. Statt auf externe Informationen zuzugreifen, werden die dafür benötigten Daten mittels Sensorik, wie Kameras, Laser, Radar oder Ultraschall, eigenständig aus der Umgebung extrahiert. Anwendungen von SLAM finden sich unter anderem im autonomen Fahren [35, 36, 37, 38], in der Augmented Reality [39, 40, 41] sowie in unterschiedlichen Arten von Robotern, ausgehend von simplen Staubsaugrobotern [42] über autonome Tauch- [43, 44] und Flugrobotern [45] hin zu humanoiden Robotern [46, 47, 48, 49].

Im folgenden Verlauf des Kapitels wird ein grundlegendes Lösungsverfahren zum bildbasierten, sogenannten Visual-SLAM, erläutert. Hierzu wird zunächst der Ablauf einer typischen SLAM-Pipeline aufgezeigt (Abschnitt 3.1). Im Anschluss werden die Herausforderungen und Limitationen derzeitiger Lösungsverfahren genauer analysiert (Abschnitt 3.2), welche als Motivation der Folgekapitel

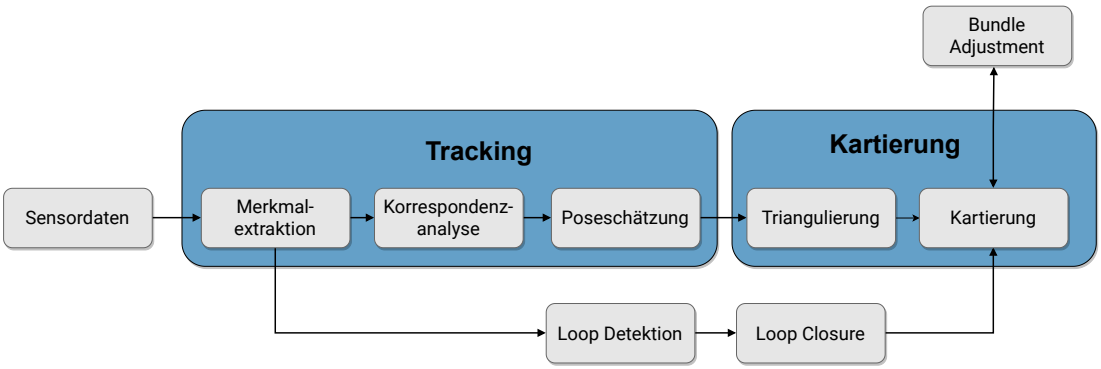


Abbildung 3.1: Schematische Verarbeitungspipeline eines merkmalbasierten SLAM-Algorithmus.

dieses Teils der Arbeit dienen. Abschließend werden zwei populäre Datensätze vorgestellt, welche typischerweise zur Evaluierung von SLAM-Methoden verwendet werden (Abschnitt 3.3).

3.1 SLAM-Pipeline

SLAM [50] beschreibt ein "Henne und Ei"-Dilemma: Wäre die Position des Roboters bekannt, könnte anhand dessen mit den Sensordaten des Roboters eine Karte erzeugt werden. Wäre die Umgebung bekannt, könnte der Roboter die Sensordaten der Umgebung mit der Karte abgleichen, um seine Position darin zu bestimmen. Da jedoch beides, Umgebung und Position des Roboters, unbekannt sind, wird SLAM zu einer Herausforderung.

Diese Arbeit befasst sich mit bildverarbeitenden Verfahren. Deshalb wird in den folgenden Abschnitten der Fokus auf Visual-SLAM gelegt, bei dem Kameras als Hauptsensor verwendet werden. Visual-SLAM wird in der Regel in indirekte (merkmalbasierte) und direkte (bildbasierte) Methoden unterteilt. Merkmalbasierter SLAM verwendet markante Bildpunkte zur Verfolgung der Positionsänderungen, während direkte Methoden das gesamte Bild betrachten und vergleichen, um die Pose und Karte durch Minimierung des photometrischen Fehlers zu rekonstruieren. Aufgrund seiner Effizienz und Robustheit hat sich merkmalbasierter SLAM für den Einsatz auf mobilen Systemen mit Echtzeit-Applikation als geeigneter erwiesen [51, 52]. Abbildung 3.1 zeigt die typische Pipeline eines merkmalbasierten SLAM-Algorithmus, welcher sich allgemein in einen Tracking- und Kartierungsprozess (auch als Frontend und Backend bezeichnet) untergliedern lässt. Im Folgenden wird ein kurzer Überblick über die Verfahrensweisen der einzelnen Komponenten gegeben.

3.1.1 Tracking

Der Trackingprozess inkludiert üblicherweise einen Initialisierungsschritt zur Erzeugung eines ersten Sets von 3D-Merkmalen bzw. einer lokalen Karte, welche für die darauffolgenden Schritte als Basis dient und inkrementell erweitert wird. Die Erstellung der Karte folgt den Ansätzen aus dem Bereich des *Structure from Motion* [53]. Hierbei werden zwei oder mehr Bilder der gleichen Szene aus unterschiedlichen Perspektiven nach markanten Merkmalen (engl. *features*) analysiert und untereinander verglichen. Ziel ist es, Merkmal-Korrespondenzen zwischen den Bildern zu finden,

deren Projektion auf der Bildebene durch dieselben 3D-Punkten der Szene hervorgerufen wird. Dieses Prinzip ist in Abbildung 3.2 veranschaulicht. Mittels dieser Korrespondenzen können im Anschluss Rückschlüsse auf den Positionswechsel der Kamera zwischen den Aufnahmen gezogen werden (Poseschätzung).

Merkmalsextraktion und -korrespondenzanalyse Bei der Merkmalsextraktion werden im SLAM aufgrund ihrer Effizienz typischerweise immer noch *hand-crafted* Merkmale verwendet. Typische Merkmale sind BRIEF [54], BRISK [55], FREAK [56], ORB [57], SIFT [58] und SURF [59]. Neben diesen klassischen Merkmalsvarianten gibt es auch lernbasierte Merkmale [60, 61, 62, 63, 64], die auch als Deep-Features bezeichnet werden.

Neben der Erfassung geeigneter Merkmale ist insbesondere ihre adäquate Encodierung wichtig, um Verzerrungen in der Erscheinung, induziert durch den Perspektivwechsel zwischen den Beobachtungen, nicht in die Korrespondenzanalyse mit einzubeziehen. Dazu zählt insbesondere eine rotations- und skalierungsinvariante Merkmalsbeschreibung.

Auf Basis der Merkmalskodierungen werden im Anschluss Distanzberechnungen durchgeführt. Merkmalspaare mit geringer Distanz weisen starke Ähnlichkeiten auf, die auf Korrespondenzverhalten hindeuten und so einander zugewiesen werden können.

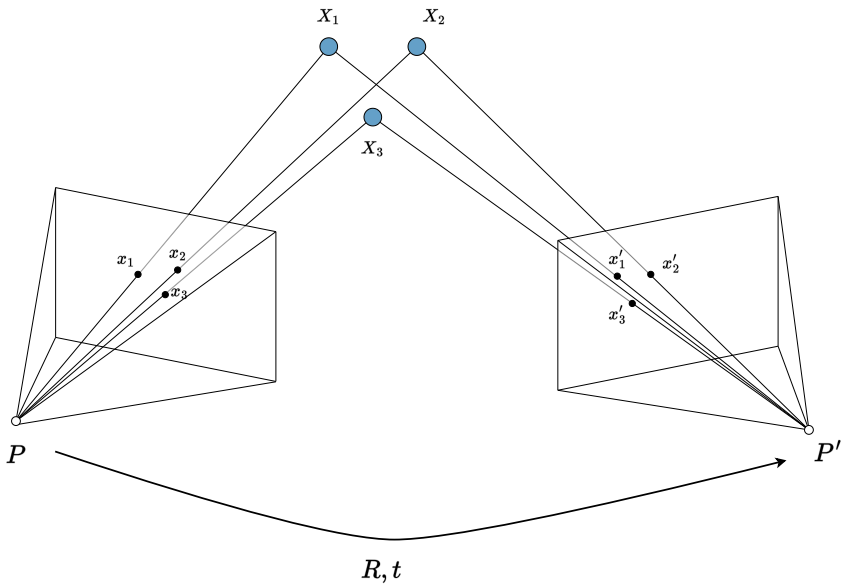


Abbildung 3.2: Structure from Motion: Photogrammetrisches Prinzip.

Poseschätzung Sind genügend Merkmal-Korrespondenzen zwischen zwei Bildern gefunden, kann mittels Epipolargeometrie, im Speziellen der 8-Punkt-Algorithmus [65], die Transformation zwischen beiden Kameraperspektiven ermittelt werden. Die Transformation ist bei unkalibrierten Kameras als Fundamentalmatrix F definiert, welche die beiden Merkmal-Korrespondenzen x und x' in $x^T F x' = 0$ in Beziehung setzt. Fx' beschreibt die Epipolarlinie, auf der x in der korrespondierenden Bildebene liegt. Die Fundamentalmatrix lässt sich mit den intrinsischen Kameraparametern K und K' zur Essential-Matrix erweitern, definiert als $E = K^T F K'$.

Diese encodiert die Relation beider Kamerapositionen mit $E = R\,[t]_\times$. R entspricht der 3×3 Rotationsmatrix zwischen beiden Kameras und t ist ein dreidimensionaler Translationsvektor (siehe Abbildung 3.2). Zur Initialisierung wird die Position der ersten Kamera mit R als Einheitsmatrix und t als Nullvektor als Ursprung definiert, sodass die Pose der zweiten Kamera in Relation dazu der aus der Essential-Matrix ermittelten Rotation R und der Translation t entspricht.

3.1.2 Kartierung

Der Prozess der Kartierung wird auch als Backend bezeichnet. Während die Primäraufgabe des Trackings (Frontend) die Sensordatenverarbeitung ist, werden im Backend die daraus resultierenden Daten verwendet, um eine Karte aufzubauen und Optimierungsprozesse durchzuführen.

3D-Triangulierung der Objektpunkte Aus der Pose (den extrinsischen Parametern) und den intrinsischen Parametern K lässt sich nun die Projektionsmatrix als $P = K[I|0]$ für die erste Kamera, bzw. $P' = K'[R|t]$ für die zweite Kamera bestimmen. Mit deren Hilfe werden im Anschluss die 3D-Position X des beobachteten Merkmals in Weltkoordinaten hergeleitet. Die Beziehung ist definiert als $x = PX$, sodass sich

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K[R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.1}$$

ergibt, mit s als Skalierungsfaktor. Für denselben 3D-Punkt X gilt auch für die zweite Kamera $x' = P'X$ mit dem 2D-Merkmal x' und der Projektionsmatrix der zweiten Kamera P'. In einem idealen Szenario mit perfekt kalibrierten Kameras ist X in beiden Fällen identisch. In der Realität kann es jedoch zu Abweichungen kommen, sodass sich die Strahlen nicht in X schneiden, sondern durch Rauschen gestört werden. In diesem Fall wird versucht, durch Optimierungsmethoden (z. B. Maximum-Likelihood-Verfahren) sich dem tatsächlichen 3D-Punkt möglichst nah anzunähern.

Kartenaufbau Das zuvor beschriebene Prinzip aus Merkmalsextraktion, Korrespondenzanalyse und Poseschätzung kann in der Theorie beliebig oft mit jedem neuen Bildpaar durchgeführt werden. In der Praxis hat sich dies jedoch als unpraktikabel und ineffizient erwiesen. Stattdessen wird versucht, zuvor lokalisierte und triangulierte Merkmalspunkte zu tracken, um auf dessen Basis wiederum die neue Position zu bestimmen und gleichzeitig neue Merkmale im 3D-Raum zu registrieren. Hierfür werden typischerweise Methoden wie Perspective-n-Point eingesetzt [66].

3.1.3 Bundle Adjustment

Bundle Adjustment ist ein Ansatz, um die Positionen von Punkten einer 3D-Szene, sowie die unterschiedlichen Positionen der Kamera, von denen aus die 3D-Szene eingefangen wurde, in einem gemeinsamen Prozess zu optimieren. Im Bereich SLAM wird Bundle Adjustment in unterschiedlichen Variationen eingesetzt, um Drifts in der Trajektorie und Kartierungsfehler zu reduzieren [67, 68, 69].

Gegeben seien N Kameras bzw. Kameraperspektiven, definiert als $\Pi = \{\pi_i\}$ mit intrinsischen und extrinsischen Parametern. $X_w = \{\chi_p^w\}$ sei eine Menge von M 3D-Punkten in Weltkoordinaten. $X_s = \{\chi_{ip}^s\}$ entspricht deren Projektion auf der Bildebene als 2D-Punkte in allen i Kameras. Das Ziel von Bundle Adjustment ist nun die Minimalisierung des Reprojektionsfehlers über alle Beobachtungen.

$$\Pi^*, W_w^* = \arg\min_{\Pi, X_w} \sum_{i=1}^{N} \sum_{i=1}^{M} w_{fp} ||x_{ip}^s - \chi_i(x_p^w)||_2^2 \tag{3.2}$$

Die Minimierung erfolgt üblicherweise über Methoden der kleinsten Quadrate, von denen der Levenberg-Marquardt Algorithmus besonders populär ist [70]. Da beliebig viele 3D-Punkte und Kameraperspektiven in die Gleichung einbezogen werden können, eignet sich Bundle Adjustment als flankierender Ansatz [71, 72, 73], um eine globale Optimierung der gesamten Karte und der Kameratrajektorie durchzuführen, sowie inkonsistente Punkte und Kamerapositionen zu entfernen. Gleichzeitig hat sich der Einsatz auch für kleinere, lokale Teilbereiche der Karte bewährt [74, 75, 76].

3.1.4 Loop Closure

Loop Closure befasst sich mit den Herausforderungen zur Erkennung und Verarbeitung von Situationen, in denen das mobile System zu einem bereits zuvor besuchten Ort zurückkehrt. Dies ermöglicht, den Drift durch die Fehlerfortpflanzung in der Schätzung der Position und der Karte zu korrigieren und macht Loop Closure zu einem essenziellen Bestandteil robuster und langer Fahrten [77, 78]. Zur Detektion von Loops werden üblicherweise Bag-of-Words (BoW)-Ansätze [79, 80] verwendet, mittels derer die extrahierten Features aus der gesamten Karte effizient in einer Datenbank gespeichert werden. Dadurch können neue Merkmale mit der gesamten Karte abgeglichen werden, um die Beobachtung bereits zuvor kartierten Orten wiederzuerkennen. Üblicherweise folgt auf einen detektierten Loop Closure auch eine globale Optimierung mittels Bundle Adjustment, um weiteres Optimierungspotenzial zu nutzen.

3.2 Herausforderungen beim Visual-SLAM

SLAM-Verfahren werden bereits seit Jahrzehnten erforscht und konnten im Laufe dieser Zeit ihre Performanz deutlich verbessern. Trotz dieser Fortschritte ist Visual-SLAM jedoch immer noch mit einer Reihe von Herausforderungen konfrontiert, die dessen Genauigkeit, Robustheit und Anwendbarkeit negativ beeinflussen können.

Wesentliche Herausforderungen sind die Langzeitstabilität und der damit verbundene Drift, eine hohe Umgebungsvariabilität durch wechselhafte Beleuchtung und die Skalierbarkeit in großen, komplexen Umgebungen. Eine der bedeutendsten Herausforderungen ist jedoch der Umgang mit dynamischen Umgebungen. Bewegliche Umgebungselemente wie Menschen, Fahrzeuge oder Tiere können die Genauigkeit der Lokalisierung und Kartierung erheblich beeinträchtigen. Das System muss in der Lage sein, zwischen temporären und permanenten Veränderungen in der Umgebung zu unterscheiden, um eine stabile und zuverlässige Karte zu erstellen.

Bei Standard SLAM-Verfahren wird der Einsatz in einer statischen Umgebung vorausgesetzt. Bei der Detektion und Verarbeitung von Merkmalen wird deshalb nicht geprüft, ob die gewählten Umgebungspunkte sich dynamisch verhalten oder nicht. Für den Einsatz in komplexen, dynamischen Orten müssen deshalb zusätzliche Methoden entwickelt werden, die entweder eine Vorsegmentierung der Umgebung auf dynamische Elemente durchführen, oder im SLAM-Prozess selbst ein potenzielles dynamisches Verhalten der Umwelt einbeziehen.

Ein weiterer limitierender Faktor ist der Informationsgehalt der erzeugten Umgebungskarten. Typischerweise liegt dieser für Kartenpunkte lediglich in einer Positionsbeschreibung mit Farbwert. Die Nutzung dieser Karten für den Aufbau eines Verständnisses über die Umgebung und die Erzeugung kontextueller Interaktionsfähigkeit erfordert weitergehende Beschreibungen. Hierzu zählen insbesondere semantische Informationen, die die Umgebungspunkte zu Objekten und Bereichen zusammenfassen und ihnen eine Bedeutung zuweisen.

3.3 Datensätze

Um die Eigenschaften und Performanz von SLAM-Methoden zu evaluieren und untereinander vergleichbar zu machen, wurde in den letzten Jahren eine Reihe unterschiedlicher Datensätze veröffentlicht. Im Folgenden werden die Datensätze TUM [81] und KITTI [82] vorgestellt, welche zu den populärsten SLAM-Benchmarks gehören und auch in den nachfolgenden Kapiteln dieser Arbeit Anwendung finden.

RGB-D TUM Datensatz Der RGB-D TUM Datensatz besteht aus einer Reihe von RGB-D-Bildsequenzen, welche mittels einer Microsoft Kinect Kamera in Büro- und ähnlichen Innenraumumgebungen aufgenommen wurden. Die Sequenzen haben eine Bildrate von 30 und eine Bildauflösung von 640×480. Neben den Farb- und Tiefenbildern wird die zugehörige Kamera-Odometrie als Grundwahrheit zur Verfügung gestellt, welche mittels eines Motion-Capture-Systems mit acht Hochgeschwindigkeit-Tracking-Kameras ($100 Hz$) erfasst wurde.

Der Datensatz enthält insgesamt 47 Sequenzen mit Grundwahrheiten und zusätzliche 42 Sequenzen ohne Grundwahrheiten. Die Sequenzen haben eine Dauer von 15 Sekunden bis hin zu 2 Minuten und sind in unterschiedliche Kategorien unterteilt. In einer Aufnahmeklasse wurde die Kamera bei den Aufnahmen per Hand gehalten, bei einem weiteren Teil von Szenarien ist das Kamerasystem auf einem Pioneer-Robot installiert. Bei den handgeführten Aufnahmen wurden in einigen Sequenzen dynamische Objekte eingebunden, bei denen Personen in den Bildbereich treten, dort interagieren

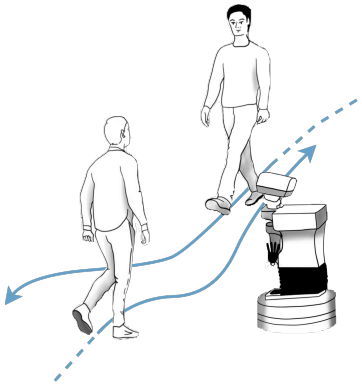
oder sich anderweitig bewegen. Eine weitere besondere Charakteristik in dieser Kategorie sind die unterschiedlichen handgeführten Kamerabewegungen und -drehungen, welche die Szenarien für SLAM-Methoden besonders herausfordernd machen. Hierzu gehört das Abfahren entlang der XYZ-Achsen, die Rotation über die Roll-, Nick- und Gierwinkel und das Abfahren einer Halbkugel. Abbildung 3.3 zeigt einige exemplarische Aufnahmen aus dem Datensatz.

Abbildung 3.3: Beispielbilder aus dem RGBD-TUM Datensatz.

KITTI Datensatz Der KITTI Datensatz [82] ist eine Benchmark-Suite mit Szenarien aus dem Außenbereich und dient durch seine umfangreiche Annotation als Basis für Training und Validierung mehrerer unterschiedlicher Computer Vision Aufgaben. Für die Erzeugung des Datensatzes wurde das Dach eines Autos mit einem Stereo-Kamerasystem, einem Velodyne Laserscanner und einem GPS-Modul bestückt. Im Anschluss wurden unterschiedliche Strecken innerstädtisch, auf Landstraßen und auf der Autobahn abgefahren, um insgesamt ca. 6 Stunden an Datenmaterial zu sammeln (siehe Abbildung 3.4). Die Grundwahrheit der Trajektorie wurde durch den Laserscanner und das GPS-Modul automatisch bestimmt. In nachfolgenden Arbeiten wurden weitere Informationen wie vorbeifahrende Autos und Passanten auf dem Fußweg annotiert, sowie weitere semantische Szenensegmentierungen für die LIDAR-Daten ergänzt [83]. Der KITTI-Datensatz zählt zu den Primärdatenbanken, um Langzeit-SLAM-Methoden zu evaluieren und das Verhalten im Außenbereich und an den damit einhergehenden Lichtverhältnissen zu testen.

Abbildung 3.4: Beispielbilder aus dem KITTI Datensatz.

KAPITEL 4



Odometriestabilisierung in dynamischer Umgebung

Dieses Kapitel behandelt die Herausforderungen, die mit der Grundannahme einer statischen Umgebung einhergehen. Diese Annahme ermöglicht es, markante Landmarken aus der Umgebung als Referenzpunkte zu extrahieren, um sie zur Bestimmung der Eigenbewegung und zum Aufbau einer Umgebungskarte einzusetzen. Weisen die verwendeten Referenzpunkte jedoch kein statisches Verhalten auf, kann unter der Annahme einer statischen Umgebung die Genauigkeit der Odometriebestimmung reduziert und Kartenelemente fehlerhaft platziert werden, bis hin zum vollständigen Verlust der Orientierung (Tracking-Verlust).

In typischen Anwendungsbereichen und damit gleich dem Zielszenario dieser Arbeit erfolgt der Einsatz eines mobilen Cobots in dynamischen Arbeitsumfeldern mit vielseitigen Interaktionen mit Personen und Objekten. Die Bedingung einer statischen Umgebung ist damit nicht gegeben. Deshalb müssen SLAM-Methoden optimiert, erweitert oder neu gedacht werden, um sie für diese neuen Herausforderungen dynamischer Umgebungen einsatzfähig zu machen.

Im Folgenden wird eine neue Methode vorgestellt, die SLAM-Algorithmen in die Lage versetzt, robuster gegenüber dynamischen Objekten und Veränderungen in der Umgebung zu werden, um Inkonsistenzen in der Datenverarbeitung zu reduzieren und somit die Lokalisierungspräzision der SLAM-Algorithmen zu verbessern. Zuerst werden in Abschnitt 4.1 bestehende Ansätze diskutiert, welche die Fehlerreduzierung von Visual-SLAM in dynamischer Umgebung adressieren. Im Anschluss wird in Abschnitt 4.2 eine eigene Methodik vorgestellt, welche dynamische Bildelemente in einem vorgelagerten Ansatz erfasst, sodass diese vor dem Trackingprozess eliminiert werden können. In Abschnitt 4.3 wird der vorgestellte Ansatz auf öffentlichen Datensätzen quantitativ und qualitativ

evaluiert und mit dem Stand der Technik verglichen. Das Kapitel schließt mit einer Zusammenfassung und einer Diskussion in Abschnitt 4.4.

Forschungsbeitrag

» Es wird ein neuer Ansatz zur Segmentierung dynamischer Bildelemente vorgestellt, um den Lokalisierungsprozess von SLAM-Methoden in nicht-statischen Umgebungen zu verbessern.

» Der Ansatz benötigt kein *a priori*-Wissen über die Szene und erhält durch bildspezifische und pixelweise Segmentierung eine maximal statische Referenzfläche.

» Die vorgeschlagene Methode erzielt in mehreren Experimenten eine Fehlerreduktion von bis zu 98% und kann damit maßgeblich zu einer verbesserten Orientierung mobiler Roboter in realen Umgebungen beitragen.

4.1 Verwandte Arbeiten

Die Behandlung dynamischer Umgebungen in SLAM-Verfahren ist Gegenstand einer Vielzahl unterschiedlicher Lösungsansätze, um genaue Lokalisierung und Karten zu gewährleisten. Typische Strategien für merkmalbasierte SLAM-Verfahren zielen auf einen zusätzlichen Vorverarbeitungsschritt ab, der dynamische Bildelemente detektiert und segmentiert. So können diese zum einen nicht die Odometrie negativ beeinflussen, zum anderen werden sie so für die Kartierung der Umgebung berücksichtigt. Gliedert man einen solchen Ansatz in die SLAM-Pipeline als Vorverarbeitungsschritt ein, verhindert man außerdem, dass diese herausfordernden Bildbereiche in nachfolgenden Prozessen weitere Ressourcen binden.

Alcantarilla *et al*. [84] erzeugen einen dichten Szenenfluss [85] mittels Stereo-Kamera, in dem dynamische Zielobjekte als Ausreißer durch die Mahalanobis-Metrik [86] lokalisiert und segmentiert werden. Jaimez *et al*. [87] berechnen den Szenenfluss über K-Means Clustering und segmentieren dynamische Elemente über eine grobe Schätzung der Kamerabewegung. Kim *et al*. [88] präsentieren einen *background model-based dense-visual-odometry*-Algorithmus (BaMVO), der dynamische Objekte durch Modellierung eines statischen Hintergrunds auf Basis mehrerer aufeinanderfolgender Tiefenbilder herausfiltert. Andere Methoden operieren auf Merkmalsebene: Li *et al*. [89] nutzen Merkmale mit Tiefeninformationen, um mittels eines *Intensity Assisted Iterative Closest Point*-Ansatzes (IAICP) dynamische Ausreißer zu determinieren. Sun *et al*. [90] entfernen dynamische Bildregionen aus RGB-D-Daten durch die Verfolgung von Partikeln auf Bildern mit Eigenbewegungskompensation. Wang *et al*. [91] segmentiert dynamische Bildbereiche durch die Einbeziehung epipolarer Randbedingungen und geclusterter Tiefenkarten.

Weitere Ansätze aus jüngerer Zeit machen sich Deep Learning zunutze, um die dynamische Objektsegmentierung zu verbessern. Ein wesentliches Verfahren wurde von Bescos *et al*. [92] präsentiert,

welches eine CNN-basierte Objektsegmentierung [93] einsetzt und Objektbereiche von vordefinierten Objektklassen aus der SLAM-Pipeline entfernt. In ähnlicher Weise nutzen Liu *et al*. [94] statt eines Objektsegmentierungsmodells den Objektdetektor YOLO [95], um Objektinstanzen von vordefinierten, potenziell dynamischen Objektklassen, zu erkennen. In einem zweiten Schritt werden die Zielobjekte mithilfe des optischen Flusses zusätzlich auf ihr dynamisches Verhalten analysiert und gegebenenfalls segmentiert. Yu *et al*. [96] konstruieren einen statischen Hintergrund mittels SegNet [97]. Zhang *et al*. [98] bestimmen dynamische Pixel, indem sie den von PWC-Net [99] erzeugten optischen Fluss mit einer Eigenbewegungsschätzung validieren.

Die beschriebenen Methoden sind an Abhängigkeiten geknüpft, die die Robustheit und die Genauigkeit der eingesetzten Verfahren wesentlich beeinflussen. Der Ansatz, dynamische Bildbereiche *a priori* durch Objekterkennung auszuschließen [94, 96, 92], setzt zum einen voraus, dass die Objekte dem neuronalen Netz überhaupt bekannt sind. Zum Anderen wird durch die Segmentierung eines potenziellen dynamischen Objekts die Möglichkeit ausgeschlossen, dass sich potenziell dynamische Objekte statisch verhalten und daher als Teil des statischen Hintergrunds verwendet werden können. Nimmt man ein Szenario mit vielen, potenziell dynamischen Personen, z. B. eine überfüllte Tribüne in einer Oper, wird hier das rigorose Segmentieren zum Scheitern des Trackings führen, da der gesamte Bildbereich mit Zuschauern segmentiert wird, obwohl deren tatsächliche Gesamtbewegung minimal ist. Einige Verfahren [94, 96] gehen dieses Problem an, indem sie zusätzlich den optischen Fluss berechnen und mittels eines zusätzlichen Verarbeitungsschrittes für Objekte der Zielklassen die bildspezifische Bewegung bestimmen. Dies kann die Robustheit in Szenen mit vielen potenziellen, aber nicht tatsächlich dynamischen Objekten erhöhen, berücksichtigt aber nicht, dass auch üblicherweise statische Objekte (z. B. Bücher, Stühle, Tische) bewegt werden und daher zu dynamischen Objekten werden können. Eine semantisch basierte Segmentierung ist deshalb sehr stark an Szene-spezifische Anwendungen gebunden.

Weiterhin ist die Verwendung von ausschließlich planaren Informationen entweder für die Schätzung des optischen Flusses [96], die Schätzung der Eigenbewegung [90] oder das Feature Matching fehleranfällig, um Bewegungen kleinerer Bildfragmente, die orthogonal zur Bildebene verlaufen, zu bestimmen. Viele SLAM-Algorithmen verwenden Tiefendaten von IR-Sensoren oder Stereo-Kameras, um die Schätzung der Eigenbewegung im dreidimensionalen Raum zu verbessern und eine metrische Distanzskalierung für die Karte zu ermöglichen. Es ist daher sinnvoll, diesen zusätzlichen Informationskanal auch für die dynamische Szenenanalyse zu nutzen. Allerdings neigt die Tiefenschätzung insbesondere unter Verwendung von ToF-Techniken dazu, rauschanfällig und mit zunehmender Entfernung ungenau zu werden. Dies macht die tiefenbasierte Segmentierung von dynamischen Objekten zu einer großen Herausforderung, insbesondere wenn amerainduzierte Bewegung und Objektbewegung in Kombination auftreten.

4.2 Ansatz zur bildbasierten Odometriebestimmung in dynamischer Umgebung

Im Folgenden wird ein eigener Ansatz zur pixelweisen Segmentierung dynamischer Bildelemente durch Berechnung des Reprojektionsfehlers gegenüber einer geschätzten 3D-Homografie vorge-

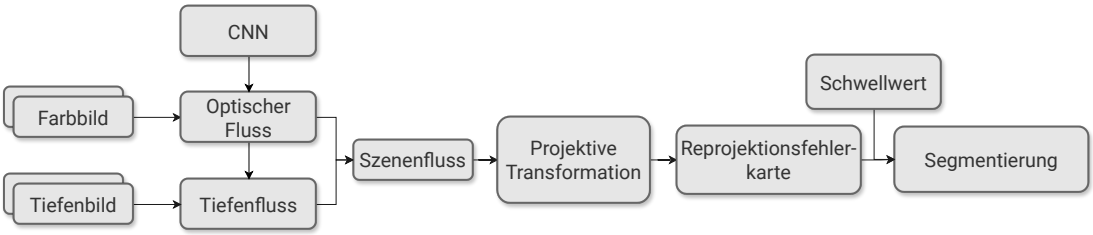


Abbildung 4.1: Schematischer Ablauf der vorgeschlagenen Methode zur Segmentierung dynamischer Bildelemente.

schlagen. Die 3D-Transformationsmatrix wird auf Grundlage eines Szenenflusses geschätzt, welcher wiederum aus einem prädizierten optischen Fluss und dazugehörigen Tiefen-Informationen interpoliert wird. Die pixelweise Segmentierung stellt sicher, dass ein möglichst großer Bereich des statischen Hintergrunds erhalten bleibt, sodass Robustheit und Genauigkeit der Odometrie gewahrt werden. Die iterative Berechnung der projektiven Transformation deckt jede Art von auffälliger dynamischer Pixelbewegung gegenüber der Eigenbewegung der Kamera auf und ist somit frei von semantischen Randbedingungen.

Abbildung 4.1 veranschaulicht das vorgeschlagene Verfahren, welches im Folgenden im Detail beschrieben wird.

4.2.1 CNN-basierte Interpolation des 3D-Szenenflusses

Der Begriff Szenenfluss (engl. *scene flow*) hat seinen Ursprung in der Arbeit von Vedula *et al*. [85] und beschreibt die Erweiterung des optischen Flusses in den 3D-Raum als dreidimensionales Bewegungsfeld von Punkten. Seither wird der Szenenfluss aus Stereo-[100, 84] oder RGB-D-Daten [101, 102, 103, 104, 87] erzeugt. Analog zu den Methoden des optischen Flusses basieren derzeitige Methoden zur Erzeugung von dichtem/pixelweisem Szenenfluss [105, 106, 107] auf Deep Learning-Verfahren. Trotz ihrer akkuraten Prädiktionsfähigkeiten benötigen diese Verfahren Laufzeiten von wenigen Sekunden bis zu vielen Minuten für ein einzelnes Bildpaar und sind deshalb für die Echtzeit-Anwendung (noch) nicht geeignet [108, 109, 110]. In dieser Arbeit wird deshalb ähnlich dem Ansatz von Schuster *et al*. [111] ein leistungsfähiges Modell zur Prädiktion des optischen Flusses (2D) mit Tiefeninformationen kombiniert, um so den Szenenfluss anstelle eines End-to-End-Ansatzes in einem Zwei-Schritt-Verfahren abzuschätzen.

Wie in Abbildung 4.1 dargestellt, bestehen die Eingangsdaten aus zwei konsekutiven Farb- und Tiefenbildpaaren. Die Verarbeitung erfolgt über die Prädiktion des optischen Flusses, über die Interpolation des Tiefenflusses, hin zur Erzeugung eines Szenenflusses.

Prädiktion des optischen Flusses Im ersten Schritt wird auf Basis eines konsekutiven RGB-Bildpaars der optische Fluss prädiziert. Hierfür wird das CNN-basierte FlowNet2 [112, 113] Modell verwendet. Es zielt darauf ab, mittels kaskadierten CNNs die Verschiebung von Pixeln zwischen zwei Bildern zu erfassen. Mit wachsender Tiefe des Netzes wird die Auflösung des optischen Flusses

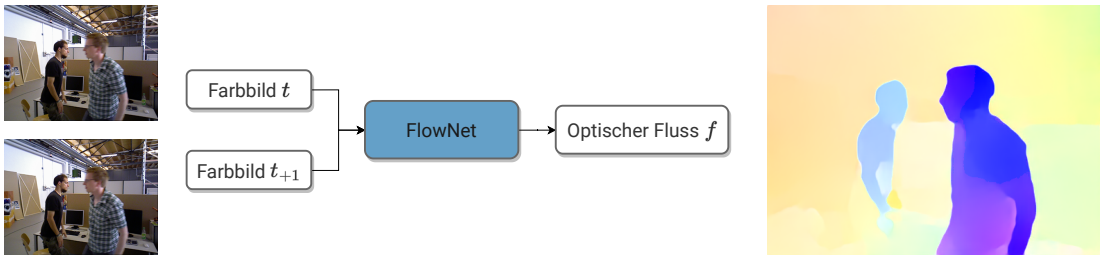


Abbildung 4.2: Prädiktion des optischen Flusses zwischen zwei RGB-Bildern durch FlowNet2 [112, 113].

sukzessiv gesteigert, sodass im finalen Abschnitt eine Schätzung der Verschiebung auf Subpixel-Ebene erzielt werden kann. Abbildung 4.2 zeigt ein exemplarisches Ergebnis einer solchen Prädiktion. Farbe und Farbintensität zeigen die Richtung und die Weite der Verschiebung auf.

Der optische Fluss bildet primär die planare Bewegung auf Bildebene ab, wobei der Verschiebungsvektor $f_{I_1,I_2} = [v_x\ v_y]^T$ die Bewegung zwischen dem Punkt $\tilde{x} = [x\ y]^T$ und seinem korrespondierenden Bildpunkt $\tilde{x}' = [x'\ y']^T$ beschreibt. Daraus folgt die Gleichung $\tilde{x}' = \tilde{x} + f_{I_1,I_2}$.

Interpolation des Tiefenflusses Im nächsten Schritt wird das entsprechende Bildpaar D_1, D_2 des Tiefensensors herangezogen und die Gleichung wie folgt erweitert. Zuerst wird $z_{D_1}(\tilde{x})$ als Tiefenwert in Pixeleinheiten aus den Koordinaten von $\tilde{x}$ und sein korrespondierender Gegenpart $z'_{D_2}(\tilde{x}')$ bestimmt. Da das FlowNet2 die Bewegung im Subpixelbereich prädiziert, wird z' durch bilineare Interpolation bestimmt. Die Unsicherheit über die gemessene Tiefe nimmt üblicherweise mit zunehmendem Abstand zur Kamera zu. Um dies zu berücksichtigen, wird eine Tiefengewichtung μ nach Herbst *et al.* [102] mit berücksichtigt, welche die Tiefe mit dem optischen Fluss gewichtet. Es folgt $\mu = \frac{\sigma_z(1)}{\sigma_z(Z)}$ mit $\sigma_z(Z)$ als Tiefenauflösung für die Entfernung Z. Die Unsicherheit wächst bei gängigen RGB-D-Kameras wie der Kinect etwa quadratisch mit der Entfernung. Es ergibt sich für jedes Pixel ein Bewegungsvektor für die Tiefe von $v_z = \delta z = \mu\ z' - \mu\ z$, welcher in Summe als Maske dargestellt werden kann (siehe Abbildung 4.3) und zur Erweiterung des optischen Flusses f zum Szenenfluss eingesetzt wird.

Erzeugung Szenenfluss Der Szenenfluss s wird definiert als

$$s_{I_1,I_2,D_1,D_2} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}. \tag{4.1}$$

Die Bewegungspunkt-Korrespondenzbeziehung zwischen dem Szenenpunkt $p_{I_1,D_1} = [x\ y\ z]^T$ und p'_{I_2,D_2} ist folglich

$$\begin{bmatrix} x' \\ y' \\ \mu\ z' \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} x \\ y \\ \mu\ z \end{bmatrix}. \tag{4.2}$$

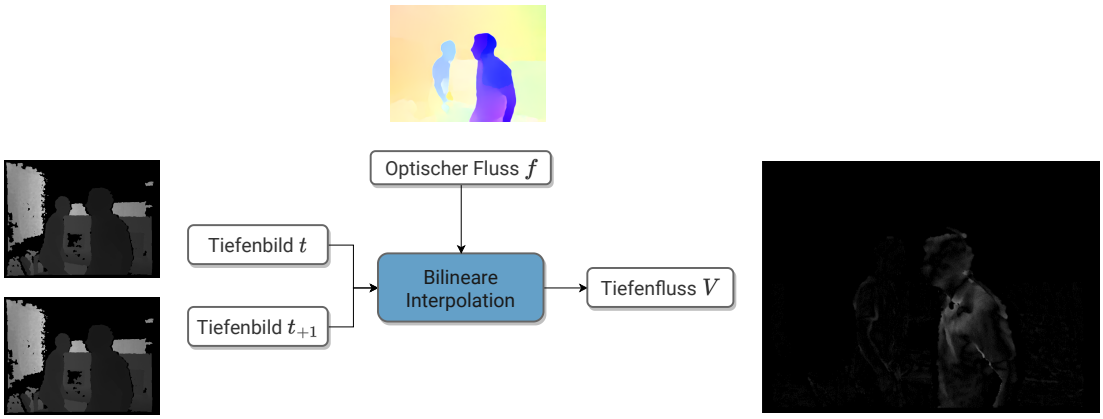


Abbildung 4.3: Generierung der Tiefenfluss-Karte durch bilineare Interpolation auf Basis des optischen Flusses.

Die resultierende Szenenflusskarte enthält jedoch Lücken, wenn der optische Flussvektor die Bildgrenzen überschreitet oder der Tiefensensor keine Distanzinformationen für bestimmte Pixel liefert. Daher wird die Tiefenkarte mit einem Medianfilter geglättet, um kleine Lücken zu verkleinern oder sogar zu schließen. Da sich dennoch Lücken in der Szenenflussmaske ergeben können, wird der vorgeschlagene Ansatz unter dieser Bedingung weiterhin als *semi-dense* bezeichnet.

4.2.2 Ermittlung einer projektiven Transformationsmatrix

Der in Abschnitt 4.2.1 erzeugte Szenenfluss zeigt auf, wie sich die Umgebung zwischen zwei konsekutiven Aufnahmen auf der Bildebene verändert. In einer statischen Welt würden diese Änderungen ausschließlich durch die Bewegung der Kamera hervorgerufen werden. Befinden sich dynamische Objekte in der Umgebung, erzeugen diese Bewegungsvektoren, welche sich in ihrer Intensität und Ausrichtung von den kamerainduzierten Bewegungsvektoren unterscheiden. Hierbei wird die Randbedingung eingeführt, dass die in der Umgebung befindlichen dynamischen Objekte eine kleinere Fläche auf der Bildebene einnehmen als der restliche statische Hintergrund. Abbildung 4.4 veranschaulicht diesen Zusammenhang mit dem durch die Eigenbewegung hervorgerufenen Bewegungsfluss in Grün und den Bewegungsvektoren von dynamischen Objekten in Orange.

Als Nächstes wird eine geeignete Transformationsmatrix gesucht, die diese Eigenbewegung (in Grün) in Bezug auf Rotation und Translation aus den Bewegungsvektoren abbildet. Dies wird durch eine projektive Transformation realisiert:

$$p'_i = H p_i = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{34} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ \mu\ z \\ 1 \end{bmatrix} \tag{4.3}$$

H ist eine 4×4-Transformationsmatrix mit 15 Freiheitsgraden ($h_{44} = 1$). Sie kann mit der Anzahl P von 5 Punktkorrespondenzen $p_{I_1,D_1} \leftrightarrow p'_{I_2,D_2}$ mit homogenen Punktkoordinaten $[x\ y\ z\ 1]^T$ bestimmt werden. Zur Lösung der Gleichung kann die Methode der kleinsten Quadrate verwendet werden.

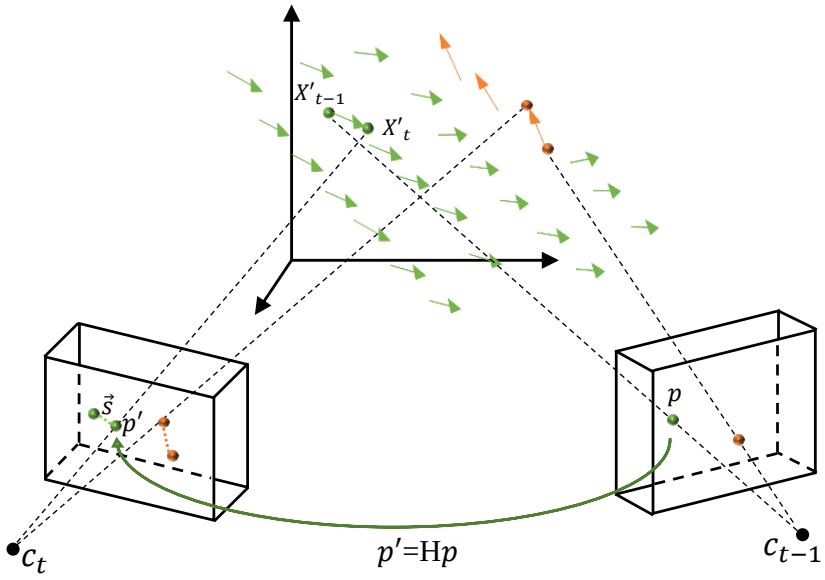


Abbildung 4.4: Geometrische Beziehung zwischen der Ansicht zweier räumlich-zeitlich aufeinanderfolgender Bildaufnahmen. Die dreidimensionale Bewegung der projizierten Punkte wird mithilfe einer Transformationsmatrix modelliert. Die meisten Szenenflussvektoren werden durch die Kamerabewegung zwischen Position c_t und Position c_{t-1} verursacht. Vektoren von dynamischen Objekten (orange) neigen dazu, auffällige Bewegungsmuster aufzuzeigen. Sie gilt es aus der Verarbeitungspipeline zu exkludieren. © 2020 IEEE

Dieser Ansatz ist aber sehr empfindlich gegenüber Ausreißern, die in diesem Fall als Bewegungsvektoren der dynamischen Objekte vorhanden sind. Deshalb wird die robustere RANSAC-Methode genutzt, um Ausreißer nicht in die gewünschte Transformationsmatrix einzubeziehen und lediglich die Eigenbewegung herauszuarbeiten.

Generierung von Transformationen Unter der konservativen Annahme, dass mindestens 55% des beobachteten Hintergrunds statisch ist und die minimale Stichprobengröße von $s = 5$ Punktkorrespondenzen verwendet wird, ergibt sich bei einer Ziel-Wahrscheinlichkeit von $p = 0,98$, um einen Satz ohne Ausreißer zu finden, folgende minimale Anzahl von RANSAC-Iterationen:

$$n = \frac{log(1 - 0.98)}{log(1 - (1 - 0.55)^5)} \approx 75 \tag{4.4}$$

Es wird eine präemptive RANSAC-Methode mit der Beschränkung der Anzahl über $n = 75$ Modellschätzungen gewählt. Somit ergibt sich eine Wahrscheinlichkeit von 98 Prozent, eine gültige Probe aus einem statischen Hintergrund ohne Ausreißer auf dynamischen Objekten zu finden. Die Modelle werden verifiziert, indem das Bild in 3×4 Gitterzellen unterteilt wird und zehn Zufallsstichproben aus jeder dieser Zellen entnommen werden. Daraus ergibt sich ein Verifizierungssatz von 120 Stichproben. Die Größe des Gitters wurde so gewählt, dass alle Teile des Bildes erfasst werden und die Stichproben gleichmäßig verteilt sind. Jedes geschätzte Modell wird anhand eines Schwellenwerts bewertet und nach der Anzahl der Ausreißer geordnet.

Evaluierung der modellierten Transformationsmatrizen Zunächst wird der Reprojektionsfehler des Modellkandidaten H_k gegenüber den Verifizierungs-Stichproben berechnet:

$$\varepsilon = H_k p - p' \tag{4.5}$$

ε ist der Reprojektionsfehler zwischen dem prädizierten $\tilde{p'}$ und dem gegebenen p' des Pixels x' und wird als Vektor $\varepsilon = [\tilde{v}_x\ \tilde{v}_y\ \tilde{v}_z\ 1]^T$ definiert. Als euklidischer Abstand formuliert, bestimmt dieser den Endpunktfehler (EPE) und zeigt auf, wie nahe der projizierte Fluss der erzeugten Szenenflusskarte ist.

Zur Beurteilung wird die Metrik

$$J(s_i) = \left\{ inlier \mid ||\varepsilon_{H_k}(p \leftrightarrow p')|| < \tau \right. \tag{4.6}$$

verwendet. Um zu überprüfen, ob der getestete Flussvektor mit der Eigenbewegung konsistent ist, wird ein adaptiver Schwellenwert τ verwendet, der wie folgt definiert ist:

$$\tau = \begin{cases} \sigma(\hat{S}), & \text{if } \sigma(\hat{S}) \geq \gamma \\ \gamma, & \text{otherwise} \end{cases} \tag{4.7}$$

mit $\hat{S} = \{||s_1||, ||s_2||, ..., ||s_N||\}$.

Der Schwellenwert wird bestimmt durch die Standardabweichung $\sigma(\hat{S})$ des Szenenflusses in Bezug auf den Median über alle Flussvektoren in l_2-Norm. Der Median wird anstelle des Mittelwerts eingesetzt, um den Einfluss von Ausreißern zu vermindern und eine konsistente Aussage über die Verteilung der Kameraflussgeschwindigkeiten zu erhalten. Die Standardabweichung kann sich in Szenen mit wenig dynamischen Objekten und geringer Eigenbewegung schnell verringern. Infolgedessen steigen selbst kleine Reprojektionsfehler über den Schwellenwert, der zur Ausreißerabweisung vieler statischer Hintergrundflussvektoren führt. Abhängig von der Entfernung der Objekte in der Szene, können die kamerainduzierten Flussvektoren in ihrer Geschwindigkeit variieren. Es wird daher eine Max-Funktion eingeführt, um einen minimalen Schwellenwert des Hyperparameters γ zu gewährleisten, der sicherstellt, dass auch Flussvektoren am Rand der Eigenfluss-Verteilung als Ausreißer erhalten bleiben.

Generierung der Zieltransformation Die getesteten Ausreißer werden zusammen mit ihrem Transformationsmodell gespeichert. Nach n getesteten Modellen wird jenes mit der geringsten Anzahl an Ausreißern ausgewählt. Auf Basis dessen wird ein neues Modell generiert, indem es mit all seinen Inliern unter Verwendung der Einzelwertzerlegung (SVD) erneut berechnet wird. Das Ergebnis ist eine geeignete Schätzung des kamerainduzierten Szenenflusses über alle Ebenen, definiert als

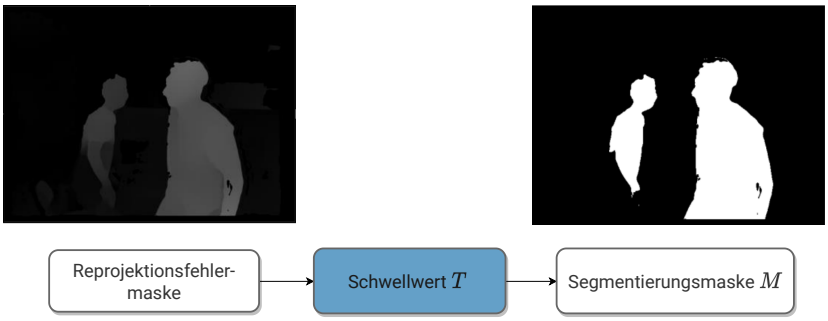


Abbildung 4.5: Erzeugung der Segmentierungsmaske mittels adaptiven Schwellenwerts auf Basis der Reprojektionsfehlermaske.

$$\hat{P}' \approx H\hat{P}, \tag{4.8}$$

mit $\hat{P} = [p_1\ p_2\ ...\ p_n]$ unter Verwendung aller n Korrespondenzen, anstelle der zuvor stichprobenartig gewählten Mindestanzahl von $n = 5$.

4.2.3 Segmentierung dynamischer Bildpixel

Mit dem letzten Schritt zur Verfeinerung der Transformationsmatrix wurde ein Projektionsmodell auf Basis des Szenenflusses bestimmt, welches die Eigenbewegung der Kamera zwischen zwei Aufnahmen widerspiegeln soll. Mithilfe der Transformationsmatrix wird nun eine Projektionsfehlermaske für das gesamte Bild erstellt, indem der Reprojektionsfehler für jeden Szenenflussvektor im Bild unter Verwendung der Gleichung 4.5 berechnet wird. Je größer der Fehler ist, desto wahrscheinlicher gehört der entsprechende Flussvektor zu einem dynamischen Objekt. Folglich kann ein Schwellenwert zur Trennung zwischen dynamischen und statischen Bildpixeln festgelegt werden, um eine Segmentierungsmaske M zu erhalten (siehe Abbildung 4.5).

$$M_t = \left\{ p_t |\ ||\varepsilon_H(p \leftrightarrow p')|| > T \quad \text{mit} \quad T = \alpha \cdot \tau \right. \tag{4.9}$$

Die Schwelle T ist eine Kombination aus dem zuvor verwendeten τ und des Hyperparameters α. Durch die Anpassung von α kann die Strenge der Zuordnung reguliert werden, um festzulegen, welches Maß an Dynamik akzeptiert wird. Je niedriger die Schwelle, desto mehr Pixel mit selbst geringer Bewegung werden als dynamisch segmentiert. Andererseits erhöht ein niedriger Schwellenwert das Risiko, statische Pixel fälschlicherweise als dynamisch zu klassifizieren.

Im Allgemeinen besteht eine ideale Szene aus zwei Ebenen, bei der eine Ebene aus einem oder mehreren dynamischen Objekten besteht und die andere Ebene dem statischen Hintergrund entspricht. In diesem Fall würden alle statischen Szenenflussvektoren den gleichen Abstand zur Kamera aufweisen und weisen daher sehr ähnliche Verschiebungsweiten auf. Die berechnete Homografie würde präzise die Kamerabewegung abbilden und es ermöglichen, dynamische Pixel mit einem strengen Schwellenwert abzulehnen. Je mehr statische Ebenen in der Szene vorhanden sind, desto allgemeiner muss die Homografie geschätzt werden, um alle Bereiche abzudecken. Dies kann zu einer schlechteren Ausgangssituation führen, um geringere dynamische Ausreißer zu eliminieren. In experimentellen

Versuchen haben sich die Einstellungen $\gamma = 1,5$ und $\alpha = 0,7$ als bester Kompromiss herausgestellt, um dynamische Objekte zu segmentieren und gleichzeitig Falsch-Positiv-Klassifikationen statischer Bildbereiche auf ein Minimum zu reduzieren.

4.3 Experimente

Im Folgenden wird der in Abschnitt 4.2 vorgestellte Ansatz einer experimentellen Evaluation unterzogen. Dazu wurde die Methode als Vorverarbeitungsschritt in ein merkmalsbasiertes SLAM-System (ORB-SLAM2) [114] integriert. Im nächsten Schritt wird untersucht, welchen Einfluss der Ansatz zur Segmentierung dynamischer Bildelemente auf die Odometrie des SLAM-Systems hat (Abschnitt 4.3.1). Weiterhin werden zusätzliche quantitative Vergleiche mit anderen Ansätzen für dynamische Umgebungen aus dem Stand der Technik durchgeführt (Abschnitt 4.3.1). Hierfür wurden Methoden ausgewählt, die ähnlich zu dem in dieser Arbeit vorgeschlagenen Ansatz eine Segmentierung von dynamischen Bildbereichen durchführen, ohne Annahmen über potenzielle dynamische Objekte durch semantische Interpretationen zu treffen.

In weiteren Ablationsstudien wird die Parametrierung des eingesetzten RANSAC-Verfahrens analysiert und der Rechenaufwand der einzelnen Teilprozesse des Systems gemessen.

Alle Experimente werden auf dem TUM-RGB-D-Datensatz durchgeführt, der 39 Innenraumsequenzen mit unterschiedlichen dynamischen Umgebungen und Eigenbewegungen enthält (siehe Abschnitt 3.3). Sequenzen mit dem Titel *sitting* zeigen, wie zwei Personen an einem Schreibtisch sitzen und nur geringe Bewegungen durch Gesten und kleine Körperbewegungen erzeugen. In den *walking*-Szenen bewegen sich beide Personen um den Schreibtisch herum und verursachen dadurch anspruchsvolle dynamische Bewegungen. Die Kamerabewegung ist in vier verschiedene Kategorien aufgeteilt: *static* - die Kamera wird statisch in den Händen gehalten, *xyz* - die Kamera bewegt sich entlang der x-, y- und z-Achsen, *rpy* - die Kamera führt eine Rotation entlang der Roll-, Pitch- und Yaw-Achsen durch, *halfsphere (hs)* - die Kamerabewegung bildet eine Halbkugel.

Evaluierungsmetriken Als Evaluationsmetrik wird der Root Mean Squared Error (RMSE) verwendet. Der absolute Trajektorienfehler zwischen der geschätzten Trajektorie und der entsprechenden Grundwahrheits-Trajektorie zum Zeitpunkt i ist definiert als

$$E_i := Q_i^{-1} S P_i, \tag{4.10}$$

wobei S eine Starrkörpertransformation ist, die die geschätzte Trajektorie P auf die wahre Trajektorie Q abbildet. Der Fehler der Sequenz ist die durchschnittliche Abweichung der geschätzten Pose zur echten Pose pro Bild:

$$ATE_{rmse} := \left(\frac{1}{n} \sum_{i=1}^{n} ||trans(E_i)||^2 \right)^{1/2}. \tag{4.11}$$

	ORB-SLAM2 Standard		ORB-SLAM2 ***Vorg. Ansatz***		Differenz	
	Median	SD	Median	SD	Median [%]	SD [%]
static_hs	0,025	(±0,052)	0,018	(±0,039)	28,00	25,00
static_xyz	0,009	(±0,001)	0,011	(±0,001)	-22,22	0
walking_hs	0,511	(±0,173)	0,034	(±0,091)	93,35	47,40
walking_xyz	0,679	(±0,079)	0,018	(±0,123)	97,35	-55,70
walking_rpy	0,801	(±0,128)	0,144	(±0,111)	82,02	13,28
walking_static	0,384	(±0,031)	0,009	(±0,004)	97,66	87,10

Tabelle 4.1: RMSE des absoluten Trajektoriefehlers [m] ohne (*Standard*) und mit (*Vorg. Ansatz*) der Vorverarbeitung zur Eliminierung dynamischer Bildelemente.

4.3.1 Quantitative Evaluation

Zur quantitativen Evaluierung wird zunächst die Implementierung mit und ohne die vorgeschlagene Methode verglichen, um den Einfluss der dynamischen Segmentierung auf die Robustheit und Genauigkeit der Trajektorieeschätzung beurteilen zu können (Abschnitt 4.3.1). Im Anschluss wird das Verfahren mit anderen Methoden aus dem Stand der Technik verglichen (Abschnitt 4.3.1).

Vergleich mit Baseline Tabelle 4.1 zeigt die Ergebnisse des Vergleichs zwischen dem SLAM-Algorithmus ORB-SLAM2[1] [114] mit und ohne die vorgeschlagene Methode als Vorverarbeitungsschritt (*Standard* ↔ *Vorg. Ansatz*). Die dritte Spalte zeigt die Differenz zwischen beiden Methoden, aufgeführt als prozentuale Verbesserung (positive %) oder Verschlechterung (negative %). Aufgrund einer randomisierten Feature-Extraktion der ORB-Merkmale im ORB-SLAM2 ist die Merkmalsextraktion und somit die Trajektorieschätzung nicht deterministisch. Deshalb werden für jede Sequenz zehn Iterationen durchgeführt und anschließend dessen Median als RMSE herangezogen. Zusätzlich wird die Standardabweichung berücksichtigt, um die Robustheit der Ergebnisse einzubeziehen.

Die Ergebnisse des ORB-SLAM2 ohne Methode zur Entfernung dynamischer Bildbereiche (Standard) zeigen einen starken Anstieg der Fehlerrate in hochdynamischen *walking*-Szenen gegenüber den *sitting*-Sequenzen mit eher geringem Maß an dynamischer Bewegung. Dieser Effekt wird durch Szenen mit starken Kamerabewegungen wie *halfsphere* und *rpy* gegenüber eher statischen Kameraszenen wie *static* verstärkt. Die Kombination aus starker Eigenbewegung und einer dynamischen Umgebung führt neben der ungenauen Odometrie-Schätzung zu kurzzeitigen Tracking-Verlusten und gestörtem Merkmalsabgleich, was die Robustheit reduziert und zu einer höheren Standardabweichung führt.

Die SLAM-Methode mit dem integrierten vorgeschlagenen Ansatz (*Vorg. Ansatz*) führt zu einer signifikanten Verbesserung des RMSE in fast allen Sequenzen. Die besten Ergebnisse werden in Szenen mit geringer Eigenbewegung und sehr dynamischen Objekten erzielt. Die Standardabweichung wird unter den gleichen Umständen überwiegend verringert, verschlechtert sich aber in Szenen mit geringem Bewegungsgrad. Der Grund dafür liegt im dynamischen Schwellenwert, der in solchen

[1] https://github.com/raulmur/ORB_SLAM2

Szenarien besonders niedrige Werte annimmt. Dies erfordert eine Feinabstimmung des Parameters γ von (4.7). Aufgrund der insgesamt geringen Dynamik lässt eine weniger rigorose Einstellung von γ jedoch viele dynamische Pixel zu, wodurch der RMSE steigt. Bei einer strikteren Parametrisierung werden dynamische Elemente, aber auch verrauschte statische Elemente herausgefiltert. In diesem Fall nimmt die Standardabweichung der Ergebnisse zu, wie die Ergebnisse in der Tabelle 4.1 für die *sitting*-Sequenzen aufzeigen. Bei den *walking*-Sequenzen kann eine Fehlerreduzierung von bis zu $97,7\%$ erzielt werden. Die Zuverlässigkeit, in mehreren Iterationen die gleichen Ergebnisse zu erhalten, wird in drei von fünf Fällen verbessert. In der *walking-xyz*-Sequenz führt die Methode jedoch zu fluktuierenden Ergebnissen. Die Untersuchung dieses Falles zeigt auf, dass die hohe Standardabweichung durch einige wenige Bilder der Szene verursacht wird, in denen eine Person sich orthogonal zur Bildebene bewegt. Diese Bewegung wird von den interpolierten Bewegungsvektoren aus den Tiefenbildern erfasst, aber aufgrund ihres Gewichtungsfaktors ist die Ausprägung reduziert und liegt genau an der Grenze zur Ablehnung als dynamische Bewegung. Infolgedessen werden die Merkmalspunkte auf der sich bewegenden Person in wenigen Iterationen akzeptiert, was zu einer starken Verzerrung der Trajektorie führt.

Vergleich mit dem Stand der Technik Tabelle 4.2 zeigt die Ergebnisse des vorgeschlagenen Ansatzes im Vergleich zu anderen öffentlich zugänglichen Methoden aus dem Stand der Technik, die auf die Reduzierung des Einflusses dynamischer Umgebungen auf die Odometrie abzielen. Zusätzlich zur Fehlerrate des Trajektorienfehlers wird auch die relative Verbesserung jedes ursprünglichen SLAM-Systems gegenüber seiner verbesserten Version der Bewegungserkennung (BE) untersucht.

Die Ergebnisse zeigen auf, dass die in diesem Kapitel vorgeschlagene Methode die Vergleichs-Algorithmen in der Mehrzahl an Sequenzen übertrifft. Die signifikanteste Fehlerreduzierung wird in Szenen mit überwiegend translatorischer oder allgemein geringer Kamerabewegung in Kombination mit hochdynamischen Umgebungen erzielt. Hierbei stechen insbesondere die Ergebnisse für die Sequenz *walking-xyz* hervor, bei der ein RMSE von $0,018\ m$ erreicht wird und die prädizierte Trajektorie damit $80,65\ \%$ näher an der Grundwahrheit liegt als der zweitbeste Ansatz von Sun *et al*. [90] mit $0,093\ m$. In Sequenzen mit weniger dynamischen Objekten werden ähnliche Ergebnisse wie DynaSlam [115] mit Multi-View-Geometrie-Ansatz erzielt, wie die Ergebnisse in *sitting-halfsphere* und *walking-static* aufzeigen. In *sitting-xyz* wird eine etwas weniger genaue Trajektorie im Vergleich zum Stand der Technik erreicht. In dieser Szene sind die dynamischen Bewegungen sehr gering und haben keinen Einfluss auf die Trajektorienschätzung, wie auch die Ergebnisse der anderen Methoden aus dem Stand der Technik aufzeigen. Ein zusätzlicher Vorverarbeitungsschritt zur Filterung ist daher nicht notwendig und birgt sogar das Risiko, die Ergebnisse zu beeinträchtigen.

Evaluation der RANSAC Parametrisierung In Abschnitt 4.4 wurde die Anzahl der RANSAC Iterationen mit $n \approx 75$ auf der Grundlage der minimalen Stichprobenmenge $s = 5$, der Chance von $p = 0,98$, eine gültige Stichprobe zu finden und der Annahme, dass das Ausreißer-Verhältnis $p_s = 0,55$ ist, berechnet. Diese Näherung kann sehr ungenau werden, wenn dynamische Objekte in die Nähe der Kamera kommen und somit mehr als die geschätzten 45% des Bildes erfassen können. Da der statische Hintergrund nur auf die Hälfte der Bildfläche schrumpft, würde es die gewünschte

Sequenze	Motion Removal[90]			DynaSLAM(G[1])[92]			Wang et al. [91]			**Vorgestellter Ansatz**		
	w/o MD [m]	w/ MD [m]	Diff. [%]	w/o MD [m]	w/ MD [m]	Diff. [%]	w/o MD [m]	w/ MD [m]	Diff. [%]	w/o MD [m]	w/ MD [m]	Diff. [%]
s-hs	0,062	0,047	23,70	0,020	**0,018**	10,00	0,023	0,020	13,04	0,025	**0,018**	**28,00**
s-xyz	0,051	0,048	**4,55**	0,009	**0,009**	0	-	-	-	0,009	0,011	-22,22
w-hs	0,529	0,125	76,32	0,351	0.035	90,03	0,689	0,312	54,72	0,511	**0,034**	**93,35**
w-xyz	0,597	0,093	84,38	0,459	0,312	32,03	0,733	0,305	59,01	0,679	**0,018**	**97,35**
w-rpy	0,730	**0,133**	81,75	0,662	0,251	59,64	0,552	0,498	9,78	0,801	0,144	**82,02**
w-static	0,212	0,066	69,06	0,090	**0,009**	90,00	0,384	0,308	19,71	0,384	**0,009**	**97,66**

Tabelle 4.2: Vergleich des RMSE der absoluten Trajektorienfehler [m] zwischen unterschiedlichen Methoden aus dem Stand der Technik.

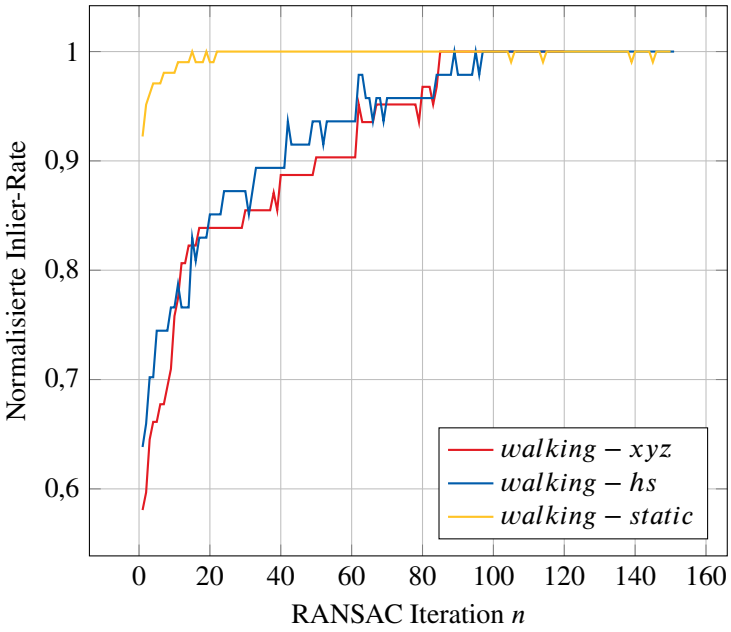


Abbildung 4.6: Inlier-Verhältnis bei unterschiedlicher RANSAC-Iterationsanzahl n.

Iterationszahl n schnell auf 123 anwachsen lassen. Um ein besseres Verständnis für das Verhalten des RANSAC in Bezug auf die Iterationszahl zu bekommen und eine geeignete Einstellung zu finden, wurde das RANSAC-Modell in mehreren realen Szenarien genauer analysiert. Abbildung 4.6 zeigt die Ergebnisse von Sequenzen aus dem TUM-RGB-D-Datensatz mit steigenden Iterationszahlen von $n = 1$ bis $n = 150$ und die daraus resultierende durchschnittliche Anzahl von Ausreißern pro Sequenzdurchlauf. Die Anzahl der Ausreißer kann sich je nach Sequenz ändern. Daher wird eine normalisierte Inlier-Rate in Bezug auf die maximale Anzahl an Ausreißern jeder Sequenz verwendet, was die Suche nach einem bestmöglichen Transformationsmodell anzeigt.

Es zeigt sich, dass in den sehr dynamischen Szenen *walking-xyz* und *walking-halfsphere* die Inlier-Rate von $n = 1$ bis $n = 30$ stark ansteigt und danach einen zunehmenden Abfall der Steigung erfährt. Bei $n = 75$ erhält das beste Modell Unterstützung von etwa 95% der maximalen Anzahl von Inliern. Das Maximum wird zwischen $n = 85$ und $n = 90$ erreicht. In der *walking-static*-Sequenz mit einem geringen Grad an Eigenbewegung erhält der erste Modellkandidat bereits über 90% der Inlier-Unterstützung. Dieses Verhältnis steigt an, bis es bei $n = 15$ das Maximum erreicht.

Ab der Iterationsanzahl von 22 ist das Inlier-Verhältnis stabil auf seinem Maximum. In dieser Szene würde somit $n = 22$ ausreichen, um die höchste Wahrscheinlichkeit zu erreichen, eine optimale Transformationsmatrix zu finden. Das vorgeschlagene $n = 75$ ist daher in dieser speziellen Szene sehr konservativ. In Sequenzen mit mehr Eigenbewegung der Kamera (*walking-hs, walking-xyz*) kann die Chance, ein besser passendes Modell zu erhalten, mit $n > 75$ bei abnehmender Verbesserungsrate geringfügig steigen, was jedoch mit einer linear wachsenden Verarbeitungszeit als Nachteil einhergeht (wird im Abschnitt 4.3.1 näher erläutert). Daher wird die zuvor vorgeschlagene RANSAC-Iterationszahl von $n = 75$ als weiterhin valider Kompromiss betrachtet, um ein passendes Transformationsmodell in einer angemessenen Zeit zu finden.

Laufzeitanalyse Die durchschnittlichen Rechenkosten für das Generieren eines Homografie-Modells wachsen linear mit zunehmender Iterationszahl. Bei der vorgeschlagenen Anzahl $n = 75$ beträgt der Rechenaufwand für das Finden der Transformationsmatrix 60 ms. Dies bezieht sich nur auf die Modellschätzung und die Berechnung des Szenenflusses. Im Falle eines direkten SLAM muss anschließend eine Segmentierungskarte berechnet werden. Bei merkmalbasiertem SLAM kann effizienter vorgegangen werden, indem nur die Szenenflussvektoren analysiert werden, die auf einem Merkmalspunkt liegen. Die durchschnittlichen Verarbeitungszeiten für jeden Schritt sind wie folgt:

Prozessabschnitt	Ausführungszeit
Modelgenerierung ($n = 75$)	60,0 ms
Erzeugung der Segmentierungsmaske	15,7 ms
Merkmalssegmentierung	1,2 ms

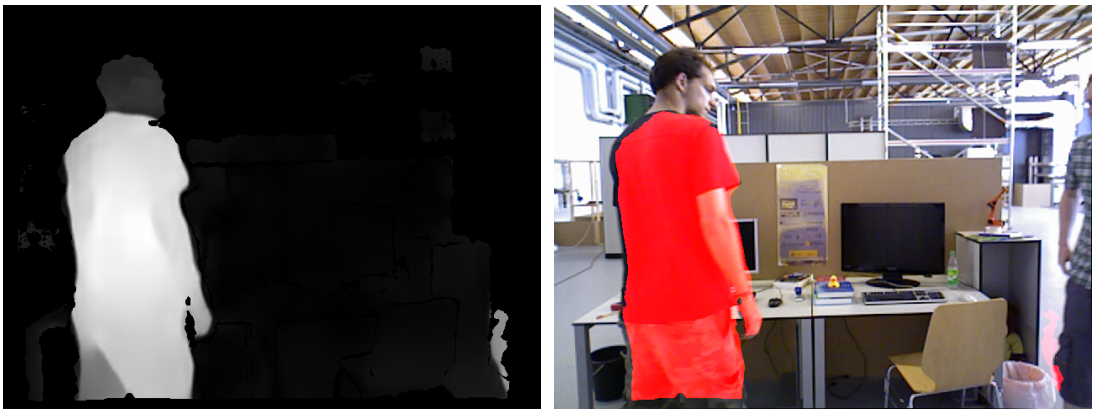
Tabelle 4.3: Übersicht der Prozessdurchlaufzeiten.

Die Erstellung der Segmentierungsmaske besteht aus der Berechnung des Reprojektionsfehlers für jeden gültigen Szenenflussvektor im Bild. Für die dynamische Merkmal-Rückweisung wird ein 3×3-Pixel-Patch um jeden Merkmalspunkt auf dynamisches Verhalten analysiert. ORB-SLAM2 verwendet standardmäßig 1.000 Merkmalspunkte, welche eine Analysezeit von durchschnittlich $1,19$ ms benötigt. Alle Verarbeitungsschritte werden auf einem Intel i7-6850K (6 Kerne @ 3.6GHz) mit 32Gb RAM durchgeführt und mit der EIGEN-Bibliothek optimiert. Zusätzliche Rechenkosten entstehen durch die Erzeugung des optischen Flusses.

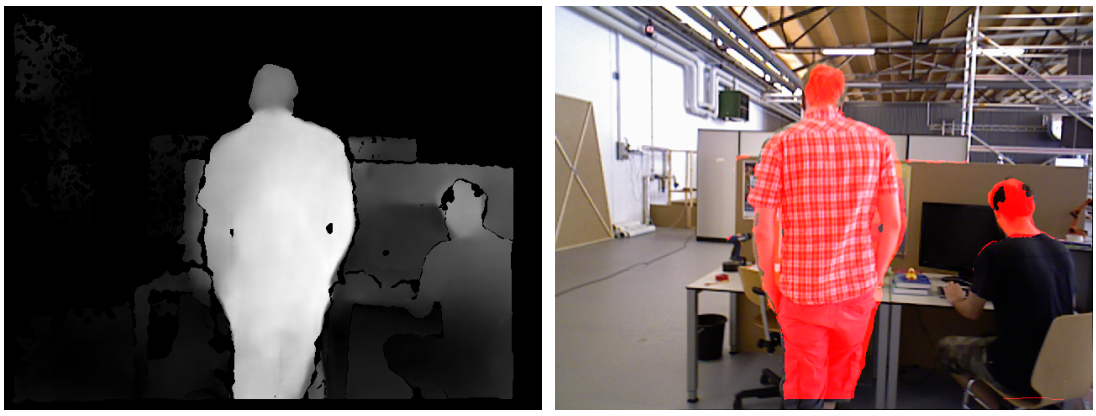
4.3.2 Qualitative Evaluation

Im vorangegangenen Abschnitt wurde der Einfluss der vorgeschlagenen Methode auf die Odometriepräzision eines SLAM-Algorithmus analysiert. Um einen tieferen Einblick in das Segmentierungsverhalten des Ansatzes zu erhalten, werden in diesem Abschnitt einige qualitative Ergebnisse präsentiert.

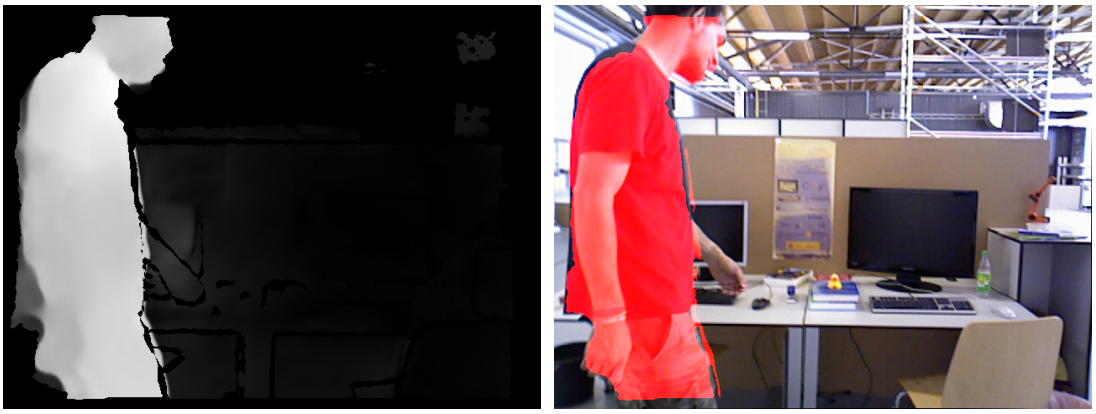
Abbildung 4.7 zeigt einige exemplarische Ausschnitte aus der getesteten *walking-xyz* Sequenz, mit der normalisierten Reprojektionsfehlermaske auf der linken Seite und der resultierenden Segmentierungsmaske auf der rechten Seite. Die Abbildungen veranschaulichen, dass die vorgeschlagene



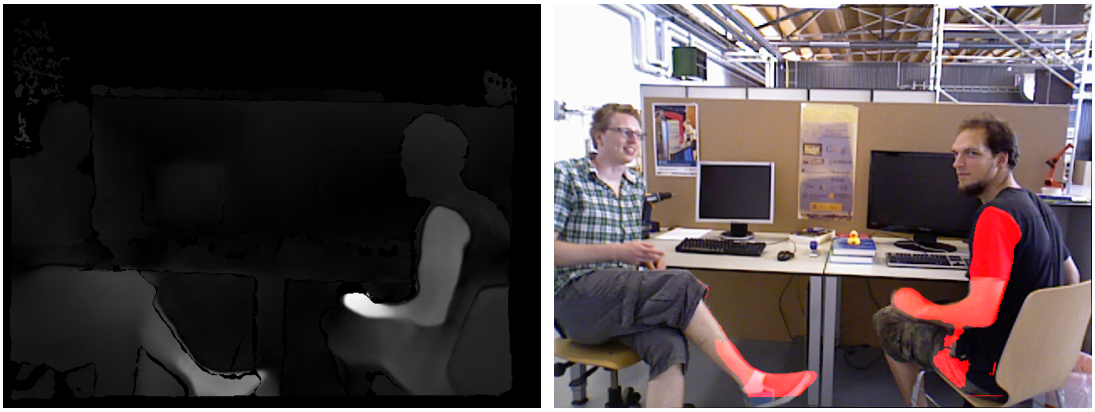
(a)



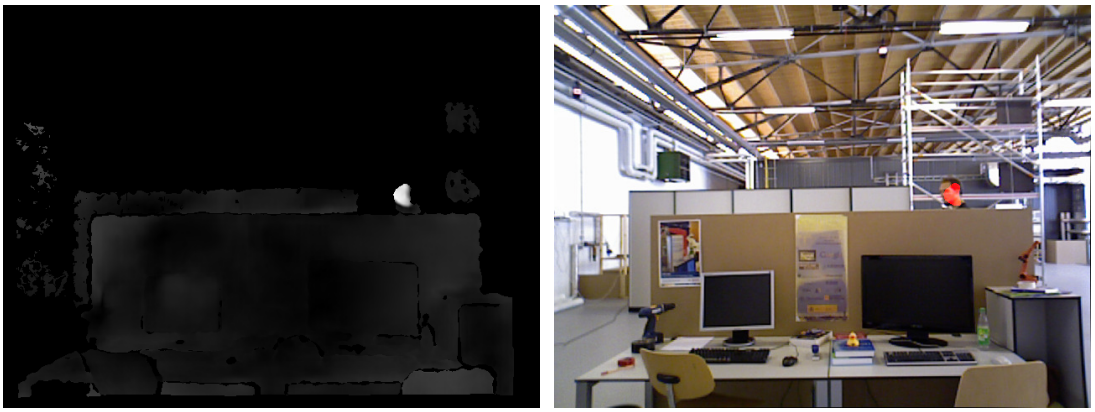
(b)



(c)



(d)



(e)

(f)

Abbildung 4.7: Qualitative Ergebnisse des vorgestellten Ansatzes mit Bildpaaren von normalisierten Reprojektionsfehlerkarten und den daraus resultierenden Segmentierungsmasken. © 2020 IEEE.

Methode in der Lage ist, dynamische Bildteile, hier in Form der Personen, auf der Grundlage der Reprojektionsfehlermaske präzise zu segmentieren. Hierbei hebt sich die Methode auch deutlich von Ansätzen ab, die auf Objektdetektoren basieren und somit keine pixelbasierte Bewertung vornehmen. Dies zeigt sich insbesondere in Abbildung 4.7d, in der die Person auf der linken Seite ihren Fuß bewegt und die Person auf der rechten Seite nur leichte Bewegungen des Armes durchführt. Bei einer Objektdetektor-basierten Methode würden in diesem Fall konsequenterweise beide Personen in Gänze segmentiert und somit wichtige Referenzbereiche zur Orientierung entfernt werden. Mit dem vorgeschlagenen Ansatz werden hingegen nur die dynamischen Zielbereiche segmentiert. Diese Effizienz erklärt die erzielte Genauigkeit und Robustheit in der quantitativen Evaluation in Abschnitt 4.3.1, welche den derzeitigen Stand der Technik signifikant übertreffen. Ein weiteres Beispiel hierfür liefert Abbildung 4.7e, in der sogar ein weit entfernter Kopf als dynamischer Bildbereich segmentiert werden kann.

In Abbildung 4.7b wird die Herausforderung deutlich, eine passende Homografie zu ermitteln. Zu diesem Zeitpunkt der Sequenz findet die überwiegende Bewegung in der Tiefe statt (Person bewegt sich von der Kamera weg), welche nur bedingt durch den optischen Fluss erfasst werden kann und auch im Szenenfluss unterrepräsentiert wird. Als Resultat umfasst die Homografie auch einige Teile des statischen Hintergrundes in Form des Tisches. Hierbei wird die Signifikanz der Wahl eines geeigneten Schwellenwerts deutlich, um eine erfolgreiche Segmentierung aus der Fehlerprojektionsmaske herauszuarbeiten. Die Segmentierungsmaske zeigt, dass dies durch den adaptiven Schwellenwert gelungen ist und nur ein dünner Randbereich des Tisches zurückgewiesen wird.

Eine weitere Herausforderung zeigt Abbildung 4.7f, in der eine schwarze Lücke unter dem Kopf der erfolgreich segmentierten Person vorhanden ist. In diesem Bereich liefert die Tiefenkarte keine Tiefeninformationen und konnte daher nicht auf dynamische Objekte analysiert werden.

4.4 Diskussion

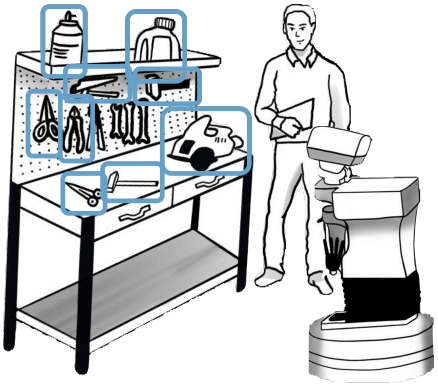
In diesem Kapitel wurde eine neue Methode zur pixelweisen Segmentierung dynamischer Bildelemente vorgestellt, welche die Verbesserung der Lokalisierung in dynamischen Umgebungen ermöglicht und damit eine der fundamentalsten Herausforderungen in der autonomen mobilen Robotik angeht. Der Ansatz kann in Umgebungen ohne a-priori-Wissen über mögliche dynamische Objekte eingesetzt werden und ist aufgrund seiner pixelweisen Segmentierungsprozedur sowohl in merkmalsbasierten SLAM als auch in bildbasierten SLAM-Methoden anwendbar.

In quantitativen Experimenten auf öffentlichen Datensätzen wurde nachgewiesen, dass der Lokalisierungsfehler mit der vorgestellten Methode um bis zu 98% gesenkt werden konnte. Dabei übertrifft die Methode andere vergleichbare Ansätze aus dem Stand der Technik in mehreren Testsequenzen.

Eine einschränkende Randbedingung ist die Wahl geeigneter Hyperparameter für die Schwellenwertberechnung von 4.7, da eine optimale Verfeinerung a priori-Informationen über den Grad der Dynamik in der Umgebung und der Eigenbewegung erfordern würde. Die dynamische Anpassung der Parameter auf der Grundlage eines Messfensters der Ablehnungsraten in den letzten Frames könnte

eine mögliche Verbesserung für zukünftige Arbeiten sein, um einer Über- oder Untersegmentierung entgegenzuwirken.

Weiterhin entspricht der Prozessschritt der Modellgenerierung durch das aufwendige RANSAC-Verfahren einem Flaschenhals in der Verarbeitungspipeline. Zukünftige Optimierungen in diesem Bereich können dadurch erheblich zu einer Verschlankung des Systems beitragen und den Einsatz in Ressourcen-limitierten Hardwarestrukturen erleichtern.



KAPITEL 5

Semantische Umgebungserfassung

Die visuelle, simultane Lokalisierung und Kartierung nimmt in vielen autonomen mobilen Systemen eine Schlüsselrolle ein. Üblicherweise basieren die eingesetzten Methoden auf lokalen geometrischen Merkmalen wie Punkten [116, 117, 118], Linien [119, 120, 121, 122, 123] oder Oberflächenfeldern [124]. Andere verwenden das Kamerabild in ihrer Gesamtheit, um Bildintensitäten abzugleichen [125, 126, 127]. In beiden Fällen ist die erzeugte Karte eine ausschließlich geometrisch verankerte Darstellung der Umgebung. Dadurch sind darauf aufbauende Algorithmen auf grundlegende Navigations- und Hinderniserkennungsaufgaben limitiert.

In jüngerer Vergangenheit hat sich innerhalb des SLAM-Bereichs ein neues Forschungsfeld etabliert, das sich darauf konzentriert, zusätzliche semantische Informationen in die Darstellung der Umgebung einzubinden. Dies führt zu einem bedeutenden Fortschritt in der Robotik und der autonomen Navigation, der es erlaubt, Robotern ein tieferes Verständnis ihrer Umgebung zu vermitteln, indem nicht nur die geometrische Struktur des Raumes kartiert, sondern auch die Bedeutung und Funktion der darin enthaltenen Objekte erfasst wird.

Durch die Anreicherung der Karten mit semantischen Informationen können SLAM-Systeme komplexe Szenarien besser verstehen und ermöglichen dadurch die Generierung und Ausführung intelligenterer Handlungsmechanismen, z B. für die Mensch-Roboter-Interaktion [128] und die Objektmanipulation [129].

Das Kapitel beginnt mit einer Diskussion über den derzeitigen Stand der Technik (Abschnitt 5.1). Darauf aufbauend wird im nächsten Abschnitt eine neue Methode zur echtzeitfähigen, semantischen Kartierung vorgeschlagen (Abschnitt 5.2). Es folgt eine ausführliche Evaluation über die quantitative und qualitative Performanz auf öffentlichen Datenbanken (Abschnitt 5.3). Daraufhin wird die Methode auf einem echten Roboter implementiert und eine weitere qualitative Evaluation in

realer Umgebung durchgeführt (5.4). Das Kapitel schließt mit einer Diskussion der Limitationen (Abschnitt 5.5) und einer abschließenden Zusammenfassung (Abschnitt 5.6).

Forschungsbeitrag

» Es wird eine neuer, echtzeitfähiger Ansatz zur semantischen Kartierung vorgeschlagen, welcher geometrische Karten mit bedeutungs- und funktionsbezogenen Objekten erweitert.

» Es wird ein sukzessives Tracklet-Verfahren angewendet, um dynamische Objekte und fehlerhafte Objektdetektion in einem vorgelagerten Prozess auszuschließen.

» Die vorgeschlagene semantische Kartierung ermöglicht durch das Tracking von Objekten ein flankierendes Korrekturverfahren zur Optimierung der Odometrie von SLAM.

5.1 Verwandte Arbeiten

Die Verwendung und Einbettung semantischer Informationen in SLAM-Algorithmen wurde in vorangegangenen Arbeiten mittels unterschiedlicher Ansätze untersucht. Eine Strategie zielt auf die Integration von Objektinformationen ab, bei der die vorhandenen geometrischen Kartenpunkte mit semantischen Relationen verknüpft werden [130, 131, 132, 133]. Andere Ansätze fügen vollständige, neue Objektinstanzen in die Karte ein [134, 135]. Ein dritter Ansatz, oft bezeichnet als *object-slam* [136, 137, 138, 139], basiert ausschließlich auf semantischen Objektlandmarken, ohne die Einbeziehung herkömmlicher Bildmerkmale. Diese Methoden sind jedoch aufgrund der geringeren Anzahl an Landmarken besonders anfällig für inkorrekte Datenassoziation und erfordern Szenen mit genügend bekannten Objekten und präziser Objektlokalisierung, um zuverlässig die Positionsbestimmung durchführen zu können. Andererseits verhält sich objektbasierter SLAM robust gegenüber Fehlern durch Perspektivwechsel und Umgebungen mit homogenen Texturen sowie Oberflächen mit wenig prägnanten geometrischen Merkmalen.

Besonders vielversprechend sind deshalb Methoden, welche die Nutzung von herkömmlichen Merkmalen mit semantischen Landmarken in einem System vereinen. Bernreiter *et al*. [140] wenden eine solche hybride Lösung an, bei der semantische Submaps mit Odometriemessungen verschmolzen werden. Die Submaps werden durch Multi-Hypothesen-Tracking erzeugt und mit einem angepassten Kuhn-Munkres-Algorithmus zugeordnet. Li *et al*. [141] leiten 3D-Quader von Objekten aus einer Sequenz von 2D-Detektionen ab, um eine Relokalisierung bei großen Perspektivwechseln zu ermöglichen. Martins *et al*. [142] erweitern konventionelle geometrische Karten mit semantischen 3D-Formprioritäten, die ähnlich aussehen wie projizierte Objekterkennungen. Im Gegensatz dazu schlagen Bowman *et al*. [143] und Doherty *et al*. [144] vor, semantische Beobachtungen direkt in ein probabilistisches Optimierungsframework einzubinden.

Allen beschriebenen Methoden ist gemein, dass sie CNNs verwenden, um semantische Informationen in Form von Objektdetektion und -klassifikation in den SLAM-Prozess zu integrieren. Diese Ob-

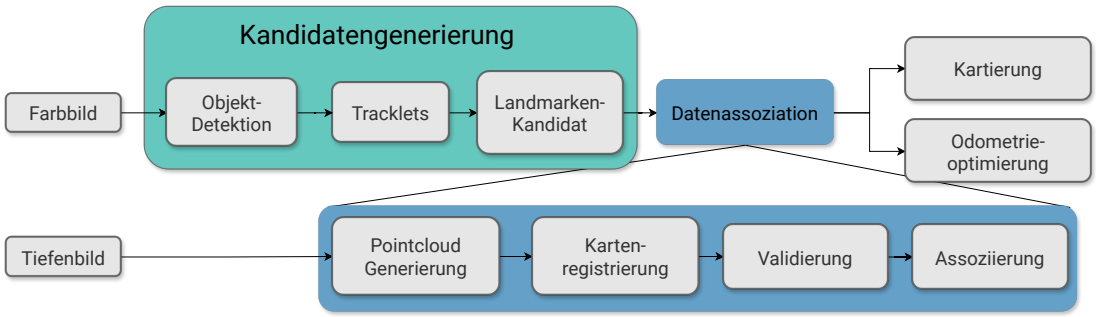


Abbildung 5.1: Überblick über die vorgeschlagene Methode. Zunächst werden 2D-Objektdetektionen auf dem RGB-Bild durchgeführt und als Tracklets im zeitlichen Verlauf mittels IOU-Trackers verfolgt. Robust wiederkehrende Detektionen werden als Landmarken-Kandidaten vorgeschlagen, um sie in die Umgebungskarte einzugliedern. Nach einem Validierungsschritt wird jeder Kandidat in einem Datenassoziationsschritt weiterverarbeitet. Die resultierenden Landmarken werden als Kugeln abgebildet und als Randbedingung zur Verbesserung der Posenschätzung eingesetzt.

jektdetektoren erzeugen nicht-gaußsche und diskrete Zufallsvariablen und bergen daher das Risiko, dass falsche Objektklassifikationen in den Datenassoziationsprozess einfließen. Die Auswirkung fehlerhafter Erkennungen kann zu einem erhöhten Rechenaufwand führen oder sogar die gesamte Posenschätzung verzerren. Während Bowman *et al.* [143] von vornherein falsch-negative und falsch-positive Objektprädiktionen vernachlässigen, verwenden Li *et al.* [141] Objektdetektionen, die von beiden Kameras des Stereoaufbaus prädiziert werden. Bernreiter *et al.* [140] beziehen Falsch-Detektionen explizit als eine mögliche Submap-Option mit ein, mit dem Nachteil einer erhöhten Rechenkomplexität. Nicholson *et al.* [136] hingegen nehmen die Datenzuordnung als gegeben an.

5.2 Ansatz zur objektiv-basierten semantischen Kartierung

Im Folgenden wird ein eigener Ansatz zur objektbasierten, semantischen Kartierung vorgeschlagen. Die Objektinstanzen werden hierbei mittels eines CNN zur 2D-Objektdetektion erfasst, in den 3D-Raum projiziert und anschließend in die Umgebungskarte registriert.

Den Herausforderungen der Datenassoziation wird mittels eines Vorauswahlverfahrens auf Basis von *Tracklets* – lokal assoziierter Erkennungen – begegnet. Dadurch wird die Generierung semantischer Landmarken auf Basis falscher Objektprädiktionen bereits in einem vorgelagerten Prozess unterdrückt, sodass Rechenaufwand und die Fehlerfortpflanzung in nachfolgenden Verarbeitungsschritten reduziert werden. Weiterhin ermöglicht der Einsatz von Tracklets weitere Analysen über das spartiale Verhalten der detektierten Objekte, um nicht-statisches Verhalten zu erkennen. Dadurch werden Objekte, die für den Lokalisierungsprozess potenziell schädlich sind, zurückgewiesen.

Abbildung 5.1 zeigt eine schematische Übersicht der entwickelten Methode, welche in zwei Prozessschritte unterteilt werden kann:

1. Die Generierung semantischer Objektkandidaten:
   - Objektdetektion
   - Erzeugung von Tracklets aus Objektdetektionen

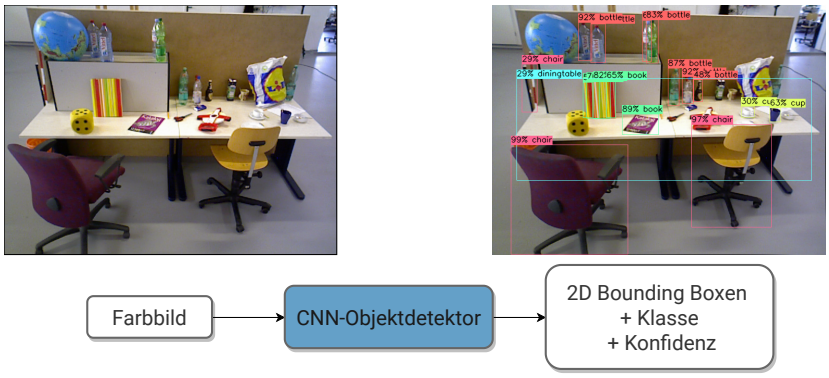


Abbildung 5.2: Exemplarisches Beispiel einer CNN-basierten Objektdetektion.

- Überführung von Tracklets zu Objektkandidaten

2. Die Datenassoziation:
   - Untersuchung von Objektkandidaten nach dynamischem Verhalten
   - Kandidatenlokalisierung
   - Kandidatenassoziation

Flankierende Verarbeitungen umfassen die kontinuierliche (3.) Positionsoptimierung der Landmarken und die (4.) Landmarkenverfeinerung.

Im Folgenden werden die einzelnen Verarbeitungsschritte im Detail beschrieben.

5.2.1 Kandidatengenerierung

Die Kandidatengenerierung befasst sich mit der Erfassung von semantischen Objekten in der Umgebung. Dazu wird zunächst eine 2D-Objektdetektion auf der Bildebene unter Verwendung eines CNNs durchgeführt (Abschnitt 5.2.1). Im Anschluss werden die einzelnen voneinander unabhängigen Detektionen per Tracking-by-Detection-Prinzip in Tracklets zusammengeführt (Abschnitt 5.2.1). Schließlich werden gültige Tracklets als Objektkandidaten aufgenommen (Abschnitt 5.2.1).

2D-Objektdetektion Künstliche neuronale Netze zur Lokalisierung von Objekten in Bildern haben sich in den letzten Jahren durch ihre Robustheit, Genauigkeit und Effizienz als effektives Mittel zur bildbasierten Objektlokalisation durchgesetzt [145, 146, 147, 148, 149, 150]. Deep Learning-basierte Objektdetektoren durchlaufen üblicherweise zwei Prozessschritte: Die Generierung von *region proposals*, Bildbereiche, in denen Objekte vermutet werden und die Klassifikation dieser Bildbereiche. Diese Art von Detektoren wird auch Two-Stage-Detektor genannt [151, 152, 146, 153, 154]. Die You only look once (YOLO) Architektur [148] hingegen vereint Lokalisierung und Klassifizierung in einem gemeinsamen Schritt und erzielt deshalb besonders kurze Verarbeitungszeiten. Hierfür wird das Eingangsbild in ein Grid unterteilt, in dem jede Gridzelle separat auf Objekte untersucht wird. Das Ergebnis sind Bounding-Boxen Prädiktionen mit Klassenzugehörigkeit und Prädiktionskonfidenz

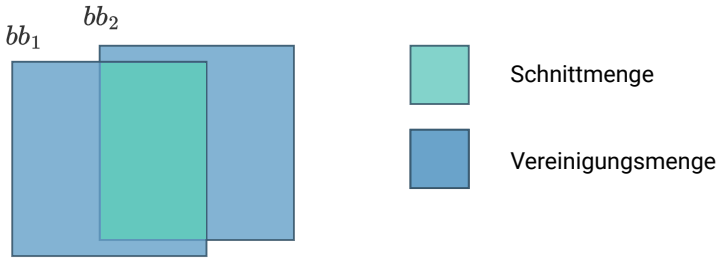


Abbildung 5.3: Schnittmenge und Vereinigungsmenge zweier Bounding-Boxen.

(siehe Abbildung 5.2). Aufgrund der geringen Inferenz wird im vorgeschlagenen Ansatz ein YOLO-Objektdetektor verwendet, welcher auf dem COCO Datensatz [155] trainiert wurde und in der Lage ist, Objekte aus 80 verschiedenen Klassen zu detektieren.

Erzeugung von Tracklets aus Objektdetektionen Die Präzision von CNN-Prädiktionen kann stark von den Umgebungsbedingungen beeinflusst werden. Ungewöhnliches Aussehen von Objekten und unzureichende Beleuchtung können die Genauigkeit verringern und die Zahl der falsch positiven Ergebnisse und Falsch-Klassifizierungen erhöhen. Für den Einsatz bei einem mobilen Roboter bedeutet dies, dass bei der Erkundung der Umgebung die Objekte im Raum durch unterschiedliche Perspektiven inkonsistente Detektionsergebnisse hervorrufen können. Für Kartierungszwecke kann dies zur Einbeziehung nicht vorhandener oder falsch klassifizierter Objekte, zu Verzerrungen der Datenzuordnung und schließlich zu übermäßigen Rechenkosten führen. Bei der Verwendung semantischer Objektinstanzen zur Posenschätzung oder -optimierung kann zudem die korrekte Bestimmung der Trajektorie gehemmt werden. Um diese negativen Auswirkungen zu vermeiden, wird ein Auswahlprozess auf niedriger Verarbeitungsebene durchgeführt, um Fehlprognosen, potenzielle dynamische Objekte und andere inkonsistente Objekterkennungen zu detektieren und zu eliminieren, bevor sie an den Prozess der Datenzuordnung weitergeleitet werden. Genauer wird ein Tracking-by-Detection-Paradigma angewendet, welches mittels eines Intersection-Over-Union-Trackers (IOU-Tracker) [156] implementiert wird.

Der IOU-Tracker assoziiert die Bounding Boxen von Objekt-Prädiktionen mit hohem Jaccard-Index J und übereinstimmendem Klassenlabel über zeitlich aufeinanderfolgende Eingangsbilder. Der Jaccard-Index dient als Kennzahl zur Bestimmung der Ähnlichkeit zweier Bounding Boxen bb_1 und bb_2, indem er das Verhältnis zwischen ihrer Schnittmenge und Vereinigungsmenge berechnet (siehe Abbildung 5.3).

$$J(bb_1, bb_2) = \frac{|bb_1 \cap bb_2|}{|bb_1 \cup bb_2|} \tag{5.1}$$

Die Detektionen in aufeinanderfolgenden Bildern mit geringfügigen Änderungen des Blickwinkels, die einen hohen Jaccard-Koeffizienten aufweisen, legen nahe, dass sie von demselben realen Objekt im Raum auf das Kamerabild projiziert werden, da ihre Bildposition und -abmessungen ähnlich sind.

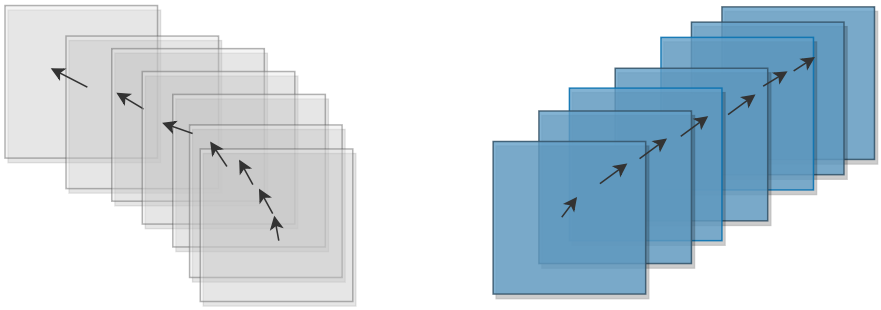


Abbildung 5.4: Aufbau eines Tracklets durch Assoziierung zeitlich aufeinander folgender Detektionen mit gleicher Klassenzugehörigkeit.

Auf diese Weise können schrittweise Tracklets von räumlich und zeitlich assoziierten Objekterkennungen erzeugt werden, bei denen jede zusätzliche *Messung* (Detektion) die Wahrscheinlichkeit einer korrekten Lokalisierung und Klassenzugehörigkeit eines Objekts erhöht. Weiterhin ermöglicht dieses Verfahren, die Abbildung dynamischer Objekte zu vermeiden, da deren Jaccard-Index aufgrund der Bewegung der Bounding Boxen dazu neigt, unter die Akzeptanzschwelle zu fallen.

Überführung von Tracklets zu Objektkandidaten Wenn die Länge N eines Tracklets eine bestimmte Mindestgröße erreicht, wird sie als gültige Erfassung des Objekts akzeptiert. An diesem Punkt wird das gesamte Tracklet als Landmarkenkandidaten vorgeschlagen und an den nächsten Verarbeitungsschritt weitergegeben.

5.2.2 Datenassoziierung

Die Datenassoziierung der Objektkandidaten kann in drei Prozessschritte untergliedert werden. Zuerst folgt eine erste statistische Analyse auf konsistente Messdaten, um dynamische Objekte und ungleichmäßige Messungen zurückzuweisen (Abschnitt 5.2.2). Im Anschluss wird auf Grundlage der unterschiedlichen Messungen innerhalb des Tracklets ein probabilistischer Ansatz zur Lokalisierung des Objektmittelpunkts angewendet (Abschnitt 5.2.2). Im letzten Schritt wird die tatsächliche Assoziierung mit der Roboterkarte durchgeführt, um zu entscheiden, ob das Objekt einer neuen Landmarke entspricht oder einer bereits kartierten Landmarke zugeordnet werden muss (Abschnitt 5.2.2).

Untersuchung von Objektkandidaten nach dynamischem Verhalten Die Tatsache, dass mit einem Tracklet mehrere Messungen des gleichen Objekts durchgeführt werden, ermöglicht es, weitere analytische Verfahren durchzuführen. Dabei soll jeder Kandidat auf dynamisches und inkonsistentes Verhalten (unter anderem wechselhafte Objektgröße verursacht durch Verdeckungen) untersucht werden. Hierzu wird zunächst eine quaderförmige Punktwolke für jede Messung auf der Dimensionsgrundlage der prädizierten Bounding Box erzeugt und in die Weltreferenz auf Basis der Robotertrajektorie projiziert. Im nächsten Schritt wird der Mittelpunkt für jede Wolke bestimmt, um daraus die mittlere absolute Abweichung (MAD) D der Mittelpunkte über alle Messungen des

Tracklets zu berechnen:

$$D = \frac{1}{n} \cdot \sum_{i=1}^{n} |x_i - \bar{x}| \tag{5.2}$$

Eine hohe MAD weist auf eine instabile Position hin und führt zur Ablehnung des Kandidaten. Sie wird in der Regel durch dynamische Objekte oder wechselnde Verdeckungen des Objekts entlang der Messaufnahme verursacht. In diesen Fällen können keine zuverlässigen Rückschlüsse auf die korrekte Lokalisierung gezogen werden, um eine statische Landmarke zu erstellen.

Kandidatenlokalisierung In einem nächsten Schritt wird der Objektmittelpunkt des Kandidaten als Position zur Einbettung in die Umgebungskarte ermittelt. Der Mittelpunkt sei hierbei definiert als dreidimensionaler Punkt $\mathbf{X} = (X, Y, Z)^T$ im Raum, welcher mit der Projektionsfunktion f an der Stelle $x = f(\mathbf{X})$ auf die Bildebene projiziert wird. Dieses Prinzip wird in Abbildung 5.5 veranschaulicht. Um seine Wahrscheinlichkeit zu modellieren, wird der Ansatz von Hartley und Zisserman [157, 141] vorgeschlagen, welcher das Messrauschen als Gaußsches Rauschen mit dem Mittelwert 0 einführt. Das Messmodell erweitert sich dementsprechend zu $x = f(\mathbf{X}) + \eta$ mit $\eta \sim \mathcal{N} = (\mathbf{0}, \Sigma)$. Die Wahrscheinlichkeit, dass die gemessenen Mittelpunkte $x_{1:t} = \{x_1, x_2, x_3, ..., x_t\}$ der Tracklets in den 3D-Punkt $\mathbf{X}$ projiziert werden, kann wie folgt ausgedrückt werden:

$$p(x_{1:t}|\mathbf{X}) = \frac{exp(-\frac{1}{2}(f(\mathbf{X}) - x_{1:t})^T \Sigma^{-1} (f(\mathbf{X}) - x_{1:t}))}{\sqrt{(2\pi)^2 |\Sigma|}} \tag{5.3}$$

Die Posteriorverteilung von $\mathbf{X}$, die zu den Objektdetektionen des Tracklets $x_{1:t}$ führt, ist nach Bayes

$$p(\mathbf{X}|x_{1:t}) = \frac{p(x_{1:t}|\mathbf{X})P(\mathbf{X})}{p(x_{1:t})}. \tag{5.4}$$

Unter der Annahme einer gleichmäßigen Prioritätsverteilung und unabhängiger Messungen erhält man

$$p(\mathbf{X}|x_{1:t}) = \prod_{t=1}^{T} p(x_t|\mathbf{X}), \tag{5.5}$$

wobei das Ziel darin besteht, den Objektmittelpunkt $\mathbf{X}^*$ zu finden, der die maximale unnormalisierte *a posteriori*-Wahrscheinlichkeit darstellt.

$$\mathbf{X}^* = \arg\max_{\mathbf{X}} p(\mathbf{X}|x_{1:t}) \tag{5.6}$$

Zu diesem Zweck werden Positionshypothesen der Landmarke erzeugt, indem Stichproben aus $p(\mathbf{X}|x_{1:t})$ gezogen und deren Wahrscheinlichkeit mit Gleichung 5.6 berechnet wird. Zur Lösung

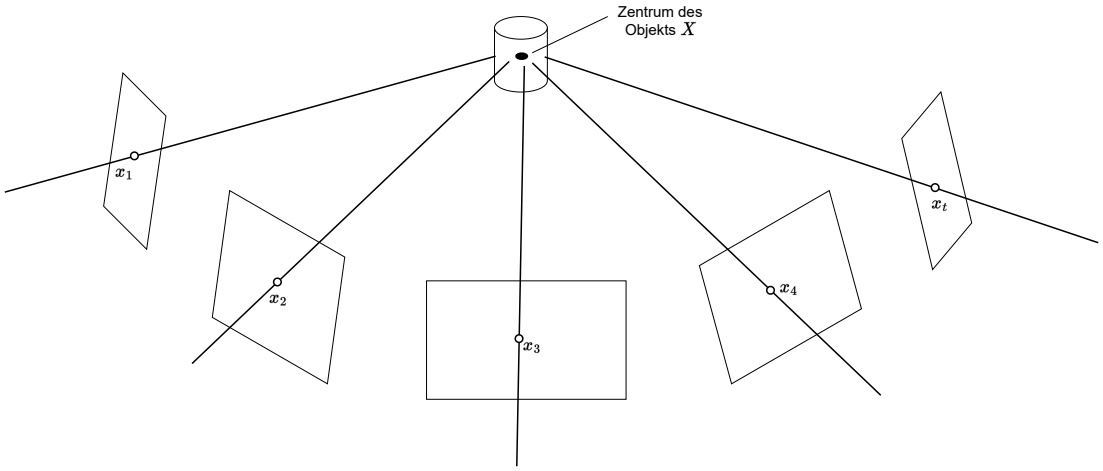


Abbildung 5.5: Illustration der Ausgangssituation zur Kandidatenlokalisierung.

der Gleichung wird zunächst ein 3D-Punkt aus den gegebenen Stichproben als Ausgangspunkt trianguliert und im Anschluss mit einem zufälligen Monte-Carlo-Sampling fortgefahren.

Kandidatenassoziation Im letzten Schritt werden die Kandidaten in die Umgebungskarte integriert. Hierbei zeigen sich zwei mögliche Optionen auf:

- Der Kandidat entspricht einem zuvor bisher nicht kartierten Objekt und kann folglich als neue Instanz in die semantische Karte mit aufgenommen werden.
- Das dem Kandidaten zugrunde liegende observierte Objekt wurde bereits in einer früheren Prozessiteration in die Karte registriert und sollte nicht als neues Objekt eingebunden werden.

Die Prüfung der beiden Optionen ist üblicherweise einer der rechenintensivsten Schritte bei Methoden zur semantischen Kartierung. Zur Minimierung der Durchlaufzeit wird ein schrittweises Verfahren aus

1. einem groben Positionsabgleich mit bestehenden Landmarken in der Umgebung

   und

2. einem feinen Abgleich von Objekten, die als potenzielle Assoziationsobjekte aus Schritt 1 hervorgegangen sind,

durchgeführt.

Letzterer Schritt wird nur in Anspruch genommen, sofern aus Schritt 1 tatsächlich mehrere valide Assoziationskandidaten hervorgegangen sind.

1. **Bestimmung von Landmarken zum Abgleich mittels Validierungsfenster**

   Es wird ein Validierungsfenster, innerhalb dessen eingebettete Objekte als mögliche Assoziationslandmarken klassifiziert werden, eingesetzt. Die Größe des Validierungsfensters wird

dynamisch ermittelt, um die Unsicherheit über die Positionen der Landmarken einzubeziehen. Denn diese ist vollständig abhängig von der Eigenbewegungsschätzung des zugrundeliegenden Lokalisierungsalgorithmus. Um mögliche *Drifts* in der prädizierten Trajektorie der Kamera einzubeziehen, wird die Größe des Validierungsfensters r in Relation zur Periode Δt und der Objektgröße s in *Metern* gesetzt. Δt beschreibt die Zeit in *Sekunden* seit der letzten Assoziation einer bekannten Landmarke.

$$r = \sqrt{\frac{\Delta t}{u} \cdot s} \tag{5.7}$$

Je länger ein neuer Kandidat nicht mit einer bekannten Landmarke assoziiert werden konnte, desto geringer wird die Konfidenz über ihre geschätzte Position. Die Variable u moduliert die Grundunsicherheit darüber, wie stark sich das Validierungsfenster mit der Zeit vergrößert. Empirisch hat sich $u = 10$ als zuverlässigste Wahl ergeben, mit der das Validierungsfenster gerade groß genug ist, um Zielassoziationslandmarken zu finden, aber klein genug, um falsche Assoziationen zu unterdrücken. Mit dem Validierungsfenster r ist nun der Radius definiert, innerhalb dessen bekannte Landmarken mit der gleichen Klasse als Assoziationskandidaten infrage kommen. Ist in diesem Bereich keine Landmarke vorhanden, wird der Landmarkenkandidat als neue Landmarke in die semantische Karte mit aufgenommen. Liegt innerhalb des Validierungsfensters eine einzelne passende bestehende Landmarke vor, wird diese automatisch mit dem Landmarkenkandidaten assoziiert. Als dritte Möglichkeit werden innerhalb des Validierungsfensters mehrere Assoziierungskandidaten ermittelt. In diesem Fall wird ein feinerer Abgleich durchgeführt, um den korrekten Assoziierungskandidaten zu ermitteln.

2. **Feinassoziierung mittels Nearest Neighbor Verfahren**

In diesem Fall wird die Annahme getroffen, dass der Vergleich der Abstände zwischen den Mittelpunkten der Landmarken nicht mehr zuverlässige Resultate liefert. Es kann davon ausgegangen werden, dass die Punktwolke, die das Objekt repräsentiert, aufgrund der Aufnahme aus nur einer Perspektive nicht vollständig ist. Der prädizierte Mittelpunkt kann somit nur als Näherung an den tatsächlichen Mittelpunkt betrachtet werden. Statt des Abgleichs der Mittelpunkte wird deshalb ein Abgleich der Punktwolken vorgeschlagen. Hierfür wird der euklidische Abstand zwischen den Punktwolken des Kandidaten c und der Landmarke l innerhalb des Tores r mithilfe der Nearest Neighbor Methode berechnet.

$$l_t(c) = \arg\min_{l \in L} \sum_{i=1}^{P_c} d(p^i, p_l^{NN}) \cdot \frac{1}{P_c} \tag{5.8}$$

Ziel ist es, für jeden Punkt p in der Kandidaten-Punktwolke P_c seinen nächsten Nachbarpunkt p_l^{NN} aus der Landmarken-Punktwolke P_l zu finden und somit aus der Menge der Landmarken L diejenige Landmarke l zu finden, die im Mittel den kürzesten Abstand d zum nächsten Nachbarn hat. Dies hat den Vorteil, dass man nicht den kürzesten Abstand zwischen den Mittelpunkten wählt, sondern die Landmarke mit der stärksten Überschneidung ermittelt.

5.2.3 Positionsoptimierung

Die Betrachtung desselben statischen Objekts aus verschiedenen Blickwinkeln eröffnet die Möglichkeit, Rückschlüsse auf die Kamera-Trajektorie zu ziehen. Um dies zu nutzen, werden die Beobachtungen von bereits kartierten Landmarken dem im SLAM zugrunde liegenden Pose-Graphen zur Verfügung gestellt, um für die Prädiktion der Roboterpose einbezogen zu werden. Hierfür wird jeder Landmarke eine eindeutige ID zugewiesen. Wenn eine bereits kartierte Landmarke erneut gesichtet wird, wird ihre Position auf der Bildebene zusammen mit der Landmarken-ID an den Lokalisierungsalgorithmus übergeben, wo sie als zusätzliche Randbedingung im Pose-Graphen zur Optimierung der Trajektorieschätzung eingesetzt wird (siehe Bundle-Adjustment in Abschnitt 3.1.3).

5.2.4 Landmarkenverfeinerung

Bei der Optimierung der Trajektorie durch die Sichtung bekannter semantischer Landmarken (oder anderer Optimierungsansätze für Loop-Closure) werden die kartierten Landmarkenpositionen entsprechend dem korrigierten Pose-Graphen aktualisiert. Dies erlaubt es, frühere Assoziationsfehler zu korrigieren. Falls die geschätzte Robotertrajektorie von der tatsächlichen abweicht, kann es sein, dass ein neuer Landmarkenkandidat seine Assoziationslandmarke innerhalb des Validierungsfensters nicht findet und diese als vermeintlich neue Landmarke registriert wird. In diesem Fall werden am Ende zwei Landmarken mit unterschiedlichen IDs von demselben realen Objekt erzeugt. Nach der Loop-Closure-Verarbeitung der Trajektorie werden die Landmarken nachträglich korrigiert und auf die fehlende Assoziationslandmarke von vorher gesetzt. Auf diese Weise können übermäßig überlappende Landmarken der gleichen Klasse erkannt und zu einer einzigen Landmarke verschmolzen werden. Auf diese Weise können auch große Objekte abgebildet werden, indem schrittweise mehrere semantische Landmarken zu einer Einzigen zusammengeführt werden.

5.3 Experimente

Im Folgenden wird die in den vorherigen Abschnitten vorgeschlagene Methode zur semantischen Kartierung experimentell evaluiert. Zuerst wird hierfür die Implementierung der Methode in ein SLAM-Framework beschrieben (Abschnitt 5.3.1). Im Anschluss werden auf Basis dieser Implementierung Testdurchläufe auf den öffentlichen SLAM-Datenbanken TUM RGB-D [81] und Stereo KITTI [82] zur quantitativen Evaluierung (Abschnitt 5.3.2) durchgeführt. Es folgt eine exemplarische Evaluierung der qualitativen Resultate (Abschnitt 5.3.3), gefolgt von einer Datenerhebung zur Laufzeitanalyse (Abschnitt 5.3.4) in Anbetracht der angestrebten Echtzeitfähigkeit der Anwendung.

5.3.1 Implementierung

Die vorgeschlagene Methodik zur semantischen Kartierung wurde als Erweiterungsmodul für konventionelle SLAM-Implementierungen konzipiert. Dazu wurde es als kompatibles Softwarepaket zum Robot Operating System (ROS) [158] entwickelt, eine weitverbreitete Middleware zur Robotersteuerung. Als Beispiel wurde die SLAM-Implementierung RTAB-Map [159] als SLAM-Framework ausgewählt, welche einen hohen Grad an Modularität bietet und den Einsatz unterschiedlicher Arten

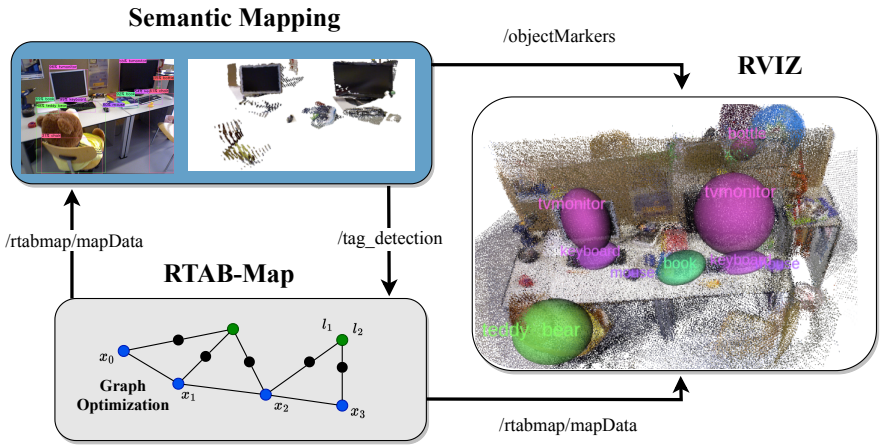


Abbildung 5.6: Übersicht der Implementierung des vorgeschlagenen Ansatzes zur semantischen Kartierung. Die Methode wurde als separates Modul entwickelt, das über ROS [158] mit dem RTAB-Map-SLAM-Framework kommuniziert. Über den */rtabmap/mapData*-Kommunikationkanal werden Informationen über den Pose-Graph angefordert, während über */tag_detection* Informationen zu semantischen Objekten der Graph-Optimierung bereitgestellt werden. Die generierten Landmarken werden der Punktwolkenkarte im RVIZ-Visualisierungstool hinzugefügt.

von Lokalisierungs- und Graph-Optimierungsalgorithmen ermöglicht. Abbildung 5.6 zeigt eine Übersicht darüber, wie das vorgeschlagene Semantic Mapping Modul mit dem RTAB-Map Algorithmus interagiert, um die Umgebungskarte mit semantischen Objekten zu erweitern. Für die Projektion und Assoziierung von Landmarken (siehe Abschnitt 5.2) wird die letzte Aktualisierung des Posen-Graphen von RTAB-Map abgerufen. Semantische Landmarken sind als AprilTag [160]-Nachrichten konzipiert, welche nativ von RTAB-Map unterstützt werden. Zur Visualisierung werden die semantischen Landmarken als kugelförmige Marker in die Punktwolkenkarte in RVIZ eingesetzt. Die Größe der Kugel wird auf der Grundlage der Objektabmessungen der Punktwolke geschätzt.

Alle Experimente wurden auf einem Ubuntu 16.04 Linux-System mit einer AMD Ryzen 3950x CPU bei 3,5 GHz und 64 GB RAM und einer NVIDIA RTX 2080 TI durchgeführt.

Objektdetektion Für die Objektdetektion innerhalb des vorgeschlagenen Moduls wurde das YOLOv4 [149] integriert. Es wurde auf dem COCO-Datensatz [155] trainiert und ist in der Lage, zwischen 80 verschiedenen Klassen zu detektieren. Eine Liste weiterer Systemeinstellungen, die für die weiteren Experimente genutzt werden, findet sich in Tabelle 5.1. Ihre korrekte Konfiguration hängt von der Szene und der Kamerabewegung ab: Hektische Kameratrajektorien erfordern beispielsweise weniger strikte IOU-Tracker-Regeln, um sicherzustellen, dass neue Detektionen mit bestehenden Tracklets in Verbindung gebracht werden können.

Graphen-Optimierung Als Graph-Optimierer wurde GTSAM [161] verwendet, um die semantischen Landmarken in den Pose-Graphen mit einzubinden. Zusätzlich wurde minimaler zeitlicher Abstand zwischen zwei Landmarken-Erkennungen für dasselbe Objekt von 2 Sekunden festgelegt.

Systemeinstellungen		
Objektdetektion	CNN Konfidenzschwelle	0,4
	Minimaldistanz	0,2 m
	Maximaldistanz	25 m
IOU Tracker	σ_{IOU}	0,2
	Minimale Tracklet-Länge	5
	Maximale zeitl. Distanz [1]	0,5
Landmarkenassoziation	u	10

[1] Maximaler zeitlicher Abstand zwischen zwei Detektionen (Sekunden).

Tabelle 5.1: Systemkonfiguration für alle durchgeführten Experimente.

Evaluationsmetriken Als Fehlermaß wurde der absolute mittlere quadratische Trajektorienfehler (RMSE) gewählt, welcher bereits im vorherigen Kapitel in Abschnitt 4.3 vorgestellt wurde.

5.3.2 Quantitative Evaluation

Zur quantitativen Evaluation der vorgeschlagenen Methoden werden die öffentlichen Datensätze RGB-D TUM und KITTI eingesetzt. Beide Datensätze stellen die Grundwahrheiten über die Kameratrajektorien bereit, allerdings nicht die über die Positionen der Objekte in den Szenarien. Daher fokussiert sich die quantitative Evaluation auf die Analyse des Einflusses der semantischen Objekte auf die Trajektorienoptimierung.

Trajektorienevaluation auf RGB-D TUM Es wurden insgesamt neun Sequenzen aus den Kategorien *Handheld SLAM* und *Dynamische Objekte* der RGB-D TUM Datenbank ausgewählt, die sich zur Evaluation von SLAM-Methoden bewährt haben (weitere Informationen in Abschnitt 3.3). Drei Szenarios davon (*fr1_desk, fr2_desk, fr3_long_office_household*) wurden mittels einer handgeführten Kamera aufgenommen und zeigen eine Büroumgebung mit unterschiedlichen, auf Schreibtischen platzierten Objekten. In den anderen Sequenzen interagieren zwei Personen an einem Schreibtisch. Die Kamera führt verschiedene Arten von Rotations- und Translationsbewegungen aus, während die Personen als dynamische Objekte im Bildbereich fungieren. Die Kombination aus abwechselnder Eigenbewegung der Kamera und dynamischer Umgebung macht diese Sequenzen besonders geeignet, um die Robustheit und Präzision von Methoden zur Trajektorienschätzung zu evaluieren.

Tabelle 5.2 zeigt die Ergebnisse von Lokalisierungsfehlern für die in Abschnitt 5.2 vorgestellte und in Abschnitt 5.3.1 implementierte Methode zur semantischen Kartierung. Für jedes Experiment wurde der ORB-SLAM2 [162] als Baseline verwendet. Anschließend wurde die zusätzliche semantische Kartierungseinheit einbezogen.

Es zeigt sich, dass der vorgestellte Ansatz zur semantischen Kartierung in der Lage ist, die Trajektorienschätzung insbesondere in Szenen zu verbessern, in denen eine hochdynamische Umgebung die Genauigkeit des Baseline-Models stark beeinträchtigt. Während jenes Modell alle extrahierten Merkmale verwendet, einschließlich derer, die auf Bildebene auf dynamischen Objekten liegen,

Sequenz	ORB-SLAM2 Standard	ORB-SLAM2 + ***Semantic Mapping***	Differenz
fr1_desk	0,052	0,042	19,2%
fr2_desk	0,072	0,047	34,5%
desk_with_person	0,081	0,062	23,5%
sitting_xyz	0,012	0,014	-16,7%
walking_static	0,042	0,047	-11,9%
walking_hs	0,259	0,171	34,0%
walking_rpy	0,457	0,377	17,5%
walking_xyz	0,387	0,124	68,0%

Tabelle 5.2: RMSE des absoluten Trajektorienfehlers [m] für unterschiedliche Test-Sequenzen aus dem RGB-D TUM Datensatz.

verhindert der vorgeschlagene Tracklet-Ansatz und die zusätzliche Validierung nach dynamischem Verhalten deren Einbeziehung als semantische Landmarken. Infolgedessen wurden die Landmarken für die Optimierung des Pose-Graphen nur aus den statischen Objekten auf dem Schreibtisch im Hintergrund erstellt, was zu einer Korrektur der verfälschten Posenschätzung aus dem ORB-SLAM2 führt. Insbesondere in der Sequenz *walking_xyz* demonstriert die semantische Kartierung, dass die Verfolgung der Gegenstände auf dem Schreibtisch und deren Verwendung für die Pose-Optimierung dazu beiträgt, den durch die dynamischen Objekte verursachten Trajektoriendrift zu korrigieren.

In Szenen mit einer statischen Umgebung konnte der Trajektoriefehler durch die semantische Kartierung nicht verbessert werden. Ein Grund dafür ist, dass die Trajektorienschätzung der Baseline bereits sehr genau ist. Ein weiterer Grund ist die ungenaue Prädiktion der Objektzentren (Abschnitt 5.2.2). Der ermittelte Mittelpunkt stellt immer nur eine Annäherung an das tatsächliche Objektzentrum dar, da er nur auf Einzelbetrachtungen des gesamten Objekts beruht. Werden diese Annäherungen der Pose-Graphen als Randbedingung übergeben, kann in der Optimierung die Trajektorie negativ beeinflusst werden. Dieser negative Effekt variiert mit der Größe des Objekts und geht mit der Differenz zwischen dem geschätzten und dem tatsächlichen Zentrum der Landmarke einher. Das macht große Objekte wie Tische weniger nützlich für die Einbeziehung in den Lokalisierungsprozess. In Sequenzen mit allgemein niedrigem Trajektoriefehler können selbst kleine Ungenauigkeiten bei der Objektmittelpunktschätzung zu einem Präzisionsverlust führen. Die zeigt sich insbesondere in der Sequenz *fr3_long_office_household*.

Trajektorienevaluation auf KITTI Für die Evaluation auf dem KITTI-Datensatz wurden anlehnend an vorherige Arbeiten [140] vier Szenen (00, 05, 06, 07) ausgewählt, in denen die Trajektorie Loops durchläuft. Dies eröffnet die Möglichkeit zu untersuchen, ob das Verfahren zur semantischen Kartierung in der Lage ist, zuvor kartierte Objekte wiederzuerkennen (Loop Closure, siehe Abschnitt 3.1.4), und die Genauigkeit der Trajektorie zu verbessern. Die Ergebnisse werden mit zwei weiteren Ansätzen zur semantischen Kartierung von Doherty *et al*. [144] und Bernreiter *et al*. [140] aus dem Stand der Technik verglichen. Zusätzlich werden mehrere Testdurchläufe mit

Sequenz	KITTI 00	KITTI 00	KITTI 06	KITTI 07
Doherity *et al*. [144]	-	5,718	-	-
Bernerheiter *et al*. [140]	4,54	4,4	**2,3**	2,9
ORB-SLAM2	4,841	3,263	3,554	1,929
ORB-SLAM2 + LC	**4.228**	3,245	3,428	1,801
ORB-SLAM2 + SM	4,495	3,202	3,438	1,808
ORB-SLAM2 + SM + LC	4,389	**3,162**	3,043	**1,775**

Tabelle 5.3: RMSE des absoluten Trajektorienfehlers [m] für KITTI-Sequenzen.

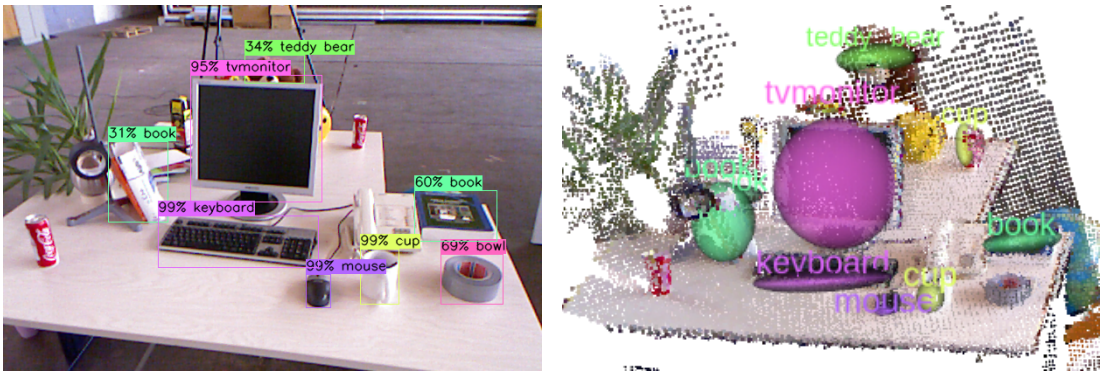
unterschiedlichen Kombinationen aus Loop-Closure-Detektion (*LC*) durchgeführt, um den Einfluss der einzelnen Methoden auf die Gesamtleistung zu ermitteln.

In der Ergebnistabelle 5.3 weist der Stereo ORB-SLAM2 ohne Loop-Closure-Detektion oder der Einbeziehung semantischer Landmarken den höchsten Trajektorienfehler auf. Der Grund liegt im Drift, der sich kontinuierlich mit der Zeit verstärkt und im Laufe der Sequenz nicht durch den Loop-Closure-Algorithmus korrigiert wird. Bei Testdurchläufen mit Loop-Closure oder semantischer Kartierung kann über alle Sequenzen hinweg eine Reduzierung des RMSE erzielt werden, wobei der Loop-Closure-Algorithmus in den meisten Sequenzen $(00, 06, 07)$ etwas besser abschneidet als das semantische Mapping. Für Sequenz 06 entspricht die Fehlerdifferenz 0,1%, während für Sequenz 07 der Loop Closure mit 0,3% bessere Ergebnisse erzielt. Die Leistungsunterschiede sind dementsprechend marginal. Für die Sequenz 00 hingegen erzielt der Loop Closure einen fast 20% geringeren Trajektoriefehler gegenüber der semantischen Kartierung. Der Grund dafür ist, dass in dieser Sequenz der Loop-Closure aus der gleichen Perspektive erkannt wird und somit ideale Voraussetzungen für 2D-Merkmal-basierte Loop-Closure-Erkennungsmethoden bietet. Die Kombination aus Loop-Closure-Detektion und semantischer Kartierung führt bis auf einen Fall zu den besten Ergebnissen und übertrifft damit den Stand der Technik von Bernreiter *et al*. [140] und Doherty *et al*. [144].

In Sequenz 00 hingegen erzielt die Loop-Closure-Detektion den geringsten Trajektoriefehler und schlägt damit den Stand der Technik und die Kombination mit der semantischen Kartierung. Bei der Analyse der Sequenzverarbeitung zeigt sich, dass ihre erhöhte Fehlerrate, ähnlich wie in Abschnitt 5.3.2, durch eine semantische Landmarke hervorgerufen wird, die im Laufe der Sequenz erfolgreich wiedererkannt wird, deren Mittelpunkt aufgrund von wechselnder Perspektive jedoch von der vorherigen Schätzung abweicht. Infolgedessen führt der Abstand zwischen den beiden geschätzten Mittelpunkten zu einer Beeinträchtigung der Trajektorienoptimierung.

5.3.3 Qualitative Evaluation

In der vorangegangenen quantitativen Evaluation der vorgeschlagenen Methode wurde ihr Einfluss auf die Trajektorienschätzung untersucht. Die Performanz der Erfassung und Lokalisierung von semantischen Objekten in den untersuchten Szenen wurde dabei aufgrund fehlender Grundwahrheiten nicht quantitativ ausgewertet. Stattdessen werden im Folgenden einige qualitative Ergebnisse der vorgeschlagenen Methode anhand von Sequenzen aus den vorgestellten Datensätzen vorgestellt.



(a) Bild aus einer Testsequenz mit Objektdetektionen und entsprechenden 2D-Bounding-Boxen.

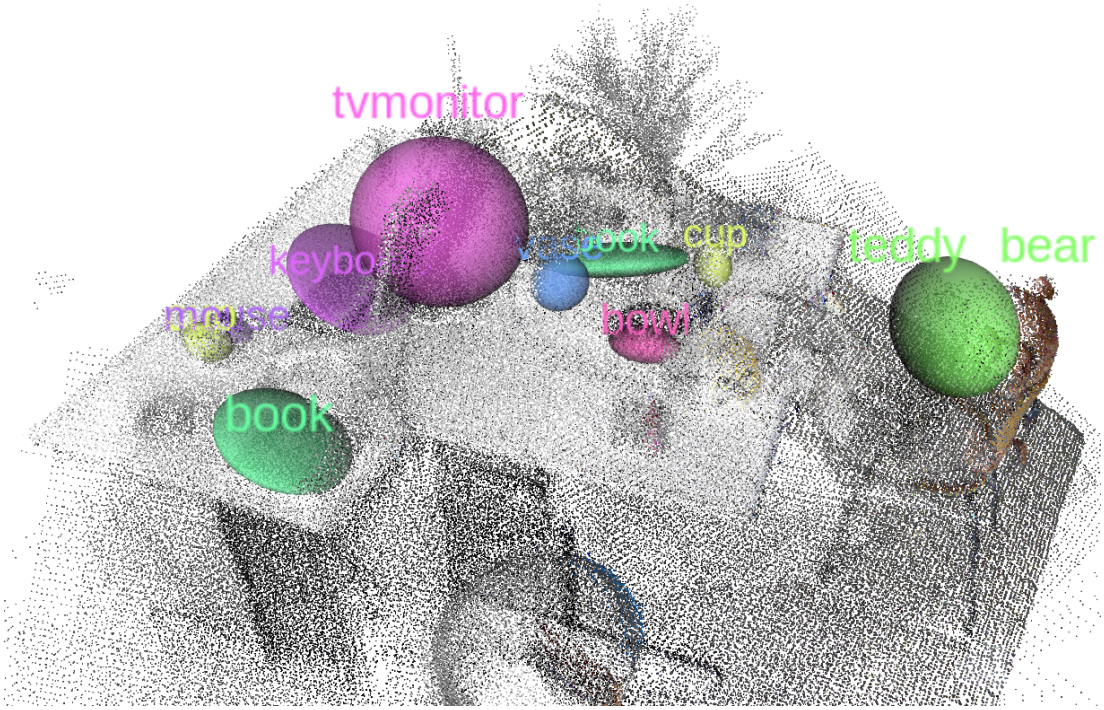
(b) 3D-Kartenwolke mit entsprechenden Objektlandmarken.

Abbildung 5.7: Beispielbild einer Punktwolkenkarte, die mit aus 2D-Erkennungen abgeleiteten 3D-Objektmarkern ergänzt wurde. Die Marker sind als beliebig gefärbte Kugeln abgebildet.

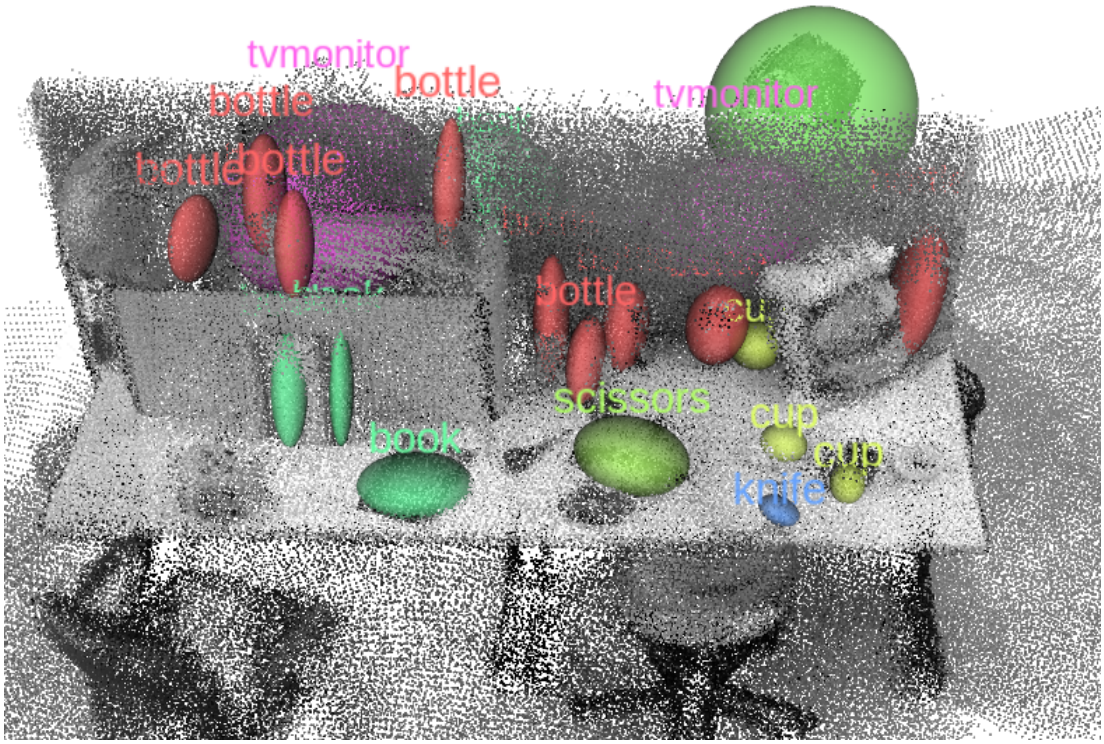
Semantische Kartierung auf RGB-D TUM Abbildung 5.7 zeigt einen exemplarischen Vergleich zwischen dem Eingabebild mit Objektdetektionen in Abbildung 5.7a und der entsprechenden Punktwolkenkarte mit angereicherten semantischen Objekten in Abbildung 5.7b. Mehrere Objekte werden erfolgreich erkannt und als Kugel bzw. Ovaloid in die Karte projiziert. Falsch positive Vorhersagen (z. B. SSchale"für das Klebeband) werden im Prozessschritt zur Generierung von Landmarkenvorschlägen zurückgewiesen. Zudem konnte zusätzlich, basierend auf der Bounding-Box der Objektdetektion und der extrahierten Punktwolke, die ungefähre Größe der Objekte geschätzt werden, um sich grob ihrer allgemeinen Erscheinungsbild zu nähern. Dank des Schrittes zur Verfeinerung der Landmarken (siehe Abschnitt 5.2.4) konnte zudem die Schätzung der Objektgröße mit wachsendem Informationsgehalt im Verlauf der Sequenz mit der tatsächlichen Größe konvergieren. Ein Beispiel hierfür ist die Objektkugel des Teddybären, die nur einen Teil des Gesichts erfasst. Der Grund dafür ist, dass der Landmarkenvorschlag, der zu dieser Objektmarkierung führte, aus Bildern zum Beginn der Sequenz generiert wurde, in denen der Tisch frontal gefilmt und nur ein kleiner Teil des Gesichts des Teddybären zu sehen war (siehe Abbildung 5.7a). Mit der Einführung weiterer Landmarkenvorschläge aus anderen Blickwinkeln vergrößert sich die Kugelgröße mit jeder Landmarkenfusion und bedeckt schließlich den Teddybären vollständig.

Dies wird auch in Abbildung 5.8 verdeutlicht. Hier werden drei weitere Punktwolken aus RGB-D-TUM-Sequenzen illustriert, die mit semantischen Markern aus den generierten Landmarken erweitert wurden, wobei Abbildung 5.8a auf derselben Sequenz wie Abbildung 5.7 basiert. Es zeigt sich, wie sich der semantische Marker des Teddybär-Objekts bereits vergrößert hat, da mehr Kandidatenvorschläge in ihn verschmolzen wurden. In Abbildung 5.8b wird ersichtlich, dass die vorgeschlagene Methode dank des adaptiven Validierungsfensters auch in der Lage ist, zwischen kleinen Gegenständen der gleichen Klasse zu unterscheiden, die durch *bottles* und *cups* repräsentiert werden.

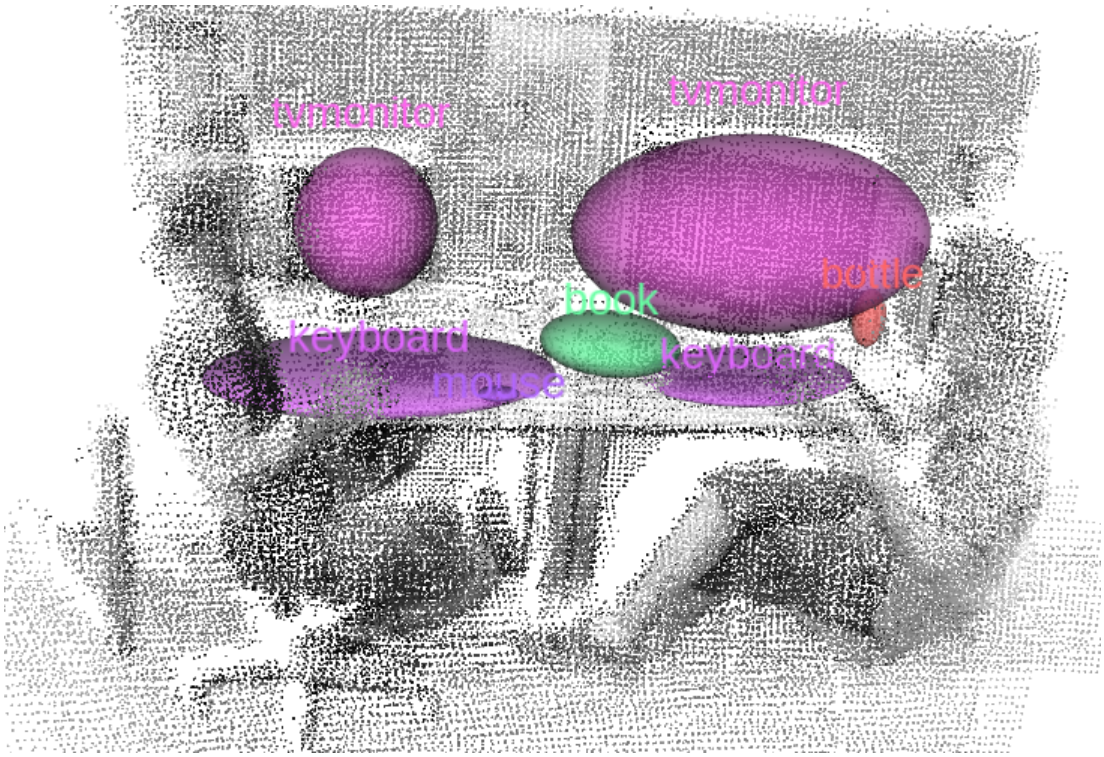
Auch in diesem Szenario ist eine Teddybär-Instanz vorhanden. Die Aufnahme wurde am Ende der Sequenz festgehalten, als sich die Kamera bereits um den gesamten Arbeitsplatz bewegt hatte. Daher umspannt der semantische Marker das Teddybär-Objekt vollständig.



(a) Punktwolke mit semantischen Objekten aus der Sequenz *fr2_desk*.



(b) Punktwolke mit semantischen Objekten aus der Sequenz *fr3_long_office_household*.

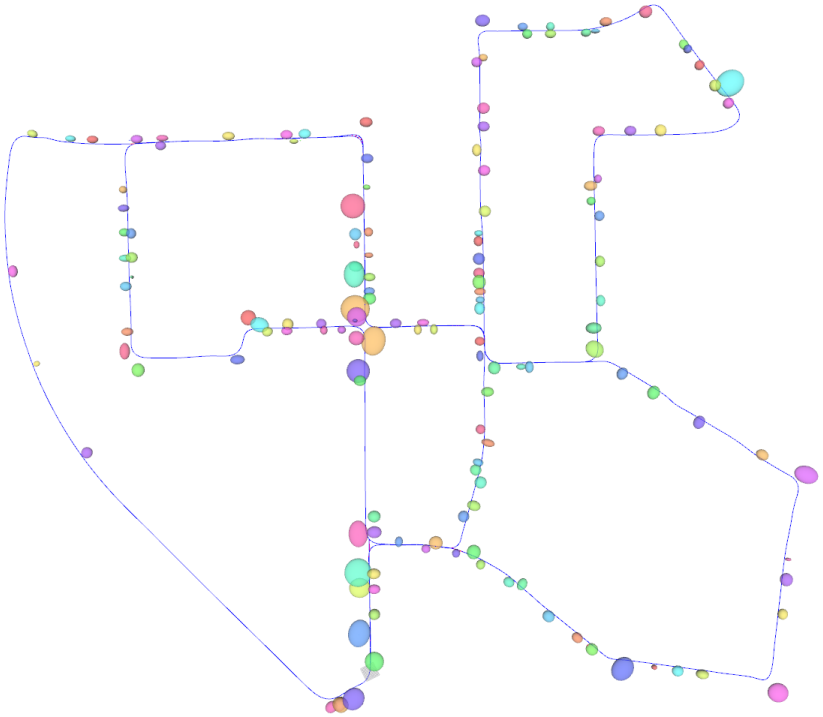


(c) Punktwolke mit semantischen Objekten aus der Sequenz *fr3_sitting_xyz*.

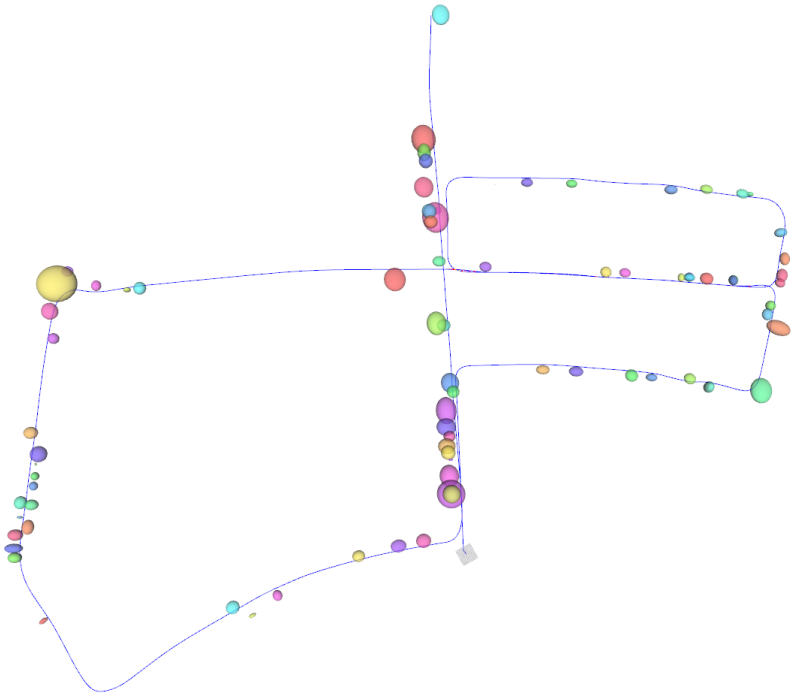
Abbildung 5.8: Qualitative Ergebnisbilder der semantischen Kartierung auf Basis von Sequenzen des RGB-D-TUM-Datensatzes.

5.3.3.1 Semantische Kartierung auf KITTI Während der RGB-D TUM Datensatz im Innenraum aufgenommen wurde und nur eine limitierte Anzahl an Objekten umfasst, bietet der KITTI Datensatz die Möglichkeit, die Kartierungsperformanz im Außenbereich mit einer größeren Anzahl an semantischen Objekten zu evaluieren. Abbildung 5.9 zeigt zwei beispielhafte Trajektorienkarten aus dem KITTI-Datensatz aus der Top-Down-Perspektive für die Sequenzen 00 und 05. Sie zeigen auf, dass die vorgeschlagene Methode in der Lage ist, die semantische Kartierung auch in größerem Maßstab durchzuführen. Zur besseren Sichtbarkeit wurde dabei die Punktwolke entfernt und die Objekt-Kugel um den Faktor 4 vergrößert. Die Farbwahl der Objekt-Kugel ist dabei zufällig gewählt und nicht klassenabhängig, um die Übersichtlichkeit zu gewährleisten.

Die erste Karte aus Abbildung 5.9a wurde aus der Sequenz *KITTI 00* erzeugt. Insgesamt wurden 153 semantische Objekte erfolgreich lokalisiert und der Karte im Verlauf der Szene hinzugefügt. Von diesen 153 Objekten werden 150 als Auto, zwei als Lkw und eins als Motorrad klassifiziert. In ähnlicher Weise wurden in Abbildung 5.9b 80 semantische Objekte aus der Sequenz *KITTI 05* extrahiert, die in 72 Autos und acht Lastwagen unterteilt sind. Abbildung 5.10 zeigt ein Beispielbild aus der Kameraperspektive in der Mitte der Sequenz, in der zwei Autos als semantische Landmarken hinzugefügt wurden. Die Größe der Kugel zeigt, dass unsere Methode in der Lage ist, die Positionen und Größen der Objekte auch im größeren Maßstab im Vergleich zum RGB-D TUM Datensatz genau zu schätzen.



(a) Erzeugte semantische Karte aus der Sequenz *KITTI 00*.



(b) Erzeugte semantische Karte aus der Sequenz *KITTI 05*.

Abbildung 5.9: Semantische Kartierungsergebnisse für Sequenzen aus dem KITTI Datensatz aus der Top-Down Sicht.

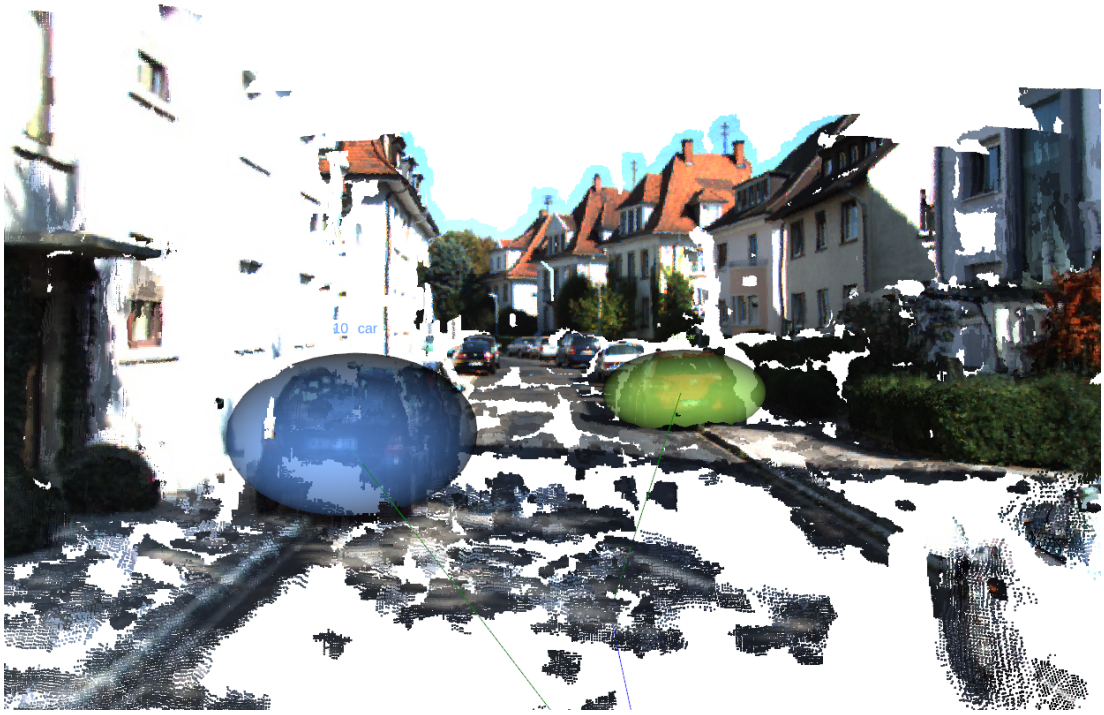


Abbildung 5.10: Generierung von Objektkugeln mit Größenanpassung aus der Sequenz KITTI 00 aus der Kameraperspektive.

5.3.4 Laufzeitanalyse

Für den anwendungsorientierten Einsatz zusammen mit anderen Echtzeit-SLAM-Implementierungen muss unsere Methode ebenso recheneffizient sein. Tabelle 5.4 zeigt die durchschnittliche Verarbeitungszeit für jeden Schritt, gemessen an der *fr2_desk*-Sequenz. Ein durchschnittlicher Zyklus des Hauptthreads dauert 65 ms. In einigen Fällen, in denen die Assoziation mithilfe der Validierung nicht zu einer eindeutigen Entscheidung führt, wird die zweite Assoziationsstrategie (*Kandidatenassoziation 2*) in Form einer punktwolkenbasierten Nearest-Neighbour-Methode durchgeführt. In diesem Fall werden etwa 150 ms zusätzlich für die Assoziationsentscheidung benötigt. Die Objekterkennung und der Schritt des Kandidatenvorschlags werden in einem separaten Thread verarbeitet. Dadurch wird sichergestellt, dass das System in effizienter Weise gültige Objektvorschläge für aufeinanderfolgende Eingangsbilder finden kann, ohne durch den zeitaufwendigen Assoziationsschritt verlangsamt zu werden. Im Allgemeinen wird die Verarbeitungszeit von mehreren Faktoren stark beeinflusst. Kleine Objekte, die durch eine kleinere Punktwolke dargestellt werden, werden schneller verarbeitet als größere Objekte. Je mehr potenzielle Assoziationskandidaten sich innerhalb des Validierungsbereichs eines neuen Vorschlags befinden, desto länger dauert die Entscheidung. In den durchgeführten Experimenten lief das System nahtlos neben RTAB-Map und war in der Lage, mehrere Erkennungen für jede Trajektorieschätzung zu liefern.

Prozess	Dauer (ms)
Objektdetektion	22,62
Generierung von Objektkandidaten	0,069
Landmarkenassoziierung 1 (/ + 2)	7,8 (/ + 150)
Landmarkenupdate & -verschmelzung	23,16

Tabelle 5.4: Durchschnittliche Laufzeiten für jede Prozessstufe über eine gesamte Iteration auf Basis *fr2_desk*-Sequenz. Die Landmarkenassoziierung 2 wird nur ausgeführt, sofern Landmarkenssoziierung 1 zu keinem klaren Ergebnis führt.

5.4 Implementierung am Roboter

Der vorgeschlagene Ansatz zur semantischen Kartierung hat in mehreren Experimenten aufgezeigt, dass er in der Lage ist, Objekte zu detektieren, zu kartieren, und dies in einer Weise, die sogar für die Optimierung der Lokalisierung eingesetzt werden kann.

Im Folgenden wird der vorgeschlagene Implementierungsansatz aus Abschnitt 5.3.1 in einen mobilen Roboter integriert. Um eine Echtzeitfähigkeit zu gewährleisten, wird das Modul auf einer mobilen Grafikkarte (NVIDIA Jetson AGX Orin) eingesetzt, deren Hardware zur Verarbeitung von neuronalen Netzen ausgelegt ist.

Abbildung 5.11 zeigt eine Szene, in der der Roboter sich vor einem Schreibtisch befindet. Der linke Ausschnitt zeigt eine Aufnahme der Szene als RGB-Bild. Auf dem Tisch befinden sich unter anderem ein PC, ein Monitor und verschiedene Dosen. Der rechte Ausschnitt zeigt einen Teil der Umgebungskarte des Roboters aus ähnlicher Perspektive. Neben der erzeugten Punktwolke sind auch semantische Landmarken für den Monitor und die Dosen (als *cups*) integriert. Die Erkennung und Kartierung der Objekte erfolgt während der Anfahrt in Richtung des Tisches, sodass die exakte Positionsbestimmung und die Datenassoziation besonders herausfordernd sind. Dennoch schafft es der vorgeschlagene semantische Kartierungsansatz, die Zentren der Dosen trotz ihres geringen Ausmaßes zuverlässig zu bestimmen. Dies bietet eine gute Grundlage für die Generierung nachfolgender Handlungen (z. B. Greifoperationen). Da die exakte Form von Objekten jedoch nicht im Detail erfasst wird, sind für die Berechnung von Greifoperationen ggf. weitere Verarbeitungsschritte notwendig.

In Abbildung 5.12 ist ein Ausschnitt aus der gleichen Umgebungskarte dargestellt. Diesmal wird ein großer Fernseher durch die semantische Kartierung erfasst und in die Karte registriert. Dieses Fallbeispiel zeigt, dass auch größere Objekte im gleichen Setup durch die vorgeschlagene Methode lokalisiert und in ihrer Größe bestimmt werden können.

5.5 Limitationen

Das in diesem Kapitel vorgestellte Verfahren zur semantischen Kartierung verfolgt einen deterministischen, *greedy* Ansatz zur Datenassoziation, der die Echtzeitanwendung (siehe Abschnitt 5.3.4) und eine Vielzahl von Anwendungsmöglichkeiten (z. B. über die AprilTag-Schnittstelle) ermöglicht. Diesen Vorteilen steht der Nachteil gegenüber, dass mögliche Fehler in der Datenassoziation

Abbildung 5.11: Implementierung der semantischen Kartierung in den TIAGo Roboter. Links: Reales Bild der Szene. Rechts: Kartierung durch den Roboter inklusive semantischer Objekte (*tvmonitor*, *cup*).

die Präzision der Odometrieschätzung stark negativ beeinflussen können. Eine weitere Schwierigkeit sind Objekte der gleichen Klasse, die sehr nahe beieinander liegen. In diesem Fall muss das Validierungsfenster kleiner sein als der Abstand zwischen ihnen, damit sie nicht als ein einziges Objekt verschmolzen werden. Eine mögliche zukünftige Lösung für diese Einschränkungen könnte die Anwendung probabilistischer Ansätze sein, um nur sehr sichere Assoziationen zu akzeptieren und potenzielle Kandidaten zurückzuhalten, bis ihre Zuordnung durch nachfolgende Erkennungen bestätigt wird.

5.6 Diskussion

In diesem Kapitel wurde eine Methode zur echtzeitfähigen semantischen Kartierung vorgestellt, welche in der Lage ist, herkömmliche SLAM-Methoden mit der Erfassung von semantischen Umgebungsobjekten auf robuste und effiziente Weise zu erweitern. Die Einbeziehung semantischer Objekte ermöglicht, das perzeptive Verständnis über die Umgebung zu erhöhen und die Fähigkeiten des mobilen Systems für Interaktionen zu erweitern. Zusätzlich können durch die Objektassoziationen Rückschlüsse auf die Trajektorie gezogen werden, durch die in quantitativen Experimenten eine Verbesserung des Trajektoriefehlers nachgewiesen werden konnte.

Eine elementare Herausforderung der semantischen Kartierung besteht in der präzisen Lokalisierung der Objekte auf Basis der 2D-Detektionen. Der vorgeschlagene probabilistische Ansatz bietet dabei eine robuste Methode zur Annäherung, die sich in den Szenarien aus den Datensätzen und in Testfahrten auf einem mobilen Roboter bewiesen hat. In den durchgeführten Experimenten auf den

Abbildung 5.12: Semantische Kartierung eines Fernsehers.

Testdatensätzen wurde auch festgestellt, dass Messungen aus unterschiedlichen Perspektiven einen erheblichen Einfluss auf den prädizierten Mittelpunkt des Objektes haben und die Trajektorieoptimierung negativ beeinflussen können. Die Verwendung von Detektoren zur 3D-Objektdetektion könnte die Variabilität der Objektlokalisierung verringern und damit auch der Datenassoziation zugutekommen. Derzeit sind die Datensätze mit 3D-Objekten-Annotation jedoch noch stark limitiert, sodass entsprechende Detektoren bisher nicht zur Verfügung stehen.

Durch die semantische Kartierung werden für die Interaktion zwischen Mensch und Roboter neue Möglichkeiten des Austausches geschaffen. So können dem Roboter weitere Aufgaben übertragen werden, die das Suchen von Objekten, das Tragen und Überreichen von Objekten umfassen, sowie weitergehende Aufgabenkombinationen, die aus diesen Fähigkeiten resultieren. Um Objekte präzise greifen zu können, ist es jedoch notwendig, deren genaue Maße und Formen zu erfassen, idealerweise auch die Materialbeschaffenheit. Diese Faktoren werden im derzeitigen Ansatz weniger berücksichtigt, könnten aber in zukünftigen Entwicklungen die Fähigkeiten des Roboters verbessern und seine Interaktionskompetenzen nachhaltig erweitern.

Teil II

Personenanalyse

KAPITEL 6

Robuste Kopfposeschätzung im gesamten Rotationsbereich

Methoden der bildbasierten Kopfposeschätzung verfolgen das Ziel, die Orientierung des menschlichen Kopfes anhand von Bildern zu prädizieren. Die Kopfpose ist ein essenzielles Merkmal zur Bestimmung des menschlichen Zustands und findet deshalb in einer breiten Palette von Fragestellungen Anwendung. Dazu gehört unter anderem die Aufmerksamkeitsschätzung [163, 164, 165], die Gesichtsidentifikation [166, 167] und die Schätzung von Gesichtsattributen [168, 169], welche wiederum wichtige Indikatoren für Fahrerassistenzsysteme [170, 171, 172], Augmented Reality [173, 174] und Mensch-Roboter-Interaktion [175, 176, 177] darstellen.

Aufgrund der charakteristischen Merkmale innerhalb des Gesichtsbereichs hat dessen Erfassung und Analyse bereits in der klassischen Bildverarbeitung große Popularität erfahren und robuste Verarbeitungspipelines hervorgebracht. So konnte bereits vor 20 Jahren auf Basis der geometrischen Erfassung von Mund, Nase und Augen die Kopfpose effizient abgeleitet werden [178]. Der Einzug von Deep Learning sorgte für noch robustere und zuverlässigere Prädiktionen. In beiden Fällen beschränkt sich die Erfassung der Orientierung des Kopfes jedoch auf die frontale Ansicht. Die Schätzung von stärker rotierten Kopfposen ist nicht vorgesehen. Somit sind derzeitige Methoden auf die Erfassung von maximal der Hälfte der möglichen Posen limitiert und restringieren damit die Systeme, in denen sie eingesetzt werden.

Im Rahmen dieses Kapitels wird eine neue Methode zur bildbasierten Kopfposeschätzung vorgestellt, die erstmals das effiziente Lernen und Prädizieren vom gesamten Rotationsbereich des Kopfes ermöglicht. Zunächst wird ein Überblick über den aktuellen Stand der Technik und derzeitige Limitationen bildbasierter Verfahren zur Kopfposeschätzung gegeben (Abschnitt 6.1), gefolgt von einer Erläuterung unterschiedlicher Rotationsformalismen (Abschnitt 6.2). Im Anschluss wird auf

dieser Basis ein neues Verfahren zur effizienten und robusten Kopfposeschätzung vorgestellt, das nicht auf den Frontalbereich limitiert ist, sondern uneingeschränkt jegliche Posevariationen erfassen kann (Abschnitt 6.3). Diese wird einer quantitativen und qualitativen Evaluation unterzogen (Abschnitt 6.6). Das Kapitel schließt mit einer Beschreibung von Limitationen (Abschnitt 6.7) und einer kritischen Diskussion (Abschnitt 6.8).

Forschungsbeitrag

» Es wird eine neue, effiziente Rotationsrepräsentation hergeleitet und proponiert, welche durch ihre Kontinuität und Eindeutigkeit gegenüber herkömmlichen Ansätzen geeignetere Eigenschaften für das Trainieren neuronaler Netze aufweist.

» Es wird durch eine quantitative Evaluation systematisch nachgewiesen, dass die neu eingeführte Rotationsrepräsentation in Kombination mit einer geodätischen Distanzfunktion den Stand der Technik in Genauigkeit und Robustheit übertrifft.

» Es wird ein neuer Datensatz vorgestellt, welcher die derzeitig verfügbaren Datensätze in ihrer Rotationsebene erweitert und somit die Poseprädiktion auch für extreme Rotationen ermöglicht.

6.1 Verwandte Arbeiten

Aktuelle Methoden zur bildbasierten Kopfposeprädiktion lassen sich in landmarkenbasierte und landmarkenfreie Ansätze untergliedern.

Landmarkenbasierte Ansätze Landmarkenbasierte Methoden [179, 180, 181, 182] prädizieren in einem ersten Schritt zuerst Gesichtslandmarken und ermitteln auf deren Basis anschließend die Orientierung des Kopfes. Dies erfolgt, indem sie die relativen Positionen der geschätzten Landmarken mit einem standardisierten 3D-Kopfmodell abgleichen [183, 184, 185]. Unter idealen Umständen können mit diesem Ansatz sehr genaue Schätzungen der Kopfpose erzielt werden [186, 187, 188, 189], vorausgesetzt einer exakten Bestimmung der Landmarken. Allerdings hat der Grad der Ähnlichkeit der Kopfform zum 3D-Kopfmodell Auswirkungen auf die Suche nach dem globalen Optimum beim Alignment-Prozess. Überdies befinden sich die gesuchten Landmarken ausschließlich im Gesichtsbereich, sodass Posen mit Verdeckungen und stark abgewandte Köpfe nur bedingt verarbeitet werden können [190, 191].

Landmarkenfreie Ansätze Landmarkenfreie Ansätze sind von diesen Einschränkungen nicht betroffen, da sie die Kopfpose ohne Zwischenschritte direkt aus den Bildern mittels End-to-End-Strategie herleiten. Diese Methoden verwenden üblicherweise tiefe neuronale Netze, welche die Aufgabe zur Rotationsprädiktion allein auf die Merkmale der verarbeitenden Bilder zentrieren.

Erste Ansätze der End-to-End Kopfposenprädiktion [192, 193, 194] verfolgen ein Regression-via-Classification-Schema [195]. Hierbei wurde der Suchbereich der drei eulerschen Winkel (Gier, Nick, Roll) im dreidimensionalen euklidischen Raum auf den Drehbereich von -99 Grad bis $+99$ Grad begrenzt. Weitere Ansätze [196] verfolgen eine stufenweise Regressions- und Merkmalsaggregation zur Vorhersage und nutzen statt Eulersche Winkel die drei Einheitsvektoren der Rotationsmatrix [197, 198]. Hierbei macht man sich einen zusätzlichen Orthogonalitäts-Loss zur Stabilisierung der Vorhersagen zunutze. Zur Optimierung des Trainings wird zudem die Fisher-Verteilung eingesetzt [198], um eine Modellierung der Rotationsunsicherheit zu generieren und bessere Wahrscheinlichkeitsannahmen treffen zu können. Eine andere Labeloptimierung wurde von Liu *et al*. [199] vorgeschlagen, die auf Gauß'schen Label-Verteilungen trainiert, um so einen probabilistischen Ansatz zu erzielen. Weitere Ansätze konzentrieren sich auf eine optimierte Merkmalsextraktion [200] und die Diversifizierung der Trainingsdaten [201] mittels rigoroser Augmentierungsstrategien.

Im Allgemeinen hat sich in den letzten Jahren insbesondere durch den Einzug von neuronalen Netzen die Kopfposeschätzung für den Frontalbereich kontinuierlich verbessert. Es fehlen allerdings immer noch konsequente Lösungen für die Vorhersage des gesamten Rotationsbereichs, auch über den Frontalbereich hinaus. Weitere Limitation findet sich in der gängigen Konvention, die kontinuierlichen Rotationsvariablen in *Bins* aufzuteilen, um das Problem in eine Klassifizierungsaufgabe umzuwandeln, um so einen Stabilisierungseffekt beim Training zu erzeugen [192, 193, 202, 194, 200]. Dies birgt den Nachteil, dass das Beschneiden von Winkelsegmenten in Bins zu einem Informationsverlust führt. Weiterhin wird dieser einschränkende Ansatz häufig mit einer Reduzierung des Zielraums kombiniert [192, 193, 202, 194, 200], wodurch die Möglichkeit zur Schätzung des kompletten Rotationsspektrums versagt bleibt. Einige wenige Arbeiten überwinden diese Limitation, indem sie die Rotationsmatrix als geeignetere Rotationsdarstellung verwenden [197, 198, 203]. Sie befassen sich aber weder mit effizienteren Methoden der Regression noch mit ihrem Potenzial zur Erweiterung des Prädiktionsbereichs. Folglich ist die Problemstellung zur Vorhersage des vollständigen Rotationsbereichs immer noch kaum erforscht.

6.2 Rotationsformalismen in drei Dimensionen

Die Orientierung eines starren Körpers kann im dreidimensionalen Raum durch mehrere unterschiedliche Arten mathematischer Darstellungen beschrieben werden. Die am häufigsten verwendete und weitverbreitete ist die Euler-Winkel-Darstellung, welche die Drehung um jede Achse des Koordinatensystems beschreibt (typischerweise als Roll-Nick-Gier, engl: $yaw, pitch, roll$ bezeichnet). Trotz ihrer Popularität und Intuitivität haben Euler-Winkel Einschränkungen, wenn es um den spezifischen Orientierungszustand geht, bei dem die zweite elementare Drehung $\pm$ 90° erreicht. In dieser Konstellation sind yaw und $pitch$ in der gleichen Ebene ausgerichtet und erzeugen unendlich viele Lösungen für den gleichen Rotationszustand. Dieses Verhalten wird als $gimbal\ lock$ bezeichnet, da die erste und dritte Achse unter dieser speziellen Bedingung "blockiert" sind. Der Gimbal-Lock stellt hierbei den Extremfall der Einschränkungen von Euler-Winkeln dar. Die Abhängigkeit zwischen dem ersten und dem dritten Winkel ist jedoch eine grundlegende Eigenschaft von Euler-Winkeln, die umso stärker wird, je weiter sich der Nickwinkel dem Gimbal Lock Zustand nähert. Als Konsequenz

Repräsentationsform	Parametrisierung	
Eulersche Winkel $(roll, pitch, yaw)$	$(87, 73, 89, 32, -87, 93)$	$(-92, 51, -98, 02, 81, 73)$
Quaternions (q_0, q_1, q_2, q_3)	$(0,707, -0,023, -0,707, 0,031)$	$(-0,707, -0,029, 0,706, 0,041)$
Achsenwinkel $(\tilde{x}, \tilde{y}, \tilde{z}, \theta)$	$(0,707, -0,023, -0,707, 3.080)$	$(-0,707, -0,029, 0,706, 3,060)$
Rotationsmatrix	$\begin{bmatrix} 0,000 & 0,012 & -0,999 \\ -0,076 & -0,997 & -0,012 \\ -0,997 & 0,076 & 0,000 \end{bmatrix}$	$\begin{bmatrix} 0,002 & -0,017 & -0,999 \\ 0,100 & -0,995 & 0,017 \\ -0,995 & -0,100 & -0,001 \end{bmatrix}$

Tabelle 6.1: Vergleich von Parametrisierungen unterschiedlicher Rotationsformalismen für zwei Datenproben aus dem 300W-LP Datensatz.

verhält sich die Euler-Winkel-Parametrisierung nicht in der gleichen kontinuierlichen Form wie ihre visuelle Erscheinung. Dieses Verhalten kann sich negativ auf die Lernleistung von neuronalen Netzen auswirken.

Eine weitere Art zur Beschreibung von Orientierungen wird als Achsenwinkel-Darstellung bezeichnet. Diese besteht aus dem Einheitsvektor $v = (\tilde{x}, \tilde{y}, \tilde{z})$, der die Achse der Rotation definiert, und dem Winkel θ, der die Rotation des Vektors beschreibt. Eine weitere ähnliche Darstellung ist die Quaternion-Darstellung q mit ebenfalls vier Parametern q_0, q_1, q_2, q_3, die durch $q_0 = cos(\frac{\theta}{2}), q_1 = \tilde{x}\ sin(\frac{\theta}{2}), q_2 = \tilde{y}\ sin(\frac{\theta}{2}), q_3 = \tilde{z} sin(\frac{\theta}{2})$ aus der Achsenwinkel-Beschreibung abgeleitet werden können. Quaternionen und die Achsenwinkel-Darstellung sind nicht vom Gimbal Lock betroffen, aber sie haben immer noch eine Mehrdeutigkeit, welche durch ihre antipodale Symmetrie mit $-v = v$ und $-q = q$ bestimmt wird. Dadurch kann jede Orientierung durch zwei unterschiedliche Parametrisierungen beschrieben werden, die mathematisch maximal voneinander entfernt sind. Folglich sind auch diese beiden Arten für den Einsatz in neuronalen Netzen suboptimal.

Eine eindeutige Notation ist die Rotationsmatrix R^{3x3}, welche aus neun Parametern besteht. Trotz ihrer erhöhten Anzahl an Parametern hat dieser Formalismus den entscheidenden Vorteil, dass er zum einen für jede Rotation eine eindeutige Parametrisierung aufweist. Zum anderen verhält sich diese Parametrisierung kontinuierlich bei wechselnden Rotationen und linear zum visuellen Pedant. Dies

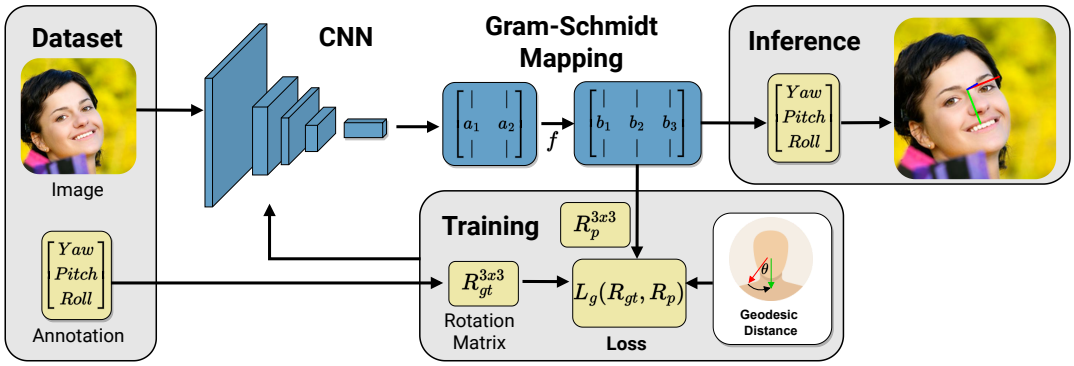


Abbildung 6.1: Übersicht der vorgestellten Methodik zur End-to-End-Kopfposebestimmung.

schafft eine optimale Voraussetzung, um ein systematisches Verständnis der Rotationscharakteristika aus den Trainingsdaten aufzubauen. Abbildung 6.1 zeigt ein Beispiel von zwei Datenproben mit ähnlicher visueller Rotations-Ausprägung. Trotz visueller Ähnlichkeit weisen die Euler-Winkel, Quaternions und Achsenwinkel-Beschreibung jedoch starke Unterschiede in ihrer Parametrisierung auf. Nur die Rotationsmatrizen spiegeln die Ähnlichkeit beider Posen wider.

6.3 Prädikation mittels 6D-Formalismus

In Abschnitt 6.2 wurden unterschiedliche Rotationsformalismen und ihre Eigenschaften vorgestellt. Aus ihnen geht hervor, dass die Rotationsmatrix maßgeblich die geeignetsten Eigenschaften aufweist, um für Aufgaben zur Rotationsbestimmung mittels neuronaler Netze eingesetzt zu werden. Dies wurde von Zhou *et al.* [204] in unterschiedlichen Experimenten bestätigt. So ist jede Rotationsdarstellung mit vier oder weniger Parametern für neuronale Netzwerke suboptimal und mit Problemen im Trainingsprozess verbunden, insbesondere im Fall von vollständigen Rotationsvorhersagen.

Im Folgenden werden weitere mathematische Grundlagen der Rotationsmatrix R erläutert und eine Methodik vorgestellt, welche es ermöglicht, die Matrix in eine noch effizientere Form zu transformieren, um optimale Voraussetzungen für das Training mittels neuronaler Netze zu schaffen. In der Drehgruppe $SO(3)$ hat die Matrix-Darstellung R die Größe 3×3 mit einer Orthogonalitätsbedingung $RR^T = I$, wobei R^T die transponierte Matrix und I die Identitätsmatrix ist.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{6.1}$$

Möchte man die Rotationsmatrix mittels neuronaler Netze direkt regressieren, müssten alle neun korrekten Parameter gefunden werden, die gleichzeitig die Orthogonalitätsbedingung einhalten. Die Orthogonalität kann auch in einem Nachverarbeitungsschritt erzeugt werden, entweder durch die Verwendung des Gram-Schmidt-Verfahrens oder der Singulärwertzerlegung (SVD). Die SVD ist ein aufwendiger Ansatz zur Bestimmung der Vektoren der Rotationsmatrix, die den zuvor Regressierten am nächsten liegen und gleichzeitig orthogonal zueinander sind.

Das Gram-Schmidt-Verfahren erfordert das Verwerfen eines Vektors, um die orthogonale Matrix aus den verbleibenden beiden Vektoren wiederherzustellen. Für den Einsatz in neuronalen Netzen entspricht dies einem weiteren Vorteil, weil statt neun Parametern nun nur noch sechs Parameter prädiziert werden müssen, aus denen im nachfolgenden Schritt mittels des Gram-Schmidt-Orthogonalisierungsverfahrens eine vollständige, orthogonale Rotationsmatrix generiert wird.

Folglich wird der letzte Spaltenvektor der Rotationsmatrix verworfen, wodurch die 3×3-Matrix in eine 6D-Rotationsdarstellung überführt wird:

$$g_{GS} = \left(\begin{bmatrix} | & | & | \\ a_1 & a_2 & a_3 \\ | & | & | \end{bmatrix} \right) = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}. \tag{6.2}$$

Anschließend wird die prädizierte 6D-Darstellungsmatrix wieder in $SO(3)$ überführt mit

$$f_{GS} = \left(\begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix} \right) = \begin{bmatrix} | & | & | \\ b_1 & b_2 & b_3 \\ | & | & | \end{bmatrix}, \tag{6.3}$$

wobei die resultierenden Spaltenvektoren definiert sind als

$$\begin{aligned} b_1 &= \frac{a_1}{||a_1||}, \\ b_2 &= \frac{u_2}{||u_2||} \text{ mit } u_2 = a_2 - (b_1 \cdot a_2) b_1, \\ b_3 &= b_1 \times b_2. \end{aligned} \tag{6.4}$$

Dabei wird der letzte Spaltenvektor einfach durch das Kreuzprodukt bestimmt, das sicherstellt, dass die Orthogonalitätsbedingung für die resultierende 3×3 Matrix erfüllt ist. Folglich muss nach diesem Prinzip das Modell nur sechs Parameter prädizieren, die in einem nachfolgenden Transformationsprozess in eine 3×3 Rotationsmatrix überführt werden, in der die Orthogonalitätsbedingung ebenfalls berücksichtigt ist.

6.4 Geodäsie-basierte Verlustfunktion

Beim Training von Modellen zur Kopfposeschätzung wird üblicherweise die L^2-Norm als Kostenfunktion verwendet, um die Distanz zwischen prädizierter Rotation und der Grundwahrheit zu berechnen. Im Fall von Rotationen im dreidimensionalen Raum mit Rotationsmatrizen entspricht dies der Frobenius-Norm. Dessen Einsatz würde jedoch die geometrische Mannigfaltigkeit der Drehgruppe $SO(3)$ missachten. Stattdessen wird der kürzeste Weg zwischen zwei 3D-Rotationen geometrisch als die geodätische Distanz interpretiert. Seien R_p und $R_{gt} \in SO(3)$ die Prädiktion und die Grundwahrheit in Matrixform, dann ist die geodätische Distanz zwischen beiden Rotationsmatrizen definiert als:

$$L_g = cos^{-1}\left(\frac{tr(R_p R_{gt}^T) - 1}{2}\right) \tag{6.5}$$

Im Folgenden wird diese Metrik als Verlustfunktion im Trainingsprozess der Modelle verwendet, um genaue Distanzinformationen zwischen der prädizierten Kopfpose und der Grundwahrheit zu berechnen. Abbildung 6.1 zeigt das Zusammenspiel des zuvor vorgestellten Ansatzes in seiner Gesamtheit.

6.5 Datensätze

Zum Training und zur Evaluierung der erzeugten Modelle wurden in den vergangenen Jahren unterschiedliche Datensätze der Öffentlichkeit zugänglich gemacht [205, 206, 207, 208, 209, 210, 211]. Die gängigsten Datensätze sind 300W-LP [212], AFLW2000 [213] und BIWI [214], welche im folgenden Abschnitt beschrieben werden. Weiterhin wird der CMU-Panoptic-Datensatz [215] vorgestellt, der durch seine speziellen Eigenschaften zur Generierung neuer Daten im erweiterten Rotationsbereich verbesserte Prädiktionsfähigkeiten für neuronale Netze ermöglicht.

300W-LP 300W-LP besteht aus 66.225 Proben, die aus mehreren Datenbanken zusammengesetzt wurden, darunter LFPW [216], AFW [217], HELEN [218] und iBUG [219]. Der gesamte Datensatz basiert auf rund 4.000 realen Bildern, die durch generative Ansätze synthetisch erweitert wurden (siehe Abbildung 6.2). Der Rotationsbereich der Proben liegt bei ± 89° für den Gierwinkel und für Roll und Nick darunter. Die Grundwahrheit wird im Euler-Winkel-Format bereitgestellt.

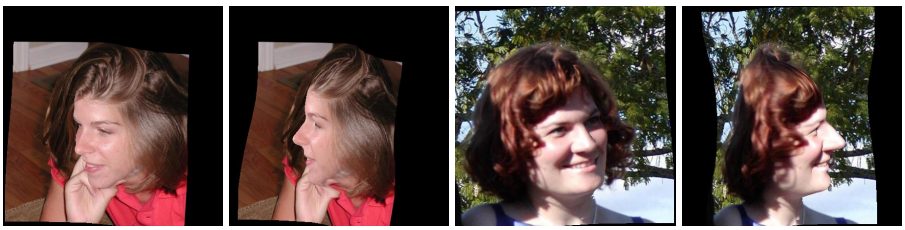


Abbildung 6.2: Datenproben aus der 300W-LP Datenbank.

AFLW2000 Der AFLW2000 Datensatz enthält die ersten 2.000 Proben aus ALFW [220], die mit den 3D-Gesichtern der Grundwahrheit und den entsprechenden 68 Landmarken annotiert sind. Er enthält Proben mit vielseitigen Rotations-Variationen und wechselnden Beleuchtungs- und Verdeckungsbedingungen. Der Gierwinkel liegt bei einigen Proben bei bis zu ± 120°, für den Roll- und Nickwinkel im Bereich ± 90°. Trotz des leicht erweiterten Rotationsbereichs ist der AFLW2000 Datensatz aufgrund der geringen Gesamtanzahl an Proben nur für das Testen von Modellen geeignet. Er ist dafür aufgrund seiner variationsreichen realen Bilder sehr aufschlussreich (siehe Abbildung 6.3).

Abbildung 6.3: Datenproben aus der AFLW2000 Datenbank.

BIWI Der BIWI Datensatz umfasst 15.678 Bilder, die in einer Laborumgebung mit 20 Teilnehmern erstellt wurden. Durch die Laborumgebungen sind Lichtverhältnisse und Kameraperspektive konstant und es treten keine Verdeckungen auf (siehe Abbildung 6.4). Der Rotationsbereich der Kopfposen liegt für den Gierwinkel bei ± 75°, für den Nickwinkel bei ± 60°und für den Rollwinkel bei ± 50°. In diesem Datensatz nimmt der Kopf nur einen kleinen Bereich in den Bildern ein. Daher verwenden wir den MTCNN [221]-Gesichtsdetektor, um die Köpfe locker aus den Bildern zu schneiden.

Abbildung 6.4: Datenproben aus der BIWI Datenbank.

Die drei vorgestellten Datensätze sind die populärsten Datengrundlagen zum Training und zur Validierung von End-to-End Modellen für Kopfpose-Pädiktion [222]. Mit ihrer Verwendung werden jedoch aufgrund ihrer Beschaffenheit die erzeugten Modelle in ihren Prädiktionsfähigkeiten beschränkt, da ihre Datenproben nur Frontalansichten von Gesichtern zur Verfügung stellen, nicht aber die Seitenansicht oder Ansichten des Hinterkopfes. So können nach dem Training eines Modells auf Basis des 300W-LP Datensatzes nur Kopfposen im Winkelbereich von ± 89° zuverlässig bestimmt werden, da darüber hinaus keine Trainingsdaten zur Verfügung stehen. Für den Einsatz in realen Szenarien, z. B. Mensch-Roboter-Interaktionen, birgt dies ein Risiko, da in diesem Fall Kopfposen außerhalb dieses Bereichs vom Modell »geraten« werden. Dies kann folglich zu falschen Schlussfolgerungen im Interaktionsprozess führen. Um das Modell für den gesamten Rotationsbereich des Kopfes anzulernen, wird deshalb der zusätzliche Datensatz CMU-Panoptic [215] vorgestellt.

CMU-Panoptic Der CMU-Panoptic Datensatz ist ursprünglich für die Analyse von Körperposen in Interaktionsszenarien erzeugt worden. Dazu führen Probanden unterschiedliche Aufgaben innerhalb einer Halbkugel aus, die mit 31 gleichmäßig angeordneten HD-Kameras ausgestattet ist. Das Hauptaugenmerk dieses Datensatzes liegt auf der Erfassung der Posen der Versuchspersonen, aber er stellt auch 3D-Annotationen von Gesichtsmerkmalen sowie intrinsische und extrinsische Kamera-Parameter zur Verfügung. Dies ermöglicht die Extraktion von Kopfposen aus unterschiedlichen Kamerawinkeln, die ursprünglich von Zhou *et al.*. [194] nutzbar gemacht

wurden (siehe Abbildung 6.5). Es sind 30 öffentliche Sequenzen mit mehreren Personen pro Szene verfügbar, die in einem Ring stehen, wobei jede Person auf die Mitte der Halbkugel ausgerichtet ist.

Abbildung 6.5: Datenproben aus der verarbeiteten CMU-Panoptic Datenbank.

Datenfusion aus CMU-Panoptic und 300W-LP

Um die Limitation des Rotationsbereichs gängiger Kopfposedatensätze zu überwinden, wird ein neuer Datensatz aus 300W-LP und CMU-Panoptic Daten zusammengesetzt. Bei der Extraktion der Kopfausschnitte aus CMU-Panoptic wurden nur solche mit einer Mindestgröße von 320 für beide Achsen einbezogen, was zu einer Probengröße von insgesamt 113.914 führt. Aufgrund der räumlichen Ausrichtung der Personen während der Interaktionen bestehen die Proben überwiegend aus Ansichten des Hinterkopfes der Probanden. Proben mit frontaler Gesichtsansicht werden überwiegend aufgrund geringer Auflösung aussortiert, da diese Gesichtsbilder aus größerer Entfernung aufgenommen wurden.

Die Proben für Frontalansichten werden hauptsächlich durch den 300W-LP-Datensatz gespeist. Zusammen ergibt sich ein Datensatz mit insgesamt 236.364 Datenproben, welche insbesondere im Gierbereich die gesamte Rotationsebene abdecken. Der Bereich der Neigung wird leicht erweitert, da auch die Stichproben verwendet werden, die von den an der Decke der CMU-Panoptic-Kuppel angebrachten Kameras erzeugt wurden. Die Verteilung dieser neuen Trainingsdaten ist in Abbildung 6.6 dargestellt. Es ist zu beachten, dass die Eulersche Winkelbeschreibung zu Präsentationszwecken verwendet wird, die die Verteilung des visuellen Erscheinungsbildes im Datensatz jedoch nicht exakt widerspiegeln kann, wie im Abschnitt 6.2 erläutert.

6.6 Experimente

In diesem Abschnitt werden neue Modelle zur Kopfposeprädiktion auf Basis des vorgestellten Ansatzes in Abschnitt 6.3 erzeugt und auf dem vorgestellten sowie dem neu zusammengesetzten Datensatz aus Abschnitt 6.5 trainiert. Darauf folgt eine detaillierte Evaluation und ein Vergleich mit dem Stand der Technik in Abschnitt 6.6.1 und weiteren detaillierten Fehleranalysen zum verbesserten Verständnis des vorgeschlagenen Ansatzes.

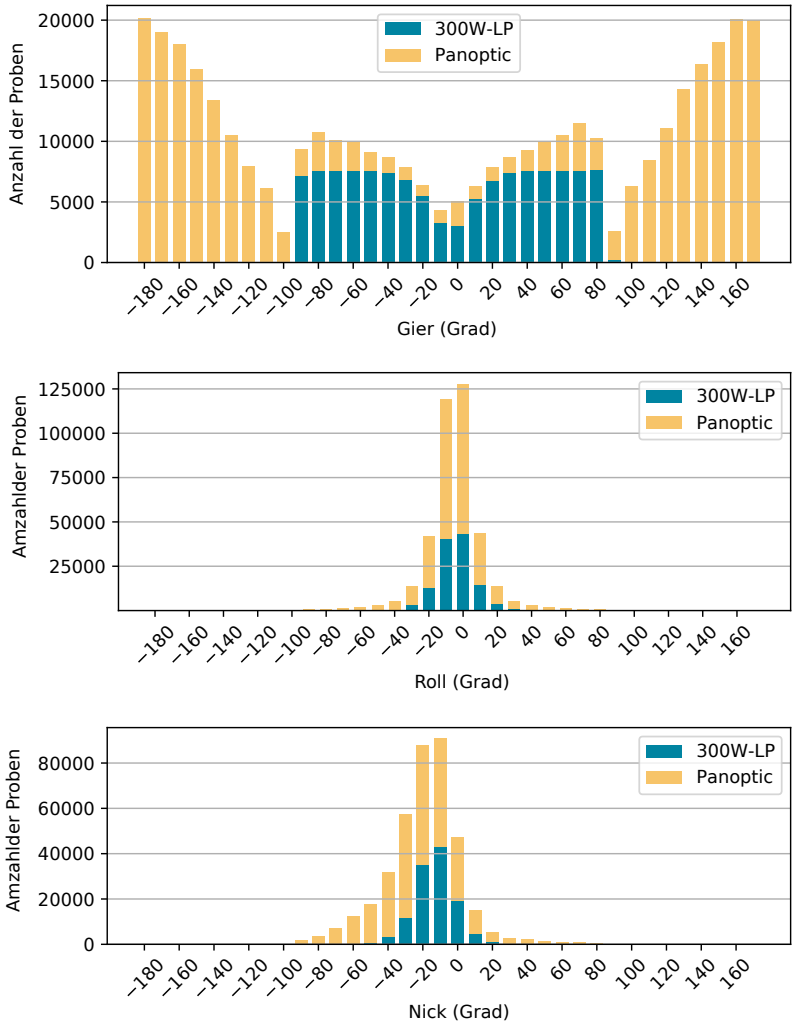


Abbildung 6.6: Labelverteilung des neu fusionierten Datensatzes aus CMU-Panoptic und 300W-LP Datenproben.

Evaluierungsmetriken Zur Evaluierung der Modelle für Kopfposen werden zwei Bewertungsmetriken verwendet, um den Fehler der Kopfausrichtungsschätzungen zu quantifizieren. Die erste ist der mittlere absolute Fehler (MAE) der Euler-Winkel:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}(|x_g - x_p|), \tag{6.6}$$

wobei N die Anzahl der Gesichtsbilder ist und x_g und x_p die Grundwahrheiten bzw. die prädizierte Pose repräsentieren.

Als weitere Metrik wird der mittlere absolute Fehler der Vektoren (MAEV) der Rotationsmatrix vorgestellt. Diese Metrik wurde erstmals von Cao *et al.* [197] eingeführt, um die Einschränkungen der Euler-Repräsentation zu überwinden und ein aussagekräftigeres Bild der Erscheinungsunterschiede zwischen Prädiktionen und Grundwahrheiten zu liefern. Der MAEV definiert den Winkelfehler

zwischen den drei Vektoren der Rotationsmatrix:

$$\text{MAEV} = \frac{1}{N}\sum_{i=1}^{N}\cos^{-1}\left(\frac{v_g \cdot v_p}{|v_g||v_p|}\right), \tag{6.7}$$

wobei N erneut die Anzahl der Gesichtsbilder im Datensatz ist und v_g und v_p die Grundwahrheiten bzw. die prädizierte Pose repräsentieren.

Implementierungsdetails Das vorgestellte Modell wird im Rahmen der ML-Bibliothek PyTorch [223] implementiert. Als Backbone wird das ResNet50 [224] verwendet. Diese Architektur wird in vielen anderen Ansätzen zur Kopfposeprädiktion eingesetzt und ermöglicht deshalb einen fairen Vergleich [192, 225, 197, 193, 201] mit diesen. Die Gewichte des Backbones werden mit dem ImageNet-Datensatz [226] vortrainiert, damit die Kernel des CNNs bereits die Extraktion fundamentaler Merkmale erlernen. Hierfür wird das Netz mit einem Objektklassifikator in den finalen Schichten ausgestattet. Nach dem Training wird dieser Teil vom neuronalen Netz getrennt und stattdessen das in diesem Kapitel vorgeschlagene Modell erzeugt, welches ein einziges Fully Connected Layer mit sechs Neuronen verwendet. Das Modell wird für 80 Epochen mit einer Batch-Größe von 80 unter Verwendung des Adam-Optimizers mit einer Lernrate von $1e^{-4}$ trainiert. Um das volle Generalisierungspotenzial auszunutzen, wird der neue Trainingsdatensatz aus CMU-Panoptic und 300W-LP unter Verwendung von Albumentations [227] augmentiert. Die Augmentierungsstrategie umfasst die zufällige horizontale Spiegelung, zufällige Skalierung und Cropping, zufällige Rotationen bis zu $\pm$ 45 Grad, zufällige Verdeckungen und weitere Bildfarboperationen wie Unschärfe, Helligkeitskontraständerungen und RGB-Verschiebungen.

6.6.1 Cross-Dataset Evaluation

In einem ersten Experiment wird der implementierte Ansatz aus Abschnitt 6.6 mit Methoden aus dem Stand der Technik verglichen. Zu diesem Zweck werden zwei Modelle trainiert. Das erste Modell (*6DRepNet*) wird streng nach der üblichen Trainingskonvention angelernt, indem der synthetische 300W-LP-Datensatz zum Training und die beiden realen Datensätze AFLW2000 und BIWI zum Testen verwendet werden. Dieses Modell gibt Aufschluss über die Performanz des in Abschnitt 6.3 vorgestellten Ansatzes zur Prädiktion von Rotationsmatrizen und der Geodäsiebasierten Verlustfunktion (Abschnitt 6.4).

Für das zweite Modell (*6DRepNet360*) wird die Trainingskonfiguration angepasst, sodass statt des in Abschnitt 6.5 vorgestellten 300W-LP-Datensatzes der kombinierte Datensatz (CMU-Panoptic + 300W-LP) für das Training des vollständigen Rotationsbereichs eingesetzt wird. Die übrige Trainingskonfiguration bleibt gleich, um den Fokus auf die Auswirkungen der neuen Trainingsdaten auf die Auswertung zu legen.

Tabelle 6.2 und Tabelle 6.3 zeigen die Ergebnisse der beiden Modellkonfigurationen im Vergleich mit den Ergebnissen anderer Methoden aus dem Stand der Technik. Zur besseren Interpretation wurde zudem eine zusätzliche Spalte (R) hinzugefügt, um zu zeigen, welche Methoden für die Vorhersage eines größeren Bereichs von Rotationen trainierbar sind und welche ihre Vorhersagen

Methode	R	Euler							
		AFLW2000				BIWI			
		Gier	Nick	Roll	MAE	Gier	Nick	Roll	MAE
3DDFA [228]	✗	4,71	27,1	28,4	20,1	5,50	41,1	13,2	20,2
Dlib [181]	✗	8,49	11,3	22,9	14,2	11,9	13,0	19,6	14,8
HopeNet [192]	✗	6,40	6,53	5,39	6,11	4,54	5,15	3,37	4,36
FSA-Net [196]	✗	4,83	6,25	4,94	5,34	4,64	5,61	3,57	4,61
HPE [202]	✗	4,80	6,18	4,87	5,28	3,12	5,18	4,57	4,29
QuatNet [193]	✗	3,97	5,62	3,92	4,50	**2,94**	5,49	4,01	4,15
TriNet [197]	✗	4,36	5,81	4,51	4,89	3,11	5,09	5,20	4,47
WHENet-V [194]	✗	4,44	5,75	4,31	4,83	3,60	4,10	2,73	3,48
WHENet [194]	✓	5,11	6,24	4,92	5,42	3,99	4,39	3,06	3,81
FDN [200]	✗	3,78	5,61	3,88	4,42	4,52	4,70	**2,56**	3,93
Viet *et al*. [203]	✓	-	-	-	-	4,62	4,29	4,52	4,48
MFDNet [198]	✗	4,30	5,16	3,69	4,38	3,40	4,68	2,77	3,62
DDD-Pose [201]	✗	4,38	4,85	3,44	4,22	4,60	6,02	2,94	4,52
Liu *et al*. [199]	✗	**3,03**	5,06	3,68	3,93	4,12	5,61	3,15	4,29
img2pose [229]	✗	3,42	5,03	3,28	3,91	4,56	**3,54**	3,25	3,78
MNN [189]	✗	3,34	4,69	3,48	3,83	3,98	4,61	2,39	3,66
RankPose [225]	✗	3,26	4,72	3,23	3,74	4,54	5,61	3,05	4,40
6DRepNet ***(Vorg. Ansatz)***		3,27	**4,58**	**2,98**	**3,61**	3,23	5,32	2,78	3,78
6DRepNet360 ***(Vorg. Ansatz)***		3,73	5,52	3,53	4,26	3,37	3,87	2,93	**3,39**

Tabelle 6.2: Vergleich des MAE der Euler-Prädiktion mit anderen Methoden aus dem Stand der Technik auf den AFLW2000- und BIWI-Datensätzen. Alle Modelle wurden auf dem 300W-LP-Datensatz trainiert. Methoden mit positivem R sind methodisch in der Lage, einen größeren Bereich von Rotationen vorherzusagen.

inhärent auf frontale Posen beschränken. Von den 15 aufgelisteten Methoden sind dabei nur zwei für einen erweiterten Rotationsbereich einsetzbar.

6DRepNet Die Ergebnistabellen zeigen auf, dass das 6DRepNet Modell, das ausschließlich auf dem 300W-LP Datensatz trainiert wurde, alle anderen Methoden aus dem Stand der Technik auf dem AFLW2000 Testdatensatz übertrifft und den derzeitigen Spitzenreiter RankPose auf AFLW2000 bei den Euler- und Vektorfehlern unterbietet. Neben der Gesamtfehlerrate erzielt das vorgeschlagene Modell die niedrigste Fehlerrate für den Nick- und Rollfehler und nahezu identische Ergebnisse zum Stand der Technik im Gierbereich. Dies deutet auf ein sehr stabiles Lernverhalten während des Trainingsprozesses hin, um die dargestellten Prädiktionen zu erzielen. Beim BIWI-Datensatz erzielt es konkurrenzfähige Ergebnisse in Bezug auf den MAE und beste Ergebnisse in Bezug auf den MAEV. Letzteres sollte mit Vorsicht interpretiert werden, da für die MAE-Spitzenreiter keine MAEV-Ergebnisse von den Autoren veröffentlicht wurden.

Methode	R	Vektor							
		AFLW2000				BIWI			
		V1	V2	V3	MAEV	V1	V2	V3	MAEV
3DDFA [228]	✗	30,6	39,1	18,5	29,4	23,3	45,0	35,1	34,5
Dlib [181]	✗	26,6	28,5	14,3	23,1	24,8	21,7	14,3	20,3
HopeNet [192]	✗	7,98	6,40	8,54	7,64	6,21	5,75	7,05	6,33
FSA-Net [196]	✗	6,88	6,52	7,28	6,89	6,27	6,29	7,38	6,65
TriNet [197]	✗	6,16	5,95	6,82	6,31	6,58	5,80	7,55	6,64
img2pose [229]	✗	6,00	5,20	6,55	5,92	4,83	5,28	6,04	5,38
RankPose [225]	✗	4,40	4,42	5,08	4,63	5,81	5,91	7,39	6,37
6DRepNet ***(Vorg. Ansatz)***		**4,33**	**4,17**	**5,06**	**4,52**	**4,66**	5,29	**6,03**	**5,32**
6DRepNet360 ***(Vorg. Ansatz)***		5,18	4,70	6,04	5,31	**4,64**	**4,57**	**5,34**	**4,85**

Tabelle 6.3: Vergleich des MAE der Vektor-Prädiktion (MAEV) mit Methoden aus dem Stand der Technik auf den AFLW2000- und BIWI-Datensätzen. Alle Modelle wurden auf dem 300W-LP-Datensatz trainiert. Methoden mit positivem ***R*** sind methodisch in der Lage, einen größeren Bereich von Rotationen vorherzusagen.

6DRepNet360 Das zweite Modell, 6DRepNet360, erzielt sehr konkurrenzfähige Ergebnisse bei AFLW2000 und sogar neue Spitzenergebnisse bei BIWI, indem es WHENet-V um 3% übertrifft. Bemerkenswerterweise unterscheidet sich dieses Modell nur in seinen Trainingsdaten, wobei die hinzugefügten Daten darauf abzielen, den vorhersagbaren Erkennungsbereich der Gierdrehung zu erweitern. Diese Proben enthalten jedoch zahlreiche stärkere Gierdrehungen als 300W-LP (siehe Abbildung 6.6).

Es lässt sich argumentieren, dass diese Stichproben die Leistung des Modells bei der Verarbeitung der anspruchsvollen Posen aus dem BIWI-Datensatz begünstigen, da der Fehler für die Neigung im Vergleich zum ausschließlich auf 300W-LP trainierten Modell (6DRepNet) um 33% reduziert wird. Bemerkenswerterweise wurde WHENet [194] auch für unbegrenzte Gier-Prädiktionen (± 360 Grad) trainiert und ist daher am besten für einen Vergleich mit dem 6DRepNet360 geeignet. Während WHENet sogar schlechter abschneidet als sein 300W-LP-Äquivalent WHENet-V, erreicht das 6DRepNet360-Modell bei AFLW2000 eine um über 20% niedrigere Fehlerrate und bei BIWI eine um über 10% höhere Genauigkeit. Es kann daher angenommen werden, dass die Wahl der 6D-Rotationsmatrix als Rotationsdarstellung anstelle von WHENets Euler-Winkel einen wesentlichen Einfluss auf die geringere Fehlerrate des 6DRepNet360-Modells hat.

In Bezug auf die Rotationsdarstellung ist TriNet [197] die ähnlichste Methode zum 6DRepNet. Aber im Gegensatz zum 6-Parameter-Ansatz mit anschließender neun Parameter Rekonstruktion mittels Gram-Schmidt-Ansatz wird beim TriNet die gesamte Rotationsmatrix mit allen neun Parametern geschätzt und anschließend mittels SVD korrigiert. Basierend auf den Ergebnissen lässt sich deshalb argumentieren, dass der in diesem Kapitel vorgeschlagene Ansatz effektiver ist und durch ein effizienteres Training zu einer höheren Prädiktions-Genauigkeit führt.

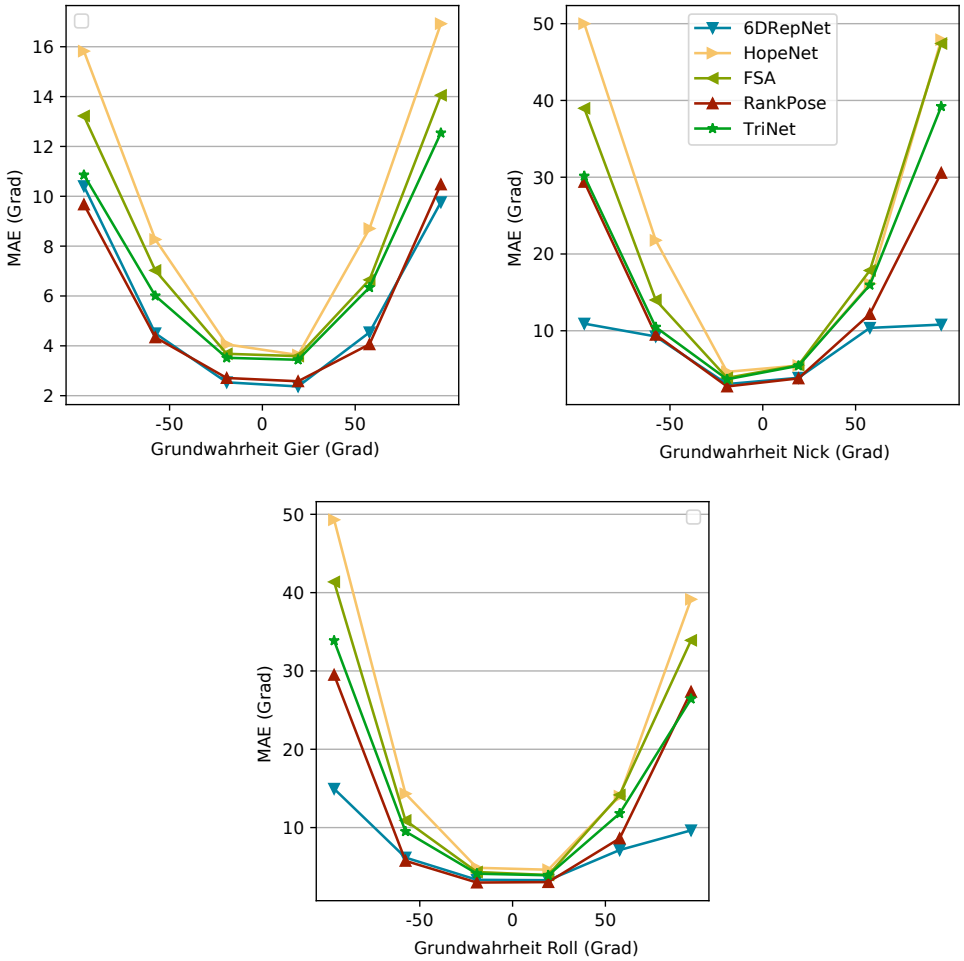


Abbildung 6.7: Fehleranalyse für Label-Intervalle auf dem AFLW2000 Datensatz.

Fehleranalyse für Label-Intervalle Um einen detaillierteren Eindruck vom Prädiktionsverhalten des vorgeschlagenen 6DRepNet-Ansatzes zu erhalten, wird eine weitere Fehleranalyse mit vier anderen Methoden aus dem Stand der Technik (HopeNet, FSA-Net, RankPose, TriNet) durchgeführt, bei der die Prädiktions-Fehler auf AFLW2000 in Intervalle von 33 Grad aufgeteilt werden. So kann genauer untersucht werden, wie sich die Modelle in den unterschiedlichen Rotationsbereichen verhalten. Alle Modelle werden ausschließlich auf dem 300W-LP Datensatz trainiert. Die Ergebnisse sind in Abbildung 6.7 illustriert, wobei jeder Eulersche Winkel (Roll, Nick, Gier) in einem eigenen Diagramm abgebildet ist.

Die Diagramme zeigen auf, dass im Allgemeinen die Fehlerrate für alle Methoden mit stärkerer Rotation zunehmen. Auffallend ist jedoch, dass dieser Fehleranstieg bei 6DRepNet im Vergleich zu allen anderen Methoden deutlich geringer ausfällt, insbesondere für den Nick- und Rollwinkel. Während Tabelle 6.2 aufzeigt, dass der vorgeschlagene Ansatz das RankPose Modell insgesamt um 3% übertrifft, zeigt diese detaillierte Fehleranalyse, dass das 6DRepNet bei extremen Nick- und Rolldrehungen über 60% geringere Fehlerraten erzielt. Dies ist eine weitere Bestätigung dafür, dass der vorgeschlagene Ansatz nicht nur den aktuellen Stand der Technik übertrifft, sondern gleichzeitig

Methode	BIWI Euler			
	Gier	Nick	Roll	MAE
HopeNet [192]	3,35	4,66	3,00	3,67
FSA-Net [196]	6,79	9,18	4,56	6,84
FDN [200]	3,00	3,98	2,88	3,29
MDFNet [198]	2,99	3,68	2,99	3,22
TriNet [197]	2,79	3,28	2,53	2,87
DDD-Pose [201]	3,04	**2,94**	2,43	2,80
6DRepNet	**2,39**	2,96	**2,05**	**2,47**

Tabelle 6.4: In-Dataset Vergleich der Euler-Fehler mit Methoden aus dem Stand der Technik auf dem BIWI Datensatz.

auch in extrem herausfordernden Proben robuste Einschätzungen liefert.

6.6.2 In-Dataset Evaluation

In diesem Abschnitt werden mehrere Experimente auf dem BIWI Datensatz und dem neu vorgestellten kombinierten CMU-Panoptic + BIWI Datensatz durchgeführt. Hierbei werden die Datensätze in Trainings- und Testdaten aufgeteilt. Die Testdurchläufe zielen auf ein besseres Verständnis des Prädiktionsverhaltens des vorgestellten Modells innerhalb eines Datensatzes im Vergleich zu anderen Ansätzen aus dem Stand der Technik.

Methode	BIWI Vektor			
	V1	V2	V3	MAEV
FSA-Net [196]	9,09	10,19	11,26	10,18
HopeNet [192]	5,55	5,64	5,78	5,66
TriNet [197]	4,12	4,47	4,24	4,28
6DRepNet	**3,39**	**3,27**	**3,89**	**3,52**

Tabelle 6.5: In-Dataset Vergleich der Vektor-Fehler mit Methoden aus dem Stand der Technik auf dem BIWI Datensatz.

BIWI In einen BIWI-Testdurchlauf wird die von Yang *et al*. [196] vorgeschlagene Evaluationsstrategie durchgeführt und der BIWI Datensatz nach dem Zufallsprinzip in einem Verhältnis von 7:3 zum Trainieren und zum Testen aufgeteilt.

Tabelle 6.4 (MAE) zeigt die Ergebnisse des 6DRepNet Modells im Vergleich zu anderen Methoden aus dem Stand der Technik. Die Tabelle verdeutlicht, dass die 6DRepNet Methode alle anderen Ansätze mit einem Abstand von mehr als 10% übertrifft. In Bezug auf die einzelnen Eulerschen Winkel liefert der vorgeschlagene Ansatz sehr konsistente Ergebnisse und erzielt die besten Ergebnisse für den Gier- und Rollwinkel und gleichwertige Ergebnisse zum Stand der Technik DDD-Pose [201] für den Nickwinkel. Dies unterstützt die beobachtete Robustheit in der Datensatz-übergreifenden Auswertung und zeigt, dass das Erreichen stabiler, genauer Ergebnisse für alle drei Winkel nicht nur

Methode	CMU-Panoptic + 300W-LP			
	Gier	Nick	Roll	MAE
Viet et al. [203]	9,55	11,29	8,32	9,72
WHENet [194]	8,51	7,67	6,78	7,66
6DRepNet360	**2,08**	**3,16**	**2,75**	**2,66**

Tabelle 6.6: Testergebnisse auf dem kombinierten CMU-Panoptic + 300W-LP Datensatz. 70% des Datensatzes werden zum Training und die restlichen 30% zum Testen verwendet.

vom trainierten Datensatz abhängt, sondern vielmehr von der in diesem Kapitel vorgeschlagenen Methode selbst. Dies spiegelt sich auch in Tabelle 6.5 wider, in der der vorgestellte Ansatz die besten MAEV-Gesamtergebnisse sowie die besten Ergebnisse für jeden einzelnen Vektor erzielt.

CMU-Panoptic + 300W-LP In einem letzten Experiment werden die Methoden in einem In-Datensatz-Test auf dem vorgeschlagenen kombinierten Datensatz untersucht, der die Daten des CMU-Panoptic und des 300W-LP-Datensatzes umfasst. Auch hier wurde der Datensatz nach dem Zufallsprinzip in 70% Trainingsdaten und 30% Testdaten aufgeteilt. Die Ergebnisse zeigt die Tabelle 6.6.

Viet et al. [203] und WHENet [194] sind dabei die einzigen Methoden, die ihn ähnlicher Weise Testergebnisse auf dem CMU-Panoptic Datensatz veröffentlicht haben. Die Prädiktionsergebnisse von [203] berücksichtigen jedoch zusätzlich die Gesichtserkennung, und ihr Testsatz umfasst ausschließlich Stichproben von CMU-Panoptic. WHENet [194] testet mit einer Kombination aus CMU-Panoptic und 300W-LP, deren Größe und Zusammensetzung jedoch nicht spezifiziert sind. Somit basieren die verglichenen Ergebnisse auf nicht identischen Testbedingungen und bieten keine optimale Vergleichsgrundlage. Die Ergebnisse dienen hauptsächlich als Referenz für zukünftige Arbeiten, um zukünftigen Ansätzen die Möglichkeit eines genauen Vergleichs zu ermöglichen.

6.6.3 Ablationsstudie

Im Folgenden wird eine Ablationsstudie durchgeführt und analysiert, wie sich die einzelnen Komponenten des 6DRepNet Modells auf die erzielten Ergebnisse auswirken. Dazu gehören das Backbone, das für die Merkmalsextraktion verantwortlich ist, die 6D-Rotationsrepräsentation und die vorgeschlagene Verlustfunktion, die geodätischen Distanz, die sich von anderen Methoden aus der Literatur unterscheidet.

Verlustfunktion Methoden aus dem Stand der Technik verwenden üblicherweise den mittleren quadratischen Fehler (MSE)

$$L_{MSE} = \frac{1}{N}\sum_{i=0}^{N}(y_p - y_{gt})^2 \tag{6.8}$$

Methode	AFLW2000				BIWI			
	Gier	Nick	Roll	MAE	Gier	Nick	Roll	MAE
L_{MSE}	3,38	4,89	3,33	3,87	3,19	6,52	2,81	4,17
$L_g + L_{MSE}$	**3,26**	4,65	3,09	3,67	**3,17**	5,69	**2,76**	3,88
L_g ***(Vorg. Ansatz)***	3,27	**4,58**	**2,98**	**3,61**	3,23	**5,32**	2,78	**3,78**

Tabelle 6.7: Analyse des Einflusses der Verlustfunktionen L_{MSE} und L_g auf den MAE.

Methode	Distanzfunktion	AFLW2000			
		Gier	Nick	Roll	MAE
Euler	L_{MSE}	9,57	6,35	5,16	7,03
Quaternion	L_{MSE}	6,33	6,18	5,07	5,86
Rotation Matrix	L_{MSE}	4,00	5,34	3,96	4,43
6D ***(Vorg. Ansatz)***	L_{MSE}	3,38	4,89	3,33	3,87
6D ***(Vorg. Ansatz)***	L_g ***(Vorg. Ansatz)***	**3,27**	**4,58**	**2,98**	**3,61**

Tabelle 6.8: Trainingsverhalten unterschiedlicher Rotationsrepräsentationen und Distanzfunktionen.

für die Berechnung der Distanz zwischen Grundwahrheit und Prädiktion im Trainingsprozess. In diesem Kapitel wurde jedoch die geodätische Distanz als bessere Alternative vorgeschlagen. Um dies zu beweisen, wird ein weiteres Experiment durchgeführt. Bei diesem wird der Trainingsprozess mit unterschiedlichen Verlustfunktionen wiederholt: Der MSE-Distanzmetrik, der Kombination aus MSE und dem geodätischen Verlust L_g (siehe Gleichung 6.5) und ausschließlich der geodätischen Distanz. Tabelle 6.7 zeigt die Ergebnisse des Experiments und gibt Aufschluss darüber, dass das Modell mit der geodätischen Distanzmetrik besser abschneidet als jenes mit MSE und der Kombination aus MSE und L_g.

Rotationsrepräsentation Ein weiterer elementarer Faktor ist der verwendete Rotationsformalismus. Wir testen mehrere Trainingsdurchläufe mit verschiedenen Rotationsformalismen. Hierbei wird das Modell entweder mit drei Ausgangsneuronen für das auf Euler basierende Modell, vier Neuronen für das Quaternionen-basierte Modell, sechs für den 6D-Formalismus oder neun für das auf der Rotationsmatrix basierende Modell ausgestattet. Alle Modelle werden unter Verwendung des ResNet50-Backbones und der MSE-Verlustfunktion trainiert. Zusätzlich wird ein weiteres auf 6D basierendes Modell erzeugt, das die vorgeschlagene geodätische Distanz als Verlustfunktion verwendet.

Die Ergebnisse werden in Tabelle 6.8 präsentiert. Sie zeigen den höchsten Fehler für das Euler-Winkelmodell auf, gefolgt von den Quaternionen- und Rotationsmatrix-basierten Modellen. Die auf dem 6D-Formalismus basierenden Modelle erzielen die besten Ergebnisse, wobei das Modell mit geodätischem Verlust beim Fehler in Gier-, Nick- und Rollwinkel die MSE-Modelle übertrifft. Diese Resultate bestätigen die Leistungsfähigkeit des vorgeschlagenen Ansatzes, die 6D-Rotationsrepräsentation mit der geodätischen Distanz zu kombinieren und somit ein effizientes Training zu ermöglichen.

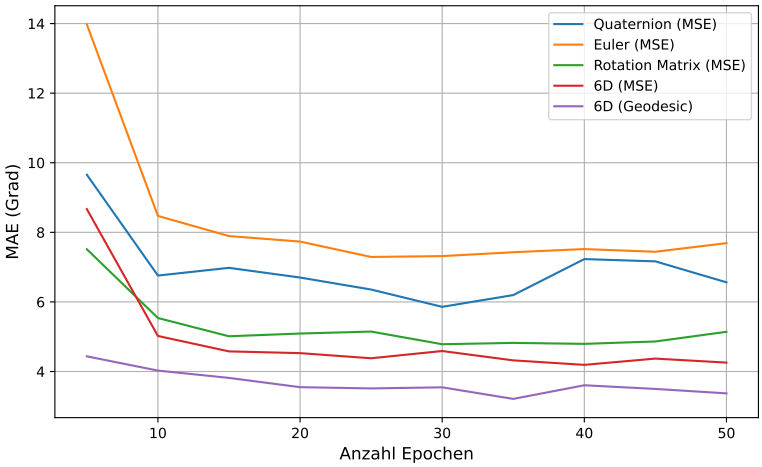


Abbildung 6.8: Vergleich des MAE für verschiedene Rotationsformalismen auf der Grundlage eines ResNet50-Backbones und unterschiedlicher Verlustfunktionen. Für die vorgeschlagene 6D-Darstellung wird eine zusätzliche Reihe mit geodätischem Verlust bereitgestellt. Alle Ergebnisse basieren auf dem AFLW2000-Testsatz.

Methode	AFLW2000				BIWI			
	Gier	Nick	Roll	MAE	Gier	Nick	Roll	MAE
ResNet18	3,18	4,81	3,26	3,75	3,09	5,94	2,93	3,99
ResNet50	3,27	4,58	2,98	3,61	3,23	5,32	2,78	3,78

Tabelle 6.9: Vergleich der Ergebnisse zwischen einem ResNet18 und einem ResNet50 Backbone.

Um den Trainingsprozess weiter zu analysieren, wird in Abbildung 6.8 die Testleistung der trainierten Modelle am AFLW2000-Testset über die Trainingsepochen hinweg illustriert. Es zeigt, dass das auf 6D basierende Modell mit geodätischem Verlust bereits ab Epoche fünf alle anderen Modelle über alle Epochen hinweg übertrifft und damit seine schnelle Konvergenzrate demonstriert.

Backbone In einem letzten Experiment wird der Einfluss des gewählten Backbones auf die Ergebnisse analysiert. Die Ergebnisse aus Tabelle 6.2 und Tabelle 6.5 haben bereits die Überlegenheit des vorgeschlagenen Ansatzes gegenüber anderen Methoden aus dem Stand der Technik mit demselben Backbone aufgezeigt. Dennoch sollen die Auswirkungen der Anzahl der Parameter auf die erwarteten Ergebnisse untersucht werden.

In Tabelle 6.9 werden deshalb die bisherigen Ergebnisse des 6DRepNet Modells (ResNet50) mit einem kleineren Modell verglichen, das mit dem ResNet18 trainiert wurde. Es ist bemerkenswert, dass das um 50% kleinere ResNet18-Modell immer noch bessere Ergebnisse für den AFLW2000-Datensatz erzielt, als alle anderen Vergleichs-Methoden aus Tabelle 6.2, mit einer Ausnahme. Für den BIWI-Datensatz verringert sich die Genauigkeit im Vergleich zu ResNet50 nur um einen sehr

geringen Prozentsatz. Dies bestätigt, dass die Gesamtleistung des vorgestellten Modells in erster Linie auf die 6D-Rotationsdarstellung und kaum auf das verwendete Backbone zurückzuführen ist. Außerdem zeigt es, dass das häufig verwendete ResNet50 nicht notwendig ist, um eine angemessene Genauigkeit zu erreichen, da das effizientere ResNet18 eine ähnliche Prädiktionsgenauigkeit zulässt. Dies ist eine wichtige Erkenntnis, wenn die Kopfposenprädiktion in Umgebungen mit begrenzten Rechenressourcen eingesetzt werden soll.

6.6.4 Qualitative Ergebnisse

Abbildung 8.7 zeigt einige exemplarische qualitative Ergebnisse auf Basis des 6DRepNet360-Modells. Die erste Zeile illustriert die Visualisierung von Prädiktionen auf Testbildern aus dem AFLW2000-Datensatz mit starken Variationen und wechselnden Hintergründen, Beleuchtung und Kamera-Perspektiven. Die zweite Reihe zeigt Testergebnisse mit sehr starken Kopfdrehungen aus dem CMU-Panoptic-Testset, die über die üblichen $\pm$ 99° Einschränkungen hinausgehen. Im Gegensatz zum AFLW2000 Datensatz wurden die Daten in einem Labor mit gleichbleibenden Beleuchtungsbedingungen und Hintergrundumgebungen aufgenommen. Dennoch ist der 6DRepNet Ansatz in der Lage, die Kopfposen aus unterschiedlichen Kamera-Perspektiven robust zu prädizieren.

Ein sehr bemerkenswertes Beispiel ist die letzte Testprobe, die durch besonders wenig markante Merkmale ein besonders herausforderndes Beispiel darstellt. Während bei frontalen Gesichtern sogar stärker gedrehte Posen aussagekräftige Merkmale liefern, beschränken sich in diesem Beispiel die visuellen Orientierungspunkte hauptsächlich auf die Form des Kopfes. Dennoch ist das vorgeschlagene Modell in der Lage, selbst für diese schwierigen Kopfhaltungen zuverlässige Orientierungen vorherzusagen.

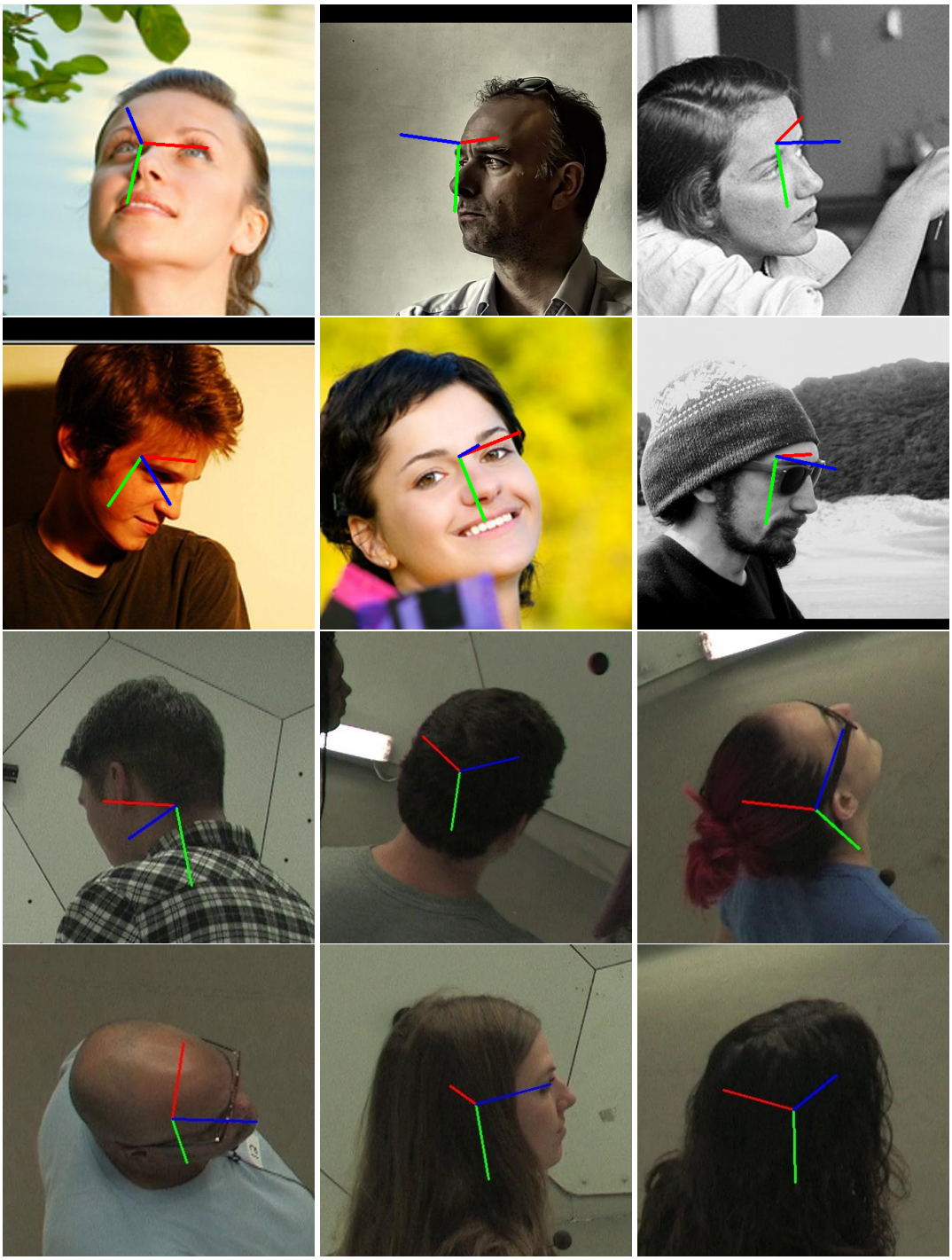


Abbildung 6.9: Qualitative Beispiele von Poseprädiktion auf Basis des AFLW2000-Datensatzes (ersten zwei Reihen) und des CMU-Panopic-Testdatensatzes (letzten zwei Reihen). © 2024 IEEE

6.7 Limitationen

Der in diesem Kapitel vorgeschlagene Ansatz zur bildbasierten Kopfposeschätzung erreicht eine genaue und robuste Prädiktion für einen stark erweiterten Rotationsbereich. Dies gilt insbesondere für den Gierwinkel, der in gängigen Anwendungsszenarien die stärksten Drehungen erfährt. Aber

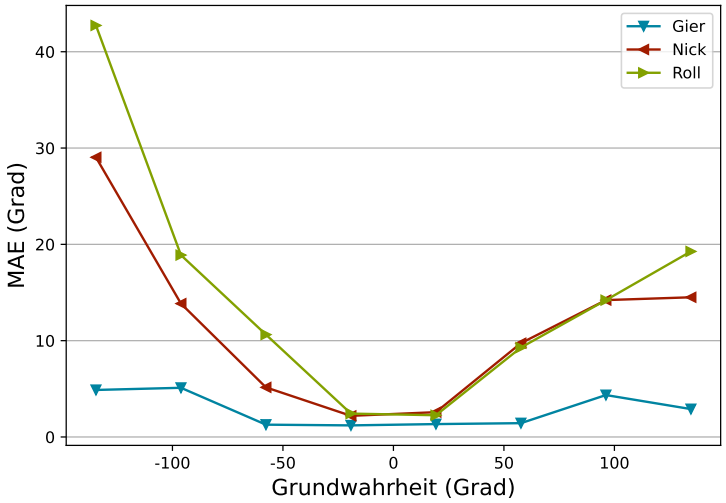


Abbildung 6.10: Euler-Fehler für das CMU-Panoptic + 300W-LP Testset.

auch der Roll- und der Nickwinkel können starke Rotationen erreichen, die in unseren Trainingsdaten nur am Rande vertreten sind (siehe Abbildung 6.6). Dies kann zu einer verminderten Robustheit und Genauigkeit in Anwendungsszenarien mit ungewöhnlichen Kamerawinkeln und Kopfhaltungen führen. Um dies zu analysieren, wurde der Fehler des 6DRepNet360 Modells auf dem Testset des vorgeschlagenen CMU-Panoptic + 300W-LP 70/30-Split aus dem Abschnitt 6.6.2 intervallweise berechnet. Die Ergebnisse sind in Abbildung 6.10 dargestellt und demonstrieren, dass die Fehlerrate für den Gierwinkel konstant niedrig bleibt, während die Fehlerrate für den Roll- und Nickwinkel bei stärkeren Rotationen zunimmt. Dies zeigt, dass es noch an Trainings- und auch Testdaten für diesen erweiterten Rotationsbereich mangelt. Im kombinierten Testsatz überschreiten nur drei Proben den Bereich ± 100 Grad auf der Roll-Achse und nur fünf Proben den Bereich ± 100 Grad auf der Nick-Achse. In den vorgestellten Experimenten wurde versucht, diesem Mangel an Daten entgegenzutreten, indem durch synthetische Bildrotationen im Augmentierungsprozess der Roll- und Nickwinkel einiger Trainingsdaten stichprobenartig vergrößert wurden.

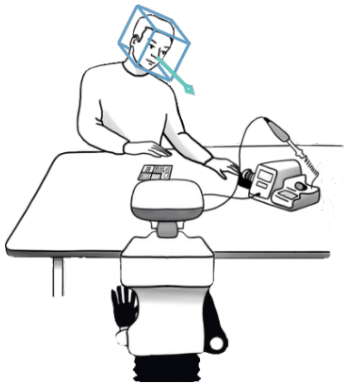
6.8 Diskussion

In diesem Kapitel wurde ein neuer Ansatz zur bildbasierten Kopfposeschätzung vorgestellt, welcher in der Lage ist, präzise und robust einen stark erweiterten Rotationsbereich zu verarbeiten und zu prädizieren. Insbesondere die Schätzung von stark rotierten Kopfposen (siehe Abbildung 8.7) stellt eine Herausforderung dar, die im Stand der Technik bisher wenig Forschung erfahren hat. Um sich diesem Problem anzunehmen, wird zunächst eine effiziente 6D-Rotationsmatrixdarstellung formuliert, um eine eindeutige und kontinuierliche Parametrisierung zu erhalten. Dieser Ansatz bildet die Grundlage für ein stabiles und präzises Modelltraining. Weiterhin wurde zur Verbesserung des Trainingsprozesses Geodäsie-basierte Distanzfunktion vorgeschlagen.

Für das Training des Modells wird ein neuer Datensatz mit erweitertem Rotationsbereich generiert. Hierzu wird auf Basis des CMU-Panoptic-Datensatzes der bisher genutzte 300W-LP-Datensatz

systematisch erweitert, um einen neuen, größeren Trainingsdatensatz zu erzeugen, der insbesondere im Gierbereich den kompletten Rotationsbereich abdeckt.

Das vorgeschlagene 6DRepNet Modell wurde in vielfältigen Experimenten detailliert evaluiert. Dabei konnte aufgezeigt werden, dass die vorgeschlagene 6D-Rotationsdarstellung im Vergleich zum Stand der Technik überlegene Prädiktionsgenauigkeiten erzielt und in der Lage ist, den gesamten Bereich der Kopfcodeorientierung effizient zu erlernen. Mit der Generierung eines neuen Trainingssatzes durch die Kombination von Proben aus CMU-Panoptic und 300W-LP wurde ein Modell erzeugt, welches robust und präzise bisher unerreichte Rotationsbereiche prädizieren kann und damit einen wichtigen wissenschaftlichen Beitrag darstellt.



KAPITEL 7

Simultane Kopfpose- und Blickrichtungsschätzung

Im vorangegangenen Kapitel wurde ein Ansatz zur Kopfposeprädiktion vorgestellt, anhand dem die Ausrichtung der visuellen Aufmerksamkeit von Personen abgeleitet werden kann. Die Kopfpose ist besonders dann ein zuverlässiges und robustes Merkmal, wenn die Augen der Personen nicht genau zu erkennen oder bedeckt sind. Sobald die Augen jedoch erfassbar sind, bietet die Blickrichtungserkennung anhand der Pupillen eine noch genauere Einschätzung der momentanen Aufmerksamkeit.

In diesem Kapitel wird eine neue Methode vorgeschlagen, welche die Kopfposeschätzung und die Blickrichtungsschätzung mittels eines Multi-Task-Ansatzes in einem Modell vereint. Die Motivation liegt hierbei in der Ausschöpfung synergetischer Effekte zwischen beiden Merkmalsausprägungen. Durch den Einbezug von Trainingsdaten aus beiden Disziplinen kann die Varianz innerhalb des Datensatzes gestärkt und somit die Performanz und Generalisierungseigenschaften der Modelle verbessert werden. Gleichzeitig sorgt das Teilen von Parametern (*parameter sharing*) im Vergleich zum Einsatz von Modellen als Einzelaufgabe zu einer Reduzierung des rechnerischen Aufwands.

Im ersten Abschnitt dieses Kapitels wird zunächst eine Übersicht über verwandte Arbeiten gegeben (Abschnitt 7.1). Im Anschluss wird der Ansatz über das Multi-Task-Modell zur simultanen Kopfpose- und Blickrichtungsschätzung erläutert und Details zur Architektur und Trainingsstrategie diskutiert (Abschnitt 7.2). Es folgt die Vorstellung von relevanten Datensätzen (Abschnitt 7.3) und mehrere Experimente zur Evaluierung der eingeführten Methodik. Diese werden ergänzt durch eine Ablationsstudie, die das Modellverhalten näher beschreibt (Abschnitt 7.4). Das Kapitel schließt mit einer zusammenfassenden Diskussion (Abschnitt 7.5).

Forschungsbeitrag

» Es wird ein neues, effizientes Multi-Task-Modell zur simultanen Prädiktion von Kopfpose und Blickrichtung vorgeschlagen.

» Es wird eine Trainingsstrategie eingesetzt, die es ermöglicht, das Modell iterativ über zwei verschiedene Datensätze zu trainieren.

» Der symbiotische Einsatz beider Aufgaben in einem gemeinsamen Netz führt zu Verbesserung der Generalisierung mit bis zu 28% besseren Prädiktionsergebnissen gegenüber dem Stand der Technik.

7.1 Verwandte Arbeiten

Die bildbasierte Erkennung der Blickrichtung ist seit jeher eine intensiv erforschte Herausforderung, zu der bereits eine Vielzahl an Datensätzen [230, 231, 232, 233] und Methoden [234, 230, 235, 236, 237] vorgestellt wurden. Typische Ansätze basieren auf CNNs, die die Blickrichtungsschätzung auf Basis eines einzelnen Bildes prädizieren. Hierbei entsprechen die Eingangsbilder entweder einzelnen Bildauschnitten der Augen [238], des gesamten Gesichts oder ihrer Kombination [237, 231]. Der Einbezug von Bildausschnitten der Augen vereinfacht den Modellen das Erlernen essenzieller Merkmale aus der Augenregion, benötigt jedoch einen separaten Verarbeitungsschritt zur Detektion und Extraktion der Augenpartie aus den Bildern. Bei der Eingabe des gesamten Gesichtsbereichs ist dies nicht nötig. Lediglich die Gesichtsdetektion wird benötigt, welche in einem multifunktionalen System jedoch oftmals bereits vorliegt (z. B. zur Schätzung der Kopfpose). Auf der anderen Seite ist bei diesem Ansatz das Anlernen des Modells herausfordernder, da der Großteil des Gesichtsbereichs weniger relevante Informationen über die genaue Blickrichtung enthält. Der Einfluss der unterschiedlichen Aufnahmebedingungen wie Lichtverhältnisse, Gesichtsausprägungen und Expressionsverhalten der Probanden können somit stärker zu Tragen kommen und Einfluss auf das Prädiktionsverhalten und die Genauigkeit nehmen. Dies äußert sich insbesondere in geringerer Cross-Dataset-Performanz von Modellen, die auf vollständige Gesichtsausschnitte angelernt wurden und zeugen damit von geringerer Generalisierungsfähigkeit.

Ein Lösungsansatz zur Verbesserung der Generalisierung zielt auf die Reduktion von datensatzspezifischen Bildattributen ab, bevor die eigentliche Schätzung der Blickrichtung durchgeführt wird. Dieser Ansatz wird *Domänen-Generalisierung* [239, 240] genannt und soll die relevanten Merkmale für die Blickrichtung beim Lernprozess in den Vordergrund rücken und das Auswendiglernen datensatzspezifischer Charakteristika vermeiden.

Ein weiterer Ansatz ist die *Domänen-Adaption*. Hierbei wird eine geringe Anzahl an Proben aus der Zieldomäne, annotiert oder unannotiert, im Trainingsprozess eingesetzt, um die Generalisierung für die Zielumgebung zu verbessern [241, 242, 243, 244, 245, 246, 247]. Kothari *et al*. [247] wenden eine schwach überwachte Methode an, um auf Basis von Bildern von Menschen, die einander

ansehen, die Leistung zur Generalisierung zu verbessern. Wang *et al*. [248] nutzen bayesianisches adversariales Lernen, um die Erscheinungsmerkmale in den Quell- und Zieldomänen anzugleichen. Kellnhoder *et al*. [230] optimieren das trainierte Modell mit einer Mischung aus annotierten Proben aus der Quelldomäne und nicht annotierten Proben aus der Zieldomäne. Liu *et al*. [245] verwenden ein Ensemble von Modellen für kollaboratives Lernen unter Einbezug von Ausreißern. Sie optimieren ihre Modelle zudem dadurch, dass sie die Rotations-Konsistenz-Eigenschaft bei der Blickschätzung nutzen [243]. Da sich die Ansätze zur Domänen-Adaption jedoch nur auf eine bestimmte Zieldomäne konzentrieren, kann nur bedingt von Generalisierung gesprochen werden.

Multi-Task Learning Multi-Task-Ansätze haben in zahlreichen Bereichen der Computer Vision Einzug gehalten [249, 250, 251, 252]. Hierzu zählt die Handlungserkennung (*Action Recognition*) [253] und die simultane Detektion und Segmentierung (*Detection and Segmentation*) von Objekten [254]. Im Bereich der Gesichtsanalyse wurden Multi-Task-Ansätze zur simultanen Schätzung von Kopfhaltung und Gesichtsmerkmalen [255, 189], von Geschlechts- und Gesichtserkennung [169] und sogar zur Erfassung von zusätzlichen Gesichtsattributen [256] verwendet. Insbesondere die Synergien zwischen Gesichtsattributen und Landmarkendetektion sind das Ziel umfangreicher Forschung [257, 258, 259]. Es gibt bisher jedoch nur ein Verfahren von Ghosh *et al*. [260], das Multi-Task-Learning auf die Blickrichtungsschätzung anwendet. Bei diesem wird die Kopfpose lediglich als Hilfsaufgabe eingesetzt, um die Blickrichtungsprädiktion zu verbessern.

7.2 Multi-Task-Learning für Kopfpose- und Blickrichtungsschätzung

Das Multi-Task-Learning-Paradigma (MTL) ist eine Methode, die es ermöglicht, mehrere Aufgaben mittels eines einzigen neuronalen Netzes zu lösen. Dieser Ansatz basiert auf dem Konzept einer gemeinsamen Repräsentation, das Ähnlichkeiten und Unterschiede zwischen den Aufgaben nutzt, um die Gesamtperformanz über alle Aufgaben hinweg zu verbessern. Dies ermöglicht den Einsatz eines größeren Spektrums an Daten für das Training, welches die Regularisierung während des Trainings und somit die Generalisierung des resultierenden Modells unterstützt. Weiterhin kann die Verwendung von Multi-Task-Learning das Modell in Bezug auf den Speicherverbrauch und die Rechenleistung effizienter machen. Der am häufigsten verwendete Ansatz im MTL ist das *harte* Parameter-Sharing, das eine gewisse Anzahl an Parametern bzw. Layern auf mehrere Aufgaben aufteilt. Der in diesem Kapitel vorgeschlagene Modellansatz folgt diesem Prinzip, bei dem beide Aufgaben, Blickrichtungs- und Kopfhaltungsschätzung, dieselben Verarbeitungsschichten teilen und sich erst in den finalen Layern aufspalten, um getrennt voneinander die Blickrichtung und Kopfpose zu prädizieren.

7.2.1 Fusionierung von Kopf- und Blickrichtung

Das vorgeschlagene Modell, im weiteren Verlauf als MTGH-Net (*Multi-Task Gaze and Head Pose*) bezeichnet, besteht aus einem modifizierten ResNet-50-Backbone und zwei separaten Fully Connected Layern für die Blickrichtungs- und Kopfposeschätzung. Die Standard-Architektur von ResNet-50 umfasst einen Convolution Layer, vier Residualblöcke und einen globalen Average Pooling Layer

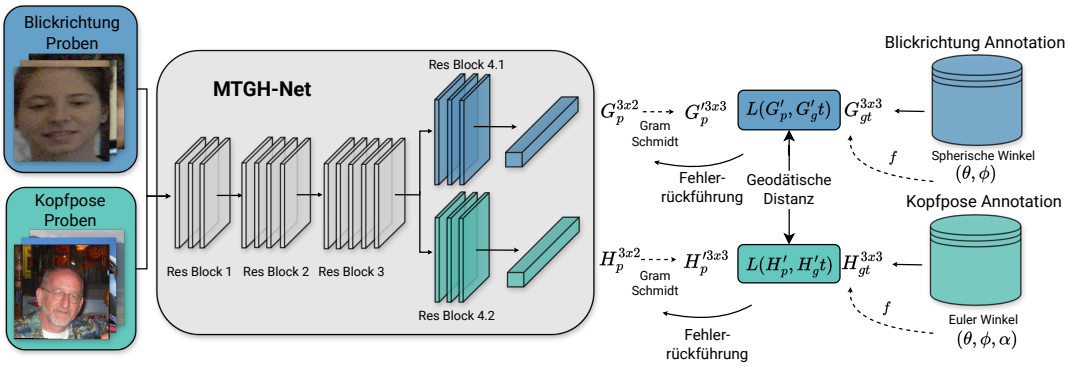


Abbildung 7.1: Gesamtübersicht des vorgeschlagenen MTGH-Modells, welches zur simultanen Bestimmung von Kopfpose und Blickrichtung trainiert wird.

(siehe Abschnitt 2.4.1). Für das MTGH-Net wird die ResNet-50-Architektur leicht angepasst, indem die ersten drei Residualblöcke als initiale Layer für beide Aufgaben eingesetzt werden, während sich beim letzten Residualblock das Netz für die Blickrichtungs- und Kopfposeschätzung aufspaltet. Schließlich werden zwei Fully Connected Layer zur Prädiktion verwendet. Die Motivation für dieses Prinzip liegt darin, in den ersten gemeinsamen Schichten die groben, übergeordneten Merkmale zu extrahieren, die für beide Aufgaben relevant sind, um im Anschluss im letzten Block die feinen, aufgabenspezifischen Merkmale zu erfassen. Die Architektur ist in Abbildung 7.1 zur Veranschaulichung dargestellt und wird im weiteren Verlauf genauer beschrieben.

7.2.2 Rotationsformalismus und Distanzfunktion

Ähnlich wie bei der Kopfpose (siehe Abschnitt 6.2) kann die Blickrichtung durch verschiedene mathematische Repräsentationen beschrieben werden. Üblicherweise werden für die Disziplin der Blickrichtungsschätzung Kugelkoordinaten (θ, ϕ) verwendet [231, 230, 232, 237], die den 3D-Blickvektor im Augenkoordinatensystem g definieren, wobei $\theta = -\arctan\frac{g_x}{g_z}$ und $\phi = -\arcsin g_y$ entspricht. Würde man beim Multi-Task-Ansatz für jede Aufgabe eine unterschiedliche Repräsentation wählen, würde dies beim Trainingsprozess zu unterschiedlichen Distanzgrößen bei der Verlustrechnung führen. Dies wiederum hätte Einfluss auf die Lernausrichtung der gemeinsamen Parameter und würde als natürlicher Gewichtungsfaktor eine Aufgabe gegenüber der anderen priorisieren.

Um ein solches Verhalten zu vermeiden, wird die vorgeschlagene 6D-Rotationsbeschreibung aus Abschnitt 6.3 übernommen und für die Kopfposeprädiktion sowie auch für die Blickrichtungsprädiktion eingesetzt. Da bei der Blickrichtungsprädiktion die Rollwinkel vernachlässigt werden, wird für die Distanzberechnung die Roll-Grundwahrheit auf null gesetzt. Parallel zum 6D-Rotationsformalismus wird auch die geodätische Distanz aus dem vorherigen Kapitel übernommen und zur Fehlerrückführung während des Trainings für Kopfpose und Blickrichtung eingesetzt.

7.2.3 Trainingsstrategie

Das Training eines Multi-Tasks-Netzes basiert üblicherweise auf einem gemeinsamen Datensatz, der für beide Aufgaben Grundwahrheiten zur Verfügung stellt. Für die Blickrichtung und die Kopfpose

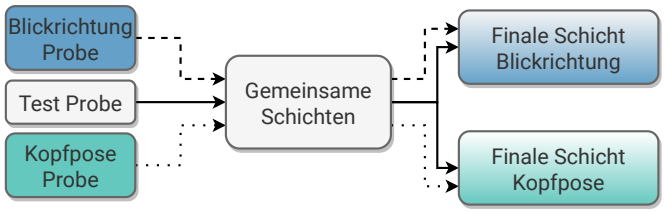
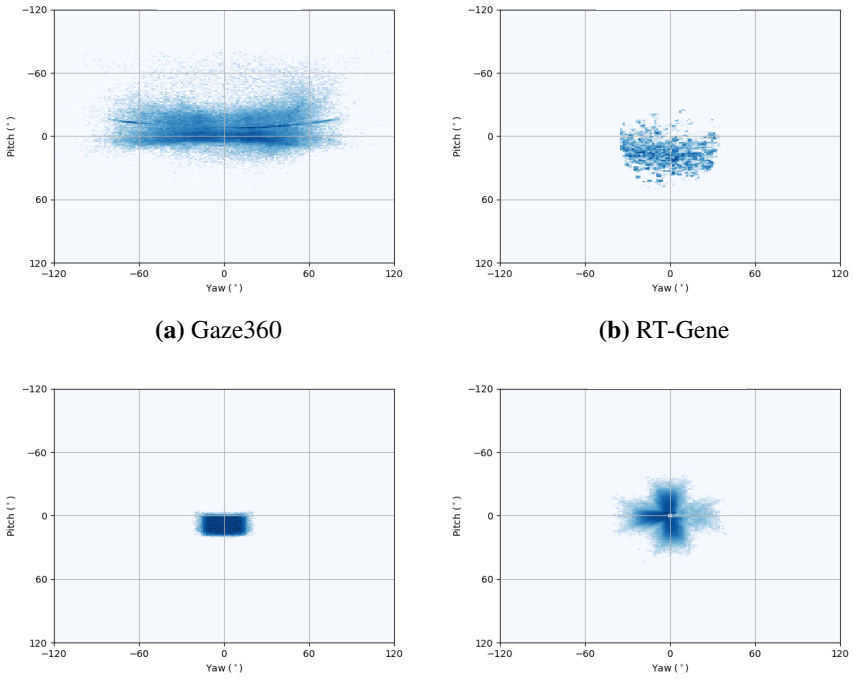


Abbildung 7.2: Verarbeitungsprinzip für Trainings- und Testproben.



(a) Gaze360 **(b)** RT-Gene

(c) MPIIFaceGaze **(d)** GazeCapture

Abbildung 7.3: Labelverteilung der Blickrichtungs-Datensätze.

gibt es jedoch keinen öffentlichen, qualitativen Datensatz, der Annotationen für beide Aufgaben bereitstellt. Daher wird eine neue Trainingsstrategie vorgeschlagen, welche einen Trainingssatz für die Kopfpose und einen Trainingssatz für die Blickrichtung in einem Training miteinander verknüpft. Hierbei werden beim Trainingsprozess abwechselnd Batches der unterschiedlichen Datensätze verarbeitet und entsprechend ihrer Grundwahrheit durch das neuronale Netz geführt. Beide Arten von Proben werden zuerst durch die gemeinsamen Layer geleitet. Kopfposeproben werden anschließend durch die dedizierten Schichten für die Kopfpose verarbeitet. Blickrichtungsproben werden analog dazu von den Schichten der Blickrichtungsprädiktion verarbeitet. Auf diese Weise werden die gemeinsamen Layer für beide Aufgaben trainiert, während die finalen Schichten nur Proben für ihre spezifische Aufgabe verarbeiten. Im Fall des Testens (*Inferenz*) des Netzes werden die Bilder wiederum durch beide Zweige geleitet, um gleichzeitig Kopfpose und Blickrichtung zu prädizieren. Dieses Prinzip wird in Abbildung 7.2 illustriert.

7.3 Datensätze

Für das Training des MTGH-Modells werden Datensätze aus dem Bereich der Blickrichtungsprädiktion und dem Bereich der Kopfposeprädiktion eingesetzt.

Blickrichtung

Zum Training und zum Testen von Modellen zur Blickrichtungsprädiktion wurde in der Vergangenheit eine Reihe von Datensätzen veröffentlicht. Aus diesen haben sich vier Datensätze als wesentliche Trainingsgrundlage hervorgetan, welche für die folgenden Experimente verwendet werden: Gaze360 [230], RT-GENE [231], GazeCapture [231] und MPIIFaceGaze [232].

Gaze360 Gaze360 enthält 84.000 Bilder von 238 Personen unterschiedlichen Alters, Geschlechts und ethnischer Zugehörigkeit. Die Datenaufnahme fand sowohl in Innen- als auch in Außenumgebungen statt. Aufgrund der starken Variationen der Kopfposen und den unterschiedlichen Distanz- und Lichtverhältnissen ist dieser Datensatz besonders herausfordernd, spiegelt aber die Bedingungen für reale Applikationsumgebungen wider.

Abbildung 7.4: Datenproben aus der Gaze360 Datenbank.

RT-Gene RT-Gene besteht aus 122.531 Proben von 15 Personen, die im Innenraum unter Einbezug großer Variationen von Blick- und Kopfhaltung aufgenommen wurden. Die Blickrichtung wurde mittels tragbarer Eye-Tracking-Brille bestimmt. Um den Einfluss der Brille auf Prädiktionsmodelle zu eliminieren, wurde sie mittels Inpainting-Verfahren nachträglich aus den Bilddaten entfernt (siehe Abbildung 7.5).

Abbildung 7.5: Datenproben aus der RT-Gene-Datenbank.

GazeCapture GazeCapture ist der größte verfügbare Datensatz für Blickrichtungserkennung. Die Daten basieren auf Aufnahmen von Bildschirmkameras von 1,450 Probanden, die zu unterschiedlichen Tageszeiten und in wechselnden Umgebungen aufgenommen wurden. Der

Trainingssatz umfasst 1,379,083 Bilder, der Testdatensatz 191,842 Bilder und der Validierungsdatensatz 63,518 Bilder.

Abbildung 7.6: Datenproben aus der GazeCapture Datenbank.

MPIIFaceGaze MPIIFaceGaze beinhaltet 213.659 Proben von 15 Personen, die während ihres täglichen Lebens über mehrere Monate aufgenommen wurden. Es enthält Bilder mit vielfältigen Hintergründen und Beleuchtungsbedingungen, was es für die ungehinderte Blickschätzung geeignet macht.

Abbildung 7.7: Datenproben aus der MPIIFaceGaze-Datenbank.

Für die Daten von Gaze360, RT-Gene und MPIIFaceGaze wird das Vorverarbeitungs-Verfahren von Cheng *et al*. [261, 262] angewendet, um aus den Rohdaten normalisierte Gesichtsausschnitte zu erzeugen. Für GazeCapture wird der Ansatz verwendet, welcher in Yu *et al*. [263, 262] vorgestellt wurde, um die Gesichtsausschnitte zu normalisieren. Die Normalisierung zielt darauf ab, die Merkmalsausprägung durch variierende Kopfposen zu reduzieren und den Fokus auf den Augenbereich zu legen.

Kopfpose

Für die Kopfpose werden die bekannten Datensätze 300W-LP, AFLW2000, BIWI eingesetzt. Diese wurden bereits im vorherigen Kapitel in Abschnitt 6.5 ausführlich vorgestellt und erfahren deshalb in diesem Abschnitt keine separate Beschreibung.

7.4 Experimente

In diesem Abschnitt wird das vorgeschlagene Modell aus Abschnitt 7.2 auf die in Abschnitt 7.3 vorgestellten Datensätze trainiert und getestet. Zuerst werden dafür die eingesetzten Evaluationsmetriken eingeführt. Es folgt eine quantitative Evaluation zur Blickrichtungsprädiktion unter Einbeziehung

Aufgabe	Abkürzung	Datensatz
Blickrichtung	D_G	Gaze360
	D_M	MPIIFaceGaze
	D_C	GazeCapture
	D_R	RT-Gene
Kopfpose	D_W	300W-LP
	D_A	AFLW2000
	D_B	BIWI

Tabelle 7.1: Übersicht der Abkürzungen der eingesetzten Datensätze.

des Stands der Technik für Standardmodelle und spezielle Modelle aus der Domänen-Adaption und der Domänen-Generalisierung. Anschließend wird analog die Kopfposeprädiktion mit dem Stand der Technik verglichen. Weiterhin wird eine Ablationsstudie durchgeführt, um den Einfluss der einzelnen Komponenten des vorgeschlagenen Ansatzes auf dessen Performanz zu untersuchen. Der Abschnitt schließt mit einer Evaluation auf qualitativer Ebene.

Evaluierungsmetriken Für die Evaluierung der Prädiktionsergebnisse wird für die Kopfposeschätzung und die Blickrichtungsschätzung jeweils eine eigene Performanzmetrik angewendet.

Für die Blickrichtung wird der mittlere Winkelfehler als Bewertungsmaßstab für die Blickschätzung eingesetzt, wobei g_p dem prädizierten Blickvektor entspricht und g_g der Grundwahrheit der Probe.

$$\text{Mittlerer Winkelfehler} = \frac{1}{N}\sum_{i=1}^{N}\frac{g_p \cdot g_g}{|g_p||g_g|} \tag{7.1}$$

Für die Kopfpose wird angelehnt an den Experimenten des vorherigen Kapitels erneut der mittlere absolute Abstand (MAE) gewählt.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}(|x_g - x_p|), \tag{7.2}$$

Implementierung Im Folgenden werden zur besseren Übersichtlichkeit Abkürzungen für die eingesetzten Datensätze eingeführt. Eine Übersicht darüber befindet sich in der Tabelle 7.1. Bei der Wahl der Aufteilung der Blickrichtungs-Datensätze wird die gängige Konvention von vorherigen Ansätzen [243, 245, 240, 239] befolgt und der Gaze360-Datensatz D_G als Trainingsdatensatz verwendet. Dieser weist die größte Bandbreite an Blickwinkeln auf (siehe Abbildung 7.3), sodass die Validierung auf den anderen Datensätzen konsistente Ergebnisse verspricht. Zum besseren Vergleich wird ein zweites Modell trainiert, welches den RT-GENE-Datensatz D_R als Trainingssatz verwendet. Für beide wird der 300W-LP-Datensatz D_W für das Training der Kopfpose eingesetzt. Die verbleibenden Datensätze für Blick- und Kopfposenschätzung, GazeCapture D_C, MPIIFaceGaze D_M, AFLW2000 D_A und BIWI D_B werden für die Validierung der Modelle verwendet.

Kategorie	Methode	$\mathcal{D}_G \rightarrow \mathcal{D}_M$	$\mathcal{D}_G \rightarrow \mathcal{D}_C$	$\mathcal{D}_R \rightarrow \mathcal{D}_M$	$\mathcal{D}_R \rightarrow \mathcal{D}_C$
Standard Modelle	Full-Face [264]	13,53°	22,23°	14,40°	18,16°
	RTGene [231]	14,52°	20,11°	10,95°	15,62 °
	CA-Net [237]	16,27°	22,11°	15,62°	18,23°
	Dilated-Net [236]	11,72°	18,9°	8,92°	14,21°
	ETH-XGaze [235]	10,3°	16,98°	12,0°	13,2°
	Gaze360 [230]	8,15°	17,8°	9,12°	13,58°
Domänen Generalisierung	PureGaze [239]	9,28°	17,22°	-	-
	SelfAtt [240]	8,31°	17,7°	-	-
Multi-Task	MTGH (***Vorg. Ansatz***)	**6,0**°	**15,50**°	**8,29**°	**11,41**°

Tabelle 7.2: Vergleich des mittleren Winkelfehlers zwischen dem vorgeschlagenen MTGH-Modell und weiteren Methoden aus dem Stand der Technik.

Der vorgeschlagene Ansatz aus Abschnitt 7.2 wird mittels PyTorch [223] implementiert. Die Gewichte der ResNet-Schichten werden auf dem ImageNet [226]-Datensatz vortrainiert. Da die Anzahl der Bilder in den Blick- und Kopfpose-Datensätzen nicht identisch ist, wird die Batch-Größe für beide Datensätze variabel angepasst, sodass beide Datensätze zum gleichen Zeitpunkt eine Epoche vollständig durchlaufen werden. Dadurch wird sichergestellt, dass alle Bilder in den Datensätzen einbezogen werden. Das Netzwerk wird für 50 Epochen mit dem Adam-Optimierer mit einer Lernrate von $1e^{-5}$ trainiert.

7.4.1 Vergleich der Blickrichtungsschätzung mit dem Stand der Technik

In einem ersten Experiment wird die Cross-Dataset-Performanz des vorgestellten MTGH-Modells mit anderen Standardmethoden für Blickrichtungsprädiktion aus dem Stand der Technik sowie mit Methoden aus dem Bereich der Domänen-Generalisierung verglichen. Die Standardmodelle umfassen Full-Face [234], RT-Gene [231], CA-Net [237], Dilated-Net [236], ETH-XGaze [235] und Gaze360 [230]. Diese Methoden fokussieren sich hauptsächlich darauf, die Prädiktionsgenauigkeit innerhalb eines Datensatzes zu verbessern. Die Ansätze zur Domänen-Generalisierung, SelfAtt [265] und PureGaze [239] hingegen versuchen, die Ergebnisse bei einer Cross-Dataset-Validierung zu optimieren, ohne dabei zusätzliche Proben aus der Ziel-Domäne zu verwenden.

Tabelle 7.2 zeigt die Ergebnisse der getesteten Modelle im Vergleich. Die Standardmodelle schneiden auf dem MPIIFaceGaze Testset mit überwiegend zweistelligem Winkelfehler auffällig am schlechtesten ab. Davon ausgenommen ist das Gaze360 Modell, welches mit einem Winkelfehler von 8.15° einen leicht geringeren Fehler aufweist als beide Modelle aus der Domänen-Generalisierung. Der in diesem Kapitel vorgeschlagene Ansatz erzielt 6.0° und schlägt damit den Stand der Technik bei den Standardmodellen um mehr als 26% und das beste Domänen-Generalisierung-Modell SelfAtt um fast 28%.

Ähnlich verhält es sich bei allen anderen Testsätzen, bei denen das MTGH-Modell den Stand der Technik sowohl bei den Standardmodellen als auch den Modellen der Domänen-Generalisierung signifikant übertrifft. Dieser erhebliche Performanzgewinn weist auf Synergieeffekte zwischen

Kategorie	Methode	Proben aus Zieldomäne	$D_G \rightarrow D_M$	$D_G \rightarrow D_C$
Domänen Adaption	GazeAdv [266]	> 100	8,19°	19,21°
	Gaze360Adv [230]	> 100	7,45°	17,12°
	DAGEN [267]	~ 500	6,61°	-
	ADDA [242]	~ 500	8,59°	-
	UMA [244]	~ 100	9,17°	-
	RUDA [243]	< 100	6,20°	-
	PNP-GA [245]	< 100	6,18°	-
Multi-Task	MTGH (***Vorg. Ansatz***)	**N/A**	**6,04°**	**15,71°**

Tabelle 7.3: Vergleich des MTGH-Ansatzes mit Methoden aus der Domänen-Adaption, welche Proben aus der Zieldomäne in ihr Training einbeziehen. Für das vorgeschlagene MTGH-Modell werden keine Proben genutzt.

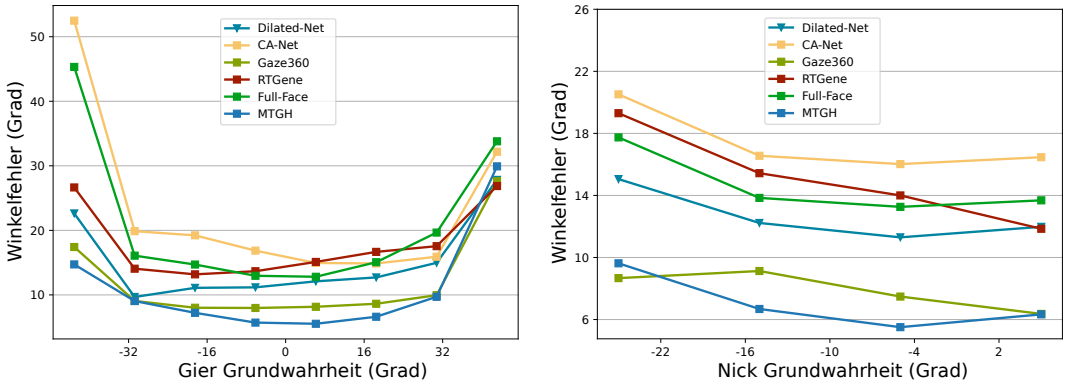
Kopfpose- und Blickrichtungsdatensatz hin, die sich in einer verbesserten Generalisierung widerspiegeln. Diese Effekte sind dabei sogar stark genug, um die Prädiktionsfehler stärker zu senken als die Modelle der Domänen-Generalisierung.

Vergleich mit Domänen-Adaptions-Modellen In einem zweiten Ansatz wird der vorgeschlagene Ansatz mit dem Stand der Technik aus der Domänen-Adaption verglichen. Hierzu zählen GazeAdv [266], Gaze360Adv [230], DAGEN [267], ADDA [242], UMA [244], RUDA [243] und PNP-GA [245]. Diese Methoden nutzen alle das ResNet18 als Backbone. Für einen fairen Vergleich wird deshalb auch das MTGH mit einem ResNet18 Backbone trainiert. Das Experiment soll darüber Aufschluss geben, ob die in Tabelle 7.2 beobachtete Generalisierung sogar solche Ansätze übertreffen kann, die für explizite Domänen trainiert wurden.

Tabelle 7.3 zeigt die Ergebnisse der Winkelfehler (Grad) des Experiments und gibt an, wie viele Proben aus den Zieldomäne (dem Testset) genutzt wurden, um die Generalisierung auf dieser Datenbasis zu verbessern. Die Modelle der Domänen-Adapation werden somit für jedes Testset neu trainiert und darauf abgestimmt, während das vorgeschlagene MTGH nur einmal trainiert wird, ohne dabei Proben aus den Testsets einzubeziehen. Dennoch zeigen die Ergebnisse, dass der vorgeschlagene Ansatz auch in diesem Experiment alle anderen Methoden aus dem Stand der Technik übertrifft.

Aus beiden Experimenten lassen sich folgende Erkenntnisse gewinnen: Die Prädiktionsergebnisse des vorgeschlagenen Multi-Task-Ansatzes mit simultaner Blickrichtung- und Kopfposeschätzung weisen auf starke Synergieeffekte hin, die die Generalisierung der Blickrichtungsschätzung verbessern und so führende Methoden aus Domänen-Generalisierung und Domänen-Adaption durch geringere Winkelfehler schlagen. Der Multi-Task-Ansatz benötigt dabei trotz Parameter-Sharing keine Erhöhung der Parameteranzahl und wird insbesondere ohne Vorverarbeitungsschritte (der Domänen-Generalisierung) und ohne Proben der Ziel-Domäne (der Domänen-Adaption) trainiert.

Fehleranalyse für Label-Intervalle Die zuvor diskutierten Ergebnisse zeigen nur durchschnittliche Werte für die gesamten Datensätze an. Um das Prädiktionsverhalten für unterschiedliche



(a) Winkelfehler im Vergleich über den Verlauf des Gier-Rotationsbereichs.

(b) Winkelfehler im Vergleich über den Verlauf des Nick-Rotationsbereichs.

Winkel analysieren zu können, wird eine weitere Evaluation durchgeführt, bei der der Winkelfehler über mehrere Labelintervalle aufgegliedert wird. Hierfür werden die Modelle verwendet, die auf dem Gaze360-Datensatz trainiert und auf dem MPII-FaceGaze-Datensatz getestet wurden. Neben dem vorgeschlagenen MTGH-Modell werden weitere Modelle aus dem Stand der Technik in den Test einbezogen, deren Implementierung veröffentlicht wurden (Full-Face [234], RT-Gene [231], CA-Net [237], Dilated-Net [236] und Gaze360 [230]).

Da der MPIIFaceGaze-Datensatz unterschiedliche Neigungs- und Gierwinkelbereiche aufweist (siehe Abbildung 7.3), wird der Nickwinkel in Intervalle von je 10° und der Gierwinkel in Intervalle von je 20° aufgeteilt. Die Ergebnisse der Gier- und Nick-Fehleranalyse sind in Abbildung 7.8a und Abbildung 7.8b dargestellt. Die Diagramme zeigen, dass das MTGH-Modell in allen Intervallen mit Ausnahme von zwei den geringsten Fehler unter den Methoden aus dem Stand der Technik aufweist. Dies zeigt, dass sich die Robustheit und Effektivität des gewählten Ansatzes zur Verbesserung der Generalisierung über den gesamten Prädiktionsbereich erstreckt.

7.4.2 Vergleich der Kopfposeprädiktion mit dem Stand der Technik

In den vorherigen Experimenten wurde die Leistung der Blickrichtungsschätzung des MTGH-Modells untersucht. Nun soll das Verhalten der Kopfposeschätzung evaluiert werden. In Tabelle 7.4 wird das MTGH in Bezug auf die Kopfposeschätzung mit anderen aktuellen Methoden aus dem Stand der Technik verglichen. Alle verglichenen Methoden verwenden die gleiche Backbone-Architektur in Form eines ResNet50 und verfolgen ausschließlich die Aufgabe zur Prädiktion der Kopfpose. Die Evaluierungskonventionen mit den AFLW2000 und BIWI Datensätzen und die getesteten Ansätze richten sich nach denen im vorherigen Kapitel verwendeten (siehe Abschnitt 6.6). Unter den verglichenen Methoden befindet sich außerdem das in Kapitel zuvor eingeführte 6DRepNet und 6DRepNet360, welche den Stand der Technik anführen und die passende Benchmark als Single-Task Ansatz zur Verfügung stellen.

Die Ergebnisse aus Tabelle 7.4 zeigen auf, dass im Gegensatz zur Blickrichtungserkennung die Kopfposeprädiktion nicht den Stand der Technik schlagen kann. Auf dem AFLW2000 Datensatz liegt

Kategorie	Methode	$D_W \rightarrow D_A$				$D_W \rightarrow D_B$			
		Gier	Nick	Roll	MAE	Gier	Nick	Roll	MAE
Single-Task	3DDFA [228]	4,71	27,1	28,4	20,1	5,50	41,1	13,2	20,2
	Dlib [181]	8,49	11,3	22,9	14,2	11,9	13,0	19,6	14,8
	HopeNet [192]	6,40	6,53	5,39	6,11	4,54	5,15	3,37	4,36
	FSA-Net [196]	4,83	6,25	4,94	5,34	4,64	5,61	3,57	4,61
	HPE [202]	4,80	6,18	4,87	5,28	3,12	5,18	4,57	4,29
	QuatNet [193]	3,97	5,62	3,92	4,50	**2,94**	5,49	4,01	4,15
	TriNet [197]	4,36	5,81	4,51	4,89	3,11	5,09	5,20	4,47
	WHENet-V [194]	4,44	5,75	4,31	4,83	3,60	4,10	2,73	3,48
	WHENet [194]	5,11	6,24	4,92	5,42	3,99	4,39	3,06	3,81
	FDN [200]	3,78	5,61	3,88	4,42	4,52	4,70	**2,56**	3,93
	Viet et al [203]	-	-	-	-	4,62	4,29	4,52	4,48
	MFDNet [198]	4,30	5,16	3,69	4,38	3,40	4,68	2,77	3,62
	DDD-Pose [201]	4,38	4,85	3,44	4,22	4,60	6,02	2,94	4,52
	Liu*et al*. [199]	**3,03**	5,06	3,68	3,93	4,12	5,61	3,15	4,29
	img2pose [229]	3,42	5,03	3,28	3,91	4,56	**3,54**	3,25	3,78
	MNN [189]	3,34	4,69	3,48	3,83	3,98	4,61	2,39	3,66
	RankPose [225]	3,26	4,72	3,23	3,74	4,54	5,61	3,05	4,40
	6DRepNet	3,27	**4,58**	**2,98**	**3,61**	3,23	5,32	2,78	3,78
	6DRepNet360	3,73	5,52	3,53	4,26	3,37	3,87	2,93	**3,39**
Multi-Task	MTGH (***Vorg. Ansatz***)	4,46	5,11	3,54	4,37	4,65	3,77	2,74	3,72

Tabelle 7.4: Vergleich des MAE der Euler-Prädiktion mit anderen Methoden aus dem Stand der Technik auf den AFLW2000- und BIWI-Datensätzen. Alle Modelle wurden auf dem 300W-LP-Datensatz trainiert.

der mittlere absolute Fehler leicht hinter 6DRepNet, kann sich aber gegenüber anderen Methoden durchsetzen. Ähnliche Ergebnisse weist die Evaluation auf dem BIWI Datensatz auf. Hier liegt der Prädiktionsfehler ca. 10% über dem Stand der Technik und liegt daher im Mittelfeld der verglichenen Methoden.

Diese quantitative Analyse deutet darauf hin, dass der vorgeschlagene Multi-Task Ansatz seine Leistung bei der Kopfpositionsabschätzung mit derselben Anzahl von Parametern im Vergleich zu einem Einzelaufgabenansatz marginal verschlechtert. Eine Verbesserung der Generalisierung wie in der Blickrichtungsprädiktion konnte nicht festgestellt werden. In Anbetracht der geteilten Parameter und somit effizienten Schätzung von Kopfpose und Blickrichtung ist der Prädiktionsfehler dennoch konkurrenzfähig.

7.4.3 Ablationsstudie

In den bisherigen Experimenten wurde der vorgeschlagene Ansatz im Bereich der Blickrichtungs- und der Kopfposeprädiktion mit dem Stand der Technik verglichen. Hierbei konnte eine signifikante Verbesserung in der Generalisierung der Blickrichtungsschätzung sowie ebenbürtige Ergebnisse bei der Kopfposeschätzung festgestellt werden. Um dieses Verhalten besser zu verstehen, wird im folgenden Abschnitt eine Ablationsstudie durchgeführt, um den Einfluss der einzelnen Komponenten auf die erzielten Ergebnisse zu analysieren. Hierfür wird zuerst ein Vergleich zwischen Multi-Task und

Methods	$\mathcal{D}_G \rightarrow \mathcal{D}_M$	$\mathcal{D}_G \rightarrow \mathcal{D}_C$	$\mathcal{D}_R \rightarrow \mathcal{D}_M$	$\mathcal{D}_R \rightarrow \mathcal{D}_C$
Single-Task (Nur Blickrichtung)	7,60°	17,95°	8,75°	12,72°
Multi-Task (MTGH) (***Vorg. Ansatz***)	**6,01°**	**15,50°**	**8,29°**	**11,41°**

Tabelle 7.5: Vergleich des MTGH als Single-Task und Multi-Task für die Blickrichtungsaufgabe.

Methods	$\mathcal{D}_G \rightarrow \mathcal{D}_M$	$\mathcal{D}_G \rightarrow \mathcal{D}_C$	$\mathcal{D}_R \rightarrow \mathcal{D}_M$	$\mathcal{D}_R \rightarrow \mathcal{D}_C$
MTGH-SL2	7,26°	15,78°	8,31°	11,90°
MTGH (***Vorg. Ansatz***)	**6,01°**	**15,50°**	**8,29°**	**11,41°**

Tabelle 7.6: Vergleich der Blickleistung zwischen MTGH-SL2 (Netzwerk mit sphärischen Winkeln und L2-Verlust) und des vorgeschlagenen MTGH.

Single-Task für die Blickrichtungsprädiktion durchgeführt, um den Einfluss des Kopfpose-Trainings auf den Winkelfehler zu messen. Anschließend wird das MTGH-Modell mit unterschiedlichen Distanzfunktionen sowie Variationen in den Convolution Blöcken und Backbones trainiert und getestet.

Vergleich von Single-Task zu Multi-Task In den ersten Experimenten in Abschnitt 7.4.1 zur Evaluierung der Cross-Dataset-Leistung der Blickrichtungsschätzung konnte eine Prädiktionsgenauigkeit des vorgeschlagenen Multi-Task-Ansatzes aufgezeigt werden, die den Stand der Technik übertrifft. Um nachzuweisen, dass die betrachtete Fehlerreduktion aus dem Multi-Task-Ansatz hervorgeht und nicht aus der zusätzlich verwendeten 6D Blickrichtungsprädiktion, wird der Multi-Task-Ansatz und der Single-Task-Ansatz in einem separaten Test gegenübergestellt. Beide Modelle sind identisch aufgebaut, jedoch wird bei einem Modell (Single-Task) die Kopfpose nicht mittrainiert, sodass ausschließlich die Blickrichtungsprädiktion ausgeführt wird. Tabelle 7.5 zeigt, dass das MTGH, welches zusätzlich zur Kopfposeprädiktion trainiert wurde, das Single-Task-Modell auf allen Testdatensätzen schlägt. Der Multi-Task-Ansatz senkt die Fehlerrate auf bis zu 20% (Testset MPIIFaceGaze). Dadurch zeigt sich, dass ausschlaggebend die Synergien zwischen Kopfpose und Blickrichtung zu einer verbesserten Generalisierung der Blickrichtung führen.

Vergleich von unterschiedlichen Distanzfunktionen In einem zweiten Experiment wird der Einfluss der Rotationsmatrix als Repräsentationsform in Kombination mit der geodätischen Distanz auf die Gesamtperformanz untersucht. Dazu wird ein zweites Modell zum Vergleich herangezogen, welches sphärische Winkel zur Repräsentation der Blickrichtung in Kombination mit der L2-Distanz einsetzt. Die entspricht dem typischen Aufbau herkömmlicher Methoden [241]. Für einen fairen Vergleich verwenden beide Netzwerke die gleiche Trainingskonfiguration. Tabelle 7.6 zeigt die Prädiktionsergebnisse beider Modelle und weist dem vorgeschlagenen Modell für jedes Testset einen geringeren Fehler auf als das verglichene Modell. Dies bestätigt die Hypothese, dass die Verwendung derselben Repräsentationen für beide Aufgaben zu einem effizienteren Training führt. Hierdurch wird verhindert, dass das Netz durch eine der Aufgaben voreingenommen oder dominiert wird. Zudem wird das Diskontinuitätsproblem vermieden, welches in Abschnitt 6.2 erläutert wurde.

Methode	$\mathcal{D}_G \rightarrow \mathcal{D}_M$	$\mathcal{D}_G \rightarrow \mathcal{D}_C$	$\mathcal{D}_R \rightarrow \mathcal{D}_M$	$\mathcal{D}_R \rightarrow \mathcal{D}_C$
Geteilter ResNet-Block	6,86°	16,62°	9,25°	11,50°
Getrennter ResNet-Block (***Vorg. Ansatz***)	**6,01°**	**15,50°**	**8,29°**	**11,41°**

Tabelle 7.7: Vergleich zwischen getrenntem und geteiltem ResNet-Block.

Vergleich von getrennten Convolution Blöcken In einem dritten Experiment wird die Bedeutung des letzten ResNet-Blocks der vorgeschlagenen Multi-Task-Netzwerkarchitektur für die Gesamtleistung der Blickrichtungsprädiktion evaluiert. Dieser wird für beide Aufgaben getrennt, um für jede Aufgabe die Generierung von unterschiedlichen High-Level-Merkmalen zu ermöglichen und folglich die aufgabenspezifische Leistung zu verbessern. Zur Evaluation dieser Hypothese wurde ein neues Modell trainiert, bei dem auch dieser ResNet-Block für beide Aufgaben geteilt wird. Tabelle 7.7 zeigt den Vergleich beider Modelle und verdeutlicht, dass die Trennung des Blocks zur Verbesserung der Gesamtleistung führt. Hierbei muss bei der Interpretation jedoch auch berücksichtigt werden, dass durch die Trennung des Blocks auch die Gesamtzahl an trainierbaren Parametern steigt.

Vergleich von unterschiedlichen Backbones Im letzten Experiment wird die Auswirkung des Austauschs vom ResNet-50 durch ein kleineres ResNet-18 [230] untersucht. Dies analysiert, ob die Anzahl an Parametern bei gleichbleibender Genauigkeit reduziert werden kann. Die in Tabelle 7.8 dargestellten Ergebnisse zeigen, dass der vorgeschlagene Ansatz die Performanz auch mit dem kleineren ResNet-18 beibehält, jedoch mit einem leichten Rückgang von etwa 1% im Vergleich zu ResNet-50. Somit kann für reale Anwendungen auch die Nutzung von kleineren Architekturen in Betracht gezogen werden, bei denen die Effizienz des Modells eine primäre Rolle spielt.

Methode	$\mathcal{D}_G \rightarrow \mathcal{D}_M$	$\mathcal{D}_G \rightarrow \mathcal{D}_C$	$\mathcal{D}_R \rightarrow \mathcal{D}_M$	$\mathcal{D}_R \rightarrow \mathcal{D}_C$
ResNet-18	6,04°	15,71°	8,54°	11,45°
ResNet-50	6°	15,50°	8,29°	11,41°

Tabelle 7.8: Vergleich der Blickrichtungsprädiktion zwischen ResNet-18 und ResNet-50.

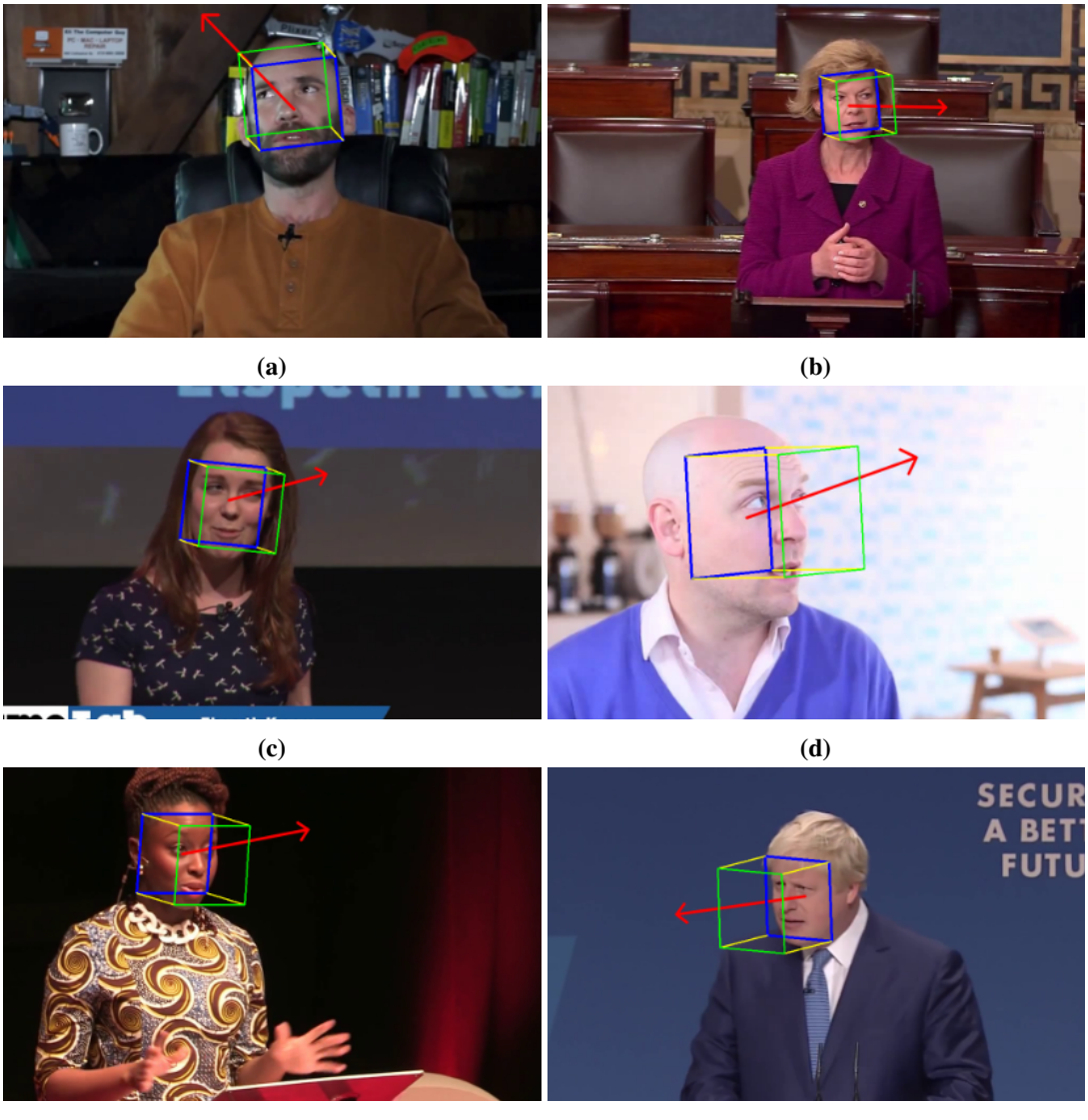
7.4.4 Qualitative Ergebnisse

Zur qualitativen Bewertung wurden einige exemplarische Bilder aus den Videos des 300VW-Datensatzes [268] entnommen und durch den MTGH-Ansatz verarbeitet. Da die Aufnahmen nicht ausschließlich die Köpfe der Personen darstellen, sondern diese überwiegend nur einen kleinen Teil der Bildebene einnehmen, werden in einem vorgelagerten Schritt die Gesichter mittels eines Gesichtsdetektors lokalisiert und ausgeschnitten. Für die Ausschnitte wird im nächsten Schritt durch das MTGH-Verfahren Blickrichtung und Kopfpose simultan geschätzt. Das Ergebnis zeigt Abbildung 7.9. Hierbei wird die Kopfpose mittels eines Kubus visualisiert, dessen Rotation der Ausrichtung des Kopfes entspricht. Für die Darstellung der Blickrichtung wird ein Pfeil verwendet.

Die Beispielbilder zeigen variierende Hintergründe und Beleuchtungsverhältnisse sowie wechselnde Kamerawinkel. Sowohl Kopfpose als auch Blickrichtung können robust erfasst werden. Um die

Wechselwirkung zwischen beiden Disziplinen besser zu verstehen, sind insbesondere jene Fälle von Bedeutung, in denen Kopfpose und Blickrichtung stark voneinander abweichen. Da bei natürlichem Verhalten beide Ausrichtungen starke Korrelationen aufweisen, besteht die Gefahr, dass sich dies in Form eines Bias auf das MTGH-Modell überträgt. Dadurch würde die Prädiktionsgenauigkeit in Beispielen, in denen Kopfpose und Blickrichtung divergieren, sinken.

Abbildung 7.9a, 7.9b und 7.9d zeigt Aufnahmen, in denen diese Divergenz stark ausgeprägt ist. Die Prädiktionen zeigen, dass der vorgestellte Ansatz dennoch in der Lage ist, dieses Verhalten zu erkennen und beide Ausrichtungen eigenständig zu prädizieren.



(a) **(b)**

(c) **(d)**

(e) **(f)**

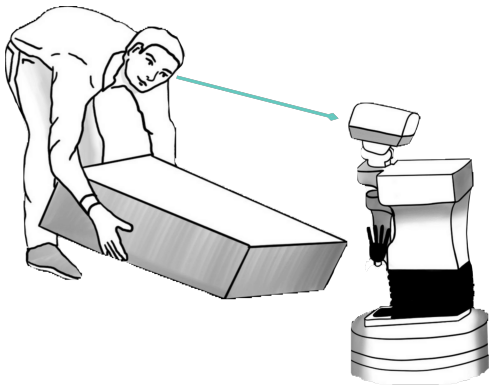
Abbildung 7.9: Simultane Blick- und Kopfposeschätzung auf exemplarischen Bildern aus dem 300VW [268]-Datensatz. Der rote Pfeil repräsentiert die Blickrichtung, der Kubus die Kopfhaltung.

7.5 Diskussion

In diesem Kapitel wurde ein neuer Trainings-Ansatz vorgestellt, der sich die starke Korrelation zwischen Blickrichtung und Kopfpose mittels eines Multi-Task-Modells zunutze macht. Dabei wird eine Trainingsstrategie vorgeschlagen, die zwei Datensätze mit unterschiedlichen Annotationen (Blickrichtung und Kopfpose) verarbeiten kann. Auf diese Weise wird der Einfluss von Variationen des Aussehens und der Kopfhaltung reduziert und die Generalisierung des Modells verbessert.

Die Experimente weisen eine Verbesserung der Blickschätzungsleistung in einem Bereich von $6-21\%$ im Vergleich zu einem Single-Task-Modell auf, das in vier Experimenten evaluiert wurde. Es konnte jedoch keine signifikante Verbesserung der Kopfposeprädiktion festgestellt werden. Um einen fairen Vergleich mit früheren Blickschätzungsmethoden zu ermöglichen, wurden die Blickrichtungs-Datensätze durch eine Normalisierung der Bilder vorverarbeitet, welche in diesem Bereich eine häufig verwendete Technik ist [266, 230, 235, 231, 236, 237]. Die Normalisierung wurde angewandt, um die Auswirkungen von Variationen der Kopfhaltung auf die Bilder zu eliminieren, um so die Blickrichtungsschätzung bei Einzelaufgaben zu verbessern. Dies führte jedoch auch zu einem geringeren Nutzen für das Training der Kopfhaltung, da die Variationen in der Kopfhaltung im Datensatz eliminiert wurden.

KAPITEL 8



Blickkontaktschätzung aus der Egoperspektive

Blickkontakt spielt in alltäglichen sozialen Interaktionen eine wichtige Rolle und ist einer der relevantesten Mechanismen in der nonverbalen Kommunikation. Er dient als Signal zur Initiative für Kommunikation [269], zur Regulierung von Interaktionen (z. B. Aufbau und Aufrechterhaltung gemeinsamer Aufmerksamkeit [270, 271, 272]) und zur Verdeutlichung von Kommunikationszielen. Die psychologischen Mechanismen des Augenkontakts zwischen Menschen werden auch in Mensch-Roboter-Interaktionsszenarien beobachtet. Wenn Personen Blickkontakt mit einem humanoiden Roboter aufnehmen, löst dieser ähnliche Arten von instinktiven, affektiven und aufmerksamkeitsbezogenen Reaktionen aus, wie sie bei Blickkontakt mit einem anderen Menschen auftreten [273, 274, 275]. Überdies wirkt sich der Blickkontakt mit einem Roboter positiv auf dessen Sympathiewert und die Zuschreibung von menschlichen Attributen aus [276] und kann sogar die Ehrlichkeit eines Menschen beeinflussen [277].

Menschen verfügen über eine bemerkenswerte Fähigkeit, Augenkontakt selbst unter schwierigen Bedingungen präzise wahrzunehmen. Die robuste Erkennung von Blickkontakt bei Maschinen, insbesondere im Bereich der Mensch-Roboter-Interaktion, ist jedoch weitgehend unerforscht und stellt bisher eine unangetastete Herausforderung dar. Einer der Hauptgründe hierfür ist der Mangel an umfassenden und qualitativ hochwertigen Datensätzen, mit denen neuronale Netze effektiv trainiert werden können, um Blickkontakte in unkontrollierter Umgebung robust zu erfassen.

Im Rahmen dieses Kapitels wird sich diese Herausforderung angenommen und ein umfassender Datensatz mit dem Namen **NITEC** (**N**euro-Information **T**echnology Group **E**ye **C**ontact) vorgestellt. Zunächst werden dafür relevante Arbeiten aus dem aktuellen Stand der Technik diskutiert

(Abschnitt 8.1). Im Anschluss wird die Zusammensetzung der vorgestellten NITEC Datenbank diskutiert, die Annotationspipeline besprochen und ein Vergleich mit anderen existierenden Datenbanken zur Blickkontakterkennung gezogen (Abschnitt 8.2). Im nächsten Abschnitt werden einige Baselinemodelle auf der NITEC Datenbank und anderen öffentlich zugänglichen Datenbanken trainiert und eine quantitative und qualitative Evaluation des Datensatzes durchgeführt (Abschnitt 8.3). In einem abschließenden Experiment wird das NITEC Modell in einer Probandenstudie zur Mensch-Roboter-Interaktion getestet, um Eindrücke aus der realen Interaktionen zu gewinnen. Das Kapitel schließt mit einer zusammenfassenden Diskussion (Abschnitt 8.4).

Forschungsbeitrag

» Es wird ein neuer, einzigartiger Datensatz (NITEC) für Blickkontakt aus der Egoperspektive vorgestellt, der 36.000 Hand-annotierte Labels enthält. NITEC ist der erste öffentliche Datensatz in dieser Dimension und Qualität, um zuverlässig für das Trainieren von End-to-End Modellen zur Blickkontakterkennung eingesetzt zu werden.

» Der vorgestellte Datensatz wird in zahlreichen quantitativen Experimenten evaluiert. Diese demonstrieren, dass auf NITEC trainierte Klassifikationsmodelle die genauesten und robustesten Klassifikationsergebnisse erzielen.

» Es werden zusätzliche Ablationsstudien durchgeführt, um wertvolle Informationen über das Prädiktionsverhalten der trainierten Modelle zu erhalten.

» Es wird eine Probandenstudie durchgeführt, die die Effektivität von NITEC bestätigt und die Relevanz von Blickkontakt für Mensch-Roboter-Interaktionen untermauert.

8.1 Verwandte Arbeiten

Es gibt zahlreiche Forschungsansätze, die sich mit dem Blickkontakt zwischen Mensch und Roboter mithilfe spezieller Interaktionssysteme befassen [278, 279, 280, 276, 281, 282, 283]. Die meisten dieser Systeme konzentrieren sich auf die Erzeugung realistischen Roboterverhaltens, während der Blick des Menschen von hardwarebasierten Eye-Trackern nachverfolgt wird [284, 285]. Diese Ansätze sind für reale Szenarien jedoch nicht geeignet. Andere bildbasierte Methoden konzentrieren sich hauptsächlich auf die Prädiktion des Blickvektors oder die Schätzung der Kopfhaltung, um den aktuellen Fokus der Aufmerksamkeit zu lokalisieren. Hierbei wird der Blickkontakt als Teilaufgabe formuliert, indem dieser als spezifischer Blickwinkel definiert wird (z. B. $\pm 5°$). Inwiefern dieser Ansatz tatsächlich zuverlässige Ergebnisse liefert, wird im Abschnitt zu den Experimenten (siehe Abschnitt 8.3) genauer untersucht.

Der Blickkontakt als Klassifizierungsaufgabe hat in letzter Zeit insbesondere im Automobilbereich großes Interesse geweckt, um die Aufmerksamkeit und das Situationsbewusstsein von Fußgängern in Verkehrssituationen einzuschätzen. Onkhar *et al.* [286] haben eine Methode vorgestellt, bei der ein

am Kopf getragener Eye-Tracker für den Fußgänger und eine Stereokamera im Fahrzeug verwendet werden. Mordan *et al*. [287] stellte ein End-to-End-Multitask-CNN für die Analyse von Fußgängern vor, einschließlich Blickkontakt, basierend auf dem *JAAD* Datensatz [288]. Ein weiterer Datensatz namens *LOOK* für die Erkennung von Blickkontakten bei Fußgängern wurde von Belkada *et al*. [289] vorgestellt, die zusätzlich einen Körperpose-basierenden Klassifizierungsansatz vorschlagen. Dies ist darauf zurückzuführen, dass in der Automobilbranche die meisten Fußgänger aus großer Distanz erfasst und analysiert werden, wobei deren Gesichter allein nicht genügend aussagekräftige Merkmale liefern. Smith *et al*. [290] haben eine der ersten Arbeiten zur Klassifizierung von Blickkontakt im Nah- bis Mittelfeld auf der Grundlage eines speziell erstellten Blickdatensatzes vorgestellt, welcher jedoch nicht der Öffentlichkeit zugänglich gemacht wurde. Ye *et al*. [291] stellen eine weitere lernbasierte Methode vor, welche die erfasste Kopfpose mit einem temporalen Conditional Random Field koppelt. Inwiefern allein die Kopfpose ausreichende Informationen über die Erfassung von Blickkontakt liefert, wird jedoch nicht überprüft.

Chong *et al*. [292] haben einen bildbasierten Datensatz mit mehr als 4.000.000 Proben aufgebaut und vorgestellt, um die wichtigsten Blickmuster für die Diagnose von Autismus-Spektrum-Störungen zu identifizieren. Jedoch ist auch ihr Datensatz nicht öffentlich zugänglich. Einzig Zhang *et al*. [293] und Mitsuzumi *et al*. [294] haben zusammen mit ihren Modellvorschlägen Ego-Vision-basierte neue Datensätze mit wenigen tausend Proben veröffentlicht, welche im Verlauf dieses Kapitels näher betrachtet werden.

Es kann zusammengefasst werden, dass die Erfassung von Blickkontakt für nahe und mittlere Distanzen im Bereich der Mensch-Roboter-Interaktionen bisher nur vereinzelt von Forschungsvorhaben betrachtet wurde. Die wenigen vielversprechenden Datensätze wurden der Öffentlichkeit nicht zugänglich gemacht, weshalb sich ein hoher Bedarf an qualitativen Daten aufzeigt.

8.2 Generierung der NITEC Datenbank

In diesem Abschnitt wird ein detaillierter Einblick in das Generierungsverfahren und die Struktur des NITEC Datensatzes gegeben. Hierzu wird mit einer kurzen Analyse bestehender Datensätze begonnen, gefolgt von Details zur Erstellung von NITEC und seiner Annotationspipeline. Abschließend wird ein Vergleich des endgültigen NITEC mit anderen veröffentlichten Datensätzen vorgenommen.

Bis heute gibt es nur zwei öffentlich verfügbare Datensätze, die über Annotationen für Blickkontakt für Nah- und Mitteldistanzen verfügen: DEEPEC [294] und OFDIW [293].

DEEPEC DEEPEC liefert manuell annotierte Bilder aus den Datensätzen LFPW [216], Helen [218], AFW [295] und IBUG [296], die insgesamt 4.150 Proben umfassen. Der Datensatz ist aufgeteilt in 53% Proben mit Blickkontakt/Augenkontakt und 47% Proben mit abgewandtem Blick. Die Quelldatensätze wurden ursprünglich für die Gesichtsanalyse verwendet und liefern hochauflösende, meist nicht verdeckte Gesichter in wechselhaften Umgebungen.

OFDW OFDIW ist aufgeteilt in einen Blickkontaktdatensatz für Menschen und einen Blickkontaktdatensatz für Tiere. Der menschliche Datensatz besteht aus 16.548 Proben mit Bildern aus dem LFW-Datensatz [297], der ursprünglich für Aufgaben der Gesichtserkennung veröffentlicht wurde.

Ein dritter – nicht öffentlich verfügbarer – Datensatz wurde von Chong *et al.* [292] vorgestellt. Dieser wurde als Teil einer Studie mit menschlichen Probanden aufgenommen und beinhaltet 4.339.879 annotierte Bilder (281.152 mit Blickkontakt) für das Training und 353.924 annotierte Bilder (25.112 mit Blickkontakt) für die Validierung.

8.2.1 Datensatz Komposition

Das Ziel für den Aufbau der Datenbank ist es, einen umfassenden Datensatz für die Klassifizierung von Blickkontakt aus der Egoperspektive zu erstellen, der durch eine hohe Anzahl an Proben, Variabilität und qualitativer Annotationen besticht und damit die Grundlage für robuste und vielfältig einsetzbare Modelle bietet. Als Datenbasis werden öffentlich verfügbare Datenproben aus vier verschiedenen Datensätzen mit sich ergänzenden Merkmalen vorgeschlagen: WIDER FACE [298], Gaze360 [241], CelebA [299] und Helen [218].

WIDER FACE ist ein großangelegter "in-the-wild"-Datensatz, der in erster Linie für Aufgaben der Gesichtsdetektion erstellt wurde. Er enthält Bilder mit verschiedenen Szenenkontexten, z. B. »Picknick«, »Parade«, und »Sportevent«. Die überwiegende Anzahl an Bildern zeigt mehrere Personen aus größerer Distanz, sodass die geringe Bildauflösung ihrer Gesichter die Klassifizierung von Blickkontakt besonders herausfordernd macht.

Abbildung 8.1: Datenproben aus der WIDER FACE Datenbank.

Gaze360 ist ein Datensatz zur Blickrichtungsschätzung und wurde bereits im vorherigen Kapitel in Abschnitt 7.3 vorgestellt. Er wurde mithilfe von 238 Probanden in Innen- und Außenbereichen erzeugt und umfasst 172.000 Proben mit einer Vielzahl von Blickrichtungen in Kombination mit einer großen Bandbreite von Kopfhaltungen.

CelebA ist ein großer Datensatz, der für die Erfassung von Gesichtsattributen erstellt wurde. Er umfasst hauptsächlich Bilder von Prominenten während ihrer Teilnahme an Veranstaltungen. Die verfügbaren Bilder bieten deshalb gut ausgeleuchtete Gesichter mit vielen Merkmalen und anspruchsvollen Blickrichtungen (z. B. leicht neben der Kamera). Durch das gleichbleibende Setting variieren die Kameraperspektiven jedoch nur gering.

Abbildung 8.2: Datenproben aus der Gaze360 Datenbank.

Abbildung 8.3: Datenproben aus der CelebA Datenbank.

Helen ist ein weiterer Datensatz, der auch auf die Analyse von Gesichtsmerkmalen abzielt. Im Gegensatz zum CelebA Datensatz basieren die Bilder jedoch auf Internetfunden und bieten deshalb eine große Bandbreite an Bildauflösungen, Kameraperspektiven, Verdeckungen, Belichtungsverhältnissen und Expressionen.

Abbildung 8.4: Datenproben aus der Helen Datenbank.

8.2.2 Annotationspipeline

Um einen möglichst variantenreichen Datensatz zu erzeugen, werden Datenproben aus allen vier Datensätzen, WIDER FACE, Gaze360, CelebA und Helen, in einem Set vereint. Die Annotation bezieht sich dabei auf das Binärproblem Blickkontakt und Kein-Blickkontakt. Ersteres entspricht dabei dem direkten Blick in die Kameralinse. Mit Ausnahme von Gaze360 werden alle Proben manuell mit einem eigens dafür programmierten Annotations-Tool gelabelt, das mittels eines Detektors [300] Gesichts-Bounding-Boxen vorschlägt und anschließend zusammen mit der Annotation abspeichert. Auf diese Weise könnten die Annotatoren den Kontext außerhalb der Gesichtsregion in ihren Entscheidungsprozess einbeziehen. Bei der Beurteilung der Proben wird das subjektive Empfinden in den Vordergrund gestellt, um die menschliche Intuition in die Labels einfließen zu lassen. Hierbei gibt es auch die Möglichkeit, Proben aus dem Datensatz auszuschließen, sofern keine eindeutige Entscheidung über die Klassifikation getroffen werden kann. Insgesamt ergaben sich aus

(Teil-)Datensatz	Anzahl der Proben	Konflikte	Blickkontakt in Konflikten
NITEC-WIDER FACE	2.829	15,6 %	24,7 %
NITEC-CelebA	2.430	17,4 %	92,0 %
NITEC-Helen	525	8,2 %	88,4 %
NITEC	5.784	15,7 %	59,1 %

Tabelle 8.1: Auswertung des annotierten NITEC Testdatensatzes.

dieser Prozedur 35.919 Proben, von denen 13.829 aus WIDER FACE, 7.214 aus Gaze360, 12.226 aus CelebA und 2.650 aus Helen stammen.

Der Datensatz wurde in 29.003 Trainingsbilder und 6.916 Testbilder aufgeteilt, was einem Aufteilungsverhältnis von etwa 80/20 entspricht. Während beim Trainingsdatensatz jede Probe nur einfach gelabelt wurde, wurde jede Probe des Testdatensatzes von drei verschiedenen Annotatoren beurteilt. Das endgültige Label wurde anschließend durch eine Mehrheitsentscheidung bestimmt. Tabelle 8.1 gibt einen Überblick über die Anzahl von Annotationen des Testdatensatzes und die Art der Konflikte in den einzelnen Teilmengen. Sie zeigt eine Konfliktrate von etwa 15% für WIDER FACE und CelebA, und etwa 8% für Helen. Interessanterweise entschied sich in den beiden letztgenannten Fällen die Mehrheit in 90% für eine Probe mit Augenkontakt, während bei WIDER FACE die meisten Konflikte ohne Augenkontakt ausgewählt wurden.

Im Gegensatz zu den anderen Datensätzen verfügt der Gaze360 Datensatz über 3D-Blickvektor-Grundwahrheiten. Angesichts dessen wurde für diese Teilmenge an Daten statt einer manuellen Annotation eine automatische vorgezogen. Hierfür wird die 3D-Blickrichtung in einen 2D-Vektor umgewandelt, der aus einem Gierwinkel, einem Nickwinkel und einem Einheitsvektor für die Länge besteht. Im Anschluss konnten Proben aus dem Trainings- und Testdatensatz gesammelt werden, bei denen Gier- und Nickwinkel innerhalb des Bereichs $\pm$ 5° liegen. Dieser Bereich entspricht etwa dem Fokus des Blickes auf das Zentrum der Kamera. Daraufhin wurde nach dem Zufallsprinzip die gleiche Anzahl von Proben mit einer Blickrichtung außerhalb des Zielbereichs ausgewählt, um Proben ohne Blickkontakt zu erzeugen.

8.2.3 Vergleich zu anderen öffentlichen Datensätzen

Tabelle 8.2 und Abbildung 8.5 zeigen einen Vergleich des vorgestellten NITEC Datensatzes mit den beiden anderen öffentlichen Datensätzen. Mit rund 36.000 Proben ist der NITEC mehr als doppelt so groß wie OFDIW, der 16.648 Proben enthält. Der dritte Datensatz, DEEPEC, umfasst nur 4.150 Proben und ist damit der kleinste Datensatz.

Mit einem Label-Split von 47,5% für Blickkontakt und 52,5% für Nicht-Blickkontakt ist DEEPEC jedoch der ausgewogenste Kandidat, gefolgt von NITEC mit einem Anteil von 40,5% Augenkontakt. Der leichte Überhang an Nicht-Augenkontakt-Proben wird vor allem durch die Proben aus dem

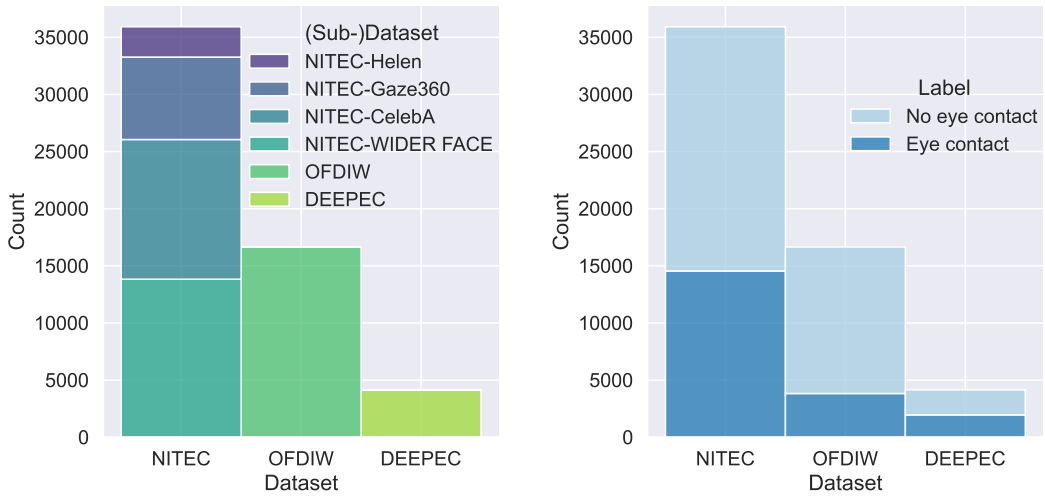


Abbildung 8.5: Vergleich des NITEC Datensatzes mit zwei anderen öffentlichen Datensätzen in Bezug auf Größe und Label-Verteilung. © 2024 IEEE

Datensatz	Anzahl der Proben (Blickkontakt [%])		
	Train	Test	$\sum$
OFDIW	11.511 [22,7]	4.137 [23,93]	16.648 [23,0]
DEEPEC	4.150 [47,5]		4.150 [47,5]
NITEC-WIDER FACE	11.000 [23,2]	2.829 [20,0]	13.829 [22,6]
NITEC-CelebA	9.829 [49,5]	2.397 [63,4]	12.226 [52,2]
NITEC-Helen	2.125 [52,0]	525 [57,1]	2.650 [53,0]
NITEC-Gaze360	6.049 [50,5]	1.165 [50,6]	7.214 [50,5]
NITEC	29.003 [39,9]	6.916 [43,0]	**35.919** [40,5]

Tabelle 8.2: Vergleich des NITEC Datensatzes mit anderen öffentlichen Datensätzen für Blickkontakt aus der Egoperspektive.

WIDER FACE-Set hervorgerufen, bei dem nur etwa jede fünfte Stichprobe mit Blickkontakt annotiert ist. Dies liegt an der Beschaffenheit des WIDER FACE Datensatzes. Dieser enthält hauptsächlich Gesichter, die aus großer Entfernung aufgenommen wurden und bei denen die aufgenommenen Personen sich der Kamera nicht bewusst waren. Diese Daten wurden jedoch bewusst mit in den NITEC Datensatz einbezogen, um falsch-positive Ergebnisse in den Zielmodellen für solche Fälle zu reduzieren, in denen die Zielgesichter nur wenige Merkmale aufweisen. Die übrigen NITEC Teildatensätze sind mit etwa 50% ausgewogen. Der OFDIW Datensatz weist hingegen eine ähnliche Label-Verteilung auf wie der WIDER FACE-Subdatensatz mit unausgeglichenen 23% Blickkontaktquote.

8.3 Experimente

Im folgenden Abschnitt werden mehrere Experimente durchgeführt, um die Leistung und Qualität des vorgestellten NITEC Datensatzes zu analysieren und mit anderen veröffentlichten Datensätzen zu vergleichen. Zuerst wird hierfür eine Cross-Dataset-Evaluation unter Einbezug der DEEPEC und OFDIW Datensätze vorgenommen (Abschnitt 8.3.1). Im Anschluss werden weitere Methoden aus dem Stand der Technik zum Vergleich herangezogen (Abschnitt 8.3.1), unter anderem mit Ansätzen aus dem Bereich der Blickrichtungs- und Kopfposeprädiktion, um die Fähigkeit zur Blickkontaktprädiktion zu analysieren. Schließlich werden Datensatz-interne Experimente durchgeführt, die weiteren Aufschluss über die Auswirkungen der einzelnen Komponenten des NITEC Datensatzes auf die Gesamtleistung geben (Abschnitt 8.3.1). Abschließend werden einige qualitative Analysen durchgeführt, die das Prädiktionsverhalten genauer untersuchen (Abschnitt 8.3.2).

Evaluationsmetriken Für die quantitative Evaluation dienen unterschiedliche Metriken als Grundlage, welche im Folgenden näher beschrieben werden.

Bei einem Klassifikationsproblem mit der Hauptklasse »*Blickkontakt*« können bei einer Modellprädiktion vier unterschiedliche Fälle auftreten. Im ersten Fall wird bei gegebener Grundwahreit von »*Blickkontakt*« die korrekte Prädiktion durchgeführt. Dies entspricht einem True Positive (TP). Wird hingegen die Grundwahrheit »*Kein-Blickkontakt*« korrekt prädiziert, wird dies als True Negative (TN) bezeichnet. Würde in diesem Fall die Prädiktion »*Blickkontakt*« heißen, bezeichnet man dies als False Negative (FN). Umgekehrt entspräche es einem False Positive (FP).

Aus diesen vier Fällen lassen sich unterschiedliche Grundmetriken definieren. Die Erste ist die Genauigkeit (engl. *Precision*), die besagt, mit welcher Wahrscheinlichkeit die Hauptklasse korrekt klassifiziert wurde:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{8.1}$$

Eine weitere Metrik ist die Sensitivität (engl. *Recall*). Diese zeigt auf, mit welcher Wahrscheinlichkeit eine prädizierte Hauptklasse tatsächlich der Hauptklasse entspricht:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{8.2}$$

Aus Precision und Recall lässt sich nun die F1-Metrik generieren, welche dem harmonischen Mittel beider Faktoren entspricht:

$$\text{F1} = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8.3}$$

Als weitere populäre Evaluationsmetrik gilt die Treffergenauigkeit (engl. *Accuracy*). Sie zeigt auf, wie hoch die Wahrscheinlichkeit ist, dass das Modell eine korrekte Prädiktion erzeugt.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{8.4}$$

Der AP-Wert (*Average Precision*) fasst eine Precision-Recall-Kurve als gewichtetes Mittel der Präzision über alle Schwellwerte zusammen, wobei die Differenz in der Sensitivität (Recall) vom vorherigen Schwellenwert als Gewicht dient.

$$\text{Average Precision (AP)} = \int_{r=0}^{1} p(r)dr \tag{8.5}$$

Implementierung Für die zuverlässige Evaluation der Klassifizierung zwischen Blickkontakt und Nicht-Blickkontakt werden vergleichbare Baseline-Modelle trainiert, die bereits für unterschiedliche Applikationen im Bereich der Computer Vision zum Einsatz kommen. Aus dem Bereich der CNNs wurde das ResNet [301] (Modellgröße ResNet18 und ResNet50) und aus dem Bereich der Vision Transformer der SWIN-Transformer [21] (Modelgröße Tiny, Small und Base) als Backbone eingesetzt, welche jeweils mit einem finalen Layer mit zwei Ausgangsneuronen ausgestattet werden. Die Klassifikation wird über einen finalen Soft-Max-Layer herbeigeführt. Die Eingangsdaten bestehen aus den ausgeschnittenen Gesichtsbildern. Die Augmentierung der Bilder beschränkt sich dabei auf zufälliges Cropping und zufälliges horizontales Spiegeln. Ersteres soll beim Training die Suche nach den markanten Merkmalsbereichen (vorwiegend die Augenpartie) verbessern.

Alle Modelle werden für 20 Epochen unter Verwendung der binären Cross-Entropie mit der Adam-Optimierungsfunktion [302] trainiert. Hierbei wird für die ResNet-Architektur eine Start-Lernrate von $0,0001$, und für die Swin-Transformer von 0,001 angesetzt, mit einer Batchgröße von 80. Die Augmentierung wird auf randomisiertes Cropping und horizontales Spiegeln beschränkt.

8.3.1 Quantitative Evaluation

Beim Vergleich der verfügbaren Datensätze, die speziell für die Erkennung von Blickkontakt entwickelt wurden (OFDIW, DEEPEC und der neu vorgestellte NITEC Datensatz), wurden die Modelle auf den jeweiligen Trainingsdatensätzen jedes Datensatzes trainiert und ihre Ergebnisse auf allen Testdatensätzen der in Tabelle 8.3 dargestellten Datensätze und Sub-Datensätze verglichen.

Die Aufteilung von Trainings- und Testdatensatz ist für den NITEC und den OFDIW Datensatz fest definiert. Für den DEEPEC Datensatz hingegen ist keine Aufteilung festgelegt, deshalb werden die Daten nach dem Zufallsprinzip in einem Verhältnis von 80/20 für Training und Testen aufgeteilt.

Für den Vergleich wird die Teststrategie von Belkada *et al.* [289] herangezogen und die Average Precision als Hauptmetrik angeführt. Diese wird ergänzt durch den F1-Score, um einen Einblick in die Fähigkeit des Klassifikationsmodells zu geben, positive Instanzen zuverlässig zu klassifizieren und gleichzeitig False Positives und False Negatives zu minimieren. Die Untersuchung der Ergebnisse von ResNet18 zeigt eine klare Performanzdifferenz zwischen Modellen, die auf unterschiedlichen Trainingsdatensätzen trainiert wurden. DEEPEC schneidet bei allen Testdatensätzen durchweg am schlechtesten ab und zeigt signifikante Unterschiede im Vergleich zu den anderen Modellen

Datensatz		Blickkontakt Klassifikation (AP) ↑ [F1-Score ↑]						
		OFDIW	DEEPEC	NITEC-WF	NITEC-Gaze360	NITEC-CelebA	NITEC-Helen	NITEC
OFDIW	RS18	57,4 [33,1]	70,8 [61,2]	44,3 [37,2]	76,3 [19,9]	91,6 [76,7]	92,1 [75,4]	80,4 [61,2]
DEEPEC	RS18	31,2 [16,3]	69,6 [62,7]	27,4 [23,9]	57,8 [27,7]	80,4 [42,1]	88,6 [74,5]	62,0 [39,9]
NITEC *(Vorg. Ansatz)*	RS18	**59,5 [55,3]**	**72,4 [73,3]**	**57,0 [59,8]**	**93,0 [86,6]**	**96,0 [90,2]**	**95,6 [89,5]**	**88,9 [83,6]**
OFDIW	RS50	55,2 [40,0]	68,8 [59,3]	41,3 [38,7]	68,5 [19,2]	90,4 [73,7]	90,1 [72,7]	75,8 [59,0]
DEEPEC	RS50	31,2 [10,7]	**75,7** [65,8]	26,3 [17,7]	54,3 [22,5]	83,1 [38,8]	93,0 [74,5]	63,1 [37,1]
NITEC *(Vorg. Ansatz)*	RS50	**57,2 [53,6]**	73,8 **[71,7]**	**57,2 [57,0]**	**89,7 [84,5]**	**95,2 [90,6]**	**96,7 [90,5]**	**87,9 [83,0]**

Tabelle 8.3: Vergleich der unterschiedlichen Datensätze mit einfachen Baselinemodellen auf der Grundlage von ResNet18 und ResNet50. Die Klassifizierung von Blickkontakt wird anhand der Average Precision (AP) und des F1-Scores bewertet.

(was auf die Größe des Trainingsdatensatzes zurückzuführen sein könnte). Das OFDIW ResNet18-Modell schneidet ebenfalls durchweg schlechter ab als das NITEC-ResNet18-Modell, sogar bei den Testdaten des eigenen OFDIW Datensatzes selbst. Dass die NITEC Modelle durchweg dominante Ergebnisse liefern, spricht für eine hohe Generalisierung, die vom NITEC-Datensatz vermittelt werden kann und durch die Kombination der unterschiedlichen Datenquellen motiviert wurde. Ferner ist die modellübergreifende geringe Performanz auf dem OFDIW Testdatensatz auffällig. Der Grund hierfür lässt sich durch starkes Label-Rauschen argumentieren, welches in stichprobenartigen qualitativen Tests bekräftigt werden konnte.

Bei der Analyse der Testdatensätze, bei denen die Unterschiede zwischen den Modellen am deutlichsten sind, zeigt sich, dass die Ergebnisse bei dem besonders anspruchsvollen WIDER FACE-Datensatz nicht nur schlechter sind als bei anderen Testdatensätzen, sondern auch erhebliche Unterschiede zwischen den Modellen aufweisen, wobei das NITEC Modell durchweg besser abschneidet als die anderen. Ähnlich verhält es sich mit dem anspruchsvollen CelebA-Datensatz, der schwierige Blickwinkel enthält, die nur marginal keinem Blickkontakt entsprechen. Da diese Ergebnisse sowohl für die durchschnittliche Genauigkeit als auch für den F1-Score gelten, ermöglicht der NITEC Datensatz im Vergleich zu den anderen Datensätzen eine bessere Generalisierung der relevanten Merkmale für Blickkontaktdaten. Diese Erkenntnisse lassen sich auf die komplexeren ResNet50-Modelle übertragen. Auch hier schneidet das mit NITEC trainierte Modell besser ab als die Vergleichsmodelle, mit Ausnahme des DEEPEC Testdatensatzes, bei dem das auf DEEPEC Daten trainierte Modell eine etwas bessere durchschnittliche Präzision erreicht, während der F1-Score für das NITEC Modell höher bleibt. Ein Vergleich der ResNet18-Modelle mit den ResNet50-Modellen zeigt, dass die größeren Modelle keine signifikanten Verbesserungen bringen (mit Ausnahme einer leichten Verbesserung bei DEEPEC), sondern eher an Robustheit verlieren. Es lässt sich argumentieren, dass die ResNet50-Modelle aufgrund der schnelleren Konvergenz innerhalb von 20 Epochen eine Überanpassung auf den Datensätzen dieser Größe aufweisen.

Methode	Backbone	Blickkontakt Klassifikation (Accuracy) ↑ [F1-Score] ↑						
		OFDIW	DEEPEC	NITEC-WF	NITEC-Gaze360	NITEC-CelebA	NITEC-Helen	NITEC
6DRepNet-5	ResNet50	75,4 [3,6]	54,1 [3,1]	79,6 [2,4]	50,4 [5,9]	36,4 [1,3]	43,6 [3,9]	57,0 [2,7]
6DRepNet-15	ResNet50	60,9 [28,8]	55,1 [33,5]	74,5 [22,3]	59,5 [48,2]	47,9 [42,9]	49,9 [35,7]	60,9 [39,0]
6DRepNet-25	ResNet50	50,0 [33,8]	54,7 [49,5]	70,8 [37,3]	61,0 [56,8]	61,0 [69,4]	59,0 [61,8]	64,9 [59,4]
L2SC-Net [303]-5	ResNet50	72,0 [18,6]	53,4 [24,8]	79,1 [15,7]	54,5 [21,6]	46,7 [34,1]	51,0 [30,4]	61,6 [27,9]
L2SC-Net [303]-15	ResNet50	59,4 [39,7]	59,4 [57,8]	74,0 [41,3]	68,6 [62,5]	66,0 [73,9]	70,3 [72,2]	70,1 [64,9]
L2SC-Net [303]-25	ResNet50	46,7 [41,2]	56,7 [62,5]	66,0 [45,1]	77,3 [78,9]	69,5 [79,5]	73,9 [79,3]	69,8 [71,1]
Gaze360 [241]*-5	RS18-LSTM	53,9 [11,0]	75,1 [3,7]	79,2 [4,2]	49,8 [2,3]	38,5 [9,4]	45,9 [12,3]	57,6 [7,3]
Gaze360 [241]*-15	RS18-LSTM	56,2 [42,0]	68,3 [20,6]	75,3 [25,8]	57,2 [32,8]	52,9 [51,9]	57,0 [50,4]	63,1 [43,1]
Gaze360 [241]*-25	RS18-LSTM	57,9 [31,3]	57,1 [57,2]	68,6 [35,9]	63,3 [54,9]	64,4 [71,8]	66,9 [71,2]	66,1 [60,7]
Chong [292]*	ResNet50	59,3 [47,8]	68,2 [45,9]	68,3 [45,9]	76,1 [73,9]	75,3 [79,9]	81,9 [83,8]	73,1 [70,2]
OFDIW [293]	ResNet18	79,3 [33,1]	67,6 [61,2]	81,7 [37,2]	54,4 [19,9]	74,8 [76,7]	76,6 [75,4]	74,3 [61,2]
OFDIW [293]	ResNet50	79,7 [40,0]	66,5 [59,3]	80,5 [38,7]	53,8 [19,2]	71,7 [73,7]	74,1 [72,7]	72,5 [59,0]
DEEPEC [294]	ResNet18	75,3 [10,7]	67,5 [62,7]	77,7 [23,9]	53,4 [27,7]	51,2 [42,1]	74,7 [74,5]	64,2 [39,9]
DEEPEC [294]	ResNet50	75,3 [10,7]	70,0 [65,8]	77,6 [17,7]	49,8 [22,5]	50,3 [38,8]	79,4 [79,5]	63,6 [37,1]
NITEC ***(Vorg. Ansatz)***	ResNet18	**80,6** [55,3]	**74,3 [73,3]**	**84,3** [59,8]	87,1 [86,7]	88,1 [90,3]	88,6 [89,5]	**86,4 [83,6]**
NITEC ***(Vorg. Ansatz)***	ResNet50	77,8 [53,6]	72,2 [71,7]	82,8 [57,0]	85,1 [84,5]	**88,3 [90,6]**	**89,3 [90,5]**	85,6 [83,0]
NITEC ***(Vorg. Ansatz)***	SWIN-Tiny	79,0 **[55,7]**	74,2 [71,5]	84,1 **[60,6]**	**87,8 [87,8]**	85,6 [88,1]	86,7 [87,9]	85,4 [82,6]
NITEC ***(Vorg. Ansatz)***	SWIN-Small	80,0 [53,9]	73,0 [70,0]	82,9 [57,4]	81,0 [78,8]	86,5 [88,7]	85,9 [87,1]	84,1 [80,4]
NITEC ***(Vorg. Ansatz)***	SWIN-Base	80,1 [44,1]	72,4 [67,1]	83,4 [52,8]	84,5 [83,1]	74,3 [75,3]	80,2 [79,8]	80,2 [73,0]

Tabelle 8.4: .
Vergleich der Modelle zur Klassifizierung des Blickkontakts, einschließlich Methoden zur Kopfposeschätzung und Blickrichtungsschätzung. Die verwendeten Metriken sind Accuracy und F1-Score (in eckigen Klammern). Modellergebnisse mit * werden von den Originalautoren bereitgestellt.

Vergleich mit dem Stand der Technik In Tabelle 8.4 werden weitere Ergebnisse eines quantitativen Experiments aufgeführt. Anstelle der durchschnittlichen Präzision wird die Accuracy zusammen mit dem F1-Score gewählt, um zusätzliche Modelle aus dem Stand der Technik in den Vergleich einbeziehen zu können. Hierzu wird das 6DRepnet [8] verwendet, welches primär zur Schätzung von Kopfposen eingesetzt wird. Zwei weitere Modelle dienen primär der Blickrichtungsschätzung (L2SC-Net [303], Gaze360 [241]). Beide Modelle wurden mit dem Gaze360-Datensatz trainiert. Zusätzlich wird das von Chong *et al*. [292] vorgeschlagene Modell als Vergleichsmaßstab herangezogen.

Für die Modelle der Kopfposeschätzung und der Prädiktion der Blickrichtung wird derselbe Ansatz verwendet, wie bei der Annotation der Gaze360 Daten für den NITEC Datensatz. Dazu werden Proben als Augenkontakt annotiert, wenn die Gier- und Nickwinkel im Bereich von $\pm$ 5° liegen. Alle anderen Vorhersagen werden als Nicht-Augenkontakt definiert. In zwei weiteren Tests wurde diese Schwelle auf $\pm$ 15° und $\pm$ 25° erhöht.

Die Ergebnisse zeigen, dass Kopfposemodelle (6DRepNet) für die Vorhersage von Blickkontakt nicht geeignet sind. Obwohl bei den Testdaten von OFDIW und WIDER FACE eine Genauigkeit von über 70 Prozent erreicht wird, sollten diese Ergebnisse mit Vorsicht interpretiert werden, da die Label-Verteilungen für diese beiden Testdatensätze stark ungleich gewichtet sind (siehe auch Tabelle 8.2). Die Ergebnisse des F1-Scores zeigen, dass weder der Recall noch die Precision kompetitive Werte erreichen. L2SC erzielt ähnliche Genauigkeitsergebnisse, erreicht aber zweistellige Werte im F1-Score und zeigt damit bessere, aber bisher nicht zufriedenstellende Ergebnisse. Die Resultate von Gaze360 ähneln denen von 6DRepNet bei einem Schwellenwertbereich von $\pm$ 5°. Wenn der Schwellenwert jedoch auf $\pm$ 15° oder sogar $\pm$ 25° erhöht wird, zeigen alle drei Methoden eine signifikante Verbesserung des F1-Scores.

Dies kann darauf zurückgeführt werden, dass die Vorhersagefehler der Modelle für Blickrichtung und Kopfhaltung größer sind als die betrachteten Intervalle für den Blickkontakt. Dieses Verhalten wird in Abschnitt 8.3.3 weiter analysiert.

Für NITEC Modelle auf der Basis von ResNet und der SWIN-Transformer-tiny [21]-Architekturen werden mit deutlichem Abstand die besten Ergebnisse erzielt, sodass sie sich auch in diesem Vergleich als das effizienteste und am besten generalisierte Modell durchsetzen können. Es lässt sich argumentieren, dass zur Erzielung besserer Ergebnisse für die *small* und *base* SWIN-Architekturen, mehr Trainingsdaten erforderlich sind, als der NITEC Datensatz derzeit zu bieten hat.

Vergleich ResNet18 zu ResNet50 Um besser zu verstehen, welche Merkmale für die Klassifikation der Proben ausschlaggebend sind, wurden in Abbildung 8.6 für drei exemplarische Datenproben die Gradientenaktivierungen visualisiert. Die Darstellung verdeutlicht, dass das ResNet18 Modell die elementaren Bildinformationen des Gesichtsbereichs effektiver ausnutzt. Teilbild 8.6 a) zeigt, dass ResNet18 umfangreichere Gesichtsregionen einbezieht, was die Integration von nuancierten Aspekten, wie der Kopfhaltung ermöglicht. Darüber hinaus können auf diese Weise weitere relevante Gesichtsmerkmal wie Lächeln und weitere prägnante Gesichtsausdrücke in ResNet18 berücksichtigt werden, insbesondere wenn die Erkennung der Augen eine Herausforderung darstellt. Es ist jedoch unverkennbar, dass die Einbeziehung zusätzlicher Gesichtsbereiche immer nur in Kombination

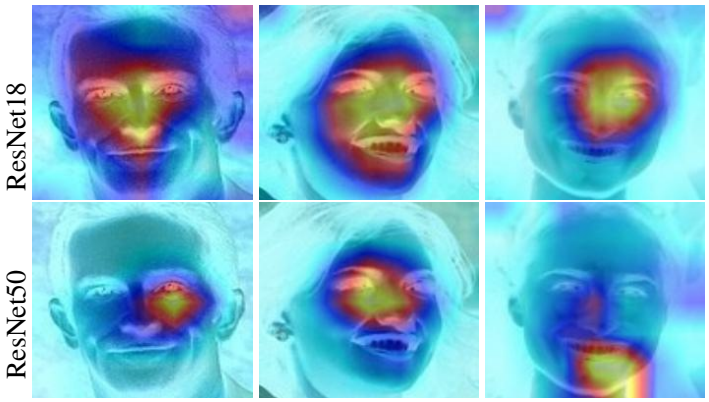


Abbildung 8.6: Visualisierung der Gradient Class Activation Maps [304] zwischen ResNet18 und ResNet50 auf Basis von Bildern aus dem CelebA-Datensatz [299]. © 2024 IEEE

Train \ Test	Blickkontakt-Klassifikation (AP) ↑ [F1-Score] ↑				
	WF	Gaze360	CelebA	Helen	NITEC
WF	52,8 [46,5]	75,7 [35,5]	82,1 [56,8]	87,4 [73,4]	76,6 [54,0]
Gaze360	38,8 [35,8]	87,4 [82,7]	80,5 [66,9]	84,3 [66,7]	75,5 [64,6]
CelebA	53,4 [55,0]	77,4 [52,0]	91,8 [86,0]	93,0 [86,1]	81,3 [74,1]
Helen	43,0 [39,8]	65,5 [36,7]	78,0 [61,0]	84,1 [74,9]	69,5 [54,6]
NITEC	**57,0 [59,8]**	**93,0 [86,6]**	**96,0 [90,2]**	**95,6 [89,5]**	**88,9 [83,6]**

Tabelle 8.5: NITEC-Subset-Evaluation auf Grundlage des ResNet18-Baseline-Modells.

mit dem Augenbereich einhergeht, wie aus dem Teilbild 8.6 b) hervorgeht. Umgekehrt neigt das Modell auf Basis des ResNet50 dazu, sich zu sehr auf bestimmte Zusatzattribute zu verlassen, was unbeabsichtigt dazu führt, dass die Bedeutung der Augeninformationen nicht ausreichend einbezogen wird. Das ResNet50 konzentriert sich in erster Linie auf einzelne hervorstechende Indikatoren für die Bewertung des Augenkontakts, jedoch oft ohne robuste Integration weiterer Gesichtsmerkmale, wie Teilbild 8.6 c) aufzeigt. Insbesondere die abgegrenzten Bildregionen, die sich auf die Augen beziehen, sind in höherem Maße vom breiteren Gesichtskontext isoliert, was zu einer geringeren Robustheit bei der Erkennung führt.

Schlussfolgernd lässt sich sagen, dass Aufgaben zur Erkennung von Blickkontakt mit weniger komplexen Architekturen hinreichend gelöst werden können. Dies verdeutlicht, dass durch den Datensatz die relevanten Merkmale für die Erkennung von Augenkontakt in verschiedenen Szenarien erfasst und effizient antrainiert werden können.

In-Dataset Evaluation In einem nächsten Schritt liegt der Fokus darauf, den Einfluss der einzelnen Subsets von NITEC auf dessen Gesamtqualität zu untersuchen. Dazu wurde ein In-Dataset

Experiment (Tabelle 8.5) durchgeführt, bei dem das ResNet18-Baseline-Modell auf jedem der Subsets trainiert und getestet wurde. Hierfür wurde jedes Subset in 80% Trainings- und 20% Testdaten aufgeteilt. Für die Evaluation wurde die durchschnittliche Präzision als primäre Metrik, ergänzt durch den F1-Score (in Klammern), verwendet. Das Modell, das auf dem CelebA Subset trainiert wurde, zeigt die stärkste Performanz, da es nicht nur auf seinem eigenen Testset, sondern auch auf dem WIDER FACE und Helen Testset besser abschneidet. Bemerkenswerterweise übertrifft das auf dem gesamten Datensatz trainierte Modell alle anderen Modelle, was den Synergieeffekt der Zusammensetzung der ausgewählten Daten aus den unterschiedlichen Datensätzen verdeutlicht.

8.3.2 Qualitative Evaluation

Für die qualitative Analyse wurden fünf Beispielbilder verwendet und die ResNet18-Baseline-Modelle für NITEC, OFDIW sowie das Chong *et al*.- und das blickrichtungsbasierte L2SC-Net-Modell (mit einem Schwellenwert von 5°) eingesetzt. Die Ergebnisse sind in Abbildung 8.7 dargestellt. Es wird deutlich, dass das OFDIW und das L2SC Modell nicht hinreichend in der Lage sind, die Mehrzahl an Gesichtern mit Blickkontakt korrekt zu bestimmen, während OFDIW Gesichter von geringer Qualität falsch klassifiziert (zweite und dritte Reihe). Die NITEC und Chong *et al*. Modelle sind hingegen in der Lage, die Augenkontaktkandidaten in Reihe zwei und vier korrekt zu bestimmen. Ein besonderer Unterschied zwischen diesen beiden Modellen zeigt sich bei den schwierigeren Beispielen in der dritten und fünften Reihe. Hier prädiziert Chong *et al*. Fasch-Positive für stark verschwommene Gesichter im Hintergrund, während das vorgestellte NITEC Modell zu strengeren Entscheidungen neigt. Der Grund dafür könnte in der Wahl von Proben aus dem WIDER FACE-Subdatensatz zu finden sein, die mit der Motivation eingesetzt wurden, potenzielle Falsch-Positive in schwierigen Bildern zu unterdrücken. In einigen Fällen kann dies jedoch zu Falsch-Negativen führen, wie es in der fünften Reihe beim Mädchen im Vordergrund auffällig ist. Dieser Effekt wird in Abschnitt 8.3.3 genauer analysiert.

8.3.3 Prädiktionsverteilungsanalyse

Abbildung 8.8 zeigt einen weiteren qualitativen Vergleich der Prädiktion von Blickkontakt und Nicht-Blickkontakt auf Daten des MPIIFaceGaze-Datensatzes [305] unter Verwendung der Baseline-Modelle von Chong *et al*. [292], Gaze360 [241] und des NITEC ResNet18-Modells. MPIIFaceGaze ist, ähnlich wie Gaze360, ein Datensatz für die Blickrichtungsbestimmung und besteht aus 37.788 Proben. Die Proben wurden in einer Laborumgebung aufgenommen, wobei die Probanden in einem relativ kleinen Bereich um die Kamera herum gleichmäßig auf die Kameraebene fokussieren, mit Ausnahme des Bereichs über der Kamera, was zu einem Mangel an Informationen in diesem Bereich führt. Außerdem haben die Personen einen ähnlichen Abstand zur Kamera.

(a) NITEC **(b)** Chong *et al.* **(c)** OFDIW **(d)** L2SC-Net

Abbildung 8.7: Exemplarische qualitative Ergebnisse der Blickkontaktklassikation auf Basis der Modelle von NITEC, Chong, OFDIW und L2SC. © 2024 IEEE

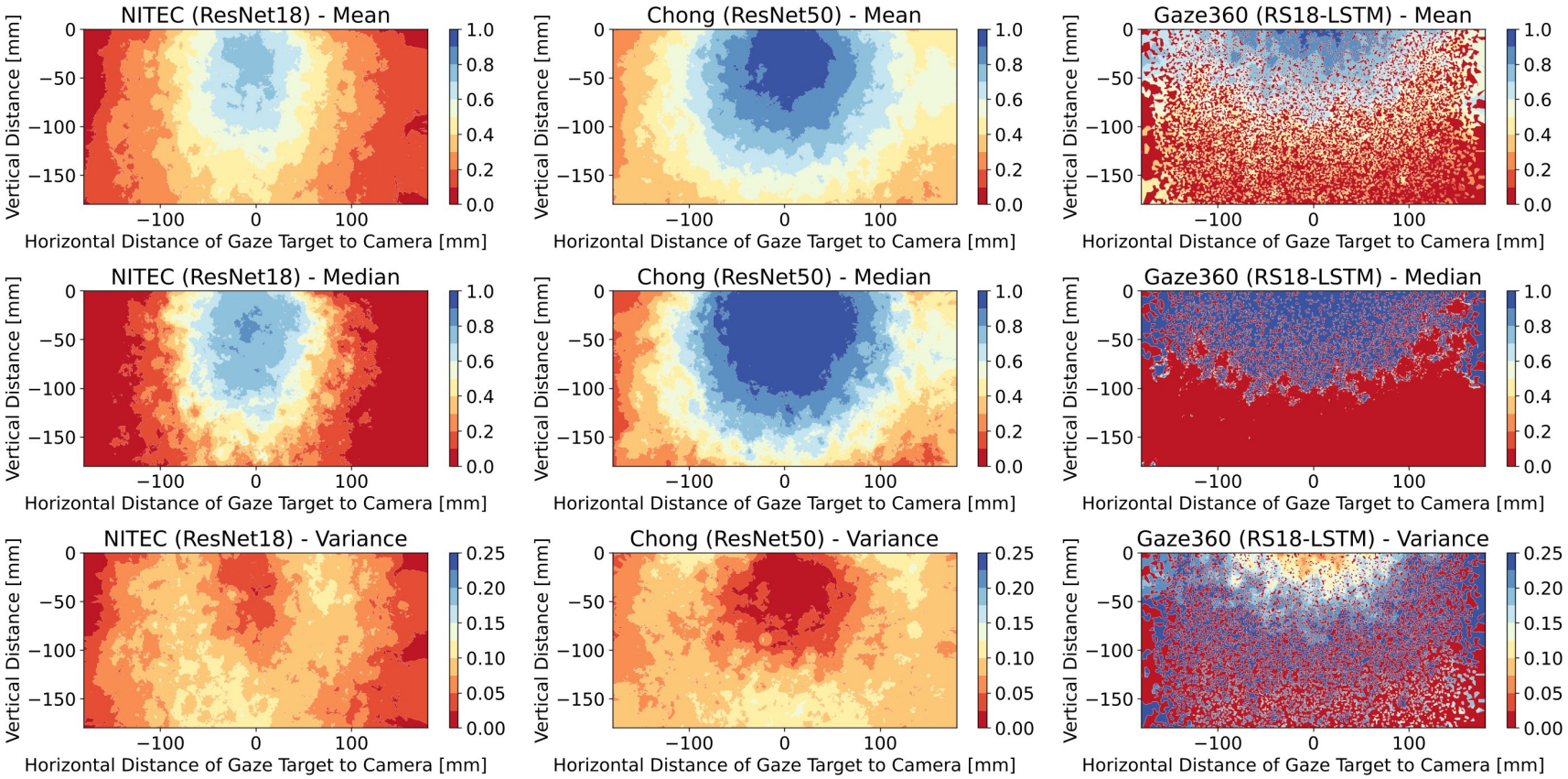


Abbildung 8.8: Qualitativer Vergleich zwischen unterschiedlichen Datensätzen unter Verwendung einfacher Baseline-Modelle auf dem MPIIFaceGaze-Datensatz [305]. © 2024 IEEE

In Abbildung 8.8 werden die vorhergesagten Werte mithilfe eines k-nearest-neighbors-Algorithmus (k=100) aggregiert und mit ihren spezifischen Blickzielpositionen relativ zur Kamera und den Prädiktionen in Farbe dargestellt. Der Vergleich umfasst das arithmetische Mittel, den Median und die Varianz. Bei der Betrachtung der Mittelwerte und Mediane für das Chong *et al*. und NITEC Modell ist eine Verschiebung der Hauptregion, die von dem Modell als Blickkontakt klassifiziert wird, nach unten erkennbar. Die Untersuchung des Gaze360 Modells zeigt jedoch keine solche Verschiebung, wenn nur die Blickrichtung betrachtet wird. Dies deutet darauf hin, dass die Diskrepanz nicht im MPIIFaceGaze Datensatz oder dessen Auswertung liegt, sondern eher in den Trainingsdaten der Modelle. Eine Erklärung hierfür liegt im Annotationsprozess der Daten, bei dem die Annotatoren auch dann Augenkontakt erkannt haben, wenn der Blick auch auf die Gesichtsregion unter den Augen fiel. Da die Daten handannotiert sind und sich die Augen im oberen Drittel des Gesichts befinden, tritt die Verschiebung in der Region auf, in der Augenkontakt erkannt wird. Die Diagramme für den Mittelwert und den Median zeigen auch, dass die Modelle an Vertrauen gewinnen, wenn sich der tatsächliche Brennpunkt auf der horizontalen Achse der Kamera nähert und höhere Werte erreichen, wenn er sich auf der vertikalen Achse knapp unterhalb der Kamera befindet. Sowohl das Chong *et al*. Modell als auch das NITEC Modell zeigen eine gleichmäßige Abnahme der prädizierten Blickkontaktwerte mit zunehmender Entfernung vom Zentrum des Blickkontakts (sowohl im Mittelwert als auch im Median). Im Vergleich zu Chong *et al*. ist das NITEC Modell jedoch konservativer. Nur 6,7% der vom NITEC Modell vorhergesagten Werte liegen über eine Konfidenz von 0,75, während bei Chong *et al*.-Modell 35,7% der Werte über 0,75 liegen.

Andererseits liegen 56,6% der vom NITEC-Modell vorhergesagten Werte unter einer Konfidenzrate von 0,25, während beim Modell von Chong *et al*. nur 7,4% der Datenpunkte unter 0,25 prädiziert wurden. Dies ermöglicht größere Modifikationspotenziale beim NITEC Modell, um mit der Auswahl eines geeigneten Schwellenwertes die Erkennungsrate für ein bestimmtes Szenario anzupassen. Der Vergleich von Mittelwert und Median zeigt außerdem, dass die Entscheidungen des Modells eher zu den Extremen tendieren und der Übergang zwischen Augenkontakt und Nicht-Augenkontakt bei der Betrachtung von Durchschnittswerten langsamer ist als das, was das Modell in den meisten Fällen vorhersagen würde, wie in den Teilgrafiken für den Median ersichtlich ist. Ein weiteres wichtiges Maß für die Generalisierungsfähigkeit des Modells ist die Streuung der Vorhersagen. Daher wird auch die Varianz innerhalb der Regionen, die aus dem k-nearest-neighbors-Algorithmus abgeleitet wurden, auf der Grundlage jedes Satzes von 100 Proben dargestellt.

Wie erwartet, nimmt die Varianz in den Übergangsbereichen zu. Insgesamt zeigt sich, dass Modelle, die direkt auf Blickkontakt trainiert wurden, eine signifikant geringere Varianz aufweisen, was auf eine höhere Robustheit hindeutet als Modelle, die nur auf Blickrichtung trainiert wurden. Ähnlich wie bei der quantitativen Analyse wird deutlich, dass die Erkennung von Blickkontakt besondere Herausforderungen mit sich bringt, die mit den bestehenden Ansätzen für die Blickrichtung nicht adäquat gelöst werden können. Dies ist in erster Linie auf den erheblichen Vorhersagefehler und die damit verbundene Varianz zurückzuführen. Dies rechtfertigt die Notwendigkeit eines eigenen Datensatzes und spezieller Modelle für den Blickkontakt. Diese Modelle bieten mehr Flexibilität bei der Anpassung des Schwellenwerts an den spezifischen Anwendungsbereich des Blickkontakts und bieten eine deutlich höhere Robustheit. Weitere qualitative Analysen müssen berücksichtigt werden, z. B.

die Entfernung der Probanden von der Kamera und ihre nicht auf die Kameraebene ausgerichteten Brennpunkte. Darüber hinaus kann eine qualitative Untersuchung der Vorhersagefehler der Modelle Aufschluss über die Grenzen des aktuellen Datensatzes und mögliche Verbesserungsbereiche geben.

8.3.4 Probandenstudie

In den vorangegangenen Experimenten wurden die erzeugten Modelle mehreren quantitativen und qualitativen Analysen unterzogen. Um die tatsächliche Applikationsfähigkeit der Modelle zu untersuchen, wurde eine zusätzliche Probandenstudie durchgeführt, in der die Probanden kurze Interaktionen mit einem Roboter durchlaufen. Hierfür wurde ein humanoider Roboter eingesetzt, der eine Kamera im Kopf an der Stelle der Augen montiert hat. Somit bietet dieser die idealen Voraussetzungen, um Augenkontakt ausgehend von menschlichen Interaktionspartnern zu erfassen.

Für die Studie wurden insgesamt neun Probanden eingesetzt, die bisher keine Erfahrungen mit diesem Roboter hatten. Ihnen wurde die Aufgabe gegeben, den Roboter anzusprechen und die Überreichung eines Objektes zur erfragen. Auf diese Anfrage dreht sich der Roboter um und hebt einen Würfel von einem Tisch auf, um diesen im Anschluss der Person zu übergeben. Die Aufnahme von Augenkontakt war weder Teil der Aufgabenstellung für die Probanden, noch wurden die Probanden über die Zielstellung der Studie im Vorhinein in Kenntnis gesetzt. Der Blickkontakt erfolgte deshalb ausschließlich aus natürlicher und intuitiver Motivation heraus.

Die aufgenommenen Kamerabilder wurden anschließend durch einen Annotator mit Grundwahrheiten versehen. Der sich daraus ergebene Datensatz beinhaltet 5.888 Proben. Abbildung 8.9 illustriert den Vergleich der Performanz zwischen den Blickkontakt-Klassifikationsmodellen auf den einzelnen Probandenszenarien. Als Evaluationsmetrik wird die bereits zuvor verwendete Average Precision (AP) eingesetzt. Während für die OFDIW, DEEPEC und Chong Modelle eine ResNet-Variante als Backbone dient, basiert das NITEC-Modell auf dem Vision-Transformer MobileVit [306] der Größe *xxs*.

Abbildung 8.9: Vergleich der Average Precision über die unterschiedlichen Probanden der Studie.

Es zeigt sich, dass das NITEC Modell in sieben der neun Szenarien die höchste Average Precision erzielt und damit übergreifend die beste Erkennungsrate demonstriert. In einem der Szenarien wird dabei ein Wert von 0,89 erreicht, während für dieselben Samples das DEEPEC-Modell nur 0,32 erreicht. In den anderen zwei Szenarien erreicht das Modell von Chong *et al*. die höchsten AP Werte, welcher auch in den anderen Szenarien meistens an zweiter Stelle steht. Diese Ergebnisse reflektieren das Prädiktionsverhalten, welches bereits in den Experimenten zuvor herausgearbeitet wurde. Somit erzielen die auf NITEC trainierten Modelle die höchsten Erkennungsraten im Vergleich zum Stand der Technik und zeigen auf, dass die Qualität des Datensatzes von entscheidender Bedeutung ist und im vorgelegten Fall die Millionen Proben von Chong. *et al*.übertreffen.

Abbildung 8.10 zeigt einige exemplarische Aufnahmen aus der Roboterperspektive, die während der Studie aufgenommen wurden. Diese verdeutlichen die herausfordernden Lichtverhältnisse, unter denen die Szenarien aufgenommen wurden und die Prädiktionen des Blickkontakts erschweren. Dennoch ist das vorgeschlagene NITEC Modell in der Lage, trotz der wenig ausgeleuchteten Gesichter robuste Entscheidungen zu treffen.

Abbildung 8.10: Qualitative Ergebnisse aus der Probandenstudie, aufgenommen aus der Perspektive des Roboters. Die grüne (Blickkontakt) und rote Schraffierung (kein Blickkontakt) deutet die Klassifikation des NITEC-Modells an.

8.4 Diskussion

Blickkontakt ist einer der wichtigsten nonverbalen Kommunikationswege zwischen Menschen und ein elementares Werkzeug zur missverständnisfreien Interaktion. In diesem Kapitel wurde ein neuer, handannotierter Datensatz für die bildbasierte Erkennung von Blickkontakt aus der Egoperspektive

mit dem Namen NITEC vorgestellt. Mit diesem Datensatz wird auch Robotern Blickkontakt als essenzielles Kommunikationswerkzeug zugänglich gemacht.

Der NITEC Datensatz enthält insgesamt 35.919 handannotierte Proben und übersteigt mit seinem Umfang alle bisher veröffentlichten Datenbanken für Blickkontakt. Anhand mehrerer quantitativer Auswertungen konnte die Qualität des Datensatzes systematisch nachgewiesen und das Generalisierungspotenzial selbst bei kleinen Baseline-Modellen aufgezeigt werden. Gleichzeitig wurde gezeigt, dass die bisher verwendeten Modelle zur Prädiktion der Kopfpose und der Blickrichtung nicht für die Erkennung von Blickkontakt geeignet sind, was die Relevanz des vorgeschlagenen Datensatzes bekräftigt. In weiteren qualitativen Experimenten zeigen die NITEC-Modelle eine hohe Robustheit bei besonders schwierigen Proben, bei denen andere Modelle aus dem Stand der Technik keine zufriedenstellende Leistung erzielen. In einem zusätzlichen Experiment wurde das Prädiktionsverhalten verschiedener Modelle im Hinblick auf die Vorhersageverteilung analysiert, um Einblicke in ihre Zuverlässigkeit und Konsistenz zu gewinnen.

Um die Performanz unter realen Bedingungen zu prüfen, wurde eine Probandenstudie durchgeführt, in der Testpersonen mit einem humanoiden Roboter intuitiv interagierten. Aus den Szenarien wurden im Anschluss weitere 5.888 Proben herausgearbeitet und annotiert. Der Test auf diesen Daten bestätigt die vorangegangenen Ergebnisse und zeigt ein zuverlässiges und robustes Prädiktionsverhalten der NITEC-Modelle auf, das den Stand der Technik übertrifft.

KAPITEL 9

Zusammenfassung und Ausblick

Mobile Roboter sind bereits in vielen Bereichen der Industrie, Medizin und des Dienstleistungssektors ein essenzieller Bestandteil der Wertschöpfungskette und tragen zur Automatisierung, Kostenreduzierung und Genauigkeit von Prozessen bei. Um ihre Flexibilität und Einsatzmöglichkeiten zu erweitern und ihnen ein breiteres und komplexeres Spektrum an Aufgaben, insbesondere in Zusammenarbeit mit Menschen, zugänglich zu machen, müssen jedoch ihre Fähigkeiten zur Autonomie gestärkt werden.

Eine entscheidende Herausforderung liegt in der gezielten Erfassung der Umgebung, um situativ dem Kontext angepasste Handlungen vornehmen zu können. Hierzu zählt eine präzise Erfassung des Aktionsraums des Roboters. Eine solche Analyse muss sowohl geometrische Aspekte – wie Abmessungen und die räumliche Anordnung der Umgebung – als auch semantische Komponenten – wie die Position, Bedeutung und Funktion von Objekten – umfassen. Dabei erfordern die eingesetzten Methoden eine robuste Verhaltensweise gegenüber dynamischen Veränderungen und unübersichtlichen Szenenwechseln.

Weiterhin ist für eine effektive und intuitive Interaktion zwischen Mensch und Roboter die detaillierte Wahrnehmung und Erfassung von Personen unerlässlich. Hierzu gehört nicht nur die reine Detektion von Personen in der Nähe des Roboters, sondern auch die Erfassung des momentanen Fokus der Aufmerksamkeit. Dies ermöglicht eine harmonische Zusammenarbeit, bei der der Roboter als ein hilfreicher und intelligenter Partner fungieren kann, der auf die Bedürfnisse und Sicherheit der Menschen in seiner Umgebung eingeht.

9.1 Zusammenfassung und wissenschaftliche Beiträge

In dieser Arbeit werden systematisch eine Reihe von bildbasierten Deep Learning Methoden entwickelt, implementiert und evaluiert, welche die soziale Autonomie mobiler Roboter verbessern

und den Informationsgehalt zur Bestimmung adäquater Verhaltensstrategien steigern. Die Ansätze umfassen fünf unterschiedliche wissenschaftliche Beiträge, die sich auf die Bereiche der robusten räumlich-semantischen Umgebungsanalyse und der Analyse menschlicher Interaktionspartner aufteilen.

In Kapitel 4 befasst sich der erste wissenschaftliche Beitrag mit der Orientierung mobiler Roboter in dynamischen Umgebungen. Dieser Aspekt entspricht einer wesentlichen Herausforderung von mobilen Robotern, deren autonome Orientierung auf visuellem SLAM basiert. Diesem Problemfeld wird mit einem neuen Lösungsansatz begegnet, bei dem ein Deep Learning generierter optischer Fluss zu einem Szenen-Flow erweitert wird, anhand dessen eine Homografie der Kamerabewegung geschätzt wird. Mittels Reprojektionsfehler wurden im Anschluss Bewegungsvektoren segmentiert, deren Bewegungsverhalten sich von der geschätzten Kamerabewegung abhebt. Durch diesen Ansatz wird eine feine, pixelbasierte Erfassung von dynamischen Bildelementen ermöglicht, welche im Gegensatz zum vorherigen Stand der Technik weder *a priori* Wissen benötigt, noch eine Übersegmentierung von statischem Hintergrund verursacht. Eine Evaluation auf öffentlichen Datenbanken weist dem vorgeschlagenen Ansatz eine Effektivität nach, mit der der Trajektorienfehler um bis zu 98% verringert werden kann. Die bildpaarbasierte Verarbeitung ermöglicht eine zielgenaue Eliminierung dynamischer Bildelemente, erfordert jedoch eine lange Prozesslaufzeit, welche die Echtzeitfähigkeit von mobilen Robotern beeinträchtigen kann.

Kapitel 5 stellt eine neue Methode zur semantischen Kartierung vor, bei der rein geometrische Umgebungskarten durch semantische Objekte erweitert werden. Die Erkennung und Lokalisierung semantischer Objekte verbessert das Verständnis der Umgebung und ermöglicht das Greifen und Transportieren von Objekten – ein essenzieller Bestandteil für viele Arten von Mensch-Roboter-Interaktionen. Die Methode verfolgt eine 2D-zu-3D-Pipeline, bei der Prädiktionen eines bildbasierten Deep Learning Objektdetektors in den 3D-Raum projiziert werden. Für die Datenassoziation wurde ein Zwei-Stufen-Verfahren vorgestellt. In der ersten Stufe wird ein effizienter Schnellabgleich durchgeführt, der potenziell assoziierbare Objekte bestimmt. In einem zweiten Schritt wird zwischen potenziellen Kandidaten ein komplexerer, punktwolkenbasierter Abgleich durchgeführt. Letzteres wird jedoch nur bei Bedarf realisiert, sofern mehrere vorhandene Objekte zur Assoziation infrage kommen. Dies ermöglicht eine effiziente Skalierung und den Einsatz mobiler Verarbeitungseinheiten.

Kapitel 6 befasst sich mit der bildbasierten Erkennung der menschlichen Kopfpose, die wichtige Informationen zur Bestimmung der Aufmerksamkeit und entsprechend zur Mensch-Roboter-Interaktion bereithält. Hierbei wird eine neue Methode vorgestellt, welche die Rotationsrepräsentationsinhärenten Limitationen bisheriger Ansätze überwindet und damit erstmals ein Modell erzeugt, welches in der Lage ist, den gesamten Rotationsbereich des Kopfes robust zu erlernen und zu prädizieren. Hierzu wird ein neuer Datensatz vorgestellt, der sich aus mehreren bestehenden Datensätzen zusammensetzt und Proben über den gesamten Bereich der Rotation bereitstellt. Die vorgestellte Methode erzeugt dadurch nicht nur Prädiktionsgenauigkeiten, welche den aktuellen Stand der Technik übertreffen, sondern ist auch in der Lage, den Rotationsbereich bisheriger Methoden zu erweitern und somit komplexen und herausfordernden Interaktionen zwischen Mensch und Roboter zugänglich zu machen.

Kapitel 7 erweitert den vorgeschlagenen Ansatz aus Kapitel 6 zu einem Multi-Task Modell, bei dem parallel zur Kopfpose auch die Blickrichtung prädiziert wird. Hierbei werden die Synergien zwischen Kopfpose und Blickrichtung genutzt, um über ein simultanes Training die Genauigkeit und Robustheit beider Aufgaben zu verbessern. Eine umfangreiche Evaluation zeigt auf, dass insbesondere die Blickrichtungserkennung durch das Multi-Task-Learning eine signifikante Stärkung der Generalisierung in ihrem Prädiktionsverhalten erfährt.

Kapitel 8 behandelt eine essenzielle Kommunikationsform unter Menschen, die im Forschungsbereich der Mensch-Roboter-Interaktion jedoch bisher wenig Beachtung gefunden hat – den Blickkontakt. Blickkontakt ist ein elementares Werkzeug der nonverbalen Kommunikation und dient u. A. zur Kontaktaufnahme sowie zur Signalisierung von Akzeptanz, Interesse und Sympathie. Das Kapitel stellt einen Datensatz mit 35,919 Bild-Datenproben vor, in denen Personen in die aufnehmende Kamera schauen und somit Blickkontakt aus der Ego-Perspektive repräsentieren. Mithilfe dieser Daten werden Methoden zur Detektion angelernt, um Robotern den Aspekt des Blickkontakts zugänglich zu machen. Der Datensatz ist der größte und vielseitigste im Vergleich zum Stand der Technik und ermöglicht die Erzeugung besonders robuster Modelle, deren Performanz neben einer Reihe von quantitativen Experimenten auch in einer zusätzlichen Probandenstudie mit einem humanoiden Roboter bestätigt wird.

9.2 Ausblick

Die vorgestellten Methoden ermöglichen die bildbasierte Extraktion und Encodierung elementarer Informationen aus dem Umfeld von Robotern, die zur Verbesserung intuitiver, objektbezogener Mensch-Roboter-Interaktionen eingesetzt werden können. Um aus diesen Informationen jedoch adäquate Handlungsstrategien ableiten zu können, müssen nachfolgend weitere Herausforderungen bewältigt werden.

Die Fähigkeiten des *Reasoning* und *Decision Making* sind wichtige Elemente, um kontextbewusste und adaptive Entscheidungen zu treffen. Reasoning versteht sich hierbei als das Ableiten des benötigten Wissens (*Interaktionsbereitschaft, Zielobjekt*) aus dem, was explizit im verfügbaren Wissen dargelegt ist (*Kopfpose, Blickrichtung, Blickkontakt, semantische Karte*) [307]. Das Decision Making befasst sich im Anschluss mit der Wahl einer geeigneten Handlungsstrategie, sofern das benötigte Wissen generiert wurde. Beide Aufgaben stellen wesentliche Herausforderungen in der kognitiven Robotik dar. Je fundamentaler das Reasoning durchgeführt wird (z. B. durch Einbezug einer allgemein gegenwärtigen Wissensbasis / Common Sense), desto generalisierter ist die Einsatzfähigkeit des Roboters. Gleiches gilt für das Decision Making, welches mit wachsender Anzahl an Entscheidungsoptionen die Sensitivität und Feinfühligkeit intelligenter Interaktionen verbessern kann.

Während beide Fähigkeitsbereiche in den vergangenen Jahren zwar wachsende Relevanz in der Forschung gewonnen haben, bieten die jüngsten Durchbrüche in der Generative AI (z. B. ChatGPT, Midjourney und DALL-E) neue Perspektiven, um aus den Informationen der Umgebungs- und

Personenanalyse ein kontextspezifisches Situationsverständnis und menschenzentrierte Interaktionsstrategien zu erzeugen.

Literaturverzeichnis

[1] The rise of the cobots, 2019. URL https://commercial.allianz.com/news-and-insights/expert-risk-articles/global-risk-dialogue-cobots.html. Aufgerufen: 15.06.2024.

[2] Collaborative robot market size, share, industry report, revenue trends and growth drivers 2030, 2023. URL https://www.marketsandmarkets.com/Market-Reports/collaborative-robot-market-194541294.html. Aufgerufen: 15.06.2024.

[3] Judith A. Hall, Terrence G. Horgan, and Nora A. Murphy. Nonverbal communication. *Annual Review of Psychology*, 70(1):271–294, 2019. doi: 10.1146/annurev-psych-010418-103145. URL https://doi.org/10.1146/annurev-psych-010418-103145. PMID: 30256720.

[4] C. Breazeal, C.D. Kidd, A.L. Thomaz, G. Hoffman, and M. Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 708–713, 2005. doi: 10.1109/IROS.2005.1545011.

[5] Chapa Sirithunge, A. G. Buddhika P. Jayasekara, and D. P. Chandima. Proactive robots with the perception of nonverbal human behavior: A review. *IEEE Access*, 7:77308–77327, 2019. doi: 10.1109/ACCESS.2019.2921986.

[6] Thorsten Hempel and Ayoub Al-Hamadi. Pixel-wise motion segmentation for slam in dynamic environments. *IEEE Access*, 8:164521–164528, 2020. doi: 10.1109/ACCESS.2020.3022506.

[7] Thorsten Hempel, Magnus Jung, Ahmed A. Abdelrahman, and Ayoub Al-Hamadi. Nitec: Versatile hand-annotated eye contact dataset for ego-vision interaction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4437–4446, January 2024.

[8] Thorsten Hempel, Ahmed A. Abdelrahman, and Ayoub Al-Hamadi. 6d rotation representation for unconstrained head pose estimation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2496–2500, 2022. doi: 10.1109/ICIP46576.2022.9897219.

[9] Thorsten Hempel and Ayoub Al-Hamadi. Slam-based multistate tracking system for mobile human-robot interaction. In *International Conference on Image Analysis and Recognition*, pages 368–376. Springer, 2020.

[10] Thorsten Hempel and Ayoub Al-Hamadi. An online semantic mapping system for extending and enhancing visual slam. *Engineering Applications of Artificial Intelligence*, 111:104830, 2022. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2022.104830. URL https://www.sciencedirect.com/science/article/pii/S095219762200094X.

[11] Thorsten Hempel and Ayoub Al-Hamadi. Pixel-wise motion segmentation for slam in dynamic environments. *IEEE Access*, 08:164521 – 164528, 09 2020. doi: 10.1109/ACCESS.2020.3022506.

[12] Ahmed A Abdelrahman, Dominykas Strazdas, Aly Khalifa, Jan Hintz, Thorsten Hempel, and Ayoub Al-Hamadi. Multi-modal engagement prediction in multi-person human-robot interaction. *IEEE Access*, 2022.

[13] Thorsten Hempel, Ahmed A. Abdelrahman, and Ayoub Al-Hamadi. Towards robust and unconstrained full range of rotation head pose estimation. *IEEE Transactions on Image Processing*, pages 1–1, 2024. doi: 10.1109/TIP.2024.3378180.

[14] Thorsten Hempel, Laslo Dinges, and Ayoub Al-Hamadi. Sentiment-based engagement strategies for intuitive human-robot interaction. In *VISIGRAPP*, 2023. URL https://api.semanticscholar.org/CorpusID:255570053.

[15] Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. In *35th Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. doi: 10.1109/ICCV.2015.123.

[17] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. ISSN 0033-295X. doi: 10.1037/h0042519. URL http://dx.doi.org/10.1037/h0042519.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019.

Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[20] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[22] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[23] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58452-8.

[24] Robin Strudel, Ricardo Garcia Pinel, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7242–7252, 2021. URL https://api.semanticscholar.org/CorpusID:234470051.

[25] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[26] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *EMNLP*, pages 1412–1421. The Association for Computational Linguistics, 2015. ISBN 978-1-941643-32-7.

[27] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54 (10s):1–41, 2022.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern*

Recognition, CVPR '16, pages 770–778. IEEE, June 2016. doi: 10.1109/CVPR.2016.90. URL http://ieeexplore.ieee.org/document/7780459.

[30] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL http://dblp.uni-trier.de/db/journals/corr/corr1704.html#HowardZCKWWAA17.

[31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[32] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[33] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[34] Giulio Reina, Andres Vargas, Keiji Nagatani, and Kazuya Yoshida. Adaptive kalman filtering for gps-based mobile robot localization. In *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 1–6. IEEE, 2007.

[35] Ashutosh Singandhupe and Hung Manh La. A review of slam techniques and security in autonomous driving. In *2019 third IEEE international conference on robotic computing (IRC)*, pages 602–607. IEEE, 2019.

[36] Jun Cheng, Liyan Zhang, Qihong Chen, Xinrong Hu, and Jingcao Cai. A review of visual slam methods for autonomous driving vehicles. *Engineering Applications of Artificial Intelligence*, 114:104992, 2022.

[37] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.

[38] Shuran Zheng, Jinling Wang, Chris Rizos, Weidong Ding, and Ahmed El-Mowafy. Simultaneous localization and mapping (slam) for autonomous driving: Concept and analysis. *Remote Sensing*, 15(4):1156, 2023.

[39] Long Chen, Wen Tang, Nigel W John, Tao Ruan Wan, and Jian Jun Zhang. Slam-based dense surface reconstruction in monocular minimally invasive surgery and its application to augmented reality. *Computer methods and programs in biomedicine*, 158:135–146, 2018.

[40] Joao Pedro Mucheroni Covolan, Antonio Carlos Sementille, and Silvio Ricardo Rodrigues Sanches. A mapping of visual slam algorithms and their applications in augmented reality. In *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, pages 20–29. IEEE, 2020.

[41] Li Jinyu, Yang Bangbang, Chen Danpeng, Wang Nan, Zhang Guofeng, and Bao Hujun. Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality. *Virtual Reality & Intelligent Hardware*, 1(4):386–410, 2019.

[42] Young-Ho Choi, Tae-Kyeong Lee, and Se-Young Oh. A line feature based slam with low grade range sensors using geometric constraints and active exploration for mobile robot. *Autonomous Robots*, 24:13–27, 2008.

[43] Song Zhang, Shili Zhao, Dong An, Jincun Liu, He Wang, Yu Feng, Daoliang Li, and Ran Zhao. Visual slam for underwater vehicles: A survey. *Computer Science Review*, 46:100510, 2022.

[44] Xiaotian Wang, Xinnan Fan, Pengfei Shi, Jianjun Ni, and Zhongkai Zhou. An overview of key slam technologies for underwater scenes. *Remote Sensing*, 15(10):2496, 2023.

[45] Adrian Manzanilla, Sergio Reyes, Miguel Garcia, Diego Mercado, and Rogelio Lozano. Autonomous navigation for unmanned underwater vehicles: Real-time experiments using computer vision. *IEEE Robotics and Automation Letters*, 4(2):1351–1356, 2019. doi: 10.1109/LRA.2019.2895272.

[46] Arnaud Tanguy. *Visual SLAM for humanoid robot localization and closed-loop control*. Theses, Université Montpellier, November 2018. URL `https://theses.hal.science/tel-02147610`.

[47] Raluca Scona, Simona Nobili, Yvan R. Petillot, and Maurice Fallon. Direct visual slam fusing proprioception for a humanoid robot. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1419–1426, 2017. doi: 10.1109/IROS.2017.8205943.

[48] Raluca Scona, Simona Nobili, Yvan R Petillot, and Maurice Fallon. Direct visual slam fusing proprioception for a humanoid robot. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1419–1426. IEEE, 2017.

[49] Olivier Stasse, Andrew J Davison, Ramzi Sellaouti, and Kazuhito Yokoi. Real-time 3d slam for humanoid robot considering pattern generator information. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 348–355. IEEE, 2006.

[50] Tim Bailey Hugh Durrant-Whyte. Simultaneous localisation and mapping (slam):part i the essential algorithms, 2006. URL `https://people.eecs.berkeley.edu/~pabbeel/cs287-fa09/readings/Durrant-Whyte_Bailey_SLAM-tutorial-I.pdf`. `https://people.eecs.berkeley.edu/~pabbeel/cs287-fa09/readings/Durrant-Whyte_Bailey_SLAM-tutorial-I.pdf`.

[51] Khalid Yousif, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. An overview to visual odometry and visual slam: Applications to mobile robotics. *Intelligent Industrial Systems*, 1:289–311, 2015. URL `https://api.semanticscholar.org/CorpusID:131102208`.

[52] Amir Vedadi, Aghil Yousefi-Koma, Parsa Yazdankhah, and Amin Mozayyan. Comparative evaluation of rgb-d slam methods for humanoid robot localization and mapping. *2023 11th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pages 807–812, 2023. URL https://api.semanticscholar.org/CorpusID:266818546.

[53] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[54] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: binary robust independent elementary features. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15560-X, 978-3-642-15560-4. URL http://dl.acm.org/citation.cfm?id=1888089.1888148.

[55] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, 2011. doi: 10.1109/ICCV.2011.6126542.

[56] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, 2012. doi: 10.1109/CVPR.2012.6247715.

[57] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, November 2011. doi: 10.1109/ICCV.2011.6126544. URL https://ieeexplore.ieee.org/document/6126544/.

[58] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[59] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. ISSN 1077-3142. doi: 10.1016/j.cviu.2007.09.014. URL http://www.sciencedirect.com/science/article/pii/S1077314207001555. Similarity Matching in Computer Vision and Multimedia.

[60] Jiexiong Tang, Ludvig Ericson, John Folkesson, and Patric Jensfelt. Gcnv2: Efficient correspondence prediction for real-time slam. *IEEE Robotics and Automation Letters*, 4(4): 3505–3512, 2019. doi: 10.1109/LRA.2019.2927954.

[61] Dongjiang Li, Xuesong Shi, Qiwei Long, Shenghui Liu, Wei Yang, Fangshi Wang, Qi Wei, and Fei Qiao. Dxslam: A robust and efficient visual slam system with deep features. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4958–4965, 2020. doi: 10.1109/IROS45743.2020.9340907.

[62] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 337–33712, 2018. doi: 10.1109/CVPRW.2018.00060.

[63] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *Advances in neural information processing systems*, 31, 2018.

[64] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129:23–79, 2021.

[65] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[66] Xiao Xin Lu. A review of solutions for perspective-n-point problem in camera pose estimation. *Journal of Physics: Conference Series*, 1087(5):052009, sep 2018. doi: 10.1088/1742-6596/1087/5/052009. URL `https://dx.doi.org/10.1088/1742-6596/1087/5/052009`.

[67] Alvaro Parra Bustos, Tat-Jun Chin, Anders Eriksson, and Ian Reid. Visual slam: Why bundle adjust? In *2019 international conference on robotics and automation (ICRA)*, pages 2385–2391. IEEE, 2019.

[68] Muhammet Fatih Aslan, Akif Durdu, Abdullah Yusefi, Kadir Sabanci, and Cemil Sungur. A tutorial: Mobile robotics, slam, bayesian filter, keyframe bundle adjustment and ros applications. *Robot Operating System (ROS) The Complete Reference (Volume 6)*, pages 227–269, 2021.

[69] Luca Di Giammarino, Emanuele Giacomini, Leonardo Brizi, Omar Salem, and Giorgio Grisetti. Photometric lidar and rgb-d bundle adjustment. *IEEE Robotics and Automation Letters*, 8(7):4362–4369, 2023. doi: 10.1109/LRA.2023.3281907.

[70] Jorge J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In G.A. Watson, editor, *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, pages 105–116. Springer Berlin Heidelberg, 1978.

[71] Álvaro Parra Bustos, Tat-Jun Chin, Anders Eriksson, and Ian Reid. Visual slam: Why bundle adjust? In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2385–2391, 2019. doi: 10.1109/ICRA.2019.8793749.

[72] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[73] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3727–3737, October 2023.

[74] Zheng Liu and Fu Zhang. Balm: Bundle adjustment for lidar mapping. *IEEE Robotics and Automation Letters*, 6(2):3184–3191, 2021. doi: 10.1109/LRA.2021.3062815.

[75] Liang Zhao, Shoudong Huang, Lei Yan, Jack Jianguo Wang, Gibson Hu, and Gamini Dissanayake. Large-scale monocular slam by local bundle adjustment and map joining. In *2010 11th International Conference on Control Automation Robotics & Vision*, pages 431–436. IEEE, 2010.

[76] Duncan Frost, Victor Prisacariu, and David Murray. Recovering stable scale in monocular slam using object-supplemented bundle adjustment. *IEEE Transactions on Robotics*, 34(3): 736–747, 2018.

[77] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss. OverlapNet: Loop Closing for LiDAR-based SLAM. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[78] Mathieu Labbe and François Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666. IEEE, 2014.

[79] Yunge Cui, Xieyuanli Chen, Yinlong Zhang, Jiahua Dong, Qingxiao Wu, and Feng Zhu. Bow3d: Bag of words for real-time loop closing in 3d lidar slam. *IEEE Robotics and Automation Letters*, 8(5):2828–2835, 2023. doi: 10.1109/LRA.2022.3221336.

[80] Yunge Cui, Xieyuanli Chen, Yinlong Zhang, Jiahua Dong, Qingxiao Wu, and Feng Zhu. Bow3d: Bag of words for real-time loop closing in 3d lidar slam. *IEEE Robotics and Automation Letters*, 8(5):2828–2835, 2022.

[81] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[82] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[83] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[84] Pablo Fernández Alcantarilla, José Yebes, Javier Almazán, and Luis Bergasa. On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. pages 1290–1297, 05 2012. ISBN 978-1-4673-1403-9. doi: 10.1109/ICRA.2012.6224690.

[85] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 722–729 vol.2, 1999.

[86] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[87] Mariano Jaimez, Christian Kerl, Javier González, and Daniel Cremers. Fast odometry and scene flow from rgb-d cameras based on geometric clustering. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3992–3999, 2017.

[88] Deok-Hwa Kim and Jong-Hwan Kim. Effective background model-based rgb-d dense visual odometry in a dynamic environment. *IEEE Transactions on Robotics*, 32(6):1565–1573, 2016. doi: 10.1109/TRO.2016.2609395.

[89] Shile Li and Dongheui Lee. Rgb-d slam in dynamic environments using static point weighting. *IEEE Robotics and Automation Letters*, PP:1–1, 07 2017. doi: 10.1109/LRA.2017.2724759.

[90] Yuxiang Sun, Ming Liu, and Max Meng. Improving rgb-d slam in dynamic environments: A motion removal approach. *Robotics and Autonomous Systems*, 89, 11 2016. doi: 10.1016/j.robot.2016.11.012.

[91] Runzhi Wang, Wenhui Wan, Yongkang Wang, and Kaichang Di. A new rgb-d slam method with moving object detection for dynamic indoor scenes. *Remote Sensing*, 11:1143, 2019.

[92] Berta Bescos, Jose Facil, Javier Civera, and Jose Neira. Dynaslam: Tracking, mapping and inpainting in dynamic scenes. 3:1–1, 07 2018. doi: 10.1109/LRA.2018.2860039.

[93] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[94] Hanjie Liu, Guoliang Liu, Guohui Tian, Shi-Qing Xin, and Ze Ji. Visual slam based on dynamic object removal. *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 596–601, 2019.

[95] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[96] Chao Yu, Zuxin Liu, Xinjun Liu, Fugui Xie, Yi Yang, Qi Wei, and Fei Qiao. DS-SLAM: A semantic visual SLAM towards dynamic environments. *CoRR*, abs/1809.08379, 2018.

[97] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. URL `http://dblp.uni-trier.de/db/journals/corr/corr1511.html#BadrinarayananK15`.

[98] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. Flowfusion: Dynamic dense rgb-d slam based on optical flow, 03 2020.

[99] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *CoRR*, abs/1709.02371, 2017.

[100] Andreas Wedel, Thomas Brox, Tobi Vaudrey, Clemens Rabe, Uwe Franke, and Daniel Cremers. Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision*, 95:29–51, 10 2011. doi: 10.1007/s11263-010-0404-0.

[101] Antoine Letouzey, Benjamin Petit, and Edmond Boyer. Scene flow from depth and color images. In *British Machine Vision Conference*, 2011. URL `https://api.semanticscholar.org/CorpusID:560203`.

[102] E. Herbst, X. Ren, and D. Fox. Rgb-d flow: Dense 3-d motion estimation using color and depth. In *2013 IEEE International Conference on Robotics and Automation*, pages 2276–2282, 2013.

[103] J. Quiroga, F. Devernay, and J. Crowley. Local/global scene flow estimation. In *2013 IEEE International Conference on Image Processing*, pages 3850–3854, 2013.

[104] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers. A primal-dual framework for real-time dense rgb-d scene flow. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 98–104, 2015.

[105] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3614–3622. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00373.

[106] Yi-Ling Qiao, Lin Gao, Yu-Kun Lai, Fang-Lue Zhang, Mingzhe Yuan, and Shihong Xia. Sf-net: Learning scene flow from RGB-D images with cnns. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 281. BMVA Press, 2018.

[107] René Schuster, Oliver Wasenmüller, Christian Unger, Georg Kuschk, and Didier Stricker. Sceneflowfields++: Multi-frame matching, visibility prediction, and robust interpolation for scene flow estimation. *International Journal of Computer Vision*, 11 2019. doi: 10.1007/s11263-019-01258-1.

[108] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *Advances in Neural Information Processing Systems*, 34:7838–7851, 2021.

[109] Nathaniel Chodosh, Deva Ramanan, and Simon Lucey. Re-evaluating lidar scene flow. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*, pages 5993–6003. IEEE, 2024. doi: 10.1109/WACV57701.2024.00590. URL `https://doi.org/10.1109/WACV57701.2024.00590`.

[110] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 529–537, 2019.

[111] René Schuster, Christian Bailer, Oliver Wasenmüller, and Didier Stricker. Combining stereo disparity and optical flow for basic scene flow. In *Commercial Vehicle Technology Symposium (CVT) 2018 |. Commercial Vehicle Technology Symposium (CVT-18), March 13-15, Kaiserslautern, Germany*. Commercial Vehicle Alliance, Springer, 2018.

[112] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. pages 1647–1655, 07 2017. doi: 10.1109/CVPR.2017.179.

[113] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL `http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15`.

[114] Raul Mur-Artal and Juan Tardos. Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras. *IEEE Transactions on Robotics*, PP, 10 2016. doi: 10.1109/TRO.2017.2705103.

[115] Berta Bescos, José M. Fácil, Javier Civera, and José Neira. DynaSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes. 2018.

[116] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, 30(1):158–176, 2014. doi: 10.1109/TRO.2013.2277549.

[117] Fabian Schenk and Friedrich Fraundorfer. Reslam: A real-time robust edge-based slam system. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 154–160, 2019. doi: 10.1109/ICRA.2019.8794462.

[118] Juan José Tarrio and Sol Pedre. Realtime edge-based visual odometry for a monocular camera. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 702–710, 2015. doi: 10.1109/ICCV.2015.87.

[119] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francese Moreno-Noguer. Pl-slam: Real-time monocular visual slam with points and lines. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4503–4508, 2017. doi: 10.1109/ICRA.2017.7989522.

[120] Qiuyuan Wang, Zike Yan, Junqiu Wang, Fei Xue, Wei Ma, and Hongbin Zha. Line flow based simultaneous localization and mapping. *IEEE Transactions on Robotics*, 37(5):1416–1432, 2021. doi: 10.1109/TRO.2021.3061403.

[121] Hyunjun Lim, Jinwoo Jeon, and Hyun Myung. Uv-slam: Unconstrained line-based slam using vanishing points for structural mapping. *IEEE Robotics and Automation Letters*, 7(2): 1518–1525, 2022. doi: 10.1109/LRA.2022.3140816.

[122] Andrew P. Gee and W. Mayol-Cuevas. Real-time model-based slam using line segments. In *International Symposium on Visual Computing*, 2006. URL `https://api.semanticscholar.org/CorpusID:17489512`.

[123] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer. PL-SLAM: Real-Time Monocular Visual SLAM with Points and Lines. In *International Conference in Robotics and Automation*, 2017.

[124] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *International Journal of Robotic Research - IJRR*, 31:647–663, 04 2012. doi: 10.1177/0278364911434148.

[125] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*, September 2014.

[126] Alejo Concha, Giuseppe Loianno, Vijay Kumar, and Javier Civera. Visual-inertial direct slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1331–1338, 2016. doi: 10.1109/ICRA.2016.7487266.

[127] Vladyslav C. Usenko, Jakob J. Engel, J. Stückler, and Daniel Cremers. Direct visual-inertial odometry with stereo cameras. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1885–1892, 2016. URL `https://api.semanticscholar.org/CorpusID:5689214`.

[128] Dominykas Strazdas, Jan Hintz, Anna-Maria Felßberg, and Ayoub Al-Hamadi. Robots and wizards: An investigation into natural human–robot interaction. *IEEE Access*, 8:207635–207642, 2020. doi: 10.1109/ACCESS.2020.3037724.

[129] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI - Künstliche Intelligenz*, 24:345–348, 2010.

[130] Yoshikatsu Nakajima, Keisuke Tateno, Federico Tombari, and Hideo Saito. Fast and accurate semantic mapping through geometric-based incremental segmentation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 385–392, 2018. doi: 10.1109/IROS.2018.8593993.

[131] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4628–4635, 2017. doi: 10.1109/ICRA.2017.7989538.

[132] Yoshikatsu Nakajima and Hideo Saito. Efficient object-oriented semantic mapping with object detector. *IEEE Access*, 7:3206–3213, 01 2019. doi: 10.1109/ACCESS.2018.2887022.

[133] T. Nguyen, B. Michaelis, A. Al-Hamadi, M. Tornow, and M. Meinecke. Stereo-camera-based urban environment perception using occupancy grid and object tracking. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):154–165, 2012. doi: 10.1109/TITS.2011.2165705.

[134] Asif Iqbal and Nicholas Gans. Data association and localization of classified objects in visual slam. *Journal of Intelligent & Robotic Systems*, 100:1–18, 10 2020. doi: 10.1007/s10846-020-01189-x.

[135] Taih'u Pire, Javier Corti, and Guillermo Grinblat. Online Object Detection and Localization on Stereo Visual SLAM System. *Journal of Intelligent & Robotic Systems*, August 2019. ISSN 1573-0409. doi: 10.1007/s10846-019-01074-2.

[136] Lachlan Nicholson, Michael Milford, and Niko Sunderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, PP:1–1, 08 2018. doi: 10.1109/LRA.2018.2866205.

[137] Renato Salas-Moreno, Richard Newcombe, Hauke Strasdat, Paul Kelly, and Andrew Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 06 2013. doi: 10.1109/CVPR.2013.178.

[138] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, PP:1–14, 05 2019. doi: 10.1109/TRO.2019.2909168.

[139] J. Mccormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger. Fusion++: Volumetric object-level slam. In *2018 International Conference on 3D Vision (3DV)*, pages 32–41, 2018. doi: 10.1109/3DV.2018.00015.

[140] Lukas Bernreiter, Abel Gawel, Hannes Sommer, Juan Nieto, Roland Siegwart, and Cesar Cadena. Multiple hypothesis semantic mapping for robust data association. *IEEE Robotics and Automation Letters*, PP:3255 – 3262, 06 2019. doi: 10.1109/LRA.2019.2925756.

[141] Jimmy Li, David Meger, and Gregory Dudek. Semantic mapping for view-invariant relocalization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7108–7115, 2019. doi: 10.1109/ICRA.2019.8793624.

[142] Renato Martins, Dhiego Bersan, Mario F. M. Campos, and Erickson R. Nascimento. Extending maps with semantic and contextual object information for robot navigation: a learning-based framework using visual and depth cues. *Journal of Intelligent & Robotic Systems*, 99(3-4): 555–569, Feb 2020. ISSN 1573-0409. doi: 10.1007/s10846-019-01136-5. URL `http://dx.doi.org/10.1007/s10846-019-01136-5`.

[143] Sean L. Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J. Pappas. Probabilistic data association for semantic slam. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, 2017. doi: 10.1109/ICRA.2017.7989203.

[144] K. Doherty, David Baxter, Edward Schneeweiss, and John F. Leonard. Probabilistic data association via mixture models for robust semantic slam. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1098–1104, 2020.

[145] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. A survey of modern deep learning based object detection models. *Digital Signal Processing*, 126:103514, 2022. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2022.103514. URL `https://www.sciencedirect.com/science/article/pii/S1051200422001312`.

[146] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 06 2015. doi: 10.1109/TPAMI.2016.2577031.

[147] Glenn Jocher. by Ultralytics, 5 2020. URL `https://github.com/ultralytics/yolov5`.

[148] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. doi: 10.1109/CVPR.2016.91.

[149] Alexey Bochkovskiy, Chien-Yao Wang, and H. Liao. Yolov4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020.

[150] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58452-8.

[151] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 580–587, June 2014. doi: 10.1109/CVPR.2014.81. URL `https://ieeexplore.ieee.org/abstract/document/6909475/`.

[152] Ross Girshick. Fast r-cnn. *CoRR*, abs/1504.08083, 2015. URL `http://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf`.

[153] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV (3)*, volume 8691 of *Lecture Notes in Computer Science*, pages 346–361. Springer, 2014. ISBN 978-3-319-10577-2. URL `http://dblp.uni-trier.de/db/conf/eccv/eccv2014-3.html#HeZR014`.

[154] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2017. URL `https://api.semanticscholar.org/CorpusID:206596979`.

[155] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. URL `http://arxiv.org/abs/1405.0312`. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.

[156] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2017. doi: 10.1109/AVSS.2017.8078516.

[157] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. ISBN 0521540518.

[158] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. URL `https://www.ros.org`.

[159] Mathieu Labbe and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation: LabbÉ and michaud. *Journal of Field Robotics*, 36, 10 2018. doi: 10.1002/rob.21831.

[160] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[161] F. Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.

[162] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, Oct 2017. ISSN 1552-3098. doi: 10.1109/TRO.2017.2705103.

[163] Andrea Veronese, M. Racca, Roel Pieters, and Ville Kyrki. Probabilistic mapping of human visual attention from head pose estimation. *Frontiers Robotics AI*, 4:53, 2017.

[164] Sileye Ba and Jean-Marc Odobez. Recognizing visual focus of attention from head pose in natural meetings. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 39:16–33, 10 2008. doi: 10.1109/TSMCB.2008.927274.

[165] Tripti Singh, Mohan Mohadikar, Shilpa Gite, Shruti Patil, Biswajeet Pradhan, and Abdullah Alamri. Attention span prediction using head-pose estimation with deep neural networks. *IEEE Access*, 9:142632–142643, 2021. doi: 10.1109/ACCESS.2021.3120098.

[166] Feng-Ju Chang, A. Tran, Tal Hassner, Iacopo Masi, Ramakant Nevatia, and Gérard G. Medioni. Faceposenet: Making a case for landmark-free face alignment. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1599–1608, 2017.

[167] Li Wei and Eung-Joo Lee. Multi-pose face recognition using head pose estimation and pca approach. *JDCTA*, 4:112–122, 02 2010. doi: 10.4156/jdcta.vol4.issue1.12.

[168] Amit Kumar, Azadeh Alavi, and Rama Chellappa. Kepler: Simultaneous estimation of keypoints and 3d pose of unconstrained faces in a unified framework by learning efficient h-cnn regressors. *Image and Vision Computing*, 79:49–62, 2018. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2018.09.009. URL `https://www.sciencedirect.com/science/article/pii/S0262885618301549`.

[169] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:121–135, 2019.

[170] Erik Murphy-Chutorian, Anup Doshi, and Mohan Manubhai Trivedi. Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In *2007 IEEE Intelligent Transportation Systems Conference*, pages 709–714, 2007. doi: 10.1109/ITSC.2007.4357803.

[171] Sumit Jha and Carlos Busso. Estimation of driver's gaze region from head position and orientation using probabilistic confidence regions. *IEEE Transactions on Intelligent Vehicles*, PP:1–1, 01 2022. doi: 10.1109/TIV.2022.3141071.

[172] Guido Borghi, Marco Venturelli, Roberto Vezzani, and Rita Cucchiara. Poseidon: Face-from-depth for driver pose estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5494–5503, 2017. doi: 10.1109/CVPR.2017.583.

[173] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):300–311, 2010. doi: 10.1109/TITS.2010.2044241.

[174] Marcel C. Buehler, Abhimitra Meka, Gengyan Li, Thabo Beeler, and Otmar Hilliges. Varitex: Variational neural face textures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[175] Dominykas Strazdas, Jan Hintz, and Ayoub Al-Hamadi. Robo-hud: Interaction concept for contactless operation of industrial cobotic systems. *Applied Sciences*, 11(12), 2021. ISSN 2076-3417. doi: 10.3390/app11125366. URL `https://www.mdpi.com/2076-3417/11/12/5366`.

[176] Andre Gaschler, Kerstin Huth, Manuel Giuliani, Ingmar Kessler, Jan Peter de Ruiter, and Alois Knoll. Modelling state of interaction from head poses for social human-robot interaction. In *HRI 2012*, 2012.

[177] Mary Ellen Foster, Andre Gaschler, and Manuel Giuliani. Automatically classifying user engagement for dynamic multi-party human–robot interaction. *International Journal of Social Robotics*, 9, 11 2017. doi: 10.1007/s12369-017-0414-y.

[178] T. Vatahska, Maren Bennewitz, and Sven Behnke. Feature-based head pose estimation from images. *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 330–335, 2007. URL `https://api.semanticscholar.org/CorpusID:971626`.

[179] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017.

[180] Philipp Werner, Frerk Saxen, and Ayoub Al-Hamadi. Landmark based head pose estimation benchmark and method. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3909–3913, 2017. doi: 10.1109/ICIP.2017.8297015.

[181] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014. doi: 10.1109/CVPR.2014.241.

[182] Aryaman Gupta, Kalpit Thakkar, Vineet Gandhi, and P J Narayanan. Nose, eyes and ears: Head pose estimation by locating facial keypoints. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1977–1981, 2019. doi: 10.1109/ICASSP.2019.8683503.

[183] Shiqi Li, Chi Xu, and Ming Xie. A robust o(n) solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1444–1450, 2012.

[184] Shay Ohayon and E. Rivlin. Robust 3d head tracking using camera pose estimation. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 1063–1066, 2006. doi: 10.1109/ICPR.2006.999.

[185] Seong G. Kong and Ralph Oyini Mbouna. Head pose estimation from a 2d face image using 3d face morphing with depth parameters. *IEEE Transactions on Image Processing*, 24:1801–1808, 2015. URL `https://api.semanticscholar.org/CorpusID:15392290`.

[186] Cho-Ying Wu, Qiangeng Xu, and Ulrich Neumann. Synergy between 3dmm and 3d landmarks for accurate 3d facial geometry. In *2021 International Conference on 3D Vision (3DV)*, 2021.

[187] H. Li, B. Wang, Y. Cheng, M. Kankanhalli, and R. T. Tan. Dsfnet: Dual space fusion network for occlusion-robust 3d dense face alignment. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4531–4540, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.00440. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.00440`.

[188] T. Martyniuk, O. Kupyn, Y. Kurlyak, I. Krashenyi, J. Matas, and V. Sharmanska. Dad-3dheads: A large-scale dense, accurate and diverse dataset for 3d head alignment from a single image. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20910–20920, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.02027. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.02027`.

[189] Roberto Valle, José Miguel Buenaposada, and Luis Baumela. Multi-task head pose estimation in-the-wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:2874–2881, 2021.

[190] Xavier P. Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *2013 IEEE International Conference on Computer Vision*, pages 1513–1520, 2013. doi: 10.1109/ICCV.2013.191.

[191] Yue Wu and Qiang Ji. Robust facial landmark detection under significant head poses and occlusion. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3658–3666, 2015. doi: 10.1109/ICCV.2015.417.

[192] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2155–215509, 2018.

[193] Heng-Wei Hsu, Tung-Yu Wu, Sheng Wan, Wing Hung Wong, and Chen-Yi Lee. Quatnet: Quaternion-based head pose estimation with multiregression loss. *IEEE Transactions on Multimedia*, 21(4):1035–1046, 2019. doi: 10.1109/TMM.2018.2866770.

[194] Yijun Zhou and James Gregson. Whenet: Real-time fine-grained estimation for wide range head pose. In *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020. URL `https://www.bmvc2020-conference.com/assets/papers/0907.pdf`.

[195] Luís Torgo and João Gama. Regression by classification. In Díbio L. Borges and Celso A. A. Kaestner, editors, *Advances in Artificial Intelligence*, pages 51–60, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg. ISBN 978-3-540-70742-4.

[196] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu Chuang. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[197] Zhiwen Cao, Zongcheng Chu, Dongfang Liu, and Yingjie Chen. A vector-based representation to enhance head pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1188–1197, January 2021.

[198] Hai Liu, Shuai Fang, Zhaoli Zhang, Duantengchuan Li, Ke Lin, and Jiazhang Wang. Mfdnet: Collaborative poses perception and matrix fisher distribution for head pose estimation. *IEEE Transactions on Multimedia*, 24:2449–2460, 2022. doi: 10.1109/TMM.2021.3081873.

[199] Zhaoxiang Liu, Zezhou Chen, Jinqiang Bai, Shaohua Li, and Shiguo Lian. Facial pose estimation by deep learning from label distributions. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1232–1240, 2019. doi: 10.1109/ICCVW.2019.00156.

[200] Hao Zhang, Mengmeng Wang, Yong Liu, and Yi Yuan. Fdn: Feature decoupling network for head pose estimation. In *AAAI*, 2020.

[201] Nima Aghli and Eraldo Ribeiro. A data-driven approach to improve 3d head-pose estimation. In *Advances in Visual Computing: 16th International Symposium, ISVC 2021, Virtual Event, October 4-6, 2021, Proceedings, Part I*, page 546–558, Berlin, Heidelberg, 2021. Springer-Verlag. ISBN 978-3-030-90438-8. doi: 10.1007/978-3-030-90439-5_43.

[202] Bin Huang, Renwen Chen, Wang Xu, and Qinbang Zhou. Improving head pose estimation using two-stage ensembles with top-k regression. *Image Vis. Comput.*, 93:103827, 2020.

[203] Hoang Nguyen Viet, Linh Nguyen Viet, Tuan Nguyen Dinh, Duc Tran Minh, and Long Tran Quoc. Simultaneous face detection and 360 degree head pose estimation. *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–7, 2021.

[204] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5738–5746, 2019.

[205] Bima Sena Bayu Dewantara and Jun Miura. The aisl head orientation database and preliminary evaluations. In *2015 International Electronics Symposium (IES)*, pages 140–144, 2015. doi: 10.1109/ELECSYM.2015.7380830.

[206] Mohamed Selim, Ahmet Firintepe, Alain Pagani, and Didier Stricker. Autopose: Large-scale automotive driver head pose and gaze dataset with deep head orientation baseline. In *VISIGRAPP*, 2020. URL `https://api.semanticscholar.org/CorpusID:211162192`.

[207] Steffen Walter, Sascha Gruss, Hagen Ehleiter, Junwen Tan, Harald C. Traue, Philipp Werner, Ayoub Al-Hamadi, Stephen Crawcour, Adriano O. Andrade, and Gustavo Moreira da Silva. The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system. In *2013 IEEE International Conference on Cybernetics (CYBCO)*, pages 128–131, 2013. doi: 10.1109/CYBConf.2013.6617456.

[208] M. La Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000. doi: 10.1109/34.845375.

[209] Wen Gao, Bo Cao, Shiguang Shan, Xilin Chen, Delong Zhou, Xiaohua Zhang, and Debin Zhao. The cas-peal large-scale chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 38(1):149–161, 2008. doi: 10.1109/TSMCA.2007.909557.

[210] CCNUHead. Ccnu—head dataset. `https://universe.roboflow.com/ccnuhead/ccnu-head`, jan 2023. URL `https://universe.roboflow.com/ccnuhead/ccnu-head`. visited on 2023-11-27.

[211] Markus Roth and Dariu M. Gavrila. Dd-pose - a large-scale driver head pose benchmark. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 927–934, 2019. doi: 10.1109/IVS.2019.8814103.

[212] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and S. Li. Face alignment across large poses: A 3d solution. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 146–155, 2016.

[213] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 787–796, 2015. doi: 10.1109/CVPR.2015.7298679.

[214] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. van Gool. Random forests for real time 3d face analysis. *International Journal of Computer Vision*, 101(3):437–458, 2013. doi: 10.1007/s11263-012-0549-0.

[215] *Panoptic Studio: A Massively Multiview System for Social Motion Capture*, 2015.

[216] Peter N. Belhumeur, David W. Jacobs, David J. Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 35, pages 2930–2940, December 2013.

[217] Adrian Bulat and Georgios Tzimiropoulos. Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge. In *ECCV Workshops*, 2016.

[218] Vuong Le, Jonathan Brandt, Zhe L. Lin, Lubomir D. Bourdev, and Thomas S. Huang. Interactive facial feature localization. In *ECCV*, 2012.

[219] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. A semi-automatic methodology for facial landmark annotation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 896–903, 2013. doi: 10.1109/CVPRW.2013.132.

[220] Peter M. Roth Martin Koestinger, Paul Wohlhart and Horst Bischof. Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. In *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[221] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23:1499–1503, 2016.

[222] Andrea Asperti and Daniele Filippini. Deep learning for head pose estimation: A survey. *SN Comput. Sci.*, 4(4), apr 2023. doi: 10.1007/s42979-023-01796-z. URL `https://doi.org/10.1007/s42979-023-01796-z`.

[223] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[224] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[225] Donggen Dai, Wangkit Wong, and Zhuojun Chen. Rankpose: Learning generalised feature with rank supervision for head pose estimation. In *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020. URL `https://www.bmvc2020-conference.com/assets/papers/0401.pdf`.

[226] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[227] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. ISSN 2078-2489. doi: 10.3390/info11020125. URL https://www.mdpi.com/2078-2489/11/2/125.

[228] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li. Face alignment in full pose range: A 3d total solution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(1):78–92, jan 2019. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2778152. URL https://doi.org/10.1109/TPAMI.2017.2778152.

[229] Vítor Albiero, Xingyu Chen, Xi Yin, Guan Pang, and Tal Hassner. img2pose: Face alignment and detection via 6dof, face pose estimation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7613–7623, 2021. doi: 10.1109/CVPR46437.2021.00753.

[230] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6912–6921, 2019.

[231] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 334–352, 2018.

[232] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):162–175, 2017.

[233] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

[234] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 2299–2308. IEEE, 2017.

[235] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020.

[236] Zhaokang Chen and Bertram E Shi. Appearance-based gaze estimation using dilated-convolutions. In *Asian Conference on Computer Vision*, pages 309–324. Springer, 2018.

[237] Yihua Cheng, Shiyao Huang, Fei Wang, Chen Qian, and Feng Lu. A coarse-to-fine adaptive network for appearance-based gaze estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10623–10630, 2020.

[238] Gang Liu, Yu Yu, Kenneth A. Funes Mora, and Jean-Marc Odobez. A differential approach for gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3): 1092–1099, 2021. doi: 10.1109/TPAMI.2019.2957373.

[239] Yihua Cheng, Yiwei Bao, and Feng Lu. Puregaze: Purifying gaze feature for generalizable gaze estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 436–443, 2022.

[240] Jun O Oh, Hyung Jin Chang, and Sang-Il Choi. Self-attention with convolution and deconvolution for efficient eye gaze estimation from a full face image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4992–5000, 2022.

[241] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[242] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[243] Yiwei Bao, Yunfei Liu, Haofei Wang, and Feng Lu. Generalizing gaze estimation with rotation consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4207–4216, 2022.

[244] Minjie Cai, Feng Lu, and Yoichi Sato. Generalizing hand segmentation in egocentric videos with uncertainty-guided model adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14392–14401, 2020.

[245] Yunfei Liu, Ruicong Liu, Haofei Wang, and Feng Lu. Generalizing gaze estimation with outlier-guided collaborative adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3835–3844, 2021.

[246] Yaoming Wang, Yangzhou Jiang, Jin Li, Bingbing Ni, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Teng Li. Contrastive regression for domain adaptation on gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19376–19385, June 2022.

[247] Rakshit Kothari, Shalini De Mello, Umar Iqbal, Wonmin Byeon, Seonwook Park, and Jan Kautz. Weakly-supervised physically unconstrained gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9980–9989, 2021.

[248] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

[249] Marc-André Fiedler, Philipp Werner, Aly Khalifa, and Ayoub Al-Hamadi. Sfpd: Simultaneous face and person detection in real-time for human–robot interaction. *Sensors*, 21(17), 2021. ISSN 1424-8220. doi: 10.3390/s21175918. URL `https://www.mdpi.com/1424-8220/21/17/5918`.

[250] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[251] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[252] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021.

[253] An-An Liu, Yu-Ting Su, Wei-Zhi Nie, and Mohan Kankanhalli. Hierarchical clustering multi-task learning for joint human action grouping and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):102–114, 2017. doi: 10.1109/TPAMI.2016.2537337.

[254] Huy Hoang Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8, 2019.

[255] Amit Kumar, Azadeh Alavi, and Rama Chellappa. Kepler: Keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors. *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 258–265, 2017.

[256] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 17–24, 2017. doi: 10.1109/FG.2017.137.

[257] Hu Han, Anil K. Jain, Fang Wang, S. Shan, and Xilin Chen. Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2597–2609, 2018.

[258] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 94–108. Springer, 2014.

[259] Max Ehrlich, Timothy J Shields, Timur Almaev, and Mohamed R Amer. Facial attributes classification using multi-task representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 47–55, 2016.

[260] Shreya Ghosh, Munawar Hayat, Abhinav Dhall, and Jarrod Knibbe. Mtgls: Multi-task gaze estimation with limited supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3223–3234, 2022.

[261] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. Appearance-based gaze estimation with deep learning: A review and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2024. doi: 10.1109/TPAMI.2024.3393571.

[262] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Revisiting data normalization for appearance-based gaze estimation. In *Proceedings of the 2018 ACM symposium on eye tracking research & applications*, pages 1–9, 2018.

[263] Yu Yu, Gang Liu, and Jean-Marc Odobez. Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11937–11946, 2019.

[264] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 2299–2308. IEEE, 2017.

[265] Jun O Oh, Hyung Jin Chang, and Sang-Il Choi. Self-attention with convolution and deconvolution for efficient eye gaze estimation from a full face image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4992–5000, 2022.

[266] Kang Wang, Rui Zhao, Hui Su, and Qiang Ji. Generalizing eye tracking with bayesian adversarial learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11907–11916, 2019.

[267] Zidong Guo, Zejian Yuan, Chong Zhang, Wanchao Chi, Yonggen Ling, and Shenghao Zhang. Domain adaptation gaze estimation by embedding with prediction consistency. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[268] Jie Shen, Stefanos Zafeiriou, Grigoris G Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 50–58, 2015.

[269] Roxane J. Itier and Magali Batty. Neural bases of eye and gaze processing: The core of social cognition. *Neuroscience & Biobehavioral Reviews*, 33(6):843–863, 2009. ISSN 0149-7634. doi: https://doi.org/10.1016/j.neubiorev.2009.02.004. URL https://www.sciencedirect.com/science/article/pii/S0149763409000207.

[270] Sophie Wohltjen and Thalia Wheatley. Eye contact marks the rise and fall of shared attention in conversation. *Proceedings of the National Academy of Sciences*, 118(37):e2106645118, 2021. doi: 10.1073/pnas.2106645118. URL https://www.pnas.org/doi/abs/10.1073/pnas.2106645118.

[271] Heidi Mauersberger, Till Kastendieck, and Ursula Hess. I looked at you, you looked at me, i smiled at you, you smiled at me—the impact of eye contact on emotional mimicry. *Frontiers in Psychology*, 13, 2022.

[272] Sophie Wohltjen and Thalia Wheatley. Eye contact marks the rise and fall of shared attention in conversation. *Proceedings of the National Academy of Sciences*, 118(37):e2106645118, 2021.

[273] Helena Kiilavuori, Veikko Sariola, Mikko J. Peltola, and Jari K. Hietanen. Making eye contact with a robot: Psychophysiological responses to eye contact with a human and with a humanoid robot. *Biological Psychology*, 158:107989, 2021. ISSN 0301-0511. doi: https://doi.org/10.1016/j.biopsycho.2020.107989. URL https://www.sciencedirect.com/science/article/pii/S0301051120301496.

[274] Yanxia Zhang, Jonas Beskow, and Hedvig Kjellström. Look but don't stare: Mutual gaze interaction in social robots. In *International Conference on Software Reuse*, 2017.

[275] Marwen Belkaid, Kyveli Kompatsiari, Davide De Tommaso, Ingrid Zablith, and Agnieszka Wykowska. Mutual gaze with a robot affects human neural activity and delays decision-making processes. *Science Robotics*, 6(58):eabc5044, 2021. doi: 10.1126/scirobotics.abc5044. URL `https://www.science.org/doi/abs/10.1126/scirobotics.abc5044`.

[276] Kyveli Kompatsiari, Francesca Ciardo, Vadim Tikhanoff, Giorgio Metta, and Agnieszka Wykowska. It's in the eyes: The engaging role of eye contact in hri. *International Journal of Social Robotics*, 13:1–11, 06 2021. doi: 10.1007/s12369-019-00565-4.

[277] Elef Schellen, Francesco Bossi, and Agnieszka Wykowska. Robot gaze behavior affects honesty in human-robot interaction. *Frontiers in Artificial Intelligence*, 4, 2021. ISSN 2624-8212. doi: 10.3389/frai.2021.663190. URL `https://www.frontiersin.org/articles/10.3389/frai.2021.663190`.

[278] Tetsuya Sano, Akishige Yuguchi, Gustavo Alfonso Garcia Ricardez, Jun Takamatsu, Atsushi Nakazawa, and Tsukasa Ogasawara. Evaluating imitation of human eye contact and blinking behavior using an android for human-like communication. *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–6, 2019.

[279] Alap Kshirsagar, Melanie Mei Hsia Lim, Shemar Christian, and Guy Hoffman. Robot gaze behaviors in human-to-robot handovers. *IEEE Robotics and Automation Letters*, 5:6552–6558, 2020.

[280] Chinmaya Mishra and Gabriel Skantze. Knowing where to look: A planning-based architecture to automate the gaze behavior of social robots*. *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1201–1208, 2022.

[281] Shayla Sharmin, Mohammed Moshiul Hoque, S. M. Riazul Islam, Md. Fazlul Kader, and Iqbal H. Sarker. Development of duplex eye contact framework for human-robot inter communication. *IEEE Access*, 9:54435–54456, 2021. doi: 10.1109/ACCESS.2021.3071129.

[282] Matthew K. X. J. Pan, Sungjoon Choi, James Kennedy, Kyna McIntosh, Daniel Campos Zamora, Günter Niemeyer, Joohyung Kim, Alexis Wieland, and David L. Christensen. Realistic and interactive robot gaze. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11072–11078, 2020.

[283] Abolfazl Zaraki, Daniele Mazzei, Manuel Giuliani, and Danilo De Rossi. Designing and evaluating a social gaze-control system for a humanoid robot. *IEEE Transactions on Human-Machine Systems*, 44:157–168, 2014.

[284] Tetsuya Sano, Akishige Yuguchi, Gustavo Alfonso Garcia Ricardez, Jun Takamatsu, Atsushi Nakazawa, and Tsukasa Ogasawara. Evaluating imitation of human eye contact and blinking behavior using an android for human-like communication. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–6, 2019. doi: 10.1109/RO-MAN46459.2019.8956387.

[285] Zhefan Ye, Yin Li, Alireza Fathi, Yi Han, Agata Rozga, Gregory D. Abowd, and James M. Rehg. Detecting eye contact using wearable eye-tracking glasses. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, page 699–704, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312240. doi: 10.1145/2370216.2370368. URL https://doi.org/10.1145/2370216.2370368.

[286] V. Onkhar, P. Bazilinskyy, J.C.J. Stapel, D. Dodou, D. Gavrila, and J.C.F. de Winter. Towards the detection of driver–pedestrian eye contact. *Pervasive Mob. Comput.*, 76(C), sep 2021. ISSN 1574-1192. doi: 10.1016/j.pmcj.2021.101455. URL https://doi.org/10.1016/j.pmcj.2021.101455.

[287] Taylor Mordan, Matthieu Cord, Patrick P'erez, and Alexandre Alahi. Detecting 32 pedestrian attributes for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23:11823–11835, 2020.

[288] Amir Rasouli, Iuliia Kotseruba, and John K. Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 206–213, 2017. doi: 10.1109/ICCVW.2017.33.

[289] Younes Belkada, Lorenzo Bertoni, Romain Caristan, Taylor Mordan, and Alexandre Alahi. Do pedestrians pay attention? eye contact detection in the wild, 2021.

[290] Brian A. Smith, Qi Yin, Steven K. Feiner, and Shree K. Nayar. Gaze locking: Passive eye contact detection for human-object interaction. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, page 271–280, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450322683. doi: 10.1145/2501988.2501994. URL https://doi.org/10.1145/2501988.2501994.

[291] Zhefan Ye, Yin Li, Yun Liu, Chanel Bridges, Agata Rozga, and James M. Rehg. Detecting bids for eye contact using a wearable camera. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 1:1–8, 2015.

[292] Eunji Chong, Elysha Clark-Whitney, Audrey Southerland, Elizabeth Stubbs, Chanel Miller, Eliana L. Ajodan, Melanie R. Silverman, Catherine Lord, Agata Rozga, Rebecca Merrill Jones, and James M. Rehg. Detection of eye contact with deep neural networks is as accurate as human experts. *Nature Communications*, 11, 2020.

[293] Dingwen Zhang, Bo Wang, Gerong Wang, Qiang Zhang, Jiajia Zhang, Jungong Han, and Zheng You. Onfocus detection: identifying individual-camera eye contact from unconstrained images. *Science China Information Sciences*, 65, 2021.

[294] Yu Mitsuzumi, Atsushi Nakazawa, and Toyoaki Nishida. Deep eye contact detector: Robust eye contact bid detection using convolutional neural network. In *British Machine Vision Conference*, 2017.

[295] Martin Koestinger, Paul Wohlhart, Peter M Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In

IEEE International Conference on Computer Vision Workshops, pages 2144–2151. IEEE, 2011.

[296] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*, 47:3–18, 2016. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2016.01.002. URL https://www.sciencedirect.com/science/article/pii/S0262885616000147. 300-W, the First Automatic Facial Landmark Detection in-the-Wild Challenge.

[297] Gary B. Huang Erik Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. Technical Report UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.

[298] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[299] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[300] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5203–5212, 2020.

[301] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[302] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[303] Ahmed A. Abdelrahman, Thorsten Hempel, Aly Khalifa, Ayoub Al-Hamadi, and Laslo Dinges. L2cs-net : Fine-grained gaze estimation in unconstrained environments. In *2023 8th International Conference on Frontiers of Signal Processing (ICFSP)*, pages 98–102, 2023. doi: 10.1109/ICFSP59764.2023.10372944.

[304] Jacob Gildenblat and contributors. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021.

[305] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 2299–2308. IEEE, 2017.

[306] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

[307] Michael Beetz, Raja Chatila, Joachim Hertzberg, and Federico Pecora. Ai reasoning methods for robotics. *Springer Handbook of Robotics*, pages 329–356, 2016.