HAW HAMBURG

MASTERTHESIS
Dustin Spallek

Deep Learning basierte Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Dustin Spallek

Deep Learning basierte Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang *Master of Science Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Prof. Dr.-Ing. Andreas Meisel

Eingereicht am: 15.09.2020

Dustin Spallek

Thema der Arbeit

Deep Learning basierte Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten

Stichworte

maschinelles lernen, künstliche Intelligenz, Objekterkennung, Posenerkennung, synthetische Datenerzeugung, Augmented Reality, sechs Freiheitsgrade

Kurzzusammenfassung

Aufbauend auf der in [41] erläuterten Pipeline sowie der in [42] umgesetzten Erstellung eines synthetisch erzeugten Datensatzes zur Erkennung und 6-DOF (degrees of freedom, deutsch Freiheitsgrade) Lagebestimmung von Objekten. Erfolgt in dieser Arbeit eine Untersuchung der Deep Learning basierten Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten. Hierzu gehört eine Bewertung der Übertragbarkeit der trainierten Modelle beim Einsatz in realen Umgebungen und die Einschätzung von geeigneten Anwendungsmöglichkeiten.

Dustin Spallek

Title of Thesis

Deep learning based recognition of 3D object poses based on synthetically generated data

Keywords

machine learning, artificial intelligence, object recognition, pose estimation, synthetic data generation, augmented reality, six degrees of freedom

Abstract

Based on the pipeline explained in [41] and the creation of a synthetically generated data set for the recognition and 6-DOF (degrees of freedom) position determination of objects implemented in [42]. In this thesis, the deep learning based recognition of 3D

object poses on the basis of synthetically generated data is investigated. This includes an evaluation of the transferability of the trained models when used in real environments and the assessment of suitable application possibilities.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Im Rahmen der fortlaufenden Digitalisierung gewöhnt sich der Mensch immer mehr an die einfache Zugänglichkeit von Informationen. Gleichzeitig berherbergen diese Informationen unter anderem ein großes Potenzial, das Verstehen unserer Welt zu vereinfachen. So können diverse Medien genutzt werden, um sich beispielsweise mithilfe von Augmented Reality Anwendungen in der Welt zurechtzufinden. Eine der Herausforderungen in solchen Augmented Reality Anwendungen ist die Anreicherung von Informationen an Objekten, wenn diese Objekte beispielsweise über eine Kamera zu sehen sind. Dafür ist es notwendig einem Computer eine Art von Bewusstsein zu schaffen, damit dieser in der Lage ist die Position eines gewünschten Objektes zu bestimmen, um anschließend eine Annotation des Objektes im Kamerabild zu bewirken. Noch schwieriger wird es, wenn der Computer aus einem einzelnen Kamerabild die 3D Pose eines Objektes bestimmen soll, um beispielsweise einen Handwerker bei Montagearbeiten zu unterstützen. Diesbezüglich könnten Objekte innerhalb einer Brille mit durchsichtigen Displays optisch hervorgehoben werden. Für diese Art der Posenerkennung von Objekten gibt es bereits analytische Ansätze wie SIFT, SURF und ORB, welche jedoch anfällig gegen die Verdeckung der zu erkennenden Objekte sind. Der Ansatz des maschinellen Lernens hingegen bietet die Möglichkeit beim Training eines Modells eine Robustheit gegenüber der Verdeckung von Objekten zu erlernen. Dafür ist jedoch eine große Datenbasis mit Bildern notwendig, die korrekte Metadaten über die 3D-Pose der Objekte beinhalten. Beschriftete Daten für die 3D-Objekterkennung lassen sich manuell kaum erzeugen, da hierfür erhebliche Aufwände wie Messungen des Abstandes zur Kamera oder der Rotation der Objekte aus verschiedenen Kameraperspektiven nötig sind. Selbst halb automatische Beschriftungen mit Werkzeugen wie beispielsweise LabelFusion [27] sind arbeitsintensiv, wenn Trainingsdaten mit ausreichender Variation zu generieren sind. Gleichzeitig sind keine realen Trainingsdaten für Positionsschätzungen bekannt, die extreme Lichtverhältnisse oder die Posen von Objekten beinhalten. Um diese Einschränkungen der realen Daten zu überwinden und für die robuste Annotation der 3D-Pose aus einzelnen RGB-Bildern von Objekten mittels maschinellem Lernen, wird in dieser Arbeit auf synthetisch generierte

Daten zurückgegriffen. Dies ermöglicht letzten Endes eine Einschätzung der Übertragbarkeit der Modelle beim Einsatz in realen Umgebungen.

In dieser Arbeit wird sich auf starre, bekannte Objekte konzentriert, für die eine vorläufige Trainingszeit zum Erlernen des Aussehens und der Form der Objekte vorgesehen ist. Weiter sind die Objekte drei verschiedenen Objektklassen (6-DOF, 5-DOF und 3-DOF) zugeordnet, welche sich nach der Erkennbarkeit aus den Freiheitsgraden ergibt. Außerdem wird unter Verwendung synthetischer Daten, die mithilfe der Unreal Engine[1] und dem von Jonathan Tremblay et al. [44] entwickelten NDDS-Plugin erstellt wurden, in dieser Arbeit ein hochmodernes tiefes neuronales Netz verwendet [47], das in der Lage ist, Objektpositionen mit einem einzigen Blick zu ermitteln. Dieses Netzwerk lässt sich auch besser auf neuartige Umgebungen einschließlich extremer Lichtverhältnisse verallgemeinern, was die Experimente in dieser Arbeit zeigen. Mit diesem Netzwerk wird somit ein System zur Schätzung von Objektposen mit ausreichender Genauigkeit für das semantische Erfassen von bekannten Gegenständen in der realen Welt und der damit verbundenen Überbrückung der Realitätslücke [48] demonstriert. Gleichzeitig werden Experimente zur Überprüfung des Nutzens der Deep Object Pose Estimation (DOPE) im Zusammenhang mit der 6-DOF-, 5-DOF- und 3-DOF-Posenschätzung durchgeführt.

1.1 Motivation

Oftmals sind Trainingsdaten öffentlich nicht verfügbar oder sie existieren gar nicht erst. Eine Möglichkeit, mit dieser Problematik umzugehen, ist, die Trainingsmenge selbst zu erstellen. Dann stellt sich jedoch die Frage, wie möglichst effektive Trainingsdaten erzeugt werden können. Um diese Einschränkungen der realen Daten zu überwinden, kann wie von Jonathan Tremblay et al. [47] auf synthetisch generierte Daten zurückgegriffen werden. Wenn Netzwerke mit synthetischen Daten trainiert wurden, bestand bisher eine der größten Herausforderungen in der Schließung der sogenannten Realitätslücke, welche die korrekte Funktionsweise der Modelle beeinflusst, sobald sie mit Daten aus der realen Welt konfrontiert werden. Diesbezüglich ist an dem für diese Arbeit verwendeten synthetischen Datensatz besonders, dass bei diesen Daten eine Kombination aus nicht-fotorealistischen Umgebungsdaten (Domain Randomized) und fotorealistischen Daten zur Datenaugmentierung stattfindet, wodurch die Stärken von beiden Bereichen mit in die Traininsmenge

[1] https://www.unrealengine.com/en-US/, 15.09.2020

einfließen, was einer Überanpassung (englisch overfitting) und einer zu großen Realitätslücke entgegenwirkt. Weiter haben Modelle auf Basis synthetischer Daten den zusätzlichen Vorteil, dass sie robust gegenüber Lichtveränderungen, Kameravariationen und unterschiedlichen Hintergründen sind. In diesem Zusammenhang verspricht das Training tiefer neuronaler Netze für die 3D-Posenschätzung unter Verwendung synthetisch erzeugter Daten eine nahezu unbegrenzte Menge an vorbeschrifteten Trainingsdaten, die für unterschiedlichste Objekte sicher und unbedenklich generiert werden können.

In diesem Sinn ist der Weg für Anwendungen frei, um Objekte und deren Pose in der Realität zu erkennen. Wobei es nun gilt, deren Anwendbarkeit zu überprüfen, um geeignete Anwendungsfälle zu finden.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist die korrekte Annotation der 3D-Pose aus einzelnen RGB-Bildern von Objekten verschiedener Objektklassen (6-DOF, 5-DOF und 3-DOF) mittels maschinellem Lernen auf Basis synthetisch erzeugter Daten und die Einschätzung der Übertragbarkeit der Modelle beim Einsatz in realen Umgebungen.

1.3 Aufbau der Arbeit

Im Anschluss an diese Einleitung folgt das Kapitel der Analyse zur Eingliederung dieser Arbeit in ihr Themengebiet. Dazu gehört die Eingrenzung der Arbeit in ihre wichtigen Teilbereiche. Angefangen bei Augmented Reality zur Herausarbeitung der Notwendigkeit von Posenschätzungen. Weiter zu der Bildverarbeitung, welche die notwendige Basis des für diese Arbeit erstellten synthetischen Datensatzes mit RGB-Bildern, Tiefeninformationen und Segmentierungen ist. Über die Computergrafik als technische Lösung zur Erstellung von synthetischen Daten, bis hin zur Annotation von Objekten zur Ergründung des Mittels der Wahl zum Thema der Deep Learning basierten Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten. Weiter wird auf den Reality Gap (deutsch, Realitätslücke) hingewiesen, da dessen Beachtung essenziell bei der Erstellung synthetischer Daten ist. Anschließend erfolgt eine Analyse zu verwendeten Messmethoden zur Bestimmung der Qualität von Posenerkennungen, um die Wahl der für diese Arbeit

verwendete Messmethode zu begründen. Zuletzt werden Rückschlüsse auf den aktuellen Forschungsstand zur Eingliederung der Forschungsfrage für diese Arbeit geschlossen.

Nach der Analyse erfolgt das Kapitel der 'Methodik und Forschung', in welcher zunächst das ART-System als Benchmark, eine Eingliederung in Objektklassen und der allgemeine Aufbau der Experimente für diese Arbeit beschrieben wird. Daraufhin folgen die Experimente. Angefangen bei der Gegenüberstellung eines selbst erstellten Modells und einem Modell von NVIDIA-Research, um die Validität der weiteren Experimente zu belegen. Weiter über das Experiment, in dem domänenrandomisierte und fotorealistische Datensätze als Grundlage für die Modellerstellung gegenübergestellt werden, um die Notwendigkeit von fotorealistischen Daten zur Überbrückung des Reality Gaps nach Jonathan Tremblay et al. [47] zu überprüfen. Daraufhin folgen Experimente bezüglich der 5-DOF und 3-DOF-Objektklassen, um die Anwendbarkeit von Deep Learning basierte Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten einzugrenzen.

Angeknüpft an die 'Methodik und Forschung' folgt das Kapitel 'Ergebnisse und Diskussion', in dem diverse Fragestellungen debattiert werden. Zu den Fragestellungen gehört die allgemeine Anwendbarkeit, die Notwendigkeit des Fotorealismus beim Datensatz, Optimierungsmöglichkeiten sowie Alternativen.

Abschließend findet ein Fazit statt, in dem die wichtigsten Bestandteile dieser Arbeit zusammengefasst werden und ein Ausblick, der weitere Untersuchungen bezüglich des Themengebiets dieser Arbeit vorschlägt.

2 Analyse

Zum Verständnis dieser Arbeit erfolgt in diesem Kapitel eine analytische Auseinandersetzung mit den einzelnen notwendigen Teilbereichen zur Erkennung der 6-DoF-Pose von Objekten. Hierzu gehört eine Erläuterung des Einsatzes der Bildverarbeitung, die Möglichkeiten des Einsatzes von maschinellem Lernen sowie der Einsatz von Computergrafik für die Erstellung synthetischer Daten. Weiter werden vergleichbare Arbeiten vorgestellt, dich sich mit dem Thema der Posenschätzung von Objekten auf Bildern mittels Deep Learning beschäftigen, um Rückschlüsse auf den aktuellen Forschungsstand zu ziehen.

2.1 Augmented Reality

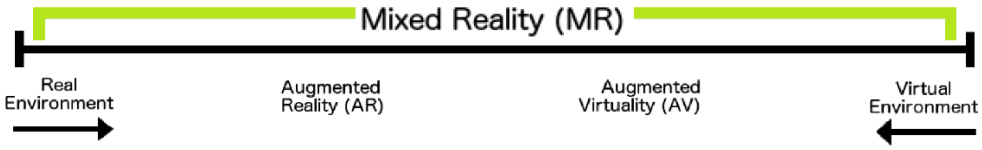

Abbildung 2.1: Kontinuum der Realität und Virtualität [30]

Im Laufe der Zeit entwickelten sich unterschiedliche Klassifizierungen bezüglich der Gratwanderung zwischen Realität und Virtualität, welche heutzutage in der Literatur häufig durch das Schaubild des 'Kontinuum der Realität und Virtualität' von Paul Milgram [30] erläutert wurden. Hier ordnet sich auch der Begriff der erweiterten Realität (englisch Augmented Reality, kurz AR) ein, welche die computerunterstützte Wahrnehmung bzw. Darstellung virtueller Objekte in der realen Welt in Echtzeit beschreibt. Azuma [3] erklärt Augmented Reality diesbezüglich als immersive Erfahrung, die virtuelle dreidimensionale Objekte mit der realen Umgebung überlagert, sodass die Illusion entsteht, dass die virtuellen und realen Objekte im selben Raum existieren.

In diesem Sinn kann durch die Verwendung von Kameras, die Realität erfasst und durch zusätzliche Informationen direkt in das aktuelle Abbild der realen Welt eingearbeitet werden. So entstehen diverse Möglichkeiten, wie die in der Einleitung erwähnte Unterstützung von Handwerkern bei Montagearbeiten. Für diesen Anwendungsfall ist in Kombination mit Augmented Reality jedoch die Erkennung von Objekten sowie deren Pose notwendig, an welche im weiteren Verlauf dieser Arbeit eingegangen wird.

2.2 Bildverarbeitung

Digitale Bildverarbeitung wird heutzutage im breiten Feld der Computer Vision eingesetzt. Bezüglich der Bildverbesserung beispielsweise gab es bereits 1984, wie von Edward H Adelson et. al [1] beschrieben, eine Vielzahl von Methoden, um Bildverschlechterungen zu entfernen und wichtige Bildinformationen hervorzuheben. Weiter können in der Computergrafik digitale Bilder für eine Vielzahl von visuellen Effekten erzeugt, modifiziert und kombiniert werden. Bei der Datenkomprimierung können Bilder effizient gespeichert und übertragen werden, wenn sie in einen kompakten digitalen Code umgewandelt werden. Bei der maschinellen Bildverarbeitung können automatische Prüfsysteme und Roboter einfache Entscheidungen auf der Grundlage des digitalisierten Eingangs von einer Kamera treffen. So ist es nicht verwunderlich, dass die Verwendung von Bildverarbeitung heutzutage ihren Nutzen im Feld des maschinellem Lernens gefunden hat, um künstlichen Intelligenzen ein gewünschtes Verhalten anzutrainieren.

Diese Sektion beschäftigt sich mit den Techniken der Bildverarbeitung. Konkret wird zwischen RGB-Bildern, Tiefeninformation und Segmentation zum Zwecke der Lokalisierung und Isolierung von Merkmalen für das Training des neuronalen Netzes zu dieser Arbeit unterschieden.

2.2.1 RGB-Bilder

Bei der Betrachtung eines RGB-Bildes gelingt es uns Menschen, eine enorme Menge an Informationen aus diesem Bild zu entnehmen. So brauchen wir beispielsweise nur ein Bild betrachten, um sofort eine ungefähre Vorstellung über die auf dem RGB-Bild abgebildete Szene zu haben. In diesem Zusammenhang können wir bereits Objekte erkennen und auch abschätzen, wie weit die Objekte ungefähr voneinander entfernt sind. Aus diesem Grund

ist das RGB-Bild auch eine wichtige Datenquelle für die Interpretation von Szenen für künstliche Intelligenzen.

Für ein erfolgreiches Training einer künstlichen Intelligenz im Rahmen dieser Arbeit sind Bilder notwendig, die möglichst genau der Realität [46, 24] entsprechen, um der künstlichen Intelligenz eine Transferleistung zwischen Realität und Virtualität zu ermöglichen. In diesem Zusammenhang sollten die Bilder ein natürliches Verhalten von Umgebungen berücksichtigen. Gleichzeitig müssen die Bilder Informationen beinhalten, die für das gewünschte Lernziel notwendig sind.

Abbildung 2.2: Beispiel eines RGB-Bildes aus dem eigens erstellten Datensatz

Abbildung 2.2 zeigt ein übliches RGB-Bild, welches im Kontext dieser Arbeit verwendet wurde. Auf dem Bild ist ein Wohnzimmer zu sehen, in dem für ein Wohnzimmer typische Elemente, wie Teppiche, Deko, Regale, usw. zu erkennen sind. Die Beleuchtung des Raumes ist dynamisch, sodass das Licht, das aus einem Fenster in den Raum hell hinein scheint realistisch wirkende Schatten produziert. Gleichzeitig gibt es Bereiche, die weniger gut ausgeleuchtet sind. Somit übermittelt das Bild beim Betrachten insgesamt eine für ein Wohnzimmer typische Atmosphäre und wirkt realistisch, obwohl es sich hierbei um ein synthetisch erzeugtes RGB-Bild handelt. Außerdem sind hierbei Objekte zu erkennen, die für das spätere Training des neuronalen Netzes berücksichtigt werden sollen (Banane, Cracker-Box, Star Wars - TIE Fighter, ...).

2.2.2 Tiefeninformation

Die Schätzung der Tiefeninformation ist eine wichtige Komponente für das Verständnis geometrischer Beziehungen innerhalb einer Szene. Die Einbeziehung der Tiefeninformation hilft in diesem Zusammenhang zu einer reichhaltigeren Darstellung von Objekten und deren Umgebung, was beispielsweise laut Nathan Silberman et. al [40] bei der Objekterkennung häufig zu Verbesserungen führt. Darüber hinaus entstehen durch die Tiefeninformation weitere Anwendungsmöglichkeiten, beispielsweise in der 3D-Modellierung [37, 17], bei der Erstellung physikalischer Modelle [40], in der Robotik [13, 29] oder auch in der Erkennung der Verdeckung von Objekten. Für diese Arbeit dienen die Tiefeninformationen als Augmentation des Datensatzes zum Training des neuronalen Netzes zur Schätzung der Pose von Objekten und tragen zur Erkennung der Entfernung von Objekten zur Kamera bei.

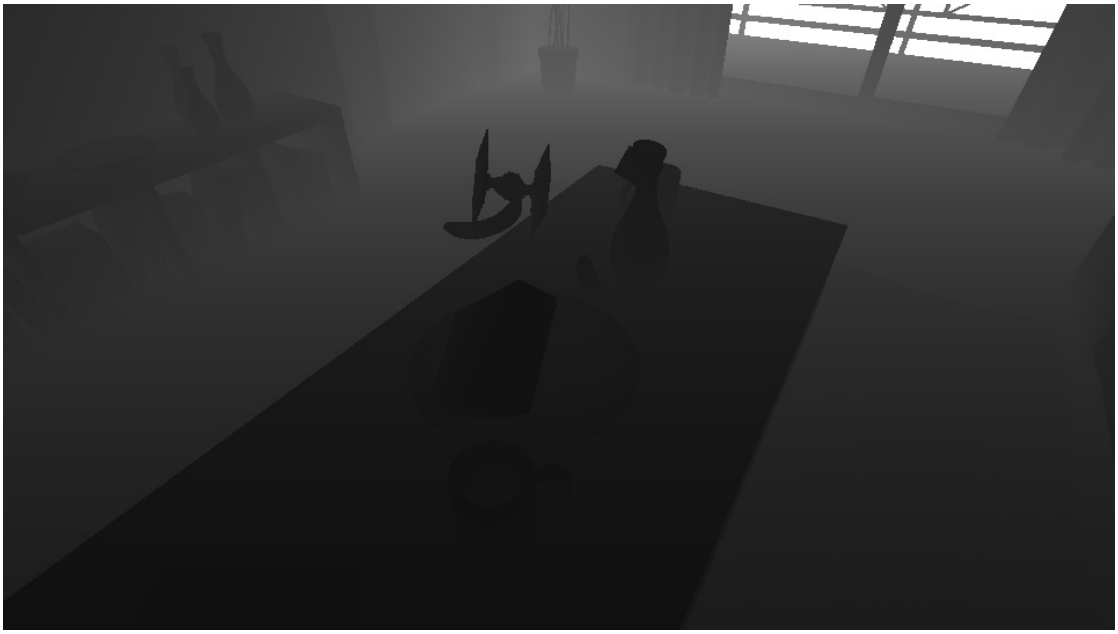

Abbildung 2.3: Beispiel eines Tiefenbildes aus dem eigens erstellten Datensatz

Abbildung 2.3 zeigt ein Bild, welches ausschließlich die Tiefeninformationen darstellt. Je dunkler die Bereiche innerhalb eines Bildes sind, desto geringer sind die Objekte von der Kamera entfernt. Auch wenn ausschließlich die Tiefeninformationen dargestellt werden, lässt sich auf den Inhalt innerhalb des Bildes schließen. So ist der gleiche Raum zu erkennen, wie bereits als RGB-Bild in Abbildung 2.2 dargestellt.

Das Auffinden von Tiefenbeziehungen aus Einzelbildern ist weniger einfach und erfordert die Integration sowohl globaler als auch lokaler Informationen über die aufgenommene Szene. Diesbezüglich können beispielsweise Stereobilder verwenden werden, bei denen die lokalen Abstände und Einstellungen der Kameras genügend Information für eine grobe Schätzung der Tiefeninformationen beinhalten. Im Hinblick auf die Erstellung eines Datensatzes zur Schätzung der Pose von Objekten kommt erschwerend hinzu, dass eine große Menge von Daten mit Tiefeninformation benötigt wird. Aus diesem Grund geschieht die in dieser Masterarbeit verwendete Methode zur Extraktion der Tiefeninformation mit der Hilfe einer Game-Engine, welche im Abschnitt 2.3 "Computergrafik" beschrieben wird.

2.2.3 Segmentierung

Der letzte Punkt im Bezug zur Bildverarbeitung ist die Segmentierung. Auch bei der Segmentierung geht es darum, zu einem gewissen Grad den menschlichen Erkennungsprozess nachzuahmen, indem Merkmale aus einem Bild extrahiert werden. Wenn wir als Menschen Objekte betrachten, können wir in der Regel mit Leichtigkeit sagen, an welchen Stellen sich auf einem Bild Objekte voneinander abgrenzen. In diesem Zusammenhang ist die Bildsegmentierung die Aufteilung eines Bildes in verschiedene Regionen, die jeweils bestimmte Eigenschaften haben.

Oft ist laut King-Sun Fu et. al [11] die Bildsegmentierung der erste Schritt der Bildanalyse, der entweder auf eine Beschreibung eines Bildes oder auf eine Klassifizierung des Bildes abzielt, wenn eine Klassenbezeichnung sinnvoll ist. Die Segmentierung innerhalb eines Bildes ist eine kritische Komponente bei der Objekterkennung, da sich Fehler in der Segmentierung auf die Merkmalsextraktion und -klassifizierung ausbreiten können. Insgesamt wurden im Laufe Forschung im Feld der Bildverarbeitung zahlreiche Bildsegmentierungstechniken [11, 53, 22] vorgeschlagen. Im Rahmen dieser Arbeit wird eine Segmentierung entlang der Silhouette eines Objektes vorgenommen, welche die Erkennung der 2D-Position von Objekten auf einem Bild unterstützt. Diese Silhouette werden weiter den jeweiligen Objektklassen zugeteilt, die beim späteren Training des neuronalen Netzes berücksichtigt werden sollen.

Abbildung 2.4 zeigt ein Bild, welches ausschließlich die Segmentierung von Objekten in Graustufen darstellt. Gut zu erkennen sind unter anderem die Umrisse einer Kaffeetasse, die des Star Wars - TIE Fighters oder auch der Banane. Bis auf die Umrisse sind keine weiteren Details auf den Objekten zu erkennen.

Abbildung 2.4: Beispiel eines Segmentierung-Bildes aus dem eigens erstellten Datensatz

Wie bereits im Abschnitt zur Tiefeninformation erwähnt, kommt auch bei der Segmentierung erschwerend hinzu, dass eine große Menge von Segmentierungsdaten benötigt wird. Auch hierbei bieten Game-Eniges eine gute Möglichkeit die Segmentierung von Objekten auf Bildern hervorzunehmen, was in der nun folgenden Sektion beschrieben wird.

2.3 Computergrafik

Eines der schwierigsten Probleme der Computergrafik [32] ist es laut John Amanatides [2], Bilder zu erzeugen, die realistisch erscheinen, d. h. Bilder, die einen menschlichen Betrachter bei der Darstellung auf einem Bildschirm täuschen können. Die in [2] von John Amanatides dargestellten Objekte waren damals offensichtlich nicht realistisch, enthielten aber genügend Informationen für Aufgaben wie zum Beispiel computergestützte Designs. Diese Technologie, die in der Lage war Bilder darzustellen, eröffnete die Forschung in diese Richtung und entwickelte sich im Laufe der Zeit zu einem weiteren Teilbereich für Erkennung der 6-DoF-Pose von Objekten - so auch für diese Masterarbeit. Heute lassen sich komplexe Formen geometrisch modellieren, um realistisch wirkende virtuelle Umgebungen zu erstellen. Innerhalb dieser virtuellen Umgebungen besteht weiter die Möglichkeit, die in der vorherigen Sektion behandelten RGB-Bilder, Tiefeninformationen oder Segmentierungen zu berechnen.

2.3.1 Erstellung realistischer Bilder

Ein Erstreben innerhalb der Computergrafik ist es, die virtuelle Umgebung möglichst realistisch darzustellen. Hierbei kann, wie von Margaret A. Hagen [14] beschrieben, zwischen drei verschiedenen Varianten unterschieden werden:

- Physikalischer Realismus: gleiche visuelle Stimulation wie in der Realität
- Fotorealismus: gleiche visuelle Reaktion wie in der Realität
- Funktionaler Realismus: gleiche visuelle Information wie in der Realität

Das Kriterium für den physikalischen Realismus ist hier, dass das Bild die gleiche visuelle Stimulation wie die Realität bieten muss. Wenn wir die optische Filterung und Streuung im Auge vernachlässigen, bedeutet dies, dass das Bild eine genaue Punkt-für-Punkt-Darstellung der spektralen Bestrahlungsstärkewerte an einem bestimmten Punkt in der Szene sein muss. Dies stellt laut James A. Ferwerda [10] hohe Anforderungen an den Bilderzeugungsprozess. Zunächst muss das Modell genaue Beschreibungen der Formen, Materialien und Beleuchtungseigenschaften wie in der Realität enthalten. Als Nächstes muss der Renderer in der Lage sein, die spektralen und intensiven Eigenschaften der Lichtenergie, die am Standpunkt des Beobachters ankommt, genau zu simulieren.

Schließlich muss das Anzeigegerät in der Lage sein, diese Energien genau zu reproduzieren. Obwohl die ersten beiden Ziele mit physikalisch basierten Bildsynthesemethoden erreicht werden können, können herkömmliche Displays die gerenderten Lichtenergien im Allgemeinen nicht reproduzieren. So ist laut James A. Ferwerda [10] die Erzeugung physikalisch realistischer Bilder derzeit nur unter eingeschränkten Bedingungen möglich ist.

Wenn wir in der Computergrafik von Fotorealismus sprechen, meinen wir in der Regel, dass wir ein Bild schaffen wollen, das von einer Fotografie einer Szene nicht zu unterscheiden ist. Das Ziel wirft aber die Frage nach dem Realismus auf, da es nicht erklärt, warum ein Foto realistisch ist. Die für diese Arbeit verwendete Definition von Fotorealismus besteht darin, zu sagen, dass das Bild fotometrisch realistisch sein muss. Die Photometrie ist das Maß für die Reaktion des Auges auf Lichtenergie, daher erfordert diese Definition, dass das Bild die gleiche visuelle Reaktion wie die Szene erzeugen muss, auch wenn die physikalische Energie, die vom Bild ausgeht, anders sein kann als die Szene.

In der Computergrafik ist ein hohes Maß an Realität jedoch nicht notwendig. Diese Überlegungen legen einen dritten Standard für Realismus in der Computergrafik nahe, und das ist der funktionale Realismus. Hier ist das Kriterium für den Realismus, dass das Bild die gleichen visuellen Informationen wie die Szene liefern muss. Information bedeutet hier Wissen über die sinnvollen Eigenschaften von Objekten in einer Szene, wie z. B. ihre Formen, Größen, Positionen, Bewegungen und Materialien, das einem Beobachter erlaubt, zuverlässige visuelle Beurteilungen zu treffen und nützliche visuelle Aufgaben zu erfüllen.

Im Rahmen dieser Arbeit werden Bilder entsprechend des Fotorealismus (fotorealistische Szenen), als auch des funktionalen Realismus erstellt. Bezüglich der funktional realistischen Bilder wird im Kontext dieser Arbeit weiter von dem von Tobin et al. [45] eingeführten Begriff der Domänen Randomisierung gesprochen. Diese Komposition von realistischen synthetischen und Domänen randomisierten Bildern ist nach Tobin et al. [45] und Jonathan Tremblay et al. [46] wichtig, um die sogenannte Realitätslücke zu schließen, damit neuronale Netzwerke, die auf synthetischen Daten trainiert wurden, korrekt funktionieren, wenn sie mit Daten aus der realen Welt konfrontiert werden.

Für die Erstellung von realistischen Bildern wird heutzutage unterschiedliche 3D Grafiksoftware eingesetzt. Welche Unterschiede zwischen verschiedenen 3D Grafiksoftware Programmen bestehen, wird im weiteren Verlauf geklärt.

2.3.2 3D Grafiksoftware

Computerprogramme, mit denen sich dreidimensionale Szenen erstellen und / oder rendern (aus ihnen Bilder oder Computeranimationen errechnen) lassen, werden als 3D Grafiksoftware bezeichnet. Beispiele hierfür sind Blender[1], Unreal Engine[2] oder Unity 3D[3], wobei die Unreal Engine als auch Unity 3D allgemein auch als Spiele Engines bezeichnet werden.

Jede 3D Grafiksoftware beinhaltet eine Grafik Engine, welche für die grafische Darstellung von digitalen Objekten auf Bildschirmen zuständig ist. Unterschiedliche 3D Grafiksoftware grenzt sich häufig in der intern verwendeten Render Engine ab. So beinhalten beispielsweise Spiele Render Engines, die darauf ausgelegt sind realistische Bilder effizient in möglichst kurzer Zeit zu produzieren. Gleichzeitig ist das Ziel von einer Grafiksoftware wie beispielsweise Blender, Bilder mit einem hohen Anspruch an die Qualität, jedoch auf Kosten der Berechnungszeit zu erzeugen.

Abbildung 2.5: Gegenüberstellung der Bildqualität in Render Engines

Abbildung 2.5 zeigt die Unterschiede zwischen einem mit hohen Anspruch an Qualität in Blender erstellten Bild (links) und einem Screenshot (rechts), welches in Echtzeit mithilfe der Unreal Engine erzeugt wurde. Hierbei ist gut zu erkennen, dass das linke Bild aufgrund höherer Schatten- sowie Reflexionsdetails wesentlich realistischer als das rechte Bild aussieht. Gleichzeitig nehmen wir als Menschen beide Objekte eindeutig als Donut wahr.

[1] https://www.blender.org/, 15.09.2020
[2] www.unrealengine.com, 15.09.2020
[3] https://unity.com/de, 15.09.2020

Im Rahmen dieser Arbeit wurde die Unreal Engine unter Verwendung des von Thang To et al. [44] im Auftrag von Nvidia[4] erstellten Plugins genutzt, da damit in akzeptabler Zeit eine große Menge von ausreichend realistisch wirkenden synthetischen Daten erzeugt werden kann. Zudem sei erwähnt, dass in vergleichbaren Arbeiten, wie beispielsweise von Yu Xiang et al. [52] andere 3D-Grafiksoftware genutzt wurde, die alle notwendigen Werkzeuge zur schnellen Erzeugung qualitativ hochwertiger Bilder bereitstellt.

2.4 Annotation von Objekten

Um Objekte der Welt durch Informationen beispielsweise in einem Kamerabild annotieren zu können, ist der erste notwendige Schritt die Erkennung der Objekte und weiter die Bestimmung der Pose. Allgemein wird das Verfahren zur Identifizierung von bekannten Objekten mittels optischer, akustischer oder anderer physikalischer Methoden [5] als Objekterkennung (engl., Objectrecognition) beschrieben. Zur Bestimmung der Präsenz eines Objektes in einem Bild oder dessen Position und Lage existieren in der Computer Vision unterschiedliche Herangehensweisen. In diesem Zusammenhang kann grob zwischen analytischen Ansätzen aus der Bildverarbeitung und Deep Learning Verfahren aus dem Bereich der künstlichen Intelligenz unterschieden werden. In dieser Sektion folgt eine grobe Einleitung der analytischen Ansätze sowie der Übergang zu der für diese Arbeit verwendeten Methodik des maschinellen Lernens.

2.4.1 Analytische Ansätze

Zur Erkennung von Objekten dienen in der Computer Vision häufig sogenannte Merkmale. Ein Merkmal beschreibt in diesem Zusammenhang beispielsweise einen für ein Objekt markanten optischen Bestandteil, der ausschlaggebend für die Wiedererkennung des Objektes ist. Diese Merkmale machen sich die zu den analytischen Ansätze zählenden Deskriptoren wie SIFT, SURF und ORB zunutze. Dabei ist ein Deskriptor ein Vektor, dessen Aufgabe es ist, die Umgebung eines gefunden Merkmals zu beschreiben, sodass es möglich ist, dieses Merkmal in anderen Bildern wiederzufinden. Ein guter Deskriptor ist laut Shaharyar Ahmed Khan Tareen et al. [43] gegen Rotation und unterschiedliche Beleuchtung invariant und beinhaltet alle nützlichen Informationen, um wichtige Entscheidungen bei visuell basierten Anwendungen zu treffen.

[4] https://www.nvidia.com, 15.09.2020

SIFT

SIFT ist eine von David G. Lowe [26] vorgestellte Methode zur Extraktion unveränderlicher Unterscheidungsmerkmale, mit deren Hilfe ein zuverlässiger Abgleich zwischen verschiedenen Ansichten eines Objektes durchgeführt werden kann. Die Merkmale sind invariant gegenüber dem Abbildungsmaßstab und der Bilddrehung und zeigen einen robusten Abgleich über einen erheblichen Bereich von leichter Verzerrung, Änderung des 3D-Blickwinkels, Bildrauschen und Änderung der Beleuchtung. In diesem Zusammenhang eignet sich die Extraktion von Merkmalen mittels SIFT zur Verwendung für die Objekterkennung. Die Erkennung erfolgt durch Zuordnung einzelner Merkmale zu einer Datenbank mit Merkmalen bekannter Objekte unter Verwendung eines schnellen Algorithmus für den nächsten Nachbarn [21]. Anschließend folgt eine Hough-Transformation [18] zur Identifizierung von Clustern sowie eine Verifizierung durch die Lösung der kleinsten Quadrate zur Ermittlung konsistenter Posenparameter. Mit diesem Erkennungsansatz können Objekte unter Clutter und Okklusion robust identifiziert werden.

SURF

SURF (Speeded Up Robust Features) ist ein auf SIFT aufbauender Algorithmus zur schnellen und robusten Erkennung von Bildmerkmalen für maschinelles Sehen und wurde erstmalig von Herbert Bay et al. [4] vorgestellt. Indem sich SURF auf integrale Bilder für Bildfaltungen stützt und auf die Stärken der führenden existierenden Detektoren und Deskriptoren aufbaut, hat SURF sehr schnell bei der Erkennung von Merkmalen mit einer hohen Robustheit und Unterscheidbarkeit an Bedeutung gewonnen.

ORB

SIFT und SURF sind zwei der derzeitigen Methoden, welche auf kostspieligen Deskriptoren für eine hoch präzise Erkennung und Abgleichung von Merkmalen beruhen. ORB ist eine von Ethan Rublee et al. [36] vorgestellte alternative zu SIFT und SURF und als sehr schneller binärer Deskriptor bekannt, der rotationsinvariant und rauschresistent ist.

2.4.2 Maschinelles Lernen

Welche Komponenten sind für den Erfolg oder Misserfolg eines lernenden Systems verantwortlich? Welche Änderungen an ihnen bewirken eine Unterstützung des Lernerfolges von intelligenten Systemen? Marvin Minsky [31] stellte sich 1963 diese Fragen, um eine bessere Auswahl bei der Vergabe von Krediten zu bewirken. Seither wird das maschinelle Lernen als möglicher universeller Problemlöser gesehen. So hat es auch das inzwischen kommerziell wichtige Teilgebiet des Tiefen Lernens (engl. Deep Learning) von künstlichen neuronalen Netzen in diese Arbeit geschafft. Ein neuronales Netz besteht standardmäßig aus vielen einfachen, miteinander verbundenen Knoten, die Neuronen genannt werden und jeweils eine Folge von Aktivierungen erzeugen. An dieser Stelle erfolgen jedoch keine weiteren tiefergehenden Erklärungen zu neuronalen Netzen, da bereits genügend andere Arbeiten [31, 38, 41] dieses Thema behandelt haben. Während analytische Ansätze anfällig gegenüber Verdeckungen und der optischen Veränderungen von Objekten sind [43, 7, 23], bieten Verfahren des maschinellen Lernens durch ihre Fähigkeit der Generalisierung hierbei eine höhere Resistenz. In dieser Arbeit wird der auf Klassifikation beruhende Ansatz des maschinellen Lernens im Gegensatz zum Ansatz des bestärkenden Lernens [34] bei der Posenbestimmung verwendet. Zum Verständnis folgen nun ein paar Worte zu Convolutional Neural Networks, Deep Learning sowie Data Augmentation.

Convolutional Neural Network

Die Grundbausteine für moderne Posenschätzung von Objekten sind laut Jürgen Schmidhuber [38] neuronale Netze. Ihre Stärke liegt darin, viele Feinheiten zu erfassen, die klassische Ansätze nicht erfassen können. Die meisten Netze zur Posenerkennung verwenden Convolutional Neural Networks (deutsch, Faltungsnetzwerke) zur Klassifizierung von Objekten in Bildern. Fortgeschrittenere Architekturen verwenden Faster R-CNN zur regionalen Positionsbestimmung bzw. MASK R-CNN zur Segmentierung von Objekten in Bildern. Eine detailliertere Beschreibung hierzu wurde von mir in [41] vorgenommen.

Deep Learning

Eines der Probleme von tiefen neuronalen Netzen waren lange Zeit die verschwindenden Gradienten. In diesem Zusammenhang nahm die Lernfähigkeit von neuronalen Netzen ab, wenn diese zu tief wurden. Beim Deep Learning geht es jedoch genau um das

Training von künstlichen Intelligenzen zum Erlernen von Fähigkeiten über viele Stufen hinweg. In diesem Zusammenhang ist der für diese Arbeit verwendete Ansatz der Posenerkennung von Jonathan Tremblay et al. [48] unter Verwendung von Convolutional Pose Machines (CPM) [49] besonders. Dieser Ansatz verwendet sogenannte "intermediate supervision" (Zwischenbegutachtungen), um das Problem der verschwindenden Gradienten für Faltungsnetze erfolgreich zu vermindert. Für eine detailliertere Erklärung zu CPMs möchte ich auch an dieser Stelle auf [41] verweisen.

2.5 Reality Gap

Die Übertragbarkeit eines Verhaltens aus einer simulierten Umgebung auf die Realität ist laut Inman Harvey [15] eine der größten Schwierigkeiten in der Robotik. Bei der Deep Learning basierten Erkennung von 3D-Objektposen auf Basis synthetischer Daten tritt dieses Problem insofern wieder auf, dass laut Jonathan Tremblay et al. [46] Faktoren wie der Verdeckungsgrad, die Texturierung und die Pose von Objekten in Kombination mit verschiedenen Licht- und Schattenverhältnissen zu berücksichtigen sind. Dies wirkt sich auf ein mittels Deep Learning trainiertes Modell insofern negativ aus, als dass das Modell weniger Robust gegenüber diesen Faktoren ist. In diesem Zusammenhang gilt es den Reality Gap zwischen synthetisch erzeugten Daten und Daten aus der Realität möglichst zu verringern.

2.5.1 synthetische Data Augmentation

Der wichtigste Faktor für ein erfolgreiches Training von neuronalen Netzen sind Daten. Häufig sind Daten jedoch gar nicht oder nur in einer begrenzten Form verfügbar. Hinzu kommt, dass die Anzahl und Qualität von Daten je nach Anwendungsfall hoch sein muss [39]. Die Data Augmentation (deutsch, Erweiterung von Daten) bietet hierfür einen alternativen Lösungsweg. In diesem Zusammenhang bewiesen Luis Perez und Jason Wang [35] die Wirksamkeit der Erweiterung von Daten durch einfache Techniken wie Beschneiden, Drehen und Spiegeln von Eingabebildern.

Synthetische Bild-Datensätze

Um in dieser Arbeit mit der Variabilität von Daten aus der realen Welt umzugehen, stützt sich das Netz zur Posenerkennung auf die synthetische "Data Augmentation" - Technik der Domänen-Randomisierung, bei der die Parameter des Simulators wie Beleuchtung, Pose, Objekttexturen usw. auf nicht-realistische Weise randomisiert werden, um das neuronale Netz zu zwingen, die wesentlichen Merkmale des zu erkennenden Objekts zu lernen. Für die Erstellung der synthetischen Daten wurde der NVIDIA Deep learning Dataset Synthesizer [44], wie auch von Jonathan Tremblay et al. [46] verwendet. Wie genau die Erzeugung der synthetischen Daten für diese Arbeit erfolgte, kann in [42] nachgelesen werden.

2.5.2 fotorealistische vs. randomisierte Daten

Inwieweit der Einfluss von randomisierten Daten sich auf das Training eines tiefen neuronalen Netzes auswirkt wurde erstmalig von Jonathan Tremblay et al. [46] untersucht. Dabei zeigten sie, dass die Domänenrandomisierung (DR) eine effektive Technik zur Überbrückung der Realitätslücke ist. Allein mit synthetischen DR-Daten haben sie ein neuronales Netzwerk so trainiert, dass es komplexe Aufgaben wie die Objekterkennung mit einer Leistung bewältigen kann, die mit arbeitsintensiveren (und damit teureren) Datensätzen vergleichbar ist. Durch die zufällige Störung dieser synthetischen Bilder während des Trainings verzichtet DR absichtlich auf Fotorealismus, um das Netzwerk zu zwingen, zu lernen, sich auf die relevanten Merkmale zu konzentrieren. Mit der Feinabstimmung an realen Bildern haben sie gezeigt, dass die DR sowohl photorealistischere Datensätze übertrifft als auch die Ergebnisse verbessert, die allein mit realen Daten erzielt werden. Daher ist die Verwendung von DR für das Training tiefer neuronaler Netze ein vielversprechender Ansatz, um die Leistungsfähigkeit der synthetischen Daten zu nutzen.

Im Zusammenhang mit dieser Arbeit sind die erstmaligen Untersuchungen von Jonathan Tremblay et al. Motivation für eigene Experimente zur Verifikation ihrer Aussagen mit der genaueren Untersuchung bezüglich der Vorteile durch fotorealistische Daten.

2.6 Messungen zur Qualität von Posenerkennungen

Die Messung der Qualität der trainierten Modelle gestaltet sich bei der 3D-Posenbestimmung auf Basis synthetischer Daten schwieriger als bei der herkömmlichen Untersuchung von trainierten Modellen. Während bei der üblichen Herangehensweise zur Qualitätsmessung von neuronalen Netzen laut Junhua Ding et al. [9] neben dem Trainingsdatensatz auch ein Testdatensatz zur Validation verwendet wird, reicht dies bei der 3D-Posenbestimmung nicht aus. Der Grund hierfür ist, dass sich die Qualität bei der 3D-Posenbestimmung nach dem korrekten Verhalten in der Realität richtet. Existieren jedoch nur synthetisch erzeugte Daten, kann nicht auf die Realität überprüft werden, was den Einsatz anderer Messansätze zur Bestimmung der Qualität fordert.

Bei der Recherche nach einer Methode zur Messung der Qualität einer 3D-Posenerkennung von Objekten wurden diverse Arbeiten [6, 33, 52, 8] herangezogen. Auffällig in all diesen Arbeiten war die Verwendung von Marker-Tracking [20] als Grundwahrheit. Somit gilt in der Literatur die Genauigkeit vom Marker-Tracking als ausreichend für Messungen von Objektposen.

Weiter wurde Jonathan Tremblay angeschrieben, um eine Vergleichbarkeit zu seiner Arbeit [47] zu gewährleisten. Doch auch er sagte:"We have other scripts to test the neural network that we did not share publicly. Since the network is meant to work on wild poses and objects. It was mainly tested on our robotics system. Sorry my help is limited here."[5].

Die Analyse der Möglichkeiten zur Messung führte dazu, dass in dieser Arbeit ein ähnlicher Ansatz wie bei der Messung mit den Marker-Trackern verwendet wird. Jedoch wird anstatt der Marker ein hoch genaues 'Advanced Realtime Tracking'-System[6] verwendet, welches im Kapitel "Methodik und Forschung" genauer beschrieben wird.

2.7 Vergleichbare Arbeiten

Im Verlauf dieser Arbeit wurden die Arbeiten von Jonathan Tremblay et al. [48, 46] häufig zitiert, weil diese als Grundlage zum Abgleich unter der Verwendung der gleichen Technologie (des NDDS [44] und des Deep Object Pose Estimation Netzes [47]) genutzt

[5] https://github.com/NVlabs/Deep_Object_Pose/issues/100, 15.09.2020
[6] https://ar-tracking.com/, 15.09.2020

werden. In diesem Zusammenhang untersuchten Jonathan Tremblay et al. die Realitätslücke im Zusammenhang mit der 6-DoF-Posenschätzung bekannter Objekte aus einem einzigen RGB-Bild. Sie zeigten, dass für dieses Problem die Realitätslücke durch eine einfache Kombination von domänenrandomisierten und fotorealistischen Daten erfolgreich überbrückt werden kann. Unter Verwendung synthetischer Daten, die auf diese Weise erzeugt wurden, stellen sie ein tiefes neuronales Netz vor, das in der Lage ist, konkurrierend gegen ein hochmodernes Netz zu arbeiten, das auf einer Kombination aus realen und synthetischen Daten trainiert wurde. Jonathan Tremblay et al. Wissens nach ist deren Netzwerk das erste tiefe Netzwerk, das nur auf synthetischen Daten trainiert wurde und das in der Lage ist, eine dem Stand der Technik entsprechende Leistung bei der 6-DoF-Objekt-Posenschätzung zu erzielen. Deren Netzwerk verallgemeinert besser auf neuartige Umgebungen einschließlich extremer Lichtverhältnisse, für die sie qualitative Ergebnisse zeigen. Insgesamt demonstrieren sie ein Echtzeitsystem zur Schätzung von Objektposen mit ausreichender Genauigkeit für das semantische Erfassen von bekannten Haushaltsgegenständen, damit reale Roboter diese Objekte greifen können.

Weitere Arbeiten, die sich mit dem Thema der Schätzung der Pose von Objekten auf zweidimensionalen Bildern beschäftigten wurden von Yu Xiang et al. [52], Adrien Gaidon et al. [12] und Keunhong Park et al. [34] unternommen.

So arbeiteten Yu Xiang et al. [52] an der Schätzung der 6D-Pose bekannter Objekte, um die Interaktion von Robotern mit der realen Welt zu ermöglichen. Auch sie gehen speziell auf die Herausforderung der Vielfalt von Objekten sowie der Komplexität innerhalb einer Szene aufgrund von Unordnung und Verdeckungen zwischen den Objekten ein. In ihrer Arbeit stellen sie PoseCNN vor, ein Convolutional Neural Network zur Schätzung der Pose von 6D-Objekten. PoseCNN schätzt die 3D-Translation eines Objekts, indem es sein Zentrum im Bild lokalisiert und seine Entfernung von der Kamera vorhersagt. Die 3D-Rotation des Objekts wird durch Regression auf eine Quaternionenrepräsentation geschätzt.

Adrien Gaidon et al. [12] nutzen in ihrer Arbeit die Computergrafik, um vollständig beschriftete, dynamische und fotorealistische virtuelle Welten zu generieren. Sie schlagen eine effiziente Methode zum Klonen von reale in virtuelle Welten vor und validieren deren Ansatz, indem sie einen Videodatensatz mit dem Namen "Virtual KITTI" erstellen und öffentlich zugänglich machen. Außerdem liefern sie quantitative experimentelle Beweise, die darauf hindeuten, dass moderne Deep Learning Algorithmen, die auf realen Daten trainiert wurden, sich in realen und virtuellen Welten ähnlich verhalten und dass das

vorherige Training auf virtuellen Daten die Leistung verbessert. Somit verkleinern sie die Lücke zwischen realen und virtuellen Welten und ermöglichen die Messung des Einflusses verschiedener Wetter- und Abbildungsbedingungen auf die Erkennungsleistung. Zusätzlich sei erwähnt, dass die Arbeit von Adrien Gaidon et al. als Benchmark von Jonathan Tremblay et al. [48] herangezogen wurde.

Während der Ansatz in dieser Arbeit auf das Training eines Convolutional Neural Netzworks mittels synthetischer Daten basiert, gehen Keunhong Park et al. [34] einen anderen Weg und sind deshalb nur teilweise vergleichbar. Ihr Ansatz erfordert keine 3D-Modelle für jedes Objekt und erfordert kein zusätzliches Training, um neue Objekte einzubeziehen. In ihrer Arbeit schlagen sie einen neuartigen Rahmen für die 6D-Posenschätzung von ungesehenen Objekten vor. Sie entwerfen ein durchgehendes neuronales Netzwerk, das eine latente 3D-Darstellung eines Objekts unter Verwendung einer kleinen Anzahl von Referenzansichten des Objekts rekonstruiert und mit Hilfe der gelernten 3D-Darstellung in der Lage ist, das Objekt aus beliebigen Ansichten zu rendern. Indem sie ihr Netzwerk mit einer großen Anzahl von 3D-Formen für die Rekonstruktion und das Rendern trainieren, verallgemeinert sich ihr Netzwerk gut auf ungesehene Objekte.

2.7.1 Rückschlüsse auf Forschungsstand

Wie in den vergleichbaren Arbeiten bereits dargestellt, beschäftigen sich aktuelle Forschungsarbeiten im Gebiet der Posenschätzung von Objekten auf Bildern mittels Deep Learning intensiv mit der Erstellung eines geeigneten Datensatzes. Der aktuelle Forschungsstand ist dahingehend soweit fortgeschritten, dass synthetisch erzeugte Bilder für ein erfolgreiches Training eines neuronalen Netzes genutzt werden können [48, 52, 12]. Das Verfahren der dreidimensionalen Objekterkennung mittels Deep Learning stößt aktuell jedoch an seine Grenzen, wenn es um die Erkennung von neuen oder vielen verschiedenen zu erkennenden Objekten geht. In diesem Zusammenhang muss einerseits ein Training für jedes zu erkennende Objekt einzeln durchgeführt werden, andererseits ist die Erkennung von mehreren Objekten gleichzeitig sehr rechenintensiv und aus diesem Grund durch die verwendete Hardware limitiert. Nur der Ansatz von Keunhong Park et al. [34] zeigt aktuell eine vielversprechende Variante neue Objekte schnell zu erkennen. Weiter konnten bei der Recherche aktuell keine Untersuchungen hinsichtlich einer Unterteilung von Objekten in Objektklassen gefunden werden. Hierbei ist interessant welche Objektklassen besonders gut oder besonders schlecht für die Posenschätzung mittels Deep Learning geeignet sind - was eine der grundsätzlichen Fragestellungen für diese Arbeit ist.

2.8 Zusammenfassung Analyse

Die Analyse hat einen Überblick über die wichtigen Teilgebiete dieser Arbeit gegeben. Angefangen bei Augmented Reality, weiter zur Bildverarbeitung über Computergrafik bis zur Annotation von Objekten. Bei der Bildverarbeitung wurde dargestellt, welche Bildarten in Verbindung mit dieser Arbeit als wertvolle Datengrundlage für das Training eines neuronalen Netzes verwendet werden können. Dazu gehören RGB-Bilder, die eine Umgebung realistisch in vielen Farben darstellt, sowie Segmentierungs- und Tiefenbilder, die Objekte innerhalb eines Bildes hervorheben und aus diesem Grund eine gute Datenquelle für einen Trainingsdatensatz darstellen. Weiter wurde über die Computergrafik als Methode zur Darstellung und Erzeugung von synthetischen realistisch wirkenden Bildern berichtet. In diesem Zusammenhang wurde dargestellt, welche Kriterien bei der Erstellung von realistisch wirkenden Bildern zu berücksichtigen sind und wie sich diese Kriterien auf den Erstellungsprozess auswirken. Dabei wurde hervorgehoben, dass ein hoher Anspruch an die Qualität stets auf Kosten der Berechnungsdauer geht, wobei aktuelle Spiele Engines einen guten Mittelweg für die Erzeugung von ausreichend realistisch wirkenden Bildern in akzeptabler Zeit bieten.

Bezüglich der Annotation von Objekten wurden analytische Ansätze und Ansätze des maschinellen Lernens vorgestellt. Zu den analytischen Ansätzen gehören die SIFT, SURF und ORB Deskriptoren, welche bereits gute Ergebnisse bei der Objekterkennung liefern. Gleichzeitig bestehen deren Schwächen in der Erkennung von verdeckten und teilweise veränderten Objekten. Das maschinelle Lernen ermöglicht das Umgehen dieser Schwächen durch die gewonnene Fähigkeit der Generalisierung und ist die für diese Arbeit verwendete Methode zur Objekterkennung. Weiter wurde bezüglich des maschinellen Lernens die Verwendung von Convolutional Neural Networks innerhalb des Themengebiets des Deep Learning als ”Stand der Technik”-Lösung für moderne Posenschätzung von Objekten erläutert.

Weiter wurde der Begriff Reality Gap eingebracht, welcher den Unterschied zwischen realen und synthetischen Daten beschreibt. Um diesen Unterschied zu verringern wurde auf die essenzielle Notwendigkeit von Data Augmentation hingewiesen, welche eine der wichtigsten Faktoren für diese Arbeit ist. In diesem Zusammenhang wurde der Begriff der Domänen-Randomisierung vorgestellt, welcher mithilfe der Bildverarbeitung und der Computergrafik erfolgreich eingesetzt werden kann, um neuronale Netz zu zwingen, die wesentlichen Merkmale des zu erkennenden Objektes zu lernen. Außerdem wurde auf

die Verwendbarkeit von randomisierten Daten im Vergleich zu fotorealistischen Daten hingewiesen.

Im Anschluss folgte eine Analyse über die Messung zur Qualität von Posenerkennungen, welche die Wahl der in dieser Arbeit verwendeten Messmethode mithilfe eines 'Advanced Realtime Tracking'-Systems begründet.

Abschließend wurden vergleichbare Arbeiten im Themengebiet der Posenschätzung mittels Deep Learning vorgestellt und Rückschlüsse auf den aktuellen Forschungsstand gezogen. Bei der Recherche konnten keine Untersuchungen hinsichtlich einer Unterteilung von Objekten in Objektklassen gefunden werden. Somit stellt sich die Frage, inwieweit die Posenschätzung mittels Deep Learning für Objekte geeignet ist und deren Unterscheidbarkeit beispielsweise nur in fünf Freiheitsgraden oder nur drei Freiheitsgraden möglich ist. Diese Fragestellung wird in dieser Arbeit mithilfe des NDDS [44] von Nvidia und des Deep Object Pose Estimation Netzes [47] von Jonathan Tremblay et al. analysiert werden.

3 Methodik und Forschung

In diesem Kapitel wird der Versuchsaufbau zur Messung und Validierung der Genauigkeit des Deep Object Pose Estimation Netzwerks auf Basis der eigens erzeugten synthetischen Daten erläutert. Hierzu gehört die Wahl der ausgesuchten Objekte aus einer Objektklasse inklusive der Begründung, weshalb speziell diese Objektklassen für die Messung gewählt wurden. Außerdem wird in diesem Kapitel die Durchführung der Experimente beschrieben und analysiert.

3.1 ART-System als Benchmark

Als Maßstab für den Vergleich von Leistungen wird ein Advanced Realtime Tracking - System[1] verwendet. Durch eine hohe Präzision und wenig Schwankungen beim Tracking bietet diese Technologie eine gute Benchmarkfähigkeit. Beim ART-System werden Objekt-Tracker verwendet (siehe Abbildung 3.1) deren Position durch acht in einem Würfel angeordnete Kameras berechnet und verfolgt wird. Der für diese Arbeit verwendete Objekt-Tracker besteht aus vier Lichtreflektoren, die jeweils als Marker für das zu verfolgende Objekt dienen. Durch eine individuelle Zusammenstellung der Marker kann ein Tracker von dem ART-System stets identifiziert werden. Um die Orientierung eines Objekt-Trackers messen zu können, müssen mehrere ($>= 3$) dieser Marker bei festgelegter Geometrie angeordnet werden.

Der Objekt-Tracker befindet sich während der Messung innerhalb der Traverse im CSTI[2] im Sichtfeld der oben erwähnten acht Tracking-Kameras. Diese tasten ein bestimmtes Volumen ab und deflektieren das Licht, das von den Markern kommt. Ihre Bilder werden verarbeitet, um potenzielle Markerpositionen (in Bildkoordinaten, 2-DOF) mit hoher Genauigkeit zu identifizieren und zu berechnen; eine mittlere Genauigkeit von 0,04 Pixeln ist bei ART-Tracking-Systemen Standard.

[1] https://ar-tracking.com/, 15.09.2020
[2] https://csti.haw-hamburg.de/, 15.09.2020

Diese 2 DOF-Daten werden kombiniert, um 3 DOF-Positionen einzelner Marker oder 6 DOF-Positionen starrer Anordnungen mehrerer Marker (Objekt-Tracker) zu berechnen. Dazu sind einige zusätzliche Informationen über das Trackingsystem notwendig, die zuvor in Kalibrierprozessen gesammelt werden müssen: Position und Orientierung der Trackingkameras sowie die Geometrie der Rigid Bodies (d. h. die Positionen der Marker innerhalb eines Körpers).

Das Ergebnis jeder Messung sind Koordinaten, die die Position der Marker und damit die Position und Orientierung des Objekt-Trackers, der die Marker trägt, beschreiben.

Abbildung 3.1: ART-Tracker

3.1.1 ART-Leistungsübersicht

Da sich je nach Kalibrierung die Leistung der Tracker unterscheiden kann und zur Vermeidung von Fehlinterpretationen für spätere Messungen, zeigt Tabelle 3.1 das durchschnittliche Messwertrauschen für alle sechs Freiheitsgrade des für jede Messung verwendeten Tracker-Objektes. Graphen, die das Messwertrauschen der einzelnen Achsen darstellen, sind im Anhang zu finden (siehe A.1).

Achse	Durchschnittliches Messwertrauschen	Dimension
X-Position	-0.000003125000000103462	Meter
Y-Position	-0.00000167999999855728	Meter
Z-Position	0.000007015000000028041	Meter
X-Rotation	-0.0026325000000079735	Grad
Y-Rotation	0.0016129999999997845	Grad
Z-Rotation	-0.0011185000000197576	Grad

Tabelle 3.1: ART-Leistungsübersicht

Anhand der Tabelle kann man sehen, dass die Genauigkeit der dreidimensionalen Positionsbestimmung mit Schwankungen im Mikrometerbereich sehr stabil ist. Die Genauigkeit der Rotation hingegen fällt etwas geringer aus, nichtsdestotrotz sind die Schwankungen in den einzelnen Winkeln mit unter 0.003 Grad noch sehr stabil und somit für das Messvorhaben für diese Arbeit geeignet.

3.2 Objektklassen

In dieser Arbeit wird die Leistung der Deep Object Pose Estimation unter anderem mittels drei verschiedener Objektklassen unterschieden. Die Unterschiede der Objektklassen beziehen sich hierbei auf die Erkennbarkeit einzelner Objekte in unterschiedlichen Freiheitsgraden. In diesem Zusammenhang gehört ein Objekt in die 6-DOF Objektklasse, wenn die eindeutige Erkennung der Pose eines Objektes aus allen sechs Freiheitsgraden möglich ist. Ist die Pose eines Objektes aus einer Richtung nicht eindeutig zu erkennen, gehört dieses Objekt in die 5-DOF Objektklasse. Dies ist beispielsweise der Fall, wenn ein Objekt um die y-Achse gedreht wird, wobei in einzelnen Momentaufnahmen kein vorne und hinten des Objektes bestimmt werden kann. Ist nur die dreidimensionale Position eines Objektes, jedoch nicht die Orientierung des Objektes im Raum zu erkennen, so gehört das Objekt zur 3-DOF Objektklasse. Durch die Klassifizierung lässt sich auf die Generalisierbarkeit des trainierten neuronalen Netzes bei der Erkennung der Pose eines Objektes schließen.

Abbildung 3.2 zeigt alle verwendeten Messobjekte. Zu sehen ist hierbei eine Cheez-It Cracker-Box, welche durch ihren individuellen Aufdruck aus allen sechs Freiheitsgraden zu erkennen ist und somit zur 6-DOF Objektklasse gehört. Weiter wird eine Bierflasche für die 5-DOF Objektklasse verwendet, da diese Flasche in der y-Achse von allen Seiten gleich aussieht. Als 3-DOF Objektklasse wird eine Billardkugel verwendet.

Abbildung 3.2: Alle Messobjekte

3.3 Aufbau der Experimente

Für den Aufbau der Experimente wird ein ART Objekt-Tracker an dem für die Messungen verwendeten Objekt platziert. Außerdem wird ein ART-Tracker auf einer Webcam platziert, welche den Input für das trainierte DOPE-Model liefert. Abbildung 3.3 zeigt den beschrieben Aufbau. Allgemein ist der Messaufbau in drei verschiedene Ansichten

Abbildung 3.3: Allgemeiner Messaufbau

unterteilt. Bei der ersten Ansicht ist das Messobjekt frontal und frei erkennbar, bei der

zweiten Ansicht wird das Messobjekt etwas rotiert und bei der letzten Ansicht wird das Messobjekt von anderen Objekten teilweise abgedeckt. Diese Unterteilung ermöglicht es, mehr Aussagen über das Verhaltens des jeweiligen trainierten DOPE-Modelles zu treffen. In diesem Zusammenhang kann mit der frontalen Ansicht geprüft werden, ob das Messobjekt überhaupt erkannt wird, da die trainierten DOPE-Modelle hierbei die leichteste Aufgabe haben. Dies hat sich durch empirische Beobachtung bereits während erster Versuche gezeigt. Fallen die Ergebnisse bei der frontalen Ansicht schlecht aus, ist zu vermuten, dass die beiden anderen Ansichten noch schlechter ausfallen. Mithilfe der zweiten Ansicht können bessere Aussagen über die Erkennung der Orientierung getroffen werden. Bei der dritten Ansicht wird auf den Vorteil von künstlichen Intelligenzen gegenüber analytischen Methoden geprüft, da analytische Methoden wie SIFT [25], SURF [16], ORB [36] (siehe vorheriges Kapitel: 'Analyse') stark anfällig gegenüber einer Verdeckung der Objekte sind. Während der Messungen variierte der Abstand zwischen der Kamera und dem Messungsobjekt zwischen 1,5 m und 3,0 m. Die Posen, welche das empirisch beobachtete durchschnittliche Erkennungsverhalten repräsentierten wurden gemessen und werden im Laufe dieses Kapitels vorgestellt.

Weiter wurde eine Java-Springbootanwendung entwickelt, welche einerseits die Posen-Daten des ART-Systems und andererseits die Pose-Daten aus der Deep Object Pose Estimation entgegennimmt und diese Daten einer eigens entwickelten Angular Frontend-Anwendung zur Verfügung stellt. Die Angular Frontend-Anwendung ruft die jeweiligen Posen-Daten ab und visualisiert diese innerhalb einer 3D-Szene (siehe Abbildung 3.4). Weiter werden die Daten innerhalb der Frontend-Anwendung an ein Analyse-Werkzeug weitergeleitet, welches die Posen-Daten sowie einen Posen-Vergleich mithilfe von Diagrammen visualisiert.

Insgesamt dient die Frontendanwendung dazu, die Koordinatensysteme des ART-Systems mit den berechneten Posen aus den DOPE-Modellen zu synchronisieren, um den Vergleich der Leistungen beider Tracking-Methoden zu ermöglichen. Ausgeführt wurden die Experimente auf einem Desktop PC mit zwei Zota 11 GB D5X GTX 1080 TI, einem Intel Core i7-7700K und 64 Gigabyte Arbeitsspeicher. Als Betriebssystem wurde Ubuntu in der Version 16.04.6 LTS verwendet.

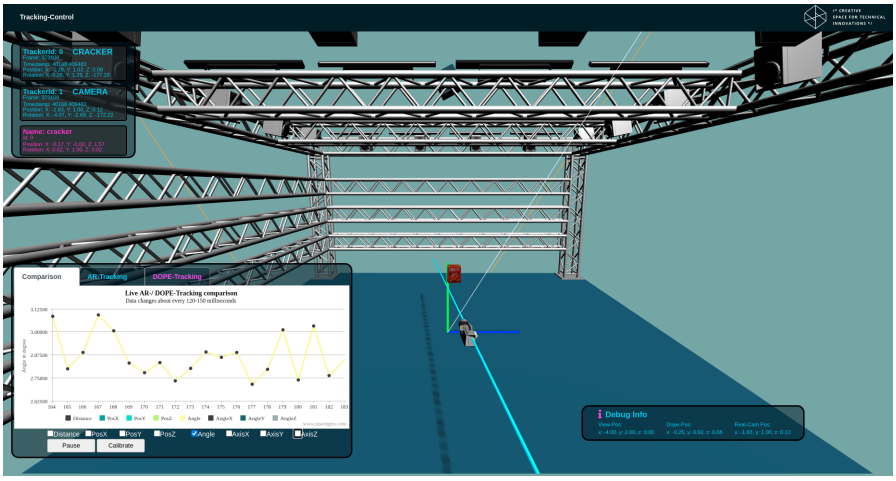

Abbildung 3.4: Frontendanwendung zur Messungaufgabe

3.4 Experiment - Selbst erstellt vs. NVIDIA

Wie im Kapitel der Analyse erwähnt, stellt NVIDIA die von ihnen erstellten Modelle öffentlich zur Verfügung. Dieses Experiment dient zur Validierung, dass der für diese Arbeit selbst erstellte Trainingsdatensatz sowie die selbst erstellten Modelle mit dem Datensatz von NVIDIA und deren Modelle vergleichbar ist. Außerdem wird durch dieses Experiment die Glaubhaftigkeit der weiteren Messvorhaben in dieser Arbeit gesetzt. Beide Modelle wurden für 60 Epochen trainiert. Die Datensätze für die Trainings beinhalteten die gleiche Anzahl an fotorealistischen, wie auch an Domänen randomisierten Bildern. Somit liegt der Unterschied der Datensätze ausschließlich bei der Erstellung.

3.4.1 Aufbau

Die Messungen für dieses Experiment fanden ausschließlich mithilfe einer Cheez-It Cracker-Box als Messobjekt statt. In diesem Zusammenhang wurde nur das Verhalten der Objekterkennung im Bezug auf die 6-DOF Objektklasse geprüft. Der Aufbau wurde auf das selbst erstellte Modell angepasst, weil alle anderen Messungen in dieser Arbeit ausschließlich auf Basis von selbst erstellten Modellen liegen.

Abbildung 3.5 zeigt den Aufbau aller drei Ansichten zur Gegenüberstellung des Verhaltens bei der Erkennung der Pose. Ganz links ist die frontale Ansicht, mit einen freien Blick auf das Messobjekt. In der Mitte wurde der Tisch rotiert, um wie oben beschrieben

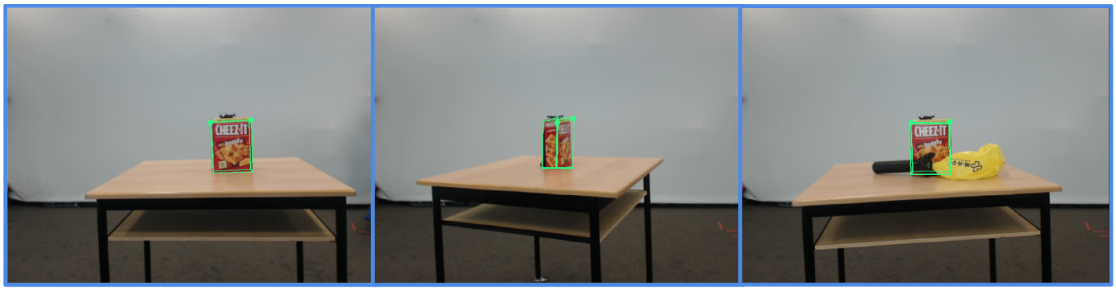

Abbildung 3.5: Aufbau des Vergleichs - selbst erstellt vs. NVIDIA

das Verhalten der Erkennung der Pose des Messobjektes bei einer Rotation zu prüfen. Rechts wurde das Messobjekt ein wenig verdeckt, sodass beide trainierten DOPE-Modelle (das selbst erstellte und das von Nvidia) das Messobjekt erkennen konnten. Hierzu sei erwähnt, dass ein paar Aufbauten mit Verdeckung nicht gemessen werden konnten, weil das DOPE-Modell von Nvidia das Messobjekt nicht erkannte (siehe Abbildung 3.6).

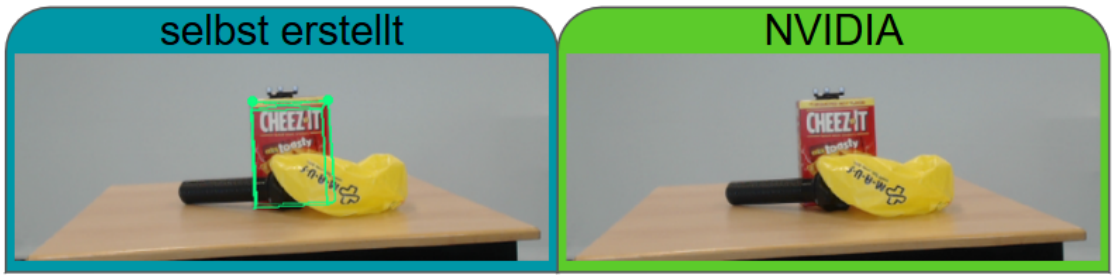

Abbildung 3.6: Halbe Verdeckung - selbst erstellt vs. NVIDIA

Weiter sei an dieser Stelle auf nicht beherrschbare Messbedingen hingewiesen, welche einen Einfluss auf die Messergebnisse haben können. Hierzu gehören nicht beherrschbare Einflüsse wie die Beleuchtung durch die Fenster in dem Raum, in dem die Messungen vorgenommen wurden oder die Vibration des Bodens beim Einfahren einer Bahn in die nahe gelegene U-Bahn-Station.

Insgesamt wurden zu allen drei Ansichten die Daten von jeweils 100 der durch DOPE erkannten Posen gesammelt und dem ART-System gegenübergestellt.

3.4.2 Analyse - Leistung der Modelle

Für eine aussagekräftige Analyse der Posenerkennung beider DOPE-Modelle, wurden jeweils das durchschnittliche Messwertrauschen in allen Ansichten ermittelt und können der Tabelle 3.2 entnommen werden.

Frontal

Bei Betrachtung der Tabelle 3.2 ist das Messwertrauschen der Positionen in der Frontalansicht sowohl im Fall des von Nvidia trainierten DOPE-Modells als auch im Fall des selbst trainierten DOPE-Modells im Millimeterbereich. Zur Erinnerung sei erwähnt, dass das Messwertrauschen des ART-Systems im Mikrometerbereich liegt und somit nur einen sehr geringen Einfluss auf die hier dargestellten Messergebnisse hat. Die größten Ausschläge der Position sind in beiden Modellen in der z-Achse vorhanden, welche in der Realität die Entfernung des Messobjektes zur Kamera darstellt. Weiter zeigt die Messung, dass das Messwertrauschen der Position des selbst trainierten Modells in der x-Achse um ca. 0,06 Millimeter, in der y-Achse um ca. 0,27 Millimeter und in der z-Achse um ca. 0,74 Millimeter geringer ausfällt.

Auch bei der Betrachtung der Werte im Bezug zur Orientierung lässt sich ein geringeres Messwertrauschen in allen Achsen beim selbst trainierten DOPE-Modell beobachten. In diesem Zusammenhang fällt das Messwertrauschen in der x-Achse um ca. 0.03°, in der y-Achse um ca. 0.07° und in der z-Achse um ca. 0.06° geringer aus.

Die Aktualisierungsrate während der Erkennung variierte in beiden Modellen zwischen 130 bis 200 Millisekunden, durchschnittlich lag sie bei 155 Millisekunden.

Rotiert

Bei den nach der Rotation des Messobjektes gemessenen Werten ist zu sehen, dass das Messwertrauschen der Position bei beiden Modellen höher ausfällt. Auch hier zeigt die Messung, dass das Messwertrauschen der Position des selbst trainierten Modells geringer ausfällt. So ist das Messwertrauschen in der x-Achse um ca. 4,58 Millimeter, in der y-Achse um ca. 1,06 Millimeter und in der z-Achse um ca. 35,01 Millimeter geringer. Hierzu sei erwähnt, dass sich während der Messung des Nvidia-DOPE-Modells ein häufiges und

Achse	Durchschnittliches Messwertrauschen		Dim.
Frontal			
	selbst erstellt	**NVIDIA**	
X-Position	0.0000413	0.0001044	Meter
Y-Position	0.0000224	0.0002957	Meter
Z-Position	0.0001130	0.0008625	Meter
X-Rotation	0.0019524	-0.0336966	Grad
Y-Rotation	0.0000036	0.0073656	Grad
Z-Rotation	0.0255939	-0.0360765	Grad
Rotiert			
X-Position	0.0001558	0.0047443	Meter
Y-Position	0.0000066	0.0010719	Meter
Z-Position	0.0002709	0.0352838	Meter
X-Rotation	0.0020177	4.9967495	Grad
Y-Rotation	0.0073003	13.332980	Grad
Z-Rotation	0.0045554	1.6644013	Grad
Verdeckt			
X-Position	0.0000016	0.0000286	Meter
Y-Position	0.0002386	0.0006142	Meter
Z-Position	0.0015117	0.0701241	Meter
X-Rotation	0.0045084	0.0517956	Grad
Y-Rotation	0.0061118	0.0092937	Grad
Z-Rotation	0.0688651	0.2637075	Grad

Tabelle 3.2: Leistungsübersicht (Messwertrauschen) - selbst erstellt vs. NVIDIA

deutliches Springen der Position beobachten ließ, was den höheren Wert in der z-Achse erklärt.

Bei der Betrachtung der Werte im Bezug zur Orientierung lässt sich erneut ein geringeres Messwertrauschen in allen Achsen beim selbst trainierten DOPE-Modell im Vergleich zum Nvidia-DOPE-Modell beobachten. So fällt das Messwertrauschen in der x-Achse um ca. 4.99°, in der y-Achse um ca. 13.32° und in der z-Achse um ca. 1.66° geringer aus.

Die Aktualisierungsrate während der Erkennung variierte in den zwei Modellen unterschiedlich. Das selbst erstelle Modell variierte von 130 bis 200 Millisekunden, wobei der Durchschnitt bei ca. 156 Millisekunden lag. Das Modell von Nvidia hingegen aktualisierte zwischen 121 und 187 Millisekunden, wobei der Durchschnitt auch hier bei ca. 156 Millisekunden lag.

Verdeckt

Auch bei den Messungen in der Ansicht, in der das Messobjekt teilweise verdeckt wurde, ist zu sehen, dass das Messwertrauschen der Position bei beiden Modellen höher ausfällt. Wie zuvor zeigt die Messung, dass das Messwertrauschen der Position des selbst trainierten Modells geringer ausfällt. So ist das Messwertrauschen in der x-Achse um ca. 0,02 Millimeter, in der y-Achse um ca. 68,61 Millimeter und in der z-Achse um ca. 35,01 Millimeter geringer. Wie zuvor bei der rotierten Ansicht konnte bei der Messung des Nvidia-DOPE-Modells ein häufiges Springen der Position beobachtet werden, was die deutlich höheren Werte in der y- und z-Achse erklärt.

Die Werte im Bezug zur Orientierung zeigen im Vergleich zur frontalen Ansicht ein höheres Messwertrauschen in fast allen gemessenen Werten. Das größte Messwertrauschen lässt sich bei beiden Modellen in der z-Achse beobachten. Gleichzeitig ist erneut ein geringeres Messwertrauschen in allen Achsen beim selbst trainierten DOPE-Modell im Vergleich zum Nvidia-DOPE-Modell festzustellen. Hierbei fällt das Messwertrauschen in der x-Achse um ca. 0.05°, in der y-Achse um ca. 0.003° und in der z-Achse um ca. 0.19° geringer ausfällt.

Die Aktualisierungsrate während der Erkennung variierte in den zwei Modellen abermals nahezu identisch. So variierte die Aktualisierungsrate beider Modell zwischen 110 bis 186 Millisekunden mit einem Durchschnitt von ca. 156 Millisekunden.

3.4.3 Analyse - Vergleich zum ART-System

Weil das Messwertrauschen bei der Erkennung der Pose keine Aussage über die Korrektheit der erkannten Pose beinhaltet, erfolgt in diesem Abschnitt der Vergleich der Modelle gegenüber dem ART-System. Weiter wird von einem ART-Objekt gesprochen, welches das 3D-Objekt in der Frontend-Szene ist, das an den verwendeten ART-Tracker gekoppelt wurde.

Zur Vereinfachung der Interpretation der im Folgenden analysierten Werte, stellt Abbildung 3.7 die Unterschiede zwischen den gemessenen Werten visuell dar. Innerhalb der Abbildung gibt es zwei farblich markierte Reihen - blau für Bilder der Posen aus dem selbst erstellten Modell und grün für Bilder der Posen aus dem von Nvidia erstellten

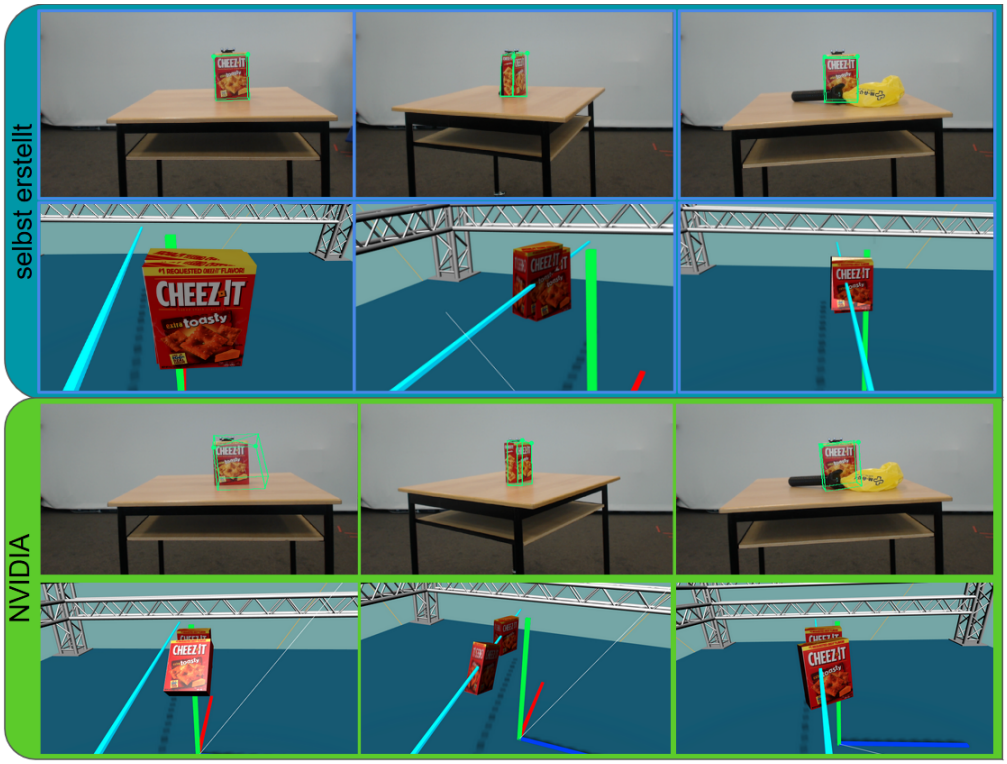

Abbildung 3.7: Visuelle Darstellung der Unterschiede

Modell. Die Bilder innerhalb der farblich markierten Reihen stellen jeweils die gemessenen Ansichten (frontal, rotiert und verdeckt) dar. Gleichzeitig existieren zu jeder Ansicht zwei Bilder, wobei es sich bei dem oberen Bild stets um einen Screenshot aus dem ROS-Framework mit eingerechnetem 3D-Rahmen handelt. Bei dem unteren Bild handelt es sich um einen Screenshot aus der Szene der eigens entwickelten Frontend-Anwendung.

Frontal

Die Abbildung 3.8 zeigt die im Folgenden analysierten Werte der Distanz und der Winkelunterschiede im Zeitverlauf aus der frontalen Ansicht. Die Abszisse stellt den Zeitpunkt dar, zu dem eine Aktualisierung der DOPE-Pose stattfand. Hierbei ist zu erkennen, wie genau die Position als auch die Orientierung mittels beider Modelle berechnet wurde. Desto geringer die Zahlen in der Ordinate, desto genauer ist die Erkennung.

Bei Betrachtung der Tabelle 3.3 ist zu sehen, dass die gemessene Distanz zwischen dem

Abbildung 3.8: Vergleich ”frontal” - Distanz (oben) - Winkel (unten) mit Messwertrauschen

ART-Objekt und der Position aus dem selbst erstellten Modell mit ca. 0,01 m sehr gering ist. Die gemessene Distanz zwischen dem ART-Objekt und der Position aus dem von Nvidia erstellten Modell liegt hingegen bei ca. 0,43 m. Die größten Unterschiede der berechneten Position beider Modellen liegen jeweils in der x-Achse, welche in dieser Messsituation die Linie zwischen dem ART-Tracker und der Webcam darstellt.

Bei der Betrachtung der Werte im Bezug zur Orientierung fällt auf, dass die berechneten Winkel aus dem Nvidia-Modell sich mit einem Unterschied von ca. 46.50° stark von dem ART-Objekt unterscheiden. Der Winkelunterschied zwischen dem berechneten Winkel aus dem selbst erstellten Datensatz und dem ART-Objekt fällt mit ca. 4.62° deutlich geringer aus. Die größten Unterschiede des Eulerwinkels beider Modelle entstehen in der x-Achse, welche die vertikale Neigung vom Messobjekt zur Kamera entspricht (siehe Abbildung 3.7).

Rotiert

Die Abbildung 3.9 zeigt die im Folgenden analysierten Werte der Distanz und der Winkelunterschiede im Zeitverlauf aus der rotierten Ansicht. Die Abszisse stellt den Zeitpunkt dar, zu dem eine Aktualisierung der DOPE-Pose stattfand. Hierbei ist zu erkennen, wie genau die Position als auch die Orientierung mittels beider Modelle berechnet wurde. Desto geringer die Zahlen in der Ordinate, desto genauer ist die Erkennung.

Achse	Durchschnittlicher Unterschied		Dim.
Frontal			
	selbst erstellt	**NVIDIA**	
Distanz	0.0161372	0.4310302	Meter
X-Position	0.0168109	-0.4278047	Meter
Y-Position	0.0012464	0.0385768	Meter
Z-Position	-0.0046358	0.0074128	Meter
Winkel	4.6228383	46.5080794	Grad
X-Rotation	-3.7881993	12.5967316	Grad
Y-Rotation	2.2776416	5.3440218	Grad
Z-Rotation	-0.0058136	-0.3662340	Grad
Rotiert			
Distanz	0.0508315	0.15762178	Meter
X-Position	-0.0446703	-0.1522269	Meter
Y-Position	0.0228751	0.0199643	Meter
Z-Position	-0.0110827	0.0158002	Meter
Winkel	7.2102080	26.5485304	Grad
X-Rotation	-1.6664541	-9.4544188	Grad
Y-Rotation	1.1536715	4.3756257	Grad
Z-Rotation	4.5110652	24.0925376	Grad
Verdeckt			
Distanz	0.0106811	0.0971359	Meter
X-Position	-0.0060869	-0.0758298	Meter
Y-Position	0.0062364	0.0182819	Meter
Z-Position	-0.0052515	0.0083850	Meter
Winkel	22.0715449	17.9819282	Grad
X-Rotation	-2.2572931	-5.4259796	Grad
Y-Rotation	3.9311289	-2.5959900	Grad
Z-Rotation	0.4587213	1.3143509	Grad

Tabelle 3.3: Unterschied zum ART System - selbst erstellt vs. NVIDIA

Bei den nach der Rotation des Messobjektes gemessenen Werten ist zu sehen, dass die gemessene Distanz zwischen dem ART-Objekt und der Position aus dem selbst erstellten Modell mit ca. 0,05 m im Vergleich zur frontalen Ansicht verschlechtert hat. Die gemessene Distanz zwischen dem ART-Objekt und der Position aus dem von Nvidia erstellten Modell liegt hingegen bei ca. 0,15 m, womit sich die geschätzte Distanz im Gegensatz zur frontalen Ansicht deutlich verringert hat. Die größten Unterschiede der berechneten Position beider Modellen liegen wieder jeweils in der x-Achse.

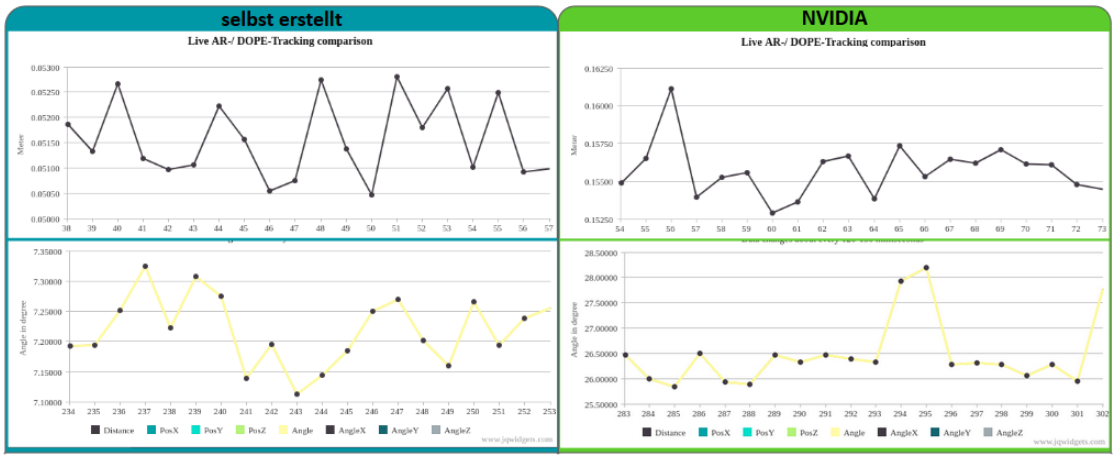

Abbildung 3.9: Vergleich "rotiert"- Distanz (oben) - Winkel (unten) mit Messwertrauschen

Bei der Betrachtung der Werte im Bezug zur Orientierung fällt auf, dass die berechneten Winkel aus dem Nvidia-Modell sich mit einem Unterschied von 26.54° wieder stark von dem ART-Objekt unterscheiden. Gleichzeitig ist auch hier eine deutlich geringere Abweichung gegenüber der frontalen Ansicht festzustellen. Der Winkelunterschied zwischen dem berechneten Winkel aus dem selbst erstellten Datensatz und dem ART-Objekt fällt mit 7.21° wieder geringer aus. Die größten Unterschiede der Eulerwinkel beider Modelle entstehen in der z-Achse.

Verdeckt

Die Abbildung 3.10 zeigt die im Folgenden analysierten Werte der Distanz und der Winkelunterschiede im Zeitverlauf aus der verdeckten Ansicht. Die Abszisse stellt den Zeitpunkt dar, zu dem eine Aktualisierung der DOPE-Pose stattfand. Hierbei ist zu erkennen, wie genau die Position als auch die Orientierung mittels beider Modelle berechnet wurde. Desto geringer die Zahlen in der Ordinate, desto genauer ist die Erkennung.

Bei den Messungen in der Ansicht, in der das Messobjekt teilweise verdeckt wurde, ist zu sehen, dass die gemessene Distanz zwischen dem ART-Objekt und der Position aus dem selbst erstellten Modell mit ca. 0,01 m wieder sehr gering ist. Die gemessene Distanz zwischen dem ART-Objekt und der Position aus dem von Nvidia erstellten Modell liegt hingegen bei ca. 0,09 m. Somit ist die geschätzte Distanz des Nvidia-Modells wieder geringer im Vergleich zur frontalen Messansicht.

Abbildung 3.10: Vergleich "verdeckt"- Distanz (oben) - Winkel (unten) mit Messwertrauschen

Die Werte im Bezug zur Orientierung zeigen, dass die berechneten Winkel aus beiden Modellen relativ stark von der Orientierung des ART-Objektes abweichen. Das selbst trainierte Modell berechnet Winkel, die sich um insgesamt 22.07° vom ART-Objekt unterscheiden und die vom Nvidia-Modell errechneten Winkel unterscheiden sich insgesamt um 17.98°. Eine klare Tendenz, in welchen der von den Modellen berechneten Eulerwinkel abgewichen wird ist nicht zu erkennen, da beide Modelle in unterschiedlichen Achsen ausschlagen.

3.4.4 Evaluation

Nach der vorigen methodischen Ergebung der Daten aus dem Experiment mit der Gegenüberstellung der Leistung beider vorgestellten Modelle zur Erkennung der Pose der Cracker-Box, erfolgt nun eine Evaluation, um die Untersuchung nachvollziehbar und überprüfbar zu halten.

Beim Leistungsvergleich der einzelnen Modelle konnte festgestellt werden, dass das Messwertrauschen der gemessenen Positions- als auch Orientierungswerte in allen drei gemessenen Ansichten stabil genug ist, um Aussagen über ein allgemeines Verhalten bei der Erkennung des Messobjektes zu treffen.

In der frontalen Messansicht war zu sehen, dass die berechnete Pose aus dem Nvidia-Modell von der realen Pose in Position und Orientierung stark abwich. Dieses eigenartige Verhalten trat während des Aufbaus der frontalen Messansicht aus verschiedenen Positionen häufig auf. Aus diesem Grund wurde das eigenartig wirkende Verhalten bei der Analyse berücksichtigt. Hierzu sei erwähnt, dass das Verhalten während der Messungen je nach Beleuchtung, Hintergrund und Positionierung stark variierte. Gleichzeitig musste für alle weiteren Aufbauten ein einfacher und reproduzierbarer Rahmen für einen Aufbau gefunden werden. Nichtsdestotrotz ist hierbei abzuleiten, dass das selbst erstellte Modell resistenter gegenüber der frontalen Ansicht war und deutlich korrektere Ergebnisse geliefert hat.

In der rotierten Messansicht performte das selbst erstellte Modell wieder besser. Gleichzeitig zeigte sich im Vergleich zur frontalen Ansicht, dass die Berechnung der Position und der Orientierung ungenauer wurde. Das Modell von Nvidia berechnete die Position in der rotierten Ansicht jedoch besser als in der frontalen Ansicht.

Bei der Analyse der Werte aus der verdeckten Ansicht wiesen beide Modelle Probleme bei der Berechnung der Orientierung auf. Weiter ist aufgefallen, dass das selbst erstellte Modell resistenter gegen die Verdeckung des Messobjektes war. In diesem Zusammenhang ließ sich beim Aufbau der verdeckten Messansicht beobachten, dass das Modell von Nvidia das Messobjekt teilweise gar nicht erkannte. Weiter ist aufgefallen, dass die z-Achsen der Positionsvektoren beider Modelle die größten Schwankungen aufwiesen. Dieses Verhalten lässt unter anderem annehmen, dass die Modelle aufgrund der Verdeckung Schwierigkeiten in der Schätzung der Größe der Messobjekte haben. Hierzu wären Optimierungen vorstellbar, in denen der Grad der Verdeckung mit in das Training des Netzes einfließen.

Insgesamt hat die Analyse der einzelnen Ansichten gezeigt, dass das selbst erstellte Modell gegenüber dem öffentlich zur Verfügung gestellten Modell von Nvidia in allen Ansichten besser abschneidet. Als Grund für dieses Ergebnis kann nur die Annahme über den Datensatz getroffen werden, dass das Messobjekt möglicherweise seltener in dem Datensatz vorkommt. Was die Resistenz gegenüber der Verdeckung des Objektes betrifft, ist anzunehmen, dass das Messobjekt innerhalb des Datensatzes seltener von anderen Objekten verdeckt wurde. Dies kann nur bei einer detaillierten Analyse des von Nvidia verwendeten Datensatzes überprüft werden, was jedoch kein Teil dieser Arbeit ist.

Allgemein ist die grundlegende hohe Leistungsfähigkeit eines Trainings mittels des selbst erstellten Datensatzes bewiesen, sodass weitere Aussagen im Kontext dieser Arbeit gewissenhaft getätigt werden können.

3.5 Experiment - domänenrandomisiert & fotorealistisch

Im Kapitel der Analyse wurde auf die Arbeit von Jonathan Tremblay et. al [47] referenziert, in welcher beschrieben wird, dass das Problem der Realitätslücke bei der Erkennung der Objektpose mittels maschinellem Lernen durch eine einfache Kombination von domänenrandomisierten und fotorealistischen Daten erfolgreich überbrückt werden kann. Das in dieser Sektion anstehende Experiment widmet sich dieser Frage, um die Aussage von Jonathan Tremblay et. al, anhand für diese Arbeit selbst erstellter Modelle zu überprüfen. Bei den für das vorgestellte Experiment verwendeten Modellen handelt es sich einerseits um ein Modell, welches mit fotorealistischen als auch domänenrandomisierten Daten trainiert wurde (FRDR-Modell) andererseits um ein Modell, welches ausschließlich mit domänenrandomisierten Daten trainiert wurde (DRDR-Modell). In diesem Zusammenhang ist interessant, wie effizient das DRDR-Modell die Objektposen erkennt. Beide Modelle wurden für 30 Epochen trainiert. Die Datensätze, die für die Trainings der Modelle verwendet wurden, beinhalteten die gleiche Anzahl an Bildern. Somit liegt der Unterschied der Datensätze ausschließlich in der Einteilung FRDR (50% fotorealistich, 50% domänenrandomisiert) und DRDR (100% domänenrandomisiert).

3.5.1 Aufbau

Die Messungen für dieses Experiment fanden wie zuvor ausschließlich mithilfe einer Cheez-It Cracker-Box als Messobjekt statt. In diesem Zusammenhang wurde das Verhalten der Objekterkennung ausschließlich in Bezug auf die 6-DOF Objektklasse geprüft. Der Aufbau gestaltet sich wie der Messaufbau des Vergleichs des selbst erstellten Modells gegenüber dem Nvidia-Modell. Diesbezüglich wurden die Messungen wieder aus drei Ansichten (frontal, rotiert und verdeckt) vorgenommen (siehe Abbildung 3.5). Zuletzt wurden wieder die Daten aller drei Ansichten der jeweils 100 durch DOPE erkannten Posen gesammelt und dem ART-System gegenübergestellt.

3.5.2 Analyse - Leistung der Modelle

Für eine aussagekräftige Analyse der Posenerkennung beider Modelle wurden jeweils das durchschnittliche Messwertrauschen aus 100 Aufnahmen in allen Ansichten ermittelt und können der Tabelle 3.4 entnommen werden.

Achse	Durchschnittliches Messwertrauschen		Dim.
Frontal			
	FRDR	DRDR	
X-Position	0,0000221	0,0000107	Meter
Y-Position	0,0000085	0,0000086	Meter
Z-Position	0,0001719	0,0000685	Meter
X-Rotation	-0,0022526	0,1314620	Grad
Y-Rotation	0,0013358	0,0004932	Grad
Z-Rotation	0,0307935	-0,0086336	Grad
Rotiert			
X-Position	0,0000436	0,0000108	Meter
Y-Position	0,0000513	0,0000246	Meter
Z-Position	0,0006215	0,0001468	Meter
X-Rotation	0,0030347	0,0062718	Grad
Y-Rotation	0,0037802	0,0041682	Grad
Z-Rotation	0,0066627	0,0032958	Grad
Verdeckt			
X-Position	0,0009540	0,0000269	Meter
Y-Position	0,0001111	0,0001264	Meter
Z-Position	0,0919255	0,0002763	Meter
X-Rotation	14,4566014	0,0400781	Grad
Y-Rotation	25,9993484	0,0796473	Grad
Z-Rotation	4,3327917	0,0290305	Grad

Tabelle 3.4: Leistungsübersicht (Messwertrauschen) - FRDR vs. DRDR

Frontal

Bei Betrachtung der Tabelle 3.4 ist das Messwertrauschen der Positionen in der Frontalansicht sowohl im Fall des DRDR-Modells als auch im Fall des FRDR-Modells im Millimeterbereich. Die größten Ausschläge der Position sind in beiden Modellen in der z-Achse vorhanden, welche in der Realität die Entfernung des Messobjektes zur Kamera darstellt. Weiter zeigt die Messung, dass das Messwertrauschen der Position des DRDR-Modells in der x-Achse um ca. 0,01 Millimeter und in der z-Achse um ca. 0,10 Millimeter geringer ausfällt. Das Messwertrauschen in der y-Achse der Position liegt in beiden Modellen im Mikrometerbereich und befindet sich somit im Genauigkeitsbereich des ART-Systems.

Auch bei der Betrachtung der Werte in Bezug zur Orientierung lässt sich insgesamt ein geringeres Messwertrauschen in allen Achsen beider Modelle beobachten. Allerdings fällt das Messwertrauschen des DRDR-Modells in der x-Achse um ca. 0.13° Grad mi-

nimal höher als beim FRDR-Modells aus. Gleichzeitig ist das Messwertrauschen des DRDR-Modell in der y-Achse und in der z-Achse nicht nennenswert geringer als beim FRDR-Modell.

Die Aktualisierungsrate während der Erkennung variierte in beiden Modellen zwischen 108 bis 188 Millisekunden, durchschnittlich lag sie bei 153 Millisekunden.

Rotiert

Bei den nach der Rotation des Messobjektes gemessenen Werten ist zu sehen, dass sich das Messwertrauschen der Position bei beiden Modellen ähnlich zu dem der frontalen Ansicht verhält. Die Messung zeigt, dass das Messwertrauschen der Position beim DRDR-Modells allgemein ein wenig geringer ausfällt. So ist das Messwertrauschen in der x-Achse um ca. 0,03 Millimeter, in der y-Achse um ca. 0,02 Millimeter und in der z-Achse um ca. 0,47 Millimeter geringer.

Bei der Betrachtung der Werte in Bezug zur Orientierung ist das Messwertrauschen in beiden Modellen nahezu identisch. Die Werte in allen Achsen variieren hierbei nur zwischen ca. 0.003° und 0.007° und sind somit sehr stabil und auf kurze Distanz auch optisch kaum wahrnehmbar.

Auch die Aktualisierungsrate variierte in den zwei Modellen ähnlich zwischen 116 und 185 Millisekunden, wobei der Durchschnitt bei ca. 166 Millisekunden lag.

Verdeckt

In der Ansicht, wo das Messobjekt teilweise verdeckt wurde, fallen deutlichere Unterschiede in den gemessenen Werten zwischen den Modellen auf. So wurden beim FRDR-Modell die höchsten Ausschläge mit 90 Millimetern in der z-Achse der Position gemessen. Dies war auch während der Durchführung der Messungen durch ein Springen der im Frontend visuell dargestellten Position des FRDR-Modells zu beobachten. Dabei wechselte das visuell dargestellte Messobjekt mehrfach die Position zwischen der Position des ART-Objektes und der Position der Kamera. Im Fall des DRDR-Modells fielen die Ausschläge mit durchschnittlich ca. 0,2 Millimetern deutlich geringer aus.

Eine noch höhere Streuung der mittels des FRDR-Modell berechneten Werte konnte bei der Orientierung gemessen werden. Hierbei variierte das Messwertrauschen in der x-Achse um ca. 14.45°, in der y-Achse um ca. 26.00° und in der z-Achse um ca. 4.33°. Diese kontinuierliche Rotation ließ sich auch während der Durchführung der Messungen in der Frontendanwendung beobachten und werden im Abschnitt des Vergleichs zum ART-Tracking deutlicher hervorgehoben.

Die Aktualisierungsrate während der Erkennung variierte in den zwei Modellen abermals nahezu identisch. So variierte die Aktualisierungsrate beider Modell zwischen 100 bis 200 Millisekunden, mit einem Durchschnitt von ca. 150 Millisekunden.

3.5.3 Analyse - Vergleich zum ART-System

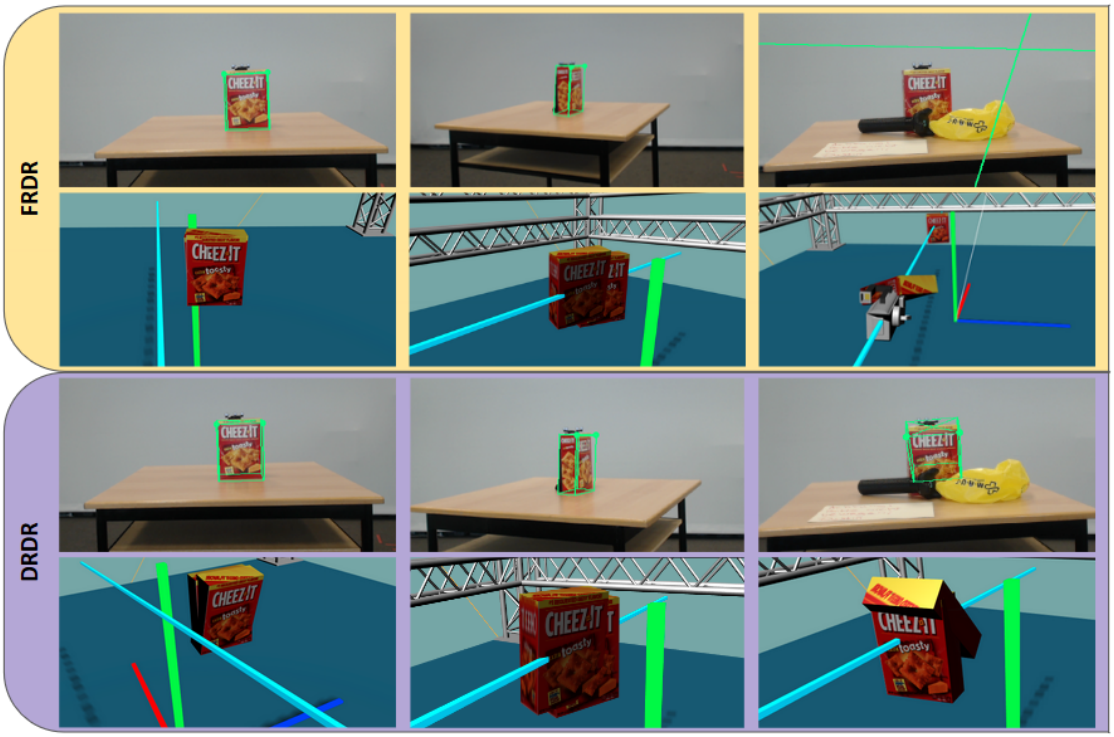

Abbildung 3.11: "FR vs. DR"- Visuelle Darstellung der Unterschiede

In diesem Abschnitt findet der Vergleich des FRDR- und des DRDR-Modelles gegenüber dem ART-System statt.

Zur Vereinfachung der Interpretation der im Folgenden analysierten Werte, stellt Abbildung 3.11 die Unterschiede zwischen den gemessenen Werten visuell dar. Innerhalb der Abbildung gibt es zwei farblich markierte Reihen - gelb für Bilder der Posen aus dem FRDR-Modell und lila für Bilder der Posen aus dem DRDR-Modell. Die Bilder innerhalb der farblich markierten Reihen stellen jeweils die gemessenen Ansichten (frontal, rotiert und verdeckt) dar. Gleichzeitig existieren zu jeder Ansicht zwei Bilder, wobei es sich bei dem oberen Bild stets um einen Screenshot aus dem ROS-Framework mit eingerechnetem 3D-Rahmen handelt. Bei dem unteren Bild handelt es sich stets um einen Screenshot aus der Szene Frontendanwendung.

Frontal

Abbildung 3.12: ”DR vs. FR”-Vergleich ”frontal”- Distanz (oben) - Winkel (unten) mit Messwertrauschen

Die Abbildung 3.12 zeigt einen Ausschnitt des Verlaufs der gemessenen Distanzwerte und Winkelunterschiede während der Messung in der frontalen Ansicht. Die Durchschnittswerte sind der Tabelle 3.5 zu entnehmen. Die Abszisse stellt den Zeitpunkt dar, zu dem eine Aktualisierung der DOPE-Pose stattfand. Hierbei ist zu erkennen, wie genau die Position als auch die Orientierung mittels beider Modelle berechnet wurde. Je geringer die Zahlen in der Ordinate, desto genauer ist die Erkennung.

Bei Betrachtung der Tabelle 3.5 ist zu sehen, dass die gemessene durchschnittliche Distanz zwischen dem ART-Objekt und der Positionen aus beiden Modellen sehr gering ist. Die gemessene Distanz zwischen dem ART-Objekt und der Position aus dem DRDR-Modell

liegt bei ca. 0,02 m und ist im Vergleich zum FRDR-Modell um ca. 0,017 m höher. Diese Unterschiede sind auch in der Abbildung 3.11 erkennbar. Hierbei sei erwähnt, dass die Position der Objekte immer dem Mittelpunkt des verwendeten Messobjektes entspricht.

Bei der Betrachtung der Werte in Bezug zur Orientierung fällt auf, dass die berechneten Winkel aus dem DRDR-Modell sich mit einem Unterschied von ca. 13.23° bemerkbar von dem ART-Objekt unterscheiden. Die berechnete Winkelabweichung des FRDR-Modells liegt bei 7.09°, womit die Erkennung der Orientierung beim FRDR-Modell um ca. 6.13° genauer gemessen wurde. Auch diese Ergebnisse lassen sich bei einer Betrachtung der Abbildung 3.11 nachvollziehen. Die größten Unterschiede der Eulerwinkel beider Modelle entstehen in der x-Achse, die der vertikalen Neigung vom Messobjekt zur Kamera entspricht.

Rotiert

Abbildung 3.13: ”DR vs. FR”-Vergleich ”rotiert”- Distanz (oben) - Winkel (unten) mit Messwertrauschen

Die Abbildung 3.13 zeigt die im Folgenden analysierten Werte der Distanz und der Winkelunterschiede aus der Tabelle 3.5 im Zeitverlauf während der Messung in der rotierten Ansicht. Die Abszisse stellt wieder den Zeitpunkt dar, zu dem eine Aktualisierung der DOPE-Pose stattfand. Visuell lässt sich auf einen Blick beobachten, dass sich die Bestimmung der Position und Orientierung der Objekte bei beiden Modellen relativ gleich akkurat verhält.

Achse	Durchschnittlicher Unterschied		Dim.
Frontal			
	FRDR	DRDR	
Distanz	0,0037553	0,0210037	Meter
X-Position	0,0012102	-0,0197676	Meter
Y-Position	0,0033898	0,0059112	Meter
Z-Position	-0,0017128	0,0046583	Meter
Winkel	7,09840432	13,2313823	Grad
X-Rotation	-5,6418492	-8,7305797	Grad
Y-Rotation	2,8447326	2,8944136	Grad
Z-Rotation	-0,1329218	0,0230344	Grad
Rotiert			
Distanz	0,0845925	0,0512626	Meter
X-Position	-0,0792385	-0,0458125	Meter
Y-Position	0,0267098	0,0219037	Meter
Z-Position	-0,0013951	-0,0075599	Meter
Winkel	8,2829061	7,5208745	Grad
X-Rotation	1,6556519	-1,7032967	Grad
Y-Rotation	0,6880761	0,8103289	Grad
Z-Rotation	4,8595507	4,9797696	Grad
Verdeckt			
Distanz	1,5545458	0,0472848	Meter
X-Position	-1,5360393	0,0313663	Meter
Y-Position	0,0748950	0,0328554	Meter
Z-Position	0,2270763	-0,0089442	Meter
Winkel	108,9090321	30,3947374	Grad
X-Rotation	11,3034449	4,0336234	Grad
Y-Rotation	0,5046642	0,5814941	Grad
Z-Rotation	22,3951273	-0,6520449	Grad

Tabelle 3.5: Unterschied zum ART System - FRDR vs. DRDR

Bei den nach der Rotation des Messobjektes gemessenen Werten ist zu sehen, dass sich die gemessene durchschnittliche Distanz zwischen dem ART-Objekt und der Position aus beiden Modellen im Vergleich zur Frontalansicht geringfügig verschlechtert hat. Die gemessene Distanz zwischen dem ART-Objekt und der Position aus dem DRDR-Modell, liegt bei ca. 0,05 m und die gemessene Distanz aus dem FRDR-Modell liegt bei 0,08 m. Somit ist die durch das DRDR-Modell um nur 0,03 m geringer.

Die Werte in Bezug zur Orientierung unterscheiden sich diesmal nicht nennenswert zwi-

schen den beiden Modellen und sind fast identisch. Der einzige merkliche Unterschied liegt in der x-Achse der Orientierung, welcher durch unterschiedliche Vorzeichen bedingt ist. Visuell sah die geschätzte Orientierung des FRDR-Modells im Verlauf der Messung genauer aus (siehe Abbildung 3.11).

Verdeckt

Abbildung 3.14: ”DR vs. FR”-Vergleich ”verdeckt”- Distanz (oben) - Winkel (unten) mit Messwertrauschen

Die Abbildung 3.14 zeigt die im Folgenden analysierten Werte der Distanz und der Winkelunterschiede im Zeitverlauf während der Messung in der verdeckten Ansicht. Die Abszisse stellt wieder den Zeitpunkt dar, zu dem eine Aktualisierung der DOPE-Pose stattfand. Auf den ersten Blick fällt auf, dass die Distanz beim FRDR-Modell mehrfach springt, obwohl der Verlauf ansonsten gleichmäßig ist.

Bei den Messungen in der Ansicht, in der das Messobjekt teilweise verdeckt wurde, ist zu sehen, dass die gemessene durchschnittliche Distanz zwischen dem ART-Objekt und der Position DRDR-Modell mit ca. 0,04 m deutlich geringer ist, als beim FRDR-Modell, wo der durchschnittliche Distanzwert bei 1,55 m liegt. In der Abbildung 3.14 liegt der durchschnittliche berechnete Distanz-Wert des FRDR-Modells etwas über 1,50 m mit drei Ausreißern nach unten. Während der Messung konnte in der Frontendanwendung beobachtet werden, dass die Ausreißer allerdings die Distanz-Werte sind, welche ein korrekteres Ergebnis liefern würden. Der Fehler, der hier zustande kommt ist, dass die mittels des DRFR-Modells geschätzte Entfernung der Kamera zum Messobjekt bei

nahezu 0 liegt. Dies erklärt auch das gemessenen stark erhöhte Messwertrauschen in der z-Achse der Position (Entfernung zur Kamera). Bei einem genaueren Blick auf Abbildung 3.11 ist in diesem Zusammenhang zu erkennen, dass die des FRDR-Modells geschätzte Position des Messobjektes unmittelbar vor der Kamera liegt. Das DRDR-Modell hingegen schätzt die Distanz wesentlich realistischer ein, was darauf hindeutet, dass das DRDR-Modell resistenter gegen die Verdeckung von Objekten ist.

Die Werte in Bezug zur Orientierung schwanken beim FRDR-Modell mit durchschnittlich 108.90° auch sehr stark. Dieses Verhalten konnte auch während der Messung in der Frontendanwendung beobachtet werden, weil das in der Frontend-Szene dargestellten Messobjektes sich ständig drehte. Die mittels des DRDR-Modells berechnete Orientierung ist mit einem durchschnittlichen Winkelversatz von 30.39° auch sehr ungenau, jedoch stabiler. Bei Betrachtung der Abbildung 3.11 fallen diesbezüglich zwei Spitzen beim Verlauf der Winkelabweichung auf. Diese Ausschläge belaufen sich bei genauerer Betrachtung jedoch nur um ca. 2.50°.

3.5.4 Evaluation

Der Leistungsvergleich der einzelnen Modelle zeigte, dass sich das Messwertrauschen der gemessenen Positions- als auch Orientierungswerte in der frontalen als auch in der rotierten Messansicht relativ stabil verhielt. In der verdeckten Messansicht des FRDR-Modells ließ sich jedoch ein starkes Messwertrauschen in der Position und Orientierung feststellen.

In der frontalen Messansicht war die berechnete Position aus dem DRDR-Modell geringfügig genauer als die des FRDR-Modells. Während die Orientierung im FRDR-Modell deutlich korrekter berechnet wurde.

In der rotierten Messansicht performten beide Modell wieder ähnlich zur frontalen Ansicht. Auch hier zeigte sich, dass die Berechnung der Orientierung beim FRDR-Modell deutlich korrekter war. Unter Berücksichtigung der Analyseergebnisse der frontalen und rotierten Messansicht bestätigt sich die Aussage von Jonathan Tremblay et. al [47], dass die Kombination von fotorealistischen als auch domänenrandomisierten Daten beim Training zur Überbrückung der Realitätslücke zu korrekteren Ergebnissen führt. Somit scheint es, dass die Integration von fotorealistischen Daten beim Training, durch realis-

tische Licht- und Schattenverhältnisse sowie Reflexionseffekte dazu führt, dass vor allem die Orientierung der zu erkennenden Objekte korrekter berechnet werden kann.

Bei der Analyse der Werte aus der verdeckten Ansicht zeigte sich jedoch, dass das DRDR-Modell gegenüber dem FRDR-Modell wesentlich besser abschnitt. In diesem Zusammenhang konnte das FRDR-Modell weder die Position noch die Orientierung des Messobjektes korrekt berechnen, was sich auch in dem gemessenen Messwertrauschen widerspiegelte. Das DRDR-Modell hingegen konnte zumindest die Position des Messobjektes relativ genau bestimmen, während die Orientierung zwar stabil, jedoch im Vergleich zur Realität sehr ungenau war. Somit bestätigt sich auch hier die Aussage von Jonathan Tremblay et. al [47], dass die Verwendung von domänenrandomisierten Daten die Modelle resistenter gegen die Verdeckung von Objekten macht. Dies ist darauf zurückzuführen, dass bei der Erzeugung der domänenrandomisierten Daten Distraktoren verwendet werden, die oftmals die freie Sicht auf die zu erkennenden Objekte verdecken.

Weiter konnten keine nennenswerten Unterschiede in der Aktualisierungsrate erkannt werden, sofern das Objekt erkannt wurde. In diesem Zusammenhang ist nur die verwendete Hardware ausschlaggebend für die Geschwindigkeit der Aktualisierungsrate bei der Schätzung der Pose von Objekten mittels Deep Objekt Pose Estimation.

3.6 Experiment - 5-DOF Objektklasse

In den Experimenten zuvor wurde die Cracker-Box als Messobjekt genutzt, deren Orientierung aus allen sechs Freiheitsgraden (6-DOF) feststellbar ist. In dieser Sektion wird das Verhalten eines trainierten Modells zur Schätzung der 6-DOF Pose einer etikettfreien Bierflasche untersucht. In diesem Zusammenhang ist interessant, dass die Bierflasche selbst für das menschliche Auge nur aus 5-DOF erkennbar ist. Meines Wissens ist dies das erste Experiment, das gezielt das Verhalten eines 5-DOF-Objektes bei der Erkennung mittels eines Deep Objekt Pose Estimation Netzerkes untersucht. Das Training des folgend untersuchten Modells erfolge über 60 Epochen und der verwendete Datensatz ist derselbe, der für das Experiment beim Vergleich "selbst erstellt vs. Nvidia" verwendet wurde. Somit ist die grundsätzliche Funktionsfähigkeit des Modells sichergestellt.

3.6.1 Aufbau

Dieses Experiment fand ausschließlich mithilfe einer unettikettierten Bierflasche als Messobjekt statt. In diesem Zusammenhang wurde das Verhalten der Objekterkennung ausschließlich in Bezug auf die 5-DOF Objektklasse geprüft. Der Aufbau gestaltet sich wie in den Experimenten zuvor. Diesbezüglich wurden die Messungen wieder aus drei Ansichten (frontal, rotiert und verdeckt) vorgenommen. Die gemittelten Daten zu allen drei Ansichten entstanden aus jeweils 100 mittels der DOPE erkannten Posen.

3.6.2 Analyse - Leistung des Modells

Damit bei der Analyse die Streuung der gemessenen Werte berücksichtigt wird, erfolgt nun eine Analyse in Bezug zum gemessenen Messwertrauschen, die der Tabelle 3.6 entnommen werden können.

Frontal

Bei Betrachtung der Tabelle 3.6 bewegt sich das Messwertrauschen der Positionen in der Frontalansicht im Millimeterbereich. Die größten Ausschläge der Position sind, wie in den Experimenten zuvor, in der z-Achse vorhanden. Das Messwertrauschen der Position liegt in der x-Achse bei ca. -0,10 Millimeter, in der y-Achse bei ca. -0,11 Millimeter

Achse	Durchschnittliches Messwertrauschen	Dim.
Frontal		
X-Position	-0,0001041	Meter
Y-Position	-0,0001116	Meter
Z-Position	0,0008167	Meter
X-Rotation	0,0123832	Grad
Y-Rotation	-0,0370052	Grad
Z-Rotation	0,0990610	Grad
Rotiert		
X-Position	0,0000939	Meter
Y-Position	-0,0003761	Meter
Z-Position	-0,0050623	Meter
X-Rotation	0,0089889	Grad
Y-Rotation	0,0095270	Grad
Z-Rotation	-0,0659527	Grad
Verdeckt		
X-Position	-0,0001181	Meter
Y-Position	0,0001930	Meter
Z-Position	0,0009974	Meter
X-Rotation	0,0079055	Grad
Y-Rotation	0,0120341	Grad
Z-Rotation	-0,0231878	Grad

Tabelle 3.6: Leistungsübersicht (Messwertrauschen) - 5-DOF Messobjekt

und in der z-Achse bei ca. 0,82 Millimeter. Alle drei Achsen zeigen somit einen höheres Messwertrauschen als das ART-System.

Auch bei der Betrachtung der Werte in Bezug zur Orientierung lässt sich ein geringeres Messwertrauschen in allen Achsen beobachten. In diesem Zusammenhang liegt das Messwertrauschen in der x-Achse bei ca. 0.01°, in der y-Achse bei ca. -0.03° und in der z-Achse bei ca. 0.1°.

Die Aktualisierungsrate während der Erkennung variierte zwischen 99 bis 197 Millisekunden, durchschnittlich lag sie bei 152 Millisekunden.

Rotiert

Nach der Rotation des Messobjektes konnte gemessenen werden, dass das Messwertrauschen der Position in der x-Achse, als auch in der y-Achse ähnlich wie in der frontalen

Ansicht ausfällt. So liegt das Messwertrauschen in der x-Achse bei ca. 0,09 Millimeter und in der y-Achse bei ca. 0,37 Millimeter. In der z-Achse liegt es jedoch bei ca. -5,06 Millimeter. Damit fällt das Messwertrauschen in der rotierten Ansicht um ca. -4,24 Millimetern stärker aus. Interessant ist jedoch, dass in der rotierten Ansicht der Tisch so gedreht wurde, dass sich das Modell an dieser Stelle wie in der frontalen Ansicht verhalten sollte. Der Unterschied beim Messwertrauschen kann an dieser Stelle an vielen Umständen aus der Umgebung, wie z.B. die Reflexion des Lichtes auf der Bierflaschenoberfläche, liegen.

Bei der Betrachtung der Werte in Bezug zur Orientierung wurde ein geringeres Messwertrauschen gemessen. In diesem Zusammenhang liegen die Werte in der x-Achse bei ca. 0.009°, in der y-Achse bei ca. -0.01° und in der z-Achse bei ca. -0.07°. Auch hier sind die Unterschiede der Werte auf die unbeherrschbaren Messbedingungen zurückzuführen.

Die Aktualisierungsrate variierte während der Erkennung zwischen 95 und 233 Millisekunden, wobei der Durchschnitt bei ca. 151 Millisekunden lag. Der erhöhte Maximalwert erklärt sich dadurch, dass das Objekt in einigen Zeitintervallen während der Messung für längere Zeit nicht erkannt wurde, ohne dass sich die beherrschbaren Messungsbedingungen geändert haben.

Nicht beherrschbare Änderung, wie die Beleuchtung durch die Fenster in dem Raum in dem die Messungen vorgenommen wurden, können jedoch einen Einfluss auf die Erkennung des Objektes haben. Bei Beleuchtung des Messobjektes mit einer Taschenlampe war beispielsweise zu sehen, dass die geschätzten Werte des trainierten Modells stark variierten.

Verdeckt

In der verdeckten Ansicht fallen die Messungen des Messwertrauschens in allen Achsen der gemessenen Position sehr ähnlich zu den Werten der frontalen Ansicht aus.

Auch die Werte bezüglich der Orientierung fallen relativ stabil aus, sodass hierbei keine nennenswerten Unterschiede zu den anderen Ansichten auffallen.

Die Aktualisierungsrate während der Erkennung variierte in beiden Modellen zwischen 100 bis 226 Millisekunden, durchschnittlich lag sie bei 152 Millisekunden. Der hohe Maximalwert zeigt, dass das Objekt im Messungszeitraum teilweise nicht erkannt wurde.

Dieses Verhalten trat jedoch nur selten auf, da der Durchschnittswert weiterhin bei ca. 152 Millisekunden lag.

Weiter ist während der Messung aufgefallen, dass das Objekt bei minimaler Verdeckung nur noch selten erkannt wurde. Sobald das Messobjekt jedoch in einer Position erkannt wurde, lief die Erkennung relativ stabil, was die Aktualisierungsrate zeigt.

3.6.3 Analyse - Vergleich zum ART-System

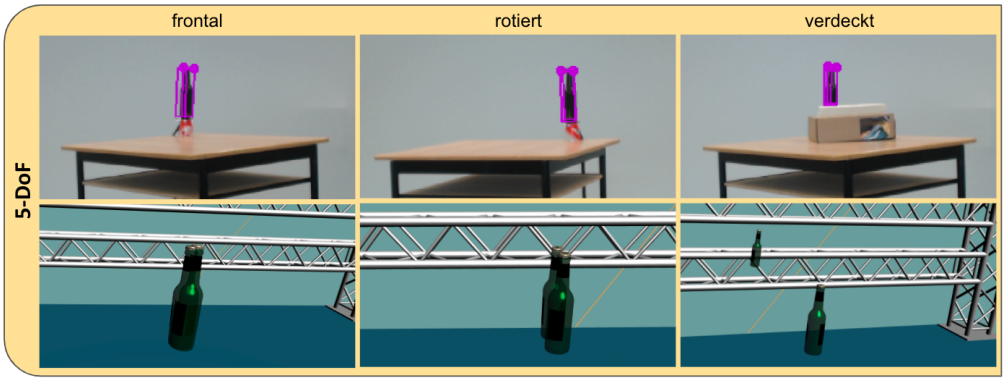

Abbildung 3.15: ”5-DOF”- Visuelle Darstellung der Unterschiede

In diesem Abschnitt wird das Modell zur Erkennung einer unetikettierten Bierflasche mit dem ART-System verglichen. Zur Vereinfachung der Interpretation der im Folgenden analysierten Werte, stellt Abbildung 3.15 die Unterschiede zwischen den gemessenen Werten visuell dar. Im Unterschied zu den vorherigen Analysen zeigt die Abbildung nur eine farblich markierte Reihe mit Screenshots aus dem ROS-Framework mit einem 3D-Rahmen, der die geschätzten Koordinaten darstellt (oben) und Screenshots aus der Frontendanwendung (unten).

Frontal

Die Abbildung 3.16 zeigt einen Ausschnitt des Verlaufs der gemessenen Distantwerte und Winkelunterschiede während der Messung in der frontalen Ansicht. Die Abszisse stellt den Zeitpunkt dar, zu dem eine Aktualisierung der DOPE-Pose stattfand. Die durchschnittlichen Werte sind der Tabelle 3.7 zu entnehmen.

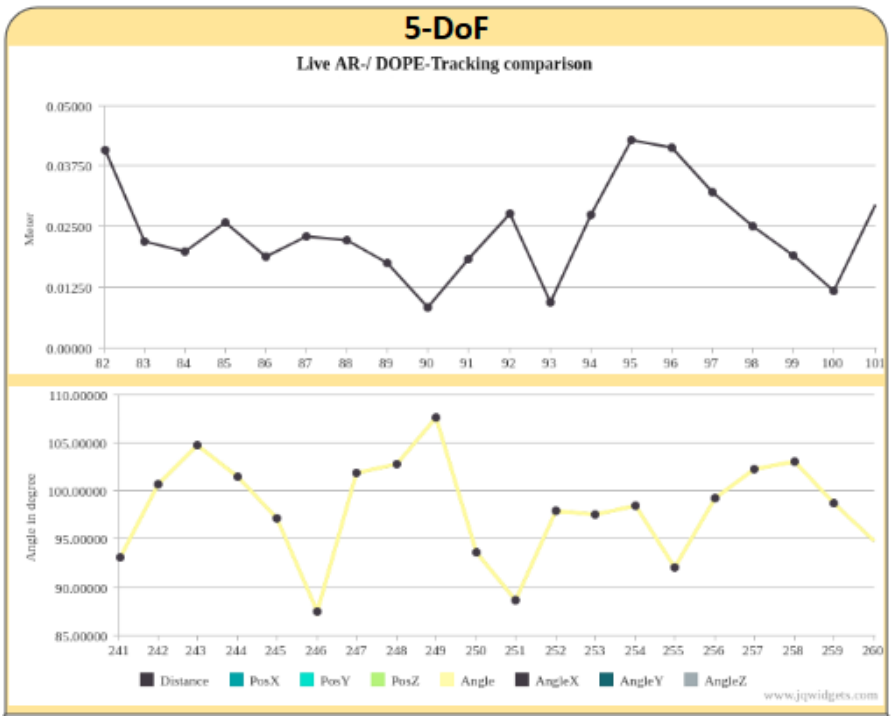

Abbildung 3.16: 5-DOF-Vergleich ”frontal”- Distanz (oben) - Winkel (unten) mit Messwertrauschen

Hierbei ist zu erkennen, wie genau die Position als auch die Orientierung mithilfe des Modells berechnet wurde. Desto geringer die Zahlen in der Ordinate, desto genauer ist die Erkennung. Bei Betrachtung der Tabelle 3.7 ist zu sehen, dass die gemessene durchschnittliche Distanz zwischen dem ART-Objekt und der Position aus dem Modell sehr gering ist. Die gemessene Distanz liegt bei ca. 0,014 m.

Bei der Betrachtung der Werte in Bezug zur Orientierung liegt der Winkelversatz bei ca. 100.47° im Vergleich zum ART-System. Somit sieht es so aus, als würde die Orientierung fehlerhaft berechnet werden. Bei Betrachtung der Abbildung 3.15 scheint sie jedoch korrekt zu sein. In der Tabelle 3.7 sehen wir den größten Unterschied in der z-Achse, welche genau der Achse entspricht, die bei einer Rotation auch für das menschliche Auge nur schwer unterscheidbar ist. In diesem Zusammenhang kann also festgehalten werden, dass die Berechnung der Orientierung unter Vernachlässigung der z-Achse weiterhin ziemlich genau ist.

Achse	Durchschnittlicher Unterschied	Dim.
Frontal		
Distanz	0,0235151	Meter
X-Position	-0,0227543	Meter
Y-Position	0,0038720	Meter
Z-Position	0,0071537	Meter
Winkel	100,4655197	Grad
X-Rotation	11,5840612	Grad
Y-Rotation	8,3867283	Grad
Z-Rotation	-90,7356492	Grad
Rotiert		
Distanz	0,0833102	Meter
X-Position	0,0792725	Meter
Y-Position	-0,0062783	Meter
Z-Position	-0,0052712	Meter
Winkel	68,5573552	Grad
X-Rotation	52,2205565	Grad
Y-Rotation	3,7392048	Grad
Z-Rotation	-33,6173475	Grad
Verdeckt		
Distanz	0,8262900	Meter
X-Position	0,8264772	Meter
Y-Position	0,0546098	Meter
Z-Position	0,0020994	Meter
Winkel	94,9120330	Grad
X-Rotation	4,4342999	Grad
Y-Rotation	6,2917940	Grad
Z-Rotation	-82,0142986	Grad

Tabelle 3.7: Unterschied zum ART System - 5-DOF Messobjekt

Rotiert

Die Abbildung 3.18 zeigt die im Folgenden analysierten Werte der Distanz und der Winkelunterschiede aus der rotierten Ansicht. Die Abszisse stellt wieder den Zeitpunkt dar, zu dem eine Aktualisierung der DOPE-Pose stattfand.

Wie bereits in der frontalen Ansicht sind die Unterschiede in den Werten bezüglich der Position im Vergleich zur frontalen Ansicht mit einem Unterschied von ca. 0,06 m nur gering.

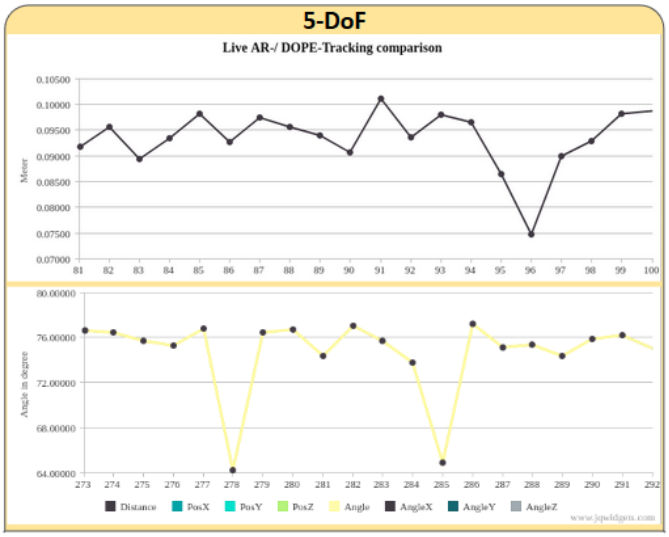

Abbildung 3.17: 5-DOF-Vergleich ”rotiert”- Distanz (oben) - Winkel (unten) mit Messwertrauschen

Die Euler-Werte der Orientierung deuten wie bereits in der frontalen Ansicht auf einen Anzeigefehler hin. Gleichzeitig sieht die visuelle Darstellung in Abbildung 3.15 abermals vernünftig aus. Hiermit zeigt sich erneut, dass die Berechnung der Orientierung, unter Vernachlässigung der z-Achse, ziemlich genau ist.

Verdeckt

In Abbildung 3.18 fällt auf, dass die Distanz zwischen der geschätzten Objektposition und der Position des ART-Systems mit durchschnittlich ca. 0,83 m relativ groß ist. Bei einem Blick auf Abbildung 3.15 ist gut zu erkennen, dass die beiden visuell dargestellten Bierflaschen in einer Linie relativ weit voneinander entfernt liegen. Auf dem Screenshot aus dem ROS-Framework ist der Grund dafür zu erkennen, der darin liegt, dass das Objekt mit einem weißen Block verdeckt wurde. Dieser weiße Block wurde so platziert, dass der untere Bereich der stehenden Bierflasche teilweise abgeschnitten ist. Diesbezüglich scheint es so als würde das Modell die Bierflasche als ”kleiner” wahrnehmen, was den Sachverhalt erklären würde, dass die geschätzte Position der Bierflasche in weiterer Entfernung liegt.

Die gemessenen Unterschiede in der Orientierung verhalten sich wie bereits in der frontalen und rotierten Ansicht und sind aus diesem Grund nicht weiter nennenswert.

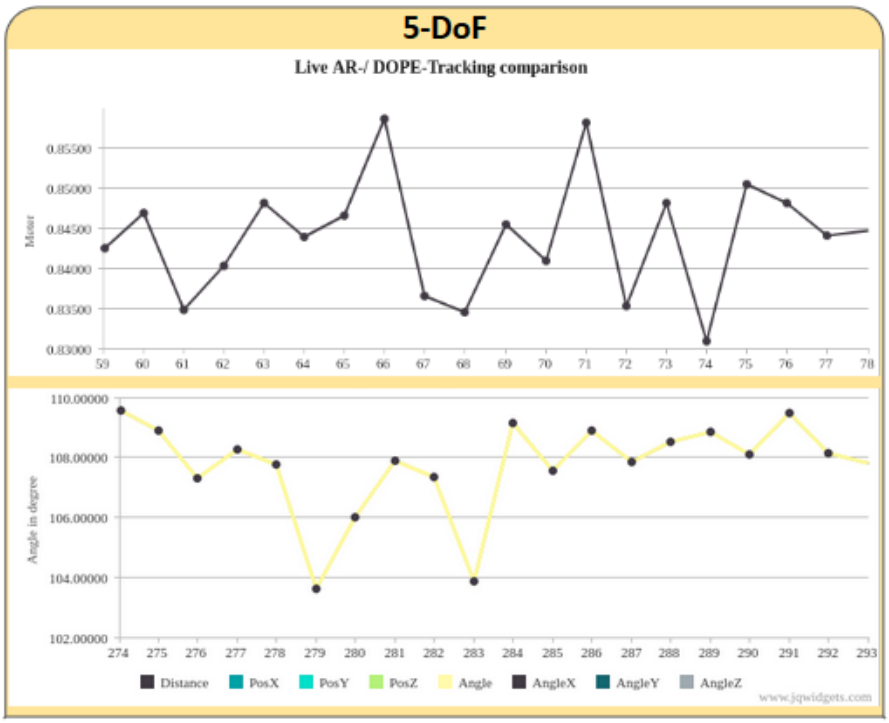

Abbildung 3.18: 5-DOF-Vergleich ”verdeckt”- Distanz (oben) - Winkel (unten) mit Messwertrauschen

3.6.4 Evaluation

Die Leistung des Modells zur Posenschätzung eines 5-DOF Objektes zeigte bei Betrachtung des Messwertrauschens keine Besonderheiten auf. Alle Werte lagen dabei in einem stabilen Rahmen ohne nennenswerte Ausreißer in der Positionsbestimmung sowie der Orientierung.

In der frontalen sowie in der rotierten Messansicht wichen die berechneten Positionen des Modells in einem akzeptablen Rahmen von der Position des ART-Systems ab. Auffällig war hier die geschätzte Orientierung des Messobjektes, welche starke Unterschiede in der z-Achse des Objektes aufzeigte. Dies war bei der Analyse zu erwarten, da die Bierflasche als 5-DOF Messobjekt aus einem Freiheitsgrad nicht erkennbar ist, welcher im Messaufbau der z-Achse entsprach. Bei einem Blick auf die Abbildung 3.15 war zu erkennen, dass die geschätzte Orientierung relativ korrekt ist. Weiter ließ sich während der Messungen in der frontalen und rotierten Messansicht beobachten, dass sich nicht beherrschbare Messbedingungen stark auf die Schätzung der Pose des Messobjektes aus-

wirken. Dieses Verhalten wurde empirisch beobachtet, indem das Messobjekt mit einer Taschenlampe beleuchtet wurde.

In der verdeckten Messansicht ließ sich beobachten, dass bereits eine minimale Verdeckung der Bierflasche dazu führt, dass das Objekt teilweise gar nicht erkannt wird. Somit besteht bei dem 5-DOF Messobjekt keine starke Resistenz gegenüber der Verdeckung. Gleichzeitig war zu sehen, dass bei horizontaler Verdeckung das Objekt als weiter entfernt geschätzt wird.

3.7 Experiment - 3-DOF Objektklasse

Nach der 5-DOF Objektklasse folgt nun eine Untersuchung bezüglich eines Objektes aus der 3-DOF Objektklasse. Hierbei ist interessant, welche Leistung das Modell zeigt, wenn bei der Schätzung die drei Freiheitsgrade der Orientierung nicht verwendet werden können. Somit bleibt dem Modell für die Erkennung der Position und Orientierung des Messobjektes alleine die Größe. Meines Wissens ist dies das erste Experiment, das gezielt das Verhalten eines 3-DOF-Objektes bei der Erkennung mittels eines Deep Objekt Pose Estimation Netzerkes untersucht. Das Training des folgend untersuchten Modells erfolge über 15 Epochen und der verwendete Datensatz ist derselbe, der für das Experiment beim Vergleich ”selbst erstellt vs. Nvidia” verwendet wurde. Somit ist die grundsätzliche Funktionsfähigkeit des Modells sichergestellt.

3.7.1 Aufbau

Dieses Experiment fand ausschließlich mithilfe einer Billard-Kugel als Messobjekt statt. Damit wurde das Verhalten der Objekterkennung ausschließlich in Bezug auf die 3-DOF Objektklasse geprüft. Im Unterschied zu den vorherigen Experimenten besteht dieser Aufbau aus nur einer Ansicht,da die Erkennung des 3-DOF Messobjektes nur in sehr seltenen Fällen funktionierte.

3.7.2 Analyse - Leistung des Modells

Die Tabelle 3.8 zeigt das gemessenen Messwertrauschen aus der allgemeinen Ansicht. Stark auffällig ist hierbei ein sehr hoher Wert in der z-Achse, der bei ca. 7,97 m liegt.

Achse	Durchschnittliches Messwertrauschen	Dim.
Allgemein		
X-Position	-0,0438203	Meter
Y-Position	-0,0049485	Meter
Z-Position	7,9737860	Meter
X-Rotation	27,5760327	Grad
Y-Rotation	31,1673321	Grad
Z-Rotation	19,1240427	Grad

Tabelle 3.8: Leistungsübersicht (Messwertrauschen) - 3-DOF Messobjekt

Dieser Wert zeigt, dass permanent starke Änderungen in der geschätzten Entfernung des Messobjektes zur Kamera gemessen wurden.

Auch die Werte in Bezug zur Orientierung schwanken sehr stark, sodass das Modell bei der Schätzung der still liegenden Billard-Kugel eine permanente Rollbewegung wahrgenommen hat.

Die Aktualisierungsrate während der Erkennung variierte zwischen 113 bis 420 Millisekunden, durchschnittlich lag sie bei 153 Millisekunden. Der hohe Maximalwert zeigt, dass das Objekt über ein längeres Zeitintervall teilweise gar nicht erkannt wurde. Dies ließ sich auch durch empirische Beobachtung bestätigen, wodurch sich der Messaufbau sehr schwierig gestaltete.

3.7.3 Analyse - Vergleich zum ART-System

Es folgt der Vergleich des Modells zur Erkennung einer Billard-Kugel gegenüber dem ART-System. Auf Abbildung 3.19 ist die allgemeine Ansicht aus einem ROS-Screenshot (oben) und einem Screenshot aus der Frontendanwendung (unten) zu sehen. Außerdem zeigt ein grüner Pfeil auf eine kleine rote Kugel, um auf die geschätzte Pose des Modells hinzuweisen. Weiter ist auf dem Screenshot aus dem ROS-Framework zu sehen, dass die Billardkugel vor einer schwarzen Box platziert wurde. Dies hatte den Grund, damit sich die Billardkugel optisch mehr von dem Hintergrund abhebt. Ohne die Box konnte keine erfolgreiche Erkennung der Billardkugel gemessen werden.

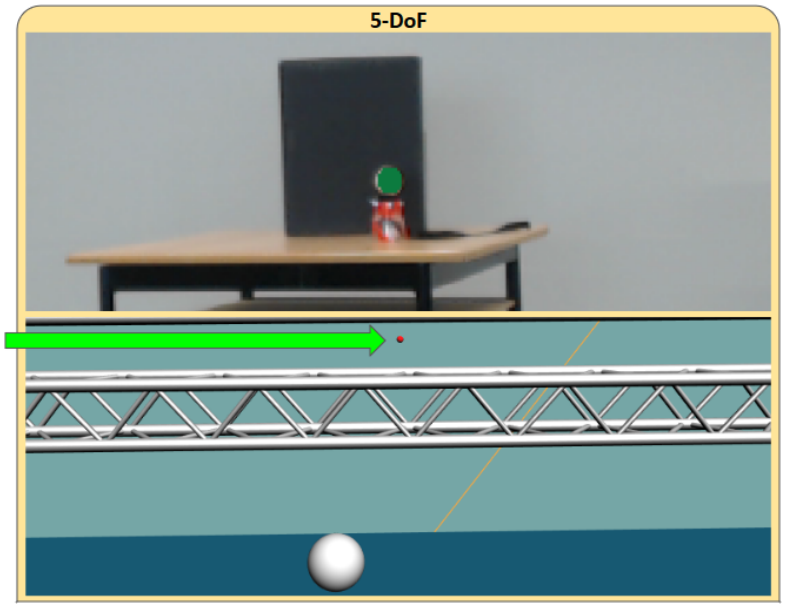

Abbildung 3.19: ”3-DOF”- Visuelle Darstellung der Unterschiede

Achse	Durchschnittlicher Unterschied	Dim.
Frontal		
Distanz	11,0538433	Meter
X-Position	8,8952446	Meter
Y-Position	-0,4426461	Meter
Z-Position	3,5131574	Meter
Winkel	147,1457194	Grad
X-Rotation	-0,0300473	Grad
Y-Rotation	25,3517230	Grad
Z-Rotation	66,8068457	Grad

Tabelle 3.9: Unterschied zum ART System - 3-DOF Messobjekt

Allgemein

Bei der Betrachtung der Messwerte in Bezug zur Objektposition fällt der sehr hohe Distanz-Wert von ca. 11,05 m auf. Abbildung 3.19 stellt diesen Distanzunterschied sehr gut dar. Weiter zeigt Abbildung 3.20 die große Varianz der Distanz im Zeitverlauf.

Die Messergebnisse der Orientierung zeigen stark unterschiedliche Werte im Vergleich zum ART-System. Hier liegt der gemessene durchschnittliche Winkelversatz bei ca. 147.15°. Unter Berücksichtigung des Messwertrauschens steckt in diesem Durchschnittswert je-

doch keine Aussagekraft, da eine permanente Änderung der Orientierungswerte gemessen wurde.

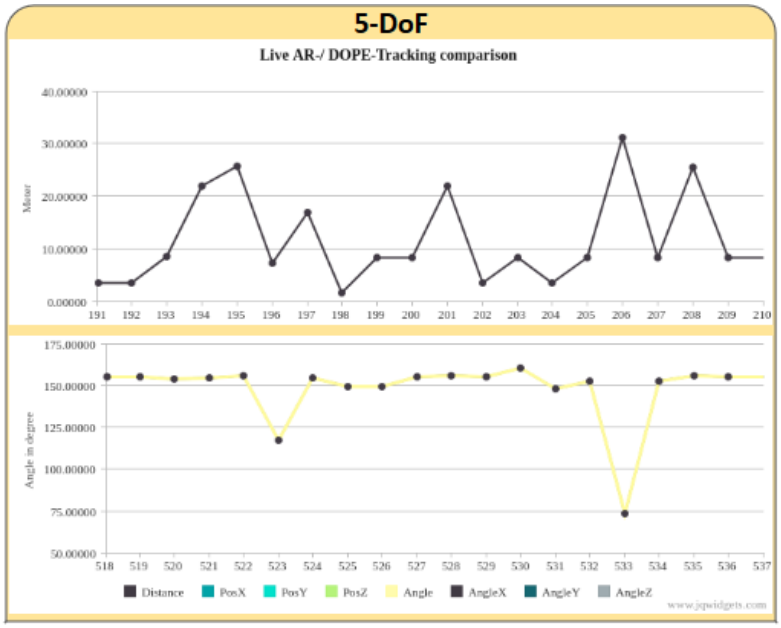

Abbildung 3.20: 3-DOF-Vergleich - Distanz (oben) - Winkel (unten) mit Messwertrauschen

3.7.4 Evaluation

Die Verwendung von Deep Objekt Pose Estimation zur Erkennung von Objekten aus der 3-DOF Objektklasse ist für die Praxis allgemein ungeeignet.

Die Erkennung des Objektes gelang nur selten und zufällig. Aus diesem Grund wurde die Erkennung mit Modellen aus unterschiedlichen Epochen getestet (10, 15 bis 20, 30, 45 und 60). Auffällig war dabei, dass eine Erkennung nur bei Modellen erfolgte, die mit vergleichsweise wenig Epochen (15 bis 20) trainiert wurden. In allen anderen Epochen konnte keine Erkennung gemessen werden.

Die Messung in Bezug zur Pose der Billardkugel zeigen eine starke Varianz in allen gemessenen Achsen. Interessant ist hierbei besonders die gemessene Distanz zwischen Kamera und Messobjekt: Mit mehr als 11 m liegt der Wert weit außerhalb der Distanzintervalle innerhalb der Aufnahmestellen, die bei der Erstellung des Datensatzes zum Training des Modells aufgestellt wurden (siehe [42]).

4 Ergebnisse und Diskussion

In diesem Kapitel Erfolg eine fachliche Diskussion und Einordnung der Anwendbarkeit des Deep Objekt Pose Estimation Netzwerkes auf Basis der im Kapitel 'Methodik und Forschung' ausgewerteten und interpretierten Ergebnisse.

4.1 Allgemein

Die Deep Learning basierte Erkennung von 3D-Objektposen mithilfe eines Deep Object Pose Estimation Netzes auf Basis synthetisch erzeugter Daten hat sich als funktionsfähig erwiesen. Die trainierten Modelle zeigten, dass die Verwendung von synthetisch erzeugten Daten eine Transferleistung von Virtualität zur Realität für künstliche Intelligenzen ermöglichen. Zu Berücksichtigen ist jedoch, dass bei der Erstellung von synthetischen fotorealistischen Bildern ein größerer einmaliger Aufwand durch den notwendigen Aufbau von fotorealistischen 3D-Szenen entsteht, wie er in [42] beschrieben wurde. Bei dem Aufbau der fotorealistischen 3D-Szenen sowie der domänenrandomisierten Szene wurde die Entfernung der Kamera zu den zu erkennenden Objekten auf 0,5 bis 4 Meter eingerichtet. Bei den Messungen konnte beobachtet werden, dass in diesem Bereich die Objekte besonders gut erkannt wurden. Darüber hinaus wurden die Objekte auch erkannt, gleichzeitig ließ sich mit zunehmendem Abstand ein schlechteres Verhalten bei der Schätzung der Pose beobachten.

Das Training der Modelle für 60 Epochen benötigte ca. 11 Tage auf der von Matthias Nitsche und Stephan Halbritter beschriebenen Renderfarm der HAW Hamburg [28] und ein Modell kann bei DOPE nur zur Erkennung eines Objektes trainiert werden. Dieser hohe Zeitaufwand sollte bei der Verwendung je nach Anwendungsfall berücksichtigt werden.

Das Objekt selber wird im Videolivestream in einem Abstand zwischen 100 und 200 Millisekunden erkannt. Der Durchschnitt mit der verwendeten Hardware lag bei 150 Millisekunden. Damit sind Modelle, die mittels DOPE trainiert wurden aktuell nicht

echtzeitfähig. Gleichzeitig besteht hierbei die Möglichkeit der Optimierung mithilfe besserer Hardware, wobei die für die Experimente verwendete Hardware bereits der Leistung eines aktuellen High-End Rechners entspricht.

6-DOF Objektklasse

Das Modell funktionierte zuverlässig bei der Bestimmung der Pose von Objekten der 6-DOF Objektklasse sofern diese gut aus allen sechs Freiheitsgraden zu Erkennen sind. In diesem Zusammenhang lässt sich die Pose des Objektes in einer frontalen, rotierten und teilweise verdeckten Ansicht gut schätzen. Verformungen des Objektes wirken sich direkt negativ auf die Erkennung aus. Gleichzeitig gelang es dem Modell das Objekt bei Verformung zu erkennen, jedoch konnte die Objektpose dabei nicht optimal bestimmt werden. Somit ist eine gute Erkennungsleistung des DOPE-Netzes nur bei Objekten der 6-DOF Objektklasse gewährleistet, deren Körper nicht veränderbar ist.

5-DOF Objektklasse

Auch bei Objekten der 5-DOF Objektklasse leistet das DOPE-Netz eine gute Leistung, sodass beispielsweise eine Bierflasche gut erkannt wurde. Bei der Posenschätzung zeigt sich ein starkes Messwertrauschen in der Rotationsachse, welche den fehlenden sechsten Freiheitsgrad beschreibt. Dieses Verhalten ist jedoch unproblematisch bei der Schätzung der Objektpose. Verdeckungen des Objektes aus der 5-DOF Objektklasse wirken sich stark auf die geschätzte Pose des Objektes aus, sodass sich eine geringere Resistenz gegenüber der Verdeckung des Objektes beobachtet werden konnte. Dieses Verhalten schränkt die Anwendbarkeit von DOPE bei 5-DOF Objekten stark ein, was je nach Anwendungsfall zu berücksichtigen ist.

3-DOF Objektklasse

Für die Erkennung von Objekten der 3-DOF Objektklasse hat sich die Deep Learning basierte Erkennung von 3D-Objektposen als ungeeignet herausgestellt. Ein Objekt, dessen Pose nur anhand der Größe bestimmbar ist, bietet für das Training einer künstlichen Intelligenz zu wenig Anhaltspunkte, anhand derer das Objekt erkannt werden kann. So war es im Laufe der Experimente nur sehr selten der Fall, dass die für das Experiment

verwendete Billardkugel erkannt wurde. Und selbst wenn die Billardkugel erkannt wurde, wich die geschätzte Pose extrem von der Realität ab.

4.2 Anwendbarkeit

Die Schätzung der Pose mithilfe von Deep Object Pose Estimation (DOPE) zeigte insgesamt eine gute Leistung. In diesem Zusammenhang ist eine Anwendbarkeit innerhalb von beispielsweise Augmented Reality Anwendungen zur Annotation von nahe gelegenen Objekten durchaus denkbar und bietet mit der richtigen Konfiguration eine ausreichend hohe Genauigkeit, um virtuelle und reale Objekte im selben Raum existieren zu lassen.

Allerdings gilt es je nach Anwendung die Art der Objektklasse zu berücksichtigen, da Objekte der 5-DOF-Objeklasse nur schwer erkannt werden, wenn diese verdeckt werden. Die 3-DOF-Objektklasse ist für jegliche Anwendungen, in denen DOPE verwendet wird, ungeeignet.

Eine Anwendbarkeit der DOPE in der Robotik zur Roboterarmsteuerung und dem damit verbundenen Greifen von Objekten ist auch valide. Durch die Einschränkung der hohen Anfälligkeit gegenüber der Verdeckung von Objekten aus der 5-DOF-Objektklasse müssten je nach Anwendungsfall gegenebenfalls Abhilfen gefunden werden. In diesem Zusammenhang könnten die zu erkennenden Objekte beispielsweise auf ein Rüttelband gelegt werden, damit die Objekte sich durch das Rütteln des Bandes voneinander befreien, wodurch eine freie Sicht auf die Objekte entsteht.

In allen Anwendungsfällen gilt außerdem die hohe benötigte Rechenleistung sowie die nicht echtzeitfähige Aktualisierungsrate bei der Erkennung zu berücksichtigen. Außerdem ist die fehlende Skalierbarkeit im Sinne des Hinzufügens neuer zu erkennender Objekte ein Punkt, der die Anwendbarkeit je nach Anwendungsfall einschränkt. So können neue zu erkennende Objekte nur mit viel Aufwand erkannt werden. Die dafür benötigte Zeit auf aktueller Hardware beläuft sich für ein 60 Epochen trainiertes Modell auf ungefähr 10 bis 14 Tage. Diese Zeitspanne kommt dadurch zustande, dass das zu erkennende Objekt zunächst in die Szene integriert (ca. 1 Tag), der Datensatz erstellt (ca. 2 Tage) und das Modell trainiert (ca. 11 Tage) werden muss.

4.3 Fotorealismus beim Datensatz

Eine der Kernfragen, welche erstmalig von Jonathan Tremblay et. al [47] untersucht wurde, ist die Frage nach dem Aufbau des Datensatzes zum Training eines DOPE-Netzwerks. Für diese Untersuchung erstellten Jonathan Tremblay et. al Datensätze, die rein aus realistischen Bildern, rein aus domänenrandomisierten Bildern und einen Datensatz, der zu gleichen Teilen mit fotorealistischen als auch aus domänenrandomisierten Bildern bestehen. Weil diese Untersuchungen die Ersten in diesem Zusammenhang sind, wurde zur Validierung ein eigenes Experiment für diese Arbeit durchgeführt.

Die eigenen Untersuchungen zeigten hierbei, dass ein Datensatz der alleine auf domänenrandomisierten Daten trainiert wurde, bereits gute Ergebnisse liefert. Die Schätzung der Position des Objektes war erfolgreich und stabil. Besonders ist aufgefallen, dass das Modell sehr resistent gegenüber der Verdeckung des Messobjektes war. Das Modell, welches zu gleichen Teilen aus fotorealistischen und domänenrandomisierten Daten trainiert wurde, zeigte sich auch resistent gegenüber der Verdeckung des Messobjektes, gleichzeitig zeigte sich das Modell weniger resistent gegen Verdeckungen. Dieses Ergebnis war jedoch zu erwarten, da der domänenrandomisierte Datensatz Distraktoren beinhaltet, welche die zu erkennenden Objekte teilweise verdecken.

Die Verwendung von fotorealistischen Daten zielt in diesem Zusammenhang jedoch auf etwas anderes ab: Mithilfe der fotorealistischen Daten soll die Transferleistung zwischen Realität und Virtualität für eine künstliche Intelligenz vereinfacht werden (siehe Reality Gap [46]). In diesem Zusammenhang soll das trainierte Modell resistent gegenüber unterschiedlichen Licht- und Schattenverhältnissen sowie Reflexionen werden. Diesbezüglich zeigte sich durch die Experimente, dass die Schätzung der Orientierung der Messobjekte mithilfe des gemischten Datensatzes besser mit der Realität übereinstimmte. Somit bestätigt sich die von Jonathan Tremblay et. al [46] gemessene Steigerung der Qualität bei der Posenerkennung durch die Mischung von fotorealistischen und domänenrandomisierten Bildern.

4.4 Optimierungsmöglichkeiten

Im Rahmen der Experimente ließ sich das Verhalten von DOPE auf verschiedene Umstände gut beobachten. Diese Beobachtungen beinhalten einen umfangreichen Interpre-

tationsspielraum für die Möglichkeiten der Optimierung der Posenbestimmung mittels DOPE, welche im Folgenden vorgestellt werden.

4.4.1 Rahmenbedingungen für Anwendungsfälle

Eine Möglichkeit, die Erkennung zu verbessern, besteht darin, die Rahmenbedingungen bei der Erstellung der Szenen zur Datensatzerstellung auf einen entsprechenden Anwendungsfall zuzuschneiden. Sollen beispielsweise Objekte innerhalb einer Lagerhalle erkannt werden, besteht hierbei die Möglichkeit, den fotorealistischen Teil des Datensatzes innerhalb einer realistisch wirkenden Lagerhallenumgebung zu generieren. Hierbei könnten die Licht- Schatten- und Reflexionseffekte dahingehend angepasst werden, dass das Modell beim Training besonders robust für diese Art von Umgebung wird.

Weiter kann die Entfernung der Kamera zu den zu schätzenden Objekten auf den Bereich beschränkt werden, der zum größten Teil in dem jeweiligen Anwendungsfall zu erwarten ist. Ist in einem Anwendungsfall beispielsweise zu erwarten, dass Objekte minimal 0,5 bis maximal einen Meter von der Kamera entfernt sind, dann können diese Entfernungen innerhalb des Datensatzes gezielt generiert werden. Dies hätte zur Folge, dass das Entfernungsintervall besonders präzise vom Modell geschätzt werden würde.

Zuletzt ist eine Anpassung der Rahmenbedingungen an große und weit entfernte Objekte möglich, um beispielsweise weit entfernte Drohnen, Flugzeuge, Autos oder auch Gebäude zu erkennen. Hierbei sind jedoch tiefgreifende Arbeiten in der Gestaltung der Szene notwendig, die sich auf die Größe des zu erkennenden Objektes beziehen. Diesbezüglich setzt sich Thomas Kanne-Schludde [19] mit der Erkennung des Notre Dame und der daraus abgeleiteten Erzeugung von GPS-Koordinaten auseinander.

4.4.2 Einsatz von Distraktoren

Der Versuchsaufbau 'domänenrandomisiert & fotorealistisch' zeigte, dass das Modell, welches ausschließlich mit domänenrandomisierten Daten trainiert wurde, robuster im Hinblick auf die Verdeckung der Objekte war. Dies hängt vor allem damit zusammen, dass bei der Erstellung des Datensatzes aus der domänenrandomisierten Szene Distraktoren verwendet werden, welche die freie Sicht auf die Objekte teilweise einschränken.

Damit die Robustheit gegenüber der Verdeckung bereits in den fotorealistischen Szenen verstärkt wird, können Distraktoren auch bei der Generierung der fotorealistischen Daten berücksichtigt werden. Dies hätte weiter den Vorteil, dass die Distraktoren einen Einfluss auf die Licht-, Schatten- und Reflexionseffekte bei den fotorealistisch generierten Daten hätte. Gleichzeitig sollte bei dem Einsatz von Distraktoren darauf geachtet werden, dass die Anzahl der Distraktoren überschaubar bleibt, sodass der generierte Datensatz genügend Daten enthält, in denen die zu schätzenden Objektposen gut erkennbar sind.

4.4.3 Low-Poly versus High-Poly

Die Erzeugung des für diese Arbeit verwendeten synthetischen FRDR-Datensatz benötigte auf einem Rechner mit einer Geforce RTX 2080, einem Ryzen 7 1700 und 16 GB Arbeitsspeicher ungefähr 10 Stunden. Diese Zeit kommt dadurch zustande, weil beim Erstellungsprozess die 3D-Objekte zufällig in einem definierten Bereich innerhalb der Szene erscheinen. Dabei werden jedes Mal die einzelnen Polygone der 3D-Modelle gerendert. Da für 2D-Bilder jedoch keine High-Poly-Modelle verwendet werden müssen, lässt sich der Erstellungsprozess verkürzen, indem nur Modelle erzeugt werden, deren Polygonanteil gering ist. Dabei sollte jedoch darauf geachtet werden, dass der Polygonanteil hoch genug ist, damit das 3D-Modell das reale Objekt repräsentiert.

Gleichzeitig können High-Poly-Modelle mehr Information beinhalten, die das Modell beim Training als Features erkennt. Diese Features können weiter zur genaueren Bestimmung der Objektpose dienen.

4.4.4 Prozedurale Erzeugung fotorealistischer domänenrandomisierter Umgebungen

Bevor ein Datensatz mithilfe des NDDS-Plugins generiert werden kann, müssen unter anderem innerhalb der Unreal Engine mehrere fotorealistische Szenen erstellt werden. Dieser Arbeitsschritt kann ziemlich aufwendig sein und benötigt in der Praxis mehrere Tage bzw. Wochen. Eine interessante Möglichkeit, den Prozessschritt zu automatisieren, wäre die prozedurale Erzeugung der Umgebung, wie durch Peter Wonka et al. [50] anhand der Erstellung von Gebäuden im Bruchteil einer Sekunde vorgestellt. Hiermit ist es möglich, fotorealistische Szenen automatisch erstellen zu lassen, wodurch zusätzlich

durch die zufällige Anordnung von ebenfalls zufälligen 3D-Szenenobjekten der Aspekt der Domänenrandomisierung in den Erstellungsprozess einfließen würde.

Ein Beispiel für prozedural generierte Umgebungen zeigt das Spiel No Man's Sky, in dem ganze Planeten sowie deren Flora und Fauna prozedural erzeugt werden.[1]

4.4.5 Erweiterung durch Objektvarianten

Damit ein mittels DOPE trainiertes Modell eine höhere Generalisierbarkeit bei der Erkennung eines Objektes aufweist, kann der Datensatz mit weiteren Varianten eines Objektes gefüllt werden, welche innerhalb des Datensatzes namentlich nicht unterschieden werden. Diesbezüglich könnten beispielsweise zur Erkennung von baugleichen Dosen mit unterschiedlichen Etiketten diese unterschiedlichen Etiketten als Texturen beim Erstellungsprozess zufällig auf dem 3D-Modell ausgetauscht werden. Einen erfolgreichen ähnlichen Ansatz zeigen hierbei Jonathan Tremblay et, al. [46] bei der Erkennung eines Autos. Dabei tauschen sie bei der Erstellung des Datensatzes das 3D-Modell des Autos mit einem anderen Modell aus einer Datenbank von 36 Automodellen aus.

Wie genau sich diese Herangehensweise auf die Qualität der Erkennung einer Dose auswirkt, ist jedoch nicht abzuschätzen und erfordert zur genauen Analyse weitere Experimente.

4.4.6 Änderung der Netzarchitektur

Weitere Optimierungen zur Posenschätzung mittels DOPE können in der in [41] vorgestellten Netzarchitektur unternommen werden. In diesem Zusammenhang könnten Änderungen an den Hyperparametern unternommen werden. Weiter besteht die Möglichkeit des Trainings eines Genoms, das die optimale Netzarchitektur für das Training ermittelt. Dadurch könnte die empirische Herangehensweise des Hinzufügens und Entfernens der verwendeten Schichten des tiefen neuronalen Netzes umgangen werden. Gleichzeitig bedarf die Ermittlung der optimalen Netzarchitektur mittels Genomen, dass unterschiedliche Trainings stattfinden, anhand derer sich an die optimale Netzarchitektur herangetastet wird. Aufgrund der hohen benötigten Trainingszeit des DOPE-Netzes ist dieser Ansatz extrem rechenintensiv.

[1] https://www.newyorker.com/magazine/2015/05/18/world-without-end-raffi-khatchadourian, 15.09.2020

4.5 Alternativen

Unter Berücksichtigung der Schwächen der Posenschätzung mittels DOPE sollte grundsätzlich je nach Anwendungsfall die Wahl der Methode zur Schätzung der 3D-Pose eines Objektes gut überlegt werden. Insbesondere sollten zur Erkennung von Objekten der 3-DOF-Objektklasse andere Verfahren gesucht werden. Gleichzeitig schwindet der Vorteil des in dieser Arbeit gezeigten Deep Learning basierten Verfahrens bei der Erkennung von Objekten der 5-DOF-Objektklasse, da sich hierbei keine Robustheit gegenüber einer teilweisen Verdeckung der Objekte zeigt. Wie bereits im Kapitel der Analyse erwähnt, existieren analytische Ansätze wie SIFT, SURF und ORB zur Objekterkennung und Posenbestimmung, welche für diese Fälle eine bessere Wahl wären. OpenCV[2] stellt eine Implementierung dieser Deskriptoren öffentlich zur Verfügung. Je nach Anwendungsfall eignen sich diese analytischen Verfahren aufgrund ihrer schnellen und zugänglicheren Anwendbarkeit gegebenenfalls mehr. Gleichzeitig liegen deren Schwächen in der Erkennung von verdeckten Objekten [43, 7, 23].

Weitere Implementierungen zur 3D-Objekterkennung bietet Vuforia[3]. Mit dem Vuforia Object Scanner können 3D-Objekte gescannt werden, um deren Position im Anschluss zu erkennen. Hierbei liegt das Problem jedoch darin, dass deren Verfahren aktuell für kleine und höchstens mittelgroße Objekte möglich ist. Weiter besteht auch hierbei das Problem, dass die Objekterkennung nicht sehr robust gegenüber der Verdeckung von Objekten ist.

[2] https://docs.opencv.org, 15.09.2020
[3] https://library.vuforia.com, 15.09.2020

5 Schlussteil

Folgend werden die Inhalte dieser Arbeit kurz zusammengefasst und über die wesentlichen Aussagen und Ergebnisse berichtet. Zuletzt folgt ein Ausblick über weiterführende Möglichkeiten, um in dem Themengebiet dieser Arbeit weiter zu forschen.

5.1 Zusammenfassung

In Kapitel zwei erfolgte zunächst eine Analyse über den aktuellen Stand der Technik sowie eine Eingliederung der Arbeit in ihr Themengebiet. Dabei fand eine Eingrenzung in ihre wichtigen Teilbereiche statt. Zur Herausarbeitung der Notwendigkeit von Posenschätzungen wurde zunächst das Thema der Augmented Reality aufgespannt. Weiter wurde die notwendige Wissensbasis des für diese Arbeit erstellten synthetischen Datensatzes bezüglich RGB-Bildern, Tiefeninformationen und Segmentierungen vermittelt. Als technische Lösung zur Erstellung von synthetischen Daten wurde die Computergrafik beschrieben. Anschließend wurde auf unterschiedliche Möglichkeiten der Annotation von Objekten hingewiesen. Dabei wurde die Wahl der verwendeten Deep Learning Methode begründet, die in der vergleichsweise höheren Robustheit gegenüber verdeckten Objekten liegt. Nachfolgend wurde auf den Reality Gap hingewiesen, dessen Beachtung essenziell bei der Erstellung synthetischer Daten ist. Um die Wahl der verwendeten Messmethode mithilfe eines 'Advanced Realtime Tracking'-System zu begründen, folgte anschließend eine Analyse über verwendete Messmethoden zur Bestimmung der Qualität von Posenerkennungen. Zuletzt wurden Rückschlüsse auf den aktuellen Forschungsstand zur Eingliederung der Forschungsfrage geschlossen. Dabei zeigte sich die Deep Learning basierte Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten als eine der "State-of-the-Art"-Lösungen.

Das dritte Kapitel beschäftigte sich mit der Methodik und Forschung, wobei zunächst das 'Advanced Realtime Tracking'-System als Benchmark vorgestellt wurde. Weiter folgte eine Eingliederung von Objekten in Objektklassen sowie der allgemeine Aufbau der

Experimente. Anschließend folgten die Experimente, bei denen zunächst ein selbst erstelltes Modell zur Erkennung einer Cheez-It Crackerbox dem von NVIDIA öffentlich zur Verfügung gestellten Modell gegenübergestellt wurde. Dabei zeigte sich, dass das selbst erstellte Modell in allen Messungen besser performte. Damit wurde die grundsätzliche Validität für die weiteren Experimente dieser Arbeit belegt.

Im nächsten Experiment 'domänenrandomisiert und fotorealistisch' wurden zwei selbst erstellte Modelle gegenübergestellt. Ein Modell wurde hierbei ausschließlich anhand von domänenrandomisierten Daten und das andere Modell wurde mit domänenrandomisierten und fotorealistischen Daten trainiert. Hierbei wurden die Vorteile der Datensätze insofern deutlich, als dass das domänenrandomisierte Modell sich wesentlich robuster gegenüber Verdeckungen verhielt als das gemischte Modell. Gleichzeitig zeigte sich das gemischte Modell robuster gegenüber Reflexionen und Lichteffekten, was zu einer korrekteren Posenbestimmung bei einer freien Sicht auf die Objekte führte. Damit bestätigte sich die Notwendigkeit der Kombination von domänenrandomisierten und fotorealistischen Daten zur Überbrückung des Reality Gaps.

Nachfolgend wurde die 5-DOF-Objektklasse untersucht, wobei als 5-DOF-Objekt eine unetikettierte Bierflasche diente. Das Experiment zeigte, dass das mittels DOPE erstellte Modell in der Lage ist, die Objektposition von 5-DOF-Objekten korrekt zu bestimmen. Gleichzeitig zeigte sich eine hohe Anfälligkeit von 5-DOF-Objekten gegenüber der Verdeckung dieser Objekte, sodass deren Pose bei bereits geringer Verdeckung nicht mehr korrekt genug berechnet wurde.

Das darauf folgende Experiment widmete sich der 3-DOF-Objektklasse, wobei als Objekt eine Billardkugel verwendet wurde. Hierbei war das mittels DOPE erstellte Model kaum in der Lage, die Pose der Billardkugel ansatzweise korrekt zu bestimmen. Damit stellte sich die Verwendung von DOPE, in diesem Fall von 3-DOF-Objekten, als ungeeignet heraus.

Zuletzt folgte das vierte Kapitel "Ergebnisse und Diskussion", in der eine Bewertung der Deep Learning basierten Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten stattfand. Allgemein ist der Ansatz funktionsfähig, da die trainierten Modelle unter Verwendung von synthetisch erzeugten Daten eine Transferleistung von Virtualität zur Realität erbrachten. In diesem Zusammenhang zeigte sich, dass Objekte der 6-DOF-Objektklasse bedenkenlos verwendet werden können. Gleichzeitig wurden die Schwächen bei der Erkennung von Objekten der 5-DOF- und 3-DOF-Objektklasse hervorgehoben. So werden Elemtente der 5-DOF-Objektklasse nur sehr schlecht erkannt, sobald diese

auch nur teilweise verdeckt sind während Objekte der 3-DOF-Objektklasse insgesamt ungeeignet sind. Weiter wurde auf die Anwendbarkeit für Augmented Reality Anwendungen zur Annotation von Objekten hingewiesen. Insbesondere wurde hierbei der hohe Aufwand und die damit verbundene benötigte Zeit von ca. zwei Wochen für die Integration neuer Objekte verdeutlicht. Anschließend wurden diverse Optimierungsmöglichkeiten vorgestellt, welche je nach Anwendungsfall eingesetzt werden können. Abschließend wurden Alternativen wie SIFT, SURF und ORB zur Objekterkennung und Posenbestimmung noch einmal herangezogen. Insbesondere bei der Erkennung von 5-DOF- und 3-DOF-Objekten sollte deren Einsatz überlegt werden, da hierbei der Vorteil der Resistenz gegenüber Verdeckungen von Deep Learning basierten Ansätzen nicht zu Trage kommt.

5.2 Ausblick

Die Digitalisierung schritt im Laufe dieser Arbeit erwartungsgemäß weiter voran, weshalb in ihr weiterhin ein großes Potenzial steckt, auch in Zukunft das Verstehen unserer Welt zu vereinfachen. Wie diese Arbeit gezeigt hat, ist die Anwendbarkeit von Deep Learning basierter 3D-Objektposenerkennung mit einer ausreichenden Genauigkeit gegeben, was in Zukunft beispielsweise in der Industrie 4.0 ermöglicht, Techniker bei Wartungsarbeiten zu unterstützen. Einen ähnlichen Ansatz hierfür verfolgte auch Henrik Wortmann [51] in seiner Arbeit, in der er erfolgreich das Verständnis von Schaltschränken durch Augmented Reality vereinfachte. In dieser Arbeit wurde jedoch eine allgemeinere Lösung für unterschiedliche Objekte vorgestellt, die an einigen Punkten noch offene Fragestellungen beinhaltet.

Einsatz auf leistungsschwächeren Plattformen

Eine der offenen Fragestellungen ist hierbei eine Erprobung des Einsatzes der Deep Learning basierten Erkennung von 3D-Objektposen auf leistungsschwächeren Plattformen wie Smartphones. Ein interessanter Ansatz wäre der Aufbau einer plattformunabhängigen Client-Server-Architektur, der die in dieser Arbeit verwendete Funktionalität der Posenerkennung integriert. Innerhalb dieser Client-Server-Architektur müsste eine automatische Konfiguration der Kameraparameter für die DOPE auf Basis der intrinsischen Werte der Kamera eines Nutzers berücksichtigt werden.

Erkennung großer und weit entfernter Objekte

Bezüglich der Erkennung großer, weit entfernter Objekte wird aktuell eine Fragestellung von Thomas Kanne-Schluddes [19] untersucht. Thomas Kanne-Schludde versucht hierbei, die GPS-Koordinaten aus Kamerabildern bei der Betrachtung von Sehenswürdigkeiten (am Beispiel Notre Dame) herauszurechnen. Wie die Untersuchungen in dieser Arbeit zeigen, ist der Einsatz von DOPE eine vielversprechende Möglichkeit, um die Posen von 6-DOF-Objekten zu bestimmen. Eine Sehenswürdigkeit wie Notre Dame kann diesbezüglich als Objekt der 6-DOF-Objektklasse gesehen werden. Somit wäre eine Kombination dieser Arbeit mit dem Ansatz von Thomas Kanne-Schludde interessant, in dem die Notre Dame in einem kleineren Maßstab in den Szenen zur Erstellung des Datensatzes integriert wird. Das Training des Modells sollte daraufhin analog wie in dieser Arbeit funktionieren. Anschließend müsste der Maßstab auf die korrekte Entfernung zurückgerechnet werden, um anhand dieser Information eine GPS-Koordinate zu ermitteln.

Erweiterung der Verdeckungsresistenz

Eine weitere offene Fragestellung steckt in der Erweiterung der Verdeckungsresistenz. Diese Arbeit hat gezeigt, dass die Entfernung des Objektes zur Kamera bei der Verdeckung von Objekten der 5-DOF-Objektklasse teilweise größer geschätzt wird. Diesbezüglich wären Optimierungen vorstellbar, in denen der Grad der Verdeckung eines Objektes mit in das Training eines tiefen neuronalen Netzes einfließt. In diesem Zusammenhang kann erprobt werden, dem Netz beizubringen, dass es ein Objekt nur zu einem prozentualen Anteil sieht, sodass anhand dieser Information der Wert der Entfernung korrigiert werden kann.

Qualitätsermittlung bezüglich des Datensatzes

Eine der größten Unsicherheiten bei der Verwendung von DOPE ist die Qualität des Datensatzes. Durch die zufällige Anordnung von Objekten auf den Bildern kann die Repräsentation eines Objektes im Datensatz geringfügig ausfallen. Dies hätte einen direkten Einfluss auf die Qualität bei der Posenerkennung mittels Deep Learning, da das Netz beim Training zu wenig Informationen über das Objekt erhält. In diesem Zusammenhang wäre eine automatisierte Ermittlung der Qualität eines Datensatzes hilfreich, welche prüft, ob die verschiedenen Objekte gut genug innerhalb eines Datensatzes repräsentiert werden.

Skalierbarkeit für neue Objekte

Zuletzt sei noch die offene Fragestellung hinsichtlich der Skalierbarkeit für neue Objekte erwähnt. Wie diese Arbeit zeigte, dauert die Integration neuer zu erkennender Objekte mit dem gewählten Verfahren ca. 10 bis 14 Tage - je nach Qualitätsanspruch. Diese Einschränkung ist für diverse Anwendungsfälle unpraktikabel. Ein Ansatz in diesem Zusammenhang wird aktuell von Keunhong Park et al. [34] untersucht. Diese versuchen, die Skalierbarkeit durch die Verwendung des bestärkenden Lernens (engl. reinforcement learning), was die Erkennung neuer Objekte ohne ein erneutes Training eines neuronalen Netzes ermöglicht, zu erhöhen. In diesem Zusammenhang kann untersucht werden, wie gut dieser Ansatz bezüglich der Posenerkennung im Vergleich zum Ansatz dieser Arbeit funktioniert.

Literaturverzeichnis

[1] Adelson, Edward H. ; Anderson, Charles H. ; Bergen, James R. ; Burt, Peter J. ; Ogden, Joan M.: Pyramid methods in image processing. In: *RCA engineer* 29 (1984), Nr. 6, S. 33–41

[2] Amanatides, John: Realism in computer graphics: A survey. In: *IEEE Computer Graphics and Applications* 7 (1987), Nr. 1, S. 44–56

[3] Azuma, Ronald T.: A survey of augmented reality. In: *Presence: Teleoperators & Virtual Environments* 6 (1997), Nr. 4, S. 355–385

[4] Bay, Herbert ; Tuytelaars, Tinne ; Van Gool, Luc: Surf: Speeded up robust features. In: *European conference on computer vision* Springer (Veranst.), 2006, S. 404–417

[5] Belongie, Serge ; Malik, Jitendra ; Puzicha, Jan: Shape matching and object recognition using shape contexts. In: *IEEE transactions on pattern analysis and machine intelligence* 24 (2002), Nr. 4, S. 509–522

[6] Brachmann, Eric ; Krull, Alexander ; Michel, Frank ; Gumhold, Stefan ; Shotton, Jamie ; Rother, Carsten: Learning 6d object pose estimation using 3d object coordinates. In: *European conference on computer vision* Springer (Veranst.), 2014, S. 536–551

[7] Chien, Hsiang-Jen ; Chuang, Chen-Chi ; Chen, Chia-Yen ; Klette, Reinhard: When to use what feature? SIFT, SURF, ORB, or A-KAZE features for monocular visual odometry. In: *2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)* IEEE (Veranst.), 2016, S. 1–6

[8] Choi, Changhyun ; Baek, Seung-Min ; Lee, Sukhan: Real-time 3D object pose estimation and tracking for natural landmark based visual servo. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* IEEE (Veranst.), 2008, S. 3983–3989

[9] Ding, Junhua ; Kang, Xiaojun ; Hu, Xin-Hua: Validating a deep learning framework by metamorphic testing. In: *2017 IEEE/ACM 2nd International Workshop on Metamorphic Testing (MET)* IEEE (Veranst.), 2017, S. 28–34

[10] Ferwerda, James A.: Three varieties of realism in computer graphics. In: *Human Vision and Electronic Imaging VIII* Bd. 5007 International Society for Optics and Photonics (Veranst.), 2003, S. 290–297

[11] Fu, King-Sun ; Mui, JK: A survey on image segmentation. In: *Pattern recognition* 13 (1981), Nr. 1, S. 3–16

[12] Gaidon, Adrien ; Wang, Qiao ; Cabon, Yohann ; Vig, Eleonora: Virtual worlds as proxy for multi-object tracking analysis. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 4340–4349

[13] Hadsell, Raia ; Sermanet, Pierre ; Ben, Jan ; Erkan, Ayse ; Scoffier, Marco ; Kavukcuoglu, Koray ; Muller, Urs ; LeCun, Yann: Learning long-range vision for autonomous off-road driving. In: *Journal of Field Robotics* 26 (2009), Nr. 2, S. 120–144

[14] Hagen, Margaret A.: *Varieties of realism: Geometries of representational art*. CUP Archive, 1986

[15] Harvey, Inman ; Husbands, Philip ; Cliff, Dave u. a.: *Issues in evolutionary robotics*. School of Cognitive and Computing Sciences, University of Sussex, 1992

[16] Herbert, Bay ; Tinne, Tuytelaars: Van Gool Luc. In: *SURF: speeded up robust features* (2006), S. 404–417

[17] Hoiem, Derek ; Efros, Alexei A. ; Hebert, Martial: Automatic photo pop-up. In: *ACM SIGGRAPH 2005 Papers*. 2005, S. 577–584

[18] Hough, Paul V.: *Method and means for recognizing complex patterns*. Dezember 18 1962. – US Patent 3,069,654

[19] Kanne-Schludde, Thomas: Aufbau einer Pipeline zur Bestimmung von GPS-Koordinaten aus Fotos mithilfe neuronaler Netze. In: *Hochschule für Angewandte Wissenschaften Hamburg (HAW Hamburg)* (2020)

[20] Kato, Hirokazu ; Billinghurst, Mark: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: *Proceedings 2nd IEEE and*

ACM International Workshop on Augmented Reality (IWAR'99) IEEE (Veranst.), 1999, S. 85–94

[21] Keller, James M. ; Gray, Michael R. ; Givens, James A.: A fuzzy k-nearest neighbor algorithm. In: *IEEE transactions on systems, man, and cybernetics* (1985), Nr. 4, S. 580–585

[22] Khan, Waseem: Image segmentation techniques: A survey. In: *Journal of Image and Graphics* 1 (2013), Nr. 4, S. 166–170

[23] Kulkarni, AV ; Jagtap, JS ; Harpale, VK: Object recognition with ORB and its Implementation on FPGA. In: *International Journal of Advanced Computer Research* 3 (2013), Nr. 3, S. 164

[24] LeCun, Yann ; Huang, Fu J. ; Bottou, Leon: Learning methods for generic object recognition with invariance to pose and lighting. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Bd. 2 IEEE (Veranst.), 2004, S. II–104

[25] Lowe, David G.: Object recognition from local scale-invariant features. In: *Proceedings of the seventh IEEE international conference on computer vision* Bd. 2 Ieee (Veranst.), 1999, S. 1150–1157

[26] Lowe, David G.: Distinctive image features from scale-invariant keypoints. In: *International journal of computer vision* 60 (2004), Nr. 2, S. 91–110

[27] Marion, Pat ; Florence, Peter R. ; Manuelli, Lucas ; Tedrake, Russ: Label fusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)* IEEE (Veranst.), 2018, S. 1–8

[28] Matthias Nitsche, Stephan H.: Development of an End-to-End Deep Learning Pipeline. In: *Hochschule für Angewandte Wissenschaften Hamburg (HAW Hamburg)* (2019)

[29] Michels, Jeff ; Saxena, Ashutosh ; Ng, Andrew Y.: High speed obstacle avoidance using monocular vision and reinforcement learning. In: *Proceedings of the 22nd international conference on Machine learning*, 2005, S. 593–600

[30] Milgram, Paul ; Takemura, Haruo ; Utsumi, Akira ; Kishino, Fumio: Augmented reality: A class of displays on the reality-virtuality continuum. In: *Telemanipulator and telepresence technologies* Bd. 2351 International Society for Optics and Photonics (Veranst.), 1995, S. 282–292

[31] Minsky, Marvin: Steps toward artificial intelligence. In: *Proceedings of the IRE* 49 (1961), Nr. 1, S. 8–30

[32] Nischwitz, Alfred ; Fischer, Max ; Haberäcker, Peter: *Computergrafik und Bildverarbeitung*. Springer, 2007

[33] Oberweger, Markus ; Rad, Mahdi ; Lepetit, Vincent: Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, S. 119–134

[34] Park, Keunhong ; Mousavian, Arsalan ; Xiang, Yu ; Fox, Dieter: LatentFusion: End-to-End Differentiable Reconstruction and Rendering for Unseen Object Pose Estimation. In: *arXiv preprint arXiv:1912.00416* (2019)

[35] Perez, Luis ; Wang, Jason: The effectiveness of data augmentation in image classification using deep learning. In: *arXiv preprint arXiv:1712.04621* (2017)

[36] Rublee, Ethan ; Rabaud, Vincent ; Konolige, Kurt ; Bradski, Gary: ORB: An efficient alternative to SIFT or SURF. In: *2011 International conference on computer vision* Ieee (Veranst.), 2011, S. 2564–2571

[37] Saxena, Ashutosh ; Sun, Min ; Ng, Andrew Y.: Learning 3-d scene structure from a single still image. In: *2007 IEEE 11th International Conference on Computer Vision* IEEE (Veranst.), 2007, S. 1–8

[38] Schmidhuber, Jürgen: Deep learning in neural networks: An overview. In: *Neural networks* 61 (2015), S. 85–117

[39] Shorten, Connor ; Khoshgoftaar, Taghi M.: A survey on image data augmentation for deep learning. In: *Journal of Big Data* 6 (2019), Nr. 1, S. 60

[40] Silberman, Nathan ; Hoiem, Derek ; Kohli, Pushmeet ; Fergus, Rob: Indoor segmentation and support inference from rgbd images. In: *European conference on computer vision* Springer (Veranst.), 2012, S. 746–760

[41] Spallek, Dustin: PIPELINE ZURERKENNUNG UNDSCHAETZUNG DER6-DOF-POSE BEKANNTEROBJEKTE. In: *Hochschule für Angewandte Wissenschaften Hamburg (HAW Hamburg)* (2019)

[42] Spallek, Dustin: Erzeugung synthetischer Daten zur Erkennung der 6-DoF-Pose bekannter Objekte mittels Deep Learning. (2020)

[43] Tareen, Shaharyar Ahmed K. ; Saleem, Zahra: A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In: *2018 International conference on computing, mathematics and engineering technologies (iCoMET)* IEEE (Veranst.), 2018, S. 1–10

[44] To, Thang ; Tremblay, Jonathan ; McKay, Duncan ; Yamaguchi, Yukie ; Leung, Kirby ; Balanon, Adrian ; Cheng, Jia ; Hodge, William ; Birchfield, Stan: *NDDS: NVIDIA Deep Learning Dataset Synthesizer*. 2018. – https://github.com/NVIDIA/Dataset_Synthesizer

[45] Tobin, Josh ; Fong, Rachel ; Ray, Alex ; Schneider, Jonas ; Zaremba, Wojciech ; Abbeel, Pieter: Domain randomization for transferring deep neural networks from simulation to the real world. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* IEEE (Veranst.), 2017, S. 23–30

[46] Tremblay, Jonathan ; Prakash, Aayush ; Acuna, David ; Brophy, Mark ; Jampani, Varun ; Anil, Cem ; To, Thang ; Cameracci, Eric ; Boochoon, Shaad ; Birchfield, Stan: Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, S. 969–977

[47] Tremblay, Jonathan ; To, Thang ; Sundaralingam, Balakumar ; Xiang, Yu ; Fox, Dieter ; Birchfield, Stan: Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects. (2018)

[48] Tremblay, Jonathan ; To, Thang ; Sundaralingam, Balakumar ; Xiang, Yu ; Fox, Dieter ; Birchfield, Stan: Deep object pose estimation for semantic robotic grasping of household objects. In: *arXiv preprint arXiv:1809.10790* (2018)

[49] Wei, Shih-En ; Ramakrishna, Varun ; Kanade, Takeo ; Sheikh, Yaser: Convolutional pose machines. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, S. 4724–4732

[50] Wonka, Peter ; Wimmer, Michael ; Sillion, François ; Ribarsky, William: Instant architecture. In: *ACM Transactions on Graphics (TOG)* 22 (2003), Nr. 3, S. 669–677

[51] Wortmann, Henrik: Objekterkennung unter Nutzung von MachineLearning für Augmented Reality Anwendungen. In: *Hochschule für Angewandte Wissenschaften Hamburg (HAW Hamburg)* (2020)

[52] Xiang, Yu ; Schmidt, Tanner ; Narayanan, Venkatraman ; Fox, Dieter: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In: *arXiv preprint arXiv:1711.00199* (2017)

[53] Zaitoun, Nida M. ; Aqel, Musbah J.: Survey on image segmentation techniques. In: *Procedia Computer Science* 65 (2015), S. 797–806

A Anhang

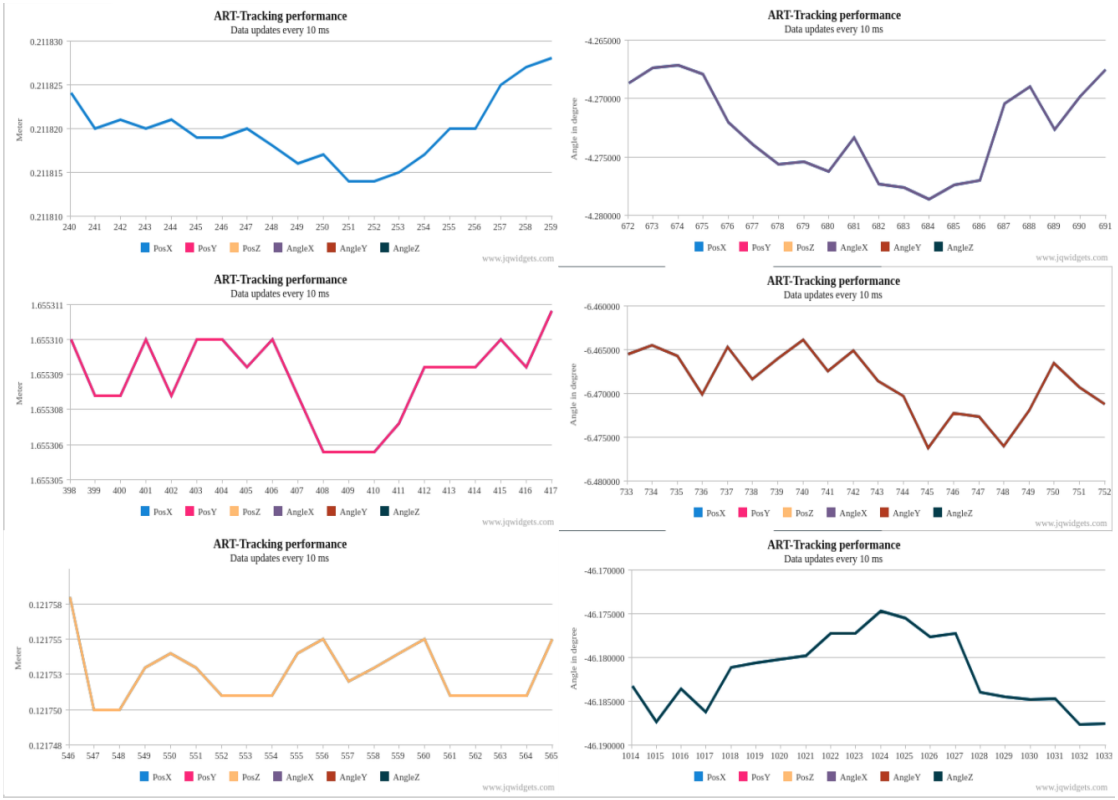

Abbildung A.1: ART-Performance-Graph

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Masterarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Deep Learning basierte Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	_____
Ort	Datum	Unterschrift im Original