HAW HAMBURG

Project Report

Nadia Hintze

Erzeugung von Trainingsdaten zur Bestimmung der 3D-Pose von Schachfiguren

Betreuung durch: Prof. Dr. Kai von Luck
Eingereicht am: 18.08.2023

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

Inhaltsverzeichnis

1 Einleitung

Künstliche neuronale Netze sind eines der aktuell aktivsten Forschungsfelder der Informatik. Die Pose realer Objekte im Raum zu bestimmen ist eine Aufgabe, die mit überwachtem Lernen bearbeitet wird. Die Ergebnisse solcher Verfahren sind jedoch stark von den zur Verfügung stehenden Trainingsdaten abhängig. Um diese zu gewinnen, müssen sowohl Aufnahmen der zu suchenden Objekte, als auch Messungen ihrer Posen für die dazu passenden Labels vorliegen (siehe Abb. 1). Die Messung einer Pose findet dabei in bis zu neun Freiheitsgraden[1] statt, jeweils drei für ihre Position, Rotation und Skalierung. [5]

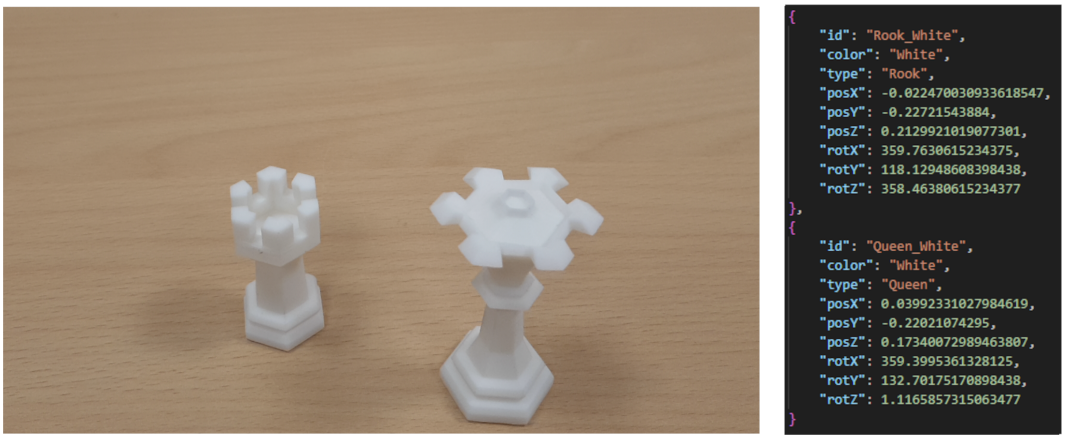

Abbildung 1: a) Beispiel für ein Trainings-Bild, b) Dazugehöriges Label mit sechs DoF
Quelle: Eigene Abbildung

Es gibt bereits einige Datensätze mit dazugehörigen Projekten, die dokumentieren, wie diese Daten gewonnen werden können (siehe Kap. 2). Im vorliegenden Projekt wird eine alternative Methode vorgeschlagen, die auf den Objekten befestigte 2D-Marker verwendet, um ihre Pose im Raum festzustellen. Im diesem Projekt sollen damit die Posen von Schachfiguren erkannt werden. Dies soll als Grundlage dafür dienen, eine Augmented Reality Anwendung zu entwerfen, die es ermöglicht, mit einem virtuellen Spieler Schach spielen zu können (siehe Kap. 5). Der erste Zweck der Messung besteht darin, den aktuellen Zustand des Spielfelds erkennen zu können. Der zweite ist die Gestaltung eines User-Interfaces, dass sich in AR optisch in die Realität einfügt, indem es sich an den Posen der Figuren im Raum orientiert. Um beide Aufgaben erfüllen zu können, werden sechs Freiheitsgrade benötigt: drei der Rotation und drei der Position. Die Skalierung

[1] Im Folgenden abgekürzt durch „DoF“

kann dabei außer Acht gelassen werden, da diese bei den Schachfiguren unveränderbar festgelegt ist.

Im Rahmen des vorliegenden Projekts wird ein prototypisches System erstellt, das die Marker Erkennung des Tracking Frameworks Vuforia dafür nutzt, zunächst die Posen gesuchter Figuren zu ermitteln. Auf Basis dieser Messungen labelt das System automatisch Fotos der Figuren, ohne dass die Marker in den Trainingsdaten zu sehen sind (siehe Abb. 1). Auf diesem Weg soll am Ende des Projekts ein Datensatz entstehen, der zukünftig für weitere Projekte verwendet werden kann.

Zu diesem Zweck werden im folgenden Kapitel 2 zunächst verschiedene Methoden zur Gewinnung vergleichbarer Trainingsdaten vorgestellt und eine Abgrenzung der Methoden von der in diesem Projekt umgesetzten Herangehensweise wird vorgenommen. Anschließend wird das eigene System in Kap. 3 umgesetzt und die Entwicklung der Pipeline dokumentiert. Kap. 4 enthält eine Evaluation der gewählten Methode und Kap. 5 bietet einen Ausblick auf zukünftig verfolgbare Erweiterungen des hier bearbeiteten Projekts. In Kap. 6 werden schließlich die wichtigsten Ergebnisse des Projekts zusammengefasst und ein Fazit wird gezogen.

2 Methoden zur Trainingsdaten Erstellung

An dieser Stelle werden einige Herangehensweisen vorgestellt, mit denen Trainingsdaten für die 6DoF-Posen-Erkennung von Objekten erstellt werden können. Dabei wird sich insbesondere darauf konzentriert, wie die Ground Truth Posen für das Labeling der Daten ermittelt wurden. Zum Schluss wird die in diesem Projekt umgesetzte Methode vorgestellt und von den anderen abgegrenzt.

Es existiert eine Projekt Homepage, die Datensätze für Benchmarks in dem Anwendungsbereich der Objektposen-Erkennung sammelt [2]. Die ersten folgenden Methoden orientieren sich an der dort veröffentlichten Liste.
Ein in einigen dieser Projekte eingesetztes Hilfsmittel zur Erkennung von Posen ist die Erkennung von Markern. Beispielsweise kann dafür eine Tischfläche mit einer Vielzahl solcher Marker ausgestattet werden (siehe Abb. 2).

In der Mitte liegen dann die abzufilmenden Objekte, sodass auf diese Weise die Kameraposition relativ zu der Tischfläche bestimmt werden kann. Diese Technologie wurde

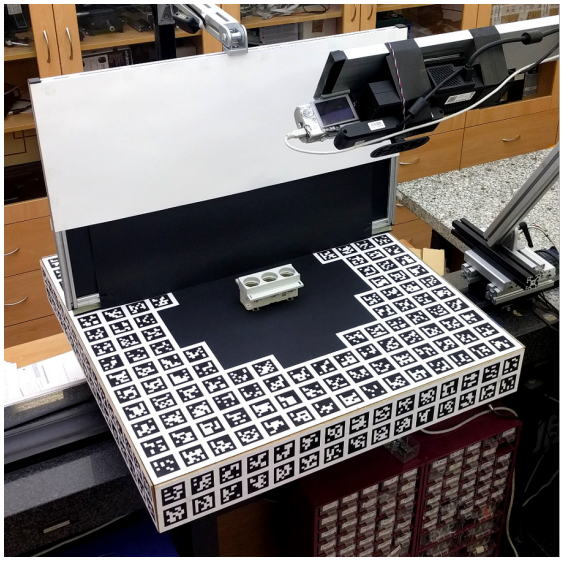

Abbildung 2: Marker-Tisch zum Abfilmen der gesuchten Objekte
Quelle: Hodan u. a. [10]

z.B. in dem Datensatz *T-Less* von Hodan u. a. [10] und in *HomebrewDB* von Kaskman u. a. [14] verwendet. In beiden Arbeiten musste anschließend noch die Pose der einzelnen Objekte relativ zur Tischfläche bzw. zur Kamera festgestellt werden. Hierfür wurden die Aufnahmen mit einem RGB-D Sensor angefertigt, sodass aus diesen Aufnahmen ein digitales 3D-Modell der Umgebung rekonstruiert werden konnte. Hodan u. a. [10] haben für die Feststellung der Objekt-Posen einen Benutzer manuell CAD-Modelle der gesuchten Objekte so ausrichten lassen, dass sie zu dem rekonstruierten Umgebungsmodell passen. Kaskman u. a. [14] haben dieses Problem mithilfe des von Drost u. a. [3] entwickelten Algorithmus gelöst. Hierbei wird die Pose ermittelt, indem zuerst Merkmals-Repräsentationen von Punkt-Paaren berechnet werden und anschließend mithilfe einer Abstimmung ein lokales Matching stattfindet.

Ein weiteres Beispiel aus der BOP-Liste ist der Datensatz *Hope* von Tyree u. a. [23]. Hier wurde kein Marker-Tisch verwendet, sondern die Pose der Objekte direkt mithilfe eines RGB-D Sensors bestimmt. Dafür haben die Autoren zwei Varianten implementiert. Bei der ersten bestimmt ein Benutzer manuell korrespondierende Punkte in dem RGB-Bild der Umgebung und der 2D-Texture-Map des gesuchten Objektes. Anschließend wird die Pose des Objektes im Raum bestimmt, indem über die PnP-Methode und RANSAC die Verschiebung zueinander bestimmt wird. Die zweite implementierte Variante nutzt ein texturiertes CAD-Modell des gesuchten Objektes und eine RGB-D Aufnahme der

Umgebung. Hier wird die Procrutes Analyse verwendet, um die Pose des Modells in der Aufnahme zu bestimmen[23].

Wenn das gesuchte Objekt als digitales 3D-Modell existiert, können außerdem rein synthetische Daten erzeugt werden. Das geschieht, indem in einer Engine eine virtuelle Umgebung inklusive des gesuchten Objekts zufällig generiert wird [21][19]. Auf diese Weise ist die Pose des Objekts bekannt und eine Aufnahme der virtuellen Umgebung kann erstellt werden. Diese Methode hat die Vorteile, dass zum einen eine große Vielzahl von Daten innerhalb sehr kurzer Zeit erstellt werden kann, nachdem die Anwendung zur Generierung geschrieben wurde und zum anderen die Umgebung beliebig gestaltet werden kann und sich nicht an Restriktionen der Realität halten muss. Tremblay u. a. [22] zeigten außerdem, dass der Unterschied zwischen virtuellen und realen Daten, der so genannte Reality Gap, für das Trainingsergebnis kein Problem darstellt, wenn ausreichend realistische Daten verwendet werden.

In den folgenden Kapiteln wird ein anderer Ansatz verfolgt. Hier ist die Grundidee, jedes auf den Trainingsbildern gesuchte, reale Objekt mit einem eigenen Marker auszustatten, der ähnlich wie in den Arbeiten mit dem Marker-Tisch zur Kalkulation der Kamera Pose relativ zu der des jeweiligen Objektes verwendet wird. Hierbei wird also der Schritt übersprungen, dass aus einem RGB-D Bild der Umgebung ein CAD-Modell angefertigt und dann die Pose der einzelnen Objekte darin festgestellt werden muss.
Die Marker werden zwar für die Bestimmung der Pose gebraucht, sollen jedoch nicht auf den Trainingsbildern zu sehen sein, da sie ansonsten das Trainings-Ergebnis beeinflussen können. Daher muss ein Mechanismus geschaffen werden, durch den die Marker für die Posen-Bestimmung an den Objekten angebracht werden und für die Aufnahme der Trainingsbilder wieder abgesetzt werden können, ohne dass die Objekte währenddessen ihre Pose verändern.
Da in Smartphones für eine Posen-Bestimmung nutzbare Sensoren, wie eine RGB-Kamera und weitere IMU-Sensorik bereits integriert sind und es einige Frameworks gibt, die das Tracking von Markern damit unterstützen, wurde sich für die Entwicklung einer Smartphone Anwendung entschieden.

3 Umsetzung

Im Folgenden wird eine Anwendung entwickelt, die diese Methode prototypisch implementieren und auf diese Weise testbar machen soll. Dafür werden zunächst in Kap. 3.1 Tools und Technologien ausgesucht, die für die Umsetzung verwendet werden. Anschließend widmet sich Kap. 3.2 der Gestaltung der zu trackenden Figuren mit aufsetzbaren Markern zur Ermittlung ihrer Pose. In Kap. 3.3 wird schließlich die Anwendung erstellt, die automatisch aus damit geschossenen Aufnahmen der Figuren die Posen ermittelt, die Fotos mit den dazugehörigen Informationen labelt und das Ganze als .json-Datei abspeichert.

3.1 Tools und Technologien

Zunächst wird ein Tool benötigt, mit dem die Schachfiguren und die Halterungen für die 2D-Marker virtuell modelliert werden können. Zu diesem Zweck wurde hier das Tool *Blender 3.5* [1] verwendet. Gedruckt wurden alle Modelle mit dem Kunstharz-Drucker *Formlabs 3+* [6]. Die 2D-Marker wurden mit dem Bildbearbeitungsprogramm *Gimp 2.10* [20] entworfen.
Darüber hinaus wird eine Entwicklungsumgebung gebraucht, in der die Anwendung zur Trainingsdaten-Aufnahme erstellt werden kann. Dafür wurde hier die 3D-Engine *Unity3D* [24] eingesetzt. Sie bringt einige nützliche Funktionalitäten mit. Unity3D unterstützt nativ die Entwicklung für verschiedene Plattformen, darunter iOS- und Android-Geräte. Die Zielplattform kann dabei für das Deployment flexibel gewechselt werden, ohne dass bei der sonstigen Anwendungsentwicklung unterschieden werden muss. Außerdem unterstützen einige Funktionen gezielt die Entwicklung von Augmented Reality Anwendungen. So sind beispielsweise verschiedene Frameworks, mit denen Objekte getrackt werden können, einfach importierbar. Ein solches Framework, das hier benutzt wird, ist *Vuforia* [16]. Es funktioniert sowohl auf iOS- als auch auf Android-Geräten und hat bereits in früheren Projekten ein stabiles Tracking mit sechs Degree of Freedom geliefert [9].

3.2 Gestaltung der physischen Schachfiguren mit Marker-Halterungen

Als Grundlage für das Design der Schachfiguren dienen 3D-Modelle von dem Sketchfab Benutzer „hanoldaa“ [8]. Wie in Kap. 1 bereits erwähnt wurde, ist es besonders heraus-

fordernd Objekte zu erkennen, die wenig optisch markante Merkmale aufweisen, was bei Schachfiguren oftmals der Fall ist. Um diesen Umstand abzumildern, sind die Objekte so geformt, dass die Elemente, die sich von Figur zu Figur unterscheiden, optisch hervorgehoben werden. Das wird erzielt, indem insbesondere die Köpfe der Figuren mit ihren für die Figur spezifischen Silhouetten betont werden. So ist beispielsweise die Anzahl der Polygone des kugelförmigen Kopfes des Bauern erhöht, sodass dieser runder wirkt und die Zinnen des Turms sind in die Länge gezogen, sodass dort die senkrechten Kanten visuell prominenter wirken. Außerdem sind die Zinnen so geformt, dass die Winkel nicht alle 90 Grad groß sind und auf diese Weise zusätzliche Schatten dazwischen geworfen werden, wenn das Licht beispielsweise von oben kommt (siehe Abb. 3).

Abbildung 3: links: Original Schachfiguren aus [8], rechts: veränderte Version
Quelle: Eigene Abbildung

Die nächste Aufgabe besteht darin, einen Mechanismus zu entwerfen, über den 2D-Marker auf den Figuren befestigt und wieder abgenommen werden können, ohne dass die Figur dafür bewegt wird. Eine erste Option für die Befestigung ist der Einsatz von Magneten. Dafür sind sowohl in den Marker-Halterungen als auch in den Figuren Öffnungen vorgesehen, durch die Magneten jeweils nahe an der ansetzenden Flächen befestigt werden können (siehe Abb. 4).

Da die Marker Ausrichtung bei mehreren Messungen konstant zur Ausrichtung der Figur passen muss, muss gewährleistet werden, dass der Marker genauso ausgerichtet wieder auf eine Figur gesetzt werden kann, wie er abgenommen wurde. Um das zu erreichen, gehört eine Steckverbindung zu den Modellen, durch die der Marker nur in bestimmten Ausrichtungen befestigt werden kann (siehe Abb. 5). Diese Verbindung ist so gestaltet, dass die Halterung weder in der Öffnung zu weit drehbar ist und dadurch die Ausrichtung nicht akkurat bleibt, noch das Einstecken zu schwer funktioniert, sodass die Figur wahrscheinlich beim Abnehmen des Markers bewegt wird. Da die Schachfiguren jeweils zu sechs Seiten symmetrisch aufgebaut sind, reicht es, die Steckverbindung ebenfalls so zu gestalten. Selbst wenn der Marker anders wieder befestigt wird als er abgenommen wurde, stimmt die Optik der Figur durch die Symmetrie mit der Ausrichtung des Marker

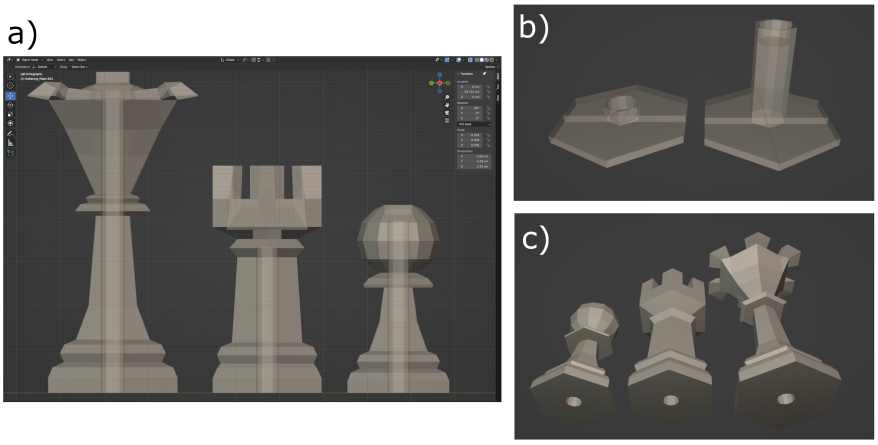

Abbildung 4: Magnet Öffnungen a) in den Schachfiguren (transparente Ansicht), b) in den Marker-Halterungen (transparente Ansicht), c) am Boden der Schachfiguren
Quelle: Eigene Abbildung

überein. Zugleich ist es es relativ einfach eine Drehung des Markers zu finden, mit der die Halterung auf die Figur passt, da es sechs Möglichkeiten gibt, im Gegensatz zu z.B. einer dreieckigen Gestaltung des Steck-Mechanismus.

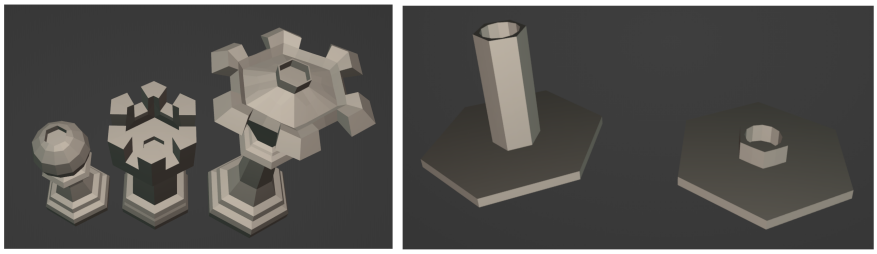

Abbildung 5: Sechseckige Steckverbindung für Marker-Halterungen, Quelle: Eigene Abbildung

Auch die Vuforia-Marker müssen für den Versuchsaufbau entworfen werfen. Hier bilden drei Bilder von Iago Casabiell González [11][13][12] die Grundlage. Da diese Bilder in der ursprünglichen Form nicht genügend visuell markante Features aufweisen, werden sie nicht zuverlässig von Vuforia erkannt. Aus diesem Grund wurden sie durch weitere optische Details erweitert.

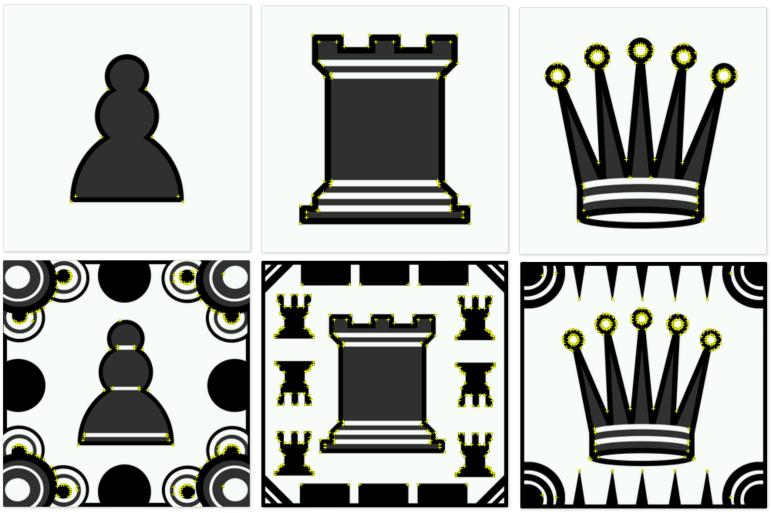

Abbildung 6: Marker vor und nach der Bearbeitung. Von Vuforia wahrgenommene Features wurden gelb markiert.
Quelle: Eigene Abbildung

3.3 Entwicklung der Anwendung zur Trainingsdaten Erstellung

Um die benötigten Funktionen der zu erstellenden Anwendung zu definieren, wurde zuerst festgelegt, wie eine Aufnahme von Trainingsdaten ablaufen soll. Dieser Vorgang sieht wie folgt aus:

1. Der Benutzer platziert die zu suchenden Schachfiguren vor der Kamera und befestigt die passenden Marker darauf.
2. Der Benutzer stellt über Feedback des User Interfaces fest, ob die Marker von Vuforia richtig erkannt wurden.
3. Der Benutzer löst über einen Knopfdruck die Aufnahme der Tracking-Ergebnisse aus.
4. Die Anwendung merkt sich die Posen und IDs der erkannten Marker.
5. Die Anwendung gibt das Feedback, dass diese Daten aufgenommen wurden.
6. Der Benutzer entfernt die Marker von den Schachfiguren, ohne dabei die Figuren zu bewegen.
7. Der Benutzer drückt einen Knopf, um ein Foto zu schießen.

8. Die Anwendung speichert das geschossene Foto und die vorher erkannten Marker-Daten als .json-Datei ab.

9. Die Anwendung liefert das Feedback, dass die Aufnahmen erstellt wurden.

Um die darin enthaltenen Aufgaben der Anwendung umzusetzen, wurden die in Abb. 7 dargestellten Abläufe implementiert.

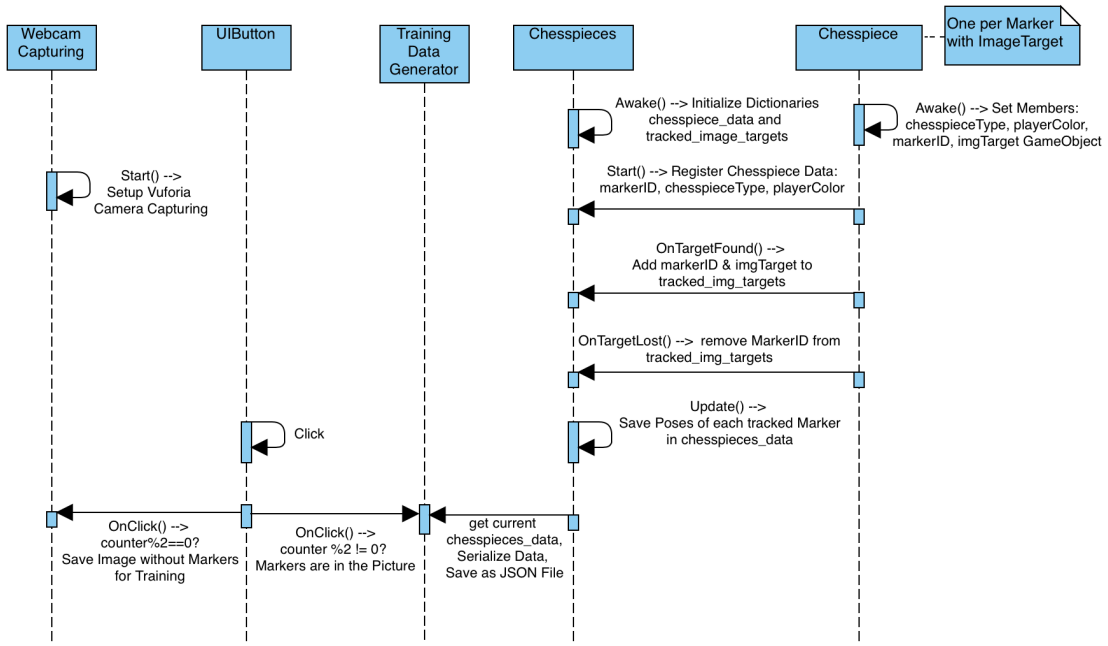

Abbildung 7: Pipeline Ablauf, Quelle: Eigene Abbildung

Chesspiece ist ein Script, von dem es für jeden erkannten Marker eine Instanz gibt, die jeweils Informationen über die Pose und die ID der Schachfigur verwalten und weitergeben kann. *Chesspieces* existiert zur Laufzeit dagegen nur einmal und enthält Daten-Kollektionen, um die aufgenommenen Daten der *Chesspiece* Instanzen zu verwalten. Das Script *Webcam Capturing* ist dafür zuständig, Screenshots von dem über Vuforia aufgenommenen Kamerabild festhalten zu können. Der *Training Data Creator* hat die Aufgabe, alle gesammelten Daten in einer .json-Datei zu verpacken und abzuspeichern.

Das User Interface besteht neben dem Kamerabild aus drei digitalen Elementen und ist in Abb. 8 zu sehen.

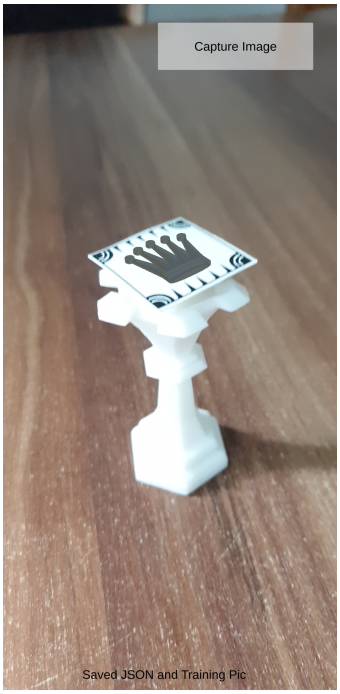

Abbildung 8: User Interface, Quelle: Eigene Abbildung

Das erste Element ist ein Knopf, der vom Benutzer gedrückt wird, wenn die Posen der Marker im Bild festgehalten oder wenn das Kamerabild ohne Marker abgespeichert werden soll. Das zweite Element ist ein Text-Feld, dass Bescheid gibt, wenn die beiden Abläufe jeweils erfolgt sind, sodass der Benutzer weiß, dass er nun die Marker abnehmen oder die Objekte neu platzieren kann. Das letzte Element sind Augmented Reality Objekte, die auf den Markern angezeigt werden. Sie wurden passgenau für die Symbole auf den Markern angefertigt (siehe Abb. 9), sodass man über ihre Position auf dem Marker sehen kann, ob die Erkennung der Lage des Markers korrekt ist.

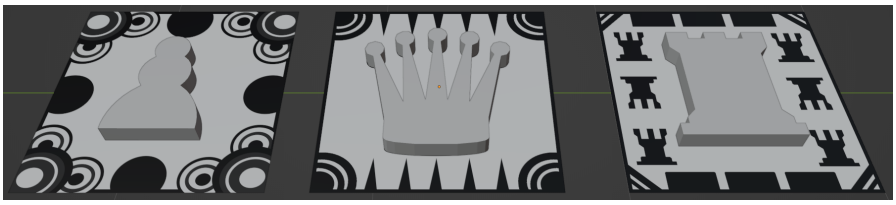

Abbildung 9: Marker Objekte, Quelle: Eigene Abbildung

Zum Testen der Anwendung wurde ein Datensatz mit 53 Bildern aufgenommen. Die Anwendung läuft auf dem Android Smartphone Samsung Galaxy S9[17], das wiederum von

dem Stabilisator „Vlog Pocket“ von Feiyutech gehalten wurde, damit es nicht in Bewegung gerät, während die Objekte hinter der Kamera bewegt werden. Um zu gewährleisten, dass die Schachfiguren beim Abnehmen des Markers nicht bewegt werden, wurden die Figuren zusätzlich mit doppelseitigem Klebeband am Boden befestigt. Abb. 10 zeigt ein Beispiel für eine Aufnahme von Trainingsdaten.

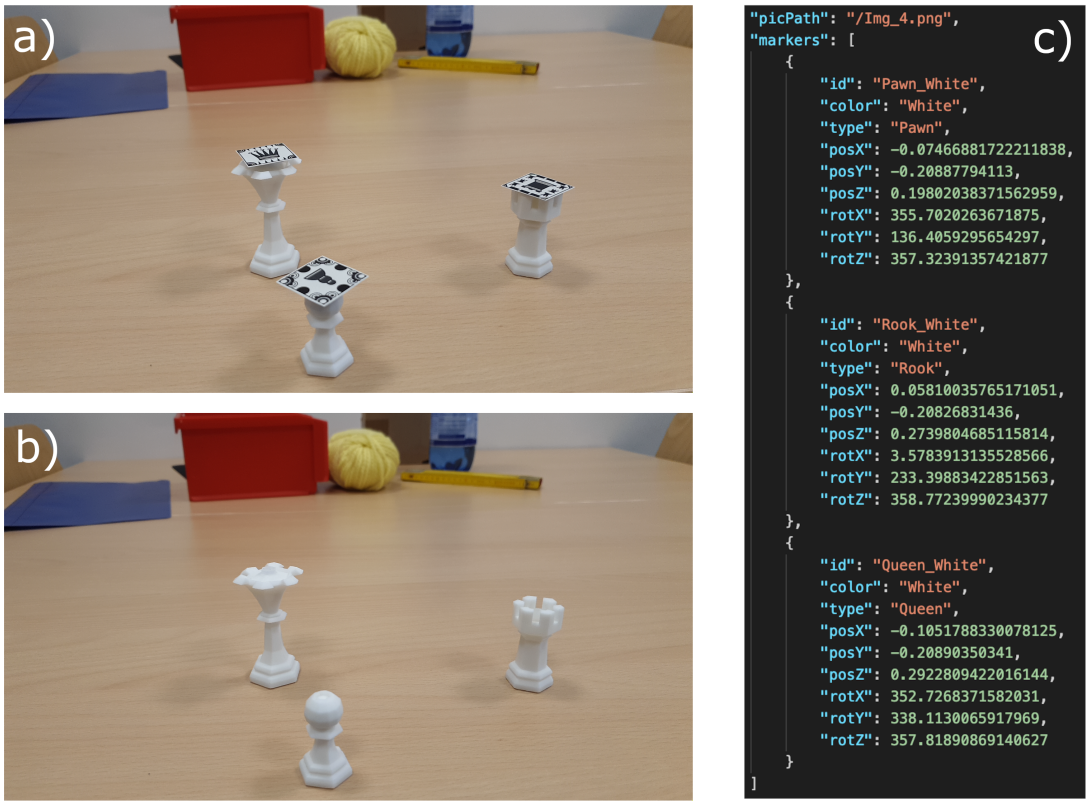

Abbildung 10: Aufnahmen Beispiel: a) mit Markern b) ohne Marker c) Ground Truth Messung, Quelle: Eigene Abbildung

4 Evaluation

In diesem Kapitel findet eine Beurteilung statt, wie gut sich die hier verfolgten Methoden eignen, um auf diese Weise Trainingsdaten zu erstellen.

Zuerst einmal wird der 3D-Druck der Schachfiguren inklusive der Erstellung der Marker-Halterungen betrachtet. Die hier zur Modellierung verwendete Software Blender bietet die Funktion an, virtuelle Objekte zu vermessen, sodass die gedruckte Version bestimmten Maßen entsprechen kann. Das ist insbesondere für den Entwurf der Steckverbindungen und der Magnet-Öffnungen essenziell. Jedoch bringen Drucke mit dem hier benutzten 3D-Drucker die Eigenschaft mit sich, dass sich das Kunstharz ungewollt in kleinen Ecken sammeln kann, sodass die gedruckte Form verändert wird. So mussten zum Beispiel die Magnet Öffnungen meistens aufgebohrt oder die Böden abgeschliffen werden. Schon kleine Abweichungen der gedruckten Figur zu der modellierten Version können dazu führen, dass die Marker Halterung nicht mehr zuverlässig so sitzt, wie es für die Anwendung festgelegt wurde. Insbesondere die Rotation der Halterung kann dabei ein Problem werden, wenn die Halterung beispielsweise in der Öffnung kippelt. Darauf ist somit bei der Erstellung der physischen Figuren stark zu achten, was ggf. durch die Wahl des 3D-Druckers und der Nachbearbeitung abgemildert werden kann. Ein weiterer Ansatz zur Verbesserung wäre eine Kalibrierung der Anwendung. Beispielsweise könnte dafür in AR die modellierte Figur mit der Halterung im in Blender modellierten Soll-Zustand unter dem realen Marker eingeblendet werden. Indem der Nutzer nun die Neigung der virtuellen Figur optisch an der Realität anpasst, könnte in der Anwendung die erkannte Pose nachjustiert werden.
Ein weiterer Faktor, der hier die Korrektheit der Posen-Messung maßgeblich mitbestimmt, ist die Wahl des Tracking-Frameworks. Das Tracking von Vuforia ist nicht immer fehlerlos. Gerade, wenn die Marker annähernd orthogonal zum Kamera-Blickwinkel gedreht sind, kann die gemessene Pose stark von der tatsächlichen abweichen. Um diesen Umstand für den Benutzer transparenter zu machen, wurden in Kap. 3.3 AR-Objekte hinzugefügt, die optisch mit den Markern abgeglichen werden können. Somit kann der Benutzer zumindest grob falsche Messungen vermeiden. Kleine Fehler werden dadurch jedoch nicht unbedingt offensichtlich, sodass diese kleinen Abweichungen in den Messungen auftreten können.

Neben Ungenauigkeiten der Messungen hat die in diesem Projekt erarbeitet Methode außerdem Restriktionen, was die Anordnung aufzunehmender Objekte anbelangt. So können beispielsweise verdeckte Marker dazu führen, dass die dazugehörigen Figuren nicht

von der Anwendung wahrgenommen werden. Da hier die Marker oben auf den gesuchten Objekten verankert sind, kann ein zu niedriger Blickwinkel der Kamera außerdem zum Problem werden, wenn die Marker nicht mehr sichtbar sind. Dieser Umstand wurde hier jedoch vernachlässigt, da in der Anwendung, für die die Trainingsdaten zukünftig benutzt werden sollen, die Figuren in erster Linie von oben, aus der Sicht eines Spielers zu sehen sind.

5 Ausblick

In der Zukunft sind die Implementierung weiterer Möglichkeiten zur Evaluation denkbar. So kann beispielsweise die visuelle Unterscheidbarkeit der erstellten Schachfiguren gemessen werden. Dafür könnten visuell markante Merkmale, wie die *SIFT*-Features [15], auf Kamerabildern aus festen Blickwinkeln bestimmt werden. Durch die Anzahl und Unterschiedlichkeit der Features kann die Erkennbarkeit der Objekte bewertet werden. Ähnlich dazu ist die Einschätzung der Wiedererkennbarkeit von Markern durch Vuforia in der vorherigen Abb. 6. Eine weitere Möglichkeit zur Evaluation betrifft die Korrektheit der gemessenen Posen. Dafür ist z.B. denkbar, in der Engine Unity3D eine Funktionalität zu implementieren, die die gemessene Pose auf das passende virtuelle Modell anwendet, um anschließend zu vergleichen, ob das resultierende Kamerabild mit dem aufgenommenen übereinstimmt.

Endgültig evaluierbar sind die Ergebnisse dieses Projektes erst, wenn sie in einem KNN als Trainingsdaten verwendet wurden. Dort können jedoch auch Faktoren zu unzureichenden Ergebnissen führen, die nicht direkt mir der hier verwendeten Methode zur Trainindsdaten Erstellung zu tun haben. So haben beispielsweise die Menge der Trainingsdaten oder die vorhandene Varianz der Daten maßgebliche Auswirkungen auf ein KNN Trainings-Ergebnis [7]. Eine Variation der Trainingsdaten mit der in diesem Projekt entwickelten Anwendung ist möglich, indem beispielsweise Hintergründe verändert, andere Objekte mit in die Umgebung gesetzt, verschiedene Lichtverhältnisse oder unterschiedliche Fokus-Modi der Kamera verwendet werden. Jedoch ist die Anwendung darauf angewiesen, dass 2D-Marker nicht visuell verdeckt werden (siehe Kap. 4). Daher ist eine Variation der Trainingsdaten nur bedingt möglich.

Wenn die so erstellte Variation des Datensatzes nicht ausreicht, könnte der Datensatz außerdem mit Methoden aus der Literatur erweitert werden. Da die Schachfiguren als

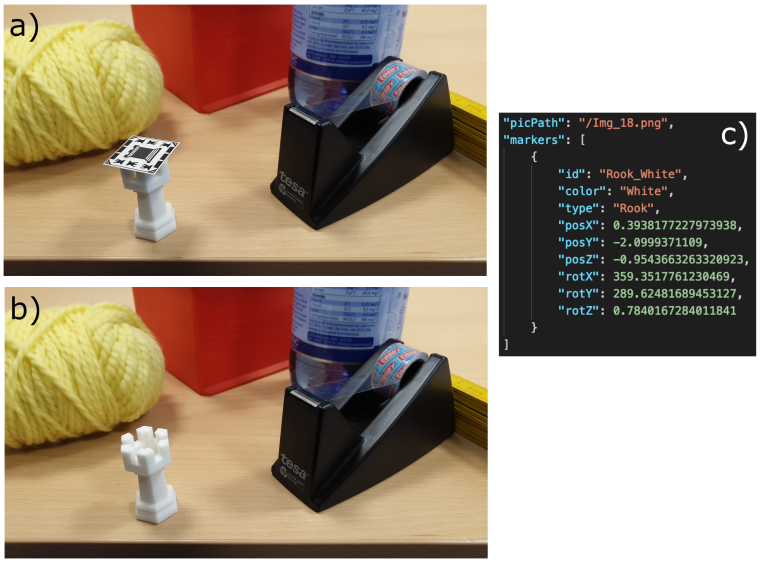

Abbildung 11: Trainingsdaten Beispiel mit anderen Objekten in der Umgebung, Quelle: Eigene Abbildung

virtuelle 3D-Modelle vorliegen, könnten z.B. zusätzlich synthetische Daten erzeugt werden, mit denen die Variation durch die rein virtuelle Umgebung weiter erhöht werden kann (siehe Kap. 2).
Außerdem können im Rahmen einer *Data Augmentation* die vorhandenen Trainingsbilder durch Operationen verändert werden, die den Label-Inhalt jedoch nicht beeinflussen. Dafür werden Filter-Operationen benutzt, die Bilder verändern, ohne dabei die Pose des gesuchten Objekts abzuwandeln. Dazu gehören z.B. andere Kontrast-Werte oder das Einsetzen von Unschärfe. [18][4]

6 Fazit

In Rahmen des vorliegenden Projektes wurde eine Methode entwickelt, Trainingsdaten für eine 6DoF-Posen-Erkennung von Objekten zu erstellen. Dafür wurden Schachfiguren gestaltet, an denen 2D-Marker angebracht und wieder abgenommen werden können, ohne die Figuren dafür zu bewegen. Dies wird dafür verwendet, anhand der Marker die Posen erkennen und die Trainingsbilder trotzdem ohne Marker aufnehmen zu können. Die entwickelte Anwendung unterstützt den Benutzer dabei, mit diesem Setup die Trainings-

bilder aufzunehmen und durch die Anwendung automatisiert passende Labels für die Bilder zu generieren und somit einen vollständigen Trainings-Datensatz zu erstellen.

Literatur

[1] BLENDER: *blender Features.* – URL https://www.blender.org/features/. – Zugriffsdatum: 2023-07-20

[2] BOP: *BOP: Benchmark for 6D Object Pose Estimation.* – URL https://bop.felk.cvut.cz/datasets/. – Zugriffsdatum: 2023-07-20

[3] DROST, Bertram ; ULRICH, Markus ; NAVAB, Nassir ; ILIC, Slobodan: Model globally, match locally: Efficient and robust 3D object recognition. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, URL https://doi.org/10.1109/CVPR.2010.5540108, 2010, S. 998–1005

[4] EGGERT, Daniel: *Erweiterung eines Trainingdatensatzes für einen Objektdetektor im Kontext eines botanischen Gartens mit Methoden der Data Augmentation.* 2022. – URL https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2022-proj/eggert_hp.pdf

[5] FAN, Zhaoxin ; ZHU, Yazhi ; HE, Yulin ; SUN, Qi ; LIU, Hongyan ; HE, Jun: Deep Learning on Monocular Object Pose Detection and Tracking: A Comprehensive Overview. In: *ACM Comput. Surv.* (2022), mar. – URL https://doi.org/10.1145/3524496. – ISSN 0360-0300

[6] FORMLABS: *Form 3+.* – URL https://formlabs.com/de/3d-printers/form-3/. – Zugriffsdatum: 2023-07-20

[7] GOODFELLOW, I. ; BENGIO, Y. ; COURVILLE, A. ; SAFARI, an O'Reilly Media C.: *Deep Learning – Grundlagen, aktuelle Verfahren und Algorithmen, neue Forschungsansätze.* mitp Verlag, 2018. – ISBN 978-3958457003

[8] HANOLDAA: *Low Poly Chess Set.* – URL https://sketchfab.com/3d-models/low-poly-chess-set-0f440e2b01ca42f8b3fdee8178c51f20. – Zugriffsdatum: 2023-07-20

[9] Hintze, Nadia: *Entwicklung und Einsatzmöglichkeiten einer AR-Anwendung im Rahmen von Museumsausstellungen*. Dezember 2018. – URL https://users.informatik.haw-hamburg.de/~abo781/abschlussarbeiten/ba_hintze.pdf

[10] Hodan, Tomas ; Haluza, Pavel ; Obdrzalek, Stepan ; Matas, Jiri ; Lourakis, Manolis ; Zabulis, Xenophon: *T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects*. 2017. – URL https://doi.org/10.48550/arXiv.1701.05498

[11] Iago Casabiell González: *Ash Pawn Xogos da Meiga chess icons family*. – URL https://commons.wikimedia.org/wiki/File:Ash_Pawn_Xogos_da_Meiga_chess_icons_family.svg. – Zugriffsdatum: 2023-07-20

[12] Iago Casabiell González: *Ash Queen Xogos da Meiga chess icons family*. – URL https://commons.wikimedia.org/w/index.php?curid=120950895. – Zugriffsdatum: 2023-07-20

[13] Iago Casabiell González: *Ash Rook Xogos da Meiga chess icons family*. – URL https://commons.wikimedia.org/w/index.php?curid=120961175. – Zugriffsdatum: 2023-07-20

[14] Kaskman, Roman ; Zakharov, Sergey ; Shugurov, Ivan ; Ilic, Slobodan: *HomebrewedDB: RGB-D Dataset for 6D Pose Estimation of 3D Objects*. 2019. – URL https://doi.org/10.48550/arXiv.1904.03167

[15] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision 60*, URL https://doi.org/10.1023/B:VISI.0000029664.99615.94, 2004, S. 91–110

[16] PTC Inc.: *Vuforia Enterprise Augmented Reality (AR) Software*. – URL https://www.ptc.com/de/products/vuforia. – Zugriffsdatum: 2023-07-20

[17] Samsung: *Galaxy S9*. – URL https://www.samsung.com/de/support/model/SM-G960FZPADBT/. – Zugriffsdatum: 2023-07-20

[18] Shorten, Connor ; Khoshgoftaar, Taghi M.: A survey on Image Data Augmentation for Deep Learning. In: *Journal of Big Data*, URL https://doi.org/10.1186/s40537-019-0197-0, 01 2019

[19] Spallek, Dustin: *Deep Learning basierte Erkennung von 3D-Objektposen auf Basis synthetisch erzeugter Daten*. September 2020. – URL https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/spallek.pdf

[20] The Gimp Team: *GIMP - GNU Image Manipulation Program*. – URL https://www.gimp.org. – Zugriffsdatum: 2023-07-20

[21] 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

[22] Tremblay, Jonathan ; To, Thang ; Birchfield, Stan: Falling Things: A Synthetic Dataset for 3D Object Detection and Pose Estimation, 06 2018, S. 2119–21193

[23] Tyree, Stephen ; Tremblay, Jonathan ; To, Thang ; Cheng, Jia ; Mosier, Terry ; Smith, Jeffrey ; Birchfield, Stan: *6-DoF Pose Estimation of Household Objects for Robotic Manipulation: An Accessible Dataset and Benchmark*. 2022. – URL https://doi.org/10.48550/arXiv.2203.05701

[24] Unity Technologies: *Unity*. – URL https://unity.com/. – Zugriffsdatum: 2023-07-20