Entwicklung eines teilautomatisierten Systems zur Bestimmung von Ground-Truth Posen teilweise verdeckter Objekte

Masterarbeit
Informatik

Constantin Wellhausen

22. November 2018

Erstgutachter: Prof. Dr.-Ing. Udo Frese
Zweitgutachter: Dr. habil. Hagen Langer

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Urheberrechtliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich meine Masterarbeit „Entwicklung eines teilautomatisierten Systems zur Bestimmung von Ground-Truth Posen teilweise verdeckter Objekte“ selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe und dass ich alle Stellen, die ich wörtlich oder sinngemäß aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe. Die Arbeit hat bisher in gleicher oder ähnlicher Form oder auszugsweise noch keiner Prüfungsbehörde vorgelegen.

Bremen,

(Unterschrift)

1 Einleitung

1.1 Motivation

In der modernen Computer Vision sind umfangreiche Vergleichsdaten, anhand welcher entwickelte Algorithmen evaluiert werden können, eine wertvolle Ressource. Das Ziel in der Computer Vision ist in der Regel, aus Bildern Informationen über die aufgenommene Szene zu gewinnen. So werden beispielsweise Objekte in Bildern identifiziert und ihre Namen oder ihre Positionen bestimmt. Um bewerten zu können, ob ein Algorithmus die Objekteigenschaften korrekt bestimmt hat, sind Vergleichsdaten notwendig. Es werden also Bilder benötigt, in denen die Namen und Positionen aller Objekte bereits bekannt sind. Mithilfe der Vergleichswerte kann entschieden werden, ob der Algorithmus das korrekte Ergebnis liefert. Solche Vergleichsdatensätze werden auch Ground-Truth Datensätze genannt. Die Beschaffung solcher Datensätze ist allerdings eine große Herausforderung, da sie in der Regel für jeden einzelnen Anwendungsfall speziell erstellt werden müssen. Viele Algorithmen sind abhängig von speziellen Objekten, auf die sie trainiert sind. Diese müssen entsprechend im Datensatz genutzt werden. Auch die Umgebung, in welcher der Datensatz aufgenommen wird, sollte den echten Einsatzort möglichst gut widerspiegeln. Nur so können realistische Lichtverhältnisse und Verdeckungssituationen entstehen. Da das Erstellen solcher Datensätze sehr aufwendig ist, wird in dieser Arbeit die Frage gestellt, wie sich solche Datensätze mit möglichst geringem Zeitaufwand aufnehmen lassen.

Die in dieser Arbeit betrachteten Ground-Truth Datensätze bestehen aus dem Bild einer 3D-Kamera, in welchem alle Objekte mit ihren Posen und Namen annotiert sind. Ein Beispiel eines solchen Bildes ist in Abbildung 1.1 zu sehen.

Die Herausforderung beim Erstellen eines Datensatzes ist, dass in jedem Bild der 3D-Kamera zunächst alle Posen und Namen der enthaltenen Objekte unbekannt sind. Es stellt sich daher die Frage, wie die Posen der Objekte ausgemessen werden können und wie ihnen Namen zugeordnet werden. Da Ground-Truth Datensätze aus mehreren tausend Bildern bestehen können, ist es nicht möglich, in jedem Bild die Objekte manuell zu annotieren. Selbst wenn die Objekte nur mit Namen annotiert werden müssten, würde die manuelle Annotation bereits viel Zeit in Anspruch nehmen. Angenommen, es werden 3 000 Bilder aufgenommen, in denen durchschnittlich 25 Objekte enthalten sind, so müsste der Annotator[1] bereits 75 000 Objektnamen annotieren. Weiterhin ist anzunehmen, dass

[1] In der folgenden Arbeit wird aus Gründen der besseren Lesbarkeit ausschließlich die männliche Form

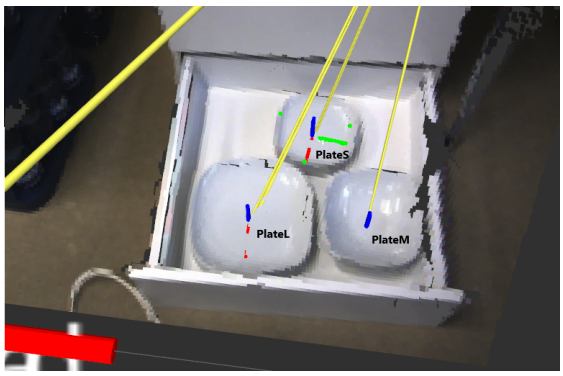

Abbildung 1.1: Ground-Truth Szene
Ein Bild einer 3D-Kamera, in welchem die Objekte annotiert sind. Rote Striche kennzeichnen die x-Achse, grüne Striche die y-Achse und blaue Striche die z-Achse der lokalen Koordinatensysteme der Objekte. Diese Achsen sind in die Punktwolke der Szene eingezeichnet, wobei die Punkte die zu den Objekten gehören teilweise die Achsen verdecken, weshalb nicht alle Achsen sichtbar sind. Gelbe Striche stellen Verbindungen zu anderen Koordinatensystemen dar und können in dieser Abbildung ignoriert werden.

die manuelle Annotation von Posen unabhängig von dem konkreten Verfahren mehr Zeit beansprucht als das Annotieren der Objektnamen. So wird deutlich, dass ein System zum Aufnehmen von Ground-Truth Daten zu einem gewissen Grad automatisiert werden muss.

1.2 Zielsetzung

In dieser Arbeit soll daher ein teilautomatisierter Ansatz entwickelt werden, welcher verschiedene Sensoren und Bildverarbeitungsalgorithmen kombiniert, um eine möglichst genaue Abschätzung der Objektposen zu erhalten. Mit diesem Ansatz soll es möglich sein, viele verschiedene annotierte Szenen aufzunehmen, wobei der Aufwand pro Aufnahme möglichst gering ausfallen soll. Der in dieser Arbeit entwickelte Ansatz soll es dem Annotierer daher ermöglichen, die Pose eines Objekts direkt beim Platzieren zu erfassen und sie zusammen mit allen anderen Objektposen in einem gemeinsamen Koordinatensystem zu speichern. Dies soll einen effizienten Arbeitsfluss ermöglichen, in welchem die Szene nach und nach mit Objekten gefüllt wird, ohne in jedem aufgenommenen Bild alle Objekte neu erfassen zu müssen. So ist die Anzahl der benötigten Annotationsvorgänge unabhängig von der Anzahl der aufgenommenen Bilder. Werden 25 Objekte in der Szene platziert, so müssen entsprechend alle 25 Objekte einmalig annotiert werden. Das hierfür benötigte gemeinsame Koordinatensystem soll von einem Infrarot-Tracking System bereitgestellt

verwendet. Sie bezieht sich auf Personen beiderlei Geschlechts.

werden. Dieses System soll es zusätzlich ermöglichen, die Pose der 3D-Kamera zu erfassen. Solange die Posen der Objekte und die Posen der 3D-Kamera von dem Infrarot-Tracking System erfasst werden, lassen sich daraus automatisch die Posen der Objekte im Bild der 3D-Kamera berechnen.

Während das System einen möglichst effizienten Arbeitsfluss ermöglichen soll, spielt die Genauigkeit dieses Verfahrens eine so wichtige Rolle, dass alle vollautomatischen Verfahren zur Annotation der Objektposen ausgeschlossen sind. Solche Algorithmen sind häufig unzuverlässig in ihrer Erkennungsrate und bestimmen beispielsweise die Position von Objekten lediglich auf einige Zentimeter genau [6]. Für einen Ground-Truth Datensatz werden Posen benötigt, welche eine Größenordnung genauer sind als die Algorithmen, die mit ihm evaluiert werden sollen. Die Posenbestimmung, welche in dieser Arbeit entwickelt wird, muss entsprechend Ergebnisse liefern, deren Abweichungen im Millimeterbereich liegen. Die Genauigkeit des in dieser Arbeit entwickelten Systems soll entsprechend untersucht werden. Dabei wird die Frage gestellt, ob die entwickelten Verfahren die gewünschte Genauigkeit erreichen können und somit zur Evaluierung von Posenerkennungsalgorithmen geeignet sind.

Als Produkt dieser Arbeit soll mit der Aufnahme eines Datensatzes begonnen werden. Dabei geht es vorrangig darum, die Einsetzbarkeit des entwickelten Ground-Truth Systems zu demonstrieren. Die vollständige Aufnahme des Datensatzes ist dementsprechend kein Ziel dieser Arbeit. Um trotzdem einen vollständigen Datensatz zu erhalten, soll die Aufnahme unabhängig von dieser Arbeit fortgeführt werden. Der Datensatz soll dabei Küchenobjekte enthalten, um als Testdatensatz für Haushaltsroboter zu dienen. Aufgenommen wird er entsprechend in einer Küchenumgebung. Diese soll im Umfang dieser Arbeit aufgebaut und mit den Objekten für den Datensatz ausgestattet werden.

1.3 Anforderungen an das Ground-Truth System

Das in dieser Arbeit entwickelte Ground-Truth System muss es ermöglichen, Objekte in einer Szene mit ihren Namen und Posen zu annotieren. Die Posen müssen dabei mit einer bekannten Genauigkeit bestimmt werden, sodass der Nutzer des Datensatzes weiß, welche Abweichungen in den Daten zu erwarten sind. Zu beachten ist, dass in dieser Arbeit zwischen zwei verschiedenen Nutzern unterschieden wird: Der Nutzer des Datensatzes, welcher die aufgenommenen Ground-Truth Daten verwendet und der Nutzer, welcher das Ground-Truth System zum Aufnehmen von Daten einsetzt. Um in dieser Arbeit zwischen diesen Rollen abzugrenzen, wird der Nutzer des Ground-Truth Systems als Annotator bezeichnet. Der Begriff Nutzer beschreibt lediglich den Nutzer des Datensatzes.

In dieser Arbeit wird bewusst darauf verzichtet, zu bewerten, ob das System in jedem Fall eine ausreichende Genauigkeit erzielt, um Ground-Truth Daten zu erzeugen. Eine solche Aussage muss für jeden Anwendungsfall individuell getroffen werden. Stattdessen werden Evaluationsmetriken aus der aktuellen Literatur zur Posenbestimmung von Objek-

ten herausgearbeitet. Zu jeder der Metriken wird angegeben, ob das Ground-Truth System ausreichende Genauigkeiten erzielt hat, um der Evaluation von Algorithmen zur Posenbestimmung zu dienen. So werden dem Nutzer bereits Metriken vorgeschlagen, welche er zur Evaluation nutzen kann. Er kann aber auch anhand der angegebenen Genauigkeiten selbst entscheiden, ob sich dieses Ground-Truth System für den eigenen Anwendungsfall eignet. Die Genauigkeiten der Objektpose werden dabei in der Abweichung der Rotation und Translation des Objekts angegeben.

Weiterhin muss das Ground-Truth System ermöglichen, auch in Szenen mit starken Verdeckungseffekten Objektposen aufzunehmen. Schubladen und Schränke in der Küche zeichnen sich typischerweise dadurch aus, dass viele, häufig verschiedene Objekte nah beieinander oder aufeinander liegen. So entstehen Verdeckungseffekte, welche das Aufnehmen von Ground-Truth Objektposen erschweren. In dieser Arbeit werden vier Verfahren entwickelt, welche Objektposen auf verschiedene Arten aufnehmen. Diese Verfahren gehen unterschiedlich mit Verdeckungseffekten um und sind daher in verschiedenen Situationen einsetzbar. Sie basieren alle darauf, die Pose eines Objekts anhand von bekannten Punkten auf dem Objekt aufzunehmen. Wie diese Punkte aufgenommen werden, welche Punkte in Frage kommen und wie aus ihnen eine Objektpose berechnet werden kann, hängt von dem Verfahren ab. So entsteht ein System, mit welchem Objektposen in nahezu allen Konfigurationen aufgenommen werden können.

1.4 Beitrag dieser Arbeit

Im Verlauf dieser Arbeit wurden die folgenden Beiträge erfolgreich umgesetzt:

- Entwicklung eines Systems zur praktikablen Aufnahme von Küchenobjekten inklusive der Ground-Truth Posen
- Dazu Entwicklung und Evaluation zwei verschiedener Verfahren zum Berechnen von Objektposen
- Dazu Entwicklung und Evaluation vier verschiedener Verfahren zum Aufnehmen von Punkten auf Objekten
- Dazu Entwicklung und Evaluation verschiedener Kalibrierungsverfahren zur Nutzung aller Sensoren zum Aufnehmen von Punkten
- Entwicklung einer Smartphone-Anwendung zur Steuerung des Ground-Truth Systems
- Aufnehmen der ersten neun Küchenszenen eines Datensatzes mit verschiedenen Küchenobjekten

1.5 Aufbau

Die Arbeit ist in neun Kapitel geteilt. Im zweiten Kapitel wird der aktuelle Stand der Technik im Bereich Ground-Truth Systeme vorgestellt. Es werden einige existierende Systeme vorgestellt und auf ihre Anwendbarkeit in dieser Arbeit geprüft. Das dritte Kapitel liefert einige Grundlagen, welche in der Arbeit in vielen Bereichen genutzt werden. Es werden Transformationen, Kamerakalibrierungen, Non-linear Least-Square Probleme und Voraussetzungen zur Objektposenbestimmung vorgestellt. Das vierte Kapitel gibt einen Überblick über das Ground-Truth System. Dabei wird vorgestellt, wie das System aufgebaut ist, welche Sensoren verwendet werden und wie mit ihrer Hilfe Objektposen aufgenommen werden. Im fünften Kapitel werden verschiedene Kalibrierungsverfahren dargestellt, die in dieser Arbeit benötigt werden. Das sechste Kapitel beschäftigt sich mit der Frage, wie Punkte auf den Objekten aufgenommen werden können, aus welchen sich später Objektposen berechnen lassen. Hierzu werden vier Verfahren vorgestellt und auf ihre Genauigkeit geprüft. Im siebten Kapitel wird beschrieben, wie die Punkte auf den Objekten zu einer Objektpose kombiniert werden können. Anschließend wird auch hier die Genauigkeit der entwickelten Verfahren evaluiert. Das achte Kapitel stellt den in dieser Arbeit aufgenommenen ersten Teil des Datensatzes vor. Dabei werden angewendete Methodiken aufgezeigt und es wird ein Überblick über die Anwendbarkeit der entwickelten Verfahren in verschiedenen Situationen gegeben. Im letzten Kapitel wird ein abschließendes Fazit zu dieser Arbeit gezogen. Dies umfasst auch mögliche Verbesserungen, mit welchen diese Arbeit in Zukunft fortgeführt werden könnte.

2 Stand der Technik

In diesem Kapitel werden existierende Verfahren zur Ground-Truth Posenbestimmung vorgestellt. Zu den Verfahren wird dabei jeweils ihre Anwendbarkeit in einer Küchenumgebung bewertet.

Während es noch keine Ground-Truth Systeme gibt, die eine Lösung zum Aufnehmen aller verschiedenen Objektarten bereitstellt, gibt es bereits einige vielversprechende Ansätze, die das Aufnehmen von Ground-Truth Daten auf unterschiedliche Arten angehen. In der Arbeit von Godil et al.[12] werden die existierenden Ansätze in vier Gruppen unterteilt:

- Annotations-/Kennzeichnungs-basierte Systeme
- Sensorisch wahrnehmende Ground-Truth Systeme
- Plattform-basierte Systeme
- Simulations-basierte Systeme

Die simulations-basierten Systeme sind dabei für diese Arbeit nicht relevant, da sie nur simulierte Bilder erzeugen können, was nicht im Interesse dieser Arbeit ist. In den anderen drei Kategorien existieren bereits Lösungen, deren Herangehensweise durchaus wichtig für das Aufnehmen von Ground-Truth Daten in einer Küchenumgebung sind. Im Folgenden werden einige Beispiele aus diesen Kategorien vorgestellt und auf ihre Anwendbarkeit geprüft.

2.1 Annotations-/Kennzeichnungs-basierte Systeme

Das System Viper-GT wurde 2003 von Mihalcik und Doermann vorgestellt[19]. Es arbeitet auf Videos und erlaubt es dem Annotator, die gewünschten Objekte mit einer Bounding-Box zu annotieren, also einer Box, welche das Objekt im Bild einschließt. Dies ist in Abbildung 2.1 zu sehen. So lassen sich die Videos Bild für Bild per Hand annotieren. Dabei entstehen allerdings lediglich Regionen, in denen sich ein Objekt befindet. Dadurch kann zwar abgeschätzt werden, in welcher Richtung das Objekt zu verorten ist, unbekannt bleibt aber, wie weit dieses Objekt von der Kamera entfernt ist. Für die Annotation von 3D-Posen, die in dieser Arbeit benötigt werden, sind solche Tiefeninformationen auch erforderlich.

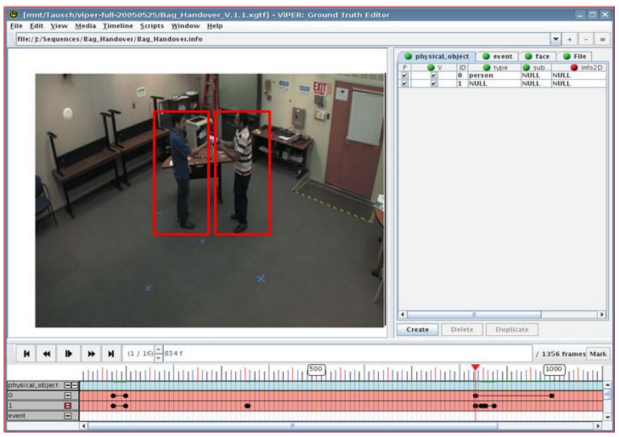

Abbildung 2.1: Ground-Truth Daten in ViperGT
Eine annotierte Szene in ViperGT. Die roten Markierungen stellen die Bounding-Boxen dar, in welchen sich Objekte, in diesem Fall Personen, befinden[12].

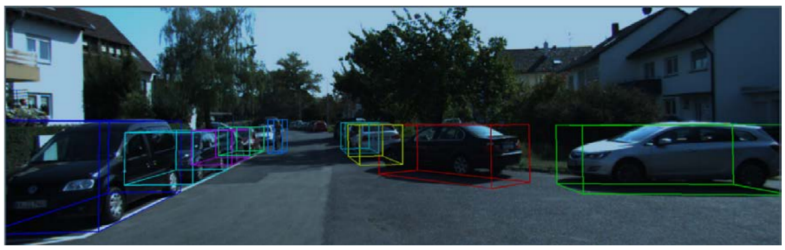

Abbildung 2.2: Ground-Truth Daten in KITTI
Eine annotierte Szene in KITTI. Die 3D-Bounding-Boxen sind farbig markiert. Sie schließen die zu erkennenden Objekte ein (Leicht modifiziert aus der Arbeit von Geiger et al. [11]).

Hierzu wurde von Geiger et al. 2012 die KITTI Vision Benchmark Suite entwickelt[10], welche zusätzlich zu RGB-Bildern auch noch auf Daten von Laserscannern arbeitet. Diese Daten werden von dem Toolkit als 3D-Szene visualisiert, in welcher der Annotator die Objekte mit 3D-Bounding-Boxen annotieren kann. Hierzu muss die Bounding-Box manuell korrekt platziert und auf die richtige Größe skaliert werden, wie in Abbildung 2.2 zu sehen ist. Damit liefert das Tool zunächst einmal alle benötigten Hilfsmittel, um 3D-Szenen zu annotieren. Die Genauigkeit dieses Systems hängt aber stark von dem Annotator ab, sowie von der Zeit, die in die Annotation investiert wird. Zusätzlich wird die Qualität der Posen in diesem System von der Qualität der Tiefeninformationen beeinflusst. In der ursprünglichen Arbeit von Geiger et al. werden Laserscanner genutzt, um diese Informationen zu erhalten, was ein relativ genaues Ergebnis verspricht. Die in dieser

Arbeit eingesetzten 3D-Kameras liefern relativ zur Szenengröße im Vergleich deutlich ungenauere Tiefeninformationen. Dadurch ist es insbesondere bei kleinen Objekten wie Teelöffeln schwierig, eine hinreichend genaue Bounding-Box zu platzieren.

Annotations-basierte Systeme sind sehr intuitiv in ihrer Anwendung. Sie ermöglichen es, Bild für Bild einen Datensatz zu annotieren mit nur wenig Arbeitsschritten pro Objekt. Insbesondere in Arbeitsumgebungen wie der Küche werden die Nachteile dieser Systeme jedoch deutlich. Da die Objekte teils sehr klein sind, müsste die 3D-Kamera sehr präzise Tiefeninformationen liefern, um beispielsweise den zwei Millimeter dicken Griff eines Löffels korrekt erfassen zu können. Ein weiteres Problem ist, dass die Genauigkeit der Annotationen nicht durch das System vorgegeben ist. Sie hängt vom Annotator sowie von der Präzision der Tiefendaten der Sensoren ab. Zusätzlich ist die Annotation jedes Bildes bei einem Datensatz von mehreren tausend Bildern ein großer Zeitaufwand. Aus diesen Gründen sind annotations-basierte Systeme für diese Arbeit nicht anwendbar.

2.2 Sensorisch wahrnehmende Systeme

Im Gegensatz zu annotations-basierten Systemen, welche sich dadurch auszeichen, dass die Bilder per Hand von Menschen annotiert werden, verwenden sensorisch wahrnehmende Systeme verschiedene Sensoren, um die Objekte selbst oder Markierungen an den Objekten wahrzunehmen und automatisch zu annotieren. Diese Sensoren ermöglichen meist präzise Messungen. Beispiele hierfür sind Indoor GPS Systeme und Kamerasysteme.

Ein Beispiel für ein Indoor GPS System ist das Nikon iGPS System[21]. Dieses nutzt zwei oder mehr Transmitter, welche jeweils Infrarotsignale in einem Fächer um sich selbst herum aussenden. Ein Sensor in der Mitte wird von diesen Signalen getroffen. Treffen ihn zwei Strahlen von unterschiedlichen Transmittern, so ist der Schnittpunkt dieser Strahlen die Position des Sensors. Auf diese Art lassen sich laut Hersteller Genauigkeiten von 200 Mikrometern erzielen. Die Funktionsweise ist in Abbildung 2.3 visualisiert.

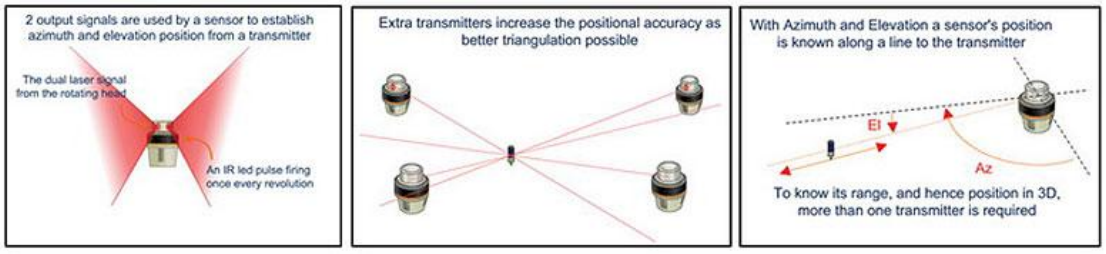

Abbildung 2.3: Funktionsweise des iGPS Systems
Transmitter senden Infrarotstrahlen aus, um die Position des Sensors entlang einer Geraden zu bestimmen. Es werden mindestens 2 Transmitter benötigt, um die Position zu bestimmen, mehrere Transmitter verbessern die Genauigkeit [21].

Während dieses System eine sehr genaue und vollautomatische Positionsbestimmung bietet, ist die Anwendbarkeit stark dadurch eingeschränkt, dass ein iGPS Sensor für die Lokalisierung benötigt wird. Dieser ist zu groß, um ihn an kleinen Küchenobjekten anzubringen. Auch liefert dieses System zunächst einmal nicht die benötigten Posen, sondern nur die Position des Sensors. Demnach würden mehrere iGPS Sensoren an einem Objekt benötigt, um eine vollständige Pose zu erhalten.

Das System TRACKPACK der Firma ART bietet eine ähnliche Lösung, welche mit zwei oder mehr Infrarot-Kameras den Arbeitsraum überwacht[2]. Die Infrarot-Kameras beleuchten den Raum mit Infrarotlicht, welches von extra angefertigten reflektierenden Markern zurückgeworfen wird. Diese Reflexion wird von den Infrarot-Kameras aufgenommen. Wird ein Marker von mehreren Kameras gesehen, so ergeben sich wieder mehrere sich schneidende Geraden, wie bereits im iGPS System. Dieser Schnittpunkt ist die entsprechende 3D-Position des Markers. Dieses Prinzip ist nicht abhängig von einem Sensor im Arbeitsbereich. Es werden lediglich reflektierende Oberflächen benötigt, welche für mindestens zwei am Rand des Arbeitsbereichs aufgebaute Kameras sichtbar sind.

Hierzu bietet die Firma ART verschiedene vorgefertigte Lösungen an. Für Annotatoren, die lediglich an Positionsdaten interessiert sind, wird eine Folie angeboten, welche beliebig zugeschnitten werden kann. Auf der Rückseite ist die Folie mit einem Klebestreifen versehen, wodurch sie an beliebigen Objekten angebracht werden kann. Die Folie hat eine besondere Oberfläche, welche auch dann noch Licht zurück in die Kamera reflektiert, wenn sie leicht geneigt zur Kamera ist. Ist die Folie allerdings zu weit geneigt, wird kein Licht mehr zur Kamera reflektiert und das System kann die Folie nicht mehr wahrnehmen.

Soll eine Position unabhängig von der Neigung zur Kamera bestimmt werden, so kann auf kugelförmige Marker zurückgegriffen werden. Diese sind mit der gleichen Folie beklebt, haben aber dadurch, dass sie kugelförmig sind, immer einen Bereich, welcher der Kamera zugewandt ist.

Diese kugelförmigen Marker lassen sich kombinieren, um mit ihnen vollständige Objektposen zu bestimmen. Die Firma ART bietet Halterungen, auch genannt Targets, an, die fünf kugelförmige Marker in einer festen Konfiguration fixieren. Ein solches Target is in Abbildung 2.4 zu sehen. Aus den fünf Markerpositionen wird von dem System automatisch die Pose des Targets berechnet.

Insgesamt liefert das System TRACKPACK von ART einige Funktionen, welche zum Aufnehmen von Objektposen nützlich sind. Aufklebbare Marker könnten beispielsweise Punkte auf Objekten liefern, welche zur Berechnung der Objektposen benötigt werden. Dies kann allerdings nur funktionieren, solange die Objekte nicht zu stark geneigt sind. Die Targets selbst sind zwar zu groß, um an den Objekten befestigt zu werden, sie können aber beispielsweise an den 3D-Kameras befestigt werden, um deren Pose im Raum zu bestimmen. Das System liefert somit einige Lösungen, es zeigt allerdings auch einige Probleme auf, welche gelöst werden müssen, bevor mit ihm vollständige Küchenszenen aufgenommen werden können.

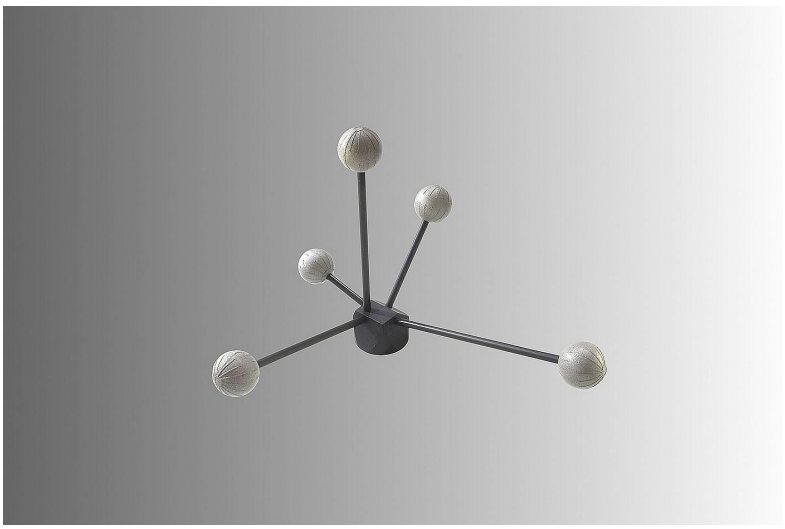

Abbildung 2.4: Tracking Target
Ein Tracking Target mit fünf reflektierenden Kugelmarkern[4].

2.3 Plattform-basierte Systeme

Bei Plattform-basierten Systemen handelt es sich um solche, bei denen Objekte in für sie vorgesehene Plattformen oder Schablonen hineingelegt werden. In der Arbeit von Marvel et al.[18] werden drei Systeme vorgestellt, welche am National Institute of Standards and Technology, kurz NIST, eingesetzt werden. Im Bild 2.5 sind diese Systeme abgebildet.

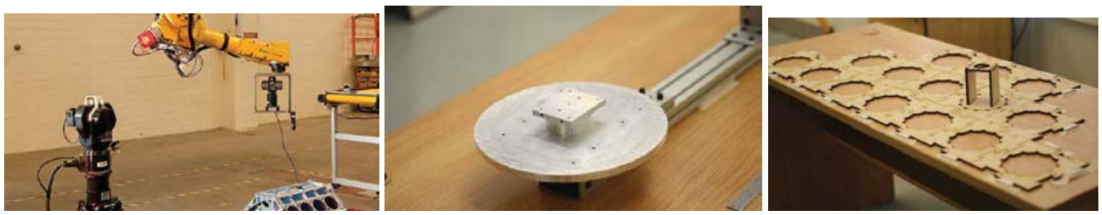

Abbildung 2.5: Plattform-basierte Systeme
Von links nach rechts: Ein Laserscanner, welcher ein am Roboterarm befestigtes Objekt erfasst, eine Aluminiumplattform und eine Zusammenstellung von Hartfaserzuschnitten[18].

Das erste System, welches in der Arbeit von Marvel et al. als Laser-Tracker System bezeichnet wird, nutzt einen Laserscanner, um ein Objekt zu verfolgen, welches an einem industriellen Roboterarm angebracht ist. Hierzu wird ein Tracking-Target am Objekt angebracht, welches dann an dem Roboterarm befestigt wird. Daraufhin kann

sich der Roboterarm beliebig bewegen, während die Pose kontinuierlich erfasst wird. Das zweite System mit dem Namen GT2011 nutzt eine mechanische Aluminiumhalterung, auf welcher an festen Punkten Objekte angebracht werden können. Diese Plattform lässt sich rotieren, und es lassen sich bis zu vier Objekte gleichzeitig anbringen, um verschiedene Objektkonfigurationen und Überlappungen aufnehmen zu können. Das letzte vorgestellte System, GT2012, wird aus beliebig vielen Hartfaserplatten zusammengestellt, welche von einem Laserschneider zugeschnitten werden können. In diesen Platten sind Fassungen zur Platzierung der Objekte eingelassen. Diese können in 15 Grad Intervallen per Hand gedreht werden. Die vielen verschiedenen Platten erlauben es dabei, eine Vielzahl an verschiedenen Objekten gleichzeitig zu erfassen.

Die hier vorgestellten Plattform-basierten Systeme zeichnen sich alle durch ihre genauen Posenbestimmungen aus. Das Laser-Tracker System erreicht Genauigkeiten von $\frac{1}{100}$ Millimetern und $\frac{3}{100}$ Grad. Die anderen beiden Systeme bewegen sich in Bereichen zwischen $\frac{1}{2}$ bis $\frac{2}{3}$ Millimeter Positionsgenauigkeit und $\frac{1}{6}$ Grad Rotationsgenauigkeit. Problematisch ist die Nutzbarkeit von plattform-basierten Systemen. Während sie für wenige Objekte sehr gute Resultate erzielen, ist es nicht möglich, Szenen mit vielen Objekten zu konstruieren. Das System GT2012 ermöglicht es zwar theoretisch, beliebig viele Objekte aufzunehmen, Szenen mit starken Überlappungen der Objekte sind damit allerdings durch den festen Abstand der Fassungen nicht möglich. Trotzdem liefern plattform-basierte Systeme einige interessante Ideen, welche sich auch in einer Küchenumgebung anwenden lassen.

2.4 Simulations-basierte Systeme

Die Nutzung simulations-basierter Ground-Truth Systeme zum Aufnehmen von Testdaten ist ein aktueller Forschungsgegenstand. In dieser Arbeit werden diese Systeme allerdings außer Acht gelassen, da weiterhin ein Unterschied zwischen simulierten und echten Daten besteht, welcher einen Einfluss auf die Nutzbarkeit der aufgenommenen Daten haben könnte. In dieser Arbeit wird daher nur auf echten Szenen gearbeitet.

3 Grundlagen

Bevor im folgenden Kapitel vorgestellt wird, wie das in dieser Arbeit entwickelte Ground-Truth System aufgebaut ist, werden in diesem Kapitel zunächst einige Grundlagen gelegt. Es werden vier Konzepte erläutert, welche in dieser Arbeit genutzt werden: Transformationen, Kamerakalibrierungen, Non-linear Least-Square Probleme und die Bestimmung von Objektposen.

3.1 Transformationen

Im Zentrum dieser Arbeit steht die Frage, wie es möglich ist, die Pose eines Objekts in dem Koordinatensystem der 3D-Kamera zu bestimmen. Die Antwort auf diese Frage wird eine Kette verschiedener Transformationen sein, welche entweder mit Sensoren gemessen oder ausgerechnet werden müssen. Es gibt allerdings eine Vielzahl an Möglichkeiten, Transformationen auszudrücken. Dieses Kapitel wird eine kurze Einführung in Transformationen geben. Anschließend werden die wichtigsten Arten vorgestellt, Transformationen auszudrücken und verwendete Notationen eingeführt.

Transformationen definieren eine Abbildung von einem Koordinatensystem in ein anderes. Die Transformation enthält dabei die Informationen, wie die zwei Koordinatensysteme zueinander gedreht und verschoben sind. So lassen sich Punkte, Vektoren oder Posen, die in dem einen Koordinatensystem ausgedrückt sind, durch Anwendung der Transformation in das andere umrechnen. Für diese Arbeit bedeutet das, dass Punkte, die im Koordinatensystem des Smartphones (SM) bekannt sind, beispielsweise auch in das Koordinatensystem der 3D-Kamera (C) umgerechnet werden können, solange die Transformation zwischen diesen beiden Koordinatensystemen bekannt ist. Dabei wird ausgenutzt, dass Transformationen verkettet werden können. Ist beispielsweise ein Punkt $p^{(SM)}$ im Smartphone-Koordinatensystem bekannt sowie zwei Transformationen $T_{W \leftarrow SM}$ vom Smartphone-Koordinatensystem in ein Weltkoordinatensystem (W) und $T_{C \leftarrow W}$ von dem Weltkoordinatensystem in die 3D-Kamera, so lässt sich $p^{(SM)}$ durch Anwendung der ersten Transformation in das Weltkoordinatensystem transformieren. Von dort kann der Punkt durch Anwendung der zweiten Transformation weiter in das Koordinatensystem der 3D-Kamera umgerechnet werden. So lässt sich das Problem, Objektposen in der 3D-Kamera aufzunehmen, auch in anderen Sensoren lösen, solange zwischen diesen Sensoren und der 3D-Kamera eine bekannte Transformationskette existiert.

Eine intuitive Art, sich Transformationen vorzustellen, ist die Betrachtung ihrer Verschiebung sowie der Drehung um die x-,y- und z-Achse. Die Verschiebung, im Folgenden

auch Translation genannt, lässt sich als dreidimensionaler Vektor v wie folgt darstellen:

$$v = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Dabei stellen die Komponenten x, y, z die Verschiebung entlang der gleichnamigen Achsen dar. Die Drehung um die drei Achsen wird anhand der Winkel beschrieben. Diese Art, Rotationen darzustellen, nennt sich Euler-Winkel. Während Euler-Winkel intuitiv leicht zu verstehen sind, haben sie einige grundlegende Probleme, die die Arbeit mit ihnen erschweren. Die Reihenfolge, in welcher die Rotationen durchgeführt werden, spielt bei Euler-Winkeln eine große Rolle. Abhängig davon, ob zunächst um die x- oder um die z-Achse gedreht wird, fällt das Ergebnis in den meisten Fällen unterschiedlich aus. Weiterhin lässt sich mit Euler-Winkeln nicht leicht rechnen. Soll beispielsweise die Transformation auf einen Vektor angewendet werden, so müsste die Transformation zunächst in andere Repräsentationen konvertiert werden. Aus diesen Gründen werden für Berechnungen in dieser Arbeit keine Euler-Winkel verwendet. Sie werden dennoch in den folgenden Kapiteln genutzt, um einige Rotationen für den Leser besser nachvollziehbar zu machen. In diesem Fall ist die Konvention, dass zuerst um die x-Achse rotiert wird, dann um die y-Achse und zuletzt um die z-Achse.

Eine Repräsentation, mit welcher sich gut rechnen lässt, ist die Rotationsmatrix. Rotationsmatrizen im dreidimensionalen Raum sind 3×3 Matrizen, welche multipliziert mit einem Vektor p einen neuen Vektor p' produzieren. Dieser ist um die in der Rotationsmatrix definierte Rotation gedreht. Rotationsmatrizen definieren somit auch die Abbildung eines Koordinatensystems A in ein zu A rotiertes Koordinatensystem B. Anders ausgedrückt kann ein Richtungsvektor $p^{(A)}$, also ein Vektor der lediglich eine Richtung in dem Koordinatensystem beschreibt, vom Koordinatensystem A in das Koordinatensystem B wie folgt transformiert werden:

$$p^{(B)} = R_{B \leftarrow A} \cdot p^{(A)}$$

Dabei ist $R_{B \leftarrow A}$ die Rotationsmatrix, welche die Rotation vom Koordinatensystem A in das Koordinatensystem B definiert. Der resultierende Vektor $p^{(B)}$ ist derselbe Vektor wie $p^{(A)}$, nun aber ausgedrückt im Koordinatensystem B. Für Ortsvektoren, also Vektoren vom Ursprung zu einem Punkt im Koordinatensystem, ist die Rechnung ähnlich. Hier muss allerdings berücksichtigt werden, dass die Koordinatensysteme auch um einen Vektor t zueinander verschoben sein können, was die folgende Gleichung ergibt:

$$p^{(B)} = R_{B \leftarrow A} \cdot p^{(A)} + t$$

Bislang wurde in der eigentlichen Repräsentation lediglich die Rotation berücksichtigt, die Translation wurde extra behandelt. Es ist allerdings auch möglich, sowohl die

Rotation als auch die Translation in einer einzigen 4×4 Matrix auszudrücken, indem Gebrauch von affinen Transformationen gemacht wird. Affine Transformationen haben einige Einschränkungen im Vergleich zu allgemeinen Transformationen. Bei der Anwendung einer affinen Transformation bleiben Punkte, gerade Linien und Ebenen erhalten. Diese Einschränkungen sind durchaus sinnvoll für die Transformation von Objekten, da das Objekt vor und nach der Transformation die gleichen Proportionen haben muss. Es sind allerdings auch Transformationen wie Skalierungen möglich, welche die Größe des Objekts ändern. Solche Operationen dürfen entsprechend nicht verwendet werden, da sie die Objekteigenschaften verändern. Eine affine Transformationsmatrix ist wie folgt aufgebaut:

$$T = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Dabei ist R die zuvor vorgestellte Rotationsmatrix, t ist die Translation. Die letzte Zeile der Matrix ist für jede in dieser Arbeit genutzte Transformation dieselbe. Alle Spalten, die Teile der Rotationsmatrix enthalten, sind in der letzten Zeile mit einer 0 markiert, die Spalte mit der Translation ist mit einer 1 gekennzeichnet.

Mit dieser Matrix lässt sich nun genauso rechnen wie mit der Rotationsmatrix zuvor. Der einzige Unterschied ist, dass nun alle Koordinaten homogen sein müssen. Demnach werden sie um einen vierten Wert wie folgt erweitert:

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \rightarrow \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ 0/1 \end{pmatrix}$$

Dabei wird der Vektor um eine 0 erweitert, wenn es sich um einen Richtungsvektor handelt. Eine 1 markiert einen Ortsvektor. Definiert die Matrix $T_{C \leftarrow A}$ beispielsweise eine Abbildung vom Koordinatensystem A in das Koordinatensystem C, so lässt sich eine Transformation mit ihr folgendermaßen ausdrücken:

$$p^{(C)} = T_{C \leftarrow A} \cdot p^{(A)}$$

Dabei ist $T_{C \leftarrow A}$ eine 4×4 Matrix und $p^{(A)}$ ein Punkt, welcher wie oben beschrieben erweitert wurde. Diese Repräsentation von Transformationen ist eindeutig und kann direkt per Matrix-Vektor-Multiplikation angewendet werden. In dieser Arbeit wird intern in allen Implementierungen, soweit möglich, diese Repräsentation von Transformationen verwendet.

In einigen Fällen wird es nötig sein, eine alternative Repräsentation von Transformationen zu nutzen, welche die Rotation mit nur vier Zahlen ausdrückt statt wie in der Rotationsmatrix mit neun. Diese Repräsentation nutzt Einheitsquaternione zum Ausdrücken von Rotationen. Quaternione bestehen aus vier Skalaren w, x, y, z. Ein Quaternion wird dabei durch die Gleichung $q = w + xi + yj + zk$ ausgedrückt, wobei i, j, k imaginäre Zahlen sind. Soll wie zuvor ein Richtungsvektor p vom Koordinatensystem A in das rotierte Koordinatensystem B transformiert werden, so lässt sich dies mit Quaternionen wie folgt ausdrücken, wobei :

$$p^{(B)} = q_{B \leftarrow A} \cdot p^{(A)} \cdot \overline{q_{B \leftarrow A}}$$

Dabei ist $\overline{q} = w - xi - yj - zk$. Diese Art, Rotationen darzustellen, ist deutlich kompakter als eine Rotationsmatrix und ist auch in der Berechnung effizienter. Es gibt allerdings keine Möglichkeit, die Translation in dieser Repräsentation unterzubringen. Um Ortsvektoren zu transformieren, müsste diese wie bereits bei den Rotationsmatrizen nach der Rotation noch auf das Ergebnis angewendet werden. Aus diesem Grund sind Quaternione in der Implementierung etwas aufwendiger zu verwalten. Sie werden in dieser Arbeit nur dort genutzt, wo ihre Anwendung notwendig ist.

3.2 Kamerakalibrierung in OpenCV

Das Kalibrieren der genutzten Kamera ist ein essentieller Schritt in den meisten Anwendungen, die mit Kamerabildern arbeiten. Ist die Kamera kalibriert, so ist es möglich, zwischen Pixeln im Bild und einfallenden Lichtstrahlen im Raum zu konvertieren. Hierzu werden die Eigenschaften der Kamera möglichst genau erfasst. Zu diesen Eigenschaften zählen Verzerrungseffekte, Bildweite und Bildmittelpunkt der Kamera. Diese Informationen sind notwendig, um aus dem verzerrten Bild, das die Kamera liefert, ein entzerrtes Bild berechnen zu können, in welchem die Parameter der Kamera bekannt sind. Diese Entzerrung ist in der Regel der erste Schritt in der Bildverarbeitung, damit alle weiteren Schritte auf diesem korrigierten Bild arbeiten können. Ist das Bild entzerrt, so können die Bildweite und der Bildmittelpunkt genutzt werden, um zwischen Informationen im Bild und dreidimensionalen Informationen im Kamerakoordinatensystem zu transformieren. Diese Operation wird beispielsweise in der Markererkennung mit dem Smartphone in Kapitel 6 eine wichtige Rolle spielen.

In dieser Arbeit werden einige Kamerasysteme verwendet, die kalibriert werden müssen. Hierzu werden verschiedene Verfahren verwendet, welche alle auf die in der Bibliothek OpenCV implementierten Algorithmen zur Kamerakalibrierung zurückgreifen [22]. OpenCV ist eine Bibliothek, in welcher eine Vielzahl an Bildverarbeitungsalgorithmen implementiert sind. In diesem Abschnitt wird beschrieben, wie das Kalibrierungsverfahren in OpenCV umgesetzt ist. In allen beschriebenen Modellen wird dabei von einem sogenannten Lochkameramodell ausgegangen. Dies ist ein Kameramodell, bei welchem

das Licht nur durch einen Punkt einfallen kann, im Gegensatz zu echten Kameras, bei welchen das Licht durch eine Linse fokussiert wird. Solche ideal punktförmigen Öffnungen existieren in der Realität nicht, es ist aber eine nötige Verallgemeinerung, um die Komplexität der Kalibrierung zu reduzieren. In dem Lochkameramodell gibt es allerdings dementsprechend keine Verzerrungseffekte, weshalb das Modell in OpenCV entsprechend erweitert wurde.

Insgesamt wird die Verzerrung anhand der radialen und tangentialen Verzerrung modelliert. Radiale Verzerrung ist eine Folge der eingesetzten Linsen. Um einen möglichst weiten Winkel mit der Kamera erfassen zu können, müssen die einfallenden Lichtstrahlen am Rand der Linse stark gebrochen werden. Dadurch kann das Licht noch auf den häufig sehr kleinen Bildsensor geleitet werden. Viele moderne Kameras arbeiten mit Sensoren die elf Millimeter oder kleiner in der Diagonale sind. So entsteht der sogenannte Barrel- oder Fish-eye-Effekt. Tangentiale Verzerrung entsteht hingegen, wenn die Linse nicht parallel zum Bildsensor liegt. Dies kann in der Fertigung der Kamera passieren. Diese Verzerrungseffekte werden in OpenCV anhand der folgenden Parameter kalibriert:

$$Distortion_{coefficients} = (k_1, k_2, p_1, p_2, k_3)$$

Sind diese Parameter bekannt, so lassen sich Punkte, die im Kamerakoordinatensystem bekannt sind, neu berechnen, sodass sich für jeden Punkt eine korrigierte Position (u', v') ergibt, an welcher sich der Punkt in einer Kamera mit dem entsprechenden Verzerrungsmodell befinden würde. Hierzu wird für einen Eingabepunkt $p^{(C)}$ im Kamerakoordinatensystem mit Koordinaten x, y, z zunächst eine Perspektivprojektion wie folgt durchgeführt:

$$perspProj(p^{(C)}) = \begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \end{pmatrix}$$

Auf diesem projizierten Punkt wird anschließend wie folgt das Verzerrungsmodell angewendet:

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = distort\left(perspProj(p^{(C)}), Distortion_{coef}\right)$$

Dabei ist *distort* die Funktion, welche die neue Position des Punktes berechnet, abhängig von dem gegebenen projizierten Punkt sowie den Verzerrungsparametern. Diese Funktion ist in OpenCV implementiert, die konkrete Umsetzung ist für diese Arbeit nicht relevant und wird daher nicht behandelt.

Die korrigierten Positionen können anschließend genutzt werden, um zwischen Punkten in Kamerakoordinaten und Pixelkoordinaten zu konvertieren. Dies wird zum einen für die Kalibrierung der Bildweite f_x, f_y und des Bildmittelpunktes c_x, c_y benötigt, die Konvertierung wird zum anderen im Verlauf der Arbeit viele weitere Anwendungen

finden, weshalb im Folgenden einige Notationen eingeführt werden. Die Bildweite und der Bildmittelpunkt werden dabei wie folgt angegeben:

$$Mapping_{coef} = (f_x, f_y, c_x, c_y)$$

Um die Bildweite und den Bildmittelpunkt zu bestimmen, wird ausgenutzt, dass von einem Lochkameramodell ausgegangen wird. In diesem Modell lässt sich die Transformation zwischen Punkten in Kamerakoordinaten und Pixelpositionen wie folgt darstellen:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = imageMapping(u', v', Mapping_{coef}), \text{wobei}$$

$$imageMapping(u', v', Mapping_{coef}) = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}$$

Hier wird die korrigierte Position (u', v') verwendet, welche entsprechend des Verzerrungsmodells berechnet wurde. Das Ergebnis ist die Pixelposition (u, v) im Bild der Kamera. Es werden in der Transformation die vier neuen Parameter f_x, f_y, c_x, c_y verwendet. Die Matrix, in welcher sie enthalten sind, wird als Kameramatrix bezeichnet. Die Parameter f_x sowie f_y beschreiben die Bildweite in Pixeln. Dabei ist zu beachten, dass diese Parameter in der Literatur häufig als Brennweite bezeichnet werden. Dies ist allerdings nur eine Annäherung. Es handelt sich hier tatsächlich um die Bildweite, was für die spätere Berechnung der Markerentfernungen eine wichtige Rolle spielt. Die Parameter c_x und c_y geben an, auf welchem Pixel der Mittelpunkt des Bildes liegt. Der Mittelpunkt ist dabei der Punkt, an dem ein Lichtstrahl senkrecht auf den Bildsensor trifft. Da in dem Kameramodell von einer punktförmigen Öffnung der Kamera ausgegangen wird, kann es nur einen solchen Punkt geben. Ist dieser Ort bekannt, so können leichte Verschiebungen des Bildsensors modelliert werden.

Insgesamt ergeben sich für die Kalibrierung demnach neun Parameter: die fünf Parameter, welche die Verzerrung modellieren, sowie die vier Parameter der Kameramatrix. Diese Parameter werden auch intrinsische Parameter genannt, weshalb der in diesem Kapitel vorgestellte Kalibrierungsprozess die intrinsische Kalibrierung genannt wird. Die intrinsischen Parameter Θ werden im Verlauf der Arbeit wie folgt angegeben:

$$\Theta = (Distortion_{coef}, Mapping_{coef})$$

Zusammen definieren das Verzerrungsmodell und die Kameramatrix eine Abbildung f, welche gegebene Punkte in Kamerakoordinaten in Pixelpositionen im Bild der Kamera konvertiert. Die Funktion berechnet die Pixelposition (u, v) für einen Punkt $p^{(C)}$ im

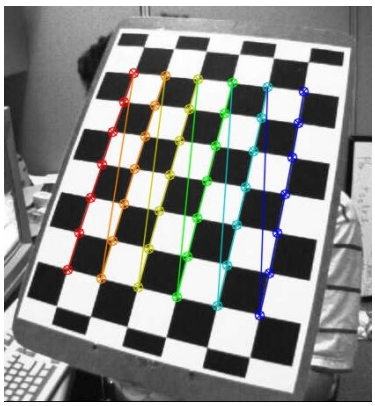

Abbildung 3.1: Kalibrierungspattern
Ein in OpenCV erkanntes Schachbrettmuster[25]

Kamerakoordinatensystem, wobei die intrinsischen Parameter Θ für die Konvertierung benötigt werden:

$$f(p^{(C)}, \Theta) = imageMapping(distort(perspProj(p^{(C)}), Distortion_{coef}), Mapping_{coef})$$

Diese Funktion f wird in folgenden Kalibrierungen in dieser Arbeit eine wichtige Rolle spielen. Um einige Berechnungen in dieser Arbeit zu vereinfachen, wird eine allgemeinere Notation für f wie folgt eingeführt:

$$f(p^{(X)}, T_{C \leftarrow X}, \Theta) = f(T_{C \leftarrow X} \cdot p^{(X)}, \Theta)$$

In dieser Notation ist der Punkt nicht bereits in Kamerakoordinaten gegeben. Stattdessen ist er in einem beliebigen Koordinatensystem X ausgedrückt. Um den Punkt trotzdem in das Bild projizieren zu können, wird die Transformation $T_{C \leftarrow X}$ vom Koordinatensystem X in das Kamerakoordinatensystem als Parameter übergeben.

Mithilfe des oben definierten Kameramodells lässt sich nun die Kalibrierung durchführen. Kalibriert werden die intrinsischen Parameter mithilfe eines Kalibrierungspatterns. Auf diesem Pattern soll es möglichst leicht sein, Punkte im Bild zu erkennen, welche einen bekannten oder konstanten Abstand zueinander haben. Aus diesem Grund werden häufig Schachbrettmuster oder Kreismuster genutzt. Ein solches Schachbrettmuster ist in Abbildung 3.1 gezeigt. Sowohl die Eckenerkennung im Schachbrettmuster als auch die Kreiserkennung ist in OpenCV mit Subpixel Genauigkeit implementiert [23], was die Wahl des Musters zu einem vernachlässigbaren Faktor in der Kalibrierung macht. In dieser Arbeit wurden Schachbrettmuster genutzt. Dieses Schachbrettmuster wird mit der zu kalibrierenden Kamera aus verschiedenen Winkeln aufgenommen. Dabei ist es insbesondere wichtig, dass das Schachbrett über alle aufgenommenen Bilder insgesamt

den ganzen Bildraum abdeckt. Vor allem an den Kanten des Bildes kommt es zu starken Verzerrungseffekten, weshalb das Schachbrett in einigen Bildern auch ganz am Rand zu sehen sein muss. In einem typischen Kalibrierungsprozess wird das Schachbrett aus mindestens drei verschiedenen Entfernungen aufgenommen. In jeder dieser Entfernungen wird es so bewegt, dass es überall im Bild zu sehen war. Dabei wird das Schachbrett gelegentlich geneigt, um auch in der Tiefe Informationen zu erhalten.

Die Eingabe der Kalibrierung setzt sich demnach aus den verzerrten Bildern zusammen sowie den auf ihnen erkannten Positionen der Ecken auf dem Pattern. Das Ziel der Kalibrierung ist nun, aus diesen Eingabebildern die neun intrinsischen Parameter für die Kamera zu schätzen. Zur Kalibrierung versucht der Kalibrierungsalgorithmus von OpenCV, die Distanz zwischen zwei Mengen an Punkten zu minimieren, nachdem die eine Menge mithilfe des oben genannten Verzerrungsmodells auf die andere abgebildet wurde. Die erste Menge an Punkten, die Bildpunkte, besteht aus den zuvor erkannten Ecken im Schachbrettmuster. Die zweite Menge enthält die erwarteten Punkte, auch genannt Objektpunkte, an denen die Ecken zu sehen sein sollten. Die Einheit dieser Punkte spielt hier keine Rolle, da bei dieser Kalibrierung nur die intrinsischen Parameter kalibriert werden. Wichtig ist nur, dass die Abstände zwischen diesen Punkten korrekt sind. Das Kalibrierungstool erstellt die Punkte, indem die Punkte des $N \times M$ Schachbretts durchnummeriert werden. N ist dabei die Anzahl der Ecken in der Breite des Schachbretts, während M die Anzahl in der Höhe beschreibt. Der erste Punkt liegt demnach bei $(0, 0, 0)^T$, der letzte Punkt bei $(N, M, 0)^T$. Die z-Koordinate kann hier beliebig gewählt werden, sie muss aber für alle Punkte gleich sein, da es sich bei dem Schachbrett um eine Ebene handelt.

Nachdem diese Menge an Objektpunkten erstellt wurde, können die beiden Mengen der OpenCV Kalibrierungsfunktion übergeben werden. Diese versucht, die oben genannten Parameter zu optimieren, sodass der Fehler zwischen je zwei korrespondierenden Punkten aus den beiden Mengen möglichst niedrig ist, nachdem der Objektpunkt anhand der Funktion f in das Bild projiziert wurde. Das Ergebnis enthält zum einen die neun kalibrierten Parameter, mit welchen von nun an die Kamerabilder entzerrt werden können. Zum anderen beinhaltet es den Reprojektionsfehler. Gute Kalibrierungen zeichnen sich in den meisten Fällen durch einen niedrigen Reprojektionsfehler aus. Dieser wird als quadratisches Mittel, im Folgenden auch RMS(root mean square) genannt, angegeben. Zum Berechnen des RMS des Reprojektionsfehlers wird zu jedem ins Bild projizierten Punkt betrachtet, wie weit er von dem tatsächlich beobachteten Punkt in Pixeln entfernt ist. Diese Werte werden jeweils quadriert und die Quadrate aufsummiert. Anschließend wird diese Summe durch die Anzahl n an Punkten geteilt und von diesem Ergebnis die Wurzel berechnet. Als Gleichung ergibt sich:

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||(z_i - f(p_i^{(O)}, T_{C\leftarrow O}, \Theta))||^2}$$

Dabei ist z_i der erkannte Bildpunkt, während p_i der Punkt auf dem Schachbrett ist, welcher ins Bild projiziert wird. Das Koordinatensystem O ist das lokale Koordinatensystem des Schachbretts, in welchem die Objektpunkte definiert wurden. Die Länge der resultierenden Vektoren wird genutzt, um den RMS zu berechnen. Insgesamt entsteht so ein Mittelwert, welcher höhere Abweichungen stärker gewichtet als niedrige Abweichungen. Diese Art, Mittelwerte von Verteilungen zu beschreiben, ist weit verbreitet und wird auch in dieser Arbeit in der Evaluation der entwickelten Verfahren eingesetzt.

3.3 Non-linear Least-Square Probleme

In dieser Arbeit werden an verschiedenen Stellen Non-linear Least-Square Probleme definiert, um Optimierungsprobleme zu lösen. Zur Lösung solcher Probleme wird in dieser Arbeit die Bibliothek Ceres [1] genutzt. Bei Non-Linear Least-Square Problemen sollen n nicht-lineare Parameter eines Modells so optimiert werden, dass sie m Observationen möglichst präzise beschreiben. Eine präzise Beschreibung zeichnet sich dabei durch eine niedrige Distanz entsprechend der Least-Square Methode aus. Zu jeder der m Observationen werden hierzu anhand des Modells ein oder mehrere sogenannte Residuen r_i berechnet, welche angeben, wie groß der Fehler zwischen der Observation i und dem aktuell geschätzten Modell ist. Aus der quadrierten Summe dieser Residuen wird dann ein Maß D dafür berechnet, wie gut das aktuell geschätzte Modell die Observationen modelliert. Es ergibt sich folgende Gleichung:

$$D(x) = \sum_{i=1}^{m} ||r_i(x)||^2, \hat{x} = \arg\min_x D(x)$$

Dieses Non-Linear Least-Square Problem versucht, $D(x)$ zu minimieren. Dabei beschreibt der Parameter x das zu optimierende Model. Es muss so angepasst werden, dass die damit berechneten Residuen möglichst nah an den Observationen liegen. Die Gleichung lässt vektorielle r_i zu, bei denen jede Komponente ein einzelnes Residuum definiert. Dies erleichtert das Notieren der konkreten Modelle im Verlauf dieser Arbeit. Dieses Optimierungsproblem lässt sich in Ceres modellieren und approximativ lösen. Das bedeutet, dass Ceres nicht in jedem Fall das globale Minimum für $\hat{x}$ findet. Es kann auch passieren, dass in einem lokalen Minimum eine Lösung gefunden wird, über den gesamten Wertebereich der Parameter aber noch bessere Lösungen existieren. Daher wird Ceres jeweils eine initiale Schätzung aller Parameter gegeben, um dieses Risiko zu verringern. Zum Lösen von Optimierungsproblemen benötigt Ceres lediglich eine Methode zur Berechnung der Residuen sowie die Information, welche Parameter Teil des Modells sind und entsprechend für die Optimierung berücksichtigt werden sollen. Alle weiteren benötigten Informationen leitet sich Ceres automatisch aus den gegebenen ab. Wichtig ist dabei, dass durch Threading in der Berechnung kein Determinismus garantiert werden

kann. Zwischen zwei Berechnungen auf den selben Eingabedaten können unterschiedliche Ergebnisse produziert werden.

3.4 Bestimmung von Objektposen

Eine grundlegende Fragestellung beim Aufnehmen der Ground-Truth Daten in dieser Arbeit ist, wie die Objektposen bestimmt werden können. Während die verwendeten Methoden im späteren Verlauf der Arbeit vorgestellt werden, soll hier bereits eingeordnet werden, welche Informationen zur Bestimmung einer Objektpose benötigt werden.

Bei den in dieser Arbeit genutzten Posen handelt es sich um Posen mit sechs Freiheitsgraden, auch genannt 6DoF Posen. Die sechs Freiheitsgrade sind zum einen die Verschiebung im dreidimensionalen Raum. Zum anderen enthalten sie die Rotation, welche durch Euler-Winkel beschrieben werden kann. Die Freiheitsgrade sind in Abbildung 3.2 abgebildet.

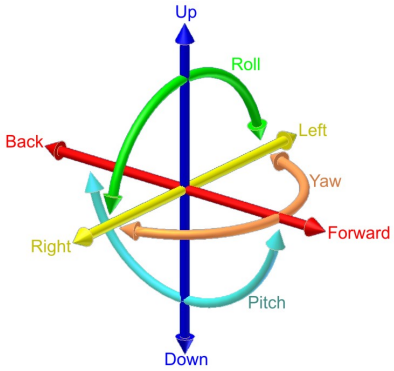

Abbildung 3.2: Freiheitsgrade
Die sechs Freiheitsgrade einer Objektpose im dreidimensionalen Raum [15].

Diese Repräsentation ist identisch zu der von Transformationen. Objektposen beschreiben, wo ein Objekt in einem externen Koordinatensystem zu verorten ist. Diese Eigenschaft lässt sich als Transformation zwischen dem Objektkoordinatensystem und dem externen Koordinatensystem ausdrücken. Posen und Transformationen können daher auf die gleiche Weise repräsentiert werden.

Die Freiheitsgrade der Pose müssen nun mithilfe von Messungen bestimmt werden, um eine eindeutige Objektpose zu erhalten. Hierzu wird zunächst einmal davon ausgegangen, dass es auf dem Objekt bekannte Punkte gibt, die mithilfe von Sensoren wiedererkannt werden können und deren 3D-Position bestimmt werden kann. Nimmt der Annotator diese Punkte nun nach und nach auf, so werden mit jeder Messung Freiheitsgrade eliminiert. Der erste Punkt eliminiert drei der sechs Freiheitsgrade, da es sich bei dem Punkt um eine Messung mit drei Dimensionen handelt. Die möglichen Posen werden

dadurch aber noch nicht weit genug einschränkt, da dieser Punkt nun zwar fixiert ist, das Objekt aber um diesen Punkt beliebig rotiert sein kann. Wird der zweite Punkt aufgenommen, so sind bereits sechs Freiheitsgrade bestimmt. Diese sollten eigentlich genügen, um die Pose zu bestimmen. Das ist allerdings nicht der Fall. Einer der bisher sechs bestimmten Freiheitsgrade liefert keine Informationen über die Objektpose, sondern über die Beziehung zwischen den zwei bisher aufgenommenen Punkten, weshalb effektiv noch ein Freiheitsgrad unbestimmt bleibt. Das Objekt kann noch um die Gerade zwischen den beiden bekannten Punkten beliebig rotiert sein. Ein dritter Punkt eliminiert dann auch diesen Freiheitsgrad, wodurch die Pose vollständig bestimmt ist.

Diese Abschätzung der benötigten Messungen zeigt, dass mindestens drei Punkte auf dem Objekt bestimmt werden müssen, um eine Pose zu berechnen. Um die benötigten Arbeitsschritte zum Aufnehmen von Objekten minimal zu halten, wird das System auch nur mit diesen drei Punkten arbeiten. Die Vorbedingung hierzu war allerdings, dass die Punkte auf dem Objekt bekannt sind. Die Objekte müssen also ausgemessen werden, sodass die Positionen der Punkte am Objekt bekannt sind.

Zum Ausmessen der Objekte muss zunächst in jedem Objekt festgelegt werden, wo der Ursprung ist. Hierzu werden drei Regeln verwendet, welche für alle Objekte bestimmen, wo der Ursprung liegt. Die folgenden Regeln gelten:

- Der Ursprung liegt in der Auflagefläche des Objekts
- Hat das Objekt von oben betrachtet eine Form, deren Mittelpunkt leicht zu bestimmen ist (Kreis, Viereck), so liegt der Ursprung im Mittelpunkt dieser Form
- Hat das Objekt einen Griff, so liegt der Ursprung am Anfang des Griffs

Diese Regeln decken alle Objekte im genutzten Datensatz ab. Nun muss lediglich pro Objekt festgelegt werden, wie die Achsen verlaufen. Hierfür lassen sich nicht so leicht Regeln bestimmen, weshalb dies pro Objekt festgelegt wird. Für alle Objekte mit Griff verläuft beispielsweise die x-Achse entlang des Griffs, die z-Achse zeigt für alle Objekte nach oben. Anhand dieser Informationen lassen sich nun die Objekte ausmessen. Dies kann Beispielsweise mit einem Messschieber getan werden. Es muss für jeden Punkt auf dem Objekt, der später zur Bestimmung von Objektposen dienen soll, die x-, y- und z-Koordinate bestimmt werden. Hierzu wird die Distanz der Punkte zum Ursprung auf den jeweiligen Achsen gemessen.

4 Überblick über das System

Bevor in den folgenden Kapiteln verschiedene Methoden vorgestellt werden, Objektpunkte und Posen zu bestimmen, wird in diesem Kapitel zunächst ein Überblick über das Ground-Truth System im Ganzen gegeben. Es wird vorgestellt, wie die Arbeitsumgebung aufgebaut ist, welche Sensoren das System verwendet und auf welche Art mit diesen Sensoren Ground-Truth Daten erfasst werden.

4.1 Aufbau des Systems

Die grundlegende Aufgabe des Ground-Truth Systems ist es, anhand von drei Punkten an einem Objekt die Pose dieses Objekts zu bestimmen. Im Ground-Truth Datensatz werden diese Objektposen im Koordinatensystem der 3D-Kamera benötigt, mit welcher die Bilder aufgenommen werden. Die Posen direkt in diesem Koordinatensystem aufzunehmen ist allerdings nicht praktikabel, da die Kamera zur Berechnung der Posen drei bekannte Punkte an einem Objekt erkennen müsste. Selbst wenn es möglich wäre, diese drei Punkte direkt aus dem Kamerabild präzise zu bestimmen, würden durch Verdeckungseffekte häufig Punkte gar nicht im Bild gesehen werden. Stattdessen werden die Punkte in dieser Arbeit von zusätzlichen Sensoren aufgenommen, welche eine bekannte Transformation zu der Kamera haben. Dadurch ist es möglich, die Punkte an den Objekten aufzunehmen, obwohl sie von der Kamera aus nicht direkt sichtbar sind. Insgesamt ist das System also so aufgebaut, dass die 3D-Kamera die Bilder für den Ground-Truth Datensatz aufnimmt. In diesen Bildern sind Objekte zu sehen, welche mit Posen annotiert werden müssen. Diese Posen werden von extra Sensoren geliefert, indem mit diesen Sensoren drei Punkte aufgenommen werden, aus welchen dann eine Objektpose berechnet wird. Hierzu arbeitet das System insgesamt mit drei Sensoren: Dem Kamerasensor eines Smartphones, einem Infrarot-Tracking System, und der 3D-Kamera, mit welcher der Datensatz aufgenommen wird. Dieser Aufbau ist in Abbildung 4.1 zu sehen.

Der erste Sensor ist die Kamera eines Android Smartphones. In dieser Arbeit wurde das Samsung Galaxy J5 [29] genutzt. Das Smartphone dient zur Erkennung der Positionen von Markern an den Objekten. Es ist zugleich aber auch das User Interface des Systems. Hiermit kann der Annotator entscheiden, welche Objekte aktuell aufgenommen werden, welches Verfahren hierzu verwendet wird und verschiedene andere Einstellungen treffen. Abbildung 4.2 zeigt das Smartphone sowie die von ihm aufgenommenen Marker an einem Objekt. Auch abgebildet ist das Tracking-Target, welches über dem Smartphone montiert ist.

Abbildung 4.1: Systemaufbau
In rot: Die Kameras des DTrack2 Systems
In blau: Die 3D-Kamera rc_visard
In gelb: Das Smartphone

Der zweite Sensor, welcher in dieser Arbeit genutzt wird, ist eine 3D-Kamera. Diese liefert die 3D-Bilder, für welche die Ground-Truth Posen berechnet werden sollen. Zum Erstellen des Datensatzes werden zwei verschiedene 3D-Kameras verwendet, welche abwechselnd eingesetzt werden. Zum einen wird die Kinect2 [16] der Firma Microsoft verwendet, zum anderen die rc_visard [26] der Firma Roboception. Diese zwei Systeme unterscheiden sich stark in der Art, wie sie Bilder aufnehmen. Damit einher gehen verschiedene Stärken und Schwächen. Insbesondere liefern die beiden Systeme unterschiedliche Daten. Die Kinect liefert ein Farbbild und ein Tiefenbild, in welchem zu jedem Pixel die Distanz zu der Kamera gespeichert ist. Die Tiefeninformationen der Pixel werden daraus berechnet, wie lange ausgesendetes Infrarotlicht benötigt, um zurück in eine Infrarotkamera reflektiert zu werden, ein sogenanntes Time-of-Flight Verfahren [20]. Die rc_visard hingegen liefert zwei Farbbilder und ein Tiefenbild, welches die Tiefe der Pixel in dem Farbbild der linken Kamera angibt. Die rc_visard Kamera berechnet das Tiefenbild mithilfe eines Stereoalgorithmus, also aus Bildern zweier Kameras, welche in

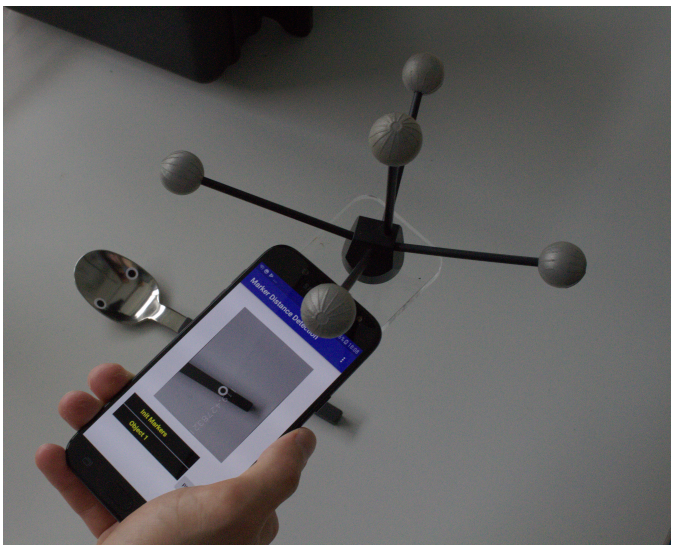

Abbildung 4.2: Smartphone mit Tracking-Target
Das Smartphone mit dem an ihm befestigten Tracking-Target. Im Bild des Smartphones ist ein Kreismarker zu sehen, welcher vom Smartphone aufgenommen wird.

einem festen Abstand zueinander angebracht sind [26]. Für Nutzer des Datensatzes kann es nützlich sein, diese zwei Farbbilder für jede aufgenommene Szene zu erhalten. Um einen Datensatz aufzunehmen, der von diesen Unterschieden unabhängig ist, werden in dieser Arbeit beide Systeme verwendet. Genau wie das Smartphone werden auch die beiden 3D-Kameras mit Tracking-Targets versehen. So ist es möglich, vom Koordinatensystem des Smartphones in das Koordinatensystem der jeweiligen 3D-Kamera zu transformieren.

Der letzte verwendete Sensor ist das in Kapitel 2 vorgestellte Infrarot-Tracking System TRACKPACK, hier in Rot markiert. Dieses System definiert in dieser Arbeit das Weltkoordinatensystem, also das Koordinatensystem, in welches alle anderen aufgenommenen Posen und Koordinaten transformiert werden können. Wo sich dieses Weltkoordinatensystem tatsächlich in der Arbeitsumgebung befindet, ist kalibrierbar und somit in zwei verschiedenen Datensätzen nicht zwangsläufig am selben Ort. Es befindet sich aber in der Regel auf der Arbeitsfläche der Küche.

Intuitiv ermöglicht das Trackingsystem somit, die in anderen Sensoren aufgenommenen Objektposen in das Koordinatensystem der 3D-Kamera zu transformieren. Dadurch können 3D-Bilder aufgenommen werden, in welchen bekannte Objekte mit Posen annotiert sind. Um dies zu ermöglichen, müssen die Sensorposen im Weltkoordinatensystem bekannt sein. Zum Bestimmen der Posen werden die an den beiden Sensoren befestigten Targets verwendet. Diese werden von dem Trackingsystem aufgenommen und liefern ihre 3D-Pose direkt im Weltkoordinatensystem. Die Pose des Targets ist allerdings nicht die gesuchte Pose des Sensors. Der Sensor ist zu dem Target verschoben und gedreht.

Beide sind aber fest verbunden, weshalb eine Transformation zwischen Target- und Sensorkoordinatensystem gefunden werden kann. Diese Transformation wird in dieser Arbeit kalibriert, was im Kapitel 5 erklärt ist.

4.2 Koodinatensysteme

Durch die Nutzung mehrerer Sensoren existiert eine Vielzahl verschiedener Koordinatensysteme in der Arbeitsumgebung. Um darüber einen besseren Überblick zu geben, sind diese in Abbildung 4.3 in die Arbeitsumgebung eingezeichnet. Dabei repräsentieren grüne Linien gemessene Transformationen, während blaue Linien statische Transformationen beschreiben, die zunächst kalibriert werden müssen. In Klammern neben den Namen der Koordinatensysteme sind ihre Abkürzungen aufgeführt. Diese werden im Verlauf der Arbeit entsprechend genutzt.

Nachdem nun der Aufbau mit seinen verschiedenen Koordinatensystemen feststeht, lässt sich präzisieren, was das gewünschte Resultat des Messvorgangs ist und welche Schritte hierfür benötigt werden. Gesucht ist eine Transformation $T_{C\leftarrow O}$, die Punkte von Objektkoordinaten in das Koordinatensystem der 3D-Kamera transformiert. Als Gleichung dargestellt ergibt sich also $p^{(C)} = T_{C\leftarrow O} \cdot p^{(O)}$, wobei $p^{(O)}$ der Punkt in Objektkoordinaten ist und $p^{(C)}$ derselbe Punkt, ausgedrückt im Koordinatensystem der 3D-Kamera. Im Folgenden ist ein Beispiel gegeben, wie diese Transformation in der Praxis als Kette an Transformationen berechnet werden kann. Je nachdem wie die Objektposen aufgenommen werden, wird sich diese Transformationskette in der Praxis leicht unterscheiden. In dem Beispiel wird das Smartphone verwendet.

Die Transformation $T_{C\leftarrow O}$ lässt sich wie folgt ausdrücken:

$$T_{C\leftarrow O} = T_{C\leftarrow CT} \cdot T_{CT\leftarrow W} \cdot T_{W\leftarrow ST} \cdot T_{ST\leftarrow S} \cdot T_{S\leftarrow O}$$

Es wird also die Objektpose im Kamerakoordinatensystem des Smartphones aufgenommen ($T_{S\leftarrow O}$). Diese Pose lässt sich mit der Target-Kamera Kalibrierung zwischen Smartphonekamera und Target in das Koordinatensystem des Targets transformieren($T_{ST\leftarrow S}$). Die Pose des Smartphone-Targets in der Welt ist bekannt, da dieses von dem Infrarot-Tracking System verfolgt wird ($T_{W\leftarrow ST}$). Dasselbe gilt für das Target an der 3D-Kamera, was die Transformation von Weltkoordinaten in das System des 3D-Kamera-Targets erlaubt ($T_{CT\leftarrow W}$). Nun folgt noch eine letzte Transformation, welche die kalibrierte Target-Kamera Transformation zwischen 3D-Kamera und ihrem Target nutzt ($T_{C\leftarrow CT}$). Das Resultat ist eine Transformation, die Punkte im Objektkoordinatensystem in Punkte im Koordinatensystem der 3D-Kamera transformiert. Diese Transformationskette kann in Abbildung 4.3 nachvollzogen werden. Transformiert wird immer zwischen zwei verbundenen Koordinatensystemen, wobei im Koordinatensystem des Objekts begonnen und im Kamerakoordinatensystem aufgehört wird.

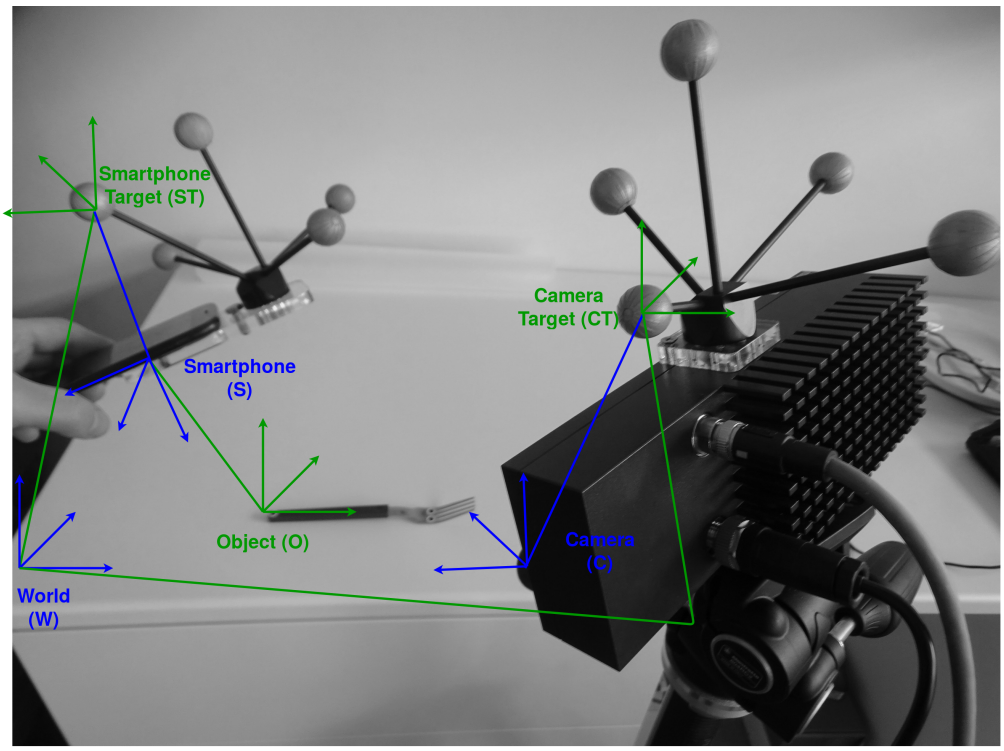

Abbildung 4.3: Koordinatensysteme in der Arbeitsumgebung
Visualisierung der verschiedenen Koordinatensysteme in der Arbeitsumgebung. Zwei Koordinatensysteme sind verbunden, wenn die Transformationen zwischen ihnen kalibriert oder gemessen wird. Kalibrierte Transformationen und die dazugehörigen Koordinatensysteme sind blau markiert, während gemessene grün markiert sind. Die Position und Ausrichtung der Koordinatensysteme entspricht zu Visualisierungszwecken eventuell nicht genau der Realität.

Wie bereits angemerkt, ist die tatsächliche Transformationskette abhängig von dem eingesetzten Messverfahren, da in einigen Fällen die Punkte nicht mit dem Smartphone aufgenommen werden, sondern mit anderen Verfahren. In einigen Verfahren wird die Objektpose beispielsweise nicht in das Koordinatensystem des Smartphones transformiert, sondern in das Tracking-Target oder direkt in das Weltkoordinatensystem. Wichtig ist, dass jedes dieser Verfahren es ermöglicht, die Objektpose in das Weltkoordinatensystem zu transformieren. Von dort kann die Pose weiter in das Koordinatensystem der 3D-Kamera transformiert werden. Das Prinzip ist also für alle Verfahren dasselbe, weshalb hier nicht alle Transformationsketten explizit aufgeführt werden. Im Folgenden werden die verwendeten Messverfahren kurz vorgestellt.

4.3 Eingesetzte Messverfahren

Insgesamt werden in dieser Arbeit vier verschiedene Messverfahren genutzt, um Objektposen aufzunehmen. Alle vier nutzen auf dem Objekt angebrachte Punkte verschiedener Art, deren Lage im Objektkoordinatensystem vorher eingemessen wurden. Die Verfahren werden im Verlauf dieser Arbeit wie folgt benannt:

- Marker3D
- Taststab
- MarkerRef
- Marker2D

Die Bezeichnung ist anhand der Art gewählt, wie die Punkte auf dem Objekt aufgenommen werden. Das Verfahren Marker3D nutzt die Smartphonekamera, um 3D-Positionen von auf den Objekten angebrachten Markern zu bestimmen. Das Verfahren Taststab hingegen verzichtet auf per Kamera erkannte Marker. Der Annotator berührt die Punkte stattdessen mit einem Taststab, welcher an dem Smartphone befestigt ist, um so ihre 3D-Positionen zu bestimmen. Das Verfahren MarkerRef nutzt, anders als die anderen Verfahren, nicht das Smartphone, sondern die Daten des Tracking-Systems. Die 3D-Positionen werden von reflektierenden Markern an den Objekten geliefert. Das letzte Verfahren, Marker2D, funktioniert ähnlich wie das Verfahren Marker3D, mit dem Unterschied, dass zu den Markern keine 3D-Positionen berechnet werden. Stattdessen werden die 2D-Pixelpositionen der Marker im Bild des Smartphones genutzt, um die Objektpose zu bestimmen. Diese vier Verfahren sind im Folgenden genauer beschrieben.

4.3.1 Marker3D

Das erste eingesetzte Verfahren, Marker3D, nutzt die Smartphone-Kamera um kreisrunde Marker zu erkennen, die, wie in Abbildung 4.2 gezeigt, an den Objekten angebracht sind. Diese Marker sind mit einem Durchmesser von fünf Millimetern möglichst klein gehalten, um das Aussehen des Objekts so wenig wie möglich zu beeinflussen. Um solche Marker präzise zu erkennen, muss ein sehr kurzer Arbeitsabstand eingehalten werden. Das Aufnehmen eines Objektes mit dieser Methode funktioniert, indem drei an dem Objekt befestigte Kreismarker nacheinander von einer für das Smartphone entwickelten Anwendung erkannt werden. Hierzu wird das Smartphone in eine Position gebracht, in welcher der Marker in der Mitte des Kamerabildes zu sehen ist. Mit einem Tippen auf den Bildschirm wird die Aufnahme bestätigt. In der Anwendung wird daraufhin die entsprechende 3D-Koordinate im Koordinatensystem der Smartphone-Kamera berechnet. Hierzu wird in der Berechnung ausgenutzt, dass die Marker eine bekannte Größe von fünf Millimetern haben. Aus den drei so aufgenommenen Punkten kann dann entsprechend die

Abbildung 4.4: Aufnehmen von Punkten mit dem Taststab

Objektpose berechnet werden. Voraussetzung hierfür ist, dass die Punkte im Koordinatensystem des Objekts bekannt sind. Hierzu werden die Punkte zuvor beispielsweise mit einem Maßband ausgemessen. Die Zuordnung der Punkte am Objekt zu den gemessenen Punkten ist dann gegeben durch die Reihenfolge, in der die Punkte aufgenommen werden. Die Anforderung, dass die Punkte ausgemessen werden müssen, wird auch für alle anderen Messverfahren bestehen und ist ein einmaliger Aufwand bei der Anschaffung von neuen Objekten.

4.3.2 Taststab

Das zweite Verfahren nutzt einen Taststab, welcher an dem Smartphone angebracht werden kann. Mit diesem Taststab können Punkte per Berührung aufgenommen werden. Dieses Prinzip ist in Abbildung 4.4 gezeigt. Hierfür wird ausgenutzt, dass die Spitze des Taststabs eine feste Verschiebung zum Target an dem Smartphone hat. Diese Verschiebung wird kalibriert, was im Kapitel 6 beschrieben ist. Mithilfe dieser Kalibrierung lässt sich die Position der Spitze des Taststabs in der Welt bestimmen. Da die Position des Targets in der Welt bekannt ist, muss lediglich noch die statische Verschiebung angewendet werden, um direkt von der Target-Pose auf die Position des Punktes zu schließen. Das Aufnehmen der Objektpose funktioniert dann ähnlich wie für die Kreismarker. Mit dem Taster werden nacheinander drei bekannte Punkte am Objekt berührt. Dies können beispielsweise die Mittelpunkte der Kreismarker sein. Alle anderen Punkte sind aber auch möglich, solange ihre Position im Objekt ausgemessen wurde. Bestätigt werden Punkte wieder durch Antippen des Smartphonebildschirms.

Abbildung 4.5: Reflektierende Marker auf einer Pappschablone

4.3.3 MarkerRef

Das dritte Verfahren, MarkerRef, verzichtet in der Erkennung der Punkte auf das Smartphone. Stattdessen werden reflektierende Marker an den Objekten angebracht, welche von dem Infrarot-Tracking System direkt erkannt werden. Diese Marker sind in Abbildung 4.5 zur Demonstration auf einer Pappschablone angebracht. Da die Marker von dem Infrarotsensor alle zur gleichen Zeit gesehen werden, ist die Zuordnung der Punkte zu Punkten auf dem Objekt etwas schwieriger als bei den anderen Verfahren, wo diese Zuordnung durch die Aufnahmereihenfolge gegeben ist. Stattdessen werden die Punkte anhand ihrer Abstände zu den anderen Punkten sortiert. Hierfür muss beim Aufkleben der Marker dafür gesorgt werden, dass kein Punkt zu den beiden anderen Punkten denselben Abstand hat. Sind die drei Punkte vom Infrarot-Tracking System aufgenommen worden und entsprechend den Objektpunkten zugeordnet, so sind auch in diesem Verfahren die Voraussetzungen gegeben, um die Objektpose zu bestimmen. Das Aufnehmen der Marker wird wie in den anderen Verfahren per Berührung des Smartphone-Bildschirms ausgelöst.

4.3.4 Marker2D

Das letzte Verfahren, Marker2D, ist ein experimentelles Verfahren, welches hinzugenommen wurde, um potentiell eine genauere Objektpose bestimmen zu können als in dem Verfahren Marker3D. Hierzu werden wie im ersten Verfahren Kreismarker im Bild des Smartphones erkannt. Anders als beim Marker3D-Verfahren werden zu diesen Kreismarkern aber keine 3D-Punkte im Koordinatensystem der Smartphone-Kamera berechnet. Stattdessen wird nur mit den 2D-Bildkoordinaten der Kreismittelpunkte gearbeitet, wel-

che im Vergleich zu den 3D-Punkten deutlich genauer berechnet werden. Diese Bildpunkte werden als Strahlen betrachtet, welche vom Smartphone aus durch die zugehörigen Punkte des Objekts verlaufen. Um aus diesen Informationen eine Objektpose zu erhalten, wird ein Non-linear Least-Square Problem formuliert und in einer dafür vorgesehen Bibliothek gelöst. Dies ist in Kapitel 7 genauer beschrieben. Verwendet wird dieses Verfahren genau wie das Verfahren Marker3D. Auch hier werden drei Kreismarker nacheinander mit dem Smartphone aufgenommen. Es unterscheidet sich lediglich in der internen Berechnung der Objektpose.

5 Kalibrierung

Wie sich bereits in den vorherigen Kapiteln angekündigt hat, werden einige Kalibrierungen benötigt, bevor das System in Betrieb genommen werden kann. Zunächst ist es nötig, das Infrarot-Tracking System vor dem Einsatz zu kalibrieren. Hierbei wird das Weltkoordinatensystem definiert. Weiterhin benötigen alle Kamerasensoren eine intrinische Kalibrierung. Diese wird demnach für die Smartphonekamera sowie für die beiden 3D-Kameras durchgeführt. Letztere verwenden intern jeweils zwei Kameras, welche auch extrinisch zueinander kalibriert werden müssen, um genaue Tiefendaten erzeugen zu können. Während für die Kameras der rc_visard bereits in der Produktion eine extrinsische Kalibrierung durchgeführt wurde [27], muss für die Kinect eine solche Kalibrierung noch erstellt werden. Abschließend werden die Transformationen der Targets zu den Sensoren kalibriert. Diese verschiedenen Kalibrierungsverfahren werden im Folgenden vorgestellt.

5.1 Intrinische Kalibrierung

Die intrinsische Kalibrierung aller Sensoren nutzt die in Kapitel 3 erläuterten Methoden der Bibliothek OpenCV. Diese Methode der Kalibrierung wird in dieser Arbeit um die Möglichkeit erweitert, Punkte auf dem Schachbrettmuster teilweise manuell zu erfassen. Dabei wird auf einem Tool aufgebaut, welches Udo Frese im Rahmen der Vorlesungsreihe Echtzeitbildverarbeitung entwickelte [9]. In diesem ist es möglich, Ecken auf Schachbrettmustern anzuklicken, um Bildkoordinaten der Ecke zu erhalten, die dem angeklickten Punkt am nächsten ist. In der Implementierung in OpenCV werden die Ecken auf dem Schachbrett vollautomatisch erkannt, ohne die Möglichkeit bei schlechten Ergebnissen einzugreifen. Auch tritt das Problem auf, dass Schachbretter, die sich am Rand des Bildes befinden, nicht erkannt werden. Darunter können die Kalibrierungsergebnisse leiden, da insbesondere am Rand des Bildes starke Verzerrungseffekte auftreten können. Das manuelle Anklicken dieser Punkte bietet dem Kalibrierer die Möglichkeit, die Ergebnisse zu prüfen und so Bilder zu erhalten, auf welchen alle Ecken des Schachbretts korrekt erkannt wurden. Dadurch ist das Verfahren unabhängig von der Position des Schachbretts im Bild.

In der intrinischen Kalibrierung geht es darum, die neun Kalibrierungsparameter so zu optimieren, dass ins Bild projizierte Objektpunkte gut auf die erkannten Bildpunkte passen. Objektpunkte sind dabei die Punkte der Ecken auf dem Schachbrett in Objektkoordinaten, Bildpunkte sind die Punkte, die im Bild erkannt werden. Das entwickelte Tool zum manuellen Kalibrieren ersetzt den Kalibrierungsschritt, in welchem die Bildpunkte

Parameter	Wert
f_x	539.288 561 72
f_y	540.285 217 62
c_x	320.520 501 44
c_y	240.062 731 41
k_1	−0.024 141 84
k_2	0.095 572 88
p_1	0.000 813 03
p_2	0.001 326 67
k_3	−0.082 679 11

(a) rc_visard

Parameter	Wert
f_x	731.247 310 52
f_y	317.064 087 34
c_x	731.185 663 9
c_y	241.437 463 17
k_1	0.288 132 729
k_2	−2.049 482 21
p_1	−0.000 039 210
p_2	0.000 497 920
k_3	3.642 943 08

(b) Smartphone

Tabelle 5.1: Kalibrierungsergebnisse der rc_visard und des Smartphones

automatisch erkannt werden. Statt eine Schachbretterkennung auf dem Bild auszuführen, wird dem Kalibrierer das Bild präsentiert. Dieser kann nun nach und nach die einzelnen Ecken des Schachbretts anklicken. An der Stelle, an der geklickt wurde, wird dann die in OpenCV implementierte subpixelgenaue Eckenerkennung ausgeführt. Die resultierenden Ecken werden in einer Datei pro Bild abgelegt. Sobald alle Bilder verarbeitet sind, werden die Bildpunkte für jedes Bild eingelesen und wie zuvor der Funktion zur Kamerakalibrierung übergeben. Während dieses Verfahren deutlich zeitaufwendiger ist, entsteht hierdurch eine Kalibrierung, welche auf Bildpunkten im gesamten Bild gearbeitet hat. Dieses Verfahren wurde für die Kamera des Smartphones sowie für die 3D-Kamera rc_visard durchgeführt. Die Ergebnisse sind in der Tabelle 5.1 aufgeführt. Die Kalibrierung der rc_visard erreicht dabei einen RMS von 0.167 Pixeln, während in der Kalibrierung des Smartphones ein RMS von 0.133 Pixeln zu beobachten ist.

Für die Kinect wurde dieses Kalibrierungsverfahren nicht genutzt, da für diesen Sensor bereits eine Bibliothek existiert, welche die intrinische und extrinsische Kalibrierung in einem Verfahren vereint. Diese ist im Folgenden beschrieben.

5.2 Extrinsische Kalibrierung der Kinect

Eine extrinsische Kalibrierung dient dazu, die feste Lage eines Kamerasensors zu einem externen Koordinatensystem zu bestimmen, in diesem Fall die Lage zu einem Infrarotkamerasensor.

In der Kinect ist eine extrinsische Kalibrierung notwendig, da die Kinect mit zwei verschiedenen Sensoren gleichzeitig arbeitet. Der erste ist der Kamerasensor, welcher ein Farbbild liefert. Der zweite ist ein Infrarotsensor, welcher die Entfernung jedes

Farbpunktes zur Kamera bestimmt. Durch eine Kombination dieser Sensoren entsteht ein Bild, in welchem zu jedem Pixel die Farbe sowie die 3D-Koordinate im Koordinatensystem der Kinect vorliegt. Um die Informationen der Sensoren zu kombinieren, muss allerdings die Lage der Sensoren zueinander bekannt sein. Dafür sorgt die extrinische Kalibrierung.

Zur extrinsischen Kalibrierung der Kinect wird die Bibliothek IAI Kinect2 genutzt [31]. Für die vollständige Kalibrierung müssen insgesamt drei Datensätze aufgenommen werden. Die ersten beiden Datensätze sind Bilder von Schachbrettmustern, die mit der RGB-Kamera und der Infrarotkamera aufgenommen werden, um sie jeweils mit dem beschriebenen Verfahren intrinsisch zu kalibrieren.

Für die eigentliche extrinsische Kalibrierung muss ein dritter Datensatz angelegt werden. In diesem Datensatz wird das Schachbrettmuster von beiden Sensoren gleichzeitig aufgenommen. Da die Kinect es nicht unterstützt, beide Bilder zur gleichen Zeit aufzunehmen, muss darauf geachtet werden, dass sich die Szene in der Zeit zwischen den Bildern nicht ändert. Das Schachbrettmuster kann daher für diese Kalibrierung nicht in der Hand gehalten werden, sondern muss in eine Halterung eingespannt werden. Dadurch wird sichergestellt, dass beide Sensoren exakt dasselbe Objekt aufnehmen. Aus den aufgenommenen Bildern soll die Transformation zwischen den beiden Sensoren bestimmt werden. Die Funktion stereoCalibrate, welche in OpenCV implementiert ist, liefert diese Transformation. Dazu nutzt sie aus beiden Bildern die erkannten Ecken im Schachbrettmuster sowie die zuvor kalibrierten intrinsischen Parameter. Theoretisch wäre es an dieser Stelle auch möglich, die intrinsischen Parameter mit berechnen zu lassen. Dies wären allerdings noch zusätzliche Parameter, welche potentiell eine zu hohe Dimensionalität des Optimierungsproblems beim Schätzen der Parameter herbeiführen würden. Dies würde die Genauigkeit der Kalibrierung verschlechtern, sodass eventuell einige Parameter nicht korrekt bestimmt würden.

Auch stereoCalibrate liefert, ähnlich wie in der intrinsischen Kalibrierung, neben der Transformation zwischen den Sensoren noch einen Reprojektionsfehler. Dieser gibt an, wie weit die im einen Bild erkannten Punkte auf dem Schachbrett von den Punkten im anderen Bild entfernt sind, nachdem sie mithilfe der kalibrierten Transformation in dieses Bild projiziert wurden. Die Ergebnisse der intrinsischen Kalibrierung der RGB-Kamera und der Infrarotkamera sowie der extrinsischen Kalibrierung sind in Tabelle 5.2 aufgeführt. Für die extrinische Kalibrierung wurden zur Darstellung Euler-Winkel rx, ry, rz genutzt, welche die Rotation um die x-, y- und z-Achse in Grad angeben. Die Verschiebung ist entlang der jeweiligen Achsen in Millimetern angegeben.

Der Reprojektionsfehler der intrinsischen Kalibrierung der RGB-Kamera liegt bei 0.1434 Pixeln, während der Reprojektionsfehler in der Kalibrierung der Infrarotkamera bei 0.1134 Pixeln liegt. Auch in dieser Kalibrierung wird damit die durchschnittliche Distanz der ins Bild projizierten Ecken auf dem Schachbrett zu den im Bild erkannten Ecken angegeben. Die extrinsische Kalibrierung erreicht einen Reprojektionsfehler von 0.21326 Pixeln. Alle Reprojektionsfehler sind dabei gemäß der RMS-Methode berechnet.

Parameter	Wert
f_x	1068.439 551
f_y	1062.649 462
c_x	964.285 813 9
c_y	544.331 655 2
k_1	0.077 340 761
k_2	−0.141 483 296
p_1	−0.000 128 378
p_2	0.001 709 520
k_3	0.064 851 600

(a) RGB-Kamera

Parameter	Wert
f_x	368.382 237 5
f_y	366.025 397 2
c_x	254.080 232 4
c_y	207.602 632 3
k_1	0.099 081 804
k_2	−0.295 457 08
p_1	0.001 277 805
p_2	−0.000 205 17
k_3	0.116 543 756

(b) Infrarotkamera

Parameter	Wert
x	−27.2418
y	−2.3783
z	−9.5343
rx	−0.183 590 5
ry	−2.500 004 9
rz	0.235 664

(c) Extrinsische Kalibrierung

Tabelle 5.2: Kalibrierungsergebnisse der RGB-Kamera, der Infrarotkamera und der extrinsischen Kalibrierung

5.3 Kalibrierung des Infrarot-Tracking Systems

Das Infrarot-Tracking System nutzt mehrere Infrarotkameras, um die Positionen und Posen von Targets in der Arbeitsumgebung zu verfolgen. Dabei definiert es das Weltkoordinatensystem, in welchem die Posen aller Sensoren bekannt sind. Bevor das Tracking-System einsatzbereit ist, benötigt es die Information, wie die Infrarotkameras zueinander ausgerichtet sind. Hierzu wird das System mithilfe von mitgelieferter Hardware kalibriert, welche in Abbildung 5.1 zu sehen ist.

Die Kalibrierung wird mithilfe des Tools DTrack2 [3] durchgeführt, welches von der Herstellerfirma ART des Tracking-Systems entwickelt wurde. Zur Kalibrierung wird der Koffer an einem beliebigen Ort in der Arbeitsumgebung platziert, sodass die vier Kugelmarker in dem Koffer von den Infrarotkameras gesehen werden können. Die Position des Koffers definiert das Weltkoordinatensystem, der Ursprung liegt in dem vorderen

Abbildung 5.1: Hardware zum Kalibrieren des Tracking-Systems
Die Hardware umfasst einen Koffer sowie einen Stab mit kugelförmigen Targets.

linken Marker. Die x-Achse verläuft vom Ursprung aus durch die zwei Marker weiter rechts, während die y-Achse entsprechend durch den letzten verbleibenden Marker nach hinten verläuft. Die z-Achse ist nach oben gerichtet. Kalibriert wird das System, indem der Stab mit den zwei Kreismarkern durch die Arbeitsumgebung geführt wird. Der Stab sollte im Verlauf der Kalibrierung durch den gesamten Arbeitsbereich geführt und dabei gedreht werden, um möglichst viele verschiedene Punkte in der Szene aufzunehmen. Aus diesen Punkten berechnet DTrack2 die Rotation und Translation zwischen den Infarot-Kameras. Dies ist ebenfalls eine Art der extrinsischen Kalibrierung. Ähnlich wie bereits in der Kalibrierung der Kinect wird hier die Transformation zwischen verschiedenen Kamerasensoren bestimmt.

Wurde dieser Kalibrierungsprozess durchgeführt, so ist das Tracking-System einsatzbereit. Es liefert nun Positionen von Markern sowie Posen von Targets im Weltkoordinatensystem, welches in dieser Kalibrierung definiert wurde.

5.4 Target-Kamera Kalibrierung

Nachdem die Sensoren kalibriert wurden, gilt es nun, alle Lücken in der Transformationskette des Systems zu füllen. Wie in Abbildung 4.3 gezeigt wurde, müssen einige der Transformationen kalibriert werden. Dabei handelt es sich jeweils um die festen Transformationen der Sensoren zu den an ihnen befestigten Tracking-Targets. Insgesamt gibt es drei Sensoren, für die eine solche Kalibrierung benötigt wird: Das Smartphone sowie die zwei 3D-Kameras, rc_visard und die Kinect. Für alle drei Sensoren kann die Kalibrierung

auf die gleiche Art durchgeführt werden. Es wird ein Non-linear Least-Square Problem wie in Kapitel 3 beschrieben definiert, welches mit dem Solver Ceres [1] gelöst wird.

5.4.1 Modellierung des kamerabasierten Non-linear Least-Square Problems

Um das Verfahren der Kalibrierung zu erläutern, wird zunächst betrachtet, wie die Residuen berechnet werden, anhand welcher Ceres die Transformation zwischen Target und Sensor bestimmt. Hierzu wird angenommen, dass bereits eine geschätzte Pose des Sensors in der Welt vorliegt. Weiterhin ist die fixe Position eines Kreismarkers in der Welt bekannt. Dieser Kreismarker ist im Bild der Kamera zu sehen. Der Aufbau ist in Abbildung 5.2 visualisiert, wobei hier der Marker bereits aus mehreren Perspektiven betrachtet wird, was es ermöglicht, die Transformation zu bestimmen.

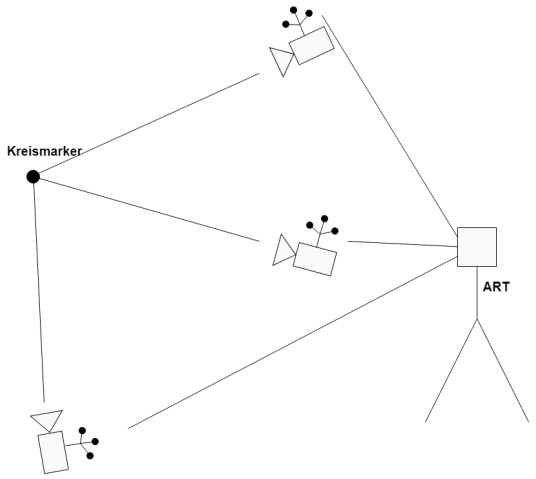

Abbildung 5.2: Aufbau der Target-Kamera Kalibrierung
Ein Marker wird von der Kamera aus mehreren Perspektiven aufgenommen und seine Position im Kamerabild sowie die Pose des Targets in der Welt bestimmt. Daraus werden die Position des Markers in der Welt und die Pose des Targets relativ zum Sensor bestimmt. ART stellt das Infrarot-Tracking System dar.

Mithilfe der Transformation von Weltkoordinaten in Sensorkoordinaten lässt sich der Kreismarker ins Sensorkoordinatensystem umrechnen. Damit ist die Position des Markers im Koordinatensystem der Kamera bekannt. Diese Position kann nun mithilfe der in Kapitel 3 vorgestellten Funktion f ins Bild projiziert werden. Die resultierende Pixelposition wird mit dem Ergebnis einer in dieser Arbeit entwickelten Kreiserkennung verglichen, welche in Kapitel 6 genauer beschrieben ist. Die von der Kreiserkennung berechneten Pixelpositionen dienen Ceres somit als Observationen, welche das geschätzte Modell möglichst genau repräsentieren soll. So lassen sich, gegeben der Pose des Sensors und einem Kreismarker in der Welt, Residuen für Ceres berechnen.

Zur Berechnung der Pixelposition wurde in diesem Beispiel angenommen, dass die Pose des Sensors in der Welt bereits schätzungsweise bekannt ist. Das zugrundeliegende Problem dieser Kalibrierung ist aber, dass die Transformation zwischen dem Target und dem Sensor bestimmt werden soll. Solange diese unbekannt ist, ist auch die Pose des Sensors in der Welt unbekannt. Die Transformation zwischen dem Target und dem Sensor ist der Parameter, der von Ceres durch das Optimierungsproblem bestimmt werden soll. Auch die Position des Kreismarkers wurde als bekannt angenommen. Da zur Kalibrierung immer derselbe Punkt aufgenommen wird, wäre es theoretisch möglich, ihn einmalig in Weltkoordinaten auszumessen. Es ist allerdings schwierig, die Position eines Punktes in einem dreidimensionalen Raum per Hand exakt auszumessen. Stattdessen wird auch dieser Punkt von der Kalibrierung bestimmt. Damit ergeben sich die Parameter, welche von Ceres kalibriert werden sollen. Die Position $p^{(W)}$ des Markers wird anhand seiner Koordinaten x, y, z kalibriert. Zur Kalibrierung der Transformation muss eine der in Kapitel 3 vorgestellten Repräsentationen ausgewählt werden. Ceres selbst unterstützt die Kalibrierung von Quaternionen, sodass das Quaternion nach jedem Kalibrierungsschritt noch normalisiert ist, also die Länge 1 hat. Das ist nötig, da nur normalisierte Quaternione genutzt werden können, um Rotationen darzustellen. Da Quaternione bereits unterstützt werden, werden diese für die Kalibrierung verwendet. Dementsprechend müssen die vier Werte des Quaternions q_w, q_x, q_y, q_z und die drei Koordinaten für die Translation x_t, y_t, z_t kalibriert werden. Die resultierende Transformation wird im Folgenden mit $T_{T \leftarrow S}$ bezeichnet, also eine Transformation vom Koordinatensystem des Sensors in das Koordinatensystem des Targets. Damit ist das Modell x für Ceres wie folgt gegeben:

$$x = (T_{T \leftarrow S}, p^{(W)})$$

Der Kalibrierungsprozess wird durchgeführt, indem der Kreismarker mit der Kamera aus verschiedenen Perspektiven aufgenommen wird. Dabei wird bei jeder Aufnahme die Pixelposition (u_i, v_i) des Kreismarkers im Bild gespeichert. Diese wird Ceres als Observation dienen, welche das Modell möglichst gut beschreiben soll. Weiterhin wird jeweils die aktuelle Transformation $T_{W \leftarrow T_i}$ des Targets in der Welt abgespeichert, um in Verbindung mit der kalibrierten Transformation die Pose des Sensors berechnen zu können. Diese Transformation liefert das Tracking-System. Für jede der Observationen müssen nun von Ceres Residuen berechnet werden. Hierzu wird wie folgt vorgegangen:

$$r_i(x) = \begin{pmatrix} u_i \\ v_i \end{pmatrix} - f(p^{(W)}, T_{S \leftarrow W_i}(x), \Theta)$$

Dabei lässt sich die Transformation $T_{S \leftarrow W_i}$ der Kamera in der Welt folgendermaßen berechnen:

$$T_{S \leftarrow W_i} = T_{T \leftarrow S}^{-1} \cdot T_{W \leftarrow T_i}^{-1}$$

An diesem Punkt hat Ceres alle benötigten Informationen, um den Optimierungsprozess zu starten. Die Parameter sind festgelegt, es sollen die Transformation zwischen Target und Sensor sowie die Position des Kreismarkers geschätzt werden. Weiterhin ist die Methode zur Berechnung der Residuen definiert, welche Ceres ein Maß dafür liefert, wie gut die Parameter geschätzt sind. Es werden nur noch genügend Observationen benötigt, um von ihnen auf die zehn Parameter schließen zu können.

Für die Kalibrierung wurde der Kreismarker pro Sensor etwa 70 Mal aufgenommen. Dabei wurde der Sensor in verschiedene Posen gebracht, um unterschiedliche Observationen für Ceres zu liefern. Aufnahmen aus der gleichen Pose wären effektiv immer dieselbe Observation, was für Ceres keine neuen Inforamtionen über das zu lösende Optimierungsproblem liefern würde. Hat Ceres zu wenig Informationen, so könnten Lösungen gefunden werden, die gut zu den Observationen passen, tatsächlich aber falsch sind.

Weiterhin wird für die Kalibrierung nach Augenmaß eine initiale Schätzung der Parameter angelegt. Diese Schätzung ist nötig, da Ceres sonst in vielen Fällen in lokalen Minima hängen bleibt, in denen die Residuen zwar lokal optimiert sind, im gesamten Wertebereich der Parameter aber noch bessere Lösungen zu finden sind. Die initialen Schätzungen sowie die resultierenden kalibrierten Transformationen der drei Sensoren sind in Tabelle 5.3 aufgeführt. Die Quaternione wurden zur besseren Anschaulichkeit in Euler-Winkel umgerechnet, da die initialen Schätzungen ebenfalls in Euler-Winkeln gemessen wurden. Die Winkel werden in Grad angegeben, während die Verschiebung in Millimetern gemessen wird.

Die Kalibrierung des Smartphones liefert dabei einen durchschnittlichen Reprojektionsfehler von 1.718 Pixeln. Dieser Wert gibt die durchschnittliche Pixeldistanz der ins Bild projizierten Punkte zu den von der Kreismarkererkennung berechneten Punkten an. Die Kamera rc_visard weist in der Kalibrierung einen Reprojektionsfehler von 0.636 Pixeln auf, während der Fehler in der Kalibrierung der Kinect bei 1.13 Pixeln liegt.

5.4.2 Evaluation

Die Transformation zwischen dem Koordinatensystem der 3D-Kamera und dem des Targets ist ein wichtiger Bestandteil der Transformationskette, welche die aufgenommenen Objektposen in das Koordinatensystem der 3D-Kamera transformiert. Ist die Rotation beispielsweise etwas falsch geschätzt, so kann das zu einer falschen Objektpose im Kamerakoordinatensystem führen. Bei einem Fehler von einem Grad kommt es auf einer Distanz von einem Meter bereits zu rund 1.7 Zentimetern Fehler in der Position des Objekts. Um eine Aussage darüber treffen zu können, wie groß dieser Fehler in dem in dieser Arbeit aufgenommenen Datensatz sein kann, wird die Target-Kamera Transformation im Folgenden evaluiert. Dabei werden lediglich die beiden 3D-Kameras betrachtet. Eine Evaluation des Smartphones an dieser Stelle ist nicht notwendig, da die Punkterkennung des Smartphones in Kapitel 6 evaluiert wird. Dort werden auch die Punkte nach der hier kalibrierten Transformation betrachtet.

rc	initial	kalibriert
x_t	30	33.096
y_t	-66	−67.688
z_t	-20	−27.263
rx	120	125.011
ry	0	−5.347
rz	0	3.616

(a) rc_visard

kinect	initial	kalibriert
x_t	20	21.94
y_t	-75	−69.483
z_t	10	12.013
rx	60	64.111
ry	-10	−28.364
rz	-110	−124.302

(b) Kinect

sm	initial	kalibriert
x_t	-80	−90.3654
y_t	-75	−68.8218
z_t	-30	−37.1583
rx	90	93.9828
ry	-130	−121.3415
rz	-180	−175.4012

(c) Smartphone

Tabelle 5.3: Kalibrierungsergebnisse der rc_visard, der Kinect und des Smartphones

Zur Evaluation der Kalibrierung wird ähnlich verfahren wie bereits zur Kalibrierung selbst. Es wird ein Kreismarker wiederholt aufgenommen und mithilfe der Kreismarkererkennung die jeweilige Pixelposition berechnet. Diese wird mit der Pixelposition des ins Bild projizierten Kreismarkers verglichen. Im Kalibrierungsschritt waren die Transformation zwischen dem Sensor und dem Target sowie die Position des Markers in der Welt unbekannt. Sie wurden von Ceres kalibriert. Für die Evaluation werden diese Werte nun benötigt, um die Position des Kreismarkers in das Bild zu projizieren. Für die Transformation wird das Ergebnis der Kalibrierung verwendet, welches evaluiert werden soll. Die Position des Markers muss in diesem Fall allerdings ausgemessen werden. Hierzu wird ein Stück reflektierende Folie in der Mitte des Kreismarkers angebracht. Das Infrarot-Tracking System nimmt diesen Punkt auf und liefert seine Position in der Welt. Dieser Aufbau ist in Bild 5.3 abgebildet.

Genau wie in der Kalibrierung wird der Marker mehrfach aus verschiedenen Perspektiven aufgenommen. Insgesamt werden pro Kamera 25 Punkte aufgenommen. Zu jeder dieser Aufnahmen wird ein Reprojektionsfehler berechnet, welcher beschreibt, wie wie stark der projizierte Punkt von dem erkannten Punkt abweicht. Diese Ergebnisse sind in Abbildung 5.4 visualisiert.

Hier sind die Pixeldistanzen auf der x- und y-Achse dargestellt. Zu beachten ist, dass

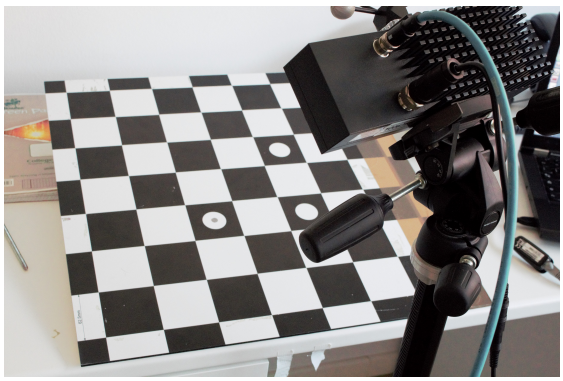

Abbildung 5.3: Aufbau zur Evaluation der Target-Kamera Kalibrierung
Der linke weiße Kreis wird als Kreismarker verwendet. In ihm ist reflektierende Folie angebracht, um den Punkt auszumessen.

die zwei Sensoren mit unterschiedlich aufgelösten Bildern arbeiten, weshalb ein großer Unterschied bezüglich der Fehlerwerte in Pixelkoordinaten zu beobachten ist. Während das Bild der Kinect mit 1920×1080 Pixeln aufgelöst ist, wurde für die Kamera rc_visard eine Auflösung von 640×480 verwendet. Um Fehlerwerte zu erhalten, die unabhängig von der Auflösung der Kamera sind, wurde die Distanz der Pixel in eine Distanz in Grad umgerechnet, was auf der zweiten x- und y-Achse in dem Diagramm zu sehen ist. Hierzu wird die Pixeldistanz durch die kalibrierte Bildweite der jeweiligen Kamera geteilt. Dies ist zwar lediglich eine Annäherung, in welcher angenommen wird, dass der Winkel zwischen zwei benachbarten Pixeln im gesamten Bild konstant ist, was in der Realität nicht der Fall ist, sie genügt aber zur Abschätzung der möglichen Abweichungen.

Anhand der Abweichungen in Grad lassen sich nun Aussagen darüber treffen, welche Genauigkeiten die Target-Kamera Kalibrierung erreicht. Hierzu wird die euklidische Distanz der Punkte zu dem Referenzpunkt gebildet. Diese Distanz in Pixeln lässt sich wie zuvor beschrieben durch die Bildweite der jeweiligen Kamera teilen, um sie in eine Winkeldistanz zu konvertieren. Für diese Ergebnisse wurde der RMS berechnet. Dabei ergibt sich für die Kinect ein RMS von 0.19 Grad, während die Evaluation der rc_visard einen RMS von 0.18 Grad aufweist.

Um diesen Wert nun mit tatsächlichen Fehlern in der Szene in Beziehung zu bringen, wird dieser Fehler von Grad in Millimeter umgerechnet. Hierzu muss allerdings eine Distanz fixiert werden, da für größer werdende Distanzen auch der zu erwartende Fehler in Millimetern ansteigt. Daher wird für diese Evaluation die Abweichung auf einer Distanz von einem Meter betrachtet. Dies ist eine übliche Arbeitsdistanz beim Aufnehmen der Bilder in einer Küchenumgebung, weshalb sie sich gut als Vergleichswert eignet. Auf einer Distanz von einem Meter ist bei einem Fehler von 0.18 Grad etwa eine Verschiebung von 3.13 Millimetern zu erwarten. Für Abweichungen von 0.19 Grad liegen die Fehler bei

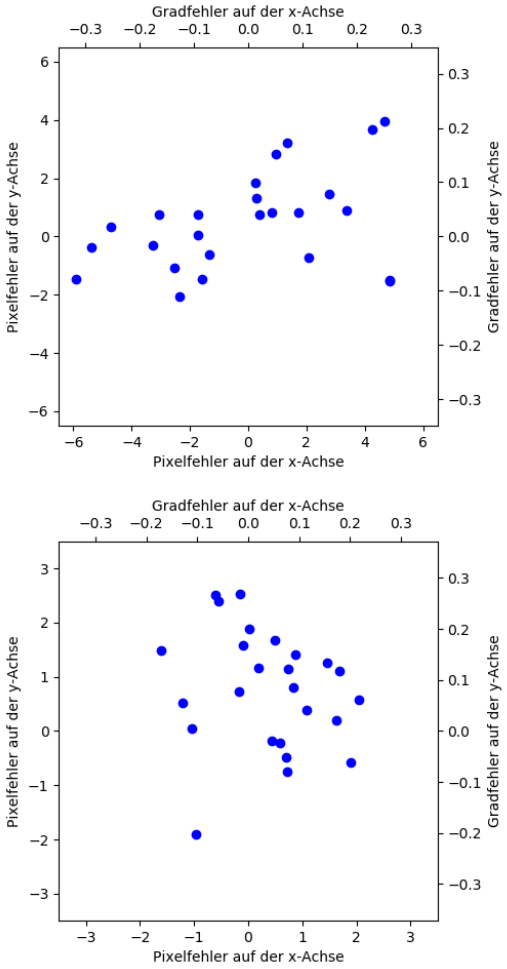

Abbildung 5.4: Ergebnisse der Evaluation der Target-Kamera Kalibrierung in Pixeln Evaluationsergebnisse der Kinect(oben) und der rc_visard(unten)

etwa 3.32 Millimetern. Da die Abweichung für die beiden Kameras sehr ähnlich ausfällt werden die beiden Kameras im Folgenden nicht mehr getrennt betrachtet. Stattdessen wird die Abweichung von 3.32 Millimetern als Fehlerwert der Target-Kamera Kalibrierung verwendet.

Zur Bewertung, welche Fehler für Ground-Truth Verfahren akzeptabel sind, hat sich in der Literatur keine einheitliche Metrik etabliert. Tatsächlich werden zu häufig verwendeten Datensätzen wie dem YCB object set [7] und dem Linemod Datensatz [13] keine Angaben darüber gemacht, welche Genauigkeiten in der Aufnahme der Ground-Truth Daten erreicht werden. Um in dieser Arbeit trotzdem eine Aussage über die Qualität der

Ground-Truth Daten treffen zu können und insbesondere darüber, ob der Datensatz als Ground-Truth Datensatz einsetzbar ist, werden Metriken zur Evaluation von Algorithmen zur 6DoF-Posenbestimmung herangezogen. Diese Metriken werden in der Literatur genutzt, um zu bestimmen, ob die Pose eines Objekts korrekt bestimmt wurde. Da anhand des in dieser Arbeit erstellten Datensatzes aber genau dies getestet werden soll, müssen die hier aufgenommenen Ground-Truth Posen deutlich genauer sein als die in der Literatur angesetzten Metriken. In der Evaluation der in dieser Arbeit aufgenommenen Ground-Truth Daten wird daher zwar an solche Metriken angelehnt, die aufgenommenen Ground-Truth Daten müssen aber um einen Faktor genauer sein. Da sich in der Literatur kein solcher Faktor etabliert hat, wird in dieser Arbeit die Annahme getroffen, dass mindestens ein Faktor von 4 benötigt wird, um ausreichende Genauigkeiten zu erreichen. Dies bedeutet, dass die Ground-Truth Posen vier Mal genauer sein müssen, als die Metrik zur Evaluation von Posen es fordert. Welche Genauigkeiten dadurch tatsächlich gefordert werden, hängt von der Metrik ab.

In der Arbeit von Li et al. [17] werden drei verschiedene Metriken angewendet, nach welchen bestimmt wird, ob Posen korrekt aufgenommen wurden. Die erste Metrik betrachtet den Fehler der bestimmten Pose in Millimetern in der Translation sowie in Grad in der Rotation. Diese Metrik wurde in der Arbeit von Shotton et al. [30] eingeführt, in welcher maximale Abweichungen von 5 Grad und 5 Zentimeter vorgeschlagen werden. In der Arbeit von Li et al. werden weiterhin die Werte 2 Grad und 2 Zentimeter sowie 10 Grad und 10 Zentimeter vorgeschlagen. Da die Ground-Truth Daten aber lediglich einen Viertel dieser Abweichungen erreichen dürfen, müsste der Datensatz Genauigkeiten von 0.5 Grad und 0.5 Zentimeter, 1.25 Grad und 1.25 Zentimeter oder 2.5 Grad und 2.5 Zentimeter erreichen. Erreicht der in dieser Arbeit erstellte Datensatz also beispielsweise eine Genauigkeit von 0.7 Grad in der Rotation und 0.7 Zentimetern in der Translation, so könnte er genutzt werden, um Algorithmen anhand der 5 Grad und 5 Zentimeter Metrik zu evaluieren, für 2 Grad und 2 Zentimeter wäre der Datensatz allerdings leicht zu ungenau.

Weiterhin wird in der Arbeit von Li et al. die sogenannte average distance (ADD) Metrik verwendet, welche von Hinterstoisser et al. [13] vorgeschlagen wurde. Diese Metrik betrachtet die durchschnittliche Distanz zwischen Punkten auf dem 3D-Modell eines Objekts, nachdem es zum einen in die Ground-Truth Pose und zum anderen in die berechnete Pose transformiert wird. Posen werden als korrekt erkannt markiert, wenn diese Distanz höchstens 10% des Objektdurchmessers beträgt. Weiterhin werden auch die Werte 5% und 2% in der Arbeit von Li et al. betrachtet. Diese Metrik ist allerdings in dieser Arbeit nicht direkt anwendbar, da auf die Benutzung von 3D-Modellen für die Objekte verzichtet wurde. Daher wird sie für die Evaluation in dieser Arbeit nicht betrachtet. Eine weitere vorgestellte Metrik betrachtet den Reprojektionsfehler, nachdem Punkte des 3D-Modells in das Bild der Kamera projiziert wurden. Dabei ist der Reprojektionsfehler die Differenz der Punkte auf dem Objekt mit der Ground-Truth Pose und der Punkte auf dem Objekt mit der berechneten Pose. Für diese Methode ergibt sich allerdings

dasselbe Problem wie zuvor, dass in dieser Arbeit keine 3D-Modelle der Objekte genutzt wurden. Aus diesem Grund wird zur Evaluation die erste Metrik betrachtet. Es wird also betrachtet, ob die Ground-Truth Daten Genauigkeiten von 0.5 Grad und 0.5 Zentimeter, 1.25 Grad und 1.25 Zentimeter oder 2.5 Grad und 2.5 Zentimeter erreichen.

Das Ergebnis der Evaluation der Target-Kamera Kalibrierung kann nun anhand dieser Metriken bewertet werden. Wie sich gezeigt hat, ist ein durchschnittlicher Fehler in der Translation von etwa 3.32 Millimetern zu erwarten. Dieser Wert ist genau genug, um die aufgenommenen Daten für alle drei vorgeschlagenen Metriken als Ground-Truth Daten zu nutzen. Es ist aber zu beachten, dass diese Evaluation lediglich eine Hälfte der Transformationskette untersucht. Bislang ungeklärt ist, wie sich die Bestimmung der Objektposen auf die Genauigkeit der Daten auswirkt. Hierzu wird in den folgenden Kapiteln zunächst vorgestellt, wie diese Objektposen aufgenommen werden. Im Kapitel 7 werden diese Objektposen evaluiert und ebenfalls anhand der hier vorgestellten Metriken bewertet.

6 Bestimmung der Punkte auf Objekten

Nachdem nun alle Transformationen zwischen Sensoren und Targets kalibriert wurden, sind die Bedingungen gegeben, um mit dem Aufnehmen von Objektposen zu beginnen. Das zugrundeliegende Verfahren wird dabei unabhängig vom verwendeten Sensor immer dasselbe sein. Es werden zunächst einmal drei markierte Punkte aufgenommen, welche dann in einem weiteren Schritt zu einer Objektpose kombiniert werden. Dieses Kapitel wird sich damit beschäftigen, wie solche Punkte aufgenommen werden können, während im nächsten Kapitel beschrieben wird, wie diese Punkte zu einer Objektpose kombiniert werden. Die verwendeten Methoden, welche im Kapitel 4 bereits kurz eingeführt wurden, sind die folgenden:

- Marker3D
- Marker2D
- Taststab
- MarkerRef

Jede dieser Methoden wird im Folgenden erläutert und abschließend auf ihre Genauigkeit untersucht.

6.1 Marker3D (Kreismarkererkennung)

Das Verfahren Marker3D, im Folgenden auch Kreismarkererkennung genannt, nutzt kleine Kreismarker, welche an bekannten Punkten an den Objekten angebracht sind, um die Objektpunkte zu erfassen. Die Marker haben einen Gesamtdurchmesser von 8 Millimetern, wovon 3 Millimeter weißer Rand sind. Die Mitte des Markers ist dagegen schwarz und hat einen Durchmesser von 5 Millimetern. Durch den weißen Rand um den schwarzen Punkt ist eine sichtbare Kante immer gewährleistet. Würde der schwarze Punkt direkt auf dem Objekt kleben, so könnte man ihn auf dunklen Objekten nicht erkennen. Abbildung 6.1 zeigt einen solchen Marker sowie ein Objekt, an welchem die Marker angebracht wurden. Hier wird auch der Vorteil dieser Marker deutlich: Durch ihren kleinen Durchmesser fallen sie im Bild nicht besonders auf und lassen sich an den meisten Objekten anbringen. In vielen Fällen gibt es sogar mehrere verschiedene Konfigurationen, in welchen drei Marker an einem Objekt angebracht werden könnten. In

Abbildung 6.1: Erkannter Marker und berechnete Distanz zum Smartphone

Anbetracht des Ziels, auch teils sich überlagernde Objekte aufnehmen zu können, ist es nützlich, nicht von einer festen Konfiguration abhängig zu sein. Werden Teile des Objekts so verdeckt, dass Marker in einer Konfiguration nicht aufgenommen werden können, so kann das Objekt stattdessen mit einer anderen Konfiguration beklebt und aufgenommen werden.

Ausgeführt wird die Kreismarkererkennung auf dem Smartphone Galaxy J5. Die Kamera des Smartphones liefert das Bild, in welchem die Kreismarker erkannt werden sollen. Grundsätzlich wäre es auch möglich, solche Kreismarker direkt in der 3D-Kamera zu erkennen, was jegliche Transformationen zwischen Koordinatensystemen der Sensoren überflüssig machen würde. Dieser Ansatz hätte allerdings einige Einschränkungen, welche in der Praxis das Aufnehmen von Objekten wenig praktikabel machen würde. Zum einen sind die Marker sehr klein und dementsprechend im Bild der 3D-Kamera kaum noch zu sehen. Dies würde die Genauigkeit der Erkennung verringern. Zum anderen müssten die Marker immer von der 3D-Kamera gesehen werden können. Diese Bedingung ist in der Realität nicht zu erfüllen. Insbesondere in Szenen, in welchen Verdeckungseffekte eine Rolle spielen, werden einige Marker hinter anderen Objekten verschwinden. Auch wäre es nicht möglich, Marker an der Rückseite von Objekten anzubringen.

Die Nutzung eines Smartphones löst diese Probleme. Mithilfe des Smartphones können die Marker aus einer niedrigen Entfernung aufgenommen werden, um möglichst präzise Messungen zu erzielen. Die Kamera des Smartphones erlaubt es dabei, Bilder in geringen Entfernungen scharf aufzunehmen. Weiterhin können auch teilweise durch andere Objekte verdeckte Marker oder solche, die an der Rückseite von Objekten angebracht sind, von dem Smartphone aufgenommen werden. Es bietet dem System die Möglichkeit, um

Objekte herum zu schauen. Objekte können aus allen Richtungen aufgenommen werden, in denen genug Platz ist um das Smartphone dort hin zu bewegen. Solange die Objekte mit Bedacht platziert werden, sollte es selten zu Situationen kommen, in denen das Smartphone den Marker nicht erreichen kann. Es kann bei der Aufnahme vorkommen, dass mehrere Kreismarker gleichzeitig im Bild des Smartphones zu sehen sind. In diesem Fall entscheidet sich die Kreismarkererkennung immer für den Marker mit der niedrigsten Distanz zum Bildmittelpunkt. So hat der Annotator die Kontrolle darüber, welchen Punkt er aufnimmt.

Die Kreismarkererkennung bietet somit ein passendes Werkzeug zum Aufnehmen von Objektpunkten. Sie ist für Android Geräte in Java geschrieben. Für die Bildverarbeitung werden einige Funktionen der Bibliothek OpenCV genutzt, welche in der Bibliothek OpenCV4Android für Android Geräte nutzbar gemacht wurde [24]. OpenCV ist eine Bibliothek, die eine breite Auswahl an Bildverarbeitungsmethoden und Visualisierungsfunktionen bietet. Die in der Kreiserkennung genutzten Funktionen beschränken sich allerdings zum großen Teil auf die Verwaltung und Visualisierung des Bildes. Die entwickelte Anwendung baut dabei auf der Bibliothek android-camera-calibration [28] auf, welche die Kommunikation zwischen der Smartphonekamera und OpenCV implementiert. Weiterhin bietet die Bibliothek bereits Einstellungsmöglichkeiten der Kamera wie die gewünschte Auflösung und ob ein Farb- oder Graustufenbild aufgenommen werden soll. Für diese Anwendung werden Bilder mit der Auflösung 640×480 aufgenommen. Farbinformationen werden nicht benötigt, weshalb das Graustufenbild verwendet wird.

Die Kreismarkererkennung kann als Kette von Bildverarbeitungsoperationen aufgefasst werden, wobei die Eingabe das Bild der Smartphone Kamera ist. In jedem Zwischenschritt werden die Informationen über das Bild weiter verarbeitet. Resultat des letzten Schrittes sind die Parameter, die den erkannten Kreis beschreiben. Insbesondere wird seine 3D-Position im Kamerakoordinatensystem ausgegeben. Die ausgeführten Operationen sind in Bild 6.2 aufgelistet. Im Folgenden werden die einzelnen Bildverarbeitungsschritte vorgestellt.

6.1.1 Bild entzerren

Wie in der Abbildung zu sehen ist, beginnt die Bildverarbeitung mit dem Entzerren des Bildes. Dies ist ein notwendiger erster Schritt und ist in Kapitel 4 genauer beschrieben. Das Resultat ist ein entzerrtes Bild, welches die Szene so widerspiegelt, als sei sie mit einer Kamera aufgenommen worden, welche die in Kapitel 5 kalibrierte Bildweite und den Bildmittelpunkt hat. Im Folgenden können dementsprechend diese Bildweite und der Bildmittelpunkt genutzt werden, um zwischen Pixeln und Punkten im Koordinatensystem der Kamera zu konvertieren.

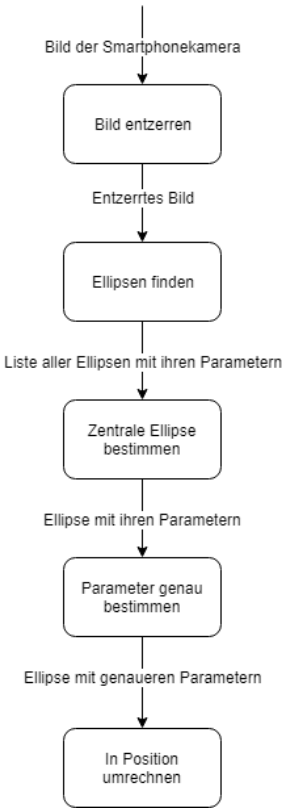

Abbildung 6.2: Zur Kreiserkennung ausgeführte Operationen und ihre Ergebnisse

6.1.2 Ellipsen finden

Im zweiten Schritt der Bildverarbeitung werden alle Ellipsen im Bild identifiziert. Es genügt nicht, Kreise im Bild zu identifizieren, da die Kreismarker aus unterschiedlichen Perspektiven aufgenommen werden. Im Kamerabild entsteht nur dann ein Kreis, wenn die Kamera den Marker genau von oben aufnimmt. Jede andere Perspektive ergibt im Bild eine Ellipse. Während sich ein Kreis durch seine Position und seinen Durchmesser beschreiben lässt, benötigen Ellipsen mehr Parameter zur Beschreibung. Sie lassen sich durch ihre Position, die Rotation der Hauptachse, die Länge der Hauptachse und die Länge der Nebenachse beschreiben. Diese Parameter sind in Abbildung 6.3 visualisiert.

Die Ellipsenerkennung ist an ein Verfahren angelehnt, welches in der Vorlesungsreihe Echtzeitbildverarbeitung von Udo Frese vorgestellt wurde [8]. Es ist ein effizientes Verfahren zur Bestimmung der Ellipsenparametern von allen Bereichen im Bild, die unter einen bestimmten Helligkeitsgrenzwert fallen. Dementsprechend werden nicht nur Ellipsenparameter für Bildbereiche berechnet, die die korrekte Form haben, sondern für

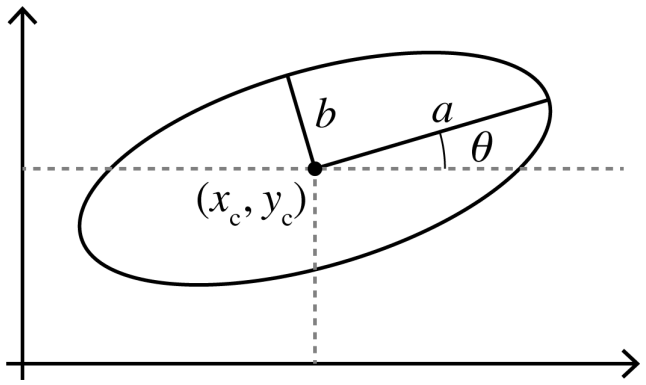

Abbildung 6.3: Parameter einer Ellipse
Die Parameter a, b beschreiben die Länge der Haupt- und Nebenachse, während der Parameter θ die Rotation der Hauptachse beschreibt. Die Parameter (x_c, y_c) geben den Mittelpunkt der Ellipse an [32].

alle dunklen Bereiche im Bild. Einer dieser Bereiche ist der Kreismarker. Das Verfahren nimmt hierzu als Eingabe das zuvor entzerrte Bild. Die Ausgabe des Verarbeitungsschritts sind alle Bildregionen, die dunkler als der Schwellwert sind. Zu jeder dieser Regionen werden die Ellipsenparameter berechnet, wobei es nicht darum geht, diese exakt zu bestimmen. Sie werden lediglich grob bestimmt, um herauszufinden, in welchen Regionen sich Ellipsen befinden. Die präzisen Parameter des Kreismarkers werden in späteren Schritten berechnet.

Das Verfahren zur Lokalisierung der Ellipsen bestimmt zu allen zusammenhängenden schwarzen Bereichen im Bild, also allen potentiellen Ellipsen, ihre Momente nullter bis zweiter Ordnung. Diese verschiedenen Momente beschreiben unterschiedliche Eigenschaften eines Bildbereiches Ω. So lassen sich mit diesen Momenten beispielsweise der Schwerpunkt des Bereichs, seine Fläche und seine Hauptträgheitsachsen bestimmen. Ein Moment über dem Bereich Ω ist dabei wie folgt definiert:

$$I_{x^i y^j} = \int_{(x,y)\in\Omega} x^i y^i dx dy$$

Dabei sind $(x, y) \in \Omega$ die Pixel, welche der Region, also der potentiellen Ellipse, zugehören. Die Ordnung des Moments berechnet sich aus $i + j$. Damit ist das nullte Moment beispielsweise direkt die Menge der Pixel in der Region. Die Momente erster Ordnung addieren hingegen die Pixelpositionen auf den beiden Achsen auf. In den Momenten zweiter Ordnung werden sowohl die Quadrate der Pixelpositionen als auch die Beziehung $x \cdot y$ zwischen den beiden aufaddiert. Aus diesen Momenten lassen sich nun die Ellipsenparameter der Region bestimmen. Die Fläche ist hierzu bereits durch

das nullte Moment gegeben. Der Schwerpunkt, also die Position der Ellipse, wird sich aus den Momenten erster Ordnung berechnen lassen. Für die Ausrichtung und Länge der Hauptträgheitsachsen werden die Momente zweiter Ordnung genutzt.Die Fläche A einer Region ist also, wie bereits erwähnt, lediglich eine Aufsummierung der in der Region enthaltenen Pixel. Damit ergibt sich folgende Formel:

$$A = I = \int_{(x,y)\in\Omega} 1 dx dy$$

Zur Berechnung des Schwerpunkts (x_s, y_s) werden die Momente erster Ordnung durch die Fläche geteilt. Die Formel hierfür lautet entsprechend:

$$x_s = \frac{I_x}{I}$$
$$y_s = \frac{I_y}{I}$$

Zur Berechnung der Parameter der Hauptträgheitsachsen werden zunächst die mittleren Werte der Momente zweiter Ordnung berechnet, wobei der Ursprung in den zuvor berechneten Schwerpunkt der Region gelegt wird. Hierzu wird wie folgt vorgegangen:

$$I'_{xx} = \frac{I_{xx}}{I} - x_s^2$$
$$I'_{xy} = \frac{I_{xy}}{I} - x_s y_s$$
$$I'_{yy} = \frac{I_{yy}}{I} - y_s^2$$

Diese Werte werden folgendermaßen in eine 2×2 Matrix eingetragen:

$$\begin{pmatrix} I'_{xx} & I'_{xy} \\ I'_{xy} & I'_{yy} \end{pmatrix}$$

Die Eigenwerte λ_1, λ_2 und Eigenvektoren dieser Matrix können nun verwendet werden, um die Hauptträgheitsachsen zu bestimmen. Die Eigenvektoren werden als Winkel θ dargestellt. Diese drei Werte werden wie folgt bestimmt:

$$\theta = \frac{1}{2} \arctan 2(-2I'_{xy}, I'_{yy} - I'_{xx})$$
$$c = \cos\theta$$
$$s = \sin\theta$$
$$\lambda_1 = c^2 I'_{xx} + 2csI'_{xy} + s^2 I'_{yy}$$
$$\lambda_2 = s^2 I'_{xx} - 2csI'_{xy} + c^2 I'_{yy}$$

Die Richtungen der Hauptträgheitsachsen sind nun entsprechend gegeben durch θ für die Hauptachse und $\theta + \frac{\pi}{2}$ für die Nebenachse. Die Ausdehnung wird durch den Halbmesser der Ellipse beschrieben. Die Hauptachse hat die Länge $a = 2\sqrt{\lambda_1}$, während die Länge der Nebenachse durch $b = 2\sqrt{\lambda_2}$ bestimmt wird.

Diese Art, Ellipsenparameter zu berechnen, ist bislang noch nicht besonders effizient. Zum Berechnen der Momente müssen alle Momente pro Pixel aufsummiert werden. Der Ansatz von Frese nutzt zur Verbesserung der Laufzeit aus, dass die Summe aller Momente jedes Teilbereichs der Region insgesamt das Moment der gesamten Region ist. Somit lässt sich die Ellipse in kleinere Teilbereiche aufteilen, auf welchen effizient die Momente berechnet werden können. In diesem Verfahren wird eine Aufteilung in Reihen genutzt, im Folgenden auch Intervalle genannt. Zu jeder Reihe muss lediglich die zugehörige y-Koordinate sowie die erste und letzte x-Koordinate gespeichert werden. Zwischen diesen x-Koordinaten sind automatisch alle Pixel in der Region enthalten. Diese Reihen müssen gruppiert werden, um entscheiden zu können, welche davon zu derselben Ellipse gehören. Für Intervalle, die sich in der y-Koordinate nur um einen Pixel unterscheiden muss hierzu geprüft werden, ob es in den zugehörigen x-Koordinaten eine Überschneidung gibt. Ist dies der Fall, so gehören die Intervalle zur selben Region.

Nun lassen sich auf jedem dieser Intervalle die Momente berechnen. Hierzu werden in der Arbeit von Frese Gleichungen für die Momentberechnung aufgestellt, welche nur auf den Grenzen der Intervalle arbeiten. Daher muss nicht mehr über alle Pixel iteriert werden, es wird lediglich eine Rechnung pro Intervall und Moment benötigt. Die expliziten Gleichungen und ihre Herleitungen sind für diese Arbeit allerdings nicht von Relevanz, weshalb sie hier nicht aufgeführt sind. Aus der Summe der so berechneten Momente über alle Intervalle lassen sich wie zuvor beschrieben die Ellipsenparameter bestimmen.

Die letzte offene Frage ist nun, wie entschieden wird, wo ein Intervall anfängt und wo es aufhört. Hierzu wird in dem vorgestellten Verfahren ein einfacher Schwellwert auf die Pixelhelligkeit angewendet. Er werden alle Pixel zeilenweise durchlaufen. Sobald ein Pixel dunkler als der angegebene Schwellwert ist, wird ein neues Intervall erstellt, mit dem aktuellen x-Wert als erste x-Koordinate des Intervalls und der Zeile als y-Koordinate. Daraufhin läuft das Verfahren weiter, bis es auf einen Pixel trifft, der über dem Schwellwert liegt, also nicht mehr schwarz ist. Der Vorgänger dieses Pixels ist die letzte x-Koordinate, die noch zu dem Intervall gehört und wird ebenfalls abgespeichert. Da die Parameter der Ellipse in späteren Schritten neu geschätzt werden, hat die Wahl des Schwellwerts keinen Einfluss auf die Genauigkeit, mit welcher Ellipsen erkannt werden. Wichtig ist, dass er dunkle und helle Bereiche korrekt trennt. In dieser Arbeit wurde bei einem Wertebereich der Helligkeit von $0 - 255$ ein Schwellwert von 100 genutzt. Das Resultat dieses Bildverarbeitungsschrittes ist anhand eines Beispiels aus einer Küchenszene in Abbildung 6.4 visualisiert. Zu sehen sind die erkannten Regionen sowie ihre Ellipsenparameter.

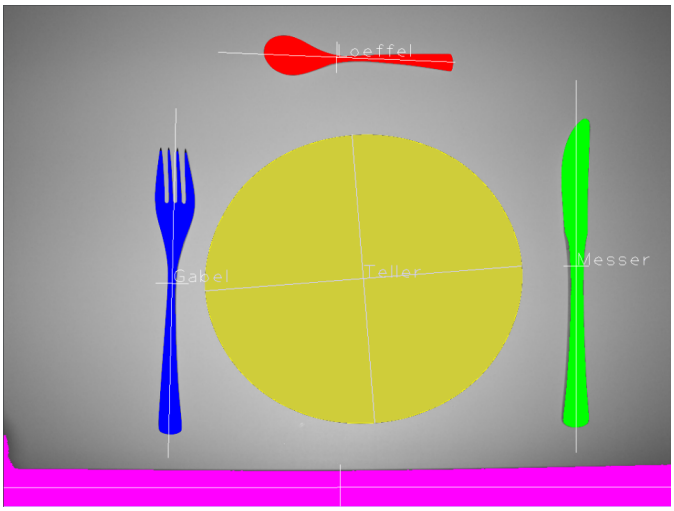

Abbildung 6.4: Erkannte Ellipsen
Alle Regionen, die unter den Schwellwert fallen sind farbig markiert, zusammengehörige Regionen haben die gleiche Farbe. Die berechneten Ellipsen sind anhand ihrer Hauptträgheitsachsen eingezeichnet [8].

6.1.3 Zentrale Ellipse bestimmen

Resultat dieses Schritts sind also alle in dem Bild enthaltenen dunklen Bereiche mit ihren Ellipsenparametern. Einer dieser Bereiche ist die gesuchte Ellipse. Der nächste Schritt in der Bilderkennung ist nun die Wahl des korrekten Bereichs. Diese Entscheidung wird getroffen, indem die Pixelposition der Ellipsen betrachtet wird. Der Kreismarker soll bei der Aufnahme immer möglichst nah am Mittelpunkt des Bildes sein. Demnach wird hier der Bereich gewählt, dessen Schwerpunkt am nächsten am Mittelpunkt des Bildes ist.

6.1.4 Parameter genau bestimmen

Zu diesem Zeitpunkt ist somit die Pixelposition des Kreismarkers im Bild bereits bestimmt. Auch seine Größe und sein Durchmesser sind bereits gegeben. In der Berechnung der Parameter blieb allerdings die Bildunschärfe unberücksichtigt, welche die Ränder des Markers leicht verschwimmen lässt. An Randbereichen kommt es dadurch zu unvorhersehbaren Ergebnissen. Abhängig von dem gewählten Schwellwert und der Bildhelligkeit, werden leicht verschwommene Randpixel entweder zu dem Kreis gezählt oder nicht. Dieser Effekt kann in der Berechnung der Momente berücksichtigt werden, um auch die unscharfen Bereiche korrekt dem Kreis zuzuordnen. Im nächsten Schritt wird deshalb auf der Ellipse eine optimierte Momentbestimmung ausgeführt, welche für diesen Zweck im Umfang der Arbeit entwickelt wurde.

Abbildung 6.5: Ausgeschnittene Bildregion

Zur exakten Bestimmung der Ellipsenparameter muss das Bild, auf dem gearbeitet wird, auf den relevanten Bildbereich reduziert werden. Hierzu wird das Bild zunächst zurechtgeschnitten. Anhand der zuvor geschätzten Ellipsenparameter lässt sich eine Bounding Box berechnen, welche in jedem Fall die gesamte Ellipse enthält. Der Mittelpunkt dieser Region ist der Mittelpunkt der Ellipse, während die Höhe und Breite genau der Ausdehnung der längeren Hauptträgheitsachse entsprechen. So ist sichergestellt, dass kein Teil der Ellipse abgeschnitten wird. Für die Berechnung der neuen Parameter wird im nächsten Schritt allerdings auch der weiße Rand des Markers benötigt. Die Höhe und Breite der Region wird daher um 30 Prozent vergrößert. Während zuvor nur der 5 Millimeter große schwarze Kreis in der Region enthalten war, sind nach dieser Vergrößerung auch 1.5 Millimeter weißer Rand enthalten.

Es kann in Ecken des Bildes dazu kommen, dass Bereiche außerhalb des weißen Randes gesehen werden. Diese Bereiche müssen entfernt werden, um ein Bild zu erhalten, welches nur den Kreismarker und seinen weißen Rand enthält. Zum Entfernen wird eine Maske in OpenCV angelegt. Diese Maske ist eine Art Binärbild, in welchem alle Pixel den Wert 1 haben, die behalten werden sollen. Alle anderen Pixel werden auf 0 gesetzt. Solch eine Maske lässt sich in OpenCV erstellen. Hierzu wird eine Funktion genutzt, welche Ellipsen in ein Bild einzeichnet. Dieser Funktion werden die zuvor berechneten Ellipsenparameter übergeben. Um wieder den weißen Rand zu behalten, werden auch hier die Ausdehnungen der Hauptträgheitsachsen um 30 Prozent vergrößert. Die Funktion setzt nun jeden Pixel, der in dieser 30 Prozent größeren Ellipse liegt, auf 1. Alle Pixel die außerhalb liegen haben den Wert 0. Diese Maske wird auf das Bild angewendet, wodurch ein neues Bild entsteht, welches lediglich den Kreismarker und seinen Rand enthält. Ein solches Bild ist in Abbildung 6.5 zu sehen.

Auf diesem Bild lassen sich nun die Momente korrekt bestimmen. Hierzu soll, wie

bereits angekündigt, die Unschärfe am Rand des Kreises korrekt erfasst werden, sodass die Länge der Hauptträgheitsachsen trotz der Unschärfe korrekt ist. Unschärfe entsteht dadurch, dass Licht von einem Punkt in der Welt nicht korrekt auf einen Punkt auf dem Sensor fokussiert wird, sondern auf einen Bereich um diesen Punkt herum streut. Im Kreisinneren hat dies keinen Effekt, da auch die umliegenden Bereiche schwarz sind, und daher durch Zufall trotzdem die korrekte Farbe dargestellt wird. Am Rand des Kreises streut aber zum Teil Licht aus dem schwarzen Bereich und aus dem weißen Bereich auf einen Punkt auf dem Sensor. Dadurch entstehen entsprechend Grautöne zwischen den beiden korrekten Farben. Pixel, die weiter in Richtung des schwarzen Bereichs liegen, werden dabei dunklere Grautöne haben, bis irgendwann kein Licht mehr aus dem weißen Bereich weit genug streut und der Pixel wieder schwarz ist. In die Richtung des weißen Bereichs gilt dasselbe, mit dem Unterschied, dass der Grauton immer heller wird. So ergibt sich am Rand des Kreises eine Kante, die gleichmäßig von schwarz nach weiß übergeht. Die echte Kante des Kreises liegt genau in der Mitte zwischen dem weißen und dem schwarzen Bereich.

Diese Beziehung zwischen der Helligkeit des Grautons und der Zugehörigkeit zu dem Kreis wird nun genutzt, um die Ellipsenparameter erneut zu schätzen. Während die Momente der Ellipse im vorherigen Verfahren anhand der Momente aller Intervalle geschätzt wurden, müssen die Momente in diesem Schritt pixelweise berechnet werden. Grund dafür ist, dass Pixel nicht mehr als 1 oder 0 betrachtet werden, je nachdem ob sie zu der Ellipse gehören oder nicht. Stattdessen wird für jeden Pixel ein Prozentsatz berechnet, mit welchem er zu der Ellipse gehört. Dieser Prozentsatz ergibt sich aus der prozentualen Helligkeit des Pixels. Ist der Pixel genau so hell wie das Innere des schwarzen Kreises, so ist der Prozentsatz 0 Prozent und der Pixel fließt als 1 in die Momentberechnung ein. Liegt der Pixel aber im unscharfen Bereich und ist bei 50 Prozent Helligkeit, so fließt der Pixel als 0.5 ein. Pixel, die zu 100 Prozent weiß sind, fließen entsprechend gar nicht in die Momentberechnung ein, da sie kein Teil des schwarzen Kreises mehr sind.

Zur Berechnung der prozentualen Helligkeit wird die durchschnittliche innere und äußere Helligkeit des Kreises gemessen. Hierzu werden auf einer Ellipse im Inneren, schwarzen Bereich alle Helligkeiten der Pixel gemittelt. Dasselbe wird für den äußeren, weißen Bereich getan. Um zu bestimmen, wo die Ellipse verlaufen muss um nur schwarze oder weiße Punkte zu treffen, werden wieder die zuvor bestimmten Parameter genutzt. Um schwarze Pixel zu treffen, wird die Ausdehnung der Hauptträgheitsachsen um 40 Prozent reduziert, während sie um 20 Prozent vergrößert wird, um eine Ellipse im weißen Bereich zu erhalten. Die durchschnittliche weiße Helligkeit wird im Folgenden mit l_w beschrieben, während die schwarze Helligkeit mit l_b abgekürzt wird. Zur Berechnung der prozentualen Helligkeit p eines Pixels mit der Helligkeit l_p wird wie folgt vorgegangen:

$$p = \frac{1}{l_w - l_b}(l_p - l_b)$$

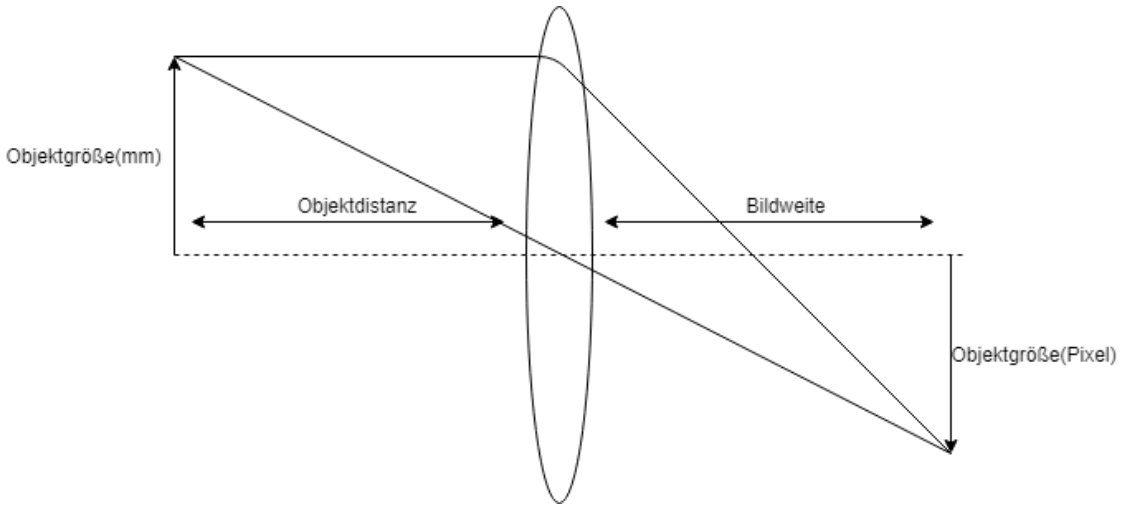

Abbildung 6.6: Modell einer Linse

Zur Berechnung der Momente auf dem Bild wird über alle Pixel iteriert. Dies ist möglich, da zuvor alle Teile des Bildes, die nicht zum Marker gehören, aus dem Bild entfernt wurden. Zu jedem Pixel werden wie zuvor beschrieben die Momente aufaddiert. Da Pixel aber nur zu einem Teil zum Kreis gehören können, müssen die addierten Werte jeweils mit $p(x, y)$ multipliziert werden, also der zuvor berechneten prozentualen Helligkeit des Pixels x, y. Daraus ergibt sich die folgende Formel:

$$I_{x^i y^j} = \int_{x,y \in \Omega} p(x, y) \cdot x^i \cdot y^j dx dy$$

Mit diesen Momenten lassen sich die Parameter der Ellipse wie zuvor bestimmen. So ergibt sich eine Ellipse, deren Kante genau in der Mitte des unscharfen Bereichs verläuft. Zu der Ellipse sind nun also der Mittelpunkt und die Längen der beiden Hauptträgheitsachsen bekannt. Aus diesen Informationen ist es möglich, die Position des Kreismarkers im Koordinatensystem der Kamera zu berechnen. Hierzu wird das Linsenmodell für dünne Linsen genutzt, welches in Abbildung 6.6 visualisiert ist. Zur Berechnung der Objektdistanz d in Millimetern wird ausgenutzt, dass das Dreieck, welches von der Objektgröße im Bild o_i und der Bildweite i definiert wird, zwei gleiche Winkel zu dem Dreieck hat, welches durch die Objektdistanz und die Objektgröße o_w in der Welt gegeben ist. Dementsprechend lässt sich die folgende Formel aufstellen und nach d auflösen:

$$\frac{d}{i} = \frac{o_w}{o_i}$$
$$d = i\frac{o_w}{o_i}$$

In dieser Formel sind alle Parameter bekannt, die benötigt werden, um d zu berechnen. Die Bildweite i ist in der Kalibrierung der Kamera bestimmt worden. Die Größe o_w des Objekts in der Welt ist die echte Größe des Markers, also 5 Millimeter. Die Größe o_i des Objekts im Bild ist gegeben durch die Ausdehnung der längeren Hauptträgheitsachse der erkannten Ellipse. Abhängig vom Kamerawinkel wird die kürzere Hauptträgheitsachse bei steileren Winkeln immer kleiner. Die längere Achse behält aber immer die gleiche Länge, solange sich die Distanz der Kamera zum Ellipsenmittelpunkt nicht ändert. Diese Länge wäre der Durchmesser des Kreises, wenn die Kamera aus der selben Distanz direkt von oben auf den Kreis schauen würde. Damit ist sie die korrekte Größe für den Wert o_i.

6.1.5 In Position umrechnen

Mit der berechneten Distanz des Markers zur Kamera lässt sich nun die dreidimensionale Position des Markers im Kamerakoordinatensystem berechnen. Hierzu kommen wieder die kalibrierte Bildweite f_x, f_y und der Bildmittelpunkt c_x, c_y zum Einsatz. Weiterhin wird der Schwerpunkt der Ellipse x_s, y_s benötigt. Die Koordinaten werden wie folgt berechnet:

$$
\begin{aligned}
x &= \frac{d}{f_x}(x_s - c_x) \\
y &= \frac{d}{f_y}(y_s - c_y) \\
z &= d
\end{aligned}
$$

Damit ist die Berechnung der Markerposition abgeschlossen. Das Smartphone bietet nun die Möglichkeit, die Kreismarker an einem Objekt aufzunehmen und ihre Position im Koordinatensystem der Smartphonekamera zu berechnen. Ein erkannter Kreismarker mit seiner berechneten Distanz ist in Abbildung 6.1 zu sehen.

6.2 Marker2D

Das System Marker2D, welches Ceres zur Berechnung der Objektposen nutzt, bietet eine Alternative zu dem Ansatz, auf den Positionen der Kreismarker in der Welt zu arbeiten. Wie sich in der Evaluation der Kreismarkererkennung zeigen wird, ist die Berechnung der Distanz der Marker deutlich ungenauer als die Erkennung ihrer Pixelkoordinaten. Daher wurde in dieser Arbeit das Verfahren Marker2D entwickelt, welches lediglich die Pixelpositionen der drei Marker im Bild des Smartphones benötigt, um daraus eine Objektpose zu berechnen. Zusätzlich benötigt dieses Verfahren die Information, wo sich die Smartphonekamera zu dem Zeitpunkt der Aufnahme befunden hat.

Die Beschaffung dieser Informationen lässt sich mit den bisher vorgestellten Methoden leicht umsetzen. Die Pixelpositionen der Marker im Bild lassen sich mit der zuvor

beschriebenen Kreismarkererkennung berechnen. Die Pose der Smartphonekamera lässt sich aus der Transformation des Targets in der Welt sowie der kalibrierten Transformation zwischen Target und Smartphone berechnen. Somit sind bereits alle Informationen gegeben, welche Ceres benötigt, um daraus eine Objektpose in der Welt zu berechnen.

6.3 Taststab

Das Taststab System nutzt einen an dem Smartphone angebrachten Metallstab, mit welchem die Punkte an dem Objekt berührt werden, um so ihre Positionen zu bestimmen. Hierzu wird in diesem Kapitel eine weitere Ceres-Kalibrierung vorgestellt, welche die Position der Metallstabspitze im Koordinatensystem des Tracking-Targets bestimmt. Mithilfe dieser Kalibrierung ist es möglich, von der Pose des Tracking-Targets direkt auf die Position der Spitze im Weltkoordinatensystem zu schließen.

Der Pointer bietet im Gegensatz zu allen anderen Verfahren den Vorteil, dass er unabhängig von Markern an dem Objekt ist. Es wird lediglich eine beliebige Markierung an dem ausgemessenen Punkt auf dem Objekt benötigt, um genau zu wissen, welcher Punkt ausgemessen wurde. Beispielsweise können Objektpunkte auch mit einem Permanentmarker markiert werden. Insbesondere für Objekte mit wenig Fläche, auf der die Kreismarker angebracht werden können, ist dieser Ansatz eine gute Alternative, um die Objekte trotzdem aufnehmen zu können.

Problematisch ist an dem Ansatz, dass er stark von dem Annotator abhängig ist. Um die Punkte zu treffen, benötigt er eine ruhige Hand. Wird beim Aufnehmen zu stark gewackelt, kann es dazu kommen, dass entweder falsche Punkte aufgenommen oder sogar die Objekte verschoben werden. Es muss also vom Annotator abgewogen werden, für welche Objekte sich der Einsatz lohnt.

Die Kalibrierung der Position der Metallstabspitze nutzt eine ähnliche Idee wie die Kalibrierung der Transformationen zwischen Sensoren und ihren Targets. Auch hier wird ein Punkt wiederholt aus verschiedenen Perspektiven aufgenommen. Aufgenommen heißt in dem Fall, dass der Punkt von der Metallstabspitze berührt wird und der Annotator dies im Smartphone-Interface bestätigt. Zum Zeitpunkt der Bestätigung wird jeweils die Pose des Smartphone-Targets in der Welt gespeichert. Diese Pose ist die Observation, die Ceres benötigt, um von der Stabspitze auf einen Punkt in der Welt zu schließen. Die Kalibrierung versucht, anhand dieser Observationen eine bestimmte Position im Koordinatensystem des Targets zu finden. Diese Position soll für alle Observationen die Eigenschaft haben, dass sie nach der Transformation in das Weltkoordinatensystem denselben Punkt in der Welt beschreibt. Dieser Punkt im Koordinatensystem des Targets ist entsprechend die Spitze des Metallstabs. Da sich diese Spitze im Verhältnis zum Target immer am gleichen Ort befindet, kann dieser Punkt immer genutzt werden, um von der Pose des Targets auf die Position der Spitze zu schließen.

Ceres benötigt für die Kalibrierung wie zuvor die Information, welche Parameter

$p_p^{(ST)}$	initial	kalibriert
x	60	87.6452
y	200	−202.316
z	0	−12.2078

Tabelle 6.1: Kalibrierungsergebnisse des Pointers

kalibriert werden sollen und wie die Residuen zu berechnen sind. Die Parameter sind für diese Kalibrierung deutlich reduziert. Statt wie zuvor eine ganze Transformation zu kalibrieren, wird für diese Kalibrierung lediglich ein Punkt im Koordinatensystem des Targets kalibriert, welcher nach der Kalibrierung die Spitze des Metallstabs darstellen soll. Weiterhin wird auch in dieser Kalibrierung der Punkt in der Welt mit kalibriert, welcher mit dem Pointer aufgenommen wird. Damit ergeben sich sechs zu bestimmende Werte: Die Position $p_m^{(W)}$ des Punktes in der Welt sowie die Position $p_p^{(ST)}$ der Spitze des Pointers im Koordinatensystem des Targets. Demnach ist das von Ceres zu schätzende Modell x wie folgt gegeben:

$$x = (p_p^{(ST)}, p_m^{(W)})$$

Die Residuen werden berechnet, indem die Stabspitze $p_p^{(ST)}$ in das Weltkoordinatensystem transformiert wird. Dort wird der Punkt mit der Position des Markers in der Welt verglichen. Die Transformation wird für den i-ten Punkt wie folgt durchgeführt:

$$p_{pi}^{(W)} = T_{W \leftarrow ST_i} \cdot p_p^{(ST)}$$

Daraus ergeben sich wie folgt Residuen für Ceres:

$$r_i(x) = p_{pi}^{(W)}(x) - p_m^{(W)}$$

Damit sind für Ceres alle Informationen gegeben. Die Kalibrierung wurde in dieser Arbeit mit etwa 50 Observationen durchgeführt. Das Ergebnis ist in Tabelle 6.1 aufgeführt. Der kalibrierte Punkt in der Welt ist dabei wie zuvor nicht mit aufgeführt, da er lediglich eine Hilfsgröße für das Kalibrierungsverfahren ist. Aufgeführt sind die x-, y- und z-Koordinate der Stabspitze im Koordinatensystem des Targets.

Der durchschnittliche Reprojektionsfehler dieser Kalibrierung liegt bei 0.43 Millimetern. Mithilfe dieser Position der Stabspitze im Target lassen sich nun mit der Formel $p_p^{(W)} = T_{W \leftarrow ST} \cdot p_p^{(ST)}$ die Objektpunkte berechnen.

6.4 MarkerRef

Das Verfahren MarkerRef nutzt Kreismarker, die aus reflektierender Folie ausgeschnitten wurden, um die Punkte auf dem Objekt zu bestimmen, wie in Abbildung 4.5 zu sehen ist. Diese reflektierenden Kreismarker werden von dem Tracking System aufgenommen, weshalb ihre Position direkt im Weltkoordinatensystem bekannt ist. Dadurch liefert dieses Verfahren sehr genaue Positionen. Diese werden in dieser Arbeit unter anderem dazu genutzt, die Genauigkeit der anderen Verfahren zu evaluieren. Da die Punkte direkt im Weltkoordinatensystem aufgenommen werden, wird für dieses Verfahren keine weitere Berechnung benötigt. Alle Daten werden direkt von dem Tracking System geliefert. Zu beachten ist allerdings, dass die Punkte in diesem Verfahren alle gleichzeitig aufgenommen werden, da das Tracking System alle Punkte im Raum zur gleichen Zeit sieht. Bei allen anderen Verfahren ist die Zuordnung der aufgenommenen Punkte zu Punkten auf dem Objekt dadurch gegeben, dass die Punkte in einer bestimmten Reihenfolge aufgenommen werden. Diese Zuordnung ist in diesem Verfahren nicht gegeben. Stattdessen werden die erkannten Positionen der Marker nach ihren Distanzen zu den beiden anderen Punkten sortiert. Hierfür werden für jeden Punkt auf dem Objekt die zwei Distanzen zu den anderen Punkten berechnet. Dies wird sowohl für die ausgemessenen Punkte als auch für die vom Tracking System erkannten Punkte getan. Ein Punkt des Tracking Systems wird dann einem Punkt auf dem Objekt zugewiesen, wenn er die gleichen Distanzen zu den zwei anderen Punkten hat.

Während sich dieses System durch seine präzise Punktbestimmung auszeichnet, hat es ein Problem, welches es nur in sehr wenigen Situationen anwendbar macht. Die Marker dürfen nicht zu stark zu den Infrarotkameras des Tracking Systems geneigt sein. Dies führt dazu, dass alle Marker nahezu auf einer Ebene platziert werden müssen, welche von den Kameras gut gesehen wird. Daher ist es für viele Objekte, die keine solche Ebene haben, nicht einsetzbar. Für Objekte, die eine solche Ebene haben, ist es aber ein sehr nützliches Verfahren, da alle Marker auf einmal aufgenommen werden.

6.5 Evaluation

Nachdem nun alle Verfahren zur Punktaufnahme vorgestellt wurden, sollen diese Verfahren auf ihre Genauigkeit geprüft werden. Hierzu werden im Folgenden die einzelnen Verfahren evaluiert.

6.5.1 Marker3D

Die Evaluation der Kreismarkererkennung soll feststellen, wie sich die Genauigkeit des Ansatzes in verschiedenen Testsituationen verhält. Die folgenden Testreihen wurden in der Evaluation durchgeführt:

- Erkennung des Markers bei sich ändernden Lichtverhältnissen
- Erkennung des Markers in einer statische Szene
- Erkennung des Markers bei Änderung der Markerposition
- Erkennung des Markers bei Neigung des Markers

Jede dieser Testreihen sollen verschiedene Eigenschaften der Kreismarkererkennung testen. Die Tests in einer statischen Szene bestimmen, wie stark die Erkennungsergebnisse von Rauschen, also minimalen Änderungen im Kamerabild, beeinflusst werden. Eine Änderung der Lichtverhältnisse kann insbesondere in Anwendungen, die mit Helligkeitswerten arbeiten, zu Fehlern in der Erkennung führen. Die Änderung der Markerposition soll die Genauigkeit bestimmen, mit welcher der Punkt im Koordinatensystem der Kamera bestimmt wird, während das Neigen des Markers betrachtet wird, um sicherzustellen, dass die Marker aus verschiedenen Perspektiven aufgenommen werden können.

Zur Evaluation der Kreismarkererkennung wird der Kreismarker wiederholt in diesen Situationen aufgenommen und die erkannte Position im Kamerakoordinatensystem überprüft. Abschließend wird die berechnete Position des Markers im Weltkoordinatensystem evaluiert, also nachdem die Transformation $T_{W \leftarrow S} = T_{W \leftarrow ST} \cdot T_{ST \leftarrow S}$ auf den Punkt angewendet wurde.

Die erste durchgeführte Testreihe untersucht das Verhalten der Kreismarkererkennung unter verschiedenen Lichteinflüssen. Hierzu wird die Belichtungszeit der Kamera konstant gesetzt und die Belichtung des Markers mit einer Taschenlampe variiert. Die Distanz der Kamera zum Marker wird dabei konstant gehalten, indem das Smartphone in eine Halterung eingespannt wird, wie in Abbildung 6.7 zu sehen ist.

Der Versuch wird durchgeführt, indem das Smartphone etwa eine halbe Minute kontinuierlich die Position eines Kreismarkers berechnet. In dieser Zeit wird mit der Taschenlampe die Belichtung variiert. Zu dem erkannten Kreismarker wird seine Distanz zur Kamera sowie die Helligkeit des Markerrandes gespeichert. Diese Helligkeit wird als Indikator

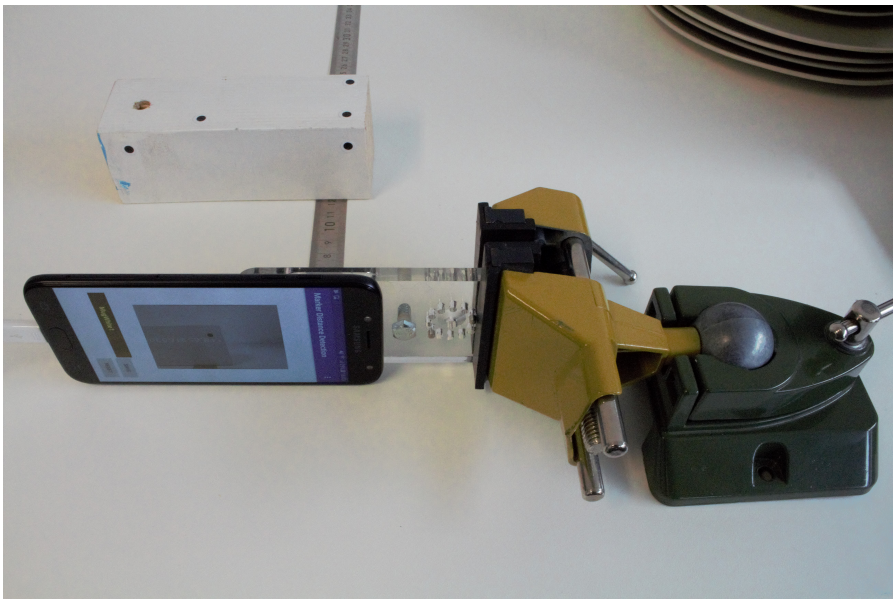

Abbildung 6.7: Aufbau der Evaluation des Smartphones

dafür genutzt, wie stark das Bild aktuell beleuchtet wird. Das Ergebnis dieses Versuchs ist in Abbildung 6.8 zu sehen.

Wie in dieser Abbildung zu sehen ist, hat die Helligkeit des Markers einen starken Einfluss auf die erkannte Distanz. Dieser Effekt ist allerdings bereits im Kamerabild zu beobachten, bevor der erste Schritt der Kreismarkererkennung ausgeführt wird. Bei wenig Licht ist der unscharfe Rand insgesamt dunkler als bei viel Licht, auch nachdem die Helligkeit auf den Bereich zwischen der Helligkeit des schwarzen Markers und der des weißen Randes normalisiert wurde. Dies führt dazu, dass der Kreis im Bild bei wenig Licht größer ist als bei viel Licht. Ein größerer Kreis im Bild bei konstanter Größe in der Welt bedeutet aber, dass der Kreis näher an der Kamera sein muss. Dadurch entsteht der beobachtete Effekt. Da er aber bereits in der Erstellung des Bildes in der Kamera entsteht, muss der Effekt nachträglich entfernt werden. Hierzu wird eine Gerade durch alle Messpunkte gelegt, wie in Abbildung 6.8 zu sehen ist. Diese Gerade gibt an, wie stark die berechnete Distanz ansteigt, abhängig von der aktuellen Helligkeit des Markers. Um den Effekt umzukehren, wird in der Implementierung die Distanzberechnung angepasst. Die Distanz wird zunächst einmal wie zuvor berechnet, und dann um einen Faktor angepasst, welcher sich aus der Geradengleichung und der Helligkeit des Markers ergibt. Alle Distanzen werden auf die Helligkeit 160 normalisiert, was in etwa der durchschnittlichen Helligkeit in dem Versuch entspricht. Hierzu wird der prozentuale Anstieg der Distanz in der Ausgleichsgeraden zwischen der aktuellen Markerhelligkeit und einer Helligkeit von 160 berechnet. Dieser prozentuale Anstieg wird als Korrekturfaktor auf die berechnete

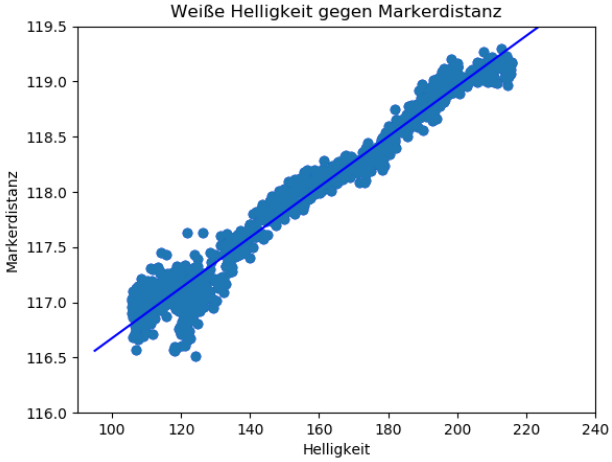

Abbildung 6.8: Beziehung zwischen Helligkeit des Markers und der berechneten Distanz In dieser Grafik ist die Distanz in Millimetern zwischen Marker und Kamera gegen die Helligkeit des Markerrandes aufgetragen, wobei die Helligkeit insgesamt zwischen 0 und 255 liegt. Höhere Werte beschreiben hellere Punkte. Die Gleichung der Ausgleichsgeraden ist $y = 114.394561 + 0.0228116369 \cdot x$.

Distanz angewendet.

Um die Effektivität dieser Korrektur zu evaluieren, wurde nach der Korrektur ein zweiter Datensatz nach dem oben beschriebenen Verfahren aufgenommen. Das Ergebnis ist in 6.9 aufgetragen.

Wie dort zu sehen ist, ist der Effekt nun nicht mehr zu beobachten. Die Korrektur der Distanz wird dementsprechend in die Markererkennung mit aufgenommen und für jeden erkannten Marker durchgeführt. Auch in den folgenden Evaluationen sind die aufgeführte Distanzen immer mit diesem Verfahren korrigiert.

Die nächste Evaluation soll zeigen, wie stark das entwickelte Verfahren von Rauschen des Kamerasensors beeinflusst wird, solange sich in der Szene sonst nichts verändert. Es soll also geprüft werden, in welchen Wertebereichen die berechneten Parameter liegen. Die betrachteten Parameter sind dabei der Mittelpunkt der Ellipse in Pixeln sowie die Position des Markers im Kamerakoordinatensystem. Hierzu wird das Smartphone wie zuvor eingespannt und ein Marker im Sichtfeld der Kamera platziert. Die Szene wird einige Sekunden lang aufgenommen und die erkannten Parameter gespeichert. Diese Parameter sind in Abbildung 6.10 gegeneinander aufgetragen.

Wie in den Abbildungen zu sehen ist, streut die erkannte Ellipsenmitte in der y-Achse um etwa 0.1 Pixel, während in der x-Achse eine Streuung von 0.16 Pixeln zu beobachten ist. Die Koordinaten im Kamerakoordinatensystem weisen ebenfalls eine gewissen Streuung auf. Auf der x- und y-Achse liegt diese jeweils bei circa 0.025 Millimetern. Auf der

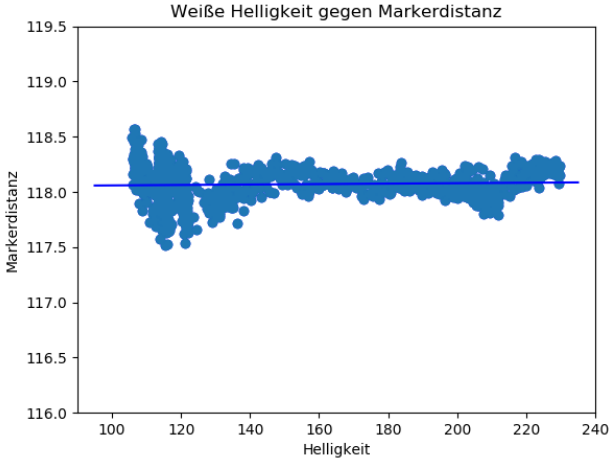

Abbildung 6.9: Beziehung zwischen Helligkeit des Markers und der korrigierten Distanz In dieser Grafik ist die Beziehung zwischen Distanz und Helligkeit aufgetragen, nachdem die Distanz korrigiert wurde.

z-Achse ist die Streuung etwas größer, weshalb sie in der Abbildung etwas genauer anhand ihrer Gaußverteilung evaluiert wurde. Daraus ergibt sich eine Standardabweichung von 0.065 Millimetern bei einem Mittelwert von 118.57 Millimetern. Insgesamt sind diese Streuungseffekte so gering, dass sie einen vernachlässigbaren Einfluss auf die Erkennungsgenauigkeit der Kreismarker haben.

In dieser Evaluation ist allerdings ein interessanter Effekt zu beobachten. Die x- und y-Koordinaten weisen einen korrelierten Fehler auf, mit einer Abhängigkeit zwischen den beiden Parametern. Direkt nach dem Fokussieren der Kamera ist zu beobachten, dass der Fokus sich über einen kurzen Zeitraum weiterhin minimal ändert. Grund dafür könnte etwas Spielraum im Fokussiermechanismus und die dadurch entstehende Änderung der Brennweite sein. Eine Änderung der Brennweite würde sowohl die x- als auch die y-Koordinate beeinflussen, was den beobachteten Effekt erkären könnte. Dies ist allerdings lediglich eine Vermutung und wurde nicht untersucht. Bei Streuungen von bis zu 0.16 Pixeln ist der Effekt gering genug, um ihn in dieser Arbeit nicht weiter zu berücksichtigen.

In der nächsten Evaluation soll bestimmt werden, wie korrekt die Distanz des Markers berechnet wird. Auch hier wird wieder ein ähnlicher Versuchsaufbau wie zuvor gewählt. In Abbildung 6.7 war bereits ein Lineal zu sehen, welches in den vorigen Evaluationen noch nicht benötigt wurde. Dieses Lineal ist orthogonal zu dem Smartphone befestigt und wird in dieser Evaluation genutzt, um die echte Distanz zwischen dem Marker und der Smartphonekamera zu messen. Problematisch ist hierbei, dass die Distanz von der Kamera ab dem Kamerasensor gemessen wird. Dieser ist an einem unbekannten Ort in dem Smartphone verbaut, weshalb das Lineal nicht exakt auf der gleichen Höhe angelegt

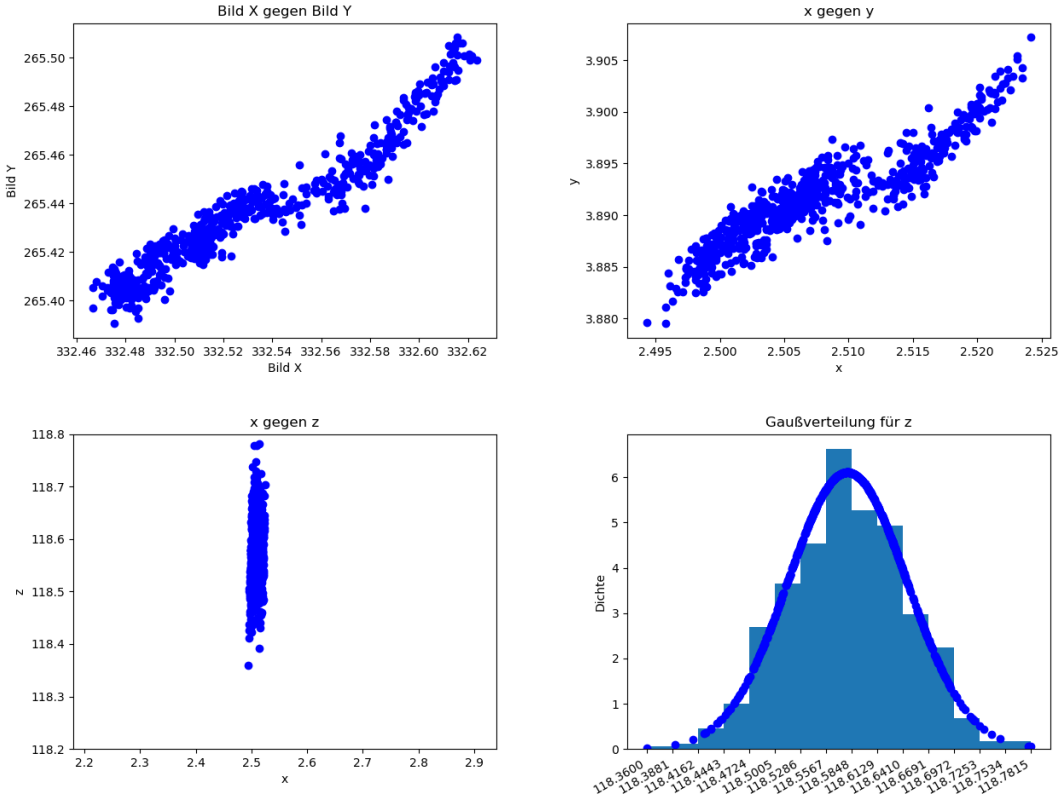

Abbildung 6.10: Streuung der Ellipsenparameter
Gegeneinander aufgetragen sind die Bildkoordinaten der Ellipse sowie die x-, y- und z-Koordinaten des Punktes. Zu der z-Koordinate ist zusätzlich die Gausverteilung abgebildet.

werden kann. Stattdessen wird für diese Evaluation das Lineal so angelegt, dass die berechnete und die gemessene Distanz eines 135 Millimeter entfernten Markers exakt übereinstimmen. Bei einer korrekten Distanzmessung müsste sich nun bei einer Änderung der Markerdistanz die berechnete Distanz genau so verändern wie die gemessene Distanz.

Diese Veränderung wurde evaluiert, indem der Marker auf dem Lineal in 5 Millimeter Intervallen weiter vom Sensor weg geschoben wurde. Die erste betrachtete Distanz liegt bei 80 Millimetern, die letzte bei 170. In Tabelle 6.2 sind die gemessene sowie die berechnete Distanz aufgetragen.

Wie in dieser Tabelle zu sehen ist, werden im Bereich zwischen 95 und 160 Millimetern Distanzabweichungen von unter einem Millimeter beobachtet. Außerhalb dieses Bereichs werden die berechneten Distanzen deutlich ungenauer. Die empfohlene Arbeitsentfernung zum Aufnehmen der Marker liegt daher in dem Bereich zwischen 95 und 160 Millimetern.

gemessen	80	85	90	95	100	105	110	115	120	125
berechnet	81.6	86.2	91.2	95.7	100.6	105.6	110.3	115.3	120.2	125.1
gemessen	130	135	140	145	150	155	160	165	170	
berechnet	130.2	135.0	139.9	145.3	150.4	155.6	160.8	166.0	171.5	

Tabelle 6.2: Gemessene und berechnete Distanzen in Millimetern

Dieser Abstand wird in allen in dieser Arbeit aufgenommenen Datensätzen eingehalten.

Nachdem die Genauigkeit der Distanzberechnung bestimmt wurde, muss noch geprüft werden, ob auch bei Verschiebungen im Bild der Punkt korrekt berechnet wird. Hierfür wird ein ähnlicher Versuchsaufbau gewählt, bei welchem der Punkt nicht in der Tiefe verschoben wird, sondern in etwa entlang der x-Achse. Der Punkt wird wieder in 5 Millimeter Intervallen entlang eines fixierten Lineals verschoben, welches nun aber parallel zum Smartphone angebracht wird. Zu jedem aufgenommenen Punkt wird die Distanz zum vorherigen Punkt berechnet. Werden die Koordinaten der Punkte korrekt bestimmt, so sollte auch die Distanz zwischen diesen Punkten 5 Millimeter betragen. Das Ergebnis dieses Versuchs ist in Abbildung 6.11 abgebildet.

Die Evaluation der Genauigkeit der bestimmten Koordinaten zeigt, dass auch die berechneten Punkte immer in etwa einen Abstand von 5 Millimetern zueinander haben. Anhand der Distanz zur Geraden zwischen dem ersten und letzten Punkt lässt sich schätzungsweise Ablesen, ob die einzelnen Punkte korrekte Distanzen zueinander haben. Diese Distanzen liegen zwischen 0 und 0.4 Millimetern, was darauf hindeutet, dass Verschiebungen im Bild korrekt aufgenommen werden. Wie sich auf der roten y-Achse zeigt, ist die Gesamtdistanz des letzten Punktes zum ersten mit 74 Millimetern sehr nah an den gemessenen 75 Millimetern, was dafür spricht, dass auch größere Distanzen zwischen Punkten korrekt abgebildet werden.

Die letzte Evaluation betrachtet, wie robust die Distanzberechnung gegenüber Neigungen des Markers ist. Zur Berechnung der Distanz wurde angenommen, dass die längere Hauptträgheitsachse als Größe der Ellipse genutzt werden kann, da sie den Durchmesser eines gleichgroßen Kreises in der gleichen Distanz beschreibt. Diese Annahme wird durch diese Evaluation überprüft. Hierzu wurde ein Marker an einem frei rotierbaren Holzblock angebracht. Von diesem Marker aus wurde mit einem Winkelmaß ein Strich gezogen, welcher in einem 90 Grad Winkel zum Tisch verläuft und den Marker genau in der Mitte trifft. Dieser Strich definiert nun eine Achse, um welche der Marker gedreht werden kann, ohne seine Position zu ändern. Um beim Rotieren des Blocks die Position des Markers nicht zu verändern wurde zusätzlich der Ort gekennzeichnet, wo der Strich auf den Tisch trifft. Solange diese Kennzeichnung und der Strich sich berühren, befindet sich der Marker an der richtigen Position. Zur Evaluation wurde der Marker zunächst gerade zur Kamera ausgerichtet und in dieser Position ausgemessen. Die so aufgenommene Distanz liefert

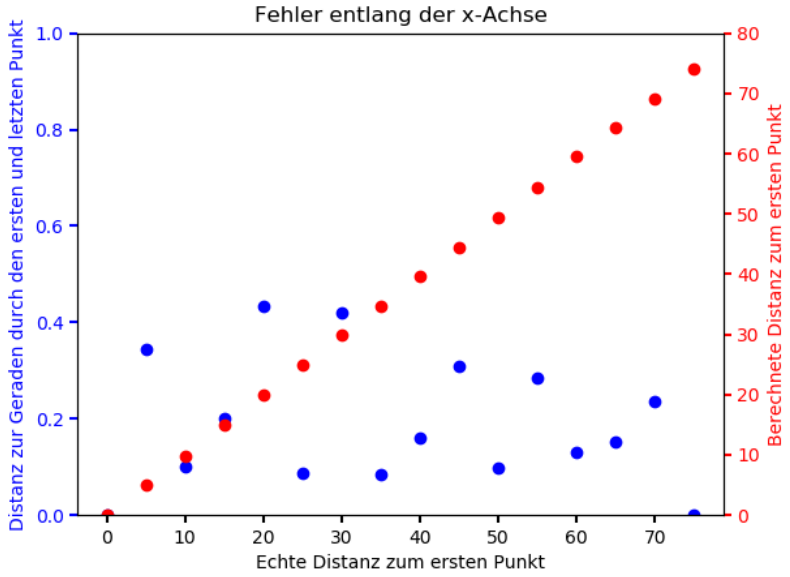

Abbildung 6.11: Distanzen des Markers zum erwarteten Punkt(Blau) und zum ersten Punkt(Rot)
Auf der x-Achse ist die echte Distanz des aktuellen Punktes zum ersten Punkt aufgetragen. Die blaue y-Achse gibt die Distanz zur Geraden an, die vom ersten berechneten Punkt zum letzten verläuft, während in Rot die berechnete Distanz zum ersten Punkt aufgetragen ist.

den Vergleichswert für die Aufnahmen mit rotierten Markern. Anschließend wurde der Marker in zehn Grad Intervallen rotiert, bis zu einem Winkel von 40 Grad. In steileren Winkel werden die Kreismarker nicht mehr erkannt. Die Ergebnisse sind in Tabelle 6.3 aufgeführt.

Wie in dieser Tabelle zu sehen ist, ist die Annahme gültig, dass die längere Hauptträgheitsachse als Größe genutzt werden kann. Selbst bei starken Neigungen von 40 Grad liegt der Fehler in der Distanzberechnung bei nur 0.7 Millimetern. Ein etwas ungenaueres Ergebnis ist bei starken Neigungen zu erwarten, da die Fläche der Ellipse immer weiter reduziert wird, je stärker der Marker geneigt ist. Die Genauigkeiten liegen aber auch für starke Neigungen noch in einem akzeptablen Bereich.

Damit ist die Evaluation der Kreismarkererkennung vollständig. Sie wurde auf ihre Genauigkeit unter verschiedenen Einflüssen geprüft. Hierzu zählen unterschiedliche Lichteinflüsse, Verschiebungen auf verschiedenen Achsen und Rotationen des Markers. Es konnte für alle betrachteten Effekte gezeigt werden, dass die Position des Markers trotz dieser Effekte auf unter einen Millimeter genau bestimmt werden kann, wenn sie mit den

Winkel	−40°	−30°	−20°	−10°	0°	10°	20°	30°	40°
Distanz	118,2	117,8	117,8	117,9	117,8	118	118,1	118,5	118,5

Tabelle 6.3: Gemessene Winkel in Grad und berechnete Distanzen in Millimetern

jeweiligen Referenzpositionen verglichen wird.

Abschließend muss evaluiert werden, wie genau die Punkte im Weltkoordinatensystem sind. Hierzu müssen die Punkte vom Kamerakoordinatensystem in das Weltkoordinatensystem transformiert werden. Die hierfür genutzte Transformation $T_{W\leftarrow S} = T_{W\leftarrow ST} \cdot T_{ST\leftarrow S}$ setzt sich zusammen aus der vom Tracking System bestimmten Transformation $T_{W\leftarrow ST}$ und der kalibrierten Transformation $T_{ST\leftarrow S}$. Die Evaluation wird durchgeführt, indem ein fest platzierter Marker 50 Mal aus verschiedenen Winkeln und Distanzen mit dem Smartphone aufgenommen wird. Als Referenz wird nach der Durchführung reflektierende Folie auf dem Marker angebracht. Dadurch wird der Punkt von dem Tracking System aufgenommen. Die Distanz zu diesem Referenzpunkt liefert den Fehler der berechneten Markerposition im Weltkoordinatensystem. Diese Fehler der Markerpositionen sind in Abbildung 6.12 aufgeführt.

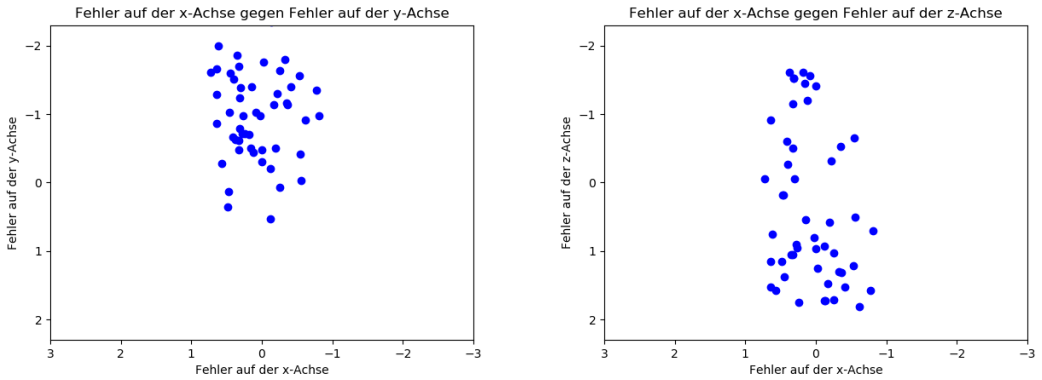

Abbildung 6.12: Marker3D-Fehler auf der x-, y- und z-Achse
Distanzen auf der x-, y- und z-Achse, gegeneinander aufgetragen.

Wie in diesen Grafiken zu sehen ist, liegen die Fehlerwerte nach der Transformation etwas höher als zuvor. Die Fehler auf der x-Achse bewegen sich zwischen 0.8 und -0.8 Millimetern. Auf der y-Achse liegen die Fehler zwischen 0.5 und -2 Millimetern. Auf der z-Achse sind Abweichungen von 1.6 bis -1.6 Millimetern zu beobachten. Weiterhin ist zu sehen, dass die Punkte auf der y-Achse nicht um einen Fehler von 0 Millimeter verteilt sind, sondern in etwa um -1. Solch ein konstanter Fehler ist möglicherweise eine Folge des Überklebens des Markers mit der reflektierenden Folie. Hierbei kann es passieren, dass die Folie nicht exakt platziert wird.

Die Abweichung der berechneten Punkte zu dem Referenzpunkt wird wieder anhand des RMS-Werts evaluiert. Hierzu wird zu jedem Punkt die euklidische Distanz zu dem Referenzpunkt berechnet und auf diesen Distanzen der RMS berechnet. So ergibt sich für diese Evaluation eine Abweichung von 1.69 Millimetern. Eine solche Genauigkeit ist selbst für die restriktivste Metrik 0.5 Grad 0.5 Zentimeter ausreichend. Das Verfahren ist daher zum Aufnehmen von Punkten auf den Objekten nutzbar.

6.5.2 Marker2D

Wie zuvor bereits erwähnt, nutzt Ceres für die Posenbestimmung die in der Kreiserkennung bestimmten Positionen der Ellipse. Daher ist die Evaluation der berechneten Punkte bereits in der Evaluation der Kreismarkererkennung enthalten. Interessant ist für Ceres die Evaluation der Posenbestimmung, welche in dem an dieses Kapitel anschließenden Abschnitt vorgestellt wird.

6.5.3 Taststab

Für den Taststab soll, ähnlich wie für die Kreismarkererkennung, evaluiert werden, wie genau die Punkte im Weltkoordinatensystem aufgenommen werden. Hierzu wird ein ähnlicher Testaufbau verwendet, wie für die Kreismarkererkennung. Es wird ein Punkt 50 Mal aufgenommen, welcher anschließend mit der reflektierenden Folie beklebt wird, um seine echte Position in der Welt zu erhalten. Die Ergebnisse sind in Abbildung 6.13 zu sehen.

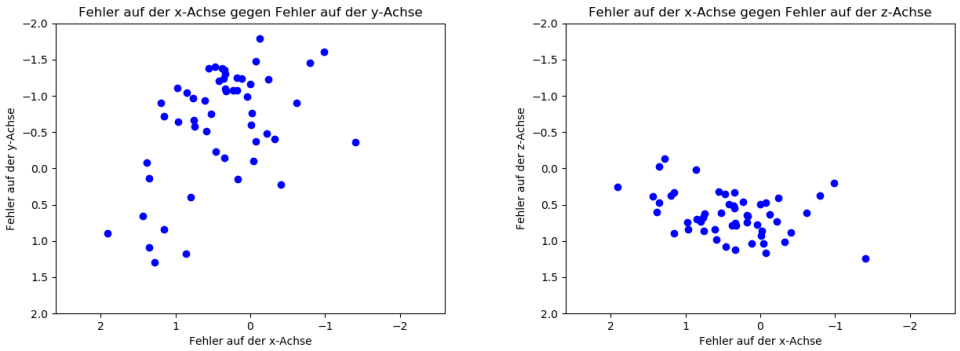

Abbildung 6.13: Taststab-Fehler auf der x-, y- und z-Achse
Distanzen auf der x-, y- und z-Achse, gegeneinander aufgetragen.

Auf der x-Achse liegen die Abweichungen zwischen 2 und -1.5 Millimetern, während auf der y-Achse Fehler von 1.5 bis -2 Millimetern zu beobachten sind. Die Abweichungen auf der z-Achse liegen zwischen 1.5 und -0.5 Millimetern. Auch für dieses Ergebnis wird der RMS berechnet, indem die euklidischen Distanzen aller aufgenommenen Punkte zum

Referenzpunkt berechnet werden. Für diese Evaluation ergibt sich dadurch ein RMS von 1.43 Millimetern. Damit erreicht diese Punkterkennung insgesamt ein leicht genaueres Ergebnis als das Marker3D-Verfahren. Es ist somit auch für die Aufnahme von Punkten geeignet.

6.5.4 MarkerRef

Die Positionen, die das Tracking System für die reflektierenden Marker berechnet, werden in dieser Arbeit nicht evaluiert. Während der Hersteller über die Genauigkeit des Systems keine Angaben macht, würde eine Evaluation dieses Systems nicht in den Rahmen dieser Arbeit passen. Die Punkte werden daher als korrekt angenommen.

7 Bestimmung der Objektposen

Nachdem im letzten Kapitel beschrieben wurde, wie an jedem Objekt drei Punkte aufgenommen werden können, wird es in diesem Kapitel darum gehen, aus diesen Punkten eine vollständige Objektpose zu berechnen. Drei der vier Verfahren zum Erkennen der Objektpunkte haben als Resultat die 3D-Punkte im Weltkoordinatensystem berechnet. Diese Verfahren waren Marker3D, Taststab und MarkerRef. Da im Folgenden nur auf diesen Punkten gearbeitet wird und auf keinerlei Zwischenergebnissen, ist die Berechnung der Objektpose unabhängig davon, wie die Punkte aufgenommen wurden. Es kann daher eine Methode für alle drei Verfahren genutzt werden. Das Verfahren Marker2D, welches Ceres zur Berechnung der Objektpose nutzt, berechnet lediglich die Bildkoordinaten und die Pose des Smartphones in der Welt. Die Bestimmung der Objektpose mit Ceres wird sich daher stark von der anderen Methode unterscheiden. Im Folgenden werden die zwei Methoden zur Berechnung der Objektposen vorgestellt. Anschließend werden die Verfahren, Objektposen zu berechnen, auf ihre Genauigkeit geprüft. Da die Genauigkeit der Objektpose unter anderem davon abhängt, wie genau die Punkte auf dem Objekt bestimmt wurden, wird die Evaluation für alle vier Verfahren durchgeführt.

7.1 3D-Punkte zu Objekt

Wie in Kapitel 3 beschrieben, bestimmen drei Punkte im dreidimensionalen Raum alle sechs Freiheitsgrade der Objektpose. Damit sollte es also möglich sein, eine Objektpose eindeutig zu bestimmen. Was allerdings bislang noch nicht berücksichtigt wurde, ist, dass die berechneten Objektpunkte zu einem gewissen Grad ungenau sind. Die ausgemessenen und die berechneten Punkte werden demnach nicht perfekt zusammenpassen. Keine Objektpose wird das Objekt auf eine Art in der Welt platzieren können, auf welche die berechneten Punkte exakt die korrekte Position auf dem Objekt haben.

Die Objektpose soll trotz dieser Problematik möglichst genau platziert werden. Hierzu wird ein Least-Square Ansatz von Arun et al. [5] angewendet, welcher basierend auf der Singulärwertzerlegung die Rotation und Translation zwischen zwei Mengen an Punkten bestimmt. Die Implementierung dieses Verfahrens ist dabei an die in der Arbeit von Nghia Ho [14] vorgeschlagene Implementierung angelehnt. Die erste der beiden Mengen wird die mit den verschiedenen Verfahren berechneten Punkte auf dem Objekt beinhalten. Die zweite Menge enthält die zuvor ausgemessenen Punkte auf dem Objekt. Die Zuordnung zwischen den Elementen der Mengen ist durch das jeweilige Aufnahmeverfahren gegeben. Ein Blick in die Koordinatensysteme, in welchen diese Punkte aufgenommen wurden,

macht deutlich, warum die Transformation zwischen diesen beiden Mengen an Punkten die gesuchte Transformation $T_{W\leftarrow O}$ ist. In der ersten Menge sind Punkte enthalten, die im Weltkoordinatensystem aufgenommen wurden. Angenommen es wurden N Punkte aufgenommen, so wird der i-te Punkt durch $p_i^{(W)}$ dargestellt. In der zweiten Menge sind Punkte enthalten, die in Objektkoordinaten ausgemessen wurden. Der korrespondierende i-te Punkt wird entsprechend durch $p_i^{(O)}$ dargestellt. Würden die Sensoren die Punkte exakt bestimmen, so würde nun für die gesuchte Transformation $T_{W\leftarrow O}$ die folgende Formel gelten:

$$p_i^{(W)} = T_{W\leftarrow O} \cdot p_i^{(O)}$$

Da die Punkte aber nicht exakt bestimmt werden können, muss eine Transformation gefunden werden, welche die Objektpunkte möglichst genau auf die Punkte in der Welt abbildet. Hierzu wird in der Arbeit von Arun et al. [5] eine Lösung des folgenden Least-Square Problems gesucht:

$$\Sigma^2 = \sum_{i=1}^{N} ||p_A^i - (Rp_B^i + T)||^2$$

Dabei sind R und T die gesuchte Rotation und Translation zwischen den beiden Mengen an Punkten. p_A^i ist der i-te Punkt in einer Punktmenge A, auf die abgebildet werden soll, während p_B^i der korrespondierende Punkte aus einer Menge B ist. Das Problem hat die allgemeine Form eines Non-linear Least-Square Problems, könnte also mit Ceres gelöst werden. Die hier verwendete Lösung nach Arun et al. benötigt aber keine initiale Schätzung und wird daher bevorzugt.

Um dieses Problem zu lösen, wird ein zweistufiger Prozess vorgeschlagen. Zunächst wird die Rotation R mithilfe der Singulärwertzerlegung bestimmt. Anschließend wird anhand dieser Rotation die Translation T bestimmt. Zunächst wird die Rotation bestimmt. Hierzu zerlegt die Singulärwertzerlegung eine Matrix M in drei Matrizen, sodass die folgende Formel gilt:

$$\begin{aligned}(U, S, V) &= \mathrm{SVD}(M)\\ M &= USV^T\end{aligned}$$

Die Methode SVD führt in dieser Formel die Singulärwertzerlegung durch. Die resultierenden Matrizen können, wenn M korrekt gewählt ist, die Rotation zwischen den beiden Punktmengen beschreiben. Die Matrix M wird hierzu wie folgt aufgebaut:

$$M = \sum_{i=1}^{N} (p_A^i - \mathrm{centroid}_A)(p_B^i - \mathrm{centroid}_B)^T$$

Dabei sind die Punkte centroid_A, centroid_B die Schwerpunkte der jeweiligen Menge. Es werden also beide Punktmengen verschoben, sodass ihre Schwerpunkte im Ursprung liegen. Diese Punkte werden entsprechend genutzt, um in M eine 3×3 Matrix zu akkumulieren. Nun lässt sich aus dieser Matrix wie folgt die Rotation R berechnen:

$$[U, S, V] = \text{SVD}(M)$$
$$R = VU^T$$

Diese Rotation ist in den meisten Fällen korrekt, es gibt allerdings einen Sonderfall, in welchem die Singulärwertzerlegung eine falsche Rotationsmatrix liefert. Um diesen Fall abzufangen wird die Determinante der Matrix R berechnet. Ist sie kleiner als 0, so wird die dritte Spalte der Matrix R mit -1 multipliziert. Die resultierende Matrix ist die gesuchte Rotationsmatrix.

Nun, da die Rotation bestimmt wurde, kann mit ihrer Hilfe die Translation wie folgt berechnet werden:

$$T = -R \times \text{centroid}_A + \text{centroid}_B$$

Damit sind sowohl die Rotation als auch die Translation bestimmt. Zusammen geben sie eine Transformation von der Menge A in die Menge B an. Im Anwendungsfall der Punkte auf Objekten beschreibt die Menge A die Punkte im Koordinatensystem des Objekts, während die Menge B die Punkte in Weltkoordinaten sind. So ergibt sich die gesuchte Transformation $T_{W \leftarrow O}$ von Objektkoordinaten nach Weltkoordinaten. Damit ist die Berechnung der Objektpose mithilfe der Verfahren Marker3D, MarkerRef und Taststab abgeschlossen.

7.2 2D-Punkte zu Objekt

Offen bleibt damit nur noch die Berechnung der Objektpose mithilfe der Bibliothek Ceres für das Marker2D-Verfahren. In diesem Ansatz wird die Objektpose nicht anhand von drei 3D-Punkten berechnet, sondern anhand der Pixelkoordinaten der Marker im Bild des Smartphones. Kennt man den Öffnungswinkel einer Kamera, so können Pixelkoordinaten auch als Winkelmaß betrachtet werden. Jeder Pixel beschreibt eine Gerade, die von dem Kamerasensor aus durch die Linse verläuft, bis sie den Punkt in der Welt erreicht, welcher von der Kamera an diesem Pixel aufgenommen wurde. Diese Gerade hat auf der x- und y-Achse einen gewissen Winkel, welcher sich bei bekanntem Öffnungswinkel aus den Pixelkoordinaten ablesen lässt. Ist zusätzlich zu diesen Winkeln noch die Pose der Kamera in der Welt bekannt, so ergibt sich eine Gerade in der Welt.

Nach diesem Prinzip können für die drei erkannten Kreismarker drei Geraden in der Welt berechnet werden, welche an einem unbekannten Punkt auf diesen Geraden auf die Kreismarker treffen. Dieses Prinzip ist in Abbildung 7.1 visualisiert.

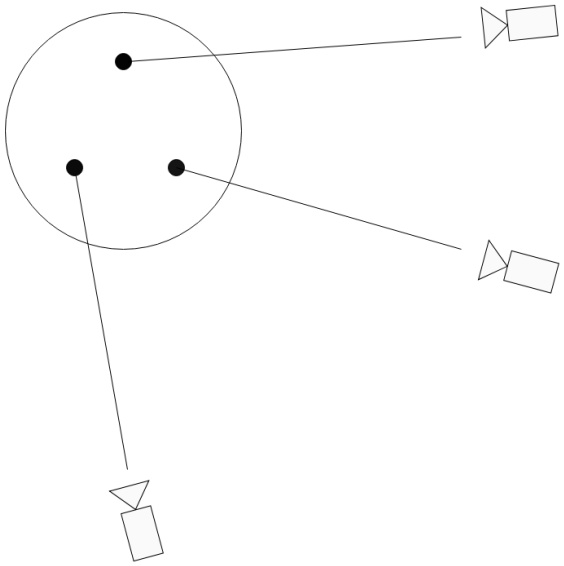

Abbildung 7.1: Von dem Smartphone ausgehende Geraden durch die Mittelpunkte der Kreismarker auf einem Objekt, aus drei unterschiedlichen Perspektiven

Ceres soll nun anhand dieser drei Geraden schätzen, wo sich das Objekt befunden haben muss, damit die drei Geraden durch die Mittelpunkte der zugehörigen Kreismarker verlaufen. Dazu muss wie zuvor angegeben werden, welche Parameter Ceres zu kalibrieren hat. Ähnlich wie in der Kalibrierung zwischen den Sensoren und ihren Targets wird in dieser Kalibrierung eine Transformation gesucht. Dementsprechend werden auch hier sieben Parameter kalibriert: die vier Werte q_w, q_x, q_y, q_z des Quaternions, welches

die Rotation des Objekts beschreibt und die drei Werte x_t, y_t, z_t für die Translation des Objekts. Kombiniert beschreiben diese Parameter die Transformation $T_{W \leftarrow O}$ von Objektkoordinaten in das Weltkoordinatensystem. Dementsprechend ist das Modell x, welches Ceres optimieren soll, wie folgt gegeben:

$$x = (T_{W \leftarrow O})$$

Weiterhin benötigt Ceres die Information, wie die Residuen des Non-Linear Least-Square Problems zu berechnen sind. Auch in der Berechnung der Residuen ähnelt dieses Vorgehen stark dem Verfahren der Kalibrierung der Target-Transformationen. Ceres wird die Pose des Objekts in der Welt schätzen. Anhand dieser Pose lassen sich auch die Positionen der Marker in der Welt berechnen, da die Positionen der Marker auf dem Objekt bekannt sind. Nun können diese Positionen der Marker wie zuvor in das Bild des Smartphones projiziert werden und dort mit dem Resultat der Kreismarkererkennung verglichen werden. Die Distanz dieser Pixelpositionen wird als Residuum dienen. Zur Berechnung der Residuen wird auf die in Kapitel 3 vorgestellte Funktion f zurückgegriffen, welche gegebene Punkte in das Kamerabild projiziert. So lassen sich die Residuen wie folgt berechnen:

$$r_i(x) = \begin{pmatrix} u_i(x) \\ v_i(x) \end{pmatrix} - \begin{pmatrix} u_i' \\ v_i' \end{pmatrix}, \text{wobei}$$

$$\begin{pmatrix} u_i(x) \\ v_i(x) \end{pmatrix} = f(p_i^{(O)}, T_{S \leftarrow O}(x), \Theta)$$

Dabei sind Θ die intrinsischen Parameter der Kamera, $p_i^{(O)}$ ist die auf dem Objekt ausgemessene Position des Markers. Die Werte u_i', v_i' sind auch in dieser Kalibrierung das Resultat der Kreismarkererkennung. Die Transformation $T_{S \leftarrow O}$ zwischen Objektkoordinatensystem O und Kamerakoordinatensystem S setzt sich dabei wie folgt zusammen:

$$T_{S \leftarrow O}(x) = T_{S \leftarrow ST} \cdot T_{ST \leftarrow W} \cdot T_{W \leftarrow O}$$

Da die Differenz in Bildkoordinaten betrachtet wird, werden insgesamt pro Kreismarker zwei Residuen berechnet. Demnach ergeben sich für die drei Kreismarker auf einem Objekt insgesamt sechs Residuen. Anhand dieser Informationen könnte Ceres bereits die Pose des Objekts schätzen. Das auf diese Weise modellierte Problem hat allerdings nicht in jedem Fall genau eine Lösung. So ist es beispielsweise möglich, dass alle Geraden exakt parallel zueinander verlaufen und die Position des Objekts in der Tiefe nicht eingeschränkt wird. Entlang der Geraden könnte sich das Objekt in jeder Distanz befinden, während immer noch alle Geraden durch die Mittelpunkte der Kreismarker verlaufen. Um falsche Lösungen zu vermeiden, muss auch in diesem Ansatz die Entfernung der Marker zum

Smartphone mit einbezogen werden. In der Berechnung der Residuen wird sie allerdings entsprechend ihrer Genauigkeit niedriger gewichtet.

Um die Distanz als Residuum einfließen zu lassen, wird berechnet, welche Größe der Punkt auf der von der Kreismarkererkennung bestimmten Distanz im Bild haben sollte. Hierzu wird wie zuvor in der Berechnung der Distanz von Kreismarkern das Modell für dünnen Linsen genutzt, um folgendermaßen die Größe o des Kreismarkers im Bild zu berechnen:

$$o = i \cdot \frac{o_w}{d}$$

Dabei ist o_w die Größe des Markers in der Welt, i ist die kalibrierte Bildweite und d ist die Distanz des Markers zur Kamera. Das Resultat o, also die Größe der Ellipse im Bild, kann nun mit dem Resultat der Kreismarkererkennung o' verglichen werden. Daraus lässt sich ein neues Residuum berechnen. Um die Residuen allerdings entsprechend der Genauigkeiten der Parameter zu gewichten, wird die Berechnung der Residuen wie folgt angepasst, wobei sich die Werte u_i, v_i durch die Anwendung der Funktion f wie oben beschrieben ergeben:

$$r_i(x) = \begin{pmatrix} \sqrt{\frac{1}{4}} & 0 & 0 \\ 0 & \sqrt{\frac{1}{4}} & 0 \\ 0 & 0 & \sqrt{1} \end{pmatrix} \left(\begin{pmatrix} u_i(x) \\ v_i(x) \\ o_i(x) \end{pmatrix} - \begin{pmatrix} u'_i \\ v'_i \\ o'_i \end{pmatrix} \right)$$

Dabei werden die Residuen anhand ihrer erwarteten Abweichungen gewichtet. Die Abweichungen in den beiden Bildkoordinaten u, v sind auf $\frac{1}{4}$ Pixel geschätzt. Der Durchmesser o, welcher aus der etwas ungenaueren Distanz des Punktes berechnet wurde, wird mit einer erwarteten Abweichung von einem Pixel gewichtet. So fließen die Residuen für die Werte u, v nun mit einem höheren Gewicht in die Berechnung der Objektpose ein als das Residuum für o.

Als letzten Schritt benötigt auch dieses Verfahren eine initiale Schätzung der Pose, da andernfalls ein lokales Minimum als Ergebnis gefunden werden könnte. Da für dieses Verfahren die Kreismarker aufgenommen wurden, sind hier auch die 3D-Positionen der Marker bekannt. Daher kann als initiale Schätzung die Pose genutzt werden, die mit dem Verfahren Marker3D berechnet wird.

Damit sind für Ceres alle Informationen gegeben, um eine Objektpose zu berechnen. Insgesamt werden drei Punkte auf den Objekten aufgenommen, wobei jeder dieser Punkte drei Residuen liefert. Anhand dieser Residuen wird die Transformation $T_{W \leftarrow O}$ geschätzt.

7.3 Evaluation

Nachdem im letzten Kapitel evaluiert wurde, mit welcher Genauigkeit in den einzelnen Verfahren die Punkte aufgenommen werden, wird sich diese Evaluation damit beschäftigen,

wie präzise die daraus berechneten Posen bestimmt werden. Hierzu wird entsprechend der in diesem Kapitel beschriebenen Methoden mehrfach die Pose des selben Objekts bestimmt. Als Objekt wird dabei eine Pappschablone verwendet, auf welcher sowohl reflektierende Marker als auch Kreismarker angebracht werden. Diese Schablone wird für die Durchführung des Versuchs an einer Holzplatte festgeklebt, um ein Verrutschen des Objekts zu vermeiden. Insgesamt werden für jede der vier Methoden 50 Posen aufgenommen. Der erste Versuchsdurchlauf wird mit den reflektierenden Markern durchgeführt, sodass die von dem System bestimmte Pose im Verlauf der Evaluation als Vergleichswert für die anderen Verfahren genutzt werden kann.

In der Abbildung 7.2 sind die Verteilungen der einzelnen Parameter der Objektposen abgebildet, welche mithilfe der reflektierenden Marker aufgenommen wurden. Die Pose ist anhand der Translation auf der x-, y- und z-Achse sowie der Rotation um diese Achsen in Form von Eulerwinkeln angegeben. Da für diese Werte keine Vergleichspose existiert, wird stattdessen lediglich das Rauschen der Objektpose betrachtet, also wie stark sich zwei Messungen des gleichen Objekts unterscheiden können.

Wie die Grafiken zeigen, ist nur leichtes Rauschen in der Bestimmung der Objektposen zu beobachten. In der Translation sind auf allen Achsen zwischen zwei Messungen maximal 0.15 Millimetern Abweichungen zu beobachten. Auch in der Rotation wird die Pose sehr stabil bestimmt. Die Abweichungen in den Rotationen um die einzelnen Achsen liegen bei maximal 0.1 Grad. Die Posen die mit diesem Verfahren aufgenommen werden, dienen im Folgenden als Vergleichswerte für die anderen Verfahren. Da immer dasselbe Objekt in der selben Pose aufgenommen wurde, lässt sich die Genauigkeit der anderen Verfahren anhand der Distanz zu einer mit reflektierenden Markern aufgenommenen Pose evaluieren. Hierzu wird die mittlere Pose aller MarkerRef-Posen verwendet.

Als zweites wird das Marker3D-Verfahren evaluiert. Hierzu wird dieselbe Pose wie zuvor 50 Mal mit dem Marker3D-Verfahren bestimmt. In Abbildung 7.3 sind die gemessenen Posen anhand ihrer Translation und Rotation visualisiert und gegen die Referenzpose aufgetragen.

Wie hier zu sehen ist, kommt es wie erwartet bei der Kreiserkennung zu einer größeren Streuung der erkannten Objektposen. In der Translation ist auf der x-Achse eine Streuung der Position zwischen 228.6 und 230 zu beobachten. Auf der y-Achse liegen die Werte zwischen -26.6 und -28.1. Auf der z-Achse sind Werte zwischen -74.4 und -77 zu beobachten. Auch in der Rotation sind die Abweichungen etwas größer. Die Rotation um die x-Achse liegt zwischen -1.5 und 1, die Rotation um die y-Achse hingegen zwischen -2 und 1.5. Die Rotation um die z-Achse streut zwischen 62.9 und 63.6. Verglichen mit der Pose, die anhand der reflektierenden Marker aufgenommen wurde, zeigt sich hier ein leichter konstanter Fehler. So liegen die Werte auf der x-Achse mit diesem Verfahren im Durchschnitt etwa 0.5 Millimeter höher als der Referenzwert. Der Grund hierfür ist, dass die beiden Verfahren unterschiedliche Marker nutzen. Wird beim Ausmessen der Marker auf einem Objekt ein kleiner Fehler gemacht, so verschiebt sich dadurch der Ursprung dieses Objekts leicht und die Pose wird leicht rotiert. Wird dieser Fehler bei anderen

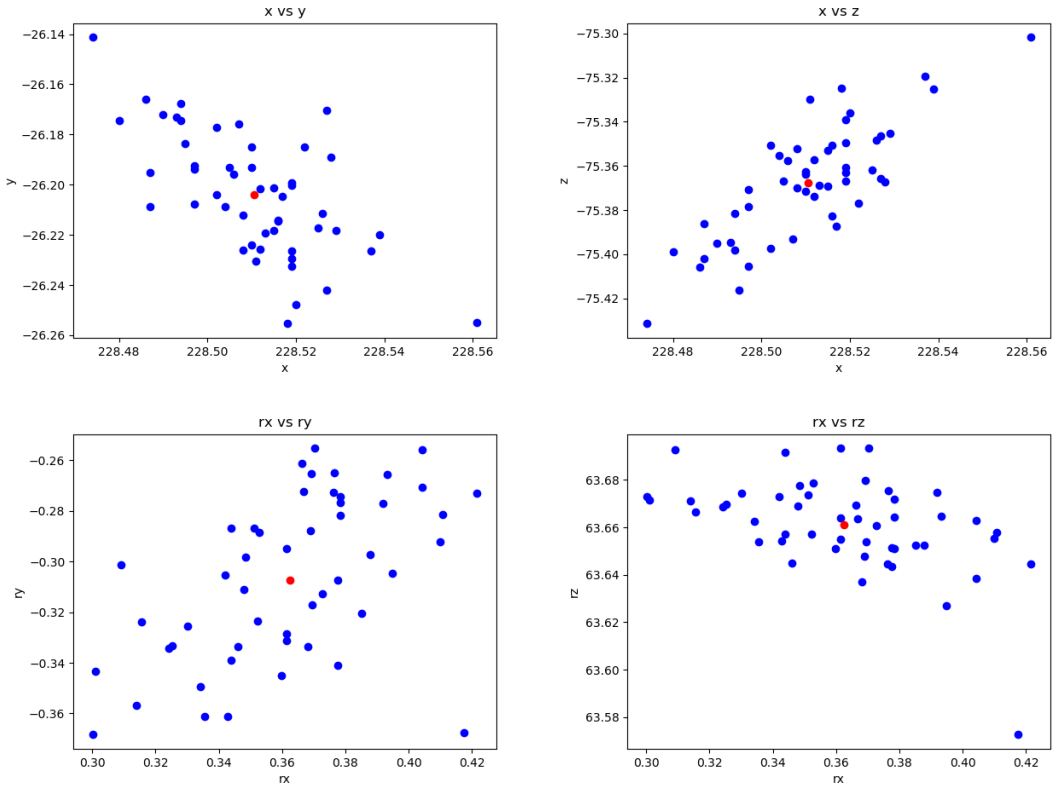

Abbildung 7.2: Werte für die Rotation in Grad und Translation in Millimetern der mit dem MarkerRef-Verfahren aufgenommenen Objekte
Gemessene Punkte sind blau markiert, der Mittelwert und spätere Referenzpunkt rot.

Markern nicht gemacht, so kann es passieren, dass es zwischen zwei Messverfahren einen konstanten Fehler gibt. Da das Ausmessen der Marker per Hand durchgeführt werden muss, wird es zu einem gewissen Grad bei jeder Messung zu solchen Fehlern kommen.

Um nun die mit der Kreismarkererkennung berechnete Pose mit der tatsächlichen Pose des Objekts zu vergleichen, wird in Abbildung 7.4 die Differenz zur Referenzpose anhand des Fehlers in der Translation sowie in der Rotation visualisiert. Die Fehler in der Translation werden dabei anhand der euklidischen Distanz der Objektursprünge angegeben, während die Fehler in der Rotation berechnet werden, indem die kürzeste Rotation zwischen den beiden Posen bestimmt wird. Hierzu wird Eulers Rotationstheorem ausgenutzt, nach welchem jede Rotation durch eine Achse und einen Rotationswinkel um diese Achse ausgedrückt werden kann. Betrachtet man die Rotation zwischen der Referenzpose und der aktuellen Objektpose, so ist dieser Rotationswinkel die gesuchte Abweichung in der Rotation.

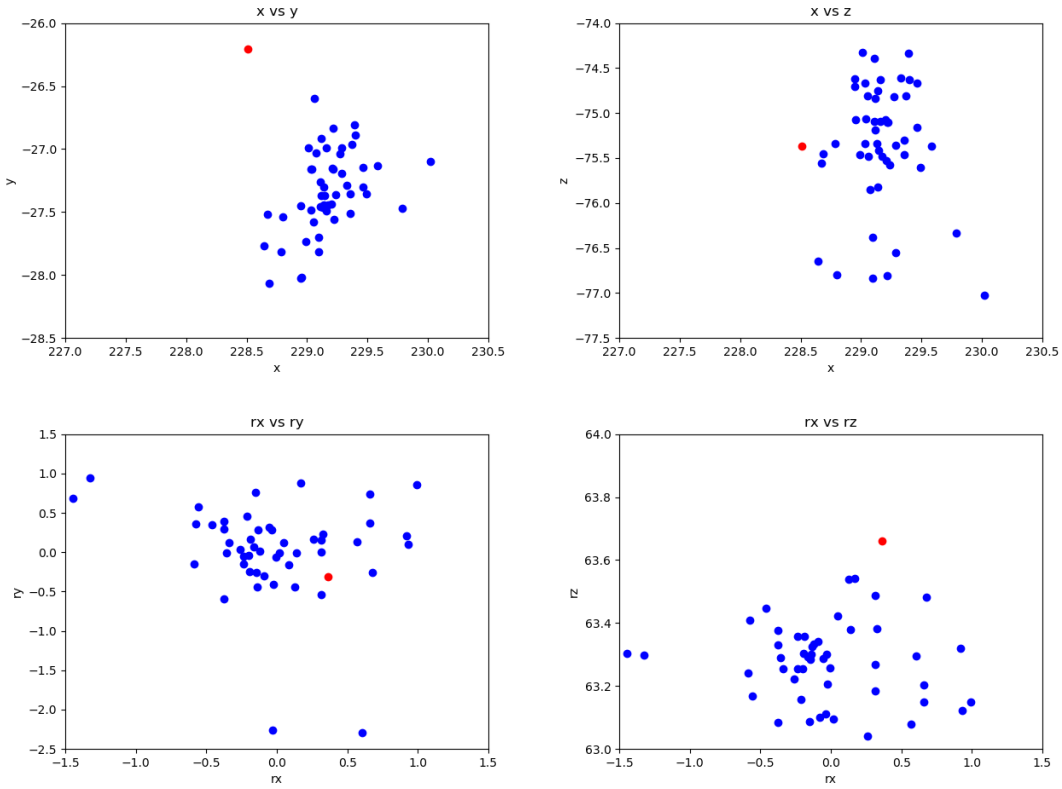

Abbildung 7.3: Werte für die Rotation in Grad und Translation in Millimetern der mit dem Marker3D-Verfahren aufgenommenen Objekte
Die Referenzpose ist in Rot dargestellt. Aufgrund des deutlich größeren Wertebereichs unterscheidet sich die Skalierung der Achsen im Vergleich zur Auswertung der MarkerRef-Posen.

In dieser Abbildung ist zu sehen, dass die Abweichungen in der Translation zwischen 0.8 und 2 Millimetern liegen, mit einigen Ausreißern bei bis zu 2.5 Millimetern Fehler. In der Rotation liegen die meisten Fehler im Bereich zwischen 0.4 und 1.5 Grad. Auch hier sind einige Ausreißer zu beobachten, welche einen Fehler von bis zu 2.1 Grad erreichen. Auch zu diesem Ergebnis wird der RMS-Wert berechnet. Für die Rotation ergibt sich ein RMS von 1.04 Grad, während in der Translation der RMS 1.57 Millimeter beträgt.

Die dritte Evaluation betrachtet die vom Pointer aufgenommenen Objektposen. Auch hier wurde auf dieselbe Art verfahren wir zuvor. Es wurde dieselbe Pose 50 Mal aufgenommen und mit der Pose verglichen, die aus den reflektierenden Markern berechnet wurde. Hierzu sind in Abbildung 7.5 die aufgenommenen Posen visualisiert.

Wie in der Abbildung zu sehen ist, fällt die Streuung insgesamt etwas niedriger aus.

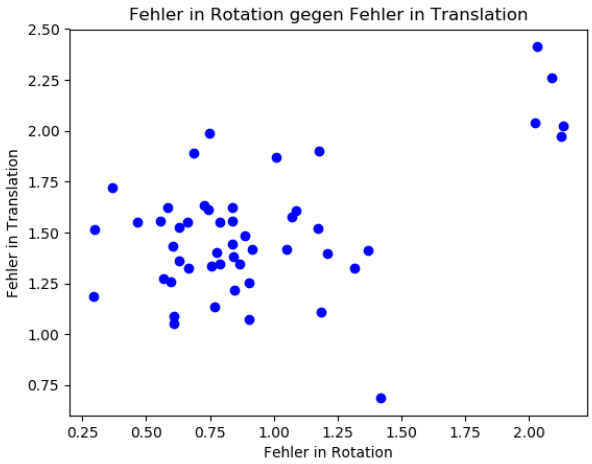

Abbildung 7.4: Fehler der berechneten Posen in der Translation in Millimetern und in der Rotation in Grad

Auf der x-Achse liegen die Werte zwischen 229 und 230.2, auf der y-Achse zwischen -27.4 und -28.4, während die Werte auf der z-Achse im Bereich zwischen -75.3 und -75.7 liegen. Auch in der Rotation ist die Streuung geringer. Die Rotation um die x-Achse liegt zwischen 0.38 und 0.55. Auf der y-Achse liegen die Werte zwischen -0.025 und -0.175. Die Rotation um die z-Achse liegt im Bereich zwischen 63.1 und 63.6. Während die Streuung deutlich geringer ausfällt, ist auch in diesem Ergebnis ein konstanter Fehler zu beobachten. In Abbildung 7.6 ist die Distanz zwischen den gemessenen Posen und der Referenzpose aufgetragen.

Wie man in dieser Abbildung sieht, liegen die Fehler in der Translation zwischen 1.4 und 2.6 Millimetern. Der Grund hierfür ist der konstante Fehler der Translationen, welcher dafür sorgt, dass die Fehlerwerte nicht um 0 herum streuen. In der Rotation wirkt sich dies deutlich geringer aus, dort liegt der Fehler zwischen 0.2 und 0.6 Grad. Insgesamt ereben sich RMS-Werte von 0.38 Grad in der Rotation und 2.05 Millimeter in der Translation.

Die letzte Evaluation betrachtet die mithilfe von Ceres berechneten Objektposen und ihre Genauigkeit. Beim Aufnehmen von Objektposen mit Ceres hat sich schnell herausgestellt, dass dieser Ansatz keine ausreichende Genauigkeit erzielt. In Abbildung 7.7 sind die Fehler in der Rotation und Translation jeweils gegen den Reprojektionsfehler von Ceres aufgeführt.

Wie hier zu sehen ist, erreichen die Fehler in der Rotation Werte von bis zu 14 Grad, in der Translation sind Fehler von bis zu 9 Millimeter zu beobachten. Dabei fällt auf, dass die Fehler in der Rotation und in der Translation nicht von dem Reprojektionsfehler

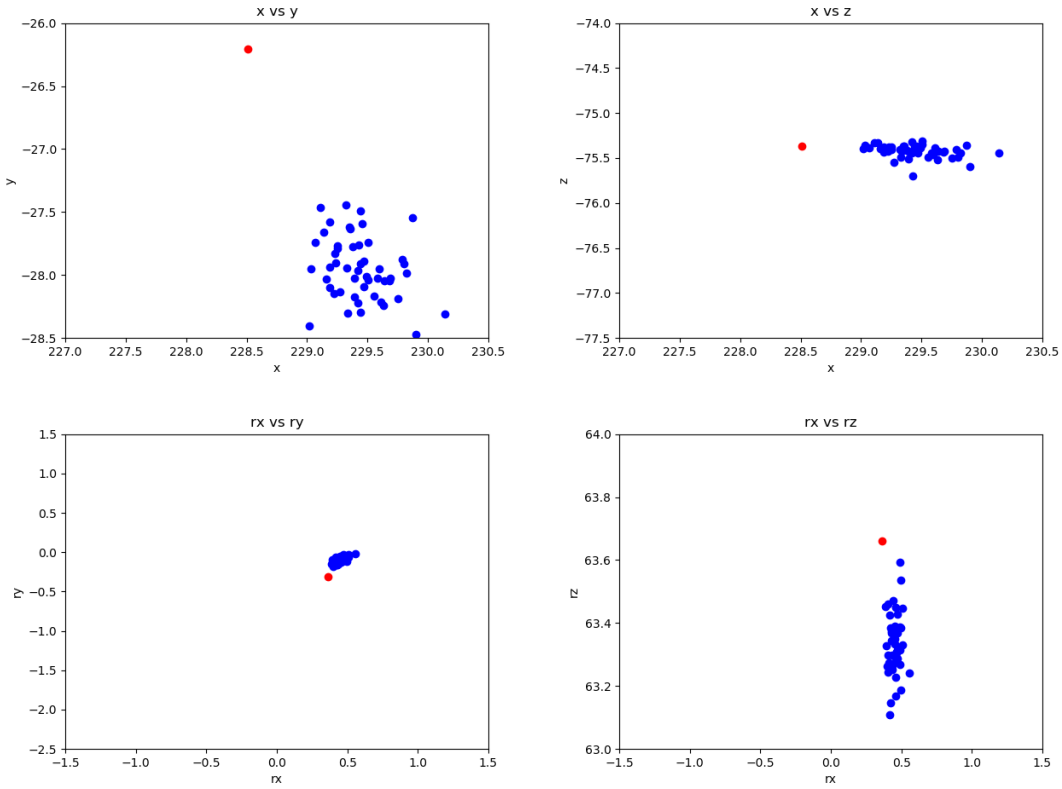

Abbildung 7.5: Werte für die Rotation in Grad und Translation in Millimetern der mit dem Taststab-Verfahren aufgenommenen Objekte
Die Referenzpose ist in Rot dargestellt. Die Skalierung der Achsen ist zu Vergleichszwecken dieselbe wie in der Evaluation des Marker3D-Verfahrens.

abhängen. Für hohe Reprojektionsfehler kann das Verfahren trotzdem gute Ergebnisse erzielen und umgekehrt. Posen mit solchen Genauigkeiten eignen sich nicht für Ground-Truth Datensätze. Dieses Ergebnis ist zunächst überraschend, da Ceres eingesetzt wurde, um auf den genauer bestimmten Pixelpositionen zu arbeiten und so auf die ungenaueren Punkte in der Welt zu verzichten. Während es im Rahmen dieser Arbeit nicht möglich war, den Grund für diese ungenaueren Ergebnisse zu bestimmen, wurde ein Test durchgeführt, um zumindest Rückschlüsse auf einen möglichen Grund für diese Ergebnisse treffen zu können.

In der Benutzung des Infrarot-Tracking Systems fällt auf, dass die Pose von den Tracking-Targets leicht sensibel gegenüber Verdeckungen einzelner Kugelmarker ist. Wird eine Kugel verdeckt, so sind Abweichungen in der Pose des Targets zu beobachten. Solche Verdeckungen kommen in der Benutzung des Systems regelmäßig vor, insbesondere

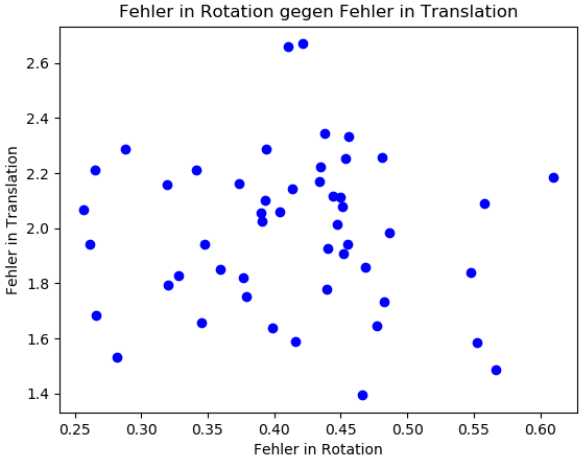

Abbildung 7.6: Fehler der berechneten Posen in der Translation in Millimetern und in der Rotation in Grad

können sich Kugelmarker auch gegenseitig verdecken. Solche Fehler in der Pose des Targets haben auf die erkannten Kreismarkerpositionen nur eine minimale Auswirkung. Da die Kreismarker aus einer Distanz von circa 100 Millimetern aufgenommen werden, würde selbst ein Fehler von einem Grad nur zu einer Abweichung von etwa 1.7 Millimetern führen. Im Ansatz von Ceres ist eine solche Abschätzung nicht möglich. Das Problem wurde so modelliert, dass die Strahlen von der Kamera aus durch die Mittelpunkte der Kreismarker verlaufen sollen. Verläuft nun einer dieser Strahlen nicht exakt durch den Mittelpunkt des Markers, so ist nicht leicht absehbar, wie das Objekt in der Optimierung von Ceres in der Welt platziert wird. Es ist möglich, dass dieses Verfahren deutlich stärker auf leichte Abweichungen in den Messwerten reagiert als die Berechnung anhand von 3D-Positionen. Insbesondere würde dies erklären, warum es keinen Zusammenhang zwischen dem Reprojektionsfehler und dem Fehler der Pose gibt. Sind bereits die Messungen inkorrekt, so kann es sein, dass ein gutes Modell für die Messungen bestimmt wird, welches aber nicht die korrekte Pose des Objekts beschreibt. Um die Sensibilität gegenüber Messfehlern zu testen, wurde ein Versuch durchgeführt, in welchem künstlich solche Fehler produziert wurden. Der Versuch ist im Folgenden beschrieben ist.

Zur Durchführung des Versuchs wurden die drei Kreismarker an einem Objekt mit dem Smartphone aufgenommen. Um nun den Effekt von inkorrekten Target-Posen zu simulieren, wurden zu den drei Aufnahmen der Marker jeweils 100 weitere Markerdaten berechnet. Diese Daten wurden berechnet, indem auf die Pose des Targets, die von dem Tracking-System geliefert wird, eine zufällige Transformation angewendet wird, welche die Pose um jede Achse zufällig um bis zu einen Grad rotiert. Dies wirkt sich

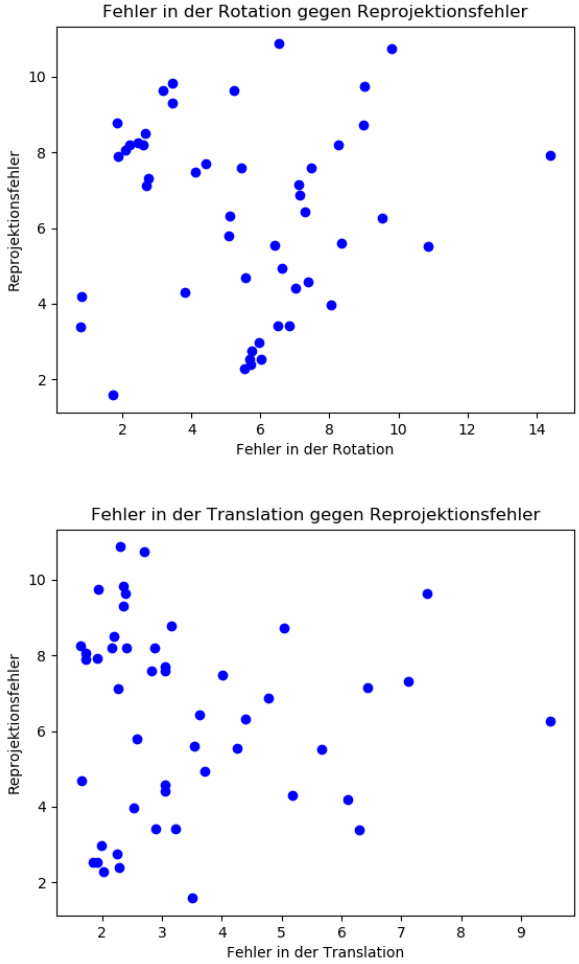

Abbildung 7.7: Fehler in der Rotation in Grad sowie in der Translation in Millimetern gegen den berechneten Reprojektionsfehler

sowohl auf die 3D-Positionen der Marker in der Welt als auch auf die Geraden zwischen Smartphone und Marker aus. Beide treffen nicht exakt die korrekte Position. Interessant ist nun, wie stark die berechnete Pose des Objekts von diesen Fehlern abhängt. Hierzu wird wieder der Fehler in der Rotation sowie in der Translation betrachtet zwischen der zuerst aufgenommenen Pose und den Posen, die mit den verrauschten Daten berechnet wurden. Dabei zeigt sich ein deutlicher Unterschied zwischen den beiden Verfahren. Während in der Berechnung anhand der 3D-Punkte ein durchschnittlicher Fehler von 1.63 Grad in der Rotation und 3.6 Millimetern in der Translation zu beobachten ist, liegen die Abweichungen der von Ceres berechneten Posen deutlich höher. Dort sind

durchschnittliche Abweichungen von 6.54 Grad in der Rotation und 6 Millimetern in der Translation zu sehen. Damit wird die Vermutung bestätigt, dass die Berechnung mit Ceres sensibler auf Rauschen in den Sensordaten reagiert. Während diese Evaluation nicht ausschließen kann, dass auch andere Faktoren eine Rolle in den schlechteren Ergebnissen spielen, zeigt sie doch, dass dieser Faktor bereits einen großen Einfluss hat.

Zusammenfassend haben drei der vier entwickelten Verfahren zum Aufnehmen von Objektposen eine ausreichende Genauigkeit erzielt, um sie für die Aufnahme von Ground-Truth Daten zu berücksichtigen. Da für die Posen, die mit dem MarkerRef Verfahren berechnet wurden, keine Vergleichswerte vorhanden sind, wurden diese Anhand ihrer Streuung evaluiert. Die mit diesem Verfahren aufgenommenen Posen weisen nur minimale Abweichungen auf. Alle Translationen liegen im Bereich von 0.15 Millimetern zueinander, die Rotationen unterscheiden sich lediglich um etwa 0.1 Grad zwischen zwei Messungen. Anhand der mit diesem Verfahren aufgenommenen Posen wurden die weiteren Verfahren getestet. Die Posen, welche mit dem Marker3D-Verfahren aufgenommen wurden, erreichten Abweichungen von durchschnittlich 1.04 Grad in der Rotation und 1.57 Millimetern in der Translation. Die Posen, die mit dem Taststab aufgenommen wurden, hatten durchschnittliche Abweichungen von 0.38 Grad in der Rotation und 2.05 Millimeter in der Translation. Das Verfahren Marker2D, welches Ceres zur Berechnung von Objektposen nutzt, erzielte keine ausreichend genauen Ergebnisse. Mit Abweichungen von bis zu 9 Millimetern in der Translation und bis zu 14 Grad in der Rotation ist es nicht für die Aufnahme von Ground-Truth Daten geeignet.

In der Evaluation der Target-Kamera Transformationen in Kapitel 5 wurden die Metriken 2 Grad 2 Zentimeter, 5 Grad 5 Zentimeter und 10 Grad 10 Zentimeter vorgestellt. Zu diesen Metriken wurde vorgeschlagen, dass die für sie genutzten Ground-Truth Daten höchstens einen Viertel dieser Fehler haben dürfen. Daraus ergeben sich die Werte 0.5 Grad und 0.5 Zentimeter, 1.25 Grad und 1.25 Zentimeter sowie 2.5 Grad und 2.5 Zentimeter zur Evaluation von Ground-Truth Daten. Wie sich in dieser Evaluation zeigt, erreicht das Verfahren MarkerRef Abweichungen von maximal 0.1 Grad und 0.15 Millimetern. Damit ist es für alle vorgestellten Metriken geeignet, Ground-Truth Daten aufzunehmen. Der Taststab erreicht Genauigkeiten von 0.38 Grad und 2.05 Millimeter. Auch diese Genauigkeiten erlauben es, Daten für alle drei Metriken aufzunehmen. Mit einer durchschnittlichen Abweichung von 1.04 Grad und 1.57 Millimetern ist das Verfahren Marker3D zu ungenau, um Ground-Truth Daten für die 2 Grad 2 Zentimeter Metrik zu liefern. Die Metriken 5 Grad 5 Zentimeter sowie 10 Grad 10 Zentimeter können aber anhand der mit dem Marker3D-Verfahren aufgenommenen Ground-Truth Daten evaluiert werden.

Diese Evaluation hat bisher lediglich betrachtet, ob die Posen im Weltkoordinatensystem mit eine ausreichenden Genauigkeit bestimmt werden. Um im Gesamtsystem bewerten zu können, ob die Posen ausreichende Genauigkeiten erreichen, muss die erwartete Abweichung mit einbezogen werden, welche von der Transformation in das Koordinatensystem der 3D-Kamera erzeugt wird. In der Evaluation der Target-Kamera

Genauigkeiten pro Verfahren	Translation	Rotation	Translation Gesamtsystem
MarkerRef	0.15	0.1	3.47
Marker3D	1.57	1.04	4.89
Taststab	2.05	0.38	5.37
Marker2D	9.0	14.0	12.32

Tabelle 7.1: Genauigkeiten der entwickelten Verfahren
Genauigkeiten der Verfahren in der Translation in Millimetern und in der Rotation in Grad. Zusätzlich ist die kombinierte Genauigkeit in der Translation angegeben, welche die Genauigkeiten der Target-Kamera Kalibrierung und der Posenberechnung vereint.

Kalibrierung wurde bestimmt, dass auf eine übliche Arbeitsdistanz von einem Meter etwa 3.32 Millimeter Abweichugen in der Position der Posen entstehen können. Diese Abweichung muss entsprechend auf die erwarteten Fehler jedes Verfahrens addiert werden. Für die Verfahren MarkerRef und Marker3D ändert sich durch diese zusätzliche Ungenauigkeit nichts. Beide Verfahren liegen in der Translation weiterhin unter der 0.5 Zentimeter Grenze. Das Verfahren Taststab liegt allerdings durch diese zusätzliche Ungenauigkeit mit einer Abweichung von 5.37 Millimetern minimal über der 0.5 Zentimeter Grenze. Eine solche Überschreitung kann noch als akzeptabel bewertet werden, hier muss der Nutzer des Datensatzes selbst entscheiden, ob die Genauigkeit zur Evaluation anhand der 2 Grad 2 Millimeter Metrik ausreicht. Alle drei Verfahren erreichen aber weiterhin Genauigkeiten, die die Evaluation anhand der Metriken 5 Grad 5 Millimeter und 10 Grad 10 Millimeter erlaubt. Die Ergebnisse aller Verfahren wurde ist in Tabelle 7.1 zusammengefasst. Die Genauigkeiten der Systeme MarkerRef und Marker2D sind dabei lediglich Annäherungen. Für das Verfahren MarkerRef wurde keine Evaluation anhand von Referenzposen durchgeführt, weshalb hier die maximale Streuung angegeben ist. Das Verfahren Marker2D wurde aufgrund der deutlich zu ungenauen Ergebnisse nicht detailliert evaluiert. Hier sind lediglich die maximalen Abweichungen angegeben.

8 Erstellung des Datensatzes

Neben der Entwicklung eines Systems zum Aufnehmen von Ground-Truth Daten soll in dieser Arbeit auch der erste Teil eines Datensatzes erstellt werden. Während bisher ausschließlich beschrieben wurde, wie das Aufnehmen eines solchen Datensatzes technisch funktionieren kann, wird dieses Kapitel behandeln, wie die Aufnahmen durchgeführt wurden. Hier wird darauf eingegangen, welche Szenen erstellt wurden, welche Methodiken dafür angewendet wurden und welche der entwickelten Verfahren sich in bestimmten Aufnahmesituationen gut eignen.

8.1 Übersicht über aufgenommene Szenen

Insgesamt wurden im Rahmen dieser Arbeit neun verschiedene Szenen aufgenommen. Unter den dafür genutzten Objekten befinden sich Küchenutensilien aus allen Bereichen der Küche von Besteck über Töpfe bis hin zu Gläsern, Tellern und Schüsseln. Die Szenen teilen sich in die einzelnen Schubladen und Fächer einer Küchenzeile auf. So enthält ein Datensatz jeweils alle Ground-Truth Daten zu den Objekten, die sich in der jeweiligen Schublade befinden. Abbildung 8.1 gibt einen Überblick über den aufgenommenen Datensatz. Jedes Bild zeigt dabei eine vollständige Szene, in welcher alle Objekte mit ihren Posen annotiert wurden.

Zu jeder dieser Szenen wurden im Durchschnitt 266 Bilder aufgenommen, was einen Datenumfang von etwa 2 400 annotierten Bildern ergibt. Dabei wurde die Hälfte dieser Bilder mit der Kinect aufgenommen, während die anderen Hälfte mit der rc_visard aufgenommen wurde. Diese Bilder wurden ungefähr in einem Zeitrahmen von 24 Stunden aufgenommen, was eine Bildrate von 100 annotierten Bildern pro Stunde ergibt. In den Bildern der Szenen sind dabei nach und nach mehr Objekte enthalten. So sind in den ersten $12 - 18$ Bildern in der Regel nur ein bis zwei Objekte enthalten. Die Szenen, die in Abbildung 8.1 gezeigt sind, entsprechen jeweils den letzten aufgenommenen Bildern. Dort sind alle Objekte enthalten, die in dem Datensatz aufgenommen wurden. Die Szenen werden also nach und nach aufgebaut. Diese Vorgehensweise beim Aufnehmen der Objekte ist im folgenden Kapitel beschrieben.

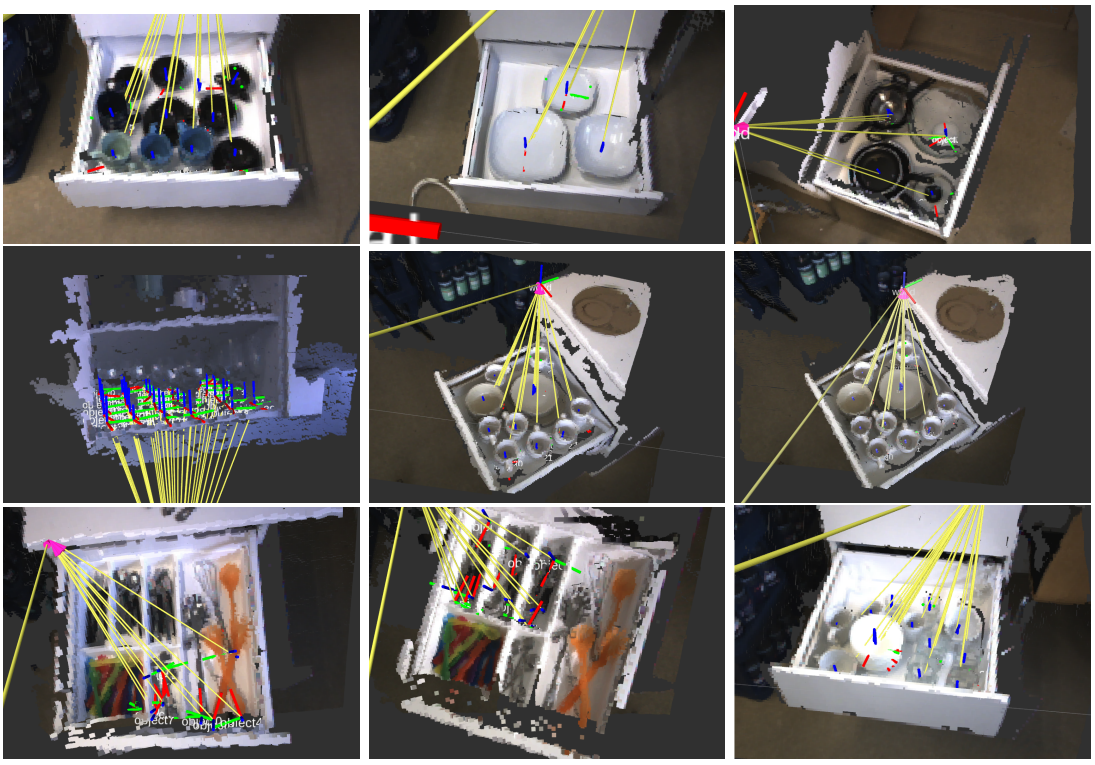

Abbildung 8.1: Übersicht über alle aufgenommenen Szenen

8.2 Inkrementeller Aufbau

Im Entwurf des Ground-Truth Systems standen zwei Aspekte im Vordergrund: Es sollten zum einen unabhängig von Verdeckungseffekten Objektposen erfasst werden können und zum anderen einmal aufgenommene Posen in jedem Bild weiterverwendet werden können. Methodisch werden diese Aspekte unterstützt, indem Szenen grundsätzlich inkrementell aufgebaut werden. Objekte werden also nach und nach in der Szene platziert und direkt nach dem Platzieren aufgenommen. Diese Methodik ermöglicht es, Szenen mit starken Verdeckungseffekten zu erstellen. Beim Aufnehmen der Objekte sind die Objekte aber noch nicht zwangsläufig verdeckt, was die Aufnahme von Objekten wie Tellern oder ähnlich dicht stapelbaren Objekten deutlich vereinfacht. In nachfolgenden Bildern können die Objekte dann beliebig verdeckt werden. Solange sie nicht bewegt werden, bleibt ihre Pose bekannt.

Diese Eigenschaft ermöglicht weiterhin einen effizienten Arbeitsfluss, welcher den Annotieraufwand im Vergleich zur manuellen Annotation deutlich reduziert. Angenommen, jedes Objekt müsste pro Bild annotiert werden, so müssten für 2 400 Bilder mit durchschnittlich 16 Objekten pro Bild insgesamt $2400 \cdot 16 = 38400$ Objekte annotiert werden.

Abbildung 8.2: Pappschablone auf einem Objekt

Eine solche Abschätzung ist für das Ground-Truth System nicht notwendig, da die Anzahl der Annotationsvorgänge nicht von der Anzahl der Bilder abhängt. Stattdessen ist lediglich interessant, wie viele verschiedene Objekte in den Szenen platziert wurden. So wurden in den 2 400 Bildern des in dieser Arbeit aufgenommenen Datensatzes etwa 150 Objekte platziert. Dementsprechend wurde effektiv die Anzahl an benötigten Annotationen von 38 400 auf 150 reduziert. Insbesondere ist es möglich, nach dem Aufnehmen eines Objektes beliebig viele weitere annotierte Bilder aufzunehmen. In dem in dieser Arbeit erstellten Datensatz wurden in der Regel 16 Bilder aufgenommen, bevor ein neues Objekt in der Szene platziert wurde. Die Anzahl ist aber beliebig und kann für jeden Anwendungsfall variiert werden.

8.3 Schablonen als Hilfsmittel

Ein Hilfsmittel, welches sich für viele Objekte aus dem Datensatz anbietet, ist eine Pappschablone. Diese lässt sich beispielsweise auf Tassen, Tellern und Gläsern fixieren. Eine solche Schablone ist in Abbildung 8.2 zu sehen.

Die Schablone besteht aus zwei Teilen, welche zusammengeklebt sind. Der obere Teil wird, wie in dem Bild zu sehen ist, mit einem Koordinatensystem versehen, welches mit einem Stift auf die Schablone gezeichnet wird. Der untere Teil passt exakt in die Öffnung der Tasse, wodurch die Schablone nicht verrutschen kann. Das Koordinatensystem verläuft durch den Mittelpunkt der Schablone. Da die Mitte der Öffnung der Tasse den gleichen Mittelpunkt hat wie die Basis der Tasse, muss lediglich die Höhe der Schablonenoberfläche gemessen werden, um von dem Mittelpunkt der Schablone auf den Ursprung des Objekts

zu schließen. Der Ursprung von allen Objekten liegt dabei immer in der Auflageebene des Objekts auf dem Tisch. Bei Objekten mit symmetrischen Formen wie zum Beispiel diese Tasse, liegt der Ursprung weiterhin immer im Mittelpunkt dieser Form.

Nun ist es sehr leicht, die Positionen der Marker auf dieser Schablone auszumessen. Die Marker haben lediglich eine Verschiebung entlang der x- und y-Achse. Diese lässt sich leicht mit einem Lineal oder einem Messschieber ausmessen. Die Höhe der Schablone ist ebenso leicht zu bestimmen. Auf diesen Schablonen lassen sich auch gut die reflektierenden Marker anbringen, um entsprechend die schnellen und präzisen Messungen auszunutzen. Ein weiterer Vorteil dieser Schablone ist, dass die Marker nicht direkt auf dem Objekt kleben. Die Schablone kann nach dem Ausmessen vom Objekt entfernt werden. Alle Objekte, für die sich solche Schablonen anfertigen lassen, wurden entsprechend mit ihnen ausgemessen. Für Objekte wie zum Beispiel Besteck lassen sich solche Schablonen allerdings nicht anfertigen, dort müssen die Punkte auf andere Weisen markiert werden.

Mit diesen Pappschablonen lässt sich außerdem eine weitere Methode zum Aufnehmen von Objekten umsetzen. Zu Anfang der Arbeit wurden die sogenannten Plattform-basierten Ground-Truth Systeme vorgestellt, in welchem beispielsweise Objekte in Fassungen gestellt werden können, um so von der Position der Fassung auf die Pose des Objekts schließen zu können. Während der Aufwand zu groß wäre, solche Fassungen für so viele verschiedene Objekte zu erstellen, lässt sich eine ähnliche Idee leicht mit einem Stift umsetzen. Die Sekt-, Rotwein- und Weißweingläser haben alle die gleiche Basis. Sie stehen auf einem runden kreisförmigen Glasständer. Die Position dieser Glasständer in der Schublade oder im Schrank kann mit einem Stift markiert werden, indem entlang des Randes der Kreis auf den Boden gemalt wird. Nun wird lediglich eine Schablone benötigt, welche die selbe Größe hat wie der Glasständer. Diese Schablone wird in die Markierung gelegt und ausgemessen. Anschließend kann dort ein Glas hineingestellt werden. Diese Art, Objekte aufzunehmen, ist hilfreich, wenn beispielsweise kein Platz über den Objekten ist, um die Schablonen aufzunehmen, während sie in den Öffnungen der Gläsern liegen. Zusätzlich lassen sich Objekte zwischenzeitlich entfernen und später wieder platzieren, ohne dass ihre Pose neu erfasst werden muss. Dadurch wird das Aufnehmen von Objekten auf engem Raum erleichtert, wo es sonst schwierig wäre, die Marker aufzunehmen ohne die anderen Objekte zu verschieben.

8.4 Vergleich der entwickelten Verfahren in der Anwendung

Der Grund für die Entwicklung verschiedener Verfahren zum Erfassen der Objektposen ist, dass alle entwickelten Verfahren bestimmte Stärken und Schwächen haben. Teilweise sind Verfahren in bestimmten Situationen gar nicht anwendbar, während sie sich in anderen Situationen besondern gut eignen. Dieses Kapitel gibt einen Überblick über diese Stärken und Schwächen, um in der Entscheidung zu helfen, welches Verfahren wann genutzt werden sollte.

Die Kreismarkererkennung ist ein sehr umfassendes Werkzeug zum Aufnehmen von Objektposen. Kreismarker können an den meisten Objekten angebracht werden, und überall am Objekt von der Smartphonekamera aufgenommen werden. Das Verfahren ist nicht abhängig von menschlichen Einflüssen wie dem Zittern der Hand, da sich dieses Zittern im Bild widerspiegelt und trotzdem der korrekte Punkte erkannt wird. Nachteile der Kreismarkererkennung sind zum einen die Abhängigkeit von Kreismarkern auf den Objekten und zum anderen die im Vergleich zu den anderen Verfahren niedrige Genauigkeit. Die Abhängigkeit von Kreismarkern auf den Objekten lässt sich zum Teil durch die Pappschablonen vermeiden, dies ist aber nicht für alle Objekte anwendbar. Weiterhin lassen sich mit diesem Verfahren nur Punkte aufnehmen, solange sie nicht mehr als 40 Grad zu der Kamera geneigt sind. In einigen Situationen, in denen nur wenig Platz in der Szene ist, kann dies eine Einschränkung sein.

Auch mit dem Taststab können in sehr vielen Situationen die Objekte aufgenommen werden. Tatsächlich lässt der Taststab es zu, in nahezu allen Situationen Punkte aufzunehmen. Es gibt hier keine Einschränkungen, aus welchen Winkeln Punkte aufgenommen werden können. Auch ist dieses Verfahren nicht von Markern auf den Objekten abhängig. Die Punkte können auch beispielsweise mit einem Stift markiert werden. Dabei zeichnet sich diese Methode zusätzlich in der Evaluation durch eine gute Genauigkeit aus. Problematisch ist, dass die Genauigkeit stark vom Annotator abhängt. Zittert die Hand etwas, in welcher das Smartphone gehalten wird, so bewegt sich auch der Taststab. Dadurch kann es zu inkorrekten Aufnahmen kommen. Auch können beim Zittern aus Versehen Objekte bewegt werden. Werden dadurch auch andere, bereits aufgenommene Objekte bewegt, so kann dies eine vollständige Szene unbrauchbar machen. Dieses Verfahren sollte daher immer mit großem Bedacht genutzt werden.

Die reflektierenden Marker zeichnen sich durch ihre sehr genaue Posenbestimmung aus. Dabei können die Objektpunkte mit einem Klick aufgenommen werden, ohne dass das Smartphone in der Szene benötigt wird, was den Aufnahmeprozess leicht beschleunigt. Diese Methode wird allerdings dadurch eingeschränkt, dass sie nur für wenige Objekte anwendbar ist. Die Objekte benötigen eine Fläche, auf welcher drei reflektierende Marker angebracht werden können. Diese Fläche muss immer den Infrarotkameras zugeneigt sein, da die Marker sonst nicht aufgenommen werden können. Richtet man die Infrarotkameras korrekt aus, so können sie die Marker auf Pappschablonen beispielsweise auf Tellern und Gläsern erkennen. Ist dies möglich, so lohnt es sich immer, dieses Verfahren zu nutzen. Für viele Szenen können die Infrarotkameras allerdings auch nicht beliebig platziert werden, da sie die gesamte Szene überblicken müssen. Deshalb ist das Verfahren nicht in allen Situationen anwendbar.

9 Fazit

In dieser Arbeit wurde ein System entwickelt, welches es erlaubt, mit einer 3D-Kamera aufgenommene Szenen mit Ground-Truth Daten zu annotieren. Diese Ground-Truth Daten bestehen aus einem Objekttypen und der Pose des Objekts im Koordinatensystem der 3D-Kamera. Das System wurde soweit automatisiert, dass jede Objektpose in der Szene einmalig beim Platzieren aufgenommen werden muss. In allen folgenden Aufnahmen wird die Pose automatisch in das Koordinatensystem der 3D-Kamera umgerechnet. Das System reduziert dadurch die benötigten Annotationsvorgänge erheblich. Wie viele Annotationen durchgeführt werden, hängt nicht mehr von der Anzahl der aufgenommenen Bilder ab. Der einzige Faktor ist die Anzahl der platzierten Objekte.

Das System arbeitet hierzu insgesamt mit drei Sensoren. Eine 3D-Kamera nimmt die Daten auf, welche annotiert werden sollen. Ein Smartphone liefert Objektposen und dient als Benutzerschnittstelle. Ein Infrarot-Tracking System verknüpft diese beiden Sensoren, wodurch es möglich ist, im Smartphone aufgenommene Objektposen in das Koordinatensystem der 3D-Kamera zu transformieren. Zusätzlich bietet das Tracking System ebenfalls die Möglichkeit, Objektposen aufzunehmen.

Insgesamt wurden vier verschiedene Verfahren entwickelt, Objektposen aufzunehmen, welche unterschiedliche Stärken und Schwächen aufweisen. Das Verfahren Marker3D nutzt eine in dieser Arbeit entwickelte Kreiserkennung, um anhand von 3D-Positionen von Kreismarkern auf den Objekten ihre Posen zu bestimmen. Das Verfahren Marker2D hingegen nutzt die Pixelpositionen der Kreismarker im Bild der Smartphonekamera, um diese Berechnung durchzuführen. Weiterhin wurde ein Taststab entworfen, mit welchem die Punkte auf den Objekten berührt werden können, um so ihre Position aufzunehmen. Das letzte entwickelte Verfahren, MarkerRef, nutzt reflektierende Marker, welche von dem Infrarot-Tracking System aufgenommen werden, um daraus die Objektpose zu bestimmen.

In der Evaluation wurde betrachtet, ob diese entwickelten Verfahren genau genug sind, um Ground-Truth Daten für die Metriken 2 Grad 2 Zentimeter, 5 Grad 5 Zentimeter und 10 Grad 10 Zentimeter zu liefern. Dabei hat sich herausgestellt, dass das Verfahren MarkerRef genutzt werden kann, um Ground-Truth Daten für alle diese Metriken zu erstellen. Die Verfahren Marker3D und Taststab können für die Metrik 5 Grad 5 Zentimeter eingesetzt werden und entsprechend für alle Metriken die ein weniger genaues Ergebnis fordern. Das Verfahren Marker2D hat sich als zu ungenau für das Aufnehmen von Ground-Truth Daten herausgestellt.

Im Umfang dieser Arbeit wurde ein Datensatz mit neun verschiedenen Szenen aufgenommen, welche insgesamt 2 400 Bilder mit annotierten Objektposen enthalten. Die

Szenen werden dabei schrittweise aufgebaut, wodurch nach und nach Verdeckungssituationen entstehen, welche sich über den gesamten Aufnahmeprozess durch neue Objekte mehrfach ändern können.

Während in dieser Arbeit bereits ein nutzbares Ground-Truth System entstanden ist, mit welchem Datensätze aufgenommen werden können, gibt es insbesondere in der Benutzung des Systems noch einige Verbesserungsmöglichkeiten.

Die Umrechnung der Objektposen in das Koordinatensystem der 3D-Kamera ist bereits vollständig automatisiert. Die 3D-Kamera steht aktuell allerdings auf einem Stativ, da nicht garantiert werden kann, dass die Bilder der 3D-Kamera zum gleichen Zeitpunkt aufgenommen wurden wie die Posen der Tracking-Targets. Sind diese nicht synchron, so kommt es bei Bewegungen zu Fehlern der Objektposen im Kamerabild. Dieses Problem ließe sich durch eine Zeitsynchronisation zwischen der 3D-Kamera und dem Tracking System lösen. Ließe sich die interne Uhr der beiden Sensoren synchronisieren, so könnten Messungen der beiden Systeme ausgewählt werden, die in etwa zur selben Zeit durchgeführt wurden. Dadurch ließe sich die 3D-Kamera frei in der Hand führen und jedes gelieferte Bild könnte mit Ground-Truth Daten annotiert werden. Es werden allerdings von dem Tracking System sowie von der Kinect keine ausreichend genauen Synchronisationsverfahren angeboten. Durch eine Lösung dieses Problems würde sich die Möglichkeit bieten, in zukünftigen Arbeiten die Aufnahmegeschwindigkeit des Systems deutlich zu verbessern.

Um die Genauigkeit des Tracking Systems besser auszunutzen, könnte ein neues Verfahren zur Aufnahme von Objektposen entwickelt werden, welches die Tracking-Targets verwendet, um Objektposen aufzunehmen. Diese Targets könnten an Schablonen angebracht und ihre Transformation zum Objektursprung ausgemessen werden. Sowohl in der Implementierung als auch in der praktischen Umsetzung ist dieser Ansatz allerdings eine Herausforderung. Die Anzahl an Targets ist begrenzt, weshalb sie zwischen Schablonen getauscht werden müssten. Dies erfordert Fassungen in den Schablonen, damit die Targets immer an derselben Stelle angebracht werden. In Pappschablonen wäre das schwierig umzusetzen. Weiterhin ist das Messen der Transformation zwischen Objekt und Tracking-Target eine Herausforderung. Nach Augenmaß Transformationen zu schätzen ist nahezu unmöglich. Hier müsste eine neue Lösung entwickelt werden.

Literatur

[1] Agarwal, Sameer; Mierle, Keir et al. *Ceres Solver*. http://ceres-solver.org. Accessed September 21, 2018.

[2] ART (Advanced Realtime Tracking). *ART TRACKPACK*. https://ar-tracking.com/products/tracking-systems/trackpacke/. Accessed September 21, 2018.

[3] ART (Advanced Realtime Tracking). *DTrack2*. https://ar-tracking.com/products/software/dtrack2/. Accessed October 11, 2018.

[4] ART (Advanced Realtime Tracking). *TreeTarget*. https://ar-tracking.com/products/markers-targets/targets/#uid-155-14. Accessed September 28, 2018.

[5] Arun, K Somani; Huang, Thomas S; Blostein, Steven D. „Least-squares fitting of two 3-D point sets“. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5 (1987), S. 698–700.

[6] Azad, Pedram; Asfour, Tamim; Dillmann, Rüdiger. „Stereo-based vs. monocular 6-dof pose estimation using point features: A quantitative comparison“. In: *Autonome Mobile Systeme*. Hrsg. von Dillmann, Rüdiger; Beyerer, Jürgen; Stiller, Christoph; Zöllner, Marius; Gindele, Tobias. Berlin Heidelberg: Springer, 2009, S. 41–48.

[7] Calli, B.; Walsman, A.; Singh, A.; Srinivasa, S.; Abbeel, P.; Dollar, A. M. „Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set“. In: *IEEE Robotics Automation Magazine* 22.3 (2015), S. 36–52.

[8] Frese, Udo. *Differenzbilder Trägheitsmomente Histogramm Automatischer Schwellwert*. https://svn-agbkb.informatik.uni-bremen.de/ufrese/teaching/ebv/slides/ebvpublic/slides/ebvfrese1303.pdf. Accessed September 28, 2018.

[9] Frese, Udo. *Vorlesungsreihe Echtzeitbildverarbeitung, Übungsbetrieb*. http://www.informatik.uni-bremen.de/agebv/de/VeranstaltungEBV15. Accessed October 17, 2018.

[10] Geiger, A.; Lenz, P.; Urtasun, R. „Are we ready for autonomous driving? The KITTI vision benchmark suite“. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, S. 3354–3361.

[11] Geiger, Andreas; Lenz, Philip; Stiller, Christoph; Urtasun, Raquel. http://www.cvlibs.net/datasets/kitti/video/kitti_trailer.jpg. Accessed September 28, 2018.

[12] Godil, Afzal; Eastman, Roger; Hong, T. „Ground Truth Systems for Object Recognition and Tracking“. In: *NIST Interagency/Internal Report (NISTIR 7923)* (2013).

[13] Hinterstoisser, Stefan; Lepetit, Vincent; Ilic, Slobodan; Holzer, Stefan; Bradski, Gary; Konolige, Kurt; Navab, Nassir. „Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes“. In: *Computer Vision – ACCV 2012*. Hrsg. von Lee, Kyoung Mu; Matsushita, Yasuyuki; Rehg, James M.; Hu, Zhanyi. Berlin, Heidelberg: Springer, 2013, S. 548–562.

[14] Ho, Nghia. *Finding optimal rotation and translation between corresponding 3D points*. `http://nghiaho.com/?page_id=671`. Accessed October 09, 2018.

[15] Ionescu, Horia. *The six degrees of freedom: forward/back, up/down, left/right, yaw, pitch, roll*. `https://en.wikipedia.org/wiki/Six_degrees_of_freedom#/media/File:6DOF_en.jpg`. Accessed November 04, 2018.

[16] Lachat, Elise; Macher, Hélène; Landes, Tania; Grussenmeyer, Pierre. „Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling“. In: *Remote Sensing* 7.10 (2015), S. 13070–13097.

[17] Li, Yi; Wang, Gu; Ji, Xiangyang; Xiang, Yu; Fox, Dieter. „DeepIM: Deep Iterative Matching for 6D Pose Estimation“. In: *Computer Vision – ECCV 2018*. Hrsg. von Ferrari, Vittorio; Hebert, Martial; Sminchisescu, Cristian; Weiss, Yair. Cham: Springer International Publishing, 2018, S. 695–711.

[18] Marvel, Jeremy A; Falco, Joe; Hong, Tsai. „Ground truth for evaluating 6 degrees of freedom pose estimation systems“. In: *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*. ACM, 2012, S. 69–74.

[19] Mihalcik, David; Doermann, David. „The design and implementation of ViPER“. In: *Technical report* (2003), S. 234–241.

[20] Mutto, Carlo Dal; Zanuttigh, Pietro; Cortelazzo, Guido M. *Time-of-flight cameras and microsoft kinect (TM)*. New York: Springer, 2012.

[21] Nikon. *iGPS - System für Messungen, Positionsbestimmungen und Ortungen in der gesamten Produktionsumgebung*. `http://www.nikonmetrology.com/de/product/igps`. Accessed September 21, 2018.

[22] OpenCV. *Camera calibration With OpenCV*. `https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html`. Accessed September 21, 2018.

[23] OpenCV. *Feature Detection*. `https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html`. Accessed October 17, 2018.

[24] OpenCV. *Introduction into Android Development*. https://docs.opencv.org/2.4/doc/tutorials/introduction/android_binary_package/android_dev_intro.html. Accessed October 23, 2018.

[25] OpenCV. *OpenCV Schachbrettmuster*. https://docs.opencv.org/3.4.1/dc/dbb/tutorial_py_calibration.html. Accessed September 21, 2018.

[26] Roboception. *3D Data*. https://roboception.com/en/rc_visard-en/. Accessed October 17, 2018.

[27] Roboception. *Stereokamera*. https://doc.rc-visard.com/latest/de/stereo_camera.html#planar-rectification. Accessed October 22, 2018.

[28] Robot Perception and Navigation Group. *android-camera-calibration*. https://github.com/rpng/android-camera-calibration. Accessed October 31, 2018.

[29] Samsung. *Samsung Galaxy J5 (2017)*. https://www.samsung.com/de/smartphones/galaxy-j5-j530f/SM-J530FZKDDBT/. Accessed October 22, 2018.

[30] Shotton, Jamie; Glocker, Ben; Zach, Christopher; Izadi, Shahram; Criminisi, Antonio; Fitzgibbon, Andrew. „Scene coordinate regression forests for camera relocalization in RGB-D images“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, S. 2930–2937.

[31] Wiedemeyer, Thiemo. *IAI Kinect2*. https://github.com/code-iai/iai_kinect2. Accessed September 21, 2018. University Bremen: Institute for Artificial Intelligence, 2014 – 2015.

[32] Zucker, Matt. *Unprojecting text with ellipses*. https://mzucker.github.io/2016/10/11/unprojecting-text-with-ellipses.html. Accessed October 23, 2018.