Universität Bremen

Augmented-Reality als Plattform zur Echtzeitvisualisierung von Algorithmen der Bildverarbeitung

Bachelorarbeit am Cognitive Systems Lab
Prof. Dr.-Ing. Tanja Schultz
Fachbereich 3
Universität Bremen

von

Jonah Klöckner

Betreuer:

Dr. Felix Putze

Gutachter:

Dr. Felix Putze
Prof. Dr.-Ing. Udo Frese

Tag der Anmeldung: 20. August 2019
Tag der Abgabe: 10. Dezember 2019

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Bremen, den 10. Dezember 2019

Zusammenfassung

Im Rahmen dieser Arbeit wurde ein System zur Echtzeitvisualisierung verschiedener Algorithmen der Bildverarbeitung in Augmented Reality (AR) entwickelt und anhand des Beispiels der Objekterkennung untersucht. Dieses wurde in einer Server-Client-Architektur entworfen. Der Server ist dabei für die Berechnung der Algorithmen der Bildverarbeitung zuständig. Der Client, bestehend aus einem AR-Headset, präsentiert die Ergebnisse der Berechnung in Echtzeit. Als AR-Headset kam die HoloLens von Microsoft zum Einsatz.

Unter Anwendung einer Objekterkennung auf Basis von YOLOv3 wurde dieses System zum einen in Form einer Nutzungsstudie unter der These, dass es Versuchspersonen einfacher fallen würde die Möglichkeiten und Grenzen des präsentierten Algorithmuses einzuschätzen, untersucht. Es zeigten sich einige Indikatoren, die zur Unterstützung dieser These beitrugen.
Zum Anderen wurde eine technische Analyse des entwickelten Systems (ebenfalls unter dem Anwendungsbeispiel der Objekterkennung) durchgeführt. Hierbei wurden vor allem verschiedene Latenzen des Systems untersucht, um dessen Performance objektiv einschätzen zu können.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

1.1 Motivation und Zielsetzung

Augmented Reality (AR) schlägt eine Brücke zwischen der vollständigen Immersion von virtueller Realität und der unveränderten und direkten Interaktion mit der echten Umgebung der Nutzenden, wodurch sich neue Interaktionsmöglichkeiten ergeben können. Diese reichen von unterstützenden Aufgaben während Operationen[1] über vereinfachte Navigation für Fußgänger in Smartphones[2] bis zu Anwendungen in der Fertigungsindustrie[3].
Eine weitere interessante Möglichkeit der Nutzung von AR besteht in der Visualisierung geeigneter Algorithmen. Dies könnte zu einem besseren Verständnis führen, da die Algorithmen interaktiv und intuitiv und zudem in Echtzeit dargestellt werden können. Hierfür eignen sich besonders Algorithmen der Bildverarbeitung, da diese direkt mit der von Kameras aufgenommenen Umgebung interagieren und sie im Rahmen der AR manipulieren können.

Darüber hinaus können diese Algorithmen nicht nur visualisiert werden, sondern ebenfalls, integriert in anderen interaktiven Applikationen, Anwendung finden. Beispielsweise könnte eine Echtzeit-Objekterkennung der Umgebung Hilfestellungen für Menschen mit Einschränkungen des Sehens bieten [EBF18]. Ein weiteres mögliches Einsatzgebiet bestünde darin, Menschen mit Gedächnisproblemen dahingehend zu unterstützen, als dass das System eine Karte der Umgebung mit möglichen Alltagsgegenständen anfertigt und die Position dieser auf Anfrage bereitstellen kann, um so die Lebensqualität in der eigenen Wohnung zu verbessern.
Auch für solche Anwendungsfälle soll das hier entwickelte System eine Plattform bilden.

[1] O. Tepper, H. Rudy, et al., „Mixed Reality with HoloLens: Where Virtual Reality Meets Augmented Reality in the Operating Room“, Plastic and Reconstructive Surgery, 2017, Vol. 140, Iss 5

[2] C. Drees, „Google Maps: AR-Live View und Reise-Funktionen für Smartphones“, 2019, URL: https://www.mobilegeeks.de/news/google-maps-ar-live-view-und-reise-funktionen-fuer-smartphones/

[3] D.W.F. van Krevelen, „Augmented Reality: Technologies, Applications, and Limitations“, 2007, Vrije Universiteit Amsterdam

1.2 Vorgehensweise

Um die Funktionsweise von verschiedenen Algorithmen der Bildverarbeitung (AdB) wie etwa Objekterkennung anschaulicher zu machen, gilt es eine modulare und wiederverwendbare Applikation zu entwerfen, welche in der Lage ist, ebenjene Algorithmen in Echtzeit und in einer für Augmented Reality sinnvollen Art zu Weise abzubilden.

Als Grundlage bzw. Plattform für die Darstellung der Augmented Reality Inhalte soll die von Microsoft entwickelte HoloLens dienen. Diese bietet sich auf Grund ihres hervorragenden Trackings und Spatial Mappings sowie dem Vorhandensein einer die Umgebung filmenden Kamera an.

Es ist zwar prinzipiell möglich Bildverarbeitung direkt auf der HoloLens zu betreiben, so kann beispielsweise OpenCV auf der HoloLens genutzt werden[4]. Auch das Verfolgen von Objekten zu medizinischen Zwecken mit Hilfe von Vuforia [vuf] [FJDV18] kann direkt auf der HoloLens berechnet werden.
Viele andere moderne AdB wie etwa Objekterkennung, Salienzberechnung oder Segmentierungen können jedoch sehr rechenaufwendig sein oder benötigen gar angepasste Hardware für eine echtzeitnahe Ausführung. Für solche Algorithmen wird die eigentliche Verarbeitung der von der HoloLens aufgenommen Bilder nicht von dieser vorgenommen werden können. Hierzu wird ein dedizierter Computer benötigt werden, um die Echtzeitfähigkeit des Gesamtsystems sicherzustellen. Nach der Berechnung der Algorithmen werden die aufbereiteten Ergebnisse zurück an die HoloLens gesendet. Auf Grund dessen ist eine möglichst verzögerungsarme Übertragung für alle anfallenden Daten zu finden, da eine erhöhte Verzögerung innerhalb des Systems die empfundene Stabilität maßgeblich beeinflusst [AHJ+01]. Als weiteres Ziel der Arbeit ist Bestimmung und Bewertung der Leistungsfähigkeit verschiedener Ansätze der Videoübertragung von der HoloLens und der Datenübertragung zu der HoloLens zu nennen.

Zur Evaluation der so entstandenen Visualisierung von bildverarbeitenden Algorithmen wurde eine Versuchsreihe mit Testpersonen durchgeführt, welche die subjektive Qualität des Systems anhand eines Fragebogens bewerteten und einen Vergleich der Visualisierung ohne Agumented Reality zogen. Neben allgemein auffallenden Verbesserungen und Einschränkungen des entwickelten Systems gegenüber einer Videopräsentation, wurde die Hypothese untersucht, dass es Versuchspersonen leichter fällt die Möglichkeiten und Grenzen eines Algorithmuses zu erkennen und vorherzusagen. Zur Untersuchung dieser Hypothese wurden mehrere Parcours aufgebaut, welche die Versuchspersonen mit der HoloLens und ihrer Echtzeitvisualisierung sowie als Videoaufnahme erlebten. Daraufhin bewerteten sie ihre Erfahrungen sowie die Leistungsfähigkeit der Systeme anhand eines Fragebogens.

Darüber hinaus erfolgte eine technische, quantitative Auswertung des Softwaresystems, bei der vor allem die Performance in Hinblick auf Verzögerungen der Übertragungen (über die Zeit), erkannte und gesendete Frames pro Sekunde sowie Stabilität der Algorithmen unter verschiedenen Lichtverhältnissen, variierender Dynamik der Objekte des Parcours bzw. der Versuchsperson untersucht wurde.

[4] HoloLensForCV: „ComputeOnDevice“ stellt eine Beispielapplikation für die Nutzung von OpenCV auf der HoloLens dar [Ols17].

2. Grundlagen

2.1 Augmented Reality

2.1.1 Definition

Als Augmented Reality (AR), zu Deutsch in etwa: „Erweiterte Realität“, bezeichnen Carmigiani et al. [CFA+11] die Erweiterung der physikalischen Umgebung mit computergenerierten Objekten bzw. Informationen in Echtzeit. In Abgrenzung zur virtuellen Realität bleiben bei AR stets Teile der echten Umgebung für die nutzende Person wahrnehmbar. Idealerweise sind dabei virtuelle Objekte jedoch nicht von ihren physikalisch existierenden Pendants zu unterscheiden.

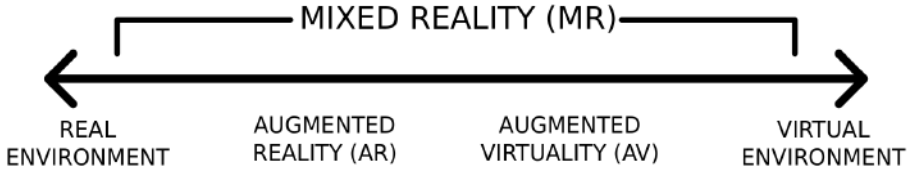


Abbildung 2.1: Realität-Virtualität Kontinuum [Vin11] nach Milgram [MTUK95]

Wie Abb. 2.1 zu entnehmen ist, ordnet sich AR nach Milgram's Realität-Virtualitäts Kontinuum zwischen der echten bzw. „normalen“ Realität (Real Environment) und der komplett immersiven Virtuellen Realität (Virtual Environment) ein. Da sich AR auf das Hinzufügen von virtuellen Objekten in die echte Umgebung beschränkt, ordnet Milgram sie näher der unveränderten Realität als der Virtuellen Realität zu [MTUK95].

Anders als Carmigiani et al. beschränkt sich diese Arbeit in ihrer Definition und Andwendung von AR auf rein visuelle Darstellungen. Damit folgt sie der Auslegung von R. T. Azuma [Azu97], wonach ein visuelles System als Augmented Reality zu bezeichnen ist, wenn folgende drei Eigenschaften gelten:

1. Kombination von Realität und Virtualität

2. Interaktion in Echtzeit
3. Dreidimensionalität

Dies löst die reine Definition von Augmented Reality von bestimmten Display-Technologien wie etwa Head-Mounted-Displays (HMD) los.

Generell kann im Feld von AR zwischen zwei verschiedenen Display-Technologien unterschieden werden[MTUK95]:

> **„See-through“ AR Bildschirme.** Zu Deutsch durchschaubare Bildschirme, bezeichnet Bildschirme deren Charakteristische Eigenschaft es ist, dass die nutzende Person durch sie hindurch ihre physikalische Umgebung wahrnehmen kann. Hierzu werden spezielle Bildschirme benötigt. Diese werden häufig direkt vor dem Gesicht des Nutzenden platziert, um ein möglichst großes Sichtfeld einzunehmen. In jenem Fall spricht man von „Head-Mounted-Displays“(HMD).
>
> **„Window-on-the-World“ AR Bildschirme.** Bei dieser Art der Darstellung werden die Computer-generierten Zusatzinformationen über ein Video einer Digitalkamera gelegt und anschließend auf einem konventionellen Bildschirm angezeigt. Hierbei sollte es sich um ein Live-Video handeln, da sonst keine Interaktion in Echtzeit möglich ist und somit die von Azuma aufgestellte, zweite Eigenschaft eines AR-Systems nicht gegeben ist.

Die Untersuchungen in dieser Arbeit beschränken sich auf AR in Form von „See-through“ Bildschirmen, sodass die HoloLens als Head-Mounted-Display ein ideales Beispiel für diesen Ansatz bildet.

2.1.2 Microsoft HoloLens

Abbildung 2.2: Microsoft HoloLens [Kre15]

Die HoloLens ist eine vom US-Amerikanischen Unternehmen Microsoft Inc. entwickelte Brille für Augmented Reality Anwendung und lässt sich als Kombination

aus einer Datenbrille und einem Head-Mounted-Display (HMD) beschreiben. Sie wurde Januar 2015 vorgestellt [Kre15] und ist seit Ende November 2016 auch in Deutschland verfügbar [Cen16].

Die HoloLens unterscheidet sich als AR-Headset von anderen Virtual Reality Headsets (VR-Headsets) dadurch, dass zu jedem Zeitpunkt Interaktion mit der realen Umgebung möglich ist, da dargestellte Objekte lediglich über das Sichtfeld des Benutzenden gelegt werden und zudem eine gewisse Transparenz aufweisen.

Als Betriebssystem der HoloLens kommt das hauseigene Windows 10 in einer angepassten Version zum Einsatz. Ausgeführt werden können hier nur Universal Windows Platform (UWP) kompatible Programme. Angetrieben wird die HoloLens von einem 32-Bit Intel Atom x5-Z8100 Prozessor, 2 Gigabyte RAM sowie einer nicht genauer beschriebenen von Microsoft entwickelten Grafiklösung. Für kabellose Konnektivität stehen Bluetooth und WLAN zur Verfügung [MZ18a][Jel16]. Dies ermöglicht einen Betrieb ohne zusätzlichen Computer und somit eine potenziell erhöhte Mobilität.

Neben diesen internen Spezifikationen beinhaltet die HoloLens Sensoren, welche die Umgebung erfassen und eine räumliche Zuordnung sowie Lage innerhalb dieses Raums ermöglichen. Diese bestehen aus einer Inertialen Messeinheit (IMU), welche mit Hilfe von Beschleunigungs- und Winkelgeschwindigkeitssensoren die Lage der HoloLens im Raum misst [Flu10]. Für die räumliche Zuordnung ist eine Time of Flight Kamera oder Tiefenkamera zuständig, welche die Entfernung der HoloLens zu ihrer Umgebung darstellen kann. Mit Hilfe dieser Tiefeninformationen ist es möglich Hologramme, wie Microsoft alle von der HoloLens gerenderten Objekte nennt, in einer bestimmten Entfernung anzuzeigen und diese entsprechend der Beschaffenheit des Raums zu platzieren und manipulieren. So können beispielsweise geöffnete Fenster scheinbar an die Zimmerwand gepinnt werden oder Teile von Hologrammen ausgeblendet werden, sollte sich ein Objekt zwischen ihrer angestrebten Entfernung und der die HoloLens tragenden Person befinden. Dieses Erfassen der Beschaffenheit des umliegenden Raumes wird von Microsoft Spatial Mapping genannt, da eine Karte der Umgebung angefertigt wird, welche jedem Punkt um der Person, die die HoloLens trägt, eine Entfernung zuordnet. Eine Visualisierung dessen ist in Abbildung 2.3 zu sehen. Dieses Problem wird allgemein auch als „Simultaneous Localization and Mapping“ oder SLAM-Problem bezeichnet[DB06].

Als Eingabemethoden dienen der HoloLens vornehmlich eine Kombination der, von der IMU bereitgestellten, Position im Raum und einer Gestensteuerung. Aus dieser Position berechnet die HoloLens die Blickrichtung der nutzenden Person. Diese wird als Punkt auf den Bildschirmen sichtbar gemacht, sodass ein intuitives Ausrichten der eigenen Blickrichtung möglich ist. Die Gestensteuerung erlaubt es sich im Mittelpunkt des Blickfelds befindliche Objekte

Abbildung 2.3: Visualisierung des Spatial Mappings der HoloLens.

auszuwählen. Die Erkennung der Gesten erfolgt über ein ständiges Verfolgen der Armbewegung über vier weitere Kameras [MZ19a]. Darüber hinaus gibt es eine Sprachsteuerung, welche ebenfalls in Zusammenhang mit der Gestensteuerung funktioniert, indem auf ein bestimmtes Hologramm geschaut wird und daraufhin die Beschriftungen aller Knöpfe, Menüs, usw. dieses Hologramms als Eingabe für Sprache dienen [MZ19c].

Von besonderer Relevanz für diese Arbeit ist eine Kamera, welche in Blickrichtung ausgerichtet ist. Die Kamera kann für die Objekterkennung genutzt werden, da sie mit zwei Megapixeln auflöst und somit das komplette Blickfeld des Nutzenden durch die Bildschirme aufnimmt. Sie befindet sich an der Vorderseite des Headsets mittig zwischen den Augen, in etwa einen Zentimeter über der Augenhöhe. Ihre genauere Positionierung ist auf Abb. 2.4 zu sehen.

Bei Videoaufnahmen ist sie in der Lage mit bis zu 1280 x 720 Pixeln aufzulösen und bietet dabei laut Microsoft ein horizontales Sichtfeld (fov) von 45°. Eigene Messungen ergaben ein fov von ungefähr 42,5°.

Abbildung 2.4: Frontansicht der HoloLens mit hervorgehobener Weltkamera.

Die beiden Bildschirme der HoloLens befinden sich in einem ungefähren Abstand von zwei Zentimetern vor den Augen der anwendenden Personen und lösen mit einer maximalen Auflösung von 1280 x 720 Pixel pro Auge auf [McC19]. Eigenen Messungen zufolge bieten diese Bildschirme ein horizontales Sichtfeld von ca. 30,5°.

2.2 Bildverarbeitung

In ihrem Werk „Image processing - the fundamentals“[PP10] zählt M. Petrou vier Ziele der Bildverarbeitung auf:

Bildverbesserung. Es wird versucht die subjektive Qualität eines Bildes zu erhöhen. Dies geschieht häufig über eine Erhöhung des Kontrastes.

Bildkompression. Ziel ist es ein Bild auf eine Datenmenge so klein wie möglich zu reduzieren und dabei die Qualität so wenig wie möglich zu beeinflussen.

Bildwiederherstellung verbessert die objektive Qualität eines Bildes. Dies kann zum Beispiel über eine Verminderung des Bildrauschens geschehen.

Merkmalsextraktion beschreibt das explizit Machen bestimmter Charakteristiken des Bildes. Diese können für das Erkennen bestimmter Inhalte des Bildes genutzt werden.

In dieser Arbeit wird es ausschließlich um Algorithmen der Merkmalsextraktion gehen, welche mit Hilfe von Augmented Reality (der HoloLens) visualisiert werden sollen.

2.2.1 Objekterkennung

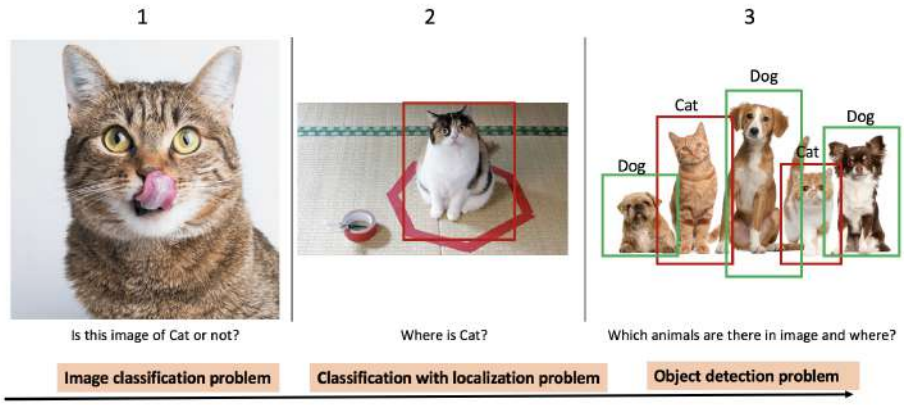


Abbildung 2.5: **Verschiedene Kategorien der Objekterkennung:** 1 zeigt Bildklassifikation, 2 zeigt lokalisierte Bildklassifikation, 3 zeigt Multi-Objekterkennung und Lokalisation

Eine solche Kategorie von Algorithmen der Merkmalsextraktion der Bildverarbeitung ist die Objekterkennung. Objekterkennung ist der Prozess des semantischen Einteilens von Bildern nach bestimmten Klassen. Das erkannte Objekt auf dem Bild entspricht hierbei der Klasse des Bildes. Auch hier kann zwischen drei Kategorien unterschieden werden:

- Bei der **Bildklassifikation** wird das gesamte Bild nach einem Objekt durchsucht und danach einer Klasse zugeordnet. Dies ist zu sehen in Abbildung 2.5 Bild 1
- Die **Objektklassifizierung und Lokalisation** bestimmt neben der Klasse des Bildes auch noch die Position des erkannten Objektes. Bild 2 in Abbildung 2.5.
- Die **Objektklassifizierung und Lokalisation mehrerer Objekte** erweitert diesen Ansatz insofern, als das auf dem Bild mehrere Objekte der gleichen oder unterschiedlichen Klasse geben kann, die jeweils einzeln gelabelt und lokalisiert werden. Im Folgenden wird diese Art einfach als Objekterkennung bezeichnet. In Abbildung 2.5 ist dies auf Bild 3 zu sehen.
- Die **Instanzsegmentierung** ist eine Variante der Objektklassifizierung und Lokalisation mehrerer Objekte bei der anstelle von Rechtecken Segmente also zusammenhängende Pixelregionen zur Positionsbestimmung den Output des Algorithmuses bilden.

Als die für diese Arbeit relevanteste Variante ist die Objektklassifizierung und Lokalisation mehrerer Objekte zu nennen, da Algorithmen dieser Art eine Visualisierung voraussetzen, welche auch für beide anderen Kategorien genutzt werden kann, indem jeweils nur ein Label entweder über das gesamte Bild oder über das lokalisierte Objekt angezeigt wird. Im Nachfolgenden werden einige dieser Algorithmen vorgestellt.

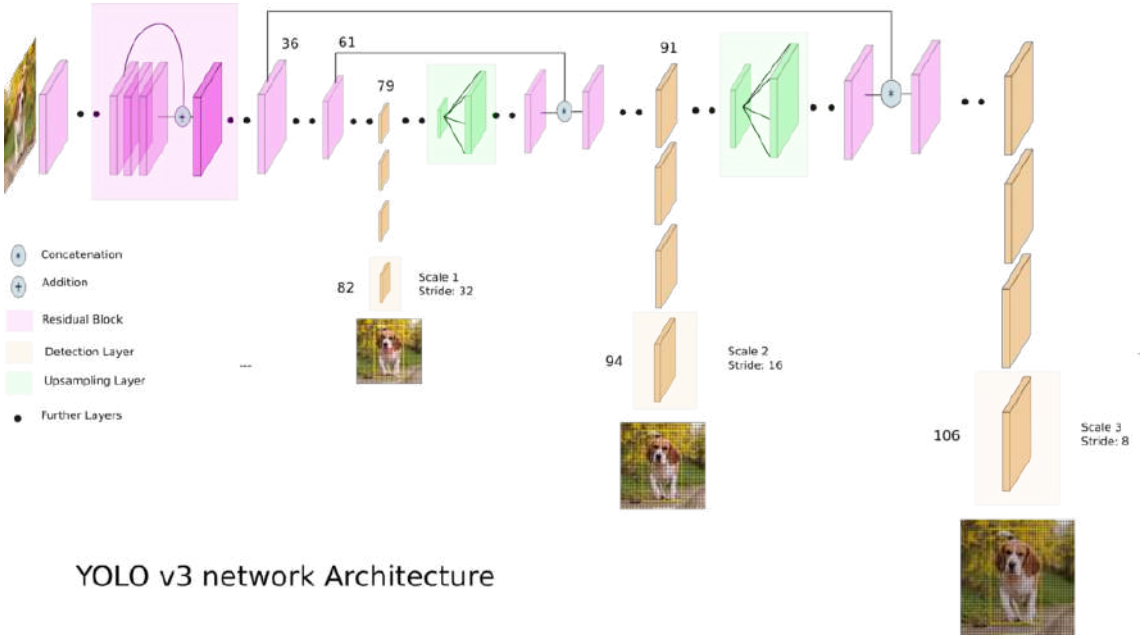


Abbildung 2.6: Netzwerkarchitektur von YOLOv3 [Kat18]

YOLOv3

YOLOv3 steht für „You only look once“ Version 3 und ist ein von Redmon et al. entwickelter Algorithmus zur Objekterkennung in Bildern [Red18].
Es handelt sich um ein Convolutional Neural Network mit insgesamt 106 Layern, welches neben den Convolutional Layern auch Residuale Layer mit überspringenden Verbindungen sowie upsampling in die Netzstruktur integriert, wie in Abb. 2.6 zu sehen ist [Kat18].
Bei einer Bildgröße von 416 x 416 Pixeln benötigt YOLOv3 29ms pro Bild auf einer Nvidia Titan X Grafikkarte, sodass in dieser Konfiguration eine Echtzeitverarbeitung mit über 30 Bildern pro Sekunde möglich ist [Red18].
Im Unterschied zu vielen anderen Objekterkennungsalgorithmen setzt YOLO weder auf sliding windows, bei denen ein Klassifikator in festen Abständen das gesamte Bild oder bestimmte Regionen vom Bild klassifiziert, noch auf region proposal Methoden, welche erst potenzielle Bounding Boxen erkennen, um diese in Nachverarbeitungsprozessen zu verbessern. In YOLO wird das Objekterkennungsproblem nicht als Klassifikationsproblem, sondern als Regressionsproblem betrachtet. Hierbei durchläuft das gesamte Bild das neuronale Netzwerk nur ein einziges Mal [RG15].

Die ersten 53 Layer des kompletten Netzes sind für die feature extraction bestimmt und heißen Darknet-53. Sie bestehen aus 1×1 und 3×3 Convolutional Layern sowie fünf residual Layern mit denen Layer übersprungen werden können.
Danach folgen weitere 53 Layer für die Objektdetektierung. Dabei wird das Bild in den Layern 82, 94 und 106 in drei unterschiedlichen Größen detektiert, nachdem es in vorherigen Layern skaliert wurde. Für jede Größe wird eine mögliche Klasse vorhergesagt. Dazu unterteilt YOLO die aktuelle feature map in $13 \cdot 13$ Boxen. Jede Box ist für die Erkennung einer Klasse zuständig und erstellt dafür mehrere potenzielle Bounding Boxen [Nay18]. Diese enthalten a priori Wissen über die jeweiligen Klassen, welches während des Trainings aus den Labeln des Trainingssets durch Clustering mit Hilfe des k-Means-Algorithmus ermittelt wird [RF16]. Auf diese potenziellen

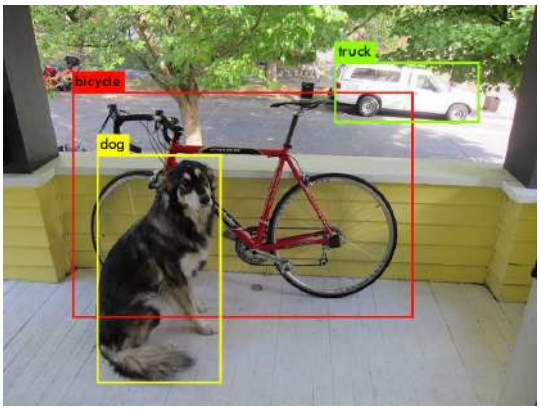


Abbildung 2.7: Beispielhafte Detektion von YOLOv3

Bounding Boxen wird nun eine $1 \cdot 1 \cdot (B \cdot (5 + C))$ große Faltungsmatrix angewandt. B entspricht hierbei der Anzahl der Bounding Boxen pro Box. Die 5 rührt von den vier Koordinaten der Bounding Box plus die vorhergesagte Wahrscheinlichkeit, dass in dieser Box ein Objekt zu sehen ist her und das C entspricht der Anzahl der möglichen Klassen. In dieser Arbeit wird YOLO mit dem COCO-Dataset verwendet[1], sodass das Netz zwischen 80 Klassen unterscheiden muss [Red18][Kat18]. Das „COCO dataset“ (Common Objects in Context) ist ein Datensatz bestehend aus 328.000 Bildern, welche mit insgesamt 2.5 Millionen Labeln versehen sind. Auf den Bildern befinden sich Objekte, welche von einem durchschnittlichem vierjährigen Kind erkannt werden können [LMB+14].

Diese Faltungsmatrix wird bei allen drei Detektionslayern (82, 94, 106) ausgeführt. Durch das vorherige Skalieren des Bildes ist gewährleistet, dass das Netz sowohl kleinere als auch größere Objekte erkennt.

Nach allen drei Detektionen gibt das Netz als Ausgabe für jede Box die Wahrscheinlichkeit, dass sich dort ein Objekt befindet, an. Sollte dieser Wert über einen bestimmten Schwellwert steigen, so wird zusätzlich die wahrscheinlichste Klasse und die Koordinaten der Bounding Box als Punkt um die Mitte sowie Breite und Höhe der Bounding Box angegeben [Kat18]. Eine beispielhafte Ausgabe ist in Abb. 2.7 zu sehen.

Weitere State-of-the-Art-Algorithmen

Neben YOLOv3, welches in dieser Arbeit als Beispiel für ein merkmalsextrahierenden Algorithmus der Bildverarbeitung genutzt wird, existieren noch weitere State-of-the-Art-Algorithmen, mit ähnlichen Zielen und Anwendungsgebieten.

R-CNN/Fast-CNN R-CNN steht für „Regions with Convolutional Neural Network features“ und bezeichnet einen von R. Girshick et al. [GDDM14] entwickelten Algorithmus zur Detektion und Lokalisierung von Objekten in Bildern.
Um die Position von Objekten im Bild zu bestimmen teilt der R-CNN Algorithmus dieses zunächst in viele Kandidaten für Regionen auf. Danach werden ähnliche Regionen vereinigt, um auch größere Objekte erkennen zu können. Aus diesen teilweise vereinigten Regionskandidaten werden 2000 Regionen gebildet, in denen, mit Hilfe

[1] Coco Dataset: http://cocodataset.org/.

eines Convolutional Neural Networks, Feature Vektoren erzeugt werden. Aus diesen Feature Vektoren sagt eine Support Vector Machine (SVM) für jede Region voraus, ob und welche in dieser Region abgebildet ist [Gan18].
„Fast R-CNN“ stellt eine Verbesserung dieses Algorithmuses in Hinblick auf die Trainings- und Inferenzzeit dar und wurde ebenfalls von R. Girshick et al. entwickelt. Indem darauf verzichtet wurde, jede der 2000 Regionen durch das CNN zu schicken und stattdessen mit einer Inferenz des CNN eine Feature Map zu erzeugen, konnte ein Zuwachs an Geschwindigkeit im Training und der Inferenz errungen werden.

RetinaNet RentinaNet ist ein Objekterkennungsalgorithmus, welcher aus einem vereinigten neuronalen Netzwerk besteht. Dieses setzt sich aus einem Hauptnetzwerk und zwei aufgabenspezifischen Netzwerken zusammen. Das Hauptnetzwerk ist ein Standard Convolutional Neuronal Network und ist für die Generierung einer Feature Map zuständig. Hierfür wird ResNet50 oder ResNet101 genutzt. Die beiden aufgabenspezifischen Netzwerke dienen der Klassifikation und Lokalisierung der im Hauptnetzwerk erzeugten Features [Jay18].

SSD: Single Shot MultiBox Detector Der Single Shot MultiBox Detector [LAE+16] oder kurz SSD basiert ebenfalls auf einem Convolutional Neural Network und lässt sich in zwei Komponenten unterteilen. Der Name single Shot leitet sich hierbei von dem verwendeten Prinzip ab, ein Bild nach nur einem unidirektionalen Durchlauf durch das neuronale Netz zu Klassifizieren und Lokalisieren.
In einem ersten Schritt werden Feature Maps extrahiert. Hierfür wird ein Neuronales Netz mit dem Namen „VGG16“ verwendet, welches ein neuronales Netzwerk zur Klassifikation von Bildern ist und aus 13 Convolutional Layern mit 3×3 Kerneln besteht [SZ14].

Als nächstes werden mit Hilfe der Feature Maps in weiteren Convolutional Layern Objekte detektiert. Hierbei wird das Bild in 38×38 Zellen unterteilt. Für jede Zelle werden vier Objektvorschläge generiert. Für jeden dieser Vorschläge existiert eine Bewertung der Wahrscheinlichkeit jeder Klasse (plus keine Klasse), die Klasse mit der höchsten Wahrscheinlichkeit wird gewählt.
Um die Größe der Bounding-Box besser abschätzen zu können verwendet SSD sog. „Default boundary boxes“, welche a priori Wissen über die Form von bestimmten Objekten in die Erkennung während des Trainings einfließen lassen.

Mask-RCNN Eine weitere Variante der Objekterkennung ist die Instanzsegmentierung (siehe Abschnitt 2.2.1). Ein Algorithmus hierfür bildet der von K. He et al. [HGDG17] entwickelte Algorithmus „Mask-RCNN“. Im Gegensatz zum oben beschriebenen R-CNN/Fast-RCNN Algorithmus wurde eine weitere Ausgabe hinzugefügt, welche die Maske des Bildes beschreibt. Dies geschieht mit zwei weiteren Convolutional Layern. Um die Maske bzw. die Segmentierung besser berechnen zu können, wurde das „ROI-Pooling“ genannte Zusammenfassen von Regionen von Interesse weiter verfeinert, indem Zellen nicht an der Input Feature Map ausgerichtet werden müssen, um bei diesem Prozess weniger Informationen zu verlieren und eine bessere Accuracy zu erreichen[Hui18].

Warum wurde YOLOv3 gewählt?

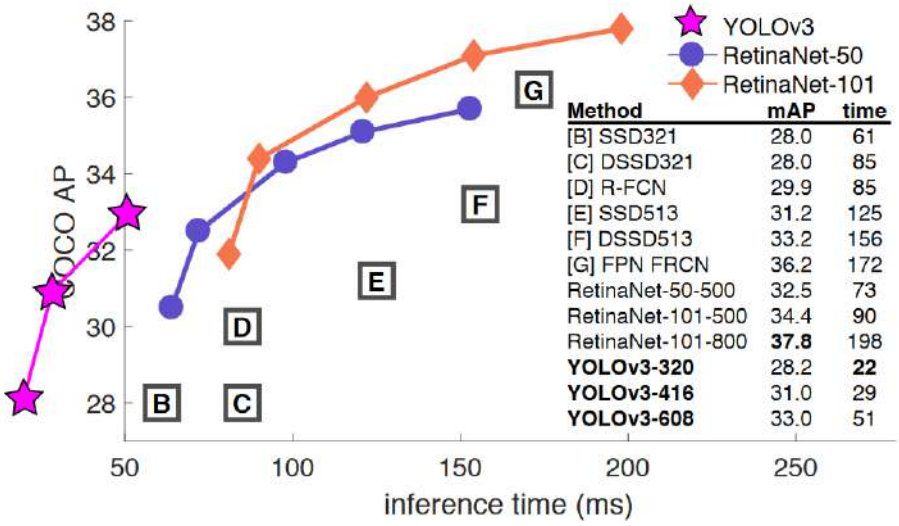


Abbildung 2.8: Vergleich der Performance verschiedener Objekterkenner über der Berechnungszeit pro Bild[Red18]

Vergleicht man YOLOv3 mit anderen State-of-the-Art Objekterkennungsalgorithmen wie RetinaNet so fällt auf, dass YOLOv3 zwar eine schlechtere mean Average Precision auf dem coco-Dataset mit 28.2 bis 33.0 je nach Netzkonfiguration aufweist als beispielsweise RetinaNet mit 32.5 bis 37.8, dabei jedoch im Durchschnitt etwa um den Faktor 3.5 schneller ist als RetinaNet, wie in Abbildung 2.8 zu sehen ist.

Da Echtzeitfähigkeit eine wichtige Rolle für das angestrebte System darstellt, wurde YOLOv3 als Beispielalgorithmus zur Visualisierung verwendet.

2.3 Lab Streaming Layer (LSL)

In dieser Arbeit wird LSL bzw. HoloLSL zur Übertragung von zeitunsynchonisierten Daten zwischen der HoloLens und dem Companion-Computer (Server) verwendet.

Das Projekt LSL beschreibt sich auf seiner Github-Wiki Seite folgendermaßen [Bou18]:

> The lab streaming layer (LSL) is a system for the unified collection of measurement time series in research experiments that handles both the networking, time-synchronization, (near-) real-time access as well as optionally the centralized collection, viewing and disk recording of the data.

Zu Deutsch etwa: Der Lab Streaming Layer (LSL) ist ein System für das einheitliche Sammeln von Messzeitreihen in wissenschaftlichen Experimenten und kümmert sich sowohl um Netzwerkangelegenheiten, Zeitsynchronisation, einen (nahezu) Echtzeitzugriff auf die Daten, sowie der optionalen zentralen Sammlung, Möglichkeit der Betrachtung und Speicherung der Daten.

Im LSL können Messgeräte oder andere Daten produzierende Geräte sog. „outlets“ erstellen. In diese können dann regel- oder unregelmäßig Datenpakete mit einer vorher festgelegten Struktur geschrieben werden. Die outlets sind für alle sich im gleichen Netzwerk befindlichen Computer sichtbar. Sogenannte „inlets“ können diese outlets empfangen und weiterverarbeiten, also zum Beispiel persistieren.

2.3.1 HoloLSL

HoloLSL ist eine von Sebastian Bliefert und Felix Kroll entwickelte Anbindung der HoloLens an LSL. Sie besteht aus zwei Komponenten [Kro18]:

- Der **Python Server** implementiert ein spezielles, auf TCP basierendes Protokoll. Der Server erhält eine Liste von LSL inlets sowie LSL outlets. Alle Daten, die den Server über ein inlet erreichen, werden in das benutzerdefinierte Protokoll umgewandelt und an die HoloLens gesendet. An alle definierten LSL outlets werden immer dann Daten gesandt, wenn diese den Server über das benutzerdefinierte Protokoll von der HoloLens erreichen.
- Der **HoloLens Client** bildet die Gegenseite zum Python Server. Er stellt APIs in Unity zum Senden und Empfangen von Daten vom Server zur Verfügung.

2.4 Verwandte Arbeiten

M. Eckert, M. Blex et al. [EBF18] untersuchten 2018 die Verwendung der HoloLens in Verbindung mit Objekterkennung der YOLOv2 um Menschen mit Sehbehinderungen eine auditive Schnittstelle zur Positions- und Distanzangabe von Objekten zu ermöglichen. Hierzu wurde eine Server-Client-Struktur entwickelt, die HoloLens bildet dabei den Client, der für die Darstellung der vom Server durchgeführten Objekterkennung zuständig ist.
Dies ähnelt in der technischen Struktur dem in dieser Arbeit gewähltem Ansatz, da auch Eckert, Blex et al. das Problem der ungenügenden Rechenleistung der HoloLens zu lösen versuchen. Allerdings verzichteten sie dabei, auf Grund ihres gewünschten Anwendungszwecks, auf eine visuelle Umsetzung der Objekterkennung sowie auf einen Vergleich mit konventionellen Visualisierungen der Objekterkennung.

Y. Park, V. Lepetit und W. Woo. [PLW08] zeigen eine Möglichkeit des Trackens von Objekten im dreidimensionalen Raum von Augmented Reality auf. Dies geschieht über das Anfertigen von 3D-Scans und Referenzbildern der Objekte, die dann mit Hilfe von Methoden der Featureerkennung Frame für Frame verfolgt werden können. Welche Ojekte genau verfolgt werden wird jedoch nicht weiter erfasst. Des Weiteren schränkt die Notwendigkeit des vorherigen Anfertigens von 3D-Scans die Nutzbarkeit weiter ein.

C. M. M. Mojica, N. V. Navkar et al. [MNT+17] entwickelten 2017 eine Echtzeitvisualisierung für durch Magnetresonanztomographie (MRT) entstandene Bilder mit Hilfe der Microsoft HoloLens. Diese Bilder sowie daraus berechnete Renderings sollen während Operationen am Gehirn in Echtzeit an ihren entsprechenden Stellen der zu behandelnden Person sichtbar gemacht werden. Des Weiteren können die Bilder und verarbeiteten Informationen mittels Gesten zu Planungszwecken manipuliert werden. Diese Arbeit richtet sich im Gegensatz dazu nicht direkt an einen bestimmten Anwendungsfall, sondern möchte eine allgemeinere Plattform für die Visualisierung von Bildverarbeitungsalgorithmen schaffen, sodass die allgemeinere Darstellungsform durch beschriftete Rechtecke gewählt wurde. Eine direkte Interaktion mit der Visualisierung mittels Gesten ist mit der in dieser Arbeit vorgestellten Methode jedoch nicht möglich.

3. Entwicklungsprozess der Software-Plattform

3.1 Problematik

Ein generelles Problem der Echtzeitvisualisierung von Algorithmen der Bildverarbeitung (AdB) auf AR-Headsets wie der HoloLens besteht in der beschränkten Rechenleistung dieser Headsets und der Rechenkomplexität vieler gewünschter Algorithmen.
Diese eingeschränkte Rechenleistung lässt sich auf die Mobilität der Headsets zurückführen, sodass nur Hardware mit geringer Leistungsaufnahme verwendet werden kann. Als Lösung dieses Konfliktes kann eine Server-Client-Architektur dienen. Der Server in Form eines rechenstarken Computers übernimmt dabei die Berechnung der AdB. Das AR-Headset stellt lediglich dem Server die benötigten Bilddaten zur Verfügung und nach der Verarbeitung dieser die Ergebnisse dar.

Diese Arbeit konzentriert sich vor allem auf Objekterkennung, sodass, sollte das in diesem Rahmen entstandene System für andere Algorithmen angewandt werden, folgende Bedingungen für alle darzustellenden AdB gelten:

1. Es handelt sich um einen merkmalsextrahierenden Algorithmus (Genaueres in Abschnitt 2.2).
2. Das Ergebnis des Algorithmus lässt sich als minimal umspannendes Rechteck um ein Teil des Bildes darstellen.
3. Der Algorithmus produziert für subjektiv ähnliche Eingaben ähnliche Ausgaben, ist also stabil gegenüber Bildrauschen, veränderten Lichtverhältnissen, kleineren Bewegungen der Kamera oder ähnlichen Artefakten der digitalen Videographie.

Diese Einschränkungen sind neben der Fokussierung auf Objekterkennung der geringen Rechenleistung des verwendeten AR-Headsets geschuldet, die ein Tracking der vorhandenen Rechtecke nötig macht (Siehe auch Abschnitt 3.4).

Als weitere AdB, die diese Bedingungen erfüllen, ließen sich somit neben der Objekterkennung noch Gesichtserkennung, Salienzhervorhebung oder Bewegungserkennung nennen.

3.2 Aufbau

Das gesamte Software-System lässt sich in zwei Hauptkomponenten unterteilen:

Die PC-Applikation empfängt einen Video-Stream der HoloLens, wendet auf diesem den gewählten Bildverarbeitungsalgorithmus an und schickt die Ergebnisse zurück an die HoloLens. Dies entspricht dem Server der oben genannten Client-Server-Architektur.

Die HoloLens-Applikation sendet ein Video-Stream der Weltkamera der HoloLens an den Server und wartet auf dessen Ergebnisse. Sind diese verfügbar, stellt die HoloLens-Applikation sie, mit Hilfe ihrer Sensorik zum Spatial-Mapping, im dreidimensionalen Raum dar.

Abbildung 3.1 zeigt dieses Zusammenspiel anhand des Beispiels der Objekterkennung als UML-Sequenzdiagramm.

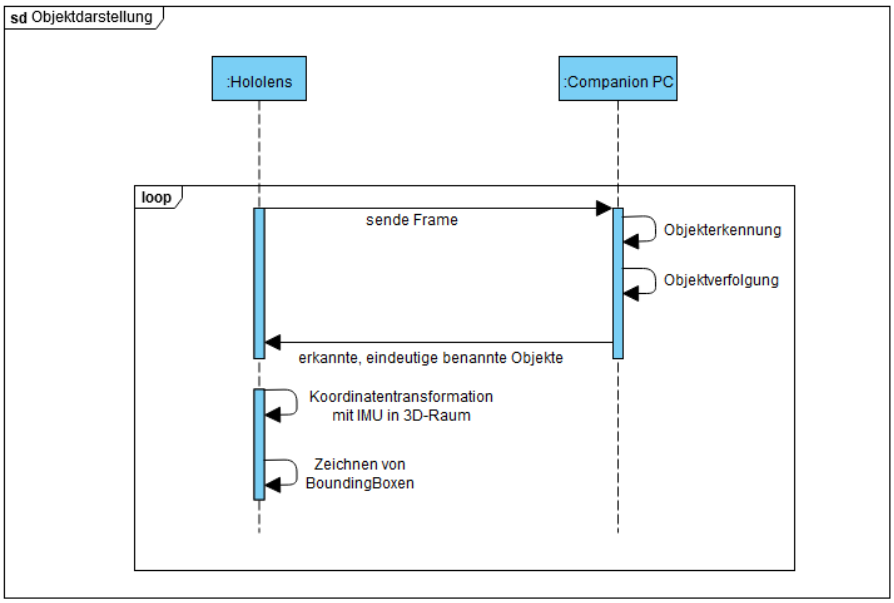


Abbildung 3.1: Kommunikationsablauf PC-HoloLens am Beispiel Objekterkennung

3.2.1 Aufbau der PC-Applikation

Die PC-Anwendung wurde in Python geschrieben und ist für das Empfangen von Bilddaten der HoloLens, das Verarbeiten dieser und das Senden von Ergebnissen zurück an die HoloLens, verantwortlich. Für die Übermittlung der Daten wurden während des Entwicklungsprozesses verschiedene Verfahren entwickelt:

1. Einzelbilder über Unity-APIs
2. Reseach-Mode
3. Device-Portal

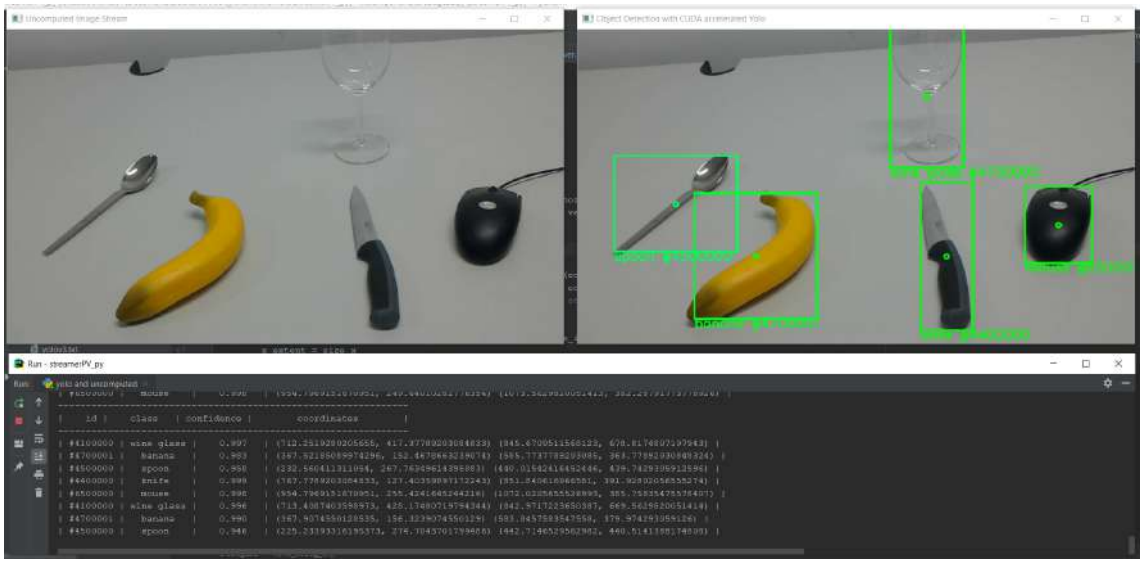


Abbildung 3.2: Die PC-Python-Applikation mit unverarbeitetem Videostream sowie mit Objekterkennung

Genauere Beschreibungen und Vor- sowie Nachteile der jeweiligen Modi befinden sich im Abschnitt 3.3. Letztlich gewählt wurde der Übertragungsmodus über das Device-Portal, da hier die beste Performance gemessen werden konnte.

Ist ein Frame empfangen worden, so leitet die Anwendung diesen an jeden, vor dem Start gewählten, Bildverarbeitungsalgorithmus weiter. Das Ergebnis dieser Berechnungen wird für jeden Algorithmus dargestellt, wie in Abbildung 3.2 zu sehen ist. Diese Ergebnisse werden, wie in Abschnitt 3.4 erläutert, über mehrere Frames verfolgt, um so Rechenzeit auf der HoloLens zu sparen. Danach überträgt die Anwendung die Ergebnisse mit Hilfe von HoloLSL zurück an die HoloLens. Ein einzelnes Ergebnis besteht aus zwei Punktkoordinaten, um welche ein Rechteck gespannt wird, bei Objekterkennung der erkannten Klassse sowie der durch das Tracking generierten einzigartigen Identifikationsnummer (ID) des Objekts. Genaueres hierzu befindet sich im Abschnitt 3.3.4.

3.2.2 Aufbau der HoloLens-Applikation

Die Hauptaufgaben der (clientseitigen) HoloLens-Applikation bestehen aus dem Bereitstellen bzw. Senden des Kamerastreams sowie aus dem Darstellen der Ergebnisse des Servers im dreidimensionalen Raum. Wie in Abschnitt 3.3 beschrieben, fragt der Server den Videostream über eine Web-API ab, sodass dies nicht explizit gehandhabt werden muss.

Das eigentliche Programm ist eine in Unity[1] mit Hilfe der Bibliothek HoloToolKit[2] geschriebene Anwendung, die für die Darstellung und Platzierung der empfangenen Ergebnisse als Rechtecke im dreidimensionalen Raum zuständig ist. Unity unterteilt die Ausführung der Anwendung in Zyklen. Jeder Zyklus dauert dabei ein Frame. Damit eine subjektiv ausreichende Qualität gewährleistet werden kann, sollten 15 Frames pro Sekunde erreicht werden, wie M. und K. Claypool und F. Damaa 2006

[1] Verwendete Version: 2017.3.23f1, https://unity3d.com/de/unity/qa/lts-releases (17.09.2019)

[2] Verwendete Version: 2017.4.3.0, https://github.com/microsoft/MixedRealityToolkit-Unity/releases/tag/2017.4.3.0-Refresh (17.09.2019)

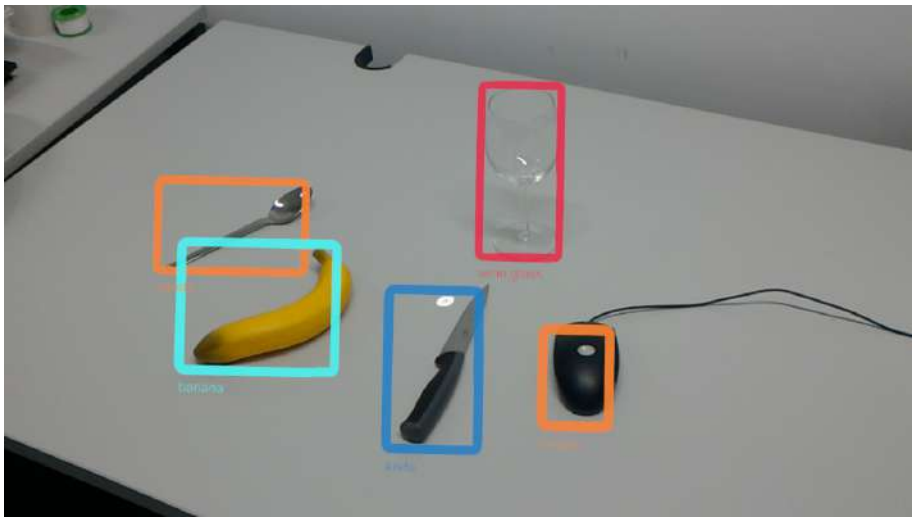


Abbildung 3.3: Applikation auf der HoloLens mit Boundingboxen über einigen erkannten Objekten

feststellten [MC06]. Damit sollte die Berechnung einer Iteration nach ca. $66.67ms$ abgeschlossen sein.

Um dies zu erreichen, ist ein teures Erstellen neuer Boxen (oder Rechtecke) möglichst zu vermeiden. Hierzu kann eine festlegbare Anzahl von Bounding Box Objekten zum Start des Programms erzeugt werden, welche zur weiteren Laufzeit lediglich verschoben werden. Sollte die Anzahl an gleichzeitig in einer Szene erkannten Objekte über den Wert dieses Parameters geraten, lässt sich dieses teure Erstellen jedoch nicht vermeiden. Serverseitiges Tracking von dem gleichen Objekt zugeordneter Rechtecke verhindert darüber hinaus das Oszillieren von Rechtecken zwischen verschiedenen Frames.

Empfangene Ergebnisse des Servers werden, nach ihrer eindeutigen Identifikationsnummer sortiert, in Paketen gespeichert. Die dabei übertragenen Koordinaten des Rechtecks wurden entsprechend dem Kamerabild der HoloLens erstellt und auf das Koordinatensystem der Bildschirme der HoloLens umgerechnet (s. Abbschnitt 3.5). Sie liegen deshalb für einen zweidimensionalen Raum vor und müssen für die korrekte Platzierung in dreimdimensionale Koordinaten umgewandelt werden. Dafür wird ein Strahl ausgehend vom hypothetischen Auge der HoloLens tragenden Person durch die Koordinaten des Bildschirms der HoloLens projiziert. Dies wird für alle empfangenen Koordinaten durchgeführt. Um eine direkte 3D-Koordinate zu erstellen ist die Angabe einer Entfernung auf den Strahlen notwendig. Aus dem Mittelpunkt der gesendeten Koordinaten wird ein weiterer Strahl gebildet. Dieser trifft auf die, die HoloLens umgebende, Entfernungskarte des Spatial Mappings. Die Strecke auf dem Strahl bis zum Schnittpunkt mit der Karte gibt die Entfernung für die dreidimensionalen Koordinaten des Rechtecks an.

Da die Berechnung der Entfernung und die Erzeugung der Strahlen von der Messung der Lage der HoloLens im Raum abhängen, kann es zu Ungenauigkeiten kommen. Deshalb zeigt die HoloLens-Applikation ein empfangenes Rechteck erst dann, wenn es mindestens n mal empfangen worden ist. Vorher wird die Position jeder Koordinate des Rechtecks folgendermaßen gemittelt, wobei k_i die Koordinate angibt, die beim i-ten Empfangen des der Rechtecks-ID berechnet wurde, m die Anzahl ist, wie oft

die ID schon empfangen wurde und e die aktuell empfange Koordinate angibt:

$$\text{Für } m < n\text{: } k_m = k_{m-1} + \frac{e}{m}$$

Es zeigte sich, dass $n = 5$ einen guten Kompromiss zwischen dem Minimieren von Ungenauigkeiten und schneller Erkennung von Objekten bietet.

Damit auch nach dem n-maligen Empfangen der ID sowohl eine rauschunterdrückende Mittlung vorhanden ist, als auch eine größere Änderung der Position des Rechtecks möglich ist, wird eine Art Infinite-Discounted-Horizon-Model verwendet:

$$\text{Für } m > n\text{: } k_m = k_{m-1} \cdot (1 - \alpha) + e \cdot \alpha$$

α ist hierbei der discount-Faktor und gibt an, wie stark vorherige Positionen mit in die aktuelle Position einfließen. Ist $\alpha = 1$, so wird nur das aktuelle Paket beachtet, ist $\alpha = 0$ so findet kein Update statt. Für die Objekterkennung auf Basis von YOLOv3 wurde, nach einigen Experimenten, $k = 0.8$ gewählt.

Empfängt die Anwendung über mehrere Frames eine bestimmte ID nicht mehr, so wird die dazugehörige Bounding Box ausgeblendet. Nach weiteren Frames des Nicht-Empfangens wird sie zurückgesetzt und für andere Objekte freigegeben.

3.2.3 Verwendete Hardware

Um die Echtzeitfähigkeit des Systems sicherzustellen, ergeben sich besondere Anforderungen an die zu verwendende Hardware. So muss sowohl die Berechnung der verwendeten Bildverarbeitungsalgorithmen wie Objekterkennung als auch die Übertragungen des Videofeeds vom AR-Headsets und die Rückübertragung des berechneten Ergebnises zum AR-Headset möglichst kurz gehalten werden.

PC

Für den dedizierten PC, zur Berechnung der Algorithmen der Bildverarbeitung, wurde folgende Hardware verwendet:

Prozessor: Intel i7-9700
Arbeitsspeicher: 64 Gigabyte
Grafikkarte: Nvidia Geforce RTX 2080

HoloLens

Die Augmented Reality Plattform bildet die erste Generation der Microsoft HoloLens. Weitere Informationen über dieses Head-Mounted-Display befinden sich in Kapitel 2.1.2.

Netzwerk

Ebenfalls von gewisser Relevanz für eine verzögerungsarme Übertragung von großen Datenmengen ist die verwendete Netzwerkinfrastuktur. Server und Client befanden sich im gleichen lokalen Netzwerk, der Client (bzw. die HoloLens) war über folgende kabellose Verbindung mit dem kabelgebundenen Server verbunden:
5 GHz Wifi, 802.11n 450Mbits/s, gemessene Geschwindigkeit: ca. 200Mbits/s.

3.3 Übertragungsmodi

Ein zentraler Aspekt der Software und Gegenstand mehrerer Iterationen war der Modus der Übertragung des Videofeeds der Weltkamera der HoloLens auf den dedizierten Computer, um dort rechenaufwendige Operationen durchzuführen.

Auf Grund des Anspruchs der Echtzeit des Gesamtsystems ergeben sich mehrere Anforderungen an mögliche Arten der Videoübertragung:

Verzögerung: Die gesamte Übertragung pro Frame sollte so wenig Zeit wie möglich benötigen. Dazu zählen neben der reinen Kommunikation zwischen beiden Geräten an sich auch entstehende Rechenzeiten durch Kompression auf der Seite der HoloLens sowie Dekompression auf der Empfängerseite.

Zeitstabilität: Diese Verzögerung der Gesamtübertragung sollte auch über lange Zeiträume stabil bleiben. Auf jeden Fall zu vermeiden sind ansteigende Verzögerungen in Abhängigkeit mit der Laufzeit des Systems.

Qualität: Um eine Weiterverarbeitung der Daten mit verschiedenen AdB zu ermöglichen, ist ein Mindestmaß an Bildqualität nötig. Diese setzt sich aus der Auflösung der Bilder und der verwendeten Bitrate zusammen. Faktoren wie Bildrauschen werden nicht von der Übertragungsart beeinflusst und deshalb zunächst nicht beachtet.

Bildfrequenz: Die Bildfrequenz, auch fps (frames per second) genannt gibt an, wie viele Bilder pro Sekunde übertragen werden. Wie Zinner et al. [ZHAH10] zeigen, trägt dieser Wert maßgeblich zu einer schlechter wahrgenommenen Bildqualität bei. Dies ist besonders bei Werten unter 15 fps der Fall, sodass dies als Richtwert für die weitere Bewertung der verschiedenen Übertragungsmodi genutzt werden kann.

Datenmenge/Kompression: Die zu übertragenen Daten sollten möglichst klein gehalten werden, um die Abhängigkeit der Applikation von der Verfügbarkeit von schneller drahtlosen Datenübertragung zu verringern. Dies kann jedoch im Widerspruch zu den zur Bildfrequenz und/oder Qualität stehen.

3.3.1 Unity-API

Als erste betrachtete Möglichkeit der Videoübertragung ist die Benutzung von Unity-APIs zu nennen. Die Idee ist hierbei über von Unity bereitgestellte Schnittstellen auf die Kamera der HoloLens zuzugreifen und so einen Videostream zu erhalten, welcher dann verschickt werden kann [Zel18]. Ein direktes Verschicken des Videostream ist jedoch nicht möglich, da die Schnittstellen lediglich das Speichern auf der HoloLens zulassen. Somit ist diese Methode nicht für Echtzeitanwendungen nutzbar.

Unity-APIs erlauben auch das Aufnehmen einzelner Fotos. Diese können als Rawbytes im Speicher der HoloLens gehalten werden, was eine Weiterverarbeitung ermöglicht. Da eine Anbindung an LSL [Bou18] zur Weiteren einfachen Verwendung und Modularität des Gesamtsystems behilflich ist, erfolgte die Übertragung der Daten mit HoloLSL.

Als problematisch stellte sich jedoch die für HoloLSL zu große Datenmenge pro Bild heraus. Diese war dem Umstand der fehlenden Kompression geschuldet. Da verschiedene Einschränkungen der Softwareumgebung eine einfache Anpassung von HoloLSL

oder eine Pro-Bild-Kompression erschwerten, wurde dieser Übertragungsmodus nicht weiter verfolgt.

3.3.2 Research-Mode

Abbildung 3.4: Blick durch die HoloLens auf eine Beispielapplikation von Microsoft, die einige verfügbare Streams des Research-Modes in Echtzeit anzeigt. [MZ18b]

Der Research-Mode ist eine Funktion der HoloLens, die es erlaubt direkt auf verschiedene Sensoren zuzugreifen. Diese Streams können entweder direkt auf der HoloLens weiterverarbeitet oder gespeichert werden oder über das Netzwerk in Echtzeit an einen anderen Computer gesendet werden. Zu den verfügbaren Streams gehören [MZ18b]:

- Vier Umgebungskameras, die von der HoloLens für Tracken von Kopfbewegungen genutzt werden
- Eine hochfrequente Tiefenkamera (30 fps), die zur Erkennung der Gestensteuerung genutzt werden kann und nur im Nahbereich aufnimmt
- Eine niederfrequente Tiefenkamera (1 fps), welche für das Spatial-Mapping zum Einsatz kommt
- Zwei Versionen eines Infrarot-Kamera-Streams. Dieser misst die Reflexionen einer in der HoloLens verbauten Infrarotdiode.
- Eine Weltkamera, welche nach vorne in Richtung des Blickfelds der nutzenden Person gerichtet ist. Der Stream dieser Kamera wird für die weitere Verarbeitung genutzt werden.

Eine Übersicht über einige Streams ist in Abbildung 3.4 zu sehen.

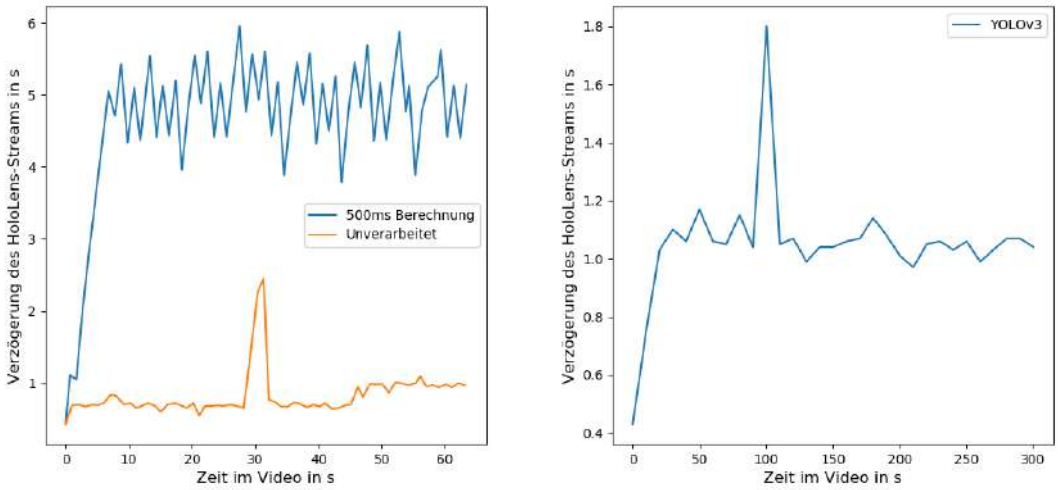
Die Nutzung des Research-Modes unterliegt einigen Einschränkungen. So gibt es derzeit keine direkte Unterstützung innerhalb von Unity, welches als Framework für die Darstellung vorberechneter Ergebnisse von Bildverarbeitungsalgorithmen

unerlässlich ist. Als Folge dessen ist es notwendig eine weitere Applikation auf der HoloLens auszuführen, welche den Research-Mode-Stream initialisiert.

Microsoft stellt auf Github[3] einige Beispielimplementationen für verschiedene Weiterverarbeitungsweisen (Off- / Online) sowie anzusteuernde Sensoren zur Verfügung. Auf Grundlage der Beispielapplikationen „StreamerPV“ und „sensor_receiver.py“ wurde das Streamen der Bilder der Weltkamera an einen sich im gleichen Netzwerk befindlichen Computer realisiert.

Die Applikation „StreamerPV“ wurde so modifiziert, dass diese im Hintergrund auf der HoloLens ausgeführt werden konnte und einen Stream der Daten der Weltkamera startet. Das Python-Skript „sensor_receiver.py“ empfängt diese Daten und teilt sie in einzelne Bilder der Kamera auf, welche dann im Rahmen der AdB weiterverarbeitet werden.

Eine nachträgliche Komprimierung vor dem Verschicken der Rohdaten ließ sich auf Grund der stark gekapselten und integrierten API des Research-Modes nicht umsetzen. Zudem ließ sich die Auflösung der Kamera nicht ändern. Dies führt zu einem hohen Datenaufkommen, da jedes Bild bei einer Auflösung von 1280×720 Pixeln eine Größe von ca. 2.6367 Mebibyte besitzt[4]. Mit der verwendeten Netzwerkhardware konnten auf diese Weise maximal zwischen sechs und sieben Bilder pro Sekunde verschickt werden.



(a) (b)

Abbildung 3.5: Verzögerungen der Bildübertragung im Research-Mode. (a) zeigt einen Vergleich der Verzögerung zwischen eines sehr rechenaufwendigen Algorithmus und keinem zusätzlichen Rechenaufwand über 60 Sekunden. (b) trägt die Verzögerung bei Objekterkennung mit YOLOv3 über fünf Minuten ab.

Eine wesentliche Metrik der Übertragungsmethode ist die Verzögerung der Übertragung, also die Zeitdifferenz zwischen dem Auftreten von Ereignis a und dem

[3] HoloLensForCV [Ols17]

[4] Berechnung mit Auflösung von $1280 \cdot 720$, 3 Farben à Pixel, 8 Bit à Pixel ($\frac{1280 \cdot 720 \cdot 3 \cdot 8}{1024 \cdot 1024 \cdot 8}$)

Wiedergeben bzw. Empfangen der Bilddaten von *a* auf dem Server. Um diese Verzögerung zu messen, ohne Performanceeinbußen in Kauf zu nehmen, wurde eine das System nicht beeinträchtigende, externe Messmethode gewählt. Dieses Verfahren wird in Kapitel 5.1 genauer beschrieben.

Wie in Abbildung 3.5a zu sehen ist, hängt diese Verzögerung stark vom gewählten Algorithmus ab. Wird kein Bildverarbeitungsalgorithmus auf dem Companion-Computer ausgeführt (Fall „Unverarbeitet“ in Abb. 3.5a), so liegt die Latenz der Übertragung und Darstellung um eine Sekunde. In dieser Konfiguration wird auch die maximale Bildfrequenz des Research-Modes als Übertragungsmethode von ca. 7 fps erreicht. Wird zur Berechnung des Bildverarbeitungsalgorithmus mehr Zeit benötigt, in diesem Fall wurden 500ms simuliert (Fall „500ms Berechnung“ in Abb. 3.5a), so ist ein erheblicher Anstieg der Verzögerung festzustellen. Die Verzögerung schwankt nun zwischen vier und knapp sechs Sekunden und ist dabei, im Vergleich zu keiner simulierten Prozesszeit für die Algorithmusberechnung weitaus weniger stabil. Zusätzlich sinkt die Bildfrequenz auf durchschnittlich zwei fps.

Abbildung 3.5b stellt die Verzögerung der Übertragung mit YOLOv3 als berechneten Algorithmus dar. Abgesehen vom Verhalten direkt nach dem Start der Übertragung und einem Ausreißer, bleibt die Verzögerung bei annähernd gleicher Komplexität des Bildverarbeitungsalgorithmus auch über längere Zeiträume weitestgehend stabil. Nichtsdestotrotz schränkt die durchschnittliche Verzögerung von ca. einer Sekunde dieser Übertragungsmethode den Einsatz in einem Echtzeitsystem enorm ein.

3.3.3 Device-Portal

Das Device-Portal der HoloLens ist eine Schnittstelle um das Gerät über das Netzwerk via HTTP oder über eine USB-Verbindung zu konfigurieren und weitere Diagnose-Tools auszuführen. Es liegt sowohl als Webserver als auch als REST-API[5] auf der HoloLens vor [Sat19].

Als weitere Funktion bietet das Device-Portal ein Live-Video aus der Sicht der nutzenden Person. Hierzu wird die Weltkamera der HoloLens genutzt. Abbildung

[5]REST-API Beschreibung: https://docs.microsoft.com/en-us/windows/uwp/debug-test-perf/device-portal-api-core (letzter Zugriff: 10.09.19)

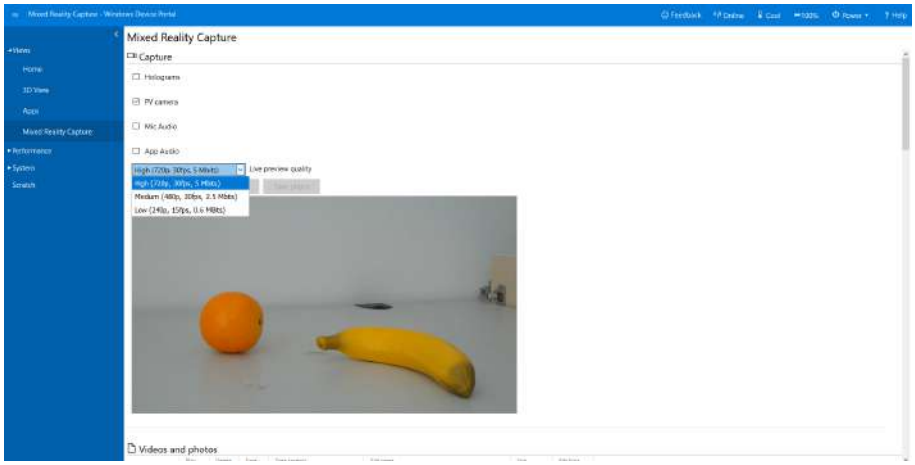


Abbildung 3.6: Device-Portal der HoloLens

3.6 zeigt diese Vorschau in den drei verfügbaren Qualitätseinstellungen, welche in Tabelle 3.1 aufgelistet sind.

Das Live-Video kann im lokalen Netzwerk abgerufen und am Companion-Computer weiterverarbeitet werden. Um eine möglichst verzögerungsarme und verzögerungszeitstabile Übertragung zu erreichen, verwirft dieser alle empfangenen Frames, solange die Berechnung des Bildverarbeitungsalgorithmus für den letzten Frame noch nicht abgeschlossen ist. Durch die so nicht beachteten Frames ergibt sich die Differenz der gesendeten und verarbeiteten Bilder pro Sekunde, wie in Tabelle 3.1 dargestellt. Ebenfalls dort befindet sich die durchschnittliche Verzögerung des Streams für alle drei Qualitätseinstellungen, wenn YOLOv3 als Berechnungsaufwand gewählt wird. Für die Bilderkennung mit YOLOv3 benötigt der verwendete PC ungefähr 70ms.
Da anzunehmen ist, dass die meisten Bildverarbeitungsalgorithmen von einer besseren Bildqualität profitieren und es in der genutzten Konfiguration keinerlei Bandbreitenbeschränkungen gab, wurde im weiteren Verlauf immer mit der höchsten Qualitätseinstellung gearbeitet. Des Weiteren ist die gemessene durchschnittliche Verzögerung der Übertragung hier mit 0.568s am geringsten.

Qualität	Auflösung	gesendete Bildfrequenz	verarbeitete Bildfrequenz	Bandbreite (in MBit/s)	ø - Verzögerung (in s)
hoch	1280×720	30	13.890	5.0	0.568
mittel	640×480	30	13.796	2.5	0.582
niedrig	320×240	15	7.507	0.6	0.723

Tabelle 3.1: Mögliche Qualitätseinstellungen des Video-Streams

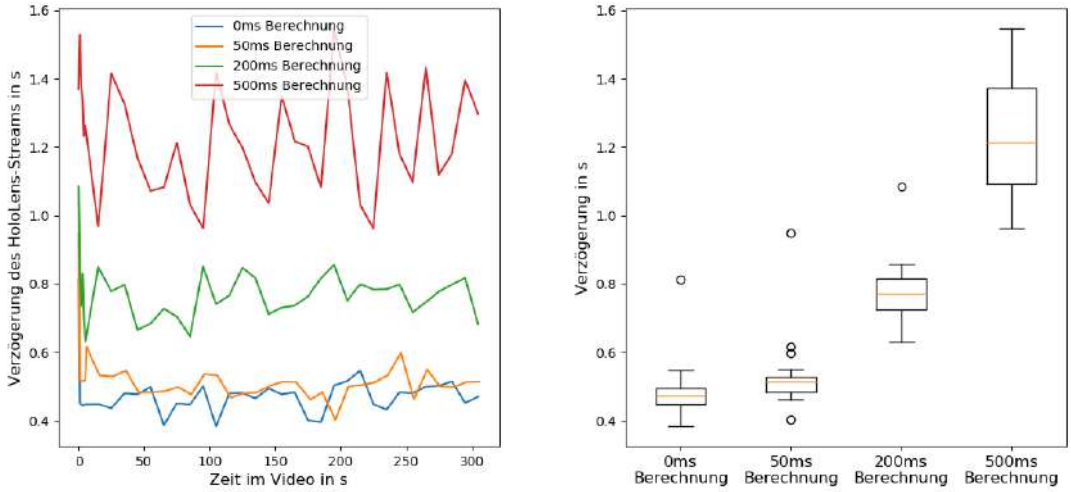
Zur Einschätzung der zu erwartenden Verzögerung für verschiedene AdB wurde die Verzögerung der Übertragung bei verschieden langer Berechnung gemessen. Abbildung 3.7 zeigt die Ergebnisse als Graphen und Box-Whisker-Plot. Es ist zu beobachten, das mit steigender Berechnungszeit die Verzögerung ebenfalls steigt. Führt man eine lineare Einfachregression auf den arithmetischen Mittelwerten der vier verfügbaren Berechnungszeiten durch, so erhält man mit $R^2 = 0.9986$ folgenden Zusammenhang zur Berechnung der zu erwartenden Verzögerung des Streams:

$$f(x) = 0.0015363x + 0.461444$$

Wobei x der Berechnungszeit in Millisekunden und $f(x)$ der erwarteten Verzögerung in Sekunden entspricht.
Darüber hinaus ist auch ein Anstieg der Standardabweichung bei steigender Berechnungszeit zu vermerken. Dies fällt besonders bei einer Berechnungszeit von 500ms ins Auge. Es könnte von einer Oszillation gesprochen werden, die jedoch nicht mit Sicherheit bestimmt werden kann, da die genutzte Messfrequenz von 0.1Hz bei den in der Messreihe erreichten durchschnittlichen 1.874 Frames pro Sekunde nur etwa 5.3% der empfangenen Frames betrachtet. Eine höherfrequente Messung wäre mit der gewählten Methodik zu aufwendig.

Bei einem Vergleich der Übertragungsmethode mit Hilfe des Research-Modes fällt direkt die geringere Latenz des Device-Portals auf. Dies gilt sowohl bei unverarbeiteten Daten als auch bei (simulierten) Verzögerungen auf der Seite des Servers. Auf Grund dessen wurde im weiteren Verlauf der Arbeit ausschließlich die Übertragungsmethode des Device-Portals genutzt.

3.3.4 Rückübertragung der Ergebnisse

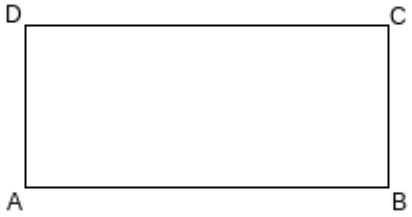


(a) (b)

Abbildung 3.7: Verzögerungen der Bildübertragung via Device-Portal mit unterschiedlichem Berechnungsaufwand. (a) trägt die Verzögerung mit unterschiedlich hohem, simulierten Berechnungsaufwand über fünf Minuten ab. (b) zeigt die gleichen Daten als Boxplot mit 1.5-fachem Interquartilsabstand der Whisker.

Ist die Berechnung des Bildverarbeitungsalgorithmus auf dem dedizierten Computer abgeschlossen und liegt ein Ergebnis vor, so muss es zurück an die HoloLens übertragen werden. Dies geschieht pro verarbeitetem Frame in Form eines n-Tupels.

Abbildung 3.8: Notation Ecken



Für den Algorithmus der Objekterkennung müssen für jedes erkannte Objekt die Koordinaten des minimal umgebenden Rechtecks, die Klasse des erkannten Objekts t sowie eine Identifikationsnummer i übertragen werden. Diese Nummer muss eindeutig auf ein Objekt verweisen, sodass rechenaufwendige Initialisierungen immer neuer Bounding-Box-Objekte auf der HoloLens vermieden werden können. Um dies sicherzustellen ist ein Tracking-Algorithmus erforderlich, dieser wird in Abschnitt 3.4 diskutiert.

Um die Position, Größe und Form eines Rechtecks genau zu bestimmen, reichen zwei Punkte, die auf der gleichen Diagonalen durch das Rechteck liegen. Gesendet werden deshalb lediglich die Koordinaten der unteren linken Ecke A sowie der oberen rechten Ecke C. Diese Koordinaten müssen vorher auf ein Koordinatensystem des Sichtfeldes der nutzenden Person transformiert werden, mehr dazu befindet sich in Kapitel 3.5.

Daraus ergibt sich folgender Tupel zur Übertragung für jedes erkannte Objekt:

$$(A_x, A_y, C_x, C_y, i, t)$$

Die Übertragung an die HoloLens erfolgt mit Hilfe von HoloLSL (s. Abschnitt 2.3.1).

3.4 Tracking

Durch die in Sektion 3.1 festgelegten Einschränkungen der verwendbaren Algorithmen der Bilderkennung für dieses System, kann gefolgert werden, dass bei einer statischen Umgebung sowie geringer Eigenbewegung der nutzenden Person von Frame zu Frame nur kleine Änderungen der Ergebnisse, vor allem auch deren Position im Bild, zu erwarten sind. Demnach ist es für eine effizientere Darstellung auf der HoloLens sinnvoll, jedes erkannte Objekt zu verfolgen und zu identifizieren, sodass es der HoloLens möglich ist, lediglich die Änderungen zum letzten Frame darzustellen, da jedes Objekt eine eindeutige und einmalige Identifikationsnummer erhält. Dadurch wird ein dauerhaftes, rechenaufwendiges Initialisieren neuer Rechtecksobjekte bzw. das Oszillieren dieser auf der HoloLens vermieden.

Dieses Tracking erfolgt serverseitig und ist vom verwendeten AdB unabhängig, da es nach jedem verarbeiteten Frame ausgeführt wird und dazu nur die hervorgehobenen Regionen und deren Benennung benötigt.
Auf Grund dessen kann der Trackingalgorithmus Objekte, welche die Szene zum Zeitpunkt t_0 verlassen haben, nicht wieder als gleiches Objekt erkennen, sollte ein Objekt der gleichen Klasse die Szene an ähnlicher Stelle zum Zeitpunkt $t_0 + \Delta t$ wieder betreten, da kein Wissen über die Lage und Bewegung der HoloLens im Raum vorliegt und sich die nutzende Person zum Zeitpunkt $t_0 + \Delta t$ in einer komplett anderen Position im Raum befinden könnte.

3.4.1 Der Algorithmus

Der Trackingalgorithmus basiert auf einem Centroid-Tracker wie ihn A. Rosebrock [Ros18] beschrieben hat und basiert auf der kleinsten euklidischen Distanz von zwei Rechtecken zwischen zwei Frames:

- Es wird eine Liste mit bereits getrackten Objekten gehalten.
- Von jedem neu erkannten Rechteck wird der Mittelpunkt gebildet (Centroid).
- Dieser Mittelpunkt wird mit jedem Mittelpunkt bereits erkannter Rechtecke verglichen.
- Aus dieser Liste von euklidischen Abständen wird der geringste gewählt. Es wird angenommen, dass sich das Objekt zu den nun neuen Koordinaten bewegt hat und die dazugehörigen Koordinaten werden aktualisiert.
- Übrig gebliebene Rechtecke werden als neue Objekte zur Liste der getrackten Objekte hinzugefügt.
- Der Zuletzt-gesehen-Zähler bereits getrackter Rechtecke, denen kein neues Rechteck zugeordnet werden konnte, wird inkrementiert.
- Rechtecke, die lange nicht gesehen wurden, bzw. deren Zähler einen Schwellwert überschreitet, werden gelöscht.

Dieser Algorithmus wird für jede Klasse von Ergebnissen separat durchgeführt, solange angenommen werden kann, dass die markierte Region ihre Klasse nicht wechselt. Da dies bei der Objekterkennung der Fall ist, wird ein Centroid-Tracker für jedes mögliche Objekt vorgehalten.

3.5 FOV-Berechnungen

Um berechnete Ergebnisse des Servers an der richtigen Stelle im AR-Headset anzeigen zu können, bedarf es einer Abbildung vom Sichtfeld der Kamera der HoloLens auf die Bildschirme der HoloLens. Obgleich einer gleichen Auflösung von 1280×720 Pixeln beider optischen Geräte kann dies keine Abbildung der Form $f : (x, y) \mapsto (x, y)$ sein, da unterschiedliche Sichtfelder vorliegen.

Zur Quantifizierung der verschiedenen horizontalen Sichtfelder wurde eine Messung durchgeführt. Für diese wurde die HoloLens jeweils so vor einer Wand platziert, dass das horizontale Sichtfeld der Weltkamera bzw. des Bildschirms auf dieser 150cm betrug. Der Bildschirm zeigte zu diesem Zweck eine einzelne Farbe an, um dessen Enden besser sehen zu können. Gemessen wurde daraufhin der Abstand zur Wand, wie in Abbildung 3.9a und 3.9b zu sehen ist. Aus den Abständen, 193cm für die Weltkamera und 275cm für die Bildschirme, kann das halbe Sichtfeld mittels des Sinussatzes berechnet werden.:

$$\begin{aligned} \frac{a}{sin(\alpha)} &= \frac{b}{sin(\beta)} \\ \alpha &= sin^{-1}(sin(\beta) \cdot \frac{a}{b}) \\ \alpha &= sin^{-1}(1 \cdot \frac{a}{\sqrt{75^2 + x^2}})) \end{aligned}$$

Für $x = 193$ ergibt sich dabei ein komplettes horizontales Sichtfeld von $\alpha = 42.472°$ und für $x = 275$ ein Sichtfeld von $\alpha = 30.51°$.
Normalisiert man den Abstand und legt die Sichtfelder beider optischen Geräte übereinander, wie in 3.9c geschehen, so lässt sich eine Abbildung für die horizontalen Pixel von der Webcam auf die Displays der HoloLens erstellen, wobei w der horizontalen Auflösung also 1280 Pixel entspricht:

$$\begin{aligned} f : x &\mapsto (x - \frac{w}{2}) \cdot \frac{\overline{AD}}{\overline{BC}} + \frac{w}{2} \\ f : x &\mapsto (x - \frac{w}{2}) \cdot \frac{0.389}{0.273} + \frac{w}{2} \end{aligned}$$

Die Subtraktion von $\frac{w}{2}$ sorgt für eine Verschiebung um die Mitte der Bildschirme, da hier keine Verzerrung notwendig ist und demnach $f(\frac{w}{2}) = \frac{w}{2}$ gelten muss.
Für die vertikalen Pixelkoordinaten kann analog vorgegangen werden.

In der praktischen Anwendung kamen noch einige manuelle Anpassungen des Koeffizienten $\frac{0.389}{0.273}$ hinzu, um Messungenauigkeiten auszugleichen sowie dem Umstand zu begegnen, dass sich die Weltkamera der HoloLens ein paar Zentimeter über den Bildschirmen befindet.

Neben dieser Abbildung zur Umrechnung der Koordinaten wurde auch versucht, die Weltkamera der HoloLens in Unity zu modellieren, um so die verschiedenen Sichtfelder zu berücksichtigen. Auf Grund von Einschränkungen der verwendeten Bibliothek HoloToolKit ließ sich dies jedoch nicht verwirklichen.

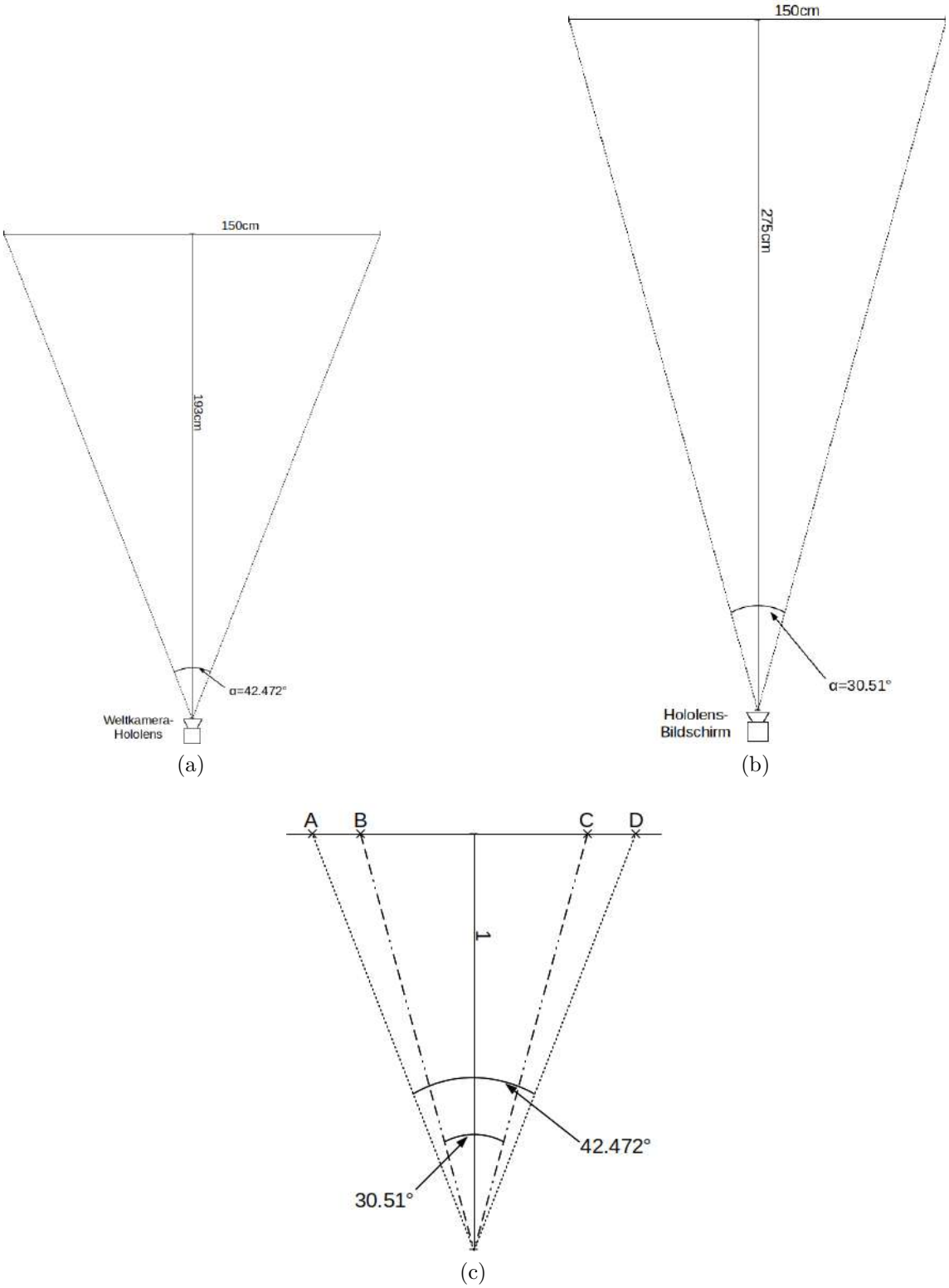


Abbildung 3.9: Vergleich des horizontalen Sichtfeldes der Weltkamera und der Bildschirme der HoloLens. (a) zeigt das horizontale Sichtfeld der Weltkamera. Im Abstand von 193cm beträgt es 150cm. (b) zeigt das horizontale Sichtfeld der Bildschirme der HoloLens bei 275cm Abstand. (c) vergleicht beide Sichtfelder auf normalisierter Länge.

4. Evaluation anhand einer Nutzungsstudie

Mit Hilfe einer qualitativen Evaluation in Form einer Nutzendenstudie wird ein Vergleich zu konventionelleren Darstellungsmöglichkeiten gezogen.
Die durchgeführte Nutzendenstudie hat das Ziel Unterschiede zwischen dem entwickelten System, also einer explorativen Darstellung eines Algorithmuses der Bildverarbeitung, sowie der Repräsentation dieses Algorithmuses als Video zu finden. In diesem Zusammenhang wird die eingangs gestellte Hypothese untersucht, ob es Versuchspersonen leichter fällt die Möglichkeiten und Grenzen eines Algorithmuses zu erkennen und vorherzusagen, sollte dieser ihnen als Echtzeitvisualisierung in einer Augmented Reality-Umgebung präsentiert werden. Der in der Studie untersuchte Algorithmus der Bildverarbeitung war Objekterkennung auf Grundlage von YOLOv3.

4.1 Metadaten des Versuchs

Durchgeführt wurden die Experimente an der Universität Bremen im Seminarraum der Arbeitsgruppe *Cognitive-Systems-Labs (CSL)*. Es gab insgesamt $N = 14$ Versuchspersonen, von denen zehn ebenfalls an der Universität Bremen studieren. Das Durchschnittsalter betrug 27.57 Jahre ($\sigma = 12.95$). Auf die Frage ihres Geschlechts antworteten vier der Teilnehmer*Innen mit weiblich, zehn mit männlich.
Die Experimente fanden in einem gesonderten Raum statt, um Einflüsse der Umgebung zu minimieren. Während ihrer Durchführung stand die Versuchsleitung jederzeit für Rückfragen der Teilnehmenden zur Verfügung.

4.2 Versuchsaufbau

Der Versuchsaufbau war für alle teilnehmenden Personen gleich und lässt sich in folgende drei Komponenten aufteilen:

Parcours und Hololens Um den Versuchsteilnehmenden explorative Möglichkeiten für die Objekterkennung zu geben, wurden verschiedene Objekte, von

Abbildung 4.1: Versuchsperson mit Hololens in Parcours mit Konfiguration A

denen die meisten von ebenjener Objekterkennung erkannt werden können, auf und um Tische verteilt. Für den Versuch wurden zwei leicht unterschiedliche Konfigurationen dieses Parcours (A und B) bereitgestellt. Jede versuchsteilnehmende Person durchschritt beide Anordnungen. Konfiguration A mit einer Versuchsperson während eines Experimentes ist in Abbildung 4.1 zu sehen.

Videoaufnahmen Als vergleichende Repräsentation der Objekterkennung wurde ein Video genutzt. Dieses zeigt das Durchschreiten des Parcours in jeweils einer der beiden Konfigurationen. Als Kamera wurde die selbe Kamera der HoloLens genutzt, welche auch für die Echtzeitvisualisierung zum Einsatz kam. Dem Video wurde nachträglich die Objekterkennung hinzugefügt, das entwickelte System der Echtzeitübertragung und Übermittlung der Ergebnisse kam dabei nicht zum Einsatz.

PC mit Fragebogen In unmittelbarer Nähe zum Parcours befand sich ein PC auf welchem Fragen nach jedem Durschschreiten des Parcours sowie Anschauen der Videos beantwortet werden konnten. Der genaue Fragebogen mit allen Antworten befindet sich im Anhang dieser Arbeit (s. A.2.2).

4.3 Versuchsablauf

Der Versuchsablauf war ebenfalls für alle Teilnehmenden gleich. Nach der Einführung und Klärung eventueller Rückfragen lässt er sich in fünf Phasen unterteilen:

1. Durchqueren des Parcours in Konfiguration A mit der HoloLens

   Beantwortung der Fragenmenge F für das Durchqueren des Parcours

2. Anschauen eines Videos des Parcours in Konfiguration A

   Beantwortung der Fragenmenge F für das Anschauen des Videos

3. Durchqueren des Parcours in Konfiguration B mit der HoloLens

   Beantwortung der Fragenmenge F für das Durchqueren des Parcours

4. Anschauen eines Videos des Parcours in Konfiguration B

   Beantwortung der Fragenmenge F für das Anschauen des Videos

5. Beantwortung allgemeiner Fragen zur Anwendung u. a. bezüglich der Usability

Zum Durchqueren des Parcours wurden den teilnehmenden Personen in etwa die gleiche Zeit wie für das Anschauen der Videos gewährt.

Die Fragenmenge F bestand aus acht Fragen. Die ersten beiden Fragen erlaubten Multiple-Choice Antworten aus einer Liste von jeweils 32 Objekten. Sechs weitere Fragen ließen Antworten auf einer Likert-Skala von 1 bis 7 zu. Folgende Fragen wurden gestellt:

1. Welche Objekte hast du während des Experimentes erkannt?
2. Welche Objekte wurden von der Objekterkennung erkannt?
3. Als wie gut bewertest du die gerade von dir erlebte Objekterkennung?
4. Wie gründlich konntest du die Objekterkennung testen?
5. Wie sicher bist du dir in der Einschätzung der Erkennungsqualität?
6. Fandest du, dass du die Kontrolle über die Objekterkennung hattest?
7. Wie genau war die räumliche Abbildung der Rechtecke um erkannte Objekte?
8. Wie sehr würde sich deiner Meinung nach deine Einschätzung ändern, solltest du das Experiment wiederholen?

4.4 Versuchsauswertung

4.4.1 Fragen 3. bis 8.

Um einen Überblick über die Verteilung der Antworten der erhobenen Fragen zu bekommen, wurden zunächst Durchschnittswerte und Standardabweichung für die Fragen 3. bis 8. der Fragenmenge F gebildet. Das Ergebnis ist in Abbildung 4.2 zu sehen.
Bei erster Betrachtung anhand der Mittelwerte fällt auf, dass die Versuchspersonen beim Gebrauch der HoloLens das Gefühl hatten die Objekterkennung gründlicher testen zu können und mehr Kontrolle über sie zu haben (Frage vier und sechs). Bei der allgemeinen Qualität sowie der räumlichen Präzision der Bilderkennung bewerteten die Versuchspersonen das Video besser (Frage drei und sieben). In der Sicherheit ihrer Einschätzung liegen mit den Fragen drei und sechs gegensätzliche Tendenzen der teilnehmenden Personen vor.

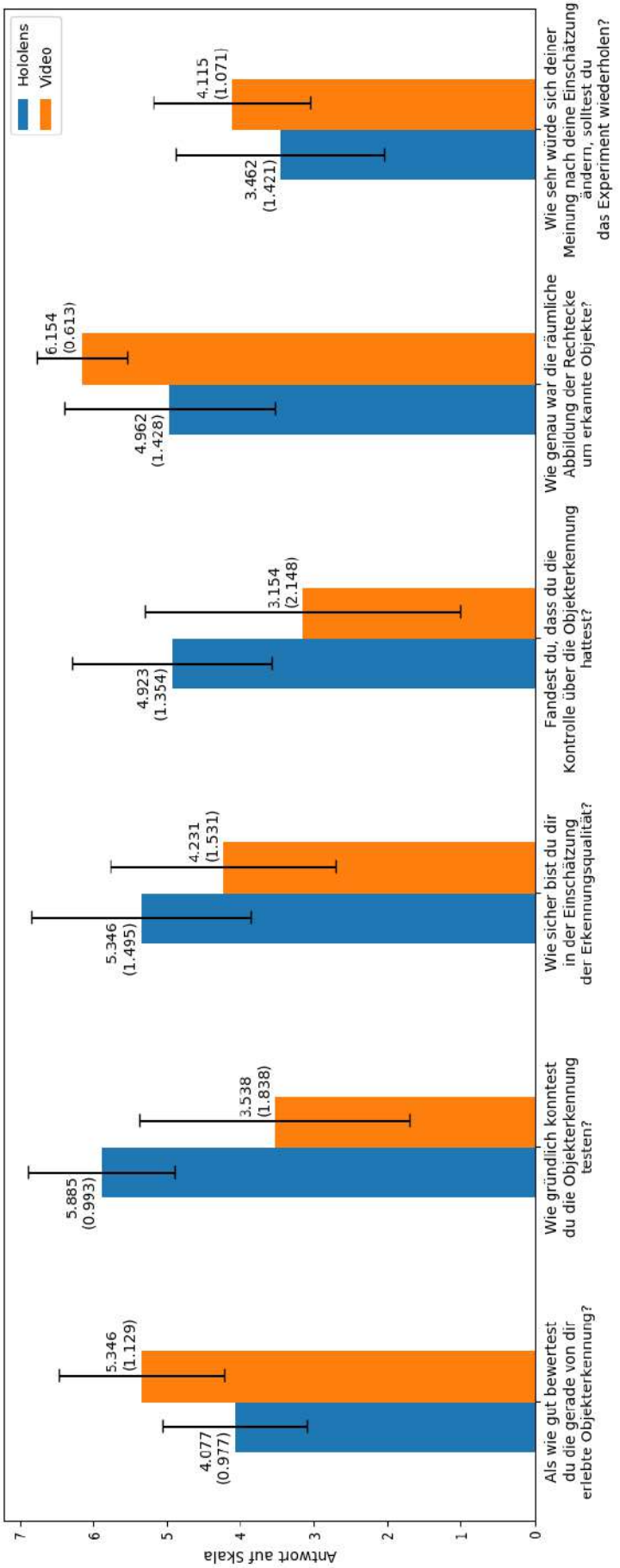


Abbildung 4.2: Durchschnitt und Standardabweichung der dritten bis achten gestellten Frage mit numerischer Antwortskala.

Zur genaueren Untersuchung dieser Unterschiede bieten sich statistische Tests an, mit welchen die Wahrscheinlichkeit bestimmt werden kann, dass die, als Samples zu betrachtenden, Antworten einer Versuchsperson aus unterschiedlichen Grundgesamtheiten stammen. Ist die Gegenwahrscheinlichkeit dieser Wahrscheinlichkeit kleiner als ein anfangs gewähltes Signifikanzniveau, so lässt sich sagen, dass sich die Grundgesamtheiten beider Fragen signifikant unterscheiden [Fro17].
Verglichen wurden jeweils die gleichen Fragen, einmal im Kontext des Anschauen des Parcours mit der HoloLens gestellt, einmal nach dem Anschauen des Videos.

Es handelt sich bei diesen Fragen um Antworten einer Likert-Skala. Dies erschwert die statistische Verwertung, da auch wenn möglichst neutrale Beschriftungen für diese gewählt wurden, es Diskussionen um die Äquidistanz der möglichen Anwortpunkte sowie der Normalverteilung gibt [Rob12].

Unter Berücksichtigung dieses Umstands untersuchte P. Cairns 2019 [Cai19] sechs parametrische und nicht-parametrische, statistische Tests für ihre Verwendung zur Analyse von Likert-Skalen. Alle Tests wurden auf Fehler der ersten Art sowie auf Trennschärfe hin untersucht. Unter den verglichenen Tests stellte sich der Brunner-Munzel-Test (BM) [BM00] als am Aussagekräftigsten heraus.
Hierbei handelt es sich um einen nicht parametrischen Test, welcher, im Unterschied zu anderen gängigen Tests wie etwa der t-Test, die Daten auf stochastische Dominanz hin untersucht.

Wendet man den BM auf die Fragen 3. bis 8. der Fragenmenge F an, so erhält man bei einem Signifikanzniveau von $\alpha = .05$ statistische Signifikanz bei allen Fragen außer der achten Frage. Die Ergebnisse dieser Tests sind in Tabelle 4.1 zu sehen. Es kann also mit einer Sicherheit von (95%) davon ausgegangen werden, dass sich die Grundgesamtheiten der Fragenpaare der Fragen 3. bis 7. unterscheiden und sich Tendenzen anhand der Mittelwerte ablesen lassen.

Fragenummer	Freiheitsgrade	Brunner-Munzel W-Statistik	p-value
3.	50	4.80	<.001
4.	50	-6.91	<.001
5.	50	-3.07	.00356
6.	50	-3.34	.00202
7.	50	4.27	<.001
8.	50	1.74	.08794

Tabelle 4.1: Ergebnisse des Brunner-Munzel-Test für die Fragen drei bis acht

Zur weiteren Vereinfachung wurden Fragen mit ähnlicher Intention zusammengefasst. Dies geschah folgendermaßen:

Frage 3 und 7: Beide Fragen beziehen sich auf die wahrgenommene Qualität der Objekterkennung und wurden unter dem Aspekt der subjektiven Qualität zusammengefasst.

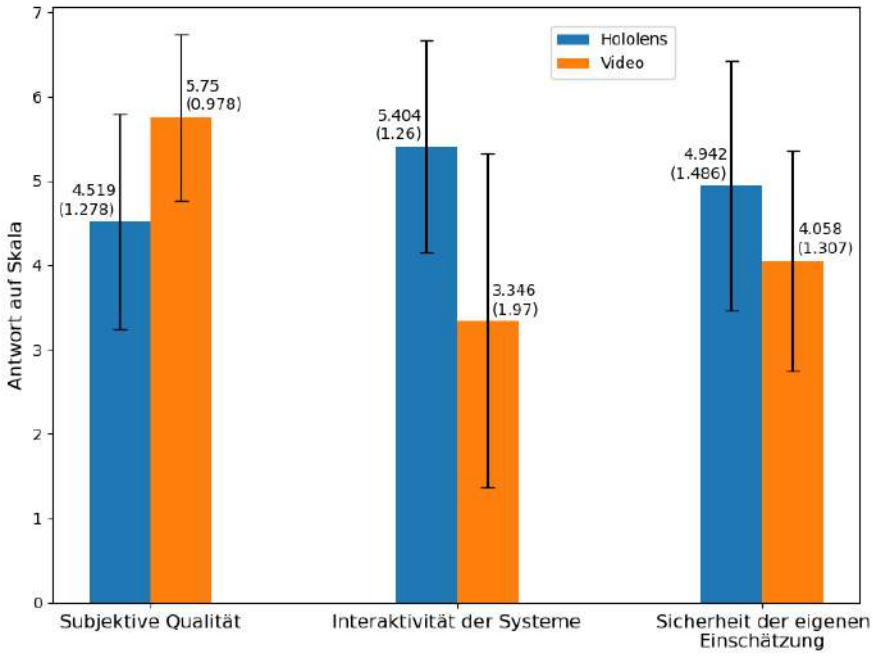


Abbildung 4.3: Arithmetisches Mittel und Standardabweichung der nach Intention kombinierte Fragen der Fragemenge F.

Frage 4 und 6: Die Intention dieser Fragen ist die Handhabung des jeweiligen Systems. Beide betrachten außerdem die Interaktivität der Systeme.

Frage 5 und 8: Die Fragen zielen auf die Sicherheit der eigenen Einschätzung der Versuchspersonen ab. Sollten hier eine erhöhte Sicherheit für das System mit der HoloLens festgestellt werden, so spricht dies für die zu untersuchende Hypothese, dass es Versuchspersonen leichter fällt die Möglichkeiten und Grenzen des dargestellten Algorithmuses zu erkennen.

Zur Untersuchung der Signifikanz dieser Fragenpaare wurden die Antworten beider Fragen kombiniert und ebenfalls ein Brunner-Munzel-Test (BM) mit $\alpha = .05$ als Signifikanzniveau durchgeführt. Die Arithmetischen Mittel sowie die Standardabweichungen dieser kombinierten Fragen sind in Abbildung 4.3 dargestellt.
Ein BM für die Fragen 3. und 7. konnte ein signifikanten Unterschied ermitteln, $BM(102) = 6.00,\ p < .001$. Es lässt sich somit also sagen, dass die Qualität der Objekterkennung mittels Video intersubjektiv besser ist.
Auch für die Fragen 4. und 6. liegt ein kombinierter signifikanter Unterschied vor, $BM(102) = -6.68,\ p < .001$. Die Versuchspersonen schätzten die Interaktivität der Objekterkennung während der HoloLens dementsprechend höher ein als während der Präsentation über ein Video.
Auch für die Sicherheit der eigenen Einschätzung liegt nach der Kombination der Fragen fünf sowie acht ein signifikanter Unterschied vor, $BM(102) = -3.23,\ p \approx .00167$. Da hier die Differenz der Antworten zugunsten der HoloLens ausfällt, die Versuchspersonen also eine größere Sicherheit in ihren Antworten sahen, stützt dies die These, dass das entwickelte System der Einschätzung der Möglichkeiten und Grenzen des dargestellten Algorithmuses zuträglich ist.

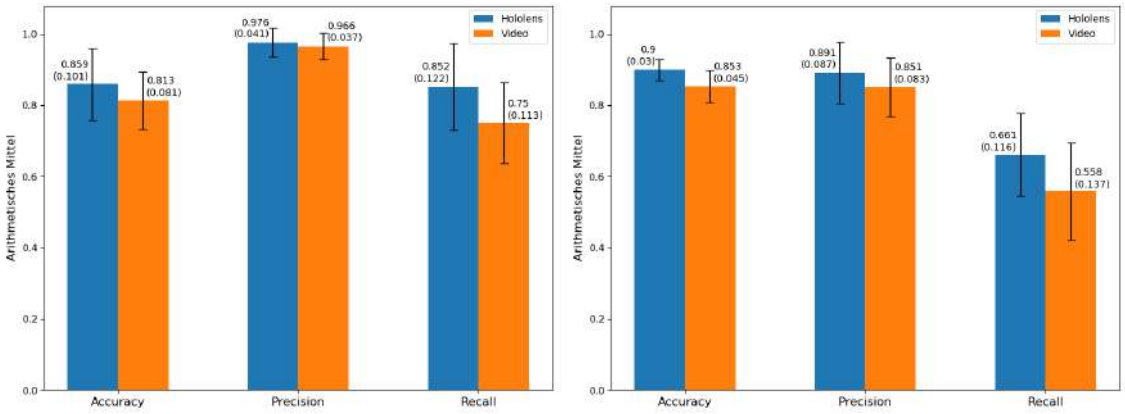
4.4.2 Fragen 1. und 2.

Anders als die im vorherigen Abschnitt beschriebenen Fragen handelt es sich beim Antwortraum der Fragen 1. und 2. (1.: „Welche Objekte hast du während des Experimentes erkannt?“ und 2.: „Welche Objekte wurden von der Objekterkennung erkannt?“) nicht um Likert-Skalen, sondern um Multiple-Choice Antworten. Zur Auswahl standen jeweils 32 Objekte, von denen 27 im Parcours zu finden waren und 20 von der Objekterkennung erkannt werden konnten. Für die erste Frage wurden die gegebenen Antworten mit einer manuell erstellten Grundwahrheit verglichen. Die Auswertung der zweiten Frage verglich die Antworten mit, während des Experiments erstellen, Logs für die Frage zur HoloLens bzw. mit, aus den Videos erstellten Logs, für die entsprechende Frage zu den Videos.

Es wurden folgende Metriken erhoben: Accuracy, Precision und Recall.
Genauere Ausführungen zur Bedeutung, Nutzung sowie einigen Einschränkungen, welche bei der verwendeten Anwendung entstehen, sind in Kapitel 5.2 zu finden. Die Ergebnisse sind in Abbildung 4.4 zu sehen.

Zur Überprüfung der Hypothese, dass sich die Grundgesamtheiten dieser Metriken zwischen den Fragen für die HoloLens und für das Video unterscheiden wurde ein gepaarter t-Test durchgeführt. Die Voraussetzungen für diesen wurden folgendermaßen erfüllt [Hem19]:

- **Abhängigkeit der Messungen.** Es besteht eine Abhängigkeit zwischen beiden Fragen, da dieselbe Person befragt wurde.
- **Die abhängige Variable ist mindestens intervallskaliert.** Die abhängige Variable besteht in den Fragen 1. und 2. aus den berechneten Metriken Accuracy, Precision und Recall. Diese werden als Zahlen ausgedrückt und sind somit intervallskaliert.
- **Die unabhängige Variable ist nominalskaliert und hat zwei Ausprägungen.** Bei der unabhängigen Variablen handelt es sich hierbei um die Art der Präsentation der Objekterkennung. Es steht das System mit der HoloLens und ein Video zur Auswahl. Somit gibt es zwei Ausprägungen, welche keiner geordneten Reihenfolge zugeordnet werden können.
- **Ausreißer.** Als Ausreißer wurden solche Samples definiert, die um mehr als die dreifache Standardabweichung vom Mittelwert abwichen. Es traten keine Ausreißer dieser Art auf.
- **Normalverteilung.** Der gepaarte t-Test erwartet eine Normalverteilung der Differenzen der Daten beider Gruppen. Um dies zu überprüfen, wurde ein Shapiro-Wilk-Test [SW65] durchgeführt. Dieser konnte bei einem Signifikanzniveau von $\alpha = .05$ eine Normalverteilung der Differenzen bei allen Metriken der Fragen außer der Precision der ersten Frage feststellen. Da es jedoch genügend Belege für die Robustheit des t-Tests auch bei der Verletzung der Normalverteilung [GPS72] [MRH92] [LKK96] gibt, wurde von einer weiteren Anpassung der Daten abgesehen.



(a) (b)

Abbildung 4.4: Accuracy, Precision und Recall der ersten Fragen gegen die Grundwahrheit und der zweiten Fragen gegen entsprechende Logs.

Frage 1

„Welche Objekte hast du während des Experimentes erkannt?“

Das Arithmetische Mittel sowie die Standardabweichung für die erhobenen Metriken der ersten Frage in beiden Testumgebungen (HoloLens-System sowie Video) ist in Abbildung 4.4a zu sehen.

	gepaarter t-Test			Shaprio-Wilk-Test	
	Freiheitsgrade	t-Statistik	p-value	t-Statistik	p-value
Accuarcy	54	2.08	.04700	.98	.82030
Precision	54	1.26	.21805	.91	.02201
Recall	54	3.83	<.001	.98	.87472

Tabelle 4.2: Ergebnisse des Shapiro-Wilk-Tests sowie eines gepaarten t-Tests für die Frage eins auf den erhobenen Metriken

Ein gepaarter t-Test zeigt bei einem Signifikanzniveau von $\alpha = .05$ signifikante Unterschiede der Grundgesamtheit bei der Accuracy und dem Recall zu Gunsten des HoloLens-Systems. Diese Ergebnisse sind in Tabelle 4.2 zu sehen.
Aus ersterem lässt sich schließen, dass bei der Verwendung des Systems mit der HoloLens allgemein mehr richtige Antworten gegeben wurden, bzw. es den Versuchspersonen leichter fiel, sich zu merken welche Objekte sich im Parcours befanden. Dennoch sind diese Unterschiede eher klein, die arithmetischen Mittel beider Datensätze unterscheiden sich lediglich um 4.6 Prozentpunkte bzw. Die Accuracy während der Videos entspricht 94.64% der Accuracy des HoloLens-Systems.
Der signifikante Unterschied des Recalls zeigt, dass die Versuchspersonen während der Betrachtung des Parcours mit der HoloLens weniger False Negative Antworten produzierten, also weniger Objekte übersehen wurden. Dies könnte auf die explorativen Eigenschaften des Systems zurückgeführt werden, welche es den versuchsteilnehmenden Personen ermöglichen könnte einen besseren Überblick über die Szene zu

bekommen. Hier beläuft sich der Unterschied der Mittelwerte auf 10.2 Prozentpunkte und Der Recall des Video-Datensatzes entspricht lediglich 88.0% des des HoloLens-Systems.
Die Precision ist in beiden Fällen auf einem sehr hohen Niveau mit jeweils über 96%. Da hier kein signifikanter Unterschied der Grundgesamtheit festgestellt werden konnte, kann davon ausgegangen werden, dass die Art der Betrachtung keinen Einfluss darauf hat, wie viele nicht vorhandene Objekte von Versuchspersonen als vorhanden erkannt werden.

Frage 2

„Welche Objekte wurden von der Objekterkennung erkannt?“

	gepaarter t-Test			Shaprio-Wilk-Test	
	Freiheitsgrade	t-Statistik	p-value	t-Statistik	p-value
Accuarcy	42	3.95	<.001	.98	.86976
Precision	42	2.17	.04194	.97	.72323
Recall	42	3.35	.02874	.98	.87546

Tabelle 4.3: Ergebnisse des Shapiro-Wilk-Tests sowie eines gepaarten t-Tests für die Frage zwei auf den erhobenen Metriken

Ein ebenso auf den zweiten Fragen durchgeführter gepaarter t-Test ($\alpha = .05$) zeigt signifikante Unterschiede der Grundgesamtheit bei allen drei Metriken. Dessen Ergebnisse sind in Tabelle 4.3 dargestellt.
Da die Performance des HoloLens-System hier in allen drei Metriken besser abschneidet, lässt sich sagen, dass es den Versuchspersonen einfacher fiel den Objekterkennungsalgorithmus einzuschätzen, wenn dieser über das HoloLens-System präsentiert wurde.
Die Unterschiede des Mittelwerts der Accuracy sowie der Precision sind mit 4.7 Prozentpunkten (94.8% Video ggü. HoloLens) bzw. 4 Prozentpunkten (95.5% Video ggü. HoloLens) jedoch eher gering. Beide Metriken bewegen sich bei knapp 90% für das HoloLens-System sowie um 85% für das Video.
Ein etwas anderes Bild zeichnet der Recall. Zunächst fällt auf, dass dieser mit 66.1% für das HoloLens-System sowie 55.8% für die Videos insgesamt hinter den anderen beiden Metriken zurück fällt. Es lässt sich also darauf schließen, dass die Versuchspersonen insgesamt einige Objekte übersehen haben, obwohl diese von der Objekterkennung erkannt wurden. Dieser Effekt ist jedoch bei dem HoloLens-System weniger stark ausgeprägt, wie der statistisch signifikante Unterschied der Grundgesamtheiten belegt. Anders formuliert fällt es den Versuchspersonen also leichter die Markierung von von der Objekterkennung erkannten Objekten wahrzunehmen und sich diese zu merken.
Es ist jedoch nicht komplett auszuschließen, dass dieser Umstand den in Kapitel 5.2 beschriebenen Unzulänglichkeiten bzw. Einschränkungen der Erhebung der Metriken geschuldet ist.

4.5 Zusammenfassung der Ergebnisse

Zusammenfassend lässt sich zu den Ergebnissen sagen, dass einige statistisch signifikante Unterschiede zwischen beiden Darstellungsformen existieren.

So bietet die Darstellung als konventionelles Video eine subjektiv bessere und genauere Objekterkennung. Versuchsteilnehmende Personen gaben den für diesen Aspekt relevanten Fragen für das System der HoloLens im Mittel 4.519 von 7 möglichen Zustimmungspunkten, die Darstellung als Video erhielt im Mittel 5.75 Punkte.
Das System der HoloLens erzielte hingegen höhere Zustimmungspunkte bei der subjektiven Kontrolle über das System. Relevante Fragen erhielten hier 5.404 Punkte für die HoloLens bzw. 3.346 Punkte für das Video als Darstellungsform.
Der Sicherheit ihrer eigenen Aussagen gaben die Versuchspersonen dem System der HoloLens ebenfalls höhere Zustimmungspunkte. Die relevanten Fragen erhielten für das System mit der HoloLens durchschnittlich 4.942 Punkte, 4.058 Punkte entfielen auf Präsentation als Video.
Demnach eignet sich ein Video besser für eine präzisere Darstellung der Objekterkennung. Das entwickelte System überzeugt durch eine intersubjektiv bessere Interaktivität.
Des Weiteren konnte festgestellt werden, dass es Versuchspersonen leichter fiel sich an Objekte im Parcours zu erinnern, sollten sie diesen durch die HoloLens wahrgenommen haben. Ähnliches gilt für die Markierungen für von der Objekterkennung erkannte Objekte. Auch hier fiel es Versuchspersonen leichter, sich diese, bzw. den Namen der erkannten Objekte, zu merken, wenn die Repräsentation über das HoloLens-System geschah. Dies kann ebenfalls für die aufgestellte Hypothese, die Möglichkeiten und Grenzen des präsentierten Algorithmuses besser Einschätzen zu können, sprechen, da für eine solche Einschätzung ein besseres Aufnahmevermögen der Ergebnisse des Algorithmuses sicherlich zuträglich ist.

5. Technische Auswertung

Um eine Einschätzung der Performance des System unabhängig vom subjektiven Empfinden der Testpersonen zu erhalten, wurden eine Reihe von synthetischen Tests durchgeführt. Diese dienen der Quantifizierung verschiedener Latenzen des Gesamtsystems, geben Aufschluss über die allgemeine Performance und bewerten die Objekterkennung im speziellen unter verschiedenen Bedingungen.

5.1 Messmethode

Um Latenzen zu messen, ohne Performanceeinbußen, welche die Messung beeinträchtigen würden, in Kauf zu nehmen, wurde eine vom System unabhängige, externe Messmethode gewählt.
Diese bestand aus einem Video des Bildschirmes des Servers, welches eine Millisekunden anzeigende Uhr und das übertragene Video der HoloLens zeigte. Die Kamera der HoloLens war dabei auf ebenjenen Bildschirm gerichtet, sodass im gespeicherten Video sowohl die aktuelle Uhrzeit als auch die von der HoloLens aufgenommene und zurück an den Server geschickte Uhrzeit zu sehen war. Ein Schnappschuss eines solchen Videos mit einem simulierten Algorithmus, welcher 500ms Berechnungszeit benötigt, zeigt Abbildung 5.1. Aus der Differenz beider Zeiten lässt sich die Verzögerung der Übertragung berechnen. Um den Aufwand der Messung auf ein realisierbares Niveau zu senken wurde lediglich alle zehn Sekunden ein Messpunkt gesetzt.

Diese Methodik beinhaltet einige Nachteile beziehungsweise Ungenauigkeiten:

- Die Genauigkeit der gemessenen Latenz ist von der Bildwiederholungsrate des genutzten Bildschirms sowie des aufgenommenen Videos abhängig. Beide betrugen 60Hz, sodass durch diese Einschränkung eine Ungenauigkeit von maximal 16.67ms entsteht.
- Der Bildschirm braucht eine gewisse Zeit, bis ein empfangenes Signal angezeigt wird. Diese Latenz des Bildschirms gibt der Hersteller mit 5ms an.
- Das verwendete Python-Skript auf dem Server benötigt einige Zeit um angezeigte Bilder zu aktualisieren bzw. die getätigte Eingabe via HoloLSL zu

senden. Dies kann als Teil der zu messenden Latenz gesehen werden, da dies eine inhärente Verzögerung des Systems ist und auch ohne eine Latenzmessung auftritt.

- Da beide Uhrzeiten manuell verglichen werden müssen, wächst der Zeitaufwand proportional mit der Messgenauigkeit bzw. der Messfrequenz und der Länge der Messung. Langzeitmessungen lösen deshalb weniger genau auf.

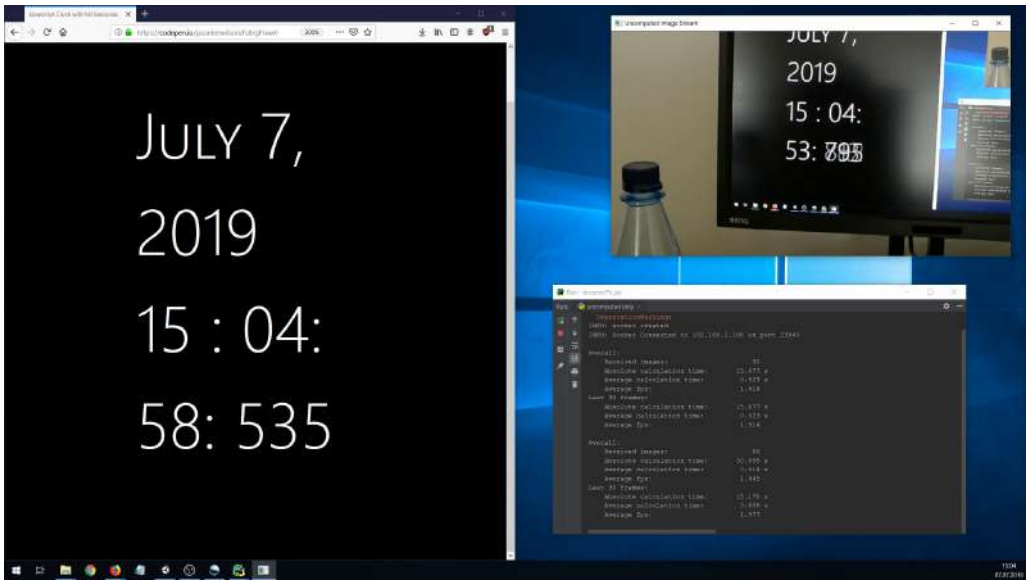


Abbildung 5.1: Schnappschuss während des Videos zur Bestimmung der Latenz des Reseach-Modes mit rechenaufwendigem Algorithmus

5.2 Allgemeine Systemperformance

Aus dem in Kapitel 4 erhobenen Logs sowie der erstellen Videos wurden Metriken zur allgemeinen Systemperformance erstellt, um einen Vergleich zwischen dem erstellen System und der offline-Objekterkennung im Video zu ziehen. Hierbei wurden die Logs (für die HoloLens) bzw. die Videos gegen eine manuell erstellte Grundwahrheit, von im Parcours vorkommenden Objekten, verglichen.

5.2.1 Erkennung von Fehlern

Die Erkennung von Fehlern geschah technischen Gründen und um den Umfang der Analyse den Rahmen dieser Arbeit nicht sprengen zu lassen mengenbasiert. Fehler wurden nur über das ganze Video bzw. den ganzen Log akkumuliert betrachtet.

Folgende Definition von Erkennungs- sowie Fehlerarten kam zum Einsatz:

TP: True Positive. Ein in der Realität vorkommendes Objekt wurde richtig erkannt.

TN: True Negative. Ein in der Realität nicht vorkommendes Objekt wurde nicht erkannt.

FP: False Positive. Ein nicht in der Realität vorkommendes Objekt wurde erkannt.

FN: False Negative. Ein in der Realität vorkommendes Objekt wurde nicht erkannt.

Als erkannt galt ein Objekt dann, wenn es mindestens in fünf aufeinander folgenden Frames erkannt wurde.

Daraus ergeben sich zwei Arten von Fehlern, die nicht berücksichtigt wurden:

- Existiert Objekt A an einer beliebigen Stelle, so kann jedes beliebige Objekt als A erkannt werden. Pro fälschlich als A erkanntem Objekt wird ein False Positives nicht berücksichtigt.
- Es Existiert lediglich Objekt A, Objekt B nicht. Objekt B wird als Objekt A erkannt. Hierbei finden sowohl ein False Negative als auch ein False Positive keine Beachtung.

5.2.2 Erhobene Metriken

Zur Bewertung der Sytemperformance wurden drei Metriken erhoben [Nic18]:

Accurary beschreibt wie oft das System insgesamt richtig liegt. Sie wird folgendermaßen berechnet: $\frac{TP+TN}{TP+TN+FP+FN}$

Precision. Wie häufig ist ein erkanntes Objekt richtig erkannt worden. Die Berechnung erfolgt folgendermaßen: $\frac{TP}{TP+FP}$

Recall Wie viele der zu erkennenden Objekte wurden erkannt. Der Recall wird nach dieser Formel berechnet: $\frac{TP}{TP+FN}$

5.2.3 Ergebnisse

Wie in Abbildung 5.2 zu sehen ist, gibt es zwischen den verwendeten Systemen nur geringe Unterschiede des Mittelwertes in den verwendeten Metriken.
Laut Kubinger, Rasch, und Moder (2009) [KRM09] kann ein Welch-Test auch ohne vorherige Prüfung auf Normalverteilung als Alternative zum Zweistichproben-t-Test verwendet werden. Zudem lässt er unterschiedlich viele Samples für die zu vergleichenden Populationen zu. Ein Welch-Test zeigt bei einem Signifikanzniveau von $\alpha = .05$ bei sowohl bei der Accuracy ($w(22) = 0.12,\ p \approx .90879$), der Precision ($w(22) = 0.93,\ p \approx .36426$) als auch beim Recall ($w(22) = -1.93,\ p \approx .06599$) keinen signifikanten Unterschied.
Diesem Umstand ist jedoch mit Vorsicht zu begegnen, da für die Videos lediglich zwei Samples vorlagen und deshalb die Normalverteilung nicht geprüft werden konnte.
Somit lässt sich unter Vorbehalt sagen, dass das entwickelte System bei den verwendeten Metriken, den oben genannten Einschränkungen in der Fehlererkennung sowie der für die Videos recht kleinen Samplegröße von zwei, keine gravierenden Einflüsse auf die Erkennungsperformance hat.

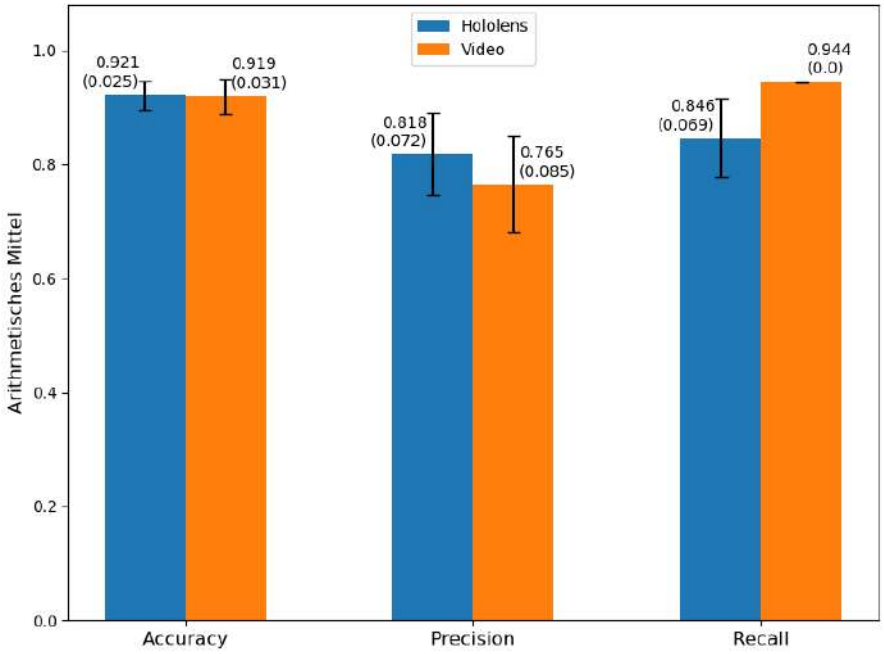


Abbildung 5.2: Accuracy, Precision und Recall der Objekterkennung im Vergleich zwischen dem erstellten System (HoloLens) und eines Videos

5.3 Latenzen

Als Latenz ist die Verzögerung verschiedener Vorgänge innerhalb des Systems zu betrachten. Untersucht wurden folgende Latenzen:

Positionsänderungsverzögerung Die Zeit, welche zwischen der Erkennung einer Änderung einer Position eines bereits angezeigten Rechtecks am Server und der Übernahme dieser Änderung in der HoloLens, vergeht.

Größenänderungsverzögerung Die Zeit, welche zwischen der Erkennung einer Änderung einer Größe eines bereits angezeigten Rechtecks am Server und der Übernahme dieser Änderung in der HoloLens, vergeht.

Klassenänderungsverzögerung Die Zeit, welche zwischen der Erkennung einer neuen Klasse und der Übernahme dieser neuen Klasse in der HoloLens, vergeht

Erkennungsverzögerung mit YOLOv3 Wie lange es am Beispiel von YOLOv3 dauert, bis ein neu sichtbares Objekt in der HoloLens als erkannt markiert wird.

Abbildung 5.3 zeigt diese Latenzen im Vergleich. Auffällig ist der Unterschied zwischen den Aktionen, bei denen eine neue Klasse und somit eine andere Identifikationsnummer (ID) gesendet wird und dem reinen Ändern der Größe und Position eines Rechtecks. Bei letzteren lag die Verzögerung mit durchschnittlich 0.577 Sekunden für Größenänderungen sowie 0.622 Sekunden für Positionsänderungen nicht weit entfernt von den 0.568 Sekunden gemessenen Verzögerung für die Übertragung des Videos, sodass die reine Übertragung von Ergebnissen vom Server auf die HoloLens mit neun Millisekunden für Größenänderungen und 54 Millisekunden für Positionsänderungen im Vergleich zur Gesamtverzögerung nur ein geringes Gewicht aufweist. In der Praxis

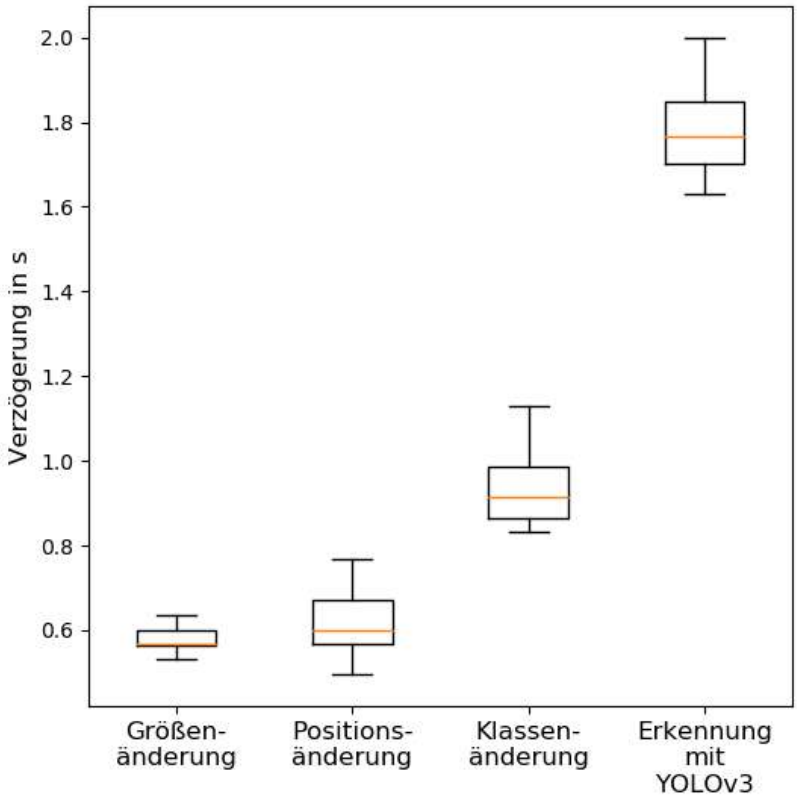


Abbildung 5.3: Latenzen verschiedener Anwendungsfälle im Vergleich

sind diese Werte von neun bzw. 54 Millisekunden alleine jedoch wahrscheinlich ohne große Bedeutung, da neu gesendete Ergebnisse oft vom empfangenen Bild abhängen, sofern der genutzte Bildverarbeitungsalgorithmus auf jedem empfangenen Bild unabhängig durchgeführt wird.

Für die Aktion der Klassenänderung lässt sich die höhere Latenz vor allem mit dem Design der HoloLens-App erklären. So wird eine neu empfangene ID erst dann angezeigt, wenn diese mindestens n-mal mal empfangen wurde. Während der Tests war dieser Parameter auf fünf gesetzt. Bei einer durchschnittlichen Bildfrequenz von verarbeiteten 13 Bildern pro Sekunde entsprechen dies ungefähr 0.385s. Zusammen mit der Latenz der Bildübertragung von ungefähr einer halben Sekunde entspricht dies einer knappen Sekunden, was sich mit dem Median der Verzögerung einer Klassenänderung deckt. Die Streuung dieser Verzögerung in Form einer Standardabweichung von 0.094 Sekunden lässt sich unter anderem durch eine Schwankung in der Übertragungsverzögerung erklären. Diese ist mit dem in Abbildung 3.7b zu sehenden Boxplot für eine 50ms andauernde Berechnungszeit zu vergleichen, da auf der genutzten Hardware YOLOv3 eine ähnlich lange Berechenungszeit pro Bild benötigt.

Für eine Erklärung der Differenz der Verzögerung zwischen einer Klassenänderung und der vollständigen Erkennung eines Objektes mit YOLOv3 wären weitere Untersuchungen sinnvoll. Als Vermutung für die höhere Latenz im Anwendungsfall YOLOv3 ließe sich anbringen, dass sich bei einer Klassenänderung ein Rechtecksobjekt an

genau der richtigen Position befindet und so als eine Art Cache fungieren könnte, da die HoloLens App dann lediglich Klassenlabel und ID ändern muss. Bei der Erkennung eines neuen Objektes ist dies nicht der Fall, sodass ein neues Rechtecksobjekt angelegt werden muss, was die erhöhte Verzögerung erklären könnte, da dies ein vergleichsweise teurer Vorgang ist.

5.4 Langzeitverhalten

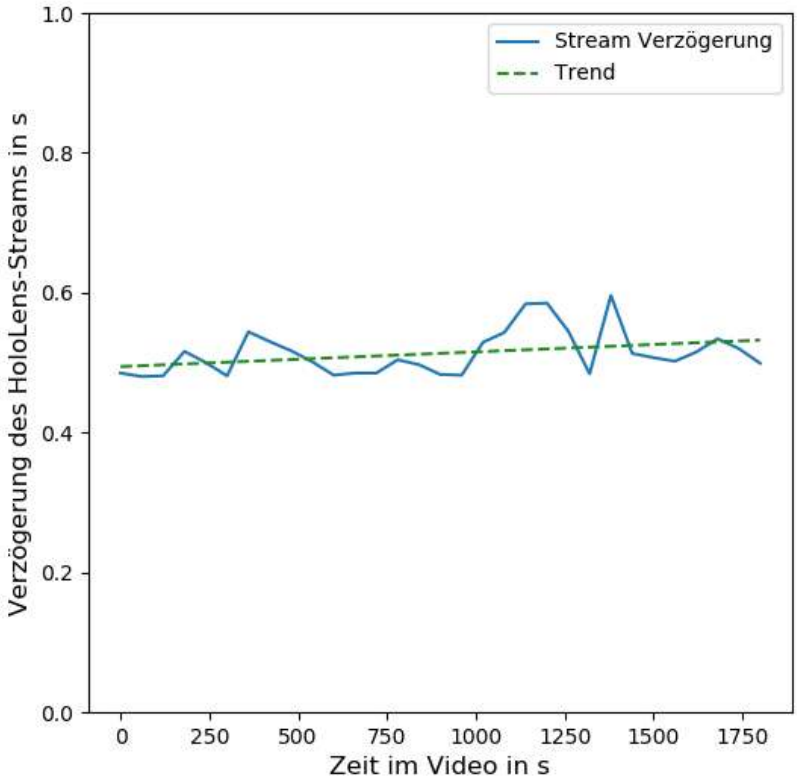


Abbildung 5.4: Verhalten der Verzögerung der Bildübertragung über 30 Minuten

Dieser Test dient dazu sicherzustellen, dass sich das System auch über längere Zeiträume vergleichbar in Hinblick auf die Verzögerung der Übertragung verhält. Auch wenn hierbei ebenfalls Latenzen gemessen werden, ist er von den oben durchgeführten Latenz-Tests abzugrenzen, da hier die Änderung der Latenz über einen Zeitraum im Vordergrund steht. Mögliche Ursachen für eine Verschlechterung der Latenz über die Zeit könnten sowohl auf Probleme der Software als auch auf Hardwareprobleme zurückgeführt werden.
Auf der Softwareebene können sich langsam füllende Buffer ein Symptom dafür sein, dass die Verarbeitung der Daten langsamer voranschreitet als deren Bereitstellung. Ist der Buffer ein First-In-First-Out Puffer (also eine Warteschlange), so arbeitet der verarbeitende Prozess bei nicht leerem Buffer nicht auf den aktuellsten Daten. Je länger die Schlange ist, desto älter sind die enthaltenden Daten. Füllt sich der Buffer während der Laufzeit immer weiter, so erhöht sich auf diese Weise auch

die Verzögerung. Weiterhin können Speicherlecks die Verarbeitungsgeschwindigkeit über einen längeren Zeitraum beeinträchtigen, da immer weniger Speicherplatz zur Verfügung steht und deshalb eventuell auf langsamere Speichermedien zurückgegriffen werden muss.

Auf der Hardwareebene kann thermische Drosselung auftreten, vor allem da die HoloLens ohne aktive Kühlung auskommt. Dabei handelt es sich um eine hardwareseitige Reduzierung der Leistung, um einer Überhitzung dieser entgegenzuwirken [Fis18].

In Abbildung 5.4 ist die Verzögerung des Bildstreams über einen Zeitraum von 30 Minuten aufgetragen. Als Berechnungsaufwand wurde wieder YOLOv3 gewählt. Der Aufbau der Messung entspricht im allgemeinen dem in Abschnitt 5.1 beschriebenen Vorgang. Auf Grund der Länge der Aufnahme wurde jedoch lediglich ein Messpunkt pro Minute angesetzt, sodass die gezeigte Auflösung nur ein Sechstel der sonst verwendeten Genauigkeit entspricht. Damit die Objekterkennung während der Messung unterschiedliche Objekte erkennen kann, wurde der Kamera der HoloLens zusätzlich eine Diashow mit erkennbaren Objekten präsentiert. Dies diente der Aufdeckung potentieller Speicherlecks in der Software und sollte den Berechnungsaufwand auf Seiten der HoloLens erhöhen, um eine thermische Drosselung zu provozieren.

Die in Abbildung 5.4 zu sehende Trendlinie zeigt über den Zeitraum von 30 Minuten einen Anstieg der Verzögerung von knapp 38 Millisekunden, was ungefähr 7.4% der durchschnittlichen Verzögerung während dieser Messung entspricht. Diese Differenz unterschreitet kurzzeitige Schwankungen deutlich (siehe 3.7a 50ms zB. bei 250s), sodass, zumindest für den gemessenen Zeitraum, keine subjektiven Performanceeinbußen durch die längere Laufzeit bemerkbar waren.

5.5 Performanz der Objekterkennung

Zur Bewertung der Leistung des Gesamtsystems unter Verwendung von YOLOv3 wurden folgende Eigenschaften untersucht:

- Die **Bildfrequenz des Streams** von der HoloLens an den Server bei verschieden vielen erkannten und dargestellten Objekten in einer Szene.
- Verhalten bei Änderungen der **Perspektive**.

5.5.1 Bildfrequenz in Abhängigkeit der Anzahl erkannter und dargestellter Objekte

Tabelle 5.1 zeigt die Bildfrequenz des übertragenden Streams sowie die ungefähre Prozessorauslastung der HoloLens bei unterschiedlich vielen Objekten in der Szene. Für jede Szene wurde das System neu gestartet und die Einstellung eine Minute lang betrachtet. Die angegebenen Werte für Bildfrequenz und Prozessorauslastung sind Durchschnittswerte dieser Zeitspanne. Die Prozessorauslastung wurde mit Hilfe des Device-Portals der HoloLens abgelesen, weshalb eine genaue Bestimmung nicht möglich war. Die Bildfrequenz entspricht den verarbeiteten Bildern auf Seiten des Servers sowie den gesendeten Antworten desselben.

Bezüglich der durchschnittlichen Bildfrequenz lässt sich keine Abhängigkeit zur Anzahl der Objekten in der Szene erkennen. Bezieht man die Prozessorauslastung

Dargestellte Objekte in Szene	Durchschnittliche Bildfrequenz pro Sekunde	Ungefähre Prozessorauslastung HoloLens
0	14.225	48 %
1	14.573	52 %
2	14.521	55 %
5	15.013	62 %
10	14.545	80 %

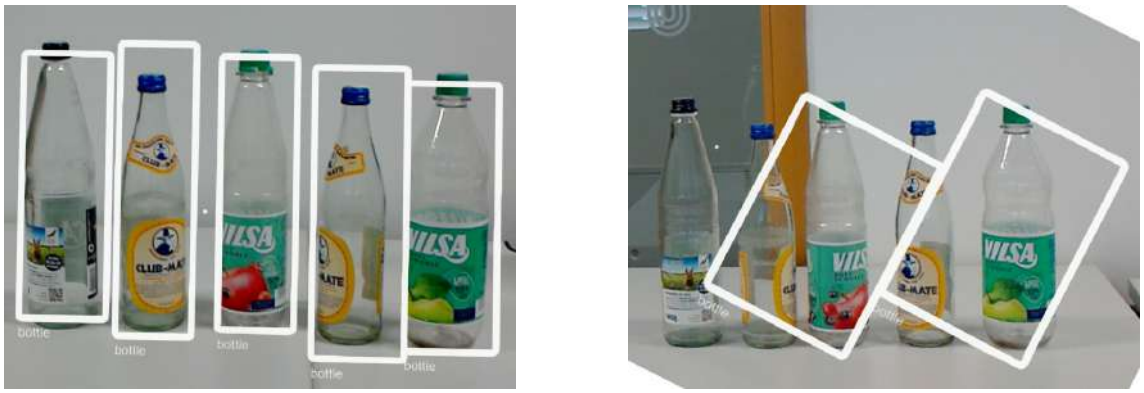
Tabelle 5.1: Bildfrequenz und CPU-Auslastung der HoloLens für bei variierender Objektanzahl

mit ein, so fällt auf, dass diese mit zunehmender Anzahl an Objekten ansteigt. Extrapoliert man diesen Anstieg ($f(x) = 48.2025 + 3.11043x$ mit $R^2 \approx 0.993$) linear, so wäre der Prozessor bei über 16 gleichzeitig zu zeichnenden Objekten vollständig ausgelastet. Dies wurde jedoch nicht explizit getestet.

5.5.2 Verhalten bei Perspektivänderungen

Ein Vorteil von Augmented Reality besteht in der potenziellen Mobilität der nutzenden Personen, sodass die, um virtuelle Objekte erweiterte, Umgebung aus frei wählbaren Perspektiven betrachtet werden kann. Diese Freiheit stellt jedoch eine Herausforderung für das Platzieren von virtuellen Objekten dar, da die Perspektive der nutzenden Person berücksichtigt werden muss. Eine andere Formulierung dieses Problem findet sich in dem als „Simultaneous localization and mapping“ (SLAM) [DB06] bekanntem Problem wieder. Dieses versucht die HoloLens mit Hilfe ihrer Sensorik zu lösen (s. auch Abschnitt 2.1.2). Die hierbei erstellte Tiefenkarte besitzt jedoch eine relativ kleine Auflösung und wird nur ein mal pro Sekunde aktualisiert. Abbildung 2.3 zeigt eine Visualisierung dieser. Eine Lösung für eine genauere Tiefenkarte entwickelten M Garon, P.-O. Boulet et al. 2016 mit Hilfe einer externen Tiefenkamera [GBD+16], die in dieser Arbeit jedoch keine Verwendung fand. In Folge dessen konnte die Orientierung, der unter Verwendung von YOLOv3 erkannten Objekte, relativ zur HoloLens bei der Platzierung der Rechtecke nicht berücksichtigt werden. Wie in Kapitel 3.2.2 beschrieben, wird lediglich ein Tiefenpunkt pro Rechteck genutzt. Daraus resultiert eine Orientierung für alle Rechtecke, bei der die Blickrichtung der nutzenden Person eine Normale zur Rechtecksfläche bildet. Diese Orientierung stimmt im Allgemeinen jedoch nicht mit der tatsächlichen Orientierung von Objekten in der Umgebung überein.

Als weiterer Freiheitsgrad des Menschen sind Kopfneigungen zu betrachten. Mit Kopfneigungen sind hier Drehungen des Kopfes um die Achse der Blickrichtung gemeint (Rollen/Wanken). Die durch die HoloLens gerenderten Rechtecke bleiben konstant relativ zum Kopf der tragenden Person, nicht zum Boden. Da keine Information über die Position der HoloLens an den Server gesandt werden, kann zum Beispiel im Fall der Objekterkennung mit YOLOv3 die Rotation relativ zum Boden der gesendeten Bilder nicht rekonstruiert werden. Dies stellt eine Problematik für die Erkennungskonfidenz von YOLOv3 bei geneigten Bildern dar, da die meisten Trainingsbilder des Coco-Datasets eine solche Rotation nicht aufweisen.



(a) (b)

Abbildung 5.5: Vergleich der Objekterkennung mit YOLOv3 beim Wanken des Kopfes. a) Nahezu kein Wanken, alle Flaschen werden korrekt erkannt. b) Wanken um ca. 27°. Es werden nur noch 2 Flaschen erkannt. Das Bild wurde für eine bessere Vergleichbarkeit manuell gedreht.

Abbildung 5.5 zeigt ein Beispiel an Hand von Flaschen. Werden bei aufrechter Kopfhaltung alle fünf Flaschen korrekt erkannt und gelabelt, so geschieht dies bei einem Kopfwanken von ungefähr 27° nur noch bei zwei Flaschen. Zu beachten ist, dass das Bild 5.5b manuell gedreht wurde. Die verwendeten Algorithmen der Bildverarbeitung (YOLOv3) erhielten in diesem Fall ein um 27° entgegen dem Uhrzeigersinn gedrehtes Bild.

Ferner fällt auf, da YOLOv3 lediglich die Koordinaten des Mittelpunktes des erkannten Objektes sowie dessen Breite und Höhe zurück gibt, dass die Bounding Boxen für Objekte, deren größte Ausdehnung nicht parallel zu einer Außenkante der Bounding Box verläuft, kein minimal umspannendes Rechteck um das Objekt bilden. Dies ist ebenfalls in Abbildung 5.5b zu sehen.

6. Ausblick

Dieses Kapitel gibt einen Überblick über mögliche weitergehende Forschungsfragen, sowie Verbesserungsvorschläge für das entwickelte System.

6.1 Verbesserungen

In diesem Abschnitt werden Verbesserungen aufgeführt, deren Implementierung möglicherweise eine bessere Performance des Systems zur Folge hätten.

6.1.1 Anderes Trackingverfahren

Das Verfolgen von erkannten Objekten bzw. von Algorithmen der Bildverarbeitung gekennzeichneten Stellen im Bild dient der fehlerfreien Darstellung der hervorgehobenen Stellen in der HoloLens.
Das implementierte Trackingverfahren blickt lediglich auf die euklidische Distanz zwischen den Schwerpunkten erkannter Objekte und verbindet neue Objekte mit der niedrigsten Distanz zu bereits bestehenden Objekten. Nachteile dieses Verfahren sind unter anderem folgende:

Rechenaufwendiges Verfahren Die Objekt*erkennung* und -verfolgung arbeiten getrennt voneinander. Daher muss für jeden Frame eine Berechnung beider Algorithmen erfolgen. Da die Objekterkennung ein teures Verfahren ist, ließe sich durch eine Kombination beider Aspekte eine weniger rechenaufwendige Lösung finden.

Bildränder Verschwindet ein Objekt aus dem Bildrand, wird dies nicht modelliert und ein Wiederauftreten des selben Objektes erzeugt eine neue ID.

Wiederverfolgen verlorener Objekte Wird ein Objekt für mehrere Frames nicht verfolgt, so ist der verwendete Algorithmus meist nicht in der Lage dieses Objekt wieder als das selbe Objekt zu identifizieren und weiter zu verfolgen.

Ein komplexeres Trackingverfahren könnte in der Lage sein, einige oder alle dieser Unzulänglichkeiten zu eliminieren. Einen Vergleich verschiedener Trackingverfahren zogen Wu et al. 2013 [WLY13].

6.1.2 Kameralatenz verringern

Als KameraLatenz wird die Zeit zwischen dem Eintreten eines Ereignisses und dem Eintreffen der digitalen Repräsentation dieses Ereignisses am dedizierten PC bezeichnet. Je länger dieses Zeitdelta ist, desto höher ist die empfundene Verzögerung des Systems, was sich negativ auf die empfundene Qualität auswirkt. Im Verlauf dieser Arbeit wurden verschiedene Möglichkeiten der Bildübertragung von der HoloLens zum dedizierten PC in Erwägung gezogen. Trotzdem konnte die Verzögerung nicht maßgeblich unter eine halbe Sekunde verringert werden. Dies lag vor allem an Hardware-Limitierungen der HoloLens. Das Zurückgreifen auf andere Hardware, wie z.B. eine externe Kamera könnte die Verzögerung wesentlich reduzieren.

6.1.3 Mehr Informationsaustausch

Zwischen Server und Client wurden nur die nötigsten Informationen ausgetauscht. Der Client sendet ausschließlich Bilddaten an den Server. Dieser antwortet lediglich mit Informationen zu der Position, Größe, Klasse sowie eindeutigen Identifikation einzelner Bildausschnitte. Würden mehr Informationen ausgetauscht, könnte dies der Gesamtqualität des Systems zuträglich sein.

Telemetrie der HoloLens für Objekterkennung nutzen

Eine Möglichkeit des erweiterten Informationsaustausches bestände im Senden von Telemetriedaten der HoloLens an den Server. Mit Hilfe dieser Daten wäre es dem Server möglich die Position der HoloLens im Raum zu berechnen und so dem in Kapitel 5.5.2 beschriebenen Abfall der Erkennungsrate bei Kopfneigungen entgegenzuwirken.

Telemetrie-Historie

Des Weiteren wäre es möglich, dass die HoloLens eine Telemetrie-Historie ihrer eigenen Bewegung bereit hält, um die Abbildung empfangener Ergebnisse nicht in Abhängigkeit von der aktuellen Position zu berechnen. Stattdessen könnte hierfür diejenige Position verwendet werden, welche die HoloLens zu dem Zeitpunkt inne hielt als die Bildinformationen für ebenjenes Ergebnis gesendet wurden. Damit eine korrekte Zuordnung der Ergebnisdaten mit der Telemetrie-Historie erfolgen kann, müssen dem Kamerabildstream von der HoloLens zum Server sowie dem Ergebnisdatenstream Zeitstempel hinzugefügt werden. Im Kamerabildstream zum Server geben diese an, zu welcher Zeit der einzelne Frame gesendet wurde. Der jeder Übertragung der Ergebnisse kann daraufhin dieser Zeitstempel angehängt werden, sodass die Zuordnung mit der Telemetrie-Historie auf der HoloLens erfolgen kann. Diese Vorgehensweise könnte zur Eliminierung der Ungenauigkeiten, welche durch die Latenzen der Übertragungen der Bild- und Ergebnisdaten entstehen, beitragen.

6.1.4 Verschiedene AR-Plattformen

Eine Maßnahme, welche zur allgemeinen Verbesserung des Systems führen könnte, ist die Verwendung eines anderen (neueren) AR-Headsets. Hier böte sich beispielsweise das Nachfolgemodell der HoloLens, die HoloLens 2[1] an. Diese bietet neben Kameras mit höherer Auflösung und Bildfrequenz [MZ19b] auch verbesserte Rechenleistung in Form eines, nun ARM-basierten, Prozessors. Ob die damit erreichbare Rechenleistung genügt, um (zumindest einige) Algorithmen der Bildverarbeitung direkt auf dem Head-Mounted-Display zu berechnen, müssen weitere Untersuchungen zeigen.

[1] Hololens 2 von Microsoft: https://www.microsoft.com/de-de/hololens/ (last checked: 22.11.2019)

7. Fazit

Im Rahmen dieser Arbeit wurde ein System zur Echtzeitvisualisierung von Algorithmen der Bildverarbeitung im Rahmen von Augmented Reality (AR) entwickelt. Das erstellte System besteht dabei aus der AR-Plattform, wofür eine Microsoft HoloLens genutzt wurde, und einem dediziertem Server, welcher für die Berechnung der Algorithmen in Echtzeit zuständig ist. Auf der AR-Plattform können Algorithmen, welche bestimmte Bedingungen erfüllen, dargestellt werden. Die Darstellung erfolgt über beschriftete Rechtecke, welche an die korrekten Stellen über dem Sichtfeld der nutzenden Person gerendert werden.
Als, sich durch die gesamte Arbeit ziehendes, Beispiel wurde als Algorithmus eine Objekterkennung auf Basis von YOLOv3 verwendet. Das System wurde anhand diese Beispiels der Objekterkennung im Rahmen einer Nutzungsstudie weiter untersucht und dabei mit der Präsentation der Objekterkennung über ein konventionelles Video verglichen.
Hierbei stellte sich heraus, dass die Hypothese, dass es Versuchspersonen leichter fällt die Möglichkeiten und Grenzen eines Algorithmuses zu erkennen und vorherzusagen, sollte dieser ihnen innerhalb des entwickelten Systems präsentiert werden, teilweise bestätigen ließ. Die Versuchspersonen stimmten der gleichen Frage nach der Kontrolle über das verwendete System signifikant mehr zu, wenn das entwickelte System zum Einsatz kam. Des Weiteren wurden die vom System erkannten Objekte nach verschiedenen Metriken genauer eingeschätzt.

Neben der Nutzungsstudie wurde eine technische Auswertung durchgeführt, welche die Performance des entwickelten Systems anhand verschiedener Merkmale untersuchen sollte. Es wurden verschiedene Latenzen innerhalb des Systems gemessen. So dauert die Übertragung eines Frames vom AR-Headset zum Server durchschnittlich 568ms. Weitere Latenzen wie Größen-, Positions- und Klassenänderungen wurden untersucht. Zudem wurde die Verzögerung vom Sichtbarwerden bis zum Erkennen eines Objektes auf durchschnittlich 1.785 Sekunden beziffert.
Darüber hinaus wurde das Langzeitverhalten des Systems in Bezug auf eventuell sich akkumulierende Latenzen überprüft. Hierbei konnte keine relevante Steigerung der Latenz über einen Zeitraum von 30 Minuten festgestellt werden. Eine Untersuchung der Performanz in Abhängigkeit von der in der Szene beobachteten Objekte konnte lediglich einen Anstieg der Prozessorauslastung der HoloLens feststellen. Daraus

ergab sich eine theoretische, hardwarebedingte Grenze bei über 16 Objekten. Des Weiteren wurden einige Überlegungen zur Abhängigkeit der Erkennungsqualität von der Perspektive des betrachteten Objekts durchgeführt. Hier sind besonders Neigungen des Kopfes der Versuchsperson zu nennen. Bei einem Wanken von 27° sank die Erkennungsrate rapide.

A. Anhang

Im folgenden werden einige Zusatzmaterialien aufgelistet, welche zur Arbeit beigetragen haben und auf die verwiesen wurde.

A.1 Versuchsdaten

Ein Google-Spreadsheet, welches alle Ergebnisse des Fragebogens enthält:
https://bit.ly/35sH9Ur

A.2 Versuchsdokumente

Zu den Versuchsdokumenten gehören die Versuchsbeschreibung, welche allen Versuchspersonen zu Beginn durchlasen sowie der Fragebogen mit allen Fragen, die während des Versuchs beantwortet wurden.

A.2.1 Versuchsbeschreibung

Die Versuchsbeschreibung wurde jeder Versuchsperson vor dem Start des Experiments ausgehändigt.

Versuchsbeschreibung

Hallo liebe Versuchperson!
Vielen Dank, dass Du dich dazu entschlossen hast bei diesem Experiment mitzumachen!
Du wirst heute an einem Versuch teilnehmen, bei dem es um einen Vergleich von Objekterkennung in Augmented Reality mit konventioneller Objekterkennung geht.
Ziel dieser Studie ist es, herauszufinden, ob und welche Vorteile es hat Objekterkennung in Augmented Reality zu präsentieren. Dafür werden dir immer mal wieder Fragen in Form eines Fragebogens gestellt.

Versuchsablauf

Die Augmented Reality, die du erfahren wirst, wird durch die Microsoft Hololens erzeugt werden. Diese ist ein Headset welches du aufsetzt und dir virtuelle Objekte in dein normales Sichtfeld projiziert. Diese Objekte werden beschriftete Rechtecke sein, die sich um jenen Objekten befinden, deren Beschriftung sie tragen. Dir wird also eine Objekterkennung präsentiert. Mit dieser Objekterkennung durchläufst du einen Parcours bzw. umrundest einen Tisch mit Gegenständen, die erkannt werden können. Schaue dich dabei ausgiebig um und lasse die Objekterkennung auf dich wirken.
Nach dem Durchqueren des Parcours wäre es schön, wenn du ein paar Antworten zu deinen Erfahrungen geben könntest.
Das Ganze wird dann noch einmal mit einem leicht verändertem Parcours durchgeführt.
Im einzelnen läuft der gesamte Versuch also folgendermaßen ab:

1. Du durchläuft den Parcours mit der Hololens.
2. Du beantwortest einige Fragen zu deiner Einschätzung des erlebten.
3. Innerhalb dieses Fragebogens siehst du ein Video von einem ähnlichen Parcours mit der gleichen Objekterkennung.
4. Weitere Fragen über das Video erwarten Dich.
5. Du wiederholst die Punkte 1. bis 4. noch einmal mit einem anderen Parcours.

Vielen Dank für Deine Zeit und viel Spaß beim Erkennen von Objekten!

A.2.2 Fragebogen

Der Fragebogen, wie ihn alle Versuchpersonen ausfüllten. Die Bearbeitung während des Experimentes fand digital statt.

Fragebogen Echtzeitvisualisierung in AR

Umfrage zur Bachelorarbeit: "Augmented Reality als Plattform zur Echtzeitvisualisierung von Algorithmen der Bildverarbeitung"

* Erforderlich

Einleitung

Im Folgenden wirst du je zweimal eine Objekterkennung mit Hilfe von Augmented Reality sowie in Form eines Videos erfahren können. Daraufhin werden dir jeweils ein paar Fragen gestellt. Bitte beantworte diese nach deinem besten Gewissen.
Vielen Dank und Viel Spaß!

Hololens - Aufbau A

Du hast dir gerade verschiedene Gegenstände angeguckt und wurdest dabei von einer Objekterkennung unterstützt. Bitte beantworte nun diese Fragen

1. **Welche Objekte hast du während des Experimentes erkannt?** *
   Wählen Sie alle zutreffenden Antworten aus.

   - [ ] Weinglas
   - [ ] Schale
   - [ ] Löffel
   - [ ] Gabel
   - [ ] Messer
   - [ ] Apfel
   - [ ] Banane
   - [ ] Teller
   - [ ] Flasche
   - [ ] Person
   - [ ] Maus
   - [ ] Tastatur
   - [ ] Bildschirm
   - [ ] Orange
   - [ ] Stuhl
   - [ ] Sonnenbrille
   - [ ] Tisch
   - [ ] Lampe
   - [ ] Smartphone
   - [ ] Etui
   - [ ] Fernbedienung
   - [ ] Buch
   - [ ] Zahnbürste
   - [ ] Birne
   - [ ] Uhr
   - [ ] Regenschirm
   - [ ] Zitrone
   - [ ] Plastiktüte
   - [ ] Pflanze
   - [ ] Becher
   - [ ] Stift
   - [ ] Sonstiges: _____

2. **Welche Objekte wurden von der Objekterkennung erkannt?** *
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Weinglas
- [ ] Schale
- [ ] Löffel
- [ ] Gabel
- [ ] Messer
- [ ] Apfel
- [ ] Banane
- [ ] Teller
- [ ] Flasche
- [ ] Person
- [ ] Maus
- [ ] Tastatur
- [ ] Bildschirm
- [ ] Orange
- [ ] Stuhl
- [ ] Sonnenbrille
- [ ] Tisch
- [ ] Lampe
- [ ] Smartphone
- [ ] Etui
- [ ] Fernbedienung
- [ ] Buch
- [ ] Zahnbürste
- [ ] Birne
- [ ] Uhr
- [ ] Regenschirm
- [ ] Zitrone
- [ ] Plastiktüte
- [ ] Pflanze
- [ ] Becher
- [ ] Stift
- [ ] Sonstiges: _____

3. **Als wie gut bewertest du die gerade von dir erlebte Objekterkennung?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

4. **Wie gründlich konntest du die Objekterkennung testen?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

5. **Wie sicher bist du dir in der Einschätzung der Erkennungsqualität?**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr unsicher	◯	◯	◯	◯	◯	◯	◯	sehr sicher

6. **Fandest du, dass du die Kontrolle über die Objekterkennung hattest?**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
starke Ablehnung	◯	◯	◯	◯	◯	◯	◯	starke Zustimmung

7. **Wie genau war die räumliche Abbildung der Rechtecke um erkannte Objekte?** *
   Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr ungenau	◯	◯	◯	◯	◯	◯	◯	sehr genau

8. **Konntest du sehen, welche Bezeichnungen die Objekterkennung zugeordnet hat?**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

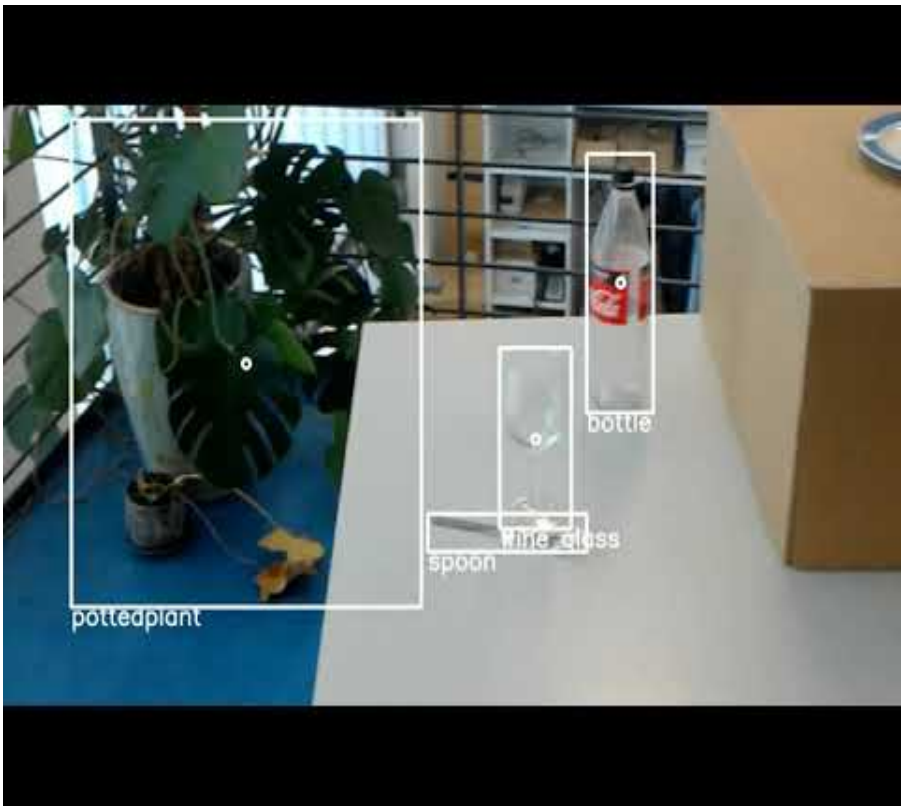
9. **Wie sehr würde sich deiner Meinung nach deine Einschätzung ändern, solltest du das Experiment wiederholen?** *
   Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
meine Einschätzung bliebe vollständig erhalten	◯	◯	◯	◯	◯	◯	◯	Meine Einschätzung würde sich komplett ändern

Videovergleich - Aufbau A

Vergleiche nun deine Erfahrungen mit folgendem Video

Objekterkennung im Video



http://youtube.com/watch?v=ZQCg_IZi6J4

10. **Welche Objekte hast du während des Videos erkannt?** *
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Weinglas
- [ ] Schale
- [ ] Löffel
- [ ] Gabel
- [ ] Messer
- [ ] Apfel
- [ ] Banane
- [ ] Teller
- [ ] Flasche
- [ ] Person
- [ ] Maus
- [ ] Tastatur
- [ ] Bildschirm
- [ ] Orange
- [ ] Stuhl
- [ ] Sonnenbrille
- [ ] Tisch
- [ ] Lampe
- [ ] Smartphone
- [ ] Etui
- [ ] Fernbedienung
- [ ] Buch
- [ ] Zahnbürste
- [ ] Birne
- [ ] Uhr
- [ ] Regenschirm
- [ ] Zitrone
- [ ] Plastiktüte
- [ ] Pflanze
- [ ] Becher
- [ ] Stift
- [ ] Sonstiges: _____

11. **Welche Objekte wurden von der Objekterkennung erkannt?** *
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Weinglas
- [ ] Schale
- [ ] Löffel
- [ ] Gabel
- [ ] Messer
- [ ] Apfel
- [ ] Banane
- [ ] Teller
- [ ] Flasche
- [ ] Person
- [ ] Maus
- [ ] Tastatur
- [ ] Bildschirm
- [ ] Orange
- [ ] Stuhl
- [ ] Sonnenbrille
- [ ] Tisch
- [ ] Lampe
- [ ] Smartphone
- [ ] Etui
- [ ] Fernbedienung
- [ ] Buch
- [ ] Zahnbürste
- [ ] Birne
- [ ] Uhr
- [ ] Regenschirm
- [ ] Zitrone
- [ ] Plastiktüte
- [ ] Pflanze
- [ ] Becher
- [ ] Stift
- [ ] Sonstiges: _____

12. **Als wie gut bewertest du die Objekterkennung in diesem Video?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

13. **Wie gründlich konntest du die Objekterkennung testen?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

14. **Wie sicher bist du dir in der Einschätzung der Erkennungsqualität?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr unsicher	◯	◯	◯	◯	◯	◯	◯	sehr sicher

15. **Fandest du, dass du die Kontrolle über die Objekterkennung hattest?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
starke Ablehnung	◯	◯	◯	◯	◯	◯	◯	starke Zustimmung

16. **Wie genau war die räumliche Abbildung der Rechtecke um erkannte Objekte?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr ungenau	◯	◯	◯	◯	◯	◯	◯	sehr genau

17. **In wie weit würde sich deiner Meinung nach deine Einschätzung ändern, wenn du dir das Video noch einmal anschauen würdest?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
meine Einschätzung bliebe vollständig erhalten	◯	◯	◯	◯	◯	◯	◯	Meine Einschätzung würde sich komplett ändern

Hololens - Aufbau B

18. **Welche Objekte hast du während des Experimentes erkannt?** *
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Weinglas
- [ ] Schale
- [ ] Löffel
- [ ] Gabel
- [ ] Messer
- [ ] Apfel
- [ ] Banane
- [ ] Teller
- [ ] Flasche
- [ ] Person
- [ ] Maus
- [ ] Tastatur
- [ ] Bildschirm
- [ ] Orange
- [ ] Stuhl
- [ ] Sonnenbrille
- [ ] Tisch
- [ ] Lampe
- [ ] Smartphone
- [ ] Etui
- [ ] Fernbedienung
- [ ] Buch
- [ ] Zahnbürste
- [ ] Birne
- [ ] Uhr
- [ ] Regenschirm
- [ ] Zitrone
- [ ] Plastiktüte
- [ ] Pflanze
- [ ] Becher
- [ ] Stift
- [ ] Sonstiges: _____

19. **Welche Objekte wurden von der Objekterkennung erkannt?** *
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Weinglas
- [ ] Schale
- [ ] Löffel
- [ ] Gabel
- [ ] Messer
- [ ] Apfel
- [ ] Banane
- [ ] Teller
- [ ] Flasche
- [ ] Person
- [ ] Maus
- [ ] Tastatur
- [ ] Bildschirm
- [ ] Orange
- [ ] Stuhl
- [ ] Sonnenbrille
- [ ] Tisch
- [ ] Lampe
- [ ] Smartphone
- [ ] Etui
- [ ] Fernbedienung
- [ ] Buch
- [ ] Zahnbürste
- [ ] Birne
- [ ] Uhr
- [ ] Regenschirm
- [ ] Zitrone
- [ ] Plastiktüte
- [ ] Pflanze
- [ ] Becher
- [ ] Stift
- [ ] Sonstiges: _____

20. **Als wie gut bewertest du die gerade von dir erlebte Objekterkennung?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

21. **Wie gründlich konntest du die Objekterkennung testen?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

22. **Wie sicher bist du dir in der Einschätzung der Erkennungsqualität?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr unsicher	◯	◯	◯	◯	◯	◯	◯	sehr sicher

23. **Fandest du, dass du die Kontrolle über die Objekterkennung hattest?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
starke Ablehnung	◯	◯	◯	◯	◯	◯	◯	starke Zustimmung

24. **Wie genau war die räumliche Abbildung der Rechtecke um erkannte Objekte?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr ungenau	◯	◯	◯	◯	◯	◯	◯	sehr genau

25. **Konntest du sehen, welche Bezeichnungen die Objekterkennung zugeordnet hat?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

26. **Wie sehr würde sich deiner Meinung nach deine Einschätzung ändern, solltest du das Experiment wiederholen?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
meine Einschätzung bliebe vollständig erhalten	◯	◯	◯	◯	◯	◯	◯	Meine Einschätzung würde sich komplett ändern

Videovergleich - Aufbau B

Vergleiche nun deine Erfahrungen mit folgendem Video

Objekterkennung im Video

http://youtube.com/watch?v=v5qh7TvSYd8

27. **Welche Objekte hast du während des Videos erkannt?** *

Wählen Sie alle zutreffenden Antworten aus.

- [ ] Weinglas
- [ ] Schale
- [ ] Löffel
- [ ] Gabel
- [ ] Messer
- [ ] Apfel
- [ ] Banane
- [ ] Teller
- [ ] Flasche
- [ ] Person
- [ ] Maus
- [ ] Tastatur
- [ ] Bildschirm
- [ ] Orange
- [ ] Stuhl
- [ ] Sonnenbrille
- [ ] Tisch
- [ ] Lampe
- [ ] Smartphone
- [ ] Etui
- [ ] Fernbedienung
- [ ] Buch
- [ ] Zahnbürste
- [ ] Birne
- [ ] Uhr
- [ ] Regenschirm
- [ ] Zitrone
- [ ] Plastiktüte
- [ ] Pfanze
- [ ] Becher
- [ ] Stift
- [ ] Sonstiges: _____

28. **Welche Objekte wurden von der Objekterkennung erkannt?** *
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Weinglas
- [ ] Schale
- [ ] Löffel
- [ ] Gabel
- [ ] Messer
- [ ] Apfel
- [ ] Banane
- [ ] Teller
- [ ] Flasche
- [ ] Person
- [ ] Maus
- [ ] Tastatur
- [ ] Bildschirm
- [ ] Orange
- [ ] Stuhl
- [ ] Sonnenbrille
- [ ] Tisch
- [ ] Lampe
- [ ] Smartphone
- [ ] Etui
- [ ] Fernbedienung
- [ ] Buch
- [ ] Zahnbürste
- [ ] Birne
- [ ] Uhr
- [ ] Regenschirm
- [ ] Zitrone
- [ ] Plastiktüte
- [ ] Pflanze
- [ ] Becher
- [ ] Stift
- [ ] Sonstiges: _____

29. **Als wie gut bewertest du die Objekterkennung in diesem Video?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

30. **Wie gründlich konntest du die Objekterkennung testen?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

31. **Wie sicher bist du dir in der Einschätzung der Erkennungsqualität?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr unsicher	◯	◯	◯	◯	◯	◯	◯	sehr sicher

32. **Fandest du, dass du die Kontrolle über die Objekterkennung hattest?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
starke Ablehnung	◯	◯	◯	◯	◯	◯	◯	starke Zustimmung

33. **Wie genau war die räumliche Abbildung der Rechtecke um erkannte Objekte?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr ungenau	◯	◯	◯	◯	◯	◯	◯	sehr genau

34. **In wie weit würde sich deiner Meinung nach deine Einschätzung ändern, wenn du dir das Video noch einmal anschauen würdest?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
meine Einschätzung bliebe vollständig erhalten	◯	◯	◯	◯	◯	◯	◯	Meine Einschätzung würde sich komplett ändern

Allgemeine Performance AR

Erinnere dich noch einmal an deine Erfahrungen mit der Hololens und versuche folgende Fragen zu beantworten

35. **Wie gut fandest du das Design der Applikation?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

36. **Die Applikation war einfach zu bedienen.** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schwer	◯	◯	◯	◯	◯	◯	◯	sehr einfach

37. **Die Objekterkennung funktioniert fehlerfrei.** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
Voller Fehler	◯	◯	◯	◯	◯	◯	◯	Fehlerfrei

38. **Wie geeignet fandest du die Darstellungsform der erkannten Objekte als Rechtecke** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

39. **Wie könnten erkannte Objekte besser dargesellt werden?**

40. **Wie gut reagierte die Objekterkennung auf deine Bewegungen?** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr schlecht	◯	◯	◯	◯	◯	◯	◯	sehr gut

41. **Die Objekterkennung behinderte mich in meiner Orientierung.** *
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
Sehr starke Einschränkung	◯	◯	◯	◯	◯	◯	◯	Keine Einschränkung

42. **Welche weiteren Algorithmen der Bildverarbeitung könntest du dir von diesem System visualisiert vorstellen?**

43. **In welchen Anwendungsbereichen kannst du dir ein solches System vorstellen?** *

44. **Wie könnte die Objekterkennung verbessert werden?** *

Dein Wohlbefinden ist uns wichtig

Bitte beantworte zum Schluss ein paar Fragen, die deine Konvenienz betreffen

45. **Nach oder während des Versuchs... ***
Wählen Sie alle zutreffenden Antworten aus.

- [ ] war mir Unwohl
- [ ] hatte ich Kopfschmerzen
- [ ] wurde mir übel
- [ ] empfand ich Müdigkeit
- [ ] erfuhr ich Bewegungsinstabilität / war mir schwindelig
- [ ] Sonstiges: _____

46. **Das Headset war angenehm zu tragen ***
Markieren Sie nur ein Oval.

	1	2	3	4	5	6	7	
sehr unangenehm	◯	◯	◯	◯	◯	◯	◯	sehr angenehm

47. **Für welchen Zeitraum könntest du dir vorstellen, das Headset am Stück zu tragen? ***
_____
Beispiel: 8:30 Uhr

48. **Was fiel dir unangenehm beim Tragen des Systems auf? ***
_____
_____
_____
_____
_____

49. **Ist dir sonst noch etwas aufgefallen?**
_____
_____
_____
_____
_____

Bereitgestellt von
Google Forms

Abkürzungsverzeichnis

AdB Algorithmen der Bildverarbeitung. v, 2, 13, 18, 20, 22, 24
AR Augmented Reality. v, 1, 3, 4

BM Brunner-Munzel-Test. 31, 32

CSL Cognitive-Systems-Labs. 27

fov Field of View. 6

HMD Head-Mounted-Display. 4, 5

ID Identifikationsnummer. 15, 17, 40–42
IMU Inertiale Messeinheit. 5

LSL Lab Streaming Layer. 11, 12, 18

UWP Universal Windows Platform. 5

Glossar

Augmented Reality (dt. „Erweiterte Realität“) Erweiterung der physikalischen Umgebung mit computergenerierten Objketen bzw. Informationen in Echtzeit. v, 1–4, 6, 12, 17, 27, 44, 49

average Precision (dt. „durchschnittliche Genauigkeit“) Metrik, um die Leistungsfähigkeit von Objekterkennern zu bestimmen. 11

Bounding Box (dt. „Begrenzungsrahmen“) Ein Rechteck, welches ein Objekt umspannt bzw. einschließt. 8, 9, 16, 17, 45

Companion-Computer hier: Computer, welcher zur externen Berechung von best. Algorithmen genutzt wird. Hier auch „Server“ genannt. 11, 21, 22

convolutional Layer (dt. „faltende Schicht“) Teil eines Convolutional Neural Network. Über die Eingabe wird eine Faltungsmatrix, auch Filter genannt, bewegt.. 8, 10

Convolutional Neural Network (dt. „faltendes Neuronales Netzwerk“) Eine Art eines Neuronalen Netzwerkes. Findet häufig Anwendung in bildverarbeitenden Algorithmen. 8, 10

Echtzeitvisualisierung Eine Darstellung ohne (merkliche) zeitliche Verzögerung. v, 2, 12, 13, 27, 28, 49

Head-Mounted-Display Ein Bildschirm, welcher in unmittelbarer Nähe zu den Augen vor dem Gesicht getragen wird. 4, 17, 48

HoloLens Eine Augmented-Reality-Datenbrille von Microsoft, die es erlaubt AR wahrzunehmen. v, 2, 4–6, 11–21, 23–26, 28, 29, 31–38, 40–44, 47–49

Immersion („Eintauchen“) Effekt von virtuellen Umgebungen, bei der diese von den nutzenden Personen als reale Welt wahrgenommen werden. 1

Inertialen Messeinheit Ein elektronisches Bauteil, welches mit Hilfe von Beschleunigungs- und Winkelgeschwindigkeitssensoren eigene Lage im Raum misst. 5

Latenz Eine Verzögerung innerhalb eines Systems. v, 21, 22, 37, 38, 40–42, 48, 49

Nutzungsstudie Auch Nutzerstudie. Eine Studie bei der eine Software auf ihre Benutzbarkeit untersucht wird. v, 49

OpenCV Eine quelloffene Bibliothek für Bildverarbeitung. 2

residuale Layer (dt. „Restschicht“)Teil eines Neural Networks. Beinhaltet Abbkürzungen, die es der Eingabe ermöglichen mehrere Layer im Netz zu überspringen. 8

Salienz (dt. „Auffälligkeit“) Ein (oftmals visueller) Reiz, der dem Bewusstsein auffälliger erscheint. 2, 13

Segmentierung Unterteilung eines Bildes in (semantisch) zusammenhängende Regionen. 2, 10

Spatial Mapping (dt. „Räumliche Kartierung“) Wahrnehmung und Kartierung der räumlichen Umgebung durch ein Softwaresystem. 2, 5, 16

Time of Flight Kamera (dt. „Flugzeitkamera“) Eine Kamera, die anhand der Flugzeit des Lichtes Entfernungen misst. Ein Pixel einer solchen Kamera gibt nicht Farben sondern Entfernungen an. 5

Tracking (dt. „Verfolgen“) Das Verfolgen von Objekten über Zeit und/oder Raum. 2, 15, 16, 24, 47

YOLOv3 Eine Objekterkennungssoftware, die auf Bildern 80 verschiedene Objekte erkennen und deren Position angeben kann. v, 8, 9, 11, 17, 20–22, 27, 40, 41, 43–45, 49

Literaturverzeichnis

[AHJ+01] R. S. Allison, L. R. Harris, M. Jenkin, U. Jasiobedzka, and J. E. Zacher. Tolerance of temporal delay in virtual environments. In *Proceedings IEEE Virtual Reality 2001*, pages 247–254, March 2001. doi:10.1109/VR.2001.913793.

[Azu97] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997. URL: https://doi.org/10.1162/pres.1997.6.4.355, arXiv:https://doi.org/10.1162/pres.1997.6.4.355, doi:10.1162/pres.1997.6.4.355.

[BM00] Edgar Brunner and Ullrich Munzel. The nonparametric behrens-fisher problem: Asymptotic theory and a small-sample approximation. *Biometrical Journal*, 42(1):17–25, 2000. doi:10.1002/(SICI)1521-4036(200001)42:1<17::AID-BIMJ17>3.0.CO;2-U.

[Bou18] Tristan Stenner; Chadwick Boulay. labstreaminglayer, Nov 2018. URL: https://github.com/sccn/labstreaminglayer/wiki [cited 05.09.2019].

[Cai19] Paul Cairns. *Doing Better Statistics in Human-computer Interaction*. Cambridge University Press, 2019.

[Cen16] Microsoft News Center. Microsoft announces global expansion for hololens, Okt 2016. URL: https://news.microsoft.com/en-au/2016/10/12/microsoft-announces-global-expansion-for-hololens/#sm.0000hacx3x7r6denyg81m2pmrthzn [cited 31.07.2019].

[CFA+11] Julie Carmigniani, Borko Furht, Marco Anisetti, Paolo Ceravolo, Ernesto Damiani, and Misa Ivkovic. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1):341–377, Jan 2011. URL: https://doi.org/10.1007/s11042-010-0660-6, doi:10.1007/s11042-010-0660-6.

[DB06] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, June 2006. doi:10.1109/MRA.2006.1638022.

[EBF18] Martin Eckert, Matthias Blex, and Christoph Friedrich. Object detection featuring 3d audio localization for microsoft hololens. In *BIOSTEC 2018 : Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies*, A deep learning based sensor substitution approach for the blind, pages 555–561. SCITEPRESS - Science and Technology Publications, 2018.

[Fis18] Ryan Fischer. What's thermal throttling and how to prevent it, Jun 2018. URL: https://www.techspot.com/article/1638-what-is-thermal-throttling/ [cited 28.09.2019].

[FJDV18] T. Frantz, B. Jansen, J. Duerinck, and J. Vandemeulebroucke. Augmenting microsoft's hololens with vuforia tracking for neuronavigation. *Healthcare Technology Letters*, 5(5):221–225, 2018. doi:10.1049/htl.2018.5079.

[Flu10] Holger Fluehr. *Flugzeugsensoren*, pages 261–285. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. URL: https://doi.org/10.1007/978-3-642-01612-7_10, doi:10.1007/978-3-642-01612-7_10.

[Fro17] Irasianty Frost. *Statistische Testverfahren, Signifikanz und p-Werte*. Springer, 2017.

[Gan18] Rogith Gandhi. R-cnn, fast r-cnn, faster r-cnn, yolo — object detection algorithms, Jul 2018. URL: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

[GBD+16] M. Garon, P. Boulet, J. Doironz, L. Beaulieu, and J. Lalonde. Real-time high resolution 3d data on the hololens. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 189–191, Sep. 2016. doi:10.1109/ISMAR-Adjunct.2016.0073.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2014. URL: http://dx.doi.org/10.1109/CVPR.2014.81, doi:10.1109/cvpr.2014.81.

[GPS72] Gene V Glass, Percy D. Peckham, and James R. Sanders. Consequences of failure to meet assumptions underlying the fixed effects analyses of variance and covariance. *Review of Educational Research*, 42(3):237–288, 1972. URL: https://doi.org/10.3102/00346543042003237, arXiv:https://doi.org/10.3102/00346543042003237, doi:10.3102/00346543042003237.

[Hem19] Wanja Hemmerich. Gepaarter t-test in spss: Voraussetzungen und annahmen, 2019. URL: https://statistikguru.de/spss/gepaarter-t-test/voraussetzungen-und-annahmen.html [cited 09.11.2019].

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[Hui18] Jonathan Hui. Image segmentation with mask r-cnn, Apr 2018. URL: https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272.

[Jay18] Prakash Jay. The intuition behind retinanet, Mrz 2018. URL: https://medium.com/@14prakash/the-intuition-behind-retinanet-eb636755607d [cited 12.09.2019].

[Jel16] Albert Jelica. Hololens: Hier sind die spezifikationen von microsofts ar-brille, Mai 2016. URL: https://windowsarea.de/2016/05/microsoft-hololens-spezifikationen/ [cited 31.07.2019].

[Kat18] Ayoosh Kathuria. What's new in yolo v3?, Apr 2018. URL: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b [cited 01.08.2019].

[Kre15] Matthias Kremp. Hologramm-brille stiehlt windows 10 die schau, Jan 2015. URL: https://www.spiegel.de/netzwelt/gadgets/microsoft-stellt-windows-10-und-computerbrille-hololens-vor-a-1014204.html/ [cited 31.07.2019].

[KRM09] Klaus D. Kubinger, Dieter Rasch, and Karl Moder. Zur legende der voraussetzungen des t-tests für unabhängige stichproben. *Psychologische Rundschau*, 60(1):26–27, 2009. URL: https://doi.org/10.1026/0033-3042.60.1.26, `arXiv:https://doi.org/10.1026/0033-3042.60.1.26`, `doi:10.1026/0033-3042.60.1.26`.

[Kro18] Felix Kroll. Hololslbridge, Sep 2018. URL: https://gitlab.csl.uni-bremen.de/sbliefert/HoloLSL [cited 12.09.2019].

[LAE+16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.

[LKK96] Lisa M. Lix, Joanne C. Keselman, and H. J. Keselman. Consequences of assumption violations revisited: A quantitative review of alternatives to the one-way analysis of variance "f"test. *Review of Educational Research*, 66(4):579–619, 1996. URL: http://www.jstor.org/stable/1170654.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *Lecture Notes in Computer Science*, page 740–755, 2014. URL: http://dx.doi.org/10.1007/978-3-319-10602-1_48, `doi:10.1007/978-3-319-10602-1_48`.

[MC06] Feissal Damaa Mark Claypool, Kajal Claypool. The effects of frame rate and resolution on users playing first person shooter games. In *Multimedia Computing and Networking 2006*, volume 6071, 2006. URL: https://doi.org/10.1117/12.648609, `doi:10.1117/12.648609`.

[McC19] Matt Zeller; Jesse McCulloch. Rendering, Feb 2019. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/rendering [cited 31.07.2019].

[MNT+17] C. M. Morales Mojica, N. V. Navkar, N. V. Tsekos, D. Tsagkaris, A. Webb, T. Birbilis, and I. Seimenis. Holographic interface for three-dimensional visualization of mri on hololens: A prototype platform for mri guided neurosurgeries. In *2017 IEEE 17th International Conference*

on Bioinformatics and Bioengineering (BIBE), pages 21–27, Oct 2017. doi:10.1109/BIBE.2017.00-84.

[MRH92] E. N. Rubinstein M. R. Harwell. Summarizing monte carlo results in methodological research: The one-and two-factor fixed effects anova cases. *Journal of educational statistics*, 17(4):315–339, 1992. URL: https://journals.sagepub.com/doi/pdf/10.3102/10769986017004315.

[MTUK95] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–292. International Society for Optics and Photonics, 1995.

[MZ18a] Brandon Bray Matt Zeller. Hololens (1st gen) hardware details, Mrz 2018. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details [cited 31.07.2019].

[MZ18b] David Gedye Matt Zeller. Hololens research mode, Mai 2018. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/research-mode [cited 09.09.2019].

[MZ19a] Brandon Bray Matt Zeller. Gestures, Feb 2019. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/gestures [cited 31.07.2019].

[MZ19b] Chris Edmonds Matt Zeller. Locatable camera, Jun 2019. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/locatable-camera [cited 22.11.2019].

[MZ19c] Lia Martinez Matt Zeller. Voice input, Feb 2019. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/voice-input [cited 31.07.2019].

[Nay18] Sunita Nayak. Deep learning based object detection using yolov3 with opencv (python / c++), Aug 2018. URL: https://www.learnopencv.com/deep-learning-based-object-detection-using-yolov3-with-opencv-python-c/ [cited 01.08.2019].

[Nic18] Chris Nicholson. Accuracy, precision, recall or f1?, Mrz 2018. URL: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 [cited 02.11.2019].

[Ols17] Johannes Schönberger; Pawel Olszta. Hololensforcv, Aug 2017. URL: https://github.com/microsoft/HoloLensForCV/wiki [cited 09.09.2019].

[PLW08] Youngmin Park, Vincent Lepetit, and Woontack Woo. Multiple 3d object tracking for augmented reality. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pages 117–120, Washington, DC, USA, 2008. IEEE Computer Society. URL: https://doi.org/10.1109/ISMAR.2008.4637336, doi:10.1109/ISMAR.2008.4637336.

[PP10] M. Petrou and C. Petrou. *Image Processing: The Fundamentals*. John Wiley & Sons, 2 edition, 2010. doi:10.1002/9781119994398.

[Red18] Ali Redmon, Joseph und Farhadi. YOLOv3: An Incremental Improvement. *arXiv e-prints*, page arXiv:1804.02767, Apr 2018. arXiv:1804.02767.

[RF16] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. *arXiv e-prints*, page arXiv:1612.08242, Dec 2016. arXiv:1612.08242.

[RG15] Santosh Redmon, Joseph und Divvala and Ali Girshick, Ross und Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *arXiv e-prints*, page arXiv:1506.02640, Jun 2015. URL: https://ui.adsabs.harvard.edu/abs/2015arXiv150602640R, arXiv:1506.02640.

[Rob12] Judy Robertson. Likert-type scales, statistical methods, and effect sizes. *Commun. ACM*, 55(5):6–7, Mai 2012. URL: http://doi.acm.org/10.1145/2160718.2160721, doi:10.1145/2160718.2160721.

[Ros18] Adrian Rosebrock. Simple object tracking with opencv, Jul 2018. URL: https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/ [cited 17.09.2019].

[Sat19] Patrick Farley; Mike Jacobs; Michael Satran. Windows device portal overview, Aug 2019. URL: https://docs.microsoft.com/en-us/windows/uwp/debug-test-perf/device-portal [cited 10.09.2019].

[SW65] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965. URL: http://www.jstor.org/stable/2333709.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. arXiv:1409.1556.

[Vin11] Giovanni Vincenti. Reality-virtuality continuum in according to milgram, takemura, utsumi and kishino (1994), Apr 2011. URL: https://commons.wikimedia.org/wiki/File:Reality-Virtuality_Continuum.svg [cited 09.12.2019].

[vuf] vuforia. Developing vuforia engine apps for hololens. URL: https://library.vuforia.com/articles/Training/Developing-Vuforia-Apps-for-HoloLens [cited 28.11.2019].

[WLY13] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[Zel18] Brandon Zeller, Matt und Bray. Locatable camera in unity, Mrz 2018. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/locatable-camera-in-unity [cited 05.09.2019].

[ZHAH10] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld. Impact of frame rate and resolution on objective qoe metrics. In *2010 Second International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 29–34, June 2010. doi:10.1109/QOMEX.2010.5518277.