



Universität Regensburg

**Philosophische Fakultät III**  
**Sprach-, Literatur- und Kulturwissenschaften**  
**Institut für Information und Medien, Sprache und Kultur (I:IMSK)**  
**Lehrstuhl für Informationswissenschaft / Lehrstuhl für Medieninformatik**

---

Algorithmen für Mensch-Maschine-Interaktion  
Modul: INF-M34  
Wintersemester 2018/2019  
Leitung: Prof. Dr. Bernd Ludwig

## **Eyetracking in der Küche**

Markus Bosek  
Matr.-Nr.: 1622088  
2. Semester M.Sc. Medieninformatik  
E-Mail: markus.bosek@stud.uni-regensburg.de

Johannes Büttner  
Matr.-Nr.: 1650726  
2. Semester M.Sc. Medieninformatik  
E-Mail: johannes1.buettnerk@stud.uni-regensburg.de

Tobias Hauser  
Matr.-Nr.: 1535759  
4. Semester M.Sc. Medieninformatik  
E-Mail: tobias-manfred.hauser@stud.uni-regensburg.de

Abgegeben am 08.04.2019

# **Inhalt**

<b>1</b>	<b>Motivation .....</b>	<b>4</b>
<b>2</b>	<b>Vorüberlegungen und Konzept.....</b>	<b>6</b>
2.1	Assistenzbedarf .....	6
2.2	Assistenzziele und geplante Umsetzung.....	7
<b>3</b>	<b>Experiment.....</b>	<b>8</b>
3.1	Aufbau .....	8
3.2	Technisches Setup .....	9
<b>4</b>	<b>Auswertung und Ergebnisse .....</b>	<b>10</b>
4.1	Eyetrackingdaten.....	10
4.2	Exploration und Vorverarbeitung .....	12
4.3	Koordinatentransformationen.....	17
4.4	Auswertung der Blickdaten.....	24
4.5	Ergebnisse und Einschränkungen .....	26
<b>5</b>	<b>Ausblick .....</b>	<b>29</b>
	<b>Literaturverzeichnis.....</b>	<b>31</b>

## Abbildungsverzeichnis

Abbildung 1: Augenlandmarken OpenFace 2.0 .....	10
Abbildung 2: Plot der Augenposition und Blickrichtung.....	16
Abbildung 3: Bilder des 3D-Modells der Küche.....	16
Abbildung 4: Verschiebung, Rotation und Skalierung (v.l.n.r.) der Kamera .....	22
Abbildung 5: X-Achse (rot), Y-Achse (grün) und Z-Achse (blau) der Kamera nach einer Transformation.....	22
Abbildung 6: Augenpositionen im Koordinatensystem des 3D-Modells ohne Rotation und Translation .....	23
Abbildung 7: Augenpositionen im Koordinatensystem des 3D-Modells nach Rotation und Translation .....	23
Abbildung 8: Definierte AOIs .....	25
Abbildung 9: Heatmap für AOI „Wand zwischen Schränken“ (Fr 13-02).....	25
Abbildung 10: Gesichtslandmarken und Blickrichtungen von OpenFace 2.0 ....	27

## 1 Motivation

Die Lebenserwartung in Deutschland steigt jedes Jahr um drei Monate und Forscher sagen voraus, dass die meisten Menschen in 60 Jahren bis zu 100 Jahre und länger leben werden. Umgekehrt sinkt die Zahl der jüngeren Menschen im Vergleich zum Anstieg der älteren Bevölkerung dramatisch (Neßelrath, Lu, Schulz, Frey, & Alexandersson, 2011). Dieser demografische Wandel führt zu einem Ungleichgewicht zwischen alten und jungen Menschen. Körperliche und kognitive Beeinträchtigungen führen dazu, dass ältere Menschen mehr Unterstützung in ihrem täglichen Leben benötigen. Dies wird zum Problem, wenn nicht genügend Pflegekräfte zur Verfügung stehen, um die ältere Generation zu unterstützen (Neßelrath et al., 2011). Daraus ergibt sich ein enormes Potenzial im Bereich der Lebenshilfe für behinderte und ältere Menschen, die an allen Arten von Behinderungen, wie z.B. neurologischen Veränderungen, Sehschärfenänderungen und vestibulären Beeinträchtigungen, leiden (Nehmer, Becker, Karshmer & Lamm, 2006). Eine Lösung für dieses zukunftsnahe Problem sind Assistenzsysteme für intelligente Umgebungen. Ein intelligentes Zuhause, das auch seine alternden Bewohner im Alltag unterstützt, steigert einerseits die Lebensqualität, andererseits verlängert sich durch solche System auch die Zeit, in der ältere Menschen in ihren eigenen vier Wänden ein autonomes Leben führen können, ohne auf menschliche Unterstützung angewiesen zu sein (Neßelrath et al., 2011).

Als Grund für eine mangelnde Verbreitung moderner Assistenzsystemen wird oft eine fehlende Akzeptanz der Endnutzer angenommen (Lübbecke, 2016). Allerdings sei gerade die Akzeptanz seitens der älteren Menschen für den Einsatz von Assistenzsystemen unumgänglich (Gräuel & Speilerberg, 2007). Brush et al. (2011) zeigten in ihrer Studie vier Hindernisse, die eine breitere Akzeptanz erschweren:

- hohe Kosten
- Inflexibilität
- schlechte Handhabbarkeit
- Schwierigkeiten in der Erreichung ausreichender Sicherheit (Brush et al., 2011)

Gräuel und Speilerberg (2007) kamen in ihrer Studie zum Ergebnis, dass die meisten Befragten weiterhin selbstständig wohnen wollen, wobei 25 % nichts gegen die Verwendung moderner Technologien hätten, um das alleinige Wohnen trotz Beeinträchtigungen zu ermöglichen. Weiter kam heraus, dass die Akzeptanz neuer Technik sogar ähnlich hoch ist wie das Vertrauen in professionelle soziale Dienste. Schlussendlich zeigen ihre Ergebnisse, dass auch wenn die Technikkompetenz besonders bei Bildschirmanwendungen durchaus bedeutsam ist, nicht nur ältere Menschen mit Computererfahrung bereit sind moderne Techniken zu nutzen, sondern auch weniger affine Menschen sich eine Bereitschaft vorstellen können (Gräuel & Speilerberg, 2007).

## 2 Vorüberlegungen und Konzept

### 2.1 Assistenzbedarf

Anhand der im Verlauf des Kurses zur Verfügung gestellten Logdaten, die im Rahmen der Masterarbeit von Alexander Frummet<sup>1</sup> erhoben wurden, können Bedürfnisse und Gewohnheiten der Nutzer während eines Kochvorgangs abgeleitet werden. Um ein Eyetracking System in der Küche sinnvoll einsetzen zu können, wurde aus diesem gegebenen Korpus der Assistenzbedarf ermittelt.

Hierbei ist sehr auffällig, wie oft die Nutzer bei den Kochexperimenten nach dem nächsten Zubereitungsschritt gefragt haben.

**Tabelle 1: Assistenzbedarf**

Äußerungen (Nutzer)	ermittelter Assistenzbedarf
<ul style="list-style-type: none"> <li>• Ok, und dann?</li> <li>• Wie geht es (dann) weiter?</li> <li>• Ok, dann?</li> <li>• Wie geht's weiter?</li> <li>• Und jetzt?</li> <li>• Und danach?</li> <li>• ....</li> </ul>	<ul style="list-style-type: none"> <li>• Kontaktaufnahme mittels Eyetracking</li> <li>• Entlastung des Nutzers</li> <li>• fortlaufender, flüssiger Kochvorgang</li> </ul>

---

<sup>1</sup>[https://wiki.mi.ur.de/arbeiten/detecting\\_domain-specific\\_information\\_needs\\_in\\_conversational\\_search\\_discourse](https://wiki.mi.ur.de/arbeiten/detecting_domain-specific_information_needs_in_conversational_search_discourse)

## 2.2 Assistenzziele und geplante Umsetzung

Das während des Semesters bereits ermittelte und besprochene Assistenzziel, das in diesem Experiment untersucht werden soll, definiert sich nach Wandke (2005), als Orientierungsassistentz. Diese Art der Unterstützung ermöglicht das Setzen und Ändern von Zielen. Im Kontext der Kontaktaufnahme, regt diese Assistenz den Zielwechsel an. Konkret bedeutet dies, den aktuellen Task zu beenden und den Nächsten zu beginnen.

Zunächst war angedacht, die Abarbeitung der einzelnen Tasks aufgrund des ermittelten Assistenzbedarfs zu automatisieren. Dies sollte umgesetzt werden, indem der Nutzer ein definiertes Objekt (z.B. Tablet) fokussiert, um den nächsten Rezeptschritt zu starten, ohne dass er zusätzlich noch mit Sprache interagieren muss. Die Ermittlung des Blickpunktes soll dabei über ein Kamerasystem unter Verwendung des Frameworks OpenFace 2.0<sup>2</sup> gehandhabt werden. Durch Erfassen der Blickdaten beim Beenden, bzw. Starten jedes Tasks, wäre es zusätzlich möglich, die Dauer dieser Tasks, als auch die Überwachung welchen Task der Nutzer gerade ausführt, zu ermitteln.

Als angenommene Rahmenbedingungen sollte als Systemumgebung ein 3D-Modell der Küche vorliegen. Zudem sollte eine Umgebung zur Verfügung gestellt werden, das über das Framework die Bilder der Kameras analysiert und die Blicke des Nutzers in das Koordinatensystem des 3D-Modells umrechnet. So könnten Blicke auf bestimmte *areas of interest* (AOIs) in diesem Modell erfasst werden.

---

<sup>2</sup> <https://github.com/TadasBaltrusaitis/OpenFace>

## 3 Experiment

### 3.1 Aufbau

Das Experiment wurde größtenteils in der Küche in der Künstlergarderobe der Universität Regensburg durchgeführt. Es fanden 13 Versuche verteilt auf fünf Tage im Zeitraum vom 25.02.2019 bis 01.03.2019 (Montag bis Freitag) statt. An jedem Tag wurde ein Rezept gekocht. Die Benennung und Nummerierung der folgenden Rezepte wurde entsprechend zu den auf der e-Learning Plattform GRIPS vorhandenen Drehbücher vorgenommen:

- Montag: Hähnchen süß-sauer (Rezept 1; 3 Probanden)
- Dienstag: Gefüllte Spinat-Pfannkuchen (Rezept 3; 2 Probanden)
- Mittwoch: Dal mit Fladenbrot (Rezept 4; 3 Probanden)
- Donnerstag: Spaghetti alla Carbonara mit Salat (Rezept 5; 3 Probanden)
- Freitag: Koteletts mit Champignon-Soße und Nudeln (Rezept 6; 2 Probanden)

Die Versuchsdurchführung am Dienstag fand in der Teeküche des Lehrstuhls Psychologie statt (Raumnummer PT 4.0.108).

Während des ganzen Experiments waren die Probanden dazu angehalten, ihre Vorgehensweisen und Überlegungen laut mitzusprechen (*think aloud* Prinzip).

Die Rezepte wurden den Probanden Schritt für Schritt auf einem vor ihnen platzierten Tablet angezeigt und per Sprachausgabe vorgelesen.



### 3.2 Technisches Setup

Für die Durchführung des Experiments wird eine Probanden- und eine Versuchsleiteranwendung verwendet. Den Probanden werden mit der Probandenanwendung Informationen und Anweisung zum aktuellen Task bereitgestellt. Nach der Fertigstellung eines Schritts, bestätigen die Probanden per Touchinteraktion. Zusätzlich existiert eine Testleiterapp. Mit Hilfe dieser, können die Testleiter den Kochvorgang überwachen und teils beeinflussen. Es besteht die Möglichkeit, einen Timer zu stellen, falls dieser von den Probanden gewünscht ist, die auftretenden Informationsbedürfnisse zu loggen und einen Alarm zu setzen, um eine Warnung zu bekommen, falls sich der Lebensmittelvorrat dem Ende neigt.

Bei den Experimenten am Montag und Dienstag, war es den Probanden möglich, den nächsten Rezeptschritt durch Klicken eines „Weiter“-Buttons anzeigen zu lassen. Bei den Experimenten von Mittwoch bis Freitag wurde den Probanden der nächste Rezeptschritt erst angezeigt, nachdem sie auf dem Tablet das Beenden des Schrittes bestätigt haben und zugleich die Testleiter auf der Testleiterapp das Anzeigen des nächsten Schrittes bestätigten, indem sie auf der Anwendung auf „Weiter“ klickten.

Alle Interaktionen der beiden Apps, sowie der Audiostream werden in LSL (Lab Streaming Layer) abgespeichert. Zudem sind zwei Kameras installiert. Eine Kamera ist für die Aufzeichnung der Eyetrackingdaten zuständig, die separat abgespeichert wurden.

Die zweite Kamera kam bei den Experimenten ab Mittwoch zum Einsatz, um zusätzliche Videodaten zu den Experimenten zu erhalten, falls diese später für eine Vergleichsanalyse der notwendig wären, um die Eyetrackingdaten zu verifizieren. Die beiden Kameras waren an der Küchenzeile im linken oberen Eck positioniert, sodass die Arbeitsfläche, der Herd und auch der Nutzer erfasst werden konnten.

## 4 Auswertung und Ergebnisse

In diesem Kapitel wird die Datenausgangslage beschrieben und es wird das Vorgehen bei der Exploration, Vorverarbeitung und Auswertung der Eyetrackingdaten dargestellt. Abschließend werden die Ergebnisse und Einschränkungen diskutiert. Der verwendete Code und die Daten sind dem Abgabeverzeichnis beigelegt. Die durchgeführten Schritte können in den jeweiligen Jupyter Notebooks nachvollzogen werden.

### 4.1 Eyetrackingdaten

Die Eyetrackingdaten werden mit der OpenFace 2.0 Software erzeugt und liegen als CSV Dateien im entsprechenden Format vor. Jeder Datensatz beinhaltet die Positionsdaten der definierten Gesichtslandmarken und der Blicke und allgemeine Metadaten. Die Metadaten setzen sich aus den Spalten *frame* (Nummerierung des Frames), *face\_id* (Id des erkannten Gesichts), *timestamp* (Timestamp in Sekunden), *confidence* (Konfidenz der Landmarkenerkennung) und *success* (Erfolgreiches Tracking eines Gesichts) zusammen. Die Blickdaten bestehen aus je einer Spalte für die X, Y und Z Dimension des normierten Richtungsvektors für jedes Auge. Die Blickvektoren des rechten Auges sind dabei in den Spalten *gaze\_0* (*gaze\_0\_x*, *gaze\_0\_y* und *gaze\_0\_z*) und die des linken Auges in den Spalten *gaze\_1* (*gaze\_1\_x*, *gaze\_1\_y* und *gaze\_1\_z*) angegeben. Die Daten zu den Gesichtslandmarken bestehen unter anderem aus den Spalten zu den Augenlandmarken (Baltrusaitis, Zadeh, Lim & Morency, 2018)<sup>3</sup>:

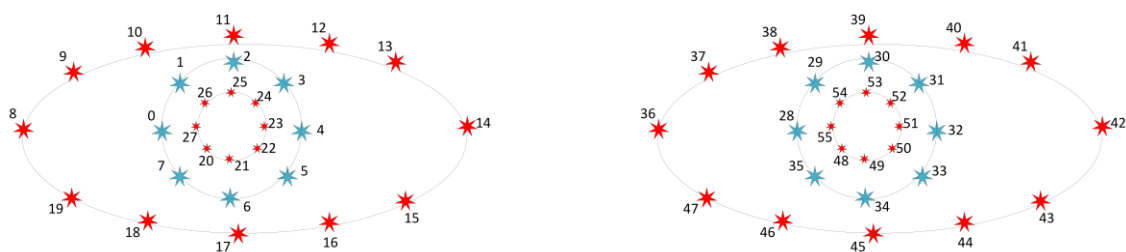


Abbildung 1: Augenlandmarken OpenFace 2.0<sup>4</sup>

<sup>3</sup> <https://github.com/TadasBaltrusaitis/OpenFace/wiki/Output-Format>

<sup>4</sup>

[https://raw.githubusercontent.com/wiki/TadasBaltrusaitis/OpenFace/images/eye\\_lmk\\_markup.png](https://raw.githubusercontent.com/wiki/TadasBaltrusaitis/OpenFace/images/eye_lmk_markup.png)

Die 3D Koordinaten dieser Augenlandmarken liegen jeweils in den Spalten *eye\_lmk\_X\_i*, *eye\_lmk\_Y\_i* und *eye\_lmk\_Z\_i* vor. Zu den bereits beschriebenen Daten beinhaltet das OpenFace 2.0 Format noch weitere Landmarken und andere Informationen. Für die Auswertung im Rahmen dieser Arbeit sind diese Informationen jedoch nicht relevant.

Das Koordinatensystem, in welchem alle Positionsdaten angegeben sind, ist durch die verwendete Kamera definiert. Für jeden Frame werden die Achsen folgendermaßen repräsentiert:

- X-Achse: Von links nach rechts, wobei links negativ und rechts positiv ist.
- Y-Achse: Von oben nach unten, wobei oben negativ und unten positiv ist.
- Z-Achse: Je weiter weg, desto höher der Wert (Tiefe der Kamera)

Der Ursprung dieses Koordinatensystems ist die Kamera. Betrachtet man eine Bildaufnahme der Kamera, ist ein Punkt mit X- und Y-Koordinaten gleich null also horizontal und vertikal in der Mitte des Bildes. Laut Entwickler der OpenFace 2.0 Software wird das Kamerakoordinatensystem als Weltkoordinatensystem angegeben. Für den weiteren Verlauf der Auswertung und insbesondere bei der Transformation der Koordinaten wird angenommen, dass es sich bei der Abbildungsmatrix zur Transformation der Koordinaten in das Koordinatensystem des 3D-Modells der Küche um eine affine Abbildung handelt (Baltrusaitis et al., 2018).

## 4.2 Exploration und Vorverarbeitung

Um einen Überblick über die Beschaffenheit der gesammelten Daten zu bekommen, werden die aufgezeichneten Eyetrackingdaten zunächst exploriert und anschließend in Vorbereitung auf die Auswertung vorverarbeitet. Dazu werden die vorliegenden Rohdaten (CSV-Dateien von OpenFace 2.0 generiert) eingelesen und es werden für jede Datei bestimmte Kennzahlen berechnet bzw. ausgegeben:

### Frames:

Anzahl der analysierten Frames - entspricht dem Wert der *frame* Spalte (Nummerierung des Frames) des letzten Frames in der Datei.

### Dauer (s):

Dauer der Analyse in Sekunden - entspricht dem Wert der *timestamp* Spalte des letzten Frames in der Datei.

### FPS:

Analysierte Frames pro Sekunde – naive Berechnung mit dem Wert der *frame* Spalte geteilt durch den Wert der *timestamp* Spalte des letzten Frames in der Datei.

### Gesicht:

Anteil an der gesamt Framezahl und Anzahl der Frames mit erkannten Gesichtslanmarken – bestimmt durch Frames, deren 3D Koordinaten der Augenlandmarke 0 (*eye\_lmk\_X\_0*, *eye\_lmk\_Y\_0* und *eye\_lmk\_Z\_0*) nicht gleichzeitig null sind.

### Blicke:

Anteil an der gesamt Framezahl und Anzahl der Frames mit erkannten Blicken bzw. Blickrichtungen – bestimmt durch Frames, deren 3D Koordinaten des Blickvektors des rechten Auges (*gaze\_0\_x*, *gaze\_0\_y* und *gaze\_0\_z*) nicht gleichzeitig null sind.

### Success und Konfidenz:

Anteil an der gesamt Framezahl – bestimmt durch entsprechende Filterung der Datensätze nach den Spalten *success* und *confidence*.

Anhand der definierten Kennzahlen soll unter anderem ein erster Einblick in die Qualität und Zusammensetzung der Eyetrackingdaten ermöglicht werden. Für die Filterung der Datensätze mittels der *confidence* Spalte wird ein Wert von 90% angesetzt. Alle Frames mit einem Konfidenzwert größer oder gleich 0,9 werden somit als konfidente bzw. erfolgreiche Landmarkenerkennung gewertet. Die Tabelle 2 stellt die festgelegten Kennzahlen für alle in Form von CSV-Dateien vorliegenden Rohdaten dar.

Die Tabelle zeigt, dass einige Dateien die Aufnahme über einen längeren Zeitraum als das jeweilige Experiment beinhalten. Des Weiteren ist zu erkennen, dass die Anzahl der Frames mit erfolgreicher Landmarken-, Blick- und Gesichtserkennung für alle Dateien verhältnismäßig gering ist. Im Abschnitt 4.5 wird dazu näher auf die Faktoren eingegangen, welche möglicherweise für diese schlechte Datenqualität verantwortlich sein könnten. Die Werte der Kennzahlen zu den Blicken bzw. zu Success und Konfidenz in Verbindung zu der FPS Kennzahl zeigen, dass die FPS Anzahl für höhere Erkennungsraten geringer ist. Diese Beobachtung kann so begründet werden, dass der Rechenaufwand zur Bestimmung der Positionsdaten der Landmarken, der Blicke und anderer Informationen durch die OpenFace Software bei erfolgreicher Gesichtserkennung erhöht ist. Durch die Durchführung der Analyse auf einem Raspberry Pi, lässt sich auch der deutliche Unterschied begründen (*Fr 11-00*: 8,90 vs. *Mi 13-06*: 0,71), da dieser Rechner im allgemeinen schwache Rechenleistung hat. Außerdem ist festzuhalten, dass die Spalten zu den Positionsdaten der 3D-Augenlandmarken auch bei nicht erfolgreicher Erkennung Werte ungleich null beinhalten.

Im nächsten Schritt werden jeweils die für die Auswertung relevanten Frames extrahiert. Es werden also nur diejenigen Frames herangezogen, welche in der *success* Spalte den Wert 1 bzw. in der *confidence* Spalte einen Wert größer oder gleich 0,9 aufweisen. Außerdem werden die Daten zusätzlich über die Augenlandmarken und Blickrichtungen gefiltert. Dementsprechend werden die Datensätze mit Werten in den Spalten *eye\_lmk\_X\_0*, *eye\_lmk\_Y\_0* und *eye\_lmk\_Z\_0* bzw. *gaze\_0\_x*, *gaze\_0\_y* und *gaze\_0\_z* gleich null entfernt. Die Datensätze vom Dienstag werden ebenfalls entfernt, da die zugehörigen Versuche in einer anderen Küche durchgeführt wurden und für diese Küche kein 3D-Modell vorliegt. Nach dieser Vorverarbeitung der Daten liegen nun nur

noch Datensätze mit erfolgreicher Gesichts-, Augenlandmarken- und Blickrichtungserkennung vor. Zur Bestimmung der Blicke werden neben den Richtungsvektoren auch die Zentren der Pupillen benötigt. Dazu werden jeweils die Mittelwerte der X-, Y- und Z-Dimension der Pupillenlandmarken berechnet, für das linke Auge also die Mittelwerte der Augenlandmarken 48-55 und für das rechte Auge die Mittelwerte der Augenlandmarken 20-27 (siehe Abbildung 1). Die Abbildung 2 zeigt das Pupillenzentrum des linken Auges als grünes Kreuz und die Richtung des dazugehörigen Blickes als rote Linie für die gefilterten Datensätze der Datei *Mi 14-57* <sup>5</sup>.

Die Koordinaten der visualisierten Daten sind im Kamerakoordinatensystem angegeben. Entsprechend ist zu erkennen, dass der Großteil der erkannten Blicke in negative Z-Richtung (zur Kamera), in negative X-Richtung (aus Sicht der Kamera nach links) und in negative Y-Richtung (aus Sicht der Kamera nach unten) gehen.

Für eine Auswertung der Blickdaten im Kontext der Küche, müssen die Kamerakoordinaten in ein durch die Küche definiertes Koordinatensystem transformiert werden. Als Ausgangspunkt für das Küchenkoordinatensystem liegt ein 3D-Modell der Küche vor. Für die X-Y-, X-Z- und Z-Y Ebenen des Modells werden Bilder erstellt, so dass die X-, Y- und Z-Dimension in jedem Bild jeweils der gleichen Pixeldimensionen entspricht (siehe Abbildung 3). Die Y-Achse repräsentiert die Höhe, die X-Achse die Breite (entlang der Schrankrückwand) und die Z-Achse die Tiefe der Küche. Der Ursprung ist auf jedem Teilbild mit einem roten Rechteck markiert. Im Kontext der realen Küche ist der Ursprung, aus der frontalen Sicht auf die Küchenzeile hin, in der linken unteren Ecke.

---

<sup>5</sup> 1\_Daten/2\_processed/2\_wednesday/webcam\_2019-02-27-14-57.csv

Tabelle 2: Kennzahlen der Rohdaten des Eyetrackingdatensatzes

<i>Datei</i>	<b>Frames</b>	<b>Dauer (s)</b>	<b>FPS</b>	<b>Gesicht (Frames)</b>	<b>Gesicht (Anteil)</b>	<b>Blicke (Frames)</b>	<b>Blicke (Anteil)</b>	<b>Success (Anteil)</b>	<b>Konfidenz&gt;90% (Anteil)</b>
<i>Mo 21-05</i>	5344	3559,00	1,50	5344	100,00%	183	3,42%	3,42%	1,52%
<i>Mo 23-00</i>	11264	4093,893	2,75	11264	100,00%	193	1,71%	1,71%	0,74%
<i>Di 11-00</i>	21546	14153,174	1,52	21542	99,98%	948	4,40%	4,40%	2,29%
<i>Di 15-00</i>	2199	1578,254	1,39	2199	100,00%	462	21,01%	21,01%	11,01%
<i>Mi 11-10</i>	10542	4661,934	2,26	10541	99,99%	405	3,84%	3,84%	2,36%
<i>Mi 13-00</i>	1918	238,706	8,03	1918	100,00%	9	0,47%	0,47%	0,21%
<i>Mi 13-06</i>	2095	2957,093	0,71	2094	99,95%	426	20,33%	20,33%	10,50%
<i>Mi 14-57</i>	10419	4383,042	2,38	10419	100,00%	434	4,17%	4,17%	1,52%
<i>Do 11-11</i>	7360	5974,273	1,23	7360	100,00%	772	10,49%	10,49%	6,98%
<i>Do 13-16</i>	38322	11268,384	3,40	38322	100,00%	1281	3,34%	3,34%	1,85%
<i>Fr 11-00</i>	1968	221,135	8,90	1968	100,00%	85	4,32%	4,32%	0,20%
<i>Fr 11-08</i>	4273	4249,251	1,01	4273	100,00%	407	9,52%	9,52%	2,71%
<i>Fr 13-02</i>	4922	4682,726	1,05	4922	100,00%	758	15,40%	15,40%	9,18%

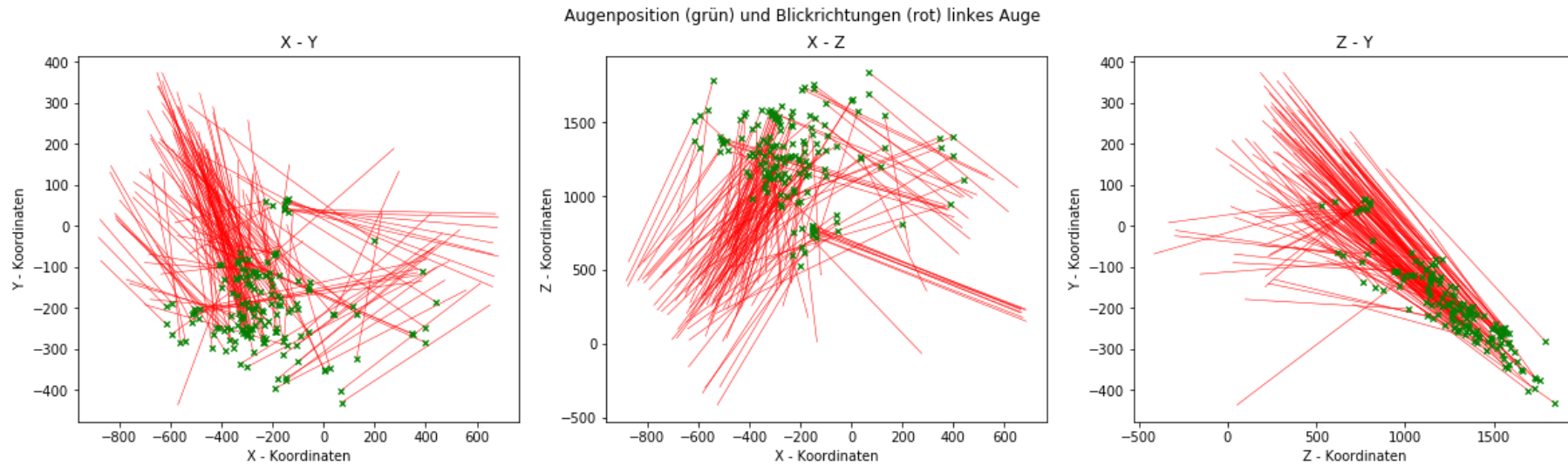


Abbildung 2: Plot der Augenposition und Blickrichtung



Abbildung 3: Bilder des 3D-Modells der Küche



### 4.3 Koordinatentransformationen

Damit die gesammelten Blickdaten mit Hilfe des 3D-Modells visualisiert und ausgewertet werden können, werden die Daten in das Koordinatensystem des Modells transformiert. Für diese Transformation der Blickdaten in das Zielkoordinatensystem des 3D-Modells, wird eine Transformationsmatrix benötigt. Die dreidimensionalen Koordinatensysteme und die entsprechenden Elemente werden wie folgt definiert:

$K_K$ :

Kamerakoordinatensystem

$$\mathbf{v}_K = \begin{pmatrix} x_K \\ y_K \\ z_K \end{pmatrix} \in K_K:$$

Repräsentation des dreidimensionalen Punktes  $v$  im Kamerakoordinatensystem  $K_K$  mit X-, Y- und Z-Koordinaten  $x_K$ ,  $y_K$  und  $z_K$

$K_P$ :

Durch Pixeldimensionen definiertes Koordinatensystem des 3D-Modells

$$\mathbf{v}_P = \begin{pmatrix} x_P \\ y_P \\ z_P \end{pmatrix} \in K_P:$$

Repräsentation des dreidimensionalen Punktes  $v$  im Koordinatensystem des 3D-Modells  $K_P$  mit X-, Y- und Z-Koordinaten  $x_P$ ,  $y_P$  und  $z_P$

Wie in Abschnitt 104.1 beschrieben, wird angenommen, dass es sich bei dieser Abbildung um eine affine Abbildung handelt. Die Transformationsmatrix kann somit als Abfolge von elementaren Transformationsmatrizen beschrieben werden. Diese elementaren Transformationen setzen sich aus Translationen, Rotationen und Skalierungen in den X-, Y- und Z-Dimensionen zusammen. Durch den Einsatz von homogenen Koordinaten wird sichergestellt, dass die Transformationsmatrix als Produkt von elementaren Transformationsmatrizen geschrieben werden kann (Nischwitz, 2011). Die Koordinatensysteme und ihre Elemente werden dementsprechend dann wie folgt repräsentiert:

$K_K^H$ :

Kamerakoordinatensystem mit homogenen Koordinaten

$$v_K^H = \begin{pmatrix} x_K \\ y_K \\ z_K \\ 1 \end{pmatrix} \in K_K^H:$$

Repräsentation des dreidimensionalen Punktes  $v$  im Kamerakoordinatensystem mit homogenen Koordinaten  $K_K^H$  mit X-, Y- und Z-Koordinaten  $x_K$ ,  $y_K$  und  $z_K$

 $K_P^H$ :

Koordinatensystem des 3D-Modells mit homogenen Koordinaten

$$v_P^H = \begin{pmatrix} x_P \\ y_P \\ z_P \\ 1 \end{pmatrix} \in K_P^H:$$

Repräsentation des dreidimensionalen Punktes  $v$  im Koordinatensystem des 3D-Modells mit homogenen Koordinaten  $K_P^H$  mit X-, Y- und Z-Koordinaten  $x_P$ ,  $y_P$  und  $z_P$

Die Verwendung der homogenen Koordinaten impliziert die nachfolgende triviale Abbildung und ihre Umkehrabbildung:

$$g: K \rightarrow K^H, \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad g^{-1}: K^H \rightarrow K, \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Für die Transformationsmatrix  $M$  und alle dreidimensionalen Punkte  $v$  soll dann gelten:

$$M \cdot v_K^H = v_P^H$$

$$M \cdot \begin{pmatrix} x_K \\ y_K \\ z_K \\ 1 \end{pmatrix} = \begin{pmatrix} x_P \\ y_P \\ z_P \\ 1 \end{pmatrix}$$

Die elementaren Transformationsmatrizen für dreidimensionale Koordinatensysteme mit homogenen Koordinaten, bestehend aus Translationen ( $T_{x,y,z}$ ), Rotationen ( $R_\alpha^x$ ,  $R_\alpha^y$  und  $R_\alpha^z$ ) und Skalierungen ( $S_{x,y,z}$ ), werden im Folgenden definiert (Nischwitz, 2011):

$$T_{x,y,z} = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_\alpha^x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_\alpha^y = \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_\alpha^z = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$S_{x,y,z} = \begin{pmatrix} x & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Translationen und Skalierungen werden dabei wegen ihrer Beschaffenheit in einer Matrix zusammengefasst. So kann beispielsweise eine Translation entlang der X-Achse um den Wert 25 und entlang der Y-Achse um den Wert 15 als  $T_{0,15,0} \cdot T_{25,0,0} = T_{25,15,0}$  beschrieben werden. Bei der Aneinanderreihung von Transformationsmatrizen muss, wegen der Nichtkommutativität der Matrizenmultiplikation, im Allgemeinen die Reihenfolge beachtet werden. So ergibt z.B. eine Skalierung aller Achsen um den Wert 20 und eine anschließende Translation entlang der X-Achse um den Wert 15 eine andere Transformationsmatrix als umgekehrt ( $T_{15,0,0} \cdot S_{20,20,20} \neq S_{20,20,20} \cdot T_{15,0,0}$ ). Aus der oben beschriebenen Gleichung  $M \cdot v_K^H = v_P^H$  und der Assoziativität der Matrizenmultiplikation ergibt sich für die als Produkt aus elementaren Transformationsmatrizen be-

schriebene Matrix  $M$ , dass die entsprechenden Faktoren von rechts nach links angewendet werden. Ist die Matrix  $M$  wie beim vorherigen Beispiel durch eine Skalierung gefolgt von einer Translation definiert so gilt dann:

$$T_{15,0,0} \cdot (S_{20,20,20} \cdot v_K^H) = (T_{15,0,0} \cdot S_{20,20,20}) \cdot v_K^H = M \cdot v_K^H$$

Da die Transformationsmatrix alle dreidimensionalen Punkte vom Kamerakoordinatensystem in das Koordinatensystem des 3D-Modells transformiert, gilt dies insbesondere auch für den Ursprung des Kamerakoordinatensystems, also die Position der Kamera. Anhand der Kameraposition und -rotation können somit die benötigten Werte für die Rotations- und Translationsmatrizen ermittelt werden. Dadurch dass die tatsächlichen Rotationswinkel und die Verschiebungen im Kontext der realen Küche für die Versuche nicht dokumentiert wurden, müssen diese geschätzt werden. Da die Kamera in allen Versuchen an der linken Wand, in Richtung Kühlschrank gerichtet, angebracht war, werden für die Ermittlung der Transformationsmatrix folgende Schritte durchgeführt:

1. Spiegelung der Y-Achse
2. Rotation der Kamera
3. Skalierung der Koordinaten
4. Translation der Kamera
5. Z- und X-Achse vertauschen

Das von OpenFace 2.0 definierte Kamerakoordinatensystem gibt die Y-Achse von oben (positiv) nach unten (negativ) an. Die Y-Achse im Koordinatensystem des 3D-Modells verläuft von unten (negativ) nach oben (positiv). Daher wird im Schritt 1 diese Orientierung umgekehrt. Dies entspricht einer Skalierung der Y-Achse um den Faktor -1, also der Transformationsmatrix  $S_{1,-1,1}$ . In den Schritten 2-4 wird die Rotation, Verschiebung und Skalierung der Kamera ausgeglichen (siehe Abbildung 4 und Abbildung 5). Die Werte zur Bestimmung der entsprechenden Transformationsmatrizen werden durch Schätzungen anhand der Punkteverteilung der jeweiligen Datei ermittelt (vgl. Abbildung 6 und Abbildung 7). In Schritt 2 werden die Rotationen der Kame-

ra um die X-, Y- und Z-Achse ( $R_\gamma^x$ ,  $R_\beta^y$  und  $R_\alpha^z$ ) so ausgeglichen, dass die Punktemenge hinsichtlich ihrer Verteilung realistisch abgebildet wird. Zu beachten ist, dass nach dem Rotationsausgleich die Z- und X-Achsen der Kamera gegenüber dem 3D-Modell noch vertauscht sind. In Schritt 3 wird die Achsenskalierung ausgeglichen. Es wird angenommen, dass der Faktor der Skalierung für alle Achsen gleich ist. Da für alle Versuche die gleiche Kamera verwendet wurde, wird außerdem angenommen, dass dieser Wert für alle Datensätze gleich ist. Der Wert wird so gewählt, dass die Verteilung der Punktemenge realistisch an die Pixeldimension angepasst ist ( $S_{f,f,f}$ ). In Schritt 4 wird die Verschiebung der Kamera bestimmt ( $T_{dx,dy,dz}$ ), wobei die Z- und X-Achsen der Kamera gegenüber dem 3D-Modell immer noch vertauscht sind. Abschließen wird in Schritt 5 die Vertauschung der X- und Z-Achsen mit einer Rotation um die Y-Achse um  $-90^\circ$  und eine anschließende Skalierung der X-Achse mit dem Faktor -1 ausgeglichen ( $S_{-1,1,1} \cdot R_{-90^\circ}^y$ ). Die Transformationsmatrix  $M$  kann also folgendermaßen beschrieben werden:

$$M = S_{-1,1,1} \cdot R_{-90^\circ}^y \cdot T_{dx,dy,dz} \cdot S_{f,f,f} \cdot R_\alpha^z \cdot R_\beta^y \cdot R_\gamma^x \cdot S_{1,-1,1}$$

Die Schritte zur Ermittlung von  $M$  und insbesondere die Schritte 2-4 zur Ermittlung der Werte der Variablen  $dx$ ,  $dy$ ,  $dz$ ,  $f$ ,  $\alpha$ ,  $\beta$  und  $\gamma$  werden für jeden gefilterten Datensatz durchgeführt. Anhand dieser Werte können alle im Kamerakoordinatensystem angegebenen Positionsdaten jetzt in das 3D-Modell bzw. in das durch die Pixeldimensionen der Teilbilder definierte Koordinatensystem des 3D-Modells überführt werden.

Sind die Werte der Variablen, also insbesondere die Rotation der Kamera um die eigene X- ( $\gamma$ ), Y- ( $\beta$ ) und Z-Achse ( $\alpha$ ), sowie die Verschiebung der Kamera im Kontext des 3D-Modells in X- ( $dz$ ), Y- ( $dy$ ) und Z-Richtung ( $dx$ ) bekannt bzw. dokumentiert, so können diese Werte direkt in die oben beschriebene Formel eingefügt werden.

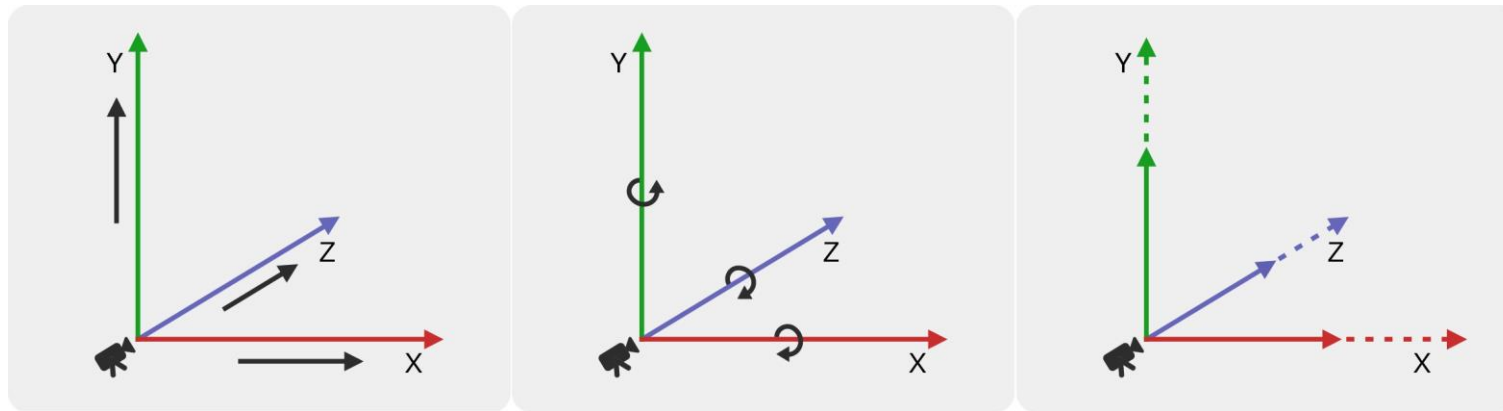


Abbildung 4: Verschiebung, Rotation und Skalierung (v.l.n.r.) der Kamera

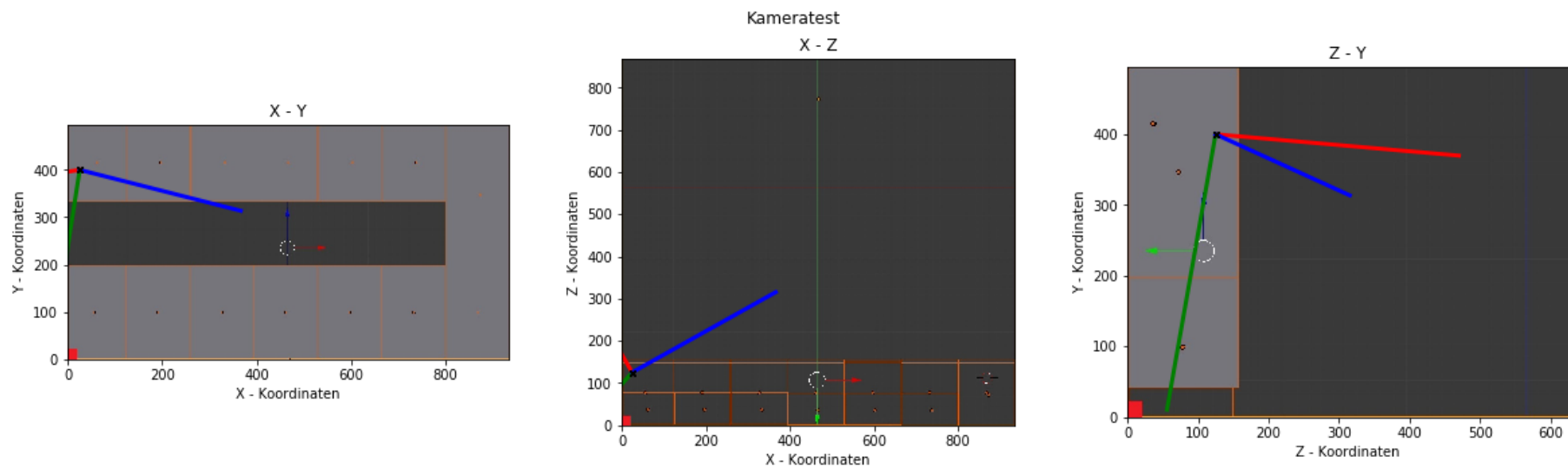


Abbildung 5: X-Achse (rot), Y-Achse (grün) und Z-Achse (blau) der Kamera nach einer Transformation

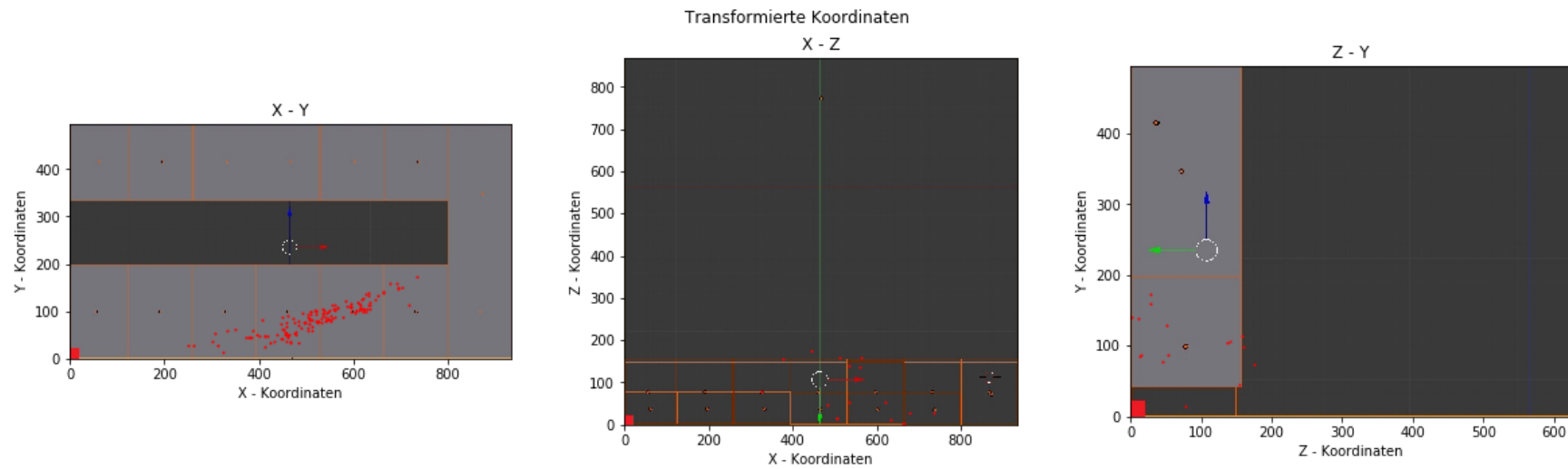


Abbildung 6: Augenpositionen im Koordinatensystem des 3D-Modells ohne Rotation und Translation

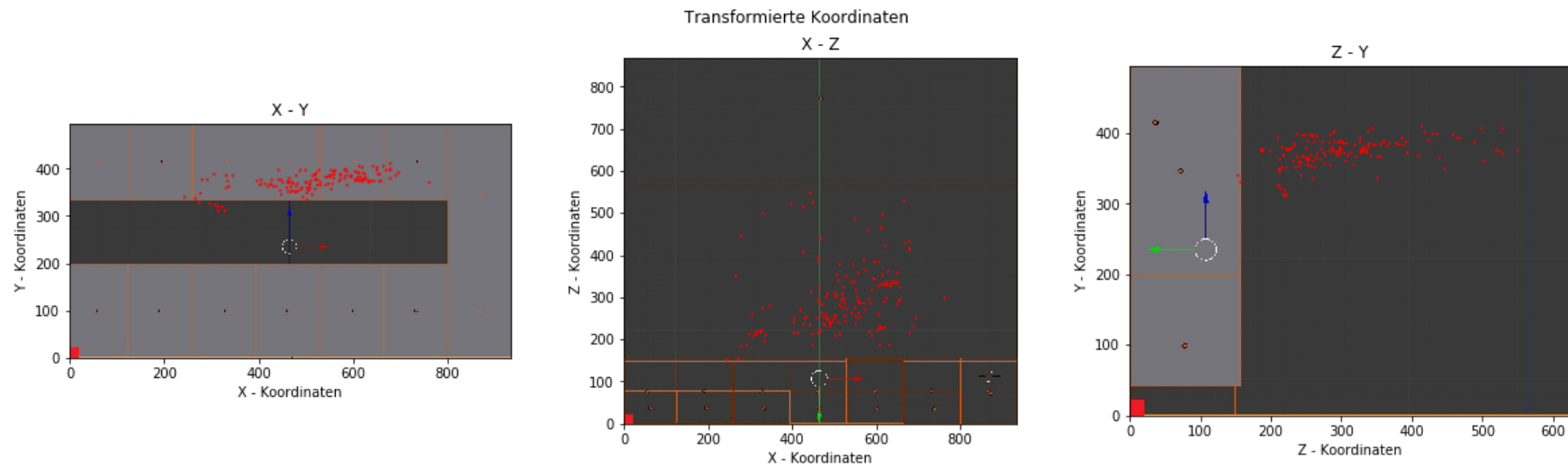


Abbildung 7: Augenpositionen im Koordinatensystem des 3D-Modells nach Rotation und Translation

#### 4.4 Auswertung der Blickdaten

Angelehnt an die Filterung der Rohdaten und die Bestimmung der jeweiligen Transformationsmatrizen werden die relevanten Eyetrackingdaten in diesem Abschnitt ausgewertet. Dazu werden zunächst die Positionsdaten der Pupillenzentren und die dazugehörigen Blickrichtungen in das Koordinatensystem des 3D-Modells transformiert. Für die Transformation eines Pupillenzentrums  $z_K$ , wird dieser dreidimensionale Punkt mit der entsprechenden Transformationsmatrix  $M$  multipliziert ( $z_P^H = M \cdot z_K^H$ ). Die Blickrichtungen  $b_K$  werden mit dem Hilfspunkt  $e_K = z_K + b_K$  transformiert. Wie das Pupillenzentrum, wird dieser Hilfspunkt mittels Multiplikation mit der Transformationsmatrix  $M$  transformiert ( $e_P^H = M \cdot e_K^H$ ). Anschließend wird die Blickrichtung im Koordinatensystem des Modells durch Subtrahieren des Vektors  $z_P$  vom Vektor  $e_P$ , gefolgt von einer Normierung bestimmt ( $b_P = \frac{e_P - z_P}{|e_P - z_P|}$ ).

Ausgehend von den transformierten Augenpositionen und Blickrichtungen, können die Endpunkte der Blicke bestimmt werden. Dazu werden für das 3D-Modell der Küche bestimmte AOIs festgelegt. Diese AOIs werden hierbei als dreidimensionale Rechtecke, also insbesondere als durch vier Eckpunkte bestimmte und endliche Teilflächen von Ebenen, definiert. Für die erste Auswertung werden zunächst die gesamte Arbeitsplatte (grün), die oberen (orange) und unteren Schränke (weiß), die Wand hinten zwischen den Schränken (blau) und der Kühlschrank (türkis) als AOIs verwendet (siehe Abbildung 8).

Zur Auswertung der Blicke werden für alle festgelegten AOIs die Schnittpunkte der durch die Augenposition und Blickrichtung definierten Geraden, mit der durch die Eckpunkte der AOI definierten Ebene berechnet. Für jedes Paar aus Augenposition und Blickrichtung wird dann derjenige Schnittpunkt als Endpunkt festgehalten, der im Vergleich zu den anderen Schnittpunkten die geringste Distanz aufweist. Es wird also der erste Schnittpunkt mit einer AOI gewertet. Die Blicke eines Datensatzes werden dann mit Hilfe von Heatmaps für jede AOI dargestellt (vgl. Abbildung 9).



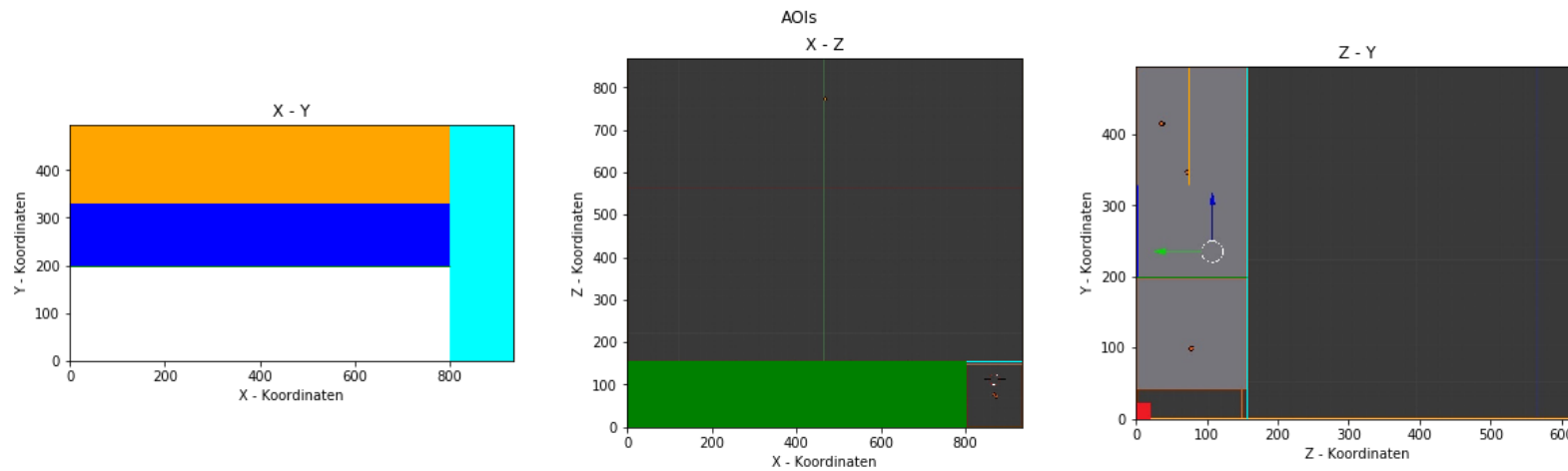


Abbildung 8: Definierte AOIs

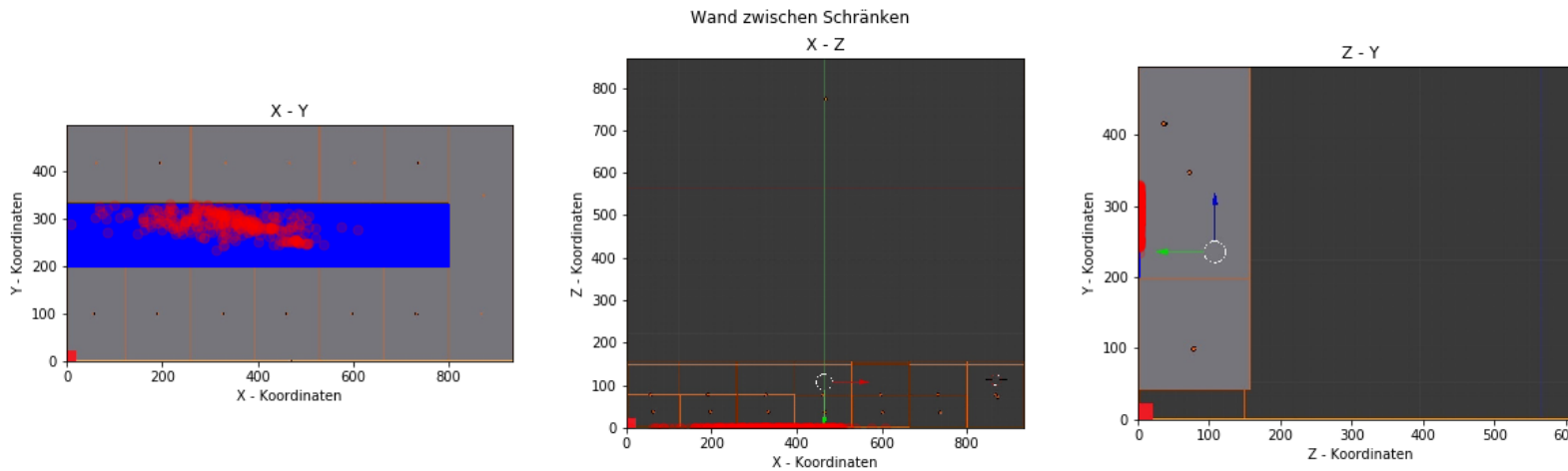


Abbildung 9: Heatmap für AOI „Wand zwischen Schränken“ (Fr 13-02)

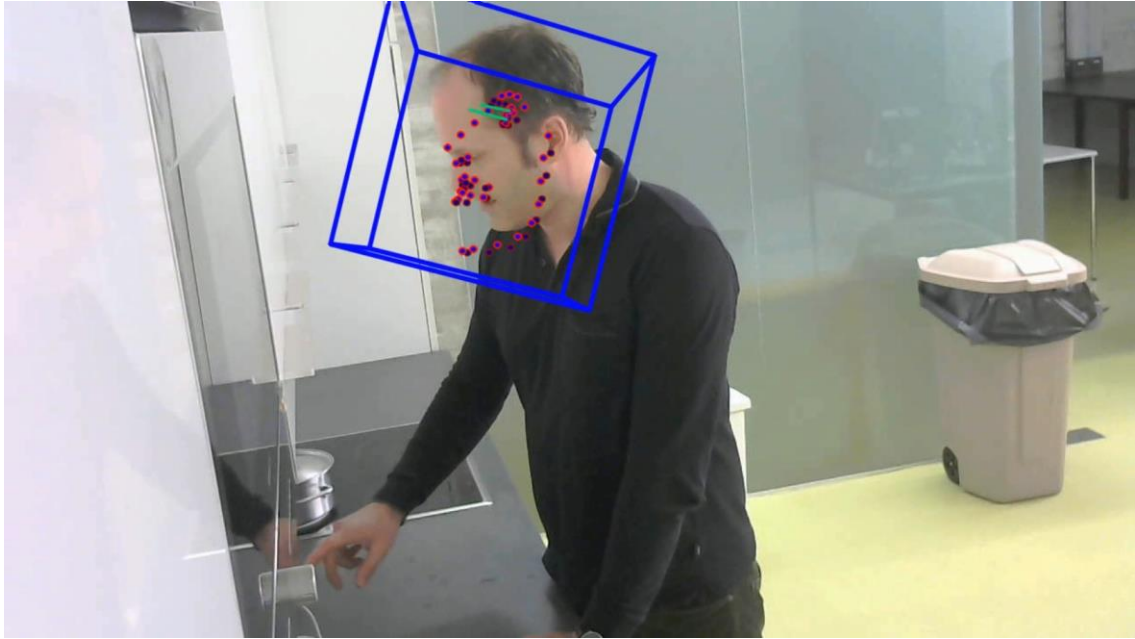
## 4.5 Ergebnisse und Einschränkungen

Im nächsten Schritt der Auswertung müssten die Blicke in eine zeitliche Verbindung mit den anderen vorliegenden Daten des Experiments gebracht werden. So könnte dann abgeleitet werden, zu welchem Rezeptschritt bzw. welcher Äußerung aus dem think aloud Protokoll die Probanden welchen Bereich fixieren. Wegen der schlechten Qualität der Eyetrackingdaten und weil die relativen Timestamps der Eyetrackingdaten nicht mit den Timestamps der anderen Daten synchronisiert werden können, werden diese Auswertungen in dieser Seminararbeit nicht durchgeführt.

Im Rahmen der Auswertung der Eyetrackingdaten wurden folgende Probleme bei der Datenqualität identifiziert:

- geringe FPS bei erfolgreicher Erkennung
- schlechte Erkennungsrate
- relative Timestamps
- geschätzte Werte für Transformationsmatrix

Die geringe FPS Zahl kann, wie bereits in Abschnitt 4.2 beschrieben, durch die Rechenleistung des verwendeten Rechners begründet werden. Dieser Faktor wirkt sich auf die Genauigkeit bei der Bestimmung der Blicke in der Küche aus, da durch das Eyetracking mit weniger als einem Frame pro Sekunde möglicherweise relevante Fixationen nicht analysiert werden. An dieses Problem könnte z.B. mit der Verwendung eines leistungsfähigeren Rechners oder mit der Aufzeichnung und nachträglichen Analyse eines Videos herangegangen werden. Die schlechte Erkennungsrate wirkt sich stark auf die Qualität der Daten aus, da bei diesem Experiment im besten Fall nur ca. 20% der Frames erfolgreich analysiert wurden. Dies bedeutet, dass für jede Aufnahme mindestens ca. 80% der aufgezeichneten Daten unbrauchbar sind. Beim Heranziehen der zusätzlichen Videoaufnahme der zweiten Kamera und nachträglicher Analyse mit OpenFace 2.0 hat sich herausgestellt, dass die Position links oben in der Küche schlecht gewählt war. Bei dem dadurch entstehenden Winkel der Kamera, hat die Software Probleme, die Gesichts und Blickerkennung erfolgreich durchzuführen. Die folgende Abbildung zeigt einen Frame mit den von OpenFace 2.0 berechneten Informationen:



**Abbildung 10: Gesichtslandmarken und Blickrichtungen von OpenFace 2.0**

Dieses Problem der schlechten Erkennungsrate könnte möglicherweise durch die Optimierung der Kameraposition und -rotation bzw. durch die Verwendung mehrerer Kameras behoben werden. Dadurch, dass im OpenFace 2.0 Format die Timestamps nur in relativer Angabe zum Startzeitpunkt der Aufnahme vorliegen, können diese Daten nur schwer in Relation zu den Tasks und anderen Daten gebracht werden. Dies hat Einfluss auf die Datenqualität, da somit keine Aussagen über Zusammenhänge zwischen Blicken und Tasks getroffen werden können. Durch eine zusätzliche Verwendung von absoluten Timestamps in allen aufgezeichneten Daten wäre dieses Problem behebbar. Weil in allen Versuchen die Position und Rotation der Kamera nicht dokumentiert wurden, mussten die entsprechenden Werte, wie in Abschnitt 4.3 beschrieben, jeweils anhand der Positionsdaten geschätzt werden. Dies beeinflusst die Datenqualität, da nicht sichergestellt ist, dass diese geschätzten Werte die realen Werte repräsentieren. Mit dem Dokumentieren der Kamerarotation und -verschiebung z.B. durch ein Foto oder ein festes Stativ, könnte man auch dieses Problem verhindern. Eine andere Herangehensweise an die Bestimmung der Transformationsmatrix wäre die Verwendung von Fixpunkten. Mit diesen Fixpunkten könnte eine Art Kalibrierung durchgeführt werden, wodurch ein Gleichungssystem zur Bestimmung der Transformationsmatrix gelöst wird.

Als Ergebnis der Auswertung liegen für alle relevanten Datensätze Heatmaps für jede definierte AOI vor. Durch die schlechte Datenqualität sind diese Heatmaps jedoch nicht weiter auswertbar, da zum einen kein Bezug zu den anderen Daten hergestellt werden kann und zum anderen keine für den Untersuchungsgegenstand interessanten Blicke getrackt worden sind. Als Ergebnis der Seminararbeit kann festgehalten werden, dass eine Methode zur Umrechnung von Eyetrackingdaten in den Kontext eines 3D-Modells entwickelt wurde. Außerdem konnten bei der Erarbeitung und Durchführung dieser Methode vermeidbare Probleme identifiziert werden, die sich im negativen Sinne auf die Datenqualität auswirken. Es wurde versucht, die Ursachen der Probleme zu ermitteln und es wurden Lösungsansätze empfohlen.

## 5 Ausblick

Das Hauptziel der vorliegenden Arbeit war es eigentlich, eine Eyetracking Anwendung als Komponente eines Küchenassistenzsystems umzusetzen und dabei abhängig vom Blickverhalten des Nutzers einen Versuch der Kontaktaufnahme des Nutzers mit dem System zu erkennen und zu verarbeiten. Durch Veränderungen in den Rahmenbedingungen, musste diese Idee jedoch verworfen bzw. abgewandelt werden. Stattdessen war das neue Ziel die Umrechnung der Blickdaten von den Koordinaten der Kamera in das Koordinatensystem des 3D-Modells, da entgegen der Annahme diesbezüglich noch keine Lösung vorhanden war. Somit wurde die Erstellung eines Scripts für diese Koordinatentransformation für das Erreichen der angedachten Untersuchung notwendig. Schlussendlich wurde durch die Transformation für jeden getrackten Blick ein Schnittpunkt mit dem 3D-Modell der Küche errechnet. Die Heatmaps bzw. Blickpunkte sind jedoch relativ, da kein Zusammenhang mit den anderen Daten (Rezeptschritte, think aloud, Appinteraktionen) hergestellt werden kann. Des Weiteren sind die Werte zur Bestimmung der Transformationsmatrix anhand der Punkteverteilung geschätzt. Somit ist die Korrektheit der Werte nicht sichergestellt.

Aus diesem Grund wäre es für zukünftige Kurse, die gleiche bzw. ähnliche Experimente durchführen, von Vorteil, zusätzlich zu den relativen auch absolute Timestamps, die sowohl die exakte Uhrzeit als auch das Datum beinhalten, zu loggen, um den Experimenten einen eindeutigen Zeitpunkt zuordnen zu können. Durch einen jeweiligen präzisen Start- bzw. Endpunkt der Aufnahme kann das Blickverhalten der Probanden besser mit den wirklichen Rezeptschritten verknüpft und analysiert werden.

Ein weiterer verbesserungswürdiger Aspekt für die Zukunft sind die Kamerabedingungen, die zu den gewünschten Eyetrackingdaten führen sollen. Da in unserem Experiment eine einzelne Kamera aus der Sicht des Probanden links oben befestigt wurde, konnten nur wenige Blickvektoren aufgezeichnet werden. Dies führte aufgrund dieser schlechten Datengrundlage zu keinen nennenswerten Ergebnissen. Für neue Forschungsprojekte wäre es deshalb wichtig, neue Ansatzpunkte bezüglich der Kameraanzahl und der Kameraposition in Betracht zu ziehen. Eine zusätzliche zweite Kamera oder eine zentrierte Positionierung der Kamera könnte hierbei schon eine Verbesserung der Logdaten zur Folge haben. Wichtig dabei ist, dass die Kamerapositionen für jedes Experiment gleich sind, um eventuelle Verfälschungen der Daten vorzubeugen. Da bei unserem Experiment jedoch nicht immer dieselben Versuchsleiter anwesend waren, könnte mit Hilfe einer genauen Dokumentation, z.B. in Form von Fotos, die Kameraverschiebung bzw. -rotationen festgehalten werden, um sicherzustellen, dass beim nächsten Experiment gleiche Bedingungen herrschen können. Auch eine Weitwinkelkamera, frontal vor der Testperson positioniert, wäre eventuell eine Möglichkeit, um mehr Blicke mit ihren jeweiligen Richtungen sichern zu können.

## Literaturverzeichnis

- Baltrusaitis, T., Zadeh, A., Lim, Y.C., & Morency L. (2018). *OpenFace 2.0: Facial Behavior Analysis Toolkit*. 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018). <https://doi.org/10.1109/FG.2018.00019>
- Brush, A. J. B., Lee, B., Mahajan, R., Agarwal, S., Saroiu, S., & Dixon, C. (2011). *Home automation in the wild: challenges and opportunities*. Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI '11, 2115. <https://doi.org/10.1145/1978942.1979249>
- Gräuel, J., & Speilerberg, A. (2007). *Akzeptanz neuer Wohntechniken für ein selbstständiges Leben im Alter – Zeitschrift für Sozialreform*, 53(2). <https://doi.org/10.1515/zsr-2007-0205>
- Lübbecke, T. (2016). *Akzeptanz von Smart Homes–von der Theorie zur Vermarktung*. WISSENSCHAFT TRIFFT PRAXIS, 21.
- Nehmer, J., Becker, M., Karshmer, A., & Lamm, R. (2006). *Living assistance systems: an ambient intelligence approach*. Proceeding of the 28th International Conference on Software Engineering - ICSE '06, 43. <https://doi.org/10.1145/1134285.1134293>
- Neßelrath, R., Lu, C., Schulz, C. H., Frey, J., & Alexandersson, J. (2011). *A Gesture Based System for Context – Sensitive Interaction with Smart Homes*. In R. Wichert & B. Eberhardt (Hrsg.), *Ambient Assisted Living* (S. 209–219). [https://doi.org/10.1007/978-3-642-18167-2\\_15](https://doi.org/10.1007/978-3-642-18167-2_15)
- Nischwitz, A. (2011). *Koordinatensysteme und Transformationen In Computergrafik und Bildverarbeitung (Computergrafik)*. Wiesbaden: Vieweg Teubner.
- Wandke, H. (2005). *Assistance in human–machine interaction: a conceptual framework and a proposal for a taxonomy*. *Theoretical Issues in Ergonomics Science*, 6(2), 129–155. <https://doi.org/10.1080/1463922042000295669>

## **Inhalt des beigefügten Verzeichnisses**

/1_Daten/1_raw	Die Eyetrackingdaten im OpenFace 2.0 CSV Format sortiert nach Tag
/1_Daten/2_processed	Die vorverarbeiteten und gefilterten Daten inkl. Kameratransformationswerten
/2_Notebooks	Quellcode in Form von Jupyter Notebooks zu den Berechnungen in den Abschnitten des Kapitels 4
/3_Lib	Blender 3D-Modell der Küche und Python Datei mit Hilfsfunktionen
/4_Bilder/1_Modell	Bilder der Achsenansichten des 3D-Modells
/4_Bilder/2_AOIs	Ergebnisse der AOI-Auswertung (Heatmaps)
/5_Entwicklung	Die Notebooks und Daten, die während der Bearbeitung verwendet bzw. erstellt wurden
/6_Ausarbeitung/1_Seminararbeit	Seminararbeit als PDF und Worddatei
/6_Ausarbeitung/2_Bilder	Alle in der Seminararbeit verwendeten Abbildungen