

# Inhaltsverzeichnis

<b>1 Einführung</b>	<b>1</b>
1.1 Abstarct . . . . .	1
1.2 Intension . . . . .	1
1.3 Problemstellung . . . . .	2
<b>2 Grundlagen</b>	<b>4</b>
2.1 Hardware . . . . .	4
2.2 Software . . . . .	4
2.3 Das Klassenzimmer - Umgebung des Eye-Tracking . . . . .	4
2.4 Grundlegende Verfahren . . . . .	5
2.4.1 Künstliches neuronales Netz . . . . .	5
2.4.2 Convolutional Neural Network (CNN) . . . . .	5
2.4.3 Active Appearance Model (AAM) . . . . .	6
2.4.4 Constrained Local Model (CLM) . . . . .	6
2.4.5 Constrained Local Neural Fields (CLNF) . . . . .	7
2.4.6 Non-maximum suppression (NMS) . . . . .	7
2.4.7 Patch Experts . . . . .	7
2.4.8 Point Distribution Model (PDM) & Generalized Adaptive View-based Appearance Model (GAVAM) . . . . .	7
<b>3 Umsetzung</b>	<b>8</b>
3.1 Ablauf der Implementierung . . . . .	8
3.2 MTCNN Face Detection . . . . .	9
3.2.1 Die 3 Stufen der Verarbeitung . . . . .	10
3.2.2 Qualität . . . . .	11
3.3 Skalieren von Bildern . . . . .	11
3.3.1 Bicubic-Skalierung . . . . .	12
3.3.2 Lanczos-Skalierung . . . . .	12
3.3.3 Linear-Skalierung . . . . .	12
3.3.4 Nearest-Neighbor-Skalierung . . . . .	13
3.3.5 Qualität der Skalierung . . . . .	13
3.3.6 Ergebnisse Angeben . . . . .	14
3.4 OpenFace . . . . .	19
3.4.1 Bestimmung der Landmarks . . . . .	19
3.4.2 Veröffentlichte Genauigkeit . . . . .	20
3.4.3 Auswirkung der Größe . . . . .	21
3.4.4 Auswirkung der verschiedenen Skalierungsverfahren auf Detektion . . . . .	21
3.4.5 Auswirkung von Pixelrauschen auf Detektion . . . . .	22
3.4.6 Arbeitsbereich bezüglich Rotation . . . . .	22
3.5 Umwandlung von Farbbild nach Graubild . . . . .	24
3.5.1 Gleam-Verfahren . . . . .	24

3.5.2	Gleam-New-Verfahren . . . . .	24
3.5.3	Luminance-Verfahren . . . . .	25
3.5.4	Min-Max-Verfahren . . . . .	25
3.5.5	Quadrat-Verfahren . . . . .	25
3.5.6	Normalisierung von Graubilder . . . . .	25
3.6	ElSe . . . . .	28
3.6.1	Aufbereitung der Bildinformation in der Augenregion . . . . .	28
3.6.2	Beschreibung . . . . .	28
3.6.3	Versuchsaufbau für die Auswirkung der Graubild-Verfahren auf ElSe . . . . .	30
3.6.4	Auswirkung des Radius . . . . .	30
3.6.5	Auswirkung der verschiedenen Graubild-Verfahren . . . . .	30
3.6.6	Vergleich zu OpenFace . . . . .	31
3.6.7	Ergebnis . . . . .	35
3.7	Bestimmung des Ziels der Aufmerksamkeit . . . . .	35
3.7.1	Bestimmung der Position & Orientierung des Gesichts . . . . .	35
3.7.2	Größe und Genauigkeit . . . . .	37
3.7.3	Bestimmung eines Punktes, auf der die Aufmerksamkeit liegt . . . . .	39
3.8	Vorversuche . . . . .	41
3.8.1	Arbeitsbereich der Verfahren - Versuch 1 . . . . .	41
3.8.2	Arbeitsbereich der Verfahren - Versuch 2 . . . . .	43
3.8.3	Auswertung der Augenpartie - Versuch 3 . . . . .	43
3.8.4	Ergebnis der Vorversuche . . . . .	44
3.9	Aufmerksamkeitsmessung - Versuch . . . . .	44
3.9.1	Versuchsdurchführung . . . . .	45
3.9.2	Fehleranalyse des Versuches . . . . .	47
<b>4</b>	<b>Ergebnisse</b>	<b>50</b>
4.1	Fehleranalyse . . . . .	50
4.2	Zusammenfassung . . . . .	50
4.3	Verbesserung . . . . .	50
<b>Literaturverzeichnis</b>		<b>51</b>

# 1 Einführung

## 1.1 Abstarct

Der übliche Aufbau zur Analyse von Gesichter ist, das verwenden von eines Messgerät pro Person und Merkmal, wie Beispielsweise Eye-Tracking Brillen. Soll eine Auswertung auf mehreren Probanden gleichzeitig durchgeführt werden, so ist die Verwendung von weniger Geräten einfacher.

Im Rahmen der Aufmerksamkeitsmessung im Unterricht, soll eine Messung der Aufmerksamkeit einer ganzen Klassen durchgeführt werden. Um Anhaltspunkte eines effizienten Aufbaus des eigentlichen Versuchs zu erhalten, sollen die Grenzen und Qualität bei der Gesichtsanalyse basierend auf Bildmaterial einer einzelnen Kamera aufgezeigt werden. Diese wird fest montiert um eine Frontalaufnahme aller Probanden gleichzeitig zu erhalten, wobei die gesamte Klasse im Fokus der Kamera liegt. Durch diesen Aufbau ergibt sich die Problematik mit sehr unterschiedlichen Distanzen zur Kamera und der dargestellten Größe aller Probanden im Bild.

Um alle Probanden im Frame zu analysieren, werden zuerst die einzelnen Gesichter im Bild gesucht, den Probanden zugeordnet und aufbereitet. Anschließend werden die Gesichter analysiert um ihre Position und Orientierung zu bestimmen. Die Augenregion wird zusätzlich behandelt, um genauere Ergebnisse bei der Bestimmung der Blickrichtung zu erhalten.

Die Versuche haben ergeben, das mit den heutigen HD Kameras eine gleichzeitige Analyse von mehreren Probanden im selben Frame möglich ist, die auf der Fläche eines üblichen Klassenzimmers verteilt sind. Für die Analyse kann meist nur auf den Gesichtern gearbeitet werden, da für die Bestimmung der Blickrichtung zu wenige Informationen in den kleinen Bildern vorhanden sind.

## 1.2 Intension

Die Grundlage für erfolgreiches Lernen ist die Aufmerksamkeit der Schüler und daher ausschlaggebend für die Qualität des Unterrichtes. Das Verhalten kann eingeteilt werden in on-Task (der Schüler ist aufmerksam bei der Sache) und off-Task (der Schüler ist unaufmerksam). Allerdings ist das erfassen der Aufgabe zugewandten Aufmerksamkeit recht schwierig und verschiedene Erfassungsmethoden versuchen dies zu erreichen. Ein Vorschlag von Ehrhardt, Findeisen, Marinello und Reinhartz-Wenzel (1981) umfasst die Parameter Blickrichtung, Körperhaltung und Tätigkeit.

Zur Erfassung werden z.B. Fragebögen eingesetzt, die Schüler und Lehrer selbst ausfüllen oder es gibt ein Beobachter der die Aufmerksamkeit einzelner Schüler bewertet.

Für das „Das Münchener Aufmerksamkeitsinventar (MAI)“[HR92] wird beispielsweise die Kategorien „ON-TASK, reaktiv/fremd-initiiert: der Schüler reagiert auf eine entsprechende Aufforderung oder Frage des Lehrers“oder „OFF-TASK - aktiv, interagierend, störend: Der Schüler nimmt die Lerngelegenheit nicht nur nicht wahr, sondern ist erkennbar anderweitig engagiert“festgelegt. Um das Verhalten eines Schülers zu bewerten wird dieser 5s lange beobachtet und eine Kategorie wird zuzuordnen.

Bei der „Videostudie zur Wirksamkeit des Unterrichtsprozesses “[App15] wurden die Kriterien „Blickkontakt zum legitimen Sprecher oder Objekt, Aktive Beteiligung an der Aufgabe, keine Ausübung anderer Tätigkeiten, keine Motorische Unruhe und keine themenferne Kommunikation“festgelegt. Dann wurde der Schüler in einem ein Minuten-Intervall beobachtet und bewertet. Sind drei oder mehr Punk-

te erfüllt, gilt die Aufmerksamkeit des Schüler als on-Task.

Bei dieser Art der Auswertung gibt es allerdings Interpretationsfreiheiten, die von jedem Beobachter anders ausgelegt werden können. Außerdem ist sie sehr zeitintensiv, alleine eine einzige Beurteilung jedes einzelnen Schülers einer Klasse, etwa 30 Personen nach Vorgabe der Klassenbildung [kla16], benötigt mindestens 30 Minuten. Somit kann eine Auswertung aller Schüler während einer Unterrichtsstunde schnell 15 und mehr Arbeitsstunden dauern. Um eine subjektive Bewertungen zu vermeiden, sollte außerdem ein beträchtlicher Teil der Daten von mindestens zwei Beobachtern parallel ausgewertet werden, um deren Übereinstimmung beurteilen zu können.

Basiert die Auswertung auf wenigen Zeitintervalle um Arbeitszeit zu sparen, wird das gesamte Verhalten eines Schülers während des Unterrichts mit nur wenigen beobachteten Minuten beschrieben und ist entsprechend ungenau. Somit können sowohl quantitativ genaue, als auch temporal hochauflösende Daten nicht erstellt werden.

So kann bei grob gewählten Auswertungsintervallen nur eine Aussage über den gesamten Unterricht gemacht werden und nicht beispielsweise über einzelne Übungen oder über einen einzelnen Schüler.

## 1.3 Problemstellung

Im Rahmen dieser Arbeit sollen die Grenzen aufgezeigt werden, wie weit es technisch möglich ist Filmmaterial einer einzigen Kamera Auszuwerten im Bezug auf Blickrichtungen bzw. Ausrichtung des Gesichts und mit welchen Einschränkungen und Genauigkeiten zu rechnen sind, wenn im Bild eine gesamte Klasse dargestellt ist.

Eine automatisierte Auswertung der Blickrichtung wäre erstrebenswert, da dies einer der wichtigsten Indikatoren für gerichtete Aufmerksamkeit ist, allerdings würde eine Bestimmung der Kopforientierung ausreichen, da sie in etwa der Blickrichtung entspricht.

Die Messung soll den Unterricht möglichst wenig beeinträchtigen, wodurch hierfür üblicherweise verwendete Geräte, wie z.B. Eye-Tracking Brillen, nicht verwendet werden können. Zum einen ist die Anschaffung einer großen Stückzahl dieser Geräte teuer und wurde bisher nur in wenigen speziell eingerichteten Laboratorien durchgeführt wie z.B. TüDiLab [Tue]. Zum anderen sind die Geräte entweder Ablenkend (Brillen) oder schränken den Aktionsradius ein (Remote Tracker).

Die hier bestimmten Grenzen ergeben Anhaltspunkte, wie das Setup (Anzahl und Position der Kameras und deren Auflösung) für ein größeres Experiment aussehen muss, um die Aufmerksamkeit einer ganzen Klasse zu erfassen. Wären man in der Lage, solch eine qualitativ hochwertige Auswertung mit nur wenigen Kamera durchführen zu können, so ist der Aufbau und die Aufnahmen der Daten auch für technische Laien durchführbar.

Eine Möglichkeit für das automatische Erfassen der Aufmerksamkeit wird in „Real time detection of driver attention“[GM14] vorgestellt. Bei diesem Verfahren ist eine Kamera direkt von vorn auf den Fahrer gerichtet und anhand der Kopf und Augenposition bewertet ob dieser aktiv auf den Verkehr achtet.

Ein weiteres dazu passendes Verfahren wird in „AggreGaze“[YS16] präsentiert, dabei wird eine einzige Kamera fest auf einem Bildschirm montiert, um die Blickrichtung der Passanten auf den Bildschirm zu bestimmen, dieses Verfahren arbeitete allerdings nur auf einem recht begrenzen Bereich.

Um die Machbarkeit der Analyse zu untersuchen, wurden verschiedene Videoaufnahmen verwendet. Unter anderem zwei Originalaufnahmen eines Englischunterrichtes aus dem Datensatz ‚To Do: Tue-DI?‘, diese besitzen allerdings nur eine sehr geringe Auflösung ( $640 \times 480$  Pixel) und zeigen die gesamte Klasse aus Richtung der Tafel.

Für die Vorversuche wurde eine Actioncam verwendet um erste Eindrücke zu erhalten, bezüglich der Auswirkung von Position und Zielpunkt der Aufmerksamkeit.

*4. Juli 2017*

Um mehr Messwerte für unterschiedlichen Zielpunkte zu erhalten wurde ein weiterer Video-Datensatz mit der Logitech erstellt, bei der die Probanden ein bewegtes Ziel beobachten sollten.

## 2 Grundlagen

### 2.1 Hardware

Als Messinstrument für die Versuche wurden verschiedenen Farbkameras eingesetzt.

Das Videomaterial der Schulkasse wurde mit einer unbekannten Videokamera aufgezeichnet, daher sind nur die Parameter des Filmes ( $640 \times 480$  Pixel mit  $25\text{Fps}$ ) bekannt.

Für die Messungen im Versuch wurde die Explorer 4K Action Camera verwendet, sie besitzt ein  $170^\circ$  Weitwinkel-Linse mit großer Schärfentiefe. Mit ihrer 2.7K Einstellung wird ein  $2688 \times 1520$  Video mit 30FPS aufgezeichnet. Leider ist das Bild stark von Pixelrauschen betroffen.

Außerdem die Logitech c920 HD Pro Webcam, diese liefert ein  $15\text{Fps}$  Video mit einer Auflösung von  $1600 \times 896$  Pixel. Die Kamera besitzt einen horizontalen Blickwinkel von etwa  $70^\circ$ .

### 2.2 Software

Für die Umsetzung wurden Open Source Computer Vision (OpenCV 3.1) verwendet. Dies ist eine C/C++ Bibliothek von Algorithmen zur Bildverarbeitung in Echtzeit, veröffentlicht unter der BSD Lizenz (Berkeley Software Distribution)

[BK08][Wik17c]

### 2.3 Das Klassenzimmer - Umgebung des Eye-Tracking

Die Anwendung ist für den Unterricht ausgelegt, wie in der Abschnitt 1.3 beschrieben. Ein deutsches Klassenzimmer soll laut Baden-Württembergischen Schulbauempfehlungen eine Grundfläche von  $54 - 66\text{m}^2$  aufweisen für maximal 28-32 Schülern [bau13]. Da noch die Tafel usw. beachtet werden muss, ergibt sich einen Bereich von etwa  $2.5 - 8\text{m}$  auf  $6\text{m}$  in dem sich die Schüler aufhalten werden. Somit muss der Linsenwinkel mindestens  $100^\circ$  betragen mit entsprechender Schärfentiefe, damit alle im Bild sind.

Der Unterricht soll durch die Messung möglichst wenig beeinflusst werden, somit ergeben sich folgende Randbedingungen:

- Brillen, Kontaktlinsen und ähnliches sind bei den Probanden erlaubt, ebenso beliebige Frisuren, Make-up usw.
- Die üblichen Bewegungen im Unterricht wie Sprechen, Kopfdrehungen usw. der Schüler sind gestattet.
- Das Verfahren soll gleichzeitig auf Distanzen von  $2.5 - 8\text{m}$  zur Kamera auf einer Breite von  $6\text{m}$  funktionieren.
- Es werden keine Markierungen (Tacker oder ähnliches) an den Schülern angebracht, noch werden diese exakt ausgemessen.

Außerdem soll die Anwendung auch auf schon vorhanden Aufnehmen eines Unterrichtes arbeiten, die oben genannten Bedingungen erfüllen.

Für die Anwendung werden zusätzlich folgende Annahmen gemacht, die sich vor allem auf die Sitzordnung der Schüler sowie die Umgebung beziehen.

- Die Szene ist innerhalb eines Gebäudes, mit ausreichend gleichmäßiger Beleuchtung.
- Die Kamera befindet sich vor der Klasse, so dass die Hauptblickrichtung der Schüler in Richtung Kamera verläuft.  
Gleichzeitig kann die Kamera jedoch nicht ohne weiteres ganz zentral angebracht werden, da dieser Raum für den Unterricht (Tafel/Lehrer) benötigt wird.
- Die Gesichter sind komplett sichtbar und nicht verdeckt durch andere Schüler oder von der Kamera abgewandt.  
Eine Sitzordnung, wie sie hauptsächlich im Frontalunterricht üblich ist.
- Möglichst alle Blickrichtungen bzw. die Gesichtsorientierung der Schüler sollen so exakt wie möglich erfasst werden.
- Die Überführung zwischen Welt- und Kamerakoordinatensystem ist bekannt.

## 2.4 Grundlegende Verfahren

Gesichtserkennung ist eine der fortschrittlichen Verfahren in der maschinellen Bildverarbeitung und wird ständig weiterentwickelt. Darunter fallen neben der Detektion des Gesichtes auch deren Analyse wie Orientierung, Übereinstimmungen oder das Erkennen von Mimik wie beispielsweise das Lächeln von Personen zum auslösen einer Kameras.

Bei vielen Anwendungen ist der Stand der Technik ein Neuronales Netz beteiligt.

### 2.4.1 Künstliches neuronales Netz

Ein künstliches neuronales Netz besteht aus miteinander verknüpften künstlichen Neuronen. Jedes Neuron besitzt Eingangswerte und einen Ausgabewert.

Um die Ausgabe zu bestimmen, werden die einzelne Eingangswerte des Neurons individuell Gewichtet, mit einer Übertragungsfunktion zusammengefasst und mittels einer Schwellenwertfunktion das Ergebnis bestimmt.

Um die Parameter (Gewichtung und Funktionen) des Neurons zu bestimmen, werden diese zufällig initialisiert und anschließend so angepasst, dass es zu einer gegebenen Eingabe das gewünschte Ergebnis liefert und der Fehler über dem gesamten Trainingsdatensatz minimal wird.

Soll ein gesamtes Netz trainiert werden, so wird jedes einzelne Neuron zufällig Initialisiert und anschließend so angepasst das der Fehler des Netzes auf dem Trainingsdatensatz minimal wird.  
[Kin94]

### 2.4.2 Convolutional Neural Network (CNN)

Die CNN definieren in vielen Anwendungsbereichen momentan der Stand der Technik. Sie sind eine Weiterentwicklung der neuronalen Netze die vor allem im Bereich Klassifizierung eingesetzt werden, unter anderem bei der Bild- und Spracherkennung. Der Unterschied liegt bei der Verwendung von gewichteten Faltungen der Eingabe.

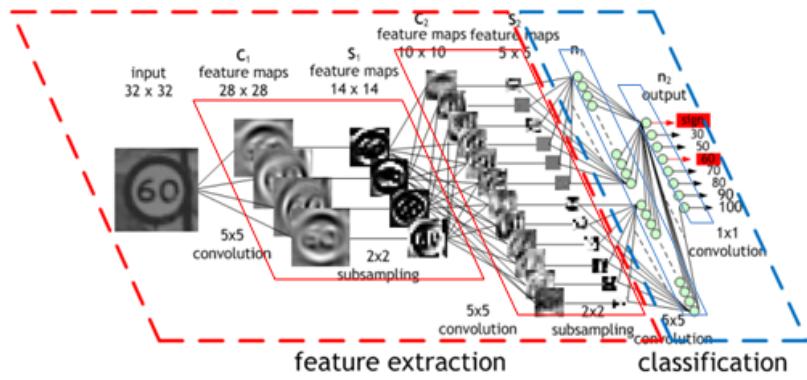


Abbildung 2.1: Beispiel für den Aufbau eines CNN zur Klassifizierung von Zahlen [Pee]

Durch die Faltung werden die Information aus den umliegenden Punkten eines Bereiches zusammengefasst und komprimiert an die nächste Schicht weitergegeben, um in der untersten Schicht alle vorhandenen Informationen zusammenzuführen. Der Faltungskern kann je nach Anwendung beliebig gestaltet sein, so ist eine Glättung durch einen Gauß-Kernel oder Kantendetektion durch einen Kirsch-Operator möglich.

Ein CNN kann in zwei Bereiche aufgeteilt werden, Feature Extraktion und Klassifizierung. Bei der Feature Extraktion werden verschiedene Kernel und Komprimierung auf den Eingabeinformationen angewendet um sie für den zweiten Teil aufzubereiten. Gelernt werden kann jeder einzelne Kernel für sich und die jeweiligen Bewertungen der Kernel und Neuronen.

[Ste12][Wik17b]

### 2.4.3 Active Appearance Model (AAM)

Dies ist ein Verfahren der Bildverarbeitung um Übereinstimmungen zu einem Modell zu finden. Dazu wird aus dem Trainingsdatensatz eine typische einheitliche Form des Objektes generiert mit seinen signifikanten Landmarks.

Soll nun zu einem Eingabebild die Übereinstimmung ermittelt werden, wird zuerst versucht es bestmöglich mittels Transformation in die typische einheitliche Form zu überführen. Sind dennoch Unterschiede vorhanden, liegt diese an der Erscheinung des Objektes.

[Wik14]

### 2.4.4 Constrained Local Model (CLM)

Dies ist ein Verfahren um mehrere Punkte eines Objektes zu lokalisieren. Dabei wird eine Wahrscheinlichkeitskarte für jeden einzelnen Punkt erstellt, wo dieser sich aufhalten kann, basierend auf der Ähnlichkeit des Punktes zur Darstellung im Bild. Nun wird versucht für das Bild, auf welchem gerechnet werden soll, für jeden Punkt den maximalen Wert zu erreichen zwischen passendem Farbverlauf und Wahrscheinlichkeit basierend auf der Position alle Punkte.

Dieser Art der Bestimmung von positionsabhängigen Punkten ist ziemlich zuverlässig und dennoch dynamisch genug um auch mit kleinen Veränderungen klar zu kommen.

Dies ist wichtig, bei der Detektion von leicht verformbaren Objekten wie Gesichter und ist zuverlässiger als das Active Appearance Model (AAM).

[CC06]

### 2.4.5 Constrained Local Neural Fields (CLNF)

Dabei handelt es sich um einen Gesichtsdetektor. Für die Detektion wird für jedes Merkmal ein eigener Detektor eingesetzt der auf einem Bildbereich arbeitet und eine Wahrscheinlichkeitskarte für dieses Merkmal erstellt.

Als nächster Schritt wird das Ergebnissen der Detektoren mit einer Karte der Position aller Landmarks mit ihren jeweiligen Abweichungen, kombiniert um somit die beste Position der Landmarks zu erhalten im Bezug auf den Farbverlauf und dem Verhältnis zu den anderen Landmarks.

[TB13]

### 2.4.6 Non-maximum suppression (NMS)

Ein Verfahren um ein lokales Maximum zu bestimmen und kann z.B. in einem Bild eingesetzt werden um Kanten exakter zu bestimmen. Als Eingabe für das Verfahren im Beispiel, wird das Ergebnis eines Kantendetektor z.B. Kirsch-Operator verwendet. Dabei gibt die Höhe des Farbwertes eines Pixels an, wie nahe es an einer Kante im Originalbild liegen. Bei der Verarbeitung wird nun der Farbwert jedes einzelnen Pixels des Eingabebildes mit seinen umliegenden verglichen und sollte es nicht maximal sein auf Null gesetzt.

Auf diese Weise bleibt nur noch ein Kantenpixel übrig. Wird das Verfahren auf die Bestimmung von Boxen eingesetzt, so wird jene Fläche bestimmt die von allen am ehesten beschreiben wird.

[NVG06][Wik16b]

### 2.4.7 Patch Experts

Eine Bewertung, wie wahrscheinlich ein Landmark an einer bestimmten Position im Bild dargestellt ist. Dazu wird ein Bereich um die Position ausgewertet.

[TB13]

### 2.4.8 Point Distribution Model (PDM) & Generalized Adaptive View-based Appearance Model (GAVAM)

Mit Point Distribution Model (PDM) können verformbare Objekte recht gut modelliert werden. Dabei wird die durchschnittliche Form  $\bar{X}$  des Objekts anhand der Eingabe bestimmt und eine Matrix  $P$  von Eigenvektoren ermittelt, um die möglichen Deformierungen darzustellen.

$$X = \bar{X} + P \cdot b$$

Somit kann durch einen Skalierungsvektor  $b$  alle möglichen Eingabeformen  $X$  des Objektes aus dem Durchschnittsmodell wiederhergestellt werden. Zur Vereinfachung reicht es, die signifikantesten Eigenvektoren in  $P$  aufzunehmen und dennoch  $X$  ausreichend genau beschreiben zu können.

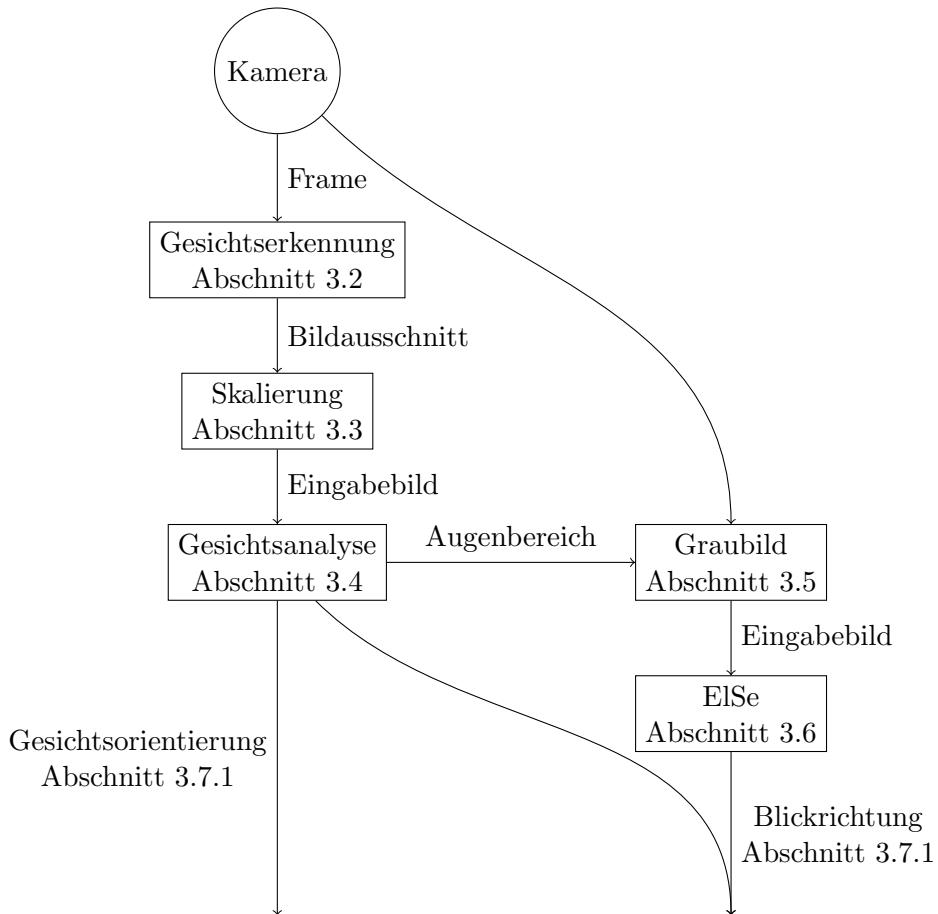
Ist bekannt welche Art der Verformung durch den Eigenvektor dargestellt wird, z.B. eine bestimmte Orientierung, so kann anhand des Skalierungsvektors die Rotation der Eingabe bestimmt werden, siehe Generalized Adaptive View-based Appearance Model (GAVAM).

Eine Problematik bei dieser Art der Bestimmung der Rotation entsteht, wenn neben der Verschiebung der Landmarks durch die Rotation, auch eine Deformierung des Objektes stattgefunden hat und somit keine eindeutige Lösung gefunden werden kann. Dies ist ein Problem wenn auf Gesichtern gerechnet wird, da immer eine Veränderung der Mundwinkel oder Augenlider vorhanden ist.

[Kyb07][MWM08][Wik17d]

# 3 Umsetzung

## 3.1 Ablauf der Implementierung



Da nur eine einzige fest montierte Kamera ohne Zoom eingesetzt wird, muss diese eine entsprechend hohe Auflösung besitzen, damit alle Personen zu erkennen sind. Zur Bestimmung der Blickrichtung sowie Kopfposition und Orientierung wird ein mehrstufiges Verfahren eingesetzt um alle Teilprobleme zu lösen.

Am Anfang müssen alle Gesichter, die im aktuellen Frame vorhanden sind, detektiert werden da nur auf diesen eine Berechnung ausgeführt wird. Dabei machen die relevanten Bereiche nur einen sehr geringen Anteil des gesamten Bildes aus. Dazu wird die MTCNN Face Detection eingesetzt, siehe Abschnitt 3.2, da dieses Verfahren im Vorabtests auf Probebildern einen sehr guten Eindruck gemacht hat und die meisten Gesichtern mit verschiedenen Größen und Blickrichtungen finden konnte. Laut Beschreibung des Verfahrens sollen sogar recht kleine Gesichter mit  $20 \times 20$  Pixeln erfassbar sein.

Für die weiteren Berechnungen muss bekannt sein, welcher Bereich von einem Gesicht im Frame eingenommen wird, um die relevanten Bildausschnitte aufzubereiten. Dabei muss das gesamte Gesicht in

der Box sein, weitere Besonderheiten gibt es nicht, da OpenFace einen eigenen Facedetector besitzt. Je nach verwendetem Trainingsdatensatz und darin enthaltener Annotation werden z.B. Kinn und Haaransatz noch als Gesichtsbereich oder schon als außerhalb betrachtet. So geben beiden Methoden (OpenFace und MTCNN-Face) Boxen aus, diese sind in ihren Ausmaßen allerdings nicht identisch. Da die folgende Verarbeitung eine OpenFace-skalierte Box erwartet, hat sich eine Vergrößerung der Box um 30% als sinnvoll erwiesen, bei Verwendung des MTCNN-Face Detektors um Ungenauigkeiten bezüglich der Position und Dimension des Kopfes im Bild entgegen zu wirken.

Sind mehrere Gesichter in mehreren Frames des Videos abgebildet, so muss auch eine Identitätszuordnung vorgenommen werden, damit bekannt ist welches Gesicht in Bild 1 welches in Bild 2 entspricht. Für die Zuordnung reicht es meist aus, jene Box zu wählen, die am ehesten den selben Bildausschnitt repräsentiert wie im vorigen Frame. Dies ist ausreichend, da die Gesichter sich meist weder groß bewegen noch sich die einzelnen Boxen der Probanden überlappen.

Damit sicher auf allen Gesichter gerechnet werden kann, ist eine semiautomatische Korrektur erforderlich um Falsch-Detektionen zu entfernen und fehlende Boxen der Gesichtern ergänzen zu können. Die gefundenen 5 Landmarks von MTCNN-Face Detection sind für die nachfolgende Berechnung nicht relevant, da sie gerade bei kleinen Gesichtern zu ungenau sind. Daher können alle bisher unternommenen Schritte auch von anderen Verfahren übernommen werden, da es sich hierbei nur um ein Vorverarbeitungsschritt handelt und zur Beschleunigung sowie Stabilität des späteren Berechnung beitragen soll. Damit das Verfahren im nächsten Schritt zuverlässig arbeiten kann, werden alle zu kleinen Bildbereiche hochskaliert, um die Gesichter auf eine Mindestgröße zu bringen, siehe Abschnitt 3.3

Diese Bildbereiche werden nun von OpenFace weiterverarbeitet um die Landmarks, die signifikanten Punkte eines Gesichtes, zu bestimmen. Durch die vorige Zuordnung der Gesichert kann das Verfahren gezielt auf den Person arbeiten und entsprechend einstellte CLNF verwenden, um bessere Ergebnisse zu erzielen, siehe Abschnitt 3.4. Außerdem können alle gefundenen Personen gleichzeitig (parallel) ausgewertet werden.

Um die Position der Pupille noch exakter zu ermitteln wird ElSe verwendet, wozu eine Graukonvertierung durchgeführt werden muss, siehe Abschnitt 3.5. Ziel ist es, durch eine exakte Bestimmung der Pupillenposition, auch eine genaue Blickrichtungsbestimmung zu erhalten. Allerdings muss das Ergebnis von ElSe auf Plausibilität geprüft werden, um grobe Fehler zu vermeiden. Für die Umsetzung siehe Abschnitt 3.6.

Nun wird auf Basis der Landmarks und Kameraparameter die Position und Orientierung der Gesichter sowie die Blickrichtung bestimmt, siehe Abschnitt 3.7. Diese Ergebnisse können dann von weiteren Anwendungen verwendet werden.

## 3.2 MTCNN Face Detection

Multi-task Cascaded Convolutional Networks (MTCNN) ist ein Algorithmus zur Detektion von Gesichtern und Bestimmung von 5 Gesichts-Landmarks in Farbbildern. Dabei werden drei CNN auf einer Bildpyramide angewendet um zuverlässig Gesichter verschiedenster Größe zu erkennen. Des Weiteren wird für die Detektion der Gesichter auch deren Ausrichtung berücksichtigt, um bessere Ergebnisse zu erzielen.

Sein Einsatzgebiet ist die Vorverarbeitung eines Frames für die spätere Auswertung. Somit soll dieser Schritt von einem möglichst robusten Verfahren durchgeführt werden. Dabei wird auf einem hochauflösendem Bild gearbeitet mit verhältnismäßig kleinen, verschiedenen Großen und weit verteilten Gesichter.

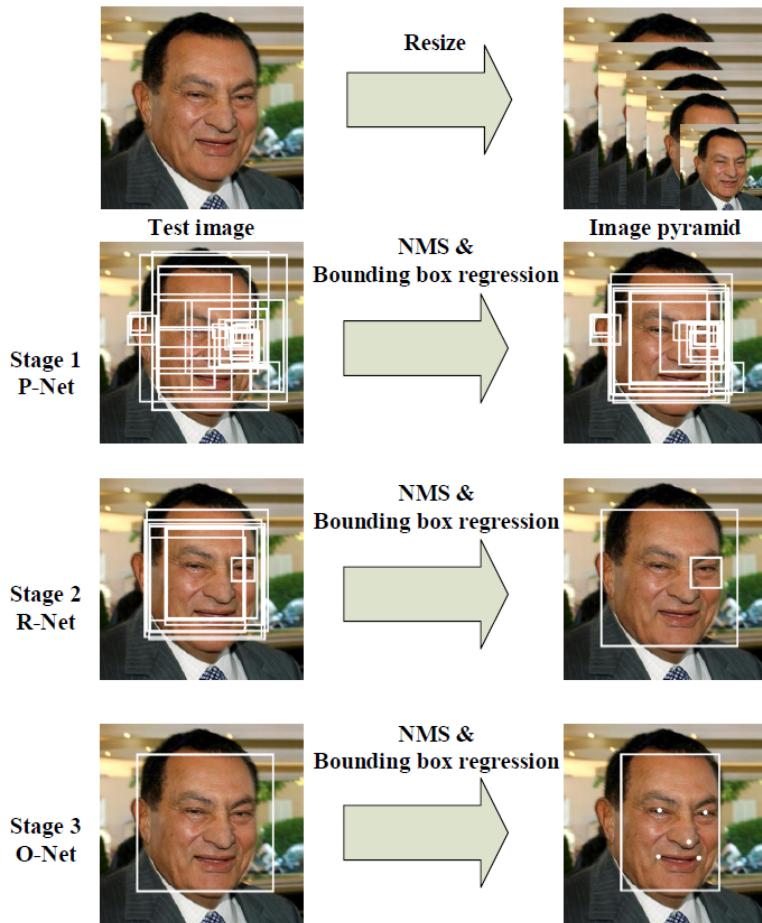


Abbildung 3.1: Darstellung des Funktionsablaufes von MTCNN[KZ15]

### 3.2.1 Die 3 Stufen der Verarbeitung

Für die gute Detektionsqualität sorgt die dreistufige Verarbeitung mit verschiedenen CNN auf einer Bildpyramide. Bei der Bildpyramide handelt es sich um ein in verschiedenen Größen skaliertes Bild, damit der gesuchte Inhalt in der gewünschten Auflösung abgebildet ist, ohne etwas über den Inhalt zu wissen.

Dies ist von Vorteil, damit das CNN auf eine feste Größe von Gesichtern optimiert werden kann, um das Lernen nicht zusätzlich zu erschweren. So werden nur die Farbverläufe gelernt und nicht weite durch die Skalierung erschwert, wodurch das CNN auf seine jeweilige Aufgabe besser optimiert werden kann.

#### Stufe 1

Beim ersten Verarbeitungsschritt werden alle Bereiche eines Bilds gesucht, in denen möglicherweise ein Gesicht zu erkennen ist. Dazu wird für die Detektion ein CNN, dem sogenannten Proposal Network (P-Net) eingesetzt, das alle möglichen Bounding-Boxen ermittelt in denen ein Gesicht zu sehen sein könnte. Diese Bounding-Boxen werden anschließend mit einem NMS ausgedünnt, um die am stärksten überlappenden Boxen zusammen zu fassen. Dies ist notwendig, da dieses CNN zwar recht schnell arbeitet, allerdings auch mit einer sehr großen False-True-Fehlerrate (Erkennen trotz nicht vorhanden).

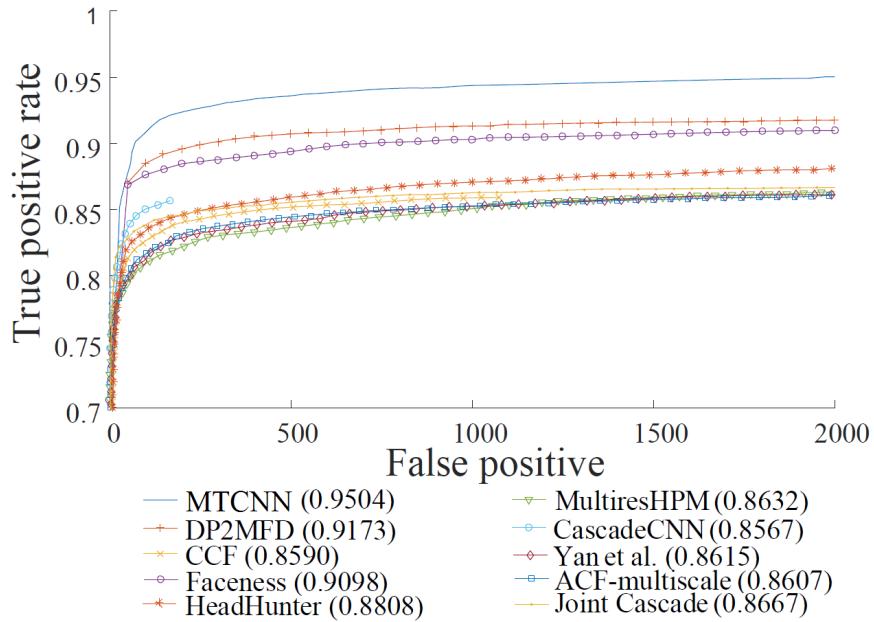


Abbildung 3.2: Qualität der Detektion der verschiedenen Verfahren im Vergleich [KZ15]

## Stufe 2

Anschließend werden die möglichen Bereiche mittels eines weiten CNN analysiert, damit alle Nicht-Gesichtsbereiche erkannt und entfernt werden können. Dies wird von dem Refine Network (R-Net) übernommen und anschließend die möglichen Bounding-Boxen mittels NMS noch weiter reduziert.

## Stufe 3

Der letzte Schritt wird von einem deutlich genaueren CNN übernommen, um ein Gesicht zu detektieren, dem sogenannten Output Network (O-Net). Womit die resultierenden exakten Boxen mit ihren jeweiligen 5 Landmarks ermittelt werden.

### 3.2.2 Qualität

MTCNN Face Detection ist bei der Zuverlässigkeit im Vergleich zu anderen bekannten Verfahren überlegen, siehe Abbildung 3.2, und zudem Echtzeit fähig auf  $640 \times 480$  Großen Bilder. Dabei können auch Gesichter mit einer Größe von  $20 \times 20$  Pixel erfolgreich erkannt werden.

Somit sind alle Anforderungen erfüllt um mit diesem Verfahren den vorhanden Frame für die nachfolgenden Berechnungen vorzubereiten. Ein Test bestätigt diese Annahme, siehe Abbildung 3.3.

## 3.3 Skalieren von Bildern

OpenFace arbeitet laut Angabe im Paper [TB16] am besten auf Gesichtern mit einer Größe von 100 Pixel, daher werden die Bildbereiche auf diese Größe gebracht. Dies ist notwendig, da die Berechnungen meist auf recht kleinen Bildausschnitten ausgeführt werden muss.

Dabei ist es wichtig, dass die Gesichtsmerkmale möglichst gut rekonstruiert werden, um die entsprechenden Landmarks zu bestimmen, dabei erhöht sich der Informationsgehalt der Bilder nicht, sie sind

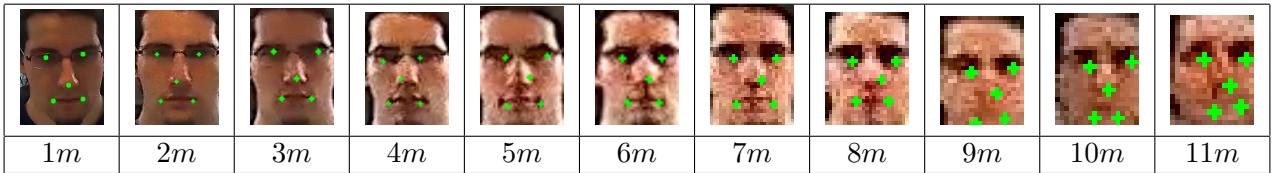


Abbildung 3.3: Dargestellt ist die Box und die 5 Landmarks von MTCNN-Face bei verschiedenen Distanzen des Probanden zur Kamera

nur besser nutzbar, da sie dem Trainingsdatensatz stärker ähneln.

Die von MTCNN gelieferten und vergrößerten Boxen werden auf  $130 \times 180$  Pixel gebracht, damit das beinhaltete Gesicht auf der gewünschte Größe dargestellt wird. Neben der Skalierung des Bildausschnittes muss bekannt sein, wie Punkte im skalierten Bildausschnitt in das Frame überführt werden können, damit dies bei späteren Berechnungen berücksichtigt wird.

Der Skalierungsfaktor ist für jeden Bildausschnitt individuell und kann sich über die Zeit ändern, wenn sich z.B. die Distanz zwischen Person und Kamera ändert. Von einer zu starken Vergrößerung ist abzuraten, da sich der Rechenaufwand pro Gesicht erhöht und die Zuverlässigkeit der Berechnungen von OpenFace sinkt, z.B. durch Falschdetektion.

### 3.3.1 Bicubic-Skalierung

Der neue Farbwert wird ermitteln, indem die umliegenden  $4 \times 4$  Pixelwerte betrachtet werden um den Farbverlauf als eine Funktion 3. Grades zu bestimmen. Somit werden feinere Details besser dargestellt als beim linearen Verfahren und Kanten bleiben eher erhalten. Allerdings kann es durch den bestimmten Verlauf auch zum Überschwingen kommen, wodurch Fehlfarben entstehen können. Ein Beispiel als Ergebnis dieses Verfahrens ist in Abbildung 3.4 zu sehen.

[Wik16a]

### 3.3.2 Lanczos-Skalierung

Dieser Filter basiert auf einem bewerteten Durchschnitt der umliegenden Pixel um den neuen Pixelwert zu erhalten. Die Bewertung der einzelnen Pixel wird durch eine Sinc-Funktion bestimmt, damit weiter entferntere Pixel schwächer bewertet werden als näher liegende, siehe Abbildung 3.5.

Die Funktion kann und wird für die Anwendung auf einen  $8 \times 8$  Pixel großen Bereich begrenzt. Außerdem wird durch den Kurvenverlauf der Bewertungsfunktion eine gewisse Bildschärfe erreicht.

[Wik16c]

$$L(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} \cdot \frac{\sin(\pi \frac{x}{a})}{\pi \frac{x}{a}} & \text{wenn } -a < x < a, a \neq 0 \\ 1 & \text{wenn } x = 0 \\ 0 & \text{sonst} \end{cases}$$

### 3.3.3 Linear-Skalierung

Um den neuen Farbwert zu ermitteln, wird zwischen den nächstgelegenen umliegenden Pixel linear Interpoliert, wodurch weitere Farbwerte entstehen. Das Ergebnis ist gleichmäßiger als Neares Neighbor, und dennoch ein recht einfaches Verfahren. Die Kanten wirken allerdings unscharf, siehe Abbildung 3.6.



Abbildung 3.4: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels bikubischem Verfahren auf 512 Pixel vergrößert und bei Lena die Differenz zum originalen Lena-Bild bestimmt



Abbildung 3.5: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels Lanczus-Verfahren auf 512 Pixel vergrößert und bei Lena die Differenz zum originalen Lena-Bild bestimmt

### 3.3.4 Nearest-Neighbor-Skalierung

Dieses Verfahren verwendet als neuer Farbwert, den gleichen Wert wie das nächstgelegene Pixel. Dadurch werden nur die ehemaligen Pixel größer und das Gesicht wirkt sehr Kantig, da keine neuen Farbwerte bestimmt werden, siehe Abbildung 3.7. Bei der Vergrößerung des Schachbretts sind kein Farbfehler aufgetreten, da nur zwei Farben vorhanden und Positionsabhängig sind.

### 3.3.5 Qualität der Skalierung

Nun wird die Auswirkung der verschiedenen Skalierungsverfahren auf die Qualität der Berechnung untersucht. Dazu wurde der „Datensatz“ verwendet, linear um den angegebenen Faktor verkleinert und mittels der angegebenen Verfahren wieder vergrößert und von OpenFace ausgewertet.

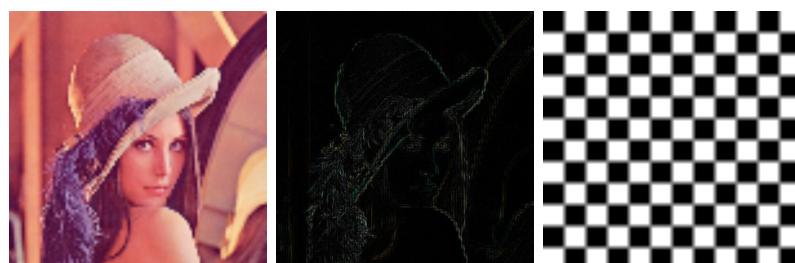


Abbildung 3.6: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels linearer Interpolation auf 512 Pixel vergrößert und bei Lena die Differenz zum originalen Lena-Bild bestimmt



Abbildung 3.7: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels Nearest-Neighbor auf 512 Pixel vergrößert und bei Lena die Differenz zum originalen Lena-Bild bestimmt

## Position

Als erstes werden die berechnete Distanzen mit denen des Datensatzes verglichen. In Abbildung 3.8 ist die Abweichung entlang der X-Achse dargestellt. Nearest-Neighbor liefert die genauesten Ergebnisse, auch wenn die Detektion früher ausfällt als die der anderen drei.

Auf der Y-Achse ist das Lineare-Verfahren etwas besser als die der Anderen, das Nearest-Neighbor ist hierbei überraschend das Schlechteste, siehe Abbildung 3.9.

Nur schwer zu erkennen da der Unterschied minimal ausfällt, ist bei der Bestimmung der Z-Position das Nearest-Neighbor-Verfahren ebenfalls am Besten, siehe Abbildung 3.10. Die anderen drei liefern nahezu eine identisch Qualität. Bei sehr kleinen Skalierungen existieren durchaus auch sehr große Fehler, diese wurden allerdings bei der Darstellung abgeschnitten, da bei dieser Größe die Detektionsrate so klein ist, dass sie nahezu irrelevant werden.

## Orientierung

Des weiteren wird der berechneten Winkel um die jeweilige Achse betrachtet und mit den korrekten verglichen. Die geringste Abweichung bei der bestimung der X-Rotation liefert Nearest-Neighbor, siehe Abbildung 3.11. Auffällig ist außerdem der kleinere Wertebereich des Linearen-Verfahrens.

Auch bei der Y-Rotation schneidet Nearest-Neighbor am besten ab, siehe Abbildung 3.13, allerdings sind die unterscheide minimal.

Bei der Z-Rotation ist kein erkennbarer Unterschied zwischen den einzelnen Verfahren, wobei bei Nearest-Neighbor deutlich früher der Wertebereich sinkt.

### 3.3.6 Ergebnisse Angeben

Es zeigt sich, dass bei der Analyse von Gesichter das Nearest-Neighbor Verfahren die genauesten Ergebnisse liefert. Allerdings ist der Arbeitsbereich deutlich eingeschränkter als im Vergleich zu den anderen Verfahren. Die benötigte Mindestgröße des Gesichts im Orginal und dem geringen Wertebereich im Bezug auf die Rotationen zeigt, das dieses Verfahren eher ungeeignet ist.

Bei dem Linearen-Verfahren ist die Abweichung bei den Rotationen am größten, auch wenn es sich nur um etwa ein halbes Grad handelt. Zwischen dem Bicubic- und Lanczos-Verfahren gibt es in den relevanten Bereichen keinen signifikanten Unterschied, wobei das Lanczos in den kleineren Bereichen gleichmäßiger Ergebnisse. Somit kann die Wahl des Verfahrens vom Rechenaufwand abhängig gemacht werden.

**ToDo:** Bessere Auswertung

Bicubig am besten

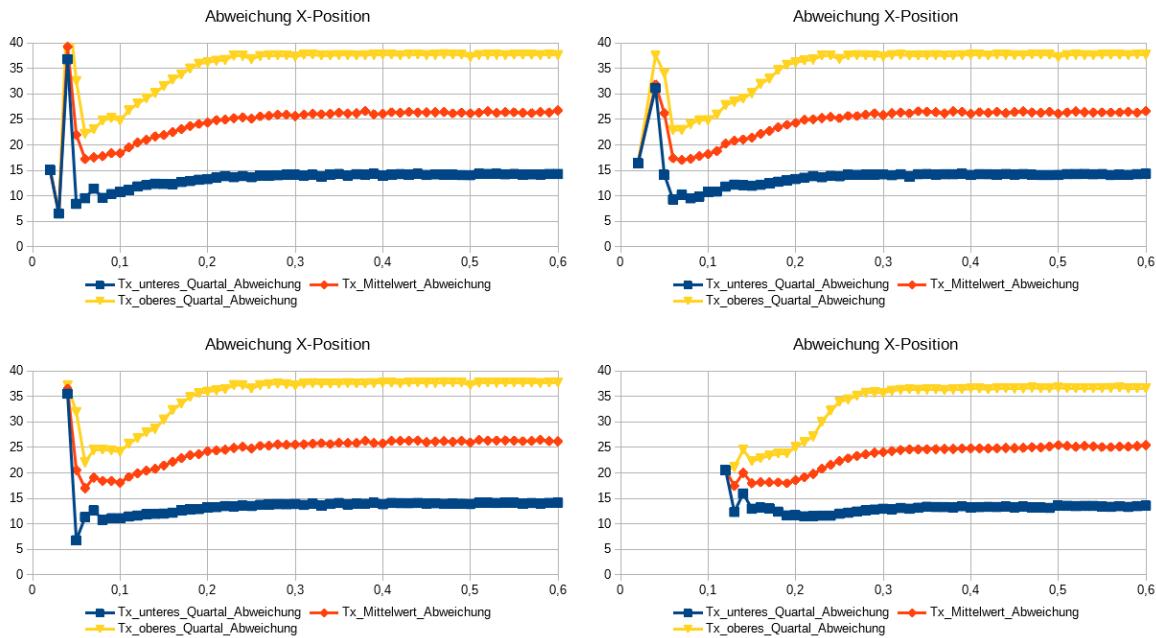


Abbildung 3.8: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung in X-Richtung (Y-Achse) in Millimeter. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

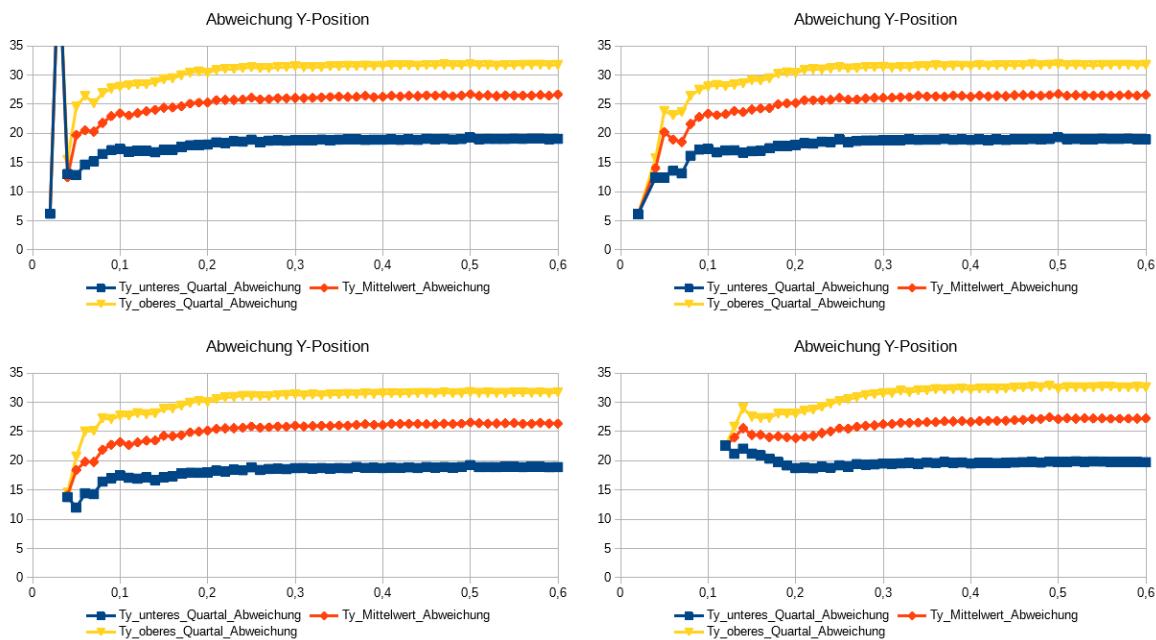


Abbildung 3.9: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung in Y-Richtung (Y-Achse) in Millimeter. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

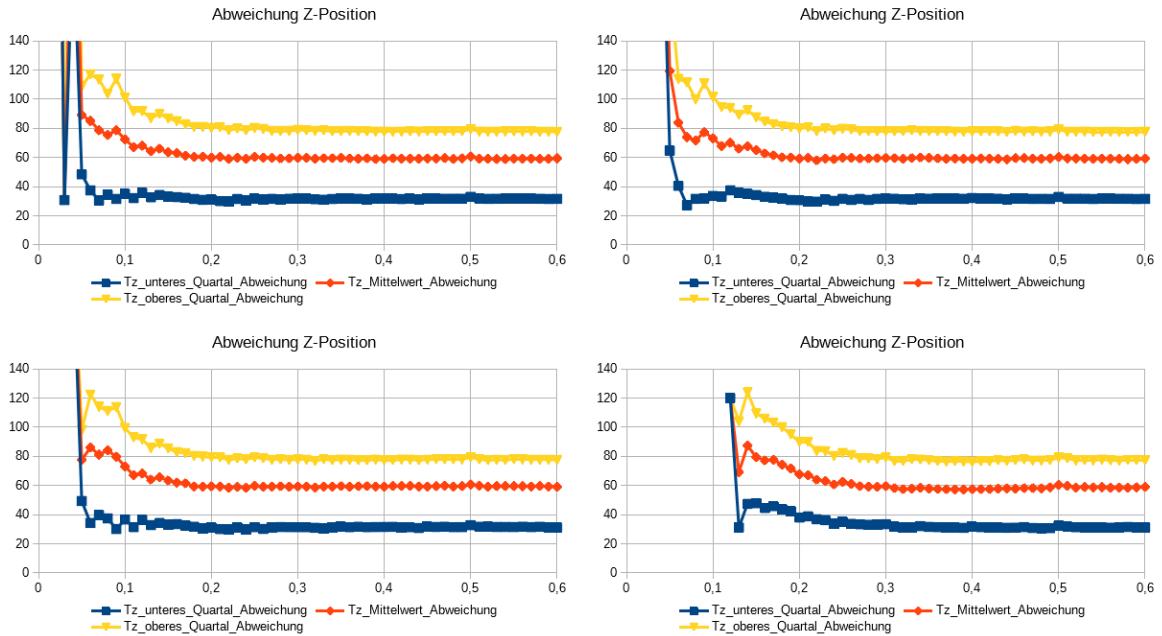


Abbildung 3.10: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung in Z-Richtung (Y-Achse) in Millimeter. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

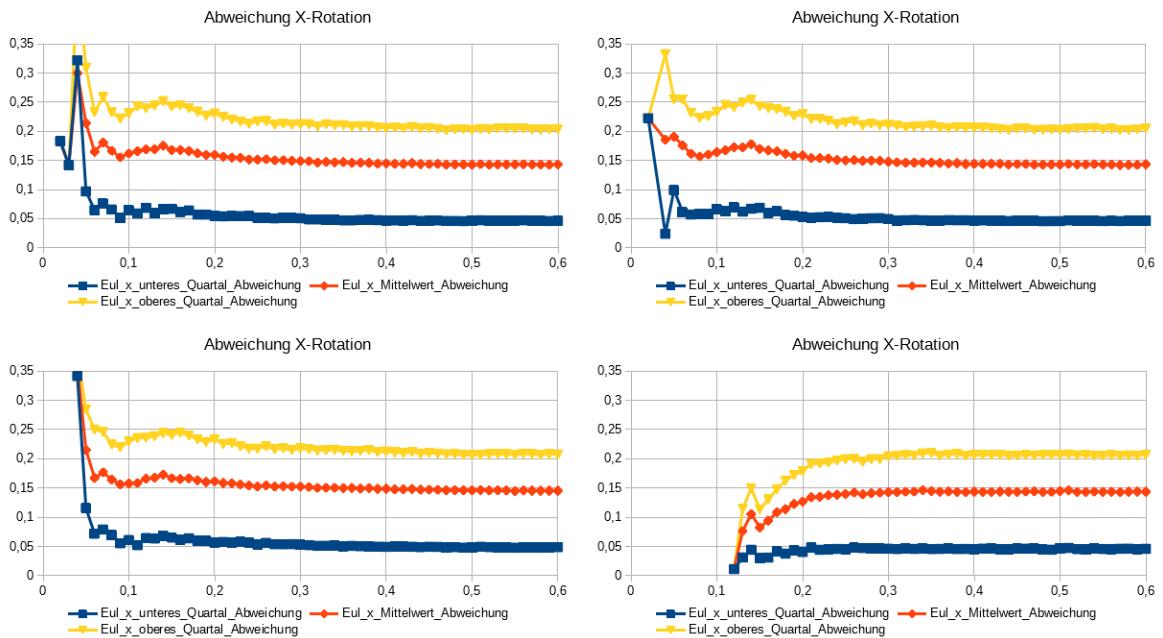


Abbildung 3.11: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in X-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

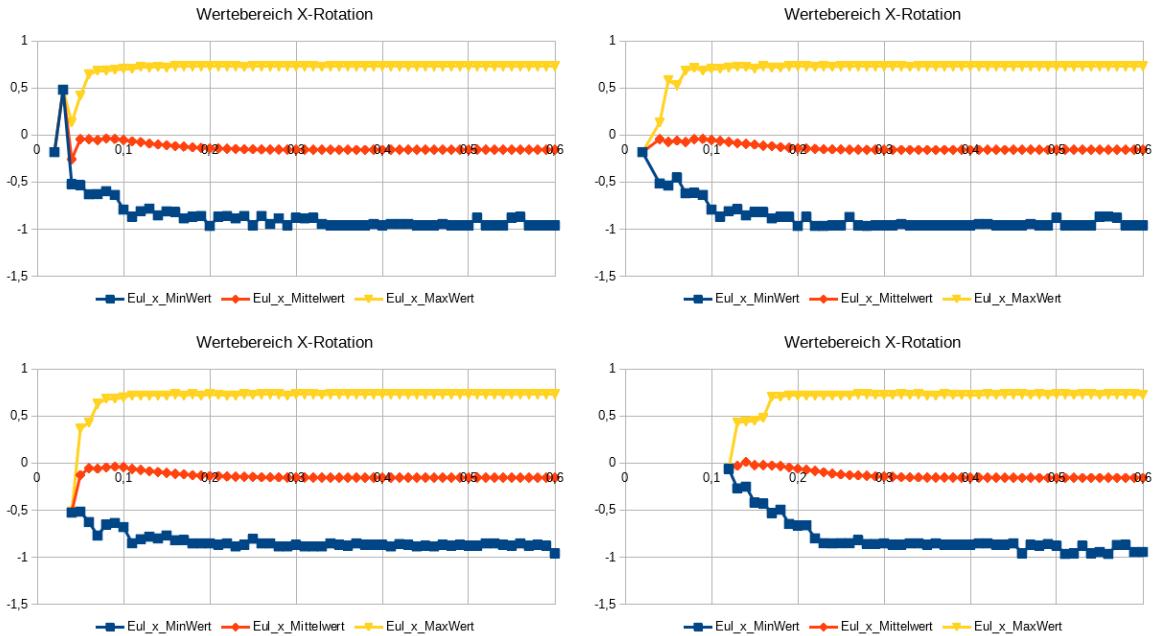


Abbildung 3.12: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in X-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

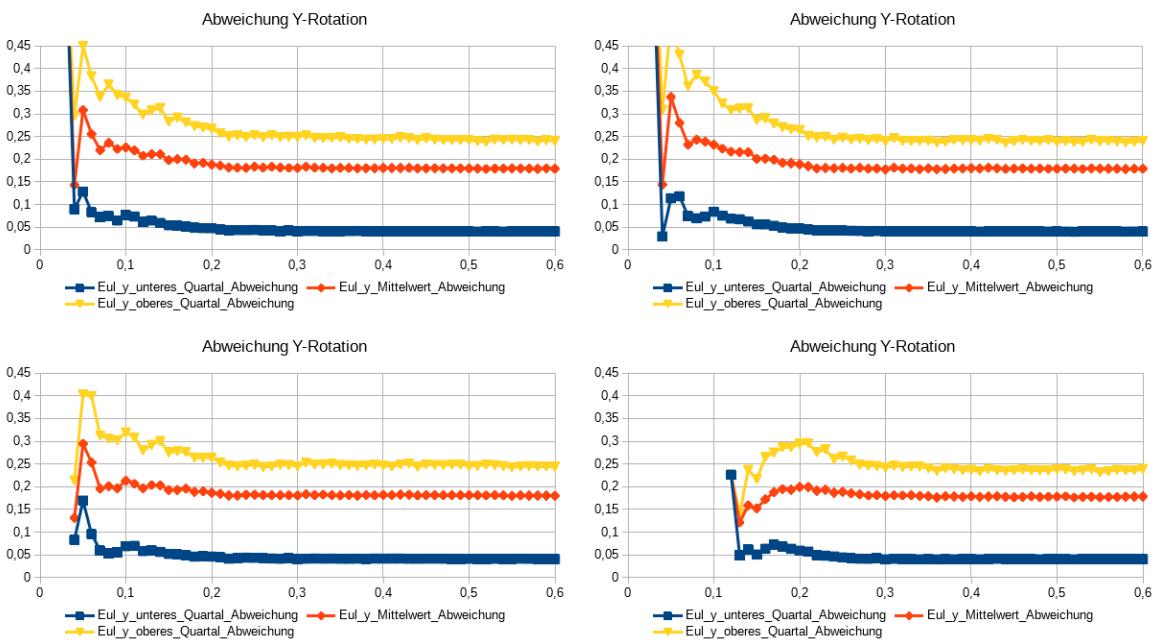


Abbildung 3.13: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in Y-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

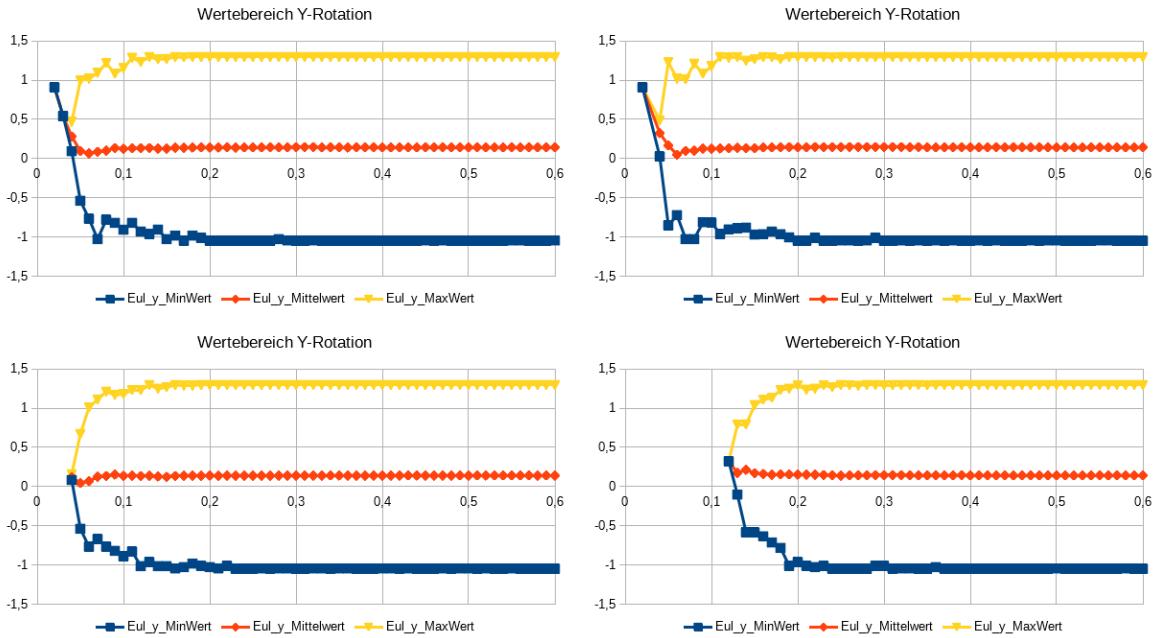


Abbildung 3.14: Zusammenhang zwischen der Skalierung (X-Achse) und der Wertebereich des Winkels in Y-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

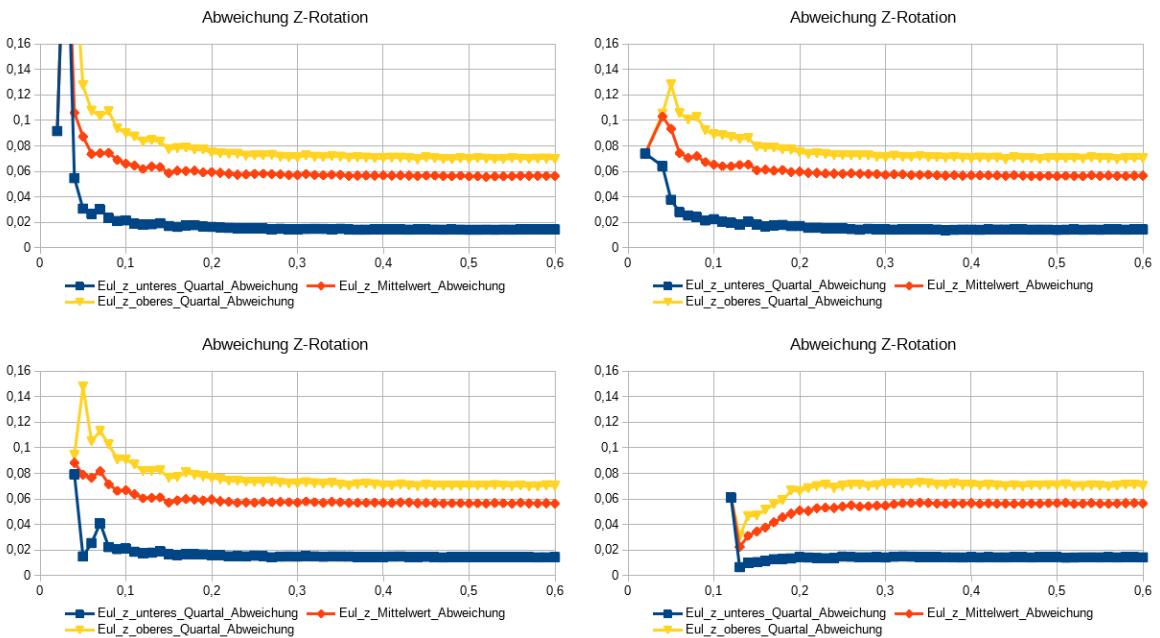


Abbildung 3.15: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in Z-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

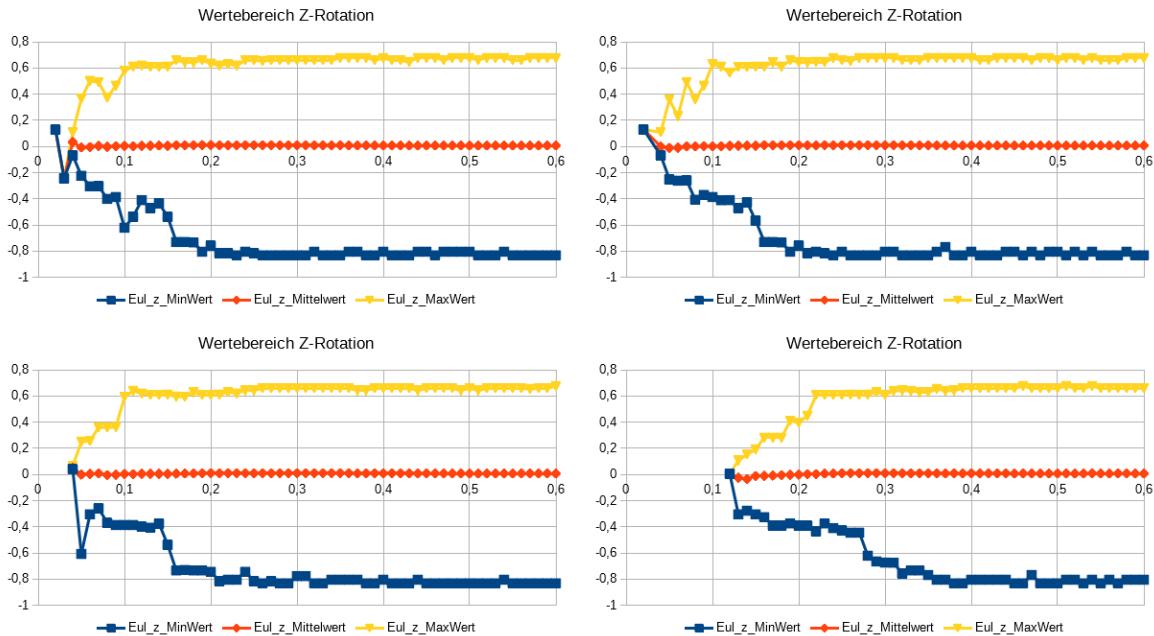


Abbildung 3.16: Zusammenhang zwischen der Skalierung (X-Achse) und der Wertebereiche des Winkels in Z-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

## 3.4 OpenFace

Ein Open-Source Echtzeitverfahren auf Basis von CLNF zur Bestimmung und Analyse von Gesichtsmerkmalen in Grau-Bildern und Videos. Dabei stehen für diese Anwendung nur die Kameraparameter zur Verfügung und keinerlei Zusätze wie ein Tiefenbild (kann mitverwendet werden wenn vorhanden) oder Infrarotbeleuchtung der Szene.

OpenFace kann 68 Landmarks ermitteln die das Gesicht beschreiben, und mit deren Hilfe Position, Blickrichtung und Gesichtsmerkmale bestimmen. Sollte ein Video als Quelle fungieren, kann OpenFace auch lernen, wodurch eine zuverlässigere Verarbeitung erzielt werden kann.

Als Ergebnis ist die Kopfposition (Translation und Orientierung) sowie Blickrichtung von Interesse, da mit ihnen zurückrechnet werden kann wohin die Person schaut.

Der Rechenaufwand zur Verarbeitung des Eingabebildes ist so ausgelegt, dass ein Webcam-Video in Echtzeit ausgewertet werden kann, dies ist im aktuellen Fall nicht notwendig, da es sich um eine nachträgliche Auswertung handelt, bei der es vor allem um Genauigkeit geht.

### 3.4.1 Bestimmung der Landmarks

Für die Bestimmung der Landmarks wird OpenFace auf den Bildausschnitten eingesetzt. Dies bietet mehrere Vorteile, so wird nur auf Bildbereichen gearbeitet, in denen ein Gesicht zu sehen ist und unnötige Suche vermieden. Außerdem kann für jede Person die passende Initialisierung des CLNF basierend auf dem letzten Ergebnis dieser Person gewählt werden, auch für jene die nur selten dargestellt sind. Auf diese Weise kann der Bildausschnitt möglichst exakt und gleichzeitig mit den anderen ausgewertet werden.

Für die eigentliche Bestimmung der Landmarks bietet OpenFace zwei verschiedene Methoden, die Berechnung auf Bildern und Videos. Der Hauptunterschied ist das Lernen, dass bei der Videoauswer-

tung verwendet wird, wodurch sich der Arbeitsbereich deutlich erhöht und bessere Ergebnisse liefert werden. Dies liegt an der Anpassung des Modells und dem möglichen Tracking der Landmarks. Dies ist interessant für die spätere Anwendung, da somit auch Einzelbilder verwendet werden können, die eine deutlich höhere Auflösung besitzen als ein Video. Allerdings sinkt bei der Verwendung von Einzelbildern der maximale Winkel relativ zur Kamera beträchtlich. Außerdem hat sich gezeigt, dass bei Verwendung eines Videos das Gesicht deutlich kleiner dargestellt sein kann bis keine Auswertung mehr möglich ist und sollte ein Gesicht im aktuellen Farame erfolgreich detektiert werden, auch die nachfolgenden Frames durch das lernen ausgewertet werden können.

Dennoch kann es passieren, dass trotz allem ein Gesicht falsch detektiert wird, wie z.B. das Erkennen eines sehr kleinen Gesichtes innerhalb einer Ohrmuschel. In solch einem Fall muss das CLNF zurückgesetzt werden, damit sich der Fehler nicht fortpflanzt.

### **Gesichts-Landmarks: Detektion und Verfolgung**

Für die Bestimmung und Tracking der Landmarks wird ein Conditional Local Neural Fields (CLNF) eingesetzt. Dabei handelt es sich im Grunde um ein Constrained Local Model (CLM) nur mit verbesserten Patch Experts und Optimierungsfunktionen.

Die beiden Hauptkomponenten des CLNF von OpenFace ist das Point Distribution Model (PDM) zur Erfassung der Anordnung der Landmarks und Patch Experts zum Erfassen der Variante der einzelnen Landmarks.

Zu Beginn werden verschiedene initiale Hypothesen aus der dlib-Bibliothek verwendet und die Passende zur Eingabe ausgewählt. Bei den unterschiedlichen initial Hypothesen handelt es sich um die Darstellung verschiedener Gesichtsorientierungen auf denen unterschiedliche Netze trainiert wurden. Dieser Herangehensweise ist langsam, aber auch exakter als eine einfache Hypothese. Wird ein Tracing, das Verfolgen der Landmarks über mehrere Frames, durchgeführt, wird als initiale Hypothese das Ergebnis aus dem letzten Frame verwendet. Sollte das Tracing scheitern, wird das CNN reseted um Neu zu beginnen.

Auf diese Weise werden 68 Gesichts-Landmarks und weitere 28 pro Auge erfasst. Zur Berechnung auf den Gesichtern sollten diese laut Paper [TB16] eine Optimalgröße von 100 Pixeln für eine zuverlässige Detektion aufweisen.

### **Detection der Gesichtsmerkmale**

Dieser Schritt kann von OpenFace ausgeführt werden, ist aber im aktuellen Fall nicht von Relevanz, da die Blickrichtung von Interesse ist und nicht die Mimik der Probanden.

#### **3.4.2 Veröffentlichte Genauigkeit**

Um die Qualität der Berechnung auf dem Kopf zu bewerten wurde der „Biwi Kinect head pose“[FGG11], „ICT-3DHP“[BRM12] und „BU Datensatz“[CSA00] ausgewertet. Dabei handelt es sich um Portrait-Fotos von Probanden, deren Körper in Richtung Kamera ausgerichtet ist und ihren Kopf in eine beliebige Richtung drehen. Für die Genauigkeit der Kopfposition haben sich folgend Werte ergeben in Grad:

	Yaw	Pitch	Roll	Mean	Median
Biwi Kinect	7.9	5.6	4.5	6.0	2.6
BU dataset	2.8	3.3	2.3	2.8	2.0
ICT-3DHP	3.6	3.6	3.6	3.6	-

Für die Qualität wurde der Augendatensatz „Appearancebased gaze estimation in the wild“[XZ15]

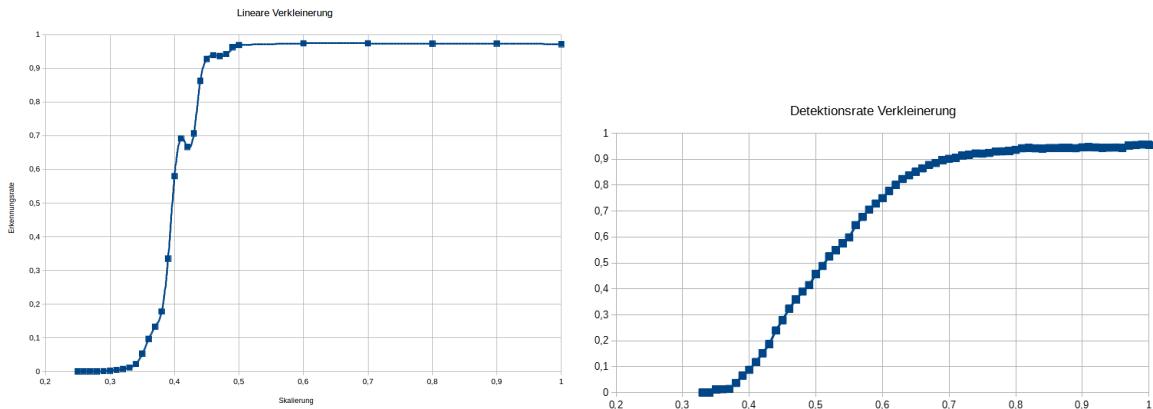


Abbildung 3.17: Die Bilder aus Labeled Faces in the Wild [HMLLM12] (links) und Random Forests [FDG<sup>+</sup>13] wurden mit den Faktor auf der X-Achse linear verkleinert und die Erkennungsrate Y-Achse abgebildet

zur Bestimmung der Blickrichtung verwendet und es ergab sich ein durchschnittlicher Fehler von 9.96 Grad.

### 3.4.3 Auswirkung der Größe

Durch die Aufgabenstellung muss das Verfahren zuverlässig bezüglich der Distanzen bzw. Darstellungsgröße sein. Zur Messung wurde der Datensatz von Labeled Faces in the Wild [HMLLM12] verwendet. In diesem Datensatz ergibt sich im Originalbild eine durchschnittliche Kopfbreite von 94 Pixel. Bei Random Forests for Real Time 3D Face Analysis [FDG<sup>+</sup>13] ist die durchschnittliche Breite 78 Pixel. Zur Durchführung wurden die Größe der Bilder mit dem Skalierungsfaktor multipliziert und linear verkleinert um so kleinere, weiter entfernte Gesichter zu erhalten und anschließend mit dem Image-Detector von OpenFace zu verarbeiten. Das Ergebnis ist in Abbildung 3.17 dargestellt.

Es ist zu erkennen, dass die Wahrscheinlichkeit auf eine erfolgreiche Detektion ab 0.5, also gesichert mit etwa 47 Pixel Breite, rapide abnimmt. Bei der in Abschnitt 2.1 beschriebenen Kamera entspricht dies einer Distanz von etwa 4.5m.

Bei der maximalen Distanz auf der gearbeitet werden soll (8.5m) ergibt sich eine Gesichtsgröße von etwa 22 Pixel, das einer Skalierung von 0.25 entspricht. Bei dieser Bildgröße ist in der Standardanwendung ohne Skalierung keine Detektion möglich, siehe Abbildung 3.17.

### 3.4.4 Auswirkung der verschiedenen Skalierungsverfahren auf Detektion

Um auf den gewünschten Distanzen arbeiten zu können, wird der jeweilige Bereich hochskaliert. Dazu wird das ursprüngliche Bild ( $250 \times 250$ ) linear um den angegebenen Faktor verkleinert und anschließend mit den angegebenen Verfahren auf  $300 \times 300$  wieder vergrößert, damit die abgebildeten Gesichter in etwa 100 Pixel groß sind. Die Wahrscheinlichkeit auf eine Detektion ist in Abbildung 3.18 dargestellt. Es ist zu erkennen, dass durch die Vergrößerung, jene Gesichter in Bereichen die normal nicht erkennbar sind, ausgewertet werden können.

Als das ungeeignetste Verfahren hat sich Nearest-Neighbor herausgestellt, siehe blaue Linie Abbildung 3.18, da die Detektionsrate deutlich früher abfällt als bei den anderen. Die anderen haben sehr ähnliche Ergebnisse, nur das Lineare Verfahren ist etwas schlechter. Dennoch werden die Anforderungen, eine Detektion auf Gesichtern mit 22 Pixel (Skalierung 0.25), von allen erfüllt.

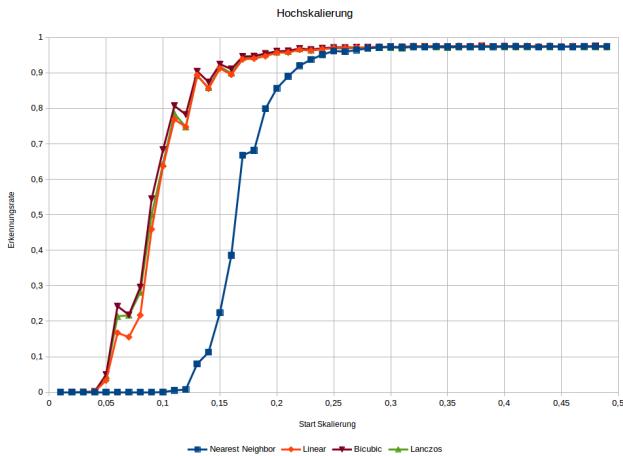


Abbildung 3.18: Die Bilder aus Labeled Faces in the Wild [HMLLM12] wurden mit den Faktor auf der X-Achse linear verkleinert und mit den verschiedenen Verfahren wieder vergrößert Abschnitt 3.3. Aufgetragen gegen die Detektionswahrscheinlichkeit. Nearest-Neighbor (blau), Linear (rot), Bicubic (braun), Lanczos (grün)

Ausgehend vom Skalierungsfaktor des Linearen-, Bicubic- und Lanczos-Verfahren wären mit der verwendeten Kamera auch Distanzen bis zu 14m möglich. Allerdings ist das Bild durch die Verkleinerung deutlich besser als Originalaufnahmen, da Pixelrauschen und Ähnliches nicht vorhanden ist.

### 3.4.5 Auswirkung von Pixelrauschen auf Detektion

Mit diesem Test soll geprüft werden, welches der Verfahren auch stabil gegen Rauschen ist. Um Pixelrauschen zu simulieren, wurden die Bilder aus Labeled Faces in the Wild [HMLLM12] entsprechend verkleinert, mit Rauschen versehen um sie anschließend mit den unterschiedlichen Verfahren zu vergrößern.

Das Rauschen wird für jedes Pixel simuliert, indem eine Wahrscheinlichkeit von 50% besteht, eine gleich verteilte Abweichung von  $\pm 10\%$  des Originalen Farbwertes. Dieser Vorgang wurde für jedes Bild viermal wiederholt um Zufälligkeiten bei der Rauschsimulation zu vermeiden.

Wie zu erwarten ist Nearest-Neighbor am schlechtesten, aber auch zwischen den anderen Verfahren sind nun unterscheiden zu erkennen, siehe Abbildung 3.19. Die gesamte Erkrankungsrate ist signifikant kleiner als ohne Rauschen, wobei die Position (0.15), ab welcher die Erkennungsrate rapide abfällt, beibehalten wird.

### 3.4.6 Arbeitsbereich bezüglich Rotation

Von Interesse sind auch die Winkel, bei den Gesichter in verschiedenen Skalierungen noch erkannt werden, siehe Abbildung 3.20.

Hier ist zu erkennend das der Wertebereich ab 0.7 abnimmt und ab 0.5 recht schnell. Dieser Bereich ist von Interesse, da selbst wenn ein Gesicht in dieser Größe allerdings außerhalb des Wertebereiches vorhanden sein sollte, dieses nicht erkannt wird.

Der Wertebereich auf den einzelnen Achsen ist ausreichend für die Anwendung sollte das Ziel der Aufmerksamkeit sich in der näher der Kamera befinden. Auch wenn die Rotation parallel zur Horizontalen etwas größer sein könnte.

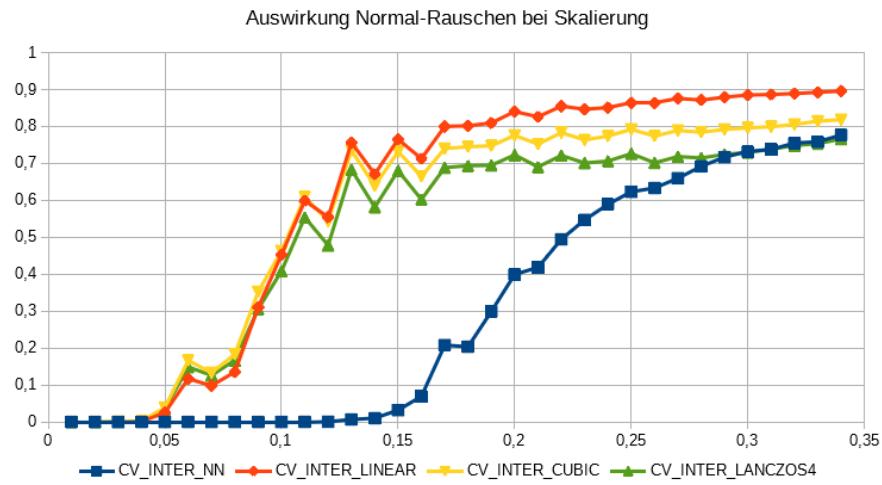


Abbildung 3.19: Bilder aus Labeled Faces in the Wild [HMLLM12], mit dem X-Faktor verkleinert, um jedes Pixel mit 50% Wahrscheinlichkeit auf  $\pm 10\%$  Gleichverteilung der Abweichung

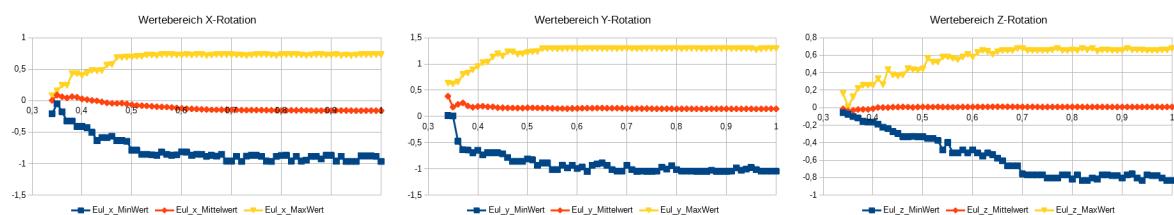


Abbildung 3.20: Darstellung der noch detektierten Wertebereiche in Bogenmaß.



Abbildung 3.21: Dies sind die Eingabebilder der verschiedenen Konverter von Farbe nach Grau. Links eine Farbpalette, Mitte Lena und Rechts ein Augenausschnitt aus dem Augendatensatz [WBZ<sup>+</sup>15]

### 3.5 Umwandlung von Farbbild nach Graubild

Da die Berechnungen von ElSe auf Graubildern arbeitet und das Eingabebild in Farbe ist, muss es in ein Graubild umgewandelt werden.

Die Wahl des Verfahrens beruht auf der Anforderung, dass vor allem der Farbunterschied zwischen Pupille und der Umgebung maximal ist. Die Pupille soll möglichst dunkel und das restliche Auge hell sein. Die Farbe der Iris erschwert die Differenzierung zusätzlich, wenn diese recht dunkel ausfällt ist auch der Unterschied zur Pupille entsprechend gering in den Grauwerten. Außerdem ist das Erkennen der Pupille bei sehr kleinen Bildern schwierig bis unmöglich wodurch auf der Iris gerechnet werden muss, und daher diese weiterhin erhalten bleiben sollte.

Nach der Umwandlung wird für die Anwendung das Graubild noch normiert, damit Mindestens ein schwarzes und ein weißes Pixel vorhanden ist.

#### 3.5.1 Gleam-Verfahren

Bei dem Gleam-Verfahren wird jede Farbe (Rot, Gelb und Grün) gleich stark bewertet allerdings wird jeder Farbwert mittels einer Gamma-Korrektur verbessert und das Bild wirkt heller als bei dem Luminance-Verfahren, siehe Abbildung 3.22.

Durch die Gamma-Korrektur wird vor allem der helle Bereich weiter erhöht, somit wird der Farbunterschied zwischen Iris und Auge vermindert, wodurch die Pupille der einzige dunkle Bereich wird. Allerdings wird auch dieser Farbwert erhöht und sollte die Pupille nicht schwarz sein, sie eher ins Graue überführt wird.

Dieses Verfahren wurde gewählt, da es im Vergleich zu den anderen Verfahren im Test von „Color-to-Grayscale: Does the Method Matter in Image Recognition?“[CK12] am besten abgeschnitten hat.

$$G_{Gleam} = \frac{R^{\frac{1}{2.2}} + G^{\frac{1}{2.2}} + B^{\frac{1}{2.2}}}{3}$$

#### 3.5.2 Gleam-New-Verfahren

Dies ist eine Variante von Gleam bei dem zuerst das gesamte Bild analysiert wird um die Parameter für die jeweilige Gamma-Korrektur zu ermitteln. Dies ist etwas aufwendiger, aber für die kleinen Bilder hinnehmbar.

Durch die individuelle Veränderung der Farbkanäle, werden Farbunterschiede minimiert und somit alle stark farbigen Bereiche ebenfalls dunkel dargestellt. Der Kontrast zwischen der farbigen Iris und dem weißen Auge wird verbessert, siehe Abbildung 3.23.

Da allerdings alle Farben dunkel werden, entstehen weitere dunkle Bereiche die die Detektion der Pupille beeinträchtigen können.

$$G_{GleamNew} = \frac{R^r + G^g + B^b}{3}$$

Wobei gilt  $\{r, g, b\} = \frac{\log(V_{\max})}{\log(\{R, G, B\}_{\max})}$  mit  $V_{\max}$  als maximal möglicher Farbwert und  $R_{\max}$  als maximal Vorhandener Rot-Farbwert,  $G_{\max}$  und  $B_{\max}$  äquivalent.

### 3.5.3 Luminance-Verfahren

Dies ist ein lineares Verfahren, das der menschlichen Farbwahrnehmung entspricht und oft der Standard bei der Umwandlung von Farbbild nach Graubild darstellt. Eine Gamma-Korrektur wird bei der Umwandlung nicht verwendet.

Somit entsteht ein natürlicher Farbverlauf, bei dem der Farbunterschied zwischen Pupille, Iris und Auge auf einem mittleren Niveau bleibt, siehe Abbildung 3.24.

$$G_{Luminance} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

### 3.5.4 Min-Max-Verfahren

Dabei handelt es sich eigentlich um zwei verschiedene Varianten, allerdings funktionieren beide nach dem selben Prinzip, als Grauwert wird der jeweilige Extremwert aus den einzelnen Farbkanälen gewählt.

Durch Verwendung der Extremwerte, wird das gesamte Bild deutlich heller bzw. dunkler und von Relevanz ist nur noch der Wert, nicht die eigentliche Farbe.

Bei dem Max-Verfahren werden alle farbigen und helle Bereiche hell dargestellt und nur gleichmäßig dunkel Bereiche bleiben dunkel wie es bei schwarz der Fall ist. Wenn der Minimalwert anstelle verwendet wird, bleiben nur gleichmäßig helle Bereiche hell, alles anderen werden abgedunkelt.

$$\begin{aligned} G_{\max} &= \max(R, G, B) \\ G_{\min} &= \min(R, G, B) \end{aligned}$$

### 3.5.5 Quadrat-Verfahren

Dies ist ein Verfahren, dass das Eingabebild verdunkelt und vom Aufbau dem Inversen von Gleam entspricht. Somit ist das gesamte Bild dunkler als bei dem Luminance-Verfahren, siehe Abbildung 3.26. Durch die Abdunklung werden kleine Farbänderungen in den dunklen Bereichen reduziert, wodurch die Pupille sehr dunkel und der Farbunterschied zur Iris geringer dargestellt wird.

$$G_{Quadrat} = \frac{R^2 + G^2 + B^2}{3}$$

### 3.5.6 Normalisierung von Graubildern

Um ein Graubild zu erhalten, dass das volle Spektrum der möglichen Werte erfüllt, wird das Eingabebild normalisiert. Dazu wird der Maximale  $G_{\max}$  und Minimale  $G_{\min}$  Wert im Bild gesucht. Anschließend wird der neue Grau-Wert  $G_{new}$  wie folgt bestimmt, dabei ist  $V_{\max}$  der maximale mögliche Wert in der Ausgabe und  $G$  der aktuelle Grau-Wert.

$$G_{new} = (G - G_{\min}) \cdot \frac{V_{\max}}{G_{\max} - G_{\min}}$$



Abbildung 3.22: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Glean-Verfahren



Abbildung 3.23: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Gleam-New-Verfahren

Da für die Anwendung ein Schwarzer Bereich gesucht wird gegen einen Hellen Hintergrund, wird für die Bestimmung der Extremwerte nicht das originale Eingenbild verwendet, sonder ein Gauß-gefiltertes. Dies hat den Vorteil, dass einzelne lokal auftretende Werte, z.B. Reflektionen, nicht als Extremwert verwendet werden, wodurch die Pupille gleichmäßiger dunkler und das gesamte Bild stärker aufgehellt wird.

To Do: Berechnungen des Datensatzes Neu (auch für Video?)

### Auswirkung des Gauß-Filters

Dies ist ein Tiefpassfilter und wird verwendet um das Eingangssignal zu glätten. Dies hat in der Bildverarbeitung den Effekt, dass Details im Bild verschwinden und das Bild unscharf wird.

Die einzelnen Werte werden ihrer Umgebung angepasst, wodurch lokal auftretende Extremwerte verschwinden bzw. abgeschwächt werden.



Abbildung 3.24: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Luminance-Verfahren



Abbildung 3.25: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Extremwert-Verfahren.  
Oben: Max-Verfahren, Unten: Min-Verfahren



Abbildung 3.26: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Quadrat-Verfahren

## 3.6 ElSe

Ellipse Selection for Robust Pupil Detection (ElSe), ist ein Algorithmus zur Bestimmung der Pupille in einem hochauflösenden Aufnahme des Auges unter realen Bedingungen.

### 3.6.1 Aufbereitung der Bildinformation in der Augenregion

Zur Bestimmung der Blickrichtung ist die Augenregion natürlich von besonderer Bedeutung. Aus diesem Grund werden die Landmarks der Augenregion nochmals gesondert betrachtet. Aufgrund der besonderen Bedeutung existiert eine große Anzahl an Algorithmen, die speziell auf eine hochgenaue Bestimmung von Augenmerkmalen optimiert sind, wie Beispielsweise ElSe [WF16], Goutam [GM13], Starburst [DL05], Swirski [SBD12].

Daher bestimmt OpenFace zusätzlich zu den 64 Landmarks, die das Gesicht beschreiben, weitere 28 Landmarks pro Auge, aus denen die Blickrichtung ermittelt wird. Dazu kommt ein weiteres CLNF zum Einsatz, das auf Augen Trainiert wurde. Dabei zeigten die Vorabtests, dass die Detektionsgenauigkeit bei den getesteten kleinen Gesichtern unzureichend ausfällt.

Um die Position der Landmarks zu verbessern, kann auf dem Bildausschnitt der Augen der ElSe-Algorithmus eingesetzt werden. Dieser Algorithmus arbeitet auf einem Farbbild um so die Umrisse der Pupille zu berechnen. Dieses Verfahren wurde gewählt, da es im Test [WF16] am besten abgeschnitten hat und direkt das Zentrum der Pupille liefert.

Für die Bestimmung der Blickrichtung ist vor allem das Zentrum der Pupille und Iris sowie deren Umrisse ausschlaggebend, daher müssen diese aus dem Ergebnis von ElSe abgeleitet werden.

Der ElSe Algorithmus wurde für Eye-Tracking Brillen entwickelt, die die Augenregion hochauflösend abbilden. Entsprechend nimmt die Detektionsleistung bei niedriger auflösenden Bildern rasch ab und da diese Berechnung unabhängig der Landmarks ausgeführt wird, empfiehlt sich das Ergebnis zu überprüfen, damit die bestimmten Landmarks auch innerhalb der Augenhöhle liegen.

Bei der Berechnung wird jedes Auge unabhängig vom anderen ausgeführt. Durch die Messungenauigkeit und bei nahe an der Person befindlichen Blickzielen können die Blickrichtungen beider Augen verschieden sein. Wird ein weiter entfernter Punkt von beiden Augen fokussiert, so kann die Blickrichtung beider Augen als parallel angenommen werden, da der Unterschied zwischen Beiden minimal ausfällt. Um den Fehler zu minimieren wird als Ergebnis die durchschnittliche Blickrichtung beider Augen verwendet.

### 3.6.2 Beschreibung

Bei realen Aufnahmen sind Bildfehler unvermeidlich, so können Reflektionen (Brille, Kontaktlinse usw.), Make-Up und körperliche Eigenschaften wie Augenfarbe die Detektion erschweren.

Der Ursprüngliche ElSe-Algorithmus ist für Graubilder einer Eye-Tracking-Brille ausgelegt und optimiert. Dies betrifft vor allem die Qualität der Aufnahme im Bezug auf die Auflösung und die Infrarotbeleuchtung des Bildes, zudem ist es auf diesen Bildern zu einer Echtzeitauswertung in der Lage. Die Infrarotbeleuchtung wird verwendet, damit das Auge ausreichend beleuchtet ist ohne den Probanden zu blenden.

Für die Anwendung wurde ElSe angepasst um auf Farbbilder die nach Grau konvertiert wurden, arbeiten zu können. Ziel ist es die Blickrichtung möglichst exakt zu bestimmen, wofür die Landmarks der Pupille ausschlaggebend sind.

Als Ergebnis liefert ElSe eine Ellipse, die den Umriss der Pupille im Bild beschreibt, aus der die Landmark abgeleitet werden können.

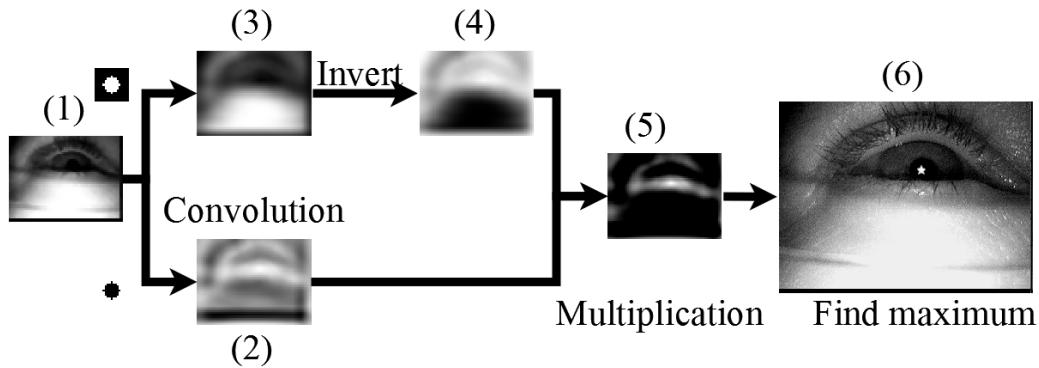


Abbildung 3.27: Ablauf der alternativen Berechnung zur Pupillen-Detektion von [WF16]

### Pupille bestimmen mit Kantendetektion

Da die Pupille als schwarzen Fleck im Bild dargestellt ist und die Iris einen helleren Farbton aufweist, wird ein Kantendetektor verwendet, der alle Pixel markiert, bei denen eine starke Farbänderung auftritt. Bei ElSe wird ein Morphologischen Ansatz eingesetzt, von Relevanz sind nur zusammenhängende Kantenpixel um die Kante zwischen Pupille und Iris zu finden, alle anderen können ignoriert werden. Wobei jedes Kantenpixel als Startpunkt der Berechnung dienen kann.

Um jene Kantenpixel zu erhalten, die die Pupille beschreiben, wird versucht fortlaufende Kanten zu finden, die eine Ellipse bilden. Jene die nicht diesen Anforderung entsprechen, können recht schnell ignoriert werden. Anschließend können auch alle offenen Ellipsenverläufe und jene die am meisten vom bestimmten Verlauf abweichen, verworfen werden.

Das beste Ergebnis aller bestimmten Ellipsen wird als Lösung verwendet.

### Grobe Bestimmung der Pupille

Sollte die Bestimmung der Ellipse, wie im letzten Kapitel beschreiben, scheitern, so wird das Zentrum des dunkelsten Kreises ermittelt. So ein Punkt kann immer gefunden werden, ist aber nicht zwingend die Pupille.

Auf einem verkleinerten Bild Abbildung 3.27 (1) wird ein kreisförmiger Mean-Filter eingesetzt mit Ergebnis in Abbildung 3.27 (3). Zur zweiten Faltung wird der Durchschnitt über ein Quadrat ohne inneren Kreis eingesetzt mit Ergebnis in Abbildung 3.27 (2), wobei bei beiden Kreisen der selbe Radius verwendet wird.

Nun wird das Ergebnis des Quadratischen Mean-Filters invertiert Abbildung 3.27 (4) und mittels Punkt-Multiplikation mit dem anderen Meanfilter zusammengebracht Abbildung 3.27 (5). Im resultierendem Bild wird nun der höchste Wert gesucht, da dies das Zentrum des dunkelsten kreisförmigen Ortes im Bild ist.

Ergebnis des Beispiels ist als Kreuz in Abbildung 3.27 (6) markiert.

### Ergebnisse

Für den Test, wurden Bilder von  $384 \times 288$  Pixel Größe verwendet. Im Vergleich zu den anderen Verfahren, ist ElSe in den meisten Fällen ihnen überlegen, mit einer Verbesserung der Erkennungsrate um 14.53% auf dem verwendeten Datensatz [WF16].

Ein Problem entsteht wenn der Farbunterschied zwischen Iris und Pupille recht gering ausfällt oder durch Reflektionen der Kantenverlauf gestört wird.

Für die Anwendung im aktuellen Fall, ist der Bereich der Augen sehr klein und eine eindeutige Detektion entsprechend schwierig, wodurch vor allem die grobe Bestimmung der Ellipse von Interesse ist.

### 3.6.3 Versuchsaufbau für die Auswirkung der Graubild-Verfahren auf ElSe

Um die einzelnen Verfahren besser vergleichen zu können, wurden künstliche Augen aus dem Datensatz [WBZ<sup>+</sup>15] verwendet damit die exakte Position der Landmarks bekannt ist.

Ein gutes Verfahren muss stabil gegenüber der Skalierung sein, damit es auch auf kleinen Bereichen zuverlässig arbeitet. Da für die spätere Anwendung vor allem das Zentrum der Pupille von Interesse ist, wird der Abstand zum Zentrum als Qualitätsmaß verwendet.

Da ElSe für Eye-Tracking Brillen entwickelt wurde, also für ein Qualitativ hochwertiges Bild eines Auges, wurde der Bildbereich soweit verkleinert das noch alle Landmarks des Auges mit etwas Rand dargestellt wird, um diesen Anforderungen entsprechend nahe zu kommen.

Somit sind die Bildausschnitten im Datensatz auf denen gerechnet wird etwa 64 auf 29 Pixel groß und werden für die Verarbeitung auf eine Breite von 384 Pixeln vergrößert. Somit ergibt sie die Bildgröße für die ElSe entwickelt wurde, da durch die Skalierung allerdings keine zusätzlichen Informationen entstehen, ist vor allem die grobe Bestimmung der Ellipse, beschreiben in Abschnitt 3.6.2, von Interesse. Diese Auswahl des Bildbereiches kann auch in der späteren Anwendung eingesetzt werden, da der Augenbereich durch eigene Landmarks in der Gesichtsanalyse relativ genau bestimmt ist.

Um die Qualität der Berechnung bei verschiedenen Größen zu simulieren, wurde das Bild linear verkleinert.

### 3.6.4 Auswirkung des Radius

Ein wichtiger Parameter des ElSe-Verfahrens ist der Radius des Filters. Um den besten Parameter zu bestimmen wurde der Augen-Datensatz [WBZ<sup>+</sup>15] verwendet und die Augenpartie ausgeschnitten. Im Datensatz besitzen die abgebildeten Augen durchschnittlich 15 Pixel Breite Pupille und eine Iris von 34 Pixel Durchmesser.

In Abbildung 3.30 und Abbildung 3.28 ist zu erkennen, dass der Radius signifikant für die Qualität der Berechnung ist. Da für die spätere Anwendung vor allem das Zentrum der Pupille von Interesse ist, vgl. Abschnitt 3.7.1, muss ElSe in diesem Aspekt zuverlässig Ergebnisse liefern.

Im Versuch hat sich ein Radius von etwa einem Zwölftel des zu erwarteten Durchmesser der Iris bzw. Pupille als sinnvoll erwiesen, um deren Ausmaße möglichst exakt zu bestimmen. Im Versuch entspricht dies 8 und 18 Pixel. Um die Position des Zentrums der Iris und der Pupille möglichst gut zu bestimmen, erwies sich ein Radius von 10 am besten, siehe Abbildung 3.28, wobei dieser Fehler nicht so sehr steigt bei Veränderung des Radius, als bei der Größenbestimmung von Pupille und Iris.

### 3.6.5 Auswirkung der verschiedenen Graubild-Verfahren

Es zeigt sich das die Verfahren, um den Farbwert in einen Grauwert zu überführen, durchaus Auswirkungen auf die Qualität der Berechnung hat.

Das beste Ergebnis liefert das Gleab-Verfahren (Beschreiben in Unterabschnitt 3.5.1) mit einer Abweichung von 5.89 Pixeln, siehe Abbildung 3.28, da die Abweichung vom Zentrum minimal ist. Ein mittleres Ergebnis liefert das Luminance-Verfahren, beschreiben in Unterabschnitt 3.5.3, mit welchem eine Abweichung auf dem Augen-Trainingsdatensatz von 6.42 Pixel erreicht wird.

Im Vergleich liefert das Quadratische-Verfahren, beschreiben in Unterabschnitt 3.5.5, die schlechtesten Ergebnis, da die durchschnittliche Abweichung bei 7.23 Pixel liegt.

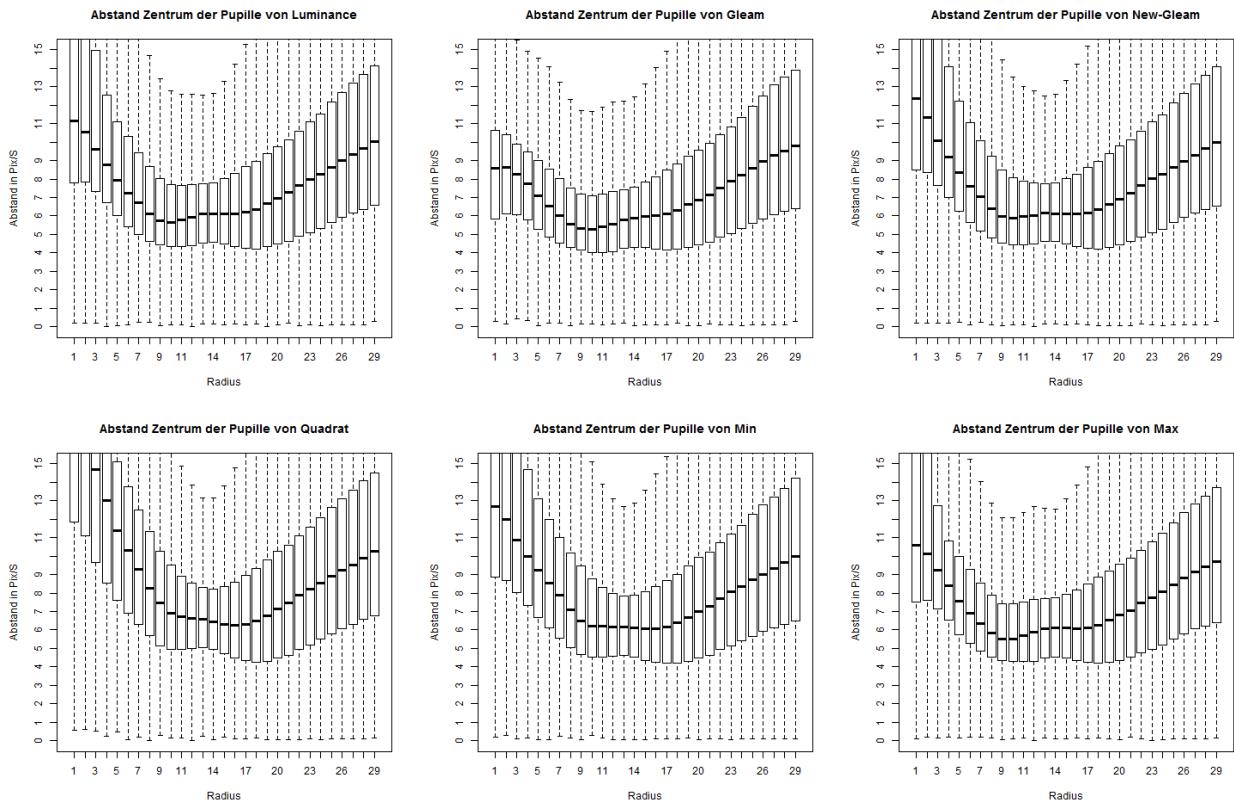


Abbildung 3.28: Abstand des Zentrums der Landmark-Pupille und der berechneten Ellipse in [Pixel/Skalierung]

Oben-Links: Luminance, Oben-Mitte: Gleam, Oben-Rechts: Gleam New, Unten-Links: Quadrat, Unten-Mitte: Min-Wert, Unten-Rechts: Max-Wert

Bei der Berechnung auf verschiedenen groß skalierten Bildern ist die Abweichung von ElSe bei Verwendung von Gleam konstant bei etwa 5.9 Pixel und arbeitet somit stabil, siehe Abbildung 3.31.

### 3.6.6 Vergleich zu OpenFace

Als Referenz wird das Ergebnis von OpenFace, für die zusätzlich bestimmten Landmarks der Augen, verwendet. Dies wurde auch auf dem Augendatensatz [WBZ<sup>+</sup>15] angewendet um vergleichbare Ergebnisse zu erhalten.

In Abbildung 3.32 ist zu erkennen dass dieses Verfahren im Schnitt oft schlechtere Ergebnisse liefert als das Ergebnis von ElSe, allerdings ohne das begehen von großen Fehlern und auch öfters genauere. Da die hohe Qualität von ElSe nur erreicht werden kann, wenn es auf passenden Bildausschnitt angewendet wird, ist auch die Detektion des Auge von Interesse.

Nach Abbildung 3.33 ist zu entnehmen, das der Bereich des Auges zwar nicht so exakt bestimmt wird, allerdings überdeckt er den relevanten Bereich ausreichend genau. Dargestellt sind Koordinaten, X- und Y-Position in Pixel sowie die Ausdehnung der Box (Width und Height) ebenfalls in Pixel relativ zur umschließenden Box der Landmarks. Somit liegen die Landmarks der Augen im Bildausschnitt, wodurch diese Ausschnitt verwendet werden kann als Eingabe von ElSe.

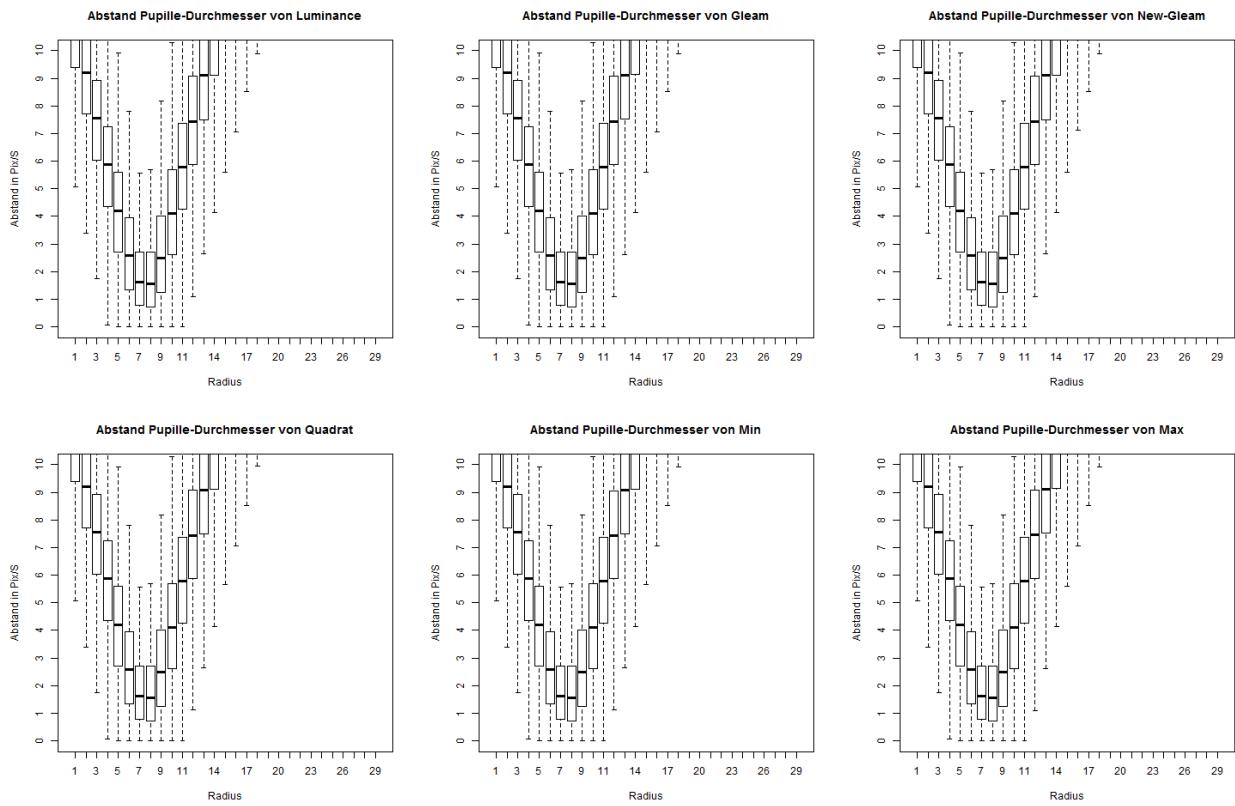


Abbildung 3.29: Unterschied Zwischen den Radien der Landmark-Pupille und der Berechneten Ellipse in [Pixel/Skalierung]  
 Oben-Links: Luminance, Oben-Mitte: Gleam, Oben-Rechts: Gleam New, Unten-Links: Quadrat, Unten-Mitte: Min-Wert, Unten-Rechts: Max-Wert

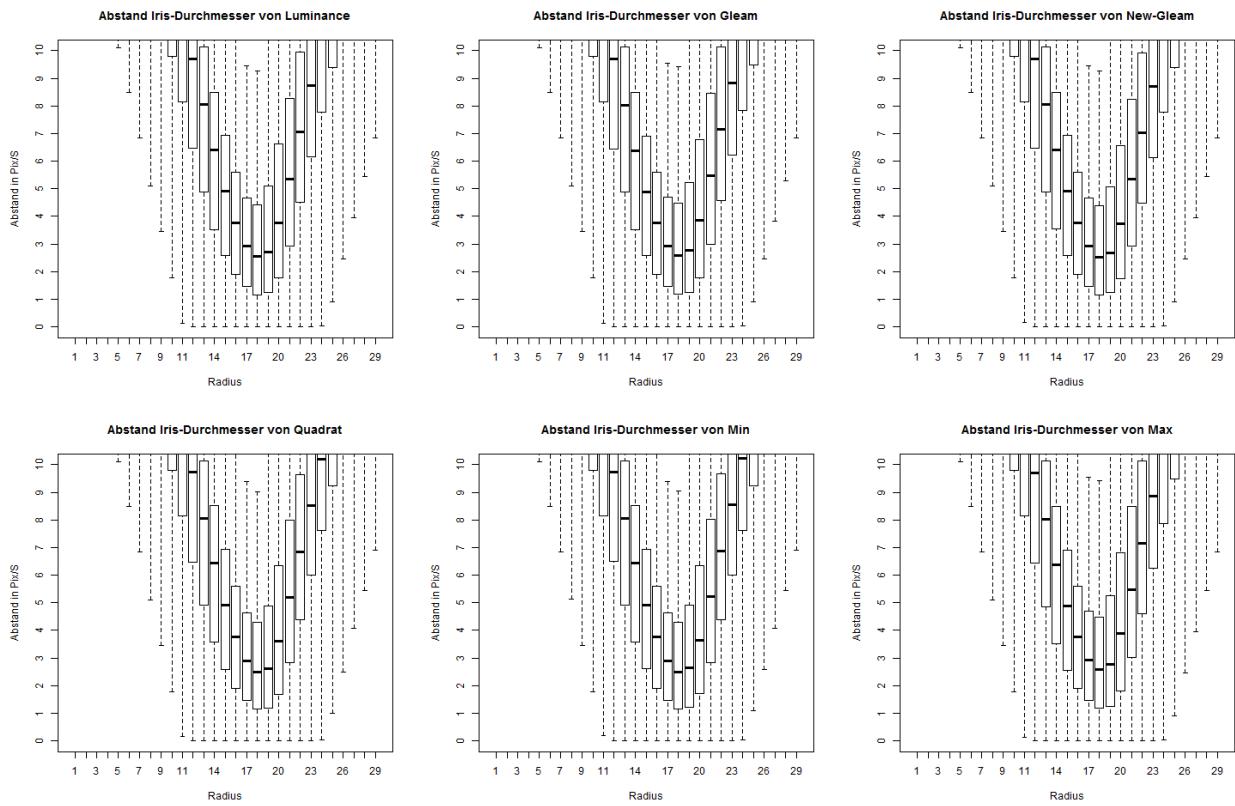


Abbildung 3.30: Unterschied Zwischen den Radien der Landmark-Iris und der Berechneten Ellipse in [Pixel/Skalierung] gegen die Radius-Größe.  
 Oben-Links: Luminance, Oben-Mitte: Gleam, Oben-Rechts: Gleam New, Unten-Links: Quadrat, Unten-Mitte: Min-Wert, Unten-Rechts: Max-Wert

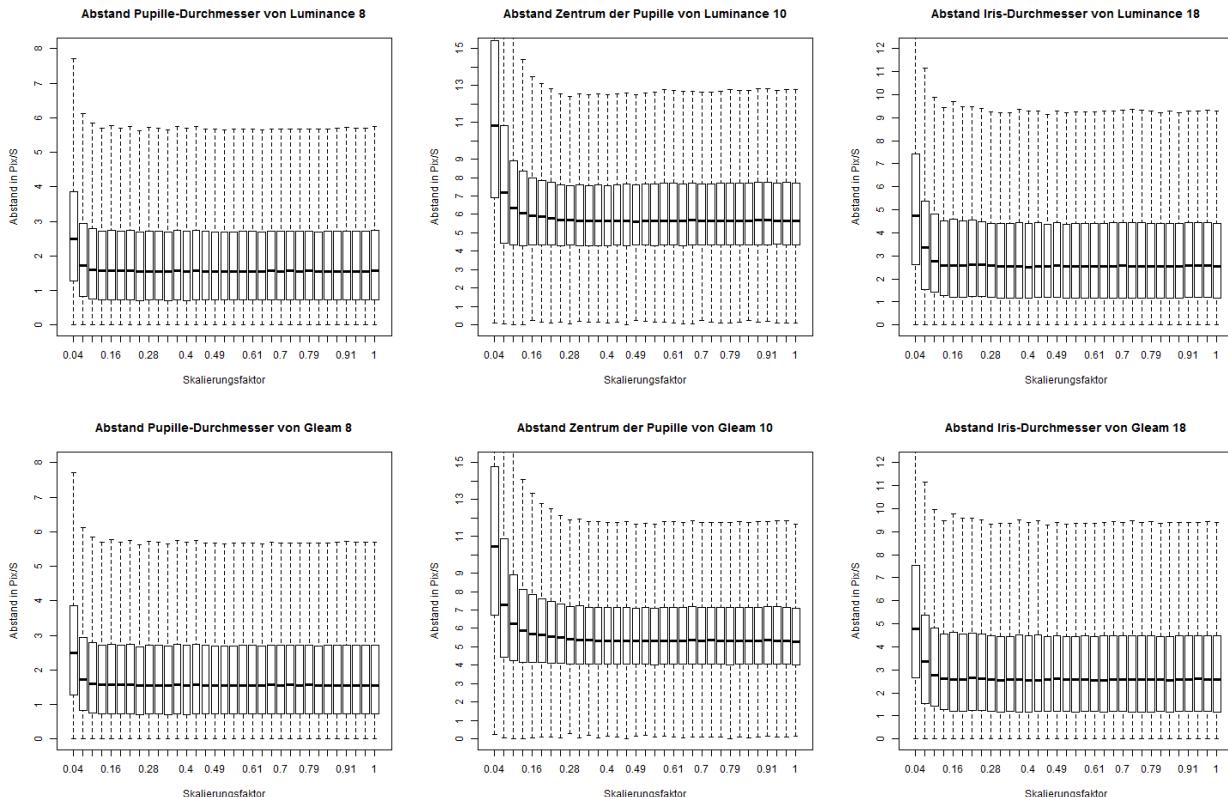


Abbildung 3.31: Auswirkung von der Bildgröße auf die Qualität der Berechnung.  
Oben: Luminance, Unten Gleam

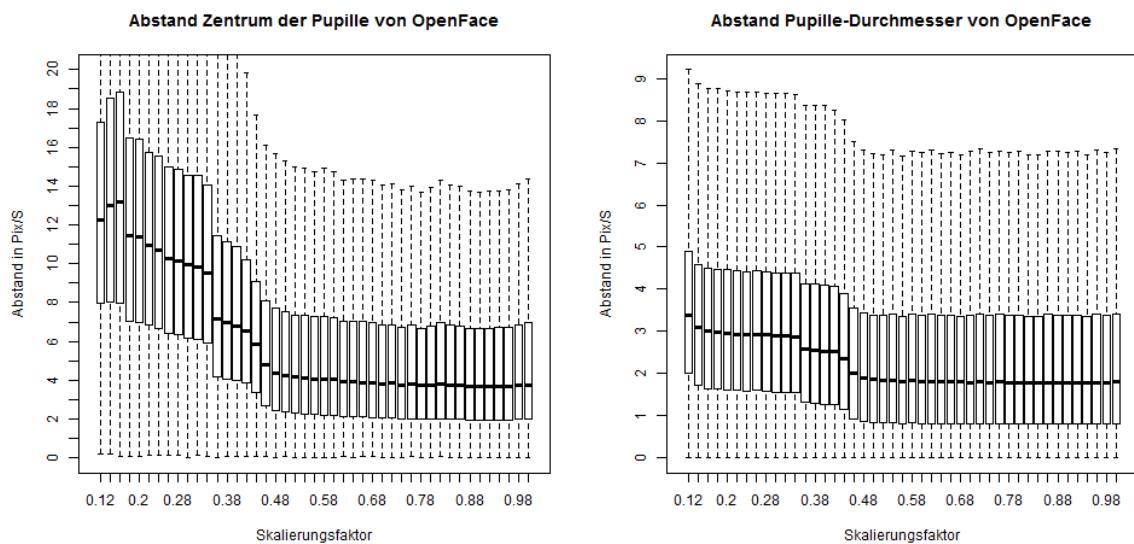


Abbildung 3.32: Auswirkung von Skalierung auf die Qualität der Augendetektion von OpenFace

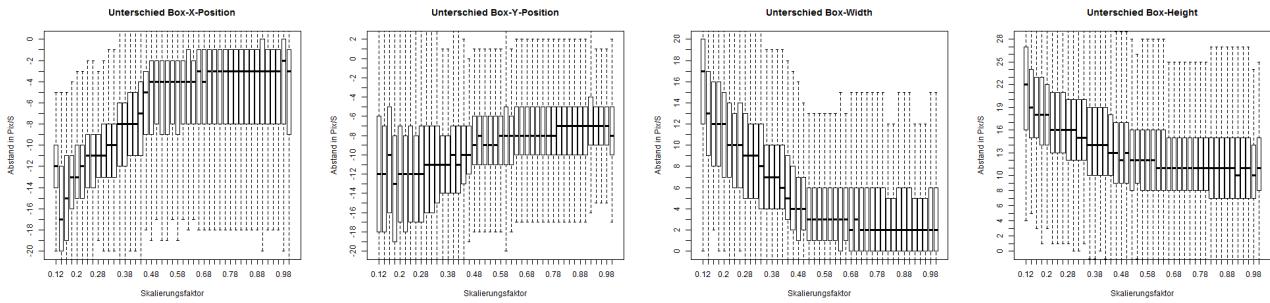


Abbildung 3.33: Bestimmung der Box ums Auge

### 3.6.7 Ergebnis

So ist im Test der Durchschnitt bei allen Skalierungen ElSe den Ergebnisse von OpenFace überlegen, durch die Verteilung ist allerdings eine Kombination beider Verfahren sinnvoll, so kann das Ergebnis von OpenFace bei Bildern in denen die Iris größer als 21 Pixel ist direkt als Lösung verwendet werden, da der mögliche Fehler von OpenFace geringer ist als von ElSe.

Im Bereich zwischen 21 und 15 Pixel können beide Ergebnisse kombiniert werden, da sie ungefähr gleich gute Ergebnisse liefern.

Sollte die Iris im Originalbild noch kleiner sein, so ist ElSe deutlich genauer, da es noch bis zu einer Irisgröße von 3 Pixel noch stabil funktioniert.

## 3.7 Bestimmung des Ziels der Aufmerksamkeit

Um das Ziel der Aufmerksamkeit einer Person zu bestimmen, muss die reale Position ermittelt werden. Die Orientierung des Gesichtes und die Blickrichtung können als Verlauf einer Ursprungsgerade betrachtet werden, mit einem Ursprung an der Position des Gesichtes im Raum.

Ist der Ursprung und die Gerade bekannt, so kann ermittelt werden, ob sie durch bestimmte Punkte im Raum verläuft. Ist dies der Fall, so wird dieser Punkt wahrscheinlich betrachtet und ist Ziel der Aufmerksamkeit.

### 3.7.1 Bestimmung der Position & Orientierung des Gesichts

Zur Bestimmung der Translation und Orientierung des Gesichtes wird ein CLNF bzw. PDM eingesetzt. Dabei wurde es mit der Kameraabbildung von 3D-Landmarks eines normierten Kopfes in verschiedenen Ausrichtungen initialisiert. Das normierte Ergebnis kann mit den passenden Kameraparameter von der Aufnahme angepasst werden um die reale Position und Orientierung zu bestimmen.

### Abschätzen der Kameraparameter

Sind keine Kameraparameter bekannt, so können diese anhand der Bildauflösung grob geschätzt werden. Bei der Schätzung der Brennweite für ein Bild mit einer Dimension  $I_x \times I_y$  wird das Standardobjektiv mit einer Auflösung von  $640 \times 480$  Pixel angenommen, somit ergeben sich die Brennweiten

$f_x$  und  $f_y$  wie folgt:

$$f_x = 500 \cdot \frac{I_x}{640}$$

$$f_y = 500 \cdot \frac{I_y}{480}$$

### Position & Orientierung

Zur Bestimmung der Kopfposition  $P = (X_{avg} \ Y_{avg} \ Z_{avg})^t$  im Kamerakoordinaten wird die Größe, ein Skalierungsfaktor der normierten Kopfgröße  $S_G$ , im Bild verwendet.

Da bei der Abbildung von Welt- nach Bild-Koordinaten gilt:  $x = f \cdot \frac{X}{Z}$  und  $y = f \cdot \frac{Y}{Z}$ , kann die Tiefe wie folgt abgeschätzt werden.

Sei  $P_1 = (X_1 \ Y_1 \ Z_1)^t$ ,  $P_2 = (X_2 \ Y_2 \ Z_2)$  die Beschreibung der Größe  $G$  eines Kopfes mit:

$$a = \frac{\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}}{\frac{|Z_1 - Z_2|}{2}} = \frac{G}{Z_{avg}}$$

$$S = \frac{S_G}{G}$$

$$\Rightarrow a \cdot f = f \cdot \frac{G}{Z_{avg}} = S_G$$

$$Z_{avg} = \frac{f}{S_G} \cdot G = \frac{f}{S}$$

$$X_{avg} = \frac{x \cdot Z_{avg}}{f}$$

$$Y_{avg} = \frac{y \cdot Z_{avg}}{f}$$

Dies beschreibt allerdings nur eine Annäherung an die tatsächliche Position, da die Distanz mit Hilfe einer Durchschnittlichen Kopfgröße geschätzt wird.

[TB16]

### Bestimmung der Blickrichtung

Für möglichst genaue Ergebnisse wird für die Augenpartie ein weiteres CNN eingesetzt das nur auf diesem Bildaufschnitt arbeitet und weitere 28 Landmarks bestimmt. Durch diese werden die Lider, Iris und Pupille dargestellt und für jedes Auge separat bestimmt.

Zur Bestimmung der Blickrichtung wird wie folgt vorgegangen: Zuerst wird der Strahl bestimmt der, ausgehend vom Zentrum der Kamera, durch das Zentrum der Pupille verläuft. Nun wird der Schnittpunkt zwischen diesem Strahl und einer Sphäre bestimmt, die das Auge repräsentiert. Anschließend wird ein Strahl bestimmt der vom Zentrum der Sphäre ausgehend durch den berechneten Schnittpunkt verläuft, dies ist die resultierende Blickrichtung.

### Zusammenhang von Bildposition & Weltposition

Als Ausgangspunkt werden die Ergebnisse des CNN verwendet um die Position zu bestimmen. Zur Bestimmung der Orientierung  $R$  liefert auch das CNN ein Ergebnis  $R_{CNN}$ . Allerdings stimmt es nur im

Zentrum des Bildes, da am Rand immer mehr die Orientierung der einzelnen Pixel mit berücksichtigt werden muss.

$$\begin{aligned}euler_x &= \tan^{-1}\left(\frac{\sqrt{X^2 + Z^2}}{Z^2}\right) \\euler_y &= \tan^{-1}\left(\frac{\sqrt{Y^2 + Z^2}}{Z^2}\right)\end{aligned}$$

$R_{pos} = R(euler_x, euler_y, 0)$  Umwandlung zur Rotationsmatrix

$$R = R_{CNN} \cdot R_{pos}$$

Eine weitere Verbesserung kann erreicht werden, indem die gefundenen 2D-Landmarks mit Hilfe des PDM in 3D zu überführen. Um anschließend die Überführung von 2D nach 3D-Koordinaten erneut zu bestimmen um die Orientierung und Position zu ermitteln. Auch bei diesem Verfahren muss die Pixelorientierung beachtete werden. Allerdings ist auch ein Tiefenbild nötig, da ansonsten die Fehler weiter verstärkt werden. Daher ist es in der aktuellen Anwendung nicht sinnvoll einsetzbar.

### 3.7.2 Größe und Genauigkeit

Um die Qualität auf verschiedenen Distanzen zu ermitteln, wurde der Datensatz Forests for Real Time 3D Face Analysis [FDG<sup>+</sup>13] verwendet, da für jedes Gesicht die Position und Orientierung bekannt ist. Die durchschnittliche Distanz zwischen Kamera und Kopf beträgt ca 70cm bei einer Kopfbreite von 78 Pixel. Um die verschiedenen Distanzen zwischen Probanden und Kamera zu simulieren, wurden die Bilder mit dem angegebene Skalierungsfaktor (X-Achse) verkleinert und mit dem Original verglichen. Da verschiedene Verfahren zur Bestimmung der Position und Orientierung zur Verfügung stehen, sollen diese miteinander verglichen werden. Zur Bestimmung wurde nur das RGB-Bild verwendet und nicht zusätzlich die Tiefeinaufnahme, da dies in der Anwendung auch nicht vorhanden sind.

#### Position

Zur Bestimmung der Position gibt es zwei Verfahren, die direkte mittels Brennweite und Skalierung oder die Überführungsmatrix von 3D zu 2D Landmarks.

Die Funktionen Pose Camera und Pose World (Oben in Abbildung 3.34) verwenden die einfache Bestimmung mittels Skalierung. Dargestellt sind nur die X-Werte, da die Y-Werte eine recht ähnliche Verteilung aufweisen.

Bei den Z-Werten ergibt sich ein etwas anderer Verlauf, bei dem allerdings sie Fehlerquote bei kleinen Bildern gut sichtbar wird, siehe Abbildung 3.35.

Der schnelle Abfall der Genauigkeit ist an der selben Stelle (0.5) an der auch die Detektionsrate stark absinkt, siehe Unterabschnitt 3.4.4. Somit kann das Verfahren bis zu seiner Grenze eingesetzt werden und erst, wenn die Detektion schwierig wird steigt auch der Fehler.

#### Orientierung

Auch bei der Orientierung werden die verschieden Methoden miteinander verglichen. Die Analyse hat gezeigt, dass die Qualität der Verfahren von den einzelnen Rotationen abhängt.

Bei der X-Rotation, dargestellt in Abbildung 3.36 können die rechten Verfahrenen (Pose World und Correct Pose World) überzeugen. Vor allem Pose World hat selbst bei kleinen Abbildungen nur eine mittlere Abweichung von 8.5°

Um die Y-Rotation zu ermitteln ist nun allerdings die linken (Pose Came und Correct Pose Came)

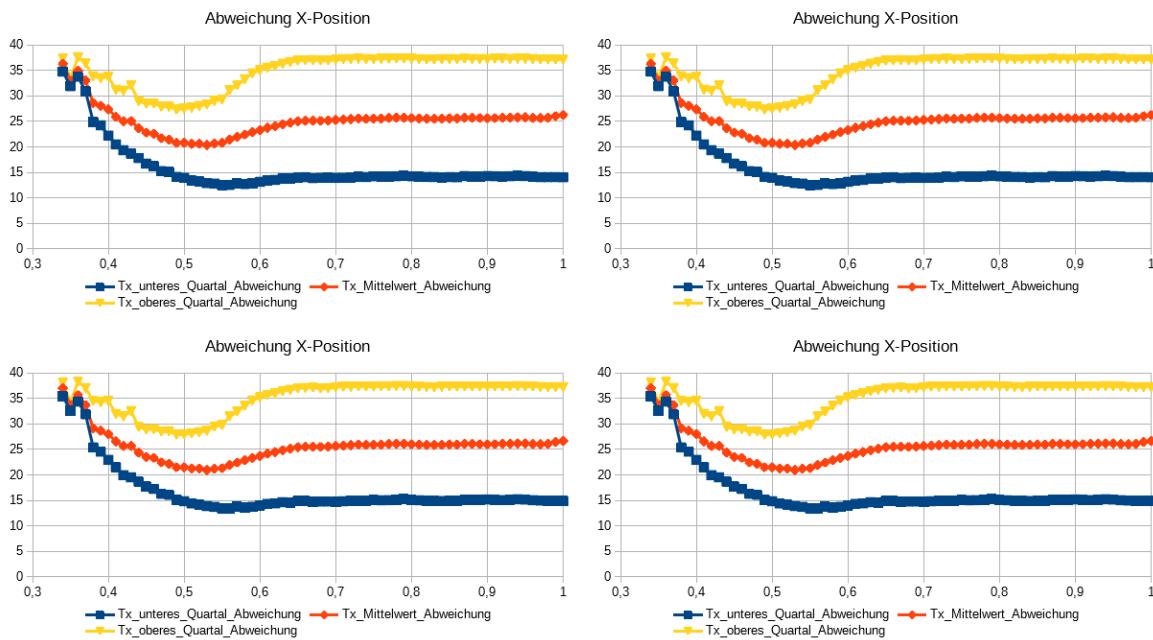


Abbildung 3.34: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist in Millimeter.

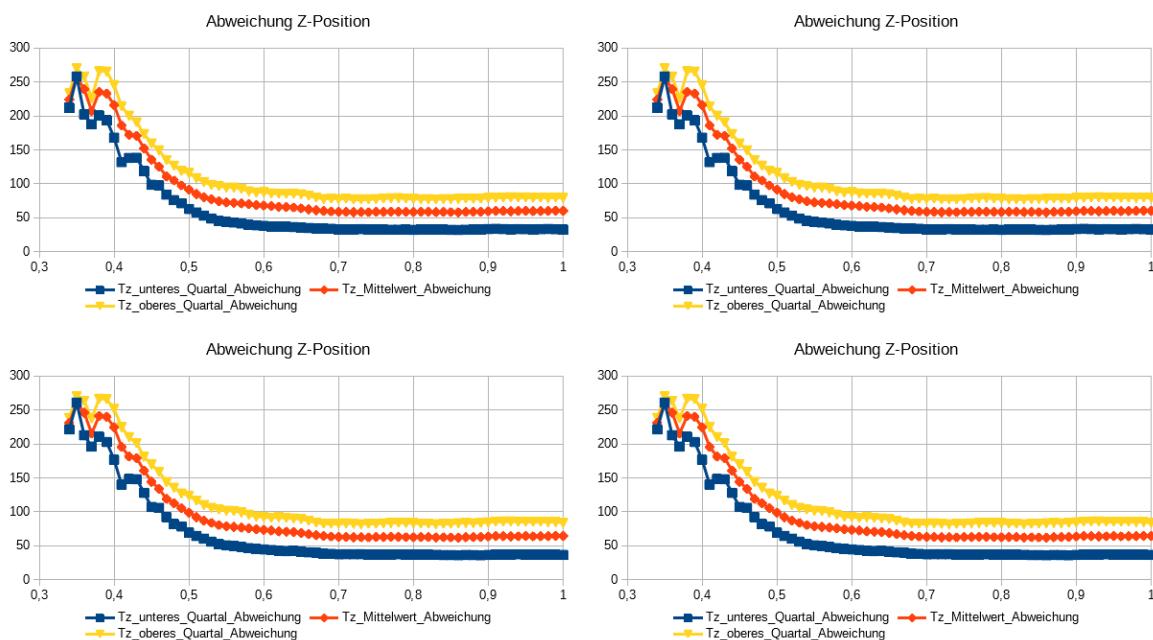


Abbildung 3.35: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist in Millimeter.

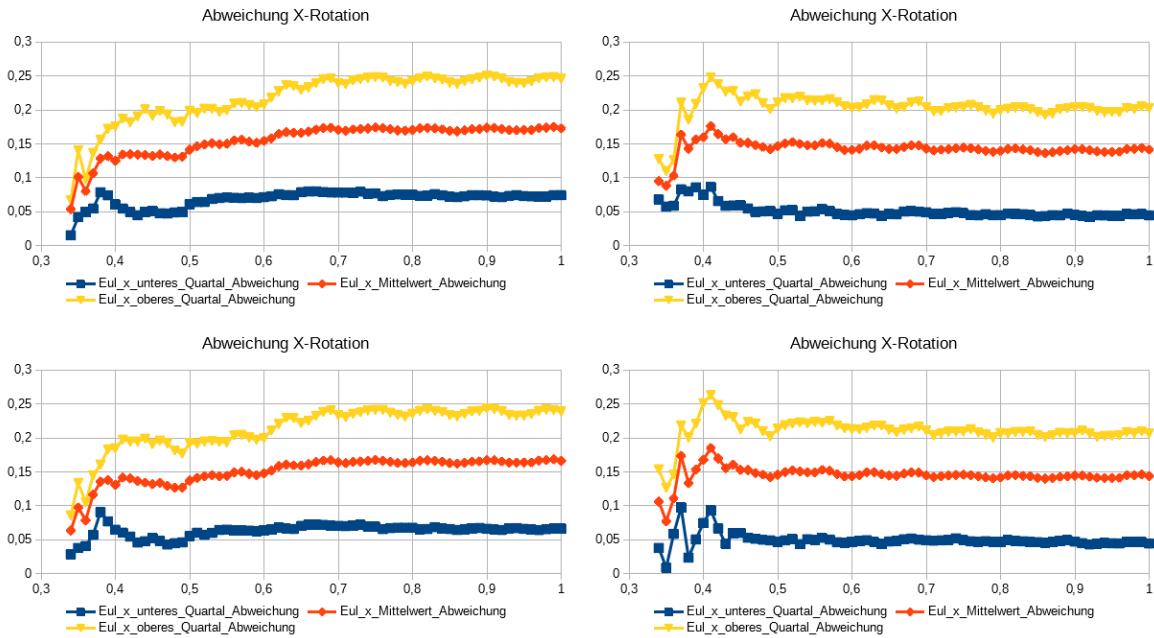


Abbildung 3.36: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist im Bogenmaß.

den rechten (Pose World und Correcht Pose World) deutlich überlegen, siehe Abbildung 3.37. Auch hier liegt der mittlere Fehler über lange Zeit bei etwa  $9^\circ$ .

Bei der Bestimmung von der Z-Rotation sind die Correct Pose Came und Pose Came nahe zu gleich gut, Correkt Pose World allerding schlechter und Pose World besser, siehe Abbildung 3.38. Wobei auffällt, dass Pose World bei Werten unter 0.4 plötzlich der Fehler sehr stark zunimmt.

## Ergebnis

Es zeigt sich, dass Pose World, also die einfache Bestimmung der Position mittels Skalierungsfaktor und zusätzlicher Korrektur der Wikle die besten Ergebnisse liefert.

Die Bestimmung mittels der Überführung von 3D zu 2D Punkten ist nicht notwendig, da ein schlechteres Ergebnis erzielt wurde.

### 3.7.3 Bestimmung eines Punktes, auf der die Aufmerksamkeit liegt

Von Interesse ist vor allem der Punkt auf den der Blick ruht bzw. das Gesicht ausgerichtet ist.  
Bestimmung des Richtungsvektors  $o$  aus der Rotationsmatrix

$$O = R \cdot (0, 0, -1)^T$$

Aus der Blickrichtung mehrerer Probanden kann auch der reale Punkt der Aufmerksamkeit ermittelt werden. Dazu wird die Blickrichtung als Linie  $L_i = s \cdot n_i + p_i$  beschrieben mit  $s \in \mathbb{R}$  und  $n_i, p_i \in \mathbb{R}^3$  verwendet.

$$c = \left( \sum_i I - n_i n_i^T \right)^{-1} \left( \sum_i (I - n_i n_i^T) \cdot p_i \right)$$

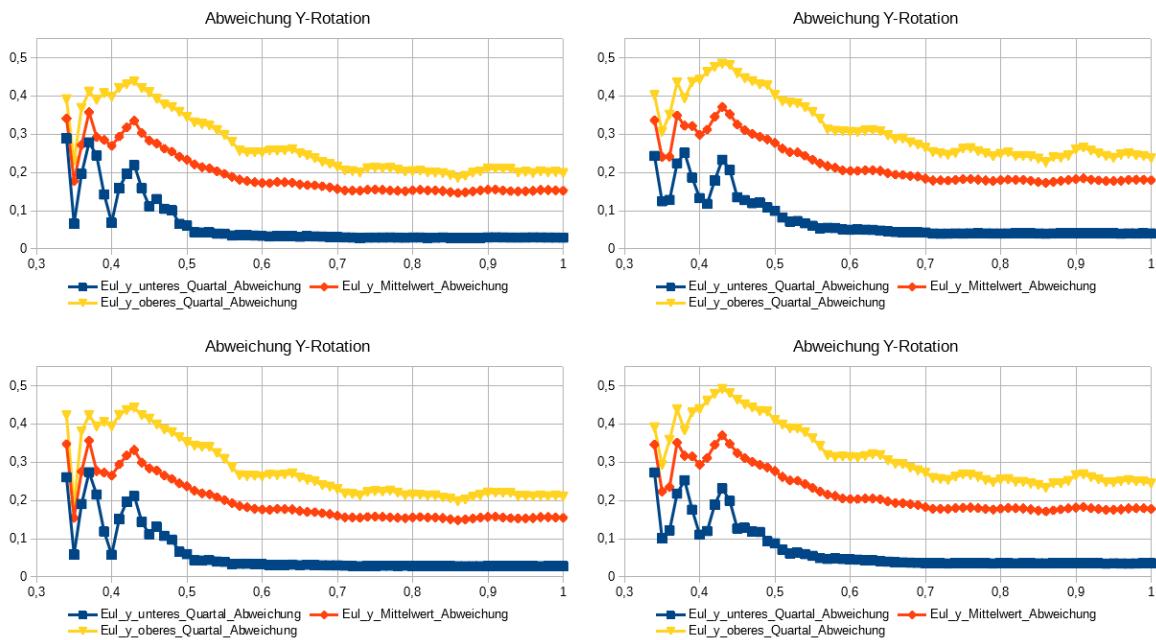


Abbildung 3.37: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist m Bogenmaß.

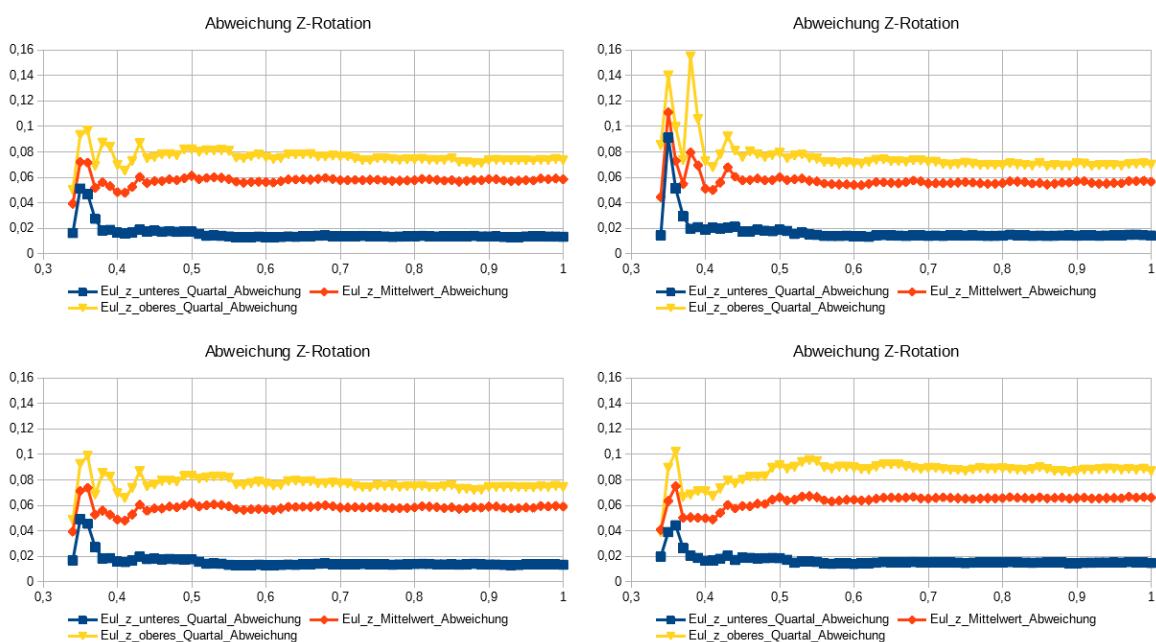


Abbildung 3.38: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist im Bogenmaß.

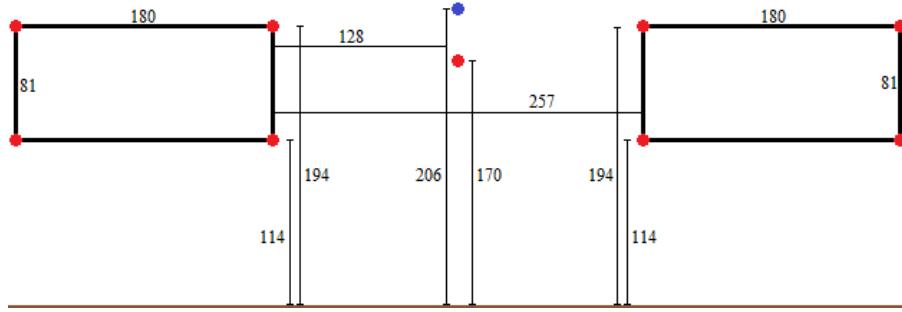


Abbildung 3.39: Aufbau der Targets im Vorversuch, alle Angaben gerundet in Zentimeter  
rote Punkte: Target, blauer Punkt: Kamera

Bei Verwendung der Gesichtsorientierung ergibt sich das Problem den konkreten Blickpunkt zu ermitteln, da die Augenbewegung nicht erfasst werden kann. So muss ein Kegel, der den üblichen Bereich der Augenbewegung umfasst, um die Orientierung berücksichtigt werden als Fehlertoleranz und der gesamte Bereich kommt als Lösungen in Frage. Außerdem liegt der Punkt der Aufmerksamkeit meist außerhalb des Bildbereiches der Kamera und muss entsprechend von einer Anwendung interpretiert werden.

Soll die Position des Ziels auf nahezu parallel verlaufende oder stark verrausche Ergebnisse berechnet werden, so ist die Bestimmung des Schnittpunkts nach dem obigen Verfahren nicht möglich. Eine einfache Variante ist das Verwenden des durchschnittlichen Richtungsvektors  $O_{avg}$  und Position  $P_{avg}$  der Probanden. Die Tiefe  $a$  muss nun geschätzt werden um das Ziel  $P = O \cdot a$  zu bestimmen.

## 3.8 Vorversuche

Um einen Eindruck über den zu erwartenden Datensatz und Schwierigkeiten zu erhalten, wurde einige Testdatensätze mit der Actioncam erstellt.

### 3.8.1 Arbeitsbereich der Verfahren - Versuch 1

Mit diesem Versuch soll der Zusammenhang zwischen Standort der Probanden und Targets untersucht werden. Dazu wird ein Klassenzimmer simuliert mit weit verteilten Schülern, die den gesamten frontalen Betrachten.

#### Versuchsaufbau

In einem Raum wurde die Kamera in 2.06m Höhe 31cm hinter den Targets so montiert, das der gesamte Raum im Fokus liegt. Als Targets wurden 9 Punkte auf einer Ebene markiert mit der Kamera im Zentrum. Die Anordnung der Targets ist in Abbildung 3.39 dargestellt.

Als Position wurde ein Rastfeld mit 1m Kantenlänge im Raum eingezeichnet auf einer Fläche von 7x11m. Die Probanden stellten sich auf diesen Positionen um nacheinander alle Targets zu betrachten.

#### Detektion mit MTCNN

Um die Detektionswahrscheinlichkeit der MTCNN-Face Detektor zu testen wurden dieses Videos analysiert.

+5m						
+4m						
+3m						
+2m	/					
+1m						
	-2m	-1m	Kamera	+1m	+2m	+3m

Abbildung 3.40: Dargestellt ist der horizontale Winkebereich in dem mit der Image-Verarbeitung ein Gesicht erkannt wurden.

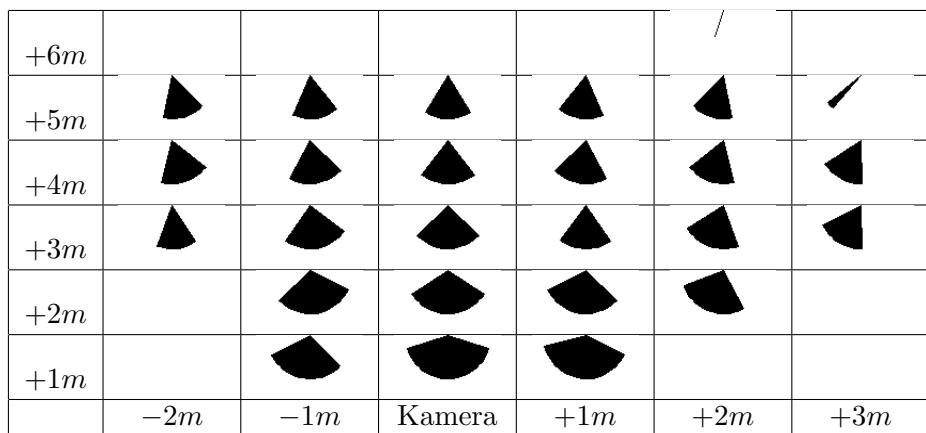


Abbildung 3.41: Dargestellt ist der horizontale Winkebereich in dem mit der Video-Verarbeitung ein Gesicht erkannt wurde.

Es zeigt sich, das auf allen Positionen die Probanden erfolgreich erkannt wurden und die Boxen das Gesicht recht gut beschreibt. Allerdings ist zu erkennen, das die Landmarks unzureichend genau sind. Sie sollten die Mundwinkel, Nasenspitze und beide Augen markieren, liegen aber schon bei recht großen Bildern weit daneben, siehe Abbildung 3.3

## Auswertung

Für die Analyse wurde aus dem Video jene Frames ausgewählt in denen ein Target fokussiert wurde und analysiert.

Für eine Analyse wurde zuerst die Einzelbildauswertung von OpenFace auf die Frames angewendet und jene resultierende Kopfrotationen markiert, an denen eine erfolgreich ein Gesicht erkannt wurde. In Abbildung 3.40 ist der horizontale Wertebereich dargestellt in denen an der jeweiligen Position ein Gesicht erfolgreich erkannt wurde.

Im zweiten Teil wurden die selben Frames für die Messung verwendet, dieses mal allerdings wurde das gesamte Video analysiert. Der Winkelbereich in denen auf der horizontalen Achse an den entsprechenden Positionen ein Gesicht erkannt wurde, ist in Abbildung 3.41 dargestellt.

Das Fehlen von Ergebnissen in Spalte  $-3m$  liegt an der unzureichenden Detektion. Als Ursache kann die Überbeleuchtung durch das einfallende Licht der Fenster angenommen werden.

## Ergebnis

Es zeigt sich, dass eine Auswertung auf einem Video deutlich zuverlässiger arbeitet als auf Einzelbilder, vor allem der größere Rotationsbereich ist von Vorteil.

Durch die Verwendung des Weitwinkelobjektivs, kann die gesamte Breite eines Klassenzimmers erfasse werden und der Winkelbereich für eine erfolgreiche Detektion ist breit genug um Schüler erfassen zu können, sie selbst die fordern Eckpunkte eines Klassenzimmers betrachten.

Bei der Distanz zur Kamera (Tiefe) ist Handlungsbedarf, als Ziel wurde *8m* angesetzt und das aktuelle Verfahren endet bei *5m*.

Eine signifikante Aussage bezüglich des vertikalen Winkel kann aus diesem Aufbau nicht getroffen werden, da die Neigungswinkel zu ähnlich ausfallen bei stehenden Personen und beide einem geradeaus Blick ähneln.

### 3.8.2 Arbeitsbereich der Verfahren - Versuch 2

Da ein aufmerksamer Schüler durchaus auch auf den Tisch blicken kann, z.B. beim Schreiben, so soll getestet werden wie weit die Analyse in solchen Situationen funktioniert.

#### Versuchsaufbau

Für diesen Versuch wurde die Kamera auf *1.88m* Höhe und *3m* vor den vordersten Standort der Probanden aufgestellt.

Als Standorte wurde eine Markierung mit einem Meter Abstand zueinander auf eine Gerade bei *3m* und *9m* verwendet.

Als Target diente die Kamera, ein Punkt *78cm* unterhalb der Kamera und einer *40cm* über dem Boden und *50cm* vor der Kamera. Alle anderen Targets befinden sich *1m* vor den Standorten.

Diesmal war das Versuchsgelände draußen an einem bedeckten Tag, wodurch eine helle schattenlose Szene entsteht.

#### Auswertung

ToDo

#### Ergebnisse

Es zeigt sich, dass eine Videoanalyse auch bei starke Neigung nach unten möglich ist. Die Einzelbildauswertung liefert erneut deutlich schlechter Ergebnisse als des Videos.

Dabei funktioniert das Traking nur, wenn die Versuchsperson zuerst in die Kamera geschaut hat, um es zu beginnen. Auch die stärkere gleichmäßige Beleuchtung ist hilfreich, da die Problematik mit Gegenlicht und Schatten entsteht.

### 3.8.3 Auswertung der Augenpartie - Versuch 3

Um einen Eindruck von ElSe zu erhalten mit hochauflösenden Aufnahmen, wurde mit einer Fotokamera (Sony ILCE-6000, Farbbild  $6000 \times 4000$  Pixel, Brennweite *16mm*) an den selben Positionen wie in Versuch 1 ein weiterer Datensatz von Einzelbilder erstellt, dabei wurden nur Aufnahmen mit der Kamera als Target gemacht. Von Interesse ist die Augenpartie vor allem OpenFace Eye-Detektor im Vergleich zu ElSe. Dabei wurde ElSe in der Basis Konfiguration eingesetzt, dies bedeutet das Luminance-Verfahren, siehe Unterabschnitt 3.5.3 als Graukonvertierer und einem Radius der Maske von 12 Pixel.

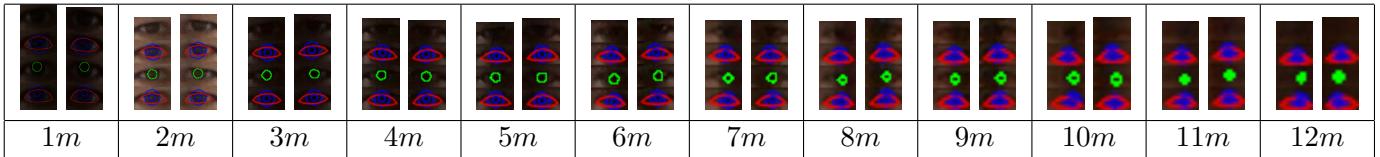


Abbildung 3.42: Dargestellt sind die Ergebnisse von OpenFace und ElSe abhängig von der Distanz.

Von Oben nach Unten: Augenpartie, Ergebnis OpenFace, Ergebnis ElSe, ElSe Ergebnis als Landmarks

## Auswertung

Für die Analyse wurde zuerst mit OpenFace das Gesicht soweit analysiert um die Augenpartie zu bestimmen. Ganz oben in Abbildung 3.42 ist die Augenpartie dargestellt, darunter das Ergebnis von OpenFace mit seinen zusätzlichen 28 Landmarks des Auges. Darunter die berechnete Ellipse auf diesem Bildausschnitt von ElSe in grün. Im untersten Bild wurde aus der berechneten Ellipse von ElSe die Landmarks der Pupille und Iris abgeleitet. Die einzelnen Augenpaare stammen von der selben Person, die sich bei der angegebenen Distanz befand.

## Ergebnis

Es zeigen sich zwei Problematiken, die schnelle Abnahme an Bildinformationen trotz hoher Auflösung, so wie die deutliche Einschränkung der Detektionswahrscheinlichkeit auf Einzelbilder.

Als Ergebnis ist zu erkennen, dass ElSe oft bessere Ergebnisse liefert als OpenFace, wobei auch grobe Fehler auftreten können.

### 3.8.4 Ergebnis der Vorversuche

Es zeigt sich, dass der Arbeitsbereich in Hinblick auf Rotationen ausreichend ist um alle üblichen Bewegungen eines Schülers zu erfassen. Auch die Fläche auf dem sich die Schüler verteilen können ist vielversprechend, nur die Distanz muss noch verbessert werden.

Auch MTCNN-Face ist als Detektor geeignet, er findet zuverlässige alle Gesichter im Frame, unabhängig ihrer Größe und Orientierung. Sogar jene die von OpenFace auch bei der Videoanalyse nicht verwendbar sind. Einzige Anmerkung ist die etwas ungenaue Box, dies kann aber mit einer einfachen Verschiebung der Boxränder korrigiert werden.

## 3.9 Aufmerksamkeitsmessung - Versuch

Für den Versuch wurde ein Video verwendet, welches ein bewegtes Kreuz zeigt, das als Ziel der Aufmerksamkeit dient. Dieses Kreuz sollten die Probanden normal im Auge behalten, damit für jeden Zeitpunkt bekannt ist wo das Ziel der Aufmerksamkeit liegt.

Die Anordnung der Eckpunkte des bewegten Ziels sind in Abbildung 3.44 dargestellt und wurden mittels eines Projektors auf eine Größe von  $2.88 \times 1.49m$  gebracht.

Das Ziel welches betrachtet werden soll (Target) beginnt immer in der Mitte und bleibt dort 1s stehen, bewegt sich innerhalb von 4 Sekunden zu einem der Randpunkte, verweilt dort für eine Sekunde und begibt sich in 4s zu einem nächstgelegenen Randpunkt, bleibt dort 1s und geht zurück zum Zentrum, dies wiederholt sich für alle Eckpunkte. Ein gesamter Durchlauf dauert 2min und 1s.

Die Versuchspersonen befinden sich etwa 1.5m vor der Leinwand, die Kamera befand sich 24cm unter-



Abbildung 3.43: Foto der Versuchsdurchführung

halb und 12.5cm vor dem zentralen Punkt des Targets mit Blickrichtung zum Projektor und Personen, siehe Abbildung 3.43.

### 3.9.1 Versuchsdurchführung

Um die ungefähre Position des Kopfes relativ zur Leinwand zu bestimmen, wurde die Distanz zwischen der Stirn am Nasenrücken und den 4 Eckpunkten durch einen Laserdistanzmessers bestimmt und trianguliert. Während der Aufnahme wurde auf weitere Messung der exakten Position verzichtet. Die 6 Probanden (5 Männlich, 1 Weiblich, 3 mit Brille und 5 ohne Brille) verfolgten das Ziel natürliche Weise.

Um die Bewegung des Targets mit der Aufzeichnung der Kopfbewegung zu synchronisieren, war im Kamerabild der duplizierte Bildschirm zum Projektorbild zusehen.

Die Aufnahmen wurden mit der Logitech-Webcam Abschnitt 2.1 erstellt.

#### Erster Eindruck

Dargestellt in Abbildung 3.45 sind alle Auftreffpunkte der Blickrichtung auf die Leinwand während der gesamten Aufnahme.

Es ist zu erkennen, dass die eigentlichen Kopfbewegungen sichtbar sind, es aber vor allem in den Randbereichen zu einer großen Differenz kommt.

#### Qualität

Durch die begrenzte Auflösung der Kamera und dem großen Distanzbereich auf dem gearbeitet werden muss, ist vor allem die Stabilität bezüglich Skalierung wichtig.

Bei der Bestimmung des horizontalen Winkels der Kopforientierung zeigt sich das die berechneten Werte im Schnitt etwas zu gering ausfallen. Die Orientierung in Richtung Kamera kann zuverlässig

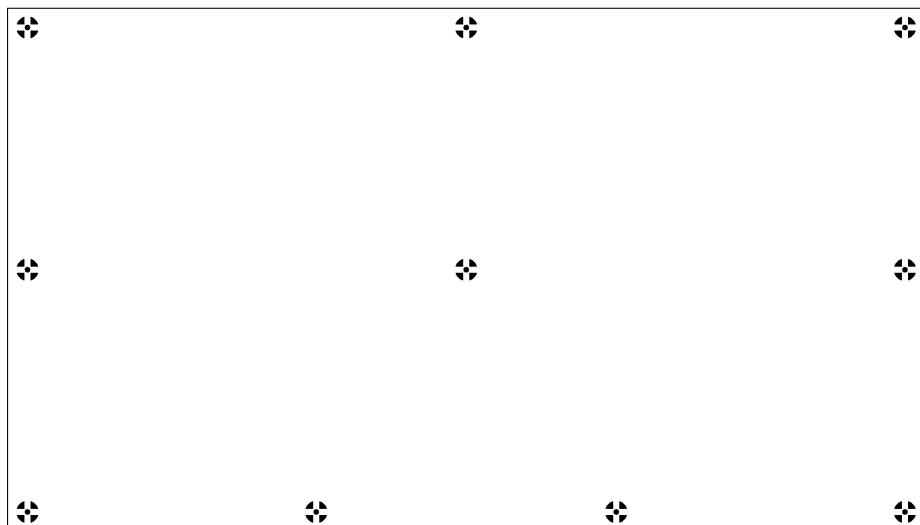


Abbildung 3.44: Eckpositionen des Bewegten Ziels bei der Videoaufnahme

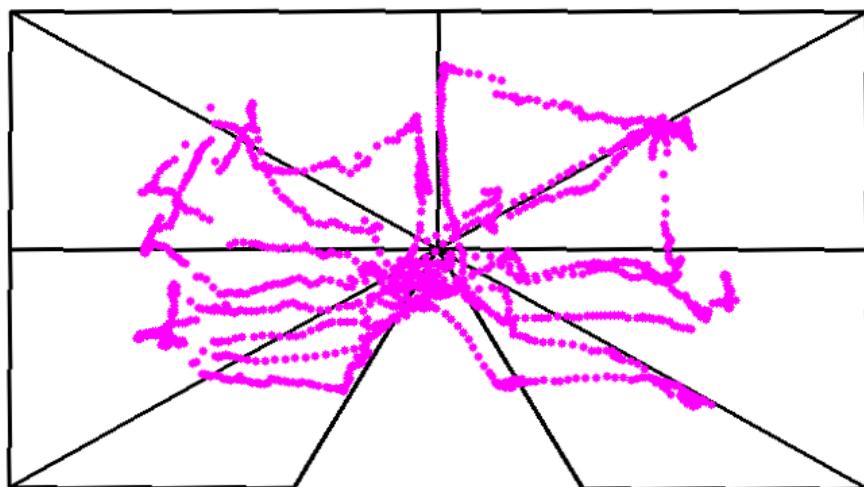


Abbildung 3.45: Dargestellt sind alle gemessene Auftreffpunkte der Gesichtsorientierung auf die Leinwand (Rosa) und des Targets (Schwarz)

bestimmt werden, weben so wenn der Proband seinen Kopf in eine Richtung dreht. Dabei wird der Fehler um so stärker je größer der zu messende Winkel wird. Betrachtet man in der Originalgröße die jeweiligen Quartale (Abbildung 3.46), so sind diese etwa  $5^\circ$  auseinander. Genug um einzelne Bereiche differenzieren zu können.

Bei der Bestimmung des vertikalen Winkels zeigt sich, das dieser Wert nur sehr ungenau bestimmt werden konnte, vor allem der Winkel nach Oben ist fast nicht messbar. Jener Richtung Boden wird besser erfasst, allerdings ist, bedingt durch den Versuchsausbau, der Wertebereich recht gering.

Die bestimmte Blickrichtung ist trotz Verbesserung durch ElSe und Mittlung beider Augen, schon in der Originalgröße nur begrenzt verwendbar. Die Mittelwerte liegen selbst bei den Maximal Werten sehr eng beieinander und die Bereiche überschneiden sich stark. Die Differenz der Mittelwerte zwischen den Extremar sind nur etwa  $20^\circ$  weit auseinander, dabei liegen diese Punkte im Original etwa  $90^\circ$  weit auseinander.

Die Auswirkung der Skalierung ist hinnehmbar gering, allgemein steigt die Abweichung und der Bereich einer erfolgreichen Detektion sinkt. Bei einem Skalierungsfaktor von 0.01 können die einzelnen Bereiche noch gut getrennt werden, siehe Abbildung 3.46, dies entspricht eine Distanz von etwa 14m. Auf der horizontalen Achse liegt der Abstand der Quartale etwa  $9^\circ$  weit auseinander, nur  $4^\circ$  mehr als im Original. Bei der Bestimmung des vertikalen Winkels ergibt sich ein ähnliches Verhalten, wobei vor allem der Wertebereich auf  $30^\circ$  sinkt.

Das Ergebnis der Blickrichtung kann bei der 0.01 Skalierung nicht verwendet werden, da die Differenz zwischen dem Rechten und Linken Maximalwert nur  $8^\circ$  beträgt und die Quartale sich fast vollständig überschneiden.

Überraschend ist das Ergebnis bei dem Skalierungsfaktor von 0.05 (ca 24m). Die Ausrichtungen sind, zumindest horizontal, noch erkennbar und soweit differenzierbar um grobe Richtungsänderungen zu erkennen. Allerdings ist die Detektionsrate sehr gering und kann als Obergrenze angenommen werden. Die Auswertung des Versuches hat die Erwartungen und Problematiken aus den Vorversuchen bestätigt. Eine Verarbeitung des Videomaterials ist sogar bei sehr niedriger Auflösung noch möglich, wobei die Qualität besser sein könnte.

### 3.9.2 Fehleranalyse des Versuches

Eine Betrachtung der Fehlerquellen die bei der Messung entstanden sind bzw. die durch den Aufbau entstehen, sowie bei der Berechnung.

Da nur der Unterschied zwischen Target und Auftreffpunkt der gemessenen Gesichtsorientierung aufgezeigt werden kann, kommt es zu verschiedenen Fehlern, vor allem wird das Target mit den Augen gefolgt. So wird zu Beginn der Bewegung, dem Target nur mit den Augen gefolgt wird, bis sich der Kopf in Bewegung setzt. Dies wird so lange fortgeführt, bis die Kopfdehnung unangenehm und das Ende der Bewegung absehbar wird. So wird der letzte Teil der Bewegungen nur noch von den Augen verfolgt.

#### Messung

Die erste Ungenauigkeit liegt bei der Distanz zur Leinwand, diese wurde nur vor der eigentlichen Aufnahme bestimmt. Somit entsteht eine Abweichung, da die Kopfbewegung während der Aufnahme nicht erfasst wird.

Die eigentliche Messung der Distanz vom Kopf der Personen zur Leinwand ist ebenfalls ungenau, da sie eine Abweichung von etwa 1cm in alle Richtungen abweicht. Außerdem liegt der Ursprung des Kopfes in der Anwendung etwas tiefer und weiter hinten als der ausgemessene Nasenrücken.

Auch die Parameter für der Überführungsmatrix von Welt- nach Kamerakoordinaten sowie die Brenn-

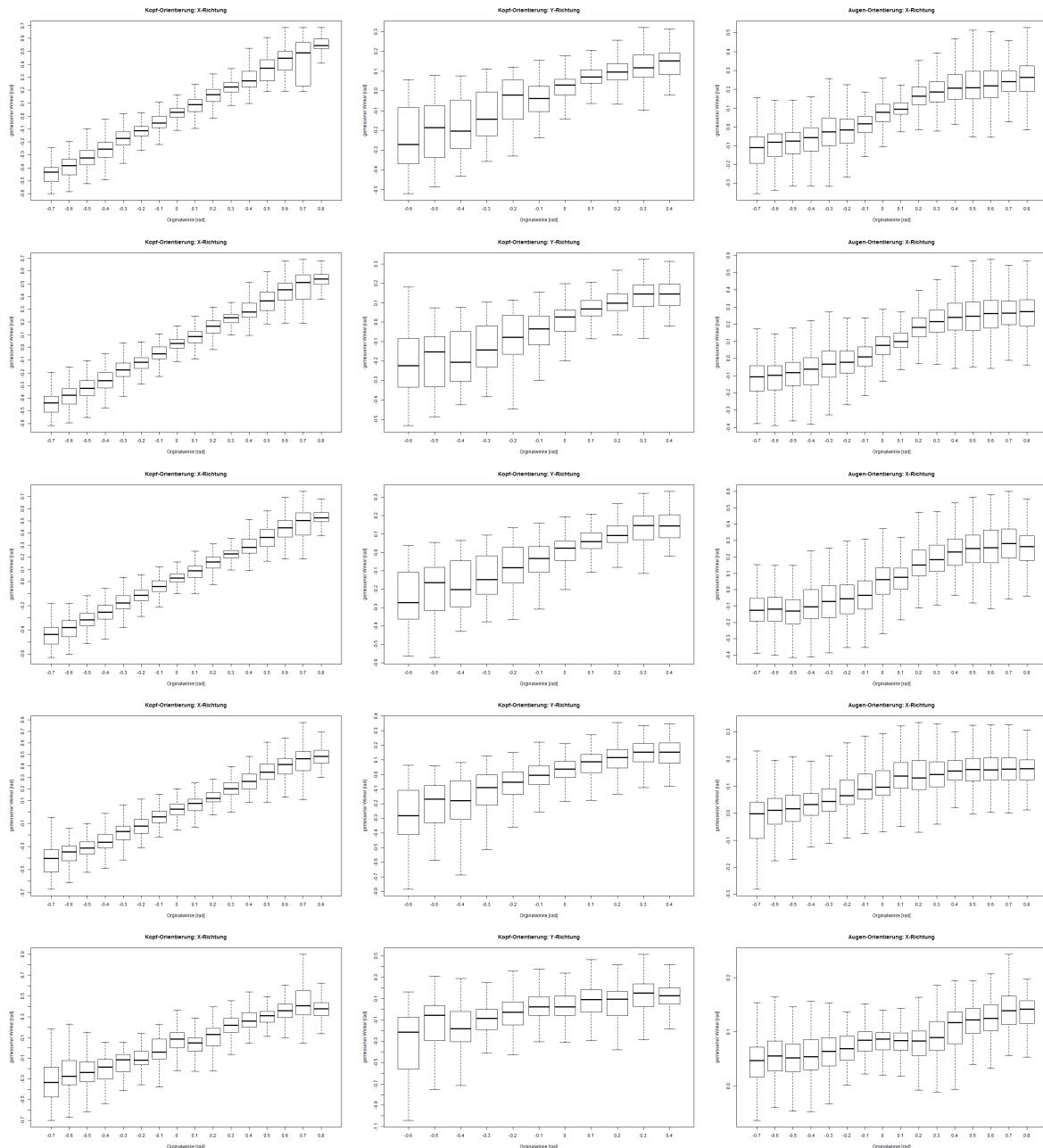


Abbildung 3.46: Dargestellt ist die Auswertung der Videoaufnahme mit der Kopfausrichtung Horizontal (Links), Kopforientierung Vertikal (Mitte) und die X-Ausrichtung der Augen (Rechts)

Skalierungsfaktor von oben nach unten (1/0.5/0.25/0.1/0.05)

Alle Boxplot sind über min. 1000 Messpunkte (Ausnahme Y-Achse -40, dort sind es nur 200)

weite wurden zwar sorgsam bestimmt, sind aber dennoch nicht perfekt.

Bedingt durch den Aufbau und der verwendeten Hardware, musste die Kamera in Richtung des Projektors ausgerichtet werden, wodurch diese vor dem direkten Licht geschützt werden muss. Somit konnte sich die Kamera nicht im Zentrum der Messpunkte befinden.

Da Kamera und Leinwand fest montiert sind, ergibt sich auch die Problematik das der Kopf der Probanden nicht im Zentrum des Kamerabildes sind und somit hat die Kamera immer einen Blickwinkel von unten auf das Gesicht.

Da die Probanden ebenfalls zwischen der Leinwand und dem Projektor standen, verdeckten diese das Bild, wodurch es manchmal passierte das der Zielpunkt im Schatten verschwand und keine zentrale Messung mit Blickrichtung nach unten möglich ist.

## **Umgebung**

Bei der Aufzeichnung hat sich vor allem das Problem mit der ungleichmäßigen Beleuchtung bzw. dem Gegenlicht ergeben. Diesem wurde durch Abdunkeln der Fenster und Verwendung der Tafelbeleuchtung entgegengewirkt, damit das Gesicht gut erkennbar ist. Ein Problem das auch in der realen Anwendung auftreten wird.

Ein weiteres allgemeines Problem ist die Auflösung des Gesichtes, somit ist eine Berechnung auf dem Gesicht zwar möglich, auf den Augen allerdings nur bedingt.

Somit ergibt sich ein weiteres Problem, da im allgemeinen eine Exkursionen, der Winkelbereich der üblichen Augenbewegungen, bis etwa  $20^\circ$  stattfindet [Wik17a] und diese nicht erfasst werden können. Ein weiteres nicht zu verachtendes Problem ist die Reflexion, vor allem auf den Brillen, die die Pupille überdecken, von den starken Lichtquellen wie z.B. Fenster, Projektor und dessen Bild, sowie Lampen usw. Auch Schatten gerade in den Augenhöhlen erschweren die Auswertung.

# 4 Ergebnisse

## 4.1 Fehleranalyse

Mit entsprechend hochauflösenden Kameras, können auch bessere Resultate auf größeren Distanzen erzielt werden. Gerade die Bestimmung der Blickrichtung auf großer Distanz ist meist nicht möglich, da die Augenpartie viel zu klein für eine Berechnung ist. So bleibt meist nur die Gesichtsorientierung mit ihr natürlichen Ungenauigkeit.

Da Bewegung erlaubt ist, passiert es immer wieder, dass Teile des Gesichtes verdeckt werden, durch Hände beim Melde, andere Schüler oder dem Lehrer selbst, der vor der Kamera steht oder sich der Kopf zu weit wegdreht und das Tracking scheitert. Aber auch die Frisuren spielen eine Rolle, da dadurch diese einige Landmarks verdeckt werden können, wie z.B. die Augenbrauen, und das Gesicht nicht erkannt wird .

## 4.2 Zusammenfassung

Für die Analyse der Gesichter in einem Video, wurden zuerst die einzelnen Gesichtern mittels MTCNN-Face Detection (Abschnitt 3.2) in allen vorhanden Frames gesucht.

Anschließend wird jede Einzelperson unterscheiden und alle gefunden Bildbereiche der jeweiligen Person zugeordnet. Diese Bildbereiche werden nun auf eine Mindestgröße gebracht (Abschnitt 3.3), damit sie den Trainingsdatensatz des nächsten Schrittes stärker ähneln. Dazu wurde die Auswirkung der verschiedenen Skalierungsverfahren auf die nachfolgende Analyse untersucht.

Nun werden die einzelnen Bildbereiche Ausgewertet (Abschnitt 3.4) und die Gesichtsorientierung kann bestimmt werden. Um die Bereiche zu simulieren in denen das Verfahren eingesetzt werden kann, wurde durch lineare Skalierung die Bild des Trainingsdatensatzes verkleinert um die verschiedenen Distanzen zu simulieren.

Um die Detektion der Pupille zu verbessern wurde ElSe (Abschnitt 3.6) verwendet, mit dem Ziel, die Blickrichtung exakter zu ermitteln. Dazu wurde die Auswirkung der verschiedenen Farbbild nach Graubild Konvertierer (Abschnitt 3.5) untersucht, sowie die Veränderung des Radius der Maske.

Abschließend wurde getestet, wie zuverlässig das gesamte Verfahren auf Videos eingesetzt werden kann, um die Aufmerksamkeit zu ermitteln, siehe Abschnitt 3.9. Dazu wurde ein Versuch durchgeführt, bei dem die Probanden ein Ziel verfolgen sollten und ermittelt wie exakt das Ziel der Aufmerksamkeit bestimmt werden kann.

Durch den nachgewiesenen Wertebereich in dem eine Auswertung möglich ist, kann das gesamte Klassenzimmer mit nur einer Kamera erfasst werden. Bei Probanden deren Blickrichtung recht stark von der Kamera abweicht, ist das Erfassen zwar möglich, allerdings starker Fehlerbehaftet.

## 4.3 Verbesserung

Die größte Problematik bei der Auswertung einer ganzen Schulklasse, ist, dass immer wieder Teile der Gesichter verdeckt werden, sei es durch den Arm eines Anderen Schülers oder der Frisur oder völlig verdeckt durch den Lehrer und ähnliches.

Dieser Problematik kann entgegen gewirkt werden, indem mehrere Kameras verwendet werden die beispielsweise an der Seite der Tafel platziert sind. Dies bietet neben der Möglichkeit einer 3D-Rekonstruktion der Szene auch die Chance das Gesicht vollständig erfasst wird.

Durch den großen Bereich in dem das Verfahren funktioniert ist die Positionswahl der Kameras recht frei und kann so gewählt werden, dass sie die gesamte Klasse erfassen, selten verdeckt und den Unterricht wenig beeinflusst.

Für die hintersten Reihen ist der Einsatz einer eigenen Kamera zu empfehlen, da diese vor allem recht klein dargestellt und oft durch die vorderen Reihen verdeckt werden.

Für eine Auswertung der Aufmerksamkeit ist die erreichte Genauigkeit ausreichend, die Tendenzen sind klar erkennbar und können entsprechend interpretiert werden.

Da der große Erfassungsbereich nur auf Videos erreicht wird, wäre es von Vorteil die Detektion und Tracking soweit zu ergänzen, dass auf Profilbilder gearbeitet werden kann um Landmarks zu erkennen. Somit kann das Tracking auch begonnen werden, wenn die Probanden nicht grob in Richtung Kamera blicken und ist gegenüber einer Drehung robuster.

# Literaturverzeichnis

- [App15] Johannes Appel. Die bedeutung der aufgaben für das beteiligungsverhalten der schüler : eine videostudie zur wirksamkeit des unterrichtsprozesses, 2015.
- [bau13] Empfehlungen für einen zeitgemäßen schulhausbau in baden-württemberg. *Ministeriums für Kultus, Jugend und Sport Baden-Württemberg*, 2012/2013.
- [BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly Media Inc., 2008.
- [BRM12] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 3d Constrained Local Model for Rigid and Non-Rigid Facial Tracking. In *Computer Vision and Pattern Recognition (CVPR 2012)*, Providence, RI, June 2012.
- [CC06] David Cristinacce and Tim Cootes. Feature detection and tracking with constrained local models, 2006.
- [CK12] Garrison W. Cottrell Christopher Kanan. Color-to-grayscale: Does the method matter in image recognition? 2012.
- [CSA00] Marco La Cascia, Stan Sclaroff, and Vassilis Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(4):322–336, 2000.
- [DL05] Derrick J. Parkhurst Dongheng Li, David Winfield, 2005.
- [FDG<sup>+</sup>13] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *Int. J. Comput. Vision*, 101(3):437–458, February 2013.
- [FGG11] Gabriele Fanelli, Juergen Gall, and Luc J. Van Gool. Real time head pose estimation with random regression forests. In *CVPR*, pages 617–624. IEEE Computer Society, 2011.
- [GM13] Debotosh Bhattacharjee Goutam Majumder, Mrinal Kanti Bhowmik, 2013.
- [GM14] E. Gross G.L. Masala. Real time detection of driver attention: Emerging solutions based on robust iconic classifiers and dictionary of poses, 2014.
- [HMLLM12] Gary B. Huang, Marwan Mattar, Honglak Lee, and Erik Learned-Miller. Learning to align from scratch. In *NIPS*, 2012.
- [HR92] Andreas Helmke and Alexander Renkl. Das Muenchener Aufmerksamkeitsinventar (MAI): Ein Instrument zur systematischen Verhaltensbeobachtung der Schueleraufmerksamkeit im Unterricht. *Diagnostica*, 38(2):130–141, 1992.
- [Kin94] Werner Kinnebrock. *Neuronale Netze: Grundlagen, Anwendungen, Beispiele*. Oldenbourg, 1994.

- [kla16] Vorgaben für die klassenbildung - schuljahr 2016/2017, 8 2016.
- [Kyb07] Jan Kybic. Point distribution models. 2007.
- [KZ15] Zhifeng Li Yu Qiao Kaipeng Zhang, Zhanpeng Zhang. Joint face detection and alignment using multi-task cascaded convolutional networks. 2015.
- [MWM08] Louis-Philippe Morency, Jacob Whitehill, and Javier Movellan. Generalized Adaptive View-based Appearance Model: Integrated Framework for Monocular Head Pose Estimation. In *8th International Conference on Automatic Face and Gesture Recognition*, Amsterdam, The Netherlands, 2008.
- [NVG06] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 03*, ICPR '06, pages 850–855, Washington, DC, USA, 2006. IEEE Computer Society.
- [Pee] Maurice Peemen.
- [SBD12] Lech Świrski, Andreas Bulling, and Neil A. Dodgson. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of ETRA*, March 2012.
- [Ste12] Vitalij Stepanov. Analyse komplexer szenen mit hilfe von convolutional neural networks, 2012.
- [TB13] Louis-Philippe Morency Tadas Baltrušaitis, Peter Robinson. Constrained local neural fields for robust facial landmark detection in the wild, 2013.
- [TB16] Louis-Philippe Morency Tadas Baltrušaitis, Peter Robinson. Openface: an open source facial behavior analysis toolkit. 2016.
- [Tue] Tübingen digital teaching lab (tüdilab).
- [WBZ<sup>+</sup>15] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proc. of the IEEE International Conference on Computer Vision (ICCV 2015)*, 2015.
- [WF16] Thomas Kübler Enkelejda Kasneci Wolfgang Fuhl, Thiago C. Santini. Else: Ellipse selection for robust pupil detection in real-world environments. 2016.
- [Wik14] Wikipedia. Active appearance model — wikipedia, die freie enzyklopädie, 2014. [Online; Stand 16. Juni 2017 ].
- [Wik16a] Wikipedia. Bicubic interpolation — wikipedia, the free encyclopedia, 2016. [Online; accessed 6-May-2017].
- [Wik16b] Wikipedia. Canny-algorithmus — wikipedia, die freie enzyklopädie, 2016. [Online; Stand 28. Juni 2017 ].
- [Wik16c] Wikipedia. Lanczos-filter — wikipedia, die freie enzyklopädie, 2016. [Online; Stand 6. Mai 2017].
- [Wik17a] Wikipedia. Augenbewegung — wikipedia, die freie enzyklopädie, 2017. [Online; Stand 13. Juni 2017 ].

- [Wik17b] Wikipedia. Convolutional neural network — wikipedia, die freie enzyklopädie, 2017. [Online; Stand 29. Juni 2017 ].
- [Wik17c] Wikipedia. Opencv — wikipedia, die freie enzyklopädie, 2017. [Online; Stand 16. Juni 2017].
- [Wik17d] Wikipedia. Point distribution model — wikipedia, the free encyclopedia, 2017. [Online; accessed 9-May-2017].
- [XZ15] Mario Fritz Andreas Bulling Xucong Zhang, Yusuke Sugano. Appearance-based gaze estimation in the wild, 2015.
- [YS16] Andreas Bulling Yusuke Sugano, Xucong Zhang. Aggregaze: Collective estimation of audience attention on public displays, 2016.