

# 1 Einführung

## 1.1 Abstarct

## 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 Aufgaben zugewandte Aufmerksamkeit recht schwierig und unterschiedliche Erfassungsmethoden versuchen dies zu bewerten. So werden z.B. Fragebögen eingesetzt, die Schüler und Lehrer selbst ausfüllen oder es gibt ein Beobachter der die Aufmerksamkeit einzelner Schüler bewertet.

Für die Bewertung von on/off-Task werden bei einer Studie verschiedene Kriterien festgelegt, wie Blickrichtung, Körperhaltung und Tätigkeit um damit das tatsächlich beobachtete Verhalten der Schüler zu bewerten.

Bei der „Videostudie zur Wirksamkeit des Unterrichtsprozesses“ [App15] wurden z.B. 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 im ein Minuten-Intervall der Schüler beobachtet und bewertet. Sind drei oder mehr Punkte 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 diese Art der Bewertung sehr zeitintensiv. Alleine eine einzige Beurteilung jedes einzelnen Schülers einer Klasse, etwa 30 Personen nach Vorgabe der Klassenbildung [kla16], benötigt dies mindestens 30 Minuten. Somit kann eine Auswertung aller Schüler während einer Unterrichtsstunde schnell 15 und mehr Arbeitsstunden dauern. Um 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 Zeit zu sparen, so wird das gesamte Verhalten eines Schülers während des Unterrichts mit nur wenigen beobachteten Minuten beschrieben und ist entsprechend ungenau. Wodurch sowohl quantitativ genaue, als auch temporal hochauflösende Daten nicht erstellt werden können.

So kann bei zu grob gewählten Auswertungsintervallen nur eine Aussage über den gesamten Unterricht gemacht werden und nicht beispielsweise über einzelne Übungen oder über einen einzelnen Schüler. [App15]

## 1.3 Problemstellung

Ziel dieser Arbeit ist es, eine automatisierte Auswertung der Blickrichtung, einem der wichtigsten Indikatoren für gerichtete Aufmerksamkeit, auf einer ganzen Klasse zu bestimmen. 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 Labora-

torien durchgeführt (TüDiLab [Tue]) Zum anderen sind die Geräte entweder Ablenkend (Brillen) oder schränken den Aktionsradius ein (Remote Tracker). Wären wir in der Lage solch eine Auswertung mit nur einer einzigen Kamera durchführen zu können, so ist der Aufbau und die Aufnahmen auch für technische Laien durchführbar.

Diese Arbeit untersucht, wie weit es technisch möglich ist das Filmmaterial einer Kamera, das die gesamte Klasse aufzeichnet, Auszuwerten im Bezug auf Blickrichtungen bzw. Ausrichtung des Gesichts und mit welchen Einschränkungen und Genauigkeiten zu rechnen ist.

[HR92]

# 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  $25Fps$ ) 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.

Außerdem die Logitech c920 HD Pro Webcam, diese liefert ein  $15Fps$  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 werden folgende Software-Elemente aus fremder Quelle eingesetzt.

### 2.2.1 ElSe

Ellipse Selection for Robust Pupil Detection (ElSe), ein Algorithmus zur Bestimmung der Pupille in einem hochauflösenden Bild des Auges. Der Ursprüngliche ElSe-Algorithmus ist für Graubilder mit Infrarotbeleuchtung ausgelegt und wurde für diese Anwendung angepasst um Farbbilder verarbeiten zu können.

[WF16]

### 2.2.2 MTCNN Face Detection

Multi-task Cascaded Convolutional Networks ist ein Algorithmus zur Detektion von Gesichtern und Bestimmung von 5 Gesichts-Landmarks in Farbbildern. Dabei werden drei CNN auf eine Bildpyramide angewendet um so zuverlässig Gesichter verschiedenster Größe im Bild zu erkennen.

[KZ15]

### 2.2.3 OpenCV

Open Source Computer Vision, ist eine C/C++ Bibliothek von Algorithmen zur Bildverarbeitung in Echtzeit, veröffentlicht unter der BSD Lizenz (Berkeley Software Distribution)

[Wik17a][BK08]

### 2.2.4 OpenFace

Ein Open-Source Echtzeitverfahren auf Basis von CLNF zur Bestimmung und Analyse von Gesichtsmerkmalen in Grau-Bildern und Videos. Dabei werden 68 signifikante Punkte im Gesicht bestimmt und auf Basis jener Position und Orientierung ermittelt.

[TB16]

## 2.3 Das Klassenzimmer - Umgebung des Eye-Tracking

Die Anwendung ist für den Unterricht ausgelegt, wie in der Problemstellung Abschnitt 1.3 beschrieben und soll diesen möglichst wenig beeinflusst, ergeben sich folgende Randbedingungen:

- Brillen, Kontaktlinsen und ähnliches sind bei den Probanden erlaubt, ebenso Frisuren, Make-up usw.
- Die üblichen Bewegungen im Unterricht wie Sprechen, Kopfdrehungen usw. der Schüler sind gestattet.
- Das Verfahren soll gleichzeitig auf Distanzen zwischen  $2.5 - 8m$  zur Kamera auf einer Breite von  $6m$  funktionieren.
- Möglichst alle Blickrichtungen bzw. die Gesichtsorientierung der Schüler sollen so exakt wie möglich erfasst werden.

Ein deutsches Klassenzimmer soll laut Baden-Württembergischen Schulbauempfehlungen eine Grundfläche von  $54 - 66m^2$  aufweisen für maximal 28-32 Schülern. Da noch die Tafel usw. beachtet werden muss ergibt sich einen Abstand von  $2.5 - 8m$  zwischen Kamera und Schüler auf einer Breiten von  $6m$ , dabei befindet sich die Kamera in der Nähe der Tafel. Somit muss der Linsenwinkel mindestens  $100^\circ$  betragen, damit alle im Bild sind, mit entsprechender Schärfentiefe.

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

[bau13]

### 2.3.1 Randbedingungen der Anwendung

Zusätzlich werden folgende Annahmen gemacht, die sich vor allem auf die Sitzordnung der Schüler und die Umgebung beziehen.

- Die Szene ist innerhalb eines Gebäudes, mit ausreichend gleichmäßiger Beleuchtung.
- Die Überführung zwischen Welt- und Kamerakoordinatensystem ist bekannt.
- 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.

## 2.4 Grundlagen

Gesichtserkennung ist eine der fortschrittlichen Verfahren in der maschinellen Bildverarbeitung und wird ständig weiter entwickelt. Darunter fallen neben der Detektion des Gesichtes auch seine Analyse wie Orientierung oder das Erkennen von Mimik wie Lächeln bei Kameras und Übereinstimmungen. Bei vielen Anwendungen ist der Stand der Technik oft ein Neuronales Netz beteiligt.

### 2.4.1 Künstliches neuronales Netz

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

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

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

Soll ein gesamtes Netz trainiert werden, so wird jedes einzelne Neuron zufällig Initialisiert und anschließend so angepasst das der Fehler auf einem Trainingsdatensatz minimal ist.

[Kin94]

### 2.4.2 Convolutional Neural Network (CNN)

CNN ist 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 erreicht. Die CNN definieren in vielen Anwendungsbereichen momentan den Stand der Technik.

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 können jeder einzelne Kernel für sich und die jeweiligen Bewertungen der einzelnen Kernel und Neuronen.

Quelle & Bild

### 2.4.3 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, auf Basis eines Trainingsdatensatzes. Nun wird versucht für das Bild, auf welchem gerechnet werden soll, für jeden Punkt den maximalen Wert zu erreichen zwischen passendem Farbverlauf und seiner Wahrscheinlichkeit.

Dieser Art der Bestimmung von Punkten mit Positionsabhängigkeiten 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 Gesichtern und ist zuverlässiger als das Active Appearance Model (AAM).

Quelle & Detecton der Landmarks

### 2.4.4 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 einer 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]

#### **2.4.5 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 der Eingabeformen  $X$  des Objektes aus dem Durchschnittsmodell dargestellt werden. Zur Vereinfachung reicht es, die signifikantesten Eigenvektoren in  $P$  auf zu nehmen und dennoch  $X$  ausreichend genau beschreiben zu können.

Ist bekannt welche Art der Verformung durch den Eingenvektor dargestellt ist, 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 eine Problematik, wenn auf Gesichtern gerechnet wird, da immer eine Veränderung der Mundwinkel oder Augenlider vorhanden ist.

[Wik17b][Kyb07][MWM08]

#### **2.4.6 Non-maximum suppression (NMS)**

Ein Verfahren um Kanten in einem Bild exakter zu bestimmen. Als Eingabe für das Verfahren, wird das Ergebnis eines Kantendetektor z.B. Kirsch-Operator verwendet. Dabei gibt die Stärke der Farbe 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.

#### **To Do**

Quelle

### **2.5 MTCNN Face Detection**

Bei Multi-task Cascaded Convolutional Neuronal Network (MTCNN) handelt es sich um ein Verfahren dass bei der Detektion von Gesichtern auch deren Ausrichtung berücksichtigt, um bessere Ergebnisse zu erzielen.

#### **2.5.1 Constrained Local Neural Fields (CLNF)**

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

Als nächster Schritt wird das Ergebnissen der Detektoren mit einer Karte der Position aller Landmarks, mit ihrer Abweichung, kombiniert um somit die beste Position der Landmarks zu erhalten auf Bezug des Farbverlaufes und relativ zu den anderen Landmarks. [TB13]

### 2.5.2 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]

### 2.5.3 Anforderungen

Sein Einsatzgebiet ist die Vorverarbeitung eines Frames für die spätere Auswertung. Somit soll dieser Schritt von einem möglichst robusten Verfahren zur Detektion von Gesichtern durchgeführt werden. Dabei kann auf einem hochauflösendem Bild mit verhältnismäßig kleinen, verschieden großen und weit verteilten Gesichtern gearbeitet werden.

### 2.5.4 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 dass 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, neben dem möglichen Farbverläufen, durch die Skalierung nicht zusätzlich zu erschweren und die CNN können auch auf ihre jeweilige Aufgabe besser optimiert werden können.

#### 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 um alle möglichen Bounding-Boxen zu ermitteln 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.

#### 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 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 und mit ihren jeweiligen 5 Landmarks ermittelt werden.

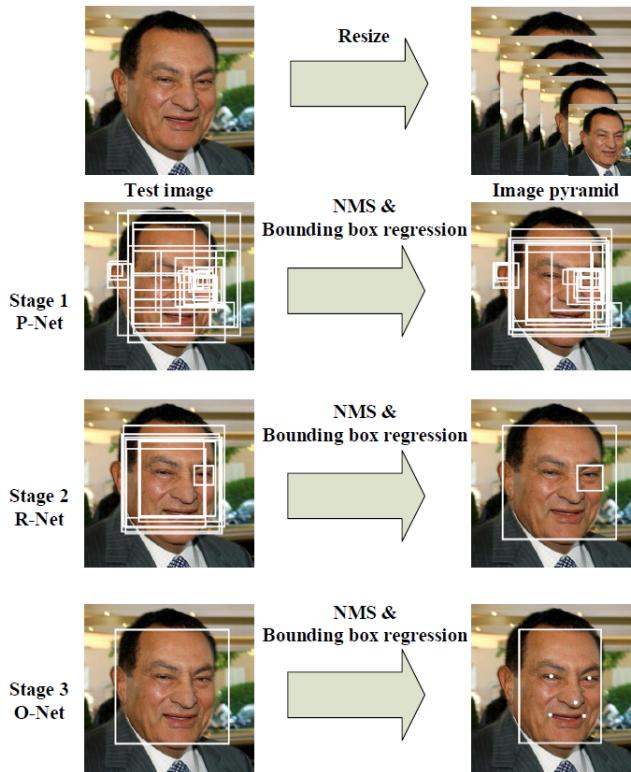


Abbildung 2.1: Darstellung des Funktionsablaufes von MTCNN[KZ15]

## 2.5.5 Qualität

MTCNN Face Detection ist bei der Zuverlässigkeit im Vergleich zu anderen bekannten Verfahren überlegen, siehe Abbildung 2.2 und zudem Echtzeit fähig. Im Test-Datensatz sind auch Gesichtern mit einer Größe von  $20 \times 20$  Pixel enthalten und wurden erfolgreich erkannt.

Somit sind alle Anforderungen erfüllt um mit diesem Verfahren den vorhanden Frame für die nachfolgenden Berechnungen vorzubereiten.

## 2.6 Skalieren von Bildern

Da die Berechnungen meist auf recht kleinen Bildausschnitten ausgeführt wird, müssen diese für weitere Rechenschritte hochskaliert werden, damit es von OpenFace besser verarbeitet wird.

Dabei ist es wichtig, dass die Gesichtsmerkmale möglichst gut rekonstruiert werden, um die entsprechenden Landmarks zu bestimmen. Dazu können verschiedene Verfahren verwendet werden.

### 2.6.1 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 2.3. Bei der Vergrößerung des Schachbretts ist kein Fehler aufgetreten, da nur zwei Farben vorhanden und Positionsabhängig sind.

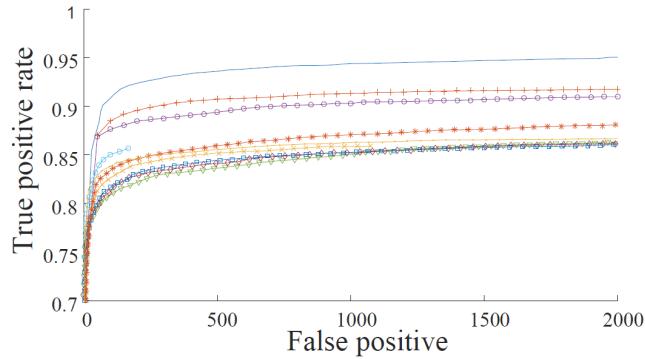


Abbildung 2.2: normale blaue Linie[KZ15]

### 2.6.2 Linear-Skalierung

Um den neuen Farbwert zu ermitteln, wird zwischen den nächst gelegenen 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 2.4.

### 2.6.3 Bicubic-Skalierung

Um den Farbwert zu ermitteln, werden die umliegenden  $4 \times 4$  Pixelwerte betrachtet 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 ist in Abbildung 2.5 zu sehen. [Wik16a]

### 2.6.4 Lanczos-Skalierung

Dieser Filter basiert auf einem bewerteten Durchschnitt der umliegenden Pixel. Die Bewertung der einzelnen Pixel wird mit einer Sinc-Funktion bestimmt damit weiter entfernte Pixel schwächer zu bewerten als die nächstliegenden, um den neuen Pixelwert zu erhalten.

Außerdem wird durch den Kurvenverlauf der Bewertungsfunktion eine gewisse Bildschärfe erreicht, siehe Abbildung 2.6. Die Funktion kann und wird für die Anwendung auf einen  $8 \times 8$  Pixel großen Bereich begrenzt. [Wik16b]

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

## 2.7 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 Problematik bei der Wahl des Verfahrens liegt in der Anforderung, vor allem soll der Farbunterschied zwischen Pupille und der Umgebung maximal sein, die Pupille möglichst dunkel und das

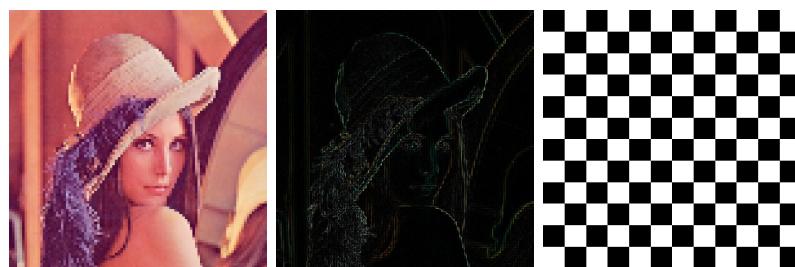


Abbildung 2.3: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels Nearest-Neighbor auf 512 Pixel vergrößert und bei Lena die Differenz zum originalen Lena-Bild bestimmt

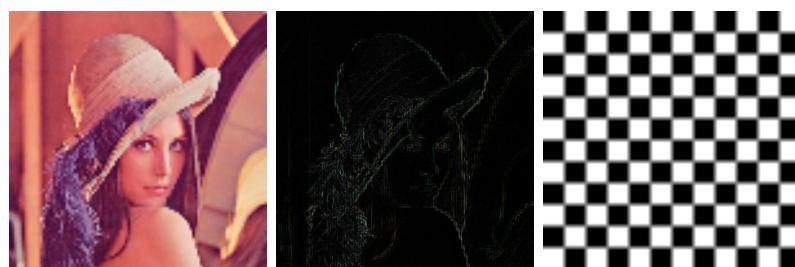


Abbildung 2.4: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels linearer Interpolation auf 512 Pixel vergrößert und bei Lena die Differenz zum originalen Lena-Bild bestimmt



Abbildung 2.5: 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

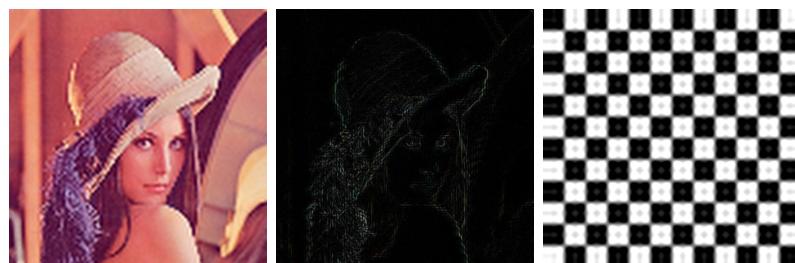


Abbildung 2.6: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels Lanczus-Verfahren auf 512 Pixel vergrößert und bei Lena die Differenz zum originalen Lena-Bild bestimmt

restliche Auge hell. Die Farbe der Iris erschwert die Differenzierung, wenn sie recht dunkel ist, der Grauwert zur Pupille entsprechend gering ausfällt. Andererseits, 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.

### 2.7.1 Luminance-Verfahren

Dies ist ein lineares Verfahren, das der menschlichen Farbwahrnehmung entspricht. 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. Außerdem ist dieses Verfahren oft Standard bei der Umwandlung von Farbbild nach Graubilder, siehe Abbildung 2.8.

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

### 2.7.2 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 2.9.

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.

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

### 2.7.3 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, allerdings für die kleinen Bereiche 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 2.10.

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.

### 2.7.4 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 2.11. Durch die Abdunklung werden kleine Farbänderungen in den dunklen Bereichen reduziert, wodurch die Pupille sehr dunkel zu sehen sein sollte, der Farbunterschied zur Iris wird allerdings ebenfalls verringert.

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

### 2.7.5 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 kleinere Farbänderungen werden entfernt.

Bei dem Max-Verfahren werden alle farbigen und helle Bereiche hell bleiben 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 andre wird abgedunkelt.

$$G_{max} = \max(R, G, B)$$

$$G_{min} = \min(R, G, B)$$

### 2.7.6 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  $G_{min}$  im Bild gesucht um anschließend wird der neue Grau-Wert  $G_{new}$  wie folgt bestimmt, dabei ist  $V_{max}$  der maximal größte Wert.

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

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, das einzelne lokal auftretende Werte nicht als Extremwert verwendet werden. Dies hat zur Folge, dass die Pupille gleichmäßiger dunkler wird und Pixel die eine Reflektion darstellen ignoriert werden, wodurch das gesamte Bild stärker aufgehellt wird.

### 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 verschwimmen und das Bild unscharf wird.

## 2.8 OpenFace

Die Aufgaben von OpenFace ist die Analyse des Gesichtes, basierend auf Bilddaten. Dabei stehen für die Anwendung nur die Kameraparameter zur Verfügung und keinerlei Zusätze wie ein Tiefenbild oder Infrarotbeleuchtung der Szene.

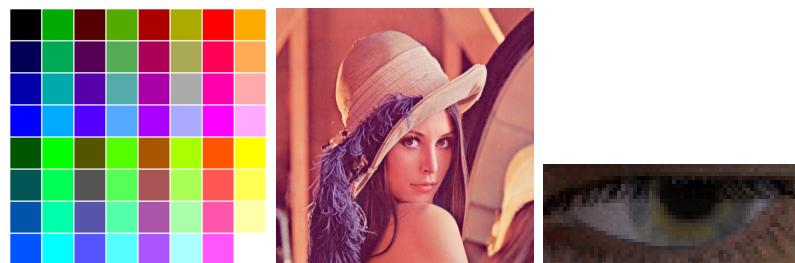


Abbildung 2.7: Dies sind die Eingabebilder der verschiedenen Konverter von Farbe nach Grau. Links eine Farbpalette, Mitte Lena und Rechts ein Augenausschnitt aus dem Augendatensatz [WBZ<sup>+</sup>15]



Abbildung 2.8: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Luminance-Verfahren



Abbildung 2.9: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Glean-Verfahren



Abbildung 2.10: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Gleam-New-Verfahren



Abbildung 2.11: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Quadrat-Verfahren



Abbildung 2.12: Ergebnis der Umwandlung von Farb- nach Grauwert mittels Extremwert-Verfahren.  
Oben: Max-Verfahren, Unten: Min-Verfahren

OpenFace kann neben den Landmarks des Gesichtes auch die Position, Blickrichtung und Gesichtsmerkmale bestimmen, die ein dargestelltes Gesicht aufweist.

Sollte ein Video als Quelle fungieren, so 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.

### 2.8.1 Verarbeitungsschritte

Der Rechenaufwand zur Verarbeitung des Eingabebildes ist so ausgelegt, das 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.

#### 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.

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. Dies Herangehensweise ist langsam, aber auch exakter als eine einfache Hypothese. Wird ein Tracing, das Verfolgen der Landmarks über mehrere Frames, auf Videos durchgeführt, so wird als initiale Hypothese das Ergebnis aus dem letzten Frame verwendet. Sollte das Tracing scheitern, so wird das CNN reseted um Neu zu beginnen.

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.

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

#### Bestimmung der Gesichtsposition

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. Sind keine Parameter 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 ergibt sich die Brennweiten  $f_x$  und  $f_y$  wie folgt:

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

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

## 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.

## 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.

### 2.8.2 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] zur Bestimmung der Blickrichtung verwendet und es ergab sich ein durchschnittlichen Fehler von 9.96 Grad.

## 2.9 Berechnung der Position

Zur Bestimmung der Kopfposition  $P = (X_{avg}; Y_{avg}; Z_{avg})$  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  $P_1 = (X_1; Y_1; Z_1)$ ,  $P_2 = (X_2; Y_2; Z_2)$  die Beschreibung der Größe  $G$  eines Kopfes mit:

$$\begin{aligned}
a &= \frac{\sqrt{(X_1 - X_2)^2 + (Y_1 + Y_2)^2}}{\frac{|Z_1 - Z_2|}{2}} = \frac{G}{Z_{avg}} \\
S &= \frac{S_G}{G} \\
\Rightarrow a \cdot f &= f \cdot \frac{G}{Z_{avg}} = S_G \\
Z_{avg} &= \frac{f}{S_G} \cdot G = \frac{f}{S} \\
X_{avg} &= \frac{x \cdot Z_{avg}}{f} \\
Y_{avg} &= \frac{y \cdot Z_{avg}}{f}
\end{aligned}$$

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

### 2.9.1 Zusammenhang von Bildposition & Weltposition

Als Ausgangspunkt werden die Ergebnisse des CNN eingesetzt um mit deren Hilfe wie in Abschnitt 2.9 beschreiben 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.

$$\begin{aligned}
euler_x &= \tan^{-1}\left(\frac{\sqrt{X^2 + Z^2}}{Z^2}\right) \\
euler_y &= \tan^{-1}\left(\frac{\sqrt{Y^2 + Z^2}}{Z^2}\right) \\
R_{pos} &= R(euler_x, euler_y, 0) && \text{Umwandlung zur Rotationsmatrix} \\
R &= R_{CNN} \cdot R_{pos}
\end{aligned}$$

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 beachtet werden. Allerdings ist auch ein Tiefenbild nötig, da ansonsten die Fehler weiter verstärkt werden. Daher ist es in der aktuellen Anwendung nicht sinnvoll einsetzbar.

## 2.10 Bestimmung der Position & Orientierung

Für die Bestimmung der Position und Orientierung des Gesichtes wird wie in Abschnitt 2.9 beschrieben ausgeführt. Dies kann Wiederrum von OpenFace übernommen werden, dazu muss nur das Zentrum des Bildes und Brennweite  $f_x, f_y$  bekannt sein. Außerdem werden noch erweiterte Verfahren angeboten, bei dem die Position im Bild besser mit einbezogen werden, um die Winkel der Kameraabbildung zu berücksichtigen.

Der signifikanteste Parameter für die Position ist die Brennweite  $f_x$ , da mit ihm die Tiefe geschätzt

wird und sollte entsprechend exakt bestimmt sein. Von Interesse ist vor allem der Punkt auf den der Blick bzw. das Gesicht ausgerichtet ist, dadurch muss neben der Position im Kamerakoordinatensystem auch die Orientierung bekannt sein.

Da nur die Position des Kopfes und seine Orientierung bestimmt werden kann, ergibt sich das Problem, den konkreten Blickpunkt zu ermitteln, da ein ganzer Kegel, wenn eine Fehlertoleranz berücksichtigt wird, als mögliche Lösungen in Frage kommen.

Außerdem liegt der Blickpunkt meist außerhalb des Bereiches der Kamera und muss entsprechend von einer Anwendung interpretiert werden.

## 2.11 ELSE

Um die Blickrichtung möglichst exakt zu bestimmen, sind die Landmarks der Pupille ausschlaggebend. Zu diesem Zweck kann ElSe eingesetzt werden, da dies ein Verfahren zur Detektion von Pupillen in Bildern unter realen Bedingungen ist.

### 2.11.1 Funktion

Das Verfahren ist in der Lage aus Bildern die Umrisse einer Pupille zu ermitteln. Bei realen Aufnahmen sind Bildfehler unvermeidlich, so können Reflexionen (Brille, Kontaktlinse usw.), Make-Up und körperliche Eigenschaften wie Augenfarbe die Detektion erschweren.

Als Ergebnis liefert ElSe eine Ellipse, die den Umriss der Pupille beschreibt.

### 2.11.2 Funktionsablauf

Als Input wird im Original ein Graubild einer Eye-Tracking-Brille verwendet, auf dem das Infrarot beleuchtete Auge abgebildet ist. Die Infrarotbeleuchtung wird verwendet, damit das Auge ausreichend beleuchtet ist ohne den Probanden zu blenden. Für den Test im Vergleich zu anderen Verfahren, wurden Bilder von  $384 \times 288$  Pixel Größe verwendet und ist auf diesen zu einer Echtzeitauswertung fähig.

#### 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 morphologischer 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.

#### Bestimmen der Ellipse

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.

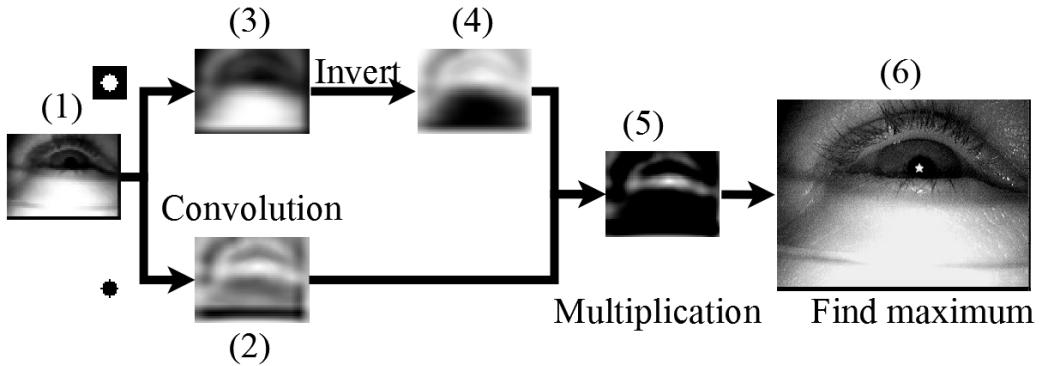


Abbildung 2.13: Ablauf der alternativen Berechnung zur Pupillen-Detektion von [WF16]

### Grobe Bestimmung der Pupille

Sollte die Bestimmung der Ellipse, wie im letzten Kapitel beschrieben, 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 2.13 (1) wird ein kreisförmiger Mean-Filter eingesetzt mit Ergebnis in Abbildung 2.13 (3). Zur zweiten Faltung wird der Durchschnitt über ein Quadrat ohne inneren Kreis eingesetzt bestimmt mit Ergebnis in Abbildung 2.13 (2), wobei beides mal der selbe Radius verwendet wird.

Nun wird das Ergebnis des Quadratischen Mean-Filters invertiert Abbildung 2.13 (4) und mittels Punkt-Multiplikation mit dem anderen Meanfilter zusammengebracht Abbildung 2.13 (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 2.13 (6) markiert.

### 2.11.3 Ergebnisse

Im Vergleich zu den anderen Verfahren im Test, ist ElSe in den meisten Fällen 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 Reflexionen der Kantenverlauf gestört wird.

Für die Anwendung ist der Bereich der Augen sehr klein und eine klare Detektion einzusprechend schwierig, wodurch vor allem die grobe Bestimmung der Ellipse von Interesse ist.

## 2.12 Bestimmung einer Position auf der die Aufmerksamkeit liegt

Ist bekannt wohin die Personen alle Blicken, kann dies aus ihrer Blickrichtung bestimmt werden.

### 2.12.1 Schnittpunkt berechnen

Verwende Blickrichtung mit Linie  $L_i = s \cdot n_i + p_i$  mit  $s \in \mathbb{R}$  und  $n_i, p_i \in \mathbb{R}^3$

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

### 2.12.2 Mittelwert

Ermittle aus den bestimmten Winkel die Blickrichtung mit  $O = R_{a,b,c} \cdot (0, 0, -1)^T$  und deren Durchschnitt  $O_{avg}$  und die Durchschnittliche Position  $P_{avg}$  der Personen. Bei der Bestimmung der Tiefe muss nun geschätzt werden. Wenn das Ziel die Tafel ist und die Kamera an der Tafel platziert wurde, kann die Tiefe im Bild verwendet werden um somit die Position der Tafel zu ermitteln.

Dies muss angewandt werden, wenn die Blickrichtungen zu sehr parallel verlaufen oder so verrauscht sind um ein sinnvolles Ergebnis mit der Bestimmung des Schnittpunkt Unterabschnitt 2.12.1 zu erhalten.

# 3 Implementierung

## 3.1 Ablauf der Implementierung

Zur Bestimmung der Kopfposition und Orientierung wird ein mehrstufiges Verfahren eingesetzt. Am Anfang müssen alle Gesichter, die im aktuellen Frame vorhanden sind, detektiert werden. Dazu wird die MTCNN Face detection verwendet, da dieses Verfahren auch kleinste Gesichter erkennt, siehe Abschnitt 3.2

Für die weiteren Berechnungen muss bekannt sein, welchen Bereich von einem Gesicht im Frame eingenommen wird. Sind mehrere Gesichter in mehreren Frames des Videos abgebildet, so muss auch eine Identitätszuordnung vorgenommen werden, damit bekannt ist welches Gesicht in Bild 1 welchem in Bild 2 entspricht.

Damit OpenFace 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 mit OpenFace weiterverarbeitet, um die Position der Landmarkes zu bestimmen. Durch die vorige Zuordnung der Gesichert kann OpenFace gezielt auf dieser Person arbeiten und sich entsprechend darauf einstellen, um so bessere Ergebnisse zu erzielen. Außerdem könne alle gefundenen Personen gleichzeitig (parallel) ausgewertet werden, siehe Abschnitt 3.4

Um die Position der Pupille noch exakter zu ermitteln kann ElSe verwendet werden, nur durch eine exakte Bestimmung der Pupillenposition ist auch eine genaue Blickrichtungsbestimmung möglich. Allerdings muss das Ergebnis von ElSe auf Plausibilität geprüft werden, um grobe Fehler zu vermeiden, siehe Abschnitt 3.5.

Nun wird auf Basis der Landmarks und der Kameraparameter die Position und Orientierung des jeweiligen Gesichtes bestimmt. Diese kann dann von weiteren Anwendungen verwendet werden. Abschnitt 2.10

## 3.2 Gesichtserkennung

Da nur eine einzige fest montierte Kamera ohne Zoom eingesetzt wird, muss diese eine entsprechend hohe Auflösung besitzen damit alle Personen zu erkennen sind. Allerdings machen die eigentlichen Bereiche der Gesichter nur einen sehr geringen Anteil des gesamten Bildes aus und diese relevanten Bildausschnitte müssen für die spätere Anwendung noch aufbereitet werden, siehe Abschnitt 3.3

Für die automatische Detektion wird MTCNN-Face eingesetzt, da dieses Verfahren im Vorabtests auf Probebildern einen sehr guten Eindruck gemacht hat und die meisten Gesichtern mit verschiedenen Größen und Blickrichtungen finden konnte. Sogar recht kleine mit  $20 \times 20$  Pixeln soll laut Beschreibung des Verfahrens Abschnitt 2.5 möglich sein. Bei diesem Schritt müssen alle Gesichert gefunden werden, auf denen die Berechnung stattfinden soll. Dabei muss das gesamte Gesicht in der Box sein, weitere Besonderheiten gibt es nicht, da OpenFace einen eigenen Facedetector besitzt.

Die von den beiden Methoden (OpenFace und MTCNN-Face) ausgegebenen Boxen sind allerdings in ihren Ausmaßen nicht identisch. Je nach verwendetem Trainingsdatensatz und darin enthaltener Annotation werden z.B. Kinn und Haaransatz noch als Gesichtsbereich oder schon als außerhalb betrachtet. 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.

Ebenfalls in diesem Schritt werden die einzelnen Boxen den Personen zugeordnet, damit im späteren Verlauf das korrekte CLNF für die Person verwendet werden kann. Für die Zuordnung reicht es meist aus, jene Box zu wählen die am ehesten den selben Bereich wie im vorigen Frame einnimmt. Dabei wird einfach für jede Box im neuen Frame die Box im vorigen Frame gesucht die den selben Bildausschnitt repräsentiert. Dies ist ausreichend, da die Gesichter sich meist weder groß Bewegen noch sich die einzelnen Boxen der anderen 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 kann dieser Bereich 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.

### 3.3 Skalierung auf Mindestgröße

Da OpenFace optimiert ist auf Gesichtern von mindestens 100 Pixel zu arbeiten, werden die Bildbereiche auf diese Größe hochskaliert. Abschnitt 2.6

Dies erhöht den Informationsgehalt der Bilder nicht, sie sind nur besser nutzbar, da sie dem Trainingsdatensatz stärker ähneln. Die von MTCNN gelieferten und vergrößerten Boxen werden nun auf  $130 \times 180$  Pixel gebracht um Ungenauigkeiten bezüglich der Position und Dimension des Kopfes im Bild entgegen zu wirken. Neben der Skalierung des Bildausschnittes, muss bekannt sein, wie Punkte im skalierten Bildausschnitt in das Frame überführt werden kann, damit dies bei späteren Berechnungen berücksichtigt wird.

Die Skalierung ist für jeden Bildausschnitt individuell und kann sich durchaus über die Zeit ändern, wenn sich z.B. die Distanz zwischen Person und Kamera verändert.

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.

### 3.4 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 vermeiden. Außerdem kann für jede Person die passende Initialisierung des CLNF basierend auf dem letzten Ergebnis dieser Person gewählt werden, auch auf für jene Personen die nur selten zu sehen sind. Auf diese Weise kann der Bildausschnitt möglichst exakt und allen gleichzeitig 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 Videoauswertung verwendet wird, wodurch sich der Arbeitsbereich auf dem Ergebnisse liefert werden sich deutlich erhöht. 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, zu Gunsten der maximalen Distanz. Außerdem hat sich gezeigt das bei Verwendung eines Videos das Gesicht deutlich kleiner sein kann bis endgültig keine Auswertung mehr möglich ist, kann bei einer erfolgreichen Detektion auch die nachfolgenden Frames 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.

## 3.5 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.

Da für die Bestimmung der Blickrichtung die Umrisse und vor allem das Zentrum von Pupille und Iris ausschlaggebend sind, 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 sich bestimmte Ellipse auch innerhalb der Augenhöhle befindet.

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.

### 3.5.1 Auswirkung der verschiedenen Verfahren

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 auch in der späteren Anwendung der Augenbereich genauer bestimmt ist, als in den Bildern dargestellt, wurde der Bildbereich soweit verkleinert damit noch alle Landmarks des Auges mit etwas Rand in diesem liegen. Somit sind die Bildausschnitte auf denen gerechnet wird etwa 64 auf 29 Pixel groß und werden für die Verarbeitung auf eine Breite von 384 Pixeln vergrößert. Um die Qualität der Berechnung bei verschiedenen Größen zu simulieren, wurde das Bild linear verkleinert.

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

Es sind Unterschiede zwischen den einzelnen Verfahren zu erkennen. Das beste Ergebnis liefert das

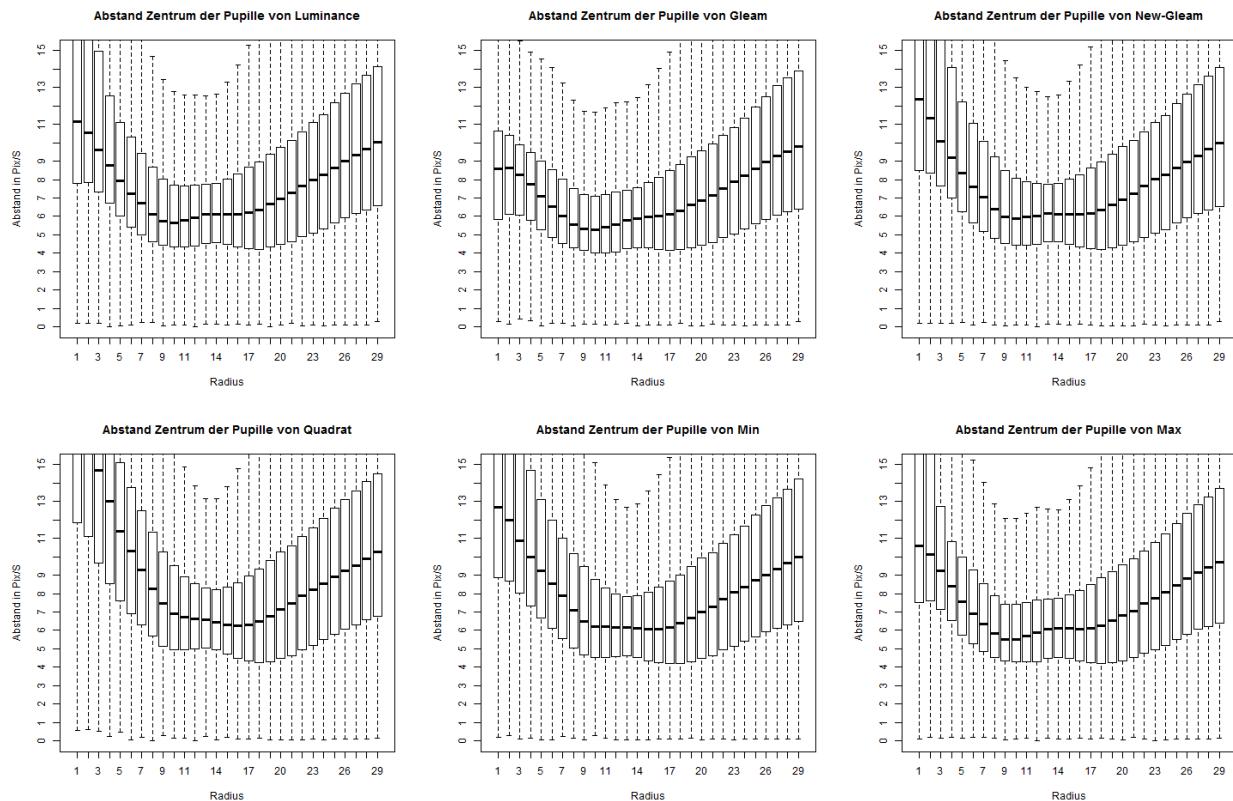


Abbildung 3.1: Abstand des Zentrums der Landmark-Pupille und der berechneten Ellipse in [Pixel/Skalierung]

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

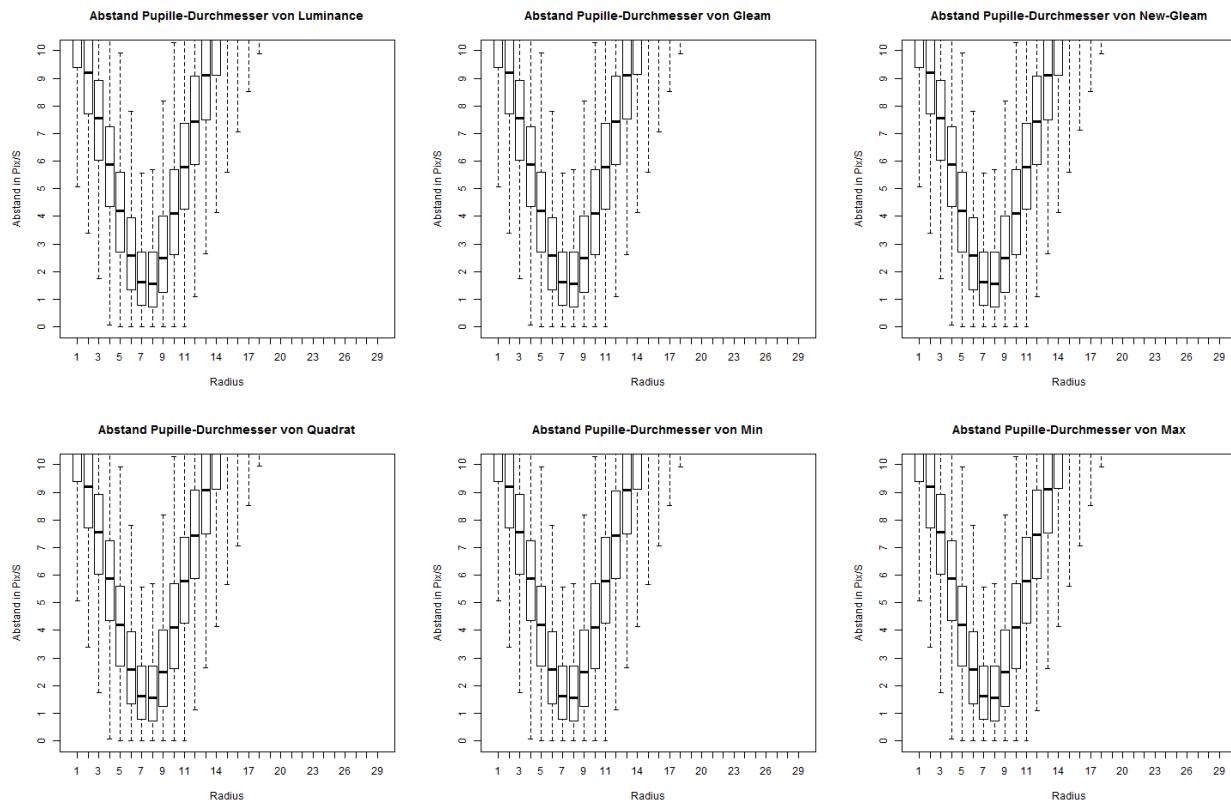


Abbildung 3.2: Unterschied Zwischen den Radien der Landmark-Pupille und der Berechneten Ellipse in [Pixel/Skalierung]

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

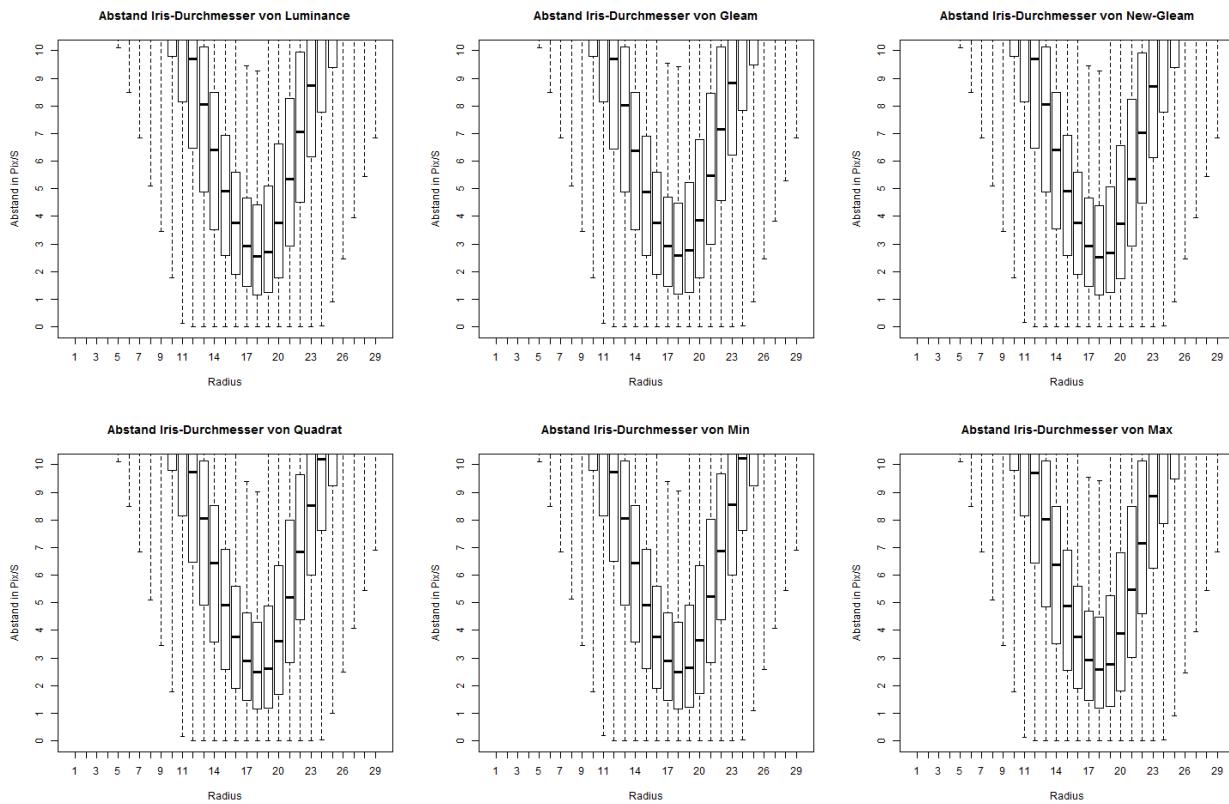


Abbildung 3.3: Unterschied Zwischen den Radien der Landmark-Iris und der Berechneten Ellipse in [Pixel/Skalierung] gegen die Radius-Größe.

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

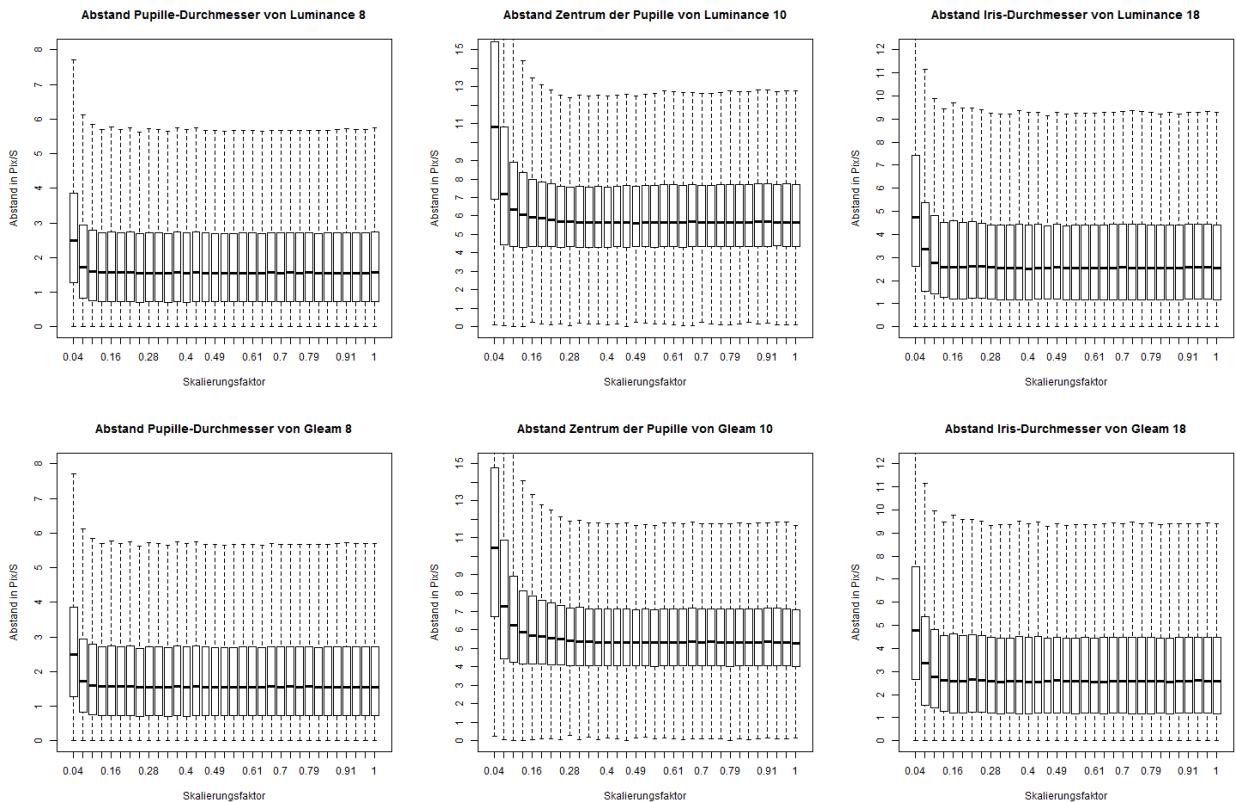


Abbildung 3.4: Auswirkung von der Bildgröße auf die Qualität der Berechnung.  
Oben: Luminance, Unten: Gleam

Gleab-Verfahren (Beschreiben in Unterabschnitt 2.7.2) heraus mit einer Abweichung von 5.89 Pixeln, siehe Abbildung 3.1, da die Abweichung vom Zentrum minimal ist. Ein mittleres Ergebnis liefert das Luminance-Verfahren, beschreiben in Unterabschnitt 2.7.1, mit welchem eine Abweichung auf dem Augen-Trainingsdatensatz von 6.42 Pixel erreicht wird.

Im Vergleich liefert das Quadratische-Verfahren, beschreiben in Unterabschnitt 2.7.4 hat im Test die schlechtesten Ergebnisse, da die durchschnittliche Abweichung bei 7.23 Pixel liegt.

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 3.4.

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.

### 3.5.2 ElSe - Auswirkung des Radius

Ein weiter wichtiger Parameter des ElSe-Verfahrens ist der Radius des Filters. Wiederrum wurde der Augen-Datensatz [WBZ<sup>+</sup>15] verwendet und die Augenpartie ausgeschnitten. Im Datensatz besitzen

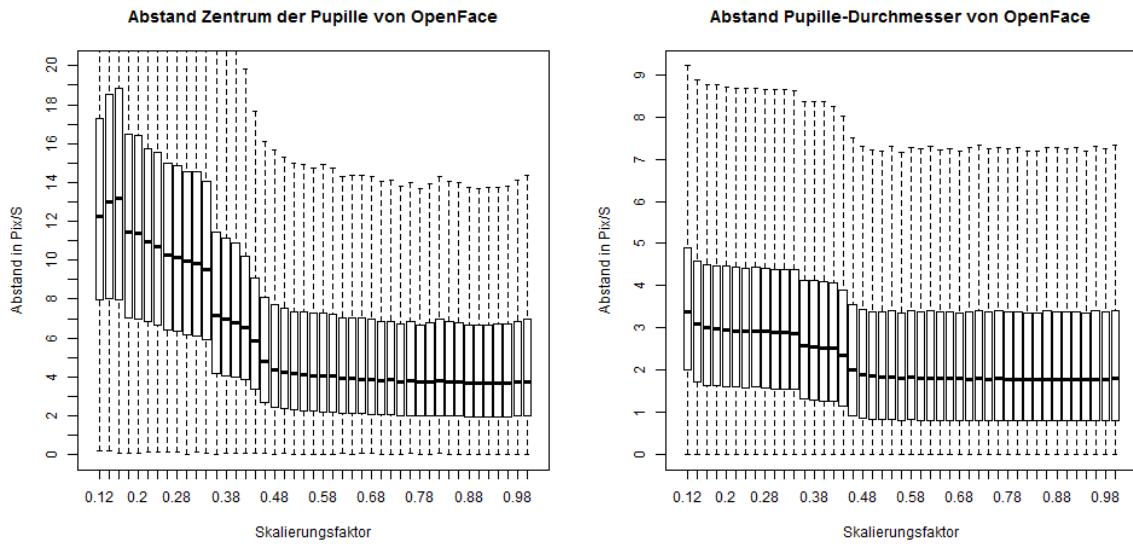


Abbildung 3.5: Auswirkung von Skalierung auf die Qualität der Augendetektion von OpenFace

die abgebildeten Augen eine Durchschnittlich Pupille von 15 Pixel und eine Iris von 34 Pixel. In Abbildung 3.3 und Abbildung 3.1 ist zu erkennen, dass die Wahl des Radius signifikant für die Qualität der Berechnung ist. Da für die spätere Anwendung vor allem das Zentrum der Pupille von Interesse ist, siehe Abschnitt 2.8.1, muss ElSe in diesem Aspekt zuverlässig Ergebnisse liefern. Im Versuch hat sich ein Radius von etwa einem Zwölftel des zu erwartetem Durchmesser der Iris bzw. Pupille als sinnvoll erwiesen, um deren Dimension 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 3.1, wobei dieser Fehler nicht so sehr steigt bei Veränderung des Radius, als bei der Größenbestimmung von Pupille und Iris.

### 3.5.3 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.

Es 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 Ergebnisse.

Da die hohe Qualität von ElSe nur erreicht werden kann, wenn es auf den passenden Bildausschnitt angewendet wird, ist auch die Detektion des Auge von Interesse.

Nach Abbildung 3.6 wird der Bereich des Auges zwar nicht so exakt bestimmt, allerdings überdeckt er den relevanten Bereich ausreichend genau. Dargestellt sind Koordinaten, X- und Y-Position in Pixel sowie die Ausdehnung, Width und Height bestenfalls 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.

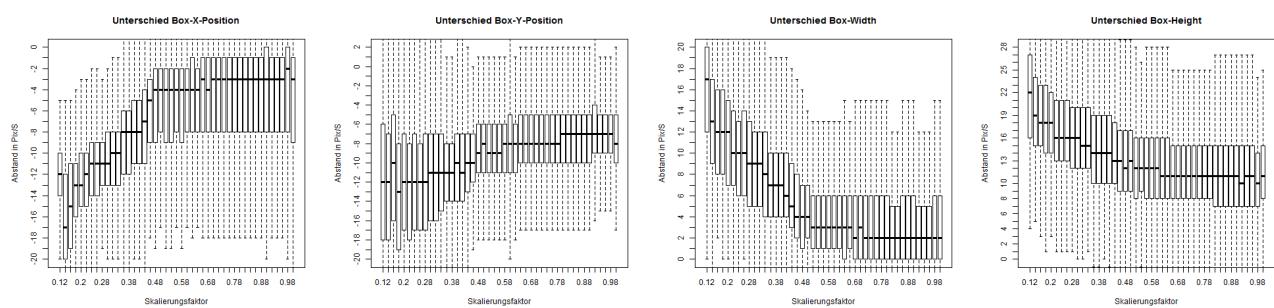


Abbildung 3.6: Bestimmung der Box ums Auge

# 4 Ergebnisse

## 4.1 Erreichte Werte

### 4.1.1 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 Beschleunigung wurde OpenFace zu erst auf das gesamte Bild eingesetzt um die möglichen Gesichter zu finden, in jeder Skalierungsstufe wurde nur der Gesichtsbereich, mit Toleranz, verwendet. Zur Durchführung wurden die Größe der Bilder mit dem Skalierungsfaktor multipliziert um so kleinere Gesichter zu erhalten und anschließend mit dem Image-Detector von OpenFace zu verarbeiten. Das Ergebnis ist in Abbildung 4.1 dargestellt.

Es ist zu erkennen, dass die Wahrscheinlichkeit auf eine erfolgreiche Detektion ab 0.5, also etwa Gesichert mit 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 4.1.

### 4.1.2 verschiedenen Skalierungsverfahren

Um auf den gewünschten Distanzen arbeiten zu können, wird der jeweilige Bereich Hochskaliert. Dazu wird das Ursprüngliche Bild ( $250 \times 250$ ) linear um den angegebene Faktor verkleinert und anschließend mit den angegebenen Verfahren auf  $300 \times 300$  wieder vergrößert. Die Wahrscheinlichkeit auf eine Detektion ist in Abbildung 4.2 dargestellt.

Es ist zu erkennen das durch die Vergrößerung, Gesichter in Bereichen die normal nicht erkennbar sind, ausgewertet werden können. Als das ungeeignetste Verfahren hat sich Nearest-Neighbor herausgestellt, siehe blaue Linie Abbildung 4.2. 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.

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 Verkeilung deutlich besser als Originalaufnahmen, da Pixelrauschen und Ähnliches nicht vorhanden ist.

### 4.1.3 Pixelrauschen bei den Skalierungsverfahren

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.

Mit diesem Test soll geprüft werden, welches der Verfahren auch stabil gegen Rauschen ist.

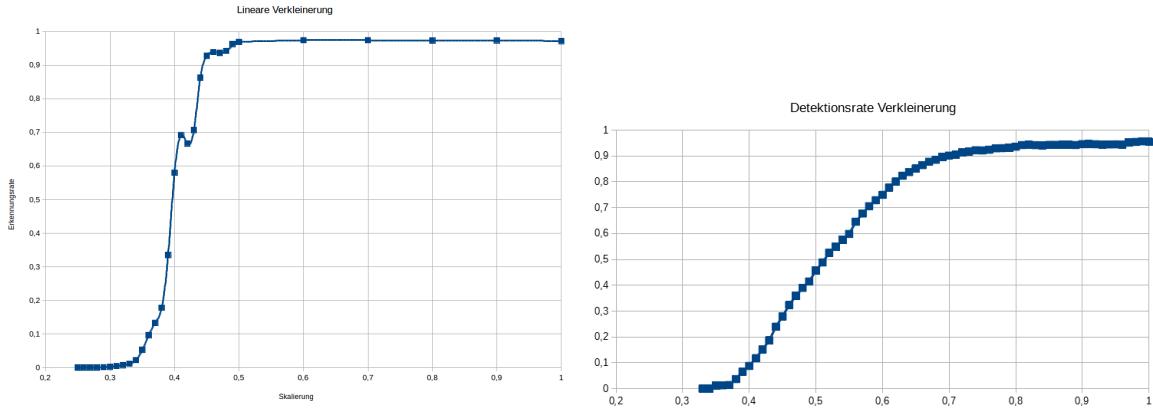


Abbildung 4.1: Die Bilder aus Labeled Faces in the Wild [HMLLM12] (links) und Random Forests [FDG<sup>+</sup>13] wurden mit den Faktor auf der X-Achse linear verkleinert und die Erkennungsrate Y-Achse abgebildet

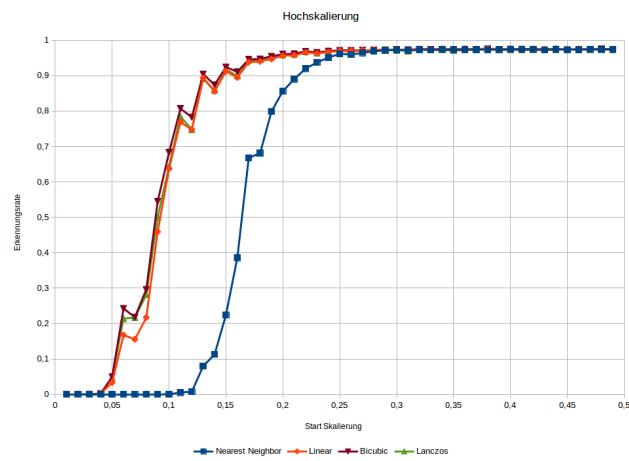


Abbildung 4.2: Die Bilder aus Labeled Faces in the Wild [HMLLM12] wurden mit den Faktor auf der X-Achse linear verkleinert und mit den verschiedenen Verfahren wieder vergrößert Abschnitt 2.6. Aufgetragen gegen die Detektionswahrscheinlichkeit. Nearest-Neighbor (blau), Linear (rot), Bicubic (braun), Lanczos (grün)

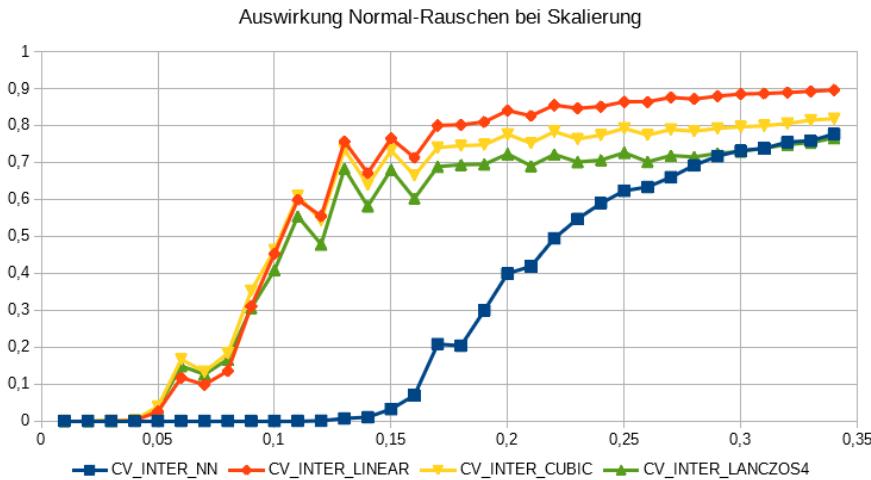


Abbildung 4.3: Bilder aus Labeled Faces in the Wild [HMLLM12], mit dem X-Faktor verkleinert, um jedes Pixel mit 50% Wahrscheinlichkeit auf  $\pm 10\%$  Gleichverteilung der Abweichung

Das Rauschen wird für jedes Pixel mit einer Wahrscheinlichkeit von 50% besteht auf eine gleich verteilte Abweichung von  $\pm 10\%$  des Originalen Farbwertes. Dieser Vorgang wurde simuliert, indem für jedes Bild vier mal 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, die gesamte Erkrankungsrate ist signifikant kleiner als ohne Rauschen, wobei die Position (0.15) ab der die Erkennungsrate rapide abfällt beibehalten wird.

#### 4.1.4 Auswirkung von Pixelrauschen

Durch Aufnahme eines Schwarzbildes der Actioncam zeigt sich, dass das Pixelrauschen recht hoch ist, siehe Abbildung 4.4. Das Rauschen hat keine Normalverteilung, sondern es besteht aus kleinen Bereiche, die den selben fehlerhaften Farbwert besitzen.

#### 4.1.5 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 sein 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 würden die Bilder mit dem angegebene Faktor (X-Achse) verkleinert und mit dem Original verglichen.

Da verschiedene Verfahren angeboten werden zur Bestimmung der Position und Orientierung, werden diese miteinander verglichen, siehe Abbildung 4.5. Zur Bestimmung wurde nur das RGB-Bild verwendet und nicht zusätzlich die Tiefeinaufnahme, da dies in der Anwendung auch nicht vorhanden sind. Es zeigt sich, dass Pose World, also die einfache Bestimmung der Position mittels Skalierungsfaktor und zusätzlicher Korrektur der Winkel die besten Ergebnisse liefert.

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



Abbildung 4.4: Aufnahme eines Schwarz-Bildes ( $2688 \times 1520$ ) der Actioncam um den Faktor 7 verstkt und invertiert.

## Position

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

Die Funktionen Pose Camera und Pose World (Oben in Abbildung 4.5) verwenden die einfache Bestimmung mittels Skalierung. Dargestellt sind nur die X-Werte, da die Y-Werte eine recht hnliche 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 4.6.

Der schnelle Abfall der Genauigkeit ist an der selben Stelle (0.5) an der auch die Detektionsrate stark absinkt. 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 verschiedenen Methoden miteinander verglichen. Die Analyse hat gezeigt, dass die Qualitt der Verfahren von den einzelnen Rotationen abhngt.

Bei der X-Rotation, dargestellt in Abbildung 4.7 knnen die rechten Verfahrenen (Pose World und Correct Pose World) berzeugen. Vor alle, Pose World hat selbst bei kleinen Abbildungen nur eine mittlere Abweichung von  $8.5^\circ$

Um die Y-Rotation zu ermitteln ist nun allerdings die linken (Pose Came und Correct Pose Came) den rechten (Pose World und Correcht Pose World) deutlich berlegen, siehe Abbildung 4.8. Auch hier liegt der mittlere Fehler ber lange Zeit bei etwa  $9^\circ$ .

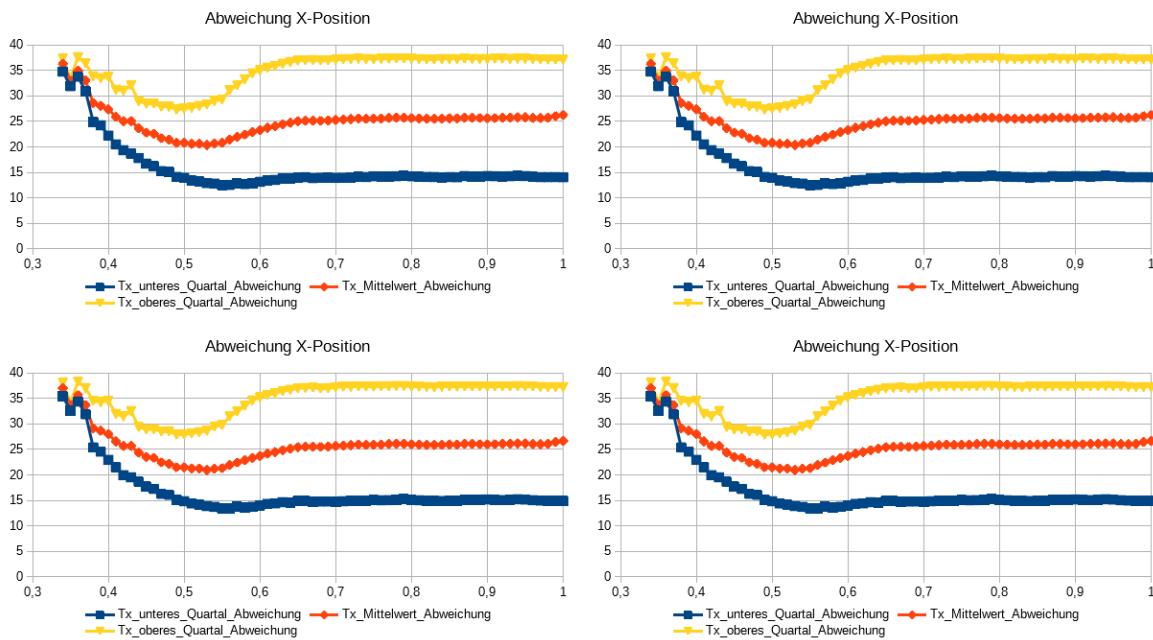


Abbildung 4.5: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist in Millimeter.

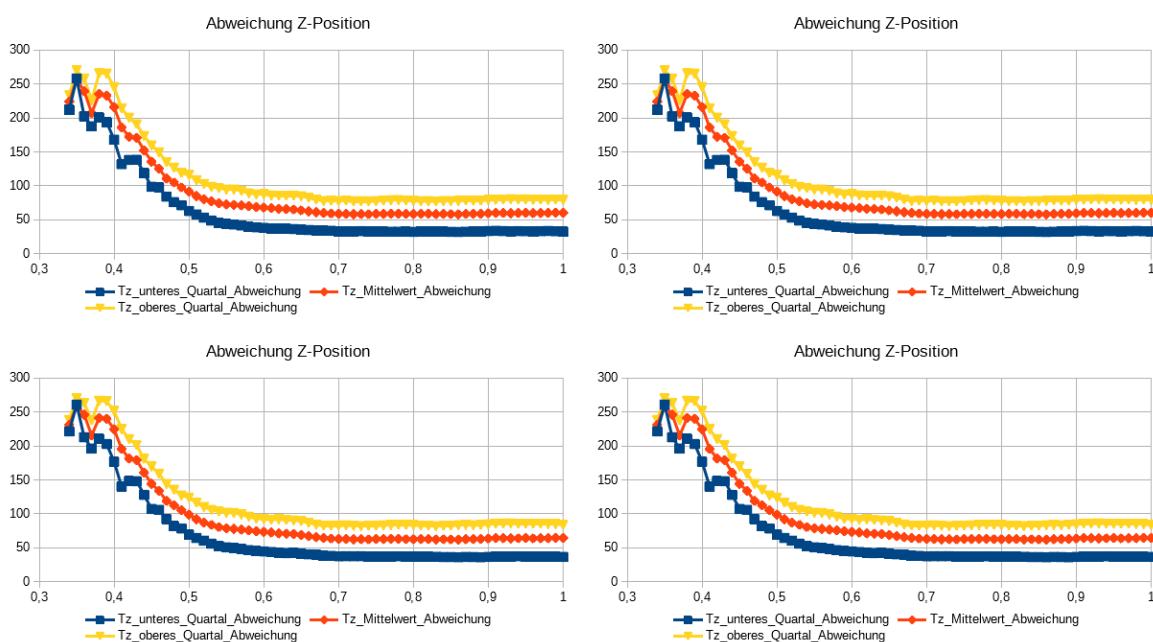


Abbildung 4.6: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist in Millimeter.

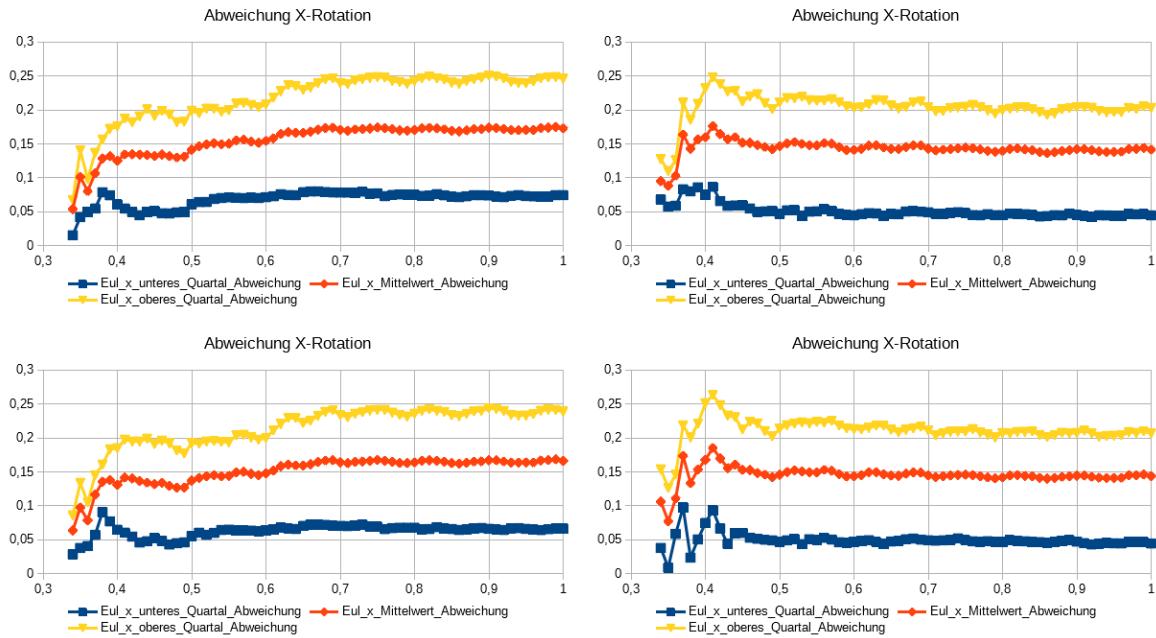


Abbildung 4.7: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist im Bogenmaß.

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 4.9. Wobei auffällt, dass Pose World bei Werten unter 0.4 plötzlich der Fehler sehr stark zunimmt.

### Wertebereich Rotation

Von Interesse sind auch die Winkel, bei den Gesichter in verschiedenen Skalierungen noch erkannt werden, siehe Abbildung 4.10.

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 sein für die Anwendung, auch wenn die Rotation parallel zur Horizontalen etwas größer sein könnte.

### 4.1.6 Qualität der Skalierung

Nun wird der Zusammenhang zwischen den verscheiden Skalierungsverfahren und der Qualität der Berechnung gesucht.

Es zeigt sich, dass bei der Bestimmung der Parameter das Nearest-Neighbor Verfahren die am genauesten Ergebnisse liefert. Allerdings ist der Wertebereich deutlich eingeschränkt. Die Mindestgröße des Gesichts im Orginal und den geringer Wertebereich bei den Rotationen ist dieses Verfahren eher ungeeignet.

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

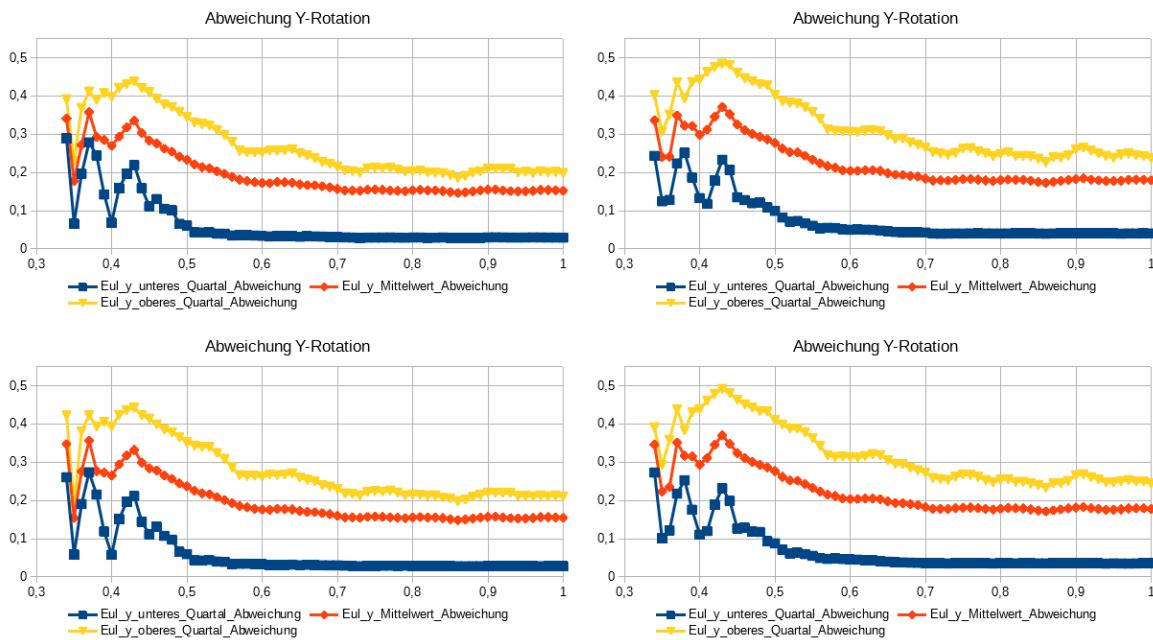


Abbildung 4.8: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist m Bogenmaß.

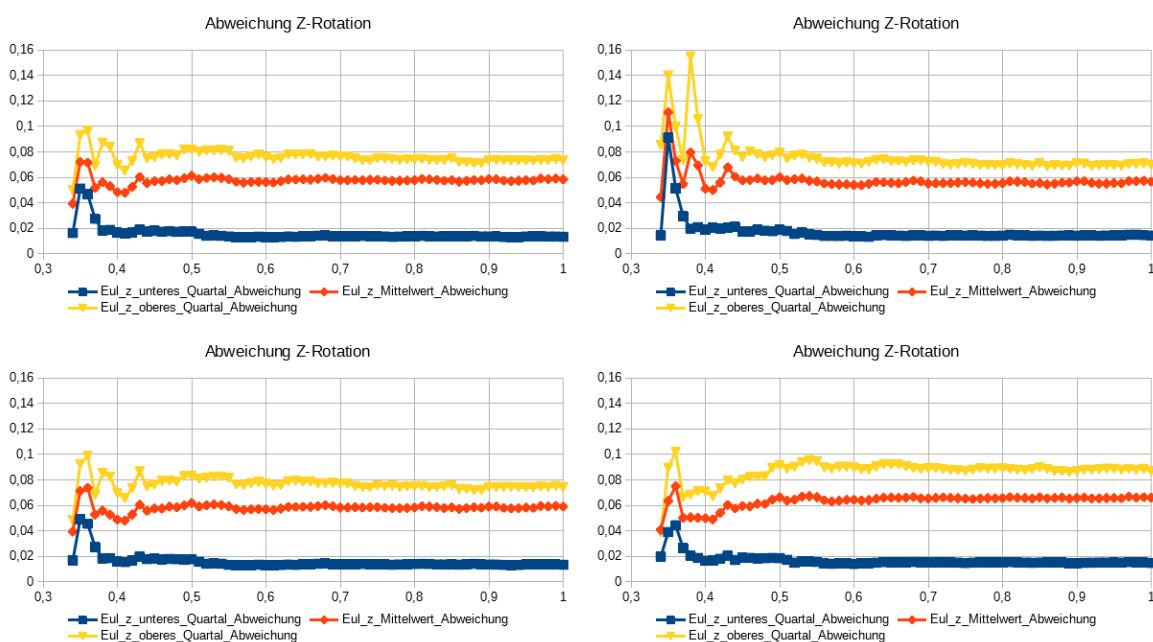


Abbildung 4.9: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist im Bogenmaß.

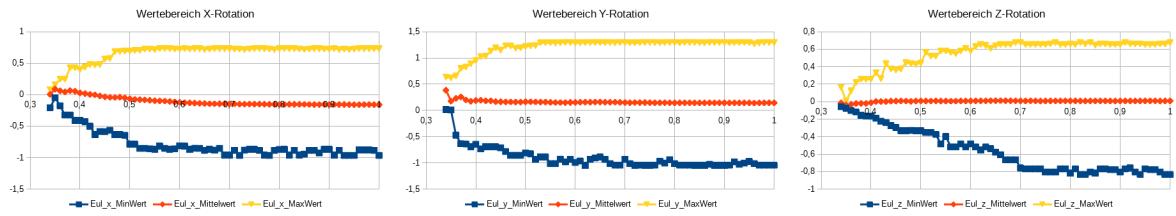


Abbildung 4.10: Darstellung der noch detektierten Wertebereiche in Bogenmaß.

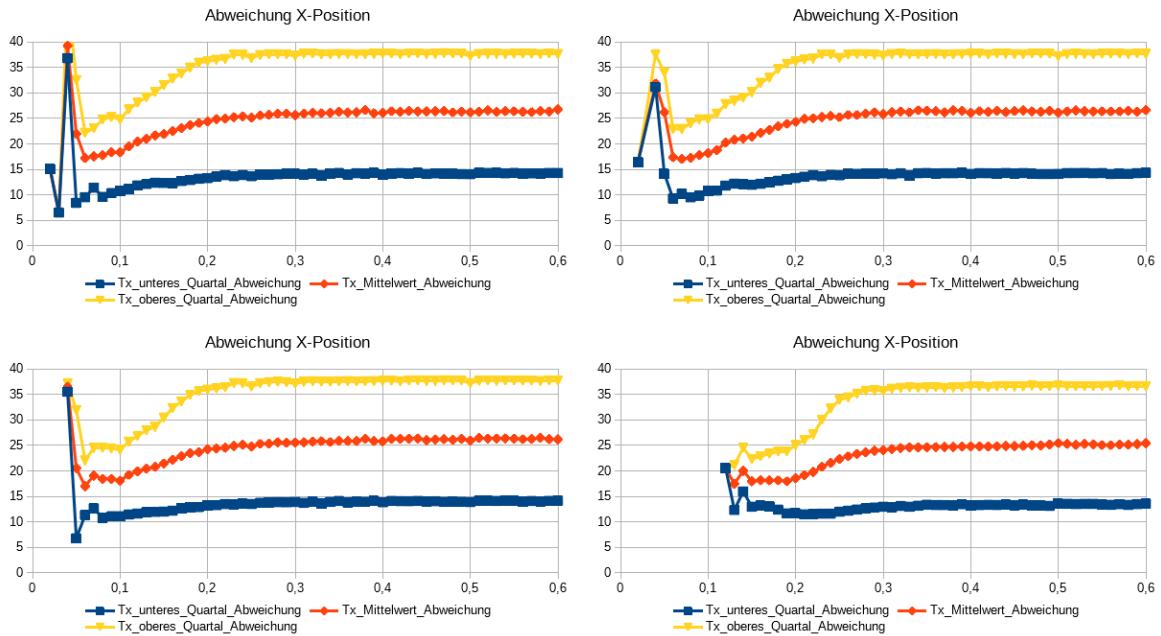


Abbildung 4.11: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung in X-Richtung (Y-Achse) in Millimeter. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

gleichmäßiger Ergebnisse. Somit kann die Wahl des Verfahrens vom Rechenaufwand abhängig gemacht werden.

## Position

Als erstes wird die berechnete Distanz miteinander verglichen.

In Abbildung 4.11 ist die Abweichung entlang der X-Achse dargestellt. Nearest-Neighbor liefert die genauesten Ergebnisse, auch wenn durch die schlechtere Detektionsrate dieses Verfahren früher ausfällt als die anderen drei.

Auf der Y-Achse ist das Lineare-Verfahren etwas besser als die Andren, das Nearest-Neighbor ist hierbei überraschend das Schlechteste, siehe Abbildung 4.12.

Nur schwer zu erkennen, da der Unterschied nur minimal ausfällt, ist auch bei der Bestimmung der Z-Position das Nearest-Neighbor-Verfahren am Besten, siehe Abbildung 4.13. Die anderen drei sind nahezu identisch. 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.

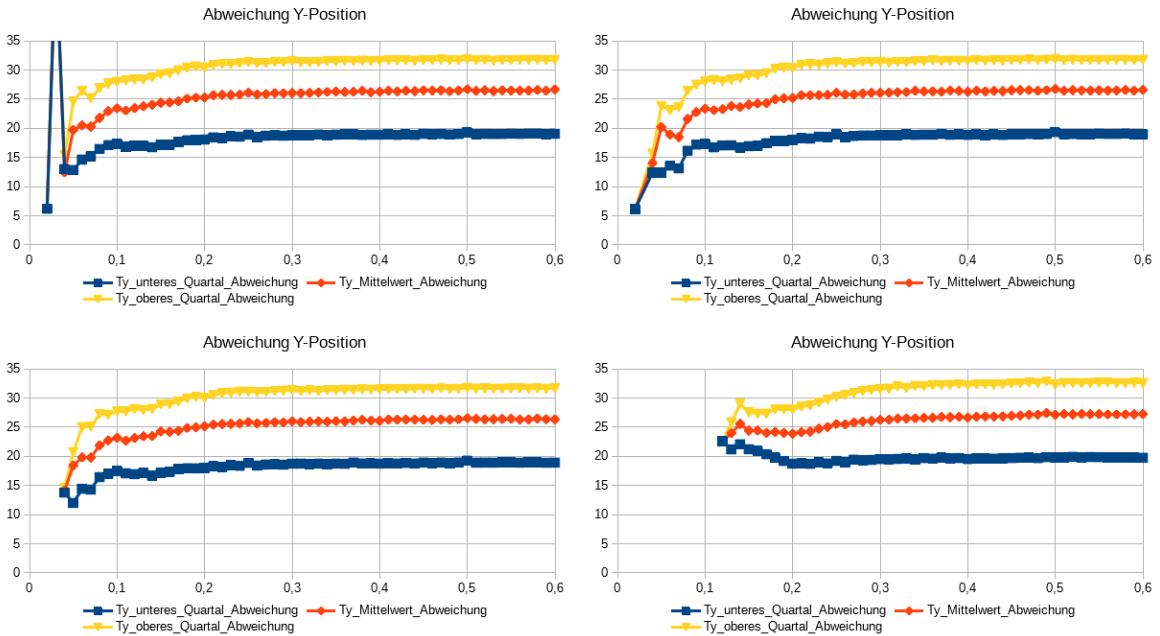


Abbildung 4.12: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung in Y-Richtung (Y-Achse) in Millimeter. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

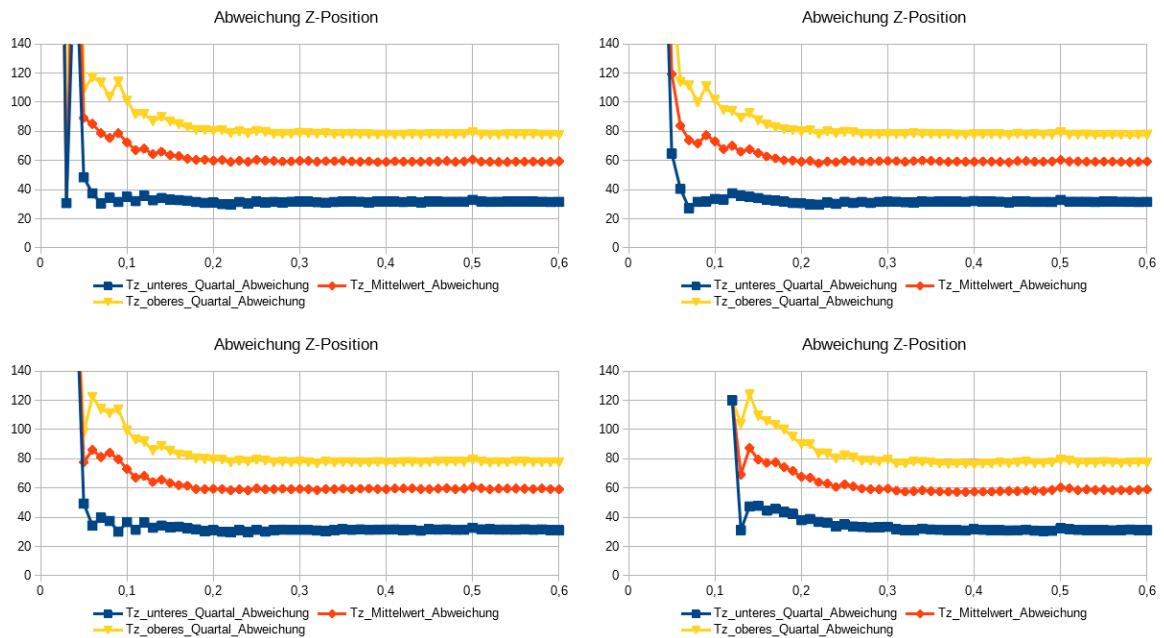


Abbildung 4.13: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung in Z-Richtung (Y-Achse) in Millimeter. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

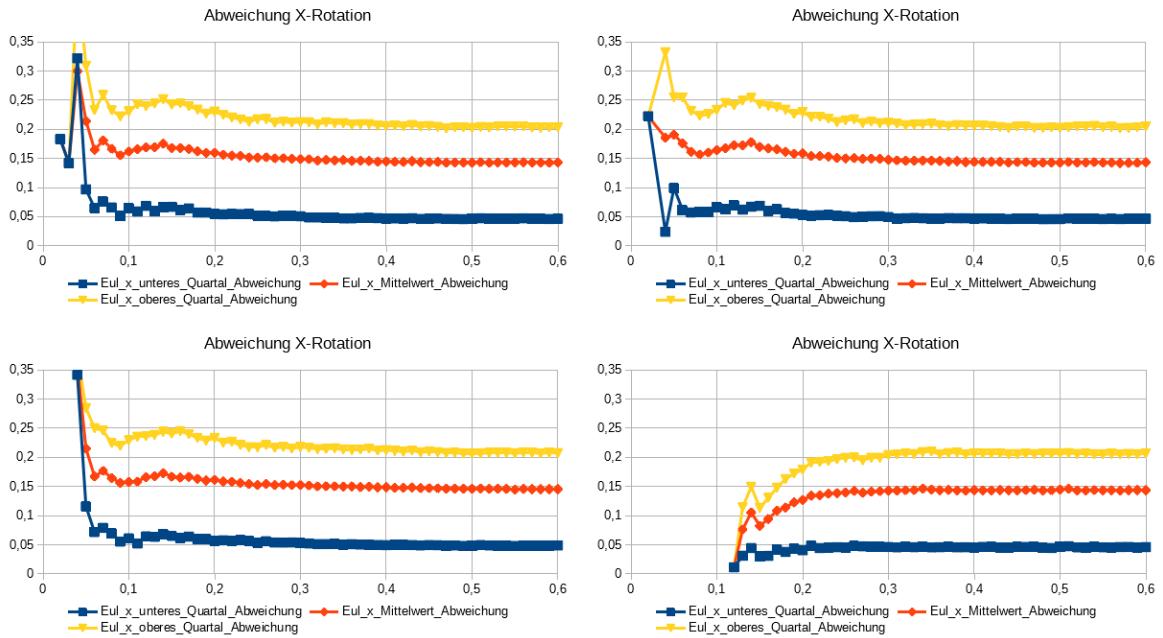


Abbildung 4.14: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in X-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

## 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 4.14. Auffällig ist außerdem der kleinere Wertebereich des Linearen-Verfahrens.

Auch bei der Y-Rotation schneidet Nearest-Neighbor am besten ab, siehe Abbildung 4.16, allerdings sind die unterscheide minimal.

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

### 4.1.7 To Do

- Patch Experts und Optimierungsfunktionen CLM

## 4.2 OpenFace auf Video

Durch das Lernen von OpenFace muss auch die Qualität auf einem Video betrachtet werden. Dazu wurde ein eigener Datensatz erstellt und ausgewertet.

Für den Versuch wurde ein Video verwendet, welches ein bewegtes Kreuz zeigt. Dieses Kreuz sollten die Probanden mit dem Blick normal folgen damit für jeden Zeitpunkt das Ziel der Aufmerksamkeit bekannt ist.

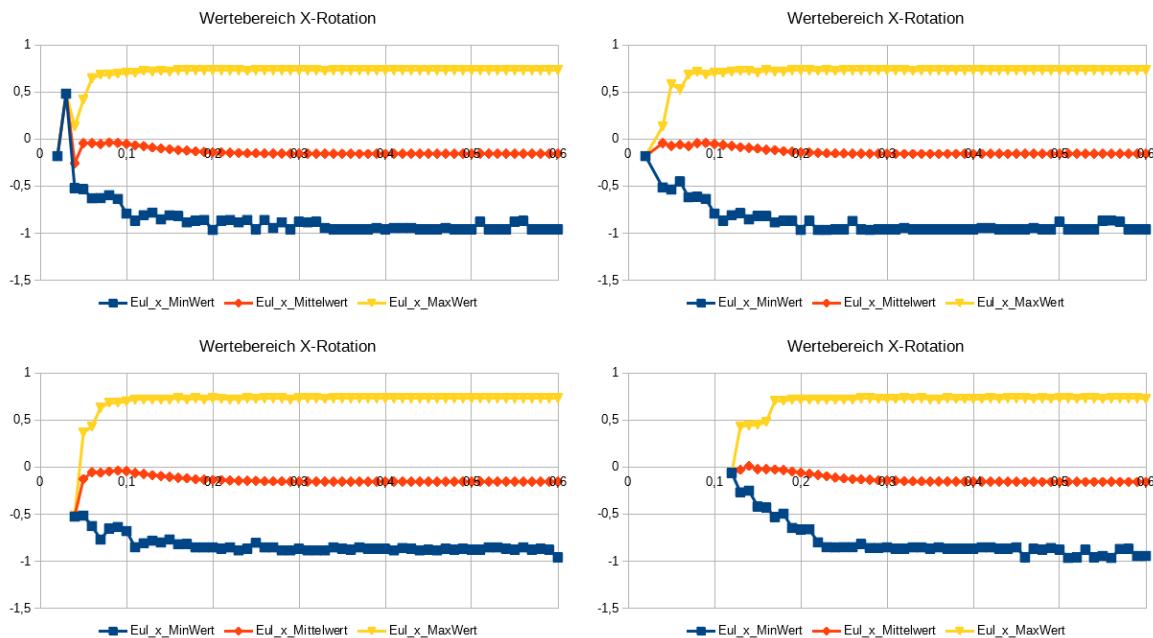


Abbildung 4.15: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in X-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

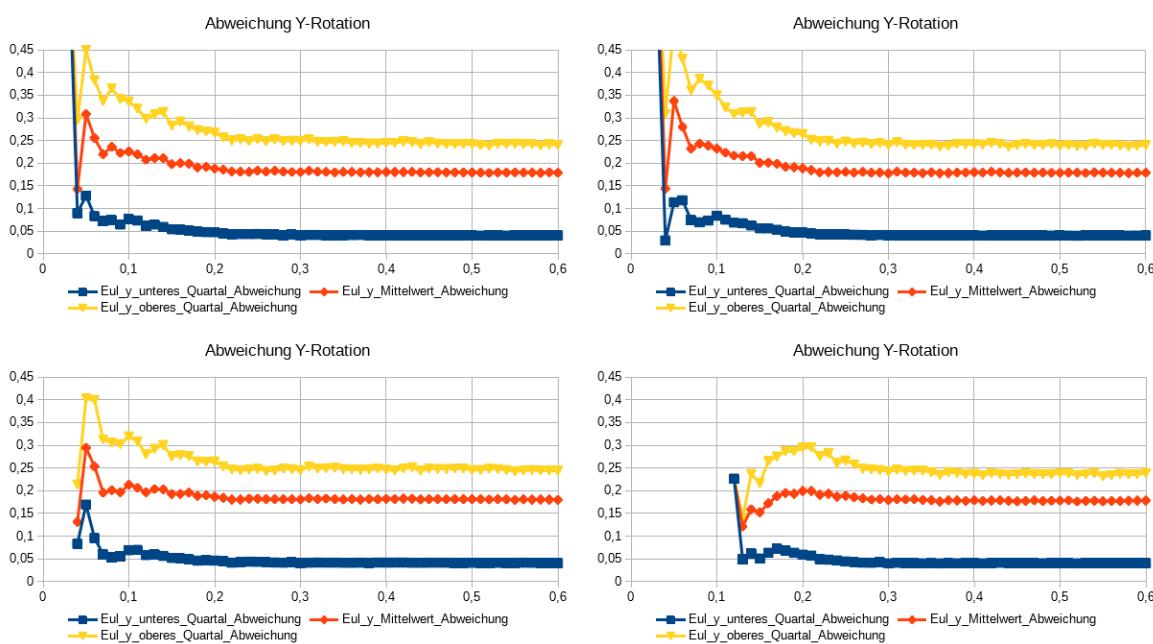


Abbildung 4.16: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in Y-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

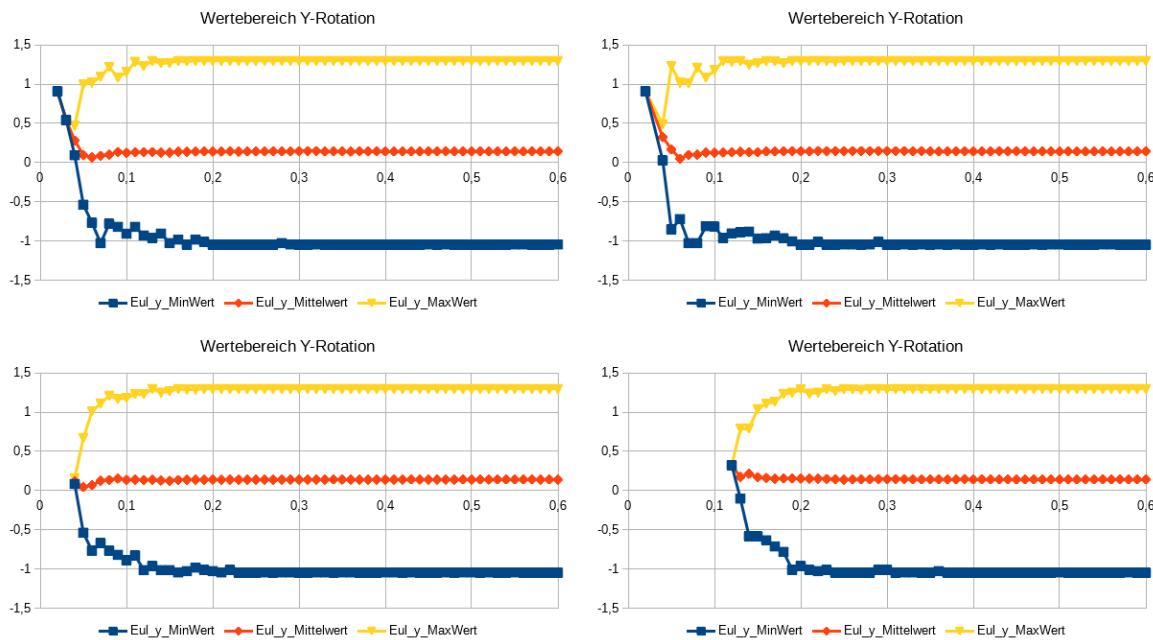


Abbildung 4.17: Zusammenhang zwischen der Skalierung (X-Achse) und der Wertebereich des Winkels in Y-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

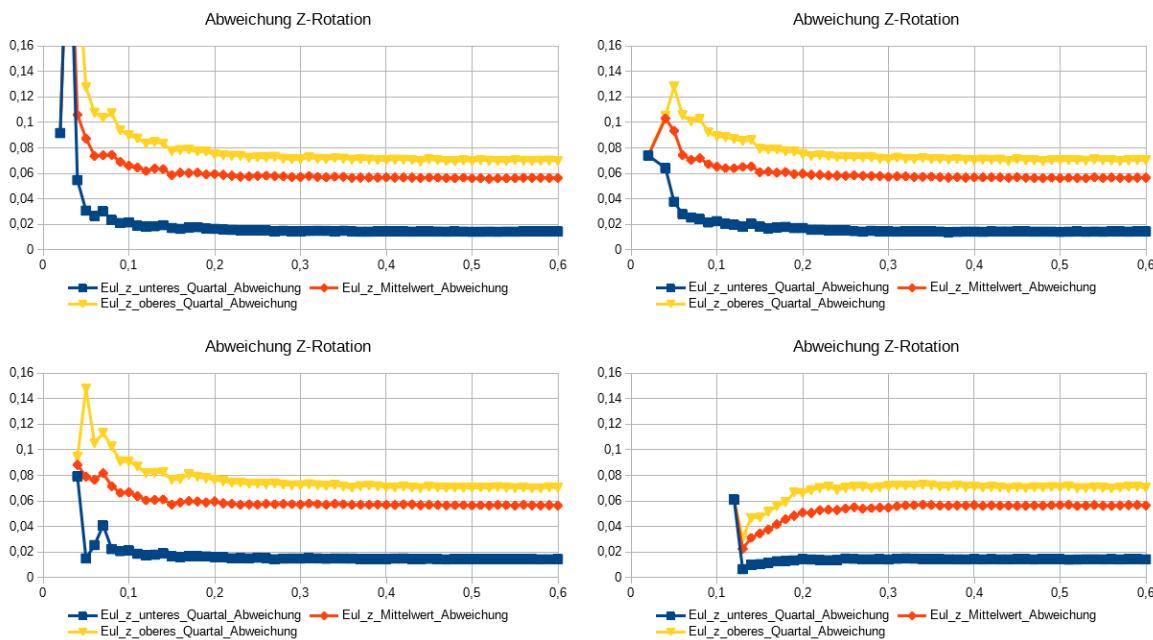


Abbildung 4.18: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in Z-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

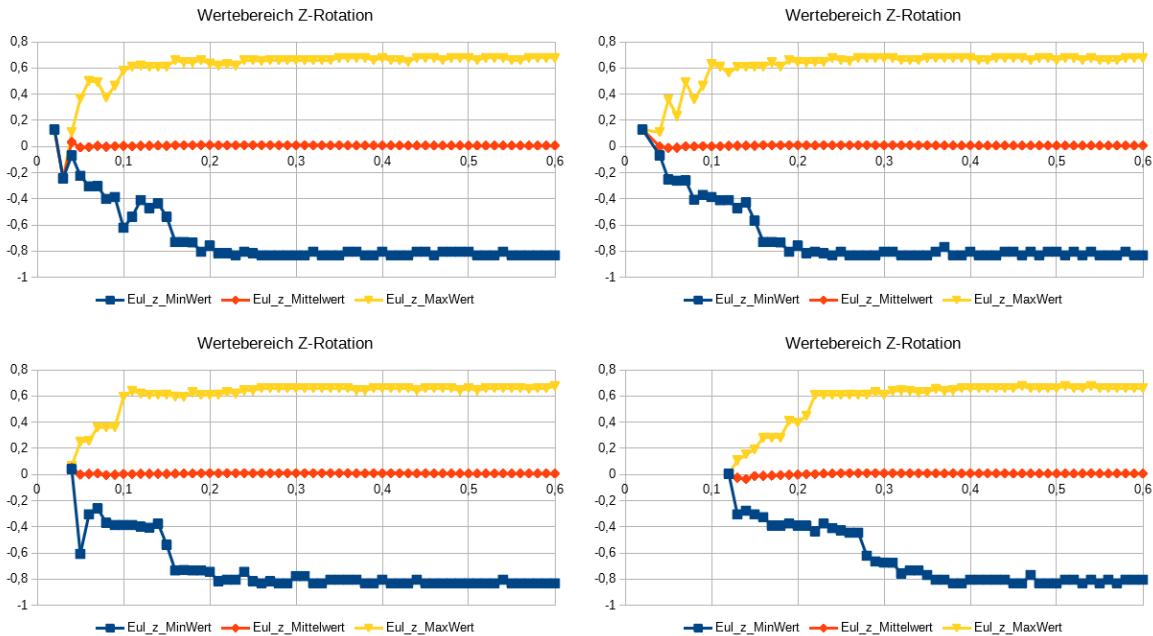


Abbildung 4.19: Zusammenhang zwischen der Skalierung (X-Achse) und der Wertebereiche des Winkels in Z-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

#### 4.2.1 Versuchsaufbau

Die Anordnung der Eckpunkte sind in Abbildung 4.20 dargestellt und wurden mittels eines Projektors auf eine Breite von  $2.88m$  und eine Höhe von  $1.49m$  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 einen 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 stellten sich etwa  $1.5m$  vor der Leinwand entfernt auf, die Kamera befand sich  $24cm$  unterhalb und  $12.5cm$  vor dem zentralen Punkt der Targets mit Blickrichtung zum Projektor und Personen.

#### 4.2.2 Versuchsdurchführung

Um die ungefähre Position des Kopfes zu ermitteln, wurde die Distanz zwischen Stirn auf dem Nasenrücken und den 4 Eckpunkten mittels eines Laserdistanzmessers bestimmt um die Position relativ zur Leinwand und Kamera ermitteln zu können. Während der Aufnahme wurde auf weitere Messung der exakten Position verzichtet.

Die 6 Probanden (5 Männlich, 1 Weiblich, 3 Brille, 5 Ohne) verfolgten das Ziel natürliche Weise.

Um die Bewegung des Punktes mit der Aufgezeichneten Kopfbewegung zu Synchronisieren, war im Kamerabild der duplizierte Bildschirm zum Projektorbild zusehen.

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

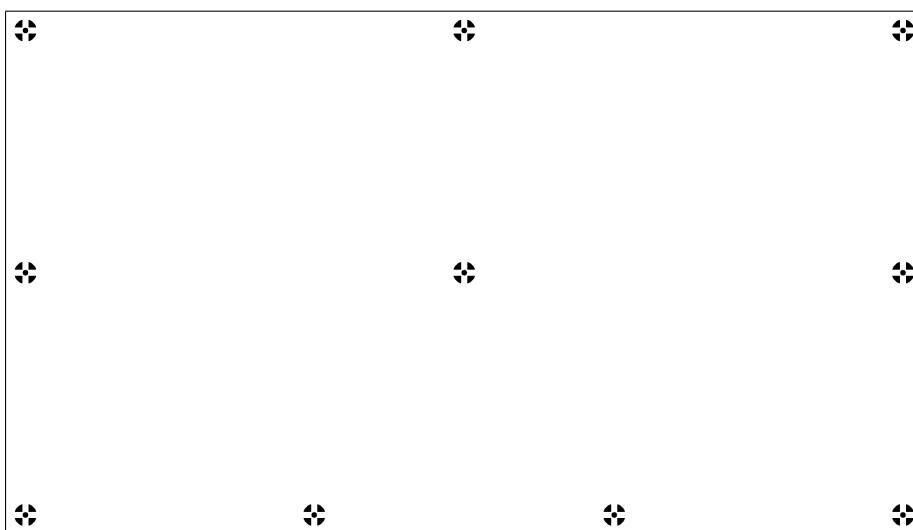


Abbildung 4.20: Eckpositionen des Bewegten Ziels bei der Videoaufnahme

### 4.2.3 Ergebnis

Die Auswertung des Versuches hat die Erwartungen und Problematiken bestätigt. Eine Verarbeitung des Videomaterials ist sogar bei sehr niedriger Auflösung noch möglich, wobei die Qualität besser sein könnte.

#### Erster Eindruck

Dargestellt in Abbildung 4.21 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.

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 wodurch zu Beginn der Bewegung, dem Target nur mit den Augen gefolgt wird, bis sich der Kopf bewegt. Dies wird so lange fortgeführt, bis die Kopfdehnung unangenehm wird und der das Ende absehbar ist, wodurch die letzten Bewegungen nur noch von den Augen gemacht werden (Quelle).

#### Qualität

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

Bei der Bestimmung des horizontalen Winkels der Kopforientierung zeigt sich, dass die bestimmten Werte im Schnitt etwas zu gering ausfallen, die Orientierung in Richtung Kamera kann zuverlässig bestimmt werden, je größer der zu messende Winkel wird, desto stärker wird auch der Fehler.

Betrachtet man in der Originalgröße die jeweiligen Quartale, so sind die Grenzen etwa  $5^\circ$  auseinander. Genug um einzelne Bereiche differenzieren zu können, jedoch zu ungenau für Berechnungen.

Bei der Bestimmung des vertikalen Winkels zeigt sich, dass 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

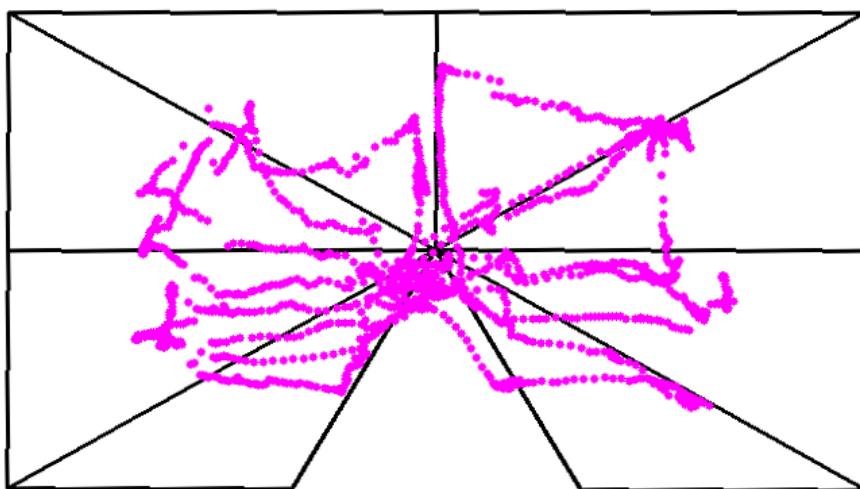


Abbildung 4.21: Dargestellt sind alle gemessene Auftreffpunkte der Gesichtsorientierung auf die Leinwand (Rosa) und des Targets (Schwarz)

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 von den Extremar sind nur etwa  $20^\circ$  auseinander liegen, bei einer eigentlichen Differenz von etwa  $90^\circ$ .

Die Auswirkung der Skalierung ist annehmbar gering, allgemein steigt die Abweichung und der Bereich der Detektion sinkt. Bei einem Skalierungsfaktor von 0.01 können die einzelnen Bereiche noch gut getrennt werden, dies entspricht eine Distanz von etwa  $14m$ . Auf der horizontalen Achse liegt der Abstand der Quartale etwa  $9^\circ$  weit auseinander.

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 dieser 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.

#### 4.2.4 Fehleranalyse des Versuches

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

##### Messung

Die erste Ungenauigkeit liegt bei der Distanz zur Leinwand, diese wurde nur vor der eigentlichen Aufnahme bestimmt. Somit ist 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 aufweist. Außerdem liegt der Ursprung des Kopfes in der Anwendung etwas Tiefer und weiter Hinten als der gemessene Nasenrücken ist. Die Parameter

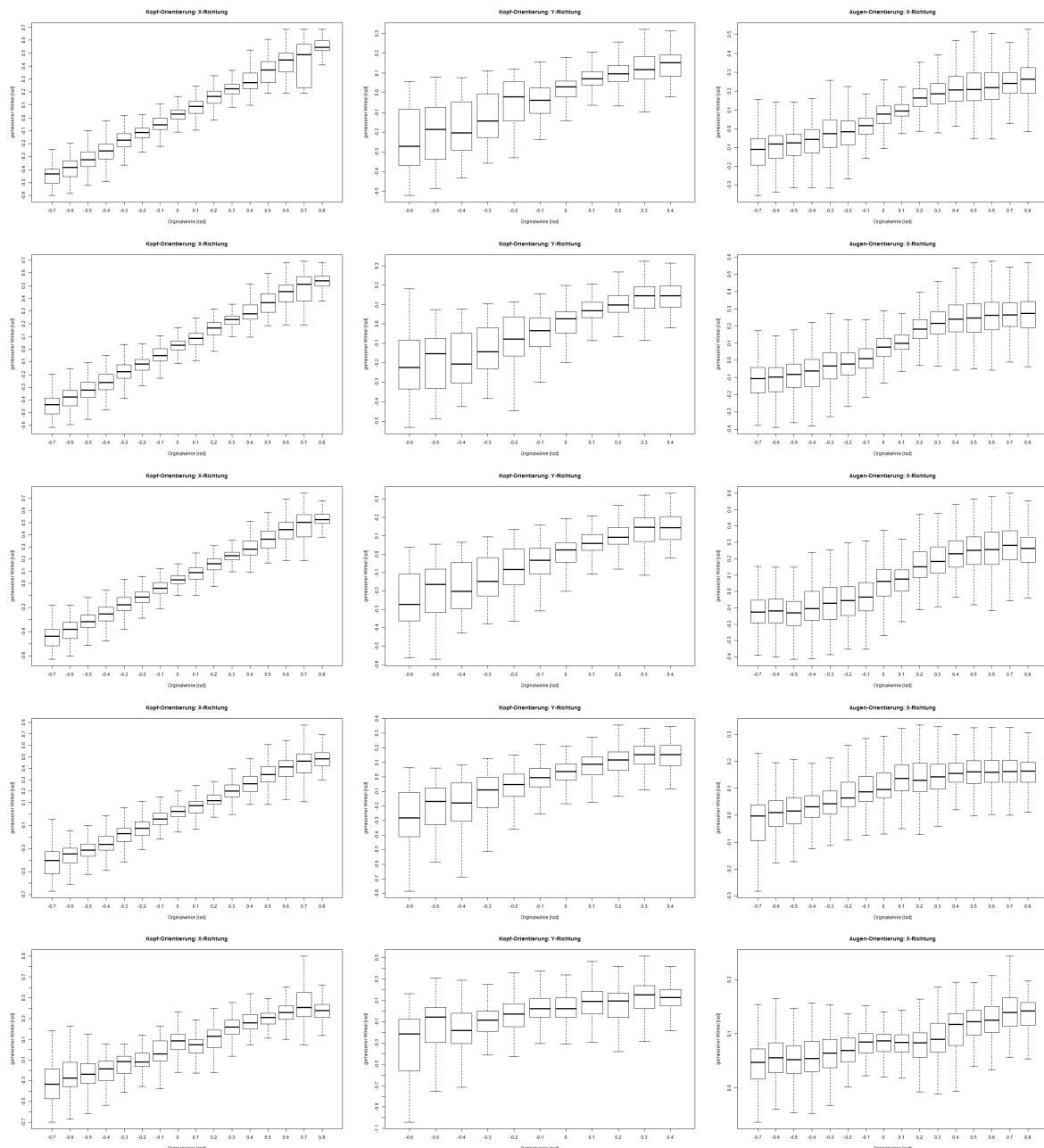


Abbildung 4.22: Dargestellt ist die Auswertung der Videoaufnahme mit der Kopfausrichtung Horizontal (Links), Kopforientierung Vertikal (Mitte) und die X-Ausrichtung der Augen (Rechts)  
Skalierungsfaktor von oben nach unten (1/0.5/0.25/0.1/0.05)

für der Überführungsmatrix von Welt- nach Kamerakoordinaten sowie die Brennweite 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 wiederum von dem direkten Licht geschützt werden musste. Somit konnte sich die Kamera nicht im Zentrum der Messpunkte befinden.

Da die Kamera und die Leinwand fest Montiert sind, ergibt sich auch die Problematik das der Kopf der Probanden ebenfalls nicht im Zentrum des Kamerabildes Befinden 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.

### **Umgebung**

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

Ein weiteres allgemeines Problematik zeigt sich auch wieder bei der Auflösung des Gesichtes, somit ist eine Berechnung auf dem Gesicht zwar möglich, auf den Augen allerdings nicht.

Somit ergibt sich ein weiteres Problem, da im allgemeinen eine Exkursionen, der Winkelbereich der üblichen Augenbewegungen, bis etwa  $20^\circ$  stattfindet und diese nicht erfasst werden können.

Ein weiteres nicht zu verachtendes Problem ist die Reflektion vor allem auf den Brillen, von den starken Lichtquellen wie z.B. Fenster, Projektor- und dessen Bild, sowie Lampen die Pupille verdecken. Auch Schatten gerade bei den Augenhöhlen erschweren die Auswertung.

- Bild für den Versuchsaufbau

### **4.3 Fehleranalyse**

Mit entsprechend hochauflösenden Kameras können auch bessere Resultate auf größerer Distanz erreicht werden. Gerade die Bestimmung der Blickrichtung ist meist nicht möglich, da die Augenpartie viel zu klein für eine Berechnung ist. So bleibt meist nur die Gesichtsorientierung mit ihrer 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, der vor der Kamera steht oder sich der Kopf zu weit wegdreht und das Tracking nicht mehr möglich ist.. Aber auch die Frisuren spielen eine Rolle, da dadurch diese einige Landmarks verdeckt werden und so das Gesicht nicht erkannt wird wie z.B. die Augenbrauen.

Eine Lösungsansatz währen Landmarks in Profilbildern zu detektieren und sowie das verwenden von weiteren Kameras aus anderen Perspektiven.

### **4.4 Verbesserungen**

- Mehrere Kameras für 3D und weniger verdecken und wegdrehen

# Literaturverzeichnis

- [App15] Johannes Appel. *Die Bedeutung der Aufgaben für das Beteiligungsverhalten der Schüler : eine Videostudie zur Wirksamkeit des Unterrichtsprozesses*. PhD thesis, 2015.
- [bau13] 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.
- [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.
- [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.
- [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.
- [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. Lanczos-filter — wikipedia, die freie enzyklopädie, 2016. [Online; Stand 6. Mai 2017].
- [Wik17a] Wikipedia. Opencv — wikipedia, die freie enzyklopädie, 2017. [Online; Stand 16. Juni 2017].
- [Wik17b] 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.