

# 1 Einführung

## 1.1 Intension

Zur Bewertung der Qualität des Unterrichtes, wird die Aufmerksamkeit der Schüler verwendet, da zwischen beiden ein Zusammenhang besteht. Allerdings ist der Parameter Aufmerksamkeit recht schwierig zu erfassen, wodurch verschiedene Verfahren verwendet werden. Unter anderem Fragebögen die ein Schüler selbst ausfüllen sollen oder die Auswertung durch einen Beobachter der bewertet ob ein einzelner Schüler Aufmerksam (on-Task) oder nicht (off-Task) ist.

Für die Bewertung von on/off-Task werden Kriterien festgelegt, wie Blickrichtung, Körperhaltung und Tätigkeit, dann wird die Person beobachtet wie diese sich verhält.

Bei der Videostudie zur Wirksamkeit des Unterrichtsprozesses 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 wird immer in einem Minuten-Intervalle der Schüler beobachtet und bewertet. Sollte drei oder mehr Punkte erfüllt sein, gilt der Schüler als on-Task.

Diese Art der Auswertung ist recht einfach, allerdings gibt es Interpretationsfreiheiten, gerade bei den Bewertungen der Tätigkeiten, die von jedem Beobachter anders ausgelegt werden kann. Außerdem ist diese Art der Bewertung sehr zeitintensiv, so werden alleine zum anschauen des Videos für jeden Schüler bei einer Klassen (30 Personen) 30 min gebraucht und sollte jeder Schüler mehrfach beobachtet werden entsprechend mehr.

Werden nur wenige Zyklen durchgeführt, so wird das gesamte Verhalten eines Schülers in der Unterrichtsstunde mit nur wenigen beobachteten Minuten beschrieben und ist aber sehr ungenau, die Auswertung benötigt allerdings entsprechend weniger Zeit. Soll sie genauer sein, so muss deutlich mehr Zeit investiert werden.

Durch eine zu geringe Auswertungsrate kann nur eine Aussage über den gesamten Unterricht gemacht werden und nicht über einzelne Übungen oder ähnliches, auch die Bewertung eines einzelnen Schülers ist nur schwer möglich.

[App15]

## 1.2 Problemstellung

Die aktuellen Verfahren zur Analyse von Aufmerksamkeit im Unterricht werden meist äquivalent zu Abschnitt 1.1 durchgeführt. Der jeweilige Schüler wird von einer Person beobachtet und dann nach bestimmten Kriterien bewertet. In die Bewertung fließt allerdings auch die Meinung/Auffassung des Bewerters ein und ist daher nicht objektiv.

Diese Art der Auswertung ist recht ungenau und Arbeitsintensiv, da sie von einer Person ausgeführt werden muss. Ein wichtiger Parameter ist die Blickrichtung des einzelnen Schülers, da sie meist dorthin gerichtet ist, wo auch die Aufmerksamkeit liegt.

Ziel dieser Arbeit ist es nun, mit möglichst geringem Aufwand an Hardware eine Bestimmung der Blickrichtung einer ganzen Klasse vorzunehmen. Die Messung soll den Unterricht möglichst wenig beeinträchtigen, wodurch Eye-Tracking Brillen nicht verwendet werden, wegen den Kosten und der

Ablenkung. Auch der Aufbau soll recht einfach und für Laien anwendbar sein, somit wird nur eine einzige festmontierte Kamera vor der Klasse eingesetzt.

Für diese Anforderungen soll nun ein Verfahren entwickelt werden, mit dem es möglich ist, das Filmmaterial von einer gesamten Klasse auf einmal auszuwerten, um von allen Personen die Blickrichtungen bzw. die Gesichtsorientierung während einer Schulstunde zu bestimmen.

[HR92]

## 1.3 Gesetzte Bedingungen der Anwendung

Damit der Unterricht, wie im Szenario der Problemstellung beschreiben Abschnitt 1.2, möglichst wenig beeinflusst wird, ergeben sich folgende Randbedingungen:

- Brillen, Kontaktlinsen und ähnliches sind erlaubt.
- Die üblichen Bewegungen im Unterricht wie Sprechen, Kopfdrehungen usw. der Schüler sind gestattet.
- Es soll gleichzeitig auf Distanzen zwischen  $2.5 - 8m$  zur Kamera auf einer Breite von  $6m$  funktionieren.
- Möglichst alle Blickrichtungen der Schüler sollen so exakt wie möglich erfasst werden.

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

Außerdem soll die Anwendung auf schon vorhanden Aufnehmen eines Unterrichtes arbeiten, bei denen oben genannten Bedingungen erfüllt sind.

[bau13]

### 1.3.1 Randbedingungen der Anwendung

Des weiteren werden folgende Annahmen gemacht:

- Die Szene ist innerhalb eines Gebäudes, mit ausreichend gleichmäßiger Beleuchtung.
- Die Überführung zwischen Welt- und Kamerakoordinatensystem bekannt.
- Die Kamera befindet sich vor der Klasse, so dass die Hauptblickrichtung der Schüler in ihrem Fokus liegt.
- Die Gesichter sind komplett sichtbar und nicht verdeckt.

Natürlich sind auch alle inneren Parameter der Kamera bekannt.

## 1.4 Hardware

Als Messinstrument wird nur eine einzelne Farbkamera eingesetzt. Das Videomaterial der Schulklassen wurde mit einer unbekannten Videokamera aufgezeichnet, daher sind nur die Parameter des Filmes ( $640 \times 480$  mit  $25Fps$ ) bekannt.

Für die Messungen im Versuch wurde die Explorer 4K Action Camera verwendet, sie besitzt eine  $170^\circ$  Weitwinkel-Linse mit großem Field of View. Mit der 2.7K Einstellung wird ein  $2688 \times 1520$  Video mit 30FPS aufgezeichnet.

Sowie die Logitech c920 HD Pro Webcam, diese liefert ein  $1600 \times 896$  Farbbild mit  $15Fps$ . Beide wurden für die Aufnahmen fest montiert und die relevanten Parameter bestimmt.

## 1.5 Software

Für die Umsetzung werden folgende Software-Elemente aus fremder Quelle verwendet.

### 1.5.1 ElSe

Ellipse Selection for Robust Pupil Detection in Real-World Environments, ein Algorithmus zur Bestimmung der Pupille in einem Bild. Der ursprüngliche ElSe-Algorithmus ist für Graubilder mit Infrarotbeleuchtung ausgelegt, wurde für diese Anwendung angepasst um Farbbilder verarbeiten zu können. Entwickelt von der Uni Tübingen. [WF16]

### 1.5.2 MTCNN Face Detection

Multi-task Cascaded Convolutional Networks, ein Algorithmus zur Detektion von Gesichtern und Bestimmung von 5 Gesichts-Landmarks in Farbbildern. Dabei werden drei CNN auf einer Bildpyramide angewendet um so zuverlässig Gesichter mit verschiedenster Größe zu erkennen. [KZ15]

### 1.5.3 OpenCV

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

[Wik17a][BK08]

### 1.5.4 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.

Entwickelt von der University of Cambridge [TB16]

## 2 Theorie & Grundlage

### 2.1 Grundlagen

Die Gesichtserkennung ist Teil der Bildverarbeitung und wird ständig weiterentwickelt. Darunter fallen neben der Detektion des Gesichtes auch seine Analyse wie Orientierung oder das erkennen von Mimiken und Übereinstimmungen.

#### 2.1.1 Künstliches neuronales Netz

Ein künstliches neuronales Netz besteht aus miteinander verknüpften künstlichen Neuronen. Jedes Neuron erhält Eingangswerte, diese werden individuelle Gewichtet, mittels einer Übertragungsfunktion zusammengefasst und durch eine Schwellenwertfunktion das Ergebnis bestimmt.

Um die Parameter der Neuronen zu bestimmen, werden sie zufällig initialisiert und dann so angepasst, dass sie zu einer gegebenen Eingabe das gewünschte Ergebnis liefert und der Fehler über dem gesamten Trainingsdatensatz minimal ist.

#### To Do

Quelle

#### 2.1.2 Convolutional Neural Network (CNN)

CNN ist eine Weiterentwicklung der neuronalen Netzen und werden zur Klassifizierung verwendet, unter anderem im Bereich Bild- und Spracherkennung. Dies wird durch eine gewichtete Faltung der Eingabe erreicht und sind state of the art bei vielen Anwendungen.

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 ja 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, der Feature Extraktion und der Klassifizierung. Bei der Feature Extraktion werden verschiedene Kernel und Komprimierung die Eingabeinformationen verwendet um sie für den zweiten Teil auf zu bereiten. Gelernt werden können jeder einzelne Kernel für sich und die jeweiligen Bewertungen der Kernel und den Neuronen.

#### To Do

Quelle  
Bild

### 2.1.3 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 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 verschiedenen leicht verformbaren Objekten wie Gesichter und daher zuverlässiger als das Active Appearance Model (AAM).

#### To Do

Quelle  
zur Detecton der Landmarks

### 2.1.4 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 einem Bild die Übereinstimmung bestimmt werden, wird versucht es bestmöglich in die typische Form zu überführen. Sind dennoch Unterschiede vorhanden, liegt diese an der Erscheinung des Objektes.

[Wik14]

### 2.1.5 PDM & GAVAM

Mit Point Distribution Model (PDM) können verformbare Objekte recht gut modelliert werden. Dabei wird die durchschnittliche Form  $\bar{X}$  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 Formen  $X$  des Objektes dargestellt werden. Zur Vereinfachung reicht es, die signifikantesten Eigenvektoren in  $P$  auf zu nehmen und dennoch  $X$  ausreichend genau zu beschreiben.

Ist bekannt welche Art der Verformung durch den Eigenvektor dargestellt ist, z.B. eine bestimmte Orientierung, so kann anhand des Skalierungsvektors die Rotation des berechneten Objektes 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]

### 2.1.6 Non-maximum suppression (NMS)

Ein Verfahren um Kanten in einem Bild exakter zu bestimmen. Dabei wird der Farbwert des Pixels mit dem umliegenden verglichen und sollte es nicht maximal sein auf Null gesetzt. Auf diese Weise bleibt nur noch ein Kantenpixel übrig.

## To Do

Quelle

## 2.2 MTCNN Face Detection

Bei Multi-task Cascaded Convolutional Network handelt es sich um ein Verfahren dass bei der Detektion von Gesichtern auch deren Ausrichtung berücksichtigt wird, um so bessere Ergebnis zu erzielen.

### 2.2.1 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 wird auf recht großen Bild gearbeitet mit verhältnismäßig kleinen und verschiedenen großen Gesichtern.

Außerdem sollte das Verfahren ausreichend schnell sein, da es sich hierbei nur um ein Vorverarbeitungsschritt handelt und zur Beschleunigung der späteren Berechnung beitragen soll.

### 2.2.2 Die 3 Stufen der Verarbeitung

Für die gute Detektionsqualität sorgt die dreistufige Verarbeitung auf der 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 bekannt ist.

Dies ist von Vorteil, damit das CNN auf eine feste Größe von Gesichtern optimiert werden kann, um neben dem möglichen Farbverläufen, durch die Skalierung das Lernen nicht zusätzlich zu erschweren.

#### Stufe 1

Beim ersten Verarbeitungsschritt werden alle Bereiche eines Bilds gesucht, in denen möglicherweise ein Gesicht zu sehen ist. Dazu wird zuerst ein einfaches CNN eingesetzt und die Ergebnisse, die sich sehr stark überlappen, zusammengefasst.

Für die Detektion wird von einem CNN, dem sogenannte Proposal Network (P-Net), eingesetzt und sehr viele Bounding-Boxen ermittelt. Diese werden nun mit einem NMS ausgedünnt, um die am stärksten überlappenden Bounding-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 5 Landmarks ermittelt werden.

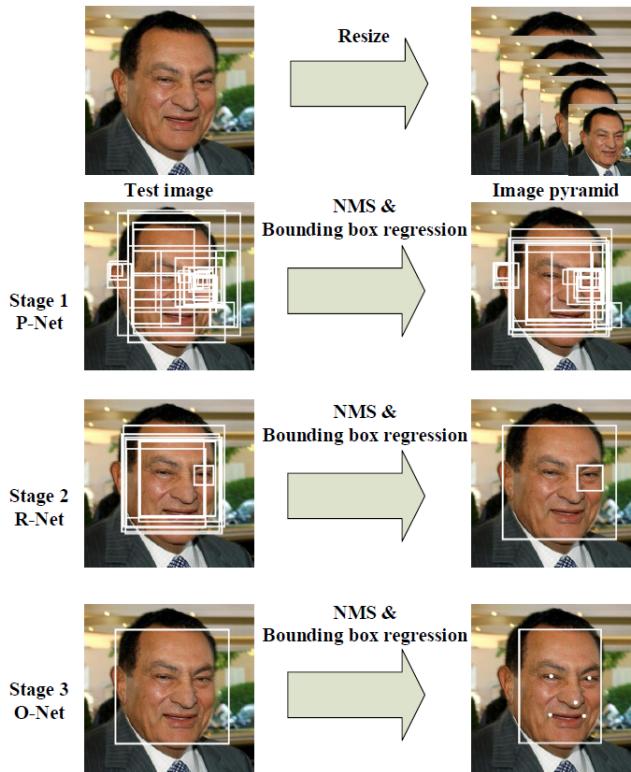


Abbildung 2.1: Darstellung des Funktionsablaufes von MTCCN[KZ15]

### 2.2.3 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  $20 \times 20$  enthalten und wurden erfolgreich erkannt.

Somit sind alle Anforderungen erfüllt um mit diesem Verfahren den vorhanden Frame für die nachfolgenden Berechnungen vorzubereiten, daher wird es auch hier eingesetzt.

## 2.3 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.

### 2.3.1 Nearest-Neighbor

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 2.3.

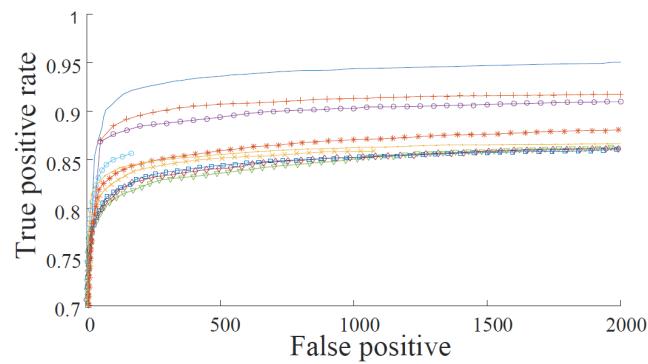


Abbildung 2.2: normale blaue Linie[KZ15]

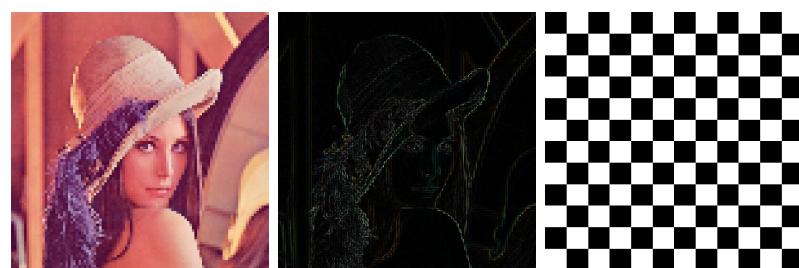


Abbildung 2.3: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels Nearest-Neighbor auf 512 Pixel vergrößert und bei Lena die Differenz bestimmt



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 bestimmt

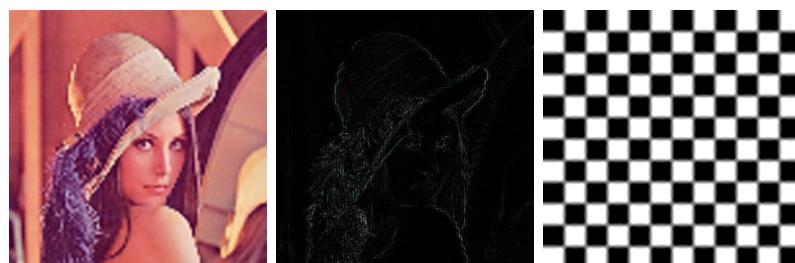


Abbildung 2.5: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels bikubischem Verfahren auf 512 Pixel vergrößert und bei Lena die Differenz bestimmt

### 2.3.2 Linear

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 2.4.

### 2.3.3 Bicubic

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 2.5 zu sehen. [Wik16a]

### 2.3.4 Lanczos

Dieser Filter basiert auf einer Sinc-Funktion über einen Bereich, um so eine Bewertung der benachbarten Pixelwerte zu erhalten. Somit ergibt sich der neue Farbwert aus den bewerteten umliegenden Pixeln, wobei durch hie Fehler entstehen können, siehe 2.6. Die Funktion kann und wird für die Anwendung auf einen  $8 \times 8$  Bereich begrenzt. [Wik16b]

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



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

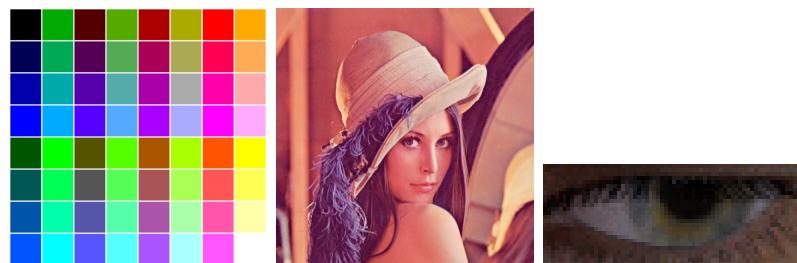


Abbildung 2.7: Dies sind die Eingabebilder der verschiedenen Konverter von Farbe nach Grau. Links eine Farbpalette, Mittel Lena und Rechts ein Augenausschnitt aus dem Augendatensatz [WBZ<sup>+</sup>15]

## 2.4 Farb- nach Grau-Bild

Da die Berechnungen von ElSe auf Grau-Bildern arbeitet und das Eingabebild in Farbe ist, muss es in ein Grau-Bild umgewandelt werden. Dabei soll vor allem der Farbunterschied zwischen Pupille und der Umgebung maximal sein.

Die Problematik bei der Wahl des Verfahrens liegt in der Anforderung, da die Pupille möglichst dunkel sein soll und das restliche Auge hell. Die Farbe der Iris erschwert die Differenzierung da, wenn sie recht dunkel ist, der Grau-Wert zur Pupille entsprechend gering ist. 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 muss.

Um die Auswirkung der verschiedenen Farb- nach Grau-Bilder Konverter zu ermitteln wurden einige Verfahren verwendet um ihre Auswirkung auf die Detektion zu ermitteln. Nach der Umwandlung von Farb- nach Grauwert wird für die Anwendung das Grau-Bild noch normiert, damit Mindestens ein schwarzes und ein weißes Pixel vorhanden ist.

### 2.4.1 Luminance

Dies ist ein lineares Verfahren, das der menschlichen Farbwahrnehmung entspricht. Eine Gamma-Korrektur wird bei der Umwandlung nicht verwendet, siehe Abbildung 2.8.

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 Farb- nach Grau-Bilder.

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



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



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

## 2.4.2 Gleam

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, nur noch ein grauer Bereich vorhanden ist.

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

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

## 2.4.3 Gleam-New

Dies ist eine verbesserte Variante von Gleam bei dem zuerst das gesamte Bild analysiert wird um die Parameter für die jeweilige Gamma-Korrektur zu ermitteln.

Durch die individuelle Veränderung der Farbkanäle, werden Farbunterschiede minimiert und somit alle stark farbigen Bereiche ebenfalls dunkel dargestellt. Somit wird der Kontrast zwischen der farbigen Iris und dem weißen Auge verbessert, siehe Abbildung 2.10.

Da allerdings alle Farben dunkel werden, entstehen weitere dunkle Bereiche die die Detektion der Pupille beeinträchtigen können. Ein kleiner Nachteil ist die vorige Analyse des Bildes um die Parameter



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

entsprechend passend wählen zu 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.

#### 2.4.4 Quadrat

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 klarer zu sehen sein sollte, der Farbunterschied zur Iris allerdings ebenfalls verringert wird.

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

#### 2.4.5 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 Farben 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 bleiben Hell und nur gleichmäßig dunkel Bereiche bleiben dunkel wie es bei schwarz der Fall ist.



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

Wenn der Minimalwert anstelle verwendet wird, bleiben nur gleichmäßig helle Bereiche hell, alles andre wird abgedunkelt.

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

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

#### 2.4.6 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 verwendet werden als Extremwert. Mit dem Ergebnis, dass die Pupille gleichmäßiger dunkler wird und Pixel die eine Reflektion darstellen ignoriert werden und somit das gesamte Bild stärker aufgehellt wird.a

### 2.5 OpenFace

Die Aufgaben von OpenFace ist die Analyse der Gesichtes basierend auf Bildern. Dabei sind für die Anwendung nur Kameraparameter bekannt und keinerlei Zusätze wie eine Tiefenbild oder Infrarotbeleuchtung der Szene vorhanden. Dabei ist für die Anwendung die Kopfposition (Translation und Orientierung) und Blickrichtung von Interesse, da mit ihnen zurückrechnet werden kann wohin die Person schaut.

OpenFace kann neben den Landmarks auch die Position, Blickrichtung und Gesichtsmerkmale bestimmen, basierend auf einem einfachen Bild. Sollte ein Video als Quelle fungieren, so kann OpenFace auch lernen. Somit sind die Resultate basierend auf Videos besser als auf einfachen Bildern.

### 2.5.1 Verarbeitungsschritte

Der Rechenaufwand ist so ausgelegt, dass alle Berechnungen auf einer Webcam in Echtzeit ausgeführt werden können, dies ist im aktuellen Fall nicht notwendig, da es sich um eine nachträgliche Auswertung handelt. Durch den Aufbau sind nur recht kleine Farbbilder der Gesichter in einem Video vorhanden wodurch eine Auswertung erschwert wird.

#### 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 ausgewählt. Bei den initiale Hypothesen handelt es sich um verschiedene Gesichtsorientierungen auf denen verschiedene Netze gelernt wurden. Dies ist zwar langsamer, aber auch exakter als eine einfache Hypothese. Wird ein Tracing 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 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 Brechung auf den Gesichtern sollten sie eine Mindestbreite von 100 Pixeln für eine zuverlässige Detektion Originalgröße besitzt.

#### Bestimmung der Gesichtsposition

Zur Bestimmung der Translation und Orientierung des Gesichtes wird ein CLNF bzw. PDM eingesetzt. Dabei wurde es mit der Kameraabbildung der 3D-Landmarks eines Kopfes in verschiedenen Positionen initialisiert. Womit auf eine Normierte Abbildung gerechnet wird, diese kann mit den passenden Kameraparameter für die Aufnahme angepasst werden um die reale Position zu bestimmen. Sind keine Parameter bekannt, so können diese anhand der Bildauflösung geschätzt werden.

Bei der Schätzung der Brennweite für ein Bild mit einer Dimension  $I_x \times I_y$  wird das Standardobjektiv mit 50 mm und 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

Durch die Landmarks der Augen werden die Augenlider, Iris und Pupille dargestellt und für jedes Auge separat bestimmt. Dabei wird der Augenbereich, basierend auf dem detektierten Gesicht, verwendet, um mit einem weiten CNN die 28 Landmarks des Auges zu bestimmen.

Zur Bestimmung der Blickrichtung wird wie folgt vorgegeben. 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. Nun 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.

#### **2.5.2 Veröffentlichte Genauigkeit**

Die Messung wurde auf dem Biwi Kinect head pose und BU Datensatz ausgeführt. Für die Genauigkeit der Kopfposition haben sich folgend Werte ergeben in Grad:

	Yaw	Pitch	Roll	Mean	Median
Biwi Kinect [FGG11]	7.9	5.6	4.5	6.0	2.6
BU dataset [CSA00]	2.8	3.3	2.3	2.8	2.0
ICT-3DHP [BRM12]	3.6	3.6	3.6	3.6	-

Für die Qualität zur Bestimmung der Blickrichtung ergab sich ein durchschnittlichen Fehler von 9.96 Grad.

## **2.6 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.

#### **2.6.1 Funktion**

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

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

#### **2.6.2 Funktionsablauf**

Als Input wird im Original ein Graubild verwendet, auf dem das Infrarot beleuchtete Auge abgebildet ist. Für den Test im Vergleich zu anderen Verfahren, wurden Bilder von  $384 \times 288$  Pixel Größe verwendet und ist auf diesen Echtzeit fähig.

#### **Kantendetektion**

Da die Pupille als schwarzen Fleck sichtbar 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, alle anderen können ignoriert werden.

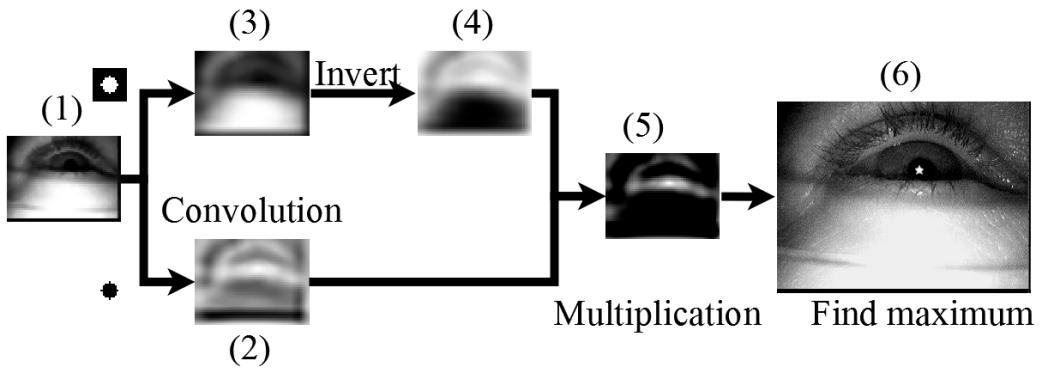


Abbildung 2.13: Ablauf der alternativen Berechnung zur Pupillen-Detektion

### 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 Anforderungen 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 so bestimmten Ellipsen, wird als Lösung verwendet.

### Grobe Bestimmung - To Do

Sollte die Bestimmung der Ellipse 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 Abbildung 2.13 (3). Zur zweiten Faltung mit Ergebnis Abbildung 2.13 (2) wird der negative Durchschnitt über ein Quadrat ohne inneren Kreis eingesetzt, wobei der Mean- und negativen Mean-Filter den selben Radius haben.

Nun wird das Ergebnis des Mean-Filters invertiert Abbildung 2.13 (4) und mittels Punkt-Multiplikation mit dem negativen Meanfilter zusammengebracht Abbildung 2.13 (5). In diesem Bild, wird nun der Höchste Wert gesucht, da dies das Zentrum des dunkelsten kreisförmigen Ortes im Bild ist.

Ergebnis des Bsp ist als Kreuz in Abbildung 2.13 (6) markiert.

### 2.6.3 Ergebnisse

Im Vergleich zu anderen Verfahren im Test, zeigt sich das ElSe in den meisten Fällen als Sieger hervorgeht mit einer Verbesserung der Erkennungsrate um 14.53%.

Ein Problem entsteht wenn der Farbunterschied zwischen Iris und Pupille recht gering ist, oder durch Reflexionen der Kantenverlauf gestört wird.

Für die Anwendung ist der Bereich der Augen sehr klein und eine klare Detektion einzuordnen schwierig, wodurch vor allem dieser Teil für die Anwendung von Interesse ist.

To DO: Quellen für Vergleich

## 2.7 Berechnung der Position

Zur Bestimmung der Position  $P = (X_{avg}; Y_{avg}; Z_{avg})$  des Gesichtes im Kamerakoordinaten wird die Größe, ein Skalierungsfaktor  $S$ , des Kopfes im Bild verwendet.

Da bei der Abbildung von den Koordinaten ins Bild gilt  $x = f \cdot \frac{X}{Z}$  und  $y = f \cdot \frac{Y}{Z}$ , somit 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 mit einem Durchschnittlichen Kopfgröße gerechnet wird.

### 2.7.1 Zusammenhang Bildposition & Weltposition

Als Ausgangspunkt werden die Ergebnisse des CNN eingesetzt um mit deren Hilfe wie in Abschnitt 2.7 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.

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

Eine weitere Verbesserung kann erreicht werden, indem die gefunden 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.

# 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 erkennen kann. Abschnitt 3.2

Für die weiteren Berechnungen muss bekannt sein welchen Bereich das Gesicht in Frame einnimmt und um welches es sich handelt. Der Bereich wird vom MTCNN als Box geliefert, als Zuordnung zur Person wird ein Matsching zum vorigen Frame verwendet.

Damit auch eine Berechnung auf den kleineren Gesichtern stattfinden kann, werden alle zu kleinen Bildbereiche hochskaliert. Dabei muss wegen Ungenauigkeiten die gefundene Box etwas räumlich vergrößert und dann auf eine Mindestgröße gebracht werden. Abschnitt 3.3

Diese Bildbereiche werden nun mit OpenFace weiterverarbeitet, um die Position der Landmarks im Bild zu bestimmen. Durch die Berechnung auf der selben Person kann das CNN sich auf jeden einzeln einstellen, um so bessere Ergebnisse zu erreichen. Außerdem könne alle gefundenen Personen gleichzeitig (parallel) ausgewertet werden. Abschnitt 3.4

Bei großen Gesichtern wird nun ElSe auf den Augenbereich angewendet, um die Position der Pupille noch exakter zu ermitteln, damit die Blickrichtung genauer wird. Dazu muss allerdings die Differenz zwischen ElSe-Ergebnis und OpenFace-Ergebnis betrachtet werden um Fehler zu erkennen. Abschnitt 3.5

Nun wird auf Basis der Landmarks und der Kameraparameter die Position und Orientierung des jeweiligen Gesichtes bestimmt und können für weitere Anwendungen verwendet werden. Abschnitt 3.6

## 3.2 Detektion der Gesichter

Da nur eine einzige fest montierte Kamera ohne Zoom eingesetzt wird, muss sie 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 aus und diese müssen noch Nachbearbeitet werden. Siehe Abschnitt 3.3

Für die automatische Detektion wird Face-MTCNN Abschnitt 2.2 eingesetzt, da dieses Verfahren die meisten Gesichtern mit verschiedenen Größen im selben Bild findet, sogar recht kleine mit  $20 \times 20$  Pixeln. Bei diesem Schritt müssen alle Gesichter gefunden werden, auf denen die Berechnung stattfinden soll. Dabei muss das gesamte Gesicht in der Box sein, ansonsten muss es nicht sehr exakt sein, da OpenFace einen eigenen Facedetector besitzt. Wird MTCNN-Face dedector eingesetzt hat sich eine Vergrößerung der Box um 30% als sinnvoll erwiesen, damit sichergestellt wird, dass alle Merkmale wie Nasenspitzen, Kinn, Augenbrauen usw. sicher im Bildausschnitt enthalten sind.

Ebenfalls in diesem Schritt werden die einzelnen Boxen den Personen zugeordnet, damit im späteren Verlauf das korrekte CNN verwendet wird. Für die Zuordnung reicht meist einen einfache Übereinstimmung der aktuellen Box zum vorigen Frame, da die Gesichter sich meist weder groß Bewegen noch sich die Boxen überlappen.

Damit auf allen Gesichter gerechnet werden kann, ist eine Semiautomatische Korrektur erforderlich damit Falsch-Detectionen entfernt und fehlende Boxen ergänzt werden können. Alle nicht gefundenen Gesichtern können manuelle gesetzt oder zwischen dem letzten und nächsten Frame interpoliert werden.

Die gefundenen 5 Landmarks sind für die nachfolgende Berechnung nicht relevant, da sie gerade bei kleinen Gesichtern zu ungenau sind.

### 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 2.3

Die von MTCNN gelieferten und vergrößerten Boxen werden nun auf mindestens  $130 \times 180$  Pixel gebracht, sollte sie kleiner sein. Neben der einfachen Skalierung, muss die Überführung von Bildkoordinaten des Bildausschnittes in die Koordinaten im Frame bekannt sein, damit dies bei späteren Berechnungen berücksichtigt werden können.

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 dann der Rechenaufwand pro Gesicht erhöht und die Zuverlässigkeit der Berechnungen von OpenFace wieder sinkt, z.B. durch Falschdetektion des Gesichtes.

### 3.4 Bestimmung der Landmarks

Für die Bestimmung der Landmarks wird OpenFace eingesetzt. Dabei wird jeder Bildausschnitt unabhängig der anderen Betrachtet und da bekannt ist, um welche Person es sich im Bild handelt, kann direkt mit dem jeweiligen CNN gearbeitet werden, das auf diese Person optimiert wurde.

Durch die vorige Selektion wird nur auf jenen Bildausschnitten gerechnet auf denen auch die Person zu sehen ist, wodurch nicht unnötig gesucht werden muss und auch ein Lernen auf Personen stattfinden kann die nur selten zu sehen sind, da sie nur resettet werden, wenn sie eigentlich zu sehen sein müssten aber nicht detektiert wurden.

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 die Bereiche, auf denen Ergebnisse geliefert werden, deutlich erhöht.

Dies ist interessant für die spätere Anwendung, da somit auch Einzelbilder verwendet werden können, die eine deutlich höhere Auflösung haben als ein Video. Allerdings sinkt dann der maximale Winkel relativ zur Kamera beträchtlich, zu Gunsten der maximalen Distanz. Außerdem können schon kleinsten Farbänderungen im Bild beim Hochskalieren ausschlaggebend sein, ob ein Gesicht erkannt werden kann, wodurch bei gleicher Bildqualität Gesichter im Video besser erkannt werden.

Da die gesamte Berechnung auf Grau-Bildern basiert ist auch eine Farbkorrektur, wie Verbesserung des Kontrast, Farbverlauf usw. möglich, um etwaige Einflüsse bei der Aufnahme zu korrigieren.

Dennoch kann es passieren, dass trotz allem ein Gesicht falsch detektiert wird, wie z.B. das erkennen eines Gesichtes in der Ohrmuschel, diese müssen entsprechend behandelt werden, da ansonsten das Lernen auf diese Bereiche stattfindet und im nächsten Frame erneut nach diesen Merkmalen gesucht wird.

- Verbesserung durch Farbkorrektur

## 3.5 Verbesserung der Augen

Zusätzlich zu den 64 Landmarks, die ein Gesicht beschreiben, kann von OpenFace weitere 28 Landmarks für ein Auge bestimmt werden, aus denen dann die Blickrichtung ermittelt wird.

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.

Da unter den 28 Landmarks die Umrisse von Pupille und Iris beschrieben wird, müssen diese aus dem Ergebnis von ElSe abgeleitet werden. Dabei hat sich eine Veränderung des Radius mit ?? für Pupille und ?? für die Iris bewährt.

Allerdings muss das Auge für die Berechnung aus entsprechend vielen Pixeln bestehen, wodurch es im Originalbild mindestens mit 10 Pixeln dargestellt wird, um sinnvolle Ergebnisse zu erhalten. Da diese Berechnung unabhängig der Landmarks ausgeführt wird, empfiehlt sich das Ergebnis zu überprüfen, damit die bestimmte Ellipse auch innerhalb der Augenhöhle liegt.

Dabei wird jedes Auge unabhängig vom anderen betrachtet, wodurch sich verschiedene Blickrichtung ergeben. Ab einer Distanz von mehr als ??cm kann die Blickrichtung beider Augen als parallel angesehen und kann entsprechend behandelt werden. Eine Verbesserung ergibt sich, wenn beide Augen anhängig von einander bestimmt werden, damit sich der Fehler minimiert.

### 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, da die Exakte Position der Landmarks bekannt sind. Da auch in der späteren Anwendung der Augenbereich genauer bestimmt ist, bevor ElSe zum Einsatz kommt wurde, nur der Bildbereich in dem alle Landmarks der Augenlider liegen, somit sind die Bilder etwa 64 auf 29 Pixel groß. Um die Qualität der Berechnung bei verschiedenen Größen zu simulieren, wurde das Bild um den angegebenen Faktor linear verkleinert.

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.

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

Als Bestes Verfahren stellte sich Geam heraus, siehe Abbildung 3.1, da die Abweichung vom Zentrum minimal ist. Der Unterschied zwischen den einzelnen Verfahren ist signifikant. Ein Mittleres Ergebnis liefert das Luminance-Verfahren, Unterabschnitt 2.4.1, mit welchem eine Abweichung auf dem Augentrainingsdatensatz [WBZ<sup>+</sup>15] von 6.42 Pixel erreicht wird.

Im Vergleich liefert das Quadratische-Verfahren Unterabschnitt 2.4.4 im Test die schlechtesten Ergebnisse, da die durchschnittliche Abweichung bei 7.23 Pixel liegt. Diese Ungenauigkeit lässt sich wahrscheinlich auf den fehlenden Farbunterschied zwischen Iris und Pupille zurück zu führen ist.

Das beste Ergebnis wurde mit Geam erreicht, eine mittlere Abweichung von 5.89 Pixel. Durch die Gamma-Korrektur ist zwar die Kontur der Iris deutlich schlechter zu detektieren, dies wird allerdings durch die Genauigkeit bei weiten wett gemacht.

Da die Abweichung von ElSe konstant bei etwa 5.9 Pixel bei Glean liegt, arbeitet ElSe stabil bei den Skalierungen, siehe Abbildung 3.4, womit es vor allem bei kleinen Bildern zuverlässig Ergebnisse liefern kann als OpenFace.

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,

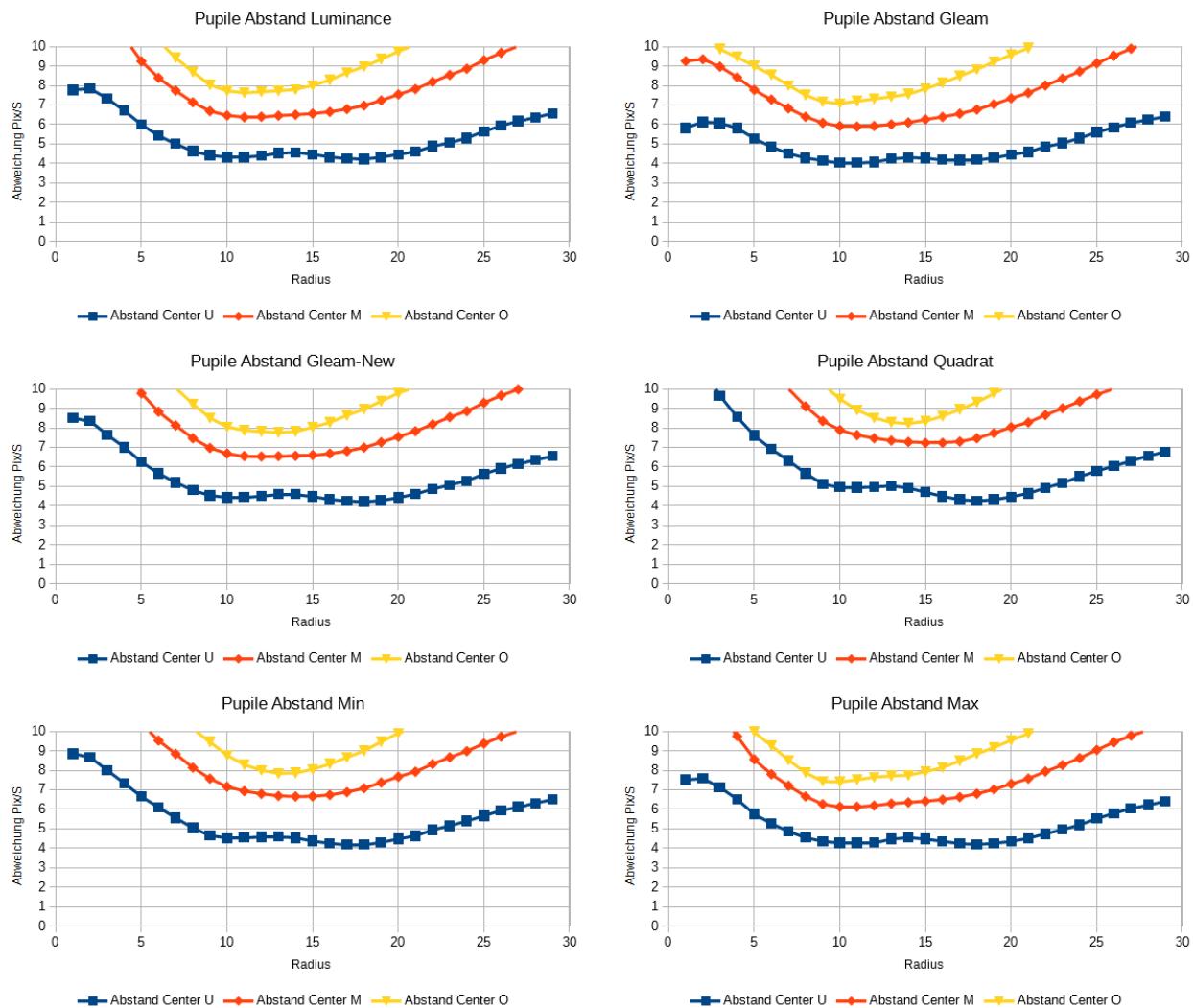


Abbildung 3.1: Abstand des Zentrums der Landmark-Pupille und der Berechneten Ellipse in [Pixel/Skalierung] Oben-Links: Luminance, Oben-Rechts: Gleam, Mitte-Links: Gleam New, Mitte-Rechts: Quadrat, Unten-Links: Min-Wert, Unten-Rechts: Max-Wert

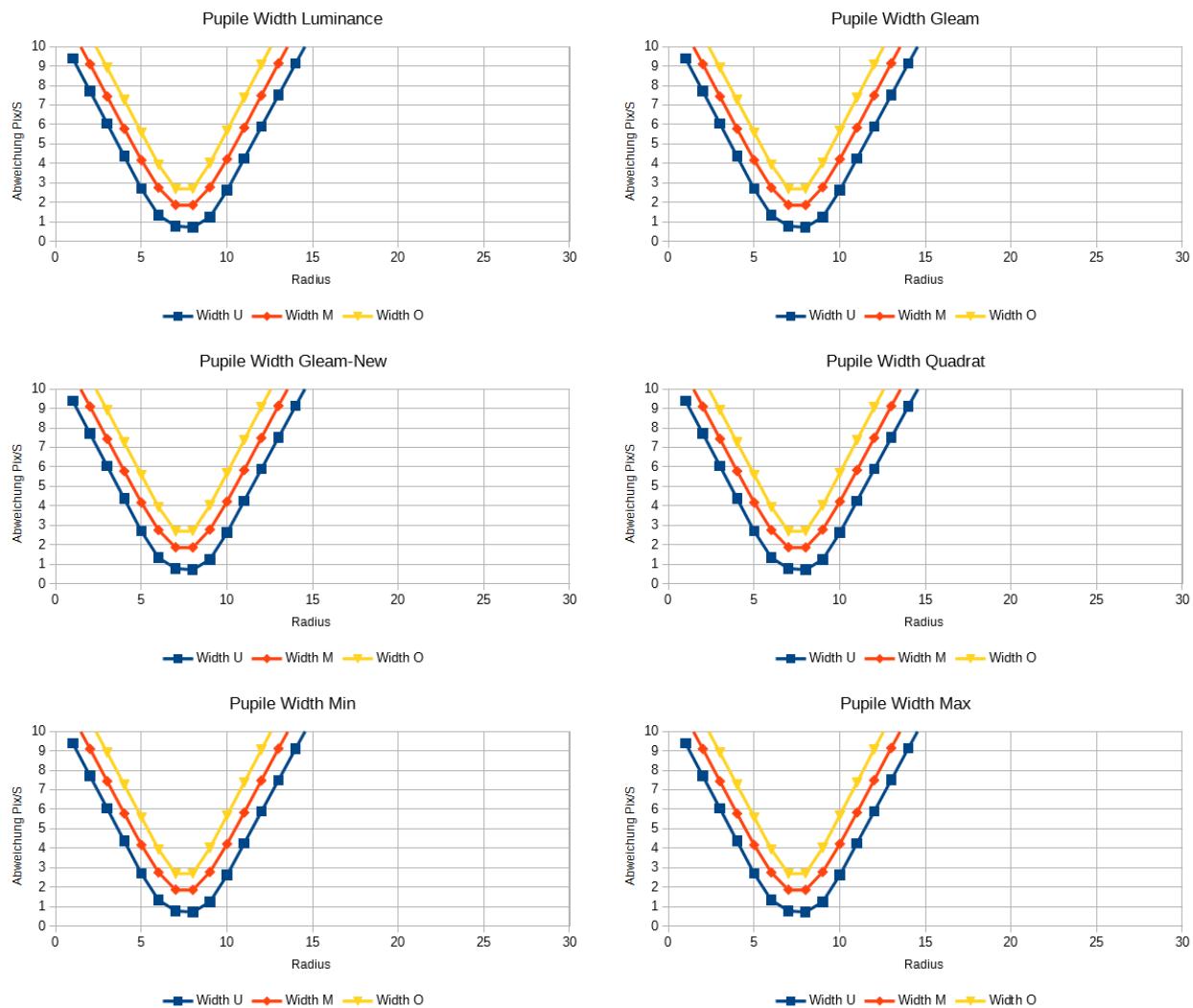


Abbildung 3.2: Unterschied Zwischen den Radien der Landmark-Pupille und der Berechneten Ellipse in [Pixel/Skalierung] Oben-Links: Luminance, Oben-Rechts: Gleam, Mitte-Links: Gleam New, Mitte-Rechts: Quadrat, Unten-Links: Min-Wert, Unten-Rechts: Max-Wert

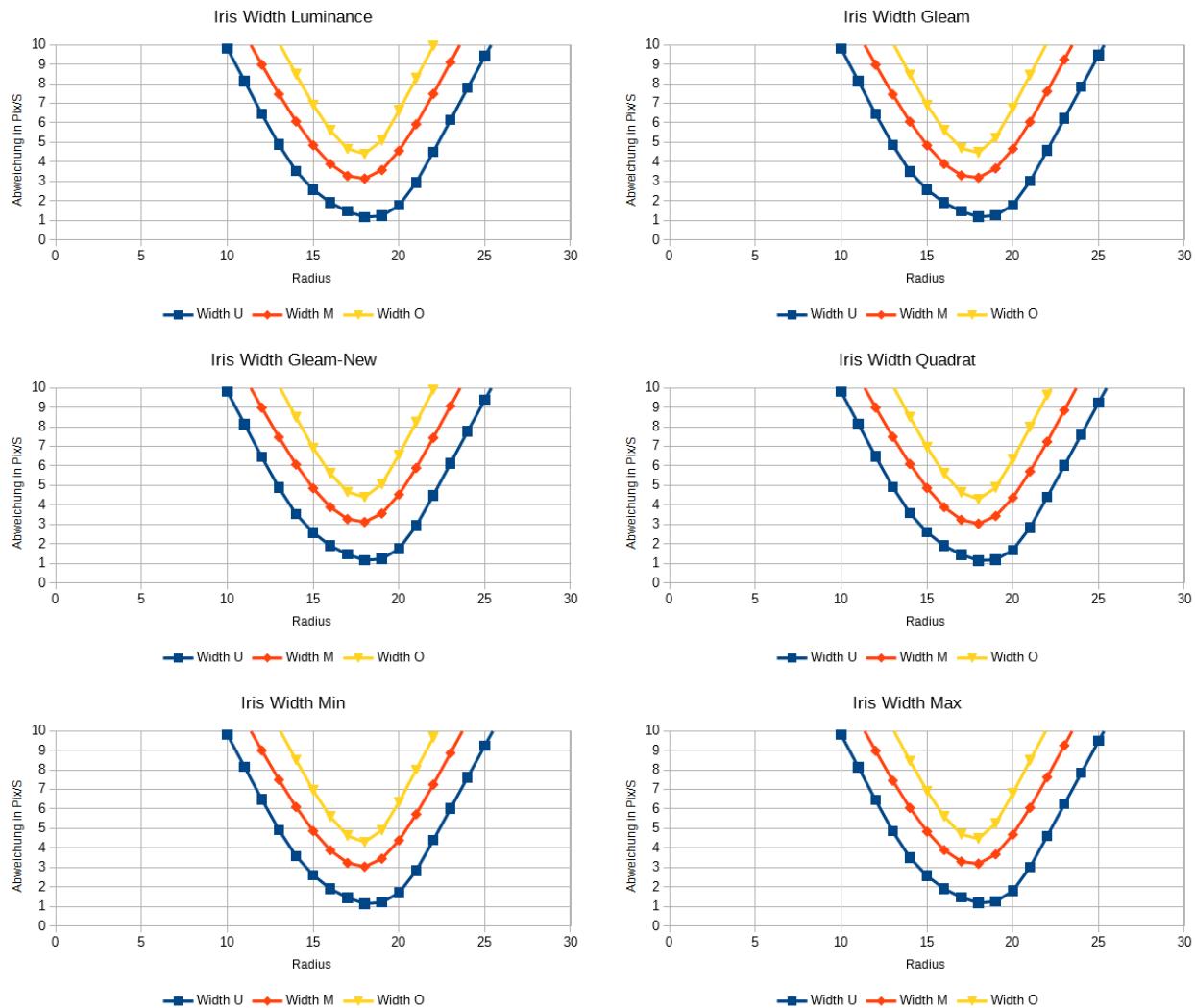


Abbildung 3.3: Unterschied Zwischen den Radien der Landmark-Iris und der Berechneten Ellipse in [Pixel/Skalierung] gegen die Radius-Größe. Oben-Links: Luminance, Oben-Rechts: Gleam, Mitte-Links: Gleam New, Mitte-Rechts: Quadrat, Unten-Links: Min-Wert, Unten-Rechts: Max-Wert

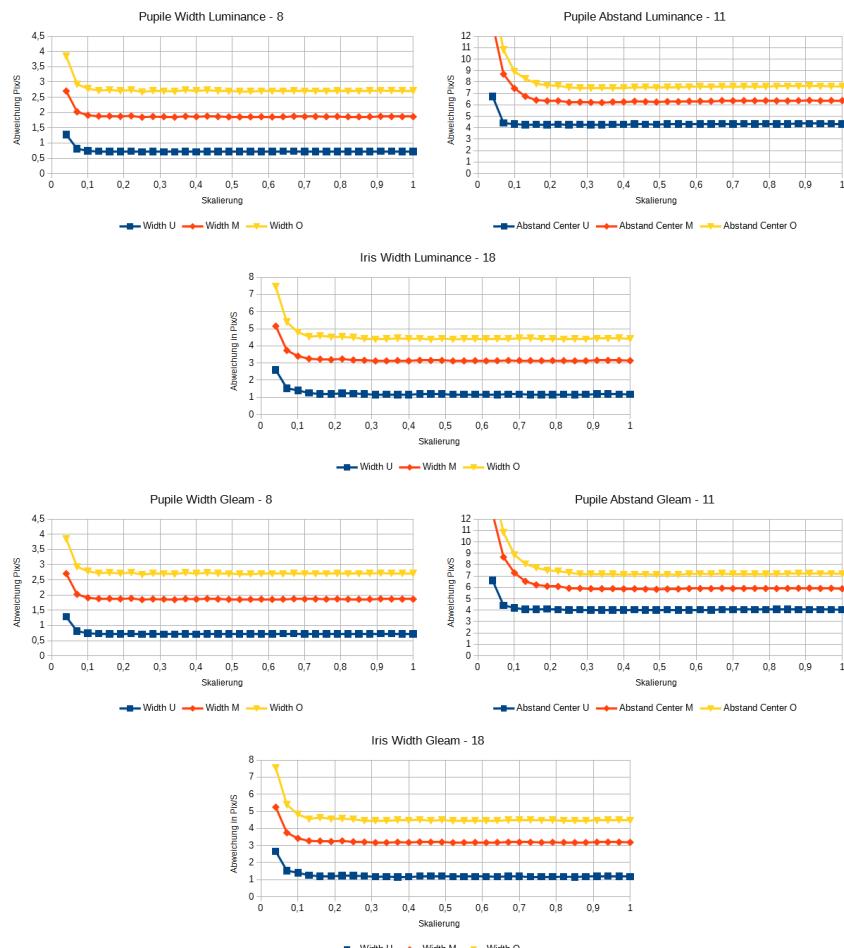


Abbildung 3.4: Auswirkung von der Bildgröße auf die Qualität der Berechnung. Oben: Luminance, Unten Gleam

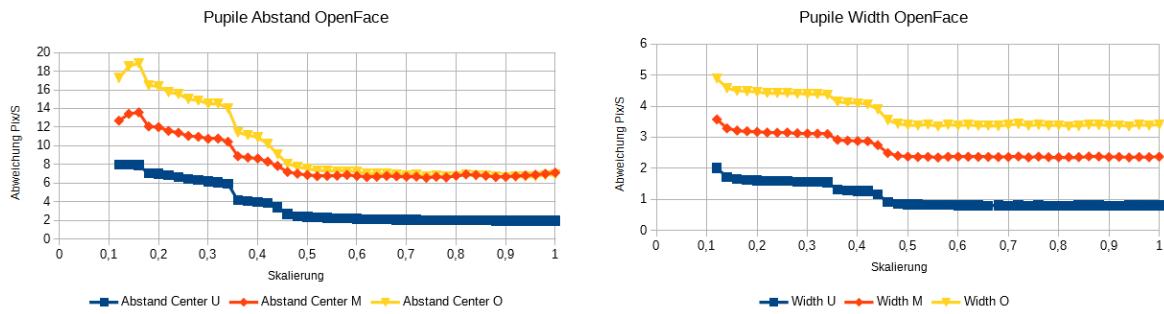


Abbildung 3.5: Auswirkung von Skalierung auf die Qualität der Augendetektion von OpenFace

da der mögliche Fehler von OpenFace geringer ist als von ElSe.

Im Bereich zwischen 21 und 15 Pixel können beide Ergebnisse Kombiniert werden, beide Verfahren arbeiten ungefähr gleich gut. Sollte die Iris im Originalbild noch kleiner sein, so ist auf ElSe mehr verlass, 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 das Auge ausgeschnitten um diesen Bereich auf 384 Pixel als Eingabebild für ElSe zu vergrößern. Im Datensatz besitzen 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 2.5.1, muss ElSe in diesem Aspekt zuverlässige Ergebnisse liefern.

Im Versuch hat sich ein Radius von etwa einem Zwölftel des zu erwartetem Durchmesser der Iris bzw. Pupille als sinnvoll erwiesen, siehe Abbildung 3.1 und Abbildung 3.2 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 11 am besten, siehe Abbildung 3.1, wobei der 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. 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 im Bildausschnitt, wodurch er verwendet werden kann als Eingabe von ElSe.

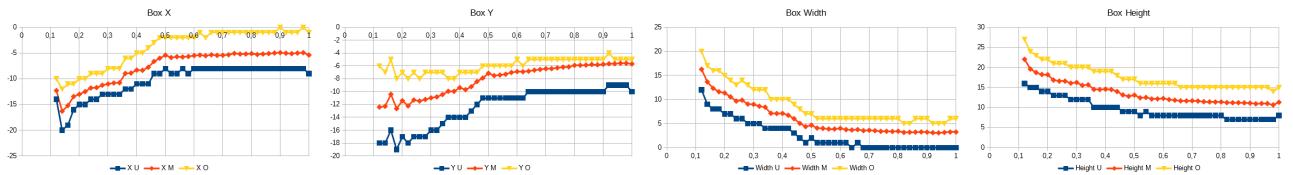


Abbildung 3.6: Bestimmung der Box ums Auge

### 3.6 Bestimmung der Position & Orientierung

Für die Bestimmung der Position und Orientierung des Gesichtes wird wie in Abschnitt 2.7 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.

### 3.7 Bestimmung einer Position auf der die Aufmerksamkeit liegt

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

#### 3.7.1 Schnittpunkt berechnen

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

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

#### 3.7.2 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 zu verrauscht sind um ein sinnvolles Ergebnis mit den Schnittpunkt Unterabschnitt 3.7.1 zu erhalten.

# 4 Ergebnisse

## 4.1 Erreichte Werte

### 4.1.1 Auswirkung der Größe

Durch den Aufbau, muss das Verfahren zuverlässig bezüglich der Größe 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, betrachtet

Zur Durchführung wurden die Größe der Bilder mit dem Faktor multipliziert um so kleinere Gesichter zu erhalten und anschließend mit dem Image-Detector von OpenFace zu detektieren, siehe Abbildung 4.1.

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 verwendeten Kamera Abschnitt 1.4 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 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 abgebildet.

Es ist zu erkennen das durch die Vergrößerung, Gesichter in Bereichen die normal nicht erkennbar sind, bestimmbare werden. 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, einer Detektion auf Gesichtern von 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 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 und dann mit Rauschen versehen um sie anschließend mit den verschiedenen Verfahren zu vergrößern.

Somit soll geprüft werden, welches der Verfahren auch stabil gegen Rauschen ist. Das Rauschen wird

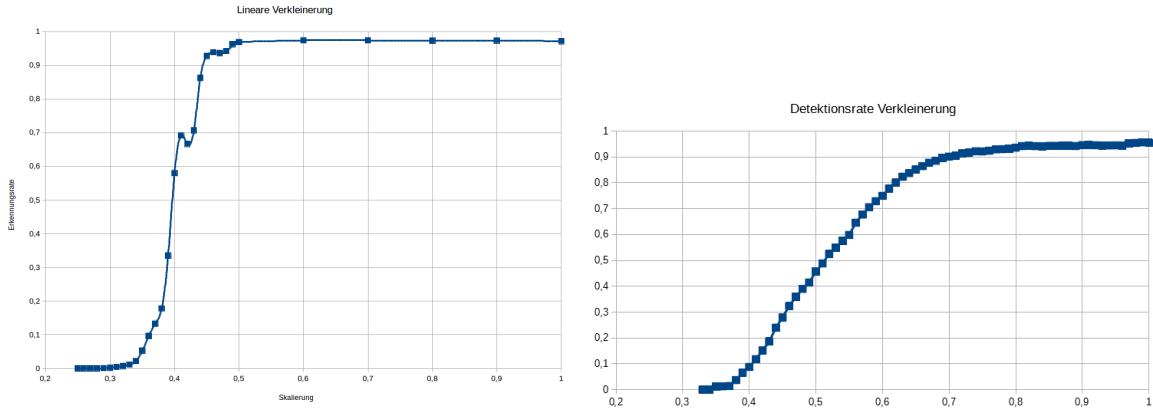


Abbildung 4.1: Die Bilder aus Labeled Faces in the Wild [HMLLM12] (links) und Random Forests [FDG<sup>+</sup>13] wurden mit den Faktor auf der X-Achse linear verkleinert und die Erkennungsrate Y-Achse abgebildet

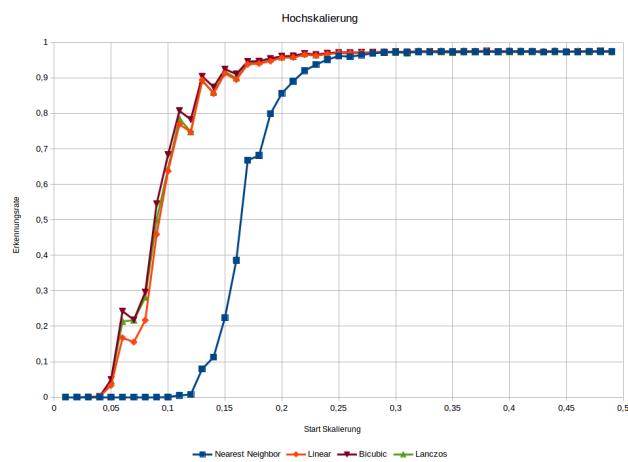


Abbildung 4.2: Die Bilder aus Labeled Faces in the Wild [HMLLM12] wurden mit den Faktor auf der X-Achse linear verkleinert und mit den verschiedenen Verfahren wieder vergrößert Abschnitt 2.3. Aufgetragen gegen die Detektionswahrscheinlichkeit. Nearest-Neighbor (blau), Linear (rot), Bicubic (braun), Lanczos (grün)

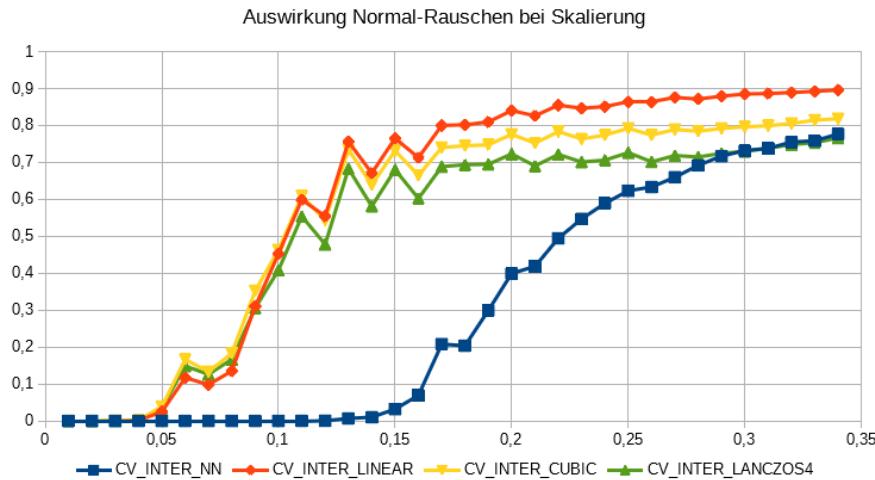


Abbildung 4.3: Bilder aus Labeled Faces in the Wild [HMLLM12], mit dem X-Faktor verkleinert, um jedes Pixel mit 50% Wahrscheinlichkeit auf  $\pm 10\%$  Gleichverteilung der Abweichung

für jedes Pixel mit einer Wahrscheinlichkeit von 50% auf eine gleich verteilte Abweichung von  $\pm 10\%$  des Originalen Farbwertes simuliert. Anschließend wird das verrausche Bild mit den verschiedenen Verfahren vergrößert. Dieser Vorgang wurde für jedes Bild vier mal wiederholt um Zufälligkeiten 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 seine Position und Orientierung bekannt ist. 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 den 3D und 2D Landmarks.

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

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.

Zur Bewertung, die Durchschnittliche Distanz zwischen Kamera und Kopf betrgt ca  $70\text{cm}$  bei einer Kopfbreite von 78 Pixel. Der schnelle Abfall der Genauigkeit ist an der selben Stelle (0.5) an der auch die Detektionsrate stark absinkt.

## 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 (Pos 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 (Posw Worls und Correcht Pose World) deutlich berlegen, siehe Abbildung 4.8. Auch hier liegt der mittlere Fehler ber lange Zeit bei etwa  $9^\circ$

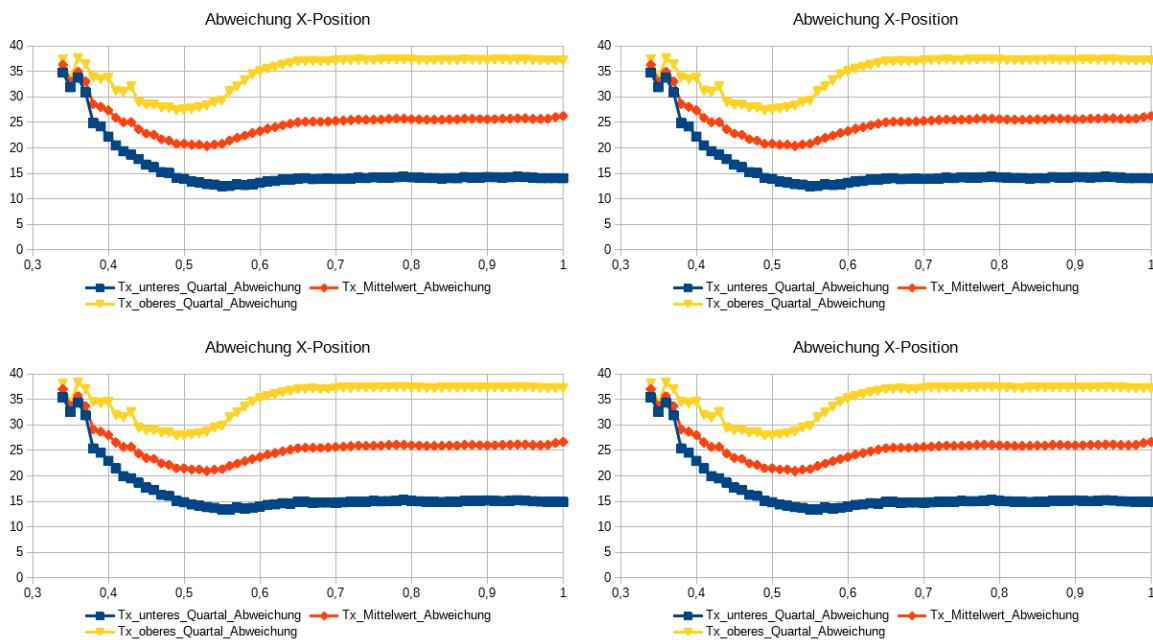


Abbildung 4.5: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist in Millimeter.

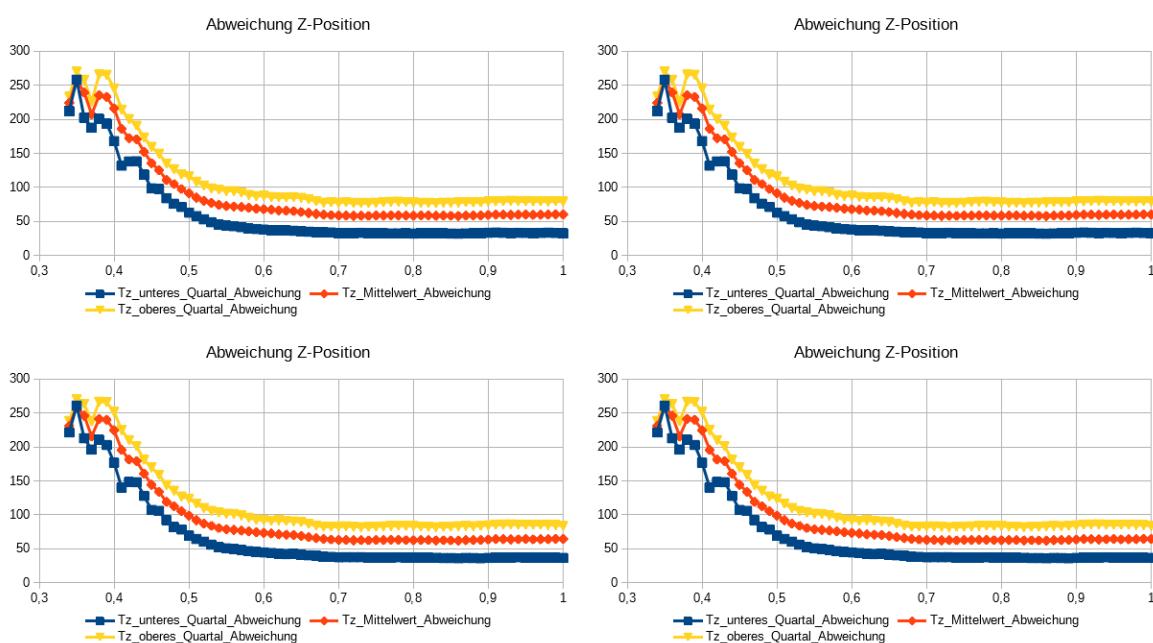


Abbildung 4.6: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist in Millimeter.

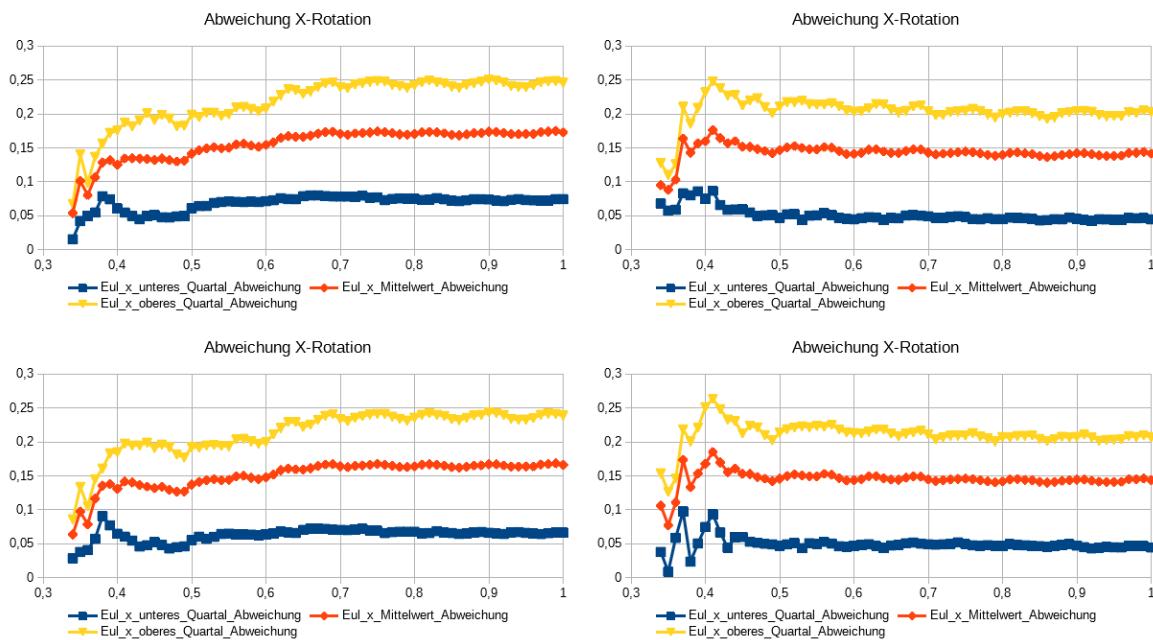


Abbildung 4.7: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist im Bogenmaß.

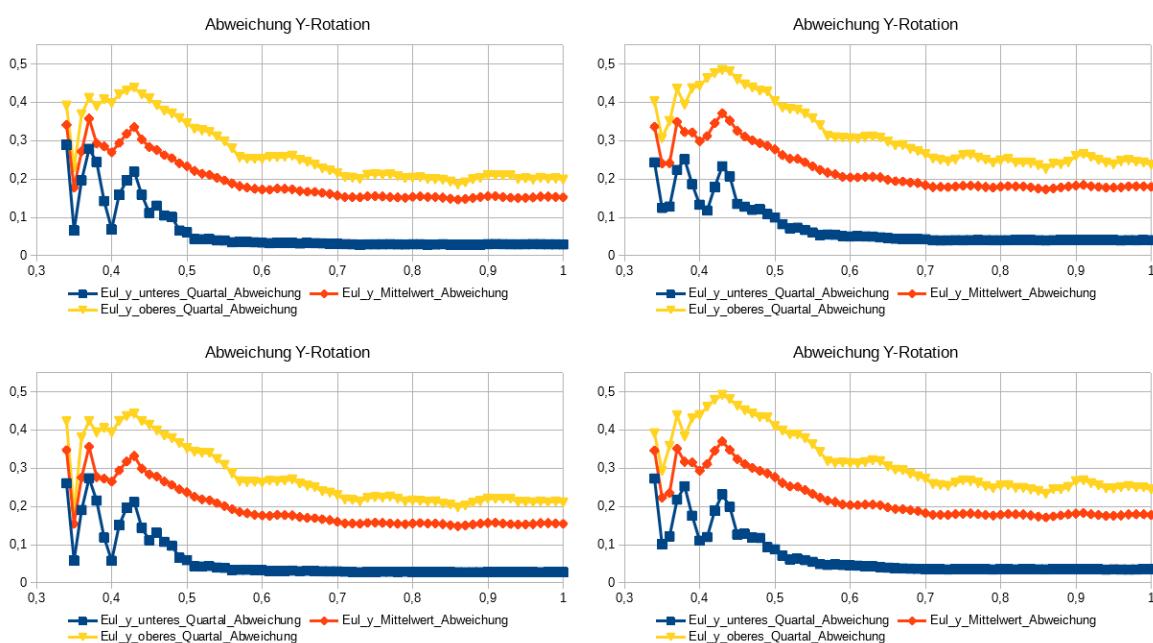


Abbildung 4.8: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist m Bogenmaß.

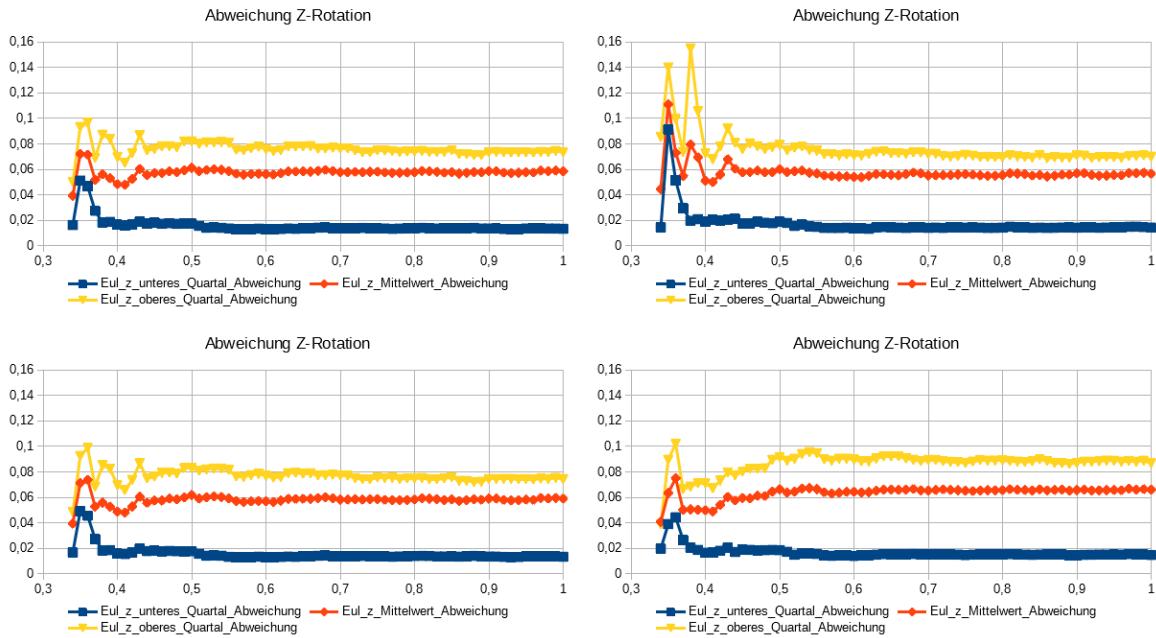


Abbildung 4.9: Pose World (links oben), Pose World (rechts oben), Correct Pose Camera (links unten) und Coorect Pose World, der Abstand (Y-Achse) ist im Bogenmaß.

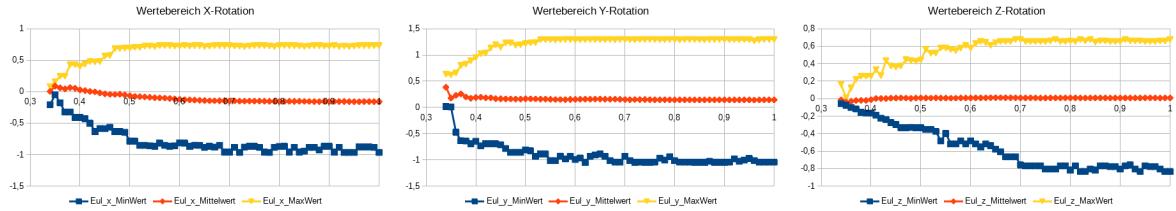


Abbildung 4.10: Darstellung der noch detektierten Wertebereiche in Bogenmaß.

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 sehr schlecht wird.

### 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. Dies ist wichtig zu wissen, da wenn kein gesicht in diesen Bereichen nicht erkannt werden kann auch die späteren Verfahren nicht bestimmt werden können.

Der Wertebereich auf den einzelnen Achsen sollte ist ausreichend sein für die Anwendung, auch wenn die Rotation pralle zur Horizontalen etwas größer sein könnte.

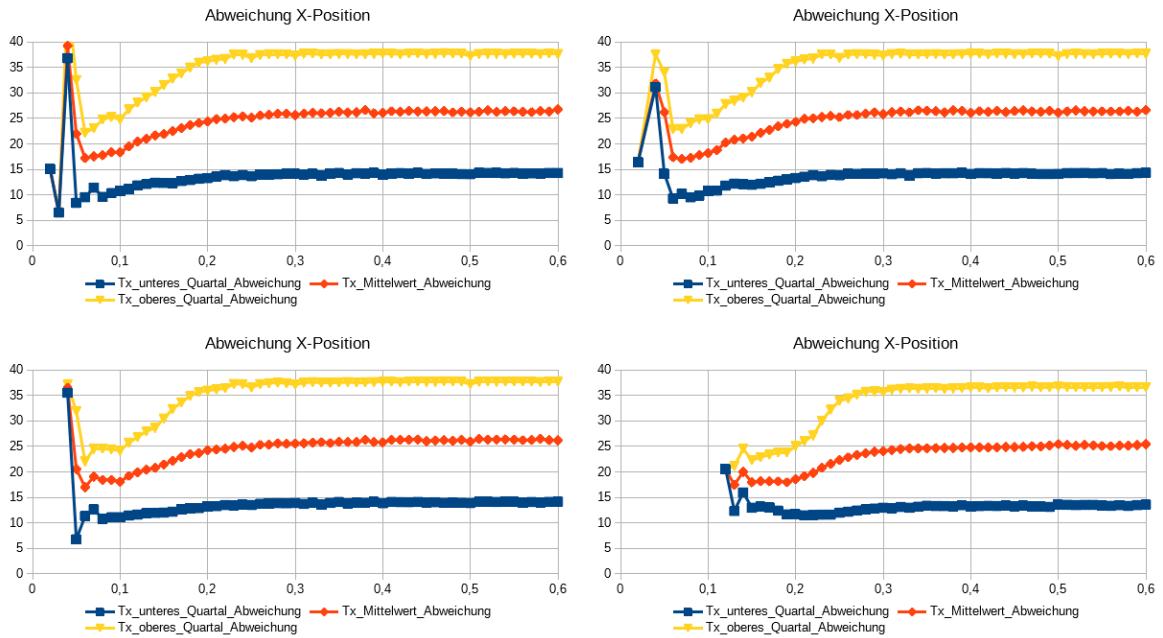


Abbildung 4.11: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung in X-Richtung (Y-Achse) in Millimeter. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

#### 4.1.6 Qualität der Skalierung

Nun wird der Zusammenhang zwischen den verschiedenen Skalierungsverfahren und der Qualität der Ergebnisse gesucht.

Es zeigt sich, dass bei der Bestimmung der Parameter ist das Nearest-Neighbor Verfahren am genauesten, allerdings sit 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 gleichmäßige Ergebnisse, kann aber vom Rechenaufwand abhängig gemacht werden welches Verfahren gewählt werden soll.

#### 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 besten 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 ist, ist auch hier das Nearest-Neighbor genauer, siehe Abbildung 4.12. 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.

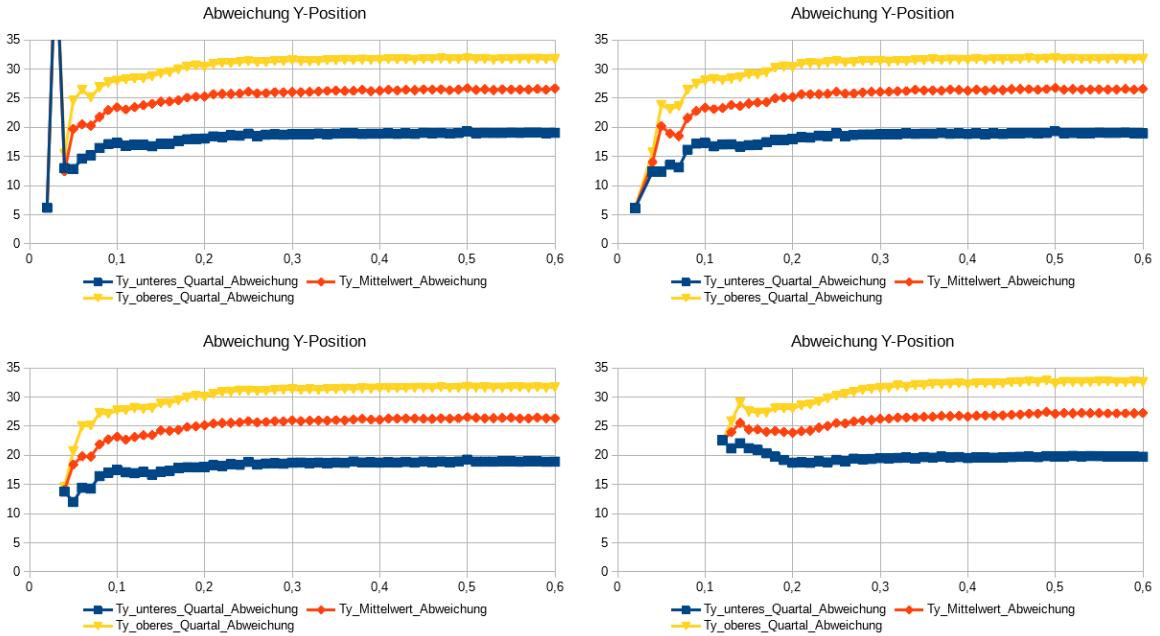


Abbildung 4.12: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung in Y-Richtung (Y-Achse) in Millimeter. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

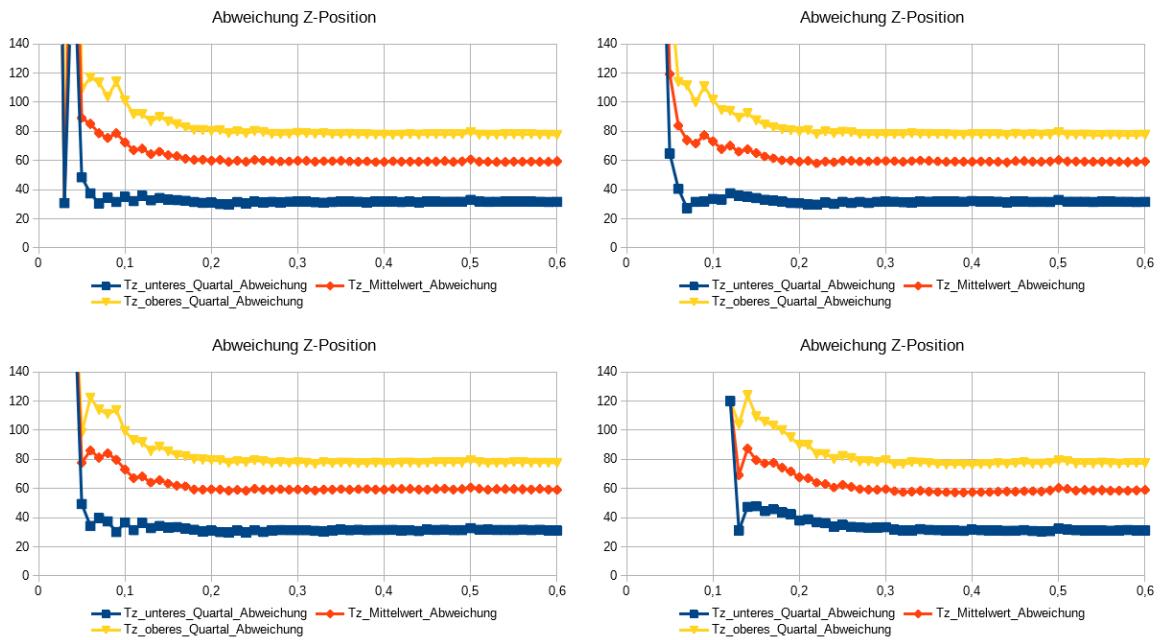


Abbildung 4.13: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung in Z-Richtung (Y-Achse) in Millimeter. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

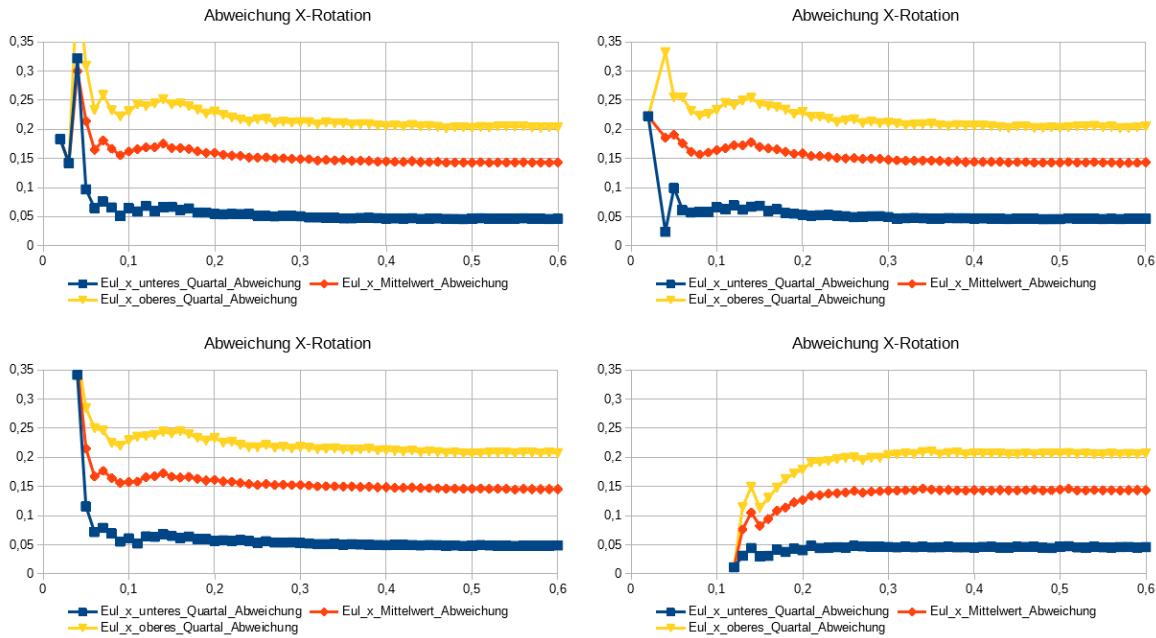


Abbildung 4.14: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in X-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

## Orientierung

Als weitere Bestimmung wird die berechneten Winkel um die jeweilige Achse. Geringste Abweichung bei der bestimung der X-Rotation bei 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.

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
- Auswirkung von Pixelrauschen
  - Rauschen der Actioncam bestimmen  
Done
  - Simulation des Rauschens  
Add Gaußverteilung auf Image
- Angabe des Koordiantansystems
- Auswerten der Messung
- Wann ELSE

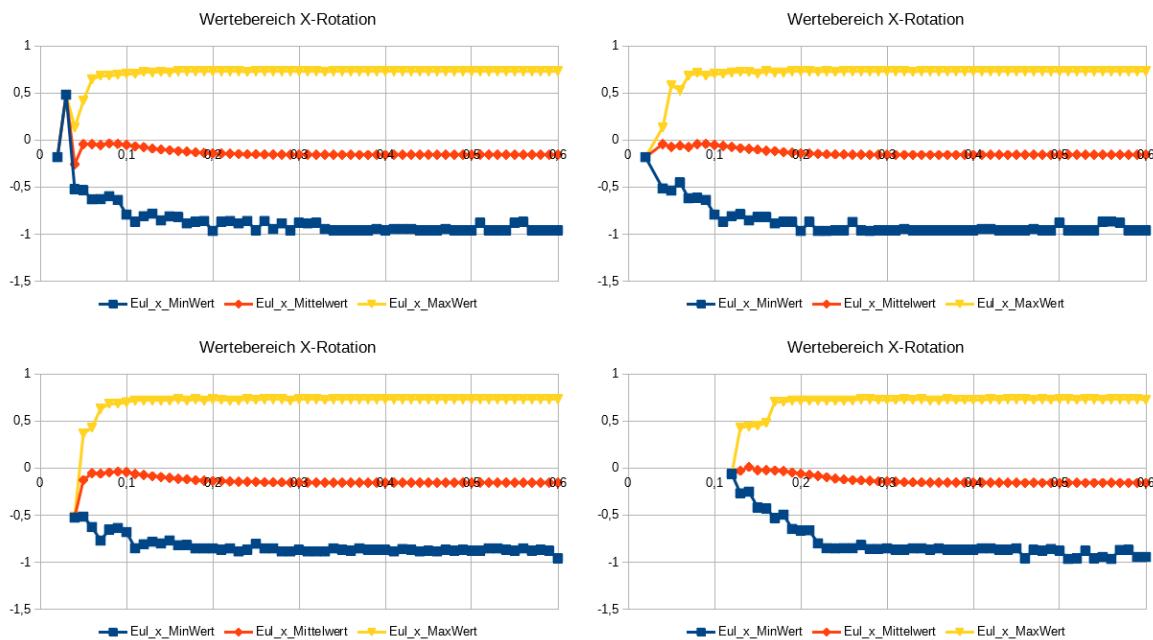


Abbildung 4.15: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in X-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

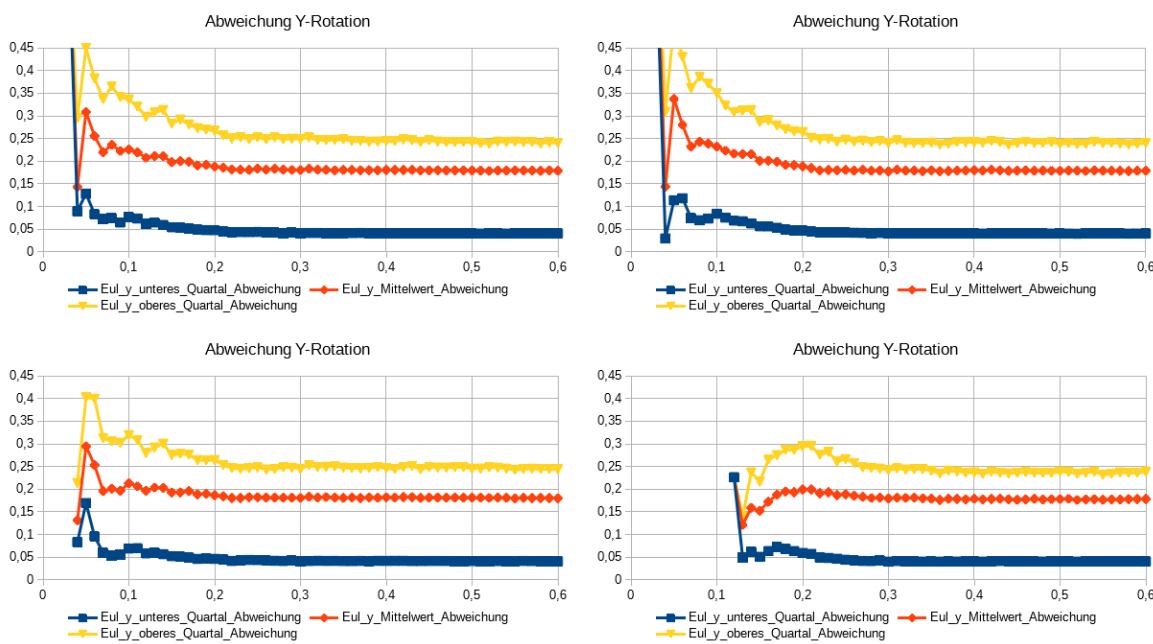


Abbildung 4.16: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in Y-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

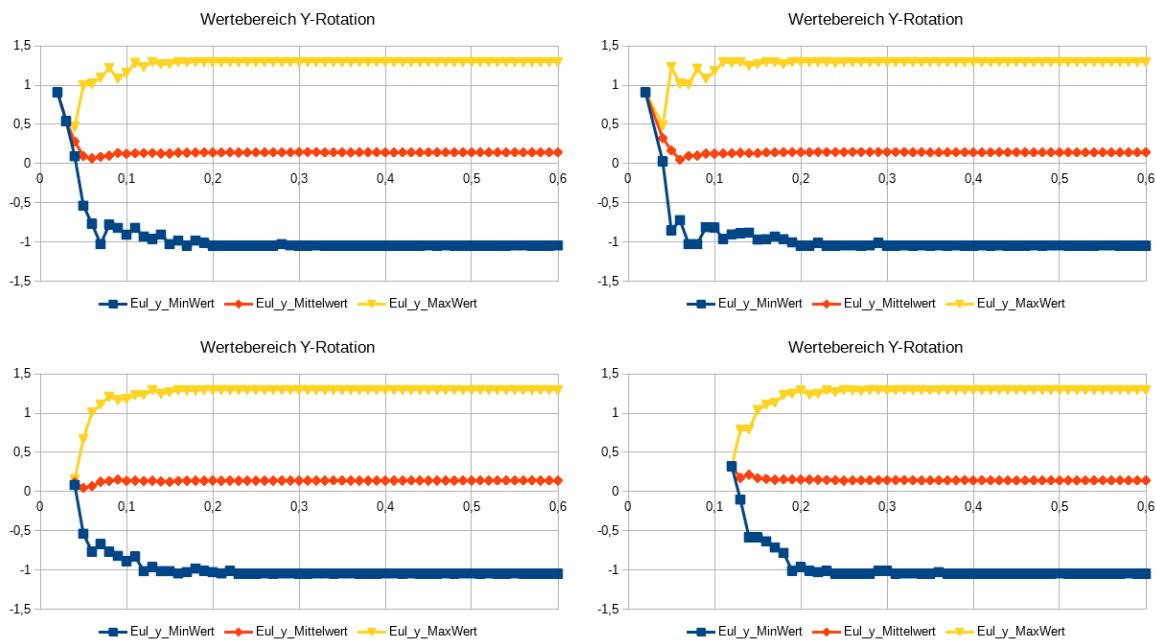


Abbildung 4.17: Zusammenhang zwischen der Skalierung (X-Achse) und der Wertebereich des Winkels in Y-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

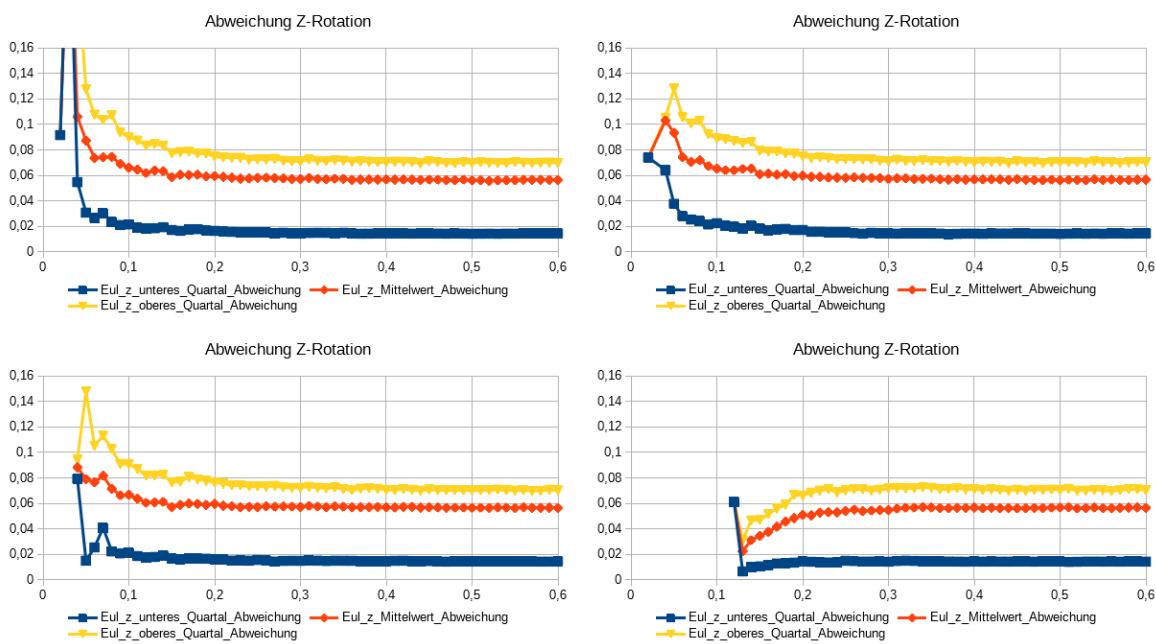


Abbildung 4.18: Zusammenhang zwischen der Skalierung (X-Achse) und der Abweichung des Winkels in Z-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

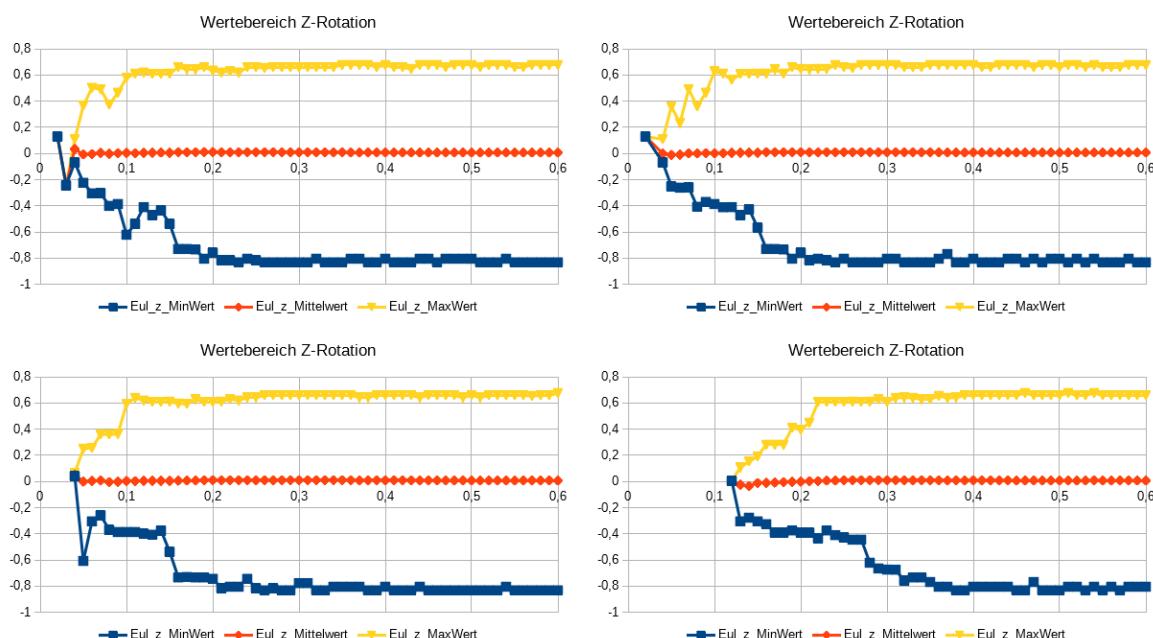


Abbildung 4.19: Zusammenhang zwischen der Skalierung (X-Achse) und der Wertebereiche des Winkels in Z-Richtung, Angabe in Bogenmaß. Bicubic (oben links), Lanczos (oben rechts), Linear (unten links), Nearest-Neighbor (unten rechts)

- Mittlung Ergebnis / Landmarks
- Zuverlässigkeit mit Farbkorrektur

## 4.2 OpenFace auf Video - To Do

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 normal mit dem Blick folgen damit für jeden Zeitpunkt die Blickrichtung bekannt ist.

### 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  $1.49m$ .

Das Ziel das Betrachtet werden soll (Target), beginnt immer in der Mitte und bleibt dort  $1s$  Stehen, bewegt sich innerhalb von 4 Sekunden einen der Randpunkte, dargestellt in Abbildung 4.20, verweilt dort für eine Sekunde und begibt sich in  $4s$  zu einem nächstgelegenen Randpunkt, bleibt dort  $1s$  und geht zurück zum Zentrum, dies wiederholt.

Die Versuchspersonen stellten sich etwa  $1.5m$  von der vor der Leinwand entfernt auf, die Kamera befand sich  $24cm$  unterhalb und  $12.5cm$  vor dem Zentralen Punkt der Targets mit Blickrichtung von den Targets weg.

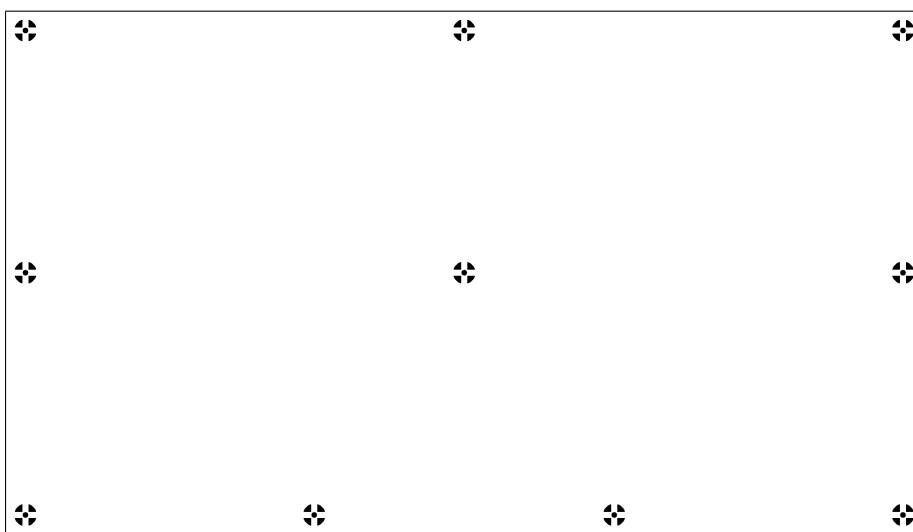


Abbildung 4.20: Eckpositionen des Bewegten Ziels bei der Videoaufnahme

#### 4.2.2 Versuchs - Durchführung

Um die ungefähre Position des Kopfes zu ermitteln, wurde die Distanz zwischen dem Nasenrücken und den 4 Eckpunkten mittels eines Laserdistanzmessers bestimmt um die Position relativ zur Leinwand und Kamera zu ermitteln.

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 2m und 1s auf natürlicher weise.

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

Die Aufnahme wurde mit  $15Fps$  in Farbe mit einer Auflösung von  $1600 \times 896$  Pixel aufgezeichnet. Die Kamera besitzt einen horizontalen Blickwinkel von etwa  $70^\circ$ .

#### 4.2.3 Ergebnis - To Do

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

- Graphik mit den Blickverfolgung
- Plot Winkel gegen Winkel und Abweichung
- 

#### 4.2.4 Fehleranalyse

Eine Betrachtung der Fehlerquellen die Bei der Messung entstanden sind bzw. die durch den Aufbau Entstehen. Außerdem eine weitere bei der Berechnung.

#### Messung

Die erste Ungenauigkeit liegt bei der Distanz zur Leinwand, diese wurde nur zu beginn, vor der eigentlichen Aufnahme bestimmt. Somit ist entsteht eine Abweichung da Kopf in Bewegung ist, auch

währen der Aufnahme.

Die eigentliche Messung der Distanz ist ebenfalls ungenau, da sie eine Abweichung von etwa 1cm in alle Richtungen aufweist. Außerdem liegt der Ursprung der Rechnung etwas Tiefer und weiter Hinten als der Messpunkt. Die Parameter für der Überführungsmatrix von Welt- nach Kamerakoordinaten sowie die Brennweite wurden zwar sorgsam bestimmt, sind aber dennoch nicht perfekt.

Durch den Bedingten Aufbau, 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 immer ein Blickwinkel von unten auf das Gesicht entsteht.

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 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 Augenbewegungen, bis etwa 20° stattfindet und diese nicht erfasst werden können.

Ein Weiterer nicht zu verachtendes Problem ist die Reflexion vor allem auf den Brillen, von den starken Lichtquellen wie Fenster, Projektor- und dessen Bild sowie der Lampen. Auch Schatten gerade bei den Augenhöhlen erschweren die Auswertung.

- Bild für den Versuchsaufbau
- Typ der Kamera

## **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.

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 wegdrehen. 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 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.
- [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.
- [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.
- [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.
- [TB16] Louis-Philippe Morency Tadas Baltrušaitis, Peter Robinson. Openface: an open source facial behavior analysis toolkit, 2016.

- [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].