



UNIVERSITÄT  
KOBLENZ · LANDAU

Fachbereich 4: Informatik



# Untersuchung von Pupil und Glint Center Estimation Algorithmen auf synthetischen Testbildern

Bachelorarbeit  
zur Erlangung des Grades  
BACHELOR OF SCIENCE  
im Studiengang Computervisualistik

vorgelegt von

Lydia Rebecca Weiland

**Betreuer:** Dipl.-Inform. Detlev Droege, Institut für Computervisualistik,  
Fachbereich Informatik, Universität Koblenz-Landau  
**Erstgutachter:** Prof. Dr.-Ing. Dietrich Paulus, Institut für  
Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau  
**Zweitgutachter:** Dipl.-Inform. Detlev Droege, Institut für  
Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau

Koblenz, im Juni 2010



## Kurzfassung

Es gibt einige Gaze Tracking Systeme, sowohl high- als auch low-cost. Low-cost Systeme gehen meist mit low-resolution Kameras einher. Da hier die Bildqualität schlechter ist, müssen die Algorithmen umso besser arbeiten. Aber wie soll man die Algorithmen die der Erkennung der Blickrichtung dienen, testen, wenn die Bildqualität geringer ist und man nie korrekte Aussagen über die Referenzpunkte treffen kann? Hier greift die Idee dieser Arbeit: Mit Hilfe synthetischer Augenbilder testet man die betreffenden Algorithmen und kann diese, da die Referenzpunkte bekannt sind, analysieren. Eine Veränderung der Komplexität dieser Bilder z.B. mit Hilfe eines zuschaltbaren Gaußrauschens oder eines weiteren Reflektionspunktes, macht es möglich, diese in Stufen der Realität anzunähern. Im Idealfall kann man die Algorithmen mit den aus den Testreihen gewonnenen Erkenntnissen verbessern und bei Anwendung innerhalb eines low-resolution Systems dessen Genauigkeit erhöhen.

## Abstract

There are a few systems high and low-cost ones for gaze tracking. Normally low-cost systems go in hand with low-resolution cameras. Here the image quality is poor, so the algorithms for detecting the gaze have to work more precisely. But how to test and analyse them correctly, when there is a bad image quality and no reference point known? The idea of this work is, to generate synthetic eye images, where the reference points are known, because they are mainly manually set and then to test and analyse the algorithms with these synthetic images. By switching on features like gaussian noise or a second glint-like reflection point, it is possible to stepwise approximate the synthetic images close to reality. In fact the experiments will lead to an improvement of the algorithms used in a low-resolution system environment.



## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Vereinbarung der Arbeitsgruppe für Studien- und Abschlussarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. ja  nein

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja  nein

Koblenz, den 7. Juni 2010



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>13</b>
1.1	Motivation . . . . .	13
1.2	Zielsetzung . . . . .	15
1.3	Aufbau der Arbeit . . . . .	15
<b>2</b>	<b>Hintergrund</b>	<b>17</b>
2.1	Das menschliche Auge . . . . .	17
2.1.1	Aufbau des Auges . . . . .	17
2.1.2	Drehachsen des Auges . . . . .	18
2.1.3	Bewegung des Auges . . . . .	18
2.1.4	Das Auge in Zahlen . . . . .	18
2.2	Synthetische Bilder . . . . .	19
2.2.1	Aufbau der Bilder . . . . .	20
2.2.2	Die Schwierigkeitsgrade . . . . .	20
2.3	Algorithmen zur Schätzung des Pupillenmittelpunktes . . . . .	21
2.3.1	Center of Gravity . . . . .	21
2.3.2	Algorithmus nach Pérez, Córdoba, García, Méndez, Muñoz, Pedraza und Sánchez . . . . .	22
2.3.3	Ellipse Fitting . . . . .	24
2.3.4	Gradient Direction Consensus . . . . .	26
2.3.5	Image Based Template Matching . . . . .	26
2.4	Algorithmen zur Schätzung des Reflektionsmittelpunktes . . . . .	27
2.4.1	Center of Gravity . . . . .	27
2.4.2	Algorithmus nach Pérez, Córdoba, García, Méndez, Muñoz, Pedraza und Sánchez . . . . .	27
<b>3</b>	<b>Untersuchung der Pupil Center Estimation Algorithmen</b>	<b>29</b>
3.1	Versuchsreihen . . . . .	29
3.2	1. Versuchsreihe . . . . .	29
3.2.1	Bescheinung . . . . .	29
3.2.2	CoG . . . . .	30

3.2.3	Pérez . . . . .	31
3.2.4	Ellipse Fitting . . . . .	32
3.2.5	Template Matching . . . . .	32
3.3	2. Versuchsreihe . . . . .	33
3.3.1	Beschreibung . . . . .	33
3.3.2	CoG . . . . .	34
3.3.3	Pérez . . . . .	34
3.3.4	Template Matching . . . . .	35
3.4	3. Versuchsreihe . . . . .	36
3.4.1	Beschreibung . . . . .	36
3.4.2	CoG . . . . .	36
3.4.3	Pérez . . . . .	38
3.5	4. Versuchsreihe . . . . .	39
3.5.1	Beschreibung . . . . .	39
3.5.2	CoG . . . . .	39
3.5.3	Pérez . . . . .	40
3.6	5. Versuchsreihe . . . . .	41
3.6.1	Beschreibung . . . . .	41
3.6.2	CoG . . . . .	41
3.6.3	Pérez . . . . .	42
<b>4</b>	<b>Untersuchung eines Glint Center Estimation Algorithmus</b>	<b>45</b>
4.1	Versuchsreihen . . . . .	45
4.2	1. Versuchsreihe . . . . .	45
4.2.1	Bescheibung . . . . .	45
4.2.2	CoG . . . . .	46
4.3	2. Versuchsreihe . . . . .	47
4.3.1	Beschreibung . . . . .	47
4.3.2	CoG . . . . .	47
4.4	3. Versuchsreihe . . . . .	47
4.4.1	Beschreibung . . . . .	47
4.4.2	CoG . . . . .	48
<b>5</b>	<b>Fazit</b>	<b>51</b>
5.1	Fazit . . . . .	51
5.1.1	Probleme . . . . .	53
5.1.2	Ausblick . . . . .	54
<b>6</b>	<b>Anhang A</b>	<b>55</b>
6.1	Hinweise zur Verwendung von <i>AGG</i> unter Windows . . . . .	55

# Tabellenverzeichnis

3.1	CoG Schätzung in 1. Versuchsreihe . . . . .	30
3.2	Pérez Schätzung in 1. Versuchsreihe . . . . .	31
3.3	CoG Schätzung in 2. Versuchsreihe . . . . .	34
3.4	Pérez Schätzung in 2. Versuchsreihe . . . . .	35
3.5	CoG Schätzung in 3. Versuchsreihe . . . . .	37
3.6	Pérez Schätzung in 3. Versuchsreihe . . . . .	38
3.7	CoG Schätzung in 4. Versuchsreihe . . . . .	39
3.8	Pérez Schätzung in 4. Versuchsreihe . . . . .	41
3.9	CoG Schätzung in 5. Versuchsreihe . . . . .	43
3.10	Pérez Schätzung in 5. Versuchsreihe . . . . .	44
4.1	CoG Glint Schätzung in 1. Versuchsreihe . . . . .	47
4.2	CoG Glint Schätzung in 2. Versuchsreihe . . . . .	48
4.3	CoG Glint Schätzung in 3. Versuchsreihe . . . . .	49



# Abbildungsverzeichnis

2.1	Schema des menschlichen Auges [Med07]	17
2.2	Drehachsen des Auges [Sch99]	18
2.3	Schema der <i>AGG</i> rendering pipeline	19
2.4	Mit <i>AGG</i> erzeugtes synthetisches Bild	21
3.1	Merge des original Bildes und des Templates nach dem ersten Durchlauf	33
3.2	Merge des original Bildes und des Templates nach dem letzten Durchlauf. Iris und Pupille sind jeweils gleich groß	33
3.3	Synthetisches Auge mit leichter Verdeckung der Iris und des Limbus	33
3.4	Synthetisches Auge mit Gaußrauschen	36
3.5	<i>Gnuplot</i> Auswertung einer x-Achsen Subpixelverschiebung von 210.1 bis 212.9 und CoG Schätzung	40
3.6	<i>Gnuplot</i> Auswertung einer x-Achsen Subpixelverschiebung von 210.1 bis 212.9 und Perez Schätzung	42
3.7	<i>Gnuplot</i> Auswertung einer y-Achsen Subpixelverschiebung von 300.1 bis 302.9 und CoG Schätzung	43
3.8	<i>Gnuplot</i> Auswertung einer y-Achsen Subpixelverschiebung von 300.1 bis 302.9 und Perez Schätzung	44
4.1	Synthetisches Auge nach Anwendung der Binarisierung des Bereichs um die Pupille mit 1.5-fachen geschätztem Pupillenradius	46
4.2	Synthetisches Auge mit Gauss-Rauschen nach Anwendung eines Mittelwertfilters und der Binarisierung des Bereichs um die Pupille mit 1.5-fachen geschätztem Pupillenradius	48
5.1	Synthetisches Augenbild mit Gradientenverlauf innerhalb der Iris	54
6.1	Inkludieren der GDI32 Bibliothek unter Wascana	56
6.2	Angeben der Pfade zu den .h Dateien von <i>AGG</i> und <i>coreDataStructs</i>	56
6.3	Integration der nötigen .cpp Dateien	56



# Kapitel 1

## Einleitung

*“Art is a tool of empowerment and social change, and I consider myself blessed to be able to create and use my work to promote health reform, bring awareness about ALS and help others.”*

(Tony Quan, aka Tempt One [Hei10a])

### 1.1 Motivation

Gaze Tracking Systeme bieten nicht nur für Menschen mit Behinderungen eine Möglichkeit der Bedienung von beispielsweise Computern und damit einen Weg in die Kommunikation und die digitale Datenwelt, sondern haben durch ihr Potential bereits in unserem Alltag, die Medizin und Forschung, die Industrie, die Psychologie und nicht zu Letzt die HCI Einzug erhalten [Hei10b] [Mac07]. So helfen sie außerdem bei Laseroperationen der Augen, eben dieses durch gezielte Steuerung des Lasers zu schützen oder unterstützen bei der Usability-Forschung [Web10], genauso, wie sie es Psychologen möglich macht, Lernprozesse zu analysieren [Sco07]. Man unterscheidet also generell zwischen aktiven Systemen, also jenen, die eine Interaktion via Eye bzw. Gaze Tracking möglich machen und, passiven Systemen, die die Augenbewegung und Blickrichtung aufzeichnen [LWP05]

Betrachtet man die Hardware, die für das Gaze Tracking von Nöten ist, dann differenziert man auch hier wieder zwei Systeme: Zum einen die head-mounted, mit denen ein höherer Bewegungsgrad für den Kopf erreicht wird, die aber gleichzeitig dadurch, dass sie eben am Kopf fixiert werden müssen, einschränken, und zum anderen die fest aufgestellten sogenannten remote Systeme [LWP05], die zwar nicht am Benutzer angebracht werden müssen und damit das Empfinden eines störenden Fremdkörpers vermeiden, die aber empfindlich auf Bewegungen bspw. des Kopfes

reagieren [KR99].

Zumeist wird für die Aufnahme der Augenbilder eine Infrarot Lichtquelle benutzt, da diese für eine gleichmäßige Beleuchtung des Auges sorgt; es gibt zwar Systeme die mit dem sichtbaren Spektrum arbeiten, allerdings bieten diese jene Gleichmäßigkeit nicht. Man trennt den Systemaufbau, bei dem sich die IR Lichtquelle auf der optischen Achse bzw. in unmittelbarer Nähe von dieser befindet und, der dafür sorgt, dass die Pupille als helle Fläche zu erkennen ist, auch bekannt unter dem Namen *bright pupil*, von jenem, bei dem sich die Lichtquelle weiter entfernt von der optischen Achse befindet und damit den *dark pupil* Effekt hervorruft, die Pupille ist hier also dunkel [Ebi98] [OMY02]. Bei letzterem ist zusätzlich die Cornea Reflektion zu sehen, die für Algorithmen, die mit dem Pupillen-Glint Vektor arbeiten wichtig ist.

Um nun die Augen und ihre Bewegungen aufzuzeichnen, benötigt man eine Kamera. Obwohl in den letzten Jahren die Hardwarepreise enorm gesunken sind, ist eine high-resolution Kamera immer noch im Vergleich zu low-resolution Kameras teuer und macht damit, neben den Kosten für die kommerzielle Software und den hinter der Erkennung liegenden implementierten Algorithmen, einen nicht unwesentlichen Teil der Gesamtkosten für ein Gaze Tracking System aus, was im Fazit dazu führt, dass zuverlässige Gaze Tracking Systeme nicht für jeden zugänglich sind. Ziel ist es nun eine low-cost Lösung zu schaffen, die dennoch nicht an der Genauigkeit spart. Auf Grund der geringeren Auflösung, müssen die Algorithmen umso exakter arbeiten.

Hier greift die Idee dieser Arbeit: Mit Hilfe synthetisch generierter Augenbilder Algorithmen, die zur Erkennung der Blickrichtung dienen, auf ihre Exaktheit zu prüfen. Die synthetischen Bilder können durch verschiedene Parameter, z.B. durch das Hinzufügen von Rauschen, erweitert und damit immer mehr der Realität angenähert werden. Dadurch bieten sie die Möglichkeit, die Algorithmen auf verschiedenen Niveaustufen zu testen. Synthetische Bilder haben gegenüber Bildern von Augen menschlicher Probanden den Vorteil, dass die Ergebnisse der Algorithmen auf Richtigkeit geprüft werden können. Was die Analyse über die Genauigkeit von diesen erleichtert und damit eine Plattform schafft, die Algorithmen noch weiter zu verbessern.

## 1.2 Zielsetzung

Ziel dieser Arbeit ist die Implementierung einer Umgebung, basierend auf der sub-pixel genauen Graphik-Bibliothek *AGG*<sup>1</sup>, für die Erstellung synthetischer Augenbilder. Diese Implementation soll derart gestaltet sein, dass die Augenbilder mit Veränderung einiger weniger Parameter skalierbar sind und, im Ganzen auf so wenig wie möglich festen, stattdessen relativen Werten aufbauen. Neben der Erzeugung der Augenbilder, sollen bereits Störfaktoren, wie z.B. Rauschen zur Verfügung stehen und es muss die Möglichkeit der Erweiterung existieren, so dass eine Grundlage für weitere Untersuchungen und Versuchsreihen garantiert ist. Außerdem besteht die Arbeit bereits aus einer Reihe von Versuchsdurchläufen, mit verschiedenen Schwierigkeitsgraden der synthetischen Bilder, welche die hier vorgestellten Algorithmen, die elementar für das Gaze-Tracking sind, prüfen und analysieren.

## 1.3 Aufbau der Arbeit

Zunächst wird das Auge in seinen für die Erzeugung von synthetischen Bildern wichtigsten Grundzügen erläutert, woraufhin die Beschreibung des Aufbaus der synthetischen Bilder folgt, mit einer anschließenden Einführung in die Idee und den Ablauf einiger pupil und glint center estimation Algorithmen.

Im nächsten Teil erfolgt die Beschreibung der Versuchsreihen, inklusive einer kurzen Analyse der Ergebnisse für die getesteten Algorithmen. Anschließend werden alle durch die Arbeit gesammelten Erkenntnisse in einem Gesamtergebnis zusammengefasst, und ein darauf folgender Ausblick erleichtert das Fortführen der Forschung basierend auf dieser Arbeit oder einiger der hier genannten Teilaspekte.

---

<sup>1</sup>[www.antigrain.com](http://www.antigrain.com)



# Kapitel 2

## Hintergrund

### 2.1 Das menschliche Auge

Wichtig für die Generierung und Erstellung der synthetischen Bilder ist, dass Elemente aus denen ein Auge besteht, ihre Geometrie und den ihnen zugeordneten Rollen bei der Erkennung der Blickrichtung, bekannt sind.

#### 2.1.1 Aufbau des Auges

Zu diesen Elementen gehören: Der weiße sichtbare Teil der Lederhaut oder auch Sclera, die Iris, welche durch den Limbus, also den zumeist dunkleren schmalen Rand von der Sclera abgetrennt ist, die Pupille, und die Hornhaut bzw. Cornea, da auf dieser die bereits erwähnte Cornea Reflektion stattfindet. Nicht zu vergessen die Lider, die bei gewissen Augenbewegungen oder bei manchen Menschen generell Teile der Iris oder sogar der Pupille verdecken.

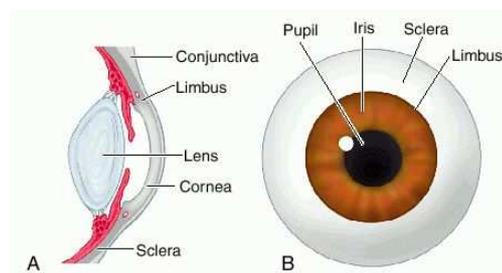


Abbildung 2.1: Schema des menschlichen Auges [Med07]

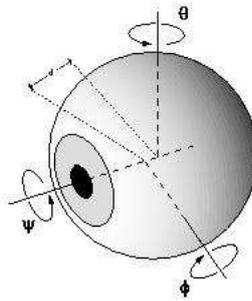


Abbildung 2.2: Drehachsen des Auges [Sch99]

### 2.1.2 Drehachsen des Auges

Das Auge ist entlang von drei Drehachsen beweglich (Abbildung 2.2). Überträgt man diese Achsen auf ein linkshändiges Koordinatensystem, bedeutet das, dass die  $y$ -Achse ( $\Theta$ ) vertikal durch das Auge verläuft und für horizontale Drehungen verantwortlich ist. Studien haben ergeben [Sch99], dass die Drehachse für vertikale Drehungen um einen Abstand  $d$  nach vorne in Richtung Pupille verschoben ist, dies ist übertragen auf das Koordinatensystem die  $x$ -Achse ( $\Phi$ ). Die verbliebene  $z$ -Achse ( $\Psi$ ) verläuft durch den Mittelpunkt der Pupille.

### 2.1.3 Bewegung des Auges

Man unterscheidet verschiedene Arten der Bewegungen die ein Auge vollführen kann. Für das Finden der Blickrichtung, betrachten wir aber hauptsächlich nur zwei: Die Fixation, also die Fokussierung eines Punktes, die allerdings im Gegensatz dazu, was der Name vermuten lässt, unter minimalen Zitterbewegungen stattfindet und die Sakkade, die die Sprünge zwischen den Fixaktionen bezeichnet [Klo09].

### 2.1.4 Das Auge in Zahlen

Da die Genauigkeit der Pupil und Glint Center Estimation Algorithmen mit dieser Arbeit überprüft werden soll, sollte man sich ein paar Kennzahlen die das Auge betreffen ansehen, weil nur auf diese Art sichergestellt ist, dass man Abweichungen um z.B. 1 Pixel in die richtige Relation setzt. Alle Angaben sind ca. Werte und gemessen am durchschnittlichen Erwachsenen [Sch00].

Durchmesser der Iris: 12mm

Durchmesser des Augapfels: 22mm

Durchmesser der Linse: 6,5 -9mm

## 2.2 Synthetische Bilder

Die synthetischen Bilder basieren auf der Graphik-Bibliothek *AGG*<sup>1</sup>, welche unter der GNU GPL Lizenz verwendbar ist. Diese Bibliothek bietet nicht nur den Vorteil, dass sie in C++ implementiert wurde und damit ein leichtes Anknüpfen an vorherige Gaze Tracker Projekte ermöglicht. Diese Bibliothek unterstützt ebenfalls das Anti-Aliasing und arbeitet mit einer Genauigkeit auf Subpixel-Ebene. Für die Fortführung der Arbeit an einem low-cost Gaze Tracker System ist außerdem interessant, dass einzelne Elemente der library leicht durch eigene Implementierungen und damit für individuelle Optimierungen zu ersetzen sind. Ebenso stellt die Bibliothek die Möglichkeit die Rendering Pipeline den eigenen Bedürfnissen anzupassen zur Verfügung (Abbildung 2.3).

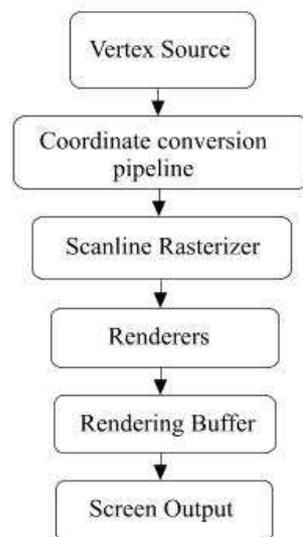


Abbildung 2.3: Schema der *AGG* rendering pipeline

Die vom System generierten synthetischen Bilder werden alle im gray8 Farbraum erzeugt, aber auch hier bietet die *AGG* die Möglichkeit der Variation und Substitution durch bspw. RGBA8. Neben der Anti-Grain Geometry (*AGG*) wurde auch die *coreDataStructs* Library (die Arbeitsgruppe Aktives Sehen besitzt eine Reihe von Software Komponenten die im Gesamten *agasForge* genannt werden, hiervon bilden die *coreDataStructs* einen Teil) verwendet. Dies dient dazu eine geeignete Basis für die in den folgenden Algorithmen zur Berechnung benötigten Vektoren zur Verfügung zu haben und, um die für die Augengeometrien wichtigen Koordi-

---

<sup>1</sup>[www.antigrain.com](http://www.antigrain.com)

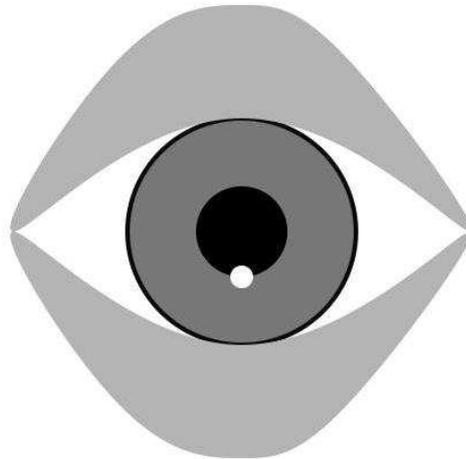
naten in Vektoren abzulegen. Die linke untere Ecke ist mit den Koordinaten(0.0, 0.0) definiert. Es gibt einen Row-Pointer, der immer auf das erste Pixel der betreffenden Reihe zeigt. Dieses Verfahren ermöglicht das Arbeiten mit einem kleinen Teilbereich des Buffers und macht das Transportieren zu anderen Implementierungen einfacher.

### 2.2.1 Aufbau der Bilder

Die synthetischen Bilder bestehen im Wesentlichen aus einem Auge, welches aus einer Iris, einer Pupille, einem Reflektionspunkt, dem Limbus und Ober- und Unterlid aufgebaut ist(Abbildung 2.4). Es wird angenommen, dass Pupille, Iris und Reflektionspunkt eine Kreisform haben. Um einen Kreis unter AGG zu erzeugen, müssen Radien in x- und y-Richtung angegeben werden, es besteht also die Möglichkeit die Formen weniger optimal zu gestalten. Das Auge beginnt standardmäßig auf der Hälfte der Bildhöhe mit 10 Pixeln Abstand zur y-Achse. Um die Größe zu verändern, reicht es aus, die Bildgröße zu verändern und den Iris- und Pupillenradius anzugeben. Wobei man letzteres auch noch in die für den Endanwender verdeckte Berechnung mit einbauen könnte, indem man eine prozentuale Relation von Iris und Pupille definiert. Durch direktes Ansprechen der Pupillenposition, kann diese variiert und damit eine einfache Augenbewegung imitiert werden. Sowohl bei horizontalen als auch bei vertikalen Bewegungen bleibt der Pupillenmittelpunkt immer im Mittelpunkt der Iris verankert. Der Reflektionspunkt bewegt sich bei diesen Pupillenverschiebungen gemäß der zu Grunde liegenden Beleuchtungsposition und der in [GE06] beschriebenen augenbetreffenden Mathematik, um bei der künstlich erzeugten Umgebung trotzdem eine möglichst realitätsnahe Testumgebung zu schaffen.

### 2.2.2 Die Schwierigkeitsgrade

Zunächst wird ein einfaches Auge ohne Verdecken der Iris oder der Pupille durch die Lider generiert. Als nächste Stufe empfiehlt es sich, die Position der Pupille durch entsprechendes Verändern der x- und y-Koordinaten zu translieren, die Irisposition wird ebenso wie das Setzen der Glintkoordinaten angeglichen. Gleichzeitig wird bei Veränderung der y-Koordinate ein erstes Verdecken der Iris erreicht. Durch hinzuschaltbares Gaußrauschen, wird der nächste Schwierigkeitsgrad erreicht. Außerdem ist es möglich neben dem Glint beliebig viele, etwas dunklere(auch hier ist die Farbe frei wählbar) Reflektionen einzublenden, die beispielsweise eine Spiegelung durch den Bildschirm oder ein sich in der Nähe befindliches Fenster imitieren. Neben diesen veränderbaren Parametern, kann man die Lidpositionen direkt ansprechen und damit ein halb geschlossenes Auge oder eines das sehr weit nach



**Abbildung 2.4:** Mit *AGG* erzeugtes synthetisches Bild

unten oder oben ausgerichtet ist reproduzieren. Darüber hinaus kann man die Einfarbigkeit der Augengeometrien durch Gradienten ersetzen und nähert sich damit noch mehr der Realität, sprich einer mehrfarbigen Iris und einem weniger scharfkantig verlaufendem Glint an.

## 2.3 Algorithmen zur Schätzung des Pupillenmittelpunktes

Es werden im Folgenden einige Algorithmen zur Schätzung des Pupillenmittelpunktes, im Englischen *pupil center estimation algorithms*, vorgestellt, deren Exaktheit mit Hilfe einiger Serien der synthetischen Bilder untersucht worden sind.

### 2.3.1 Center of Gravity

Der Center of Gravity Algorithmus oder kurz CoG, zu deutsch Algorithmus zur Bestimmung des Schwerpunktes, bestimmt den Schwerpunkt, also den Mittelwert der sich aus den Positionen der zu einem Objekt gehörigen Massenpunkte ergibt. Da der Schwerpunkt eines Kreises genau in seinem Zentrum liegt und wir davon ausgehen, dass die Pupille kreisrund oder kreisähnlich ist, findet man mit diesem Algorithmus nicht nur den Schwerpunkt sondern gleichzeitig auch das Zentrum der Pupille.

Für die Anwendung dieses Algorithmus auf einer Sequenz eines real existierenden Gesichtes oder Auges bedeutet dies, dass der Bildausschnitt zunächst auf das Auge reduziert wird. Bei Verwendung der synthetischen Bilder ist keine solche Ausschnittsreduktion nötig, da nur Augen und keine umliegenden Gesichtspartien synthetisch generiert werden. Nach Festlegung des zu betrachtenden Bereichs, wird ein Schwellwertverfahren angewendet, um die Pupille vom restlichen Teil des Auges differenzieren zu können. Das Problem besteht darin, ein geeignetes Verfahren für die Definition des Schwellwertes zu finden. Einfach zu implementierende Verfahren wie das nach Otsu [Ots79] eignen sich nicht, da hier mit dem Argumentum Maximum des Quotienten aus den Varianzen zwischen den beiden Klassen und innerhalb der Klassen gearbeitet wird, dies hätte für ein Bild mit einer geringen Auflösung zur Folge, dass zu viele Pixel, die nicht zur Pupille gehören, mit zur Klasse der Pupille gezählt würden, da diese zu nah an dem durchschnittlichen Wert, den eine Pupille in einem solchen Bild hat, liegen.

**Kurzfassung:**

- Ausschnitt auf das Auge reduzieren
- Binarisierung des Ausschnittes
- Schwerpunkt bestimmen

**2.3.2 Algorithmus nach Pérez, Córdoba, García, Méndez, Muñoz, Pedraza und Sánchez**

[PCG<sup>+</sup>03]Im Folgenden mit Pérez benannt. Die Idee dieses Algorithmus besteht in der Schätzung der Blickrichtung mit Hilfe des Pupillen-Glint Vektors.

Zunächst werden die Positionen des Pupillen- und Glintmittelpunktes geschätzt. Den ungefähren Pupillenmittelpunkt erhält man, indem auf den vorher präsentierten Center of Gravity Algorithmus zurückgegriffen wird; es wird also ein Binärbild, durch einen Schwellwert, der nah an dem Wert für Schwarz liegt, erstellt, um dann von den im Binärbild 0-wertigen Pixeln den Massenschwerpunkt zu berechnen. Um zu vermeiden, dass durch Rauschen oder sonstige Bildstörungen ein falsches Gravitationszentrum berechnet wird, wird der Prozess der Findung des CoG mit Pixeln im näheren Umfeld des vorher bestimmten temporären Zentrums, solange wiederholt, bis sich das gefundene Zentrum stabilisiert hat.

Um die Rechenzeit für den nächsten Schritt des Algorithmus zu reduzieren, wird der Bildausschnitt, auf dem im Weiteren gearbeitet wird verkleinert. Dieser

### 2.3. ALGORITHMEN ZUR SCHÄTZUNG DES PUPILLENMITTELPUNKTES<sup>23</sup>

Ausschnitt sollte etwas größer als die Iris sein. Auf diesem Ausschnitt wird nun ein Filter angewendet, damit eine Rauschreduktion erzielt wird. Hierfür eignet sich bspw. ein Mittelwertfilter, da dieser auf homogene Graubereiche keinerlei Auswirkungen hat, aber die Streuung verkleinert und der Mittelwert beibehalten bleibt. Ein Nachteil ist allerdings, dass Kanten geglättet werden. Eine Alternative an dieser Stelle bietet eine Anwendung eines Medianfilters. Nachdem ein Schwellwert die zur Pupille gehörigen Pixel schwarz und den Rest weiß einfärbt, wird ein Laplace-Filter für die Kantendetektion verwendet. Auch hier ist eine Variation mit einem anderen Kantendetektor (bspw. Sobel) möglich.

Nun kommt der Teil, der die Neuerung des Algorithmus nach Pérez et al. ausmacht: In der Annahme, dass die Kontur der Pupille eine Kreisform besitzt, werden ausgehend vom vorher gefundenem Pupillenzentrum eine Anzahl von  $n$ -Strahlen in Richtung Pupillenkantur ausgesendet. Wenn alle diese Strahlen bzw. ihre Kreuzungspunkte mit den von ihnen gefundenen Pupillenkantur einen gleichen Abstand zum Pupillenzentrum haben, wird das vorher definierte Pupillenzentrum verifiziert und die These, dass die Pupille eine Kreisform besitzt gestützt. Kreuzungspunkte, die zu sehr vom mittleren Radius abweichen, bspw. weil statt Pupillenkantur eine Kante des Glints gefunden wurde, werden ignoriert. Sollte dieser optimale Fall nicht eintreten, werden aus den nun zu selektierenden - zu kurze und zu lange Diagonalen werden nicht in die Berechnung mit einbezogen - Strahlen der mittlere Radius berechnet und durch den Schnittpunkt der Diagonalen der neue Pupillenzentrum festgelegt oder als Option, die Schritte die zur Findung der Strahlen dienen so lange wiederholt, bis eine Übereinstimmung der zu den Diagonalen gehörenden Mittelpunkte gefunden werden. Die Wiederholung kann man beliebig oft durchführen, man sollte allerdings bedenken, dass nicht bei jedem Bild die Annahme, dass die Pupille ein Kreis ist, bestätigt wird, es folglich also keine optimale Lösung geben kann und man zusätzlich eine Höchstgrenze an Iterationen einfügen sollte.

Die Iris besitzt ebenfalls wie die Pupille eine Kreisform und hat den gleichen Mittelpunkt. Theoretisch könnte man nun für die Iris den gleichen Mittelpunkt übernehmen, wie für die Pupille im vorherigen Abschnitt gefunden wurde, damit man aber ein möglichst genaues Ergebnis erzielt, verfährt man bei der Festlegung des Irismittelpunktes ähnlich, wie vorher bei der Findung des Pupillenzentrum: Ein Schwellwert fasst den Bereich der Iris, der Pupille und durch zumeist vorhandene Überschneidungen des Lides und der Iris zwangsläufig auch Wimpernbögen zu einem Bereich zusammen. Wieder wird Laplace, also ein Kantendetektor, benutzt und eine erneute Binarisierung stellt die gefundenen Kanten als weiße Linien auf schwarzem Grund dar. Die zuletzt gefundenen Werte des Pupillenzentrum-

telpunktes werden nun temporär als Irismittelpunkt übernommen. Von dort aus werden ebenfalls Strahlen ausgesendet, deren Schnittpunkte mit den vorher gefundenen Kanten, auf gleiche Art und Weise der Findung von Radius und Mittelpunkt der Iris dienen, und bei denen ebenso vorher eine Auswahl von Diagonalen, die zu weit vom Mittelwert abweichen, durchzuführen ist.

**Kurzfassung:**

- Pupillenzentrum mit CoG schätzen
- Bild auf Iris-Ausschnitt verkleinern
- Filtern zur Rauschreduzierung
- Binarisierung mit geeignetem Schwellwert für Pupillenfindung
- Kantendetektion mit z.B. Laplace
- Strahlen vom CoG Pupillenzentrum aussenden
- Ungeeignete Strahlen aussortieren
- Iterativ verfahren, bis Mittelpunkt für alle Strahlen gleich weit entfernt
- Mit Iris ab Schritt der Binarisierung ähnlich verfahren

**2.3.3 Ellipse Fitting**

Dieser Algorithmus hat Ähnlichkeiten mit dem vorhergehenden Algorithmus nach Pérez et al. [PCG<sup>+</sup>03] und den Algorithmen von Li [LWP05] und Daunys und Ramanauskas [DR04]. Es wird ebenfalls mit einem vorher geschätzten Pupillenmittelpunkt die Pupillenkantur durch Aussendung von Strahlen gefunden. Um nun die Kanten zu finden, muss man ebenfalls die Helligkeitswerte der Pixel betrachten, die von den Linien geschnitten werden. Entweder benutzt man dafür den Algorithmus nach Bresenham [Bre65], welcher schnell, aber ungenau rechnet oder man arbeitet mit Unterabtastung von einzelnen Koordinaten [DP10]. Anstatt nun direkt alle Punkte zu betrachten wird ein sogenannter RANSAC Algorithmus [FB81] auf die vorher gefundenen Punkte angewandt.

Ein Random Sample Consensus, kurz RANSAC, Algorithmus wird benutzt, um ein Modell innerhalb einer gegebenen Menge von Punkten zu schätzen und damit ein möglichst genaues Ergebnis, das die Waage zwischen Erkennung von Ausreißern und Exaktheit hält, bietet. Das einfachste Modell, bei welchem ein RANSAC

Algorithmus verwendet werden kann, ist das Suchen einer optimalen Linie in einer Menge von Punkten. Damit eine Linie aufgespannt bzw. definiert werden kann, benötigt man mindestens 2 Punkte. Eine Linie ist für unser Vorhaben, sprich das Ellipse Fitting, nicht angemessen, wir benötigen eine Fläche als Modell, um noch genauer zu sein: eine Ellipse. Die geometrisch einfachste Ellipse ist der Kreis, um diesen von Punkten aufspannen zu lassen, benötigt man mindestens 3 Punkte. Ein RANSAC Algorithmus verfährt iterativ, folglich werden aus der gefundenen Punktmenge 3 beliebige Punkte ausgewählt, mit der Hypothese, dass diese zum Modell und nicht zu den Ausreißern gehören. Mit Hilfe der Koordinaten der Punkte, des Kreiswinkelsatzes, der besagt, dass der Mittelpunktswinkel doppelt so groß ist, wie einer der Umfangswinkel, und der trigonometrischen Funktion, um genauer zu sein der Sinusfunktion des Mittelpunktswinkels, ist es möglich, den Radius und die Koordinaten des Mittelpunktes des durch die drei Punkte aufgespannten Kreises zu definieren. Nun werden die anderen Punkte mit Hilfe eines Schwellwertes, der Ausreißer und zum Modell gehörige Punkte definiert, überprüft. Sollten hierbei für ein Consensus set ausreichend Punkte mit dem Modell übereinstimmen, liegt ein für die Punktmenge optimales Modell vor, andernfalls, werden die Schritte ab der Wahl der drei beliebigen Punkte iterativ mit anderen wiederholt, solange bis ein consensus set vorliegt.

Da die gefundene Punktmenge die Helligkeitswerte und damit die Kante der Pupille repräsentiert, ist der Radius des Modells des definierten consensus sets gleichzeitig der geschätzte Radius der Pupille und gleiches gilt für den Mittelpunkt.

RANSAC bietet also eine andere Möglichkeit Ausreißer zu detektieren und dessen Erfolg ist nicht abhängig von der vollkommenen Geometrie eines Kreises, da das Modell nicht mit den exakten Schnittpunkten der Strahlen rechnet, sondern das Modell eine Annäherung darstellt, das eben auch Punkte akzeptiert die nicht exakt auf dem Kreis liegen sondern innerhalb des Schwellwertbereichs.

#### **Kurzfassung:**

- Pupillenzentrum mit CoG schätzen
- Strahlen vom CoG Pupillenzentrum aussenden
- Bresenham oder bspw. Bi-Lineare Interpolation
- RANSAC

### 2.3.4 Gradient Direction Consensus

[DP10]Dieser Algorithmus macht sich die Eigenschaften der Gradienten zu Nutze, dass diese in die Richtung zeigen, in der die stärkste Änderung stattfindet und deren Länge Aufschluss darüber geben, wie stark diese Änderung ist.

In Bezug auf die Bilder von Augen bedeutet dies, dass die Gradienten, abgesehen von denen die vom Glint oder durch Rauschen erzeugt werden, in Richtung Pupillenmittelpunkt zeigen. Diese Gradienten erhält man durch eine Bildfilterung mit Sobel. Um von vornherein Ausreißer außer Acht zu lassen, werden zwei Schwellwerte benutzt, die zum einen zu geringe und zum anderen zu starke Gradienten aussortieren. Die anderen Gradienten werden benutzt, um mit Hilfe eines M-Schätzers oder M-estimators, den Mittelpunkt zu finden, der den geringsten Abstand zu allen zu betrachtenden Gradienten hat. Der Algorithmus wird in dieser Arbeit nicht getestet.

#### **Kurzfassung:**

- Gradienten mit Hilfe von Sobel berechnen
- Intervall mit zwei Schwellwerten definieren
- M-Estimator anwenden

### 2.3.5 Image Based Template Matching

[DP10]Arbeitet mit synthetisch generierten Bildern von Augen, die als Templates benutzt werden. Um abzugleichen, wie nah die Templates an das Original herantreten, benutzt man entweder die mittlere quadratische Abweichung, auch Mean Squared Error (MSE), oder die Kreuzkorrelation(COR). Je nach Ergebnis, werden die Templates iterativ angeglichen, angefangen bei der Farbe, über die Größe der Radien, bis hin zu den Positionen der Augengeometrien.

#### **Kurzfassung:**

- Pupillenmittelpunkt und -radius bspw. mit Pérez oder Starbust schätzen
- Template erstellen
- Original und Template mit MSE oder COR vergleichen
- Template iterativ anpassen

## 2.4 Algorithmen zur Schätzung des Reflektionsmittelpunktes

Nachfolgend werden einige Algorithmen zur Erkennung des Reflektionsmittelpunktes, im Englischen *glint center estimation algorithms*, präsentiert, von denen das CoG Verfahren innerhalb der Versuchsreihen untersucht wurde.

### 2.4.1 Center of Gravity

Das CoG Verfahren für die Erkennung des Glintmittelpunktes ist ähnlich dem vorangegangenen CoG zur Detektion des Pupillenmittelpunktes. Als Initialwert nutzt man den Pupillenmittelpunkt, um von dort aus den zu untersuchenden Bereich auf die Iris und Pupille zu reduzieren. Dieser Schritt ist notwendig, da sonst Werte der Sclera durch ihre naturgegebene Helligkeit mit in die Berechnung für den Massenschwerpunkt einbezogen würden. Im Folgenden wird dieser Bildausschnitt mit Hilfe eines geeigneten Schwellwertes binarisiert, damit der Massenschwerpunkt des nun weißen Bereichs festgelegt und damit als Glintmittelpunkt definiert werden kann.

### 2.4.2 Algorithmus nach Pérez, Córdoba, García, Méndez, Muñoz, Pedraza und Sánchez

Auch in Pérez et al. wird ein Verfahren zur Bestimmung des Glints bzw. Glintmittelpunktes beschrieben, da der gesamte Algorithmus zur Blickerkennung auf dem Pupillen-Glint Vektor aufbaut und somit sowohl der Pupillenmittelpunkt als auch der bzw die Mittelpunkte der Glints gefunden werden müssen.

Der Algorithmus baut auf der Blob Analyse, die ihren Namen in Anlehnung an das Binary Large Objekt aus der Datenbanktechnik erhalten hat, auf. Hierbei wird nach Bereichen gesucht, die den gleichen logischen Zustand haben, also bspw. in einem Binärbild den Wert 255 besitzen. Anschließend kann man aus diesen Bereichsdefinitionen weitere Informationen, wie z.B. die Größe oder den Schwerpunkt dieser Flächen, gewinnen.

Pérez et al. binarisiert das Bild und wendet darauf die Blob Analyse an, die zunächst den Ursprung eines Blobs extrahiert und danach die 8 Nachbarpixel untersucht, wobei jedes Pixel dem so genannten Pixel labeling Prozess(eine Methode zur Klassifizierung) unterzogen wird. Außerdem wird das zentrale Moment und die Grenze eines jeden Blobs bestimmt. Anschließend werden Blobs aussortiert, die entweder zu groß sind oder außerhalb der Iris liegen. In dem Paper werden vier L-förmige Glints erzeugt deren Ecken eine Art quadratischen Rahmen bilden.

Auf Grund dieser Eigenschaft, wird bei den Blobs nach der jeweils äußeren Ecke gesucht, wobei die Indizierung bei dem linken oberen Blob beginnt und im Uhrzeigersinn fortgesetzt wird. Erhält man also mehr als einen Blob, der die vorhergehenden Bedingungen erfüllt, und damit Glint, kann man für jeden dieser Glints den Pupillen-Glint Vektor bestimmen und bekommt nach angewandeter Kalibrierung redundante Informationen über die Blickrichtung.

# Kapitel 3

## Untersuchung der Pupil Center Estimation Algorithmen

Einige der Algorithmen, die im vorherigen Kapitel zur Detektion des Pupillenmittelpunktes präsentiert wurden, sind auf verschiedenen Variationen und Niveaustufen der synthetischen Bilder, die mit *AGG* erzeugt wurden, getestet worden. Die hierfür nötigen Implementationen der Algorithmen erfolgten kompatibel zum Gaze-Tracker, aber auf Basis der *AGG* Bibliothek. Ergebnisse werden im Folgenden vorgestellt und analysiert. Da davon ausgegangen wird, dass die Augen symmetrisch sind, wurden alle Versuchsreihen nur auf einem Auge durchgeführt.

### 3.1 Versuchsreihen

#### 3.2 1. Versuchsreihe

##### 3.2.1 Bescheibung

Es liegt ein einfaches Augenbild vor, bei dem die Iris und Pupille nicht verdeckt sind. Der initiale Pupillenmittelpunkt befindet sich auf den Koordinaten (210.0, 300.0) und wird im Folgenden um insgesamt 10 Pixel auf der x-Achse auf Position 220.0, in 1 Pixel Sprüngen verschoben. Der Radius der Pupille beträgt 40 Pixel und der der Iris 100. Der Reflektionspunkt befindet sich auf gleicher y-Position, wie der Startmittelpunkt der Pupille und hat eine x-Koordinate von 260.0. Die Strukturen sind einfarbig, die Farbe des Limbus ist, ebenfalls wie die Pupille Schwarz. Es existiert kein Rauschen.

### 3.2.2 CoG

#### Konditionen

Wie im Kapitel 2 bereits erwähnt wurde, eignet sich Otsu als Schwellwertverfahren nicht; dieses findet nämlich einen Schwellwert von 179. Also wurde ein manueller Schwellwert in Höhe von 0 gewählt, damit in der ersten Serie nur Pupillen- und zwangsläufig auch Limbuspixel ausgewählt werden. Letzteres sollte auf die Ergebnisse aber keinerlei negative Beeinflussung haben, da der Limbus den gleichen Massenschwerpunkt besitzt wie die Pupille.

#### Ergebnis und Interpretation

Wie aus Tabelle 3.1 zu entnehmen ist, beträgt die durchschnittliche Abweichung der y-Position 0.54770 und die durchschnittliche Abweichung der x-Koordinate -0.3829. Zu begründen ist dies unter Umständen mit der Wahl des Schwellwertes: Dadurch, dass die AGG Library mit Anti-Aliasing arbeitet, bekommen die äußeren Pixel, die für den menschlichen Betrachter noch zu der Pupille gehören einen höheren Wert als den für schwarz zugewiesen, folglich fallen sie bei der Binarisierung aus dem der Pupille zugehörigen Bereich. Der beste Wert für die x-Position wurde zu dem Zeitpunkt erzielt, je weiter der Reflektionspunkt vom Pupillenmittelpunkt entfernt war. Ebenfalls lässt sich die Verschiebung um fast 0.5 Pixel des y-Wertes entlang der positiven y-Achse mit der Lage des Glints begründen.

Pupillen Zentrum		CoG Schätzung	
x-Position	y-Position	CoG x	CoG y
210	300	209.5	300.61
211	300	210.528	300.61
212	300	211.558	300.61
213	300	212.585	300.585
214	300	213.611	300.569
215	300	214.637	300.552
216	300	215.659	300.528
217	300	216.681	300.503
218	300	217.698	300.471
219	300	218.714	300.439
	Abweichung	-0.38290	0.54770

**Tabelle 3.1:** CoG Schätzung in 1. Versuchsreihe

### 3.2.3 Pérez

#### Konditionen

Statt Laplace wurde ein Sobeloperator zur Detektion der Kanten benutzt. Der manuell ausgewählte Schwellwert wurde auf 20 im Gegensatz zum Schwellwert bei CoG erhöht.

#### Ergebnis und Interpretation

Wenn man auch hier die mittlere Abweichung der Ergebnisse von den tatsächlichen Positionen betrachtet, erhält man gegenüber der Mittelpunktsschätzung allein mit CoG keine Verbesserung (Tabelle 3.2). So weicht die y-Koordinate 0.6143 und die x-Koordinate im Schnitt mit -0.4396 ab. Auch hier fällt wiederum auf, dass die Ergebnisse im einzelnen exakter werden, je weiter sich der Glint vom Pupillenmittelpunkt entfernt, wieder sind die Werte der Schätzung mit einer höheren x-Position näher am Pupillenmittelpunkt. Obwohl die y-Positionen in dieser Versuchsreihe nicht verändert werden, beeinflusst das Verschieben der Pupille auf der x-Achse die Ergebnisse der Schätzung für die y-Position positiv. Eine mögliche Begründung ist, dass der Reflektionspunkt insgesamt weniger Fläche auf der Pupille einnimmt und dadurch weniger Diagonalen beeinflusst.

Pupillen Zentrum		Pérez Schätzung	
x-Position	y-Position	Pérez x	Pérez y
211	300	210.289	300.533
212	300	211.311	300.517
213	300	212.333	300.517
214	300	213.361	300.461
215	300	214.383	300.439
216	300	215.417	300.428
217	300	216.444	300.389
218	300	217.45	300.344
219	300	218.4834	300.328
	Abweichung	-0.4396	0.6143

**Tabelle 3.2:** Pérez Schätzung in 1. Versuchsreihe

### 3.2.4 Ellipse Fitting

Da das Ellipse Fitting Ähnlichkeiten mit dem vorher getesteten Algorithmus nach Pérez besitzt, wird dieser nur in einer späteren Versuchsreihe separat getestet, bzw. wird dieser vom Template Matching Verfahren benutzt.

### 3.2.5 Template Matching

#### Konditionen

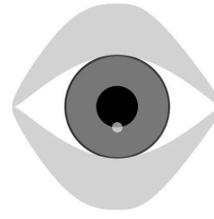
Dadurch, dass generell schon mit einfachen synthetischen Bildern als Originale gearbeitet wird, mussten Templates gefunden werden, die noch einfacher sind, um die Realrelation zwischen Original und Template wenigstens einigermaßen rekonstruieren zu können. Als Template werden also nur zwei Kreise generiert, die zum einen die Iris darstellt und zum anderen die Pupille. Lid und Reflektionspunkt spielen in den Templates keine Rolle. Durch die vorherige Anwendung des Algorithmus nach Pérez oder des Ellipse Fittings, liegt bereits eine sehr gute Schätzung des Pupillenmittelpunktes und des zugehörigen Radius vor. Ein beliebig ausgewählter Faktor der eine Relation zwischen Pupillen- und Irisradius herstellt, bildet den Initialwert für letzteren. Zum Vergleich der beiden Bilder wurde der Mean Squared Error benutzt.

#### Ergebnisse und Interpretation

Zunächst wird ein MSE Wert von 1825.27 gefunden (Abbildung 3.1). Der MSE kann im Gesamten auf 1054.65 reduziert werden (Abbildung 3.2). Da das Template nie exakt wie das Original werden kann, wird dieser Wert niemals gleich null sein. Deshalb muss man sich ein Intervall überlegen, innerhalb welches ein Template akzeptiert und nicht durch weitere Iterationsschritte in Größe oder Farbe verändert wird. Legt man Template (mit Transparenz) und Original übereinander, stellt das menschliche Betrachterauge keinen Unterschied zwischen den Pupillen- und Irisgrößen und deren Positionen fest.



**Abbildung 3.1:** Merge des original Bildes und des Templates nach dem ersten Durchlauf

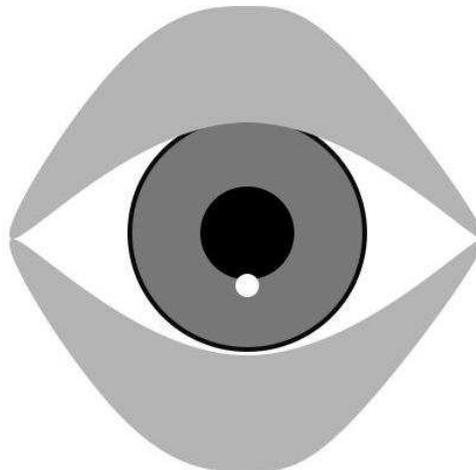


**Abbildung 3.2:** Merge des original Bildes und des Templates nach dem letzten Durchlauf. Iris und Pupille sind jeweils gleich groß

## 3.3 2. Versuchsreihe

### 3.3.1 Beschreibung

Gegenüber der vorherigen Versuchsreihe wurde statt einer Translation in x-Richtung eine Verschiebung in positiver y-Richtung vorgenommen. Als Folge davon, wurde das erste Mal die Iris und damit auch der Limbus zu einem Teil verdeckt (Abbildung 3.3).



**Abbildung 3.3:** Synthetisches Auge mit leichter Verdeckung der Iris und des Limbus

### 3.3.2 CoG

#### Konditionen

Es lagen die gleichen Konditionen wie bei der 1. Versuchsreihe zu Grunde.

#### Ergebnisse und Interpretation

Dadurch, dass der Reflektionspunkt unterhalb des Pupillenmittelpunktes liegt und sich der Pupillenmittelpunkt weiter davon entfernt, hat das eine höhere positive Gewichtung in der Auswirkung der Testergebnisse, als die Tatsache, dass der Limbus, der durch die Wahl des Schwellwertes mit in die CoG Berechnung einfließt, teilweise durch das obere Lid verdeckt wird. Die Abweichung vom tatsächlichen x-Wert bleibt bei allen Einzelbildern konstant bei -0.5, die kleinste Abweichung der y-Position liegt bei 0.036(Tabelle 3.3).

Pupillen Zentrum		CoG Schätzung	
x-Position	y-Position	CoG x	CoG y
210	303	209.5	303.2550
210	304	209.5	304.1030
210	305	209.5	304.9640
	Abweichung	-0.5	0.1313

**Tabelle 3.3:** CoG Schätzung in 2. Versuchsreihe

### 3.3.3 Pérez

#### Konditionen

Es lagen die gleichen Konditionen, wie bei der 1. Versuchsreihe zu Grunde.

#### Ergebnisse und Interpretation

Die durchschnittliche Abweichung der gefundenen y-Position fällt wiederholt geringer aus, als bei der CoG Berechnung. Sie liegt bei 0.1223(Tabelle 3.4) und kann nur mit der besseren Handhabung der Beeinflussung durch den Reflektionspunkt zu begründen sein. Andererseits fällt die Abweichung zu der tatsächlichen x-Koordinate mit -0.72 schlechter aus, als bei CoG.

Pupillen Zentrum		Pérez Schätzung	
x-Position	y-Position	Pérez x	Pérez y
210	303	209.283	303.0440
210	304	209.2610	304.8780
210	305	209.2720	304.6830
	Abweichung	-0.7243	0.1223

**Tabelle 3.4:** Pérez Schätzung in 2. Versuchsreihe

### 3.3.4 Template Matching

Da dem Template Matching die Pupillendaten entweder von Pérez oder dem Ellipse Fitting Algorithmus zu Grunde liegt, und sich nichts an den Bildparametern, die das Ergebnis des Template Matchings verändern würden, geändert hat, wird der Algorithmus in dieser Versuchsreihe nicht getestet.

## 3.4 3. Versuchsreihe

### 3.4.1 Beschreibung

Da die Aufnahmen von Augen in der Realität nicht so perfekt sind, vor allen Dingen, weil wir die Algorithmen in Hinsicht auf low-resolution Aufnahmen testen wollen, wird in dieser Versuchsreihe ein Gaußrauschen mit  $\mu = 0$  und  $\sigma = 20$  aufaddiert (Abbildung 3.4). Die Pupille bewegt sich, wie in der 1. Versuchsreihe 10 Pixel zu je 1 Pixel Schritten in positiver x-Richtung.

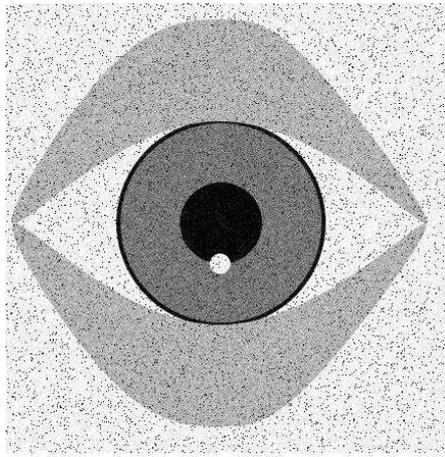


Abbildung 3.4: Synthetisches Auge mit Gaußrauschen

### 3.4.2 CoG

#### Konditionen

Um das Rauschen zu minimieren, wurde vorher ein Mittelwertfilter angewendet. Zwar hat dieser keine kantenerhaltende Eigenschaft, aber auf diese Weise bietet sich die Möglichkeit die Algorithmen auch auf relativ unscharfen und low-resolution ähnlicheren Bildern auf Zuverlässigkeit zu testen. Der manuell gesetzte Schwellwert wurde bei 20 belassen, da Kontraste durch solche Bildoperationen abgeschwächt werden.

### Ergebnisse und Interpretation

Wie zu erwarten war, ist die durchschnittliche Abweichung höher (Tabelle 3.5), als bei den vorherigen Versuchsreihen. Interessant zu beobachten ist aber, dass manche Werte nahezu exakt bis auf die zweite Nachkommastelle mit dem tatsächlichen Wert übereinstimmen und dass, obwohl mit höherem x-Wert der Glint weiter außerhalb der Pupille liegt, die y-Werte in Relation zu den vorherigen Werten stark ansteigen.

Pupillen Zentrum		CoG Schätzung	
x-Position	y-Position	CoG x	CoG y
211	300	209.929	298.988
212	300	211.386	299.591
213	300	213.103	300.54
214	300	215.118	299.965
215	300	216.864	300.534
216	300	215.988	300.733
217	300	216.184	301.115
218	300	216.386	301.489
219	300	217.78	302.308
	Abweichung	-0,9002	0.8939

**Tabelle 3.5:** CoG Schätzung in 3. Versuchsreihe

### 3.4.3 Pérez

#### Konditionen

Ebenfalls wie bei CoG wird zunächst mit einem Mittelwertfilter gefaltet. Alle anderen Einstellungen stimmen mit denen der vorherigen Versuchsreihen überein.

#### Ergebnisse und Interpretation

Die mittlere Abweichung ist hier ebenfalls doppelt (Tabelle 3.6) so hoch, wie bei den ersten Versuchsreihen und wieder fällt auf, dass die Werte, je weiter sich die Pupille vom Reflektionspunkt entfernt, exakter werden.

Pupillen Zentrum		Pérez Schätzung	
x-Position	y-Position	Pérez x	Pérez y
211	300	211.328	296.744
212	300	213.139	297.356
213	300	214.839	298.494
214	300	215.111	299.783
215	300	216.85	299.511
216	300	217.106	299.467
217	300	217.556	299.794
218	300	218.167	300.000
219	300	219.006	300.806
	Abweichung	0.8995	-0.973

**Tabelle 3.6:** Pérez Schätzung in 3. Versuchsreihe

## 3.5 4. Versuchsreihe

### 3.5.1 Beschreibung

Diesmal wird der Pupillenmittelpunkt in x-Richtung von 210 bis 212.9 in 0.1er Schritten bewegt, wovon die Werte 210.1 bis 210.9 den jeweiligen Tabellen zu entnehmen sind, um die Subpixelgenauigkeit testen zu können.

### 3.5.2 CoG

#### Konditionen

Es lagen die gleichen Konditionen wie in der vom Versuchsaufbau äquivalenten Versuchsreihe 1 vor.

#### Ergebnisse und Interpretation

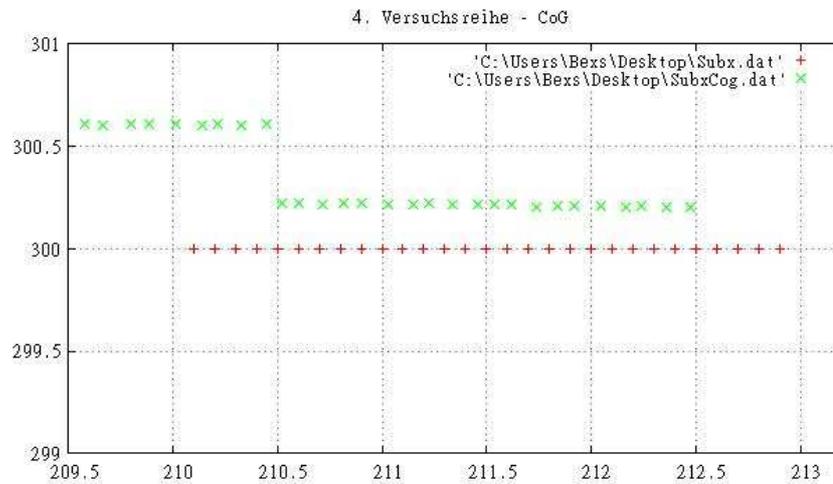
Obwohl die Subpixelverschiebung in x-Richtung stattfindet, wird der y-Wert konstant um eine Verschiebung von 0.6 verschätzt. Wenn man sich nun die Ergebnisse der ersten Versuchsreihe ansieht, bei der ebenfalls eine Verschiebung in x-Richtung, allerdings mit ganzen Pixeln stattgefunden hat, stellt man fest, dass die Genauigkeit im Subpixelbereich geringer ist. Siehe hierzu auch die *Gnuplot*<sup>1</sup> Abbildung 3.5.

Pupillen Zentrum		CoG Schätzung	
x-Position	y-Position	CoG x	CoG y
210.1	300	209.5820	300.6090
210.2	300	209.6690	300.6010
210.3	300	209.8010	300.6080
210.4	300	209.8870	300.6080
210.5	300	210.0140	300.6060
210.6	300	210.1380	300.6000
210.7	300	210.2120	300.6090
210.8	300	210.3290	300.6010
210.9	300	210.4460	300.6090
	Abweichung	-0.4880	0.6

**Tabelle 3.7:** CoG Schätzung in 4. Versuchsreihe

---

<sup>1</sup>[www.gnuplot.info](http://www.gnuplot.info)



**Abbildung 3.5:** Gnuplot Auswertung einer x-Achsen Subpixelverschiebung von 210.1 bis 212.9 und CoG Schätzung

### 3.5.3 Pérez

#### Konditionen

Es lagen die gleichen Konditionen wie in der vom Versuchsaufbau äquivalenten Versuchsreihe 1 vor.

#### Ergebnisse und Interpretation

Hier findet ebenfalls die schon in der CoG Versuchsreihe beobachtete Verschiebung in y-Richtung statt; diese hat sich hier eingependelt auf ca. 0.5. Die Schätzung der für diese Versuchsreihe relevanten x-Werte ist minimal schlechter gegenüber der CoG Schätzung, was nicht zuletzt darauf zurückzuführen ist, dass Pérez et al. auf einer CoG Schätzung aufbaut. Des weiteren fällt hierbei ein sinuskurvenähnlicher Verlauf der Schätzung auf (Abbildung 3.6).

Pupillen Zentrum		Pérez Schätzung	
x-Position	y-Position	Pérez x	Pérez y
210.1	300	209.3610	300.5520
210.2	300	209.4940	300.5000
210.3	300	209.5500	300.5060
210.4	300	209.6110	300.5110
210.5	300	209.7610	300.5440
210.6	300	209.8780	300.5390
210.7	300	209.9220	300.5610
210.8	300	210.0060	300.5560
210.9	300	210.1610	300.5610
	Abweichung	-0.7507	0.5

Tabelle 3.8: Pérez Schätzung in 4. Versuchsreihe

## 3.6 5. Versuchsreihe

### 3.6.1 Beschreibung

Diesmal wird der Pupillenmittelpunkt in y-Richtung von 300 bis 302.9 in 0.1er Schritten bewegt (auch hier ein Auszug der Werte im Bereich von 300.1 bis 300.9) um die Subpixelgenauigkeit testen zu können.

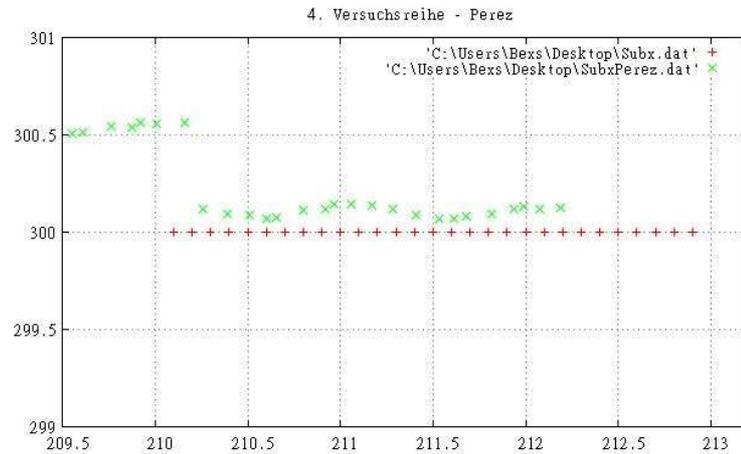
### 3.6.2 CoG

#### Konditionen

Es lagen die gleichen Konditionen wie in der vom Versuchsaufbau äquivalenten Versuchsreihe 2 vor.

#### Ergebnisse und Interpretation

Auch hier ist wieder eine konstante Verschätzung des Wertes zu beobachten, der in dieser Versuchreihe gar nicht verändert wird (Tabelle 3.9), die Verschiebung entspricht ebenfalls fast -0.5. Aus Abbildung 3.7 wird deutlich, wie sehr die Position des Glints (unterhalb des Pupillenmittelpunktes) die Schätzung, die den Pupillenmittelpunkt höher vermutet, beeinflusst.



**Abbildung 3.6:** Gnuplot Auswertung einer x-Achsen Subpixelverschiebung von 210.1 bis 212.9 und Perez Schätzung

### 3.6.3 Pérez

#### Konditionen

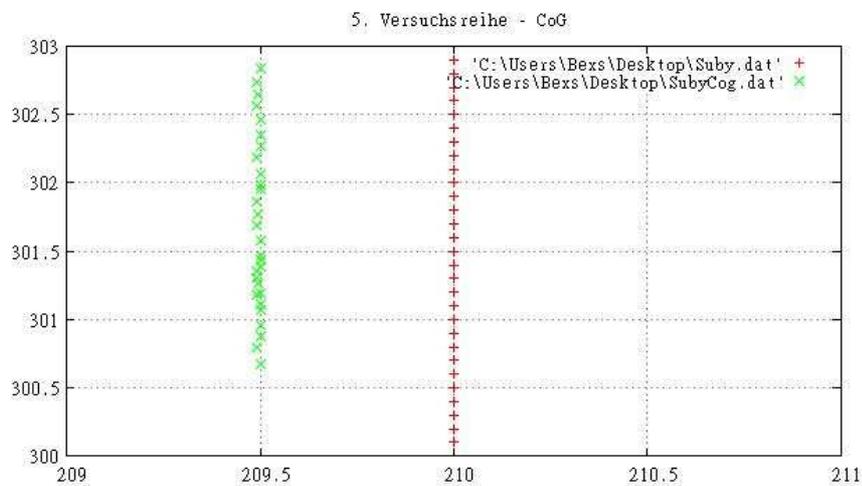
Es lagen die gleichen Konditionen wie in der vom Versuchsaufbau äquivalenten Versuchsreihe 2 vor.

#### Ergebnisse und Interpretation

Auch hier kann man beobachten, dass eine Verschiebung um sogar  $-0.7$  des nicht veränderten Wertes stattfindet und, dass die Schätzung des veränderten y-Wertes gegenüber der CoG Schätzung im Mittel, trotz wiederholtem sinuskurvenähnlichen Verlaufs, minimal besser ist (Abbildung 3.8).

Pupillen Zentrum		CoG Schätzung	
x-Position	y-Position	CoG x	CoG y
210	300.1	209.5000	300.6740
210	300.2	209.4920	300.7930
210	300.3	209.5000	300.8770
210	300.4	209.5000	300.9620
210	300.5	209.5000	301.0710
210	300.6	209.4920	301.1800
210	300.7	209.4960	301.2620
210	300.8	209.4920	301.3550
210	300.9	209.5000	301.4390
	Abweichung	-0.5031	0.5681

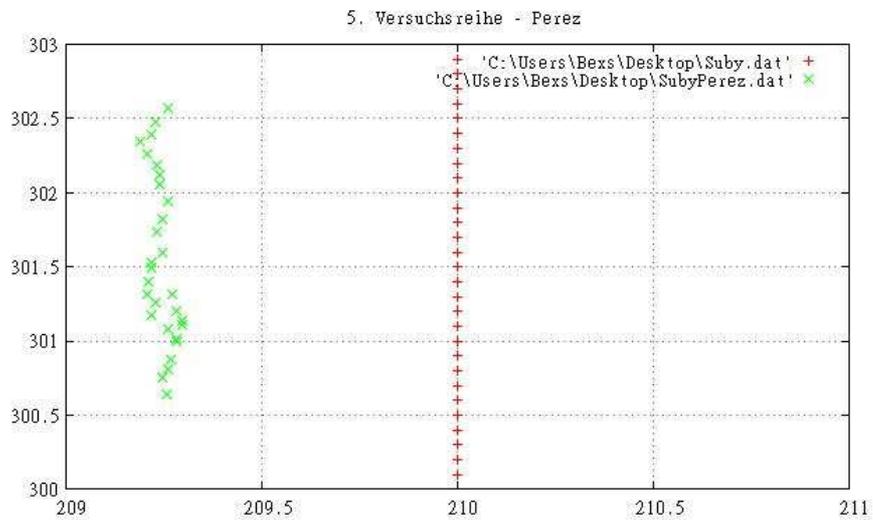
**Tabelle 3.9:** CoG Schätzung in 5. Versuchsreihe



**Abbildung 3.7:** Gnuplot Auswertung einer y-Achsen Subpixelverschiebung von 300.1 bis 302.9 und CoG Schätzung

Pupillen Zentrum		Pérez Schätzung	
x-Position	y-Position	Pérez x	Pérez y
210	300.1	209.2560	300.6390
210	300.2	209.2440	300.7500
210	300.3	209.2610	300.8060
210	300.4	209.2670	300.8720
210	300.5	209.2830	301.0000
210	300.6	209.2940	301.1060
210	300.7	209.2940	301.1390
210	300.8	209.2830	301.2060
210	300.9	209.2720	301.3170
	Abweichung	-0.7273	0.4817

**Tabelle 3.10:** Pérez Schätzung in 5. Versuchsreihe



**Abbildung 3.8:** Gnuplot Auswertung einer y-Achsen Subpixelverschiebung von 300.1 bis 302.9 und Perez Schätzung

# Kapitel 4

## Untersuchung eines Glint Center Estimation Algorithmus

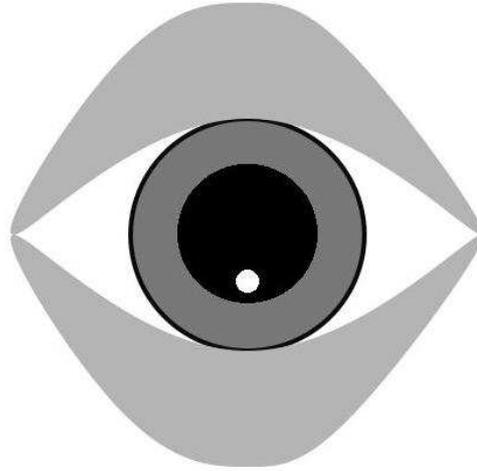
Ebenso wie die Pupil Center Estimation Algorithmen wurde mit den erzeugten synthetischen Bildern ein Glint Center Estimation Algorithmus, der Center of Gravity, auf seine Genauigkeit getestet. Die Versuchsreihen und ihre Schwierigkeitsgrade sind die gleichen, wie bei der im vorhergehenden Kapitel durchgeführten Schätzung des Pupillenmittelpunktes.

### 4.1 Versuchsreihen

#### 4.2 1. Versuchsreihe

##### 4.2.1 Beschreibung

Es liegt ein einfaches Augenbild vor, bei dem Iris und Pupille nicht verdeckt sind. Der initiale Pupillenmittelpunkt befindet sich auf den Koordinaten (210.0, 300.0) und wird im Folgenden um insgesamt 10 Pixel auf der x-Achse auf Position 220.0, in 1 Pixel Sprüngen verschoben. Der Radius der Pupille beträgt 40 Pixel und der der Iris 100. Der Reflektionspunkt befindet sich auf gleicher y-Position, wie der Startmittelpunkt der Pupille und hat eine x-Koordinate von 260.0. Die Strukturen sind einfarbig, die Farbe des Limbus ist, ebenfalls wie die Pupille Schwarz. Es existiert kein Rauschen.



**Abbildung 4.1:** Synthetisches Auge nach Anwendung der Binarisierung des Bereichs um die Pupille mit 1.5-fachen geschätztem Pupillenradius

## 4.2.2 CoG

### Konditionen

Zunächst wird mit Hilfe des CoG für Pupillen der Pupillenmittelpunkt festgelegt. Anschließend erhält man mit Anwendung des Algorithmus nach Pérez et al. den geschätzten Radius der Pupille. Aus Kapitel 2.1.4 [Sch00] wissen wir, dass die Iris ungefähr den 1.5- bis 2-fachen Radius der Pupille hat, damit und mit den bisher bekannten Informationen, kann man den Bereich definieren, der binarisiert werden soll, um dann vom weißen Bereich den Schwerpunkt zu bestimmen. Um die Schleifen so klein, wie möglich zu halten, wird mit dem 1.5-fachen Radius begonnen (Abbildung 4.1), sollte in diesem Bereich kein Glint gefunden werden, wird der Radius iterativ vergrößert.

### Ergebnis und Interpretation

Wie aus Tabelle 4.1 zu entnehmen ist, hat die Bewegung der Pupille in x-Richtung keinerlei Einfluss auf das Ergebnis des Glintzentrums. Es bleibt bei einer von 0.5 in x und 0.59 in y-Richtung. Als Begründung hierfür kann man unter Umständen eine Verschiebung, wie sie auch schon bei den *pupil center estimation* Algorithmen vorgekommen ist, verantwortlich machen und die Tatsache, dass durch das Anti-Aliasing zu viele Pixel mit in die Schwerpunktsberechnung einbezogen werden.

Pupillen Zentrum		Glint Zentrum		CoG Schätzung	
x-Position	y-Position	Glint x	Glint y	CoG x	CoG y
210	300	210	260	209.5	259.41
211	300	210	260	209.5	259.41
212	300	210	260	209.5	259.41
213	300	210	260	209.5	259.41
214	300	210	260	209.5	259.41
215	300	210	260	209.5	259.41
216	300	210	260	209.5	259.41
217	300	210	260	209.5	259.41
218	300	210	260	209.5	259.41
219	300	210	260	209.5	259.41
			Abweichung	0.5	0.59

Tabelle 4.1: CoG Glint Schätzung in 1. Versuchsreihe

## 4.3 2. Versuchsreihe

### 4.3.1 Beschreibung

Gegenüber der vorherigen Versuchsreihe wurde statt einer Translation in x-Richtung eine Verschiebung in die positive y-Richtung vorgenommen. Als Folge davon, wurde die Iris und damit auch der Limbus zu einem Teil verdeckt.

### 4.3.2 CoG

#### Konditionen

Es lagen die gleichen Konditionen, wie bei der 1. Versuchsreihe zu Grunde.

#### Ergebnisse und Interpretation

Die Ergebnisse sind hier äquivalent zu denen aus der 1. Versuchsreihe(Tabelle 4.2).

## 4.4 3. Versuchsreihe

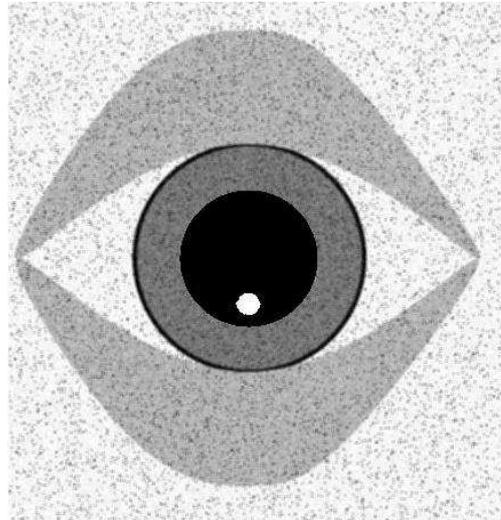
### 4.4.1 Beschreibung

Da die Aufnahmen von Augen in der Realität nicht so perfekt sind, vor allen Dingen, weil wir die Algorithmen in Hinsicht auf low-resolution Aufnahmen testen

Pupillen Zentrum		Glint Zentrum		CoG Schätzung	
x-Position	y-Position	Glint x	Glint y	CoG x	CoG y
210	303	210	260	209.5	259.41
210	304	210	260	209.5	259.41
210	305	210	260	209.5	259.41
			Abweichung	0.5	0.59

**Tabelle 4.2:** CoG Glint Schätzung in 2. Versuchsreihe

wollen, wird in dieser Versuchsreihe ein Gaußrauschen mit  $\mu = 0$  und  $\sigma = 20$  aufaddiert. Die Pupille bewegt sich, wie in Versuchsreihe 1 wieder 10 Pixel in positiver x-Richtung. Es wurde ebenfalls wieder mit dem 1.5-fachen geschätzten Pupillenradius begonnen (Abbildung 4.2).



**Abbildung 4.2:** Synthetisches Auge mit Gauß-Rauschen nach Anwendung eines Mittelwertfilters und der Binarisierung des Bereichs um die Pupille mit 1.5-fachen geschätztem Pupillenradius

#### 4.4.2 CoG

##### Konditionen

Um das Rauschen zu minimieren, wurde vorher ein Mittelwertfilter angewendet. Zwar hat dieser keine kantenerhaltende Eigenschaft, aber so bietet sich die Möglich-

keit die Algorithmen auch auf relativ unscharfen und low-resolution ähnlicheren Bildern auf Zuverlässigkeit zu testen.

### Ergebnisse und Interpretation

Trotz Rauschen kann das Glintzentrum einigermaßen exakt bestimmt werden, allerdings ist auch hier wieder eine Verschlechterung vor allem der nicht veränderten y-Koordinate erkennbar. Durch die vorher angewendeten Bildoperationen und die nahe Schätzung des Pupillenmittelpunktes und -radius, werden weiterhin die hauptsächlich korrekten Pixel durch die Binarisierung dem Glint zugeordnet.

Pupillen Zentrum		Glint Zentrum		CoG Schätzung	
x-Position	y-Position	Glint x	Glint y	CoG x	CoG y
211	300	210	260	209.5380	259
212	300	210	260	209.5380	259
213	300	210	260	209.5030	259
214	300	210	260	209.5030	259
215	300	210	260	209.5030	259
216	300	210	260	209.4720	258.9930
217	300	210	260	209.4440	259.0070
218	300	210	260	209.4440	259.0070
219	300	210	260	209.4440	259.0070
			Abweichung	0.5123	0.9984

**Tabelle 4.3:** CoG Glint Schätzung in 3. Versuchsreihe



# Kapitel 5

## Fazit

### 5.1 Fazit

Betrachtet man zunächst die Ergebnisse der Versuchsreihen der *pupil center estimation* Algorithmen, stellt man fest, dass abgesehen von einer standartmäßigen Verschiebung stellenweise genaue und insgesamt gute Ergebnisse erzielt werden. Allerdings zeigt auch ein Versuch wie die 3. Versuchsreihe, dass sobald Bildstörungen, und die wird es nun einmal in einem *low-resolution* System geben, auftreten, die Algorithmen nicht mehr derart stabil laufen. Man sollte sich hierfür vor allen Dingen vor Augen halten, dass in Versuchsreihe 3 einzig ein Gaußrauschen hinzugeschaltet wurde und, dass eine Abweichung von fast einem Pixel im Verhältnis zu einer Pupille die einen Radius von 40 Pixeln hat, 2.5% beträgt.

Wenn man außerdem auch noch die *Gnuplot* Auswertungen, vor allem der Subpixel-Versuchsreihen, in das Ergebnis mit einbezieht, fallen nicht nur die Verschiebungen auf, sondern auch der sinuskurven-ähnliche Verlauf der Pérez-Schätzung. Ebenso ist interessant zu beobachten, dass gerade die Werte schlechter geschätzt werden, die in der betreffenden Versuchsreihe konstant geblieben sind.

Die Versuchsreihen zeigen des Weiteren, dass der Einfluss der Position des Glints nicht unterschätzt werden darf. Abgesehen von einer Ausnahme, schätzt Pérez die Koordinate, die in der Versuchsreihe verändert wurde, besser als eine CoG Schätzung, obwohl die CoG Schätzung den initialen Wert für den Pérez Algorithmus darstellt.

Die Ergebnisse aus Kapitel 4 unterstreichen die These, dass es Sprünge um ca. 0.5 Pixel gibt. Außerdem geht auch hieraus hervor, dass eine Bildstörung zu einer fast doppelt so hohen Abweichung der Schätzung führt, so dass man letztendlich sagen kann:

Eine gute Vorverarbeitung ist unerlässlich, um den eigentlichen Algorithmen zur

Blickerkennung bzw. in dem hier vorliegenden Fall zur Detektion der Pupillen- und Glimtmittelpunkte die Möglichkeit zu schaffen, stabil und genau arbeiten zu können.

Durch die hier vorliegende Arbeit inklusive des verwendeten Codes zur Generierung synthetischer Augenbilder ist also eine Grundlage geschaffen worden, um generell die Genauigkeit und Performanz von Algorithmen unter variablen Bedingungen zu betrachten. Außerdem hat man hiermit die Möglichkeit, Algorithmen zu verbessern, da im Gegensatz zur Realität immer die gleiche Versuchsumgebung geschaffen werden kann. Folglich können einzelne Versuchsreihen und Algorithmen, deren Kombinationen und Variationen miteinander verglichen werden.

Die hier durchgeführten Versuche und ihre Ergebnisse geben aber keinen allgemeingültigen Aufschluss zu einem gewissen Effekt oder einer Abweichung der Schätzung.

Es sollten also noch weitere Versuche durchgeführt werden, um die hier aufgestellten Thesen empirisch zu widerlegen oder aber zu stützen. Die hier durchgeführten Testreihen haben einen Einblick geben können, sowohl in die möglichen Gründe für eine Fehlschätzung, als auch in die Komplexität, trotz in Relation zur Realität einfacher Augenbilder, der möglichen Permutationen von Versuchsreihen und Vorbedingungen.

### 5.1.1 Probleme

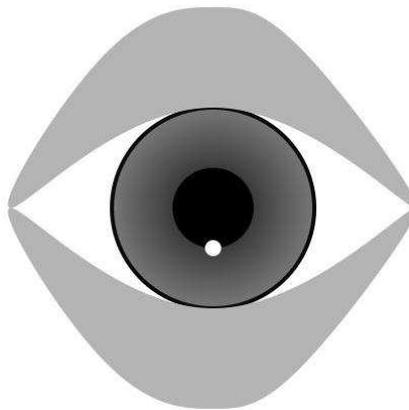
Zunächst stellte sich als Problem heraus *AGG* unter Windows einzubinden bzw. zu verwenden, da *AGG* stellenweise plattformspezifische Libraries bzw. Eigenschaften verwendet, die es zusätzlich einzubinden galt (siehe hierzu auch Anhang A “Hinweise zur Verwendung von *AGG* unter Windows“).

Nachdem die Augenbilder erzeugt waren, mussten die Algorithmen getestet werden, was zum nächsten Problem führte: Das bereits vorhandene System der *AGAS* zum Gaze Tracking von Wladimir Krebs (*gTrack*) besitzt ebenfalls, wie *AGG* plattformspezifische Eigenschaften und baut auf weiteren Bibliotheken auf, welche nicht auf Antriebe unter Windows lauffähig waren. Außerdem ist dieses System ungeeignet, um zunächst nur auf einem Auge zu testen.

Letztendlich blieb die Wahl zwischen der Integration eines weiteren Systems unter Windows oder aber die Algorithmen größtenteils selber zu implementieren (der Algorithmus nach Pérez wurde von Wladimir Krebs System übernommen und an *AGG* angepasst). Auf Grund der bereits langwierigen Arbeit *AGG* unter Windows zu integrieren, entschied ich mich für die Eigenimplementierung der Algorithmen, da außerdem zu dem Zeitpunkt noch kein lauffähiges Plugin für das Einlesen der mit *AGG* generierten synthetischen Augenbilder in das *Gaze-Tracking* System vorhanden war.

### 5.1.2 Ausblick

Durch die vorher genannten Probleme, konnten nicht alle Algorithmen auf ihre Genauigkeit getestet werden. Die synthetischen Bilder sind ebenfalls noch von der Realität und niedrig aufgelösten Augenbildern entfernt. Im Folgenden sollte man also die Algorithmen auf schwierigeren Bildern, bspw. Iris mit Gradientenverlauf testen (Abbildung 5.1), und die einzelnen Module (z.B. Schwellwert- oder Kanten-detectionen) aus denen ein Algorithmus besteht, gegen ein ähnliches austauschen oder um weitere Module/Algorithmen ergänzen, um auf Verbesserung zu testen, oder um mögliche algorithmenspezifische Probleme, die mit fehlender Auflösung auftreten können, frühzeitig zu erkennen und ihnen vorzubeugen.



**Abbildung 5.1:** Synthetisches Augenbild mit Gradientenverlauf innerhalb der Iris

# Kapitel 6

## Anhang A

### 6.1 Hinweise zur Verwendung von *AGG* unter Windows

Der Teil des Codes, der in eigener Arbeit erzeugt wurde, wurde mit Hilfe von *Wascana*<sup>1</sup>, einer Eclipse Distribution, programmiert. Die nun folgende Beschreibung und ihre Bilder bezieht sich darauf. Sie ist auf Eclipse übertragbar, für Entwicklungsumgebungen, wie Visual Studio, müssen die entsprechenden Pfade bzw. umgebungsspezifische Konventionen angepasst werden.

Da *AGG* unter Windows .bmp Dateien erzeugt, muss man zunächst GDI32 als Bibliothek hinzufügen. Arbeitet man unter Linux mit *AGG*, wird in .ppm gespeichert und man muss SDL als zusätzliche library angeben. Des Weiteren ist es notwendig den Pfad zu dem Ordner anzugeben, der die Headerdateien(.h) der *AGG* Bibliothek beinhaltet. An dieser Stelle fügt man auch den Pfad des „lib“ Ordners der in Kapitel 2.2 erwähnten AGAS-eigenen Bibliothek hinzu. Zuletzt muss man alle *AGG* .cpp Dateien, die benötigt werden manuell dem Projekt hinzufügen.

---

<sup>1</sup>[wascana.sourceforge.net](http://wascana.sourceforge.net)

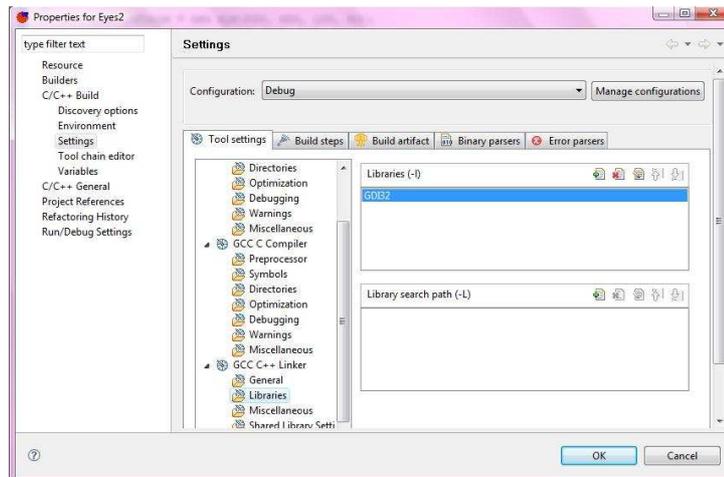


Abbildung 6.1: Inkludieren der GDI32 Bibliothek unter Wascana

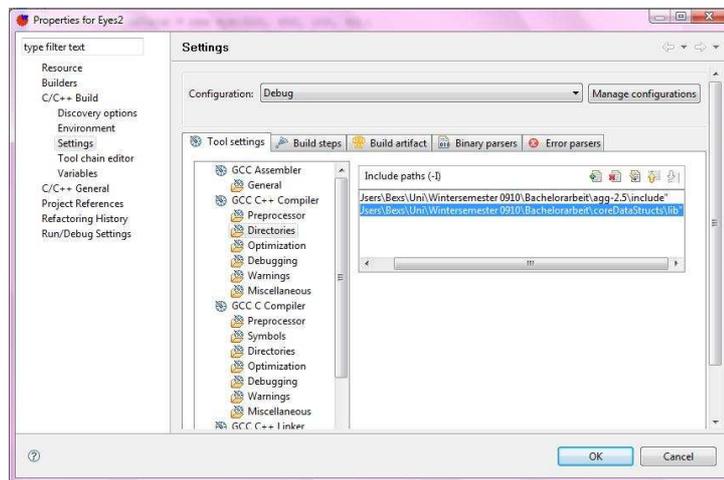


Abbildung 6.2: Angeben der Pfade zu den .h Dateien von AGG und coreDataStructs

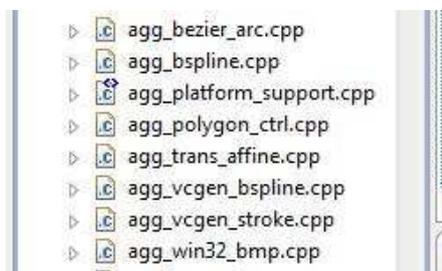


Abbildung 6.3: Integration der nötigen .cpp Dateien

# Literaturverzeichnis

- [Bre65] BRESENHAM, J. E.: Algorithm for Computer Control of a Digital Plotter. In: *IBM System Journal* 4 (1965), Nr. 1, 25–30. <http://dx.doi.org/10.1145/280811.280913>
- [DP10] DROEGE, Detlev ; PAULUS, Dietrich: Pupil center detection in low resolution images. (2010), S. 169–172
- [DR04] DAUNYS, Gintautas ; RAMANAUSKAS, Nerijus: The accuracy of eye tracking using image processing. (2004), S. 377–380
- [Ebi98] EBISAWA, Yoshinobu: Improved Video-Based Eye-Gaze Detection. In: *IEEE Transactions on Instrumentation and Measurement* 47 (1998), Nr. 4
- [FB81] FISCHLER, Martin A. ; BOLLES, Robert C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM*, <http://cacm.acm.org/> 24 (1981), Nr. 6, 381-395. <file:///lab/as/Docs/Articles/Fischler1981RSC.pdf>. ISBN 0001–0782
- [GE06] GUESTRIN, Elias D. ; EIZENMAN, Moshe: General theory of remote gaze estimation using the pupil center and corneal reflections. In: *IEEE Transactions on Biomedical Engineering, TBioEng*, <http://ieeexplore.ieee.org> 53 (2006), Nr. 6, 1124-1133. <http://elmar-inc.com/Papers/Instrumentation/GeneralTheoryOfRemoteGazeEstimationUsingThePupilCenterAndCornealReflections.pdf>
- [Hei10a] HEISE: *Gelähmte Künstler zeichnen mit Eyetracker-Brille*. <http://www.heise.de/open/meldung/Gelaehmte-Kuenstler-zeichnen-mit-Eyetracker-Brille-911819.html>, 2010

- [Hei10b] HEISE: *MWC: Kopfhörersteuerung per Augenbewegung.* <http://www.heise.de/newsticker/meldung/MWC-Kopfhörersteuerung-per-Augenbewegung-934711.html>, 2010
- [Klo09] KLOCKE, Heiner: *Eyetracking - Introduction.* [http://www.gm.fh-koeln.de/~hk/lehre/sgmci/ss2009/folien/1\\_Eyetracking-Introduction\\_sgmci\\_ss09.pdf](http://www.gm.fh-koeln.de/~hk/lehre/sgmci/ss2009/folien/1_Eyetracking-Introduction_sgmci_ss09.pdf), 2009
- [KR99] KIM, Kyung-Nam ; RAMAKRISHNA, R. S.: Vision-Based Eye-Gaze Tracking for Human Computer Interface. In: *IEEE Transactions on Instrumentation and Measurement* (1999)
- [LWP05] LI, Dongheng ; WINFIELD, David ; PARKHUST, Derrick J.: Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. (2005), S. 79–86
- [Mac07] MACWELT: *Eyebox 2: Eye-Tracking für Plakate.* [http://www.macwelt.de/artikel/Artikel/79994/eyebox\\_2\\_eye\\_tracking\\_fuer\\_plakate/1](http://www.macwelt.de/artikel/Artikel/79994/eyebox_2_eye_tracking_fuer_plakate/1), 2007
- [Med07] MEDICALDICTIONARY: *Corneal Limbus of the eye.* [http://medical-dictionary.thefreedictionary.com/\\_/viewer.aspx?path=dorland&name=limbus\\_corneae.jpg](http://medical-dictionary.thefreedictionary.com/_/viewer.aspx?path=dorland&name=limbus_corneae.jpg), 2007
- [OMY02] OHNO, Takehiko ; MUKAWA, Naoki ; YOSHIKAWA, Atsushi: FreeGaze: A Gaze Tracking System for Everyday Gaze Interaction. In: *Proceedings of the symposium on ETRA 2002: eye tracking research and applications symposium*, 2002, 125-132
- [Ots79] OTSU, Nobuyuki: A Threshold Selection Method from Gray-Level Histogramms. In: *IEEE Transactions on Systems, Man and Cybernetics* (1979)
- [PCG<sup>+</sup>03] PÉREZ, A. ; CÓRDOBA, M.L. ; GARCIA, A. ; MÉNDEZ, R. ; MUNOZ, M.L. ; PEDRAZA, J.L. ; SÁNCHEZ, F.: A Precise Eye-gaze Detection and Tracking System. (2003)
- [Sch99] SCHREIBER, Kai: *Erstellung und Optimierung von Algorithmen zur Messung von Augenbewegungen mittels Video-Okulographie-Methoden.* <http://genista.de/manches/diplom/diplom.html>, 1999
- [Sch00] SCHREIBER, Kai: *Das Auge in Zahlen.* <http://www.onjoph.com/patinfo/funktion/zahlen.html>, 2000

- [Sco07] SCOREBERLIN: *Mousetracking and Eyetracking*.  
<http://www.scoreberlin.de/usability-artikel/mousetracking-eyetracking/>, 2007
- [Web10] WEB, Dr: *Eye-Tracking: Möglichkeit, Grenzen und Anwendung*. <http://www.drweb.de/magazin/eye-tracking-moeglichkeiten-grenzen-und-anwendung/#more-24281> [<http://www.webup.biz/einstazmoeglichkeiten.html>], 2010