Erstellung und Optimierung von Algorithmen zur Messung von Augenbewegungen mittels Video-Okulographie-Methoden

Kai Schreiber
Sieben-Höfe-Straße 105
72072 Tübingen

22. Januar 1999

Inhaltsverzeichnis

1 Einführung **7**

1.1 Übersicht . . . 7

1.2 Modell der Augenbewegung . . . 8

1.2.1 Aufbau des Auges . . . 8

1.2.2 Freiheitsgrade des Auges . . . 10

1.3 Mathematischer Formalismus . . . 10

1.3.1 Koordinatensysteme . . . 11

1.3.2 Drehungen . . . 12

1.3.3 Darstellung mit Quaternionen . . . 14

1.4 Motivation der VOG–Methode . . . 16

1.4.1 Nutzen von Augenbewegungsmessungen . . . 16

1.4.2 Vor- und Nachteile verschiedener Meßverfahren . . . 20

1.5 Augenbewegungsmessung mittels VOG . . . 23

1.5.1 Apparative Mittel . . . 23

1.5.2 2–D–Messungen . . . 26

1.5.3 Geometrische Näherung . . . 28

1.5.4 Messung der Torsion . . . 29

2 2–D–Kalibration **35**

2.1 Problemstellung . . . 35

2.2 Bisheriges Verfahren . . . 36

2.3 Neuer Algorithmus . . . 36

2.4 Optimierung im vierdimensionalen Parameterraum . . . 38

2.5 Bestimmung der Kamerarotation . . . 39

2.6 Überprüfung der Stabilität . . . 40

2.6.1 Analytische Überprüfung . . . 40

2.6.2 Überprüfung durch Simulation . . . 40

2.7 Auswirkungen der geometrischen Näherung . . . 41

2.7.1 Verbesserung des Verfahrens? . . . 41

2.7.2 Berechnung des Meßfehlers in Simulationen . . . 41

3 Simulation von Messungen **43**
3.1 Problemstellung 43
3.2 Torsionseignung 44
3.2.1 Halbwertsbreite 44
3.2.2 Gütekurve und meßbare Winkelbreite 44
3.3 Verwendetes Irismuster 46
3.4 Das verwendete Abbildungsmodell 46
3.5 Simulationsablauf 48
3.6 Güte der bewegungsunscharfen Muster 49
3.7 Ergebnisse 49
3.7.1 Qualitative Beurteilung 49
3.7.2 Quantifizierung 50

4 Optimales Segment **55**
4.1 Problemstellung 55
4.2 Musterkriterien 56
4.3 Die Kriterien im Einzelnen 57
4.3.1 Halbwertsbreite des Maximums 57
4.3.2 Frequenzverteilung 57
4.3.3 Qualitative Analyse 60

5 Folgerungen und Ausblicke **61**
5.1 Kalibration des Systems 61
5.1.1 Numerischer Algorithmus 61
5.1.2 Kalibrations–Setup 61
5.2 Exakter Meßalgorithmus 62
5.3 Verkürzung der Belichtungszeit 64
5.4 Segmentanzahlerhöhung 65
5.5 Musterfindung 65
5.5.1 Verfahren zur Musterfindung 66

A Algorithmen **67**
A.1 2-D-Kalibration 67
A.1.1 Kalibration 67
A.1.2 Überprüfung 68
A.2 Belichtungssimulation 70
A.2.1 Berechnung der bewegungsunscharfen Muster 70
A.3 Segmentanalyse 72

B Ergebnisse **75**
B.1 Kalibration . . . 75
B.1.1 Fixationspunkte im Raster . . . 75
B.1.2 Fixtationspunkte zufällig verteilt . . . 77
B.2 Fehler durch die geometrische Näherung . . . 79
B.3 Simulation . . . 84
B.3.1 Meßbare Winkelbreiten . . . 84
B.4 Musterfindung . . . 85

C Optische Augengeometrie **93**
C.1 Geometrie des Auges . . . 93
C.2 Funktionsweise des visuellen Systemes . . . 94
C.3 Aufbau des Auges . . . 95
C.4 Achsen des Auges . . . 95

D Drehpunkte des Auges **97**

E Programmcode **99**
E.1 Kalibration . . . 99
E.1.1 KALIB . . . 99
E.1.2 ABWEICH . . . 101
E.1.3 ABSTAND . . . 101
E.1.4 KALTEST . . . 102
E.1.5 TEST_KALIB . . . 104
E.1.6 DARSTELLEN . . . 107
E.1.7 FINDNAME . . . 108
E.1.8 MESSTEST . . . 109
E.1.9 TEST_MESS . . . 111
E.1.10 MESSE . . . 113
E.2 Belichtungssimulation . . . 114
E.2.1 RTRANS . . . 114
E.2.2 HINTRANS . . . 114
E.2.3 INTENS . . . 115
E.2.4 SIMULROT . . . 116
E.2.5 SIMULPLAN . . . 117
E.2.6 SIMAUSWERTEN . . . 118
E.2.7 GUETEKURVE . . . 119
E.2.8 WINKELBREITE . . . 120
E.2.9 PARAMAX . . . 121
E.2.10 LADEN . . . 122
E.3 Musterfindung . . . 125

E.3.1 HALBWERT 125
E.3.2 FINDBEST 127
E.3.3 BUTTERN 129
E.3.4 FILMCHEN 130

Kapitel 1

Einführung

1.1 Übersicht

Die Augen gelten sprichwörtlich als "Spiegel der" oder auch "Fenster zur Seele". Wer seinen Blick tief in den seines Gegenübers versenkt, kann innigen Kontakt mit ihm oder ihr aufnehmen und erfährt, so die dem Sprichwort zu entnehmende Vorstellung, Verschiedenes über verborgene innere Eigenschaften des Anderen. Denn ablesen läßt sich an den beiden blitzenden Murmeln ja so allerhand: ein schlechtes Gewissen führt zum ausweichenden Blick, Verlegenheit zu scheu niedergeschlagenen Augen. Der wilde Zorn blitzt uns zuerst aus dem Auge entgegen, der Schalk wiederum sitzt zwar zweifellos im Nacken, schaut aber am Ende eben doch zu den Augen heraus — wo auch sonst?

Daß jemand es nicht ganz ehrlich mit uns meint, lesen wir ihm an den Augen ab, ebenso manchen Wunsch und heimliche Absicht, die der Eigentümer der Sehinstrumente, wer weiß, lieber weiterhin verborgen hätte. Vielleicht trägt er ja auch deshalb diese verspiegelte Sonnenbrille.

Seit langem bekannt sind die Aufzeichnungen der Blickfolgen, die beim Betrachten von Bildern entstehen. Wie an diesem Beispiel deutlich erkennbar ist, konzentriert sich die Aufmerksamkeit des Betrachters auf das Gesicht, und hier besonders auf Mund und Augen—den Mund, aus dem gesprochen, und die Augen, aus denen Blickstrahlen versandt werden, eine alte Vorstellung, die vermutlich in den Energiestrahlen der Augen der Comic–Superhelden weiterlebt.

Aber das Blickfolgenbild hat ja noch mehr mit dem Auge zu tun; schließlich ist es überhaupt erst entstanden durch sorgfältige Beobachtung der Augen des Betrachters—was jemandem wichtig ist, so die These dieser Blickfolgenuntersuchungen, erkennt man daran, wie lange er die Dinge ansieht. Und wie lange er die Dinge ansieht, erkennt man an den Bewegungen der Augen.

Aber Beobachtungen der Details dieser Bewegungen verraten noch viel mehr. Der genaue Verlauf der Reaktion auf definierte äußere Reize erlaubt dem Neurologen, Auskunft über den genauen Ort und die Natur von Schädigungen im Gehirn zu geben. Die Untersuchung der normalen Funktionsweise erlaubt eine Beschreibung des neuro–motorischen Apparates, der für die Bewegungssteuerung verantwortlich ist und, in begrenztem Rahmen, auch Aufschluß über die Vorgänge, die bei der Verarbeitung visueller Eindrücke im Gehirn ablaufen.

Tatsächlich also sind die Augen auch in psychologischer und neurologischer Hinsicht durchaus Fenster zur Seele, die allerdings nicht durch innige Blicke, sondern durch genaue Messungen aufgestoßen werden können. Hierfür gab es im Verlauf der Geschichte der Augenbewegungsmessungen verschiedene Verfahren. Über die wichtigsten dieser Verfahren sowie einige Anwendungsbereiche der bei den Messungen gewonnenen Daten werde ich im Abschnitt 1.4 berichten.

Darunter sind auch die durch Fortschritte in der Computertechnologie in immer größerem Umfang möglich werdenden Messungen mittels Video–Okulographiemethoden (VOG), auf deren spezifische Vor– und Nachteile ich kurz eingehen werde. Schließlich werde ich kurz die in der Praxis der VOG angewandten Meßverfahren vorstellen.

Der Hauptteil der Arbeit widmet sich dann einzelnen theoretischen und praktischen Fragen der VOG–Methode.

Es wird in Kapitel 2 ein neues Verfahren zur Kalibration von VOG–Systemen vorgeschlagen und auf seine Stabilität und Konvergenz untersucht. Anschließend wird in Kapitel 3 mit Simulationsserien die Frage untersucht, welchen Einfluß die endliche Belichtungszeit bei Videoaufnahmen auf die Qualität der Messung der torsionellen Komponente der Augenbewegung hat. Schließlich wird in Kapitel 4 ein Verfahren vorgeschlagen, mit dem aus einem gegebenen Bild eines Auges die zur Messung der torsionellen Komponente am besten geeigneten Irissegmente automatisch ausgewählt werden können.

Die Ergebnisse der Simulationen und Berechnungen führen zu einigen Folgerungen für die Praxis der Video–Okulographie, die ich im fünften Kapitel vorstellen werde.

1.2 Modell der Augenbewegung

1.2.1 Aufbau des Auges

Wenn man Detailfragen der Bewegungen des Auges untersuchen oder sich durch Simulationen einen Eindruck von den prinzipiellen Grenzen der Mes-

sung verschaffen möchte, ist es zunächst erforderlich, die Mechanik der Bewegungen mit hinreichender Genauigkeit zu kennen und in mathematisch eindeutiger Weise zu beschreiben.

Für unsere Art der Betrachtung der Augenbewegungen läßt sich der Augapfel als eine Kugel betrachten, die von der Augenmuskulatur um mehrere im Kopf feste Achsen gedreht wird.

Die Symmetrieachse dieser Kugel, die durch den Kugelmittelpunkt und den Mittelpunkt der Pupille führt, heiße optische Achse. Wenn das Auge einen Punkt fixiert, wird angenommen, daß diese optische Achse in die Richtung weist, in der der Punkt zu sehen ist.[1]

Die Iris ist nun eine senkrecht zu dieser optischen Achse stehende, flache Kreisscheibe, deren Mittelpunkt auf der optischen Achse liegt. Die Entfernung des Pupillenmittelpunktes vom Mittelpunkt der Kugel soll R_{auge} oder "Radius des Auges" heißen. Diese Bezeichnung ist in diesem Fall nicht irreführend, weil aufgrund der Eigenarten des angewandten Meßverfahrens der eigentliche Augenradius ohnehin nicht zugänglich ist.

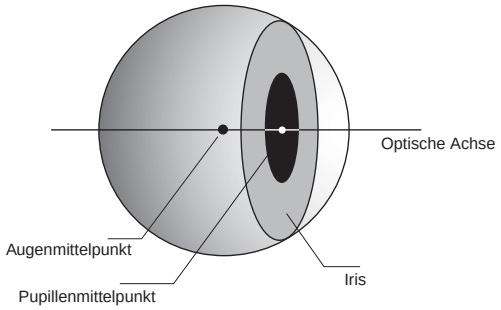

Abbildung 1.1: Optische Achse des Auges und Lage der Iris

Mit dieser Modellannahme über die Form des Auges und der Iris trifft man die tatsächlichen Gegebenheiten hinreichend genau. Der größte verbleibende Fehler betrifft die Bildverzerrungen, die durch die Linsenwirkung der Cornea, das heißt der vor der Iris befindlichen Hornhaut, entstehen.

Diese Wirkung läßt sich bei Verwendung des Modellauges von Gullstrand mit einer einzelnen Vergrößerungslinse modellieren, die dazu führt, daß die scheinbare Iris um etwa 13% vergrößert[2] parallel vor dem tatsächlichen Ort der Iris erscheint. Dies wird bei der VOG durch einen geringfügig größeren Augenradius modelliert und ergibt sich somit aus der Kalibration. Die

[1] Die erheblich komplizierteren tatsächlichen Verhältnisse werden in Anhang C dargestellt. Dort wird auch näher auf das Problem der Blickachse eingegangen.

[2] Vergleiche hierzu [Moore 1996] und [Gullstrand 1909] sowie die Seiten 59–61 in [Le Grand 1980].

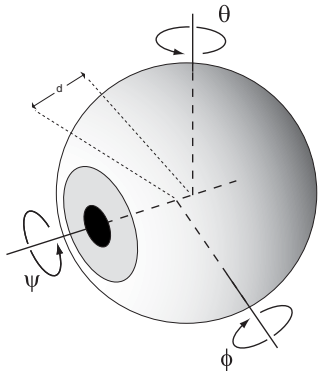

Abbildung 1.2: Freiheitsgrade und Drehachsen des Auges in Referenzposition. Die Rotation um θ erfolgt um die y–Achse, die Rotation um ϕ um eine gegenüber der x–Achse in negativer z–Richtung um d verschobene Achse. Vergleiche zur Lage der Achsen Abbildung 1.4.

Verzerrung durch die Cornea braucht also nicht gesondert berücksichtigt zu werden.

1.2.2 Freiheitsgrade des Auges

Das Auge hat für die Bewegung nun drei Freiheitsgrade (horizontal, vertikal, torsionell) innerhalb des Kopfes[3]. Die Bewegung wird durch eine Kombination aus drei Drehungen um drei unabhängige Achsen beschrieben. Empirische Untersuchungen der Mechanik haben hier ergeben, daß die tatsächliche Bewegung des Auges von einem Modell gut erfaßt wird, das die Drehachse für horizontale Drehungen senkrecht durch den Augenmittelpunkt legt, wogegen die Drehachse für vertikale Drehungen durch einen um einen Abstand d nach vorne verschobenen Punkt auf der z–Achse verläuft[4] (Siehe Abbildung 1.2).

Der dritte Freiheitsgrad, die sogenannte Torsion, wird durch eine Drehung um ψ um die optische Achse modelliert.

1.3 Mathematischer Formalismus

Die mathematische Beschreibung dieser Bewegung erfolgt entweder über Drehmatrizen, oder, inhaltlich gleichwertig und formal eleganter—aber weniger

[3]Translationen des Auges innerhalb der Augenhöhle spielen keine Rolle.

[4]Für eine genauere Darstellung der Verhältnisse bei der Drehung des Auges im Kopf verweise ich auf Anhang D.

verbreitet—über Vierer-Vektoren, sogenannte Quaternionen. Obwohl in dieser Arbeit grundsätzlich die Darstellung mit Drehmatrizen für alle Probleme gewählt wird, werde ich am Ende dieses mathematischen Abschnittes aus Gründen der Vollständigkeit einen kurzen Überblick auch über die Formulierung mit Quaternionen geben.

1.3.1 Koordinatensysteme

Da für spätere Betrachtungen stets definierte Punkte oder Punktmengen auf der Iris von Bedeutung sind, wird für solche Punkte zunächst ein Polarkoordinatensystem (Abbildung 1.3) eingeführt, dessen Ursprung im Zentrum der Pupille liegt, und dessen r-Achse bei geradeaus blickendem Auge von vorne betrachtet nach rechts weisen soll. In diesem augefesten Koordinatensystem hat ein Punkt die Koordinaten $\vec{x}_{iris} = \binom{r_{iris}}{\varphi_{iris}}$.

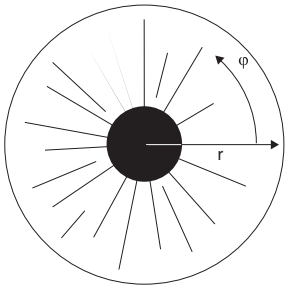

Abbildung 1.3: Koordinatensystem auf der Iris

Für die dreidimensionale Beschreibung der Koordinaten wählen wir ein kopffestes, euklidisches rechtshändiges Koordinatensystem, dessen positive x–Achse von vorne betrachtet nach rechts, dessen positive y–Achse von vorne betrachtet nach oben und dessen positive z–Achse nach vorne in Referenzrichtung weisen soll (Abbildung 1.4). Referenzrichtung heißt dabei jene Richtung, in die die optische Achse weist, wenn das Auge geradeaus nach vorne blickt. Dieses Koordinatensystem heißt von nun an das "kopffeste Koordinatensystem".

Im kopffesten Koordinatensystem hat der Pupillenmittelpunkt in Referenzposition (Blick nach vorne) die Koordinaten

$$\vec{x}_{ref,kopf} = \begin{pmatrix} 0 \\ 0 \\ R_{auge} \end{pmatrix} \tag{1.1}$$

Dabei ist R_{auge}, wie im vorigen Abschnitt beschrieben, der Abstand vom Zentrum des Augapfels zum Mittelpunkt der Pupille.

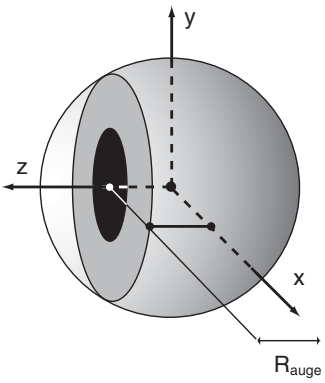

Abbildung 1.4: Lage der Achsen des kopffesten Koordinatensystems, Auge in Referenzposition

1.3.2 Drehungen

Die im Folgenden notwendige Zerlegung einer allgemeinen Drehung des Auges in Einzeldrehungen um drei verschiedene Achsen wirft ein grundsätzliches mathematisches Problem auf: da Drehungen um endliche Winkel nicht kommutativ sind, ist es von großer Bedeutung für den Wert der Drehwinkel, in welcher Reihenfolge die einzelnen Drehungen gedacht werden und ob es sich um aktive oder passive Drehungen handelt.

Bei passiven Drehungen wird um mit dem Objekt geführte Achsen gedreht, das heißt eine Drehung des Auges um 45° nach links führt dazu, daß eine anschließende Drehung nach oben nicht um die kopffeste horizontal von Ohr zu Ohr verlaufende Achse sondern um die mit dem Auge um 45° gedrehte Achse erfolgt. Offensichtlich ist die Endposition eine andere, wenn—wie bei aktiven Drehungen—die Achse für vertikale Drehungen durch die horizontale Drehung nicht verändert wird[5].

Für die Beschreibung dreidimensionaler Drehungen gibt es zusätzlich bei gleichen Drehachsen mehrere Möglichkeiten, die Reihenfolge der Drehungen zu wählen, von denen dann abhängt, welche Drehwinkel zum Erreichen einer gegeben Endposition nötig sind.

Zwei gebräuchliche Varianten sind die Fick– und die Helmholtz–Winkel. Im Helmholtz–System wird (bei Betrachtung aktiver Drehungen) zunächst die horizontale, dann die vertikale und zuletzt die torsionelle Drehung ausgeführt, in Fick–Winkeln erfolgt zuerst die vertikale, dann die horizontale, und schließlich die torsionelle Drehung[6].

[5] Wie man sich leicht klarmacht, führt ein Wechsel von aktiver zu passiver Drehung genauer dazu, daß sich die Reihenfolge der Drehungen umkehrt.

[6] Siehe [Fick 1854] und [Helmholtz 1963]

Alle Berechnungen in dieser Arbeit erfolgen mit aktiven Drehungen im Helmholtz–Winkel–System[7].

Die Koordinaten eines beliebigen Punktes im kopffesten Koordinatensystem erhalten wir durch Verschiebung des Referenzpunktes um r_{auge} in Richtung der positiven x–Achse und anschließende Drehung um φ_{auge} um die z–Achse in mathematisch positiver Richtung.

Ein beliebiger Punkt auf der Iris hat damit im kopffesten Koordinatensystem die Koordinaten

$$\vec{x}_{kopf} = \begin{pmatrix} \cos(\varphi_{iris}) & -\sin(\varphi_{iris}) & 0 \\ \sin(\varphi_{iris}) & \cos(\varphi_{iris}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{iris} \\ 0 \\ R_{auge} \end{pmatrix} \tag{1.2}$$

und somit:

$$\vec{x}_{kopf} = \begin{pmatrix} \cos(\varphi_{iris}) r_{iris} \\ \sin(\varphi_{iris}) r_{iris} \\ R_{auge} \end{pmatrix} \tag{1.3}$$

Dieser im kopffesten Koordinatensystem beschriebene Punkt wird nun gemäß Abschnitt 1.2 zunächst um den Winkel θ in horizontaler Richtung gedreht, anschließend um den Winkel ϕ in vertikaler.

Da die Drehung in vertikaler Richtung um einen anderen Drehpunkt erfolgt, wird zunächst die Drehmatrize für die horizontale Drehung anmultipliziert, dann der Ursprung um den Parameter d in z–Richtung verschoben, nun die Drehmatrize für die vertikale Drehung aufmultipliziert und schließlich der Ursprung um d in negativer z–Richtung zurückgeschoben.

Damit erhalten wir für die Koordinaten eines allgemeinen Punktes auf dem Auge, wenn das Auge um θ in horizontaler und um ϕ in vertikaler Richtung gedreht wurde, folgenden Ausdruck:

$$\begin{aligned} \vec{x}'_{kopf} &= \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix} \cdot \\ & \left[\begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & cos(\theta) \end{pmatrix} \vec{x}_{kopf} - \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix} \right] \end{aligned} \tag{1.4}$$

Die eckige Klammer ergibt dann, setzt man $\vec{x}_{kopf}$ wie oben:

[7] Für weitere Details der mathematischen Beschreibung von Augenbewegungen siehe [Haslwanter 1995] und [Tweed 1996].

$$\begin{bmatrix} \cdots \end{bmatrix} = \begin{pmatrix} \cos(\theta)\cos(\varphi_{iris})r_{iris} + \sin(\theta)R_{auge} \\ \sin(\varphi_{iris})r_{iris} \\ -\sin(\theta)\cos(\varphi_{iris})r_{iris} + \cos(\theta)R_{auge} - d \end{pmatrix} \tag{1.5}$$

Dies ergibt insgesamt für die Koordinaten eines allgemeinen Punktes auf der Iris nach Drehung des Auges im kopffesten System, wenn man

$$\mathcal{Z} = -\sin(\theta)\cos(\varphi_{iris})r_{iris} + \cos(\theta)R_{auge} - d \tag{1.6}$$

setzt:

$$\vec{x}'_{kopf} = \begin{pmatrix} \cos(\theta)\cos(\varphi_{iris})r_{iris} + \sin(\theta)R_{auge} \\ \cos(\phi)\sin(\varphi_{iris})r_{iris} - \sin(\phi)\mathcal{Z} \\ \sin(\phi)\sin(\varphi_{iris})r_{iris} + \cos(\phi)\mathcal{Z} + d \end{pmatrix} \tag{1.7}$$

und insbesondere für die Koordinaten des Pupillenmittelpunktes ($r_{iris} = 0$ und damit $\mathcal{Z} = \cos(\theta)R_{auge} - d$) in beliebiger Position:

$$\vec{x}_{pupille,kopf} = \begin{pmatrix} \sin(\theta)R_{auge} \\ \sin(\phi)[d - \cos(\theta)R_{auge}] \\ \cos(\phi)\cos(\theta)R_{auge} + d(1 - \cos(\phi)) \end{pmatrix} \tag{1.8}$$

1.3.3 Darstellung mit Quaternionen

Quaternionen wurden zuerst durch den irischen Mathematiker William Rowan Hamilton in die Mathematik eingeführt, der Legende zufolge "sprangen die Quaternionen ins Licht, oder ins Leben, voll entwickelt", als er mit seiner Frau die Brougham Bridge in Dublin überquerte. Hamilton war wie elektrisiert und gravierte die grundlegenden Rechenregeln auf der Stelle mit einem Messer in die hölzerne Brüstung.

Der Grundgedanke der Quaternionen ist die Erweiterung des gewöhnlichen, dreidimensionalen rechtshändigen euklidischen Koordinatensystems um eine weitere skalare Komponente, so daß ein Quaternion als

$$\mathbf{q} = q_0 + q_1\vec{e}_1 + q_2\vec{e}_2 + q_3\vec{e}_3 = q_0 + \vec{q} \tag{1.9}$$

geschrieben werden kann. Hier sind die $\vec{e}_i$ die gewöhnlichen Normaleneinheitsvektoren des $\mathcal{R}^3$. Zusätzlich zu den üblichen Vektoroperationen existiert nun eine nichtkommutative, bilineare und assoziative Operation, das *Quaternionen–Produkt*, das folgenden Regeln gehorcht:

$$\vec{e}_i\,\vec{e}_i = -1$$

$$\begin{aligned}
\vec{e}_1\,\vec{e}_2 &= -\vec{e}_2\,\vec{e}_1 = \vec{e}_3 \\
\vec{e}_2\,\vec{e}_3 &= -\vec{e}_3\,\vec{e}_2 = \vec{e}_1 \\
\vec{e}_3\,\vec{e}_1 &= -\vec{e}_1\,\vec{e}_3 = \vec{e}_2
\end{aligned}$$

Für Berechnungen angenehmer ist allerdings folgendes Schema für das Produkt $r = p\,q$:

$$\begin{aligned}
r_0 &= p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3 \\
r_1 &= p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2 \\
r_2 &= p_0 q_2 + p_2 q_0 - p_1 q_3 + p_3 q_1 \\
r_3 &= p_0 q_3 + p_3 q_0 + p_1 q_2 - p_2 q_1
\end{aligned}$$

Oder, kürzer:

$$\mathbf{r} = q_0\mathbf{p} + p_0\mathbf{q} + \mathbf{q} \times \mathbf{p} \tag{1.10}$$

Der *Betrag* eines Quaternions ist $|\mathbf{q}| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$. Zu jedem Quaternion existiert ein Inverses $\mathbf{q}^{-1}$, so daß für ihr Quaternionenprodukt $\mathbf{q}\,\mathbf{q}^{-1} = \mathbf{q}^{-1}\,\mathbf{q} = 1$ gilt. Wie sich leicht zeigen läßt, ist das Inverse zu q

$$\mathbf{q}^{-1} = \frac{q_0 - q_1 - q_2 - q_3}{|\mathbf{q}|^2} = \frac{q_0 - \vec{q}}{|\mathbf{q}|^2} \tag{1.11}$$

Für die Formulierung räumlicher Drehungen mittels Quaternionen ist die Darstellung eines Einheitsquaternions als

$$\hat{\mathbf{q}} = \cos(\Theta) + n_1 \sin(\Theta) + n_2 \sin(\Theta) + n_3 \sin(\Theta) = \cos(\Theta) + sin(\Theta)\hat{n} \tag{1.12}$$

von Vorteil. $\hat{n}$ heißt in diesem Fall die *Achse*, Θ der *Winkel* des Quaternions. Achse und Winkel eines Nicht–Einheitsquaternions lassen sich durch Normalisierung und Bestimmung der Achse und des Winkels des zugeordneten Einheitsquaternions berechnen.

Die Operation

$$\vec{v}' = \mathbf{q}\,\vec{v}\,\mathbf{q}^{-1} \tag{1.13}$$

rotiert nun den Vektor $\vec{v}$ um die Achse von q um den doppelten Winkel von q.

Der Hintereinanderausführung von Drehungen entspricht dabei die Drehung mit dem Quaternionenprodukt der zugeordneten Quaternionen. Die x–te Potenz eines Quaternions, die nach

$$\mathbf{q}^x = |\mathbf{q}|^x[\cos(x\,\Theta) + \hat{n}\sin(x\,\Theta)] \tag{1.14}$$

berechnet werden kann, entspricht offensichtlich einer Drehung um den x–fachen Drehwinkel.

Schließlich läßt sich durch Bildung von rqr^{-1} die Drehachse von q entsprechend der Abbildung, die das Quaternion r vermittelt, drehen, wobei der Drehwinkel von $\mathbf{q}$ erhalten bleibt.

Wie aus dieser kurzen Übersicht klar wird, ist die Darstellung von komplizierten dreidimensionalen Drehungen durch Quaternionen an Einfachheit und Kompaktheit schwer zu überbieten. Ich habe mich aus Gründen der Übersichtlichkeit—und obwohl Hamilton selbst die Quaternionen für seine größten Beitrag zur Mathematikgeschichte hielt—dafür entschieden, in dieser Arbeit der gängigen Darstellung mit Matrizen zu folgen. Diese kurze Darstellung des Quaternionenkalküls folgt Abschnitt 7 von [Tweed 1996].

1.4 Motivation der VOG–Methode

In diesem Abschnitt, der in allen Punkten Übersichtscharakter hat und nicht ins Detail gehen kann, werden kurz ausgewählte Anwendungsbereiche der Augenbewegungsmessung oder Okulographie (OG) dargestellt. Es werden im direkten Vergleich mit anderen Methoden Vor– und Nachteile der VOG-Methode dargestellt. Die Darstellung folgt im wesentlichen [Huber 1998] und [Zee 1991].

1.4.1 Nutzen von Augenbewegungsmessungen

Wie in der Einleitung bereits angedeutet, ist die genaue Messung von Augenbewegungen für verschiedene Forschungsbereiche von Interesse. Dabei bestehen je nach Fragestellung verschiedene Anforderungen an die verwendeten Meßverfahren.

Für einen kurzen Einblick in die Anwendungsbereiche der OG und die Arbeitsweise des visuellen neurologischen Systemes möchte ich vier wichtige Felder heranziehen und im Folgenden eingehender beschreiben. Dabei habe ich insbesondere im neurologischen Teil eine ausführlichere Übersicht der Mechanismen und Gesetzmäßigkeiten gegeben, die durch Augenbewegungsmessung bereits erforscht wurden. Diese Übersicht dient vor allem dem Zweck, eine Einschätzung der Komplexität der Augenbewegungen zu erlauben und die Anforderungen an die Meßapparaturen im Bereich neurologischer Forschung zu definieren.

1.4.1.1 Psychologie und Neuropsychologie

Zu den frühesten Anwendungsgebieten der OG gehörten sicherlich die wahrnehmungspsychologischen Untersuchungen von Blickfolgen.

Hier werden meist die Fixationspunkte bei Präsentation eines visuellen Reizes ermittelt, und aus der Blickfolge, der Fixationsdauer und ähnlichen Parametern Aufschluß über psychologische Mechanismen der Bildwahrnehmung und –verarbeitung gewonnen.

Weitere, eher in den Bereich der Neuropsychologie gehörende Fragestellungen, die sich mit okulographischen Methoden untersuchen lassen, betreffen zum Beispiel die visuelle Wahrnehmung und Bildverarbeitung im Gehirn oder Fragestellungen aus dem Bereich der Psycholinguistik, die sich durch das Beobachten der Augenbewegungen beim Lesen eines Textes beantworten lassen[8].

1.4.1.2 Neurologie

Gegenstand der Forschung im Bereich der Augenbewegungen sind neben der Mechanik und Physiologie des Auges unter anderem die verschiedenen Reflexbögen und neurologischen Systeme, die das Auge zum Beispiel während Kopfbewegungen steuern.

Dabei erlaubt eine genaue Kenntnis der neurologischen Verhältnisse nicht nur tiefe Einblicke in die Funktionsweise der beteiligten Teile des Gehirnes, sondern in der Diagnose über die im nächsten Abschnitt beschriebenen Mechanismen durch die starke Modularität der verschiedenen Bewegungsformen auch genauen Aufschluß über den Ort eines neurologischen Defektes.

Der Vestibulo–Okuläre Reflex (VOR) Die zwei hauptsächlichen Aufgaben des Gehirnes bei der Steuerung des Auges sind zum einen die Fixierung der Blickrichtung im Raum bei Bewegungen des Kopfes[9], und zum anderen die willentliche Veränderung der Sichtlinie, die für die visuelle Fixation von Bedeutung ist[10].

Für die erste Aufgabe, die Fixierung der Blickrichtung bei Kopfbewegungen, sind die Vestibulo–Okulären Reflexe zuständig, die durch Informationen aus dem Vestibulären System gesteuert werden.

[8] Eine ausführliche Darstellung hierzu findet sich in [Ygge 1994].

[9] Dies ist besonders wichtig, da nur ein über längere Zeit ortsfestes Bild auf der Netzhaut zum Sehvorgang führt.

[10] Zur Beschreibung des optischen Apparates und zur Definition der Sichtlinie siehe Anhang C.

Es gibt hier zwei wichtige Reflexbögen. Zum einen werden die Informationen aus den Winkelgeschwindigkeits–Sensoren der halbkreisförmigen Bogengänge des Gleichgewichtssystems verarbeitet. Dies ermöglicht ein Nachstellen der Augenposition bei Drehung oder Verkippung des Kopfes, und damit das Aufrechterhalten der Sehfähigkeit bei Rotationen. Die Augen werden hierbei in Gegenrichtung der Drehung mit derselben Winkelgeschwindigkeit geführt.

Der zweite Bereich des VOR, der Otolitho–Okuläre–Reflex, wertet Signale aus den ebenfalls im Gleichgewichtsapparat befindlichen Otolithen aus, die auf Linearbeschleunigungen des Kopfes reagieren.

Der VOR ist dabei einer direkten optischen Rückkopplung (d.h. einer Auswertung der auf der Netzhaut aufgenommenen Bilder und einer anschließenden darauf beruhenden Richtungskorrektur des Auges) durch seine viel kürzere Zeitkonstante deutlich überlegen: Während der VOR eine Antwortzeit von nur etwa 10 ms bis 12 ms hat, braucht das visuelle System für die Verarbeitung um 100 ms.

Optokinetischer Reflex Dennoch hat auch die visuelle Rückkopplung Anteil am Reflexverhalten des Auges, da nämlich bei anhaltenden Rotationen aufgrund der mechanischen Eigenschaften der Kanäle das Signal aus dem vestibulären System nach etwa 30 Sekunden abgeklungen ist und die visuellen Reflexe in etwa mit derselben Zeitkonstanten aktiv werden. So kann auch bei fortgesetzten Drehungen die Sehfähigkeit aufrecht erhalten werden.

Schnellphasige Sakkaden Zusätzlich zum oben beschriebenen VOR und der visuellen Rückkopplung ist ein Korrekturverfahren nötig, das beispielsweise ein Abwandern der Blickrichtung an den Rand des Sichtfeldes bei anhaltenden Rotationen verhindert. Diese Funktion erfüllen die schnellen Phasen des sakkadischen Systemes, die eine Rückführung des Auges in die Nähe der zentralen Position vornehmen.

Die schnellen Phasen erreichen dabei Winkelgeschwindigkeiten von bis zu $500\,^{\circ}/\mathrm{s}$.

Willkürliche Sakkaden Tiere, deren Augen eine Fovea[11] aufweisen, haben die Fähigkeit entwickelt, die Sichtlinie auch ohne Kopfbewegungen zu verändern. Solche willkürlichen Sakkaden können von optischen oder akustischen Reizen oder aus der Erinnerung ausgelöst werden.

[11] Die Fovea ist der sogenannte "gelbe Fleck" auf der Netzhaut. Dort wird das projizierte Bild am besten aufgelöst. Vergleiche hierzu Anhang C.

Augenfolgebewegungen Durch Sakkaden alleine ist das Bild von Objekten, insbesondere das bewegter Objekte, nicht dauerhaft auf einer Stelle der Netzhaut zu halten. Deswegen verfügt das Gehirn foveater Tiere über einen speziellen Mechanismus, der die gleitende Nachführung des Auges erlaubt ("smooth pursuit"). Der Input dieses Systemes speist sich dabei nicht nur aus dem visuellen Apparat, sondern kann aus ganz unterschiedlichen Bereichen stammen. So können manche Testpersonen beispielsweise den Bewegungen ihrer Hand bei völliger Dunkelheit folgen.

Davon vermutlich verschieden ist der visuelle Apparat, der die Fixation stillstehender Objekte erlaubt. Obwohl auch die Meinung existiert, daß es sich dabei um einen Spezialfall der Augenfolgebewegung mit Objektgeschwindigkeit Null handelt, deuten jüngere Ergebnisse auf die Existenz eines eigenen Fixationssystems für unbewegte Ziele hin.

Kombinierte Augen–/Kopfbewegungen Die vorstehenden Steuersysteme der Augenbewegungen werden im Falle kombinierter Augen– und Kopfbewegungen durch andere Funktionsweisen und Gesetze ergänzt und kompliziert. Im Falle langsamer Bewegungen, bei denen Kopf und Blick einem bewegten Objekt gemeinsam folgen, würde ein funktionierender VOR beispielsweise durch Kompensation der Kopfbewegung eine Fixation unmöglich machen. Tatsächlich findet man ein Aussetzen des VOR für solche Bewegungen.

Vergenz Der letzte größere Bereich von Augenbewegungen betrifft die zur binokulären Erfassung von Objekten—und damit zum räumlichen Sehen—notwendige Koordination zwischen den Bewegungen beider Augen. Dabei muß einmal die Blickrichtung der beiden Augen koordiniert werden (fusionale Vergenz), der sich dabei ergebende Parallaxenwinkel zwischen den Blickrichtungen beider Augen ist dabei um so größer, je näher das fixierte Objekt dem Betrachter ist. Zum anderen ist eine gemeinsame Akkomodation (akkomodative Vergenz) auf das betrachete Objekt nötig.

1.4.1.3 Diagnostik

In der Neurologie verwendet man die bei der OG gewonnenen Daten routinemäßig bei der Diagnose neuronaler Läsionen und Ausfälle. Die besonders gute diagnostische Verwendbarkeit der Augenbewegungsdaten gegenüber anderen Quellen rührt dabei von einer Reihe von Besonderheiten her, die die Interpretation gewonnener Daten erleichtern.

- Es fehlt beim Auge der für andere Glieder typische monosynaptische

Dehnungsreflex—was durch die gleichbleibende mechanischen Last, gegen die die Augenmuskeln zu arbeiten haben, zu erwarten ist.

- Beim Auge treten—wie im vorigen Abschnitt angedeutet— eine ganze Reihe von klar unterscheidbaren Bewegungsformen auf, die alle eine spezifische Funktion, bestimmte Eigenschaften und ein definierte anatomisches Substrat aufweisen. Das Ausbleiben einer bekannten Reaktionsweise des visuellen Apparates führt daher in vielen Fällen rasch zu einer Lokalisation der neurologischen Störung.

- Augenbewegungen lassen sich verhältnismäßig einfach und vor allem leicht quantifizierbar messen. Sie erlauben damit genauen Aufschluß über die Arbeitsweise des motorischen Systems und die Art der neurologischen Defekte.

- Augenbewegungen sind fast reine Rotationen und damit besonders gut im Modell beschreibbar.

- Das Auge als bewegter Teil hat im Vergleich zu anderen Gliedern verschwindend geringe Trägheit.

1.4.1.4 Praktische Anwendungen

Die Augenbewegungsmessung, vor allem in der 2-D-Variante, birgt neben dem in den letzten Abschnitten angedeuteten medizinischen und wissenschaftlichen Nutzen auch Möglichkeiten zur praktischen Anwendung.

Ein wichtiger Anwendungsbereich ist dabei die Blicksteuerung technischer Geräte und Computer. Dies ist zum einen in Umgebungen interessant, in denen die Hände für andere Tätigkeiten frei bleiben sollen, zum Beispiel in der bemannten Raumfahrt, erlaubt irdisch vor allem aber auch körperlich behinderten Menschen die Nutzung ihnen sonst nicht zugänglicher Geräte.

1.4.2 Vor- und Nachteile verschiedener Meßverfahren

1.4.2.1 Nachbilder

Eine der frühesten, beispielsweise von Helmholtz und Listing[12] angewandten Methoden zur Messung von 3-D-Augenbewegungen war die Methode der Nachbilder, bei der von den Testpersonen ländere Zeit ein räumliches Gitter an einem vorgegebenen Punkt fixiert wird. Anschließend wird die Augenposition (der Fixationspunkt) verändert und das nun gesehene Gitter mit dem von der Netzhaut erzeugten Nachbild aus der ersten Fixation verglichen.

[12] [Helmholtz 1963]

Daraus läßt sich insbesondere die torsionelle Komponente der Augenbewegung bestimmen, was von Listing und Helmholtz zur Ermittlung von Listings Gesetz verwendet wurde.

Listings Gesetz ist eine Konkretisierung von Donders Gesetz[13], das besagte, daß zu jeder 2-D-Augenposition ein eindeutiger Torsionswinkel gehört, die Bewegung des Auges also nicht in vollen drei Dimensionen sondern einem zweidimensionalen Unterraum stattfindet. Listings Gesetz besagt, daß diese Torsion so ausfällt, als ob die Rotation in eine bestimmte Position aus der Referenzposition um eine einzelne Achse ausgeführt würde, die senkrecht auf der Start– und Endrichtung der optischen Achse des Auges steht. Das Auge befolgt die Vorgabe von Listings Gesetz in der Praxis während Fixationen sehr gut ($\sigma \leq 1^\circ$)

Durch ihre geringe Exaktheit und ihre beschränkte Anwendbarkeit ist die Nachbildermethode heute hauptsächlich von historischem Interesse.

1.4.2.2 Verfahren mit Kontaktlinsen–Spulen

Ein seit langer Zeit erfolgreich angewandtes Verfahren zur Messung von Augenbewegungen stellt das Aufbringen von magnetischen Induktionsspulen auf die Hornhaut dar[14]. Durch oszillierende äußere Magnetfelder wird in diesen Spulen induktiv Strom erzeugt, der als Meßsignal abgenommen wird. Mittels zwei nichtparalleler Spulen kann man die Augenposition exakt bestimmen. Da das Anbringen von zwei Spulen unhandlich ist, wählt man eine spezielle Form der Spulen, die dann vollständig in eine einzelne Kontaktlinse eingebettet werden können.

Die großen Vorteile der Spulen–Methode liegen in der hohen zeitlichen und räumlichen Auflösung. Auf der anderen Seite stehen als Nachteile die Invasivität der Methode und die Stationarität des Apparates, die beide insbesondere bei klinischen Patienten die Untersuchung sehr erschweren. Zwar gibt es Versuche, mit kleinen, kompakten Magnetfeldspulen eine mobile Meßvorrichtung zu erhalten, doch spricht auch in diesem Fall die umständliche Anwendung der Kontaktlinsen gegen eine klinische Verwendung des Verfahrens.

1.4.2.3 Elektro–Nystagmographie (ENG)

Die Messung der Augenposition im Kopf ist nicht nur durch direkte optische oder elektromagnetische Beobachtung, sondern auch durch Messung der elektrischen Dipolmomentes des Auges möglich. Zwischen Retina und

[13] [Donders 1847]
[14] [Robinson 1963]

Cornea besteht eine Spannung von etwa 1 mV, die Messung des vom Auge erzeugten Feldes ermöglicht Aufschluß über seine Lage im Raum.

Dieses Verfahren wird routinemäßig in der klinischen Praxis zu Diagnosezwecken angewandt.

1.4.2.4 Purkinje–Tracker

Eine andere Möglichkeit zur Bestimmung der Augenposition besteht darin, die Reflektionen des Lichtes von Infrarot–Leuchtdioden auf den verschiedenen optischen Grenzflächen des Auges, die sogenannten Purkinje–Bilder, zu lokalisieren und aus ihrer Verteilung auf die Richtung des Auges rückzuschließen. Für die Arbeit mit Purkinje–Trackern ist naturgemäß ein recht detailliertes Modell der verschiedenen Grenzflächen und optischen Systeme im Auge erforderlich.

1.4.2.5 Video–Okulographie (VOG)

Erst durch jüngere Entwicklungen im Bereich digitaler Datenverarbeitung möglich geworden ist die Video–Okulographie (VOG), bei der Bilder des Auges mit Video-Geräten aufgenommen und die aufgezeichneten Bilder anschließend mit Computern ausgewertet werden.

Den Vorteilen der Methode—der Nichtinvasivität und der verhältnismäßigen Mobilität der Apparate—stehen große Nachteile im Hinblick auf die zeitliche und räumliche Auflösung des Verfahrens entgegen. Während die zeitliche Auflösung durch die Aufzeichnungsfrequenz (bei Videoaufnahmen in der Regel 50 Hz oder 60 Hz) und die endliche Belichtungszeit begrenzt wird, ist die räumliche Auflösung durch die Auflösungen entweder des verwendeten Videosystems oder der eingesetzten Framegrabber-Hardware begrenzt.

Da hier erst sehr teure Geräte akzeptable Werte erzielen, ist die VOG im Verhältnis sehr teuer. Daß sie dennoch in zunehmendem Maße im klinischen Bereich eingesetzt wird, verdankt sie dem entscheidenden Vorteil der leichteren Anwendbarkeit—und den stetig fallenden Preisen digitaler Ausrüstung bei gleichzeitiger Leistungssteigerung.

Außerhalb der medizinischen Forschung ist zudem die durch Bildverarbeitung gestützte Messung von Augenbewegungen die attraktivste Variante. Für die Blicksteuerung von Windows 2000 wird sich wohl niemand Kontaktlinsen mit Magnetspulen einsetzen wollen.

Abbildung 1.5: Meßbrille des SMI–VOG–Gerätes

1.5 Augenbewegungsmessung mittels Videookulographie

1.5.1 Apparative Mittel

Übersicht Für die Messung von Augenbewegungen mit einem Videookulographiesystem wird dem Probanden eine Video–Meßbrille aufgesetzt[15], an der die Kameras angebracht sind. Die von diesen erfaßten Bilder der Augen werden zur weiteren Bearbeitung auf einem Videorekorder aufgezeichnet und anschließend offline von einem PC ausgewertet.

Meßbrille Die Brille wird am Kopf des Probanden mit Bändern möglichst rutschfest fixiert (siehe Abbildung 1.5). Diese Fixierung ist besonders wichtig, da beim gegenwärtigen System Translationen und Rotationen der Brille *nach* der Kalibration nicht mehr berücksichtigt werden und einer Verschiebung des Bildes von 1 mm eine gemessene Augenbewegung von etwa 8 ° entspricht. An der Brille ist vor jedem Auge ein halbdurchlässiger Spiegel montiert. Die Augen werden durch Infrarot–Leuchtdioden, die sich zum Beispiel in der Fassung der Brille befinden können, beleuchtet. Das Infrarot–Bild des Auges wird nun über die Spiegel und eine Linse auf eine seitlich an der Brille befestigte CCD–Kamera abgebildet.

[15]Bei der hier beschriebenen und in den Abbildungen dargestellten Apparatur, deren Funktionsweise auch die Grundlage für die weiteren Untersuchungen innerhalb dieser Arbeit war, handelt es sich um ein VOG–System der Firma SensoMotoric Instruments (SMI) in Berlin.

Optische Abbildung Die Abbildung auf die Kamera bedeutet dabei eine Transformation der in Gleichung 1.7 beschriebenen dreidimensionalen Koordinaten auf eine zweidimensionale Bildebene, wie in Abbildung 1.6 dargestellt. Dabei werden Linsenfehler ebenso wie das Verlassen der Fokalebene bei Drehungen des Auges—das zu einem leicht unscharfen Bild bei großen Winkelauslenkungen führt—vernachlässigt. Es handelt sich dann um eine einfache Zentralprojektion, deren Projektionszentrum im Zentrum der Linse liegt.

Die x- und y-Koordinaten in der Bildebene werden aus den entsprechenden Koordinaten im kopffesten Koordinatensystem durch eine Skalierung berechnet, deren Maßstab abhängig von Gegenstandsweite und Bildweite ist. Die Spiegelung der Koordinaten wird durch eine Verdrehung der Kamera ausgeglichen, sodaß das Auge im Videobild richtig herum erscheint.

Diese Abbildung läßt sich durch

$$\vec{\tilde{x}}_{bild} = \begin{pmatrix} \mathcal{F}(z_{kopf}) & 0 & 0 \\ 0 & \mathcal{F}(z_{kopf}) & 0 \end{pmatrix} \vec{x}_{kopf} + \vec{\tilde{x}}_{ref} \tag{1.15}$$

mit $\mathcal{F}(z_{kopf})$ als Skalierungsfaktor darstellen[16].

Wenn g nun der Abstand des Kameralinse vom Ursprung des kopffesten Koordinatensystems und b der Abstand des Referenzpunktes im Videobild vom Projektionspunkt ist, so ist der Skalierungsfaktor $s = \frac{g-z}{b}$ (die Gegenstandsweite also $g - z_{kopf}$) und die gesamte Abbildung damit:

[16]Die Tilde bedeutet hier und später, daß die Koordinaten im *ungedrehten* Kamerabild gemeint sind.

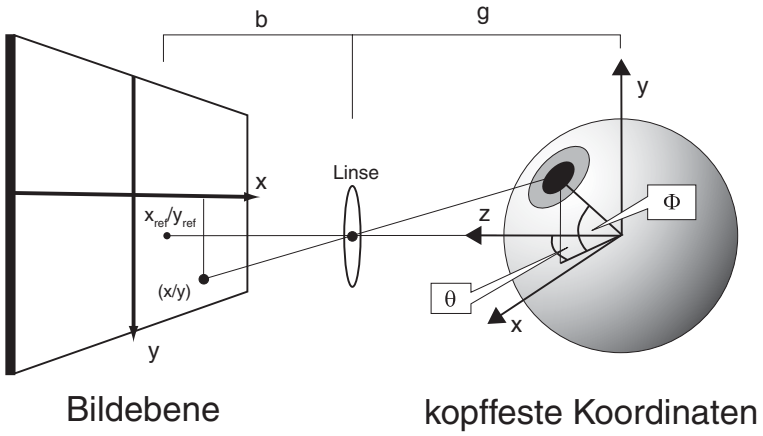

Abbildung 1.6: Abbildung auf die Kameraebene durch Zentralprojektion, aus Gründen der Übersicht ohne die Drehpunktverschiebung d

$$\vec{\tilde{x}}_{bild} = \begin{pmatrix} x_{kopf} \\ y_{kopf} \end{pmatrix} \frac{g - z_{kopf}}{b} + \begin{pmatrix} \tilde{x}_{ref} \\ \tilde{y}_{ref} \end{pmatrix} \tag{1.16}$$

Dies ergibt insbesondere für die Position des Pupillenmittelpunktes nach Gleichung 1.8 im Kamerabild:

$$\vec{\tilde{x}}_{pupille,bild} = \begin{pmatrix} \sin(\theta) R_{auge} \\ \sin(\phi)[\cos(\theta) R_{auge} - d] \end{pmatrix} \mathcal{F} + \vec{\tilde{x}}_{ref} \tag{1.17}$$

mit

$$\mathcal{F} = \frac{g - \cos(\phi)\cos(\theta) R_{auge} - d(1 - \cos(\phi))}{b} \tag{1.18}$$

Da der Verschiebungsparameter d um eine Größenordnung kleiner ist als der Augenradius, hängt die relative Änderung der Gegenstandsweite $g - z_{kopf}$ vor allem vom Verhältnis von g zur Änderung der z–Komponente von $\vec{x}_{kopf}$ ab. Beim hier betrachteten VOG–System hat die Linse eine Brennweite f von 30 mm, die Bildweite b beträgt 4 cm und damit ergibt sich nach der Gleichung für Gegenstands– und Bildweiten aus der geometrischen Optik

$$\frac{1}{f} = \frac{1}{b} + \frac{1}{g - R_{auge}} \tag{1.19}$$

die Gegenstandsweite $g - R_{auge}$, die wir erhalten, wenn das Auge in Referenzposition und das Pupillenzentrum im Kamerafokus sind, zu

$$g - R_{auge} = \frac{1}{\frac{1}{3\,cm} - \frac{1}{4\,cm}} = 12\,cm \tag{1.20}$$

Diese Gegenstandsweite schwankt bei gleichzeitiger Auslenkung von γ in Richtung der Winkel ϕ und θ größenordnungsmäßig um $\sin^2(\gamma)\,R_{auge}$, und damit für Auslenkungen von 10° um etwa $0.03\,R_{auge}$. Mit einem durchschnittlichen Augenradius von etwa 1,5 cm beträgt damit die relative Schwankung der Gegenstandsweite—und damit näherungsweise der durch Vernachlässigung dieser Schwankung entstehende Fehler[17]—etwa 0,3%. Für Auslenkungen von 20° beträgt die Schwankung etwa 1,4%.

Videoaufzeichnung Beim gegenwärtigen Stand der Digitaltechnik ist eine Online-Auswertung der aufgezeichneten Bilder unüblich, es werden daher meist die auf einem Videoband gespeicherten Bilder in mehreren Durchläufen ausgelesen und nach dem in den folgenden Abschnitten beschriebenen Verfahren analysiert.

[17] Für eine genauere Betrachtung des hier entstehenden Fehler verweise ich auf Abschnitt 2.7 über die geometrische Näherung und auf Abschnitt 5.2 zu seiner Behebung

Verdrehung der Kamerabrille Eine zusätzliche Komplikation der Abbildung entsteht dadurch, daß in der Regel die Kamerabrille nicht gerade auf dem Kopf des Probanden sitzt, wodurch die Achsen im Kamerabild nicht mit den Bildern der entsprechenden kopffesten Koordinatenachsen zur Deckung kommen. Zusätzlich zur bisher geschilderten Abbildung tritt also noch eine Drehung der Kameraebene um einen Winkel α und einen Drehpunkt $\vec{x}_{rot}$, die für positives α in mathematisch negativer Richtung erfolge.

Damit ist die Drehmatrix

$$\mathcal{S}_\alpha = \begin{pmatrix} \cos\alpha & -\sin(\alpha) \\ \sin\alpha & \cos(\alpha) \end{pmatrix} \tag{1.21}$$

und die gesamte Abbildung:

$$\begin{aligned} \vec{x}_{bild} &= \vec{x}_{rot} + \mathcal{S}_\alpha(\vec{\tilde{x}}_{bild} - \vec{x}_{rot}) && (1.22) \\ &= \mathcal{S}_\alpha \vec{\tilde{x}}_{bild} - [\mathcal{S}_\alpha \vec{x}_{rot} - \vec{x}_{rot}] && (1.23) \\ &= \mathcal{S}_\alpha \begin{pmatrix} x_{kopf} \\ y_{kopf} \end{pmatrix} \mathcal{F}(z_{kopf}) + [\mathcal{S}_\alpha(\vec{\tilde{x}}_{ref} - \vec{x}_{rot}) + \vec{x}_{rot}] && (1.24) \end{aligned}$$

Die eckige Klammer stellt hier den Ort des Blickreferenzpunktes im *gedrehten* Kamerabild und damit im von der Kamera aufgezeichneten Bild dar. Seine Koordinaten sind, im Unterschied zu den in der Klammer auftauchenden $\vec{\tilde{x}}_{ref}$ und $\vec{x}_{rot}$ einer Messung über den Kalibrationsalgorithmus zugänglich. Die vollständige Form der Abbildung lautet damit:

$$\vec{x}_{bild} = \mathcal{S}_\alpha \begin{pmatrix} x_{kopf} \\ y_{kopf} \end{pmatrix} \mathcal{F}(z_{kopf}) + \vec{x}_{ref} \tag{1.25}$$

Die komplette Abbildung aus den Helmholtz'schen Winkeln in Kamerakoordinaten sieht also schematisch folgendermaßen aus:

$$(\theta, \phi) \xrightarrow{Augen-Rot.} (\vec{x}_{kopf}) \xrightarrow{Kamera-Abb.} (\vec{\tilde{x}}_{bild}) \xrightarrow{Kamera-Rot.} (\vec{x}_{bild}) \tag{1.26}$$

1.5.2 Messung von 2-D-Augenbewegungen durch Pupillenfindung

Wir haben oben gesehen, wie die Bewegung des Auges im Kopf modelliert wird. Die Aufgabe für ein System zur Messung von Augenbewegungen besteht nun darin, die zwei Winkel θ und ϕ, die die Lage des Auges im Kopf beschreiben, aus den experimentell gewonnenen Bilddaten zu gewinnen.

Es liegt wie im vorigen Abschnitt dargestellt eine zweidimensionale Ansicht des Auges vor, in der nun über bestimmte Bildbearbeitungsverfahren[18] die Pupille aufgesucht wird. Nach dem Auffinden der Pixelkoordinaten des Zentrums der Pupille und einer ordnungsgemäßen Kalibration (vgl. Abschnitt 2.1) läßt sich prinzipiell über die Rücktransformation die horizontale und vertikale Augenposition errechnen.

Dazu muß zunächst aus den in der Messung erhaltenen Koordinaten $\vec{x}_{bild}$ die Position im ungedrehten Kamerabild errechnet werden. Dies erfolgt durch eine Rückrotation des Kamerabildes um den Winkel α und den als Drehpunkt verwendeten Referenzpunkt $\vec{x}_{ref}$. Da die Kamerarotation in mathematisch negativer Richtung gedacht wird, erfolgt die Rückrotation in mathematisch positiver Richtung.

Mit der Inversen von $\mathcal{S}_\alpha$

$$\mathcal{S}_\alpha^{-1} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{1.27}$$

schreibt sich die Rückdrehung dann in Übereinstimmung mit den Gleichungen 1.15 und 1.25

$$\begin{aligned} \vec{\tilde{x}}_{bild} - \vec{\tilde{x}}_{ref} &= \begin{pmatrix} x_{kopf} \\ y_{kopf} \end{pmatrix} \mathcal{F} && (1.28) \\ &= \mathcal{S}_\alpha^{-1}(\vec{x}_{bild} - \vec{x}_{ref}) && (1.29) \\ &= \begin{pmatrix} \cos(\alpha)(x_{bild} - x_{ref}) + \sin(\alpha)(y_{bild} - y_{ref}) \\ \cos(\alpha)(y_{bild} - y_{ref}) - \sin(\alpha)(x_{bild} - x_{ref}) \end{pmatrix} && (1.30) \end{aligned}$$

Dabei sind $\vec{x}_{bild}$ und $\vec{x}_{ref}$ direkt aus der Messung bewziehungsweise der Kalibration zugänglich, wir haben also einen direkten Zusammenhang zwischen diesen Größen und den Drehwinkeln ϕ und θ abgeleitet, der, setzen wir die Kopfkoordinaten aus Gleichung 1.17 hier ein, folgendermaßen aussieht:

$$\begin{pmatrix} \sin(\theta) R_{auge} \mathcal{F} \\ \sin(\phi)\cos(\theta) R_{auge} - d \end{pmatrix} \mathcal{F} = \mathcal{S}_\alpha^{-1}(\vec{x}_{bild} - \vec{x}_{ref}) \tag{1.31}$$

[18] Das von SMI eingesetzte Verfahren sucht im Kamerabild des Auges dunkle Ellipsen und ermittelt deren Abmessungen. Alternative Verfahren arbeiten mit Kreisen von variablem Durchmesser und Position, auf denen das Helligkeitsintegral gemessen wird. Die Kombination aus Position und Größe, für die das Integral den größten Sprung macht, wird als Rand der Pupille erkannt [Daugman 1994]. Eine weitere Methode nutzt richtungsabhängige Kantenfilter [Wildes 1996].

1.5.3 Geometrische Näherung bei der Messung

Auf unserem Weg zurück zur Winkeldarstellung der Augenposition kehren wir nun als nächstes diese optische Abbildung aus den kopffesten in die Kamerakoordinaten um. Im einzelnen ist diese Rücktransformation allerdings durch das komplizierte Aussehens des Skalierungsparameters $\mathcal{F}$ etwas unhandlich[19]. Im Augenblick wird daher in der Praxis die tatsächliche Transformation dadurch angenähert, daß man $\mathcal{F}$ konstant setzt und also davon ausgeht, daß das Pupillenzentrum die hintere Fokalebene der Linse während der Bewegung nicht verläßt (siehe Abbildung 1.7).

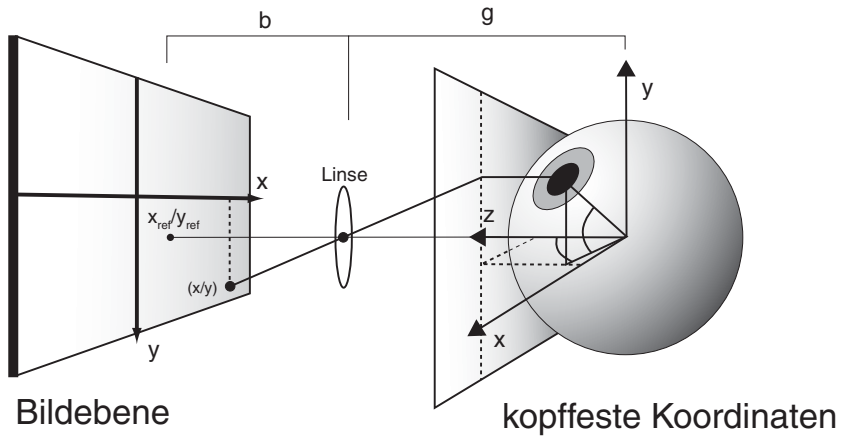

Abbildung 1.7: Die genäherte Abbildung aus den Kopf– in die Kamerakoordinaten. Die Näherung besteht dabei in einer Projektion des Punktes in Kopfkoordinaten auf die eingezeichnete Tangentialebene mit konstantem z. Die verbleibende Aufgabe für die Messung besteht dann im Schnitt der eingezeichneten Projektionsgerade durch Linse und Bildpunkt mit dieser Tangentialebene. Die Kinematik des Auges ist erneut ohne die axiale Verschiebung dargestellt.

Für den hierbei gemachten Fehler gilt größenordnungsmäßig das im vorigen Abschnitt über die Schwankung des Parameters $\mathcal{F}$ gesagte. Für eine numerische Berechnung des Fehlers verweise ich auf Anhang B.

Mit der genannten Näherung[20] ergibt sich

$$\mathcal{F} = \frac{g - R_{auge}}{b} \tag{1.32}$$

[19]Mathematisch handelt es sich um den Schnitt des Strahlweges, also einer Geraden, mit der Rotationsfigur, die der geometrische Ort der Pupillenmittelpunkte bei Rotation um θ und ϕ ist, einem Torus, vergleiche hierzu Abschnitt 5.2.

[20]Für den Vorschlag zu einem exakten Meßverfahren sowie die exakte Kinematik des Auges siehe Abschnitt 5.2.

und, da $\mathcal{F}$ nun nicht mehr von z_{kopf} und damit nicht mehr von θ und ϕ abhängt, die gesuchte Rücktransformation zu:

$$\theta = \sin^{-1}\left(\frac{\cos(\alpha)(x_{bild} - x_{ref}) + \sin(\alpha)(y_{bild} - y_{ref})}{\mathcal{F}R_{auge}}\right) \tag{1.33}$$

$$\phi = \sin^{-1}\left(\frac{\cos(\alpha)(y_{bild} - y_{ref}) - \sin(\alpha)(x_{bild} - x_{ref})}{\mathcal{F}(\cos(\theta)R_{auge} - d)}\right) \tag{1.34}$$

1.5.4 Messung der torsionellen Komponente über Segmentanalyse

Für die Messung der torsionellen Komponente der Augenbewegung zieht man bei der Video–Okulographie im Bild erkennbare Irismuskelstrukturen heran. In einem definierten Referenzbild werden ein oder mehrere Irissegmente ausgewählt und die Grauwerte entlang dieser um den Pupillenmittelpunkt zentrierten Segmente ausgelesen (siehe dazu Abbildung 1.8). Es werden also in Iriskoordinaten die N Einzelpunkte des Kreisbogens

$$\vec{x}_{seg} = \begin{pmatrix} r_{reg} \\ \varphi_0 + \frac{n}{N}\Delta\varphi \end{pmatrix} \tag{1.35}$$

mit

$$n \in [0, 1, 2, ...N] \tag{1.36}$$

aus dem Bild extrahiert. Dabei wird, um die Auflösung zu verbessern, zwischen den Pixelpositionen interpoliert, in dem aus je neun Pixelfarbwerten ein mit dem Abstand vom genauen Meßpunkt gewichteter Mittelwert berechnet wird.

Für die spätere Messung werden zunächst mit den vorgestellten 2–D–Methoden die horizontale und vertikale Augenposition bestimmt. Anschließend wird der geometrische Ort der zuvor ausgewählten Irissegmente im Bild berechnet und auch hier ein Grauwertmuster ausgelesen[21].

Die Verschiebung dieser Muster gegeneinander entspricht nun der torsionellen Drehung und kann wie folgt aus der Bildung der Kreuzkorrelationskurve der beiden Muster bestimmt werden.

[21] Die Berechnung der genauen Form des Musters im Kamerabild vor der Auslesung des Musters vermeidet einen Fehler in der Messung der Torsion von bis zu 5° bei bis zu 20° Auslenkung des Auges in horizontaler oder vertikaler Richtung. [Haslwanter-Moore 1995]

Die Muster liegen als digitale Samples R und M vor, die Formel zur Berechnung der Kreuzkorrelation $c_{R,M}(m)$ lautet für reellwertige Daten mit N Samples:

$$c_{R,M}(m) = \sum_{n=0}^{N-|m|-1} R(n) \cdot M(n + |m| + 1) \tag{1.37}$$

Diese Kreuzkorrelation hat ihr Maximum an der Stelle, die der Verschiebung der beiden Muster gegeneinander entspricht (Vergleiche Abbildung 1.9).

Die Auffindung dieses Maximums ist also gleichbedeutend mit der Messung der Torsion. Um die Auflösung der Maximumsfindung zu verbessern, wird in der Umgebung des Maximums eine Parabel an die Kreuzkorrelationskurve gefittet, und die Koordinaten ihres Scheitelpunktes mit dem Maximum identifiziert.

1.5.4.1 Komplikationen der Kreuzkorrelationsmethode

Erschwert wird die Messung der Torsion mit diesem Verfahren durch einige Komplikationen. Zunächst ist das Verfahren sehr empfindlich gegen Fehler in der 2-D-Messung. Ein auch nur um ein geringes falsch erkanntes Pupillenzentrum führt nicht nur zu inkorrekter Berechnung der Augenposition, sondern in der Folge auch zu einer Fehlberechnung der Position des Segmentes. Dadurch wird schlimmstenfalls ein ganz falsches Segment ausgelesen; die Bildung der Kreuzkorrelation kann dann kein sinnvolles Ergebnis liefern.[22]

Ein weiteres Problem ist, daß durch die torsionelle Verdrehung des Auges Irismusterteile in das Meßsegment wandern, die im Referenzmuster nicht enthalten waren. Dadurch entstehen in der Kreuzkorrelationsfunktion Artefakte, die mitunter dazu führen, daß eines der Nebenmaxima das Hauptmaximum übersteigt und die Messung zusammenbricht.

Schließlich werden Messungen der Torsion bei bewegten Augen durch die endliche Belichtungszeit erschwert. In Kapitel 3 wird die Auswirkung der Bewegungsverwischung auf die torsionale Messung genauer untersucht.

Zuletzt zeigt es sich, daß die Auswahl der Muster für die torsionale Messung, die bislang noch von Hand erfolgt, recht großen Einfluß auf die Qualität der mit diesen Mustern durchgeführten Messung hat. Daher wird in Kapitel 4 ein Verfahren entwickelt, daß die Auswahl des zur Messung herangezogenen Irissegmentes automatisiert.

Schlußbemerkung Damit ist die Darstellung der allgmeinen Gründe für die Okulographie und der speziellen Verfahren der Video–Okulographie been-

[22] [Haslwanter-Moore 1995]

det. In den folgenden Kapiteln werden drei Bereiche des VOG–Meßverfahrens genauer untersucht. In Kapitel 2 wird ein neuer Algorithmus zur Kalibration von VOG–Systemen vorgeschlagen, in Kapitel 3 wird die Auswirkung der endlichen Belichtungszeit auf die Meßgenauigkeit bei torsionellen Messungen untersucht. Kapitel 4 beschäftigt sich mit der Auffindung optimal für die torsionelle Messung geeigneter Segmente. In Kapitel 5 schließlich werden die Ergebnisse diskutiert und Folgerungen für die Praxis der VOG dargestellt.

Für jedes Kapitel finden sich die genaue Darstellung der Implementierung der erstellten Algorithmen sowie der Programmtext in Anhang A, die vollständige und genaue Darstellung der Ergebnisse in Anhang B. In den Kapiteln selbst wird über die Ergebnisse nur ein Überblick gegeben.

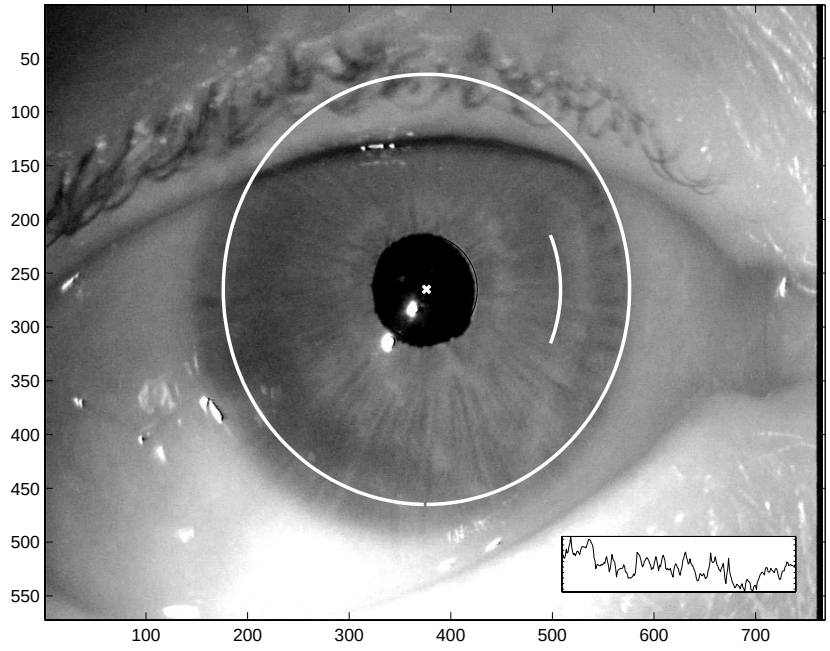

Abbildung 1.8: Die für die Simulationen in Kapitel 3 verwendete Frontalansicht eines Auges. Das Bild stammt direkt aus dem VOG-System der Firma SMI und stellt eine typische Aufnahme dar. Eingezeichnet sind kreisförmige Approximationen für den Pupillenrand und den Rand der Iris sowie ein Irissegment und das zugehörige Irismuster.

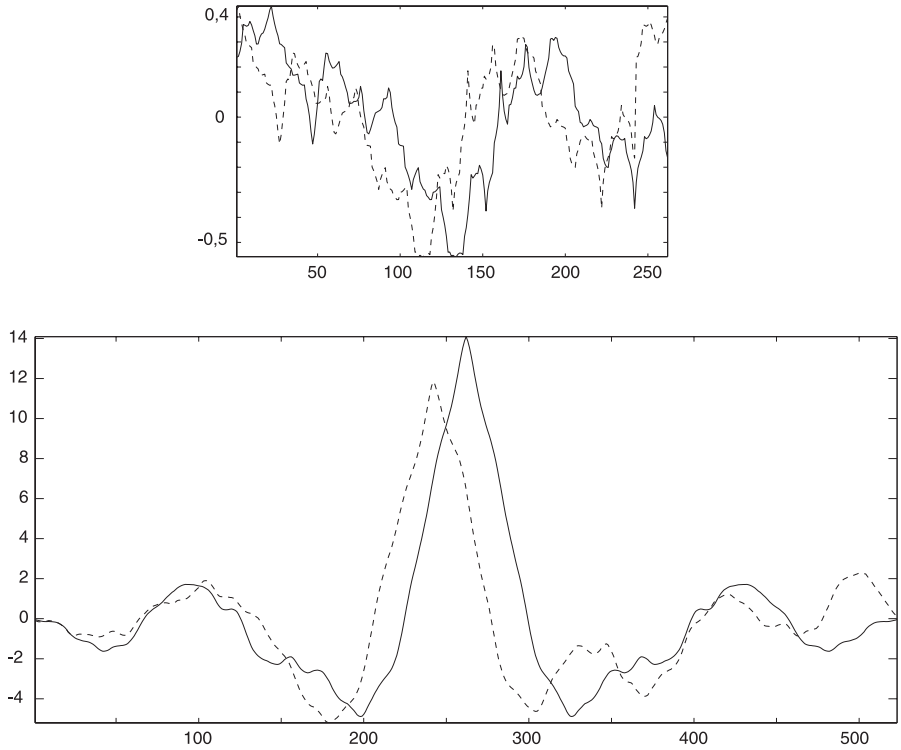

Abbildung 1.9: Schema der Messung der Musterverschiebung und damit der Torsion durch Bildung der Kreuzkorrelationskurve. Im oberen Rahmen ist durchgezogen das ursprüngliche Muster und durchbrochen das verschobene Muster dargestellt, im unteren ist durchgezogen die Autokorrelationskurve des ursprünglichen Musters und durchbrochen die Kreuzkorrelation der beiden Muster zu sehen. Gut zu erkennen ist die Verschiebung des Maximums der Kreuzkorrelation gegen das der Autokorrelation entsprechend der Verschiebung des Musters.

Kapitel 2

2–D–Kalibration des Systems

2.1 Problemstellung

Um bei der Auswertung der Messungen aus der Position der Pupille in der Aufnahme die tatsächliche Lage des Auges im Kopf berechnen zu können, muß das System vor jeder Messung zunächst kalibriert werden.

Dafür muß die Testperson eine Reihe von Punkten fixieren, die unter definierten Blickwinkeln präsentiert werden. Über die Pupillenfindung wird jedem dieser Fixationspunkte ein Ort in der Kameraebene zugeordnet, an dem das Pupillenzentrum erscheint, wenn das Auge in Richtung des Punktes blickt. Aus der Zuordnung der vorgegebenen Blickrichtungen zu den Pixelposition der Pupillenzentren werden dann die Kalibrationsparameter bestimmt.

Im Einzelnen sind die Parameter, die durch die Kalibration bestimmt werden müssen[1]:

1. Der Radius des Auges
2. Der Drehachsenabstand d[2]
3. Die Lage des Blickreferenzpunktes $\vec{x}_{ref}$ in der Kameraebene.
4. Die Verdrehung des Kamerabildes (Drehwinkel α)

Insgesamt stellt sich für die Kalibration also die Aufgabe, aus einer Reihe von Fixationspunkten, deren Positionen in Helmholtz–Winkeln gegeben sind, und dem Ort des Pupillenzentrums in der Kameraebene während der

[1] Diese Kalibration muß jedesmal neu durchgeführt werden, wenn die Meßbrille abgenommen, verschoben oder verdreht wurde, da dadurch die drei Parameter $\vec{x}_{ref}$ und α geändert werden

[2] Vgl. Abschnitt 1.2 über Augenbewegungsmodelle.

Fixation dieser Punkte den Satz der fünf Abbildungsparameter (2 Parameter für das Auge, zwei Pixelkoordinaten für $\vec{x}_{ref}$ und den Drehwinkel α) zu bestimmen.

2.2 Bisheriges Verfahren

Das zur Kalibration bislang beim VOG-Gerät der Firma SMI [SMI 1996], angewandte Verfahren enthielt dabei Näherungen, die teils zu systematisch fehlerhaften Ergebnissen führen konnten, in jedem Fall aber eine Lage der Fixationspunkte auf den Achsen zwingend erforderten.

Zunächst wurde dort die x– beziehungsweise y–Koordinate des Referenzpunktes durch Mittelung über die Koordinaten der Fixationspunkte auf der jeweils anderen Achse erhalten. Die Verdrehung der Kamera wurde anschließend über die Abweichung der Punkte von der Lage auf der senkrechten Achse berechnet.

Dieses Verfahren erzeugt aufgrund der Kameradrehung systematische Fehler, wenn die Punkte nicht symmetrisch zum Referenzpunkt gewählt werden. Außerdem ist die Einschränkung auf Achsenpunkte für die Kalibration unnötig und unerwünscht.

Daher wird im Folgenden ein neuer Algorithmus vorgeschlagen, der diese Einschränkungen nicht zeigt.

2.3 Neuer Algorithmus

Der vorgeschlagene Algorithmus funktioniert durch einen numerischen Fit im mehrdimensionalen Raum der Kalibrationsparameter. Es wird unter allen N–dimensionalen Punkten in diesem Raum derjenige ausgesucht, für den ein bestimmtes Funktional seinen minimalen Wert annimmt; dabei ist n die Anzahl der in diesem Schritt bestimmten Parameter, und somit in unserem Fall kleiner oder gleich der maximalen Parameterzahl, fünf.

Um die Stabilität des zu entwerfenden Kalibrationsalgorithmus' zu erhöhen, habe ich zunächst durch Wahl eines speziellen Kalibrationsverfahrens die Zahl der Parameter für den eigentlichen Fit von fünf Parametern auf vier reduziert. Dies geschieht durch Heranziehen der—gegen die Drehung der Kamera invarianten—Abstände der Fixationspunkte vom Referenzpunkt anstelle ihrer absoluten Koordinaten im Bild.

Die Daten, die zur Kalibration herangezogen werden, sind nun aufgrund obiger Beschreibung ein Satz von n Fixationspunkten mit bekannten, weil bei der Aufnahme der Daten vorgegebenen Winkelstellungen ϕ_i, θ_i und diesen zu-

geordneten gemessenen Pixelkoordinaten $\vec{x}_i$ (ermittelt durch Pupillenfindung auf der Videoaufzeichnung der Sequenz).

Für jeden Satz von Kalibrationsparametern $R_{auge}, d, \vec{x}_{ref}$ läßt sich aus den Winkelpositionen der für diese Parameter erwartete Satz von Pixelkoordinaten wie in Gleichung 1.17 berechnen[3] (gestrichene Größen bezeichnen die aus den Transformationen berechneten Bildkoordinaten und Abstände, ungestrichene die bei der Kalibration tatsächlich gemessenen).

$$\vec{\tilde{x}}_i' = \begin{pmatrix} \sin(\theta_i)R_{auge} \\ \sin(\phi_i)[\cos(\theta_i)R_{auge} - d] \end{pmatrix} \mathcal{F} + \begin{pmatrix} \tilde{x}_{ref} \\ \tilde{y}_{ref} \end{pmatrix} \tag{2.1}$$

mit wiederum

$$\mathcal{F} = \frac{g - \cos(\phi_i)\cos(\theta_i)R_{auge} - d(1 - \cos(\phi_i))}{b} \tag{2.2}$$

Diese Transformation berücksichtigt noch nicht die (zunächst unbekannte) Verdrehung der Kameras auf dem Kopf des Probanden. Daher wird nicht der—durch diese Verdrehung veränderte—absolute Wert der Pixelkoordinaten für die Kalibration herangezogen, sondern der Abstand der einzelnen Punkte vom Referenzpunkt, $\delta x_i'$.

$$\delta x_i'(R_{auge}, d) = \left| \begin{pmatrix} \sin(\theta_i)R_{auge} \\ \sin(\phi_i)[\cos(\theta_i)R_{auge} - d] \end{pmatrix} \mathcal{F} \right| \tag{2.3}$$

$$= \mathcal{F}\sqrt{\sin^2(\theta_i)R_{auge}^2 + \sin^2(\phi_i)[\cos(\theta_i)R_{auge} - d]^2} \tag{2.4}$$

$$= \mathcal{F}\mathcal{W} \tag{2.5}$$

mit

$$\mathcal{W} = \sqrt{\sin^2(\theta_i)R_{auge}^2 + \sin^2(\phi_i)[\cos(\theta_i)R_{auge} - d]^2} \tag{2.6}$$

Dieser Abstand läßt sich für die aus der Transformation erhaltenen Koordinaten berechnen und parallel dazu für die aus den Kalibrationsdaten gewonnenen. Das heißt, es wird für die gemessenen Punkte in der Kameraebene

[3] Dabei ist Vorsicht im Hinblick auf die Größe $\vec{x}_{ref}$ angebracht. Diese bezeichnet in der Kalibration den Ort des Referenzpunktes im *gedrehten* Kamerabild und ist keinesfalls mit dem in Gleichung 2.1 auftauchenden $\vec{\tilde{x}}_{ref} = (\tilde{x}_{ref}, \tilde{y}_{ref})$ zu verwechseln, das den Offset in der Kameraabbildung (also im ungedrehten Kamerabild) bezeichnet und für diesen Teil der Kalibration keinerlei Bedeutung hat.

$$\delta x_i(\vec{x}_{ref}) = |\vec{x}_i - \vec{x}_{ref}| \tag{2.7}$$

$$= \sqrt{(x_i - x_{ref})^2 + (y_i - y_{ref})^2} \tag{2.8}$$

berechnet. Für jeden 4–D–Kalibrationspunkt wird nun die Differenz dieser Abstände, Δ_i, und dann die Quadratsumme dieser Differenzen berechnet. Die Δ_i hängen dabei von allen vier Kalibrationsparametern ab, während die beiden Differenzen $\delta\vec{x}_i$ und $\delta\vec{x}'_i$ nur von jeweils zweien der Parameter abhängen.

$$\Delta_i = \Delta_i(R_{auge}, d, \vec{x}_{ref}) = \delta x'_i(R_{auge}, d) - \delta x_i(\vec{x}_{ref}) \tag{2.9}$$

$$\mathcal{S} = \mathcal{S}(R_{auge}, d, \vec{x}_{ref}) = \sum \Delta_i^2 \tag{2.10}$$

Dabei ist n die Anzahl der Fixationspunkte.

Die Suche nach den korrekten Kalibrationsparametern wird nun im vierdimensionalen Raum der Parameter als Suche nach dem Minimum der Abstandsquadratsumme $\mathcal{S}$ durchgeführt. Ist dieses Minimum gefunden, wird anschließend die Kamerarotation α berechnet.

2.4 Optimierung im vierdimensionalen Parameterraum

Für die eigentliche Kalibration wird ein numerischer Algorithmus herangezogen, der mit obiger Formel und den vorgegebenen Winkel- und den gemessenen Pixelkoordinaten die vier gesuchten Parameter durch Minimierung der Standardabweichung bestimmt. Es handelt sich dabei um ein Nelder–Mead Simplex–Verfahren [Nelder-Mead 1965], bei dem im N-dimensionalen Raum N+1-Punkte gewählt werden, die eine nicht degenerierte Figur (d.h. eine Figur mit endlichem Rauminhalt), das Simplex, aufspannen.

Als Startpunkt im vierdimensionalen Raum wählt man beim vorliegenden Problem die statistischen Mittel für den Augenradius und den Parameter d sowie die Winkelkoordinaten des Fixationspunktes mit dem kleinsten erwarteten Abstand vom Referenzpunkt bei diesen Werten für R_{auge} und d[4].

Die anderen N Eckpunkte des Simplex kann man nun beispielsweise erhalten, indem man den Startpunkt in die Richtungen der N Basisvektoren

[4] In der Regel wird dies der Punkt mit $\theta = 0$ und $\phi = 0$ sein

um einen charakteristischen Betrag verschiebt, etwa für R_{auge} den durchschnittlichen Radius, für d die doppelte physiologische Standardabweichung der Verschiebung der Verschiebung und für x_{ref} die Ausdehnung des aufgenommenen Bildes.

Es werden dann die Funktionswerte an den N+1 Punkten bestimmt und der Größe nach geordnet. Für die Iterationsschritte wird nun der Punkt mit dem größten Funktionswert durch die gegenüberliegende Fläche des Simplex bewegt, so daß das Volumen der Figur erhalten bleibt. Bleibt der Punkt dabei der Punkt mit dem größten Funktionswert, wird eine ein- oder mehrdimensionale Kontraktion des Simplex um den Punkt mit dem geringsten Funktionswert vorgenommen.

Der Algorithmus kann abgebrochen werden, wenn entweder die Vektordistanz, um die die Punkte in einem Schritt bewegt werden, unter eine vorgegebene Schwelle fällt, oder wenn die Abnahme der Funktionswerte an dem Punkt, der verändert wurde, unter eine solche Schwelle fällt.

Dieser Algorithmus kommt gänzlich ohne die Berechnung der Ableitungen der Funktion $\mathcal{S}$ aus. Dies vereinfacht zwar die Programmierung, führt aber dazu, daß das Konvergenzverhalten nicht optimal ist. Durch Verwendung eine höherwertigen numerischen Optimierungsverfahrens, etwa des Newton–Verfahrens kann daher die Stabilität des Algorithmus noch erhöht werden.

2.5 Bestimmung der Kamerarotation

Zur Bestimmung der Kamerarotation wird nach dem Ermitteln der ersten vier Parameter der Abstand der errechneten Punkte vom Referenzpunkt nach

$$\delta\vec{x}_i'(R_{auge}, d) = \begin{pmatrix} \delta x_i' \\ \delta y_i' \end{pmatrix} = \begin{pmatrix} \sin(\theta_i) R_{auge} \\ \sin(\phi_i)[\cos(\theta_i) R_{auge} - d] \end{pmatrix} \mathcal{F} \tag{2.11}$$

berechnet. Der gesuchte Rotationswinkel ergibt sich für jeden Fixationspunkt als Winkel zwischen diesem berechneten Abstandsvektor und dem tatsächlich im Bild gemessenen Abstand

$$\delta\vec{x}_i = \begin{pmatrix} \delta x_i \\ \delta y_i \end{pmatrix} = \begin{pmatrix} x_i - x_{ref} \\ y_i - y_{ref} \end{pmatrix} \tag{2.12}$$

und insgesamt als Mittelwert dieser Winkel zu

$$\alpha = \frac{1}{n} \sum \sin^{-1} \left(\frac{\delta x_i' \delta y_i - \delta x_i \delta y_i'}{\sqrt{(\delta x_i^2 + \delta y_i^2)(\delta x_i'^2 + \delta y_i'^2)}} \right) \tag{2.13}$$

2.6 Überprüfung der Stabilität

2.6.1 Analytische Überprüfung

Die analytische Untersuchung der Stabilität des so geschilderten Algorithmus ist zwar prinzipiell möglich, der dafür erforderliche mathematische Aufwand steht jedoch in keinem vernünftigen Verhältnis zum Nutzen. Daher wurde auf die analytische Berechnung verzichtet und die Stabilität stattdessen in Simulationen überprüft.

Für die weitere Bewertung des Algorithmus siehe 5.1.

2.6.2 Überprüfung durch Simulation

Um die Stabilität und Leistungsfähigkeit des so entworfenen Kalibrationsalgorithmus in der Praxis zu überprüfen, wurden Kalibrations–Simulationen durchgeführt und die Fehler des Algorithmus berechnet.

Dazu wurden für 1 bis 13 Fixationspunkte jeweils die fünf Kalibrationsparameter und die Winkelkoordinaten der Punkte ausgewählt, anschließend zu den Winkelkoordinaten ein statistischer Fixationsfehler addiert und aus den so gestörten Winkeln die Pixelkoordinaten der Punkte in der Kameraebene nach den Transformationsgleichungen berechnet. Schließlich wurden die ungestörten Winkelkoordinaten und die berechneten zugehörigen Pixelkoordinaten dem Kalibrationsalgorithmus übergeben.

Die vom Algorithmus zurückgelieferten Werte für die fünf Parameter wurden mit den ursprünglichen Parametern verglichen und die Abweichung von den Vorgabewerten berechnet.

Dieses Verfahren wurde für jede Anzahl von Punkten 20 mal durchgeführt, und die Fehler bei der Bestimmung der einzelnen Parameter über diese 20 Durchläufe durch Bildung der Wurzel der mittleren Quadrate der Abweichungen bestimmt.

Die Festlegung der Winkelkoordinaten der Fixationspunkte geschieht nach zwei verschiedenen Verfahren. Das erste wählt die Punkte in einem Raster mit 5–Grad–Schritten, das zweite verteilt die Punkte statistisch in der θ-ϕ–Ebene und vermeidet dabei Punkte, die sehr nahe am Referenzpunkt liegen, um die Bestimmung des Drehwinkels Alpha für alle Punkte zu ermöglichen.

An den Ergebnissen der Simulation ist deutlich erkennbar, daß der Algorithmus ab 6 Kalibrationspunkten hervorragend konvergiert, und auch für 4 und 5 Punkte brauchbare Ergebnisse liefern kann.

Der statistische Fehler bei den Fixationen wirkt sich—wie zu erwarten ist—hauptsächlich in den Parametern R_{auge} und d aus und erreicht bei schon

bei Abweichungen von einem Viertelgrad in beiden Winkeln die Größenordnung der interindividuellen Schwankung dieser Parameter.

Abhilfe kann hier einerseits durch Anpassung des Zielfunktionals für den numerischen Fit geschaffen werden, das heißt, indem bei Erreichen oder überschreiten der physiologischen Grenzen dieser Parameter ein zusätzlicher Term eingeführt wird, der das Zielfunktional stark anwachsen läßt.

Andererseits zeigte es sich in den Simulationen, daß der verwendete Nelder–Mead–Fit in Einzelfällen nicht optimal konvergiert. Dieses Verhalten tritt nur bei zufälliger Verteilung der Fixationspunkte auf, was bedeutet, daß eine geschickte Wahl dieser Punkte die Güte der Kalibration verbessern kann.

2.7 Auswirkungen der geometrischen Näherung

2.7.1 Verbesserung des Verfahrens?

Wie in Abschnitt 1.5.3 dargestellt wird beim VOG–System der Firma SMI bei der 2-D-Messung eine Ebenen-Näherung vorgenommen (siehe Abbildung 1.7 auf Seite 28), die einen systematischen Fehler bei der Messung der Augenposition zur Folge hat. Eine Möglichkeit, die Güte des Meßverfahrens zu verbessern, könnte darin bestehen, schon bei der Kalibration die später bei der Messung gemachte Näherung fester Bildweiten zu berücksichtigen, also schon für die Kalibration konstanten Abstand von den Pupillenmittelpunkten zur Kameralinse anzunehmen.

Der Algorithmus wird dann zur Berechnung der $\delta\vec{x}_i$ statt des korrekten $\mathcal{F} = (g - z_{kopf})/b$ die genäherte Form $\mathcal{F} = (g - R_{auge})/b$ verwenden und so zu modifizierten Kalibrationsparametern gelangen, die aber möglicherweise den durch die geometrische Näherung gemachten Fehler in der Messung minimieren.

2.7.2 Berechnung des Meßfehlers in Simulationen

Um diese These zu überprüfen wurden wieder Kalibrationsparameter zufällig gewählt, dann mit den oben beschriebenen Rasterfixationspunkten eine Kalibration ohne und mit Berücksichtigung der geometrischen Korrektur ausgeführt. Anschließend wurden Messungen über einen Bereich von 0-40 $^\circ$ in Richtung beider Winkel simuliert, der Fehler im erhaltenen Winkel berechnet und mit dem Fehler, der beim Standardmeßverfahren, also ohne die angepaßte Kalibration, auftritt, verglichen.

Es stellte sich jedoch heraus, daß das modifzierte Verfahren deutlich schlechtere Werte liefert als das Standardverfahren. Insbesondere liefert eine Messung mit den modifizierten Kalibrationsparametern für große Winkel mathematisch unsinnige Ergebnisse, weil das Argument des $\sin^{-1}$ größer als eins wird.

Es zeigt sich auch, das die oben angegebene Schätzung des Fehlers, der durch die Näherung entsteht, größenordnungsmäßig stimmt. Der tatsächliche Fehler bei einer Augenauslenkung von 20° beträgt absolut 0.42° und damit etwa 2.01% (siehe Abbildung 2.1). Da es sich allerdings hier um einen systematischen und hebbaren Fehler handelt, ist er dennoch nicht akzeptabel.

In Abschnitt 5.2 werden die korrekten Gleichungen und ein Meßalgorithmus angegeben, die ihn beseitigen.

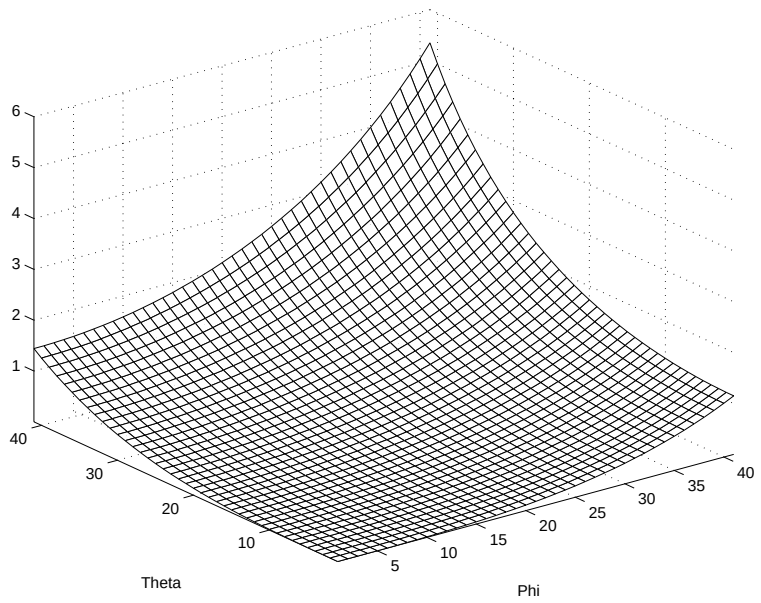

Abbildung 2.1: Betrag des Meßfehlervektors $\sqrt{\delta\phi^2 + \delta\theta^2}$ bei geometrischer Näherung. Für die getrennte Angabe der Fehler in den Winkeln ϕ und θ sowie Tabellen der Werte verweise ich auf Anhang B.

Kapitel 3

Simulation torsioneller Messungen und Bestimmung der maximalen Drehgeschwindigkeit bei 3-D-Messungen

3.1 Problemstellung

Eine der auffallendsten prinzipiellen Begrenzungen der VOG-Methode ist, wie im Einleitungskapitel bereits erwähnt, die Begrenzung der zeitlichen Auflösung durch die Aufnahmefrequenz des Videosystems und, damit zusammenhängend, die Begrenzung der Meßbarkeit der Torsion durch die meist recht lange Belichtungszeit für die einzelnen Aufnahmen und daraus resultierende Bildverwischungen.

Um die Größenordnung des sich ergebenden Problems abschätzen zu können, also Aussagen über die Grenz–Augengeschwindigkeiten machen zu können, bis zu denen torsionelle Messungen noch möglich erscheinen, und zu einer Einschätzung der Anforderungen an eine systematische Verbesserung zu gelangen, wurden simulierte Messungen an einem rechnerischen Modellauge durchgeführt.

3.2 Eignung eines Irismusters für die Torsionsmessung

Um später die errechneten, simulierten Irismuster in Bezug auf ihre Eignung zur Messung beurteilen zu können, benötigt man eine Größe, die mit dieser Eignung direkt korreliert.

3.2.1 Halbwertsbreite

Ein mögliches Maß für die Eignung eines Musters zur Messung ist die Halbwertsbreite des Maximums seiner Autokorrelationsfunktion. Je kleiner diese Halbwertsbreite ist, desto stabiler wird sich das Maximum der Kreuzkorrelationsfunktion bei Messungen auffinden lassen. Allerdings ist diese Halbwertsbreite aus den in Abschnitt 1.5.4.1 dargelegten Gründen nur ein grober Anhalt für diese Eignung des Musters.

3.2.2 Gütekurve und meßbare Winkelbreite

Daher schlagen wir statt dessen ein anderes Verfahren vor, das über Simulationen von Messungen mit dem untersuchten Muster den dabei auftretenden Meßfehler ermittelt und daraus ein Maß für die Eignung eines Musters gewinnt.

Das zu untersuchende Muster wird dazu in drei gleich große Teile zerlegt. Das mittlere Drittel wird im folgenden Referenzmuster genannt.

Nun wird für jede Startposition innerhalb der ersten zwei Drittel des gesamten Musters ein Teilmuster mit derselben Länge wie das Referenzmuster extrahiert. Diese Teilmuster werden mit dem Referenzmuster kreuzkorreliert und die Position des Maximums dieser Kreuzkorrelation bestimmt. Das entspricht einer Verschiebung des Gesamtmusters und der anschließenden Messung dieser Verschiebung mittels der Kreuzkorrelationsmethode (siehe Abbildung 3.1). Die gemessene Verschiebung wird über der Startposition der Teilmuster aufgetragen. Die entstehende Kurve nennen wir die *Gütekurve* des Musters.

Würde nun durch die Kreuzkorrelationsmethode an jeder Stelle die Verschiebung korrekt gemessen, müßte die Entstehende Gütekurve exakt der Gleichung $y = x - a$ gehorchen. Dabei ist a die Position des Referenzmusters (an der folgerichtig keine Verschiebung gemessen würde $y = x - a = a - a = 0$). Reale Gütekurven weichen im zentralen Teil mehr oder minder stark von dieser Geraden ab, zusätzlich kommt es am Rand zu plötzlichen Sprüngen. Für diese Sprünge sind anwachsende Nebenmaxima verantwortlich, die über

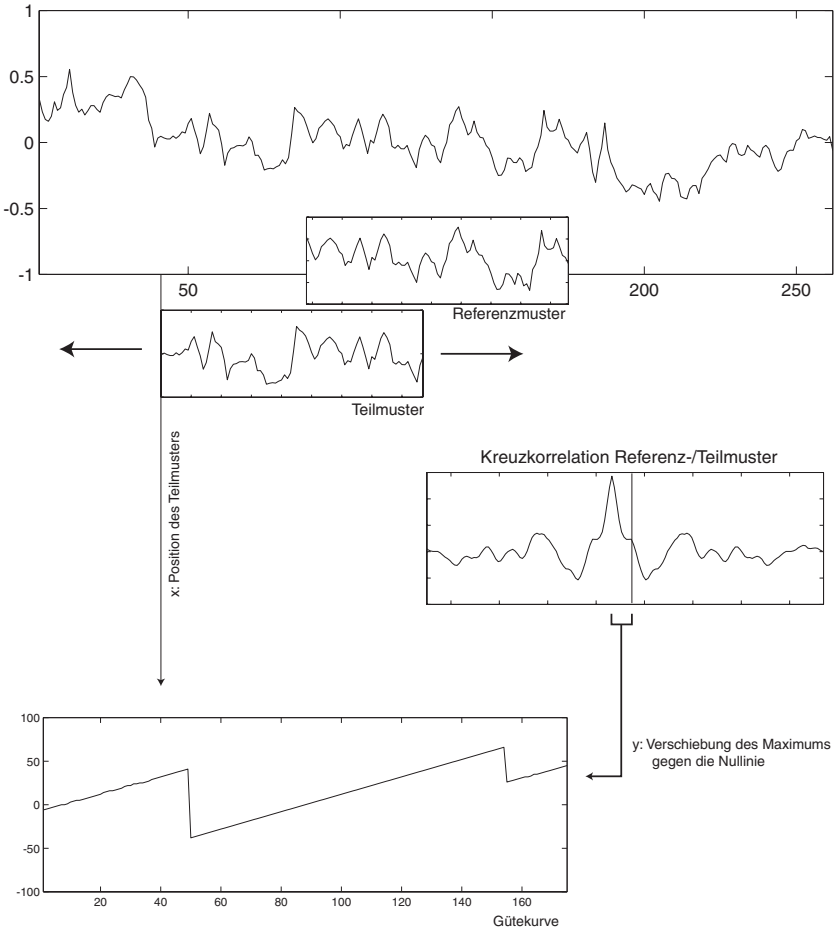

Abbildung 3.1: Schematische Darstellung des Verfahrens zur Berechnung der Gütekurve eines Irismusters. Es wird für jedes Musterdrittel die Kreuzkorrelation mit dem mittleren Drittel berechnet und die Position des Maximums aufgesucht. Aufgetragen ist die Abweichung der Maximumsposition von der Mitte der Kreuzkorrelationskurve über der Startposition des Musterdrittels. Die Breite der mittleren Schräge, auf der die Verschiebung der Kreuzkorrelation gleich der Abweichung der Startposition ist, ist die meßbare Winkelbreite.

das Hauptmaximum hinaus anwachsen.[1]

Als Maß für die Eignung eines Irismusters zur Messung wird nun die Breite des Bereiches in der Mitte der Gütekurve verwendet, in dem die Abweichung von der idealen Gerade unter einer gewissen Schwelle bleibt. Diesen Wert nennen wir die meßbare Winkelbreite des Musters.

3.3 Verwendetes Irismuster

Da zu erwarten ist, daß die Schmiereffekte der Bewegung sich bei Bewegungen senkrecht zum Muster von denen bei Bewegungen längs der Musterrichtung unterscheiden, müssen beide Varianten simuliert werden. Um andererseits die Ergebnisse dann vergleichen zu können, wird für beide Varianten dasselbe Irismuster verwandt. Wie in Abbildung 3.2 zu sehen, reicht es von -22.5 bis 22.5° in Iriskoordinaten, ist also 45° lang und um die Horizontale zentriert (siehe dazu auch Abbildung 1.8 auf Seite 32).

Die Drehung des Auges wurde dann in horizontaler (entspricht etwa der Drehrichtung senkrecht zum Muster) und vertikaler Richtung (entspricht etwa der Drehrichtung längs des Musters) simuliert.[2]

3.4 Das verwendete Abbildungsmodell

Für die Simulationsalgorithmen wird dabei vereinfachend eine Parallelprojektion aus den Kopfkoordinaten in die Kamerakoordinaten angenommen. Zusätzlich wird die Kamerarotation null gesetzt. Beide Faktoren würden bei korrekter Implementierung Übersichtlichkeit und Handhabbarkeit der Simulationsalgorithmen verringern, ohne etwas zum Verständnis der Problemes beizutragen. Zudem bedeuten die Vereinfachungen eine Verbesserung der Bedingungen, denn die tatsächliche Zentralprojektion durch die Kameralinse führt zu einer Verzerrung des Bildes, die das tatsächliche Ergebnis nur verschlechtern kann.

[1] Ihre tiefere Ursache haben diese Nebenmaxima in der Tatsache, daß bei der Verschiebung des Musters Teile der Iris ins untersuchte Muster wandern, die im Referenzmuster nicht vorhanden waren. Diese Teile können Teilen des Referenzmusters hinreichend ähneln, um zu Nebenmaxima der Kreuzkorrelationskurve zu führen, die das Hauptmaximum übersteigen. Dieser Effekt ist auch die hauptsächliche Ursache dafür, daß die Halbwertsbreite des Maximums der Autokorrelationsfunktion eines Musters nur bedingt für die Beurteilung der Eignung dieses Musters geeignet ist.

[2] Die sich ergebenden Effekte sind nicht mehr in Reinform die Effekte, die bei senkrechter und horizontaler Verschiebung aufträten, sondern bedingt durch die Kreisform der Muster schon eine Mischform. Da dies jedoch auch in der Praxis so sein wird, ist die exakt parallele oder horizontale Verschmierung wenig interessant.

Die mathematischen Grundlagen für die Abbildungen sind im Einführungsteil bereits dargestellt worden, aus Gleichung 1.7 entnimmt man die Kopf–Koordinaten eines allgemeinen Punktes auf der Iris, die sich bei Parallelprojektion direkt in die Kameraebene übertragen. Damit sind die Kamerakoordinaten eines allgemeinen Punktes auf der Iris:

$$\begin{pmatrix} x_{bild} \\ y_{bild} \end{pmatrix} = \begin{pmatrix} \cos(\theta)\cos(\varphi_{iris})r_{iris} + sin(\theta)R_{auge} \\ \cos(\phi)\sin(\varphi_{iris})r_{iris} - sin(\phi)\mathcal{Z} \end{pmatrix} + \begin{pmatrix} x_{ref} \\ y_{ref} \end{pmatrix} \tag{3.1}$$

mit

$$\mathcal{Z} = -\sin(\theta)\cos(\varphi_{iris})r_{iris} + \cos(\theta)R_{auge} - d \tag{3.2}$$

Aus diesen Gleichungen lassen sich sehr leicht umgekehrt die Polarkoordinaten auf der Iris aus den Kopfkoordinaten durch sukzessives Einsetzen berechnen:

Zunächst wird die erste Zeile von Gleichung 3.1 nach $\cos(\varphi_{iris})r_{iris}$ aufgelöst:

$$\cos(\varphi_{iris})r_{iris} = \frac{x_{bild} - x_{ref} - \sin(\theta)R_{auge}}{\cos(\theta)}$$

Mit

$$\mathcal{Z} = -\sin(\theta)\cos(\varphi_{iris})r_{iris} + \cos(\theta)R_{auge} - d$$

läßt sich dann die zweite Zeile von Gleichung 3.1 nach $\sin(\varphi_{iris})r_{iris}$ auflösen:

$$\sin(\varphi_{iris})r_{iris} = \frac{y_{bild} - y_{ref} + \sin(\phi)\mathcal{Z}}{\cos(\phi)}$$

Durch Bildung von $\tan(\varphi_{iris})$ erhalten wir dann:

$$\tan(\varphi_{iris}) = \frac{\cos(\varphi_{iris})r_{iris}}{\sin(\varphi_{iris})r_{iris}}$$

und damit für φ_{iris}:

$$\varphi_{iris} = tan^{-1}\left(\frac{\cos(\phi)[x_{bild} - x_{ref} - \sin(\theta)R_{auge}]}{\cos(\theta)[y_{bild} - y_{ref} + \sin(\phi)\mathcal{Z}]}\right)$$

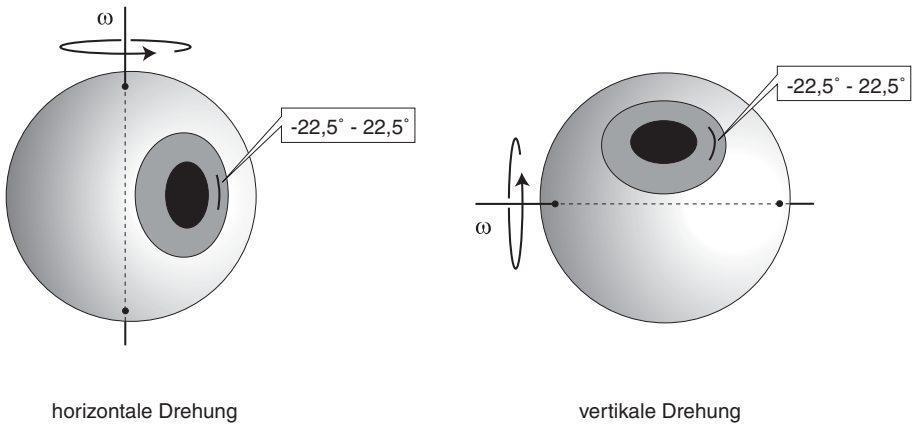

Abbildung 3.2: Segment und Drehrichtungen für die Drehsimulation

und schließlich mit diesem φ_{iris}:

$$\begin{aligned} r_{iris} &= \frac{sin(\varphi_{iris})r_{iris}}{sin(\varphi_{iris})} \\ &= \frac{y_{bild} - y_{ref} + \sin(\phi)\mathcal{Z}}{\sin(\varphi_{iris})\cos(\phi)} \end{aligned}$$

3.5 Simulationsablauf

Die Simulationen werden nun so durchgeführt, daß zunächst das Referenzmuster ohne Bewegungssimulation extrahiert wird. Dazu wird aus der Frontalaufnahme eines Auges entlang des gewählten Kreissegmentes eine Grauwertkurve ausgelesen[3].

Anschließend wird der Ort des Segmentes in der Kameraebene zum Zeitpunkt der Mitte der Belichtungszeit berechnet. Dies wird im Allgemeinen kein Kreisbogen sondern die Parallelprojektion des Segmentes vom Auge in die Kameraebene sein. An dieser Linie in der Kameraebene wird nun zu n verschiedenen Zeitpunkten—und damit mit n verschiedenen Stellungen des Auges, das sich ja währenddessen dreht—das Bild der Iris ausgelesen, die erhaltenen Muster werden punktweise addiert und gemittelt.

Diese Simulationen werden für horizontale Drehgeschwindigkeiten von 0°/Sekunde bis 300°/Sekunde in Schritten von 10°/Sekunde und für zwei

[3]In der VOG–Praxis wird dies nach der die Auswahl eines Segmentes auf einem Videobild des unbewegten, frontal erfaßten Auges geschehen.

verschiedene Drehrichtungen (vergleiche Abbildung 3.2) durchgeführt. Daraus erhalten wir für jede Bewegungsrichtung die verschmierten, am bewegten Auge abgelesenen Muster.

3.6 Güte der bewegungsunscharfen Muster

Mit der meßbaren Winkelbreite haben wir das benötigte Werkzeug an der Hand, um die Qualität der so erhaltenen bewegungsunscharfen Irismuster zu beurteilen. Für diese Bewertung erhalten wir allerdings die zusätzliche Komplikation, daß im Falle einer realen Messung das Referenzmuster am unbewegten Auge extrahiert würde, während das zur Messung herangezogene Muster bewegungsunscharf wäre. Daher verwenden wir zur Beurteilung der Mustereignung für die Berechnung der Gütekurve statt des mittleren Drittels des verschmierten Musters das Muster auf dem mittlere Drittel desselben Irissegmentes am unbewegten Auge, also das Referenzmuster ohne Verschmierung.

3.7 Ergebnisse

Die Simulation der Drehungen wurde mit zwei verschiedenen Musterradien durchgeführt, um beurteilen zu können, wie stark die Meßbarkeit der Torsion vom gewählten Muster abhängt. Dazu wurde einerseits ein Muster auf einem Radius von 80 Pixeln verwendet. Dieses Muster hatte in Bezug auf die Torsionsmessung mittlere Qualität.

Außerdem wurde im untersuchten Segment dasjenige Muster gesucht, dessen Autokorrelationsfunktion die geringste Halbwertsbreite des Maximums aufwies. Dies war der Fall für das Muster am Radius 132 Pixel. Auch für dieses Muster wurden die Simulationsrechnungen durchgeführt.

3.7.1 Qualitative Beurteilung

Die Effekte der Bewegungsunschärfe lassen sich gut erkennen, wenn man die bei verschiedenen Augengeschwindigkeiten berechneten Muster nebeneinanderstellt (Abbildungen 3.3 und 3.4). Deutlich zu erkennen ist, daß die Effekte bei den beiden verschiedenen Bewegungsrichtungen recht unterschiedlich ausfallen, ebenfalls gut zu sehen ist, daß nahezu alle charakteristischen Eigenschaften des untersuchten Musters schon bei relativ geringen Winkelgeschwindigkeiten verschwinden. Zu sehen ist das besser geeignete Muster bei einem Radius von 132 Pixeln.

Fehlerschwelle	Muster 1 (80 Pixel)	Muster 2 (132 Pixel)
0.085°	30/30	70/70
0.17°	50/50	120/70
0.34°	50/50	190/70

Tabelle 3.1: Tabelle der Grenzgeschwindigkeiten in Grad pro Sekunde, bei denen der Fehler in der Torsionsmessung bei Nulltorsion bereits über der Fehlerschwelle liegt. Die erste Zahl ist die Grenzgeschwindigkeit für Bewegungen senkrecht zur Musterrichtung, die zweite für Bewegungen parallel zur Musterrichtung.

Die Mechanismen, die zur Verschmierung der Muster führen, lassen sich an den Abbildungen 3.5 und 3.6 gut erkennen. Aufgezeichnet sind die Bereiche der Iris, die unter dem ausgelesenen Teil des Bildes während der Belichtungszeit hindurchwandern.

3.7.2 Quantifizierung

Wie nach Studium der Verschmierung kaum anders zu erwarten, fallen die Ergebnisse der Simulationen für die Praxis nicht besonders erfreulich aus.

Es zeigt sich, daß die maximale Drehgeschwindigkeit, bei der Messungen der Torsion noch möglich sind, recht stark von der Qualität des anfänglich ausgesuchten Musters abhängt. Während bei dem Muster mittlerer Qualität schon ab etwa 50°/Sekunde Drehgeschwindigkeit keine fehlerfreie Messung mehr möglich ist, läßt sich mit dem Muster geringer Halbwertsbreite bei etwa der doppelten Geschwindigkeit noch sauber messen. Insgesamt aber liegen bei beiden Mustern die Fehler in der Torsionsmessung, die allein aufgrund der Bewegungsunschärfe auftreten, bei üblichen physiologischen Geschwindigkeiten erheblich zu hoch.

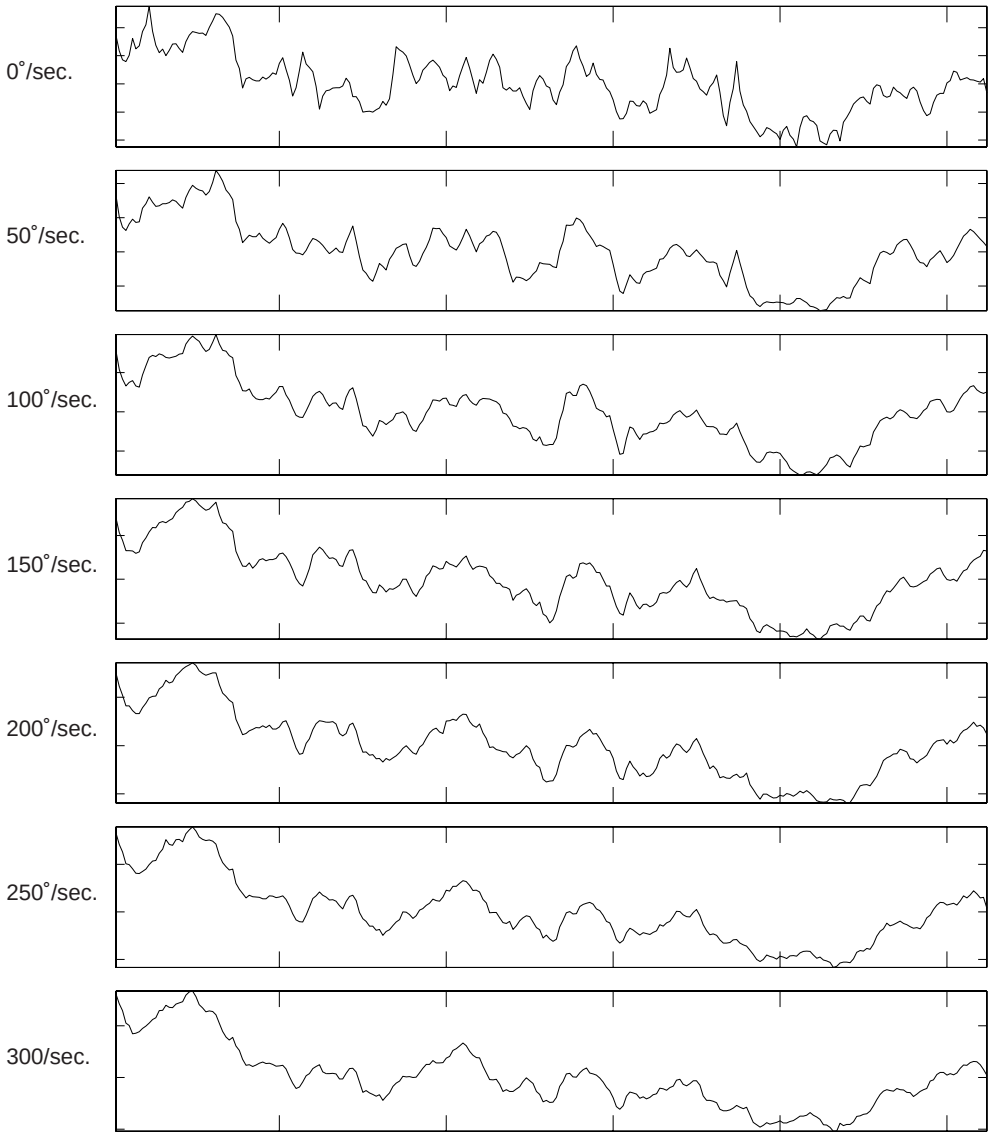

Abbildung 3.3: Veränderung des Musters bei Bewegungen senkrecht zur Musterrichtung. Ganz oben das unbewegte Auge, darunter 50, 100, 150, 200, 250 und 300°/Sekunde Drehgeschwindigkeit. Zu sehen ist das Muster mit einem Radius von 132 Pixeln

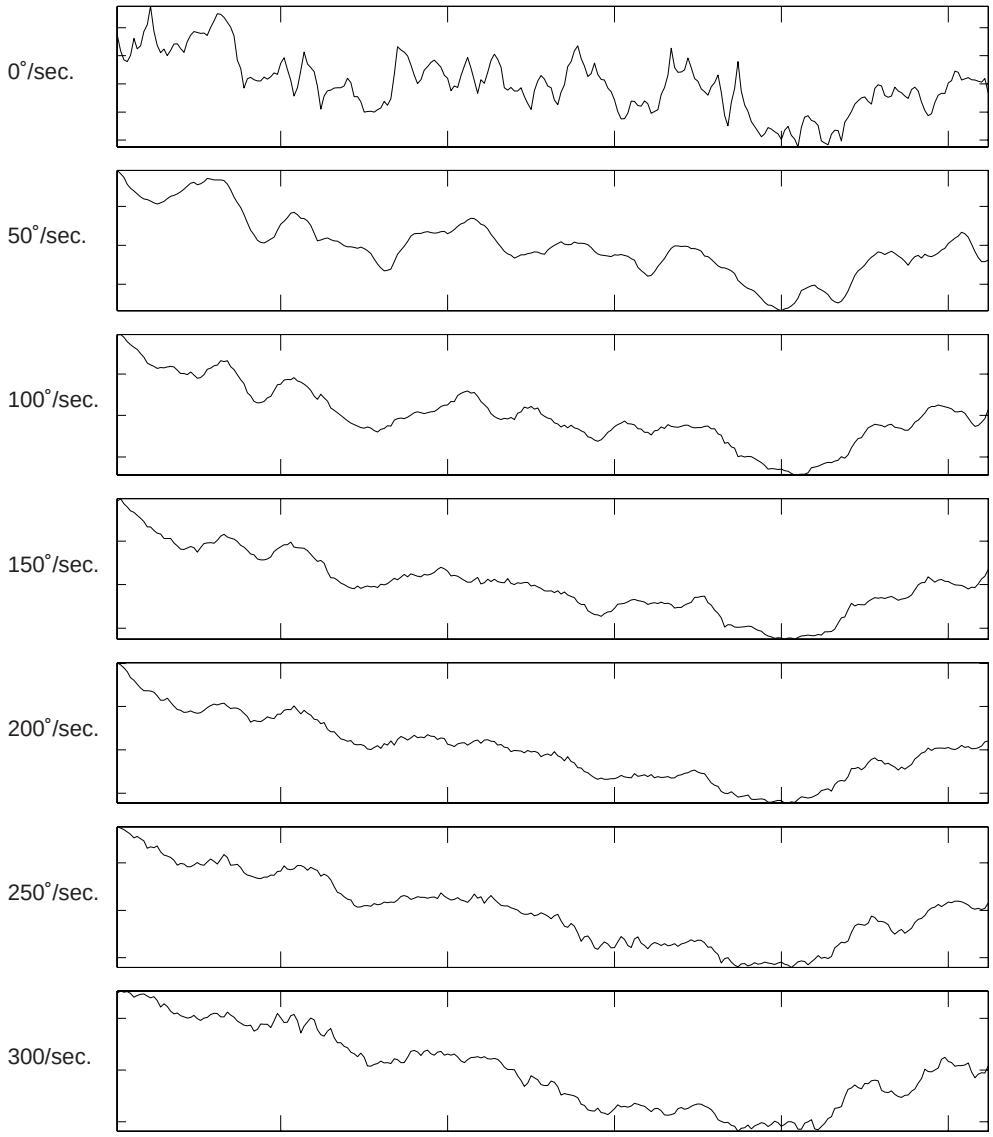

Abbildung 3.4: Veränderung des Musters bei Bewegungen parallel zur Musterrichtung. Ganz oben das unbewegte Auge, darunter 50, 100, 150, 200, 250 und 300°/Sekunde Drehgeschwindigkeit. Zu sehen ist das Muster mit einem Radius von 132 Pixeln

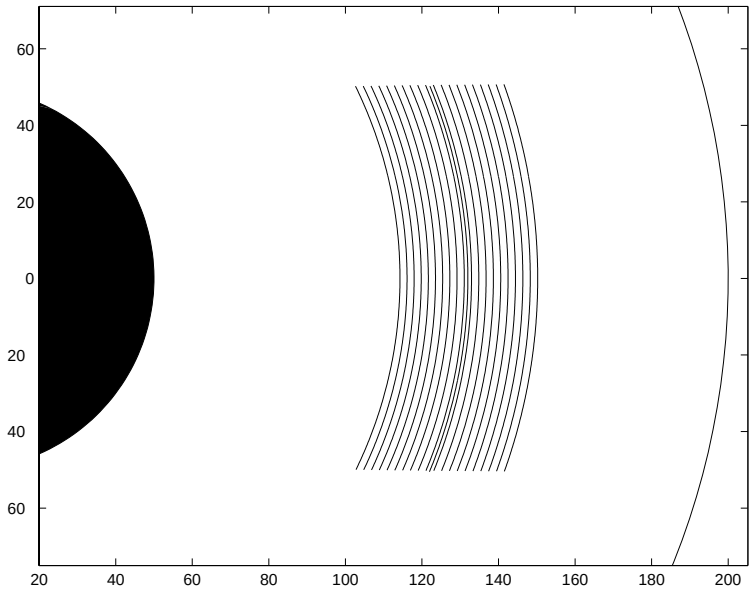

Abbildung 3.5: Bereich der Iris, der bei einer Augengeschwindigkeit von 300°/Sekunde in horizontaler Richtung zu einer Aufnahme des Irismusters an der Stelle des mittleren Bogens beiträgt. Die beiden Kreise sind der Rand der Pupille (links) und der Rand der Iris (rechts) wie in Abbildung 1.8 auf Seite 32.

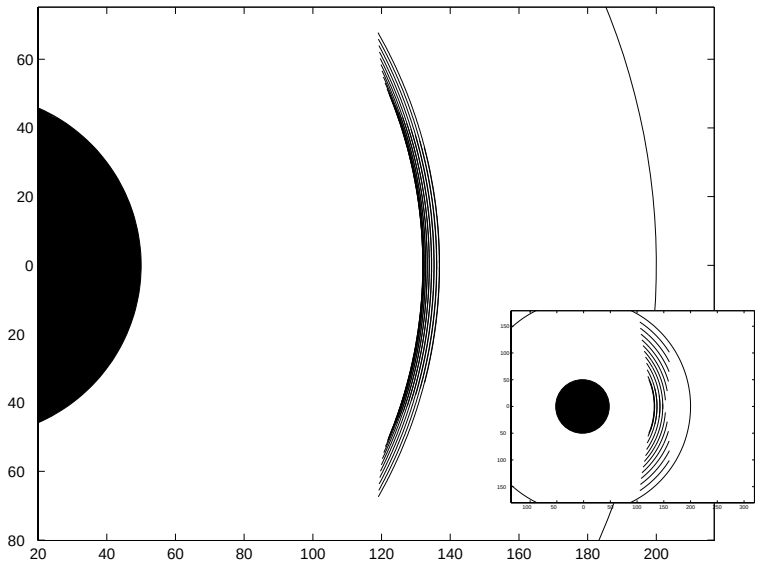

Abbildung 3.6: Bereich der Iris, der bei einer Augengeschwindigkeit von °/Sekunde in vertikaler Richtung zur Aufnahme des Irismusters beiträgt. Wie gut zu sehen ist, tragen fast nur die Muster eines bestimmten Radius zur Aufnahme bei, allerdings wird das Muster bei der Aufnahme in tangentialer Richtung stark verschmiert. Zur besseren Verdeutlichung dieses Effektes sind in dem kleinen Kasten die Muster für eine Winkelgeschwindigkeit von 2000°/Sekunde aufgezeichnet.

Kapitel 4

Bestimmung des optimalen Segmentes zur torsionellen Messung

4.1 Problemstellung

Wie im vorigen Kapitel deutlich zu sehen war, hängt die Stabilität der Messung der torsionellen Komponente der Augenbewegungen bei höheren horizontalen und vertikalen Augenbewegungsgeschwindigkeiten deutlich von den Eigenschaften des verwendeten Irissegmentes ab.

Außerdem ist auch die Stabilität der torsionellen Messung bei unbewegtem Auge davon abhängig, ob das verwendete Muster gut ausgewählt ist, das heißt, ob seine Eigenschaften der Kreuzkorrelations-Meßmethode entgegenkommen.

Für die Anwenderfreundlichkeit der VOG wie für die Sicherheit, bei einer durchgeführten Messung optimale Meßdaten zu erhalten, wäre es sehr wünschenswert, wenn die Beurteilung der Muster bezüglich ihrer Verwendbarkeit zur Messung durch einen möglichst effektiven und gleichzeitig ausreichend schnellen Algorithmus erfolgen könnte, der die zur Messung heranzuziehenden Segmente automatisiert auswählt.

Wir haben den Versuch gemacht, einen solchen Algorithmus zu entwickeln, indem wir zunächst versuchten, anzugeben, welche Kriterien ein Muster erfüllen muß, um sich für die Messung der Torsion gut zu eignen.

4.2 Musterkriterien

Der zunächst naheliegende Gedanke, die bereits eingeführte meßbare Winkelbreite hierfür heranzuziehen, stößt auf die Schwierigkeit, daß die Berechnung der Gütekurve für ein einzelnes Muster zu lange dauert. Prinzipiell ist es mit diesem Kriterium zwar möglich, gut geeignete Muster aufzufinden, für die Praxis aber ist dieses Verfahren bei gegenwärtiger Hardwareausstattung zu langsam.[1]

Ein weiterer naheliegender Gedanke ist, die Halbwertsbreite des Hauptmaximums der Autokorrelation des Musters zu verwenden. In der bisherigen Ausführung des VOG-Systemes wird die Auswahl des Musters vom Anwender vorgenommen. Zur Unterstützung wird nach erfolgter Auswahl die Autokorrelationskurve des gewählten Musters dargestellt, und wie sich in der Praxis zeigt, wählen die meisten Anwender intuitiv Muster, deren Autokorrelations-Hauptmaximum schmal ist, und deren Nebenmaxima nur schwach ausgeprägt erscheinen. Diese intuitive Auswahl entbehrt allerdings bislang einer Begründung.

Zusätzliche Komplikationen ergeben sich durch die Tatsache, daß bei einer Messung Irisbereiche eine Rolle spielen, die in die Berechnung der Autokorrelationsfunktion nicht einfließen, sodaß in jedem Falle eine geringe Halbwertsbreite dieser Funktion nicht das einzige Kriterium sein kann — wenn sie sich überhaupt als solches eignet.

Falls, wie vermutet werden mußte, die Halbwertsbreite sich nicht als Kriterium eignen sollte, hofften wir, Angaben über Raumfrequenzverteilungen machen zu können, denen die Frequenzverteilung eines guten Musters ähneln muß, um dann über Wavelet-Verfahren ein schnelles räumliches Auffinden dieser Muster zu bewerkstelligen.

Wünschenswert wäre weiterhin, daß kleine Fehler in der Messung der 2-D-Augenposition, die zum Auslesen eines auf der Iris leicht verschobenen Segmentes führen können, die Messung nicht unmöglich machen, das heißt, daß die Iris an der Stelle eines guten Musters zusätzlich in radialer Richtung möglichst wenig variieren sollte. Dies würde auch die in Kapitel 3 untersuchte Fehler aufgrund der endlichen Belichtungszeit verringern.

[1] Auf einem 233 MHz-K6-System dauerte die Analyse eines Kreissegmentes der Breite 45° in Matlab etwa vierzig Minuten. Selbst bei maschinennaher Programmierung und schnellerer Hardware wird die Komplettanalyse eines Bildes recht lange — zu lange — dauern.

4.3 Die Kriterien im Einzelnen

Da mit der meßbaren Winkelbreite bereits ein verläßliches Kriterium für die Musterqualität zur Hand ist, fällt die Entscheidung über die Relevanz vorgeschlagener Kriterien recht leicht. Es ist die Beurteilung eines Musters, die ein bestimmtes Kriterium ergibt, mit dessen Beurteilung aufgrund der meßbaren Winkelbreite zu vergleichen.

4.3.1 Halbwertsbreite des Maximums

Um den Zusammenhang zwischen der Halbwertsbreite des Maximums der Autokorrelationsfunktion eines Musters (HWBM) und seiner meßbaren Winkelbreite (MWB) zu ermitteln, wurden beide Größen für sehr viele Muster berechnet und miteinander in Beziehung gesetzt.

Auf zwei Segmenten der Iris, dem einen von -22,5° bis 22,5°, dem anderen von 157,5° bis 202,5° wurden in Radius-Schritten von 0.25 Pixeln über die gesamte radiale Breite der Iris Muster ausgelesen, für jedes MWB und HWBM berechnet und beiden Werte gegeneinander in ein Diagramm eingetragen. Das Ergebnis in Abbildung 4.1 zeigt, daß der Zusammenhang zwischen HWBM und MWB nur schwach ausgeprägt ist. Zwar ist die Tendenz erkennbar, daß Muster mit großer HWBM sich schlechter für Messungen eignen, aber bei kleinere HWBM ist lediglich eine Verschiebung der Verteilung hin zu größeren meßbaren Winkelbreiten zu beobachten. Ein eindeutiger Zusammenhang, der kleine HWBM mit guter Verwendbarkeit (sprich großer meßbarer Winkelbreite) verknüpft, ließ sich nicht nachweisen. Wie an den Abbildungen 4.1 und 4.2 jedoch erkennbar, läßt sich ein abgeschwächter Zusammenhang vermuten,

4.3.2 Frequenzverteilung

Noch weniger erfolgreich waren die Versuche, über die Frequenzverteilung geeigneter Muster zu Aussagen zu gelangen. Die Anwendung digitaler Butterworth-Filter auf ein ausgewähltes Muster und die anschließende Bestimmung der meßbaren Winkelbreite des gefilterten Musters ergab keinen erkennbaren Zusammenhang zwischen den Filtergrenzen und der resultierenden meßbaren Winkelbreite. Weitere Untersuchungen in dieser Richtung wurden wegen der wenig erfolgversprechenden Ergebnisse nicht durchgeführt.

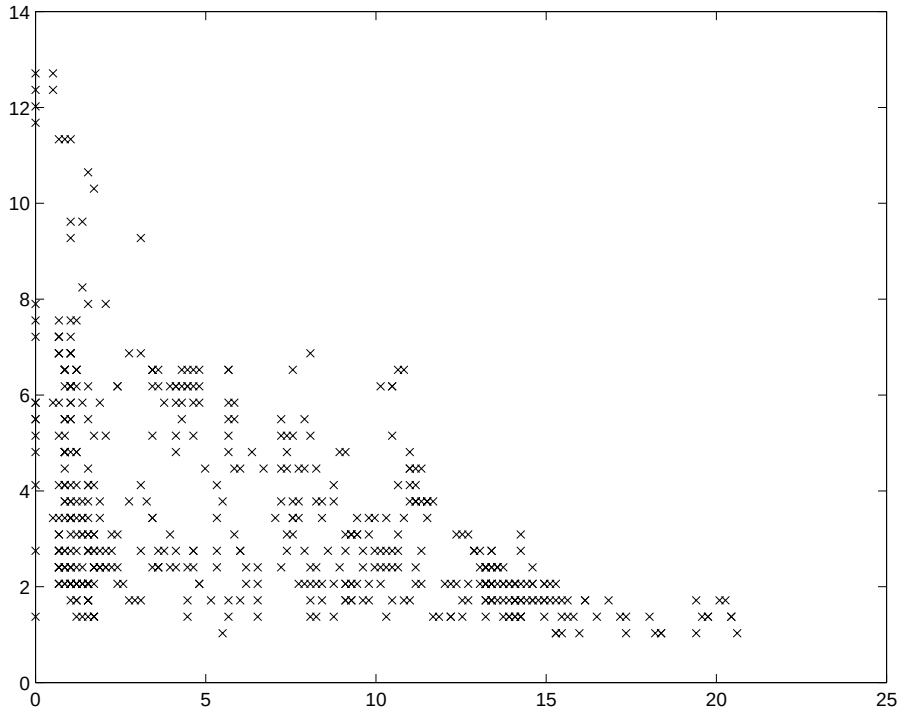

Abbildung 4.1: Die Halbwertsbreite der Autokorrelationsfunktion der Irismuster mit variablem Radius auf dem Segment von -22,5° bis 22,5° über ihrer meßbaren Winkelbreite. Einheiten sind bei beiden Größen Grad. Die meßbare Winkelbreite wurde aus der Gütekurve mit einer Fehlerschranke von 0.5 Pixeln berechnet.

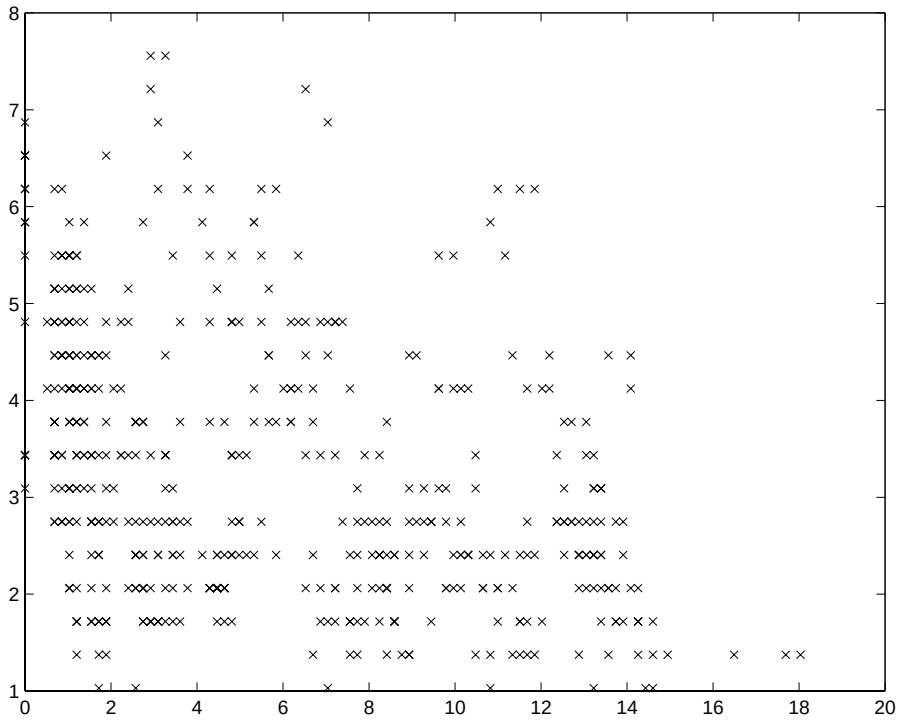

Abbildung 4.2: Dasselbe wie in Abbildung 4.1, mit Mustern eines anderen Irissegmentes.

4.3.3 Qualitative Analyse

Auch der Versuch, die Muster noch ihrer meßbaren Winkelbreite zu ordnen, um auf diesem Wege einen Eindruck davon zu bekommen, welche Kriterien eine gute Verwendbarkeit des Musters in der Messung garantieren, führte zu keinem Ergebnis.

Kapitel 5

Praktische Folgerungen aus den Ergebnissen

5.1 Kalibration des Systems

5.1.1 Numerischer Algorithmus

Der zur Durchführung der Kalibration in dieser Arbeit herangezogene numerische Nelder–Mead–Algorithmus erfüllt, wie aus den Ergebnissen der Simulationsläufe deutlich geworden ist, seinen Zweck recht gut. Allerdings zeigte sich auch, daß das Verfahren bei ungünstiger Wahl der Fixationspunkte Fehler erzeugt. Es ist daher für eine Anwendung des beschriebenen Kalibrationsverfahrens in der Praxis unumgänglich, den Nelder–Mead–Fit durch ein höherwertiges numerisches Verfahren, beispielsweise ein Newton–Verfahren, zu ersetzen. Zusätzlich können noch die physiologischen Randbedingungen der Kalibration, also die natürlichen Schwankungsbereiche der Parameter R_{auge} und d in ein modifiziertes Zielfunktional einbezogen werden, in dem dafür gesorgt wird, daß bei Verlassen des physiologischen Fensters das zu minimierende Zielfunktional stark anwächst.

5.1.2 Kalibrations–Setup

Wie aus den Simulationsläufen zur Bestimmung der Güte des Kalibrationsalgorithmus weiter deutlich wurde, hängt das Konvergenzverhalten des Algorithmus von der Verteilung der Fixationspunkte ab. Prinzipiell ist es nun zwar möglich, diese Abhängigkeit genauer zu untersuchen und Angaben über optimale Punktverteilungen zu machen, allerdings würde eine solche Untersuchung den Rahmen dieser Arbeit sprengen, und wäre dem Problem auch nicht angemessen.

5.2 Exakter Meßalgorithmus

Die Annahme, daß der Pupillenmittelpunkt sich bei Drehungen des Auges in einer Ebene bewegt, die parallel zur Kameraebene ist, führt zwar, wie in Anhang B dargestellt, auf einen Fehler von maximal lediglich etwa 0.5° bei Auslenkungen von bis zu 20° in gleichzeitig horzontaler und vertikaler Richtung und wäre somit tolerierbar.

Andererseits jedoch handelt es sich um einen systematischen Fehler, der durch exakte Rechnung beseitigt werden kann. Daher wird hier der exakte Algorithmus für die Messung von horizontalen und vertikalen Augenpositionen gegeben.

Wie man sich aus der Darstellung des Drehmodells klar machen kann, bewegt der Mittelpunkt der Pupille sich bei Drehungen des Auges in horizontaler und vertikaler Richtung auf einem Torus, dessen Grundradius dem Parameter d, und dessen zweiter Radius dem Augenradius R_{auge} entspricht (Siehe Abbildung 5.1 und 5.2).

Für einen allgemeinen Punkt P(x,y,z) auf dem Torus gilt nun:

$$x^2 + \left[\sqrt{y^2 + (z-d)^2} + d\right]^2 = R_{auge}^2 \tag{5.1}$$

Zur Ableitung dieser Gleichung siehe Abbildung 5.1.

Das allgemeine Problem einer Messung mit dem Video–Okulographie–System ist nun, die durch die Lage der Linse und die Lage des Meßpunktes im Kamerabild definierte Gerade mit diesem Torus zu schneiden.

Diese Gerade verläuft durch die beiden Punkte (0,0,g) (Linsenzentrum) und (x,y,b+g) (Bildpunkt) und hat damit, parametrisiert in z, die Gleichung

$$\vec{x}_{abb} = \frac{z-g}{b} \begin{pmatrix} x_{bild} \\ y_{bild} \\ z\frac{b}{z-g} \end{pmatrix} \tag{5.2}$$

Das eingesetzt in die Torusgleichung 5.1 ergibt eine Gleichung 4. Ordnung für z:

$$\left(x_{bild}\frac{b}{g-z}\right)^2 + \left[\sqrt{\left(y_{bild}\frac{b}{g-z}\right)^2 + (z-d)^2} + d\right]^2 = R_{auge}^2 \tag{5.3}$$

Zwar ließe sich diese Gleichung prinzipiell in geschlossener Form lösen. Diese Lösung fiele allerdings reichlich unhandlich aus, so daß eine numerische Lösung naheliegt.

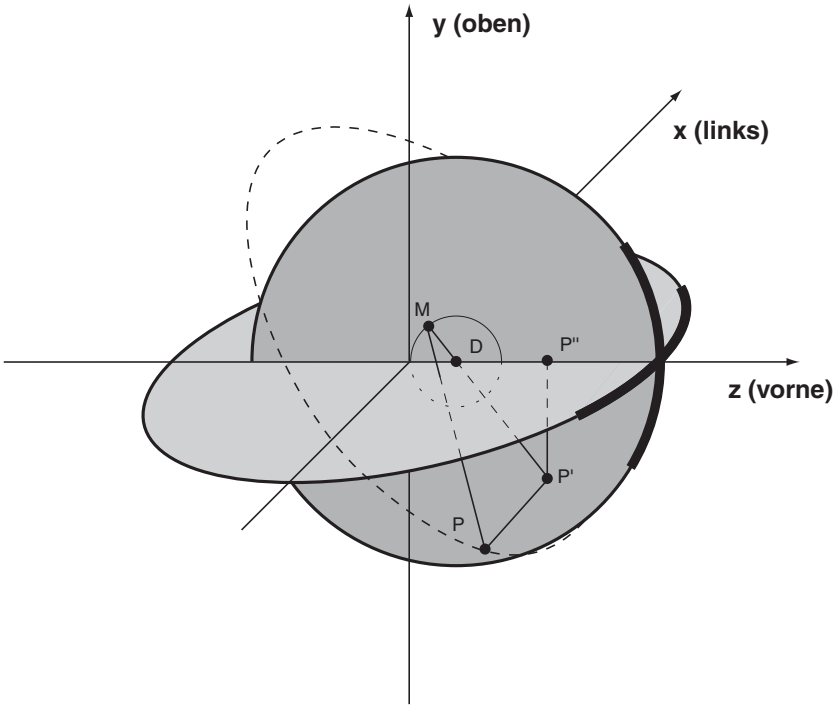

Abbildung 5.1: Ort der Pupillenmittelpunkte bei Drehung des Auges. Die Drehachse für vertikale Drehungen verläuft parallel zur x–Achse durch den Punkt D und hat zum Ursprung des Koordinatensystems den Abstand d. Die Drehachse für horizontale Drehungen verläuft tangential zum eingezeichneten Kreis um D durch den Punkt M, der sich mit vertikaler Auslenkung auf den Kreis bewegt und in der Ruhestellung im Ursprung liegt. Die eingezeichneten gefärbten Kreise entsprechen rein horizontalen bzw. rein vertikalen Auslenkungen. Durch eine vertikale Bewegung des Auges wird also der Kreis horizontaler Auslenkung um die durch D verlaufende Achse gekippt. Ein solcher sich ergebender gekippter Kreis ist durchbrochen eingezeichnet. Auf ihm liegt der allgemeine Punkt P, für den sich die Torusgleichung direkt ablesen läßt. Der Punkt P' stellt die Projektion von P auf die z-y-Ebene, der Punkt P" die Projektion von P' auf die z-Achse dar. Damit gilt für das Dreieck DP'P": $(DP')^2 = (P'P'')^2 + (DP')^2 = y^2 + (z-d)^2$. Damit erhalten wir $MP' = d + DP' = d + \sqrt{y^2 + (z-d)^2}$. Aus dem Dreieck MPP' erhalten wir dann die eigentliche Torusgleichung: $R^2_{auge} = (MP)^2 = (MP')^2 + (PP')^2 = (d + \sqrt{y^2 + (z-d)^2})^2 + x^2$. Die Winkel ϕ und θ finden wir in der Zeichung als die Winkel $\phi = PMP'$ und $\theta = P''DP'$. Siehe auch Abbildung 5.2.

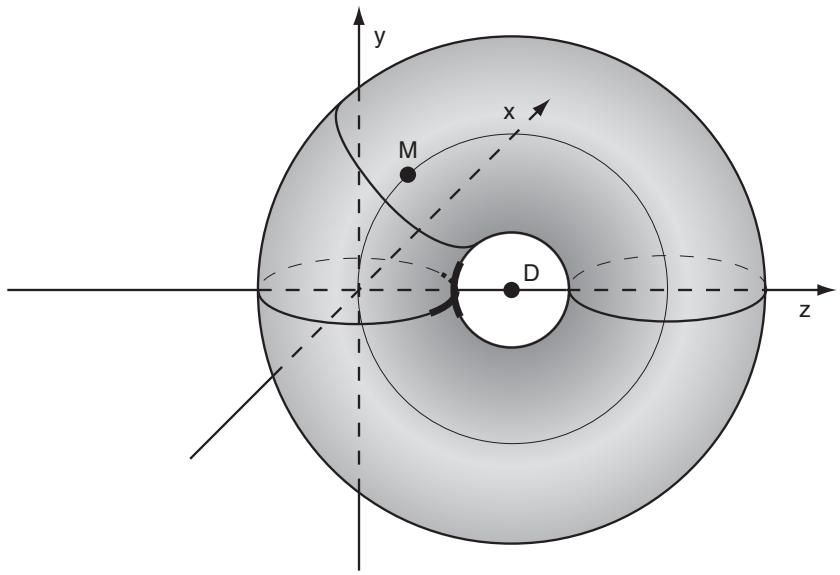

Abbildung 5.2: Für eine extrem große axiale Verschiebung d tritt die entstehende Torusform besser erkennbar hervor. Der Torus in diesem Bild entsteht völlig analog zu dem in Abbildung 5.1 gezeigten durch Rotation des Kreises horizontaler Auslenkung um die durch den Punkt D verlaufende Achse.

Abermals wird hierfür ein Nelder–Mead–Verfahren verwandt, diesmal im eindimensonalen Raum des Parameters z.

Ist mit dem numerischen Verfahren nun z ermittelt, so können wie im Fall der Ebenen–Näherung die Winkelkoordinaten aus $x_{bild}, y_{bild}, \mathcal{F} = \frac{g-z}{b}$ und den Kalibrationsparametern nach Gleichung 1.34 bestimmt werden.

Die Simulation von Messungen zeigt, daß der Meßfehler sich mit diesem Verfahren im Rahmen der numerischen Genauigkeit des verwendeten Systems bewegt.

5.3 Verkürzung der Belichtungszeit

Wie sich gezeigt hat, ist mit dem gegenwärtigen Meßverfahren eine Messung der Torsion des Auges nur sehr begrenzt möglich. Insbesondere führen die Effekte aufgrund der Bewegungsunschärfe dazu, daß mit Mustern mittlerer Qualität bei einer relativ geringen Augengeschwindigkeit in horizontaler oder vertikaler Richtung die Torsionsmessung unmöglich wird.

Die einfachstmögliche apparative Verbesserung, die dieses Problem einschränken oder beseitigen könnte, bestünde darin, die kontinuierlich leuchtenden Infrarot–Leuchtdioden durch Infrarot–Blitzlichter zu ersetzen, die

im zeitlichen Mittel dieselbe Energie abstrahlen und das Auge somit nicht stärker belasten, aber die Belichtungszeit deutlich verkürzen.

Eine Verkürzung der Belichtungszeit schlägt sich direkt antilinear auf die maximale Augengeschwindigkeit nieder, das heißt, eine Verkürzung der Belichtungszeit auf ein Zehntel und damit eine Blitzdauer von etwa 0.2 ms bei einer Frequenz von 50 Hertz erbringt eine Torsionsmeßbarkeit bis zu Augengeschwindigkeiten von etwa 500° pro Sekunde. Das ist etwa die physiologische Grenze für Augenbewegungen und sollte also ausreichend sein.

5.4 Verwendung mehrerer Segmente für die torsionelle Messung

Ebenfalls aus den Ergebnissen der Bewegungssimulationen läßt wurde erkennbar, daß die Artefaktbildung und die Verwischungseffekte bei Irismustern deutlich von der Richtung der Augenbewegung in Bezug auf das zur Messung herangezogene Irismuster abhängen.

Um eine möglichst optimale Stabilität der Messungen gegen Störeinflüsse zu gewährleisten, scheint es daher angeraten, anstelle eines einzelnen Irismusters mehrere Muster in unterschiedlicher Lage und Ausrichtung zu verwenden.

Dies hätte den zusätzlichen wichtigen Vorteil, daß Fehler bei der Pupillenfindung, die zu einer scheinbaren Torsion führen, (da die Muster an der falschen Stelle ausgelesen werden) korrigiert werden könnten.

Wird beispielsweise das Pupillenzentrum fehlerhaft an einer nach unten gegen das tatsächliche Pupillenzentrum verschobenen Stelle erkannt, so führt dies dazu, daß bei Segmenten, die im Bild links vom Pupillenzentrum liegen, eine Torsion im Uhrzeigersinn, bei Segmenten rechts vom Puillenzentrum eine gleich große Torsion im Gegenuhrzeigersinn gemessen würde. Bei Verwendung eines einzelnen Musters läßt sich dieser Meßfehler nicht erkennen, bei mehreren Mustern läßt er sich nicht nur erkennen, sondern im Idealfall auch beseitigen.

5.5 Musterfindung

Die Auffindung von für die Messung der Torsion besonders geeigneten Mustern stellte sich als schwieriger heraus, als von uns gehofft. Weder ließen sich Angaben über die Frequenzverteilung geeigneter Muster machen, noch stellt eine geringe Halbwertsbreite des Maximums der Autokorrelation (HWBM) eine gute Verwendbarkeit eines Musters sicher. Das zuverlässige Kriterium

der meßbaren Winkelbreite schließlich ist für die Mustersuche im Bild zu aufwendig zu berechnen.

Um dennoch eine Verbesserung der bislang intuitiv vorgenommenen (und, wie sich bei der Untersuchung der HWBM gezeigt hat, wenig verläßlichen) Auswahl zu ermöglichen, könnte allerdings eine Mischung aus HWBM und meßbarer Winkelbreite zum Einsatz kommen.

Zwar ist eine geringe HWBM kein Garant für eine große meßbare Winkelbreite, aber die Wahrscheinlichkeit, mit geringer Halbwertsbreite auch ein gutes Muster vorliegen zu haben, scheint ausreichend hoch, um folgendes Verfahren anwenden zu können.

5.5.1 Verfahren zur Musterfindung

Die Muster eines in Frage kommenden Irisbereiches werden mittels Autokorrelationsberechnung und anschließender Bestimmung der HWBM kartiert. Für die n Muster mit der geringsten HWBM wird anschließend die meßbare Winkelbreite berechnet und das nach diesem Kriterium beste ausgewählt. Die Zahl n ist dabei so zu wählen, daß mit hoher Wahrscheinlichkeit sehr gute Muster in der Auswahl enthalten sind, andererseits jedoch der Aufwand für die Berechnung der n Gütekurven relativ gering bleibt. Eine Zahl von 10 bis 20 Mustern scheint aufgrund der gesammelten Daten vernünftig.

Dieses Verfahren bietet zwar keine Garantie, daß das beste Muster im untersuchten Bereich tatsächlich gefunden wird, aber eine ausreichend hohe Wahrscheinlichkeit, daß das ermittelte Muster sich gut oder sehr gut zur Messung eignet.

Gibt man dem Anwender zusätzlich die Möglichkeit, das automatisch gefundene Muster abzulehnen oder neu suchen zu lassen, so sollte dieses Verfahren einen großen Fortschritt gegenüber einer rein manuellen Auswahl darstellen.

Aber auch bei einer Beschränkung auf manuelle Auswahl ist die Berechnung und Anzeige der meßbaren Winkelbreite deutlich besserer Anhaltspunkt und Entscheidunshilfe für die Richtigkeit der Wahl des Musters als es die Autokorrelationsfunktion wäre.

Anhang A

Algorithmen

Alle in dieser Arbeit beschriebenen und verwendeten Algorithmen wurden in der mathematischen Programmiersprache Matlab, Versionsnummer 5.2, verfaßt. Der Programmtext findet sich in Anhang E und enthält selbst eine ausführliche Dokumentation. Zusätzlich wird hier für jeden einzelnen Programmabschnitt in einer Übersicht die Funktion der einzelnen Module genau erläutert.

A.1 2-D-Kalibration

A.1.1 Kalibration

Der eigentliche Kalibrationsalgorithmus verteilt sich auf die drei Dateien `kalib`, `abweich` und `abstand`. In `abstand` wird mittels der Gleichungen 1.8 und 1.16 der Offset der Datenpunkte vom Referenzpunkt im ungedrehten Bild ($\delta\vec{\tilde{x}}_i'$) berechnet. `abweich` verwendet diesen Offset, subtrahiert den Offset der Meßdaten ($\delta\vec{x}_i$), berechnet die Quadratsumme der Differenzen und liefert sie als Ergebnis zurück. `kalib` ruft zur eigentlichen Kalibrierung die in Matlab eingebaute Funktion `fmins`, die den Nelder–Mead–Fit durchführt, mit dem Auftrag auf, `abweich` bezüglich der vier Kalibrationsparameter zu minimieren. Die von `fmins` zurückgegebenen Parameter werden zur Berechnung des Kamerarotationswinkels α verwendet und die fünf so bestimmten Parameter dann als Ausgabe zurückgegeben.

`kalib` hat zusätzlich zu den Listen von Fixationswinkeln und Pixelkoordinaten als weiteren Parameter noch ein Flag, das angibt, ob bei der Berechnung des Abstandes vom Referenzpunkt die genaue optische Abbildung oder die Näherung herangezogen werden soll. Dies ist für die Überprüfung der Messgenauigkeit und den damit zusammenhängenden Optimierungsvor-

schlag in 2.7 von Bedeutung.

A.1.2 Überprüfung

Die praktische Überprüfung wird wie in Kapitel 2 erläutert durch Simulation tatsächlicher Kalibrationsvorgänge vorgenommen. Dafür zuständig ist das Programmodul `kaltest`, das die Anzahl der Simulationen, den Namen der Protokolldatei, falls es eine geben soll, die Fehlerschwankung der Fixationspunkte und ein Flag übergeben bekommt. Das Flag gibt an, ob die Fixationspunkte zufällig verteilt oder auf einem vorgegebenen Raster (siehe Abbildung A.1) liegen und ob die Daten der einzelnen Simulationen (alle Winkel und Pixelkoordinaten, sowie die vorgegebenen und die von `kalib` errechneten Parameter) zur Kontrolle gespeichert werden sollen. `kaltest` ruft nun das Modul `test_kalib` so oft auf, wie im Parameter `anzahl` vorgegeben. `test_kalib` nimmt den eigentlichen Test wie folgt vor und liefert die 5 Fehlerwerte in einem Vektor zurück, den `kaltest` aufzeichnet.

`test_kalib` erhält von `kaltest` das Flag sowie die Größe der vorgesehenen Fehlerschwankung übergeben und legt nun die fünf Kalibrationsparameter mittels eines Zufallsgenerators fest. Dann wird eine Liste von Fixationswinkeln zufällig angelegt oder im Raster gewählt und über die Funktion `abstand` die Lage der Pupillenmittelpunkte im Bild berechnet—inklusive des nach Vorgabe berechneten statistischen Fixations–Fehlers. Anschließend ruft `test_kalib` den Kalibrationsalgorithmus `kalib` auf und übergibt ihm die Winkel- und Pixelkoordinaten. Aus den von `kalib` erhaltenen Parameterwerten wird die Abweichung von den vorgegebenen Parametern bestimmt und an `kaltest` zurückgegeben.

Nach der vorgegebenen Anzahl von Aufrufen von `test_kalib` mittelt `kaltest` die Fehlerwerte über alle Simulationsdurchläufe.

Dieser Vorgang wird für einen bis 13 Punkte wiederholt, anschließend wird der gemittelte Fehler entweder grafisch ausgegeben, oder, falls ein Dateiname angegeben war, als Tabelle in einer Datei gespeichert.

Die Wertebereiche der Kalibrationsparameter betragen dabei (in Klammern jeweils die Einheit)

$$\begin{array}{rcl} 290 \leq & x_{ref}(\text{Pixel}) & \leq 350 \\ 170 \leq & y_{ref}(\text{Pixel}) & \leq 230 \\ 97 \leq & R_{auge}(\text{Pixel}) & \leq 103 \\ 19,5 \leq & d(\text{Pixel}) & \leq 20,5 \\ -5 \leq & \alpha(\text{deg}) & \leq 5 \end{array}$$

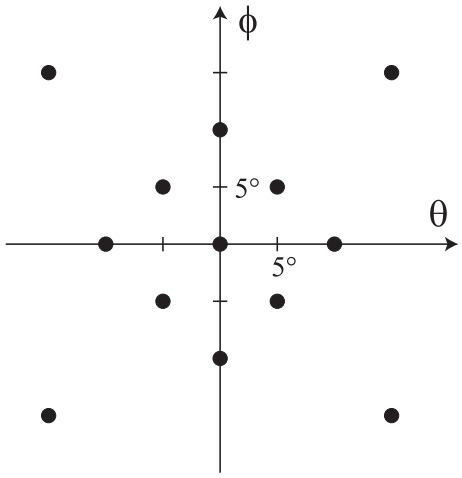

Abbildung A.1: Verteilungsraster für die Fixationspunkte

$$\begin{array}{lll} -30 \leq & \theta_i(\text{deg}) & \leq 30 \\ -30 \leq & \phi_i(\text{deg}) & \leq 30 \end{array}$$

Das Modul `darstellen` dient der Einsicht in die von `test_kalib` auf Wunsch erzeugten Protokolldateien und zu ihrer automatischen Löschung, das Modul `findname` erzeugt automatisch Namen für diese Dateien.

A.1.2.1 Geometrische Näherung

Der Fehler bei der geometrischen Näherung konstanter Gegenstandsweite wurde über simulierte Kalibrations–/Meßvorgänge berechnet. Dazu ruft das Programm `messtest` das Modul `test_mess` eine Anzahl von Malen auf und gibt die mittleren Fehler aus. Für die Graphen und Tabellen in Anhang B wurde allerdings nur ein einzelner Messvorgang simuliert.

`test_mess` selbst legt zunächst die Kalibrationsparameter fest, berechnet die Bildkoordinaten einer Anzahl von Fixationspunkten und übergibt beides an `kalib`. Je nach Wahl des Programmparameters `geon` verwendet `kalib` nun die geometrische Näherung oder nicht. Wenn die geometrische Näherung angewandt wird, wird das zuvor geschilderte Kalibrationsverfahren wie folgt modifiziert:

Die mit den korrekten Abbildungsgleichungen berechneten Pixelkoordinaten der Fixationspunkte werden an `kalib` übergeben, das nun die Kalibration unter Annahme der Näherung durchführt. Das heißt, die für jeden Parametersatz berechneten erwarteten Pixelkoordinaten werden berechnet, *als ob* die

Pupille in einer zur Linsenebene parallelen Ebene bliebe, und auf Basis dieser Annahme werden die Abstände und die Standardabweichung bestimmt. Die Modifikation der Formeln kann im Modul `abstand` gesehen werden und besteht im Einsetzen des genäherten R_{auge} für das korrekte z_{kopf}.

Nachdem nun `kalib` die Kalibrationsparameter bestimmt hat, wird nun in `test_mess` ein 40°x 40° großes Raster mit einem Grad Punktabstand erzeugt. Dieses Raster wird mit den vorgegebenen Kalibrationsparametern auf die Kameraebene abgebildet. Das hierdurch erhaltene Pixelraster wird nun dem Modul `messe` zusammen mit den aus der Kalibration erhaltenen Parametern übergeben. `messe` berechnet nun aus den Pixelwerten und den Parametern wieder die Winkelwerte. Von diesen von `messe` berechneten Winkeln wird nun das ursprüngliche Winkelraster abgezogen. Die sich ergebende Differenz, die den durch die Näherung erzeugten Fehler bezeichnet, gibt `test_mess` zurück an `messtest`, wo er über die Zahl der Durchläufe gemittelt wird.

Im Modul `messe` gibt es zusätzlich noch die Möglichkeit, die exakten Gleichungen für die optische Abbildung auch bei der Messung zu berücksichtigen. Dazu wird zunächst aus den Pixelkoordinaten über ein numerisches Verfahren die Koordinate z_{kopf} des Pupillenmittelpunktes berechnet, und dann mit dieser Koordinate die exakte geometrische Rücktransformation durchgeführt.

A.2 Belichtungssimulation

A.2.1 Berechnung der bewegungsunscharfen Muster

Die Berechnung der durch die Drehung des Auges bewegungsunscharfen Irismuster erfolgt im Modul `simulrot`, das dazu auf die beiden Transformationen `rtrans` und `hintrans` zurückgreift. `rtrans` berechnet die Polarkoordinaten auf der Iris aus den Pixelkoordinaten in der Kameraebene bei vorgegebenen Augendrehwinkeln, `hintrans` berechnet umgekehrt die Kamerakoordinaten eines Punktes, der in Iriskoordinaten gegeben ist.

`simulrot` erhält die Lage des Segmentes, die Position des Auges in θ und ϕ zur Mitte der Belichtungszeit, die Winkelgeschwindigkeit sowie die Belichtungsdauer und die Anzahl der berechneten Teilbilder übergeben. Es berechnet nun zunächst die Pixelkoordinaten des gewählten Segmentes zum Zeitpunkt der Mitte der Belichtungszeit und liest an dieser Stelle das Muster, das zu diesem Zeitpunkt dort liegt, aus.

Das Auslesen von Mustern erfolgt dabei stets so, daß zunächst aus den Pixelkoordinaten die Iriskoordinaten mittels `rtrans` berechnet werden, dann wird der Grauwert an einer Stelle in Iriskoordinaten mit dem Modul `intens` aus einem vorliegenden Bild eines Auges ausgelesen.

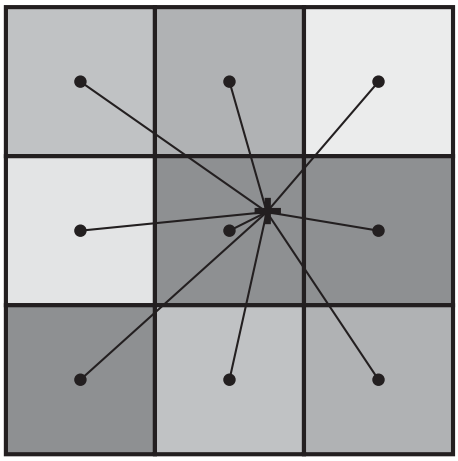

Abbildung A.2: Gewichtung der einzelnen Pixel beim Auslesen der Bildinformation. Eingezeichnet sind die Abstände des durch ein Kreuz angedeuteten exakten Meßpunktes von den Pixelkoordinaten. Das Gewicht der einzelnen Pixel ist der Kehrwert des eingezeichneten Abstandes in Einheiten der Pixelbreite, maximal jedoch 100. Das heißt, Pixel die näher als ein hundertstel der Pixelbreite am eigentlichen Meßpunkt liegen, werden so gewichtet, als lägen sie in dieser Entfernung vom Meßpunkt.

Dazu werden zunächst aus den Polarkoordinaten die Pixelkoordinaten in der vorliegenden Aufnahme des Auges mit `hintrans` berechnet. Dies werden im allgemeinen gebrochene Koordinaten sein. Der Grauwert an der Stelle dieser gebrochenen Koordinaten wird nun aus dem Grauwert des unter den genauen Koordinaten liegenden Pixels sowie den Grauwerten der 8 dieses Pixel umgebenden Pixel berechnet. Dazu werden die jeweiligen Grauwerte mit ihrem Abstand von den genauen Pixelkoordinaten gewichtet. Der Ort eines Pixels für die Berechnung des Abstandes ist dabei der Kreuzungspunkt seiner Diagonalen (siehe Abbildung A.2). Das so erhaltene Grauwertmuster wird nun auf einen Kontrast von eins normalisiert und auf den Mittelwert null geschoben.

Nun werden an der vorgebenenen Anzahl von Zeitpunkten während der Belichtungszeit die Winkelpositionen des Auges berechnet, anschließend wieder mittels `rtrans` die Iriskoordinaten der unter dem voreingestellten Pixelsegment liegenden Punkte auf dem Auge und schließlich die zum jeweiligen Zeitpunkt unter dem Segment liegenden Muster mit `intens` ausgelesen.

Über all diese Muster wird nun noch über die Belichtungszeit gemittelt, und das Ergebnis zusammen mit dem Referenzmuster in eine Datei, deren Name `simulrot` als Parameter übergeben wurde, gespeichert[1].

[1] Zusätzlich wird noch ein weiteres Muster, das Basismuster, gespeichert. Dieses Muster

`simulplan` ruft `simulrot` mit verschiedenen Winkelgeschwindigkeiten auf und erzeugt die Namen der Datendateien. Diese Datendateien können durch Aufruf von `simulplan(2)` auch unter Benutzung des Moduls `laden` betrachtet werden. `laden` erlaubt dann die Betrachtung der Muster selbst, die Bildung der Fouriertransformierten, die Bildung der Kreuzkorrelation aus Referenzmuster und verschmiertem Muster und die Berechnung der Gütekurve beider Muster.

A.2.1.1 Auswertung der Daten

Die Auswertung der von `simulplan` geschriebenen Dateien erfolgt im Modul `simauswerten`. `simauswerten` lädt die von `simulplan` erzeugten Musterdateien und übergibt das verschmierte Muster und das mittlere Drittel des Referenzmusters an `guetekurve`, das wie zu erwarten Gütekurve berechnet. Die Gütekurve wird anschließend an `winkelbreite` übergeben, das daraus die meßbare Winkelbreite bestimmt.

Für die Berechnung der Kurve werden von `guetekurve` die Kreuzkorrelationen der Teile des Musters mit dem mittleren Drittel des Referenzmusters berechnet, und durch `paramax` der Parabelfit an die Kreuzkorrelationskurve durchgeführt. Die erhaltenen Maximumspositionen werden mit den erwarteten verglichen und die Abweichung (also die Gütekurve) als Vektor zurückgegeben.

`paramax` verwendet für den Parabelfit die in Matlab eingebaute Funktion `polyfit` und nutzt die 15 Pixel um das Maximum der Kreuzkorrelationskurve für die Berechnung der Parabelparameter.

`winkelbreite` bestimmt schließlich aus einer gegebenen Gütekurve die meßbare Winkelbreite. Dazu wird ihm die maximale Abweichung übergeben, bis zu der die Messung der Torsion als geglückt gelten soll.

Das Modul `laden` schließlich wird von `simulplan` aufgerufen, wenn die Daten betrachtet werden sollen, und erlaubt die Online-Analyse einzelner Muster.

A.3 Segmentanalyse

Für die Analyse der Segmente wurden zunächst für das Irissegment von -22,5° bis 22,5° Polarwinkel die Muster extrahiert und mit zwei verschiedenen Methoden bewertet. Zum einen wurde im Modul `halbwert`, das auch mittels der bereits im Simulationsteil zum Einsatz gekommenen Moduls `intens` die

---

wird in einer Auge–Frontal–Stellung ($\theta = 0, \phi = 0$ ausgelesen und diente der Kontrolle der Transformationsmodule `rtrans` und `hintrans`.

Musterauslesung vornimmt, die Halbwertsbreite des Maximums des Autokorrelationsfunktion des Muster bestimmt, zum anderen im Modul `findbest` die meßbare Winkelbreite des Musters.

Die verschiedenen Filter wurden im Modul `buttern` angewandt, das auch die Ergebnisse darstellt. `buttern` erhält dazu das Muster zusammen mit Angaben über die vorzunehmenden Filterungen übergeben. Für die Berechnung der Filterparameter wurde die in der Statistics-Toolbox von Matlab enthaltene Funktion `butter`, zur Anwendung der Filter die ebenfalls in dieser Toolbox enthaltene Funktion `filter` verwendet.

Anhang B

Ergebnisse

B.1 Fehler des Kalibrationsalgorithmus

Wie in Anhang A beschrieben wurde der Kalibrationsalgorithmus mit verschiedenen Fehlerwerten für die Fixation und mit zufällig verteilten und in einem Raster arrangierten Fixationspunkten getestet.

Die unten dargestellten Ergebnisse sind die absoluten Fehler in den fünf Parametern in Abhängigkeit von der Anzahl der Kalibrationspunkte.

Die dargestellen Werte sind die über 20 Durchläufe quadratisch gemittelten Fehlerwerte.

B.1.1 Fixationspunkte im Raster

B.1.1.1 Fehler ± 0.00 Grad

Punkte	$x_{ref}(Px)$	$y_{ref}(Px)$	$R_{auge}(Px)$	$d(Px)$	$\alpha(deg)$
1	0.00	0.00	1.54	0.32	3.07
2	0.00	0.00	0.43	0.87	0.28
3	0.00	0.00	0.35	0.69	0.22
4	0.00	0.00	0.27	0.55	0.06
5	0.00	0.00	0.33	0.66	0.00
6	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00

B.1.1.2 Fehler ± 0.25 Grad

Punkte	$x_{ref}(Px)$	$y_{ref}(Px)$	$R_{auge}(Px)$	$d(Px)$	$\alpha(deg)$
1	1.41	0.95	2.05	0.30	2.43
2	1.40	0.92	4.63	1.36	2.81
3	1.17	1.09	2.05	0.93	1.68
4	0.91	1.47	2.19	0.54	1.58
5	0.95	0.85	1.95	0.63	1.15
6	0.45	1.10	4.26	6.23	0.81
7	0.78	0.94	3.11	4.67	0.88
8	0.71	0.48	2.57	3.28	0.62
9	0.75	0.67	2.06	2.32	0.57
10	0.58	0.69	1.95	3.35	0.42
11	0.61	0.76	1.24	1.79	0.37
12	0.68	0.64	1.57	3.10	0.63
13	0.68	0.74	1.39	2.46	0.39

B.1.1.3 Fehler ± 0.50 Grad

Punkte	$x_{ref}(Px)$	$y_{ref}(Px)$	$R_{auge}(Px)$	$d(Px)$	$\alpha(deg)$
1	2.00	2.26	2.04	0.27	3.08
2	3.07	1.44	7.90	1.29	6.62
3	2.02	1.99	4.11	0.90	2.46
4	1.99	2.38	5.04	0.94	2.99
5	1.49	2.07	3.53	0.92	1.96
6	1.82	1.83	9.07	12.26	1.64
7	1.84	1.53	6.88	8.47	1.37
8	1.34	1.22	5.33	5.63	1.25
9	1.12	1.48	3.86	5.16	1.15
10	1.27	1.07	3.25	4.51	1.26
11	1.03	1.49	2.75	4.66	0.84
12	1.03	1.21	2.62	5.30	0.75
13	1.18	1.27	2.57	4.12	0.64

B.1.1.4 Fehler ± 1.00 Grad

Punkte	$x_{ref}(Px)$	$y_{ref}(Px)$	$R_{auge}(Px)$	$d(Px)$	$\alpha(deg)$
1	6.58	3.40	1.68	0.35	3.13
2	5.00	3.14	21.99	3.17	13.21
3	3.93	4.13	8.57	1.37	5.06
4	4.11	5.42	9.25	2.56	4.35
5	3.06	4.84	6.17	1.04	4.04
6	2.93	2.90	12.03	15.55	3.45
7	2.59	3.28	11.22	14.73	3.12
8	3.09	1.76	13.09	13.89	2.64
9	2.52	2.66	7.92	9.88	2.54
10	2.17	2.70	5.42	10.31	2.37
11	2.83	2.56	4.83	10.51	2.38
12	3.12	3.11	4.55	7.71	2.23
13	1.78	3.37	4.45	9.05	1.87

B.1.2 Fixtationspunkte zufällig verteilt

B.1.2.1 Fehler ± 0.00 Grad

Punkte	$x_{ref}(Px)$	$y_{ref}(Px)$	$R_{auge}(Px)$	$d(Px)$	$\alpha(deg)$
1	0.00	0.00	1.61	0.31	2.80
2	0.00	0.00	0.66	0.95	0.22
3	0.03	0.05	0.35	0.53	0.04
4	0.08	0.13	0.32	0.51	0.09
5	0.02	0.04	0.11	0.17	0.00
6	0.01	0.01	0.02	0.05	0.00
7	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00

B.1.2.2 Fehler ± 0.25 Grad

Punkte	$x_{ref}(Px)$	$y_{ref}(Px)$	$R_{auge}(Px)$	$d(Px)$	$\alpha(deg)$
1	1.40	1.18	1.74	0.26	2.79
2	1.54	1.02	4.07	0.78	3.60
3	1.61	0.84	33.84	35.90	1.54
4	0.93	1.07	3.51	6.75	1.29
5	1.35	2.48	6.07	10.18	0.84
6	1.63	1.88	6.03	9.03	1.57
7	0.92	1.21	1.99	3.72	1.01
8	0.99	1.67	2.78	5.59	1.13
9	0.94	1.17	2.21	3.58	0.97
10	0.79	0.86	2.00	2.65	0.72
11	0.50	0.77	1.89	3.55	0.65
12	0.57	0.80	1.55	2.97	0.56
13	0.48	0.74	1.38	1.96	0.48

B.1.2.3 Fehler ± 0.50 Grad

Punkte	$x_{ref}(Px)$	$y_{ref}(Px)$	$R_{auge}(Px)$	$d(Px)$	$\alpha(deg)$
1	3.07	2.27	1.57	0.25	2.97
2	2.45	2.27	7.56	1.56	3.47
3	2.84	2.58	19.68	40.31	6.52
4	3.53	3.07	5.87	22.61	4.27
5	3.22	5.04	7.37	15.57	1.76
6	3.14	3.99	8.28	19.25	1.75
7	2.74	2.01	7.18	11.16	2.17
8	1.45	1.94	6.82	11.27	1.32
9	1.48	2.85	3.42	10.15	1.46
10	1.61	2.01	4.37	10.44	2.59
11	1.26	1.29	3.60	5.53	1.86
12	1.06	1.76	2.62	4.05	1.70
13	1.07	1.15	2.66	5.10	1.14

B.1.2.4 Fehler ± 1.00 Grad

Punkte	$x_{ref}(Px)$	$y_{ref}(Px)$	$R_{auge}(Px)$	$d(Px)$	$\alpha(deg)$
1	6.26	4.18	1.64	0.23	2.44
2	5.85	4.07	20.30	2.51	10.47
3	4.76	3.76	38.94	62.67	8.93
4	4.71	4.47	21.65	40.08	5.97
5	3.67	5.43	12.06	35.33	3.91
6	4.99	4.54	11.79	20.22	4.30
7	3.80	3.56	11.91	18.08	3.68
8	4.09	4.11	11.14	20.23	2.50
9	2.34	3.49	4.82	10.52	1.89
10	2.90	2.71	6.92	13.81	4.46
11	4.06	3.10	8.58	14.49	2.25
12	4.08	2.46	8.15	11.90	2.79
13	2.74	2.13	7.23	11.41	2.32

B.2 Fehler durch die geometrische Näherung

Durch die Annahme konstanter Gegenstandsweiten ergibt sich bei Messungen ein Fehler, dessen Größe von den Auslenkungen des Auges abhängt. Die Größe des Fehlers ist in den Graphen und Tabellen ab Seite 80 dargestellt. Seine Form entspricht der aufgrund der Form des Abbildungsparameters $\mathcal{F}$ erwarteten.

Ebenfalls dargestellt ist der beim exakten Meßverfahren verbleibende Fehler, gleichfalls in einer Mittelung über 20 Simulationsdurchläufe.

Es ist deutlich zu sehen, wie der Meßfehler bei großen Augenauslenkungen anwächst, und noch deutlicher zu sehen, daß der Fehler bei Verwendung der numerischen Lösung der exakten Gleichungen (also der exakten Torus-Kinematik anstelle der Ebenen-Näherung) verschwindet.

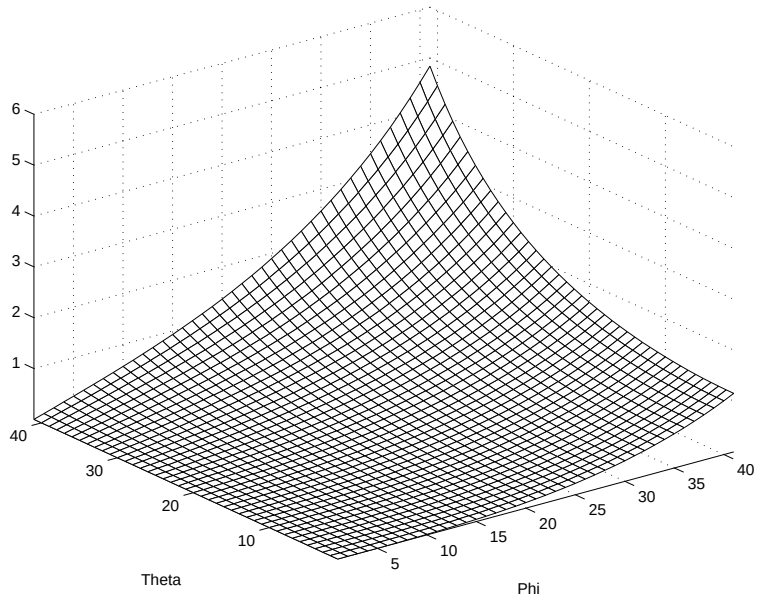

Abbildung B.1: $\Delta\phi$: Meßfehler bei geometrischer Näherung im ermittelten Winkel ϕ. Zu erkennen ist, daß der Meßfehler in ϕ mit wachsendem ϕ größer wird. Zusätzlich ist zu erkennen, daß für große θ (für stark exzentrische Positionen) dieses Anwachsen stärker ist.

$\phi\backslash\theta$	0	5	10	15	20	25	30	35	40
0	-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	-0.000	-0.000
5	0.002	0.004	0.010	0.020	0.033	0.050	0.070	0.094	0.121
10	0.019	0.023	0.035	0.054	0.080	0.114	0.154	0.201	0.255
15	0.066	0.072	0.089	0.117	0.156	0.206	0.266	0.336	0.415
20	0.160	0.167	0.189	0.226	0.278	0.343	0.422	0.514	0.618
25	0.318	0.327	0.354	0.400	0.463	0.543	0.640	0.752	0.880
30	0.564	0.574	0.606	0.659	0.733	0.827	0.941	1.073	1.223
35	0.925	0.937	0.974	1.034	1.118	1.224	1.353	1.503	1.674
40	1.442	1.455	1.495	1.562	1.654	1.773	1.916	2.082	2.272

Tabelle B.1: Fehler in Phi

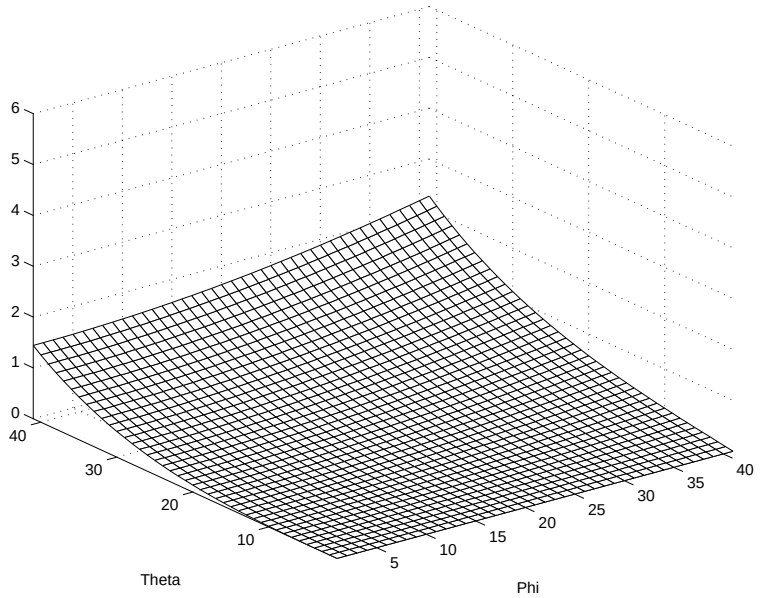

Abbildung B.2: $\Delta\theta$: Meßfehler bei geometrischer Näherung im ermittelten Winkel θ. Es ist dieselbe Asymmetrie zu beobachten wie in Abbildung B.1. Insgesamt ist der Fehler im Winkel θ, also in horizontaler Richtung, schwächer, weil die besondere Form der Kinematik des Auges sich besonders stark in vertikaler Richtung bemerkbar macht.

$\phi\backslash\theta$	0	5	10	15	20	25	30	35	40
0	0.000	0.002	0.016	0.053	0.128	0.254	0.450	0.739	1.150
5	0.000	0.004	0.021	0.061	0.138	0.268	0.469	0.763	1.180
10	0.000	0.012	0.036	0.085	0.172	0.313	0.526	0.836	1.273
15	0.000	0.026	0.064	0.128	0.232	0.392	0.629	0.966	1.438
20	0.000	0.047	0.107	0.195	0.326	0.516	0.788	1.169	1.695
25	0.000	0.079	0.172	0.295	0.465	0.701	1.026	1.471	2.079
30	0.000	0.126	0.268	0.443	0.671	0.973	1.376	1.917	2.646
35	0.000	0.196	0.412	0.665	0.980	1.382	1.903	2.588	3.500
40	0.000	0.306	0.635	1.010	1.460	2.017	2.723	3.635	4.841

Tabelle B.2: Fehler in Theta

$\phi\backslash\theta$	0	5	10	15	20	25	30	35	40
0	0.000	0.002	0.016	0.053	0.128	0.254	0.450	0.739	1.150
5	0.002	0.006	0.023	0.064	0.142	0.273	0.474	0.768	1.186
10	0.019	0.026	0.050	0.100	0.190	0.333	0.548	0.859	1.298
15	0.066	0.076	0.110	0.173	0.280	0.443	0.683	1.023	1.497
20	0.160	0.173	0.218	0.299	0.428	0.620	0.894	1.277	1.805
25	0.318	0.336	0.394	0.497	0.656	0.886	1.209	1.652	2.257
30	0.564	0.588	0.663	0.794	0.994	1.277	1.667	2.197	2.915
35	0.925	0.958	1.057	1.229	1.486	1.846	2.335	2.993	3.880
40	1.442	1.487	1.624	1.860	2.207	2.685	3.329	4.189	5.348

Tabelle B.3: Betrag des Meßfehlers bei geometrischer Näherung. Die graphische Darstellung dieses Fehlers findet sich im Hauptteil der Arbeit.

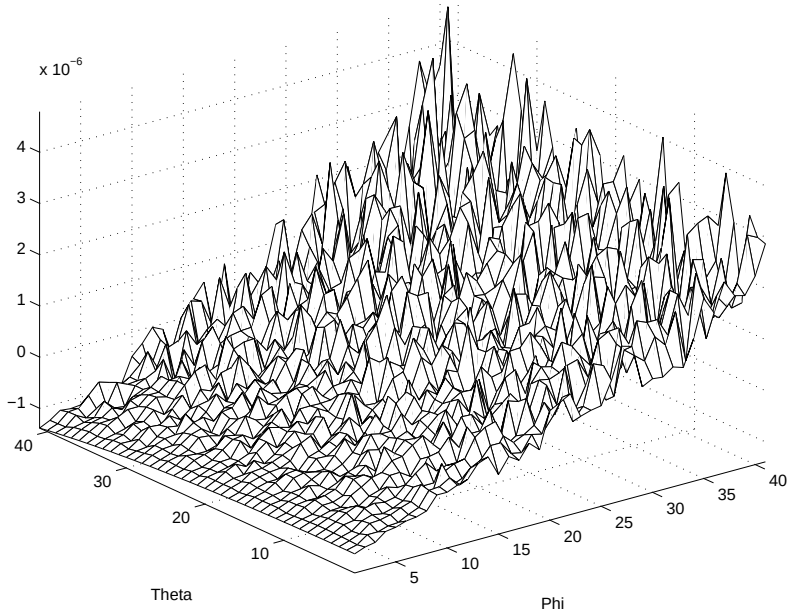

Abbildung B.3: Meßfehler bei exakter Messung im ermittelten Winkel ϕ. Wie erkennbar ist, ist der verbleibende Fehler im wesentlichen Artefakt der Rechengenauigkeit. Auf das Abdrucken der Tabellen wurde daher für diese Daten verzichtet. Sie enthalten ausschließlich Nullwerte.

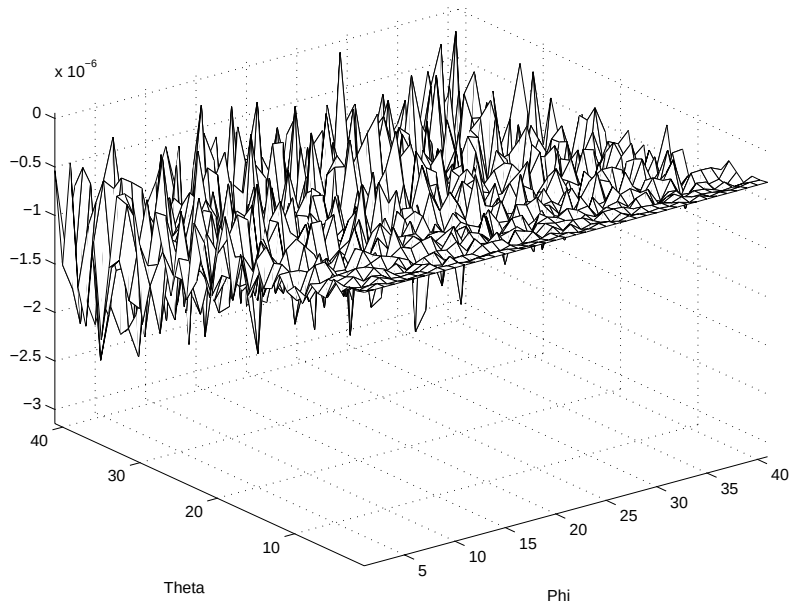

Abbildung B.4: Meßfehler bei exakter Messung im ermittelten Winkel θ

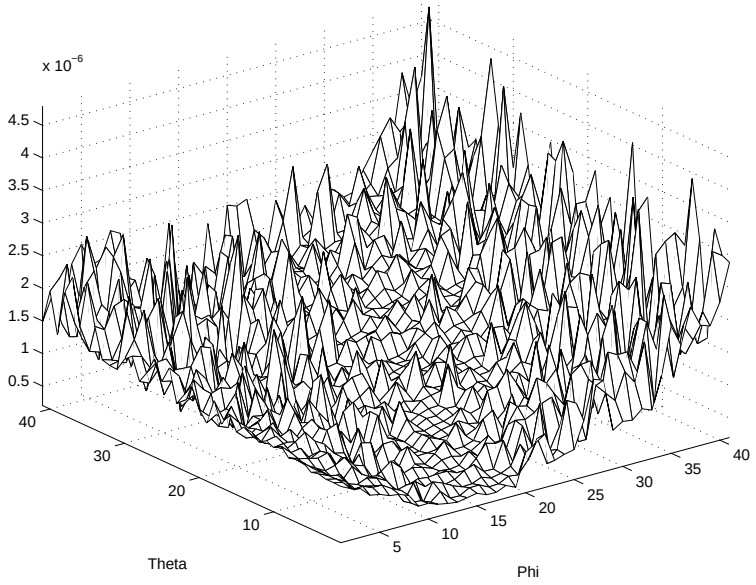

Abbildung B.5: Betrag des Fehlervektors $\sqrt{\delta\phi^2 + \delta\theta^2}$ bei exakter Messung

B.3 Ergebnisse der Simulationsläufe

B.3.1 Meßbare Winkelbreiten

Die Übersicht über die Ergebnisse wurde in Kapitel 3 bereits gegeben. Hier folgen noch die Graphen der meßbaren Winkelbreiten der beiden Muster bei verschiedenen Augengeschwindigkeiten.

Wie man in Abbildung B.7 sehen kann, fällt die meßbare Winkelbreite beim Muster auf 80 Pixeln Radius schon ab einer Geschwindigkeit von 30°/Sekunde unabhängig von der Bewegungsrichtung auf 0° ab. Die Fehlerschranke für diese Kurve lag dabei bei einem halben Pixel, was 0.085 Grad entspricht. (262 Pixel entsprechen 45°). Das heißt, wenn man die Torsion unter Verwendung dieses Musters auf ungefähr 0,1° genau messen möchte, muß die horizontale und vertikale Augengeschwindigkeit sich unter etwa 30° pro Sekunde halten. Erhöht man die Fehlertoleranz, so wächst auch die maximale Augengeschwindigkeit leicht an, bei einer Fehlerschranke von 2 Pixeln oder 0,34° ist eine Messung der Torsion mit diesem Muster bis zu Winkelgeschwindigkeiten von 50° pro Sekunde möglich.

Sehr deutlich ist zu erkennen, daß das aufgrund der Eigenschaft der besten Halbwertsbreite ausgewählte zweite Irismuster ein deutlich stabileres Meßverhalten zeigt. Bei der Fehlerschranke von 0,5 Pixeln/0.085° (Abbildung B.11) liegt die höchste Augengeschwindigkeit bei 70° pro Sekunde, bei einer Fehlertoleranz von 0.34° läßt sich für Bewegungen senkrecht zur Musterrichtung bis zu 170°/Sekunde noch messen, für parallele Bewegungen liegt die Grenzgeschwindigkeit bei 70°/Sekunde.

Der deutlich sichtbare Wiederanstieg der meßbaren Winkelbreite im Falle senkrechter Bewegungsrichtung ist ein Artefakt, das sehr wahrscheinlich auf die Einmischung von dem ursprünglichen Muster sehr ähnlichen Iristeilen in das ausgelesene Bild zurückzufuührren ist.

In Tabelle 3.1 auf Seite 50 sind die Grenzgeschwindigkeiten in der Übersicht zu sehen.

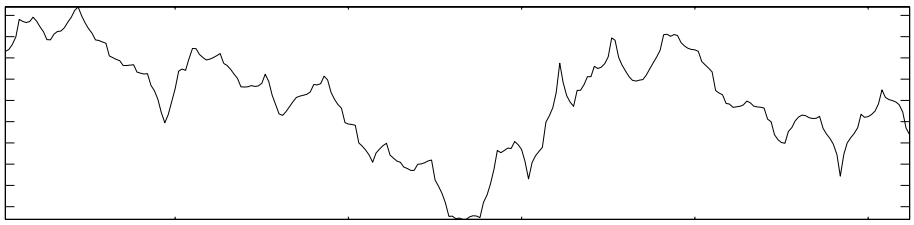

Abbildung B.6: Das Irismuster auf einem Radius von 80 Pixeln

B.4 Ergebnisse der Musterfindung

Die Ergebnisse der Filterläufe lassen praktisch keinen Zusammenhang zwischen der angewandten Filtermethode und der Qualität des gefilterten Musters erkennen. Es wurden variable Tief-, Hoch- und Bandpaßfilter auf die Muster angewendet

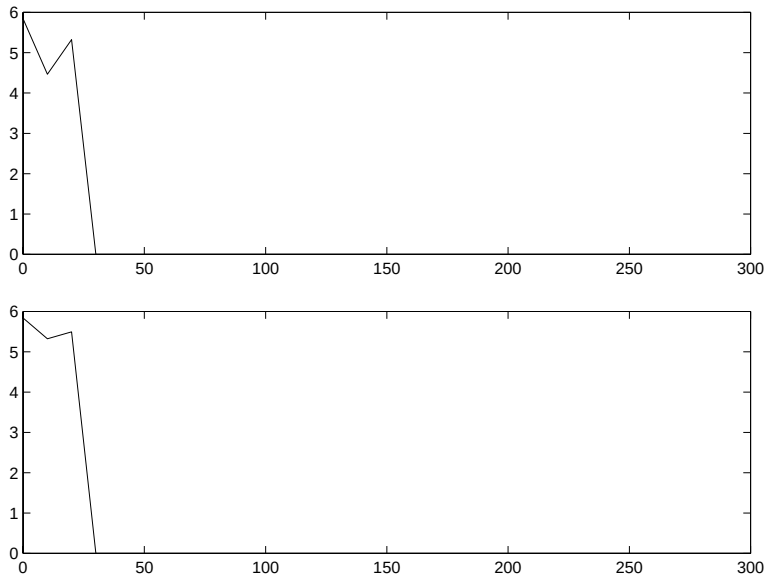

Abbildung B.7: Meßbare Winkelbreite für das Muster auf einem Radius von 80 Pixeln. Die obere Kurve zeigt die meßbare Winkelbreite für Bewegungen senkrecht zum Muster, die untere für Bewegungen parallel zum Muster. Aufgetragen ist der maximale Torsionswinkel, für den der Torsions-Meßfehler unter 0,5 Pixeln oder 0.085° bleibt (meßbare Winkelbreite) gegen die Winkelgeschwindigkeit des Auges in horizontaler Richtung.

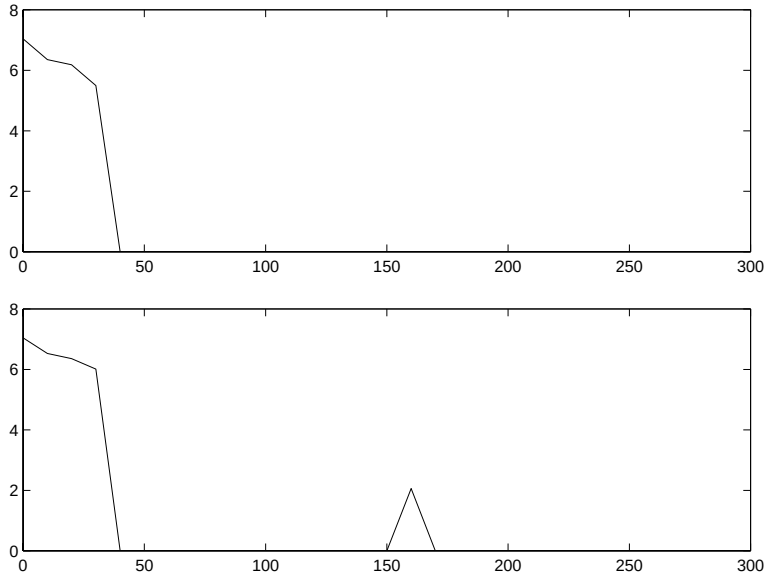

Abbildung B.8: Wie B.7, mit einer Fehlerschranke von 1 Pixel oder 0.17°

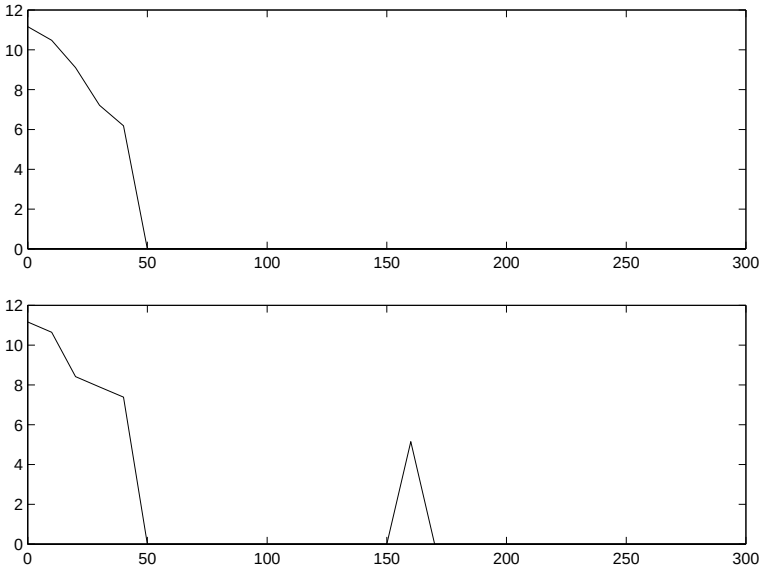

Abbildung B.9: Wie B.7, mit einer Fehlerschranke von 2 Pixeln oder 0.34°

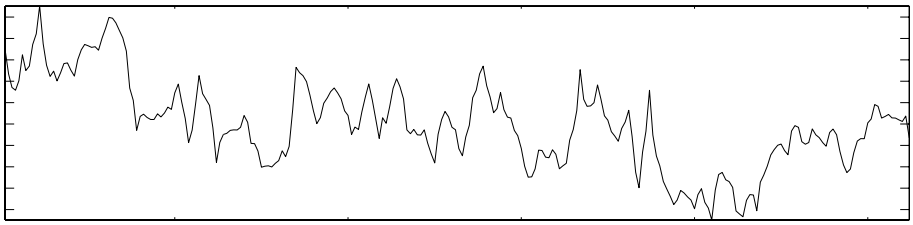

Abbildung B.10: Das Irismuster auf einem Radius von 132 Pixeln

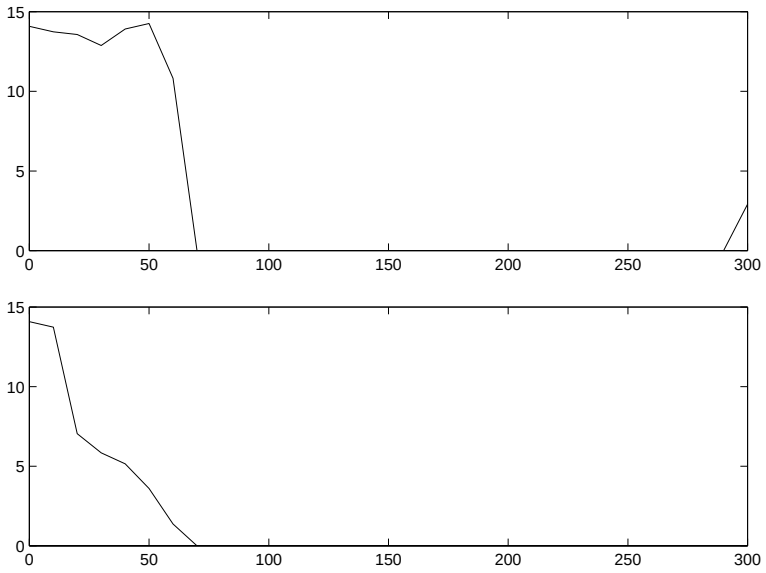

Abbildung B.11: Meßbare Winkelbreite für das Muster auf einem Radius von 132 Pixeln. Die obere Kurve zeigt die meßbare Winkelbreite für Bewegungen senkrecht zum Muster, die untere für Bewegungen parallel zum Muster. Aufgetragen ist der maximale Torsionswinkel, für den der Torsions-Meßfehler unter 0,5 Pixeln oder 0.085° bleibt (meßbare Winkelbreite) gegen die Winkelgeschwindigkeit des Auges in vertikaler Richtung.

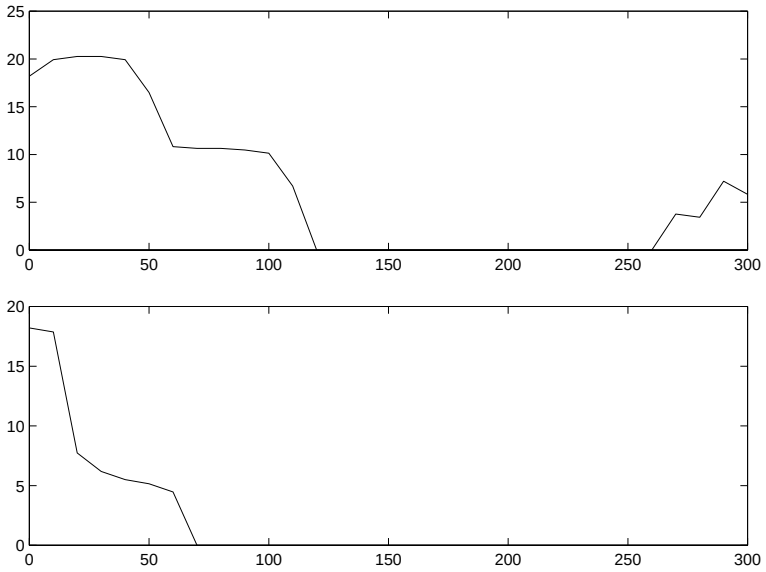

Abbildung B.12: Wie B.11, mit einer Fehlerschranke von 1 Pixel oder 0.17°

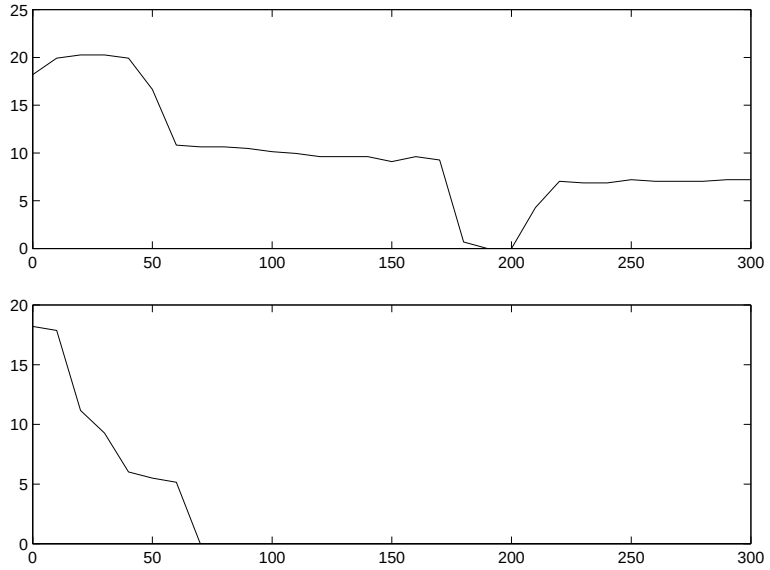

Abbildung B.13: Wie B.11, mit einer Fehlerschranke von 2 Pixeln oder 0.34°

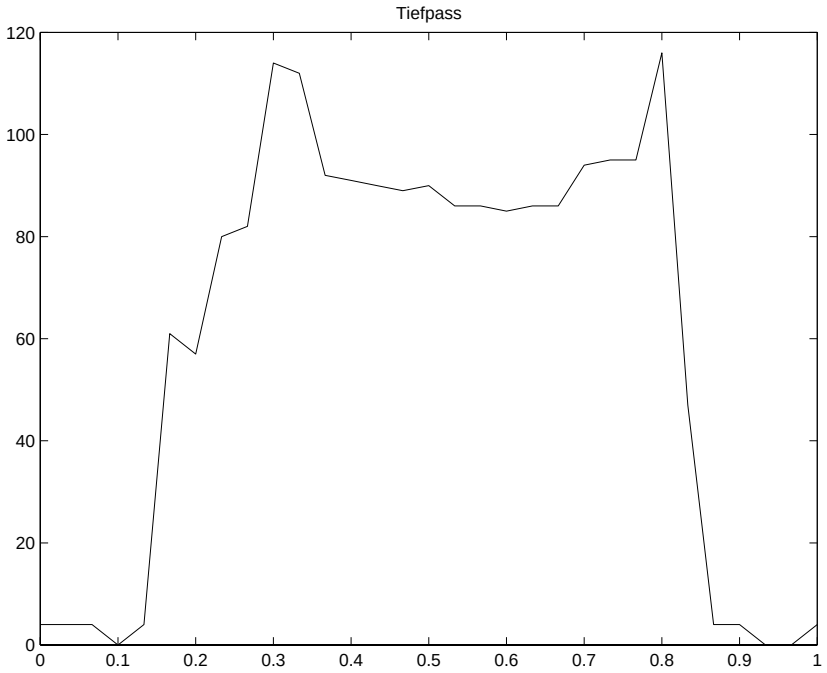

Abbildung B.14: Meßbare Winkelbreite von Irismustern, die durch einen digitalen Butterworth-Tiefpaß-Filter 30. Ordnung gefiltert wurden. Die Abszisse zeigt die Filtergrenze. Dem rechten Rand des Ausschnittes (Wert 1) entspricht die höchste im Muster vorkommende Raumfrequenz (Abtasttheorem), dem linken Rand die tiefste (endliche Musterlänge).

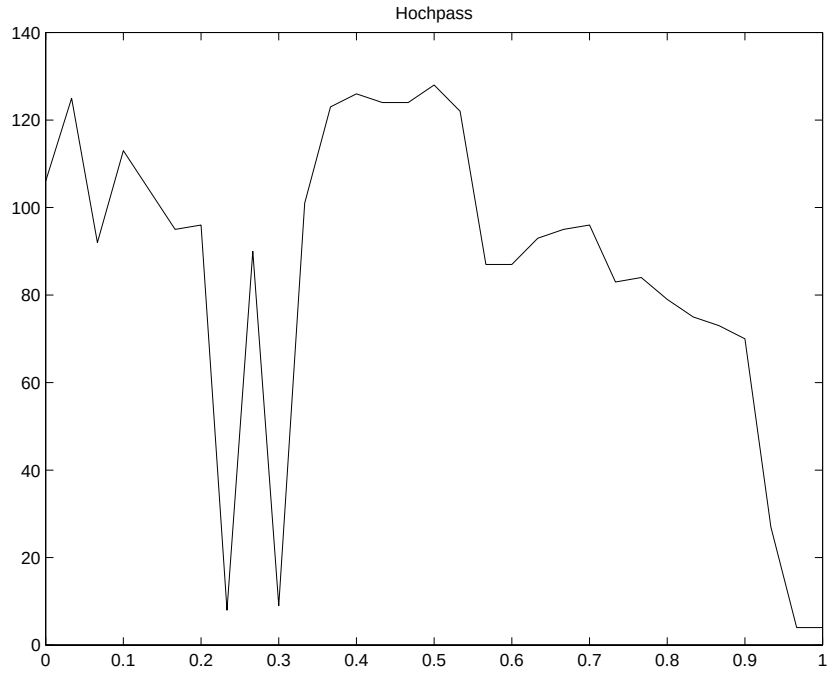

Abbildung B.15: Wie Abbildung B.14 aber mit einem Hochpaßfilter.

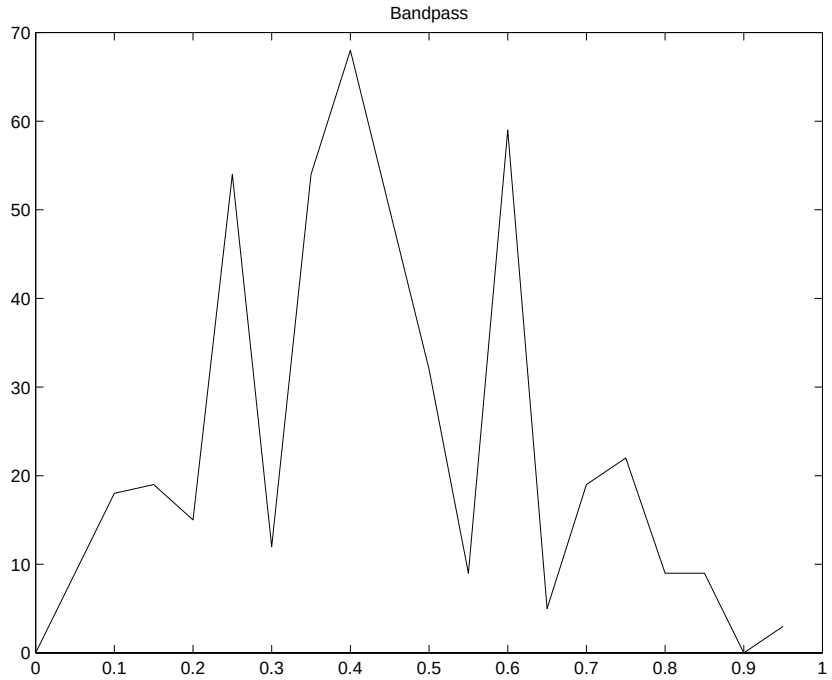

Abbildung B.16: Wie Abbildung B.14 aber mit einem Bandpaßfilter, dessen Bandbreite einem zwanzigstel der maximalen Raumfrequenz entsprach.

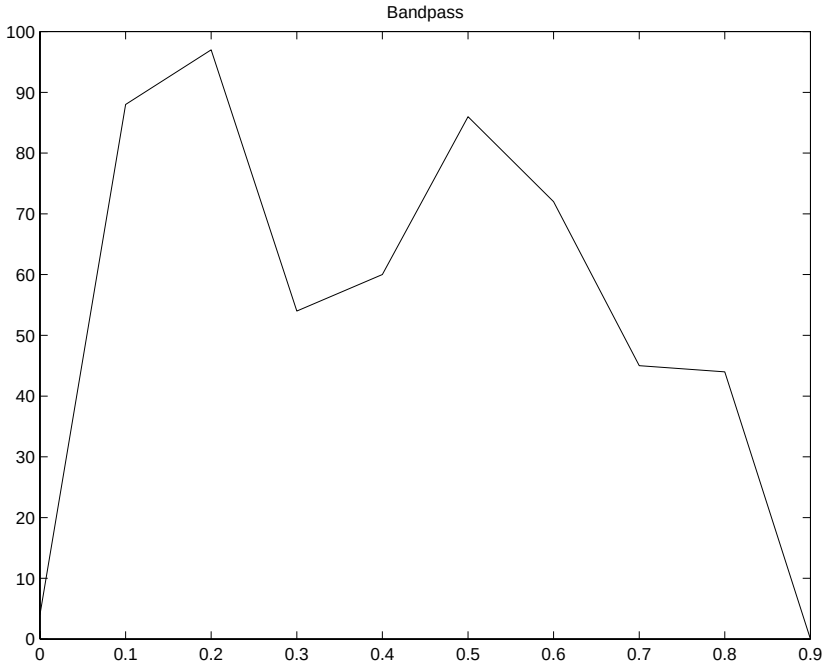

Abbildung B.17: Wie Abbildung B.14 aber mit einem Bandpaßfilter, dessen Bandbreite einem zehntel der maximalen Raumfrequenz entsprach.

Anhang C

Genaue optische Geometrie des Auges

Für diese Arbeit war die genaue Organisation des optischen Apparate Auge nur insoweit von Belang, als es die Kalibration des Meßapparates betraf. Aus Gründen der Vollständigkeit und zur Motivation des in Kapitel 2 dargestellten Verfahrens soll in diesem Anhang ein detaillierter Überblick über die geometrische Optik des Systems Auge gegeben werden. Die Referenz für diesen Teil findet sich in [Le Grand 1980] und [Huber 1998].

C.1 Geometrie des Auges

Das Auge ist, wie alle biologischen Systeme, nur näherungsweise durch einfache geometrische Objekte zu beschreiben. Es hat sich allerdings als nützlich und zweckmäßig erwiesen, das Auge durch eine Kugel (sklerale Sphäre) mit einem durchschnittlichen Radius von 12 mm zu beschreiben, die auf der Vorderseite von einer zweiten, transparenten Kugel (corneale Sphäre) mit einem Radius von durchschnittlich 8 mm abgeschlossen wird (vergleiche Abbildung C.1). Die Mittelpunkte der beiden Kugeln liegen dabei etwa 5 mm auseinander auf der geometrischen Achse, die damit Rotationsachse des Gebildes ist.

Diese Darstellung ist allerdings idealisiert, tatsächlich ist das Auge vorne ein wenig ausgebeult und hinten leicht abgeplattet, und auch die Rotationssymmetrie gilt nur näherungsweise.

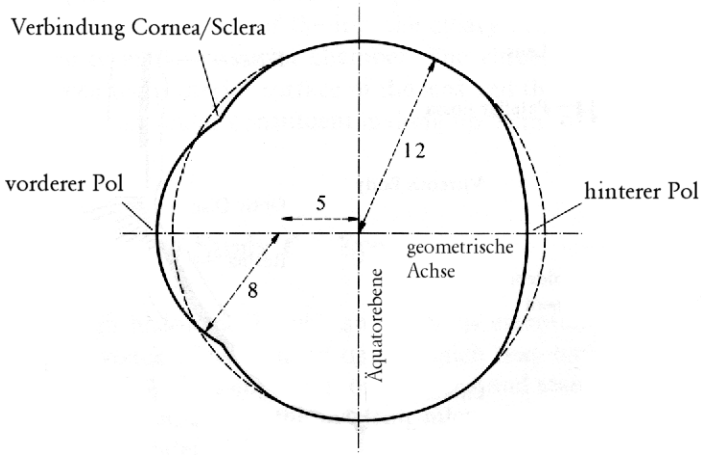

Abbildung C.1: Geometrie des Auges, aus [Le Grand 1980].

C.2 Funktionsweise des visuellen Systemes

Die Aufgabe des visuellen Systemes ist die Aufnahme optischer Bilder der Umgebung und ihre Zuleitung in den visuellen Kortex, wo sie weiterverarbeitet werden.

Dazu wird das einfallende Licht über einen mehrteiligen optischen Apparat auf die Netzhaut projiziert, wo es lichtempfindliche Rezeptorzellen anregt. Es gibt von diesen Zellen im Ganzen vier verschiedene Typen. Die sogenannten "Stäbchenzellen" können allein Helligkeitswerte unterscheiden, während drei Arten von sogenannten "Zäpfchenzellen" für die Wahrnehmung dreier Farben zuständig sind.

Während die Zäpfchen sich im gelben Fleck, der Fovea, konzentrieren, finden sich die meisten Stäbchenzellen etwa 20 $^\circ$ entfernt.

Die Signale aus den Rezeptoren werden dann über mehrere Lagen neurologischer Verschaltung über den Sehnerv in die höheren kortikalen Strukturen geleitet.

Dabei werden unterschiedliche Orte auf der Netzhaut in unterschiedlich starkem Maße zusammengefaßt. Das beste Verhältnis von beteiligten Rezeptorzellen zur Repräsentation im Kortex haben dabei die Zellen der Fovea centralis. Da hier außerdem die Rezeptorendichte am größten ist, ergibt sich für die auf die Fovea fallenden Bilder die höchste Auflösung.

Der visuelle Apparat steuert daher das Auge so, daß fixierte Punkte auf die Fovea projiziert werden.

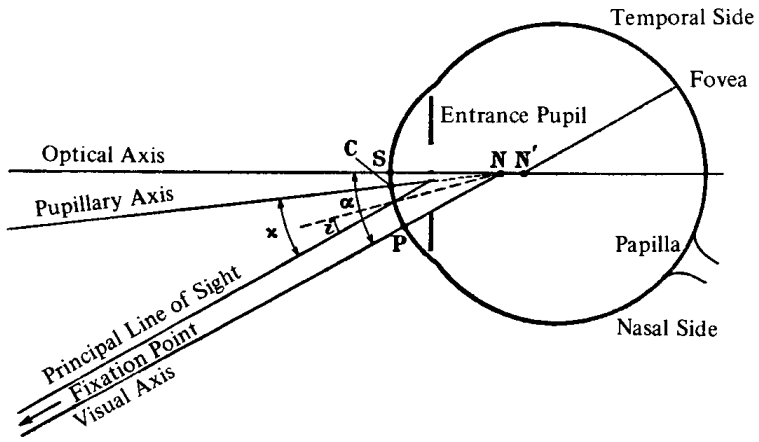

Abbildung C.2: Die verschiedenen Achsen des Auges, aus [Le Grand 1980]. Die Winkel sind stark übertrieben, um die Geometrie deutlicher zu machen.

C.3 Aufbau des Auges

Für dieses Darstellung ist das sogenannte theoretische Auge von Bedeutung, in dem die verschiedenen brechenden Oberflächen der Hornhaut (Cornea) sowie der Linse und die variable Blende der Pupille auf der optischen Achse angeordnet sind.

C.4 Achsen des Auges

Man unterscheidet beim theoretischen Auge zwischen wenigstens drei verschiedenen Achsen, die wie folgt definiert sind (die Lage der Achsen kann Abbildung C.2 entnommen werden.):

Die Optische Achse ist jene, auf der näherungsweise die verschiedenen berechenden Oberflächen des Auges, also von vorne nach hinten, die beiden Oberflächen der Cornea und die Oberflächen der Linse, angeordnet gedacht werden können.

Die Pupillenachse verläuft senkrecht durch die Oberfläche der Cornea und den Pupillenmittelpunkt.

Die visuelle Achse bezeichnet die Linie, die durch den Fixationspunkt und den Objektknotenpunkt verläuft. Sie trifft in der Fovea auf die Netzhaut.

Aufgrund dieser Darstellung ist ersichtlich, daß die optische Achse des Auges und die visuelle Achse nicht zusammenfallen. Der physiologische Grund hierfür ist die Abweichung der Lage der Fovea vom Schnittpunkt der optischen Achse mit der Netzhaut.

Dies führt vermutlich dazu, daß die tatsächliche Richtung der visuellen Achse um 4 bis 8 Grad von der Richtung der optischen Achse abweicht. Die unmittelbare Folge dieser in den Meßalgorithmen bislang nicht berücksichtigten Abweichung ist ein statischer Offset in den Winkelstellungen des Auges, der zu einem Fehler in der Kalibration und Messung führen wird.

Bislang ist die Größe dieses Fehlers nicht untersucht worden. Seine Behebung würde die Einführung eines weiteren Parameteres in der Kalibration erfordern, was vermutlich die Stabilität der numerischen Algorithmen verringert. Obwohl also vom theoretischen Standpunkt aus eine Kalibration der Abweichung der Blickrichtung von der optischen Achse einbezogen werden müßte, ist es sehr wahrscheinlich, daß der Fehler klein genug ist, ihn zunächst hintanzustellen, solange andere, größere Probleme noch der Lösung harren.

An einer Untersuchung zu seiner theoretisch zu erwartenden tatsächlichen Größe dürfte allerdings dennoch nichts vorbeiführen.

Anhang D

Exakte Darstellung der Augendrehung

Das im Hauptteil dieser Arbeit angenommene Modell für die Lage der Drehpunkte stellt eine Vereinfachung der tatsächlichen Verhältnisse dar.

Die Bewegung des Auges wird durch drei Muskelpaare erzeugt, die aus den in Abbildung D.1 gezeigten Richtungen am Auge angreifen. Zwei der Muskelpaar greifen dabei so von hinten an den Seiten des Auges an, daß ihre Kontraktion vertikale und horizontale Drehungen ermöglicht, das dritte Paar greift oben und unten von der Seite an und ermöglicht torsionelle Bewegungen.

Frühe überlegungen zur Drehmechanik des Auges, bis zurück auf Kepler und Scheiner[1], nahmen schon an, es gebe ein ausgezeichnetes Rotationszentrum des Auges, erste Untersuchungen[2] lokalisierten es am hinteren Pol des Auges. Volkmann[3] zeigte 1836, daß das Rotationszentrum nahe dem Zentrum der skleralen Sphäre liegen muß. 1862 maß Donders[Donders 1864] für den Abstand des Rotationszentrums vom Apex der Cornea einen Wert von 13,5 mm.

Genauere Studien in neuerer Zeit zeigten allerdings, daß die Annahme eines festen Rotationszentrums eine Näherung darstellt. Das Zentrum variiert bei tatsächlichen Bewegungen nach einer Untersuchung von Park[4] um etwa einen Millimeter, Fry und Hill veröffentlichen 1962 eine ausführliche Untersuchung, in der sie zeigten, daß es auf Grundlage ihrer Daten keinen festen Rotationspunkt geben kann.

Zur genauen Beschreibung der Bewegung des Auges müßte man daher

[1] [Kepler 1604] und [Scheiner 1619]
[2] [Müller 1825]
[3] [Volkmann 1936]
[4] [Park 1933]

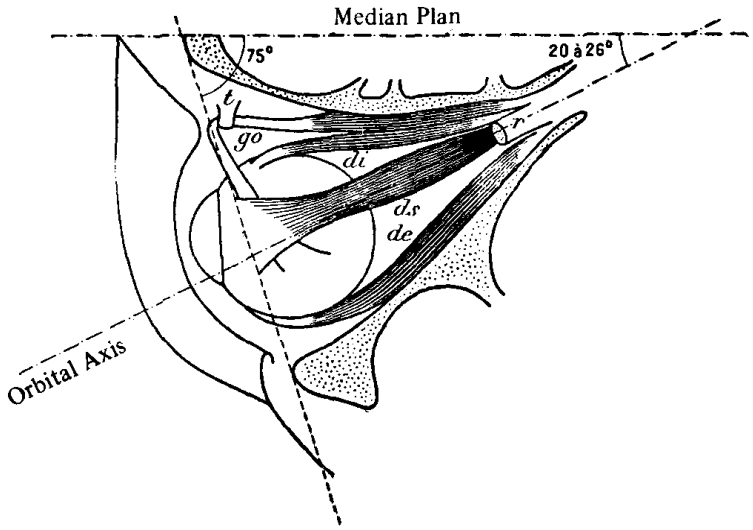

Abbildung D.1: Die Angriffsrichtung der Augenmuskeln. Aus: [Le Grand 1980]. Zu sehen ist das linke Auge in einer Ansicht von oben. Die sichtbaren Muskel sind dabei: *ds - superior rectus; di - medial rectus; de - lateral rectus; go - superior oblique; r - levator.* Nicht zu sehen sind die unter dem Auge liegenden *inferior rectus* und *inferior oblique* sowie der vordere Teil des aus Gründen besserer Einsicht entfernten *levator*, der für das Heben des oberen Augelides verantwortlich ist.

annehmen, daß jede elementare horizontale Drehung des Auges um ein momentanes Drehzentrum erfolgt, und daß der Ort dieser Drehzentren im Kopf eine Kurve ist, die man Zentrode nennt. Eine ähnliche Zentrode bezeichnet den Ort der Drehzentren in Bezug auf das Auge. Auch vertikale Bewegungen werden über solche Zentroden beschrieben.

Aus praktischen Gründen geht man in der Regel, auch aufgrund der nachgewiesen relativ geringen Schwankung der Position des Rotationszentrums[5], dennoch davon aus, daß ein solches Zentrum existiert und identifiziert seinen Ort mit dem Schwerpunkt der Verteilung aller Orte solcher Rotationszentren während Drehungen.

Um den Fehler dabei möglichst klein zu halten, und weil gezeigt wurde, daß das Rotationszentrum für vertikale Bewegungen deutlich verschieden von dem für horizontale Bewegungen ist, nimmt man für horizontale und vertikale Bewegungen unterschiedliche Rotationszentren an, die um einen Abstand d voneinander entfernt sind.

[5] [Donders 1864]

Anhang E

Programmcode

E.1 Kalibration

E.1.1 KALIB

```
function [x_ref,y_ref,r_auge,d,alpha]=kalib(p1,p2,p3,p4,geon)

% KALIB(x,y,Theta,Phi,geon) -> x_ref,y_ref,r,d,alpha
%
% Erwartet zwei Vektoren mit Pixel- bzw. Winkelkoordinaten
% der  Fixationspunkte und errechnet draus den Augenradius,
% den Parameter d fuer vertikale Augenbewegungen und die
% Koordinaten des Referenz-Punktes. Geon ist ein Flag
% fuer die Anwendung der geometrischen Naeherung
% unveraenderter Objektweiten;
%
% geon=0: korrekte Rechung. geon=1: Ebenennaeherung
% geon=2: Kugelkinematik (d=0)
%
%                                              KS, 02.03.1998
% nutzt        abstand, abweich
% genutzt von  test_kalib

% global, damit die Daten in den Subroutinen zur
% Verfuegung stehen, ohne uebergeben werden zu muessen
global Theta Phi x y

% Daten einlesen
x=p1;
y=p2;
Theta=p3;
Phi=p4;
punkte=[length(x) length(y) length(Theta) length(Phi)];
```

```
% Korrekte Eingabe?
if std(punkte) ~= 0
   msgbox('Ist nicht Ihr Ernst. Verschieden lange Listen?', ...
      'Eingabefehler','error');
   punkte
   x_ref=-1;
   y_ref=-1;
   r_eye=-1;
   d=-1;
   alpha=-1;
   return
end

% Anzahl der Datenpunkte
punkte=punkte(1);

% Startwerte festlegen
% Suche den Punkt, der dem Referenzpunkt am naechsten liegt
% (Mit exakter Abbildung)
[d,i]=min(abs(abstand(Theta,Phi,100,20,0)));
x_ref=x(i);
y_ref=y(i);

% r_auge und d auf den Mittelwert der Verteilung
r_auge=100;
d=20;

% Nelder--Mead--Optimierung
options=foptions;
options(14)=8000;
erg = fmins('abweich',[x_ref y_ref r_auge d],options,[],geon);

% Ergebnis auslesen
x_ref=erg(1);
y_ref=erg(2);
r_auge=erg(3);
d=erg(4);

% Schmeiss d weg fuer geon=2
if geon==2
   d=0;
end

% Berechne den Rotationswinkel
if punkte>1
   [dx,dy,dr] = abstand(Theta,Phi,r_auge,d,geon);
   beta = asin( (dx(2:punkte) .* (y(2:punkte)-y_ref) - ...
      dy(2:punkte) .* (x(2:punkte)-x_ref)) ./ ...
      sqrt( dr(2:punkte) .* ...
```

```
      ((x(2:punkte)-x_ref).^2 + (y(2:punkte)-y_ref).^2 )));
   alpha=mean(beta);
else
   alpha=0;
end
alpha=alpha/pi*180;
```

E.1.2 ABWEICH

```
function dv = abweich(v,geon)

% ABWEICH([x_ref y_ref R_auge d],geon) -> d
%
% Berechnet die Standardabweichung des Abstandes der Punkte
% in der Liste vom Referenzpunkt fuer einen gegebenen
% Parametersatz
%
%                                                  02.03.98 KS
% benutzt von     kalib
% nutzt           abstand

x_ref=v(1);
y_ref=v(2);
R_auge=v(3);
d=v(4);

global Theta Phi x y;

% Berechne delta x_i'
[x_diff,y_diff,dist] = abstand(Theta,Phi,R_auge,d,geon);

% Berechne die Standardabweichung der Delta
dv = sum( (dist - ( (x-x_ref).^2 + (y-y_ref).^2 )).^2 );
dv = abs(dv);
if R_auge<0
   dv=1000+dv*1000;
end
```

E.1.3 ABSTAND

```
function [x_bild,y_bild,abst]=abstand(Theta,Phi,R_auge,d,geon)

% ABSTAND(Theta,Phi,R,D,geon) -> x_bild, y_bild, abst
%
% Berechnet die Entfernung der Punkte vom Referenzpunkt,
% Parameter im Winkelmass. Geon ist gibt die verwendete
% Naeherung an.
%
% benutzt von     abweich, test_kalib
```

```
%                                                KS 02.03.98

% Winkel auf Bogenmass bringen
Theta=Theta./180.*pi;
Phi=Phi./180.*pi;

% Wenn Kugelkinematik gewuenscht wird, schmeiss d weg
if geon==2
   d=0;
   geon=0;
end

% Optische Konstanten, Augenradius ist 100 Pixel = 1,5 cm
g = 900; % entspricht 13,5 cm = 12 cm + 1,5 cm
b = 800/3;  % entspricht 4 cm

% Berechnung nach Modell in Kopfkoordinaten
x_kopf = sin(Theta) * R_auge;
y_kopf = sin(Phi) .* ( cos(Theta) * R_auge - d);

% Abbildung auf die Kameraebene
if geon==0
   % Keine Naeherung
   z_kopf = cos(Phi) .* cos(Theta) * R_auge + d * ( 1 - cos(Phi));
   x_bild = x_kopf .* (g-z_kopf)/b;
   y_bild = y_kopf .* (g-z_kopf)/b;
else
   % Ebenennaeherung
   x_bild = x_kopf .* (g-R_auge)/b;
   y_bild = y_kopf .* (g-R_auge)/b;
end
abst =  x_bild.^2 + y_bild.^2;
```

E.1.4 KALTEST

```
function kaltest(dp,name,flag,fehler)

% KALTEST(anzahl,name,flag,fehler)
%
% Fuehrt test_kalib mit einer variablen Anzahl von Punkten aus und zeigt
% den mittleren prozentualen Fehler an. Von einem bis zu 13 Datenpunkten
% wird dabei jeweils Anzahl mal kalibriert. Speichert die Fehler in der
% Datei name, wenn ein Dateiname uebergeben wird, sonst Ausgabe am
% Bildschirm. Der Parameter fehler wird auf die errechneten Pixelkoordinaten
% statistisch addiert.
%
% flag=1: Zufallsverteilung         flag=2: Im Raster
% flag=3: Zufall, speichern         flag=4: Im Raster, speichern
%
```

```
% See also     test_kalib
%                                          02.03.98 Kai Schreiber

global m_res

% Loesche die Variablen fuer die Fehler
m_res=[];
res=[];

if mod(flag,2)==1
   zuf=' zufaellige Verteilung';
else
   zuf=' Punkte im Raster';
end

fprintf(strcat('\nFehler +- %4.2f Grad,',zuf),fehler);
for nr_p=1:13
   fprintf('\n %3.0f \\\\',nr_p);
   fz(nr_p)=0;
   for t=1:dp
      fprintf('%2.0f, ',t);
      res(nr_p,t,:) = test_kalib(nr_p,flag,fehler);
   end
   for i=1:5
      m_res(nr_p,i) = sqrt(mean(res(nr_p,:,i).^2));
   end
end

if isempty(name)
   darst(1,13);
   darst(5,13);
   darst(7,13);

   fehl=1;
   fehl_wind=figure;
   while fehl ~= 0
      darstellen = inputdlg('Welche Punktanzahl:','Fehler anzeigen');
      figure(fehl_wind);
      if isempty(darstellen{1})
         darstellen{1}='0';
      end
      fehl=str2num(darstellen{1});
      if fehl ~=0 & fehl < 14
         plot(1:dp,res(fehl,:,1),'r',1:dp,res(fehl,:,2),...
            'g',1:dp,res(fehl,:,3),'b',1:dp,res(fehl,:,4),...
            'k',1:dp,res(fehl,:,5),'b:');
      end
   end
   close all;
```

```
else
   fid=fopen(name,'w');
   s=size(m_res);
   ausgabe=zeros(s(1),s(2)+1);
   ausgabe([1:s(1)],[2:s(2)+1])=m_res;
   ausgabe([1:s(1)],1)=[1:s(1)]';
   fprintf(fid,'\\subsubsection{Fehler $\\pm$ %4.2f Grad} \n',fehler);
   fprintf(fid,'\\begin{tabular}{|r||r|r|r|r|r|}\n\\hline\n');
   fprintf(fid,'Punkte & $x_{ref} (Px)$ & $y_{ref} (Px)$ &');
   fprintf(fid,'$R_{auge} (Px)$ & $d (Px)$ & $\\alpha (deg)$\\\\');
   fprintf(fid,'\\hline\\hline\n');
   fprintf(fid,'%2.0f&%7.2f&%7.2f&%7.2f&%7.2f&%7.2f\\\\\\hline\n',ausgabe');
   fprintf(fid,'\\end{tabular}');
   fclose(fid);
end


function darst(a,b)

global m_res

figure;
plot(a:b,m_res(a:b,1),'r-',a:b,m_res(a:b,2),'g-', ...
   a:b,m_res(a:b,3),'b--',a:b,m_res(a:b,4),'k:', ...
   a:b,m_res(a:b,5),'b:');
```

E.1.5 TEST_KALIB

```
function [AIP] = test_kalib(points,flag,fehler)

% TEST_KALIB(Punktzahl,Flag,Fehler)
%
% Berechnet zufaellige Saetze von Punkten fuer zufaellige Parameter und berechnet
% diese Parameter anschliessend mittels kalib; gibt einen Vektor zurueck, der den
% relativen Fehler dieser Parameter enthaelt
%
% Flag=1: Verwende zufaellige Punkte, nicht speichern
% Flag=2: Verwende Punkte im Raster, nicht speichern
% Flag=3: Zufaellige Punkte, speichern
% Flag=4: Punkte im Raster, speichern
%
% Der Parameter Fehler ist die Breite der Fixationsschwankung in Pixeln.
%
% Verwendet        kalib
%                                          Kai Schreiber, 1998

% Setze die Parameter
R=97+rand*6;
D=19.5+1*rand;
```

```
x_ref = 290 + rand*60;
y_ref = 170 + rand*60;
clear T_Theta T_Phi T_x T_y;
Ref_the = [0 5 -5  5 -5  0 10   0 -10 15 -15  15 -15];
Ref_phi = [0 5 -5 -5  5 10  0 -10   0 15 -15 -15  15];

speichern=0;
if flag>2
   flag=flag-2;
   speichern=1;
end

% Setze die Fixationsfehler des Probanden auf +-fehler, Normalverteilung
fehler_t= randn([points,1]) .* fehler;
fehler_p= randn([points,1]) .* fehler;

% Erzeuge die Fixationspunkte
if flag==1   % Zufaellige Lage im I. Quadranten
   T_Theta = rand([points,1]) .* 15;
   T_Phi = rand([points,1]) .* 15;
   T_Theta(1)=0;
   T_Phi(1)=0;
   % Sind Punkte in der Naehe von 0/0? Dann weg damit
   Off=T_Theta(2:points).^2+T_Phi(2:points).^2;
   [Omin,Opos]=min(Off);
   while Omin<1
      T_Theta(Opos+1)=30-rand(1) *60;
      T_Phi(Opos+1)=30-rand(1)*60;
      Off=T_Theta(2:points).^2+T_Phi(2:points).^2;
      [Omin,Opos]=min(Off);
   end
   % Verteile die Punkte auf die Quadranten
   for i=2:points
      switch mod(i,4)
      case 3
         T_Phi(i)=-T_Phi(i);
      case 0
         T_Phi(i)=-T_Phi(i);
         T_Theta(i)=-T_Theta(i);
      case 1
         T_Theta(i)=-T_Theta(i);
      end
   end
else         % Wie in den obigen Listen
   T_Theta = Ref_the(1:points)';
   T_Phi   = Ref_phi(1:points)';
end

% Berechne den Offset vom Referenzpunkt im
```

```
% ungedrehten Bild (inkl. Fehler)
[x_d,y_d,d] = abstand(T_Theta + fehler_t,T_Phi + fehler_p,R,D,0);

% Berechne die Lage der Punkte
T_x = x_ref + x_d;
T_y = y_ref + y_d;

% Drehwinkel Alpha = +-5 Grad
Alpha = 5 - rand*10;

% Rotiere um den Winkel Alpha um x_ref (s.u.)
[T_x,T_y] = rotiere(T_x,T_y,Alpha,x_ref,y_ref);

% Parameter merken
Vorgabe = [x_ref y_ref R D Alpha];

% Kalibriere exakt (geon=0)
[B(1),B(2),B(3),B(4),B(5)] = kalib(T_x,T_y,T_Theta,T_Phi,0);

% Fehler berechnen
AIP = Vorgabe - B;

% Protokolldatei speichern
if speichern==1
   fnm = findname(points);
   save(fnm,'T_*','B','x_ref','y_ref','R','D','Alpha');
end

function [xo,yo] = rotiere(x,y,phi,x0,y0)

% Rotiert eine gegebene Punktmenge um den Winkel phi

% Winkel in Bogenmass umrechnen
phi=phi/180*pi;

% An den Ursprung veschieben
x = x - x0;
y = y - y0;

% Rotieren
xo = cos(phi) .* x - sin(phi) .* y;
yo = sin(phi) .* x + cos(phi) .* y;

% Zurueckverschieben
xo = xo + x0;
yo = yo + y0;
```

E.1.6 DARSTELLEN

```
function darstellen

% DARSTELLEN
%
% Zeigt die Fehlerdateien von kaltest an und loescht sie
% auf Wunsch

clear T_x T_y T_Theta T_Phi B R D Alpha

ztake=0;
z=1;
fig=figure;
while z ~= 0
   if ztake == 0
      z=input('Welche Punktzahl?');
      if isempty(z)
         z=0;
      end
   else
      z=ztake;
      ztake=0;
   end
   [ndump,anzahl]=findname(z);
   fnum=min(1,anzahl);

   if fnum == 0 & z ~= 0
      disp('Keine Daten vorhanden!');
   end

   while fnum ~= 0
      load(strcat(num2str(z),'_',num2str(fnum),'.mat'));
      figure(fig);
      subplot(2,1,1);
      plot(T_x,T_y,'bo',x_ref,y_ref,'gx',B(1),B(2),'rx');
      subplot(2,1,2);
      plot(D,R,'gx',B(4),B(3),'rx');
      title(strcat(num2str(z),' Punkte, Satz Nr. ',...
         num2str(fnum)));
      f = input(strcat('Welcher Punktesatz (1-',...
         num2str(anzahl),') ?' ));
      if isempty(f)
         f=fnum+1;
         if fnum == anzahl
            ztake=z+1;
            f=0;
         end
      end
```

```
      fnum=min(anzahl,f);
   end
end
close(fig);
ant=input('Alle Dateien loeschen (123)? ');

if isempty(ant)
   ant=0;
end

if ant == 123
   for i=1:13
      nummer = 1;
      while nummer ~= 0
         [name,nummer]=findname(i);
         if nummer ~=0
            delete( strcat(num2str(i),'_',...
               num2str(nummer),'.mat' ) );
         end
      end
   end
   disp('Alles geloescht.');
end
```

E.1.7 FINDNAME

```
function [name,zahl] = findname(p)

% FINDNAME(n)
%
% Findet den naechsten nichtexistierenden Speicherdateinamen
% der Syntax n_zzz.out. Dabei ist n die Anzahl der Punkte und
% zzz eine laufende Nummer. Die Funktion gibt einen Dateinamen
% und die Anzahl der vorhandenen Dateien zurueck.
%                                                  KS 23.02.1998

zahl=0;
handle=0;
while handle ~= -1
   zahl=zahl+1;
   name=strcat(num2str(p),'_',num2str(zahl));
   handle=fopen(strcat(name,'.mat'));
   if handle ~= -1
      fclose(handle);
   end
end
zahl=zahl-1;
```

E.1.8 MESSTEST

```
function messtest(anzahl,geon,verf)

% MESSTEST(Anzahl,geon,verf)
%
% Fuehrt test_mess Anzahl mal aus und mittelt die Fehler
%
% Wenn geon=1 ist, wird dabei die Kalibration mit geometrischer Naeherung
% vorgenommen, fuer geon=0 wird exakt kalibriert
%
% verf=0/1: Ebenennaeherung/Torusgleichungen
%

clear C D A B;
global C D
close all;
[C,D] = test_mess(geon,verf);
if anzahl>1
   for i=2:anzahl
      [A,B]=test_mess(geon,verf);
      C=C+A;
      D=D+B;
   end
end
C=C/anzahl;
D=D/anzahl;
E=sqrt(C.^2+D.^2);
figure;
mesh(C);
axis tight;
xlabel('Phi');
ylabel('Theta');
if verf==0
   v=axis;v(6)=6;axis(v);
   print f_theta -dmfile
   print ../Bilder/f_theta.eps -deps
else
   print xf_theta -dmfile
   print ../Bilder/xf_theta.eps -deps
end
title('Fehler in Theta');
figure;
mesh(D);
axis tight;
xlabel('Phi');
ylabel('Theta');
if verf==0
   v=axis;v(6)=6;axis(v);
```

```
   print f_phi -dmfile
   print ../Bilder/f_phi.eps -deps
else
   print xf_phi -dmfile
   print ../Bilder/xf_phi.eps -deps
end
title('Fehler in Phi');
figure;
mesh(E);
axis tight;
xlabel('Phi');
ylabel('Theta');
if verf==0
   v=axis;v(6)=6;axis(v);
   print f_total -dmfile
   print ../Bilder/f_total.eps -deps
else
   print xf_total -dmfile
   print ../Bilder/xf_total.eps -deps
end
title('Mittlerer quadratischer Fehler ueber Winkel');
ausgabe=zeros(9,10);
ausgabe(:,1)=[0:5:40]';
ausgabe(1:9,2:10)=C(1:5:41,1:5:41);
gibaus(ausgabe,'fphi.tex','Fehler in Phi',verf);
ausgabe(1:9,2:10)=D(1:5:41,1:5:41);
gibaus(ausgabe,'ftheta.tex','Fehler in Theta',verf);
ausgabe(1:9,2:10)=E(1:5:41,1:5:41);
gibaus(ausgabe,'ftotal.tex','Fehlerbetrag',verf);

function gibaus(ag,n1,n2,v);

if v==1
   n1=strcat('x',n1);
end
fid=fopen(n1,'w');
fprintf(fid,'\\begin{figure}[p]\n\\centering\n');
fprintf(fid,...
   '\\begin{tabular}{|r||c|c|c|c|c|c|c|c|c|}\n\\hline\n');
fprintf(fid,...
 strcat('$\\phi\\backslash\\theta$ & 0 & 5 & 10 & 15',...
 ' & 20 & 25 & 30 & 35 & 40\\\\'));
fprintf(fid,'\n\\hline\\hline\n');
fprintf(fid,...
 strcat('%3.0f & %8.3f & %8.3f & %8.3f & %8.3f & %8.3f',...
 ' & %8.3f & %8.3f & %8.3f & %8.3f\\\\\\hline\n'),...
 ag');
fprintf(fid,'\\end{tabular}\n');
fprintf(fid,'\\caption{');
```

```
fprintf(fid,n2);
fprintf(fid,'}\n');
fprintf(fid,'\\end{figure}\n');
fclose(fid);
```

E.1.9 TEST_MESS

```
function [fehler1,fehler2] = test_mess(geon,verf)

% TEST_MESS(geon,verf)
%
% Legt Parameter fest, simuliert Fixation und kalibriert.
% Eine anschliessend auf Basis der urspruenglichen Parameter simulierte
% Messung wird mit den kalibrierten ausgewertet.
%
% geon: 0/1   mit/ohne geometrischer Naeherung bei Kalibration
% geon: 2     Kalibriere auf eine Kugel
% verf: 0/1 mit Ebenennaeherung/Torusgleichungen
%
%                                                  KS 08.09.98

% Setze die Parameter
R=97+rand*6;
D=19.5+1*rand;
x_ref = 290 + rand*60;
y_ref = 170 + rand*60;
clear T_Theta T_Phi T_x T_y;
T_Theta = [0 5 -5  5 -5  0 10   0 -10  15 -15  15 -15 20 ...
       20 -20 -20  30 -30 -30  30 40 -40 -40  40];
T_Phi =   [0 5 -5 -5  5 10  0 -10   0  15 -15 -15  15 20 ...
      -20  20 -20  30  30 -30 -30 40  40 -40 -40];

% Berechne den Offset vom Referenzpunkt im ungedrehten Bild
% (inkl. Fehler)
[x_d,y_d,d] = abstand(T_Theta,T_Phi,R,D,0);

% Berechne die Lage der Punkte
T_x = x_ref + x_d;
T_y = y_ref + y_d;

% Drehwinkel Alpha = +-15 Grad
Alpha = 15 - rand*30;

% Rotiere um den Winkel Alpha um x_ref (s.u.)
[T_x,T_y] = rotiere(T_x,T_y,Alpha,x_ref,y_ref);

% Kalibriere
[B(1),B(2),B(3),B(4),B(5)] = kalib(T_x,T_y,T_Theta,T_Phi,geon);
```

```
% Jetzt haben wir die Parameter mit normaler Kalibration in B,
% die mit Naeherung berechneten in C vorliegen

% Erzeuge Messpunkte von 0 bis 40 Grad in 1 Grad-Schritten
x=zeros(41,41);
y=x;
for th=0:40
   [x(th+1,:),y(th+1,:),d]=abstand(th*ones(1,41),0:40,R,D,0);
end
x=x + x_ref * ones(41,41);
y=y + y_ref * ones(41,41);

% Rotiere das Raster um x_ref/y_ref um den Winkel Alpha
for ph=0:40
   [x(:,ph+1),y(:,ph+1)]=rotiere(x(:,ph+1),y(:,ph+1),Alpha,x_ref,y_ref);
end

t=zeros(41,41);
p=t;

% Jetzt die Messungen ausfuehren
for ph=0:40
   [t(:,ph+1),p(:,ph+1)] = messe(x(:,ph+1),y(:,ph+1),...
      B(1),B(2),B(3),B(4),B(5),verf);
end

% Einheitsgradraster
einer = ones(1,41);
spalte=[0:40];
thnorm = spalte' * einer;
phnorm = thnorm';
fehler1=t-thnorm;
fehler2=p-phnorm;

fprintf(strcat('\nVorgabe   : x=%7.3f, y=%7.3f,',...
   ' R=%7.3f, d=%7.3f, a=%7.3f'),...
   x_ref,y_ref,R,D,Alpha);
fprintf(strcat('\nKalibriert: x=%7.3f, y=%7.3f,',...
   ' R=%7.3f, d=%7.3f, a=%7.3f\n'),...
   B(1),B(2),B(3),B(4),B(5));

function [xo,yo] = rotiere(x,y,phi,x0,y0)

% Rotiert eine gegebene Punktmenge um den Winkel phi

% Winkel in Bogenmass umrechnen
phi=phi/180*pi;

% An den Ursprung veschieben
```

```
x = x - x0;
y = y - y0;

% Rotieren
xo = cos(phi) .* x - sin(phi) .* y;
yo = sin(phi) .* x + cos(phi) .* y;

% Zurueckverschieben
xo = xo + x0;
yo = yo + y0;
```

E.1.10 MESSE

```
function [th1,ph1] = messe(x,y,xr,yr,r,d,a,verf)

% MESSE(x,y,x_ref,y_ref,R,D,Alpha)
%
% Ermittelt die zum Punkt x/y gehoerenden Winkelwerte
% anhand der Kalibrationsparameter
% verf=0 verwendet das Naeherungsverfahren mit der Ebene
% verf=1 rechnet mit den exakten Torusgleichungen

b = 800/3;
g = 900;
a=a/180*pi;

if verf==0
   F = (g - r)/b;
end
if verf==1
   xd = cos(a) .* (x-xr) + sin(a) .* (y-yr);
   yd = cos(a) .* (y-yr) - sin(a) .* (x-xr);
   % finde den Schnittpunkt der Geraden mit dem Torus
 abst=inline(strcat('abs((x.*b./(g-z)).^2+',...
  '(sqrt((y.*b./(g-z)).^2+(z-d).^2)+d).^2-r.^2)'),...
  'z','x','y','r','d','g','b');
   for i=1:length(x)
      z(i) = fmin(abst,0,r+0.01,FOPTIONS,xd(i),yd(i),r,d,g,b);
   end
   F = (g - z')/b;
end

th1 = asin( (cos(a) .* (x-xr) + sin(a) .* (y-yr)) ./ (F .* r));
ph1 = asin( (cos(a) .* (y-yr) - sin(a) .* (x-xr)) ./ ...
   ( F .* ( cos(th1) .* r - d ) ) );
th1 = th1 ./pi .* 180;
ph1 = ph1 ./pi .* 180;
```

E.2 Belichtungssimulation

E.2.1 RTRANS

```
function [r,phi_iris] = rtrans(x_ref,y_ref,r_auge,...
   d,x,y,theta,phi)

% RTRANS(x_ref,y_ref,r_auge,d,x,y,theta,phi)
%
% Berechnet die Polarkoordinaten eines Punktes auf dem Auge,
% dessen Pixel-Koordinaten auf dem Bildschirm x und y sind,
% wenn die Pupillenmitte in Geradeausposition auf x_ref/y_ref
% liegt und das Auge um theta und phi horizontal bzw. vertikal
% gedreht ist (ohne Kamerarotation)
% r_auge und d sind die geometrischen Parameter des Auges.
% Die Funktion liefert die Polarkoordinaten r und phi in einem
% Spaltenvektor [r;alpha].

cphir = ((x-x_ref) - sin(theta) .* r_auge) ./ cos(theta);
Z     = - sin(theta) .* cphir + cos(theta) .* r_auge - d;
sphir = ((y-y_ref) + sin(phi) .* Z) ./ cos(phi);

phi_iris = atan2(sphir,cphir);

if sin(phi_iris)==0
   r=cphir/cos(phi_iris);
else
   r=sphir/sin(phi_iris);
end
```

E.2.2 HINTRANS

```
function [x,y] = hintrans(x_ref,y_ref,r_auge,...
   d,r,phi_iris,theta,phi)

% HINTRANS(x_ref,y_ref,r_auge,d,r,alpha,theta,phi)
%
% Berechnet die Pixelkoordinate eines Punktes auf dem Auge,
% dessen Polar-Koordinaten auf dem Auge in Bezug zur
% Pupillenmitte r und alpha sind, wenn die Pupillenmitte
% in Geradeausposition auf x_ref und y_ref liegt und das
% Auge und theta und phi horizontal bzw. vertikal gedreht
% ist.
% r_auge und d sind die geometrischen Parameter des Auges.
% Die Funktion liefert die absoluten Koordinaten x und y.

Z = - sin(theta) .* cos(phi_iris) .* r + cos(theta) .*...
   r_auge - d;
x = x_ref + (cos(theta) .* cos(phi_iris) .* r + ...
```

```
   sin(theta) .* r_auge);
y = y_ref + (cos(phi) .* sin(phi_iris) .* r - sin(phi) .* Z);
```

E.2.3 INTENS

```
function muster = intens(r,alpha)

%  INTENS(r,alpha)
%
%  Diese Funktion wird von Simulrot aufgerufen und liest
%  ein Irispattern aus einer Frontalaufnahme eines Auges.
%  Fuer alpha werden Vektoren akzeptiert.
%
% KS 15.06.98
%

% Korrektes Format?
s=size(alpha);
if s(2)~=1
   display('Falsches Winkelformat');
   muster=zeros(length(alpha));
   return
end

global Auge

% Umrechnung in Pixelkoordinaten
x=r.*cos(alpha)+374;
y=-r.*sin(alpha)+265;

% Bestimmung des Pixels, in dem der Punkt liegt
xg=round(x+0.5);
yg=round(y+0.5);

% Auslesen
muster=zeros(length(alpha),1);
wichtung=muster;
for xl=-1:1
   for yl=-1:1
      wfkt=sqrt( (xg-x+xl).^2 + (yg-y+yl).^2 );
      wfkt=max([ wfkt' ; 0.01.*ones(1,length(wfkt)) ])';
      wichtung=wichtung + 1./wfkt;
      muster = muster + ...
         double(diag(Auge(yg+yl,xg+xl))) ./ wfkt;
   end
end

muster=muster./wichtung;
```

```
e = ones(length(alpha),1);
muster = muster - (e*min(muster));
muster = muster ./ (e*max(muster));
muster = muster - (e*mean(muster));
```

E.2.4 SIMULROT

```
function simulrot(Phi,Theta,d_phi,d_theta,name,steps,Xp,Bst,r)

% SIMULROT(Phi,Theta,d_phi,d_theta,name,Steps,Xp,Bogenstart,r)
%
% Simuliert eine Rotation eines idealen Auges und Aufnahmen
% mit Belichtungszeiten in denen sich das Bild signifikant
% aendert. Errechnet die Kreuzkorrelation des ausgelesenen
% Musters mit dem idealen Muster. Es werden waehrend der
% Belichtungszeit (Xp) mehrere Bilder errechnet und der
% Durchschnitt gebildet.
% Die Anzahl der Bilder ist Steps, die Geschwindigkeit
% des Auges waehrend der Aufnahme d_phi bzw. d_theta
% Phi und Theta bezeichnen die Position des Auges zur halben
% Belichtungszeit. Bogenstart ist der Winkel an dem das
% ausgelesene Muster starten soll, r der Radius.
%
% Alle Winkelparameter in Grad!

% Parameter festsetzen
x_ref=0;
y_ref=0;
r_auge=400;
d_auge=0;
ang_span=pi/4;
px_span=ang_span/0.003+1;
Bst=min(max(Bst,0),315);

% Rechne Grad nach Bogenmass
Phi=Phi/180*pi;
Theta=Theta/180*pi;
d_phi=d_phi/180*pi;
d_theta=d_theta/180*pi;
Bst=Bst/180*pi;

% Berechne Winkelaenderung waehrend der Belichtungszeit
D_theta=d_theta*Xp;
D_phi=d_phi*Xp;

% Kontrolle: Ablesen des Musters am ungedrehten Auge
alpha=[Bst:0.003:ang_span+Bst]';
px=[1:px_span];
basis(px)=intens(r,alpha);
```

```
plot(basis);

% Ermittlung der Koordinaten des Segments zur Zeit t=0
% (Mitte der Belichtung)
[x(px),y(px)]=hintrans(x_ref,y_ref,r_auge,...
   d_auge,r,alpha,Theta,Phi);

% Auslesen des Referenzbildes zur Zeit t=0
[r(px),w(px)]=rtrans(x_ref,y_ref,r_auge,...
   d_auge,x,y,Theta,Phi);
ref_muster=intens(r',w');

% Auslesen der steps Bilder waehrend
% der Belichtungszeit -xp/2...xp/2
muster=[];
steps=steps-1;
for bild=0:steps
   T=Theta-D_theta/2+bild*D_theta/steps;
   P=Phi-D_phi/2+bild*D_phi/steps;
   [r(px),w(px)]=rtrans(x_ref,y_ref,r_auge,d_auge,x,y,T,P);
   if isempty(muster)
      muster=intens(r',w');
   else
      muster=muster+intens(r',w');
   end
end
muster=muster./(steps+1);
save(name,'basis','ref_muster','muster');
```

E.2.5 SIMULPLAN

```
function SIMULPLAN(modus)

% SIMULPLAN(modus)
%
% Ruft Simulrot mit einer Reihe von Parametern auf, speichert
% die Ergebnisse und erlaubt das Betrachten der Dateien
%
% modus=1: Erzeugung & Abspeichern der Daten, keine Betrachtung
% modus=2: Datenbetrachtung, keine Berechnung, kein Abspeichern
% modus=3: Erzeugen, Abspeichern & Betrachten
%
% Die Ergebnisse werden im Verzeichnis simuldata abgelegt
% die Dateinamen sind phxx und thxx
% Dabei ist xx die vertikale Augengeschwindigkeit zum
% Zeitpunkt der Aufnahme

global Auge
Auge=imread('auge.bmp');
```

```
% Berechnung und Speicherung der Datensaetze, falls erwuenscht
if (modus~=2)
   [name,pfad]=uigetfile('*.mat',...
      'Bitte beliebige Datei im Zielpfad waehlen');
   cd(pfad);
      for d_winkel=0:10:300
         name1=strcat('th',num2str(d_winkel));
         name2=strcat('ph',num2str(d_winkel));
         simulrot(0,0,0,d_winkel,name1,20,0.018,-22.5,150);
         simulrot(0,0,d_winkel,0,name2,20,0.018,-22.5,150);
      end
end

% Betrachtung der Datensaetze, falls erwuenscht
if (modus~=1)
   clear all;
   close all;
   laden(1);
end
```

E.2.6 SIMAUSWERTEN

```
function simauswerten(flag,px);

% SIMAUSWERTEN(flag,px)
%
% Liest die Simulationsdurchlaeufe ein und ermittelt aus den
% Profilen der realen Muster deren stabile Breite; Ergebnis
% ist die Breite ueber der Geschwindigkeit fuer jede
% Augenstellung.
% Px ist die halbe Breite des ausgemessenen
% Bereichs fuer flag=2
%
% flag=0: gemachte Auswertung laden und darstellen
% flag=1: durchfuehren und und darstellen
% flag=2: durchfuehren auf Basis der Abweichung im Zentrum
%
% See also GUETEKURVE, FINDBEST, SIMULPLAN, WINKELBREITE

if flag==0
   [name,pfad]=...
      uigetfile('*.mat','Bitte Auswertungsdatei waehlen');
else
   [name,pfad]=uigetfile('*.mat',...
      'Bitte beliebige Datei im Datenpfad waehlen');
end

cd(pfad);
```

```
methode=flag-1;

if (flag~=0)
   if (flag==1)
      name=strcat('fbsimerg-err-',num2str(px));
   else
      name=strcat('fbsimerg-kum-',num2str(2*px+1));
   end
   % Alle Winkel, alle Geschwindigkeiten
   th=zeros(1,31);
   ph=th;
   fprintf('Geschwindigkeit: ');
   for d_theta=0:10:300
      name1=strcat('th',num2str(d_theta));
      name2=strcat('ph',num2str(d_theta));
      load(name1);
      % Referenzmuster ist das mittlere Drittel von ref_muster:
      m=fix(length(ref_muster)/3);
      th(d_theta/10+1)=winkelbreite(guetekurve(muster,...
         ref_muster(m+1:2*m)),methode,px);
      load(name2);
      ph(d_theta/10+1)=winkelbreite(guetekurve(muster,...
         ref_muster(m+1:2*m)),methode,px);
      fprintf('%3.0f, ',d_theta);
   end
   fprintf('\n');
   save(name,'th','ph');
end

load(name);
figure;
subplot(2,1,1);
plot([0:10:300],th(:)/262*45);
title(strcat('Auswertungsdatei:',name));
subplot(2,1,2);
plot([0:10:300],ph(:)/262*45);
```

E.2.7 GUETEKURVE

```
function position=guetekurve(muster,varargin)

% guetekurve(muster[,referenz])
%
% Teilt das Muster in drei Teile und berechnet die
% Kreuzkorrelation des mittleren Teiles mit jedem
% anderen gleich langen Ausschnitt des Musters. Gibt
% das Maximum der Kreuzkorrelation ber der Position
% des Anfangs des Musters zurck, d.h. die
% Autokorrelation liegt in der Mitte des Ergebnisvektors
```

```
%
% Wird ein Referenzmuster angegeben, so wird statt des
% mittleren Teiles dieses Referenzmuster verwendet.
%
% See also SIMAUSWERTEN

if isempty(varargin)
   auto=1;
else
   auto=0;
   ref_muster=varargin{1};
end

% Drittelbar machen, ml=Musterlaenge
ml=3*fix(length(muster)/3);
muster=muster(1:ml);

if auto==1
   % Referenzmuster reicht von ml/3+1 bis 2*ml/3
   start_ref=ml/3+1;
   end_ref=2*ml/3;
   ref_muster=muster(start_ref:end_ref);
end

if length(ref_muster)~=length(muster)/3
   display('Falsche Referenzmusterlnge');
   return;
end

% Berechne die Guetekurve
for musterstart=1:2*ml/3+1
   korr=xcorr(ref_muster,...
      muster(musterstart:musterstart+ml/3-1));
   [hoehe,lage]=paramax(korr');
   position(musterstart)=ml/3-lage;
end

px=[1:length(position)];
position = px - position - length(position)/2 - 0.5;
```

E.2.8 WINKELBREITE

```
function breite=winkelbreite(kurve,methode,n);

% WINKELBREITE(Guetekurve,Methode,n)
%
% Erwartet eine Guetekurve und bestimmt Qualitaet
% des gemessenen Muster durch
%
```

```
% Methode=0: Messen der fixen Breite (maximale Abweichung n)
% Methode=1: Messen der kumulativen Abweichung in den
%              mittleren 2*n+1 Pixeln

mitte=fix(length(kurve)/2);
breite=0;

if methode==0
   B1=mitte;
   B2=mitte;
   while (abs(kurve(B1))<n & B1>1)
      B1=B1-1;
   end
   while (abs(kurve(B2))<n & B2<length(kurve)*2)
      B2=B2+1;
   end
   breite=B2-B1;
else
   for p=mitte-n:mitte+n
      breite=breite+abs(p-kurve(p)-mitte-1);
   end
end
```

E.2.9 PARAMAX

```
function [h,p] = paramax(daten)

% PARAMAX(Muster)
%
% Ersetzt die eingebaute Funktion max. Ermittelt
% das Maximum eines Datensatzes durch Fitten einer
% Parabel ans Hauptmaximum
% (Vorsicht: bei nicht-parabolischen Maxima wird
% Unsinn berechnet)

% Bestimme geschaetzte Werte
[gh,gx] = max(daten);

% Parabelfit zwischen m1 und m2
m1=max(1,gx-7);
m2=min(length(daten),gx+7);
P = polyfit(m1:m2,daten(m1:m2),2);

% Keine Parabel gefittet/Daten=Konstante
if P(1)==0
   h=gh;
   p=gx;
   return
end
```

```
% Berechnung des Scheitelpunktes
p=-P(2)/(2*P(1));
h=P(3)/P(1)- P(2)^2/(4*P(1)^2);
```

E.2.10 LADEN

```
function laden(para)

% LADEN
%
% Ldt einen Satz von Simulrot-Daten und stellt ihn dar
%
% NICHT AUFRUFEN!
%
% Stattdessen simulplan(2) starten.

global basis muster ref_muster f_menu_handle skala pb;

if (para~=1)
   pb=para;
end

if isempty(pb)
   pb=4;
end

if (para==1)
   [name,pfad]=uigetfile('*.mat','Bitte auswhlen');
   cd(pfad);
   skala=[0:0.1719:45];
   load(name);
   laden(pb);
   if isempty(f_menu_handle)
      f_menu_handle=0;
   end
   if (f_menu_handle==0)
      f_menu_handle=uimenu('Label','Simulrot');
      uimenu(f_menu_handle,'Label',...
         'Laden','Callback','laden(1)');
      uimenu(f_menu_handle,'Label',...
         'Abziehen','Callback','laden(2)');
      uimenu(f_menu_handle,'Label',...
         'Xkorr','Callback','laden(3)');
      uimenu(f_menu_handle,'Label',...
         'Fourier (Betrag)','Callback','laden(5)');
      uimenu(f_menu_handle,'Label',...
         'Fourier (Phase)','Callback','laden(6)');
      uimenu(f_menu_handle,'Label',...
```

```
         'Muster','Callback','laden(4)');
      uimenu(f_menu_handle,'Label',...
         'Guetekurve','Callback','laden(7)');
   end
end

if (para==2)
   figure(1);
   subplot(3,1,1);
   plot(skala,basis'-ref_muster);
   title(strcat('Kontrolle: Abweichung des Referenzmusters',...
      ' vom Basismuster (sollte 0 sein)'));
   subplot(3,1,2);
   plot(skala,muster-ref_muster);
   title('Abweichung des Realmusters vom Referenzmuster.');
   subplot(3,1,3);
   plot(skala,(muster-ref_muster)./ref_muster);
   title('Abweichung relativ zum Muster');
end

if (para==3)
   figure(1);
   subplot(3,1,1);
   plot(xcorr(ref_muster));
   title('Autokorrelation Referenzmuster');
   subplot(3,1,2);
   plot(xcorr(ref_muster,muster));
   title('Kreuzkorrelation Referenzmuster<->Realmuster');
   subplot(3,1,3);
   plot(xcorr(ref_muster)-xcorr(ref_muster,muster));
   title('Differenz der Kreuzkorrelationen');
end

if (para==4)
   figure(1);
   subplot(3,1,1);
   plot(skala,basis);
   title('Basismuster');
   subplot(3,1,2);
   plot(skala,ref_muster);
   title('Referenzmuster');
   subplot(3,1,3);
   plot(skala,muster);
   title('Realmuster');
end

if (para==5)
   figure(1);
   subplot(3,1,1);
```

```
   plot(abs(fft(ref_muster)));
   title('Fouriertransformierte des Referenzmusters (Betrag)');
   subplot(3,1,2);
   plot(abs(fft(muster)));
   title('Fouriertransformierte des Realmusters (Betrag)');
   subplot(3,1,3);
   plot(abs(fft(ref_muster)/fft(muster)));
   title('Quotient der Fouriertransformierten (Betrag)');
end

if (para==6)
   figure(1);
   subplot(3,1,1);
   plot(angle(fft(ref_muster)));
   title('Fouriertransformierte des Referenzmusters (Phase)');
   subplot(3,1,2);
   plot(angle(fft(muster)));
   title('Fouriertransformierte des Realmusters (Phase)');
   subplot(3,1,3);
   plot(angle(fft(ref_muster)/fft(muster)));
   title('Quotient der Fouriertransformierten (Phase)');
end

if (para==7)
   figure(1);
   subplot(3,1,1);
   m=guetekurve(ref_muster);
   m1=length(m);
   plot(m,'b');
   hold on;
   plot(zeros(1,m1),'r');
   hold off;
   title('Guetekurve des Referenzmusters');
   subplot(3,1,2);
   po=fix(length(ref_muster)/3);
   m=guetekurve(muster,ref_muster(po+1:2*po));
   m1=length(m);
   plot(m);
   hold on;
   plot(zeros(1,m1),'r');
   hold off;
   title('Guetekurve des Realmusters');
   subplot(3,1,3);
   f=winkelbreite(m,0,1);
   title(num2str(f));
end
```

E.3 Musterfindung

E.3.1 HALBWERT

```
function halbwert(alpha,name)

% HALBWERT(alpha,name)
%
% errechnet aus einem Testbild eine Kurve der Halbwertsbreiten
% der Kreuzkorrelationen der Irismuster in einem durch den
% Winkelvektor Alpha gegebenen Segment (0 Grad liegen links,
% Zunahme des Winkels im Gegenuhrzeigersinn) Die Auflsung des
% Winkels wird direkt in Pixelauflsung bersetzt, sie ist also
% so zu whlen, da bis zu einem Radius von etwa 200 Pixeln eine
% vernnftige Musterauflsung gegeben ist.
%
% Speichert das Ergebnis im file mit dem Namen name
% und wird dann beendet.
%
% Wenn kein Name gegeben ist, wird ein Modul zur Darstellung der
% Ergebnisse aufgerufen, das das beste und schlechteste Muster
% darstellt und mittels testmove eine Qualittskurve errechnet.
%
% Verwendet intens zur Auslesung der Muster
%
% See also INTENS, TESTMOVE

% Festlegung des verwendeten Bildes und seiner Parameter
% geschieht in intens

muster = zeros(800,length(alpha));
if ~isempty(name)
   global Auge
Auge=imread('auge.bmp');
   for r=196:800
      muster(r,:)=intens(r/4,alpha')';
      corr(r,:)=xcorr(muster(r,:));
      fprintf('%2.0f\n',r);
   end
   save(name,'corr','muster');
   return
end

if isempty(name)
   wd=cd;
   [name,pfad]=uigetfile('*.mat','Bitte Datei whlen');
   cd(pfad);
   load(name);
   cd(wd);
```

```
end

% Jetzt haben wir in jedem fall muster und corr im speicher
[radii,pix]=size(muster);

% Nimm nur das mittlere Drittel zur Berechnung der Gte wg.
% anschlieenden Tests mit testmove
pa=fix(pix/3);
pe=2*pa;
for i=10:radii
   hb(i)=hwb(xcorr(muster(i,pa:pe)));
end
sr=200;
[w,mxp]=max(hb(sr:700));
[w,mnp]=min(hb(sr:700));
mxp=mxp+sr-1;
mnp=mnp+sr-1;

close all;
figure(1);
plot(hb);
figure(2);
subplot(2,1,1);
plot(muster(mxp,:));
title(num2str(mxp/4));
subplot(2,1,2);
plot(muster(mnp,:));
title(num2str(mnp/4));
figure(3);
subplot(2,1,1);
plot(xcorr(muster(mxp,pa:pe)));
subplot(2,1,2);
plot(xcorr(muster(mnp,pa:pe)));
figure(4);
subplot(2,1,1);
trans=fft(muster(mxp,:));
plot(abs(trans(1:round(length(trans)/2))));
subplot(2,1,2);
trans=fft(muster(mnp,:));
plot(abs(trans(1:round(length(trans)/2))));
figure(5);
subplot(2,1,1);
e1=guetekurve(muster(mxp,:)');
plot(e1);
hold on;
e2=length(e1)/2;
plot(zeros(1,e2*2),'r');
subplot(2,1,2);
plot(guetekurve(muster(mnp,:)'));
```

```
hold on;
plot(zeros(1,e2*2),'r');

function B=hwb(M)

M = M - min(M) * ones(1,length(M));
[T,P] = max(M);
B1=P;
B2=P;
while (M(B1)>T/2 & B1>1)
   B1=B1-1;
end
while (M(B2)>T/2 & B2<length(M))
   B2=B2+1;
end
B=B2-B1;
return
```

E.3.2 FINDBEST

```
function findbest(para)

% FINDBEST(para)
%
% Ldt ein von halbwert erzeugte Datei und
% berechnet die Gte des Musters anhand der
% Breite des stabilen Bereichs der Findung
% der Verschiebung der Musterdrittel gegeneinander.
% Steht in para eine 1 oder 2, werden die
% erzeugten Daten unter dem Namen der Quelldatei
% mit einem vorangestellten fb gespeichert, sonst
% wird nur eine alte Auswertung geladen.
%
% para-1 wird an winkelbreite als Methode weitergereicht.
% also: para=1: Messung der fixen Breite
%       para=2: Messung der kum. Abweichung
%
% See also GUETEKURVE

% lade eine Datei
if (para~=0)
   wd=cd;
   [name,pfad]=uigetfile('*.mat','Bitte Quelldatei whlen');
   cd(pfad);
   load(name);
   [radii,pix]=size(muster)
   for i=10:radii
      fprintf('%2.0f, ',i)
      hb(i)=winkelbreite(guetekurve(muster(i,:)'),para-1,0.5);
```

```
   end
   save(strcat('fb',name),'hb');
   cd(wd);
end
wd=cd;
[name,pfad]=uigetfile('fb*.mat','Bitte Ergebnisdatei auswhlen');
cd(pfad);
load(name);
load(name(3:length(name)));
[radii,pix]=size(muster);
pa=fix(pix/3);
pe=2*pa;
cd(wd);

% Bestes und schlechtestes Muster finden
sr=50;
[w,mxp]=min(hb(sr:140));
[w,mnp]=max(hb(sr:140));
mxp=mxp+sr-1;
mnp=mnp+sr-1;
close all;
figure(1);
plot(hb);
figure(2);
subplot(2,1,1);
plot(muster(mxp,:));
title(num2str(mxp));
subplot(2,1,2);
plot(muster(mnp,:));
title(num2str(mnp));
figure(3);
subplot(2,1,1);
plot(xcorr(muster(mxp,pa:pe)));
subplot(2,1,2);
plot(xcorr(muster(mnp,pa:pe)));
figure(4);
subplot(2,1,1);
trans=fft(muster(mxp,:));
plot(abs(trans(1:round(length(trans)/2))));
subplot(2,1,2);
trans=fft(muster(mnp,:));
plot(abs(trans(1:round(length(trans)/2))));
figure(5);
subplot(2,1,1);
e1=guetekurve(muster(mxp,:)');
e2=length(e1)/2;
plot(e1,'b');
title(winkelbreite(e1,0,0.5));
hold on;
```

```
plot(zeros(1,2*e2),'r');
subplot(2,1,2);
e1=guetekurve(muster(mnp,:)');
plot(e1,'b');
hold on;
plot(zeros(1,2*e2),'r');
title(winkelbreite(e1,0,0.5));
```

E.3.3 BUTTERN

```
function buttern(muster,flag)

% BUTTERN(muster,flag)
%
% Das uebergebene Muster wird verschiedenen
% Butterworth-Filtern (Hoch-, Tief- und Bandpass)
% unterworfen, und die messbare Winkelbreite
% der resultierenden Muster aufgezeichnet.
%
% Flag ist ein Vektor [t h b], dessen Elemente
% 1 sind, wenn die entsprechende Filtermethode
% angewandt werden soll, und 0 sonst.
%
% See also GUETEKURVE, WINKELBREITE, BUTTER, FILTER

close all;

% Tiefpass-Butterworth-Filterung
if flag(1)==1
   for Mn=0:20
      [b,a]=butter(10,Mn/20);
      filtermuster=filter(b,a,muster);
      tief(Mn+1)=winkelbreite(guetekurve(filtermuster),...
         0,1);
   end
   figure;
   plot([0:0.05:1],tief);
   title('Tiefpass');
end

% Hochpass-Filterung
if flag(2)==1
   for Mn=0:20
      [b,a]=butter(10,Mn/20,'high');
      filtermuster=filter(b,a,muster);
      hoch(Mn+1)=winkelbreite(guetekurve(filtermuster),...
         0,1);
   end
   figure;
```

```
   plot([0:0.05:1],hoch);
   title('Hochpass')
end

% Bandpass-Filterung
if flag(3)==1
   for Mn=0:17
      [b,a]=butter(10,[(Mn+1)/20 (Mn+2)/20]);
      filtermuster=filter(b,a,muster);
      band(Mn+1)=winkelbreite(guetekurve(filtermuster),...
         0,1);
   end
   figure;
   plot([0.05:0.05:0.90],band);
   title('Bandpass');
end
```

E.3.4 FILMCHEN

Abbildungsverzeichnis

1.1 Lage der Iris 9
1.2 Drehachsen des Auges 10
1.3 Iriskoordinaten 11
1.4 Kopfkoordinaten 12
1.5 Meßbrille 23
1.6 Projektion auf die Kameraebene 24
1.7 Ebene der geometrischen Näherung 28
1.8 Verwendetes Augenbild 32
1.9 Torsionsmessung durch Kreuzkorrelation 33

2.1 Betrag des Fehlervektors bei Näherung 42

3.1 Schema der Berechnung der Gütekurve 45
3.2 Irissegmente der Simulation 48
3.3 Musterübersicht, Bewegung senkrecht 51
3.4 Musterübersicht, Bewegung parallel 52
3.5 Musteranteile horizontal 53
3.6 Musteranteile vertikal 54

4.1 Halbwertsbreite über Winkelbreite 58
4.2 Halbwertsbreite über Winkelbreite 59

5.1 Pupillenmittelpunkts–Torus 63
5.2 Torus für extremes d 64

A.1 Fixationspunktraster 69
A.2 Pixelwichtung 71

B.1 Näherungsfehler in ϕ 80
B.2 Näherungsfehler in θ 81
B.3 Meßfehler in ϕ bei exakter Messung 82
B.4 Meßfehler in θ bei exakter Messung 83
B.5 Betrag des Fehlervektors bei exakter Messung 83

B.6 Muster bei 80 Pixeln Radius 84
B.7 Winkelbreite, Fehler unter 0.5 Pixel 86
B.8 Winkelbreite, Fehler unter 1 Pixel 86
B.9 Winkelbreite, Fehler unter 2 Pixel 87
B.10 Muster bei 132 Pixeln Radius 87
B.11 Winkelbreite, Fehler unter 0.5 Pixel 88
B.12 Winkelbreite, Fehler unter 1 Pixel 88
B.13 Winkelbreite, Fehler unter 2 Pixel 89
B.14 Winkelbreite gefilterter Muster/Tiefpaß 89
B.15 Winkelbreite gefilterter Muster/Hochpaß 90
B.16 Winkelbreite gefilterter Muster/Bandpaß 90
B.17 Winkelbreite gefilterter Muster/Bandpaß 91

C.1 Augengeometrie 94
C.2 Achsen des Auges 95

D.1 Die Augenmuskulatur 98

Literaturverzeichnis

[Bos 1994] Dos J E, De Graaf B; Ocular torsion quantification with video images, IEEE Trans. Biomed. Eng., vol. 41, pp 351–357.

[Daugman 1994] Daugman J G; Biometric personal identification system based on iris analysis, U.S. Patent No. 5,291,560. 01.03.1994.

[Donders 1847] Donders F C; Beitrag zur Lehre von den Bewegungen des menschlichen Auges. Holländ Beitr Anat Physiol Wiss 1; pp 104–145, 1847

[Donders 1864] Donders F C; On the Anomalies of Accomodation and Refraction of the Eye, London, 1964.

[Fick 1854] Fick A, Die Bewegungen des menschlichen Augapfels, Zeitschrift für rationelle Medizin, 4, pp. 109–128, 1854

[Fry 1962] Fry G A, Hill W W; The center of ratation of the eye, Am. J. Opt., Vol 39, No. 11, pp. 581–595, 1962.

[Groen 1998] Groen E, Bos J E, Nacken P F M, de Graaf B; Determination of Ocular Torsion by Means of Automatic Pattern Recognition, IEEE Transactions on biomedical engineering, ol. 43, No 5., pp 471–477, 1998

[Gullstrand 1909] Gullstrand A; Appendix II.3 The optical system of the eye, in: von Helmholtz H, Handbuch der physiologischen Optik, 1909

[Haslwanter 1995] Haslwanter T, Mathematics of Three–dimensional Eye-Movements, Vision Res., Vol 35, No. 12, pp. 1727–1739, 1995.

[Haslwanter-Moore 1995] Haslwanter T, Moore S T; A Theoretical Analysis of Three–Dimensional Eye Position Measurement Using Polar Cross–Correlation, IEEE Transactions on biomedical engineering, Vol. 42, No. 11, pp. 1053–1061, 1995.

[Helmholtz 1963] von Helmholtz H; Ueber die normalen Bewegungen des menschlichen Auges. Arch. Ophtalmol IX; pp 153–214, 1963

[Huber 1998] Huber A, Kömpf D; Klinische Neuroophtalmologie, Georg Thieme Verlag, Stuttgart–New York, 1998

[Kepler 1604] Kepler J; Paralipomena ad Vitellionem, Frankfurt, 1604.

[Le Grand 1980] Le Grand Y, El Hage S G; Physiological Optics, Springer–Verlag Berlin Heidelberg New York, 1980

[Moore 1996] Moore S T, Haslwanter T, Curthoys I S, Smith S T; A Geometric Basis for Measurement of Three-dimensional Eye Position Using Image Processing, Vision Res., Vol 36, No. 3, pp. 445-459, 1996

[Müller 1825] Müller J; Zur Physiologie des Gesichtssinnes, Leipzig, 1825.

[Nelder-Mead 1965] Nelder J A, Mead R; A Simplex Method for Function Minimization, Computer Journal, Vol. 7, pp. 308–313, 1965.

[Park 1933] Park R S, Park G E; The center of ocular rotation in the horizontal plane, Am. J. Phys. 104, pp. 545–551, 1933.

[Robinson 1963] Robinson D A; A method of measuring eye movement using a cleral search coil in a magnetic field. IEEE Transactions on Biomedical Electronics, BME-10, 137–145, 1963

[Scheiner 1619] Scheiner C; Oculus sive Fundamentum Opticum, Innsbruck 1619.

[SMI 1996] SensoMotoric Instruments GmbH, VOG for Windows, Three Dimensional Video-Oculography, Eye Movement Analysis System, Version 3.08., 1996

[Tweed 1996] Tweed D, Vilis T; Mathematics for Rotations. Im Internet unter http://www.physiology.uwo.ca/TweedWeb/frameset.htm, 1996

[Volkmann 1936] Volkmann A W; Neue Beiträge zur Physiologie des Gesichtssinnes, Leipzig, 1936.

[Wildes 1996] Wildes R P et al.; Automated, non-invasive iris recognition system and method, U.S. Patent No. 5,572,596. 5.11.1996.

[Wolff 1940] Wolff E, The anatomy of the eye and orbit (2nd edn.), London: H. K. Lewis & Co., 1940

[Ygge 1994] Ygge J, Lennerstrand G; Eye Movements in Reading, Wenner–Gren International Series: Vol. 64. Elsevier Science Ltd., 1994.

[Zee 1991] Zee D S, Leigh R J; The Neurology of Eye Movements, F.A. Davis Company, Philadelphia, 1991