



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Diplomarbeit

Sinazi Ramadani

Entwicklung eines Systems zur Steuerung des
Mauszeigers mit Hilfe von Augenbewegungen

Sinazi Ramadani

Entwicklung eines Systems zur Steuerung des
Mauszeigers mit Hilfe von Augenbewegungen

Diplomarbeit eingereicht im Rahmen der Diplomprüfung
im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Ing. Andreas Meisel
Zweitgutachter: Prof. Dr. rer. nat. Reinhard Baran

Abgegeben am 21. August 2008

Sinazi Ramadani

Thema der Diplomarbeit

Entwicklung eines Systems zur Steuerung des Mauszeigers mit Hilfe von Augenbewegungen

Stichworte

Eyetracking, Kreisdetektion, Hough-Transformation, Monitor-Kalibrierung, Pupillendetektion, Bresenham-Algorithmus

Kurzzusammenfassung

Im Rahmen dieser Diplomarbeit wird ein System zur Steuerung des Mauszeigers mit Hilfe von Augenbewegungen entwickelt. Im Vordergrund steht dabei die Ausarbeitung eines Verfahrens, welches die Augenbewegungen des Anwenders erfasst und die Position der Pupille ermittelt. Nach dem erfolgreichen Detektieren dieser Koordinaten wird der Mauszeiger an denjenigen Bildschirmpunkt bewegt, auf den der Anwender geschaut hat. Das Verfahren basiert auf den Grundlagen der Bildbearbeitung. Die Augenbewegungen werden von einer günstigen Webkamera aufgezeichnet.

Sinazi Ramadani

Title of the paper

Development of a system to control a cursor by eye movements

Keywords

Eye tracking, circle detection, Hough-Transformation, monitor calibration, pupil detection, Bresenham-algorithm

Abstract

Goal of this thesis is to develop a system to control a cursor by eye movements. In the center of attention is the creation of a method which identifies the user's eye movements and detects the position of the pupil. After the successful detection of these coordinates the cursor is being moved to that screenarea on which the user has looked at. The whole procedure is based upon image processing. A low-cost webcam is used to record the eye movements.

Danksagung

Besonders herzlich möchte ich mich bei meinem Betreuer Herrn Prof. Dr.-Ing. Andreas Meisel bedanken. Durch sein Engagement war es mir möglich, immer wieder gewinnbringende und anregende Diskussionen mit ihm zu führen. Hierbei erhielt ich durch sein umfangreiches Wissen und seinen großen Erfahrungsschatz wichtige Impulse und Tipps für meine Diplomarbeit. Gleichzeitig hat er mir viel Raum gelassen, um meine Gedanken und die Realisierung meiner Diplomarbeit selbstständig zu entwickeln und zu gestalten.

Ein weiterer Dank gilt meinem Zweitgutachter Herrn Prof. Dr. rer. nat. Reinhard Baran, der sich spontan und kurzfristig bereit erklärt hat, meine Diplomarbeit zu bewerten.

Den Ausgangspunkt meiner Diplomarbeit, das Kennenlernen der Anwender und Systeme in den Elbe-Werkstätten ermöglichte mir Jan Heidemann, dem ich an dieser Stelle ebenfalls herzlich danken möchte.

Außerdem danke ich herzlich meiner Frau, die mir tatkräftig und unterstützend zur Seite stand und meine Diplomarbeit mit viel Geduld korrigiert hat. Weiterhin möchte ich meiner Schwiegermutter danken, die die Zweitkorrektur übernommen und liebevoll meinen 3-jährigen Sohn betreut hat.

Mein letzter Dank geht an meine Probanden, die sich bereitwillig die Zeit für die Registrierung deren Augenbewegungen nahmen.

Inhaltsverzeichnis

Tabellenverzeichnis	7
Abbildungsverzeichnis	8
1 Einführung	10
1.1 Motivation zur Beschäftigung mit der Steuerung des Mauszeiger mit Hilfe von Augenbewegungen	11
1.2 Gliederung der Diplomarbeit	13
2 Bestehende Steuerungssysteme des Mauszeigers	15
2.1 Monitor- und kopfbasierte Systeme	15
2.2 Methoden und Funktionalität solcher Systeme	17
2.2.1 Augenbewegungen	18
2.2.1.1 Anatomischer Aufbau des Auges	18
2.2.1.2 Arten von Augenbewegungen	20
2.2.2 Der adäquate Reiz	21
2.2.3 Methoden der Augenbewegungsregistrierung	22
2.2.3.1 Videobasierte Erfassung	23
2.2.4 Tabellarischer Vergleich der Systeme	26
3 Anforderung, Signal und Bildmaterial	33
3.1 Anforderung	33
3.2 Eingangssignal und Methode der Augenbewegungsmessung	34
3.3 Bildmaterial und Interpretation	35
4 Konstruktives Detektieren der Pupille	42
4.1 Entscheidende Merkmale	43
4.2 Das Detektieren der Pupille	44
4.2.1 Vorverarbeitung und Segmentierung	44
4.2.2 Merkmalsextraktion	56
4.2.2.1 Breseham Algorithmus	62
5 Ermittlung der Position einer Augenbewegung	67

5.1	Kalibrierung und Ermittlung der Position auf dem Bildschirm einer Augenbewegung	67
6	Realisierung und diskretes Verfahren	70
6.1	Entwicklungsumgebung	70
6.2	Realisierung	72
6.3	Ergebnisse des Detektierens der Pupille	74
7	Resümee und Ausblick	79
7.1	Zusammenfassung	79
7.2	Ausblick	80
	Literaturverzeichnis	82

Tabellenverzeichnis

2.1	Bestandteile und Funktionen des PowerTalker-Systems	28
2.2	Bestandteile und Funktionen des HeadMouse-Systems	29
2.3	Bestandteile und Funktionen des Tracker One-Systems	31
2.4	Bestandteile und Funktionen des MyTobii-Systems	32
6.1	Rechner und Webkamera	74
6.2	Tabellarische Darstellung der Werte für das Resultat des ersten Probanden, siehe Abbildung [6.3 S. 77]	75
6.3	Tabellarische Darstellung der Werte für das Resultat des zweiten Probanden, siehe Abbildung [6.4 S. 78]	76
6.4	Tabellarische Darstellung der Werte für das Resultat des dritten Probanden, zweite Reihe, siehe Abbildung [6.5 S. 78]	76

Abbildungsverzeichnis

1.1	Gestaltung einer Excel-Tabelle von einer behinderten nicht sprechenden Person	12
2.1	Aufbau des Auges [Onmeda (2008)]	18
2.2	Augenmuskeln [Onmeda (2008)]	19
2.3	Entstehung des ersten Purkinje-Bildes und seine Verschiebung bei Rotation des Auges [Rickheit u. a. (2003)]	24
2.4	Cornea-Reflex-Methode a) Primärposition b) nach der Kopfverschiebung c) und d) nach Augenbewegungen [Metrovision (2008)]	24
2.5	Doppelte Purkinje Darstellung mit vier Referenzpunkten [Dresden (2008)]	25
2.6	a) Kopfbaisiertes System und b) Monitorbasiertes System	26
2.7	PowerTalker	27
2.8	Ein Beispiel aus ISAAC, wie man einen Satz codiert. [ISAAC (2008)]	27
2.9	Tower des HeadMouse-Systems	27
2.10	Die Kamera mit einem beweglichen Kopf und Halter des Tracker One-Systems	30
2.11	MyTobii	30
3.1	Brillengestell mit Webkamera	35
3.2	Eingangsbild	37
3.3	Mögliche 2D-Gitter mit ihrer Nachbarschaft	37
3.4	Fokussierung des Auges in die Mitte des Bildschirms	39
3.5	Fokussierung des Auges oben links des Bildschirms	40
3.6	Fokussierung des Auges oben rechts des Bildschirms	40
3.7	Fokussierung des Auges unten rechts des Bildschirms	41
3.8	Fokussierung des Auges unten links des Bildschirms	41
4.1	Ablaufdiagramm für das Detektieren der Pupille	42
4.2	Median-Filter bei einer 3x3 Filtermaske	46
4.3	Kantenbild des Bildes [4.2 S. 46]	46
4.4	Median-Filter bei einer 9x9 Filtermaske	47
4.5	Kantenbild des Bildes [4.4 S. 47]	47
4.6	Invertiertes Bild	48
4.7	Maximum Filter bei einer 5x5 Faltungsmaske	48
4.8	Maximum Filter bei einer 5x5 Faltungsmaske des Quell-Bildes [3.5 S. 40]	49

4.9	Erste und zweite Ableitung einer Funktion	51
4.10	Ablaufdiagramm des Canny-Operators	52
4.11	Gradbereiche zur Diskretisierung der Kantenorientierungen	53
4.12	Kantenbild des Bildes [3.6 S. 40] bei Faltungsmaske: 9x9, Variance: 9,9, Schwellenwert-Max: 0,01, Schwellenwert-Min: 0,8	54
4.13	Detektierter Kreis zum Kantenbild [4.12 S. 4.12] bei Faltungsmaske: 9x9, Variance: 9,9, Schwellenwert-Max: 0,01, Schwellenwert-Min: 0,8	54
4.14	Kantenbild des Bildes [3.6 S. 40] bei Faltungsmaske: 9x9, Variance: 9,9, Schwellenwert-Max: 0,05, Schwellenwert-Min: 0,8	55
4.15	Detektierter Kreis zum Kantenbild [4.14 S. 4.14] bei Faltungsmaske: 3x3, Variance: 0,5, Schwellenwert-Max: 0,5, Schwellenwert-Min: 0,1	55
4.16	Verknüpfung des Kantenbild mit dem Maximum-Bild	56
4.17	HT für Geraden	57
4.18	HT für Kreise mit dem jeweiligen Parameterraum	59
4.19	Detektierte Pupille des Quell-Bildes [3.4 S. 39]	61
4.20	Der Hough-Raum zur detektierten Pupille des Bildes [4.19 S. 61]	61
4.21	Kreis und fortgesetzte Pixelwahl beim Bresenham-Algorithmus	62
5.1	Detektierte Pupille des ersten Probanden	68
6.1	Integrierte Entwicklungsumgebung der Bildbearbeitung	70
6.2	Verfahren zum Detektieren der Pupille	77
6.3	Detektierte Pupille des ersten Probanden	77
6.4	Detektierte Pupille des zweiten Probanden	78
6.5	Detektierte Pupille des dritten Probanden	78

1 Einführung

„Die Lebenswelt sollte in einer Form gestaltet werden, in der der physische Mangel weniger oder gar nicht zur Behinderung wird.“ [KI->I (2008)]

Mit zunehmender Informationstechnologie wächst die Anforderung an Forscher und Entwickler zur Integration und Verbesserung der Lebenssituation von alten Menschen und Menschen mit Beeinträchtigung beizutragen. Diese kontinuierliche Anforderung wird von Forschern und Entwicklern schrittweise studiert und bewältigt, um unserem Lebensraum allgemeinnützlich innovative Produkte verfügbar zu machen.

Viele Menschen mit physischen Mängeln wie cerebrale Bewegungsstörungen oder Verletzungsfolgen, sind nicht in der Lage, kommerzielle Eingabegeräte wie die Maus und Tastatur eines Rechners zu bedienen.

Damit solche Menschen in der Zukunft besser unterstützt und physische Mängel weniger bzw. gar nicht zur Behinderung führen, wird in dieser Diplomarbeit nach einem System gesucht, dass die Steuerung des Mauszeigers mit Hilfe von Augenbewegungen ermöglicht. Die Steuerung des Mauszeigers bezieht sich in diesem Kontext auf die Anwendung einer beliebigen Software, bei der der Anwender mit Hilfe seiner Augenbewegungen den Mauszeiger steuert bzw. anwendet. Damit ein solches System realisiert werden kann, werden eine Kamera mit einer Halterung und ein adäquates Verfahren benötigt. Mit Hilfe der Kamera wird die Augenbewegung erfasst. Das Signal bzw. die erfassten Bilder werden einem Rechner zugeführt, der mittels des Verfahrens bestimmte Merkmale wie einen Kreis bzw. die Pupille detektiert. Nach dem erfolgreichen Detektieren der Pupille ist ein Referenzpunkt im zweiten dimensional Raum verfügbar, der durch die y- und x-Achse beschreibbar ist. Anhand der Kalibrierung eines Bildschirms und dem Referenzpunkt wird die Position der Augenbewegung ermittelt. Der Mauszeiger wird an diese Stelle, wo die ermittelte Position hinzeigt, hingesteuert.

Die Entwicklung des Verfahrens zum Detektieren der Pupille beinhaltet den Hauptkern dieser Diplomarbeit.

Solche Systeme, die die Steuerung eines Mauszeigers ermöglichen, sind in verschiedenen Bereichen zu finden, in denen eine Mensch-Maschine-Interaktion nötig ist. Einige Bereiche, in denen die Mensch-Maschine-Interaktion dringend benötigt wird, sind der medizinische und der soziale Bereich. Dort werden solche Systeme eingesetzt, um die Kommunikation der

schwerstbehinderten nicht sprechenden Menschen zu ermöglichen, indem die Bildschirm-tastatur bzw. Software durch Augenbewegungen angewendet wird. Auf diese Weise können diese Menschen, die keine konventionellen Eingabegeräte bedienen können, nur durch Ansehen einzelner Buchstaben ganze Wörter und Sätze zusammensetzen und sich dadurch eigenständig und selbstbestimmt artikulieren.

1.1 Motivation zur Beschäftigung mit der Steuerung des Mauszeiger mit Hilfe von Augenbewegungen

Mein Interesse an einer Steuerung des Mauszeigers mit Hilfe von Augenbewegungen wurde mit der Begegnung des Systems namens „PowerTalker“ geweckt. Dem System „PowerTalker“ begegnete ich durch einen Besuch in den Elbe-Werkstätten Hamburg. Dort wird dieses System vielseitig zur Kommunikation als auch für das didaktische Erlernen einer Sprache bezüglich nicht sprechender Personen eingesetzt. Ein weiteres System namens „Tracker One“, das ich mir anschaute, wird grundsätzlich für die Anwendung einer beliebigen Software auf einem Rechner eingesetzt. Diese beiden Systeme werden im zweiten Kapitel ausführlich beschrieben.

Der Umgang mit dem Rechner bzw. die Anwendung eines Programms der oben genannten Personen hinterließ beeindruckende Bilder bei mir. Trotz ihrer geistigen Behinderung, Sprach- und/oder Bewegungsstörungen zeigten die Anwender erstaunlich gute Leistungen bei der Anwendung und dem Umgang mit dem Mauszeiger. Unten in der Abbildung [1.1 S. 12] ist eine Excel-Tabelle zu sehen, die in den Elbe-Werkstätten Hamburg von einer nicht sprechenden körperlich und geistig eingeschränkten Person gepflegt und erweitert wird. Anhand seiner noch möglichen Kopfbewegungen steuert er den Mauszeiger und bedient das Excel-Programm.

Der Wunsch dieser Person, die Software eines Rechners statt mit Kopfbewegungen mit Augenbewegungen zu nutzen und damit die Kopfanstrengung bzw. Nackenschmerzen zu vermeiden, ist sehr groß. Leider werden aus Kostengründen trotz der fortschrittlichen und leistungsfähigen Technologie solche Personen geringfügig unterstützt, da in diesem Bereich finanzielle Mittel minimal zur Verfügung gestellt werden.

Daher gilt meine Motivation der Schaffung eines solchen Systems, das die Steuerung des Mauszeigers mit Hilfe von Augenbewegungen übernimmt. Hierbei sollen günstige Webkamera benutzt werden, damit die Kosten möglichst gering gehalten werden. Damit steigt die Hoffnung, dass diese Menschen bald kostengünstig solche Systeme anwenden können.

Ein anderer Anhaltspunkt, der mich auf diesem Gebiet interessiert, ist die Registrierung sowie das Verfolgen eines Objektes, wie z.B. Auge, Gesicht etc.. Heutzutage - mit zuneh-

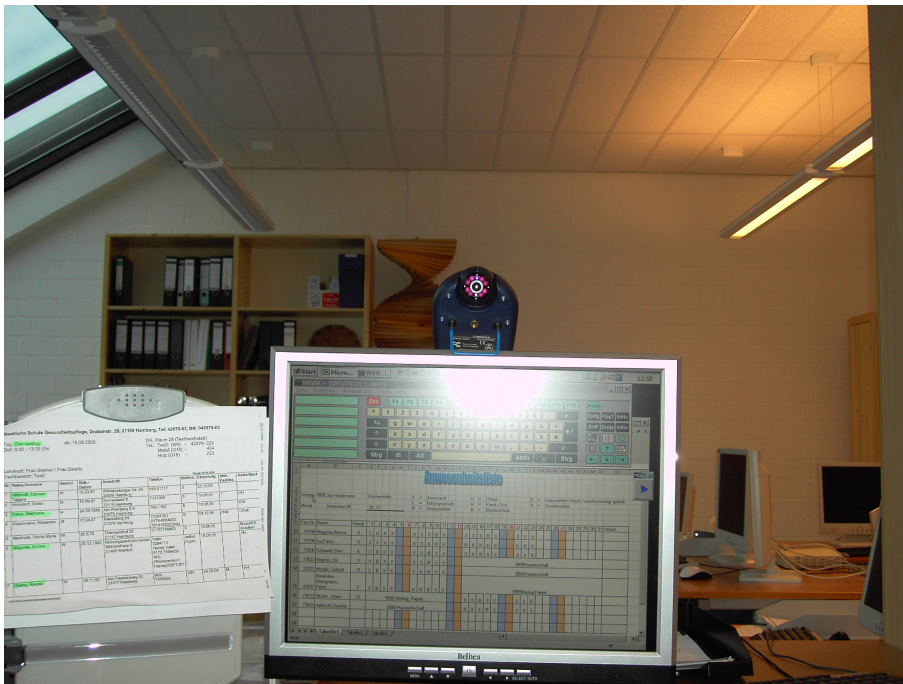


Abbildung 1.1: Gestaltung einer Excel-Tabelle von einer behinderten nicht sprechenden Person

menden technologischen Fortschritten - ist in vielen Industriegebieten das Registrieren und Verfolgen von Objekten anzutreffen.

Ein weiterer Anreiz dieser Diplomarbeit ist die Beschäftigung mit der Mensch-Maschine-Interaktion unter dem Aspekt „Programmieren, Schreiben ohne Maus und Tastatur“. Inwieweit lässt sich durch die Steuerung des Mauszeigers mit Hilfe der Augenbewegung und Spracherkennung der Rechner anwenden? Inwieweit lassen sich hiermit die rein kognitiven Prozesse umsetzen, und wie schnell lässt sich ein Programm schreiben? Werden wir bald keine Tastatur und Maus mehr benötigen und damit die schmerzhaften Symptome wie Verspannungen vermeiden können?¹

¹Nach dem Wissenschaftsmagazin „New Scientist“ [GOLEM (2006)] leiden 22 Prozent aller amerikanischen Programmierer an dem RSI-Syndrom (Repetitive Strain Injury Syndrom), ein komplexes Krankheitsbild mit Schmerzen des Armes und der Hand, das auf Grund einer chronischen und wiederholten Schädigung des Bewegungsapparates von Hand, Arm, Schulter und Nackenbereich entsteht. [med. Thomas Wallny (31.10.2007)]

1.2 Gliederung der Diplomarbeit

1. Kapitel: Am Anfang dieses Kapitels wurde die Problematik vieler Menschen mit physischen Mängeln, die kommerzielle Eingabegeräte nicht benutzen können, angedeutet. Anschließend wurden Beispiele für den Einsatz solcher Systeme erläutert, bei denen der Mauszeiger mit Hilfe von Kopf- oder Augenbewegungen gesteuert wird. Weiterhin wurde der Schwerpunkt dieser Diplomarbeit, die Entwicklung eines Verfahrens, dargelegt. Zum Schluss wurde auf die Motivation des Autors eingegangen, sich in der Diplomarbeit mit dieser Thematik zu beschäftigen.

2. Kapitel: Im zweiten Kapitel werden die Systeme beschrieben, die in der Lage sind, den Mauszeiger mit Hilfe einer Bewegung des Kopfes oder des Auges zu steuern. Am Anfang wird auf die Konstruktionsarten solcher Systeme eingegangen. Danach werden die Systeme tabellarisch dargestellt und verglichen. Dazu werden vorher kurz die Begriffe erklärt, die unerlässlich für das Verstehen des tabellarischen Vergleichs sind, wie die Anatomie des Auges, Augenbewegungen, der adäquate Reiz und die Methoden der Augenbewegungsregistrierung.

3. Kapitel: Dieses Kapitel beschäftigt sich mit der Anforderung sowie der Konstruktion des zu entwickelnden Systems. Hierbei werden die Methode für die Augenbewegungsregistrierung als auch das Signal mit der dazugehörigen Halterung dargestellt. Im Anschluss daran wird auf das erzeugte Bildmaterial eingegangen, und es werden wichtige Grundlagen dazu erwähnt.

4. Kapitel In diesem Kapitel wird das Verfahren zum Detektieren der Pupille konstruktiv aufgebaut. Hierbei werden die benötigten Grundlagen der Bildbearbeitung beschrieben. Anhand eines Ablaufdiagramms werden das Herangehen und die Entwicklungsschritte erklärt. Außerdem werden die dazugehörigen Operatoren sowie Algorithmen angegeben. Hierzu werden die Ziele sowie das jeweilige Ergebnis mittels Bilder dargelegt und besprochen.

5. Kapitel Das Kapitel befasst sich mit der Ermittlung der Position einer Augenbewegung. Mit der Position ist der Bildschirm gemeint, auf die der Anwender während der Anwendung schaut. Hierbei wird eine Beziehung zwischen der Position der Pupille und den Kalibrierungspunkten des Bildschirms hergestellt, und an diese ermittelte Position wird der Mauszeiger hinbewegt bzw. dort positioniert.

6. Kapitel In diesem Kapitel wird auf die Realisierung des Systems eingegangen. Hierzu wird die Entwicklungsumgebung mit dem jeweiligen diskreten Verfahren erläutert. Innerhalb des Verfahrens werden die Makros mit den Eingangs-, Ausgangs- und Einstellungsparametern kurz beschrieben. Am Schluss dieses Kapitels werden die Testergebnisse des Verfahrens angeführt und dargestellt.

7. Kapitel Das Kapitel beinhaltet die Zusammenfassung der Realisierung des Systems und den Ausblick. Hierbei werden das Ergebnis und die Erweiterung des System diskutiert.

2 Bestehende Steuerungssysteme des Mauszeigers

Ziel dieses Kapitels ist es, einige bestehende Systeme, die die Steuerung des Mauszeigers ermöglichen, zu analysieren, um die Funktionalität solcher Systeme zu erkennen. Zunächst werden in diesem Kapitel die verschiedenen Systeme und deren Hardware erwähnt. Mit Hilfe der Hardware wird die Funktionsweise der Systeme beschrieben. Hierbei wird auf die bestehenden Methoden eingegangen, die die Erfassung der Augen- und Kopfbewegungen ermöglichen. Weiterhin wird der Zusammenhang zwischen der erfassten Augen- und Kopfbewegung und der Steuerung des Mauszeigers analysiert. Anhand der tabellarischen Darstellung werden die Systeme verglichen sowie deren Vor- und Nachteile erläutert.

2.1 Monitor- und kopfbasierte Systeme

In der Literatur [[Duchowski \(2003\)](#) S. 55], [[Rickheit u. a. \(2003\)](#) S. 147] sowie bei den Herstellern werden verschiedene Begriffe verwendet, um die Systeme zu unterscheiden. In dieser Arbeit wird in monitorbasierte und kopfbasierte Systeme unterschieden, da diese Unterteilung für das Verständnis der verschiedenen Systeme im Hinblick auf das zu entwickelnde Verfahren am sinnvollsten ist.

Bei den kopfbasierten Systemen muss der Anwender vor und während der Anwendung eine der jeweiligen Hardware-Komponenten an den Kopf bzw. an das Auge anbringen. Diese Systeme können weiterhin auf Grund ihres Eingangssignals in kopfbasierte Systeme mit Infrarotlichtsignal und in kopfbasierte Systeme mit Videosignal (Bild) unterteilt werden.

Das System mit dem Infraroteingangssignal verfügt als Hardware-Komponente über einen Infrarotstrahler und einen Infrarotempfänger, wie auf der Abbildung [[2.7 S. 27](#)] dargestellt ist. Der Infrarotstrahler wird am Kopf durch jeweilige Halterungen angebracht, sodass das ausgestrahlte Infrarotlicht vom Infrarotempfänger im vorgegebenen Empfangsfeld abgefangen werden kann. Der Infrarotempfänger ist in dem PowerTalker integriert. Durch Kopfbewegungen wird der PowerTalker veranlasst, den Mauszeiger zu steuern.

Das andere System, das das Infrarotlicht ebenfalls als Eingangssignal verwendet, wie auf der Abbildung [2.9 S. 27] dargestellt ist, verfügt über eine andere Komponente, nämlich einen Reflektorpunkt. Dieser Reflektorpunkt dient der Reflexion des ankommenden Infrarotlichts. Das Infrarotlicht wird von dem Tower des HeadMouse-Systems ausgestrahlt. Der Tower verfügt über einen integrierten Infrarotempfänger, womit das reflektierte Infrarotlicht empfangen wird. Das weitere Nachfolgersystem „Tracker One“ verwendet als Eingangssignal das Videosignal, wie auf der Abbildung [2.10 S. 30] dargestellt ist. Dieses System verfügt über eine Kamera, bei der das Objektiv mit Infrarotlicht LED bestückt ist. Bei diesem System wird ebenfalls ein Reflektorpunkt benutzt, der durch das Infrarotlicht aufleuchtet. Der aufgeleuchtete Reflektorpunkt wird von der Kamera als heller Punkt erfasst. Dieser Reflektorpunkt kann auf der Stirn oder an der Brille angebracht werden. Durch Kopfbewegungen wird veranlasst, dass der Mauszeiger gesteuert wird.

Die monitorbasierten Systeme sind dadurch von den kopfbasierten Systemen zu unterscheiden, als dass der Anwender während der Anwendung keine zusätzlichen Hardware-Komponenten benötigt. Dieses System basiert nur auf dem Videosignal, das als Eingangssignal benutzt wird. Als Hardware-Komponente ist hier eine Kamera zu erwähnen, die im Bildschirm integriert oder auf dem Bildschirm positioniert ist. So ein System, in dem die Kamera integriert ist, ist auf der Abbildung [2.11 S. 30] namens „MyTobii“ zu sehen. Die Kamera muss in diesem Zusammenhang so positioniert sein, dass die Bewegung des Auges im Blickfeld des Objektivs erfasst werden kann. Dementsprechend existiert hierfür ein Erkennungsbereich, innerhalb dieses die Augenbewegung registriert wird. Hierbei ist zu betonen, dass der Anwender über einen gewissen Komfort als auch eine gewisse Einschränkung während der Anwendung verfügt. Mit dem Komfort ist gemeint, dass der Anwender sofort, sobald er sich im Erkennungsbereich befindet, die Steuerung des Mauszeigers mit Hilfe der Augenbewegung übernehmen kann. Die Einschränkung bezieht sich darauf, dass der Anwender den vom Hersteller angegebenen Erkennungsbereich während der Anwendung konstant einhalten muss. Weiterhin ist hierzu zu erwähnen, wie der Hersteller angegeben hat, dass schnelle Augenbewegungen nicht erfasst werden können.

Bezogen auf diese Diplomarbeit bieten die Systeme mit dem Videoeingangssignal zentrale Anhaltspunkte, sodass im Folgenden auf deren Funktionsweise eingegangen wird, um Erkenntnisse zu gewinnen.

Die Systeme mit dem Videoeingangssignal haben ein gemeinsames Ziel: Das Erfassen der Richtung der Augen- und Kopfbewegungen, in die der Anwender während der Anwendung den Mauszeiger steuern und positioniert haben möchte. Dies wird durch eine Kamera anhand der Kopf- und Augenbewegungen registriert. Aus den registrierten Daten (Bildern) wird mittels Bildverarbeitungsverfahren das gesuchte interessante Merkmal detektiert. Nach dem erfolgreichen Detektieren wird eine Position berechnet, die das gesuchte Merkmal als Referenzpunkt darstellt.

Grundsätzlich, bevor solche Systeme angewendet werden, wird eine Kalibrierung des Bildschirms benötigt. Diese kann automatisch oder durch Eingabe des Anwenders erfolgen. Mit den Kalibrierungsdaten und dem Referenzpunkt wird die gewünschte Zielposition des Anwenders auf dem Bildschirm ermittelt. Die ermittelte Zielposition wird durch x- und y-Werte beschrieben, sodass der Mauszeiger zu dieser Stelle hinbewegt und dort positioniert wird.

Die Steuerung, wie schon erwähnt wurde, kann je nach System durch einen Reflektorpunkt oder durch die Augenbewegung veranlasst werden. Bei dem System „Tracker One“ wird die Steuerung des Mauszeigers durch die Kopfbewegung veranlasst. Nachdem der Reflektorpunkt erfolgreich detektiert ist, wird die Zielposition mittels der Bildbearbeitungsverfahren ermittelt. Bei dem System „MyTobii“ wird die Steuerung durch die Augenbewegung veranlasst. Hier wird je nach Verfahren die Pupille detektiert, sodass darauffolgend die Zielposition ermittelt wird. Nach einer gewissen Zeit kann der positionierte Mauszeiger ein Ereignis, wie z. B. Mausklick auf das Objekt (Button), auslösen.

Die Verfolgung der Bewegungen eines Auges wird allgemein mit dem Begriff „Eyetracking“¹ beschrieben. Eyetracking fungiert als Grundbaustein der Steuerung eines Mauszeigers durch die Augenbewegungen. In der Literatur werden die Begriffe „Augen- als auch Blickbewegungen“ verwendet. In der vorliegenden Diplomarbeit wird der Begriff „Augenbewegungen“ benutzt.

2.2 Methoden und Funktionalität solcher Systeme

Die Funktionalität solcher Systeme ist ein wichtiger Aspekt, um zu verstehen, wie die Konstellation des Eingangssignals (Bild) im Zusammenhang mit dem Infrarotlicht steht. Weiterhin ist es hierzu erforderlich zu verstehen, welche Methoden bezüglich der Messung der Augenbewegungen angewendet werden und welche Referenzpunkte als Parameter dienen. Außerdem muss erwähnt werden, dass es mit einigen Systemen, wie auf der Abbildung [2.11 S. 30], nicht möglich gewesen ist, explizite Untersuchung durchzuführen, sodass sich mit Hilfe der Herstellerangaben an die Funktionalität dieses Systems herangetastet wird.

Damit auf die Zusammenhänge dieser Konstellationen eingegangen werden kann, sind einige Vorkenntnisse erforderlich, die nun dargestellt werden. Zuerst wird auf die Anatomie des Auges sowie auf die Arten der Augenbewegungen eingegangen. Danach wird kurz der adäquate Reiz erwähnt, damit deutlich wird, welches Licht das Auge nicht sehen kann

¹„Unter Eyetracking (ET), zu Deutsch Blickregistrierung, versteht man Verfahren, die den Blickverlauf einer Person beim Betrachten eines Bildes registrieren und festhalten. Das heißt, die Bewegung des Auges über eine Bildfläche wird ebenso festgehalten wie die Fixationsdauer bestimmter Punkte. Es wird also gemessen, in welcher Reihenfolge, wann und wie lange die Versuchsperson einen bestimmten Bereich des Bildes betrachtet.“ [Fakultät (2008)]

und damit als nicht störend empfunden. Daran anknüpfend werden die Methoden der Augenbewegungsregistrierung beschrieben, um konkrete Aussagen über die Funktionalität der Systeme zu ermöglichen. Die Augenbewegungsregistrierung basiert auf den anatomisch-physiologischen Eigenschaften des Auges. Diese Methoden liefern wichtige Erkenntnisse über den Stand und die Entwicklung der Augenbewegungsregistrierung. In dem anatomisch-physiologischen Bereich wird seit längerer Zeit über das Verhalten und die Bewegungen des Auges geforscht, sodass Eyetracking auf diesem Gebiet ein wesentlicher Bestandteil ist.

2.2.1 Augenbewegungen

2.2.1.1 Anatomischer Aufbau des Auges

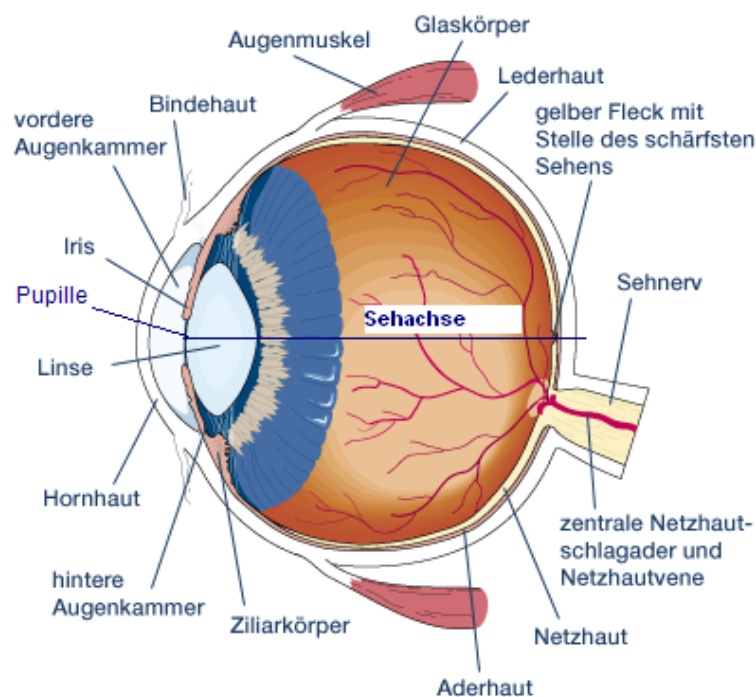


Abbildung 2.1: Aufbau des Auges [Onmeda (2008)]

Das Auge gehört zu den wichtigsten Organen, um visuelle Informationen aufzunehmen. Das menschliche Auge besteht grob aus:

- dem Augapfel (Bulbus oculi)
- Schutzvorrichtungen (Augenhöhle, Augenlider, Bindehaut, Tränenapparat)

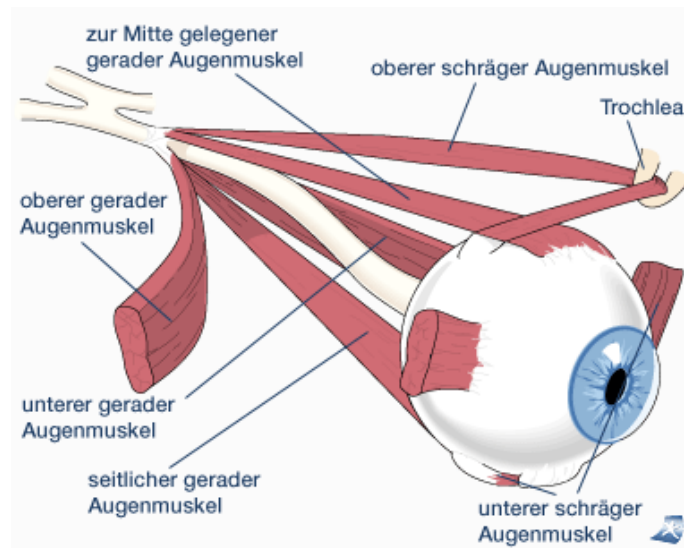


Abbildung 2.2: Augenmuskeln [Onmeda (2008)]

- dem Bewegungsapparat
- dem Sehnerv

Nach [Schmidt u. a. (2003) S. 370-371], [Mehrle (2003) S. 1-6] liegt der Augapfel, der ungefähr die Form einer Kugel hat und sich auf der horizontalen Mittellinie des Auges befindet, in der Augenhöhle (Orbita). Das Auge wird in der Augenhöhle, die durch sieben angrenzende Schädelknochen gebildet ist, vor äußerer Gefahr geschützt. Der Augapfel ist durch sechs Augenmuskeln beweglich. Die sechs Augenmuskeln bestehen aus vier geraden und zwei schrägen, die paarig arbeiten. Das heißt, dass jeweils ein Augenmuskel den Zug des anderen ausbalanciert. Der obere und der untere Augenmuskel ist für die Augenbewegung in die entsprechenden Richtungen zuständig. Die Seitwärtsbewegungen werden durch den inneren und äußeren Augenmuskel getätigt, während das Rotieren des Augapfels auf die beiden schrägen Muskeln zurückzuführen ist.

Die vordere Augenkammer (Vorderkammer) befindet sich im vorderen Teil des Auges hinter der durchsichtigen Hornhaut (Cornea) mit dem Kammerwasser. Vor der vorderen Augenkammer befindet sich die Regenbogenhaut, die Iris, die die Augenfarbe bestimmt. Durch die Pupille, die im Loch der Iris liegt, kommen die Lichtstrahlen ins Augeninnere. Der Durchmesser der Pupille verändert sich je nach Tageslichteinfall und variiert zwischen 4 und 8 mm. Die Pupille verbindet die Vorderkammer mit der hinteren Augenkammer. Die hintere Augenkammer liegt zwischen der Iris und der Linse, sodass das Kammerwasser durch die Pupille aus einer Kammer in die andere fließen kann. Das Kammerwasser gelangt im Winkel zwischen Hornhaut und Iris durch ein bindegewebiges Geflecht (Trabekelwerk) und den

Schlemm-Kanal und kann somit abfließen. Die Augenlinse, die etwas elastisch ist, hängt an ebenfalls elastischen Zonulfasern.

Das Licht gelangt zunächst durch die Pupille, dann durch die Linse und den Glaskörper auf die Netzhaut (Retina), wo es auf die Makula trifft. Die Makula befindet sich auf der Höhe des Sehnervs und enthält die lichtempfindlichsten Zellen. Da die Zellen in der Makula reich an gelbem Farbstoff sind, nennt man die Makula auch gelben Fleck. Hier findet das schärfste Sehen statt. Der Sehnerv, der gleich neben der Makula aus dem Auge tritt, leitet die aufgenommenen Sinneseindrücke an das Gehirn weiter. Diese Stelle wird auch als blinder Fleck bezeichnet, da es hier keine lichtempfindlichen Zellen in der Netzhaut gibt. Die Aderhaut, in der sich die Blutgefäße befinden, versorgt die Netzhaut. Die Lederhaut des Auges gewährleistet die Festigkeit des Auges und grenzt den Augapfel nach außen hin ab.

2.2.1.2 Arten von Augenbewegungen

Nach [Rickheit u. a. (2003) S. 142-144] umfassen Augenbewegungen die Bewegungen des Augapfels, des Lidschlags und der Pupillenmotorik. Durch das Weiten und Verengen der Pupille werden die Lichtmengen reguliert, die auf die Netzhaut gelangen. Die Bewegungen des Lidschlags schützen das Auge vor dem Austrocknen und vor Fremdkörpern. Bei den Bewegungen des Augapfels werden fünf verschiedene Augenbewegungen unterschieden, die im Folgenden kurz angeführt werden.

Vestibuläre Augenbewegungen: Die vestibulären² Augenbewegungen sind Anpassungsbewegungen des Auges, um sich auf einen Punkt fixieren zu können, wenn sich der Kopf und/ oder Körper bewegt. Diese Anpassungsbewegungen stehen in Verbindung mit dem vestibulären System und sind unabhängig vom visuellen Stimulus. Die Augenbewegungen sind biphasisch; das heißt, die Augenbewegungen wechseln zwischen langsamen Gleitbewegungen und schnellen Rückstellbewegungen. Diesen Mechanismus nennt man auch vestibulären Nystagmus.

Optikinese: Die Optikinese wiederum sorgt für ein stabiles Netzhautbild, wenn sich ein Großteil der visuellen Umgebung bewegt. Dabei kommt es zum optokinetischen Nystagmus: Es wechseln sich langsame und schnelle Augenbewegungen ab, wie bei den vestibulären Augenbewegungen.

Folgebewegungen des Auges: Die Folgebewegungen des Auges dienen dem Fixieren eines sich bewegenden Blickobjektes. Dabei werden die relativ langsamen und gleitenden Folgebewegungen willentlich vom Auge gesteuert. Bewegt sich das Blickobjekt schneller sind Sakkaden notwendig.

²vestibulär: den Gleichgewichtssinn betreffend

Sakkaden: Sakkaden sind kleine ruckartige Bewegungen, die das Auge auf ein Blickobjekt richten oder zwischen mehreren Blickobjekten hin und her schnellen lässt und ballistisch sind³. Ihre Funktion besteht darin, den Punkt des schärfsten Sehens auf den im Interesse liegenden Teil eines Objektes zu richten. Die Sakkaden werden durch kleine Pausen - den Fixationen - unterbrochen. Die Fixationen dienen der Aufnahme der visuellen Informationen. Dabei fokussieren Minimalbewegungen das Abbild auf der Retina, damit es nicht sofort verblasst, da die Retina vor allem auf veränderliche Reize reagiert. Diese Minimalbewegungen lösen die Reizung der Sinneszellen auf der Netzhaut aus, sodass das Objekt auf der Netzhaut fixiert werden kann. Die Minimalbewegungen der Augen lassen sich in Drift⁴, Tremor⁵ und Mikrosakkade⁶ unterteilen. Fixationen haben eine Dauer von 100-2000 ms, wobei viele Fixationen im Bereich 200-600 ms liegen. Nach [Schmidt u. a. (2003) S. 376] beträgt die Sakkadendauer zwischen 10-80 ms.

Vergenzbewegungen: Bei diesen Augenbewegungen bewegen sich beide Augen gegenläufig, wenn von einem weit entfernten Objekt zu einem nah gelegenen Objekt geschaut wird oder umgekehrt. Die Vergenzbewegungen sichern, dass Objekte in der Sehgrube von beiden Augen abgebildet werden und gewährleisten damit die Tiefenschärfe.

Bei den verschiedenen Augenbewegungen richten sich die Sakkaden, die Folgebewegungen und die Vergenzbewegungen zielgerichtet auf ein Objekt, wohingegen die vestibulären Augenbewegungen und die Optokinese der Stabilisierung der Bilder auf der Netzhaut dienen, wenn sich der Körper oder die Umwelt bewegen. Von den Augenbewegungen, die sich auf ein Objekt richten sind für die Registrierung der Augenbewegungen, um den Mauszeiger an die gewünschte Stelle zu positionieren, vor allem die Sakkaden mit Fixationen relevant, da diese ein Blickobjekt fokussieren oder zwischen mehreren Blickobjekten hin und her wechseln. Die Folgebewegungen sind irrelevant, weil sie ein sich bewegendes Blickobjekt verfolgen, was bei der Anwendung strenggenommen nicht vorkommt. Die Vergenzbewegungen beziehen sich auf die Sicherung der Tiefenschärfe, wenn zwischen Objekten hin und hergeschaut wird, die sich in unterschiedlicher Entfernung befinden, was ebenfalls bei der Anwendung nicht auftritt.

2.2.2 Der adäquate Reiz

Der adäquate Reiz für das Sehen ist eine elektromagnetische Strahlung mit einer Wellenlänge von 400 bis 750 nm, die als Licht wahrgenommen wird, und einer Leuchtdichte von 10^{-4} bis 10^{-7} cd/m^2 [Schmidt u. a. (2003) S. 368-369]. Die Strahlung hat verschiedene

³ballistisch: einmal initiiert, können sie nicht mehr unterbrochen werden

⁴langsam gleitende bogenförmige Bewegungen

⁵zitterartige Bewegungen

⁶ruckhafte Bewegung, um den Fixationspunkt herzustellen

Wellenlängen. Die infrarote Strahlung mit einer Wellenlänge oberhalb von 750 nm und die ultraviolette Strahlung mit einer Wellenlänge unterhalb von 400 nm ist für den Menschen nicht wahrnehmbar. Licht mit nur einer Wellenlänge nennt man monochromatisch. Da die Gegenstände in unserer Umwelt Licht mit unterschiedlichen Wellenlängen absorbieren und reflektieren, nehmen wir sie je nach spektraler Reflektanz, Hell-Dunkelkontrasten und Farbkontrasten unterschiedlich wahr. Die Oberflächen der Gegenstände erscheinen uns dann farbig, wenn die spektrale Reflektanz ungleichmäßig über das sichtbare Spektrum - also 400-750 nm - verteilt ist. Ein Maß für die gesehene Helligkeit ist die Leuchtdichte ($cd = \text{candela}$). Sie bezieht sich auf den von Gegenständen ausgehenden Lichtstrom pro Fläche ($cd \times m^{-2}$). Der physikalische Kontrast (C) wird durch den Unterschied der Leuchtdichte von benachbarten Strukturen bestimmt. Der physikalische Kontrast ergibt sich folgendermaßen: $C = (I_h - I_d)/(I_d + I_h)$, wobei I_h die Leuchtdichte von helleren und I_d die Leuchtdichte von dunkleren Gegenständen meint. Sehen wird mit Hilfe von Adaptionsprozessen in einem Leuchtdichtebereich zwischen 10^{-4} bis $10^{-7} cd/m^2$ möglich.

Das Infrarotlicht, das eine Wellenlänge oberhalb von 750 nm hat, wie oben beschrieben wurde, wird bei der Bestrahlung des Auges nicht als störend empfunden. Das Infrarotlicht wird benötigt, wie im nächsten Abschnitt beschrieben wird, um einen Punkt auf der Hornhaut zu projizieren oder um die Pupille auszuleuchten.

2.2.3 Methoden der Augenbewegungsregistrierung

Die Augenbewegungsregistrierung kann mittels Fremdbeobachtung, wie z.B. Kamera oder technisch anhand der Spannung des Auges, nach [Rickheit u. a. (2003) S. 146-151] erfasst werden.

Bei der Fremdbeobachtung werden die Eigenschaften der Hornhaut als auch der Pupille genutzt, um bestimmte Grenzflächen zu kennzeichnen und Reflexionspunkte zu erzeugen. Um einen Reflexionspunkt zu erzeugen, dienen die Reflexionseigenschaften der Hornhaut. Um eine Grenzfläche der Pupille zu kennzeichnen, wird der Farbübergang zwischen Pupille und Iris genutzt.

Die Messprinzipien der Augenbewegung werden wie folgt unterschieden:

- Subjektive Erfassung
- Getrennte Erfassung horizontaler und vertikaler Bewegung
- Videobasierte Erfassung

Die videobasierte Erfassung und Bildverarbeitung ist ein aktuelles Messprinzip, das bezüglich der Steuerung des Mauszeigers eine wichtige Rolle spielt. Bezüglich der subjektiven

Erfassung und der getrennten Erfassung horizontaler und vertikaler Bewegung können weitere vertiefende Kenntnisse in dem Buch [Rickheit u. a. (2003)] erworben werden.

2.2.3.1 Videobasierte Erfassung

Bei der videobasierten Erfassung von Augenbewegungen dienen einige optische Charakteristika für die Registrierung. Es gibt Charakteristika des Auges, die optisch ohne Schwierigkeiten identifizierbar sind. Deshalb haben sich diese für die kontinuierliche Registrierung von Augenbewegungen bewährt. Die horizontalen Augenbewegungen werden am besten auf der Grenzlinie zwischen Augapfel und Iris, dem Limbus, registriert. Das Augenlid eignet sich zur Erfassung der vertikalen Augenbewegungen, da das Augenlid den vertikalen Augenbewegungen folgt. Die Pupille dient der Registrierung der horizontalen als auch der vertikalen Augenbewegungen und spielt eine entscheidende Rolle. In allen drei Fällen kann die Registrierung der Bewegung durch eine Videokamera oder über photoelektrische Registrierung erfolgen.

Durch die videobasierte Erfassung wird ein Bild des Auges aufgezeichnet, auf dem die Pupille zu sehen ist. Damit die interessanten charakteristischen Merkmale des Augenbildes extrahiert, d.h. herausgefiltert, werden können, wird es einer rechnergestützten Bildverarbeitung zugeführt. Die Extraktion der charakteristischen Merkmale kann nach folgenden Methoden geschehen.

Cornea-Reflex-Methode: Wenn das infrarote Licht neben einer Kamera mit einer entsprechenden Halterung am Auge positioniert ist, wird es unter anderem an der Oberfläche der Hornhaut (Cornea) reflektiert, da die Oberfläche glänzend ist, wie auf der Abbildung [2.4 S. 24] dargestellt ist. Diese Reflexion wird als Corneareflexion oder als erstes Purkinje Bild bezeichnet und stellt den ersten Referenzpunkt dar. Die Corneareflexion befindet sich 3,5 mm hinter der Augenoberfläche. Die Pupillenmitte des Auges repräsentiert den zweiten Referenzpunkt und dient als Rotationspunkt R_r des Auges. Der Radius r_c der Hornhaut beträgt etwa 8mm. Wenn sich das Auge um einen Winkel θ bewegt, verschiebt sich die Corneareflexion um den Betrag h_q .

$$h_q = (r_r - r_c) * \sin \theta$$

Siehe Abbildung [2.3 S. 24]

Die Corneareflexion und die Pupillenmitte bewegen sich bei translatorischen (in Abhängigkeit des Kopfes) Augenbewegungen mit konstantem Abstand in die gleiche Richtung. Der Abstand der Corneareflexion und der Pupillenmitte zueinander variiert jedoch bei rotierenden (in Abhängigkeit eines Objektes) Augenbewegungen. Daher kann über die relative Distanz und Lage der Corneareflexion zur Pupillenmitte die rotierende Augenbewegung in Abhängigkeit zu einem betrachteten Objekt bestimmt werden.

Wie oben bereits erwähnt, geschieht dies üblicherweise über Bildverarbeitungsverfahren. Anhand dieser werden aus dem erfassten Bild zuerst die Referenzpunkte extrahiert. Nach der berechneten Distanz zwischen den Punkten und mit Hilfe der Kalibrierungspunkte kann nun die Position des Auges berechnet werden.

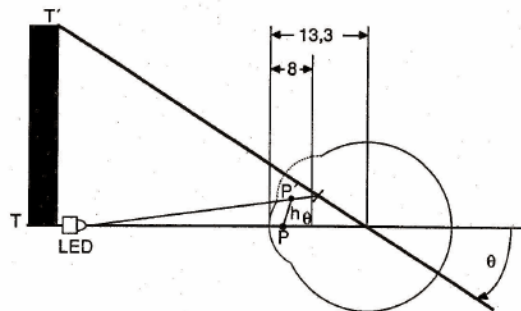


Abbildung 2.3: Entstehung des ersten Purkinje-Bildes und seine Verschiebung bei Rotation des Auges [Rickheit u. a. (2003)]

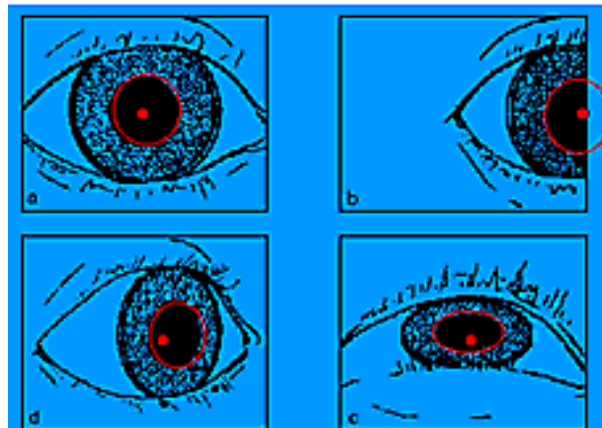


Abbildung 2.4: Cornea-Reflex-Methode a) Primärposition b) nach der Kopfverschiebung c) und d) nach Augenbewegungen [Metrovision (2008)]

Blickachsenmessung (Point of Regard Measurement): Die Blickachsenmessung bezieht sich auf einen festen Punkt des Auges und einen Lichtreflex hinsichtlich der Blickachse. Diese Methode ist unter dem Namen „Point of Regard Measurement“ bekannt [Duchowski (2003)]. Normalerweise wird, wie oben beschrieben wurde, die Distanz zwischen dem Corneareflex und dem Mittelpunkt der Pupille gemessen. Andere Referenzpunkte bei der Distanzmessung können ein Lichtreflex und ein markantes Blutgefäß auf der Netzhaut sein. Wenn sich der Kopf bewegt, verschiebt sich die Distanz der Messpunkte nicht. Wenn sich

jedoch die Augen bewegen, kommt es zur systematischen Verschiebung des Corneareflexes gegenüber dem Pupillenmittelpunkt (Fixpunkt).

Doppelte Purkinje-Bild-Technik: Auf das Auge können mehrere Infrarotlichtstrahlen projiziert werden, sodass auf der Oberfläche des Auges das Licht gebrochen wird, wie auf der Abbildung [2.5 S. 25]. Das erste Purkinje-Bild entspricht, wie bereits erwähnt, der ersten Reflexion an der Oberfläche der Hornhaut, nämlich der Corneareflexion, und stellt bei dieser Methode den ersten Referenzpunkt dar. Das zweite Purkinje-Bild entsteht an der Grenzfläche zwischen Hornhaut und Kammerwasser, das dritte entsteht zwischen Kammerwasser und Linse und das vierte entsteht zwischen Linse und Glaskörper. Der Unterschied zur vorherigen Methode besteht in dem zweiten Referenzpunkt. Hier wird anstelle der Pupillenmitte das vierte Purkinje-Bild benutzt. Dies ist ebenfalls eine Reflexion des Auges, die an der hinteren Seite der Linse entsteht. Das vierte Purkinje-Bild ist eine nur sehr schwache Lichtreflexion. Daher muss bei der Aufzeichnung der Augenbewegungen mit dieser Methode der Raum abgedunkelt werden. Ansonsten könnten bei der Auswertung des Videosignals Schwierigkeiten bei der Unterscheidung bzw. Erkennung der beiden Reflexionen auftreten. Beide Reflexionen bewegen sich während translatorischer (vom Kopf abhängig) Augenbewegungen mit konstantem Abstand in die gleiche Richtung. Ihr Abstand zueinander variiert jedoch bei rotierenden Augenbewegungen. So kann genau wie bei der vorherigen Methode auf Grund des Abstandes und der Lage beider Reflexionen auf Augenbewegungen in Abhängigkeit eines Objektes geschlossen werden.

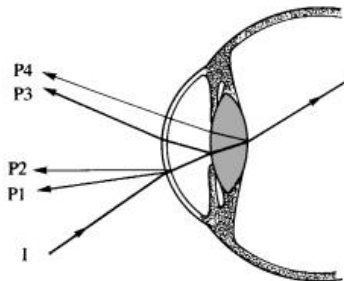


Abbildung 2.5: Doppelte Purkinje Darstellung mit vier Referenzpunkten [Dresden (2008)]

Bright Pupil: Wenn man das Auge mit einem Licht, das sich in der optischen Achse der Kamera befindet, beleuchtet, wird dieses Licht von der Netzhaut reflektiert. Die Pupille erscheint dann hell. Diese Methode ist bekannt unter dem Namen „bright pupil“.

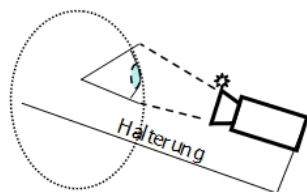
Kontaktlinsenmethode (videobasiert): Bei der Kontaktlinsenmethode, die videobasiert ist, wird die Reflexion eines Spiegels, der in die Kontaktlinse integriert ist, aufgezeichnet. Bei [Rickheit u. a. (2003)] heißt es, dass die Augenbewegungen eine doppelt so große Bewegung

des Reflexes auslösen, wenn es sich um einen bestimmten Winkel handelt. Das heißt, dass bei diesem Verfahren eine höhere ortsbezogene Auflösung (bei gleichem optischen Sensor) realisiert wird.

2.2.4 Tabellarischer Vergleich der Systeme

In diesem Abschnitt folgt der tabellarische Vergleich, um die Eigenschaften der Systeme darzustellen. Hierbei richtet sich die Aufmerksamkeit u.a. auf die Eigenschaften „Wichtige Funktionen“, „Arten der Augen- oder Kopfbewegungen“ und „Methode der Augenbewegungsregistrierung“. Anhand dieser Eigenschaften sollen Erkenntnisse gewonnen werden, welche Methode der Augenbewegungsregistrierung für ein System geeignet ist. Hierbei muss erwähnt werden, wie die Methoden mit dem Infrarotsignal, bezogen auf das jeweilige System, zusammenhängen. Weiterhin werden die Referenzpunkte aufgeführt, die durch das projizierte Infrarotlicht entstehen und die sich auf ein gesuchtes Merkmal beziehen. Des Weiteren wird hier kurz auf die Konstruktion der Halterung und Positionierung der Kamera eingegangen. Wie aus der Abbildung [2.6 a) S. 26] ersichtlich ist, benötigen die kopfbasierten Systeme eine Halterung, mit der die Kamera nah am Auge positioniert werden kann. Dagegen kann bei einem monitorbasierten System die Kamera im Monitor eingebaut sein, um die Augenbewegungen zu registrieren, wie auf der Abbildung [2.6 b) S. 26] dargestellt ist.

a) Kopfbasiertes System



b) Monitorbasiertes System

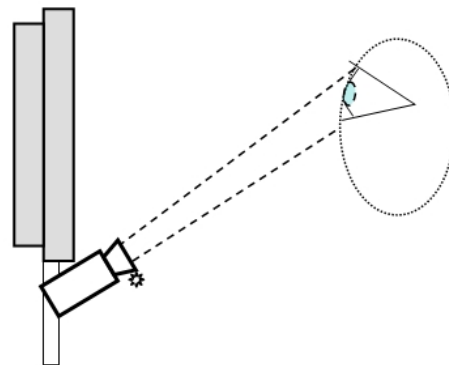


Abbildung 2.6: a) Kopfbaisiertes System und b) Monitorbasiertes System

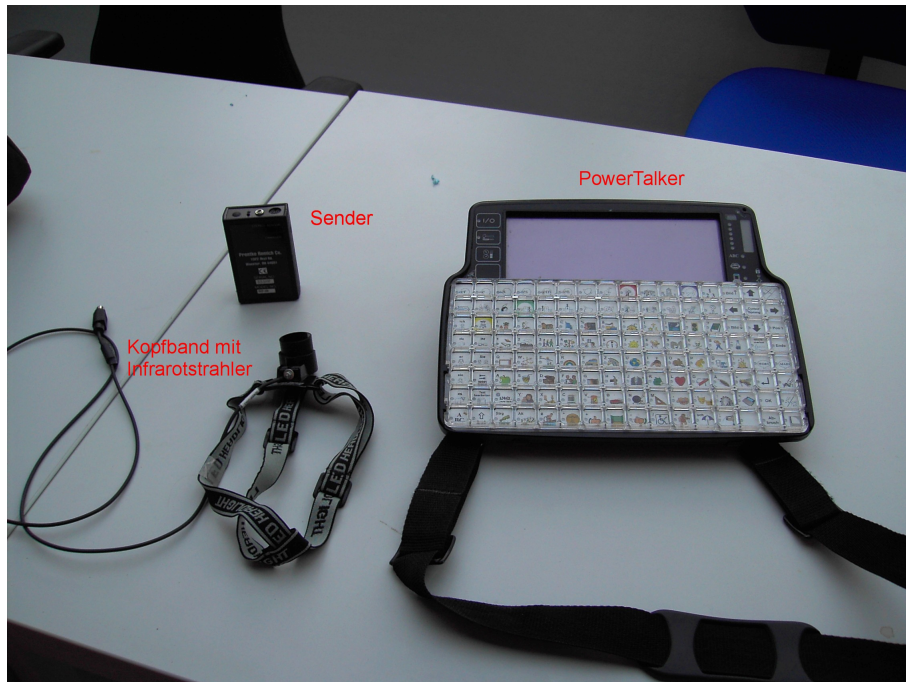


Abbildung 2.7: PowerTalker



Abbildung 2.8: Ein Beispiel aus ISAAC, wie man einen Satz codiert. [ISAAC (2008)]



Abbildung 2.9: Tower des HeadMouse-Systems

Tabelle 2.1: Bestandteile und Funktionen des PowerTalker-Systems

System	Kopfbasiert
Hersteller	Prentke Romich [Romich (2008)]
Produkt	PowerTalker Abbildung: [2.7 S. 27]
Hardware-Komponenten	Infrarotempfänger, Infrarotstrahler, Infrarot-Empfänger/ -Sender
Schnittstellen	serielle Schnittstelle
Signal	Infrarot
Entfernung	zwischen Infrarotstrahler und PowerTalker max. bis zu 1,5 m
Positionierung der Komponenten	Befestigung des Infrarotstrahlers am Kopf mit Hilfe der jeweiligen Halterung. Der PowerTalker kann sich z.B. auf dem Tisch befinden.
Wichtige Funktionen	Detektieren der Kopfbewegungen mittels Infrarotstrahler. Ermittlung der Position der Kopfbewegung in Bezug auf den Display mit Hilfe der Display-Kalibrierung. Positionierung des Mauszeigers nach erfolgreichem Detektieren der Kopfposition. Beherrschung der internationalen Code-Symbole für die Codierung von Sätzen. Abbildung: [2.8 S. 27]
Anwendung und Eingabe	Steuerung des Mauszeigers durch Kopfbewegungen. Auslösung des Klickereignisses, z. B. nach 3 s., wenn der Mauszeiger seine Position nicht verändert hat.
Arten der Kopfbewegungen	Schnelle und kleine Bewegungen werden präzise erkannt. Zeitverluste zwischen Kopfbewegung und Mauszeigersteuerung, sodass sich der Mauszeiger langsamer als der Kopf bewegt, sind nicht bemerkbar.
Methode	Basiert auf dem Infrarotsignal. Hierfür ist eine extra Methode vorgesehen.
Vorteile	Genaue Positionierung und Steuerung des Mauszeigers. Zeitverluste zwischen der Kopfbewegung und Mauszeigersteuerung sind nicht bemerkbar.
Nachteile	Überbeanspruchung der Nackenmuskulatur sowie Kopfschmerzen, Müdigkeit und Konzentrationsschwäche.

Tabelle 2.2: Bestandteile und Funktionen des HeadMouse-Systems

System	Kopfbasiert
Hersteller	Origin Instruments [Instruments (2008)]
Produkt	HeadMouse Abbildung: [2.9 S. 27]
Hardware-Komponenten	Infrarot-Tower mit Infrarot LED. Reflektorpunkt
Schnittstellen	Infrarot-Empfänger/ -Sender an dem Tower, USB, COM
Signal	Infrarot
Entfernung	zwischen Kopf und Tower max. bis zu 1,5 m
Positionierung der Komponenten	Ankleben des Reflektorpunkts an der Stirn oder dem Brillengestell. Der Tower kann sich z.B. auf dem Tisch befinden.
Wichtige Funktionen	Detektieren der Kopfbewegung mittels Reflektorpunkt. Ermittlung der Position der Kopfbewegung in Bezug auf den Bildschirm mit Hilfe der Bildschirm-Kalibrierung.
Anwendung und Eingabe	Steuerung des Mauszeigers durch Kopfbewegung. Auslösung des Klickereignisses, z. B. nach ca. 3 s, wenn der Mauszeiger seine Position nicht verändert hat.
Arten der Kopfbewegungen	Schnelle und kleine Bewegungen werden präzise erkannt. Zeitverluste zwischen Kopfbewegung und Mauszeigersteuerung, sodass sich der Mauszeiger langsamer als der Kopf bewegt, sind nicht bemerkbar.
Methode	Basiert auf dem Infrarotsignal. Hierfür ist eine extra Methode vorgesehen.
Vorteile	Genaue Positionierung und Steuerung des Mauszeigers. Zeitverluste zwischen Kopfbewegung und Mauszeigersteuerung sind nicht bemerkbar.
Nachteile	Überbeanspruchung der Nackenmuskulatur sowie Kopfschmerzen, Müdigkeit und Konzentrationsschwäche.



Abbildung 2.10: Die Kamera mit einem beweglichen Kopf und Halter des Tracker One-Systems



Abbildung 2.11: MyTobii

Tabelle 2.3: Bestandteile und Funktionen des Tracker One-Systems

System	Kopfbasiert
Hersteller	Origin Instruments [Instruments (2008)]
Produkt	Tracker One Abbildung: [2.10 S. 30]
Hardware-Komponenten	Videokamera (Objektiv mit Infrarot LED), Reflektorpunkt
Schnittstellen	COM, USB
Signal	Video (Infrarotsensitive Kamera)
Entfernung	zwischen Kopf und Kamera max. bis zu 1,5 m
Positionierung der Komponenten	Die Kamera ist am Bildschirm positioniert, sodass sie den Reflektorpunkt erfassen kann. Ankleben des Reflektorpunkts am Kopf oder Brillengestell.
Wichtige Funktionen	Detektieren der Kopfbewegung mittels Videoaufnahme (Bilder). Ermittlung der Zielposition anhand der Kopfbewegung und der Kalibrierungsdaten des Bildschirms.
Anwendung und Eingabe	Steuerung des Mauszeigers durch Kopfbewegung. Auslösung des Klickereignisses, z. B. nach ca. 3 s, wenn der Mauszeiger seine Position nicht verändert hat.
Arten der Kopfbewegungen	Schnelle Bewegungen werden erkannt. Kleine Bewegungen werden je nach Distanz zwischen Kamera und Reflektorpunkt mal erkannt mal nicht. Zeitverluste zwischen Kopfbewegung und Mauszeigersteuerung, sodass sich der Mauszeiger langsamer als der Kopf bewegt, sind bei sehr schnellen Kopfbewegungen minimal bemerkbar.
Methode	Ungefähr wie die Blickachsenmessung. Allerdings wird hier der Reflektorpunkt als Referenzpunkt registriert.
Vorteile	Genauere Positionierung und Steuerung des Mauszeigers.
Nachteile	Überbeanspruchung der Nackenmuskulatur sowie Kopfschmerzen, Müdigkeit und Konzentrationsschwäche.

Tabelle 2.4: Bestandteile und Funktionen des MyTobii-Systems

System	Monitorbasiert
Hersteller	Tobii [Tobii (2008)]
Produkt	Tracker One Abbildung: [2.11 S. 30]
wichtige Hardware	Videokamera (Objektiv mit Infrarot LED und CCD)
Schnittstellen	USB
Signal	Video
Wichtige Funktionen	Detektieren der Pupille, um die Position dieses Referenzpunktes zur Verfügung zu stellen. Ermittlung der Zielposition mittels der Kalibrierungsdaten und der Position des Referenzpunktes und Positionierung des Mauszeigers an dieser Stelle. Auslösung des Klickereignisses, z. B. nach ca. 3 s.
Positionierung der Komponenten	Die Kamera ist am Bildschirm so positioniert, dass sie die Augenbewegungen erfassen kann (Blickachse).
Anwendung und Eingabe	Steuerung des Mauszeigers durch Augenbewegungen.
Arten der Augenbewegungen	Langsame Sakkaden mit Fixationen und Folgebewegungen werden erkannt. Schnelle Sakkaden und Mikrobewegungen werden laut Herstellerangaben ignoriert, also nicht erkannt.
Methode	Blickachsenmessung in Verbindung mit Bright Pupil. Cornea-Reflex-Methode mit Bright-Pupil auch möglich. Die Pupille ist der Referenzpunkt.
Vorteile	Angenehmes Gefühl beim Steuern des Mauszeigers durch Augenbewegungen.
Nachteile	Zeitverluste zwischen Kopfbewegung und Mauszeigersteuerung sollen bemerkbar sein. Somit keine schnelle präzise Positionierung des Mauszeigers.

3 Anforderung, Signal und Bildmaterial

In diesem Kapitel wird zuerst die Anforderung erläutert, damit bestimmte Ziele formuliert werden können. Anhand der Anforderung wird entschieden, welche Methode für die Messung der Augenbewegungen geeignet ist. Außerdem wird auf das Signal sowie die Halterung für die Positionierung der Webkamera eingegangen. Das vorgesehene Bildmaterial wird vorgestellt und beschrieben. Aufgrund der Störfaktoren, die innerhalb der Bilder vorkommen können, wird das Bildmaterial interpretiert.

3.1 Anforderung

Wie aus der Einführung hervorgeht, soll ein Verfahren entwickelt werden, mit dem die Augenbewegungen eines Anwenders registriert werden. Aus den registrierten Bilddaten soll die Pupille detektiert und dabei durch einen Kreis gekennzeichnet werden. Hierzu soll der Mittelpunkt der Pupille bzw. des Kreises als Position festgehalten werden.

Diese Position dient als Referenzpunkt, ist durch einen x- und y-Wert beschrieben und für die Kalibrierung wichtig. Die Kalibrierungsdaten korrelieren mit dem Referenzpunkt, sodass daraus die Position der Augenbewegung, wohin der Anwender auf den Bildschirm schaut, ermittelt werden kann.

Des Weiteren soll das Verfahren bei einer unzureichenden Beleuchtung möglichst stabil verlaufen, sodass die Pupille detektiert wird. Bezüglich der Rechenzeit soll das Verfahren nicht zu langsam sein.

Ein weiterer wichtiger Aspekt ist, dass für die Registrierung der Augenbewegungen eine sehr günstige Webkamera verwendet werden soll. Dies ist deswegen erforderlich, damit erforscht werden kann, inwieweit mit einer billigen Webkamera das Verfahren für das Detektieren funktioniert, um damit die Kosten solcher Systeme zu senken und möglichst vielen Anwendern zur Verfügung zu stellen.

3.2 Eingangssignal und Methode der Augenbewegungsmessung

Wie in der Anforderung dargelegt wurde, soll die Augenbewegung mit Hilfe eines Videosignals registriert werden. Diesbezüglich wird für das Verfahren eine günstige Webkamera gewählt, die ein Videosignal liefert.

Damit die Bewegung des Auges registriert wird, ist eine Halterung für die Webkamera erforderlich. Bevor auf die Konstruktion der Halterung eingegangen wird, ist zu entscheiden, nach welcher Systemart (kopfbasiert/monitorbasiert) das System gestaltet werden soll. Da das genaue Messergebnis sehr von der Distanz zwischen Augenbewegung und Objektiv der Kamera abhängt, wird sich an dieser Stelle für das kopfbasierte System entschieden. Je größer die Distanz zwischen Auge und Objektiv ist, desto schwieriger ist es (wie bei dem System „MyTobii“ angegeben ist), die schnellen Sakkaden zu erkennen, wobei hier die Brennweite des Objektivs eine wichtige Rolle spielt. Ein weiterer wichtiger Anhaltspunkt dieses kopfbasierten Systems ist der Erkennungsbereich. Viele behinderte Anwender können eventuell aufgrund einer Liegeposition diesen Erkennungsbereich bei monitorbasierten Systemen nicht einhalten.

Passend zum kopfbasierten System und wie man aus der Darstellung der Methoden zur Augenbewegungsregistrierung ersehen kann, verspricht die Cornea-Reflex-Methode ein genaues Ergebnis, sodass die Aufmerksamkeit zuerst auf diese Messmethode gerichtet wird. Wie anhand dieser Methode zu erkennen ist, beinhaltet sie zwei Referenzpunkte. Der erste Referenzpunkt beschreibt die Position der Pupille und der zweite beschreibt den durch Infrarotlicht projizierten Punkt. Der erste Punkt verdient am meisten Aufmerksamkeit, da dieser dynamisch ist und in dem zu entwickelnden Verfahren eine zentrale Rolle spielt. In diesem Zusammenhang wird erstmal auf Infrarot-Leuchtdioden verzichtet, da dies nicht notwendig für das Detektieren der Pupille ist.

Damit die Webkamera nah am Auge positioniert wird, ist eine Halterung notwendig. Diese Halterung, wie auf der Abbildung [3.1 S. 35] dargestellt ist, wird aus einem Brillengestell und aus einem Draht zusammengestellt. Mit dem Draht wird die Webkamera an dem Brillengestell befestigt. Die Biegsamkeit des Drahtes ermöglicht es, je nach Wunsch, die Webkamera nah am Auge zu positionieren.

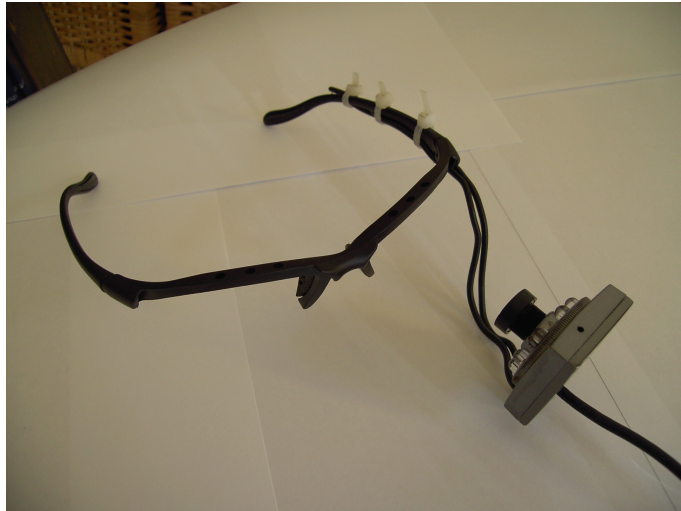


Abbildung 3.1: Brillengestell mit Webkamera

3.3 Bildmaterial und Interpretation

Nachdem die Webkamera nah am Auge positioniert ist, werden einige Bilder für das Bildmaterial erzeugt. Hierbei werden die Augenbewegungen registriert, damit die Bilder interpretiert werden können. Die Interpretation ist deswegen notwendig, damit die Störfaktoren bei dem Detektieren der Pupille im Voraus erkannt werden und bei dem Herangehen und der Entwicklung des Verfahrens geeignete Maßnahmen dagegen ergriffen werden können. Das Herangehen sowie die Entwicklung des Verfahrens orientieren sich stark am Ablaufdiagramm für das Detektieren der Pupille, siehe Abbildung [4.1 S. 42]. Bevor auf die Interpretation des Bildmaterials eingegangen wird, werden einige Grundlagen aufgegriffen, damit diese Störfaktoren verstanden und unterschieden werden. Hierzu werden die Grundlagen wie Aufnahmesystem und Nachbarschaft eines Bildpunktes erwähnt.

Digitales Bild: Wie auf der Abbildung [4.1 S. 42] zu sehen ist, beginnt der Ablauf für das Detektieren der Pupille mit der Aufnahme eines Objekts, für die ein Bildaufnahmesystem zuständig ist.

Ein wesentlicher Mechanismus zur Erzeugung von Bilddaten, wie aus der Literatur hervorgeht [Gonzalez und Woods (2003) S. 26-58], [Jähne (2005) S. 17-38, S. 243-255], ist die Umwandlung einfallender Lichtintensität in räumlich verteilte elektrische Signale.

Diese Signale werden in digitale Repräsentationen umgewandelt, damit sie mit einer Rechenanlage verarbeitet werden. Der Frame Grabber gehört als ein wichtiger Bestandteil zur Hardware eines Bildaufnahmesystems und sorgt dafür, dass die Signale digitalisiert und gespeichert werden. Dazu wird ein Verstärker benötigt, der das Signal aufbereitet, und ein

Analog-Digital-Wandler, der das Signal digitalisiert. Das heißt, dass jeder Bildpunkt (Pixel) einen Wert der Bildhelligkeit der jeweiligen Bitmuster enthält. Bei der Digitalisierung des Signals wird die Projektion eines Objekts auf der Bildebene (örtlich) durch ein regelmäßiges Raster von Photosensoren abgetastet, und die einzelnen Intensitätswerte werden quantisiert.

Quantisierung bedeutet, wenn von der kontinuierlichen zur diskreten Darstellung übergegangen wird und die Helligkeit von Pixeln (Bildpunkten) mit Hilfe einer bestimmten Grauwertmenge bewertet wird. Bei der Quantisierung verfügt ein Bildpunkt über ein binäres Wort der Länge k , der grundsätzlich einen der $(2^k - 1)$ Werte annimmt. Mathematisch gesehen kann ein Grauwertbild als eine Funktion $f(x, y)$ mit den zwei örtlichen Variablen x und y betrachtet werden. Der Funktionswert an der Position (x, y) entspricht dem Grauwert bzw. der Helligkeit des Bildes. Der Wert der Bildfunktion wird auf positive ganze Zahlen begrenzt.

$$0 \leq f(x, y) \leq g_{max} \quad (3.1)$$

Auf Farbbilder wird im Rahmen der vorliegenden Diplomarbeit nicht eingegangen, da sie im praktischen Teil bei der Untersuchung keine Rolle spielen.

Während der Bildraasterung werden rechteckige Gitter benutzt, sodass hierdurch rechteckige Bildpunkte (Pixel) entstehen. Außer dem rechteckigen Gitter können im Bildaufnahmesystem die dreieckigen und die sechseckigen (hexagonalen) Gitter unterschieden werden, siehe Abbildung [3.3 S. 37]. Je nach Gitterart wird die Nachbarschaft der Bildpunkte definiert. Bei den teuren Bildaufnahmesystemen werden sechseckige (hexagonale) Gitter benutzt, die für die bessere Flächenfüllung und damit eine bessere Qualität eines Bildes sorgen.

Die Diskretisierung der Ortsvariablen und Grauwerte ist die Voraussetzung, damit der Frame Grabber das Bild bearbeiten kann. Wenn ein digitales Bild diskretisiert worden ist, ergibt sich eine zweidimensionale Matrix mit M Spalten und N Zeilen. Die Grauwerte befinden sich als positive Koordinaten zwischen $0 \leq x \leq M$ und $0 \leq y \leq N$. Der Nullpunkt im Koordinatensystem liegt in der linken oberen Ecke, wie es auf der Abbildung [3.2 S. 37] dargestellt ist.

Die Nachbarschaftsrelation in diskreten Bildern gehört zu den elementaren Konzepten, um die Bildregionen von Objekten in einem Bild zu bestimmen. Damit der Zusammenhang zwischen den Bildpunkten analysiert werden kann, ob sie in einer Beziehung zueinander stehen, muss die Nachbarschaft eines Bildpunktes bekannt sein.

In einem quadratischen Gitter werden die Bildpunkte als benachbart bezeichnet, wenn sie eine gemeinsame Kante oder mindestens eine gemeinsame Ecke aufweisen, wie es auf der Abbildung [3.3 a) S. 37] dargestellt ist. Dementsprechend werden in einem quadratischen Gitter die 4er-Nachbarschaft und die 8er-Nachbarschaft unterschieden. Wenn vom Startbildpunkt über einen Weg die benachbarten Zielbildpunkte erreicht werden können, spricht man von einer zusammenhängenden Bildregion.

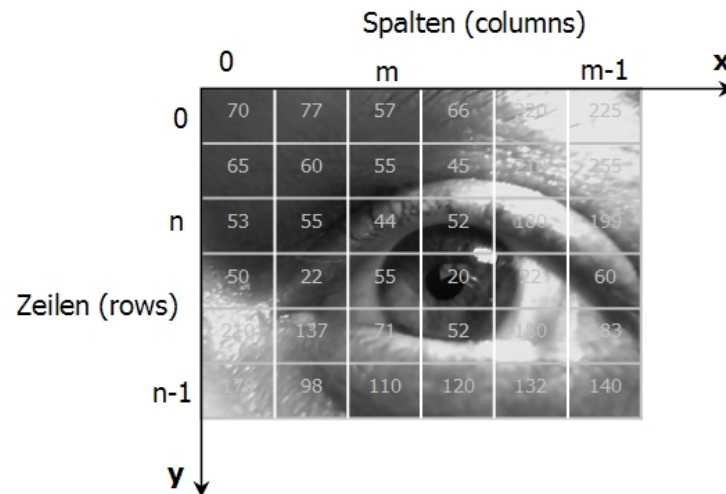


Abbildung 3.2: Eingangsbild

Die Menge der 4er-Nachbarn eines Bildpunktes (n) wird $N_4(n)$ genannt. Diese Nachbarschaft wird deswegen so genannt, weil im Abstand dieser Zuordnung $(x - 1, y)$, $(x + 1, y)$, $(x, y - 1)$, $(x, y + 1)$ die horizontalen und vertikalen Nachbarn beinhaltet sind. Die Nachbarn in der Diagonalen $(x - 1, y - 1)$, $(x + 1, y + 1)$, $(x + 1, y - 1)$, $(x - 1, y + 1)$ eines Bildpunktes (n) heißen $N_D(n)$. Mit der Bezeichnung $N_8(n)$ werden die Nachbarn eines Bildpunktes der Diagonalen und der 4er-Nachbarn beschrieben.

Im hexagonalen Raster sind die Bildpunkte regelmäßig angeordnet, sodass sie eine gemeinsame Kante und zwei gemeinsame Ecken haben. Somit ergibt sich nur eine einzige 6-Nachbarschaft, was ein großer Vorteil ist.

In dreieckigen Gittern können 3er- und 12er-Nachbarschaften auftreten.

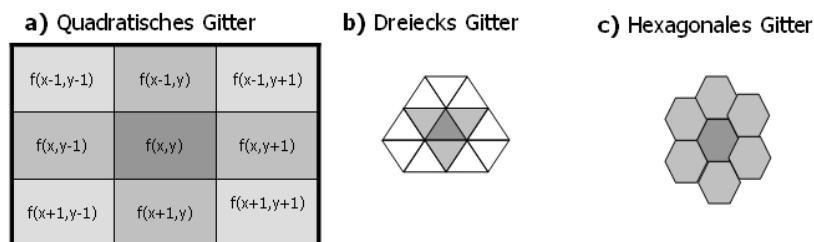


Abbildung 3.3: Mögliche 2D-Gitter mit ihrer Nachbarschaft

Heutzutage verfügen die Bildaufnahmesysteme über verschiedene Typen von Bildsensoren,

die für das Bildsignal zuständig sind. Die am meisten verbreiteten Typen von Bildsensoren sind die CCD (Charge-coupled device) und CMOS (Complementary Metal-Oxide Semiconductor). Die CCD-Bildsensoren sind überwiegend in der industriellen Anwendung verbreitet, während die CMOS-Bildsensoren in der privaten Anwendung anzutreffen sind. Grob besteht ein CCD-Bildsensor aus vielen photosensitiven Elementen, die elektrische Ladungen bei der Akkumulationsphase erzeugen. Die elektrischen Ladungen werden bei der Auslesephase durch das Transportieren über die Sensorelemente auf dem Chip in elektrische Spannung verwandelt. Im Gegensatz zu den CCD-Bildsensoren ist bei den CMOS-Bildsensoren diese Auslesephase in das jeweilige Element integriert, sodass dieses Ladungspaket in elektrische Spannung umgewandelt wird.

Durch die Rasterung und Quantisierung der elektrischen Prozesse im Bildaufnahmesystem entstehen abhängig vom Bauteil die Qualität eines Bildes und verschiedene Fehlerarten. Je nach Bauteil, Optik, Sensor, Verstärker, A/D-Wandler können die Fehlerarten klassifiziert werden. [Steinbecher (2003) S. 30-36] Eine der am meisten anzutreffende Fehlerart ist das Rauschen. Das Rauschen kann durch den Verstärker verursacht werden, indem Störungen des Zeitgebers auftreten können. Hierdurch verändern die Störungen die Ortsinformation, sodass die Intensität eines Bildpunktes verändert bzw. das Rauschen im Bild verursacht wird. Das Rauschen ist insofern wichtig, da es zu falschen Ergebnissen bei der Segmentierung und Merkmalsextraktion führen kann.

Nachdem das Rauschen sowie andere elementare Kenntnisse dargestellt worden sind, wird nun auf das Bildmaterial eingegangen. Das Bildmaterial, siehe Abbildungen [3.4 S. 39, 3.5 S. 40, 3.6 S. 40, 3.7 S. 41, 3.8 S. 41], ist dargestellt und beinhaltet Augenbewegungen, wie Sakkaden mit Fixationen. Die erzeugten Bilder beziehen sich auf einen TFT-19-Zoll Bildschirm, auf dessen vier Ecken das Auge gerichtet ist. Auf den Bildern werden die Augenbewegungen des linken Auges registriert. Hierzu ist zu bemerken, dass ungenügend auf die exakte Positionierung der Kamera auf das Auge geachtet wurde. Daraus ergibt sich, dass die Bilder einen falschen Eindruck von den Augenbewegungen erwecken können, vergleiche Abbildung [3.8 S. 41].

Wie auf der Abbildung [3.4 S. 39 siehe heller kleiner Kreis (Glühbirne) auf der Iris] zu sehen ist, können die projizierten Gegenstände der Umgebung Störfaktoren sein. Dadurch, dass die Oberfläche der Hornhaut glänzend ist, werden die Gegenstände durch das einfallende Tageslicht bzw. Lampenlicht auf der Oberfläche der Hornhaut projiziert. Diese projizierten Gegenstände können zum Teil die Pupille derart verdecken, dass die Kante zwischen Iris und Pupille nicht erkannt wird. Dies kann dazu führen, dass die Pupille gar nicht detektiert wird. Die Reflexion der Gegenstände auf der Hornhaut wird mit jedem Blinzeln fortgeführt. Durch das Blinzeln wird das Auge mit Tränenflüssigkeit befeuchtet, sodass hierdurch die Reflexion zunimmt. Somit kann das Verfahren ein falsches Ergebnis liefern. Aus dieser Darstellung wird verständlich, warum bei den Messmethoden der Augenbewegungen die Pupille oft mit Infrarotlicht beleuchtet wird, damit es eben nicht zur Reflexion von Gegenständen auf der

Hornhaut kommt.

Ein weiterer entscheidender Störfaktor, der bei der Extraktion der Pupille das Ergebnis verschlechtern oder verfälschen kann, ist der Schatten. Der Schatten kann dann vorkommen, wenn das Auge nicht adäquat beleuchtet ist. Bei schräg einfallendem Licht wirft die Nase einen Schatten auf das Auge, sodass bei der Aufnahme ein Schatten auf dem Bild entsteht. Außerdem kann der Schatten bei schlechter Raumbeleuchtung auftreten. Hierbei spielt die Augenhöhle (Orbita), die durch sieben angrenzende Schädelknochen gebildet ist, eine wesentliche Rolle. Wenn die Augen sehr weit hinten in den Augenhöhlen liegen, sodass das Licht dorthin ungenügend vordringt, bilden sich innerhalb der Augenhöhle Schatten, siehe Abbildung [3.4 S. 39, s. oben links in der Augenhöhle]. Diese Schatten können bei der Aufnahme dieselben Grauwerte und dieselbe Form wie die Pupille haben. In einem solchen Fall kann das Verfahren bei der Extraktion nicht unterscheiden, ob es sich um einen Schatten oder die Pupille handelt.



Abbildung 3.4: Fokussierung des Auges in die Mitte des Bildschirms



Abbildung 3.5: Fokussierung des Auges oben links des Bildschirms

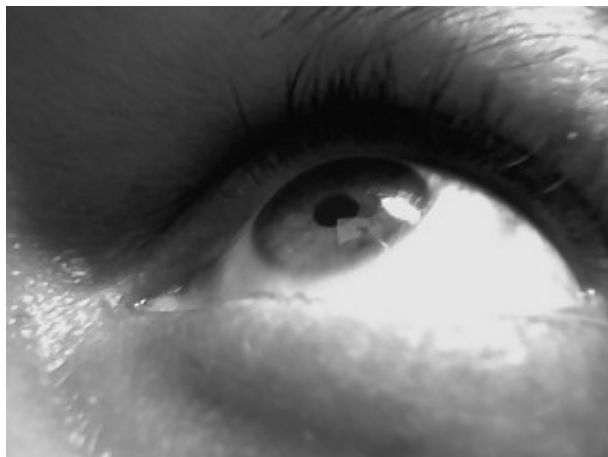


Abbildung 3.6: Fokussierung des Auges oben rechts des Bildschirms



Abbildung 3.7: Fokussierung des Auges unten rechts des Bildschirms



Abbildung 3.8: Fokussierung des Auges unten links des Bildschirms

4 Konstruktives Detektieren der Pupille

In diesem Kapitel wird konstruktiv bzw. gezielt, wie in der Anforderung gefordert ist, auf das Verfahren zum Detektieren der Pupille eingegangen. Anhand der gewonnenen Erkenntnisse und der Bildbearbeitung wird die Pupille schrittweise detektiert. Die bekannten Störfaktoren werden mit vorgesehenen Operatoren umgangen, indem die Störfaktoren zum Teil in ihrer Intensität geschwächt oder ausmaskiert bzw. ausgeblendet werden. Hierbei werden die Operatoren mit der jeweiligen Funktionalität erläutert. Das Herangehen innerhalb der Bildbearbeitung ist im Ablaufdiagramm in der Abbildung [4.1 S. 42] dargestellt. Das Ablaufdiagramm basiert auf der Mustererkennung und beinhaltet die für das Detektieren der Pupille wichtigen Verarbeitungsschritte. Hierzu werden die Entwicklungsschritte, die als Orientierung dienen, mit den jeweiligen Operatoren erläutert. Weiterhin wird das Ergebnis der jeweiligen Operatoren anhand der Bilder dargestellt und erläutert.

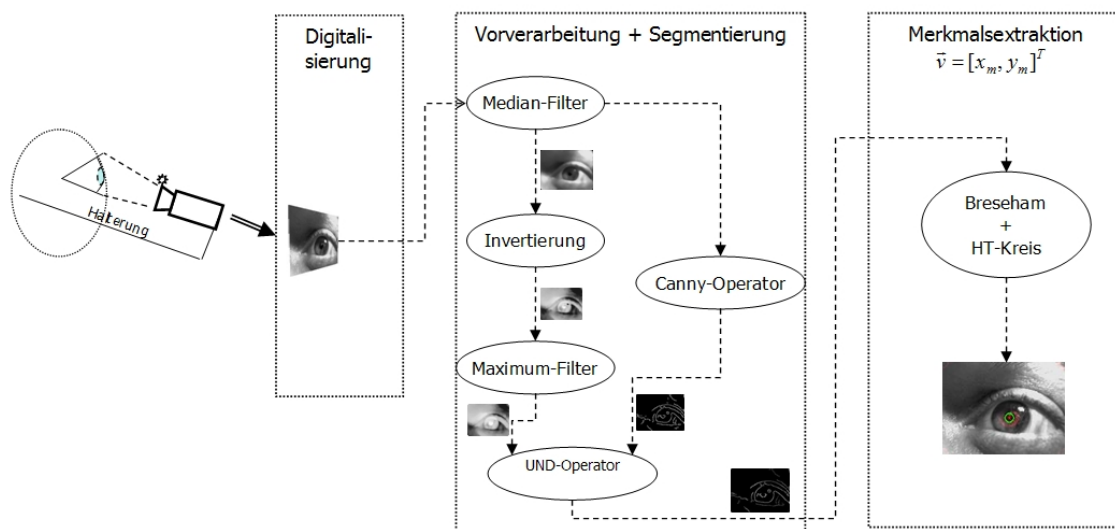


Abbildung 4.1: Ablaufdiagramm für das Detektieren der Pupille

4.1 Entscheidende Merkmale

Aus dem vorgestellten Bildmaterial geht es erstmal darum, bestimmte Informationen, die der Definition der Merkmale dienen, zu gewinnen. Diese elementare Aufgabe, bei der ein Muster mit den jeweiligen Merkmalen definiert wird, gehört zu den ersten Arbeitsvorgängen in der Bildbearbeitung. Bevor auf die Definition der Merkmale eingegangen wird, werden einige Grundlagen hinsichtlich dieses Anhaltspunktes erläutert.

Die Mustererkennung spielt in der digitalen Bildbearbeitung eine wichtige Rolle, um einen Informationsgewinn zu gewährleisten. Wie in der Literatur [Meisel (2008) Script-Teil: 1] erwähnt wird, können mit Hilfe des Musters Bildobjekte klassifiziert werden, die durch mehrere Merkmale charakterisiert sind. In einem Rechner kann ein Muster als Vektor $\vec{v} = [x_1, x_2, \dots, x_n]^T$ erfasst werden. Die Messgrößen x_1, x_2, \dots, x_n repräsentieren die Merkmale. Nach Anzahl der Messgrößen x_n wird der Vektor v als n -dimensionaler Merkmalsvektor bezeichnet. Die Merkmale sollen eine möglichst hohe Unterscheidungsfähigkeit aufweisen, so dass die Dimension des Merkmalraums nach Bedarf reduziert werden kann. Denn je größer ein Merkmalraum ist, desto zeit- und rechenintensiver ist die Suche nach einem bestimmten Bildobjekt. Anhand der Merkmale soll in einer Bilddatenmenge das gesuchte Bildobjekt erkannt werden.

Wie in der Anforderung, siehe Abschnitt [3.1], beschrieben wurde, geht es, darum die Pupille auf einem Bild zu detektieren. Wie auf dem Bildmaterial zu sehen ist, hat die Pupille eine runde Form, womit diese durch einen Kreis bzw. eine Ellipse beschrieben werden kann. Ellipsen werden jedoch nicht berücksichtigt, da sie einen großen Merkmalraum aufweisen. Diesbezüglich ist das Detektieren der Pupille sehr rechenintensiv und damit nicht brauchbar. [Burger und Burge (2005) S. 168]

Ein Kreis kann zu den n -Ecken geometrischen Figuren wie Dreieck, Quadrat, etc. klassifiziert werden und ist wie folgt in der Literatur [Papula (1998) S. 218] definiert: „Der Kreis ist definitionsmäßig der geometrische Ort aller (ebenen) Punkte P , die von einem festen Punkt, dem Kreismittelpunkt M , den gleichen Abstand r (Radius genannt) besitzen.“ Mathematisch ist ein Kreis durch folgende Kreisgleichung definiert:

$$(x - x_m)^2 + (y - y_m)^2 = r^2 \quad M = (x_m; y_m) \quad (4.1)$$

Daraus kann y wie folgt berechnet werden:

$$y = y_m \pm \sqrt{r^2 - (x - x_m)^2} \quad (4.2)$$

$$(x_m - r \leq x \leq x_m + r)$$

Oberer und unterer Halbkreis

Bezogen auf die Gleichung [4.1 S. 43] sind die drei Variablen von großem Interesse, nämlich x_m , y_m für den Mittelpunkt und r für den Radius. Mittels dieser drei Variablen soll in einer vorgegebenen Bildregion ein Kreis detektiert werden. Diesbezüglich entsprechen diese drei Variablen den Merkmalen.

Als Vektor können diese drei Merkmale zusammengefasst werden: $\vec{v} = [x_m, y_m, r]^T$. Wenn dieser Vektor \vec{v} aufgespannt wird, sind drei Dimensionen im Raum zu sehen. Dieser Vektor kann somit als dreidimensionaler Merkmalsvektor bezeichnet werden. Wie aus der Literatur [Tönnies (2005) S. 258-259] hervorgeht, ist die Suche nach einem Kreis mit den drei Merkmalen im 3-dimensionalen Merkmalsraum rechen- und zeitintensiv. Aus diesem Grund wird bei dem Merkmalsvektor $\vec{v} = [x_0, y_0, r]^T$ der Radius nicht berücksichtigt, sodass dieser dreidimensionale Merkmalsraum auf einen zweidimensionalen $\vec{v} = [x_m, y_m]^T$ reduziert wird. Wie aus dem Abschnitt [2.2.1] hervorgeht, bewegt sich der Durchmesser einer Pupille je nach Tageslicht zwischen 4 und 8 mm, sodass der Radius im Voraus mit einem festen Wert initialisiert werden kann.

Nachdem die Merkmale festgelegt sind und definiert ist, wonach in einem Bild gesucht werden soll, ist es zunächst erforderlich das Bildmaterial mittels der Bildvorverarbeitung zu verbessern. Durch die Vorverarbeitung und Segmentierung wird das Bild für die Merkmalsextraktion vorbereitet.

4.2 Das Detektieren der Pupille

Anhand der Verarbeitungsprozesse, siehe Abbildung [4.1 S. 42], folgt die Vorverarbeitung und die Segmentierung. In diesem Verarbeitungsprozess geht es allgemein um die Minimierung des Rauschens und die Verbesserung eines Bildes. Hierfür werden der Median-Filter, die Invertierung und der Maximum-Filter angewendet. In Bezug auf die Segmentierung wird ein Kantenbild mit dem Canny-Operator erzeugt, das unter anderem den gesuchten Kreis der Pupille beinhaltet. Hinsichtlich des Kantenbildes und des Maximum-Bildes wird ein logischer Operator angewendet, mit dem relevante Informationen hervorgehoben werden. Am Schluss wird die Hough-Transformation (HT) angewendet, um die Pupille zu detektieren. Angewendete Operatoren sowie die Filter mit den jeweiligen Ergebnissen und deren Bedeutung werden während der Entwicklungsschritte im kommenden Abschnitt erläutert.

4.2.1 Vorverarbeitung und Segmentierung

In der Vorverarbeitung geht es um Verfahren, mit denen zweckgebunden das Bild verbessert oder die als Vorstufe zur Segmentierung eingesetzt werden. Mit der Bildverbesserung wird erreicht, dass relevante Informationen, die bei einer speziellen Anwendung redundant oder

sogar störend sind, unterdrückt werden, um dadurch wichtige Bildinhalte deutlicher hervorzuheben.

Die Operationen der jeweiligen Filteroperatoren können grob wie folgt untergliedert werden: Operationen bei denen der Grauwert eines Bildpunktes direkt manipuliert wird - die sogenannten Bildpunktoperationen (Invertierung und logischer UND-Operator) - und Operationen, bei denen der Grauwert eines Bildpunktes in Abhängigkeit zu seiner Nachbarschaft manipuliert wird, die sogenannten Filter im Ortsbereich (Median-Filter und Maximum-Filter).

Bei den Filtern im Ortsbereich kann die lokale Nachbarschaft (N_x) eines Bildpunktes p n Nachbarschaftspunkte enthalten. Die Anzahl der Nachbarschaftspunkte hängt von der Faltungsmaske ab. Aus Gründen der Symmetrie wird meistens eine quadratische Umgebung der Gesamtgröße bei einer Faltungsmaske von 3×3 oder 5×5 verwendet. Die Filteroperationen werden kontextunabhängig und gleichmäßig für jeden Bildpunkt durchgeführt. Aus der Betrachtung der Umgebung wird ein neuer Zielgrauwert im Ausgangsbildpunkt $A(x, y)$ berechnet. Je nach Größe der Filtermaske kommt es bei der Konvolution bzw. Faltung vor, dass die Filtermaske des Eingangsbildes am Bildrand min. ein Pixel breit ist und dass sie außerhalb des Randes liegt. In diesem Fall kann der Zielgrauwert eines Ausgangsbildes nicht berechnet werden, weil dessen benachbarte Bildpunkte des Quellgrauwertes nicht existieren. Hierzu werden je nach Filter die gängigen Methoden, wie am Rand keinen Filter, Zero- oder Grauwert-Padding, periodisches Fortsetzen etc., für die Randbehandlung angewendet.

Das primäre Ziel der Verfahren ist die Verbesserung der Bilder im Ortsbereich. Hiermit sollen die Kanten, die an die Bildobjekte angrenzen, verbessert bzw. geglättet werden. Denn je deutlicher und vollständiger die Kante zwischen der Iris und der Pupille detektiert wird, desto genauer kann die Pupille detektiert werden.

Einer der Filter im Ortsbereich, der diese Anforderung erfüllt, ist der Median-Filter. Allgemein gehört der Median-Filter zu den Rangordnungsfilttern und ordnet die Grauwerte nach ihrer Größe. Der Grauwert dieser Anordnung, der in der Mitte liegt, wird als Zielgrauwert eines Zielbildpunktes gewählt.

Bei einer 3×3 Faltungsmaske ergibt sich, z. B. für einen Quellbildpunkt, eine Nachbarschaft N_8 , und dabei werden neun Grauwerte betrachtet. Diese werden nach ihrer Größe sortiert. Der Wert, der sich in der Mitte dieser Anordnung befindet, wird ausgewählt.

Bei diesem Filter hat sich eine Faltungsmaske von 3×3 , 5×5 und 7×7 bewährt, wobei dies vom Zusammenhang mit den anderen Einstellungsparametern der jeweiligen Operatoren, z.B. Downsampling, abhängt. Das Bild, bei dem eine 3×3 Faltungsmaske angewendet wurde, ist auf der Abbildung [4.2 S. 46] zu sehen. Ab einer gewissen Größe der Faltungsmaske, z. B. 9×9 , werden Teile der Kanten so verschmiert, dass sie nicht mehr erkannt werden. Das Ergebnis des Median-Filters, bei dem eine 9×9 Faltungsmaske angewendet wurde, ist auf der Abbildung [4.4 S. 47] zu sehen. Damit die Auswirkung des Median-Filters deutlich wird, bei dem die Kanten verschmiert sind bzw. vom Kantendetektor nicht mehr erkannt werden, sind

die Kantenbilder der jeweiligen Median-Bilder dargestellt. Hierzu soll das Kantenbild [4.3 S. 46] mit dem Kantenbild [4.5 S. 47] verglichen und dabei auf die Kanten des Pupillenkreises geachtet werden.



Abbildung 4.2: Median-Filter bei einer 3x3 Filtermaske



Abbildung 4.3: Kantenbild des Bildes [4.2 S. 46]

Ein weiteres Ziel innerhalb der Bildbearbeitung ist die Hervorhebung von Bildobjekten. Mit der Hervorhebung ist gemeint, dass die Bildoberfläche dieses Objektes so betont werden soll, dass das gesuchte Objekt - in diesem Fall die Pupille - möglichst deutlich wird und präzise von den anderen Bildobjekten unterschieden werden kann. Dies ist deswegen nicht einfach, da andere Bildobjekte, die als Störfaktoren gelten, wie z. B. Schatten, mit hervorgehoben werden können. Wenn der Schatten denselben Grauwert wie der Grauwert der Pupille hat, dann wird dieser Schatten automatisch wie die Pupille mitbetont.

Für die Hervorhebung werden insgesamt zwei Operatoren benötigt: Der erste Operator ist für die Invertierung zuständig. Mit diesem Operator werden die Grauwerte eines Bildpunktes invertiert. Der zweite Operator ist der Maximum-Filter. Mit diesem Filter wird das Maximum einer Ortsumgebung gefiltert, und er ist für die Hervorhebung der Bildobjekte als auch für die Ausmaskierung von Störfaktoren zuständig.



Abbildung 4.4: Median-Filter bei einer 9x9 Filtermaske



Abbildung 4.5: Kantenbild des Bildes [4.4 S. 47]

Die Invertierung kann den Bildpunktoperationen zugeordnet werden, da die Grauwerte bildpunktorientiert geändert werden. Bei der Invertierung von Grauwerten wird aus dem Wertebereich g_{max} , siehe Gleichung [3.1 S. 36] der Grauwert eines Bildpunktes subtrahiert. Dies kann wie folgt dargestellt werden:

$$f_{inv}(x, y) = g_{max} - f(x, y) \quad (4.3)$$

Wie auf der Abbildung [4.6 S. 48] dargestellt ist, wird die Pupille heller. Weiterhin ist zu bemerken, dass das projizierte Objekt (Glühbirne) dunkler wird.

Der zweite Operator nämlich der Maximum-Filter, gehört zu den Filtern im Ortsbereich, da der maximale Grauwert eines Bildpunktes in Abhängigkeit zu seiner Nachbarschaft N_x bestimmt wird. Hierbei werden z. B. die neun Grauwerte einer 3x3 Faltungsmaske nach der Größe des Grauwertes sortiert. Der Grauwert mit dem Maximum wird als Zielgrauwert eines Zielbildpunktes gewählt. Das Ergebnis des Maximum-Filters ist auf der Abbildung [4.7 S. 48] zu sehen. Hier ist die hervorgehobene Pupille deutlich zu erkennen. Wenn sich reflektierte Objekte auf der Pupille befunden hätten, die kleiner als die Pupille sind, wären sie durch den Maximum-Filter ausmaskiert bzw. ausgeblendet worden. Die Ausmaskierung der Glühbirne



Abbildung 4.6: Invertiertes Bild



Abbildung 4.7: Maximum Filter bei einer 5x5 Faltungsmaske

auf der Iris ist auf der Abbildung [4.7 S. 48] zu bemerken, während sie auf dem invertierten Bild, siehe Abbildung [4.6 S. 48], zu sehen ist.

Auf der Abbildung [4.8 S. 49] ist zu erkennen, dass schattige Bildregionen mit hervorgehoben worden sind. Dies ist zwischen dem Augenlid und dem Knochen der Augenhöhle zu sehen. Diese Bildregion hat denselben Grauwert wie die Pupille und kann unter Umständen deshalb mit der Pupille verwechselt und fälschlicherweise detektiert werden.

Die Verwendung des Maximum-Filters dient dem späteren Detektieren der Pupille und gewährleistet eine gewisse Stabilität des Verfahrens, die später erläutert wird.

Weiterhin ist ein Kantenbild erforderlich, in dem der Kreis der Pupille detektiert werden soll. Hierfür ist die Kantendetektion zuständig.

Kantendetektion: Ziel des Kantenbildes ist, die Grenzübergänge zwischen der Iris und der Pupille zu zeichnen. Dies soll durch eine Kante dargestellt werden und die Kante soll möglichst vollständig einen Kreis darstellen, sodass die Pupille zu erkennen ist. Denn je deutlicher und vollständiger der Kantenverlauf des Kreises ist, desto besser kann die Pupille mit ihrem jeweiligen Mittelpunkt detektiert werden.



Abbildung 4.8: Maximum Filter bei einer 5x5 Faltungsmaske des Quell-Bildes [3.5 S. 40]

Damit verstanden wird, wie eine Kante auf dem Kantenbild zustande kommt, wird sowohl auf die Grundlage als auch den jeweiligen erforderlichen Operator eingegangen.

Kanten spielen eine wichtige Rolle beim menschlichen Sehen und in der digitalen Bildverarbeitung. Durch eine Kante kann die Begrenzung zwischen den Bildobjekten dargestellt und die Objekte können besser erkannt werden. Daher ergibt sich folgende Anforderung [Gonzalez und Woods (2003) S. 704] an die Kantendetektion: Damit bei der Kantendetektion möglichst wenige Fehler auftreten, müssen die Kanten als solche erkannt werden und von Bildpunkten unterschieden werden, die zu keiner Kante gehören. Die Wichtigkeit der Lokalisation bezieht sich auf das Detektieren der Kanten an der richtigen Position. „Um ein Verschmieren einer Kante zu vermeiden, sollte auf eine Kante auch nur eine Antwort kommen.“ [Steinbecher (2003), S. 139]

Nach [Steinbecher (2003), S. 139] ist es bei der Kantendetektion besonders schwierig, zwischen den durch Rauschen bzw. durch die Störfaktoren verursachten zufälligen Signalveränderungen und den variierenden Grauwerten, die die tatsächlichen Kanten anzeigen, zu unterscheiden. Bezüglich einer solchen Anforderung ist eine generelle Vorgehensweise bei der Kantendetektion erforderlich. Beim ersten Schritt ist eine Glättung notwendig, damit bei der Detektion von Kanten die Artefakte des verrauschten Zielbildpunktes minimiert werden. Weiterhin wird ein Kantendetektionsalgorithmus angewendet, mit dem die Kanten detektiert werden. Damit die Kontur ein gutes Ergebnis erzielt, werden die Kantenpunkte ausgedünnt und kurze Kanten werden eliminiert. Damit eine geschlossene Kante repräsentiert wird, werden die Kantenpunkte mittels der Konturpunktverkettung verbunden.

Nun ist die Frage, wie Kanten grob detektiert werden können. Kanten können an dem Ort im Bild beschrieben werden, an dem sich die Grauwerte auf kleinem Raum und entlang einer ausgeprägten Richtung stark ändern. Je stärker sich die Grauwerte verändern, umso wahrscheinlicher ist es, dass sich dort eine Kante befindet. Tritt also eine Diskontinuität im Verlauf der Grauwerte auf, wird das Vorhandensein einer Kante angedeutet. Die Stärke der Grauwerte hinsichtlich der Distanz zwischen Bildpunkten entspricht der ersten Ableitung

einer diskreten Funktion [Gonzalez und Woods (2003) S. 693]. Angenommen, wie auf der Abbildung [4.9 S. 51] dargestellt ist, es handelt sich um eine eindimensionale, kontinuierliche Funktion $f(x)$, die eine Bildzeile darstellt. An dieser Bildzeile kann dann eine Kante detektiert werden, wenn eine Grauwertänderung vorliegt. Grauwertänderungen sind Funktionswertänderungen $f(x)$, sodass diese durch die erste Ableitung dieser diskreten Funktion f berechnet werden können. Somit werden die Kanten dort detektiert, „wo der Betrag der Ableitung lokal maximal ist.“ [Tönnies (2005) S. 117]

Ein Problem bezüglich der ersten Ableitung einer Kantendetektion ist, wenn Kanten genauso breit werden wie die Länge des zugehörigen Anstiegs in der Bildfunktion und ihre genaue Position dadurch schwierig zu lokalisieren ist. Aus diesem Grund lassen sich Kantenpunkte durch die zweite Ableitung einer Funktion finden, indem die Nulltedurchgänge dieser Funktion gesucht werden. Kanten bzw. Punkte mit hoher Grauwertänderung werden über den Gradienten bestimmt, wenn das Bild als Funktion $f(x, y)$ beschrieben wird. Bei der Bestimmung einer partiellen Ableitung in einem zweidimensionalen Bild ist die Rede vom Gradienten der Funktion f an der Stelle (x, y) . Partielle Ableitung meint, dass sich die Funktionswertänderungen nur auf die angegebene kollineare Richtung beziehen. Somit ist die Herstellung eines Gradientenbildes möglich, indem für jeden Bildpunkt die Differenzierung in zwei orthogonalen Richtungen durchgeführt wird. Die Berechnung der Gradienten mittels eines Vektors sieht wie folgt aus:

$$\nabla \mathbf{f}(\mathbf{x}, \mathbf{y}) = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \right)^T$$

Der Gradientenbetrag des Gradienten wird wie folgt berechnet:

$$G(x, y) = \sqrt{G_x^2 + G_y^2} \quad (4.4)$$

Die Richtung des Gradienten berechnet sich wie folgt:

$$G(x, y) = \arctan\left(\frac{G_x}{G_y}\right) \quad (4.5)$$

Canny-Operator: Einer der bekannten in der Praxis häufig verwendeten Kantenoperatoren, der oben erwähnte Anforderung erfüllt, ist der Canny-Operator. Der Canny-Operator, wie aus der Literatur hervorgeht [Canny (1986)], [Meisel (2008) Script-Teil: 4 S. 22-27], [Gonzalez und Woods (2003) S. 719-725], erzielt erfolgsversprechende Ergebnisse, sodass auf diesen Operator ausführlicher eingegangen wird.

Auf der ersten Stufe, wie auf der Abbildung [4.10 S. 52] dargestellt ist, wird die Bildfunktion

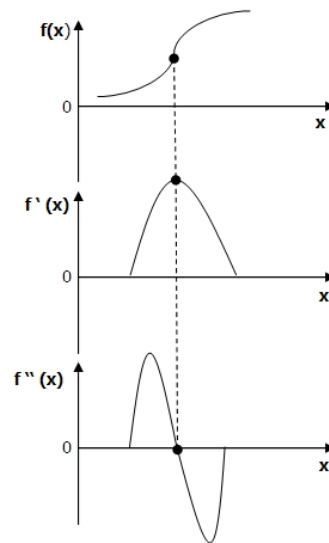


Abbildung 4.9: Erste und zweite Ableitung einer Funktion

$f(x, y)$ mit den Richtungsableitungen der Gaussfunktion in x- und y-Richtung gefaltet.

$$G_x = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.6)$$

Anhand dieser Gauss-Faltung wird die Glättung durchgeführt, indem die Nachbarn - z. B. N_8 eines Bildpunktes berücksichtigt werden. Dabei sind die Bildpunkte mit den größten Grauwerten im Zentrum, sodass diese einen stärkeren Einfluss auf den Zielintensitätswert ausüben, als die niedrigen Grauwerte, die weiter entfernt vom Zentrum liegen. Der Ergebniswert wird zuletzt normalisiert, indem dieser durch die Summe der Faltungsmaskenwerte geteilt wird. Das Ergebnis der Anwendung des Gaussfilters ist die Minimierung des Rauschens.

Auf der zweiten Stufe wird das geglättete Eingangsbild bezüglich der x- und y-Achse differenziert. Für die Differenzierung ist ein Grauwertbild erforderlich. Wenn an einer Stelle zwischen zwei Bildpunkten ein starker Kontrast der Grauwerte vorhanden ist, liegt dort eine Kante vor. Gleichzeitig befindet sich dort ein großer Steigungswert, vorausgesetzt es handelt sich um eine Funktion $f(x, y)$ mit der y-Koordinate und der x-Position als x-Koordinate. Indem die Nachbarpunkte x_{i-1} und x_{i+1} betrachtet werden, wird sich an die Steigung eines Punktes x_i angenähert. Dies ist nichts anderes als die Ableitung an der Stelle von x_i . Die Steigung wird mit der folgenden Faltungsmaske $[-1|0|1]$ berechnet.

Weiterhin, um das Rauschen besser zu unterdrücken, wird die dritte Ableitung benutzt. Um die Steigung einer Zeile zu berechnen, werden die zwei Zeilen unter und über der betreffen-

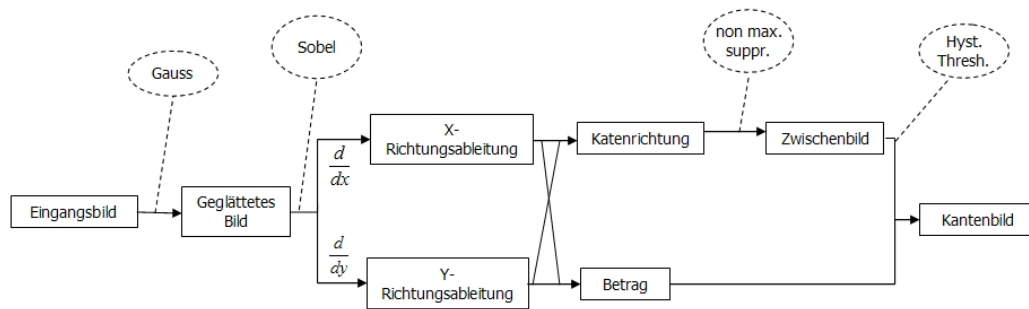


Abbildung 4.10: Ablaufdiagramm des Canny-Operators

den Zeile mitberücksichtigt. Dies entspricht den Faltungsmasken des Sobel-Operators.

$$\mathbf{G}_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\mathbf{G}_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{pmatrix}$$

Mit dem Sobel-Operator wird das Eingangsbild in beide Richtungen separat, horizontal mit der Filtermaske G_y und vertikal mit der Filtermaske G_x , durchlaufen, um eine Stärkung der Kanten zu erreichen. Nachdem das Bild mit dem Sobel-Operator gefaltet worden ist, erfolgt die Berechnung des Gradienten. Der Gradient zeigt die Richtung an, die mit folgender Gleichung berechnet wird: [4.5 S. 50].

Aus dieser Berechnung folgen vier Richtungen: 0° für die Horizontale, 90° für die Vertikale, 45° für die steigende Diagonale und 135° für die fallende Diagonale. Damit der Wert des berechneten Winkels diskretisiert wird, wird dieser dem in der Abbildung [4.11 S. 53] dargestellten Bereich zugeteilt.

Damit die Entscheidung getroffen werden kann, welcher Bildpunkt zur Kante gehört, wird der „nonmaximum suppression“ angewendet. Hierbei werden die Kanten abgetastet und die Grauwerte der Bildpunkte dieser Kante werden auf ein Maximum hin überprüft. Wenn kein Maximum vorliegt, bedeutet das, dass der Bildpunkt nicht zur Kante gehört, und dieser Wert des Bildpunktes kann gleich Null gesetzt werden. Diesbezüglich enthält die entstandene Kantenlinie eine Breite von einem Bildpunkt bzw. Pixel.

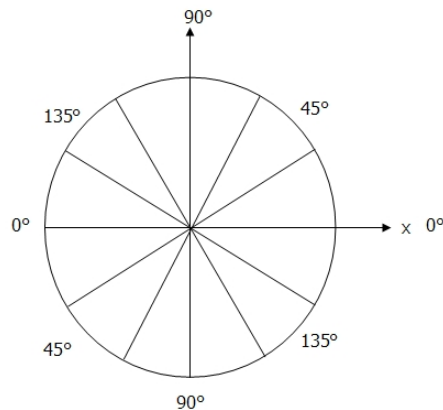


Abbildung 4.11: Gradbereiche zur Diskretisierung der Kantenorientierungen

Damit Lücken in den Kanten verhindert werden, wird als letztes der „Hysterese-Thresholding“ durchgeführt. Hierbei stellt der Canny-Operator zwei einstellbare Schwellenwerte „minimum Threshold“ und „maximum Threshold“ zur Verfügung, mit denen entschieden wird, welche Kante zum Kantenbild gehören soll. Wenn eine Kante mit ihrer Kantenstärke den oberen Schwellenwert überschreitet, gehört diese Kante zum Kantenbild. Wenn eine Kante mit ihrer Kantenstärke unter dem unteren Schwellenwert liegt, gehört diese Kante nicht zum Kantenbild. Falls die Kantenstärke einer Kante innerhalb der beiden Schwellenwerte liegt, wird anhand der Nachbarn entschieden, ob die Kante zum Kantenbild gehören soll.

Nachdem die Grundlagen als auch der Canny-Operator erläutert wurden, wird auf das Ergebnis des Canny-Operators eingegangen. Wie schon erwähnt wurde, soll der Kreis der Pupille möglichst vollständig gezeichnet werden. Hierbei sollen die Kanten der irrelevanten Bildobjekte als auch die Kanten der Störfaktoren (Schatten) möglichst nicht mitgezeichnet werden. Des Weiteren soll der Mittelpunkt des Kreises der Mitte der Pupille entsprechen. Hierfür stellt der Canny-Operator insgesamt vier Einstellungsparameter zur Verfügung - Faltungsmaske, Variance, „minimum Threshold“ und „maximum Threshold“ - mit denen das Ergebnis eines Kantenbildes beeinflusst werden kann. Die Ergebnisse des Canny-Operators bei folgender Einstellung der Parameter: Faltungsmaske = 9×9 , Variance = 9, 8, Schwellenwert-Min = 0, 05 und Schwellenwert-Max = 0, 8, sind auf der Abbildung [4.3 S. 46] und auf der Abbildung [4.5 S. 47] zu sehen.

Weiterhin besteht die Schwierigkeit bei dem Canny-Operator darin, dass die Einstellungsparameter richtig eingestellt sein müssen, nämlich so, dass die Kanten des Kreises möglichst vollständig detektiert und die anderen Kanten der irrelevanten Objekte unterdrückt bzw. nicht detektiert werden sollen. Je mehr Kanten der irrelevanten Objekte vorhanden sind - z. B. Objekte, die sich außerhalb des Kreises befinden - desto ungenauer wird der Kreis der Pupille detektiert. Hierbei kann es vorkommen, dass eine Menge von Kanten außerhalb der

Pupille vorkommen, die fälschlicherweise als Pupille detektiert werden, weil sie kreisförmig verlaufen. Hierzu folgende Abbildungen [4.12 S. 54 4.13 S. 54 vgl. mit 4.14 S. 55 4.15 S. 55] ansehen und vergleichen. Wie auf der Abbildung [4.13 S. 54] zu erkennen ist, wird die Pupille durch eine kleine Veränderung des Max-Schwellenwert ungenau detektiert.



Abbildung 4.12: Kantenbild des Bildes [3.6 S. 40] bei Faltungsmaske: 9x9, Variance: 9,9, Schwellenwert-Max: 0,01, Schwellenwert-Min: 0,8

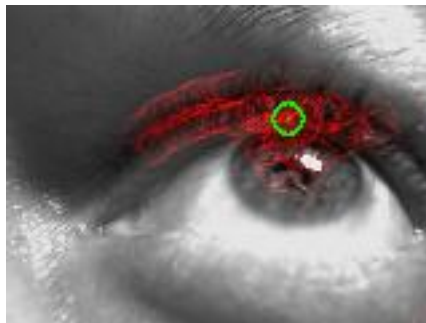


Abbildung 4.13: Detektierter Kreis zum Kantenbild [4.12 S. 4.12] bei Faltungsmaske: 9x9, Variance: 9,9, Schwellenwert-Max: 0,01, Schwellenwert-Min: 0,8

Nachdem nun das Ergebnis des Kantenbildes vorliegt und erläutert wurde, ist es offensichtlich, dass die Pupille, wie auf der Abbildung [4.13 S. 54] zu sehen ist, nicht immer richtig detektiert werden kann. Daher geht die Überlegung dahin, dass ein Bildpunktoperator benötigt wird, mit dem die Intensität der Grauwerte der irrelevanten Kanten abgeschwächt werden kann.

Damit dieses Ziel erreicht werden kann, wird das Bild gebraucht, das mit dem Maximum-Filter gefaltet wurde, nämlich das Maximum-Bild auf der Abbildung [4.7 S. 48]. Diesbezüglich wird der logische UND-Operator angewendet, womit das Kantenbild mit dem Maximum-Bild verknüpft wird. Dies kann wie folgt dargestellt werden:

$$f_{out}(x, y) = f_2(x, y) \text{ AND } f_1(x, y) \quad (4.7)$$



Abbildung 4.14: Kantenbild des Bildes [3.6 S. 40] bei Faltungsmaske: 9x9, Variance: 9,9, Schwellenwert-Max: 0,05, Schwellenwert-Min: 0,8



Abbildung 4.15: Detektierter Kreis zum Kantenbild [4.14 S. 4.14] bei Faltungsmaske: 3x3, Variance: 0,5, Schwellenwert-Max: 0,5, Schwellenwert-Min: 0,1

Wie aus der Abbildung [4.16 S. 56] hervorgeht, wurde die Intensität vieler irrelevanten Kanten, die außerhalb der Pupille liegen, abgeschwächt. Damit ist nun gewährleistet, dass bei der kommenden Pupillendetektion tatsächlich nur die Kanten der Pupille mit einbezogen werden.

Anhand der Abbildung [4.14 S. 55] ist zu bemerken, dass bei einigen Bildern der Kreis kaum zu erkennen ist. Aus diesem Grund sind für das weitere Detektieren der Pupille besondere Maßnahmen erforderlich. Hiermit sind Operatoren gemeint, die in der Lage sind die Pupille zu detektieren, obwohl nur ein Quadrant des Kreises auf dem Kantenbild gezeichnet ist. Das bedeutet, dass anhand einiger Bildpunkte innerhalb des gezeichneten Kreises im Kantenbild die Pupille mit Hilfe einer Transformation zuverlässig detektiert werden kann. Eine solche Transformation, die diese Anforderung erfüllen kann, folgt im kommenden Abschnitt.



Abbildung 4.16: Verknüpfung des Kantenbild mit dem Maximum-Bild

4.2.2 Merkmalsextraktion

Wie aus der Abbildung [4.1 S. 42] hervorgeht, folgt die Merkmalsextraktion. Eine wichtige Eigenschaft der Merkmalsextraktion ist, dass das gesuchte Bildobjekt automatisch detektiert wird. Damit dies möglich ist, sind Verfahren bzw. Transformationen erforderlich, die anhand der Merkmale, das gesuchte Bildobjekt bzw. die Pupille identifizieren. Daraufhin ist es möglich, Bildobjekte in Kategorien einzuordnen. Die Merkmale, wie sie am Anfang erwähnt wurden, beinhalten oft komprimierte Informationen. Diese Informationen geben den Bildinhalt oder die semantische Bildinformation wieder.

Eine Voraussetzung für die Merkmalsextraktion ist, dass das Bild - wie schon erwähnt - nach dem gesuchten Bildobjekt segmentiert ist bzw. ein Kantenbild darstellt. Die Merkmale, die bereits als Vektor definiert sind, korrelieren mit der Form eines gesuchten Bildobjektes. Im Zusammenhang mit dem Detektieren der Pupille stellt die Form einen wichtigen Anhaltspunkt dar.

Bei diesem Anhaltspunkt geht es darum, die Form anhand der Merkmale so zu parametrisieren, dass der Kreis der Pupille detektiert werden kann. Eine der erfolgreichen Transformationen, die die Erkennung von parametrisierbaren Formen erlaubt, ist die Hough-Transformation (HT). Die bekannteste Version der HT dient dem Detektieren von Geraden und Kreisen. Das Prinzip kann auch zum Detektieren anderer parametrisierbaren Formen verwendet werden. Wobei hier, je nach parametrisierbarer Form und Anwendung, die Rechenleistung ein entscheidender Faktor ist. Je komplexer eine parametrisierbare Form mit vielen Merkmalen ist, desto rechenzeitintensiver ist das Detektieren im Parameterraum.

Die HT ist bekannt für ihre Zuverlässigkeit, da sie unauffällig gegenüber Rauschen sowie gegenüber solchen Störfaktoren ist, die im Zusammenhang mit dem Detektieren der Pupille vorkommen. Weiterhin hat sich die HT bei vielen parametrisierbaren Formen bewährt, die innerhalb eines Kantenbildes nicht vollständig bzw. mit Lücken von einem Kantendetektor gezeichnet sind.

Damit die Funktionsweise der HT verstanden wird, wird zunächst auf die Grundform der HT eingegangen. Weiterhin wird anhand dieser Grundlage die HT für Kreise beschrieben, mit der der Kreis der entsprechenden Pupille detektiert werden soll. Hierzu wird auf die benötigten Algorithmen, wie z. B. Breseham, zurückgegriffen.

Bei der Grundform der HT geht es darum, im Kantenbild die Gerade zu detektieren, die möglichst viele Bildpunkte in diesem Kantenbild erfasst. Eine Gerade kann mathematisch, siehe Gleichung [4.8 S. 57], durch zwei Parameter - x und b - beschrieben werden. Hierzu stellt m die Steigung und b den Schnittpunkt mit der y -Achse dar.

$$y = mx + b \quad (4.8)$$

Das Ziel ist nun, mit den Parametern m und b die Gerade zu detektieren. Hierzu soll festgestellt werden, wie viele Bildpunkte auf der Geraden liegen. Bezüglich dieser Anforderung, wie auf der Abbildung [4.17 S. 57] dargestellt ist, wird die Gleichung [4.9 S. 57] transformiert.

$$b = -(x)m + y \quad (4.9)$$

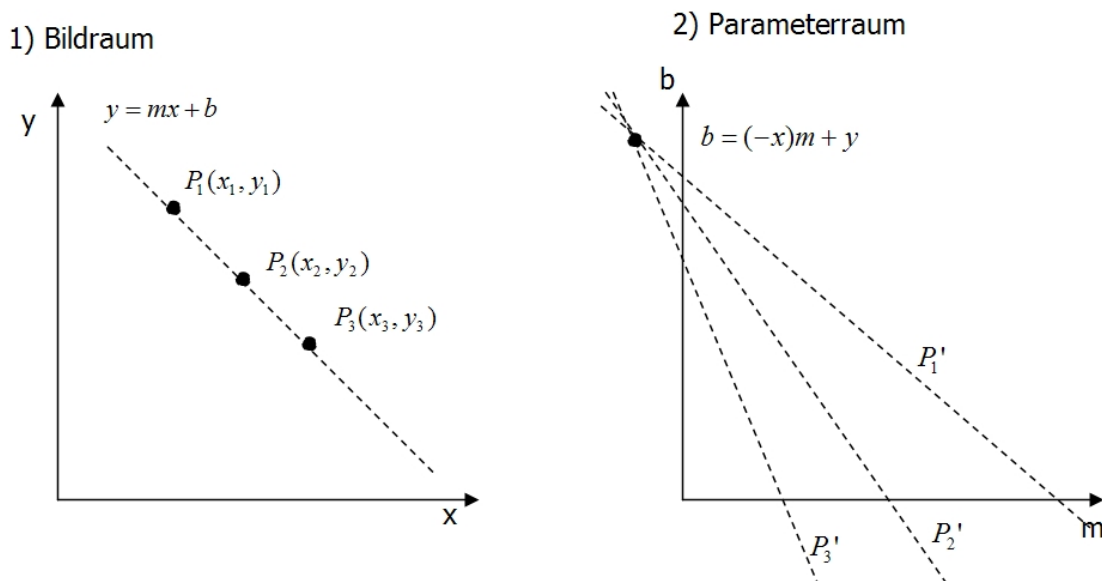


Abbildung 4.17: HT für Geraden

Wenn anhand dieser Gleichungen [4.9] die Parameter m und b in dem Parameterraum aufgespannt werden, ist zu bemerken, dass jeder Kantenbildpunkt $P(x_n, y_n)$ mit einer Geraden im m - b -Raum korrespondiert. Dieser Raum ist unter dem Namen Hough-Raum bekannt. An dem Punkt im Parameterraum, wo sich die korrespondierenden Gleichungen am meistens

schneiden, also ein Bündel von Geraden bilden, befindet sich der interessante Anhaltspunkt. Dieser Anhaltspunkt weist darauf hin, dass sich auf dem Kantenbild die gesuchte Gerade mit der größten Anzahl von Bildpunkten befindet. Wenn in dem Parameterraum ein Punkt vorhanden ist, bei dem sich eine höchste Anzahl n Geraden schneiden, wird auf dem Kantenbild eine Linie mitgezeichnet, die genau n Punkte enthält. Damit festgestellt wird, wie viele Bildpunkte auf der Geraden im Kantenbild liegen, werden die Geraden an der Position im Parameter, wo sich alle Geraden schneiden, inkrementiert. Dies wird in einem sogenannten Akkumulator-Array festgehalten. Aus dem Grund, dass die Steigung $m = \infty$ für vertikale Geraden ist, wird diese kaum in der Praxis angewendet. Denn hierdurch wird der Akkumulator-Array unendlich. An dieser Stelle wird die sogenannte „hess'sche Normalform (HNF)“ der Geradengleichung eingesetzt.

$$r = x * \cos \theta + y * \sin \theta \quad (4.10)$$

Mittels der HNF wird der Hough-Raum mit den Parametern, r für den Radius und θ für den Winkel, so linear quantisiert, dass der Hough-Raum diese Parameter im Raum aufspannen kann. Somit steht jeder vorhandene Bildpunkt im Kantenbild für die Relation für den Winkelbereich $0 \leq \theta < \pi$. Wenn es darum ginge den Bildpunkt, der in der Mitte des Bildes bei M -Spalten N -Zeilen liegt, zu referenzieren, wäre der Bereich für den Radius auf die Hälfte der Bilddiagonale beschränkt.

$$r_{max} = \frac{1}{2} \sqrt{M^2 + N^2} \quad (4.11)$$

Der nächste Schritt bezieht sich auf die Auswertung des maximalen Wertes im Akkumulator-Array. Hierbei sollte, trotz der verteilten Schnittpunkte von mehreren Geraden in dem Parameterraum, der maximale Wert lokalisiert werden. Für die Auswertung können verschiedene Verfahren, wie die Schwellenwertoperation oder Non-Maximum-Suppression, verwendet werden. In dem Akkumulator-Array können alle Werte, die kleiner als der Schwellenwert sind, z. B auf Null gesetzt werden. Mit dem Non-Maximum-Suppression können im Akkumulator-Array alle Werte, die kein Maximum aufweisen, unterdrückt werden. Dabei wird jeder Wert einer Zelle mit den Werten der Nachbarzellen verglichen.

Damit diese Transformation effizienter wirkt, sind verschiedene Möglichkeiten vorgesehen. Eine Möglichkeit, womit die Kante mit den größten Grauwerten im Kantenbild detektiert werden kann, ist die Addition von Grauwerten. Anstatt den Wert einer Akkumulator-Zelle zu inkrementieren, ist es möglich, den Grauwert eines Bildpunktes im Kantenbild aufzuaddieren. Der Grauwert kann dann aufaddiert werden, wenn es zu den korrespondierenden Kantenbildpunkten einen Schnittpunkt gibt, bei dem sich die größte Anzahl von Geraden schneiden. Zusätzlich zur Grauwertaddition, kann die lokale Kantenrichtung berücksichtigt

werden. Mit Hilfe der Kantenrichtung kann die Anzahl der Zellen im Akkumulator-Array dadurch eingeschränkt werden, indem im Parameterraum entlang der θ -Achse abhängig von der Kantenrichtung die Berechnung der Zellen durchlaufen wird.

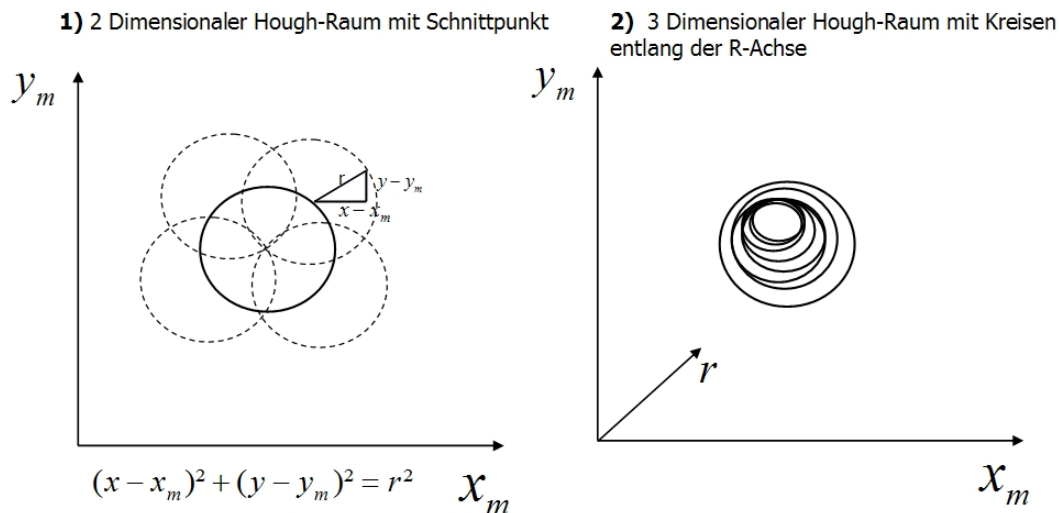


Abbildung 4.18: HT für Kreise mit dem jeweiligen Parameterraum

Bezogen auf die HT für Kreise, wozu das Detektieren von Kreisen im Kantenbild dienen soll, kann die HT analog von der Geraden übernommen werden. Nach der Gleichung [4.1 S. 43] sind insgesamt drei Parameter, nämlich x_m -, y_m -Koordinate und der r Radius vorhanden, anhand deren der Parameterraum aufgespannt werden kann. Die x_m - und y_m -Koordinate geben den Mittelpunkt des Kreises, an wie auf der Abbildung [4.18 S. 59].

Im Gegensatz zur HT für Geraden existiert zwischen den Koordinaten im Parameterraum keine einfache funktionale Abhängigkeit. Bei der HT für Geraden ist die funktionale Abhängigkeit dadurch gegeben, dass ein Kantenbildpunkt von dem Winkel und von dem Radius im Parameterraum abhängig ist. Bei der HT für Kreise ist diese funktionale Abhängigkeit nicht gegeben. Deswegen kann anhand einer Kombination der Parameterwerte nicht entschieden werden, ob ein Bildpunkt zum Kreis gehört.

Damit ein Kreis innerhalb des Kantenbildes detektiert werden kann, wird für ein Kantenbildpunkt mit seiner Mittelpunktposition (x_m, y_m) im Parameterraum ein Kreis gebildet. Dieser wird durch den Bresenham-Algorithmus, der im kommenden Unterkapitel [4.2.2.1] erklärt wird, erzeugt. Die Erzeugung des Kreises wird im Algorithmus [1] Zeile 1-8 angedeutet. Auf den Kreisbogen wird der Grauwert dieses Mittelpunktes (x_m, y_m) gelegt. Dies ist deswegen notwendig, da die funktionale Abhängigkeit nicht vorhanden ist. Nachdem der Parameterraum mit relevanten Bildpunkten aufgefüllt ist, geht es darum, die Kreise zu detektieren. Aus Effizienzgründen wird der Ebene entlang der r -Radius für $r = r_{min} \dots r_{max}$ ein Kreis mit dem

Mittelpunkt x_m, y_m durchlaufen, siehe Algorithmus [1] Zeile 13. Somit ergibt sich eine kegel-förmige Fläche, siehe Abbildung [4.18 2) S. 59]. Hierbei wird für jeden Kreis der Grauwert aufaddiert, siehe Zeile 15. Das Maximum entspricht dem Schnittpunkt, der auf den Mittelpunkt des detektierten Kreises, siehe Abbildung [4.18 1) S. 59] hinweist. Damit entschieden wird, ob ein Punkt zum Kreis gehört, muss die Bedingung, siehe Zeile: 6, erfüllt sein. Eine weitere Effizienzsteigerung, wie bei der HT für Geraden mit dem Schwellenwert gezeigt wurde, kommt in diesem Algorithmus auch vor. In Zeile 3 werden durch einen Schwellenwert, die relevanten Kantenbildpunkte ausgewählt.

Anhand der oben erwähnten HT für Kreise wird der Kreis der Pupille detektiert. Hierzu wird mit den definierten Merkmalen bzw. dem Vektor $\vec{v} = [x_n, y_n]^T$ der Parameterraum zwei-dimensional aufgespannt. Aus Effizienzgründen und aus dem Grund, dass der Radius der Pupille bekannt ist, wird dieser zweidimensionale Parameterraum gewählt.

Algorithm 1 Modifizierter HT-Circle

```

1: for all image coordinates  $f(x, y)$  do
2:    $GrayValue \leftarrow f(x, y)$ 
3:   if  $GrayValue > Treshold$  then
4:     for  $BresemList(b_x, b_y)$  do
5:        $HoughSpace(x + (b_x \cdot x), y + (b_y \cdot y)) + = f(x, y)$ 
6:     end for
7:   end if
8: end for
9: for all image coordinates  $f(x, y)$  do
10:   $spaceMaxGrayValue \leftarrow HoughSpace(x, y)$ 
11:  if  $spaceMaxGrayValue > tmpMax$  then
12:     $tmpMax \leftarrow spaceMaxGrayValue$ 
13:    for  $\delta(x, y, r) < Radius$  do
14:      if  $(\delta(x)^2 + (\delta(y)^2 = r^2)$  then
15:         $MaxGrayValue + = HoughSpace(x, y)$ 
16:      end if
17:    end for
18:  end if
19:  if  $tmpMax < MaxGrayValue$  then
20:     $PupilPoint(x, y)$ 
21:  end if
22: end for

```

Nachdem durch den logischen UND-Operator die relevanten Kanten zwischen der Iris und der Pupille hervorgehoben sind, wird die Pupille mit der HT detektiert. Bevor die Bildpunkte der relevanten Kanten ausgewertet werden, wird vorerst ein Schwellenwert eingeführt. Mit

diesem Schwellenwert werden nur die Kantenbildpunkte dem Parameterraum zugeführt, die unter anderem die gesuchten Objekte - nämlich die Pupille - repräsentieren. Hierzu werden anhand des generierten Bresenham-Kreises die Kreise im Parameterraum gebildet. Weiterhin werden mit Hilfe der HT für Kreise die Bildpunkte im Parameterraum ausgewertet. Hierbei werden die Bildpunkte des Parameterraums entlang des eingegebenen Radius durchlaufen. Die Grauwerte der Bildpunkte, die mit dem Mittelpunkt korrelieren, werden aufaddiert. Dadurch wird das Maximum dort berechnet, wo sich die meisten Kreise schneiden. Anhand der lokalisierten Position x, y dieses maximalen Wertes, wird der Kreis auf der detektierten Pupille gezeichnet.

Das Ergebnis kann auf der Abbildung [4.19 S. 61] nachvollzogen werden. Das jeweilige Parameterraumbild ist auf der Abbildung [4.20 S. 61] zu sehen. Wie schon erwähnt wurde, hängt das Detektieren der Pupille mit dem dazugehörigen Mittelpunkt sehr stark von dem UND-Operator erzeugten Kantenbild ab. Je mehr die Kanten der irrelevanten Objekte in der Intensität geschwächt sind, desto genauer kann der Kreis mit dem jeweiligen Mittelpunkt detektiert werden. Weitere Ergebnisse der detektierten Pupille, können im Abschnitt [6.3 S. 74] nachgelesen werden.

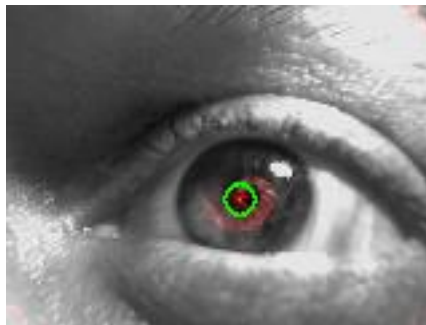


Abbildung 4.19: Detektierte Pupille des Quell-Bildes [3.4 S. 39]



Abbildung 4.20: Der Hough-Raum zur detektierten Pupille des Bildes [4.19 S. 61]

4.2.2.1 Breseham Algorithmus

Damit die Kreise generiert werden, wird ein Algorithmus benötigt, der eine lineare Berechnung durchführt. Ein in der Computergrafik erfolgsversprechender Algorithmus ist der nach dem Erfinder benannte Bresenham-Algorithmus. Dieser Algorithmus dient, wie aus der Literatur [Bungartz u. a. (2002) S. 30-37] hervorgeht, dem Zeichnen von Kreisen auf dem Bildschirm. Der Algorithmus beruht auf einer ganzzahligen Arithmetik und weist zu n Bildpunkten ein lineares Verhalten auf. Diese Vorteile werden für die Generierung von Kreisen genutzt.

Angenommen in einer Bildregion befindet sich ein Kreis mit dem Mittelpunkt $M(x, y)$, dann muss entschieden werden, welche Bildpunkte in dem Kreis liegen. Bezogen auf die Gleichung [4.2 S. 43], ist diese Berechnung durch die jeweiligen Operationen (Gleitpunktoperation-Wurzel, Addition, etc.) rechenintensiv, sodass dies hinsichtlich der Rechenzeit nicht effizient ist. Aus diesem Grund folgt die Überlegung, dass zuerst die Bildpunkte, die im ersten Quadranten auf dem Kreisbogen liegen, betrachtet werden.

In diesem Zusammenhang wird nur $1/8$ des Kreisbogens im ersten Quadranten in dem Bereich $y \geq x$ betrachtet, siehe Abbildung [4.21 a) S. 62]. Die anderen Bildpunkte auf der anderen Hälfte des Quadranten werden durch Spiegelung gewonnen. Auf diese Weise werden erstmal folgende Bildpunkte (x, y) , (y, x) im ersten Quadranten und dann die sechs dazugehörigen Bildpunkte $(y, -x)$, $(x, -y)$, $(-x, -y)$, $(-y, -x)$, $(-y, x)$ und $(-x, y)$ in den übrigen drei Quadranten segmentiert.

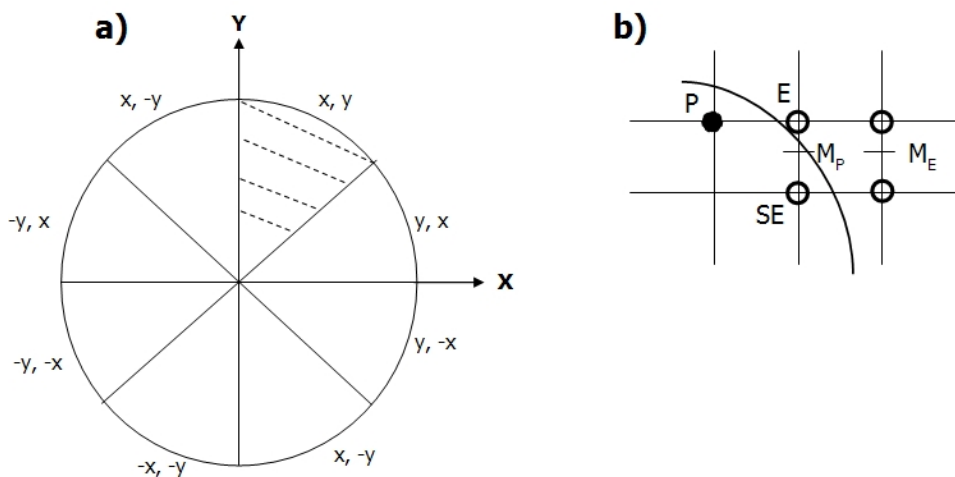


Abbildung 4.21: Kreis und fortgesetzte Pixelwahl beim Bresenham-Algorithmus

Der nachfolgende Bildpunkt, ausgehend von dem Bildpunkt $P = (x_p, y_p)$, wie auf der Abbildung [4.21 b) S. 62] dargestellt ist, wird aus den beiden potenziellen Nachfolgern

$E = (x_p + 1, y_p)$ und $SE = (x_p + 1, y_p - 1)$ ausgesucht. Der Bildpunkt, der sich am nächsten am Kreisbogen befindet, wird als Nachfolger von P ausgewählt, indem die Relativlage des Mittelpunktes $M_p = (x_p + 1, y_p - 1/2)$ zwischen den beiden potentiellen Nachfolgern E und SE zum Kreisbogen ermittelt wird. Falls sich M_p innerhalb des zu zeichnenden Kreises befindet, dann wird E als Nachfolger ausgewählt und segmentiert. Falls sich M_p außerhalb des zu zeichnenden Kreises befindet, dann wird SE als Nachfolger ausgewählt und segmentiert. Wenn M_p genau auf der Kreislinie ist, dann wird entweder E oder SE ausgewählt. Damit man die Lage von M_p relativ zum Kreis herausfinden kann, wird folgende Funktion $F(x, y)$ benötigt:

$$F(x, y) = x^2 + y^2 - r^2 \quad M = (x; y) \quad (4.12)$$

Bezüglich dieser Funktion [4.12 S. 63] gilt, dass $F(x, y)$ für alle Bildpunkte auf dem Kreis gleich Null ist. Somit kann anhand des Vorzeichens von F die Relativlage des Bildpunktes (x, y) zum betrachteten Kreis wie folgt ermittelt werden:

$$F(x, y) = \begin{cases} = 0 : (x, y) & \text{liegt auf dem Kreis.} \\ > 0 : (x, y) & \text{liegt außerhalb des Kreises.} \\ < 0 : (x, y) & \text{liegt innerhalb des Kreises.} \end{cases} \quad (4.13)$$

Da die Entscheidungsvariablen D_p vorerst nicht auf ganzen Zahlen beruhen,

$$D_p := F(M_p) = (F(x_p + 1, y_p - 1/2)) = ((x_p + 1)^2 + (y_p - 1/2))^2 - r^2$$

lassen sich zwei Fälle unterscheiden:

- **Fall 1:** ($D_p < 0$) (M_p liegt innerhalb des Kreises: Der zu setzende Nachfolgebildpunkt ist E).

$$\begin{aligned} & D_E = F(M_E) \\ & = (F(x_p + 2, y_p - 1/2)) \\ \Rightarrow & = ((x_p + 2)^2 + (y_p - 1/2))^2 - r^2 \\ & = D_p + \Delta_E \\ & \text{mit } \Delta_E := 2x_p + 3 \end{aligned}$$

- **Fall 2:** ($D_p \geq 0$) (M_p liegt außerhalb des Kreises oder auf dem Kreis: Der zu setzende

Nachfolgebildpunkt ist SE .

$$\begin{aligned} D_{SE} &= F(M_{SE}) \\ &= (F(x_p + 2, y_p - 3/2)) \\ \Rightarrow &= ((x_p + 2)^2 + (y_p - 3/2)^2) - r^2 \\ &= D_p + \Delta_{SE} \\ &\text{mit } \Delta_{SE} := 2x_p - 2y_p + 5 \end{aligned}$$

Befindet sich M_p auf dem Kreis, ergibt sich willkürlich SE als Nachfolger.

Die ganzzahligen Inkremente Δ_E und Δ_{SE} sind jedoch nun für die Entscheidungsvariable D_p linear in x_p und y_p . Somit wird der Startbildpunkt D_{Start} mit dem Startbildpunkt $(0, r)$ wie folgt festgelegt:

$$\begin{aligned} D_{Start} &:= F(1, r - 1/2) \\ &= 5/4 - r \end{aligned}$$

Durch die Substraktion von $1/4$ von D_{Start} ist D_p wieder ganzzahlig, womit die Auswahl der Bildpunkte unverändert bleibt. Das bedeutet, dass der Startwert ($D_{Start} := 1 - r$) und die Inkremente D_p immer ganzzahlig sind und damit auch die Entscheidungsvariable D_p .

Die Berechnung der Inkremente Δ_E und Δ_{SE} erfolgt inkrementell, da sich die Inkremente linear zu x_p und y_p verhalten. Außerdem werden sie durch die Differenzen mittels zweiter Ordnung berechnet. Dadurch, dass sie ohne Multiplikation berechnet werden, kann sich die Effizienz weiterhin steigern. Diesbezüglich ergibt sich für den Kreis:

$$\begin{aligned} D_p &\xrightarrow{\Delta_E \text{ und } \Delta_{SE} \text{ linear}} D_{SE}, D_{SE}) \\ \Delta_E &= 2x_p + 3, \Delta_{SE} = 2(x_p - y_p) + 5 \end{aligned}$$

Durch konstante Inkremente werden Δ_E und Δ_{SE} aktualisiert:

$$\begin{aligned} D_{E,P} &\xrightarrow{\text{konstante Inkremente}} \Delta_{E,E}, \Delta_{E,SE}) \\ D_{SE,P} &\xrightarrow{\text{konstante Inkremente}} \Delta_{SE,E}, \Delta_{SE,SE}) \end{aligned}$$

Durch das Vorzeichen der ganzzahligen Entscheidungsvariable D_p ergeben sich wiederum zwei unterschiedliche Fälle:

- **Fall 1:** ($D_p < 0$)
(E) ist der Nachfolgebildpunkt.

$$\begin{aligned} \Rightarrow \quad \Delta_{E,E} &= 2(x_p + 1) + 3 \\ &= \Delta_{E,P} + 2, \\ \Delta_{SE,E} &= 2(x_p + 1) - 2y_p + 5 \\ &= \Delta_{SE,P} + 2 \end{aligned}$$

- **Fall 2:** ($D_p \geq 0$) (SE) ist der Nachfolgebildpunkt.

$$\begin{aligned} \Rightarrow \quad \Delta_{E,SE} &= 2(x_p + 1) + 3 \\ &= \Delta_{E,P} + 2, \\ \Delta_{SE,SE} &= 2(x_p + 1) - 2y_p + 5 \\ &= \Delta_{SE,P} + 4 \end{aligned}$$

Dadurch, dass der Startbildpunkt $0, r$ ist, werden für $\Delta_E = 2x_p + 3$ bzw. $\Delta_{SE} = 2y_p + 5$ folgende Startwerte $\Delta_{E,Start}$ bzw. $\Delta_{SE,Start}$ ausgewählt. Somit ergibt sich folgende Initialisierung:

$$\begin{aligned} \Rightarrow \quad \Delta_{E,Start} &:= 3 \\ \Delta_{E,P} &:= -2r + 5 \end{aligned}$$

Dementsprechend folgt der modifizierte Algorithmus [2]:

Algorithm 2 Bresenham-Circle

```
1: x := 0
2: y := r
3: d := 1 - r
4: deltaE := 3
5: deltaSE := -2 * r + 5
6: PointList.clear
7: Call segmentationCirclePixel(x, y)
8: while y > x do
9:   if d < 0 then
10:    d = d + deltaE;
11:    deltaE := deltaE + 2
12:    deltaSE := deltaSE + 2
13:    x := x + 1
14:  else
15:    d = d + deltaSE
16:    deltaE := deltaE + 2
17:    deltaSE := deltaSE + 4
18:    x := x + 1
19:    y := y + 1
20:  end if
21:  Call segmentationCirclePixel(x, y)
22: end while
23: void segmentationCirclePixel(x, y)
24: PointList.push(point(x, y))
25: PointList.push(point(-x, y))
26: PointList.push(point(x, -y))
27: PointList.push(point(-x, -y))
28: PointList.push(point(y, tmp_x))
29: PointList.push(point(-y, x))
30: PointList.push(point(y, -x))
31: PointList.push(point(-y, -x))
32: end segmentationCirclePixel
```

5 Ermittlung der Position einer Augenbewegung

Dieses Kapitel beinhaltet die Kalibrierung, mit der die Positionen der Augenbewegungen auf dem Bildschirm ermittelt werden sollen. Die Augenbewegungen beziehen sich auf die Sakkaden mit den jeweiligen Fixationen, die bei der Anwendung immer wieder vorkommen. Mit der Kalibrierung soll die Beziehung zwischen den Bildschirmpunkten und der detektierten Pupille hergestellt werden, um die Augenbewegungen des Anwenders herzuleiten.

5.1 Kalibrierung und Ermittlung der Position auf dem Bildschirm einer Augenbewegung

Wie aus der Literatur [[Duchowski \(2003\)](#) S. 77-82], [[Ohno und Mukawa \(2004\)](#)] hervorgeht, existieren diverse mathematische Verfahren, mit denen die Augenbewegungen auf dem Bildschirm erfasst werden können. Je nach System, ob monitor- oder kopfbasiert, kann das entscheidende Verfahren eingesetzt werden, wobei hier die Komplexität ($O(n)$) bezogen auf die Rechenzeit des Verfahrens nicht vernachlässigt werden darf. Da dieses Thema sehr umfangreich ist und der projizierte infrarote Referenzpunkt fehlt, wird hier ein einfaches Verfahren eingesetzt, das im Folgenden dargestellt wird. Nach [[Duchowski \(2003\)](#) S. 77-82] existiert ein lineares Verfahren, bei dem anhand der Kalibrierungspunkte und der Position der Pupille die Augenbewegung ermittelt wird. Allerdings werden hierbei nur fünf Kalibrierungspunkte in die Berechnung miteinbezogen, was zu einem sehr ungenauen Ergebnis führt. Aus diesem Grund werden, wie aus der Literatur [[Heinsen und Vogt \(2003\)](#) S. 154-159] hervorgeht, neun Kalibrierungspunkte berücksichtigt, um die Position genauer zu ermitteln.

Zuerst wird der Bildschirm durch neun Kalibrierungspunkte, wie auf der Abbildung [[5.1 S. 68](#)] dargestellt ist, in neun Felder geteilt. Somit ergeben sich spaltenweise auf der x-Achse drei Geraden, die jeweils drei Kalibrierungspunkte erfassen. Bezüglich dieser Punkte, die auf der Geraden liegen, werden für die x-Achse der minimale x_l , der mittlere x_m und der maximale x_r Wert ermittelt. Dasselbe wird zeilenweise für die y-Achse durchgeführt: Berechnung des minimalen x_o , des mittleren y_m und des maximalen Wertes y_u . Hier befinden sich ebenfalls

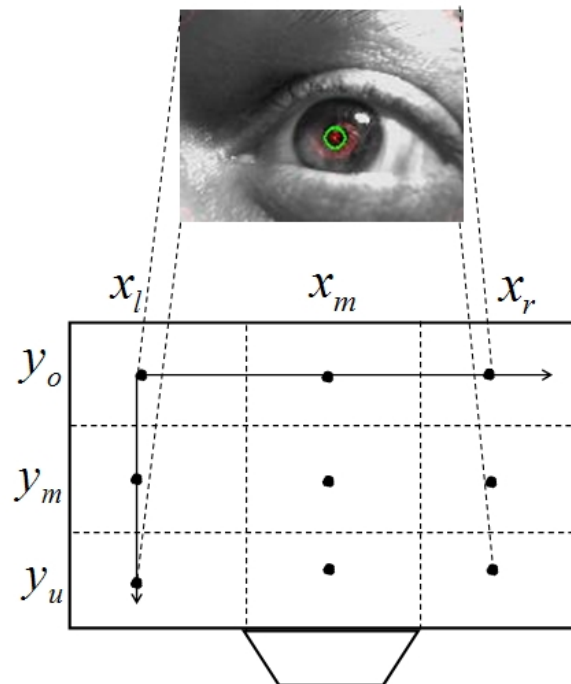


Abbildung 5.1: Detektierte Pupille des ersten Probanden

drei Geraden mit jeweils drei Kalibrierungspunkten. Die Mittelung wird durch eine If-Abfrage auf der x-Achse durchgeführt.

Zudem werden die Breite und die Höhe nach der folgenden Gleichung berechnet:

$$Breite = \frac{x_r - x_l * B_x}{B_x - Offset - x * 2} \quad (5.1)$$

Mit B_x ist die Bildschirmbreite gemeint. Offset ist für einen geschätzten Wert gedacht und in diesem Fall auf Null gesetzt.

$$Hhe = \frac{y_u - y_o * B_y}{B_y - Offset - x * 2} \quad (5.2)$$

Mit B_y ist die Bildschirmhöhe gemeint.

Nach dieser Berechnung werden einige Felder zu Sektoren zusammengefasst, sodass innerhalb des Sektors das Minimum und das Maximum auf der x- und y-Achse berechnet wird.

$$x_{l-min} = \frac{x_m - Breite}{2} \quad x_{l-max} = \frac{x_{l-min} + Breite}{3} \quad (5.3)$$

$$x_{m-min} = x_{l-max} \quad x_{m-max} = \frac{x_{m-min} + Breite}{3} \quad (5.4)$$

$$x_{r-min} = x_{m-max} \quad x_{r-max} = \frac{x_{r-min} + Breite}{3} \quad (5.5)$$

Dieselbe Berechnung, die für x durchgeführt wurde, wird auch für y durchgeführt.

Nachdem das Maximum und das Minimum auf der x- und y-Achse berechnet ist, kann die Position der detektierten Pupille in Verbindung mit den Kalibrierungsdaten wie folgt berechnet werden:

$$X_{blick} = B_x * \frac{x_{pupille} - x_{l-min}}{x_r - max - x_{min}} \quad (5.6)$$

Mit den Koordinaten $x_{pupille}$ und $y_{pupille}$ ist die Position der detektierten Pupille gemeint.

$$Y_{blick} = B_y * \frac{y_{pupille} - y_{l-min}}{y_r - max - y_{min}} \quad (5.7)$$

Anhand dieser berechneten Position, die andeutet, auf welchen Bildschirmpunkt der Anwender schaut, wird der Mauszeiger bewegt.

Laut des Testergebnisses wird der Mauszeiger zwar in die richtige Richtung hinbewegt, aber der Mauszeiger wird nicht am gewünschten Zielpunkt positioniert.

6 Realisierung und diskretes Verfahren

In diesem Kapitel wird zuerst auf die integrierte Entwicklungsumgebung (IDE engl. integrated development environment) eingegangen. Dort werden die integrierten Werkzeuge (engl. tools) vorgestellt, mit denen das Verfahren entwickelt wurde. Weiterhin wird auf das Werkzeug „Impresario“ und seine Funktionalität lediglich in Ansätzen eingegangen, da es sonst den Rahmen dieser Arbeit sprengen würde. Daran anknüpfend wird das diskrete Verfahren mit den jeweiligen Makros erläutert. Hierzu wird am Ende das Ergebnis der detektierten Pupille von unterschiedlichen Probanden vorgestellt.

6.1 Entwicklungsumgebung

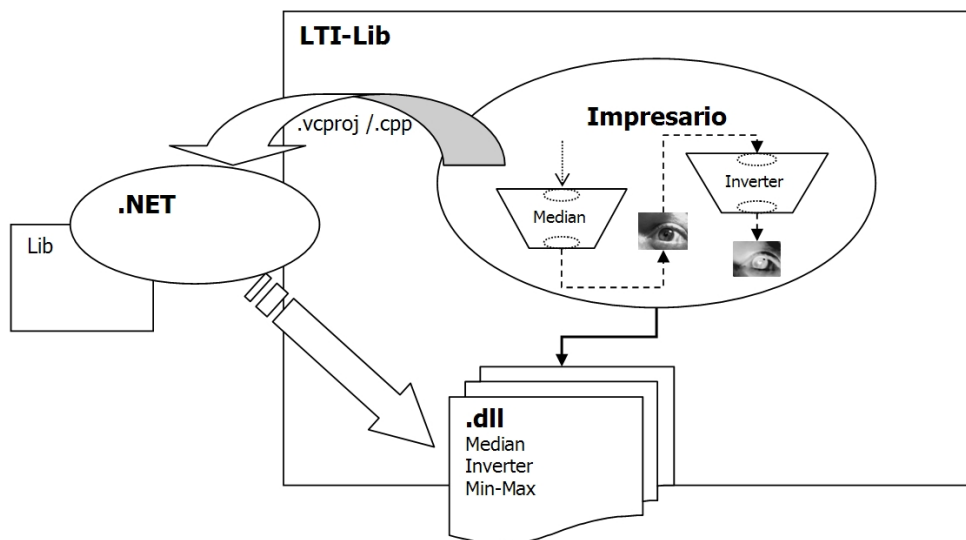


Abbildung 6.1: Integrierte Entwicklungsumgebung der Bildbearbeitung

Die integrierte Entwicklungsumgebung, wie aus der Abbildung [6.1 S. 70] ersichtlich ist, basiert auf der C++ Programmiersprache und ist auf .NET, Impresario und LTI-Lib zugeschnitten.

ten. LTI-Lib ist eine kompakte Bibliothek mit diversen Operatoren, die aus der Bildbearbeitung bekannt sind. .NET ist ein sehr umfangreiches Werkzeug und dient in diesem Zusammenhang der Entwicklung und Kompilierung einer DLL-Datei. Impresario wird als visuelles Werkzeug eingesetzt, mit dem die Funktionalität der jeweiligen Makros einer DLL-Datei angewendet werden können.

Wie aus der Abbildung [6.1 S. 70] hervorgeht, wird aus Impresario per Script-Sprache ein Projekt (.vcproj) generiert. In das Projekt wird die LTI-Lib-Bibliothek automatisch eingebunden. Das Projekt wird so eingestellt, dass es eine DLL-Datei erzeugt. Hierzu werden einige Template-Klassen, sowie eine Klasse nach dem Projektnamen benannt, generiert. Diese Klasse definiert den Einstiegspunkt für die DLL-Anwendung. Die DLL-Datei beinhaltet die Makros und eine Vektorliste, die die Objekte der Makros verwaltet. Eine Klasse (.cpp/.h) eines Makros kann ebenfalls mit Impresario generiert werden, und sie sieht vom Quellcode wie folgt aus, siehe Klasse Inverter [6.1 S. 71].

Inverter

```

1  #include ".\Inverter.h"
2  CInverter::CInverter(void){
3      SetMacroName("Inverter");
4      SetMacroCreator("My_Name");
5      SetMacroGroup("My_Group");
6      SetMacroDescription("My_Description");
7      AddMacroInput("...", "&m_pchn8Input");
8      AddMacroOutput("...", "&m_chnEdges");
9      AddMacroParameter(tList, ...);
10 }
11 CInverter::~CInverter(void){}
12 bool CInverter::InitMacro(){return true;}
13 bool CInverter::Apply(){return true;}
14 bool CInverter::ExitMacro(){return true;}
15 void CInverter::ParameterChanged(int nParameter){
16     switch(nParameter){
17     case 0:
18         break;}
19 }
```

Diese Klasse kann in ein Projekt aufgenommen werden. In der Klasse, die den Einstiegspunkt für die DLL-Anwendung definiert, wird dieser generierten Klasse ein Objekt in die Vektorliste eingetragen, sodass Impresario das Makro zur Verfügung stellt.

Wie aus der Inverter-Klasse zu ersehen ist, entspricht die Struktur einer gewöhnlichen C++-Klasse. Die Variablen einer Klasse werden wie bekannt in der Header-Datei deklariert. In

dem Konstruktor können sie initialisiert werden. Außerdem werden in dem Konstruktor Eigenschaften eines Makros wie Name, Autor, Gruppe (siehe Zeile 3-6) festgelegt. Damit das Makro mit einem anderen Makro durch Verbindungslinien zusammenarbeitet, werden die definierten Variablen für die Eingangsschnittstelle (siehe Zeile 7) und Ausgangsschnittstelle (siehe Zeile 8) festgelegt. In der Zeile 9 wird ein Einstellungsparameter für die Initialisierung einer Variablen definiert und eingefügt. Außer dem gewöhnlichen Konstruktor und Destruktor gibt es weitere wesentliche Methoden, wie InitMacro, Apply, ExitMacro und Parameter-Changed. Diese Methoden sorgen für die Kompatibilität als auch für die Funktionalität eines Makros. In der Apply-Methode werden die Funktionalität bzw. der Algorithmus eines Makros definiert. Diese Methode wird für die Bearbeitung eines Bildes ständig aufgerufen. Mit der InitMacro-Methode können z.B. Variablen initialisiert werden. Diese Methode wird einmal vor der Apply-Methode aufgerufen. Die ExitMacro-Methode wird einmal nach der Apply-Methode aufgerufen. Die Parameter-Changed-Methode sorgt dafür, dass die Variable mit den eingestellten Parameterwerten initialisiert wird. Dies kann auch während der Laufzeit geschehen. Diese Methode wird ständig vor der Apply-Methode aufgerufen.

Nach erfolgreicher Kompilation wird die vom .NET erzeugte DLL-Datei dem Impresario zur Verfügung gestellt. Impresario verwendet das Add-in-Prinzip, indem beim Start des Impresarios die DLL-Datei gelesen wird und die jeweiligen Makros zur Verfügung stellt. Ein Makro kann durch die Eingangsschnittstelle, Ausgangsschnittstelle und Einstellungsparameter gesteuert werden. Durch die visuelle Bedienbarkeit können die Komponenten verschiedener Makros per Drag and Drop angewendet werden. Durch eine Verbindungslinie können die ausgewählten Komponenten anhand der Eingangs- und Ausgangsschnittstelle in Verbindung gesetzt werden. Das Ergebnis der Eingangsschnittstelle als auch der Ausgangsschnittstelle kann, je nach Deklaration der Variablen, angeschaut werden.

6.2 Realisierung

Wie aus der Abbildung [6.2 S. 77] hervorgeht, besteht das integrierte Verfahren im Impresario aus einer Sequenz von Makros. Die Makros sind mittels ihrer Eingangs- und Ausgangsschnittstellen miteinander verbunden. Durch die Eingangs- und Ausgangsschnittstellen werden die erforderlichen Daten transferiert, die wiederum anhand des Algorithmus eines Makros bearbeitet werden. Das Ergebnis wird mittels seiner vorgesehenen Ausgangsschnittstelle zur Verfügung gestellt.

Das Verfahren, wie aus der Abbildung hervorgeht, beinhaltet folgende Makros, auf die im weiteren Verlauf eingegangen wird:

Video Capture Device: Dieses Makro ist für die Aufnahme der Augenbewegungen zuständig. Das Makro an sich beinhaltet einen Frame Grabber, mit dem die Bilder der Augenbewe-

gung für ein folgendes Makro zur Verfügung gestellt werden. Mit den Einstellungsparametern dieses Makros kann der Treiber der jeweiligen Webkamera ausgewählt werden. Weiterhin kann je nach Kompatibilität durch das Makro das Bedienungsprogramm des jeweiligen Treibers aufgerufen werden. Dies ist eine hilfreiche Eigenschaft, da die direkt einprogrammierte Funktionalität, z. B. das Zoomen des Auges, angewendet werden kann. Die Auflösung eines Bildes kann direkt anhand des Makros eingestellt werden. Durch seine Ausgangsschnittstelle stellt das Makro das aufgenommene Bild dem folgenden Makro zur Verfügung.

SplitImage: Hiermit kann die Zersplittung eines Bildes in den RGB-Kanälen vorgenommen werden. Dies ist notwendig, damit ein Intensitätsbild, nämlich "Kanal8", erzeugt wird.

Flip vertical: Dieses Makro ist für die vertikale Spiegelung des Bildes gedacht, und zwar für Testzwecke, damit getestet werden kann, wie sich das Verfahren zur Laufzeit verhält, wenn das Bild gespiegelt wird. Durch die Einstellungsparameter kann das Bild zur Laufzeit gespiegelt werden. Der Sourcecode für dieses Makro befindet sich auf der beiliegenden CD.

Downsampling: Mit Hilfe dieses Makros ist es möglich, die Auflösung des Bildes entlang der Spalten und entlang der Zeilen zu reduzieren. Außerdem bietet dieses Makro die Möglichkeit, die Bildpunkte zu interpolieren.

Median-Filter: Dieser Filter ist im Abschnitt [4.2.1 S. 44] ausführlich beschrieben und dient der Glättung eines Bildes. Durch den Einstellungsparameter kann die Faltungsmaske bestimmt werden.

Inverter: Der Inverter dient der Invertierung der Grauwerte eines Bildes. Durch den Einstellungsparameter kann der maximale Grauwert, je Kanaltyp z.B. 8 Bit, eingestellt werden. Der Sourcecode für dieses Makro befindet sich auf der beiliegenden CD.

Maximum-Filter: Der Filter dient der Bildung des Maximums eines Grauwertes bezüglich seines Ortsbereiches. Dieser Filter ist ebenfalls im Abschnitt [4.2.1 S. 44] ausführlich besprochen worden. Mittels seiner Einstellungsparameter kann die Faltungsmaske bestimmt werden. Der Sourcecode für dieses Makro befindet sich auf der beiliegenden CD.

Canny-Operator: Dieser Operator ist zuständig für die Erzeugung eines Kantenbildes. Durch seine Einstellungsparameter können die Faltungsmaske, die Variance, Min-Schwellenwert sowie Max-Schwellenwert eingestellt werden. Die ausführliche Beschreibung findet sich im Abschnitt ().

CPoinoperation: Zwei Bilder können anhand der Grauwerte aufaddiert oder subtrahiert werden. Außerdem beinhaltet dieses Makro die logischen Operatoren. Dieses dient der logischen UND-Verknüpfung der Grauwerte zweier Bilder.

Find Eye-Circle: Dieses Makro repräsentiert die HT für Kreise. Mit diesem Makro wird die Pupille detektiert und auf dem Bild durch einen Kreis mit dem jeweiligen Mittelpunkt gekennzeichnet. Durch den Einstellungsparameter wird der Schwellenwert für die Bildpunkte, die im Parameterraum berücksichtigt werden, eingestellt. Ferner kann der Radius eingegeben werden. Damit die Kreise vorher generiert werden, wird der Bresenham-Algorithmus verwendet. Hierfür ist ein Einstellungsparameter vorgesehen, und dieser kann während der Laufzeit, z. B. auf Chain-Code, umgestellt werden. Dies ist für die Erweiterung des Makros vorgesehen. Der Sourcecode für dieses Makro befindet sich auf der beiliegenden CD.

Calibration and Cursor: Dieses Makro ist für die Ermittlung der Position einer Augenbewegung gedacht. Bei diesem Makro können die Kalibrierungspunkte eines Bildschirms eingestellt werden. Des Weiteren kann der Mauszeiger eingeschaltet werden, sodass der Mauszeiger zur ermittelten Position hinbewegt wird. Der Sourcecode für dieses Makro befindet sich auf der beiliegenden CD.

6.3 Ergebnisse des Detektierens der Pupille

In diesem Abschnitt werden die Ergebnisse der detektierten Pupille vorgestellt. Hierzu werden die wesentlichen Punkte, unter anderem die Stabilität dieses Verfahrens, erläutert. Die tabellarische Darstellung wird für den Vergleich der Rechenzeit und der Einstellungsparameter der jeweiligen Makros angestellt. Die Überschriften der Makros sind wie folgt spaltenweise abgekürzt: *VCD* = *Video Capture Device*, *SP* = *Split Image*, *D* = *Downsampling*, *FeC* = *Find eye-circle*, *CE* = *Canny Edges*, *I* = *Inverter*, *CP* = *CPointoperation* Das Detektieren

Das Detektieren der Pupille bei den Probanden ist auf dem folgenden Rechner ausgeführt worden, siehe Tabelle [6.1 S. 74].

CPU	Intel Pentium 4 mit 3.40 GHz
Grafikkarte	NVIDIA GeForce 6610XL, mit 400 MHz, mit 128 MB
Support	DirectX 9c
Webkamera	no-name
Bild Sensor	CMOS, color 24-Bit YUY2
Anschluss	USB 2.0
Framerate	30 F/s bei 320x240
Max.-Auflösung	1280x1024
Preis	6 Euro

Tabelle 6.1: Rechner und Webkamera

1. PROBAND	Bild	VCD	SI	D	MF	CE	I	MF	CP	FeC
Auflösung	106x80									
Faltungsmaske					3x3	9x9	5x5			
Variance						9,9				
Schwellenwert-min.						0,2				
Schwellenwert-max.						0,05				236
Radius										4
Gewicht-A									1,0	
Gewicht-B									1,0	
Beleuchtet	ja							1,0		
Iris-Farbe	Braun							1,0		
Rechenzeit [ms]		356	001	001	001	005	001	006	007	025

Tabelle 6.2: Tabellarische Darstellung der Werte für das Resultat des ersten Probanden, siehe Abbildung [6.3 S. 77]

Anhand der Kantenbilder des UND-Operators auf der Abbildung [6.3 S. 77] ergibt sich die Schlussfolgerung, dass die HT für Kreise ein zuverlässiges Resultat liefert.

Bei dem dritten Probanden handelt sich um einen Jugendlichen, bei dem unter extrem schlechten Bedingungen die Augenbewegungen registriert worden sind. Dieser Fall ist absichtlich so gewählt, um das Verhalten des Verfahrens in Bezug auf die Zuverlässigkeit und Stabilität zu testen. Hier war zu erwarten, dass das Verfahren die Pupille nicht detektieren kann. Wie schon aus der Abbildung [6.5 S. 78] ersichtlich ist, verfügt der Proband über extrem dunkle Augen, sodass die Pupille kaum von der Iris unterschieden werden kann. Außerdem ist bei diesem Probanden das Augenlid nicht soweit aufgeklappt, sodass die Pupille von der Webkamera nicht immer erfasst werden kann. Die Abbildung [6.5 S. 78] stellt das erwartete Ergebnis dar, dass die Pupille nicht detektiert werden konnte. In einem solchen Fall kann, wie in der Abbildung [6.5 S. 78] in der zweiten Reihe zu sehen ist, statt der Pupille die Iris detektiert werden.

2. PROBAND	Bild	VCD	SI	D	MF	CE	I	MF	CP	FeC
Auflösung	106x80									
Faltungsmaske					3x3	9x9	5x5			
Variance						9,9				
Schwellenwert-min.						0,5				
Schwellenwert-max.						0,03				209
Radius										4
Gewicht-A									1,0	
Gewicht-B									1,0	
Beleuchtet	Nein									
Iris-Farbe	Blau									
Rechenzeit [ms]		336	001	001	001	002	001	004	007	038

Tabelle 6.3: Tabellarische Darstellung der Werte für das Resultat des zweiten Probanden, siehe Abbildung [6.4 S. 78]

3. PROBAND	Bild	VCD	SI	D	MF	CE	I	MF	CP	FeC
Auflösung	106x80									
Faltungsmaske					3x3	9x9	5x5			
Variance						9,9				
Schwellenwert-min.						0,5				
Schwellenwert-max.						0,02				250
Radius										4
Gewicht-A									1,0	
Gewicht-B									1,0	
Beleuchtet	Nein									
Iris-Farbe	Schwarz									
Rechenzeit [ms]		340	001	001	001	002	001	004	005	035

Tabelle 6.4: Tabellarische Darstellung der Werte für das Resultat des dritten Probanden, zweite Reihe, siehe Abbildung [6.5 S. 78]

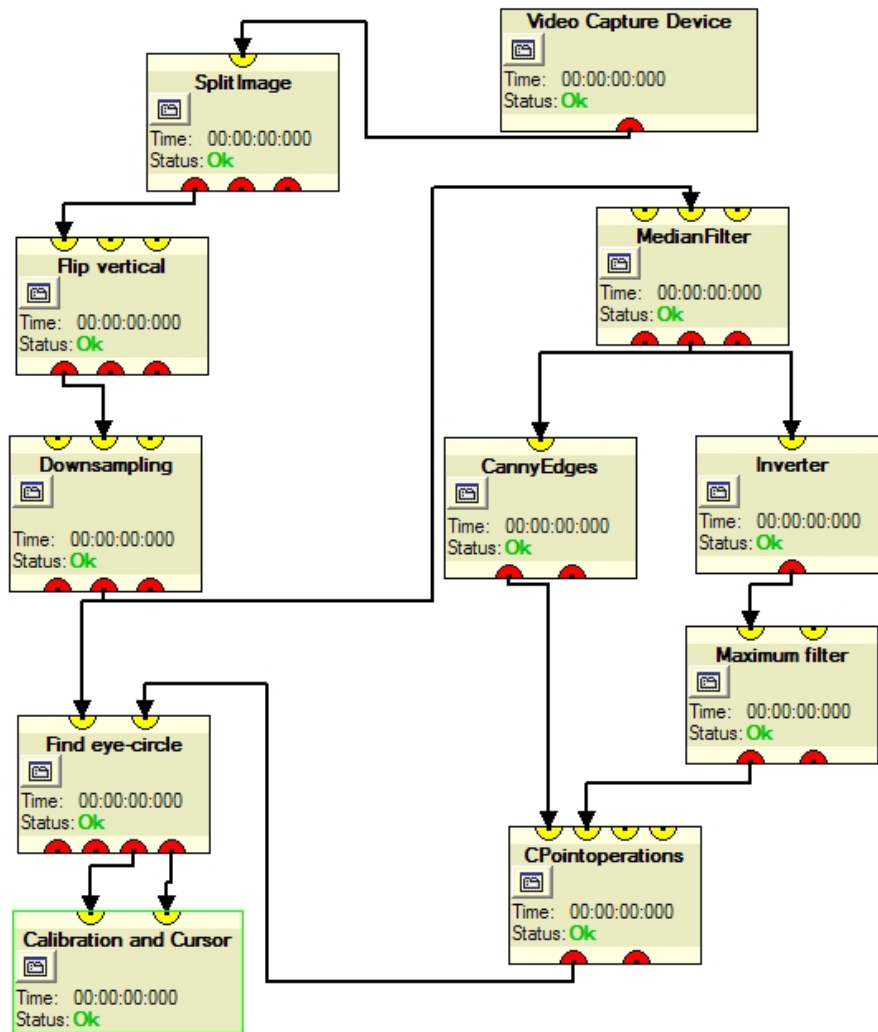


Abbildung 6.2: Verfahren zum Detektieren der Pupille

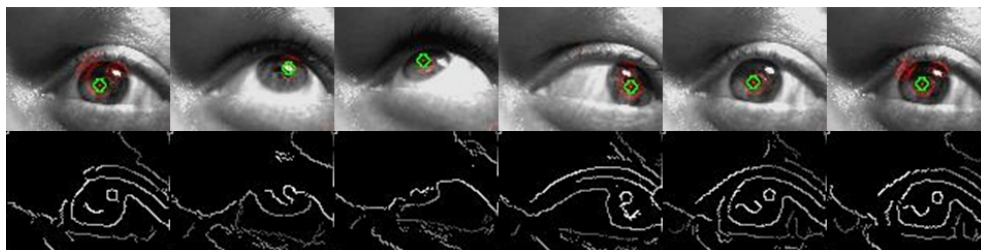


Abbildung 6.3: Detektierte Pupille des ersten Probanden

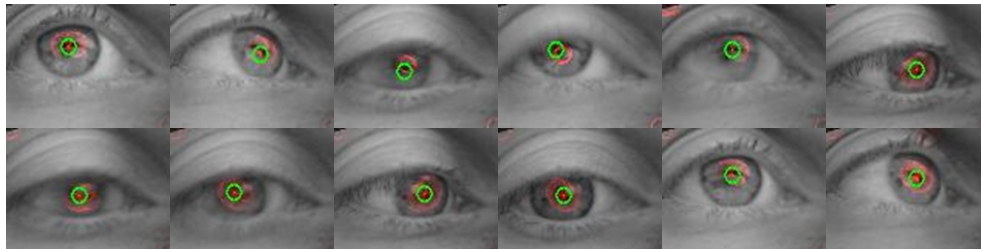


Abbildung 6.4: Detektierte Pupille des zweiten Probanden

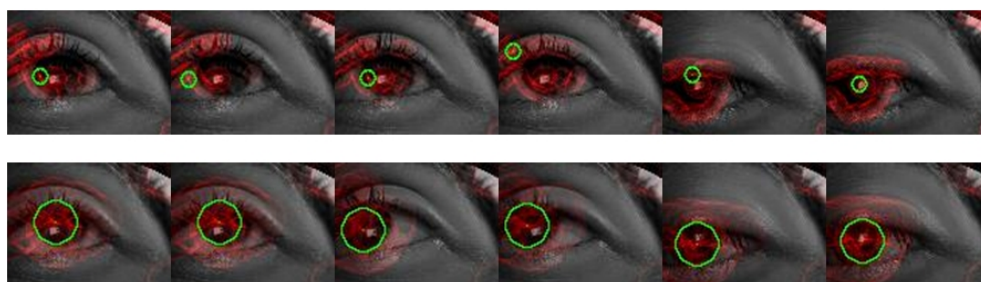


Abbildung 6.5: Detektierte Pupille des dritten Probanden

7 Resümee und Ausblick

An dieser Stelle wird auf die erreichten Ziele eingegangen, und dabei werden die Vorteile als auch die Nachteile erläutert. In Bezug auf die Schwierigkeiten wird eine mögliche theoretische Lösung vorgeschlagen.

7.1 Zusammenfassung

Wie aus der Anforderung hervorgeht, ist das Ziel anhand des hergestellten Systems erreicht worden. Das System, wie gefordert war, setzt eine billige Webkamera ein, mit der die Augenbewegungen eines Anwenders, mit z. B. cerebralen Bewegungsstörungen, registriert werden können. Hierbei wurde eine Halterung (Brillengestell mit Webkamera) bezüglich solcher Anwender prototypisch entwickelt. Aus den registrierten Bilddaten wurde mit dem Verfahren die Pupille detektiert, indem sie durch eine Position (x-Koordinate, y-Koordinate) ermittelt und visuell auf dem Bild durch einen Kreis dargestellt wurde. Der Mittelpunkt dieses Kreises entspricht dem Mittelpunkt der Pupille und stellt den Referenzpunkt für die Kalibrierung dar.

Die Kalibrierung, die prinzipiell keine zentrale Rolle gespielt hat, wurde ebenfalls realisiert. Hierbei wurden insgesamt neun Kalibrierungspunkte berücksichtigt, die in Verbindung mit der Position der Pupille gebracht wurden. Somit wurden die Augenbewegungen des Anwenders ermittelt. Die Kalibrierung zeigte insgesamt kein zufriedenstellendes Ergebnis, da der Mauszeiger nicht genau an die ermittelte Position gesteuert wurde. Dennoch wurde ein Anhaltspunkt gegeben und gezeigt, dass die Kalibrierung innerhalb der Entwicklungsumgebung integriert werden kann.

Wie aus den Testergebnissen hervorgeht, zeigt das Verfahren ein gutes Resultat. Trotz vieler Störfaktoren und unzureichender Beleuchtung wurde die Pupille zuverlässig detektiert. Bei dem dritten Probanden, siehe Abbildung [6.5 S. 78] war ein schlechtes Resultat aufgrund der dunklen Augen zu erwarten, dass nämlich die Pupille nicht detektiert werden würde. In diesem Fall, wie gezeigt wurde, konnte anstatt der Pupille die Iris detektiert werden, dessen Mittelpunkt den Referenzpunkt für die Kalibrierung darstellt.

Hinsichtlich der Rechenzeit wies das Verfahren eine akzeptable Rechenzeit auf, da ca. 2 Bilder pro Sekunde verarbeitet werden konnten.

7.2 Ausblick

Trotz einiger guten Ergebnisse entspricht das System insgesamt nicht dem Optimum, als dass es dem Anwender zur Verfügung gestellt werden könnte.

Während des Verfahrens wurde deutlich, dass für das Detektieren der Pupille, je nach Beleuchtung und Intensitätswertänderung zwischen der Iris und der Pupille, Einstellungen der jeweiligen Parameter der Makros notwendig sind. Damit diese Einstellungen automatisch laufen können, könnte hierzu ein Histogramm benutzt werden. Mittels des Histogramms sind die Informationen über die Dynamik und den Kontrast eines Bildes bekannt. Nach dieser Information kann z. B. ein Intervall von Werten gebildet werden, das der Einstellung des Parameters dienen soll. Anhand des eingestellten Parameters kann, mittels der gesuchten Form, geprüft werden, ob sich dieser Wert auf dem Kreis befindet. Hierzu könnte durch Abfragen, ob die Pupille bzw. die Iris vollständig gekennzeichnet ist, sichergestellt werden, dass die Pupille bzw. die Iris richtig detektiert wird.

Bezogen auf die Lichtverhältnisse, die sich ständig ändern, ist es möglich nach vorgesehener Berechnung [Jähne (2005) S. 129] diese zu kontrollieren. Aus dem Ergebnis kann je nach Anforderung die Helligkeit korrigiert werden oder das Infrarot-LED eingeschaltet werden. Viele Webkameras bieten heutzutage Infrarotlicht, das sich bei einer schlechten Beleuchtung einschaltet.

Wie schon erwähnt, wurde der Mauszeiger nicht exakt an der gewünschten Stelle positioniert. Diese Messung hängt stark von den Kalibrierungsdaten ab. Die Kalibrierungspunkte wurden per Default-Werte mit der Breite und Höhe des Bildes initialisiert. Aus der Literatur [Heinsen und Vogt (2003) S. 154-159] geht hervor, dass der Anwender instruiert wird, eine Kalibrierung durch das Hin- und Herbewegen seiner Augen auf dem Bildschirm, vorzunehmen. Das bedeutet, dass hierzu weitere Kalibrierungspunkte einbezogen werden können. Die Berechnung kann mit mehr als neun Punkten durchgeführt werden, um eine genauere Position der Augenbewegung zu ermitteln.

Eine Zukunftsvision, die für das kopfbasierte System spricht, ist, dass die Bildübertragung zwischen der Webkamera und dem Rechner durch Bluetooth oder WLAN stattfinden könnte. Hier muss je nach passender Übertragungsrates und Übertragungskapazität über das passende Hardware-Teil nachgedacht werden. Mit dem Komfort der kabellosen Vernetzung ist es möglich, weitere Funktionen einzubauen. So könnte der Rechner als multimediale Fernbedienung gestaltet werden, die für einen bewegungseingeschränkten Anwender von großer Hilfe sein könnte.

Die Geschwindigkeit des Verfahrens kann an vielen Stellen verbessert werden. Für die Erzeugung des Kreises kann anstatt des Bresenham-Algorithmus der Chain-Code benutzt werden. Der erzeugte Code wird innerhalb einer Liste verwaltet, sodass anhand dieses Codes

der gesuchte Kreis im Parameterraum gezeichnet wird. Später wird dieser Kreis innerhalb der HT detektiert. Weiterhin kann die Geschwindigkeit des Verfahrens - wie aus der Literatur [[Steinbecher \(2003\)](#) S. 129] hervorgeht - durch den Bubblesort innerhalb des Median-Filters verbessert werden: „Einer der größten Nachteile des Medianfilters ist die laufend notwendige Sortierung der Eingangswerte. Diese Sortierung kann mit bekannten Verfahren wie Quicksort, Bubblesort, mit verketteten Listen usw. durchgeführt werden.“

Abschließend ist festzustellen, dass die Messung der Augenbewegung durch das Detektieren der Pupille mit zunehmendem technologischen Fortschritt immer wichtiger wird. In vielen Industriebereichen - wie in der Automobilbranche, im militärischen Bereich, in dem Mensch-Computer-Interaktion-Bereich - wird geforscht, und zum Teil wird das Verfolgen der Augenbewegungen durch das Detektieren der Pupille eingesetzt. Dies spricht dafür, dass das Forschen und die Erweiterung dieses Themas viele Menschen in unterschiedlichen Bereichen unterstützen kann.

Literaturverzeichnis

- [Bungartz u. a. 2002] BUNGARTZ, Hans-Joachim ; GRIEBEL, Michael ; ZENGER, Christoph: *Einführung in die Computergraphik*. 2. Auflage. Vieweg, 2002. – 30–37 S. – ISBN 3528167696
- [Burger und Burge 2005] BURGER, Von W. ; BURGE, Mark J.: *Digitale Bildverarbeitung*. 8. Auflage. Springer, 2005. – 164–166 S. – ISBN 3540214658
- [Canny 1986] CANNY, J.: *A computational approach to edge detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society, 1986
- [Dresden 2008] DRESDEN, Technische U.: *Bildquelle*:. 2008. – URL http://rcswww.urz.tu-dresden.de/~cogsci/welcome_g.html?/~cogsci/augenbewegungen.html
- [Duchowski 2003] DUCHOWSKI, Andrew T.: *Eye Tracking Methodology: Theory and Practice*. 1. Auflage. Springer, 2003. – ISBN 1852336668
- [Fakultät 2008] FAKULTÄT, Saarlandes-Philosophische: *Was ist Eyetracking?* 2008. – URL <http://usability.is.uni-sb.de/methoden/eyetracking.php>
- [GOLEM 2006] GOLEM: *Programmieren ohne Tastatur. VoiceCode-Spracherkennung für Python-Entwickler*. 2006. – URL <http://www.golem.de/0604/45005.html>
- [Gonzalez und Woods 2003] GONZALEZ, Rafael C. ; WOODS, Richard E.: *Digital Image Processing*. 3. Auflage. Upper Saddle River, New Jersey : Pearson Prentice Hall, 2003. – K. 2, K. 5, K. 10, K. 11, 35–103 S. – ISBN 013168728X
- [Heinsen und Vogt 2003] HEINSEN, Sven ; VOGT, Petra: *Usability praktisch umsetzen*. 1. Auflage. Carl Hanser, 2003. – 154–159 S. – ISBN 3-446-22272-3
- [Instruments 2008] INSTRUMENTS, Origin: *Origin Instruments*. 2008. – URL <http://www.orin.com/3dtrack/>
- [ISAAC 2008] ISAAC: *Bildquelle*:. 2008. – URL <http://www.isaac-online.org/en/home.shtml>
- [Jähne 2005] JÄHNE, Bernd: *Digitale Bildverarbeitung*. 5. Auflage. Springer, 2005. – 17–38, 243–255 S. – ISBN 3540249990

- [KI->I 2008] KI->I: *KI->I (Kompetenznetzwerk Informationstechnologie zur Förderung der Integration von Menschen mit Behinderungen)*. 2008. – URL <http://www.ki-i.at/>
- [Mehrle 2003] MEHRLE, Von G.: *Augenheilkunde für Pflege- und Gesundheitsfachberufe*. 6. Auflage. Elsevier, 2003. – 1–6 S. – ISBN 343727600X
- [Meisel 2008] MEISEL, Prof. Dr.-Ing. A.: *Robot Vision*. 2008
- [Metrovision 2008] METROVISION: *Instruments for measuring eye movements*. 2008. – URL <http://www.metrovision.fr/mv-po-notice-us.html#POG>
- [Ohno und Mukawa 2004] OHNO, Takehiko ; MUKAWA, Naoki: *A Free-head, Simple Calibration, Gaze Tracking System That Enables Gaze-Based Interaction*. Tokyo pp.115-122, 2004. – 1–8 S
- [Onmeda 2008] ONMEDA: *Bildquelle:*. 2008. – URL http://www.onmeda.de/lexika/anatomie/auge_anatomie.html?p=6
- [Papula 1998] PAPULA, Lothar: *Mathematik für Ingenieure und Naturwissenschaftler Band 1*. 8. Auflage. Friedr. Vieweg und Sohn GmbH, Braunschweig/Wiesbaden : Vieweg, 1998. – 217–219 S. – ISBN 3-528-74236-4
- [Rickheit u. a. 2003] RICKHEIT, Gert ; HERRMANN, Theo ; DEUTSCH, Werner: *Psycholinguistik*. 1. Auflage. Berlin : Walter de Gruyter, 2003. – 143–151 S. – ISBN 3-11-011424-0
- [Romich 2008] ROMICH, Prentke: *PowerTalker*. 2008. – URL <http://www.prentke-romich.de/111-0-powertalker.html>
- [Schmidt u. a. 2003] SCHMIDT, Dr. Robert F. ; THEWS, Dr. G. ; LANG, Dr. F.: *Physiologie des Menschen mit Pathophysiologie*. 29. Auflage. Berlin : Springer, 2003. – 143–145 S. – ISBN 3540218823
- [Steinbecher 2003] STEINBECHER, Rainer: *Bildverarbeitung in der Praxis*. 1. Auflage. Oldenbourg, 2003. – 30–36 S. – ISBN 3-486-22372-0
- [med. Thomas Wallny 31.10.2007] THOMAS WALLNY, Dr. med.: *RSI-Syndrom, Repetitive Strain Injury Syndrom*. 31.10.2007. – URL http://www.netdokter.de/krankheiten/fakta/rsi_syndrom.htm
- [Tobii 2008] TOBII: *Tobii*. 2008. – URL http://www.tobii.com/archive/pages/17749/mytobii_p10.aspx
- [Tönnies 2005] TÖNNIES, Klaus D.: *Grundlagen der Bildverarbeitung*. 1. Auflage. Kösel Krugzell, 2005. – 117, 258–259 S. – ISBN 3-8273-7155-4

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw. §24(4) bzw. §25(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 21. August 2008

Ort, Datum

Unterschrift