

Dissertation

**Bildverarbeitung eines
ophthalmologischen Messsystems
zur Charakterisierung des
Tränenfilms**

zur Erlangung des akademischen Grades

Dr. rer. nat.

am Fachbereich Mathematik

der Justus-Liebig-Universität Gießen

vorgelegt von

Dipl. Math. Thomas Alexander Pitz

geb. in Göttingen

Datum der Disputation: 04. Oktober 2012

Erstgutachter:	Prof. Dr. Tomas Sauer
Zweitgutachterin:	Prof. Dr. Brigitte Forster-Heinlein
Drittgutachter:	Prof. Dr. Martin Buhmann

Diese Arbeit entstand im Rahmen einer Kooperation der numerischen Mathematik der Universität Gießen, dem Institut für medizinische Optik der Universität Erlangen, der Tomey GmbH in Erlangen, der Weiss Medizintechnik GmbH in Erlangen und der Augenklinik des Universitätsklinikums Erlangen. Daher möchte ich zunächst allen Projektbeteiligten für die gute Zusammenarbeit danken.

Mein besonderer Dank gilt Herrn Professor Dr. Sauer für seine sehr gute Betreuung und den Mitarbeitern und Mitarbeiterinnen des Lehrstuhls für numerische Mathematik an der Justus-Liebig Universität Gießen für die schöne Zeit.

Abschließend möchte ich noch allen ganz herzlich danken, die mich bei dieser Arbeit unterstützt haben.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgabenstellung	1
1.2	Hardware	2
1.3	Software	3
1.4	Klinische Tests	3
2	Grundlagen	5
2.1	Medizinische Grundlagen	5
2.1.1	Das Auge	5
2.1.2	Korneale Topographie	10
2.1.3	Analyseverfahren	15
2.2	Technische Grundlagen	19
2.2.1	Digitale Signalverarbeitung	19
2.2.2	Faltungen und Fouriertransformationen	21
2.2.3	Bildverarbeitung	27
2.2.4	Kantendetektion	31
2.2.5	Image Registration	37
2.2.6	Splines	40
3	Verarbeitung	44
3.1	Voranalyse	44
3.2	Image Registration	46
3.2.1	Kantenbild	49
3.2.2	Mittelpunktsuche	57
3.2.3	Polartransformation	66
3.2.4	Feature Detection	70
3.2.5	Feature Matching	73
3.2.6	Funktionsklassen	75
3.2.7	Modellberechnung	82
3.2.8	inverse Polartransformation	85
3.2.9	Direkte Registrierung	85

4	Ergebnisse	88
4.1	Ergebnisse Detektion deformierter Kreise	88
4.2	Ergebnisse Image Registration	91
4.3	Ergebnis Tränenfilmaufriss	93
5	Zusammenfassung	95
6	Ausblick	95
7	Anhang	96
7.1	Feature Detection Implementierungen	96
	Literaturverzeichnis	104
	Stichwortverzeichnis	109
	Notationen	109

1 Einleitung

Gegenstand dieser Arbeit ist eine automatisierte, nicht invasive Analyse des Abrissverhaltens des Tränenfilms und der vorderen Hornhautoberfläche, um damit ein standardisiertes Verfahren zur Diagnostik des *Dry-Eye-Syndroms* zu generieren. Das trockene Auge oder Dry-Eye-Syndrom ist eine weit verbreitete Funktionsstörung des Tränenfilmes. Symptome, wie Trockenheitsgefühl, brennende Augen (stechende Schmerzen), Kopfschmerzen oder Sehbeeinträchtigungen¹ wie Schleiersehen oder Visusminderung können bei diesem Krankheitsbild auftreten. Die Ursachen für dieses Krankheitsbild sind vielfältig, wie beispielsweise Luftverschmutzung, Klimaanlage, Kontaktlinsen oder auch übermäßige Bildschirmarbeit. Wie man an diesen Ursachen erkennen kann, wird sich diese Krankheit in den nächsten Jahren weiter verbreiten. Eine vollständige Heilung ist laut [10] nicht möglich, lediglich eine Linderung der Beschwerden. Da sich die Symptome im Krankheitsverlauf verschlimmern, ist eine rechtzeitige Diagnose für die Behandlung essenziell. Therapiemöglichkeiten sind, je nach Grundleiden des Patienten, die Verabreichung von Tränenersatzflüssigkeit, Seitenschutz an Brillengestellen und evtl. auch operative Maßnahmen.

Aktuelle Diagnosetechniken, wie beispielsweise der Schirmer Test oder das Messen der Tränenfilmaufrisszeit mit der Spaltlampe, sind meist invasiv und nicht standardisiert. Diese werden in 2.1.3 näher beschrieben.

1.1 Aufgabenstellung

In dieser Arbeit wird nun ein geeignetes, nicht-invasives, automatisiertes Diagnose-tool vorgestellt. Mittels einer CMOS-Kamera soll das dynamische Verhalten eines auf die Hornhautoberfläche projizierten Musters aufgenommen werden. Gleichzeitig soll eine auf den Lipidfilm fokussierte Kamera das dynamische Verhalten des Lipids aufzeichnen. Danach sollen die so erhaltenen Daten verarbeitet werden und auf Pathologien des Auges untersucht werden. Da die Cornea elektromagnetische Strahlung diffus streut, sollte das Muster auf den Bildern an den Stellen des Tränenfilmaufrisses nicht mehr reflektiert werden. Diese Art von Verfahren wird schon in anderen Bereichen der Ophthalmologie eingesetzt (siehe 2.1.2), insbesondere auch bei der Analyse des Abrissverhaltens des Tränenfilms, allerdings beruhen die Verarbeitungen dieser Daten auf zweidimensionaler bzw. Einzelbildanalyse (siehe 2.1.3). Mit unserem Ansatz möchten

¹In [17] wird belegt, dass die Sehschärfe von Patienten mit Dry-Eye-Syndrom beeinträchtigt ist.

wir nun die Genauigkeit erhöhen, indem wir eine dreidimensionale Analyse der Daten vornehmen.

Anforderungen:

- Die Datenverarbeitung soll auf einem handelsüblichen Windows-PC lauffähig sein. (Hardwareunabhängig)
- Der zeitliche Verarbeitungsaufwand soll möglichst gering sein. (Geschwindigkeit)
- Dreidimensionale Verarbeitung der Daten, um die Genauigkeit des Verfahrens zu erhöhen. (Qualität)

Offensichtlich ist es nicht möglich, alle diese Anforderungen gleichzeitig zu optimieren.² Daher ist das Ziel dieser Arbeit, einen Ansatz zu finden, welcher sich auf Veränderungen der Prioritäten der vier Anforderungen anpassen lässt.

1.2 Hardware

Der erste Versuchsaufbau wurde 2007 entwickelt. Dabei bestand der Prototyp aus einer CMOS Kamera und einer Vorrichtung, die mittels acht Leuchtdioden und stellenweise unterschiedlich eingefärbte Kunststoffscheiben, welche dann ein bestimmtes Muster projizierten. Für eine genauere Beschreibung siehe [7].

Es stellte sich heraus, dass eine CCD-Kamera für die Entwicklung besser ist und somit wurde die CMOS-Kamera ausgetauscht. Danach wurden die acht Leuchtdioden durch eine homogenere Lichtquelle ersetzt, um die Lichtabfälschungen auf den Daten zu entfernen.

Eine weitere Entwicklung war der Einbau einer zweiten Kamera, welche nicht auf das reflektierte Muster fokussiert war, sondern auf direkt auf die Hornhautoberfläche. Dadurch wurde das Muster zwar unscharf, allerdings war es möglich, den Verlauf des Lipidfilms zu messen, welcher vermutlich ebenfalls mit dem Abrissverhalten des Tränenfilms in Verbindung steht. Zusätzlich wurde noch ein Fixierlicht in den Projektaufbau eingebracht, damit das menschliche Auge einen Fixierpunkt hat und somit die Sakkaden minimiert werden. Eine genau Beschreibung über den jetzigen Prototypen findet sich in [18]. Im folgenden wird die erste Kamera, welche auf das Muster fokussiert,

²Siehe Projektdreieck, bzw. Projektviereck in Projektmanagementliteratur.

als *Musterkamera* und die Kamera, welche den Verlauf des Lipidfilms aufzeichnet, als *Lipidkamera* bezeichnen.

1.3 Software

Wie in [7] beschrieben, bietet sich hierfür ein Ansatz aus der Wavelettheorie an. Dieses Verfahren konnte auch erfolgreich an Modellaugen getestet werden und lieferte recht gute Ergebnisse. Allerdings wurde festgestellt, dass diese Art der Verarbeitung ein wesentliches Problem aufweist, welches sich am Modellauge nicht zeigt. Sakkaden des menschlichen Auges sind schnelle, ruckartige Bewegungen mit bis zu 15 Winkelminuten (siehe [8]) des Auges. Diese Augenbewegungen sind willkürliche Bewegungen und können vom Probanden nicht unterdrückt werden. Die Verarbeitungsform in [7] beruht allerdings darauf, dass die in einem Video zusammengefassten Bilder örtlich dieselben Objekte abbilden. Daher muss ein Vorverarbeitungsschritt eingefügt werden, welchen die Muster auf den Bildern aneinander anpasst (*Image Aligning*). Im folgenden Kapitel werden einige Ansätze beschrieben und die Probleme damit erläutert. Abschliessend sollen die ermittelten und verarbeiteten Daten noch (halb-) automatisch ausgewertet werden. Die theoretischen Grundlagen hierfür finden sich in [5]. Allerdings können die Auswertungen erst vorgenommen werden, wenn die Verarbeitungsalgorithmen feststehen und bekannt ist, welche Features aus den Daten ermittelt werden können.

Hierzu ist es auch nötig, für die zweite Kamera eine Verarbeitung zu finden. Um den Verlauf des Lipidfilms beschreiben zu können, ist allerdings auch eine Anpassung der Muster auf den Bildern erforderlich. Diese Anpassung wird von der ersten Kamera ermittelt und kann dann direkt für die Bilder der zweiten Kamera verwendet werden.

1.4 Klinische Tests

Die Hardware befindet sich im Moment (Stand August 2011) im Universitätsklinikum Erlangen. Dort ist im Fachzentrum für Augenheilkunde damit begonnen worden, ca. 200 Datensätze zu erheben. Dabei sollen 100 Probanden mit diagnostiziertem Dry-Eye-Syndrom und 100 gesunde Probanden an der Studie teilnehmen. Von allen Probanden werden diverse relevante Daten erhoben, wie z.B. die subjektive Messung der BUT³ mittels Spaltlampe, Alter, Geschlecht, etc. Nach der Erhebung der Daten werden diese statistisch ausgewertet und zusammen mit den mittels Software verarbeitete-

³Break-Up-Time, Tränenaufrisszeit

ten Featuredaten in einem lerntheoretischen Algorithmus verarbeitet. Dieser Ansatz ist allerdings nicht mehr Gegenstand dieser Arbeit. Die Ergebnisse werden in einer separaten wissenschaftlichen Arbeit veröffentlicht.

2 Grundlagen

In diesem Abschnitt werden grundlegende Aspekte erläutert, die zur Problemlösung beigetragen haben. Der erste Teil befasst sich mit dem menschlichen Auge, wobei zunächst eine kleine anatomische Übersicht über dessen Eigenschaften gegeben wird, gefolgt von mathematischen Modellen des Auges und die pathologische Beschreibung des Dry-Eye-Syndroms. Im zweiten Abschnitt wird eine kurze Einführung zu kornealen Topographieverfahren gegeben, da dieses Thema sehr nah mit der Messung des Tränenfilms zusammenhängt. Im dritten Abschnitt werden aktuell verwendete Analyseverfahren des Dry-Eye-Syndroms beschrieben und diskutiert. Die darauf folgenden Abschnitte befassen sich mit grundlegenden mathematischen/technischen Themen, die eine Basis für das Verständnis der in der Verarbeitung gewählten Algorithmik erzeugen sollen.

2.1 Medizinische Grundlagen

2.1.1 Das Auge

Anatomie und Funktionsweise

Das Auge ist ein anatomischer Apparat zur Messung elektromagnetischer Strahlung, die im Wellenlängenbereich des Emissionsspektrums der Sonne liegt. Beim Menschen beträgt dieser Wellenlängenbereich der Strahlung etwa zwischen 380 nm und 780 nm. Die generelle Struktur des Auges ist in Abbildung 1 abgebildet. Das Auge ist so konstruiert, dass es eintreffende Lichtstrahlen durch das optische System (Tränenfilm, Hornhaut, Linse, etc.) auf die Netzhaut fokussieren kann. Für diese Arbeit ist lediglich der Tränenfilm und der optische Apparat von Interesse, daher sei an dieser Stelle für eine genauere Beschreibung der Physiologie des Auges auf beispielsweise [21] und [22] verwiesen. Eine anatomische Beschreibung des menschlichen Auges findet sich in [23].

Tränenfilm

Die Intaktheit des Zusammenwirkens zwischen Tränenproduktion und -abtransport ist Voraussetzung für die Unversehrtheit des vorderen Augenabschnittes. Durch die Tränenflüssigkeit erfolgt die mechanische Reinigung von Hornhaut und Bindehaut, der Gehalt an bakteriziden Lysozymen und Immunglobulinen erhält die Keimfreiheit und durch die Benetzung der Hornhaut wird die Aufnahme von Sauerstoff aus der Luft

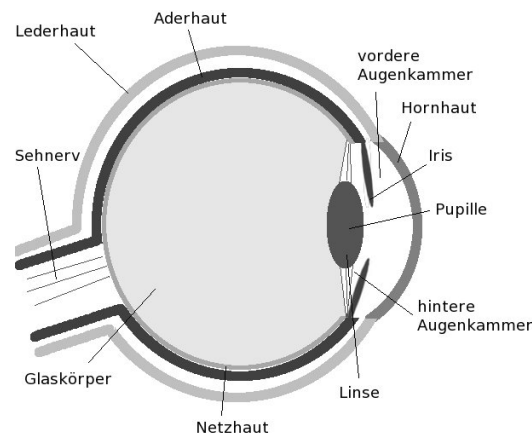


Abbildung 1: Seitenansicht des schematischen Aufbaus des menschlichen Auges

möglich. Der Tränenfilm besteht aus drei Schichten: der Muzinschicht, der wässrigen Schicht und der Lipidschicht. Die wässrige Schicht besteht zu 98% aus Wasser und wird gleichmäßig durch den Lidschlag über die Hornhaut verteilt. Die Muzinschicht ist die Verbindung zwischen wässriger Schicht und Hornhautoberfläche. Sie ist wichtig für die bessere Haftung der wässrigen Schicht an der Hornhautoberfläche. Die Lipidschicht liegt über der wässrigen Schicht und verhindert durch ihren hohen Fettgehalt das Austrocknen der wässrigen Schicht.

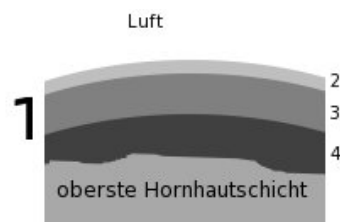


Abbildung 2: Der Tränenfilm: 1. gesamter Tränenfilm 2. Fettschicht (Lipidschicht) 3. wässrige Schicht 4. Schleimschicht (Muzinschicht)

Beim Zwinkern entstehen mikroskopische Bläschen. Nach ca. 0,5 bis 1,5 Sekunden (je nach Konsistenz des Films) lösen sie sich auf und durch den Lidschlag erzeugte Verdickungen und Unebenheiten auf der Kornea werden ausgeglichen. Dadurch entsteht ein gleichmäßig über die Kornea verteilter Film, der durch seine Konsistenz elektromagnetische Strahlung (Licht, Infrarot) reflektiert. Dieser bricht, je nach Qualität und

Quantität, nach etwa 5-15 Sekunden an verschiedenen Stellen wieder auf.

Die Kornea an sich streut elektromagnetische Strahlung diffus. Daher wird eine Projektion auf die Kornea an den Stellen, an denen der Tränenfilm aufgerissen ist, verzerrt. Mehr Informationen zu diesem und dem folgenden Abschnitt sind beispielsweise in [10] zu finden.

Die Kornea

Im Folgenden wird die korneale Struktur des normalen (standardisierten) menschlichen Auges beschrieben. Dazu ist in Abbildung 3 die Gliederung der vorderen Kornea in die vier Zonen (zentrale, parazentrale, peripherale und limbale Zone) abgebildet. Im Durchschnitt hat die Vorderfläche der Kornea einen Radius von 12 mm und eine Krümmung von 7,8 mm. In Tabelle 1 wird eine kleine tabellarische Übersicht über die normalen anatomischen und optischen Indizes des vorderen Segmentes der Kornea gegeben.

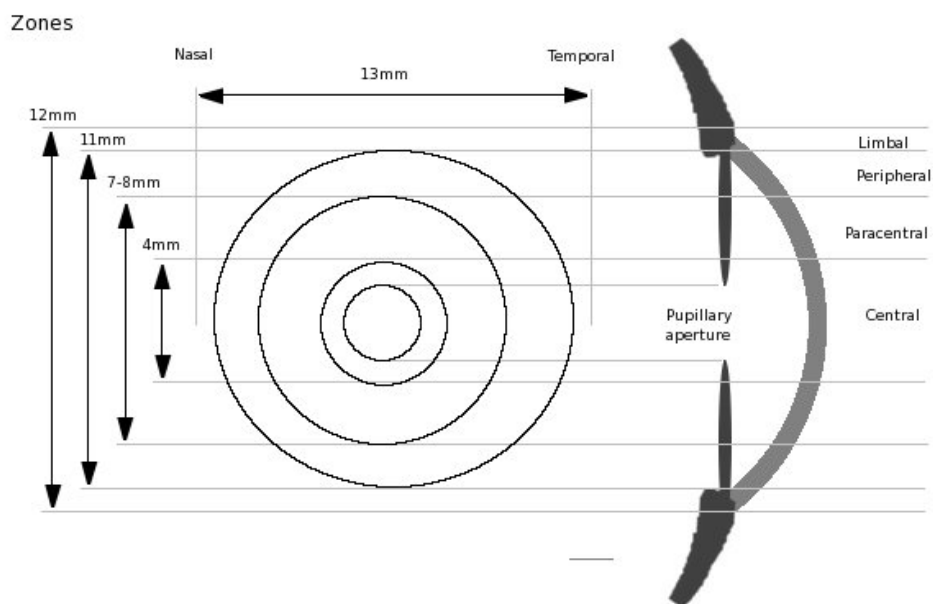


Abbildung 3: Zonen und Maße der Kornea. Schematische Darstellung der rechten Kornea in vertikaler (links) und frontaler Ansicht (rechts). Die Vorderfläche wird in vier Zonen aufgeteilt: zentrale, parazentrale, peripherale und limbale Zone[1]

Der zentrale Bereich der Kornea (4 mm, siehe Abb. 3) ist nahezu sphärisch. Außerhalb des Bereichs wird die Kornea asphärisch und radial asymmetrisch: Der Krümmungsradius verändert sich vom Zentrum zum Limbalbereich mit verschiedenen Stärken auf

Vordersegmentstruktur	Durchschnitt	Bereich
Refraktive Indizes		
Luft	1,0	
Kornea	1,376	
wässrige Phase	1,373	
Linse	1.336	
Zentralradius der Krümmung		
Vorderfläche der Kornea	7,8 mm	7,0 - 8,6 mm
hintere Fläche der Kornea	6,7 mm	
Dioptrische Stärke		
Vorderfläche der Kornea	49,50 D	39-48 D
hintere Fläche der Kornea	-6,00 D	
Nennleistung Kornea	43,50 D	
Nennleistung Linse	20,00 D	
Totalstärke des Auges	63,5 D	
Dicke		
zentrale Kornea	0,56 mm	50-60 μ m
periphere Kornea	1,20 mm	
korneales Epithel	0,06 mm	

Tabelle 1: Anatomische und optische Indizes des normalen vorderen Kornealsegmentes.[1]

unterschiedlichen Teilmeridianen. In [3] werden die verschiedenen mathematischen Modelle von Helmholtz-Laurance (1909), Gullstrand (1911), Emsley (1946), Schwiederling (1995) und Liou-Brennan (1997) aufgeführt und verglichen. Als einfachster mathematischer Ansatz für die Beschreibung der Kornea, wird beispielsweise in [2] die Formel $x^2 + y^2(1 + K) + z^2 - 2Rz = 0$ angegeben, wobei R als apikaler Radius der Oberfläche und k als konische Konstante definiert wird. In der Literatur ist die Formel $x^2 + y^2 + Qz^2 - 2Rz = 0$ mit Q als kornealer Asphärizität am Häufigsten vertreten. In [1] wird für die Asphärizität folgende Einteilung gemacht:

wodurch hier der Normalwert als $Q = -0,26$ angegeben wird. Dieser Wert wird in [30] ebenfalls als Standardwert beschrieben.

Augenbewegungen

Augenbewegungen werden in drei Klassen eingeteilt: Duktionen, Vergenzen und Versionen. Bei Duktionen handelt es sich um Drehbewegungen eines einzelnen Auges. Vergenzen sind die gegensinnige Bewegungen der Augen (Konvergenz nach innen, Divergenz nach außen). Im Folgenden wird die weitere Unterteilung von Versionen

Asphärizität	Form	Beschreibung	Beispiel
> 0	ansteigend	peripheraler Anstieg	radiale Keratotomie
0	sphärisch	einheitliche Krümmung	Kalibrationskugel
< 0	abgeflacht	peripherale Abflachung	
-0,26	abgeflacht	leichte peripherale Abflachung	Normale Kornea
< -0,26	abgeflacht	merkliche peripherale Abflachung	Keratokonus

Tabelle 2: korneale Asphärizität. Der Wert Q beschreibt den Anstieg/die Abflachung der Kornea vom Zentrum zum peripheralen Bereich.[1]

beschrieben, welche die gleichsinnige Bewegung beider Augen darstellen.

Die Augen können langsam (Folgebewegungen, langsame Phase des optokinetischen Nystagmus) oder schnell (Blicksprünge, Blickzielbewegung, Sakkaden) bewegt werden.

Langsame Augenbewegungen werden über eine retinale Bildverschiebung gestartet und so kontrolliert, dass die retinale Bildverschiebung möglichst „null“ ist.

Schnelle Augenbewegungen hingegen werden in der retikulären Formation des Hirnstammes generiert. Die PPRF (paramediane pontine retikuläre Formation) erzeugt horizontale, der riMLF (rostraler interstitieller Kern des medialen longitudinalen Fasziulus) vertikale Bewegungen. Hinzu kommen noch die Mikrobewegungen, die wiederum in drei Arten eingeteilt werden: Langsame Mikrobewegungen/Drifts, Mikrosakkaden und Mikrotremor.

Langsame Drifts und Mikrosakkaden dienen dabei der Fixationskontrolle und werden reflexartig ausgeführt, um die Blicklinien immer wieder auf das Fixationsobjekt zurückzuführen, von dem sie aufgrund der Lokaladaptation immer wieder langsam abweichen. Die Geschwindigkeiten der Augenbewegungen werden in den Lehrbüchern unterschiedlich angegeben. In [24] wird die Amplitude von Mikrosakkaden mit weniger als 15 argmin angegeben (Dieser Winkel entspricht ca. $80 \mu\text{m}$ oder einer Verschiebung von ~ 40 Photorezeptoren der Netzhaut). In [25] wird darauf hingewiesen, dass sich Sakkaden durch Müdigkeit/Aufmerksamkeit des Patienten und Helligkeit der Umgebung mit und ohne Fixierungspunkt signifikant verändern. Die Arbeit [26] beschäftigt sich mit dem Versuch, die aus der Astronomie stammende Theorie der „Hauptsequenz“ auf die Beziehungen zwischen Länge, Stärke und Spitzengeschwindigkeit der menschlichen Sakkaden anzuwenden.

Benetzungsstörungen

Benetzungsstörungen des Auges können durch zwei Ursachen hervorgerufen werden.

Die erste Ursache ist die Störung des Tränenfilmverteilersystems, wobei ein normal zusammengesetzter Tränenfilm vorliegt. Ursache hierfür können beispielsweise Lidfehlstellung, Lidfehlfunktionen, Veränderungen der Hornhautoberfläche oder Fehlen des Kornealreflexes sein. Das Krankheitsbild ist abhängig vom Grundleiden, allerdings ist ein Fremdkörpergefühl bei inaktiver Hornhautsensibilität oft ein Frühzeichen.

Die andere Ursache ist eine Störung des Sekretorsystems, wobei die Tränenfilmzusammenstellung pathologisch verändert ist. Hierbei kann es sich um einen Mangel am Gesamtvolumen handeln, aber auch einen Mucin-/Lipidmangel oder einen Mangel an Flüssigkeit in der wässrigen Schicht. Beide Störungen können auch kombiniert auftreten. Therapiemöglichkeiten sind, je nach Grundleiden des Patienten, die Verabreichung von Tränenersatzmittel, Seitenschutz an Brillengestellen und evtl. auch operative Maßnahmen.

2.1.2 Korneale Topographie

Da die Grundlagen dieser Arbeit nahe mit den Verfahren der kornealen Topographie verwandt sind, wird an dieser Stelle ein kleiner Überblick über Verfahren der kornealen Profilmessung gegeben. Korneale Topographiemessungen sind nicht-invasive medizinische Methoden zur Bestimmung der Krümmung und der dreidimensionalen Struktur des Auges. Die Mehrheit der Topographiesysteme im klinischen Einsatz basiert auf einer Reflexionsmethode, wie beispielsweise das Ophthalmometer, das Photo- oder das Videokeratoskop. Bei diesen kornealen Topographieverfahren wird ausgenutzt, dass sich die Kornea ähnlich wie ein konvexer Spiegel verhält. Die Methoden werden unterteilt in reflexionsbasierende und projektionsbasierende Verfahren. Erstere messen die korneale Abweichung von einer Kugelgestalt (oder einem Modell) und ermitteln darüber die Krümmung und Brechkraft. Leider können diese Systeme die Berechnungen nur mit vielen vorher definierten Voraussetzungen berechnen, so dass die korneale Gestalt nicht alleine durch das Messen der Reflexion bestimmt werden kann. Projektionsbasierende Methoden, wie beispielsweise die Slit Photographie, Rasterstereographie, Moire Interferenz und Laser Interferometrie, können direkt das korneale Profil messen (im Sinne von 3D-Höhenangaben), wodurch sich dann Krümmung, Brechkraft und Abweichung von der Kugelgestalt berechnen lassen.

Funktionsweise von reflektiven Methoden

Wie schon erwähnt, kann man die Eigenschaften der Kornea mit einem konvexen

Spiegel vergleichen. Das Bild, welches durch einen konvexen Spiegel zurückgeworfen wird, kann folgendermaßen beschrieben werden (vergl. Abbildung 4):

Die Vergrößerung eines konvexen Spiegels entspricht dem Verhältnis der Bildgröße (I) zur Objektgröße (O) und dies ist proportional zum Verhältnis der Abstände vom Bild (v) und Objekt (u) zum Spiegel:

$$\text{Vergrößerung} = \frac{I}{O} = \frac{v}{u}.$$

In der Praxis kann man nun $v = r/2$ setzen, wodurch man durch Umstellen der Formel

$$I = O \frac{r}{2u}$$

die Formel

$$r = 2u \frac{I}{O}$$

erhält. Dies zeigt nun, dass der Krümmungsradius proportional zur Bildgröße ist, falls die Objektgröße und der u konstant bleiben. Also kann die Vergrößerung des Bildes zur Ermittlung des Krümmungsradius genutzt werden.

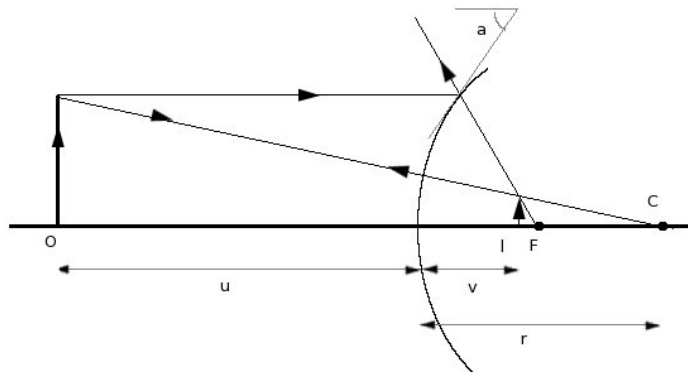


Abbildung 4: schematische Darstellung eines konvexen Spiegels (O = Objekt, I = Bild, F = Fokus, C = Krümmungsmittelpunkt, u = Abstand von Objekt und Spiegelfläche, v = Abstand von Bild und Spiegelfläche, r = Radius der Krümmung der Spiegelfläche)[1]

Im Folgenden werden wir uns näher mit der Videokeratoskopie beschäftigen, die mit dem Thema dieser Arbeit zusammenhängt.

Videokeratoskopie

In diesem Abschnitt wollen wir uns mit der computergestützten placidoringbasierten⁴ Videokeratoskopie beschäftigen, die unserem Ansatz am Ähnlichsten ist. Bei diesem Verfahren wird ein Muster von 15-38 konzentrischen Ringen auf die Kornea projiziert und mittels Videokamera (CCD) wird die Reflexion aufgenommen, digitalisiert und computergestützt analysiert. Dabei werden detaillierte Informationen über die Kontur der Kornea erzeugt. Die Genauigkeit solcher Systeme wird in [31] beschrieben. Im Allgemeinen haben sie bei sphärischen Testobjekten eine Genauigkeit von maximal 0,25D in einem Bereich, äquivalent zu den zentralen 70 % der vorderen kornealen Oberfläche. Durch computergestützte Methoden erreichen viele Verfahren eine Genauigkeit von unter 0,15D, was innerhalb der nötigen medizinischen Toleranz liegt. Im klinischen Einsatz läuft das Verfahren meist folgendermaßen ab:

1. Der Patient wird vor dem Videokeratographen positioniert und fixiert.
2. Das Placidoringmuster wird beleuchtet und auf die Kornea projiziert.
3. Der Arzt fokussiert und justiert die Kamera auf das Muster.
4. Die Aufnahme wird gestartet und der CCD-Sensor nimmt die Bilder auf.
5. Die Positionen der Ringe werden ermittelt.
6. Referenzpunkte werden eingefügt.
7. Die Datenpunkte werden lokalisiert.
8. Algorithmen werden angewendet.
9. Die Ergebnisse werden angezeigt.

Die Qualität der Ausführung dieser Schritte kann die Exaktheit der Methode ebenfalls erheblich beeinflussen.

Die Mehrheit der Videokeratographen nutzt ein auf die Kornea projiziertes Placidoringmuster. Anzahl, Stärke und Abstand der Ringe sind dabei von System zu System unterschiedlich. Position, Größe und Abstand der Ringe auf dem reflektierten Bild werden durch die korneale Form beeinflusst. Leider sind Details der Algorithmen der kommerziellen Hersteller von Videokeratographen wohlgehütete Firmengeheimnisse.

⁴Placidoringe sind konzentrische Ringe

Trotzdem wollen wir hier einen kleinen allgemeinen Überblick über die Verarbeitungstechniken angeben.

Nachdem ein Video aufgenommen und digitalisiert wurde, werden die Einzelbilder analysiert. Als erstes wird ein Referenzpunkt generiert, anhand dessen sich alle anderen Punkte auf den konzentrischen Kreisen identifizieren lassen (Beispielsweise der Mittelpunkt). Danach wird der Schnitt von Teilmeridianen und Ringen gebildet. Bei 15-38 Ringen und 256-360 Meridianen erhält man theoretisch zwischen 6.000 und 11.000 Datenpunkte. In der Praxis weicht diese Zahl allerdings vom theoretischen Wert ab.

Die Datenpunkte werden nun mit Modelldaten verglichen. Dabei wird jedes Gerät vor dem Einsatz mithilfe einer perfekten Kugel kalibriert. Die Abweichungen der Datenpunkte der perfekten Kugel zur gemessenen Kornea können also genutzt werden, um die korneale Form zu ermitteln. Die dreidimensionale Form der gemessenen Datenpunkte werden berechnet bzw. approximiert. Dieser Schritt variiert von System zu System. Oft werden, wie im nächsten Absatz beschrieben, Zernikepolynome genutzt, um die Form zu approximieren. Dadurch können bestimmte Eigenschaften, z.B. Astigmatismus, anhand eines Koeffizienten der Polynome bestimmt werden.

Zum Abschluss wird hier ein Beispiel angegeben, wie in manchen Systemen die Höhendaten (rekonstruierten dreidimensionalen Daten) mithilfe von Zernike-Polynomen ermittelt werden. Zernike-Polynome haben drei Eigenschaften, die sie wesentlich von anderen orthogonalen Polynomen unterscheiden:

1. einfache rotationssymmetrische Darstellung mittels dem polynomialen Produkt

$$R(\rho)G(\phi),$$

wobei $G(\phi)$ eine stetige 2π -periodische Funktion ist und

$$G(\phi + \alpha) = G(\phi)G(\alpha)$$

gilt. Die trigonometrischen Funktionen

$$G(\phi) = e^{\pm im\phi}$$

erfüllen diese Eigenschaften für $m \in \mathbb{N}_0$.

2. Die radiale Funktion muss ein Polynom in ρ mit Grad $n \in \mathbb{N}$ sein und darf keinen

Term vom Grad kleiner m enthalten.

3. Der Grad von $R(\cdot)$ ist gerade, falls m gerade ist, und ungerade, falls m ungerade ist. Daraus ergeben sich die Zernike-Polnome für gerades m als

$$Z_n^m(\rho, \phi) = R_n^m(\rho) \cos(m\phi)$$

und für ungerades m

$$Z_n^{-m}(\rho, \phi) = R_n^m(\rho) \sin(m\phi).$$

Die radialen Polynome R_n^m sind Spezialfälle der Jacobipolynome. Ihre Orthogonalität ist gegeben durch

$$\int_0^1 R_n^m(\rho) R_{n'}^m(\rho) \rho d\rho = \frac{1}{2(n+1)} \delta_{nn'} R_n^m(1),$$

wobei das Kronecker-Delta als

$$\delta_{n,n'} = \begin{cases} 1 & n = n' \\ 0 & \text{sonst} \end{cases}$$

definiert ist, und sie werden durch

$$R_n^m(1) = 1$$

normalisiert.

Nun können beispielsweise

$$W(\rho, \phi) = \sum_{k=0}^8 Z_k R_k(\rho, \phi)$$

berechnet werden und anhand der Koeffizienten aus 4 die Abweichungen der Kornea angegeben werden.

Es gibt nun verschiedene Möglichkeiten, die Zernike-Koeffizienten aus den gegebenen Messdaten zu approximieren. Der einfachste und schnellste Ansatz ist das Polynom-Fitting-Verfahren mittels Least Squares. Ein Beispiel hierfür findet sich in [34], wobei

n	m	Nummer k	Polynom R_k	
0	0	0	1	
1 45°	1	1	$\rho \cos(\phi)$	
	0	2	$\rho \sin(\phi)$	
		3	$2\rho^2 - 1$	
2	2	4	$\rho^2 \cos(2\phi)$	
		5	$\rho^2 \sin(2\phi)$	
		1	6	$(3\rho^2 - 2)\rho \cos(\phi)$
	0	7	$(3\rho^2 - 2)\rho \sin(\phi)$	
		8	$6\rho^4 - 6\rho^2 + 1$	
	3	3	9	$\rho^3 \cos(3\phi)$
			10	$\rho^3 \sin(3\phi)$
2		11	$(4\rho^2 - 3)\rho^2 \cos(2\phi)$	
		12	$(4\rho^2 - 3)\rho^2 \sin(2\phi)$	
1		13	$(10\rho^4 - 12\rho^2 + 3)\rho \cos(\phi)$	
		14	$(10\rho^4 - 12\rho^2 + 3)\rho \sin(\phi)$	
		0	15	$20\rho^6 - 30\rho^4 + 12\rho^2 - 1$

Tabelle 3: Die ersten 15 Zernike-Polynome (Radialer Anteil)

zu beachten ist, dass die Daten mit einem Hartmann-Shack-Sensor aufgenommen wurden.

2.1.3 Analyseverfahren

Es existieren diverse Verfahren zur Bestimmung von Pathologien des Tränenfilms. Ein Verfahren zur Bestimmung des wässrigen Anteils des Tränenfilms ist der Schirmer Test. Hierbei wird ein 3,5 cm langer und 0,5 cm breiter Streifen roten Lackmuspapiers in die Bindehaut der beiden Augen eingehängt. Durch die leicht alkalische Tränensekretion ($\text{pH} = 7,35$) verfärbt sich das Papier blau und nach etwa 5 Minuten kann die Strecke gemessen werden, die von der Tränenflüssigkeit verfärbt wurde. Laut [10] sollte nach 5 Minuten ein mindestens 1,5 cm langes Stück des aus der Lidspalte heraushängenden Streifens befeuchtet sein. Das Verfahren nach Jones ist eine Weiterentwicklung des Schirmertests. Hierbei werden durch vorheriger Applikation eines Oberflächenanästetikums und durch Verdunklung des Raumes die unphysiologischen Reize minimiert. Hierbei ist die Norm 10 mm Befeuchtung in 5 Minuten.

Eine andere Eigenschaft des Tränenfilms ist die Tränenaufrisszeit (engl. Literatur: Break-Up-Time), welche mittels einer Spaltlampe gemessen werden kann. Dabei wird ein Tropfen Fluoreszein (Fluoreszein-Natrium (0,1–0,5 %)) in den Bindehautsack ge-

Koeffizient	Beschreibung
Z_0	
Z_1	Neigung in x-Richtung
Z_2	Neigung in y-Richtung
Z_3	Defokus
Z_4	Astigmatismus bei 0°
Z_5	Astigmatismus bei 45°
Z_6	Koma in x-Richtung
Z_7	Koma in y-Richtung
Z_8	Spärischer Defokus

Tabelle 4: Koeffizienten der Polynome R_k und ihre Eigenschaften.

träufelt und nach mehrmaligem Lidschluss die Stabilität unter Vorschalten eines Kobalt-Blaufilters gemessen. In [10] wird erwähnt, dass eine Festlegung der Norm der physiologischen Benetzungsdauer schwierig ist.

Diese Verfahren sind offensichtlich stark invasiv und verfälschen sich schnell in Relation zur Sensibilität des Augenapparates des Patienten.

Es existieren ebenfalls mehrere nichtinvasive Verfahren⁵, welche vorverarbeitete Bilder aus einem Augenmessgerät einzeln verarbeiten. Im Folgenden werden einige dieser Ansätze beschrieben:

Für das RT-7000 von Tomey existiert beispielsweise das System TSAS (Tear Stability Analysis System). Hierbei werden zur Analyse der Stabilität des Tränenfilms zwei Werte herangezogen: Erstens die Differenz zwischen dem ersten Bild und den Folgebildern und zweitens die Anzahl der Veränderungen an einem Ring. Durch Thresholding wird dann ermittelt, ob es sich um einen Tränenaufriss handelt. Dieses System benutzt ein 6-10 Sekunden Video mit einem Bild pro Sekunde.

In [11] wird eine ähnliche Methode beschrieben. Hierbei werden die Daten aus einem Videokeratographen (TMS-2N) verwendet, um bei einem 10 Sekunden Video (1 Bild pro Sekunde) auf jedem Bild einzeln die Tear-Breakuptime (TMS-BUT) und das Verhältnis des Aufrissareals zur gesamten Farbcode-Fläche (TMS-BUA) zu bestimmen. Diese Werte wurden an 48 Probanden mit dem oben beschriebenen Spaltlampenverfahren (SLE-BUT) verglichen. Dabei wurde eine positive Korrelation von SLE-BUT und TMS-BUT festgestellt und eine negative Korrelation von SLE-BUT und TMS-

⁵Wie schon im vorherigen Kapitel erwähnt, werden die meisten Algorithmen und Techniken solcher Verfahren von den meisten Herstellern geheim gehalten und sind u.U. nicht identisch mit den veröffentlichten Algorithmen

BUA.

Drei Methoden werden in [12] zur Ermittlung der Tränenfilmstärke mittels Interferenz-Methoden angegeben. Dabei werden die drei Methoden wavelenght-dependent fringes, thickness-dependent fringes und angle-dependent fringes diskutiert und verglichen, jedoch nicht mit klinischen Daten hinterlegt.

Eine andere Methode zur Ermittlung der präkornealen Tränenfilmstabilität mittels interferometrischer Technik wird in [13] beschrieben. Dabei werden Interferenzmuster des Tränenfilms mittels schneller Fouriertransformation verarbeitet. Die Methode basiert auf der Evaluation des Grades der Störung des Streifenmusters durch Berechnung des zweiten Moments des Fourierspektrums.

[14] beschreibt ein Verfahren mittels Reflective meniscometry. Diese Methode ist den Verfahren mit Videokeratographdaten ähnlich, jedoch wird statt der Placidoringe ein Muster aus horizontalen Streifen auf das Auge projiziert, um damit den unteren Tränenfilmmeniskus zu messen. Anhand von 45 Probanden wurde dann der normale Radius der Krümmung des unteren Tränenfilmmeniskus ermittelt.

Für den Hartmann-Shack-Sensor wurde in [15] ein Verfahren entwickelt. Dabei wurden 11 Probanden mit einem modifizierten Hartmann-Shack-Sensor aufgenommen und mittels Fourier transform reconstructor Algorithm ([16]) die Wellenfrontabweichungen ermittelt. Erhöhte Unebenheiten, die in der Zeit auftraten, erzeugten sichtbare Wellenfrontvariationen.

optische Aberrationen

In dieser Arbeit geht es um monochromatische Abbildungsfehler, also um Fehler, die nicht von der Wellenlänge des Lichtes abhängen. Um Abbildungsfehler von Linsen zu messen wird in der Optik folgender Ansatz verwendet. Es wird angenommen, dass Lichtstrahlen nahezu parallel zur optischen Achse verlaufen und sich die Abweichungen von der perfekten Linse dieser Lichtstrahlen durch $\sin(x)$ beschreiben lassen. Die Maclaurin Reihe für den Sinus lautet

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1},$$

also $\sin(x) = x - x^3/6 + x^5/120 \dots$. Die Ordnung der Aberrationen ist nun der höchste Grad der Erweiterung. Demnach ist die Aberrationen erster Ordnung $\sin(x) \approx x$. Näheres dazu findet sich in [50]. Zernike beschrieb die Abbildungsfehler am menschlichen Auge mittels Zernike-Polynomen. Für das menschliche Auge werden dabei die

optischen Fehler in Aberrationen niederer (Zernike-Polynome Z_0, \dots, Z_2) und höherer Ordnung (Z_3, \dots, Z_5) unterteilt. In der Literatur findet sich ebenso die Unterteilung in Aberrationen erster (Zernike-Polynome Z_0, \dots, Z_2), zweiter (Z_3, \dots, Z_5) und höherer Ordnung ($Z_j, j > 5$). Wir werden für den Term niederer Ordnung für Zernike Polynom bis zur Ordnung fünf verwenden und den Begriff höherer Ordnung für Zernike Polynome von Ordnung größer gleich sechs. Zernike-Polynome höherer Ordnung ($Z_j, j > 5$) können nur wenig bis gar nicht durch eine Brille korrigiert werden, daher spielen sie in der Augenheilkunde bestenfalls eine untergeordnete Rolle. Es ist bekannt (siehe z.B. [53]), dass optische Aberrationen niederer Ordnung für 85 % der Wellenfrontaberrationen verantwortlich sind. Diese Aberrationen niederer Ordnung (Z_0, \dots, Z_5) beschreiben beim menschlichen Auge die refraktiven Fehler Kurzsichtigkeit (Myopie), Weitsichtigkeit (Hyperopie) und Astigmatismus.

2.2 Technische Grundlagen

Bildverarbeitung (Image Processing) beschäftigt sich mit der Analyse und der Verarbeitung von Bilddaten. Dieses Gebiet ist sehr umfangreich, da durch die fortschreitenden Entwicklungen in der Foto- und Videotechnik weitreichende Anwendungsgebiete im wirtschaftlichen und privaten Bereich geschaffen wurden. Daher wird sich dieses Kapitel nur mit den Teilgebieten der Bildverarbeitung befassen, welche für diese Arbeit relevant sind. Grundlegende und weitergehende Literatur über Bildverarbeitung sind beispielsweise [36] (Grundlegendes über Bildverarbeitung), [37] (Praktische Anwendungen), [38] (3D Bildverarbeitung), [39] (Medizinische Bildverarbeitung) und [40] (Bildverarbeitung und Multiskalenanalyse).

Im Folgenden werden wir uns zunächst mit einfachen Filtereigenschaften beschäftigen, welche wir später zur Vorverarbeitung der Bilder benötigen. Anschließend werden wir uns mit den Grundlagen der „Image Registration“ befassen.

2.2.1 Digitale Signalverarbeitung

Wenn man sich mit Bildverarbeitung (insbesondere mit Bildfiltern) beschäftigt, kommt man um die grundlegenden Theorien und Eigenschaften der digitalen (eindimensionalen) Signalverarbeitung nicht herum. Dabei sind an dieser Stelle als Literaturangaben [20], [41], [42] und [43] besonders zu nennen. Digitale Signalverarbeitung beschäftigt sich mit der Verarbeitung von elektrischen Signalen mittels digitalen Rechnern. Die Schritte eines typischen Systems der digitalen Signalverarbeitung ist im Folgenden abgebildet:

1. Das elektrische Signal wird (analog) aufgezeichnet
2. Das analoge Signal wird analog vorverarbeitet
3. Das Signal wird digital abgetastet
4. Das digitale Signal wird mittels digitaler Filter verarbeitet
5. Das digitale Signal wird in ein analoges Signal umgewandelt
6. Es erfolgt eine analoge Nachbearbeitung der Signals

Die Anwendung der digitalen Signalverarbeitung sind sehr vielfältig und können daher nicht in aller Ausführlichkeit vorgestellt werden. Ein Signal f repräsentiert in der

Praxis meist eine physikalische Größe, wie Geschwindigkeit eines Körpers, die Temperatur eines chemischen Prozesses, Spannungsschwankungen an der Kopfoberfläche beim EEG oder, wie in unserem Fall, elektromagnetische Strahlung.

Mathematisch gesehen wird ein Signal als eine Funktion $f(t)$ mit der Variablen t bezeichnet. Diese weitreichende Definition wird meist auf Funktionen $f : D \rightarrow \mathbb{R}$ eingeschränkt, wobei der Definitionsbereich D beispielsweise zeitbeschränkt ($D = [a, b], a, b \in \mathbb{R}$), zeitdiskret ($D = \mathbb{Z}$), periodisch ($D = \mathbb{T}$) oder unbeschränkt ($D = \mathbb{R}$) sein kann. Formal definiert bedeutet dies Folgendes.

Definition 2.1 Die Menge $L(\mathbb{K})$ wird als Menge aller Funktionen von $\mathbb{K} = \mathbb{R}$, $\mathbb{K} = \mathbb{C}$ oder $\mathbb{K} = \mathbb{T}$ nach \mathbb{C} bezeichnet, wobei \mathbb{T} den Torus bezeichnet.

Die Menge $l(\mathbb{K})$ wird als Menge aller Funktionen von $\mathbb{K} = \mathbb{Z}$ oder $\mathbb{K} = \mathbb{Z}_N$ nach \mathbb{C} bezeichnet, wobei \mathbb{Z}_N den Restklassenring $\mathbb{Z}/n\mathbb{Z}$ bezeichnet.

1. Mit C^n bezeichnen wir die Menge aller n -mal stetig differenzierbaren Funktionen.

Für $n = 0$ ist dies die Menge aller stetigen Funktionen.

2. Mit $L_p(\mathbb{K})$, $p \in \mathbb{N}$, $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$, bezeichnen wir den Signalraum der (p -) integrierbaren Funktionen

$$L_p(\mathbb{K}) = \{f \in L(\mathbb{K}) : \|f\|_p < \infty\},$$

wobei $L(\mathbb{R})$ die Menge aller (lokal integrierbaren) reellwertigen Funktionen bezeichnet.

3. Mit $l_p(\mathbb{Z})$, $p \in \mathbb{N}$ bezeichnen wir den Signalraum der (p -) summierbaren Folgen

$$l_p(\mathbb{Z}) = \{c \in l(\mathbb{Z}) : \|c\|_p < \infty\},$$

wobei $l(\mathbb{Z})$ die Menge aller reellwertigen Folgen bezeichnet.

4. Analog seien die zweidimensionalen Mengen und Räume für $\mathbb{K} = \mathbb{C}^2, \mathbb{R}^2, \mathbb{T}^2, \mathbb{Z}^2, \mathbb{Z}_N^2$ definiert.

Für die Anwendung verwenden wir hier lediglich die Fälle $p = 1, 2$.

Zunächst gehen wir vom eindimensionalen Fall aus.

2.2.2 Faltungen und Fouriertransformationen

Für die Anwendung in der Signalverarbeitung ist die im Folgenden definierte Faltung ein grundlegendes Konzept. Für die Faltung gibt es unterschiedliche Versionen, deren grundlegende Idee jedoch ähnlich ist.

Definition 2.2 (Faltung) 1. Für $f, g \in L_1(\mathbb{R})$ ist die kontinuierliche Faltung als

$$f *_k g(x) = \int_{\mathbb{R}} f(t)g(x-t)dt$$

für fast alle $x \in \mathbb{R}$ definiert.⁶

2. Wir bezeichnen für $c \in l_1(\mathbb{Z}), g \in L_1(\mathbb{R})$ die Faltung

$$c *_{sd} g(x) = \sum_{j \in \mathbb{Z}} c[j]g(x-j)$$

als semidiskrete oder halbdiskrete Faltung.

3. Für zwei Folgen $c, d \in l_1(\mathbb{Z})$ ist die diskrete Faltung als

$$c *_d d[k] = \sum_{j \in \mathbb{Z}} c[j]d[k-j]$$

definiert.

4. Für zwei zyklische Folgen $c, d \in l_1(\mathbb{Z}_N)$ ist die diskrete periodische Faltung als

$$c *_d^N d[k] = \sum_{j \in \mathbb{Z}_N} c[j]d[k-j]$$

definiert.

Da sich die jeweilige Faltung meist aus dem Zusammenhang ergibt, werden wir im Folgenden den Term $*$ als Faltung nutzen.

Die vier verschiedenen Faltungen werden für unterschiedliche Zwecke benötigt.

Die kontinuierliche Faltung dient dazu, theoretische Aussagen beweisen zu können und bei Anwendungen die äquidistante Diskretisierung erst nach einer kontinuierlichen Analyse durchführen zu können.

Die semidiskrete Faltung dient einerseits für theoretische Aussagen, andererseits für Abstraten ungleich \mathbb{Z} und letztendlich ist es die Formel für Interpolation an äquidi-

⁶Im Folgenden gilt für alle Eigenschaften von L_p Räumen der Zusatz „fast überall“.

stanten Stützstellen. Die dritte Definition bietet die Möglichkeit der effektiven Verarbeitung von diskreten Daten und die vierte Faltungsformel wird für die schnelle Verarbeitung der Bilddaten benötigt.

Lemma 2.1 1. Für $f, g \in L_1(\mathbb{R})$ ist $f * g(x) \in L_1(\mathbb{R})$.

2. Für $c \in l_1(\mathbb{Z}), g \in L_1(\mathbb{R})$ ist $c * g(x) \in L_1(\mathbb{R})$.

3. Für $c, d \in l_1(\mathbb{Z})$ ist $c * d[k] \in l_1(\mathbb{Z})$.

4. Für $c, d \in l_1(\mathbb{Z}_N)$ ist $c * d[k] \in l_1(\mathbb{Z}_N)$.

Beweis: 1.

$$\begin{aligned} \int_{\mathbb{R}} \left| \int_{\mathbb{R}} f(t)g(x-t)dt \right| dx &\leq \int_{\mathbb{R}} \int_{\mathbb{R}} |f(t)||g(x)| dt dx \\ &= \int_{\mathbb{R}} |f(t)| dt \int_{\mathbb{R}} |g(x)| dx \end{aligned}$$

Die anderen drei Beweise funktionieren analog. □

Ein wichtiges Werkzeug zur Analyse von Signalen ist die *Fouriertransformierte* bzw. das *Spektrum*. Das Spektrum ist, einfach ausgedrückt, die Zerlegung des Signals in Kosinus- und Sinusschwingungen. Natürlich kann man nicht für jede beliebige Funktion die Fouriertransformierte berechnen, weshalb wir die Signalmräume eingeschränkt haben.

Definition 2.3 (Fouriertransformationen) 1. *Kontinuierliche Fouriertransformation:*

$$\hat{f}(\xi) = \int_{\mathbb{R}} f(x)e^{-i\xi x} dx$$

für $f \in L_1(\mathbb{R})$

2. *Fourierreihe*

$$\hat{c}(\xi) = \sum_{j \in \mathbb{Z}} c[j]e^{-i\xi j}$$

für $c \in l_1(\mathbb{Z})$. Dann gilt $\hat{c} \in L_1(\mathbb{T})$

3. *Diskrete Fouriertransformation (DFT):*

$$\hat{c}[k] = \sum_{j=0}^{N-1} c[j]e^{-2\pi i \frac{kj}{N}}$$

für periodische $c \in l_1(\mathbb{Z}_N)$ mit Periode N .

Bemerkung 2.1 Da die Fouriertransformation in dieser Arbeit nur einen sehr kleinen Anteil einnimmt, werden die Konzepte lediglich definiert bzw. ohne Beweise angegeben, um durch Herleitungen nicht von wesentlicheren Themen abzulenken. In dieser Arbeit wird sowohl die Notation \widehat{f} als auch f^\wedge für die Fouriertransformierte verwendet.

Definition 2.4 (inverse Fouriertransformationen)

1. inverse kontinuierliche Fouriertransformation

$$f^\vee(x) = \frac{1}{2\pi} \int_{\mathbb{R}} f(\xi) e^{i\xi x} d\xi,$$

für $f \in L_1(\mathbb{R})$

2. inverse Fourierreihe

$$c^\vee[k] = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\xi) e^{i\xi k} d\xi,$$

für $f \in L_1(\mathbb{T}), c \in l_1(\mathbb{Z})$

3. inverse diskrete Fouriertransformation

$$c^\vee[k] = \frac{1}{N} \sum_{j=0}^{N-1} c[j] e^{2\pi i \frac{kj}{N}},$$

für $c \in L_1(\mathbb{R})$

In [20] wird nun gezeigt, dass $\widehat{f^\vee} = f$ und $\widehat{c^\vee} = c$ gilt. Die folgenden Sätze zeigen den Zusammenhang zwischen der Faltung und der Fouriertransformation.

Satz 2.2 (Diskret Periodisch) Für $c, d \in l_1(\mathbb{Z}_N)$ und $a := c * d \in l_1(\mathbb{Z}_N)$ gilt:

$$\widehat{a}[k] = \widehat{c}[k] \widehat{d}[k]$$

Beweis:

$$\begin{aligned}
 \widehat{a}[k] &= (c * d)[k] \\
 &= \sum_{j \in \mathbb{Z}_N} \left(\sum_{l \in \mathbb{Z}_N} c[l] d[j-l] \right) e^{2\pi i \frac{kj}{N}} \\
 &= \sum_{j \in \mathbb{Z}_N} \sum_{l \in \mathbb{Z}_N} (c[l] e^{2\pi i \frac{kl}{N}} d[j-l] e^{2\pi i \frac{k(j-l)}{N}}) \\
 &= \widehat{c}[k] \widehat{d}[k]
 \end{aligned}$$

□

Analog zeigt man den folgenden Satz

Satz 2.3 (Diskret) Für $c, d \in l_1(\mathbb{Z})$ ist

$$\widehat{c * d}[k] = \widehat{c}[k] \widehat{d}[k]$$

Dabei ist zu beachten, dass $\widehat{c} \in L_1(\mathbb{T})$ liegt und die Rücktransformation eine Integration ist.

Satz 2.4 (Halbdiskret) Für $a \in l_1(\mathbb{Z})$, $g \in L_1(\mathbb{R})$ sei $h := a * g \in L_1(\mathbb{R})$. Dann gilt:

$$\widehat{h}(\xi) = \widehat{a}(\xi) \widehat{g}(\xi)$$

Beweis:

$$\begin{aligned}
 (a * g)^\wedge(\xi) &= \int_{\mathbb{R}} \sum_{j \in \mathbb{Z}} a[j] g(x-j) e^{-ix\xi} dx \\
 &= \sum_{j \in \mathbb{Z}} \left(\int_{\mathbb{R}} g(x-j) e^{-i(x-j)\xi} dx \right) a[j] e^{-ij\xi}
 \end{aligned}$$

und die Behauptung folgt durch die Variablentransformation $x' = x - j$. □

Satz 2.5 (Kontinuierlich) Für $f, g \in L_1(\mathbb{R})$ ist $h := f * g \in L_1(\mathbb{R})$ und es gilt:

$$\widehat{h}(\xi) = \widehat{f}(\xi) \widehat{g}(\xi)$$

Für die Fall $p = 2$ der L_p und l_p Räume kann ebenfalls eine Fouriertransformation definiert werden. Dabei kann für $f \in L_2(\mathbb{R})$ die Fouriertransformierte nicht mit dem

Fourierintegral berechnet werden, da $f(t)e^{i\xi t}$ nicht notwendigerweise integrierbar ist, daher kann \widehat{f} nicht direkt definiert werden.

Ein Beweis zur Existenz der Fouriertransformierten in L_2 findet sich beispielsweise in [41], wir werden hier lediglich vermerken, dass ein solches Konzept in L_2 existiert.

schnelle Faltung

Die schnelle Faltung beruht auf der schnellen diskreten Fouriertransformation (FFT). Die FFT reduziert den numerischen Aufwand zur Berechnung der Fouriertransformierten eines periodisch diskretes Signals $c \in l_1(\mathbb{Z}_N)$ auf $O(N \log(N))$. Die schnelle Faltung folgt nun für $c, d \in l_1(\mathbb{Z}_N)$ mit Satz, 2.2

$$\widehat{c * d} = \widehat{c} \widehat{d},$$

wodurch sich der Aufwand von $N(N+1)$ Multiplikationen und Additionen auf $O(N \log_2(N))$ verringert. Da die schnelle Faltung ein grundlegendes Verfahren in der Signalverarbeitung ist, wird an dieser Stelle auf [20] verwiesen, ohne weitere Beweise zu diesem Thema anzuführen. Angemerkt sei noch, dass die FFT reelle Daten in Komplexe überführt. Bei der inversen Transformation werden diese Daten zwar wieder in Reelle überführt, allerdings erhöht es den Benötigten Speicherplatz. Eine effiziente Implementierung der schnellen Faltung ist die Softwarebibliothek FFTW⁷, welche in dieser Arbeit zur effizienten Berechnung genutzt wurde.

Es sei angemerkt, dass in dieser Arbeit zur Implementierung der Faltungen die FFT genutzt wird, um die Daten effizient zu verarbeiten.

Für die später folgende Kantendetektion benötigen wir noch folgende Aussagen.

Lemma 2.6 *Ist $g \in L_1(\mathbb{R})$ und $xg \in L_1(\mathbb{R})$, dann gilt*

$$\frac{d}{dx} g^\vee(x) = (i\xi g(\xi))^\vee(x).$$

Lemma 2.7 (Differential) *Ist g mindestens einmal stetig differenzierbar, so gilt:*

$$\frac{d}{dx}(f * g) = (f * \frac{d}{dx}g).$$

⁷<http://www.fftw.org/>

Beweis: Für eine diskrete Folge c und $g \in C^1(\mathbb{R})$ gilt:

$$\begin{aligned} \frac{d}{dx}(c * g) &= \frac{d}{dx} \left(\sum_{j \in \mathbb{Z}} c[j] g(\cdot - j) \right) \\ &= \sum_{j \in \mathbb{Z}} c[j] \frac{d}{dx} g(\cdot - j). \end{aligned}$$

Der Beweis für $f, g \in L_1$ findet sich in geeigneter Literatur über Funktionalanalysis. \square

Im Allgemeinen benötigt man nicht unbedingt die stetige Differenzierbarkeit von g . Geht man von der distributiven Differenzierbarkeit aus, so gilt der obige Satz für alle Distributionen. Ist g stetig differenzierbar, dann stimmt die Ableitung mit der distributiven Ableitung überein.

Bemerkung 2.2 Bei zeitdiskreten Folgen folgt die Eigenschaft mit einer geeigneten Definition der Diskretisierung eines Differentialoperators. Für $a, b \in \mathbb{N}_0$ sei ein Differentialoperator als $Dc[n] = \frac{c[n+a] - c[n-b]}{a-b}$ definiert und es gilt:

$$\begin{aligned} D(c * d)[k] &= D \left(\sum_{j \in \mathbb{Z}} c[j] d[k - j] \right) \\ &= \sum_{j \in \mathbb{Z}} c[j] g[k + a - j] / (a - b) - \sum_{j \in \mathbb{Z}} c[j] g[k - b - j] / (a - b) \\ &= \sum_{j \in \mathbb{Z}} c[j] Dg[k - j]. \end{aligned}$$

Dabei sind folgende drei Diskretisierungen nennenswert:

1. Zentrale Differenz $Dc[n] = \frac{c[n+1] - c[n-1]}{2}$,
2. Vorwärtsdifferenz $Dc[n] = c[n+1] - c[n]$,
3. Rückwärtsdifferenz $Dc[n] = c[n] - c[n-1]$.

Filter

Als diskreten Filter bezeichnet man einen linearen, zeitinvarianten Operator auf $l_2(\mathbb{Z})$.

Definition 2.5 Ein Operator $L : l_2(\mathbb{Z}) \rightarrow l_2(\mathbb{Z})$ wird als Filter bezeichnet, falls dieser linear

$$L(\alpha a + \beta b) = \alpha Fa + \beta Fb, \quad a, b \in l_2(\mathbb{Z}), \alpha, \beta \in \mathbb{R}.$$

und zeitinvariant

$$F(a[n+k]) = Fa[n+k], \quad a \in l_2(\mathbb{Z}), k \in \mathbb{Z}$$

ist.

Sei der diskrete Dirac-Impuls als

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{sonst} \end{cases}$$

definiert. Dann kann jedes diskrete Signal als

$$a[n] = \sum_{k \in \mathbb{Z}} a[k] \delta[n-k]$$

dargestellt werden. Sei der Translationsoperator als

$$\tau_k[a] = a[\cdot + k]$$

definiert, so folgt:

$$\begin{aligned} La[n] &= L\left(\sum_{k \in \mathbb{Z}} a[k] \delta[n-k]\right) \\ &= \sum_{k \in \mathbb{Z}} a[k] L[\delta[n-k]] \\ &= a * l, \end{aligned}$$

mit $L[\tau_{-k}\delta] = l$. l wird als *Impulsantwort* zum Operator L bezeichnet. Die Fourierreihe der Impulsantwort wird als *Transferfunktion* bezeichnet.

2.2.3 Bildverarbeitung

Bei der digitalen Bildverarbeitung wird die eindimensionale Theorie auf zwei Dimensionen ausgeweitet. Die Regeln für die zweidimensionalen Versionen sind analog zu den eindimensionalen Fällen, wenn die Multiplikation von zwei Elementen im Exponenten von e

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = (x_1, x_2)^T \in \mathbb{K}^2$$

und

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = (y_1, y_2)^T \in \mathbb{K}^2$$

als

$$xy := (x_1y_1 + x_2y_2) = x^T y = \langle x, y \rangle$$

definiert wird. Dabei werden wir auf die korrekte Schreibweise der Transponierten $()^T$ aus Gründen der Lesbarkeit verzichten und nur für Fälle verwenden, bei denen sich der Sinn nicht direkt aus dem Kontext ergibt.

Beispielhaft werden wir die Definitionen auf den kontinuierlichen Fall ausweiten, wodurch die Erweiterungen der anderen Fälle auf zwei Dimensionen offensichtlich werden sollte. Für Beweise und nähere Informationen sei wieder auf [20] verwiesen, da wir die Ergebnisse an dieser Stelle lediglich definieren.

Definition 2.6 (zweidimensionale kontinuierliche Fouriertransformation) Für $f \in L_1(\mathbb{R}^2)$ ist die kontinuierliche zweidimensionale Fouriertransformation als

$$\widehat{f}(\xi_1, \xi_2) = \int_{\mathbb{R}} \int_{\mathbb{R}} f(x_1, x_2) e^{-i(\xi_1 x_1 + \xi_2 x_2)} dx_1 dx_2$$

definiert. Als Kurzschreibweise für $x = (x_1, x_2)$, $\xi = (\xi_1, \xi_2)$ wird

$$\widehat{f}(\xi) = \int_{\mathbb{R}^2} f(x) e^{-i(\xi^T x)} dx = \int_{\mathbb{R}^2} f(x) e^{-i(\xi x)} dx.$$

verwendet.

Definition 2.7 (inverse zweidimensionale kontinuierliche Fouriertransformation)

Für $f, \widehat{f} \in L_1(\mathbb{R}^2)$ gilt

$$f(x_1, x_2) = \frac{1}{4\pi^2} \int_{\mathbb{R}} \int_{\mathbb{R}} \widehat{f}(\xi_1, \xi_2) e^{i(\xi_1 x_1 + \xi_2 x_2)} d\xi_1 d\xi_2$$

Für $x = (x_1, x_2)$, $\xi = (\xi_1, \xi_2)$ gilt die Kurzschreibweise

$$f(x) = \frac{1}{4\pi^2} \int_{\mathbb{R}^2} \widehat{f}(\xi) e^{i(\xi x)} d\xi.$$

Definition 2.8 (zweidimensionale kontinuierliche Faltung) für $f, g \in L_1(\mathbb{R}^2)$ ist die Faltung als

$$f *_k g(x_1, x_2) = \int_{\mathbb{R}} \int_{\mathbb{R}} f(t_1, t_2) g(x_1 - t_1, x_2 - t_2) dt_1 dt_2$$

definiert. Für $x = (x_1, x_2)$, $\xi = (\xi_1, \xi_2)$ gilt die Kurzschreibweise

$$f *_k g(x) = \int_{\mathbb{R}^2} f(t) g(x - t) dt.$$

Bemerkung 2.3 Es gilt

$$(f *_k g)^\wedge(\xi) = \widehat{f} \widehat{g}(\xi),$$

wobei \widehat{f} und \widehat{g} punktweise multipliziert wird. Man beachte, dass die Gleichheit nur fast überall gilt.

Bildabtastung

Physikalisch besteht ein Bildsensor aus Pixeln $\alpha \in \mathbb{Z}^2$, welche die eintreffenden elektromagnetischen Strahlen im Bereich des jeweiligen Pixels mitteln. Die mathematische Modellierung kann als

$$f_d[\alpha] = \int_{\mathbb{R}^2} f(x) \chi_{(-1/2, 1/2)^2}(\alpha - x) dx, \quad \alpha \in \mathbb{Z}^2$$

angegeben werden, wobei f_d das gegebene Bild beschreibt. Es ist nun offensichtlich unmöglich, f aus f_d zurückzurechnen, ohne zusätzliche Bedingungen zu stellen. Daher hat die digitale Signalverarbeitung folgenden Ansatz.

Man geht von einer Abtastung

$$f_d[\alpha] = \sum_{\beta \in \mathbb{Z}^2} f(\beta) \delta(\alpha - \beta)$$

aus. Die Fouriertransformierte von f_d ist

$$\widehat{f}_d(\xi) = \sum_{\beta \in \mathbb{Z}^2} f(\beta) e^{-i\beta\xi}.$$

In [20] findet man nun folgendes Resultat zur Abtastung und Rekonstruktion eines Signals f .

Satz 2.8 *Hat \hat{f} den kompakten Träger $[-\pi, \pi]^2$, so gilt:*

$$f(x) = \sum_{\beta \in \mathbb{Z}^2} f(\beta) \phi(x - \beta),$$

mit

$$\phi(x_1, x_2) = \frac{\sin(\pi x_1)}{\pi x_1} \frac{\sin(\pi x_2)}{\pi x_2}.$$

Dies ist die analoge zweidimensionale Version des Shannon-Whittaker Sampling Theorems (siehe z.B. [20]) und wird analog bewiesen.

2.2.4 Kantendetektion

Zu den wichtigsten Werkzeugen der Bildanalyse zählt die Kantendetektion. Dazu benötigen wir zunächst eine Definition, die den Begriff *Kante* mathematisch modelliert. Sei $f \in C^1(\mathbb{R}^2)$.

Der Gradientenvektor sei definiert als

$$\vec{\nabla} f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix}$$

und der Vektor der Richtungsableitung sei der Einheitsvektor $\vec{n} = (\cos(\varphi), \sin(\varphi))^T$ mit

$$\frac{\partial f}{\partial \vec{n}} = \frac{\partial f}{\partial x_1} \cos(\varphi) + \frac{\partial f}{\partial x_2} \sin(\varphi) = (\vec{\nabla} f)^T \vec{n}.$$

Ein Punkt $x \in \mathbb{R}^2$ kann als Kantenpunkt bezeichnet werden, falls der Gradient ein lokales Maximum bzgl. der Umgebung

$$x + \lambda \vec{\nabla} f,$$

mit $|\lambda| < \varepsilon$ (für $\varepsilon > 0$ genügend klein) ist.

Eine Kante oder Kantenkurve ist eine Verkettung der Kantenpunkte.

Für diskrete Bilder $I \in l_2(\mathbb{Z}^2)$ werden die Differentialoperatoren diskretisiert. In der Literatur der Bildverarbeitung ist das *Sobelfilter* eine weit verbreitete Version einer Diskretisierung. Dabei wird das Bild mit den Sobeloperatoren S_x, S_y folgendermaßen verarbeitet:

$$G_x = S_x * I = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * I$$

$$G_y = S_y * I = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I$$

$$\vec{\nabla} f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} \simeq \begin{pmatrix} G_x \\ G_y \end{pmatrix}.$$

In der folgenden Abbildung ist eine Anwendung abgebildet. Offensichtlich wird das Ergebnis (Bild unten) durch Rauschen stark beeinflusst. Daher ist es sinnvoll, das Bild vorher zu entrauschen bzw. zu glätten.

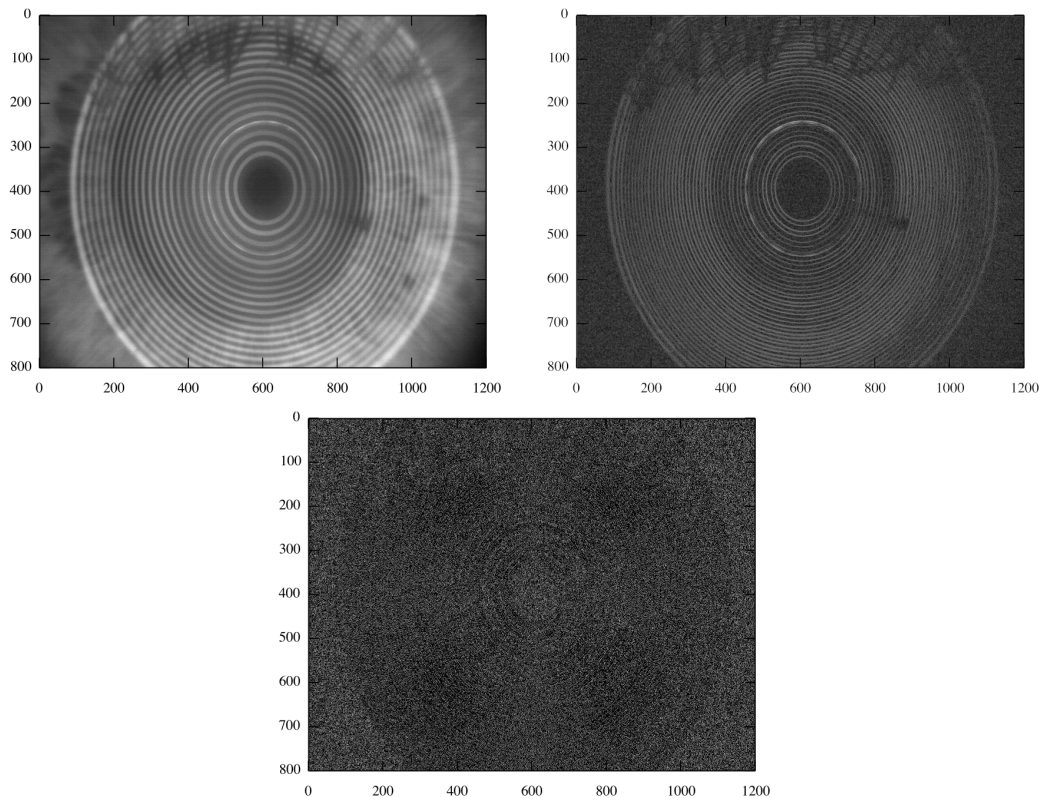


Abbildung 5: Anwendung des Sobeloperators: Das Bild links oben wird mittels Sobelfilter verarbeitet. Der Betrag des approximierten Gradienten $|\vec{\nabla} f|$ ist rechts oben abgebildet. Das untere Bild zeigt nun die lokal Extremen Punkte bzgl. der jeweiligen Filterung.

Im Allgemeinen kann man also nicht davon ausgehen, dass $f \in C^1(\mathbb{R}^2)$ gilt, sondern eher $f \in L_2(\mathbb{R}^2)$ gilt. Dadurch ergeben sich zwei Probleme: Erstens ist f nicht stetig differenzierbar, und daher der Gradient auch nicht unbedingt berechenbar, und zweitens geht man mit diesem Modell davon aus, dass die Bilder mit Rauschen behaftet sind, wodurch sich (bei angenommener stetiger Differenzierbarkeit) für kleine ε die Kanten von dem unterscheiden, was man intuitiv als Kante in einem Bild bezeichnen würde.

Daher nutzen wir folgenden Ansatz zur Kantendetektion.

Definition 2.9 (Glättungsfunktion) Sei $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{R}^2$. Eine Funktion ϕ heißt Glättungsfunktion, falls

$$\int_{\mathbb{K}} \phi(x) dx = 1.$$

Definition 2.10 (Kantenpunkt) Sei $f \in L_2(\mathbb{R}^2)$. Zusätzlich ist eine Familie von Glättungsfunktionen $\phi_s \in C^1(\mathbb{R}^2)$, $s \in \Omega$ gegeben. Ein Punkt $x \in \mathbb{R}^2$ wird als Kantenpunkt der Klasse s bezeichnet, falls der Gradient

$$\vec{\nabla}(f * \phi_s) = \begin{pmatrix} \frac{\partial(f * \phi_s)}{\partial x_1} \\ \frac{\partial(f * \phi_s)}{\partial x_2} \end{pmatrix}$$

ein lokales Extremum bzgl. der Umgebung

$$x + \lambda \vec{\nabla}(f * \phi_s)$$

mit $|\lambda| < \varepsilon$ (für $\varepsilon > 0$ genügend klein) ist.

Ein Kantenpunkt zu einer vorgegebenen Menge Ω ist nun ein Punkt $x \in \mathbb{R}^2$, so dass x für alle $s \in \Omega$ ein Kantenpunkt der Klasse s bezüglich der Glättungsfunktion ϕ_s ist. Als Kante bezeichnet man eine Verkettung dieser Punkte.

Beispiel 2.1 1. Multiskalen Wavelet Edge Detection:

Für $s \in \mathbb{R}^+$ wird $\phi_s = \frac{1}{s} \phi_s(-\frac{\cdot}{s})$ definiert. Besonders die dyadische Verarbeitung mit $s := 2^j$, $j \in \mathbb{N}_0$ ist für die diskrete Bildverarbeitung interessant. Mehr zu diesem Thema findet sich in [20].

2. Canny Edge Detector:

In [61] wird eine Methode beschrieben, bei welcher $\phi_\sigma(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$ gesetzt wird und das Bild somit mit einem Gaußkern geglättet wird. Dabei kann $s = \sigma \in \mathbb{R}^+$ gesetzt werden.

3. B-Splines:

Für $s \in \mathbb{N}$ wird

$$\phi_1 = \chi_{[-a, b]^2}, a \in \mathbb{R}_0^+, b \in \mathbb{R}^+$$

und

$$\phi_s = \phi_{s-1} * \phi_1$$

definiert. Für $a = 0, b = 1$ handelt es sich hierbei um die kardinalen B-Splines und für $a = b = 1/2$ um die zentrierten kardinalen B-Splines der Ordnung s .

Definiert man nun

$$\Psi_s^1 = \frac{\partial(f * \phi_s)}{\partial x_1},$$

$$\Psi_s^2 = \frac{\partial(f * \phi_s)}{\partial x_2},$$

so erhält man folgenden Zusammenhang:

$$\begin{aligned} \begin{pmatrix} W^1 f(u, s) \\ W^2 f(u, s) \end{pmatrix} &:= \begin{pmatrix} f * \Psi_s^1(u) \\ f * \Psi_s^2(u) \end{pmatrix} \\ &= \begin{pmatrix} f * \frac{\partial}{\partial u_1} \phi_s(u) \\ f * \frac{\partial}{\partial u_2} \phi_s(u) \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial}{\partial u_1} (f * \phi_s)(u) \\ \frac{\partial}{\partial u_2} (f * \phi_s)(u) \end{pmatrix} \\ &= \vec{\nabla}(f * \phi_s)(u) \end{aligned}$$

Es sei

$$Mf(u, s) = \sqrt{|W^1 f(u, s)|^2 + |W^2 f(u, s)|^2}$$

und

$$Af(u, s) = \begin{cases} \varphi(u), & \text{falls } W^1 f(u, s) \geq 0, \\ \pi - \varphi(u), & \text{falls } W^1 f(u, s) < 0, \end{cases}$$

wobei $\varphi(u) = \arctan\left(\frac{W^2 f(u, s)}{W^1 f(u, s)}\right)$ ist. Der Einheitsvektor⁸

$$\vec{n}_s(u) = \frac{\vec{\nabla}(f * \phi_s)(u)}{\|\vec{\nabla}(f * \phi_s)(u)\|} = (\cos(Af(u, s)), \sin(Af(u, s)))$$

ist kollinear zu $\vec{\nabla}(f * \phi_s)(u)$. Ein Kantenpunkt in der Klasse s sei nun ein Punkt v , so dass $|M(u, s)|$ ein lokales Maximum an der Stelle v bzgl. der Umgebung $v + \lambda \vec{n}_s(v)$, für $|\lambda| < \varepsilon$ ist. In [20] werden solche Punkte als *wavelet transform modulus maxima* bezeichnet. Diese Kantenpunkte sind meist auf Kurven verteilt, welche auf Bildern die

⁸Die rechte Definition dient zur theoretischen Ansicht, die linke wird zur praktischen Berechnung genutzt.

Ränder von Strukturen und Objekten markieren. Diese Kurven können durch Verkettung von einzelnen Kantenpunkten erzeugt werden. An jeder beliebigen Stelle kann man die Tangente der Kantenkurve durch die Berechnung der Tangenten der Levelmenge approximieren. Diese Tangentenrichtung kann man dann zur Verkettung der Kantenpunkte, die auf einer Kantenkurve liegen, nutzen.

Bemerkung 2.4 Sei $z \in \mathbb{C}$. so lässt sich z schreiben als

$$z = x + iy = re^{i\phi}.$$

Dabei gelten folgende Beziehungen

$$\begin{aligned} x &= \cos(\phi) &= \operatorname{Re}(z) \\ y &= \sin(\phi) &= \operatorname{Im}(z) \\ r &= \sqrt{x^2 + y^2} &= \operatorname{abs}(z) \\ &= \operatorname{arctan} 2(y, x) &= \operatorname{arg}(z) \end{aligned}$$

$\operatorname{arctan} 2$ bezeichnet eine von x und y abhängige Fallunterscheidung für die Berechnung des arctan . Anders ausgedrückt kann man den Winkel auch schreiben als

$$\phi = \begin{cases} \arccos\left(\frac{x}{\sqrt{x^2+y^2}}\right) & , \quad \text{falls } y \geq 0 \\ -\arccos\left(\frac{x}{\sqrt{x^2+y^2}}\right) & , \quad \text{falls } y < 0 \\ N/A & , \quad \text{falls } \sqrt{x^2+y^2} = 0 \end{cases}.$$

Sei eine zweidimensionale Funktion g gegeben. Die Levelmenge⁹ von g sind die Kurven $x(t)$, an denen g konstant ist. Sei $\vec{\tau} = (\tau_1, \tau_2)^T$ die Richtung der Tangente von $x(t)$, dann ist

$$\frac{\partial g(x(s))}{\partial s} = \frac{\partial g}{\partial x_1} \tau_1 + \frac{\partial g}{\partial x_2} \tau_2 = (\vec{\nabla} g)^T \vec{\tau} = 0,$$

da $g(x(t))$ konstant ist. Also ist $\vec{\nabla} g$ senkrecht zur Richtung $\vec{\tau}$ der Tangenten der Level Menge, die durch x laufen.

Setzt man $g = f * \phi_s$, so ist an einem lokal maximalen Punkt v der Vektor

$\vec{n}_s(u) = (\cos(Af(u, s)), \sin(Af(u, s)))^T$ zur Klasse s senkrecht zur Levelmenge von

⁹level-set, auch bekannt als Nullstellenmenge oder Niveaumenge

$f * \Phi_s$ durch v . Falls die Grauwertintensität einer Kante (nahezu) konstant bleibt, sind die maximalen Punkte entlang einer Level Menge. Die Tangente dieser Maximalkurve ist daher senkrecht zu $\vec{n}_s(v)$.

Bemerkung 2.5 *Bei Anwendungen ist es natürlich sehr unwahrscheinlich ¹⁰, dass f bzw. g auf einer Kurve konstant ist. Man kann aber davon ausgehen, dass die Varianz auf solchen Kurven relativ klein ist, was wiederum bei der Implementierung durch einen einfachen Threshold realisiert werden kann. Zusätzlich werden die Varianzen durch die Glättungsfunktionen verkleinert.*

¹⁰besonders wenn L_2 Funktionen betrachtet werden

2.2.5 Image Registration

Image Registration ist eine Prozedur, welche die beste räumliche Anordnung von zwei (oder mehr) Bildern ermittelt. Im Allgemeinen kann es sich nicht nur um Bilder, sondern um beliebige Datensätze handeln (bspw. 3D-Daten aus einem MRT, Audiodaten, etc.). Bei den Bildern kann es sich um Daten handeln, welche zu gleichen oder unterschiedlichen Zeitpunkten, von identischen oder unterschiedlichen Sensoren und aus derselben oder unterschiedlichen Perspektiven aufgenommen wurden. Dies ist ein wichtiger Schritt, um zwei Datensätze vergleichen zu können.

Beispielsweise unterliegen Bilddaten einer sich bewegenden Kamera grundlegenden geometrischen Verzerrungen. Einige einfache Verzerrungen und ihre mathematischen Modelle sind in Tabelle 5 aufgeführt.¹¹ Die einfachste Verzerrung ist die Verschiebung

Name	Formel
Translation	$x_0 = a_0 + x_1$ $y_0 = b_0 + y_1$
Skalierung	$x_0 = a_1 x_1$ $y_0 = b_1 y_1$
Schiefe	$x_0 = x_1 a_2 y_1$ $y_0 = y_1$
perspektivische Verzerrung	$x_0 = a_3 x_1 y_1$ $y_0 = y_1$
Rotation mit Winkel φ	$x_0 = x_1 \cos(\varphi) - y_1 \sin(\varphi)$ $y_0 = x_1 \sin(\varphi) + y_1 \cos(\varphi)$

Tabelle 5: Beispiele allgemeiner Deformationen

(Translation) der Bilddaten, welche sich mittels maximalem Korrelationskoeffizienten von zwei Bildern ermitteln lässt.

Image Registration besteht aus 4 Schritten: *Feature Detection*, *Feature Matching*, *Transform Model Estimation* und *Target resampling and transformation* (vergleiche [4])

1. Feature Detection dient dazu, aus Reference (Input) and Target (Output) Data, bestimmte Features zu ermitteln, welche in beiden Daten erhalten sind.
2. Im Feature Matching Schritt wird für jedes Feature aus den Reference Daten ein korrespondierende Feature aus den Target Daten ermittelt .
3. Beim Transform Model Estimation Schritt wird für ein vorgegebenes Motion Model die geometrische Verformung/ Veränderung berechnet.

¹¹Die Verzerrungen können natürlich auch kombiniert auftreten.

4. Beim Image Resampling werden die Target Daten entsprechend der berechneten Transform Model Estimation transformiert. Also werden vier Voraussetzungen benötigt, um ein Verfahren für Image Registration zu ermitteln: Die *Referenzdaten* und die *Zieldaten* (auch *Reference* und *Target Data*), das *Bewegungsmodell* (*Motion Model*, *Transformation Model* oder *Deformation Model*), die *Fehlerfunktion* (*Similarity Criterion*) und die *Optimierungsmethode* (*optimization method*).

Beispiel 2.2 (Translation bei Funktionen) Seien $f_1, f_2 \in C_{00}(\mathbb{R})$, wobei gilt: $f_1(x) = f_2(x+t), x \in \mathbb{R}$. Es soll die Verschiebung t berechnet werden.

1. f_1 wird als *Reference Data* und f_2 als *Target Data* gewählt. *Feature Points* seien alle $x \in \mathbb{R}$.

2. Die *Optimierungsmethode* sei die Maximierung von $g(k) = \int_{\mathbb{R}} f_1(x)f_2(x+k)dx$, indem alle Werte für $k \in \mathbb{R}$ berechnet werden und die *Fehlerfunktion* sei $\int_{\mathbb{R}} (f_1(x) - f_2(x+k))^2 dk$.

3. Das *Motion Model* sei $x+t$ und die *Transform Model Estimation* ergibt sich aus $t = \operatorname{argmax}_{k \in \mathbb{R}} g(k)$.

4. Das Ergebnis ist nun $\tilde{f}_2(x) = f_2(x+t)$.

Beliebte Fehlerfunktionen bei Bildern:

- Summe der Quadratdifferenzen
- Korrelationskoeffizient
- Korrelationsquotient
- Information
- Normalized Mutual Information

Eine andere nennenswerte Möglichkeit, Bilder aneinander anzugleichen, ist der optische Fluss (engl. optical flow). Sei $I \in C^2(\mathbb{R}^3)$ eine (zweimal stetig differenzierbare) Bildsequenz gegeben¹². Wir berechnen das Taylorpolynom

$$\begin{aligned} I(x+f_1, y+f_2, t+1) &= I(x, y, t) + \frac{\partial I(x, y, t)}{\partial x}(x+f_1-x) \\ &+ \frac{\partial I(x, y, t)}{\partial y}(y+f_2-y) + \frac{\partial I(x, y, t)}{\partial t}(t+1-t) + R, \end{aligned}$$

¹²Dabei ist mit $I(x, y, t)$ der Wert an der Position (x, y) zum Zeitpunkt t gemeint.

wobei R die Terme höherer Ordnung sind. Unter der Annahme

$$I(x + f_1, y + f_2, t + 1) = I(x, y, t),$$

erhält man (unter der zusätzlicher Annahme $R = 0$)

$$0 = I(x + f_1, y + f_2, t + 1) - I(x, y, t) = \frac{\partial I(x, y, t)}{\partial x}(f_1) + \frac{\partial I(x, y, t)}{\partial y}(f_2) + \frac{\partial I(x, y, t)}{\partial t}.$$

Daraus ergibt sich der gesuchte optische Fluss $f = (f_1, f_2)^T$ für eine Bildsequenz $I \in C^1(\mathbb{R}^3)$ mit $I : \mathbb{R}^3 \rightarrow \mathbb{R}$, $I(x, y, t) = g$, durch die *Kontinuitätsgleichung*

$$\frac{\partial I}{\partial t} + f_1 \frac{\partial I}{\partial x} + f_2 \frac{\partial I}{\partial y} = 0$$

bzw.

$$f_1 \frac{\partial I}{\partial x} + f_2 \frac{\partial I}{\partial y} = -\frac{\partial I}{\partial t}.$$

Es gibt nun diverse Ansätze und Algorithmen, um dieses Problem zu lösen¹³

Weitergehende Informationen zu diesen einfachen Modellen findet sich in [36]-[40]

¹³Allgemein ist es nicht eindeutig lösbar, da es eine Gleichung mit zwei Unbekannten ist.

2.2.6 Splines

In diesem Abschnitt sollen grundlegende Eigenschaften von Splines beschrieben werden. Ein Spline ist eine Funktion, die aus stückweisen Polynomen von Grad m zusammengesetzt ist. Zusätzlich wird verlangt, dass der Übergang an den sogenannten Bruchstellen der stückweisen Polynome $(m - 1)$ -fach stetig differenzierbar ist.

Zunächst beschäftigen wir uns mit der kubischen Splineinterpolation.

Zu gegebenen Daten $f(\tau_0), \dots, f(\tau_n)$ mit Bruchstellen¹⁴ $\tau_i \in [a, b]$, $a = \tau_0 < \dots < \tau_n = b$ nennt man eine Funktion s , welche auf jedem Intervall $[\tau_i, \tau_{i+1}]$ einem kubischen Polynom s_i entspricht und zusätzlich

$$\begin{aligned} s_i(\tau_i) &= f(\tau_i), \\ s_i(\tau_{i+1}) &= f(\tau_{i+1}), i = 0, \dots, n-1 \end{aligned}$$

erfüllt, einen stückweisen kubischen Interpolanten.

Einen stückweisen kubischen Interpolanten nennt man kubischen Splineinterpolanten, falls zusätzlich die Bedingungen:

$$\begin{aligned} s'_i(\tau_i) &= s'_{i+1}(\tau_i), \\ s''_i(\tau_i) &= s''_{i+1}(\tau_i), \quad i = 1, \dots, n-1 \end{aligned}$$

gelten. Durch diese Bedingungen ist ein kubischer Splineinterpolant im Intervall $[a, b]$ zweimal stetig differenzierbar. Jedes stückweise Polynom s_i kann als

$$s_i = a_i x^3 + b_i x^2 + c_i x + d_i$$

geschrieben werden, wodurch wir $4n$ zu bestimmende Freiheitsgrade erhalten.

An einen kubischen Splineinterpolanten werden $4n - 2$ Bedingungen gestellt, wodurch zwei weitere Bedingungen gestellt werden können, welche als Randbedingungen bezeichnet werden. Im folgenden Beispiel sind einige mögliche Randbedingungen aufgeführt.

Beispiel 2.3 1. freier Rand oder natürliche Randbedingung:

$$s''_0(\tau_0) = s''_n(\tau_n) = 0.$$

¹⁴Bei stückweisen Polynomen werden die τ_i als Bruchstellen bezeichnet. Bei Splines werden diese Knoten genannt.

2. *periodische Randbedingung:*

$$s'_0(\tau_0) = s'_n(\tau_n), \quad s''_0(\tau_0) = s''_n(\tau_n).$$

3. *Falls $g'(\tau_0)$ und $g'(\tau_n)$ bekannt sind, so können*

$$s'_0(\tau_0) = g'(\tau_0), \quad s'_0(\tau_n) = g'(\tau_n),$$

als Randbedingungen angenommen werden. Mit neu eingeführten Bruchstellen $\tau_{-1} := \tau_0$ und $\tau_{n+1} := \tau_n$ stimmt der Interpolant an den Bruchstellen $\tau_{-1}, \dots, \tau_{n+1}$ mit der Funktion g überein und wird als vollständiger kubischer Splineinterpolant bezeichnet.

Weitere Beispiele finden sich in [59], aus welchem auch das folgende Korollar stammt.

Korollar 2.9 *Von allen zweimal stetig-differenzierbaren Funktionen f , welche mit g an den Bruchstellen $\tau_{-1}, \dots, \tau_{n+1}$ übereinstimmen, ist der vollständige kubische Splineinterpolant die eindeutige Funktion, welche*

$$\int_a^b (f''(t))^2 dt$$

minimiert.

Bemerkung 2.6 *Das Minimierungsintegral im Korollar sagt aus, dass die Biegeenergie minimiert werden. Umgangssprachlich bedeutet dies, dass der Interpolant mit minimalen Wendungen die Werte an den Bruchstellen interpoliert, also möglichst wenig von der linearen Least-Squares-Approximation abweicht.*

An dieser Stelle werden wir B-Splines vom Grad m erneut definieren. Allerdings sei diesmal eine monoton aufsteigende Knotenfolge

$$T_{n,m} = \{t_0, \dots, t_{n+m+1}\}, \quad t_j \leq t_{j+m+1}, \quad j = 0, \dots, n$$

vorgegeben, welche, anders als bei der Definition als Kantenfilter, nicht unbedingt äquidistant sein muss.

Definition 2.11 (B-Spline) Für $m = 0$ gilt

$$B_{k,0}(x|T) = \begin{cases} 1, & t_k \leq x < t_{k+1} \\ 0, & \text{sonst} \end{cases}.$$

Für $m > 0$ gilt

$$B_{k,m}(x|T) := \frac{x - t_k}{t_{k+m-1} - t_k} B_{k,m-1}(x|T) + \frac{t_{k+m} - x}{t_{k+m} - t_{k+1}} B_{k+1,m-1}(x|T)$$

Bemerkung 2.7 Jeder B-Spline $B_{k,m}$ ist eine nichtnegative Funktion und hat den kompakten Träger $[t_k, t_{k+m+1}]$. Ist der Grad m vorgegeben, so wird ein B-Spline auch einfach als B_k geschrieben.

Bemerkung 2.8 Seien $n + 1$ Datenpunkte $(x_0, y_0), \dots, (x_n, y_n)$ gegeben, wobei die Interpolationspunkte x_j äquidistant sein sollen, also

$$x_k = x_{k-1} + h.$$

Sei als Grad $m = 3$ gewählt, so können die Knoten als

$$\begin{aligned} t_0 &= x_0 - 2h, \\ t_1 &= x_0 - h \\ t_{k+m-1} &= x_k, \quad k = 0, \dots, n \\ t_{n+m} &= x_n + h, \\ t_{n+m+1} &= x_n + 2h \end{aligned}$$

gewählt werden.

So können also $n + 1 + m + 1$ Knoten definiert werden, die für die B-Spline Interpolation benötigt werden.

B-Splines bilden eine Basis für den Raum der stückweise Polynome vom Grad k auf dem Intervall $[a, b]$. Der längliche Beweis kann in [59] nachgelesen werden. Dies bedeutet, dass ein Spline auf einem Intervall $[a, b]$ auch als Linearkombination von B-Splines dargestellt werden kann.

Für die B-Spline-Interpolation muss das lineare Gleichungssystem

$$\sum_{k=0}^n \alpha_k B_k(\tau_i) = f(\tau_i) \quad (1)$$

gelöst werden. Der folgende Satz beantwortet nun die Frage, wann das Gleichungssystem lösbar ist.

Satz 2.10 (Schönberg-Whitney) *Sei τ streng monoton steigend, so dass aus $a < t_i = \dots = t_{i+r} = \tau_j < b$, $r < k$ folgt. Dann ist die Matrix $A := (B_j(\tau_i))$ des linearen Gleichungssystems 1 genau dann invertierbar, falls*

$$B_i(\tau_i) \neq 0, \quad i = 1, \dots, n.$$

Dies gilt genau dann, wenn $t_i < \tau_i < t_{i+k}$ für alle i gilt.

Bemerkung 2.9 *Wir nehmen die Voraussetzungen aus Bemerkung 2.8 an.*

Dann gilt

$$t_j < t_{j+m-1} = x_j < t_{j+m+1}$$

wodurch die Schönberg-Whitney-Bedingung erfüllt ist.

3 Verarbeitung

Die Verarbeitung der Daten ist in mehrere Schritte unterteilt:

(0. Voranalyse)

1. Image Registration
2. Waveletalgorithmus
3. Auswertung.

Im ersten Schritt werden die Bewegungsartefakte der Augenbewegungen ermittelt und aus beiden Kameradatensätzen herausgerechnet. Im zweiten Schritt wird der Waveletalgorithmus aus [7] auf die Daten der beiden Kameras angewendet. Hierbei wird das Zeit-Frequenz-Verhalten ermittelt, woraus sich die Features für den dritten Schritt ergeben. Die abschließende Verarbeitung stellt eine statistische Auswertung der erhobenen Features dar, um damit auf Pathologien des Tränenfilms zu schließen.

In dieser Arbeit wird die Voranalyse und der erste Schritt der Verarbeitung dargestellt. Der zweite Schritt findet sich wie schon erwähnt in [7] und der Dritte wird in einer separaten Arbeit veröffentlicht werden.

3.1 Voranalyse

Bevor das aufgenommene Video analytisch untersucht werden kann, muss es erst zeitlich in seine Phasen aufgeteilt werden. Das Verhalten des Tränenfilms lässt sich in periodisch wiederkehrende Phasen unterteilen. Eine Periode des Tränenfilms besteht aus Phasen: Blinzeln, Aufbau, Stabilitätsphase, Degeneration. Es konnte ermittelt werden, dass die Varianz über die Pixel der einzelnen Bilder in der Blinzelpphase signifikant abfällt. Der Aufbau des Tränenfilmes beginnt und endet ziemlich direkt mit dem Öffnen des Auges ([32]). Danach verläuft die wässrige Schicht in eine Richtung (meist horizontal nach oben, manchmal auch vertikal von der Nase weg oder zur Nase). Die Lipidschicht, die auf der wässrigen Phase schwimmt, wird dabei entsprechend mitbewegt. Die Geschwindigkeit, mit der sich der Film bewegt, nimmt stetig ab und mündet in der Degenerationsphase des Tränenfilmes. In der Degenerationsphase beginnt der Tränenfilm an einigen Stellen aufzubrechen.

Für uns interessant sind lediglich die Phasen eines Videos, in denen das Auge geöffnet ist. Daher müssen die Videos zunächst in Blinzelp und Nicht-Blinzelphasen eingeteilt werden. Dazu werden die Bilder der Videos in die Klassen „offenes Auge“ und „ge-

schlossenes Auge“ eingeteilt. Abbildung 6 zeigt jeweils ein exemplarisches Bild der Klassen.

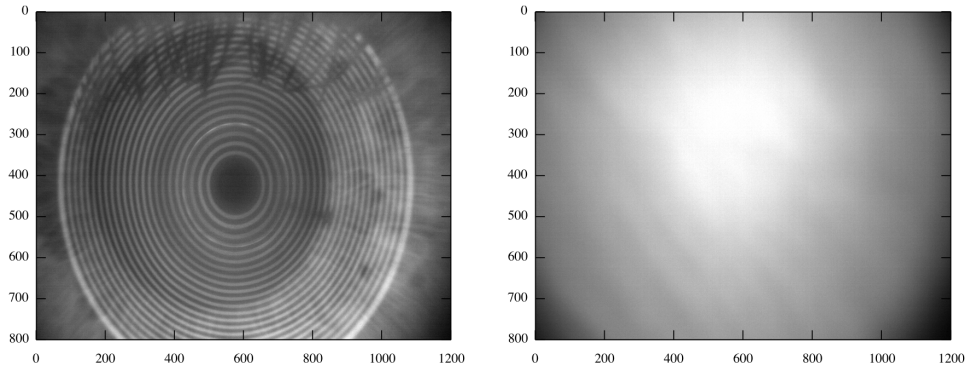


Abbildung 6: Das linke Bild zeigt das geöffnete Auge mit dem gleichmäßig projizierten Muster. Das rechte Bild zeigt das geschlossene Auge. Die Reflexionseigenschaften des Lids sorgen für eine Streuung des Musters.

Als Eigenschaften wurden die folgenden beiden schnell berechenbaren Eigenschaften der Bilder genutzt:

Definition 3.1 (Mittelwert und Varianz) Der Mittelwert einer Funktion $f \in L([-1, 1])$ wird definiert als

$$E(f) = \frac{1}{2} \int_{-1}^1 f(x) dx.$$

Die Varianz der Funktion ist definiert als

$$V(f) = E((f - E(f))^2).$$

Daraus leiten wir den Mittelwert und die Varianz für Bilder ab

Definition 3.2 (Mittelwert und Varianz für Bilder) Der Mittelwert eines Bildes $I \in l([-N, N]^2), N \in \mathbb{N}$ wird definiert als

$$E(f) = \frac{1}{N^2} \sum_{j,k} I(j,k).$$

Die Varianz der ist definiert als

$$V(f) = E((f - E(f))^2).$$

Mit dieser Idee wurden 905 Bilder in offene und geschlossene Augen eingeteilt und der Mittelwert und die Varianz berechnet. Zunächst konnte ermittelt werden, dass der Mittelwert der offenen Augen wesentlich niedriger war, als der Mittelwert der geschlossenen Augen. Allerdings konnte bei halbgeöffneten Augen (obere Hälfte des Bildes wie Klasse „geschlossenes Auge“ untere Hälfte des Bildes wie Klasse „offenes Auge“) kein signifikanter Unterschied der Mittelwerte ermittelt werden. Die Varianz dieser Bilder war allerdings signifikant größer, als die der Klasse „offenes Auge“. In der folgenden Abbildung sind die ermittelten Klassen abgebildet.

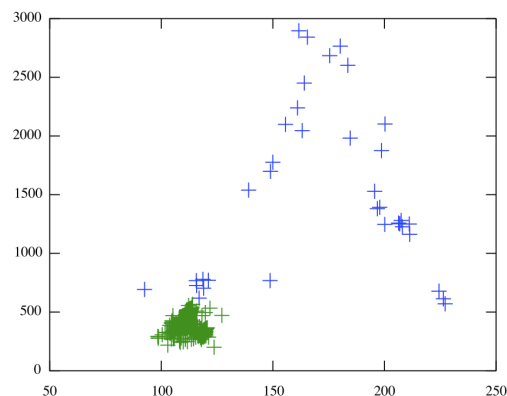


Abbildung 7: Plot der Mittelwerte auf der x- und der Varianzen auf der y-Achse der 905 Testbilder. Die Klasse der „offenen Augen“ ist grün und die Klasse der „geschlossenen Augen“ blau.

Somit können die Videos nun in die Phasen eingeteilt werden, in denen das Auge geöffnet ist. Zusätzlich kann ein erster Referenzwert für das Dry-Eye-Syndrom definiert werden: Die maximale Länge, die der Proband das Auge geöffnet lassen kann. Im Folgenden verwenden wir nun die Phase des geöffneten Auges eines Videos, welche die längste Dauer hat.

3.2 Image Registration

Damit kleine Augenbewegungen die Waveletanalyse nicht stören und die Daten verfälschen, müssen die Bilder vorher aneinander angepasst werden. Eine endgültige Aussage über die Qualität des Algorithmus kann allerdings erst dann gemacht werden, wenn genügend Datensätze zur empirischen Auswertung vorhanden sind. Es sei noch

bemerkt, dass dieses Problem aufgrund der Hardware (CMOS-Sensoren, Optik) und der Aufgabenstellung hochgradig nichttrivial ist. Die Optik leitet nur jeweils etwa 4 % des einfallenden Lichtes an die jeweilige Kamera weiter. Außerdem sind (preiswertere) CMOS-Kamerasensoren (Complementary Metal-Oxide Semiconductor) im Vergleich zu CCD-Lichtsensoren (Charge Coupled Devices) wesentlich lichtunempfindlicher, kontrastärmer und stärker durch Rauschen belastet. Daher lassen sich die Standardalgorithmen der Videokeratoskopie auf die Bilder nicht anwenden.

Aufgabe der Image Registration ist es nun, ein gegebenes Bild an ein Zweites anzugleichen. Bei dieser Anwendung sind also Abbildungen T_1, T_2 gesucht, so dass ein Bild $I \in L_2(\mathbb{R}^2)$ mit

$$I'(x, y) = I(T_1(x, y), T_2(x, y)).$$

auf eine Idealmaske $I' \in L_2(\mathbb{R}^2)$ abgebildet werden kann. Zum besseren Verständnis betrachten wir folgende Abbildung¹⁵.

In Kapitel 2.2 haben wir den Image Registration Prozess in vier Schritte unterteilt: *Feature Detection*, *Feature Matching*, *Transform Model Estimation* und *Target resampling and transformation*. Diese werden wir nun an die Anwendung anpassen. Für die Feature Detection benötigen wir zunächst Kantenbilder.

¹⁵Das Interferenzmuster ist das Differenzbild eines Bildes mit Quadratmuster ($I(x, y)$) und dem mittels den Abbildungen T_1, T_2 geometrisch verzerrten Quadratmusterbildes ($I(T_1(x, y), T_2(x, y))$). Es verdeutlicht die geometrische Veränderung vor und nach dem Image Registration Prozess.

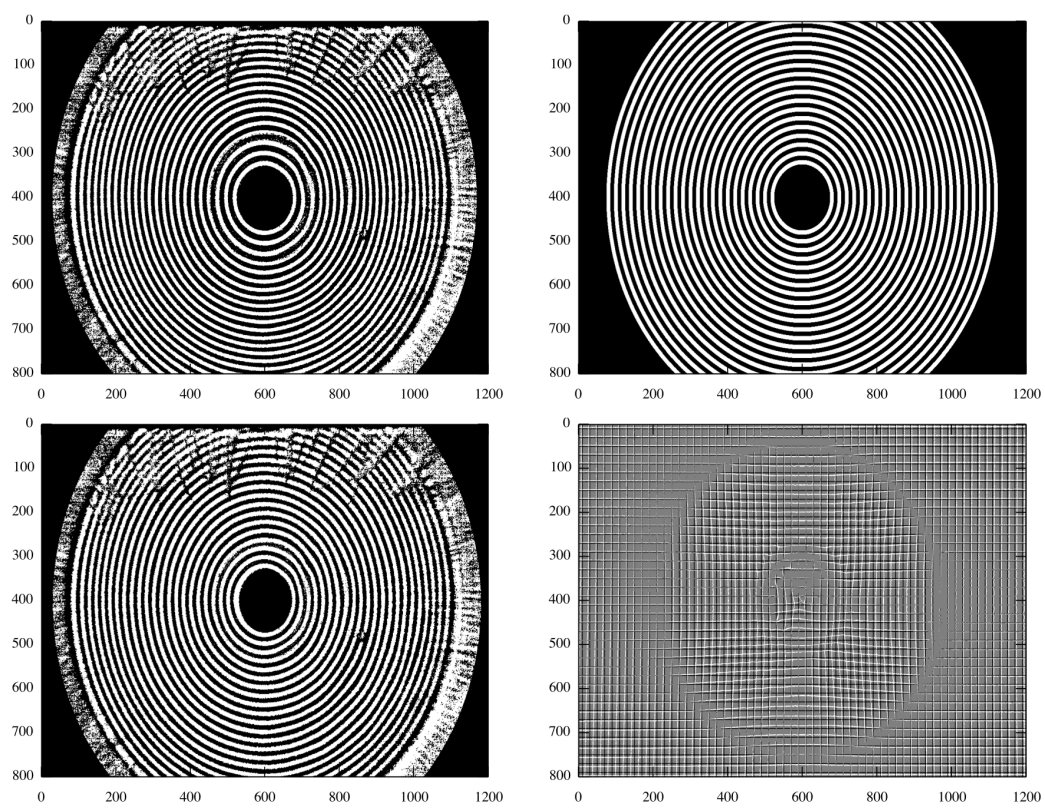


Abbildung 8: Eingabebild (o.l.), Idealmaske (o.r.), Angepasstes Bild (u.l.) und das Interferenzmuster (u.r.)

3.2.1 Kantenbild

Für die weitere Verarbeitung wollen wir ein binäres Kantenbild erzeugen, welches die Kanten der Ringe zeigt.

Aus Konsistenzgründen wollen wir zunächst einen einfachen Filterungsprozess versuchen. Es handelt sich um den Sobeloperator, bzw. die Sobelfilterung zur Kantendetektion, welche in fast jedem Buch über Bildverarbeitung¹⁶ beschrieben ist. Der Vorteil dieses Ansatzes ist die schnelle Verarbeitung und die sowohl mathematische als auch technische Simplizität. Leider reicht die Qualität des Sobelansatzes nicht für unsere Zwecke aus, wie in Abbildung 9 zu sehen ist.

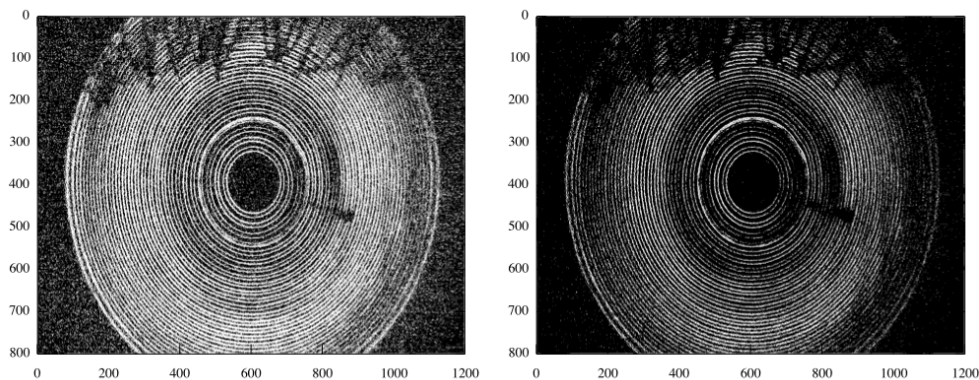


Abbildung 9: Ergebnis des Sobeloperators: Bild f mit Threshold $E(f)$ (links) und mit Threshold $E(f) + \sqrt{V(f)}$ ($E(f)$ Mittelwert, $V(f)$ Varianz.)

Der nächste Ansatz wäre der Canny Edge Detector¹⁷, welcher das Bild zunächst mit einem Gaußkern filtert und danach einen Kantenfilter (z.B. den Sobelfilter) anwendet. In Abschnitt 2.2.4 ist der Ansatz schon auf allgemeine Glättungsfunktionen ausgeweitet worden.

Mit $|Mf(u, s)|$ ist ein konkretes Werkzeug gegeben, mit dem sich Kanten in Bildern ermitteln lassen. Große Kanten sind lokal maximale $|Mf(u, s)|$ -Werte entlang einer Linie zu allen gegebenen Glättungsfunktionen. Das bedeutet, dass f am Punkt u eine Kante

¹⁶Nähere Informationen finden sich z.B. in [39].

¹⁷siehe bspw. [36]

besitzt, wenn $|Mf(u, s)|$ maximal in einer Umgebung in Richtung des Gradienten

$$\vec{\nabla}(f)(u)$$

und $-\vec{\nabla}(f)(u)$ ist, falls f an der Stelle u differenzierbar ist. Ist f an der Stelle u nicht stetig differenzierbar, so können wir den Gradienten aus folgendem Sachverhalt schätzen. Seien Φ_s n -mal stetig differenzierbare Glättungsfunktionen. Dann folgt

$$\begin{pmatrix} W^1 f(u, s) \\ W^2 f(u, s) \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial u_1} (f * \Phi_s)(u) \\ \frac{\partial}{\partial u_2} (f * \Phi_s)(u) \end{pmatrix} = \vec{\nabla}(f * \Phi_s)(u)$$

Demnach muss bei jedem Punkt u in jeder Glättungsklasse s (im Folgenden auch als Glättungsskala bezeichnet) geprüft werden, ob es sich um ein lokales Maximum bezüglich Gradientenrichtung handelt.

In der Praxis reichen diese punktwisen Informationen allerdings nicht aus, um die gesuchten Kanten zu ermitteln, da noch zu viele fehlerhafte Kanten vorhanden sind (dies kommt durch ungleichmäßige Beleuchtung, Unregelmäßigkeiten in der Iris, etc.). Daher suchen wir lokal maximale Punkte bzgl. einer Umgebung, so dass sich in einer anderen Umgebung des Punktes mindestens ein weiterer lokal maximaler Punkt befindet.

Genauer werden stückweise Kreisstücke als lokale Maxima ermittelt, die sich mit zwei Pixeln darstellen lassen. Es ist offensichtlich, dass es sich hierbei nicht nur um stückweise diskrete Kreisstücke handelt, sondern um beliebige linienartige Strukturen.

Beispiel 3.1 Um ein diagonales Kreisstück aus zwei Pixeln wie in Abbildung 10 zu ermitteln, müssten zunächst anhand der Winkel $Af(x_1, s)$ die Pixel a_1, a_2, b_1, b_2 ermittelt werden, damit die vier Vergleiche berechnet werden könnten. Außerdem muss für jeden Punkt u mit Winkel $Af(u, s)$ das Verfahren für den Fall $u = x_1$ und $u = x_2$ angewendet werden.

	a_1		
a_2		x_1	
	x_2		b_1
		b_2	

Abbildung 10: Eine schematische Darstellung der relevanten Umgebung einer Diagonalen eines Pixel x_1 in einem Bild

Dadurch werden einerseits wesentlich bessere Ergebnisse erzielt, andererseits erhöht sich die Rechenzeit durch die vielen Vergleichsoperationen unverhältnismäßig. Dieser Ansatz wird mit pre7 in Tabelle 6 bezeichnet. Da die Rechenzeit noch immer nicht für praktische Zwecke zu nutzen ist, werden die Kantenbedingungen „aufgeweicht“.

Geometrische Vereinfachung

In diesem Abschnitt wird eine Beschleunigung des Ansatzes zur Ermittlung der Kanten besprochen. Dabei betrachten wir diskrete Daten $f \in l_2(\mathbb{Z})$ und Bilder $f \in l_2(\mathbb{Z}^2)$. Um diskrete und kontinuierliche Signale zu unterscheiden, bezeichnen wir diskrete Daten mit eckiger Klammer $f[n]$ und Kontinuierliche mit $f(x)$. Die Definition eines lokalen Maximums geben wir folgendermaßen an:

Definition 3.3 (echtes lokales Maximum) $f[\alpha]$ ist echtes lokales Maximum bzgl. einer diskreten Umgebung U , falls $f[\alpha] > f[\alpha + u]$, $\forall u \in U$.

Für unsere Zwecke ergibt sich die Umgebung aus der Richtung $Af[\alpha, s]$.

Definition 3.4 Mf ist an der Stelle α (in der Skala s) lokal (betrags-) maximal (bzgl. $Af[\alpha, s]$), falls¹⁸ $|Mf[\alpha, s]| > |Mf[\alpha + \lceil Af[\alpha, s] \rceil, s]|$ und $|Mf[\alpha, s]| > |Mf[\alpha - \lfloor Af[\alpha, s] \rfloor, s]|$.

Wir betrachten nun folgenden Zusammenhang

$$a > b \Leftrightarrow a - b > 0 \Leftrightarrow \text{sign}(a - b) = 1,$$

¹⁸Mit $\lceil \cdot \rceil$ ist die Aufrundung gemeint.

wobei $sign$ die Signumfunktion

$$sign(x) = \begin{cases} 1, & \text{falls } x > 0 \\ 0, & \text{falls } x = 0 \\ -1, & \text{falls } x < 0 \end{cases}$$

bezeichnet. Das Vorzeichen wird in den meisten Zahlendatentypen durch ein einzelnes Bit codiert, wodurch sich die Signumfunktion für $a \neq b$ schnell auslesen lässt. Es wird nun versucht, in einer Achter-Nachbarschaft zu ermitteln, ob sich ein Pixel α auf einer Kreislinie befindet. Die Achter-Nachbarschaft ist ein Begriff aus der Bildverarbeitung, welcher alle diagonalen, vertikalen und horizontalen Nachbarpixel betrachtet. Mathematisch gesehen kann die Tschebyscheff-Norm zu Grunde gelegt werden und für $\alpha \in \mathbb{Z}^2$ die Punkte mit

$$\|\alpha - \beta\|_\infty = 1$$

ermittelt werden.

Da die Kreislinien isoliert sind, können wir davon ausgehen, dass sich maximal zwei weitere lokal maximale Punkte in der Achter-Nachbarschaft von α befinden. Für die restlichen sechs Punkte β_j gilt die Bedingung $\alpha > \beta_j$, bzw. $sign(\alpha - \beta_j) = 1$. Dann gibt es $\binom{8}{2} = 28$ Möglichkeiten der Verteilung der beiden lokal maximalen Punkte auf die acht Punkte der Nachbarschaft von α . Für diese beiden Punkte γ_k ist es irrelevant, ob die Bedingung $\alpha > \gamma_k$ gilt, daher gibt es noch weitere $\binom{8}{1} + \binom{8}{0} = 9$ Möglichkeiten. Sei zunächst für $f \in \mathbb{Z}^2$ die diskrete Transformation definiert als

$$Wf^k[\alpha, s] = \sum_{m \in \mathbb{Z}^2} f[m] \psi_s^k[\alpha - m] = (\widehat{f} \widehat{\psi}_s^k)^\vee[\alpha]$$

und

$$|Mf[\alpha, s]| := \sqrt{|W^1 f[\alpha, s]|^2 + |W^2 f[\alpha, s]|^2},$$

für $\alpha \in \mathbb{Z}^2, s \in \mathbb{R}^+$. Nun gilt für

$$B(\alpha) = \sum_{\zeta} sign(|Mf[\alpha, s]| - |Mf[\alpha + \zeta, s]|), \quad \zeta \in \{-1, 0, 1\}^2,$$

wobei $K(\alpha) = |Mf[\alpha, s]|, \alpha \in \mathbb{Z}^2$ ein Kantenbild darstellt.

Proposition 3.1 Ein Punkt α gehört genau dann zu der gesuchten Menge der Verteilungsmöglichkeiten, falls

$$B(\alpha) > 3.$$

Beweis: α gehört genau dann zu der gesuchten Menge, falls $\text{sign}(\alpha - \beta_j) = 1$, für $j = 1 \dots 6$ und für zwei Punkte γ_k

$$\text{sign}(\alpha - \gamma_k) = \begin{cases} 1, & \text{falls } \alpha > \gamma_k \\ 0, & \text{falls } \alpha = \gamma_k \\ -1, & \text{falls } \alpha < \gamma_k \end{cases}$$

gilt. Dann ist

$$B(\alpha) \subset \{6 - 2, \dots, 6 + 2\},$$

also $B > 3$. □

Bemerkung 3.1 Zur Erklärung der Proposition sei noch folgendes bemerkt: Sei ein echtes lokales Maximum bzgl. der Achter-Nachbarschaft an der Stelle $\alpha \in \mathbb{Z}^2$ gesucht. Das bedeutet, dass $|Mf[\alpha, s]|$ echt größer als alle umgebenden Punkte sein muss und somit wäre $B(\alpha) = 8$. Man beachte hier den wichtigen Unterscheid zwischen echt größer und größer¹⁹. Zwei in der Umgebung befindliche Werte können allerdings größer oder gleich $|Mf[\alpha, s]|$ sein, wodurch die auch Werte $B(\alpha) \in [4, 8]$ möglich werden.

Mit dieser Überlegung müssen wir also nur noch die Summe berechnen und einen Vergleich machen.

Macht man sich die möglichen Strukturen der 37 Verteilungen der beiden Punkte auf der Achter-Nachbarschaft geometrisch klar, so fällt auf, dass einige dieser Möglichkeiten nicht unbedingt eine Linienstruktur aufweisen. Die Fehlpixel, welche dadurch erzeugt werden, verfälschen das Ergebnis allerdings nur minimal, wodurch die performantere Umsetzung gegenüber der genauen Umsetzung den Vorzug hat. Dieser Ansatz wird mit pre8 in Tabelle 6 bezeichnet.

Schnelle Kantenberechnung

Um die vorhandene Datenmenge zu verarbeiten, nutzt man die schnelle Fouriertrans-

¹⁹Wird nur größer gefordert, wäre sogar $B(\alpha) = 0$ möglich.

formation. Um $|Mf[u, s]|$ zu berechnen, müssen die

$$Wf^k[\alpha, s] = (\widehat{f\psi_s^k})^\vee[\alpha]$$

mittels sechs Transformationen und einer komplexen punktweisen Matrixmultiplikation berechnet werden. Mit $f, \psi_s^k \in l_2(\mathbb{Z}^2)$ gilt nun folgender Satz, der zeigt, dass die Berechnung auch mit drei Transformationen zu meistern ist.

Satz 3.2

$$|Mf[\alpha, s]| = \left| \sum_{m \in \mathbb{Z}^2} f[m] \Psi_s[\alpha - m] \right| = |f * \Psi_s|,$$

mit $\Psi_s = (\psi_s^1 + i\psi_s^2)$

Beweis:

$$\begin{aligned} |Mf[\alpha, s]| &= \sqrt{W^1 f[\alpha, s]^2 + W^2 f[\alpha, s]^2} \\ &= |W^1 f[\alpha, s] + iW^2 f[\alpha, s]| \\ &= |(f * \psi_s^1[\alpha]) + i(f * \psi_s^2[\alpha])| \\ &= |f * (\psi_s^1 + i\psi_s^2)[\alpha]| \\ &= |f * \Psi_s[\alpha]|. \end{aligned}$$

Aus

$$\begin{aligned} f * \Psi_s[\alpha] &= (\widehat{f\Psi_s})^\vee[\alpha] \\ &= (\widehat{f\psi_s^1} + i\widehat{f\psi_s^2})^\vee[\alpha] \end{aligned}$$

folgt nun die schnelle Verarbeitung. □

Da die schnelle Fouriertransformation die reellen Daten in die komplexe Ebene transformiert, wird der Rechenaufwand durch dieses Verfahren nicht erhöht. Allerdings spart man einen Filterungsprozess, wodurch sich der Aufwand bei dieser Menge an Daten halbiert. Dieser Ansatz wird mit pre9 in Tabelle 6 bezeichnet.

Damit haben wir eine effektive Möglichkeit gefunden, Kanten zu klassifizieren. Diese Klassifikation wird nun mittels verschiedener Glättungsfunktionen durchgeführt und aufaddiert. Ist diese Summe an einem Punkt u größer als eine vorgegebene Toleranzschranke, so ist an diesem Punkt eine Kante. Die gefundenen Kanten sind in Abbildung

11 links zu sehen. Man kann erkennen, dass der Algorithmus alle gewünschten Kanten ermittelt. Allerdings sind auch Kanten vorhanden, die nicht gesucht sind. Diese lassen sich nun herausfiltern, indem eine Maske generiert wird. Dazu wird für $|Mf[u, s]|$ eine Skala s gewählt und mittels Threshold in ein Binärbild unterteilt. Eine solche Maske ist in Abbildung 11 rechts zu sehen. Durch punktweise Multiplikation werden die effektiv gesuchten Kanten ermittelt, welche in Abbildung (12) zu sehen sind.

Die in der Tabelle 6 genannten Daten wurden in Octave (inkl. ATLAS²⁰ und FFTW²¹) berechnet und dienen nur zum Vergleich der Verbesserungen mittels theoretischer Optimierung. Bei der Implementierung in C++ wurden Werte um 3 Sekunden pro Bild erzielt. Da es sich hauptsächlich um Faltungen handelt, würde eine Berechnung auf einem Grafikprozessor vermutlich im Millisekundenbereich liegen.

	1 Bild	4 Bilder	8 Bilder	16 Bilder
pre7	18	76	164	303
pre8	7	27	55	111
pre9	8	20	35	66

Tabelle 6: Rechenzeit der Algorithmen in Sekunden. Die Größe der Bilder beträgt 800x1200 Pixel (d.h. bei 16 Bildern müssen etwa 44 MB Daten verarbeitet werden).

²⁰Automatically Tuned Linear Algebra Software. <http://math-atlas.sourceforge.net>

²¹Fastest Fourier Transform in the West. <http://www.fftw.org>

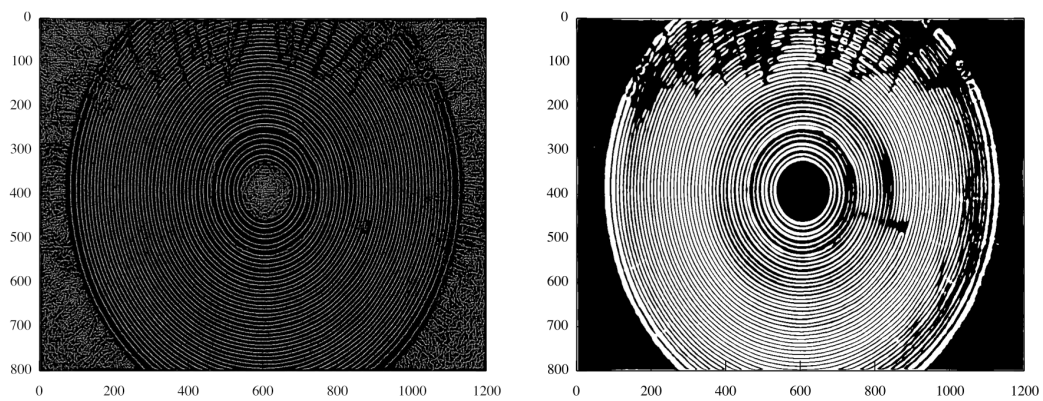


Abbildung 11: Berechnetes Kantenbild mit Algorithmus pre9 (links) und die berechnete Maske (rechts)

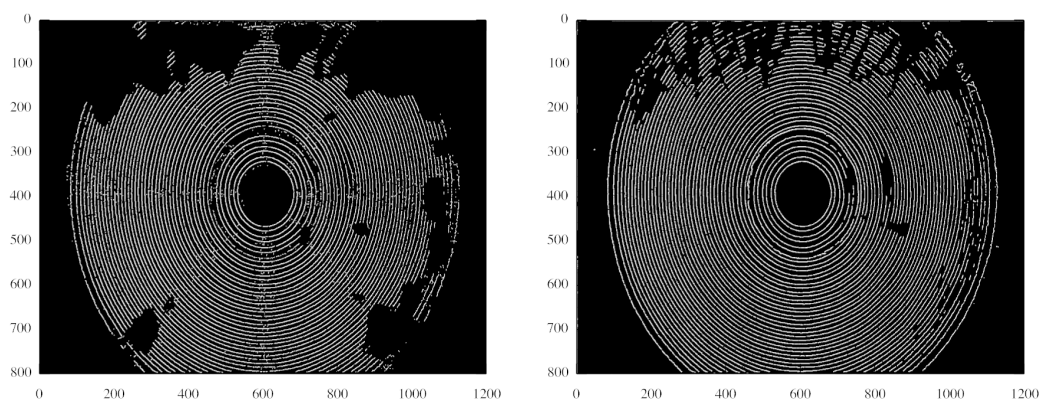


Abbildung 12: Ergebnisse der Algorithmen pre7 (links) und pre8 bzw. pre9 (rechts)

3.2.2 Mittelpunktsuche

Um den Mittelpunkt der konzentrischen Kreise zu finden, wurden diverse Verfahren getestet und entwickelt. Um die Aufgabe zu lösen, werden wir die Problemstellung spezifizieren: Es soll aus einem Binärbild der Mittelpunkt von konzentrisch angeordneten kreisförmig bis elliptischen Strukturen ermittelt werden. Dabei ist zu beachten, dass diese Strukturen in der Regel nicht geschlossen sind und die Verarbeitung möglichst schnell erfolgen soll. Zusätzlich können große Stücke in den Strukturen fehlen (diese Aufrisse werden ja später für die Analyse benötigt) und durch das schlechte Signal-Rausch-Verhältnis sind relativ viele Störpixel vorhanden. Ein Algorithmus, der unserer Ansicht nach das Problem am besten löst, wird im Folgenden beschrieben: Gegeben sei ein Bild f , welches wie in der Aufgabenstellung beschrieben konzentrische Kreise enthält. Sei Φ eine mindestens einmal stetig differenzierbare Glättungsfunktion und seien ψ_1, ψ_2 definiert als:

$$\psi^1 := \frac{\partial \Phi}{\partial x_1}, \quad \psi^2 := \frac{\partial \Phi}{\partial x_2}.$$

In den beiden durch $1 \leq k \leq 2$ indizierten Richtungen ergibt sich für f an der Stelle $u = u_1, u_2$ als

$$W^k f(u, s) = f * \psi_s^k(u).$$

Für verschiedene Glättungsfunktionen Φ_s folgt die analoge Definition

$$\psi_s^k = \frac{\partial \Phi_s}{\partial x_k}.$$

Der folgende Zusammenhang zeigt, dass die Komponenten proportional zum Gradienten der durch Φ_s geglätteten Funktion f ist:

$$\begin{pmatrix} W^1 f(u, s) \\ W^2 f(u, s) \end{pmatrix} = \vec{\nabla}(f * \bar{\Phi}_s)(u).$$

Zu Erinnerung wiederholen wir die Definitionen aus 2.2.4:

$$Mf(u, s) = \sqrt{|W^1 f(u, s)|^2 + |W^2 f(u, s)|^2}$$

und

$$Af(u, s) = \begin{cases} \alpha(u) & , \text{ falls } W^1 f(u, s) \geq 0 \\ \pi - \alpha(u) & , \text{ falls } W^1 f(u, s) < 0 \end{cases}$$

wobei $\alpha(u) = \tan^{-1}\left(\frac{W^2 f(u, s)}{W^1 f(u, s)}\right)$ ist. Der Einheitsvektor $\vec{n}_s(u) = (\cos(Af(u, s)), \sin(Af(u, s)))$ ist kollinear zu $\vec{\nabla}(f * \Phi_s)(u)$. Ein Kantenpunkt in der Glättungsskala s sei nun ein Punkt v , so dass $Mf(u, s)$ ein lokales Maximum an der Stelle v mit $u = v + \lambda \vec{n}_s(v)$, für $|\lambda|$ klein genug.²² Diese Kantenpunkte sind meist auf Kurven verteilt, welche auf Bildern die Ränder von Strukturen und Objekten markieren. Diese Kurven können durch Verkettung von einzelnen lokal (betrags-) maximalen Punkten erzeugt werden. An jeder beliebigen Stelle kann die Tangente der Kantenkurve durch die Berechnung der Tangenten der Levelmenge approximiert werden. Diese Tangentenrichtung kann man dann zur Verkettung der Kantenpunkte, die auf einer Kantenkurve liegen, nutzen.

In Abschnitt 2.2.4 haben wir den Zusammenhang der Levelmenge bei lokal konstanten Strukturen betrachtet. Für unsere Anwendung sind diese konstanten Strukturen, deren Gradient ein lokales Extremum bzgl. dem Gradientenvektor bildet, auf den Rändern der konzentrischen Ringe. Sei also $p = (p_x, p_y)$ ein Kantenpunkt auf einem der konzentrischen Ringe, welcher auf allen Glättungsskalen $s \in L$ ein lokales Maximum $Mf(p, s)$ bzgl. der Geraden $p + \lambda \vec{n}_L(p)$ aufweist²³. Da nach Voraussetzung dieser Kantenpunkt auf einem Kreis um einen Mittelpunkt $m = (m_x, m_y)$ mit Radius r liegt, folgt, dass die Gerade $p + \lambda \vec{n}_s(p)$ den Mittelpunkt m schneidet, für ein $\lambda \in \mathbb{R}$, also

$$\begin{pmatrix} m_x \\ m_y \end{pmatrix} = \begin{pmatrix} p_{1x} + \lambda \cos(Af(u, s)) \\ p_{1y} + \lambda \sin(Af(u, s)) \end{pmatrix}.$$

In diesem Fall ist es geometrisch auch sofort einzusehen, denn die Tangente in einem Punkt p eines Kreises steht immer senkrecht auf der Geraden, die durch den Mittelpunkt des Kreises und den Punkt p geht.

Diese Eigenschaft haben wir uns nun zur Problemlösung zunutze gemacht. Angenommen, alle konzentrischen Kreise in einem Bild haben einen gemeinsamen Mittelpunkt, dann schneiden sich alle Geraden $v + \lambda \vec{n}_L(v)$ (für wie oben beschriebene Punkte v) im Mittelpunkt. Man berechnet ein binäres Kantenbild, indem die Punkte gleich Eins

²²In [20] werden solche Punkte für Wavelets als *wavelet transform modulus maxima* bezeichnet.

²³zu beachten ist, dass sich der Winkel $Af(p, s)$ für $s \in L$ nur minimal ändern sollte. Ebenso sollte der Approximant in der Glättungsskala L den theoretischen Winkel möglichst genau annähern

gesetzt werden, welche auf allen Skalen $s \leq L$ ein lokales Maximum bzgl. des oben beschriebenen Verfahrens aufweisen. Danach werden M Punkte aus dem Kantenbild verwendet, um paarweise die $\binom{M}{2}$ Schnittpunkte zu berechnen. Da diese Schnittpunkte in der Praxis durch multiple Faktoren vom Mittelpunkt abweichen können, wird als approximierter Mittelpunkt der Median der Schnittpunkte berechnet. Durch N -malige Anwendung dieses Verfahrens und die Mittelwertberechnung dieser Werte, kann die Genauigkeit des Verfahrens noch erhöht werden.

Das folgende Schaubild 14 zeigt einen Berechnungsschritt dieses Verfahrens. Seien also zwei Punkte $p_k = \begin{pmatrix} p_{kx} \\ p_{ky} \end{pmatrix}$ mit Normalenvektoren $q_k = \begin{pmatrix} q_{kx} \\ q_{ky} \end{pmatrix} = \begin{pmatrix} \cos(\gamma_k) \\ \sin(\gamma_k) \end{pmatrix}$ für $k = 1, 2$. Die Punkte p_k ergeben sich für eine Glättungsfunktionenmenge S an den Stellen, an denen $Mf(p_k, s)$ ein betragsmaximaler Punkt ist, also ein lokales Extremum bzgl. der Richtungen $Af(p_k, s)$, $s \in S$. Die Winkel m_k der Normalenvektoren ergeben sich aus $\gamma_k = Af(p_k, s^*)$ für eine Skala s^* . Damit erhalten wir die Gleichungen für die Normalen an den Punkten p_k als

$$G_k(t) = p_k + tq_k.$$

Der Schnittpunkt der beiden Geraden ergibt sich also aus

$$\begin{aligned} G_1(t) = G_2(t) &\Leftrightarrow p_1 + tq_1 = p_2 + tq_2 \\ &\Leftrightarrow p_{1x} + t_1q_{1x} = p_{2x} + t_2q_{2x} \\ &\Leftrightarrow p_{1y} + t_1q_{1y} = p_{2y} + t_2q_{2y} \\ &\Leftrightarrow q_{1x}t_1 - q_{2x}t_2 = p_{2x} - p_{1x} \\ &\Leftrightarrow q_{1y}t_1 - q_{2y}t_2 = p_{2y} - p_{1y} \end{aligned}$$

welches sich als lineares Gleichungssystem

$$\begin{pmatrix} q_{2x} & -q_{1x} \\ q_{2y} & -q_{1y} \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} p_{1x} - p_{2x} \\ p_{1y} - p_{2y} \end{pmatrix}$$

schreiben lässt. Falls das LGS lösbar ist, ist der Schnittpunkt

$$\begin{pmatrix} m_x \\ m_y \end{pmatrix} = \begin{pmatrix} p_{1x} + q_{1x}t_2 \\ p_{1y} + q_{1y}t_2 \end{pmatrix} = \begin{pmatrix} p_{1x} + \cos(\gamma_1)t_2 \\ p_{1y} + \sin(\gamma_1)t_2 \end{pmatrix} \quad (2)$$

der geschätzte Mittelpunkt. Schreibt man

$$A = \begin{pmatrix} q_{2x} & -q_{1x} \\ q_{2y} & -q_{1y} \end{pmatrix} = \begin{pmatrix} \cos(\gamma_2) & -\cos(\gamma_1) \\ \sin(\gamma_2) & -\sin(\gamma_1) \end{pmatrix}$$

so ist mit $D := \cos(\gamma_1) \sin(\gamma_2) - \sin(\gamma_1) \cos(\gamma_2)$

$$A^{-1} = \begin{pmatrix} \frac{-\sin(\gamma_1)}{D} & \frac{\cos(\gamma_1)}{D} \\ \frac{-\sin(\gamma_2)}{D} & \frac{\cos(\gamma_2)}{D} \end{pmatrix},$$

falls

$$D \neq 0.$$

Dann ist

$$t_2 = \frac{-\sin(\gamma_2)}{D}(p_{1x} - p_{2x}) + \frac{\cos(\gamma_2)}{D}(p_{1y} - p_{2y}),$$

und der Mittelpunkt lässt sich ohne lösen eines LGS direkt angeben, denn setzt man t_2 in Gleichung (2) ein, so erhalten wir

$$\begin{pmatrix} m_x \\ m_y \end{pmatrix} = \begin{pmatrix} p_{1x} + \cos(\gamma_1) \left(\frac{-\sin(\gamma_2)}{D}(p_{1x} - p_{2x}) + \frac{\cos(\gamma_2)}{D}(p_{1y} - p_{2y}) \right) \\ p_{1y} + \sin(\gamma_1) \left(\frac{-\sin(\gamma_2)}{D}(p_{1x} - p_{2x}) + \frac{\cos(\gamma_2)}{D}(p_{1y} - p_{2y}) \right) \end{pmatrix}. \quad (3)$$

Auf der folgenden Seite ist der Algorithmus in kompakter Form abgebildet.

Algorithmus: Center Detection

0. Gegeben: Punkte $p_j, j = 1, \dots, n$ aus dem binären Kantenbild, q_j aus dem Winkelbild und $N, M \in \mathbb{N}$.
1. Berechne für eine (zufällige) M -elementige Teilmenge der n Punkte paarweise $\begin{pmatrix} m_x \\ m_y \end{pmatrix}$ mittels Gleichung (3).
2. Bestimme den Median der $\binom{M}{2}$ Mittelpunktschätzungen aus Schritt 1.
3. Wiederhole ab Schritt 1. N mal.
4. Berechne den Mittelwert der N Mittelpunktschätzungen aus Schritt 2.

Beispiel 3.2 *Um den Mittelpunkt eines Objektes oder eines Bildes zu ermitteln, wird in der Bildverarbeitung oft der Schwerpunkt eines Bildes genutzt.*

Der Schwerpunkt $\begin{pmatrix} s_x \\ s_y \end{pmatrix}$ eines Bildes f wird als

$$s_x = \frac{\sum_{x,y} x f[x,y]}{\sum_{x,y} f[x,y]}, \quad s_y = \frac{\sum_{x,y} y f[x,y]}{\sum_{x,y} f[x,y]}$$

*definiert. Die obere Reihe von Abbildung 13 zeigt Ergebnisse der Schwerpunktbe-
rechnung eines Grauwertbildes (links) und des Binärbildes nach der Kantenfilterung
(rechts).*

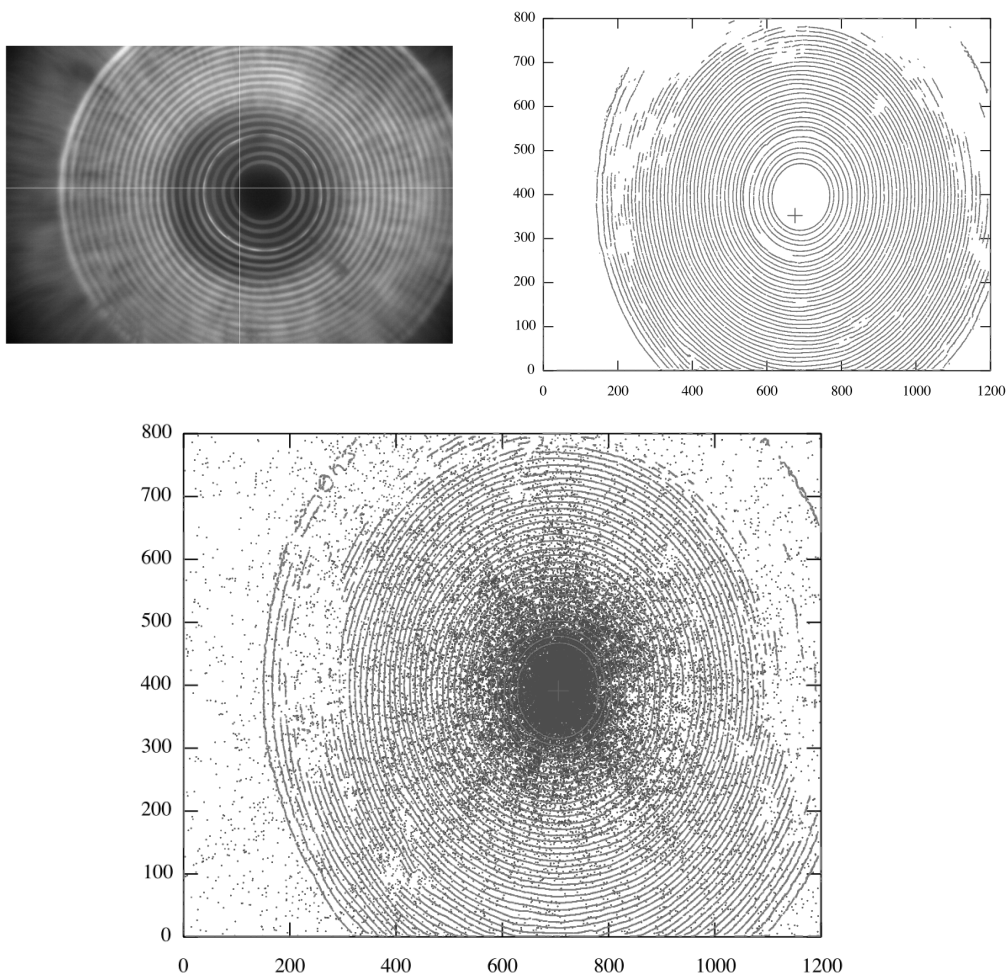


Abbildung 13: Vergleich von Schwerpunkt eines Grauwertbildes (oben links), Schwerpunkt eines Binärbildes (oben rechts) und dem beschriebenen Algorithmus zur Mittelpunktssuche (unten). Die blauen Punkte im unteren Bild sind die Ergebnisse der Schnittpunktberechnungen von Schritt 1 im Center Detection Algorithmus.

Um den Ansatz auch heuristisch zu belegen, wurde die Mittelpunkte auf 453 Bildern²⁴ bestimmt. Anschließend wurde mit dem beschriebenen Center Detector der Mittelpunkt auf den Bildern ermittelt. Dabei wurde die Abweichung der manuell bestimmten Mittelpunkte mit den Punkten des Center Detectors in der 2-Norm berechnet²⁵. Um die Reproduzierbarkeit der Ergebnisse zu gewährleisten, wurde der Center Detector Test sechs mal ausgeführt. In Tabelle 7 sind die Ergebnisse zu sehen. Es zeigt, dass die mittlere Abweichung der manuellen Mittelpunkte von den Mittelpunkten des

²⁴Größe der Bilder: 800×1200 Pixel

²⁵Zur Erklärung ist beispielsweise die Abweichung um einen diagonalen Pixel $\sqrt{2}$.

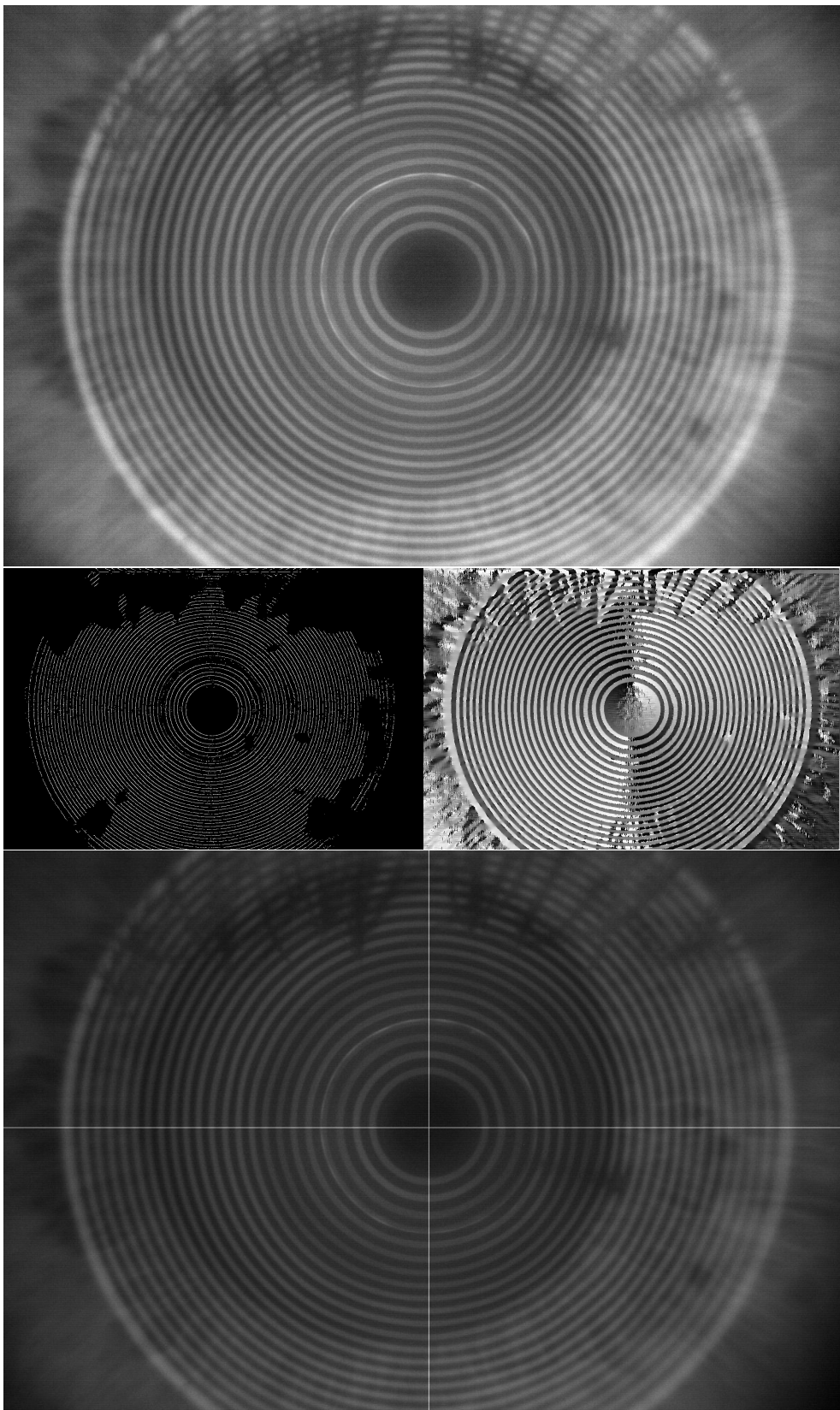


Abbildung 14: Ergebnis der Mittelpunktssuche: Ausgangsbild (oben), berechnete Kanten (mitte links) und Winkel (mitte rechts) mit dem beschriebenen Algorithmus und der berechnete Mittelpunkt (unten)

Center Detectors im Mittel um weniger als einen Pixel abweichen. Die maximale Abweichung in allen $6 \cdot 453 = 2718$ Testläufen betrug 3.5565 Pixel.

Zum Vergleich sind die Ergebnisse der Schwerpunktberechnung der 453 binären Kantenbilder in Tabelle 8 aufgeführt. Im Mittel weicht der gefundene Schwerpunkt um 51.534 ± 25.1 Pixel von dem manuellen Mittelpunkt ab.

Test Nr.	Mittelwert	Standardabweichung	Maximale Abweichung
1	0.783626	0.359777	1.726799
2	0.824529	0.374835	1.829870
3	0.843783	0.381474	3.556582
4	0.837671	0.361265	2.331034
5	0.824812	0.372022	2.511475
6	0.836818	0.363766	2.094502
gesamt	0.82521	0.36886	3.556582

Tabelle 7: Center Detector Algorithmus. Bei 453 Bildern wurde der Mittelpunkt berechnet und die Abweichung von dem manuell ermittelten Mittelpunkt in der 2-Norm berechnet.

Test Nr.	Mittelwert	Standardabweichung	Maximale Abweichung
1	51.533970	25.099596	118.472549

Tabelle 8: Schwerpunktberechnung der 453 binären Kantenbilder.

Bemerkung 3.2 Dieser Ansatz zur Mittelpunktschätzung hat nun folgende Vorteile:

1. Die Verarbeitung ist robust gegenüber fehlenden, unterbrochenen, lediglich stückweise vorkommenden und deformierten Kreisen.
2. Die Verarbeitung ist echtzeitfähig.

Deformierte Kreise in Parameterform

Dieser Abschnitt beschäftigt sich mit deformierten Kreisen in Parameterform, die wir als grundlegendes Modell für die Kanten der Ringe auf den Bildern nutzen möchten. Dazu definieren wir zunächst trigonometrische Polynome:

Definition 3.5 (trigonometrisches Polynom) Eine Funktion wird als (reelles) trigonometrisches Polynom vom Grad n bezeichnet, falls sie sich in der Form

$$T_n(\varphi) = \frac{a_0}{2} + \sum_{j=1}^n a_j \cos(j\varphi) + b_j \sin(j\varphi)$$

mit $\varphi, a_j, b_j \in \mathbb{R}$ darstellen lässt.

Definition 3.6 (komplexes trigonometrisches Polynom) Eine Funktion wird als komplexes trigonometrisches Polynom vom Grad n bezeichnet, falls sie sich in der Form

$$T_n(\varphi) = \sum_{j=-n}^n c_j e^{ij\varphi}, \varphi \in \mathbb{T}.$$

mit $x \in \mathbb{R}, c_j \in \mathbb{C}, j = -n, \dots, n$ darstellen lässt.

Bemerkung 3.3 An dieser Stelle möchten wir noch einige Eigenschaften von trigonometrischen Polynomen angeben.

1. Die trigonometrischen Polynome sind 2π -periodisch.
2. Ein komplexes trigonometrisches Polynom ist genau dann reell, falls $c_j = \overline{c_{-j}}$.
3. Für reelle trigonometrische Polynome gilt $c_0 = \frac{a_0}{2}, c_j = \frac{a_j - ib_j}{2}, c_{-j} = \frac{a_j + ib_j}{2}, j = 1, \dots, n$.
4. Umgedreht erhält man $a_0 = 2c_0, a_j = 2\operatorname{Re}(c_j) = c_j + c_{-j}, b_j = -2\operatorname{Im}(c_j) = i(c_j - c_{-j})$,

da

$$c_j = \frac{2\operatorname{Re}(c_j)}{2} - \frac{i(-2\operatorname{Im}(c_j))}{2} = \frac{a_j - ib_j}{2}$$

gilt.

Im Allgemeinen gehen wir in dieser Arbeit von reellen trigonometrischen Polynomen aus, die wir allerdings in der kompakteren komplexen Form schreiben.

Definition 3.7 (Deformierter Kreis) Wir definieren einen deformierten Kreis um den Ursprung mit einem reellen trigonometrischem Polynom $T = T_n$ mit Grad n als

$$K_T(\varphi) = \begin{pmatrix} K_{T1}(\varphi) \\ K_{T2}(\varphi) \end{pmatrix} = \begin{pmatrix} T_n(\varphi) \cos(\varphi) \\ T_n(\varphi) \sin(\varphi) \end{pmatrix}$$

mit $\varphi \in [0, 2\pi)$.

Damit haben wir den Kreis als Parameterform. Diesen können wir auch in der komplexen Ebene als

$$T_n(\varphi) \cos(\varphi) + iT_n(\varphi) \sin(\varphi)$$

ausdrücken. O.B.d.A. werden wir diese beiden Darstellungsarten simultan nutzen und beides mit K_p bezeichnen.

3.2.3 Polartransformation

Zu Analyse Zwecken ist es nützlich, das Bild I bezüglich des Mittelpunktes $m = (m_x, m_y)$ (siehe oben) in Polarkoordinaten zu transformieren. Prinzipiell nutzen wir hierfür die Formel:

$$I_p(r, \varphi) = I(m_x + r \cos(\varphi), m_y + r \sin(\varphi)),$$

wobei $r \in \mathbb{R}_0^+$ und $\varphi \in [0, 2\pi)$ ist. Praktisch²⁶ muss nun r weiter eingeschränkt und P_I diskretisiert werden. Wir haben uns hier aufgrund der besseren Performance für die *Nearest Neighbor Methode* entschieden, die man durch folgende Formel beschreiben kann:

$$I_p(r, \phi) = I([\![m_x + r \cos(\varphi)\!], [m_y + r \sin(\varphi)]],$$

wobei $[\cdot]$ die Ganzzahlrundung darstellt. Andere Möglichkeiten sind die diversen Interpolationsverfahren (wie beispielsweise lineare Interpolation²⁷), die hier aber nicht weiter diskutiert werden.

Prinzipiell ist nun noch die Frage zu klären, wie r und φ beschränkt bzw. diskretisiert werden. Dazu sei der Träger des Bildes als $\text{supp}(I) = [0, M] \times [0, N]$ gegeben. Dann

²⁶Da ein Rechner leider nur endlich viel Speicherplatz zur Verfügung hat.

²⁷Bei Bildern entspricht die (Bi-)Lineare Interpolation einer linearen Interpolation in einer Richtung gefolgt von einer linearen Interpolation in der dazu orthogonalen Richtung.

kann man r mit

$$r \leq \sqrt{M^2 + N^2}$$

abschätzen. In der Praxis stellte sich allerdings heraus, dass es am sinnvollsten ist, r durch $\max(M/2, N/2)$ zu begrenzen. Die Diskretisierung erfolgte, indem $r \in \mathbb{Z}$ angenommen wurde.

Für φ wurden folgende Restriktionen definiert:

$$\varphi = (a/N_2 2\pi),$$

mit $a \in [0, N_2 - 1] \subset \mathbb{Z}$, $N_2 = \lceil 2\pi\sqrt{M^2 + N^2} \rceil$ und $\lceil \cdot \rceil$ als Aufrundungsfunktion. Die Praxis hat allerdings gezeigt, dass es für die Weiterverarbeitung der Daten sinnvoller ist, $N_2 = 2^L$ für ein $L \in \mathbb{Z}$ zu setzen. Die Umsetzung erfolgte zunächst als Octave-/Matlab Skript. Dies war allerdings selbst für Analysezwecke zu langsam (~ 60 Sek. pro Bild), so dass es als sogenannte oct-Datei²⁸ umgesetzt wurde ($< 0,5$ Sek. pro Bild).

²⁸Eine oct-Datei ist eine dynamische Erweiterung des Octave-Interpreter. Die Quelltexte sind in C++ verfasst

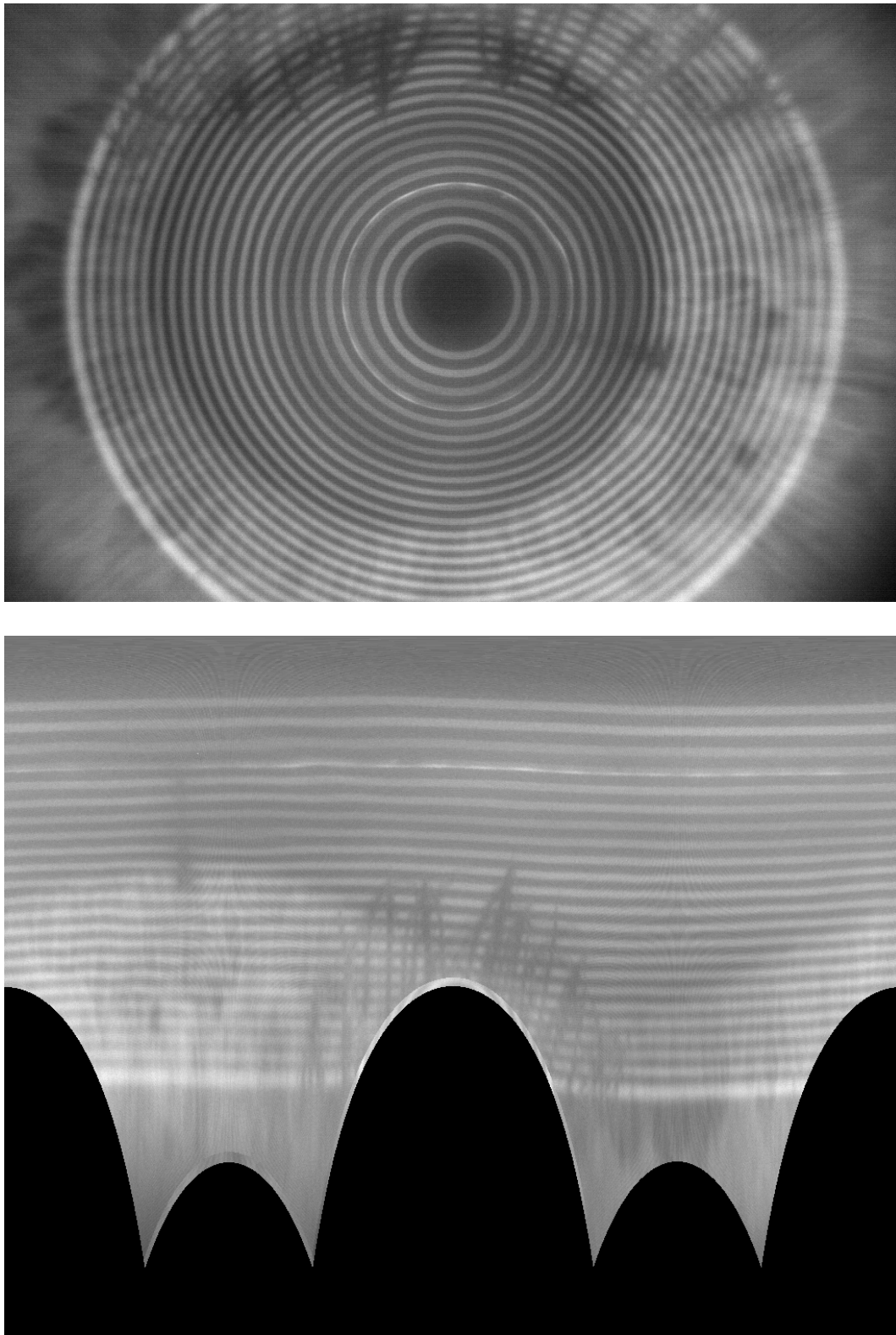


Abbildung 15: Polarkoordinatentransformation: 1. Ausgangsbild, 2. Polartransformation von 1.

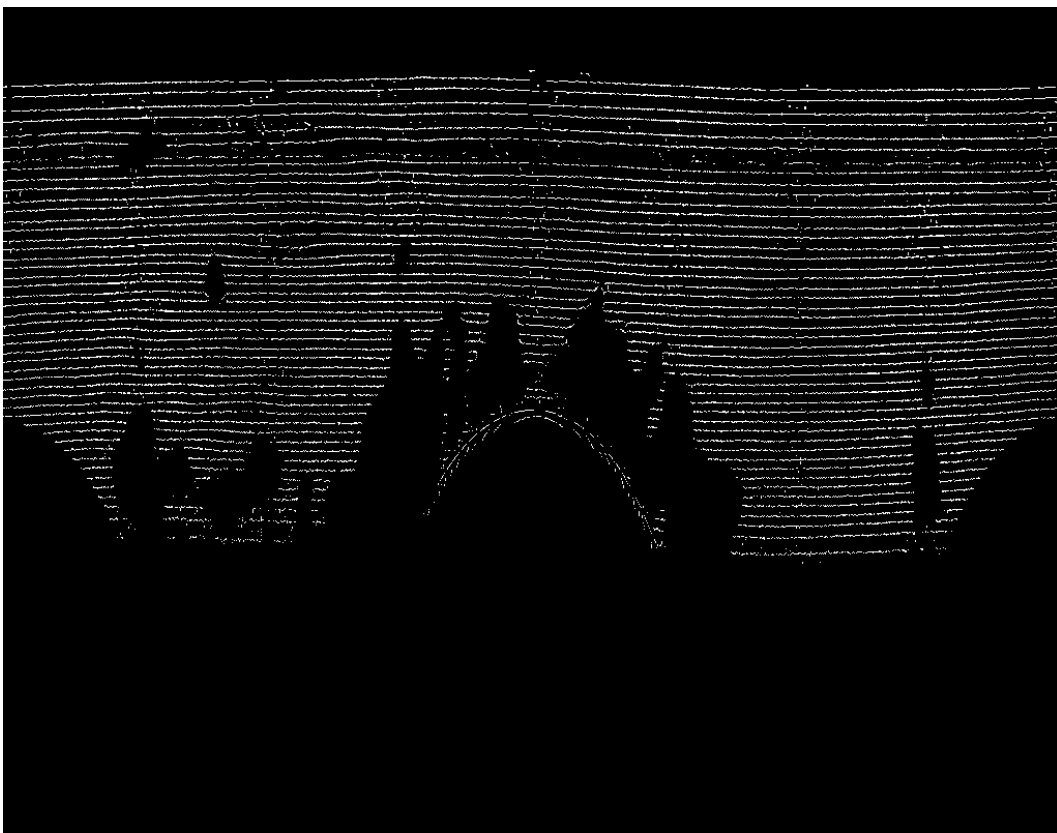
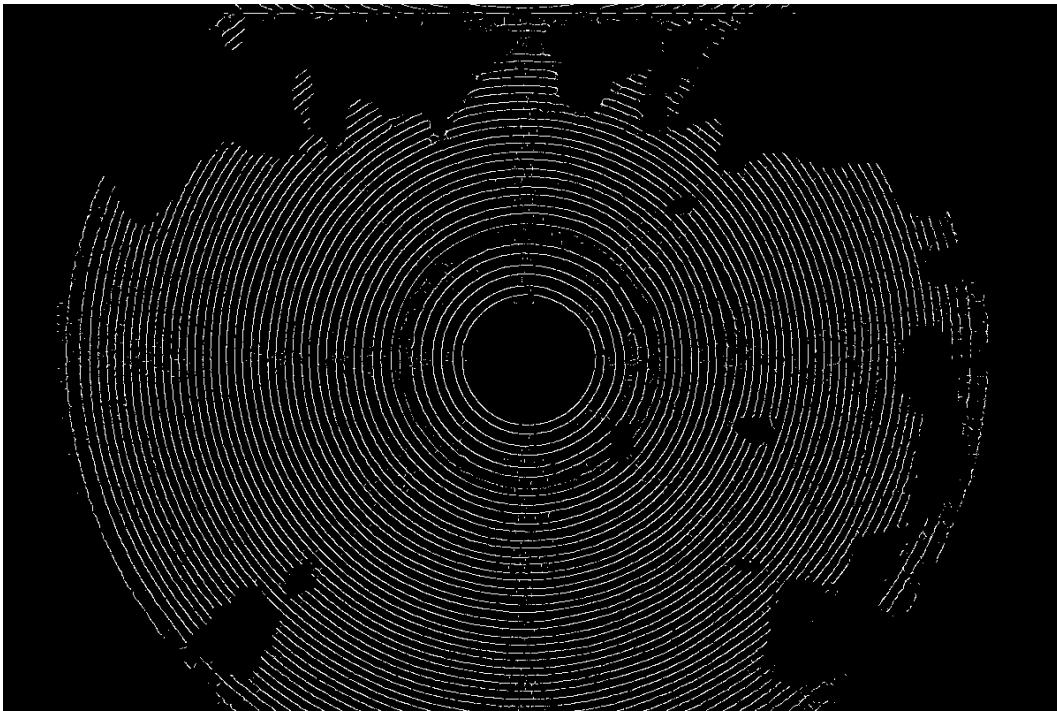


Abbildung 16: Polarkoordinatentransformation eines Kantenbildes: 1. Kantenbild, 2. Polartransformation von 1.

3.2.4 Feature Detection

An dieser Stelle möchten wir einige Vorüberlegungen anstellen, die für die spätere Verarbeitung sinnvoll erscheinen. Es sei wieder ein Bild f gegeben, wobei an den Kanten der konzentrischen Ringe Singularitäten auftreten. Wir wissen, dass $|Mf(u, s)|$ in Glättungsskalen s lokal maximal ist. Zusätzlich kennen wir den Zusammenhang

$$\begin{pmatrix} W^1 f(u, s) \\ W^2 f(u, s) \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial u_1} (f * \Phi_s)(u) \\ \frac{\partial}{\partial u_2} (f * \Phi_s)(u) \end{pmatrix} = \vec{\nabla} (f * \Phi_s)(u),$$

wodurch der Zusammenhang

$$|Mf(u, s)| = s \|\vec{\nabla} (f * \Phi_s)\|_2$$

geschaffen wird. Ein Punkt $x \in \mathbb{R}^2$ wird als Kante bezeichnet, falls der Gradient ein lokales Maximum bzgl. der Umgebung

$$x + \lambda \vec{\nabla} (f * \Phi_s),$$

mit $|\lambda| < \varepsilon$ besitzt. Außerdem ist der Einheitsvektor $\vec{n}_s(u) = \begin{pmatrix} \cos(A(f(u, s))) \\ \sin(A(f(u, s))) \end{pmatrix}$ kollinear zum Gradientenvektor $\vec{\nabla} (f * \Phi_s)(u)$. Die Levelmenge sind die Kurven $x(t)$ von $f * \Phi_s$, an denen $f * \Phi_s$ konstant ist. Sei $\vec{\tau}$ die Richtung der Tangente von $x(t)$, dann ist

$$\vec{\nabla} (f * \Phi_s) \vec{\tau} = 0.$$

Also steht der Gradient geometrisch senkrecht auf $\vec{\tau}$.

In den vorherigen Kapiteln wurde gezeigt, wie der Mittelpunkt und die Polartransformation eines Bildes angegeben werden kann. Daher nutzen wir nun für einen Pixel die Darstellungen (x_1, x_2) und (r, φ) mit $(x_1 + ix_2) = re^{i\varphi}$ simultan. Aus dem binären Kantenbild erhalten wir die Kantenpixel (x_ξ, y_ξ) , $\xi = 1, \dots, \Xi$.

Diese werden in K Klassen eingeteilt, wobei die k -te Klasse dem k -ten deformierten Kreis entsprechen soll. Anschließend sollen die Pixel aus den Klassen mittels einer geeigneten Funktionsklasse approximiert werden.

Sei eine Klasse k gegeben und (x_1, x_2) ein Pixel dieser Klasse. Wir setzen nun voraus, dass sich alle Pixel einer Klasse auf einer deformierten Kreiskurve als

$$\sum_{j=-n}^n c_j^k e^{i(j+1)\varphi} = x_1 + ix_2$$

darstellen lassen. Die Gleichung

$$\begin{aligned} \sum_{j=-n}^n c_j^k e^{i(j+1)\varphi} &= x_1 + ix_2 = re^{i\varphi} \\ &\Leftrightarrow \\ \sum_{j=-n}^n c_j^k e^{ij\varphi} &= \text{abs}(x_1 + ix_2) = r \end{aligned}$$

zeigt, dass sich die Kantenpixel als reelles trigonometrisches Polynom darstellen lassen.

Da es sich um endlich viele Punkte Klassen handelt, könnten diese eindeutig interpoliert werden. Dadurch würde allerdings der Grad der Polynome sehr groß werden, was zu numerischen Instabilitäten, langen Rechenzeiten und Oszillationen führen würde. Daher geben wir einen kleinen Grad n der Polynome vor und versuchen, die Pixel geeignet zu approximieren.²⁹

Gesucht sind also K Funktionen, die die Punkte in einer Dimension möglichst genau annähern. Genauer gesagt suchen wir die Koeffizienten c_j^k von

$$T_n^k(\varphi) = \sum_{j=-n}^n c_j^k e^{ij\varphi} = r,$$

so dass die gegebenen Punkte (r_ξ, φ_ξ) , $\xi = 1, \dots, \Xi$ von der ihr nächstgelegenen Funktion möglichst genau angenähert werden. Mathematisch betrachtet bedeutet dies, dass wir folgendes suchen

$$\arg \min_{c_{j,k}} \sum_{\xi=1}^{\Xi} \min_{k=1, \dots, K} \{ \|T_n^k(\varphi_\xi) - r_\xi\|^2 \} \quad (4)$$

²⁹Man beachte, dass die deformierten Kreise durch Setzen von $n = 0$ zu konzentrischen Kreisen ohne Deformationen werden.

Beispiel 3.3 Sei $n = 0$ und $K = 2$, dann kann man Formel (4) auch schreiben als

$$\arg \min_{c_0^k} \sum_{\xi=1}^{\Xi} \|c_0^1 - r_\xi\| + \|c_0^2 - r_\xi\| - \| \|c_0^1 - r_\xi\| - \|c_0^2 - r_\xi\| \|.$$

Dies ist damit gleichbedeutend, dass man zwei konstante Funktionen optimal durch eine Punktmenge legt. Eindeutigkeit kann man bei diesem Beispiel dadurch erlangen, dass $c_0^1 > c_0^2$ sein muss. Mit dieser Voraussetzung ist die Zielfunktion mit milden Bedingungen an die Punkte sogar konvex, allerdings ist die Zielfunktion nicht stark differenzierbar.

Nutzt man die Eigenschaft $\min(f, g) = (f + g - |f - g|)/2$, so kann die Zielfunktion als einfache Summe dargestellt werden. Zusätzlich ist es sinnvoll $T_n^k(\varphi) < T_n^{k+1}(\varphi)$ zu fordern. Mit dieser Forderung erreicht man auch direkt, dass die Funktionen sich nicht schneiden. Im Fall $n = 0$ läßt sich die Aufgabe algorithmisch mit der Abstiegsmethode lösen. Dabei ist allerdings eine Konvergenzaussage nicht gegeben. Eine effiziente Implementierung ist für größere K jedoch nur denkbar, falls $K = 2^n$, $n \in \mathbb{N}$ gilt. Für unsere Zwecke kommt ein solcher Ansatz jedoch nicht in Frage, da erstens die Laufzeit zu sehr erhöht würde und zweitens die Problemstellung noch genauer spezifiziert werden müsste, was in noch längerer Rechenzeit und komplizierteren Fallunterscheidungen münden würde.

Wir werden an dieser Stelle keinen Algorithmus für die Lösung dieser Aufgabe angeben, da sich durch die Entwicklung der Hardware³⁰ in diesem Projekt die Verarbeitungstechniken zur Problemlösung veränderten. Allerdings wird in Anhang 7.1 ein Beispiel dafür gezeigt, wie eine Implementierung einer Lösung des Klassifikationsproblems aussehen kann.

Im Folgenden gehen wir davon aus, dass die Daten klassifiziert wurden. Das bedeutet, dass jeder gegebene Punkt $(\varphi_\xi, r_\xi)^T$, $\xi = 1, \dots, \Xi$ aus den binären Kantenbildern in Polardarstellung (vergleiche Abbildung 16) einer Klasse³¹ M_k , $k \in [1, K]$ zugeordnet wird. Aus diesen Klassen ergeben sich dann die trigonometrischen Polynome T_n^k mittels

$$\arg \min_{c_{j,k}} \sum_{\xi \in M_k} \|T_n^k(\varphi_\xi) - r_\xi\|^2.$$

³⁰Verbesserung der Beleuchtung, Kamera, etc.

³¹ M_k stellt hier eine Indexmenge dar

3.2.5 Feature Matching

Um die Daten für die spätere Tränenfilmanalyse verarbeiten zu können, müssen jeweils zwei aufeinander folgende Bilder aneinander angepasst werden. Wendet man beispielsweise eine Multiskalenanalyse zwischen den Bildern³² an, so müssen im Prinzip alle Bilder aneinander angepasst sein. Daher ist es sinnvoller, die Bilder auf eine ideale Maske zu projizieren. Die ideale Maske wird definiert als

$$x_k := x_k(\varphi) = kB + O, \quad k = 1, \dots, K \quad (5)$$

mit $B, O \in \mathbb{N}$.³³ Man beachte, dass $x_k(\varphi)$ von der Wahl von φ unabhängig ist. Daraus ergibt sich die Zuordnung des Feature Matchings

$$x_k(\varphi) \sim \sum_{j=-n}^n c_j^k e^{ij\varphi}.$$

³²wie bspw. in [7]

³³Dies entspricht K konzentrischen Kreisen mit Radius $kB + O$ in Polardarstellung

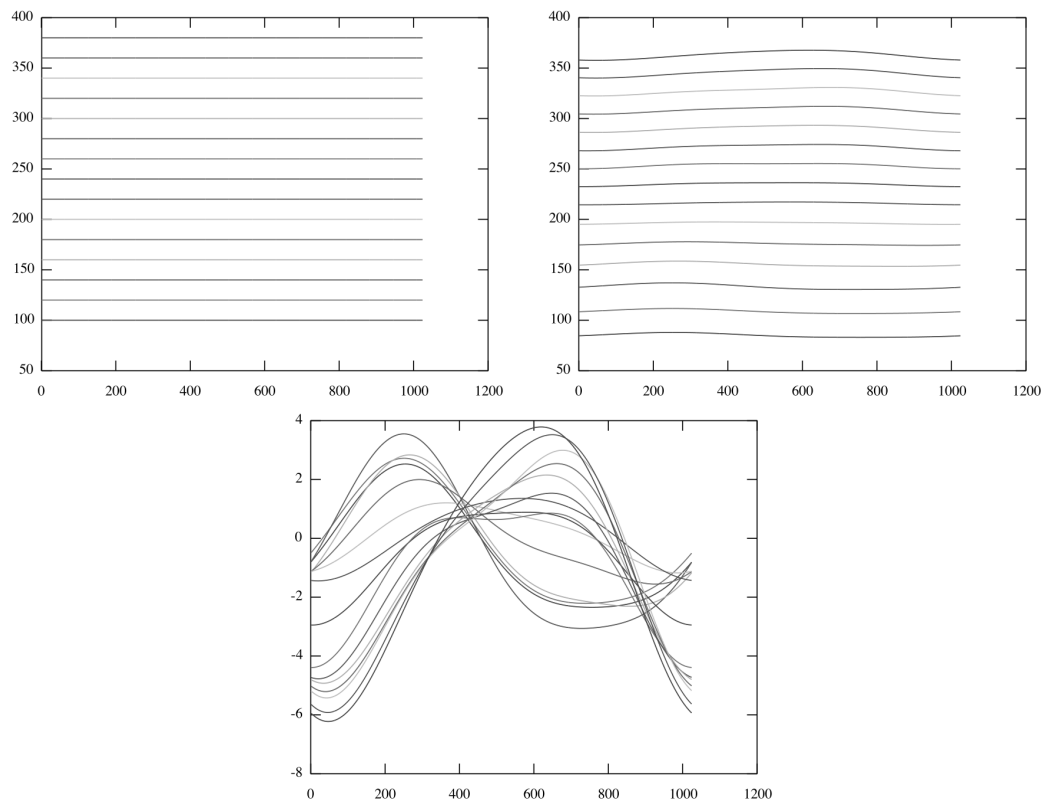


Abbildung 17: Ideale Maske in Polardarstellung(links) , berechnete trig. Polynome der deformierten Kreise des Bildes (rechts) und die trig. Polynome ohne Mittelwert ($c_0^k = 0$) (unten)

3.2.6 Funktionsklassen

Zunächst fassen wir die Ergebnisse aus den vorherigen Abschnitten zusammen. Damit die Bilder eines menschlichen Auges auf dynamische Veränderungen untersucht werden können, müssen die durch die Sakkaden entstehenden Bewegungen herausgerechnet werden. Daher werden die Bilder auf eine Idealmaske projiziert, wobei die Kanten der durch die Bewegungen deformierten Kreise denen auf perfekten³⁴ konzentrischen Kreisen entsprechen sollen. Gesucht ist also eine Abbildung T , so dass ein Bild (in Polarkoordinaten) $I_p \in L_2(\mathbb{R}^+ \times \mathbb{T})$ mit

$$I'_p(r, \varphi) = I_p(T(r, \varphi), \varphi).$$

auf eine Idealmaske $I'_p \in L_2(\mathbb{R}^+ \times \mathbb{T})$ abgebildet werden kann.

Wie in Abschnitt die Kantendetektion beschrieben, ermitteln wir die Kantenpunkte (x_ξ, y_ξ) , $\xi = 1, \dots, \Xi$. Die alternative Darstellung der Punkte in Polarkoordinaten ist

$$x_\xi + iy_\xi = r_\xi e^{i\varphi_\xi}.$$

Die Kantenpunkte werden anschließend einem deformierten Kreis $k, k = 1, \dots, K$ zugeordnet.

Unter der Voraussetzung, dass die Punkte (x_ξ, y_ξ) auf jeweils einem klassifizierten konzentrischen deformierten Kreis mit Index k liegen, lässt sich jeder Punkt als

$$\begin{pmatrix} x_\varphi^k \\ y_\varphi^k \end{pmatrix} = \begin{pmatrix} \sum_{j=-n}^n c_j^k e^{ij\varphi} \cos(\varphi) \\ \sum_{j=-n}^n c_j^k e^{ij\varphi} \sin(\varphi) \end{pmatrix}$$

mit einem Winkel φ darstellen.

Beispiel 3.4 (Motivationsbeispiel) *Bei realen Daten können, besonders bei den äußeren Kreisen, nicht immer alle deformierten Kreise ermittelt werden, wodurch eine geeignete Approximation erforderlich ist. Andererseits ist für den späteren Registrierungsprozess eine Abbildung $T(r, \varphi)$ gesucht, die sinnvollerweise mindestens stetig sein sollte.*

Die Werte c_0^k entsprechen dem mittleren Abstand des k -ten Kreises zum Mittelpunkt.

³⁴perfekt bedeutet in diesem Fall, dass sie sich der k -te Kreis als $r^k e^{i\varphi}$, für ein $r^k \in \mathbb{R}^+$, darstellen lässt.

In Abbildung 18 sind reale Daten für $k = 1, \dots, 16$ abgebildet, welche als Pluszeichen markiert sind. Dabei seien die blauen Punkte $k = 1, \dots, 8$ gegeben und die Grünen sollen geeignet approximiert werden. Abbildung 18 links oben zeigt eine Approximation mit einem kubischen Polynom, wobei der L_2 -Fehler minimiert werden soll. Damit hängt der weitere Verlauf der Funktion von allen gegebenen Datenpunkten ab (Der L_2 -Fehler beträgt 65,78135).

Die Abbildung rechts oben zeigt das Ergebnis mittels kubischer Splineinterpolation, wobei der Approximant am Rand kubisch fortgesetzt wird. Dies bedeutet, dass der weitere Verlauf der Funktion von den letzten vier Datenpunkten abhängt (Der L_2 -Fehler beträgt 335,17207).

Die rechte, untere Abbildung zeigt einen kubischen Splineinterpolanten, welcher am Rand linear fortgesetzt wird (Der L_2 -Fehler beträgt 184,61563).

Der Least-Squares Ansatz ist also wesentlich datentreuer. Allerdings hat er den Nachteil, dass die gegebenen Punkte nicht exakt interpoliert werden (Der Fehler an den gegebenen Punkten beträgt 7,0421, während der Fehler der Splineapproximanten an den gegebenen Punkten offensichtlich gleich Null ist). Daher möchten wir die beiden Verfahren kombinieren. Die Abbildung 18 rechts unten zeigt nun eine solche Kombination, welche im nächsten Abschnitt näher erläutert wird. (Der L_2 -Fehler beträgt hierbei 58,7392569277989).

Gesucht sei nun eine Funktionsklasse, die einerseits die gefundenen Linien³⁵ interpoliert, andererseits eine möglichst „glatte“ Transformation erzeugt. Wir kennen die Zuordnungen

$$x_k(\varphi) \sim \sum_{j=-n}^n c_j^k e^{ij\varphi}.$$

Anhand dieser Informationen möchten wir jetzt eine Transformationsabbildung berechnen. Es sei daran erinnert, dass die k -te Linie³⁶ als

$$\sum_{j=-n}^n c_j^k e^{ij\varphi}$$

³⁵Die trigonometrische Polynomdarstellung des radialen Terms der deformierten Kreise

³⁶Der radiale Anteil des k -ten deformierten Kreises

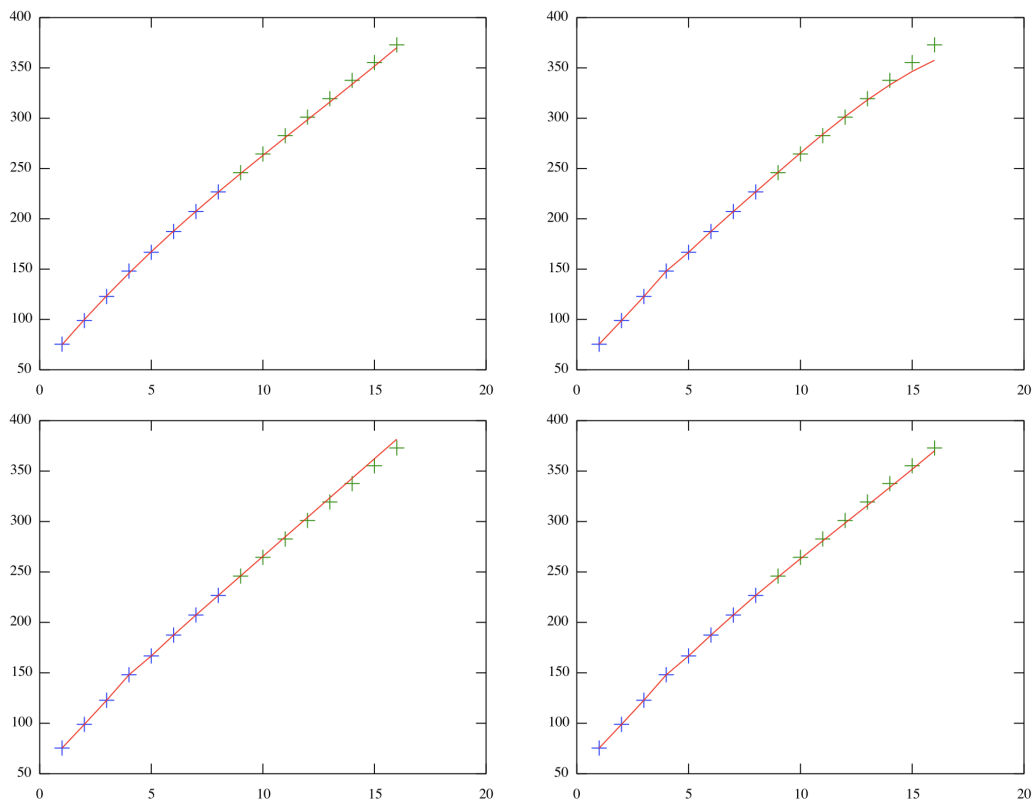


Abbildung 18: Beispiele für Interpolation und Extrapolation. Gegeben sind die blauen Punkte, welche interpoliert werden sollen und zusätzlich sollen die grünen Punkte über Extrapolation möglichst genau getroffen werden. Die folgenden Ansätze wurden verwendet: Least Squares mit kubischem Polynom (l.o.), kubischer Spline (r.o.), kubischer Spline mit linearer Randbedingung (l.u.), Least Squares mit kubischem Polynom in Kombination mit lokaler Interpolation eines kubischen Splines (r.u.)

dargestellt wird. Zunächst müssen die Linien begradigt werden, also:

$$c_j^k = 0, \forall j \neq 0, \forall k,$$

das bedeutet, dass die k -te Linie keine Schwingungen mehr enthält.

Andererseits möchten wir aber anhand der Werte³⁷ c_0^k einen Trend der Transformation erkennen, damit, falls wir nur einige Ringe aus den Bildern ermittelt konnten, trotzdem eine möglichst genaue Transformationsmaske entsteht. Für den Trend wurden algebraische Polynome gewählt.

Bemerkung 3.4 *Man mache sich hierbei den Zusammenhang mit Zernikepolynomen³⁸ klar, welche einen winkelabhängigen Term haben, welcher bei uns in der Polartransformation und in den trigonometrischen Polynomen steckt. Der radiale Term ist nun ebenso wie bei unserem Ansatz ein algebraisches Polynom. Der wesentlich Unterschied ist die Kombination der Zernikepolynome mit einer Art „Zernike-Spline“, welcher denselben winkelabhängigen Term besitzt, allerdings als radialen Term einen Spline verwendet.*

Dies führt uns zu dem im folgenden Abschnitt beschriebenen Ansatz.

Wir betrachten nun den kubischen Fall und möchten versuchen, eine zweimal stetig differenzierbare Funktion zu konstruieren, welche am im Intervall $[a, b]$ die Daten interpoliert und anschließend in zusätzlichen Randintervallen $[a', a]$, $[b, b']$, $a' \leq a$, $b \leq b'$ in die Nullfunktion übergeht. Um die Intervalle $[a', a]$ und $[b, b']$ in den Griff zu bekommen, fügen wir neue Bruchstellen ein:

$$\begin{aligned} t_0 &= x_0 - 2h, \\ t_1 &= x_0 - h \\ t_{k+m-1} &= x_k, \quad k = 0, \dots, n \\ t_{n+1} &= x_n + h, \\ t_{n+2} &= x_n + 2h, \end{aligned}$$

³⁷Zur Erinnerung: In c_0^k steckt der ungefähre Abstand zum Ursprung des k -ten Ringes. Für $j \neq 0$ sind die Werte $|c_j^k|$ relativ klein

³⁸siehe Abschnitt 2

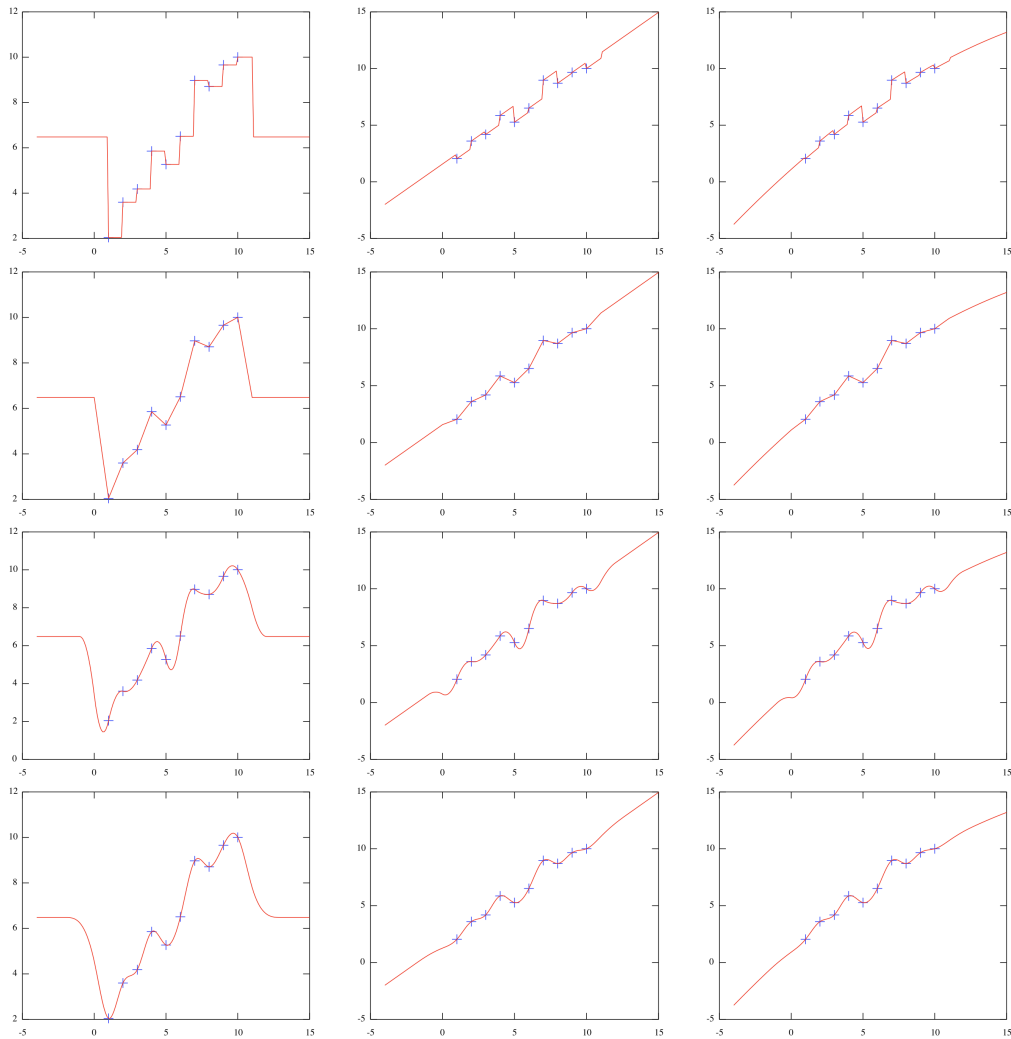


Abbildung 19: Interpolanten von $n = 10$ Punkten (blau) auf dem Intervall $[1, 10]$. Von oben nach unten wurden Splineinterpolanten von Grad $m = 0, \dots, 3$ verwendet. Auf den linken Bildern wurde die Least-Squares-Lösung der Punkte eines Polynoms vom Grad Null ermittelt und auf den Intervallen $[-5, 1 - m]$ und $[10 + m, 15]$ aufgetragen. In den Randintervallen $[1 - m, 1]$ und $[10, 10 + m]$ wurde ein $(m - 1)$ -fach stetig-differenzierbarer Übergang geschaffen. Analog wurde in den mittleren Bildern ein lineares Polynom verwendet und auf den rechten Bildern ein Quadratisches.

nutzen wir für den Spline die Interpolationseigenschaften:

$$\begin{aligned} s_0(t_0) &= 0, & s'_0(t_0) &= 0, & s''_0(t_0) &= 0 \\ s_0(t_1) &= s_1(t_1), & s'_0(t_1) &= s'_1(t_1), & s''_0(t_1) &= s''_1(t_1) \\ s_1(t_2) &= s_*(t_2), & s'_1(t_2) &= s'_*(t_2), & s''_1(t_2) &= s''_*(t_2). \end{aligned}$$

Der Spline s kann dann am linken Rand zweimal stetig differenzierbar in die Nullfunktion und rechts zweimal stetig differenzierbar in die Funktion s_* fortgesetzt werden, wobei s_* natürlich dann in die Spline-Funktion im Intervall $[a, b]$ übergehen soll. Analog kann man dieses System für das Intervall $[b, b']$ mit den Bedingungen definieren und erhält somit 16 neue Freiheitsgrade der Koeffizienten der stückweisen Polynome auf den vier neuen Intervallen und 18 Bedingungen an das Verhalten der stückweisen Polynome. Für den Spline im Intervall $[a, b]$ existieren $4n - 2$ Bedingungen (vergleiche 2.3) und $4n$ Freiheitsgrade.

Mit diesen zusätzlichen Bedingungen an h auf dem Intervall können wir h zweimal stetig differenzierbar mit der Nullfunktion außerhalb des Intervalls fortsetzen.

Mit den zusätzlichen Knoten, den Basiseigenschaften und den kompakten Trägern der B-Splines können wir nun direkt über

$$\sum_{k=0}^{n+2} \alpha_k B_k(x_{k'}) = f(x_{k'}) \quad (6)$$

mit $k' = 1, \dots, K$ interpolieren.

Um mit einem Polynom vom Grad L zu extrapolieren, benötigen wir eine Funktion der Form

$$\underbrace{f(x_{k'})}_{\text{Interpolationswerte}} = \underbrace{\sum_{l=0}^L q_l x_{k'}^l}_{\text{polynomieller Anteil}} + \underbrace{\sum_{k=0}^{n+2} \alpha_k B_k(x_{k'})}_{\text{Spline Anteil}}$$

oder in Matrixschreibweise

$$\underbrace{f}_{\text{Interpolationswerte}} = \underbrace{Px}_{\text{polynomieller Anteil}} + \underbrace{Qx}_{\text{Spline Anteil}} .$$

Damit kann $f = Px$ im L_2 -Sinne gelöst werden, um anschließend den Spline-Anteil über

$$(f - Px) = Qx$$

zu interpolieren. Damit erhalten wir eine Interpolation im Intervall $[a, b]$, einen zweimal stetig differenzierbaren Übergang in den Intervallen $[a', a]$ und $[b, b']$ und eine Extrapolation außerhalb des Intervalls $[a', b']$.

3.2.7 Modellberechnung

Nun wenden wir das eben besprochene Verfahren auf unsere Daten an.

Gesucht ist eine Abbildung T , welche die Bilder³⁹ $I_p \in L_2(\mathbb{R}^+ \times \mathbb{T})$ auf eine Idealmaske $I'_p \in L_2(\mathbb{R}^+ \times \mathbb{T})$ abbildet, so dass

$$I'_p(r, \varphi) = I_p(T(r, \varphi), \varphi).$$

Die Kantenpunkte liegen in I_p per Definition auf deformierten Kreislinien

$$T_n^k(\varphi) = \sum_{j=-n}^n c_j^k e^{ij\varphi}.$$

Sei nun ein festes j gegeben. Eine Interpolation mit Basisfunktion (B-Splines $g_k(x) = B_k(x)$) kann als

$$s_j(x) = \sum_l q_{l,j} x^l + \sum_{k=1}^K \lambda_{j,k} g_{j,k}(x)$$

definiert werden. Für die oben definierte Idealmaske mit äquidistanten Featurepunkten

$$r_k = kB + O$$

mit $B, O \in \mathbb{N}$ betrachten wir die Kurvenscharen mit $c_j(r_k) = c_j^k$ und $T_n^k(\varphi) = T_n(r_k, \varphi)$. Mit B-Splines können wir nun die Koeffizienten interpolieren.

$$c_j^{k'} = c_j(r_{k'}) = \sum_l q_{l,j} r_{k'}^l + \sum_{k=1}^K \lambda_{k,j} g_{j,k}(r_{k'}).$$

Damit haben wir die Darstellung

$$T(r, \varphi) = \sum_{j=-n}^n \left(\sum_l q_{l,j} r^l + \sum_{k=1}^K \lambda_{k,j} g_{j,k}(r) \right) e^{ij\varphi}.$$

³⁹Wir verwenden hier die Polardarstellung I_p vom Bild I . Da es sich um eine Transformation handelt, können wir beide Begriffe simultan verwenden.

Da $c_j^k \in \mathbb{C}$ ist, ist es für uns sinnvoller, folgende Gleichungssysteme zu lösen:

$$s_{j,\text{Re}}(r_{k'}) = \text{Re}(c_j(r_{k'})) = \sum_l q_{l,j,\text{Re}} r_{k'}^l + \sum_{k=1}^K \lambda_{k,j,\text{Re}} g_{j,k}(r_{k'}) \quad (7)$$

$$s_{j,\text{Im}}(r_{k'}) = \text{Im}(c_j(r_{k'})) = \sum_l q_{l,j,\text{Im}} r_{k'}^l + \sum_{k=1}^K \lambda_{k,j,\text{Im}} g_{j,k}(r_{k'}) \quad (8)$$

womit

$$c_j^k = s_{j,\text{Re}}(r_k) + i s_{j,\text{Im}}(r_k)$$

gilt.

Zusammenfassend erhalten wir für jedes gegebene (r, φ)

$$\begin{aligned} I'_p(r, \varphi) &= I_p(T(r, \varphi), \varphi) \\ &= I\left(\sum_{j=-n}^n \left(\sum_l q_{l,j} r^l + \sum_{k=1}^K \lambda_{k,j} g_{j,k}(r)\right) e^{ij\varphi}, \varphi\right). \end{aligned}$$

Mit

$$r_k = Bk + O$$

und gegebenes φ gilt dann

$$I'_p(r_{k'}, \varphi) = I_p\left(\sum_{j=-n}^n \left(\sum_l q_{l,j} r_{k'}^l + \sum_{k=1}^K \lambda_{k,j} g_{j,k}(r_{k'})\right) e^{ij\varphi}, \varphi\right).$$

Bemerkung 3.5 *Die Gleichung*

$$r_k = \sum_{j=-n}^n c_j^k e^{ij\varphi}$$

gilt im Allgemeinen nicht!

$$T_n^k(\varphi) = \sum_{j=-n}^n c_j^k e^{ij\varphi}$$

$$T_n^k(\varphi) = \sum_{j=-n}^n c_j(r_k) e^{ij\varphi}$$

Für die bekannten Paare $(T_n(r_k, \varphi), \varphi)$ gilt

$$T_n(r_k, \varphi) = \sum_{j=-n}^n c_j(r_k) e^{ij\varphi}$$

und so folgt

$$T(r, \varphi) = \sum_{j=-n}^n \left(\sum_l q_{l,j} r_k^l + \sum_{k=1}^K \lambda_{k,j} g_{j,k}(r) \right) e^{ij\varphi}$$

Beispiel 3.5 Die folgende Abbildung zeigt beispielhaft die Idee des Ansatzes.

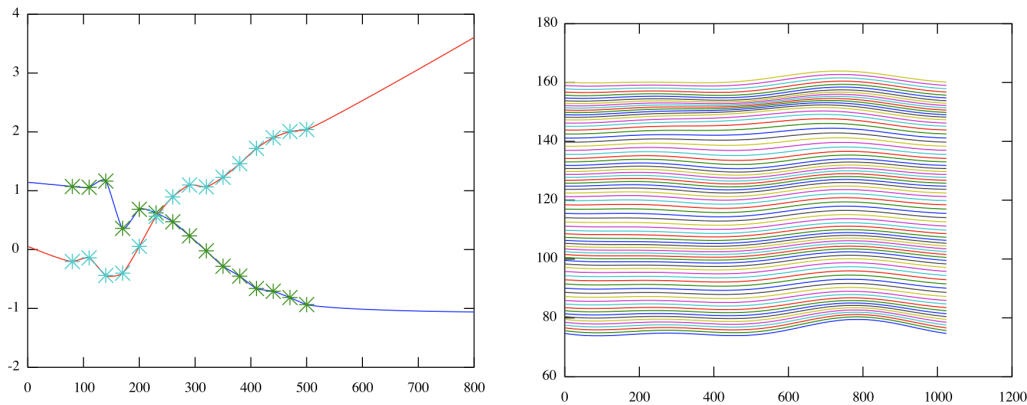


Abbildung 20: Ergebnis der Interpolation mittels kubischen B-Splines und Extrapolation mit einem quadratischen Polynom des Koeffizienten $c_1(r)$ (Realteil blau, Imaginärteil rot) eines Bildes (l.). Beispiellinien der Transformationsmaske eines Datensatzes (r.)

Bemerkung 3.6 Man könnte die Daten auch direkt über

$$T_n(r_k', \varphi) = \sum_{j=-n}^n \left(\sum_l q_{l,j} r_k'^l + \sum_{k=1}^K \lambda_{k,j} g_{j,k}(r_k') \right) e^{ij\varphi}, \varphi$$

interpolieren. Dabei ist allerdings zu beachten, dass meist ca. 50.000 Datenpunkte in Form von $(T_n(r_k', \varphi), \varphi)$ vorhanden sind. Eine Interpolation wäre daher nicht sinnvoll. Die Interpolation der wenigen Koeffizienten c_j^k ist daher unverhältnismäßig performanter.

3.2.8 inverse Polartransformation

Bei der inversen Polartransformation wird ein Bild I von Polarkoordinaten in kartesische Koordinaten überführt. Prinzipiell kann die Transformation des Punktes (x, y) von kartesischen in Polarkoordinaten über die Formeln:

$$r = \sqrt{x^2 + y^2}$$

und

$$\phi = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & x < 0, y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & x < 0, y < 0 \\ \frac{\pi}{2} & x = 0, y > 0 \\ -\frac{\pi}{2} & x = 0, y < 0 \\ 0 & x = 0, y = 0 \end{cases}.$$

definiert werden. Die Transformation eines Punktes (r, φ) von Polarkoordinaten in das kartesische Koordinatensystem kann folgendermaßen dargestellt werden:

$$x = r \cos(\varphi), \quad y = r \sin(\varphi).$$

Unsere Umsetzung erfolgte durch die Formel

$$I(m_x + r \cos(\varphi), m_y + r \sin(\varphi)) = P_I(r, \varphi)$$

mit denselben Abschätzungen und Rundungen wie in Abschnitt 3.2.3 beschrieben. In [35] wird ein Verfahren angegeben, mit dem die Qualität der diskreten Rücktransformation mithilfe von Delaunay Triangulation, bzw. baryzentrischen Koordinaten, verbessern kann. Für unsere Analysezwecke reicht allerdings das obige Rundungsverfahren.

3.2.9 Direkte Registrierung

Die Wahl der trigonometrischen Polynome zur Darstellung der Ringlinien ist nicht unbegründet. Der Zusammenhang mit den komplexen Zahlen und der Polarform gibt uns folgendes Ergebnis. Sei $z := x + iy \in \mathbb{C}$, dann gilt mit $\varphi = \arg(z)$ und $r = \text{abs}(z)$

$$T(r, \varphi) = \sum_{j=-n}^n \left(\sum_l q_{l,j} r^l + \sum_{k=1}^K \lambda_{k,j} g_{j,k}(r) \right) e^{ij\varphi}$$

Für die Bilder in Polarkoordinaten gilt

$$I'_p(r, \varphi) = I_p(T(r, \varphi), \varphi).$$

Es gilt außerdem $I_p(r, \varphi) = I(x, y)$, daher folgt

$$I'(x, y) = I_p(T(r, \varphi), \varphi).$$

Durch die komplexe Darstellung

$$\begin{aligned} T(r, \varphi) e^{i\varphi} &= T(\text{abs}(x + iy), \text{arg}(x + iy)) e^{i\text{arg}(x + iy)} \\ &= T(\text{abs}(x + iy), \text{arg}(x + iy)) \cos(\text{arg}(x + iy)) \\ &\quad + iT(\text{abs}(x + iy), \text{arg}(x + iy)) \sin(\text{arg}(x + iy)) \end{aligned}$$

erhalten wir

$$\begin{aligned} &I'(x, y) \\ &= I(T(\text{abs}(x + iy), \text{arg}(x + iy)) \cos(\text{arg}(x + iy)), T(\text{abs}(x + iy), \text{arg}(x + iy)) \sin(\text{arg}(x + iy))). \end{aligned}$$

Mit

$$r = \text{abs}(x + iy), \phi = \text{arg}(x + iy)$$

und

$$\begin{aligned} T_2(r, \varphi) &= \sum_{j=-n}^n \left(\sum_l q_{l,j} r^l + \sum_{k=1}^K \lambda_{k,j} g_{j,k}(r) \right) e^{i(j+1)\varphi} \\ &= \sum_{j=-n}^n \left(\underbrace{\sum_l q_{l,j} r^l}_{\text{polynomieller Anteil}} + \underbrace{\sum_{k=1}^K \lambda_{k,j} g_{j,k}(r)}_{\text{Splineanteil}} \right) \underbrace{e^{ij\varphi}}_{\text{trigonometrischer Anteil}} \underbrace{e^{i\varphi}}_{\text{polarer Anteil}} \end{aligned}$$

erhält man nun die Darstellung

$$\begin{aligned} I'(x,y) \\ = I(\operatorname{Re}(T_2(r,\phi)), \operatorname{Im}(T_2(r,\phi))). \end{aligned}$$

Das bedeutet, dass die Transformationsmaske direkt in kartesischen Koordinaten berechnet werden kann. Dadurch wird die Berechnung der inversen Polartransformation eingespart.

4 Ergebnisse

In diesem Abschnitt werden die Ergebnisse dieser Arbeit aufgezeigt. Im ersten Abschnitt wird die Anwendung der Detektion von konzentrischen deformierten Kreisen und die Registration der Idealmaske gezeigt. Im zweiten Abschnitt wird die Verbesserung durch die Registration durch ein Maß gezeigt. Der letzte Abschnitt beschreibt eine Anwendung auf reale Videodaten und zeigt die Möglichkeiten zur Tränenfilmanalyse auf.

4.1 Ergebnisse Detektion deformierter Kreise

An dieser Stelle möchten wir die Robustheit des Ansatzes in der Praxis vorstellen. Nachdem der Mittelpunkt auf dem Bild gefunden ist, werden die inneren 20 Kreislinien ermittelt. Dabei fehlen große Teile der Ringe und es sind zusätzlich lokale Abweichung an einigen Stellen vorhanden, welche auf Abrisse des Tränenfilms zurückzuführen sind. Die folgenden Abbildungen zeigen nun den Prozess vom Ausgangsbild über die Polartransformation⁴⁰ bis hin zur Projektion auf die Idealmaske.

⁴⁰Eine größere Abbildung von Abbildung 21 unten ist im Anhang in Abbildung 28 zu sehen.

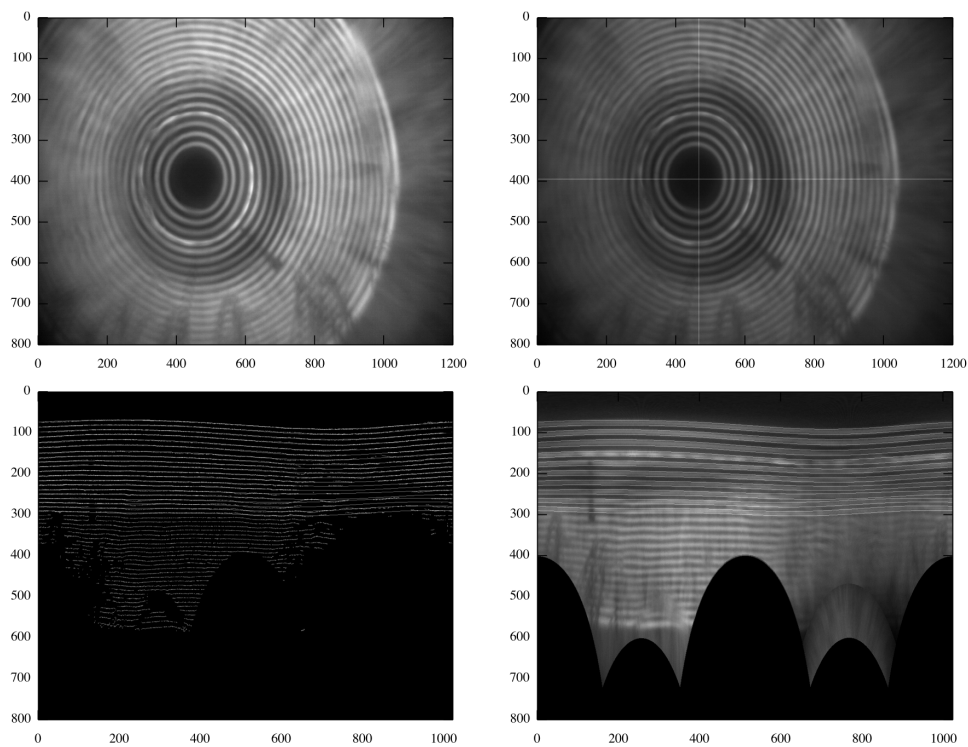


Abbildung 21: Für das Eingabebild (o.l.) wird der Mittelpunkt mittels dem in Abschnitt 3.2.2 vorgestellten Algorithmus berechnet (o.r.). In Polardarstellung nach der Kantendetektion (u.l.) kann man die Abweichungen der Linien von den perfekten Kreisen erkennen. Die berechneten Linien sind zur Anschauung noch einmal auf das Eingabebild in Polardarstellung abgebildet (u.r.). In der nächsten Abbildung sind die Ergebnisse der Registrierung zu sehen.

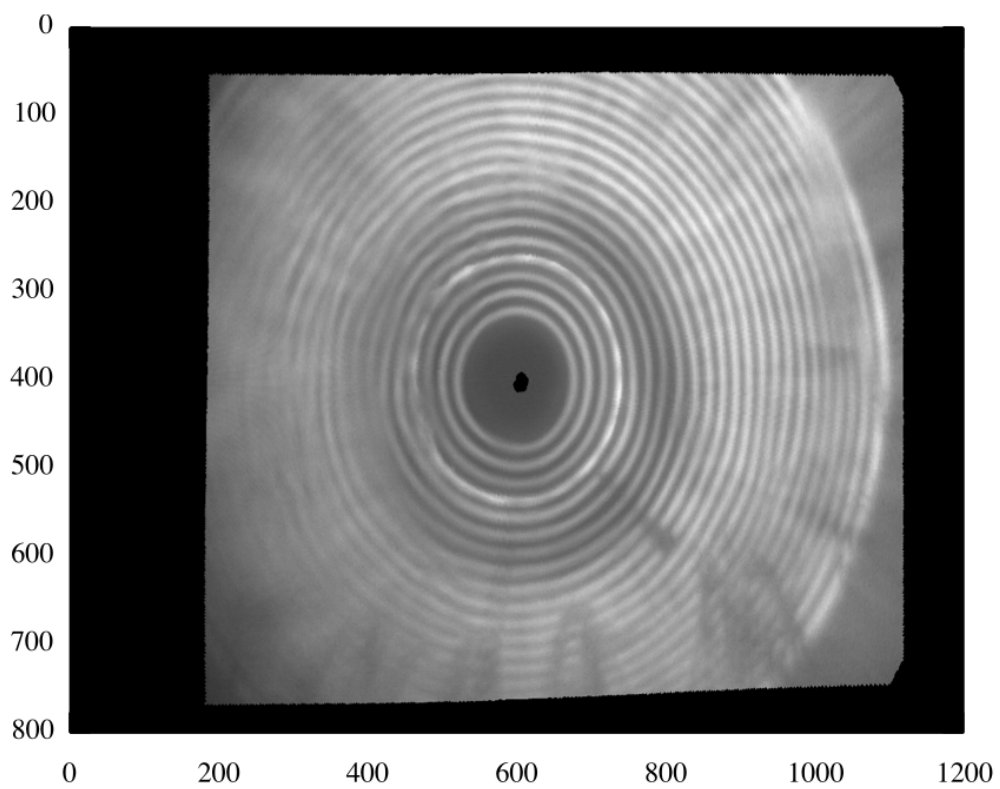
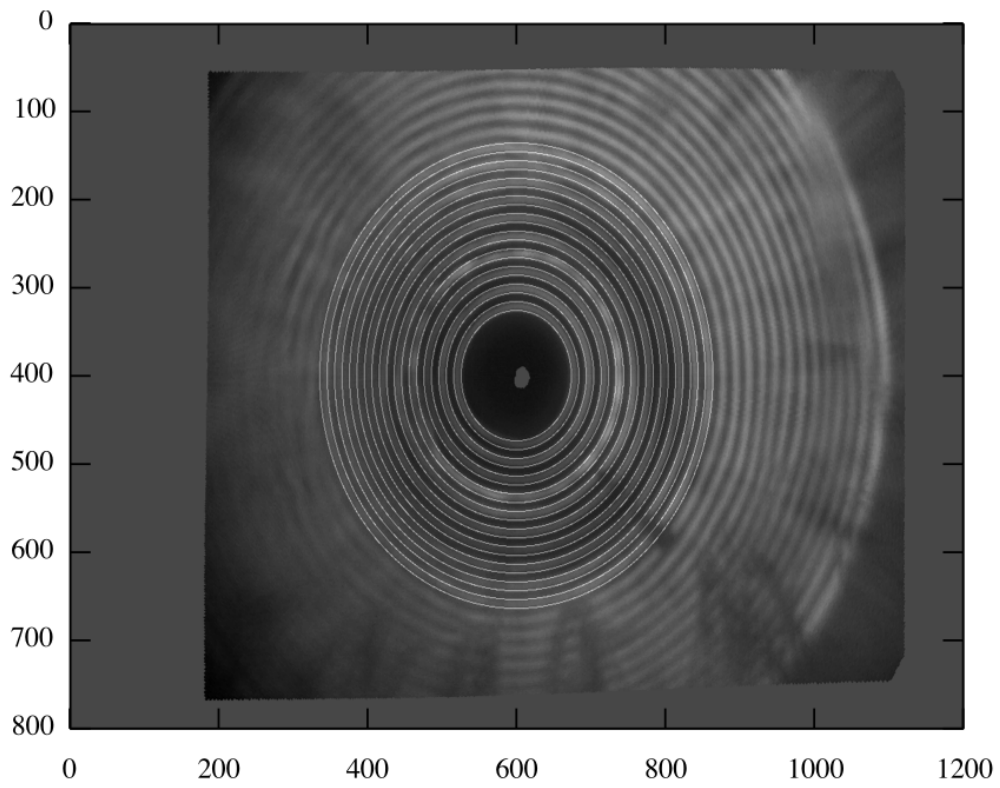


Abbildung 22: Das Ergebnis der Bildregistrierung auf die Idealmaske ist oben zu erkennen, wobei die perfekten Maskenringe hervorgehoben sind. Unten ist das Ergebnis der Anwendung der Bildregistrierung auf das Eingabebild abgebildet.

4.2 Ergebnisse Image Registration

Die Grauwerte von 16 aufeinander folgenden Bildern wurden durch einen Vorverarbeitungsschritt auf $[0, 1]$ normiert. Anschließend wurde für diese sowohl die Standardabweichung (MSE), als auch die Peak Signal to Noise Ratio (PSNR) der Differenz zum ersten Bild berechnet. Für Bilder f, g mit $M \times N$ Pixeln wird die Standardabweichung als

$$MSE(f, g) = \sum_{n=1}^N \sum_{m=1}^M \|f(n, m) - g(n, m)\|^2$$

und die Peak Signal to Noise Ratio als

$$PSNR = 10 \log_{10} \left(\frac{(\max_x(f(x)))^2}{MSE} \right)$$

definiert. Allgemein kann man sagen, dass ein größerer PSNR-Wert ausdrückt, dass f und g genauer übereinstimmen. Die folgende Tabelle zeigt die 16 Bilder vor und nach der Verarbeitung. Man sieht, dass der MSE-Wert durch die Verarbeitung sinkt und der PSNR-Wert steigt. Beides deutet darauf hin, dass sich die Bilder ähnlicher geworden sind. Der MSE der Differenz vom ersten Bild zum ersten Bild ist natürlich Null, da sie identisch sind, wodurch der PSNR-Wert natürlich Unendlich wird. Abbildung 23 zeigt, dass der Imageregistration-Algorithmus die Ähnlichkeit der Bilder verbessert, ohne die Tränenfilmaufrisse herauszurechnen.

Nr.	MSE Original	MSE Verarbeitung	PSNR Original	PSNR Verarbeitung
1	0.00000	0.00000	Inf	Inf
2	0.00490	0.00283	23.09729	25.47916
3	0.01843	0.01003	17.34549	19.98604
4	0.02027	0.00868	16.93062	20.61342
5	0.01948	0.00435	17.10354	23.61235
6	0.01947	0.00348	17.10608	24.58128
7	0.02038	0.00429	16.90708	23.67304
8	0.02478	0.00472	16.05897	23.26317
9	0.01337	0.00373	18.74018	24.27966
10	0.01202	0.00445	19.20237	23.51474
11	0.01153	0.00415	19.38173	23.81668
12	0.01458	0.00294	18.36179	25.32169
13	0.01628	0.00465	17.88425	23.32583
14	0.01614	0.00525	17.92155	22.79545
15	0.01799	0.00363	17.45020	24.39650
16	0.01421	0.00355	18.47556	24.50291

Tabelle 9: Werte der Differenzen zum ersten Bild.

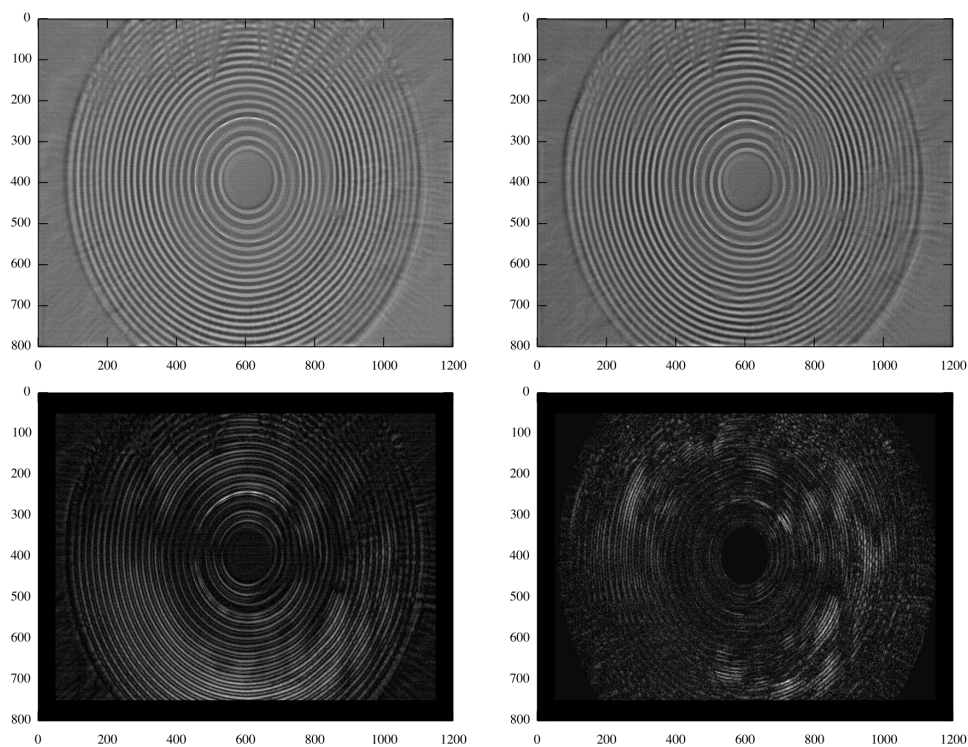


Abbildung 23: Das erste Bild (o.l.) und das 16. Bild (o.r.) der Bildserie. Unten links ist die Differenz (im Absolutbetrag) der beiden Bilder ohne Verarbeitung abgebildet, unten rechts wird die Differenz (im Absolutbetrag) nach der Verarbeitung gezeigt.

4.3 Ergebnis Tränenfilmaufriss

In Abbildung (24) sind nun 5 Videoframes aus einer Aufnahme abgebildet. Links sind die Ausgabebilder unserer Bildregistrierung abgebildet. Rechts sind die Differenzbilder vom ersten Frame und k -ten Frame abgebildet. Offensichtlich werden die Abrisse des Tränenfilmes durch deutlich helle Punkte dargestellt. Zusätzlich werden für den Menschen nicht mehr zu erkennende Abweichungen der Kreislinien gezeigt. Diese entstehen vermutlich durch lokale Verdunstung der wäßrigen Schicht, wodurch sich die Ringe nur minimal verziehen. Da es sich um Differenzbilder zum ersten Frame handelt und sich an diesen lokalen Verdunstungen ein deutlicher Verlauf im Video abzeichnet, handelt es sich hier nicht um ein Fehlverhalten des Algorithmus, sondern um eine wirkliche Strukturänderung des Tränenfilmes.

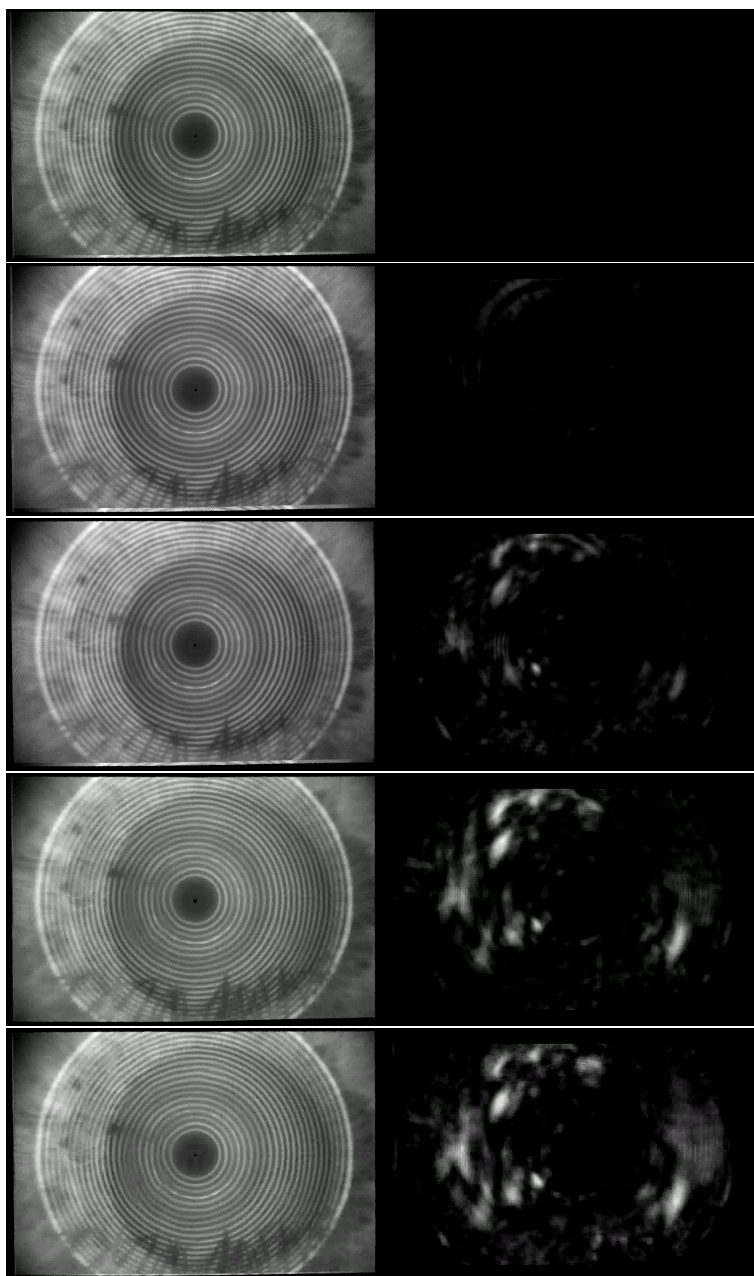


Abbildung 24: Frames 1,2,7,13,16 (von oben nach unten) aus einem Ergebnisvideo

5 Zusammenfassung

Gegenstand dieser Arbeit ist eine mathematische Methode, um aus speziellen Filmaufnahmen eines nicht-invasiven Meßgerätes das Tränenfilmverhalten des menschlichen Auges zu detektieren. Dieses automatisierte Verfahren soll Medizinern bei der objektiven Diagnostik des Dry-Eye-Syndroms unterstützen. Die Videoaufnahmen zeigen die Reflektion eines auf das Auge projizierten Musters aus konzentrischen Ringen, welche durch die Form der Hornhautvorderfläche geometrisch verzerrt werden. Durch Bewegungen der Augen, den sogenannten Sakkaden, werden die Ringe zusätzlich geometrisch deformiert, so dass sich zwei aufeinanderfolgende Bilder des Videos stark verändern können. Das in dieser Arbeit beschriebene Verfahren zeigt eine Möglichkeit, diese geometrischen Deformationen durch stabile und echtzeitfähige Algorithmen herauszurechnen, um so eine Analyse des zeitlichen Verhaltens des Tränenfilmes zu ermöglichen.

6 Ausblick

Mit den gewonnenen Erkenntnissen lässt sich nun ein Verfahren konstruieren, welches eine automatische Segmentierung der Videos in pathologische und gesunde Patienten vornimmt. Dazu müssen allerdings noch Daten gesammelt werden, um eine geeignete Klassifikation ermitteln zu können. Die genannten Ansätze sind zur Aufgabe von Feature Detection und Feature Matching den bisherigen Ansätzen in Foto-/Videokeratografen bzw. Hartmann-Shack-Sensor überlegen, da sie nicht nur die lokal umgebende Struktur eines Feature-Points betrachten, sondern auch den globalen Zusammenhang zu anderen Feature-Points durch die Verwendung von trigonometrischen Polynomen zur Beschreibung der deformierten Kreise.

Es wird vermutet, dass sich dieses Verfahren auch zur Bestimmung der kornealen Topographie nutzen lässt. Die bisherigen Verfahren mittels globalen Zernikepolynomen könnten durch⁴¹ „Zernike-Splines“ ergänzt werden, um lokale Eigenschaften der Kornea zu beschreiben und die Oszillation der Zernikepolynome höheren Grades zu vermeiden.

⁴¹Betrachtet man den polynomiellen Anteil in unserem Ansatz als Radialteil der Zernikepolynome, so ist mit „Zernike-Spline“ der Splineanteil gemeint.

7 Anhang

7.1 Feature Detection Implementierungen

An dieser Stelle möchten wir die Möglichkeiten der Feature Point Detection aus einem wie in Abbildung (16) (unten) gezeigtem Binärbild diskutieren. Zunächst werden die in den Bildern ersichtlichen Linien ermittelt. Da es sich nicht um einfache Geraden handelt, müssen die Datenpunkte durch eine geeignete Funktionsklasse beschrieben werden. Dabei bieten sich die trigonometrischen Polynome an. Diese haben erstens den Vorteil, dass sie die Linien gut approximieren und zweitens beschreiben sie die Abweichungen vom perfekten Kreis in der kartesischen Darstellung und die Abweichungen von einer konstanten Geraden in Polardarstellung. Weitere Gründe für trigonometrische Polynome sind die Orthogonalität, die Stetigkeit und die interpolativen/approximativen Eigenschaften.

Um die Datenpunkte zu unterteilen, muss nun ein Weg gefunden werden, die einzelnen Linien zu klassifizieren. Die erste Einteilung erfolgt über den Gradienten der Punkte (siehe 3.2.2). Dabei zeigt der Gradient immer in die Richtung des steilsten Anstiegs. Bei den kartesischen Koordinatenbildern bedeutet dies, dass der Vektor der Außenkanten der Ringe zum Mittelpunkt zeigt und der Vektor der Punkte auf den Innenkanten vom Mittelpunkt weg. In Polarkoordinaten können wir nun davon ausgehen, dass Außenkanten einen Gradienten mit Winkel $\frac{\pi}{2}$ und der Winkel der Gradienten der Punkte der Innenkanten $\frac{3\pi}{4}$ (siehe (25)).

Nun möchten wir die erste Linie im Bild ermitteln, welche den innersten Ring entspricht. Zunächst sei das Bild definiert als $I(r, \phi) \in \{0, 1\}$ mit $r = 1, \dots, N_1, \phi = 1, \dots, N_2$. Sei $M := \{(r(j), \phi(j)) \in \mathbb{Z} \times \mathbb{Z} : I(r(j), \phi(j)) = 1\}$ die Menge der vorhandenen Datenpunkte mit Gradientenwinkel $\frac{3\pi}{4}$ in Polarkoordinatendarstellung. Für alle $\phi(j)$ werden nun diejenigen Tupel gewählt, für die $r(j)$ minimal wird.

$$M_1 = \{(r, \phi) \in M : r = \min(r_M) \forall (r_M, \phi) \in M\}$$

Abbildung (25) zeigt das Ergebnis dieser Operation. Offensichtlich ist die erste Linie zu erkennen, allerdings sind einige Ausreißer zu erkennen. Ebenfalls sind einige Datenpunkte der dritten Linie (Anmerkung: die Gradientenwinkel der Punkte der zweiten Linie sind $\frac{\pi}{2}$) zu erkennen und zusätzlich gibt es nicht für jedes $\phi \in [\min(\phi(j)), \max(\phi(j))]$ einen minimalen Wert $r(j)$ (da natürlich nicht für jedes ϕ ein c existieren muss, so dass

$(r, \phi) \in M$.

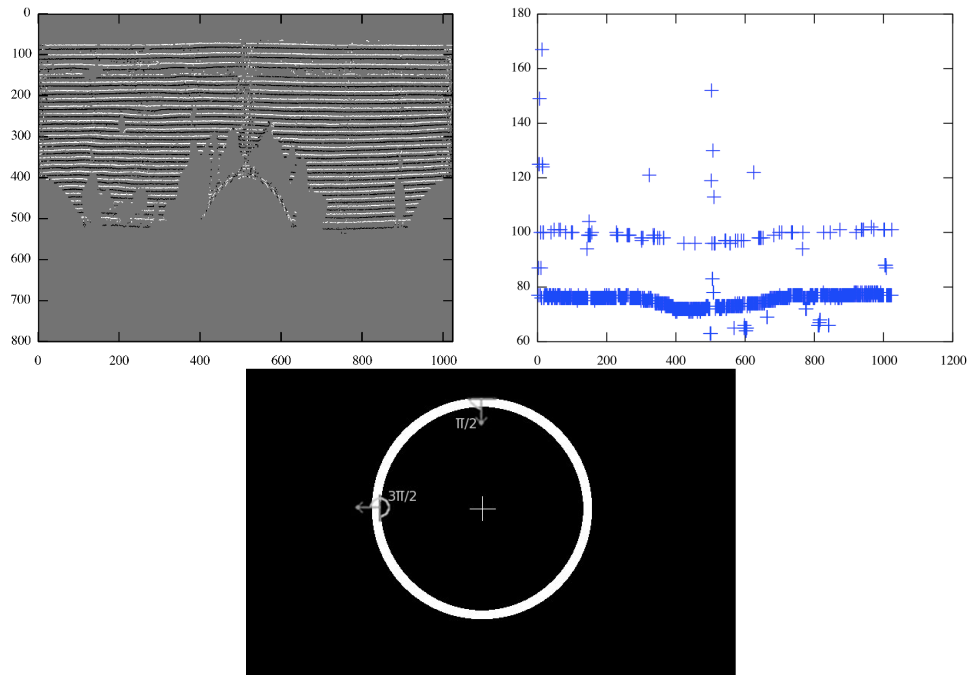


Abbildung 25: Gradientenklassifikation (links), Punkte mit minimalem Radius r (rechts), Idee der Klassifikation (unten)

Zur ersten Approximation der Linie wird nun eine Least Squares Lösung mit trigonometrischen Funktionen mit maximalem Grad zwei berechnet. Der maximale Grad wurde heuristisch anhand der vorhandenen Datensätze bestimmt und kann gegebenenfalls noch angepasst werden. Andererseits kann man den gewählten Grad auf Aberrationen niederer Ordnung zurückführen, welche beim menschlichen Auge die refraktiven Fehler wie Kurzsichtigkeit (Myopie), Weitsichtigkeit (Hyperopie) und Astigmatismus beschreiben, die durchschnittlich für 85 % der Wellenfrontaberrationen verantwortlich sind [53]. Teilweise werden die approximierten Linien durch Erhöhung des maximalen Grades besser angenähert. Dabei ist jedoch nicht klar, ob dies durch die Aberrationen und Augenbewegungen hervorgerufen wird oder durch die Ausdünnung des Tränenfilmes an diesen Stellen, wobei letzteres vermutet wird.

Zunächst möchten wir die vorhandenen Daten aus M_1 interpolieren. Ein trigonometri-

ches Polynom vom Grad n kann folgendermaßen dargestellt werden

$$p(x) = a_0 + \sum_{j=1}^n a_j \cos(jx) + \sum_{j=1}^n b_j \sin(jx).$$

Für das Polynom soll $p(x_k) = y_k$, $k = 1, \dots, 2n + 1$ gelten, wodurch sich das lineare Gleichungssystem

$$a_0 + \sum_{j=1}^n a_j \cos(jx_k) + \sum_{j=1}^n b_j \sin(jx_k) = y_k \quad (9)$$

ergibt. Gleichung (9) kann in der komplexen Ebene als

$$y_k = \sum_{j=-n}^n c_j e^{ijx_k}$$

geschrieben werden. Es sei angemerkt, dass für eine gerade Anzahl an Punkten x_k ein zusätzlicher Kosinusterm eingefügt werden kann. Sei nun n als die Mächtigkeit der Menge M_1 definiert und sei J_{M_1} eine beliebige aber feste Indexmenge von M_1 . Dann können wir das LGS für unsere Daten als

$$r_k = \sum_{j=-n}^n c_j e^{ij\phi_{k,\pi}}$$

mit $\phi_\pi = \frac{2\pi\phi}{M}$ und $(r_k, \phi_k) \in M_1$ schreiben. In Matrixschreibweise kann das LGS als $Ac = b$ dargestellt werden, wobei

$$A_{(k \in J_{M_1}, j = -|M_1|, \dots, |M_1|)} = e^{ij\phi_{k,\pi}}$$

und $b_k = r_k$, $k \in J_{M_1}$. Ist die Mächtigkeit der Menge M_1 gleich N_2 , so können die Koeffizienten c_j direkt über die schnelle inverse DFT berechnet werden. Ist dies nicht der Fall, so kann das LGS entweder über die Invertierung der Matrix A $c = A^{-1}b$ gelöst werden, oder numerisch stabiler über eine QR-Zerlegung $QR = A$, $c = R^{-1}Q^T b$. Wie oben beschrieben, sollen nur wenige Koeffizienten berechnet werden. Daher sei nun $n = 2$ und

$$r_k = \sum_{j=-2}^2 c_j e^{ij\phi_{k,\pi}}$$

bzw. $Ac = b$, wobei

$$A_{(k \in J_{M_1}, j = -2, \dots, 2)} = e^{ij\phi_k, \pi}$$

und $b_k = r_k$, $k \in J_{M_1}$. Das überbestimmte LGS kann nun über Least Squares

$$A^T A c = A^T b$$

gelöst werden. Ist hierbei die Mächtigkeit der Menge M_1 gleich N_2 (also die Anzahl der äquidistanten Stellen $\phi = 1, \dots, N_2$ und die Anzahl der gefundenen Punkte in M_2), so gilt

$$E c = A^T b N_2^{-1},$$

wobei E die Einheitsmatrix darstellt. Als nächstes wurde diese erste Näherung mittels einem Verfahren iterativ verfeinert, welches dem Iteratively Reweighted Least Squares Verfahren ähnelt. Einfach gesagt wurden alle Tupel aus M_1 entfernt, deren Abstand zum angenäherten Polynom $p(x) = \sum_{j=-2}^2 c_j e^{ijx}$ nicht innerhalb einer gewissen Toleranz

$$|p(\phi_k, \pi) - r_k| \leq \varepsilon_k$$

lag. Als Toleranz wurden diverse Ansätze getestet. Die besten Ergebnisse wurden allerdings mit einer iterativen Einschränkung $\varepsilon_k^l = (B - 1.5l)$ im Iterationsschritt l erzielt, wobei B der geschätzte durchschnittliche (radiale) Pixelabstand der Ringe ist ($B \approx 10$). Da die Ringe vorher durch die Winkel vorklassifiziert wurden (siehe (25)), verdoppelte sich der Abstand der meisten ungewollten Punkte aus M_1 .

Sei R die Indexmenge der Werte r_j aus der Menge M_1 , welche auf der gesuchten Linie liegen und G die Indexmenge aller Werte r_j mit $(r_j, \phi_j) \in M_1$. Dann ist

$$\begin{aligned} \frac{1}{|G|} \sum_{j \in G} r_j &= \frac{1}{|G|} \left(\sum_{j \in R} r_j + \sum_{j \notin R} r_j \right) \\ &= \frac{1}{|G|} \sum_{j \in R} r_j + \frac{1}{|G|} \sum_{j \notin R} r_j \end{aligned}$$

Nun soll gelten, dass $g_j = r_j$ für $j \in R$ und $g_j - 2B = r_j$ für $j \notin R$. Dies bedeutet, dass die Werte r_j auf der gesuchten Linie etwa $2B$ Pixel von der nächsten Linie entfernt

sind. Damit erhalten wir

$$\begin{aligned}
 \frac{1}{|G|} \sum_{j \in R} r_j + \frac{1}{|G|} \sum_{j \notin R} r_j &= \frac{1}{|G|} \sum_{j \in R} g_j + \frac{1}{|G|} \sum_{j \notin R} g_j + 2B \\
 &= \frac{1}{|G|} \sum_{j \in G} g_j + \frac{1}{|G|} \sum_{j \notin R} 2B \\
 &= \frac{1}{|G|} \sum_{j \in G} g_j + \frac{|R \setminus G|}{|G|} 2B
 \end{aligned}$$

und daraus folgt

$$\frac{1}{|G|} \sum_{j \in G} r_j - \frac{1}{|G|} \sum_{j \in G} g_j = \frac{|R \setminus G|}{|G|} 2B < 2B.$$

Diese erste Abschätzung zeigt also, dass die der Wert c_0 genau dann näher an der gesuchten Linie liegt, wenn $|R| > |R \setminus G|$. Unter der Annahme, dass zwei benachbarte Ringe ähnlich sind, das heißt, dass sich die darstellenden Koeffizienten $c_j, j \neq 0$ minimal ändern, hat diese Abschätzung sehr große Relevanz.

Eine Alternative ist die doppelte Standardabweichung vom Mittelwert der Punkte aus M_1 die einer weiteren Analyse bedarf. Die Ergebnisse sind in Abbildung (26) zu sehen.

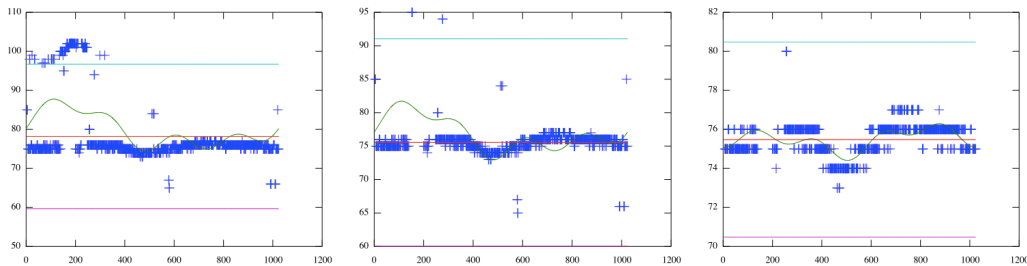


Abbildung 26: Ergebnisse nach einer (links), drei (mitte) und zehn (rechts) Iterationen

Nun haben können wir die erste angenäherte Linie mit fünf Koeffizienten darstellen. Unter der Annahme, dass sich zwei aufeinanderfolgende Ringe nur wenig verändern (d.h., dass die Differenz der Koeffizienten $c_j, j \neq 0$ der beiden Linien klein ist) kann der nächste Ring mittels Korrelationskoeffizient ermittelt werden. Diesen kann man effektiv als

$$K(r) = \sum_{\phi=1}^{N_2} I(p(\phi) + r, \phi)$$

mit $p(\phi) = \sum_{j=-2}^2 c_j e^{ij\phi 2\pi/N_2} = \sum_{j=-2}^2 c_j e^{ij\phi p_i}$ berechnen. Dies sieht man folgendermaßen: Sei

$$I_2(r, \phi) = \begin{cases} 1, & \text{falls } r = p(\phi) \\ 0, & \text{sonst} \end{cases}.$$

Dann berechnet sich die zeilenweise Korrelation als

$$K(r) = \sum_{x=1}^{N_1} \sum_{\phi=1}^{N_2} I(x+r, \phi) I_2(x, \phi)$$

und

$$I(x+r, \phi) I_2(x, \phi) = \begin{cases} 1, & \text{falls } I(x+r, \phi) = 1 \text{ und } x = p(\phi) \\ 0, & \text{sonst} \end{cases}.$$

und daraus folgt die Behauptung.

Da die einfache zeilenweise Korrelation noch zu fehleranfällig ist, wird ein zusätzlicher Toleranzterm $\varepsilon \in \mathbb{N}_0$ eingeführt, so dass wir

$$K(r) = \sum_{k=-\varepsilon}^{\varepsilon} \sum_{\phi=1}^{N_2} I(p(\phi) + k + r, \phi),$$

berechnen. Zur schnelleren Berechnung wurden die folgenden Abschätzung entwickelt:

Lemma 7.1

$$\max_{x \in \mathbb{T}} |p(x)| \leq \sum_{j=-n}^n |c_j|.$$

Beweis:

$$\begin{aligned} \max_{x \in \mathbb{T}} |p(x)| &= \max_{x \in \mathbb{T}} \left| \sum_{j=-n}^n c_j e^{ijx} \right| \\ &\leq \max_{x \in \mathbb{T}} \sum_{j=-n}^n |c_j| |e^{ijx}| \\ &= \sum_{j=-n}^n |c_j| \end{aligned}$$

□

Für die Ableitung gilt

$$\max_{x \in \mathbb{T}} |p'(x)| \leq \sum_{j=-n}^n |jc_j|.$$

In Abbildung (7.1) ist nun eine so berechnete zeilenweise Korrelation zu sehen.

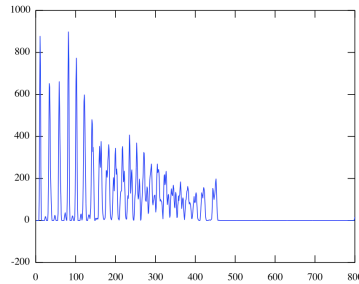


Abbildung 27: Ergebnis der zeilenweisen Korrelation

Man kann erkennen, dass die meisten Ringe schon jetzt recht gut zu differenzieren sind. Allerdings reicht „recht gut“ für eine automatische Auswertung nicht aus. Daher werden nun nicht alle Ringe gesucht, sondern lediglich der zum ersten Ring benachbarte. Man findet also eine erste Näherung für die zweite Linie durch

$$p^2(\phi) = c_0 + r^* + \sum_{j=-n, j \neq 0}^n c_j e^{ijx},$$

wobei $r^* = \min\{r : K(r) > K(r+1) \text{ und } K(r) > K(r-1)\}$. Algorithmisch berechnet man nun nicht alle lokalen Maxima und sucht nach dem ersten Auftreten, sondern berechnet die Differenzen $K_{Differenz}(r) = K(r) - K(r+1)$, und ermittelt r^* als den Punkt, an welchem das Vorzeichen von $K_{Differenz}$ das zweite mal wechselt:

$$r^* = \min\{r : \text{sgn}(K_{Differenz}(r)) > \text{sgn}(K_{Differenz}(r+1))\}.$$

Nun müssen die umgebenden Punkte der ersten Näherung ermittelt werden, welche dann wieder mittels trigonometrischer Polynome zweiten Grades approximiert werden. Es werden nun die Punkte betrachtet, welche Gradientenwinkel $\frac{\pi}{2}$ haben (siehe (25)) und in einer Umgebung von p^2 liegen und die trigonometrische Approximation erfolgt wie beim ersten Ring. Dies wird nun iterativ fortgesetzt

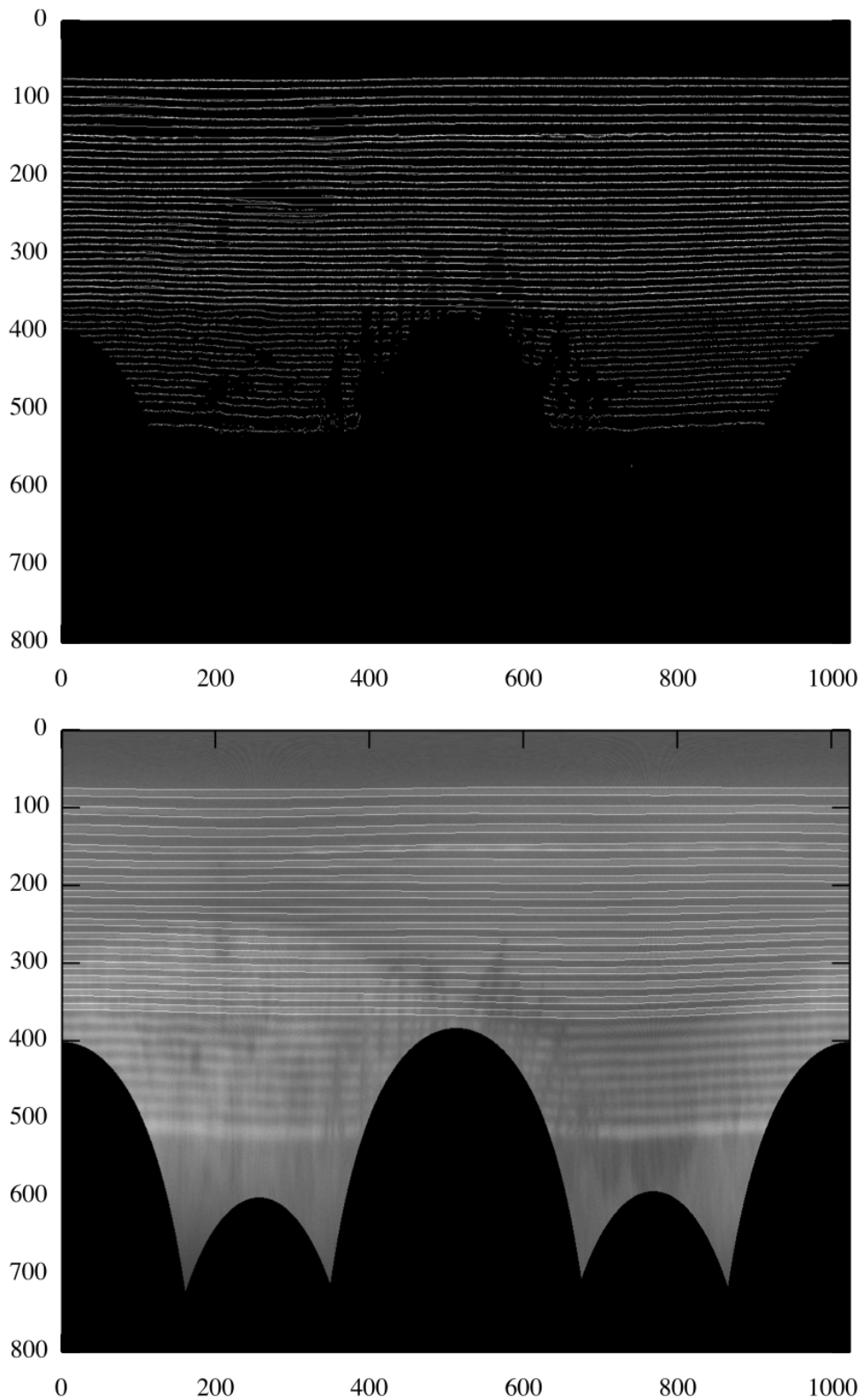


Abbildung 28: Ergebnis des iterativen Algorithmus für 30 Ringe. Die hellen Punkte zeigen perfekte Überlagerung von Punkten und den berechneten Linien. Man kann gut erkennen, wie schwerwiegend die Fehler durch Tränenfilmaufrisse für die Suche nach den Ringen sind (siehe bspw. (250,300))

Literatur

- [1] Corbett / Rosen / O'Brart: *Corneal Topography*, (BMJ Books, 1999)
- [2] Langenbacher / Viestenz / Viestenz / Brunner / Seitz : *Ray tracing through a schematic eye containing second-order (quadric) surfaces using 4 x 4 matrix notation* , *Ophthal. Physiol. Opt.* (2006) ,26, pp. 180–188
- [3] de Almeida / Carvalho: *Different Schematic Eyes and their Accuracy to the in vivo Eye: A Quantitative Comparison Study*, (*Brazilian Journal of Physics*, vol. 37, no. 2A, 378-387)
- [4] Zitova / Flusser: *Image registration methods: a survey*, (*Image and Vision Computing* 21 (2003) 977–1000)
- [5] Schölkopf / Smola: *Learning with Kernels*, (The MIT Press (2002))
- [6] Hastie / Tibshirani / Friedman: *The elements of statistical learning*, Springer (2001)
- [7] Pitz, T.: *Bildverarbeitung bei ophthalmologischen Messungen*, (Diplomarbeit), Justus-Liebig-Universität Gießen (2008)
- [8] Bahill / Adler / Stark: *Most naturally occurring human saccades have magnitudes of 15 degrees or less*, <http://www.iovs.org/cgi/reprint/14/6/468.pdf>
- [9] Lowe: *Object recognition from local scale-invariant features*, *Proceedings of the International Conference on Computer Vision* (1999), 2., pp. 1150–1157
- [10] Küchle / Busse / Küchle: *Taschenbuch der Augenheilkunde*, 4.te Auflage, Verlag Hans Huber (1998)
- [11] Goto / Zheng / Klyce / Kataoka / Uno / Karon / Tatematsu / Bessyo / Tsubota / Ohashi: *A New Method for Tear Film Stability Analysis Using Videokeratography*, *American Journal of Ophthalmology* (2003), Vol 135. No. 5, pp. 607-612
- [12] King-Smith / Fink / Fogt: *Three Interferometric Methods for measuring the thickness of layers of the tearfilm*, *Optometry and Vision Science* (1999), Vol 76, pp. 19-32

- [13] Szczesna / Kasprzak, / Jaronski / Rydz / Stenevi : *An interferometric method for the dynamic evaluation of the tear film*, Acta Ophthalmologica Scandinavica (2007), 85, pp. 202 - 208
- [14] Yokoi / Bron / TiVany / Brown / Hsuan / Fowler : *Reflective meniscometry: a non-invasive method to measure tear meniscus curvature*, Br J Ophthalmol (1999), 83, pp. 92-97
- [15] Li, K.: *Changes in aberrations and retinal image quality due to tear film dynamic*, Optics Express (2006), 14, pp. 12552-12559
- [16] Roddier, F. / Roddier, C.: *Wave-Front Reconstruction Using Iterative Fourier-Transform*, Appl. Opt. (1991), 30, pp. 1325-1327
- [17] Goto / Yagi / Matsumoto / Tsubota : *Impaired Functional Visual Acuity of Dry Eye Patients*, American Journal of Ophthalmology (2002), 133, pp. 181 - 186
- [18] Arnold / Walter / Eppig / Bruenner / Langenbucher : *Simultaneous examination of tear film break-up and the lipid layer of human eye: a novel sensor design (Part 1)*, -
- [19] Arnold / Bruenner / Langenbucher : *Simultaneous examination of tear film break-up and the lipid layer of human eye: a novel model eye for time course simulation of physiologic tear film behavior (Part 2)*, -
- [20] Mallat, S.: *A wavelet tour of signal processing*, Academic Press (1998), 2. edition
- [21] Smith G. / Atchison D.: *The eye and visual optical instruments*, Cambridge University Press (1997)
- [22] Klinke / Silbernagl: *Lehrbuch der Physiologie*, Thieme Verlag (1997)
- [23] Schünke, M.: *Prometheus: Lernatlas der Anatomie Kopf und Neuroanatomie*, Thieme Verlag (2006)
- [24] Bahill / Adler / Stark: *Most naturally occurring human saccades have magnitudes of 15 degrees or less*, Investigative Ophthalmology and Visual Science, 14,6, pp. 468 - 469

- [25] Becker / Fuchs: *Further properties of the human saccadic system: eye movements and correction saccades with and without visual fixation points*, Vision Res. (1969), 9, pp.1247-1258
- [26] Bahill / Clark / Stark: *The Main Sequence, A Tool for Studying Human Eye Movements*, Mathematical Biosciences (1975) ,24, 3-4, pp. 191-204
- [27] Tang / Yang /Liu / Ma: *Wavelet Theory and its application to pattern recognition*, World Scientific Publishing (2000)
- [28] Ruskai / Beylkin / Coifman / Daubechies / Mallat / Meyer / Raphael : *Wavelets and their applications*, Jones and Barlett Publishers (1992)
- [29] Daubechies I. : *Ten lectures on wavelets*, SIAM (1992)
- [30] Gonzalez-Meijome / Villa-Collar / Montes-Mico / Gomes : *Asphericity of the anterior human cornea with different corneal diameters*, ASCRS and ESCRS (2007)
- [31] Corbett / Rosen / O'Brart: *Corneal Topography*, (BMJ Books, 1999), p. 20, References 14-26
- [32] Nemeth / Erdelyi / Csakany / Gaspar / Soumelidis / Kahlesz / Lang : *High-speed videotopographic measurement of tear film build-up time*, Invest. Ophthalmol. Vis. Sci. (2002), 43, 6, pp. 1783– 1790
- [33] Wyant, J. : *Basic Wavefront Aberration Theory for Optical Metrology*, SIAM (1992)
- [34] Hippler, S. : *Wellenfrontanalyse mit Zernike-Polynomen und Shack-Hartmann Gradientenmessungen*, Max-Planck-Institut für Astronomie (2006), http://www.mpia.de/AOINSTRUMENTS/FPRAKT/8_f36_zernike_fit_doc.pdf
- [35] Alkhaldi, W. : *Statistical Signal and Image Processing Techniques in Corneal Matching*, (Dissertation) Darmstadt (2010)
- [36] Jähne, J. : *Digitale Bildverarbeitung*, Springer (2002)
- [37] Seul / O'Gorman / Sammon : *Practical Algorithms for image analysis*, Cambridge University Press (2000)

- [38] Girod / Greiner / Niemann : *Principles of 3D Image Analysis and Synthesis*, Kluwer Academic Publishers (2002)
- [39] Handels, H. : *Medizinische Bildverarbeitung*, Teubner (2000)
- [40] Starck / Murtagh / Bijaoui : *Image Processing and Data Analysis*, Cambridge University Press (1998)
- [41] Sauer, T.: *Digitale Signal- und Bildverarbeitung* (Vorlesungsskript), Justus-Liebig-Universität Gießen (2006), <http://www.uni-giessen.de/tomas.sauer/>
- [42] Manolakis / Ingle / Kogan : *Statistical and Adaptive Signal Processing*, McGraw-Hill Companies, Inc. (2000)
- [43] Grünigen, D. : *Digitale Signalverarbeitung*, ATVerlag (1993)
- [44] Hamming, R. W.: *Digital Filters*, Dover Publications (1998)
- [45] Vapnik, V. N.: *The Nature of Statistical Learning Theory*, Springer (1995)
- [46] Burges, C. J. C.: *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery (1998), 2, pp. 121-167
- [47] Schölkopf / Smola / Williamson / Bartlett: *New Support Vector Algorithms*, Neural Computation (2000), 12, pp. 1207– 1245
- [48] Cover, T. M. : *Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition*, IEEE Trans. Elec. Comp., 14
- [49] Forster, O.: *Analysis I*, (Vieweg, 2001)
- [50] Bass, M. et al.: *Handbook of Optics I* (McGraw-Hill, 1995)
- [51] Buhmann, M.: *Radial Basis Functions* (Cambridge University Press, 2006)
- [52] Mallat / Hwang: *Singularity Detection and processing with Wavelets* IEEE Transactions on Information Theory, 38, 2, (1992)
- [53] Lemm, D.: *Aberrationen höherer Ordnung bei Blickbewegungen* (Dissertation, 2005)

- [54] Sauer, T.: *Computeralgebra* (Vorlesungsskript), Justus-Liebig-Universität Gießen (2005)
- [55] Sauer, T.: *Numerik 2* (Vorlesungsskript), Justus-Liebig-Universität Gießen (2001)
- [56] Sauer, T.: *Splinekurven und -flächen im CAGD* (Vorlesungsskript), Justus-Liebig-Universität Gießen (2003)
- [57] Sauer, T.: *Splinekurven und -flächen im CAGD* (Vorlesungsskript), Justus-Liebig-Universität Gießen (2008)
- [58] de Boor, C.: *Splinefunktionen*, Birkhäuser, (1990)
- [59] de Boor, C.: *A practical guide to splines*, Springer, (2001)
- [60] Lin / Yuan: *Convergence rates of compactly supported radial basis function regularization* Statistica Sinica 16, (2006)
- [61] Canny, J.: *A Computational Approach to Edge Detection* IEEE Transactions on pattern analysis and machine intelligence, VOL. PAMI-8, NO. 6, (1986)

Notationen:

\mathbb{C}	Menge der komplexen Zahlen
\mathbb{N}	Menge der natürlichen Zahlen
\mathbb{R}	Menge der reellen Zahlen
\mathbb{T}	Torus $\mathbb{R}/2\pi\mathbb{Z}$
\mathbb{Z}	Menge ganzer Zahlen
$\operatorname{Re}(z)$	Realteil einer komplexen Zahl z
$\operatorname{Im}(z)$	Imaginärteil einer komplexen Zahl z
f^*	Komplexe Konjugation
$C(\mathbb{R})$	Menge stetiger reeller Funktionen
$L(\mathbb{R})$	Menge reellwertiger Funktionen
$L_{00}(\mathbb{R})$	Menge reellwertiger Funktionen mit endlichem Träger
$L_1(\mathbb{R})$	Menge reellwertiger Funktionen f mit $\int_{\mathbb{R}} f(t) dt < \infty$
$L_2(\mathbb{R})$	Menge reellwertiger Funktionen f mit $\int_{\mathbb{R}} f(t) ^2 dt < \infty$
$l(\mathbb{Z})$	Menge diskreter Folgen
$l_1(\mathbb{Z})$	Menge diskreter Folgen c mit $\sum_{t=-\infty}^{\infty} c(t) < \infty$
$l_2(\mathbb{Z})$	Menge diskreter Folgen c mit $\sum_{t=-\infty}^{\infty} c(t) ^2 < \infty$
$\ \cdot\ _p$	p-Norm
\overline{A}	algebraischer Abschluss von A
Re, Im	Realteil, Imaginärteil
T_n	trigonometrisches Polynom vom Grad n
c_j^k	j -ter Koeffizient des k -ten trig. Polynom.
g_k	Basisfunktionen
λ_k	Koeffizienten der Basisfunktionen
M, N	Träger der Bilder
K	Anzahl der Ringe/Klassen
L	Grad des Ausgleichspolynoms
Ξ	Anzahl der Datenpunkte eines Binärbildes
ϕ	Skalierungsfunktion oder RBF
ψ	Waveletfunktion

Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Gießen, den 16. Februar 2012

Unterschrift