



Masterarbeit

**Messtechnische Erfassung von
Weißlichtinterferogrammen
unter besonderer Berücksichtigung
der Messunsicherheit**

Klaus Dums

Institut für Elektrische Meßtechnik und Meßsignalverarbeitung
Technische Universität Graz
Vorstand: O. Univ.-Prof. Dipl.-Ing. Dr. techn. Georg Brasseur



Begutachter: Dipl.-Ing. Dr. techn. Markus Brandner

Graz, im Februar 2011

Kurzfassung

Schlagwörter: Weißlichtinterferometrie, Messunsicherheit, optische Messtechnik

Die Charakterisierung und Visualisierung von mikroskopischen Oberflächen hat vielfältige Anwendungen in Produktion, Qualitätssicherung und Entwicklung. Mit Hilfe von Weißlicht-Interferometern kann die Topografie von mikroskopischen Oberflächen bestimmt werden. Das Messprinzip von Weißlicht-Interferometern besteht in der Bildung von Weißlichtinterferogrammen. Im Rahmen dieser Masterarbeit wird ein Programm zur messtechnischen Erfassung von Weißlichtinterferogrammen unter besonderer Berücksichtigung der Messunsicherheit erstellt.

Im ersten Teil der Arbeit werden Voruntersuchungen durchgeführt. Es werden verschiedene Weißlicht-Interferometer und deren Anwendung vorgestellt, die Signalentstehung sowie das Messprinzip untersucht und verschiedene Möglichkeiten zur Extraktion der Höheninformation aus dem Messsignal besprochen. Des Weiteren werden Messunsicherheitseinfüsse identifiziert und Möglichkeiten zur Verringerung der Messunsicherheit diskutiert.

Der zweite Teil der Arbeit ist der Erstellung eines Programms gewidmet, das dem Messtechniker die Einstellung des Messobjekts erleichtert, die Messung sowie deren Aufzeichnung ermöglicht und eine Topografie berechnet. Zu diesem Zweck werden objektorientierte Methoden angewandt, um mit Hilfe von Anwendungsfällen einen Programm-Entwurf zu erstellen und diesen zu implementieren.

Abstract

Keywords: white-light-interferometry, measurement uncertainty, optical metrology

Characterization and visualization of microscopic surfaces has many applications in production, quality assurance and development. With white light interferometers, the topography of microscopic surfaces can be determined. The measurement principle of white light interferometers is the formation of white-light-interferograms. As part of this thesis a program for recording white-light-interferograms is created.

The first part of the work consists of preliminary studies. Multiple white-light-interferometers and their application are presented. Signal generation, measurement principle and various options for extracting height information from the measurement signal are discussed. Furthermore sources of measurement uncertainty are identified. Last but not least possibilities to improve the measurement uncertainty of the existing meter are presented.

The second part of this thesis describes the development of a program devoted to supporting the measurement engineer in setting up the measurand, recording measurement data and calculating the topography of the measurand. For this purpose, object-oriented methods are utilized to design and implement the program according to specifications that are derived from use cases.

Inhaltsverzeichnis

Einleitung	1
Theoretischer Teil	
I.1 Einleitung	3
I.2 Stand der Technik	4
I.2.1 Anwendungsbeispiele	4
I.2.2 Kommerzielle WLI	6
I.2.3 Zusammenfassung	8
I.3 Messprinzip	9
I.3.1 Interferenz zweier Wellen	9
I.3.2 Interferenz mit weißem Licht	11
I.3.3 Zusammenfassung	14
I.4 Umsetzung des Messprinzips	15
I.4.1 Mirau-Objektiv	15
I.4.2 Messgerät	16
I.4.3 Messsignal	17
I.4.4 Signalbeurteilung	18
I.4.5 Zusammenfassung	20
I.5 Signalverarbeitung	21
I.5.1 Schwerpunkt-Methode	21
I.5.2 PSI-Methoden	22
I.5.3 Kohärenz-Methode	23
I.5.4 Frequency-Domain-Analysis-Methode	28
I.5.5 Optimalfilter-Methode	30
I.5.6 STFT-Methode	31
I.5.7 Wavelet-Transformations-Methode	32
I.5.8 Zusammenfassung	33
I.6 Messunsicherheitseinflüsse	35
I.6.1 Messunsicherheitseinflüsse durch das Messobjekt	36
I.6.2 Messunsicherheitseinflüsse durch das Messgerät	38
I.6.3 Messunsicherheit durch äußere Einflüsse	41
I.6.4 Zusammenfassung	42
I.7 Verringerung der Messunsicherheit	44
I.7.1 Verbesserungen durch das Messgerät	44
I.7.2 Verbesserung durch Verringerung äußerer Einflüsse	45
I.7.3 Verbesserungen durch die Bedienbarkeit	45
Praktischer Teil	
II.1 Einleitung	49
II.2 Anwendungsfälle	50
II.2.1 Anwendungsfall 1: Vorbereitung	51
II.2.2 Anwendungsfall 2: Messung	53
II.2.3 Anwendungsfall 3: Wiederholung	55
II.2.4 Anwendungsfall 4: Wartung	56
II.3 Analyse der Anwendungsfälle	56
II.4 Entwurfskonzept	59
II.4.1 Anforderungen	59
II.4.2 Konsequenzen	60
II.4.3 Modulkonzept	60
II.4.4 Zusammenfassung	64
II.5 Entwurf und Implementierung	65
II.5.1 Systemumgebung	65
II.5.2 Implementierung von Modulen	66

II.5.3	Verwaltungsmodul	72
II.5.4	Vorbereitungs-Modul	74
II.5.5	Mess-Modul	76
II.5.6	Wiedergabe-Modul	77
II.5.7	Kamera-Modul	77
II.5.8	IO-Modul	79
II.5.9	DatenerfassungsManagement-Modul	82
II.5.10	Messdatenspeicherungs-Modul	84
II.5.11	Topografieberechnungs-Modul	85
II.5.12	Topografiespeicherungs-Modul	86
II.6	Zusammenfassung	87
	Zusammenfassung	88
	Ausblick	89

Anhang

A	Messeinrichtung	90
A.1	PC	91
A.2	Datenerfassungs-Steckkarte	91
A.3	Labor-Verstärker	91
A.4	Piezotranslator	91
A.5	Mirau-Objektiv	91
A.6	Optik	92
A.7	Lichtquelle	92
A.8	Kamera	93
A.9	Berechnungen	94
B	Algorithmen	95
B.1	RANSAC	95
B.2	Singulärwertzerlegung	96
C	Bibliotheken	98
C.1	Comedi	98
C.2	Unicap	99
C.3	Qt	99
C.4	Qwt	100
C.5	FFmpeg	100
C.6	libxml	100
C.7	boost	100
C.8	boost.log	101
D	Eidesstattliche Erklärung	102
	Literaturverzeichnis	103

Einleitung

Im Rahmen dieser Masterarbeit wird ein Programm zur messtechnischen Erfassung von Weißlichtinterferogrammen unter besonderer Berücksichtigung der Messunsicherheit erstellt. Als Weißlichtinterferogramme werden hier Messsignale bezeichnet, aus denen die topografische Karte einer mikroskopischen Oberfläche berechnet werden kann. Die so erstellte Topografie kann zur Visualisierung und Charakterisierung von Oberflächen angewandt werden.

Die Erfassung von Weißlichtinterferogrammen umfasst die Vorbereitung der Messung, die Messung selbst und die Auswertung der Messdaten. Die Aufnahme der Messdaten erfolgt durch ein bereits bestehendes am Institut für Elektrische Meßtechnik und Meßsignalverarbeitung der TU-Graz konstruiertes Weißlicht-Interferometer.

Zu Beginn der vorliegenden Arbeit werden der Stand der Technik erhoben sowie das Messprinzip und der prinzipielle Aufbau des verwendeten Weißlicht-Interferometers beschrieben. Aus einem hergeleiteten Modell des Messsignals ergeben sich Eigenschaften von Weißlichtinterferogrammen. Verschiedene Auswertemethoden werden beschrieben, die diese Eigenschaften zur Extraktion der Höheninformation aus dem Messsignal anwenden. Zum Abschluss des ersten Teils dieser Arbeit werden Einflussfaktoren, welche die Messunsicherheit erhöhen können, identifiziert und Möglichkeiten zur Verringerung der Messunsicherheit der Topografie besprochen.

Die Unterstützung des Messtechnikers durch das Programm bei Vorbereitung, Messung und Auswertung der Messdaten kann dazu beitragen, die Messunsicherheit zu minimieren. Um dies zu erreichen, werden im zweiten Teil dieser Arbeit zunächst Anwendungsfälle gebildet, die sich aus dem Arbeitsablauf bis zur Berechnung der Topografie ergeben. Unter anderem durch die Analyse der Anwendungsfälle werden Anforderungen an das Programm formuliert. Die Anwendungsfälle werden in Teilbereiche zerlegt, die in weiterer Folge in einen Entwurf entwickelt werden, mit dem die Anforderungen erfüllt werden können. Eine Beschreibung dieses Entwurfs und dessen Implementierung bilden den Abschluss dieser Arbeit.

I Theoretischer Teil

In diesem ersten Teil der vorliegenden Arbeit werden allgemeine Aspekte der Weißlicht-Interferometrie besprochen. Weißlicht-Interferometer verschiedener Hersteller und deren Anwendungsgebiete werden vorgestellt. Es wird erklärt, auf welchem physikalischen Prinzip die Messung beruht und wie ein Messgerät üblicherweise aufgebaut ist. Vorgehensweisen zur Extraktion der Höheninformation aus dem Messsignal werden gezeigt und Messunsicherheitseinflüsse identifiziert. Abschließend ist ein Kapitel der Verringerung der Messunsicherheit gewidmet.

I.1 Einleitung

Das hier beschriebene Weißlicht-Interferometer ist im Wesentlichen wie ein Mikroskop aufgebaut. Es unterscheidet sich jedoch durch das Objektiv, das eine interferometrische Messung von Entfernungsänderungen ermöglicht.

In Abbildung I.1.1 ist der grundlegende Aufbau dargestellt. Das Messobjekt wird durch das Objektiv mit weißem Licht beleuchtet. Es entsteht ein Interferenz-Bereich vor dem Objektiv. Die Distanz des Messobjekts vom Objektiv kann durch einen Translater variiert werden. Wird das Messobjekt durch den Interferenz-Bereich bewegt, entstehen Helligkeitsschwankungen, die von einer Kamera aufgezeichnet werden. Diese Schwankungen der Intensität werden als Weißlicht-interferogramm oder Korrelogramm bezeichnet. Jedes Bildelement der Kamera zeichnet ein Korrelogramm auf. Mit diesen Korrelogrammen kann die Topografie des Messobjekts bestimmt werden.

Die Messunsicherheit der Topografie wird im wesentlichen von der Oberflächenform, der Oberflächenbeschaffenheit und der Positionierung des Messobjekts beeinflusst. Der Aperturwinkel des Objektivs schränkt die Steilheit von messbaren Strukturen ein.

Wichtige Kenngrößen von Weißlicht-Interferometern sind:

- der vertikale Messbereich als der maximale Verschiebeweg des Messobjekts,
- die vertikale Auflösung als die kleinste messbare Erhöhung des Messobjekts und
- das Messfeld als den durch die Kamera beobachtbaren Bereich des Messobjekts.

Weißlicht-Interferometer zeichnen sich durch einen großen vertikalen Messbereich bei gleichzeitig hoher Auflösung und durch eine kurze Messzeit aus. Ohne zusätzliche Maßnahmen reicht das Messfeld von typischen Weißlicht-Interferometern von einem bis 800 mm² bei einem vertikalen Messbereich von 250 µm bis 70 mm und einer vertikalen Auflösung von bis zu 0,01 nm. Zusätzlich kann das Messfeld durch „stitching“ (Zusammennähen) erweitert werden. Weißlicht-Interferometer können sehr kompakt und robust aufgebaut werden und eignen sich daher auch für den Einsatz unter Produktionsbedingungen.

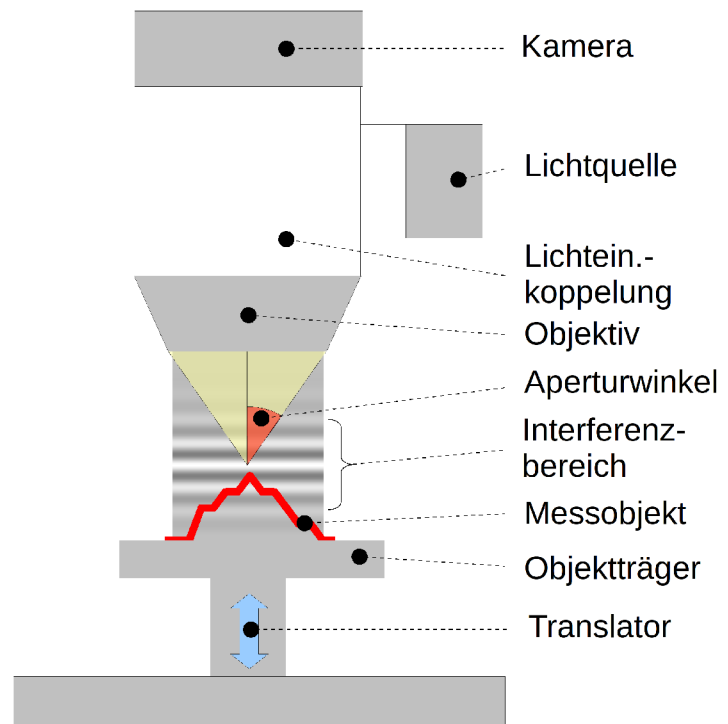


Abbildung I.1.1: Grundlegender Aufbau eines Weißlicht-Interferometers

I.2 Stand der Technik

In diesem Kapitel zeigen zunächst einige Beispiele die unterschiedlichen Einsatzgebiete von Weißlicht-Interferometern. Im Anschluss werden fünf kommerzielle Weißlicht-Interferometer vorgestellt.

I.2.1 Anwendungsbeispiele

In [1] wird eine Anwendung aus der Medizintechnik beschrieben. Implantate benötigen raue Oberflächenstrukturen. Diese ermöglichen ein Einwachsen des Implantats in den Knochen. Abbildung I.2.1 zeigt die rekonstruierte Oberfläche eines Zahnimplantats mit einer Rauheit von etwa $\pm 5 \mu\text{m}$. Mit Weißlicht-Interferometern ist die quantitative Auswertung der Oberfläche zur Qualitätskontrolle möglich.

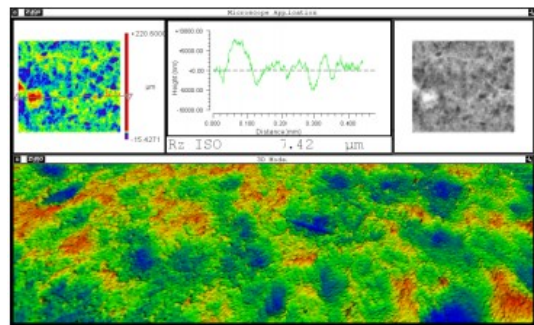


Abbildung I.2.1: Promote - Oberfläche eines Zahnimplantats der Firma Altatec GmbH, aufgenommen mit NewView 7200 der Firma ZYGO [1]

In [2] wird eine Anwendung aus der Mikromechanik beschrieben. Komponenten von Festplattenantrieben werden zur Qualitätskontrolle mittels Weißlicht-Interferometer vermessen. Abbildung I.2.2 zeigt die mit Rillen versehene Gegenplatte eines hydrodynamischen Lagers aus einem Festplattenantrieb.

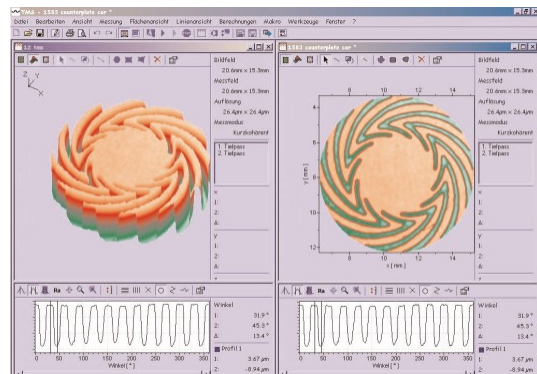


Abbildung I.2.2: Gegenplatte eines hydrodynamischen Lagers, aufgenommen mit TMS-1200 der Firma Polytec [2]

In [3] wird eine Anwendung aus der Solartechnik beschrieben. Dünnschicht-Solarzellen sind aus mehreren Schichten aufgebaut. Die oberste Schicht besteht aus einem transparenten leitfähigen Material (Transparent Conductive Oxide = TCO). Dicke und Struktur dieser Schicht nehmen maßgeblichen Einfluss auf den Wirkungsgrad einer Solarzelle. Mit einem Weißlicht-Interferometer können Schichtdicke und Struktur vermessen werden.

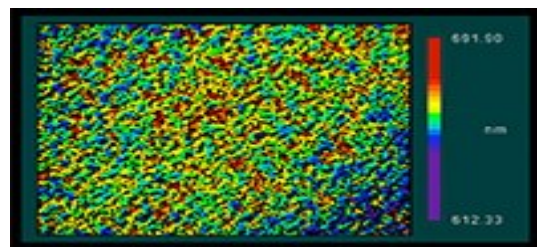


Abbildung I.2.3: TCO-Schicht-Karte, aufgenommen mit einem NewView 3DPV-TF der Firma ZYGO [3]

In [4] wird eine Anwendung aus der Mechanik beschrieben. Zur Qualitätskontrolle von Stoßdämpferkolben wird ein Weißlicht-Interferometer eingesetzt. In kurzer Zeit werden die Parallelität und die Ebenheit der Ringflächen eines Stoßdämpferkolbens überprüft.

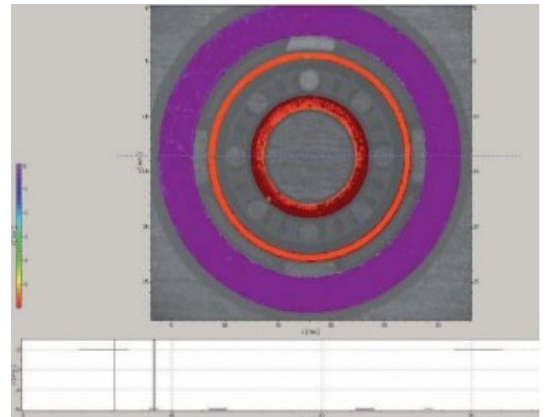


Abbildung I.2.4: Topographie der Randfläche und von zwei Innenflächen mit Linienprofil eines Stoßdämpferkolbens aufgenommen mit TMS-100 der Firma Polytec[4]

In [5] wird eine Anwendung in der Mikrooptik beschrieben. Mikrolinsen werden hauptsächlich in optischen Netzwerken verwendet, um Licht in optische Fasern zu bündeln. Mit einem Weißlicht-Interferometer können Mikrolinsen zur Qualitätssicherung vermessen werden.

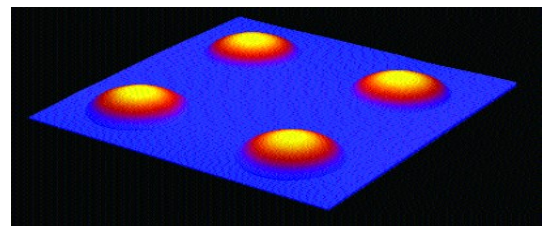


Abbildung I.2.5: Matrix von Mikrolinsen, aufgenommen mit einem Wyko NT der Firma Veeco [5]

I.2.2 Kommerzielle WLI

In diesem Unterkapitel werden exemplarisch fünf Weißlicht-Interferometer vorgestellt:

Ein Weißlicht-Interferometer der Firma Zygo wird unter dem Namen NewView 7300 (Abbildung I.2.6) vertrieben. Das Messfeld des Geräts hat eine Diagonale von maximal 14 mm. Der Messbereich beträgt maximal 20 mm. Die vertikale Auflösung wird mit weniger als 0,1 nm angegeben. Das Messobjekt kann motorisiert oder manuell positioniert werden. Mit dem motorisierten Objektträger kann die Einstellung des Fokus und der Verkippung automatisiert durchgeführt werden. Als Fokussierhilfe kann eine Kante in das optische System eingebracht werden. Zur Unterstützung der manuellen Positionierung wird ein Live-Bild angezeigt. Das Einstellen der Beleuchtungsintensität wird durch Anzeige von durchschnittlicher Intensität und durchschnittlicher Signalqualität unterstützt. In einem Live-Bild werden Pixel mit nicht ausreichender Signalqualität eingefärbt. Das zugehörige Programm „MetroPro“ steuert die Messdatenaufnahme und bietet verschiedene Methoden zur Bestimmung der Topografie an. [6]

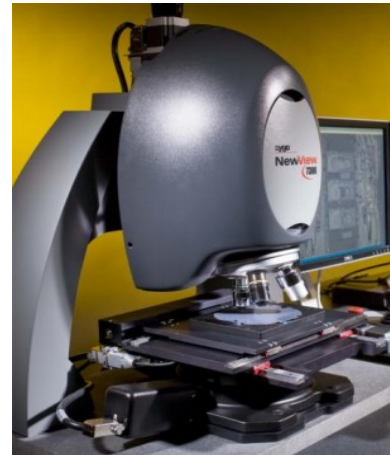


Abbildung I.2.6: NewView 7300 der Firma Zygo

Ein Weißlicht-Interferometer der Firma Veeco wird unter dem Namen Wyko NT 9800 (Abbildung I.2.7) vertrieben. Das Messfeld dieses Geräts beträgt maximal 8,45 mm in der Diagonale. Der vertikale Messbereich erreicht 10 mm bei einer vertikalen Auflösung von weniger als 0,01 nm. Bei diesem Gerät wird ein Autofokus angeboten. Zur Einstellung der Belichtung werden Pixel, deren Signalqualität einen Grenzwert unterschreitet oder die überbelichtet sind, im Live-Bild eingefärbt. Eine Verkippung kann automatisch korrigiert werden, indem die Optik des Messgeräts um den Brennpunkt des Objektivs rotiert wird. Bei Verwendung von PSI¹ als Auswerteverfahren kann auf Vibration getestet werden. Das zugehörige Programm „Vision“ steuert die Messwertaufnahme und bietet vielfältige Topografieberechnungs-Methoden an. [7]



Abbildung I.2.7: Wyko NT9800 der Firma Veeco

Das Messsystem TMS-1200 TopMap μ .Lab der Firma Polytec (Abbildung I.2.8) ist ein Weißlicht-Interferometer mit einem vertikalen Messbereich von bis zu 250 μ m und einem Messfeld von maximal 4,48 mm in der Diagonale. Die vertikale Auflösung beträgt minimal 0,035 nm. Die zugehörige Software „TMS“ bietet unter dem Namen „Smart Surface Scanning“ eine Methode an, bei der die Belichtungszeit über mehrere Messungen variiert wird. Zur Berechnung der Topografie wird für jeden Punkt das beste Signal verwendet. Die Fokussierung kann manuell oder automatisch erfolgen. Ein Live-Bild unterstützt bei der Positionierung des Messobjekts. [8]



Abbildung I.2.8: TMS-1200 der Firma Polytec

¹ Phase-Shifting-Interferometry, siehe Kapitel I.5.2.

Die selbe Firma stellt ein Weißlicht-Interferometer für relativ große Flächen her. Dieses System heißt TMS-100 TopMap Metro.Lab (Abbildung I.2.9). Das Gerät verfügt über ein Messfeld von maximal 46,4 mm in der Diagonale bei einer vertikalen Auflösung von 20 nm und einem vertikalen Messbereich von 70 mm. [9]

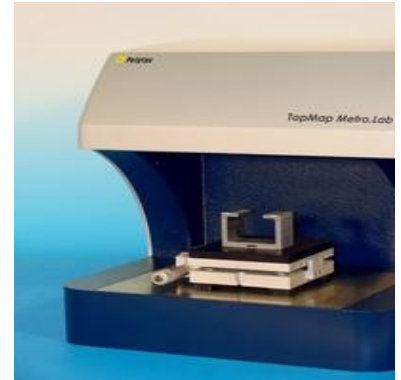


Abbildung I.2.9: TMS-100 der Firma Polytec

Die Firma Fries Research & Technology GmbH erzeugt ein Weißlicht-Interferometer mit der Bezeichnung MicroSpy Topo DT (Abbildung I.2.10). Dieses besitzt ein Messfeld maximal 8,9 mm in der Diagonale bei einem vertikalen Messbereich bis 180 μm . Die vertikale Auflösung beträgt minimal 0,1 nm. Die Positionierung des Messobjekts erfolgt motorisiert. Zusätzlich kann mit diesem Produkt auch konfokal gemessen werden. Die Fokussierung kann optional automatisiert erfolgen. Zur Unterstützung der Positionierung des Messobjekts wird ein Live-Bild angeboten. [10]



Abbildung I.2.10: MicroSpy DT der Firma FRT GmbH

I.2.3 Zusammenfassung

In diesem Kapitel zeigen Anwendungsbeispiele die Vielfältigkeit der Anwendungsbereiche von Weißlicht-Interferometern. Im Anschluss werden Messgeräte verschiedener Hersteller vorgestellt. In Tabelle I.2.1 sind die wichtigsten Eigenschaften der Geräte zusammengefasst:

	<i>NewView 7300</i>	<i>Wyko NT9800</i>	<i>MicroSpy DT</i>	<i>TMS-1200</i>	<i>TMS-100</i>
max. Messfeld ²	14 mm	8,45 mm	8,9 mm	4,48 mm	46,4 mm
max. vert. Messbereich	20 mm	10 mm	180 µm	250 µm	70 mm
vert. Auflösung	< 0,1 nm	0,1 nm	0,1 nm	0,035 nm	20 nm
Standardabweichung	< 0,01 nm	0,004 nm	-	0,25 nm	-
Software zur Messwertaufnahme	MetroPro	Vision32 Vision64	Datenauf- nahmemodul	TMS	TMS
Algorithmen ³	Schwerpunkt, FDA, VSI+PSI	Schwerpunkt, VSI, PSI, VSI+PSI	VSI, PSI	Schwerpunkt, VSI, VSI+PSI	Schwerpunkt, VSI, VSI+PSI
Automatisierte Einstellhilfen	Verkippung, Belichtung, Fokus	Verkippung, Belichtung, Fokus	Fokus	adaptive Belichtungszeit, Fokus	adaptive Belichtungszeit, Fokus
Manuelle Einstellhilfen	Live-Bilder, Belichtung, Fokussierhilfe	Live-Bilder, Belichtung	Live-Bilder, Belichtung	Live-Bilder, Belichtung	Live-Bilder, Belichtung

Tabelle I.2.1: Messgeräteübersicht

Alle Messgeräte bieten sehr ähnliche Einstellhilfen und Algorithmen an.

Für die korrekte Messwertaufnahme von besonderer Bedeutung sind die Belichtung des Bildsensors bzw. die Beleuchtung des Messobjekts. Überbelichtete Bereiche verursachen starke Verzerrungen des Messsignals. In unterbelichteten Bereichen kann das Messsignal kaum vom Rauschen getrennt werden. Üblicherweise werden Bereiche, deren Messsignal nicht den Anforderungen entspricht, im Live-Bild markiert. Die durchschnittliche Signalqualität bzw. Intensität wird ebenfalls bei einigen Messgeräten angezeigt. Die Geräte TMS-1200 und TMS-100 können mehrere Messwertaufnahmen mit verschiedenen Belichtungszeiten kombinieren. Auch wenn sich die Reflektivität von verschiedenen Bereichen des Messobjekts stark unterscheidet, kann dadurch die Topografie mit nur einer Messung ermittelt werden.

Die vorgestellten Geräte bieten Masken an, mit denen Bereiche von der Messung ausgenommen werden können. Eine Auswahl von „regions of interest“ (ROIs) im Messfeld, verbunden mit einer Optimierung des Messsignals in diesen Bereichen, wird von keinem Messgerät angeboten.

Im nächsten Kapitel wird das Messprinzip von Weißlicht-Interferometern besprochen.

² Diagonale

³ VSI (Vertical Scanning Interferometry): Verfahren, die sich auf die Einhüllende des Messsignals stützen;
PSI (Phase Shifting Interferometry): Verfahren, die sich auf die Phase des Messsignals stützen;
PSI+VSI: Kombination beide Verfahren um einen großen Messbereich bei gleichzeitig hoher Auflösung zu erhalten;
FDA (Frequency Domain Analysis): Auswertung erfolgt im Frequenzbereich;
Auswerteverfahren werden in Kapitel I.5 beschrieben.

I.3 Messprinzip

Das Messprinzip wird in ([11], Nr. 2.2) definiert als: „*Physikalische Grundlage der Messung.*“ Eine Bemerkung führt dies näher aus: „*Das Meßprinzip erlaubt es, anstelle der Messgröße eine andere Größe zu messen, um aus ihrem Wert eindeutig den der Meßgröße zu ermitteln. Es beruht auf einer immer wieder herstellbaren physikalischen Erscheinung (Phänomen, Effekt) mit bekannter Gesetzmäßigkeit zwischen der Messgröße und der anderen Größe.*“

Die physikalische Grundlage der Weißlicht-Interferometrie ist die Interferenz von weißem Licht. Als Interferenz wird die Überlagerung von Wellenzügen bezeichnet, wodurch ein neuer Wellenzug entsteht. Das Phänomen der Interferenz wird verwendet, um die Entfernungsänderung des Messobjekts zu ermitteln. Üblicherweise wird zu diesem Zweck monochromatisches Licht über zwei verschiedene optische Pfade geführt und anschließend zur Interferenz gebracht. Ändert sich die Länge eines Pfades, so ändert sich durch konstruktive bzw. destruktive Interferenz die detektierte Intensität. Die Änderung der optischen Pfaddifferenz (OPD) um eine halbe Wellenlänge bewirkt eine maximale Intensitätsänderung. Eine weitere Änderung um eine halbe Wellenlänge der OPD kehrt diese Änderung um. Daher ist die Entfernungsmessung nur innerhalb einer halben Wellenlänge der OPD eindeutig.

Durch Verwendung mehrerer Wellenlängen kann der eindeutige Messbereich vergrößert werden. Weißes Licht enthält sehr viele Wellenlängen, wodurch der Eindeutigkeitsbereich über den gewünschten Messbereich hinaus ausgedehnt werden kann. Das Interferenzmuster (Korrelogramm) erscheint nur noch in einem kleinen Bereich der OPD.

Im Folgenden wird ausgehend von der Interferenz zweier Wellen die Kohärenzfunktion als Modell des Korrelogramms entwickelt, und die Kohärenzlänge zur Abschätzung des Eindeutigkeitsbereichs eingeführt.

I.3.1 Interferenz zweier Wellen

In Anlehnung an ([12], Kap. 2) kann das elektrische Feld einer Lichtwelle als harmonische Welle, die von einer Raumkoordinate x und der Zeit t abhängig ist beschrieben werden:

$$E = A \cos\left[2\pi\left(\frac{x}{\lambda} - \frac{t}{T}\right)\right] \quad (\text{I.3.1})$$

Die Wellenlänge wird mit λ , die Zeitperiode mit T und die Amplitude mit A bezeichnet. Zwei harmonische Wellen E_1 und E_2 mit einer OPD δ können mit (I.3.1) durch

$$\begin{aligned} E_1 &= A \cos\left[2\pi\left(\frac{x}{\lambda} - \frac{t}{T}\right)\right] \\ E_2 &= A \cos\left[2\pi\left(\frac{(x-\delta)}{\lambda} - \frac{t}{T}\right)\right] \end{aligned} \quad (\text{I.3.2})$$

ausgedrückt werden. Zum Zeitpunkt $t = 0$ hat die Welle E_1 ein Maximum bei $x = 0$, die Welle E_2 hingegen bei $x = \delta$. Die Überlagerung erfolgt durch Addition der Wellen aus (I.3.2):

$$E = E_1 + E_2 = A \cos\left[2\pi\left(\frac{x}{\lambda} - \frac{t}{T}\right)\right] + A \cos\left[2\pi\left(\frac{(x-\delta)}{\lambda} - \frac{t}{T}\right)\right] \quad (\text{I.3.3})$$

I.3 Messprinzip

Unter Verwendung der Identität

$$\cos(\alpha) + \cos(\beta) = 2 \cos\left(\frac{\alpha - \beta}{2}\right) \cos\left(\frac{\alpha + \beta}{2}\right) \quad (\text{I.3.4})$$

kann (I.3.3) umgeformt werden zu:

$$E = 2A \cos\left(2\pi \frac{\delta}{2\lambda}\right) \cos\left(2\pi \left(\frac{x}{\lambda} - \frac{t}{T}\right) - 2\pi \left(\frac{\delta}{2\lambda}\right)\right) \quad (\text{I.3.5})$$

Der erste Faktor in (I.3.5) ist abhängig von der OPD δ und der Wellenlänge λ und moduliert den zweiten Faktor, der von der Zeit t und dem Weg x abhängt.

Die Interferenz zweier Lichtwellen kann nur indirekt durch Intensitätsschwankungen gemessen werden. Die Intensität I ist die mittlere Energie, die pro Zeiteinheit eine Flächeneinheit normal zur Ausbreitungsrichtung passiert. Diese ist proportional zum Quadrat des elektrischen Feldes E gemittelt über die Zeit:

$$I \propto \frac{1}{T} \int_0^T E^2 dt \quad (\text{I.3.6})$$

Mit (I.3.5) eingesetzt in (I.3.6) ergibt sich:

$$I \propto \frac{1}{T} \int_0^T \left[2A \cos\left(2\pi \frac{\delta}{2\lambda}\right) \cos\left(2\pi \left(\frac{x}{\lambda} - \frac{t}{T}\right) - 2\pi \left(\frac{\delta}{2\lambda}\right)\right) \right]^2 dt \quad (\text{I.3.7})$$

$$I \propto \frac{1}{2} \left[2A \cos\left(2\pi \left(\frac{\delta}{2\lambda}\right)\right) \right]^2$$

In (I.3.7) ist die Intensität proportional zum Quadrat des ersten Faktors aus (I.3.5) und nur von der OPD δ und der Wellenlänge λ abhängig.

Wird die OPD δ als Phasendifferenz $\Delta = 2\pi \delta / \lambda$ ausgedrückt, kann (I.3.7) umgeformt werden zu:

$$I \propto \frac{1}{2} \left[2A \cos\left(\frac{\Delta}{2}\right) \right]^2 \quad (\text{I.3.8})$$

Ein Graph der normierten Intensität über der Phase ist in Abbildung I.3.1 zu sehen. Eine eindeutige Zuordnung der Intensität zur Wegdifferenz ist nur innerhalb einer halben Wellenlänge bzw. einer Phasendifferenz von 0 bis π möglich.

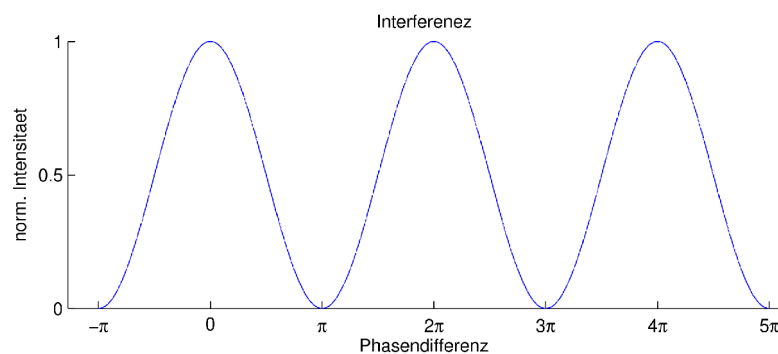


Abbildung I.3.1: Norm. Intensität der Interferenz zweier Teilstrahlen

Um den Eindeutigkeitsbereich zu vergrößern, wird im nächsten Unterkapitel ein Modell der Interferenz mit einer breitbandigen Lichtquelle entwickelt.

I.3.2 Interferenz mit weißem Licht

In Anlehnung an ([13], Kap. 2 und 3) wird im Folgenden die Kohärenzfunktion als Modell der Interferenz mit weißem Licht entwickelt. Anschließend wird der Begriff der Kohärenzlänge hergeleitet und mit dem Spektrum der Lichtquelle in Zusammenhang gebracht, sodass die Größe des Eindeutigkeitsbereichs abgeschätzt werden kann.

Gleichung (I.3.1) kann mit der Frequenz $\nu = 1/T$, der Phase $\Phi(\nu) = 2\pi x/\lambda = [2\pi x]/[c/\nu]$ und der Amplitude $a(\nu)$ sowie der Symmetrie des Cosinus umgeformt werden zur frequenz- und zeitabhängigen Größe $E(\nu, t)$:

$$E(\nu, t) = a(\nu) \cos[2\pi\nu t - \Phi(\nu)] \quad (\text{I.3.9})$$

Das resultierende elektrische Feld einer breitbandigen Lichtquelle kann durch ein Integral über alle Einzelwellen beschrieben werden:

$$V^{(r)}(t) = \int_0^{\infty} a(\nu) \cos[2\pi\nu t - \phi(\nu)] d\nu \quad (\text{I.3.10})$$

Mit der Funktion

$$V^{(i)}(t) = \int_0^{\infty} a(\nu) \sin[2\pi\nu t - \phi(\nu)] d\nu \quad (\text{I.3.11})$$

kann eine komplexe analytische Funktion

$$V(t) = V^{(r)}(t) + iV^{(i)}(t) = \int_0^{\infty} a(\nu) e^{i[2\pi\nu t - \phi(\nu)]} d\nu \quad (\text{I.3.12})$$

gebildet werden, deren Realteil dem elektrischen Feld der Lichtquelle entspricht. Analog zu (I.3.6) kann die Intensität der Lichtquelle Ausgedrückt werden durch:

$$I \propto \langle V(t) V^*(t) \rangle = 2[V^{(r)}(t)]^2 \quad (\text{I.3.13})$$

In allen weiteren Berechnungen kann $V^{(r)}(t)$ durch $V(t)$ ersetzt werden, wenn nur der Realteil des Ergebnisses betrachtet wird.

I.3.2.1 Kohärenzfunktion

Um ein Modell der Interferenz mit weißem Licht zu entwickeln, kann ein System wie in Abbildung I.3.2 betrachtet werden. Eine Lichtquelle beleuchtet einen Schirm mit den Öffnungen P_1 und P_2 . Am 2. Schirm entsteht durch das Licht aus P_1 und P_2 ein Interferenzmuster.

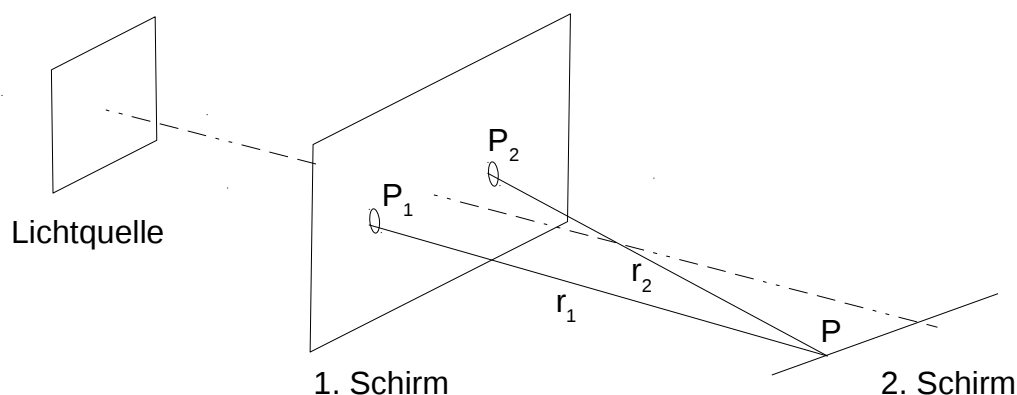


Abbildung I.3.2: System zur Entwicklung eines Kohärenz-Modells

I.3 Messprinzip

Die durch P_1 und P_2 erzeugten Strahlen werden durch die wie oben definierten analytischen Funktionen $V_1(t)$ und $V_2(t)$ beschrieben. Damit kann an Punkt P das elektrische Feld durch

$$V_P(t) = K_1 V_1(t - t_1) + K_2 V_2(t - t_2) \quad (\text{I.3.14})$$

beschrieben werden, wobei $t_1 = r_1/c$ und $t_2 = r_2/c$ die Wegzeiten der Lichtstrahlen bezeichnen. Die Konstanten K_1 und K_2 werden durch das Messgerät bestimmt. Da das Interferenzmuster nicht zeitabhängig ist, kann (I.3.14) mit $\tau = t_1 - t_2$ auch als

$$V_P(t) = K_1 V_1(t - \tau) + K_2 V_2(t) \quad (\text{I.3.15})$$

geschrieben werden.

Mit (I.3.15) ergibt sich die durchschnittliche Intensität an Punkt P analog zu (I.3.13) zu

$$\begin{aligned} I_P &\propto \langle V_P(t) V_P^*(t) \rangle \\ I_P &\propto K_1^2 \langle V_1(t + \tau) V_1^*(t + \tau) \rangle + K_2^2 \langle V_2(t) V_2^*(t) \rangle + \\ &\quad K_1 K_2^* \langle V_1(t + \tau) V_2^*(t) \rangle + K_1^* K_2 \langle V_1^*(t + \tau) V_2(t) \rangle \\ I_P &\propto K_1^2 I_1 + K_2^2 I_2 + 2 |K_1 K_2| \Re \{ \Gamma_{12}(\tau) \} \end{aligned} \quad (\text{I.3.16})$$

wobei die durchschnittliche Intensität I_1 dem Strahl aus P_1 und I_2 jenem aus P_2 zugeordnet ist. In (I.3.16) bezeichnet $\Re \{ \Gamma_{12} \} = \Re \{ \langle V_1(t + \tau) V_2^*(t) \rangle \}$ den Realteil der Korrelationsfunktion beider Strahlen.

Durch Normierung der Korrelationsfunktion mit dem geometrischen Mittel der Intensitäten I_1 und I_2 ergibt sich der Korrelationsgrad:

$$\gamma_{12}(\tau) = \frac{\Gamma_{12}(\tau)}{\sqrt{I_1 I_2}} \quad (\text{I.3.17})$$

Werden die Faktoren $K_1^2 I_1$ und $K_2^2 I_2$ zu den Intensitäten I_{P1} und I_{P2} zusammengefasst, kann (I.3.16) mit (I.3.17) umgeformt werden zu:

$$I_P \propto I_{P1} + I_{P2} + 2 \sqrt{I_{P1} I_{P2}} \Re \{ \gamma_{12}(\tau) \} \quad (\text{I.3.18})$$

Der Korrelationsgrad kann auch durch Betrag und Phase ausgedrückt werden:

$$\gamma_{12}(\tau) = m_{12}(\tau) e^{i[\alpha_{12}(\tau) - 2\pi \bar{\nu} \tau]} \quad (\text{I.3.19})$$

Der Faktor $\alpha_{12}(\tau)$ entspricht der Phasendifferenz der Strahlen aus P_1 und P_2 , $m_{12}(\tau) = |\gamma_{12}(\tau)|$ bezeichnet den Grad der Korrelation und $\bar{\nu}$ ist die Schwerpunktfrequenz der Lichtquelle.

Damit ergibt sich mit (I.3.18) die Kohärenzfunktion als Modell der Interferenz am zweiten Schirm zu:

$$I_P \propto I_{P1} + I_{P2} + 2 \sqrt{I_{P1} I_{P2}} m_{12}(\tau) \cos[\alpha_{12}(\tau) - 2\pi \bar{\nu} \tau] \quad (\text{I.3.20})$$

Der von der Laufzeitdifferenz abhängige Betrag des Korrelationsgrads $m_{12}(\tau)$ moduliert demnach ein Trägersignal der Frequenz $\bar{\nu}$. Die Frequenz $\bar{\nu}$ ist die Frequenz am Schwerpunkt des Spektrums der Lichtquelle. Da die Laufzeitdifferenz proportional zur OPD ist, kann durch den Korrelationsgrad $m_{12}(\tau)$ auf die OPD geschlossen werden.

Die Korrelationsfunktion Γ_{12} als Bestandteil des Modells führt im nächsten Kapitel zu einer Beziehung zwischen der Bandbreite der Lichtquelle und der Ausdehnung des Korrelogramms.

I.3.2.2 Kohärenzlänge

Die maximale Laufzeitdifferenz der Strahlen aus P_1 und P_2 , bei welcher der Interferenzeffekt noch auftritt, kann auf die Bandbreite der Lichtquelle zurückgeführt werden und entspricht einem Bereich der OPD, die als Kohärenzlänge bezeichnet wird.

Sind die Strahlen aus P_1 und P_2 identisch, vereinfacht sich die Korrelationsfunktion Γ_{12} aus (I.3.16) zur Autokorrelationsfunktion, die nur noch von der Laufzeitdifferenz abhängt:

$$\Gamma_{11}(\tau) = \langle V(t+\tau)V^*(t) \rangle \quad (\text{I.3.21})$$

Die normierte Korrelationsfunktion $\gamma_{11}(\tau)$ gibt den Grad der Selbstähnlichkeit als Funktion der zeitlichen Verschiebung τ an und kann mit (I.3.21) gebildet werden:

$$\gamma_{11}(\tau) = \frac{\Gamma_{12}(\tau)}{\sqrt{I_1 I_2}} = \frac{\langle V(t+\tau)V^*(t) \rangle}{\langle V(t)V^*(t) \rangle} \quad (\text{I.3.22})$$

Das Leistungsdichtespektrum $W(\nu)$ der Lichtquelle kann nach dem Wiener-Chintschin-Theorem durch Fouriertransformation der Autokorrelationsfunktion berechnet werden:

$$\mathcal{F}\{\Gamma_{11}(\tau)\} = W(\nu) \quad (\text{I.3.23})$$

Damit kann die normierte Korrelationsfunktion aus (I.3.22) durch das Leistungsdichtespektrum der Lichtquelle ausgedrückt werden:

$$\gamma_{11}(\tau) = \frac{\mathcal{F}^{-1}\{W(\nu)\}}{\int_{-\infty}^{+\infty} W(\nu) d\nu} \quad (\text{I.3.24})$$

Im Folgenden wird die effektive Bandbreite eines Spektrums benötigt. Diese ist definiert als die Breite eines Rechtecks gleicher Höhe und Fläche:

$$\Delta f_{eff} = \frac{1}{f(0)} \int_{-\infty}^{\infty} |f(t)| dt \quad (\text{I.3.25})$$

Wird eine Lichtquelle angenommen, die über eine effektive Bandbreite $\Delta\nu$ mit einer Mittelfrequenz von $\bar{\nu}$ emittiert, kann die spektrale Leistungsdichte dieser Lichtquelle durch

$$W(\nu) = \text{rect}\left[\frac{(\nu - \bar{\nu})}{\Delta\nu}\right] \quad (\text{I.3.26})$$

beschrieben werden. Die normierte Korrelationsfunktion eines solchen Spektrums ergibt sich zu:

$$\gamma_{11}(\tau) = \text{sinc}(\tau \Delta\nu) \quad (\text{I.3.27})$$

Der erste Nulldurchgang dieser Funktion liegt bei $\tau \Delta\nu = 1$. Die Kohärenzzeit wird mit τ bezeichnet. Die zugehörige Kohärenzlänge Δl ergibt sich mit der Lichtgeschwindigkeit c zu:

$$\Delta l = c \tau = \frac{c}{\Delta\nu} \quad (\text{I.3.28})$$

Nach (I.3.28) ist die Ausdehnung des Korrelogramms einer weißen Lichtquelle indirekt proportional zur Breite des Spektrums der Lichtquelle. Mit der Wellenlänge $\lambda = c/\nu$ kann die Kohärenzlänge durch

$$\Delta l \approx \frac{\bar{\lambda}^2}{\Delta\lambda} \quad (\text{I.3.29})$$

angenähert werden. In (I.3.29) werden $\bar{\lambda}$ als die mittlere Wellenlänge und $\Delta\lambda$ als Emissionsbereich der Lichtquelle bezeichnet.

In Abbildungen I.3.3 wird die inverse Proportionalität von Bandbreite der Lichtquelle zu Kohärenzlänge verdeutlicht. Oben sind die Spektren der Lichtquellen, unten die zugehörigen Korrelogramme dargestellt. Ein schmales Spektrum erzeugt ein breites Korrelogramm und umgekehrt.

I.3 Messprinzip

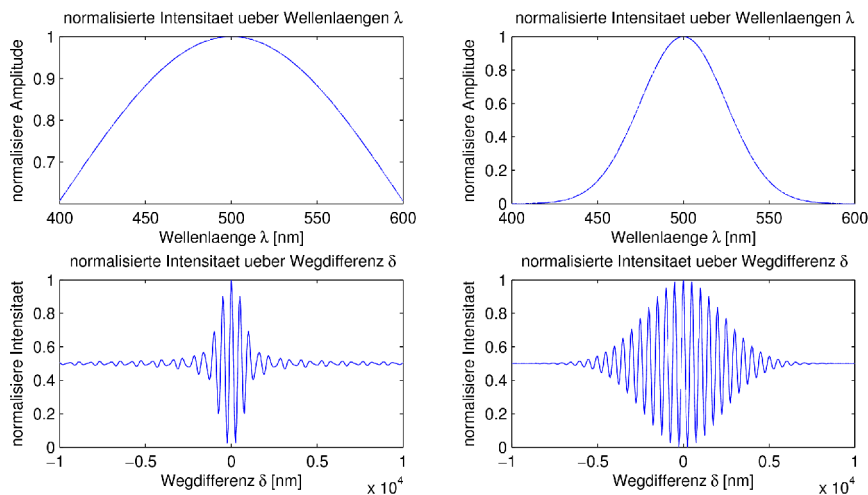


Abbildung I.3.3: Linke Spalte: Spektrum der Lichtquelle als Gauss-Verteilung mit $\mu=500$ nm und $\sigma=100$ nm und zugehöriges Korrelogramm; rechte Spalte: Spektrum der Lichtquelle als Gauss-Verteilung mit $\mu=500$ nm und $\sigma=25$ nm und zugehöriges Korrelogramm.

Unter der Annahme eines gaußverteilten Spektrums der Lichtquelle kann der Ausdruck

$$I(\delta) = C e^{-\left(\frac{\delta^2}{2\sigma^2}\right)} \cos\left(2\pi \frac{\delta}{\lambda}\right) \quad (\text{I.3.30})$$

zur Beschreibung des sich ergebenden Korrelogramms verwendet werden. Die Wegdifferenz wird als δ , die mittlere Wellenlänge des Spektrums der Lichtquelle als $\bar{\lambda}$ bezeichnet. Die Breite der einhüllenden Gaußkurve wird durch σ bestimmt und ist indirekt proportional zur Breite des Spektrums der Lichtquelle.

I.3.3 Zusammenfassung

Als Messprinzip der Weißlicht-Interferometrie dient der Effekt der Interferenz einer breitbandigen Lichtquelle. Die Interferenz ist die Überlagerung von Wellenzügen, sodass ein neuer Wellenzug entsteht.

Anhand zweier Wellen wird gezeigt, dass die messbare Intensität nur von der Wellenlänge und der optischen Pfaddifferenz (OPD) abhängt. Eine eindeutige Zuordnung von Wegänderung zu Intensitätsänderung ist hier nur im Bereich einer halben Wellenlänge möglich. Der Eindeutigkeitsbereich kann durch Verwendung einer breitbandigen Lichtquelle vergrößert werden.

Werden zwei Strahlen einer breitbandigen Lichtquelle überlagert, kommt es nur in einem kleinen Bereich der OPD zur Interferenz. Die hier beobachtbaren Intensitätsschwankungen können durch die Kohärenzfunktion beschrieben werden. Die Kohärenzfunktion beschreibt eine durch die Korrelationsfunktion der beiden Teilstrahlen modulierte Schwingung.

Die Kohärenzlänge ist jener Bereich der OPD, in der es noch zur Interferenz kommt. Zur Abschätzung der Kohärenzlänge kann das Spektrum der Lichtquelle herangezogen werden.

Im nächsten Kapitel wird der Aufbau eines Weißlicht-Interferometers besprochen.

I.4 Umsetzung des Messprinzips

Dieses Kapitel befasst sich mit dem Aufbau eines Weißlicht-Interferometers. Ein wichtiger Bestandteil ist das Mirau-Objektiv. Dieses Mikroskopobjektiv ermöglicht die interferometrische Messung der Entfernungsänderung und wird folgenden Unterkapitel besprochen.

I.4.1 Mirau-Objektiv

In der Patentschrift von André Henri Mirau [14] aus dem Jahr 1952 wird das nach ihm benannte Objektiv (Abbildung I.4.1) beschrieben. Der Elementarstrahl AB wird durch die Objektivlinse gebrochen und anschließend durch den Strahlteiler aufgeteilt. Ein Teil des Strahls verläuft über Punkt D', ein anderer Teil über D. An Punkt D' befindet sich eine spiegelnde Referenzoberfläche, an Punkt D das Messobjekt. Die Reflexionen beider Strahlen treffen an Punkt E wiederum auf den Strahlteiler. Hier interferieren die Teilstrahlen von D und D'. Nach Durchlaufen der Objektivlinse kann die Intensität des resultierenden Strahls FG mit einem Sensor aufgezeichnet werden.

Die Wegdifferenz der Pfade ABCDEFG und ABCD'EFG bestimmt die Intensität der Interferenz beider Strahlen. Das Interferenzmaximum tritt bei gleicher optischer Pfadlänge der Strahlen auf. Der Strahl ABCD'EFG wird im Weiteren als Referenzstrahl, der Strahl ABCDEFG als Objektstrahl bezeichnet.

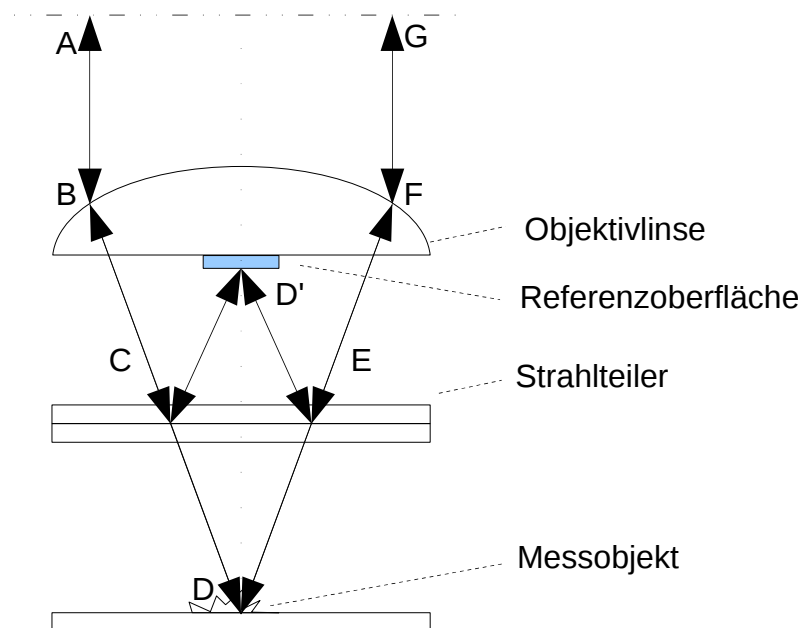


Abbildung I.4.1: Schematischer Aufbau eines Mirau-Objektivs nach [14]

Durch die Integration des Mirau-Objektivs in ein Mikroskopobjektiv ist dieses besonders robust gegenüber äußeren Einflüssen. Abhängig von der Vergrößerung des Mikroskopobjektivs besitzt es eine sehr geringe Schärfentiefe. Um im Bereich des Interferenzmaximums möglichst scharf abzubilden, sind Mirau-Objektive so aufgebaut, dass der Fokus und das Interferenzmaximum zusammenfallen. Um die weitere Auswertung zu unterstützen, sollte die Ausdehnung des Korrelogramms geringer als die Schärfentiefe sein.

I.4.2 Messgerät

In Abbildung I.4.2 ist eine schematische Darstellung des am Institut für Elektrische Messtechnik und Meßsignalverarbeitung der TU-Graz aufgebauten Messgeräts zu sehen. Das Licht einer breitbandigen Quelle wird zur Beleuchtung mit parallelem Strahlengang eingekoppelt. Durch das Mirau-Objektiv entsteht ein von der Entfernung des Messobjekts abhängiges Messsignal. Die Kamera zeichnet mit jedem ihrer Bildelemente ein Messsignal auf. Das Messobjekt kann mit Hilfe eines piezoelektrischen Translators entlang der optischen Achse des Objektivs bewegt werden. Dieser Aufbau wird auch als „vertical scanning white light interferometer“ (VSWLI) bezeichnet.

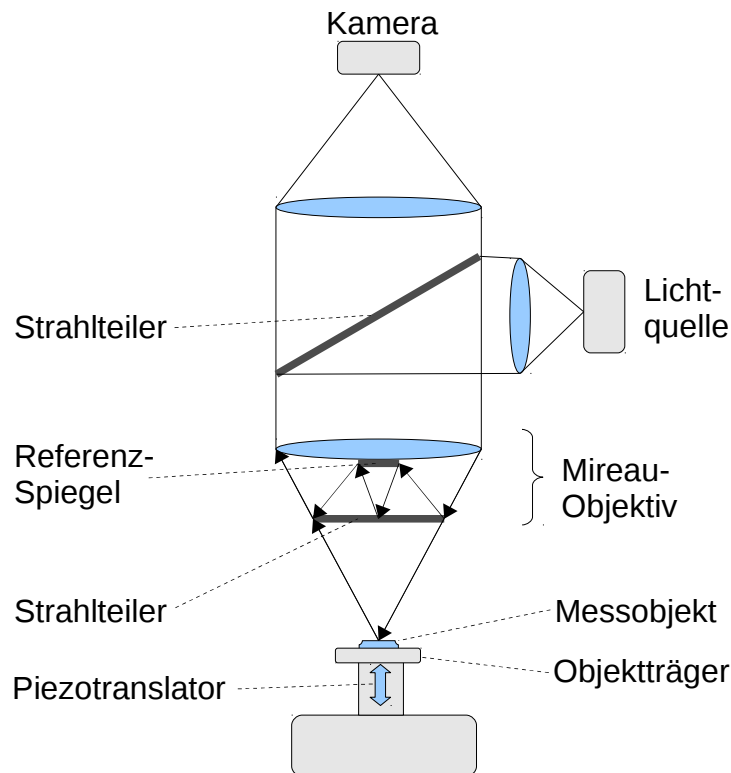


Abbildung I.4.2: Schema des Messgeräts

In Abbildung I.4.3 ist die Funktionsweise des Messgeräts skizziert. Das Profil des Messobjekts ist durch die rote Linie angedeutet. Das Objektiv erzeugt einen Interferenzbereich in einem vorgegebenen konstanten Abstand, der als Streifenmuster dargestellt ist. Das Messobjekt wird entlang der optischen Achse (in z-Richtung) dem Objektiv angenähert oder entfernt und durchfährt damit den Interferenzbereich. Jedes Bildelement der Kamera zeichnet die Intensitätsänderung des korrespondierenden Bereiches am Messobjekt auf während dieses durch den Interferenzbereich bewegt wird. Diese Messsignale und der Wert der Verschiebung bilden die Grundlage zur Bestimmung der Topografie des Messobjekts.

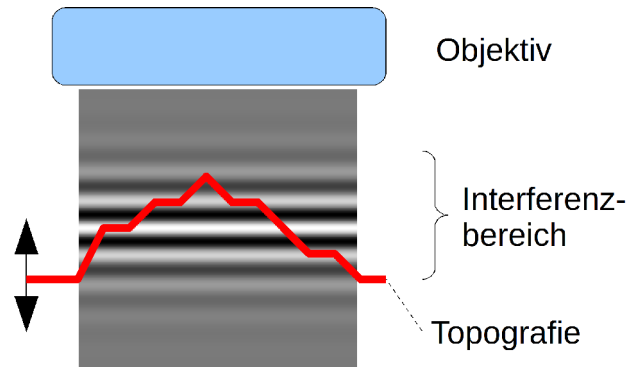


Abbildung I.4.3: Das Messobjekt (rot) ragt in den Interferenzbereich.

I.4.3 Messsignal

Unter der vereinfachten Annahme, dass die Beleuchtung mit einer Lichtquelle erfolgt, deren Spektrum einer Gaußkurve ähnelt, ergibt sich ein idealisiertes Messsignal wie in Abbildung I.4.4. In dieser Abbildung ist die Intensität über die optische Wegdifferenz des Referenzstrahls und des Objektstrahls aufgetragen.

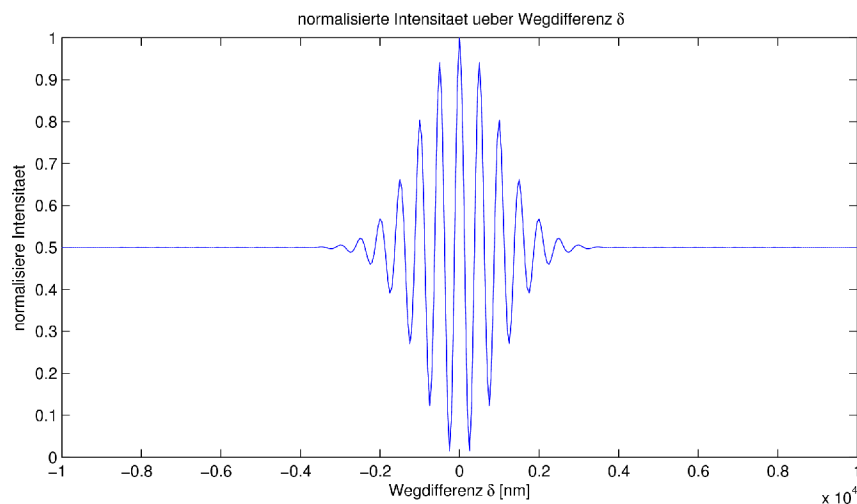


Abbildung I.4.4: Korrelogramm einer Lichtquelle mit gaußförmigen Spektrum

Das Interferenzmaximum (zentraler Streifen) befindet sich an jener Stelle, an der die OPD gleich Null ist. Hier überlagern sich alle Wellen des breitbandigen Objekt- und Referenzstrahls konstruktiv. Je weiter das Objekt bewegt wird, desto größer wird die optische Wegdifferenz beider Strahlen. Damit nimmt die Kohärenz der Überlagerung ab, bis schließlich keine Interferenz mehr auftritt.

Durch ein gaußförmiges Spektrum der Lichtquelle ergibt sich eine gaußförmige Einhüllende und ein ebenfalls gaußförmiges Spektrum des Korrelogramms. Die Phase des Korrelogramms ist linear. In Abbildung I.4.5 wird dieser Zusammenhang schematisch dargestellt. Die Amplituden der Einzelwellen λ_1 bis λ_2 sind gaußverteilt. Die Summe aller Einzelwellen erzeugt ein Korrelogramm, dessen Einhüllende ebenfalls gaußförmig ist.

I.4 Umsetzung des Messprinzips

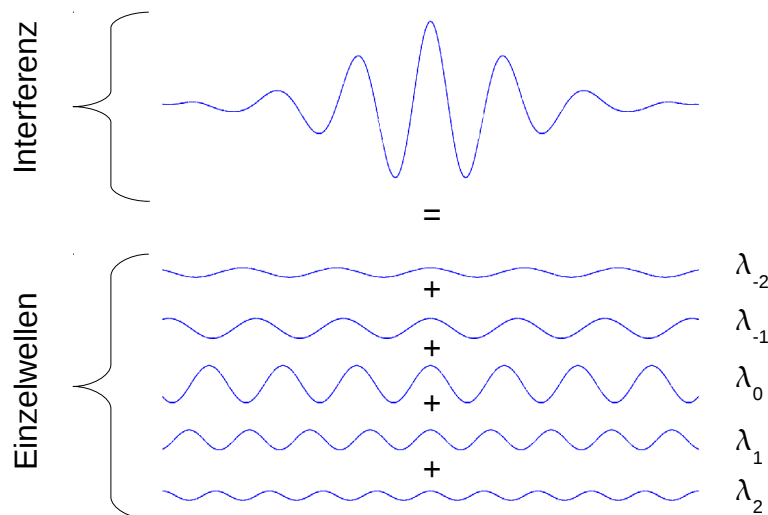


Abbildung I.4.5: Interferenz als Überlagerung von Einzelwellen mit gaußverteilten Amplituden

Durch die Beleuchtung mit anders geformten Spektren können Nebenmaxima im Korrelogramm auftreten. Aufgrund der Eigenschaften der Korrelationsfunktion von Objekt- und Referenzstrahl ergibt sich immer eine symmetrische Einhüllende mit einem Maximum am Ort gleicher Weglänge.

Dispersion führt zu einer Änderung der Phase der Einzelwellen als Funktion ihrer Frequenz. Dadurch kann sich die Position des zentralen Streifens gegenüber der Einhüllenden des Korrelogramms verschieben, und das Signal abgeschwächt werden.

I.4.4 Signalbeurteilung

Bei der Einrichtung des Messobjekts wird die Qualität des Messsignals bewertet. Das Messsignal soll den Wertebereich der Kamera möglichst gut ausnützen, darf jedoch nicht an den Grenzen des Wertebereichs abgeschnitten werden. Das Abschneiden (clipping) des Signals verändert die Einhüllende des Korrelogramms, sodass die Mehrzahl der in Kapitel I.5 vorgestellten Auswertemethoden den zentralen Streifen nicht korrekt auffinden kann. Ist die Amplitude des Korrelogramms sehr klein, tritt das Signal nicht genügend aus den Rauschen hervor und kann somit nicht ausgewertet werden.

Die Detektion von Clipping gestaltet sich einfach. Eine Abfolge von Maximal- oder Minimalwerten, die länger als ein Schwellwert ist, wird als Clipping interpretiert. Diese Detektion kann im Zuge der Messung erfolgen und benötigt nur wenig Speicherplatz.

Zur Bewertung der Aussteuerung des Signals kann die SNR (signal to noise ratio) herangezogen werden. Diese ist definiert als

$$SNR = \frac{P_{\text{Signal}}}{P_{\text{Rauschen}}} \quad (\text{I.4.1})$$

beziehungsweise als

$$SNR = \frac{\mu}{\sigma} \quad (\text{I.4.2})$$

wobei μ den Mittelwert und σ die Standardabweichung bezeichnet. Nach beiden Definitionen ist zur Berechnung der SNR das gesamte Messsignal notwendig. Aufgrund der großen Datenmengen ist dieser Ansatz nicht zielführend.

Da im Mirau-Objektiv der Objekt- und der Referenzstrahl mit Hilfe eines Strahlteilers gebildet werden, sind deren Intensitäten bei vollständiger Reflexion am Messobjekt und ohne Berücksichtigung von Streuung gleich groß (siehe Kapitel I.4.1). Sie summieren sich zur Gesamtintensität:

$$\begin{aligned} I_{ges} &= I_R + I_O \\ I_R = I_O &= \frac{I_{ges}}{2} \end{aligned} \quad (\text{I.4.3})$$

Da der Referenzstrahl innerhalb des Objektivs verläuft, erfährt nur der Objektstrahl Intensitätsänderungen.

In Kapitel I.3.2.1 wurde die im Folgenden nochmals wiedergegebene Kohärenzfunktion

$$I = I_O + I_R + 2\sqrt{I_O I_R} m_{12}(\tau) \cos[\alpha_{12}(\tau) - 2\pi \bar{v} \tau] \quad (\text{I.4.4})$$

als Modell des Korrelogramms hergeleitet. Das Korrelogramm macht üblicherweise nur einen kleinen Teil des Messsignals aus. Nur im Bereich des Korrelogramms ist der Korrelationsgrad m_{12} größer Null und lässt dort die Modulation des geometrischen Mittels der Teilstrahlen zu. Im zentralen Streifen ergibt sich daher die Intensität zu

$$I = I_O + I_R + 2\sqrt{I_O I_R} \quad (\text{I.4.5})$$

Im übrigen Teil des Messsignals ergibt sich die Intensität zu

$$I = I_O + I_R \quad (\text{I.4.6})$$

Wird der Objektstrahl nicht abgeschwächt, ergibt sich im Mittel über das gesamte Messsignal die Intensität

$$I_{mittel} = I_O + I_R = I_{ges} \quad (\text{I.4.7})$$

und im Maximum des zentralen Streifens

$$I_{max} = I_O + I_R + 2\sqrt{I_O I_R} = I_{ges} + 2\sqrt{\frac{I_{ges}^2}{4}} = 2I_{ges} \quad (\text{I.4.8})$$

Wird jedoch kein Licht des Objektstrahls vom Objektiv aufgenommen, ergibt sich im Mittel

$$I_{mittel} = 0 + I_R = \frac{I_{ges}}{2} \quad (\text{I.4.9})$$

und im Maximum des zentralen Streifens

$$I_{max} = 0 + I_R + 2\sqrt{I_R \cdot 0} = I_{ges} + 0 = \frac{I_{ges}}{2} \quad (\text{I.4.10})$$

Die mittlere Intensität kann durch die Abschwächung x des Objektstrahls ausgedrückt werden als

$$I_{mittel} = I_O + xI_R = \frac{I_{ges}}{2} + \frac{xI_{ges}}{2} = \left(\frac{1+x}{2}\right) I_{ges} \quad (\text{I.4.11})$$

Die Intensität im Maximum des zentralen Streifens kann analog beschrieben werden als

$$I_{max} = I_O + xI_R + 2\sqrt{I_O xI_R} = \frac{I_{ges}}{2} + \frac{xI_{ges}}{2} + 2\sqrt{\frac{xI_{ges}^2}{4}} = I_{ges} \frac{(\sqrt{x}+1)^2}{2} \quad (\text{I.4.12})$$

Die Gleichungen (I.4.11) und (I.4.9) können zu

$$I_{max} = \sqrt{2 I_{mittel} I_{ges} - I_{ges}^2} + I_{mittel} \quad (\text{I.4.13})$$

vereint werden. Es ist eine Proportionalität zwischen der mittleren Intensität und der maximalen Intensität im zentralen Streifen des Korrelogramms erkennbar.

I.4 Umsetzung des Messprinzips

Alternativ zur SNR wird daher als Bewertungsmaß das Verhältnis

$$V = \frac{I_{max}}{I_{mittel}} \quad (I.4.14)$$

eingeführt. Das Verhältnis V bewegt sich im Wertebereich $1 \leq V \leq 2$. Bei $V = 2$ wird der Wertebereich am besten genutzt. Bei $V = 1$ tritt keine Interferenz auf. Im Zuge einer Testmessung können Maximalwert und Mittelwert einfach bestimmt werden.

I.4.5 Zusammenfassung

Das Messprinzip der interferometrischen Messung der Entfernungsänderung wird durch das Mirau-Objektiv umgesetzt. Dieses erzeugt ein Interferenzsignal, indem die Weglänge eines intern geleiteten Strahls mit dem eines externen Strahls verglichen wird.

Das Messgerät besteht im wesentlichen aus einem Piezotranslator, der das Messobjekt durch den Interferenzbereich des Objektivs bewegt, einer Kamera, welche die Intensitätsänderungen des Interferenzsignals (Korrelogramm) aufzeichnet und dem Mirau-Objektiv. Das Messgerät kann mit Standardkomponenten aufgebaut werden.

Das gemessene Interferenzsignal ist eine positive, gerade Funktion mit linearer Phase und einem Maximum an jener Stelle, an der die Strahlen die gleiche Weglänge aufweisen. Diese Eigenschaften ermöglichen die Ermittlung der Topografie des Messobjekts.

Das Verhältnis V von maximaler Intensität zum Mittelwert der Intensität kann als Maß für die Qualität des Signals verwendet werden. Ist V gleich eins, ist das Maximum gleich dem Mittelwert, daher kann kein Korrelogramm im Signal detektiert werden. Ist V gleich zwei, ist eine maximale Auslenkung des zweifachen Mittelwerts im Signal vorhanden. Das Korrelogramm hat in diesem Fall die maximale Auslenkung.

Im nächsten Kapitel werden einige Methoden zur Auswertung des Korrelogramms vorgestellt.

I.5 Signalverarbeitung

Im vorherigen Kapitel wurden einige praktische Aspekte des Messsignals erörtert. In diesem Kapitel werden Methoden vorgestellt, mit denen die Position der maximalen Interferenz im Messsignal eines jeden Bildelements der Kamera bestimmt werden kann. Häufig werden folgende Methoden eingesetzt:

- *Schwerpunkt-Methode*: Diese berechnet den Schwerpunkt des Korrelogramms, der meist am Ort des zentralen Streifens liegt.
- *Phase-Shifting-Interferometry-Methoden*: Diese bestimmen die Phasen- und damit die Wegdifferenz der optischen Pfade des Objekt- und Referenzstrahls im Zeitbereich.
- *Kohärenz-Methode*: Diese basiert auf der Suche nach dem Scheitelpunkt der Einhüllenden des Korrelogramms.
- *Frequency-Domain-Analysis-Methode*: Diese bestimmt die Phasendifferenz des Objekt- und Referenzstrahls im Frequenzbereich.

Aufgrund des hohen Ressourcenbedarfes werden ebenfalls vorgestellte Verfahren zur Zeit-Frequenz-Analyse nicht in konventionellen Messgeräten angewendet.

Die Werte der Verschiebungen in z-Richtung an den Positionen maximaler Interferenz ermöglichen die Erstellung einer topografischen Karte des Messobjekts.

I.5.1 Schwerpunkt-Methode

Dieser in [15] und [16] beschriebene Algorithmus wird auch als „centroid“ oder „center of mass“ (COM) bezeichnet und basiert auf der Berechnung des Schwerpunktes des Messsignals durch:

$$z_0 = \frac{\sum z_i I_i^2}{\sum I_i^2} \quad (\text{I.5.1})$$

Hier bezeichnen z_0 die Position des Schwerpunktes, z_i die Abtastpositionen und I_i die Abtastwerte⁴.

Aufgrund der Form des Korrelogramms ist die Position des Schwerpunktes gleich der Position maximaler Kohärenz, wenn keine Dispersionseffekte auftreten. Durch Dispersion ändert sich die Phase eines Signals mit der Frequenz. Dadurch verändert sich die Position der maximalen Interferenz gegenüber der Einhüllenden. Verschiedene Ursachen von Dispersionseffekten werden in Kapitel I.6 besprochen.

Signale mit Abtastintervall-Fehlern können ebenso wie Korrelogramme, deren Spitzenwerte in Sättigung gehen, durch diese Methode gut ausgewertet werden. Laut [17] ist die Schwerpunkt-Methode allerdings empfindlich gegen Rauschen und daher kein robuster Schätzer. Auch Ausreißer außerhalb des Korrelogramms können einen erheblichen Einfluss auf das Ergebnis nehmen.

Diese sehr einfache Methode benötigt nur wenig Speicherplatz und Rechenzeit und wird sehr häufig angeboten.

4 In [15] wird statt dem Quadrat der Intensität deren Betrag verwendet.

I.5.2 PSI-Methoden

In Anlehnung an ([18], Kap. 11.3) wird in diesem Unterkapitel die Phase-Shifting-Interferometry (PSI) beschrieben.

PSI gehört zur Gruppe der Temporal-Phase-Modulating-Interferometry-Methoden. Die Phase der Lichtquelle wird hier zwischen den Abtastungen um einen definierten Wert verändert. Mit den gemessenen Intensitätswerten werden die Parameter eines Korrelogramm-Modells geschätzt, und so die Phase ermittelt. Der Eindeutigkeitsbereich dieses Verfahrens ist auf eine Wellenlänge beschränkt. Die gewonnene Phaseninformation muss um Phasensprünge bereinigt werden.

Die Gültigkeit des verwendeten Modells

$$I = a + b \cos(\Phi + \alpha) \quad (I.5.2)$$

ist auf monochromatische Beleuchtung oder den Bereich des zentralen Streifens bei breitbandiger Beleuchtung beschränkt. Da die Phasenmodulation α bekannt ist, werden zumindest drei Gleichungen benötigt, um die drei Unbekannten a , b und Φ zu bestimmen. Mit drei zwischen 0 und π wählbaren, aufeinanderfolgenden Phasenänderungen α_1 , α_2 und α_3 ergeben sich die folgenden Gleichungen:

$$\begin{aligned} I_1 &= a + b \cos(\Phi + \alpha_1) \\ I_2 &= a + b \cos(\Phi + \alpha_2) \\ I_3 &= a + b \cos(\Phi + \alpha_3) \end{aligned} \quad (I.5.3)$$

Durch Auflösung nach Φ ergibt sich:

$$\Phi = \frac{(I_2 - I_3) \cos \alpha_1 - (I_1 - I_3) \cos \alpha_2 - (I_1 - I_2) \cos \alpha_3}{(I_2 - I_3) \sin \alpha_1 - (I_1 - I_3) \sin \alpha_2 - (I_1 - I_2) \sin \alpha_3} \quad (I.5.4)$$

Bei einer Phasenverschiebung von $\pi/2$ pro Messwert vereinfacht sich (I.5.4) zu:

$$\Phi = \tan^{-1} \left(\frac{I_2 - I_3}{I_2 - I_1} \right) \quad (I.5.5)$$

Gleichung (I.5.2) kann mit

$$\begin{aligned} a_0 &= a = I_0 \\ a_1 &= b \cos(\Phi) \\ a_2 &= -b \sin(\Phi) \end{aligned} \quad (I.5.6)$$

nach

$$I_i = a + b \cos(\Phi + \alpha_i) = a_0 + a_1 \cos \alpha_i + a_2 \sin \alpha_i \quad (I.5.7)$$

umgeformt werden. Mit (I.5.7) kann ein überbestimmtes Gleichungssystem gebildet werden, um die Parameter a_1 , a_2 und a_3 zu schätzen. Mit diesen Parametern können für die Phase Φ und die Sichtbarkeit V die Ausdrücke

$$\begin{aligned} \Phi &= \tan^{-1} \left(\frac{-a_2}{a_1} \right) \\ V &= \frac{b}{a} = \frac{\sqrt{a_1^2 + a_2^2}}{a_0} \end{aligned} \quad (I.5.8)$$

angegeben werden. Analog zur Signalbeurteilung in Kapitel I.4.4 kann mit der Sichtbarkeit V die Qualität des Signals für das verwendete Modell in (I.5.2) beurteilt werden.

Aus (I.5.7) und (I.5.8) können mehrere Techniken zur Phasenschätzung mit drei, vier oder fünf Abtastwerten abgeleitet werden. In ([18], Kap. 11) werden sogenannte „Three-Frame“ bis „Five-Frame“ Techniken vorgestellt. Diese benötigen ein konstantes Abtastintervall, dessen Breite durch die Technik vorgegeben ist.

In ([18], Kap. 11.3) wird auch eine nach Carré benannte Technik besprochen. Im Gegensatz zu den zuvor angeführten PSI-Techniken ist hier die Phasenverschiebung zwischen den Abtastungen nicht durch die Technik vorgegeben. Nur die Konstanz der Phasenverschiebung während vier Abtastungen wird vorausgesetzt. Die Phase kann nach Carré durch

$$\Phi = \tan^{-1} \left(\frac{\sqrt{[(I_1 - I_4) + (I_2 - I_3)][3(I_2 - I_3) + (I_1 - I_4)]}}{(I_2 + I_3) - (I_1 + I_4)} \right) \quad (\text{I.5.9})$$

berechnet werden.

Mit Hilfe einer zusätzlichen Abtastung können Nichtlinearitäten und Kalibrationsfehler korrigiert werden. Diese Verfahren werden in [19] hergeleitet und als „(N+1)-Techniken“ bezeichnet. In [20] werden PSI-Techniken mit Hilfe der Monte Carlo Methode untersucht. Die geringste Messunsicherheit wird hier von der Technik „Symmetrisch 5+1“ erreicht. Diese Technik kompensiert verschiedene Messunsicherheitseinflüsse wie Vibration, Quantisierungsfehler und Kalibrationsfehler des Piezotranslators. Die Phase wird von „Symmetrisch 5+1“ durch

$$\Phi = \tan^{-1} \left(\frac{-\sum_{n=1}^{P=6} \sin\left(\frac{2\pi}{5}(p-1)\right) i_p}{\frac{1}{2}(i_1 + i_6) + \sum_{p=2}^5 \cos\left(\frac{2\pi}{5}(p-1)\right) i_p} \right) \quad (\text{I.5.10})$$

berechnet.

Eine ähnliche, jedoch analytische Untersuchung von PSI-Methoden wird in [21] durchgeführt, mit dem Ergebnis, dass eine gute Kompensation systematischer Fehler die statistischen Eigenschaften der PSI-Methoden verschlechtert und umgekehrt.

In [22] beschreibt Phillion eingehend, wie PSI-Methoden erzeugt werden können, die auf Unempfindlichkeit gegenüber bestimmten Fehlern wie Intensitätsschwankungen, Vibrationen bestimmter Frequenzen oder Nichtlinearitäten des Sensors optimiert sind.

PSI-Methoden wurden ursprünglich zur Bestimmung der Topografie bei monochromatischer Beleuchtung entwickelt. Bei breitbandiger Beleuchtung können PSI-Methoden verwendet werden, um die Phase im Bereich des zentralen Interferenzstreifens des Korrelogramms zu ermitteln, oder einen Quadratur-Filter (siehe Kapitel I.5.3.4) zu erzeugen. PSI-Methoden können sehr effizient implementiert werden und dabei eine Auflösung von unter einem Nanometer erreichen.

I.5.3 Kohärenz-Methode

Mit Hilfe der Kohärenz-Methode (coherence peak evaluation) wird die Scheitelpunkt-Position der Einhüllenden des Korrelogramms gesucht. Ohne Dispersionseffekte liegt der Scheitelpunkt an der Position maximaler Interferenz.

Da das Messsignal als Amplitudenmodulation (AM) aufgefasst werden kann, können Verfahren der AM-Demodulation eingesetzt werden, um die Einhüllende zu berechnen.

Mit den nachstehend vorgestellten Verfahren zur Demodulation kann das Maximum der Einhüllenden an einem Abtast(zeit)punkt gefunden werden. Wie in [17] beschrieben, kann die Position des wahren Scheitelpunkts durch Approximation mit einer Gaußkurve angenähert werden.

I.5 Signalverarbeitung

I.5.3.1 Fourier-Transformations-Verfahren

Nach [23] wird bei diesem Verfahren das gesamte Zeitsignal in den Frequenzbereich transformiert und anschließend nur das positive Seitenband in das Basisband verschoben. Durch Rücktransformation wird das Zeitsignal der Einhüllenden gewonnen.

Diese Vorgehensweise ist speicher- und rechenintensiv, da der gesamte Bildstapel im Speicher gehalten, sowie eine Transformation in den Frequenzbereich und eine Rücktransformation für jedes Bildelement berechnet werden müssen.

I.5.3.2 Square-Law-Verfahren

Caber verwendet in [24] dieses Verfahren. Ein amplitudenmoduliertes Signal kann durch

$$\begin{aligned} s(t) &= A(1 + m \cos(\omega_m t)) \cos(\omega_c t) \\ s(t) &= A \cos(\omega_c t) + \frac{mA}{2} \cos(\omega_c t + \omega_m t) + \frac{mA}{2} \cos(\omega_c t - \omega_m t) \end{aligned} \quad (\text{I.5.11})$$

beschrieben werden. Der Modulationsindex wird mit m , die Amplitude mit A , die Trägerfrequenz mit ω_c und die Modulationsfrequenz mit ω_m bezeichnet.

Das Signal wird mit einem geeigneten Bandpass gefiltert und anschließend quadriert. Es ergeben sich ein Gleichanteil und Frequenzen bei $\omega_c \pm \omega_m$, $2(\omega_c \pm \omega_m)$, $2\omega_c$, $2\omega_m$ und ω_m . Mit einem geeigneten Tiefpass-Filter kann nun das in das Basisband transformierte Nutzsignal extrahiert werden.

Dieses sehr einfache Verfahren benötigt je einen Band- und einen Tiefpass-Filter pro Bildelement. Werden diese als FIR-Filter implementiert, kann ein erheblicher Speicherbedarf entstehen, da scharfe Filter viele Koeffizienten benötigen und entsprechend viele Bilder im Speicher gehalten werden müssen.

I.5.3.3 Hilbert-Transformations-Verfahren

Bei diesem Demodulationsverfahren wird ein analytisches Signal gebildet und durch Betragsbildung die Einhüllende berechnet.

Wie in ([25], Kap. 9.6.2) beschrieben, kann ein analytisches Signal mit Hilfe der Hilbert-Transformation gebildet werden. In Abbildung I.5.1 ist die Vorgehensweise zur Erzeugung eines analytischen Signals dargestellt.

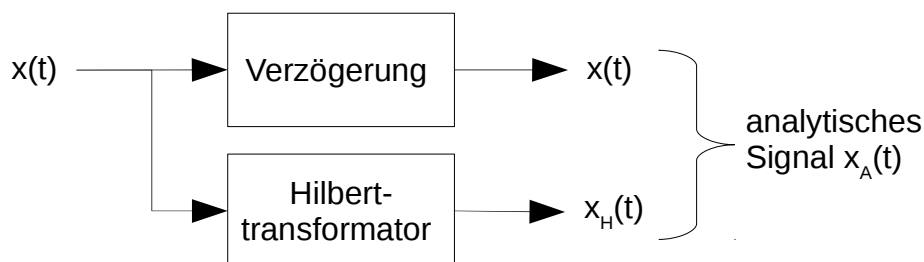


Abbildung I.5.1: Blockdiagramm zur Erzeugung eines analytischen Signals

Ein reelles Signal wird durch den Hilbert-Transformator um $\pi/2$ verschoben, sodass ein analytisches Signal $x_A(t)$ aus dem verzögerten Originalsignal $x(t)$ und dem transformierten Signal $x_H(t)$ erzeugt werden kann:

$$x_A(t) = x(t) + jx_H(t) = |X_A(t)| e^{j\Phi(t)} \quad (\text{I.5.12})$$

Die Einhüllende ergibt sich als Betrag des analytischen Signals:

$$|x_A(t)| = x_A(t) x_A^*(t) = \sqrt{x^2(t) + x_H^2(t)} \quad (\text{I.5.13})$$

Die momentane Phase ergibt sich zu:

$$\Phi(t) = \tan^{-1} \left[\frac{x_H(t)}{x(t)} \right] \quad (\text{I.5.14})$$

Die Hilbert-Transformation kann durch einen FIR-Filter angenähert werden, dessen Impulsantwort durch

$$h[n] = \begin{cases} \frac{2}{\pi} \frac{\sin^2(\pi n/2)}{n} & , n \neq 0 \\ 0 & , n = 0 \end{cases} \quad (\text{I.5.15})$$

berechnet werden kann. Der so erzeugte FIR-Filter hat eine lineare Phase und eine beinahe konstante Magnitude über einen sehr weiten Frequenzbereich.

Durch die hohe Bandbreite des Hilbert-Transformators nimmt auch Rauschen aus nicht relevanten Frequenzbereichen Einfluss auf die erzeugte Einhüllende. Durch vorhergehende Bandpass-Filterung oder die Reduktion der Bandbreite des Hilbert-Transformators kann dieses Problem verringert werden.

Der Rechenbedarf dieses Verfahrens ist gering, der Speicherbedarf ist abhängig vom FIR-Filter.

I.5.3.4 Quadratur-Filter-Verfahren

Mit Hilfe dieses Verfahrens können aus einem Eingangssignal zwei Signale mit einer Phasendifferenz von $\pi/2$ erzeugt werden. Diese bilden den Real- und den Imaginärteil eines analytischen Signals, das zur Demodulation verwendet wird.

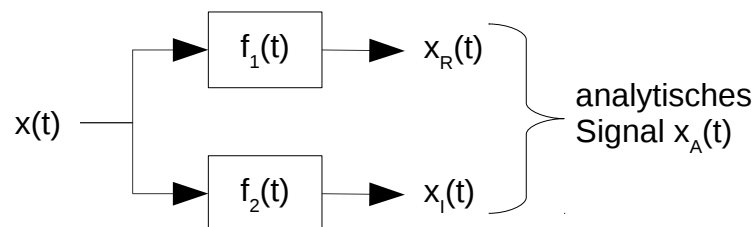


Abbildung I.5.2: Blockdiagramm eines Quadratur-Filters

I.5 Signalverarbeitung

In Abbildung I.5.2 wird diese Vorgehensweise in einem Blockdiagramm veranschaulicht. Die Filter f_1 und f_2 verschieben im Durchlassbereich das Eingangssignal um $-\pi/4$ und $+\pi/4$, sodass die Ausgangssignale eine relative Phasendifferenz von $\pi/2$ aufweisen. Die Ausgangssignale $x_R(t)$ und $x_I(t)$ bilden ein analytisches Signal $x_A(t) = x_R(t) + jx_I(t)$.

Zur Bildung der benötigten Filter können die Methoden der Phase-Shifting-Interferometry (siehe Kapitel I.5.2) als Filter formuliert werden. In [17] wird die fünfwertige PSI-Methode von Hariharan [26] zur Bestimmung der momentanen Phase verwendet:

$$\Phi = \tan^{-1} \left(\frac{2(I_2 - I_4)}{-I_1 + 2I_3 - I_5} \right) = \tan^{-1} \left(\frac{x_I(t)}{x_R(t)} \right) \quad (\text{I.5.16})$$

Der Dividend aus (I.5.16) wird als der imaginäre und der Divisor als der reale Teil eines analytischen Signals aufgefasst. Die entsprechenden FIR-Filter f_1 und f_2 ergeben sich somit mit den Verzögerungselementen δ zu:

$$\begin{aligned} h(t) &= f_1(t) + jf_2(t) \\ f_1(t) &= -\delta(t - 2\Delta) + 2\delta(t) - \delta(t + 2\Delta) \\ f_2(t) &= 2[\delta(t - \Delta) - \delta(t + \Delta)] \end{aligned} \quad (\text{I.5.17})$$

Durch Faltung der komplexen Filter-Funktion $h(t)$ mit einem Modell des Korrelogramms für monochromatische Beleuchtung

$$I_t = a + b \cos(\Phi + \omega_0 t) \quad (\text{I.5.18})$$

ergibt sich wiederum die momentane Phase zu:

$$\Phi = \tan^{-1} \left[\frac{\Im\{[h * I](0)\}}{\Re\{[h * I](0)\}} \right] = \tan^{-1} \left(\frac{2(I_{-1} - I_1)}{-I_{-2} + 2I_0 - I_2} \right) \quad (\text{I.5.19})$$

Mit Hilfe des Real- und Imaginärteils kann die Einhüllende berechnet werden:

$$|x_A(t)| = x_A(t) x_A^*(t) = \sqrt{x_R^2(t) + x_I^2(t)} \quad (\text{I.5.20})$$

Die Fourier-Transformation von f_1 und f_2 lässt sich mit

$$H(e^{j\omega}) = \sum_{n=-(N-1)/2}^{(N-1)/2} h[n] e^{-jn\omega} \quad (\text{I.5.21})$$

bilden:

$$\begin{aligned} F_1(e^{j\omega}) &= 2(e^{-j-1\omega} - e^{-j1\omega}) = 4i \sin(\omega) \\ F_2(e^{j\omega}) &= e^{-j-2\omega} - 2e^{-j0\omega} - e^{-j2\omega} = 2 - 2\cos(2\omega) \end{aligned} \quad (\text{I.5.22})$$

Die fouriertransformierten F_1 und F_2 aus (I.5.22) sind in Abbildung I.5.3 zu sehen. Wird eine Phasendifferenz von $\pi/2$ zwischen den Abtastwerten des Trägersignals eingehalten, befindet sich der Schwerpunkt des Spektrums des Korrelogramms im Bereich um $\pm\omega/2$. Die Verstärkung in diesem Bereich ändert sich nur langsam. Daraus resultiert laut [17] eine gewisse Unempfindlichkeit gegen Abtastintervall-Fehler und additives Rauschen.

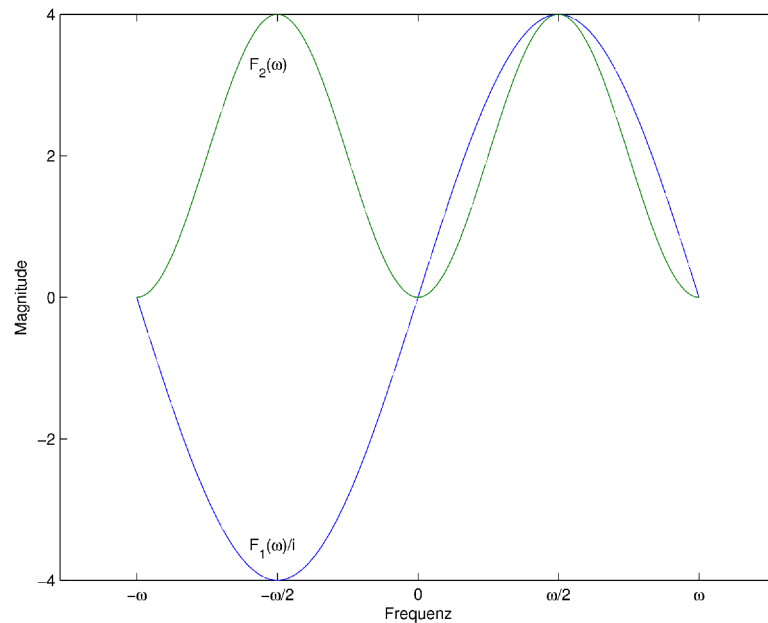


Abbildung I.5.3: Fourier-Transformation der fünfwertigen Filter aus Formel (I.5.22)

Analog kann mit Hilfe anderer PSI-Methoden unter anderem das benötigte Abtastintervall verändert werden.

Da PSI-Methoden auf geringen Speicher- und Rechenbedarf optimiert sind, haben auch die so gebildeten Quadratur-Filter einen sehr geringen Ressourcenbedarf.

I.5.3.5 Quadratur-Abtastungs-Verfahren

Auch hier wird ein analytisches Signal zur Demodulation verwendet. Diese Vorgehensweise zur Bildung eines analytischen Signals wird in ([27], Kap. 8.9) beschrieben. Aus dem amplitudenmodulierten Eingangssignal werden mit zwei Mixern zwei Signale im Basisband mit einer Phasendifferenz von $\pi/2$ erzeugt. Der nachfolgende Tiefpass-Filter entfernt Anteile außerhalb des Basisbands. Das Blockdiagramm dieser Methode ist in Abbildung I.5.4 zu sehen.

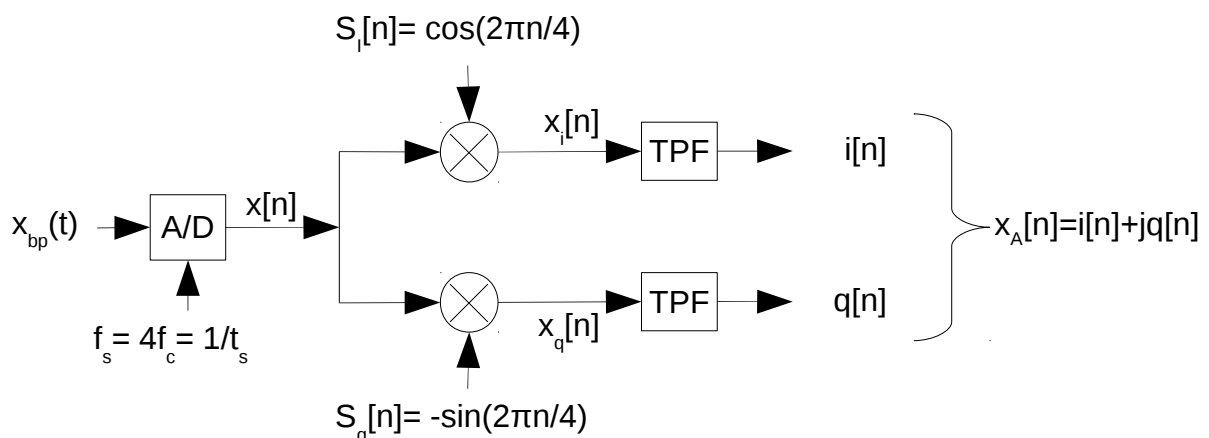


Abbildung I.5.4: Blockdiagramm Quadratur-Abtastung nach ([27], Kap. 8.9)

I.5 Signalverarbeitung

Ist die Trägerfrequenz f_c exakt bekannt, bewirkt diese Vorgehensweise eine Verschiebung der Signale $i(n)$ und $q(n)$ um genau $\pi/2$ gegeneinander. Bei Abtastung mit einer Abtastfrequenz von $f_s = 4f_c$ ergeben sich die Misch-Funktionen S_i zu $[1,0,-1,0]$ und S_q zu $[0,-1,0,1]$. Die Implementierung des Mixers erfordert somit nur Vorzeichenwechsel.

In Abbildung I.5.5 sind die Spektren des Signals im Verlauf der Verarbeitung skizziert.

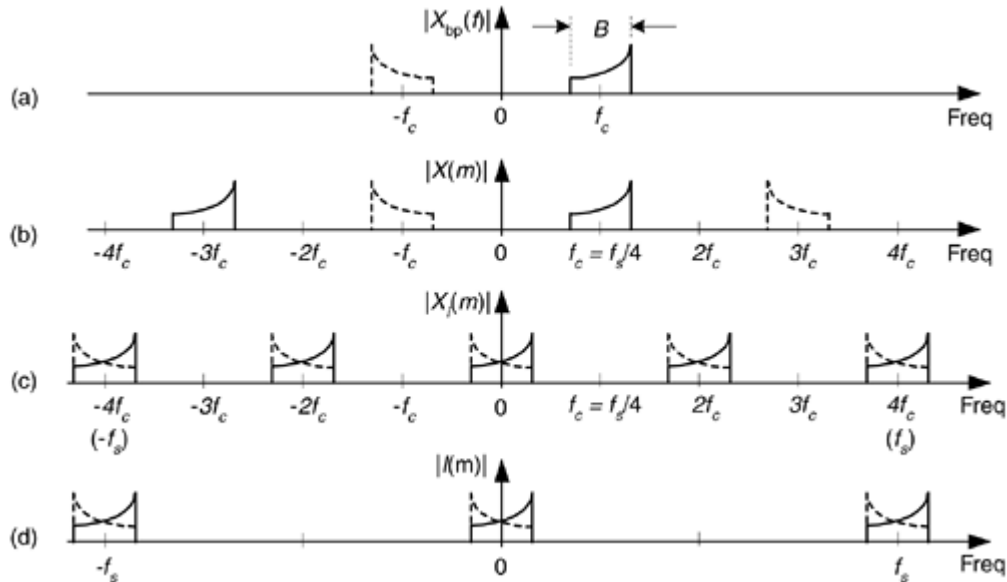


Abbildung I.5.5: Spektrale Darstellung der Quadratur-Abtastung mit digitalem Mischen. (a) Spektrum des Eingangssignals; (b) Spektrum nach Abtastung; (c) Spektrum nach Mischen; (d) Spektrum nach Tiefpass-Filterung [27]

Abbildung I.5.5 (a) zeigt das Spektrum des Eingangssignals. Nach der Abtastung in (b) sind Replika um Vielfache der Abtastfrequenz f_s zu sehen. In (c) bewirkt das Mischen mit Cosinus bzw. Sinus der Frequenz f_c eine Verschiebung der Spektren ins Basisband. Nach Tiefpass-Filterung bleibt in (d) nur das Signal im Basisband.

Um Mehrdeutigkeiten zu verhindern, muss die Bandbreite B des Signals kleiner als die Trägerfrequenz f_c bzw. ein Viertel der Abtastfrequenz f_s sein:

$$B < f_c = f_s/4 \quad (I.5.23)$$

Die Energie des Ursprungssignals wird auf $i(n)$ und $q(n)$ aufgeteilt. Die Signale $i(n)$ und $q(n)$ bilden den Real- und Imaginärteil eines analytischen Signals und können mit

$$|x_A(n)| = x_A(n) x_A^*(n) = \sqrt{i^2(n) + q^2(n)} \quad (I.5.24)$$

zur Berechnung der Einhüllenden verwendet werden.

Werden die Mischer in die Tiefpass-Filter integriert, ergibt sich im einfachsten Fall genau der in Kapitel I.5.3.4 beschriebene Quadratur-Filter nach Larkin.

Die gezeigte Bildung des analytischen Signals erlaubt eine hohe Flexibilität bei der Wahl der Abtastfrequenz und des Tiefpass-Filters und kann so dem erwarteten Signal gut angepasst werden.

Der Rechenbedarf ist hier gering, der Speicherbedarf ist vom Tiefpass-Filter abhängig.

I.5.4 Frequency-Domain-Analysis-Methode

De Groot beschreibt in [28] und [29] eine Methode, mit der das Korrelogramm im Frequenzbereich ausgewertet werden kann.

Werden die OPD des Objekt- und Referenzstrahls mit Z und die Wellenzahl mit $k = 2\pi/\lambda$ bezeichnet, so kann die Phase des Korrelogramms als

$$\Phi = kZ \quad (\text{I.5.25})$$

ausgedrückt werden. In [28] wird Z als „phase-velocity OPD“, also Phasengeschwindigkeit-OPD, in Anlehnung an die Phasengeschwindigkeit bezeichnet. Eine Taylor-Reihenentwicklung an der mittleren Wellenzahl k_0 führt zu:

$$\Phi = \Phi_0 + (k - k_0) \frac{d\Phi}{dk} \Big|_{k_0} + \frac{(k - k_0)^2}{2} \frac{d^2\Phi}{dk^2} \Big|_{k_0} + \dots \quad (\text{I.5.26})$$

Der erste Term in (I.5.26) ist die Phase Φ_0 bei $k=k_0$. Mit einer optischen Wegdifferenz Z_0 ergibt sich Φ_0 zu:

$$\Phi_0 = k_0 Z_0 \quad (\text{I.5.27})$$

Der zweite Term in (I.5.26) wird in [28] als „group-velocity OPD“, also als Gruppengeschwindigkeit-OPD bei $k = k_0$ bezeichnet:

$$G_0 = \frac{d\Phi}{dk} \Big|_{k_0} = Z_0 + k \frac{dZ}{dk} \Big|_{k_0} \quad (\text{I.5.28})$$

Die Gruppengeschwindigkeit-OPD entspricht der Steigung der Phase und ist proportional zur Position des Korrelogramms im Auswertefenster der Fourier-Transformation.

Wie in Kapitel I.4.3 beschrieben, besitzt das Korrelogramm eine lineare Phase, die bei $k = k_0$ zu Null wird. Abbildung I.5.6 stellt diesen Zusammenhang dar. Die Gruppengeschwindigkeit-OPD ist daher konstant, wenn keine Dispersion auftritt.

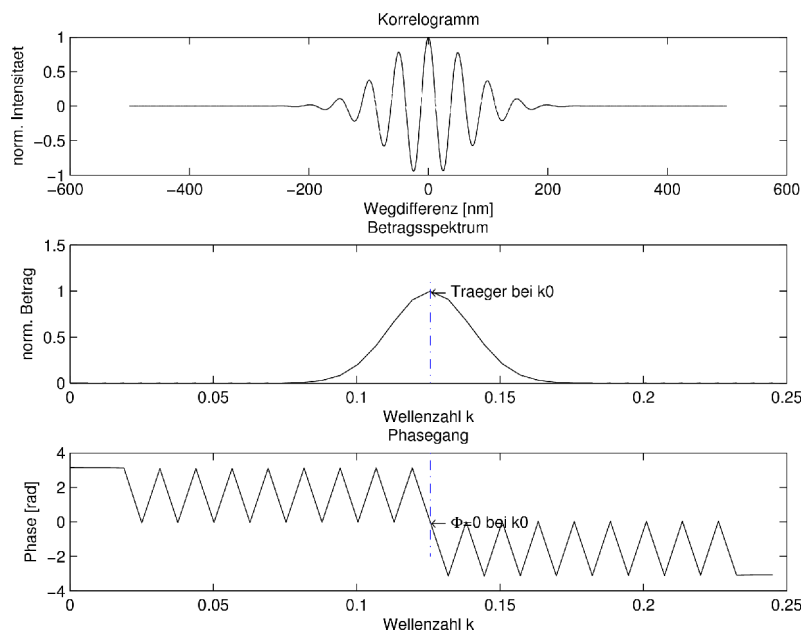


Abbildung I.5.6: Betrag und Phase eines Korrelogramms; an der Frequenz des Trägers ist die Phase gleich Null

Befindet sich das Korrelogramm genau in der Mitte des Auswertefensters, ist die Gruppengeschwindigkeit-OPD gleich Null. In Abbildung I.5.7 wird dieser Zusammenhang dargestellt. Das obere Diagramm der Abbildungen zeigt Korrelogramme an verschiedenen Positionen im Auswertefenster. Das untere Diagramm zeigt die entsprechenden Phasengänge.

I.5 Signalverarbeitung

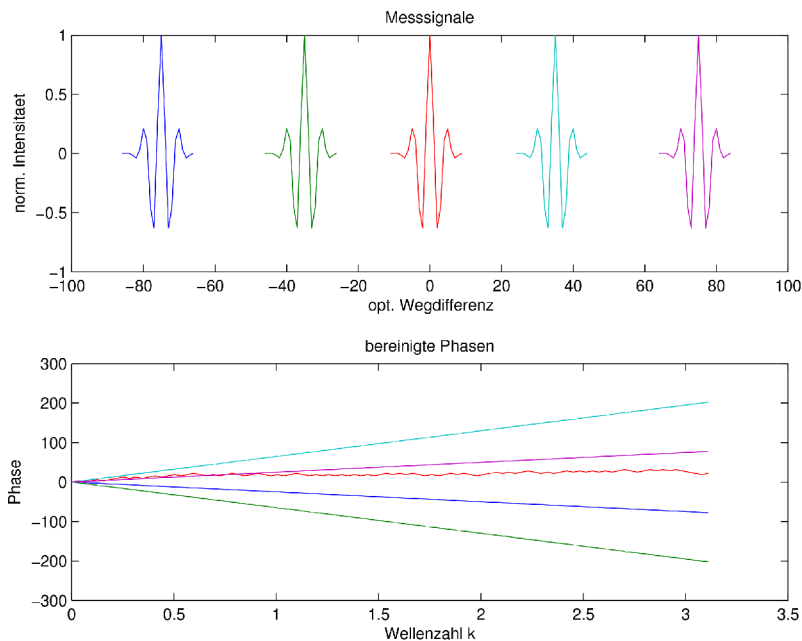


Abbildung I.5.7: oben: Korrelogramme an verschiedenen Positionen im Auswertefenster; unten: Phasengang der Korrelogramme; die Steigung der Phase ist proportional zur Position im Auswertefenster

De Grood nützt in [28] diese Eigenschaften des Messsignals durch folgende Vorgehensweise:

1. Das gesamte Messsignal wird fouriertransformiert.
2. Die Trägerfrequenz wird als Maximalwert des Betragsspektrums des Korrelogramms gesucht und mit der Wellenzahl k_0 bezeichnet.
3. Mit sieben Abtastungen der Phase um k_0 werden, nach Bereinigung von Phasensprüngen, die Steigung und der Achsenabschnitt mit der Methode der kleinsten Quadrate geschätzt.
4. Mit Hilfe der Gruppengeschwindigkeit-OPD wird die Z-Position berechnet.

Die Höhe h des Messobjekts an dem betrachteten Punkt ist proportional der Gruppengeschwindigkeit G_0 an k_0 . Gleichung (I.5.29) gibt diesen Zusammenhang wieder. Der Faktor n_g wird in Anlehnung an die Brechzahl in [28] als „group-velocity index“ bezeichnet.

$$h = \frac{G_0}{2n_g} \quad (\text{I.5.29})$$

Bei dieser Methode muss der gesamte Bildstapel im Speicher gehalten und eine Fourier-Transformation für jedes Bildelement berechnet werden. Diese Methode hat daher einen hohen Speicher- und Rechenbedarf. Dispersionseffekte können die Phasenlage verändern und so zu Messfehlern führen.

I.5.5 Optimalfilter-Methode

Der Optimalfilter (matched filter) kann als eine Autokorrelationsfunktion, die als Filter ausgeführt wird, aufgefasst werden. Das Eingangssignal wird mit dem zu detektierenden, zeitlich gespiegelten Signal gefiltert. Voraussetzung ist die Kenntnis des zu detektierenden Signals.

Nach [30] wird ein Signal umso besser detektiert und lokalisiert, je weniger es sich selbst ähnelt, wenn es verschoben wird. In Abbildung I.5.8 ist unter B die Autokorrelation von A dargestellt. Der Spitzenwert in der Mitte entspricht dem Wert der Autokorrelationsfunktion bei genauer Überdeckung. Die Nebenmaxima sind stark ausgeprägt und erschweren die Lokalisation des Pulses aus A. Im Gegensatz dazu ist der Puls in C sich selbst nur in einem kleinen Bereich der Verschiebung ähnlich. Die Autokorrelationsfunktion von C in D zeigt eine deutlich bessere Lokalisation.

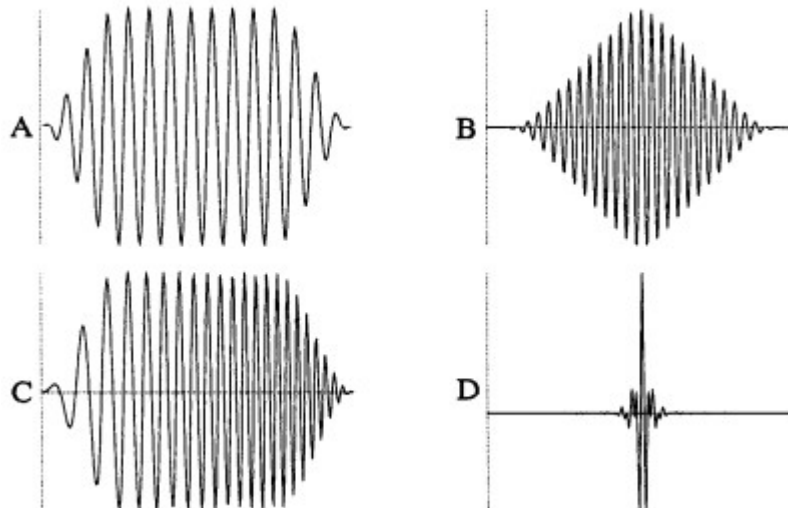


Abbildung I.5.8: Vergleich der Autokorrelation zwischen einem Impuls mit konstanter Frequenz (oben) und einem Frequenz-Chirp (unten). Rechts sind Autokorrelationsfunktionen und links die Ausgangssignale dargestellt. [30]

Das in dieser Arbeit beschriebene Messprinzip (siehe Kapitel I.3) führt zu einem Korrelogramm das dem Signal A aus Abbildung I.5.8 ähnlich ist. Es ist zur Lokalisation durch die Optimalfilter-Methode nicht gut geeignet. Zusätzlich ist die gesuchte Signalform nicht vollständig bekannt. Daher wird der Optimalfilter zur Bestimmung der Topografie nicht eingesetzt.

I.5.6 STFT-Methode

Mit Hilfe der Short-Time-Fourier-Transformation (STFT) kann das Auftreten der Trägerfrequenz des Korrelogramms lokalisiert werden. Dazu wird ein Analysefenster mit dem Signal multipliziert, welches das Signal außerhalb des interessierenden Bereiches unterdrückt. Im Bereich des Fensters kann nun das Spektrum bestimmt werden. Das Analysefenster wird anschließend weiter bewegt und wiederum das Spektrum bestimmt.

Das Auflösungsvermögen wird nach [31] durch das Unsicherheitsprinzip begrenzt:

$$\Delta_t \Delta_f \geq 1/2 \quad (\text{I.5.30})$$

Mit Δ_t wird die zeitliche (örtliche) und mit Δ_f die Frequenzauflösung bezeichnet.

Durch die große Anzahl der benötigten relativ kurzen Fourier-Transformationen ist diese Methode sehr rechenintensiv. Die Größe des Analysefensters bestimmt den Speicherbedarf.

I.5.7 Wavelet-Transformations-Methode

In [32] wird die Analyse des Korrelogramms mit Hilfe der Wavelet-Transformation beschrieben. Diese Vorgehensweise ermöglicht es nicht nur, das Korrelogramm zu lokalisieren, sondern gleichzeitig auch die Trägerfrequenz und die Phase zu bestimmen.

Die Wavelet-Transformation ist definiert durch:

$$W(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} x(t) h^* \left(\frac{t-b}{a} \right) dt \quad (\text{I.5.31})$$

Der Parameter a bestimmt die Skalierung und somit die Frequenz. Der Parameter b verschiebt das Wavelet.

Als Mutter-Wavelet eignet sich das Morlet-Wavelet besonders, da es wie das Korrelogramm als eine gaußmodulierte Schwingung definiert ist:

$$h(t) = \pi^{-1/4} e^{-j\omega_0 t} e^{(-t^2/2)} \quad (\text{I.5.32})$$

Das Tochter-Wavelet mit dem größten Korrelationskoeffizienten stimmt am besten mit dem gesuchten Signal überein. Dieses wird mit $W(a_0, b_0)$ bezeichnet. Die Index-Position im Signal ist b_0 , die Phase an dieser Stelle ist gegeben durch:

$$\phi_0 = \tan^{-1} \left\{ \frac{\Im [W(a_0, b_0)]}{\Re [W(a_0, b_0)]} \right\} \quad (\text{I.5.33})$$

Somit ergibt sich die Position

$$z = b_0 \Delta - \frac{\lambda_0}{4\pi} \phi_0 \quad (\text{I.5.34})$$

wobei Δ das Abtastintervall bezeichnet.

Die Wavelet-Transformation ist rechnerisch aufwändig und benötigt den gesamten Bildstapel im Speicher.

I.5.8 Zusammenfassung

Die in diesem Kapitel beschriebenen Methoden der Signalverarbeitung haben zum Ziel, jenen Wert der Verschiebung des Messobjekts zu finden, an dem für je ein Bildelement die Interferenz maximal ist. Zu diesem Zweck können verschiedene Methoden angewendet werden, die verschiedene Eigenschaften des Korrelogramms nutzen.

Die einfachste Methode mit sehr geringem Ressourcenbedarf ist die Schwerpunkt-Methode (Kapitel I.5.1). Hier wird die Symmetrie des Korrelogramms mit einem zentralen Maximum ausgenützt. Diese Methode ist sehr robust gegen Sättigung des Signals und Abtastintervall-Fehler. Intensitätsrauschen, Ausreißer oder Intensitätsschwankungen können jedoch das Messergebnis stark beeinflussen. Die Schwerpunkt-Methode wird sehr häufig in Weißlicht-Interferometern angeboten.

Die Kohärenz-Methode (Kapitel I.5.3) wird ebenfalls häufig eingesetzt. Dabei wird die Position des Scheitelpunktes der Einhüllenden des Korrelogramms gesucht. Die Berechnung der Einhüllenden kann durch verschiedene AM-Demodulations-Verfahren erfolgen.

Häufig wird zur Demodulation ein analytisches Signal mit Hilfe eines Quadratur-Filters (Kapitel I.5.3.4) gebildet. Die Einhüllende ergibt sich als Betrag des analytischen Signals. Auch das Square-Law-Verfahren (Kapitel I.5.3.2) wird häufig angewendet, um die Einhüllende zu finden.

Beide Verfahren konditionieren das Messsignal, sodass Rauschen, Intensitätsschwankungen und Ausreißer wenig Einfluss auf das Ergebnis haben. Diese Verfahren benötigen jedoch ein ungesättigtes Signal. Um eine Auflösung von wenigen Nanometern zu erreichen, muss der wahre Scheitelpunkt der Einhüllenden durch Approximation einer geeigneten Kurve angenähert werden.

PSI-Methoden (Kapitel I.5.2) werden nur in Kombination mit der Schwerpunkt- oder der Kohärenz-Methode verwendet. Hier wird der zentrale Streifen durch eine dieser Methoden identifiziert und anschließend durch eine PSI-Methode ausgewertet.

Mit PSI-Methoden werden die Parameter eines Modells des Korrelogramms durch Messung der Intensität in - durch die Methode - vorgegebenen Abständen geschätzt. Durch die so ermittelte Phase des Korrelogramms kann die Position des zentralen Streifens sehr genau bestimmt werden. Falls der zentrale Streifen richtig identifiziert wird, kann eine Auflösung im Sub-Nanometer-Bereich erzielt werden.

PSI-Methoden werden durch Intensitätsrauschen aus verschiedenen Quellen beeinträchtigt. Die bei Vermessung optisch rauer Oberflächen entstehenden Speckel können durch ihre zufällige Phase nicht ausgewertet werden. Treten Dispersionseffekte auf, kann der zentrale Streifen möglicherweise nicht richtig identifiziert werden.

Bei der Frequency-Domain-Analysis-Methode (Kapitel I.5.4) wird die Linearität der Phase des Korrelogramms im Frequenzbereich ausgewertet. Dazu wird das gesamte Messsignal in den Frequenzbereich transformiert, und hier die Frequenz mit der größten Amplitude gesucht. Um diese Frequenz werden Steigung und Achsenabschnitt der Phase geschätzt. Die Phase besitzt einen Nulldurchgang am Ort des zentralen Streifens, da dort die maximale konstruktive Interferenz auftritt. Damit kann die Position des zentralen Streifens ermittelt werden. Treten Dispersionseffekte auf, ändert sich die Phase des Korrelogramms. Dadurch kann es zu Messfehlern kommen.

Die Optimalfilter-Methode (Kapitel I.5.5) ist aufgrund der Symmetrie des Korrelogramms nicht gut zur Detektion bzw. zur Lokalisation des Korrelogramms geeignet.

Als weitere Methoden der Zeit-Frequenz-Analyse werden die STFT-Methode (Kapitel I.5.6) und die Wavelet-Transformations-Methode (Kapitel I.5.7) besprochen. Beide Methoden benötigen den gesamten Bildstapel im Speicher. Die entsprechenden Transformationen sind rechenintensiv. Wegen des hohen Ressourcenbedarfs werden diese Methoden nicht kommerziell verwendet.

I.5 Signalverarbeitung

Sehr einfach, robust und hoch auflösend ist die Kombination von Schwerpunkt- und PSI-Methode. Die Schwerpunktmethode liefert auch unter schwierigen Verhältnissen gute Ergebnisse, welche durch eine PSI-Methode noch stark verbessert werden können, falls die Signalqualität dies zulässt. Ein weiterer Vorteil ist der geringe Speicher- und Rechenbedarf.

Im nächsten Kapitel werden Beiträge zur Messunsicherheit der gemessenen Topografie besprochen.

I.6 Messunsicherheitseinflüsse

In diesem Kapitel werden Faktoren beschrieben, welche die Messunsicherheit der Topografie des Messobjekts beeinflussen.

Laut ([11], Nr. 3.6) dient die Messunsicherheit zusammen mit dem Messergebnis zur Kennzeichnung eines Wertebereichs für den wahren Wert der Messgröße.

Die Topografie eines Messobjekts ist die Gesamtheit der Höheninformationen auf einem zweidimensionalen Raster. Das Raster wird durch die Auflösungszellen des Bildsensors gebildet. Die Höheninformation ist die Messgröße. Diese wird an jedem Rasterelement bestimmt. In Abbildung I.6.1 ist die Topografie eines Messobjekts dargestellt. Jeder dargestellten Auflösungszelle ist eine Höhe zugeordnet.

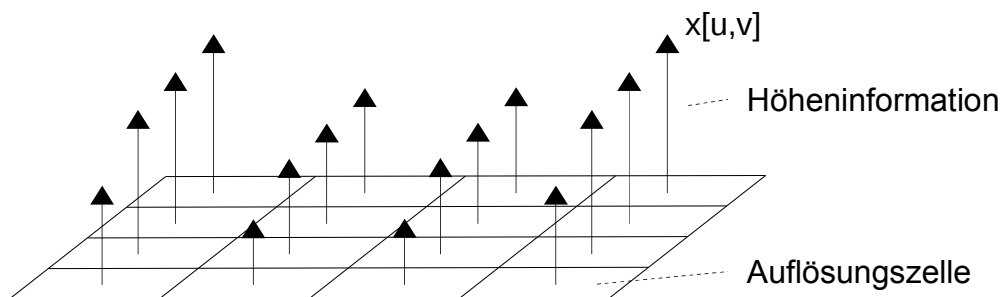


Abbildung I.6.1: Topografie aus Auflösungszellen und Höheninformationen

Die Berechnung der Topografie eines Messobjekts beruht auf Messungen und Konstanten, die mit Unsicherheiten behaftet sind. Die Ursachen der Messunsicherheiten werden gegliedert nach ihrer Lokalisation in Abbildung I.6.2 dargestellt und in den folgenden Unterkapiteln besprochen.

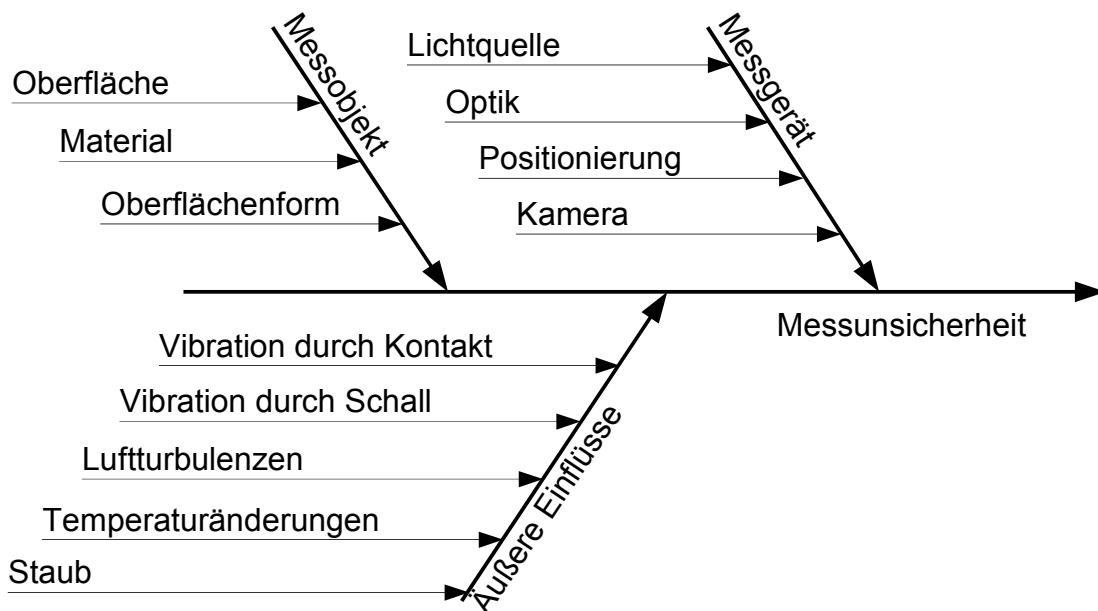


Abbildung I.6.2: Ursachen der Messunsicherheit

I.6.1 Messunsicherheitseinflüsse durch das Messobjekt

In diesem Unterkapitel werden Einflüsse des Messobjekts auf die Messunsicherheit der Topografie besprochen. Da die Messung mit Hilfe von Licht erfolgt, können alle Eigenschaften des Messobjekts, die sich auf die Reflexion auswirken, die Messung und damit auch deren Unsicherheit beeinflussen. Zu diesen Eigenschaften gehören die Oberflächenbeschaffenheit, das Material und die Oberflächenform.

I.6.1.1 Oberfläche

In [33] wird die Oberfläche des Messobjekts als eine Ansammlung von Elementarstreuern angesehen. Jeder Elementarstreuer wird durch das optische System in eine Bildebene abgebildet. Ein Sensor zeichnet mit endlich ausgedehnten Bildelementen in der Bildebene das Messsignal auf. Die Apertur des Objektivs bestimmt die Ausmaße der Auflösungszellen, welche am Messobjekt in die jeweiligen Bildelemente abgebildet werden. Die gemessene Phase und Amplitude in einem Bildelement entsprechen dem Mittelwert der von den Elementarstreuern verursachten Phasenverschiebungen und Amplituden in der korrespondierenden Auflösungszelle.

Variieren die Höhen der Elementarstreuer innerhalb einer Auflösungszelle um nicht mehr als $\pm\lambda/8$, wird in [34] von einer optisch glatten Oberfläche gesprochen. In diesem Fall ist eine konstruktive Interferenz der reflektierten Wellen möglich. Die Messunsicherheit durch die Oberflächenbeschaffenheit ist hier sehr gering.

Sind die Höhenunterschiede der Elementarstreuer in einer Auflösungszelle größer als $\pm\lambda/4$, wird in [34] von einer optisch rauen Oberfläche gesprochen. Es kann hier zu destruktiver Interferenz kommen, da die Phasen der Reflexionen benachbarter Elementarstreuer um mehr als π verschoben sein können. Ist die Verteilung der Höhen der Elementarstreuer zufällig, entstehen subjektive Speckles (kleine helle oder dunkle Flecken) in der Bildebene.

Die Ausdehnung eines subjektiven Speckle ergibt sich zu:

$$R_{\text{Speckle}} = 1,22 \frac{\lambda b}{D} \quad (\text{I.6.1})$$

Es bezeichnen D den Durchmesser der Blende und b den Abstand von Linse zu Detektor.

Die Phase innerhalb eines Speckle ergibt sich, ebenso wie die Intensität, zufällig. Die Standardabweichung der messbaren Höhe innerhalb eines Speckle ergibt sich in [34] zu:

$$\delta z = \frac{1}{\sqrt{2}} \sqrt{\frac{\bar{I}}{I}} \sigma_h \quad (\text{I.6.2})$$

Die Symbole σ_h , \bar{I} und I stehen hier für die Standardabweichung der Höhen, die mittlere Intensität des Speckle-Musters und die Intensität des betrachteten Speckle.

Durch die zufällige Phase werden Auswerteverfahren, die sich auf die Phase stützen, stark beeinträchtigt.

I.6.1.2 Material

Durch das Material des Messobjekts wird das Spektrum des Objektstrahls bei der Reflexion verändert. Diese Änderung des Spektrums wirkt sich auf die Trägerfrequenz und die Amplitude des Korrelogramms aus. Durch die Änderung der Trägerfrequenz können Voraussetzungen von Auswertemethoden, die eine bestimmte Anzahl von Abtastungen pro Wellenlänge benötigen, möglicherweise nicht erfüllt werden.

Verschiedene Materialien erzeugen bei der Reflexion auch verschiedene, von der Wellenlänge abhängige Phasensprünge. In [35] wurden einige Materialien dahingehend untersucht.

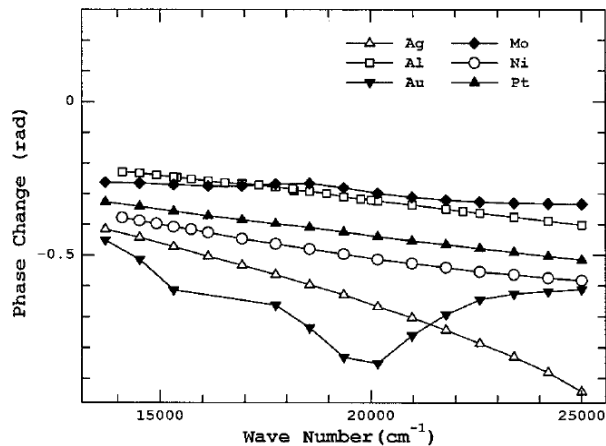


Abbildung I.6.3: Phasenverschiebung nach Reflexion an Metallen als Funktion der Wellenzahl [35]

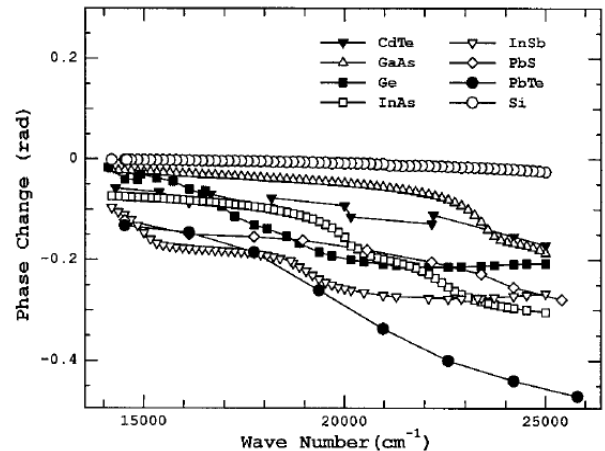


Abbildung I.6.4: Phasenverschiebung nach Reflexion an Halbleitern als Funktion der Wellenzahl [35]

Diese von der Wellenlänge abhängige Phasenänderung verschiebt den Träger des Korrelogramms gegen die Einhüllende. Ergebnisse von Auswertemethoden, die sich auf das Trägersignal stützen, können durch diesen Dispersionseffekt beeinträchtigt werden. Besteht ein Messobjekt aus mehreren Materialien, können die verschiedenen Phasensprünge Höhenunterschiede zwischen den Materialien vortäuschen.

I.6.1.3 Oberflächenform

An Stufen tritt ein Messfehler wie in Abbildung I.6.5 dargestellt auf. Diese Fehlmessung wird „bat wings“ genannt und tritt nach [37] und [38] an Stufen auf, deren Höhen kleiner als die Kohärenzlänge ist. Die Überlagerung von Reflexionen der oberen und unteren Ebene, sowie Beugungseffekte tragen zu diesem Messfehler bei und bewirken eine Verschiebung der Einhüllenden gegen den zentralen Streifen. Methoden, die sich auf die Einhüllende des Korrelogramms stützen und die Schwerpunkt-Methode sind von diesem Effekt stärker betroffen als Verfahren, die sich auf die Phase stützen.

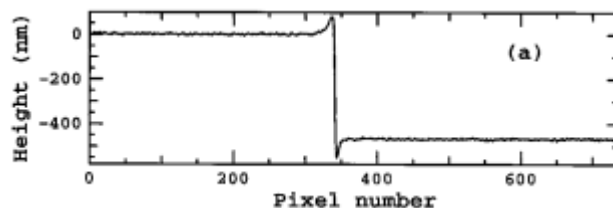


Abbildung I.6.5: fehlerhafte Messung an Stufen („bat wings“) [36]

Auch Stufen, die höher als die Kohärenzlänge sind, werden nicht fehlerfrei gemessen. Hier spielt jedoch nur die Beugung an der Kante eine Rolle. Daher ist der Fehler in diesem Fall sehr viel kleiner.

I.6 Messunsicherheitseinflüsse

Schiefe Ebenen oder die Verkipfung des Messobjekts bis zum halben Aperturwinkel bewirken eine Abschwächung der Intensität des Korrelogramms, da der reflektierte Strahl nicht mehr vollständig vom Objektiv erfasst wird. Bei größeren Neigungen kann die Reflexion nicht mehr erfasst werden. Durch die geringere Intensität verringert sich der Signal-Rausch-Abstand des Korrelogramms. Die Lokalisation des Korrelogramms wird dadurch erschwert.

Nach [34] können Elementarstrahler einer Auflösungszelle auf einer schiefen Ebene wie eine raue Oberfläche wirken, da über die Auflösungszelle gemittelt wird. Durch die Mittelung der Steigung über die Auflösungszelle entsteht eine Stufung, welche die Messunsicherheit ebenfalls erhöht.

Nach [39] können Mehrfachreflexionen in einer Kerbe eine falsche Topografie vortäuschen. Abbildung I.6.6 zeigt eine V-förmige Kerbe mit einem Öffnungswinkel von 70° . Die gestrichelte Linie zeigt die Form der Kerbe nach Herstellerangaben. Die dicke Linie zeigt die Messung. Durch Mehrfachreflexionen wird ein Grat anstelle einer Kerbe gemessen. Aufgrund der Neigung wird an den Flanken kein nutzbares Signal aufgenommen.

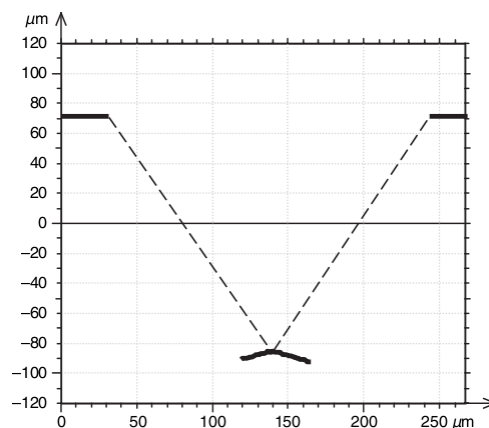


Abbildung I.6.6: Messung einer V-Kerbe. Die dicke Linie zeigt das Messergebnis, die gestrichelte Linie das wahre Profil. [39]

I.6.2 Messunsicherheitseinflüsse durch das Messgerät

In diesem Unterkapitel werden Einflüsse des Messgeräts auf die Messunsicherheit der Topografie besprochen. Jeder Teil des Messgeräts (siehe Kapitel I.4.2) kann zur Messunsicherheit beitragen und wird im Folgenden besprochen. Dabei wird dem Weg des Lichtes durch den Messaufbau gefolgt.

I.6.2.1 Lichtquelle

Das Spektrum und die Intensität realer Lichtquellen sind meist nicht völlig stationär. Strom-, Spannungs- und Temperaturänderungen sowie Alterungsprozesse können sich auf die Intensität und das Spektrum auswirken. Ändert sich das Spektrum, so verändert sich damit auch die Frequenz des Trägers des Korrelogramms. Damit können Voraussetzungen von Signalverarbeitungsmethoden möglicherweise nicht erfüllt werden. Schwankungen der Intensität können die Signalverarbeitung beeinträchtigen.

I.6.2.2 Optisches System

Im Messaufbau besteht das optische System aus dem Mirau-Objektiv und der Einkoppelung der Beleuchtung.

Auch bei einem Mirau-Objektiv (siehe Kapitel I.4.1), dessen sphärische und chromatische Aberration kompensiert sind, kann es abseits der optischen Achse zu Dispersionseffekten kommen. Dispersion verschiebt den Träger gegen die Einhüllende des Korrelogramms und kann somit eine Messabweichung verursachen.

In [37] führt Lehmann diese Dispersionseffekte auf den Aufbau des Mirau-Objektivs zurück. Das Objektiv enthält einen Strahlteiler, durch den ein Referenz- und ein Objektstrahlstrahl erzeugt werden. Der Referenzstrahl durchläuft die obere Glasscheibe des Strahlteilers insgesamt vier mal. Der Objektstrahl muss beide Glasscheiben des Strahlteilers zweimal durchlaufen.

Sind die parallelen Glasscheiben gleich dick, und ist das plane Messobjekt parallel zum Referenzspiegel ausgerichtet, wird von beiden Strahlen derselbe Weg in den Glasscheiben zurückgelegt, wodurch sie derselben Dispersion ausgesetzt sind, und daher kein Fehler auftritt.

Ist jedoch das Messobjekt zur Referenzoberfläche verkippt, legt der Strahl über das Messobjekt abseits der optischen Achse einen längeren bzw. kürzeren Weg in den Glasscheiben zurück als der Referenzstrahl. Somit erfahren die beiden Strahlen unterschiedliche Dispersion. Die resultierende Dispersion ist abhängig von der lokalen Steigung und der Position im Messfeld. Dieser Effekt kann „ghost steps“ verursachen, wenn dadurch der zentrale Streifen nicht richtig identifiziert werden kann, und „bat wings“ beeinflussen. Eine Kompensation ist nur schwer möglich.

I.6.2.3 Positionierung

Im Messaufbau erfolgt die Positionierung entlang der optischen Achse des Objektivs mit einem piezoelektrischen Translator (PZT). Dieser wird mit dem Digital-Analog-Umsetzer (DAU) einer PC-Messkarte über einen Verstärker angesteuert. Jedes Glied in der dargestellten Ansteuerungskette kann zur Unsicherheit der Positionierung des Messobjekts und damit zur Messunsicherheit der Topografie beitragen.

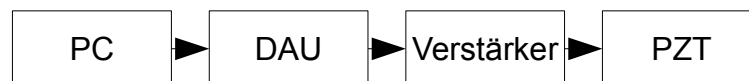


Abbildung I.6.7: Positionierung

Die DAU-Einschubkarte kann Nullpunktsfehler, Linearitätsfehler sowie Quantisierungsrauschen einbringen. Der Verstärker erzeugt Rauschen, Nichtlinearitäten und 1/f-Rauschen. Die Positionierung von unregelmäßigem PZT unterliegt laut [40] einer Hysterese, einer Drift sowie einem Kriechen. Wird zu langsam verfahren, kann der Objektträger hängen bleiben.

I.6 Messunsicherheitseinflüsse

Einen großen Anteil am Positionierfehler hat die Hysterese des unregulierten PZT. In Abbildung I.6.8 ist eine typische Hysterese abgebildet. Es sind deutliche Nichtlinearitäten zu erkennen. In diesem Beispiel ergibt sich bei Vollaussteuerung abhängig von der Verfahrrichtung ein maximaler Positionierunterschied von beinahe 5 μm .

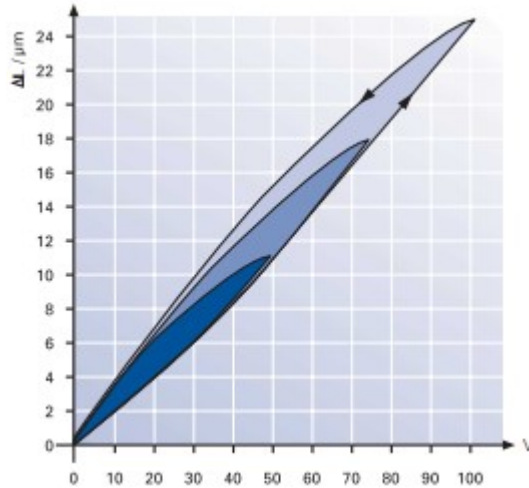


Abbildung I.6.8: Hysterese eines piezokeramischen Aktuators [40]

Zur Korrektur der Hysterese können verschiedene Methoden eingesetzt werden:

- Ein geschlossener Regelkreis kann laut [41] den PZT am besten linearisieren, benötigt aber eine Kontrolleinheit mit einem Sensor und kann instabil werden.
- In [41] wird eine inverse Störgrößenaufschaltung beschrieben, die keinen Sensor benötigt und nicht instabil werden kann.
- Eine weitere sehr einfache Möglichkeit besteht darin, die Hysterese mit zwei Korrekturkurven auszugleichen. Dieser Ansatz ist jedoch nur für eine wiederholte, gleichförmige Ansteuerung zulässig.

Der Einfluss der Hysterese auf die Positionierung kann auch in die Messunsicherheit einbezogen werden.

Viele Signalauswertungsmethoden setzen eine Abtastung des Korrelogramms in festgelegten Abständen voraus. Fehler in der Positionierung des Messobjekts können sich daher stark auf die Messunsicherheit der Topografie auswirken.

I.6.2.4 Kamera

Auch die Linearität und das Rauschen der Kamera nehmen Einfluss auf die Messunsicherheit der Topografie. Da die Auswertemethoden nicht mehrere Bildelemente gemeinsam betrachten, sind Fehler der Kamera aus dem Bereich des „Fixed Pattern Noise“, wie etwa der Dunkelstrom oder Shading nicht von Bedeutung.

Eine Nichtlinearität der Kamera kann einfach kompensiert werden, wenn sie bekannt ist. Das Rauschen der Kamera setzt sich nach [42] und [43] aus Photonen-Schrotrauschen, Dunkel-Schrotrauschen und Ausleserauschen zusammen.

Das Photonen-Schrotrauschen hat seine Ursache in der poissonverteilten Rate des Auftreffens von Photonen auf die Bildelemente des Sensors. Die Anzahl der erzeugten Rausch-Ladungen n_p kann durch

$$n_p = \sqrt{\frac{A \Phi t}{h \nu}} \eta \quad (\text{I.6.3})$$

beschrieben werden, wobei die Detektorfläche mit A , die Lichtstärke mit Φ , das Planksche Wirkungsquantum mit h , die Frequenz des Lichts mit ν , die Zeit mit t und die Quanteneffizienz mit η bezeichnet sind.

Das Dunkel-Schrotrauschen ist die statistische Variation der thermisch generierten Ladungen im Bildelement. Auch dieses Rauschen ist poissonverteilt mit

$$n_d = \sqrt{N_d} \quad (\text{I.6.4})$$

Rausch-Ladungen, wobei N_d die Anzahl der in der Belichtungszeit thermisch generierten Ladungen bezeichnet. Diese Rauschquelle kann durch Kühlung der Kamera stark vermindert werden.

Das Ausleserauschen bezieht jede Rauschquelle vom Auslesen der Ladungen über die Verstärkung bis zur Quantisierung mit ein. Der Verstärker trägt üblicherweise erheblich zum gesamten Ausleserauschen bei. Da dieser auch frequenzabhängiges $1/f$ -Rauschen besitzt, kann sich der Beitrag des Verstärkers mit der Framerate verändern. Das Quantisierungsrauschen hat laut ([18], Kap. 11) keinen signifikanten Einfluss, wenn mit acht oder mehr Bit quantisiert wird.

I.6.3 Messunsicherheit durch äußere Einflüsse

In [44] werden folgende äußere Einflüsse aufgeführt, welche die Messung beeinträchtigen können:

- Vibrationen, die durch Kontakt auf das Messgerät übertragen werden.
- Vibrationen, die durch Schall auf das Messgerät einwirken. Besonders niederfrequente Schallwellen können das Messgerät in Vibration versetzen.
- Luftturbulenzen können die Dichte der Luft im Spalt zwischen Objektiv und Messobjekt kleinräumig ändern und so den Objektstrahl beugen.
- Temperaturänderungen führen zu ungleichmäßiger Ausdehnung von Messgerät und Messobjekt und können die Messunsicherheit erhöhen.
- Staub am Aufstellungsort kann die Optik und das Messobjekt verschmutzen und so die Messung beeinträchtigen.

Hier ist der überwiegende Einflussfaktor die Vibrationen aus Körper- oder Luftschall. Das Messgerät wird dadurch in Schwingung versetzt, die sich direkt auf die Messung des Korrelogramms auswirkt. Diese Einflussfaktoren können durch geeignete Wahl des Aufstellungsortes, durch eine Verkleidung des Messgeräts und einen schwingungsdämpfenden Unterbau weitgehend vermieden werden.

I.6.4 Zusammenfassung

Ein Weißlicht-Interferometer vermisst die Oberfläche eines Messobjekts und berechnet dessen Topografie. Diese Topografie setzt sich aus den Höheninformationen an einem Raster zusammen. Das Raster wird durch den Bildsensor vorgegeben, die Höheninformation wird gemessen.

Die Beschaffenheit, das Material und die Form der Oberfläche haben einen wesentlichen Einfluss auf die Messunsicherheit der Höheninformation:

- Eine raue Oberfläche kann Speckle mit zufälliger Phase verursachen. Die Signalverarbeitung durch Methoden, die auf der Auswertung der Phase basieren, kann somit stark beeinträchtigt werden.
- Das Material des Messobjekts beeinflusst das reflektierte Spektrum und verursacht einen materialspezifischen Phasensprung, wodurch Höhenunterschiede vorgetäuscht werden können. Unterscheidet sich das reflektierte Spektrum stark von dem der Lichtquelle, kann sich eine unerwartete Trägerfrequenz des Korrelogramms ergeben. Dadurch können Methoden beeinträchtigt werden, die bestimmte, auf die Trägerfrequenz abgestimmte Abtastintervalle voraussetzen.
- Die Form des Messobjekts kann die reflektierte Intensität beeinflussen oder durch Mehrfachreflexion eine andere Form vortäuschen. Der Aperturwinkel des Mikroskopobjektivs gibt Auskunft, bis zu welchem Reflexionswinkel Strahlen vom Objektiv erfasst werden können. Besitzt das Messobjekt schiefe Ebenen, deren Neigungen den halben Aperturwinkel überschreiten, können in diesen Bereichen keine Messsignale aufgenommen werden. Durch Mittelung der Phasen und Amplituden der Elementarstreuer in den Auflösungszellen können bei geneigten Flächen nicht existierende Stufen gemessen werden. Besitzt das Messobjekt Stufen, werden Überhöhungen an der Kante der Stufe gemessen. Diese „bat wings“ werden einerseits durch Überlagerung der oberen und unteren Flächen und andererseits durch geringe Beugungseffekte an der Kante hervorgerufen. Bei Stufen, die höher als die Kohärenzlänge sind, wirken sich nur noch durch Beugungseffekte auf die Messung aus. Signalverarbeitungsmethoden, welche die Phase auswerten, werden nur von den Beugungseffekten beeinflusst. Der Messfehler ist daher geringer als bei anderen Methoden.

Die verschiedenen Einflüsse des Messobjekts wirken sich auf die Signalverarbeitungsmethoden unterschiedlich stark aus. Wenn das Messobjekt nicht gänzlich unbekannt ist, kann durch Auswahl einer geeigneten Methode die Messunsicherheit verringert werden. Nachstehende Tabelle kann hier als Auswahlhilfe dienen:

	Oberfläche		Material		Oberflächenform		
	Glatt	Rau	Mehrere	Farbe	Stufen	Kerben	Schräge
geeignete Methoden	alle	Schwerpunkt, Kohärenz	Schwerpunkt, Kohärenz	Schwerpunkt, Kohärenz ⁵	FDA, PSI	keine ⁶	keine

Tabelle I.6.1: Eignung von Signalverarbeitungsmethoden aufgrund von Eigenschaften des Messobjekts

In Kapitel I.5 werden die aufgeführten Methoden beschrieben.

⁵ mit Square-Law- oder Fourier-Transformations-Verfahren

⁶ Keine Methode ist geeignet, da kein Signal zur Verfügung steht, oder dieses eine andere, als die wahre Form wiedergibt.

Auch die Komponenten des Messgeräts beeinflussen die Messunsicherheit der Topografie:

- Durch das Spektrum der Lichtquelle wird die Trägerfrequenz und die Form des Korrelogramms bestimmt. Veränderungen des Spektrums und der Intensität der Lichtquelle durch Strom-, Spannungs- oder Temperaturschwankungen sowie Alterungsprozesse können daher die Signalverarbeitung beeinträchtigen, wenn dadurch Modellannahmen verletzt werden.
- Abbildungsfehler des optischen Systems und die dem Mirau-Objektiv inhärente Dispersion abseits der optischen Achse erhöhen ebenfalls die Messunsicherheit.
- Die Positionierung des Messobjekts ist von besonderer Bedeutung, da einige Signalverarbeitungsmethoden konstante und fixierte Abtastintervalle benötigen. Abweichungen können sich daher stark auf die Messunsicherheit der Topografie auswirken. Die Hysterese des ungerегelten Piezotranslators bewirkt eine ungleichmäßige Abtastung.
- Die verschiedenen Rauschquellen der Kamera haben bei guter Beleuchtung nur geringen Einfluss auf die Messunsicherheit; Dunkel-Schrotrauschen kann durch Kühlung des Sensors einfach vermindert werden.

Auch Umgebungseinflüsse können sich auf die Unsicherheit der gemessenen Höheninformation auswirken. Die überwiegenden Einflüsse sind hier Vibrationen, die über Luft- oder Körperschall auf das Messgerät übertragen werden. Abhilfe kann hier durch ein Gehäuse und eine schwingungsgedämpfte Aufstellung geschaffen werden.

Das nächste Kapitel befasst sich mit der Verbesserung der Messunsicherheit.

I.7 Verringerung der Messunsicherheit

In diesem Kapitel werden Ansätze zur Verringerung der Messunsicherheit der Topografie aufgezeigt. In Kapitel I.6 wurden prinzipielle Quellen der Messunsicherheit ausfindig gemacht. Von diesen sind durch das Messgerät verursachte und äußere Einflussfaktoren beeinflussbar. Darüber hinaus kann die Messunsicherheit durch die Signalverarbeitung und den Messtechniker stark beeinflusst werden.

Unter dem Begriff der Bedienbarkeit werden alle Maßnahmen zusammengefasst, die den Messtechniker bei der Bedienung der Messeinrichtung unterstützen und so zur Verringerung der Messunsicherheit beitragen. In Abbildung I.7.1 sind jene Bereiche Abgebildet, mit denen auf die Messunsicherheit Einfluss genommen werden kann.

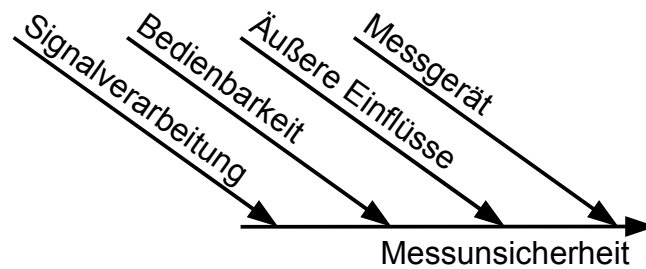


Abbildung I.7.1: Beeinflussbare Bereiche zur Verbesserung der Messunsicherheit

I.7.1 Verbesserungen durch das Messgerät

Im Folgenden werden Verbesserungsmöglichkeiten zu den Messunsicherheits-Quellen aus Kapitel I.6.2 besprochen.

Die Lichtquelle (siehe Kapitel I.6.2.1) bestimmt hauptsächlich die Form des gemessenen Korrelogramms und beeinflusst damit die Auswertemethoden. Diese setzen eine Beleuchtung mit konstantem gaußförmigen Spektrum und konstanter Intensität voraus. Verbesserungen können somit durch eine geeignete Wahl des Leuchtmittels, eine Spannungs- und Temperaturstabilisierung sowie Farbfilter zur Veränderung des Spektrums erreicht werden. Die Breite des Spektrums der Lichtquelle muss so gewählt werden, dass das erzeugte Korrelogramm kleiner als die Tiefenschärfe des Objektivs ist. Durch ihr kontinuierliches, breites Spektrum besonders geeignete Leuchtmittel sind Xenon-Kurzbogenlampen, Wolfram-Halogen-Lampen und phosphor-basierte weiße LEDs (light-emitting diode). Phosphor-basierte weiße LEDs kommen den Auswertemethoden durch die Form ihres Spektrum (siehe Abbildung I.7.2) besonders entgegen.

Das optische System (siehe Kapitel I.6.2.2) besteht im wesentlichen aus dem Objektiv, der Lichteinkoppelung und einer Anpassung an die Kamera. Verbesserungen in diesem Bereich sind durch bessere Korrektur der verschiedenen Abbildungsfehler, insbesondere der chromatischen Aberration, der Linsensysteme möglich. Eine konstruktionsbedingte Dispersion abseits der optischen Achse bei Mirau-Objektiven kann durch dünnere Strahlteiler verringert werden.

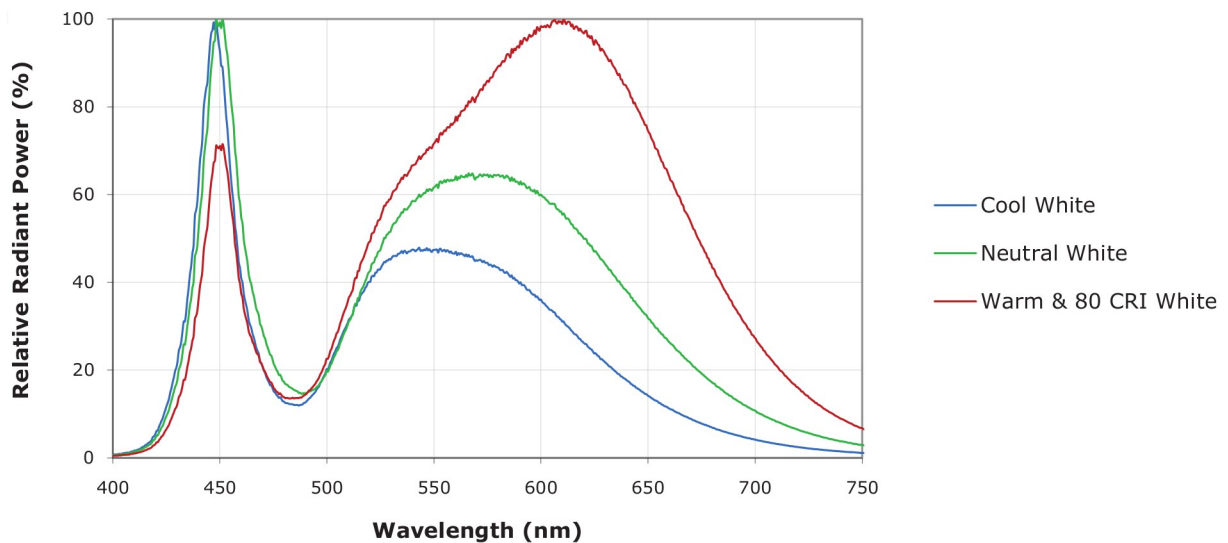


Abbildung I.7.2: Spektren von weißen LEDs der Type „XLamp XP-G“ von Cree [45]

Die Positionierung des Messobjekts (siehe Kapitel I.6.2.3) hat entscheidenden Einfluss auf die Messunsicherheit, da die verschiedenen Auswertemethoden meist eine äquidistante Abtastung des Messvolumens durch die Verschiebung des Messobjekts voraussetzen. Mit dem derzeit eingesetzten unregelmäßigen PZT ist eine solche Abtastung kaum durchführbar, wenn keine Positionsmessung möglich ist. Durch den Einsatz eines geregelten PZT kann die Positionierung erheblich genauer und vorhersehbarer erfolgen. Durch eine Positionsmessung, die nicht zur Regelung dient, in Verbindung mit geeigneten Auswertemethoden kann ebenfalls die Messunsicherheit verbessert werden.

Die Kamera (siehe Kapitel I.6.2.4) detektiert in jedem Bildelement die Intensität des Messsignals und erzeugt einen korrespondierenden Zahlenwert. Verbesserungen können im Bereich des Ausleserausens und des Dunkel-Schrotrauschens erreicht werden. Das Ausleserausens besteht im wesentlichen aus Verstärker- und Quantisierungsrauschen. Das Verstärkerausens kann durch Abschalten der Verstärkung entfernt werden. Das Dunkel-Schrotrauschen kann durch Kühlung stark vermindert werden. Die Linearität und das Quantisierungsrauschen können nur durch ein Austauschen der Kamera beeinflusst werden.

I.7.2 Verbesserung durch Verringerung äußerer Einflüsse

Äußere Einflüsse (siehe Kapitel I.6.3) können weitgehend durch konstruktive Maßnahmen verringert werden. Bedeutend sind hier vor allem Maßnahmen zur Schwingungsdämpfung, da sich Schwingungen direkt auf den Abstand zwischen Messobjekt und Objektiv auswirken. Einfache Maßnahmen sind ein schwingungsdämpfender Unterbau gegen Körperschall und eine Verkleidung des Messgeräts gegen Luftschall.

I.7.3 Verbesserungen durch die Bedienbarkeit

Die Bedienbarkeit umfasst sowohl die Bedienung des Messgeräts als auch die des Programms.

Die Bedienung des Messgeräts beschränkt sich auf die Einstellung des Messobjekts, sodass dessen interessierender Bereich im Fokus und normal zur optischen Achse ausgerichtet ist und sich im Zentrum des Messfeldes befindet. Um das Messobjekt in den Fokus zu bringen, muss der Objektträger entlang der optischen Achse verschoben werden. Zur Ausrichtung des Messobjekts normal zur optischen Achse muss der Objektträger verkippert werden. Der interessierende Bereich kann durch horizontales Verschieben des Messobjekts in das Zentrum des Messfeldes bewegt werden.

Der bestehende Aufbau des Objektträgers (Abbildung I.7.3) kann auf zwei Arten verbessert werden. Zum einen ist das Auffinden des interessierenden Bereichs durch Verschieben des Messobjekts mit freier Hand schwierig. Zum Anderen wird dieser Bereich aus dem Fokus und aus dem Messfeld bewegt, wenn der Objektträger verkippert wird.

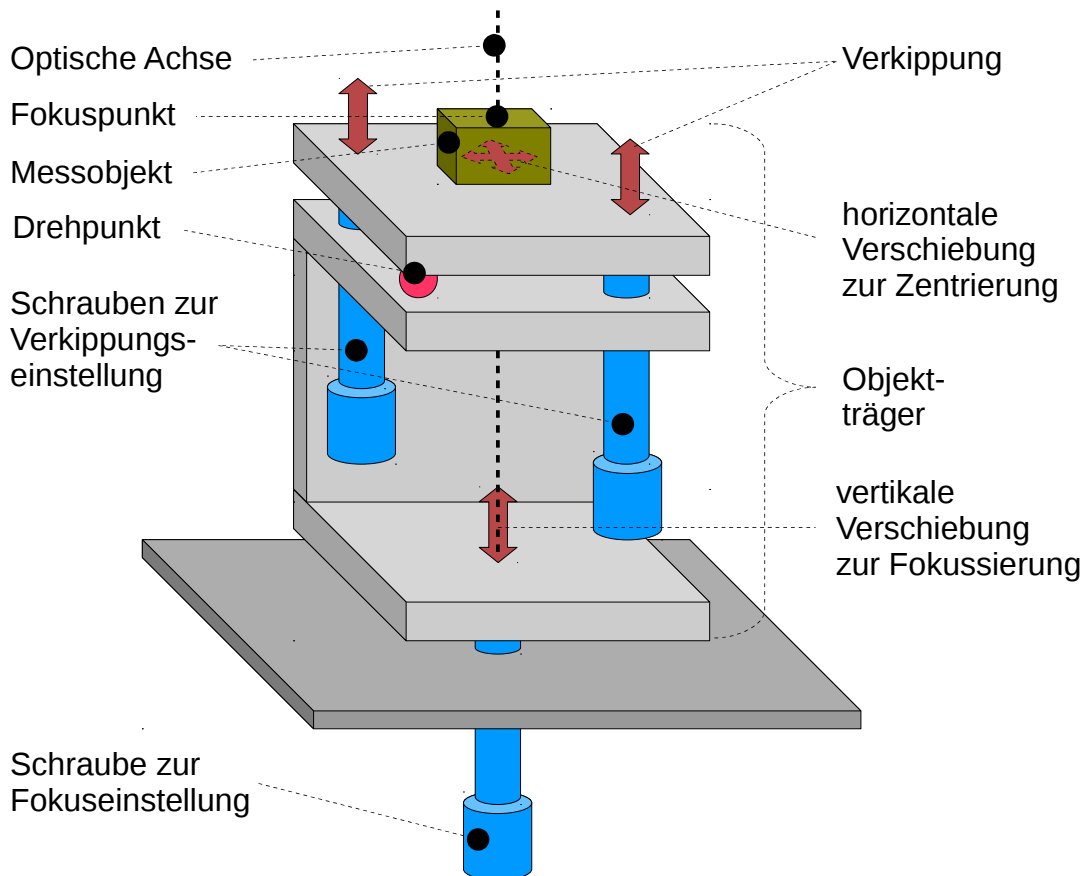


Abbildung I.7.3: Schematischer Aufbau des Objektträgers; Zur Veränderung der Verkippung wird durch Drehung an den Stellschrauben die oberste Platte um den Drehpunkt gekippt, wodurch sich das Messobjekt sowohl horizontal als auch vertikal aus dem Fokuspunkt bewegt.

Das horizontale Verschieben des Messobjekts kann vereinfacht werden, wenn zu diesem Zweck beispielsweise Mikrometer-Schrauben vorgesehen werden. Die Einstellung der Verkippung lässt sich vereinfachen, wenn das Messobjekt oder das Objektiv um den Fokuspunkt rotiert wird. Der interessante Bereich des Messobjekts bleibt dann im Fokus und im Messfeld.

Die Positionierung des Messobjekts kann durch das Programm unterstützt werden, indem ein Live-Bild angezeigt wird. Durch Testmessungen können die Signalqualität und die Verkippung bestimmt werden und die Ergebnisse im Live-Bild durch Färbung für die Signalqualität und Pfeile für die Verkippung angezeigt werden. Die Signalqualität kann hauptsächlich durch Änderung der Belichtungszeit oder der Beleuchtungsintensität beeinflusst werden. Mit Hilfe der Testmessungen kann der Messtechniker eventuell auf geeignete Auswertemethoden hingewiesen werden. Vor Vibrationen des Messgeräts kann gewarnt werden, wenn im Live-Bild ein Zittern der Interferenzstreifen detektiert wird.

I.7.4 Verbesserung durch die Signalverarbeitung

In Weißlicht-Interferometern werden üblicherweise verschiedene Methoden der Signalverarbeitung zur Auswahl angeboten. Jede Methode hat bestimmte Vor- und Nachteile, die in Kapitel I.5 besprochen werden. Die Möglichkeit, eine für das Messobjekt geeignete Signalverarbeitung zu wählen, bedingt die Implementierung einer Auswahl von Auswertemethoden. Zusätzlich zu den in Kapitel I.5 beschriebenen Auswertemethoden ist eine Auswertung von ungleichmäßig abgetasteten Korrelogrammen unter Zuhilfenahme einer Positionsmessung denkbar. Eine Auswertung im Frequenzbereich ist bei ungleichmäßiger Abtastung analog zur Frequency-Domain-Analysis-Methode in Kapitel I.5.4 etwa durch den in ([46], Kap. 13.8) beschriebenen Algorithmus möglich. Auch die Schwerpunktmethode liefert in diesem Fall gute Ergebnisse.

II Praktischer Teil

In diesem zweiten Teil der vorliegenden Arbeit wird die Entwicklung eines Programms beschrieben, das dem Messtechniker die Einstellung des Messobjekts erleichtert sowie Messdaten speichert und Auswertet. Der Entwurfsprozess umfasst die Modellierung und Analyse von Anwendungsfällen, die Entwicklung eines Entwurfskonzepts sowie den eigentlichen Entwurf und dessen Implementierung.

II.1 Einleitung

Nach ([47], Kapitel 1) enthält jeder Softwareentwicklungsprozess unabhängig von der verwendeten Vorgehensweise die folgenden Teilbereiche:

- Am Beginn jeder Softwareentwicklung werden Anforderungen erhoben. Dies geschieht durch die Modellierung von Anwendungsfällen. In Anwendungsfällen werden Akteure identifiziert, und die geforderten Funktionalitäten im Problembereich, also ohne die Implementierung zu berücksichtigen, beschrieben. Die Beschreibung erfolgt textuell und kann durch Verhaltens- oder Strukturdiagramme veranschaulicht werden.
- In einem zweiten Schritt werden gefundene Anwendungsfälle analysiert. Dabei werden Klassen und deren Interaktionen im Problembereich gebildet, mit denen die Anwendungsfälle durchgeführt werden können. Diese Klassen bilden keinen Entwurf zur Implementierung, sondern unterteilen Anwendungsfälle in Problembereiche.
- Beim Entwurf wird das Ergebnis der Analyse in einem Programm-Entwurf umgesetzt. Klassen und Interaktionen aus dem Problembereich werden jeweils ein oder mehrere Klassen des technischen Bereichs abgebildet.
- Es folgt die Phase der Implementierung, in welcher der Entwurf mit Hilfe einer objektorientierten Programmiersprache kodiert wird.
- Die Implementierung wird abschließend getestet. Verschiedene Testmethoden können hier zur Anwendung kommen:
 - Unit-Tests werden vom Programmierer idealerweise vor der Implementierung der zu testenden Klasse oder Gruppe von Klassen erstellt.
 - Integrationstests prüfen die beabsichtigte Zusammenarbeit der Komponenten.
 - Systemtests stellen sicher, dass die umzusetzende Funktionalität wie beabsichtigt funktioniert.
 - Durch Akzeptanztests kann sichergestellt werden, dass die Software die Anforderungen des Kunden erfüllt.

In den nachfolgenden Kapiteln wird nach diesem Schema vorgegangen. Zunächst werden Anwendungsfälle modelliert und analysiert. Aus der Analyse ergeben sich grundlegende Anforderungen und Klassen im Problembereich. Mit diesen Erkenntnissen und Anforderungen aus anderen Quellen wird ein Entwurfskonzept erstellt. Im abschließenden Kapitel werden die Klassen des Problembereichs in objektorientierte Entwürfe abgebildet und deren Implementierungen beschrieben.

II.2 Anwendungsfälle

In ([47], Kap. 3) wird ein Anwendungsfall definiert, als eine Anzahl von Aktionen, die ein System ausführt und die von einem Anwender initiiert werden, um ein gewünschtes Ergebnis zu erhalten.

Um die Modellierung der Anwendungsfälle zu erleichtern, wird hier zunächst eine Abfolge von Tätigkeiten beschrieben, die zu einer gemessenen Topografie führt:

- Eine neue Messeinrichtung muss üblicherweise zunächst in Betrieb genommen werden. Dabei kalibriert ein Wartungstechniker alle Komponenten, hinterlegt Korrekturen und legt diverse Einstellungen fest.
- Ein Messtechniker kann nun eine Messung vornehmen. Dazu muss das Messobjekt so eingerichtet werden, dass der interessierende Bereich zentriert und normal zur optischen Achse ausgerichtet ist. Des Weiteren muss dieser Bereich im Fokus liegen und die Signalqualität eine Auswertung erlauben.
- Ist das Messobjekt eingerichtet, kann eine Messung vorgenommen werden. Der Messtechniker kann eine Auswertemethode auswählen und Einstellungen vornehmen. Alle Einstellungen und Daten werden gespeichert, sodass die Messung vollständig dokumentiert ist.
- Eine bereits durchgeführte Messung kann wiederholt werden, indem die Daten neu ausgewertet werden.

Zunächst werden die Akteure identifiziert. Akteure sind in ([47], Kap. 3) als Entitäten die an der Verwendung des Systems interessiert oder an dieses angebunden sind, definiert. Das Programm interagiert einerseits mit dem Messgerät und andererseits mit einem Benutzer. Der Benutzer kann in zwei Rollen auftreten: als Messtechniker und als Wartungstechniker. Damit lassen sich als Akteure der Messtechniker, der Wartungstechniker und das Messgerät identifizieren.

Die obige Abfolge von Tätigkeiten kann durch die folgenden Anwendungsfälle beschrieben werden:

- *Konfiguration*: Der Wartungstechniker nimmt die Messeinrichtung in Betrieb.
- *Vorbereitung*: Der Messtechniker bereitet eine Messung vor.
- *Messung*: Der Messtechniker nimmt eine Messung vor.
- *Wiederholung*: Der Messtechniker wertet vorhandene Daten erneut aus.

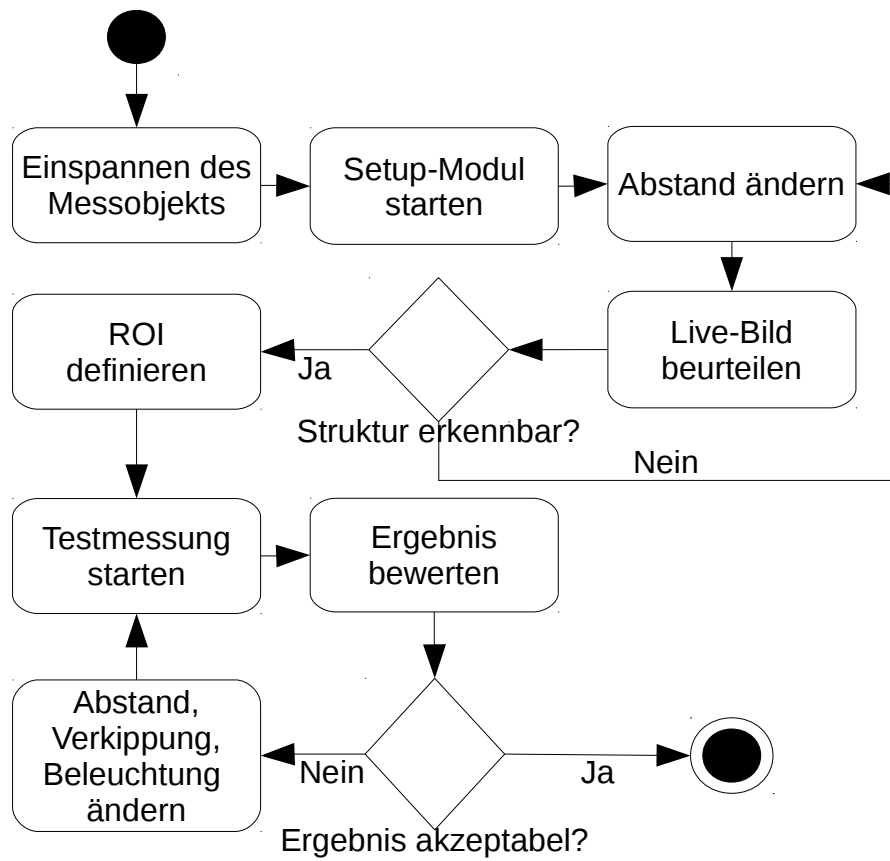
In Anlehnung an die tabellarische Beschreibung von Anwendungsfällen in ([48], Kap. 3) werden in den folgenden Unterkapiteln Tabellen verwendet, um die Anwendungsfälle zu beschreiben. Jeder Anwendungsfall wird mit einem Titel, ein oder mehreren Akteuren, einer Priorität, einer Kurzbeschreibung, einer Vor- und Nachbedingung, einer Ablaufbeschreibung und einem Ablaufdiagramm beschrieben.

II.2.1 Anwendungsfall 1: Vorbereitung

Titel:	Anwendungsfall 1: Vorbereitung
Akteure:	Messtechniker, Messgerät
Priorität:	Hoch
Kurzbeschreibung:	Der Akteur positioniert das Messobjekt, sodass eine Messung mit möglichst geringer Messunsicherheit vorgenommen werden kann.
Vorbedingung:	Die Messeinrichtung ist in Betrieb.
Nachbedingung:	Das Messobjekt ist justiert, sodass eine Messung erfolgen kann.
Ablauf:	<p>Das Messobjekt wird auf den Objektträger gelegt.</p> <p>Das Vorbereitungs-Modul der Applikation wird gestartet.</p> <p>Der Messtechniker verändert den Abstand zum Objektiv mit Hilfe einer Einstellschraube am Objektträger, sodass Strukturen im Live-Bild des Setup-Moduls erkennbar sind. Das Messobjekt ist nun im Fokus.</p> <p>Der Messtechniker verschiebt das Messobjekt auf dem Objektträger, sodass der interessierende Bereich des Messobjekts im Live-Bild erkennbar ist. Das Messobjekt ist nun zentriert.</p> <p>Einen oder mehrere ROIs werden definiert.</p> <p>Der Messtechniker startet eine Testmessung.</p> <p>Nach der Testmessung werden Bereiche mit unzureichender Signalqualität im Live-Bild angezeigt. Die zuvor definierten ROIs sollen in einer Ebene liegen und normal auf die optische Achse des Objektivs ausgerichtet sein. Eine Verkippung wird angezeigt.</p> <p>Der Messtechniker entscheidet, ob Signalqualität und Verkippung akzeptabel sind.</p> <p>Sind die Ergebnisse nicht zufriedenstellend, ändert der Messtechniker die Beleuchtungsintensität und/oder die Position des Messobjekts und führt eine weitere Testmessung durch.</p> <p>Sind die Ergebnisse zufriedenstellend, ist die Messung vorbereitet und das Setup-Modul kann geschlossen werden.</p>

II.2 Anwendungsfälle

Ablauf-
diagramm:

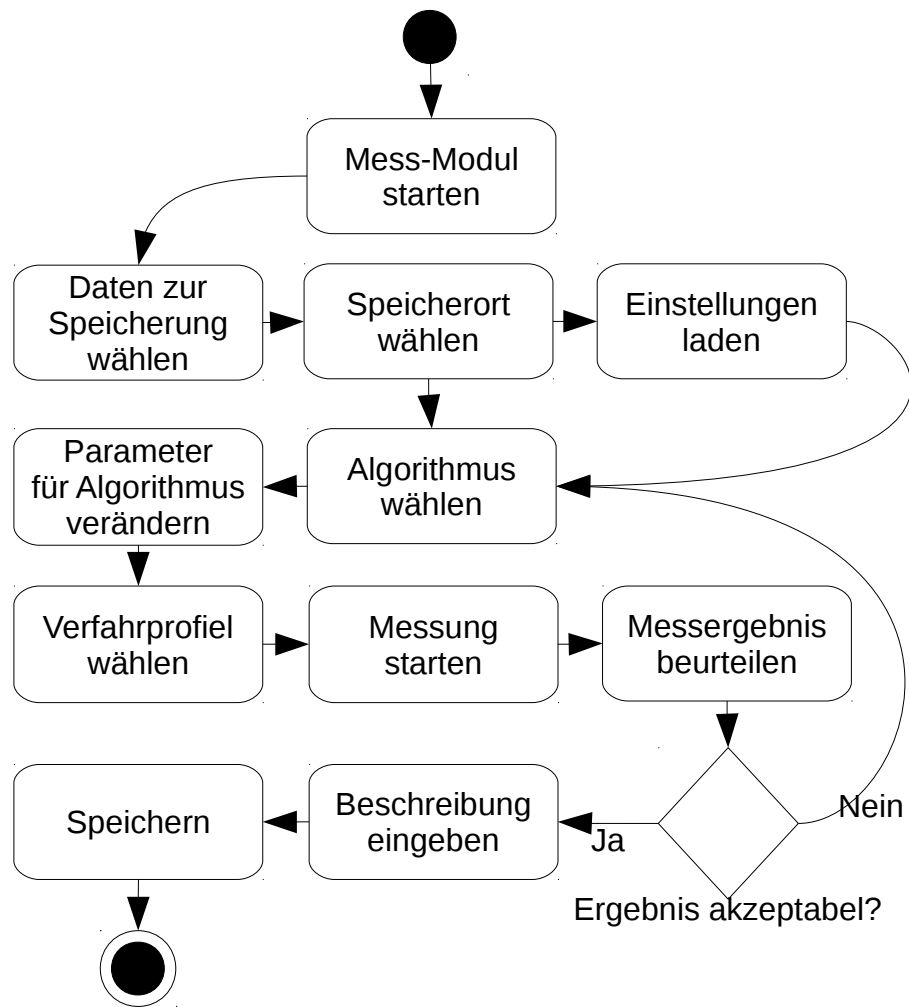


II.2.2 Anwendungsfall 2: Messung

Titel:	Anwendungsfall 2: Messung
Akteure:	Messtechniker, Messgerät
Priorität:	Hoch
Kurzbeschreibung:	Der Akteur führt eine Messung durch.
Vorbedingung:	Die Messeinrichtung ist in Betrieb und Anwendungsfall 1 oder 3 wurde ausgeführt.
Nachbedingung:	Messwerte wurden aufgenommen und eine Topografie wurde berechnet.
Ablauf:	<p>Der Messtechniker startet das Mess-Modul, das ein Live-Bild anzeigt.</p> <p>Der Messtechniker wählt die zu speichernden Daten (Messdaten, Einstellungen, Topografie).</p> <p>Ein Speicherort wird zu diesem Zeitpunkt gewählt, um nachfolgendes Kopieren großer Datenmengen zu vermeiden.</p> <p>Es können Einstellungen einer bereits durchgeführten Messung geladen werden.</p> <p>Ein Topografieberechnungs-Algorithmus wird gewählt.</p> <p>Der Messtechniker ändert Parameter des gewählten Topografieberechnungs-Algorithmus.</p> <p>Ein Verfahrenprofil wird ausgewählt.</p> <p>Der Messtechniker startet die Messung.</p> <p>Das Messergebnis (die gefundene Topografie) wird angezeigt.</p> <p>Der Messtechniker beurteilt das Ergebnis.</p> <p>Ist das Ergebnis nicht zufriedenstellend, kann der Messtechniker die oben angeführten Eingaben ändern und eine neue Messung starten.</p> <p>Ist das Ergebnis zufriedenstellend, kann der Messtechniker die Messung speichern.</p> <p>Der Messtechniker gibt eine Beschreibung der Messung ein.</p> <p>Der Messtechniker aktiviert die Speicherung.</p> <p>Die Messung ist abgeschlossen.</p>

II.2 Anwendungsfälle

Ablauf-
diagramm:



II.2.3 Anwendungsfall 3: Wiederholung

Titel:	Anwendungsfall 3: Wiederholung
Akteure:	Messtechniker
Priorität:	Mittel
Kurzbeschreibung:	Der Akteur wertet gespeicherte Messdaten erneut aus.
Vorbedingung:	Die Messeinrichtung ist in Betrieb. Anwendungsfall 2 wurde ausgeführt und Messdaten wurden gespeichert.
Nachbedingung:	Eine Topografie wurde berechnet.
Ablauf:	<p>Das Wiedergabe-Modul wird geöffnet. Eine gespeicherte Messung wird ausgewählt. Die Wiedergabe wird gestartet. Das Wiedergabe-Modul wird geschlossen. Der Messtechniker führt Anwendungsfall 2 aus. Das Wiedergabe-Modul wird geöffnet. Der Messtechniker stoppt das Wiedergabe-Modul. Damit ist die Wiedergabe beendet.</p>
Ablaufdiagramm:	<pre> graph TD Start(()) --> A[Wiedergabe-Modul öffnen] A --> B[Messung auswählen] B --> C[Wiedergabe starten] C --> D[AF 2 ausführen] D --> E[Wiedergabe-Modul öffnen] E --> F[Wiedergabe stoppen] F --> End((())) </pre>

II.2.4 Anwendungsfall 4: Wartung

Titel:	Anwendungsfall 4: Wartung
Akteure:	Wartungstechniker
Priorität:	Mittel
Kurzbeschreibung:	Der Akteur verändert Standardeinstellungen, gibt Korrekturen ein und verändert das Verschiebeprofil.
Vorbedingung:	Die Messeinrichtung ist in Betrieb.
Nachbedingung:	Die Messeinrichtung ist zur Messung bereit.
Ablauf:	Das Verwaltungsmodul wird geöffnet. Alle Parameter aller Programmteile können verändert werden. Dazu zählen unter anderem Standard-Korrekturkurve, Standard-Verschiebeprofil, Standardverzeichnisse, LogLevel, Wertebereiche, Belichtungszeit, Triggerpolarität und Umrechnungsfaktoren.

II.3 Analyse der Anwendungsfälle

In diesem Unterkapitel werden mit Hilfe der Anwendungsfälle Klassen im Problembereich gebildet. Diese Klassen können externe Systeme, Geräte oder organisatorische Einheiten modellieren. Die folgenden Klassen können durch Analyse der Anwendungsfälle definiert werden:

- **Vorbereitung:**
Diese Klasse unterstützt den Messtechniker bei der Einrichtung des Messobjekts. Sie kann eine Verschiebung des Messobjekts auslösen und Daten von einer Kamera empfangen und auswerten. Die Auswertung umfasst die Signalqualität und die Verkippung. Zur Interaktion mit dem Benutzer wird ein Live-Bild des Messobjekts angezeigt. Der Benutzer kann Bereiche definieren, die für eine Verkippungsbrechung verwendet werden.
- **Messung:**
Diese Klasse erlaubt die Ausführung einer Messung. Das Modul kann eine Verschiebung des Messobjekts veranlassen und den Weg der Verschiebung bestimmen. Es können verschiedene Signalverarbeitungs-Algorithmen ausgewählt werden. Aufgrund von Einschränkungen von Signalverarbeitungs-Algorithmen kann auf das Abtastintervall Einfluss genommen werden. Das Modul kann eine Topografie und ein Live-Bild anzeigen. Es kann die Speicherung aller Einstellungen und Daten an einem bestimmten Ort veranlassen.
- **Wiedergabe:**
Diese Klasse gibt gespeicherte Messdaten wieder, sodass diese mit der Klasse Messung erneut ausgewertet werden können.
- **Verwaltung:**
Diese Klasse entspricht der zentralen Verwaltung. LogLevel, Standardverzeichnisse und andere Parameter können hier verändert werden. Hier können Dienste angeboten werden, die von anderen Klassen benötigt werden, etwa das Lesen und Schreiben von Einstellungen.
- **Messdatenspeicherung:**
Diese Klasse speichert Messdaten.
- **Topografiespeicherung**
Diese Klasse speichert die berechnete Topografie.

- **Anzeige**
Diese Klasse visualisiert die Messdaten. Sowohl die Klasse Messung als auch die Klasse Vorbereitung benötigen eine Visualisierung der Messdaten.
- **Topografieberechnung**
Diese Klasse berechnet eine Topografie. Diese Klasse wird von den Klassen Vorbereitung und Messung benötigt.
- **Signalbeurteilung**
Diese Klasse beurteilt die Qualität des Messsignals und wird von der Vorbereitungs-Klasse benötigt.
- **Verkippungsberechnung**
Diese Klasse berechnet die Verkippung. Die Vorbereitungs-Klasse benötigt eine Berechnung der Verkippung.
- **Datenerfassung:**
Diese Klasse dient als Messdatenquelle für die Klassen Vorbereitung und Messung. Die Messdaten setzen sich aus den aufgenommenen Bildern und den entsprechenden Positionen zusammen.
- **Bewegung:**
Diese Klasse übersetzt das Fahrprofil in ein Spannungsprofil, mit dem der Piezotranslator angesteuert wird, und meldet die aktuelle Position an die Klasse Datenerfassung.
- **Kamera:**
Diese Klasse steuert die Kamera. Aufgenommene Bilder werden an die Datenerfassungs-Klasse gemeldet.

Anwendungsfall 1 kann mit den gebildeten Klassen wie in Abbildung II.3.1 als Strukturdiagramm dargestellt werden. Die Klasse Vorbereitung kann Live-Bilder von der Klasse Kamera beziehen und in der Klasse Anzeige darstellen lassen. Die Vorbereitungs-Klasse kann eine Testmessung anstoßen und die erhaltenen Daten an die Klassen Signalbeurteilung und Topografieberechnung weiterleiten. In der Verkippungsberechnung werden die Ergebnisse der Topografieberechnung verwendet. Die Berechnungsergebnisse können unter Vermittlung der Vorbereitungs-Klasse in der Anzeige visualisiert werden.

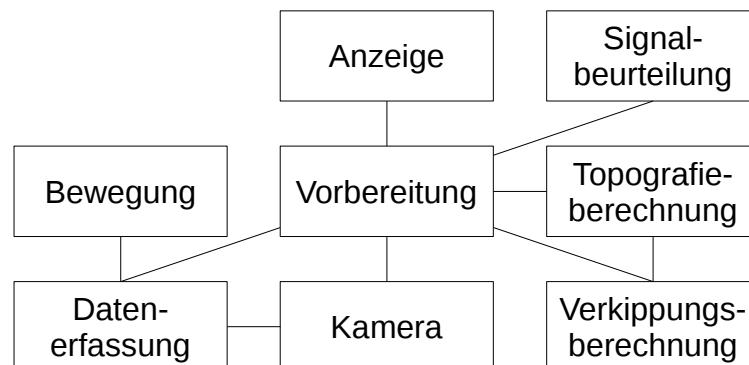


Abbildung II.3.1: Strukturdiagramm zu Anwendungsfall 1: Vorbereitung

Das Strukturdiagramm zu Anwendungsfall 2 ist in Abbildung II.3.2 zu sehen. Die Klasse Messung kann Messdaten durch die Klasse Datenerfassung beziehen und diese der Klasse Datenspeicherung und Topografieberechnung weiterleiten. Die berechnete Topografie kann in der Anzeige-Klasse visualisiert werden. Eine berechnete Topografie kann mit der Klasse Topografiespeicherung gespeichert werden.

II.3 Analyse der Anwendungsfälle

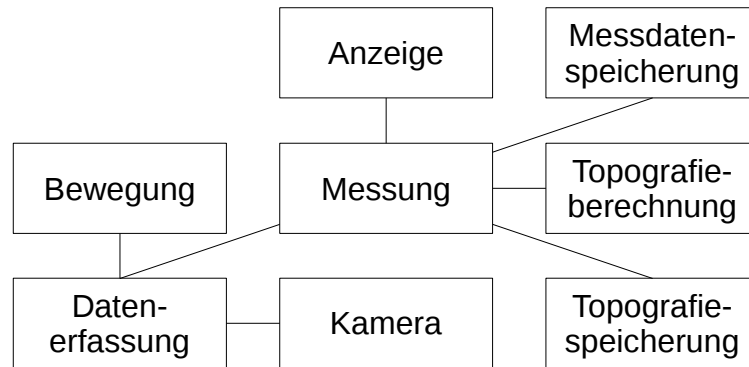


Abbildung II.3.2: Strukturdiagramm zu Anwendungsfall 2: Messung

Bei der Wiederholung einer Messung (Anwendungsfall 3) wird gegenüber Anwendungsfall 2 nur die Datenerfassungs-Klasse gegen die Wiedergabe-Klasse ausgetauscht. Das entsprechende Strukturdiagramm ist in Abbildung II.3.3 zu sehen.

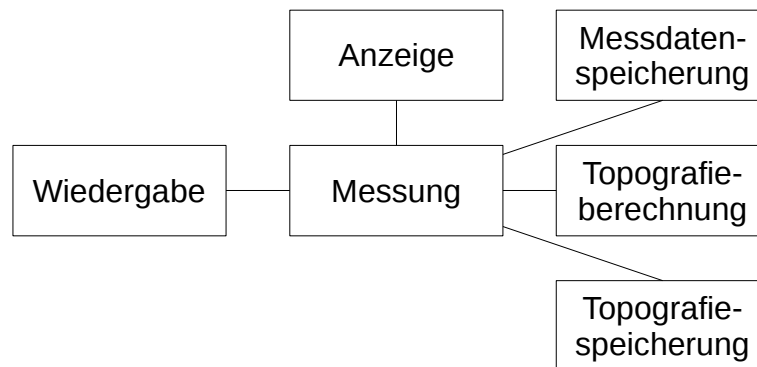


Abbildung II.3.3: Strukturdiagramm zu Anwendungsfall 3: Wiederholung

II.4 Entwurfskonzept

In diesem Kapitel werden die grundlegenden Entwurfsentscheidungen besprochen und begründet. Dieses Konzept bildet die Grundlage und Leitlinie des weiteren Entwicklungsprozesses.

In ([49], Kap. 23.4.2) wird als Ziel der Architektur die Erhaltung der Einfachheit und Verständlichkeit bei angemessener

- Flexibilität,
- Portabilität und
- Erweiterbarkeit

definiert. Diese Ziele können erreicht werden, indem Bereiche, in denen Änderungen zu erwarten sind, gekapselt werden. Durch Kapselung wird eine Lokalität von Änderungsauswirkungen erzeugt. Somit wird dem Entwickler ein Weg geöffnet, Funktionalitäten zu verändern, ohne andere Teile des Programms zu beeinflussen. Die Granularität der Kapselung kann von Funktionen über Klassen und Module bis hin zu verschiedenen Programmen reichen.

II.4.1 Anforderungen

Die hier angegebenen Anforderungen ergeben sich aus den Anwendungsfällen und aus anderen Quellen:

- In Kapitel I.7.3 wurde als Aufgabe des Programms die Unterstützung des Messtechnikers bei der Einrichtung des Messobjekts und bei der Messung gefordert, sodass die Messunsicherheit der Topografie möglichst gering ist. Zu diesem Zweck soll das Programm
 - Rückmeldungen über Signalqualität, Verkipfung und Fokus liefern,
 - die Einstellung von Parametern unmissverständlich und überprüfbar gestalten,
 - Meldungen über den Ablauf des Programms aufzeichnen, sodass unerwartetes Verhalten erkannt werden kann und
 - Hinweise auf Fehlerquellen geben, welche die Messunsicherheit erhöhen können.
- Die Messung muss vollständig gespeichert werden können. Zu diesem Zweck müssen die Messdaten, alle Einstellungen der beteiligten Komponenten und die Auswertung gespeichert werden können.
- Messungen sollen mit bestehenden Messdaten wiederholt werden können.
- Einzelne Funktionalitäten des Programms sollen einfach ausgetauscht werden können.

Die Messung erfolgt durch Aufnahme von Bildern. Das Messgerät (siehe Anhang A) ermöglicht die Aufnahme von bis zu 15 Bildern pro Sekunde mit jeweils 1,17 Megabyte (MB). Es ist daher mit großen Datenmengen zu rechnen. Weiters muss das Programm sehr unterschiedliche Hardware mit sehr unterschiedlichen Eigenschaften integrieren. Aufgrund des universitären Umfeldes sind starke Veränderungen des Programms und der Hardware zu erwarten. Daher muss bei Entwurf und Implementierung auf folgende Punkte besonders Wert gelegt werden:

- Aufgrund der zu erwartenden Änderungen in Hardware und Messdatenverarbeitung ist ein sehr hohes Maß an Flexibilität notwendig.
- Die Verarbeitung der Messdaten soll parallel und asynchron erfolgen.
- Messdaten sollen möglichst nicht kopiert werden.

II.4.2 Konsequenzen

Die Anforderung der hohen Flexibilität kann durch weitgehende Kapselung der verschiedenen funktionalen Einheiten erfüllt werden.

Dies kann auf oberster Ebene durch die Definition von Modulen mit möglichst kleinen Schnittstellen innerhalb eines monolithischen Programms erfolgen. Der Entwickler muss in diesem Fall darauf achten, bei Änderungen die Modulgrenzen einzuhalten. Der kurzfristig einfachere Weg für Entwickler ist zumeist, diese Modulgrenzen zu umgehen. Dadurch werden Abhängigkeiten geschaffen, die im Zuge von Veränderungen in kurzer Zeit zu hoher Komplexität führen können.

Etwas schwieriger ist die Umgehung von Modulgrenzen, wenn die Module als dynamisch ladbare Bibliotheken angelegt werden. Jedes Modul kann als abgeschlossene Einheit betrachtet werden, die ebenfalls nach obigen Anforderungen entworfen wird.

Eine parallele Verarbeitung kann durch Verwendung paralleler Ausführungspfade (Threads) erfolgen. Die Anwendung von Threads wird durch das Betriebssystem und durch diverse Bibliotheken unterstützt und gestaltet sich relativ einfach. Threads können durch das Betriebssystem auf mehrere Rechenkerne moderner Computer verteilt werden, wodurch rechenintensive Verarbeitungsschritte üblicherweise deutlich beschleunigt werden können. Bei asynchroner Verarbeitung werden Threads verwendet, um parallel auf Ereignisse zu warten. So kann das Programm schneller reagieren, wenn zum Beispiel ein neues Bild eintrifft. Des Weiteren kann das Betriebssystem einen wartenden Thread auslagern und somit Ressourcen schonen.

Aufgrund dieser Überlegungen werden folgende grundlegende Entwurfsentscheidungen getroffen:

- Module werden als ladbare Bibliotheken ausgeführt.
- Daten werden in Threads verarbeitet.
- Auf Ereignisse wird in Threads gewartet.

II.4.3 Modulkonzept

In diesem Kapitel wird die Entwurfsentscheidung ladbare Module einzusetzen, konkretisiert. Module sollen die Flexibilität der Implementierung erhöhen, sodass sich Änderungen in einem Bereich nicht auf andere Bereiche auswirken. Daher werden in Modulen Teile gekapselt, die sich voraussichtlich ändern werden. Die Klassen des Problembereichs aus Kapitel II.3 dienen dabei als Basis zur Definition der Module. Mit den gefundenen Modulen werden Anwendungsfälle gebildet, um mit deren Hilfe Anforderungen an das Modulkonzept zu formulieren.

II.4.3.1 Moduldefinitionen

In diesem Kapitel werden Module definiert. Die Anwendungsfälle aus Kapitel II.2 bilden bereits Basismodule. Einige Problembereichs-Klassen aus den Anwendungsfällen werden als Kandidaten für Veränderungen eingeschätzt. Zu diesen zählen Klassen, die Geräte ansteuern, Berechnungen durchführen oder Daten verändern, speichern oder laden. Diese Klassen werden in neue Module des Problembereichs ausgegliedert. Es ergeben sich folgende Basis-Module, die jeweils einen Anwendungsfall abdecken:

- **Vorbereitungs-Modul**
Dieses Modul unterstützt den Messtechniker bei der Einrichtung der Probe und setzt Anwendungsfall 1 (Kapitel II.2.1) um. Das überarbeitete Strukturdiagramm des Anwendungsfalls aus Kapitel II.3 ist in Abbildung II.4.1 zu sehen. Die Klassen Topografieberechnung, Datenerfassung, Kamera und Bewegung werden in eigene Module ausgegliedert.

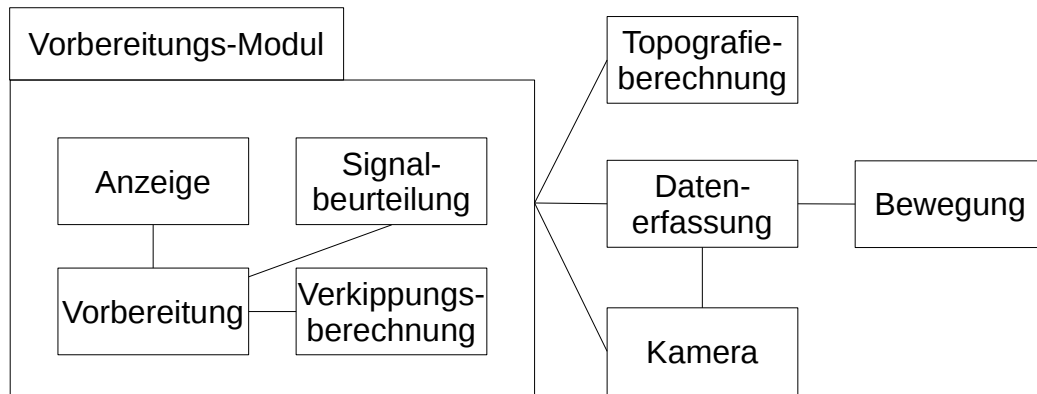


Abbildung II.4.1: Strukturdiagramm des Vorbereitungs-Moduls

- **Mess-Modul**

Dieses Modul erlaubt die Ausführung und die Speicherung einer Messung und setzt Anwendungsfall 2 (Kapitel II.2.2) um. In Abbildung II.4.2 ist das entsprechend überarbeitete Strukturdiagramm aus Kapitel II.3 dargestellt. Hier wurden zusätzlich zu den im Vorbereitungs-Modul ausgegliederten Klassen noch Messdatenspeicherung und Topografiespeicherung ausgegliedert.

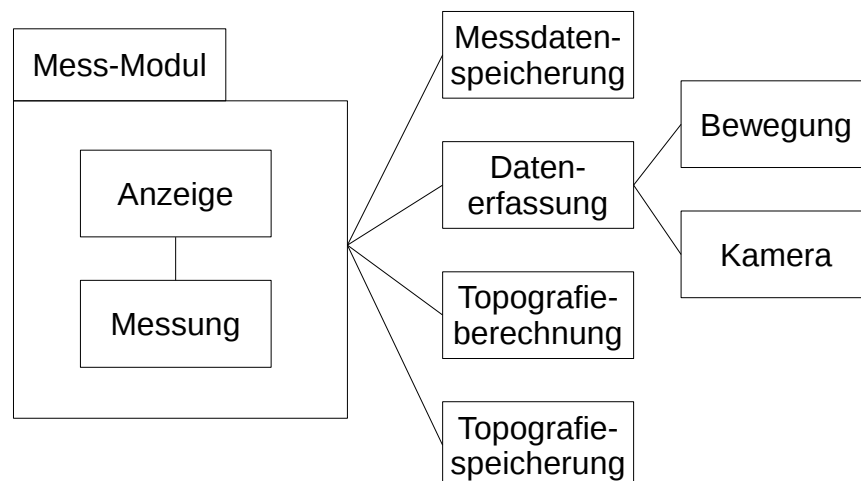


Abbildung II.4.2: Strukturdiagramm des Mess-Moduls

- **Wiedergabe-Modul**

Dieses Modul ermöglicht die erneute Ausführung einer Messung und setzt Anwendungsfall 3 (Kapitel II.2.3) um. Es entspricht der Klasse Wiedergabe aus Kapitel II.3. Es sind hier keine zusätzliche Ausgliederungen notwendig.

- **Verwaltungsmodul**

Dieses Modul ist für Verwaltung und Parametrierung zuständig und setzt Anwendungsfall 4 (Kapitel II.2.4) um. Alle definierten Module werden verwendet um deren Parameter zu verändern, zu laden und zu speichern. Es werden Dienste für andere Module zur Verfügung gestellt und die Module und deren Verbindungen verwaltet.

Zusätzlich zu den Basis-Modulen werden die ausgegliederten Module folgendermaßen definiert:

- **Messdatenspeicherungs-Modul**

Dieses Modul speichert Messdaten, die aus Bildern und den zugehörigen Positionen bestehen. Zur Speicherung dieser Daten sind verschiedene Speicherformate denkbar.

II.4 Entwurfskonzept

- **Datenerfassungs-Modul**
Dieses Modul muss Bilder der Kamera mit den entsprechenden Positionen bei der Aufnahme verknüpfen. Da die Steuerung des Piezotranslators nur asynchron erfolgen kann, müssen die Ausgaben zu Bildauslösung und Positionierung synchronisiert werden. Nur so können aufgenommene Bilder den jeweiligen Positionen zugeordnet werden.
- **Bewegungs-Modul**
Dieses Modul steuert den Piezotranslator aufgrund von Positionsdaten eines anderen Moduls.
- **Kamera-Modul**
Dieses Modul parametrisiert die Kamera, empfängt Bilder von der Kamera und gibt diese an andere Module weiter.
- **Topografieberechnungs-Modul**
Dieses Modul berechnet aus den Messdaten (Bilder mit Positionen) die Topografie. Jede Auswertemethode wird in ein Modul dieses Typs gekapselt.
- **Topografiespeicherungs-Modul**
Dieses Modul speichert topografische Daten. Es gibt verschiedene Möglichkeiten, die Topografie als dreidimensionales Objekt zu speichern. Jedes implementierte Format wird in ein Modul dieses Typs gekapselt.

II.4.3.2 Anwendungsfälle von Modulen

Welche Anforderungen das Modulkonzept erfüllen muss, kann mit Hilfe von Anwendungsfällen und deren Analyse ergründet werden. Im Folgenden werden daher Anwendungsfälle für Module konstruiert:

- AF 1: Das Bewegungs-Modul fordert vom Datenerfassungs-Modul Daten zur Ausgabe an, wenn der Ausgabepuffer zur Hälfte geleert ist.
- AF 2: Das Datenerfassungs-Modul wird vom Kamera-Modul über die Ankunft eines neuen Bildes notifiziert. Das Datenerfassungs-Modul kann dieses Bild asynchron beziehen.
- AF 3: Das Datenerfassungs-Modul kann das Vorbereitungs-Modul über die Ankunft neuer Daten informieren.
- AF 4: Das Vorbereitungs-Modul kann Messdaten vom Datenerfassungs-Modul beziehen und an das Topografieberechnungsmodul weiterleiten.
- AF 5: Das Vorbereitungs-Modul kann das Datenerfassungs-Modul und dieses wiederum das Bewegungs-Modul zum Starten der Messung veranlassen.
- AF 6: Mehrere verschiedene Topografieberechnungs-Module stehen zur Verfügung. Aufgrund einer Kategorisierung der Module können die verschiedenen Module aufgelistet und eines ausgewählt und instantiiert werden.
- AF 7: Mehrere Messdatenspeicher-Module stehen zur Verfügung. Es können mehrere gewählt werden, sodass die Topografie zugleich in mehreren Formaten gespeichert wird.
- AF 8: Nach der Inbetriebnahme der Messeinrichtung werden die Einstellungen aller Module gespeichert, sodass diese beim nächsten Start geladen werden und die Module mit diesen Parametern initialisiert werden können.
- AF 9: Nach der Inbetriebnahme der Messeinrichtung ist das Kamera-Modul mit dem Datenerfassungs-Modul verbunden. Beim nächsten Start soll diese Verbindung wiederhergestellt werden.

AF 10: Nach einer Änderung der Konfiguration bricht das Programm den Start mit einem Verweis auf einen falschen Parameter ab. Der Wartungstechniker stellt den Parameter mit einem Texteditor richtig.

II.4.3.3 Analyse der Anwendungsfälle

Die gefundenen Anwendungsfälle werden in diesem Unterkapitel analysiert, sodass allgemeine Anforderungen erstellt werden können, die durch das Modulkonzept erfüllt werden müssen:

1. Aus AF 1 ergibt sich, dass Module als Datenquelle und -senke fungieren können. Dabei kann eine Datensenke aktiv von einer passiven Datenquelle Daten abrufen. In AF 4 übergibt eine Datenquelle aktiv Daten an eine passive Datensenke. Daraus ergibt sich, dass Datenquellen und Datensenken aktiv und passiv arbeiten können. Als aktiv soll ein Modul gelten, wenn dieses eine Funktion eines anderen Moduls aufruft. Das aufgerufene Modul gilt als passiv.
2. Ein Modul kann ein anderes Modul starten und stoppen, wie aus AF 5 hervorgeht.
3. Laut AF 7 kann ein Modul A mit mehreren Modulen des Typs B verbunden werden.
4. Mit AF 6 ergibt sich, dass Module Metainformationen benötigen, etwa zur Angabe der Kategorie oder einer Beschreibung.
5. Das Datenerfassungs-Modul ist eine Datenquelle für kombinierte Bild- und Positionsdaten und eine Datensenke für Bilder. Dieses Modul muss also zugleich die Rolle einer Messdatenquelle und die einer Bilddatensenke übernehmen.
6. Verbindungen zwischen Modulen sollen wiederhergestellt werden.
7. Parameter von Modulen können von anderen Modulen verändert und abgefragt werden.
8. Gespeicherte Parameter können mit einem Texteditor verändert werden.
9. Nur Module mit passenden Schnittstellen können verbunden werden.

II.4.4 Zusammenfassung

In diesem Kapitel werden grundlegende Entwurfskonzepte und allgemeine Anforderungen an die Implementierung des Programms entwickelt. Zu diesen Anforderungen zählen:

- Das Programm muss den Messtechniker bei der Einrichtung des Messobjekts und bei der Messung unterstützen, sodass die Messunsicherheit der zu berechnenden Topografie möglichst gering ist.
- Meldungen über den Ablauf des Programms müssen aufgezeichnet werden.
- Hinweise auf Fehler müssen gegeben werden.
- Parameter müssen unmissverständlich beschrieben und deren Werte überprüfbar sein.
- Die Messung muss vollständig gespeichert werden können. Zu diesem Zweck müssen die Messdaten, alle Einstellungen der beteiligten Komponenten und die Auswertung gespeichert werden können.
- Messungen sollen mit bestehenden Messdaten wiederholt werden können.
- Einzelne Funktionalitäten des Programms sollen einfach ausgetauscht werden können.
- Die Funktionalität soll in ladbare Module gekapselt werden.
- Daten sollen asynchron verarbeitet werden.

Die folgenden Entwurfsentscheidungen werden gefällt und begründet:

- Klassen des Problembereichs werden in Module abgebildet.
- Module werden als ladbare Bibliotheken ausgeführt.
- Daten werden in Threads verarbeitet.
- Auf Ereignisse wird in Threads gewartet.

Basierend auf den in Kapitel II.3 definierten Klassen werden Module im Problembereich definiert. Diese Module dienen in den folgenden Kapiteln als Entwurfsgrundlage. Durch eine Anforderungsanalyse des Modulkonzepts ergeben sich die folgenden Anforderungen an Module:

- Verbindungen zwischen Modulen können gespeichert und wiederhergestellt werden.
- Module können als Datenquelle oder -senke auftreten und Daten aktiv oder passiv weiterleiten oder beziehen.
- Parameter von Modulen können durch andere Module verändert, gespeichert und geladen werden.
- Ein Modul kann mehrere Schnittstellen zugleich implementieren.
- Ein Modul kann mit mehreren anderen Modulen einer passenden Schnittstelle verbunden werden.
- Beschreibungen und Kategorien von Modulen können von diesen bezogen werden.
- Parameter werden so gespeichert, dass diese mit einem Texteditor veränderbar sind.
- Module können gestartet und gestoppt werden.

Das nächste Kapitel befasst sich mit dem Entwurf und der Implementierung des hier erstellten Konzepts.

II.5 Entwurf und Implementierung

Das Entwurfskonzept wird in diesem Kapitel umgesetzt. Die Module und Klassen aus dem Problembereich werden in den technischen Bereich transformiert. Der technische Bereich wird durch die Systemumgebung eingegrenzt, die im folgenden Kapitel kurz dargestellt wird. Darauf folgend wird die Implementierung des Modul-Konzepts beschrieben, das die Basis für alle weiteren Module bildet. Abschließend erfolgt die Beschreibung der einzelnen Module.

II.5.1 Systemumgebung

Die Messeinrichtung besteht im Wesentlichen aus einem Personal Computer (PC) mit einem IEEE 1394 (FireWire) Anschluss und einer Datenerfassungs-Steckkarte mit der Bezeichnung NI PCI-6070E von National Instruments, sowie dem Messgerät selbst. Die für die Implementierung des Programms wichtigen Komponenten des Messgeräts sind die CCD-Kamera und der Piezotranslator, der über einen Laborverstärker angesteuert wird. In Anhang A wird die Messeinrichtung im Detail beschrieben.

Um die Anforderungen an das Programm zu erfüllen und die gegebenen Geräte anzusteuern werden die folgenden Bibliotheken eingesetzt:

- *Comedi*: Das Comedi-Projekt bietet eine einheitliche Schnittstelle zur Ansprache verschiedener Datenerfassungs-Steckkarten.
- *Unicap*: Das Unicap-Projekt bietet eine Bibliothek zur Ansprache verschiedener Kameras mit IEEE 1394 (FireWire) oder USB Schnittstelle.
- *QT*: Dies ist eine Bibliothek zur Programmierung grafischer Benutzeroberflächen.
- *Qwt*: Diese Bibliothek stellt Komponenten zur Anzeige von Diagrammen und andere Elemente zur Erstellung von Benutzeroberflächen bereit.
- *FFmpeg*: Teile des FFmpeg-Projekts werden verwendet, um Bilder in einen Video-Container zu komprimieren.
- *libxml*: Dieses Projekt implementiert einen XML-Parser, der zum Einlesen von Konfigurationsdateien verwendet wird.
- *boost*: Dieses Projekt bietet eine Sammlung von C++ Template-Bibliotheken.
- *boost.log*: Dieses Projekt ist eine Template-Bibliothek, die umfangreiches Logging ermöglicht.

In Anhang C werden diese Bibliotheken kurz beschrieben und Quellen sowie Versionen angegeben.

Als Betriebssystem kommt die Linux-Distribution Ubuntu in Version 9.04 zum Einsatz. Eine Real-Time-Erweiterung wird nicht verwendet. Zur Erstellung des Programms wird die Programmiersprache C++ eingesetzt und mit dem entsprechenden C++ Compiler der Gnu Compiler Collection (*gcc*) übersetzt. Als Buildsystem wird *qmake* aus der QT-Bibliothek (siehe Anhang C.3) eingesetzt. Dieses Buildsystem zeichnet sich durch seine Einfachheit und Plattformunabhängigkeit aus.

Im Folgenden werden einige Stilvorgaben zur Kodierung vorgegeben, die sich an ([50], Kap. 0) orientieren:

- *Klassennamen*: Kamelhöcker-Notation mit Großbuchstaben am Anfang.
Bsp.: BeispielKlasse
- *Funktionen, Enumerierungen und Variablen*: Kamelhöcker-Notation mit Kleinbuchstaben am Anfang.
Bsp.: beispielFunktion()

II.5 Entwurf und Implementierung

- *Member-Variablen*: Kamelhöcker-Notation mit Kleinbuchstaben am Anfang und Unterstrich am Anfang.
Bsp: `_beispielMember`
- *statische Funktionen*: Kamelhöcker-Notation mit Großbuchstaben am Anfang.
Bsp: `BeispielFunktion()`
- *Makros und statische Konstanten*: Großbuchstaben.
Bsp.: `BEISPIEL_MACRO`
- *Containernamen*: Werden in Mehrzahl benannt.
Bsp.: `Kreise, Stühle, usw.`
- *Variablennamen*: Variablennamen sollen kein Typkennzeichen enthalten, sondern die Bedeutung wiedergeben.
- *Dokumentation*: Dokumentiert wird bei Deklaration, also in der Header-Datei, im JavaDoc-Stil wie in [51] definiert.
- *Sprache*: Es wird in Englisch programmiert und dokumentiert.

In den nächsten Kapiteln wird die Implementierung beschrieben.

II.5.2 Implementierung von Modulen

Das Entwurfskonzept sieht ladbare Bibliotheken vor. Bei der Implementierung dieses Konzepts sind einige Punkte zu beachten:

- Jedes Modul bzw. jede Bibliothek muss C-Funktionen exportieren, die eine grundlegende Schnittstelle bilden. Diese Funktionen können im Programm verwendet werden.
- Jede Bibliothek verwendet üblicherweise ein eigenes Speichermanagement. Wird von einem Programm ein Objekt, das in einer Bibliothek im Heap-Speicher instantiiert wurde, verwendet, muss das Objekt von dieser Bibliothek auch gelöscht werden. Andernfalls gilt der Speicherbereich für die Bibliothek noch als belegt und kann nicht neu zugewiesen werden.
- Bibliothek und Programm müssen „binärkompatibel“ sein. Dies ist der Fall, wenn beide Seiten für einen Typ die selbe V-Tabelle (virtual method table) verwenden. Der Compiler erzeugt zu jeder Klassendefinition eine V-Tabelle, in der Zeiger zu allen Methoden und Variablen verwaltet werden. Ändern sich Klassendefinitionen von Objekten, die von Bibliothek und Programm gemeinsam verwendet werden, müssen demnach Bibliothek und Programm neu übersetzt werden.

Die folgenden C-Funktionen werden von Modulen exportiert und bilden die grundlegende Schnittstelle:

- `plugins::PluginInterface * factory(void);`
Diese Funktion erzeugt eine Instanz des Moduls.
- `void deleter(void *plugin);`
Diese Funktion entfernt die Instanz aus dem Speicher.
- `plugins::PluginInformation info(void);`
Diese Funktion retourniert eine Struktur mit Informationen über das Modul.

Mit diesen Funktionen kann ein Modul instantiiert und auch wieder gelöscht werden.

Die im Verwaltungsmodul (siehe Kapitel II.5.3) enthaltene Klasse *PluginManager* lädt und instantiiert Module mit Hilfe der oben gezeigten Schnittstelle. Verwendet werden die Funktionen `factory()` zur Instantiierung und `deleter()` zum Entfernen der Instanz, sodass die Entstehung eines Speicherlochs vermieden wird.

II.5.2.1 Verbindung von Modulen

In Kapitel II.4.3.3 wird unter anderem gefordert, Verbindungen zwischen Modulen zu speichern und wiederherzustellen. Verbindungen sollen nur zwischen zueinanderpassenden Rollen der Module zugelassen werden, von denen jedes Modul mehrere übernehmen kann. Ein Modul soll in einer Rolle auch mit mehreren anderen Modulen in einer passenden Rolle verbunden werden können.

Im technischen Bereich kann eine Rolle durch eine Schnittstelle definiert werden. Die Verbindung mit mehreren Modulen kann durch mehrere Verbindungspunkte in einer Schnittstelle umgesetzt werden.

Um eine Verbindung zwischen Modulen eindeutig zu identifizieren und damit wiederherstellbar zu machen, werden Namen für die beteiligten Module, Schnittstellen und Verbindungspunkte benötigt. Module erhalten vom Programmierer eindeutige Bezeichnungen. Die Modul-Schnittstelle kann mit dessen Typ durch (Runtime Type Information) RTTI innerhalb eines Moduls eindeutig bezeichnet werden. Verbindungspunkte können nicht automatisiert benannt werden, da in jeder Modul-Schnittstelle mehrere Verbindungspunkte des gleichen Typs existieren können. Sie müssen daher durch den Programmierer innerhalb einer Modul-Schnittstellen-Implementierung eindeutig benannt werden. Die Bezeichnung eines Verbindungs-Endpunktes setzt sich demnach folgendermaßen zusammen:

<Modulname><Typ der Schnittstelle><Identifikation des Verbindungspunkts>

Eine Verbindung kann damit durch die Bezeichnungen beider Verbindungspunkte eindeutig angegeben werden.

Die Anforderungen aus Kapitel II.4.3.3 bestimmen weiters, dass Verbindungspunkte als Datenquellen oder Senken fungieren, sowie aktiv oder passiv agieren können. Wenn ein Verbindungspunkt als Quelle fungiert, muss die geeignete Gegenstelle eine Senke sein und umgekehrt. Ist ein Verbindungspunkt aktiv, so muss eine geeignete Gegenstelle passiv sein.

Eine Metapher für passives Erhalten ist ein Stecker und für aktives Liefern die Steckdose. Zusätzlich muss noch die Richtung des Datenflusses durch die Bezeichnungen Quelle und Senke definiert werden. Somit setzen sich die Bezeichnungen für einen Verbindungspunkt wie folgt zusammen:

<Bezeichnung><Source/Sink (Quelle/Senke)><Socket/Plug (Dose/Stecker)>

Sind die Richtungen des Daten- und des Informationsflusses übereinstimmend, kann auf die Angabe von Quelle oder Senke verzichtet werden.

Um sicherzustellen, dass die Verbindungspunkte zusammenpassen, und um die Typen mit RTTI extrahieren zu können, wird ein Muster namens „The Curiously Recurring Template Pattern“ wie in ([52], Kap. 9.8) angewendet. Dieses Muster ermöglicht statischen Polymorphismus, wodurch Klassen ohne gemeinsame Basisklasse in Beziehung gebracht werden können. In diesem Fall werden Socket und Plug, beziehungsweise Source und Sink verbunden.

Ein Socket und ein Plug mit der entsprechenden Source oder Sink bilden jeweils einen Signal-Endpunkt (SignalEndPoint). Die Verbindung von zwei passenden Signal-End-Punkten bildet einen Signalpfad (SignalPath). In Abbildung II.5.1 sind zwei Signal-End-Punkte definiert. Eine generische Klasse Connectable kann mit zwei Typen spezialisiert werden. Connectable realisiert eine Verbindung der Signal-End-Punkte durch einen Zeiger, dessen Typ der generische Parameter To bestimmt. Die Spezialisierung Plug fügt eine abstrakte Methode call() hinzu, die durch einen Socket aufgerufen wird und eine Variable des Typs ValueType entgegennimmt. Das Beispiel in Abbildung II.5.1 zeigt demnach zwei Signal-End-Punkte, die sich durch Austauschen von Zeigern gegenseitig referenzieren können.

II.5 Entwurf und Implementierung

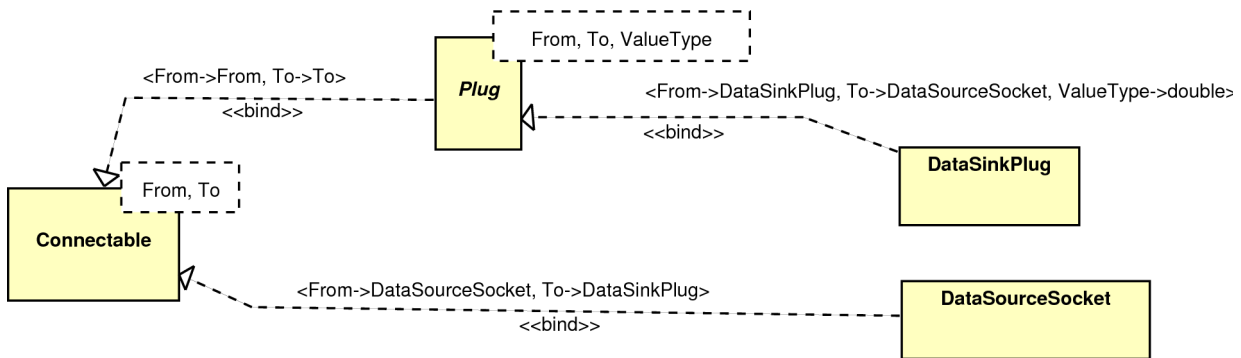


Abbildung II.5.1: DataSourceSocket und DataSinkPlug als Beispiele von Signal-End-Punkten

Das Erzeugen von neuen Signal-End-Punkten entspricht einer Spezialisierung der generischen Typen Connectable und Plug und kann wie in Listing II.5.1 durchgeführt werden.

```

class DataSinkPlug;
class DataSourceSocket :
virtual public Connectable<DataSourceSocket,DataSinkPlug>{};

class DataSinkPlug :
virtual public Plug<DataSinkPlug,DataSourceSocket,double >{};
    
```

Listing II.5.1: Definition von Signal-End-Punkten

Ein Modul soll mehrere Modul-Schnittstellen implementieren können. Innerhalb einer Modul-Schnittstelle sollen mehrere Instanzen des gleichen Signal-End-Punktes verwaltet werden können. Signal-End-Punkte sollen über eine Basis-Schnittstelle auffindbar sein.

Um diese Funktionalität zu erzeugen, werden alle Modul-Schnittstellen von einer abstrakten Basisklasse PluginInterface abgeleitet. Diese enthält Methoden, um die Typen aller Signal-End-Punkte bei Instanziierung zu sammeln. Die gesammelten Signal-End-Punkte können zu einem späteren Zeitpunkt abgefragt werden. Eine Spezialisierung dieser Klasse ist PluginSignalPath. Diese generische Klasse wird wiederum in einem „The Curiously Recurring Template Pattern“ verwendet, um die Schnittstellen-Typen für verbindbare Schnittstellen in Beziehung zu bringen. In Abbildung II.5.2 ist das Klassen-Diagramm der Modul-Schnittstellen DataSourceSocketPlugin und DataSinkPlugPlugin zu sehen. Die generische Klasse PluginSignalPath wird mit den in Listing II.5.1 und Abbildung II.5.1 definierten Signal-End-Punkten parametrisiert. Die in dieser Klasse enthaltenen generischen Methoden ermöglichen auch die Verwaltung von Verbindungen durch die Klasse SignalPathManager aus dem Verwaltungsmodul (siehe Kapitel II.5.3).

In Listing II.5.2 ist die Definition einer Modul-Schnittstelle zu sehen. Im Konstruktor muss der Typ der instantiierten Klasse an die Basisklasse zur Verwaltung übergeben werden. Als generische Parameter kommen Signal-End-Punkte zum Einsatz.

```

class DataSourceSocketPlugin:
virtual public PluginSignalPath<DataSourceSocket,DataSinkPlug>
{
    public: DataSourceSocketPlugin(){ addInterface(&typeid(*this));}
};
class DataSinkPlugPlugin:
virtual public PluginSignalPath<DataSinkPlug,DataSourceSocket>
{
    public: DataSinkPlugPlugin(){ addInterface(&typeid(*this));};
};
    
```

Listing II.5.2: Definition von Modul-Schnittstellen

Diese Konstruktion ermöglicht die Speicherung und Wiederherstellung von Verbindungen zwischen Modulen durch das Verwaltungsmodul zur Laufzeit. In einer GUI (graphical user interface) können dadurch Informationen über das Modul, die implementierten Modul-Schnittstellen und die Signal-End-Punkte angezeigt werden.

Die Definition von neuen Schnittstellen gestaltet sich einfach. Es sind zu diesem Zweck zwei Signal-End-Punkte und zwei Modul-Schnittstellen, wie in Listing II.5.1 und Listing II.5.2 gezeigt, zu definieren.

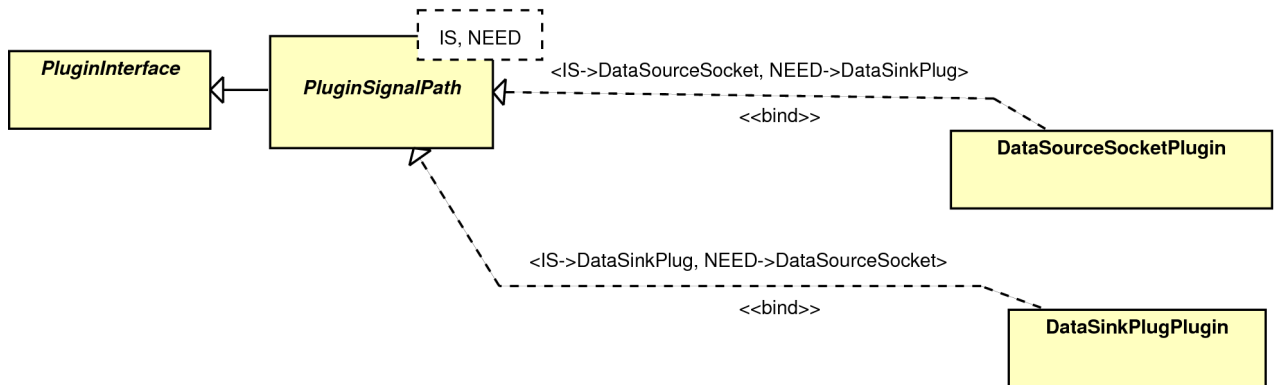


Abbildung II.5.2: Modul-Schnittstelle am Beispiel von *DataSourceSocketPlugin* und *DataSinkPlugPlugin*

II.5.2.2 Parameter von Modulen

Eine weitere Anforderung aus Kapitel II.4.3.3 besteht in der Speicherung und Wiederherstellung von Parametern eines Moduls. Die Parameter sollen dabei von anderen Modulen abgefragt und verändert werden können. Um dem Anwender die fehlerfreie Verwendung dieser Einstellmöglichkeiten zu erleichtern, wird jeder Parameter-Wert mit den folgenden Zusatzinformationen bereichert:

- **Name:** eine Bezeichnung des Parameters; z.B.: Belichtungszeit
- **Beschreibung:** eine Beschreibung des Parameters; z.B.: Zeitdauer, in der ein Bild belichtet wird.
- **Physikalische Einheit:** die Einheit des Parameters als Text; z.B.: Sec.
Nicht alle Parameter haben eine physikalische Einheit.
- **Kategorie:** Dieser Text dient zur Gruppierung von Parametern. Bei vielen Parametern kann dies die Übersichtlichkeit bei der Anzeige im GUI verbessern.
- **Kommentar:** ein Kommentar des Entwicklers. Hier können eine Berechnung, eine Referenz oder ein Hinweis angegeben werden, der die Bedeutung bzw. Verwendung des Parameters erläutert.
- **Wert:** der Wert des Parameters.
- **Maximalwert:** der maximale Wert des Parameters.
- **Minimalwert:** der minimale Wert des Parameters.
- **Standardwert:** die Standardeinstellung, die vom Entwickler empfohlen wird.
- **Schrittweite:** Viele Parameter dürfen nur bestimmte Werte annehmen. Die Belichtungszeit kann etwa nur in Schritten von 0,005 Sekunden verändert werden.
- **Automatisch:** kennzeichnet von der Hardware festgelegte Parameter.
- **Auswahl von Werten:** eine Auswahlliste von möglichen Werten, von denen ein Wert gewählt werden kann. Beispielsweise kann die Polarität eines digitalen Eingangs 'active high' oder 'active low' sein.
- **Identifikation:** Um die Parameter automatisch verarbeiten zu können, muss jeder gespeicherte Parameter eine eindeutige Identifikation beinhalten.

In der Struktur Parameter werden obige Angaben vereint. Die etwas gekürzte Struktur ist in Listing II.5.3 zu sehen. Zu oberst ist ein Typ `value_type` als `boost::variant` definiert. Dieser generische Typ ermöglicht die typsichere Speicherung von unterschiedlichen Datentypen in einer Variable. In diesem Fall können in einem `value_type` die Werte der Typen `int`, `double` und `string` also ganze Zahlen, reelle Zahlen und Zeichenketten gespeichert werden.

```

struct Parameter {
    typedef boost::variant<int,double,std::string> value_type;

    /** An index to count through. */
    unsigned int index;
    /** Unique ID to determine device and parameter this will configure. */
    std::string id;
    /** Human readable name of the parameter. */
    std::string name;
    /** Human readable description of the value. */
    std::string description;
    /** Unit of the value (Nm/s2) */
    std::string unit;
    /** For grouping related parameter. */
    std::string category;
    /** Comment of the developer to clarify usage. */
    std::string comment;
    /** The value it self. */
    value_type value;
    /** The value must not be less than this. */
    value_type min;
    /** The value must not be larger than this. */
    value_type max;
    /** Default value. */
    value_type defaultValue;
    /** Step size from min to max. */
    value_type stepping;
    /** Value set by hardware 1=on, 0=off, -1=no automation */
    short automatic;

    typedef std::vector<std::pair<value_type,std::string> > Possibilities;
    /** List of possible values. */
    Possibilities possibleValues;
}

```

Listing II.5.3: gekürzte Parameter-Struktur

Jedes Modul implementiert eine abstrakte Klasse Parametrisable, da alle Modul-Schnittstellen über die Klasse Plugininterface von Parametrisable abgeleitet sind, wie in Abbildung II.5.3 zu sehen ist. In Parametrisable werden Methoden definiert, durch welche Module Parameter erhalten und liefern können.

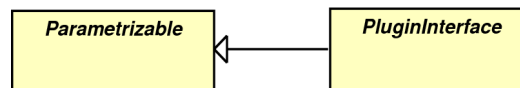


Abbildung II.5.3: Die Schnittstelle zur Parametrisierung; muss von jedem Modul implementiert werden.

Wie die Parameter im Modul verarbeitet werden, ist dem Entwickler überlassen. Eine einfache Möglichkeit besteht in der Definition eines assoziativen Containers, der die Identifikation des Parameters mit einer Rückruf-Methode verbindet. Wird ein Parameter von außen verändert, so kann mit der Identifikation die entsprechende Methode gefunden und ausgeführt werden. Diese Funktionalität wird unter anderem in der Klasse ParameterContainer zur Verfügung gestellt. Zusätzlich wird hier auch der Wertebereich bei Veränderung des Parameters geprüft.

II.5 Entwurf und Implementierung

Aufgrund der dem Parameter-Wert beigefügten Zusatzinformationen und die Verwendung von `boost::variant` ist die Klasse `Parameter` flexibel einsetzbar. So können etwa verschiedene Typen durch Auswahllisten oder Wertebereiche mit kontinuierlichen oder diskreten Schrittweiten abgebildet werden. `Parameter` können auch zum Auslösen von Aktionen verwendet werden.

II.5.3 Verwaltungsmodul

Dieses Modul bildet den Kern des Programms. Hier befindet sich der Einstiegspunkt, weshalb dieser Teil des Programms als WLI bezeichnet wird. Im Verwaltungsmodul werden Dienste und Schnittstellen für andere Module angeboten.

Eine Aufgabe des Verwaltungsmoduls ist das Laden von Modulen und die Bereitstellung von Referenzen zu diesen Modulen. Diese Aufgabe ist in der Klasse `PluginManager` implementiert. Das Singleton-Muster aus ([53], Seite 157 ff.) stellt sicher, dass nur eine Instanz dieser Klasse existiert. Die Klasse `PluginManager` stellt Methoden bereit, um Module zu laden, zu instantiieren, zu suchen und zu registrieren. Jedes Modul wird nur einmal instantiiert. Alle weiteren Anforderungen werden durch Referenzen auf die selbe Instanz bedient. In Listing II.5.4 sind die wichtigsten statischen Methoden dieser Klasse angegeben. Mit `LoadPlugin()` kann eine Bibliothek geladen werden, `GetPluginInfos()` retourniert einen Container mit `PluginInformation`-Objekten aller geladenen Module, und `Instantiate()` gibt eine Referenz auf eine Instanz des bezeichneten Moduls zurück.

```
/** Pointer to a plugin. */
typedef boost::shared_ptr<PluginInterface> Pplugin;
/** Container of plugin pointers. */
typedef std::vector<PPlugin > Plugins;
/**
 * Loads a plugin. This is not instantiation!
 * @param filename File name of the plugin.
 * @return Information about the plugin.
 */
static PluginInformation LoadPlugin(std::string &filename)
    throw(std::exception,WLILoadPluginException);
/** Get all information objects of all loaded plugins
 * without the need of an instance of PluginManager.
 */
static PluginInfos GetPluginInfos();
/**
 * Instantiate a plugin identified by the parameters
 * without the need of an instance of PluginManager.
 * @param name Name of the plugin.
 * @return A pointer to the instance.
 */
static PPlugin Instantiate(const std::string &name)
    throw(std::exception,WLIInstantiateException);
```

Listing II.5.4: Auswahl von öffentlichen Methoden der Klasse `PluginManager`

In Kapitel II.5.2.2 wurden bereits die Parameter von Modulen besprochen. Jedes Modul stellt auf Anfrage einen Container mit Parameter-Objekten bereit, die durch die Klasse `ParameterManager` als XML-Datei gespeichert werden können. Das Laden und Verteilen der Parameter wird mit Hilfe der in Anhang C.6 vorgestellten `libxml` und der Boost-Bibliothek `Statechart` durchgeführt. Die Klasse `ParameterManager` ist ebenfalls als Singleton ausgeführt.

`Libxml` ruft beim Parsen einer XML-Datei bei jedem Tag eine Rückruf-Funktion auf, die einen Statusübergang des mit `Statechart` erzeugten endlichen Automaten auslöst. Ein als XML-kodierter Parameter enthält neben einer Identifikation auch den entsprechenden Modulnamen. Ist ein Parameter vollständig eingelesen, wird im endlichen Automaten eine Referenz des betreffenden Moduls von `PluginManager` angefordert und der Parameter diesem Modul übergeben. Von der Klasse `ParameterManager` werden analog zum Speichern und Laden von Parametern auch Verbindungen zwischen den Modulen gespeichert und wiederhergestellt. Die Daten werden zum Speichern von der ebenfalls als Singleton ausgeführten Klasse `SignalPathManager` abgefragt. Auch die Wiederherstellung der Verbindungen übernimmt diese Klasse.

```

/**
 * Write parameters of one plugin to a file specified by it's file name.
 * @param filename Name of the file.
 * @param plugin Plugin to persist.
 * @return 0 if OK
 */
static short WritePluginParameters(std::string filename,
                                   std::string pluginname);
/**
 * Restores parameters to plugins.
 * @see ConnectionManager
 * @param filename File to read signal paths.
 * @return 0 if OK
 */
static short ReadAndSetPluginParameters(std::string filename);
/**
 * Write signal paths using a file name.
 * @param filename File name to write to.
 * @return 0 if OK
 */
static short WriteSignalPaths(cstd::string filename);
/**
 * Restores signal paths, which are connections between plugins.
 * @see ConnectionManager
 * @param filename File to restore from.
 * @return 0 if OK
 */
static short ReadAndSetSignalPaths(std::string filename);

```

Listing II.5.5: Auswahl von öffentlichen statischen Methoden der Klasse `ParameterManager`

Die Aufzeichnung von Meldungen erfolgt durch die Bibliothek `boost.log` (siehe Anhang C.8). Die Anwendung erfolgt wie in Listing II.5.6 gezeigt. Meldungen können in verschiedener Weise gefiltert, gespeichert und angezeigt werden.

II.5 Entwurf und Implementierung

```
//-- Header --
#include "wlilogging.h"

class Shape
{
    wli::WLILogger _log;
public:
    Shape(); // Konstruktor
    void doLog(); // Methode
}

/-- CPP -
Shape::Shape() {
    ADD_LOG_TAG(_log); // registriere Klasse
}

VisualPlugin::doLog() {
    BOOST_LOG_FUNCTION(); // Methoden-Signatur zu Meldung
    LOG_DBG(_log) << "Logging test."; // Debug-Log-Meldung
}
```

Listing II.5.6: Verwendung von boost.log

II.5.4 Vorbereitungs-Modul

Dieses Modul unterstützt den Messtechniker bei der Einrichtung des Messobjekts und implementiert das Problembereichs-Modul Vorbereitung aus Kapitel II.4.3.1.

Das Messobjekt ist eingerichtet, wenn der interessierende Bereich im Fokus liegt, normal zur optischen Achse und zentriert ausgerichtet ist, und wenn die Signalqualität akzeptabel ist.

Durch ein Live-Bild werden Fokussierung und Zentrierung unterstützt.

Die Verkippung wird mit topografischen Daten berechnet, die durch ein Topografieberechnungs-Modul bereitgestellt werden. Da die Schätzung der Ebenen-Parameter durch Ausreißer stark beeinträchtigt werden kann, wird durch die Methode des „Random Sample Consensus (RANSAC)“ (siehe Anhang B.1) die Menge der berechneten Punkte der Topografie in Inlier und Outlier eingeteilt. Als Inlier gelten Punkte, deren Entfernung von der gesuchten Ebene einen Schwellwert nicht überschreitet. Alle anderen Punkte sind Outlier und werden ignoriert. Mit der Methode der Singulärwertzerlegung (siehe Anhang B.2) werden nun die Parameter einer Ebene mit dem kleinsten quadratischen Abstand zur den in ROIs gewählten Bereichen geschätzt. Die Übereinstimmung des Normalvektors dieser Ebene mit der optischen Achse ist ein Maß für die Verkippung.

Die Qualität des Messsignals ist akzeptabel, wenn Spitzenwerte noch im Messbereich liegen, und wenn das Korrelogramm ausreichend aus dem Rauschen hervortritt. Ist ein Signal abgeschnitten, kann dies einfach detektiert werden. Ein Maß für die Amplitude des Korrelogramms ist das Verhältnis von maximaler zu mittlerer Intensität (siehe Kapitel I.4.4).

In Abbildung II.5.4 ist ein Klassendiagramm dieses Moduls zu sehen. Die Klasse SetupPlugin bildet die Schnittstelle des Moduls und implementiert mehrere Modulschnittstellen. Die Schnittstelle VisualPlugin ermöglicht das Anzeigen und Verbergen der GUI. Mit AcquisitionPlugPlugin wird der Datenaustausch mit dem DatenerfassungsManagement-Modul ermöglicht, und durch die Schnittstelle TopographyPlugPlugin können fertig berechnete Topografien bezogen werden. Die Schnittstelle AcquisitionSocketPlugin dient zur Weiterleitung von Messdaten an ein Topografieberechnungs-Modul.

Die Klasse SetupView kapselt die GUI des Moduls. Mit Hilfe der Klasse GLPictureView wird ein Live-Bild der Kamera angezeigt. Diese Klasse ermöglicht das Einzeichnen von ROIs in das Live-Bild, sodass Bereiche die in einer Ebene liegen sollen, markiert werden können. Mit topografischen Daten aus diesen Bereichen wird die Verkippung berechnet.

Das Ergebnis einer Topografieberechnung kann mit der Klasse GLElevationView angezeigt werden. Die Klasse Setup hat die Funktion eines Vermittlers ([53], S. 385 ff). Erhaltene Bilder und Positionsdaten der Klasse AcquisitionPlugConnector werden einerseits zur Topografieberechnung über die Schnittstellenklasse AcquisitionSocketConnector und andererseits in einem Ausführungspfad der Klasse SetupThread zur Signalbeurteilung durch die Klasse SignalEvaluator zugeführt. Topografische Daten werden von einem externen Modul berechnet und gelangen durch die Schnittstellenklasse TopographyCalculationConnector in das Setup-Modul. Im Ausführungskontext von SetupThread wird die Verkippung mit diesen topografischen Daten in der Klasse SVDWithRANSACTiltEstimator berechnet.

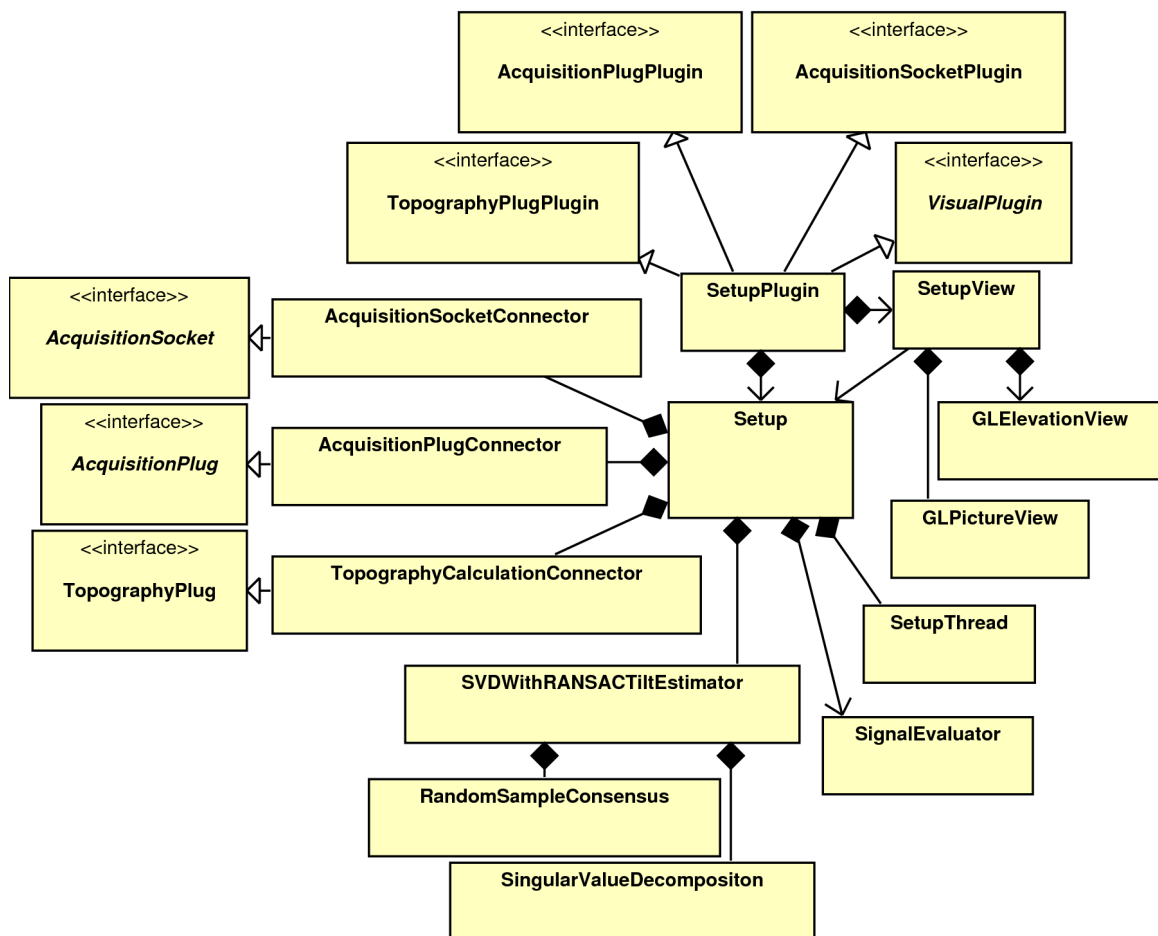


Abbildung II.5.4: Klassendiagramm des Setup-Moduls

II.5.5 Mess-Modul

Dieses Modul implementiert das Problembereichs-Modul Messung aus Kapitel II.4.3.1.

Ein Klassendiagramm dieses Moduls ist in Abbildung II.5.5 zu sehen. Die Klasse `MeasurementPlugin` bildet die Modul-Schnittstelle und implementiert die Schnittstellen `AcquisitionSocketPlugin` zur Weitergabe von Messdaten, `AcquisitionPlugPlugin` zum Empfangen von Messdaten, `TopographySocketPlugin` zur Weitergabe von topografischen Daten, `TopographyPlugPlugin` zum Empfangen von topografischen Daten sowie `VisualPlugin`, um die Anzeige einer GUI zu gestatten. Die Klasse `MeasurementView` ermöglicht die Parametrierung sowie die Verfolgung der Messung durch die Klasse `GLPictureView` und die Anzeige der berechneten Topografie durch `GLElevationView`. Die Klasse `Measurement` ist dem Vermittler-Muster aus ([53], S. 385 ff) angelehnt. Hier werden Meldungen über neue Messdaten aus der Klasse `AcquisitionPlugConnector` weitergeleitet an `AcquisitionSocketConnector`. Über die Instanzen von `AcquisitionSocketConnector` werden Module zur Messdatenspeicherung und zur Topografieberechnung mit Messdaten versorgt. Mit der Klasse `TopographyCalculationConnector` können berechnete Topografien aus dem Topografieberechnungs-Modul eingelesen und über `TopographyRecorderConnector` an ein oder mehrere Module zur Speicherung der Topografie weitergeleitet werden.

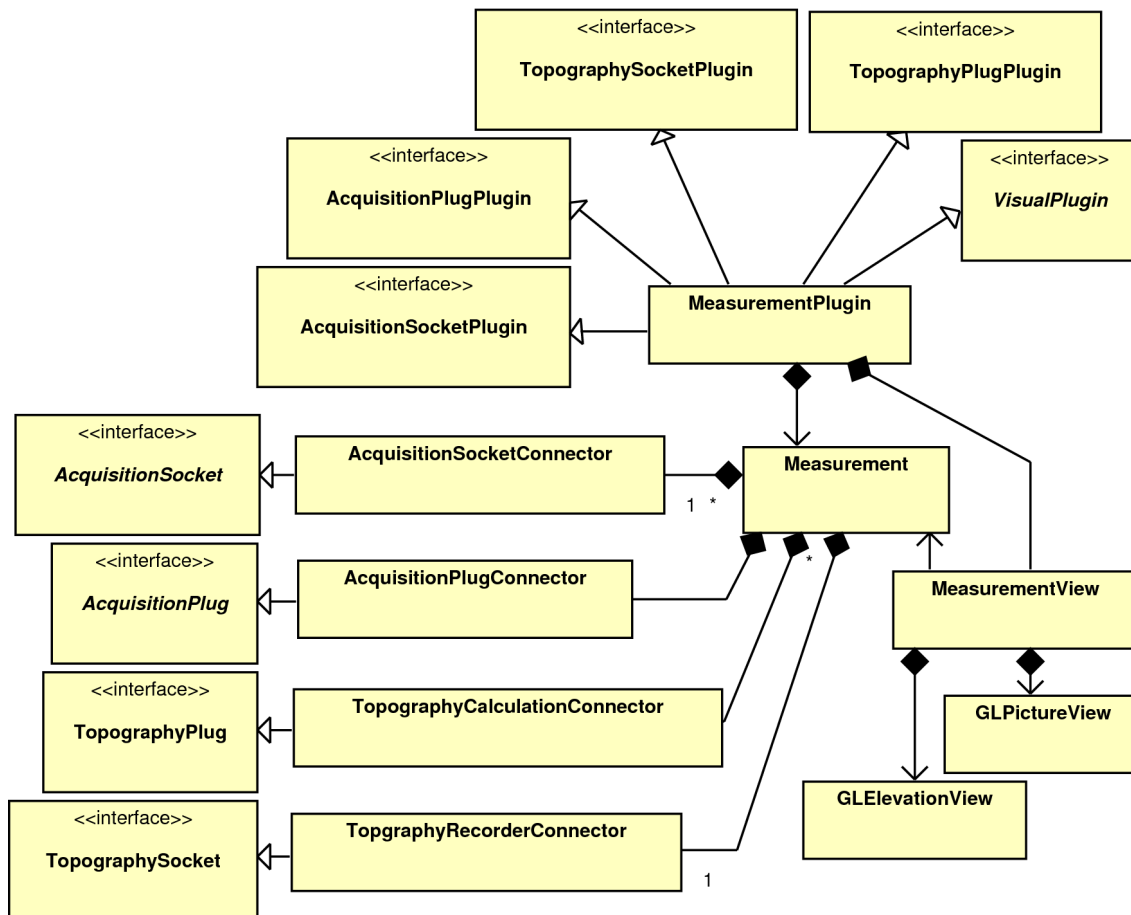


Abbildung II.5.5: Klassendiagramm Mess-Modul

II.5.6 Wiedergabe-Modul

Dieses einfache Modul ermöglicht das Ersetzen der Datenquelle des Messmoduls, sodass aufgezeichnete Messungen wiederholt werden können. Dieses Modul implementiert das Problembereichs-Modul Wiedergabe aus Kapitel II.4.3.1.

Zu diesem Zweck wird von der Klasse `DataPlaybackPlugin` die Modul-Schnittstelle `AcquisitionSocketPlugin` implementiert. In der GUI-Klasse `DataPlaybackView` kann eine Messdatendatei ausgewählt, und eine Abspielgeschwindigkeit angegeben werden. Die Klassen `DataFromFile` und `PictureFromStream` werden zum Einlesen der gespeicherten Messdaten verwendet. Die Rate der Signalisierung neuer Bilder wird durch die Abspielgeschwindigkeit vorgegeben. Einstellungen des Mess-Moduls durch die Schnittstelle `AcquisitionSocket` werden ignoriert.

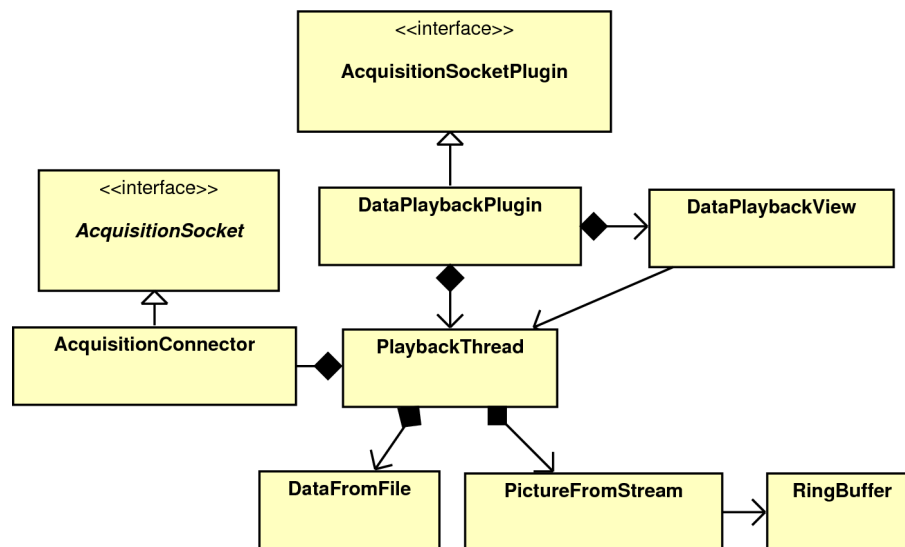


Abbildung II.5.6: Klassendiagramm Wiedergabe-Modul

II.5.7 Kamera-Modul

Das Kamera-Modul verwendet Unicap (siehe Anhang C.2), um Einstellungen der Kamera zu verändern und Bilder zu empfangen. Als Bildformat wird Y800⁷, eine Matrix von 8-Bit-Werten, verwendet. Zu jeder weiteren Bildverarbeitung wird dieses Bildformat vorausgesetzt.

In Abbildung II.5.7 ist ein Klassendiagramm dieses Moduls zu sehen. Die Klasse `CameraPlugin` implementiert die Modul-Schnittstelle `PictureSocketPlugin` und hält Zeiger auf die Klassen `CameraView` und `Camera`. Die Klasse `CameraView` ist für die Anzeige der Parameter einer Kamera zuständig und kann auch ein Live-Bild durch Instantiierung von `PictureView` starten. Da mehrere Kameras am Computer angeschlossen sein können, kann die Klasse `CameraPlugin` mehrere Instanzen der Klasse `Camera` halten. Die Klasse `Camera` kapselt die Unicap-Programmierschnittstelle. Hier werden alle Parameter der Kamera in Parameter-Objekte verpackt und an die GUI oder über die Modul-Schnittstelle nach außen weitergeleitet. Bei Erhalt eines Bildes wird dieses in den am längsten nicht beschriebenen Speicherplatz eines Ringpuffers geschrieben. Anschließend werden alle verbundenen Module über den Erhalt eines neuen Bildes informiert. Die Verwaltung des Ringpuffers übernimmt die Klasse `RingBuffer`.

Da das Kamera-Modul mehrere externe Module mit Bildern versorgen kann, können mehrere Instanzen der Klasse `CameraSocket` existieren, mit denen externe Module angebunden werden.

⁷ Detaillierte Informationen über Video-Bildformate sind unter <http://www.fourcc.org/> erhältlich.

II.5 Entwurf und Implementierung

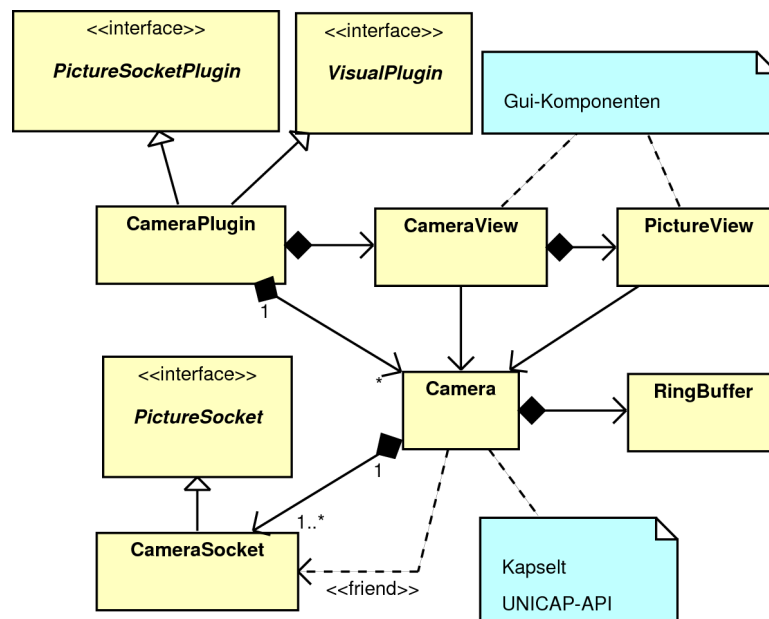


Abbildung II.5.7: Klassendiagramm des Kamera-Moduls

Bei Aufnahme eines Bildes werden die in Abbildung II.5.8 dargestellten Aktionen ausgeführt. Eine Methode der Klasse `Camera` wird dem Unicap-API als Rückruffunktion übergeben. Diese Methode veranlasst das Speichern des neuen Bildes im Ringpuffer und signalisiert danach alle verbundenen Module durch Aufruf der Methode `call()` der `PicturePlug`-Schnittstelle. Ein verbundenes Modul kann nun mit der Methode `fetchPicture()` der Schnittstelle `PictureSocket` das neue Bild beziehen.

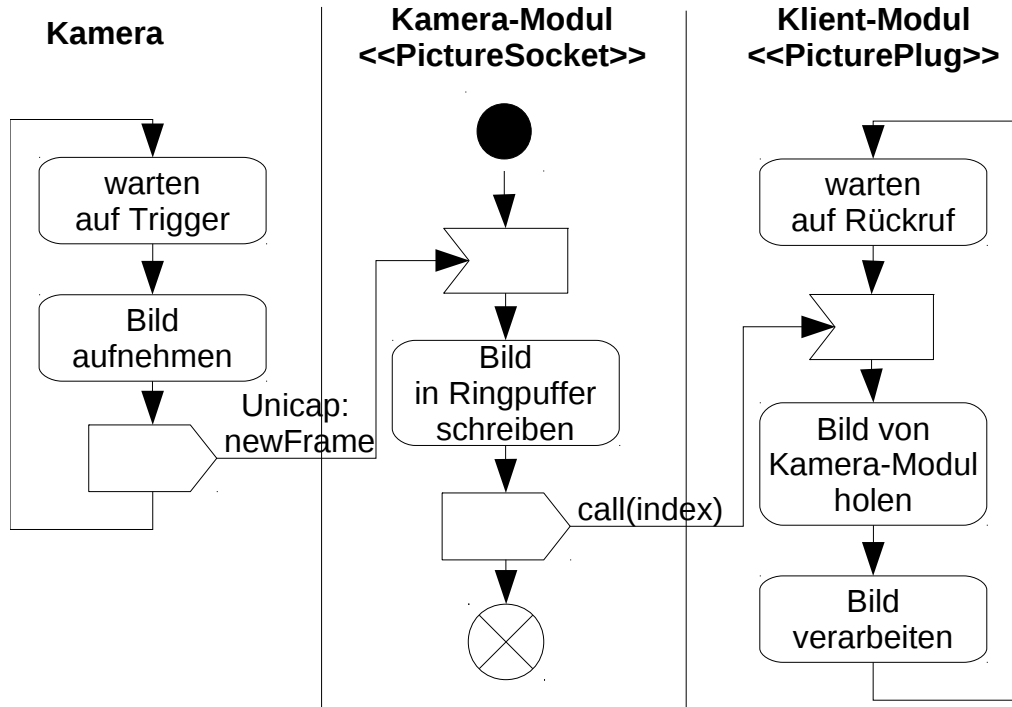


Abbildung II.5.8: Ablauf des Bildempfangs

Es ist unbedingt erforderlich, dass das Bild nicht in der Rückrufmethode `call()` der `PicturePlug`-Schnittstelle bezogen wird. Das externe Modul muss das Bild in einem eigenen Thread holen. Andernfalls wird Unicap für den Empfang weiterer Bilder blockiert, weil `call()` im Kontext von Unicap ausgeführt wird.

II.5.8 IO-Modul

Dieses Modul implementiert die Klasse `Bewegung` aus dem Problembereich (siehe Kapitel II.4.3.1). Das IO-Modul steuert die Datenerfassungs-Steckkarte mit Hilfe des `Comedi`-Pakets (siehe Anhang C.1) an. `Comedi` teilt die Hardware in `Device`, `Subdevice` und `Channel` ein, wobei ein `Device` mehrere `Subdevices` und ein `Subdevice` mehrere `Channels` enthalten kann. Dieser Struktur wird im IO-Modul gefolgt, wie im Klassen-Diagramm in Abbildung II.5.9 zu sehen ist. Da `Device`, `Subdevice` und `Channel` sehr eng zusammenarbeiten, deklarieren sich diese Klassen gegenseitig als `friend`, wodurch sie auch auf geschützte Methoden und Variablen zugreifen können.

Das IO-Modul muss primär Analog-Daten ausgeben. Mit der Datenerfassungs-Steckkarte können jedoch auch Analog-Daten eingelesen werden. Es werden daher die Modul-Schnittstellen `DataSourceSocketPlugin` und `DataSinkSocketPlugin` implementiert. Durch die Schnittstelle `DataSourceSocketPlugin` kann das Modul als Datenquelle fungieren, die aktiv Daten weitergibt. Die Schnittstelle `DataSinkSocketPlugin` erlaubt dem Modul, als Datensenke zu fungieren, die aktiv Daten anfordert.

Die Modul-Schnittstelle `VisualPlugin` ermöglicht das Anzeigen und Verbergen einer GUI.

Eine `Channel`-Instanz bildet einen Datenein- oder -ausgang. Die Klasse `Channel` bietet sich daher als `Signal-End-Punkt` an und implementiert die `Signal-End-Punkt-Schnittstellen` `DataSourceSocket` für die Rolle als Datenquelle und `DataSinkSocket` für die Rolle als Datensenke.

Mit `Comedi` können Daten synchron und asynchron eingelesen und ausgegeben werden. Die synchrone Ein-/Ausgabe blockiert allerdings den jeweiligen Ausführungspfad, bis die Daten vollständig ein-/ausgegeben sind. Soll kontinuierlich ein-/ausgelesen werden, können durch diese Vorgehensweise Daten verloren gehen, wenn der Ein-/Ausgabepuffer voll bzw. leer wird und nicht rasch genug geleert bzw. gefüllt werden kann. Bei asynchroner Ein-/Ausgabe steht mehr Zeit zum Auffüllen des Puffers zur Verfügung, da bereits vor der vollständigen Ein-/Ausgabe Daten angefordert werden können.

Asynchrone Ein-/Ausgaben werden von `Comedi` durch sogenannte Kommandos (`Command`) unterstützt. Vor dem Start eines Kommandos müssen eine Kommando-Struktur (`comedi_cmd`) mit gültigen Werten befüllt, ein Puffer bereitgestellt und das Kommando getestet werden. Nach dem Start eines Kommandos wartet der ausführende Thread, bis neue Daten zur Verfügung stehen oder angefordert werden. Diese Funktionalität wird durch die Klassen `Subdevice`, `AsynchronousSubdevice`, sowie `AnalogOutAsyncSubdevice` und `AnalogInAsyncSubdevice` implementiert.

II.5 Entwurf und Implementierung

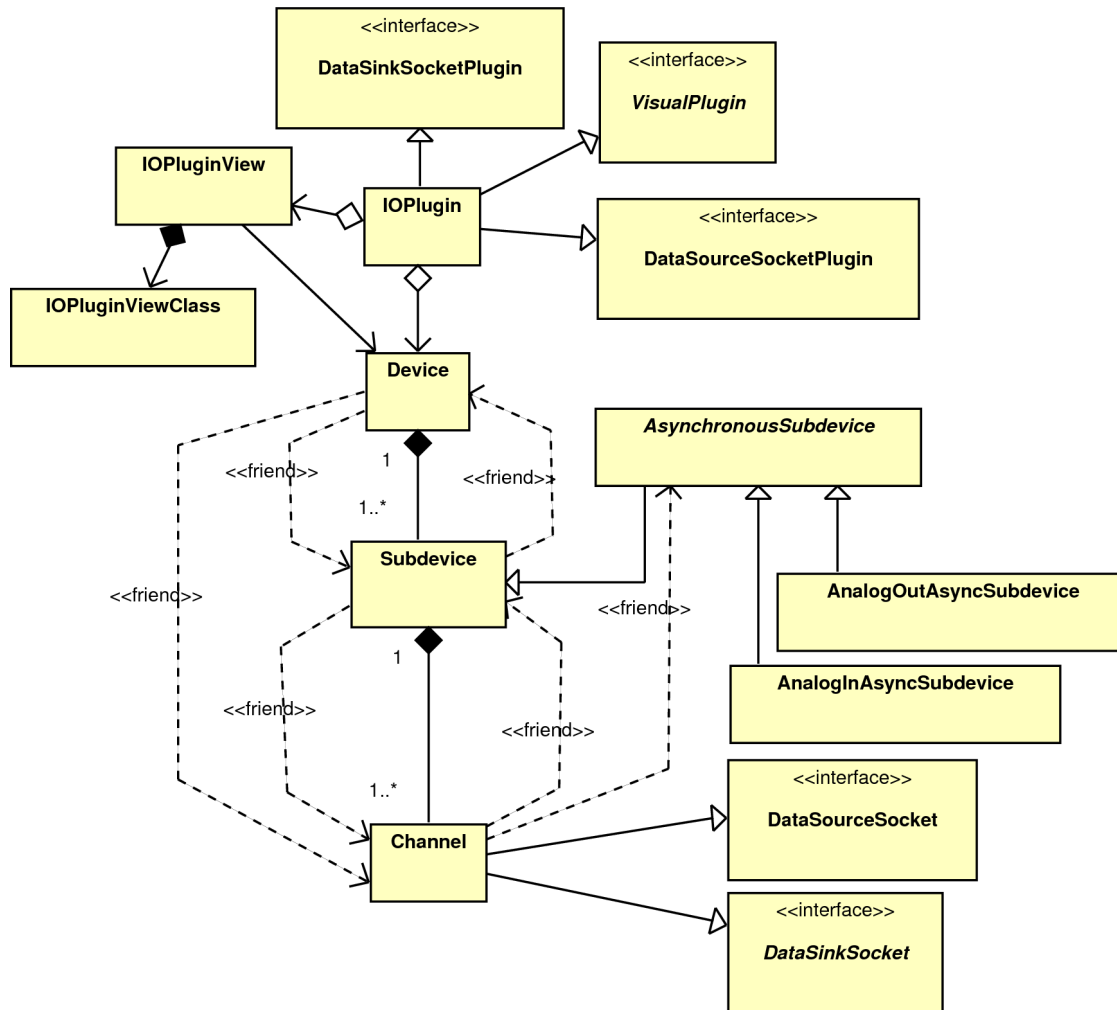


Abbildung II.5.9: Klassendiagramm IO-Modul

Der Ablauf von Aktionen bis zum Start der Datenausgabe ist in Abbildung II.5.10 dargestellt. Ein äußeres Modul ist zu Beginn über ein oder mehrere Signal-End-Punkte mit dem IO-Modul verbunden. Das verbundene Modul legt über die Schnittstelle `DataSinkSocket` die Ausgaberate fest, initialisiert die Hardware durch Aufruf von `prepare()` und startet zuletzt durch Aufruf von `start()` die Datenausgabe.

Im IO-Modul wird durch den Aufruf von `prepare()` ein Comedi-Kommando vorbereitet. Dies umfasst die Sperre des Subdevice, das Befüllen der Kommando-Struktur und das Testen und anschließende Starten des Kommandos. Das Kommando weist Comedi an, mit der Datenausgabe zu beginnen, sobald ein Ereignis empfangen wird. Der Aufruf von `start()` durch das äußere Modul löst dieses Ereignis aus.

Der Ausgabepuffer wird gefüllt, indem für jeden Wert die Methode `call()` der `DataSourcePlug`-Schnittstelle des äußeren Moduls aufgerufen wird. Der erhaltene Wert wird in eine Digital-Analog-Wandler-Stufe umgerechnet und in einen Puffer eingetragen. Ist die Hälfte des Puffers ausgegeben, werden von Comedi erneut Daten angefordert, die wie oben bereitgestellt werden.

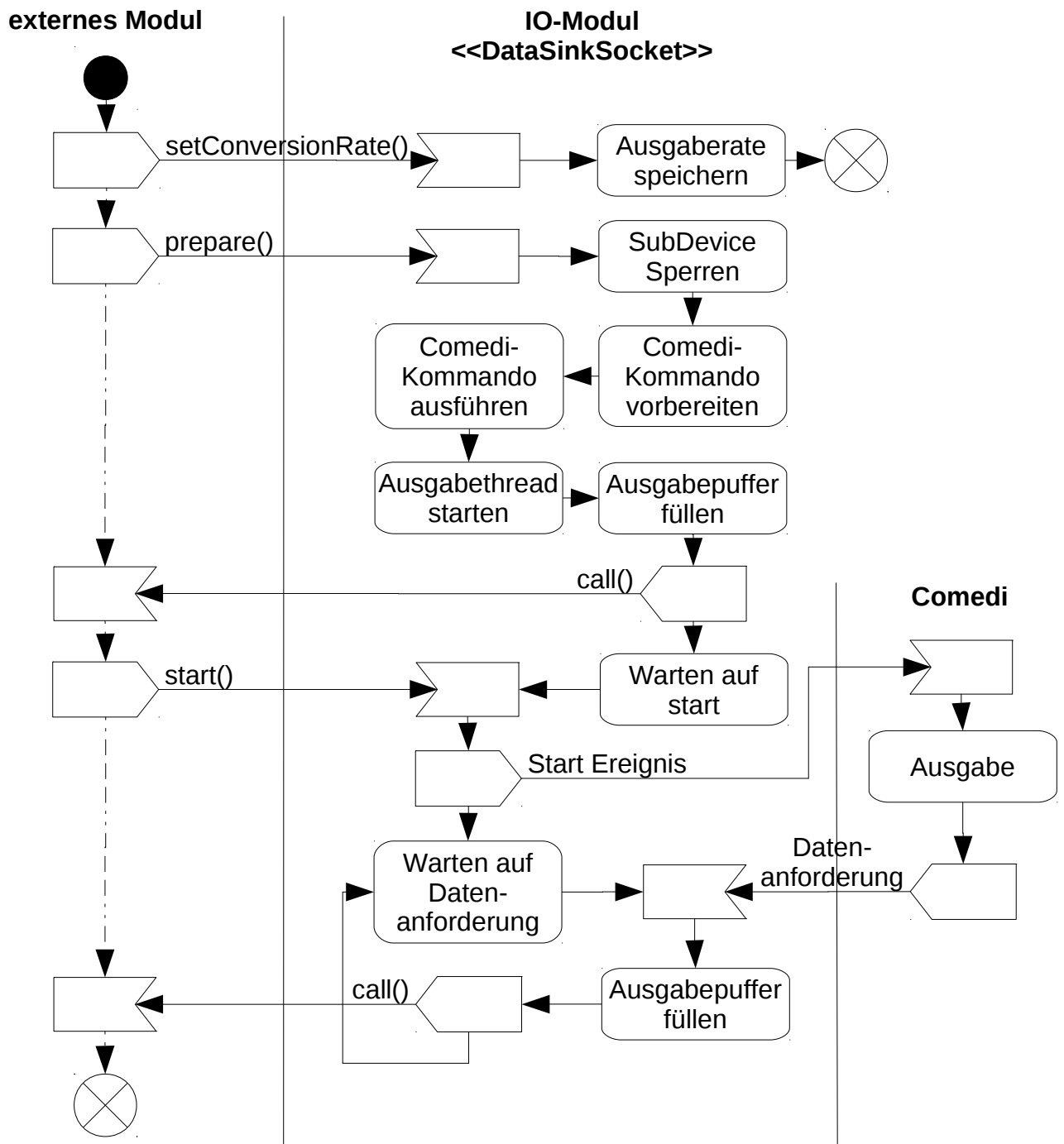


Abbildung II.5.10: Start der Datenausgabe im IO-Modul

II.5.9 DatenerfassungsManagement-Modul

Dieses Modul implementiert das Problembereichs-Modul Datenerfassung aus Kapitel II.4.3.1. Es führt Positionen und Bilder zusammen und gibt diese gemeinsam an ein äußeres Modul weiter. Dieses Modul berechnet zusätzlich Positionierungsdaten zur Weitergabe an das IO-Modul.

Positionsdaten können noch nicht direkt ermittelt werden. Ein zukünftiger Ausbau des Messgeräts soll dies ermöglichen. Um dennoch Bilddaten mit Positionsdaten verknüpfen zu können, muss die Belichtung der Kamera mit der Positionierung des Messobjekts synchronisiert werden. Dies erfolgt durch gleichzeitige Ausgabe eines Positionierungs- und eines Belichtungs-Auslöse-Signals (Triggersignal) auf zwei Analogausgängen der Datenerfassungs-Steckkarte. In einem Container werden die Belichtungspositionen gespeichert, sodass einlangende Bilder aufgrund ihrer Reihenfolge den Positionen zugeordnet werden können. Durch diese enge Verflechtung von Positionierung und Bildaufnahme ist in diesem Modul auch die Berechnung des Verfahrensprofils implementiert.

Die Angabe des Verfahrensprofils erfolgt durch Wegpunkte (WayPoints). Die in Kapitel I.5 vorgestellten Auswertemethoden benötigen lineare Verschiebungen. Daher ist die Verbindung der Wegpunkte durch Geradenstücke implementiert. Es besteht jedoch die Möglichkeit, auf Polynome mit Anschlusskriterien wie etwa die Gleichheit der ersten Ableitung zu erweitern. Mit Hilfe von Wegpunkten können einfache Schleifen realisiert werden. An jedem Wegpunkt kann die Messdatenaufnahme für eine definierte Zeitspanne unterbrochen werden. Damit können aufwendige Berechnungen ausgeführt werden, ohne die Kontinuität der Messdatenaufnahme zu gefährden. Durch diese Definition des Verfahrensprofils werden Langzeitmessungen möglich, die Aufschluss über das dynamische Verhalten von mikroskopischen Oberflächen geben können.

In Abbildung II.5.11 ist das Klassendiagramm dieses Moduls dargestellt. Die Klasse AcquisitionManagerPlugin bildet die Modul-Schnittstelle und implementiert die Schnittstellen DataSourcePlugPlugin, um Daten zur Ausgabe bereitstellen zu können, PicturePlugPlugin, um Bilder vom Kamera-Modul erhalten zu können, AcquisitionSocketPlugin, um Daten an nachfolgende Module weitergeben zu können und VisualPlugin um die Anzeige einer GUI zu ermöglichen. Dahinter übernimmt die Klasse AcquisitionManager das Halten der Parameter und die Vermittlung von Daten und Ereignissen in Anlehnung an das Vermittler-Muster aus ([53], S. 385 ff).

Die Klasse Translation berechnet aus einer Liste von WayPoints ein Verschiebeprofil. Dazu werden einige Kenndaten benötigt, die über die GUI AcquisitionManagerView oder Parameter bereitgestellt werden können:

- *Distanz pro Volt*: Dieser Wert gibt jene Distanz an, die der Objektträger, bei einer Spannungsänderung von einem Volt am Analog-Digital-Wandler, bewegt wird. Dabei ist die Spannungsverstärkung durch den Labor-Verstärker zu berücksichtigen.
- *Mittlere Wellenlänge*: Dieser Wert gibt die mittlere Wellenlänge der Lichtquelle an.
- *Belichtungszeit*: Dieser Wert gibt die Belichtungszeit der Kamera an.
- *maximale Verschiebung während der Belichtungszeit*: Dieser Wert begrenzt die Verschiebung des Messobjekts während der Belichtung.
- *Abtastpunkte pro Wellenlänge*: Dieser Wert legt die Anzahl der Abtastungen innerhalb einer Wellenlänge fest.

Die Klasse Translation berücksichtigt zusätzlich zu den obigen Parametern auch die Puffergröße, die Abtastrate und den Wertebereich des Analog-Digital-Wandlers. Diese Werte werden über die Schnittstelle DataSinkSocket des verbundenen IO-Moduls bezogen.

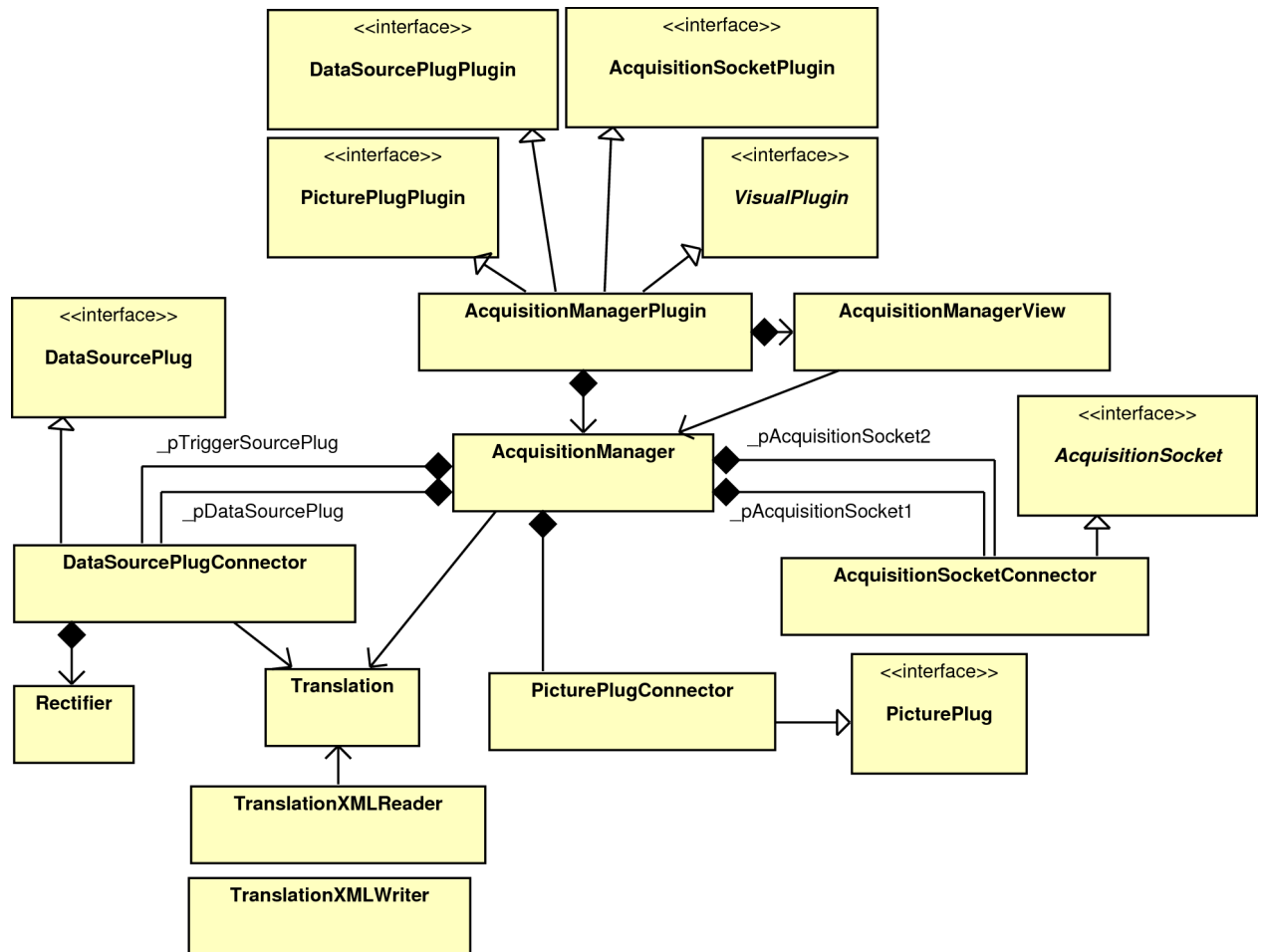


Abbildung II.5.11: Klassendiagramm des Moduls *AcquisitionManagerPlugin*

Das Lesen und Schreiben von Verschiebeprofilen übernehmen die Klassen *TranslationXMLReader* und *TranslationXMLWriter*. Wie der Name impliziert, werden die Wegpunkte in XML-Notation gespeichert.

Die Klasse *DataSourcePlugConnector* implementiert die Schnittstelle *DataSourcePlug* und bildet einen Signal-End-Punkt, der mit dem IO-Modul verbunden werden kann. Die zwei Instanzen dieser Klasse liefern einerseits das Verschiebesignal für den Piezotranslator und andererseits das Triggersignal für die Kamera und beziehen diese Daten wiederum von der Klasse *Translation*. Hier erfolgt auch die Eintragung der Bild-Positionen in einen Container. Erhaltene Bilder werden im Weiteren mit diesen Eintragungen verknüpft.

Das Verschiebesignal durchläuft vor der Übermittlung zum IO-Modul eine Korrektur durch die Klasse *Rectifier*. Diese Korrektur könnte etwa eine bekannte Hysterese des Piezotranslators ausgleichen. Eine Korrektur ist nicht implementiert.

Die Ankunft eines Bildes wird über die Schnittstelle *PicturePlug*, welche durch die Klasse *PicturePlugConnector* implementiert ist, signalisiert und an *AcquisitionManager* weitergereicht. In *AcquisitionManager* wird von *DataSourcePlugConnector* die entsprechende Position erfragt und beide Informationen über die Schnittstelle *AcquisitionSocket* an ein nachfolgendes Modul gereicht. Es sind zwei Signal-End-Punkte zur Weitergabe der kombinierten Messdaten implementiert.

II.5.10 Messdatenspeicherungs-Modul

Dieses Modul implementiert das Problembereichs-Modul Messdatenspeicherung aus Kapitel II.4.3.1. Hier werden Messdaten bestehend aus Bildern und Positionen gespeichert. Die Speicherung jedes Einzelbildes als Datei ist aufgrund des hohen Verwaltungsaufwandes im Dateisystem nicht ratsam⁸. Eine Speicherung der Bilddaten in einen Videocontainer erscheint als die effizienteste Lösung. Bilder werden durch die Klasse `PictureToStream` mit Hilfe der FFmpeg-Bibliothek (siehe Anhang C.5) in einen Videocontainer gespeichert. Die Positionsdaten speichert die Klasse `PositionToDisk` als XML-Datei. Die Klasse `StreamSinkThread` wartet in einem Ausführungspfad auf eine Benachrichtigung und bezieht nach dessen Eintreffen die entsprechenden Messdaten asynchron über die zur Schnittstelle `AcquisitionPlug` passende Gegenseite `AcquisitionSocket`. Die Messdaten werden anschließend an die Klassen `PictureToStream` und `PositionToDisk` zur Speicherung weitergereicht. Über die GUI-Klasse `DataRecordingView` können Parameter wie etwa Dateiname und Videocodec verändert werden. Die Klasse `DataRecordingPlugin` bildet die Modulschnittstelle und ermöglicht über die implementierten Schnittstellen `VisualPlugin`, `ThreadedPlugin` und `AcquisitionPlugPlugin` die Anzeige der GUI, das Starten und Stoppen des Ausführungspfad und das Beziehen von Messdaten. In Abbildung II.5.12 ist das Klassendiagramm dieses Moduls zu sehen.

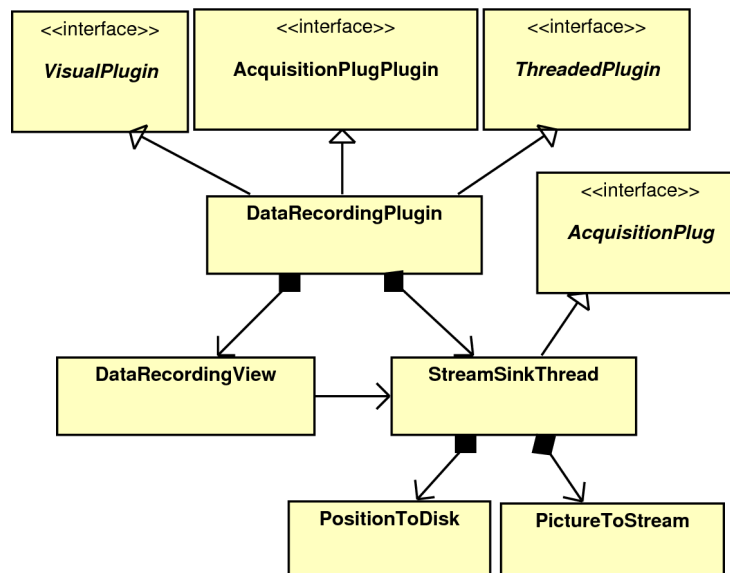


Abbildung II.5.12: Klassendiagramm des Moduls `DataRecordingPlugin`

⁸ Benchmarks verschiedener Dateisysteme sind in <http://linuxgazette.net/122/piszc.html> zu sehen.

II.5.11 Topografieberechnungs-Modul

Dieses Modul implementiert das Problembereichs-Modul Topografieberechnung aus Kapitel II.4.3.1. Die potentiell aufwendigen Berechnungen werden in einem eigenen Ausführungspfad durchgeführt. Die Klasse `CenterOfMassTopographyCalculationPlugin` bildet die Modulschnittstelle und implementiert die Schnittstellen `TopographySocketPlugin`, `ThreadedPlugin` und `AcquisitionPlugPlugin`, um die Weiterleitung topografischer Daten, das Starten und Stoppen sowie den Erhalt von Daten zu ermöglichen. Die Klasse `TopographyCalculationThread` wartet auf eine Benachrichtigung über die Schnittstelle `AcquisitionPlug` und bezieht daraufhin Messdaten, die an die Klasse `CenterOfMass` weitergeleitet werden. Ist eine Topografieberechnung beendet, wird dessen Ergebnis über `TopographySocketConnector` an ein verbundenes Modul übergeben. Weitere Auswertemethoden können analog zu dieser Implementierung realisiert werden.

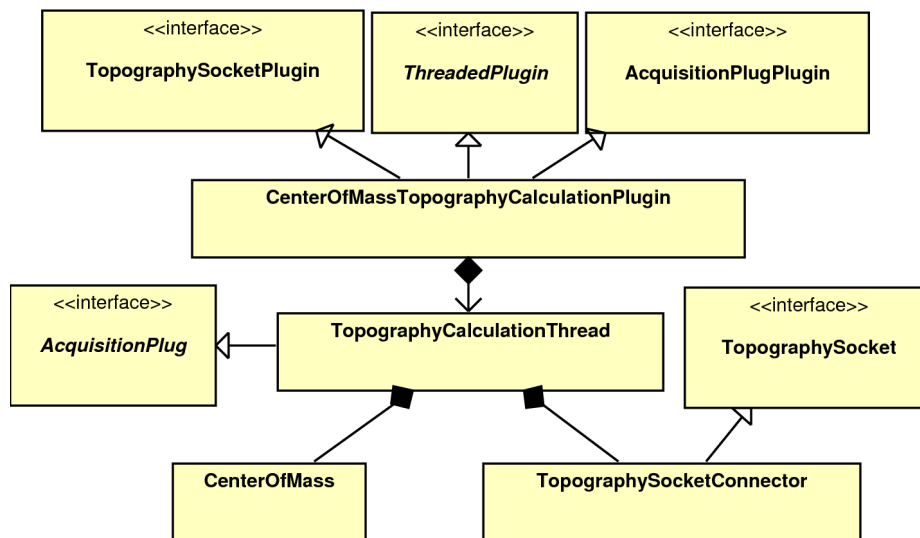


Abbildung II.5.13: Klassendiagramm eines Topografieberechnungs-Moduls

II.5.12 Topografiespeicherungs-Modul

Dieses Modul implementiert das Problembereichs-Modul Topografiespeicherung aus Kapitel II.4.3.1. Die Klasse `MatrixTopographyRecordingPlugin` bildet die Modulschnittstelle und implementiert die Schnittstellen `TopographyPlugPlugin` und `ThreadedPlugin`. Nach einer Nachricht durch die Schnittstelle `TopographyPlug` über eine neue zu speichernde Topografie werden mit `TopographyRecordingThread` asynchron die topografischen Daten bezogen und an `TopographyToMatrix` übergeben. Diese Klasse speichert die Topografie als in Matlab importierfähige Matrix.

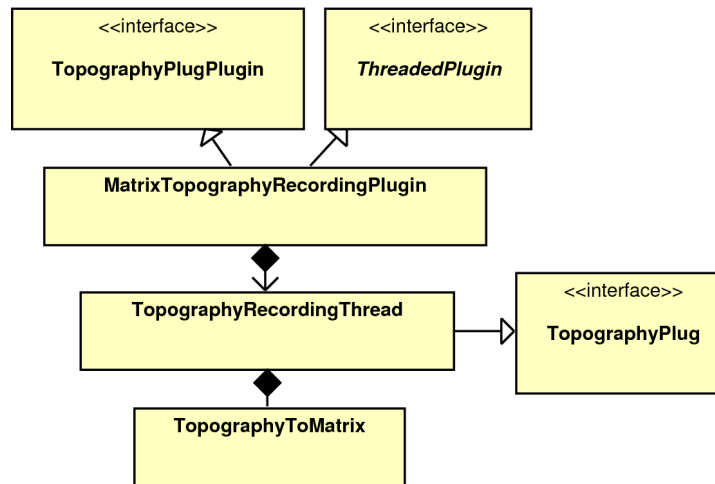


Abbildung II.5.14: Klassendiagramm eines Topografiespeicherungs-Moduls

II.6 Zusammenfassung

In diesem Teil der vorliegenden Arbeit werden die Entstehung und die Implementierung eines Programms beschrieben, das den Messtechniker bei der Bedienung des Messgeräts unterstützt und Messdaten aufnimmt und auswertet.

Zur Erstellung des Programms werden zunächst die Anwendungsfälle Vorbereitung der Messung, Messung, Wiederholung der Messung und Wartung identifiziert. Deren Analyse mündet in Klassendefinitionen, mit denen sie im Problembereich durchgeführt werden können.

Anforderungen, die sich aus den Anwendungsfällen und aus anderen Quellen ergeben, fließen in die Richtungsentscheidungen

- der Implementierung eines Modulsystems und
- der asynchronen Verarbeitung von Daten in eigenen Ausführungspfaden (Threads)

ein. Im Entwurfskonzept wird des weiteren die Modularisierung ausgearbeitet. Das Modulkonzept baut auf den oben definierten Klassen auf. Diese werden nach verschiedenen Kriterien in die folgenden Module abgebildet:

- *Vorbereitung*: Einrichten des Messobjekts, sodass die Messung optimal erfolgen kann.
- *Messung*: Die eigentliche Messung.
- *Wiedergabe*: Die Wiedergabe von Messdaten zur erneuten Auswertung.
- *Verwaltung*: Konfiguration des Programms.
- *Messdatenspeicherung*: Speicherung von Messdaten.
- *Datenerfassung*: Fusion von Daten aus mehreren Quellen.
- *Bewegung*: Durchführung der Positionierung.
- *Kamera*: Empfang und Bereitstellung von aufgenommenen Bildern.
- *Topografieberechnung*: Berechnung der Topografie des Messobjekts.
- *Topografiespeicherung*: Speichern der Topografie.

Mit diesen Modulen werden Modul-Anwendungsfälle erzeugt, und damit Anforderungen an das Modulsystem gefunden.

Die Umsetzung des Modulkonzepts erfolgt durch zur Laufzeit ladbare Bibliotheken. Durch die Verbindung von Modulen können diese gemeinsam Anwendungsfälle durchführen. Die Verbindung wird durch generische Schnittstellen ermöglicht. Jede Modul-Schnittstelle kann mehrere Verbindungspunkte gleichen Typs enthalten.

Ein Modul kann durch Mehrfachvererbung mehrere Modul-Schnittstellen implementieren. Eine gemeinsame Basisklasse der Modul-Schnittstellen ermöglicht das Auffinden von Verbindungspunkten in allen implementierten Schnittstellen. Die Speicherung und Wiederherstellung von Verbindungen ist durch eine teilautomatische eindeutige Benennung der Verbindungspunkte möglich.

Die Modularisierung ermöglicht die Erweiterung, oder Veränderung von Funktionalität, ohne Seiteneffekte befürchten zu müssen.

Neben dem Modulkonzept, ist das Parameterkonzept ein wichtiges Standbein des Programms. Parameter werden mit Zusatzinformationen angereichert, sodass der Mess- oder Wartungstechniker nur gültige und in ihrer Wirkung eindeutig beschriebene Einstellmöglichkeiten vorfindet. Die Zusatzinformationen bestehen aus Wertebereichen, Schrittweiten und Auswahlmöglichkeiten, sowie einer Bezeichnung, einer Beschreibung, einem Kommentar des Entwicklers und einer physikalischen Einheit. Parameter werden zentral gespeichert, geladen und verändert. Jedes Modul implementiert eine Schnittstelle zum Austausch von Parametern.

Zusammenfassung

Mit dieser Masterarbeit wird ein Programm zur messtechnischen Erfassung von Weißlichtinterferogrammen erstellt.

Im theoretischen Teil der vorliegenden Arbeit wird beschrieben, wie Weißlichtinterferogramme gemessen und verarbeitet werden können, und welche Einflüsse sich auf die Messunsicherheit der berechneten Topografie auswirken.

Weißlicht-Interferometer werden verwendet, um die Topografie von mikroskopischen Oberflächen zu vermessen. Ihr Aufbau ähnelt dem eines Auflicht-Mikroskops. Das Objektiv dient jedoch nicht nur zur Vergrößerung, es erzeugt auch einen Interferenzbereich. Durch diesen Bereich wird das Messobjekt entlang der optischen Achse des Objektivs bewegt, sodass sich Interferenzstreifen über die Oberfläche bewegen. Während der Bewegung werden diese Streifenmuster von einer Digitalkamera aufgenommen. An jedem Bildpunkt der Kamera entsteht so ein Messsignal, das ein Weißlichtinterferogramm enthält. Aus den Positionen dieser Korrelogramme im Messsignal kann die Topografie durch verschiedene Auswertemethoden ermittelt werden.

Viele Faktoren nehmen Einfluss auf die Messunsicherheit der ermittelten Topografie. Zum einen kann das Messobjekt selbst durch die Beschaffenheit und die Form der Oberfläche sowie dem Material die Messunsicherheit vergrößern. Auch das Messgerät kann durch ungenaue Positionierung des Messobjekts, Abbildungsfehler der Optik, Fluktuationen der Beleuchtung und Nichtlinearitäten und Rauschen der Kamera auf die Messunsicherheit Einfluss nehmen. Nicht zuletzt können Vibrationen und Temperaturschwankungen von außen das Messergebnis beeinflussen.

Auch der Messtechniker kann durch die korrekte Einrichtung des Messobjekts, die passende Wahl der Auswertemethode und die richtige Konfiguration der Messeinrichtung maßgeblich Einfluss auf die Messunsicherheit nehmen.

Das im praktischen Teil der vorliegenden Arbeit erstellte Programm soll den Messtechniker bei der Einrichtung des Messobjekts so unterstützen, dass die Messunsicherheit möglichst klein gehalten wird. Die weiteren Aufgaben des Programms umfassen die Aufzeichnung der Messdaten und der zugehörigen Einstellungen ebenso wie die Wiedergabe von Messdaten zur erneuten Auswertung. Dabei soll das Programm einfach um Auswertemethoden und andere Funktionalitäten erweitert werden können.

Zur Umsetzung des Programms werden Anwendungen des Programms in Anwendungsfällen dargestellt und analysiert und Anforderungen erhoben.

In einem Entwurfskonzept werden Grundsatzentscheidungen getroffen und begründet. Dieses Konzept sieht ein Modulsystem vor, in welchem funktionale Einheiten in Module gekapselt werden. Diese Module können zur Laufzeit geladen und verbunden werden, um gemeinsam eine Aufgabe zu erfüllen. Durch diese starke Trennung von Funktionalitäten können Erweiterungen und Veränderungen ohne Seiteneffekte durchgeführt werden. Dieses Konzept wird mit objektorientierten Methoden in einen realisierbaren Entwurf transformiert und dieser implementiert.

Ausblick

Die bestehende Messeinrichtung bietet einige Möglichkeiten zur Veränderung bzw. Erweiterung. In Kapitel I.7 werden einige Verbesserungsmöglichkeiten aufgezeigt.

In naher Zukunft wird das Messgerät um eine Möglichkeit der Positionsmessung mittels eines Michelson-Interferometers erweitert werden. Dieses Interferometer soll den tatsächlich zurückgelegten Weg bei der Verschiebung des Messobjekts messen und über Digitaleingänge der Datenerfassungs-Steckkarte oder den Druckeranschluss an das Programm melden, wenn ein Bild aufgenommen wurde.

Durch diese Rückmeldung wird zwar die Positionierung des Messobjekts nicht genauer, es wird jedoch möglich den Fehler des Abtastintervalls zu quantifizieren. Mit diesem Fehler kann die Unsicherheit der berechneten Topografie bestimmt, oder eventuell die Messdaten entsprechend korrigiert werden. Durch Erweiterung des Programms um ein abgeändertes Datenerfassungsmanagement-Modul kann diese Rückführung in das Programm integriert werden.

Auch eine gänzlich andere Art der Datenerfassung und Auswertung wird durch die Positionsmessung möglich. Bilder können durch eine frei laufende Kamera aufgenommen werden. Die Kamera signalisiert die Aufnahme dem Positionsmessgerät, das die aktuelle Position ermittelt und an das Programm übergibt. Damit kann die Messdatenaufnahme erheblich einfacher erfolgen, da die Synchronisation der Verschiebung des Messobjekts mit der Bildaufnahme entfällt.

Die potentiell ungleichmäßige Abtastung des Korrelogramms führt bei der Mehrzahl der in dieser Arbeit vorgestellten Auswertemethoden zu erheblichen Fehlern. Einzig die Schwerpunkt-Methode (siehe Kapitel I.5.1) kann diese Abtastung tolerieren. Eine Auswertung im Frequenzbereich durch die Least-squares spectral analysis (LSSA) ([46], Kapitel 13.8) scheint analog zur Frequency-Domain-Analysis-Methode ebenfalls möglich. Eine weitere Möglichkeit besteht in der Interpolation der Messwerte. Damit könnte eine äquidistante Abtastung simuliert werden.

Am Programm selbst können diverse Erweiterungen die Messung verbessern und die Bedienung vereinfachen. Durch ein Kalibrierungs-Modul könnten Korrekturkurven gemessen werden. Die entsprechende Korrektur der Messwerte beziehungsweise der Positionierung ist ebenfalls nicht implementiert. Ein Modul zur grafischen Erstellung eines Verschiebeprofils und eine Verbesserung des Verwaltungsmoduls könnten die Bedienbarkeit verbessern.

Anhang

A Messeinrichtung

Die Messeinrichtung ist in ([11], S.18) als „Die Gesamtheit aller Meßgeräte und zusätzlicher Einrichtungen zur Erzielung eines Meßergebnisses.“ definiert. In Abbildung A.1 ist die Messeinrichtung zu sehen. Diese besteht aus einem Personal Computer, der mit dem erstellten Programm das Messgerät ansteuert, einem Labor-Verstärker, der die Ausgangsspannung der Datenerfassungs-Steckkarte zur Ansteuerung des PZT verstärkt, einer Lichtquelle, einem Terminal-Block der Datenerfassungs-Steckkarte und dem Messgerät ganz rechts im Bild mit den Bezeichnungen der Komponenten. Eine schematische Beschreibung des Messgeräts ist in Kapitel I.4.2 zu finden. In den folgenden Unterkapiteln werden die relevanten Komponenten der Messeinrichtung beschrieben.

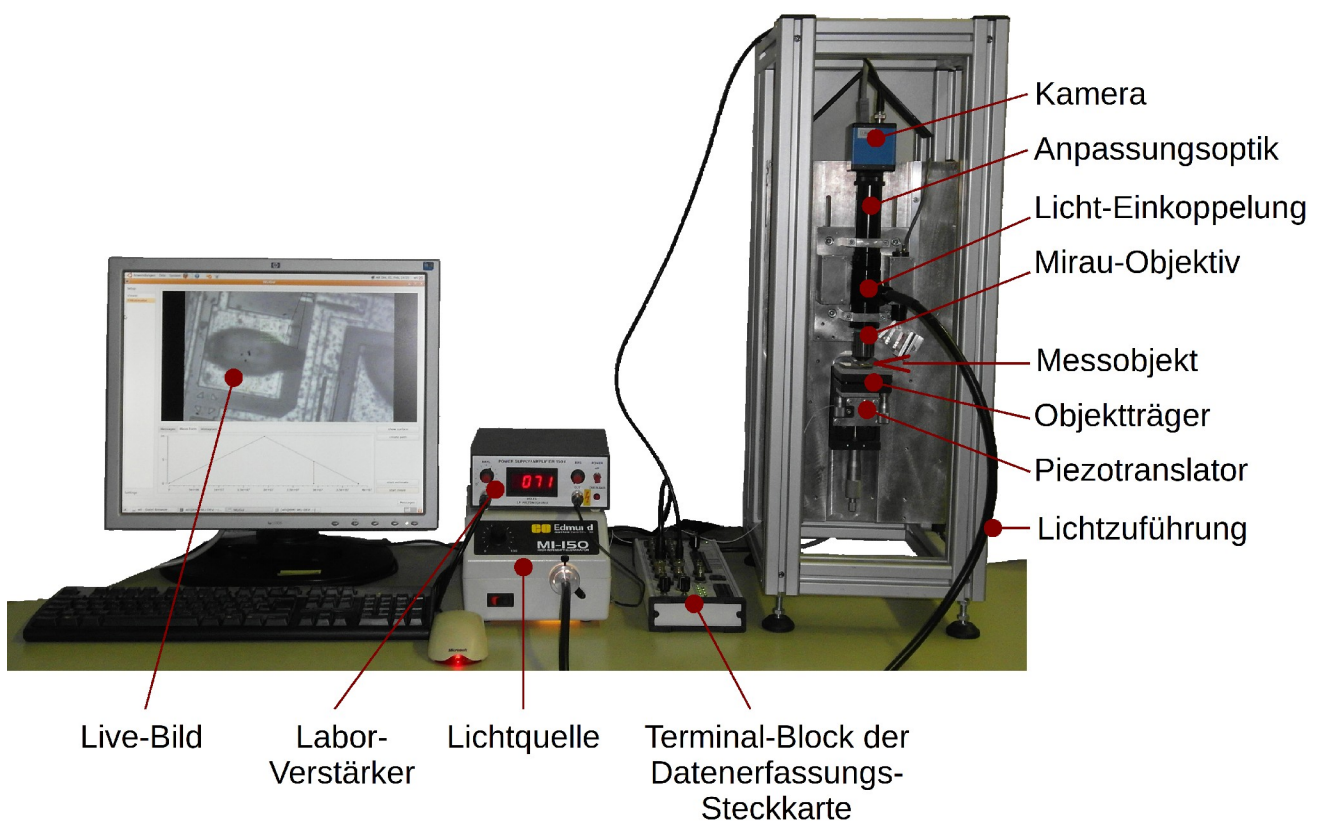


Abbildung A.1: Messeinrichtung von links nach rechts: Bildschirm mit Live-Bild, Labor-Verstärker, Lichtquelle, Terminal-Block, Weißlicht-Interferometer

A.1 PC

Als Basis dient ein handelsüblicher Personal Computer mit einem IEEE 1394 (FireWire) Anschluss. Zur Ansteuerung des Messaufbaus wird eine Datenerfassungs-Steckkarte mit mindestens zwei Analog-Ausgängen und zum Aufnehmen von Bildern eine Kamera mit IEEE 1394 Anschluss benötigt.

A.2 Datenerfassungs-Steckkarte

Das Messgerät wird im Wesentlichen durch die Vorgabe eines Signals zur Bewegung des Piezotranslators und eines Triggersignals zur Auslösung der Belichtung der Kamera angesteuert. Die Ausgabe dieser Signale erfolgt durch eine Steckkarte von National Instruments mit der Bezeichnung NI PCI-6070E. In Verbindung mit dieser Datenerfassungs-Steckkarte wird der Terminal-Block NI BNC-2120 verwendet. Die wichtigsten Eigenschaften dieser Steckkarte sind in Tabelle A.1 aus [54] wiedergegeben.

Kanaltyp	Anzahl	Wertebereich	Auflösung	Abtastrate	relative Genauigkeit	dynamische Nichtlinearität
Analog-Eingänge	16	$\pm 0.05 \text{ V min}$ $\pm 10 \text{ V max}$	12 Bit	1.25 MS/s	$\pm 0,5 \text{ LSB typ}$ $\pm 1,5 \text{ LSB max}$	$\pm 0,5 \text{ LSB typ}$ $\pm 1,0 \text{ LSB max}$
Analog-Ausgänge	2	0-10 V $\pm 10 \text{ V}$	12 Bit	1 MS/s	$\pm 0,3 \text{ LSB typ}$ $\pm 0,5 \text{ LSB max}$	$\pm 0,3 \text{ LSB typ}$ $\pm 1,0 \text{ LSB max}$
IO-Kanäle	8	0 V 5 V	1 Bit	5 kwords/s	-	

Tabelle A.1: Spezifikation NI PCI-6070E

Diese Steckkarte kann mit Hilfe des Comedi-Pakets (siehe Kapitel C.1) betrieben werden.

A.3 Labor-Verstärker

Der Labor-Verstärker kann Eingangsspannungen im Bereich von 0 bis 10 Volt auf bis zu 150 Volt verstärken und mit einem Offset beaufschlagen.

A.4 Piezotranslator

Es wird ein linearer Niedervolt-Aktuator eingesetzt, der einen Stellweg von etwa 100 μm besitzt und etwa 1 μm pro Volt verfährt. Der Translator wird ungerichtet eingesetzt und kann daher nicht genau positioniert werden.

A.5 Mirau-Objektiv

Das verwendete Mirau-Objektiv trägt die Bezeichnung „20X Nikon CF IC Epi Plan DI Interferometry Objective“. Aus dieser Bezeichnung können mit [55] folgende Informationen gewonnen werden:

- „20X“: 20-fache Vergrößerung.
- „IC“ (*infinity-corrected*): Das Objektiv hat einen parallelen Strahlengang und somit ein Bild im Unendlichen.
- „Epi“ (*episcopic*): Reflektierende im Gegensatz zur durchleuchtenden (diaskopischen) Beleuchtung. Die Probe wird durch das Objektiv beleuchtet.

A Messeinrichtung

- „Plan“: Dieses Objektiv korrigiert die Bildfeldkrümmung, sodass auf eine ebene Fläche verzerrungsfrei projiziert wird.
- „CF“ (*chrome-free*): Die chromatische Aberration ist korrigiert.
- „DI“ (*dual beam interferometry*): Bezeichnung eines Mirau-Objektivs.

Laut Datenblatt besitzt das Objektiv die in Tabelle A.2 aufgelisteten Kennwerte:

Parameter	Wert
numerische Apertur	0,4
Brennweite	10 mm
Tiefenschärfe	1,71 μm
Auflösungsvermögen	0,69 μm

Tabelle A.2: Kennwerte des Mirau-Objektivs

A.6 Optik

Zwischen Kamera und Mirau-Objektiv ist eine Optik angebracht, mit der das parallele Strahlenbündel des Objektivs auf den Sensor der Kamera fokussiert wird. Diese Aufgabe übernimmt eine „InfiniTube“ der Firma INFINITY PHOTO-OPTICAL. Ein passender Coaxial Illuminator wird zum Einkoppeln der Beleuchtung verwendet.

A.7 Lichtquelle

Als Lichtquelle dient ein Fiber-Lite Dolan-Jenner MI-150 Beleuchtungsgerät für Lichtleiter. Das verwendete Leuchtmittel ist eine Wolfram-Halogen-Lampe mit der Bezeichnung EKE/X von Osram. Das typische Spektrum dieses Leuchtmittels ist in Abbildung A.2 zu sehen.

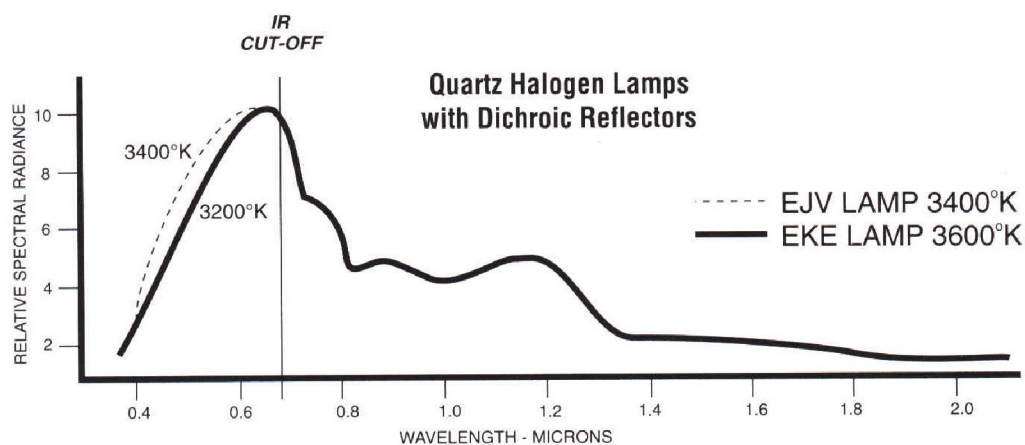


Abbildung A.2: Typische spektrale Leistungsdichte von Quarz Halogen Lampen nach [56]

A.8 Kamera

Die verwendete SW-Kamera trägt die Bezeichnung DMK 41BF02 und wurde von Imaging Source hergestellt. Die Kamera hat eine Empfindlichkeit von 0,05 lx und eine Auflösung von 1280x960 Pixel² mit einer Pixelgröße von 4,65 x 4,65 µm². Es können Schwarz-Weiß-Bilder mit einer Farbtiefe von 8 Bit bei Frameraten von 15, 7,5 oder 3,75 frames per second (fps) aufgenommen werden. Die Kamera besitzt eine IEEE 1394 Schnittstelle.

Der Trigger-Eingang dieser Kamera kann verwendet werden, um Aufnahmen auszulösen. Dabei ist das Timing wie in Abbildung A.3 dargestellt zu beachten. Die Belichtung beginnt 4,8 µs nach der Flanke des Triggersignals. Die verwendete Flanke ist konfigurierbar. Als positiv werden Spannungen von 3.3 bis 12 V bewertet. Die Belichtungszeit ist zwischen 100 µs und 30 Sekunden einstellbar. Die Auslesezeit wird von der eingestellten Framerate bestimmt. Es ist also mit dieser Kamera nicht möglich, mehr als 15 Bilder pro Sekunde zu übertragen.

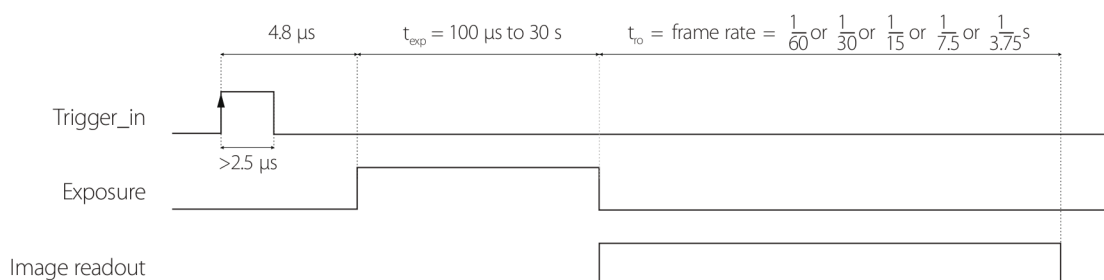


Abbildung A.3: Timing einer Belichtung durch einen Triggerimpuls [57]

In dieser Kamera ist ein CCD-Sensor von Sony mit der Bezeichnung ICX205AL verbaut. Die spektrale Empfindlichkeit des Sensors ist in Abbildung A.4 zu sehen.

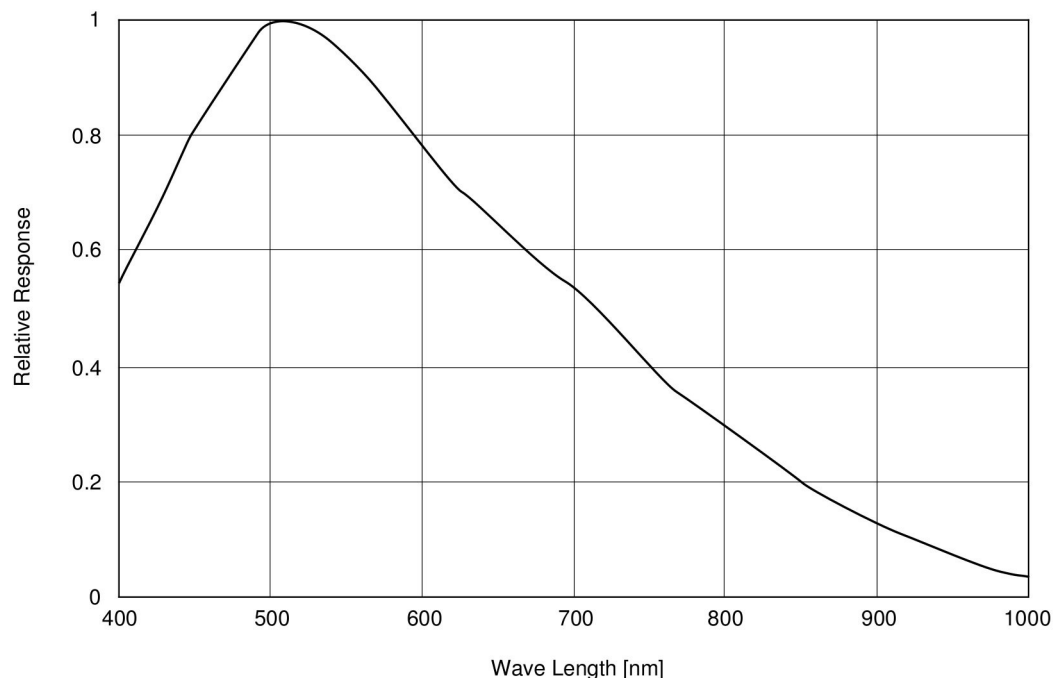


Abbildung A.4: Spektrale Empfindlichkeit des Bildsensors ICX205AL [58]

A.9 Berechnungen

Das Messfeld des Messgeräts ergibt sich durch die Vergrößerung des Objektivs und aus der Größe des Sensors. In Kapitel A.8 wurde die Auflösung des Bildsensors mit 1280x960 Pixel und die Größe eines Pixels mit 4,65x4,65 µm angegeben. Mit der in Kapitel A.5 angegebenen 20-fachen Vergrößerung ergibt sich ein Messfeld von

$$\begin{aligned} \text{Breite} &= \frac{1280 \text{ Pixel} * 4,65 \mu\text{m}}{20} = 297,6 \mu\text{m} \\ \text{Höhe} &= \frac{960 \text{ Pixel} * 4,65 \mu\text{m}}{20} = 223,2 \mu\text{m} \end{aligned} \quad (\text{A.1})$$

am Messobjekt.

Der maximale Neigungswinkel am Messobjekt, bis zu dem das Objektiv reflektiertes Licht aufnehmen kann, ergibt sich durch die numerische Apertur NA des Mikroskopobjektivs. Die numerische Apertur ist definiert als

$$NA = n \sin(\theta) \quad (\text{A.2})$$

wobei mit n der Brechungsindex des Mediums bezeichnet wird. Das verwendete Mirau-Objektiv besitzt eine numerische Apertur von 0,4. Mit n=1 für Luft ergibt sich ein Aperturwinkel von $\theta = 23,6^\circ$. Das Objektiv kann also die Reflexion von geneigten Oberflächen nur bis zu einem Neigungswinkel von $\theta/2 = 11,8^\circ$ erfassen.

Die Wellenlänge des Korrelogramms wird durch die Lage des Schwerpunkts im Spektrum der Lichtquelle bestimmt (siehe Kapitel I.3). Aus Abbildung A.2 kann der Schwerpunkt des Spektrums ermittelt werden. Die Wellenlänge am Schwerpunkt ergibt sich zu etwa 750 nm. Da die OPD der doppelten Verschiebung entspricht, liegt zwischen zwei Interferenzstreifen führt eine Verschiebung von etwa 375 nm.

Die effektive Bandbreite des Spektrums lässt sich ebenfalls aus Abbildung A.2 ermitteln und ergibt sich zu etwa 475 nm. Mit der Abschätzung der Kohärenzlänge aus Kapitel I.3.2.2 ergibt sich eine Kohärenzlänge von etwa 1184 nm.

Die Berechnungsergebnisse werden in der nachfolgenden Tabelle zusammengefasst:

Titel	Wert
Messfeld	297,6 x 223,2 µm ²
max. Neigungswinkel	11,8°
mittlere Wellenlänge	750 nm
effektive Bandbreite	475 nm
Kohärenzlänge	1,2 µm

Tabelle A.3: Eckdaten des Messgeräts

B Algorithmen

Dieses Kapitel befasst sich mit Algorithmen, die zur Implementierung des Programms verwendet werden. Im Vorbereitungs-Modul (Kapitel II.5.4) muss die Verkippung des Messobjekts berechnet werden. Zu diesem Zweck werden aus einer Menge von Punkten die Parameter einer Ebene geschätzt. Stimmt der Normalvektor dieser Ebene mit der optischen Achse des Objektivs überein, ist die Verkippung behoben. Mit Hilfe des RANSAC-Algorithmus (siehe Kapitel Fehler: Referenz nicht gefunden) werden dazu Ausreißer entfernt. Anschließend wird der Normalvektor mittels der Singulärwertzerlegung (siehe Kapitel B.2) geschätzt.

B.1 RANSAC

In diesem Kapitel wird der RANSAC (RANDOM SAMPLE CONSENSUS) -Algorithmus in Anlehnung an ([59], Kap. 4.7.1) beschrieben. Dieser robuste Schätzalgorithmus wird zur Klassifikation einer gemessenen Punktemenge verwendet. Dabei erfolgt eine Einteilung in Punkte, die ein Modell nicht erfüllen (Outlier) und jene die dieses erfüllen (Inlier). Die Menge der Inlier kann zur Schätzung der Modellparameter herangezogen werden.

Die Vorgehensweise, um Modellparameter aus einer mit Ausreißern behafteten Menge S robust zu schätzen ist folgende:

1. Eine Stichprobe von s Punkten aus der Punktemenge S wird gezogen und Modellparameter mit diesen s Punkten berechnet.
2. Klassifiziere die Punkte der Punktemenge S in eine Menge von Inliern S_i deren Abstand vom Modell kleiner als t ist.
3. Wenn der Umfang der Menge S_i größer ist als ein Schwellwert T , berechne die Modellparameter mit der Menge S_i und beende den Algorithmus.
4. Wenn der Umfang der Menge S_i kleiner als der Schwellwert T ist, gehe zu Schritt eins.
5. Nach N Durchläufen wird die größte Menge S_i gewählt und die Modellparameter mit dieser Menge berechnet.

Der Parameter s beschreibt den minimalen Umfang einer Stichprobe. Zur Bestimmung der Parameter einer Geraden werden zum Beispiel zwei Punkte ($s = 2$) benötigt. Eine Ebene kann mit drei Punkten ($s = 3$) aufgespannt werden.

Der Parameter t ist die maximale Distanz eines Inlier-Punktes vom Modell. Punkte deren Abstand größer ist werden als Outlier klassifiziert. Der Wert dieses Parameters soll so gewählt werden, dass ein Punkt mit einer Wahrscheinlichkeit α ein Inlier ist. Ist der Messfehler gaußverteilt mit $N(0, \sigma)$, so ist der orthogonale Punktabstand d^2 eine Summe von gaußverteilten Variablen und folgt der Chi-Quadrat-Verteilung mit m Freiheitsgraden. Die Anzahl der Freiheitsgrade ist gleich der Kodimension des Modells. Im Falle einer zweidimensionalen Ebene im dreidimensionalen Raum ergibt sich die Kodimension zu eins.

B Algorithmen

Die kumulierte Verteilungsfunktion der Chi-Quadrat-Verteilung ist gegeben durch:

$$F_m(k^2) = \int_0^{k^2} X_m^2(\xi) d\xi \quad (\text{B.1})$$

Damit wird maximale Distanz t mit

$$\begin{cases} \text{inlier} & d^2 < t^2 \\ \text{outlier} & d^2 \geq t^2 \end{cases} \text{ mit } t^2 = F_m^{-1}(\alpha)\sigma^2 \quad (\text{B.2})$$

gewählt. Die inverse kumulative Chi-Quadrat-Verteilung F_m^{-1} ist proportional zur inversen unvollständigen Gammafunktion:

$$F_m^{-1}(\alpha) = 2\Gamma_{m/2}^{-1}(\alpha) \quad (\text{B.3})$$

Häufig wird α so gewählt, dass 95 % der Punkte Inlier sind ($\alpha = 0.95$).

Die Anzahl der Iterationen N soll so gewählt werden, dass mit einer Wahrscheinlichkeit p zumindest eine Stichprobe ohne Ausreißer gezogen wird. Bezeichnen ε die Wahrscheinlichkeit, dass ein Punkt ein Ausreißer ist, und s den minimalen Umfang der Stichprobe, so werden zumindest

$$N = \log(1-p) / \log(1-(1-\varepsilon)^s) \quad (\text{B.4})$$

Stichproben benötigt. Als Wahrscheinlichkeit p wird häufig 99 % ($p = 0,99$) verwendet.

Der Schwellwert T bezieht sich auf den Umfang der Menge der Inlier S_i . Gewöhnlich wird der Algorithmus abgebrochen, wenn die Menge S_i die geschätzte Menge an Inliern erreicht hat. Wird mit einem Ausreißeranteil von $\varepsilon = 0,2$ gerechnet, ergibt sich T zu

$$T = (1-\varepsilon)n \quad (\text{B.5})$$

wenn die Anzahl der Punkte mit n bezeichnet wird. Ist der Anteil der Ausreißer nicht bekannt, so kann zu Beginn der Anteil mit $\varepsilon = 0,5$ zu 50 % angenommen werden. Dieser Anteil ist der schlechteste Fall, der durch robuste Algorithmen behandelbar ist. Wird bei einer Iteration ein geringer Ausreißer-Anteil gefunden, kann ε entsprechend angepasst werden.

Im Zuge dieser Anpassung kann auch der Distanz-Schwellwert t angepasst werden, indem die Streuung der Messwerte σ aus der Menge der Inlier S_i neu berechnet wird.

B.2 Singulärwertzerlegung

Mit der SVD (singular value decomposition) werden die Parameter eines Modells so geschätzt, dass der quadratische Fehler der Schätzung minimal wird. Im implementierten Programm wird dieser numerisch sehr stabile Algorithmus verwendet, um die Parameter einer Ebene zu bestimmen. Als Datenbasis dieser Berechnung dient eine Menge von Messpunkten, die keine Ausreißer enthält. Im dreidimensionalen Raum bestehen die Messpunkte aus den Komponenten x_1 , x_2 und x_3 , welche die die Ebenengleichung

$$ax_1 + bx_2 + cx_3 = 1 \quad (\text{B.6})$$

erfüllen sollen. Mit den Messwerten und der Ebenengleichung kann ein überbestimmtes Gleichungssystem gebildet werden, das durch eine Matrix A mit m Zeilen und $n = 3$ Spalten, einem Spaltenvektor x bestehend aus den Komponenten x_1 , x_2 und x_3 sowie dem Ergebnisvektor b ausgedrückt werden kann:

$$A\vec{x} = \vec{b} \quad (\text{B.7})$$

Die Parameter a , b und c können durch die SVD geschätzt werden.

Nach ([60], Kapitel 6.7) erzeugt SVD eine Faktorisierung von A in

$$A = U \Sigma V^{-1} = U \Sigma V^T \quad (\text{B.8})$$

mit den Singulärwerten $\sigma_1 \geq \dots \sigma_r > 0$ in der Diagonalmatrix Σ und den orthogonalen Basen U und V. Die Zahlen $\sigma_1^2 \geq \dots \sigma_r^2 > 0$ sind die von Null verschiedenen Eigenwerte von AA^T und $A^T A$. Die Spalten von U und V sind die Eigenvektoren von AA^T und $A^T A$. Sie bilden orthogonale Basen der zu A gehörigen fundamentalen Unterräume: dem Kern von A und A^T und dem Spaltenraum von A und A^T . Die Spalten von U und V werden folgendermaßen den Unterräumen zugeordnet:

- Erste r Spalten von V: Zeilenraum von A
- Letzte $n - r$ Spalten von V: Kern von A
- Erste r Spalten von U: Spaltenraum von A
- Letzte $m - r$ Spalten von U: Kern von A^T

Im Spaltenraum einer Matrix A befindet sich der Vektor b, falls diese eine Lösung existiert. Der Kern von A^T ist Normal zum Spaltenraum von A. Existiert die Lösung von $A \vec{x} = \vec{b}$ nicht, so kann der Vektor b auf den Spaltenraum von A projiziert werden. Der Fehlervektor $\vec{e} = \vec{b} - A \hat{x}$, zwischen der Projektion und dem Vektor b ist orthogonal zum Spaltenraum von A und liegt daher im Kern von A^T . Die Lösung \hat{x} ist jene mit dem kleinsten Fehlerquadrat $E = \|A \vec{x} - \vec{b}\|^2$. Die Berechnung von x aus (B.5) erfolgt durch Bildung der Pseudoinversen A^+ :

$$A^+ = V \Sigma^+ U^T \quad (\text{B.9})$$

Die inverse Tabelle der Singulärwerte Σ^+ wird durch Invertierung der Singulärwerte mit $[\text{diag}(1/\sigma_i)]$ gebildet. Damit ergibt sich die Lösung von (B.5) mit den kleinsten Fehlerquadraten zu

$$x = A^+ \vec{b} = V \Sigma^+ U^T \vec{b} = V [\text{diag}(1/\sigma_i)] U^T \vec{b} \quad (\text{B.10})$$

Der Vektor x ist der Normalvektor der gesuchten Ebene. Die Verkippung verschwindet für $x = [0, 0, k]^T$, da die optische Achse als z-Achse gewählt wurde.

Die Berechnung von x kann entfallen, wenn die gesuchte Ebene durch den Ursprung verläuft. In diesem Fall ist die zu lösende Gleichung durch

$$A \vec{x} = \vec{0} \quad (\text{B.11})$$

gegeben. Da die Matrix A eine Ebene repräsentiert, ergibt sich der Rang r der Matrix A zu zwei. Nach Ausführen der SVD ergeben die ersten zwei Spalten von U die Basisvektoren der gesuchten Ebene. Der dritte Vektor von U ist der Kern von A^T . Dieser ist orthogonal zum Spaltenraum von A. Der dritte Vektor von U ist daher der gesuchte Normalvektor der Ebene.

Rechenoperationen können somit durch Verschieben der gesuchten Ebene in den Ursprung eingespart werden, da der Lösungsvektor x nicht berechnet werden muss.

C Bibliotheken

Durch den Einsatz von Bibliotheken wird die Implementierung verschiedener Anforderungen erleichtert. XML-Dateien können beispielsweise durch die Bibliothek libxml2 eingelesen werden. Das Zeichnen von Graphen kann durch Qwt erfolgen, die Ansteuerung der Datenerfassungs-Steckkarte erfolgt mit Comedi und mit Unicap wird die Kamera konfiguriert.

Die folgende Tabelle listet die Bibliotheken, deren Bezugsquellen und die verwendeten Versionen auf:

Titel	Version	Quelle
Comedi	CVS-Version vom 16.5.2010	www.comedi.org
Unicap	0.9.5	http://unicap-imaging.org/
Qt	04.05.01	http://qt.nokia.com/
Qwt	05.02.11	http://qwt.sourceforge.net/
Ffmpeg	0.5	http://www.ffmpeg.org/
libxml2	02.07.03	http://xmlsoft.org/
boost	01.01.39	www.boost.org/
boost.log	RC 3	http://sourceforge.net/projects/boost-log/

Tabelle C.1: Versionen und Bezugsquellen von verwendeten Bibliotheken

C.1 Comedi

Das Comedi-Projekt entwickelt Open-Source Treiber, Werkzeuge und Bibliotheken zur Datenerfassung. Das Projekt ist in drei Teilen organisiert:

- *Comedi* ist eine Treiber-Sammlung für verschiedene Datenerfassungsgeräte. Die Treiber sind als Module im Linux-Kernel implementiert.
- *Comedilib* ist eine User-Space-Bibliothek, die eine einheitliche, einfach anwendbare Programmier-Schnittstelle zu den Comedi-Treibern im Kernel-Space zur Verfügung stellt. Ebenfalls enthalten sind Dokumentation, Beispielprogramme und Werkzeuge zur Kalibration und Konfiguration.
- *Komedilib* ist ein Betriebssystem-Kern-Modul, welches die gleiche Schnittstelle zu den Treibern wie Comedilib bietet, jedoch zur Verwendung im Betriebssystem-Kern dient. Hiermit können Echtzeit-Anwendungen im Betriebssystem-Kern entwickelt werden.

Die Dokumentation ist auf der Projektseite unter <http://www.comedi.org/documentation.html> zu finden.

Comedi organisiert unterstützte Geräte wie folgt:

- *Channel*: Ein Kanal ist die kleinste Organisations-Einheit. Er entspricht einem physischen Anschluss, etwa einem Analog-Ausgang. Jeder Kanal hat Parameter wie Spannungsbereich, Referenzspannung, Polarität, Umrechnungsfaktoren von Wert zu Spannung und umgekehrt.
- *SubDevice*: Dies ist eine Ansammlung von Kanälen mit gleicher Funktionalität, die physikalisch auf der gleichen Datenerfassungskarte implementiert sind. Diese SubDevices werden meist von der Hardware vorgegeben, wenn beispielsweise der Spannungsbereich der ADU-Kanäle nur für den gesamten SubDevice eingestellt werden kann.

- *Device*: Ein Device ist ein Gerät, auf dem mehrere SubDevices gemeinsam implementiert sind. Dies entspricht einer Datenerfassungskarte.

Die Comedi-Treiber erzeugen im Verzeichnis /dev Einträge für jedes Device und deren SubDevices:

```
/dev/comedi0
/dev/comedi0_subd0
/dev/comedi0_subd1
```

Die Gerätedatei comedi0 repräsentiert die gesamte Datenerfassungskarte. Alle SubDevices können hier angesprochen werden. Durch comedi_subd0 kann nur der erste SubDevice bedient werden. Jede Gerätedatei entspricht einer Treiber-Instanz. Jede Treiber-Instanz im Verzeichnis /dev kann nur jeweils ein SubDevice asynchron mit Daten versorgen.

Mit Comedi und Comedilib werden auf den Analog-Ausgängen der Datenerfassungskarte Signale erzeugt, die den Piezotranslator bewegen und die Kamera triggern. Im User-Mode muss zu diesem Zweck ein Puffer gefüllt werden. Dieser wird von Comedi asynchron ausgegeben. Es ist nicht möglich, aufgrund eines Signals an einem Eingang sofort das Signal am Ausgang zu verändern. Es muss erst die Ausgabe beendet, der Puffer neu befüllt und anschließend die Ausgabe neu gestartet werden. Eine Regelschleife ist daher nicht implementierbar.

Als Alternative existiert ein Treiber von National Instruments für Linux mit der Bezeichnung „NI-DAQmx for Linux“. Die Verwendung dieses Treibers bedingt allerdings den Einsatz von Datenerfassungs-Steckkarten dieses Herstellers, wohingegen von Comedi auch Steckkarten anderer Hersteller bedient werden können.

C.2 Unicap

Unicap ist eine Bibliothek, welche die Verwendung von Kameras mit IEEE 1394 oder USB Schnittstelle vereinfacht. Die Dokumentation ist auf der Projektseite unter <http://unicap-imaging.org/documentation.htm> zu finden.

C.3 Qt

Diese Bibliothek erlaubt die plattformübergreifende Entwicklung von Benutzeroberflächen in C++. Qt wird von Nokia entwickelt und unter verschiedenen freien und kommerziellen Lizenzen angeboten. Qt ist sehr gut dokumentiert und bietet eine konsistente Programmierumgebung. Diese Bibliothek erweitert C++, indem Metainformationen zu Qt/C++-Klassen generiert werden. Diese Metainformationen ermöglichen asynchrone Funktionsaufrufe, Introspektion sowie ein Signal/Slot-Konzept. Nur einige Teile der umfangreichen Qt-Bibliothek werden verwendet:

- QtCore: Kern-Klassen, die von allen anderen Modulen genutzt werden.
- QtOpenGL: Unterstützung von OpenGL.
- QtGui: Komponenten zur Gestaltung von grafischen Benutzeroberflächen.

Die Dokumentation ist unter <http://doc.qt.nokia.com/4.5/index.html> zu finden.

C.4 Qwt

Die Bibliothek Qwt bietet vielfältige Möglichkeiten, Diagramme darzustellen, sowie diverse Komponenten zur Erstellung von Benutzeroberflächen wie zum Beispiel Schalter und Anzeigen in diversen Ausführungen. Qwt wurde mit der Qt-Bibliothek implementiert. Die Umsetzung der Programmierschnittstelle orientiert sich an Qt-Richtlinien und die Integration in eine Qt-Applikation ist sehr einfach. Die Dokumentation ist auf der Projektseite unter <http://qwt.sourceforge.net/> zu finden.

C.5 FFMpeg

Das FFMpeg-Projekt definiert sich als eine plattformübergreifende Lösung zum Aufnehmen, Konvertieren und kontinuierlichen Übertragen von Audio und Video.

Zu diesem Zweck werden verschiedene Bibliotheken angeboten, von denen folgende verwendet werden:

- *libavcodec*: Hier sind alle Encoder und Decoder für Video und Audio versammelt. Darunter befinden sich auch Encoder für verlustfreie Kompression.
- *libavformat*: Mit Hilfe dieser Bibliothek werden verschiedene Containerformate gelesen und geschrieben.
- *libswscale*: Diese Bibliothek erlaubt die Konvertierung zwischen diversen Bild- und Pixelformaten. Die verwendete Kamera liefert beispielsweise Bilder im Format Y800, die nach YUV9/YV12 oder RGB24 konvertiert werden können. In diesem Zusammenhang sei auf die Webseite <http://www.fourcc.org/> hingewiesen, auf der verschiedenste Videobildformate beschrieben sind.

Die Programmierschnittstelle von FFMpeg ist spärlich dokumentiert. Schwierigkeiten können diverse Einschränkungen bereiten. So kann in der verwendeten Version 0.5 kein Bitmap mit 8 Bit Farbtiefe erstellt werden. Es können auch nicht alle Containerformate alle Videobildformate aufnehmen.

C.6 libxml

Diese Bibliothek wurde für das Gnome-Projekt entwickelt und bietet mehrere Möglichkeiten, XML-Dokumente zu parsen. XML ist ein Standard, um Informationen in maschinenlesbarer Form zu codieren. Einstellungen und Messdaten werden in XML-Notation gespeichert. Ein kurzes Tutorium über die Verwendung von libxml ist unter <http://www.jamesh.id.au/articles/libxml-sax/libxml-sax.html> zu finden.

C.7 boost

Die C++ Template-Bibliothek-Sammlung boost bietet vielfältige Funktionalitäten von grundlegenden Bibliotheken wie *smart_ptr* bis zur Abstraktion von Dateisystemen mit *FileSystem*. Aber auch diverse Datenstrukturen bis hin zur Unterstützung von Template Metaprogrammierung werden angeboten.

Von boost werden unter anderem folgende Bibliotheken vorwiegend genutzt:

- *smart_ptr*: Diese Bibliothek bietet verschiedene Kapselungen von Zeigern, sodass allozierter Speicher nach verschiedenen Kriterien automatisch freigegeben wird.
- *Statechart*: Diese Bibliothek ermöglicht das Erzeugen von endlichen Automaten, die zusammen mit libxml zum Parsen von XML-Dokumenten verwendet werden.
- *Variant*: Dies ist ein Container, in dem verschiedene Typen gespeichert werden können.

- *Thread*: Dies ist eine einfache Bibliothek zur Verwendung von Threads.
- *Function*: Diese Bibliothek kapselt Funktions- oder Methodenpointer in Funktionsobjekte, etwa zur Definition von Rückruffunktionen.
- *Bind*: Mit dieser Bibliothek können Funktionsobjekte gebildet werden, in denen ein oder mehrere Parameter der enthaltenen Funktion festgelegt werden. Es kann auch eine Methode an ein Objekt gebunden werden, sodass das dieses Funktionsobjekt als Rückruffunktion eingesetzt werden kann.
- *Lexical Cast*: Dies erlaubt das Umwandeln von Zeichenketten in Basisdatentypen und umgekehrt.
- *Date Time*: Mit Hilfe dieser Bibliothek können Zeiten und Zeitspannen definiert und manipuliert werden.

Die Dokumentation von boost befindet sich auf der Projektseite unter <http://www.boost.org>

C.8 boost.log

Diese Bibliothek ermöglicht umfangreiches Logging mit diversen Optionen in Bezug auf Formatierung, Filterung und Speicherung von Log-Meldungen. boost.log ist ein Anwarter auf die Aufnahme in die boost-Bibliothek. Die Dokumentation befindet sich auf der Projektseite unter <http://boost-log.sourceforge.net>.

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTÄTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am 14.2.2011

Klausur
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

14.2.2011
date

Klausur
(signature)

Literaturverzeichnis

- [1]: "*Weißlichtinterferometrie zur wirtschaftlichen Rautiefenmessung an Zahnimplantaten*", ZygoLOT GmbH, 2008
- [2]: "*Zerstörungsfreie Qualitätskontrolle an Festplattenbauteilen mit dem TopMap Weißlichtinterferometer*", Polytec GmbH, http://www.polytec.com/ger/158_9332.asp?, 1.2010
- [3]: "*TCO: Surface Roughness and Film Thickness*", Zygo, <http://www.zygo.com/?/met/markets/photovoltaic/newview3dpv-tf.htm>, 1.2010
- [4]: "*3D-Vermessungen an Stoßdämpferkolben*", Polytec GmbH, http://www.polytec.com/ger/_files/LM_AN_TOP-01_2005_11_D.pdf, 1.2010
- [5]: "*Measuring Micro-Lens Radius of Curvature with a White Light Optical Profiler*", Veeco, http://www.veeco.com/pdfs/appnotes/AN509-Measuring_05213lr.pdf, 1.2010
- [6]: "*NewView 7300 Specifications*", Zygo, <http://www.zygo.com/met/profilers/newview7000/nv7300spec.pdf>, 1.2010
- [7]: "*Wyko NT9800 Optical Profiling System*", Veeco, <http://www.veeco.com/pdfs/datasheets/DS542-RevA1-Wyko-NT9800-Optical-Profiling-System-Datasheet-Lo-Res.pdf>, 1.2010
- [8]: "*TMS-1200 TopMap μ .Lab - Tech. Data*", Polytec, http://www.polytec.com/ger/158_9381.asp?, 1.2010
- [9]: "*TMS-100 TopMap Metro.Lab - Tech. Daten*", Polytec, http://www.polytec.com/ger/158_9328.asp?, 1.2010
- [10]: "*FRT MicroSpy Topo DT*", Firma Fries Research & Technology GmbH, http://www.frt-gmbh.com/frt/upload/pdf_de/FRT_MicroSpy_Topo_DT_de.pdf, 1.2010
- [11]: "*DIN 1319-1: Grundlagen der Messtechnik*", Deutsches Institut für Normung, 1995
- [12]: K.D. Möller, "*Optics*", Springer Science+Business Media, 2. Ausgabe, 2007, 978-0-387-26168-3
- [13]: P. Hariharan, "*Optical Interferometry*", Elsevier Science, 2. Ausgabe, 2003, 0-12-311630-9
- [14]: André Henri Mirau, "*Interferometer US2612074*", US Patent 2612074, 1952
- [15]: S. Chen, A. W. Palmer, K. T. V. Gratten, B. T. Meggitt, "*Fringe order identification in optical fibre white-light interferometry using centroid algorithm method*", Electronics Letters, Bd. 28, Nr. 6, S. 553-555, 1992
- [16]: C. Ai, E. L. Novak, "*Centroid Approach for estimating modulation peak in broad-bandwidth interferometry*", US Patent 5633715, 1995
- [17]: K. G. Larkin, "*Efficient nonlinear algorithm for envelop detection in white light interferometry*", J. Opt. Soc. Am. A, Bd. 13, Nr. 4, S. 832-843, 1996
- [18]: Kjell J. Gasvik, "*Optical Metrology*", John Wiley & Sons, Ltd, 3. Ausgabe, 2002, 0-470-84300-4
- [19]: K. G. Larkin, B. F. Oreb, "*Design and assessment of symmetrical phase-shifting algorithms*", J. Opt. Soc. Am. A, Bd. 10, Nr. 9, S. 1740-1748, 1992
- [20]: V Alvarez-Valado et al, "*Testing phase-shifting algorithms for uncertainty evaluation in interferometric gauge blockcalibration*", Metrologia, Bd. 46, Nr. 6, S. 637-645, 2009
- [21]: C. Rathjen, "*Statistische Eigenschaften von Phasenschiebverfahren*", 3. ABW Workshop, 1996, Universität Hannover

- [22]: D. W. Phillion, "General methods for generating phase-shifting interferometry algorithms", Applied Optics, Bd. 36, Nr. 31, S. 8098-8115, 1997
- [23]: Gordon S. Kino, Stanley S. C. Chim, "Mirau correlation microscope", Applied Optics, Bd. 29, Nr. 26, S. 3775-3783, 1990
- [24]: P. J. Caber, "Interferometric profiler for rough surfaces", Applied Optics, Bd. 32, Nr. 19, S. 3438-3441, 1993
- [25]: R. L. Allen, D. W. Mills, "Signal Analysis", IEEE Press, . Ausgabe, 2004, 0-471-23441-9
- [26]: Chiayu Ai, E. L. Novak, "Centroid approach for Estimating Modulation Peak in Broadband Interferometry", US Patent 5633715, 1997
- [27]: Richard G. Lyons, "Understanding Digital Signal Processing", Prentice Hall PTR, 2. Ausgabe, 2004, 0-13-108989-7
- [28]: P. de Groot, "Method and apparatus for surface topography measurement by spatial-frequency analysis of interferograms", US Patent 5398113, 1993
- [29]: P. de Groot et al., "Determination of fringe order in white-light interference microscopy", Applied Optics, Bd. 41, Nr. 22, S. 4571-4578, 2002
- [30]: J. Stein, "Digital Signal Processing", John Wiley & Sons, 1. Ausgabe, 2000, 0-471-29546-9
- [31]: A. Mertins, "Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications", John Wiley & Sons Ltd, 1. Ausgabe, 1999, 0-471-23441-9
- [32]: M. Li, C. Quan, C. J. Tay, "Continuous wavelet transform for micro-component profile measurement using vertical scanning interferometry", Optics & Laser Technology, Bd. 40, Nr. 7, S. 920-929, 2008
- [33]: P. Ettl, "Über die Signalentstehung bei Weißlichtinterferometrie", 2001, Universität Erlangen-Nürnberg
- [34]: P. Pavliček, O. Hýbl, "White-light interferometry on rough surfaces - measurement uncertainty caused by surface roughness", Applied Optics, Bd. 47, Nr. 16, S. 2941-2949, 2008
- [35]: A. Harasaki, J. Schmit, J. C. Wyant, "Offset of coherent envelope position due to phase change on reflection", Applied Optics, Bd. 40, Nr. 13, S. 2102-2106, 2001
- [36]: A. Harasaki, J. C. Wyant, "Fringe modulation skewing effect in white-light vertical scanning interferometry", Applied Optics, Bd. 39, Nr. 13, S. 2101-2106, 2000
- [37]: P. Lehmann, "Systematic effects in coherence peak and phase evaluation of signals obtained with vertical scanning white-light Mirau interferometer", Proc. of SPIE, Bd. 6188, Nr. 11, S. 1-11, 2006
- [38]: F. Gao, R. K. Leach, J. Petzing, J. M. Coupland, "Surface measurement errors using commercial scanning white light interferometers", Meas. Sci. Technol., Bd. 19, Nr. 1, S. 015303, 2007
- [39]: J. M. Coupland, J. Lobera, "Measurement of Steep Surfaces Using White Light Interferometry", Strain, Bd. 46, Nr. 1, S. 69-78, 2010
- [40]: "Fundamentals of Piezomechanics", Physik Instrumente (PI) GmbH & Co. KG, <http://www.physikinstrumente.com/en/products/prdetail.php?sortnr=400600.20>, 10.2010
- [41]: H. Janocha, D. Pesotski, K. Kuhnen, "FPGA-Based Compensator of Hysteretic Actuator Nonlinearities for Highly Dynamic Applications", Transactions on Mechatronics, Bd. 13, Nr. 1, S. 112-116, 2010
- [42]: J. Nakamura, "Image Sensors and Signal Processing for Digital Still Cameras", Taylor & Francis Group, LLC, 1. Ausgabe, 2006, 0-8493-3545-0

- [43]: "*CCD Noise Sources and Signal-to-Noise Ratio*", T. J. Fellers, M. W. Davidson, <http://learn.hamamatsu.com/articles/ccdsnr.html>, 5.2010
- [44]: "*NewView 600 Operation Manual OMP-0528B*", Zygo Corporation, 2007
- [45]: "*Cree® XLamp® XP-G LEDs Data Sheet*", Cree, 2010
- [46]: W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, "*Numerical Recipes*", Cambridge University Press, 3. Ausgabe, 2007, 0-521-88068-8
- [47]: H. Erikson, M. Penker, B. Lyons, D. Fado, "*UML 2 Toolkit*", Wiley Publishing, Inc., 1. Ausgabe, 2004, 0-471-46361-2
- [48]: B. Oestereich, "*Analyse und Design mit UML 2.3*", Oldenbourg, 9. Ausgabe, 2009, 3486588559
- [49]: Bjarne Stroustrup, "*The C++ Programming Language*", Addison-Wesley, 3. Ausgabe, 1997, 0-201-88954-4
- [50]: H. Sutter, A. Alexandrescu, "*C++ Coding Standards: 101 Rules, Guidelines, and Best Practices*", Addison Wesley Professional, 1. Ausgabe, 2004, 0-321-11358-6
- [51]: "*doxygen*", <http://www.stack.nl/~dimitri/doxygen/docblocks.html>, 5.2010
- [52]: D. Abrahams, A. Gurtovoy, "*C++ Template Metaprogramming*", Addison Wesley Professional, 1. Ausgabe, 2004, 0-321-22725-5
- [53]: E. Gamma, R. Helm, R. Johnson, J. Vlissides, "*Entwurfsmuster*", Addison-Wesley, 1. Ausgabe, 1996, 3-8273-1862-9
- [54]: "*Data Sheet PCI-6070E*", National Instruments, 2004
- [55]: "*Microscope Objective Specifications*", M. Davidson, <http://www.microscopyu.com/articles/optics/objectivespecs.html>, 2.2010
- [56]: "*Relative Spectral Radiance for Quartz Halogen Lamps*", Dolan Jenner, http://www.dolan-jenner.com/PDFS/EKE_EJV_SpectralCurve.pdf, 12.2010
- [57]: "*Nutzung von Trigger und digitalen I/Os*", The Imaging Source Europe GmbH, http://www.theimagingsource.com/downloads/dxxxxbfxtrigio.en_US.pdf, 1.2010
- [58]: "*Datenblatt ICX205AL*", SONY, 2002
- [59]: R. Hartley, A. Zissermann, "*Multiple View Geometry in Computer Vision*", Cambridge University Press, 2. Ausgabe, 2004, 0521540518
- [60]: Gilbert Strang, "*Lineare Algebra*", Springer, 1. Ausgabe, 2003, 3-540-43949-8