

# Automated High-Precision Crack Detection in Airplane Combustion Chamber Liners Using White Light Interferometry

DISSERTATION

zur Erlangung des Grades eines Doktors  
der Ingenieurwissenschaften

vorgelegt von

Marc-André Otto, M.Sc.

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät  
der Universität Siegen

Siegen 2020

Betreuer und erster Gutachter  
Prof. Dr.-Ing. habil. Otmar Loffeld  
Universität Siegen

Zweiter Gutachter  
Prof. Dr.-Ing. Thorsten Schüppstuhl  
Technische Universität Hamburg

Tag der mündlichen Prüfung  
20.11.2020

## Zusammenfassung

Gerade in der Flugzeugindustrie ist es von zentraler Bedeutung, teure Investitionsgüter langfristig profitabel und sicher zu gestalten, weshalb die Instandhaltung (Englisch: *maintenance, repair, and overhaul* (MRO)) eine besonders große Rolle spielt.

Die Grundlage eines jeden Instandhaltungsprozesses ist die Inspektionstechnologie, welche in der Lage sein muss, alle relevanten Auffälligkeiten zu detektieren. Speziell bei sicherheitsrelevanten Bauteilen sind die Anforderungen an die Detektionsleistung sehr hoch. Weiterhin sind die Detektionsprozesse gerade bei komplexen und gebrauchten Bauteilen schwierig zu automatisieren, da sie häufig einen hohen Grad an Flexibilität erfordern. Deshalb sind zum aktuellen Zeitpunkt viele der in der Luftfahrt eingesetzten Inspektionstechnologien noch vollständig oder zumindest teilweise manuelle Prozesse.

In dieser Dissertation wird zunächst ein Überblick über die zerstörungsfreien Inspektionstechnologien in der Flugzeuginstandhaltung gegeben, gefolgt von Beispielen zur automatischen Anwendung dieser Prinzipien. Eine Technologie, welche bisher nicht automatisiert eingesetzt wird, aber ein sehr hohes Potenzial für die Etablierung eines automatischen Inspektionsprozesses für Luftfahrtbauteile aufweist, ist die Weißlichtinterferometrie. Leider hat diese Technologie Eigenschaften, welche die Entwicklung eines automatisierten Inspektionssystems schwierig machen. Da das Bildfeld des Sensors im Allgemeinen sehr viel kleiner als das zu inspizierende Bauteil ist und der Sensor gleichzeitig einen beschränkten Tiefenmessbereich hat, muss es während des Aufnahmeprozesses zu Relativbewegungen zwischen Sensor und Bauteil kommen. Die Technologie reagiert jedoch sehr empfindlich auf Schwingungen, was besondere Anforderungen an die Sensorintegration stellt. Weiterhin ist der Parameterraum des Sensors relativ groß und die Parameter hängen zudem teilweise voneinander ab, was die automatische Konfiguration des Sensors erschwert. Aus diesen Gründen wird die Weißlichtinterferometrie nach aktuellem Stand der Technik nicht in automatisierten MRO Prozessen eingesetzt.

Diese Dissertation schlägt eine Brücke zwischen der altbekannten Technologie der Weißlichtinterferometrie und deren praktischer Anwendung in automatischen Inspektionsprozessen. Um die Implikationen der Technologie vollständig zu verstehen, wurde zudem ein Überblick über die Theorie der Weißlichtinterferometrie hergeleitet. Dieser dient als Basis für die Lösungsfindung zu den praktischen Problemen, welche im Einsatz mit dieser Technologie auftreten. Obwohl die Technologie prinzipiell mit jedem beliebigen Bauteil genutzt werden kann, wurde bei der Entwicklung des neuartigen Inspektionsprozesses das äußere Flammrohr eines CFM-56 Flugzeugtriebwerks betrachtet, um die Technologie an einem praktischen Beispiel verifizieren zu können. Weiterhin wurde die Fehlerdetektion auf Risse beschränkt, da diese den Großteil der in diesem Bauteil auftretenden Schäden repräsentieren. Bei der Beschreibung der Entwicklung des neuartigen Inspektionsprozesses werden folgende Aspekte im Detail betrachtet: die Automatisierung der Weißlichtinterferometrie, die für die Rissdetektion notwendige drei- und zweidimensionale Datenverarbeitung sowie die Verifikation der potenziellen Rissanzeigen.

## Abstract

The maintenance, repair, and overhaul (MRO) industry plays an important role in increasing the long-term profitability and safety of large investments, especially in the aviation industry. MRO is, therefore, an important economic factor in its own right. The core of any successful MRO procedure is an appropriate inspection technique, which is capable of detecting all relevant damages in a given part. The requirements for these inspection procedures can be very high, especially for safety-critical parts. Furthermore, the inspection processes can be very challenging to automate, especially for complex and used parts, because they often require a high degree of flexibility. As a consequence, of the day of this writing, many of the inspection procedures in the aviation industry are still performed fully or at least partially manually.

In this dissertation, an overview of the most important non-destructive inspection (NDI) technologies in aviation MRO is given, followed by examples of their use in automated inspection systems. One technology that is not currently in automated use, but has a very high potential for the automatic high-resolution inspection of aviation parts, is white light interferometry. Unfortunately, this technology has several attributes that make it very challenging to integrate in automated processes. In general, a white light interferometer's field of view (FOV) is much smaller than the part under test, while simultaneously having a small depth of view. This means that relative movements between the part and the sensor are required. Furthermore, the technology is very sensitive to vibrations, necessitating special considerations for the sensor integration. In addition, the sensor has a relatively large parameter space, with partially interdependent settings, making it difficult to automatically find suitable parameter sets. For these reasons, white light interferometry is not currently used in automated MRO processes.

This dissertation bridges the gap between the well-established white light interferometry sensor technology and its practical use for automated inspection processes. In order to fully understand the technology's implication, an overview of the theory behind white light interferometry was derived and serves as the basis for developing solutions for the problems that arise when white light interferometers are used in practice.

Even though the novel inspection process itself can be used with any part, it was developed with the combustion chamber outer liner of a CFM-56 aircraft engine in mind, in order to verify the validity of the approach on a relevant practical example. Since cracks represent the majority of flaws in that particular part, the developed inspection method focuses on the detection of those damages. The description of the novel inspection approach focuses, in particular, on the following aspects: the automation of white light interferometry, the three- and two-dimensional data processing required to detect cracks and the verification of the potential crack indications.

## Acknowledgment

First, I would like to thank my first doctoral advisor, Prof. Dr.-Ing. habil. Otmar Loffeld, who gave me the chance to work on my Ph.D., always guided and supported me during this process and provided me with countless exceedingly helpful suggestions. Likewise, I would like to thank my second doctoral advisor, Prof. Dr.-Ing. Thorsten Schüppstuhl, for supporting me and for his extraordinarily helpful advice over the years. I would also like to thank Prof. Dr.-Ing. Claus-Peter Fritzen for chairing my Ph.D. committee. Finally, I would like to thank Prof. Dr. rer. nat. Volker Blanz for being a member of the committee.

I am honored for having been accepted into the University of Siegen's IPP Multi Sensorics program. I would like to express my gratitude to Dr.-Ing. Holger Nies, who was always very helpful with his advice.

It was a special privilege for me to be able to exchange ideas with researchers at both the ZESS *and* Prof. Schüppstuhl's Institut für Flugzeugproduktionstechnik of the Technical University of Hamburg. I would like to express my gratitude to all of my colleagues at both of these institutes for the many conversations and all the feedback I received.

It would have been impossible for me to work on my Ph.D. without the support of my employer, VMT Vision Machine Technik Bildverarbeitungssysteme GmbH in Mannheim. I am thankful to all of my coworkers but would like to explicitly express my gratitude to our CEO, Dr.-Ing. Stefan Gehlen, our CEO, Dr.-Ing. Michael Kleinkes, and our previous CTO, Dr.-Ing. Werner Neddermeyer. Furthermore, the research project AutoInspect would not have existed without Lufthansa Technik AG; I would, therefore, like to thank Lufthansa Technik AG for the chance to work on this research project. I would like to thank Michael Ernst, M.Sc., who managed the research project and fully supported my Ph.D. efforts. I would like to thank Hamburg's Behörde für Wirtschaft, Verkehr und Innovation for funding the project.

I would like to express my deepest gratitude to my colleagues who worked with me on the project (in chronological order): Dr.-Ing. Tomas Domaschke, Sönke Bahr, M.Sc., Tobias Flüh, B.Sc., and Katharina Studemund, M.Sc. Thank you all very much for supporting my Ph.D. efforts and for the time we spend together working on this project. Thank you for believing in me, even during times in which I did not believe in myself.

I would like to thank my friends at Team Choco Heidelberg, who were always there for me and provided some much needed balance in my life. I would especially like to thank my coaches Thomas Stoll, Ruadhán MacFadden, and our global head coach, Vincent Carlucci.

Working on my Ph.D., in addition to my job, was not always easy. During these difficult times, my family was always there for me. I would like to express my deepest thanks to my mother Heike, my father Thomas, and my brother Daniel. My wife, Maria, always appreciated how challenging the situation was for me and showed me her support. I dedicate this dissertation to her.

Marc-André Otto

# Table of Contents

<b>List of Figures</b>	<b>ii</b>
<b>List of Tables</b>	<b>v</b>
<b>List of Acronyms</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context . . . . .	1
1.2 Motivation for the Development of a Novel Inspection Method . . . . .	3
1.3 Constraints . . . . .	6
1.4 Research Objectives . . . . .	7
1.5 Structure of the Dissertation . . . . .	11
<b>2 State of the Art</b>	<b>13</b>
2.1 Flaw Inspection Technology Overview . . . . .	13
2.2 Existing Automated Crack Detection Systems . . . . .	32
2.3 Summary, Deficits, and Contributions . . . . .	38
<b>3 A Novel Approach for Automated Crack Detection</b>	<b>40</b>
3.1 Sensor Selection . . . . .	40
3.2 Architecture . . . . .	48
<b>4 Automated White Light Interferometer Scanning</b>	<b>57</b>
4.1 Depth Parameter Determination . . . . .	57
4.2 Time-Optimal Scanning . . . . .	61
4.3 White Light Interferometer Pre-Processing . . . . .	79
<b>5 Crack Detection in Airplane Combustion Chamber Liners</b>	<b>82</b>
5.1 Patch Splitting . . . . .	83
5.2 Scan Registration . . . . .	86
5.3 Scan Unification . . . . .	94

---

5.4	Generation of 2.5D Images . . . . .	105
5.5	Feature Detection . . . . .	119
5.6	Image Processing . . . . .	122
5.7	Crack Verification . . . . .	134
<b>6</b>	<b>Verification of the Method</b>	<b>143</b>
6.1	Daily Self-Check . . . . .	143
6.2	Probability of Detection . . . . .	150
<b>7</b>	<b>Summary, Conclusions and Outlook</b>	<b>155</b>
7.1	Summary and Conclusions . . . . .	155
7.2	Outlook . . . . .	156
	<b>Appendices</b>	<b>A1</b>
<b>A</b>	<b>Introduction to White Light Interferometry</b>	<b>A2</b>
A.1	Basics of Interferometry . . . . .	A2
A.2	White Light Interferometry . . . . .	A6
A.3	Derivation of the Resulting Correlogram for an LED . . . . .	A10
A.4	Measurement Method . . . . .	A31
<b>B</b>	<b>Inspection Technology Overview Table</b>	<b>A39</b>
<b>C</b>	<b>White Light Interferometer Driver Structure</b>	<b>A43</b>
<b>D</b>	<b>Error Handling</b>	<b>A45</b>
D.1	Software Freeze or Crash . . . . .	A45
D.2	Resume Strategy . . . . .	A47
D.3	White Light Interferometer Capture Errors . . . . .	A47
D.4	Error Handling Summary . . . . .	A49
	<b>Index</b>	<b>I</b>
	<b>List of References</b>	<b>III</b>

# List of Figures

1.1	Bathtub curve . . . . .	1
1.2	A CFM-56 outer liner . . . . .	4
1.3	Example for a patch . . . . .	10
2.1	Eddy current inspection . . . . .	16
2.2	Basic steps of magnetic particle inspection . . . . .	18
2.3	Circular and longitudinal magnetization . . . . .	19
2.4	Principle of dye penetrant inspection . . . . .	20
2.5	Pulse-echo and through-transmission . . . . .	22
2.6	Electromagnetic spectrum . . . . .	24
2.7	Ambiguity of camera projections . . . . .	25
2.8	Principle of white light interferometry. . . . .	27
2.9	Correlogram, envelope, and sampled points. . . . .	28
2.10	Comparison of different technologies and resolutions . . . . .	29
3.1	Effect of lateral movement on illuminated pixel locations. . . . .	41
3.2	Effect of lateral movement on the signal shape . . . . .	42
3.3	Effect of vertical movement on the output signal of a single pixel. . . . .	43
3.4	Over- and under-saturation . . . . .	45
3.5	Relevant coordinate systems . . . . .	49
3.6	Major system components and communication . . . . .	50
3.7	Patches and panels . . . . .	51
3.8	UML diagram of the main classes . . . . .	52
3.9	Main state machine . . . . .	53
3.10	Producer consumer structure . . . . .	54
3.11	Buffering . . . . .	55
4.1	Frame thickness and volume length . . . . .	57
4.2	Parameter space/volume length mapping . . . . .	58
4.3	Volume length lookup table . . . . .	60

---

4.4	S-curve motion profile . . . . .	66
4.5	S-curve motion profile with small jerk . . . . .	70
4.6	Influence of the trigger dead time . . . . .	74
4.7	Unshifted interferometer output . . . . .	75
4.8	Correctly shifted interferometer output . . . . .	75
4.9	Activity diagram of the adaptive scanning process . . . . .	78
4.10	AutoSense . . . . .	80
4.11	Intensity regression . . . . .	81
5.1	General crack-detection approach. . . . .	82
5.2	Patch Splitting . . . . .	83
5.3	Patch splitting with an incomplete arc . . . . .	85
5.4	Collection of scans . . . . .	86
5.5	Illustration of the transform determination steps. . . . .	89
5.6	Two overlapping scans . . . . .	90
5.7	Rectangle splitting using a central seed . . . . .	91
5.8	Alignment process . . . . .	92
5.9	Overlap detection . . . . .	93
5.10	Overlap transformation . . . . .	93
5.11	Matrix generation . . . . .	95
5.12	Overlap mapping . . . . .	96
5.13	UML class diagram: <code>ExtendedPoint</code> . . . . .	97
5.14	Pixel weights . . . . .	104
5.15	Image generation . . . . .	107
5.16	Different polynomial orders . . . . .	112
5.17	Typical input: Polynomial filtering . . . . .	113
5.18	Problematic input: Polynomial filtering . . . . .	114
5.19	RANSAC polynomial filtering . . . . .	118
5.20	Overnugget scan . . . . .	118
5.21	Laser line section scan . . . . .	120
5.22	Feature detection . . . . .	121
5.23	Feature registration . . . . .	123
5.24	Surface extension . . . . .	124
5.25	Pseudo high-pass filtering . . . . .	125
5.26	Canny edge detector . . . . .	126
5.27	Morphological closing . . . . .	128
5.28	Thinning algorithm . . . . .	129
5.29	Segmentation . . . . .	130
5.30	End vector calculation . . . . .	131

---

5.31	End vector area mapping . . . . .	131
5.32	Crack end continuation . . . . .	132
5.33	Illustration of the Hough transform . . . . .	133
5.34	Crack End Details . . . . .	136
5.35	Small segment removal . . . . .	137
5.36	Feature orientation . . . . .	138
5.37	Crack points to line mapping . . . . .	139
5.38	Region of interest creation . . . . .	140
5.39	Crack Sectioning Verification . . . . .	142
6.1	Radius estimation . . . . .	144
6.2	Amplitude filter . . . . .	146
6.3	Uniting results . . . . .	147
6.4	Secondary verification . . . . .	148
6.5	Rotational convolution and ICP . . . . .	149
6.6	Template scan result . . . . .	150
6.7	Hit/Miss diagram . . . . .	153
A.1	Michelson interferometer . . . . .	A4
A.2	Resulting interference waves . . . . .	A6
A.3	Wavelength delta/bandwidth mapping . . . . .	A8
A.4	Principle of white light interferometry. . . . .	A9
A.5	Power density spectrum of an LED . . . . .	A13
A.6	Idealized correlogram . . . . .	A21
A.7	Different correlograms . . . . .	A22
A.8	Doppler effect . . . . .	A23
A.9	Resulting modulation . . . . .	A25
A.10	Squared beat signal . . . . .	A27
A.11	Spectral broadening . . . . .	A28
A.12	Principle of quadrature demodulation. . . . .	A34
A.13	Correlogram, envelope, and sampled points. . . . .	A35
A.14	Maximum estimation with 1, 2, 3, and 4 samples . . . . .	A36
A.15	Maximum detection with five samples and significant noise. . . . .	A37
C.1	White light interferometer driver . . . . .	A44

# List of Tables

- 2.1 Attributes of the discussed inspection technologies . . . . . 31
- B.1 Advantages and disadvantages of inspection technologies . . . . . A39
- D.1 Possible errors during the scanning process. . . . . A46

# List of Acronyms

**2D** Two-dimensional.

**3D** Three-dimensional.

**AC** Alternating current.

**API** Application programming interface.

**CAAP** Civil Aviation Advisory Publication (of the civil aviation safety authority of Australia).

**CAD** Computer-aided Design.

**CCD** Charge-coupled device.

**CT** Computed tomography.

**DC** Direct current.

**DFT** Discrete Fourier transform.

**DLL** Dynamic-link library.

**DOF** Degree(s) of freedom.

**EASA** European Aviation Safety Agency.

**FAA** Federal Aviation Administration (of the USA).

**FFT** Fast Fourier transform.

**FOV** Field of view.

**FPGA** Field-programmable gate array.

**FPI** Fluorescent dye penetrant inspection.

**FWHM** Full width at half maximum.

**GPU** Graphics processing unit.

**IC** Integrated circuit.

**ICAO** International Civil Aviation Administration.

**ICP** Iterative closest points.

**IQ** In-phase and quadrature.

**IR** Infrared.

**LBA** Luftfahrt Bundesamt (Germany).

**LED** Light-emitting diode.

**LLS** Laser line section scanner.

**LM** Levenberg-Marquardt algorithm.

**LUT** Lookup table.

**MRO** Maintenance, repair, and overhaul.

**NDI** Non-destructive inspection.

**PDS** Power density spectrum.

**PLC** Programmable logic controller.

**POD** Probability of detection.

**RAM** Random access memory.

**RANSAC** Random sample consensus.

**RGB** Red, green, and blue (color model).

**ROI** Region of interest.

**SNR** Signal to noise ratio.

**SVD** Singular value decomposition.

**TCP** Tool center point.

**TCP/IP** Transmission control protocol over internet protocol (TCP/IP).

**UML** Unified modeling language.

**USB** Universal serial bus.

**UV** Ultraviolet.

**WLI** White light interferometer.

# Chapter 1

## Introduction

This dissertation is concerned with automated crack detection in aviation parts, demonstrated on airplane combustion chamber liners. In this chapter, some context for the research presented in this dissertation is given, followed by summaries of the specific research constraints and objectives. Finally, the structure of the dissertation is presented.

### 1.1 Context

All components are subject to potential failures, the probability of which follows a “bathtub curve” in many cases [AM02, p. 4], see Figure 1.1. The curve is the superposition of three contributions: first, the chance of failure due to manufacturing errors; second, the random chance of failure; and last, the chance of failure due to wear and tear. Especially for expensive and/or safety-critical parts,

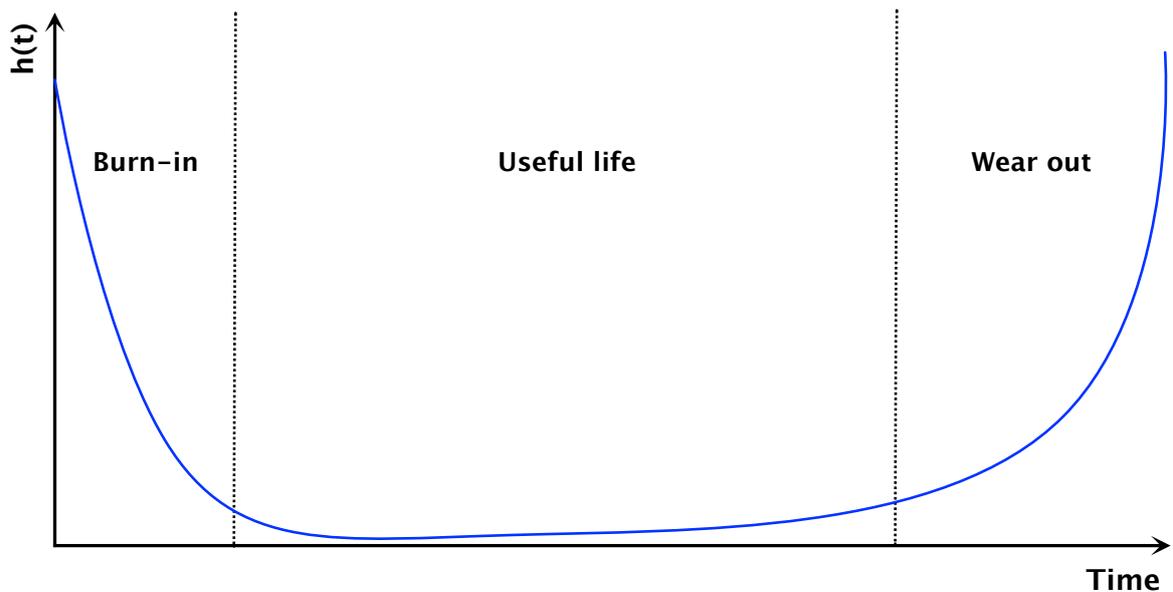


Figure 1.1: Bathtub curve, based on [AM02, p. 4].

the “useful life” period should be as long as possible. This time span is dependent on the chance of random failure, which depends on several factors, such as adequate engineering, manufacturing quality, choice of suitable materials and how the part is maintained during its lifetime. Failures due to manufacturing errors (burn-in period) can mostly be taken out of the equation by performing so-called burn-in treatments, meaning that new parts are subjected to stress before commissioning. This ensures that only reliable parts are placed into service.

The serviceable time of parts, on the other hand, can be increased by performing regular checks and repairs. This is of particular importance when it comes to safety-critical parts, such as aircraft components. By performing regular check-ups, repairs, and replacements, the airworthiness of an aircraft can be maintained over a long period of time. Bräunling states that aircraft can be used for 50 years or more [Brä01, p. 452]. In a white paper published by AVOLON, Foster writes that the average time to retirement for aviation parts is 25.7 years<sup>1</sup> as of 2015 [For15, p. 2]. These servicing measures are known as maintenance, repair, and overhaul (MRO), which has become an important industry, as recognized by acatec in 2015 [Vor+15]. The regeneration and inspection of complex parts are even being researched by a *Sonderforschungsbereich* in Hannover, Germany (collaborative research center SFB871).

For the aviation industry, the requirements for the involved processes and the frequency of inspections are governed by local<sup>2</sup> and global authorities, such as the International Civil Aviation Organization (ICAO). Details specific to individual components are usually defined in the manufacturer manuals. This is because only the manufacturer has the intellectual property necessary to engineer the appropriate MRO procedures, such as the specific materials used and the exact design specifications of the part.

The information in the following paragraph is based on [KM07, pp. 290 and 308]: there are several maintenance check levels, the so-called “letter checks:” A, B, C, intermediate layer (IL), and D, from least to most involved. For instance, an A check is performed about every six to eight weeks and only takes about 100–200 man-hours to complete. On the other end of the spectrum, a D check is performed every six to ten years and typically takes six to eight weeks to complete. It involves a complete overhaul of the aircraft, including disassembly of all components and even complete removal of the fuselage paint for defect inspection. Grounding an aircraft for such a long time is cost-intensive. It is, therefore, desirable for the airlines to reduce that time as much as possible. Thus, new technologies that are capable of reducing the inspection time, while still fulfilling the repair specifications, are in high demand. Additionally, increased accuracy or reproducibility might further increase the marketability of the method. Finally, it may also be beneficial to increase the information density and/or create a permanent record of a given component’s history, for instance for the aircraft leasing market.

One of the most critical aircraft components are the engines. The insides of engine combustion chambers can reach about 2300K [Brä01, p. 216], which far exceeds the allowed temperature range of

<sup>1</sup>Compensating for inactive time in storage; otherwise, 27.2 years.

<sup>2</sup>For instance, the Luftfahrt Bundesamt (LBA) in Germany, or the Federal Aviation Administration (FAA) in the US.

even highly specialized alloys, such as Hastelloy® X.<sup>1</sup> Even though several mechanisms are employed to protect the combustion chamber walls, such as cooling airflows, protective coatings, and specialized designs [Brä01, pp. 216 and 1169], the components are still subject to tremendous amounts of stress. Unfortunately, engines and engine components are very expensive. Therefore, it can be preferable to repair these components instead of replacing them. However, before the part can be repaired, the damages have to be located. Afterward, a repair strategy can be developed and executed if economically feasible. In the case of combustion chamber liners, all of these steps are currently performed manually. Even though manual processes have the advantage of being very adaptive, they have the distinct disadvantage of having low reproducibility, both when it comes to the reliability of results and time requirements. The inspections of combustion chamber liners are typically performed with fluorescent dye penetrant inspection (FPI, see section 2.1.3), which has additional disadvantages, such as potential health impacts.

## **1.2 Motivation for the Development of a Novel Inspection Method**

With the situation presented in the previous section in mind, it can be concluded that it would be beneficial to investigate an automated inspection solution that will accomplish the following:

- Reduces inspection times
- Matches or exceeds the reliability of manual inspections
- Increases reproducibility
- Increases the information density in order to allow automatic repairs (which may reduce the turnover time even more)
- Has none of the distinct downsides associated with FPI
- Has lower or at least comparable associated costs

There are several considerations associated with the development of new inspection methods. The first is how versatile the method needs to be, or, more specifically, whether only one particular component, an entire component family, or even several unrelated parts are required to be measured. The scope of parts is relevant because a measurement machine needs to be able to be large enough and be able to handle the different components' masses and moments of inertia, as well as be able to reach all of the areas that need to be inspected. Even for a single part, suitable sensors have to be selected, and algorithms have to be developed with respect to different surface states, the part's geometry, volume, and potential undercuts. This task is even more challenging if more than one component type has to be inspected.

---

<sup>1</sup>The Hastelloy® X data sheet states that the material was tested to withstand a temperature of 1450K for 8700 hours [Hay97, p. 3]. The melting range of the alloy is specified as 1530K–1630K [ibidem, p. 10].

The next consideration is what exactly has to be detected during the inspection. Potential damages can be a global change, such as large geometric deformations, or local ones, such as burn spots, cracks, and dents. Finally, the constraints for detecting those damages in the selected parts have to be specified. These include resolution, reproducibility, and maximum allowable time.



*Figure 1.2: A CFM-56 outer liner. The liner has been heavily repaired before and is no longer an active component.*

The ultimate goal would be to develop a fully-automated inspection approach capable of inspecting any kind of part, resulting in higher and more economical throughput. However, developing such a generic solution is almost impossible due to the large parameter space involved. Because of that, it is more feasible to select a part that can demonstrate the potential capabilities of a novel inspection approach instead of finding a universal solution right away. The part should be sufficiently challenging to inspect to truly demonstrate the functionality of the researched method. A CFM International “CFM-56” combustion chamber outer liner was chosen to demonstrate the method’s applicability since it is relatively large and has a somewhat complex shape, see Figure 1.2. Moreover, because it is a component that is regularly subjected to tremendous stress, particularly during starting and landing, it has a comparatively high likelihood of containing damage when it is inspected. All of this makes it an ideal candidate for verifying the proposed method.

Regarding the scope of damages that are to be detected, it was found in [SKS14] that cracks make up 76% of all damages in airplane combustion chamber liners. Thus, it makes sense to focus the conducted research on crack detection. As will be discussed in the state of the art (chapter 2), that kind of damage can be detected with an established state-of-the-art technology: fluorescent dye penetrant inspection (FPI). This technology has a potential resolution that is down in the micrometer

range. Thus, in order to develop a potential replacement for that technology, a sensor with a very high resolution is required. As will be discussed in chapter 2, the only suitable three-dimensional optical sensor technology to achieve this is white light interferometry. Unfortunately, with the current state of the art, white light interferometers (WLIs) are notoriously difficult to use in an automated system, as will be discussed in chapter 3.

Because the parts under consideration are much larger than the field of view of the sensor, a relative movement between the part and the sensor has to be performed [ODH13]. Furthermore, the sensor only has a relatively small measurement depth, and the observed parts are used and thus, usually differ from their original states. Because of this, the handling components' paths may have to be adapted to the current object's state. In order to perform this adaptation in a time-optimal way, the interferometer's linear axis, which is required to perform interferometric measurements anyway, is utilized to execute the adaptation movements. A secondary sensor with a lower resolution and much larger measurement depth is required [DSO14] to be able to safely observe the object's current state within a reasonable amount of time. This data can also be used to supplement the interferometer data, leading to a multi-modal inspection scheme. For the given materials, white light interferometry is a purely topological inspection technology. As such, it can only be used to detect defects that manifest themselves in some way in a surface change. This means that purely internal damages cannot be detected with this approach. But since the considered damages (cracks) are topological in nature, this is not an issue. It is essential that an optical measurement system cannot be fooled by surface discolorations, which is guaranteed if three-dimensional sensors are used. White light interferometry is, therefore, exceedingly relevant for the inspection of airplane combustion chamber liners.

It will be shown in section 3 that the integration of a white light interferometer in an automated inspection system and the processing of interferometer data are quite challenging. Those tasks are the core problems covered in this dissertation. Since the hardware components required for solving the automation problem (computers, sensors, and handling systems) are available on the market, any research conducted in this area has to go beyond a purely theoretical or simulative approach. The goal is rather to design, implement, and analyze a functioning prototype after a sound theoretical analysis of all relevant aspects. The use of white light interferometers in automated processes is, as will be shown in the literature review, still in its infancy. Since white light interferometry has several outstanding properties it is of great interest from both the scientific and industrial point of view to research ways to implement an automated inspection solution based on white light interferometry. *Lufthansa Technik AG* in Hamburg thus started the research project "AutoInspect," which was publicly funded by the *Hamburg Behörde für Wirtschaft, Verkehr und Innovation* in April 2011. The author's research on using white light interferometry for automated crack inspection, which was conducted at *VMT Vision Machine Technik Bildverarbeitungssysteme GmbH* (Mannheim) for that project, is presented in this dissertation. The research on handling (except for the aspect of handling the interferometer's axis) is not the main focus of this dissertation since the concepts of integrating the sensor in an automated system can be used with different kinds of handling machines. The handling software "AutoXIV"

and the adaptation algorithms that were used for the demonstrator were developed at the *Institut für Flugzeugproduktionstechnik (IFPT)* at the Hamburg University of Technology [Dom17].

### 1.3 Constraints

Any new method has to be at least as capable as the established method it seeks to replace. White light interferometry has a very high vertical resolution due to the properties of coherent white light (for full details, see Appendix A). This high resolution is necessary in order to capture the surface changes caused by cracks with high accuracy. During a measurement, the required signal modulation is created through a vertical scanning motion (known as “stitching”). Samples of the resulting signals are captured during this movement at different depths. These captured depth slices are called WLI frames. Aside from motion constraints, the recordable measurement depth depends on two factors: the sample buffer size and the vertical sample distance.<sup>1</sup> The lateral resolution, on the other hand, depends solely on the optical properties of the sensor and the properties of the pixel array.

Certain surfaces, such as typical metallic industrial surfaces, can feature a large spectrum of reflectivities, resulting in widely different intensities within a single field of view (FOV). In fact, the differences can be so large that the intensity range of the interferometric camera is exceeded. In order to solve this problem, a specialized solution in the pixel domain is required. Even though this technology exists,<sup>2</sup> it is only available in relatively small pixel arrays.

Since most real-world inspection problems far exceed the measurement volumes of the state-of-the-art sensors, several scans have to be taken from different positions, which then have to be combined into a large point cloud. Due to the enormous amounts of data captured during the digitization of large parts, the development of specialized data structures and algorithms is required, as presented in this dissertation.

The novel method discussed in this dissertation does not depend on a specific actuator, provided that the interferometer can be calibrated in a common coordinate system. What this means is that for any given handling configuration state, a frame can be determined that describes the current position and orientation of the sensor in that coordinate system. The coordinate system determination is, therefore, a tool center point (TCP) calibration, the specifics of which are discussed in section 3.2. Even though no specific handling hardware is required in theory, in order to achieve maximum flexibility, an industrial robot was used for the demonstrator. Furthermore, in order to achieve maximum speed, the part’s rotational symmetry was taken advantage of through the use of a rotational axis to move the part itself.

It was determined that in order to yield an economic advantage, the complete inspection and processing of a CFM-56 outer liner has to take less than 8 hours. Therefore, the processing time has to happen in parallel to the data capturing, and the data processing should be as fast as possible. As

---

<sup>1</sup>The distance between samples is not the depth resolution of the sensor, because the true signal form can be recovered, provided that the Shannon criterion is fulfilled.

<sup>2</sup>The author is contractually obligated not to reveal the specific sensor manufacturer used in this project.

long as the data processing is fast enough, the bulk of the inspection time is determined by how long it takes to capture a scan and the time the handling components need to prepare the next capture. As will be shown in section 3, most of the movements come from a combination of moving the part with an external axis and a one degree of freedom (DOF) adaptation movement through the interferometer's axis to adjust to the part's imperfect state. While the movements of the robot and the external rotational axis, as well as the adaptation calculations, are performed by the IFPT's control software AutoXIV, the actual adaptation movement has to be executed as part of each scan in preparation for the *next* scan, in order to save time. An additional time-saving measure is to adjust the measurement depth to the actual required range. Unfortunately, the measurement depth configuration depends on several different parameters (see chapter 4), some of which are interrelated, meaning that a method to set an appropriate parameter set for any desired depth range had to be developed. Finally, the researched method has to be able to scan the aforementioned CFM-56 outer liner in different surface conditions, from matted dark gray to highly reflective and from polished to oxide-blasted. The surface conditions may even change drastically on one and the same part, for instance, if only one area was repaired before the inspection. It is a requirement that the measurement system can handle all of these surface states. Limiting surface constraints are that the ceramic coating of the part has to be fully removed and that the part has to be clean and dry.

## 1.4 Research Objectives

In this section, the research objectives are detailed, under special consideration of the constraints mentioned in the previous section.

Crack inspection can be broken down into two parts: crack detection (step 1) and crack verification (step 2). In the case of the established crack inspection method of FPI, both of those steps are performed fully manually. In practice, this means that the FPI inspection process is performed (step 1), and any indications that appear to be a crack are marked as such (step 2). As long as the parameters, such as penetrant, developer, and timing, are chosen adequately, and the process is performed expertly, surface-breaking cracks of the expected dimensions should be detectable within the required detection range. The FPI verification step is based on knowledge stemming from training and experience, as well as human intuition. Because of the human factor and the vastly different detection mechanisms, it is difficult to compare an automated solution based on an optical sensor directly to FPI. Instead, the so-called probability of detection (POD) can be used as a metric, which is demonstrated in the verification discussed in chapter 6.

The novel method is modeled after this two-step approach. However, whereas with FPI, all inspection areas can be observed at once, this is not possible with the interferometry-based approach due to the interferometer's small FOV. Therefore, several scans have to be captured and combined in order to arrive at a sufficiently large area. Due to time concerns, this is done in such a way that a relatively large area is scanned, split, and then processed. During the splitting and processing steps,

the next data set is captured in parallel. Thus, the novel approach can be more accurately described as a three-step process:

1. White light interferometer scanning
2. Crack detection
3. Crack verification

The detection step is based on pure signal processing in the case of the method introduced here. Because the detection is not model-based, even parts of a test object that are intended to be there, such as holes and edges, are going to be detected as potential damages. This means that the crack verification step has to be based on specific assumptions about cracks in order to make a meaningful distinction between cracks and other anomalies. The advantage is that the model-less detection step can be executed relatively fast and delivers a much smaller data set on which to perform a model-based verification.

One of the verification aspects designed and implemented for this dissertation is based on the relative location and orientation of potential cracks to the aforementioned features. Since these features are relatively large, they can be detected robustly using a laser line section sensor (LLS) scan, which can be executed in a relatively short amount of time. Since different parts can have different feature configurations and the exact locations are, therefore, unknown, a requirement is to detect feature templates in a given LLS point cloud. These features are then used to mask those areas from crack detection and as an input for the crack verification procedure. The crack inspection process is thus based on a multi-modal approach, combining white light interferometry and laser line section sensors.

Because of the unforgiving surface conditions of combustion chamber liners, potentially containing both highly reflective and dark gray matte areas in the same part, capturing it with a regular camera chip is difficult due to the required dynamic range. Because of this, an interferometer with a specialized optical chip was chosen, which allows scanning a surface even if the surface reflectivity changes drastically within a single FOV.

Even though the chosen white light interferometer is an off-the-shelf product, it is not designed for the task presented in this dissertation, and thus measures had to be taken to make it work in this context. Furthermore, even for non-automation uses, the interferometer has certain attributes that need to be addressed to yield meaningful data. Firstly, the depth distortion caused by the sensor's lens needs to be corrected. Secondly, since the sensor has to scan both while moving forward and backward, but is unaware of the actual movement of its linear axis, the captured data has to be flipped accordingly, including the depth corrections. Thirdly, invalid signal components are not marked as such, meaning that they have to be detected. To make the data usable in subsequent steps, interferometer pre-processing methods had to be designed and implemented.

As detailed before, the interferometer's small measurement volume requires the use of handling components to allow a complete scan of the part. While the handling and path planning was developed outside of this work, the actual depth adaptation motion is actuated with the interferometer's linear

axis and is, therefore, part of the scanning motion. In order to save time, this has to be done in a way that takes the dynamics of the linear axis, as well as the specifics of the interferometric capturing process into account. These particular aspects had to be analyzed, and a solution for time-optimal scanning and adaptation was found.

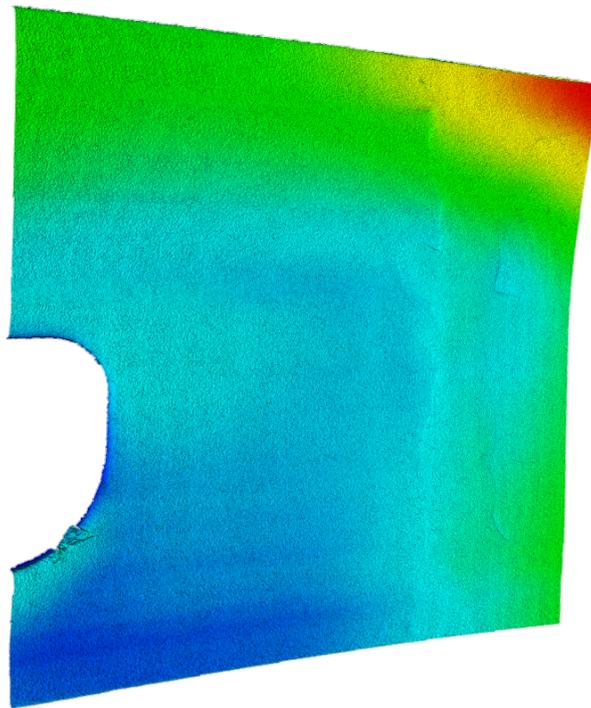
Even though the method introduced in this dissertation is actuator-agnostic, for the demonstrator handling system (see Figure 3.5), an industrial robot in combination with a calibrated external rotational axis was chosen for fast speeds and high flexibility. The robot system has the ability to calibrate the interferometer as its TCP. Since the relationship between the external rotational axis (depending on  $\varphi$ ) and the robot's base coordinate system is also known, the robot frame  ${}^A_S\mathbf{T}(\underline{q}, \varphi)$ , which is a homogenous transform matrix from the sensor coordinate system ( $S$ ) to the rotational axis coordinate system ( $A$ ), can be used to transform all of the interferometer scans from their sensor coordinate system into that of the external rotational axis, resulting in the raw point cloud representation of the scanned part. Systematic errors in the calibration and the assumed joint configuration will lead to a global offset. Positioning errors, which are ubiquitous in industrial robots due to non-linear effects, will lead to position and orientation deviations between the scans. This problem is greatly reduced because the aforementioned external rotational axis is used, exploiting the combustion chamber liner's approximate rotational symmetry. Despite that, the resulting accuracy is insufficient for creating a seamless point cloud. Therefore, a point cloud-based registration algorithm has to be employed. However, in many cases the individual scans do not have very strong features, so the outcome of a vanilla iterative closest points (ICP) approach often leads to unusable results. Because of this, a more suitable registration method had to be found.

Similarly, the measurement depth has to be adapted to the depth range present in every ring, created by each  $360^\circ$  turn of the external axis. Unfortunately, the measurement depth of a white light interferometer depends on many different interrelated parameters. Because of this, additional constraints had to be analyzed, and a method for finding optimum settings for any desired measurement depths had to be devised.

The time constraints mentioned in the previous section imply that heavy parallelization for both capturing and processing has to be used. In order to achieve maximum independence and efficient hardware use, capturing and processing had to be performed on two different computers. Thus, an architecture with minimum overhead for distributed capturing and processing was designed and implemented.

Since the resulting amount of data is too large to be processed all at once with state-of-the-art computer technology, the point cloud has to be divided into sub-point clouds, which are called "patches" in this dissertation. As will be shown in greater detail in section 3.2, natural structures of the test object are used for this division. The CFM-56 outer liner consists of different ring-wise elements, so-called "panels." Thus, the patches were chosen to be as high and wide as the height of the panel they were captured in, as shown in Figure 1.3.

In addition to parallel processing, the use of fast algorithms is mandatory, which is why as much



*Figure 1.3: Example for a patch: triangulated and height color-coded point cloud. The patch contains part of a feature (left) and a weld from a previous repair (right), as well as several scratches and grooves.*

of the processing of the generated patches as possible is performed in two-dimensional (2D) space. Even though the patch data is processed in two dimensions, the resulting data has to be represented in the common three-dimensional (3D) coordinate system (see above) so that it can be automatically marked or repaired. Thus, the results generated in their 2D representation have to be transformed back into 3D space. In other words, a means of efficiently transforming the point clouds from 3D to 2D space and back had to be found. Furthermore, the dynamic range of the 2D data had to be reduced without loss of relevant information so that images for use with image processing libraries could be generated.

It is insufficient to merely organize the individual scans into a set of scans representing a patch. Even after performing a registration algorithm resulting in a perfect alignment of the scans to each other, the data set still consists of individual point clouds. This is problematic for generating 2D images because the overlapping areas do not allow for a bijective mapping into a matrix. Thus, a fast unification method with optimum overlap point combination resulting in a single point cloud per patch had to be developed.

With tens of thousands of individual scans per combustion chamber liner that need to be captured with the interferometer, pre-processed, buffered, and sent to the processing system, it is likely that some errors will occur. Software-related fatal errors may crash either the capturing or processing software, in which case it would be time-consuming to restart the entire process. Because of that, all buffered and received results have to be stored so that an effective resume strategy can be used. In

other words, a robust resume-on-error strategy had to be developed. Since the interferometer was not designed to be used in an automated system, hardware-related errors may occur, which may lead to missing scans. Because of the highly parallelized capturing scheme, a scan cannot simply be retaken because the external rotational axis (see Figure 3.5) will have already moved the part to the next measurement position. Because a missing scan is unacceptable, a re-snap strategy had to be developed.

In sum, the overall research objective is to find an alternative and fully automated solution for crack inspection in complex and safety-critical machine parts: airplane combustion chamber liners. The applicability is to be demonstrated on a CFM-56 outer liner, but the general concepts shall be applicable to other complex goods. The measurement concept shall be based on a multi-modal approach with a white light interferometer as the primary and a laser line section scanner as the secondary sensor.

## 1.5 Structure of the Dissertation

After the motivation and research objectives have been established in the previous sections of the introduction, an overview of the current state of the art is required (chapter 2). Since this dissertation is concerned with automated crack detection using white light interferometry, the state of the art covers both other flaw inspection technologies (section 2.1) and existing crack inspection systems (section 2.2). Afterward, deficits in the current state of the art will be identified, and the author's contributions to the state of the art are highlighted in section 2.3.

Chapter 3 introduces the novel approach. First, the sensor requirements for the novel approach to crack detection are analyzed, and a concept is deduced in section 3.1, and the designed and developed architecture is introduced (section 3.2). This includes the mechanical system, software architecture, and inter-component communication.

The following chapter (chapter 4) covers the automated use of white light interferometers. This includes depth parameter determination (section 4.1) and a solution for time-optimal scanning (section 4.2). Afterward, the need for certain pre-processing steps is illustrated, and possible solutions are introduced in section 4.3.

Several steps are required for crack detection, which are covered in chapter 5. These include splitting the entire set of available scans into subsets (patch splitting, section 5.1), correcting the scans with a registration algorithm with selected degrees of freedom (section 5.2), unifying the individual scans into a large matrix, while also resolving ambiguities in the overlapping areas (section 5.3), the generation of 2.5D images (section 5.4), feature detection (section 5.5), and the image processing for the actual crack detection (section 5.6), as well as knowledge-based methods for the verification of the findings, which are presented in section 5.7.

In chapter 6, the proposed approach is verified, based on a self-check performed daily at Lufthansa Technik AG (section 6.1) and also based on statistics using probability of detection (section 6.2).

Finally, in chapter 7, the results are summarized and discussed in section 7.1, followed by an

outlook in section 7.2.

Since the main sensor used in this dissertation is a white light interferometer, an introduction to that technology is given in Appendix A. An additional overview table of the current state of the art is provided in Appendix B. Since the white light interferometer consists of hardware from two independent manufacturers, and, since no usable driver existed, a driver had to be designed and developed. The design is presented in Appendix C. Error handling plays an important part in software design. An error analysis and error handling design are presented in Appendix D.

## Chapter 2

# State of the Art

This dissertation is concerned with the automated crack inspection of safety-critical parts, airplane combustion chamber liners using a white light interferometer. Therefore, two aspects have to be considered in the state of the art:

1. Flaw inspection methods for safety-critical parts with special focus on the aviation industry.
2. Existing or actively researched automated crack detection systems, especially those using white light interferometry.

The analysis of the state of the art is accordingly split into two parts, even though there are certain overlaps.

### 2.1 Flaw Inspection Technology Overview

Because this dissertation is about flaw inspection with a specific sensor, the overview given in this section is also focused on the underlying inspection technologies that are used to yield crack indications. First, a brief overview of relevant flaw inspection categories is provided.

There are many different physical principles that can be used to generate the raw data required to detect cracks, resulting in a sheer endless variety of possible technologies. All inspection technologies can be subdivided into one of two categories:

1. Destructive inspection techniques ✗
2. Non-destructive inspection techniques (NDI/NDT) ✓

Of all possible technologies, only those from the latter group, non-destructive inspection (NDI), shall be considered. This is due to the fact that safety-critical parts, such as the CFM-56 airplane combustion chamber outer liners considered in this dissertation, require that every single part is inspected. An inspection method that alters, or even destroys, the component under test is therefore unsuitable in

this context.<sup>1</sup> Certain NDI techniques may also be used in a destructive manner; for instance, Dusek and Hunt describe a method of using dye penetrant inspection in a destructive analysis of lead-free solder joints [DH04]. Since these methods are destructive, they will not be considered in the following analysis.

Even though only non-destructive inspection methods are considered, introducing all available sensor technologies would be outside the scope of this dissertation. Therefore, only those technologies considered to be relevant for airplane inspection in the literature shall be discussed in the state of the art. Thus, these relevant technologies shall be identified, followed by a brief discussion of the selected technologies, and a summary of their individual strengths and weaknesses.

Since the part under consideration is an aircraft component, a document published by the United States Federal Aviation Administration (FAA), titled *Acceptable methods, techniques, and practices—aircraft inspection and repair* [Uni01], is used as a starting point to identify certified and commonly used inspection methods. In that document, 11 relevant NDI methods are listed<sup>2</sup> [ibidem, ch. 5, p. 6]: “Visual inspection, magnetic particle, penetrants, eddy current, radiography, ultrasonic, acoustic emission, thermography, holography, shearography, and tap testing.” Since holography and shearography are based on optical methods, they are combined here with visual inspection to form the group “visual or optical methods.” Ultrasonic and acoustic emission can be grouped together as “acoustic methods” [BL08]. The main difference between the two is that ultrasonic inspection induces acoustic energy into the object under test by means of a transducer and picks up the echo responses [Uni01, ch. 5, pp. 41] [BL08, pp. 24], whereas acoustic emission, as the name implies, picks up acoustic energy emitted by the object under mechanical or thermic load [Uni01, ch. 5, p. 53] [BL08, p. 27]. Tap testing essentially induces mechanical energy into the object, causing the emission of acoustic energy, which can then be interpreted to detect flaws.<sup>3</sup> Because the response of the test method is acoustic in nature, it will also be categorized as an acoustic inspection method in this dissertation. Thermography could potentially be categorized as an optical method, seeing how heat radiation can be captured by specialized pixel arrays designed to sense long-wave or mid-wave infrared [Kyl+14, p. 538], resulting in images similar to digital cameras sensing visible or near-visible light. Infrared thermography has sufficiently different characteristics, though, especially the ability to indicate imperfections even inside of opaque objects [BL08, p. 24], to justify putting it in its own category and will therefore be treated as such in this dissertation.

F. C. Campbell’s book *Inspection of Metals: Understanding the Basics* mentions visual/optical, magnetic particle, penetrants, eddy current, radiography, and ultrasonic [Cam13, p. 12] as relevant techniques, in agreement with [Uni01], but does not list acoustic emission, thermography, and tap testing. An earlier publication by the FAA [Ans+92] mentions the same six NDI methods as [Cam13].

---

<sup>1</sup>Destructive testing may, of course, still be used for random quality checks during production. It may also become relevant after a failure in order to analyze what led to the issue.

<sup>2</sup>It should be mentioned that these categories often contain subcategories, for instance, radiography contains x-ray and isotope (gamma-ray) radiography [Uni01, ch. 5, p. 6] [ibidem, ch. 5, pp. 37].

<sup>3</sup>The tap serves as a test input signal, resulting in a (poor) approximation of the system’s acoustic impulse response function.

In *Research into non-destructive compression crack detection methods for wood structures*, a study sponsored by the French Directorate General for Civil Aviation (DGAC), visual/optical, magnetic particle, penetrants, eddy current, radiography, ultrasonic, and acoustic emission are listed as NDI techniques [BL08, pp. 18]. In addition, permeability<sup>1</sup> testing is listed [ibidem, pp. 20]. However, this method is so specialized [ibidem, p. 33] that it will not be considered further in this dissertation. Thus, the following NDI categories can be established:

1. Eddy current testing
2. Magnetic particle inspection
3. Dye penetrant inspection
4. Radiography (including x- and  $\gamma$ -rays)
5. Acoustic inspection methods
6. Thermography
7. Optical and visual inspection

The following subsections provide an overview of the capabilities of relevant technologies from the above categories, as presented in the literature.

### 2.1.1 Eddy Current Testing

Eddy current testing is a widely used NDI technique [GGV11, p. 2526], owing to the fact that it allows the detection of defects even inside of objects with no contact [MHU07] [BHN00, p. 1]. This subsection serves only as a basic introduction to eddy current testing. For a more in-depth review of eddy current testing see [GGV11] [LWL00] [BHN00].

Eddy current testing is based on electromagnetic induction: an electric coil is coupled with the test object, which needs to be electrically conductive. When magnetic field lines go through an electric conductor, the changing magnetic field, in turn, induces currents that are encircling the magnetic field lines (Faraday's law). These currents are known as eddy currents. In accordance with Maxwell-Ampère's law, the currents induce a secondary magnetic field, affecting the primary one. The secondary field opposes the primary one, which effectively means that the coil's inductance changes [GGV11, pp. 2527]. On the other hand, since there is a current flow through a material with a resistance greater than  $0 \Omega$ , the necessary energy has to come from the primary alternating current, which means that the coil's resistance increases. Thus, the composition of the coil impedance changes to a coupled coil impedance  $\underline{Z}_c$ . The concrete nature of the change in coil impedance depends on the test object, especially on whether it is ferromagnetic [ibidem, pp. 2529]. The point at which there is no coupling between the coil and the object is termed the "air-point" [Boh12, p. 109] the complex coil impedance

---

<sup>1</sup>In this context, permeability is the ability of a solid to let a gas (or liquid) move through.

changes with the proximity to the test object. Conversely, the location on the complex impedance plane is thus a measure for the distance between the coil and the object (all other parameters being constant). This relationship is called “lift-off” [Boh12, pp. 108] [GGV11, p. 2529] [DUP07, p. 3.1].

When the coil is coupled with the test object, there is a given point in the complex impedance plane for the particular parameters of the coupling, including the distance between the coil and the object. If all parameters are kept constant and the coil is moved laterally over the test piece, the impedance will not change. This is because the unloaded impedance  $\underline{Z}_0$  is only altered to  $\underline{Z}_c$  by the coupling with the test object, and, if the coupling parameters remain constant, it can also be expected that there is no change in  $\underline{Z}_c$ . If, however, the material parameters are not constant, either due to inhomogeneous material properties or due to the presence of a defect, the coil impedance can also be expected to change. These changes are caused by a change in the eddy current path. For instance, the eddy currents cannot go through a defect and have to take a longer path around it but also potentially by a change in parameters such as relative permeability or material conductivity [GGV11, pp. 2533]. The changes due to a defect are illustrated in Figure 2.1. Thus, the changes of the complex coil impedance to  $\tilde{\underline{Z}}_c$

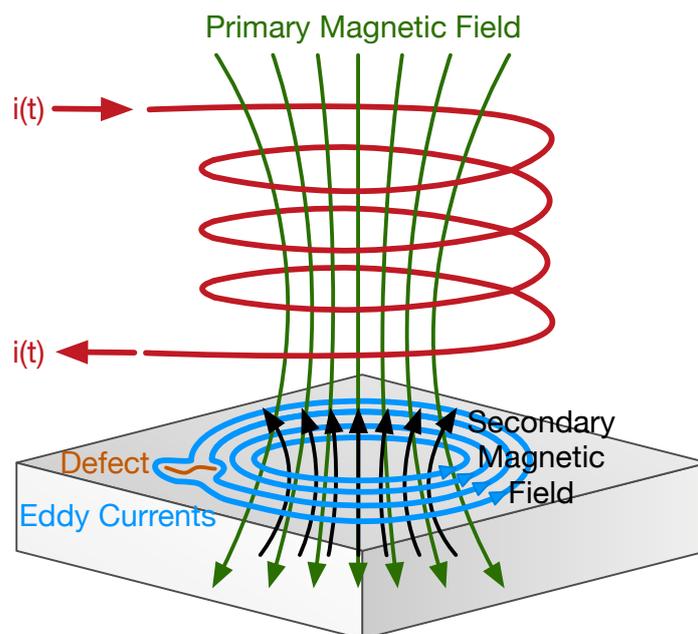


Figure 2.1: Eddy currents — gray: conductive material, red: coil current, green: primary magnetic field due to the current, blue: eddy currents due to the magnetic field, orange: defect, black: secondary magnetic field due to the eddy currents. Adapted from [GGV11].

can be used to detect defects [DUP07, p. 3.1]. One limitation of eddy current testing is that it can only detect defects that disturb the eddy current flow. Thus, it will not detect defects that are parallel to the eddy current flow.

Eddy currents can penetrate the material, depending on several parameters, such as the permeability and conductivity of the object, see for instance [MHU07, pp. 1] [BHN00, p. 649] [GGV11, p. 2540]. This means that eddy current inspection can be used to detect defects inside of the object;

however, this ability is directly limited by the attributes of the material itself.

Because the eddy current technique is essentially a point sensing technology (resulting in one complex impedance value per position), the actual scan of a part requires relative lateral movement between the part and the sensor [Boh12, p. 115]. The accuracy of those movements has to be relatively high since changes in orientation or distance can lead to incorrect readings [DUP07, p. 3.4]. Due to its dependency on material properties such as permeability and conductivity, eddy current testing can be used to analyze objects.<sup>1</sup> This, however, can be a disadvantage for crack detection because local parameter changes could lead to false positives. The detection of cracks smaller than the grain structure of materials can thus be difficult [Boh12, p. 109]. Ansley et al. go as far as to state that eddy current testing should only be used for inspecting steel parts if other methods cannot be used due to possible random (non-defective) changes inside the material [Ans+92, p. 13].

Furthermore, changes in material thickness can also lead to changes in readings, particularly if the wall thickness is about the size of the penetration depth [DUP07, p. 3.4]. Even too large surface roughness may lead to issues with the technique [Ans+92, p. 18]. According to [MHU07, pp. 4], there are several noise sources besides the aforementioned changes in material attributes and structures, including ambient fields and the sensor's circuitry. García-Martín, Gómez-Gil, and Vázquez-Sánchez mention that changes in temperature as a further potential noise source [GGV11, pp. 2544].

## 2.1.2 Magnetic Particle Inspection

Magnetic particle inspection uses the laws of electromagnetism, as does eddy current testing. The basic idea is to magnetize the object under test, which, therefore, needs to be ferromagnetic, and apply magnetic particles. This can be done either as a liquid or as a powder.<sup>2</sup> An undamaged part contains the entire magnetic flux. If, however, a damage is present, part of the magnetic field lines will leak outside the object under test, leading to increased magnetic field strengths in those areas. The field thus imparts a force on the applied particles, attracting them to the damaged areas. The outline of the damage can then be detected due to the higher concentration of magnetic particles [Cam13].

The basic steps of magnetic particle inspection are the following [Hea02]: first, the surface of the object may have to be prepared before testing, especially if the surface is coated [ibidem, p. 11] or if the surface is so coarse that it hinders proper alignment of the magnetic particles. The test object is then cleaned and demagnetized [ibidem]. In the next step, the actual inspection is performed as outlined in the previous paragraph, which is to say the part is magnetized, and the magnetic particles are applied. Afterward, the part may have to be demagnetized and cleaned. This is illustrated in Figure 2.2.

Even though the method can only be used to inspect ferromagnetic materials, its importance is stressed in the Health and Safety Executive HSE (UK) publication [ibidem, p. 4], which states that dye penetrant inspection (see next subsection) and magnetic particle inspection are the two most

<sup>1</sup>For the identification of material properties, an absolute probe has to be used.

<sup>2</sup>The FAA explicitly advises against using powders due to their lower sensitivity [Uni01, ch. 5, p. 22].

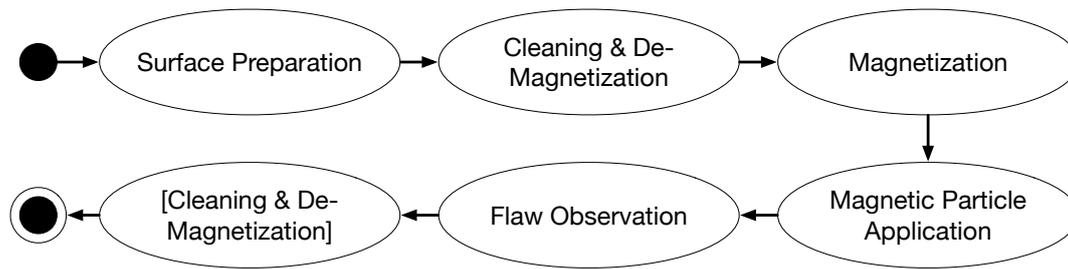


Figure 2.2: Basic steps of magnetic particle inspection, adapted from [Hea02].

widely used NDI methods. Indeed, according to [Hea02], magnetic particle inspection and dye penetrant inspection combined make up about 50% of all NDI inspections. Furthermore, magnetic particle inspection is usually preferred over dye penetrant inspection if the component can be magnetized because it can be applied with more flexibility with regard to inspection conditions, such as a certain robustness against (sufficiently thin) coatings [ibidem, p. 4]. Additionally, it is stated that magnetic particle inspection is quicker to apply than dye penetrant inspection [ibidem, p. 4]. All of that said, there are certain cases in which dye penetrant inspection shows greater sensitivity in ferromagnetic materials than magnetic particle inspection [Cam13, p. 13]. As always, the correct inspection solution has to be selected on a case-by-case basis.

It should be noted that the flux leakage only occurs when the magnetic field lines are actually interrupted by a defect. Thus, only defects that are oriented accordingly can be detected. The material can be magnetized in two general ways: longitudinally and circularly. Circular magnetization occurs when a current is passed through the test objects, resulting in encircling magnetic field lines that magnetize the object accordingly. Longitudinal magnetization can either be achieved by placing the test object inside a coil or by applying a magnetic field through either a yolk<sup>1</sup> or even a permanent magnet. Both magnetization methods and their effect on the detectability of differently oriented cracks are illustrated in Figure 2.3. If the direction of cracks is unknown, the part has to be subjected to more than one inspection pass. The United States FAA publication [Uni01, ch. 5, p. 25] mentions two passes, with the second one oriented at 90° to the first one.

In the case of surface-breaking defects, magnetic particle inspection enhances visibility. As with eddy current testing, complex parts might contain features that themselves can cause leakage and which may thus lead to false positives. When inspecting the part (either by a technician or through some form of image processing), these false indications must be recognized as such and treated accordingly.

The Australian Civil Aviation Safety Authority released an advisory publication regarding the use and implementation of magnetic particle inspection [Aus06]. In [ibidem, p. 11], it is stated that the method is able to detect defects of 2 mm in length if they break the surface. If a damage does not break the surface but is within a depth range of about 3 mm, the method can be expected to detect defects down to 5 mm in length. However, since the directional detection dependency holds, the actual

<sup>1</sup>Essentially, an electric horseshoe magnet.

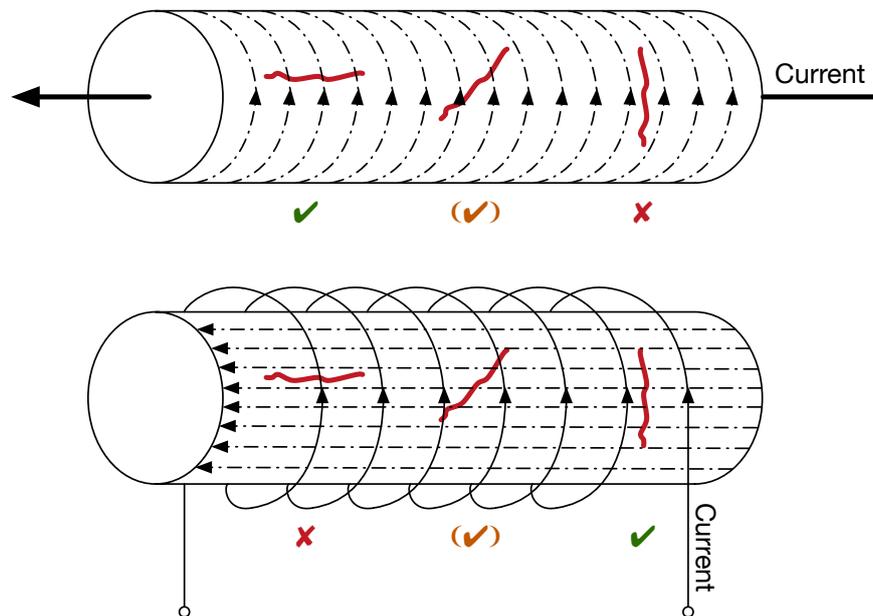


Figure 2.3: Circular and longitudinal magnetization of a simple test piece and effect on crack detection, adapted from [Ans+92]. Dashed lines are magnetic field lines.

detection performance in practice may vary greatly. Furthermore, [Hea02, page 11] states that, in some cases, magnetic particle inspection might detect even smaller defects. The Royal Australian Air Force assumes a minimum detectable crack length of 2 mm, which is confirmed for the average detection performance by Burke and Ditchburn [BD13, p. 3]. It is stated, though, that a detection performance of 2.6 mm can be expected in most cases [ibidem, p. 3].

### 2.1.3 Dye Penetrant Inspection

As mentioned in the previous subsection, dye penetrant inspection and magnetic particle inspection are the two most often used NDI methods in the aviation MRO industry, see, for instance, [Cam13, p. 11] and [Hea02, p. 4]. They are very similar in that they both highlight the location of defects with a suitable marking agent that is applied to the surface. Unlike magnetic particle inspection, which is based on magnetism, dye penetrant inspection relies on capillary action [Cam13, p. 13]. This means that dye penetrant inspection can only be used successfully with objects that do not have a very porous surface because, in that case, the surface's own porosity would interfere with the dye penetrant inspection process [Uni01, ch. 5, p. 29]. Dye penetrant inspection has two major advantages over magnetic particle inspection in that it does not require ferromagnetic materials and is not dependent on the orientation of the defects to the sensor.

Figure 2.4 illustrates the principle of dye penetrant inspection. In the first step, the surface is cleaned so that any obstructions are removed. This preparatory first step is important because unlike magnetic particle inspection, dye penetrant inspection can only detect surface-breaking cracks [Hea02, p. 11]. Next, a suitable penetrant is chosen and applied to the part (step 2) [ibidem, p. 5]. In the third

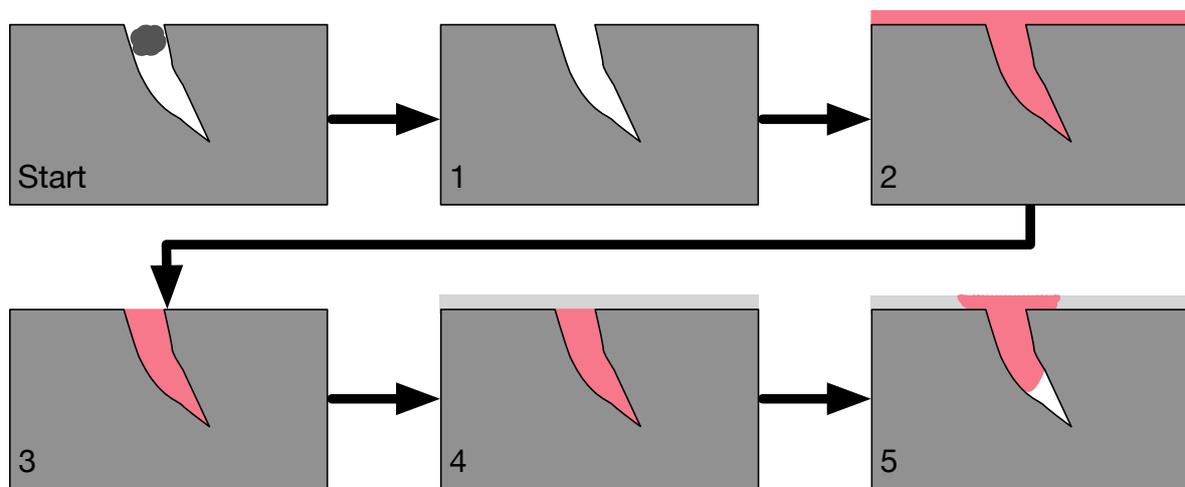


Figure 2.4: Principle of dye penetrant inspection, adapted from [Ans+92, p. 55].

step, the penetrant is removed from the surface. Next, a developer, a substance with higher capillary action than the expected defects, is applied (step 4). Dye penetrant is pulled out of the defects due to the developer, and thus their respective locations are highlighted. The fifth step is the inspection of the part, which is to say the highlighted parts are identified.

Penetrants can either be based on an inherent contrast to the developer or on fluorescence. Penetrant inspection with a fluorescent dye is known as fluorescent penetrant inspection (FPI) [Uni01, p. 34]. In the aviation industry, the latter plays by far the larger role since FPI provides increased detection rates, according to [Bra+04, p. 1]. With regard to jet engines, [ibidem, p. xiii] notes that 90% of all metallic components are subjected to FPI during product qualification and that it is an important aspect of maintenance and overhaul.

## 2.1.4 Radiography and Computed Tomography

Radiography is based on penetrating radiation [Uni01, ch. 5, p.37], mostly x-rays and  $\gamma$ -rays [Cam13, p. 16]. The basic principle of radiography is that the amount of absorption in matter depends on several aspects, such as material thickness and density. Since defective areas are, by definition, different from their surroundings, the absorption coefficient for radiation passing through the affected area is usually different from flawless areas. Thus, the general setup for detecting flaws through radiography is to aim the test radiation at the part and measure the amount of radiation that can be received at the other side of the object. This implies that the radiation has to be powerful enough to actually penetrate the part, yet weak enough for the defect to have a measurable effect.

It can therefore be concluded that the principal components of radiography include a radiation source, the test object, and a suitable detector. Whereas the previously introduced methods apply the contrast agent/energy on the same side where they capture the readings, radiographic signals are obtained from the opposite side. Thus, the object under test has to be accessible from both sides.

Another issue with radiography is that it has a certain inherent geometrical unsharpness. Therefore, the radiation source should be as far as possible away from and the detector as close as possible to the test object [Ale16, p. 17]. The intensity of the recorded signal depends on the radiation exposure time. The optimal exposure depends on many factors, such as the material and geometry of the object, the nature of the defects and the type and intensity of the radiation. In practice, not all of the variables that affect the quality of the obtained images are known and might have to be determined in a preparatory step [OT03, p. 122].

The detector observes a 2D projection of the object. Thus, it is impossible to determine the actual position of the flaw in the direction of the radiation. Furthermore, the detectability of defects depends on their alignment to the radiation/observation setup. According to [Ale16, p. 17], misaligned cracks can (at least partially) be invisible to the detector. In order to extract more information from the scene, more than one radiographic image needs to be captured and combined from different angles [Uni01, ch. 5, p. 37].

If enough images are captured, a volumetric 3D reconstruction of the object can be calculated. This is known as computed tomography (CT) (see, for instance, [Dös16, pp. 131] [BPF12, pp. 814]). The basic idea of CT is the following: for a single illumination with a parallel radiation source, each ray had to pass through the object and is then captured by the receiver. The captured intensity, therefore, depends on how much of the radiation was absorbed by the path, leading to an intensity distribution  $\mu(\underline{x})$  in terms of the detector position  $u$  and current object angle  $\varphi$  along a line, which is the Radon transform  $\mathfrak{R}(\mu(\underline{x}))$  of the absorbed radiation. Thus, if the part is scanned under several orientations, the absorption function, and therefore a volumetric representation of the scanned part can be calculated by inverting the Radon transform, through filtered back projection [Sun17, pp. 206].

## 2.1.5 Acoustic Inspection Methods

Acoustic inspection methods can be divided into two basic categories, external wave generation and wave generation by the part itself, which is called direct wave generation. In the following sections, both external and direct wave generation methods are introduced briefly.

### External Wave Generation (Transducers)

Sound waves travel through different materials with different velocities [GG06, pp. 381]. Inside an object that is homogenous in both its density and its characteristic sound velocity, the acoustic impedance does not change, allowing a sound wave to be transmitted through the medium. As soon as there is an interface between two materials with different acoustic impedances, part of the sound wave will be reflected at the interface [Cam13, p. 18]. The ratio between reflected and transmitted acoustic energy depends on the difference in acoustic impedance, with larger differences leading to larger amounts of reflection [GG06, p. 382]. Due to the fact that dry air at 20°C has an acoustic impedance of about 429 kg/(m<sup>2</sup>s) and metals in the range of tens of millions [ibidem, p. 383], metal/gas interfaces lead

to almost total reflection of the sound wave [Uni01, ch. 5, p. 41]. It is this echo effect that is used to detect flaws in an object.

External wave generation and reception are usually done using a transducer based on the piezoelectric effect [GG06, pp. 382]. Piezoelectric materials, such as lead titanate [ibidem, p. 383], release electric charges under mechanical strain and deform mechanically when exposed to an electric field. Thus, subjecting a piezoelectric object to an alternating current (AC) will make it oscillate with the frequency of the electrical signal. Conversely, the vibration caused by an ultrasonic wave deforms the transducer, resulting in a measurable electrical signal. Because piezoelectric signals typically have a large acoustic impedance, similar to metallic test objects [ibidem, p. 383], a couplant has to be used between the test object and the transducer [Uni01, ch. 5, p. 43]. Otherwise, most of the signal would be reflected at the transducer/air interface.

The actual defect inspection can be performed in two ways:

1. An ultrasonic pulse is sent into the object, and the echo is observed on the same side (pulse-echo method).
2. An ultrasonic pulse is sent into the object, and the transmission is observed on the opposite side (through-transmission method).

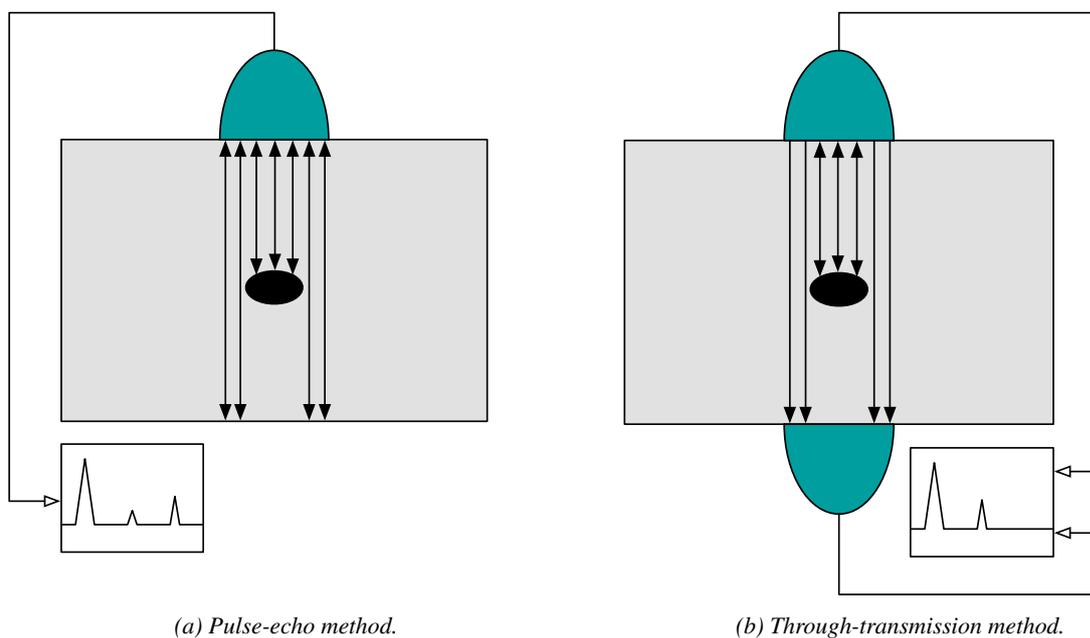


Figure 2.5: Pulse-echo (left) and through-transmission (right) methods illustrated (adapted from [GG06, p. 387] and [Uni01, ch. 5, p. 42]). The ultrasonic transducers are green. The gray blocks represent objects with (black oval) flaws in their centers. The resulting signals for the two different setups are shown on a signal monitor underneath.

The two different methods are illustrated in Figure 2.5. The pulse-echo method (Figure 2.5a) typically uses the same device for generating the ultrasonic pulse and receiving it: the ultrasound

transceiver. After the sound wave enters the test object, it travels through it until it encounters an interface. If no flaw is present, the wave travels all the way to the end of the object, is reflected at the solid/gas interface (assuming the object is contained in the air), and travels back to the transceiver. In that case, two peaks would be visible, one from a signal generated at the front of the object and one from when the wave is reflected at the end of the object. If, however, a flaw is present, the part of the wave that hits the flaw is reflected at the test object/flaw interface, provided that the acoustic properties of the test object are sufficiently different from those of the flaw. Acoustic flaw inspection thus relies on interpreting the time of arrival and the amplitude of pulses [Cam13, p. 19].

Whereas the pulse-echo method directly measures the reflection of the sound wave, the through-transmission method measures the echo indirectly by observing the part of the signal that has been transmitted, as shown in Figure 2.5b on the right. In order to get a reference signal, the emitted signal is typically fed into the signal monitor, as illustrated in Figure 2.5b. Crack inspection relies on comparing the amplitude of the reference signal to the received signal. The through-transmission method shows lower crack detection sensitivity than the pulse-echo method [Uni01, ch. 5, p. 41].

### Direct Wave Generation Methods

Direct wave generation is based on causing the part itself to release an acoustic signature, the interpretation of which allows for the inspection of the part. The simplest example for this approach is the use of a tap hammer, which generates acoustic waves as the system response upon impact. Since sufficiently dominant flaws can noticeably change the acoustic signature, a skilled worker can use this method to attest that a certain area of a part is flawed. This method is known as “tap testing” [ibidem, ch. 5, p. 53]. It can only be applied in a relatively limited set of circumstances and is highly subjective [ibidem, ch. 5, p. 53].

Another example of a mechanical direct wave generation method is acoustic emission testing. With this method, the test object is subjected to a suitable form of load. The load leads to a slight deformation of the part, which causes the release of characteristic sound waves [ibidem, ch. 5, p. 53]. These signals can be picked up by sensors, such as ultrasound transducers [Arr13, p. 4] attached to the part at various locations. This method has the advantage of allowing for the simultaneous inspection of the entire part, but, on the other hand, it does not yield specific flaw information, such as crack sizes [Uni01, ch. 5, p. 53].

Acoustic waves can also be generated utilizing a laser of sufficient energy [Miu+01, p. 3]. Two principal mechanisms can be used for wave generation: at lower power levels, the laser spot will cause local heating of the surface and, therefore, a fast expansion of the material, causing surface waves [GG06, p. 385]. At higher power levels, parts of the surface are vaporized, leading to propulsion that generates longitudinal waves [ibidem, p. 385]. In both cases, the detection of the acoustic signal can be done with a laser in a Michelson interferometer setup (see Appendix A) [Miu+01, p. 3]. This method has the advantage that measurements can be performed very rapidly and without surface contact.

A similar idea for causing vibrations in the part itself is the use of electromagnetism. However,

this method only works for conductive materials [GG06, p. 385].

## 2.1.6 Thermography

Heat can be transported in three ways [TM15, p. 633]:

1. Thermal conduction: transportation of heat inside matter.
2. Convection: transportation of heated matter.
3. Radiation.

According to the Stefan-Boltzmann law, every object with a temperature above absolute zero (0 K) emits radiation [ibidem, p. 639]. Observing this radiation is known as thermography. Outside of the visible range of wavelengths (red at around 700 to violet at around 400 nm [ibidem, p. 989]), follows higher energy/frequency radiation such as ultraviolet (UV) light, x-, and  $\gamma$ -rays, as well as lower energy radiation such as the infrared (IR) range, microwaves, and radio waves of various wavelengths (see Figure 2.6). The IR range can be further divided into [Iba+07, p. 2] the following subcategories: near-infrared (0.74 to 1  $\mu\text{m}$ ), short-wavelength IR (1 to 3  $\mu\text{m}$ ), mid-wavelength IR (3 to 8  $\mu\text{m}$ ), long-wavelength IR (8 to 14  $\mu\text{m}$ ), and very long-wavelength IR (14 to 1000  $\mu\text{m}$ ), also known as far-infrared. Of these types of IR radiation, near-infrared and short-wavelength IR can be considered non-thermal, and the remaining IR wavelength groups can be considered thermal, according to [IBM11, pp. 1].

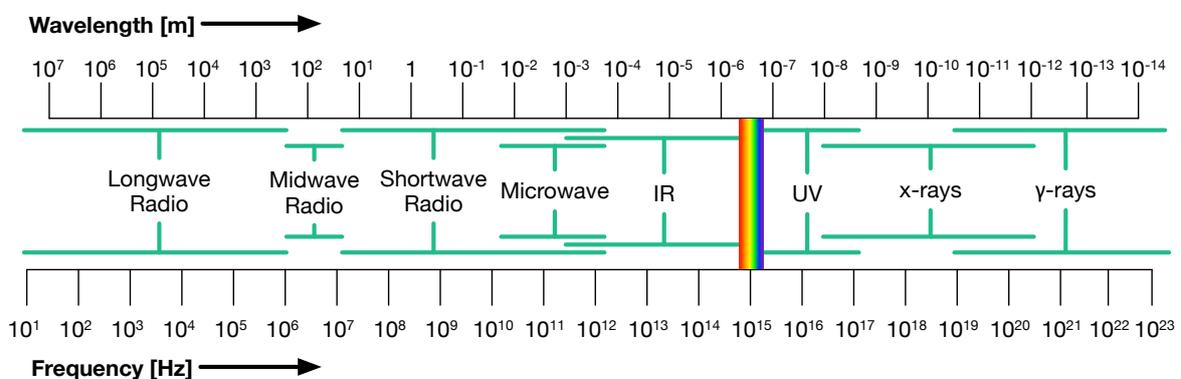


Figure 2.6: Electromagnetic spectrum (translated and adapted from [TM15, p. 989]).

If an object is heated above room temperature, the object will transfer heat to its surroundings until a new state of equilibrium is reached. The time required to transfer a given amount of heat energy depends on the thermal resistance of the material [TM15, p. 634]. This fact is used in active thermographic inspection: damage is characterized by a change, either in material properties, geometric changes, or loss of material. In all of these cases, a change of the thermal resistance in that area can be expected. Thus, if an object is heated, parts with higher thermal resistance will both heat up and release heat to its surroundings more slowly than other areas. Because of this, defective areas will have a different temperature from their surroundings until a state of equilibrium is reached. Thus, defects

can be detected through thermography, provided that the changes in thermal resistance are significant enough that they can be registered by the sensor. Such sensors are, in practice, so-called focal plane array cameras, operating in the mid-wavelength or long-wavelength IR range [Rog13] [Iba+07, p. 2].

### 2.1.7 Optical and Visual Inspection

Visual inspection, the act of identifying damages by the power of sight, is both the oldest and most widely used non-destructive testing method [Uni01, ch. 5, p. 9] [Cam13, p. 1]. The basic principle behind visual inspection is that light, electromagnetic radiation in the range of 400 to 700 nm wavelength, see Figure 2.6, interacts with matter in different ways, such as reflection and absorption. Features of the object that interacted with the light may be deduced from the intensity, color, shadowing, or shape. Matter may be opaque, in which case the light is only returned from an object's surface, or (partially) transparent,<sup>1</sup> in which case some of the inner properties of the object can potentially be inferred visually. Visual inspection thus uses the information that can be gathered from observing an object of interest visually to detect potential flaws.

A huge advantage of human inspectors over machine vision is that they can actively inspect objects [BPF12, p. 8], which is to say they intuitively adjust observation situations as necessary. Visual inspection is often performed with the naked eye but may be augmented with optical instruments such as magnifying glasses and borescopes [Ans+92, pp. 7]. Magnifying glasses are used to overcome resolution limitations in order to detect or verify small defects that would be otherwise hard to see. Borescopes, on the other hand, are optical instruments that allow the inspection of covered parts, provided that the borescope can be inserted and positioned at the area of interest. This might allow the inspection of parts without requiring a full disassembly [Uni01, ch. 5, p. 16].

The capturing of digital images of sufficient quality for solving a large class of automation tasks became feasible with the development of charge-coupled devices (CCD) in the mid-70s [Jäh12, p. 197]. Since then, industrial image processing has become an important part of automated inspection [BPF12, p. VII]. One essential problem with cameras is that they perform a projection of three-

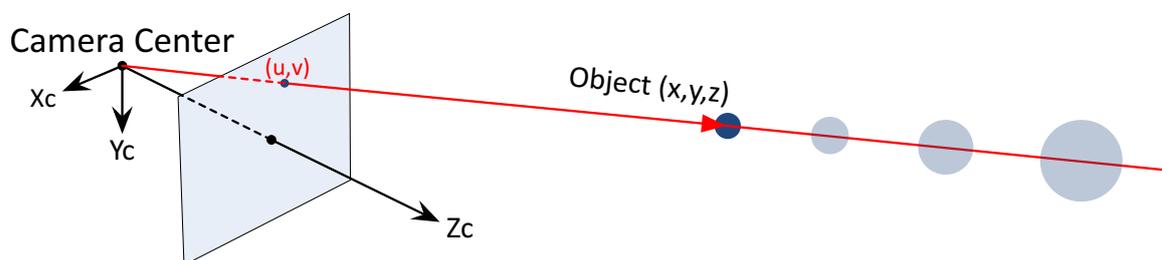


Figure 2.7: Illustration of the ambiguity of camera projections.

dimensional space onto a two-dimensional image plane:  $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ . In other words, the depth information is lost. This is illustrated in Figure 2.7: an object (blue in the image) with coordinates  $(x, y, z)$  is

<sup>1</sup>Transparency may depend on the light's wavelength.

projected onto the image plane, resulting in mapped coordinates  $(u, v)$ . If a line of sight is determined from the camera center through the image point  $(u, v)$ , there are an infinite number of solutions for  $z$  (light blue circles in the illustration): the set of equations is underdetermined. Therefore, if the missing depth information needs to be recovered, additional (linearly independent) information is required.

There are many different ways to perform three-dimensional optical sensing.<sup>1</sup> They can be broken down into active and passive methods.

One of the most well-known general principles for 3D sensing is triangulation (projective geometry), which has applications in both passive and active vision. As for active triangulation methods, these include structured light and laser line section scanning. Passive triangulation methods include the use of two (stereo vision) or more calibrated cameras that observe an object from different angles. Corresponding features are then detected in the different images, and the corresponding lines of sight are calculated. Ideally, the lines would intersect, resulting in a solution for the point. More realistically, the shortest distance between the two lines is calculated and used to estimate the  $(x, y, z)$  coordinates of the point. If more than two cameras are used, the linear set of equations is overdetermined, which can be solved in a least-squares sense.

There are several other optical 3D sensing methods that do not rely on projective geometry. One example is time of flight. Here, a light signal is emitted by the sensor, and the distance is determined by interpreting the returned signal. This may be done by pulse-ranging [Li14, p. 1], which is to say a pulse is emitted, and the absolute return time is determined. With the speed of light known,<sup>2</sup> the distance can be determined with  $c = 2 \cdot d/t \Leftrightarrow d = 1/2 \cdot c \cdot t$ . Another method is continuous-wave modulation. As the name implies, instead of emitting a single pulse, a continuous signal is emitted and received. Then the phase offset is determined, which results in a non-unique distance solution, since the phase is mapped into the interval  $[0; 2\pi)$ .

Another method for optical 3D sensing that does not rely on projective geometry is white light interferometry, which is used as the primary sensor in this dissertation and is introduced in the following subsection.

## 2.1.8 White Light Interferometry

Since the full derivation of white light interferometry is quite lengthy, this subsection only provides an overview of the technology. A full derivation of a sufficient model of white light interferometry as used in this dissertation can be found in Appendix A.

Light is an electromagnetic wave [Max65]. It can, therefore, exhibit interference phenomena if certain criteria are met. In order to observe static interference, a static phase relationship between the interfering waves is required. This is known as temporal coherence [LKW95, p. 31]. In terms of the

<sup>1</sup>Overviews can be found in [JH00, pp. 177], [Jäh12, pp. 253] and [BPF12, pp. 302].

<sup>2</sup>The speed of light is  $c = 1/\sqrt{\mu\epsilon}$ . Since the parameters  $\mu$  and  $\epsilon$  depend on  $\mu_r$  and  $\epsilon_r$  respectively, the actual speed of light depends on material properties. In air, both of these have numerical values of approximately 1 [TM15, p.778, p. 896], meaning that the speed of light in air is approximately  $c \approx 3 \cdot 10^8$  m/s.

light's electric fields, this can be expressed as follows in accordance with [Mes15, p. 534]:

$$E_1(t) = \text{const} \cdot E_2(t + \tau) \quad (2.1)$$

This is to say that if the electric fields  $E_1$  and  $E_2$  of two light beams differ only by a constant phase  $\tau$ , static interference is in principle possible. It should be noted that the time difference can both be negative, which would result in a delayed signal,<sup>1</sup> and positive, meaning the second E-field would be ahead of the first one. In practice, such a scenario is created by splitting and recombining a light beam. In the delay case, the path that the second wave has to travel would be longer, and in the lead case, it would be shorter than that of the first one. As is detailed in Appendix A, further requirements have to be considered in order to observe interference phenomena.

This basic idea can be used to build different measurement devices, one of which is a white light interferometer, as shown in Figure 2.8. The central component is a beam splitter, which divides the

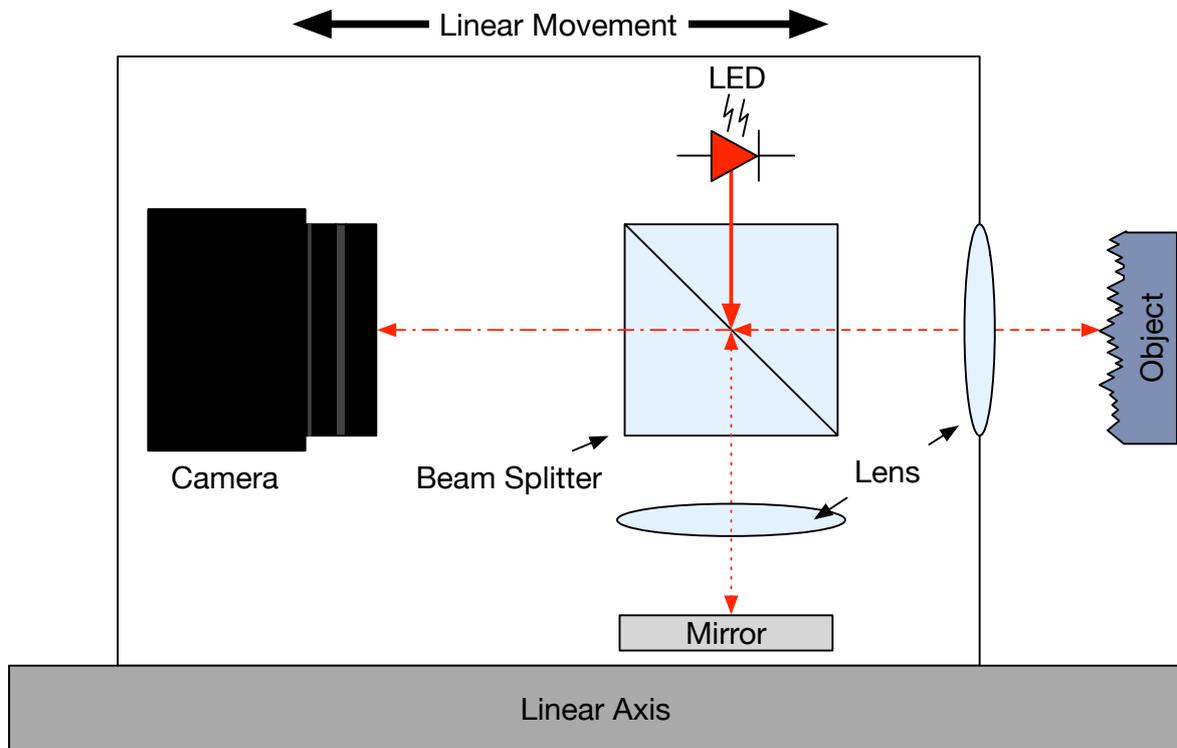


Figure 2.8: Principle of white light interferometry.

light beam emitted by an LED into a reference path (toward a mirror) and an object path (toward the object under test). The split beams of light are then recombined as they pass through the beam splitter

<sup>1</sup>Since time delays are typically written by subtracting the delay, this may be written as:

$$E_1(t) = \text{const} \cdot E_2(t - (-\tau)) = \text{const} \cdot E_2(t - \tilde{\tau})$$

once more and are then captured by a camera.

Since the light in both paths originates from the same source, it can interfere when recombined if the differences in path lengths are smaller than the so-called coherence length. Whereas the coherence length can be very long for light with very narrow bandwidths, it is in the micrometer range for white<sup>1</sup> light, such as emitted by an LED. This means that significant interference phenomena can only be observed if the path differences between the reference path and the object path are within the coherence length. Furthermore, the maximum interference signal will be observed when there is no path difference and will go down as soon as there is a difference in either direction.

The light source (LED), camera, beam splitter, mirror, and potentially lenses are arranged inside of a suitable chassis, which is then mounted onto a linear axis. The measurement system is arranged in such a way that the measurement light path illuminates the object under test and that the linear axis can move the sensor chassis toward or away from the object under test, as illustrated in Figure 2.8. This kind of white light interferometer is also known as a coherence radar [DHV92].

It can be shown for an LED with a Gaussian spectrum that the vertical sensor movement leads to an amplitude-modulated signal (correlogram) with a carrier frequency in the kilohertz range and a Gaussian envelope, as illustrated in Figure 2.9. This signal can either be sampled directly (while

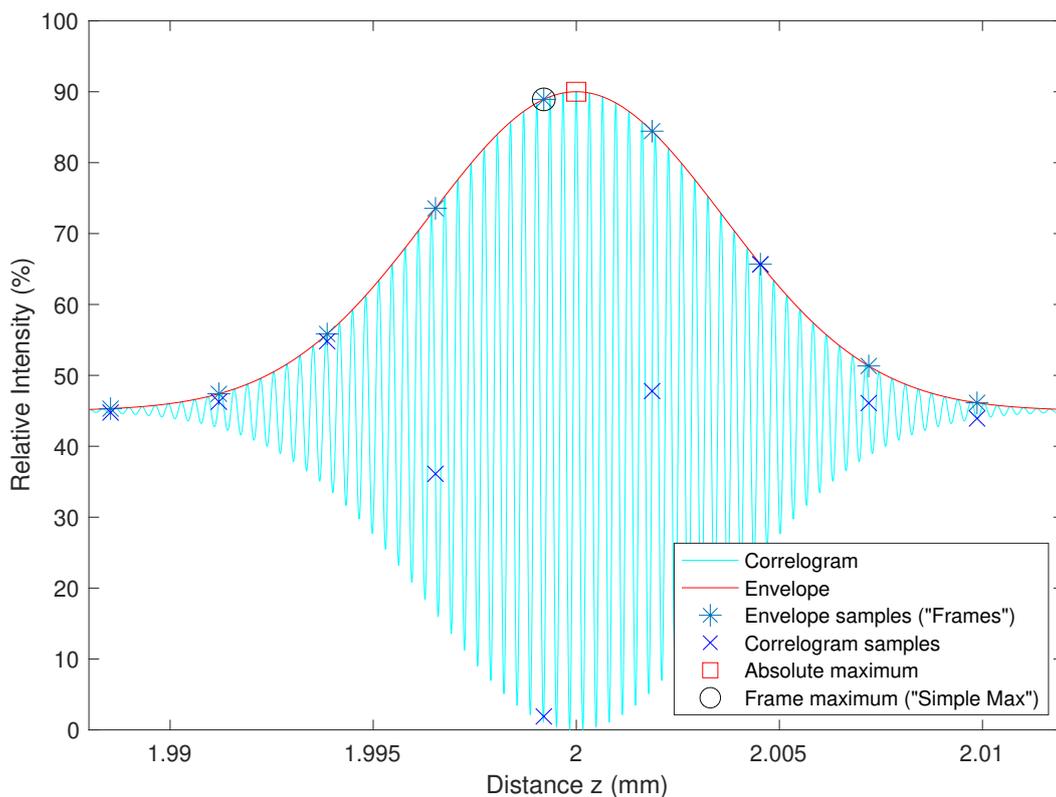


Figure 2.9: Correlogram, envelope, and sampled points.

ensuring the Shannon criterion is met), or it can be demodulated first, meaning that only the envelope

<sup>1</sup>The term “white” only refers to a relatively wide spectrum and not to the actual color of the light source.

has to be sampled. Since the signal was recorded while moving the sensor, the signal is plotted as intensity over position. Thus, the maximum of the Gaussian envelope relates to the distance from the object, at which point the difference of path lengths between the reference mirror and the object's surface in that point was zero. If this method is performed for an entire pixel array, a laterally sampled depth representation of the surface is obtained, where each lateral sample was reconstructed from its respective correlogram. Whereas the lateral resolution depends directly on the optical properties of the sensor as well as the pixel array and is typically in the micrometer range, the vertical resolution is in the order of magnitude of 100 nm for mirror-like surfaces and in the range of the surface roughness for other surfaces [DHV92, p. 920].

### 2.1.9 Summary of Inspection Technologies

As was demonstrated in the previous subsections, there is a multitude of different inspection technologies with very different attributes and physical principles. The presented sensor technologies are only a subset of the available non-destructive inspection (NDI) methods. However, the most commonly used ones in the aviation MRO industry were chosen. White light interferometry was also considered, even though it is not currently used in automated airplane part inspection, because it is the primary sensor technology investigated in this dissertation.

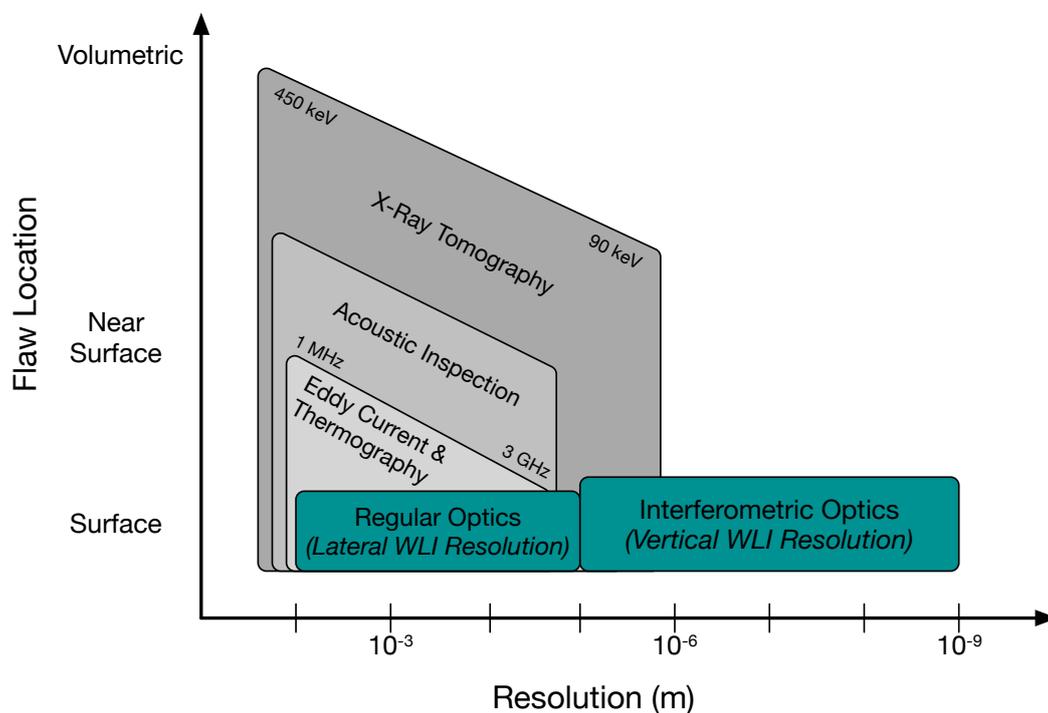


Figure 2.10: Comparison of different technologies and resolutions [adapted and translated from [GG06, p. 371]; additional information added for regular and interferometric optics with regard to WLI]. Note: WLI have limited surface penetration for certain materials.

Figure 2.10, adapted and translated from [GG06, p. 371], serves as a graphical summary of the

resolution of different inspection technologies.

Furthermore, Table 2.1 on the following page serves to complete the overview of the NDI methods that were presented in this section by summarizing different technological attributes, such as speed and resolution. Numeric values are provided whenever possible, but since the different technologies cannot always be compared directly, descriptive adjectives such as “very fast” are provided and potentially annotated with additional information. The table was compiled from similar tables in [GG06, p. 404], [Czi13, p. 47], and a table by L. Cartz as quoted in [Cam13, p. 12], as well as the author’s own research (white light interferometry section), as published in [ODH13], [Ott+14], and [DSO14]. A second table summarizing the pros and cons of the different technology in textual form is presented in Appendix B. In summary, interferometry has the highest possible resolution but is a mostly topological technology. Since the task at hand is the detection of very small topological defects (cracks), a sensor based on interferometry is the logical choice.

Table 2.1: Attributes of the discussed inspection technologies .

Technology	Inspection Type	Materials	Sensitivity / resolution	Speed	Access	Automation possible	Sizable	Locatable	Dependencies	Subjective
<i>Eddy current</i>	Surface / (penetrating)	Conductors	Resolution < 100 µm possible	Very fast (but limited due to lift-off)	Typically contact	Yes (mainly for simple geometries)	No	Lateral	Material (strong), geometry (strong)	No
<i>Magnetic particle</i>	Internal	Ferro-magnetic metals	Relatively low sensitivity	Preparation can be time-consuming; then almost immediate results	Direct access required	Difficult	(Lateral)	Lateral	Material (strong), geometry (low)	Partially
<i>Dye penetrant inspection</i>	Surface	Virtually any	High (µm-range)	Slow	Direct contact required	Difficult	Lateral, extra measurement required	Lateral	Surface condition	Partially
<i>Conventional radiography</i>	Penetrating	Several	up to 10 µm	Fast	Potentially no direct access required	Limited	Yes	No	Radiation absorption	Depends
<i>Tomography</i>	Penetrating	Several	Down to 3 µm	Relatively slow	Potentially no direct access required	N/A (enclosed measurement system)	Yes	Yes	Radiation absorption	No
<i>Acoustic inspection (transducer)</i>	Penetrating	Virtually any	100 µm range	Fast (but point-wise)	Contact	Yes (mainly for simple geometries)	Yes (comparison required)	Yes	Acoustic impedances	Depends
<i>Thermography</i>	Surface	Virtually any	µm-range possible	Relatively fast	Surface access required	Possible (heating potentially difficult)	Yes (lateral, if calibrated)	Yes (lateral, if calibrated)	Heat conductivity	No
<i>Camera</i>	Surface	Virtually any	µm-range possible	Very fast	Surface access required	Yes	Yes (lateral, if calibrated)	Yes (lateral, if calibrated)	Reflectivity	No
<i>Optical inspection (eyes)</i>	Surface	Virtually any	µm-range possible (with optical instruments)	Depends	Surface access required	No	With measurement equipment	Yes (lateral)	Reflectivity	Yes (person)
<i>Laser line triangulation</i>	Line (2D) / surface (3D, with actuator)	Many	µm-range possible	Very fast	Surface access required	Yes (lateral)	Yes (lateral)	Reflectivity	No	No
<i>White light interferometry</i>	Surface (3D) / penetrating (for certain materials)	Virtually any	nm-range possible (vertical), µm-range (lateral)	Moderately speeds possible	Surface access required	Difficult	Yes (lateral)	Yes (lateral)	Optical attributes	No

## 2.2 Existing Automated Crack Detection Systems

The previous section was mostly concerned with the inspection technologies themselves. In this section, examples of the use of such technologies for automated crack detection are given. It should be mentioned that for the specific part considered in this dissertation, the CFM-56 airplane turbine combustion chamber outer liner, only visual inspection (eyes and optical instruments) and fluorescent penetrant inspection (FPI) are inspection procedures that are certified in the manufacturer's repair manual. Therefore, those manual methods are the current state of the art used in practice, despite their disadvantages as listed in the previous section. Since FPI has the higher resolution/sensitivity of the two, it can be considered to be the current gold standard for the part's inspection. Since no automated solution for the specific part exists, other crack detection systems (even outside of aviation MRO use) are considered in this section in order to widen the scope of the overview.

### 2.2.1 Crack Detection Systems Using Cameras

Bu et al. discuss crack detection using cameras in a texture analysis-based approach to crack detection in bridges [Bu+15]. Their motivation for finding an automated alternative for crack detection in structures is very similar: manual inspection processes are subjective, time-consuming, and costly because highly trained inspectors are required [ibidem, pp. 41]. They used RGB cameras setup to allow monitoring a crack width of less than 0.1 mm. In a first step, it is determined whether an image is "complex" or "simple" [ibidem, pp. 42], based on RGB channel analysis. In the case of images deemed complex, pre-processing techniques are applied. Otherwise, the images are subjected to the crack detection algorithm directly. In comparison, since the combustion chamber liners analyzed in this dissertation tend to have been pre-processed in order to remove their thermal coating, the surfaces tend to be relatively complex/noisy. Thus, most of the images that were analyzed in the course of the research conducted for this dissertation were complex in nature. Therefore, a differentiation step, as proposed by Bu et al., can be eliminated.

Bu et al. then propose the use of wavelet features to extract features from sliding window regions in the overall image [ibidem, pp. 45]. The images are then fed into a support vector machine to classify between cracks and non-cracks [ibidem, pp. 46]. The classification approach presented in this dissertation, on the other hand, relies on known crack attributes instead of using a machine learning approach.

Similarly, Zhang et al. use cameras to detect cracks in subway tunnels [Zha+14]. They identified similar issues with manual inspection processes, as mentioned previously in this dissertation [ibidem, p. 19308]. Unlike Bu et al., who use regular 2D array cameras, Zhang et al. use several line cameras mounted to a subway vehicle [ibidem, p. 19311]. The sensor array also includes a velocity sensor, which allows combining the captured lines to images when the train is moving. Since the cameras are synchronized, their images can then be combined into an overall 2D image of the captured scene [ibidem, p. 19311]. Those images are then processed using the morphological black top hat trans-

formation<sup>1</sup> [Zha+14, p. 19313], followed by thresholding. After further filtering, misidentified cracks are removed under consideration of shape, number of pixels, and average gray levels. This approach has some similarities to one of the verification approaches described in section 5.7.1. However, one major addition in this dissertation is the distance and orientation to weak spots in the material. An additional difference is that, because a white light interferometer produces 3D data, the parameters for configuring the system can be set in metric units and are therefore easier to deduce from (existing) real-world experience. Finally, experience has shown that the shape by itself is not necessarily a good indicator for whether a finding is a crack because, in an airplane combustion chamber liner, several cracks might run together, or parts of the metal could break out. Rather than the shape, factors such as length, location, and, to a degree, variability have been found to be more useful indicators for a crack, as opposed to, for instance, a scratch mark. Like Bu et al., Zhang et al. investigated following the crack candidate extraction step with a machine learning classifier. They also took a thresholding classifier into account and found the two approaches to be similar in crack retention performance, but the learning approach to be better at rejecting false positives [ibidem, p. 19324].

Dorafshan et al. released a Utah State University tech report regarding the detection of cracks in concrete structures using image processing and classical cameras [DMQ16]. Citing time, cost, and safety concerns as motivation for finding an automated approach to crack detection [ibidem, p. 8], they propose to equip unmanned aerial vehicles (drones) with cameras and assistive sensors for navigation [ibidem, p. 8]. The captured RGB images are then processed with a method involving (among other steps) median filtering, Sobel-based edge detection, Otsu's method for obtaining an optimal threshold, the use of the morphological majority operator,<sup>2</sup> and several cleaning steps [ibidem, p. 33].

Liu et al. propose a camera-based system for micro-crack detection in plastics [Liu+16]. Since the use of simple thresholding is difficult due to uneven backgrounds, they first calculate and remove the image gradient followed by the calculation of an optimal threshold based on circularity [ibidem, pp. 283] because they found the direct application of Otsu's method unsuitable [ibidem, pp. 282]. The reasoning is that, according to Liu et al., a crack is not circular, and thus an ideal threshold would minimize the presence of circular components [ibidem, pp. 282]. The idea of removing the background by calculating the gradient is related to the method of using a 2D polynomial regression to remove the background shape of the white light interferometer images, as described by the author in [ODH13] and this dissertation. However, the polynomial method is able to remove higher-order background shapes than a simple gradient.

It was shown in this subsection that there are several existing crack detection systems based on cameras. However, there are two significant downsides to camera-based systems: first, the mentioned methods only deliver two-dimensional data. However, in the context of MRO, it is very valuable to have access to high-quality three-dimensional data because it allows for an automated repair chain, as published in a paper that was co-written by the author [Sch+16]. Second, the use of intensity-based cameras means that the system is sensitive to discolorations. Since, for instance, the CFM-56 has a

<sup>1</sup> $G_{BTH} = G \bullet M - G$  in the notation used in this dissertation, see sub-section 5.6.3.

<sup>2</sup>The majority operator delivers true if the majority in its neighborhood is true.

coating that is removed before the inspection process that might leave the part partially discolored, this downside is significant. Therefore, it can be concluded that camera-based systems are unsuitable to fulfill the task of inspecting complex three-dimensional goods.

### 2.2.2 Crack Detection Systems Using Eddy Currents

According to García-Martín et al., the eddy current method is used extensively for crack detection, including the aeronautical, nuclear, and metallurgical sectors [GGV11, p. 2558]. Furthermore, Diaz mentions several additional industries that use eddy current testing for NDI, including the automotive sector [DUP07, p. 3.1]. The most obvious limitation of the technique is that it only works with electrically conductive materials. Traditionally used for metallurgical inspection, it can also be used for less conductive materials, such as carbon fibers [HS11].

The most significant advantage of eddy current testing is that contact-less measurements can be made with relatively high speeds [GGV11, p. 2526] if the requirements for successful eddy current inspection (see section 2.1.1) are met. Additionally, the sensors permit the detection of subsurface errors in certain cases. Eddy current testing delivers instantaneous results and is relatively sensitive to small discontinuities<sup>1</sup> [Uni01, ch. 5, p. 7]. Since the technique requires minimal part preparation, eddy current is suited for in-the-field testing of localized areas, according to [ibidem, ch. 5, pp. 27]. Furthermore, since eddy current is an established technique and since it can be readily automated for simple geometries, such solutions are available on the market, for instance [Foe16] [ibg16].

An important consideration is sudden changes of the eddy current signal due to edges. This is because the eddy currents have a smaller effective area of flow in those cases, which leads to a weakening of the secondary magnetic field and thus affects both the resistance and the reactance of the coils' complex impedances, in the same way, a defect would [DUP07, p. 3.4]. Those effects need to be compensated, which gets more complicated with more complex parts, such as the CFM-56 outer liner, which has many edges and cooling holes. What makes matters even worse is the fact that the procedure described in this dissertation is concerned with the inspection of *used* parts that are pre-processed in different ways and may have been repaired and thus altered before. Thus, there might be additional purposefully placed edges/discontinuities that could potentially be difficult to compensate for. Additionally, [LWL00, p. 16] mentions that calibration reference standards must be close to the actual parts in all relevant aspects, including geometry, coating, and surface finish. This might be very difficult to guarantee in a repair shop scenario. It, therefore, follows that the existing eddy current inspection machines are currently unsuitable for the inspection of complex goods such as a CFM-56 outer liner.

---

<sup>1</sup>Especially in a differential configuration.

### 2.2.3 Crack Detection Systems Using Thermography

Whereas thermographic cameras themselves can be integrated relatively easily into an automated process, the challenge is to introduce heat energy into the test specimen. Štarman et al. introduced a system that solves the heating problem with an electrical heating coil, which induces eddy currents in the observed steel blocks while they pass through the inspection system [Šć08, p. 4]. The heated areas are then observed with four infrared cameras [ibidem, p. 3], which allows for the detection of discontinuities of less than 0.3 mm [ibidem, p. 5]. The defect detection procedure presented by Štarman et al. relies on amplitude thresholding [ibidem, p. 5] since defective areas will dissipate heat differently from flawless ones, as discussed in section 2.1.6.

Broberg et al., on the other hand, investigated the use of heat radiation sources for material heating for thermography-based crack detection. They investigated the use of a pulsed laser and a flash lamp [BR12, p. 2]. They found that with open cracks an additional effect takes part in increasing the crack area to have a higher temperature than its surroundings: according to Broberg et al., multiple reflections inside the crack leads to longer exposure to the energy and thus to an increased absorption rate [ibidem, pp. 1]. Additionally, they discovered that lasers are preferable over flash lamps because they offer more control over the heating process [ibidem, p. 5]. They claim that crack sizes down to 5-10  $\mu\text{m}$  could be detected [ibidem, p. 5]. It should be noted, however, that thermography is based on a defect's ability to change the heat distribution of its surroundings, to enlarge its area optically (in the IR range). That means that even though it may be possible to detect a damage, its dimensions might not necessarily be accurately measurable.

Maillard et al. considered both active thermography, as shown in the previous examples, and passive thermography [Mai+10]. In the latter case, an infrared camera is used in much the same way as a regular camera but circumvents the lightning issues associated with conventional cameras, particularly in stamped parts that might be covered with a residue film [ibidem, pp. 2]. When analyzing active thermography, Maillard et al. tested all three techniques that were previously mentioned (induction, laser, and flash lamp heating [ibidem, pp. 6]). Maillard et al. observed that induction heating allowed finer cracks to be detected, even when excitation with external radiation did not allow for the detection of those cracks [ibidem, pp. 7]. In a later article, Maillard et al. confirmed their findings, stating that under certain conditions, active thermography can yield similar results to manual inspection methods such as magnetic particle inspection with good repeatability [Mai+12, pp. 1986].

Unlike classical cameras, active thermography cannot easily be fooled by simple markings on the observed surfaces. However, as is the case with mono cameras, the technology does have the downside of not delivering three-dimensional data. Furthermore, as was mentioned before, thermography does not necessarily allow for accurate sizing of the defective areas because the extent might be exaggerated. It is, therefore, difficult to fulfill the set constraints for airplane combustion chamber liner inspection using thermography.

## 2.2.4 Crack Detection Systems Using Radiography

Asadizanjani explores the applicability of confocal laser microscopy and x-ray computed tomography in his dissertation [Asa14]. He concludes that confocal laser microscopy is difficult to apply with porous objects, so he proposes to repeatedly polish and scan the surface to obtain a layer-by-layer volumetric image [ibidem, p. 14-15 and p. 21-24]. Since the method is destructive (due to the polishing), it is not suitable for use in the MRO industry and is thus not further considered here. X-ray computed tomography, on the other hand, generates volumetric images from 2D slices in a non-destructive fashion [ibidem, p. 15]. Although this method has the advantage of allowing volumetric scans, and, therefore, allows the potential detection of sub-surface cracks, Asadizanjani found that it may be difficult to capture very small pores and cracks with x-ray tomography [ibidem, p. 19]. He thus proposes to combine x-ray tomography with confocal microscopy. Additionally, he states that the quality of x-ray tomography strongly depends on optimum parameters, which depend on the specific composition of the object under test. Due to these limitations, it can thus be concluded that x-ray tomography might be difficult to use in the context of MRO, where the precise object attributes may not be fully documented or may even have changed while in service.

Wang et al. developed a crack detection system based on digital x-ray radiography [Wan+11]. Since the images are two-dimensional digital projections, classical methods of digital image processing can be applied. However, Wang et al., like Asadizanjani, mentions that the radiographic parameters have to be tuned carefully in order to provide good image quality. They solved this issue with a step wedge-based calibration [ibidem, pp. 107]. The captured 2D radiographic images are then processed in order to detect cracks. The processing steps include morphological filters, image sharpening, and edge detection using the Canny edge detector [ibidem, p. 109–123]. This has certain general similarities to the approach for crack detection presented in this dissertation, simply because of the nature of cracks, as will be explained later. However, the algorithm detailed later in this dissertation involves additional steps, and the white light interferometer data has to be pre- and post-processed because it is three-dimensional in nature. Wang et al. were able to detect cracks with a length of down to 2 mm.

Xu et al. researched the use of x-ray radiography for the inspection of fastener hole cracks in airplane wings [Xu+10]. They found other methods to be incapable of detecting hidden defects without direct access<sup>1</sup> [ibidem, p. 2]. Therefore, they developed a specialized robot that allowed them to guide a digital x-ray system along an airplane wing to perform rasterized scanning of an aircraft wing [ibidem, pp. 2]. The path for the specialized kinematics system is derived from computer-aided design (CAD) data (Xu et al. do not specifically mention in their application if this process is fully automated) [ibidem, pp. 2]. They describe a data processing scheme that revolves around the detection of fastener holes, followed by the detection of cracks in those areas [ibidem, p. 3]. The detection of fastener holes is done in two steps: first, a rough detection of hole blobs is performed using a global threshold, either through manual selection or by using Otsu's method. This is done for several images, and the results are stored in a learning system [ibidem, p. 4]. The fastener holes are then detected with

---

<sup>1</sup>This is in line with Table 2.1.

knowledge, such as fastener hole sizes and maximum expected holes per area [Xu+10, p. 4]. Next, local thresholds in the fastener areas are determined using Otsu's method [ibidem, pp. 4]. The binarized images are then used to find the precise fastener hole locations [ibidem, p. 5]. Finally, the surrounding cracks are detected with sub-pixel accuracy [ibidem, pp. 5].

The work presented by Xu et al. shows certain similarities to the crack detection system developed for this dissertation. There are, however, several important differences. First of all, and most obviously, the employed sensor technologies are completely different. White light interferometry is a highly accurate topical 3D sensor, which is very sensitive to movement during the measurement and thus very challenging to integrate into an automated process. The x-ray sensor used by Xu et al., on the other hand, is a 2D sensor and offers a lower resolution than interferometry but is not limited to surface inspections. Thus, the integration of the sensor into an automated process, the capturing and processing of the data and the automated handling of the sensor are not comparable. Furthermore, the approach presented by Xu et al. is limited to certain areas (surroundings of fastener holes), whereas the work presented in this dissertation inspects the entire accessible surface. The locations of weak spots, such as cooling holes and edges, are also used by the method introduced in this dissertation, but only to increase the detection rate and as one of several criteria to confirm whether a finding is indeed a crack.

In his dissertation, O. Alekseychuck describes the detection of cracks in radiographic measurements. After he found classical local approaches such as gradient filters, Laplacian of Gaussian, Canny filter, and profile fitting to be dissatisfactory for the detection of cracks in radiographic images [Ale16, p. 21-27], due to their low signal to noise ratio (SNR) [ibidem, p. 11, p. 18], he investigates the use of global approaches, which is to say the area of a crack is regarded, instead of its edges [ibidem, p. 20]. Global approaches include histogram-based methods (e.g., Otsu's method), parametric search algorithms (e.g., the Hough transform), and the heuristic search of a directed graph containing all possible crack curves [ibidem, p. 27-44]. Since, according to Alekseychuck, the use of heuristics does not guarantee an optimal solution [ibidem, p. 39], he developed a method that yields the exact solution. However, his algorithm is too computationally expensive for interactive processing [ibidem, p. 13].

Even though radiography has the advantage of being able to find hidden defects, it has several limitations. First and foremost, the currently existing crack detection systems simply do not have the resolution required for airplane combustion chamber liner inspection. Second, the use of radiation requires protective measures that might make flexible automation solutions difficult. And third, it might be difficult to find the correct parameters for different parts [Wan+11, pp. 107]. Thus, at least with the current state of the art, it would be quite challenging to perform a high-resolution crack inspection on airplane combustion chamber liners under the constraints set in this dissertation.

## 2.2.5 Crack Detection Systems Using White Light Interferometry

Even though white light interferometry is an established technology, its automated industrial uses are incredibly limited. This is due to several downsides inherent to the technology, which are detailed in

section 3.1. For instance, with the current computer technology's processing power, one issue is the sheer amount of vertical slices that have to be captured and processed. For example, an interferometer that was originally considered in the early phases of this research was developed by German sensor manufacturer "3D-Shape." The datasheet of their sensor "Korad3D" states that it has a scanning speed of 90.8  $\mu\text{m/s}$ , resulting in a scanning time of about 33 seconds for a depth range of 3 mm. This is far too slow for the constraints mentioned in the introduction.

An extensive literature review uncovered no solutions that are similar to the robot-guided white light interferometry crack detection system developed for and documented in this dissertation. However, since white light interferometry has such an extraordinary depth resolution, it was actively researched as a sensor technology for the inspection of large parts, not only in this dissertation but also by a *Sonderforschungsbereich* in Hannover, Germany (collaborative research center SFB871) as part of a multi-modal inspection system for aviation parts. Their research on a robot-guided white light interferometer, as published in 2015 [LKR15b], has certain similarities to the research presented in this dissertation and was first published by the author in 2013 [ODH13]. Aside from the fact that the interferometer is guided by a manipulator, these similarities include the two-step approach for image stitching (rough transformation using the forward transform of the robot, followed by a fine registration algorithm). However, the specific implementations are different. Furthermore, their research includes finding feasible solutions for the scanning speed issues associated with white light interferometry and found a solution in massively parallelizing the extraction of correlogram maxima on GPGPUs [LKR15a]. The author of this dissertation opted to use an industrial interferometer that uses hardware-based IQ demodulation instead.<sup>1</sup>

## 2.3 Summary, Deficits, and Contributions

There are many different sensor technologies that have wide-spread use in the MRO industry. In the previous section, the technologies eddy current testing, magnetic particle inspection, radiographic methods, acoustic inspection methods, thermography, and optical inspection were mentioned. Furthermore, several automated crack detection schemes using some of those technologies exist for different applications.

Especially in more conservative markets, such as the aviation industry, it is difficult to introduce new technologies unless they are proven to work as well or better than existing technologies. A new method will have to go through a certification process and may then be included in the manufacturers' repair manual. Many of the traditionally used technologies are performed manually by qualified inspectors. This not only possibly introduces additional costs but also the human factor. As was detailed, this includes positive aspects, such as intuition and unsurpassed flexibility, but can also have negative implications: distractibility, subjectivity, and time-varying performance, among others. Automation rids inspection processes of the negative aspects of the human factor. Unfortunately, many

---

<sup>1</sup>The author is contractually obligated not to reveal the sensor used in this dissertation. IQ stands for "in-phase and quadrature."

of the established processes are very difficult, or even impossible to be automated economically. For instance, the repair manual of the CFM-56 combustion chamber liner allows the use of FPI, which cannot be automated very easily. Thus, new inspection methods have to be developed, despite the need for certification before official use. Since fluorescent penetrant inspection *potentially* has a very high resolution (see Table 2.1), a new technology that is to compete with FPI has to have a resolution in the same order of magnitude. A literature review revealed that even though there are several approaches to industrial crack detection in both aviation and other markets, there appears to be no solution for automated very high-resolution crack detection of large objects that results in three-dimensional data.

Interferometric optics offer an outstanding vertical and, depending on the chosen optical components, a very high lateral resolution. Despite the technology's apparent potential, there is no working automated inspection solution available either in the scientific community or on the market today. It can thus justifiably be concluded that white light interferometry for the automated inspection is a very relevant technology, both from the point of views of the scientific community (due to its high degree of novelty and relevance) and from the point of view of the industry, due to its unique positive attributes. This dissertation bridges the gap between the established technology of white light interferometry and its previously unachieved application in the automated and highly flexible inspection of large objects. In order to achieve this goal, a suitable hardware and software architecture were designed and implemented, including an interferometer driver, suitable data structures, a communication protocol, and an error handling strategy. Next, several steps were taken to use white light interferometry in an automated setting, including automatic parameterization, highly parallelized time-optimal scanning, and white light interferometer data pre-processing. After collecting large amounts of data, the overall data set was split into appropriate subsets. These subsets were then registered using a custom and highly tuned non-linear iterative closest points (ICP) approach and then combined into larger point clouds while ensuring optimal use of the overlapping data areas. Next, 2.5D images are generated from these combined point clouds after removing the background using a modified RANSAC-based polynomial regression approach. The parts also include seemingly defective areas, cooling holes and edges, which are detected using a multi-modal approach. Because those special features are relatively large, they are first roughly detected in a fast and lower resolution laser line section scan. Afterward, their precise location is determined using the white light interferometer data. The crack detection is performed as pure image processing using the generated 2.5D images and extracted feature locations. Finally, the findings are verified in the image domain, but also using a sectioning (crack profile) approach in 3D space.

Lufthansa Technik AG has performed a probability of detection (POD) study (see section 6.2), deeming the technology appropriate for use. At the time of this writing, the novel method was used at Lufthansa Technik AG in Hamburg in parallel to the certified FPI method. In order to guarantee that the system is working appropriately, a daily self-check method was developed and implemented.

## **Chapter 3**

# **A Novel Approach for Automated Crack Detection**

It was stated in the previous chapters that a new crack inspection method that aims to replace the currently used method of FPI has to have a very high resolution. A crack is a volumetric type of damage. Therefore, an optical sensor has to have both a high lateral and vertical resolution. A white light interferometer in the form of a coherence radar scans an area with a lateral resolution that is determined by the optical attributes of the sensor and the number of pixels. This resolution can be very high.

The depth resolution, on the other hand, is largely independent of the optical components. It was mentioned in the previous chapter that a predictable waveform could be recorded for every valid pixel when a surface scan is performed: a correlogram. The coherence length is relatively short for white light (a typical coherence length would be 5  $\mu\text{m}$ ). Additionally, the maximum of the correlogram can be estimated with subframe accuracy so that a vertical resolution in the sub-micrometer range can be achieved. Furthermore, a white light interferometer's camera can be parallel to the surface that it is scanning, meaning that there are no shading effects in that case. Therefore, white light interferometry is a very high-resolution, three-dimensional, optical scanning method and thus is an ideal candidate for optical crack detection.

### **3.1 Sensor Selection**

White light interferometry has several issues that make integrating it into an automated process quite challenging, some of which are interdependent. In the following subsections, the most important issues are identified, and afterward, desirable sensor attributes are derived.

### 3.1.1 Sensor Issue: Sensitivity to Movement

First, interferometry is very sensitive to relative movement between the sensor and the object that is being scanned. Lateral movement causes a relative shift between the scanning pixel array and the surface, which means that signal components of neighboring pixels are mixed at different points in time. Figure 3.1a shows two sample light rays (blue and red in the illustration) from the surface onto the pixel area, illuminating a pixel each. This means that each pixel will sense information coming from a specific part of the object. Figure 3.1b shows what happens when relative lateral movement occurs. In the given example, the red ray illuminates the lower pixel, and the blue ray illuminates a new pixel underneath the lower one. Since rough surfaces create speckles and since speckles have random phases, this means that the mixed signal will no longer resemble the expected wave shape. The resulting

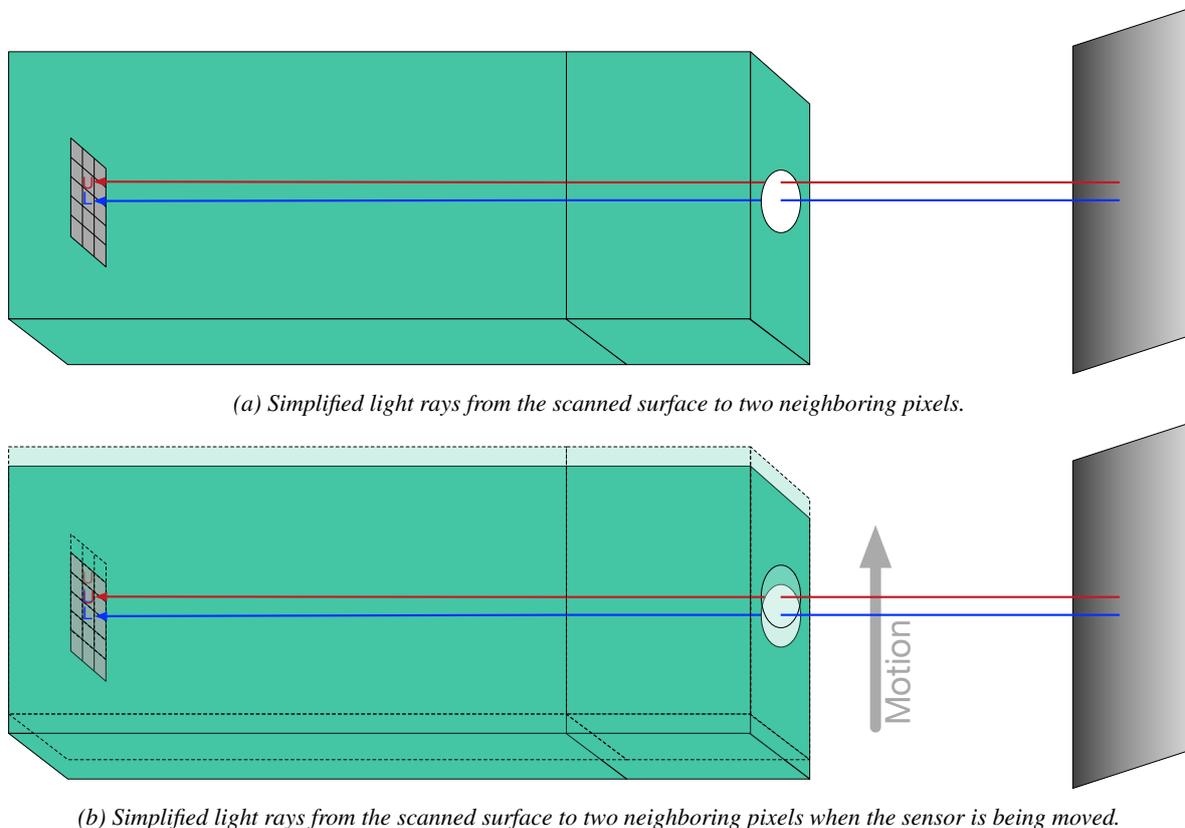


Figure 3.1: Effect of lateral movement on illuminated pixel locations.

signal for the lower pixel is illustrated in Figure 3.2. The beginning of the graph (solid blue) shows the signal coming from the original sensor position. At the vertical dashed black line, a simulated lateral sensor-shift occurred, which leads to the red component due to a different phase relationship. The dotted blue line shows the progression of the original signal if there had not been any lateral movement. It can be seen in Figure 3.2 that lateral movement will lead to incorrect positional readings because the intensity maximum is not found at the correct position. Additionally, lateral movement typically leads to lower intensities because neither one of the two mixed functions necessarily contributes its

respective maximum value, which is also illustrated in Figure 3.2.

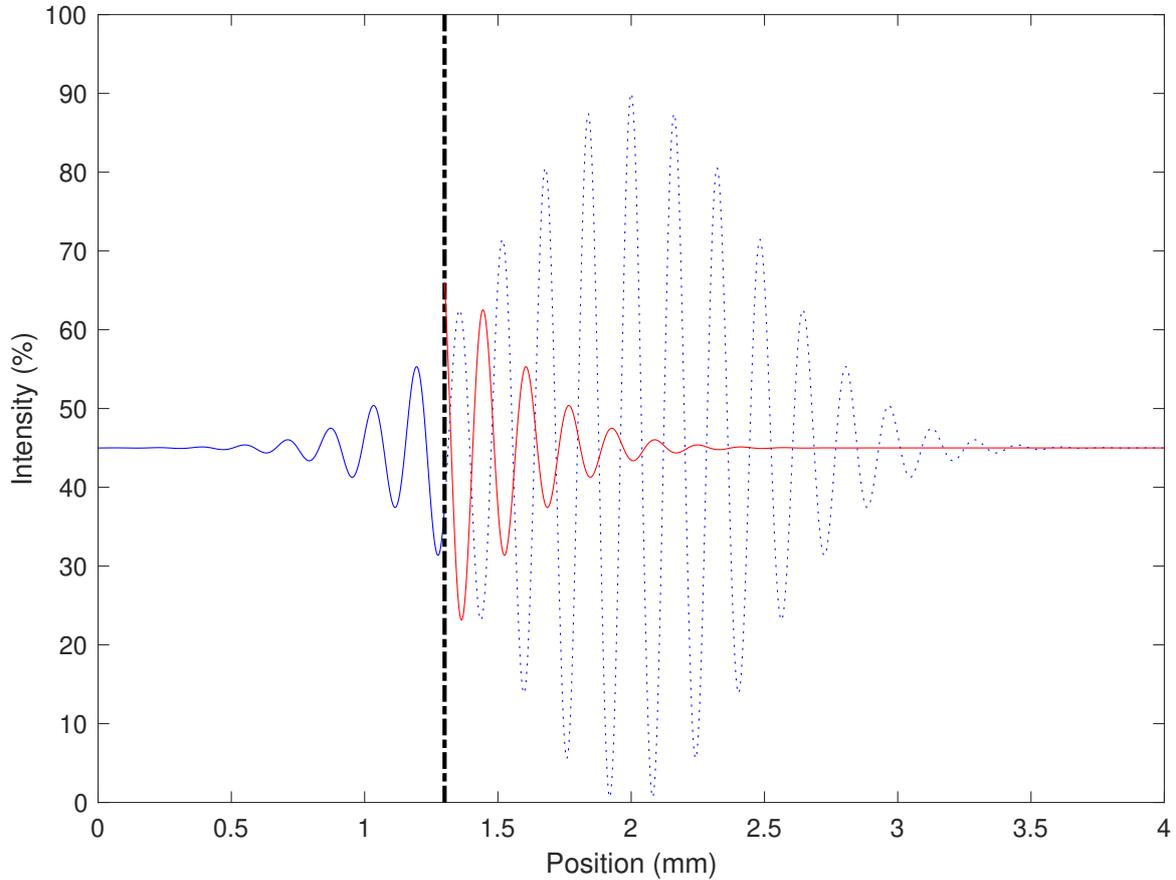


Figure 3.2: Example of the effect of lateral movement on the signal shape. Blue: original signal. Dashed black: the moment where the lateral shift occurred. Red: changed signal continuation due to different phase relationships. Dotted blue: expected signal (no lateral movement).

A different error source is vertical movement between the sensor and the part. The idea of a coherence radar is to scan the surface area in a continuous motion toward or away from the surface. In other words, the position is assumed to be monotonously increasing or decreasing over time. In the case of vertical vibration, that is no longer the case because there is an additional movement overlapping the scanning motion. Even if the signal was in no other way affected, this would lead to incorrect mappings of intensities to positions. More severely, though, a vertical change in movement means a *time-varying* velocity with respect to the measured object, which leads to the demodulation frequency<sup>1</sup>  $f_D$  to be time-varying. This means in effect that the signal no longer has the expected frequency, which means that the IQ-demodulation approach<sup>2</sup> will fail. In other words, vertical motion will lead to signal degeneration to the point of failure. This is illustrated in Figures 3.3a and 3.3b for a change in velocity from 20 to 1 mm/s, for instance, due to an incorrectly timed backward movement

<sup>1</sup>For more information on the resulting demodulation frequency, see equation A.90 in Appendix A.

<sup>2</sup>See Figure A.12 in Appendix A.

of the robot in the time and frequency domain, respectively. It can be seen that a change in effective velocity leads to a change in the modulation frequency, which leads to demodulation issues. Thus, the signal quality is strongly degraded. Even if the maximum could still be detected, the estimated position would not be the true position. If an external motion of the sensor should not change the mean velocity but rather add a vertical vibrational component, this will lead to an additional modulation of the signal, creating side maxima. Figures 3.3c and 3.3d show a combination of a velocity reduction from 20 mm/s to 5 mm/s and an overlaid 1 kHz vertical vibration in the time and frequency domain, respectively.

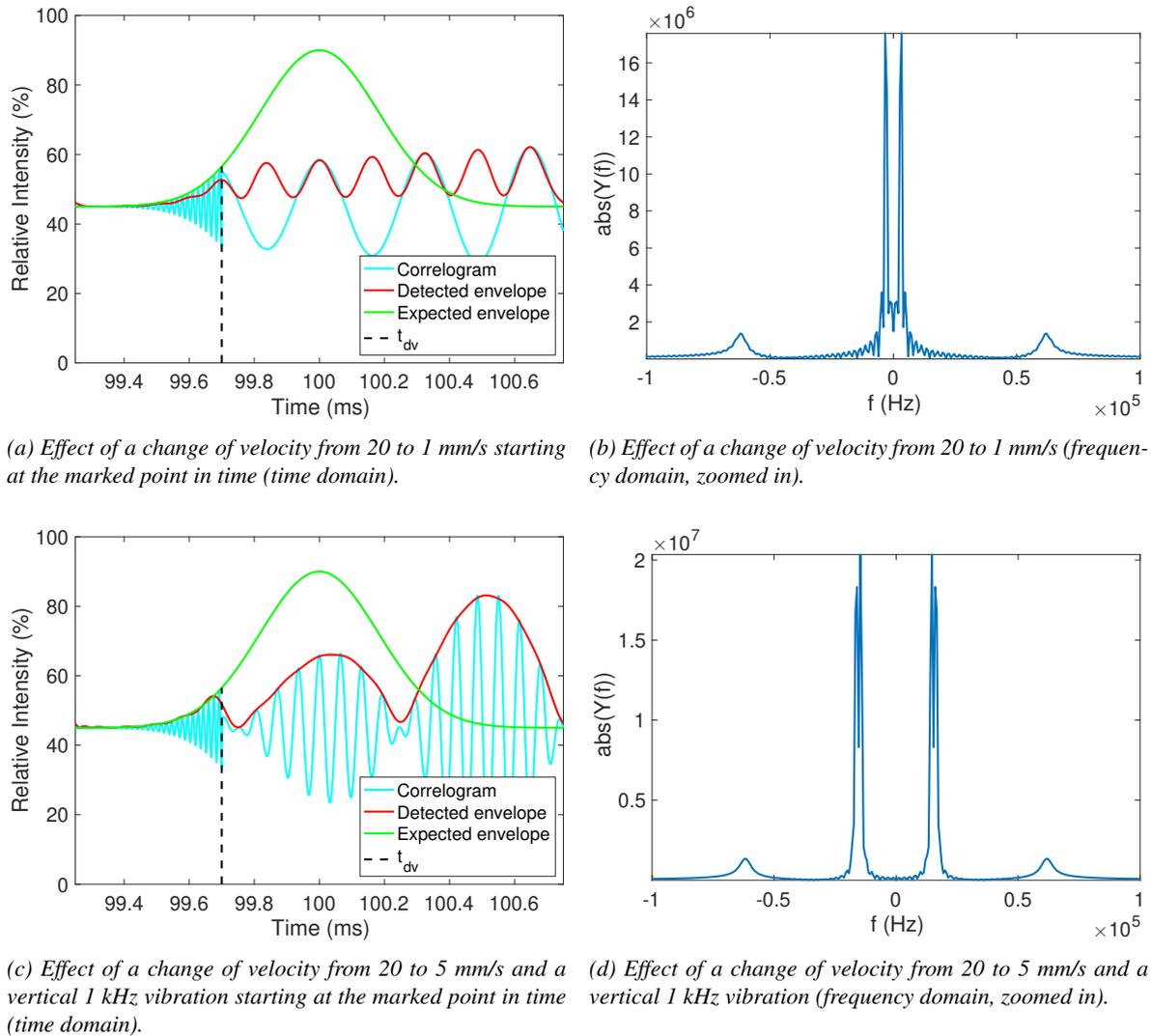


Figure 3.3: Effect of vertical movement on the output signal of a single pixel.

Finally, lateral movement can lead to another, more subtle issue if the sensor is not perfectly orthogonal to the surface. In that case, a movement with a velocity vector  $\underline{v}$  in the lateral direction also contains a velocity component in the lateral direction, depending on the angle. If that component is large enough, it could lead to similar effects as the ones associated with direct vertical movement.

It should be noted that this effect can be used to perform interferometric measurements without the need for a stitching movement. This is known as laterally scanning interferometry (LSI), see section 7.2 in the outlook.

### 3.1.2 Sensor Issue: Amount of Data

The most straightforward way to perform a vertical interferometry scan is to capture images at a data rate of at least  $f_S = 2 \cdot f_D$ , in order to fulfill the Shannon criterion. For instance, if the demodulation frequency was 40 kHz, the sampling frequency would have to be at least  $f_S = 80$  kHz. Even if the camera only had  $100 \times 100$  pixels with 16 bits resolution, this would result in 20000 bytes per image. This means that data would have to be captured at a rate of  $20000 \text{ bytes} \times 80000 \text{ Hz} \approx 1.49 \text{ GB/s}$ . If the sensor moves at  $10 \text{ mm/s}$  and a measurement depth of  $2 \text{ mm}$  has to be captured, the resulting active measurement time is  $\frac{2 \text{ mm}}{10 \text{ mm/s}} = 200 \text{ ms}$ . Thus, a single  $100 \times 100 \text{ pixels} \times 2 \text{ mm}$  depth scan would have a size of about 305 MB. From these samples, the full signal can be recovered through ideal interpolation. Ideal interpolation is most easily understood in the frequency domain. In a nutshell, sampling causes the spectrum to repeat itself centered on multiples of the sampling frequency. The copies of the spectrum are *continuous*, even though the time domain representation is not. This is why the Shannon criterion demands that the sampling frequency is twice the maximum signal frequency because otherwise, the left side of the two-sided spectrum centered around the sampling frequency would overlap with the original spectrum. This is known as aliasing and makes correct signal recovery impossible. With that in mind, it becomes clear that multiplying the spectrum with a box function with a width of the sampling frequency removes the spectral copies, effectively turning the signal back into a continuous one. In this case, it is *not* required to capture the entire signal to extract the relevant information. Rather, it is sufficient to recover the envelope, which means that significantly fewer samples are required (see, for instance, Figure A.14 in Appendix A). However, such an approach would necessitate that the signal is demodulated *before* the samples are captured. If, for instance, samples are taken every  $10 \mu\text{m}$  over a measurement range of  $2 \text{ mm}$ , this results in only  $\frac{2 \text{ mm}}{10 \mu\text{m/slice}} = 200$  slices. Thus, for the same measurement volume of  $100 \times 100 \text{ pixels} \times 2 \text{ mm}$  as in the previous example, the resulting amount of data would only be  $100 \times 100 \times 2 \text{ bytes} \times 200 = 3.81 \text{ MB}$ . Of course, in practice, larger pixel arrays are used, resulting in larger amounts of data.

### 3.1.3 Sensor Issue: Dynamic Range

An issue that all optical sensors have to deal with is dynamic range. As long as the captured object sends out light at brightness levels within the dynamic range of a given camera's pixels, capturing the object without non-linear effects is possible. It should be noted that the absolute intensity values can be accommodated by increasing or decreasing the integration time and/or the use of optical filters. Of course, this approach has limitations: there is a minimum integration time, long integration times lead to pronounced time averaging, and passive filters can only reduce, not amplify, the captured light's intensity.

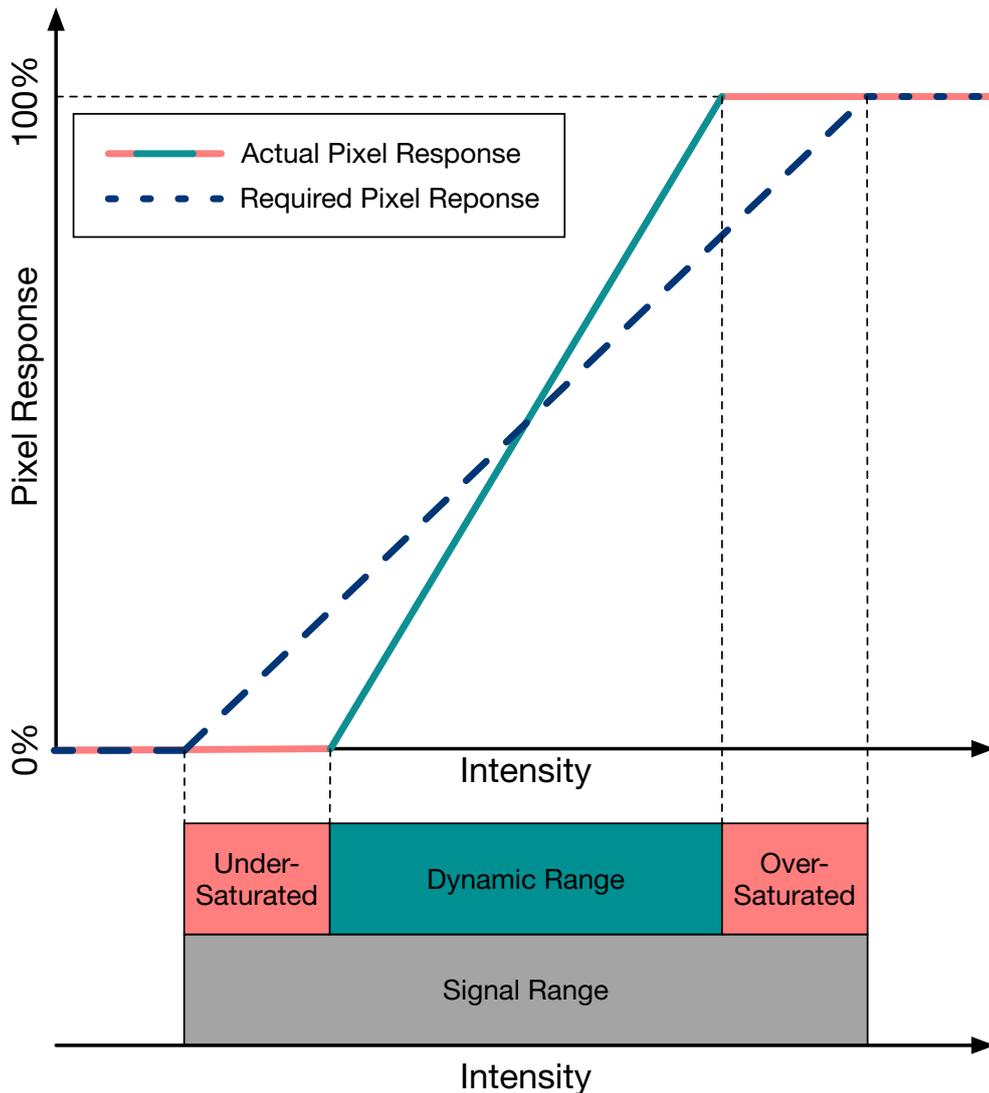


Figure 3.4: Illustration of over- and under-saturation.

It is impossible to build a pixel that has an ideal pixel response, which would mean that it could accurately detect anything from a single photon up to an arbitrarily large intensity. In classical image capturing, this means that the dynamic range has to be chosen in such a way that it covers the required range of intensities (see dashed blue line in Figure 3.4). Otherwise, even if a camera is set up correctly to be centered within a given signal's intensity range, as soon as the camera's dynamic range is exceeded, affected pixels will either sense no signal or saturate (see red lines in Figure 3.4). This non-linear behavior means that affected areas cannot be captured accurately. In the case of white light interferometry, this means that no depth information can be derived from over- or under-saturated pixels. Since the purpose of a white light interferometer is to perform a highly accurate depth scan and not merely to capture an intensity signal, and since the relevant information is in the envelope and *not* the absolute height of the intensity level, the DC offset can be fully removed without losing any

information, and, in fact, has to be removed in the processing scheme anyway.

Since the maximum of the signals's Gaussian term<sup>1</sup>  $G(\cdot)$  is normalized to one ( $G(\mu_0) = 1$ ) and since the modulation term  $\cos(x)$  also has a maximum of one ( $\cos(2\pi \cdot z) = 1$  with  $z \in \mathbb{Z}$ ), the maximum only depends on the reflectivities and  $I_0$ :

$$\max(I_D(\tau)) = [I_0 \cdot (s_r^2 + s_o^2)] + [I_0 \cdot 2s_r \cdot s_o] \quad (3.1)$$

In equation 3.1, the reflectivities  $s_r$  and  $s_o$  are for the reference and object path, respectively. The left summand ( $I_0 \cdot (s_r^2 + s_o^2)$ ) in equation 3.1 is the DC offset caused by the active illumination, and the right summand ( $I_0 \cdot 2s_r \cdot s_o$ ) is the scaling factor of the modulation. Thus, if there was a specialized pixel circuitry that could remove any DC components, the DC offset in equation 3.1, and any additional DC components caused by ambient lighting, the pixel would only need a dynamic range covering the intensity of the modulation. This would increase the effective dynamic range of the pixel. If ambient lighting is ignored, the factor would be:

$$\frac{\max(I_{D,AC})}{\max(I_D)} = \frac{2s_r \cdot s_o}{s_r^2 + s_o^2 + 2s_r \cdot s_o} \quad (3.2)$$

If the two reflectivities were equal ( $s_r = s_o = s$ ), the factor in equation 3.2 would be:

$$\frac{\max(I_{D,AC})|_{s_r=s_o}}{\max(I_D)|_{s_r=s_o}} = \frac{2s \cdot s}{s^2 + s^2 + 2s \cdot s} = \frac{2 \cdot s^2}{4 \cdot s^2} = \frac{1}{2} \quad (3.3)$$

In summary, a specialized pixel circuitry would be enormously beneficial in order to scan surfaces that would otherwise require a very large dynamic range. This might be the case with metallic surfaces that have been bead-blasted, which may result in local mirror-like spots while returning very little light in others. Another scenario would be a metallic part that was previously repaired in certain areas, and, therefore, contains areas that are polished, as well as areas that are dull right next to each other.

### 3.1.4 Desired Sensor Attributes

With the given inspection task in mind, a number of relevant attributes for a white light interferometer can be derived. Even without any specific knowledge about a specific measurement task, the following demands are universally desirable:

- Fast measurement speed
- High vertical resolution
- High lateral resolution
- Large measurement volume

---

<sup>1</sup>See equation A.89 in Appendix A.

Unfortunately, some of these requirements are contradictory. For instance, while white light interferometry has the ability to provide outstanding vertical resolution, if only the maximum envelope sample is taken as the depth information instead of calculating the true maximum, the information is only accurate to  $\pm$  half the sample distance and thus in the order of magnitude of tens of micrometers instead of in the sub-micrometer range. However, finding the optimum takes more time than a simple maximum sample detection. Another example would be that if a larger depth range and thus larger measurement volume were required, the sensor would have to move longer and capture more samples, resulting in longer scanning and processing times. A third example would be that, for a given pixel array size, the field of view would necessarily get smaller if an increased lateral resolution was chosen by using optics with a higher magnification.

With sufficiently large parts or parts that have a sufficiently complex shape, it is necessary to take measurements from several positions. This means that either the part, or the sensor, or both have to be moved. As will be detailed in the following section, the last approach was chosen for the example application of a rotationally symmetrical part, such as an airplane combustion chamber liner. Since the sensor is being moved, another requirement can be derived:

- Light-weight

Furthermore, since the sensor has to be inserted into the part in order to inspect it on the inside, yet another requirement is:

- Compact dimensions

In addition, the parts under consideration have an unpredictable state, potentially with highly reflective areas immediately neighboring highly diffuse parts. Finally, the lighting conditions are not predictable. Thus, an ideal sensor needs to fulfill the following requirement as well:

- High effective dynamic range
- Robustness against unwanted light

The requirements for the lateral resolution depend strictly on the application. If, for instance, cracks with a length of 5 mm and a width to length ratio of 1:60 are to be detected, and, if 4 pixels are required in order to perform robust crack detection, the required lateral resolution would be  $\frac{5 \text{ mm}}{60 \cdot 4} \approx 20 \mu\text{m}$ .

A strict requirement is that the measurement depth is large enough to capture the surface in every individual shot unless a deformation that is so severe that the sensor cannot capture it is defined as a defect. The measurement area, on the other hand, should be as large as possible in order to save time. Additionally, it is important to maximize the measurement speed for every individual scan. With what was said in subsection 3.1.2, this means that in order to maximize the measurement speed, a pixel-level analog demodulation should be performed so that only the envelope needs to be sampled, instead of sampling the signal directly and demodulating afterward. Ideally, some of the signal processing,

such as the maximum detection, should be done on a field-programmable gate array (FPGA) so that the amount of data that has to be transferred to the computer is drastically reduced from the  $n$  matrices containing depth samples to only two matrices containing the depth and amplitude information. Furthermore, in order to maximize the dynamic range and to be robust against external light, an active DC component suppression should be utilized on the pixel level. Thus, a specialized pixel array, with both in-pixel IQ demodulation and in-pixel current control, is required. Such a sensor is commercially available on the market. However, the author is contractually obligated not to reveal the manufacturer's name. Even though the currently available pixel IC has a comparatively small amount of pixels ( $300 \times 300$ ) and thus a small measurement area (at  $20 \mu\text{m}$ :  $6 \times 6 \text{ mm}^2$ ), it was possible to achieve the required inspection times due to time-optimal scanning (see section 4.2). As was mentioned before, the selected sensor was not designed for use in an automated system. The author, therefore, had to develop a driver that was suitable for use in the prototype software developed as proof of the concepts researched and developed for this dissertation. The basic driver structure is presented in Appendix C.

## 3.2 Architecture

In this section, the general architecture of the novel inspection system is described, including mechanical components and software aspects (software architecture and communication). Error handling is also a relevant aspect of any complex system. The error handling approach developed for this thesis can be found in Appendix D.

### 3.2.1 Mechanical System

The part under consideration is both too large and has too complex a shape to be captured with just a single scan. Rather, either a specialized or a universal handling machine with six degrees of freedom (DOF) is required in order to perform relative movements between the part and the sensor.

It was mentioned in the introduction, that the researched and developed method itself does not depend on a specific actuator, provided that the interferometer can be calibrated in a common coordinate system (a TCP calibration). In this particular case, an industrial robot with serial kinematics was chosen for the demonstrator because it combines a very high degree of flexibility, and thus technological transferability, with a large workspace. Its downsides are that due to its serial structure, it is relatively inaccurate in its absolute positioning and that relatively large masses have to be moved, potentially resulting in long settling times. It follows from the first issue that the captured scans will have to be corrected using a registration algorithm to yield a seamless point cloud. As was shown in the previous section, the sensor is highly sensitive to lateral and vertical movements. This means that the entire system has to be at rest during the measurement so that there is no movement between the sensor and the part. Because the part is significantly larger and heavier than the sensor, it is a reasonable decision to attach the sensor to the robot and position the sensor to capture the part, instead of vice versa. However, in order to reduce the influence of the robot's settling time, a seventh axis is used, which

rotates the part around its center of rotational symmetry as published in [ODH13]. This arrangement is illustrated in Figure 3.5.

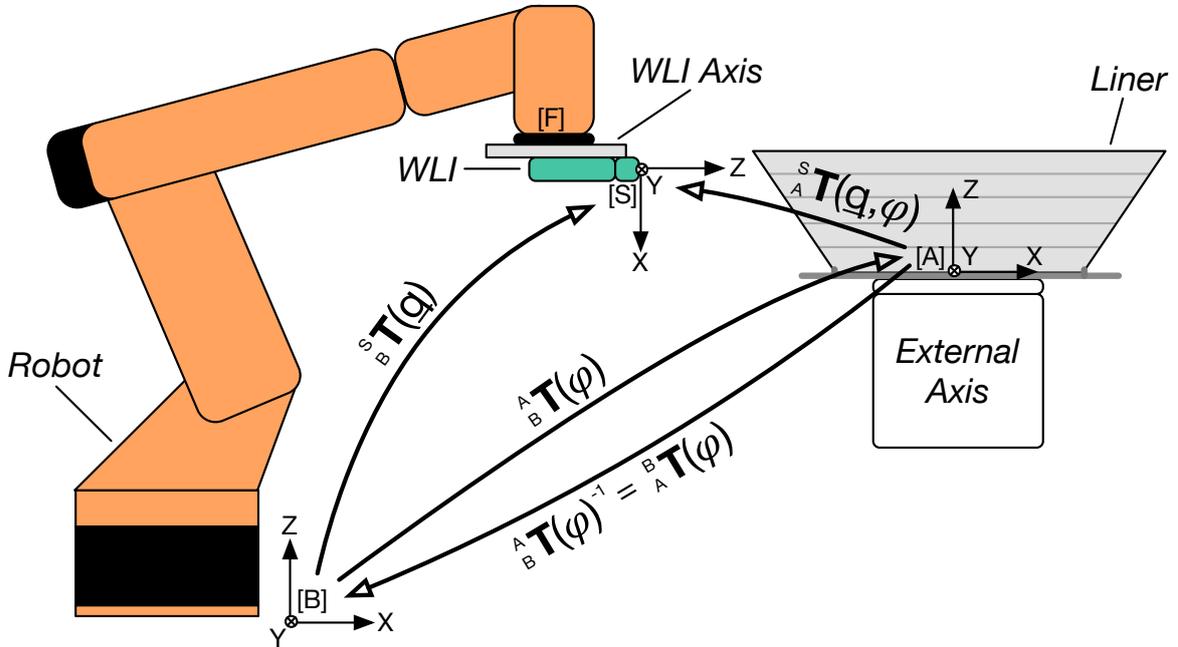


Figure 3.5: Illustration of the relevant coordinate systems.

Since the robot control system knows the Denavit-Hartenberg parameters describing the kinematic chain of the robotic arm and since it knows all the joint angles, the forward transform can be calculated, resulting in the position and orientation of the robot's flange in a common base coordinate system:  ${}^F_B \mathbf{T}(q)$ , where  $q$  is the current joint configuration. Thus, if the relationship between the robot's flange and the sensor ( ${}^S_F \mathbf{T}$ ) is known, the sensor's position and orientation in relation to the common base coordinate system is  ${}^S_B \mathbf{T}(q) = {}^F_B \mathbf{T}(q) \cdot {}^S_F \mathbf{T}$ . In general, such a calibration can be performed by scanning a well-known object under different joint configurations, extracting features, and solving a set of equations for the frame parameters (see for instance [GT11, pp. 117]). In this particular case, a sphere is captured, and a model is determined using non-linear regression. The features used for the least-squares frame calculation are the sphere center points. Here, the part is moved by an external rotational axis, which has its own coordinate system in relation to the base frame but depends on its current angle  $\varphi$ :  ${}^A_B \mathbf{T}(\varphi)$ . The sensor captures the part, which itself is moved by the external axis. This means that the sensor data has to be transformed into the axis coordinate system, not the robot's base coordinate system. The location and orientation of the axis coordinate system need to be calibrated as well, for instance, using the TCP-calibrated WLI. Since  ${}^A_B \mathbf{T}(\varphi)$  is a homogenous transform matrix, it can be inverted, yielding the opposite relationship:  ${}^A_B \mathbf{T}(\varphi)^{-1} = {}^B_A \mathbf{T}(\varphi)$ . Thus, the complete relationship transforming a given scan into the correct position and orientation in 3D space can be determined by the following expression:  ${}^S_A \mathbf{T}(q, \varphi) = {}^B_A \mathbf{T}(\varphi) \cdot {}^S_B \mathbf{T}(q)$ . This is illustrated in Figure 3.5. The ca-

libration procedure was performed by the IFPT, using the client-server architecture described in this dissertation, and the data captured and generated by the developed interferometer control software.

Because the part may have been in use, there may be local and global deformations. Therefore, the IFPT's control software AutoXIV determines the differences between a laser line section scan of the current part with that of a master sample and calculates an optimal path and adaptation depths for each measurement.

Since moving the entire robot to perform these adaptation movements would take too long, they are executed with one degree of freedom by the white light interferometer's linear axis. The corrections have to be applied in a time-optimal fashion by the software developed in this dissertation (see section 4.2). Thus, the sensor coordinate system  ${}^S\mathbf{T}(q)$  becomes dependent on the depth correction offset  $c$ :  ${}^S\mathbf{T}(q, c)$  and, consequently, the overall transform becomes  ${}^S\mathbf{T}(q, \varphi, c) = {}^B\mathbf{T}(\varphi) \cdot {}^S\mathbf{T}(q, c)$ . Since there is only one lateral degree of freedom, realized by a highly precise actuator at the very end of the kinematic chain, and since the sensor coordinate system's  $z$  axis is aligned with the correction direction, the correction can be applied directly by adding the correction value to the sensor coordinate system's origin's  $z$  value.<sup>1</sup>

### 3.2.2 Software Architecture and Communication

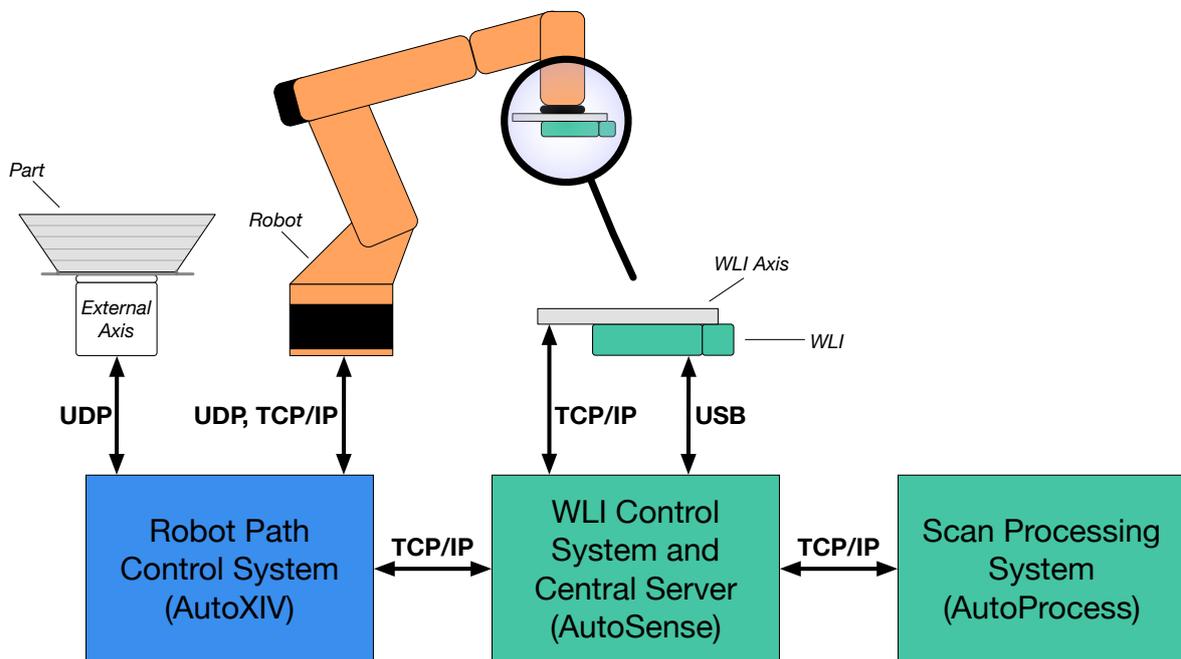


Figure 3.6: Illustration of the major system components and the communication between them.

Since time is a major factor and the overall process has the potential to be executed in a highly parallelized fashion, it was deemed beneficial to split the capturing and processing parts of the soft-

<sup>1</sup>The coordinate system was chosen in such a way that the  $z$  axis is pointing to the forward-facing measurement direction.

ware into two independent software packages. The communication between the software packages was realized with a custom communication protocol based on TCP/IP in a client-server configuration. As was mentioned in the previous subsection, the communication between the interferometer software components developed in this dissertation and the IFPT's actuator control software AutoXIV was also implemented in such a way. The fact that the software components communicate over network protocols means that the individual components can be deployed onto different computers. This is especially beneficial in the WLI control (AutoSense) and scan processing system (AutoProcess) because capturing and processing the interferometer scans requires massive amounts of RAM. The overall system structure is illustrated in Figure 3.6 on the preceding page.

### Main Data Structures

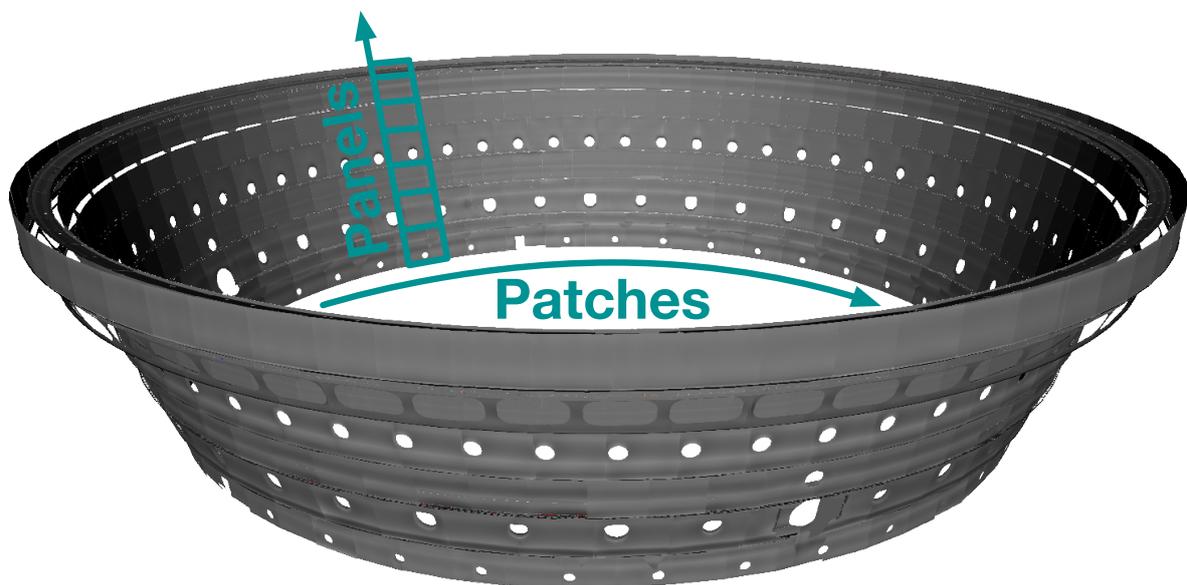


Figure 3.7: Illustration of patches and panels. The patches are continued over the entire liner circumference (as indicated by the “patches” arrow).

With the current state of the art, it would not be feasible to process the entire CFM-56 outer liner all at once. Instead, the part scans are subdivided into suitable subsets. On the one hand, these subsets should contain enough information to perform reasonable damage detection, but on the other hand, they should still be processable in a reasonable amount of time. It is recommendable to choose natural structures for the subdivision for several reasons. First, it makes locating the defective areas easier for visual confirmation of the findings on the actual part. Second, since cracks have a tendency to start from weak spots such as edges and cannot go beyond edges that are also local barriers, subsets that go from edge to edge have a good chance to contain entire cracks. In reality, simply dividing the data through natural structures might still result in datasets that are too large to handle or might have other downsides.

In the case of a CFM-56 combustion chamber outer liner, panels are a meaningful sub-division. However, as will be discussed in-depth in chapter 5, the damage detection is performed in 2D space. In order to keep the distortion minimal, this means that the impact of the part’s curvature has to be minimized. Thus, the panel is further subdivided into patches, as illustrated in Figure 3.7 on the preceding page. The liner in Figure 3.7 is shown in the “simple view” format, which uses down-scaled textures as a representation for each patch because even state-of-the-art GPUs would be unable to display the complete liner scan as a full point cloud.

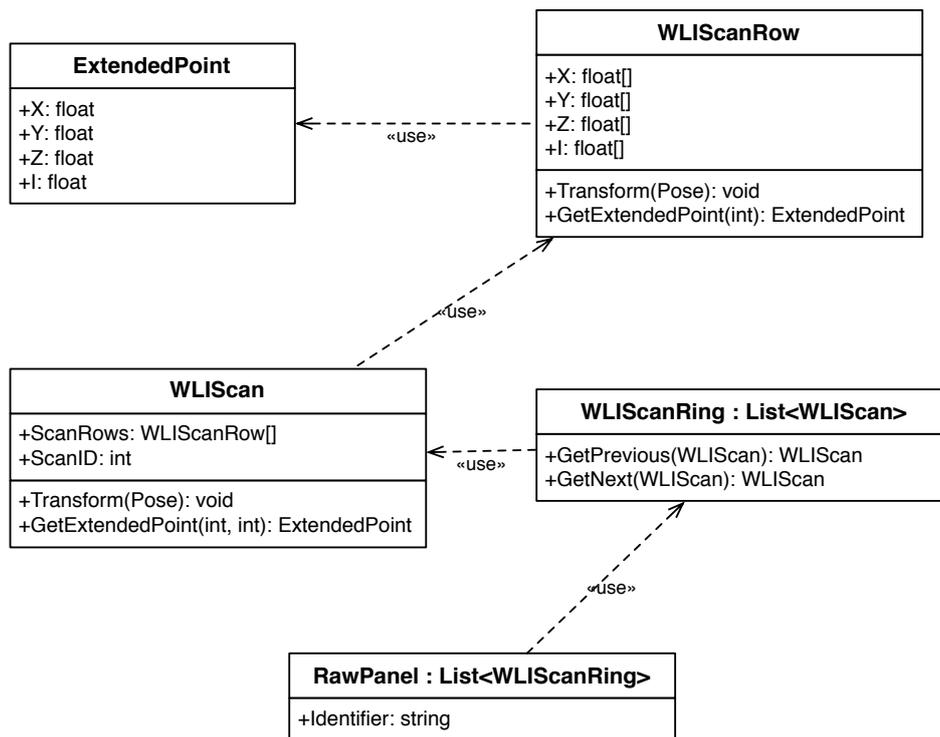


Figure 3.8: UML class diagram of *ExtendedPoint*, *WLIScanRow*, *WLIScan*, *WLIScanRing*, and *RawPanel*.

Even though the processing is performed patch-wise, as explained in the previous paragraph, the scanning is performed in a ring-wise fashion due to the mechanical setup introduced in subsection 3.2.1. This is to say the sensor remains stationary while the external axis moves the part, resulting in data along the circumference of the part. Because of this, the data is first acquired in the AutoSense application (see Figure 3.6) and stored in a panel data structure that itself contains rings of WLIScans. As soon as such a panel structure is filled with all rings belonging to the current panel, a new panel storage object is instantiated. These ring-storing data structures thus contain all the raw panel information. The class is, therefore, called “RawPanel.” Thus, every *RawPanel* object contains  $n$  *WLIScanRing* objects, which themselves contain *WLIScan* objects. *WLIScans* store their data in *WLIScanRow* objects, which ultimately store the acquired data in  $x$ ,  $y$ ,  $z$ , and intensity arrays. Furthermore, for each individual data point entry, an extended point object containing all four coordi-

nates ( $x$ ,  $y$ ,  $z$ , and intensity) can be requested in the helper structure `ExtendedPoint`. The classes detailed in this section are illustrated in a UML diagram in Figure 3.8 on the preceding page.

### Communication between Interferometer and Actuator Software

The communication between the IFPT's actuator software `AutoXIV` and the software `AutoSense`, which doubles as an interferometer control software and a communication server, is controlled through a state machine, which is depicted in Figure 3.9. The communication is implemented as a client-

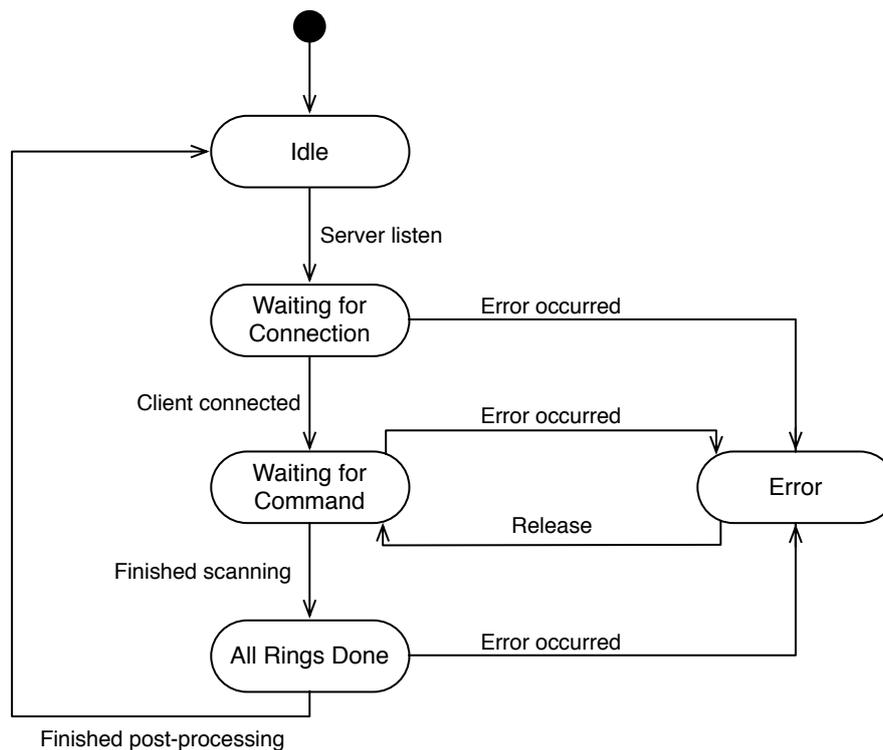


Figure 3.9: Illustration of the main state machine.

server architecture based on TCP/IP. After powering on `AutoSense`, it enters its idle state. Because the interferometer control software `AutoSense` offers the “service” of performing scans, it was chosen to be the server. It thus starts a TCP/IP server and tries to listen on the specified IP address and port as soon as it is fully initialized and the “Server listen” command is issued. If the underlying TCP/IP server can be bound to that port and successfully starts listening, the system transitions to the next state, “Waiting for Connection” and remains in there. Since the system only offers a 1:1 connection,<sup>1</sup> as soon as a client successfully connects to the system, the state changes to “Waiting for Command.” This is the main state in which the system performs most of its functionality. Upon finishing all

<sup>1</sup>It would be possible to offer a 1:n connection, because the use of the white light interferometer is synchronized using a producer-consumer approach (see Figure 3.10). However, because the interferometer is mounted onto a single robot and only has to be synchronized with one control system, it makes sense to limit the system to a 1:1 connection.

scans, the system enters the final state, “All Rings Done” in which certain post-processing tasks are performed. Afterward, the system resets itself and goes back into its initial state.

In the states “Waiting for Command” and “All Rings Done,” different commands can be received and processed. The general structure of such a command is `<command name>!<[optional:argument1; optional:argument2; ...]><cr><lf>`, where “<cr><lf>” is carriage return, line feed, equivalent to pressing the enter key on a Windows® computer. If a package is received by the communication handler, a command string is built up until <cr><lf> is found. Then the communication handler fires an event, which the command processor (containing the state machine) is registered to.

If the string happens to not contain an exclamation point, it is deemed invalid. Otherwise, the command string is split at the location of the exclamation point. Afterward, the substring before the exclamation point is checked against a list of valid commands. If a match was found and if the command expects additional arguments, the string after the exclamation point is parsed. If the remaining string does not satisfy the argument requirements, an error is thrown. Otherwise, the command is entered into a concurrent queue. This behavior serves as a producer in a producer-consumer pattern (see Figure 3.10). The consumer runs in a different thread and sequentially processes the commands in the

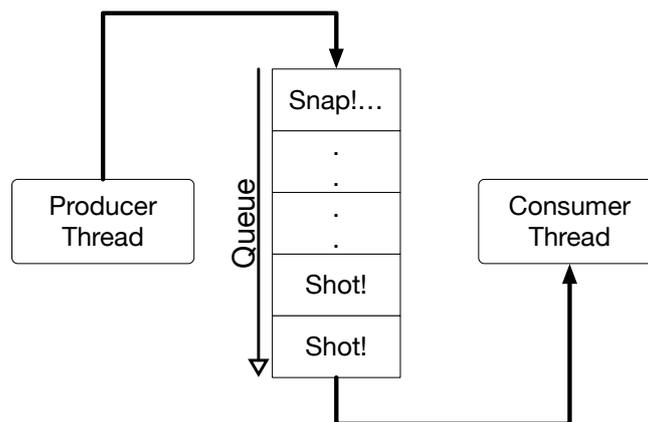


Figure 3.10: Illustration of the producer-consumer structure.

concurrent queue. As long as the consumer thread is running, it calls `TryTake` on the queue. As soon as a new data item is asynchronously entered into the queue, the runtime environment reactivates the thread, giving it CPU time to process the command. If the `TryTake` call times out, it is checked if the thread’s cancel token has been set, terminating it if required. Otherwise, `TryTake` is called again. The producer-consumer structure is illustrated in Figure 3.10. This producer-consumer approach ensures that commands are received immediately and then executed as soon as possible. There are three scan commands, which all cause the system to perform an interferometer scan. However, the data is used differently in each case.

The *Shot!* (from single shot) command takes the captured data, filters it according to the software’s filter settings and sends it to the client. In this scenario, `AutoSense` acts as a sensor driver server, meaning that the received command triggers a scan, which is sent back to the caller. In the approach

developed in conjunction with the IFPT, this particular command is used for the sensor's tool center point calibration. The robot interface software AutoXIV positions the WLI in such a way that a well-known sphere can be scanned. It then sends AutoSense the *Shot!* command to perform a scan of the sphere surface and then receives the resulting data. By combining the data with the robot's flange position and by repeating this procedure in several kinematic configurations, a linear set of equations can be constructed, the solution of which yields the interferometer's TCP.

The second scan command, *ValidationShot!...*, is used to perform a self-test: The interferometer is positioned in front of a sphere with a well-known radius, and the *ValidationShot!...* command is sent. AutoSense then scans the sphere, performs a sphere regression, and sends the resulting radius and relative sum of squared distance errors to the caller. This procedure can be used to verify that the system is working correctly and is described in detail in chapter 6.

The third and last scan command, *Snap!...*, is specific to the actual scanning process. A scan is performed and sorted into the previously mentioned data structures (see Figure 3.8) according to the metadata delivered with the command. It is, therefore, ultimately used to construct a large-scale 3D representation of the part, which is then processed by the second software developed for this dissertation: AutoProcess (see Figure 3.6 on page 50).

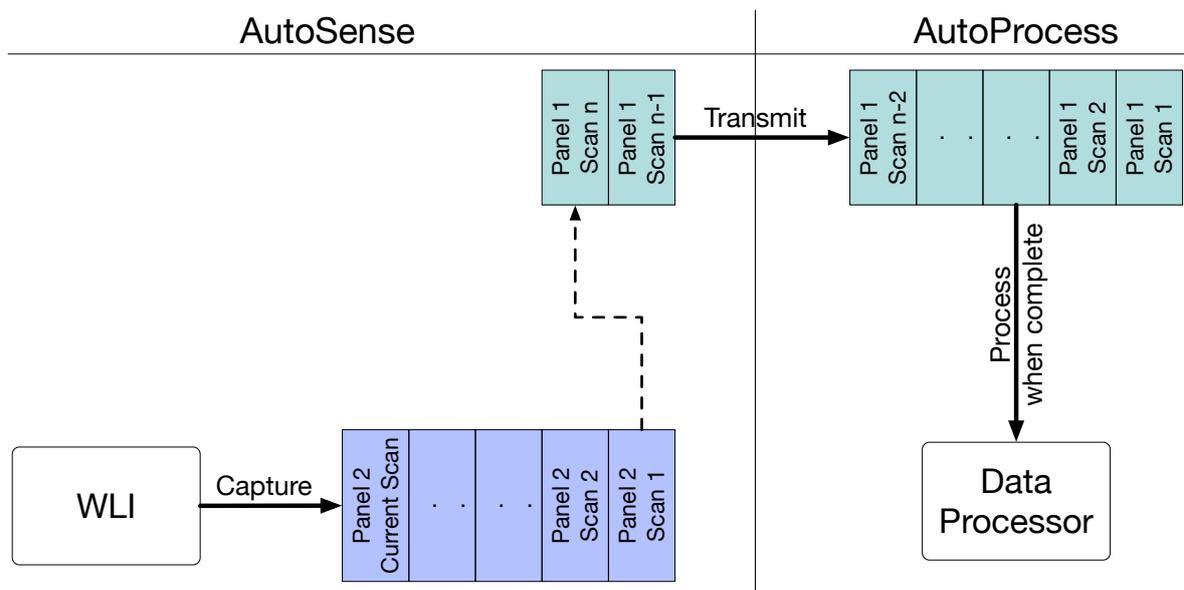


Figure 3.11: Illustration of buffering between AutoSense and AutoProcess.

The communication with AutoProcess is essentially data-driven. Since the idea is to split the part's 3D representation into panels and then into patches, all rings required for a raw data panel (see Figure 3.8) have to be completely filled first. As soon as AutoProcess has received a full panel, it can commence splitting it into raw patches (see section 5.1) and then perform the crack detection, as described in chapter 5. Since AutoProcess can only process one panel at a time due to memory restrictions, the following approach is used:

1. As long as AutoProcess is processing a panel, buffer all incoming data in appropriate data structures in AutoSense.
2. As soon as AutoProcess signals that it is done with processing the current panel, send all collected data for the *next* one.
3. As long as AutoProcess is not processing a panel and all buffered data has been sent to AutoSense, send new data as soon as it is available.

This buffering approach is illustrated in Figure 3.11 on the preceding page.

## Chapter 4

# Automated White Light Interferometer Scanning

In this chapter, the integration of a white light interferometer in an automated and highly adaptive process is described. This includes depth parameter determination, time-optimal scanning, and white light interferometer data pre-processing, all of which require an understanding of the fundamentals of white light interferometry (a complete derivation of a sufficiently accurate white light interferometry model is given in Appendix A).

### 4.1 Depth Parameter Determination

Configuring a WLI involves a relatively large parameter space. Where most of these settings can be tuned somewhat independently for a given situation, there are also several interdependent settings. Most importantly, the vertical measurement depth is dependent on the scanning frequency (camera

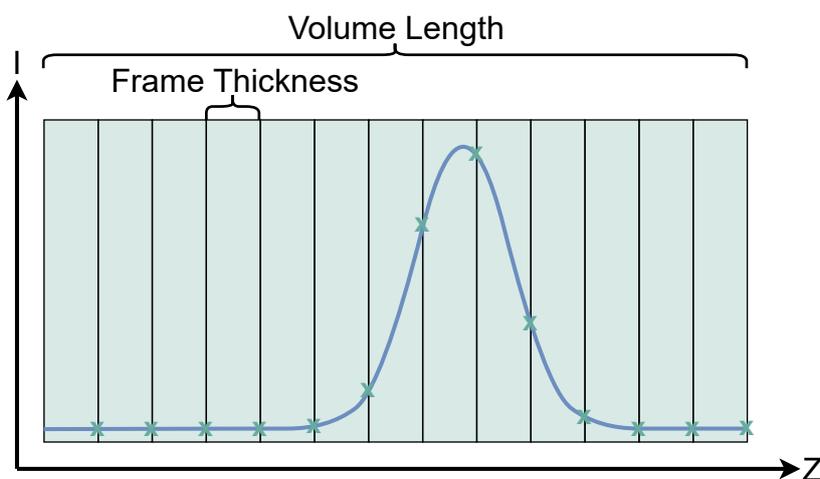


Figure 4.1: Illustration of how signal sampling (green x-markers), frame thickness, and volume length are related.

setting), the number of frames captured during a scan (camera setting), and the scanning velocity (axis setting). However, the velocity and scanning frequency also have an impact on how much time the interferometer has to integrate light per frame, as well as how “thick” the individual frames are. Figure 4.1 on the preceding page illustrates how the depth sampling rate and thickness of frames are related. Since the interferometric scan is performed during steady-state motion, the time the camera has to capture light for a single frame is directly related to the traveled distance between two frames and thus for an apparent thickness, known as frame thickness. Each frame delivers a sample per pixel, which is used for signal reconstruction and, ultimately, the distance calculation. This frame thickness is, therefore, the sampling distance. The captured volume is defined by the scanned area times the measurement depth of the volume. This measurement depth, which is called the volume length, is simply defined by the number of scanned frames times the frame thickness, as shown in Figure 4.1.

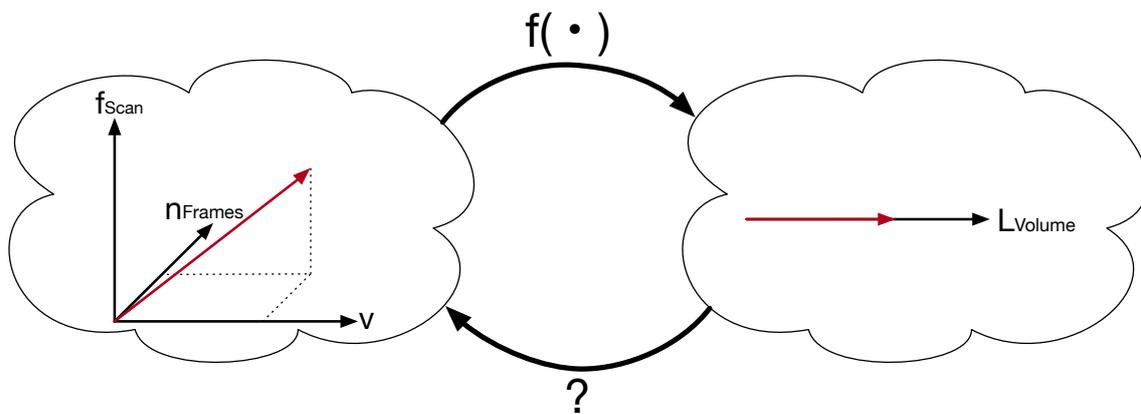


Figure 4.2: Illustration of the mapping between the parameter space and the resulting volume length.

Thus, there is a three-dimensional parameter space, in which every point is mapped into a one-dimensional measurement depth space, as illustrated in Figure 4.2. Since the mapping is surjective only, it is directly possible to find the resulting volume length for a given parameter set, but much more challenging to find suitable parameters for a given volume length (inversion problem). To find an inverse mapping, additional constraints have to be defined.

The following approach is based on the fact that the three-dimensional settings space is *discrete*, simply because the interferometric camera and axis only allow a finite and countable number of settings for each parameter. Furthermore, since  $f_{\text{Scan}}$  and  $v$  are coupled in that they both determine the thickness of the frames, the dimensionality of the parameter space can be reduced by mapping those two parameters into a thickness domain. This can be done by finding a reasonable value for the velocity and then keeping it fixed. Keeping the velocity fixed has the advantage that the other motion parameters, such as jerk, acceleration, and closed-loop controller settings, can be tuned for the chosen velocity, resulting in a smooth motion profile.

Since the distance covered between frames depends on how fast the WLI moves, the scanning velocity defines an operating point for the frame thickness. It should therefore be chosen in such a way

that the camera's scan frequency and the number of frame parameters allow to cover the required volume lengths for the given measurement process. In other words, the velocity should center the frame thickness parameter in the required range. Thus, with the known wavelength and fixed velocity, the resulting demodulation frequency can be determined,<sup>1</sup> and the corresponding demodulation registers can be set accordingly in the interferometric camera.

Even with the reduced dimensionality, the inverse mapping is not unique. Another constraint can be derived if the quality of the scan is taken into account. The scan frequency determines how fast the camera captures individual frames. Since the camera moves at a constant velocity during the active scanning time, this directly implicates that the scan frequency also determines the frame thickness. Therefore, the higher the scan frequency, the smaller the time used to capture the individual frames and thus, the smaller the frame thickness and thus volume length for a fixed number of frames. Despite the special interferometric measurement method, at the end of the day, a photoactive silicon array integrates the incoming photons. Accordingly, shorter integration times result in smaller signal intensities, which can degrade signal quality. Lower frequencies, on the other hand, result in longer exposure times and increased signal quality, as well as thicker frames and therefore bigger volume lengths. This means, however, that a longer range is averaged in every frame, leading to larger jumps in the depth discretization. However, as long as the steps are close enough to provide a sufficient amount of samples of the expected Gaussian distribution, the true depth information can be recovered with far higher accuracy than the frame resolution.<sup>2</sup> Therefore, for a wide range of cases, large frame thicknesses are preferable because they result in higher quality signals and yield chosen volume lengths at smaller numbers of frames. The latter results in an additional time benefit because fewer frames have to be processed to yield the resulting scans.

Given these constraints, a basic strategy for inverting the volume length mapping can be derived:

1. Choose a velocity.
2. Find the smallest reasonable scan frequency/number of frames combination that results in a volume length as close as possible to but at least as large as the desired one.

Here, reasonable means that stable sub-frame interpolation should be possible. It should be emphasized once more that the determined volume length has to be at least as large as the requested one in order to guarantee that the entire surface gets captured.

As is often the case in algorithm design, there is a trade-off between memory consumption and speed. Since the number of possible settings in both the number of frames and the scan frequencies is discrete and finite, it is possible in principle to use a lookup table (LUT) approach. This means that the LUT has to be populated once and yields very fast results afterward. The downside is that it permanently occupies a certain amount of RAM; however in this case, the space requirements are quite low. The two-dimensional LUT (see Figure 4.3, left) can simply be constructed by performing

<sup>1</sup>For more information, see equation A.66 in Appendix A.

<sup>2</sup>See section A.4 in Appendix A.

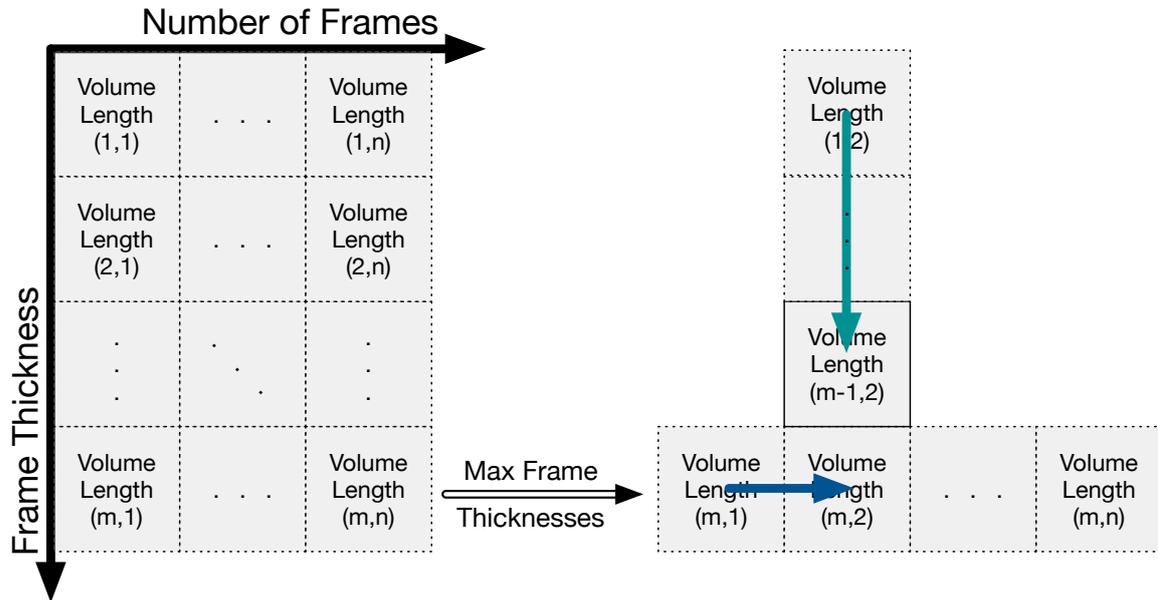


Figure 4.3: Illustration of the volume length LUT and the search algorithm. Note: the number in parentheses denote indices, not actual values. This was done for clarity. In a real-world implementation, concrete values would be used.

the volume length mapping (see Figure 4.2) for a fixed velocity  $v_c$ . Finding the closest match to the desired volume length is a two-dimensional search problem, which might take relatively long because of the potentially large parameter space. Fortunately, the parameter search can actually be decomposed into two one-dimensional searches:

Initially, the first volume length that is greater or equal to the required volume length is searched along the number of frames dimension, at maximum frame thickness. If no such entry is found, the problem is not solvable under the given constraints, and an emergency strategy has to be used. If the found volume length is exactly equal to the search volume length, the settings set has been found. Otherwise, only the optimum number of frames has been found. In that case, the best match can be found by searching along the frame thickness dimension at the determined frame number. The search objective is again to find the first entry, which is *greater or equal to* the required volume length. This approach is illustrated in Figure 4.3.

The algorithm described in the previous paragraph does not guarantee that the closest possible volume length mapping is found. It does, however, guarantee the following:

1. The determined settings set results in a volume length greater or equal to the desired one.
2. The smallest possible number of frames and largest possible frame thickness is used to fulfill the inequality while getting as close as possible to the desired volume length under the additional constraints.

In the example shown in Figure 4.3, the algorithm finds the first volume length, which is larger than the desired length at position (m,2) (blue arrow). The found volume length is not a perfect match. Thus, only the number of frames has been determined, and the next search has to be started (green arrow). In the example, the penultimate element is either the first element to be larger than the requested volume length or a perfect match. In the algorithm implemented for this dissertation, the value searches are not performed in a simple linear fashion, as suggested by the arrows in Figure 4.3. Instead, binary search is used, which is asymptotically much more efficient ( $\mathcal{O}(\log(n))$  instead of  $\mathcal{O}(n)$ ).

As mentioned before, in the event that no solution could be found for the requested volume length for the chosen velocity, an emergency strategy has to be used. One straightforward solution would be to simply return an error, especially when the sensor is operating at its quality limits. However, if there is some quality headroom, an alternative search can be performed. Under those circumstances, the maximum number of frames is chosen because this only impacts the time, but not the quality. This reduces the dimensionality of the problem to one, which means that an inverse mapping between the volume length and the frequency can be calculated directly from the frame thickness, as opposed to extracting it from the LUT. Afterward, a range check should be performed in order to determine if the resulting parameters are acceptable. A better approach would be to define the absolute maximum volume length and check against it in the beginning in order to avoid performing the described parameter search, only to discard the results immediately afterward in cases that exceed the parameter ranges.

Finally, it should be noted that even though the LUT entries have to be calculated from values with a physical meaning (frame thickness and number of frames), they can actually be replaced with the resulting camera register parameters. This means that no additional calculations have to be performed online in order to yield parameter settings from metric values. This is not the case if the alternative search is performed since there are no LUT entries for these emergency cases, meaning that both the physical values and register settings have to be determined.

## 4.2 Time-Optimal Scanning

Since several tens of thousands of scans are required in order to inspect a CFM-56 outer liner, it is imperative to reduce the time each individual scan takes as much as possible. The interferometer's axis is set up to perform a movement toward or away from the measured object inside the camera's operating distance. Once a defined position on the axis is hit, the axis triggers the interferometer's camera to start capturing. The specialized IC (see section 3.1) is configured to demodulate the captured signals and to generate a defined amount of captures, which is to say matrices of samples of the demodulated interferometric signals. The camera contains a field-programmable gate array (FPGA), which performs the envelope maximum determination in parallel and thereby creates the output data: a distance and an amplitude image. The resulting data can then be sent to the host computer via USB. Unfortunately, the data processing performed on the FPGA takes a significant amount of time.

There are several aspects of the overall automated scanning process that can be parallelized, including the  $z$ -adaptation motion and moving the part and/or robot to the next measurement position. It is important to note that the interferometric camera itself is unaware of the motion state of the linear axis. Rather it starts capturing for a set amount of time after it has received a hardware trigger signal. Thus, the WLI motion pre-planning, which is performed by the driver software developed by the author, has to ensure that all requirements for a successful scan are met while keeping the required time as short as possible and minimizing vibrations being introduced into the system. Of course, the actual motion planning and control are executed by the linear axis hardware. “Motion pre-planning” describes the steps required to make the axis perform its internal motion planning and execute its motion in the desired way. This means that the axis’ motion profile needs to be modeled in order to allow the time predictions required for the parallelization approach presented in this section.

#### 4.2.1 General Scanning Motion Profile

The white light interferometer’s heterodyne scanning approach is based on a constant scanning velocity  $v_{\text{scan}}$  during the active measurement time (see, for instance, equation A.66 in Appendix A). Because the three-dimensional direction is irrelevant for the model, velocity, acceleration, and jerk are treated as scalars instead of vectors in the following derivation.

During capture, the axis has to be in a state of constant velocity. The time requirements for the active measurement interval can thus simply be calculated by dividing the desired measurement depth by the measurement velocity (with  $v_{\text{Meas}} = v_{\text{max}}$ ):

$$t_{\text{Meas}} = \frac{d_{\text{Meas}}}{v_{\text{Meas}}} \quad (4.1)$$

However, the interferometer is not always in the required relative state of motion. Rather it needs to be brought within the vicinity of the object and then accelerated to the desired velocity. The force required to change an object’s velocity is directly proportional to the rate of change:  $F = m \cdot dv(t)/dt$ . Therefore, reaching the desired velocity immediately, i.e., make it follow a step function, would imply an infinitely large and short force input, a Dirac pulse, which is impossible to generate. Thus, in order to reach the desired velocity, the system needs to go through an acceleration phase. The time it takes to get to the set velocity has to be added to the time requirements. The overall motion profile should be symmetrical. It thus follows that the same amount of time is required to decelerate the axis to reach its initial velocity, which is zero in the axis’ frame of reference. In a coordinate system that points toward the object in a forward measurement motion, the acceleration is at first positive, then zero during the time of the measurement, due to the constant velocity requirement, and finally negative.

Unfortunately, attempting to change the acceleration instantaneously would also introduce undesired vibrations into the system. In other words, the rate of change of the acceleration, the jerk

$j(t) = da(t)/dt$ , also needs to be limited. Using  $p(t)$  for the position, it can be established that:

$$j(t) = \frac{da(t)}{dt} = \frac{d^2v(t)}{dt^2} = \frac{d^3p(t)}{dt^3} \quad (4.2)$$

A continuous linear and time-invariant state-space model without direct feedthrough can be expressed as [Lof90, p. 58]:

$$\dot{\underline{x}}(t) = \mathbf{F} \cdot \underline{x}(t) + \mathbf{B} \cdot \underline{u}(t) \quad (4.3a)$$

$$\underline{y}(t) = \mathbf{C} \cdot \underline{x}(t) \quad (4.3b)$$

Here,  $\mathbf{F}$  is the system matrix,  $\mathbf{B}$  is the input matrix,  $\mathbf{C}$  is the output matrix,  $\underline{x}(t)$  is the state vector,  $\underline{u}(t)$  is the input vector, and  $\underline{y}(t)$  is the output vector. In this case, the state  $\underline{x}$  can be modeled as a collection of increasing derivative orders of the position, starting with the position itself (see equation 4.2):

$$\underline{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ \dot{x}_1(t) \\ \ddot{x}_1(t) \end{bmatrix} = \begin{bmatrix} p(t) \\ v(t) \\ a(t) \end{bmatrix} \quad (4.4)$$

Thus, the derivative of the state is:

$$\dot{\underline{x}}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ x_3(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ a(t) \\ j(t) \end{bmatrix} \quad (4.5)$$

And the input is the scalar jerk:

$$u(t) = j(t) = \dot{x}_3(t) \quad (4.6)$$

From these definitions, the  $\mathbf{F}$  and  $\mathbf{B}$  matrices can be directly defined (see equation 4.7a). The output shall be a direct observation of the complete state vector. It, therefore, follows that  $\mathbf{C}$  is the identity matrix. A possible state space can thus be expressed in the following form:

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \underline{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot u(t) \quad (4.7a)$$

$$\underline{y}(t) = \mathbf{I} \cdot \underline{x}(t) \quad (4.7b)$$

The general solution of the set of differential equations in 4.7 can be expressed with the global

state transition matrix  $\Phi$ , which depends only on the difference between the current time  $t$  and the starting time  $t_0$  in the time-invariant case. From [Lof90, p. 66]:

$$\underline{x}(t) = \Phi(t - t_0) \cdot \underline{x}(t_0) + \int_{t_0}^t \Phi(t - \tau) \cdot \mathbf{B} \cdot u(\tau) d\tau \quad (4.8)$$

Where  $\Phi(t - t_0)$  can be understood as a Taylor series expansion of the following form [ibidem, p. 66]:

$$\Phi(t - t_0) = e^{\mathbf{F} \cdot (t - t_0)} = \mathbf{I} + \mathbf{F} \cdot (t - t_0) + \frac{1}{2!} \mathbf{F}^2 \cdot (t - t_0)^2 + \dots \quad (4.9)$$

With equation 4.9, the global state transition matrix can be determined.

$$\mathbf{F}^2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.10a)$$

$$\mathbf{F}^3 = 0_{3,3} \quad (4.10b)$$

Thus, the highest matrix power is of second order and the global state transition matrix becomes:

$$\Phi(t - t_0) = \begin{bmatrix} 1 & (t - t_0) & \frac{1}{2}(t - t_0)^2 \\ 0 & 1 & (t - t_0) \\ 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

The general solution in equation 4.8 consists of the homogenous solution  $\underline{x}_h(t) = \underline{x}_h(t) = \Phi(t - t_0) \cdot \underline{x}(t_0)$  and the particular solution  $\underline{x}_p(t)$ , which is the integral part of the sum. With equation 4.11 the homogenous solution becomes:

$$\begin{aligned} \underline{x}_h(t) &= \begin{bmatrix} 1 & (t - t_0) & \frac{1}{2}(t - t_0)^2 \\ 0 & 1 & (t - t_0) \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p(t_0) \\ v(t_0) \\ a(t_0) \end{bmatrix} \\ &= \begin{bmatrix} p(t_0) + (t - t_0) \cdot v(t_0) + \frac{1}{2}(t - t_0)^2 \cdot a(t_0) \\ v(t_0) + (t - t_0) \cdot a(t_0) \\ a(t_0) \end{bmatrix} \end{aligned} \quad (4.12)$$

The particular solution, on the other hand, consists of a causal convolution integral:

$$\underline{x}_p(t) = \int_{t_0}^t \begin{bmatrix} 1 & (t - \tau) & \frac{1}{2}(t - \tau)^2 \\ 0 & 1 & (t - \tau) \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot u(\tau) \, d\tau = \begin{bmatrix} \int_{t_0}^t \frac{1}{2}(t - \tau)^2 \cdot j(\tau) \, d\tau \\ \int_{t_0}^t (t - \tau) \cdot j(\tau) \, d\tau \\ \int_{t_0}^t j(\tau) \, d\tau \end{bmatrix} \quad (4.13)$$

Using the identity matrix as the observation matrix (see equation 4.7b), the observed state  $y(t)$  can be written as:

$$\begin{bmatrix} p(t) \\ v(t) \\ a(t) \end{bmatrix} = \begin{bmatrix} p(t_0) + (t - t_0) \cdot v(t_0) + \frac{1}{2}(t - t_0)^2 \cdot a(t_0) \\ v(t_0) + (t - t_0) \cdot a(t_0) \\ a(t_0) \end{bmatrix} + \begin{bmatrix} \int_{t_0}^t \frac{1}{2}(t - \tau)^2 \cdot j(\tau) \, d\tau \\ \int_{t_0}^t (t - \tau) \cdot j(\tau) \, d\tau \\ \int_{t_0}^t j(\tau) \, d\tau \end{bmatrix} \quad (4.14)$$

Given how the state vector was defined as a collection of increasing derivative orders of the position (starting with the position itself, see equation 4.4), equation 4.14 describes the full dynamic behavior of the linear axis. First of all, the observed acceleration can be immediately expressed as the sum of some initial acceleration (homogenous solution) and the integral of the input jerk from the starting point to the current point in time (particular solution):

$$a(t) = a(t_0) + \int_{t_0}^t j(\tau) \, d\tau \quad (4.15)$$

Similarly, the velocity expression consists of a homogenous part (the initial velocity plus the effect of the initial acceleration during the observed timeframe) and the integral of the time-weighted effects of the jerk during the observed timeframe:

$$v(t) = v(t_0) + (t - t_0) \cdot a(t_0) + \int_{t_0}^t (t - \tau) \cdot j(\tau) \, d\tau \quad (4.16)$$

And finally, the position can be determined as:

$$p(t) = p(t_0) + (t - t_0) \cdot v(t_0) + \frac{1}{2}(t - t_0)^2 \cdot a(t_0) + \int_{t_0}^t \frac{1}{2}(t - \tau)^2 \cdot j(\tau) \, d\tau \quad (4.17)$$

Since only the jerk was demanded to be bounded, but no demands were made for the jerk's derivative (the jounce), it can be modeled simply as a series of step functions with finite step size, see Figure 4.4. Both the maximum acceleration and the jerk are free parameters that have to be chosen sensibly. It should be considered that high demands on the jerk and acceleration settings might cause the system to vibrate, which would necessitate additional waiting times in order for those oscillations to subside. Very low settings, on the other hand, might lead to disproportionately long acceleration times. In practice, it was found that the choice of parameters depends on the specific hardware and might differ slightly even between two linear axes of the same type from the same manufacturer.

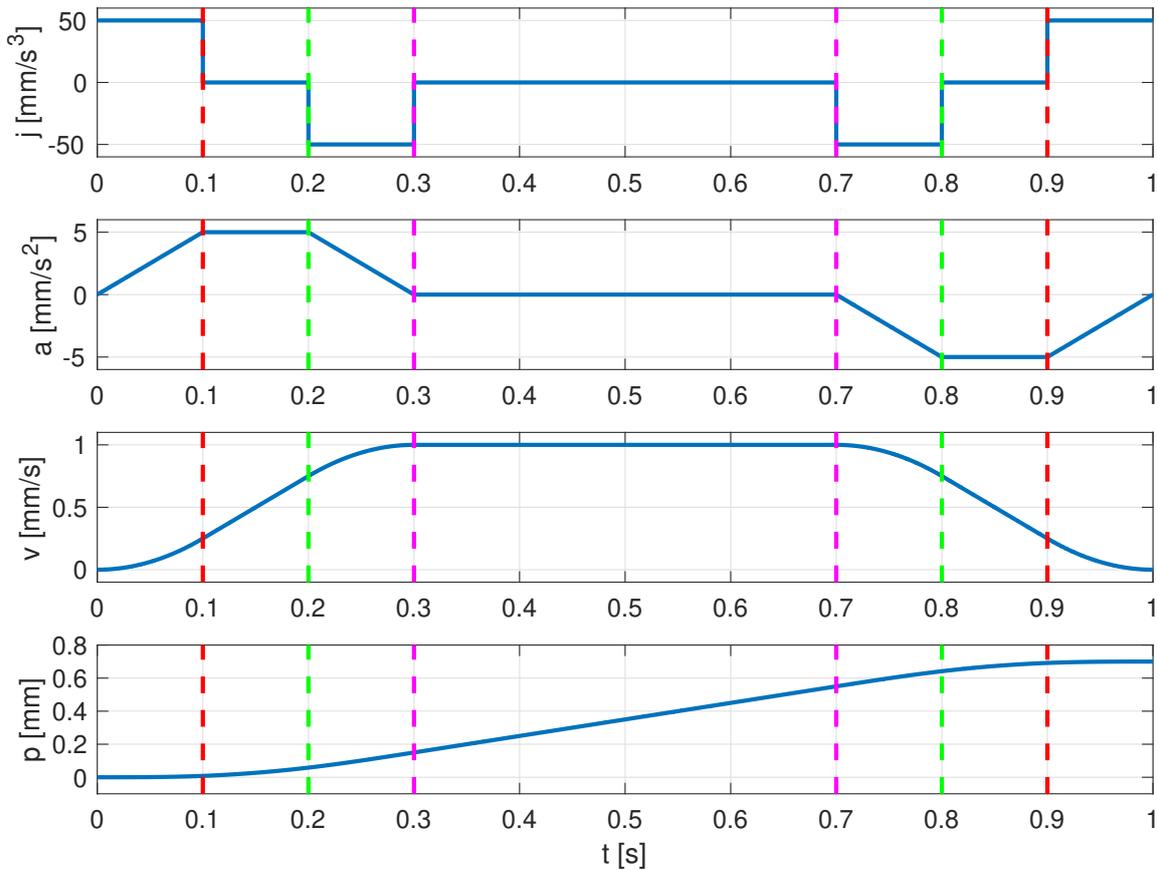


Figure 4.4: Illustration of an s-curve motion profile. The end of the first non-zero jerk phase ( $t_1$ ) is marked red, the beginning of the second one ( $t_2$ ) green, and the end of the second one ( $t_3$ ) magenta. The period of constant velocity goes to the second magenta line, which is followed by a deceleration phase, which is the symmetrical opposite of the acceleration phase.

Equations 4.15, 4.16, and 4.17 describe the entire motion profile in terms of a homogenous and a particular part, which depends on the input  $j(t)$ . In order to get the solutions for these equations, integrals have to be solved. Since the input function was defined to be piece-wise constant, the solutions can be found in those intervals. The interval borders themselves will be determined from the solved equations afterward.

**Interval 1:**  $0 < t \leq t_1$

In this interval the input is the maximum jerk:  $j(t) = j_{\max}$ . Furthermore, the linear axis is assumed to be at rest and at position 0, thus  $p(0) = 0$ ,  $v(0) = 0$  and  $a(0) = 0$ . It follows directly from equations

4.15, 4.16, and 4.17 for  $0 < t \leq t_1$ :

$$a(t) = j_{\max} \cdot t \quad (4.18)$$

$$v(t) = \int_0^t (t - \tau) \cdot j_{\max} \, d\tau = \frac{1}{2} j_{\max} \cdot t^2 = \frac{1}{2} a(t) \cdot t \quad (4.19)$$

$$p(t) = \frac{1}{2} j_{\max} \cdot \int_0^t (t - \tau)^2 \, d\tau = \frac{1}{6} j_{\max} t^3 = \frac{1}{3} v(t) \cdot t \quad (4.20)$$

**Interval 2:**  $t_1 < t \leq t_2$

In this interval the input is zero:  $j(t) = 0$ . Thus, the system behavior is entirely governed by its initial conditions, which can be determined from equations 4.18, 4.19, and 4.20 by evaluating them at  $t_1$ . It follows from equations 4.15, 4.16, and 4.17 for  $t_1 < t \leq t_2$  with the initial conditions at  $t_1$ :

$$a(t) = a(t_1) = j_{\max} \cdot t_1 \quad (4.21)$$

$$v(t) = v(t_1) + (t - t_1) \cdot a(t_1) \quad (4.22)$$

$$\begin{aligned} &= \frac{1}{2} a(t_1) \cdot t_1 + a(t_1) \cdot (t - t_1) \\ &= a(t_1) \cdot \left( t - \frac{1}{2} t_1 \right) \end{aligned} \quad (4.23)$$

$$p(t) = p(t_1) + (t - t_1) \cdot v(t_1) + \frac{1}{2} (t - t_1)^2 \cdot a(t_1) \quad (4.24)$$

$$\begin{aligned} &= p(t_1) + (t - t_1) \cdot \frac{1}{2} a(t_1) \cdot t_1 + \frac{1}{2} (t - t_1)^2 \cdot a(t_1) \\ &= p(t_1) + \frac{1}{2} a(t_1) \cdot t \cdot (t - t_1) \end{aligned} \quad (4.25)$$

**Interval 3:**  $t_2 < t \leq t_3$

In the last acceleration interval, the system is brought to its constant measurement velocity. The acceleration thus has to be at 0 at the end of the interval. This means that the jerk has to be  $j(t) = -j_{\max}$  for the same time as  $t_1$  was at the positive maximum jerk. It follows from equations 4.15, 4.16, and 4.17 for  $t_2 < t \leq t_3$ , with initial conditions from equations 4.21, 4.23, and 4.25 evaluated at  $t_2$ :

$$a(t) = a(t_2) - \int_{t_2}^t j_{\max} \, d\tau = j_{\max} \cdot (t_1 + t_2 - t) \quad (4.26)$$

$$v(t) = v(t_2) + (t - t_2) \cdot a(t_2) - j_{\max} \cdot \int_{t_2}^t t - \tau \, d\tau \quad (4.27)$$

$$\begin{aligned} &= a(t_1) \cdot \left( t_2 - \frac{1}{2}t_1 \right) + a(t_1) \cdot (t - t_2) - \frac{1}{2} \cdot j_{\max} \cdot (t - t_2)^2 \\ &= a(t_1) \cdot \left( t - \frac{1}{2}t_1 \right) - \frac{1}{2}j_{\max} \cdot (t - t_2)^2 \end{aligned} \quad (4.28)$$

$$p(t) = p(t_2) + (t - t_2) \cdot v(t_2) + \frac{1}{2}(t - t_2)^2 \cdot a(t_2) - \frac{1}{2}j_{\max} \cdot \int_{t_2}^t (t_3 - \tau)^2 \, d\tau \quad (4.29)$$

$$\begin{aligned} &= \left[ p(t_1) + \frac{1}{2}a(t_1) \cdot t_2 \cdot (t_2 - t_1) \right] + (t - t_2) \cdot \left[ a(t_1) \cdot \left( t_2 - \frac{1}{2}t_1 \right) \right] \\ &+ \frac{1}{2}(t - t_2)^2 \cdot [a(t_1)] - \frac{1}{2}j_{\max} \cdot \left[ \frac{1}{3}(t - t_2)^3 \right] \\ &= p(t_1) + \frac{1}{2}a(t_1) \cdot t \cdot (t - t_1) - \frac{1}{6}j_{\max}(t - t_2)^3 \end{aligned} \quad (4.30)$$

Due to symmetry and the necessity of zero acceleration at the end of the interval, the time delta  $t_3 - t_2$  must be exactly the same as  $t_1$ :

$$t_1 = t_3 - t_2 \quad (4.31)$$

Considering this relationship and evaluating equations 4.26, 4.28, and 4.30 at  $t_3$  yields:

$$a(t_3) = j_{\max} \cdot (t_1 - (t_3 - t_2)) = j_{\max} \cdot (t_1 - t_1) = 0 \quad (4.32)$$

Thus, the zero-acceleration demand is fulfilled. The velocity at the end of the interval is:

$$v(t_3) = a(t_1) \cdot \left( t_3 - \frac{1}{2}t_1 \right) - \frac{1}{2}j_{\max} \cdot (t_1)^2 \quad (4.33)$$

$$= a(t_1) \cdot (t_3 - t_1) = a(t_1) \cdot t_2 = v_{\max} \quad (4.34)$$

Finally, the position at the end of the acceleration phase is:

$$p(t_3) = p(t_1) + \frac{1}{2}a(t_1) \cdot t_3 \cdot (t_3 - t_1) - \frac{1}{6}j_{\max} \cdot (t_3 - t_2)^3 \quad (4.35)$$

$$\begin{aligned} &= p(t_1) + \frac{1}{2}a(t_1) \cdot t_1 \cdot t_3 - p(t_1) \\ &= v(t_1) \cdot t_3 \end{aligned} \quad (4.36)$$

**Interval 4:**  $t_3 < t \leq t_4$

In the active measurement interval, the jerk and acceleration are zero, and the velocity is the measurement velocity  $v_{\max} = v_{\text{Meas}}$ . The motion profile in the interval  $t_3 < t \leq t_4$  can thus be described as

follows:

$$a(t) = 0 \quad (4.37)$$

$$v(t) = v_{\max} \quad (4.38)$$

$$p(t) = p(t_3) + (t - t_3) \cdot v_{\max} \quad (4.39)$$

Considering that  $t_4 - t_3 = t_{\text{Meas}}$ , it follows that the position at the end of the interval is (see equation 4.1):

$$p(t_4) = p(t_3) + t_{\text{Meas}} \cdot v_{\text{Meas}} = p(t_3) + d_{\text{Meas}} \quad (4.40)$$

With the piece-wise description of the acceleration, velocity and position fully derived, the last remaining unknown parameters are the interval positions ( $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$ ).

### Determination of the First Interval

It follows from equation 4.18 evaluated at  $t_1$  that:

$$t_1 = \frac{a(t_1)}{j_{\max}} \quad (4.41)$$

Similarly, it follows from equation 4.19 evaluated at  $t_1$  that:

$$t_1 = \frac{2 \cdot v(t_1)}{a(t_1)} \quad (4.42)$$

The time  $t_1$  can also be expressed in terms of velocity and jerk by inserting equation 4.41 into 4.42:

$$t_1 = \sqrt{\frac{2 \cdot v(t_1)}{j_{\max}}} \quad (4.43)$$

However, with only two linearly independent equations (4.41 and 4.42) and three unknowns ( $t_1$ ,  $v(t_1)$ , and  $a(t_1)$ ),  $t_1$  cannot be determined without further information.

Equation 4.41 and 4.42 delivers the following expression for  $v(t_1)$ :

$$v(t_1) = \frac{a(t_1)^2}{2j_{\max}} \quad (4.44)$$

There are essentially two cases. First, the maximum jerk, acceleration, and velocity can be chosen in such a way that the maximum acceleration is reached at  $t_1$  and maintained till  $t_2$ . This case is illustrated in Figure 4.4. In that case the acceleration at point  $t_1$  is  $a(t_1) = a_{\max}$  and it thus follows

from equation 4.44:

$$v(t_1) = \frac{a_{\max}^2}{2j_{\max}} \quad (4.45)$$

The second case occurs when the maximum jerk, acceleration, and velocity are chosen in such a

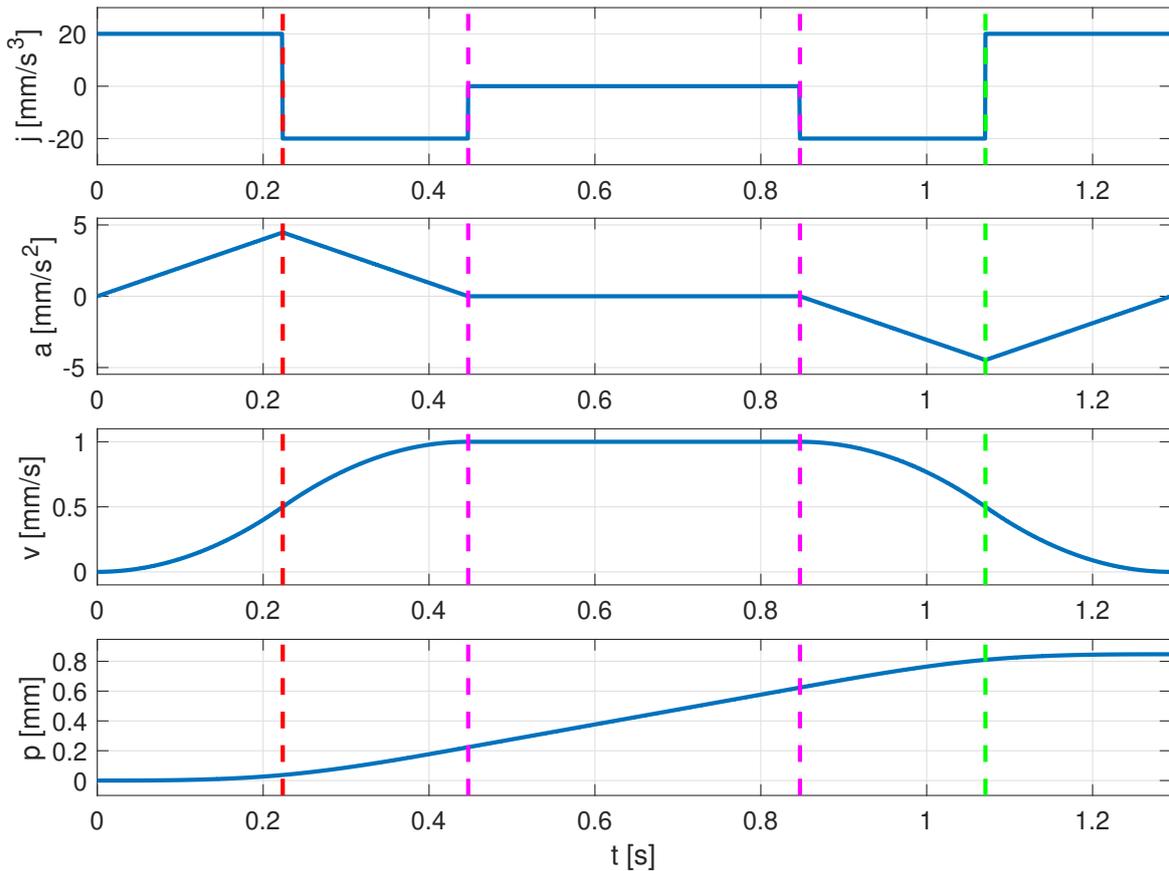


Figure 4.5: Illustration of an s-curve motion profile with very small jerk. The point of the maximum acceleration is marked green, and the end of the acceleration phase magenta. The period of constant velocity goes to the second magenta line, which is followed by a deceleration phase, which is the symmetrical opposite of the acceleration phase.

way that the maximum acceleration is reached and immediately goes down again, or cannot even be reached, as illustrated in Figure 4.5. In the second case the times  $t_1$  and  $t_2$  fall together and therefore split the acceleration phase precisely in half. Since both halves contribute the same velocity gain and since the velocity at the end of the acceleration phase is  $v_{\max}$ , the velocity at  $t_1$  is:

$$v(t_1) = v_{\max}/2 \quad (4.46)$$

Thus, a condition can be derived from 4.45 and 4.46:

$$v(t_1) < \frac{v_{\max}}{2} \quad (4.47)$$

$$\frac{a(t_1)^2}{j_{\max}} < v_{\max} \quad (4.48)$$

With the condition 4.48 and velocity equations 4.45 and 4.46  $v(t_1)$  can be expressed as follows:

$$v(t_1) = \begin{cases} \frac{a_{\max}^2}{2j_{\max}} & \frac{a(t_1)^2}{j_{\max}} < v_{\max} \\ v_{\max}/2 & \text{else} \end{cases} \quad (4.49)$$

With velocity equation 4.49 and time equation 4.43,  $t_1$  can be determined:

$$t_1 = \begin{cases} \frac{a_{\max}}{j_{\max}} & \frac{a(t_1)^2}{j_{\max}} < v_{\max} \\ \sqrt{\frac{v_{\max}}{j_{\max}}} & \text{else} \end{cases} \quad (4.50)$$

### Determination of the Second Interval

Equation 4.22 evaluated at  $t_2$  can be solved for  $t_2$ :

$$t_2 = t_1 + \frac{v(t_2) - v(t_1)}{a(t_1)} \quad (4.51)$$

There are again two cases: In the degenerate case where the maximum acceleration is not reached, the velocity  $v(t_1)$  immediately carries over into  $v(t_2)$ , and thus it follows that  $t_2 = t_1$ . In the regular case, the interval of constant acceleration (fraction in equation 4.51) needs to be determined. It follows from symmetry that the velocity contributions from the first acceleration interval  $(0; t_1]$  and the last acceleration interval  $(t_2; t_3]$  are the same. The velocity at  $t_3$  is  $v(t_3) = v_{\text{Meas}} = v_{\max}$ , it is therefore shy of  $v(t_1)$  at  $t_2$ :

$$v(t_2) = v_{\max} - v(t_1) \quad (4.52)$$

With equation 4.52 in 4.51, it follows for the normal case:

$$t_2 = t_1 + \frac{v_{\max} - 2 \cdot v(t_1)}{a_{\max}} = t_1 + \frac{v_{\max} - a_{\max} \cdot t_1}{a_{\max}} = \frac{v_{\max}}{a_{\max}} \quad (4.53)$$

Thus, combining both cases,  $t_2$  can be expressed using condition 4.48 in the following way:

$$t_2 = \begin{cases} \frac{v_{\max}}{a_{\max}} & \frac{a(t_1)^2}{j_{\max}} < v_{\max} \\ t_1 & \text{else} \end{cases} \quad (4.54)$$

### Determination of the Third and Fourth Intervals

It was already argued using symmetry that  $t_3 - t_2 = t_1$  (see equation 4.31). It thus follows immediately that:

$$t_3 = t_1 + t_2 \quad (4.55)$$

Furthermore, since the velocity is constant during the measurement and the measurement depth is known, the active measurement time can be determined (see equation 4.1). Therefore,  $t_4$  can be derived in the following way:

$$t_4 = t_3 + t_{\text{Meas}} = t_3 + \frac{d_{\text{Meas}}}{v_{\text{Meas}}} = t_3 + \frac{d_{\text{Meas}}}{v_{\text{max}}} \quad (4.56)$$

With equation 4.56, the end time of the active measurement has been determined. Since the sensor is only sensitive to vibrations in the interval  $(t_3; t_4]$ , the acceleration time  $t_3$  and the deceleration time (which due to symmetry is also of length  $t_3$ ) can be used to move the sensor and/or the part. This is described in greater detail in subsection 4.2.3.

### 4.2.2 Forward and Backward Capturing

The naive approach to WLI scanning is to always move the interferometer in the same direction (for instance, toward or away from the part). This would actually have two advantages:

1. Simple implementation.
2. No potential differences between forward and backward scans.

However, in that case, the sensor would have to travel all the way back to its initial position after a scan has been completed, before the next scan can be started. Even though this motion can be performed at a higher speed than the measurement motion, it would nevertheless take some unnecessary time. Therefore, a better scanning method is two-way capturing. This invokes several issues, which will be considered in this subsection.

As was mentioned in subsection 4.2.1, the active scan starts at the end of the acceleration phase (at  $t_3$ ), which is to say, at the point where the set velocity has been reached and remains constant. The interferometric camera itself has a hardware trigger input that is connected to the linear axis. The axis can be programmed to emit a trigger pulse whenever a chosen axis position has been crossed from a defined direction, either up or down. As soon as the camera gets triggered, it starts capturing and processing with the pre-programmed parameters (see section 4.1), independently of the axis's actual motion. This means that the trigger point has to be chosen appropriately in order to ensure that the requirements for a successful scan, especially constant velocity, are met.

Due to symmetry, the motion profile looks identical coming from either direction, with the upward acceleration interval being equal to the downward deceleration interval. The trigger points in a two-

way scanning scheme thus have to be set at the beginning of the constant velocity phase for either direction, as illustrated by the two magenta lines in Figures 4.4 and 4.5, with the left trigger point having the upward and the right trigger having the downward condition. Since the trigger points are symmetrical about the center of the motion profile, the center is chosen as a virtual operating point  $OP$ . The downward and upward trigger points are then simply the operating point  $\pm$  half the active measurement range, respectively:

$$TP_D = OP - \frac{d_{Meas}}{2} \quad (4.57a)$$

$$TP_U = OP + \frac{d_{Meas}}{2} \quad (4.57b)$$

The axis controller has no information on when the interferometric camera requires constant velocity. It only knows the start and endpoints and performs its trajectory under the set constraints maximum jerk, acceleration, and velocity. Therefore, the axis has to be brought to the correct starting point. The appropriate trigger point has to be calculated and set either upward or downward, see equation 4.57. Finally, the axis is moved to the end position with the determined motion parameters. The beginning and end position are again symmetrical about the operating point and are  $\pm$  half the length of the measurement depth:

$$Start_D = OP - \frac{d_{Acc}}{2} - \frac{d_{Meas}}{2} = OP - \frac{v_{max}}{2a_{max}} - \frac{a_{max}}{2j_{max}} - \frac{d_{Meas}}{2} \quad (4.58a)$$

$$Start_U = OP + \frac{d_{Acc}}{2} + \frac{d_{Meas}}{2} = OP + \frac{v_{max}}{2a_{max}} + \frac{a_{max}}{2j_{max}} + \frac{d_{Meas}}{2} \quad (4.58b)$$

Unfortunately, the axis trigger is subject to a certain amount of dead time. During that time, the axis continues to move, which means that the camera actually starts recording interferometric signals at a certain distance from the trigger point. This leads to different measurement volumes for forward and backward scans because the effect goes in the opposite direction, as illustrated in subfigures (c) and (d) in Figure 4.6 on the following page. Since the dead time affects the start of the measurement in the constant velocity phase, the distance scales linearly with the time:

$$\Delta d = t_{Dead} \cdot v_{Meas} \quad (4.59)$$

This distance has to be subtracted from the lower and added to the upper start and trigger points, as shown in Figure 4.6. In order for the virtual operating point to be effective, two requirements should be met: there should be no discernible difference between forward and backward scans, and the captured data should be shifted into a coordinate system that is centered around the operating point. The actual interferometer outputs are the demodulated intensities caused by the interferometric correlation process (see Appendix A), which are related to points in time. Thus, the interferometer starts capturing at a set trigger point and starts recording for a preset time. Without additional information, for instance, from the axis controller, this means that the beginning of the measurement process also marks the zero

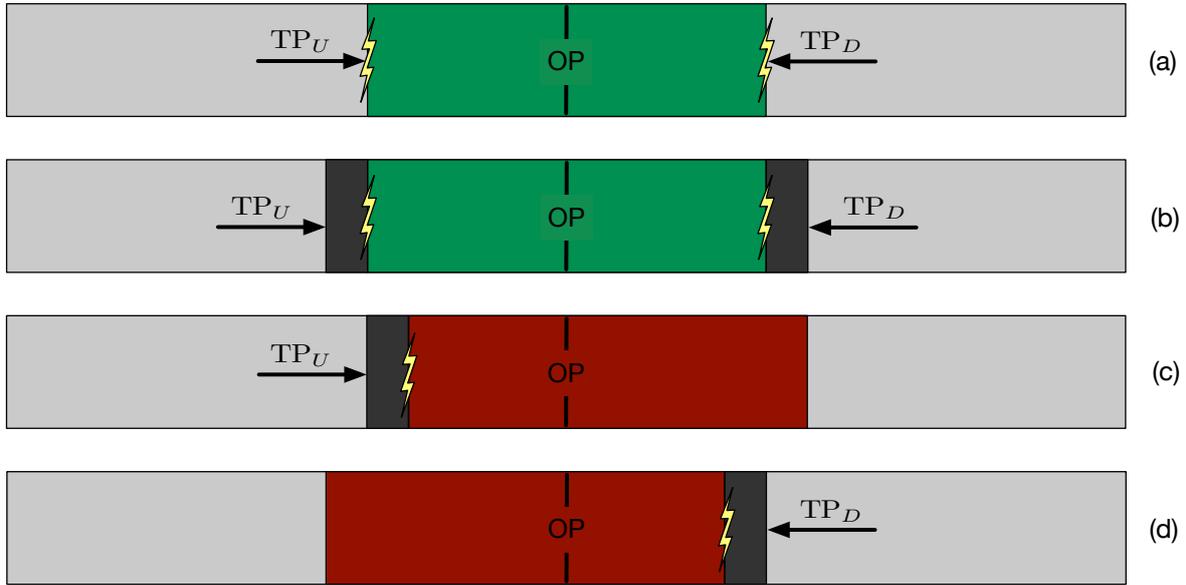


Figure 4.6: Influence of the trigger dead time. Subfigure (a) shows the ideal behavior for a system without dead time and (b) shows how the issue is dealt with. Subfigures (c) and (d) illustrate the different measurement volumes if the dead time is not compensated.  $TP_U$  is trigger up,  $TP_D$  is trigger down, and  $OP$  is the virtual operating point.

position, which would, of course, be quite different if the same measurement is performed from the other direction. For example, Figure 4.7 on the following page illustrates what would happen if the interferometer were to scan a sphere that is positioned to be in the last third of the measurement range. During the forward scan, the interferometer would at first capture the front part of the sphere, and as it travels forward, it would continue to scan parts of the sphere that are progressively more toward the sphere's equator. In the  $z$ - $y$  (or  $z$ - $x$ ) plane, this would result in a partial circle, as shown in the upper picture in Figure 4.7. However, if the scan is performed backward (with everything else being the same), the scanner would at first see parts of the sphere that are closer to the equator as it moves away from the sphere and progressively scans parts that are closer to the tip, as shown in the lower picture in Figure 4.7. In both cases, the coordinate system is not centered.

In order to shift the forward data into the correct local coordinate system, it is sufficient to subtract half the measurement range ( $d_{\text{Meas}}/2$ ). To get the correct starting value on the axis, the virtual trigger point is added. Since the linear axis should also have its coordinate system centered, finally, the center point  $CP$  has to be subtracted. Since the given axis has increasing values with increasing lift, the overall coordinate direction has to be inverted, resulting in the following expression for all upward distance values:

$$x_{\text{Up}} = (CP - OP) + \left( \frac{d_{\text{Meas}}}{2} - x_{\text{Raw}} \right) \quad (4.60)$$

The conversion for the backward scans is the same, except that the values have to be flipped in the  $z$  direction by subtracting all values from the measurement depth  $d_{\text{Meas}}$ . The expression for downward

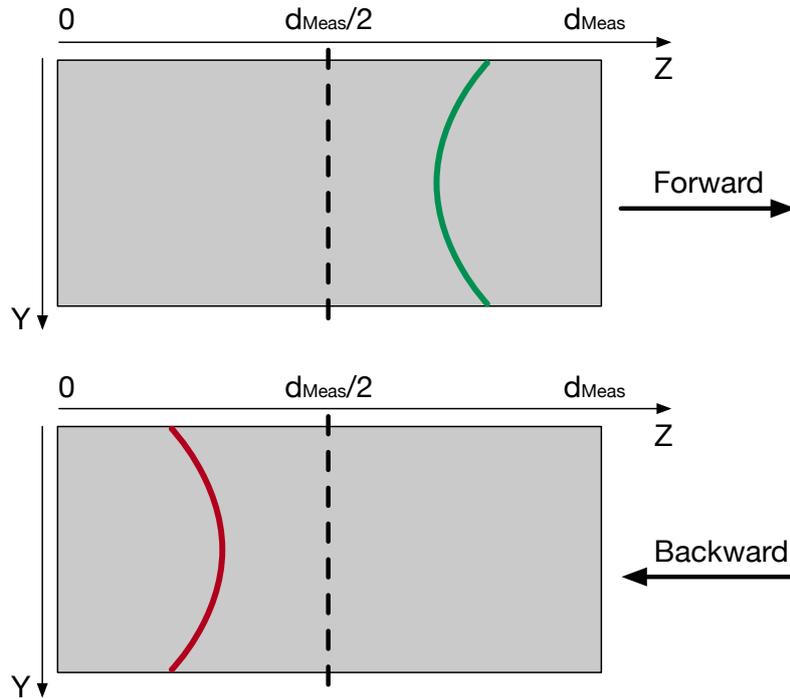


Figure 4.7: Illustration of the unshifted interferometer output of a scanned sphere in the  $z$ - $y$  plane for forward and backward scans respectively.

values is:

$$x_{\text{Down}} = (\text{CP} - \text{OP}) + \left( -\frac{d_{\text{Meas}}}{2} + x_{\text{Raw}} \right) \quad (4.61)$$

The mappings in equations 4.60 and 4.61 are illustrated in Figure 4.8.

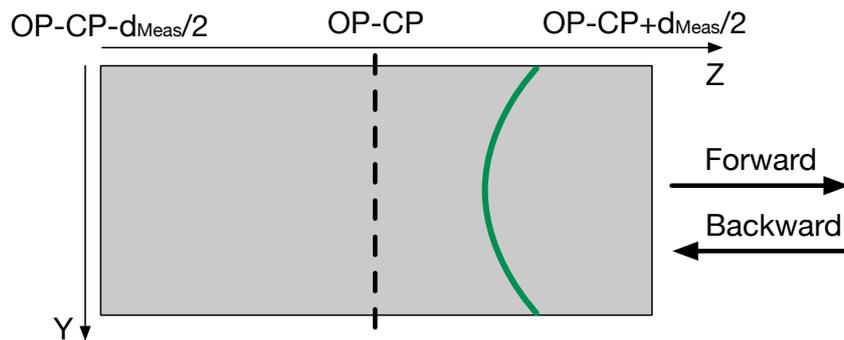


Figure 4.8: Illustration of the correctly shifted interferometer output of a scanned sphere in the  $z$ - $y$  plane for both forward and backward scans.

The way the shifting is done allows for intuitive path adaptation (see next subsection) because the deltas can be directly applied to the zero point (center of the axis).

### 4.2.3 Scan Adaptation and Parallelization

Part of the novel method is the ability to adapt the measurements to the current part. The difference detection between the master part and the current part is performed by the IFPT's software AutoXIV as part of the robot path planning algorithm. The interferometer then has to perform the one degree of freedom (1-DOF) adaptation movement.

The most straightforward but also most inefficient implementation of the adaptive scanning process would be the following: AutoXIV sends a TCP/IP package containing the *current* adaptation value. The interferometer controller (AutoSense) then moves the axis to the requested value and performs the scan. After the full axis motion has been performed, and the data was transmitted from the sensor to the computer, AutoSense informs AutoXIV that it is ready for the next scan. AutoXIV then moves the part and/or the robot to the next measurement position and sends the next adaptation value to AutoSense.

Because the capturing is effectively finished before the scanning motion ends, the described process can be modified to allow for increased parallelization. The idea is to inform the path adaptation software that the next measurement can be started at the end of the capturing process (see the second magenta line in Figure 4.4 on page 66) instead of at the end of the motion. This allows AutoXIV to move the part and/or robot while the interferometer completes its motion and transmits the captured data to the computer. While this provides increased parallelization, this solution is not yet ideal because the adaptation movement is performed only after AutoXIV issues the next measurement command, which is not necessarily at the earliest possible time, because of the acceleration phase.

In order to take advantage of the remaining potential, the command order has to be changed so that not the current but the next adaptation value is transmitted with each scan command. This means that the very first scan requires a special *initial adaptation* command, which brings the sensor into its initial adapted measurement position but does not trigger a scan. Afterward, whenever a scan motion is completed, the sensor moves to the requested next adaptation position as soon as possible.

Each adaptation command is relative to the virtual operating point and, therefore, independent of the sensor's scanning direction. There are essentially three adaptation cases to consider:

1. The virtual operating point did not change.
2. The virtual operating point changed in the direction of the current scan.
3. The virtual operating point changed in the other direction.

In the first case, the scan can be performed exactly as described in the previous subsection. In the second case, the normal scanning approach is modified slightly by changing the endpoint of the motion to the *next* operating point plus half the range of motion. This means that the axis will travel the difference between the current and the next operating point in addition to the required range of motion. This approach is ideal because no additional linear axis commands have to be issued, and the axis motion profile can be used more efficiently because it stays longer at its maximum velocity. In the

third case, the normal scan has to be performed in order to ensure that the full measurement range is covered at a constant velocity. Afterward, an adaptation motion is performed in the opposite direction. For instance, if the part contains a section with a downward slope, this means that the adaptation value would change in every scan. In that particular case, every forward scan can use the more efficient approach of simply continuing the motion, while every backward scan would have to finish the regular motion and then move forward slightly.

Unfortunately, the described motion considerations are not the only factors, which can introduce undesirable waiting times. Depending on the chosen camera parameters, the post-processing in the interferometric camera (surface extraction) can take significantly longer than the deceleration phase. Fortunately, the camera contains two buffers in its RAM. During capturing, the data is sent to the first buffer. Once the scan is completed, an API call transfers the captured stack of matrices to the second buffer, where it is processed by the FPGA. Thus, if synchronized correctly, the next capturing process can be started *before* the previous scan has been fully processed. This allows to compensate long processing times up to a certain point so that the process can be largely parallelized for appropriate parameter sets. The simplified adaptive scanning process is summarized as a UML activity diagram in Figure 4.9 on the following page.



## 4.3 White Light Interferometer Pre-Processing

The surface data captured by the interferometer requires several pre-processing steps, which are discussed in the following subsections.

### 4.3.1 Distortion Compensation

Because the interferometer is an optical system, certain distortions are present, which need to be compensated for. In this context, there are two relevant types of distortions, which act radially (pin cushion or barrel distortion) and vertically.

The radial distortion can be determined by scanning a well-known calibration target. The target's features are subsequently extracted, and the positions are fitted to the target's model. This can be used to solve for the distortion parameters, which are then used to generate a lookup table for the corrected  $x$  and  $y$  values. This process is known as camera calibration [GT11, pp. 42].

The vertical distortion, on the other hand, can be determined by scanning a very flat object such as a high-quality surface mirror. Due to the distortion, the object will appear bowl-shaped instead of perfectly flat. A two-dimensional polynomial of first order is fitted to the data, resulting in a plane, which is subtracted from the input data. This centers the data around zero and removes any gradient. Afterward, a two-dimensional polynomial of third or fourth order can be fitted to the data, resulting in a smooth representation of the distortion. This polynomial can either be used to generate a lookup table, which is the recommended approach, or to calculate the correction data on demand. Since the correction data is in the distance domain, it can simply be subtracted from the incoming raw data.

### 4.3.2 Invalid Data Detection

Due to the way the scanning procedure works, a signal is always recorded in every frame for every pixel. If there is no surface within range of a specific pixel, only noise is recorded. The envelope extraction and maximum detection algorithms can be applied to such a noise signal, even though the resulting position/amplitude tuple is meaningless because no surface was actually measured. This means that in areas that did not measure a valid surface, the scan contains random data, rather than being empty or marked as invalid. Figure 4.10 on the following page shows the software AutoSense, the white light interferometer scanning software developed for this dissertation, scanning a crack. On the left side is a false-color-coded depth image and on the right side is a logarithmically scaled amplitude image. It can be observed that the image contains meaningless depth data (left side) in the lower part of the crack, where the amplitude is black (right side).

The consequence is that the raw depth data cannot be used directly because it cannot be “trusted.” The solution is to validate the depth data using the amplitude data. However, this is not as straightforward as simply applying a threshold, due to the non-ideal lens characteristic of the sensor. More precisely, there is a radial decrease in intensities, which means that direct thresholding would overly affect outside pixels. In order to compensate for this issue, an intensity baseline scan is created by per-

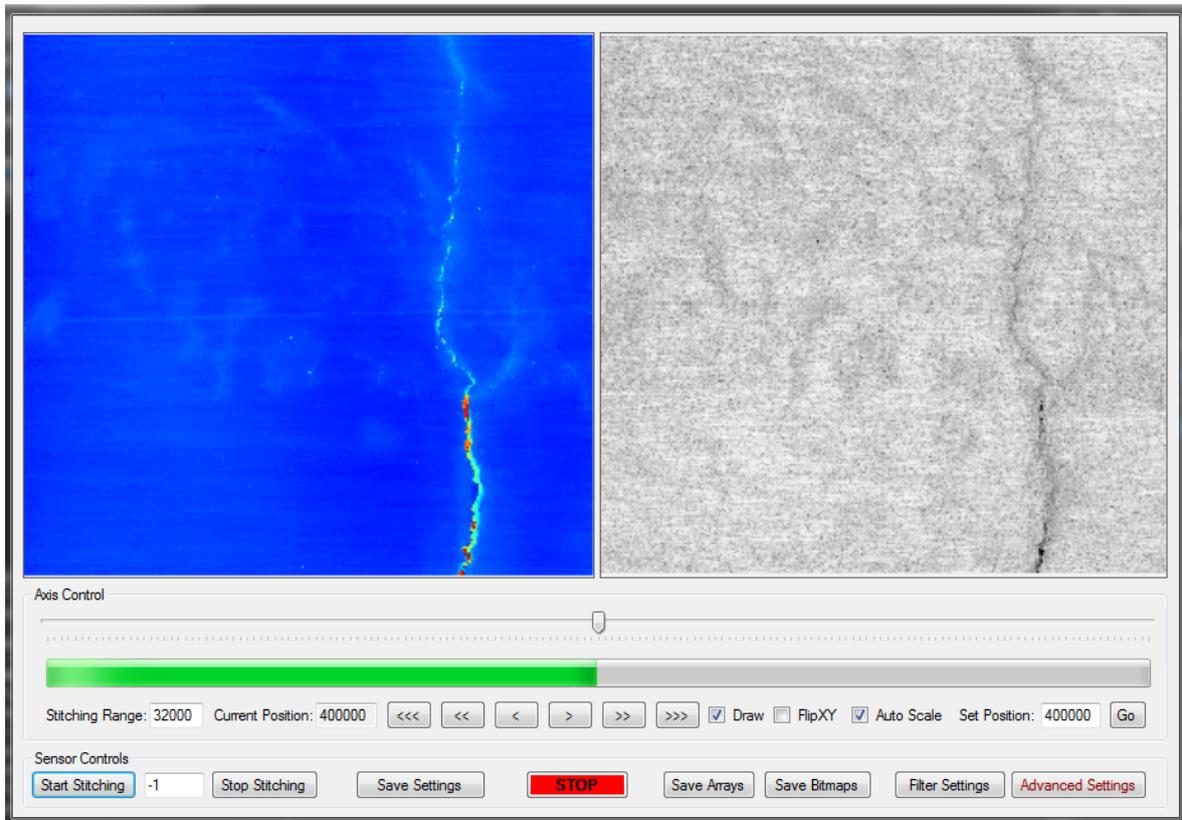


Figure 4.10: Screenshot of the software AutoSense (developed for this dissertation) scanning a crack.

forming a scan in empty space. Afterward, a two-dimensional polynomial regression is performed to find a model for the intensity distribution, as illustrated in Figure 4.11. This approach is valid because even though the resulting intensities in each pixel have a random intensity, they are distributed about a baseline depending on the pixel's response and the lens's properties in that area. In other words, despite the complex underlying scanning algorithm, the pixels in an empty scan still represent the general intensity distribution. Since the number of samples is typically far larger than the number of parameters that need to be determined, the individual errors will not have a significant impact on the overall result.<sup>1</sup> The resulting polynomial can then be used to generate a lookup table for the intensity distribution. Figure 4.11 shows such a result for a third-order two-dimensional polynomial.

In order to use this data to compensate for the lens attributes, the resulting data points need to be normalized and inverted. First, the maximum intensity point should not be affected at all. Therefore, all data points have to be divided by the maximum value, which effectively scales all points to be a percentage of the maximum intensity. Afterward, every point in the mask needs to be inverted. Thus,

<sup>1</sup>Since every pixel represents a separate random process in which the resulting intensity levels are themselves caused by different random processes, the errors can be expected to be distributed normally, according to the central limit theorem [Lof90, pp. 191].

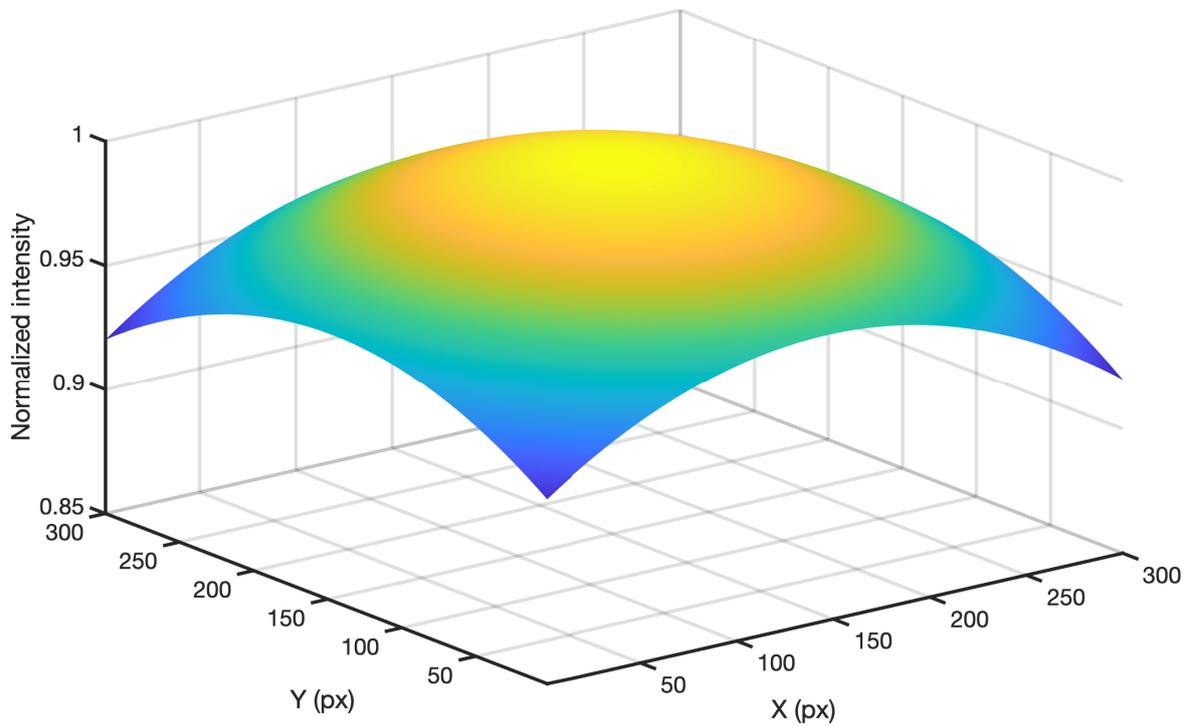


Figure 4.11: Illustration of the intensity regression (third-order two-dimensional polynomial).

for every point  $p$  in the set of data points  $P$ , the following overall operation needs to be performed:

$$p_{\max} = \max(P) \quad (4.62)$$

$$p_{\text{mask}} = \frac{1}{p/p_{\max}} = \frac{p_{\max}}{p}, \forall p \in P \quad (4.63)$$

The mask created by applying equation 4.63 to the baseline scan can simply be multiplied with every corresponding intensity point, resulting in an equalized intensity response. Afterward, invalid pixels can be determined by simple thresholding.

After the steps performed in this section, the data-scanning part has been completed and the collected data can be processed, as described in the following chapter.

## Chapter 5

# Crack Detection in Airplane Combustion Chamber Liners

In this chapter, the specific method for detecting cracks in airplane combustion chamber liners that have been scanned with the white light interferometer-based approach (introduced in the previous chapters) will be discussed. As mentioned before, the general approach is to split the unmanageably large scan collection into subsets, then to combine those subsets into scan matrices and finally to reduce their dimensionality to allow fast and robust crack detection. This approach is summarized in Figure 5.1. Whereas some of the approaches introduced in this chapter are universal, others are

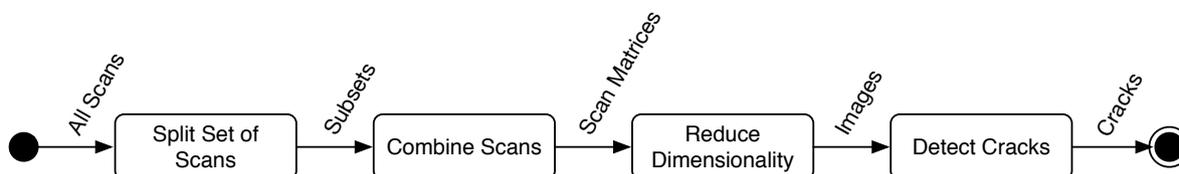


Figure 5.1: General crack-detection approach.

partially specific to combustion chamber liners (see Figure 1.2 on page 4) or at least to parts with approximate rotational symmetry. For instance, the splitting approach takes the specific shape, rotational symmetry and natural structure of the part into account, as discussed in the following section.

## 5.1 Patch Splitting

This section describes how the scan collection is split into subsets. The subsets should be small enough to be readily processable, yet large enough to contain significant amounts of data. Furthermore, it is desirable to use natural structures, because this makes it easier to reference damaged areas, for instance when displaying the results to repair technicians. For this reason, the height of the individual subsets is chosen to be equal to the height of the airplane combustion chamber liners' panels.

The angular extent of the subsets should be chosen in such a way that the curvature of the part does not play a significant role. This is vital to the basic image-processing approach presented later in this dissertation, because it is based on the assumption that the background shape can be removed without introducing artifacts. In order to keep the segment size manageable and intuitive and to allow for proper linearization, the angular extent was chosen such that the width of each subset is close to that of its height. It was also decided to perform the splitting on the scan and not the point level, meaning that individual scans are not broken up. The incoming scans are already split into raw panels

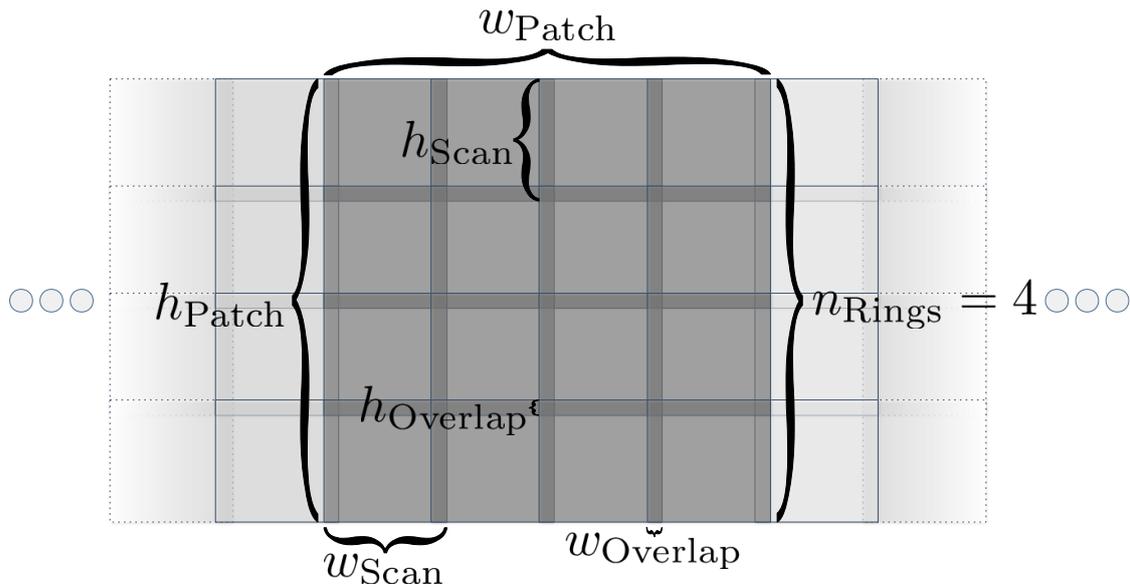


Figure 5.2: Illustration of the components required for patch splitting:  $n_{\text{Rings}}$ ,  $h_{\text{panel}}$ ,  $h_{\text{Patch}}$ ,  $h_{\text{Scan}}$ ,  $h_{\text{Overlap}}$ ,  $w_{\text{Patch}}$ ,  $w_{\text{Scan}}$ , and  $w_{\text{Overlap}}$ . The dark-shaded area is one patch, but the four rings continue until they close the loop.

using metadata; hence, each scan ring is associated with a specific panel identifier. Each panel contains a certain number of rings, see  $n_{\text{Rings}}$  in Figure 5.3. Since the scans are approximately quadratic and the horizontal and vertical overlaps are approximately equal, the splitting can be accomplished by dividing the number of scans per ring by the number of rings. If there was no overlap between the scans, the resulting patch height  $h_{\text{Patch}}$  (which is the same as the panel height  $h_{\text{panel}}$ ) would be the height of the scans  $h_{\text{Scan}}$  multiplied by the number of scans:  $h_{\text{Patch}} = n_{\text{Rings}} \cdot h_{\text{Scan}}$ . However, since the scans are overlapping, each scan except the first one is shifted up by the height of the overlap

$h_{\text{Overlap}}$  instead. This is illustrated in Figure 5.3. Since the first scan is not affected, the shift happens  $(n_{\text{Rings}} - 1)$  times. Therefore, the full height is shorter by that amount:

$$\begin{aligned} h_{\text{Patch}} &= n_{\text{Rings}} \cdot h_{\text{Scan}} - (n_{\text{Rings}} - 1) \cdot h_{\text{Overlap}} \\ &= n_{\text{Rings}} \cdot (h_{\text{Scan}} - h_{\text{Overlap}}) + h_{\text{Overlap}} \end{aligned} \quad (5.1)$$

The overlap is much smaller than the width or height of the scans and considering that the width of the scan is approximately the height of the scans. Hence, equation 5.1 can be simplified slightly through approximation:

$$h_{\text{Patch}} \approx n_{\text{Rings}} \cdot (h_{\text{Scan}} - h_{\text{Overlap}}) \quad (5.2)$$

The number of scans in a patch in the horizontal direction can be determined by dividing the number of scans in a ring by the number of patches. For the width  $w_{\text{Patch}}$  of the patches it can then be argued similarly as for the height:

$$w_{\text{Patch}} \approx \frac{n_{\text{ScansPerRing}}}{n_{\text{Patches}}} \cdot (w_{\text{Scan}} - w_{\text{Overlap}}) \quad (5.3)$$

Because the height and width of the patches should be the same, equations 5.2 and 5.3 can be combined as follows:

$$n_{\text{Rings}} \cdot (h_{\text{Scan}} - h_{\text{Overlap}}) \approx \frac{n_{\text{ScansPerRing}}}{n_{\text{Patches}}} \cdot (w_{\text{Scan}} - w_{\text{Overlap}}) \quad (5.4)$$

With  $w_{\text{Scan}} = h_{\text{Scan}}$  and  $w_{\text{Overlap}} = h_{\text{Overlap}}$ , equation 5.4 can be solved for the number of patches:

$$\begin{aligned} n_{\text{Rings}} \cdot (h_{\text{Scan}} - h_{\text{Overlap}}) &\approx \frac{n_{\text{ScansPerRing}}}{n_{\text{Patches}}} \cdot (h_{\text{Scan}} - h_{\text{Overlap}}) \\ \Leftrightarrow n_{\text{Rings}} &\approx \frac{n_{\text{ScansPerRing}}}{n_{\text{Patches}}} \\ \Leftrightarrow n_{\text{Patches}} &\approx \frac{n_{\text{ScansPerRing}}}{n_{\text{Rings}}} \end{aligned} \quad (5.5)$$

Thus, the number of approximately quadratic patches can be found by splitting the scans in each ring into groups of  $n_{\text{Rings}}$ , resulting in  $n_{\text{Patches}}$  patches, see equation 5.5.

However, simply splitting the panel with equation 5.5 only works with fully filled rings. This is not always the case, for instance when only a certain area of the part is to be inspected. Therefore, a more general splitting approach uses the cylindrical nature of the part and splits the panel using an angular criterion. These calculations can be most easily performed by transforming the cartesian

representation into cylindrical coordinates:

$$\rho = \sqrt{x^2 + y^2} \quad (5.6a)$$

$$\theta = \text{atan2}(y, x) \quad (5.6b)$$

$$z = z \quad (5.6c)$$

At first, the difference angle between the start and end scans has to be calculated. Since the end angle is not necessarily larger than the start angle, it needs to be corrected as follows:

$$\tilde{\theta}_{\text{End}} = \begin{cases} \theta_{\text{End}} & \text{for } \theta_{\text{End}} > \theta_{\text{Start}} \\ \theta_{\text{End}} + 2\pi & \text{else} \end{cases} \quad (5.7)$$

The difference angle is then:

$$\Delta\theta = \tilde{\theta}_{\text{End}} - \theta_{\text{Start}} \quad (5.8)$$

Using the number of patches (equation 5.5), the angular extent of a patch can be calculated as follows:

$$\theta_{\text{Patch}} = \frac{\Delta\theta}{n_{\text{Patches}}} \quad (5.9)$$

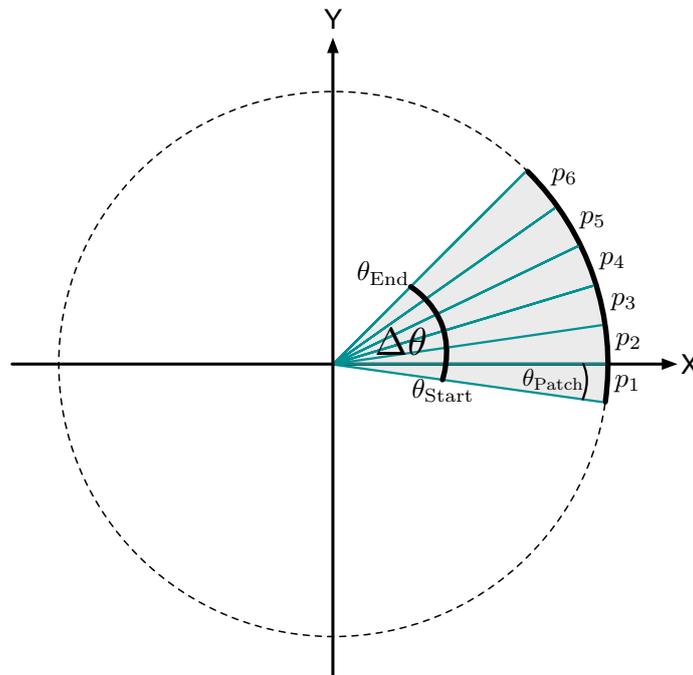


Figure 5.3: Illustration of patch splitting with an incomplete arc.

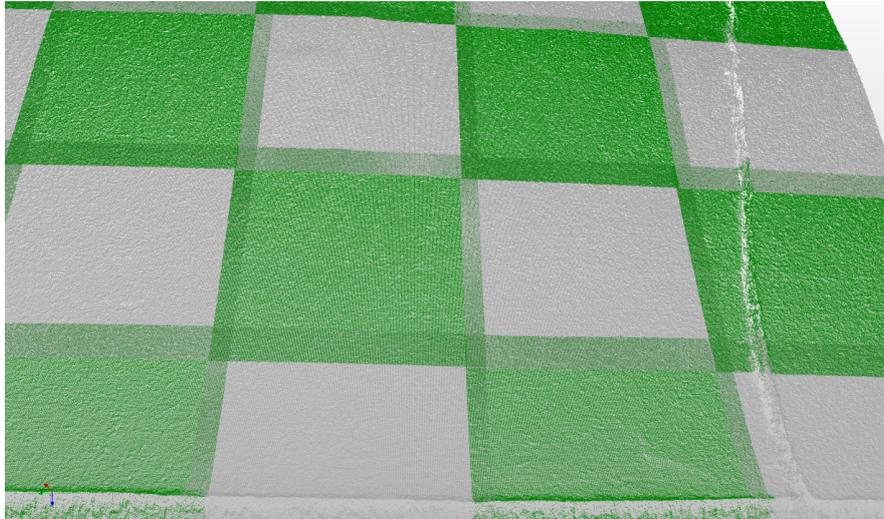
With equation 5.9, it is possible to determine the actual cutoff angles for every patch with patch ID  $p_i$  directly, using its ordered index  $i$ :

$$\theta_{\text{CutOff},i} = \theta_{\text{Start}} + i \cdot \theta_{\text{Patch}} \quad (5.10)$$

The actual splitting can then be performed in two ways: 1) the angles of every scan in every ring are checked against the calculated cutoff angles; 2) given that the number of scans per ring is kept constant within a panel, by calculating the cutoff scan IDs for the first ring and applying the determined indices to all rings in that panel. The latter approach is preferable, because significantly fewer calculations have to be performed. The angular patch-splitting approach is illustrated in Figure 5.3.

## 5.2 Scan Registration

After the scan collection is split into suitable scan subsets, the individual scans have to be combined into matrices that contain all relevant data. As mentioned in subsection 3.2.1, the sensor is TCP-calibrated, meaning that each scan can be transformed into a common coordinate system for arbitrary actuator configurations. This means that each scan can be transformed into roughly the correct spot



*Figure 5.4: Collection of scans before applying the registration algorithm. The overlaps are emphasized by manually colorizing the individual scans in a checkerboard pattern.*

in a common coordinate frame, as shown in Figure 5.4. However, since industrial robots are highly non-linear actuators, it cannot be assumed that the sensor TCP calibration is ideal, especially not in all kinematic configurations. Therefore, the transforms of the individual scans are error-prone and need to be corrected. This could, for instance, be done by finding a much better approximation of the forward transform, e.g., by using a laser tracker, or by correcting the data afterward with an overlap-based registration algorithm. Since it is desirable to keep the hardware cost as low as possible, the second approach was chosen. As was already alluded to, the algorithmic approach implies that there has to be

sufficient overlap between neighboring scans within one scan subset, otherwise the use of registration techniques is severely limited. This has the additional benefit of ensuring that the scan coverage of the part is complete.

One way of describing coordinate system transforms is with homogenous transform matrices. For three dimensions in a transform from  $a$  to  $b$ :

$${}^b_a\mathbf{T} = \begin{bmatrix} \mathbf{R} & \underline{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix} \quad (5.11)$$

where  $\mathbf{R}$  is an orthonormal  $[3 \times 3]$  matrix describing the rotation and  $\underline{t}$  is a vector of length three describing the translation. The zeros underneath the first three column vectors signify that they are free vectors, whereas the one underneath the last column vector makes it a position vector. For more information on coordinate system transforms, see for instance [Web09, pp. 29].

Registration techniques generally rely on recognizable features that are present in both scans. In point-based approaches, the registration algorithm tries to find a transform that aligns the  $N$  point pairs  $\underline{p}_i$  and  $\underline{q}_i$  as closely as possible, in a least-squares sense:

$$\arg \min_{\mathbf{R}, \underline{t}} \frac{1}{N} \sum_{i=1}^N \|\underline{p}_i - \mathbf{R} \cdot \underline{q}_i - \underline{t}\|_2^2 \quad (5.12)$$

### 5.2.1 Iterative Closest Points (ICP) Algorithm

If the mapping of corresponding features was known, the minimization problem in equation 5.12 could be solved in closed form in a single step using the Kabsch algorithm, the following description of which is based on [SR17]: First, the centroids<sup>1</sup> of the point sets  $Q$  and  $P$  are calculated:

$$\underline{c}_q = \frac{1}{n} \cdot \sum_{i=1}^n \underline{q}_i \quad (5.13a)$$

$$\underline{c}_p = \frac{1}{n} \cdot \sum_{i=1}^n \underline{p}_i \quad (5.13b)$$

Because the space under consideration is three-dimensional, each point, and therefore also the centroids, can be written as  $[3 \times 1]$  column vectors. The sets of points can therefore be written as matrices containing  $n$  of those column vectors, resulting in  $[3 \times n]$  matrices:

$$\mathbf{Q} = [\underline{q}_1, \underline{q}_2, \dots, \underline{q}_n] \quad (5.14a)$$

$$\mathbf{P} = [\underline{p}_1, \underline{p}_2, \dots, \underline{p}_n] \quad (5.14b)$$

With the  $[3 \times 1]$  centroids calculated according to equation 5.13b and the matrix representation of

<sup>1</sup>In probabilistic terminology, this would be called the mean or first moment.

the point sets from equation 5.14b, the points can be shifted to the origin:

$$\mathbf{Q}_0 = [\underline{q}_1 - \underline{c}_q, \underline{q}_2 - \underline{c}_q, \dots, \underline{q}_n - \underline{c}_q] \quad (5.15a)$$

$$\mathbf{P}_0 = [\underline{p}_1 - \underline{c}_p, \underline{p}_2 - \underline{c}_p, \dots, \underline{p}_n - \underline{c}_p] \quad (5.15b)$$

With the shifted point sets in equation 5.15b, the covariance matrix can be determined (see also [Lof90, pp. 157]):

$$\mathbf{C}_{Q,P} = \mathbf{Q}_0 \cdot \mathbf{P}_0^T \quad (5.16)$$

Since  $\mathbf{Q}_0$  is a  $[3 \times n]$  matrix and  $\mathbf{P}_0^T$  is an  $[n \times 3]$  matrix, the covariance matrix has a dimension of  $[3 \times n] \cdot [n \times 3] = [3 \times 3]$ . The covariance matrix contains information on correlations between the vectors from which it was calculated. That means it also contains information about the rotation between the point sets. Therefore, a singular value decomposition (SVD) is performed (see for instance [Fis17, pp. 448] [WRR03, pp. 1]) on the covariance matrix in 5.16 in order to extract this information. The SVD decomposes the matrix into three matrices:

$$\text{SVD}(\mathbf{C}_{Q,P}) = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^* = \mathbf{C}_{Q,P} \quad (5.17)$$

The matrices  $\mathbf{U}$  and  $\mathbf{V}^*$  describe rotations, whereas  $\mathbf{S}$  describes a scaling. Thus, to obtain only the inverse rotation, the effects of the rotational matrices have to be reversed:

$$\mathbf{R} = \mathbf{V} \cdot \mathbf{U}^* \quad (5.18)$$

Unfortunately, the resulting rotation matrix does not necessarily describe a right-handed coordinate system. This can be checked by calculating the determinant of the rotation matrix. If the determinant is smaller than zero ( $|\mathbf{R}| < 0$ ), the rotation matrix is left-handed. This can be rectified by inverting the  $z$  direction, which is done by multiplying the third column with  $-1$ .

With the rotation matrix determined, the translation vector can be calculated from the centroids. The centroid  $\underline{c}_q$  describes the translation of the center of the point set  $Q$  from the origin  $\underline{0}$ . Therefore, the vector  $-\underline{c}_q$  describes the translation from the center of  $Q$  to the origin. The rotation matrix  $\mathbf{R}$  describes a rotation that aligns the point set  $Q$ 's coordinate system with  $P$ 's. The centroid  $\underline{c}_p$  finally describes the translation from the origin  $\underline{0}$  to the center of the point set  $P$ . Combining the expressions yields the overall translation:

$$\underline{t} = \mathbf{R} \cdot (-\underline{c}_q) + \underline{c}_p = \underline{c}_p - \mathbf{R} \cdot \underline{c}_q$$

The rotation and translation can then be combined into a  $(4 \times 4)$  homogenous transform matrix, see equation 5.11. The transform determination steps are illustrated in Figure 5.5.

However, in most real-world examples, the correspondences between point sets are unknown.

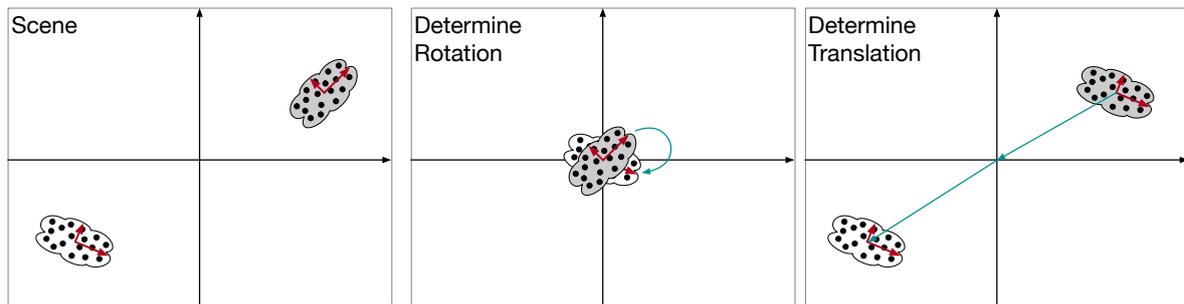


Figure 5.5: Illustration of the transform determination steps.

A solution for point cloud registration with unknown correspondences was introduced by Besl and McKay in 1992 as the iterative closest points algorithm (ICP) [BM92]. The basic idea is to find the nearest neighbor in the second set for each point in the first set and use the result as correspondence candidates. Next, the transform that minimizes the distances between the nearest neighbors is calculated. Since the nearest neighbors were not necessarily the correct correspondences, an optimum may not have been found after one iteration. Therefore, a set of pre-defined criteria (e.g., number of iterations, size of sum-of-squared errors, or change in the resulting pose from the previous iteration) has to be checked. If the criteria are not met, the process is repeated. The correspondences assigned in the first iteration are discarded and are reassigned in each iteration. The naive nearest neighbor search is computationally expensive. Therefore, the use of the space-partitioning data structure “k-d tree” is state of the art, as introduced by D. Simon in his dissertation [Sim96, pp. 28].

An issue with the ICP algorithm is that it is not guaranteed to yield the global optimum. Without special precautions, the algorithm might become trapped in a local minimum. The author found that this is especially true with data that does not have a significant amount of features, as is often the case with data captured by the proposed combustion chamber liner inspection approach. Since the overlap is relatively small and the features often do not stand out significantly, the use of the standard ICP algorithm often leads to local optimum solutions. These may satisfy the minimum distance criteria in the overlap but not the global goal of generating a continuous surface over the course of several scans. This was often due to the algorithm’s manipulation of “illogical” degrees of freedom. Thus, a different approach using a non-linear solver had to be devised.

## 5.2.2 Non-Linear Iterative Closest Points

An alternative approach to the linear algebra-based Kabsch algorithm (summarized in the previous subsection) would be to use a solver to directly minimize the cost function in equation 5.12. This could be accomplished with a robust yet fast non-linear solver like the Levenberg-Marquardt (LM) solver (see for instance [Ste17, pp. 81]). In practice, such a solver has several iteration control parameters. It is passed a function pointer, the optimization core, which takes the parameters that are to be minimized and returns a set of residuals. The function itself has access to the data points that are subject to minimization. In addition, the solver needs an initial guess as a starting point for the iterations. An

effective approach to setting up the function would be to expect one error for each point: the squared distance from its nearest neighbor, after applying the current transform iteration.

The solver program takes the master point cloud and builds a k-d tree from it. In this particular case, prior knowledge exists for the pose due to the robot's forward transform and the calibrated sensor; this is used as the initial guess for the solver parameterization. In the first iteration, that pose is applied to the second point cloud. The point cloud that is to be moved can then find its nearest neighbors using the tree. The residuals can be calculated from the squared distances. The solver will then iterate over the parameters to try to improve upon the initial guess. Under appropriate conditions, the solver will eventually reach a local or even global minimum.

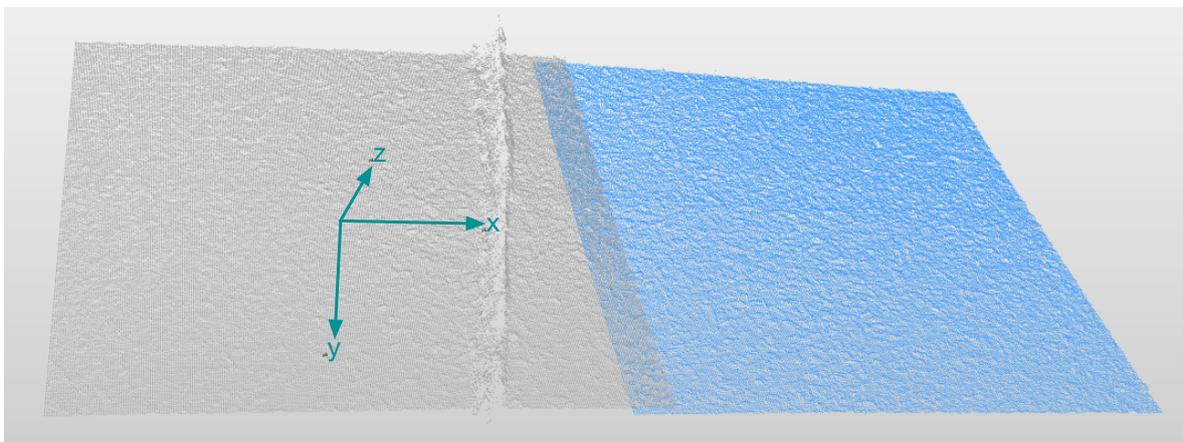


Figure 5.6: Two overlapping scans. The first scan's coordinate system is illustrated in green.

This approach in and of itself is not an improvement over the linear algebra approach. However, the advantage of the solver method is that the optimization core can readily be adapted to perform more complicated calculations and thus to consider constraints. More importantly, the minimization procedure can be adapted to lock DOF. A scan's coordinate system and the situation of two overlapping scans is shown in Figure 5.6.

As mentioned in the previous subsection, the scans do not necessarily contain significant features. This means that an unmodified ICP might continue to move the point cloud in the  $x$ - $y$  plane even though the executed shift may be far too large. One solution to prevent this would be to completely lock the DOF  $x$  and  $y$ . Particularly in the  $x$  direction, this is uncritical because it aligns with the tangent direction of the rotational axis movement and therefore can be expected to introduce relatively small errors. The  $y$  direction is more error-prone because that movement is completely executed by the robot. However, the robot paths are created in such a way that the movement from top to bottom follows largely the same general direction, meaning there is no upward motion in a series of downward motions. Therefore, the robot's hysteresis does not play as large a role. Another DOF that can cause issues — but is not strongly affected by motion errors — is the rotation about the  $z$  axis ( $rz$ ); hence, it is also locked.

The remaining degrees of freedom are  $rx$ ,  $ry$  and  $z$ . This allows for closing the gap between two

scans, which is essential to avoid line-like artifacts and to align the scans (following the curvature of the part). In the following subsection, the concrete use of the modified ICP approach is documented.

### 5.2.3 Scan Registration Scheme

The general idea for performing the registration is to use a “master” scan as a seed and ensure continuity between the seed and all following scans in a pre-defined area. It should be noted that this does not necessarily guarantee the correct form and location of the resulting point cloud collection. The most important aspect of the alignment process is to avoid artifacts and therefore to ensure continuity.

The most straightforward approach for the alignment process would be to start with a corner scan and perform the registration in two directions from there. However, this has two major disadvantages. First, the overall error becomes larger with increasing distances from the template scan due to error propagation. Since there is no larger distance in any given rectangle than from one corner to its opposite corner, the alignment error is at its maximum when the diagonal (last) point cloud is aligned. Therefore, starting the registration process in a corner causes the largest possible errors. A second disadvantage is that there is only limited potential for parallelization because the alignment process continues in only two directions from the seed.

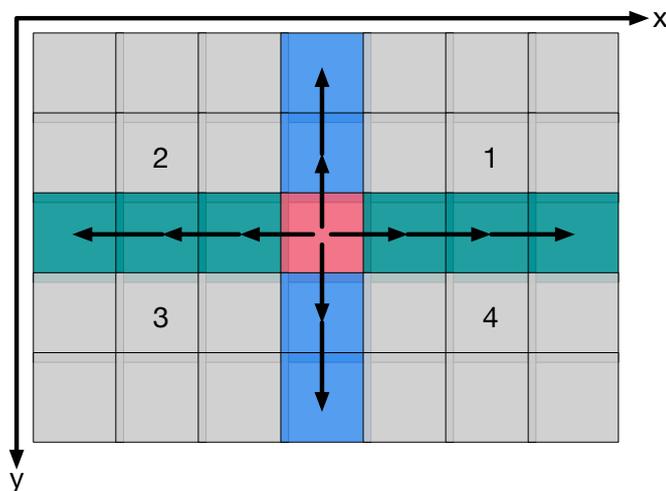


Figure 5.7: Illustration of rectangle splitting using a central seed.

A better approach is to use the center scan as the alignment seed. This both minimizes the error propagation and increases the parallelization potential, because it essentially splits the rectangular collection of scans into four smaller collections, as illustrated in Figure 5.7.

The registration procedure is as follows. First, the center cross is aligned based on the central template scan (red, green, and blue in Figure 5.7). This can be parallelized in four threads. Afterward, the cross serves as a frame for the following alignment steps. Since continuity needs to be guaranteed in both the  $x$  and  $y$  directions, it is insufficient to perform the alignments based on either the green

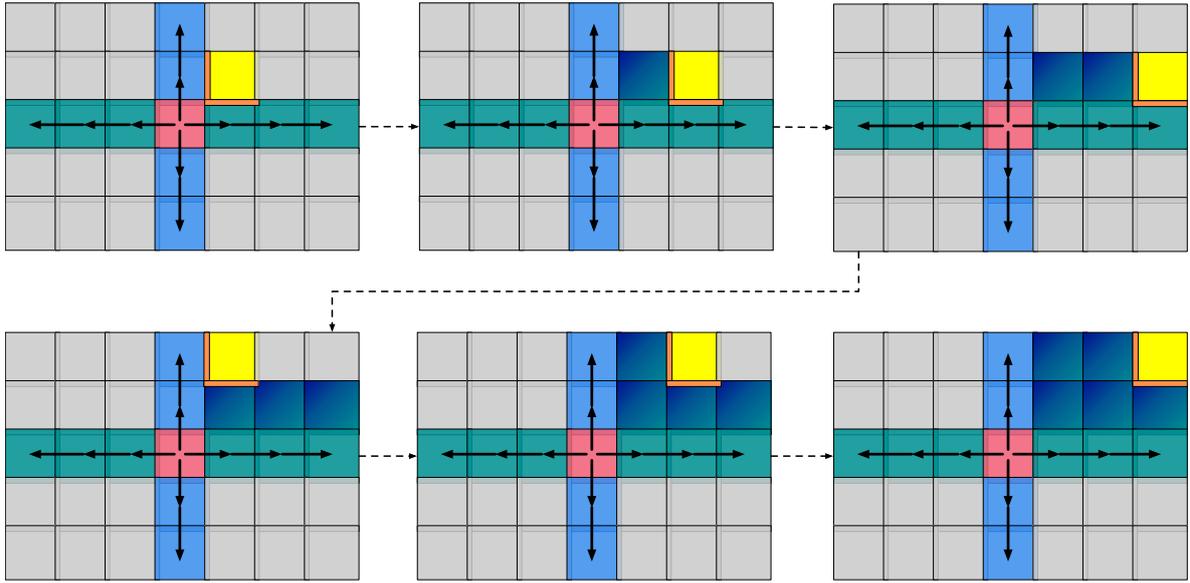


Figure 5.8: Illustration of the alignment process (shown only for the first quadrant). The currently aligned point cloud in each step is marked yellow; the used overlaps are marked orange.

or blue scans; rather, all available neighbors have to be used in each step. This process is illustrated in Figure 5.8 for the first quadrant. In the actual implementation, all four quadrants are aligned in parallel. Since the center cross was already aligned in the first step, aligning the four quadrants leads to the entire point cloud's being properly registered.

## 5.2.4 Overlap Determination

In order to speed up the alignment algorithm, the overlaps are determined in a pre-step. The alignment is then executed based on the overlaps instead of the full point clouds. The two point clouds describe the point sets  $A$  and  $B$ . The overlap set  $O_{A,B}$  is thus defined as:

$$O_{A,B} = \{p \mid p \in A \cap B\} \quad (5.19)$$

It will be shown that in this particular case, the intersection  $A \cap B$  can be determined in two-dimensional (2D) space. Figure 5.9 on the following page shows how two neighboring scans are related in space. When each scan is captured, it is in its own sensor coordinate system at  $(0, 0, 0)$ , see bottom left in Figure 5.10 on the following page. Because the robot TCP and the rotational axis angle and transform are known (see Figure 3.5 on page 49 in section 3.2.1), each and every scan can be transformed into a common coordinate system. This is illustrated in the center of Figure 5.10, where the scans were transformed into the common coordinate system through the application of the matrices  ${}^S_A\mathbf{T}$  and  ${}^S_B\mathbf{T}$  respectively. These transform matrices contain the combined effect of the robot's TCP position and the external axis configuration for each of the scanning configurations.

If one scan is used as a “master” scan, it can simply be returned to its own zero coordinate system,

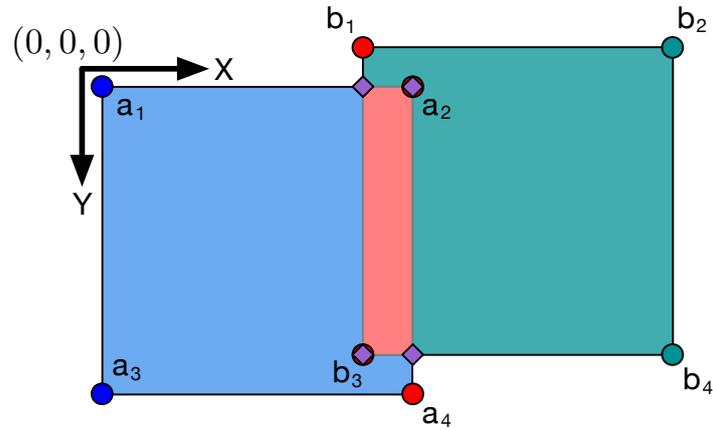


Figure 5.9: Illustration of the edge-point-based overlap detection. The overlap-determining edge points are marked red; the resulting overlap points are violet diamonds.

by transforming it with the inverse of its original transform,  $({}^S_A\mathbf{T})^{-1}$  because the multiplication of a matrix with its inverse, results, by definition, in the unit matrix. This means that the transform does not change the original point cloud (see blue image in bottom right corner of Figure 5.10). If any other

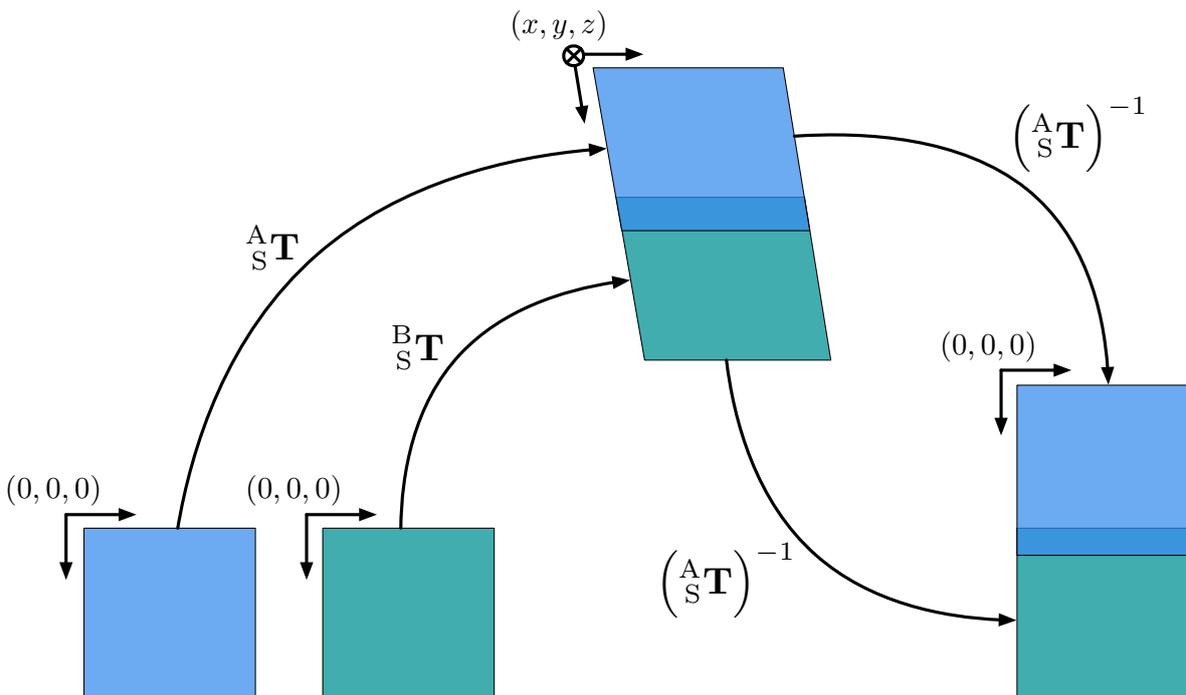


Figure 5.10: Illustration of the transformation scheme used to determine the overlaps.

scan is transformed from the common coordinate system with the master scan’s inverse transform matrix  $({}^S_A\mathbf{T})^{-1}$ , it is mapped into the master scan’s zero coordinate system. However, the matrix  ${}^S_B\mathbf{T}$ , which was originally used to transform the other scan into the common coordinate system, is not the

same as the master scan transform  $\mathcal{S}_A^T$ . Hence, the two scans do not end up in the same location but display the appropriate relative translation and rotation between them. For two neighboring scans, this means that the continuity between the two scans is preserved, but they are mapped into the first scan's zero coordinate system, see bottom right corner in Figure 5.10 on the preceding page.

Due to the equidistant pixel array, the first scan is guaranteed to be aligned perfectly with the  $x$  and  $y$  coordinate system axes. This means that the master scan can be treated as if it was in 2D space; it is therefore a 2.5D image. Due to the very transform errors the alignment procedure seeks to eliminate, as well as the shape of the measured object, this is not necessarily true for neighboring scans. However, since the entire system is calibrated, the effect of the transform errors can be expected to be relatively small between two neighboring scans. Furthermore, the curvature of the combustion chamber liner is gradual in comparison to the small measurement field of views. Additionally, the 2.5D assumption is exclusively required for the overlap determination, which is only the first step in an iterative alignment process and is therefore not required to be highly accurate. Thus, it is feasible to treat the second image as if it were a 2.5D image as well. Because of this assumption, the overlap of the two images can be directly determined in the  $x$ - $y$  plane, see Figure 5.9. This means that the overlap detection is reduced to comparing  $(x, y)$  coordinates, instead of performing an analysis in three dimensions.

In general, the overlap of a scan can be found by checking each of its points if it is inside the other scan's bounds. However, in this particular scenario the overlap determination can be sped up because a specific scan arrangement can be assumed. All scans will overlap in a near-parallel fashion, meaning that either two of their sides are perfectly aligned or that each scan has one corner point inside the other. In that case, the rectangle  $C$ , which defines the overlap in equation 5.19, can be determined by comparing the scans' corner points  $c_i$  (see blue, red, and green points in Figure 5.9):

$$\begin{aligned}
 c_1 &= (\max(a_{1,x}, b_{1,x}), \max(a_{1,y}, b_{1,y})) \\
 c_2 &= (\min(a_{2,x}, b_{2,x}), \max(a_{2,y}, b_{2,y})) \\
 c_3 &= (\min(a_{3,x}, b_{3,x}), \min(a_{3,y}, b_{3,y})) \\
 c_4 &= (\max(a_{4,x}, b_{4,x}), \min(a_{4,y}, b_{4,y})) \\
 C &= (c_1, c_2, c_3, c_4)
 \end{aligned} \tag{5.20}$$

Thus, the overlap  $O_{A,B}$  is inside the rectangle  $C$  with the four corner points  $c_i$ , see equation 5.20. This is indicated by the red area in Figure 5.9 on the preceding page.

### 5.3 Scan Unification

After the scans have been aligned successfully, they have a continuous surface in each subset. However, they are still a collection of individual overlapping scans. Since the overall strategy is to combine the scans and reduce the dimensionality (see Figure 5.1 on page 82), it is required to combine these

aligned scan subsets into scan matrices. This means that the points are in an equidistant raster in the  $x$ - $y$  plane. The overall approach to this matrix generation is illustrated in Figure 5.11. The idea is

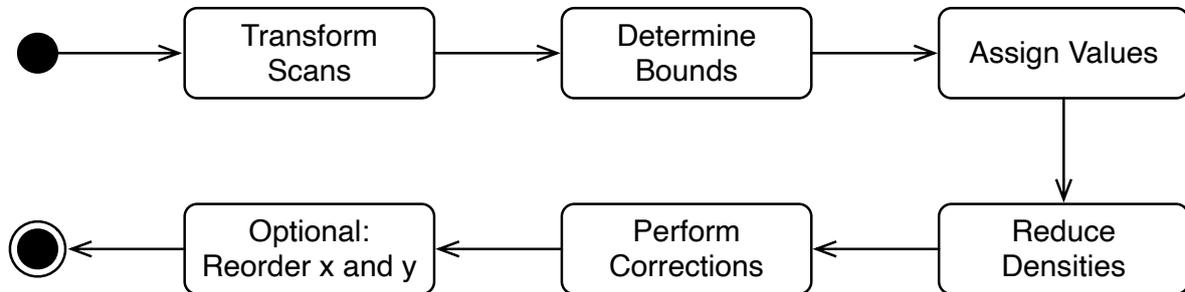


Figure 5.11: Overall approach to scan unification and matrix generation.

to transform all scans in a subset into their master scan's coordinate system. This means that after transforming the scan collection, the scans are approximately aligned with the  $x$ - $y$  plane. It should be noted that if the alignment procedure described in the previous section was performed, the scans have already been transformed. Thus, no additional transform has to be applied.

After the scans have been transformed, the matrix bounds can be determined by finding the minimum and maximum  $x$  and  $y$  values in the transformed scan collection. The matrix storage can then be allocated in the RAM and the values are then assigned to it in the appropriate locations. These first three steps are described in subsection 5.3.1.

A special challenge occurs in the overlapping portions of the scans because a bijective mapping into the matrix is no longer possible, see Figure 5.12 on the following page. Subsection 5.3.2 details how the overlap mapping can be accomplished.

Finally, certain corrections and an optional reordering of the  $x$  and  $y$  values have to be performed. These steps are described in subsection 5.3.3.

### 5.3.1 Assigning Values

Using the transform approach described in the previous subsection (see Figure 5.10 on page 93), the scan collections can be transformed into their respective master scan's coordinate systems. Since the white light interferometer has an equidistant pixel array, the master scan itself is equidistant in the  $x$ - $y$  plane if it is transformed back into its zero-coordinate system. Due to the comparatively large curvature radius of the part and the relatively small area covered by the scans, the other scans are still approximately equidistant in the  $x$ - $y$  plane when transformed into the master scan's coordinate system. This concept is similar to a linear approximation technique, such as ignoring higher order contributions of a Taylor series expansion.<sup>1</sup> Like with such a technique, the approximation gets worse with increasing distances from the center point of the approximation.

Every scan is equidistant in its own coordinate system and approximately equidistant in their master scan's coordinate system. Hence, the desired combined matrix mappings can be determined in

<sup>1</sup>For more information on the Taylor series, see for instance [Zei+12, pp. 266].

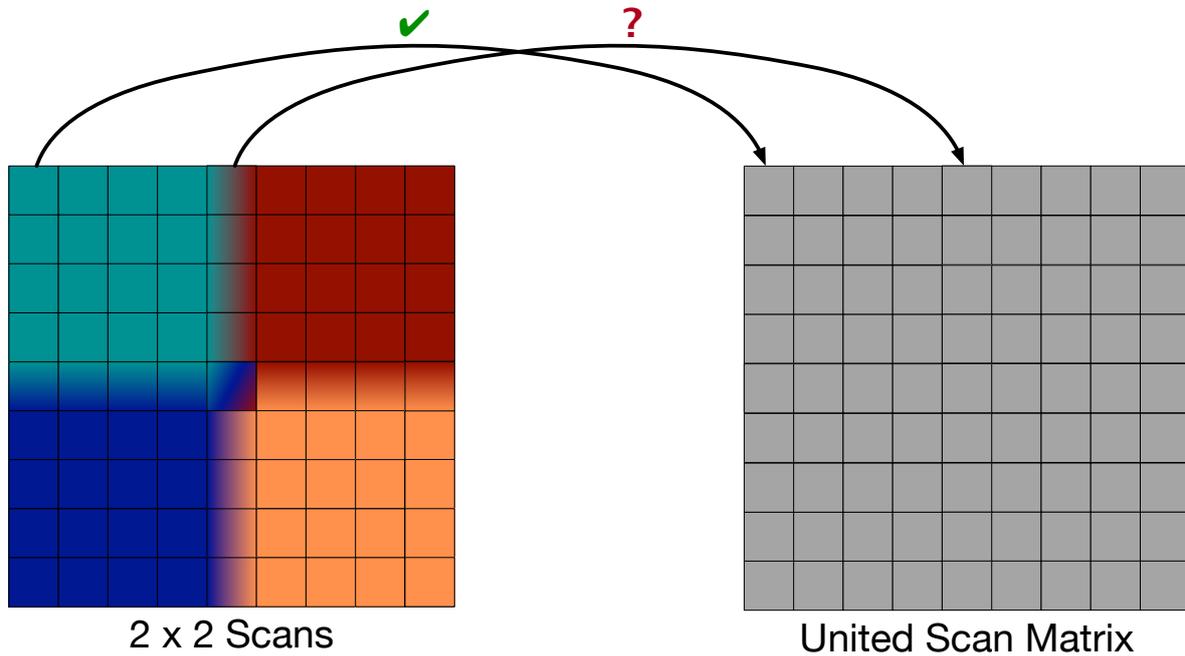


Figure 5.12: The matrix mapping procedure, highlighting issues due to multiple mappings in the overlap areas. The four individual scans on the left are given different colors (red, green, blue, and orange). The overlap areas between the individual scans are marked with the gradient colors of the overlapping scans.

a first step by converting the individual points'  $x$  and  $y$  coordinates from SI units to matrix indices. Using the known lateral discretization factors  $d_x$  and  $d_y$ , the conversion can be described as:

$$n = \left\lfloor \frac{\Delta x}{d_x} \right\rfloor = \left\lfloor \frac{x - x_{\min}}{d_x} \right\rfloor \quad (5.21a)$$

$$m = \left\lfloor \frac{\Delta y}{d_y} \right\rfloor = \left\lfloor \frac{y - y_{\min}}{d_y} \right\rfloor \quad (5.21b)$$

where  $\lfloor \cdot \rfloor$  is the floor function. Using the calculated indices, the mapping from the scans into the matrix can be performed, as illustrated in Figure 5.12.

Each matrix entry contains the following information in an `ExtendedPoint` object:  $x$ ,  $y$ ,  $z$ , and intensity values; and the flags “filtered” and “mapped,” as shown in Figure 5.13. The property “filtered” is set true if the entry’s intensity was too low to be considered a valid point (see subsection 4.3.2). The property “mapped” indicates whether a position in a matrix has been filled. Initially, all matrix points have the property “mapped” set to false.

Although the equation pair 5.21 provides a basic mapping strategy, it does not solve the overlap mapping issue. Therefore, a strategy for dealing with non-unique mappings had to be found and is described in the following subsection.

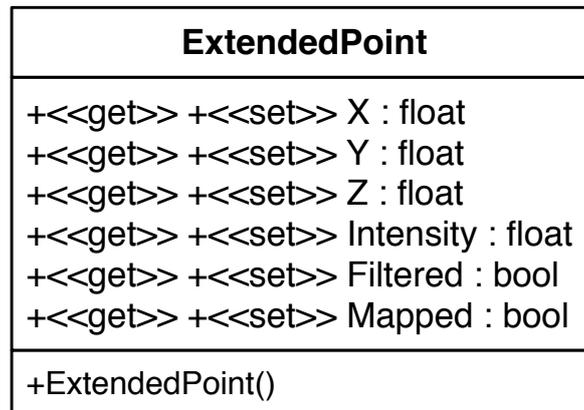


Figure 5.13: UML class diagram of the *ExtendedPoint* class, which makes up the data entries of the united scan matrix.

### 5.3.2 Overlap Mapping

The most basic way to treat the overlap mapping issue is to keep either the first- or last-mapped data point. In the first-in case, a simple check of whether the matrix cell has already been filled before each mapping is sufficient. The last-in case is even simpler to implement as no checks have to be performed at all; the data is always written. Without further considerations, the likelihood of each mapping candidate being the best choice is equal, assuming that the sensor quality is time-invariant. Whether the first-in or last-in method is the better choice is therefore not a question of quality but of which strategy is more time-intensive. The first-in method involves dereferencing an object inside of the matrix structure and then checking a Boolean member variable (“mapped”). In the other case, the appropriate object is dereferenced and modified, or simply overwritten, depending on the concrete implementation. Which strategy is more efficient depends on the overhead of checking every matrix element versus the computational expense of allocating or modifying objects in the overlap area several times. The question can therefore not be answered in general.

Another slightly more advanced strategy would be to include continuity considerations. The discrete  $x$ - $y$  plane mapping already ensures a uniform distribution of the point distances in that plane. This is not true for the  $z$  dimension, which represents the actual surface measurement. What can be expected is that in the overlap areas, the  $z$  values belonging to the same surface point should usually be very similar. However, if the registration algorithm failed to remove all misalignments between the scans, the differences might be much larger than expected. Without considering any additional information, every scan can be considered equally likely to contain the true distance information. As can be seen in Figure 5.12 on the preceding page, overlaps mostly occur between two, but up to four, scans. For two points in the same  $x$ - $y$  location, the objective is to equalize the absolute difference

between the resulting point and both original point distances:<sup>1</sup>

$$|z_1 - \bar{z}| = |z_2 - \bar{z}| \Leftrightarrow \bar{z} = \frac{z_1 + z_2}{2}$$

In general, the resulting value for more than one point is the mean of all  $N$  involved points:

$$\bar{z} = \frac{1}{N} \sum_{i=1}^N z_i \quad (5.22)$$

In the case of more than two points, it is possible that one or more points'  $z$  values deviate significantly from the others. Without considering further criteria, and assuming that the sensor mostly operates correctly and the TCP calibration is reasonably accurate, it would be reasonable to reject those outliers. A simple way to accomplish this outlier removal would be to use the median value instead of the mean, meaning that all  $z$  values are ordered and the middle value is used.

Due to the way the sensor works, an additional piece of information can be used, namely the intensity. The amount of photons integrated within a given time frame relates to how much information was extracted; in other words, more photons lead to more observations. Therefore, a higher level of trust can be associated with higher intensities. Thus, the intensity level can be used as a quality metric. There are several ways to use this metric. The most straightforward way would be to use only the point with the highest intensity value, or in the case of several points with equal intensity, their average distance. Another way would be to reject points with intensities lower than a pre-defined threshold. If all available information is used and appropriately weighted by a measure of trust, an optimal result can be achieved based on estimation theory.

According to the central limit theorem (see [Lof90, pp. 191] for more information), adding a sufficiently large number of independent random variables will result in a Gaussian distribution. Since the individual processes leading to the recorded signal envelope are independent, if the maximum of the resulting Gaussian envelope of the scan (see for instance Figure A.13 on page A35) is viewed as a random value, its distribution can be considered to be Gaussian. What is meant here is the assumed distribution of possible intensities for that particular distance, not the recorded envelope, which represents the intensities associated with *different* distances.

This means that the distribution can be fully described by only the first and second moments: mean and covariance. Therefore, a linear Gaussian model estimation approach can be used. This can be derived from the following equations for the predicted value and covariance (quoted from [ibidem, pp. 217]):

$$\mathbf{P}^+ = \left( (\mathbf{P}^-)^{-1} + \mathbf{C}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{C} \right)^{-1} \quad (5.23a)$$

$$\hat{\mathbf{x}}_r^+ = \mathbf{P}^+ \cdot (\mathbf{P}^-)^{-1} \cdot \hat{\mathbf{x}}_r^- + \mathbf{P}^+ \cdot \mathbf{C}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{y}_r \quad (5.23b)$$

---

<sup>1</sup>  $\sqrt{(\bar{z} - z_1)^2} = \sqrt{(\bar{z} - z_2)^2} \Leftrightarrow (\bar{z} - z_1)^2 = (\bar{z} - z_2)^2 \Leftrightarrow \bar{z} = \frac{z_2^2 - z_1^2}{2 \cdot (z_2 - z_1)} = \frac{z_2^2 - z_1^2}{2 \cdot (z_2 - z_1)} \cdot \frac{z_1 + z_2}{z_1 + z_2} = \frac{z_1 + z_2}{2}$

In the prediction equations 5.23, the “+” exponents denote prediction values and the “−” exponents denote previous values. The  $\mathbf{P}$ -matrices are covariances,  $\mathbf{C}$  is the observation matrix of the state-space model describing the process, and  $\mathbf{R}$  is the covariance matrix of the observation noise. The vector  $\underline{y}_r$  is the measured value and can be considered to be a realization of the random variable  $y$ . The measurements themselves are interpreted as observations of the state vector realization  $\underline{x}_r$  of the random vector  $\underline{x}$ , see [Lof90, pp. 209]. The hat ( $\hat{\cdot}$ ) denotes an estimator. The random vector realization  $\underline{x}_r$  and covariance  $\mathbf{P}$  are to be estimated, and the observation vector  $\underline{y}_r$  is measured. The  $\mathbf{C}$  and  $\mathbf{R}$  matrices have to be modeled. The resulting estimated values are optimal only with regard to the provided model.

In this particular case, there is no prior knowledge about the covariance. Since the covariance is unknown, every value has to be considered to be equally likely, meaning that the Gaussian distribution has infinite covariance and thus degenerates to a uniform distribution. Therefore, the inverse a priori covariance matrix can be modeled as the zero matrix:

$$(\mathbf{P}^-)^{-1} = \mathbf{0} \quad (5.24)$$

With equation 5.24, equations 5.23 can be simplified as follows:

$$\mathbf{P}^+ = (\mathbf{C}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{C})^{-1} \quad (5.25a)$$

$$\hat{\underline{x}}_r^+ = \mathbf{P}^+ \cdot \mathbf{C}^T \cdot \mathbf{R}^{-1} \cdot \underline{y}_r \quad (5.25b)$$

The hidden state is assumed to be the true distance from the sensor, and the sensor is assumed to yield an accurate observation of the distance, except for the influence of the observation noise. The observation matrix can therefore be modeled as a one for each observed value. Furthermore, since only one value is observed with every measurement, the observation vector size is  $n \times 1$ . Since covariance matrices are square matrices, it immediately follows that  $\mathbf{R}$  is an  $n \times n$  matrix. Because the true state is a  $1 \times 1$  scalar, the covariance  $\mathbf{P}$  also has to be scalar. It follows from equations 5.25 that  $\mathbf{C}^T$  is a  $1 \times n$  row vector and thus  $\mathbf{C}$  is a  $n \times 1$  column vector:

$$\underline{\mathbf{C}}^T = [1, 1, \dots, 1] \quad (5.26)$$

Since every measurement (observation) is performed at different points in time and for different kinematic configurations, the noise in each measurement can be assumed to be completely decoupled from the next one. Therefore, the covariance matrix of the observation noise can be modeled as a diagonal matrix containing the variances  $\sigma_i^2$  of one-dimensional Gaussian distributions:

$$\mathbf{R} = \text{diag}(\sigma_1^2 \dots \sigma_n^2) \quad (5.27)$$

Because the inverse of a diagonal matrix is again a diagonal matrix with the inverse of its elements

(see [Wal17, p. 236]), the matrix  $\mathbf{R}^{-1}$  can be expressed as:

$$\mathbf{R}^{-1} = \text{diag}(\sigma_1^{-2} \dots \sigma_n^{-2}) \quad (5.28)$$

Therefore, in order for the matrix to be invertible the individual variances need to be non-zero, because otherwise a division by zero would occur during the inversion of the matrix. In theory, this is not an issue, because a zero variance would imply that the exact value is known and thus no estimation procedure would be required in the first place. In practice, however, this should be considered for near-zero values in order to maintain numeric stability.

With equations 5.26 and 5.28, equations 5.25 can be simplified further. The expression for the prediction covariance contains two matrix multiplications:  $\underline{C}^T \cdot \mathbf{R}^{-1}$  and  $[\underline{C}^T \cdot \mathbf{R}^{-1}] \cdot \underline{C}$ . Since  $\underline{C}^T$  is a  $1 \times n$  row vector, the first multiplication maps the entire inverse measurement-noise covariance matrix onto a  $1 \times n$  row vector ( $[1 \times n] \cdot [n \times n]$ ). Since  $\mathbf{R}^{-1}$  is a diagonal matrix (see equation 5.28) and  $\underline{C}^T$  only contains ones, the result is a row vector containing the inverse variances:

$$\underline{C}^T \cdot \mathbf{R}^{-1} = [\sigma_1^{-2} \dots \sigma_n^{-2}] \quad (5.29)$$

The second multiplication maps the column vector  $\underline{C}$  onto a single scalar ( $[1 \times n] \cdot [n \times 1]$ ) through the row vector described in equation 5.29. The result is the sum of the inverse variances:

$$\underline{C}^T \cdot \mathbf{R}^{-1} \cdot \underline{C} = \sum_{i=1}^n \sigma_i^{-2} \quad (5.30)$$

Finally, with equation 5.30, the prediction covariance in equation 5.25 can be expressed as:

$$P^+ = \frac{1}{\sum_{i=1}^n \sigma_i^{-2}} \quad (5.31)$$

Similarly, the last two multiplications in the prediction-value equation in equations 5.25 can be interpreted as a mapping of the measurement  $[n \times 1]$  column vector onto a single scalar. Again, the vector resulting from the multiplication  $\underline{C}^T \cdot \mathbf{R}^{-1}$  (equation 5.29) is used. This leads to the following sum:

$$\underline{C}^T \cdot \mathbf{R}^{-1} \cdot \underline{y}_r = \sum_{i=1}^n \sigma_i^{-2} \cdot y_{r,i} \quad (5.32)$$

The prediction value is determined through the multiplication of the prediction covariance (equation 5.31) and the mapped measurements (equation 5.32). It can thus be expressed in terms of variances and measurements:

$$\hat{x}_r^+ = \frac{\sum_{i=1}^n \sigma_i^{-2} \cdot y_{r,i}}{\sum_{i=1}^n \sigma_i^{-2}} \quad (5.33)$$

The intensity can be considered to be inversely proportional to the standard deviation; the higher

the intensity, the more the value can be trusted. It follows that:

$$I_i \propto \sigma^{-1} \Leftrightarrow I_i^2 \propto \sigma_i^{-2} \quad (5.34)$$

Furthermore, with some proportionality constant  $k$ , the relationship between variance and intensity can be expressed as:

$$k \cdot I_i^2 = \sigma_i^{-2} \quad (5.35)$$

It follows from equation 5.35 inserted into equation 5.33 that:

$$\hat{x}_r^+ = \frac{\sum_{i=1}^n k \cdot I_i^2 \cdot y_{r,i}}{\sum_{i=1}^n k \cdot I_i^2} = \frac{n \cdot k \cdot \sum_{i=1}^n I_i^2 \cdot y_{r,i}}{n \cdot k \cdot \sum_{i=1}^n I_i^2} = \frac{\sum_{i=1}^n I_i^2 \cdot y_{r,i}}{\sum_{i=1}^n I_i^2} \quad (5.36)$$

Because the actual value of  $k$  is unknown, the prediction covariance can only be expressed as a proportional relationship in terms of intensity:

$$P^+ \propto \frac{1}{\sum_{i=1}^n I_i^2} \quad (5.37)$$

If  $k$  is set to 1, the estimation equation pair in terms of intensity can be expressed as:

$$P^+ = \frac{1}{\sum_{i=1}^n I_i^2} \quad (5.38a)$$

$$\hat{x}_r^+ = P^+ \cdot \sum_{i=1}^n I_i^2 \cdot y_{r,i} = \frac{\sum_{i=1}^n I_i^2 \cdot y_{r,i}}{\sum_{i=1}^n I_i^2} \quad (5.38b)$$

With equations 5.38, it was shown that the optimal estimator under the given conditions is a squared-intensity-weighted mean.

The optimum mapping method can be implemented as follows. First a two-dimensional array of extended pixels (see Figure 5.13) is allocated on the heap. The mapping equations used to calculate the appropriate indices are used to calculate the appropriate indices inside the two-dimensional array. The new  $x$  and  $y$  data is directly assigned to the extended pixel at that place, and the mapped property is set to true. The intensity is squared and added to the extended pixel's intensity value. Similarly, the current  $z$  value is multiplied by the squared intensity and added to the extended pixel's  $z$  value. After all scans have been processed and added to the two-dimensional array, the weighted mean calculation needs to be finished. First, the value calculated for the intensity needs to be inverted, resulting in  $P^+$  of equation 5.38. That value is then multiplied with the sum of the squared-intensity-weighted extended pixels'  $z$  values, resulting in  $\hat{x}_r^+$  of equation 5.38, which is assigned to the extended pixel's  $z$  value. Finally, the prediction covariance needs to be converted back to an intensity, using equation 5.35, with  $k = 1$ :  $I^+ = (P^+)^{-1/2}$ . The result is assigned to the extended pixel's intensity value. This algorithm has the advantage of performing all calculations in-place, which is feasible as long as the

expected sums are well below the limits of the employed floating-point type, which certainly is the case in this application. The procedure can be parallelized to a degree. In the second part, finishing the weighted mean calculation, all cells are completely independent and the calculations can therefore be fully parallelized. In the first part of the algorithm, there is a certain likelihood that two or more threads might try to access a given extended pixel at the same time. If the first part of the algorithm is to be parallelized, a monitor pattern has to be used on the extended pixel level in order to prevent race conditions.

### 5.3.3 Mapping Corrections

The mapping approach presented in subsection 5.3.1 has a general challenge, associated with the discrete nature of a pixel array. As can be seen in equations 5.21, a flooring operation is performed to obtain an index pair  $\in \mathbb{N}_0^2$ . The disadvantage of that approach is that the fractional part might actually round up, rather than down. This might lead to unmapped parts of the matrix.

A better approach would therefore be to modify equations 5.21 slightly to include the “round to nearest integer” function,  $\text{round}(\cdot)$ :

$$n_r = \text{round}\left(\frac{\Delta x}{d_x}\right) = \text{round}\left(\frac{x - x_{\min}}{d_x}\right) \quad (5.39a)$$

$$m_r = \text{round}\left(\frac{\Delta y}{d_y}\right) = \text{round}\left(\frac{y - y_{\min}}{d_y}\right) \quad (5.39b)$$

But even with that solution, it is not guaranteed that every single matrix position is filled, which would still result in missed mappings. These could be ignored in the matrix generation and may later be incorrectly detected as cracks and be rejected afterward. Alternatively, these minor holes could be filled by looking at their 4- or 8-neighborhoods. For instance, for a pixel’s 8-neighborhood:

$$\text{value}(p_{m,n}) = \frac{1}{C_{\text{valid}}} \sum_{i=-1}^1 \sum_{j=-1}^1 \text{valid}(p_{m+i,n+j}) \cdot \text{value}(p_{m+i,n+j}) \quad (5.40a)$$

$$\text{intensity}(p_{m,n}) = \frac{1}{C_{\text{valid}}} \sum_{i=-1}^1 \sum_{j=-1}^1 \text{valid}(p_{m+i,n+j}) \cdot \text{intensity}(p_{m+i,n+j}) \quad (5.40b)$$

$$\text{With } C_{\text{valid}} = \sum_{i=-1}^1 \sum_{j=-1}^1 \text{valid}(p_{m+i,n+j})$$

where the  $\text{valid}(\cdot)$  function is defined as:

$$\text{valid}(x, y) = \begin{cases} 1 & \text{if matrix}(x, y) \text{ is mapped} \\ 0 & \text{else} \end{cases} \quad (5.41)$$

Furthermore,  $\text{value}(\cdot)$  is the  $z$  value of the extended pixel, and  $\text{intensity}(\cdot)$  is the intensity value of the

extended pixel. Since every pixel is equally likely to be close to the true value, there is no stochastic justification to apply a weighted mean. Afterward, the averaged pixel's intensity needs to be evaluated to decide whether it needs to be filtered.

The approach of rounded mapping followed by missed-mappings corrections (equations 5.39 and 5.40) has two issues. First, a pixel that should be filtered — for instance, because it falls on a crack — might attain a higher intensity if its surrounding pixels are mostly non-crack pixels, due to the mean calculation. This would be a particular risk at the edge of cracks and very thin cracks. Second, the unrounded mapped values describe an uneven raster that is essentially forced into an equidistant one. In other words, neighboring pixels may have a certain degree of claim to be mapped into a neighboring cell in the matrix. Therefore, all pixels that are within a defined distance of the lateral discretization should be considered. In order for the mapping to yield the optimal solution, two criteria have to be fulfilled:

1. If a pixel is *exactly* at the discretization point, it should impact the cell's value at 100%.
2. If a pixel is one or more lateral discretization distances away, it should have no impact on the cell's value.

With only those two criteria, the resulting function is linearly dependent on the distance from the cell position. Since the distance in both  $x$  and  $y$  have to be considered, the resulting piecewise weight function is:

$$w(\Delta x, \Delta y) = \begin{cases} 0 & |\Delta x| > 1 \vee |\Delta y| > 1 \\ 1 - \sqrt{\Delta x^2 + \Delta y^2} & \text{else} \end{cases} \quad (5.42)$$

where  $\Delta x$  and  $\Delta y$  are the difference between a floating-point number and its rounded value, yielded from a rounding function. Since there are two lateral coordinates, which can round either up or down, there is a total of  $2^2 = 4$  combinations. Using the weight function defined in equation 5.42, these can be expressed as follows, where index  $f$  is for floor and index  $c$  is for ceiling:

$$w_{f,f}(x, y) = w(x - \lfloor x \rfloor, y - \lfloor y \rfloor) \quad (5.43a)$$

$$w_{f,c}(x, y) = w(x - \lfloor x \rfloor, y - \lceil y \rceil) \quad (5.43b)$$

$$w_{c,f}(x, y) = w(x - \lceil x \rceil, y - \lfloor y \rfloor) \quad (5.43c)$$

$$w_{c,c}(x, y) = w(x - \lceil x \rceil, y - \lceil y \rceil) \quad (5.43d)$$

The mapping equations (see equations 5.39) must be extended to include the four combinations as

well:

$$n_f = \left\lfloor \frac{\Delta x}{d_x} \right\rfloor = \left\lfloor \frac{x - x_{\min}}{d_x} \right\rfloor \quad (5.44a)$$

$$n_c = \left\lceil \frac{\Delta x}{d_x} \right\rceil = \left\lceil \frac{x - x_{\min}}{d_x} \right\rceil \quad (5.44b)$$

$$m_f = \left\lfloor \frac{\Delta y}{d_y} \right\rfloor = \left\lfloor \frac{y - y_{\min}}{d_y} \right\rfloor \quad (5.44c)$$

$$m_c = \left\lceil \frac{\Delta y}{d_y} \right\rceil = \left\lceil \frac{y - y_{\min}}{d_y} \right\rceil \quad (5.44d)$$

It should be noted that if  $x$  or  $y$  have no fractional part, their respective ceiling and floor functions would both yield the same result. This would skew the mean calculation if other pixels are involved and should therefore be checked. In those cases, only one floor-ceiling pair per variable should be used. The weighting approach is illustrated in Figure 5.14. Solid lines indicate the lateral discretization

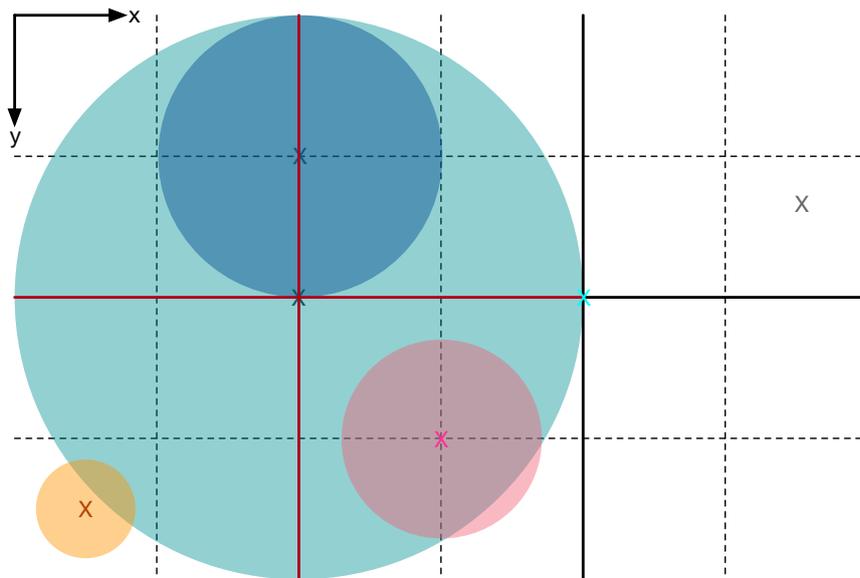


Figure 5.14: Weights of various pixels for the matrix position marked by the red lines.

points, therefore, solid crossing points indicate a matrix position ( $x, y$  pair). The dashed lines indicate half-discretization points. The solid red lines mark the matrix position under consideration, the Xs mark potential pixels, and their surrounding discs their weight. Pixels without a disc have a weight of zero. As expected, the pixel contribution that falls directly on the matrix position has 100% weight. The turquoise pixel that falls directly on the next matrix position and the gray one on the far upper right each have exactly 0% weight.

The weights are applied to the intensity and value calculations (see equations 5.38):

$$P_w^+ = \frac{1}{\sum_{i=1}^n I_i^2 \cdot w_i} \quad (5.45a)$$

$$\hat{x}_{r,w}^+ = P_w^+ \cdot \sum_{i=1}^n I_i^2 \cdot y_{r,i} \cdot w_i = \frac{\sum_{i=1}^n I_i^2 \cdot y_{r,i} \cdot w_i}{\sum_{i=1}^n I_i^2 \cdot w_i} \quad (5.45b)$$

With these mappings, it can be ensured that every matrix position is filled with appropriate values. Furthermore, in the ideal case of every  $x, y$  combination aligning perfectly with the chosen grid, the described algorithm would yield the ideal result, as required. In the case of a poor alignment, the result would be the mean of four equally weighted pixels, which effectively results in a low pass filter. Due to an equidistant pixel array, the large curvature radius of the part, the relatively small dimensions of the resulting matrix, the pre-alignment procedure described in section 5.2, and the use of the center as the template scan (see Figure 5.8), this effect can be expected to be minimal.

As a last step, the matrix  $x$  and  $y$  values can be reordered. When the points are assigned to their respective matrix cells by the algorithm described in this section, the points themselves will retain their sub-cell resolution. In some cases, it can be desirable to have a point cloud that is equidistant in  $x$  and  $y$ . Here, the  $x$  and  $y$  values of the points can be replaced by their matrix index, multiplied by the appropriate discretization, and shifted by the original  $x$  and  $y$  values of the first cell point:

$$x_n = x_{\min} + n \cdot d_x \quad (5.46a)$$

$$y_m = y_{\min} + m \cdot d_y \quad (5.46b)$$

This is the inversion of equations 5.21 (with loss of fraction).

## 5.4 Generation of 2.5D Images

After the procedure described in the previous subsection has been applied to an aligned scan collection, the result is a scan matrix that contains all the relevant information for crack detection. The result is an organized and laterally equidistant point cloud in the template scan's coordinate system. Up to this point, the procedures were not specific to the detection task. Crack detection could theoretically be performed using all three dimensions. However, it is faster and more robust to decrease the dimensionality and perform two-dimensional image processing to detect damages (see Figure 5.1). What this means is that the measured  $z$  values are used to create a 2.5D image. With a general point cloud, this kind of representation is not necessarily directly possible. However, due to the way the scan matrix was created — namely equidistant in the  $x$ - $y$  plane — this kind of 2.5D representation can be created directly from the matrix. The scan matrices consist of extended points (see Figure 5.13), which themselves contain six-dimensional information:  $x, y, z$ , intensity, mapped, and filtered. The matrices can therefore also be represented as a stack of six matrices containing one-dimensional data. From this stack, the four relevant matrices,  $z$ , intensity, mapped, and filtered can be extracted.

### 5.4.1 General Image-Generation Approach

The generated scans could theoretically be processed directly. However, it is desirable to use existing image-processing libraries to extract the crack information from the generated scans. Unfortunately, common libraries largely rely on processing single images with eight bits per channel, not stacks of two 16-bit and two one-bit images. It is, therefore, necessary to consolidate the required information into a single matrix and convert that matrix into a standard eight-bit-per-channel bitmap. The intensity information is not required in this case and is therefore discarded for the crack analysis. The bulk of information is inside the  $z$  matrix because it contains the topology of the scanned surface. The “filtered” matrix is relevant because it contains the condensed information extracted from the intensities: which pixels are valid and which ones are invalid. The “mapped” matrix indicates which matrix elements actually contain mapped data. This is important, because unmapped areas should not be considered in the crack detection process. These three matrices could be mapped directly into a three-channel bitmap, with each channel containing one of the matrices. However, viewing such an image in a standard image viewer would result in “unintuitive” images. Therefore, a different approach was chosen:

1. If the pixel is not mapped, the pixel is displayed as blue ( $R = 0, G = 0, B = 1.0$ ).
2. If the pixel is filtered, the pixel is displayed as green ( $R = 0, G = 1.0, B = 0$ ).
3. Otherwise, all channels represents the  $z$  value ( $R = f(z), G = f(z), B = f(z)$ ).

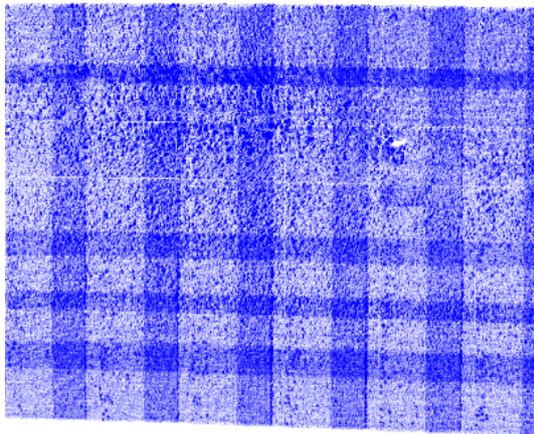
All channel values are mapped between 0 and 1.0. Thus, the resulting image is a grayscale image with blue and green parts, where pixels are not mapped or filtered, respectively, as illustrated in Figure 5.15.

### 5.4.2 Form-Removal Using Polynomial Regression in Two Variables

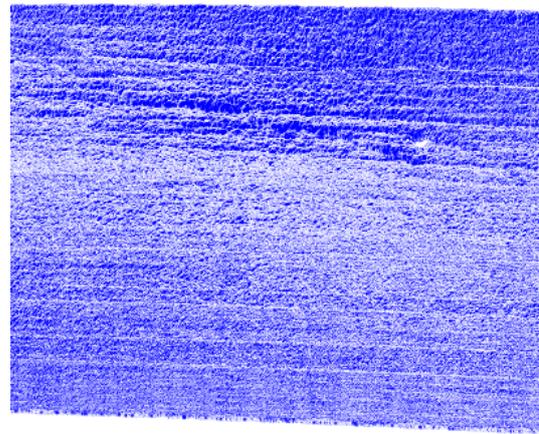
As indicated by the function notation  $f(z)$ , a mapping from the original  $z$  values into the image intensity range (normalized to 1.0) has to be performed. Since the discrete  $z$  space is larger than the discrete intensity range, this mapping causes a loss of relevant information if no further measures are taken. As will be shown in section 5.6, the crack detection approach is based on detecting changes in the surfaces. If a scan is thought of as a two-dimensional discrete real signal, it must have a two-dimensional discrete complex Fourier spectrum, which can be obtained through a two-dimensional discrete Fourier transform (DFT).<sup>1</sup> Thus, changes can be thought of as higher frequency components, which means that the lower frequency components are irrelevant for crack detection and can, therefore, be removed. In other words, a procedure with suitable high-pass characteristic can remove the unwanted information content. Afterward, a mapping function can be applied to create an image, as shown in Figure 5.15.

Since the 2.5D scan can be thought of as a point-wise defined discrete function in two variables ( $x$  and  $y$ ), it can also be represented by a less specific function, for instance, representing the scan’s

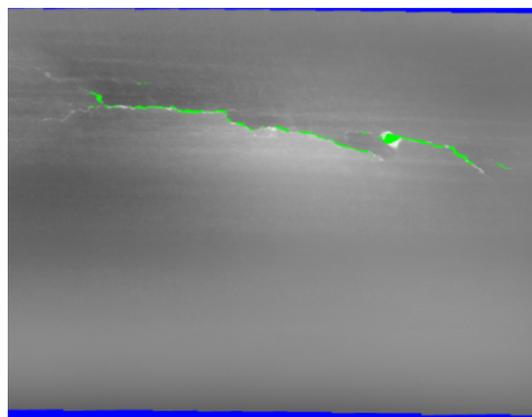
<sup>1</sup>Or its fast implementation, the fast Fourier transform (FFT).



(a) The individual input scans.



(b) The united scan matrix.



(c) The resulting image.

Figure 5.15: Example of the image creation steps in preparation for crack detection.

mean value, gradient, or  $n^{\text{th}}$ -order curvatures. Up to a certain degree of curvature, those attributes are precisely the low-frequency content that is irrelevant for crack detection, as mentioned in the previous paragraph. One way to remove those contents would be to perform a mathematical regression to calculate such a function, evaluate that function at each  $(x, y)$  pair present in the scan and subtract the resulting  $z$  value from the input scan.

A suitable class of functions is  $n^{\text{th}}$ -order polynomials in two variables. For a function in one variable  $g(x)$ , a polynomial of  $n^{\text{th}}$  order is defined as a function of the following form (see for instance [Zei+12, pp. 592]):

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n, \quad a_n \neq 0 \quad (5.47)$$

It should be noted that the degree of the polynomial is defined by the highest exponential of the variable  $x$ , which is why  $a_n$  must not be 0.

A mathematical regression is a procedure that determines function parameters to fit a given sample

set in an optimal way, typically in the least-squares sense. The procedure does neither determine the appropriate function type nor the ideal number of free parameters. If, for instance, a random set of samples is to be approximated by a line in the least-squares sense, this would be accomplished by performing a linear least-squares regression for a polynomial in one variable of first order:  $f(x) = a_0 + a_1x$ , or in more typically used notation:  $f(x) = y = mx + b$ . In order to find the optimal parameters, the observations, inputs and parameters themselves have to be combined in a linear set of equations, which can then be solved for the parameters. If there are  $n$  observations in the given example, they are presumed to have been generated from the first-order polynomial, thus:

$$\begin{aligned} a_0 + a_1x_1 &= y_1 \\ a_0 + a_1x_2 &= y_2 \\ &\vdots \\ a_0 + a_1x_n &= y_n \end{aligned}$$

This set of equations can also be expressed in matrix notation:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

With the notation  $\mathbf{M}$  for the matrix containing the mapping between the inputs and outputs,  $\underline{x}$  for the inputs,  $\underline{y}$  for the outputs and  $\underline{a}$  for the function parameters, this can also be expressed as:

$$\underline{y} = \mathbf{M} \cdot \underline{a} \quad (5.48)$$

Equation 5.48 is also valid for higher order polynomials. Here, the mapping matrix will have more columns, containing the higher order powers of the input vector, and the function parameter vector will have more rows. To obtain the function parameter vector, the matrix equation has to be left-multiplied by the inverse of the mapping matrix. However,  $\mathbf{M}$  is typically not a square matrix, which means that it cannot necessarily be inverted in the classical sense. If enough linearly independent observations exist, which is to say at least as many as the model parameters, a solution can still be found through using a pseudo inverse  $\mathbf{M}^\dagger$ . One method of arriving at such a pseudo inverse, the Moore-Penrose

inverse, involves the matrix transpose:

$$\begin{aligned}
\mathbf{M} \cdot \underline{a} &= \underline{y} \\
\Leftrightarrow \mathbf{M}^T \cdot \mathbf{M} \cdot \underline{a} &= \mathbf{M}^T \cdot \underline{y} \\
\Leftrightarrow (\mathbf{M}^T \cdot \mathbf{M})^{-1} \cdot (\mathbf{M}^T \cdot \mathbf{M}) \cdot \underline{a} &= (\mathbf{M}^T \cdot \mathbf{M})^{-1} \mathbf{M}^T \cdot \underline{y} \\
\Leftrightarrow \underline{a} &= (\mathbf{M}^T \cdot \mathbf{M})^{-1} \mathbf{M}^T \cdot \underline{y} = \mathbf{M}^\dagger \cdot \underline{y}
\end{aligned} \tag{5.49}$$

Thus, a solution for the linear set of equations exists if, and only if, the matrix  $(\mathbf{M}^T \cdot \mathbf{M})$  is invertible. It can be shown that the Moore-Penrose pseudo-inverse delivers the least-squares solution to the problem using the 2-norm of the residual vector:  $\|\underline{r}\|_2 = \|\mathbf{M} \cdot \underline{a} - \underline{y}\|_2$  (see for instance [SR14, pp. 639]). It should be noted that the direct calculation of the pseudo-inverse, shown in equations 5.49, can lead to numerical issues and should be avoided.

A better approach would be to use the singular value decomposition (SVD) to calculate the Moore-Penrose pseudo-inverse. First the decomposition is calculated, yielding  $\text{SVD}(\mathbf{M}) = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^*$ . Then the singular value matrix  $\mathbf{S}$  is inverted, which (since the the matrix is a diagonal matrix) reduces to an inversion of the non-zero elements:  $\mathbf{S}^{-1} = \text{diag}(s_1^{-1} \dots s_n^{-1})$ . The pseudo-inverse can then be expressed as  $\mathbf{M}^\dagger = \mathbf{V} \cdot \mathbf{S}^{-1} \cdot \mathbf{U}^*$  [ibidem, pp. 642] [LM11, pp. 318].

Another alternative to solve linear sets of equations is the QR decomposition. With this approach, like with the SVD decomposition, the  $[m \times n]$  matrix  $\mathbf{M}$  is decomposed into matrices with special properties. However, in this case only two matrices are required:  $\mathbf{M} = \mathbf{Q} \cdot \mathbf{R}$ . The  $\mathbf{Q}$  matrix is an orthogonal  $[m \times m]$  matrix, meaning that  $\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ . The  $\mathbf{R}$  matrix on the other hand is an  $[m \times n]$  matrix of the following form:

$$\mathbf{R} = \begin{bmatrix} \tilde{\mathbf{R}} \\ \mathbf{0} \end{bmatrix} \tag{5.50}$$

where  $\tilde{\mathbf{R}}$  is a right triangular matrix. The matrix decomposition into the QR pair can, for instance, be calculated using Householder reflections [RW17, pp. 102]. Once the QR decomposition has been determined, the linear set of equations can be rearranged as follows:

$$\begin{aligned}
\mathbf{M} \cdot \underline{a} &= \underline{y} \\
\Leftrightarrow \mathbf{Q} \cdot \mathbf{R} \cdot \underline{a} &= \underline{y} \\
\Leftrightarrow \mathbf{R} \cdot \underline{a} &= \mathbf{Q}^T \cdot \underline{y} \\
\Leftrightarrow \mathbf{R} \cdot \underline{a} &= \tilde{\underline{y}}
\end{aligned} \tag{5.51}$$

The modified linear set of equations in equation 5.51 can be easily solved for  $\underline{a}$  through back substitution, due to the structure of  $\mathbf{R}$  (see equation 5.50). There are many more approaches and considerations for numerically solving linear set of equations, an overview of which can be found in [Mat14].

As mentioned before, the linear regression will also work if the chosen model function does not match the underlying function well. The regression, regardless of how it is performed, will then determine the parameters that produce the best fit, in the least-squares sense, for the observed samples.

The extension of the concept of linear regression for  $n^{\text{th}}$ -order polynomials in one variable can be extended to two variables. The function in two variables simply contains all contributions of the individual polynomials as well as all their mixed products. One way to obtain all the combinations is by writing each single contribution as a vector and calculating the inner product:

$$f(x, y) = \langle \underline{x}, \underline{y} \rangle = \begin{bmatrix} p_0 & p_1x & \dots & p_nx^n \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1y \\ \vdots \\ q_ny^n \end{bmatrix} \quad (5.52)$$

$$= p_0q_0 + p_0q_1y + \dots + p_0q_ny^n + p_1q_0x + p_1q_1xy + \dots + p_1q_nxy^n + \dots + p_nq_nx^ny^n \quad (5.53)$$

The parameter products in equation 5.53 can be combined into single parameters:

$$f(x, y) = a_{0,0} + a_{0,1}y + \dots + a_{0,n}y^n + a_{1,0}x + a_{1,1}xy + \dots + a_{n,n}x^ny^n$$

$$= \sum_{i=0}^n \sum_{j=0}^n a_{i,j} \cdot x^i \cdot y^j \quad (5.54)$$

This means that for an  $n^{\text{th}}$ -order polynomial in two variables, there are  $(n + 1)^2$  free parameters:

$$N_{\text{Parameter}}(n) = (n + 1)^2 \quad (5.55)$$

Whereas equation 5.54 shows the full polynomial, not all  $(x, y)$  combinations are necessarily required to adequately represent the dataset. The overall goal is to find a polynomial that describes the basic shape of the scan, namely the lower frequency content. It can be argued that the larger one variable's powers are, the less important the combinations with higher-order powers of the other variable become. This is because the contributions of higher-order powers of even one of the variables are usually comparatively small in comparison to those of lower order powers. Such a polynomial may be constructed in such a way that the highest  $x$  and  $y$  powers are present only by themselves ( $x^n$  and  $y^n$ ), the next lower powers only by themselves and in combination with the first power of the other variable ( $x^{n-1}$ ,  $x^{n-1} \cdot y$ ,  $y^{n-1}$  and  $y^{n-1} \cdot x$ ), and so forth. These combinations can be found by the following double sum:

$$\tilde{f}(x, y) = \sum_{i=0}^n \sum_{j=0}^{n-i} a_{i,j} \cdot x^i \cdot y^j \quad (5.56)$$

For the number of parameters, the highest power of  $x$  ( $x^n$ ) appears only once, the next highest power ( $x^{n-1}$ ) twice, down to the lowest power ( $x^0 = 1$ ), which appears  $(n + 1)$  times.<sup>1</sup> Hence, it can be concluded that the number of parameters is  $(n + 1) + n + (n - 1) + \dots + 1$ . This is an arithmetic series [Zei+12, p. 31]:  $\sum_{k=0}^m (a + kd) = \frac{m+1}{2}(a + (a + md))$ , with  $a = 0$ ,  $d = 1$  and  $m = n + 1$ . Thus, the number of parameters is:

$$\tilde{N}_{\text{Parameter}}(n) = \frac{n+2}{2}(n+1) \quad (5.57)$$

If the number of parameters for the reduced polynomial is compared to the one required for the full polynomial, it is evident that:

$$\frac{N_{\text{Parameter}}(n)}{\tilde{N}_{\text{Parameter}}(n)} = \frac{(n+1)^2}{\frac{n+2}{2}(n+1)} = \frac{2(n+1)}{n+2} = 1 + \frac{n}{n+2} \quad (5.58)$$

For the case  $n = 0$ , the number of parameters is the same for the full and the reduced parameter sets. For  $n = 1$ , the full parameter set already requires 1/3 more (4 versus 3) parameters. For arbitrarily large  $n$ , the ratio approaches:

$$\lim_{n \rightarrow \infty} \left( 1 + \frac{n}{n+2} \right) = 1 + \lim_{n \rightarrow \infty} \frac{1}{1+2/n} = 2$$

The linear set of equations is built up analogously to the single-variable case, given a set of observations  $\underline{z}$  of the assumed polynomial function. For instance, for the equivalent element of a line in two dimensions (a plane), this can be expressed with equation 5.56 as:

$$\begin{aligned} a_0 + a_1x_1 + a_2y_1 &= z_1 \\ a_0 + a_1x_2 + a_2y_2 &= z_2 \\ &\vdots \\ a_0 + a_1x_n + a_2y_n &= z_n \end{aligned}$$

It should be stated explicitly that the  $x$  and  $y$  indices denote the two free variable values for the given observation number.

As with the single-variable case, these observations can be expressed in matrix-vector notation:

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

<sup>1</sup>The variable  $y$  appears in powers from 0 to  $n$ .

Or in symbolic notation:

$$\underline{z} = \mathbf{M} \cdot \underline{a} \quad (5.59)$$

The linear set of equations in equation 5.59 can be solved as described for the single-variable case earlier in this section.

If a regression for a polynomial of higher order has to be calculated, the higher-order powers would be added in the  $\mathbf{M}$  matrix as columns and the parameters as rows in  $\underline{a}$ . This shall be demonstrated for the case  $n = 2$ , for the reduced polynomial (equation 5.56). The number of parameters is  $\tilde{N}_{\text{Parameter}}(n) = \frac{2+2}{2}(2+1) = 6$ :

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} x_1^2 & x_1 y_1 & x_1 & y_1^2 & y_1 & 1 \\ x_2^2 & x_2 y_2 & x_2 & y_2^2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_n y_n & x_n & y_n^2 & y_n & 1 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_5 \end{bmatrix}$$

It should be noted that the linear set of equations is set up in accordance with equation 5.56. Multiplying the  $[n \times 6]$  matrix with the  $[6 \times 1]$  parameter vector  $\underline{a}$  results in an  $[n \times 1]$  vector, where each vector entry contains a sum (as described in equation 5.56).

The effect of different-order polynomial approximations in a 2.5D false color representation is shown in Figure 5.16. Figure 5.16a shows the original scan of part of a €1 coin. The other panels

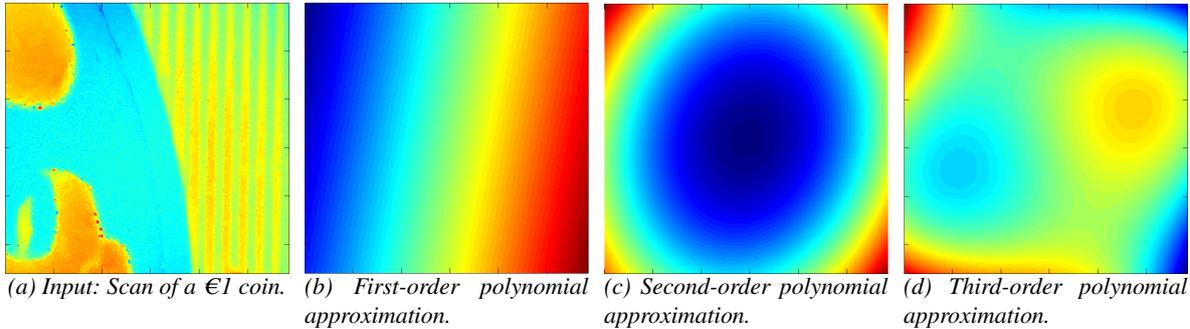


Figure 5.16: Scan of a €1 coin and polynomial approximations of different orders.

show the result of the two-dimensional polynomial approximation for the first, second, and third order, respectively. From each step to the next, the previous results were subtracted from the input image and the regression was then performed on the modified input image. This was done to see only the effect of the higher order terms, for the combined effect of higher order approximations, see for instance subsection 5.4.4. The first-order approximation shown in Figure 5.16b only has enough parameters for a gradient approximation, followed by a paraboloid for the second order (Figure 5.16c) and a cubical paraboloid for the third order (Figure 5.16d).

Unfortunately, the basic approach described in this subsection has a severe limitation, which will be discussed together with a solution in the following subsection.

### 5.4.3 RANSAC-based Polynomial Regression

The approach described in the previous subsection works well if the samples do not contain a significant number of outliers. Figure 5.17a shows a typical scan of an airplane turbine liner section and Figure 5.17b shows the polynomial-filtered result. It can be clearly seen that the surface form has been

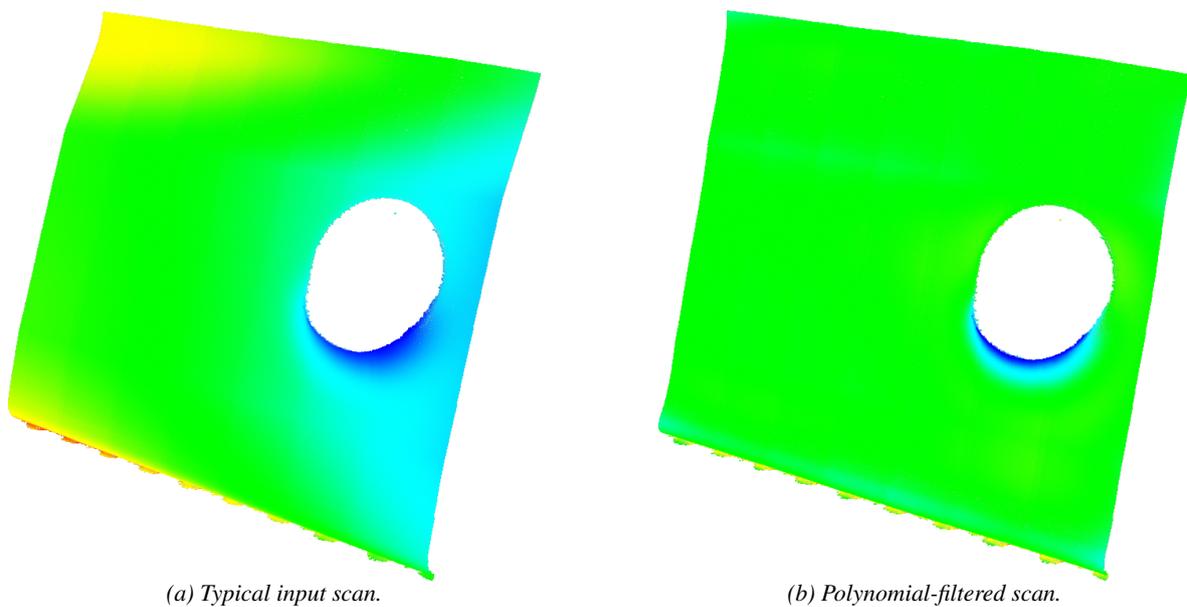


Figure 5.17: Color-coded representation of a typical input scan and the polynomial-filtered result.

removed, resulting in a uniform surface, except for areas with discontinuities.

If a scan contains significant discontinuities, for instance because the liner is severely damaged or because part of another liner panel is visible in the scan matrix, the method described in the previous subsection will fail. Such a problematic case, featuring a severely damaged part, is shown in Figure 5.18a. As expected, the resulting polynomial-filtered scan is significantly influenced by the discontinuity (see Figure 5.18b). Even if the polynomial filter had worked properly, the resulting depth range would still have been too large for proper image generation due to the presence of the discontinuity.

It is, therefore, important to recognize the affected areas, remove them and perform the polynomial regression only with the valid data. This approach is known as outlier rejection. A well-known approach for outlier rejection is the *random sample consensus* (RANSAC) algorithm. Like with regression, the approach is based on the assumption that the data set can be approximated by a function with unknown parameters. However, in this case, not the entire set of samples is used, because the sample set cannot be trusted. Rather, the minimum number of samples required to determine the parameters of the assumed function are randomly selected. For example, in order to determine the parameters of a line in 2D space, the number of required distinct samples would be two.

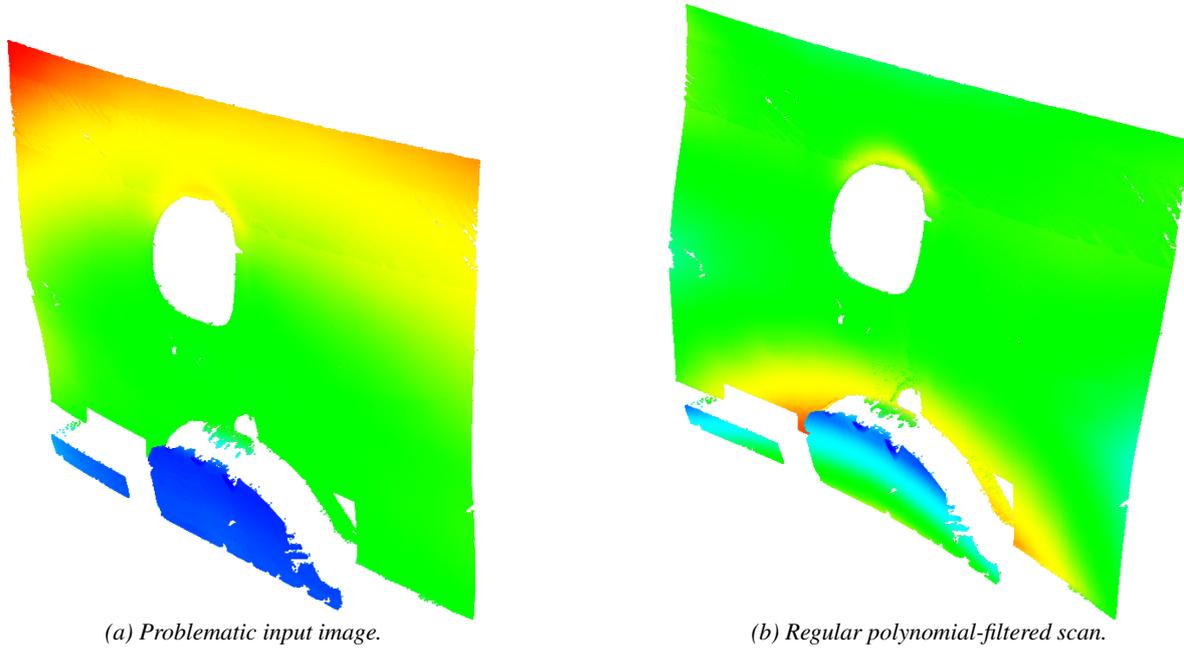


Figure 5.18: Problematic input image and its polynomial-filtered result.

In this case, the used model is a two-dimensional polynomial. After the function parameters  $\underline{a}$  are determined from the chosen samples, the function is evaluated at all sample locations  $(x_i, y_i)$ . Then the distances between the evaluated values and sample values are calculated for all points. The distance for the  $i$ -th point is:

$$d_i = \sqrt{(f_{\underline{a}}(x_i, y_i) - z_i)^2} \quad (5.60)$$

The points with distances that undercut a predefined threshold are considered to consent with the chosen samples. It should be noted that in order to avoid the computational expense of the square root operation, squared differences are used instead of the distances described in equation 5.60:

$$d_i^2 = (f_{\underline{a}}(x_i, y_i) - z_i)^2 \quad (5.61)$$

To compensate for this, the user-defined distance threshold is squared before it is compared to the squared differences.

If the number of consenting samples is greater than a second predefined threshold, the algorithm ends. The consenting samples are considered to be inliers and the other points to be outliers. If the threshold is not reached, a different sample set is randomly chosen, and a second iteration is performed. After the algorithm has terminated, either the function parameters chosen in the successful RANSAC iteration can be used directly, or the polynomial estimation described in the previous subsection can be performed on the inlier set. Additionally, the outlier set is removed from the input set. For the polynomials in two variables (covered in the previous subsection), the minimum required number of samples for the full parameter set is given in equation 5.55 and that for the reduced

polynomial in equation 5.57.

The algorithm should terminate after a predefined number of iterations so that it does not run indefinitely. The question of how many iterations have to be performed to be able to expect a successful outcome is a matter of probabilities. The probability to select an inlier out of the full point set is  $P_{\text{Inlier}}$ . In order to calculate the model, a minimum of  $n_{\text{min}}$  points is required. If the size of the point set is much larger than  $n_{\text{min}}$ , this probability can be approximated to be constant for every pick, even though the probability changes (because points are removed from the set each time). This means that the probability to select only inliers is  $(P_{\text{Inlier}})^{n_{\text{min}}}$ . The events “only inliers are selected” and “at least one outlier is selected” are disjoint. Furthermore, the probability that either of these events occurs is one. Therefore, it follows directly from Kolmogorov’s third axiom [Lof90, p. 105] that the probability of selecting at least one outlier point is  $1 - (P_{\text{Inlier}})^{n_{\text{min}}}$ . If the algorithm is iterated  $K$  times, the chance of selecting a failing set each time is  $(1 - (P_{\text{Inlier}})^{n_{\text{min}}})^K$ . Since that probability is complementary to picking at least one valid set, it follows that the probability for the algorithm to succeed is  $P_{\text{Success}} = 1 - (1 - (P_{\text{Inlier}})^{n_{\text{min}}})^K$ . The equation is solved for  $K$ :

$$P_{\text{Success}} = 1 - (1 - (P_{\text{Inlier}})^{n_{\text{min}}})^K \quad (5.62)$$

$$(1 - (P_{\text{Inlier}})^{n_{\text{min}}})^K = 1 - P_{\text{Success}}$$

$$\log((1 - (P_{\text{Inlier}})^{n_{\text{min}}})^K) = \log(1 - P_{\text{Success}})$$

$$K \cdot \log(1 - (P_{\text{Inlier}})^{n_{\text{min}}}) = \log(1 - P_{\text{Success}})$$

$$K = \frac{\log(1 - P_{\text{Success}})}{\log(1 - (P_{\text{Inlier}})^{n_{\text{min}}})} \quad (5.63)$$

The probability of success is a free parameter that needs to be chosen appropriately, for instance 99%. The parameter  $n_{\text{min}}$  depends on the model function.

The probability of selecting an inlier has to be estimated. This could theoretically be done automatically on a per-scan basis by performing a statistical analysis.<sup>1</sup> In practice, it was found to be sufficient for the probability to be estimated once, when setting up the system. As an example, if scans are expected to contain no more than 30% erroneous data, the probability would be 70%. It is advisable to assume a lower inlier probability because the worst case is that this would lead to more iterations than strictly necessary. Additionally, what is actually used in the algorithm is the number of sample sets that need to be selected, which is the number of iterations. However, the algorithm can be extended to terminate as soon as the expected inlier threshold is crossed, resulting in shorter runtimes.

<sup>1</sup>It can be assumed that the captured scans have an underlying probability density function of either Gaussian or heavy-tailed type. The similarity to the assumed distribution can be determined using a divergence measure, such as the Kullback-Leibler divergence [Gho18, pp. 5], which determines the relative entropy between two probability distributions. It would then be possible to determine the number of outliers from the assumed distribution, which can be used to determine the outlier probability. Unfortunately, if the expected conditions — such as the permissible distribution types — are not met, the automatic probability estimation could fail or yield unreasonable results. Finding solutions for the automatic outlier probability determination and dealing with the aforementioned challenges would be highly interesting for future research.

### 5.4.4 Modified RANSAC

The number of required iterations, as calculated in equation 5.63, can be reduced, or the probability for the same number of iterations can be increased, if the RANSAC algorithm uses more of the available information. With the standard RANSAC algorithm, each point has the same probability of being selected from the full point set. For  $N$  points, the probability is  $P_{\text{Select}} = 1/N$ ,  $N \in \mathbb{N}$ . The probability of choosing an inlier, is the sum of all probabilities for selecting an inlier, divided by the sum of all selection probabilities  $P_{\text{Select}}$ . It thus follows:

$$P_{\text{Select,Inlier}} = \frac{\sum_{\text{Inlier}} P_{\text{Select}}}{\sum_{\text{Points}} P_{\text{Select}}} \quad (5.64)$$

In the case of a uniform selection probability distribution, it follows from equation 5.64:

$$P_{\text{Select,Inlier}} = \frac{n_{\text{Inlier}}/N}{N/N} = \frac{n_{\text{Inlier}}}{N} \quad (5.65)$$

There are two reasons why a selected point could be an outlier:

1. It truly is a discontinuity in the surface (see Figure 5.18a on page 114).
2. It resulted from a faulty measurement.

It was shown in subsection 5.3.2 that intensity is a measure of trust, meaning that outliers of the second kind are expected to have a lower intensity. Therefore, points with higher intensities should be preferred over those with lower intensities because they have a higher likelihood of being an inlier. Two tasks arise from this observation: first, the RANSAC algorithm needs to be modified to use intensities; second, it needs to be analyzed how the modification improves the regular RANSAC algorithm.

In stochastic terms, what is needed is a suitable probability density function (PDF). Any PDF has two requirements:

1.  $0 \leq I \leq 1$ ,  $\forall$  Intensities  $\Rightarrow \int_{-\infty}^{\infty} f_X(\xi) d\xi = P(\Omega) = 1 = \int_0^1 f_X(\xi) d\xi$
2.  $F_X(\xi)$  increases monotonously  $\Rightarrow f_X(\xi) \geq 0$  for  $0 \leq \xi \leq 1$

For instance, for a linear relationship these requirements would lead to the following probability density function:

$$\begin{aligned} f_X(I) &= m \cdot I \\ \Rightarrow \int_0^1 f_X(I) dI &= \left[ \frac{1}{2} m \cdot I^2 \right]_0^1 = \frac{1}{2} m \equiv 1 \\ &\Leftrightarrow m = 2 \\ \Rightarrow f_X(I) &= 2 \cdot I \end{aligned}$$

The sensor operates with a resolution of 10 bits, meaning that there are 1024 discrete intensity levels. Because the data is discrete, it is no longer possible to describe the stochastic analysis in terms of probability densities, but rather as a probability function:

$$\begin{aligned}\Delta I &= \frac{1}{1024} \\ \Rightarrow P_k &= f_X(k) \cdot \Delta I\end{aligned}\quad (5.66)$$

This means that it is possible to create a lookup table (LUT). Since there is an LUT to yield the expected probability contributions from the intensity levels, it is possible to generate the relationship between intensities and their LUT indices. These then yield the probability contributions.

The probability of choosing an inlier is:

$$P_{\text{Inlier}} = \sum_{\text{Inlier}} f_X(i) \quad (5.67)$$

An example is given for a linear relationship between the intensity levels and the probability, for a polynomial of first order:

$$P_{\text{Inlier,linear}} = \frac{1 \cdot 0.75 + 5 \cdot 0.8 + 3 \cdot 0.85 + 1 \cdot 0.9}{1 \cdot 0.4 + 1 \cdot 0.75 + 5 \cdot 0.8 + 3 \cdot 0.85 + 0.9} = 94\%$$

For a uniform distribution the probability of choosing an inlier would have been:

$$P_{\text{Inlier,uniform}} = \frac{1 + 5 + 3 + 1}{1 + 1 + 5 + 3 + 1} = 90.9\%$$

With equation 5.62 and  $K = 3$ , it follows that:

$$\begin{aligned}P_{\text{Success,linear}} &= 1 - (1 - P_{\text{Inlier,linear}}^2)^3 = 99.84\% \\ P_{\text{Success,uniform}} &= 1 - (1 - P_{\text{Inlier,uniform}}^2)^3 = 99.5\%\end{aligned}$$

Thus, the probability of success can be improved if the intensities are taken into account. The resulting polynomial for the problematic input in Figure 5.18a is shown in Figure 5.19a and the filtered scan in Figure 5.19b. The discontinuities are removed, and the surface form is properly fitted, as desired. A prerequisite for the algorithm is that the erroneous part of the scan is smaller than the amount of valid data. Otherwise, the valid data part is considered to be the outlier part and is therefore removed. After the surface form has been removed, the remaining data can be mapped into the image range, as shown in Figure 5.15c.

### 5.4.5 Special Case: S-Curve Detection

Most panels in the part are both vertical and relatively tall. Hence, if they are scanned with an orthogonal scanner alignment and with limited scanning depth, neighboring panels are invisible, as long as

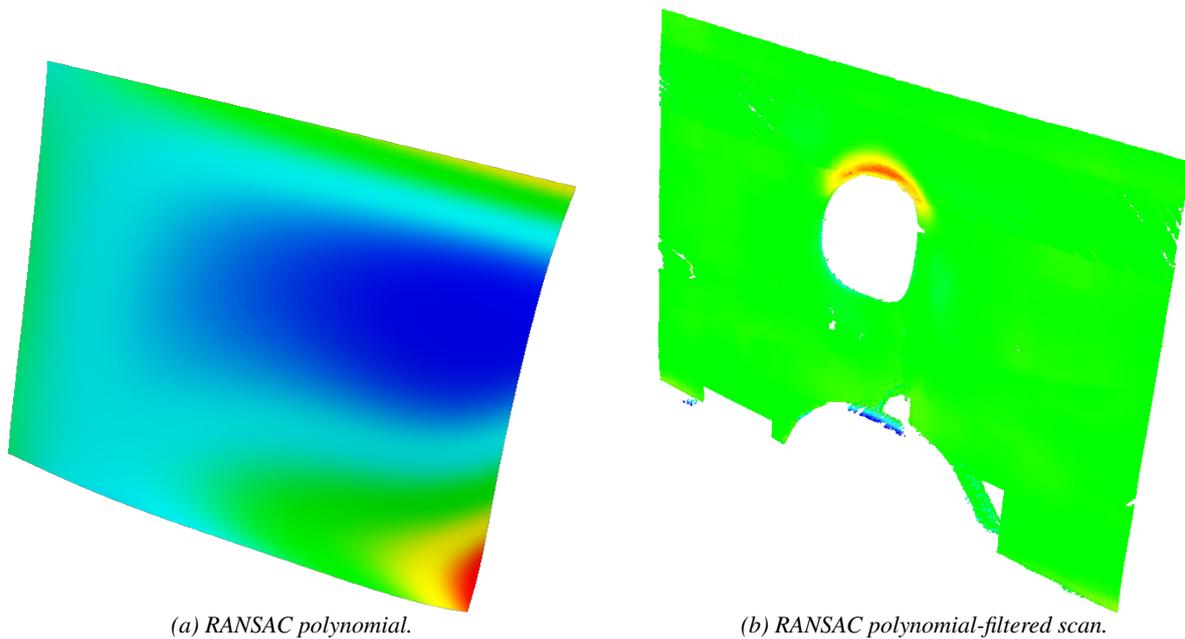


Figure 5.19: The RANSAC polynomial-filtered problematic scan (see Figure 5.18a) and the corresponding polynomial (left).

the path adaptation is working properly. In those cases, no particular care has to be taken to separate the current panel from unwanted data captured from another panel. Even if another panel were to be captured, the associated jump would allow for relatively easy clustering.

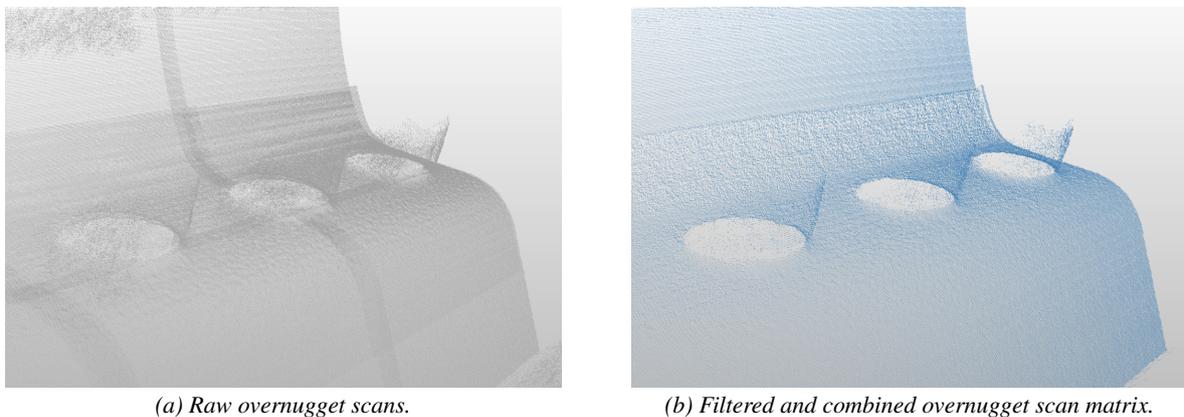


Figure 5.20: Example of an overnugget scan.

However, the connecting panels on the outside of the CFM-56 outer liner, the so-called overnuggets, are horizontal and relatively narrow. Due to their orientation they have to be scanned in a way that usually also captures their two neighboring panels, with which they are connected in an S-like shape. In order to generate a suitable image, data from the neighboring panels has to be detected and removed. An example of a raw overnugget scan is shown in Figure 5.20a and the filtered and combined scan matrix is shown in Figure 5.20b.

Since the S-shape is contained in all columns of the scan matrix, it is not necessary to perform the S-shape detection on the entire matrix but rather on a single column. However, not all columns are equally suitable for shape detection, because some of them may be interrupted by damages or features. Since overnuggets typically contain a large number of small cooling holes (see for instance Figure 5.20b), this is particularly likely. Furthermore, every single column can be expected to deviate to a certain degree from the average profile. Thus, a certain amount of suitable columns is selected and averaged to yield an approximation to the true S-curve profile.

Suitable columns are defined by the following criteria:

1. Large number of valid pixels.
2. Well outside areas with invalid pixels.

The first criterion is based on the absolute number of valid pixels, rather than the percentage, because a column might be at the very edge of the scan and would, therefore, be comparatively short (it might not contain the full S-profile). The second criterion is just as important because it ensures that columns that include invalid pixels are not taken into account. It is important to also exclude columns that are in the vicinity of invalid pixel areas, because a column at the very edge of a feature may be heavily deformed due to the feature's edge.

The mean profile is extracted as follows. First, a score is incremented for each column for every valid pixel, starting from zero. Additionally, each column's neighborhood of a predefined size is checked for whether it contains invalid pixels. If any invalid pixels are found, the column is excluded, otherwise it is added to a list containing tuples of scores and column indices. Afterward, a defined number of extracted columns are averaged to generate the mean profile.

After the mean profile is created, a median filter is applied to remove any shot-like noise that might be present. This is followed by a simple outlier rejection filter, which considers every point's two neighbors and removes the pixel if the deviation is too large. In the next step, a polynomial of sufficiently high order is fitted to the filtered profile and is then used to generate a smooth representation of the data. The resulting polynomial profile is an approximation of the S-shape. The shape can then be used to determine where the desired flat area begins and ends through analyzing the curvature.

The curvature analysis is performed by going through the profile and calculating the curvature using point differences and the  $\text{atan2}(\Delta y, \Delta x)$  function. Since the expected angular changes are known, this can be used to determine the plateau indices. These indices are subsequently applied to all columns of the scan matrix. If the resulting indices are too close together, which would result in a scan matrix that is too narrow, the determined indices are discarded, and the full matrix is processed instead.

## 5.5 Feature Detection

As mentioned earlier, objects under test may contain deliberately manufactured features that might be interpreted as damages by the novel damage-inspection system. For instance, the CFM-56 outer

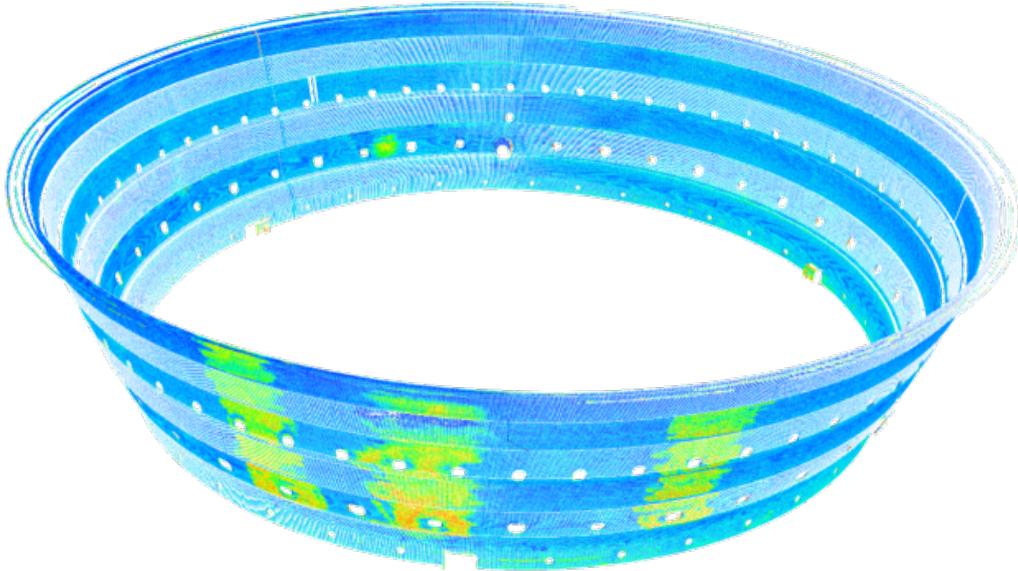


Figure 5.21: Laser line section scan of a CFM-56 combustion chamber outer liner. The intensity is color-coded (image from [Ott+15]).

liner contains several holes to optimize the airflow. Since there is a large variety of sub-models on the market, and the part itself might have been flown and repaired before, there is no robust model for where those features are located. Therefore, the features have to be detected in the data itself. Since the features are relatively large, a multi-modal approach was chosen to detect the features:

1. The LLS data captured by the handling system for path adaptation<sup>1</sup> (see Figure 5.21) is used to roughly locate the features.
2. Feature templates are registered to the WLI data in the image domain.

### 5.5.1 Laser Line Section Data Processing

Unlike the interferometer data, the laser line section data is only equidistant in one dimension (along the laser line) and only for the extent of a single scan line. However, the rotational symmetry of the part can be used to detect the features. Given the shape of the CFM-56 combustion chamber liner, cylindrical coordinates are a meaningful choice, however, the data is not necessarily centered around the origin. Because of this, the true center of the point cloud is calculated and then the point cloud is shifted to the origin. Afterward, every point in the cloud is converted into cylindrical coordinates.

Next, the point cloud is rasterized by combining defined extensions of  $z$  and  $\theta$ , as follows:

$$P_{\text{Rasterized}} = \left( \frac{\sum_{i=1}^N \rho_i}{N}, \frac{\theta_{\text{Right}} - \theta_{\text{Left}}}{2}, \frac{z_{\text{Upper}} - z_{\text{Lower}}}{2} \right) \quad (5.68)$$

<sup>1</sup>As mentioned before, the handling system and path adaptation are not part of this dissertation.

The  $N$  radii  $\rho_i$  in each raster location are averaged. After this process has been completed, the rasterized point cloud is searched for missing points. If a defined number of points is missing in an area, that area is considered to be a potential feature. The potential features' center points are determined by calculating the center point  $C$  of the surrounding points.

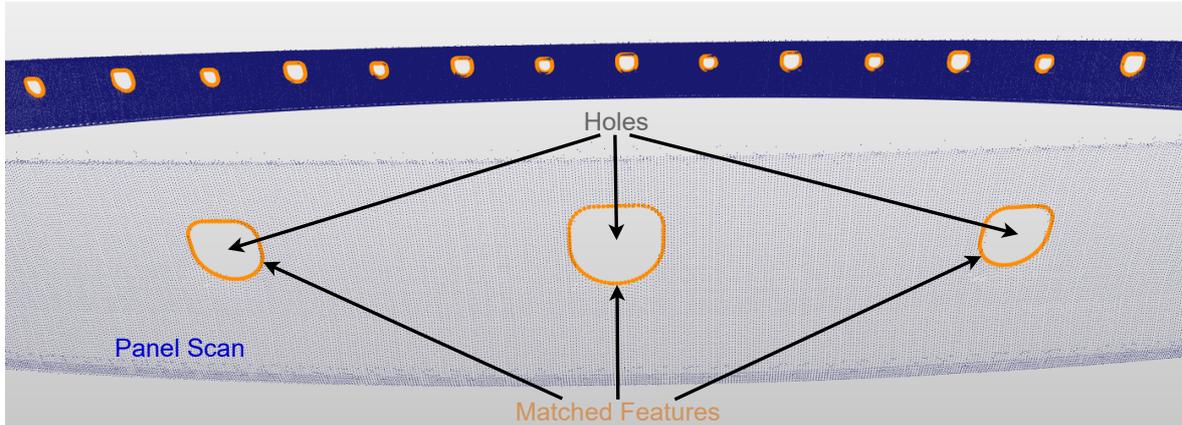


Figure 5.22: Example of feature detection for a single panel. Blue: scanned panel; orange: features.

Then all points surrounding the center points  $C$  in the original (un-rasterized) point cloud within a defined radius are extracted, and predefined templates for the expected damages are matched using the iterative closest points (ICP) algorithm (see subsection 5.2.1 for more information on the ICP). If the matching error is too large for all considered templates, the point cloud defect is no longer considered to be a feature. Otherwise, the best-fitting feature is assigned to that location. Figure 5.22 shows an example result for a single panel. The lower part of the image shows three holes (transparent) in the front of the panel with matched features (orange). The panel scan itself is blue. The upper part of the image shows part of the back of the scanned panel with more holes and matched features. It can be seen that the feature matching works as expected.

The panel edges are also detected based on the LLS point cloud. Those edges are less localized than cooling holes and are rather a feature of the entire panel. Thus, the average along  $\theta$  is calculated, resulting in a two-dimensional ( $\rho$ - $z$ ) profile. Afterward, the profile is analyzed for radial jumps, signifying the end of a panel.

## 5.5.2 Template Matching in the Image Domain

Since both the detected features and the white light interferometer scans are in a common coordinate system, it is straightforward to associate the former with the latter. In other words, features can be mapped into the combined interferometer scans in three-dimensional space. Thus, if the same transform that is applied to the interferometer scans to map them into 2D space is applied to the feature, it will also be mapped into that plane. By treating the template's points as a closed polygon, it can be flood-filled, which results in an image representation of the mapped feature template.

In principle, this means that the feature-template image can be used to manipulate the crack-detection input image to prevent the image processing algorithm from searching for cracks in that area. However, there may be a discrepancy between the laser line section scan and the interferometer data, particularly if a highly non-linear actuator such as a robot was used to guide the sensor. Therefore, in order to correctly cover the feature area in the input image with the template image, a fine registration between the template and the image should be performed in the image domain.

The registration could be performed with a two-dimensional variant of the ICP algorithm. However, only a relatively small part of the feature might be visible in the image, which leads to unreliable results using the ICP algorithm. Furthermore, observations over a relatively long period of time have shown that the misalignments are mostly lateral. Thus, it was found that using a discrete cross-correlation approach is both fast and reliable.

It was mentioned in section 5.4 that unassigned pixels are blue, filtered pixels are green, and regular pixels are gray, with the gray level being representative of the height, see Figure 5.15c. Since the template is a synthesized image and the feature area is (in the best case) purely green in the interferometry image, it is not necessary to perform a full discrete correlation. Rather, it is sufficient to generate binary images for both the template image and the interferometer scan data, where zeros represent unassigned values or regular surface pixels, and ones represent potential feature data. The template is then simply shifted over the image, and the sum of the point-wise multiplications of the template with the image is determined. Since the images are binary, the multiplications can be reduced to logical *and* operations and can, thus, be implemented very efficiently. Therefore, the two-dimensional Boolean discrete crosscorrelation between the scan and the template can be expressed as follows:

$$R_{st}(x, y) = \sum_{u=1}^N \sum_{v=1}^M s(x, y) \wedge t(x + u, y + v) \quad (5.69)$$

This alignment is not performed for the entire image but only within a pre-defined radius around the original match. Furthermore, if required, the matching process can be sped up by scaling down both the image and the template image before performing the search. A compromise between speed and accuracy can be achieved using a coarse-to-fine approach: first, an improved starting position is found using the reduced data set; second, the algorithm is repeated with the full data set but with a reduced radius. The effect of the described algorithm is illustrated in Figures 5.23 (a) and (b).

## 5.6 Image Processing

When the image-generation procedure described in the previous section is finished, the resulting images can be processed with classical image-processing techniques, which are detailed in this section.

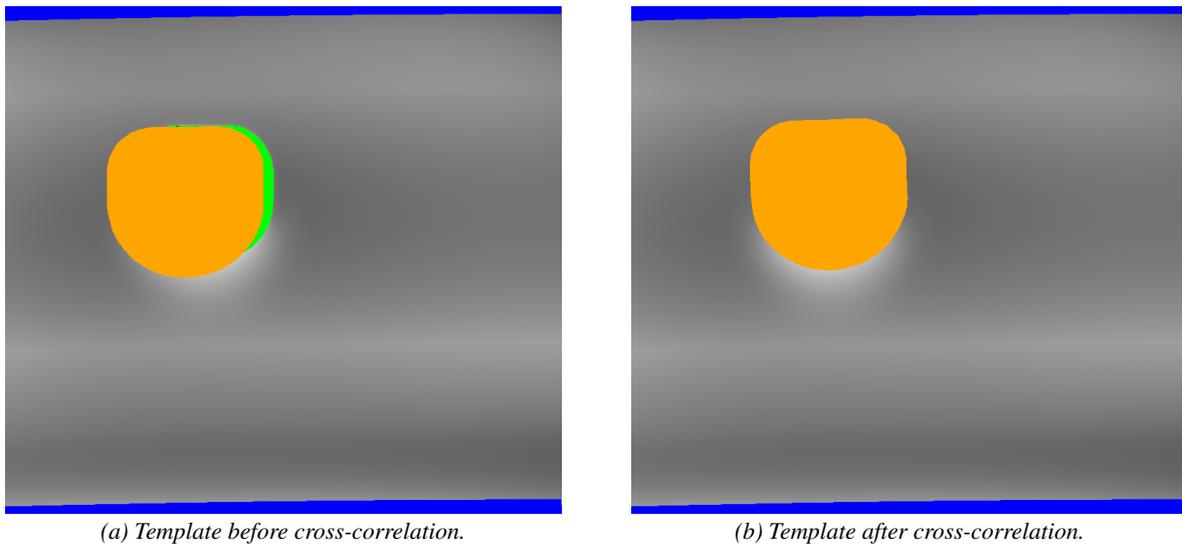


Figure 5.23: Example of a template immediately after being transformed from 3D (left) and after it has been registered using the cross-correlation approach (right).

### 5.6.1 Image Preparation

Before the actual image processing can be performed, the image has to be prepared in order to avoid incorrect findings. First, unmapped edges that might surround an image represent strong jumps in relationship to the actual surface. Furthermore, these edges can be line-like and can extend far. They might therefore cause false positives if not taken care of. Unfortunately, it is insufficient to merely remove the crack findings in those areas retrospectively, because the false edge positives can cause nearby findings to be connected, potentially leading to more false positives.

Two approaches are used to deal with this issue. The missing data in the top and bottom area are addressed with the panel-edge-detection approach based on the LLS data. These feature areas are excluded from the image processing, see for instance the blue edge templates on top and bottom in Figure 5.23b.

However, there can also be unmapped data areas on the side of the image due to distortions. These missing side mappings are masked using a surface extension approach. The algorithm goes through the image in a row-wise fashion from left to right and right to left. It finds the first and last valid pixels in each row, up to a definable maximum search length. For each of those pixels, the median value of a specified number of following pixels in that row is calculated. The result is assigned to the invalid pixels on the left and right respectively. The algorithm is illustrated in Figure 5.24 on the following page.

Next, masks are generated from the color channels. Gray pixels (representing real surface points) have the same intensity values in all three channels: red, green, and blue (RGB). However, in the case of green (filtered) and blue (unassigned) pixels, only the green and blue channels contain non-zero

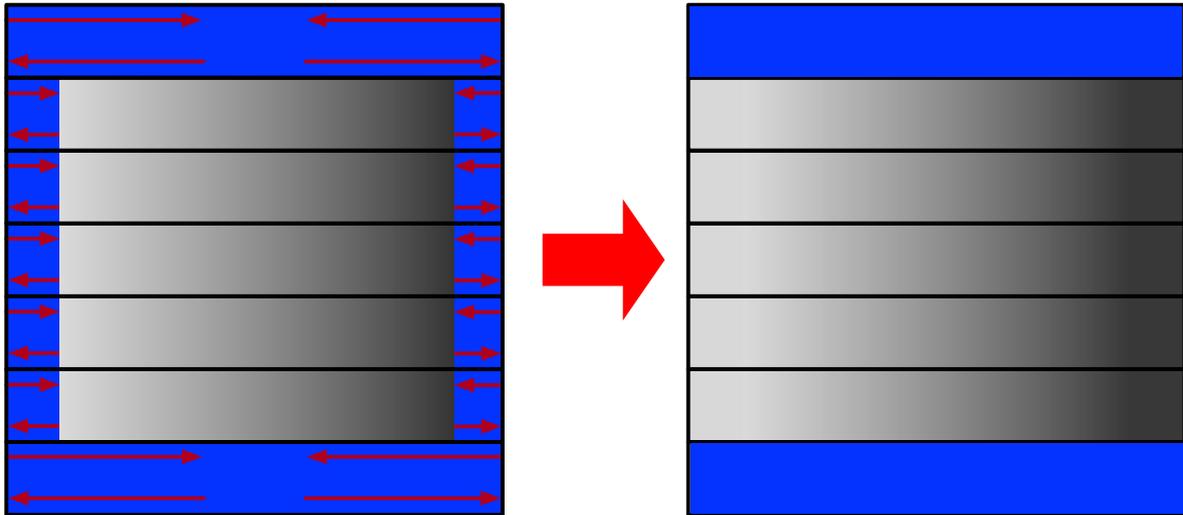


Figure 5.24: Illustration of the surface extension algorithm.

data. Thus, masks can be generated as follows:

$$\begin{aligned}\text{Mask}_{\text{Filtered}} &= G - R \\ \text{Mask}_{\text{Unassigned}} &= B - R\end{aligned}$$

where, if the previously described surface extension algorithm worked perfectly, the unassigned mask should be empty. The filtered mask is mainly used at the end of the image-processing sequence to make sure that all of the filtered values are included as damages.

## 5.6.2 Pseudo High-Pass Filtering

As explained in subsection 5.4.2, the low-frequency contents are irrelevant for crack detection. The point-based polynomial regression filter described in the previous section already removes a significant amount of lower frequency information, which allows for intelligent use of the dynamic range of images with eight bits per channel. However, the image could still contain unwanted low-frequency contents, which should therefore be removed by a high-pass filter in the image domain. This is accomplished through convolution.

Simply put, a two-dimensional convolution filter considers every pixel's  $n \times m$  neighborhood. For instance, a  $3 \times 3$  filter kernel would be a matrix containing nine elements, centered around the pixel under consideration. In the case of a mean filter, each pixel is to be replaced by the average value in its neighborhood. In the case of a  $3 \times 3$  mean filter each matrix element would therefore be  $1/9$ . This operation is not only performed for a single pixel but rather the filter kernel is shifted over the entire image, and the operation is repeated for every single pixel. The filtering process is therefore a discrete

convolution between the image  $\mathbf{I}$  and filter kernel  $\mathbf{K}$ :

$$\tilde{\mathbf{I}}(x, y) = \mathbf{I}(x, y) \star \mathbf{K}$$

Since higher frequency components are expected to vary strongly within their neighborhood, replacing the pixel value by its neighborhood's average, therefore, retains the slowly changing parts. As explained previously, it thus acts as a low-pass filter.

However, the basic mean filter is not a very good low-pass filter (see for instance [Jäh12, pp. 339]). A superior class of filters are binomial filters, which can be generated through multiple convolutions of the basic filter mask  $[1 \ 1]/2$  [ibidem, p. 342]. For instance, for a one-dimensional filter of length three, such a filter can be generated by a single convolution:  $[1 \ 1]/2 \star [1 \ 1]/2 = [1 \ 2 \ 1]/4$ . Two-dimensional filters can be generated by calculating the outer product of two one-dimensional filters:  $\mathbf{F}(x, y) = \mathbf{F}(x) \cdot \mathbf{F}(y)^T$ . For large filter orders, binomial filters approach a Gaussian filter of equivalent variance [ibidem, p. 343].



Figure 5.25: Illustration of pseudo high-pass filtering using a binomial filter [Ott+14].

If the image is convolved with a sufficiently large binomial filter, the resulting low-pass filtered image does not contain any of the higher frequency content of the input image. Thus, if the filtered image is subtracted from the input image, what remains are only the higher frequency contents, resulting in a pseudo high-pass filter (see Figure 5.25).

### 5.6.3 Edge Detection

After the image has been prepared through pseudo high-pass filtering, the crack edges can be extracted using an edge detector. Edge detectors are based on an image's first- or second-order derivative [ibidem, pp. 367]. There are several options for edge detection, which are typically readily available in image-processing libraries. An optimal edge detector was found by Canny [Can86]. Canny formulated his optimality criteria as follows [ibidem, p. 680]:

1. Good detection
2. Good localization

## 3. Single response

After numerically solving for the shape of an operator that fulfills these criteria, he found that the optimal operator can be approximated by the first derivative of the Gaussian [Can86, p. 687]. Canny proposed to follow the edge detection with a hysteresis-edge-following procedure, based on two thresholds [ibidem, p. 690]. For two dimensions, it is also important to detect the direction of the edges. Canny proposed to solve this by first convolving a two-dimensional Gaussian filter kernel with the image, followed by convolving the image with two orthogonal directional differentiation operators, which allows for calculating the direction of the edge [ibidem, p. 691]. Due to the commutativity of the convolution operator, separating the optimal edge detector into two operators is equivalent:  $\tilde{\mathbf{I}} = \mathbf{I} \star (\mathbf{K} \star \mathbf{G}) = \mathbf{I} \star \mathbf{K} \star \mathbf{G}$ . The implementation of the Canny edge detector uses a Sobel operator pair [Ope17]:

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (5.70a)$$

$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad (5.70b)$$

Since the Sobel operators (equations 5.70a and 5.70b) form an orthogonal base, their combination allows for the extraction of magnitude and direction, as desired. Afterward, non-maxima are suppressed using four discrete directions and the two-threshold hysteresis is applied [ibidem]. An example of the

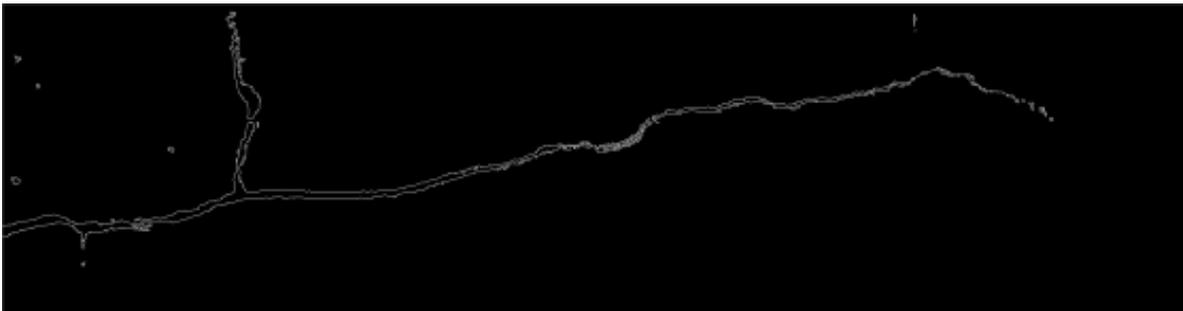


Figure 5.26: Output of the Canny edge detector applied to the image in Figure 5.25 [Ott+14].

output of the Canny edge detector is illustrated in Figure 5.26.

Despite the single response criterion suggested by Canny, a crack with a large enough width extent still results in a second edge at the other side of the crack, as can be observed in Figure 5.26. In order to obtain a complete representation of the crack, it is desirable to fill the full crack area. This can be achieved with a class of shape-affecting operators commonly used in image processing for such tasks, called morphological operators [Jäh12, pp. 555]. Simply put, morphological operators are

logical operators that can be applied to binary images, such as the output of a Canny edge detector. A basic morphological operation is performed by shifting a binary mask over the image and performing a given logical operation, such as a logical *and* or logical *or* operator. The two fundamental morphological operations are:

- An operator that removes parts of the image that are not completely covered by the binary mask: morphological erosion.
- An operator that fills parts of the image with the mask shape provided there is at least one pixel underneath the mask: morphological dilation.

The use of morphological erosion is to remove thin or small areas of the image, to separate objects, and to thin objects. Morphological dilation, on the other hand, is used to fill voids in structures and to increase their extent. In set notation, the operations can be expressed as follows (from [Jäh12, p. 558]):

$$\text{Erosion: } \mathbf{G} \ominus \mathbf{M} = \{p : \mathbf{M}_p \subseteq \mathbf{G}\} \quad (5.71a)$$

$$\text{Dilation: } \mathbf{G} \oplus \mathbf{M} = \{p : \mathbf{M}_p \cap \mathbf{G} \neq \emptyset\} \quad (5.71b)$$

where  $\mathbf{G}$  is the input image,  $\mathbf{M}$  is the morphological mask and  $\mathbf{M}_p$  is the translated mask.

The basic operations described in equations 5.71a and 5.71b can be combined in various ways to achieve more complex operations, for more details see for instance [ibidem, pp. 555]. Given the previous descriptions, it follows that an operation that allows filling the crack edges can be constructed using morphological dilation. Such an operation would respond to every edge pixel by filling the area with the operator's binary mask, which would fill the crack, given a suitable binary mask. However, the dilation operation itself would significantly extend the crack's area because the operator does not only fill the inside but also increases the outside of the crack. To remedy the enlargement of the crack, the image can be eroded with the same binary mask in a second step. This second step will remove the additional area added in the first step; however, areas that were closed in the first step remain closed. This combination of operations is known as morphological closing [ibidem, p. 562]. For completeness, the morphological closing operator's complement, namely morphological opening, shall be briefly described as a morphological erosion operation followed by morphological dilation. The morphological opening and closing operators can thus be written as follows [ibidem, pp. 561]:

$$\text{Opening: } \mathbf{G} \circ \mathbf{M} = (\mathbf{G} \ominus \mathbf{M}) \oplus \mathbf{M} \quad (5.72a)$$

$$\text{Closing: } \mathbf{G} \bullet \mathbf{M} = (\mathbf{G} \oplus \mathbf{M}) \ominus \mathbf{M} \quad (5.72b)$$

If the crack area is to be emphasized, it can be beneficial to perform the dilation with a larger binary mask than for the erosion. This results in the regular closing output but with a controlled extension of

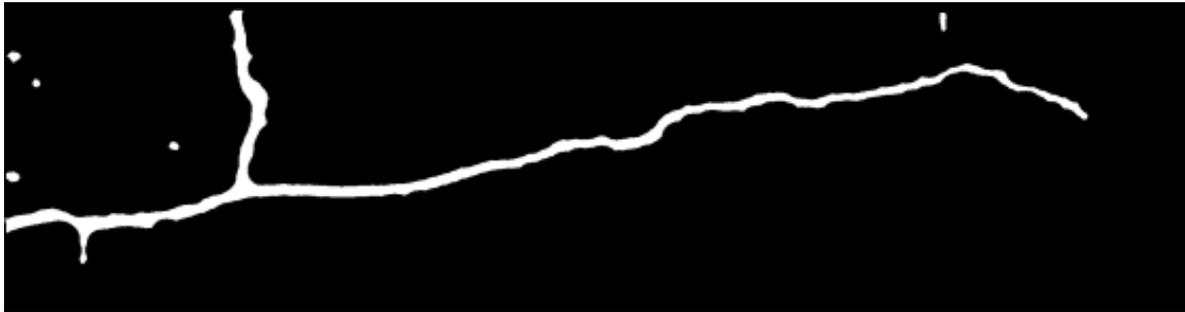


Figure 5.27: Effect of morphological closing applied to Figure 5.26 [Ott+14].

the crack area. This operation is defined here as follows:

$$\text{Extended Closing: } \mathbf{G} \blacklozenge \mathbf{M} = (\mathbf{G} \oplus \mathbf{M}^+) \ominus \mathbf{M} \quad (5.73)$$

where  $\mathbf{M}^+$  is the larger binary mask. The effect of extended morphological closing (equation 5.73) is shown in Figure 5.27.

#### 5.6.4 Crack Thinning and Segmentation

After the extended morphological closing step (described in the previous subsection) is applied to an edge image, the resulting image only contains potential crack indications. In order to create additional relevant output for subsequent repair processes but also for the crack verification process (see section 5.7), it is important to extract the cracks' center lines and break the cracks up into crack segments.

A parallel method for crack thinning (also known as skeletonization), based on two sub-iterations, was introduced by Z. Guo and R. Hall [GH89]. Their algorithm "A1" preserves curve-like features while providing appropriate thinning results. The basic idea of the algorithm is to iteratively remove pixels that fulfill certain conditions, until no more deletions occur. Guo and Hall identified three necessary criteria for pixel removal. The first criterion is concerned with checking if a pixel is eight-simple [ibidem, pp. 360]. This means that in the pixel  $p$ 's eight neighborhood, there is only precisely one side neighbor that fulfills all of the following criteria:

1. The side neighbor itself is zero.
2. At least one of the two clockwise following neighbors after that side neighbor (in  $p$ 's eight neighborhood) is non-zero.

Guo and Hall formulated the eight-simple criterion in the following expression [ibidem, p. 361]:

$$C(p) = \bar{p}_2 \wedge (p_3 \vee p_4) + \bar{p}_4 \wedge (p_5 \vee p_6) + \bar{p}_6 \wedge (p_7 \vee p_8) + \bar{p}_8 \wedge (p_1 \vee p_2) = 1 \quad (5.74)$$

This has several implications, the most important of which are that the algorithm works from the outside in (because a pixel inside a filled area has no non-zero side-pixels) and that no objects will be



- Even:  $(p_6 \vee p_7 \vee \bar{p}_1) \wedge p_8 = 0$

An example of the algorithm's output can be seen in Figure 5.28 on the preceding page.

Finally, the crack skeleton  $C$  extracted in the previous algorithm is to be subdivided into crack segments. The author defines the following criteria to define a crack segment  $S$ :

1.  $S \subseteq C$ .
2. A crack segment has either one (single pixel) or two (regular crack) end points.

It follows from the second criterion that a crack segment ends (or starts) whenever a crack forks. If a crack does not contain any forks, it, therefore, follows that  $S = C$ , otherwise  $S \subset C$ . With the given

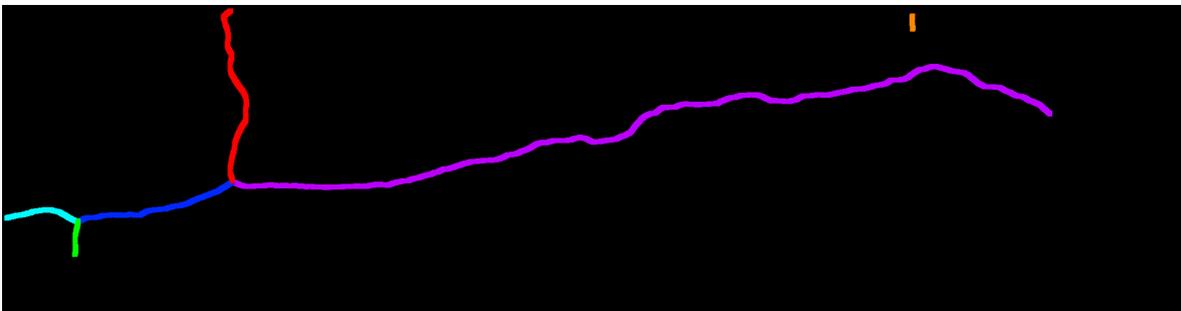


Figure 5.29: Segmentation algorithm applied to Figure 5.28 [Ott+14].

definition, it is possible to divide the crack image into crack segments simply by considering the crack pixels' neighbors. Figure 5.29 illustrates how such an algorithm would split the crack image shown in Figure 5.28 into crack segments.

### 5.6.5 Crack-End Continuation and Crack Connection

The Canny edge detector relies on two thresholds to perform its edge detection. These thresholds allow manual tuning of the crack-detection process. In smooth materials with only occasional and very clearly defined cracks, a low threshold can be used, allowing for crack detection that is both highly reliable and has a low number of false positives. In reality, however, the surfaces are often relatively coarse, resulting in a high noise floor. This means that in practice the threshold has to be set much higher in order to avoid an exorbitant amount of false positives. Furthermore, the cracks are often not very well-defined, especially not over the entire length of the crack. This has two consequences:

1. A crack might actually continue post the detection end: it was not detected in full because the threshold was too high.
2. The detected crack might incorrectly show up as several unconnected cracks. At worst, each of these crack parts might be so small that it is removed in the crack verification step, whereas the full crack would not have been eliminated.

### Crack-End Continuation

The problem of not detecting the full length of cracks can be caused by two factors: either the sensor resolution was too low, or the chosen threshold was too high. In the first case, the only remedy would be to select a higher resolution sensor. In the second case a lower threshold might deliver the expected results. Unfortunately, as was already stated, this will also increase the impact of noise — and thus the number of false positives — to a potentially unacceptable amount.

The higher impact of noise, caused by the increased level of sensitivity, can be limited by decreasing the scope of influence. This means that globally, a higher threshold — delivering stable results with a good detection and a low false positive rate — is chosen; whereas locally, much lower thresholds can be used. The areas of increased sensitivity are defined by the end points of detected cracks.

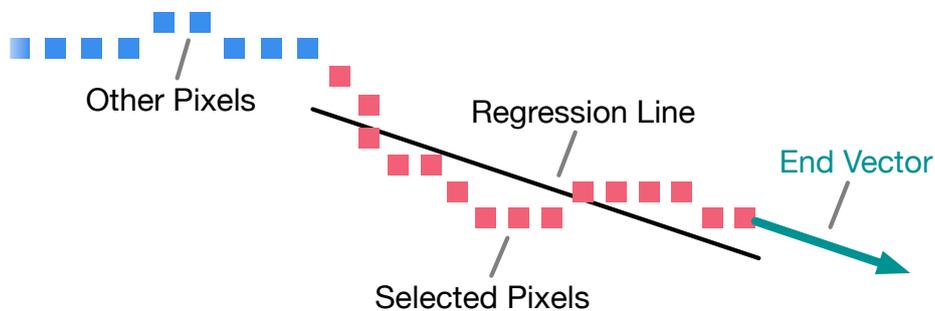


Figure 5.30: Illustration of the crack-end vector calculation [Ott+14].

Whereas the ends of cracks are simply crack pixels with only one neighbor and are thus directly identifiable, the direction of the crack end has to be calculated. As can be seen in Figure 5.29 on the preceding page, cracks tend to change their direction throughout the course of their progression. Thus, it is not meaningful to calculate the average orientation of the entire crack but rather only of their end sections. On the other hand, the number of pixels used to calculate the orientation should not be too small because it might then be strongly influenced by outlier pixels. Hence, a suitable number of pixels is selected to perform a line regression. Afterward, an end vector is constructed, which is parallel to the regression line, pointing away from the line points and starting at the last crack pixel (see Figure 5.30).

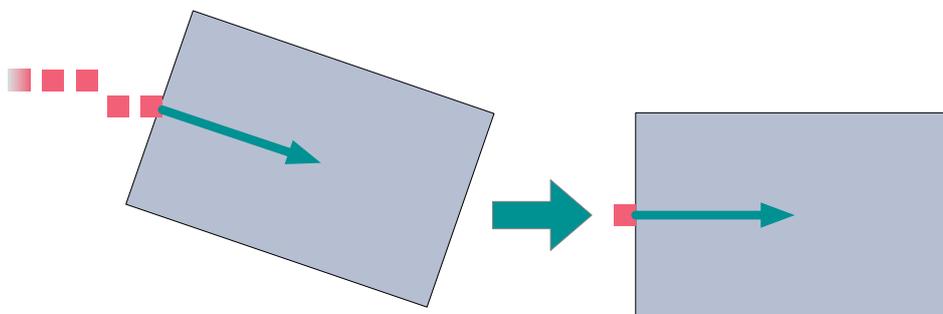
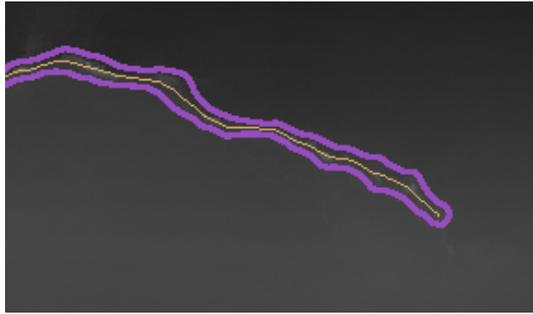
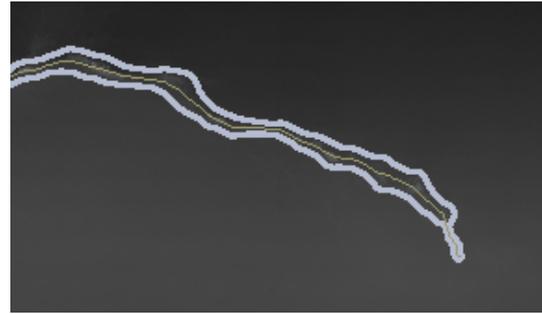


Figure 5.31: Illustration of the end-vector area mapping into a new image [Ott+14].

Afterward, a rectangular area of predefined size is oriented with the end vector in such a way that one of its sides is centered around, and is perpendicular to, the end vector. The pixels inside that area are then mapped into a new image (see Figure 5.31 on the preceding page). The resulting new image is then checked for cracks using lower thresholds, resulting in finer results. The increased number of artifacts caused by the lower threshold are limited in location due to the described mapping method. The findings are then transformed back into the original image.



(a) Example of a crack-detection result [Ott+14].



(b) Example of a crack-detection result with enabled crack-end continuation [Ott+14].

Figure 5.32: Illustration of crack-end continuation.

The difference between the original findings and the enhanced findings using crack-end continuation is shown in Figure 5.32.

A method that is classically used for the detection of structures in images is the Hough transform,<sup>1</sup> patented by Paul Hough in 1962 [Hou62] and reformulated by Richard Duda and Peter Hart in 1972 [DH72]. Here, the task of finding co-linear points in an image is replaced by finding lines in a parameter space. The original Hough transform was introduced for the detection of lines. The parameter space suggested by Duda and Hart was the  $(\theta, \rho)$  space. In that space,  $\theta$  represents the slope and  $\rho$  represents the distance to the origin, meaning the lines are described in the Hesse normal form [ibidem]:

$$\rho = x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta) \quad (5.78)$$

In order to ensure a timely computation, a reasonable discretization for both the distance and slope has to be selected. The discrete parameter space is then implemented as a two-dimensional array (a matrix) of size  $[n_\rho \times m_\theta]$ , where  $n_\rho$  is the number of discrete distances and  $m_\theta$  is the number of discrete angles. Each input pixel with coordinates  $(x_i, y_i)$  is then transformed into the parameter space using equation 5.78 by sweeping through the  $\theta$ -range. This is to say, for each discrete value of  $\theta$ , a distance value  $\rho$  is calculated using equation 5.78, with  $x_i$  and  $y_i$  as parameters and  $\theta$  as the variable. This results in matrix index pairs  $(\rho_j, \theta_j)$ , which are used to increment the matrix in those locations.

<sup>1</sup>It should be noted that the Hough transform is closely related to the Radon transform mentioned in subsection 2.1.4 in the context of computed tomography. Both transforms are concerned with transforming an input space into a parameter space. However, whereas the Hough transform pushes the information from the image space into the parameter space, the Radon transform pulls the information from the image space from the point of view of the parameter space. Additionally, the Radon transform is an integral transform and is therefore continuous in nature.

Since co-linear points in the image will generate the same indices in the parameter space, line-like structures will generate maxima. This is illustrated using an artificial “crack” image in Figure 5.33.

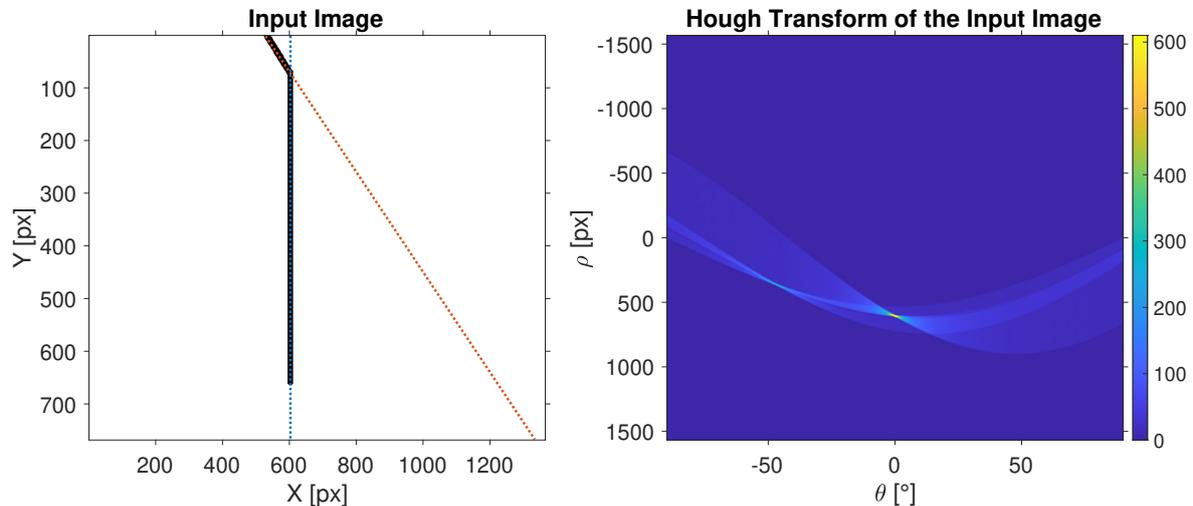


Figure 5.33: Illustration of the Hough transform. The left image, which contains an artificial “crack,” is transformed into the discrete parameter space on the right. The parameter space has two maxima, corresponding to the two lines in the image space. The extracted lines were then drawn into the left image (blue and red).

The left image contains two lines, resulting in two maxima in the discrete parameter space on the right. The maxima in the discrete parameter space can then be extracted, which immediately leads to a line representation in the image space, illustrated in blue and red in Figure 5.33 on the left. Unfortunately, the Hough transform does not give any information about where the corresponding pixels are located in the image domain, only that they are on the extracted lines.

The Hough transform could be used as a primary means to detect cracks. Furthermore, because the Hough transform delivers entire lines, it could be used to define search areas outside of the already extracted cracks, similar in effect to the crack-end continuation method described in this subsection. The use of the Hough transform was evaluated in the beginning of the project. Cracks are only line-like<sup>1</sup> and somewhat unpredictable in nature. Therefore, the parameter space becomes difficult to evaluate because it becomes relatively cluttered and the number of cracks is unknown. Furthermore, although the Hough transform can be extended to work on gray scale images, the input images are relatively noisy. It is, therefore, advisable to suppress unwanted structures first. Unfortunately, the Hough transform approach proved ultimately unsuccessful, leading to the development of the crack detection and crack-end continuation method presented in this dissertation.

### Crack Connection

Similar arguments as those for crack-end continuation can be made for why it might be reasonable to connect cracks. The sensor might simply have too low a resolution, or the threshold might have

<sup>1</sup>Although the Hough transform can be extended to support general parameter spaces, the actual shapes are unpredictable. Therefore, lines are still the best choice, especially because the parameter space is only two-dimensional.

been too high for the detection of in-between parts of certain cracks. Since each of the interrupted crack parts is considered to be a crack in its own right, the crack-end continuation approach presented previously is applied to them, which might cause the cracks to be connected automatically. However, this is not necessarily guaranteed behavior but rather depends on the specific circumstances.

The question is, therefore, whether all cracks should be connected if certain conditions are met. In order to analyze that question, it has to be taken into account what the actual goal of the crack-detection system is. If the purpose of the system is to generate a model that represents the crack distribution for a given sensor as accurately as possible, limited by factors such as sensor and actuator accuracy as well as signal theoretical and algorithmic limitations, then connecting nearby crack ends would be a falsification of data and thus inappropriate. But if the goal is to generate a crack representation that already has certain repair aspects in mind, connecting cracks might actually be very useful.

A subsequent repair process might, for instance, feature an actuator-guided milling cutter, used to follow the crack center lines generated by the novel crack-inspection method presented in this dissertation. A joint paper discussing such a repair chain, featuring a white light interferometer, was published in [Sch+16], and the author of this dissertation contributed the sensor part. Connecting cracks might be useful in this scenario, because in the worst case, a small extra piece of material between two already weakened parts of the material would get repaired even though it might not have been necessary. In the best case, the crack really did continue in between the two cracks and was thus combined correctly; and worst case, a small additional area of the part is repaired. Even in the worst case, though, connecting the cracks might simplify the actuator path planning for a subsequent repair process because there are fewer interruptions. In order to allow greater flexibility with regard to how the novel method is used, it was decided to implement the crack-connection procedure as an optional step in the inspection process chain.

The algorithm itself is again based on detecting the end vector of crack segments (see Figure 5.30) and calculating an area aligned with that vector (see Figure 5.31 on the left). Next, other crack segments are checked if they occur within that area. If so, the shortest distance between the end point and the other crack segment is calculated, and the cracks are combined.

## 5.7 Crack Verification

The image-processing steps detailed in the previous section deliver crack contours and center lines. The output can be transformed back into three-dimensional space by using the output as indices for the original 2.5D scan matrix and then applying the inverse of the homogenous transform that was used to transform the 3D scan into 2.5D space. Since it would be unacceptable to miss positive findings, the algorithm is set to produce false positives rather than to miss cracks. Because a large number of incorrect findings at the end of the process would be inappropriate, a verification step to analyze the crack results is required. This is done with two different techniques. First, the crack morphology and

the crack's relationship to other cracks and features is considered in the image domain. Second, the crack's section contour is checked against a model at several parts of the crack in the 2.5D domain. The two results are combined to determine whether a finding is a crack or not. Since the bottom of the crack is not necessarily visible, it is not reasonable to build a full three-dimensional model of the crack for verifications purposes.

As with the detection of cracks, a big challenge is that there is no formal definition of what is considered to be a crack in the CFM-56 combustion chamber outer liner's repair manual. It only states that findings under a minimum length  $l_r = 0.76$  mm do not have to be repaired. The manufacturer's definition of a crack is thus essentially that a finding that is detected by the established methods (FPI and visual inspection), is considered to be a crack by the repair technician, and has a length  $l \geq l_r$ , is to be treated as a crack finding and, therefore, should be repaired. Unfortunately, such a definition cannot be put directly into algorithmic form. Hence, the aforementioned two crack-verification methods were developed and manually tuned to yield results similar to those achieved by repair technicians. In order to allow a more direct use of the repair technicians' established knowledge, the software developed for this dissertation also allows a manual selection and deselection of damages.

### 5.7.1 Crack Verification in the Image Domain

In the image domain, several attributes can be used to decide whether or not a finding is an actual crack. These attributes can then be used to build up a multi-dimensional verification attribute space, which can be used to verify findings.

One attribute that is immediately obvious is the length of cracks, especially since an actual number is given for the minimum length of a crack. What needs to be defined is how exactly the length of a crack is determined, given that most cracks split up into crack segments. Since it is the overall length that contributes to a weakening of the airplane liner structure, it is a meaningful choice to consider the sum of the individual crack segments.

The second attribute is based on the fact that cracks often either start or terminate in features such as holes or edges. This is because features are structurally weaker and thus serve as crack seeds. Additionally, they are often natural barriers beyond which crack growth cannot continue. Therefore, there is an increased likelihood of a finding being a true crack if a damage is in the vicinity of a feature. Another factor is that it is unlikely that a crack is parallel or tangential to a feature. The reason for a crack to form has to be an effective force that is putting stress on the material in such a way that the material fails. According to Richard and Sander, there is often a tendency for the crack to be oriented approximately orthogonally to the damaging force, after a certain degree of stability in growth has been reached [RS09, p. 26] (see Figure 5.34 on the following page). It was found that this orthogonal behavior can typically be observed with the effective resolution of the novel method. It is therefore reasonable to consider the deviations from an orthogonal orientation of damages and features, not just their distances. As with the crack length consideration, it is appropriate to consider individual crack segments rather than the entire crack at once. It should be mentioned that the microscopic behavior

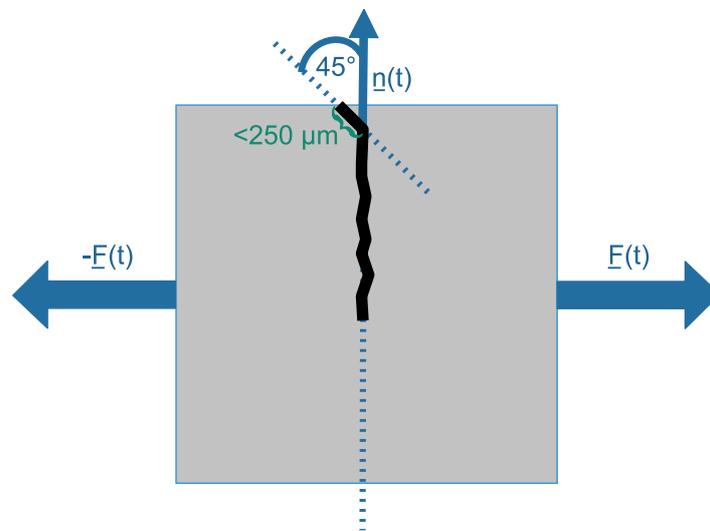


Figure 5.34: Illustration of how a crack typically forms  $45^\circ$  relative to the normal force in the beginning and then continues orthogonally to the damaging force [RS09, p. 26].

in the vicinity of weakened areas depends on the specifics but tends to be about  $45^\circ$  relative to the normal force [RS09, p. 26], see Figure 5.34.

An additional possible criterion that was considered in early stages of the research project is the variability of cracks (proposed in [Ott+14]) because a true crack seldom follows a very straight path. There is an increased likelihood that a completely straight indication is either a surface manipulation sustained from grinding in a previous repair step or is a sensor/signal processing artifact. It should be noted that such an observation might be detected as a crack segment that is part of a true crack, because, for instance, a grinding damage might intersect with a crack. Therefore, the variability consideration would also have to be performed based on crack segments. It was found, however, that variability is not a very stable criterion in this particular case, especially for shorter crack segments, and it was thus removed from the crack verification procedure. Similarly, the crack width was removed from the initially proposed collection of criteria [ibidem].

Therefore, there are three remaining crack criteria in the image domain:

- Summed crack segment lengths
- Distance to features
- Orientation to features

Since all of these criteria are based on crack segments, the entire crack verification procedure in the image domain is performed on crack segments rather than the full crack.

It was originally proposed to simply calculate and scale these metrics and calculate the 2-norm of the resulting crack verification vector [ibidem]. However, certain aspects, such as minimum lengths and interdependencies, cannot be calculated in such a fashion. Therefore, the procedure was modified

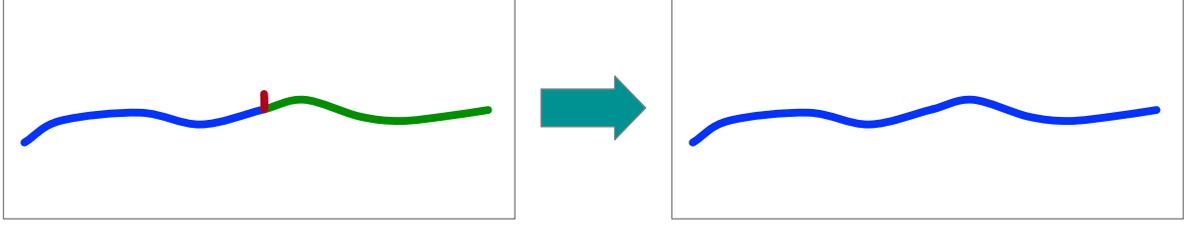


Figure 5.35: Illustration of the effect of removing small segments.

as follows. Very short segments can cause potential problems in the orientation calculation. Since they contribute little to either the overall crack length or the distance to features, they are removed in a preliminary step. The remaining crack segments are then merged where appropriate, as illustrated in Figure 5.35. This leads to increased verification speeds and more stable results because segments that might have otherwise been too short may end up being combined with other crack segments.

Next, the remaining segments are analyzed individually. First, the feature-related criteria are evaluated. It is unknown whether the damage is close to any features, and if so, which points are closest. Therefore, the point space is partitioned using a k-d tree in order to allow for fast neighborhood searches. Per feature, the crack/feature point pair with the smallest distance is considered to be the distance to that feature. If the distance is smaller than a defined threshold  $t_{\text{Distance}}$ , the crack segment is considered to be near that particular feature and a score is calculated, as follows:

$$s_{\text{Distance}} = \frac{t_{\text{Distance}} - \Delta_{\text{Feature}}}{t_{\text{Distance}}} \quad (5.79)$$

Afterward, the crack segment's orientation to the feature is calculated. For this calculation, a line regression is performed using the feature point closest to the line and a configurable amount of its neighbors (line  $l_{\text{Feature}}$  in Figure 5.36). Similarly, a line regression is performed with the closest crack segment point and its neighbors (line  $l_{\text{Crack}}$  in Figure 5.36). Then the angle between the two lines can be calculated by solving the inner product (see for instance [Zei+12, p. 361]) of the two lines' direction vectors for their angle:

$$\underline{a} \cdot \underline{b} = |\underline{a}| \cdot |\underline{b}| \cdot \cos(\varphi) \Leftrightarrow \varphi = \arccos \frac{\underline{a} \cdot \underline{b}}{|\underline{a}| \cdot |\underline{b}|} \quad (5.80)$$

If the resulting angle is bigger than  $90^\circ$ , it is changed as follows:

$$\tilde{\varphi} = \begin{cases} \varphi & \varphi \leq \pi/2 \\ \pi - \varphi & \text{else} \end{cases} \quad (5.81)$$

The angle  $\tilde{\varphi}$  is then used to scale the distance score (5.79):

$$s_{\text{Distance,Angle}} = s_{\text{Distance}} \cdot \left( -\frac{\frac{\pi}{2} - \tilde{\varphi} - \frac{\pi}{4}}{\frac{\pi}{4}} \right) = -s_{\text{Distance}} \cdot \left( 1 - \tilde{\varphi} \cdot \frac{4}{\pi} \right) \quad (5.82)$$

This means that if the crack segment is oriented orthogonally to the feature, the score is not modified.

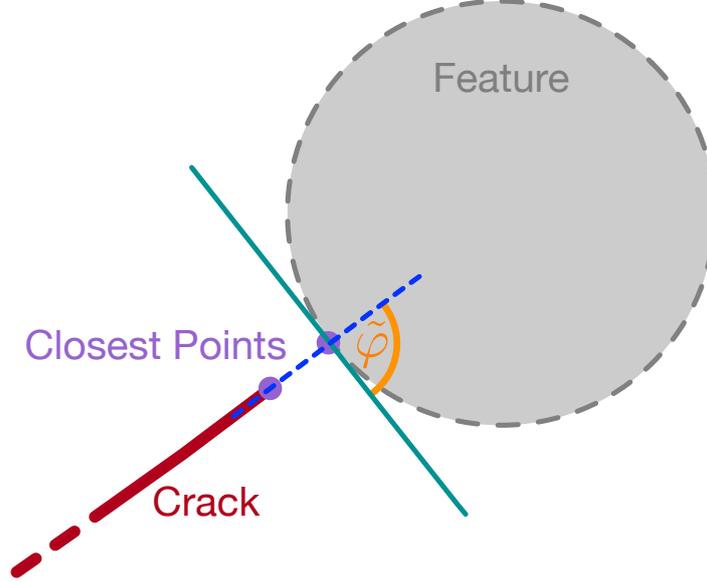


Figure 5.36: Illustration of the feature-orientation calculation.

For angles smaller than  $90^\circ$ , it approaches a score of 0 as it reaches  $45^\circ$  and becomes negative as the angles becomes smaller than  $45^\circ$ , approaching a value of  $-1$  as it goes to  $0^\circ$ .

If a crack is close to more than one feature, the same procedure is performed for all features and the individual scores are summed up to yield the overall feature score. It is somewhat unlikely to find a crack that is not close to any features and consists of only a single segment. Such a single-segment crack gets a feature score of  $-0.5$ . Crack segments that are not single-segment cracks but are not close to any features get a neutral score of 0.

The final verification score is calculated by considering the length, normalized with an average norm length:

$$s_{\text{Overall}} = (s_{\text{Distance,Angle}} + 1) \cdot \frac{\text{length}_{\text{Segment}}}{\text{length}_{\text{Norm}}} \quad (5.83)$$

The length itself is calculated as the sum of Euclidean distances between the points:

$$\text{length}_{\text{Segment}} = \sum_{i=2}^N \|\mathbf{p}_i - \mathbf{p}_{i-1}\|_2 \quad (5.84)$$

The distance score approaches 1 for very small distances between a feature and the crack segment point, and 0 for the distance approaching the near threshold. Thus, the distance score is in the interval  $[0; 1]$ . The angular score is a factor in the interval  $[-1; 1]$  and is used to scale the distance score. Hence, the combined distance/angle score is also in the interval  $[-1; 1]$ . Since a factor of 1 is added to the combined distance/angle score and the crack segment length can be arbitrarily large, the overall score

is in the interval  $[0; \infty)$ . For instance, a crack segment that is not part of a single-segment crack, is not near any features ( $s_{\text{Distance,Angle}} = 0$ ), and has a segment length equal to the norm length, has a resulting score of precisely 1. For a crack segment of norm length, intersecting orthogonally with a feature, the resulting score is precisely 2. A crack segment is considered to be part of a true crack indication for a score greater than or equal to 1.

## 5.7.2 Crack-Sectioning Verification

Whereas the method introduced in the previous subsection generates a score, the method detailed in this subsection is used as a secondary mechanism, to confirm the previous positive findings. In other words, it yields a binary decision rather than a verification score. Thus, if the previous algorithm decided that a crack segment is a false positive, that particular crack is not reprocessed by the secondary mechanism. If the previous algorithm does consider an indication to be a true crack segment, its score is either confirmed, meaning the score is left unchanged, or set to zero.

The idea of the algorithm is to create several orthogonal sections of the crack and analyze whether a given percentage of sections follow a typical crack profile. Since all cracks are unique, this is not done with a matching approach but rather by considering whether the section scans' geometries fit an abstract model. As with the previous method, the crack-sectioning approach is performed individually



Figure 5.37: Illustration of crack points to line mapping.

for each crack segment. In order to create the crack sections, a two-dimensional line regression is performed using the crack points'  $x$  and  $y$  values. Next, the original points are mapped onto the line and the minimum and maximum 2-norm mapped points are determined (blue circles in Figure 5.37). The crack sections are to be created within the line segments defined by the minimum and maximum points.

Afterward, the regression line has to be split into sections. The user cannot know beforehand how long the different segments will be. Since only one set of settings should cover most cases, it does not make sense to offer an option to define how many sections the crack segments should be divided into. It is more reasonable to define a distance  $\Delta d$  between the different sections. However, in certain cases, that can lead to a very large number of segments; hence, there should also be an upper bound for the amount  $n_{\text{Sections,max}}$ . Finally, every crack segment should yield at least one section, meaning  $n_{\text{Sections,min}} = 1$ . Therefore, as the first step, the initial number of resulting sections is calculated from the line segment length ( $\text{length}_{\text{line}} = \|p_{\text{max}} - p_{\text{min}}\|$ ):

$$n_{\text{Sections}} = \text{length}_{\text{line}} / \Delta d \quad (5.85)$$

With the given constraints, the actual number of resulting sections can be defined as:

$$\tilde{n}_{\text{Sections}} = \begin{cases} 1 & n_{\text{Sections}} < 1 \\ n_{\text{Sections,max}} & n_{\text{Sections}} > n_{\text{Sections,max}} \\ n_{\text{Sections}} & \text{else} \end{cases} \quad (5.86)$$

In case the initially calculated number of sections exceeds the maximum number of sections, the section distance has to be adjusted as well. Therefore, it can be said in general:

$$\Delta\tilde{d} = \text{length}_{\text{line}} / \tilde{n}_{\text{Sections}} \quad (5.87)$$

With these considerations, the locations of the regions of interest are defined, meaning that rotated

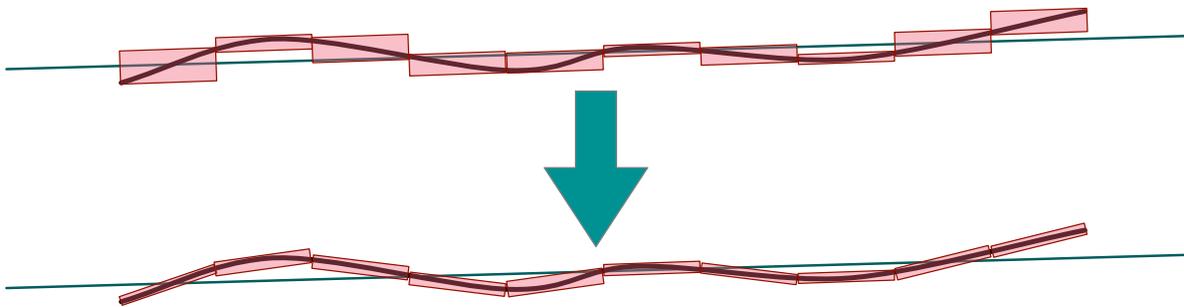


Figure 5.38: Illustration of the ROI creation.

cuboids can be placed centered around those points. Furthermore, the orientation of a cuboid is defined by the line:  $\underline{x}$  in the direction of the line,  $\underline{y}$  in the direction of the normal and  $\underline{z} = \underline{x} \times \underline{y}$ . However, the extent of the cuboid is defined only in the  $x$  direction (by the distance  $\Delta\tilde{d}$ ) and in the  $z$  direction — such that it contains the full height of the scan. The extent in the  $y$  direction still needs to be determined. In order to do so, only the  $x$  and  $y$  coordinates of the crack segment region need to be considered. All points within the  $x$  bounds of the current region of interest are analyzed to find the minimum and maximum  $y$  values  $y_{\min}$  and  $y_{\max}$ . The extent of the region of interest (ROI) in the  $y$  direction is  $\Delta y = y_{\max} - y_{\min}$ . With the full extent, the location and the orientation of the rotated cuboid, the ROI is defined. This is illustrated in Figure 5.38.

Next, the section scans are created from the contents of the ROI. In order to do so, all cloud points are checked whether they are inside the different ROIs. The resulting sub point clouds are  $P_i \subseteq P_{\text{Scan}}$ , without any particular order. The idea is to combine the points along the mean  $x$  direction of the rotated ROIs so that outliers from individual parts of the crack do not have a significant impact. As an intermediate step, an equidistant scan (in  $x$  and  $y$ ), oriented with the rotated ROI cuboid, is created. This requires a bin size being defined, typically the lateral  $y$  resolution of the sensor ( $\text{Resolution}_y$ ).

With that, the number of bins follows from the following relationship:

$$\text{Steps} = \left\lceil \frac{\text{Extent}_{\text{ROI},y}}{\text{Resolution}_y} \right\rceil \quad (5.88)$$

Using the step count, an array of index lists can be instantiated. Afterward, the association of each point with a bin can be calculated as follows:

$$\text{Index}_{\text{Bin}} = \frac{y_i + \text{Extent}_{\text{ROI},y}/2}{\text{Resolution}_y} \quad (5.89)$$

where  $y_i$  are the  $y$  coordinates of the scan points within the ROI. The indices  $i$  of the points are added to the list located at the calculated index.

With the bin lookup table (LUT) created in the previous step, it is then possible to create an equidistant scan by iterating through the LUT. The  $y$  values are calculated as:

$$y_i = -\text{Extent}_{\text{ROI},y}/2 + \text{Resolution}_y \cdot (i + 0.5) \quad (5.90)$$

The  $z$  and intensity values for the different bin positions can be calculated by combining the information available in the different bin lists; for instance, by calculating their respective means. Since the section scans do not have any extension in the  $x$  direction, all  $x$  values are zero.

In short, the section-scan generation yields an equidistant scan in the  $y$  direction, with no extension in  $x$  and the average of the  $z$  values along the  $y$  direction of the ROI.

The analysis of the section scans is based on evaluating how the scans change along the  $y$  direction. However, despite the averaging process involved in the section-scan generation, the scans can still be quite noisy. An elegant solution for dealing with this problem is the use of smoothing splines, see for instance [Pol93] and [HOO12] (co-authored by the author of this dissertation). Smoothing splines have a parameter  $\lambda$  that allows the function regression to be configured freely between a line and a regular cubic spline. For a suitable parameter  $\lambda$ , a smoothing spline regression delivers a function that is smooth yet follows the points to a certain degree. Additionally, the function is three times differentiable. The application of smoothing splines for crack analysis is illustrated in Figure 5.39 on the following page.

The modeled behavior of a crack section is the following: a crack is constant at the beginning, then can increase by a small amount and then decreases rapidly. At some point, the received signal will be invalid because not enough light returned to the sensor. It remains such for part of the width of the crack. Afterward, the expected signal behavior is to increase, potentially with a slight overshoot, and thereafter to reach a constant section again. It would be possible to analyze this behavior directly in the function itself by calculating a normal vector and monitoring its angular changes as the function progresses in the  $y$  direction. A better approach would be to consider the function's derivatives, especially since those are readily available from the smoothing spline function.

Some part of the crack area, as opposed to the regular surface, can be expected to be the global

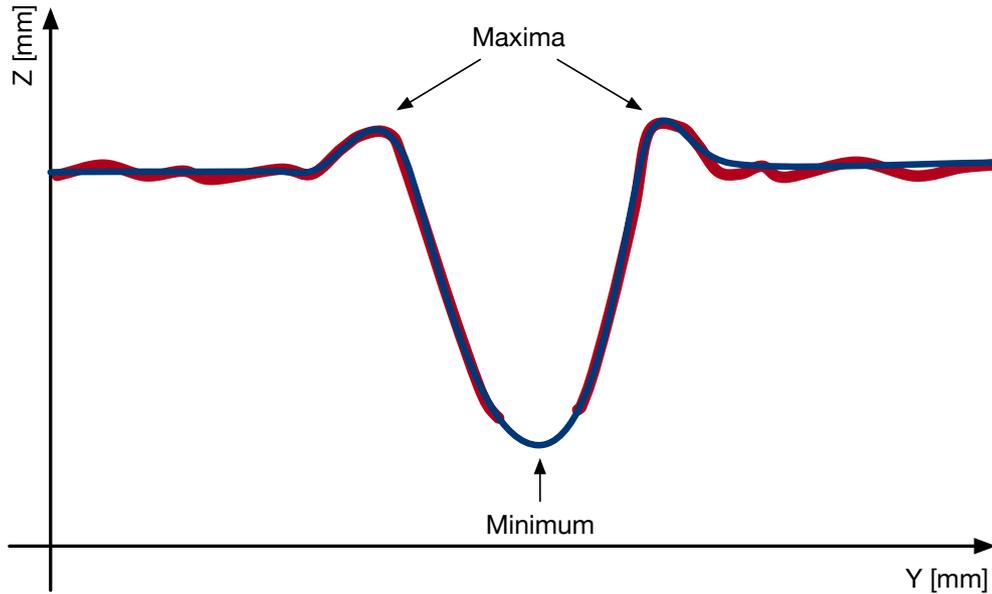


Figure 5.39: Crack-sectioning verification. The red line shows the section scan; the blue line shows the smoothing spline approximation.

minimum of the function. Furthermore, the edges of the crack should be local maxima. Thus, in order to find potential extrema, the spline's first derivative is analyzed for zero crossings because the first derivative's being zero is a necessary prerequisite for an extremum. All zero crossing locations are added to a list of extrema candidates. Next, the spline's second derivative is considered at the zero crossing locations in order to find maxima (second derivative negative) and minima (second derivative positive). The results are added to a list of extrema as  $\langle \text{index}, \text{bool} \rangle$  tuples, where the second tuple entry indicates if the entry is a maximum (false: minimum).

The list can be analyzed for a maximum-minimum-maximum pattern, where the extreme points are an appropriate distance apart. If such a pattern is found, the second derivative's first zero crossing to the left of the first maximum is searched, marking the left index. Similarly, the second derivative's first zero crossing to the right of the second maximum marks the right index. These indices are used to determine the baseline  $z$  height at the position of the global minimum:

$$z_{\text{Baseline}} = \frac{z_R - z_L}{y_R - y_L} \cdot (y_{\text{min}} - y_L) + z_L \quad (5.91)$$

The minimum  $z$  value  $z_{\text{min}}$  is then compared to the baseline value. If the difference exceeds a certain threshold, the finding is considered to be a positive.

Even for a true crack, not all crack sections might indicate a positive result; therefore, the crack-section analysis is performed in several locations of the crack segment. If the percentage of positive crack section results exceeds a predefined percentage, the overall crack segment is confirmed to be a true crack.

## Chapter 6

# Verification of the Method

In this chapter, the novel method introduced in this dissertation is verified. This is done in two parts. First, in order to guarantee that the system is functioning properly, a daily self-check was developed. This self-check has been executed many times over a long period of time, proving the system's long-term stability, see subsection 6.1. Second, the use of the novel approach resulted in a statistically significant amount of data. The data was analyzed by Lufthansa Technik AG in an internal probability of detection (POD) study, the result of which will be discussed in subsection 6.2.2.

Combined, these two aspects verify the method's feasibility. They show that the sensor and algorithms work as expected, both by example (daily self-check) and by providing a statistical analysis (POD study).

### 6.1 Daily Self-Check

The daily self-check performs two tests. First, the white light interferometer itself is verified. More specifically, the pixel field is analyzed for dead pixels and the sensor's depth data is checked for plausibility. Second, a test inspection of a known part is performed in a specific area in order to confirm that the findings agree with the expected results.

#### 6.1.1 White Light Interferometer Check

As mentioned, the WLI check includes a dead pixel check and a depth data accuracy analysis. The tests should be performed as fast as possible, meaning that a method that covers both aspects is ideal. Hence, an approach based on scanning two spheres of different well-known radii, from different positions and orientations, was developed.

In the beginning of the process, a Boolean-typed matrix with the same dimensions as the pixel array is generated and each entry is initialized as false. Every time a pixel delivers a valid intensity level in one of the scans, the according entry in the pixel field is set to true. The different positions should be programmed in such a way that every pixel has the chance to scan a valid surface.

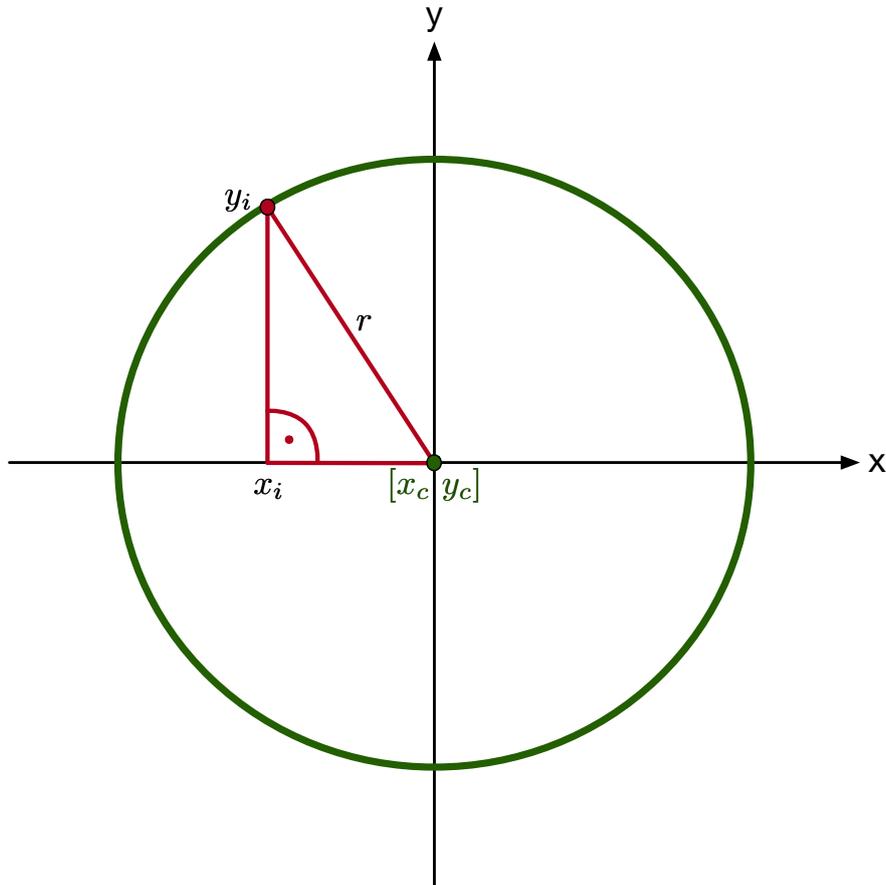


Figure 6.1: Illustration of a surface point resulting in a right triangle with the center point.

The radius of the sphere is estimated in all of these positions. A sphere can be described mathematically as follows:

$$r^2 = (x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 \quad (6.1)$$

with  $r > 0$ . This equation can be derived directly from the Pythagorean theorem, because every point on the sphere results in a right triangle. This is illustrated for a circle in Figure 6.1.

It is somewhat problematic that estimation errors in the sphere center ( $\hat{x}_c, \hat{y}_c, \hat{z}_c$ ) can be masked by errors in the radius estimation ( $\hat{r}$ ) and vice versa. Because of this, two scans are performed in each position; the first one is used to estimate the center point with fixed known radius, and the second one estimates the radius with fixed center point.

In general, every scan yields a set of points  $p_i \in P$ , all of which are either invalid or represent the true surface location, relative to the sensor at that particular spot of the pixel array, plus an unknown error. Most of the invalid points are removed from the scan using the thresholding approach introduced in 4.3.2. However, some outliers might remain, which can be remedied using a RANSAC sphere detector. For each point  $p_i$  with the coordinates  $x_i, y_i$ , and  $z_i$ , the radius  $r_i$  can be calculated with the

estimation parameters  $\hat{x}_c$ ,  $\hat{y}_c$ , and  $\hat{z}_c$ :

$$r_i = \sqrt{(x_i - \hat{x}_c)^2 + (y_i - \hat{y}_c)^2 + (z_i - \hat{z}_c)^2} \quad (6.2)$$

The squared error between the estimation parameter  $\hat{r}$  and the radius determined for the point  $p_i$  is:

$$e^2 = (r_i - \hat{r})^2 = \left( \sqrt{(x_i - \hat{x}_c)^2 + (y_i - \hat{y}_c)^2 + (z_i - \hat{z}_c)^2} - \hat{r} \right)^2 \quad (6.3)$$

Thus, the minimization objective is:

$$\arg \min_{\hat{x}, \hat{y}, \hat{z}, \hat{r}} \sum_i \left( \sqrt{(x_i - \hat{x}_c)^2 + (y_i - \hat{y}_c)^2 + (z_i - \hat{z}_c)^2} - \hat{r} \right)^2 \quad (6.4)$$

If a parameter is to be fixed, the objective function simply has to be modified to keep that parameter constant. For instance, for a sphere center estimation with fixed radius, the equation becomes:

$$\arg \min_{\hat{x}, \hat{y}, \hat{z}} \sum_i \left( \sqrt{(x_i - \hat{x}_c)^2 + (y_i - \hat{y}_c)^2 + (z_i - \hat{z}_c)^2} - r \right)^2 \quad (6.5)$$

If the radius has to be estimated for a fixed center point, the equation needs to be modified as:

$$\arg \min_{\hat{r}} \sum_i \left( \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2} - \hat{r} \right)^2 \quad (6.6)$$

Since the radius of the sphere is known with a relatively high degree of precision ( $\pm 0.01$  mm), the estimation of the center-point location (equation 6.5) should also be highly accurate, provided the sensor operates as expected. Using the center-point location estimated in the first step as fixed parameters for the sphere estimation in the second scan should therefore also deliver the expected radius with very high accuracy, provided that the sensor works as expected. The procedure is repeated several times for each sphere. Both the radius and the relative sum of squared errors are stored for each radius estimation step.

If either the percentage of dead pixels exceeds a certain threshold, the estimated radii differ too much from the inspected ones, or the sums of squared errors are too large, the self-check is not successful.

## 6.1.2 Test Inspection

If the sensor passed the previous check, a test inspection is performed in order to verify that the entire system is operating as expected. The basic idea is to perform all process steps, including tool calibrations, a laser line section scan, white light interferometer scanning, and processing of the captured data, as described in this dissertation. However, instead of performing a complete interferometer digitalization, only a relatively small section is scanned with the interferometer. The findings generated by this

reduced process are then compared to those produced by a master scan. Passing the test inspection step proves that the combination of the individual components yields the expected results.

### Analysis of the Individual Steps

Next, the verification of the novel method's individual steps as performed in the daily self-check routine are detailed here. In accordance with the main steps, these aspects include:

- Combination of several scans into larger structures: Scan unification.
- Converting the data into images: Image generation.
- Signal processing: Crack detection.
- Plausibility check: Crack verification.

After the reduced white light interferometer scanning process (see chapter 4) has been completed, the full scan collection is split into patch collections (see section 5.1). A typical patch collection is

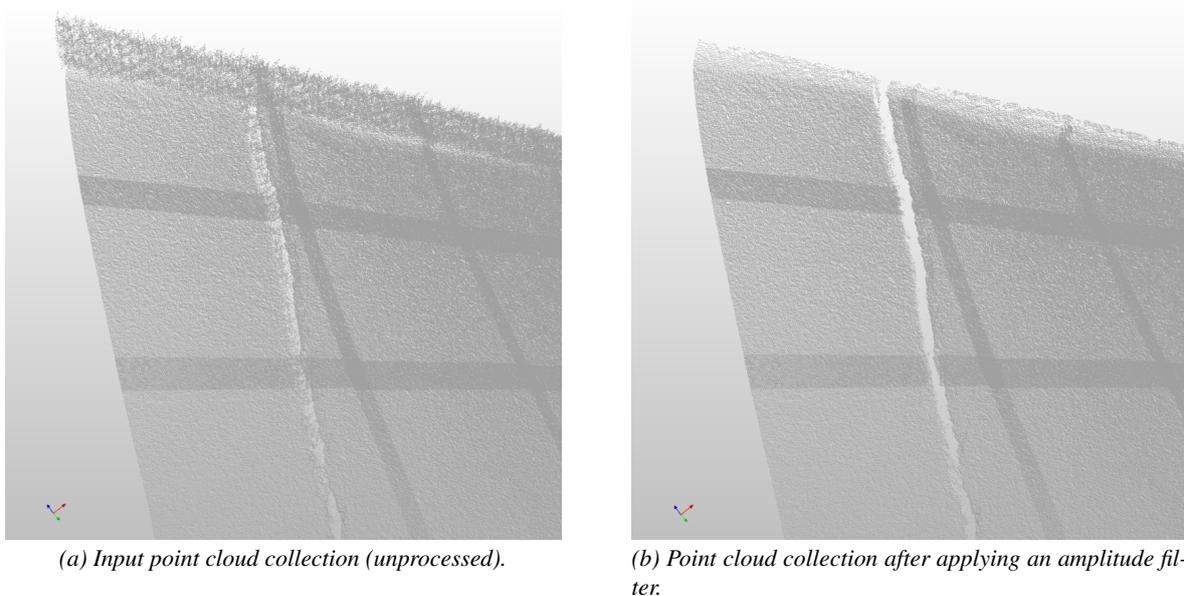


Figure 6.2: Illustration of the amplitude filter applied to the input point cloud.

shown in Figure 6.2a. As expected, the scans contain both valid and invalid data, meaning that an appropriate amplitude filter needs to be applied to each scan in the collection (see subsection 4.3.2). Applied to the input collection in Figure 6.2a, such an amplitude thresholding filter results in a filtered collection, as shown in Figure 6.2b.

It was stated in section 5.2 that a registration algorithm has to be applied to the filtered patch collection. Figure 6.3a on the following page illustrates why this is the case; here, the unification algorithm (see section 5.3) is applied directly to the filtered data set. The resulting point cloud features

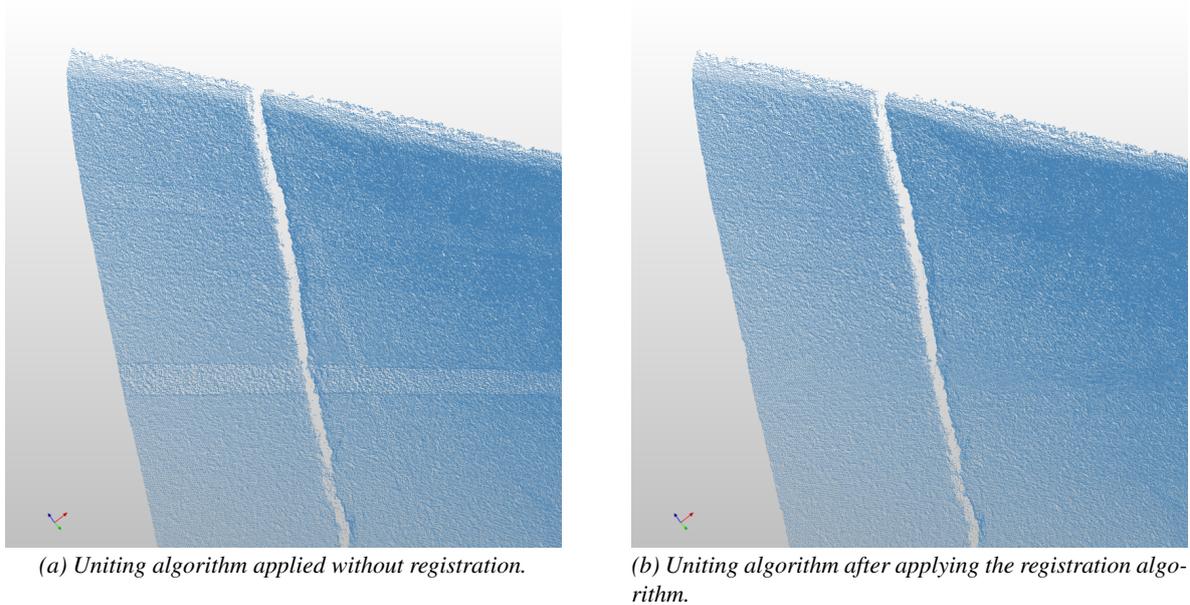


Figure 6.3: Illustration of the need for a registration algorithm.

massive artifacts, which would lead to an unacceptably large number of false positives. Applying a registration algorithm to the filtered input collection reduces the transform errors between the individual scans. Figure 6.3b shows the unified data set after the registration algorithm has been applied.

After the point cloud is transformed into 2D space and the background is removed, the point cloud is mapped into an image, in preparation for the crack-detection steps. Finally, the crack-detection and verification algorithms introduced in sections 5.6 and 5.7 are applied to the input image. The detection and verification results, without crack-sectioning verification, are visualized in Figure 6.4a on the following page. Comparing the actual cracks in the material shows that the crack in the lower left is a false positive, which failed to be deselected by the first verification algorithm. If the image-based verification is followed by the crack-sectioning verification introduced in subsection 5.7.2, the false positive is deselected, as shown in Figure 6.4b.

### Fixture Correction Transform

Whereas performing the reduced crack-detection process is essentially the same as the full process described in this dissertation, the crack comparison is an additional step. Since the part is locked into its fixture by hand, the repeatability of the part positioning is relatively low. Therefore, a transform allowing alignment of the current findings with those of the master scan needs to be determined. Since the accuracy requirements for the comparison are not exceedingly high, this transform can be found based on the laser line section scans.

Because of the way the fixture is designed, the main error occurs in the  $z$  rotation. Therefore, it is reasonable to try to determine the incorrect  $z$  rotation in a first step, using down-sampled point clouds.

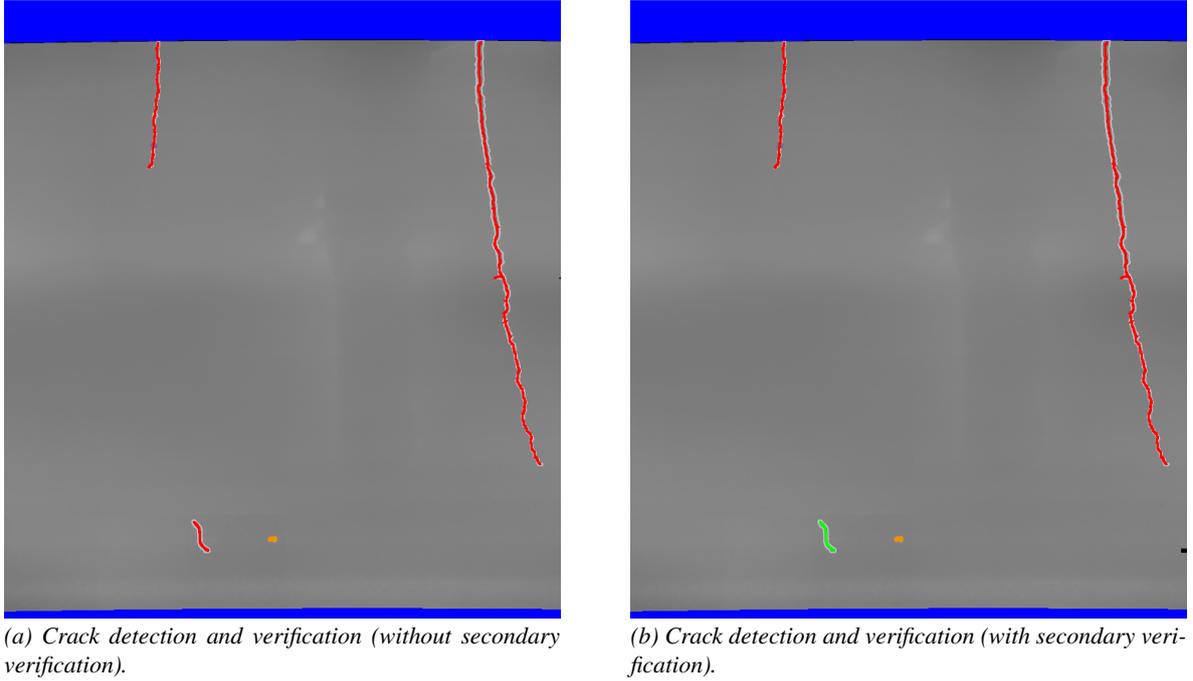


Figure 6.4: Illustration of the effect of secondary verification.

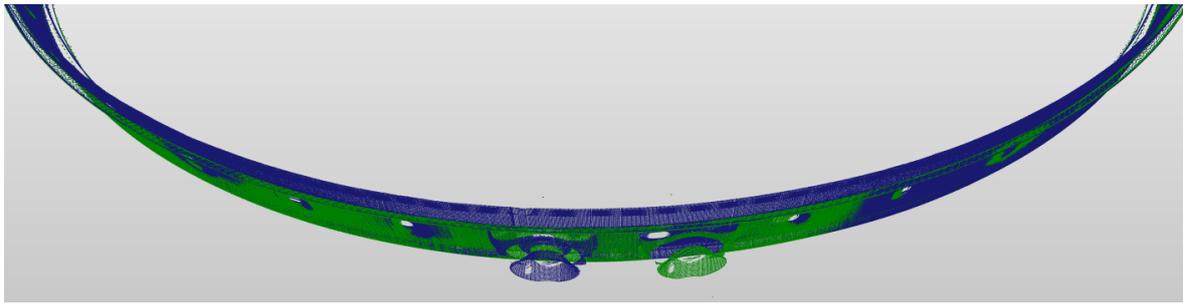
Afterward, a full registration using the original point clouds in all degrees of freedom is performed, resulting in the overall fixture correction transform. However, although the part has some unique symmetry-breaking features, it is largely similar along its  $z$  rotation. Therefore, there is a high chance of a standard point cloud registration algorithm — such as the ICP algorithm — converging locally. Because only a single degree of freedom is relevant, an approach similar to cross-correlation is used. The continuous time cross-correlation integral takes on its maximum value where the signals are most similar. For time-discrete signals, a similar expression can be found in terms of a sum. For  $T = 1$ , according to [OL10, p. 215]:

$$\varphi_{f,g}^E(m) = f(m) \star g(m) = \sum_{n=-\infty}^{\infty} f(n)^* \cdot g(n+m) \quad (6.7)$$

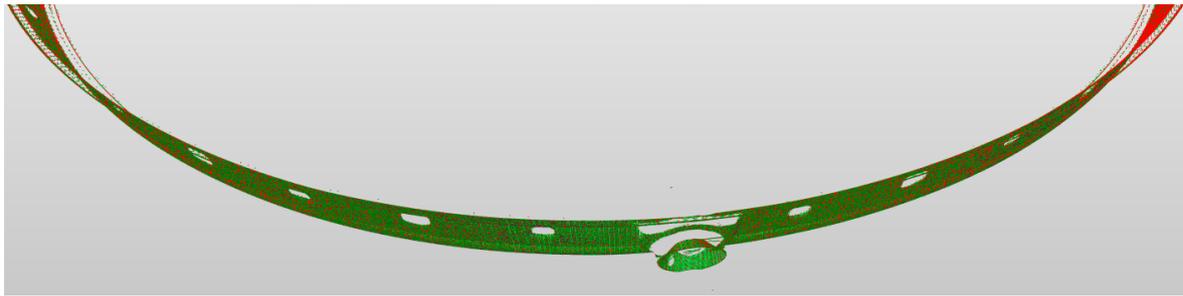
Something similar can be applied in the given case; however, an angle  $\theta_m$  is used as the offset instead of the index  $m$ . Effectively, a transform that only affects the  $z$  rotation by the current angle is applied to the point cloud, and then the convolution sum in equation is calculated for that state:

$$\varphi_{f,g}^E(\theta) = \sum_{n=-\infty}^{\infty} f(n)^* \cdot \left( \begin{smallmatrix} \theta \\ 0 \end{smallmatrix} \mathbf{T} \cdot g(n) \right) \quad (6.8)$$

This is repeated for the expected angular error range with the desired angular granularity. The result of this “angular correlation” is the transform  $\begin{smallmatrix} \theta, \text{opt} \\ 0 \end{smallmatrix} \mathbf{T}$ , which aligns the two scans with the lowest sum of squared errors. Applying this transform to the full input point cloud yields appropriate starting con-



(a) Before point cloud registration (green: template, blue: current scan).



(b) After point cloud registration (green: template, red: corrected scan).

Figure 6.5: Effect of the combined rotational convolution and ICP for point cloud registration.

ditions for a fine registration using a regular iterative closest points algorithm, resulting in a transform  ${}_{\theta, \text{opt}}^{\text{Reg}} \mathbf{T}$ . The overall transform is then:

$${}_{0}^{\text{Reg}} \mathbf{T} = {}_{0, \text{opt}}^{\theta, \text{opt}} \mathbf{T} \cdot {}_{\theta, \text{opt}}^{\text{Reg}} \mathbf{T} \quad (6.9)$$

The effect of applying the combined rotational convolution and ICP is illustrated in Figure 6.5.

### Crack Comparison

The crack comparison algorithm is executed per patch. Thus, if four patches are to be compared (see Figure 6.6 on the following page), the algorithm is executed four times. Figure 6.6 shows a typical scene that might be used for crack comparison. The image features four patches representing all relevant test scenarios. The lower right patch contains a hole feature but no cracks. The upper right patch does not contain any cracks, nor any special features, which was detected correctly. The upper left patch contains two correctly detected cracks. The lower left patch contains a crack and a welding joint from a previous repair. The crack was detected correctly, and the welding joint was detected incorrectly as a crack but was then automatically declassified. Therefore, the system works as expected.

First, the regular crack-detection and verification steps introduced in this dissertation are executed on the current data set. Next, the fixture-correction transform is determined and applied to the detected cracks. The actual crack comparison is then performed by creating an empty (black) image

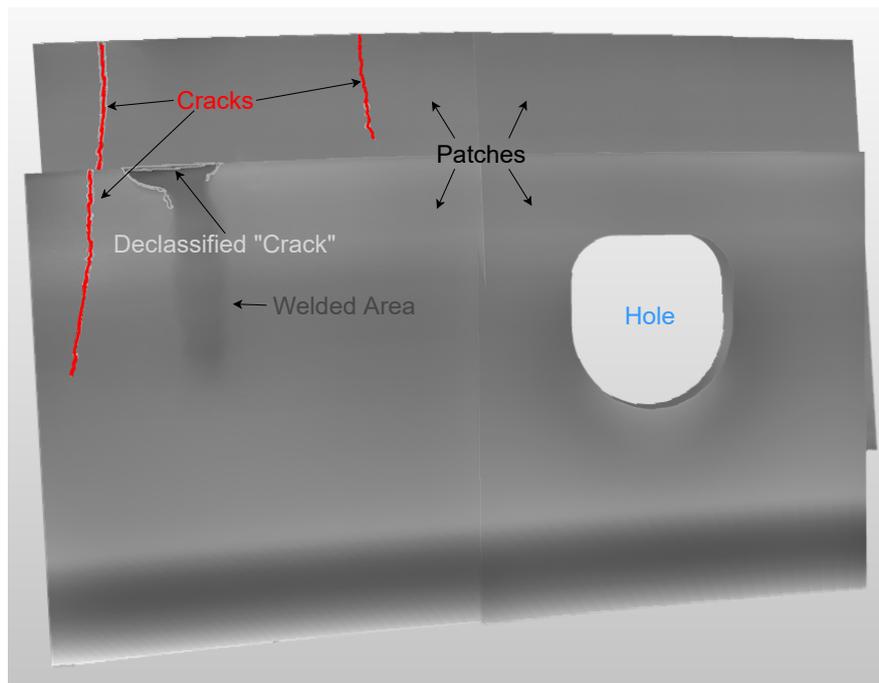


Figure 6.6: Template scan result. The four patches feature different test scenarios: no damages, hole feature, cracks, and a welding joint. All cracks were detected, and no false positives were generated, due to the welding joint having been declassified.

and drawing the (white) template cracks into it. Then the number of white pixels  $n_{\text{Template}}$  is counted. Afterward, a second black image is created and the newly detected cracks are drawn into it in white, and the image is dilated by a pre-defined amount, which increases the crack extent. Then, the second image is subtracted from the first and negative results are set to zero. This means that ideally, all crack indications from the template scan will be removed by the current test scan, thus being set to zero. The remaining number of white pixels  $n_{\text{Result}}$  is counted and compared to  $n_{\text{Template}}$ . If the template scan did not contain any crack indications, that particular sub-test is automatically passed. Otherwise, the test is passed if the percentage  $n_{\text{Result}}/n_{\text{Template}} \cdot 100\%$  is smaller than a pre-defined threshold.

## 6.2 Probability of Detection

It is impossible to guarantee a 100% detection rate for all defect sizes. Indeed, for a wide class of systems, the following assumptions can be made:

1. As the defect size approaches zero, defects are no longer detected.
2. As the defect size becomes very large, all defects are detected.

These assumptions also hold for the crack-detection system proposed in this dissertation, provided that the crack is wide enough to be captured by the WLI with the selected optical magnification.

Therefore, the lower end is defined by the lateral resolution of the WLI and the thresholds for the crack detection algorithms. As a certain crack length and width is reached, the system should be able to find all cracks reliably.

The safety-critical aspect is the largest crack size that can be missed by the system rather than the smallest crack size the system can possibly detect. Since this question cannot be answered with 100% certainty for all possible cracks, a statistical measure has to be used. Under the given circumstances, a probability of detection (POD) calculation based on hit/miss data is used in the MRO industry. A hit is defined as a crack that was found by the inspection method, whereas a miss is defined as a crack that was not detected.

### 6.2.1 Explanation of Probability of Detection

Given what was said in the beginning of this section, an ideal POD response would not be able to detect any defects up to a certain size and then all defects after that threshold [Mil09, p. 6]. In reality, inspection systems likely have a gray zone in which cracks are both missed and detected.

The only metric used by the POD analysis, in this case, is whether the crack was detected or not. The response is thus binary in nature. One option of dealing with this problem would be to generate a histogram with binned crack lengths. In the literature, the crack length is commonly referred to as “ $a$ ” [ibidem, p. 15][AST12, p. 1]. If a crack of length  $a$  is present, the probability that the crack is detected is  $\text{POD}(a)$ . The probability of detection  $\text{POD}(a)$  is then the number of hits divided by the number of defects  $N_a$ , which equals the number of hits plus the number of misses, for a given crack length  $a$ :  $\text{POD}(a) = n_{a,\text{hit}}/N_a = n_{a,\text{hit}}/(n_{a,\text{hit}} + n_{a,\text{miss}})$  [Mil09, p. 19]. The disadvantage of this approach is that the resolution tends to be poor with respect to  $a$ , because to obtain a meaningful  $\text{POD}(a)$  calculation, a certain number of defects must be considered. The only way to increase the size resolution — which means decreasing the bin extent — would be to collect a lot more samples [ibidem, page 120].

An alternative to the histogram approach is to choose an appropriate mathematical function, which is then fitted to the data in order to approximate the true POD function [ibidem]. However, the data itself is binary and therefore discontinuous as well as bounded within the interval  $[0; 1]$ . It is, therefore, impossible to fit the data to a linear model. Instead, a generalized linear model approach is used in order to perform the regression. What this means is that instead of considering the responses directly, they are linked to the outcome probability [ibidem]. The probability is also bounded by definition, but unlike the binary response it is continuous. According to the ASTM E2862-12 norm, the generalized linear model can either be of type  $g(y) = b_0 + a \cdot b_1$ , or  $g(y) = b_0 + \ln(a) \cdot b_1$ , where  $y$  is the hit/miss data,  $g(y)$  is the linking function,  $b_i$  are the function parameters, and  $a$  and  $\ln(a)$  are the continuous predictor variable or its log value [AST12, p. 2].

Several candidates for link functions  $g(y)$  are listed in the military handbook: logit (log-odds), probit (inverse normal-function), log-log, and complementary log-log functions [Mil09, p. 120]. Of those, the logit and probit functions are known to perform well in practice, according to the ASTM E2862-12 norm [AST12, p. 2]. The parameter estimation is then performed with a maximum-likelihood

approach [Mil09, p. 120]. In other words, the parameters that maximize the likelihood of the link function being observed are found. Since the link function relates to the observed hits and misses, the result is that the chosen link function is fitted to the observed data, as desired.

In other words, the method described in [ibidem] and [AST12] allows a continuous POD function to be fitted to the data in a maximum-likelihood sense. Therefore, for the currently observed experiment and for the chosen model, the calculated parameters have the greatest likelihood of producing the observed result (the hits and misses). Such a curve approximates the probability of detection for a given length for the current experiment by evaluating the generated curve at the length of interest. Conversely, it is possible to find which crack lengths in the current experiment imply which probabilities of detection. Since discontinuities are labeled with  $a$ , the maximum-likelihood crack length implied by a given POD  $p$  is written as  $a_p$ , for instance  $a_{90}$  for  $p = 90\%$ .

This basic approach holds specifically for the experimental data the curve was fitted to, but not in the general case. Indeed, general conclusions can only be drawn with a certain degree of confidence that the observed findings hold for other experiments as well. For a desired confidence level, upper and lower confidence bounds can be calculated. With the discontinuity length  $a$ , the POD  $p$ , and the confidence level  $c$ , the crack length expression is written as  $a_{p/c}$ . The literature typically uses a POD of 90% and a confidence level of 95%, resulting in the expression  $a_{90/95}$  [Mil09, p. 27] [AST12, p. 4].

Since the  $a_p$  value designates a maximum-likelihood level, about 50% of similar experiments can be expected to yield higher levels (and lower levels in the remaining cases) [Mil09, p. 16]. The  $a_{p/c}$  value on the other hand can be expected to be larger than the unknown true  $a_p$  value in  $c\%$  of similar experiments [ibidem, p. 16]. For the typical  $a_{90/95}$  value, this means that the determined value is an estimate that is larger than the true  $a_{90}$  value in 90% of similar experiments [ibidem, p. 16]. This implies two important considerations. First, the  $a_{90/95}$  is always larger than the maximum-likelihood  $a_{90}$  value. Second, even the  $a_{90/95}$  value allows for 5% of similar experiments to yield larger  $a_{90}$  estimates, which is a statistical risk [AST12, p. 4]. For binary response POD curves, the confidence bounds can be determined using the log-likelihood ratio criterion. In brief, the maximum-likelihood parameters are varied slightly, and the ratios of the logarithmic results and the logarithmic maximum values are analyzed in terms of their plausibility [Mil09, pp. 130].

## 6.2.2 Probability of Detection Analysis of the Novel Method

A hit/miss response POD analysis was performed in an internal study at Lufthansa Technik AG. It was conducted in agreement with the methodology summarized in the previous subsection, based on the ASTM E2862-12 norm [AST12] and the foundational work presented in the Military Handbook [Mil09]. The resulting POD curve is shown in Figure 6.7 with permission of Lufthansa Technik AG.

The curve model is of the logit (log-odds) type and shows the expected shape. The total number of observed events is relatively large (about 700) and the discontinuity lengths cover a wide spectrum. The resulting POD curve changes the most in the beginning because that is where the largest accu-

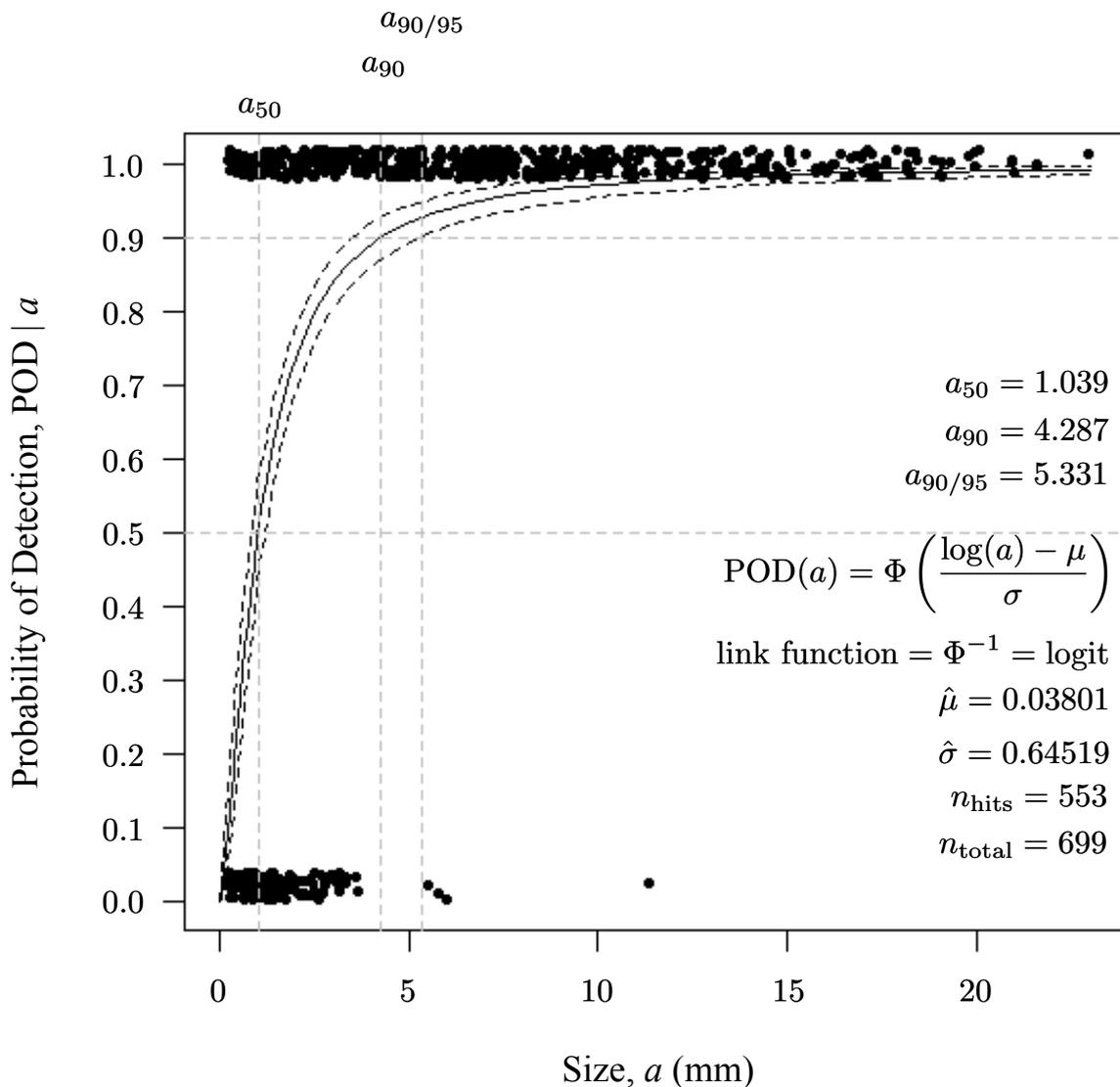


Figure 6.7: Hit/miss diagram of the novel crack-detection system [Source: adapted from an internal Lufthansa Technik study].

mulation of misses is; it then approaches the 100% POD level asymptotically for large discontinuity sizes. There are no further misses for very large discontinuities. The curve is therefore plausible.

The  $a_{90}$  value indicates that a probability of detection of 90% is reached at a length of 4.287 mm. The calculated upper and lower confidence bounds follow the shape of the maximum-likelihood curve closely and are therefore plausible as well. The confidence bounds are relatively tight, indicating that the parameters cannot be moved far from the maximum-likelihood parameters without the resulting log-likelihood ratios being changed significantly. The resulting  $a_{90/95}$  value is 5.331 mm. That means that an  $a_{90}$  estimate of 5.331 mm would be too large in 95% of similar experiments. Conversely, this means that there is only a 5% chance of not having a 90% probability of detection for cracks of that

length.

The maximum achievable performance for any given NDI system does not only depend on the system itself, but also on the parameters of the analyzed part, such as surface state and complexity of the shape. Therefore, a direct comparison of the  $a_{90/95}$  performance achieved by the novel method with values found in the literature for the state-of-the-art inspection method — fluorescent dye penetrant inspection (FPI) — is difficult. This is because the exact parameters of the study are likely different.

Furthermore, studies involving the human factor can be biased. For instance, the involved inspectors might be informed of the performance review, which could lead to an inaccurate representation of the mean capabilities of the process. Another possibility of skewing the outcome would be if only the most skillful inspectors participated in the study.

With that in mind, it is of little use to compare the novel method with any single study. A more meaningful approach would be to look at a combined estimate of the general capability of FPI. In 2011, the Australian government released a meta study reviewing the literature on probability of detection of FPI [HH11]. In that review, 12 major studies on the topic, published between 1968 and 2009, were identified and analyzed [ibidem, p. 1]. As expected, the results of the studies differ substantially, showing  $a_{90}$  values<sup>1</sup> ranging from 1.8 mm to 8.9 mm [LA03, p. 18], with a mean value<sup>2</sup> for  $a_{90}$  of 5.9 mm and a median value for  $a_{90}$  of 3.2 mm.

The authors of the meta study argue that each result from the different studies represents a snapshot of the capability of the FPI method. Therefore, they reason, it is justified to combine the studies in order to estimate values for the general performance of FPI [ibidem, p. 22]. By using the maximum-likelihood method to fit a log-normal distribution to the available  $a_{90}$  values, the best estimate for the median  $a_{90}$  value of all FPI systems was estimated to be 3.3 mm [ibidem, p. 23]. Furthermore, the authors of the meta study calculated 95% confidence intervals for their median  $a_{90}$  estimate using the log-likelihood ratio method; they found the upper 95% confidence value of the median  $a_{90}$  value of *all* FPI systems to be 4.3 mm (based on that data) [ibidem, p. 23]. Afterward, they inferred the maximum, which is to say worst case,  $a_{90}$  value that can be expected for an *individual* FPI system to be 6.2 mm, with a 95% upper confidence limit.

Comparing the results of Harding and Hugo's meta study to the performance of the novel method introduced in this dissertation shows that, even in this early stage, it is well within the expected performance level of the state-of-the-art method (FPI). The outlook given in the following chapter identifies measures that would allow to improve the performance of the method even further.

---

<sup>1</sup>Not all studies considered in the meta study provided  $a_{90/95}$  values.

<sup>2</sup>The meta study also provides a second mean value of 3.3 mm from a subset of the available studies.

## Chapter 7

# Summary, Conclusions and Outlook

In this chapter, a summary of this dissertation is given, and conclusions are drawn from the presented results. Thereafter, potential advancements of the novel method are presented as an outlook.

### 7.1 Summary and Conclusions

The established state-of-the-art inspection method for airplane combustion chamber liners (FPI) has certain weaknesses, such as the human factor and the use of a penetrant. Hence, it is desirable to find an alternative method.

White light interferometry was identified as a potential technology that could serve as the basis for an alternative to FPI. However, the current state of the art of white light interferometers is not designed for highly automated inspection methods. The two required aspects for a WLI-based automation solution were researched by the author and presented in this dissertation: the integration of interferometers in a programmable actuator system and the signal processing required to yield the desired inspection results.

The interferometer integration involved the identification of the issues associated with WLIs, most importantly the sensitivity to relative movement between the sensor and the part. Afterward, a suitable white light interferometer was specified, and a software system that can cope with the sensor's challenges was designed and integrated with an actuator system that moves both the sensor and the part. The researched and developed system performs all the required calculations to parameterize the sensor hardware and avoid common WLI issues while still achieving massively parallel scanning performance.

The signal processing part involves 1) the development of interferometry data pre-processing, 2) fast and precise methods for aligning and combining the individual point clouds of surface scans with potentially low feature density, and 3) the dimensional reduction of the captured data to allow for robust two-dimensional image processing.

Whereas these methods are universally applicable to a wide array of different scenarios, this dissertation also discusses methods specific to crack detection in interferometry scans. The latter involves

crack detection in images, based on discontinuities, as well as crack thinning and segmentation. Furthermore, a method for crack verification in the image domain and a sectioning-based verification method are introduced.

A comparison with FPI shows that the novel method is a viable alternative. Scanning times for a CFM-56 outer liner are about eight hours with a single robot and five hours with two robots in parallel. Due to the fully automated nature of the system, the human factor has no impact on the quality of the results. Therefore, all research objectives introduced at the beginning of this dissertation have been met. In addition, the method is currently officially used in parallel with FPI at Lufthansa Technik AG in Hamburg, where the method was in an industrialization phase at the time of writing. It should be mentioned that FPI is a mature technique with little room for further optimization. The novel method introduced in this dissertation, however, not only scales with more powerful hardware but also includes several aspects that can be researched in future works. This would advance the technology even further. These aspects are identified in the following outlook.

## 7.2 Outlook

Although the method already shows promising results, there are several possibilities that would allow for further optimization. There are two main aspects worth considering, which are, unfortunately, opposing demands: increased detection performance and decreased scan times.

### 7.2.1 Increased Detection Performance

The vertical resolution of the method is solely dependent on the parameters of the interferometric sensing, such as central wavelength, bandwidth and scanning speed. The lateral resolution on the other hand is simply a function of the pixel array size and the optical magnification. Fortunately, the vertical resolution is very high, and for technical surfaces is limited only by the surface roughness. Therefore, a further increase of vertical resolution is neither practical nor necessary. At 20  $\mu\text{m}$ , the lateral resolution is several orders of magnitude smaller. Because of this mismatch, crack detectability is largely limited by the lateral resolution; an increase would, therefore, lead to increased sensitivity to smaller cracks as well as more pixels representing broader discontinuities, leading to more robust detection results. It can therefore be expected that a system with higher lateral resolution would show a better NDI ( $a_{90}$  and  $a_{90/95}$ ) performance. The author performed tests with 10  $\mu\text{m}$  resolution; however, doubling the resolution leads to approximately four times the amount of data. With the current state-of-the-art processing power, that amount of data was not feasible to be processed under the given time constraints, and a higher resolution system was, therefore, postponed.

Another option to improve the detection performance, is by developing more powerful algorithms. As with higher resolution systems, this would decrease the time performance and would, therefore, demand more powerful or even specialized computational platforms. Options include everything from small changes, such as tuning solver methods to completely new crack-detection approaches, for in-

stance in three dimensions. Similarly, the crack verification could further be improved, potentially by more precise crack models or even by using machine learning approaches. The author of this dissertation has co-authored a publication exploring the use of machine learning in this area [SOB18].

### 7.2.2 Increased Measurement Speeds

To reduce time requirements, besides using more powerful computational hardware or optimized algorithms and parameters, two methods can generally be employed: increased parallelization and speeding up the scanning method.

A relatively straightforward way to increase parallelization can be realized by using several actuators and interferometers in parallel, with a centralized software to combine the parallel data parts.

Possibly the greatest impact on the reduction of scanning speeds is that the part's motion is discontinuous. That is, the part has to be moved to the next discrete measurement position and be allowed to reach steady state. Then a vertical measurement movement is performed with the interferometer. This process is repeated until a measurement ring is completed, leading to a collection of individual scan matrices with overlaps. In 2000, Olszak introduced a *continuous* white light scanning method [Ols00] based on the idea that a tilted interferometer has a velocity component toward the surface, even when the part is moved tangentially; this leads to signal modulation and therefore allows white light interferometric imaging. Such a continuous method would enable dramatically increased scanning speeds but would also increase the complexity of all steps involved in the WLI processing chain. The author of this dissertation has co-authored two publications on feasibility analyses for integrating this continuously scanning interferometry method in an automated inspection system. The conclusion was that the approach appears highly promising and is worth investigating further [Bah+17a] [Bah+17b].

Aside from further improvements, an automated repair strategy using the data generated by the method presented in this dissertation could be of great interest. The author of this dissertation has co-authored a publication describing a possible full chain from part scan to part repair [Sch+16].

# Appendices

# Appendix A

## Introduction to White Light Interferometry

In this appendix, the physics of white light interferometry are introduced. It should be emphasized that the presented introduction is tailored to the requirements of this dissertation and thus does not include all of the facets and possible details of white light interferometry.

### A.1 Basics of Interferometry

“The agreement of the results seems to show that light and magnetism are affections of the same substance, and that light is an electromagnetic disturbance propagated through the field according to electromagnetic laws.”

---

*James Clerk Maxwell, 1865*

J. C. Maxwell deduced in 1865 that light is an electromagnetic wave [Max65, p. 499], consisting of both oscillating E- and B-fields. Since a wave is a disturbance that propagates through space over time, it is possible to express the wave in terms of time and position. For an electromagnetic scalar wave, this can be expressed in terms of the wave’s E-field as  $E(z, t) = f(z - \nu t)$  [LKW95, p. 18], with  $\nu$  being the frequency of the wave. For a derivation of the full wave equation of light, see [ibidem, pp. 15].

In 1888, W. Hallwachs electrically charged metal plates and subjected them to UV light [Mes15, p. 464]. He found that electrons were emitted by negatively charged plates, and that the electrons’ energy depended on the frequency of the light rather than the intensity, which instead determined the rate of emittance [ibidem, p. 464]. This observation, however, cannot be explained by the theory of electromagnetic waves alone. A. Einstein found a solution for this problem by suggesting that light

energy is quantized [Mes15, p. 672], leading to the Einstein equation:  $E = h\nu - W_0$ . Here,  $h$  is Planck's constant ( $\approx 6.626 \cdot 10^{-34}$  Js),  $\nu$  is the frequency of the light, and  $W_0$  is the work function (which describes the work required to remove an electron from the metal plate). This quantized energy of light can also be expressed in terms of speed of light ( $c \approx 3 \cdot 10^8$  m/s) and its wavelength:  $E = h\nu - W_0 = hc/\lambda - W_0$  [TM15, page 1163]. These quantized “packages” of electromagnetic waves act like particles. They are called photons, a name proposed by G. N. Lewis in 1925 [Mes15, p. 672]. The phenomenon that photons are both waves and particles is known as wave-particle duality; it explains both the interaction with matter (the photo-electric effect) and the wave-like features of light.

One of the aforementioned wave phenomena is interference. As the name implies, two or more waves are superimposed and interfere with each other constructively or destructively. Since light is an electromagnetic wave with frequencies in the Petahertz range (see Figure 2.6 on page 24), it is impossible to observe an individual period with the current state of the art. What can be observed, however, are variations in light intensity. Intensity is defined as energy per area and time interval [LKW95, p. 25]. Since energy is proportional to the square of the electric field, it follows that  $I(z, t) \propto E^2(z, t)$ .

In order to observe static interference phenomena, a static phase relationship between the interfering waves is required. In other words, their time-dependance must be the same except for a phase contribution. For light, this can be expressed in terms of the light's electric fields in accordance with [Mes15, p. 534]:

$$E_1(t) = \text{const} \cdot E_2(t + \tau) \quad (\text{A.1})$$

This relationship is known as temporal coherence [LKW95, p. 31]. Equation A.1 states that the electric fields  $E_1$  and  $E_2$  of two light beams, differ only by a constant phase  $\tau$  and a constant amplitude scaling factor, meaning that static interference is possible in principle. The time difference can be negative, resulting in a delayed signal,<sup>1</sup> or positive, resulting in the second E-field being ahead of the first one. In practice, such a scenario is created by splitting and recombining a light beam. In the delay case, the path that the second wave has to travel would be longer; in the lead case, it would be shorter than that of the first path. There are several ways of implementing interferometers, one of which was invented by A. Michelson for his 1881 experiment to disprove the existence of the aether [TM15, p. 1123], known as the Michelson interferometer (see Figure A.1 on page A4). Here, the central component is a beam splitter, an amplitude-splitting optical device. The divided light beams travel toward two mirrors, are reflected back, pass through the beam splitter once more and can be observed upon recombination. If certain requirements are met, the superimposed light waves create interference patterns. For a harmonic function  $g(t)$  that is split in an interferometer, the resulting function can be

<sup>1</sup>Since time delays are typically written by subtracting the delay, this may be expressed as:

$$E_1(t) = \text{const} \cdot E_2(t - (-\tau)) = \text{const} \cdot E_2(t - \tilde{\tau})$$

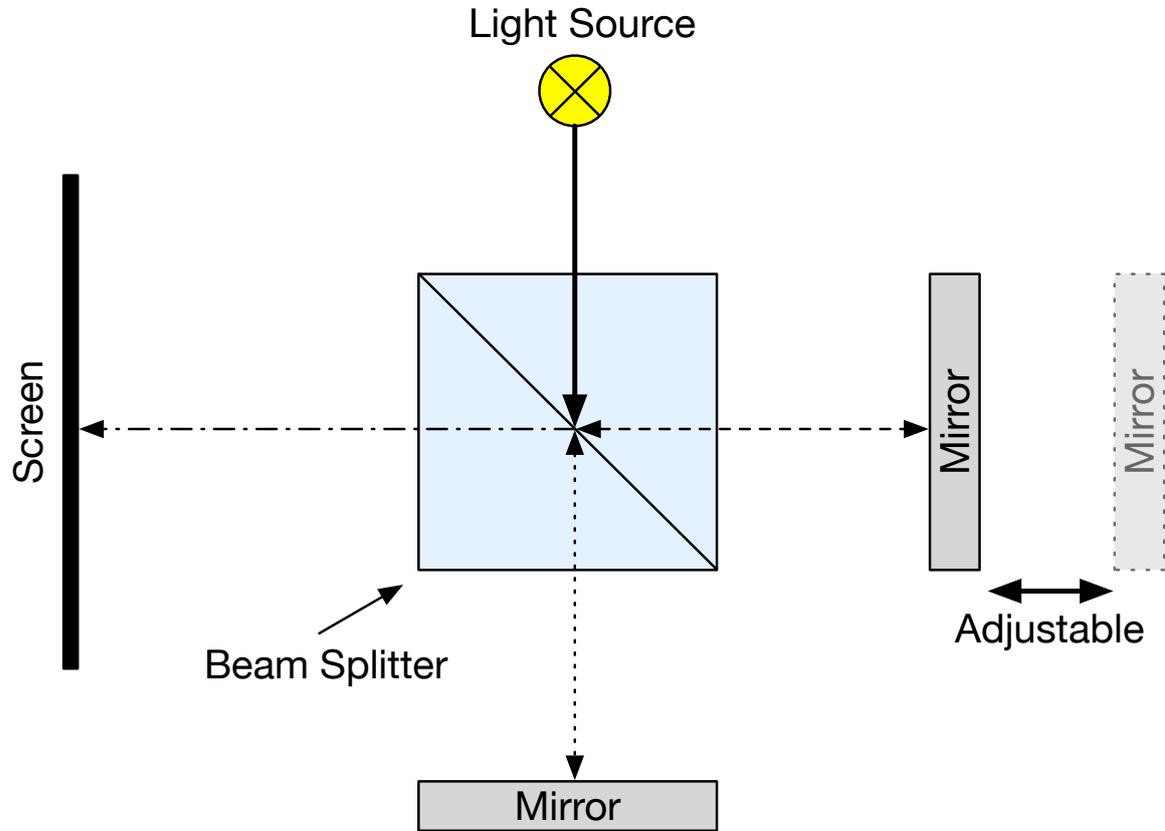


Figure A.1: Michelson interferometer (based on [LKW95, p. 32]).

expressed as:

$$g(t) = g_1(t) + g_2(t) = \cos(2\pi ft) + \cos(2\pi ft + \varphi) \quad (\text{A.2})$$

It should be noted that the phase shift in equation A.2 depends only on the length difference of the two interferometer arms and thus on the position of the mirrors in A.1. Since the phase is therefore not time-dependent, the resulting function is again a harmonic function. This can easily be proven with Euler's identity, for instance stated in [Zei+12, p. 57], with  $j$  substituted as the complex unit  $j = \sqrt{-1}$  as is common convention in engineering:

$$e^{\pm jx} = \cos x \pm j \sin x \quad (\text{A.3})$$

From equation A.3 follows the well-known fact that a cosine can be expressed in terms of exponentials:

$$\begin{aligned} \cos x &= \frac{1}{2} \cdot ((\cos x + j \sin x) + (\cos x - j \sin x)) \\ &= \frac{1}{2} \cdot (e^{jx} + e^{-jx}) \end{aligned} \quad (\text{A.4})$$

Given the situation described in equation A.2, the cosines can be expressed with equation A.4 as follows:

$$g(t) = \frac{1}{2} \cdot \left( e^{j2\pi ft} + e^{-j2\pi ft} \right) + \frac{1}{2} \cdot \left( e^{j(2\pi ft + \varphi)} + e^{-j(2\pi ft + \varphi)} \right) \quad (\text{A.5})$$

The difference between the two harmonic functions is the phase angle  $\varphi$ . Thus, if the two terms are to be recombined, the exponentials have to be rewritten in terms of  $\varphi/2$ . This can be achieved by adding  $\varphi/2 - \varphi/2 = 0$  to the exponential components of the unshifted cosine and by splitting the phase angle of the shifted cosine into  $\varphi = \varphi/2 + \varphi/2$  in equation A.5:

$$g(t) = \frac{1}{2} \cdot \left( e^{j(2\pi ft + \varphi/2 - \varphi/2)} + e^{-j(2\pi ft + \varphi/2 - \varphi/2)} \right) + \frac{1}{2} \cdot \left( e^{j(2\pi ft + \varphi/2 + \varphi/2)} + e^{-j(2\pi ft + \varphi/2 + \varphi/2)} \right) \quad (\text{A.6})$$

Equation A.6 can then be refactored by pulling the exponentials  $e^{-j\varphi/2}$  and  $e^{+j\varphi/2}$  as follows:

$$\begin{aligned} g(t) &= \frac{1}{2} \cdot e^{-j\varphi/2} \cdot \left( e^{j(2\pi ft + \varphi/2)} + e^{-j(2\pi ft + \varphi/2)} \right) + \frac{1}{2} \cdot e^{j\varphi/2} \cdot \left( e^{j(2\pi ft + \varphi/2)} + e^{-j(2\pi ft + \varphi/2)} \right) \\ &= \frac{1}{2} \cdot \left( e^{j\varphi/2} + e^{-j\varphi/2} \right) \cdot \left( e^{j(2\pi ft + \varphi/2)} + e^{-j(2\pi ft + \varphi/2)} \right) \end{aligned} \quad (\text{A.7})$$

It follows directly from equation A.7 with equation A.4:

$$g(t) = \frac{1}{2} \cdot \left( 2 \cdot \cos\left(\frac{\varphi}{2}\right) \right) \cdot \left( 2 \cdot \cos\left(2\pi ft + \frac{\varphi}{2}\right) \right)$$

Hence, the resulting signal of  $f_1(t) + f_2(t)$  is:

$$g(t) = \cos(2\pi ft) + \cos(2\pi ft + \varphi) = 2 \cdot \cos\left(\frac{\varphi}{2}\right) \cdot \cos\left(2\pi ft + \frac{\varphi}{2}\right) \quad (\text{A.8})$$

As expected, the combination of a cosine and a phase-shifted cosine with equal frequency results in a cosine of the same frequency. The overall harmonic oscillation is shifted by half the phase shift of the phase-shifted cosine, and the overall amplitude changes in terms of a cosine with half the phase shift. It can be seen in equation A.2 that the sum of the two harmonics is largest when they are in phase at  $\varphi = 0$  and  $k \cdot 2\pi$ , with  $k \in \mathbb{Z}$ . In those cases, the maximum amplitude is  $\max(g(t)|_{\varphi=0}) = \max(\cos(x)) + \max(\cos(x)) = 2$ . Conversely the sum is smallest whenever the harmonics are out of phase (at  $(2 \cdot k + 1) \cdot \pi$ , with  $k \in \mathbb{Z}$ ). In those cases, the amplitude is  $\cos(x) + \cos(-x) = \cos(x) - \cos(x) = 0$ . Since the overall phase shift depends only on the angle  $\varphi$  in the second cosine term in equation A.2, the maximum amplitude changes in the interval  $[0; 2]$  with a cosine in terms of  $\varphi/2$ , see equation A.8.

Therefore, at 0 rad phase shift, for instance, twice the amplitude will be observed (constructive interference), whereas at  $\pi$  rad phase shift, the two waves will cancel each other out completely (destructive interference). This is illustrated in A.2. As can be seen in equation A.8, the magnitude of

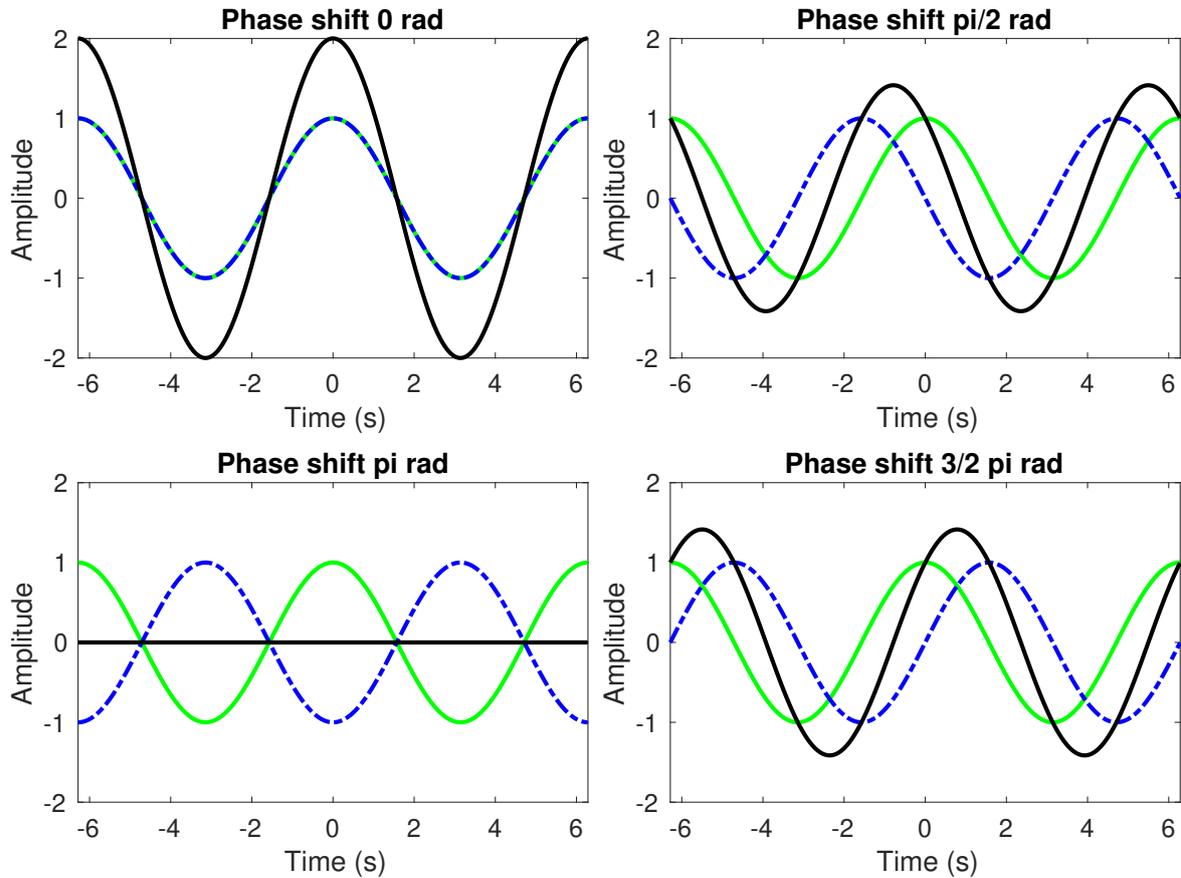


Figure A.2: Resulting interference waves (black) for the sum of a cosine function (green) with a phase-shifted copy (dotted blue lines) for different phase angles.

the observed interference depends on the phase shift  $\varphi$ , which conversely means that an observation of the magnitude allows estimation of the phase. Unfortunately, due to the cosine function in the magnitude, the estimated phase shift is not unique.

## A.2 White Light Interferometry

An important consideration is the circumstances in which the constant phase relationship demanded by equation A.1 can be fulfilled. In practice, light from one and the same source is used, by splitting the light into two paths. However, even then, the interference phenomena cannot be observed under all circumstances. One reason is that real light sources are not ideal point sources. Thus, two separate parts of the same light source may generate light through independent processes. If that is the case, light from two separate parts of the same light source act as if they are from two different sources. The degree of how strongly light from different parts of a source may interfere is called spatial coherence.

A second reason is temporal coherence. In light sources that are based on spontaneous emission, light is generated by a large number of electromagnetic dipoles, which are dominant only for a limited

period. Afterward, an entirely different set of dipoles takes over, resulting in a spontaneous phase jump. Since electromagnetic waves — such as light — travel with the relative speed of light  $c_r$ , this coherence time  $\tau_c$  corresponds to a coherence length for the material which the waves travel through [Mes15, p. 534]:

$$L_c = c_r \cdot \tau_c \quad (\text{A.9})$$

The coherence time can be as large as  $10^{-8}$  s for an individual atom [ibidem, p. 534]. Since the coherence time is basically a dampening time, the described mechanism does not result in a mono-frequent signal but rather in a mixture of different spectral components [ibidem, p. 534], namely white light.

If the resulting signal has a uniform spectrum centered around a central frequency  $f_0$ , with a bandwidth of  $BW_f = 2 \cdot \Delta f$ , then the half-bandwidth is approximately anti-proportional to the coherence time:  $\Delta f \approx 1/\tau_c$  [ibidem, p. 534]. Given that relationship, it follows from equation A.9 that:

$$L_c \approx \frac{c}{\Delta f} \quad (\text{A.10})$$

In most data sheets, however, the bandwidth and spectrum center are expressed in wavelengths, not frequencies. If the analysis is to be performed in the frequency domain, both the wavelength bandwidth and the central wavelength have to be mapped. The mapping between the central wavelength  $\lambda_0$  and the central frequency  $f_0$  is directly possible through the following well-known relationship between frequency and wavelength:

$$v = \lambda \cdot f \quad (\text{A.11})$$

For light in vacuum,  $v$  is the speed of light:  $v = c \approx 3 \cdot 10^8$  m/s. It follows for the central frequency that:

$$f_0 = c/\lambda_0 \quad (\text{A.12})$$

For example, for a red LED with a central wavelength of  $\lambda_0 = 648$  nm, the central frequency is  $f_0 = \frac{c}{\lambda_0} \approx 463$  THz

However, the mapping between the wavelength delta  $\Delta\lambda$  and the frequency delta  $\Delta f$  needs to be derived. The mapping is illustrated in Figure A.3 on page A8. The left graph shows a central wavelength  $\lambda_0$  and a wavelength extent  $2 \cdot \Delta\lambda$ , which is centered around  $\lambda_0$ . This is indicated by three vertical arrows. Given the initial definition, the positions of the three wavelength components are  $\lambda_0 - \Delta\lambda$ ,  $\lambda_0$  and  $\lambda_0 + \Delta\lambda$ . Thus, the “bandwidth” in terms of lambda  $BW_\lambda$  can be found by subtracting the position of the leftmost wavelength component from the rightmost one:  $BW_\lambda = \lambda_0 + \Delta\lambda - (\lambda_0 - \Delta\lambda) = 2\Delta\lambda$ . All three wavelength components are then mapped into the frequency domain using equation A.11, as shown in the right graph in Figure A.3. Due to the inverse nature

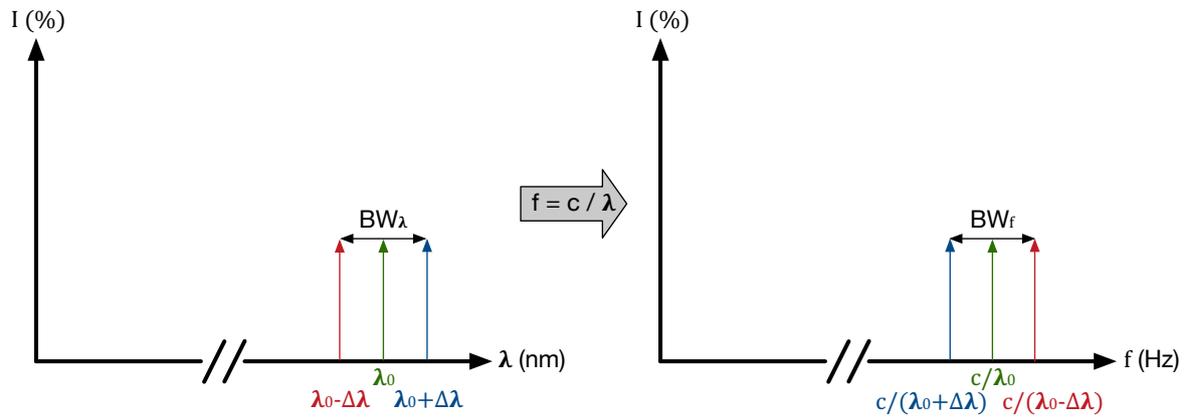


Figure A.3: Illustration of the mapping between the wavelength delta  $\Delta\lambda$  and the bandwidth  $\Delta f$ . Only the right-sided spectrum is shown.

of the mapping ( $f = c/\lambda$ ), the order of the frequency components is reversed in comparison to the wavelength components. The three resulting frequencies are thus  $\frac{c}{\lambda_0 + \Delta\lambda}$ ,  $\frac{c}{\lambda_0}$  and  $\frac{c}{\lambda_0 - \Delta\lambda}$ . The bandwidth in terms of  $f$  can be calculated by subtracting the leftmost frequency component from the rightmost one:

$$\begin{aligned} BW_f = 2\Delta f &= \frac{c}{\lambda_0 - \Delta\lambda} - \frac{c}{\lambda_0 + \Delta\lambda} = c \cdot \frac{\lambda_0 + \Delta\lambda - (\lambda_0 - \Delta\lambda)}{(\lambda_0 + \Delta\lambda) \cdot (\lambda_0 - \Delta\lambda)} \\ &= c \cdot \frac{2\Delta\lambda}{\lambda_0^2 - \Delta\lambda^2} \end{aligned} \quad (\text{A.13})$$

Since the wavelength delta is typically much smaller than the position of the central wavelength ( $\lambda_0 \gg \Delta\lambda$ ), equation A.13 can be simplified:

$$\begin{aligned} BW_f = 2\Delta f &\approx 2c \cdot \frac{\Delta\lambda}{\lambda_0^2} \\ \Leftrightarrow \Delta f &\approx c \cdot \frac{\Delta\lambda}{\lambda_0^2} \end{aligned} \quad (\text{A.14})$$

With the solution for  $\Delta f$  (equation A.14) inserted into equation A.10, an expression for the coherence length in terms of wavelengths can be found:

$$L_c \approx \frac{c}{\Delta f} \approx \frac{c}{c \cdot \frac{\Delta\lambda}{\lambda_0^2}} \approx \frac{\lambda_0^2}{\Delta\lambda} \quad (\text{A.15})$$

Furthermore, with  $\lambda_0 = c/f_0$ , a relationship between the frequency and the wavelength domains can

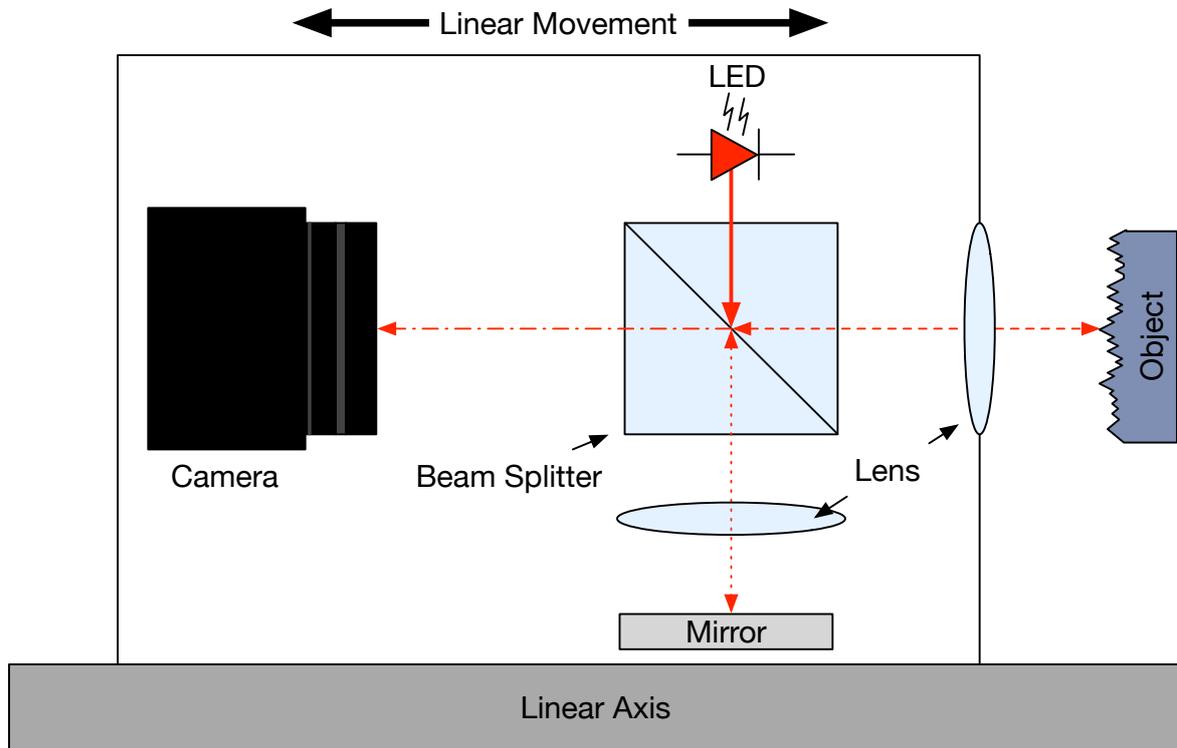


Figure A.4: Principle of white light interferometry.

be found from equation A.14:

$$\begin{aligned} \Delta f &\approx c \cdot \frac{\Delta \lambda}{\lambda_0^2} \approx c \cdot \frac{\Delta \lambda}{\lambda_0 \cdot \frac{c}{f_0}} \approx \frac{\Delta \lambda \cdot f_0}{\lambda_0} \\ \Leftrightarrow \frac{\Delta f}{f_0} &\approx \frac{\Delta \lambda}{\lambda_0} \end{aligned} \quad (\text{A.16})$$

As an example, for a light source with a 648 nm central wavelength and a uniform wavelength extent of 10 nm, the coherence length is  $L_c \approx \frac{(648 \text{ nm})^2}{10 \text{ nm}} = 41.99 \text{ } \mu\text{m}$ . This means that if light emitted by such a light source is split and then recombined, significant interference phenomena can be observed only if the difference in path lengths is smaller than 41.99  $\mu\text{m}$ . This breaks the ambiguity present for a mono-frequent signal (see equation A.8) and allows for an absolute measurement.

A device using this principle is shown in Figure A.4, the white light interferometer (WLI). A WLI is essentially a Michelson interferometer with one of the mirrors removed; the target surface replaces the mirror. Instead of an observation screen, a laterally discrete recording device — such as a CCD or CMOS camera — is used. The setup may contain further optical components to optimize the device's performance or form-factor. The device is encased in a suitable chassis and mounted on a precise linear actuator. If a highly reflective sample is scanned, the device is essentially a regular Michelson interferometer, and typical interference patterns with a predictable phase relationship can be observed. Rough surfaces, however, result in speckles, which display a random but constant phase.

Hence, instead of evaluating the phase, the entire device is shifted in a vertical scanning motion within the expected working distance. The resulting interference patterns are recorded with the camera, with sufficiently many vertical samples, while the linear axis performs a continuous motion. Such a vertically scanning interferometer is also known as a coherence radar [DHV92, pp. 919].

It will be shown in the following section that the superposition of the split signal leads to the (auto-)correlation of the signal with itself. Since the light source under consideration has a spectrum that follows a Gaussian distribution, models for that signal in the frequency and wavelength domains have to be derived. Since a Gaussian signal does not have a clear cutoff, a meaningful choice, namely the full width at half maximum (FWHM) bandwidth, is defined. Afterward, the resulting signal in the time domain is determined, which will then be used to find the effective FWHM coherence length. Furthermore, it will be shown that for a light source with an approximately Gaussian spectrum, the resulting signal of such a lateral scanning process is an amplitude-modulated Gaussian for each pixel that was within measurement range. The resulting signals are known as correlograms (illustrated in Figure A.6), because a coherence radar is an opto-mechanical autocorrelator. If the maxima of these correlograms are determined, the surface can be observed with very high accuracy (in the micrometer range [JH00, p. 201]).

### A.3 Derivation of the Resulting Correlogram for an LED

As mentioned in the previous section, in order to perform the measurement, a vertical scanning motion has to be performed. A change in the sensor's state of motion requires a force to be exerted on the interferometer by the linear actuator (see subsection 4.2.1). The actual measurement is performed only when the sensor has reached constant velocity. Therefore, the displacement is  $d = v_{\text{end}} \cdot t$  during the measurement if the initial velocity was zero. In other words, the distance  $d$  and the time  $t$  are equivalent descriptors, except for the proportionality factor  $v_{\text{end}}$ .

#### A.3.1 Intensity of the Superimposed Waves

In a Michelson interferometer, for any point  $t$  in time, two field components  $E_1(t)$  and  $E_2(t)$  are superimposed. As explained in the previous section,  $E_1(t)$  and  $E_2(t)$  are derived from a single field  $E_{\text{Source}}(t)$  that was split by an optical beam splitter. The field  $E_1(t) \equiv E(t)$  has a fixed path length relative to the reference mirror, whereas  $E_2(t)$  has a variable path length to the object under test. Because both fields are equal besides their amplitude and a time shift  $\tau$ , the second E-field can be expressed as  $E_2(t) \equiv \tilde{E}_1(t + \tau)$ . The time shift  $\tau$  can either be negative, which indicates a delay because the second path is longer; zero, which indicates equal path length; or positive, which indicates a shorter path length. The tilde signifies that the amplitude may differ from  $E(t)$ , because the object under test might not have the same reflectivity as the reference mirror. The superposition of the two E-fields is thus:

$$E_{\text{Superimposed}}(t, \tau) = E(t) + \tilde{E}(t + \tau) \quad (\text{A.17})$$

In order to perform any kind of measurement, power needs to be dissipated, which is related to the square of a signal. Since intensities are also related to the square of the electric fields, this means that intensities are observed rather than the actual signal. More specifically, an electric field's intensity can be determined through its power and can thus be written as  $I(\underline{r}) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} E(\underline{r}, t) \cdot E^*(\underline{r}, t) dt$ , in accordance with [LKW95, p. 27]. Because only a specific point is considered, the positional component  $\underline{r}$  can be ignored, reducing the equation to:

$$I = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} E(t) \cdot E^*(t) dt \equiv \langle EE^* \rangle \quad (\text{A.18})$$

Thus, the intensity of the superimposed fields in equation A.17 is simply:

$$I(\tau) = \langle E_{\text{Superimposed}} E_{\text{Superimposed}}^* \rangle = \langle (E(t) + \tilde{E}(t + \tau)) \cdot (E^*(t) + \tilde{E}^*(t + \tau)) \rangle \quad (\text{A.19})$$

Expanding equation A.19 delivers the following expression for the intensity:

$$I(\tau) = \langle E(t)E^*(t) + \tilde{E}(t + \tau)\tilde{E}^*(t + \tau) + E(t)\tilde{E}^*(t + \tau) + \tilde{E}(t + \tau)E^*(t) \rangle \quad (\text{A.20})$$

The two mixed terms are complex conjugates of each other. Hence,  $E(t)\tilde{E}^*(t + \tau) + \tilde{E}(t + \tau)E^*(t)$  can be replaced with  $2\text{Re} \left( E^*(t)\tilde{E}(t + \tau) \right)$ .<sup>1</sup> Using this equality, equation A.20 can be simplified:

$$I(\tau) = \left\langle E(t)E^*(t) + \tilde{E}(t + \tau)\tilde{E}^*(t + \tau) + 2\text{Re} \left( E^*(t)\tilde{E}(t + \tau) \right) \right\rangle \quad (\text{A.21})$$

The  $\langle \cdot \rangle$ -Operator defined in A.18 expands to an integral operation and is thus linear. Therefore, the expression in equation A.22 can be decomposed into three parts:

$$I(\tau) = \langle E(t)E^*(t) \rangle + \langle \tilde{E}(t + \tau)\tilde{E}^*(t + \tau) \rangle + 2\text{Re} \left( \langle E^*(t)\tilde{E}(t + \tau) \rangle \right) \quad (\text{A.22})$$

The first two terms directly fulfill the intensity definition (see equation A.18) so they can be replaced with their intensity counterparts. The third term, however, needs to be expanded as an integral expression. With that in mind, equation A.22 can be changed to:

$$I(\tau) = I_0 + \tilde{I}_0 + 2\text{Re} \left( \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} E^*(t)\tilde{E}(t + \tau) dt \right) \quad (\text{A.23})$$

<sup>1</sup>Proof:  $E(t)\tilde{E}^*(t + \tau) + E^*(t)\tilde{E}(t + \tau) = E^*(t)\tilde{E}(t + \tau) + (E^*(t)\tilde{E}(t + \tau))^*$ . The complex number  $E^*(t)\tilde{E}(t + \tau)$  is substituted with the complex number  $A$ . Since  $A = \text{Re}(A) + j\text{Im}(A)$  and  $A^* = \text{Re}(A) - j\text{Im}(A)$ , it follows that  $A + A^* = \text{Re}(A) + j\text{Im}(A) + \text{Re}(A) - j\text{Im}(A) = 2\text{Re}(A)$ . Thus, it was shown that  $E(t)\tilde{E}^*(t + \tau) + E^*(t)\tilde{E}(t + \tau) = 2\text{Re} \left( E^*(t)\tilde{E}(t + \tau) \right)$ .

The integral part of equation A.23 is a cross-correlation function, or in the case of  $\tilde{E}(t) = E(t)$ , an autocorrelation function.

### A.3.2 Derivation of the Autocorrelation Term

For energy signals, cross-correlation is defined as follows [OL10, p. 206]:

$$f(t) \star g(t) = \int_{-\infty}^{\infty} f(t)^* \cdot g(t + \tau) dt \quad (\text{A.24})$$

where  $(\cdot) \star (\cdot)$  is the cross-correlation operator and  $(\cdot)^*$  is the complex conjugate. For the autocorrelation function,  $g(t) = f(t)$  holds. In terms of an electric field  $E(t)$ , the autocorrelation function  $\tilde{\Psi}_{EE}(\tau)$  thus equals:

$$\tilde{\Psi}_{EE}(\tau) = \int_{-\infty}^{\infty} E^*(t) \cdot E(t + \tau) dt \quad (\text{A.25})$$

The electric field is unknown, which means that the autocorrelation function cannot be directly calculated in the time domain. However, the autocorrelation function can be determined through the inverse Fourier transform of the power density spectrum (PDS). This is known as the Wiener-Khinchine theorem [Rei12, p. 176]:

$$\tilde{\Psi}_{EE}(t) = \int_{-\infty}^{\infty} \tilde{S}(f) \cdot e^{+2\pi j t f} df \quad (\text{A.26})$$

where  $\tilde{S}(f)$  is the power density spectrum, and  $e^{+2\pi j t f}$  is the inverse Fourier transform kernel, with  $j$  being the complex unit  $j \equiv \sqrt{-1}$ .

The results in the previous chapter regarding the coherence length assumed that the bandwidth has a clear cutoff. However, the light source under consideration is an LED, which cannot be assumed to have a uniformly distributed spectrum but rather can be modeled with a Gaussian spectrum [APR02, p. 5258]. In the one-dimensional case, a normal distribution can be expressed in terms of frequency as:

$$S(f) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2} \cdot \frac{(f-f_0)^2}{\sigma^2}} \quad (\text{A.27})$$

Since a normal distribution extends infinitely far, a bandwidth cutoff has to be defined for the model. In the literature, different bandwidth cutoff definitions can be found. The most common definitions are the  $1/e$  bandwidth and the full width at half maximum (FWHM) bandwidth [ibidem] [Stü+07]. The FWHM definition is chosen for the following derivations; it is illustrated for both frequencies and wavelengths in Figure A.5. First, the relationship between the standard deviation  $\sigma$  and the FWHM value has to be determined. Since this relationship is completely independent of any shifting, the

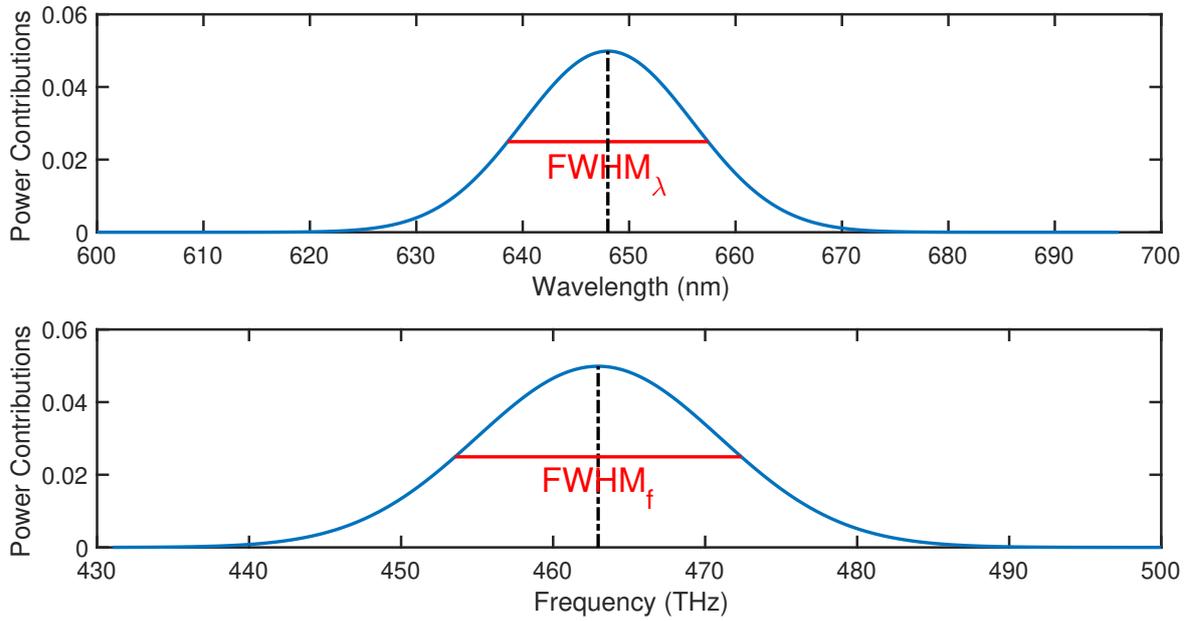


Figure A.5: Top panel: LED power-density spectrum distribution over wavelength (nm range). Bottom panel: LED power-density spectrum over frequency (THz range). The plots indicate the full width at half maximum (FWHM) bandwidths.

factor  $f_0$  in equation A.27 can be dropped. Furthermore, since the factor between the FWHM value and  $\sigma$  is independent of additional scaling, the normalization factor can be dropped as well. Lastly, the function is observed at the half-maximum point HM, resulting in a value of  $1/2$ :

$$g(\text{HM}) = e^{-\frac{1}{2} \cdot \frac{\text{HM}^2}{\sigma^2}} \equiv \frac{1}{2} = 2^{-1} \quad (\text{A.28})$$

Equation A.28 can be rearranged to solve for the half-maximum point using the natural logarithm  $\ln$ :

$$\begin{aligned} \ln \left( e^{-\frac{1}{2} \cdot \frac{\text{HM}^2}{\sigma^2}} \right) &= \ln (2^{-1}) \\ \Leftrightarrow -\frac{1}{2} \cdot \frac{\text{HM}^2}{\sigma^2} &= -\ln (2) \\ \Leftrightarrow (\text{HM})^2 &= \sigma^2 \cdot 2 \ln (2) \\ \Leftrightarrow \text{HM} &= \sigma \cdot \sqrt{2 \ln (2)} \end{aligned} \quad (\text{A.29})$$

As the name implies, the “full width at half maximum” value is two times the difference between the center point and the position at which the function first reaches half of the maximum value. The factor two means that both the lower and upper half maximum points are considered, resulting in the *full* width. Therefore, equation A.29 has to be doubled to obtain the correct relationship between the

FWHM value and  $\sigma$ :

$$\Leftrightarrow \text{FWHM} = \sigma \cdot 2\sqrt{2 \ln(2)} \quad (\text{A.30})$$

$$\Leftrightarrow \sigma = \frac{\text{FWHM}}{2\sqrt{2 \ln(2)}} \quad (\text{A.31})$$

Thus, if the light source's spectrum is normally distributed, and if the bandwidth is defined to cut off at the FWHM point, it follows from equations A.9, A.10 and A.15 together with equation A.30 that:

$$\Delta f = \frac{1}{\tau_c} = \frac{\Delta \lambda}{\lambda_0^2} \cdot c \equiv \text{FWHM} = 2\sqrt{2 \ln(2)} \cdot \sigma \quad (\text{A.32a})$$

$$\Leftrightarrow \sigma = \frac{\Delta f}{2\sqrt{2 \ln(2)}} \quad (\text{A.32b})$$

$$\Leftrightarrow \sigma = \frac{\Delta \lambda \cdot c}{\lambda_0^2 \cdot 2\sqrt{2 \ln(2)}} \quad (\text{A.32c})$$

With  $f = c/\lambda$  and equation A.32c, the normal distribution modeling the LED spectrum (equation A.27) can be expressed in terms of wavelengths:

$$\begin{aligned} S(f) &= \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2} \cdot \frac{(f-f_0)^2}{\sigma^2}} \\ \Leftrightarrow S(\lambda) &= \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp \left[ -\frac{1}{2} \cdot \frac{\left(\frac{c}{\lambda} - \frac{c}{\lambda_0}\right)^2}{\sigma^2} \right] \\ \Leftrightarrow &= \frac{1}{\sqrt{2\pi} \frac{\Delta \lambda \cdot c}{\lambda_0^2 \cdot 2\sqrt{2 \ln(2)}}} \cdot \exp \left[ -\frac{1}{2} \cdot \frac{\left(\frac{c}{\lambda} - \frac{c}{\lambda_0}\right)^2}{\left(\frac{\Delta \lambda \cdot c}{\lambda_0^2 \cdot 2\sqrt{2 \ln(2)}}\right)^2} \right] \\ \Leftrightarrow &= \frac{2\sqrt{2 \ln(2)} \cdot \lambda_0^2}{\sqrt{2\pi} \cdot c \cdot \Delta \lambda} \cdot \exp \left[ -\frac{1}{2} \cdot \left( \frac{\left(\frac{c}{\lambda} - \frac{c}{\lambda_0}\right) \cdot 2\sqrt{2 \ln(2)} \cdot \lambda_0^2}{\Delta \lambda \cdot c} \right)^2 \right] \\ \Leftrightarrow &= \frac{2\sqrt{\ln(2)} \cdot \lambda_0^2}{\sqrt{\pi} \cdot c \cdot \Delta \lambda} \cdot \exp \left[ -\left( 2\sqrt{\ln(2)} \cdot \frac{\left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right) \cdot \lambda_0^2}{\Delta \lambda} \right)^2 \right] \end{aligned} \quad (\text{A.33})$$

Furthermore, with equation A.32b, the LED spectrum can be expressed in terms of frequencies:

$$\begin{aligned}
S(f) &= \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2} \cdot \frac{(f-f_0)^2}{\sigma^2}} \\
&= \frac{1}{\sqrt{2\pi} \frac{\Delta f}{2\sqrt{2\ln(2)}}} \cdot \exp \left[ -\frac{1}{2} \cdot \left( \frac{f-f_0}{\frac{\Delta f}{2\sqrt{2\ln(2)}}} \right)^2 \right] \\
&= \frac{2\sqrt{2\ln(2)}}{\sqrt{2\pi}\Delta f} \cdot \exp \left[ -\frac{1}{2} \cdot \left( 2\sqrt{2\ln(2)} \cdot \frac{f-f_0}{\Delta f} \right)^2 \right] \\
&= \frac{2\sqrt{\ln(2)}}{\sqrt{\pi}\Delta f} \cdot \exp \left[ -\left( 2\sqrt{\ln(2)} \cdot \frac{f-f_0}{\Delta f} \right)^2 \right] \tag{A.34}
\end{aligned}$$

Since the scaling factor is irrelevant for the following discussion, it is omitted for now and is added again in the next subsection. The unscaled probability density spectrum (PDS) is referred to as  $S(f)$ , and the unscaled autocorrelation is referred to as  $\Psi_{EE}(t)$ .

The spectrum may be provided by the manufacturer as relative intensity over wavelength (see for instance [Vis16, p. 3]). The top panel in Figure A.5 on page A13 shows this type of normal distribution model in terms of power contribution over wavelength.

The PDS signifies how much the different sinusoidal components of a signal contribute to an overall signal. In the case of a Gaussian distribution, there is a large contribution around the central frequency  $f_0$ , with fast-decaying contributions as the spectral components move away from the single maximum at  $f_0$ . A Gaussian distribution is uniquely characterized by only two moments: its mean and variance. In equation A.27, the mean is  $f_0$  and the variance is  $\sigma^2$ . The goal is to find the signal's autocorrelation function, which can be found by calculating the inverse Fourier transform of the PDS. Therefore, it is sufficient to determine the first two moments of the PDS in this particular case. The PDS is expressed for both the frequency and the wavelength domain in equations A.33 and A.34, respectively. The next step is to determine the inverse Fourier transform of the normal distribution (see equation A.27) in order to find expressions for the autocorrelation signal in the time domain.

Inserting the normal distribution (equation A.27) into equation A.26, an expression for the autocorrelation function can be found using the inverse Fourier transform:

$$\Psi_{EE}(t) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2} \frac{(f-f_0)^2}{\sigma^2}} \cdot e^{+2\pi j t f} df$$

In order to solve the inverse Fourier transform integral, a substitution for  $f - f_0$  is made:

$$f - f_0 \equiv \nu \Leftrightarrow f = \nu + f_0 \tag{A.35a}$$

$$\Rightarrow \frac{d\nu}{df} = 1 \Leftrightarrow d\nu = df \tag{A.35b}$$

With the substitutions in equations A.35a and A.35b, the inverse Fourier transform integral can be rearranged as follows:<sup>1</sup>

$$\begin{aligned}\Psi_{EE}(t) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2}\frac{\nu^2}{\sigma^2}} \cdot e^{+2\pi jt(\nu+f_0)} d\nu \\ \Psi_{EE}(t) &= e^{+2\pi jtf_0} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2}\frac{\nu^2}{\sigma^2}} \cdot e^{+2\pi jt\nu} d\nu\end{aligned}\quad (\text{A.36})$$

By definition, an infinite integral over a normal distribution is precisely one. The idea for solving the integral in equation A.36 is to rearrange the expression in such a way that only a normal distribution remains underneath the integral. It should be noted that the term in front of the inverse Fourier transform kernel in the integral in equation A.36 is a normal distribution; however, the kernel itself also depends on the integration variable. Thus, as a first step, the two exponentials in equation A.36 are combined, ignoring the exponential factor in front of the integral:

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2}\frac{\nu^2}{\sigma^2}} \cdot e^{+2\pi jt\nu} d\nu = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2}\frac{\nu^2 - 4\pi jt\nu\sigma^2}{\sigma^2}} d\nu\quad (\text{A.37})$$

The denominator in the exponential in equation A.37 is an incomplete square expression. The square is completed as follows:

$$\begin{aligned}&\nu^2 - 4\pi jt\nu\sigma^2 \\ &= \nu^2 - 4\pi jt\nu\sigma^2 + (2\pi jt\sigma^2)^2 - (2\pi jt\sigma^2)^2 \\ &= (\nu - 2\pi jt\sigma^2)^2 - (2\pi jt\sigma^2)^2\end{aligned}\quad (\text{A.38})$$

With the completed square (equation A.38), equation A.37 can be rearranged as follows:

$$\begin{aligned}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2}\frac{\nu^2 - 4\pi jt\nu\sigma^2}{\sigma^2}} d\nu &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2}\frac{(\nu - 2\pi jt\sigma^2)^2 - (2\pi jt\sigma^2)^2}{\sigma^2}} d\nu \\ &= e^{\frac{(2\pi jt\sigma^2)^2}{2\sigma^2}} \cdot \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{1}{2}\frac{(\nu - 2\pi jt\sigma^2)^2}{\sigma^2}} d\nu\end{aligned}\quad (\text{A.39})$$

The  $2\pi jt\sigma^2$  term is a constant with respect to the integration variable and can thus be substituted with a constant  $\mu_0$ . Since the infinite integral over a normal distribution equals one, the integral expression

<sup>1</sup>The exponential  $e^{+2\pi jtf_0}$  can be factored out of the integral because it does not depend on  $\nu$ .

can be solved as follows, using  $j^2 = -1$ :

$$e^{\frac{(2\pi jt\sigma^2)^2}{2\sigma^2}} \cdot \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2} \frac{(\nu-\mu_0)^2}{\sigma^2}} d\nu = e^{\frac{(2\pi jt\sigma^2)^2}{2\sigma^2}} = e^{-(\sqrt{2}\pi t\sigma)^2} \quad (\text{A.40})$$

In order to arrive at the desired autocorrelation function, the factor  $e^{2\pi jt f_0}$  in equation A.36 (which was dropped temporarily as it was irrelevant for solving the integral) has to be taken into account. Inserting equation A.40 into equation A.36 yields the following expression:

$$\Psi_{EE}(t) = e^{2\pi jt f_0} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{1}{2} \frac{\nu^2}{\sigma^2}} \cdot e^{2\pi jt\nu} d\nu = e^{-(\sqrt{2}\pi t\sigma)^2} \cdot e^{2\pi jt f_0} \quad (\text{A.41})$$

Using Euler's identity, the second exponential can be expressed in terms of cosine and sine functions:

$$\Psi_{EE}(t) = e^{-(\sqrt{2}\pi t\sigma)^2} \cdot (\cos(2\pi f_0 t) + j \cdot \sin(2\pi f_0 t)) \quad (\text{A.42})$$

Finally, the factor  $\sigma$  in equation A.42 can be expressed in terms of frequencies using equation A.32b:

$$\begin{aligned} \Psi_{EE}(t) &= e^{-(\sqrt{2}\pi t\sigma)^2} \cdot (\cos(2\pi f_0 t) + j \cdot \sin(2\pi f_0 t)) \\ &= e^{-\left(\sqrt{2}\pi t \frac{\Delta f}{2\sqrt{2\ln(2)}}\right)^2} \cdot (\cos(2\pi f_0 t) + j \cdot \sin(2\pi f_0 t)) \\ &= e^{-\left(\frac{\pi t \Delta f}{2\sqrt{\ln(2)}}\right)^2} \cdot (\cos(2\pi f_0 t) + j \cdot \sin(2\pi f_0 t)) \end{aligned} \quad (\text{A.43})$$

Equation A.43 can be expressed in terms of wavelengths, using  $f_0 = c/\lambda_0$  and equation A.32a:

$$\Psi_{EE}(t) = e^{-\left(\frac{\pi t \cdot c \cdot \Delta \lambda}{2\sqrt{\ln(2)} \cdot \lambda_0^2}\right)^2} \cdot \left( \cos\left(2\pi \frac{c}{\lambda_0} t\right) + j \cdot \sin\left(2\pi \frac{c}{\lambda_0} t\right) \right) \quad (\text{A.44})$$

### A.3.3 Combining the Intensity Description with the Autocorrelation Function

The light's electric field has a constant complex scaling factor  $E_0$ , which was ignored during the preceding considerations. Therefore, the associated intensity scaling factor  $I_0$  for the unscaled autocorrelation function in equation A.42 is determined next, so that  $\Psi_{EE}(\tau) = 2I_0 \cdot \tilde{\Psi}_{EE}(\tau)$ . The factor of two derives from equation A.23. Since the scaling factor of the electric fields  $E_0$  is constant, it can be factored out of the integral in equation A.26. Both electric fields contain the factor, but since one of the fields is the complex conjugate, the term in front of the integral is  $I_0 = E_0 \cdot E_0^*$ . This can be

simplified as follows:

$$\begin{aligned}
 I_0 &= E_0 \cdot E_0^* \\
 I_0 &= (\operatorname{Re}(E_0) + j\operatorname{Im}(E_0)) \cdot (\operatorname{Re}(E_0) - j\operatorname{Im}(E_0)) \\
 I_0 &= \operatorname{Re}(E_0)^2 + \operatorname{Im}(E_0)^2 \\
 I_0 &= \left( \sqrt{\operatorname{Re}(E_0)^2 + \operatorname{Im}(E_0)^2} \right)^2 \\
 I_0 &= |E_0|^2
 \end{aligned} \tag{A.45}$$

With the intensity factor in equation A.45, the full autocorrelation can be expressed as follows:

$$\tilde{\Psi}_{EE}(\tau) = 2I_0 \cdot e^{-\left(\frac{\pi\tau\Delta f}{2\sqrt{\ln(2)}}\right)^2} \cdot (\cos(2\pi f_0\tau) + j \cdot \sin(2\pi f_0\tau)) \tag{A.46a}$$

$$\tilde{\Psi}_{EE}(\tau) = 2I_0 \cdot e^{-\left(\frac{\pi\tau \cdot c \cdot \Delta\lambda}{2\sqrt{\ln(2)} \cdot \lambda_0^2}\right)^2} \cdot \left( \cos\left(2\pi \frac{c}{\lambda_0} \tau\right) + j \cdot \sin\left(2\pi \frac{c}{\lambda_0} \tau\right) \right) \tag{A.46b}$$

Equations A.46a and A.46b are the solutions of the autocorrelation function in the time domain, expressed in terms of frequency  $f_0$  and wavelength  $\lambda_0$ , respectively. Hence, if equation A.46a is inserted in the equation for the resulting intensity signal A.23, the following equation can be obtained with  $\tilde{I}_0$  set to  $I_0$ :

$$I(\tau) = 2I_0 + 2I_0 \cdot \operatorname{Re} \left( e^{-\left(\frac{\pi\tau\Delta f}{2\sqrt{\ln(2)}}\right)^2} \cdot (\cos(2\pi f_0\tau) + j \cdot \sin(2\pi f_0\tau)) \right) \tag{A.47}$$

$$= 2I_0 \cdot \operatorname{Re} \left( 1 + e^{-\left(\frac{\pi\tau \cdot c \cdot \Delta\lambda}{2\sqrt{\ln(2)} \cdot \lambda_0^2}\right)^2} \cdot \left( \cos\left(2\pi \frac{c}{\lambda_0} \tau\right) + j \cdot \sin\left(2\pi \frac{c}{\lambda_0} \tau\right) \right) \right) \tag{A.48}$$

Extracting the real part leads to the following equation:

$$I(\tau) = 2I_0 \cdot \left( 1 + e^{-\left(\frac{\pi\tau\Delta f}{2\sqrt{\ln(2)}}\right)^2} \cdot \cos(2\pi f_0\tau) \right) \tag{A.49}$$

$$= 2I_0 \cdot \left( 1 + e^{-\left(\frac{\pi\tau \cdot c \cdot \Delta\lambda}{2\sqrt{\ln(2)} \cdot \lambda_0^2}\right)^2} \cdot \cos\left(2\pi \frac{c}{\lambda_0} \tau\right) \right) \tag{A.50}$$

With the definition  $\hat{I}_0 \equiv 2I_0$ , the previous equation can be rewritten as:

$$I(\tau) = \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{\pi\tau\Delta f}{2\sqrt{\ln(2)}}\right)^2} \cdot \cos(2\pi f_0\tau) \right) \quad (\text{A.51})$$

$$= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{\pi\tau \cdot c \cdot \Delta\lambda}{2\sqrt{\ln(2)} \cdot \lambda_0^2}\right)^2} \cdot \cos\left(2\pi \frac{c}{\lambda_0} \tau\right) \right) \quad (\text{A.52})$$

Equations A.51 and A.52 are the fully derived descriptions of the observable intensity function in the time domain, both in terms of frequency  $f_0$  and wavelength  $\lambda_0$ , if the reflectivities in both paths are equal. It can be seen that the intensity signal is an amplitude-modulated Gaussian function with a DC offset.

From equations A.51 and A.52, the effective FWHM coherence length can be determined. Since both the DC offset and the carrier function extend infinitely far without any attenuation, only the Gaussian part can be responsible for the coherence length. As mentioned before, a Gaussian function  $G(\cdot)$  also extends infinitely but is attenuated in both directions from its maximum. This means that a cutoff needs to be defined. In line with previous definitions, the full width at half maximum (FWHM) point is chosen for the cutoff. With  $L = c \cdot \tau \Leftrightarrow \tau = L/c$ , it follows from equation A.51:

$$\begin{aligned} G(L_{c,\text{FWHM}}/2) &= e^{-\left(\frac{\pi \cdot \frac{L_{c,\text{FWHM}}}{2 \cdot c} \cdot \Delta f}{2 \cdot \sqrt{\ln(2)}}\right)^2} \equiv \frac{1}{2} \\ \Leftrightarrow -\left(\frac{\pi \cdot L_{c,\text{FWHM}} \cdot \Delta f}{4 \cdot c \cdot \sqrt{\ln(2)}}\right)^2 &= \ln(1/2) = -\ln(2) \\ \Leftrightarrow (L_{c,\text{FWHM}})^2 &= \frac{16 \cdot c^2 \cdot \ln(2)^2}{\pi^2 \cdot \Delta f^2} \\ \Leftrightarrow L_{c,\text{FWHM}} &= \frac{4 \cdot c \cdot \ln(2)}{\pi \cdot \Delta f} \end{aligned} \quad (\text{A.53})$$

With equation A.14, the effective FWHM bandwidth — shown in terms of frequencies in equation A.53 — can be expressed in terms of wavelengths:

$$\begin{aligned} L_{c,\text{FWHM}} &= \frac{4 \cdot c \cdot \ln(2)}{\pi \cdot \Delta f} \\ &= \frac{4 \cdot c \cdot \ln(2)}{\pi \cdot c \cdot \frac{\Delta\lambda}{\lambda_0^2}} \\ &= \frac{4 \cdot \ln(2) \cdot \lambda_0^2}{\pi \cdot \Delta\lambda} \end{aligned} \quad (\text{A.54})$$

It also follows from A.15 that:

$$L_{c,\text{FWHM}} = L_c \cdot \frac{4 \cdot \ln(2)}{\pi} = L_c \cdot \frac{4 \cdot 2 \cdot \ln(2)}{2\pi} = L_c \cdot \left( \frac{2 \cdot \sqrt{2 \cdot \ln(2)}}{\sqrt{2\pi}} \right)^2 \quad (\text{A.55})$$

In a typical LED data sheet, for instance [Vis16], the central wavelength is either given directly or can be extracted from an intensity-over-wavelength plot. In the given example, the central wavelength (also known as the dominant wavelength) is  $\lambda_0 = 648 \text{ nm}$  (red). The coherence length, however, is not usually specified. Using the FWHM convention, the wavelength bandwidth  $\Delta\lambda$  can be extracted from the LED's intensity-over-wavelength plot. For instance, in [ibidem, p. 4, Figure 4] the bandwidth can be estimated to be around  $\Delta\lambda = 10 \text{ nm}$ . Thus, using equation A.54, the resulting FWHM coherence length is  $L_{c,\text{FWHM}} = \frac{4 \cdot \ln(2) \cdot (648 \text{ nm})^2}{\pi \cdot 10 \text{ nm}} = 37.05 \text{ }\mu\text{m}$ .

Using equation A.53, the intensity in equations A.51 and A.51 can be expressed in terms of FWHM coherence length:

$$\begin{aligned} I(\tau) &= \hat{I}_0 \cdot \left( 1 + e^{-\left( \frac{\pi \tau \Delta f}{2\sqrt{\ln(2)}} \right)^2} \cdot \cos(2\pi f_0 \tau) \right) \\ &= \hat{I}_0 \cdot \left( 1 + e^{-\left( \frac{\pi \cdot \tau \cdot \frac{4 \cdot c \cdot \ln(2)}{\pi \cdot L_{c,\text{FWHM}}}}{2\sqrt{\ln(2)}} \right)^2} \cdot \cos(2\pi f_0 \tau) \right) \\ &= \hat{I}_0 \cdot \left( 1 + e^{-\left( \frac{\tau \cdot c \cdot 2 \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}} \right)^2} \cdot \cos(2\pi f_0 \tau) \right) \end{aligned} \quad (\text{A.56})$$

$$= \hat{I}_0 \cdot \left( 1 + e^{-\left( \frac{\tau \cdot c \cdot 2 \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}} \right)^2} \cdot \cos\left( 2\pi \frac{c}{\lambda_0} \tau \right) \right) \quad (\text{A.57})$$

If the intensity is to be expressed purely in terms of FWHM coherence length, wavelength and distance, a substitution for  $\tau$  in equation A.57 needs to be found. If a light beam travels a distance  $z$  relative to a starting point  $z_0$ , the required time is  $\tilde{\tau} = \frac{z-z_0}{c}$ . Since the distance has to be covered to the object and back, the full time is  $\tau = 2 \cdot \frac{z-z_0}{c}$ . With this substitution, equation A.57 can be expressed in terms

of distance:

$$\begin{aligned}
I(\tau) &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{\tau \cdot c \cdot 2 \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi \frac{c}{\lambda_0} \tau\right) \right) \\
\Leftrightarrow I(z) &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{2 \cdot \frac{z-z_0}{c} \cdot c \cdot 2 \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi \frac{c}{\lambda_0} \cdot 2 \cdot \frac{z-z_0}{c}\right) \right) \\
&= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{4 \cdot (z-z_0) \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi \frac{2}{\lambda_0} \cdot (z-z_0)\right) \right) \quad (\text{A.58})
\end{aligned}$$

The resulting correlogram is plotted in Figure A.6.

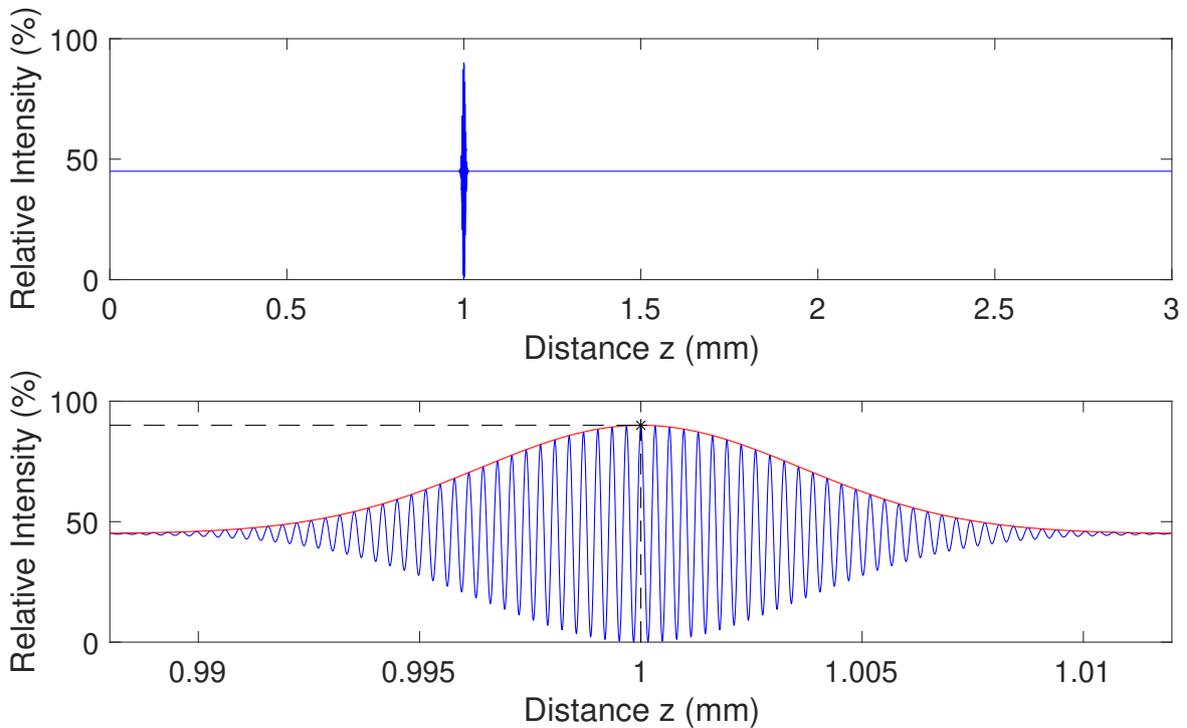


Figure A.6: Idealized correlogram for a single pixel. The top plot shows a correlogram with real values (648 nm central wavelength and 13 nm bandwidth, resulting in a coherence length of 5.14  $\mu\text{m}$ ). The bottom plot shows the same correlogram zoomed in around  $z_0 = 1$  mm. The bottom plot also illustrates the Gaussian envelope with the extracted maximum (relative amplitude of 90% at a distance of 1 mm to the object). It is important to note the difference in scale between the abscissae.

It can be seen in equation A.52 that there is an inverse relationship between the standard deviation and the bandwidth  $\Delta\lambda$  of the LED. This means that the narrower (more monochromatic) the bandwidth is, the broader the resulting correlogram becomes (larger standard deviation). In the extreme

case, only the cosine term remains, because the envelope is a constant 1:

$$\lim_{\Delta\lambda \rightarrow 0} e^{-\left(\frac{\pi\tau \cdot c \cdot \Delta\lambda}{2\sqrt{\ln(2)} \cdot \lambda_0^2}\right)^2} = e^{-\left(\frac{\pi\tau \cdot c \cdot 0}{2\sqrt{\ln(2)} \cdot \lambda_0^2}\right)^2} = 1$$

In that case, the resulting signal is not unique anymore but rather repeats itself every  $2\pi$  rad. However,

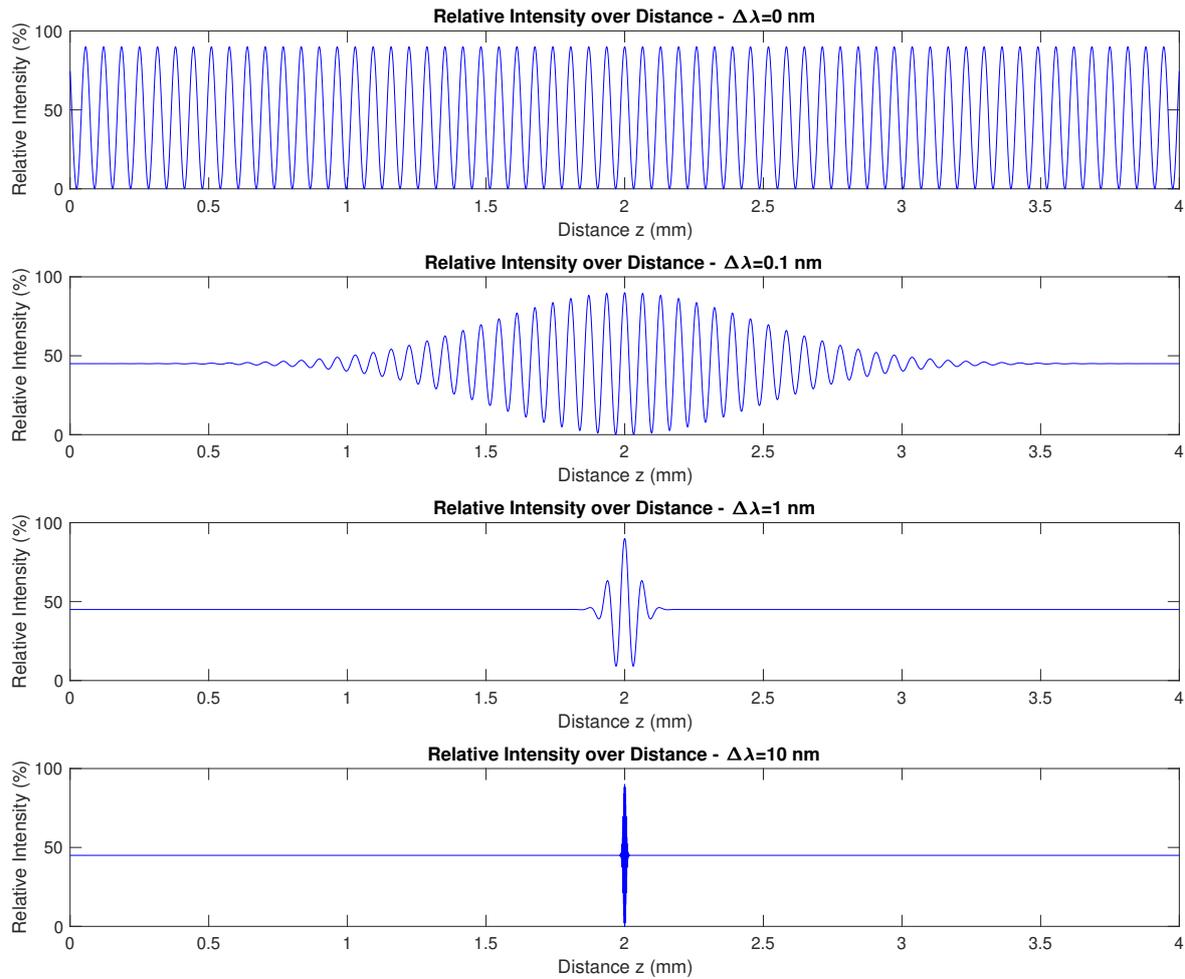


Figure A.7: The effect of the resulting correlogram is shown with various LED bandwidths and otherwise equal parameters (top plot: 0 nm, second plot: 0.1 nm, third plot: 1 nm, and bottom plot: 10 nm). The top three plots have a reduced modulation frequency (1/200) to better illustrate the effect of the bandwidth.

if the bandwidth becomes broader (i.e. whiter), the standard deviation becomes smaller and thus the signal becomes unique. The effect of the changing bandwidth is illustrated in Figure A.7. This is precisely why the technology is called *white* light interferometry.

### A.3.4 Heterodyne Scanning

In the previous subsection, equations for the observed correlogram in terms of frequency  $f_0$  and wavelength  $\lambda_0$  were derived for steady-state. In this subsection, the equations are extended to consider

the vertical movement executed in a coherence radar.

Because a coherence radar is moving toward or away from the scanned surface during the measurement, the Doppler effect has to be considered. Figure A.8 shows the situation with the WLI moving

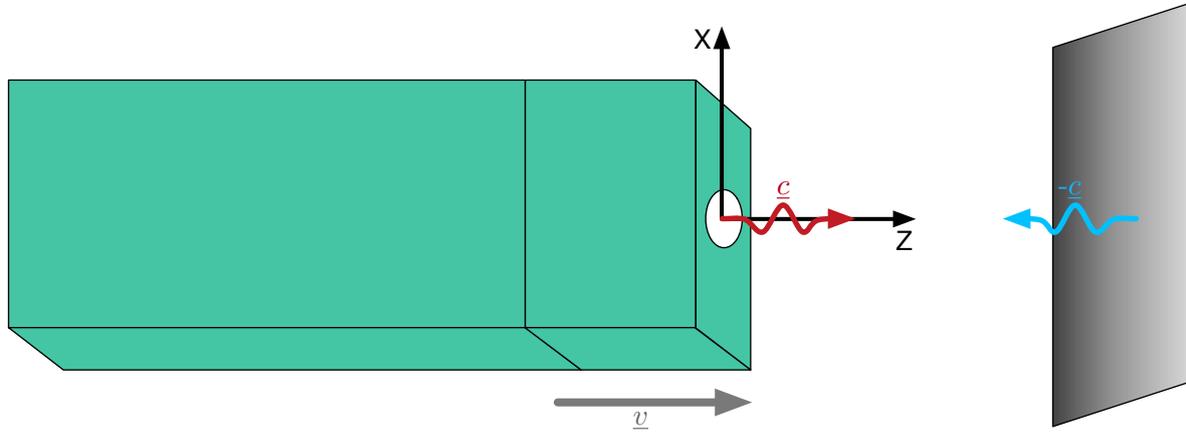


Figure A.8: Illustration of the situation that leads to the Doppler frequency  $f_D = f_0$ .

at a constant velocity  $v$ . The light waves move at the speed of light to the part and then back to the sensor. Since both the sensor and the light waves are aligned with the  $z$  axis, the relationships do not have to be analyzed vectorially.

Because the sensor is moving toward the object while emitting the light, there is a Doppler shift that increases the frequency. If the object is viewed as an intermediate “receiver” of the emitted light, this change in frequency can be calculated using the following equation for the Doppler effect with an approaching source (the WLI) and a stationary receiver (the object); equation adapted from [TM15, p. 474]:

$$\lambda_{\text{Receiver}} = \frac{c - v}{f_{\text{Source}}} \quad (\text{A.59})$$

With  $\lambda = c/f$ , it follows from equation A.59:

$$\begin{aligned} \lambda_{\text{Object}} &= \frac{c - v}{f_{\text{Source}}} \Leftrightarrow c/f_{\text{Object}} = \frac{c - v}{f_{\text{Source}}} \\ \Leftrightarrow f_{\text{Object}} &= \frac{f_{\text{Source}} \cdot c}{c - v} = \frac{f_{\text{Source}}}{1 - v/c} \end{aligned} \quad (\text{A.60})$$

If the object itself is viewed as a light source that emits light with the frequency  $f_{\text{Object}}$  (as defined in equation A.60), the resulting frequency of the light arriving at the sensor can be calculated using the following equation for the Doppler effect with an approaching receiver (the WLI) and a stationary source (the object); equation adapted from [ibidem, p. 474]:

$$\lambda_{\text{Source}} = \frac{c + v}{f_{\text{Receiver}}} \quad (\text{A.61})$$

It follows from equation A.61 and  $\lambda = c/f$ :

$$\begin{aligned}\lambda_{\text{Object}} &= \frac{c+v}{f_{\text{Sensor}}} \\ \Leftrightarrow f_{\text{Sensor}} &= \frac{c+v}{\lambda_{\text{Object}}} = \frac{(c+v) \cdot f_{\text{Object}}}{c} = (1+v/c) \cdot f_{\text{Object}}\end{aligned}\quad (\text{A.62})$$

If equation A.60 is inserted for  $f_{\text{Object}}$  in equation A.62, the resulting expression is:

$$\begin{aligned}f_{\text{Sensor}} &= (1+v/c) \cdot \frac{f_{\text{Source}}}{1-v/c} = f_{\text{Source}} \cdot \frac{1+v/c}{1-v/c} = f_{\text{Source}} \cdot \frac{1+v/c}{1-v/c} \cdot \frac{1+v/c}{1+v/c} \\ f_{\text{Sensor}} &= f_{\text{Source}} \cdot \frac{1+2(v/c) + (v/c)^2}{1-(v/c)^2}\end{aligned}\quad (\text{A.63})$$

Since  $c \gg v$ , it follows that  $(v/c)^2 \approx 0$ . With that approximation, equation A.63 can be simplified:

$$\begin{aligned}f_{\text{Sensor}} &= f_{\text{Source}} \cdot \frac{1+2v/c + (v/c)^2}{1-(v/c)^2} \approx f_{\text{Source}} \cdot \frac{1+2v/c+0}{1-0} \\ f_{\text{Sensor}} &\approx f_{\text{Source}} + 2v/c \cdot f_{\text{Source}}\end{aligned}\quad (\text{A.64})$$

The Doppler frequency  $f_D$  can be determined using equation A.64:

$$f_D = 2v/c \cdot f_{\text{Source}} \quad (\text{A.65})$$

With  $\lambda = c/f \Leftrightarrow f/c = 1/\lambda$  and  $f_{\text{Source}} = f_0$ , it follows from equation A.65 that:

$$f_D = 2v \cdot \frac{f_0}{c} = 2 \cdot \frac{v}{\lambda_0} \quad (\text{A.66})$$

Thus, the resulting frequency of the received light that was sent toward the object during the measurement movement has the following frequency shift added to its initial frequency:

$$f_{\text{Measurement}} = f_0 + f_D = f_0 + 2v \cdot \frac{f_0}{c} = f_0 + 2 \cdot \frac{v}{\lambda_0} = \frac{c}{\lambda_0} + 2 \cdot \frac{v}{\lambda_0} \quad (\text{A.67})$$

Since there is no relative movement between the beam splitter and the reference mirror, the Doppler effect does *not* affect the reference channel. Hence, the frequency in that path remains the same:

$$f_{\text{Reference}} = f_0 \quad (\text{A.68})$$

When the measurement and reference light waves are superimposed on the camera chip, two waves of slightly different frequencies are added and integrated by the pixel. The effect of such a summation is derived for a mono-frequent signal:

$$g(t) = \cos(2\pi f_0 t) + \cos(2\pi(f_0 + f_D)t) \quad (\text{A.69})$$

The trigonometric relationship in A.69 can also be expressed as a product, using the following relationship taken from [Zei+12, p. 61]:

$$\cos x + \cos y = 2 \cos \left( \frac{x+y}{2} \right) \cos \left( \frac{x-y}{2} \right) \quad (\text{A.70})$$

Thus, with equation A.70 applied to equation A.69, it follows that:

$$\begin{aligned} g(t) &= 2 \cos \left( \frac{2\pi t(f_0 + f_0 + f_D)}{2} \right) \cos \left( \frac{2\pi t(f_0 - f_0 - f_D)}{2} \right) \\ &= 2 \cos \left( \frac{2\pi t(2f_0 + f_D)}{2} \right) \cos \left( \frac{2\pi t f_D}{2} \right) \\ &= 2 \cos \left( 2\pi t \left( f_0 + \frac{f_D}{2} \right) \right) \cos \left( 2\pi t \frac{f_D}{2} \right) \end{aligned} \quad (\text{A.71})$$

As evident in equation A.71, the result is an amplitude-modulated signal with a carrier frequency of  $f_C = f_0 + f_D/2$  and an envelope with a frequency of  $f_D/2$ . This is illustrated in Figure A.9. With

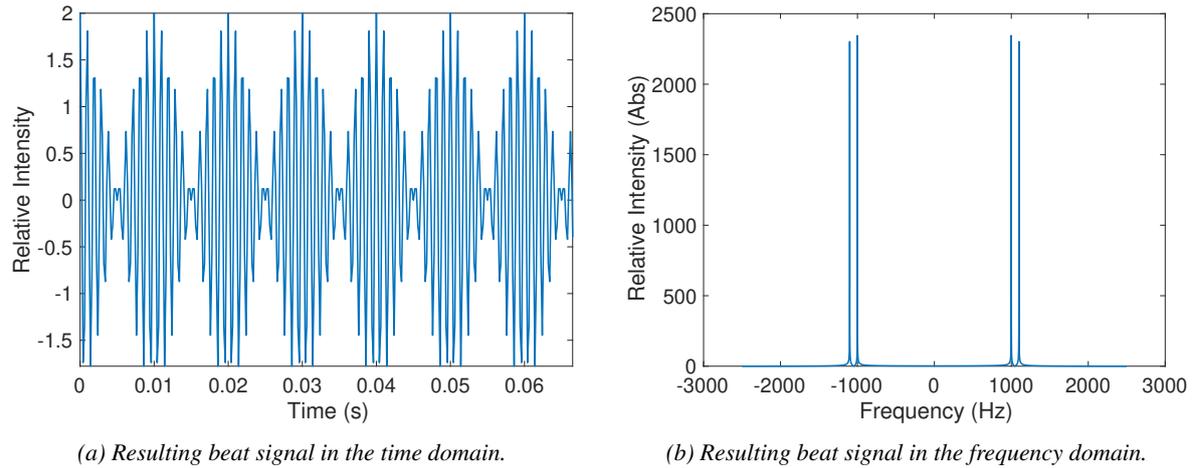


Figure A.9: Resulting amplitude modulation after two harmonic functions with different frequencies were added. Simulation parameters:  $f_0 = 1$  kHz,  $f_D = 100$  Hz, sampling frequency  $f_S = 5f_0$ , simulation time  $T = 100/f_D$ .

an LED central wavelength of  $\lambda_0 = 648$  nm, the resulting central frequency of  $f_0 = 462.963 \cdot 10^{12}$  Hz is significantly larger than, for instance, the Doppler frequency  $f_D = 30.864$  kHz that would result at a velocity of  $v = 10$  mm/s. Because of this, the beat signal is difficult to simulate. Therefore, alternative frequencies were chosen for the visualization, as described in Figure A.9's caption:  $f_0 = 1$  kHz and  $f_D = 100$  Hz, meaning that  $f_C = 1050$  Hz. As expected, the frequency bins in the two-sided spectrum shown in Figure A.9b are at  $\pm(f_C \pm f_D/2)$  (amplitude modulation), which is to say  $\pm 1000$  Hz and  $\pm 1100$  Hz. It should be noted that not the entire time signal that was mapped into the frequency domain is shown in Figure A.9b. The longer simulation time was chosen to allow for more frequency bins for the fast Fourier transform (FFT) that was applied to the time-domain signal

in order to obtain a frequency-domain signal. The time signal was then cut off for the figure, to allow for better visualization of the beat signal. Lastly, it should be mentioned that the expected frequency domain signal in continuous time would be Dirac pulses instead of the washed-out signal that resulted from the discrete-time simulation.

As mentioned before, the E-field is squared in order to yield the intensities, which can finally be detected by the imaging sensor. For the following considerations, two variables are defined:

$$\alpha = 2\pi t(f_0 + f_D/2) \quad (\text{A.72a})$$

$$\beta = 2\pi t f_D/2 \quad (\text{A.72b})$$

If the squaring operation is applied to the signal described in A.71, the resulting signal using equations A.72 is:

$$g(t)^2 = (2 \cos(\alpha) \cos(\beta))^2 = 4(\cos(\alpha) \cdot \cos(\alpha)) \cdot (\cos(\beta) \cdot \cos(\beta)) \quad (\text{A.73})$$

The multiplied parentheses in equation A.73 can also be expressed as sums, using the following relationship taken from [Zei+12, p. 61]:

$$\cos x \cos y = \frac{1}{2} (\cos(x - y) + \cos(x + y)) \quad (\text{A.74})$$

With equation A.74 applied to equation A.73, it follows that:

$$\begin{aligned} g(t)^2 &= 4 \cdot (\cos(\alpha) \cdot \cos(\alpha)) \cdot (\cos(\beta) \cdot \cos(\beta)) \\ &= 4 \cdot \left( \frac{1}{2} (\cos(2\alpha) + 1) \cdot \frac{1}{2} (\cos(2\beta) + 1) \right) \end{aligned} \quad (\text{A.75})$$

$$= (\cos(2\alpha) + 1) \cdot (\cos(2\beta) + 1) \quad (\text{A.76})$$

$$= (\cos(2\alpha) \cos(2\beta)) + \cos(2\alpha) + \cos(2\beta) + 1 \quad (\text{A.77})$$

Finally, the product in parentheses in equation A.77 can be expressed as a sum with equation A.74; hence, it follows for equation A.77 that:

$$g(t)^2 = \frac{1}{2} [\cos(2(\alpha + \beta)) + \cos(2(\alpha - \beta))] + \cos(2\alpha) + \cos(2\beta) + 1 \quad (\text{A.78})$$

Therefore, the expected frequency components are  $f_1 = \pm 2(f_\alpha + f_\beta)$ ,  $f_2 = \pm 2(f_\alpha - f_\beta)$ ,  $f_3 = \pm 2f_\alpha$ ,  $f_4 = \pm 2f_\beta$  and  $f_5 = 0$ . With the definitions for  $\alpha$  and  $\beta$  in equations A.72, it follows that the frequency components are  $f_1 = \pm 2(f_0 + f_D/2 + f_D/2) = \pm 2(f_0 + f_D)$ ,  $f_2 = \pm 2f_0$ ,  $f_3 = \pm 2(f_0 + f_D/2) = \pm(2f_0 + f_D)$ ,  $f_4 = \pm 2(f_D/2) = \pm f_D$  and  $f_5 = 0$ . With these re-substitutions,

the resulting signal is:

$$g(t)^2 = \frac{1}{2}(\cos(2\pi \cdot 2(f_0 + f_D/2 + f_D/2)t)) + \cos(2\pi(2f_0)t) + \cos(2\pi(2(f_0 + f_D/2))t) + \cos(2\pi f_D t) + 1 \quad (\text{A.79})$$

The fifth frequency (of 0 Hz, due to the constant 1) in equation A.79 is expected, because the squaring

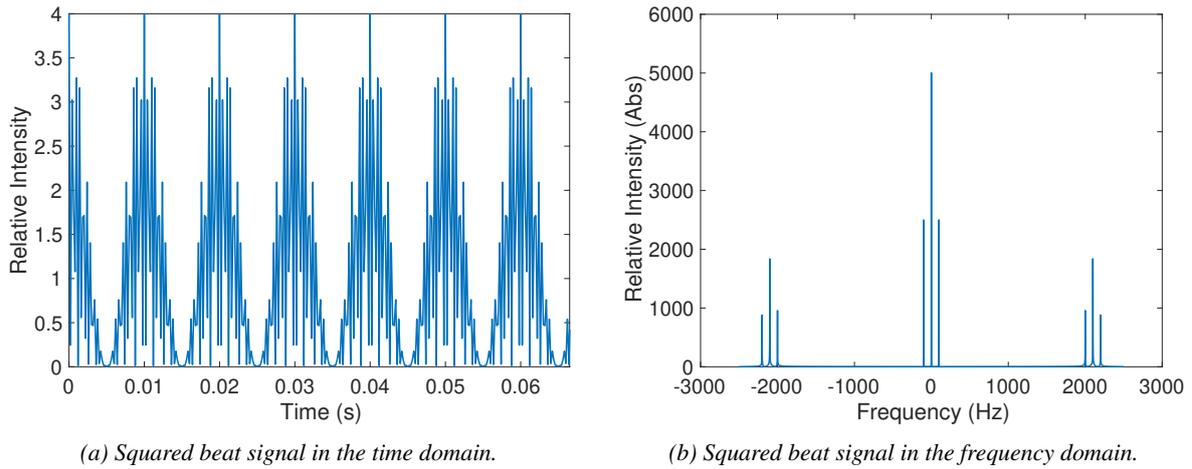


Figure A.10: Illustration of the resulting squared beat signal. Simulation parameters:  $f_0 = 1$  kHz,  $f_D = 100$  Hz, sampling frequency  $f_S = 5f_0$ , simulation time  $T = 100/f_D$ .

operation turns the negative signal components positive. The signal therefore has to contain a DC component. Besides the DC component, only one other low-frequency component is contained in the resulting signal, namely the Doppler frequency  $f_D$ . The resulting simulated signal is shown in Figure A.10 in both the time and frequency domains; the same limitations as for Figure A.9 apply.

All real devices have a bandpass characteristic, and current imaging technology can by far not record the higher frequency contents. The combination of squaring the signal, followed by the frequency cutoff due to the imager's bandpass characteristic, therefore leads to a demodulation of the signal in equation A.71 (the sum of the Doppler-shifted and the unaffected signal). Hence, the central frequency  $f_0$  can effectively be replaced with the Doppler frequency  $f_D$  [Bee+04, p. 4].

However, not only the central frequency is subject to the Doppler effect, but all spectral components are affected. Figure A.11 illustrates the influence of the Doppler effect on the spectral bandwidth. If a spectrum contains three spectral components, namely a central wavelength plus two components  $f_+$  and  $f_-$  that are symmetrical about the central wavelength  $f_0$  with a distance of  $\pm\Delta$ , that spectrum has the bandwidth  $BW_0$ . If the spectrum is shifted due to the Doppler effect (equation A.66),

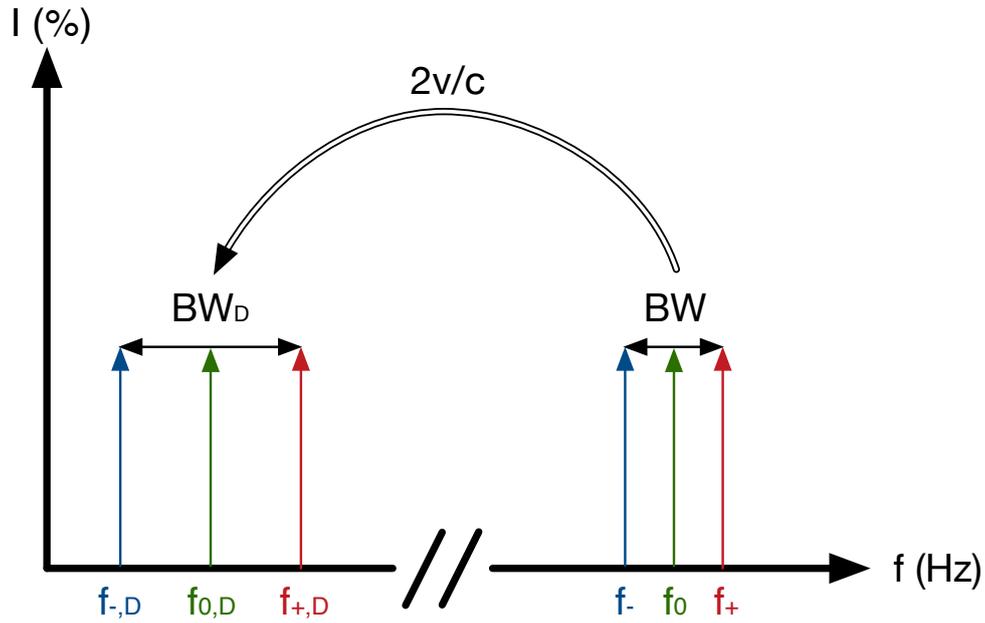


Figure A.11: Spectral broadening due to the Doppler effect.

the resulting components would be:

$$\begin{aligned}
 f_{-,D} &= f_- \cdot \frac{2v}{c} = (f_0 - \Delta f) \cdot \frac{2v}{c} \\
 f_D &= f_0 \cdot \frac{2v}{c} \\
 f_{+,D} &= f_+ \cdot \frac{2v}{c} = (f_0 + \Delta f) \cdot \frac{2v}{c}
 \end{aligned}$$

Therefore, the resulting Doppler bandwidth  $BW_D = 2\Delta f_D$  can be calculated:

$$\begin{aligned}
 BW_D &= 2 \cdot \Delta f_D \\
 &= f_{+,D} - f_{-,D} \\
 &= (f_0 + \Delta f) \cdot \frac{2v}{c} - (f_0 - \Delta f) \cdot \frac{2v}{c} \\
 &= 2 \cdot \Delta f \cdot \frac{2v}{c}
 \end{aligned} \tag{A.80}$$

$$\Leftrightarrow \Delta f = \Delta f_D \cdot \frac{c}{2v} \tag{A.81}$$

The coherence length in equation A.53 can thus be expressed in terms of  $\Delta f_D$  with equation A.81:

$$\begin{aligned} L_{c,\text{FWHM}} &= \frac{4 \cdot c \cdot \ln(2)}{\pi \cdot \Delta f} \\ &\Leftrightarrow = \frac{4 \cdot c \cdot \ln(2)}{\pi \cdot \frac{\Delta f_D \cdot c}{2v}} \\ &= \frac{8 \cdot v \cdot \ln(2)}{\pi \cdot \Delta f_D} \end{aligned} \quad (\text{A.82})$$

If the replacements  $\Delta f \rightarrow \Delta f_D$  and  $f_0 \rightarrow f_D$  are performed, the observed intensity due to lateral scanning can be expressed in the time domain:

$$I_D(\tau) = \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{\pi \tau \cdot \Delta f_D}{2\sqrt{\ln(2)}}\right)^2} \cdot \cos(2\pi f_D \tau) \right) \quad (\text{A.83})$$

$$\begin{aligned} &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{\pi \tau \cdot \frac{8 \cdot v \cdot \ln(2)}{\pi \cdot L_{c,\text{FWHM}}}}{2\sqrt{\ln(2)}}\right)^2} \cdot \cos(2\pi f_D \tau) \right) \\ &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{4\tau \cdot v \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos(2\pi f_D \tau) \right) \end{aligned} \quad (\text{A.84})$$

In the stationary case, the only motion is performed by the photons themselves, which means that the speed is the speed of light  $c$ . Thus, the distance is related to the time it takes the light to travel (at the speed of light) to the observed object and back, at a given displacement from the initial position ( $t = 2(z - z_0)/c$ ). This led to equation A.58 from equation A.57. In the case of constant velocity scanning, the time-base can be derived from the displacement that has been performed with a constant speed  $v$ :  $t = (z - z_0)/v$ . Using that expression for the time-base in equation A.84 results in an expression describing heterodyne scanning due to movement at constant velocity:

$$\begin{aligned} I_D(\tau) &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{4\tau \cdot v \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos(2\pi f_D \tau) \right) \\ \Leftrightarrow I_D(z) &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{4 \cdot \frac{z - z_0}{v} \cdot v \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi f_D \cdot \frac{z - z_0}{v}\right) \right) \\ &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{4 \cdot (z - z_0) \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi \frac{f_D}{v} \cdot (z - z_0)\right) \right) \end{aligned} \quad (\text{A.85})$$

Finally, with equation A.66, the resulting correlogram can also be expressed in terms of wavelengths:

$$\begin{aligned}
 I_D(z) &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{4 \cdot (z-z_0) \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi \frac{f_D}{v} \cdot (z-z_0)\right) \right) \\
 &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{4 \cdot (z-z_0) \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi \cdot \frac{2v}{\lambda_0 \cdot v} \cdot (z-z_0)\right) \right) \\
 &= \hat{I}_0 \cdot \left( 1 + e^{-\left(\frac{4 \cdot (z-z_0) \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi \frac{2}{\lambda_0} \cdot (z-z_0)\right) \right) \quad (\text{A.86})
 \end{aligned}$$

### A.3.5 Further Considerations

Several points should be noted with respect to how closely the derived model matches nature. First, the description of an LED's spectrum as a normally distributed function is only a model in and of itself. If the actual LED's spectrum should differ significantly from a Gaussian, the resulting autocorrelation function will no longer follow the derived model.

Second, the actual observed modulation function may also contain a potentially time-varying phase term  $\varphi(t)$  and measurement noise. However, as will be explained in the next section, the way the information is extracted from the raw data ensures measurements with very high precision nonetheless.

Third, the model derivation assumed full reflectivity in both the reference and the measurement paths in order to keep the derivation more straightforward. However, unless the interferometer is used to measure a high-quality mirror, the measured object's reflectivity will usually differ from the reference mirror's. There is a DC offset  $\hat{I}_0 = I_0 + \tilde{I}_0$  in equation A.23, which was defined as  $\hat{I}_0 = I_0 + \tilde{I}_0 = 2 \cdot I_0$  for the special case of equal reflectivities. In reality, the resulting offset depends on the signal fractions of the reference mirror ( $s_r$  and object  $s_o$ ):

$$\hat{I}_0 = I_0 + \tilde{I}_0 = I_0 \cdot s_r^2 + I_0 \cdot s_o^2 = I_0 \cdot (s_r^2 + s_o^2) \quad (\text{A.87})$$

The refraction factors in equation A.87 are squared because intensities, not electric fields, are observed. Furthermore, the scaling factor of the autocorrelation function in equation A.23 was defined as  $2I_0$ . However, if different refractions are considered, the resulting intensity can be determined as follows:

$$I_c = (s_r \cdot E_0) \cdot (s_{object} \cdot E_0^*) = (s_r \cdot s_o) \cdot (E_0 \cdot E_0^*) = (s_r \cdot s_o) \cdot |E_0|^2 = (s_r \cdot s_o) \cdot I_0 \quad (\text{A.88})$$

Thus, with the phase and refractions considered, a more complete model can be derived, based on

equation A.83:

$$\begin{aligned}
 I_D(\tau) &= (I_0 \cdot s_r^2 + I_0 \cdot s_o^2) + 2(s_r \cdot s_o) \cdot I_0 \cdot e^{-\left(\frac{\pi\tau \cdot \Delta f_D}{2\sqrt{\ln(2)}}\right)^2} \cdot \cos(2\pi f_D \tau + \varphi\tau) \\
 &= I_0 \cdot (s_r^2 + s_o^2) \cdot \left(1 + \frac{2s_r \cdot s_o}{s_r^2 + s_o^2} \cdot e^{-\left(\frac{\pi\tau \cdot \Delta f_D}{2\sqrt{\ln(2)}}\right)^2} \cdot \cos(2\pi f_D \tau + \varphi(\tau))\right) \quad (\text{A.89})
 \end{aligned}$$

The factor of two comes from equation A.23.

For instance, if the reference mirror has a reflectivity of 99% and the object has a reflectivity of 10%, the resulting DC offset has an intensity of  $I_{DC} = I_0 \cdot (0.99^2 + 0.1^2) \approx 0.99 \cdot I_0$ . The resulting correlogram intensity would be  $I_{\text{correlogram}} = 2I_0 \cdot 0.99 \cdot 0.1 \approx 0.198 \cdot I_0$ . The more complete model in equation A.89 can be similarly derived for the other relevant expressions:

$$I_D(\tau) = I_0 \cdot (s_r^2 + s_o^2) \cdot \left(1 + \frac{2s_r \cdot s_o}{s_r^2 + s_o^2} \cdot e^{-\left(\frac{4\tau \cdot v \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos(2\pi f_D \tau + \varphi(\tau))\right) \quad (\text{A.90})$$

$$I_D(z) = I_0 \cdot (s_r^2 + s_o^2) \cdot \left(1 + \frac{2s_r \cdot s_o}{s_r^2 + s_o^2} \cdot e^{-\left(\frac{4 \cdot (z - z_0) \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi \frac{2}{\lambda_0} \cdot (z - z_0) + \tilde{\varphi}(z)\right)\right) \quad (\text{A.91})$$

$$I_D(z) = I_0 \cdot (s_r^2 + s_o^2) \cdot \left(1 + \frac{2s_r \cdot s_o}{s_r^2 + s_o^2} \cdot e^{-\left(\frac{4 \cdot (z - z_0) \cdot \sqrt{\ln(2)}}{L_{c,\text{FWHM}}}\right)^2} \cdot \cos\left(2\pi \frac{f_D}{v} \cdot (z - z_0) + \tilde{\varphi}(z)\right)\right) \quad (\text{A.92})$$

Even these more complete descriptions do not contain all additional factors that could influence the observed signal. Examples are thermal influences and beam-splitter losses. However, it does explain the observed phenomena, especially if the interferometer is of high quality and measurements are performed under reasonable conditions.

## A.4 Measurement Method

In the previous section, the theoretical basis for understanding the physics of white light interferometers, as used in this dissertation, was derived. In this section, the measurement method based on white light interferometry is described in detail.

There are three signal components in equation A.90:

1. A DC offset
2. A Gaussian envelope
3. A carrier frequency  $f_D$

All those components can be used to detect different attributes of the measured object.

The intensity of the DC offset depends on the reflectivities of both the reference mirror and the measured object. Since the reference mirror has a constant reflectivity, the measured DC offset is an indicator of the reflectivity of the object.

The carrier has a frequency related to the speed between the interferometer and the measured object  $f_D = \frac{2v}{\lambda_0}$  (see equation A.66). Because the central wavelength is known and constant, a change in the observed demodulation frequency relates directly to a change in speed. Since the reference mirror is stationary, this means that the speed of the object can be determined with  $v = \frac{1}{2} \cdot f_D \cdot \lambda_0$ . For instance, if the central wavelength was  $\lambda_0 = 648$  nm and the observed demodulation frequency was  $f_D = 40$  kHz, this would relate to an object speed of 12.96 mm/s.

The Gaussian envelope of the correlation term in the distance domain contains the term  $(z - z_0)$ , as shown in equation A.92. Because all other factors are constant with respect to  $z$ , the envelope has its maximum at  $z = z_0$ , which is precisely the position at which the distances in both interferometer arms are equal. This means that if the Gaussian envelope is sampled with sufficiently many bins ( $f_s \geq f_N = 2 \cdot f_{max}$ ), the underlying function can be recovered. Its maximum corresponds to the point in space at which the distance to the reference mirror equals the distance to the measured surface.

The key aspect for coherence radar measurements is thus the maximum detection of the Gaussian envelope; this yields a highly accurate distance measurement in a coordinate system having its origin in the point of equal arm lengths  $z_0$ . However, since the envelope is amplitude-modulated, it changes rapidly, with the frequency  $f_D$ , which makes it challenging to detect the absolute maximum of the function. Therefore, in a first step, an amplitude demodulation is performed. There are several relevant technical implementations of amplitude demodulation [OL10, pp. 360]. A general signal  $s(t)$  containing an amplitude-modulated component may consist of a DC component, low-frequency components, very-high-frequency components, unwanted signal components in the frequency range of the amplitude modulated signal component, and the actual amplitude modulated signal component. All but the last of these components need to be removed from the signal as much as possible before any further processing. The frequencies of the unwanted signal components should be sufficiently far apart from the signal of interest to allow for the implementation of a technically feasible bandpass filter. Assuming that there are no parasitic signal components in the range of the amplitude modulated signal component, the bandpass filtered signal has the form  $s_{BP}(t) = g(t) \cdot \cos(2\pi \cdot f_D \cdot t + \varphi(t))$ . Here,  $g(t)$  is the original signal, in this case a Gaussian, and  $\cos(2\pi \cdot f_D \cdot t + \varphi(t))$  is the carrier.

The simplest way to demodulate such a signal  $s_{BP}(t)$  is to take the absolute value of the signal, followed by a low pass filter to remove the remaining high-frequency components:  $|g(t)| = 2 \cdot LP(|BP(s(t))|) = 2 \cdot LP(|s_{BP}(t)|)$ . This can easily be shown in the time domain using equation

A.74:

$$\begin{aligned}
 2 \cdot BP(LP(|s(t)|)) &= 2 \cdot LP(|s_{BP}(t)|) \\
 &= 2 \cdot LP(\sqrt{(g(t) \cdot \cos(2\pi \cdot f_D \cdot t + \varphi(t)))^2}) \\
 &= 2 \cdot |g(t)| \cdot LP\left(\sqrt{\cos(2\pi \cdot f_D \cdot t + \varphi(t)) \cdot \cos(2\pi \cdot f_D \cdot t + \varphi(t))}\right) \\
 &= 2 \cdot |g(t)| \cdot LP\left(\frac{1}{2} \cdot (\cos(0) + \cos(2\pi \cdot (2 \cdot f_D) + 2 \cdot \varphi(t)))\right) \\
 &= |g(t)|
 \end{aligned}$$

This simple design has the drawback of the bandpass being tuned to a specific carrier frequency. A demodulation scheme that removes this limitation is mixing the signal down to an intermediate frequency  $f_i = f_D - f_m$ , using a tunable oscillator with frequency  $f_m$ . The resulting signal  $s_i(t)$  is then fed into a modulator that is hardwired to the intermediate frequency  $f_i$  [OL10, pp. 360]. The idea is to select  $f_m$  such that the resulting intermediate frequency matches the fixed demodulator design using  $f_i = f_D - f_m$ . This can be shown in the time domain using equation A.74:

$$\begin{aligned}
 \cos(2\pi \cdot f_D \cdot t) \cdot \cos(2\pi \cdot f_m \cdot t) &= \frac{1}{2} \cdot (\cos(2\pi \cdot (f_D - f_m) \cdot t) + \cos(2\pi \cdot (f_D + f_m) \cdot t)) \\
 &= \frac{1}{2} \cdot (\cos(2\pi \cdot (f_i) \cdot t)) + \frac{1}{2} \cdot (\cos(2\pi \cdot (f_D + f_m) \cdot t))
 \end{aligned}$$

Thus, the mixing process with  $f_m$  leads to a shift to the intermediate frequency  $f_i$  and a high-frequency component at  $f_D + f_m$ .

Both of the mentioned methods only yield the absolute value of the envelope. If it is desirable to consider the phase relationship too, a quadrature receiver has to be used. Quadrature receivers use an orthogonal base to get a vector representation of the signal in the complex plane. The amplitude can then be retrieved by applying the 2-norm  $|\cdot|_2$  and the phase by applying the  $\text{atan2}(\cdot)$  function to the vector. The orthogonal base is created by splitting the signal and mixing one of the signal paths with an in-phase (cosine) signal and the other with an out-of-phase ( $\pm$ sine) signal [ibidem, pp. 360]. Because this second component is shifted by  $\pm 90^\circ$ , it is known as the quadrature component, which is how the demodulator derives its name. Because of the in-phase (I) and quadrature (Q) channels, it is also known as an IQ demodulator. The demodulation (mixing) signal is generated by a local oscillator set to the presumed carrier frequency. The principle of quadrature demodulation is illustrated in Figure A.12.

Quadrature demodulation yields the real and imaginary part of a complex number representing the amplitude and phase of an amplitude-modulated signal  $s(t) = 2 \cdot g(t) \cdot \cos(2\pi f_D t + \varphi)$  with the envelope  $g(t)$ ; the modulation frequency  $f_D$  and the phase  $\varphi$  is assumed. For the real part channel, a cosine with frequency  $f_D$  is mixed with the input signal:

$$s(t) \cdot \cos(2\pi f_D t) = 2 \cdot g(t) \cdot \cos(2\pi f_D t + \varphi) \cdot \cos(2\pi f_D t)$$

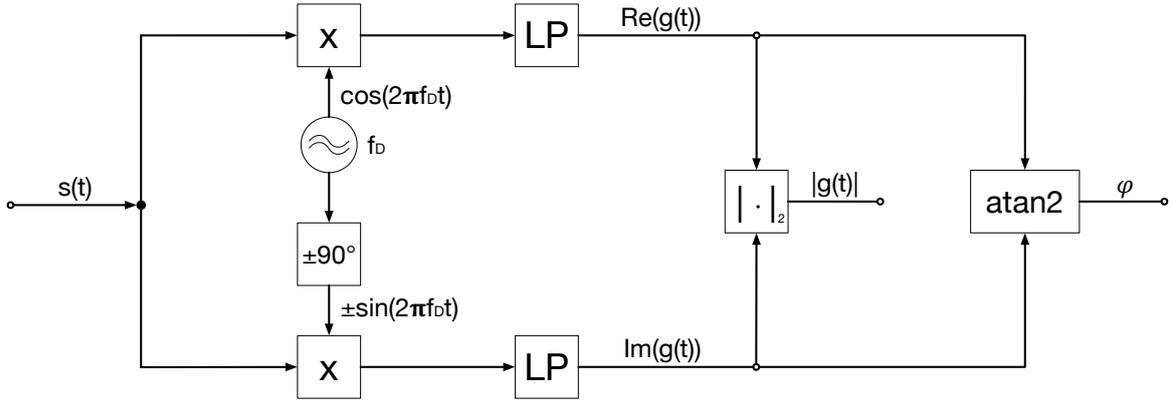


Figure A.12: Principle of quadrature demodulation.

It follows from Figure A.12 with equation A.74:

$$\begin{aligned}
 s_r(t) &= LP(2 \cdot g(t) \cdot \cos(2\pi f_D t + \varphi) \cdot \cos(2\pi f_D t)) \\
 &= g(t) \cdot LP(\cos(2\pi f_D t + \varphi - 2\pi f_D t) + \cos(2\pi f_D t + \varphi + 2\pi f_D t)) \\
 &= g(t) \cdot LP(\cos(\varphi) + \cos(2\pi(2f_D)t)) \\
 &= g(t) \cdot \cos(\varphi)
 \end{aligned} \tag{A.93}$$

As can be seen in equation A.93, with just one channel the envelope is simply scaled with the cosine of the phase. This also implies that at a constant  $\pm 90^\circ$  phase shift, the detected amplitude in the real-part channel is a constant zero.

Assuming a  $+90^\circ$  phase shift, and using  $\sin(\alpha) = \cos(\pi/2 - \alpha)$ , it follows from symmetry that  $\cos(\pi/2 + \alpha) = \sin(-\alpha) = -\sin(\alpha)$ . Thus, for the imaginary part's channel, a negative sine with frequency  $f_D$  is mixed with the input signal:

$$s(t) \cdot (-\sin(2\pi f_D t)) = -2 \cdot g(t) \cdot \cos(2\pi f_D t + \varphi) \cdot \sin(2\pi f_D t)$$

It follows from Figure A.12 with equation A.74:

$$\begin{aligned}
 s_i(t) &= LP(-2 \cdot g(t) \cdot \cos(2\pi f_D t + \varphi) \cdot \sin(2\pi f_D t)) \\
 &= LP(2 \cdot g(t) \cdot \cos(2\pi f_D t + \varphi) \cdot \cos(2\pi f_D t + \pi/2)) \\
 &= g(t) \cdot LP(\cos(2\pi f_D t + \varphi - 2\pi f_D t - \pi/2) + \cos(2\pi f_D t + \varphi + 2\pi f_D t + \pi/2)) \\
 &= g(t) \cdot LP(\cos(\varphi - \pi/2) + \cos(2\pi(2f_D)t + \pi/2)) \\
 &= g(t) \cdot \cos(\varphi - \pi/2) \\
 &= g(t) \cdot \sin(\varphi)
 \end{aligned} \tag{A.94}$$

Therefore, if  $s_r(t)$  and  $s_i(t)$  are considered to be the real and imaginary parts of a complex number,

respectively, the resulting complex representation would be:

$$\begin{aligned}
 s(t) &= s_r(t) + j \cdot s_i(t) \\
 &= g(t) \cdot \cos(\varphi) + j \cdot g(t) \cdot \sin(\varphi) \\
 &= g(t) \cdot e^{j\varphi}
 \end{aligned}
 \tag{A.95}$$

Equation A.95 shows that the result is indeed a complex pointer representation of the magnitude and phase of the incoming signal. This means that the magnitude and phase can be extracted using the 2-norm and the atan2 function, respectively.

With regard to white light interferometers, this means that the envelope and phase can be recovered by performing an IQdemodulation with the doppler frequency  $f_D$ , which is known because the interferometer moves at a known constant velocity (see equation A.66). Figure A.13 shows the resul-

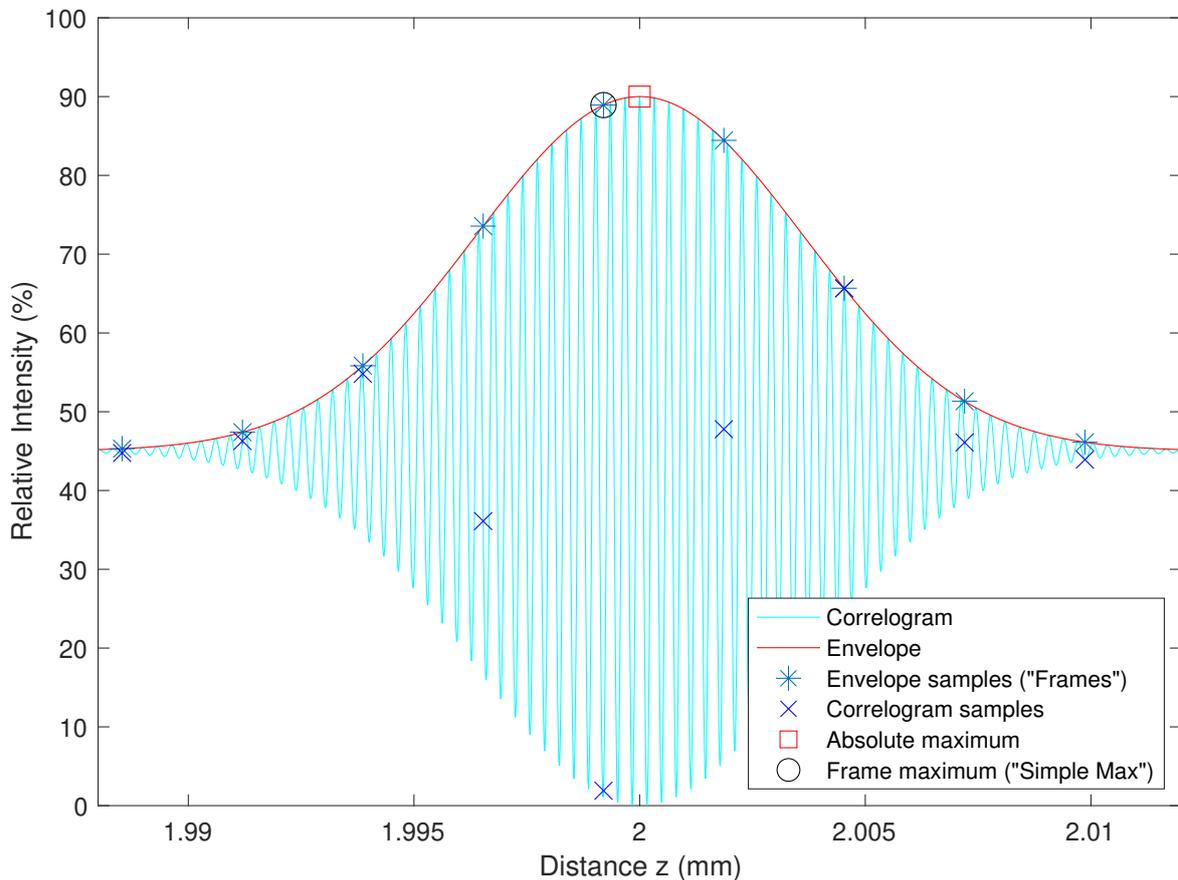


Figure A.13: Correlogram, envelope, and sampled points.

ting correlogram and its IQ-demodulated envelope. Furthermore, samples taken from the correlogram and the envelope in the same locations are indicated, illustrating why it is necessary to perform the demodulation first. Lastly, the absolute maximum, which signifies the distance to the measured object, and the maximum sample ("simple max") are shown. Simply finding the maximum sample is the

fastest option, but the detected distance might be inaccurate by up to half a sample distance in either direction. In the example in Figure A.14d, this would mean that if only the maximum sample was considered, the distance would only be accurate to  $\pm 20 \mu\text{m}$ .

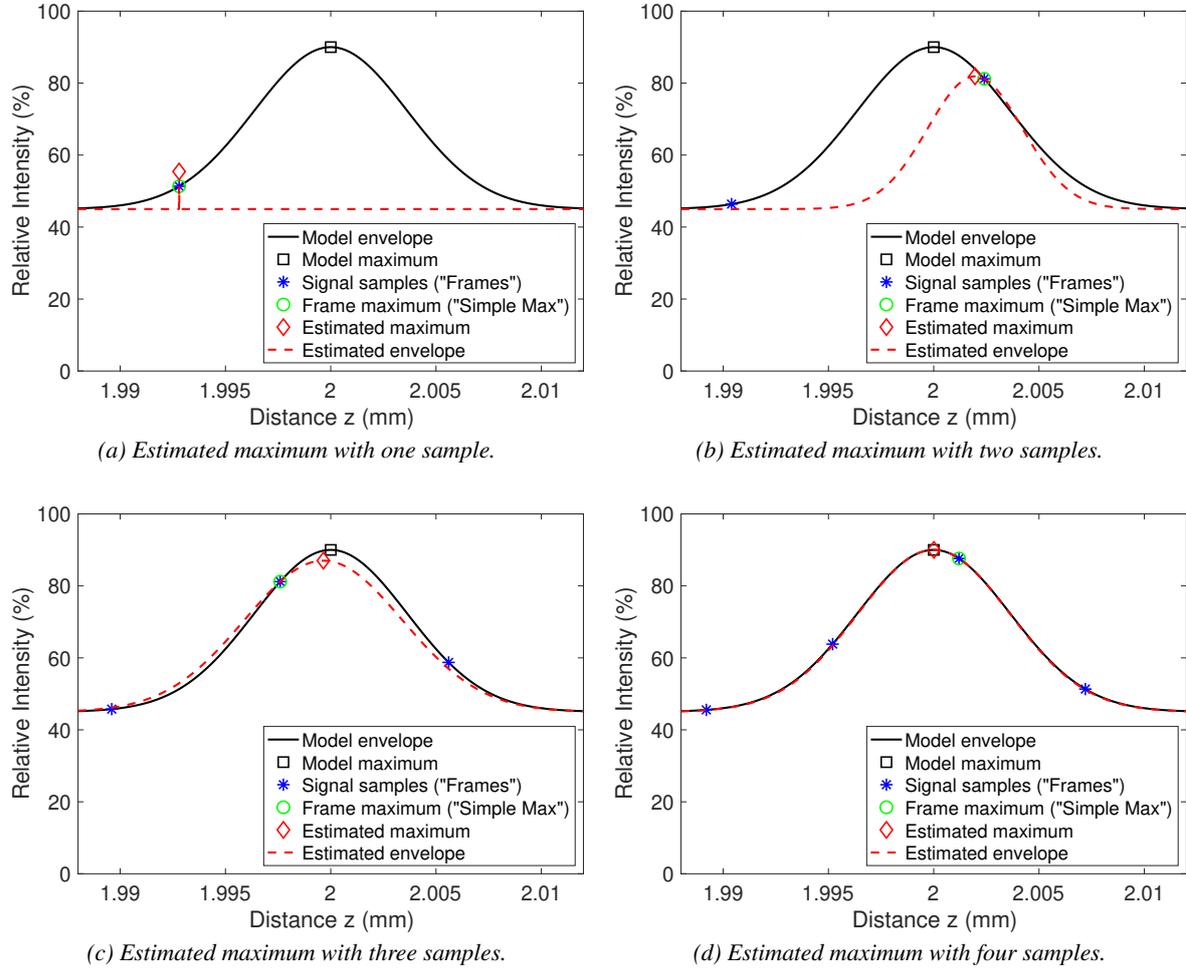


Figure A.14: Illustration of maximum estimation with 1, 2, 3, and 4 samples.

Since the envelope is a Gaussian, the maximum can be determined — with very high accuracy — by estimating the moments of the underlying function from the samples, using a weighted mean approach. The intensity can be viewed as a measure of trust or quality and is inversely related to the standard deviation. Therefore, the variance of each individual positional sample  $z_i$  is given by  $\sigma_i^2 = \hat{I}_i^{-2}$ . Since the ideal estimate of the expected value is given by an inversely variance-weighted

mean (see section 5.3.2), the estimation formula can be expressed in terms of  $z$  and  $I$ :

$$\hat{I}_i = I_i - I_{DC} \quad (\text{A.96})$$

$$\bar{z} = \frac{\sum_{i=1}^n z_i \cdot \sigma_i^{-2}}{\sum_{i=1}^n \sigma_i^{-2}} = \frac{\sum_{i=1}^n z_i \cdot \hat{I}_i^2}{\sum_{i=1}^n \hat{I}_i^2} \quad (\text{A.97})$$

$$\sigma_e^2 = \frac{\sum_{i=1}^n (z_i - \bar{z})^2 \cdot \sigma_i^{-2}}{\sum_{i=1}^n \sigma_i^{-2}} = \frac{\sum_{i=1}^n (z_i - \bar{z})^2 \cdot \hat{I}_i^2}{\sum_{i=1}^n \hat{I}_i^2} \quad (\text{A.98})$$

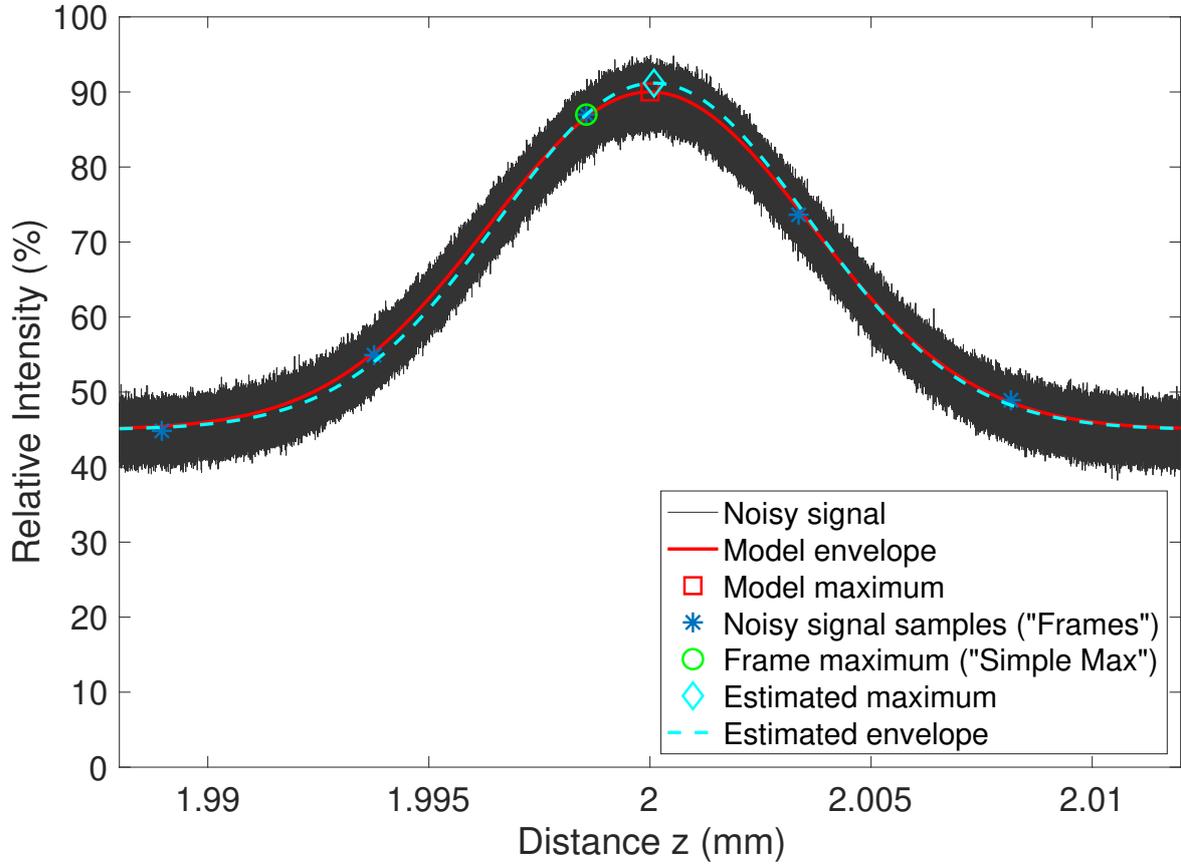


Figure A.15: Maximum detection with five samples and significant noise.

The first and second moments of the Gaussian envelope were estimated through equations A.97 and A.98. Now the maximum intensity, which is associated with the location  $\bar{z}$ , can be found using one of the samples — for instance, the maximum sample  $z_{sm}$ :

$$\begin{aligned} \hat{I}_{max} \cdot e^{-\frac{1}{2} \cdot \frac{(z_{sm} - \bar{z})^2}{(\sigma_e)^2}} &= \hat{I}_{sm} \\ \Leftrightarrow \hat{I}_{max} &= \hat{I}_{sm} \cdot e^{\frac{1}{2} \cdot \frac{(z_{sm} - \bar{z})^2}{\sigma_e^2}} \end{aligned} \quad (\text{A.99})$$

Finally, the actual observable intensity can be found by adding the DC intensity offset of the correlo-

gram (see equation A.96):

$$I_{max} = \hat{I}_{max} + I_{DC} = \hat{I}_{sm} \cdot e^{\frac{1}{2} \cdot \frac{(z_{sm} - \bar{z})^2}{\sigma_e^2}} + I_{DC} \quad (\text{A.100})$$

If the intensity expression in equation A.100 is compared to the intensity coefficients in equation A.92, it can furthermore be derived that:

$$I_{DC} = I_0 \cdot (s_r^2 + s_o^2) \quad (\text{A.101})$$

$$I_{max} = I_0 \cdot 2 \cdot s_r \cdot s_o \quad (\text{A.102})$$

Figure A.14 on page A36 illustrates the influence of the number of samples used for the maximum detection. In the last case of the simulation, with four samples, the absolute error of the detected maximum is approximately 8 nm. Figure A.15 on page A37 shows the distance estimation algorithm with five samples and a significant amount of noise. The simulated error is approximately 150 nm.

## Appendix B

# Inspection Technology Overview Table

The following table, Table B.1, provides an overview of the advantages and disadvantages of the various physical principles and methods of testing. The table was compiled from similar tables in [Uni01, ch. 5, p. 7], [GG06, pp. 372], and a table by L. Cartz, as quoted in [Cam13, p. 12]. The table also reflects the author's research on white light interferometry, as published in [ODH13], [Ott+14], and [DSO14].

*Table B.1: Advantages and disadvantages of the discussed inspection technologies (extended overview based on [Uni01, ch. 5, p. 7], [GG06, pp. 372], and a table by L. Cartz, as quoted in [Cam13, p. 12]).*

Technology	Advantage and Disadvantages
<i>Eddy Current Testing</i>	<ul style="list-style-type: none"> <li>+ Surface and near-surface defects</li> <li>+ Medium speed</li> <li>+ Portable devices possible</li> <li>+ Can be readily automated for simple geometries</li> <li>- Only conductive materials</li> <li>- Results strongly depend on material attributes</li> <li>- Difficult to use with rough surfaces</li> <li>- Surface has to be directly accessible</li> <li>- Resolution and penetration are inversely related</li> <li>- (Not necessarily contactless)</li> </ul>
<i>Magnetic Particle Inspection</i>	<ul style="list-style-type: none"> <li>+ Surface and internal defects</li> <li>+ Comparatively easy to use</li> </ul>

continued ...

...continued

Technology	Advantage and Disadvantages
<i>Dye Penetrant Inspection</i>	<ul style="list-style-type: none"> <li>- Only ferromagnetic materials</li> <li>- Material may have to be demagnetized</li> <li>- Detection quality depends on orientation</li> <li>- Difficult to use with rough surfaces</li> <li>- Surface has to be directly accessible</li> <li>+ Suitable for virtually all materials</li> <li>+ Can detect very small cracks</li> <li>- Difficult to use with rough or porous surfaces</li> <li>- Can only detect open surface defects</li> <li>- Use of a penetrant required</li> <li>- Difficult to automate</li> <li>- Surface has to be directly accessible</li> <li>- Intensive part interaction required</li> </ul>
<i>Radiography</i>	<ul style="list-style-type: none"> <li>+ Object does not have to be directly accessible</li> <li>+ Internal and surface defects can be detected</li> <li>+ True volumetric reconstruction (tomography)</li> <li>+ Suitable for many materials</li> <li>+ Virtually no part preparation required</li> <li>+ Contactless</li> <li>- Detection quality dependent on orientation (except tomography)</li> <li>- Radiation is harmful                             <ul style="list-style-type: none"> <li>- Therefore, special safety measures required</li> <li>- Therefore, acceptance can be low</li> </ul> </li> <li>- Expensive</li> </ul>
<i>Acoustic Inspection</i>	<ul style="list-style-type: none"> <li>+ Internal flaws can be detected</li> <li>+ Can be automated easily for simple geometries</li> </ul>

continued ...

...continued

Technology	Advantage and Disadvantages
<i>Thermography</i>	<ul style="list-style-type: none"> <li>- Difficult to use with rough surfaces</li> <li>- Surface has to be directly accessible</li> <li>- Typically, a couplant has to be used (for the transducer)</li> <li>- Point-wise scanning</li> <li>- Surface contact required (piezo transducer)</li> <li>+ Internal flaws can be detected</li> <li>+ Contactless</li> <li>- Surface has to be heated for active thermography</li> <li>- Surface has to be directly accessible</li> </ul>
<i>Human Inspection (eyes)</i>	<ul style="list-style-type: none"> <li>+ Human intuition and recognition abilities can be used</li> <li>+ Can be readily adopted to different situations</li> <li>+ Contactless</li> <li>+ Relatively high lateral resolutions possible (optical aids)</li> <li>+ Suitable for most materials</li> <li>- Surface has to be directly accessible</li> <li>- Can only detect surface defects and geometric deformations</li> <li>- Dependent on lighting and surface</li> <li>- High degree of subjectivity</li> </ul>
<i>Optical Inspection (cameras)</i>	<ul style="list-style-type: none"> <li>+ Can be readily adopted to different situations</li> <li>+ Contactless</li> <li>+ Relatively high lateral resolutions possible</li> <li>+ Can be automated relatively easily</li> <li>+ Suitable for most materials</li> <li>- Surface has to be directly accessible</li> <li>- Can only detect surface defects and geometric deformations (multi-camera)</li> <li>- 3D digitalization (multi-camera) has limited depth resolution</li> </ul>

continued ...

...continued

Technology	Advantage and Disadvantages
<i>Laser Line Triangulation</i>	<ul style="list-style-type: none"> <li>- Dependent on lighting and surface</li> <li>- Stereo vision: shading effects</li> <li>+ Provides relatively accurate depth measurements</li> <li>+ Can readily be automated</li> <li>+ Contactless</li> <li>+ Robust system</li> <li>+ Can be combined with an actuator to yield a 3D scan</li> <li>- Only a line scan without an actuator</li> <li>- Measurement depth/area and resolution inversely related</li> <li>- Depth resolution limited</li> <li>- Shading effects</li> <li>- Surface has to be directly accessible</li> <li>- Can only detect surface defects and geometric deformations</li> <li>- Results depend heavily on surface state</li> </ul>
<i>White Light Interferometry</i>	<ul style="list-style-type: none"> <li>+ High lateral resolution possible</li> <li>+ Very high depth resolution (sub-<math>\mu\text{m}</math>)</li> <li>+ Measurement depth/area independent of depth resolution</li> <li>+ No shading effects</li> <li>+ Potentially independent of ambient light</li> <li>+ Suitable for virtually all materials</li> <li>+ Results potentially independent of the surface state</li> <li>+ Contactless</li> <li>- Sensitive to movement during measurement</li> <li>- Difficult to automate</li> <li>- Surface needs to be accessible</li> <li>- Can only detect surface defects and geometric deformations</li> </ul>

## Appendix C

# White Light Interferometer Driver Structure

The interferometer consists of an array of optical components (beam splitter, reference mirror, LED, and lenses) as well as a specialized camera and a linear axis. The interferometer that was used in this dissertation consists of a camera with built-in interferometric components and a linear axis made by a different manufacturer.<sup>1</sup> The communication thus requires two independent hardware channels with their own communication methods. Whereas the camera uses a USB interface and comes with a C application programming interface (API), the axis communicates over a text-based TCP/IP protocol.

Since the WLI application *AutoSense* was written in C#, a wrapper for the API's dynamic-link library (DLL) had to be developed using platform invoke and marshaling techniques. Next, a camera class was developed to encapsulate the API functionality into high-level methods and to combine it with the camera's properties — such as register settings (which were encapsulated in a sub-class).

Similarly, a TCP/IP-based communication handler for the linear axis had to be developed, which is in charge of providing a secure way of sending commands, receiving responses and handling errors. The communication handler was then encapsulated in an axis class, providing high-level methods for axis control and giving object-oriented access to the axis's properties.

Finally, a white light interferometer class was developed, which contains even higher-level methods and properties, bundling the functionalities provided by the camera and axis classes. Additionally, the class handles the tasks unique to white light interferometry, such as the profile calculation required for proper axis motion and trigger settings (see chapter 4). A simplified structure of the interferometer driver is shown in a UML class diagram in Figure C.1 on page A44.

---

<sup>1</sup>The author is contractually obligated not to reveal the manufacturers' names.

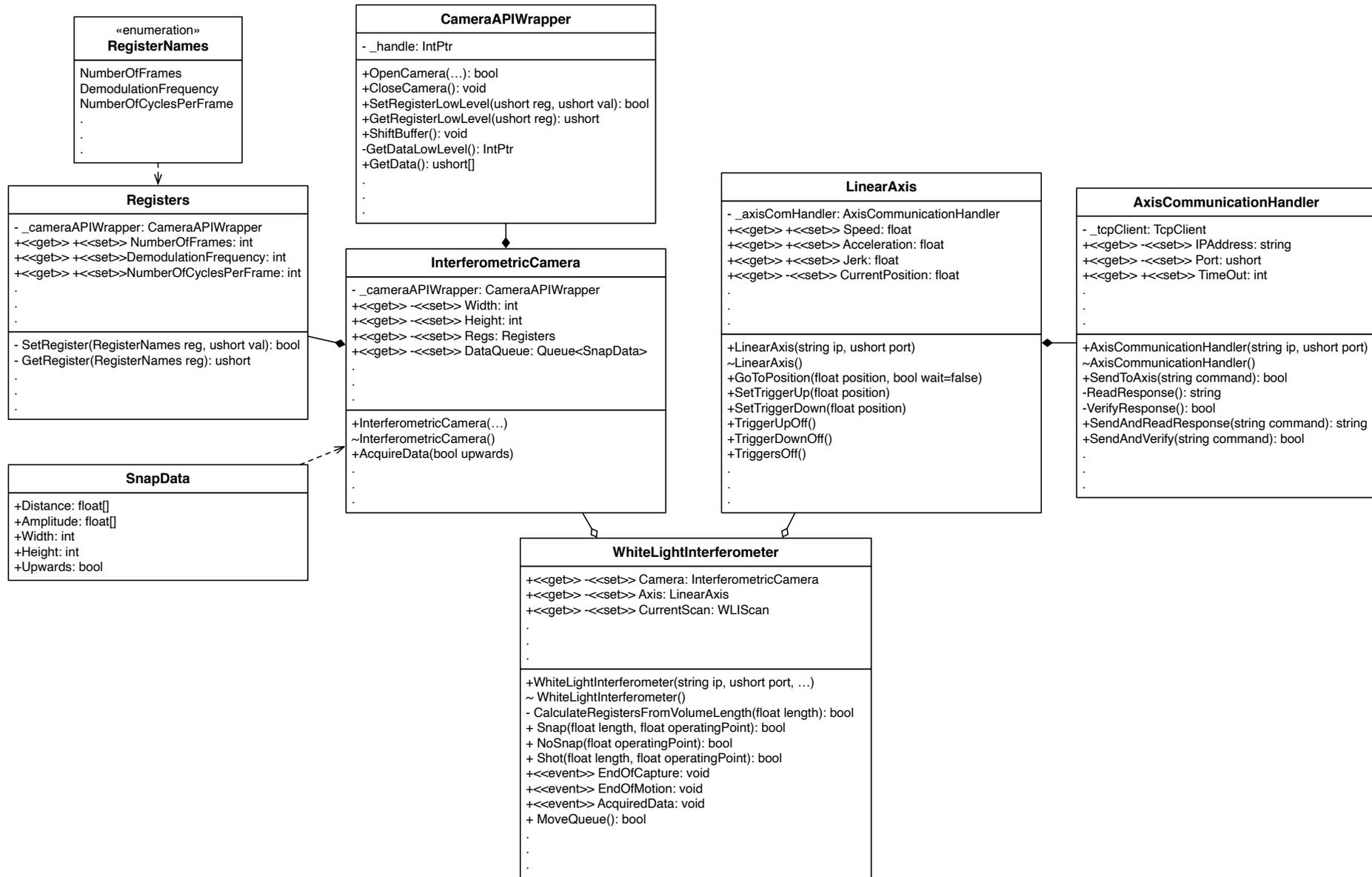


Figure C.1: UML diagram of the white light interferometer driver.

## Appendix D

# Error Handling

During the scanning of a CFM-56 outer liner, a very large number of scans is captured and transmitted between the software packages AutoSense and AutoProcess. Furthermore, there are three software packages running on up to three independent computers, as well as an industrial robot, a rotational axis, and a white light interferometer (which was not designed with a robot-guided process in mind). It is, therefore, not unlikely that an error might occur in at least one of the involved components at some point during the process. Whereas some errors are severe and invariably lead to a termination of the process, others can be handled and recovered from automatically. Because the measurement process takes several hours, a recovery strategy is crucial to improve acceptance of the system in the field. Table D.1 on page A46 lists the possible errors and their severity, together with appropriate measures.

There are two main concerns when it comes to errors. First, data loss is unacceptable because it would lead to uninspected areas, meaning that a critical error in the part might be overlooked. Second, time loss due to errors should be reduced to a minimum. Hence, every error should be treated in a way that guarantees that no loss of data occurs, while simultaneously minimizing the time impact. Thus, for instance, it would be a poor solution to simply restart the entire measurement process. Rather, individual error-handling solutions have to be designed for the various errors. In many cases, the appropriate measures enable automatic recovery of the system (see Table D.1).

### D.1 Software Freeze or Crash

“If debugging is the process of removing software bugs, then programming must be the process of putting them in.”

---

*Edsger Dijkstra*

Even if the greatest care is taken to minimize software bugs, it is virtually impossible to guarantee that complex software is error-free. Bugs have different levels of severity, from slowing down the process

Table D.1: Possible errors during the scanning process.

Error	Possible Result	Measure	Recovery
<i>Robot crash</i>	Hard stop	End of process and error assessment	✗
<i>WLI axis crash</i>	Hard stop	End of process and error assessment	✗
<i>Safety system triggered</i>	Hard stop	Safety reset, user confirmation and resume	✗
<i>Software freeze/crash</i>	Data loss	Watchdog triggers restart, software loads backup, process resumes at backup point	✓
<i>Transmission error AutoSense/AutoProcess</i>	Data loss	Resend data	✓
<i>WLI capture error</i>	Data loss	Reset hard- and software, load backup, repeat scan	✓
<i>Transmission error AutoXIV/AutoSense</i>	Missed scan	Repeat transmission	✓

unnecessarily, to producing incorrect results, or even freezing and crashing the software. Software crashes are arguably not the worst-case scenario in this application, as unnoticed incorrect results can potentially cause much more damage; however, crashes would nevertheless be a serious problem. If a system is to operate autonomously, a mechanism to address freezes and crashes has to be put in place.

Such a measure cannot be part of the software itself, because then it too might be frozen or not even running in the event of a crash. The solution is to use an independent application that monitors whether the main software is still running — a so-called *watchdog timer*. This approach has two components. One is the external software, which contains a timer as its central component. If the timer elapses, it restarts the monitored software, which then needs to take the appropriate steps to resume the previous state. The second component is inside the monitored software and regularly triggers a watchdog timer reset. Thus, if the software completely freezes or crashes, the watchdog timer will not be reset anymore and the software will be restarted.

The watchdog-timer approach assumes that the error that caused the hang-up is of random nature. That is, only certain constellations — such as timings or hardware inputs — are assumed to cause a freeze or crash. Thus, if the software is restarted by the watchdog, the constellation might be different in the next try. However, if the error is systematic, the freeze or crash will happen in the next try as well. Therefore, it is reasonable to count the number of restarts performed by the watchdog, give up after a certain number of tries, and inform the user that there seems to be a serious issue that needs to be addressed.

## D.2 Resume Strategy

If one or more of the software packages crashes or is restarted by the watchdog, it is important to be able to resume the previously started process. In other words, the software packages have to be able to preserve their states. This includes both their mode (state machine) and their data. The data is no longer available in the RAM, which means that the relevant data has to be preserved on the system's hard drive. This mechanism is different for AutoSense (capturing software) and AutoProcess (processing software). AutoSense is solely concerned with the current process and thus only has to keep the data not yet sent to AutoProcess. The state-preservation strategy developed for AutoSense is simply to serialize every scanned ring onto the hard drive, until AutoProcess confirms that it has successfully received and saved them. Then they are deleted from the hard drive because that part of the overall process state is now preserved by AutoProcess. Should AutoSense crash, the IFPT's robot control software and AutoProcess notice that the TCP/IP server has gone offline and they enter their respective waiting-to-reconnect states. Whenever AutoSense is started, it checks whether its autosave folder contains files. If not, there is no previous state to restore and the software will start up normally. Otherwise, the presence of files indicates an unfinished process. If AutoSense is restarted by the watchdog timer and finds autosave data, it will load the files to restore its state. AutoSense then causes the robot control software to reposition the part and WLI at the beginning of the current ring, and the process is resumed by rescanning the last unfinished ring. Meanwhile, the restored data is sent to AutoProcess. The AutoProcess receive strategy is to replace any existing scans by newer scans if they have the same identifier. Thus, the process can be safely resumed if AutoSense crashes. It should be noted that the process pauses when AutoSense crashes, because no new data can be collected.

AutoProcess, on the other hand, has to serialize all received data onto the hard drive in any event. This is because it not only has to generate a result but also has to keep the raw data for documentation purposes. The idea is, therefore, to utilize a work folder that contains two subfolders, one for all received raw data and one for the data that has been processed so far. At the end of a process, the work folder is renamed and moved to a job collection folder as a permanent record. In the event of a crash, AutoProcess simply looks for the presence of a work folder. If none is present, the previous process was either completed or canceled and AutoProcess starts up normally. Otherwise, the data pertaining to the unfinished panel is deserialized from the hard drive. Afterward, AutoProcess sends a signal to AutoSense that it is ready to receive new data. Hence, the process has been resumed. It should be noted that the process can continue even if AutoProcess crashes, because AutoSense can buffer all the recorded data until AutoProcess connects and starts receiving and processing the data stored by AutoSense.

## D.3 White Light Interferometer Capture Errors

The WLI's capturing routine sometimes fails. This could be because a hardware trigger misfired, because a linear axis command was not transmitted correctly, because the USB data transmission

failed, or because the set parameters do not permit correct scanning behavior.<sup>1</sup> All of these errors should be handled at the appropriate level, as detailed in the following paragraphs.<sup>2</sup>

If the TCP/IP connection to the axis is terminated, AutoSense tries to reestablish the connection. After a defined number of unsuccessful tries, the system enters the error state (see Figure 3.9). If a connection exists but the axis does not respond as expected, the command is resent a defined number of times. If this is unsuccessful, the connection is reset, and the command is resent again. If the error persists even after that last measure, the system enters the error state.

If all axis commands were processed as expected but the interferometer does not send the expected data via USB, there could be one of three causes. First, the data was requested too early from the camera, leading to a USB timeout. If the request times out, data transmission is restarted after a defined waiting period. Second, potentially no data was present in the camera's buffer, indicating that the hardware was not triggered correctly. Therefore, if the second data request also does not yield any data, a scan re-snap is performed (see the following paragraph). Lastly, if the USB connection was interrupted, this might indicate a crashed USB driver (either on the software or on the operating system level) or an interrupted hardware connection. This hardware connection issue can have several reasons: a defective or removed cable, a camera firmware crash or a power outage. The different underlying issues each require their own measures. The error-handling strategy thus progresses from least to most time-consuming. It should be noted that every one of these strategies requires a re-snap. Therefore, the simplest step of soft-restarting the interferometer is skipped because it was shown empirically that it was seldom successful and thus wastes time, mostly because of the required re-snap. Hence, the first measure is to restart the hardware by requesting a relay controlling the interferometer's power supply to toggle through the programmable logic controller (PLC). Afterward, a reconnect is performed and a re-snap requested. If the previous strategy should prove to be unsuccessful, a camera-issue marker file is written and AutoSense is restarted. The previous state is subsequently restored, and the last ring is restarted. If the camera still cannot perform any measurements and the camera-issue marker is present in the backup folder, a reboot marker is written to the backup folder and the computer is restarted. The state is then restored, and the last ring is restarted. If the interferometer issue persists, it cannot be resolved automatically, and the system therefore enters its error state (see Figure 3.9).

Because of the high degree of parallelization, the external axis will have already moved the part to the next measurement position by the time AutoSense recognizes that a re-snap must be performed (see section 4.2). Therefore, AutoSense can send the command *ReturnTo!<ScanID>* to the robot control software AutoXIV. In response, AutoXIV interrupts the current measurement process, moves the part to the requested scan position and resumes the process. In the meantime, AutoSense ignores all snap commands that are more recent than the snap that has to be repeated. Because AutoProcess always overwrites existing scans, it does not even have to be informed that a re-snap is being performed.

<sup>1</sup>However, it could also simply be caused by a programming error.

<sup>2</sup>What is not discussed here are common ways to deal with, or prevent, commonly encountered programming errors — such as data validity checks, null reference checks, `try-catch-finally` blocks and so on.

## **D.4 Error Handling Summary**

With the presented measures, a wide array of both hardware and software issues can be recovered from automatically. This complex error-handling scheme is required because the white light interferometer was not designed for use in an automated system and is thus relatively sensitive to errors. Because of the vast number of scans required to capture an entire CFM-56 outer liner, the system is bound to run into errors at some point. If every error required human interaction, the process would not be accepted in practice and could not truly be considered a fully automated process.

# Index

- API (Application programming interface), 77, A43
- CAD (Computer-aided design), 36
- CCD (Charge-coupled device), 25, A9
- CMOS (Complementary metal-oxide-semiconductor), A9
- Computed tomography (CT), 20, 21, 31, 36, A40
- DC (direct current), 45, 46, 48, A27
- Degree(s) of freedom (DOF), 7, 11, 48, 50, 76, 89, 90, 148
- Dirac pulse, 62, A26
- DLL (Dynamic-link library), A43
- FFT/DFT (Fast/Discrete Fourier transform), 106, A25
- Field of view (FOV), ii, 5–8, 47, 94
- FPGA (Field-programmable gate array), 61, 77
- FPI (Fluorescent dye penetrant inspection), 3, 4, 7, 20, 32, 39, 40, 48, 135, 154–156
- Full width at half maximum (FWHM), A12
- GPU (Graphics processing unit), 52
- Homogenous transform, 49, 87, 88, 134
- IC (Integrated circuit), 48, 61
- ICP (Iterative closest points), 9, 39, 87, 89–91, 121, 122, 148, 149
- Infrared (IR), 14, 24, 25, 35
- IQ (In-phase and quadrature), 38, 42, 48, A35
- Laser line section scanner (LLS), 8, 11, 26, 39, 50, 120–123, 145, 147
- LED (Light-emitting diode), 27, 28, A15
- Levenberg-Marquardt algorithm (LM), 89
- Lookup table (LUT), 59–61, 79, 80, 117, 141
- MRO (Maintenance, repair, and overhaul), i, ii, 2, 19, 29, 32, 33, 36, 38, 151
- NDI/NDT (Non-destructive inspection/testing), ii, 13–15, 18, 19, 25, 29, 30, 34, 154, 156
- PDF (probability density function), 116
- POD (Probability of detection), 7, 11, 39, 143, 151–154
- Probability density spectrum (PDS), A15
- Quadrature, *see* IQ (In-phase and quadrature)
- RAM (Random access memory), 51, 59, 77, 95, A47
- RANSAC (Random sample consensus), 39, 113, 114, 116, 144
- RGB (red, green, and blue (color model)), 32, 33, 123
- Robot, 6, 7, 9, 36, 38, 43, 48–50, 53, 55, 62, 76, 86, 90, 92, 122, 156, A45
- ROI (Region of interest), 140, 141

Signal to noise ratio (SNR), 37

Singular value decomposition (SVD), 88, 109

TCP (Tool center point), 6, 9, 48, 49, 55, 86,  
92, 98

TCP/IP (Transmission control protocol over  
internet protocol), 51, 53, 76, A43

Ultraviolet (UV), 24, A2

UML (Unified modeling language), 52, 53,  
97, A43

White light interferometry (WLI), ii, 5–13,  
26–28, 33, 36, 38–40, 44–46, 48–55,  
57, 58, 61, 62, 72–74, 76, 79, 82, 95,  
120–122, 134, 143, 145, 146, 150,  
151, 155, 157, A9

# List of References

- [Ale16] O. Alekseychuck. “Detection of crack-like indications in digital radiography by global optimisation of a probabilistic estimation function”. PhD thesis. TU Dresden, 2016, pages: 11, 13, 17, 18, 20, 21–27, 27–44, 39 (cited on pages 21, 37).
- [AM02] J. D. Andrews and T. R. Moss. *Reliability and Risk Assessment*. Volume 2nd ed. John Wiley and Sons, Inc, 2002, page: 4. ISBN: 9781860582905 (cited on page 1).
- [Ans+92] G. Ansley, S. Bakans, M. Castronuovo, T. Grant, and F. Vichi. *Current Nondestructive Inspection Methods For Aging Aircraft*. Technical report. United States Federal Aviation Administration, June 1992, pages 7, 13, 18, 55 (cited on pages 14, 17, 19, 20, 25).
- [APR02] C. Akcay, P. Parrein, and J. P. Rolland. “Estimation of longitudinal resolution in optical coherence imaging”. In: *Appl. Opt.* 41.25 (Sept. 2002), pages: 5256–5262 (cited on page A12).
- [Arr13] M. A. T. Arredondo. “Acoustic Emission Testing and Acousto-Ultrasonics for Structural Health Monitoring”. PhD thesis. Universität Siegen, 2013, page: 4 (cited on page 23).
- [Asa14] N. Asadizanjani. “3D Imaging and Investigation of Failure and Deformation in Thermal Barrier Coatings Using Computed X-ray Tomography”. PhD thesis. 2014, pages: 14–15, 21–24 (cited on page 36).
- [AST12] ASTM International Subcommittee E07.10. *ASTM E2862-12, Standard Practice for Probability of Detection Analysis for Hit/Miss Data*. ASTM International. West Conshohocken, PA, USA, 2012, pages: 1, 2, 4 (cited on pages 151, 152).
- [Aus06] Australian Civil Aviation Safety Authority. *Magnetic Particle Inspection - Use and Implementation of ASTM-E-1444*. 2006 (cited on page 18).
- [Bah+17a] S. Bahr, M. Otto, T. Domaschke, and T. Schüppstuhl. “Continuous Digitalization of Rotationally Symmetrical Components With a Lateral Scanning White Light Interferometer”. In: *Tagungsband des 2. Kongresses Montage Handhabung Industrieroboter*. Edited by T. Schüppstuhl, J. Franke, and K. Tracht. MHI. Springer-Verlag Berlin Heidelberg, 2017 (cited on page 157).

- [Bah+17b] S. Bahr, M. Otto, T. Domaschke, and T. Schüppstuhl. “Konzepte zur Nachführung des Messvolumens eines lateral messenden Weißlichtinterferometers”. In: *Tagungsband Automation 2017*. VDI Wissensforum GmbH. June 2017 (cited on page 157).
- [BD13] S. K. Burke and R. J. Ditchburn. *Review of Literature on Probability of Detection for Magnetic Particle Nondestructive Testing*. Technical report. Maritime Platforms Division of the Defence Science and Technology Organisation of Australia, Jan. 2013, page: 3 (cited on page 19).
- [Bee+04] S. Beer, P. Zeller, N. Blanc, F. Lustenberger, and P. Seitz. “Smart Pixels for Real-time Optical Coherence Tomography”. In: *Three-Dimensional Image Capture and Applications VI*. Edited by B. D. Corner, P. Li, and R. P. Pargas. Volume 5302. SPIE. Apr. 2004, page: 4 (cited on page A27).
- [BHN00] M. Blodgett, W. Hasan, and P. B. Nagy. “Theoretical and experimental investigations of the lateral resolution of eddy current imaging”. In: *Materials Evaluation* 58 (May 2000), pages: 647–654 (cited on pages 15, 16).
- [BL08] L. Brancheriau and J.-D. Lanvin. *Research into non-destructive compression crack detection methods for wood structures*. Technical report. Directorate General for Civil Aviation (France), Feb. 2008, pages: 18, 24, 27 (cited on pages 14, 15).
- [BM92] P. J. Besl and N. D. McKay. “A Method for Registration of 3-D Shapes”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 14.2 (Feb. 1992), pages: 239–256. ISSN: 0162-8828 (cited on page 89).
- [Boh12] M. H. Bohun. “Several Non-Destructive Inspection Methods Applied to Quantify Fretting Fatigue Damage in Simulated TI-6AL-4V Turbine Engine Dovetail Components”. PhD thesis. University of Dayton, Ohio, May 2012, pages: 109, 115 (cited on pages 15–17).
- [BPF12] J. Beyerer, F. Puente León, and C. Frese. *Automatische Sichtprüfung: Grundlagen, Methoden und Praxis der Bildgewinnung und Bildauswertung*. Berlin Heidelberg: Springer, 2012, pages: VII, 8, 302, 814 (cited on pages 21, 25, 26).
- [BR12] P. Broberg and A. Runnemalm. “Detection of Surface Cracks in Welds using Active Thermography”. In: *18th World Conference on Nondestructive Testing*. Durban, South Africa: South African Institute for Non-Destructive Testing (SAINT), Apr. 2012, pages: 1, 2, 5 (cited on page 35).
- [Bra+04] L. Brasche et al. *Engineering Studies of Cleaning and Drying Processes in Preparation for Fluorescent Penetrant Inspection*. Technical report. Iowa State University Center for Nondestructive Evaluation for the Federal Aviation Administration (of the USA), Jan. 2004, pages: xiii, 1 (cited on page 20).

- [Brä01] W. Bräunling. *Flugzeugtriebwerke: Grundlagen, Aero-Thermodynamik, Kreisprozesse, thermische Turbomaschinen, Komponenten- und Auslegungsberechnungen mit 33 vollständig durchgerechneten Beispielen*. VDI-Buch. Springer, 2001, pages: 216, 452, 1169. ISBN: 9783540675853 (cited on pages 2, 3).
- [Bu+15] G. Bu, S. Chanda, H. Guan, J. Jo, M. Blumenstein, and Y. Loo. “Crack Detection using a Texture Analysis-based Technique for Visual Bridge Inspection”. In: *Special Issue: Electronic Journal of Structural Engineering* 14(1) (2015), page 41, 42, 45, 46 (cited on page 32).
- [Cam13] F. C. Campbell. *Inspection of Metals: Understanding the Basics*. Edited by F. C. Campbell. 978-1-62708-000-2. ASM International, Apr. 2013, pages: 1, 11, 12, 13, 16, 18, 19 (cited on pages 14, 17–21, 23, 25, 30, A39).
- [Can86] J. Canny. “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (Nov. 1986), pages: 679–698 (cited on pages 125, 126).
- [Czi13] H. Czichos. *Handbook of technical diagnostics: Fundamentals and application to structures and systems*. Springer Science & Business Media, 2013, page: 47 (cited on page 30).
- [DH04] M. Dusek and C. Hunt. *Crack Detection Methods for Lead-free Solder Joings*. Technical report. National Physical Laboratory, Middlesex, UK, Mar. 2004 (cited on page 14).
- [DH72] R. O. Duda and P. E. Hart. “Use of the Hough Transformation to Detect Lines and Curves in Pictures”. In: *Communications of the ACM* 15 (1972), 11–15 (cited on page 132).
- [DHV92] T. Dresel, G. Häusler, and H. Venzke. “Three-dimensional sensing of rough surfaces by coherence radar”. In: *Applied Optics* 31.7 (Mar. 1992), pages: 919–925 (cited on pages 28, 29, A10).
- [DMQ16] S. Dorafshan, M. Maguire, and X. Qi. “Automatic Surface Crack Detection in Concrete Structures Using OTSU Thresholding and Morphological Operations”. In: *CEE Faculty Publications* UTC Report 01-2016 (Apr. 2016), pages: 8, 33 (cited on page 33).
- [Dom17] T. Domaschke. “Automatisierung der Weißlichtinterferometrie zur Inspektion rotationssymmetrischer Triebwerksbauteile”. PhD thesis. Technische Universität Hamburg-Harburg, Dec. 2017 (cited on page 6).
- [Dös16] O. Dössel. *Bildgebende Verfahren in der Medizin: Von der Technik zur medizinischen Anwendung*. Springer Berlin Heidelberg, 2016, page 131. ISBN: 9783642544071 (cited on page 21).

- [DSO14] T. Domaschke, T. Schüppstuhl, and M. Otto. “Robot Guided White Light Interferometry for Crack Inspection on Airplane Engine Components”. In: *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*. June 2014, pages: 1–7 (cited on pages 5, 30, A39).
- [DUP07] A. A. Diaz, U.S. Nuclear Regulatory Commission, and Pacific Northwest National Laboratory (U.S.) *Assessment of Eddy Current Testing for the Detection of Cracks in Cast Stainless Steel Reactor Piping Components*. English. Division of Fuel, Engineering, and Radiological Research, Division of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission Washington, DC, 2007, pages 3.1, 3.4 (cited on pages 16, 17, 34).
- [Fis17] G. Fischer. *Lernbuch Lineare Algebra und Analytische Geometrie*. Springer Fachmedien Wiesbaden, 2017, page: 448. ISBN: 978-3-658-18191-8 (cited on page 88).
- [Foe16] Foerster Group. *Completely Automatic Eddy Current Testing Systems*. <http://www.foerstergroup.de/Completely-automatic-eddy-current-testing-systems.114+M52087573ab0.0.html>. Accessed: 2016-11-7. 2016 (cited on page 34).
- [For15] D. Forsberg. *Aircraft Retirement and Storage Trends - Economic Life Analysis Reprised and Expanded*. Technical report. AVOLON, Mar. 2015, page: 2 (cited on page 2).
- [GG06] H. Gevatter and U. Grünhaupt. *Handbuch der Mess- und Automatisierungstechnik in der Produktion*. VDI-Buch. Springer Berlin Heidelberg, 2006, pages: 271, 372, 381, 382, 383, 385, 387, 404. ISBN: 9783540212072 (cited on pages 21–24, 29, 30, A39).
- [GGV11] J. García-Martín, J. Gómez-Gil, and E. Vázquez-Sánchez. “Non-Destructive Techniques Based on Eddy Current Testing”. In: *Sensors* 11.3 (2011), pages: 2525, 2526, 2527, 2529, 2533, 2540, 2544, 2558. ISSN: 1424-8220 (cited on pages 15–17, 34).
- [GH89] Z. Guo and R. W. Hall. “Parallel Thinning with Two-Subiteration Algorithms”. In: *Communications of the ACM* 32.3 (Mar. 1989), pages: 359–373 (cited on pages 128, 129).
- [Gho18] A. Ghosh A.; Basu. “A Generalized Relative  $(\alpha, \beta)$ -Entropy: Geometric Properties and Applications to Robust Statistical Inference”. In: *Entropy* 20.347 (2018), 8–39 (cited on page 115).
- [GT11] Z. Gan and Q. Tang. *Visual Sensing and Its Applications: Integration of Laser Sensors to Industrial Robots*. 1st. Springer Publishing Company, Incorporated, 2011, pages: 42, 117. ISBN: 9783642182860 (cited on pages 49, 79).
- [Hay97] Haynes International, Inc. *HASTELLOY® X Alloy*. <http://mail.haynesintl.com/pdf/h3009.pdf>. Accessed: 2018-09-13. 1997 (cited on page 3).

- [Hea02] Health and Safety Executive HSE (UK). *Best Practice for the Procurement and Conduct of Non-Destructive Testing - Part 2: Magnetic Particle and Dye Penetrant Inspection*. 2002 (cited on pages 17–19).
- [HH11] C. A. Harding and G. R. Hugo. *Review of Literature on Probability of Detection for Liquid Penetrant Nondestructive Testing*. Technical report DSTO-TR-2623. Maritime Platforms Division of the Defence Science and Technology Organisation of Australia, Nov. 2011, page: 1 (cited on page 154).
- [HOO12] M. Herrmann, M. Ottesteanu, and M. Otto. “Smoothing Splines zur automatischen Verarbeitung von 3D-Bilddaten für die Robotersichtführung”. In: *Automation, 13. Branchentreff der Mess- und Automatisierungstechnik, VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA), + CD-ROM, Baden-Baden, DE, 13.-14. Jun, 2012*. Volume 2171. VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA). VDI-Verlag, Düsseldorf, June 2012, pages: 279–283 (cited on page 141).
- [Hou62] P. V. C. Hough. “Method and means for recognizing complex patterns”. US3069654A. Dec. 18, 1962 (cited on page 132).
- [HS11] H. Heuer and M. H. Schulze. “Eddy current testing of carbon fiber materials by high resolution directional sensors”. In: *Smart Materials, Structures and NDT in Aerospace, NDT in Canada (2011)* (cited on page 34).
- [Iba+07] C. Ibarra-Castanedo, M. Genest, J.-M. Piau, S. Guibert, A. Bendada, and X. Maldague. “Active Infrared Thermography Techniques for the Nondestructive Testing of Materials”. In: *Ultrasonic and Advanced Methods for Nondestructive Testing and Material Characterization*. Edited by C. Chen. World Scientific Publishing, 2007, pages: 325–348 (cited on pages 24, 25).
- [ibg16] ibg NDT Systems Corporation. *Eddy Current Crack Testing Systems - Ball Studs, Needles, Cylinder Liners*. <http://www.ibgndt.com/crack-testing-system.php>. Accessed: 2016-11-7. 2016 (cited on page 34).
- [IBM11] C. Ibarra-Castanedo, A. Bendada, and X. Maldague. “Infrared vision applications for the nondestructive testing of materials”. English. In: *5th Pan American Conference for NDT, (Cancun, Mexico)*. Cancun, Mexico: www.ndt.net, Oct. 2011, pages: pp. 1 (cited on page 24).
- [Jäh12] B. Jähne. *Digitale Bildverarbeitung und Bildgewinnung*. Berlin: Springer Vieweg, 2012, pages: 197, 253, pp. 339, 342, pp. 367, 555, 561, 562, 680, 687, 690 691. ISBN: 978-3-642-04951-4 (cited on pages 25, 26, 125–127).
- [JH00] B. Jähne and H. Haußecker, editors. *Computer Vision and Applications: A Guide for Students and Practitioners*. Orlando, FL, USA: Academic Press, Inc., 2000, pages: 177, 201. ISBN: 0-12-379777-2 (cited on pages 26, A10).

- [KM07] N. Klußmann and A. Malik. *Lexikon Der Luftfahrt*. Springer Berlin Heidelberg, 2007, pages: pp. 290, 308. ISBN: 9783540490968 (cited on page 2).
- [Kyl+14] A. Kylili, P. A. Fokaides, P. Christou, and S. A. Kalogirou. “Infrared thermography (IRT) applications for building diagnostics: A review”. In: *Applied Energy* 134.C (2014), pages: 531–549 (cited on page 14).
- [LA03] J. Lively and T. Aljundi. “Fluorescent Penetrant Inspection Probability of Detection Demonstrations Performed For Space Propulsion”. In: *Review of Quantitative Nondestructive Evaluation* 22 (2003), pages: 18, 22, 23 (cited on page 154).
- [Li14] L. Li. *Time-of-Flight Camera – An Introduction*. White Paper SLOA190B. Texas Instruments, May 2014, page: 1 (cited on page 26).
- [Liu+16] H.-m. Liu, Z.-c. Huang, Y.-t. Zang, and A. M. A. Talab. “Shape-based Micro Crack Detection in Plastic through Image Processing”. In: *IJSIP International Journal of Signal Processing, Image Processing and Pattern Recognition* 9.1 (2016), pages: 282, 283 (cited on page 33).
- [LKR15a] Y. Li, M. Kästner, and E. Reithmeier. “Development of a compact low coherence interferometer based on GPGPU for fast microscopic surface measurement on turbine blades”. In: *SPIE Optical Metrology*. Edited by P. Lehmann, W. Osten, and A. Albertazzi Gonçalves. SPIE Proceedings. SPIE. 2015, pages: 95250R (cited on page 38).
- [LKR15b] Y. Li, M. Kästner, and E. Reithmeier. “High-precision surface measurement with an automated multiangle low coherence interferometer”. In: *Appl. Opt.* 54.6 (Feb. 2015), pages: 1232–1240 (cited on page 38).
- [LKW95] W. Lauterborn, T. Kurz, and M. Wiesenfeldt. *Coherent Optics - Fundamentals and Applications*. Springer-Verlag Berlin Heidelberg, 1995, pages: 15, 18, 25, 27, 31, 32 (cited on pages 26, A2, A3, A4, A11).
- [LM11] J. Liesen and V. Mehrmann. *Lineare Algebra: Ein Lehrbuch über die Theorie mit Blick auf die Praxis*. Bachelorkurs Mathematik. Vieweg+Teubner Verlag, 2011, pages: pp. 318. ISBN: 9783834882905 (cited on page 109).
- [Lof90] O. Loffeld. *Estimationstheorie I - Grundlagen und stochastische Konzepte*. R. Oldenbourg Verlag, 1990, pages: 58, 66, 105, pp. 157, 191, 217. ISBN: 3-486-21616-3 (cited on pages 63, 64, 80, 88, 99, 115).
- [LWL00] D. Lamtenzan, G. Washer, and M. Lozev. *Detection and Sizing of Cracks in Structural Steel Using the Eddy Current Method*. Final Report FHWA-RD-00-018. Nondestructive Evaluation Validation Center, McLean, VA, USA, 2000, page: 16 (cited on pages 15, 34).

- [Mai+10] S. Maillard, J. Cadith, D. Eschimese, H. Walaszek, H. Mooshofer, J.-C. Candoré, and J. I. Bodnar. “Towards the use of passive and active infrared thermography to inspect metallic components in the mechanical industry”. In: *10th International Conference on Quantitative InfraRed Thermography (QIRT 2010)*. Edited by X. Maldague. Quebec, Canada, June 2010, pages: 2, 6, 7 (cited on page 35).
- [Mai+12] S. Maillard, J. Cadith, P. Bouteille, G. Legros, J. L. Bodnar, and V. Detalle. “Non-destructive Testing of Forged Metallic Materials by Active Infrared Thermography”. In: *International Journal of Thermophysics* (Nov. 2012), pages: pp. 1986 (cited on page 35).
- [Mat14] MathWorks. *mldivide - Solve systems of linear equations*. <https://mathworks.com/help/matlab/ref/mldivide.html>. Accessed: 2019-11-14. Mar. 8, 2014 (cited on page 109).
- [Max65] J. C. Maxwell. “VIII. A dynamical theory of the electromagnetic field”. In: *Philosophical Transactions of the Royal Society* 155 (Dec. 1865), 459–512 (cited on pages 26, A2).
- [Mes15] D. Meschede. *Gerthsen Physik*. 25th edition. Springer Spektrum, 2015, pages: 442, 464, 534, 672 (cited on pages 27, A2, A3, A7).
- [MHU07] G. Mook, O. Hesse, and V. Uchanin. “Deep Penetrating Eddy Currents and Probes”. In: *Materials Testing* 49.5 (2007), pages: 258–264 (cited on pages 15–17).
- [Mil09] Military Handbook. “MIL-HDBK-1823A”. In: *C. Annis, Nondestructive Evaluation System Reliability Assessment. Department of Defense Handbook, Wright-Patterson AFB, USA*. 2009, pages: 6, 15, 16, 27, 120, 130 (cited on pages 151, 152).
- [Miu+01] T. Miura, M. Ochiai, H. Kuroda, S. Kanemoto, and S. Soramoto. “Laser-induced surface wave testing: a new method for measuring the depth of cracks”. In: *9. international conference on nuclear engineering*. Volume 33. 16. Apr. 2001, page: 3 (cited on page 23).
- [ODH13] M. Otto, T. Domaschke, and M. Herrmann. “Hochgenaue 3D-Schadensdetektion an rotationssymmetrischen Maschinenteilen mit robotergeführtem Weißlichtinterferometer”. In: *AUTOMATION 2013: “Automation (in the) cloud”*; *14. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden*. VDI-Verlag Düsseldorf, June 2013, 153–157. ISBN: 978-3-18-092209-6 (cited on pages 5, 30, 33, 38, 49, A39).
- [OL10] J. Ohm and H. Lüke. *Signalübertragung: Grundlagen der digitalen und analogen Nachrichtenübertragungssysteme*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2010, pages: 206, 215, 360 (cited on pages 148, A12, A32, A33).
- [Ols00] A. Olszak. “Lateral Scanning White-Light Interferometer”. In: *Applied Optics* 39 (Aug. 2000), pages: 3906–3913 (cited on page 157).

- [Ope17] OpenCV Developer Team. *Canny Edge Detector*. [http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny\\_detector/canny\\_detector.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html). Accessed: 2017-09-7. Sept. 2017 (cited on page 126).
- [OT03] K. Otsuka and M. Takeda. “Detection of Fine Cracks by X-Ray Technique Using Contrast Medium in Concrete (Translation)”. In: *Proceedings of JSCE*. Volume 58. 725. JSCE. 2003, pages: 121–136 (cited on page 21).
- [Ott+14] M. Otto, T. Domaschke, S. Bahr, and M. Schnell. “Präzise Rissdetektion und -bewertung an Flugzeugturbinenlinern mittels Weißlichtinterferometrie”. In: *Automation 2014 [Elektronische Ressource]: Smart X - powered by Automation; 15. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden*. VDI-Verlag, Düsseldorf, July 2014. ISBN: 978-3-18-092231-7 (cited on pages 30, 125, 126, 128, 130–132, 136, A39).
- [Ott+15] M. Otto, T. Domaschke, H. Eschen, S. Bahr, S. Gehlen, W. Neddermeyer, and T. Schüppstuhl. “3D-Präzisionsvermessung von Bauteilen – Automatische Erkennung von Bauteilfehlern und Datenanalyse durch Virtual-Reality-Methoden”. In: *VDI-Fachkonferenz Geometriemesstechnik in der Automobilproduktion*. 2015 (cited on page 120).
- [Pol93] D. S. G. Pollock. *Smoothing With Cubic Splines*. Technical report. London: University of London, Queen Mary and Westfield College, 1993 (cited on page 141).
- [Rei12] G. A. Reider. *Photonik*. 3. Auflage. Springer-Verlag Wien, 2012, page: 176 (cited on page A12).
- [Rog13] A. Rogalski. “Infrared detectors: Status and trends”. In: *Progress in Quantum Electronics* (Dec. 2013) (cited on page 25).
- [RS09] H. Richard and M. Sander. *Ermüdungsrisse: erkennen, sicher beurteilen, vermeiden*. Vieweg Praxiswissen. Vieweg + Teubner, 2009, page: 26. ISBN: 9783834802927 (cited on pages 135, 136).
- [RW17] T. Richter and T. Wick. *Einführung in die Numerische Mathematik*. Springer Spektrum, 2017. ISBN: 978-3-662-54178-4 (cited on page 109).
- [Šác08] Šáclav. “Automated System for Crack Detection Using Infrared Thermographic Testing”. In: *17th World Conference on Non-Destructive Testing 2008 (WCNDT 2008)*. International Committee for Non-Destructive Testing (ICNDT), Oct. 2008, pages: 3, 4, 5 (cited on page 35).
- [Sch+16] C. Schwienbacher, T. Domaschke, M. Otto, and T. Kötter. “Analysis and Reduction of Errors in an Automated Robot-Based Repair Process Chain”. In: *1st MHI Colloquium*. Trans Tech Publications, 2016 (cited on pages 33, 134, 157).
- [Sim96] D. A. Simon. “Fast and Accurate Shape-Based Registration”. PhD thesis. Carnegie Mellon University, Dec. 1996, pages: pp. 28 (cited on page 89).

- [SKS14] C. Schwienbacher, T. Kötter, and T. Schüppstuhl. “Repair of Aircraft Combustors with Industrial Robots”. In: *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*. June 2014 (cited on page 4).
- [SOB18] K. Studemund, M. Otto, and S. Bahr. “Maschinelles Lernen - Selbstlernende Detektion und Klassifizierung von Schäden in Flugzeugtriebwerksbauteilen”. In: *AUTOMATION, 19. Leitkongress der Mess- und Automatisierungstechnik - Seamless Convergence of Automation and IT*. Edited by VDI Wissensforum GmbH. 2330. VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA). VDI-Verlag GmbH Düsseldorf, July 2018, pages: 343–354 (cited on page 157).
- [SR14] H. Süße and E. Rodner. *Bildverarbeitung und Objekterkennung: Computer Vision in Industrie und Medizin*. Springer Fachmedien Wiesbaden, 2014, pages: pp. 639, pp. 642. ISBN: 9783834826060 (cited on page 109).
- [SR17] O. Sorkine-Hornung and M. Rabinovich. *Least-Squares Rigid Motion Using SVD*. Technical report. ETH Zurich Department of Computer Science, Jan. 2017 (cited on page 87).
- [Ste17] O. Stein. *Grundzüge der Nichtlinearen Optimierung*. Springer Berlin Heidelberg, 2017, pages: pp. 81. ISBN: 9783662555934 (cited on page 89).
- [Stü+07] S. Stürwald, B. Kemper, C. Denz, and G. v. Bally. “Charakterisierung von ultrahellen Leuchtdioden für den Einsatz in der digitalen Holographie”. In: *Jahrestagung der DGaO*. Volume 108. Deutsche Gesellschaft für angewandte Optik (DGaO). May 2007 (cited on page A12).
- [Sun17] D. Sundararajan. *Digital Image Processing - A Signal Processing and Algorithmic Approach*. Springer, 2017, pages: pp. 206 (cited on page 21).
- [TM15] P. A. Tipler and G. Mosca. *Physik*. Edited by J. Wagner. 7th edition. Springer Spektrum, 2015, pages: 474, 634, 639, 663, 778, 896, 989, 1123, 1163 (cited on pages 24, 26, A3, A23).
- [Uni01] United States Federal Aviation Administration. *Acceptable methods, techniques, and practices—aircraft inspection and repair*. English. U.S. Dept. of Transportation, Federal Aviation Administration [Washington, D.C.], 2001, chapter 5, pages: 6, 7, 9, 16, 22, 25, 27, 29, 34, 37, 41, 42, 43, 53 (cited on pages 14, 17–23, 25, 34, A39).
- [Vis16] Vishay Semiconductors. *TLDR5800 High Intensity LED*. <http://www.vishay.com/docs/83004/tldr5800.pdf>. Accessed: 2016-1-10. June 2016 (cited on pages A15, A20).
- [Vor+15] C. Vornholt et al. *Smart Maintenance für Smart Factories: Mit intelligenter Instandhaltung die Industrie 4.0 vorantreiben*. acatech POSITION. Herbert Utz Verlag, 2015, pages: 11–15. ISBN: 9783831644933 (cited on page 2).

- [Wal17] S. Waldmann. *Lineare Algebra 1*. Springer Spektrum, 2017, page: 236. ISBN: 978-3-662-49914-6 (cited on page 100).
- [Wan+11] X. Wang, B. S. Wong, C. S. Tan, and C. G. Tui. “Automated Crack Detection for Digital Radiography Aircraft Wing Inspection”. In: *Research in Nondestructive Evaluation 22* (2011), pages: 105–127 (cited on pages 36, 37).
- [Web09] W. Weber. *Industrieroboter: Methoden der Steuerung und Regelung*. Fachbuchverl. Leipzig im Carl-Hanser-Verlag, 2009, pages: pp. 29. ISBN: 9783446410312 (cited on page 87).
- [WRR03] M. E. Wall, A. Rechtsteiner, and L. M. Rocha. “Singular value decomposition and principal component analysis”. In: edited by D. P. Berrar, W. Dubitzky, and M. Granzow. Springer US, 2003. Chapter 5, pages: pp. 1 (cited on page 88).
- [Xu+10] J. Xu, T. Liu, X. Yin, B. S. Wong, and S. Hassan. “Automatic X-ray Crack Inspection for Aircraft Wing Fastener Holes”. In: *2nd International Symposium on NDT in Aerospace 2010*. DGZfP-Proceedings BB 124-CD, 124. 2010, pages: 2–5 (cited on pages 36, 37).
- [Zei+12] E. Zeidler et al. *Springer-Taschenbuch der Mathematik: Begründet von I.N. Bronstein und K.A. Semendjaew Weitergeführt von G. Grosche, V. Ziegler und D. Ziegler Herausgegeben von E. Zeidler*. SpringerLink : Bücher. Springer Fachmedien Wiesbaden, 2012, pages: pp. 31, 57, 61, 266, 361, 592. ISBN: 9783834823595 (cited on pages 95, 107, 111, 137, A4, A25, A26).
- [Zha+14] W. Zhang, Z. Zhang, D. Qi, and Y. Liu. “Automatic Crack Detection and Classification Method for Subway Tunnel Safety Monitoring”. In: *Sensors* 14.10 (2014), pages: 19307–19328 (cited on pages 32, 33).