



Robert Stindl, BSc

**Evaluierung eines 3D Time-of-Flight Sensors zur Bestimmung
von mechanischen Schwingungen:
Realisierbarkeit und Signalauswertung**

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Technische Physik

eingereicht an der

Technischen Universität Graz

Betreuer

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Gernot Pottlacher
Institut für Experimentalphysik

Mitbetreuer

Univ.-Prof. Mag.rer.nat. Dr.rer.nat. Alexander Bergmann
Institut für Elektronische Sensorsysteme

EIDESSTATTLICHE ERKLÄRUNG

AFFIDAVIT

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit/Diplomarbeit/Dissertation identisch.

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis/diploma thesis/doctoral dissertation.

Datum / Date

Unterschrift / Signature

Abstract

The Time-of-Flight (TOF) technology becomes accurate and more high-resolution in the visual 3D representation. This results in new applications, which have not been evaluated yet. The aim of this work is to detect vibrations contactlessly with an analysing algorithm. The algorithm uses data generated by a commercial Time-of-Flight (TOF) chip. The basic motivation for this work is the extension of a diagnostic device from AVL DiTEST with a contactless vibration measurement. The raw data is provided by a TOF-Chip from ESPROS Photonics AG and further evaluated with a MATLAB based analysing algorithm. The image sequence and the information from raw data were used to perform a Fourier transformation. The resulting frequency spectra of individual pixels were analysed and exported into a matrix with corresponding frequencies. By means of this procedure, locally different frequencies can be detected. Further, the data can be displayed in form of an image, colored according to the frequencies. Finally, the performance of this method could be demonstrated under real conditions on a starting car engine.

Zusammenfassung

Die Time-of-Flight (TOF)-Technologie wird in der bildgebenden 3D-Darstellung immer genauer und hochauflösender. Damit ergeben sich für diese Technik neue Anwendungsmöglichkeiten, die noch nicht evaluiert worden sind. Das Ziel dieser Arbeit ist, die Möglichkeit der Vibrationsdetektion mittels der TOF-Technik zu überprüfen, eine Vorgehensweise für die Analyse zu entwickeln und die derzeitigen Grenzen der Technologie für diesen Anwendungsfall zu untersuchen. Die Grundmotivation für diese Arbeit ist die mögliche Erweiterung eines Diagnosewerkzeugs der Firma AVL DiTEST mit einer berührungslosen Vibrationsmessung. Die Rohdaten der Serienbilder wurden mit einem bildgebenden TOF-Chip der Firma ESPROS Photonics AG aufgenommen und anschließend durch einen eigens dafür entwickelten Analysealgorithmus in MATLAB ausgewertet. Durch die Bildabfolge und mithilfe der ermittelten Informationen aus den Rohdaten wurde für jedes Pixel eine Fourier-Transformation durchgeführt. Die daraus resultierenden Frequenzspektren der einzelnen Pixel wurden analysiert und die entsprechenden Frequenzen ausgegeben. Durch diese Vorgehensweise können lokal unterschiedliche Frequenzen festgestellt und als frequenzabhängiges, eingefärbtes Bild dargestellt werden. Abschließend konnte die Leistungsfähigkeit des entwickelten Algorithmus in Kombination mit dem entsprechenden TOF-Chip unter realen Bedingungen anhand eines startenden Automotors gezeigt werden.

Danksagung

Es ist nicht wichtig, wie groß der erste Schritt ist, sondern in welche Richtung er geht.

[Verfasser unbekannt]

Den Schritt in Richtung eines Studiums habe ich vor einigen Jahren trotz fixer Arbeitsstelle gewagt und einen neuen Lebensabschnitt begonnen. In diesem Lebensabschnitt habe ich viel Neues erlernen dürfen, nicht nur fachlich, sondern auch persönlich. An dieser Stelle danke ich meiner Familie, meinen Freunden und meinen Kommilitonen, dass sie mich immer motivierten und bei meinem Vorhaben unterstützten. Die Finanzierung meines Studiums konnte ich durch das staatliche Selbsterhalterstipendium bewerkstelligen und ich möchte auch meinen toleranten Arbeitgebern, die mich mit flexiblen Arbeitsmöglichkeiten unterstützten, meinen Dank aussprechen.

Weiters bedanke ich mich bei meinen Betreuern, Univ.-Prof. Mag.rer.nat. Dr.rer.nat. Bergmann Alexander vom Institut für Elektronische Sensorsysteme und Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Pottlacher Gernot vom Institut für Experimentalphysik, die meine Masterarbeit begleiteten und begutachteten. Sie haben mich stets ermutigt und hilfreiche Anregungen sowie konstruktive Kritik bei der Erstellung der Arbeit gegeben. Unter anderem müssen auch meine Kolleginnen und Kollegen am Institut für Elektronische Sensorsysteme erwähnt werden, die mir während der gesamten Masterarbeit mit Rat und Tat zur Seite standen und mich auch mit Materialien, Essen und Kaffee versorgten.

Zum Abschluss bedanke ich mich auch bei der Firma AVL DiTEST, die mir die Möglichkeit gegeben hat, an einem praxisnahen Thema zu arbeiten, wodurch ich einen Einblick in moderne Arbeitsweisen gewinnen konnte.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Einsatzbereich/Motivation	1
1.2	Vibrationen	3
2	Methodik und Konzept	4
2.1	Flugzeitmessung von Photonen (Time-of-flight)	4
2.2	Direktes Time-of-Flight	5
2.3	Indirektes Time-of-Flight	7
2.3.1	Phasenvergleich	8
2.3.2	Quadraturdemodulation	10
2.3.3	Korrelationsabtastung	11
2.3.4	Amplitudenqualität	15
2.3.5	Integrationszeit	15
2.3.6	Konzept	16
2.4	Fehler der bildgebenden Phasenmessung	18
2.4.1	Statistische Fehler	19
2.4.2	Systematische Fehler	20
2.4.3	Szenenbedingte Fehler	23
2.4.4	Fehler der Frequenzerfassung (Aliasing)	25
3	Realisierung	26
3.1	Auswahl des Sensors	26
3.1.1	Weiterer TOF-Sensor	27
3.2	Beschreibung des verwendeten Evaluation-Sets	28
3.2.1	Hardware	28

3.2.2	Lichtquelle	33
3.2.3	Optik	34
3.2.4	Kontrolleinheit	35
3.2.5	Benötigte Aufnahmezeit pro Bild	35
3.2.6	Abschätzung der Datenmenge	36
3.2.7	Software	36
3.3	Aufbau.	39
3.3.1	Vibrationsquellen	39
3.3.2	Labora Aufbau	40
3.4	Analysealgorithmus	41
3.4.1	Datenerfassung	42
3.4.2	Berechnung und Optimierung.	44
3.4.3	Frequenzdetektion	47
3.4.4	3D-Bildoptimierung	49
3.4.5	Bildausgabe.	49
4	Resultate und Diskussion	51
4.1	Einfluss der Filtermethoden auf die Messung.	51
4.2	Schwellenwert für die Funktion „Signal-Rausch-Analyse“.	54
4.3	Detektion der Auslenkung.	56
4.4	Einfluss der Signalqualität und der Filtermaßnahmen auf die Messgenauigkeit	59
4.5	Einfluss der Temperatur auf die Abstandsmessung des DME660	61
4.6	Detektion zweier Vibrationsquellen in einer Szene.	63
4.7	Skalierung auf höhere Frequenzen	66
4.8	Vibrationsmessung am Automotor	69
5	Schlussfolgerungen und Ausblick	71
6	Anhang\ Matlab-Code	74
6.1	Start-Skript für die Frequenzdetektion	74
6.2	Bildrate virtuell erhöhen	79
6.3	Daten laden.	79
6.4	Daten berechnen.	80

6.5 Ableitungsmethode.	80
6.6 Median-Filter	81
6.7 Frequenzdetektion	81
6.8 Detektierte Frequenzregionen analysieren	82
6.9 Detektierte Frequenzregionen analysieren	82
6.10 Farbgebung der Pixel.	83
6.11 Zusätzliche Funktionen	84
6.11.1 Farben-RGB	84
6.11.2 3D-Plot und Point Cloud	84
6.11.3 FFT-Analyse	85
6.12 Schematische Darstellung Algorithmus	87
Abkürzungsverzeichnis	88
Literaturverzeichnis	90

Einleitung

1.1 Einsatzbereich/Motivation

Die Welt ist in zwei Dimensionen schon sehr gut erfasst, nun geht der Trend zur Detektion von drei Dimensionen (3D). Dieser Kurs zeichnet sich auch in anderen Bereichen, wie zum Beispiel der 3D Erfassung unserer Städte sowie in wissenschaftlichen Analysemethoden von Materialien ab. Diese Entwicklung ist wegweisend und soll eine neue Dimension eröffnen. Die Time-of-Flight (TOF)-Technologie steht derzeit in großem Forschungsinteresse von Chipherstellern, wodurch sich neue Möglichkeiten für diese Technik ergeben. Die Abstands- und Gestenerkennung mittels TOF wird immer öfter im Automobil- oder Mobilgerätebereich eingesetzt. Zurückzuführen ist das auf die stetig besser werdenden Halbleiterelemente und die niedrigen Serienpreise dieser Chips.

Untersucht werden die Möglichkeiten und die Einsatzbereiche des TOF-Verfahrens zur Detektion von Vibrationen. Dies soll durch die Bestimmung der zeitlichen Veränderung des Abstands und durch die Auswertung der relativen Schwankungen des detektierten Signals erreicht werden. Die Grundmotivation der Untersuchung des TOF-Verfahrens für Vibrationsmessungen ist die Erweiterung eines Diagnosewerkzeugs (Multisense 1000, Abbildung 1.1) der Firma AVL DiTEST, welches für Autowerkstätten entwickelt wurde [DiT06].



Abbildung 1.1: Das Multisense 1000 der Firma AVL DiTEST mit verschiedenen Sensoren zur Detektion von Diagnosesignalen für den Autowerkstättenbereich. [DiT06]

Beim bestehenden Multisense 1000 wird die Vibrationsmessung mittels eines Mikrophons und eines Stethoskops durchgeführt. Diese Methode erfordert allerdings einen direkten Kontakt zwischen Messgerät und zu messenden Komponenten, was bedeutet, dass der KFZ-Techniker im Motorraum operieren muss. Um die Sicherheit zu erhöhen, soll die berührungsfreie TOF-Messmethode untersucht und deren derzeitige Grenzen, Potentiale und Einsatzmöglichkeiten bestimmt werden.

Weiters soll das bildgebende TOF-Verfahren die Möglichkeit bieten, mehrere verschiedene Frequenzen in einer Szene zu erfassen und Vibrationen unterschiedlichen Ursprungs zu lokalisieren.

1.2 Vibrationen

Bei mechanischen Vibrationen handelt es sich um oszillierende, makroskopische Auslenkungen eines Objekts. Diese Schwingungen können über die Amplitude (Auslenkung) und die Frequenz charakterisiert werden. In der Literatur werden Schwingungen teilweise auch durch die Beschleunigung [m/s^2] angegeben.

In den meisten Fällen sind mechanische Schwingungen unerwünschte, nicht verhinderbare Nebeneffekte oder eine Folge von mangelhafter Verarbeitung. Die Messung von Vibrationen kann in solchen Fällen eine schnelle und einfache Diagnosemöglichkeit bieten. Störungen können je nach Anwendung die gleichen charakteristischen Vibrationen auslösen. Diese Vibrationen können für die schnelle Analyse von Fehlern sehr hilfreich sein. [PR16, Rei12]

Aufgrund der makroskopischen Auslenkungen und des niedrigen Frequenzbereichs ist es möglich, mechanische Vibrationen zu fühlen, zu sehen oder zu hören. Diese Schwingungen können für den Menschen im wahrnehmbaren sowie auch im für die Sinne nicht erfassbaren Bereich liegen und bei längerer Exposition kann es zu gesundheitlichen Beeinträchtigungen kommen. Vor allem direkte, monotone Beanspruchung durch mechanische Vibrationen belastet zum Beispiel den Stützapparat des Menschen. Auch kurzfristige Belastungen können zur Verminderung der Leistungsfähigkeit, Schwindel und Schmerzen führen. Diese Auswirkungen auf den menschlichen Organismus treten in einem großen Frequenzbereich (ab etwa 1 Hz bis zu Frequenzen im kHz-Bereich) auf. [fWuA06, Mos08]

Methodik und Konzept

2.1 Flugzeitmessung von Photonen (Time-of-flight)

Die Flugzeitmessung wird bei verschiedenen Messprinzipien für die Messung von unterschiedlichsten physikalischen Größen verwendet. Im ersten Moment kann TOF womöglich mit der bekannteren Massenbestimmung über die Flugzeitmessung verwechselt werden, da diese auch in mancher Literatur mit TOF abgekürzt wird. In dieser Arbeit wird die Abkürzung TOF nur für die Time-of-Flight-Methode der Laufzeitmessung von Photonen verwendet. Für die Streckenbestimmung bzw. Abstandsbestimmung mittels Photonen wird die Lichtgeschwindigkeit als bestimmende physikalische Größe herangezogen. Da die Lichtgeschwindigkeit in einem Medium konstant ist, kann über die Laufzeitmessung der Photonen der Abstand mit Gleichung 2.2 berechnet werden. Die Lichtgeschwindigkeit beträgt im Vakuum 299792458 m/s [NIS16], bei einem Brechungsindex der Normatmosphäre (1013,25 mbar Umgebungsdruck), mit 20 °C, von 1,000272 [SZ01] ergibt sich mit Gleichung 2.1 [Mar09] eine Lichtgeschwindigkeit in Luft von 299710936 m/s. Bei allen Messungen in dieser Arbeit wurde diese Lichtgeschwindigkeit im Medium (c_m) angenommen.

$$c_m = \frac{1}{\sqrt{\epsilon_0 \epsilon \mu_0 \mu}} = \frac{c}{n} \quad (2.1)$$

c:	Lichtgeschwindigkeit	μ_0 :	magnetische Feldkonstante
ϵ_0 :	elektrische Feldkonstante	μ :	magnetische Permeabilität
ϵ :	elektrische Permittivität	n:	Brechungsindex Medium

Es gibt verschiedene Verfahren zur Bestimmung der Laufzeitmessung, welche prinzipiell auf zwei verschiedenen Methoden beruhen. Die direkte TOF-Methode misst die Laufzeit der Photonen. Beim indirekten TOF-Verfahren erfolgt die Messung über die Phasenverschiebung und daraus kann auf die Laufzeit der Photonen geschlossen werden. Hierbei wird ein periodisches Signal ausgesendet und mit dem gemessenen Signal verglichen. Im Folgenden wird auf die verschiedenen Methoden näher eingegangen.

2.2 Direktes Time-of-Flight

Die Grundidee zur Abstandsberechnung ist, die Zeit zu messen, welche ein Lichtpuls benötigt, um von einer Lichtquelle zu einer reflektierenden Oberfläche und wieder retour zu einem Detektor zu laufen. Grundsätzlich sind solche Aufbauten unterteilt, wie in Abbildung 2.1 gezeigt.

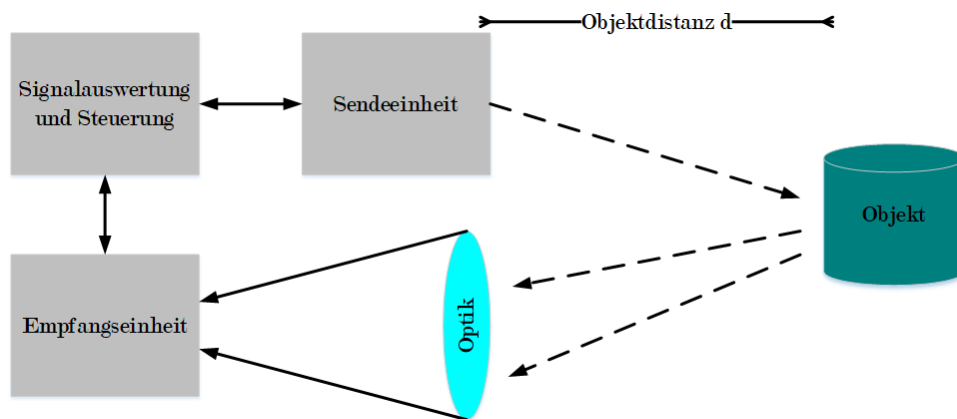


Abbildung 2.1: Blockschaltbild eines TOF-Sensors mit Sendeeinheit, Optik, Empfangseinheit und Signalauswertung.

Der Aufbau besteht aus einer Signalquelle wie einem Laser oder einer Leuchtdiode (LED), aus einer Optik zur Bündelung und Fokussierung des Lichtstrahls, einem Empfänger wie eine Photodiode, einen Lichtsensorchip oder einem Single-Photon-Avalanche-Diodes (SPAD)-Array. Zur Verarbeitung der Daten wird noch eine Steuer- und Auswerteelektronik benötigt. Aus der Zeitmessung erhält man die Zeit der Zwei-Wege-Strecke. Der Abstand kann über die Zeitmessung, wie in Gleichung 2.2 beschrieben direkt berechnet werden.

$$d = c \cdot \frac{t}{2} \quad (2.2)$$

- c : Lichtgeschwindigkeit
 t : Messzeit der Zwei-Wege Strecke
 d : Abstand zum Objekt

Beispiele für Abstandsmessung mit dieser Technik sind SLR (Satelite Laser Ranging) und LLR (Lunar Laser Ranging). Hier wird jeweils ein kurzer Laserpuls verwendet, um den Abstand zu einem Satelliten bzw. zum Mond zu ermitteln. Wenn der Laserpuls ausgesendet wird, startet die Laufzeitmessung. Der Laserpuls wird entweder an der Oberfläche des Objekts oder durch Spiegel reflektiert. Der reflektierte Strahl trifft dann mit geringerer Intensität auf einen Empfänger, welcher die Laufzeitmessung stoppt. Um genügend Intensität zu erhalten, wurde 1969 von der Apollo-11-Mission der erste Retroreflektor auf der Mondoberfläche positioniert [NAS69]. Um genaue Positionsdaten zu erhalten, gibt es einen internationalen Zusammenschluss, in welchem auch das Observatorium Lustbühel (Graz) vertreten ist. [Ser17]

Die Methode der direkten Flugzeitmessung wird bisher für die Feststellung weiter Entfernungen verwendet. Die limitierenden Faktoren sind hier die Steilheit der Flanke des Lichtpulses und die dazugehörige Zeitmessung. Dieses Verfahren der Pulszeitmessung wird direkte Flugzeitmessung (direct Time-of-Flight) genannt, da die Abstandsbestimmung über direkte Zeitmessung erfolgt. [Bog14]

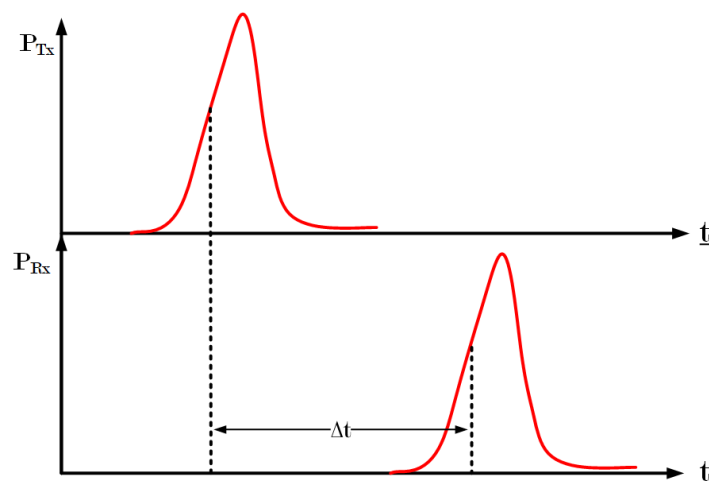


Abbildung 2.2: Illustration der Pulse und der Messzeit einer direkten TOF- Messung. P_{Tx} ist der Ausgangspuls und P_{Rx} ist der reflektierte Puls nach einer Zeit Δt . [Bog14]

Photonen legen bei Lichtgeschwindigkeit 1[m] in $3,3\text{ ns}$ zurück. Um diesen Zeitunterschied messen zu können, benötigt man mindestens eine Taktung von 333 MHz . Für einen Millimeter Auflösung benötigt man um eine drei Größenordnungen höhere Taktung. Hier zeigt sich die konventionelle Grenze dieser Technik.

In den letzten Jahren hat es auf dem Gebiet der direkten TOF einen Fortschritt bei der Messung kurzer Distanzen gegeben. Durch die Entwicklung und Verbesserung von SPADs bzw. durch den Zusammenschluss vieler SPADs zu einem Array ist es möglich, direktes TOF anzuwenden. Diese Technologie ist aus einem EU-Projekt entsprungen und konnte schon erfolgreich in ein reales Produkt für Punktmessungen von STMicroelectronics International N.V. implementiert werden. [Com15,STM14]

2.3 Indirektes Time-of-Flight

Eine weitere Möglichkeit, den Abstand zu ermitteln, ist die Messung der Phasenverschiebung zwischen empfangenem und gesendetem Signal. Hier wird nicht die Phase des Lichts, sondern die Phase eines modulierten Amplitudensignals ermittelt. Dieses Verfahren wird in der Literatur mit „indirekter Flugzeitmessung“ (indirect Time-of-Flight) bezeichnet, da hier die Laufzeit des Lichts indirekt über die Phasenverschiebung berechnet wird. Eine vollständige Periode des Intensitätsverlaufs des modulierten Lichts hat eine Wellenlänge von λ (siehe Gleichung 2.3):

$$\lambda = \frac{c_m}{f_m} \tag{2.3}$$

- c_m : Lichtgeschwindigkeit im Medium
- λ : Wellenlänge des modulierten Lichts
- f_m : Modulationsfrequenz

Der Vorteil dieser Technik im Gegensatz zur direkten TOF-Messung ist eine geringere Abtastrate für die Detektion des empfangenen Signals. Somit ist es möglich, die Abtastrate auf technisch machbare Frequenzen zu reduzieren. Um die Phasenverschiebung festzustellen, wird das Ausgangssignal mit dem Empfangssignal verglichen. Über die Phasenverschiebung kann dann mit Gleichung 2.5 der Abstand zum Messobjekt ermittelt werden. [CHHL12]

$$t = \frac{\varphi}{2\pi f_m} \quad (2.4)$$

Aus Gleichung 2.2 und 2.4 folgt:

$$d = \frac{c_m \cdot \varphi}{4\pi f_m} \quad (2.5)$$

t :	Messzeit	φ :	Phasenverschiebung
f_m :	Modulationsfrequenz	d :	Abstand zum Objekt
c_m :	Lichtgeschwindigkeit im Medium		

Der grundlegende Messaufbau ist gleich der direkten TOF-Methode, wie in Abbildung 2.1 gezeigt. Durch die Verwendung eines periodischen Signals ist es nicht möglich, über eine Wellenlänge hinaus den Abstand eindeutig zu bestimmen. Dieser Eindeutigkeitsbereich umfasst nur die Distanzen von 0 m bis zur Wellenlänge λ . Dieser Bereich ändert sich mit der eingestellten Modulationsfrequenz (siehe Gleichung 2.3) und der Berechnungsmethode. In den folgenden Abschnitten werden die verschiedenen Methoden zur Phasenbestimmung erläutert.

2.3.1 Phasenvergleich

Das Phasenvergleichen mit einer elektronischen Schaltung wird hauptsächlich in Entfernungsmessern für nicht-bildgebende Verfahren eingesetzt. Hier wird die Phasenverschiebung einer hohen Modulationsfrequenz im MHz Bereich auf eine niedrigere, leichter detektierbare Frequenz im kHz-Bereich transformiert. Die Arbeitsweise des Sensors beruht auf der Mischung von zwei Frequenzen, welche durch eine Mischfrequenz Δf getrennt sind. Ein Blockschaltbild des Phasenvergleichers ist in Abbildung 2.9 zu sehen. [uGS14]

Ein Oszillator steuert mit der Frequenz $f_0 + \frac{\Delta f}{2}$ eine Lichtquelle an, die Empfangseinheit detektiert das über die Laufzeit phasenverschobene Signal. Ein weiterer Oszillator mit der Frequenz $f_0 - \frac{\Delta f}{2}$ durchläuft keine Phasenverschiebung. Durch das Mischen der beiden Signale erhält man die in Gleichung 2.8 beschriebene Funktion. Mit einem Tiefpass wird der hochfrequente Anteil eliminiert. Daraus erhält man Gleichung 2.9, welche nur mehr den niederfrequenten Anteil Δf sowie die Information der Phasenverschiebung φ_0 beinhaltet. Beide Signale der Oszillatoren werden mit dem Signal aus dem zweiten Mischer

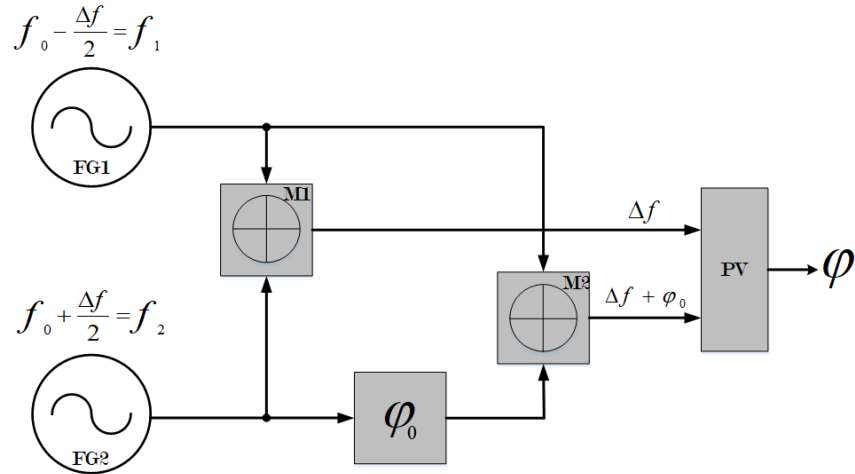


Abbildung 2.3: Blockschaltbild der Funktion eines Phasenvergleichers. FG1 und FG2 sind die Frequenzgeneratoren zur Erzeugung der Signale. M1 und M2 sind Mischer, welche die Signale verknüpfen. Die Phasenverschiebung durch Sendeeinheit, Wegstrecke und Empfangseinheit wird durch φ_0 und die ermittelte Phasenverschiebung mit φ dargestellt. [uGS14]

multipliziert, um ein nicht phasenverschobenes Vergleichssignal zu erhalten. Anschließend durchlaufen das phasenverschobene und das nicht phasenverschobene Signal einen Phasenvergleich, welcher im kHz-Bereich arbeitet. Die Ausgabe der Phaseninformation bzw. des Abstands erfolgt anschließend über eine Elektronik oder Software. [uGS14]

$$u_1(t) = u_a \cdot \sin \left[2\pi \left(f_0 - \frac{\Delta f}{2} \right) \cdot t - \frac{\varphi}{2} \right] \quad (2.6)$$

$$u_2(t) = u_b \cdot \sin \left[2\pi \left(f_0 + \frac{\Delta f}{2} \right) \cdot t + \frac{\varphi}{2} \right] \quad (2.7)$$

$$u_m(t) = \frac{u_a \cdot u_b}{2} \cdot [\cos(2\pi \cdot \Delta f \cdot t + \varphi) - \cos(4\pi \cdot f_0 \cdot t)] \quad (2.8)$$

$$u_m(t) = \frac{u_a \cdot u_b}{2} \cdot \cos(2\pi \cdot \Delta f \cdot t + \varphi) \quad (2.9)$$

φ :	Phasenverschiebung	t :	Modulationsfrequenz
f_0 :	Ausgangsfrequenz	Δf :	Mischfrequenz
u_a :	Signal aus Frequenzgenerator 1	u_b :	Signal aus Frequenzgenerator 2
u_m :	Amplitude nach Mischer		

2.3.2 Quadraturdemodulation

Bei der Quadraturdemodulation werden zwei phasenverschobene Signale erzeugt und verglichen. Die Modulationsfrequenz wird von einem Frequenzgenerator erzeugt und anders als bei dem Phasenvergleich einmal um 90° phasenverschoben. Das entsprechende Blockschaltbild ist in Abbildung 2.8 illustriert.

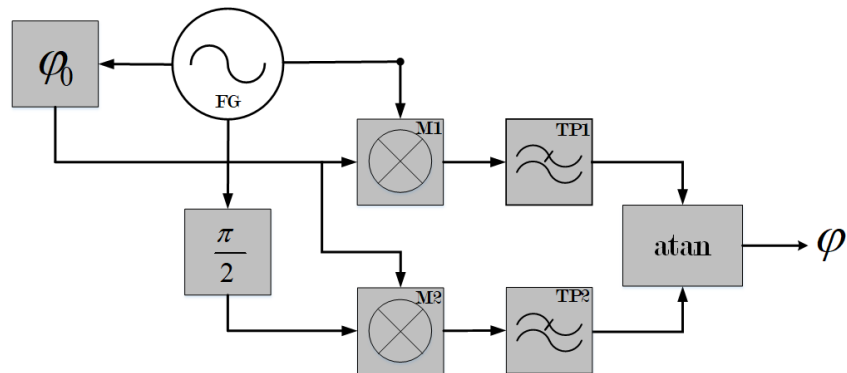


Abbildung 2.4: Blockschaltbild der Funktion der Quadraturdemodulation. Der Frequenzgenerator (FG) erzeugt ein Signal, welches mit einem Bauteil (hier $\frac{\pi}{2}$) um 90° phasenverschoben wird und zu einem Mischer (M2) gesendet wird. Das unverschobene Signal wird zum Mischer (M1) gesendet und andererseits erfährt das Signal über die Sendeeinheit, Wegstrecke und Empfangseinheit eine Phasenverschiebung, welche durch φ_0 dargestellt ist. Durch die Tiefpässe TP1 und TP2 werden die hochfrequenten Anteile herausgefiltert und man erhält die Gleichanteile, die zu $\sin(\varphi)$ und $\cos(\varphi)$ proportional sind (Gleichungen 2.10 und 2.11). Anschließend lässt sich die Phasenverschiebung mit einer Schaltung oder einer Software mittels Arkustangens berechnen. [Bog14]

Ein Frequenzgeber steuert die Sendeeinheit und einen Phasenschieber an. Das Empfangssignal, welches die Information der Laufzeit bereitstellt, wird jeweils mit dem Rohsignal und dem Ausgangssignal des Phasenschiebers multipliziert. Durch diese Vorgehensweise werden Signale generiert, welche die theoretische Form besitzen, wie in den Gleichungen 2.10 und 2.11 beschrieben. Diese Signale enthalten nun die Phasenverschiebung φ im Gleich- wie auch im Wechselanteil.

$$S_{T_x}(t) \cdot S_{R_x}(t) = \frac{1}{2} \cdot (\cos(\varphi) - \cos(4\pi f_m t + \varphi)) \quad (2.10)$$

$$S_{T_{90}}(t) \cdot S_{R_x}(t) = \frac{1}{2} \cdot (\sin(\varphi) - \sin(4\pi f_m t + \varphi)) \quad (2.11)$$

$$\arctan\left(\frac{\sin(\varphi)}{\cos(\varphi)}\right) = \varphi \quad (2.12)$$

$S_{T_x}(t)$:	Ausgangssignal	f_m :	Modulationsfrequenz
$S_{R_x}(t)$:	Empfangenes phasenverschobenes Signal	t :	Zeit
$S_{T_{90}}(t)$:	90° verschobenes Ausgangssignal	φ :	Phasenverschiebung

Durch die Anwendung eines Hochpassfilters können die periodischen Signale herausgefiltert werden. Nach dem Filtern bleibt nur noch eine Gleichspannung, welche dem Sinus und Cosinus der Phasenverschiebung entspricht. Mittels Quotientenbildung und mittels Arkustangens kann die Information der Phasenverschiebung berechnet werden, siehe Gleichung 2.12. Die Phasenverschiebung entspricht dem doppelten Abstand zum Messobjekt. Mit Gleichung 2.5 berechnet sich die Distanz vom Sensor bis zum Messobjekt. [Bog14]

2.3.3 Korrelationsabtastung

Eine weitere Möglichkeit besteht darin, am Detektor eine Abtastung des empfangenen Signals in äquidistanten Zeitabständen, welche einen Phasenunterschied von 90° zueinander aufweisen, vorzunehmen. Prinzipiell ist diese Technik an die Quadraturmodulation angelehnt. Hierbei gibt es zwei mögliche Abtasttechniken, das „2 Difference-Correlation-Sample“ (DCS2) und das „4 Difference-Correlation-Sample“ (DCS4). Hierbei wird die Korrelation von zwei oder vier verschiedenen gemessenen DCS-Signalen verwendet und mit Gleichung 2.13 oder 2.18, je nach verwendeter Methode, die Phasenverschiebung berechnet. [AG15]

Die aufeinanderfolgenden Messwerte werden relativ zueinander mit einer Phasenverschiebung von 90° aufgenommen. Durch diese Methode entsprechen bei einem periodischen Signal die Amplituden dem $\cos(\varphi)$ oder dem $\sin(\varphi)$ und somit dem Real- oder Imaginärteil. Durch diese Erkenntnis ist es möglich, wie bei der Quadraturdemodulation, die Phasenverschiebung mit Gleichung 2.12 zu bestimmen. Eine schematische Darstellung des Grundprinzips der Abtastung ist in Abbildung 2.5 ersichtlich. [CHHL12]

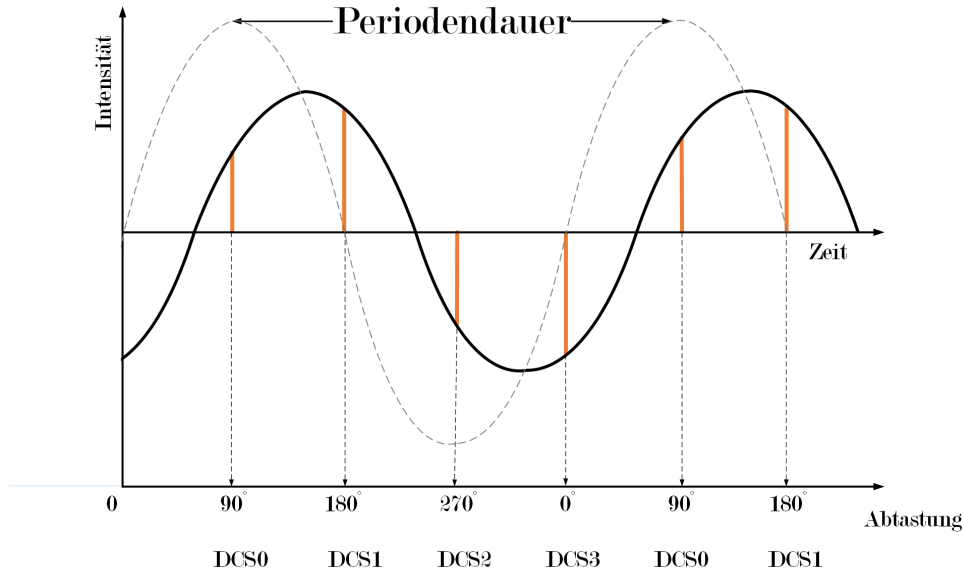


Abbildung 2.5: Darstellung des Grundprinzips der Abtastmethode mit 4 DCS-Werten. Die gestrichelte Linie symbolisiert das gesendete Signal und die durchgezogene Linie das empfangene Signal. [AG15]

$$\varphi = \arctan \left(\frac{\sin(180^\circ - \varphi)}{\sin(90^\circ - \varphi)} \right) = \arctan \left(\frac{\sin(\varphi)}{\cos(\varphi)} \right) = \arctan \left(\frac{D_1}{D_0} \right) \quad (2.13)$$

φ : Phasenverschiebung

D_x : DCS_x , Rohdaten der Berechnung für ein Tiefenbild

Es können nicht nur sinusförmige, sondern auch andere periodische Funktionen zur Modulation verwendet werden [FAT11]. Bei anderen Funktionen muss jedoch der Berechnungsalgorithmus angepasst oder im Nachhinein korrigiert werden, um Unsicherheiten zu verringern. Bei der Messung der Phasenverschiebung eines Rechtecksignals und der Auswertung mit Gleichung 2.5 tritt ein periodischer Fehler (Wiggling Error, Abschnitt 2.4) auf. Die Messung der Abtastwerte erfolgt nicht zu einem bestimmten Zeitpunkt, wie in Abbildung 2.5 gezeigt, sondern die Feststellung der Amplituden erfolgt über eine Integration. Die vier Gleichungen für die Integration von 2.14 bis 2.17 zeigen, über welchen Phasenwinkel integriert werden muss, um die entsprechenden Amplitudenwerte zu erhalten. [CHHL12]

Die Darstellung in Abbildung 2.10 zeigt die Steuersignale und die Ermittlung der vier unterschiedlichen Amplitudenwerte (Difference-Correlation-Sample 0-3 (DCS_x)). Die Steuersignale sind, wie durch die vier Integrationsgrenzen in Gleichungen 2.14 bis 2.17 vorge-

geben, phasenverschoben. Die Erfassung der vier Werte erfolgt nicht gleichzeitig, sondern wie in Abbildung 2.7 hintereinander und mit einer eingestellten Integrationszeit über mehrere Perioden. Wenn alle Amplitudenwerte erfasst sind, kann damit der Abstand mit Gleichung 2.13 oder 2.18 ermittelt werden.

$$\int_0^\pi \sin(x - \varphi) = \underbrace{-\cos(\pi - \varphi)}_{\cos(\varphi)} + \underbrace{\cos(-\varphi)}_{\cos(\varphi)} = 2 \cdot \cos(\varphi) = D_0 \quad (2.14)$$

$$\int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} \sin(x - \varphi) = \underbrace{-\cos\left(\frac{3\pi}{2} - \varphi\right)}_{\sin(\varphi)} + \underbrace{\cos\left(\frac{\pi}{2} - \varphi\right)}_{\sin(\varphi)} = 2 \cdot \sin(\varphi) = D_1 \quad (2.15)$$

$$\int_\pi^{2\pi} \sin(x - \varphi) = \underbrace{-\cos(2\pi - \varphi)}_{-\cos(\varphi)} + \underbrace{\cos(\pi - \varphi)}_{-\cos(\varphi)} = -2 \cdot \cos(\varphi) = D_2 \quad (2.16)$$

$$\int_{\frac{3\pi}{2}}^{\frac{5\pi}{2}} \sin(x - \varphi) = \underbrace{-\cos\left(\frac{5\pi}{2} - \varphi\right)}_{-\sin(\varphi)} + \underbrace{\cos\left(\frac{3\pi}{2} - \varphi\right)}_{-\sin(\varphi)} = -2 \cdot \sin(\varphi) = D_3 \quad (2.17)$$

φ : Phasenverschiebung

D_x : DCS0 bis DCS3

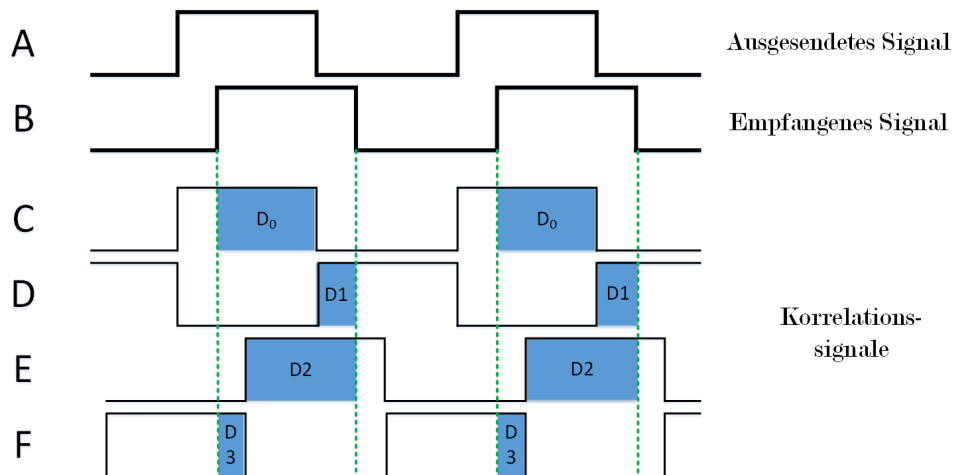


Abbildung 2.6: Darstellung des Grundprinzips der Messung aller vier DCS-Werte auf Basis von Gleichungen 2.14 bis 2.17. Zeile A stellt das ausgesendete Signal und Zeile B das empfangene Signal dar. Zeilen C bis F zeigen die Steuersignale und als eingefärbte Bereiche die erhaltenen Integrationswerte. [CHHL12]

Diese Methode ist derzeit Stand der Technik bei bildgebenden TOF-Verfahren. Herzstück dieses Verfahrens ist ein bildgebender Chip. In solch einem Sensor sind mehrere Bauteile wie Photodiode, Verstärker und Mischer für die Demodulation in einem Pixel vereint. Durch die kompakte Bauweise und den kürzeren Signalweg bis zur Zwischenspeicherung sind diese Elemente weniger störanfällig [Sch03].

Für jedes einzelne Pixel werden pro Zeitschritt die Difference-Correlation-Sample (DCS)-Daten akquiriert und über die eingestellte Integrationszeit (Zeit über mehrere Periodendauern) kumuliert [LSKK10].

Dies ergibt zwei oder vier DCS-Bilder bzw. Rohdatenmatrizen. Diese Matrizen enthalten die DCS_x-Werte und werden anschließend mit einer Software ausgewertet. Um die Phasenverschiebung, wie in Gleichung 2.13, zu berechnen, benötigt man mindestens zwei DCS-Graustufenbilder. Durch das DCS2-Verfahren kommt es aber zu größeren Pixelfehlern bzw. Auslesefehler können nicht ausgeglichen werden. Bei Aufnahme von vier DCS-Graustufenbildern und Berechnung der Phasenverschiebung mit 2.18 verringert sich die Bildrate und ein höherer Datenfluss stellt sich ein. Mit dieser Vorgehensweise können aber Auslesefehler reduziert und fixe Musterfehler (siehe Abschnitt 2.4 „Fixe Musterfehler“) kompensiert werden. [AG17]

$$\varphi = \arctan\left(\frac{D_3 - D_1}{D_2 - D_0}\right) \quad (2.18)$$

φ : Phasenverschiebung

D_x : DCS_x, Rohdaten für ein Tiefenbild

Durch die Verwendung des Arkustangens liegt der Grenzwinkel bei $\frac{\pi}{2}$ aufgrund der Mehrdeutigkeit des Ergebnisses. Mit Gleichung 2.2 berechnet sich dadurch der Eindeutigkeitsbereich auf ein Viertel der ursprünglichen Wellenlänge. Wenn die Vorzeichen der einzelnen Amplitudenwerte bekannt sind, ist es möglich, mittels Fallunterscheidung (Gleichung 2.19) den Eindeutigkeitsbereich voll auszuschöpfen. Die Fallunterscheidung entspricht der Funktion arctan2 und kann programmtechnisch leicht realisiert werden. Die unterschiedlichen Vorzeichen werden durch die Beschaltung des Analog-Digital-Converter (ADC) (zu Deutsch: Analog-Digital-Umsetzer) realisiert, dies wird in Abschnitt 3.2.1.1 genauer ausgeführt. [AG15]

$$f(x, y) = \arctan2(x, y) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{für } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{für } x < 0, y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{für } x < 0, y < 0 \\ \frac{\pi}{2} & \text{für } x = 0, y > 0 \\ \frac{\pi}{2} & \text{für } x = 0, y < 0 \\ 0 & \text{für } x = 0, y = 0 \end{cases} \quad (2.19)$$

2.3.4 Amplitudenqualität

Aus den Daten der Messung lässt sich die Amplitude des periodischen Signals bestimmen. Die Messung erfolgt digital, aus diesem Grund werden die Werte in der diskreten, dezimalen Einheit Least-Significant-Bit (LSB) angegeben, z. B. bei einer Abtasttiefe von 12 bit sind Werte zwischen 0–4096 LSB möglich. Die Berechnung der Amplitudenqualität erfolgt über den Betrag der gemessenen DCS_x-Werte und hängt von der Entfernung wie auch vom Reflexionsgrad des betrachteten Objekts ab. [FAT11]

Die Kalkulation der Amplitudenqualität ist in Gleichung 2.20 dargestellt.

$$A_{qu} = \sqrt{\left(\frac{D_0 - D_2}{2}\right)^2 + \left(\frac{D_1 - D_3}{2}\right)^2} = \frac{\sqrt{(D_0 - D_2)^2 + (D_1 - D_3)^2}}{2} \quad (2.20)$$

A_{qu} : Amplitudenqualität

D_x : DCS_x, Rohdaten der verschiedenen Korrelationsabtastungen für ein Tiefenbild

x : 0,1,2,3

2.3.5 Integrationszeit

Die Integrationszeit bestimmt, wie viele Perioden des modulierten Signals in einem Samplepunkt (DCS_x) zusammengefasst werden. Je länger die Integrationszeit, umso mehr statistische Unsicherheiten werden ausgeglichen. Die eingestellte Zeit beeinflusst jedoch die Bildrate. Für ein Tiefenbild werden zwei oder vier DCS-Bilder benötigt. Jedes Bild wird mit einer eingestellten Integrationszeit aufgenommen und anschließend ausgelesen. Die Dauer der aufeinanderfolgenden Integrations- und Ausleseschritte ergibt die Zeit für

ein Tiefenbild. Zur weiteren Erklärung ist in Abbildung 2.7 der Bildaufnahmeprozess illustriert. [AG15]

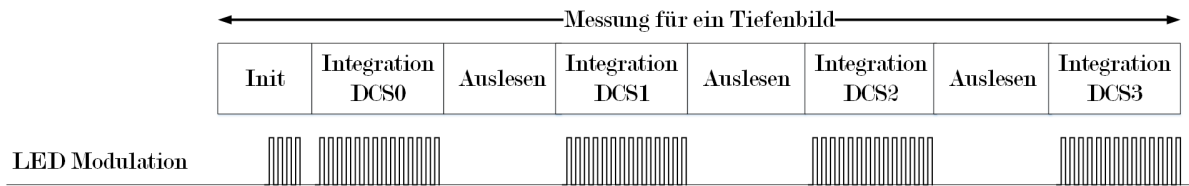


Abbildung 2.7: Schematische Darstellung der benötigten Zeit für die Erstellung eines Bildes zusammengesetzt aus Integrations- und Auslesezeit.

2.3.6 Konzept

Um eine bildgebende Tiefenmessung mittels indirektem TOF umsetzen zu können, benötigt das Messsystem einige Grundkomponenten. Das System besteht aus einer Lichtquelle, einer Optik, einem TOF-Sensor-Array in Form eines CCD-Pixelfelds und einer Auswerteeinheit für die Speicherung und Übertragung der Daten an eine Ausgabeschnittstelle. Das Funktionsprinzip dieses Aufbaus ist, dass eine Lichtquelle (Sendeeinheit) ein moduliertes Signal aussendet und dieses modulierte Licht am Objekt reflektiert wird. Durch die Optik wird auf einem Sensorchip in der Empfangseinheit ein Bild der Szene abgebildet und über die zuvor beschriebenen Auswertemöglichkeiten je Pixel ausgewertet. Durch die Erfassung der verschiedenen Amplitudenwerte lässt sich der Abstand berechnen.

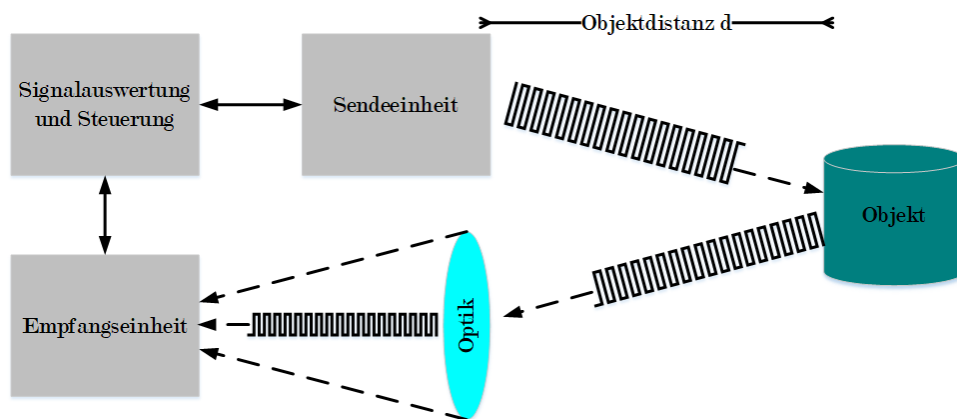


Abbildung 2.8: Blockschaltbild eines TOF-Sensors mit Sendeeinheit, Optik, Empfangseinheit und Signalauswertung.

Sendeeinheit: Die Lichtquelle muss die Modulation der Intensität des Lichts bewerkstelligen können. Die Modulationsfrequenzen, mit welchen die Lichtquelle betrieben werden kann, liegen derzeit bei 0,5 MHz bis 100 MHz für diese Methode. Diese Frequenzen entsprechen einer modulierten Wellenlänge von 600 m bis 3 m. Für diese Modulation ist es nicht notwendig, eine kohärente Lichtquelle zu verwenden. Der Emitter sollte aber nur einen Wellenlängenbereich ausstrahlen, um den Sensor und eventuell den Filter auf die Wellenlänge abstimmen zu können. In den meisten Fällen wird nahes Infrarot bzw. Infrarot verwendet. Diese Farben werden angewandt, da die Herstellung dieser Wellenlängen mittels Halbleitertechnik einfach ist und die ebenfalls dazugehörigen LEDs kostengünstig zu erhalten sind. Im Gegenstück dazu sind Halbleitersensoren für diese Wellenlängen sehr sensibel und es wird ein besseres Signal-Rausch-Verhältnis erwartet. [CHHL12]

Empfangseinheit: Als Sensor für die Tiefenmessungen wird ein Photonic-Mixing-Device (PMD) (zu Deutsch: Photomischdetektor) verwendet. Diese Technologie wurde 1998 von R. Schwarte eingeführt. Mit dem PMD ist es möglich, einen Photodetektor und einen elektronischen Mischer zu kombinieren, um somit Platz zu sparen und Weglängen der Signale zu minimieren. [GVHS01]

Das Grundprinzip ist gleich wie bei einem CCD-Sensor, wo die Elektronen-Loch-Paare durch den inneren Photoeffekt erzeugt werden und aufgrund des Potentialverlaufs gewährleistet wird, dass diese Paare nicht rekombinieren. (Wilson; Hawkes 1998 zitiert aus [Alb07])

Durch das Design des PMD werden die generierten Ladungen nicht nur auf einer, sondern auf zwei getrennten Kapazitäten gespeichert. Eine Mischfunktion steuert zwei Steuergates an, welche den Potenzialverlauf im PMD verändern können. Durch diese Änderung ist eine Trennung der generierten Elektronen zu den jeweiligen Kapazitäten möglich. (Xu; Zhanping; Schwarte 1998 zitiert aus [Alb07])

Je nachdem wie die Phasenlage des Modulationssignals ist, werden die Ladungen auf einem der beiden Kondensatoren summiert und mit einem ADC ausgelesen. Der ADC gibt anschließend die diskreten DCSx-Werte für jedes Pixel weiter. [AG15]

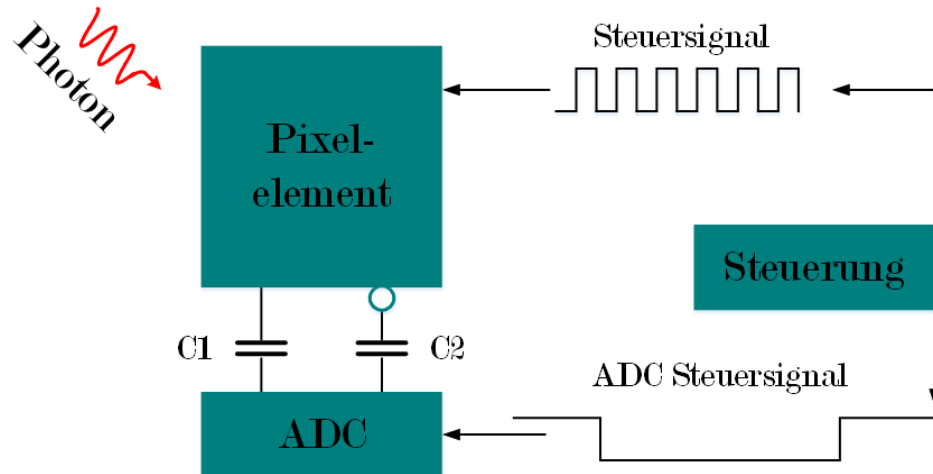


Abbildung 2.9: Bockschaltbild für die Ansteuerung und die Auswertung eines Pixels. Je nachdem zu welchem Zeitpunkt das Photon auf das Pixel trifft, wird die erzeugte Ladung im ersten Kondensator (C1) oder im zweiten Kondensator (C2) gespeichert. Ein ADC liest nach einer Integrationszeit die beiden Kondensatoren aus.

Optik: Die Optik muss je nach Fläche des Chips und dem Sichtfeld (Field-of-View (FOV)) dimensioniert werden. Es muss auch bedacht werden, dass die Transmission der verwendeten Wellenlänge gegeben ist. Um das Umgebungslicht zusätzlich zu unterdrücken, ist es möglich, einen Filter im optischen Weg einzubauen. Durch den Transmissionskoeffizienten des Filters wird aber auch das verwendete Signal für die Phasenerkennung abgeschwächt und es besteht eine Winkelabhängigkeit. Zusätzlich können durch den Filter Reflexionen zwischen diesem und dem Chip auftreten. Je nach Anwendung und Umgebungslicht muss die Verwendung von Frequenzfiltern überlegt werden. [AG15]

2.4 Fehler der bildgebenden Phasenmessung

Die Messfehler der bildgebenden Korrelationsabtastung (Abschnitt 2.3.3) sind auf verschiedene Arten von Fehlern zurückzuführen. Diese Unsicherheiten können in drei Gruppen von systematischen, statistischen und szenenbedingten Fehlern kategorisiert werden [LK07]. Um einen Überblick zu erhalten, werden im Nachfolgenden die verschiedenen Unsicherheiten der Profilmessung erörtert.

2.4.1 Statistische Fehler

Quantisierungsfehler: Der Erhalt von diskreten Werten ist eine Folge der Umwandlung eines analogen in ein digitales Signal. Je nach Bittiefe kommt es zu einem Rundungsfehler, welcher einen Informationsverlust bedeutet. Der Quantisierungsfehler zum analogen Signal berechnet sich aus dem minimalen und maximalen erfassbaren Wert des ADC, dies ergibt den minimalen, unterscheidbaren Wert. Die Messdaten, welche zur Verarbeitung für die Berechnung des Abstands bereitgestellt werden, sind Bitwerte (Einheit: LSB), welche proportional zu den Amplitudenwerten sind. Demnach ist der kleinste Fehler der Abtastung $0,5 \text{ LSB}$ [Sie16]. Durch die Quotientenbildung in Gleichung 2.13 bzw. Gleichung 2.18 und die Berechnung des Abstands mittels des Arkustangens hat das Ergebnis einen nicht-linearen Verlauf. Abbildung 2.10 zeigt den Verlauf des theoretischen Auflösungsvermögens mit einem idealen ADC bei 45° Phasenverschiebung und 24 MHz Modulationsfrequenz.

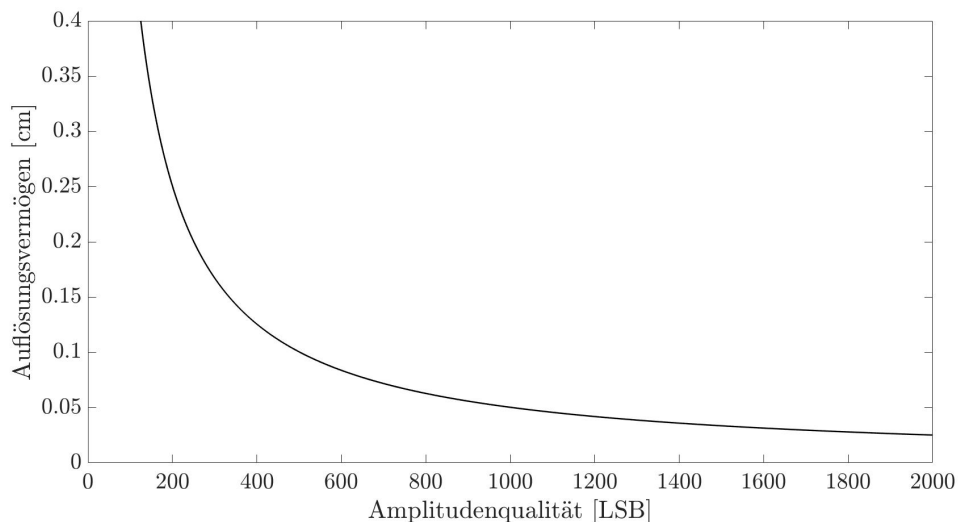


Abbildung 2.10: Simuliertes Auflösungsvermögen eines idealen ADCs. Je besser die Amplitudenqualität wird, umso genauer kann der Abstand bestimmt werden.

Bei einem idealen ADC ist der Fehler statistisch verteilt, demnach ist es möglich, mittels arithmetischen Mittelwerts einen guten Schätzwert zu erzielen [Gra11]. Dies wird über die Integrationszeit realisiert, welche bei diesem Verfahren einige hundert Mikrosekunden beträgt. Bei einer Modulationsfrequenz von 20 MHz hat das modulierte Signal eine Periodendauer von $0,04 \mu\text{s}$. Hierbei wird eine Stichprobe von 2500 Messungen bei einer Integrationszeit von $100 \mu\text{s}$ erreicht, somit wird der statistische Fehler gut ausgeglichen. Jedoch kommt es bei realen ADC zu weiteren Unsicherheiten, die nicht kompensiert werden können bzw. weiteren Informationsverlust bedeuten [Sie16].

Rauschquellen: Es gibt eine Vielzahl an Rauschquellen, die bei Halbleitersensoren und elektrischen Bauteilen auftreten können. Diese statistisch verteilten Fehler können verschiedene Ursachen haben. Albrecht hat in seiner Dissertation [Alb07] die Rauschquellen beschrieben, welche einen Einfluss auf die Messung mit einem PMD-Sensor haben. Das Photonenrauschen hängt von der einfallenden Lichtintensität ab und wird durch die statistischen Schwankungen beim Erzeugungsprozess von Elektronen-Loch-Paaren verursacht. Weitere Rauschquellen wie Dunkelstromrauschen, Reset-Rauschen und thermisches Rauschen hängen von der Temperatur ab. Alle diese Rauschquellen nehmen direkt Einfluss auf das Auflösungsvermögen der Phasenermittlung.

2.4.2 Systematische Fehler

Amplitudenfehler: Der Amplitudenfehler hängt mit der Intensität des reflektierten modulierten Lichts zusammen. Empirische Studien von Lindner und Kolb [LK07] haben ergeben, dass für die verwendete Wellenlänge weniger reflektierende, also dunkle Flächen bzw. Objekte, näher erscheinen als hellere Regionen (siehe Abbildung 2.11). Sie haben gezeigt, dass diese Phänomene meistens bei Messungen naher Distanzen auftreten. Um den Fehler in den Griff zu bekommen, konnten sie durch aufwendige Kalibrierung und Vergleich der Intensitätsinformationen je Pixel eine Reduktion dieses Fehlers erzielen.

Laut ihrer Erkenntnisse in einer weiteren Publikation ist der Ursprung dieser Unsicherheit noch immer nicht bekannt und hängt stark von der aufgenommenen Szene ab [LSKK10].

May und sein Team [MDH⁺09] haben die Vermutung, diesen Fehler auf die Charakteristik der Pixelelektronik zurückführen zu können. Begründet wird dies mit der nichtlinearen Ladung der Speicherkapazitäten und der Verstärkung dieser Information, bevor es zu einer Signalverarbeitung kommt.

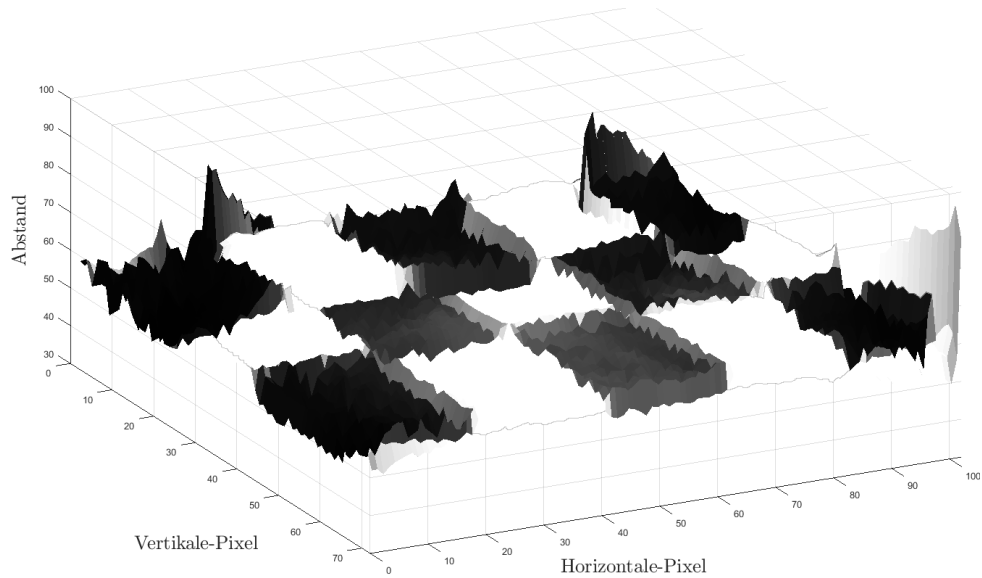


Abbildung 2.11: Aufnahme eines Tiefenbildes mit einer TOF-Kamera von einem Papier mit bedrucktem Schachmuster. Es ist zu erkennen, dass die schwarzen Flächen eine höhere Unsicherheit aufweisen.

Die Temperaturdrift: Die Temperaturabhängigkeit der verschiedenen elektronischen Komponenten beeinflusst die Messung und es kommt zu Abstandsänderungen während der Messung. Durch das Betreiben des Systems wird Leistung aufgenommen und die Bauteile erwärmen sich, dadurch verschieben sich die Arbeitspunkte der Elektronik und ein anderer DCS-Amplitudenwert wird gemessen. Der größte Effekt tritt während der Aufwärmphase auf, hier kann der Fehler mehrere Zentimeter betragen. [FPB⁺16]

Aus diesem Grund wird auch vom Hersteller empfohlen, vor dem Messen oder vor einer Nachkalibrierung des Messsystems dieses einer längeren Aufwärmphase zu unterziehen. Hier sollten der Chip und andere Bauteile in vollem Betrieb sein. [AG15]

In Abschnitt 4.5 sind die Auswirkungen der Temperaturdrift bei der Aufwärmphase des Sensors in einer Messung gezeigt.

Fixer Musterfehler (Fixed-Pattern-Noise (FPN)): Bei bildgebenden digitalen Aufnahmeverfahren tritt ein Offset für jedes einzelne Pixel auf. Ohne Nachbearbeitung kommt es zu einem fixen Bildrauschen. In der Bildverarbeitung wird dieser Fehler „Salz & Pfeffer“ genannt. Dieses fixe Rauschen bleibt größtenteils über die Zeit konstant und kann durch die Anwendung von Filtern reduziert werden. [MSMF14]

Der Ursprung liegt in den unterschiedlichen Laufzeiten, Verstärkungsfaktoren und Ausle-

sezeiten der einzelnen Pixel wie auch in den Offsets der einzelnen Schalttransistoren. Georgiev [GBG16] hat gezeigt, dass diese Unsicherheit mit der empfangenen Lichtintensität zusammenhängt und der Unterschied einige Zehntel Radiant bei der Phasenerkennung betragen kann.

Wiggling Error: Hier handelt es sich um einen methodischen, periodischen Fehler. Die Erzeugung des modulierten Lichts wird mit Rechteckschwingungen, durch Ein- und Ausschalten der LED, realisiert. Mittels Ausnützung der Tiefpass-Charakteristik der LED erhält das Signal eine sinusähnliche Form. Bei der Auswertung mit Gleichung 2.13 oder 2.18 ist aber ein idealer Sinus angenommen und es kommt zu einer periodischen Abweichung des gemessenen Abstands vom Realabstand [HKE14].

In Abbildung 2.12 ist eine Simulation des Wiggling-Errors zu sehen und diese zeigt die theoretische und simulierte Phasenverschiebung aus der Messung eines Rechtecksignals. Die Abweichungsberechnung des Abstands erfolgt mit Gleichung 2.5 und hängt von der Modulationsfrequenz ab. Bei einer Modulationsfrequenz von 24 MHz ergibt sich eine theoretische Abweichung zum Realabstand bis zu 7 cm.

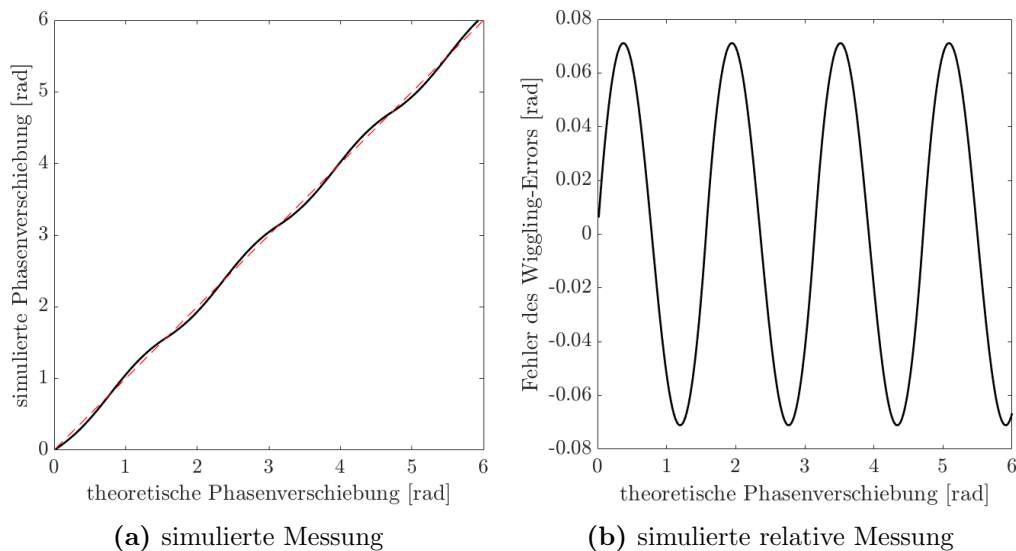


Abbildung 2.12: Die in (a) durchgezogene, schwarze Linie ist der theoretisch gemessene Wert bei einem Rechtecksignal. Die strichlierte Linie zeigt die theoretische Phasenverschiebung bei einem idealen Sinus-Modulationssignal. In (b) sind die beiden theoretischen Signale voneinander abgezogen. Je nach verwendeter Modulationsfrequenz ergibt sich ein anderer Abstandsfehler. Aus diesem Grund sind die Werte in Radiant [rad] angegeben.

2.4.3 Szenenbedingte Fehler

Mehrfachwege (Multiple Path): Mehrfachwege treten aufgrund der aktiven Szenenbeleuchtung mit dem modulierten Licht auf. Durch die Ausleuchtung einer Szene kann es vorkommen, dass das modulierte Licht von einem Objekt oder von Flächen wie Böden, Wänden und Decken nicht direkt zum Sensor reflektiert wird, sondern über Umwege zum Sensor gelangt, was einen größeren gemessenen Abstand bedeutet. Der Strahl, welcher den theoretisch direkten Weg zurücklegt (Abbildung 2.13 (e) und (f)), mischt sich mit dem indirekt reflektierten Strahl (Abbildung 2.13 (h)). In diesem Fall kommt es zu einer Überlagerung verschiedener Phaseninformationen und die zu messende Amplitude wird verfälscht. Somit kann die Phasenverschiebung nicht mehr eindeutig detektiert werden und es kommt zu einer Abweichung des berechneten Abstands zum Realabstand. Aus diesem Effekt resultieren auch fliegende Pixel, wobei hier nur einzelne Pixel betroffen sind (siehe nächster Absatz). [CHHL12]

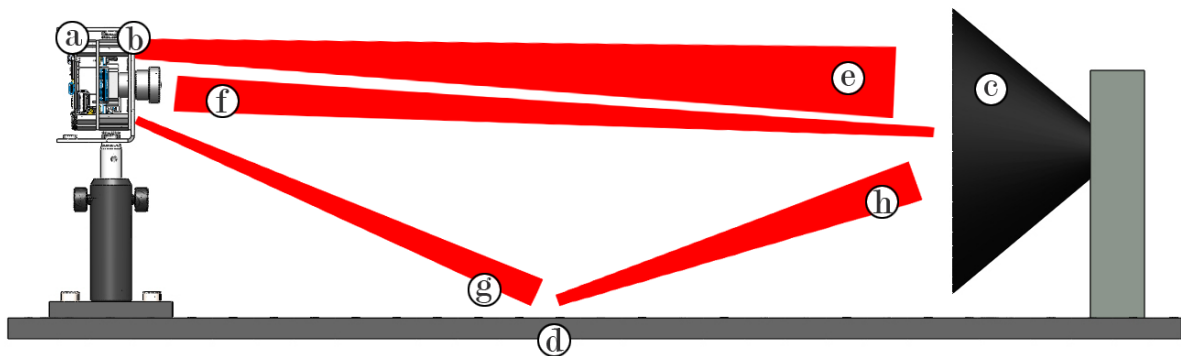


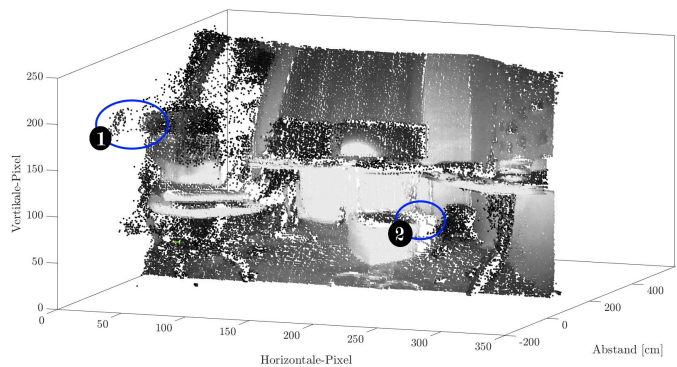
Abbildung 2.13: Darstellung der Mehrfachwege in einer Szene. (a) Kamera, (b) Lichtquelle, (c) Objekt, (d) reflektierende Oberfläche, (e) modulierte Licht zum Objekt, (f) modulierte Licht vom Objekt direkt zum Sensor der Kamera, (g) modulierte Licht zur reflektierenden Oberfläche, (h) modulierte Licht von der Oberfläche zum Objekt. In dieser Szene kann der Fall (e) zu (f) eintreten, hier wird der Abstand richtig interpretiert. Im zweiten Fall mischen sich (e) und (h), dabei wird Phaseninformation verfälscht und die Folge ist eine Abweichung des detektierten Abstands vom realen.

Fliegende Pixel (Flying Pixel): Fliegende Pixel sind Abstandserkennungen, welche scheinbar plötzlich einen nicht erwarteten Abstand interpretieren. Dieser Fehler kann verschiedene Ursachen haben. Zum einen tritt dieser Pixelfehler dann auf, wenn ein Bildpunkt verschiedene reale Abstände gleichzeitig misst. Dies geschieht zum Beispiel am Rand eines Objekts, das mit einem Abstand vor einem anderen Gegenstand steht. Hier mischen sich die Phaseninformationen von beiden Objekten. Das Resultat ist ein Abstand, welcher zwischen beiden Gegenständen liegt. Zum anderen tritt dieser Fehler an spiegelnden Oberflächen, zum Beispiel Glas, auf. [RDP⁺11]

Als Beispiel ist ein Arbeitsplatz mit Bildschirmen, Fensterscheibe, Sessel und Aufbewahrungsbox in Abbildung 2.14b zu sehen. Bei (1) treten fliegende Pixel aufgrund von Reflexionen an der Fensterscheibe auf und in (2) mischen sich Vordergrund und Hintergrund zu einem mittleren Abstandswert.



(a) Graustufenbild eines Arbeitsplatzes, erstellt aus den Informationen der Amplitudenqualität.



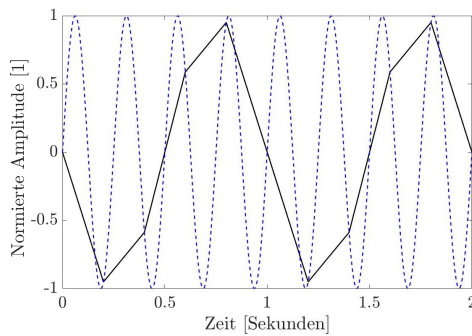
(b) Szenenbild Arbeitsplatz zur beispielhaften Darstellung von fliegenden Pixel.

Abbildung 2.14: Szenenbild Arbeitsplatz als 3D-Punktwolke (Point-Cloud) zur beispielhaften Darstellung von fliegenden Pixel. Bild (a) ist ein Graustufenbild der Szene, erstellt mit Hilfe der Amplitudeninformation. In Bild (b) sind die Effekte des Fehlers zu sehen: (1) fliegende Pixel aufgrund von spiegelnden Oberflächen (Glas), (2) Aufbewahrungsbox steht vor einem Hintergrund. Die Tiefeninformationen der Pixel mischen sich zu einem Mittelwert von Hintergrund und Objekt.

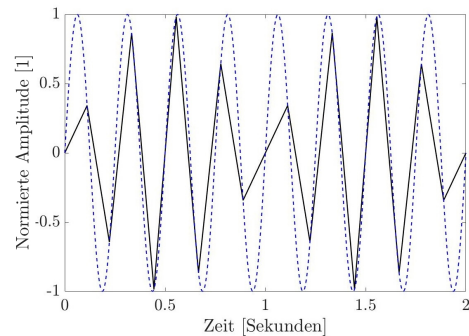
2.4.4 Fehler der Frequenzerfassung (Aliasing)

Der Alias-Effekt entsteht aufgrund der diskreten Abtastung und tritt auf, wenn in einem zu analysierenden Signal höhere Frequenzen vorkommen als die halbe Abtastrate. Es entsteht eine Unterabtastung. Die Folge ist ein irreversibler Informationsverlust. Die Über- und Unterabtastung wird durch das Abtasttheorem beschrieben. [RS14]

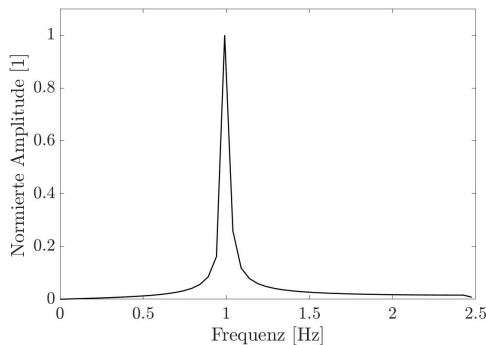
Bei der bildgebenden Vibrationsmessung handelt es sich bei der Abtastrate um die Bildrate (frame-per-second (fps)). Aufgrund des Abtasttheorems ist es nicht möglich, höhere Schwingungsfrequenzen als die halbe Bildrate zu erfassen. Eine Simulation des Alias-Effekts ist in Abbildung 2.15 dargestellt. Jeweils darunter ist das resultierende FFT-Spektrum dargestellt.



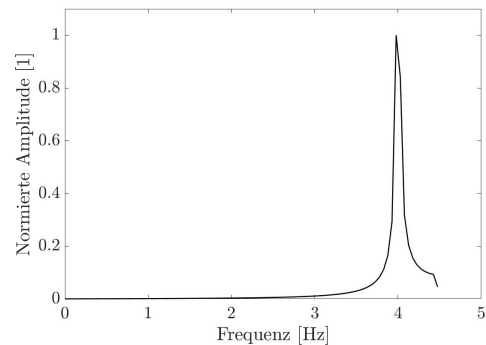
(a) Simuliertes (blau strichliert) Signal (4 Hz) und Abtastung (schwarze Linie, 5 fps).



(b) Simuliertes (blau strichliert) Signal (4 Hz) und Abtastung (schwarze Linie, 9 fps).



(c) FFT-Spektrum des Signals mit einer Abtastung von 5 fps.



(d) FFT-Spektrum des Signals mit einer Abtastung von 9 fps.

Abbildung 2.15: Die blau strichlierten Linien in (a) und (b) stellen das simulierte, zu messende Signal mit 4 Hz dar. Die Abtastrate in (a) mit 5 fps ist zu gering, um die 4 Hz zu detektieren. Hingegen kann das Signal in (b) mit 4 Hz bei einer Abtastrate von 9 fps sehr gut detektiert werden.

Realisierung

3.1 Auswahl des Sensors

Für die Untersuchung des Potentials der Vibrationsmessung mit der bildgebenden TOF-Technologie ist es erforderlich, einen entsprechend leistungsfähigen Chip zu beziehen. Die am freien Markt erhältlichen TOF-Chips der unterschiedlichen Hersteller weisen ähnliche Bild- sowie Tiefenaufösungen auf. Die Kriterien für eine möglichst gute Vibrationsmessung sind eine hohe Bildrate (fps) und eine genaue Abstandsmessung je Pixel. Einige Hersteller stellen für ihre entwickelten TOF-Chips ein Test-Set (Evaluations-Set, Eval-Board) in ihrem Portfolio zur Verfügung. Um einen entsprechenden Chip auszusuchen, wurden die Datenblätter der verfügbaren Sensoren bzw. Eval-Boards verglichen. In Tabelle 3.1 sind die Basisdaten der in Frage kommenden Chips aufgelistet.

Der TOF-Imager-Chip epc660 (epc660) eingebaut im Evaluations-Set Distance-Measurement-Engine (DME660) konnte eine vergleichbare oder bessere Bildauflösung sowie Bildrate bei der Gegenüberstellung der Konkurrenzprodukte aufweisen (siehe Tabelle 3.1). Zudem konnte der Hersteller (ESPROS-Photonics-Corporation (ESPROS), Sargans, Schweiz) durch eine kurze Lieferzeit überzeugen. Aus diesen Gründen wurde der epc660 mit dem Eval-Set DME660 für die bildgebende Messung ausgewählt und über den Distributor (Neumüller Elektronik GmbH (NEGmbH), Weisendorf, Deutschland) eine Bestellanforderung gestellt.

Tabelle 3.1: Vergleich der verschiedenen Evaluation-Sets anhand von Datenblättern. Die Kriterien sind eine hohe Bildrate und eine genaue Abstandsmessung.

F	Eval-Set	A	TA	Modf	fps	S
Melexis	EVK75123	320 x 240	3 mm	40 MHz	60	MATLAB SDK
ESPROS	epc660	320 x 240	2,5 mm	24 MHz	(65–1000*)	ROI / SDK
Mesa	SR4000/3000	176 x 144	4 mm	30 MHz	31	MATLAB API
PMD	Pico-Fam	352 x 287	1 %	•	60	MATLAB SDK
STM	VL53L0	Punkt	1 mm	direktes TOF	60	•

F:	Firma	fps:	Bildrate
Eval-Set:	Evaluation-Set	S:	Sonstiges
A:	Auflösung	SDK:	Software Development Kit
TA:	Tiefenauflösung	API:	Application Programming Interface
Modf:	Modulationsfrequenz	ROI:	Range of Interest (Bildbereich)
•:	keine Angabe		

* 1000 fps für einen verkleinerten Bildbereich möglich.

3.1.1 Weiterer TOF-Sensor

Ein weiterer TOF-Sensor wurde für Vergleichsmessungen von dem Chiphersteller STMicroelectronics International N.V. (STM)(Amsterdam, Niederlande) bezogen. Dieser TOF-Chip (VL53L0) besteht aus einem SPAD-Array und ist mit der patentierten FlightSense™-Technologie von STM ausgestattet [STM16]. Auf der Webseite des Herstellers wird FlightSense™ als ein direktes TOF-Verfahren beschrieben, wobei das Reflexionsvermögen des Objektes aufgrund der Pulsmessung keine Rolle spielt [STM14]. Das SPAD-Array ist bei diesem Chip nicht für die Bildgebung konzipiert, da er nur eine Punktmessung vornimmt. Die Technologie entspringt einem EU-Projekt, welches sich der Entwicklung eines größeren SPAD-Arrays (32 x 32 Pixel) und einer Bildrate von 10⁶ fps gewidmet hat [Com15, Cha10]. Die Vorteile der Messung mit dem VL53L0-Chip sind kleine Datenmengen und hohe Punktraten sowie direkter Zugriff der Daten über eine serielle Schnittstelle.

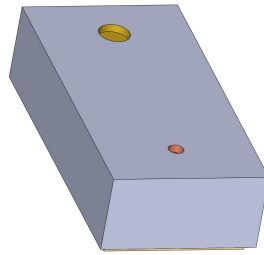


Abbildung 3.1: Schematische Darstellung des VL53L0X (konstruiert aus [STM16]). Dieser Chip hat eine Ausdehnung von 2,4 x 4,4 x 1,0 mm und wird als der kleinste TOF-Chip der Welt beworben [STM14].

3.2 Beschreibung des verwendeten Evaluation-Sets

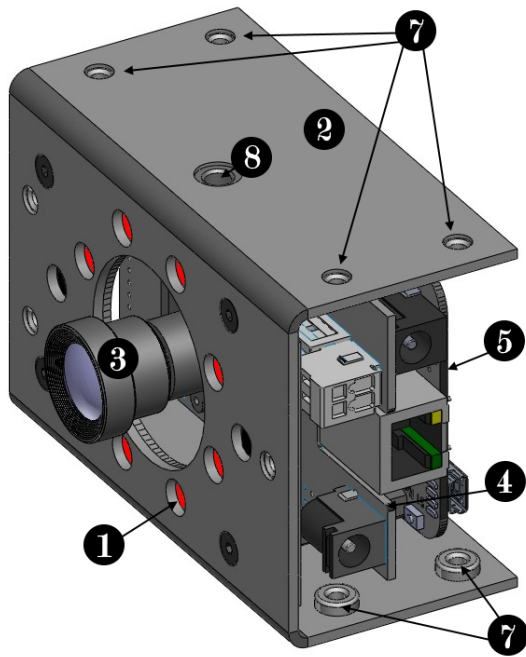
Das Evaluation-Set mit integriertem TOF-Chip epc660 wird von der Firma ESPROS geliefert und ist ein Komplettsystem für die Auswertung der Tiefendaten. Das komplette Set beinhaltet den epc660 Chip, eine Optik, eine Beleuchtungseinheit mit acht LEDs sowie eine Auswerteeinheit, welche die Daten an einen Computer sendet. Diese Bestandteile sind in einem Gehäuse verbaut. Im Folgenden wird dieses Gesamtsystem als DME660 bezeichnet. Ein 3D-Modell mit den Einzelbauteilen des DME660 ist in Abbildung 3.2 zu sehen [AG16]. Die Steuerung des DME660 erfolgt über eine USB-Schnittstelle mit einer Software (Graphical-User-Interface (GUI)), die vom Hersteller mit Quellcode bereitgestellt wird. Das Gehäuse verfügt über verschiedene Montagegewinde und kann auf einem handelsüblichen Kamerastativ (1/4 Zoll) montiert oder mit einem M4-Gewinde befestigt werden.

3.2.1 Hardware

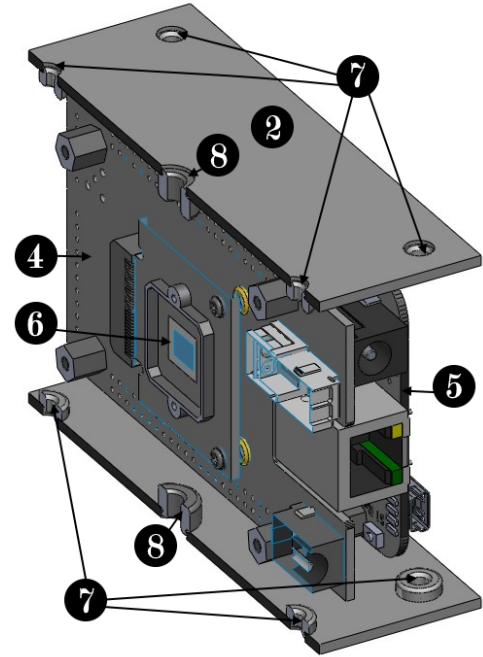
Die folgenden Ausführungen beschreiben die Hardware des DME660 und die möglichen Aufnahmeverfahren sowie die Beschaltung der Pixel. Die Informationen und die technischen Daten sind aus dem Datenblatt des epc660 [AG15] und des DME660 [AG17] entnommen.

3.2.1.1 Chip (epc660)

Der epc660 ist ein komplettes Tiefenmesssystem auf einer Platine (System-on-a-Chip (SoC)). Herzstück ist das Charge-Coupled-Device (CCD)-Pixelfeld mit einer effektiven



(a) DME660 in 3D Darstellung.



(b) 3D-Schnittzeichnung des DMA660, um die Position des Chips im Gehäuse zu sehen.

Abbildung 3.2: DME660 mit (1) LED, (2) Gehäuse, (3) Optik, (4) Chipboard, (5) BeagleBone Black (Einplatinencomputer), (6) Chip: epc660, (7) M4-Gewinde, (8) 1/4-Zoll-Gewinde. Die 3D-Daten wurden freundlicherweise von ESPROS zur Verfügung gestellt. [AG16]

Auflösung von 320 x 240 Pixel (Quarter-Video-Graphics Array (QVGA)), zusätzlich verfügt der epc660 über eine Kontrolllogik zur Steuerung und Auswertung der Daten. Ein Pixel hat eine Ausdehnung von $20\ \mu\text{m} \times 20\ \mu\text{m}$, das ergibt eine Größe des auslesbaren Pixelfelds von $6400\ \mu\text{m} \times 4800\ \mu\text{m}$, die Fehler werden im Datenblatt mit $1\ \mu\text{m}$ angegeben. Da die Sensorfläche noch zusätzliche Dummypixel beinhaltet, welche am Rand des aktiven Pixelfelds angeordnet sind, beträgt die tatsächliche Ausdehnung $6560\ \mu\text{m} \times 5040\ \mu\text{m}$. Eine schematische Darstellung des Pixelfelds ist in Abbildung 3.3 ersichtlich.

Das Pixelfeld ist aus Performancegründen in zwei Teile separiert und hat jeweils einen eigenen ADC für das Auslesen der Daten. Die beiden ADCs geben die Messwerte digital, mit 12 bit plus Sättigungsbit, für nicht interpretierbare Werte, aus (siehe Abbildung 3.3). Das Auslesen der Daten erfolgt Zeile für Zeile und der ADC liest parallel zuerst alle geraden und dann alle ungeraden Spalten aus. Dieses Ausleseverfahren wird für jede Sensorhälfte separat ausgeführt. Zudem sind die Pixel auch in Gruppen von je vier Pixel (2×2) unterteilt, um weitere Kombinationen der Pixelbeschaltung zu ermöglichen (siehe

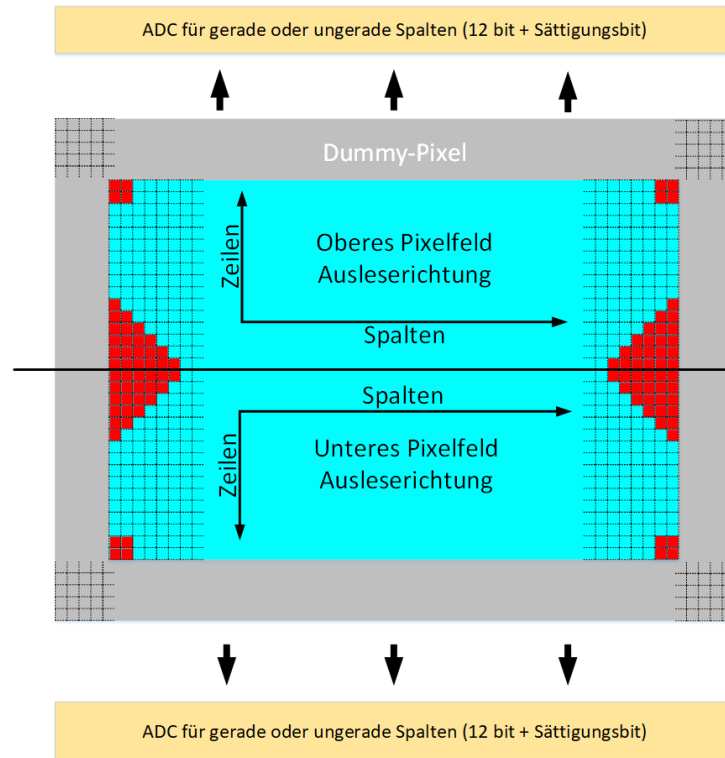


Abbildung 3.3: Pixelfeld des epc660 von ESPROS. Das effektive Messfeld hat eine Größe von 240×320 Pixel umgeben von Dummy-Bildpunkten. Der Sensor ist symmetrisch aufgebaut. Für jede Hälfte des Chips steht ein eigener ADC zur Verfügung, um die Datenrate bzw. Framerate zu erhöhen. Die Pixel sind zu einem Verbund von 2×2 Pixel zusammengeschlossen (symbolisiert durch die roten Quadrate an den Ecken des Sensors).

Abschnitt 3.2.1.2).

Jeder Bildpunkt hat zwei unterschiedliche Speicher, diese sind mit zwei verschiedenen Steuersignalen getriggert. Die jeweiligen Steuersignale sind um π zueinander phasenverschoben und je nach gewünschter DCS-Abtastung getriggert (siehe Abschnitt 2.3.3). Nach einer eingestellten Integrationszeit, welche proportional zur Anzahl der gemessenen Perioden ist, werden die Kapazitäten ausgelesen und die beiden Signale voneinander subtrahiert. Durch die Bildung der Differenz ist es möglich, Umgebungslicht zu unterdrücken und die unterschiedlichen Vorzeichen der DCS-Werte zu erhalten. Die Vorzeichen sind notwendig, um mit $\arctan 2$ (siehe Abschnitt 2.3.3 und Fallunterscheidung in Gleichung 2.19) über den gesamten 2π Bereich messen zu können. In Abbildung 3.4 ist eine schematische Darstellung der Funktion der Datenerfassung illustriert.

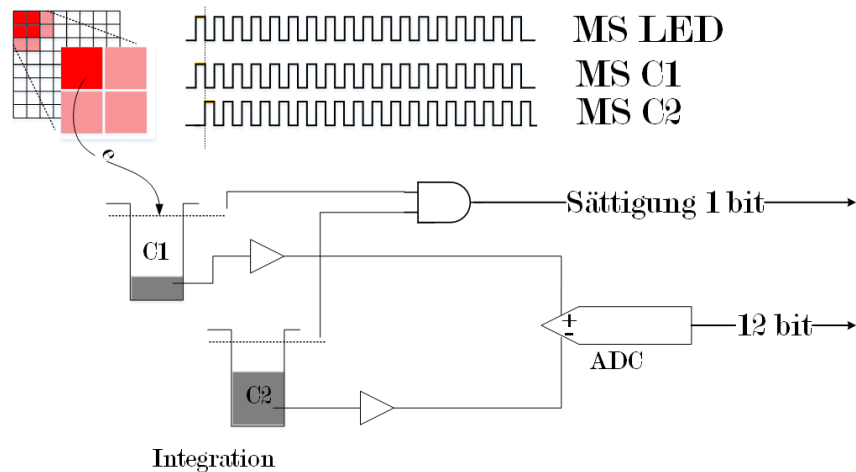


Abbildung 3.4: Schematische Darstellung der Datenerfassung, Umgebungslichtunterdrückung und Vorzeichengenerierung des epc660. MS LED: Modulationssignal, MS C1: Steuersignal für den Kondensator 1, MS C2: Steuersignal für den Kondensator 2

3.2.1.2 Steuerung der Pixel

In diesem Abschnitt werden die verschiedenen Ansteuerungsmöglichkeiten der Pixel des epc660 beschrieben. Diese Ausführungen sind aus dem Datenblatt [AG15] entnommen.

Standard-Mode: Im normalen Modus wird die Messung wie im vorherigen Abschnitt beschrieben durchgeführt.

Dual-Mode: Es besteht die Möglichkeit, jede zweite Zeile mit einem anderen Steuersignal anzusteuern. Dadurch erfolgt eine gleichzeitige Aufnahme von zwei unterschiedlichen Amplitudenwerten (DCS_0 und DCS_1 oder DCS_2 und DCS_3), welche für die Berechnung der Phasenverschiebung benötigt werden (siehe Gleichung 2.18). Die auf der Sensorfläche übereinanderliegenden Pixel werden nun für die Berechnung eines Abstandwertes herangezogen. Dies bedeutet eine Erhöhung der Tiefenbildrate, jedoch reduziert sich die effektive Auflösung auf 320×120 Pixel. Für ein besseres Verständnis ist die Methode in Abbildung 3.5 dargestellt.

HDR Mode: Im High-Dynamic-Range (HDR)-Modus wird die Steuerung der Zeilen ausgenutzt, um einen HDR-Effekt mit unterschiedlichen Belichtungszeiten zu erzielen. Um den HDR-Effekt zu realisieren, werden für einen Bildpunkt zwei übereinanderliegende Sensorelemente des (2×2) Pixel-Verbunds mit unterschiedlichen Integrationszeiten be-

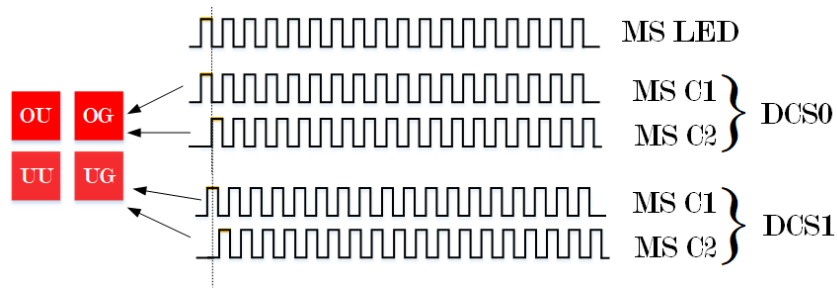


Abbildung 3.5: Datenerfassung Dual Mode im 2 x 2 Pixelverbund. OG: oben gerade, OU: oben ungerade, UU: unten ungerade, UG: unten gerade

trieben. Somit ist es möglich verschiedene Belichtungszeiten zu realisieren (Abbildung 3.6). Durch diese Vorgehensweise reduziert sich die effektive Bildauflösung auf 320 x 120 Pixel, jedoch die Bildrate verändert sich nicht.

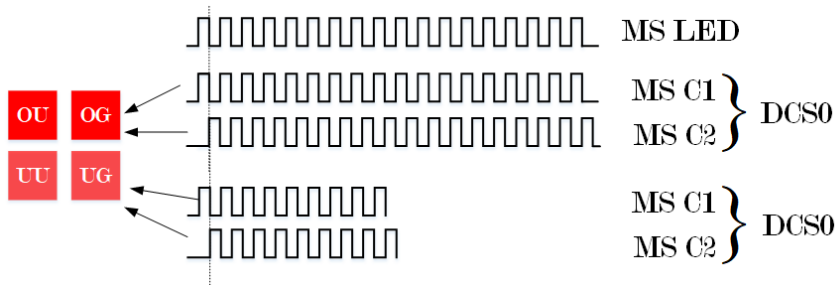


Abbildung 3.6: Datenerfassung im HDR-Modus. Jede zweite Zeile hat eine andere Integrationszeit, um unterschiedliche Belichtungszeiten zu generieren.

Pixel Binning: Wie bereits erwähnt, können die Pixel in einem (2 x 2)-Pixel-Verbund angesteuert werden (Abbildung 3.3). Beim Pixel Binning werden die Bildpunkte vertikal, horizontal oder der ganze Bereich zusammengeschlossen, um eine Vergrößerung der effektiven Pixelfläche zu ermöglichen. Mit dem Verbund von maximal 4 Pixel ist es möglich, mehr Ladungen zu akquirieren und somit die Integrationszeit zu verringern. Die Vorteile dieses Zusammenschlusses ergeben eine bessere Sensitivität und eine Steigerung der Bildrate. Das Pixel Binning reduziert aber je nach Einstellung die maximale Auflösung bis zu einem Viertel.

Bildausschnitt: Mit der Auswahl des Bildausschnitts (Range-of-Interest (ROI)) ist es möglich, nur einen Bereich des Pixelfeldes auszulesen. Hierbei verringern sich die Daten, welche verarbeitet werden müssen. Somit reduzieren sich auch der benötigte Pufferspeicher und die Prozesszeit. Daraus ergibt sich eine Erhöhung der Bildrate bei gleichbleibender

Auflösung im ausgewählten Feld. Da der Chip symmetrisch aufgebaut ist, wird die Auswahl von der Mitte aus auf die andere Hälfte des Chips gespiegelt, dies ist in Abbildung 3.7 dargestellt.



Abbildung 3.7: Schematische Darstellung des Chips und Funktion des ROI. Die dunkleren Bereiche werden deaktiviert und liefern keine Informationen. Die im oberen Bildbereich rot umrahmte Fläche (ROI) wird ausgewählt und die sich im unteren Bildbereich befindende Fläche (ROI symmetrisch) wird automatisch gespiegelt. Die gesamte hellblaue Fläche liefert dann Messdaten für die Berechnung der Tiefenbilder.

3.2.2 Lichtquelle

Als Lichtquelle werden acht LEDs verwendet, die auf einer Platine um die Linse angeordnet sind. Die Anordnung der LED ist in Abbildung 3.2 ersichtlich. Die Steuerung der LEDs wird direkt vom Chip übernommen. Für die Charakterisierung der Lichtquelle wurde einerseits das modulierte Signal bei 24 MHz betrachtet, andererseits aber auch das Spektrum der LEDs vermessen. Das modulierte Signal wurde mit einer Photodiode (Thorlabs Inc (Thorlabs) APD 120 A2\M, Avalanche Photodetector (200–1027) nm) vermessen und ist in Abbildung 3.8 zu sehen. Für die Messung ist der Photodetektor vor dem DME660 mit einem Abstand von $50,0 \text{ cm} \pm 0,5 \text{ cm}$ positioniert worden.

Die Peak-Wellenlänge des ausgestrahlten Lichts der LEDs beträgt nach Datenblatt [OSR16] 860 nm, mit einer Halbwertsbreite von 30 nm. Für die Darstellung des Spektrums in Abbildung 3.9 wurde eine Messung mit dem Spektrometer Flame-S von Ocean Optics mit einer Bandbreite von (200–1027) nm durchgeführt. Die Messung des höchsten Peaks ergab einen Wert von 856 nm mit einer Halbwertsbreite von 34 nm.

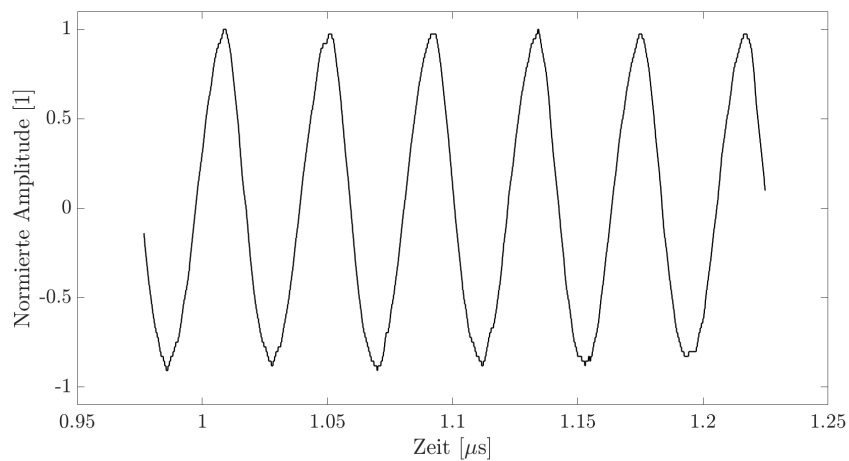


Abbildung 3.8: Normiertes modulierte Signal der LEDs. Gemessen mit einem Photodetektor (Thorlabs APD 120 A2\M, Avalanche Photodetector (200–1027) nm).

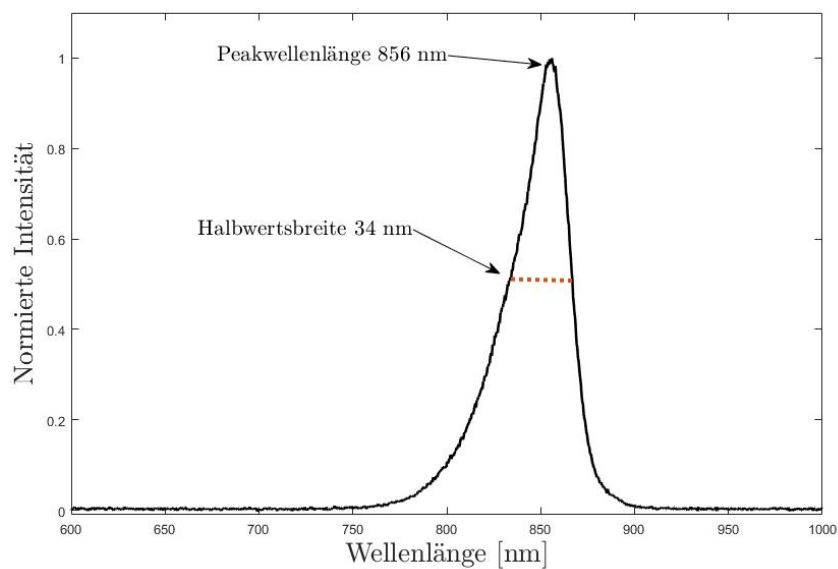


Abbildung 3.9: Normiertes Spektrum der Lichtquelle. Die Messung erfolgte mit dem Spektrometer Flame-S von Ocean Optics mit einer Bandbreite von (200–1027) nm und einer Auflösung von 0,1 nm.

3.2.3 Optik

Die Linse hat eine effektive Fokallänge von 4,1 mm, eine Blendenzahl von 2,0 und wird in ein M12 x 0,5-Gewinde eingeschraubt. Mit der Sensorgröße ergibt sich ein Sichtfeld (FOV) von 94° horizontal und 69° vertikal.

3.2.4 Kontrolleinheit

Bei der Kontrolleinheit handelt es sich um einen „BeagleBone Black“ Einplatinencomputer. Diese Kontrolleinheit verfügt über einen AM335x-Prozessor mit einer Taktung bis zu 1 GHz und einem RAM-Speicher von 512 MB DDR3. Die Kommunikation zum Computer ist entweder über Ethernet (100 Mbit/s) oder über eine USB 2.0-Schnittstelle (High Speed (480 Mbit/s)) möglich [AG15, Col13]. Bei den Messungen hat sich herausgestellt, dass diese Kontrolleinheit die Datenmengen des Chips nicht verarbeiten kann und es zu einer Reduktion der angegebenen Bildraten des TOF-Chips kommt.

3.2.5 Benötigte Aufnahmezeit pro Bild

Die Zeit, welche der Chip benötigt, um ein Bild auszulesen, beträgt laut Datenblatt [AG17] 3837 μ s. Dieser Wert setzt sich aus verschiedenen Faktoren zusammen. Zum einen schlagen sich die Auslesezeiten und Schaltzeiten auf die Aufnahmezeit nieder, zum anderen muss die eingestellte Integrationszeit berücksichtigt werden. Mit einer Integrationszeit von 1000 μ s berechnet sich die mögliche Bildrate für ein Tiefenbild mit dem DCS4-Verfahren auf ca. 51 fps. In Abbildung 3.10 ist der zeitliche Ablauf der Bilderfassung illustriert.

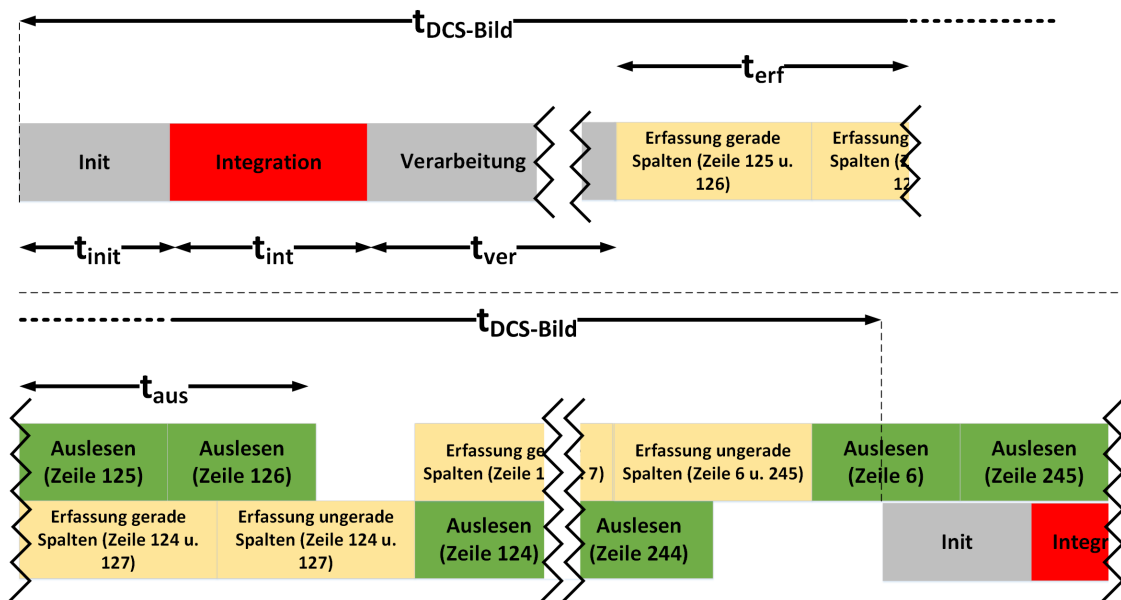


Abbildung 3.10: Schematische Darstellung der verschiedenen Prozesszeiten für die Aufnahme eines DCS-Bilds.

$t_{\text{DCS-Bild}}$:	Auslesezeit für ein DCS-Bild (3,81 ms)
t_{init} :	Schalt- und Initialisierungszeit (18 μs)
t_{int} :	Integrationszeit je nach Einstellung von 100 μs bis 2000 μs
t_{ver} :	Schaltzeit vom letzten LED-Puls bis zum Start der Datenerfassung (38,75 μs)
t_{erf} :	Zeit der Datenerfassung pro Zeile (31,25 μs)
t_{aus} :	Auslesezeit des ADC (32 μs)

3.2.6 Abschätzung der Datenmenge

Pro Pixel wird ein Datenstrang von 12 bit generiert, zusätzlich kommt ein Bit für die Sättigungsdetektion hinzu. Es entstehen also 13 bit pro Pixel. Die entstehende Datenmenge pro Bild entspricht dann der Anzahl der Pixel multipliziert mit 13 bit. Bei einer Auflösung von 328 x 252, inklusive Dummy-Pixel, benötigt der Speicher oder der Datenbus eine Kapazität von 1074528 bit pro Bild. Für eine Verarbeitung durch einem CPU muss der 13 bit Datenstrang in 16 bit bzw. 2 Byte (WORD) gepackt werden. Hierbei errechnet sich eine benötigte Speicherkapazität von 165 Kilobyte pro DCSx-Bild. Für 4 DCS-Bilder, also ein Tiefenbild, entspricht der benötigte Speicherplatz 660 Kilobyte. Um die entstehende Datenmenge bei einer Serienbildaufnahme von Tiefenbildern zu errechnen, ist der benötigte Speicher für ein Bild mit der Bildrate und der Aufnahmezeit zu multiplizieren. Der Hersteller empfiehlt im Datenblatt für die Abschätzung des Datenflusses und Dimensionierung der Speicher die doppelte Datenmenge je Bild zu berücksichtigen. Demzufolge ist bei einer Bildrate von 25 fps ein Datenaufkommen von 33 MB pro Sekunde zu erwarten. [AG15]

3.2.7 Software

Dieser Abschnitt ist aus dem Datenblatt des epc660 [AG15] und aus dem Benutzerhandbuch des DME660 [AG17] entnommen und beschreibt kurz das GUI und die verschiedenen Aufnahme- bzw. Ausgabemöglichkeiten.

GUI: Die Software (GUI) des DME660 kann die unterschiedlichen Parameter des Setups verändern, eine Neukalibrierung durchführen und die gemessenen Daten als Bild oder Rohdaten ausgeben. Die Speicherung kann optional in ein Comma-Separated-Values (CSV)-Format oder in einem Portable-Network-Graphics (PNG)-Bild erfolgen.

3.2.7.1 Mode

Diese Einstellung bestimmt, welche Art von Bild bzw. welche Daten mittels dem DME660 aufgenommen und dargestellt werden können. Im Folgenden sind die Auswahlmöglichkeiten beschrieben:

Grayscale: Grayscale nimmt ein Graustufenbild ohne Tiefeninformation auf und wird im Normalfall mit einer längeren Integrationszeit ausgeführt. Außerdem wird keine Modulation der Lichtquelle durchgeführt, die Lichtquelle ist permanent eingeschaltet. Bei der Speicherung der Daten über das GUI werden die Grauwerte in diskreten LSB Werten von 0 LSB bis 2047 LSB ausgegeben.

DCC: Distance color-coded Modus erstellt ein Bild mit eingefärbten Pixel, wobei sich die Farbe mit den entsprechenden Distanzen verändert. Zusätzlich ist es auch möglich, die Abstände in gerundeten Zentimeterschritten via einer CSV-Datei auszugeben.

ACC: Die Einstellung Amplitude color-coded speichert die Daten der Amplitudenqualität (siehe Abschnitt 2.3.4). Die Ausgabe kann entweder als Bild oder in einer CSV-Datei erfolgen.

HDR: Mit Amplitude HDR läuft der Sensor im MGX-Mode (siehe Abschnitt 3.2.1.2) und nimmt die Amplitudendaten auf.

Point cloud: Die Point cloud Option wird verwendet, um ein live 3D Bild zu erzeugen. Dieser Mode erzeugt eingefärbte Punkte, welche in einer dreidimensionalen Darstellung mit dem GUI ausgegeben werden.

DCS: Im DCS-Modus werden die DCS-Bilder 0 bis 3 ausgegeben. Mit diesen Informationen ist es anschließend möglich, mit Gleichungen 2.18 und 2.5 den Abstand für jedes Pixel zu berechnen. Im Weiteren werden diese Daten als Rohdaten bezeichnet. Jedes der Bilder hat Werte von (25 bis 2046) LSB mit positiven oder negativen Vorzeichen, somit ergibt sich eine absolute Samplingtiefe von 12 bit.

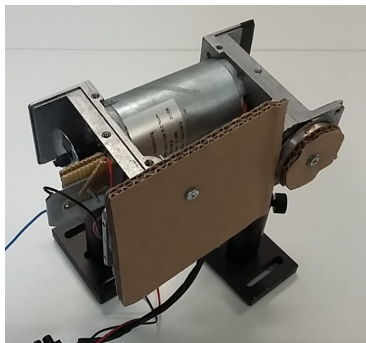
DCS differences: Der DCS differences Mode bildet die Differenzen des Divisors und des Dividenden der Gleichung 2.18 des DCS4 Aufnahmeverfahrens und gibt diese direkt in eine CSV-Datei aus. Durch diese Vorberechnung reduziert sich die Anzahl der Zeilen in der CSV-Datei und die Dateigröße verringert sich.

3.3 Aufbau

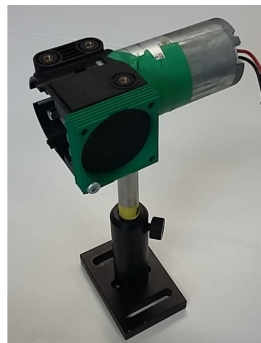
3.3.1 Vibrationsquellen

Zwei unterschiedlich adaptierte Membranpumpen, ein Lautsprecher und ein Automotor in der Startphase dienten als Vibrationsquellen.

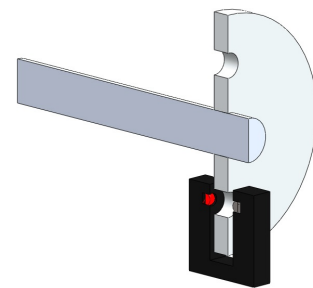
Membranpumpen: Bei den Membranpumpen wurden die Druckkammern geöffnet, um die Schwingungen der Membran detektieren zu können. Bei Pumpe 1 (Abbildung 3.11 (a)) wurde die Membran durch Metallkolben aus Aluminium ersetzt. Der Betrieb und die Drehzahländerung der Membranpumpen wurden mit einer regelbaren Gleichspannungsquelle realisiert. Um die Drehzahl bzw. die Frequenz der Schwingung vorab einstellen zu können, wurde eine Lichtschranke (EE-SX1042 von der Firma Omron Corporation) und eine Lochscheibe zur Signalunterbrechung verwendet. Die Lochscheibe mit zwei Bohrungen wurde an der Achse des Motors der Membranpumpe angebracht (Abbildung 3.11 (c)) und die Messung der Spannungsschwankungen, welche die Lichtschranke auslösen, erfolgte über ein Oszilloskop.



(a) Membranpumpe 1



(b) Membranpumpe 2

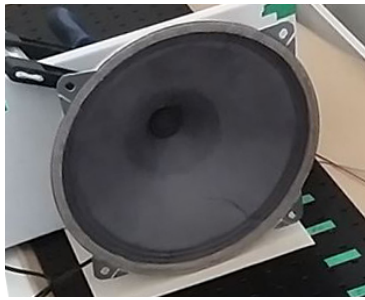


(c) Schnittdarstellung des Drehzahlmessers

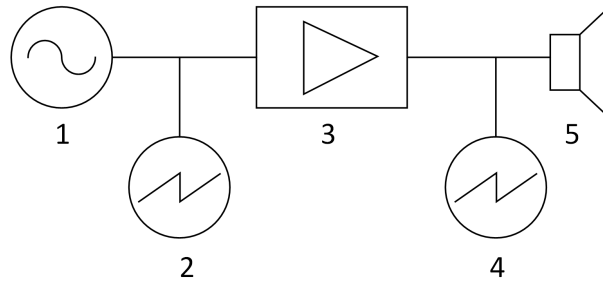
Abbildung 3.11: Bild (a) zeigt die Membranpumpe 1 mit Aluminiumscheibe als Kolben und montiertem Testmaterial (Karton). Der großflächige Karton dient der Verminderung von Reflexionen. Im Bild (b) ist die Membranpumpe 2 mit schwarzer Membran aus elastischem Kunststoff zu sehen. Darstellung (c) zeigt ein Schnittbild des Drehzahlmessers.

Lautsprecher: Ein handelsüblicher Lautsprecher wurde über einen Frequenzgenerator und einen Leistungsverstärker für Autolautsprecher betrieben (Sony XM-4020). Durch den Frequenzgenerator ist es möglich, Periodendauer und Auslenkung des schwingenden Lautsprechers zu steuern. Zur Überprüfung der eingestellten Frequenz wurde das Span-

nungssignal von einem Oszilloskop (Abbildung 3.12 (b), Bauteil (4)) aufgezeichnet.



(a) Lautsprecher



(b) Aufbau der Ansteuerung und Messung des Lautsprecherverhaltens

Abbildung 3.12: Bild (a) zeigt ein Foto des verwendeten Lautsprechers als Vibrationsquelle. Abbildung (b) zeigt die schematische Darstellung des Messaufbaus für die Messung höherer Frequenzen. Die eingesetzten Bauteile des Messaufbaus sind: (1) Frequenzgenerator, (2) Oszilloskop Kanal 1, (3) Audioverstärker, (4) Oszilloskop Kanal 2, (5) Lautsprecher

Automotor: Ein Automotor eines Audi 80 (Diesel) wurde durch Beeinflussung der Motorsteuerung für einige Sekunden im Startvorgang betrieben. Für die Einstellungen am Motor wurde Unterstützung von einem Mitarbeiter der Firma AVL DiTEST eingeholt. Die Überprüfung der Umdrehungszahl bzw. Schwingfrequenz erfolgte über eine Stromzange am Anlasser. Die gemessene Frequenz entspricht nicht direkt der Umdrehungszahl des Motors. Der Grund hierfür ist, dass ein Arbeitsspiel eines Zylinders eine Kompressionsphase durchläuft und zu diesem Zeitpunkt der Starter mehr Strom benötigt. Diese Stromschwankungen sind demnach proportional zur Umdrehungszahl. Ein Zylinder durchläuft bei einem Viertaktmotor 0,5 Arbeitsspiele pro Umdrehung. Demzufolge erhält man bei der Messung des Starterstroms eines Vierzylindermotors die doppelte Anzahl an Umdrehungen pro Minute [Bor16].

3.3.2 Laboraufbau

Das DME660 wurde auf einem schwarzen, nicht magnetischen Aluminiumboard (MB30120) von Thorlabs, welches für optische Anwendungen zur Reflexionsminimierung konzipiert ist, mittels zwei verstellbarer Halterungen (Thorlabs: HP75/M, TP75/M, BA15/M) montiert. Das zu messende Objekt wurde vor der Kamera platziert und die Messung über das GUI am Computer gestartet. Ein 3D-Modell des Grundaufbaus ist in Abbildung 3.13 zu sehen. Hier ist ein Lautsprecher als Symbolobjekt für eine Vibrationsquelle gewählt. Für die Kühlung bei längerem Betrieb und bei Temperaturversuchen wurde seitlich ein

handelsüblicher Computerlüfter mit einem Abstand von 5 cm angebracht. Mit einer verstellbaren Gleichspannungsquelle ist es möglich, den Lüfter zu betreiben und die Drehzahl und somit die Kühlleistung zu ändern.

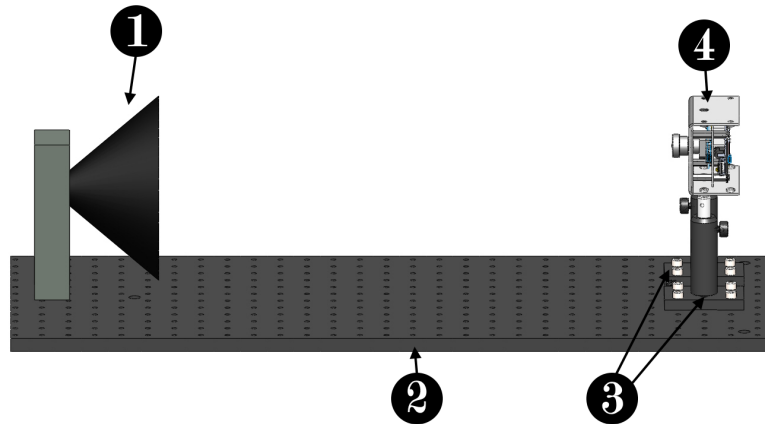


Abbildung 3.13: Grundaufbau der Messungen mit (1) Vibrationsquelle (Lautsprecher), (2) schwarze Befestigungsplatte, (3) Halterung, (4) DME660 (3D-Modelle wurden freundlicherweise von ESPROS [AG16] und Thorlabs [Tho17] zur Verfügung gestellt).

3.4 Analysealgorithmus

Der Analysealgorithmus ist in MATLAB verfasst worden und optimiert sowie verarbeitet die DCS-Daten für die Frequenzdetektion. Im Nachfolgenden wird beschrieben, wie die Daten des DME660 erfasst und ausgewertet werden, um die Frequenz aus diesen Informationen zu erhalten. Der Ablauf des Algorithmus ist in Abbildung 3.14 als Blockschaltbild dargestellt. Die einzelnen Programmteile werden in den nächsten Abschnitten detailliert erklärt.

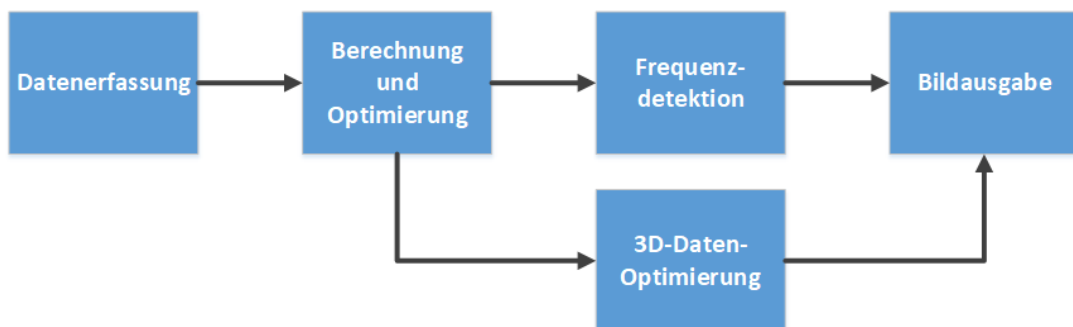


Abbildung 3.14: Blockschaltbild als Übersicht der Frequenzdetektions- und Bildverarbeitungs-Software.

3.4.1 Datenerfassung

Der Programmteil der Datenerfassung bereitet die Daten vor, um diese anschließend auswerten zu können. Das Blockschaltbild in Abbildung 3.15 zeigt die einzelnen Schritte für die Datenerfassung und die Datenvorbereitung.

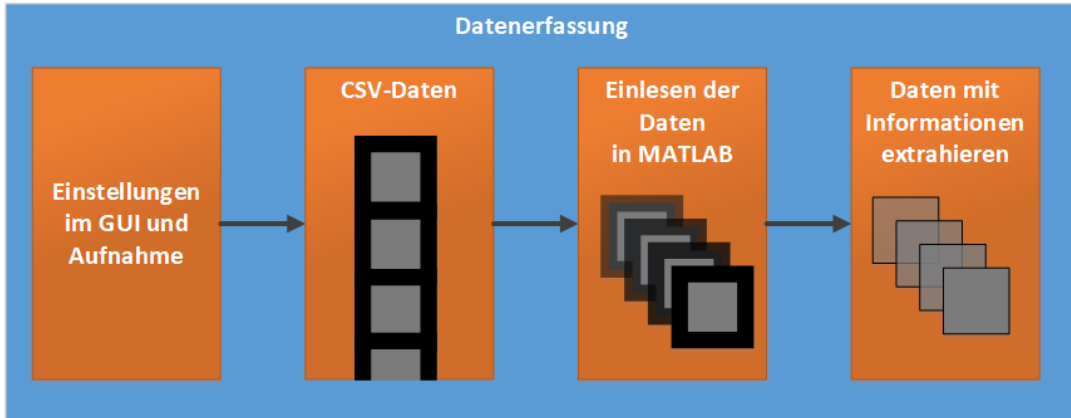


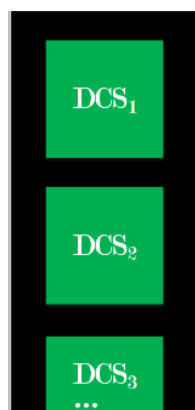
Abbildung 3.15: Blockschaltbild als Übersicht der Datenerfassung und -vorbereitung für die Weiterverarbeitung.

Einstellungen am GUI: Im ersten Schritt müssen die Integrationszeit, die Aufnahmemethode und die ROI über das GUI eingestellt werden. Um die optimale Integrationszeit zu erhalten, ist der gewünschte Detektionsbereich im Live-View zu betrachten und die Integrationszeit ist so lange zu reduzieren, bis keine gesättigten Pixel (Sättigungsbit) im ausgewählten Bildbereich vorkommen. Die Daten aus dem ADC sind diskrete Daten und werden als positive und negative Integer-Werte oder als undefinierte Datenwerte (Not-a-Number (NaN)) in einer CSV-Datei abgespeichert.

CSV-Daten: Wird als Ausgabeformat CSV gewählt, speichert das GUI alle DCS-Daten hintereinander in einer Datei ab (siehe Abbildung 3.16 (a)). In den Spalten befinden sich die horizontalen Pixel und in den Zeilen die vertikalen Pixel. Auch die Dummy-Pixel, welche am Rand des Chips angelegt sind, werden abgespeichert. Das GUI generiert demnach 328 Spalten, diese sind durch ein Komma (,) als Trennzeichen separiert. Die Zeilen sind durch einen Zeilenumbruch getrennt und variieren mit der Anzahl der aufgenommenen Bilder und der Aufnahmemethode. Werden 100 Bilder mit der DCS4-Methode (siehe Abschnitt 2.3.3) erstellt, ergibt sich eine Datei mit 328 Spalten und 100800 Zeilen. Durch das Speichern der Daten in ein CSV-Format, welches in American-Standard-

Code-for-Information-Interchange (ASCII)-Zeichen codiert ist, benötigt jedes Zeichen 8 bit [Mac79]. Bei einer durchschnittlichen LSB als dreistellige Dezimalzahl und mit dem Trennzeichen ergibt sich eine Dateigröße von 1322 Kilobyte pro Tiefenbild. Die Vorteile der CSV-Datei liegen bei der Lesbarkeit und guter Weiterverarbeitung mit verschiedenen Programmen sowie deren Verwendbarkeit auf unterschiedlichen Plattformen.

Einlesen der Daten in MATLAB: Die erste Programmfunktion ist dafür zuständig, die Daten in einer $328 \times 252 \times N$ Matrix zu erfassen und zu unterteilen. Das Einlesen der Daten wird über den Matlab-Befehl „load“ realisiert. Dieser speichert die LSB-Werte in einer Matrix (Anzahl an Spalten und Zeilen wie im vorherigen Absatz „CSV-Daten“ erläutert). Zur weiteren Verarbeitung ist diese Matrixform umständlich, da alle Bilder fortlaufend abgespeichert werden (Abbildung 3.16 (a)). Es ist sinnvoller, die einzelnen DCS-Bilder in einer eigenen Matrix zu speichern. Hierzu erfolgt eine Unterteilung in eine 3D-Matrix mit jeweils 328 Spalten 252 Zeilen und N Seiten (siehe Abbildung 3.16 (b)). Jede Matrix-Seite kann mit einer fortlaufenden Nummer angesteuert werden. Bei einer Aufnahme von 100 Tiefenbildern entspricht bei der DCS4-Methode (siehe Abschnitt 2.3.3) die erste Matrixseite den DCS₀-Informationen des ersten Bilds und die letzte, 400ste Matrixseite enthält die DCS₃-Informationen des hundertsten Tiefenbilds.



(a) Darstellung der abgespeicherten Daten in einer CSV-Datei.



(b) Darstellung der abgespeicherten Daten für die Weiterverarbeitung in unterschiedlichen 3D-Matrix-Seiten.

Abbildung 3.16: Bild (a) zeigt eine schematische Darstellung, wie die Daten in der CSV-Datei abgespeichert werden. Bild (b) zeigt die Daten für jedes Bild separiert in einer 3D-Matrix-Seite. In (a) sowie in (b) symbolisiert der schwarze Bereich die NaN's und der grüne Bereich die Daten mit Informationsgehalt.

Daten mit Informationsgehalt extrahieren: Jedes Bild ist von undefinierten Datenwerten (NaN's) umrahmt, diese Pixel enthalten keine Informationen. Die NaN's werden durch die Dummy-Pixel erzeugt oder rühren von der Auswahl eines Szenenbereichs (ROI) her. Die nutzbaren Informationen sind in jeder Matrix-Seite gleich angeordnet. Es wird mit einer Logikmatrix und dem Befehl „find“ nach dem ersten Integer-Wert und dem letzten Integer-Wert des Bildes gesucht. Diese beiden Matrix-Koordinaten spannen einen Bereich auf, in dem die nutzbaren Informationen enthalten sind. Mit diesem Bereich wird die gesamte 3D-Matrix, also alle Matrix-Seiten, zugeschnitten.

3.4.2 Berechnung und Optimierung

Die Rohdaten aus der Messung werden durch die vorhergehenden Programmfunktionen in einer 3D-Matrix gespeichert. In diesem Programmteil erfolgte die Erstellung eines Tiefenbildes und Reduzierung der Fehler. In Abbildung 3.18 ist die Abfolge der Programmfunktionen als Blockschaltbild dargestellt.

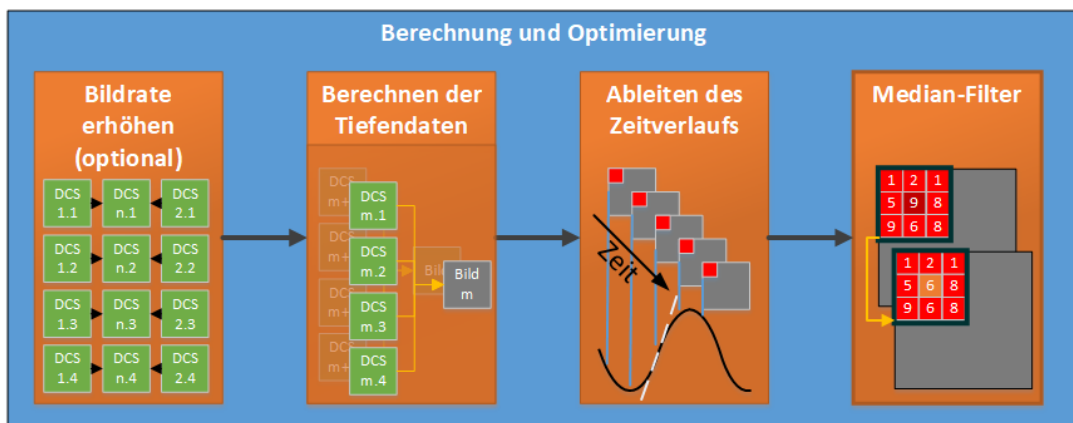


Abbildung 3.17: Blockschaltbild als Übersicht der Berechnung und Optimierung der Tiefendaten.

Bildrate erhöhen (optional): Die Idee der virtuellen Bildratenerhöhung ist, den Mittelwert aus zwei aufeinanderfolgenden DCS_x Bildern zu bilden und somit ein interpoliertes Bild zu erstellen, wobei $x = 0, 1, 2, 3$. Die Mittelwertbilder aus DCS_{x_n} und $DCS_{x_{n+1}}$ ergeben je ein neues DCS_{x_m} Bild (n beschreibt die Nummer des Tiefenbildes und m die Nummer des interpolierten Tiefenbildes). Diese Interpolation wird für jedes DCS_{x_m} Bild extra durchgeführt und in der 3D-Matrixabfolge in der entsprechenden Seite abgespeichert. Einen Informationsgewinn liefert dieser Vorgang zwar nicht, er dient aber dazu, einen virtuell größeren Frequenzbereich darstellen zu können. Dieser Programmteil ver-

steht sich auch als Platzhalter für Algorithmen, die mithilfe von Randbedingungen (z. B. zu untersuchenden Frequenzbereich) der Szene eine Verbesserung der Signalqualität liefern können.

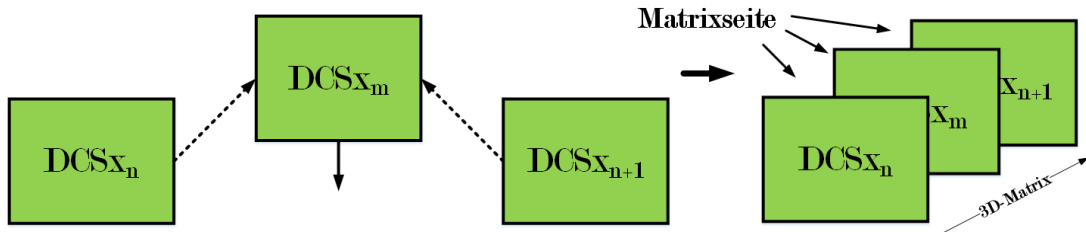


Abbildung 3.18: Schematische Darstellung der Erhöhung der Bildrate in der 3D-Matrix. ($x = 0, 1, 2, 3$ und n beschreibt die Nummer des Tiefenbildes sowie m die Nummer des interpolierten Tiefenbildes)

Berechnung der Tiefendaten und Amplitudenqualität: Um aus den gemessenen und interpolierten Rohdaten die Tiefenbilder zu ermitteln, wird Gleichung 2.5 für die Abstandsbestimmung verwendet. In dieser Gleichung gibt es nur eine Unbekannte. Die Lichtgeschwindigkeit ist definiert und die Modulationsfrequenz, mit welcher die Lichtquelle moduliert ist, wird vorgegeben. Demnach muss für die Bestimmung der Distanz nur die Phasenverschiebung φ , aus der Korrelationsabtastung (Abschnitt 2.3.3), berechnet werden. Mit den Gleichungen 2.18 oder 2.13, je nach gemessener Anzahl der DCS-Bilder pro Tiefenbild, berechnet sich die Phasenverschiebung φ . Die Kalkulation der Amplitudenqualität erfolgt über Gleichung 2.20. Die Rohdaten sind wie vorab beschrieben in einer 3D-Matrix abgespeichert und können seitenweise mit den Indizes $n \cdot 4 + 1 + x$ (n : Nummer des Tiefenbildes, $n = 0, 1, 2, 3, \dots$; x : DCS-Bildnummer, $x = 0, 1, 2, 3$) aufgerufen werden. Nach der Berechnung stehen zwei 3D-Matrizen zur Verfügung. Einmal jene Matrix mit den Tiefenbildern pro Zeiteinheit und einmal jene Matrix mit den Amplitudenqualitäten pro Zeiteinheit. Im Weiteren durchlaufen die beiden Matrizen die gleichen Bildbearbeitungsschritte, sind jedoch voneinander entkoppelt und können mit unterschiedlichen Parametern weiterverarbeitet werden.

Offset entfernen (Ableitungsmethode): Die berechneten Distanzwerte und die Amplitudenqualität unterliegen einem nichtlinearen Offset aufgrund von Temperaturdrift und nichtlinearen Effekten über die Zeit. Da bei der Frequenzdetektion bzw. Amplitudenbestimmung nur der relative Wert interessant ist, kann der Offset, den jeder Pixel und jedes Bild hat, entfernt werden. Um diesen Offset zu entfernen, ist es möglich, einen Fit anzu-

wenden und die erhaltenen Werte abzuziehen. Dieser Fit müsste jedoch für jedes Pixel ausgeführt werden. Aufgrund der geringen Performance ist dieses Verfahren nicht sinnvoll, stattdessen wird die diskrete, numerische Ableitung der Messwerte für jedes Pixel gebildet. Unter der Annahme, dass es sich bei der Messung einer Vibration um ein periodisches Signal handelt und die einzelnen Bilder in äquidistanten Schritten aufgenommen werden, ist es möglich, dieses Verfahren anzuwenden. Es fallen somit alle Offsets zwischen zwei aufeinanderfolgenden Tiefenbildern weg.

Median-Filter: Chan [CKTT08] konnte mit der Verwendung verschiedener Filter eine Rauschunterdrückung durchführen bzw. statistische Ausreißer glätten. Reynolds et al. [RDP⁺11] haben Filter für die Reduktion der „Flying Pixel“ angewandt und Swadzba et al. [SP07] kreierten einen variablen Filter, welcher sich auf Intensitätsänderung anpassen kann.

Die Idee zur Anwendung eines Median-Filters ist die Annahme, dass die nebeneinander angeordneten Pixel ähnliche reale Distanzen detektieren müssen. Der Median-Filter entfernt singuläre Störungen, aber Kanten bleiben erhalten. Bei der Anwendung des Median-Filters wird das statische Bild im Ortsraum zu einem Zeitpunkt betrachtet, nicht die zeitliche Veränderung. Für jedes Pixel wird ein eigener Wert mit Hilfe eines Filterkerns berechnet. Der Filterkern ist eine quadratische Matrix mit ungerader Anzahl an Spalten und Zeilen. Der Kern kann verschiedene Matrixgrößen besitzen und tastet die gesamte Informationsmatrix ab. [RS14]

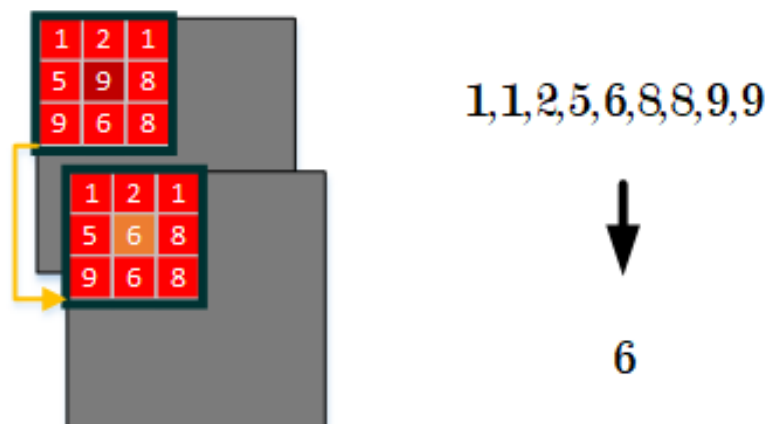


Abbildung 3.19: Schematische Darstellung der Funktion des Median-Filters. Der Median der Werte wird in dem zentralen Matrixfeld eingetragen.

Aus den erhaltenen Werten der 3D-Messung wird ein Median gebildet und dieser als

neuer Pixelwert gespeichert (siehe Abbildung 3.21). Diese Median-Bildung erfolgt über das ganze Bild und wird für alle einzelnen Tiefenbilder separat angewandt. Als Resultat wird das Bild von singulären Störungen einzelner Pixel befreit sowie das Bild geglättet. Somit verringert sich das Tiefenrauschen und jedes Tiefenbild liefert eindeutigere Werte für die Berechnung der Frequenz.

3.4.3 Frequenzdetektion

Mit der Information der Bildrate kann dann auf einen Zeitablauf geschlossen und eine Frequenz festgestellt werden. Der Ablauf der Programmfunktion zur Detektion von Frequenzen ist im Blockschaltbild in Abbildung 3.20 dargestellt.

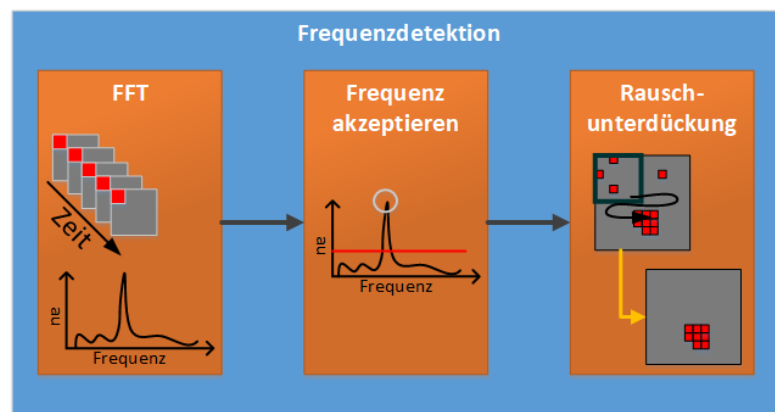


Abbildung 3.20: Blockschaltbild als Übersicht zur Frequenzdetektion.

FFT-Analyse: Die Detektion der Vibrationsfrequenz erfolgt über die Zeitdomäne und durchläuft eine Frequenzanalyse. Aus den 3D-Matrix-Seiten wird jede Pixel-Position zu einem zeitlichen Pixel-Vektor (Abbildung 3.21) zusammengefasst. Dieser Vektor enthält nun die Informationsänderung einer Pixel-Position über die Zeit und erfährt eine Fast-Fourier-Transform (FFT)-Transformation in die Frequenzdomäne. Hierzu muss der Pixel-Vektor mit der passenden Abtastrate an die FFT-Funktion übergeben werden. Das Ergebnis ist ein Frequenzspektrum für jede Pixel-Position. Aufgrund von statistischen Fehlern, welche noch nicht durch die vorhergehenden Schritte beseitigt wurden, können Frequenzen detektiert werden, obwohl real keine Schwingung gegeben ist.

Signal-Rausch-Analyse: Um zu erkennen, ob es sich bei der detektierten Frequenz um eine Vibration oder um Rauschen handelt, werden zwei weitere Schritte angewandt. Als

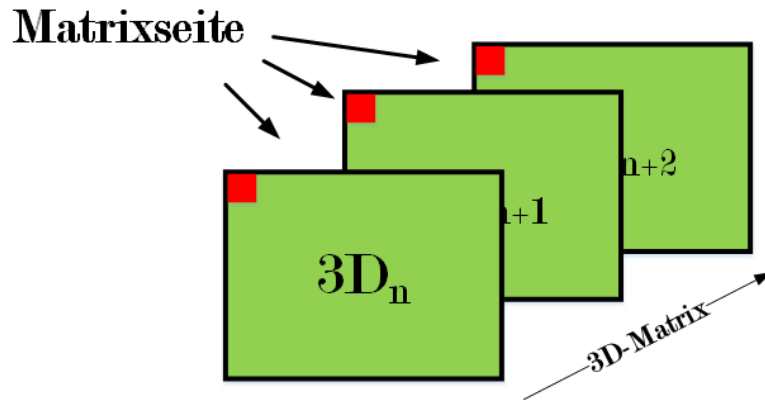


Abbildung 3.21: Schematische Darstellung der 3D-Matrix mit eingefärbten Pixeln (rot) (1,1) für jede Matrixseite. Die eingefärbten Pixel werden zeitlich versetzt aufgenommen und weiter mittels FFT-Analyse ausgewertet.

erster Schritt erfolgt die Berechnung des Signal-Rausch-Verhältnisses der einzelnen FFT-Spektren. Aufgrund der unterschiedlichen Signalqualitäten in Abhängigkeit des Abstandes und der Oberflächen ist es nicht möglich, einen eindeutigen Schwellenwert für das Rauschsignal anzugeben. Für die individuelle Berechnung des Signal-Rausch-Abstands von jedem Pixel ist zunächst der individuelle Rausch-Schwellenwert zu bestimmen. Die Bestimmung des Schwellenwertes erfolgt über den Mittelwert der detektierten Peaks des Frequenzspektrums. Das Signal-Rausch-Verhältnis berechnet sich dann mit dem dekadischen Logarithmus aus dem Verhältnis des gewählten Peaks zu dem Schwellenwert. Ein Funktionsparameter bestimmt, ab welchem Signal-Rausch-Verhältnis die Frequenz angenommen wird. Die Idee zur Ermittlung des Signalabstandes ist die Messung einer Szene, die Bereiche mit und ohne Schwingungen besitzt. Durch das Vergleichen der Ergebnisse konnte der Rauschabstand im Frequenzspektrum zu 4-6 dB bestimmt werden. Die Resultate der Messung sind in Abschnitt 4.2 ersichtlich.

Rauschunterdrückungs-Filter: Durch die statistischen Schwankungen pro Bild kommt es auch vor, dass bei einzelnen Pixel-Positionen fälschlicherweise trotz nicht vorhandener Vibration eine Frequenz akzeptiert wird. Um die Fehlinterpretation von singulären Frequenzunterschieden zu unterdrücken, wird ein Filter angewandt. Die Glättung funktioniert ähnlich wie beim Median-Filter. In der Kernmatrix werden die Anzahl der Nachbarwerte, die sich in einem Frequenz-Toleranzbereich befinden, gezählt. Die Annahme des Wertes erfolgt, wenn innerhalb der Kernmatrix ein gewisser Prozentsatz der detektierten Frequenzen in der frei wählbaren Toleranz liegen. Die Ausgabe dieser Funktion erfolgt über eine binäre Matrix, welche anschließend mit der Frequenzmatrix verknüpft wird.

3.4.4 3D-Bildoptimierung

Die Tiefendaten für die Darstellung eines 3D-Bildes werden gesondert optimiert, wie in Abbildung 3.14 schematisch gezeigt. Die Anwendung eines Median-Filters und die Mittelwertbildung von mehreren Tiefenbildern soll eine Verbesserung und Glättung der Abstandsermittlung herbeiführen, wie schon im Absatz 3.4.2 „Median-Filter“ beschrieben.

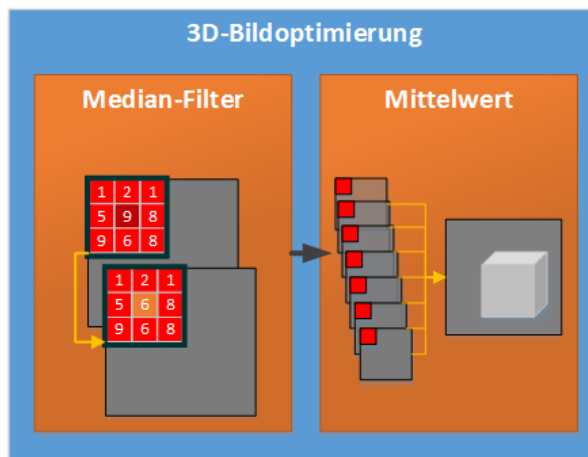


Abbildung 3.22: Blockschaltbild als Übersicht der 3D-Bildoptimierung.

Mittelwertbildung: Um nun die statistischen Fehler zu verringern und ein genaueres 3D-Bild zu erhalten, wird die Abstandsinformation je Pixel-Position über mehrere aufgenommene Tiefenbilder gemittelt.

3.4.5 Bildausgabe

Aus der Tiefeninformation oder aus der Amplitudenqualität ist es möglich, ein 2D-Graustufenbild zu erzeugen. Mit den Frequenzinformationen erfolgt danach eine Zuweisung einer Farbe zu einer Frequenz. Die 3D-Daten aus dem Programmteil „3D-Bildoptimierung“ werden mit dem eingefärbten 2D-Bild überlagert. Die schematische Abfolge ist in Bild 3.23 dargestellt.

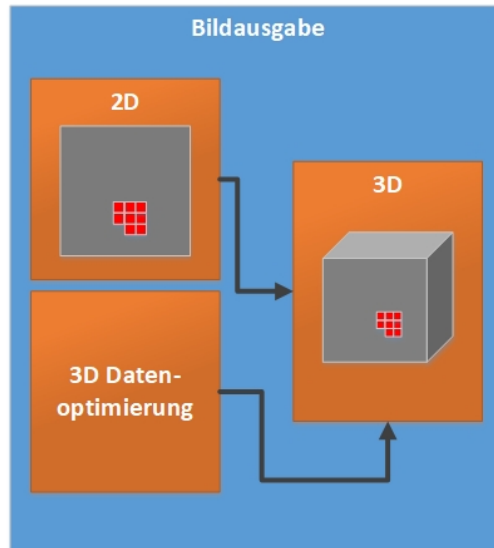


Abbildung 3.23: Blockschaltbild als Übersicht der Bilddarstellung.

2D-Bild: Für die Visualisierung der Frequenzen wird zunächst ein Graustufenbild mit den Informationen der Tiefe bzw. Amplitudenqualität erzeugt. Die abgespeicherten, detektierten Frequenzen erhalten dann über eine $n \times 3$ -Farbmatrix einen RGB-Wert. Diese erstellte Farbmatrix wird anschließend über das Graustufenbild gelegt. Durch diese Vorgehensweise entsteht ein Graustufenbild mit eingefärbten Pixeln, bei welchen eine Vibration detektiert wird.

3D-Bild: Aus den erhaltenen Informationen der 3D-Bildoptimierung und des 2D-Bilds können mit einem Surface-Plot oder einer Point-Cloud die dreidimensionalen Tiefendaten eingefärbt dargestellt werden. Somit sind die optimierten Tiefendaten mit den Frequenzinformationen in einem Bild vereint.

Resultate und Diskussion

4.1 Einfluss der Filtermethoden auf die Messung

Um zu zeigen, welchen Einfluss die Filtermethoden auf die Frequenzmatrix haben, ist eine Messung an einem handelsüblichen Lautsprecher durchgeführt worden. In Abbildung 4.1 ist der Aufbau der Messung dargestellt. Die FFT-Analyse gibt eine Matrix (Frequenzmatrix) mit den Frequenzen der einzelnen Pixel aus (Abbildung 4.2 (a)). Durch statistisches Rauschen im Signal werden auch Vibrationen detektiert, obwohl keine realen Schwingungen vorhanden sind. Um diese Fehler zu minimieren, gibt es zwei Programmfunktionen, die im Folgenden beschrieben und die Auswirkungen grafisch in Abbildung 4.2 dargestellt sind.

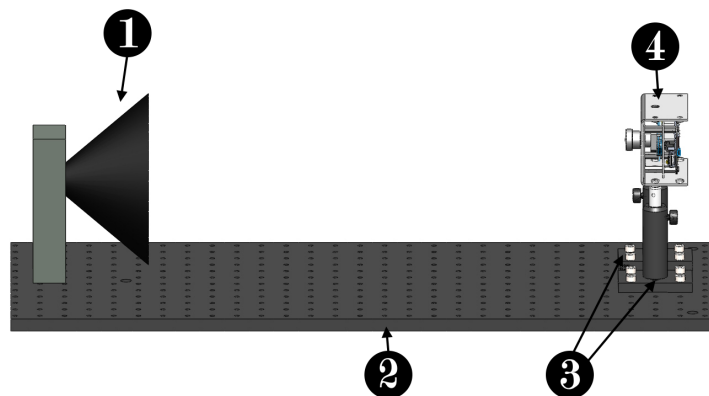
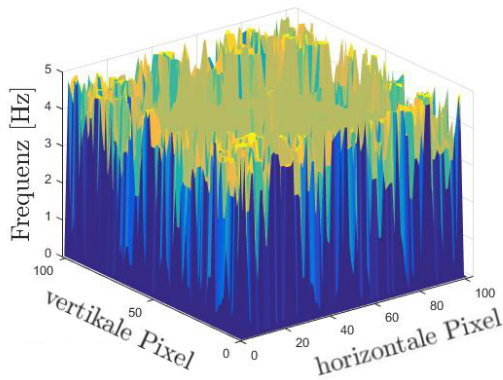


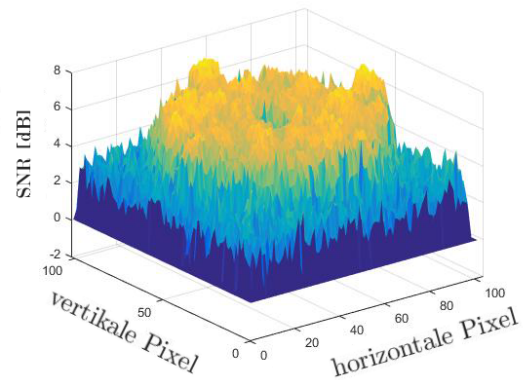
Abbildung 4.1: Grundaufbau der Messung mit (1) Vibrationsquelle (Lautsprecher), (2) Aluminiumboard, (3) Halterung, (4) DME660. Die 3D-Modelle wurden freundlicherweise von ESPROS [AG16] und Thorlabs. [Tho17] zur Verfügung gestellt.

Erkennung des Signal-Rausch-Abstands: Die Erkennung der Vibration und Annahme der Frequenz erfolgt über die Programmfunktion „Signal-Rausch-Analyse“ (siehe Abschnitt 3.4.3). Hierzu wird von jedem Pixel das Signal-to-Noise-Ratio (SNR) berechnet und in einer Matrix abgespeichert (Abbildung 4.2 (b)). Die Annahme einer Frequenz erfolgt ab einem gewissen Rauschabstand, welcher als Parameter der Funktion mitgegeben wird. Über eine erstellte Logikmatrix, wie in Abbildung 4.2 (c) gezeigt, werden anschließend die nicht angenommenen Werte aus der Frequenzmatrix entfernt.

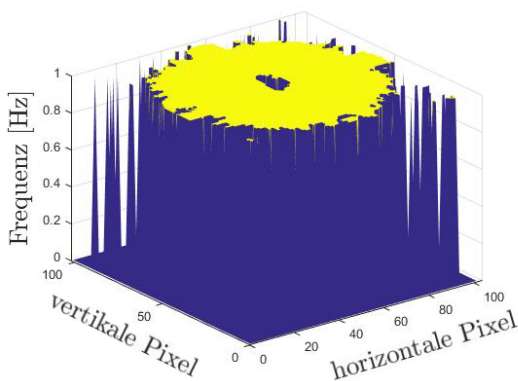
Rauschunterdrückung in der Frequenzmatrix: Nach der Annahme der Frequenzen mit der Berechnung des Signal-Rausch-Abstands können trotzdem noch vereinzelte Pixel-Positionen eine Vibration detektiert haben, der keine reale Schwingung zugrunde liegt. Um diese falsch erkannten Vibrationen zu verwerfen, ist ein weiterer Filter (Frequenzrausch-Filter) nötig, der in Abschnitt 3.4.3 „Rauschunterdrückung der Frequenzmatrix“ erläutert wird. Abbildung 4.2 (d) zeigt die erstellte Logikmatrix nach der Anwendung des Filters. Es ist zu erkennen, dass die singulären Ausreißer im Vergleich zu Abbildung 4.2 (c) nicht mehr angenommen werden. Die neu erstellte Logikmatrix wird anschließend mit der Frequenzmatrix verknüpft. Das Endresultat aus dieser Filtermaßnahme ist in Abbildung 4.2 (e) ersichtlich. Die darin enthaltenen Frequenzinformationen werden danach mit einer Farbe verknüpft und auf das erstellte 3D-Bild projiziert (Abbildung 4.2 (e)).



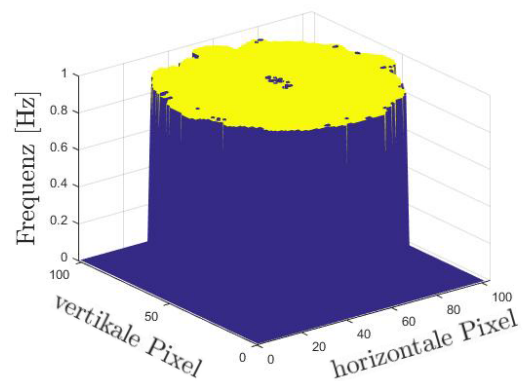
(a) Frequenzmatrix aus der FFT-Analyse.



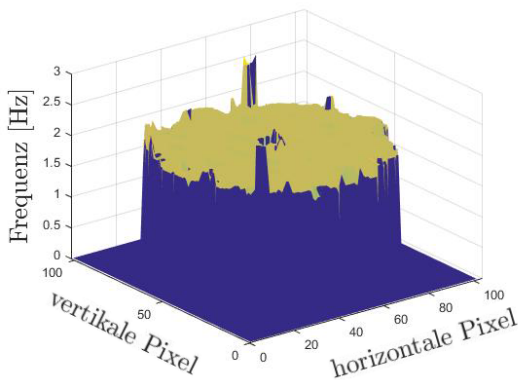
(b) SNR-Analyse



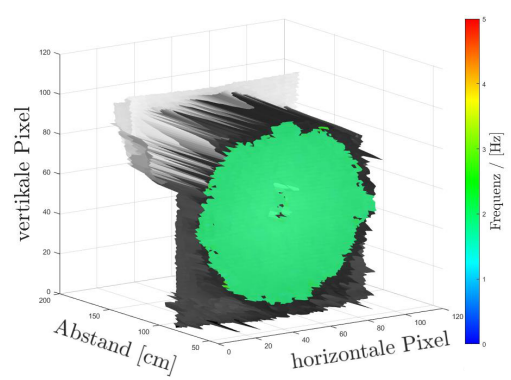
(c) Logikmatrix aus SNR-Analyse.



(d) Logikmatrix nach der Rauschunterdrückung.



(e) Logikmatrix (d) verknüpft mit der Frequenzmatrix.

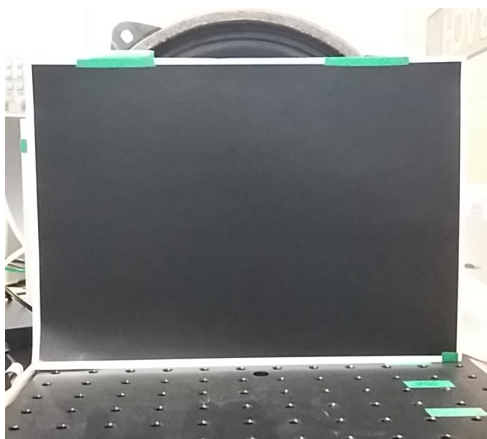


(f) 3D-Darstellung der Szene mit eingefärbten Frequenzen.

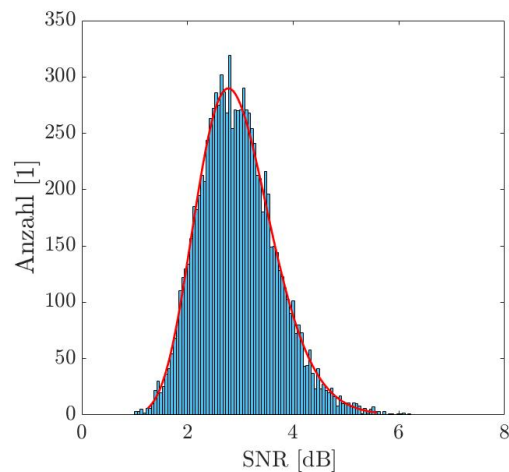
Abbildung 4.2: Auswirkungen der Filtermethoden nach der FFT-Analyse (Szene: Lautsprecher). Bild (a) zeigt die Frequenzmatrix mit den detektierten Frequenzen. Die dazugehörigen Signal-Rausch-Abstände sind in Bild (b) zu sehen. Mit dem Rausch-Schwellenwert wird aus Bild (b) die Logikmatrix (c) gebildet. Durch die Untersuchung der Frequenzmatrix können singular detektierte Frequenzen, deren keine realen Schwingungen zugrunde liegen, verworfen werden (d). Bild (e) zeigt die Frequenzmatrix verknüpft mit der Logikmatrix aus (d). Als finaler Schritt werden die detektierten Frequenzen, mit entsprechend zugeordneter Farbe, auf das 3D-Bild projiziert (f).

4.2 Schwellenwert für die Funktion „Signal-Rausch-Analyse“

Um den Signal-Rausch-Schwellenwert (siehe Abschnitt 3.4.3 „Signal-Rausch-Analyse“) zu ermitteln, wurde eine Nullmessung einer Szene ohne Vibrationsquelle bei Umgebungslicht durchgeführt. Als zu messendes Objekt diente ein schwarz bedrucktes A4 Blatt (siehe Abbildung 4.3 (a)). Die Programmfunktion „Signal-Rausch-Analyse“ (Abschnitt 3.4.3) berechnet die SNR-Abstände der Frequenzspektren und gibt diese in eine Matrix aus. Mit diesen Werten wird dann ein Histogramm erstellt. Das Histogramm soll darüber Aufschluss geben, bis zu welchem Schwellenwert es sich um Rauschen handelt. In Abbildung 4.3 (b) ist das aus der Hintergrundmessung resultierende Histogramm zu sehen. Ein Gammafit zeigt hier das beste Ergebnis, wie laut Henze [Hen11] aufgrund verschiedener Zufallsprozesse zu erwarten ist.



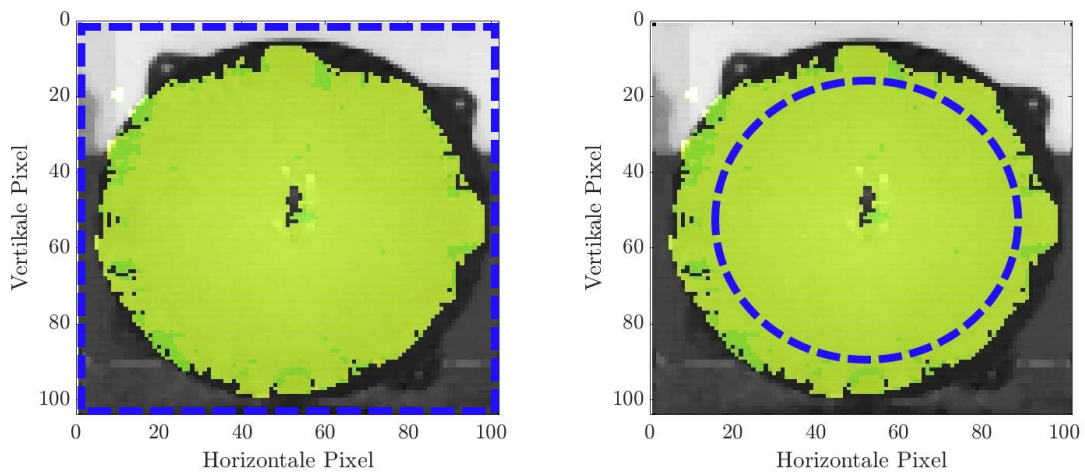
(a) Bild des Papiers mit schwarzem Aufdruck.



(b) Signal zu Rausch Histogramm der Hintergrundmessung.

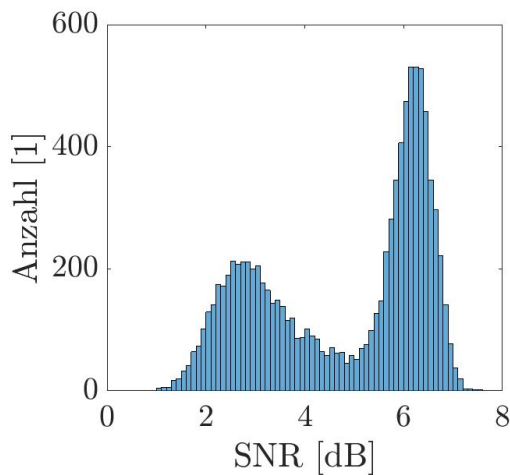
Abbildung 4.3: In Bild (a) ist das Messobjekt, ein A4-Papier mit schwarzem Aufdruck, abgebildet. Bild (b) stellt das daraus resultierende Histogramm mit einem Gammafit dar.

Die Histogramme (c) und (d) der Abbildung 4.4 zeigen die berechneten Werte der Signal-Rausch-Abstände der Frequenzspektren bei einer Szene mit Vibrationen. Als Vibrationsquelle dient der Lautsprecher bei $2,1 \text{ Hz} \pm 0,2 \text{ Hz}$ (siehe Abschnitt 4.7) und einer Auslenkung von $2,7 \text{ mm} \pm 0,4 \text{ mm}$ (siehe Abschnitt 4.3). Das Histogramm in (c) zeigt die ermittelten Signal-Rausch-Abstände des gesamten Bildes. Bild (d) stellt das Histogramm für den strichliert umrahmten Pixelbereich in (b) dar. Beim Vergleich der Histogramme ist zu sehen, dass eine bimodale Verteilung bei der Messung des gesamten Bildes entsteht und nur mehr ein Peak im ausgewählten Bereich, welche real Schwingungen aufweist, auftritt.

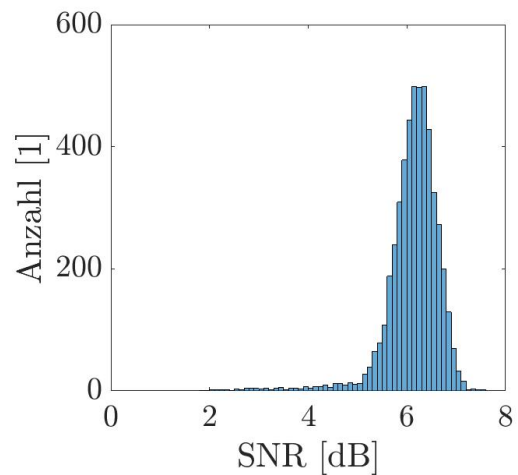


(a) Bereich für die Messung des gesamten Bildes.

(b) Bereich für die Messung der Vibrationen.



(c) Signal-Rausch-Abstände zu Bild (a).



(d) Signal-Rausch-Abstände zu Bild (b).

Abbildung 4.4: Vergleich der Bereiche mit und ohne realen Schwingungen. Bilder (a) und (b) zeigen die blau strichliert umrahmten Bereiche, welche Pixel-Positionen im Histogramm eingetragen wurden. Beide Bilder entstammen aus der gleichen Messung eines Lautsprechers mit einer Auslenkung von $2,7\text{ mm} \pm 0,4\text{ mm}$ bei $2,1\text{ Hz} \pm 0,2\text{ Hz}$. Bild (c) zeigt das Histogramm des gesamten Bildes (a) und Bild (d) zeigt das Histogramm des ausgewählten Bereichs des Bildes (b).

Durch den Vergleich der beiden Histogramme (Abbildung 4.4) mit der Hintergrundmessung (Abbildung 4.3) wird darauf geschlossen, dass Pixel, die keine Vibration detektierten, einen Signal-Rausch-Abstand bis zu 4 dB in der Frequenzdomäne aufweisen. Dieser Wert wird als Parameter für die Vibrationsdetektion in der Programmfunktion „Signal-Rausch-Analyse“ im Abschnitt 3.4.3, als Schwellenwert verwendet.

4.3 Detektion der Auslenkung

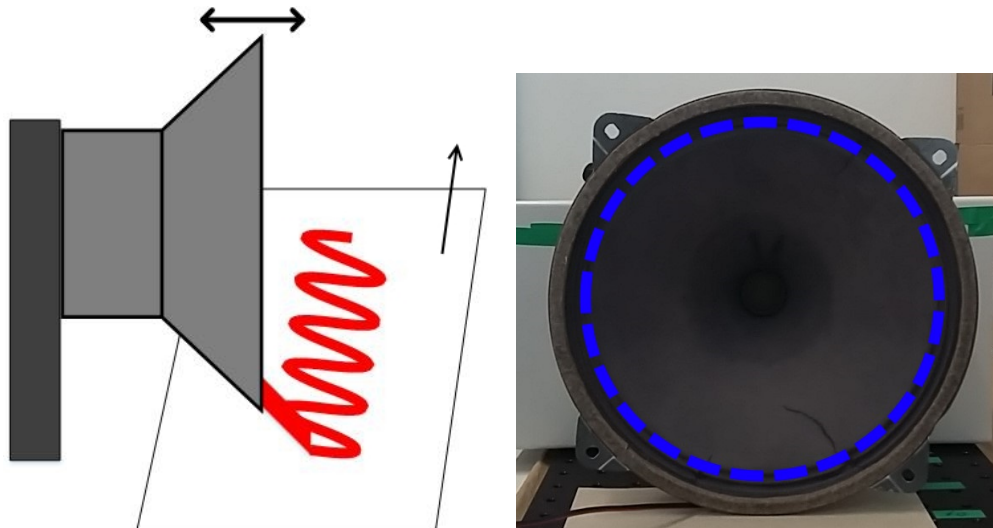
Um die Grenzen der Auslenkungsdetektion zu bestimmen, erfolgte eine Messung verschiedener Vibrationsamplituden. Als Vibrationsquelle diente ein Lautsprecher bei einer Frequenz von $2,1 \text{ Hz} \pm 0,2 \text{ Hz}$ (Abschnitt 4.7). Die Einstellung der Amplitude, mit welcher die Membran schwingt, erfolgte mittels Variation der Spannungsamplitude des Frequenzgenerators, welche über einen Audioverstärker zum Lautsprecher geführt wird. Ein Markierstift wurde auf die Lautsprechermembran geklebt und zeichnete die Schwingung mittels eines vorbeigezogenen Millimeterpapiers auf (Abbildung 4.5 (a)). Die Aufnahme der Auslenkungen erfolgt über einige Perioden. Für die Ermittlung der Spitze-Spitze-Werte wurde der Mittelwert aus mindestens 10 Messungen gebildet. Aufgrund der Elastizität der Membran, der Stiftbreite und der Maschenweite des Papierrasters wird die Unsicherheit der Spitze-Spitze-Messung mit $0,7 \text{ mm}$ angenommen. Die Charakterisierung der Vibrationen wurde aufgrund der Unsicherheit bis zu einer Auslenkung von $0,7 \text{ mm}$ durchgeführt.

Tabelle 4.1: Gemessene Auslenkung des Lautsprechers mit $2,1 \text{ Hz}$ und variabler Generatorspannung. Die Unsicherheit der berechneten Spitze-Spitze-Werte wird mit $0,7 \text{ mm}$ und die Auslenkung mit $0,4 \text{ mm}$ angenommen.

Gen. Spannung: eingestellte Spannung am Gleichspannungsgenerator
 Spitze-Spitze: Mittelwert der gemessenen Spitze-Spitze-Werte
 Auslenkung: Auslenkung des Lautsprechers, aus Spitze-Spitze-Wert

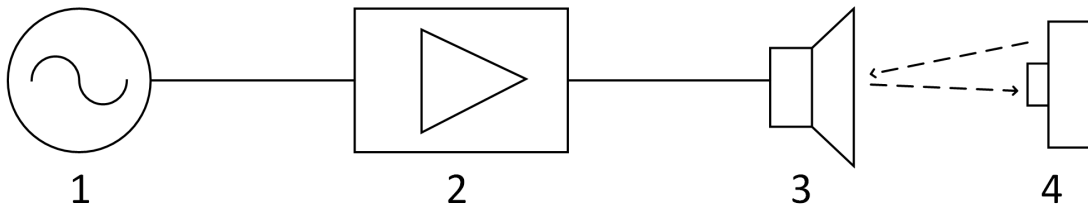
Gen. Spannung [mV]	Spitze-Spitze [mm]	Auslenkung [mm]
150	5,4	2,7
100	4,2	2,1
50	2,3	1,1
30	1,5	0,8
25	1,4	0,7

Für die ermittelten Auslenkungen aus Tabelle 4.1 wurde eine Frequenzdetektion mit dem DME660 durchgeführt. Die daraus erstellten Bilder und Histogramme sind in den Abbildungen 4.6 für die Tiefendaten und in 4.7 für die Amplitudenqualität zu sehen.



(a) Schematische Darstellung des Messaufbaus zur Messung der Auslenkung des Lautsprechers.

(b) Blau strichlierter, umrahmter Bereich, in welchem eine Vibrationsdetektion erfolgen sollte.



(c) Schematischer Aufbau der Messung.

Abbildung 4.5: Bild (a) zeigt eine schematische Darstellung des Aufbaus zur Messung der Auslenkung des Lautsprechers bei verschiedenen Spannungen am Frequenzgenerator. In Bild (b) ist der Auswertebereich zu sehen (strichlierte, blaue Linie). In diesem Bereich erfolgt eine Auslenkung und somit sollte eine Frequenz detektiert werden. Bild (c) zeigt den schematischen Aufbau der Messung mit Frequenzgenerator (1), Verstärker (2), Lautsprecher (3) und DME660 (4).

Für die Auswertung wurde zu jedem Bild ein Histogramm von einem ausgewählten Bereich, wo Schwingungen auftreten und eine Vibrationsdetektion zu erwarten ist, erstellt. Die Parametereinstellungen zur Frequenzdetektion sind hier ein minimaler Schwellenwert von 4 dB und eine Kernmatrixgröße von 5 x 5 Pixel beim Median-Filter sowie auch beim Rauschunterdrückungs-Filter (siehe Abschnitt 3.4.3). In Abbildung 4.5 (b) ist der verwendete Lautsprecher zu sehen. Innerhalb der blau strichlierten Umrandung werden die Pixel ausgewertet und gegebenenfalls Vibrationen detektiert. In Abbildungen 4.6 und 4.7 sind die erkannten Vibrationen der Pixel als eingefärbtes 2D-Bild zur Lokalisierung dargestellt und die dazugehörigen Frequenzen in ein Histogramm eingetragen.

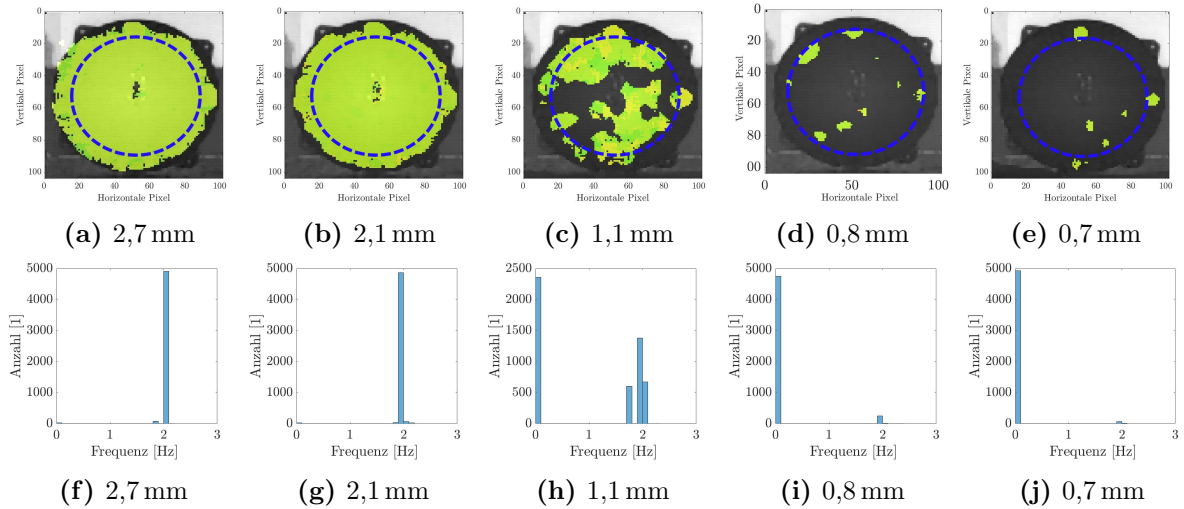


Abbildung 4.6: Berechnung der Frequenz-Histogramme aus den relativen Tiefendaten für die jeweiligen Auslenkungen. In den Bildern (a) bis (e) sind die 2D-Bilder mit den detektierten Schwingungen zu sehen. Ebenfalls ist hier für jedes Bild der analysierte Bereich mit einer blau strichlierten Linie gekennzeichnet. Die Histogramme (f) bis (j) zeigen die Anzahl der Pixel zu den detektierten Frequenzen.

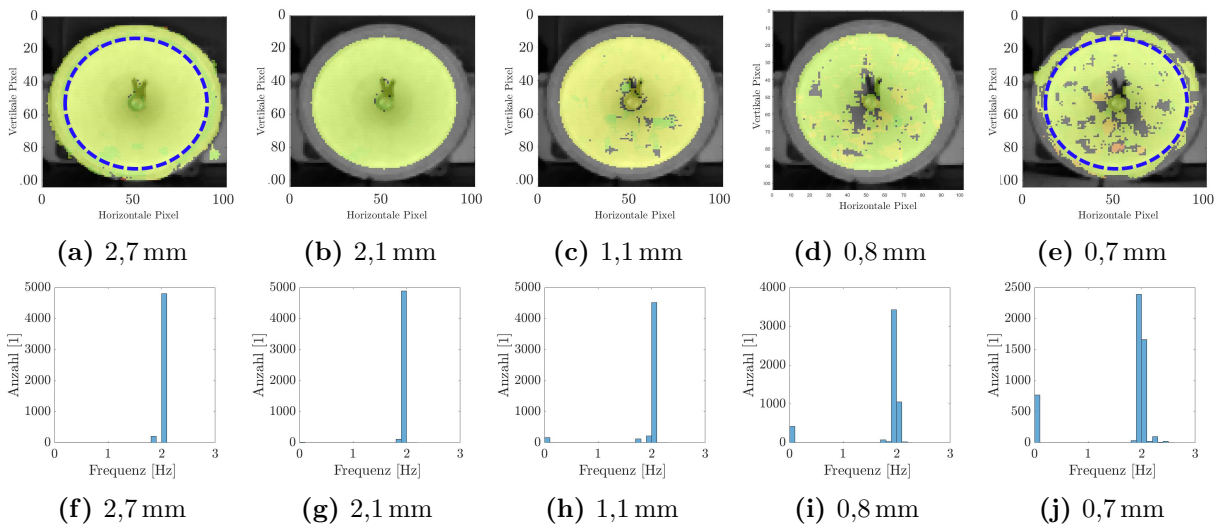


Abbildung 4.7: Berechnung der Frequenz-Histogramme aus der Amplitudenqualität für die jeweiligen relativen Auslenkungen. In Bildern (a) bis (e) sind die 2D-Bilder mit den detektierten Schwingungen zu sehen. Hier ist für die Bilder (a) und (e) der analysierte Bereich mit einer blau strichlierten Linie gekennzeichnet und in Bildern (b) bis (d) sind nur die im Histogramm eingetragenen Bereiche eingefärbt. Die Histogramme (f) bis (j) zeigen die Anzahl der Pixel zu den detektierten Frequenzen.

Es ist zu erkennen, dass die Reduzierung der Auslenkung zu einer Verringerung der Vibrationsdetektion führt.

4.4 Einfluss der Signalqualität und der Filtermaßnahmen auf die Messgenauigkeit

Für die Bestimmung der Fehler in einem Tiefenbild wurde eine Szene mit jeweils einem A4-Papier in Weiß, Grau 50 % und Schwarz als Objekt aufgenommen. Die Ausrichtung des Objekts erfolgte parallel zur Kamera in $45,0 \text{ cm} \pm 0,5 \text{ cm}$. Mit der Serienbildfunktion wurden 100 Bilder je Objekt mit einer Auflösung von 102×72 Pixel (entspricht 7344 Pixel) aufgenommen. Die Darstellung der ermittelten Tiefendaten, für alle Pixel der 100 Bilder, erfolgt über ein Histogramm. Mittels eines Gaußfits über das Histogramm wurde die Standardabweichung berechnet. Diese Kalkulation ist möglich, da real alle Parallelabstände von der Fläche (A4-Papier) bis zum Sensor dieselben sind und theoretisch alle Messungen den gleichen Wert liefern müssen. In Abbildung 4.8 ist beispielhaft das Histogramm der Messung des A4-Papiers in Weiß nach der Entfernung der Abstandsoffsets mit einem Gaußfit gezeigt. Für die Auswertung in Tabelle 4.1 wurden die Daten einmal ohne Bildbearbeitung, dann nach Anwenden des Median-Filters vor und nach dem Entfernen der Offsets sowie nach kombinierter Anwendung dieser beiden Methoden, betrachtet.

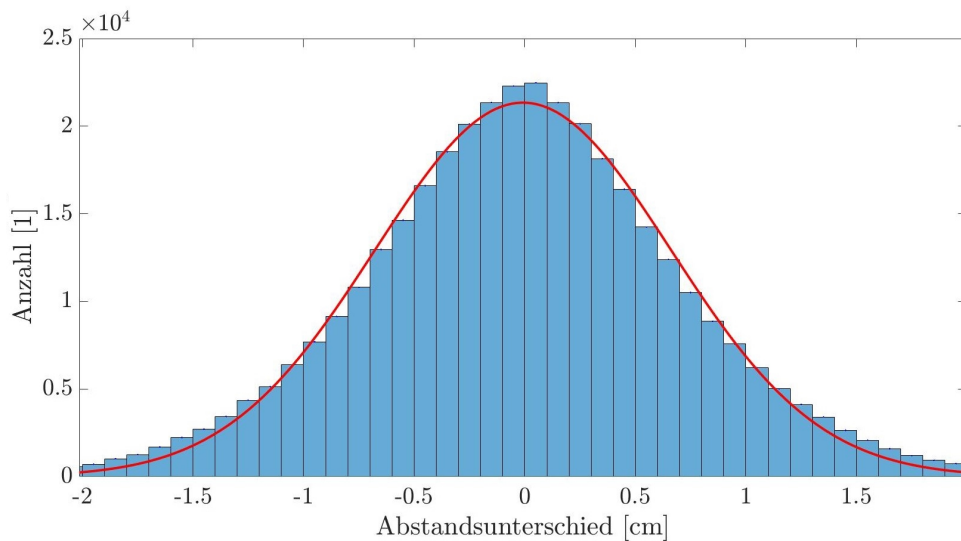


Abbildung 4.8: Histogramm mit Gauß-Verteilung nach Entfernung der Offsets (siehe 3.4.2 „Offset entfernen“). Als Objekt diente hier das Papier in Weiß.

Um die Abweichung nach der Bildbearbeitung mittels Median-Filter zu beurteilen, wurde auf jeder Matrixseite, also für jedes der 100 Bilder, der Filter angewandt.

Durch die „numerische Differentiation“ (siehe 3.4.2) wurde die konstanten Werte für jedes Pixel entfernt. Für die Feststellung der Abweichung bei Kombination der beiden Metho-

den, also Median-Filter und numerische Differentiation, erfolgte zuerst die Entfernung der Offsets und danach die Anwendung des Median-Filters. Für die Histogramm-Balkenbreite wurde das theoretische Auflösungsvermögen des ADC für die durchschnittliche Amplitudenqualität aller Pixel des Bildes herangezogen (siehe Abschnitt 2.4 „Quantisierungsfehler“).

Tabelle 4.2: Auswirkung Farbe und der Filter auf die Genauigkeit der Messung bei unterschiedlichen farbigen Flächen und unveränderten Abstand.

Aqu: durchschnittliche Amplitudenqualität
 Averm: theoretisches Auflösungsvermögen des ADC
 ONB: Tiefenbild ohne Nachbearbeitung
 MedFilt: nach Anwendung Median-Filter
 NAbl: nach Entfernung der Offsets
 NAbIMedFilt: nach Anwendung beider Methoden (MedFilt, NAbl)
 σ : Standardabweichung

	Aqu [LSB]	Averm [cm]	σ ONB [cm]
schwarz	78	0,65	4,3042
grau 50 %	345	0,15	1,3484
weiß	1072	0,05	0,9839
	σ MedFilt [cm]	σ NAbl [cm]	σ NAbIMedFilt [cm]
schwarz	4,3257	2,1333	0,3737
grau 50 %	0,5498	1,0791	0,1521
weiß	0,4366	0,6695	0,1038

In Tabelle 4.2 sind die berechneten Standardabweichungen (σ -Werte) sowie die durchschnittliche Amplitudenqualität des Bildes und das Auflösungsvermögen des ADC eingetragen. Es ist zu erkennen, dass bei steigendem Reflexionsvermögen des Objekts das Auflösungsvermögen steigt und sich die Standardabweichung durch die Bildverarbeitungsschritte verkleinert. Bei hohem Rauschen kann es vorkommen, dass der Median-Filter nicht die gewünschte Wirkung zeigt. Dies ist bei der Messung mit schwarzem Papier der Fall. Hier steigt die Abweichung vom Erwartungswert an.

4.5 Einfluss der Temperatur auf die Abstandsmessung des DME660

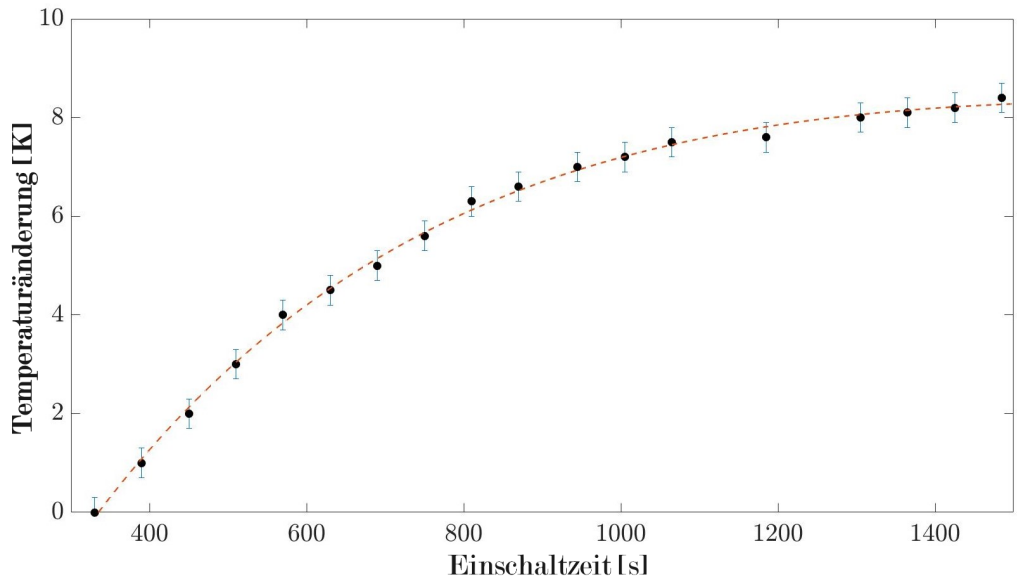
Durch den Betrieb der elektrischen Bauteile erwärmen sich diese und es kommt zu Verschiebungen der Arbeitspunkte. Daraus resultiert dann eine Drift des gemessenen Abstands.

Die Chiptemperatur wurde mit den (nicht justierten) eingebauten Temperaturfühlern des epc660 gemessen, daher werden nur relative Werte betrachtet. Aus Gründen der einfacheren Unterscheidung werden im Folgenden relative Temperaturwerte in Kelvin [K] angegeben. Laut Datenblatt [AG15] weisen die Temperatursensoren ein Auflösungsvermögen (bei einer Samplingtiefe von 14 bit) von $0,067 \frac{K}{LSB}$ auf. Die statistische Unsicherheit der Temperaturmessung mit den vier Sensoren ergibt sich zu 0,3 K.

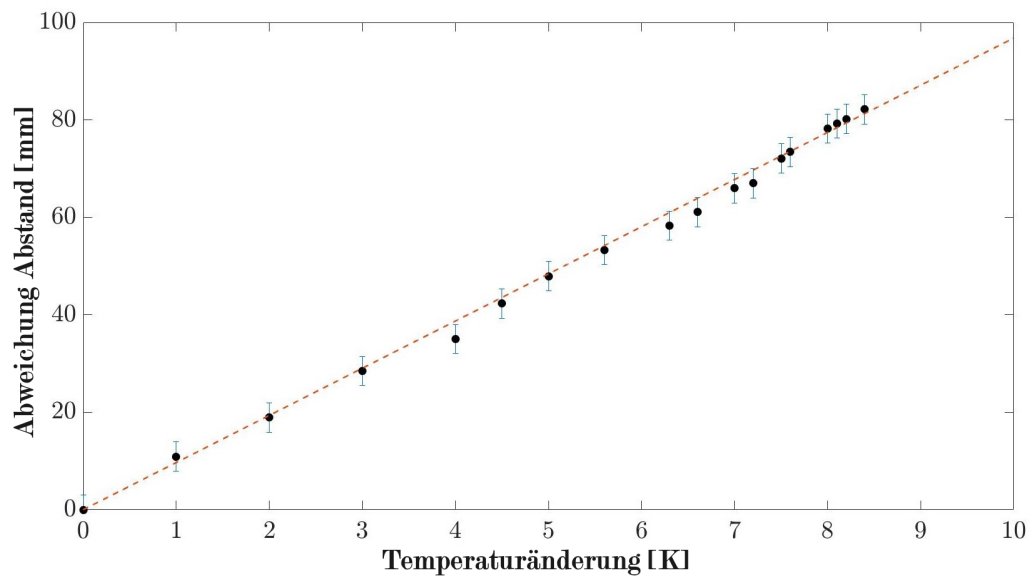
Um das Temperaturverhalten des DME660 zu zeigen, wurden zwei unterschiedliche Messungen durchgeführt. Als Erstes wurde das DME660 in Betrieb genommen und die Temperaturänderung beim Einschalten des Chips über die Zeit beobachtet. Der Offset-Punkt, von dem aus alle Temperaturwerte relativ gemessen wurden, lag bei $35,0 \text{ °C} \pm 0,3 \text{ °C}$ (nicht kalibriert). Abbildung 4.9 (a) zeigt den relativen Temperaturverlauf beim Einschalten des DME660. Eine formgebende Trendlinie von zwei addierten Exponentialfunktionen liefert den besten Fit.

Zusätzlich wurde der Abstand zu einem fixen, flachen Objekt bei verschiedenen Chiptemperaturen gemessen. Als Messobjekt wurde eine weiße Oberfläche (A4-Blatt) mit einem Abstand von $45,0 \text{ cm} \pm 0,5 \text{ cm}$ zum Sensor gewählt und der entsprechende Bereich wurde mittels Range-of-Interest im Graphical-User-Interface selektiert. Für die Bestimmung des Abstandes wurden für jede Temperatur zehn Tiefenbilder aufgenommen und die entsprechenden Pixel gemittelt. Der berechnete Abstandswert bei der niedrigsten gemessenen Temperatur wurde als Referenzpunkt (Offset-Punkt) angenommen und zu diesem die Abweichung der Abstandsmessung in Abhängigkeit der Temperatur ermittelt. Der Funktionsgraph (b) in Abbildung 4.9 zeigt die Abhängigkeit der gemessenen Distanzänderung von der Temperatur. Durch die ermittelte Steigung des linearen Fits erhält man eine Abhängigkeit von $9,7 \frac{\text{mm}}{\text{K}}$ mit einer Unsicherheit von $1,5 \frac{\text{mm}}{\text{K}}$.

Diese Messungen konnten zeigen, dass die Abstandsmessung durch die Temperatur des Chips beeinflusst wird. Um diesen Effekt zu minimieren, sollten die elektronischen Bauteile eine möglichst gleichbleibende Temperatur während der Vibrationsmessung aufweisen.



(a) Temperaturanstieg als Funktion der Zeit nach dem Einschalten.

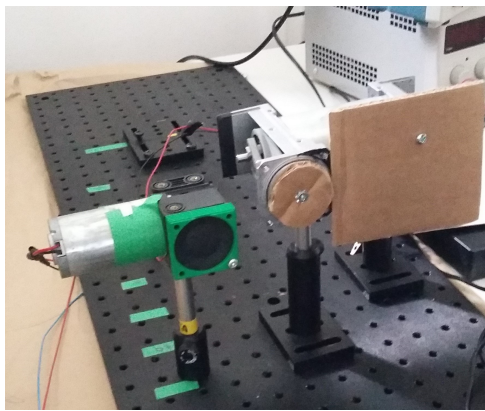


(b) Abweichung der Messung des Abstands in Abhängigkeit der Temperatur.

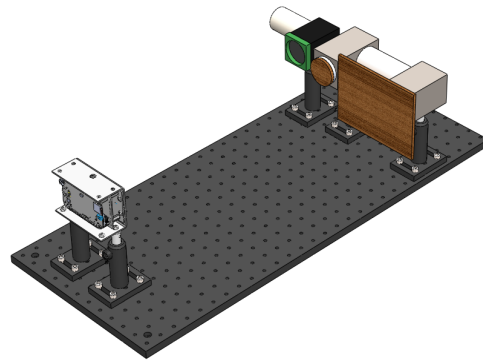
Abbildung 4.9: Bild (a) zeigt das relative Temperaturverhalten des epc660-Chips beim Einschalten, mit einer formgebenden Trendlinie. Durch den linearen Fit in Bild (b) konnte der Einfluss der Chiptemperatur auf die Abstandsmessung zu $9,7 \frac{\text{mm}}{\text{K}}$ bestimmt werden.

4.6 Detektion zweier Vibrationsquellen in einer Szene

Da jedes Pixel einen Ausgabewert liefert und die Filter (Größe der Kernmatrix, siehe 3.4.2) und Analysefunktionen immer nur auf einen Bildbereich angewendet werden, kann eine Analyse von unterschiedlichen Vibrationsquellen bzw. Frequenzen und Amplituden mit einer Aufnahme durchgeführt werden. Für diese Messung wurden die beiden Pumpen in einem Abstand von 45 bis 50 cm vor dem TOF-Sensor positioniert. Die Platzierung der Pumpen ist in Abbildung 4.10 zu sehen.



(a) Positionierung der Membranpumpe 1 (rechts) und Membranpumpe 2 (links).



(b) 3D-Darstellung der Positionierung des DME660.

Abbildung 4.10: Aufbau und Positionierung der Membranpumpen bei der Messung.

Die Schwingung der Pumpen wurde mithilfe einer Gleichspannungsquelle eingestellt. Durch Messung der Umdrehungen pro Sekunde mit dem Drehzahlmesser und einem Oszilloskop wurden die Pumpen auf einen Näherungswert 2,5 Hz und 5,5 Hz reguliert. Zum Überprüfen der detektierten Frequenzen mit dem DME660 wurde der zweite TOF-Chip der Firma STM herangezogen. Hierzu erfolgte eine separate Messung für jede Pumpe, da der Chip von STM nur für Punktmessungen konzipiert ist. Der Abstand der Sensoreinheit des STM-Chips zum Objekt bei der Kalibrierung betrug $10,0\text{ cm} \pm 0,5\text{ cm}$. Laut Herstellerangaben im Datenblatt liegt hier der genaueste bzw. sensibelste Bereich des Chips [STM16].

Die Bestimmung der Frequenz erfolgte durch den wie in Abschnitt 3.4 beschriebenen Ablauf. Aus Gründen der schnelleren Bildakquisition wurde auf den Modus ACC (siehe Abschnitt 2.3.4) des DME660, der die Signalqualität direkt in eine CSV-Datei schreibt, zurückgegriffen und der Bildbereich verkleinert.

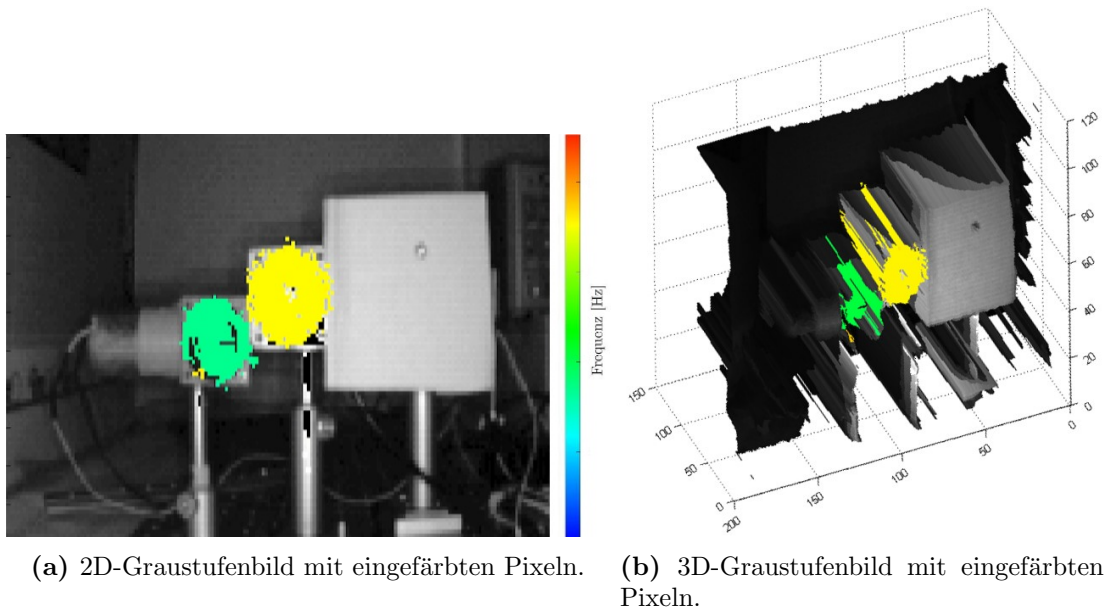
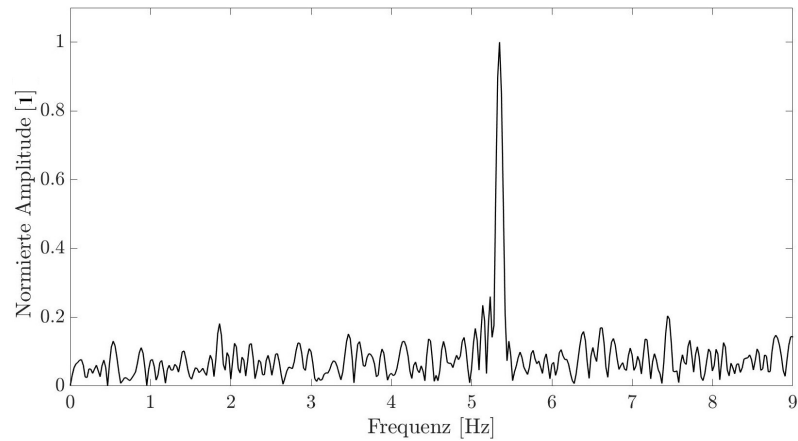


Abbildung 4.11: 2D-Graustufenbild mit überlappter Vibrationsdetektion der zwei Membranpumpen in (a) und 3D-Darstellung in (b). Die eingefärbten Pixel zeigen die detektierten Vibrationen.

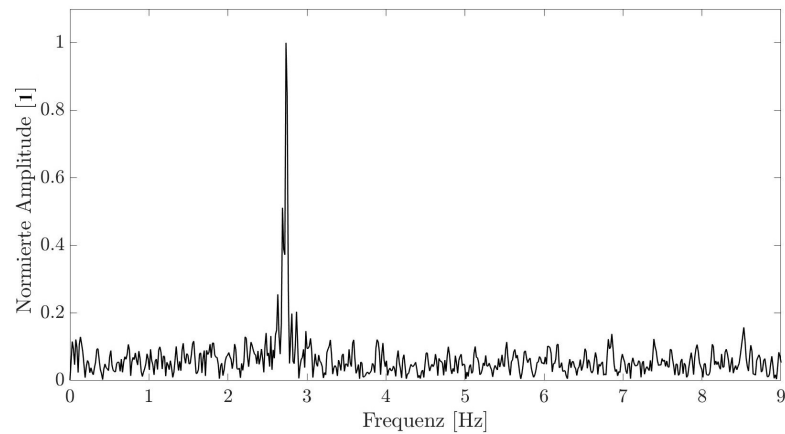
Die Messung der Frequenzen ergab für das bildgebende Verfahren mit dem DME660 für die Membranpumpe 1: 5,31 Hz mit einer Halbwertsbreite von 0,48 Hz und für die Membranpumpe 2: 2,91 Hz mit einer Halbwertsbreite von 0,44 Hz.

Die Punkt-Referenzmessung mit dem STM-Chip ergab Frequenzen von 5,35 Hz mit einer Halbwertsbreite von 0,10 Hz für die Membranpumpe 1 und von 2,74 Hz mit einer Halbwertsbreite von 0,07 Hz für die Membranpumpe 2. Die Frequenzspektren der Messungen sind in Abbildung 4.12 zu sehen.

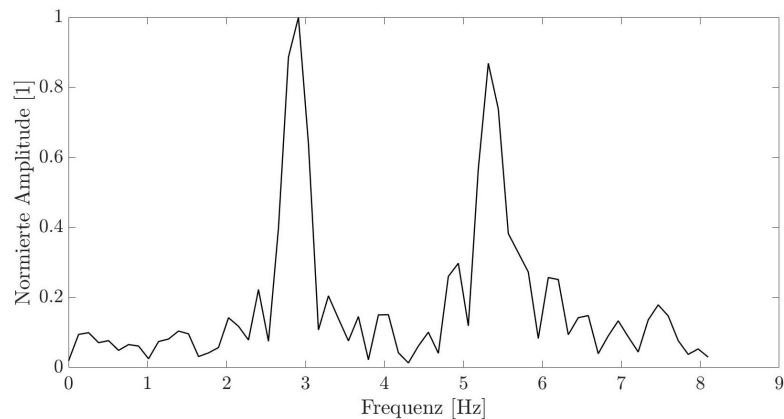
Die gemessenen Frequenzpeaks der unterschiedlichen TOF-Chips liegen innerhalb der Halbwertsbreiten. Es konnte somit gezeigt werden, dass es möglich ist, mehrere Frequenzen mit dem DME660 in einer Szene zu detektieren und als 2D- bzw. 3D-Bild zu visualisieren (siehe Abbildung 4.11).



(a) FFT-Analyse der Messung mit dem STM-Chip der Membranpumpe 1 (Peak bei 5,35 Hz mit einer Halbwertsbreite von 0,10 Hz).



(b) FFT-Analyse der Messung mit dem STM-Chip der Membranpumpe 2 (Peak bei 2,74 Hz mit einer Halbwertsbreite von 0,07 Hz).



(c) FFT-Analyse der Messung mit dem DME660 (Peaks bei 2,91 Hz mit einer Halbwertsbreite von 0,44 Hz und 5,31 Hz mit einer Halbwertsbreite von 0,48 Hz).

Abbildung 4.12: Vergleich der Messungen beider Membranpumpen mittels FFT-Analyse aus Daten der unterschiedlichen TOF-Chips. Die Graphen (a) und (b) zeigen die FFT-Spektren aus dem STM-Chip für beide Pumpen und Graph (c) für das DME660. Das bildgebende Verfahren kann die gesamte Szene gleichzeitig aufnehmen, dadurch ist es möglich beide Frequenzen gleichzeitig zu detektieren.

4.7 Skalierung auf höhere Frequenzen

Aufgrund des Designs des Evaluation-Sets DME660 ist es nicht möglich, die maximale Bildrate des TOF-Chips epc660 auszuschöpfen. Um höhere Frequenzen zu detektieren, ist eine höhere Bildrate notwendig. Um zu überprüfen, ob die Detektion von höheren Schwingungsfrequenzen mit der TOF-Methode möglich ist, wurde der TOF-Chip von STM herangezogen. Die maximale erreichte Punktrate (dots-per-second (dps)) liegt bei 48 dps. Somit ist es möglich, Vibrationen mit einer Frequenz bis zu 24 Hz zu detektieren (siehe Abschnitt 2.4.4). Der Aufbau erfolgte, wie in Abbildung 4.13 gezeigt, mit dem Lautsprecher, welcher mit dem Funktionsgenerator und dem Audioverstärker angesteuert wird.

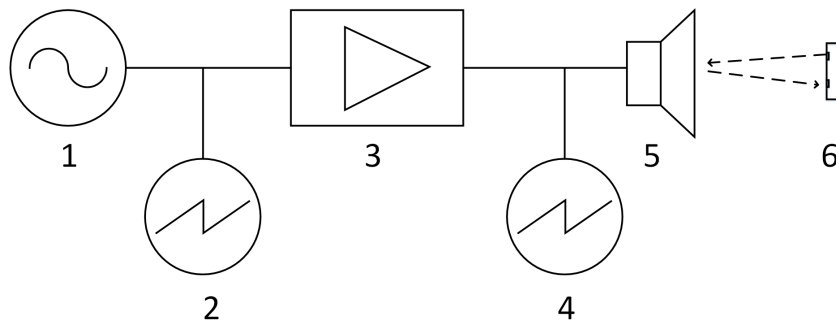


Abbildung 4.13: Schematische Darstellung des Messaufbaus für die Messung höherer Frequenzen. (1) Funktionsgenerator, (2) Oszilloskop-Kanal 1, (3) Audioverstärker, (4) Oszilloskop-Kanal 2, (5) Lautsprecher, (6) TOF-Chip STM.

In Tabelle 4.3 sind die Einstellungen am Frequenzgenerator und die Messwerte aufgelistet. Das ausgegebene Signal am Frequenzgenerator erfährt eine Verstärkung und eine Umwandlung des elektrischen Signals in eine mechanische Schwingung. Bei diesen Umwandlungsschritten kann es durch die Charakteristik der Bauteile zu Abweichungen von der Sinusform des Signals kommen. Diese Formveränderungen haben Einfluss auf das Schwingungsverhalten des Lautsprechers und kann zu Abweichungen der eingestellten Frequenz führen.

Der Vergleich der ermittelten Frequenzspektren aus der FFT-Analyse in Tabelle 4.3 zeigt eine gute Übereinstimmung mit den erhaltenen Werten der Abstandsmessung mittels Chip von STM. Eine Erklärung für die Abweichungen bei höheren Frequenzen können die statistischen Unsicherheiten bei der TOF-Messung sowie auch die charakteristische Transformation des elektrischen Signals in eine mechanische Schwingung durch den Laut-

sprecher sein.

Tabelle 4.3: Detektion und ausgewertete Frequenzen mit dem STM-Chip im Vergleich zur gemessenen Ausgangsfrequenz des Funktionsgenerators und der Steuerfrequenz des Lautsprechers.

f-Gen:	eingestellte Frequenz am Funktionsgenerator
f-Oszi 1:	FFT-Analyse, Oszilloskop Kanal 1, Funktionsgenerator
f-Oszi 2:	FFT-Analyse, Oszilloskop Kanal 2, Verstärkerausgang
O-FWHM:	Halbwertsbreite der Peaks, Oszilloskop
f-STM:	FFT-Analyse des gemessenen STM-Chip
STM-FWHM:	Halbwertsbreite der Peaks, STM-Chips

f-Gen [Hz]	f-Oszi 1 [Hz]	f-Oszi 2 [Hz]	O-FWHM [Hz]	f-STM [Hz]	STM-FWHM [Hz]
2	1,95	1,95	0,49	2,05	0,13
4	3,91	3,91	0,50	4,09	0,12
8	8,06	8,06	0,60	8,18	0,09
10	9,77	9,77	1,57	10,28	0,08
15	15,26	15,26	1,58	15,39	0,07
20	20,14	20,14	1,57	20,46	0,14

Als Beispiel für die Ausgabe ist in Abbildung 4.14 das berechnete FFT-Spektrum des STM-Chip-Signals bei eingestellten 20,14 Hz dargestellt. Hier wird eindeutig eine Vibration mit $20,46 \text{ Hz} \pm 0,14 \text{ Hz}$ detektiert. Die Abbildung 4.15 (a) zeigt die gemessenen Spannungssignale des Oszilloskops vor und nach dem Verstärker (Abbildung 4.13 (2) und (4)). Durch die Charakteristik der Bauteile verändert sich die Form des Signals. Dies ist auch in den FFT-Spektren zu erkennen (Abbildung 4.15 (b) und (c)).

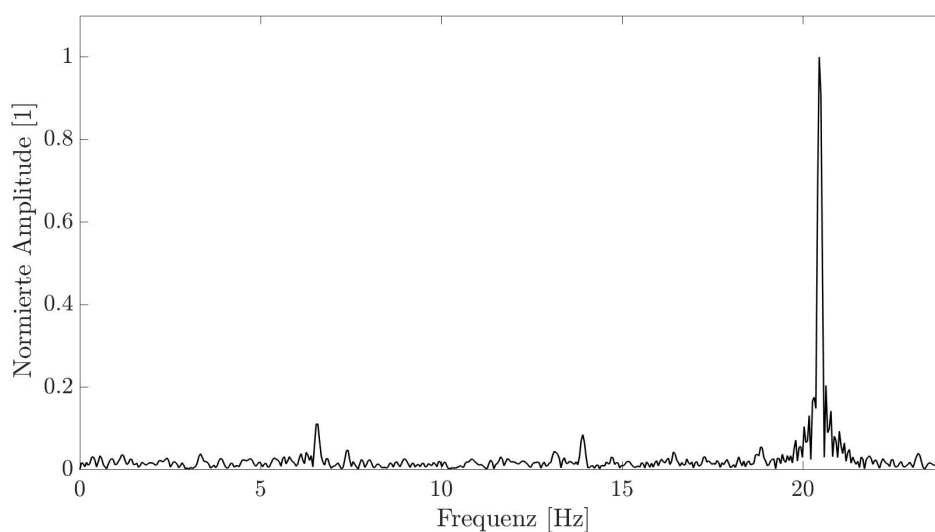
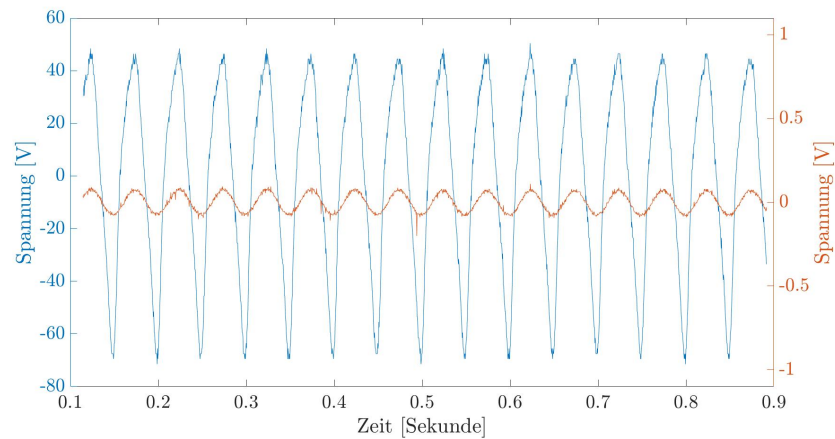
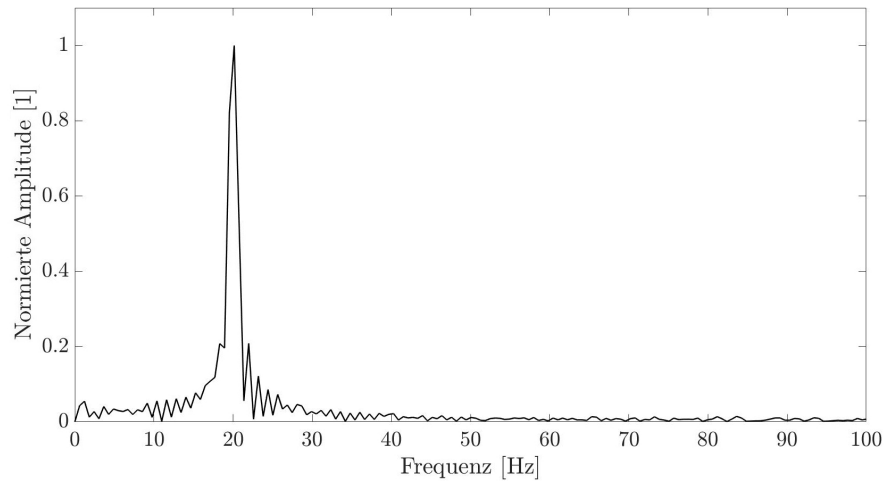


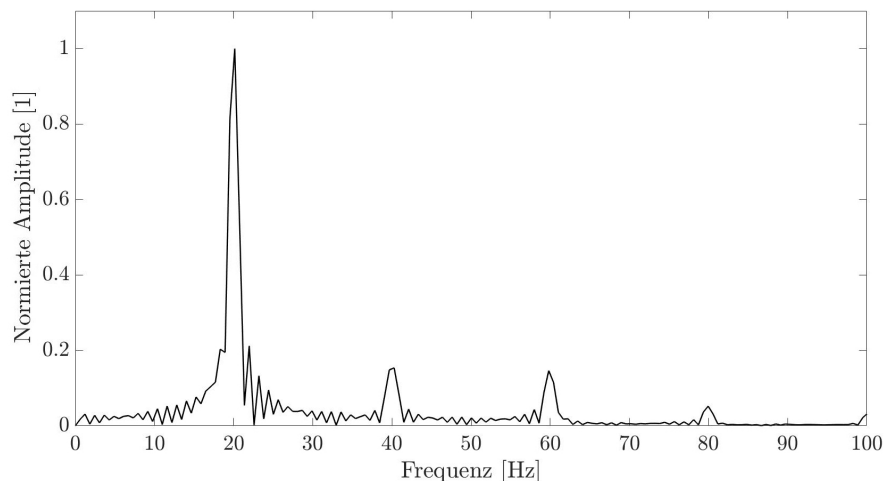
Abbildung 4.14: FFT-Spektrum der Messung mittels STM-Chip bei einer eingestellten Frequenz von 20,14 Hz und einer Detektion von 20,46 Hz.



(a) Signal vor dem Verstärker mit Daten aus dem Oszilloskop (Abbildung 4.13 (2), orange) und nach dem Verstärker mit Daten aus dem Oszilloskop (Abbildung 4.13 (4), blau) bei 20 Hz.



(b) FFT-Analyse des Spannungssignals vor dem Verstärker.

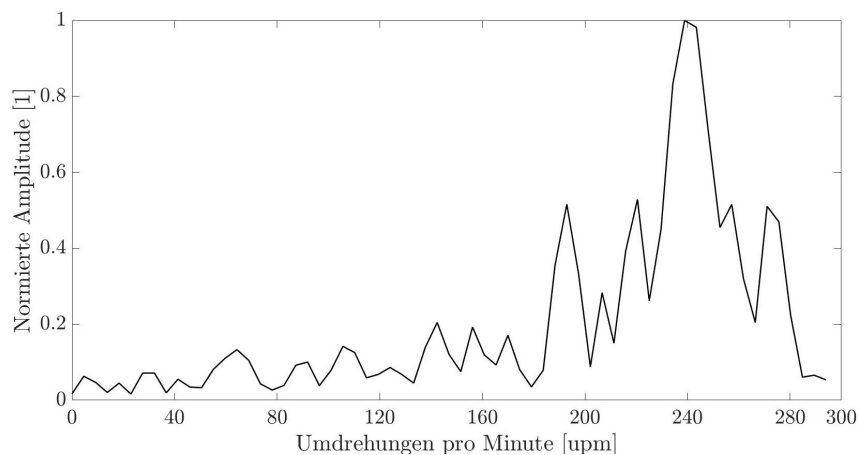


(c) FFT-Analyse des Spannungssignals nach dem Verstärker.

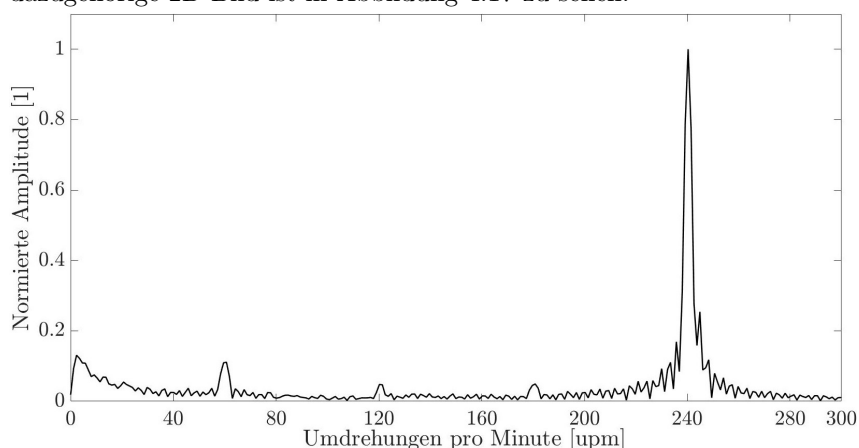
Abbildung 4.15: Vergleich der mittels Oszilloskop gemessenen Spannungen in (a) vor dem Verstärker (orange) und nach dem Verstärker (blau) bei eingestellter Frequenz von 20,14 Hz am Funktionsgenerator. Bild (b) und (c) zeigen die daraus berechneten FFT-Spektren vor und nach dem Verstärker.

4.8 Vibrationsmessung am Automotor

Die abschließende Messung an einen im Startvorgang befindlichen Automotor (Audi 80) erfolgte mit dem DME660, montiert auf einem handelsüblichen Kamerastativ. Die Messung hatte das Ziel, zu zeigen, welche Anwendungsmöglichkeiten mit dieser TOF-Technologie für die Vibrationsdetektion realisierbar sind. Es wurde ein Graustufenbild mit der Graustufenfunktion (Abschnitt 3.2.7.1 „Grayscale“) der TOF-Kamera angefertigt. Die anschließende Vergleichsmessung des Startvorganges, wie in Abschnitt 3.3.1 „Automotor“ ausgeführt, erfolgte über die Messung der Amplitudenqualität (siehe Abschnitt 2.3.4), da hier eine höhere Bildrate aufgrund der geringeren Datenmengen erreicht wird.



(a) Frequenzspektrum (umgerechnet in upm) der Vibrationsmessung mit dem DME660 (Peak bei 239 Hz mit einer Halbwertsbreite von 24 Hz. Das dazugehörige 2D-Bild ist in Abbildung 4.17 zu sehen.



(b) Frequenzspektrum (umgerechnet in upm) des Anlasserstroms als Vergleichsmessung (Peak bei 240 upm mit einer Halbwertsbreite von 4 upm)

Abbildung 4.16: FFT-Analyse der Vibrationsmessung mittels DME660 (a) und des Starterstroms (b) eines 4-Takt-Verbrennungsmotors (Audi 80) beim Anlassen.

Abbildung 4.16 zeigt das FFT-Spektrum des gesamten Bildes sowie das Referenz-FFT-Spektrum des Starterstroms. Die detektierten Frequenzen sind in Umdrehungen pro Minute (upm) umgerechnet dargestellt, um einen leichteren Vergleich machen zu können. Die gemessenen Frequenzen werden bei der bildgebenden Detektion (DME660) mit dem Faktor 60 (Sekunden) multipliziert ($a_{upm} = f_{DME660} \cdot 60$). Für die Berechnung der Umdrehungen pro Minute bei der Messung des Starterstroms muss zusätzlich noch die doppelte Frequenz der Stromschwankungen (siehe Abschnitt 3.3.1 „Automotor“) berücksichtigt werden ($b_{upm} = \frac{f_{Strom}}{2} \cdot 60$).

Durch den Vergleich der beiden Spektren in Abbildung 4.16 ist zu erkennen, dass die Detektion der Vibrationsfrequenz bzw. der Umdrehung pro Minute sehr gut mit dem erwarteten Wert aus der FFT-Analyse des Starterstroms übereinstimmt. Der Maximalwert des FFT-Spektrums der gemessenen Vibrationen liegt bei einer Umdrehungszahl von 239 upm mit einer Halbwertsbreite von 24 upm. Die gemessene Umdrehungszahl des Anlasserstroms beträgt 240 upm mit einer Halbwertsbreite von 4 upm.

Die Erstellung eines Bildes erfolgt durch die Überlagerung des zuvor aufgenommenen Graustufenbilds mit den Frequenzinformationen der Pixel. Das entstehende Bild ist in Abbildung 4.17 zu sehen. Die blau umrahmte, aufgespannte Fläche zeigt den aus Performancegründen ausgewählten Bereich, in welchem die Frequenzanalyse erfolgte.

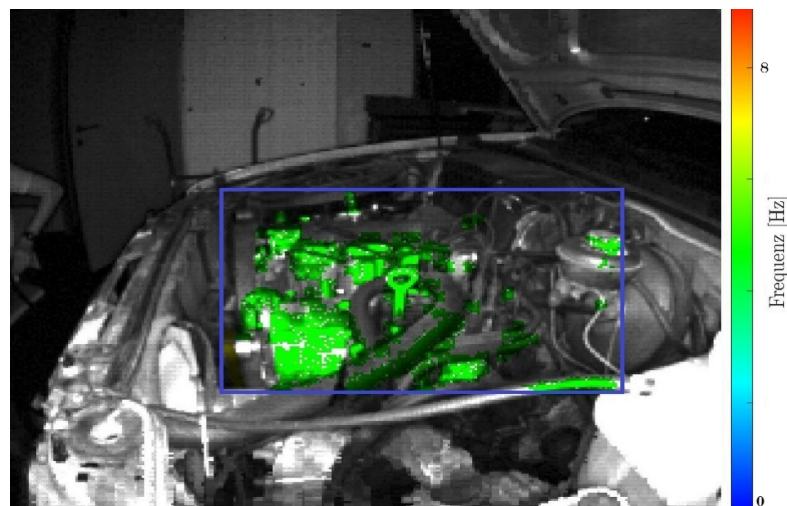


Abbildung 4.17: Graustufenbild des Automotor mit überlagerter Vibrationsdetektion im blau umrahmten Bildbereich.

Abschließend ist zu erwähnen, dass die gemessenen Vibrationen mit den erwarteten Frequenzen übereinstimmen und sich die Positionen der eingefärbten Pixel mit den realen Schwingungen des Motors decken.

Schlussfolgerungen und Ausblick

Das Ziel dieser Arbeit lag in der Evaluierung und Umsetzung einer neuen Vibrationsmessmethode auf Basis von Time-of-Flight-Sensoren. Die Tests sind mittels bildgebendem Evaluation-Set DME660 der Firma ESPROS und zudem mit einem Punktsensor der Firma STM durchgeführt worden. Um eine derartige Messmethode zu realisieren und eine Charakterisierung des möglichen Arbeitsbereichs sowie eine Analyse der nachweisbaren Fehler durchführen zu können, war es notwendig, die vorliegenden Messdaten, welche die Evaluation-Sets zur Verfügung stellen, für die Weiterverarbeitung entsprechend aufzubereiten. Für die Analyse der Daten und die Feststellung der Frequenz ist ein Analysetool mittels MATLAB erstellt worden, das die Nachbearbeitung zur Verbesserung der Bildqualität und die Signalanalyse in der Zeit- sowie in der Frequenzdomäne vornimmt. Für die Untersuchung des Potentials dieser Methode erfolgte der Aufbau und die Analyse verschiedener Vibrationsquellen.

Bei den Messungen mit dem Evaluation-Set (DME660) konnte die angegebene Bildrate des TOF-Chips (EPC660) nicht erreicht werden. Dies kann an den verschiedenen Toleranzen der Datenschnittstellen, der Architektur des Prozessors sowie an der Software des DME660 liegen. Nach Rücksprache mit dem Distributor (NEGmbH) wurde bestätigt, dass die theoretische Bildrate nicht voll ausgeschöpft werden kann. Um diese Schwachstelle des Evaluation-Sets zu beheben, entwickelt der Hersteller (ESPROS) derzeit eine Field-Programmable-Gate-Array (FPGA)-Lösung, welche im nächsten Jahr auf den Markt kommen soll.

Damit das DME660 getestet werden konnte und Ergebnisse für die mögliche Verwen-

derung der Technologie als Vibrometer zu erhalten, erfolgten die Messungen mit geringerer Auflösung und einem reduzierten Bildausschnitt. Zudem wurden die Vibrationsquellen an die Leistung des Evaluation-Sets angepasst.

Mit der Analysesoftware und der zur Verfügung stehenden Hardware ist es gelungen, Frequenzen bis zu 4 Hz mittels bildgebendem TOF-Evaluation-Set DME660 sowie bis zu 20 Hz bei der Punktmessung mittels STM-Chip zu analysieren. Die Detektion der Frequenzen kann bis zu einer Auslenkung von 1 mm über die Berechnung des Abstandes erfolgen. Des Weiteren ist es mit diesem Setup gelungen, die Frequenz eines Automotors beim Starten zu detektieren und mit dem Anlasserstrom zu vergleichen. Die Anwendungsmöglichkeit der TOF-Technologie für die bildgebende Detektion von Frequenzen konnte somit in einem eingeschränkten Bereich erfolgreich umgesetzt werden.

Zusammenfassend kann gesagt werden, dass die Vorteile dieser Methode gegenüber herkömmlichen Vibrometern der einfache Aufbau, die Robustheit und die geringeren Serienkosten bei der Herstellung sind. Die derzeitigen Grenzen dieser Technologie liegen in der Bildrate und der relativen Abstandsmessung. Um diese Herausforderungen zu meistern, können verschiedene Ansätze weiter untersucht werden, auf die im Folgenden noch kurz eingegangen wird.

Steigerung der Bildrate: Eine Erhöhung der Bildrate könnte mit der Implementierung eines SoC-basierten Prozessorsystems zur Berechnung und Auswertung der Tiefendaten erfolgen. Um die Messung zu unterteilen und einen Live-View zu realisieren, kann eine Erweiterung der FFT-Analyse zu einer Short-Time-Fourier-Transform (STFT)-Analyse (Kurzzeit-Fouriertransformation) angedacht werden. Zudem kann eine FPGA-Lösung einen weiteren Performance-Gewinn bedeuten, wie sie zum Beispiel Zhan et al. [ZTZ⁺17] in ihrer Publikation zur Erhöhung der Bildrate bei der Phase-Measuring-Profilometry (PMP) Methode angewendet haben. Dabei haben sie eine Bildrate der Rohdaten von 571 fps mit einer Auflösung von 1024 x 768 Pixel erreicht.

Steigerung der Tiefenauflösung: Das derzeitige Forschungsinteresse verschiedener Chiphersteller und Forschungsinstitute lässt darauf schließen, dass die Technologie der Abstandsmessung mittels TOF in den nächsten Jahren Fortschritte bei der Genauigkeit erzielen wird. Hier sind derzeit auch die direkte TOF-Messung mittels SPADs und die PMP-Messungen im Fokus [VCGRV17, ZTZ⁺17].

SPAD: Hinsichtlich der Entwicklung von SPAD-Arrays hat es ein EU-Projekt in Kooperation mit der Universität in Edinburgh gegeben. Dieser Chip erreicht eine Bildrate von einer Million Bilder pro Sekunde mit einer Auflösung von 128 x 128 Pixel. Diese Technologie soll für medizinische Detektoren genutzt werden und kommt heute schon bei STM für die Abstandsmessung zur Anwendung. [Com15, RS09]

Unsicherheitsbetrachtung: Für die Unsicherheitsbetrachtung der TOF-Vibrationsmessung mit dem Korrelationsabtastungs-Verfahren (siehe Abschnitt 2.3.3) müssen die Größen analysiert werden, welche eine Vibration definieren. Die erste Größe ist hier der relativ gemessene Abstand, wobei sich die Unsicherheit mit der empfangenen Amplitudenqualität (siehe Abschnitt 2.3.4) ändert. Die Standardabweichungen betragen je nach Ausnützung des ADC-Bereichs, ohne Korrektur der statistischen Fehler, 0,98 bis 4,30 cm (siehe Tabelle 4.2). Die Frequenz ist die zweite Größe welche betrachtet werden muss. Die Unsicherheit hängt hier von der Bildrate des Chips und der Geschwindigkeit des Systems (Einplatinencomputer) ab. Bei den Messungen wurden die Halbwertsbreiten der detektierten Frequenzpeaks als Fehler angenommen. Die Halbwertsbreiten konnten hier im Bereich von 0,4 bis 0,5 Hz ermittelt werden.

Ausblick

Es hat sich gezeigt, dass die Durchführung einer Detektion von Vibrationen und deren Visualisierung mittels TOF-Technologie gelingt. Es ist zu erwarten, dass es bei der derzeitigen Dynamik in der Forschung und Entwicklung von Sensor- und Prozessortechnologien in naher Zukunft möglich sein wird, die Genauigkeit der Tiefenmessung und der Bildrate zu erhöhen, um Vibrationen im akustischen Bereich detektieren zu können. Dadurch könnten auf bildgebender TOF-Technologie basierende Vibrometer günstigere und robustere Alternativen zu den derzeit erhältlichen Produkten darstellen.

Anhang\Matlab-Code

6.1 Start-Skript für die Frequenzdetektion

Im Folgenden ist das Start-Skript der Frequenzdetektion angeführt. In den weiteren Abschnitten sind die verschiedenen Unterfunktionen des Auswertealgorithmus gezeigt.

```
1 clc
2 close all
3
4 frequ_bin_kreis= false; %frequen_analyse_bin ein/aus
5 if (frequ_bin_kreis == false)
6     clear
7     frequ_bin_kreis= false;
8 end
9
10 %% Parameter einstellen %%
11 offset = 137;
12 pixx = 35;
13 pixy = 35;
14 fs = 24; %Sampling_rate TOF DCS4
15 vir_fps = false;
16 differenz = true;
17 integral = false;
18 frequ_det_on = false;
19 nan_on = true; %NaNs auf 0 setzen
20 SRAbstand = 4.1; %S/R Abstand von Frequenamplitude
21
22 %% parameter der Rauschunterdrückung %%
23 M_groesse_rausch = 5; %n x n ungerade
24 step_size = 1;
25 toleranz = 0.2; %Hz
26 r_filterpar = 0.8; % 0 bis 1, wieviel Prozent pro...
27 ...Kernmatrix mit einer Toleranz \”ubereinstimmen m\”ussen
28
29 %% Datei %%
```

```

30 filename= '..\testordner\testfiledcs4.csv';
31
32 %% Variablen definieren %%
33 dcs_typ = 0;
34 starten = true;
35
36 %% Daten Laden %%
37 pro2 = strfind(filename,'dcs2');
38 pro4 = strfind(filename,'dcs4');
39
40 if (~isempty(pro2))
41     dcs_typ = 2;
42 elseif (~isempty(pro4))
43     dcs_typ = 4;
44 elseif (exist('filename_dcc','var'))
45     dcs_typ = 0;
46 else
47     disp('Falscher Dateiname: Auswertung wird Abgebrochen')
48     starten = false;
49 end
50
51 if (starten)
52     if (dcs_typ ~= 0)
53         [part_roh] = load_dcs(filename);
54         bilder_anz = size(part_roh,3)/dcs_typ;
55         %%Bildrate virtuell erh\''ohen %%
56         if (vir_fps)
57             fs = fs*(2*bilder_anz-2)/bilder_anz;
58             part_roh = fps_raise(part_roh,dcs_typ);
59         end
60         %% Tiefendaten und Amplitudenqualit\''at berechnen %%
61         [d_tof,a_tof,bilder] = roh_daten_auswertung(part_roh,dcs_typ,offset);
62     else
63         a_tof = load_dcs(filename);
64         d_tof = load_dcs(filename_dcc);
65         bilder = size(a_tof,3);
66     end
67
68     clear 'part_roh'
69     clear 'part_roh'
70
71     if (nan_on)
72         a_tof(isnan(a_tof)) = 0;
73         d_tof(isnan(d_tof)) = 0;
74     end
75
76     %% unbearbeitete Daten zwischenspeichern %%
77     a_tof_roh = a_tof;
78     d_tof_roh = d_tof;
79
80     %% Offsets entfernen %%
81     a_tof = ableitungsmethode(a_tof,integral);
82     d_tof = ableitungsmethode(d_tof,integral);
83
84     a_tof = mean_abzug(a_tof);
85     d_tof = mean_abzug(d_tof);
86

```

```

87 bilder = bilder-2;
88
89 %% Frequenz Analyse %%
90 if (frequ_det_on)
91
92     a_tof = median_filter(a_tof);
93     d_tof = median_filter(d_tof);
94
95     [det_frequ_A,frequ_search_bin_A,det_frequ_fwhm_A,db_mat_A] = frequ_analysis(a_tof,fs,
96         SRAbstand);
97
98     [det_frequ_D,frequ_search_bin_D,det_frequ_fwhm_D,db_mat_D] = frequ_analysis(d_tof,fs,
99         SRAbstand);
100
101     % Bildbereich f\ur Auswertung der Auslenkungserfassung;
102     if (frequ_bin_kreis)
103         [y1,x1] = size(frequ_search_bin_A);
104         x = 52;
105         y = 52;
106         r = 40;
107
108         [a,b] = meshgrid([x - 1:-1:0 1:x1 - x], [y - 1:-1:0 1:y1 - y]);
109         mm = a.^2 + b.^2 <= r^2;
110         frequ_search_bin_A = frequ_search_bin_A.*mm;
111         frequ_search_bin_D = frequ_search_bin_D.*mm;
112     end
113
114     [frequ_mean_bereich_A] = region_analysis(a_tof,det_frequ_A,frequ_search_bin_A);
115     fft_analyse_norm(frequ_mean_bereich_A,fs,'A_Bereichanalyse');
116     if sum(frequ_mean_bereich_A(:))==0
117         disp('Zu wenig mit Signalabstand detektiert es wird jede detektierte Frequenz angezeigt')
118         frequ_search_bin_A(:) = 1;
119     end
120
121     [frequ_mean_bereich_D] = region_analysis(d_tof,det_frequ_D,frequ_search_bin_D);
122     fft_analyse(frequ_mean_bereich_D,fs,'D_Bereichanalyse');
123     if sum(frequ_mean_bereich_D(:))==0
124         disp('Zu wenig mit Signalabstand detektiert es wird jede etektierte Frequenz angezeigt')
125         frequ_search_bin_D(:) = 1;
126     end
127
128     frequ_search_A = rausch_filter(det_frequ_A,M_groesse_rausch,step_size,toleranz,r_filterpar,fs);
129     frequ_search_D = rausch_filter(det_frequ_D,M_groesse_rausch,step_size,toleranz,r_filterpar,fs);
130
131     %% Filter verk\upfen %%
132     frequ_anz_bin_A = frequ_search_A.*frequ_search_bin_A;
133     frequ_anz_bin_D = frequ_search_D.*frequ_search_bin_D;
134
135     frequ_anz_A = det_frequ_A.*frequ_anz_bin_A;
136     frequ_anz_D = det_frequ_D.*frequ_anz_bin_D;
137
138     %% Histogram f\ur Auslenkungserfassung%%
139     if (frequ_bin_kreis)
140         ratio = sum(frequ_anz_bin_D(:))/sum(mm(:));
141         disp(['Es sind ' num2str(ratio*100) ' Prozent der Pixel erkannt worden'])

```

```

142     LOG_D = logical(mm);
143 else
144     LOG_A = ones(size(frequ_anz_bin_A));
145     LOG_D = ones(size(frequ_anz_bin_D));
146
147 end
148
149     %% erstellen des Histogramms %%
150 oo =reshape(frequ_anz_A,[],1);
151 histogram_fig(oo,'Histogram A',0.1);
152 clear oo
153
154 oo =reshape(frequ_anz_D,[],1);
155 histogram_fig(oo,'Histogram D',0.1);
156 clear oo
157
158 oo =reshape(db_mat_A,[],1);
159 histogram_fig(oo(LOG_A),'Histogram dB-Mat A',0.1)
160 clear oo
161
162 oo =reshape(db_mat_D,[],1);
163 histogram_fig(oo(LOG_D),'Histogram dB-Mat D',0.1)
164 clear oo
165
166 %% Farbgebung der Frequenzen %%
167 if (frequ_bin_kreis)
168     [c_A] = einfaerben(a_tof_roh_alt,frequ_anz_A,frequ_anz_bin_A,fs);
169     [c_D] = einfaerben(d_tof_roh_alt,frequ_anz_D,frequ_anz_bin_D,fs);
170 else
171     a_tof_roh_alt = a_tof_roh;
172     d_tof_roh_alt = d_tof_roh;
173     [c_A] = einfaerben(a_tof_roh,frequ_anz_A,frequ_anz_bin_A,fs);
174     [c_D] = einfaerben(d_tof_roh,frequ_anz_D,frequ_anz_bin_D,fs);
175 end
176
177 %% Ausgabe 3D und 2D Daten %%
178 f3D_plot_vib(a_tof_roh,c_A,'A',fs)
179 f3D_plot_vib(d_tof_roh,c_D,'D',fs)
180
181 %% 2D Bild %%
182 c_A2D = flipud(permute(c_A,[2,1,3]));
183 c_D2D = flipud(permute(c_D,[2,1,3]));
184
185 fig3 = figure('Name','2D Abstand','NumberTitle','off');
186 set(fig3, 'Position', [300, 150, 700, 630]);
187 % Create axes
188 axes3 = axes('Parent',fig3);
189 image(c_A2D)
190 xlabel('horizontale Pixel','FontSize',30,'Interpreter','latex')
191 ylabel('vertikale Pixel','FontSize',30,'Interpreter','latex')
192 xlim([0 size(c_A2D,2)])
193 ylim([0 size(c_A2D,1)])
194
195 box(axes3,'on');
196 % Set the remaining axes properties
197 set(axes3,'FontSize',30,'TickLabelInterpreter','latex');

```

```
199
200     fig3 = figure('Name','2D Abstand','NumberTitle','off');
201     set(fig3, 'Position', [300, 150, 700, 630]);
202     % Create axes
203     axes3 = axes('Parent',fig3);
204     image(c_D2D)
205     xlabel('horizontale Pixel','FontSize',30,'Interpreter','latex')
206     ylabel('vertikale Pixel','FontSize',30,'Interpreter','latex')
207     xlim([0 size(c_D2D,2)])
208     ylim([0 size(c_D2D,1)])
209
210     box(axes3,'on');
211     % Set the remaining axes properties
212     set(axes3,'FontSize',30,'TickLabelInterpreter','latex');
213
214     else
215         %% Histogramm Auswertung des Genauigkeit der Messung ohne Vibration %%
216         d_tof_median = median_filter(d_tof);
217         d_tof_median_roh = median_filter(d_tof_roh);
218         close all
219         clc
220
221         f3D_plot_vib(d_tof_roh,0,'A',fs)
222         f3D_plot_vib(d_tof_roh,c_D,'D',fs)
223
224         disp('d_tof_roh')
225         streuung_messdaten(d_tof_roh,0.65);
226         disp('d_tof_medfilt_roh')
227         streuung_messdaten(d_tof_median_roh,0.65);
228
229         disp('d_tof')
230         streuung_messdaten(d_tof,0.65);
231         disp('d_tof_median')
232         streuung_messdaten(d_tof_median,0.65);
233
234         mean(a_tof_roh(:))
235         std(a_tof_roh(:))
236     end
237 end
```

6.2 Bildrate virtuell erhöhen

```

1 function [amp_data_raise] = fps_raise(amp_data, dcs_type)
2
3 b_size = size(amp_data);
4
5 b_size(3) = (b_size(3)-dcs_type)*2;
6
7 amp_data_raise = nan(b_size);
8
9 uu = 0;
10
11 for kk = 1:size(amp_data,3)/4-1
12
13     uu = uu+1;
14     amp_data_raise(:,:, (uu*4-3:uu*4)) = amp_data(:,:, (kk*4-3:kk*4));
15     uu = uu+1;
16     amp_data_raise(:,:, (uu*4-3:uu*4)) = (amp_data(:,:, kk*4-3:kk*4) + amp_data(:,:, (kk+1)*4-3:(kk
17         +1)*4))/2;
18
19 end
20
end

```

6.3 Daten laden

```

1 function [part_roh] = load_dcs(filename)
2     data = load(filename);
3
4     datasize = size(data);
5
6     col = datasize(2);
7
8     row = 252;
9
10    u = reshape(data.', col, row, []);
11
12    [posx, posy] = find(~isnan(u(:,:,1)));
13
14    viewx = [posx(1), posx(end)];
15    viewy = [posy(1), posy(end)];
16
17    part_roh = u(viewx(1):viewx(2), viewy(1):viewy(2), :);
18 end

```

6.4 Daten berechnen

```

1 function [d_tof,a_tof_pp,bilder] = roh_daten_auswertung(amp_data,dcstype,offset)
2
3 %% Modulationsfrequenz %%
4 frequ = 24;%MHz
5
6 %% Distanz und Amplitudenqualit\”at %%
7
8 xyz_size = size(amp_data);
9 xyz_size(3) = xyz_size(3);
10 dcs_bilder = xyz_size(3);
11
12 bilder = dcs_bilder/dcs_type;
13
14 if (dcs_type == 4)
15
16     amp(:,:,1) = amp_data(:,:,1:4:dcs_bilder);
17     amp(:,:,2) = amp_data(:,:,2:4:dcs_bilder);
18     amp(:,:,3) = amp_data(:,:,3:4:dcs_bilder);
19     amp(:,:,4) = amp_data(:,:,4:4:dcs_bilder);
20     flight_time = (pi+atan2(amp(:,:,4)-amp(:,:,2),amp(:,:,3)-amp(:,:,1)))/(2*pi*frequ);
21     a_tof_pp = sqrt(((amp(:,:,3)-amp(:,:,1)).^2+(amp(:,:,4)-amp(:,:,2)).^2)/4);
22
23 elseif (dcs_type == 2)
24     amp(:,:,1) = amp_data(:,:,1:2:dcs_bilder);
25     amp(:,:,2) = amp_data(:,:,2:2:dcs_bilder);
26     flight_time = (pi+atan2(amp(:,:,2),amp(:,:,1)))/(2*pi*frequ);
27     a_tof_pp = sqrt((amp(:,:,1).^2+amp(:,:,2).^2)/2);
28
29 end
30
31 c = 299.792458;
32
33 d_tof = c*flight_time/2;
34 d_tof = d_tof*100-offset; % cm
35
36 end

```

6.5 Ableitungsmethode

```

1 function [data_d] = ableitungsmethode(data,integral)
2 data_d = diff(data,3);
3     if (integral)
4         data_d = cumsum(data_d,3);
5     end
6 end

```


6.6 Median-Filter

```

1 function [data_medfilt] = median_filter(data)
2 bilder_anz = size(data,3);
3 data_medfilt = nan(size(data));
4     for k = 1:bilder_anz
5         data_medfilt(:,k) = (medfilt2(data(:,k),[5 5]));
6     end
7 end

```

6.7 Frequenzdetektion

```

1 function [det_frequ,frequ_search_bin,det_frequ_fwhm,db_mat] = frequ_analysis(data,fs,SRAbstand)
2
3 xyz_size = size(data);
4 det_frequ = zeros(xyz_size(1),xyz_size(2));
5 det_au = zeros(xyz_size(1),xyz_size(2));
6 db_mat = zeros(xyz_size(1),xyz_size(2));
7 det_frequ_fwhm = zeros(xyz_size(1),xyz_size(2));
8
9 if (numel(xyz_size) == 3)
10
11     for xf = 1:xyz_size(1)
12         for yf = 1:xyz_size(2)
13             pix_amp_data = reshape(data(xf,yf,:),1,[]);
14             int2 = pix_amp_data;
15             int2 = int2(end/2:end);
16             le = length(int2);
17             n = 2.^(ceil(log2(le)+1));
18             n2 = floor((n-le)/2);
19             int1 = zeros(1,n);
20             int1(n2:n2+le-1) = int2;
21
22             l = numel(int1); % length of signal
23             y = fft(int1);
24
25             p2 = abs(y/l);
26             p1 = p2(1:l/2+1);
27             p1(2:end-1) = 2*p1(2:end-1);
28
29             f = fs*(0:(l/2))/l;
30
31             [pks,locs,w] = findpeaks(p1,f,'SortStr','descend','WidthReference','halfheight');
32             det_frequ_mean = mean(pks);
33
34             if ~isempty(pks)
35                 if (pks(1) > 0)
36                     %if (pks(1) > det_frequ_mean)
37                     det_au(xf,yf) = pks(1);
38                     det_frequ(xf,yf) = locs(1);
39                     db_mat(xf,yf) = 10*log10(det_au(xf,yf)/det_frequ_mean);
40                     det_frequ_fwhm(xf,yf) = w(1);
41                 %end

```

```

42         end
43     end
44 end
45 end
46
47 frequ_search_bin = db_mat >= SRAbstand;
48 det_frequ_fwhm(det_frequ_fwhm>1) = 0;
49 frequ_search_bin = imfill(frequ_search_bin,'holes');
50 end

```

6.8 Detektierte Frequenzregionen analysieren

```

1
2 function [frequ_mean_bereich] = region_analysis(amp_data,det_frequ,frequ_search_bin)
3     s = regionprops(frequ_search_bin);
4     center_frequ_pos = ceil(cat(1, s.Centroid));
5     Area = cumsum(ceil(cat(1, s.Area)));
6     %center_frequ = zeros(length(center_frequ_pos),1);
7     frequ_median = zeros(length(center_frequ_pos),1);
8
9     s = regionprops(frequ_search_bin,'PixelList');
10    pixellist_frequ_pos = cat(1,s.PixelList);
11
12    s = regionprops(frequ_search_bin,det_frequ,'PixelValues');
13    pixel_val = cat(1,s.PixelValues);
14
15    frequ_mean_bereich = 0;
16
17    amp_data(isnan(amp_data)) = 0;
18
19    for kk = 1:size(pixellist_frequ_pos,1)
20        frequ_mean_bereich = frequ_mean_bereich + reshape(amp_data(pixellist_frequ_pos(kk,2),
21            pixellist_frequ_pos(kk,1,:),1,[])/length(pixellist_frequ_pos);
22    end
23 end

```

6.9 Detektierte Frequenzregionen analysieren

```

1
2 function [frequ_search] = rausch_filter(det_frequ,anz_pixel,step_size,toleranz,r_filterpar,fs)
3     oneL = true(anz_pixel);
4     xyz_size = size(det_frequ);
5     frequ_search = zeros(xyz_size(1),xyz_size(2));
6     det_frequ((det_frequ>(0.9*fs/2))) = 0;
7
8     for lx = 0:step_size:xyz_size(1)-anz_pixel
9         for ly = 0:step_size:xyz_size(2)-anz_pixel
10            L = false(xyz_size(1),xyz_size(2));
11            L((1:anz_pixel)+lx,(1:anz_pixel)+ly) = oneL;
12            dummy_anz = det_frequ(L);
13            if dummy_anz(1) ~= 0

```

```

14         pix_num_anz = find(((dummy_anz(1)+toleranz > dummy_anz).* (dummy_anz >
15             dummy_anz(1)-toleranz))==true);
16         pix_frequ_ok = zeros(size(oneL));
17         pix_frequ_ok(pix_num_anz) = 1;
18         if (numel(pix_num_anz)> anz_pixel.^2*r_filterpar)%floor(anz_pixel^2/2+2))
19             frequ_search(L) = pix_frequ_ok;
20         end
21     end
22 end
23 end

```

6.10 Farbgebung der Pixel

```

1 function [c] = einfaerben(amp_data_roh,det_frequ,frequ_search,fs)
2     RGBspec = RGB_spectrum;
3     xf = size(amp_data_roh,1);
4     yf = size(amp_data_roh,2);
5
6     mm = mean(mean(amp_data_roh(:,:,end)));
7     amp_data_roh(amp_data_roh>mm*2) = mm*2;
8     farben_sw = medfilt2(amp_data_roh(:,:,end)-min(min(amp_data_roh(:,:,end))));
9     farben_sw = farben_sw/max(farben_sw(:));
10    c = farben_sw;%.*abs(frequ_search(:,:,1)-1);
11    c(:,:,2) = c(:,:,1);
12    c(:,:,3) = c(:,:,1);
13    farben = floor(det_frequ/(fs/2)*1020);
14    uu = RGBspec(farben+1,:)/255;
15
16    frequ_search(:,:,2) = frequ_search(:,:,1);
17    frequ_search(:,:,3) = frequ_search(:,:,1);
18
19    gg = reshape(uu,xf,yf,3).*frequ_search;
20    frequ_search_not = abs(frequ_search-1);
21
22    c(:,:,1) = c(:,:,1).*(frequ_search(:,:,1)+frequ_search_not(:,:,1))+gg(:,:,1)*0.7;
23    c(:,:,2) = c(:,:,2).*(frequ_search(:,:,2)+frequ_search_not(:,:,1))+gg(:,:,2)*0.7;
24    c(:,:,3) = c(:,:,3).*(frequ_search(:,:,3)+frequ_search_not(:,:,1))+gg(:,:,3)*0.7;
25
26    c_max = max(c(:));
27    c = c/c_max;
28 end

```

6.11 Zusätzliche Funktionen

6.11.1 Farben-RGB

```

1 function RGBspec = RGB_spectrum()
2     RGBspec = zeros(1021,1,3);
3     RGBspec(1:255,1,3)=255; %blau
4     RGBspec(1:255,1,2)=0:254; %gruen steigen
5     RGBspec(256:1020,1,2)=255; %gruen halten
6     RGBspec(256:510,1,3) = 254:-1:0; %blau absteigen
7     RGBspec(510:765,1,1)= 0:255; % rot steigen
8     RGBspec(766:1020,1,2) = 254:-1:0; %blau absteigen
9     RGBspec(766:1020,1,1)= 255; % rot halten
10    RGBspec(1020:1021,,:) = 0;% saettigung
11 end

```

6.11.2 3D-Plot und Point Cloud

```

1 function f3D_plot_vib(data_roh,c,name,fs)
2     A = (mean(data_roh,3));
3     xyz_size = size(A);
4     [xx,yy] = meshgrid(1:xyz_size(1),1:xyz_size(2));
5
6     %%figure Bild SW mit Frequenz%%
7     f3D = figure('Name',['3D Plot mit Detektierte Frquenzen: ' name], 'NumberTitle','off');
8
9     s = surf(xx.',yy.',A,c,'edgecolor','none');
10
11    xlabel('horizontale Pixel','FontSize',30,'Interpreter','latex')
12    ylabel('vertikale Pixel','FontSize',30,'Interpreter','latex')
13    ylabel('Abstand [cm]','FontSize',30,'Interpreter','latex')
14
15    RGBmap = reshape(RGB_spectrum,1021,3)/255;
16    colormap(RGBmap)
17    cb1 = colorbar;
18
19    yyy = get(s, 'YData');
20    zzz = get(s, 'ZData');
21    set(s, 'YData', zzz, 'ZData', yyy);
22
23    zlab = get(cb1,'ylabel');
24    set(zlab,'String','Frequenz / [Hz]','FontSize',30,'Interpreter','latex');
25    caxis([0 fs/2])
26
27    if (name == 'D')
28        ylim([40 200])
29        view([-20 20])
30    else
31        ylim([50 2000])
32        view([160 20])
33    end
34
35    set(cb1,'YTick',0:ceil(fs/2/10):fs/2);
36

```

```

37     x = xx.';
38     y = yy.';
39
40     r = c(:,1);
41     g = c(:,2);
42     b = c(:,3);
43
44     fig = figure('Name','Point-Cloud mit Colormap: ' name'],'NumberTitle','off');
45
46     axes1 = axes('Parent',fig);
47     hold(axes1,'on');
48     hold on
49
50     scatter3(x(:),A(:),y(:),10,[r(:) g(:) b(:)],'filled');
51     xlabel('Horizontale-Pixel','FontSize',20,'Interpreter','latex')
52     zlabel('Vertikale-Pixel','FontSize',20,'Interpreter','latex')
53     ylabel('Abstand [cm]','FontSize',20,'Interpreter','latex')
54     view([18 -20])
55
56     box(axes1,'on');
57     % Set the remaining axes properties
58     set(axes1,'FontSize',20,'TickLabelInterpreter','latex');
59     hold off
60 end

```

6.11.3 FFT-Analyse

```

1 function [f,p1N] = fft_analyse(data,fs,daten_name,max_x_show)
2     %% Zero Padding %%
3     le = length(data);
4     n = 2.^(ceil(log2(le)+1));
5     n_start = floor((n-le)/2);
6     int1 = zeros(1,n);
7     int1(n_start:n_start+le-1) = data;
8     int1 = data;
9     %% FFT-Analyse %%
10
11     l = numel(int1);
12     y = fft(int1);
13     p2 = abs(y/l);
14     p1 = p2(1:l/2+1);
15     p1(2:end-1) = 2*p1(2:end-1);
16     f = fs*(0:(l/2))/l;
17
18     %% Ausgabe Spektrum %%
19
20     N = max(p1);
21     fig = figure('Name','FFT Analyse: ' daten_name'],'NumberTitle','off');
22     % Create axes
23     axes1 = axes('Parent',fig);
24     hold(axes1,'on');
25     % Create plot
26     plot(f,p1/N,'LineWidth',2,'Color',[0 0 0])
27     % Create xlabel
28     xlabel('Frequenz [Hz]','FontSize',30,'Interpreter','latex')

```

```
29 % Create ylabel
30 ylabel('Normierte Amplitude [au]', 'FontSize', 30, 'Interpreter', 'latex')
31 ylim([0 1.1])
32 p1N = p1/N;
33
34 if exist('max_x_show')
35     xlim([0 max_x_show])
36 end
37 box(axes1, 'on');
38 % Set the remaining axes properties
39 set(axes1, 'FontSize', 30, 'TickLabelInterpreter', 'latex');
40
41 [pks, locs, w] = findpeaks(p1, f, 'SortStr', 'descend', 'WidthReference', 'halfheight');
42 uu = numel(pks);
43 if uu > 2
44     uu = 2;
45 end
46 for kk = 1:uu
47     disp(['Der h\"ochste Peak liegt bei: ' num2str(locs(kk)) ' mit einer FWHM
48         von: ' num2str(w(kk))])
49 end
```

6.12 Schematische Darstellung Algorithmus

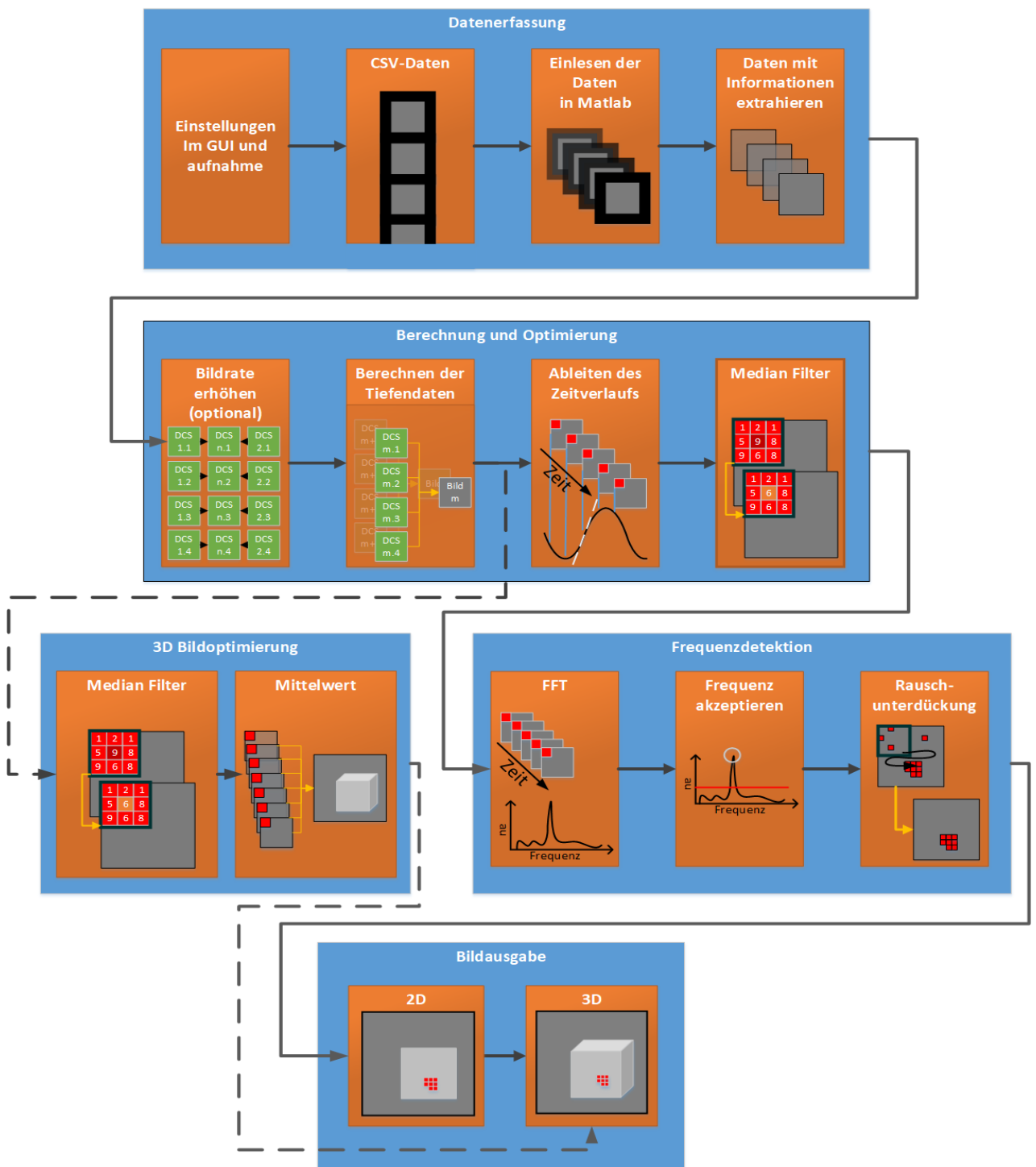


Abbildung 6.1: Gesamter Ablauf des Matlab-Algorithmus. Zuerst erfolgt die Datenerfassung dann Berechnung und Optimierung. Nach der Tiefendatenberechnung werden die Rohdaten separiert und einmal für die 3D-Darstellung sowie für die Frequenzdetektion optimiert. Nach der Berechnung der Frequenzen und der Tiefendaten werden die Daten zu einem 3D-Bild zusammengefügt.

Abkürzungsverzeichnis

TOF	Time-of-Flight	i
3D	drei Dimensionen	1
LED	Leuchtdiode	5
SPAD	Single-Photon-Avalanche-Diodes	5
DCS_x	Difference-Correlation-Sample 0-3	12
DCS	Difference-Correlation-Sample	14
ADC	Analog-Digital-Converter	14
LSB	Least-Significant-Bit	15
PMD	Photonic-Mixing-Device	17
FPN	Fixed-Pattern-Noise	21
fps	frame-per-second	25
NEG_{mbH}	Neumüller Elektronik GmbH	26
ESPROS	ESPROS-Photonics-Corporation	26
epc660	TOF-Imager-Chip epc660	26
DME660	Distance-Measurement-Engine	26
STM	STMicroelectronics International N.V.	27
GUI	Graphical-User-Interface	28
SoC	System-on-a-Chip	28
CCD	Charge-Coupled-Device	28
QVGA	Quarter-Video-Graphics Array	29
HDR	High-Dynamic-Range	31
ROI	Range-of-Interest	32
Thorlabs	Thorlabs Inc	33
FOV	Field-of-View	18
PNG	Portable-Network-Graphics	36
CSV	Comma-Separated-Values	36

NaN Not-a-Number	42
ASCII American-Standard-Code-for-Information-Interchange	42
FFT Fast-Fourier-Transform.....	47
SNR Signal-to-Noise-Ratio	52
dps dots-per-second	66
upm Umdrehungen pro Minute	70
FPGA Field-Programmable-Gate-Array	71
STFT Short-Time-Fourier-Transform	72
PMP Phase-Measuring-Profilometry	72

Literaturverzeichnis

- [AG15] ESPROS Photonics AG. DATASHEET – epc660 3D TOF Imager.
https://www.espros.com/downloads/01_Chips/Datasheet_epc660-V1.09.pdf, 2015.
geöffnet am: 05.03.2017.
- [AG16] ESPROS Photonics AG. DME660 Assembly 3D Data.
https://www.espros.com/downloads/02_Cameras_and_Modules/DME%20660_assembly_160120.STEP, 2016.
geöffnet am 03.09.2017.
- [AG17] ESPROS Photonics AG. Datasheet DME 660 V1.02.
https://www.espros.com/downloads/02_Cameras_and_Modules/Datasheet_DME_660-V1.02.pdf, 2017.
geöffnet am: 23.06.2017.
- [Alb07] Martin Albrecht. *Untersuchung von Photogate-PMD-Sensoren hinsichtlich qualifizierender Charakterisierungsparameter und -methoden*. Dissertation, Universität Siegen, 2007.
- [Bog14] Siegwart Bogatscher. *Time-of-Flight Laserkamera mit großer Empfangsapertur*. Dissertation, Karlsruher Institut für Technologie, 2014.
- [Bor16] Kai Borgeest. *Messtechnik und Prüfstände für Verbrennungsmotoren*. Springer Fachmedien Wiesbaden, 2016.
- [Cha10] E. Charbon. Million fps, Time-Correlated Single Photon Camera.
http://www.megaframe.eu/Contents/Publications/MEGAFRAME/MEGAFRAME_FinalActivityReport_v1_0FULL.pdf, 2010.
geöffnet am: 05.05.2017.

- [CHHL12] Ouk Choi, Miles Hansard, Radu Patrice Horaud, and Seungkyu Lee. *Time-of-Flight Cameras, Principles, Methods and Applications*. Springer, 2012.
- [CKTT08] Derek Chan, Young Min Kim, Sebastian Thrun, and Christian Theobalt. Design and calibration of a multi-view TOF sensor fusion system. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 00:1–7, 2008.
- [Col13] Gerald Coley. BeagleBone Black System Reference Manual. cdn.sparkfun.com/datasheets/Dev/Beagle/BBB_SRM.pdf, 2013. geöffnet am: 05.06.2017.
- [Com15] EU Commission. Time-of-Flight based on Single-Photon detectors: a long journey from the lab to the market. <https://ec.europa.eu/digital-single-market/en/news/time-flight-based-single-photon-detectors-long-journey-lab-market>, 2015. geöffnet am: 16.5.2017.
- [DiT06] AVL DiTEST. Sales Information for Multisense 1000. [https://www.avlditest.com/index.php/en/products-mu-en-multisense-1000.html?file=files/content/downloads/prospekte/prospekte-en/Salesinformation-AVL-DiTEST-Mulisense 1000-EN.pdf](https://www.avlditest.com/index.php/en/products-mu-en-multisense-1000.html?file=files/content/downloads/prospekte/prospekte-en/Salesinformation-AVL-DiTEST-Mulisense%201000-EN.pdf), 2006. geöffnet am: 30.03.2017.
- [FAT11] S. Foix, G. Alenya, and C. Torras. Lock-in Time-of-Flight (ToF) Cameras: A Survey. *IEEE Sensors Journal*, 11(9):1917–1926, 2011.
- [FPB⁺16] P. Fürsattel, S. Placht, M. Balda, C. Schaller, H. Hofmann, A. Maier, and C. Riess. A Comparative Error Analysis of Current Time-of-Flight Sensors. *IEEE Transactions on Computational Imaging*, 2(1):27–41, 2016.
- [fWuA06] Bundesministerium für Wirtschaft und Arbeit. VIBRATIONEN - Wirkungen und Gefahren für die Gesundheit. <http://www.arbeitsinspektion.gv.at/cms/inspektorat/download.html>, 2006. geöffnet am: 30.04.2017.

- [GBG16] M. Georgiev, R. Bregović, and A. Gotchev. Fixed-Pattern Noise Modeling and Removal in Time-of-Flight Sensing. *IEEE Transactions on Instrumentation and Measurement*, 65(4):808–820, 2016.
- [Gra11] Michael Grabe. *Grundriss der Generalisierten Gauß’schen Fehlerrechnung*. Springer Berlin Heidelberg, 2011.
- [GVHS01] P. Gulden, M. Vossiek, P. Heide, and R. Schwarte. Novel opportunities for optical level gauging and 3D-imaging with the photoelectronic mixing device. In *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No.01CH 37188)*, volume 3, pages 2099–2103 vol.3, 2001.
- [Hen11] Norbert Henze. *Stochastik für Einsteiger: Eine Einführung in die faszinierende Welt des Zufalls*. Vieweg+Teubner Verlag, 2011.
- [HKE14] Stephan Hussmann, Florian Knoll, and Torsten Edeler. Modulation method including noise model for minimizing the wiggling error of TOF cameras. *IEEE transactions on instrumentation and measurement*, 63(5):1127–1136, 2014.
- [LK07] Marvin Lindner and Andreas Kolb. Calibration of the intensity-related distance error of the PMD TOF-camera. *Proceedings of SPIE - The International Society for Optical Engineering*, 2007.
- [LSKK10] Marvin Lindner, Ingo Schiller, Andreas Kolb, and Reinhard Koch. Time-of-Flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding*, 2010.
- [Mac79] Charles E. Mackenzie. *Coded Character Sets: History and Development*. Addison-Wesley Pub (Sd), 1979.
- [Mar09] Othmar Marti. Lichtgeschwindigkeit im Medium und Intensität. <http://wwwex.physik.uni-ulm.de/lehre/el-2009/node38.html>, 2009. geöffnet am: 05.06.2017.
- [MDH⁺09] Stefan May, David Droschel, Dirk Holz, Stefan Fuchs, Ezio Malis, Andreas Nüchter, and Joachim Hertzberg. Three-dimensional mapping with Time-of-Flight cameras. *Journal of Field Robotics*, 26(11-12):934–965, 2009.

- [Mos08] Hanns Moshhammer. Wenn Schall und Vibrationen krank machen. <http://www.springermedizin.at/artikel/9867-wenn-schall-und-vibrationen-krank-machen>, 2008.
geöffnet am: 05.06.2017.
- [MSMF14] M. Maggioni, E. Sánchez-Monge, and A. Foi. Joint Removal of Random and Fixed-Pattern Noise Through Spatiotemporal Video Filtering. *IEEE Transactions on Image Processing*, 23(10):4282–4296, 2014.
- [NAS69] NASA. Apollo 11 preliminary science report nasa sp-214. Technical report, National Aeronautics And Space Administration, 1969.
- [NIS16] CODATA NIST. Fundamental Physical Constants, speed of light in vacuum. <https://physics.nist.gov/>, 2016.
geöffnet am: 05.06.2017.
- [OSR16] OSRAM. OSOLON Black Series (850 nm) V 1.6. https://www.osram-os.com/Graphics/XPic0/00239727_0.pdf, 2016.
geöffnet am: 03.09.2017.
- [PR16] Namdev A Patil and Laukik P Raut. Vibration Analysis of Ci Engine Using Fft Analyzer. *International Journal of Research in Engineering and Technology*, pages 293–298, 2016.
- [RDP⁺11] M. Reynolds, J. Doboš, L. Peel, T. Weyrich, and G. J. Brostow. Capturing Time-of-Flight data with confidence. In *CVPR 2011*, pages 945–952, 2011.
- [Rei12] Konrad Reif. *Sensoren im Kraftfahrzeug*. Vieweg+Teubner Verlag, 2012.
- [RS09] Community Research and Development Information Service. Million frame per second, time-correlated single photon camera, Project ID: 029217. http://cordis.europa.eu/project/rcn/79434_en.html, 2009.
geöffnet am: 23.09.2017.
- [RS14] Erik Rodner and Herbert Süße. *Bildverarbeitung und Objekterkennung*. Springer Fachmedien Wiesbaden, 2014.
- [Sch03] Bernd Schneider. *Der Photomischdetektor zur schnellen 3D-Vermessung für Sicherheitssysteme und zur*. Dissertation, Universität-Gesamthochschule Siegen, 2003.

- [Ser17] International Laser Ranging Service. Satellite Laser Ranging (SLR) and Lunar Laser Ranging (LLR). <https://ilrs.cddis.eosdis.nasa.gov/>, 2017. geöffnet am: 08.04.2017.
- [Sie16] Inge Siegl. Capacitive charge pump based low power pipeline adc. Masterarbeit, Technische Universität Graz, 2016.
- [SP07] Agnes Swadzba and Jochen Penne. A Comprehensive System for 3D Modeling from Range Images Acquired from a 3D ToF Sensor. In *International Conference on Computer Vision Systems (ICVS 2007)*, 2007.
- [STM14] STMicroelectronics. Industry-Leading Proximity Sensor from STMicroelectronics Brings Superior Distance Sensing to Mobile Phones, Consumer White Goods, and Industrial Applications. http://www.st.com/content/st_com/en/about/media-center/press-item.html/p3609p.html, 2014. geöffnet am: 14.05.2017.
- [STM16] STMicroelectronics. Datasheet - Production Data - VL53L0X. <http://www.st.com/content/ccc/resource/technical/document/datasheet/group3/b2/1e/33/77/c6/92/47/6b/DM00279086/files/DM00279086.pdf/jcr:content/translations/en.DM00279086.pdf>, 2016. geöffnet am: 10.06.2017.
- [SZ01] Jack A. Stone and Jay H. Zimmerman. Index of Refraction of Air. <http://emtoolbox.nist.gov/Wavelength/Documentation.asp>, 2001. geöffnet am: 05.06.2017.
- [Tho17] Thorlabs. 3D-Modelle von Thorlabs (MB30120/M, BA2/M, PH75/M, TR75/M, BA15/M). www.thorlabs.com, 2017. geöffnet am: 25.08.2017.
- [uGS14] Stefan Hesse und Gerhard Schnell. *Sensoren für die Prozess and und Fabrikautomation and Funktion and Ausführung and Anwendung*. Springer Fachmedien Wiesbaden, 2014.

- [VCGRV17] I. Vornicu, R. Carmona-Galán, and A. Rodríguez-Vázquez. Photon counting and direct ToF camera prototype based on CMOS SPADs. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, 2017.
- [ZTZ⁺17] Guomin Zhan, Hongwei Tang, Kai Zhong, Zhongwei Li, Yusheng Shi, and Congjun Wang. High-speed FPGA-based phase measuring profilometry architecture. *Opt. Express*, 25(9):10553–10564, 2017.