

Bachelor Arbeit

Hochschule für Technik und Wirtschaft des Saarlandes  
Implementierung und Validierung eines Synthetische Apertur  
Fokussierungs Technik - Algorithmus für die akustische  
Mikroskopie

Gutachter: Professor Dr. Robert Lemor  
Philipp Bellia  
Matrikelnummer: 38120321  
Studiengang: Biomedizintechnik  
Eleonorenstraße 4, 66119 Saarbrücken  
philipp.bellia@gmail.com

27. September 2023

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
1.1 Hintergrund und Motivation . . . . .	1
1.2 Zielsetzung der Arbeit . . . . .	2
1.3 Struktur der Arbeit . . . . .	2
<b>2 Grundlagen</b>	<b>3</b>
2.1 Physikalische Grundlagen . . . . .	3
2.1.1 Grundlagen der Akustik im Ultraschall . . . . .	3
2.1.2 Erzeugung von Ultraschall . . . . .	4
2.2 Akustische Mikroskopie . . . . .	5
2.2.1 Apertur . . . . .	5
2.2.2 Point Spread Function . . . . .	6
2.2.3 Bildrekonstruktion . . . . .	7
2.2.4 Aufbau eines akustischen Systems in der Mikroskopie . . . . .	10
2.3 Koordinatensysteme . . . . .	11
2.4 Prinzip der synthetischen Apertur . . . . .	14
2.4.1 Methode der synthetischen Apertur . . . . .	15
<b>3 Algorithmus der Synthetic Aperture Focusing Technique</b>	<b>20</b>
3.1 Algorithmische Durchführung . . . . .	20
3.1.1 Der Datensatz . . . . .	21
3.1.2 Der Algorithmus . . . . .	21
3.2 Methoden der synthetischen Apertur in der akustischen Mikroskopie . . . . .	25
<b>4 Ergebnisse</b>	<b>28</b>
4.1 Erwartungen . . . . .	28
4.2 Ergebnisse . . . . .	31
<b>5 Diskussion</b>	<b>35</b>
5.1 Interpretation der Ergebnisse . . . . .	35
5.2 Auswertung des Algorithmus . . . . .	36

<b>6 Fazit</b>	<b>38</b>
6.1 Ausblick . . . . .	39
<b>A Anhang</b>	<b>III</b>
A.1 Code zum Algorithmus . . . . .	III

# Abbildungsverzeichnis

2.1	Die Abbildung zeigt eine Skizze eines Ultraschallwandlers. Die rote Linie soll die Aperturöffnung darstellen. Eingezeichnet ist der Öffnungswinkel $\Theta$ . . . . .	5
2.2	Rekonstruktion der Punktspreizfunktion über ein Parametrisches Modell. Das Parametrische Modell bezieht sich auf die in Kapitel 2.4.1 erklärte Form der PSF und ist mathematisch betrachtet ein Rotationsvolumen, dass durch anpassen der Parameter der PSF des System gleicht. . . . .	6
2.3	Darstellung der Signale im Datensatz. Übersetzung des aufgenommenen Signals in z-Richtung . . . . .	7
2.4	Plot eines akustischen Signals (rot) und der dazu gehörigen Hilbert-Transformation (blau). Der rote Graph entspricht auch der Darstellung als A-Scan für diese Position. . . . .	9
2.5	Darstellung der Schnittbilder im Datensatz. grau: entlang der x-Achse, blau: entlang der y-Achse . . . . .	9
2.6	Beispiel von B-Scans einer 1 Cent-Münze, (a) B-Scan entlang der x-Achse, (b) entlang der y-Achse . . . . .	10
2.7	Zwei C-Scans eines akustischen Datensatzes. Wie im B-Scan wurde eine 1-Cent Münze gescannt. Die Linien entsprechen den Linien in der Abbildung 2.6 . . . . .	10
2.8	easySam Scanner. Man kann auf der Abbildung sehen, das die Linse am Scanner befestigt ist. Die Linse ist am unteren Teil des Scanners befestigt. . . . .	11
2.9	Skizze des Wandlers. Das Wandlerelement (orange) sitzt auf dem Saphir und sendet dort Schallwellen hinein (rot). Diese werden an der Apertur fokussiert und ausgesendet. . . . .	11
2.10	Skizze des elektronischen Koordinatensystems. Es wird gezeigt, dass der Bezugspunkt zum Rekonstruktionspunkt am Wandleranfang sitzt. . . . .	12
2.11	Das Linsen Koordinatensystem. Die Abbildung zeigt den verschobenen Nullpunkt auf den Austrittspunkt des Schalls. . . . .	13
2.12	Skizze des Delay-Koordinatensystems. Die Skizze zeigt, dass der Nullpunkt des Delay-Koordinatensystems auf dem Hardware-Delay liegt. . . . .	14
2.13	Abstandsveränderung $r$ wenn der Ultraschallwandler von Position P1 auf P2 wandert. Der ursprüngliche Abstand $r_1$ (blau) ist kürzer als der neue Abstand $r_2$ (orange). Die Überschneidungen von Nah- und Fernfeld zwischen den beiden Positionen sind in grün zu sehen. . . . .	16

2.14	Der Signalanteil, der aufsummiert werden muss . . . . .	17
2.15	Skizze der e-Funktionen über die z-Achse. Die zwei roten Linien sollen die Form des Schallfeld über die z-Achse skizzieren. . . . .	18
2.16	Skizze des Prinzips der synthetischen Apertur. Die Abbildung zeigt die wichtigen Abstände, die für die Berechnung der zeitlichen Unterschiede wichtig sind. (0) ist die Step size, die der Wandler zurück legt, (1) ist der eingestellte Hardware-Delay, (2) ist die Differenz zwischen dem Hardware-Delay und der Sampleposition des ersten Vorlaufechos der Linse, (3) Abstand von der nächsten Wandlerposition zum vorherigen Rekonstruktionspunkt, (4) Abstand vom der Ausgangswandlerposition zum Rekonstruktionspunkt, (5) Nebensignalpunkt zum Zeitpunkt [t], (6) Zeitliche Differenz zwischen dem Rekonstruktionspunkt der Ausgangswandlerposition und der Nebensignalwandlerposition, (7) Signalanteil aus dem Nebensignal zum Zeitpunkt [t+dt] der mit einer Gewichtung auf den Rekonstruktionspunkt aufaddiert werden soll, (8) Zeitpunkt [t + dt], (9) Zeitpunkt [t], (10) Rekonstruktionspunkt der Position von der Ausgangsposition des Wandlers, (11) Aperturdurchmesser der Linse . . . . .	19
3.1	Ergebnis ADT-SAFT, kopiert aus: [1] . . . . .	26
3.2	Ergebnis des 1D - SAFT, SAFT+CF, kopiert aus: [3] . . . . .	27
4.1	B-Scan mit der Probe im Nahfeld. . . . .	29
4.2	B-Scan mit der Probe im Fernfeld . . . . .	30
4.3	Links: B-Scan der Hilbert Daten, Mitte: SAFT-B-Scan, Rechts: Subtraktion des B-Scans und des SAFT B-Scans . . . . .	31
4.4	Auf der linken Seite sieht man den B-Scan in den Hilbert-Daten und rechts den errechneten SAFT B-Scan. Unter den Bildern sind jeweils die zugehörigen Histogramme . . . . .	32
4.5	Links: B-Scan der Hilbert Daten, mitte: SAFT-B-Scan, Rechts: Subtraktion des B-Scans und des SAFT B-Scans . . . . .	32
4.6	Auf der linken Seite sieht man den B-Scan aus den Hilbert-Daten und rechts den errechneten SAFT B-Scan. Unter den Bildern sind jeweils die zugehörigen Histogramme . . . . .	33
4.7	Abbildung zeigt das Normierungartefakt, welches sichtbar wird, wenn man über die Anzahl an Pixeln normiert, die für die synthetische Apertur betrachtet wurden . . . . .	34
A.1	Importierte Bibliotheken . . . . .	III
A.2	Funktion die die Hilbert-Transformation anwendet . . . . .	IV
A.3	Python Code zu Algorithmus 2 . . . . .	V
A.4	Python Code zu Algorithmus 2 . . . . .	VI
A.5	Zuteilung der Aufgaben für jeden Prozess . . . . .	VII

A.6	Python Code zu Algorithmus 1 . . . . .	VIII
A.7	Python Code zu Algorithmus 1 . . . . .	IX
A.8	Start des Algorithmus . . . . .	IX

# Kapitel 1

## Einleitung

Diese Arbeit thematisiert die Entwicklung eines Algorithmus, dessen Aufgabe es ist die Auflösung in der akustischen Mikroskopie zu verbessern. Dafür wird das Prinzip der synthetischen Apertur verwendet und in der Praxis umgesetzt.

### 1.1 Hintergrund und Motivation

Die akustische Mikroskopie findet, neben der Forschung in der Biologie und Krebsforschung, Anwendung in den Bereichen der Materialprüfung [10]. Akustische Mikroskope fokussieren Ultraschallwellen auf einen Bereich von Interesse und rastern diesen ab. Hierbei werden die Ultraschallwellen an Grenzflächen reflektiert und von dem Wandler wieder aufgenommen. [7, 9]. In der akustischen Mikroskopie werden Ultraschallwandler verwendet um mikroskopische Aufnahmen von biologischen Proben, wie Krebszellen zu machen. Diese Ultraschallwandler haben den Vorteil gegenüber herkömmlichen Mikroskopie-Methoden, dass man, wie auch beim klinischen Ultraschall, Schnittbilder der Probe machen kann, um an Tiefeninformationen zu gelangen [7]. Des Weiteren ist die Methode pathologisch unbedenklich für sämtliche Proben [7]. Eben genannte Ultraschallwandler geben mit einer Frequenz von 800 MHz bis 2 GHz Ultraschallwellen ab, die auf einen bestimmten Bereich gebündelt werden. Dieser Bereich wird auch Fokusbereich genannt und ist in den Experimenten, die hier durchgeführt wurden, fünf Mikrometer breit. Der Fokusbereich ist abhängig vom verwendeten System und dessen Einstellungen. In dem hier verwendeten wird die bestmögliche Auflösung mit einem 5 mm großen Fokusbereich erreicht, da damit die Punktspreizfunktion am schmalsten ist. Da nicht nur im Fokusbereich Reflektionen auftreten, können durch die Signallaufzeit auch Tiefeninformationen gewonnen werden. Allerdings ist die Auflösung durch den größeren Bereich begrenzt, je weiter man sich von der Fokustiefe entfernt. Um dies zu verbessern wurde die Technik der synthetischen Apertur von der Radartechnik [8] auf die Ultraschall Mikroskopie übertragen [1, 3]. Außerhalb des Fokusbereiches, im Nah- und Fernfeld, wird die Auflösung immer schlechter, da man immer mehr Punkte in der Probe trifft, die ein Signal zurückgeben (Kapitel 2.4). An diesem Punkt greift nun die Aufgabenstellung der Arbeit. Die Implementierung eines synthetischen Apertur Algorithmus fokussiert das Nah- und Fernfeld und soll in diesen für eine

verbesserte Auflösung sorgen. Aktueller Stand der Forschung zu diesem Gebiet ist, dass es schon mehrere Ansätze für die Technik gibt [1, 3]. Genauer wird in Kapitel 3 auf diese Techniken eingegangen, doch im Wesentlichen ist das Vorgehen ähnlich wie in dieser Arbeit. Alle Ansätze verwenden die Tatsache, dass man Teileinformationen aus Nebensignalen extrahieren kann. Dabei muss die Signallaufzeit beachtet werden. Extrahierte Signalanteile werden mit einer Gewichtung multipliziert. Das Vorgehen, wie man diese Gewichtung berechnet ist einer der Unterschiede zu dieser Arbeit. Ameri Golafsoun wichtet mittels Apodisation [1]. Der Ansatz von Gao Rongkang (et. al) berechnet anhand der gebildeten Summen einen Wichtungskoeffizienten für jede Position [3]. In vorliegenden Arbeit wird die Gewichtung mittels der systemeigenen Punktspreizfunktion ermittelt.

## 1.2 Zielsetzung der Arbeit

Diese Arbeit hat das Ziel, eine erfolgreiche Modellierung der Punktspreizfunktion für das vorliegende System zu erarbeiten. Ein weiteres Ziel ist, ein räumliches Gitter zu erstellen, welches alle Punkte beinhaltet, die von dem Ultraschallwandler angefahren werden, dieses Ziel ist wichtig für die Rekonstruktion der Daten. Genau so wichtig ist, eine Methode zu definieren, die vorgibt, wie man zu jedem relevanten Punkt alle wichtigen Parameter berechnet. Das abschließende Ziel der Arbeit ist die Validierung und die Auswertung des Algorithmus.

## 1.3 Struktur der Arbeit

Um die Zielsetzungen der Arbeit zu erreichen, werden zuerst die Grundlagen erklärt. Diese physikalischen Grundlagen sind wichtig um der Arbeit zu folgen und bilden die theoretische Grundlage, für die darauf folgende Methodik. An die Vorstellung der Methodik schließt sich die Präsentation der Ergebnisse an. Den Abschluss der Arbeit bildet die Diskussion der Ergebnisse und ein Fazit.

# Kapitel 2

## Grundlagen

In diesem Kapitel werden die verschiedenen Grundlagen erklärt, die benötigt werden um der Arbeit zu folgen. Dazu wird als erstes auf die physikalischen Grundlagen eingegangen und dieses Kapitel kann in zwei thematische Schwerpunkte aufgeteilt werden. Zum einen in das Kapitel 2.1.1 Grundlagen der Akustik im Ultraschall und zum anderen in das Kapitel 2.1.2 Erzeugung von Ultraschall. Diese Grundlagen sind relevant für das allgemeine Verständnis von einer Ultraschallaufnahme. In Kapitel 2.2 wird auf Grundlagen zur Bildrekonstruktion und Darstellungsmöglichkeiten eingegangen. Alle Aspekte werden benötigt um Kapitel 2.4 folgen zu können. Das Kapitel 2.4 erörtert das Prinzip der synthetischen Apertur.

### 2.1 Physikalische Grundlagen

Dieses Kapitel stellt die physikalischen Grundlagen vor, die benötigt werden, um der Arbeit zu folgen. Die physikalischen Aspekte die für diese Arbeit relevant sind beziehen sich auf die Akustik im Ultraschall. Die Akustik erforscht das Verhalten und die Ausbreitung von Schallwellen in verschiedenen Medien. Sie untersucht zudem, wie diese Wellen in spezifischen Umgebungen reflektiert werden. Dieser Aspekt ist auch für die Bildgebung mittels Ultraschallwellen von großer Bedeutung.

#### 2.1.1 Grundlagen der Akustik im Ultraschall

Abhängig vom vorherrschendem Medium breiten sich Ultraschallwellen unterschiedlich schnell aus. Man unterscheidet grob die Mediums-Zustände gasförmig, flüssig und fest. Im klinischen Ultraschall wird der Schall in komplexere Gewebestrukturen geschickt, diese weisen mehrere Grenzschichten auf. Die Impedanz, also der Wellenwiderstand  $Z$ , in einem Medium berechnet sich wie folgt

$$Z_i = \rho_i * c_i, \quad (2.1)$$

wobei  $\rho$  für die Dichte des Mediums steht und  $c$  für die Schallgeschwindigkeit im Medium [7]. An Grenzschichten wird der ausgesendete Schall reflektiert, gebrochen und gestreut. Trifft der Schall senkrecht auf eine Grenzfläche, so kann die Intensität des reflektierten Echos ( $I_R$ ), relativ zur einfallenden Intensität ( $I_0$ ) also der Reflexionskoeffizient  $R$  mit

$$R = \frac{I_R}{I_0} = \left( \frac{Z_2 - Z_1}{Z_1 + Z_2} \right)^2, \quad (2.2)$$

beschrieben werden.  $Z_1$  und  $Z_2$  stehen für die unterschiedlichen Impedanzen der aneinander grenzenden Gewebe [9]. Im Folgenden wird immer mit der Schallgeschwindigkeit  $1524 \frac{m}{s}$  gerechnet, weil das Kopplungsmedium Wasser ist [9]. Das Kopplungsmedium ist eine Flüssigkeit oder ein Gel, welches verwendet wird, um eine direkte Verbindung vom Wandler zur Probe herzustellen. Dies macht man, damit der Schall nicht auf Luft trifft und es so zu negativen Einflüssen auf die Qualität der Signale kommt [7].

Die Schallwellen kommen aber nicht widerstandslos durch das Medium. Sie erfahren eine Dämpfung, die miteinbezogen werden muss, wenn sie aus der Tiefe des beschallten Mediums wieder zurück kommen. Diese Dämpfung wird angegeben mit:

$$\frac{I(d, f)}{I_0} = e^{-2\alpha(f)2d}, \quad (2.3)$$

[9]

wobei  $d$  die Tiefe ist, aus der der Schall zurück kommt und  $\alpha(f)$  der Dämpfungskoeffizient. Dieser ist abhängig von der Frequenz ( $f$ ) des Schalls.

### 2.1.2 Erzeugung von Ultraschall

Heutzutage werden Ultraschallwellen in der Medizin und Technik überwiegend von piezoelektrischen Bauelementen erzeugt und empfangen [7, 9]. Deren Funktionsweisen beruhen auf dem piezoelektrischen Effekt und auf dem inversen piezoelektrischen Effekt [6]. Die Theorie dahinter sagt, dass ein piezoelektrischer Kristall sich beim anlegen einer elektrischen Spannung verformt, dies nennt sich inverser Effekt [6]. Gegensätzlich hierzu gibt der Kristall eine elektrische Spannung ab, wenn eine mechanische Spannung anliegt [6]. Man erreicht eine bestimmte Frequenz, indem man eine Wechselspannung mit der Frequenz an das Piezoelement anlegt. Die akustische Welle wird ausgesendet und an Grenzflächen reflektiert. Das Piezoelement, kann nun durch den piezoelektrischen Effekt, die mechanische Spannung durch die reflektierte Welle in ein elektrisches Signal umwandeln.

Die Intensität und Form des Schallfeldes, welches abgegeben wird, hängt von der Geometrie des Ultraschallwandlers ab [9]. Die Apertur eines Wandlers, also die effektive Fläche des Wandlerelementes und die Arbeitsfrequenz des Systems, bestimmen die räumliche Verteilung

des Schalls [9]. Die entstehende Schallkeule kann in drei Bereiche gegliedert werden. Direkt am Wandler ist das Nahfeld. Hier gibt es durch Signalinterferenzen ungenaue Informationen über den beschallten Bereich. Die Grenze zwischen dem Nahfeld und dem Fernfeld ist der Fokusbereich. Die Position  $L_N$  des Fokusbereichs ist abhängig von der Apertur, der Sendefrequenz und der Schallgeschwindigkeit des vorliegendem Mediums, dabei gilt:

$$L_N = \frac{a^2 f}{4c} = \frac{a^2}{4\lambda}, \quad (2.4)$$

wobei  $a$  für die Apertur,  $f$  für die Sendefrequenz,  $c$  für die Schallgeschwindigkeit und  $\lambda = \frac{c}{f}$  für die Wellenlänge steht [9].

## 2.2 Akustische Mikroskopie

Die akustische Mikroskopie ist ein bildgebendes Verfahren welches in wissenschaftlichen Bereichen verwendet wird. Damit können Zellen oder Zellorganismen genauer betrachtet werden. [4]. Die Akustische Mikroskopie unterscheidet sich von der herkömmlichen, d.h. optischen, Mikroskopie vor allem durch die Methode der Bilderzeugung. Während in der herkömmlichen Mikroskopie ein Licht und mindetens ein Okular verwendet werden um ein Bild zu erlangen, welches auch direkt mit dem Auge wahrgenommen werden kann, wird in der akustischen Mikroskopie Ultraschall und ein Ultraschallwandler verwendet. Dieser empfängt Daten, durch die ein Bild berechnet wird, das mit Hilfe eines Computermonitors angezeigt werden kann. Wie das Bild aus der Aufnahme gewonnen wird, wird in Kapitel 2.2.3 genauer behandelt.

### 2.2.1 Apertur

Unter der Apertur versteht man die Öffnung, des Wandlers, an dem die Schallwellen gebündelt und ausgesandt werden. Der Öffnungswinkel  $\Theta$  beeinflusst die Form des Schallfeldes und so

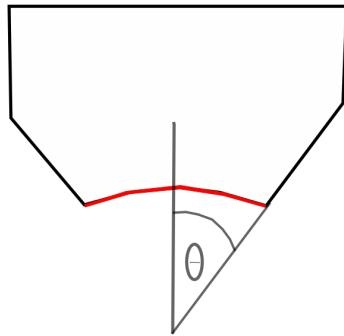


Abbildung 2.1: Die Abbildung zeigt eine Skizze eines Ultraschallwandlers. Die rote Linie soll die Aperturöffnung darstellen. Eingezeichnet ist der Öffnungswinkel  $\Theta$ .

auch die Fokustiefe des Wandlers. Je kleiner der Winkel wird, desto tiefer ist die Fokusebene.

Natürlich gilt dann auch, dass die Fokusebene näher der Apertur ist, wenn  $\Theta$  größer wird. Die synthetische Apertur ist nun ein mathematisches Verfahren, welches eine große Apertur durch viele kleine Aperturen berechnet. Mit dieser Technik ist es möglich eine bessere Auflösung zu erhalten. Auf spezifischere Erklärungen, wie die synthetische Apertur funktioniert, wird in Kapitel 2.4 eingegangen.

### 2.2.2 Point Spread Function

Jedes akustische System kann durch eine bestimmte Funktion beschrieben werden. Diese Funktion wird als Punktspreizfunktion (eng: Point-Spread-Function) bezeichnet. Faltet man ein Ultraschallphantom mit der Point-Spread-Function, so erhält man das Bild, das durch das System generiert werden würde. Visualisieren kann man die Punktspreizfunktion mit einem Scan eines infinitesimal kleinen Punkt. Das Resultat ist, die System eigene Ungenauigkeit.

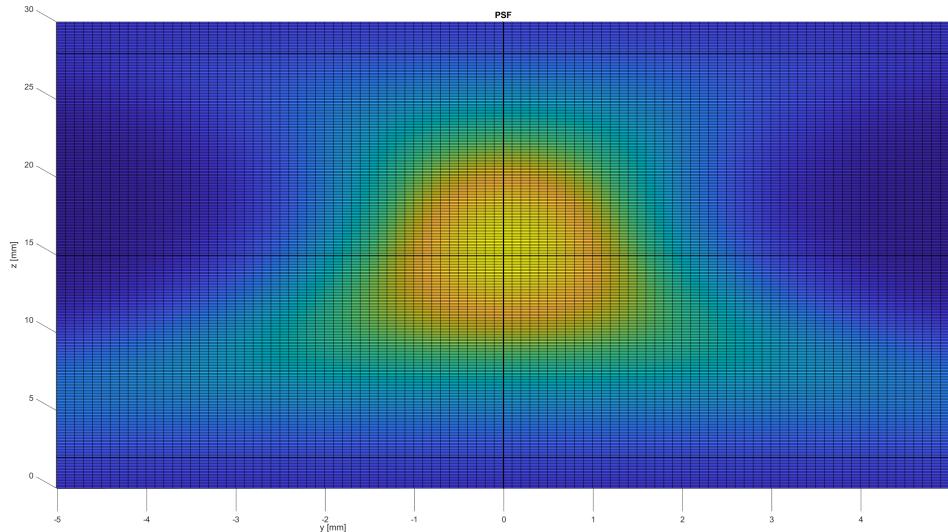


Abbildung 2.2: Rekonstruktion der Punktspreizfunktion über ein Parametrisches Modell. Das Parametrische Modell bezieht sich auf die in Kapitel 2.4.1 erklärte Form der PSF und ist mathematisch betrachtet ein Rotationsvolumen, dass durch anpassen der Parameter der PSF des System gleicht.

Wie man in Abbildung 2.2 erkennen kann, ist der Punkt, der infinitesimal klein sein soll, im Fokus genau zu erkennen. In der Abbildung ist dies der gelbe Bereich. Der orangene Bereich ist der Rand des Fokusbereiches. Der Übergang in das Nah- und Fernfeld wird mit grün dargestellt und hellblau zeigt das Nah- und Fernfeld. Der dunkelblaue Bereich kennzeichnet den Bereich außerhalb der Punktspreizfunktion.

Je weiter man sich vom gelben Punkt entfernt, desto weiter entfernt man sich von der Auflösungsgrenze. Dies liegt daran, dass das Schallfeld, im Nah- und Fernfeld, größer ist als die Schrittweite des Systems. Die Schrittweite eines Systems, ist der Abstand zwischen zwei Scanpositionen des Ultraschallwandlers. Idealerweise würde der Schall nur vom Fokusbereich reflektieren, allerdings ist dies in der Realität nicht möglich. Wenn der Ultraschallwandler einen Schritt weiter geht, wird der erste Punkt vom Schallfeld wieder erfasst und gibt ein

Signal zurück, obwohl nun ein anderer Punkt im Fokus liegt. Dadurch empfängt der Sensor weitere reflektierte Signale. Dieses Phänomen sorgt für die schlechte Auflösung, abseits des Fokusbereiches. Die zusätzlichen Informationen sind aber in der Zeit etwas verschoben. Dieses Wissen wird sich in der Synthetischen Apertur zunutze gemacht. Auf die synthetische Apertur wird ausführlicher im Kapitel 2.4 eingegangen.

### 2.2.3 Bildrekonstruktion

Ein wichtiger Bereich der akustischen Mikroskopie ist die Bildrekonstruktion. Dieser Bereich befasst sich damit, wie man aus den gewonnenen Daten ein Bild rekonstruieren kann. Das Ultraschallbild ist üblicherweise ein Graustufenbild, dass seine Qualität hauptsächlich durch seinen Kontrast und den Dynamikumfang der Grauwerte bestimmt. Der Pixelwert jedes Pixel ist der aufgenommene Signalwert. Dieser wird als Grauwert gesetzt. Die Signalwerte der Radio-Frequenz Daten liegen zwischen null und eins und können deswegen einfach als Pixelwert übernommen werden. Sollte der Datensatz mittels der Hilbert-Transformation transformiert werden, so werden Abweichungen über dem Wert eins, eins gesetzt. Der Datensatz des akustischen Mikroskops ist dreidimensional. Das Vorgehen, den Datensatz aufzubauen ist, das x- und y-Gitter ab zu rastern. Jedes aufgenommene Signal wird an der Position (x,y) in z-Richtung übersetzt. Verdeutlicht wird dies in Abbildung 2.3 Mit einem solchen Daten-

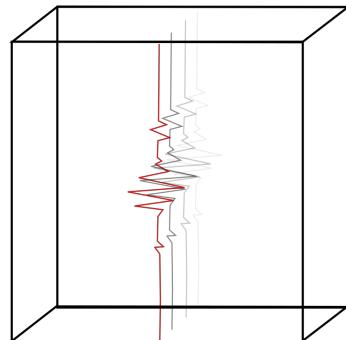


Abbildung 2.3: Darstellung der Signale im Datensatz. Übersetzung des aufgenommenen Signals in z-Richtung

satz, kann man verschiedene Arten der Visualisierung betreiben, darunter der C-, B- und A-Scan. Um optimale Ergebnisse zu erhalten, sollte der Datensatz vorher mit der Hilbert-Transformation transformiert werden [5, 9] um die sogenannte *Einhüllende* des Signals zu erhalten [1, 5].

### Die Hilbert Transformation

Die Hilbert Transformation wird in der Signalverarbeitung verwendet, um aus aperiodischen Signalen, analytische Signale zum machen [2]. Für diese Signale ist die Hilbert Transformation wie folgt definiert: [2]:

$$\mathcal{H}f(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(\tau)}{t - \tau} d\tau. \quad (2.5)$$

Voraussetzung für die fehlerfreie Anwendung der oben beschriebenen Hilbert Transformation ist, dass das Signalband begrenzt ist und eine endliche Energie aufweist [2]. Sollte das Signal eine unendliche Energie aufweisen gilt:

$$\mathcal{H}(t) = \frac{1}{2 * \pi} \int_{-\infty}^{\infty} f(\tau) \cot\left(\frac{t - \tau}{2}\right) d\tau \quad (2.6)$$

In dieser Arbeit wird die Hilbert Transformation aus 2.7 verwendet um die Einhüllende des Signals zu erhalten. Darunter versteht man, die Hüllfunktion über alle Amplituden der Magnitudenfunktion. Die Magnitudenfunktion ist das gleichgerichtete Ursprungssignal. Eine Hilberttransformierte ist in Abbildung 2.4 zu finden.

$$\mathcal{H}f(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{|f(\tau)|}{t - \tau} d\tau. \quad (2.7)$$

### A-Scan

Der A-Scan ist die einfachste Variante, Informationen des Datensatzes zu visualisieren. Hierbei wird einfach ein akustisches Signal in einem einfachen Graph dargestellt [7]. Veranschaulicht wird dies auch in Abbildung 2.4. Um einen A-Scan darzustellen, wird für eine gewollte Position  $(x, y)$  alle Werte in der z-Ebene, also der t-Achse, selektiert und extrahiert. Hat man die Samplerate des Systems, können die Werte auch der Zeit nach geplottet werden.

### B-Scan

Der B-Scan ist ein Schnittbild entlang der x- und z-Achse oder der y- und z-Achse, dargestellt in Abbildung 2.6 und Abbildung 2.5. In der Ultraschalldiagnostik ist es der am weit verbreitetste Bildmodus [7]. Das *B* steht für *brightness* und bedeutet, dass das Bild nur aus Graustufen besteht die die Helligkeit an einer Stelle repräsentieren. Um den B-Scan zu generieren, muss an der gewählten Achse (x- oder y-Achse) alle Werte, der anderen Achse festgehalten werden. Dies gilt für alle Ebenen auf der z-Achse. Das Ergebnis ist ein Schnittbild entlang der gewählten Achse.

### C-Scan

Der C-Scan ist in der akustischen Mikroskopie, das eigentliche Bild, welches als erstes abgebildet wird. Es entspricht dem Bild, welches man durch ein optisches Mikroskop sehen würde. Um den C-Scan zu generieren, müssen alle Z-Ebenen in ihren Positionen aufsummiert werden. Sprich, alle Werte von z an jeder Position  $(x, y)$  müssen summiert werden. Dadurch entsteht

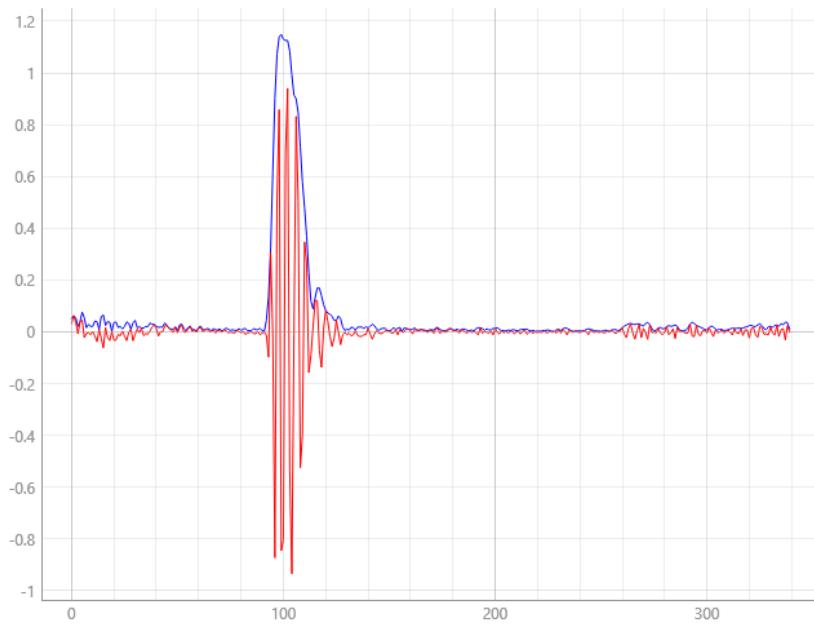


Abbildung 2.4: Plot eines akustischen Signals (rot) und der dazu gehörigen Hilbert-Transformation (blau). Der rote Graph entspricht auch der Darstellung als A-Scan für diese Position.

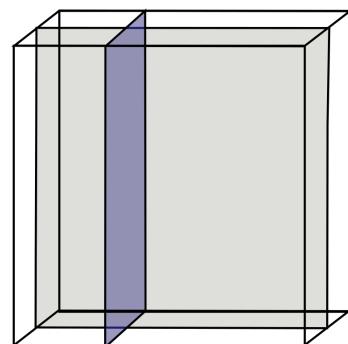
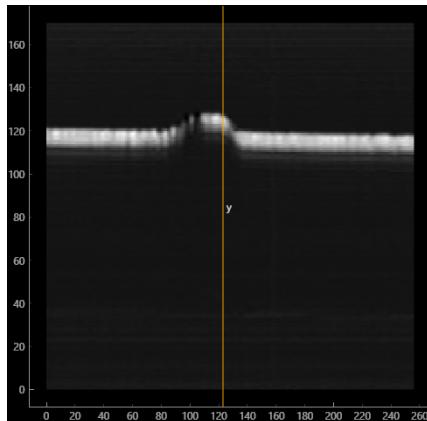
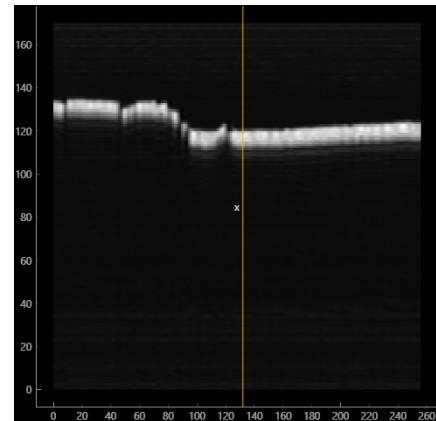


Abbildung 2.5: Darstellung der Schnittbilder im Datensatz. grau: entlang der x-Achse, blau: entlang der y-Achse



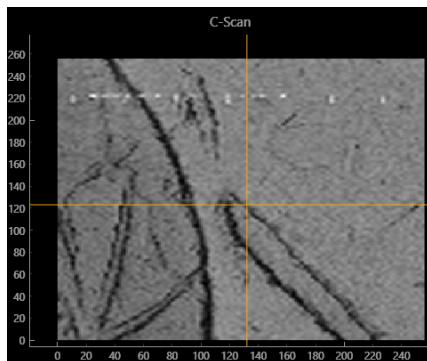
(a) B-Scan entlang der x-Achse



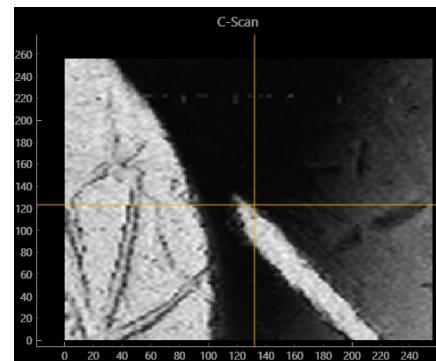
(b) B-Scan entlang der y-Achse

Abbildung 2.6: Beispiel von B-Scans einer 1 Cent-Münze, (a) B-Scan entlang der x-Achse, (b) entlang der y-Achse

das Bild. Es gibt auch die Möglichkeit, nicht alle Ebenen von z zu summieren. Dazu grenzt man den Bereich in der z-Achse ein und summiert nur diesen auf. Dies ermöglicht es, durch bestimmte Abschnitte in z zu laufen. Hier gilt das gleiche Prinzip zur Erstellung des Bildes.



(a) C-Scan über alle Ebenen in z



(b) C-Scan über die ersten 50 Ebene in z

Abbildung 2.7: Zwei C-Scans eines akustischen Datensatzes. Wie im B-Scan wurde eine 1-Cent Münze gescannt. Die Linien entsprechen den Linien in der Abbildung 2.6

## 2.2.4 Aufbau eines akustischen Systems in der Mikroskopie

Das akustische System, das in dieser Arbeit verwendet wird, ist von der Kibero GmbH. Es heißt easySAM und ist eine technische Erweiterung für Labormikroskope. Rein visuell, ist das System ein kleiner handlicher Würfel, der aus zwei Ebenen besteht. Die untere Ebene kann sich in den räumlichen Achsen ausrichten. An der untersten Ebene ist die Linse des Systems befestigt, siehe Abbildung 2.9.

Das easySAM System wird am Mikroskop oder in der in Abbildung 2.9 befestigt.

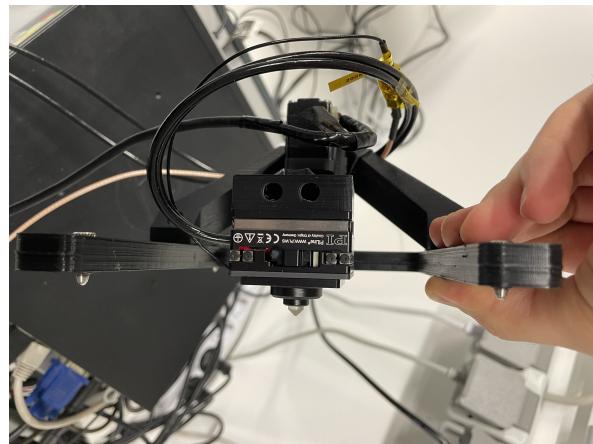


Abbildung 2.8: easySam Scanner. Man kann auf der Abbildung sehen, dass die Linse am Scanner befestigt ist. Die Linse ist am unteren Teil des Scanners befestigt.

### Die Linse des Systems

Die Linse von diesem akustischen Mikroskops ist ein geschliffener Saphir. Am oberen Ende des Saphirs, ist das Wandlerelement angebracht. Die Schallwellen wandern durch den Saphir zu der Linse am unteren Bereich des Saphirs. Hier werden die Schallwellen gebündelt und ausgesendet. Das Prinzip der akustischen Linse folgt dabei denselben Prinzipien wie optische Linsen.

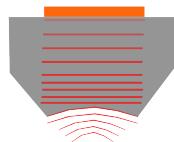


Abbildung 2.9: Skizze des Wandlers. Das Wandlerelement (orange) sitzt auf dem Saphir und sendet dort Schallwellen hinenin (rot). Diese werden an der Apertur fokussiert und ausgesendet.

## 2.3 Koordinatensysteme

Um die synthetische Apertur korrekt zu berechnen, muss das korrekte Koordinatensystem gewählt werden. Warum dies wichtig ist und was das bedeutet, wird im Folgendem erklärt.

Im Aufbau des akustischen Systems unterscheidet man drei verschiedene Koordinatensysteme. Das erste Koordinatensystem ist das elektronische Koordinatensystem. Hier betrachtet man

alle Punkte des akustischen Signals vom ersten elektronischen Impuls des piezoelektrischen Wandlers. Das zweite Koordinatensystem ist das sogenannte Linsen-Koordinatensystem. In diesem System ist der Nullpunkt an der Linse des Wandlers. Die genaue Position ist die Position des ersten Vorlaufechos der Linse. Das letzte Koordinatensystem ist das Delay-Koordinatensystem. Hier wird der Nullpunkt vom gesetzten Hardware-Delay bestimmt. Der Datensatz, liegt nach einladen immer im Delay-Koordinatensystem vor.

Der Grund, warum man diese Verschiedenen Koordinatensysteme betrachten muss ist, um Fehler bei Berechnungen von Abständen und Zeitunterschieden zu verhindern. Das Koordinatensystem, welches die richtigen Maße beinhaltet ist das Linsen-Koordinatensystem, da wir alle Punkte ab dem Zeitpunkt des Austritts der Schallwellen betrachten. Da der Datensatz im Delay-Koordinatensystem vorliegt, muss man zuerst eine Transformation von dem einen in das andere System vorgenommen werden. Alle Transformationen müssen in Sample gerechnet werden.

### **Das elektronische Koordinatensystem**

Das elektronische Koordinatensystem hat seinen Nullpunkt an der Position, an der der elektronische Impuls des Ultraschallwandlers auftritt. Um zu verdeutlichen, was damit gemeint ist folgt eine Abbildung: In diesem Koordinatensystem ist auch die Variable des Hardware-Delay

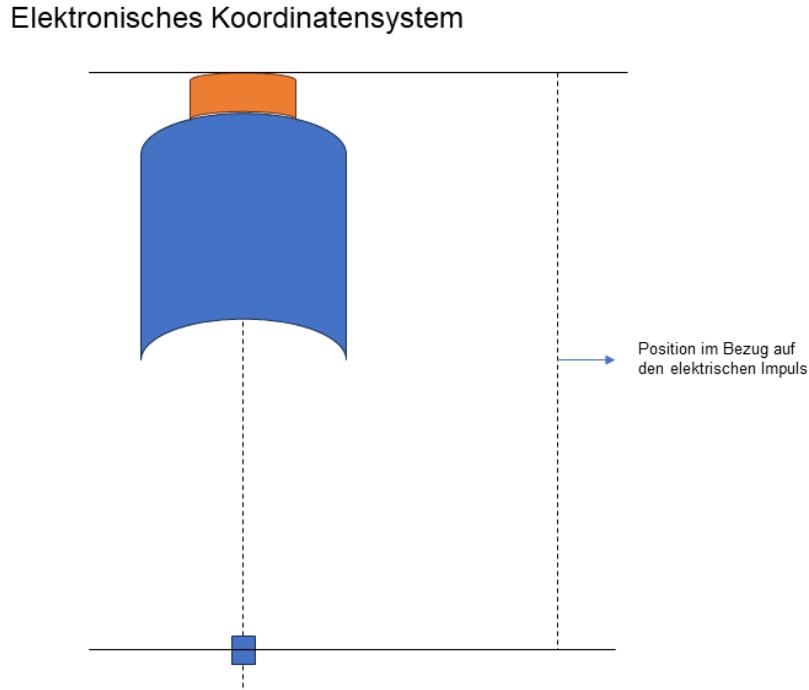


Abbildung 2.10: Skizze des elektronischen Koordinatensystems. Es wird gezeigt, dass der Bezugspunkt zum Rekonstruktionspunkt am Wandleranfang sitzt.

zu finden. Für diese Arbeit ist der Hardware Delay bei 5000 Samples. Dies bedeutet, dass das

Delay-Koordinatensystem erst nach 5000 Sample-Punkten beginnt.

### Das Linsen Koordinatensystem

Das Linsen-Koordinatensystem besitzt den Nullpunkt an dem Austrittspunkt des Schalls aus der Linse. In diesem Koordinatensystem arbeitet man mit den physikalisch korrekten Werten, wenn man Abstände oder Zeiten berechnen will. Dies ist von großer Bedeutung auf blick für die synthetische Apertur, da man genau diese Berechnungen durchführen muss, siehe Kapitel 2.4.1. Um Positionen vom elektronischen Koordinatensystem in das Linsen-Koordinatensystem um zu rechnen, muss man folgende Transformation durchführen:

$$\mathcal{L}K \Rightarrow P_{EK} - (HD - Echo_{Linse}) \quad (2.8)$$

Das  $P_{EK}$  repräsentiert einen Punkt aus dem elektronischen Koordinatensystem,  $HD$  steht für *Hardware-Delay* und  $Echo_{Linse}$  steht für den Samplewert, an dem das Linsenecho auftritt.

### Linsen Koordinatensystem

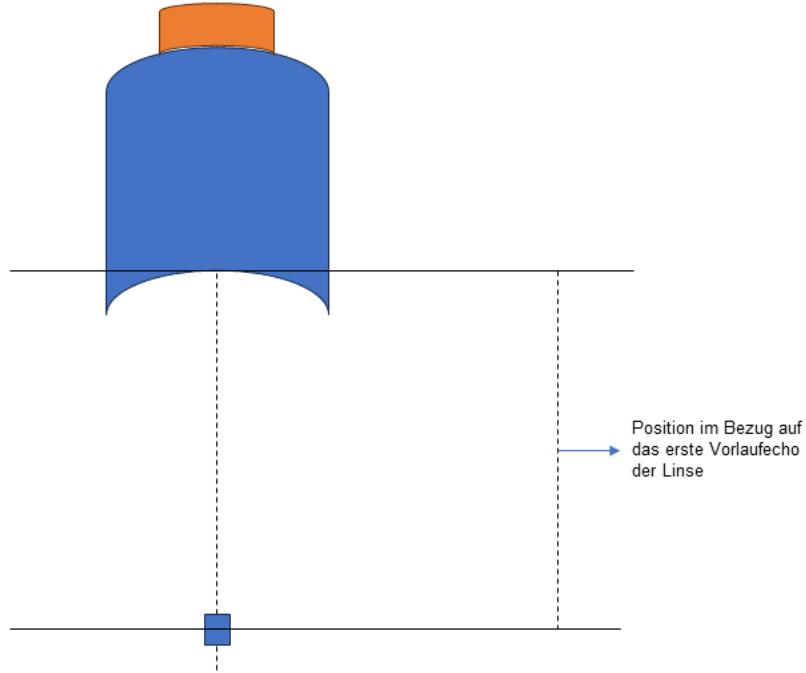


Abbildung 2.11: Das Linsen Koordinatensystem. Die Abbildung zeigt den verschobenen Nullpunkt auf den Austrittspunkt des Schalls.

### Das Delay-Koordinatensystem

Das Delay-Koordinatensystem hat seinen Nullpunkt am Hardware Delay. Der Datensatz wird im Delay Koordinatensystem gespeichert und auch wieder geladen. Das bedeutet, dass man vom Delay in das Linsen-Koordinatensystem transformieren muss um Berechnungen mit den

Daten durchzuführen. Will man dann aber nun die Ergebnisse auf den Datensatz anwenden, muss man wiederum vom Linsen-Koordinatensystem zurück transformieren.

$$\mathcal{DK} \Rightarrow \mathcal{LK} := P_{DK} + (HD - Echo_{Linse}) \quad (2.9)$$

Die Rücktransformation sieht wie folgt aus :

$$\mathcal{LK} \Rightarrow \mathcal{DK} := P_{LK} - (HD - Echo_{Linse}) \quad (2.10)$$

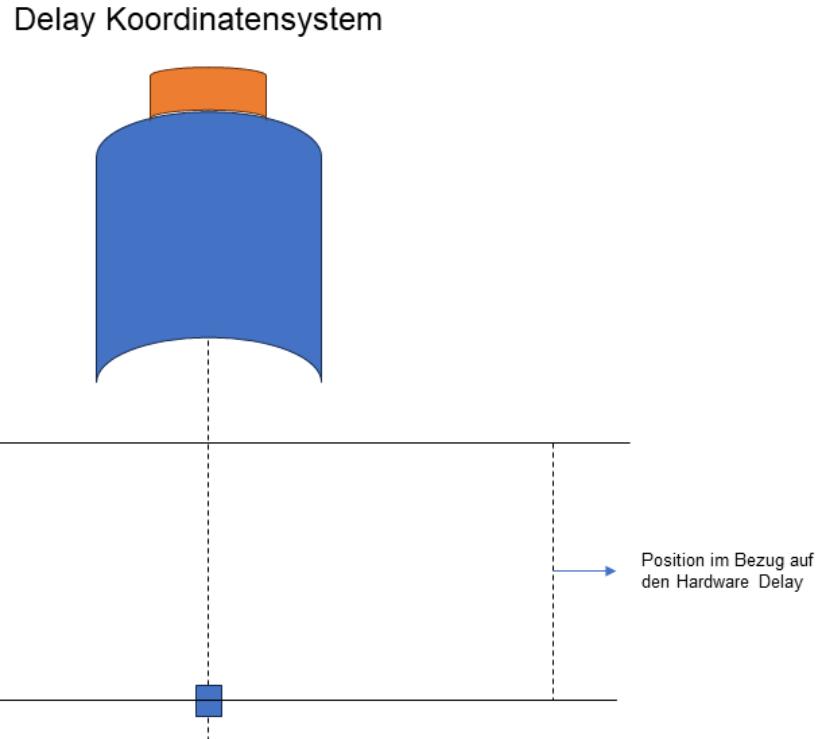


Abbildung 2.12: Skizze des Delay-Koordinatensystems. Die Skizze zeigt, dass der Nullpunkt des Delay-Koordinatensystems auf dem Hardware-Delay liegt.

## 2.4 Prinzip der synthetischen Apertur

Die Synthetische Apertur ist ein Verfahren, welches ursprünglich aus der Radartechnik stammt und auf das Gebiet der Ultraschalltechnik übertragen wurde [1, 8, 9]. Dies ist möglich, weil die Gegebenheiten beider Techniken sehr ähnlich sind. Beide Techniken senden physikalische Wellen aus und empfangen die reflektierten Signale um eine Region von Interesse zu scannen. Diese Systeme besitzen einen Fokusbereich, in dem die Auflösung des Bildes sehr gut ist. Man fokussiert die ausgesendeten Wellen auf einen Bereich, optimalerweise auf einen Bereich von Interesse. Das entstehende Problem ist, dass man, vor allem im Ultraschall, auch Informationen außerhalb dieses Fokusbereichs betrachten möchte. Wie bereits beschrieben, ist das möglich, da die Schallwellen nicht nur im Fokusbereich reflektiert werden, sondern auch in den

umliegenden Regionen (Kapitel 2.2.2). Da in den umliegenden Bereichen aber nun keine gute Auflösung vorhanden ist, muss mit der synthetischen Apertur, außerhalb des Fokusbereichs nachgeholfen werden.

Das Prinzip hinter der synthetischen Apertur ist es, eine größere Apertur des Systems mathematisch zu synthetisieren [1]. Es ist möglich, die synthetische Apertur im Zeit- sowie auch im Frequenzbereich zu implementieren. Für diese Arbeit wird sich auf die zeitliche Variante beschränkt. Das wichtige bei der synthetischen Apertur ist, zu wissen welche Daten man in die Berechnung miteinbezieht, und welche nicht. Zu viele Werte können die Arbeitslast zu groß werden lassen. Um nun zu wissen, welche Werte man einbeziehen soll und welche nicht, muss einem die genaue Punktspreizfunktion des verwendeten Systems bekannt sein. Zudem sollten sämtliche Linsenparameter bekannt sein. Die genaue Breite der Apertur der Linse gibt uns die maximale Breite unseres Schallfeldes an. Diese Breite ist eine Größe, die als Richtwert verwendet werden kann.

#### 2.4.1 Methode der synthetischen Apertur

Das Verfahren hinter der Synthetischen Apertur im Bereich der akustischen Mikroskopie, befasst sich damit, für jedes aufgenommene Signal, Signalanteile der umliegenden Aufnahmen, auf das Signal auf zu summieren. Hierbei müssen verschiedene Aspekte berücksichtigt werden, die eine wichtige Rolle spielen. Zum einen, muss eine bestimmte zeitliche Verzögerung beachtet werden. Diese zeitliche Verzögerung erhält man aus Abstandsberechnungen von verschiedenen Standpunkten des Wandlers. Zum anderen muss man die resultierenden Signalanteile, die summiert werden, entsprechend gewichten. Die folgende Abbildung zeigt den neuen geometrischen Abstand  $r$ , der durch die neue Wandlerposition entsteht.

Auf der Abbildung 2.13 ist zu sehen, dass wenn der Wandler einen Schritt weiter geht der Punkt einen neuen Abstand zum Wandler hat. Diesen Abstand benötigt man um später die zeitliche Verzögerung zu berechnen. Außerdem kann man mit dieser Abbildung verdeutlichen, dass die Schallfelder im Nah- und Fernfeld große Überschneidungen aufweisen. Diese Überschneidungen, sind in der Abbildung 2.13 grün-farbig dargestellt. Das Prinzip der Synthetischen Apertur, lässt sich demnach auf alle vorhandenen Punkte in diesen Bereichen anwenden. Außerdem ist zu erkennen, dass im Fokusbereich keine Überschneidungen vorliegen. Hier macht die Synthetische Apertur keinen Sinn.

Der Abstand  $r_2$  in der Abbildung 2.13 orange dargestellt, lässt sich wie folgt berechnen:

$$r = \sqrt{p_1^2 + p_2^2} \quad (2.11)$$

Bezieht sich auf  $P_1$  und  $P_2$  in der Abbildung 2.13. Aus diesem geometrischen Abstand und der Schallgeschwindigkeit im Medium, lässt sich nun ein  $\delta t$  berechnen.

$$\delta t = \frac{r}{c} = \frac{\sqrt{p_1^2 + p_2^2}}{c} \quad (2.12)$$

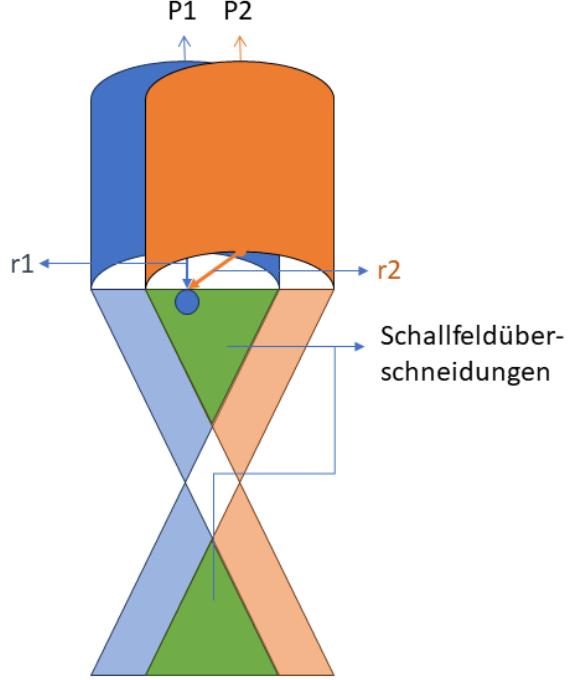


Abbildung 2.13: Abstandsveränderung  $r$  wenn der Ultraschallwandler von Position P1 auf P2 wandert. Der ursprüngliche Abstand  $r_1$  (blau) ist kürzer als der neue Abstand  $r_2$  (orange). Die Überschneidungen von Nah- und Fernfeld zwischen den beiden Positionen sind in grün zu sehen.

Mit diesem  $\delta t$  kann man nun, aus dem Nebensignal zum Zeitpunkt

$$t_{psf} = t_0 + \delta t \quad (2.13)$$

den Signalanteil extrahieren und auf unser Hauptsignal zum Zeitpunkt  $t_0$  hinzu addieren. Damit kommt man auf folgenden mathematischen Ausdruck für die synthetische Apertur:

$$saft = S_H(t_0) + \sum_N S_N(t_{psf,N}), \quad (2.14)$$

wobei  $S_H$  das Hauptsignal und  $S_N$  das Nebensignal abbildet. Dabei kann man es aber noch nicht belassen. Beide Anteile müssen noch gewichtet werden. Dies muss man machen, da man in Bereichen, in der man synthetische Apertur betreiben will davon ausgeht, dass unser Hauptsignal an diesem Punkt auch nicht zu hundert Prozent dargestellt wird. Auch der Signalanteil der Nebensignale, die man betrachtet, entfernen sich immer mehr vom Hauptsignal, womit auch die Wichtung abnimmt. Dies kann man gut an der Punktspreizfunktion ausmachen. Das bedeutet, dass beide Signalanteile, die man verrechnen möchte mit einer eigene Wichtung  $w$  multipliziert werden müssen.

$$saft = w_h * S_H(t_0) + \sum w_n * S_N(t_0 + \delta t), \quad (2.15)$$

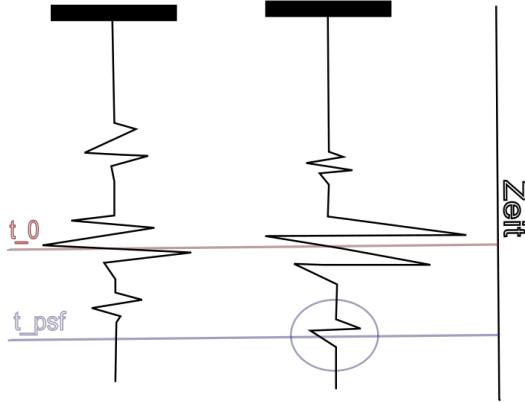


Abbildung 2.14: Der Signalanteil, der aufsummiert werden muss

wobei  $w_h$  die Gewichtung für den Wert des Hauptsignals ist und  $w_n$  der Gewichtungsfaktor für das Nebensignal. Beide Wichtungsfaktoren hängen von der aktuellen Position des Wandlers in Relation zum Punkt in Betracht ab. Die Wichtungen können mithilfe der Punktspreizfunktion berechnet werden.

### Modell der Point-Spread-Function

Eine genaue Erklärung, was die Punktspreizfunktion ist, ist in 2.2.2 zu finden. In dieser Arbeit wird die Punktspreizfunktion mittels mehrerer exponential Funktionen ( $e$ )modelliert. Durch Multiplikation aller  $e$ -Funktionen wird ein Rotationsvolumen erzeugt, welches Auskunft über die Gewichtung eines Punktes innerhalb der Funktion gibt. Grundlage des Modells ist:

$$e^{\frac{-r^2}{\sigma}}. \quad (2.16)$$

Hierbei ist das  $x$  immer der Abstand zwischen dem Punkt im Datensatz und der Wandlerposition. Das  $\sigma$  ist die Halbwertsbreite der jeweiligen Gaußfunktion an der Position. Die Halbwertsbreite ist die Position, an der die Amplitude der Gaußfunktion sich halbiert hat. Sollte ein Punkt außerhalb des Modells liegen, so ist die Gewichtung  $w = 0$  und diese Position ist vernachlässigbar. Ist ein Punkt in relevanter Nähe so gilt folgende Funktion:

$$psf = e^{-\alpha * z} * e^{-\frac{((z+\delta t)-fokusstiefe)^2}{2*\mu_z^2}} * e^{-\frac{(y-y_{wandler})^2}{2*\mu_y^2}} * e^{-\frac{(x-x_{wandler})^2}{2*\mu_x^2}} \quad (2.17)$$

Die erste der  $e$ -Funktionen ist die Dämpfung die das System durch die Tiefe erfährt 2.3. Die darauf folgenden Funktionen Bilden die Fokusstärke der Schallwellen in  $x, y, z$  ab. Setzt man nun einen beliebigen Punkt in die Funktion ein gibt sie einen Wert zurück, der als Wichtung genutzt werden kann.

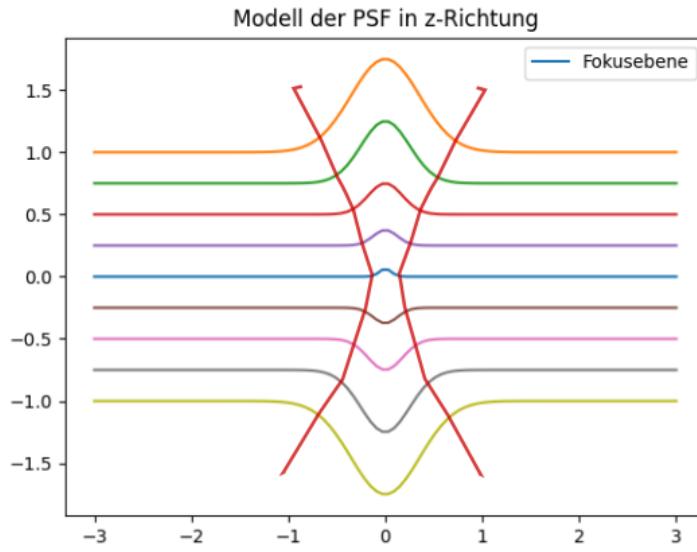


Abbildung 2.15: Skizze der e-Funktionen über die z-Achse. Die zwei roten Linien sollen die Form des Schallfeld über die z-Achse skizzieren.

Damit die Summe in Gleichung [?] nicht über den kompletten Datensatz läuft, grenzt man die Werte, die man betrachten will, ein. Die Summe soll mit Werten durchgeführt werden, die kugelförmig um den Punkt des Hauptsignals liegen. Der Radius dieser Kugel hängt von den Abmessungen der Apertur und der Schrittweite des Systems ab. Das verwendete System hat ein Aperturdurchmesser von  $750.87 \mu\text{m}$  und eine Schrittweite von  $5 \mu\text{m}$ . Damit lässt sich nun ein Bereich definieren:

$$\text{Bereich} = [\text{Anfang} : \text{Ende}] \quad (2.18a)$$

$$\text{Anfang} = -\frac{\frac{\text{Durchmesser}}{\text{Schrittweite}}}{2} \quad (2.18b)$$

$$\text{Ende} = \frac{\frac{\text{Durchmesser}}{\text{Schrittweite}}}{2} \quad (2.18c)$$

Mit den Werten, die durch den Bereich ausgewählt werden, kann nun die synthetische Apertur berechnet werden. Der Anfang des Bereichs wird negativ dargestellt, damit im Algorithmus von der aktuellen Position aus in den Daten zurück gegangen wird.

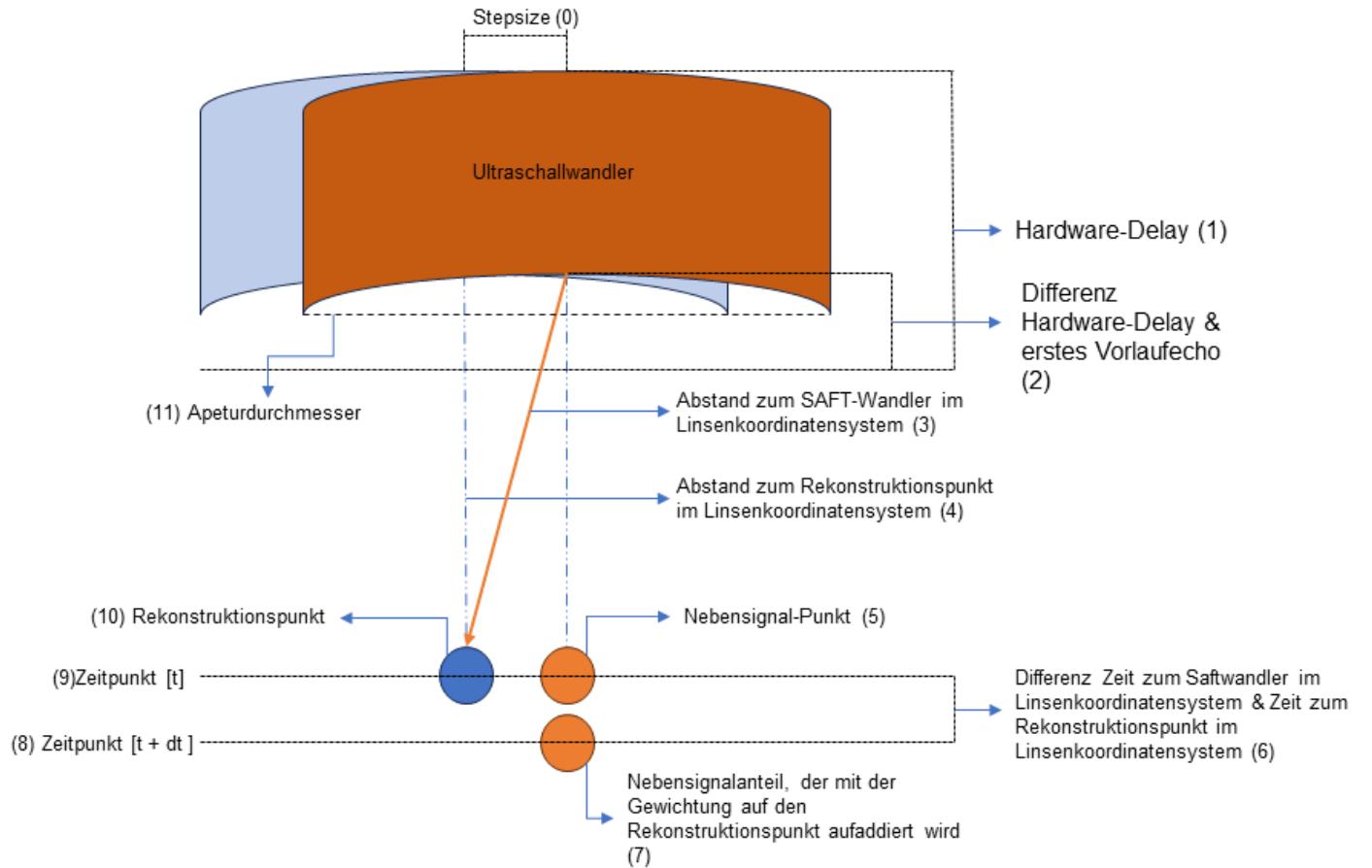


Abbildung 2.16: Skizze des Prinzips der synthetischen Apertur. Die Abbildung zeigt die wichtigen Abstände, die für die Berechnung der zeitlichen Unterschiede wichtig sind. (0) ist die Step size, die der Wandler zurück legt, (1) ist der eingestellte Hardware-Delay, (2) ist die Differenz zwischen dem Hardware-Delay und der Sampleposition des ersten Vorlaufechos der Linse, (3) Abstand von der nächsten Wandlerposition zum vorherigen Rekonstruktionspunkt, (4) Abstand vom der Ausgangswandlerposition zum Rekonstruktionspunkt, (5) Nebesignalpunkt zum Zeitpunkt [t], (6) Zeitliche Differenz zwischen dem Rekonstruktionspunkt der Ausgangswandlerposition und der Nebesignalwandlerposition, (7) Signalanteil aus dem Nebesignal zum Zeitpunkt [t+dt] der mit einer Gewichtung auf den Rekonstruktionspunkt aufaddiert werden soll, (8) Zeitpunkt [t + dt], (9) Zeitpunkt [t], (10) Rekonstruktionspunkt der Position von der Ausgangsposition des Wandlers, (11) Aperturdurchmesser der Linse

## Kapitel 3

# Algorithmus der Synthetic Aperture Focusing Technique

In diesem Kapitel wird auf den Algorithmus der synthetischen Apertur eingegangen. Es wird die informationstechnische Komponente gezeigt und wie der Algorithmus aufgebaut ist. Zudem werden zwei Ansätze der Implementierung diskutiert. Anschließend werden bereits verwendete Ansätze der synthetischen Apertur präsentiert, in ihre Funktionsweise und eventuelle Aspekte die sich mit dem Ansatz in dieser Arbeit decken.

### 3.1 Algorithmische Durchführung

Der Algorithmus wird in Python-Code entworfen und in eine eigene Umgebung eingebunden, die zur Visualisierung der Daten dient. Um den Algorithmus in Python zu realisieren, benötigt man verschiedene Bibliotheken, die importiert werden müssen:

- Numpy
- concurrent

Die Numpy-Bibliothek wird dazu verwendet mathematische Operationen auf Matrizen anzuwenden. Numpy steht für *numerical Python* und ist für Matrizenrechnungen entworfen worden. Zudem stellt Numpy Funktionen der höheren Mathematik zur Verfügung. Da die Daten in Matrizen vorliegen ist diese Bibliothek nahezu unumgänglich.

Die concurrent-Bibliothek, erlaubt es in Python Threads parallel laufen zu lassen. Da die synthetische Apertur eine sehr rechenaufwendige Methode ist, kann es durchaus zu langen Laufzeiten kommen. Mithilfe von `concurrent.futures` ist es möglich, Python-Code auf allen Prozessoren des verwendeten Computers parallel laufen zu lassen. Mit dieser Technik verkürzt man die Laufzeit deutlich.

### 3.1.1 Der Datensatz

Die Daten werden von dem Kibero-System easySAM aufgenommen. Hierbei verwendet man das Programm der Kibero GmbH und speichert den Datensatz ab. Aktuelle Datensätze haben ein Format von 1800 Samples mit einer Auflösung von 400x400 Pixeln. Diese Parameter können variieren, da sie von den Einstellungen des Systems abhängen. Durch die große Menge an Werten kommen die Datensätze auf eine Größe von ein bis zwei Gigabyte.

### 3.1.2 Der Algorithmus

Es gibt zwei Arten von Implementierung die für diese Arbeit in Betracht gezogen werden können. Die erste Variante ist, einen Algorithmus zu schreiben, der für jede Stelle in dem Datensatz das Prinzip der synthetischen Apertur ausführt. Zu beachten ist allerdings, das in dieser Variante eine enorme Rechenbelastung entsteht. Der Datensatz hat wie bereits erwähnt ein Format von  $1800 \times 400 \times 400$  was einer Anzahl von  $288 * 10^6$  Werten entspricht. Dies bedeutet, man muss den Algorithmus  $288 * 10^6$ -fach ausführen. Solch eine massive Aufgabe kann nur mit der GPU, der *Graphical Processing Unit* des Computers, durchgeführt werden um eine realistische Laufzeit zu erreichen. Selbst bei der Berechnung auf der GPU des Computers ist die Rechendauer noch zu lange für eine praktische Erweiterung in die Visualisierungsumgebung.

Die zweite Variante, um den Algorithmus zu implementieren ist die für jeden B-Scan separate Durchführung des Algorithmus. Diese Variante würde zu einer erheblichen Reduktion an Berechnungen führen. Anstatt  $1800 \times 400 \times 400$  -Werte würde man den Algorithmus nur noch für  $1 \times 400 \times 400$  -Werte ausführen. Strukturiert man die Daten sinnvoll, so kann man mit der *concurren.futures*-Bibliothek mehrere Prozesse starten, die den Algorithmus parallel laufen lassen. Im Folgenden wird die zweite Variante vorgestellt.

So wie der Algorithmus implementiert wird, besteht er aus zwei Funktionen. Die Hauptfunktion wird im Code *saft-für-eine-reihe-berechnen* benannt. Die Funktion *saft-Ausführen* ist dafür zuständig die Daten zu organisieren und zu sortieren, damit am Ende ein vernünftiges Bild entsteht. Als erstes wird auf die *saft-Ausführen*-Funktion eingegangen.

Die *saftAusführen*-Funktion ist hauptsächlich dafür verantwortlich, die Daten vorzubereiten, diese dann an den Algorithmus zu übergeben und das Ergebnis des Algorithmus zu sortieren und in ein Bild umzuwandeln. Dazu werden als erstes konstante Variablen definiert, die im Laufe der Funktion als Argumente übergeben werden. Zu den Argumenten zählen  $zd$ ,  $yd$ ,  $xd$  als Speicher des Formats der Achsen des Datensatzes, der Aperturdurchmesser, die Fokusbreite, die Schallgeschwindigkeit und den Bereich der synthetischen Apertur. Als letztes noch die Halbwertsbreiten  $\mu z$  und  $\mu xy$ .

Zuerst werden im ersten Algorithmus die Daten so selektiert, das möglichst alle Wandlerpositionen, die relevant sind, in den selektierten Daten vorhanden sind. Dabei muss auf die Grenzen des Datensatzes geachtet werden.

Nachdem ein Numpy-Array erstellt wird, das als Füllarray für die Ergebnisse da ist, werden für jede Reihe im B-Scan Argumenttupel erstellt, die alle relevanten Argumente enthalten. Die relevanten Argumente braucht der Algorithmus um die synthetischen Werte zu berechnen. Die nun entstandenen Tupel werden in Batches, also Gruppen aufgeteilt. Die Batches werden zur Bearbeitung den Prozessorkernen zugeordnet, in diesem Fall gibt es 16 Prozessorkerne und deshalb bearbeitet jeder Prozessorkern  $\frac{z-Dimension}{16}$  Reihen. Jeder Prozessorkern ruft den Algorithmus für die synthetische Apertur auf.

### **Algorithm 1** Vorbereitung, Übergabe und Auswertung der Daten

```

1: procedure SAFTAUSFÜHREN(data, achse, position, samplerate, hardwareDelay, vorlaufechoLinse)
2:   zd, yd, xd  $\leftarrow$  Numpy.Form(data)
3:   Aperturdurchmesser  $\leftarrow 750\mu\text{m}$ 
4:   Fokusbreite  $\leftarrow 5 \times 10^{-6}\text{m}$ 
5:   Schallgeschwindigkeit  $\leftarrow 1524\frac{\text{m}}{\text{s}}$ 
6:   Bereich_der_Apertur  $\leftarrow$  Rechnung(Formel: 2.18)
7:    $\mu_z = 10$ 
8:    $\mu_{xy} = 2$ 
9:   selektierte_daten  $\leftarrow$  Daten im Bereich der Apertur
10:  fuellarray  $\leftarrow$  Form(zd, xd)
11:  Liste_mit_Argumenttupeln  $\leftarrow$  400-Tupel
12:  batch_size  $\leftarrow$  zd/anzahl_CPU_kerne
13:  Liste_mit_argument_batches  $\leftarrow$  16 Listen mit 25 Sublisten
14:  Jeder Prozessorkern bekommt nun eine dieser 16 Listen und bearbeitet diese mit dem
15:  Algorithmus 2
16:  ergebnis_liste  $\leftarrow$  prozessor_ergebnis
17:  Liste mit Ergebnissen wird sortiert
18:  Rückgabe  $\leftarrow$  Bild_Array
19: end procedure
```

---

Der richtige Python Code ist im Anhang zu finden(Abbildung ??)

Das erste Argument eines Tupels ist die Tupel-ID. Der Tupel-ID folgen im Tupel die gewünschte Position des B-Scans, ein Zeitwert der zur aktuellen Ebene der Position passt, eine Liste mit den Pixeln auf der x-Achse, die zugeschnittenen Daten in einem bestimmten Bereich, die Schallgeschwindigkeit im vorliegendem Medium und die beiden  $\mu$ - Werte, die für die PSF-Funktion relevant sind 2.4.1.

Hauptteil des Algorithmus ist die For-Schleife über die Pixel-Liste. Für jeden x Index wird die komplette Berechnung durchgeführt. Das Ergebnis wird in eine Liste an die gleiche Position geschrieben.

Die Berechnung die für jeden x-Wert durchgeführt wird, läuft über zwei zusätzliche For-Schleifen. Bevor diese starten, wird noch der Bereich um den Rekonstruktionspunkt auf der x-Achse begrenzt. Dies ist wichtig, weil es ab einem bestimmten Bereich um den x-Wert keinen Sinn mehr macht, dass Vorgehen durchzuführen. Der erste Schritt rechnet den Samplewert der x-Reihe, also den  $zd$  Wert, in einen Zeitwert um. Dieser Zeitwert liegt noch im Delay-Koordinatensystem. Mehr zu den Koordinatensystemen ist im Kapitel 2.3 zu finden. Der berechnete Zeitwert muss nun in das Linsen-Koordinatensystem umgerechnet werden. Hierzu führt man die Transformation vom Delay-Koordinatensystem in das Linsen-Koordinatensystem durch, diese Transformation ist im Kapitel 2.9 erklärt. Hat man den Zeitwert in das andere System überführt, so berechnet man nun den Abstand zwischen den x-Koordinaten und y-Koordinaten der unterschiedlichen Wandlerpositionen. Diese Abstände müssen in die physikalische Größe Meter umgerechnet werden. Danach rechnet man den zuvor berechneten Zeitwert in einen Abstand im Linsen-Koordinatensystem um. Hat man diesen Abstand berechnet, folgt der Abstand vom Rekonstruktionspunkt zur nächsten Wandlerposition. Auch dieser Wert liegt im Linsen-System vor. Der neu berechnete Abstand wird nun in eine Zeit umgerechnet. Mit der Zeit vom Rekonstruktionspunkt zum Wandler an der Ursprungsposition und der Zeit vom Rekonstruktionspunkt zur nächsten Wandler Position kann man nun den Laufzeitunterschied des Schalls berechnen. Ist dieser berechnet, wandelt man ihn in einen Samplewert um und überführt ihn in das Delay-Koordinatensystem. Die Rücktransformation ist im Kapitel 2.3, Ausdruck 2.10 zu finden.

Als nächstes muss der Punkt, in Bezug auf den Laufindex x und die Verschiebungen in den Achsen gewichtet werden. Dazu wird die PSF( Kapitel 2.4.1) verwendet. Der nächste Schritt multipliziert die Gewichtung aus der PSF mit dem eigentlichen Signalwert des Punktes an der Position  $Daten[z + \Delta z, yPosition + \Delta y, x + \Delta x]$  und speichert das Ergebnis in der Liste die alle SAFT Werte zwischenspeichert. Sind die inneren zwei For-Schleifen durchgelaufen, so wird die Liste mit allen berechneten SAFT-Werten aufsummiert und durch die Anzahl der Werte dividiert, um den Wert zu normieren. Der normierte Wert wird an die Position des x-Index in ein Array gespeichert. Dieses Array wird als Rückgabewert zurückgegeben.

Der richtige Python Code ist im Anhang zu finden(Abbildung ??)

---

**Algorithm 2** Berechnung der PSF-Werte für eine Pixelreihe im B-Scan

---

```
1: procedure saft_für_eine_reihe_berechnen(Argumente)
2:   zd, yd, xd = Numpy.Form(selektierten_daten)
3:   pixel_reihe → Integer Array
4:   Bildreihe als Numpy – Array
5:   erste Stelle der Bildreihe ← Tupel ID
6:   samplerate ← 2 Ghz
7:   Dämpfung ← 0.05
8:   Stepsize ←  $0.005E - 6$ 
9:   Fokustiefe ←  $72E - 6$ 
10:  lauf_idx ← (selktierte_daten_y_länge * selktierte_daten_x_länge)
11:  Differenz Hardware Delay und Echo der Linse berechnen
12:  for x in der Range(Bildreihe) do
13:    Initialisierung Liste aller SAFT Werte
14:    Der Bereich um x wird genau wie der Bereich um y begrenzt
15:    for var1 in der Range(selktierte_daten_y_länge) do
16:      for var2 in der Range(selktierte_daten_x_länge) do
17:        → dt berechnen
18:        →  $\Delta x$  berechnen
19:        →  $\Delta y$  berechnen
20:        → Zeitwert zum Rekonstruktionsort im LK berechnen
21:        → Abstand zum Rekonstruktionsort im LK berechnen
22:        → Abstand zum SAFT-Wandler berechnen im LK
23:        → Zeit zum SAFT-Wandler berechnen im LK
24:        → Differenz der Zeiten berechnen
25:        → zeitliche Differenz in einen Abstand umrechnen
26:        → zeitliche Differenz in Samples umrechnen
27:        → Sample Differenz in das DK transformieren
28:        → Position (var1, var2) gewichteten mit PSF
29:        → Punkt in den selektierten Daten im Bezug auf die laufvariable x
30:        → Punkt gewichteten und in Liste aller SAFT Werte speichern
31:    end for
32:  end for
33:  Bildreihe an Stelle x → Summe aller Werte der Liste mit allen SAFT Werten
34:  normiert mit der anzahl aller Werte
35:  Rückgabe Bildreihe
36:
```

---

## 3.2 Methoden der synthetischen Apertur in der akustischen Mikroskopie

Um einen Einblick zu bekommen, wie die Methode bereits in der Wissenschaft verwendet wird und welche Ergebnisse diese Methoden erzeugen, soll nun auf vergleichbare Methoden eingegangen werden. Vorab wird festgelegt, dass in allen Beispielen davon ausgegangen wird, dass die physikalischen Gegebenheiten gleich sind.

### ATD-SAFT nach Ameri Golafsoun

Ein älteres Beispiel, welches schon länger bekannt ist, ist die TD-SAFT, Time-Domain SAFT von Ameri Golafsoun, welche in seiner Masterarbeit 2011 veröffentlicht wurde. In dieser Arbeit steht das TD-SAFT für Time-Domain Synthetic Aperture Das Vorgehen hier, ist genau wie in dieser Arbeit, eine Summe zu erstellen, die über einen Bereich an Signalen, Signalanteile aufsummiert und in einer bestimmten Art und Weise zu gewichten.

$$f_{ATD}(x, y) = \sum_{n=n_i}^{n_f} W_n(n - n_i) S\left(\frac{2r_n}{c}, Un\right)$$

[1] Die Summe summiert zu einer bestimmten Zeit  $\frac{2r_n}{c}$ , wobei  $r_n$  der Abstand zum Punkt  $(x, y)$  ist.

$$r_n = \sqrt{x^2 + (y - Un)^2}$$

Dazu kommt die Apodization mittels des Hemming-Fensters, indem sie zwei Komponenten miteinander multipliziert. Auch hier wird der Bereich der Apertur eingegrenzt, mit den Positionen des Wandlers. Die zwei Grenzen werden wie folgt berechnet:

$$U_{n_i} = y - \frac{L}{2}$$

und

$$U_{n_f} = y + \frac{L}{2}$$

wobei  $L$  mit

$$L = 2 * x * \tan(\Phi)$$

und  $\Phi$  mit

$$\Phi = \tan^{-1}\left(\frac{D}{2x_f}\right)$$

berechnet wird.  $D$  steht hierbei für den Aperturdurchmesser. Aus diesen Berechnungen folgen schlussendlich die Summengrenzen

$$n_i = \frac{U_{n_i}}{d_u}$$

$$n_f = \frac{U_{n_f}}{d_u}$$

wobei  $d_u$  die Schrittweite des Systems ist.

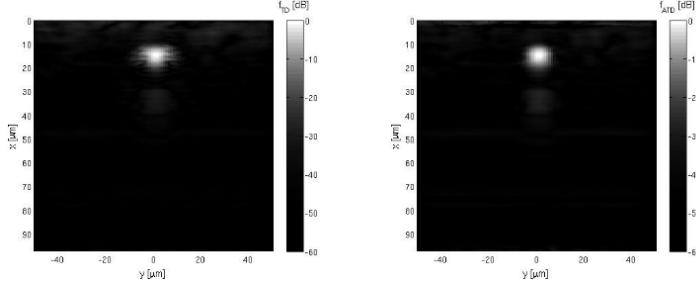


Abbildung 3.1: Ergebnis ADT-SAFT, kopiert aus: [1]

### 1D-SAFT-CF nach Gao Rongkang et al.

Ein weiter Ansatz, wurde 2022 in einem Journal [3] veröffentlicht. In diesem Bericht, werden ebenfalls mehrere Versionen einer synthetischen Apertur Fokussierung Technik angesprochen. Dies bezieht sich aber nicht auf den Frequenzbereich sondern auf die Dimension im Raum. Der Ansatz ist, dass ein  $\delta t$  berechnet wird, das zusätzlich mit einer Signum-funktion multipliziert wird. Diese verändert das Vorzeichen des  $\delta t$  und ist abhängig von den z-Koordinaten der Signale.

$$\delta t = \text{sign}(z - z_j) * \frac{r_{i+j-r_i}}{c}$$

Die Indizes kennzeichnen zu welchen Signalen die Variablen gehören.  $(i, t)$  gehören zu dem Signal und  $j$  zum Nebensignal. Die Summenformel dieses Ansatzes setzt sich aus der Summe der Signalanteile zusammen.

$$RF_{saft}(i, t) = \sum_{j=-N}^N RF(i + j, t - \delta_t)$$

Die Signalpunkte werden nun, zur Artefaktreduzierung, mit der *CF-Wichtung* verrechnet

$$RF_{SAFT-CF}(i, t) = CF(i, t) \times RF_{SAFT}(i, t)$$

wobei *CF* mit

$$CF(i, t) = \frac{|\sum_{j=-N}^N RF(i + j, t - \delta_t)|^2}{N \sum_{j=-N}^N |RF(i + j, t - \delta_t)|^2}$$

berechnet wird. Dieser Ansatz wird in dem Bericht als 1D-SAFT bezeichnet und gilt für den üblichen B-Scan in der photoakustischen Mikroskopie. In der folgenden Abbildung kann man die Ergebnisse des Algorithmus sehen:

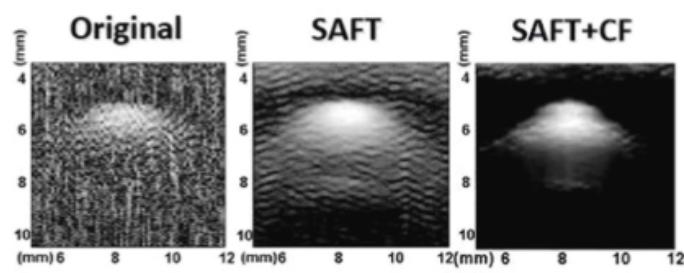


Abbildung 3.2: Ergebnis des 1D - SAFT, SAFT+CF, kopiert aus: [3]

# Kapitel 4

## Ergebnisse

In diesem Kapitel werden die Ergebnisse besprochen und ausgewertet. Dazu werden die Bilder betrachtet. Es gibt Fragestellungen die beantwortet werden sollen, "Hat sich der Kontrast verbessert?", "Ist der Dynamikumfang an Grauwerten angestiegen?". Anhand dieser Parameter soll bestimmt werden, ob eine Verbesserung stattgefunden hat. Zudem wird von dem B-Scan das berechnete Bild abgezogen um zu sehen, welche Unterschiede in den Bildern quantitativ zu sehen sind. Des Weiteren wird in diesem Kapitel der Algorithmus ausgewertet, "Wie komplex ist der Algorithmus geworden?".

Um zu verdeutlichen, wie die Qualitätsmerkmale sich in den unterschiedlichen Bildern äußern, werden im Folgenden Abbildungen eines B-Scans des Datensatzes an verschiedenen Position gezeigt. Um das Ergebnis des Algorithmus zu bewerten, werden Datensätze verwendet, die die Probe im Nah- und Fernfeld positioniert haben.

Die Abbildungen werden aus drei Bildern nebeneinander bestehen. Hierbei ist das linke Bild der B-Scan aus den Hilbert-Daten, dass mittlere Bild der B-Scan errechnet mit dem SAFT-Algorithmus und das rechte Bild die Subtraktion beider Bilder. Außerdem werden die B-Scans erneut gegenüber gestellt, zusätzlich wird noch das jeweilige Histogramm angezeigt.

Das Histogramm ist eine einfache Darstellung der absoluten Häufigkeiten aller Pixelwerte im Bild. Die Abbildungen beziehen sich auf den Querschnitt einer Münze.

Im Folgenden wird auf Erwartungen der Ergebnisse eingegangen.

### 4.1 Erwartungen

Von den Ergebnissen wird ein erhöhter Dynamikumfang der Grauwerte erwartet. Dies ergibt sich aus der Auswertung des Histogramms. Zudem sollten Artefakte mithilfe des Algorithmus verringert werden. Da die Probe zum einen im Nah- und zum anderen im Fernfeld liegt, sollten Ungenauigkeiten und die Auflösung verbessert werden.

Die folgende Abbildung zeigt eine unbearbeitete B-Scan Abbildung an Position 5 auf der y-Achse, parallel zur x-Achse.

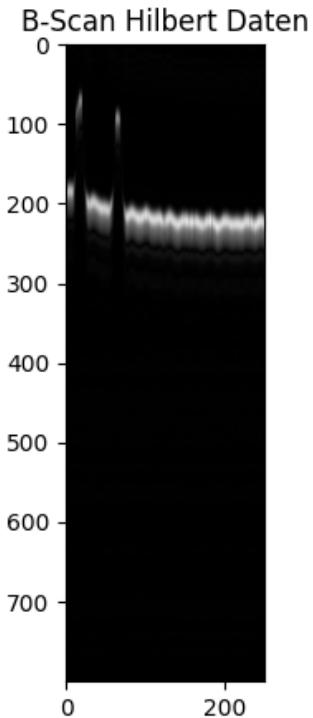


Abbildung 4.1: B-Scan mit der Probe im Nahfeld.

Abbildung 4.1 zeigt eine unveränderte Aufnahme des B-Scans. Zu erwarten ist, dass die Schleier um die Darstellung der Oberfläche verschwinden. Die zu erkennenden Wellenmuster werden nicht durch die synthetische Apertur verschwinden, da es sich um ein Wackeln des Wandlers handelt.

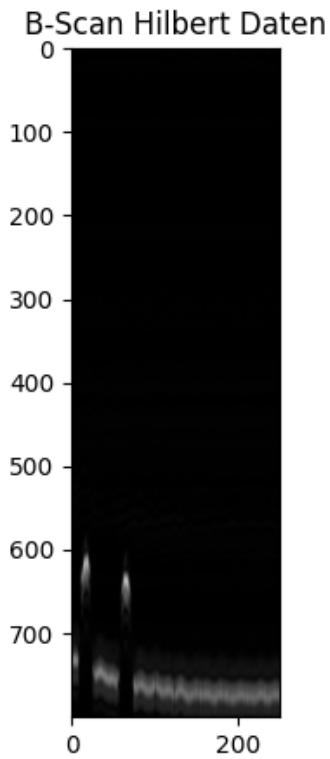


Abbildung 4.2: B-Scan mit der Probe im Fernfeld

Abbildung 4.2 zeigt wieder ein unverändertes B-Scan Bild, diesmal aber mit der Probe im Fernfeld des Schallfeldes. Hier wird die Ungenauigkeit durch die Defokussierung der Probe deutlicher. Durch den Algorithmus sollten wie bei Abbildung 4.1 erwartet, Ungenauigkeiten der Grenzen verbessert werden. Auch Artefakte sollten verschwinden.

## 4.2 Ergebnisse

In diesem Abschnitt werden die Ergebnisse des Algorithmus präsentiert. Zuerst werden die Ergebnisse im Nahfeld gezeigt. Anschließend die Ergebnisse im Fernfeld.

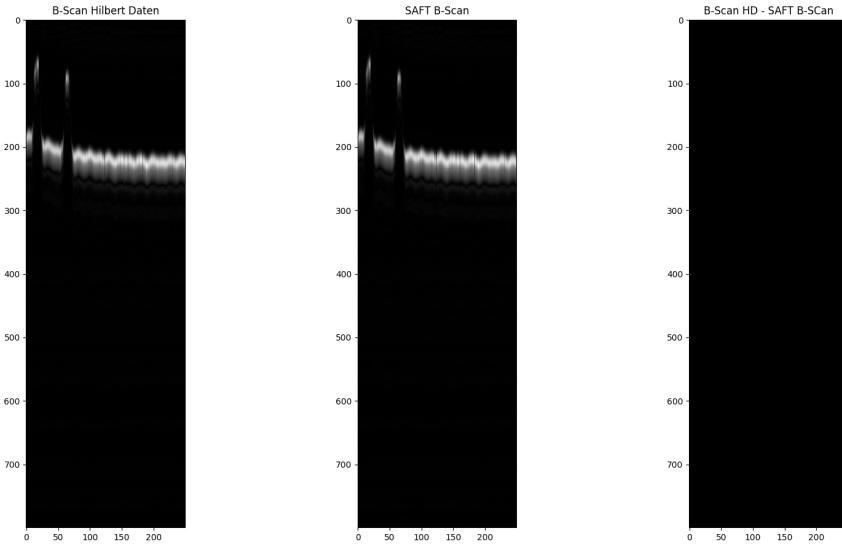


Abbildung 4.3: Links: B-Scan der Hilbert Daten, Mitte: SAFT-B-Scan, Rechts: Subtraktion des B-Scans und des SAFT B-Scans

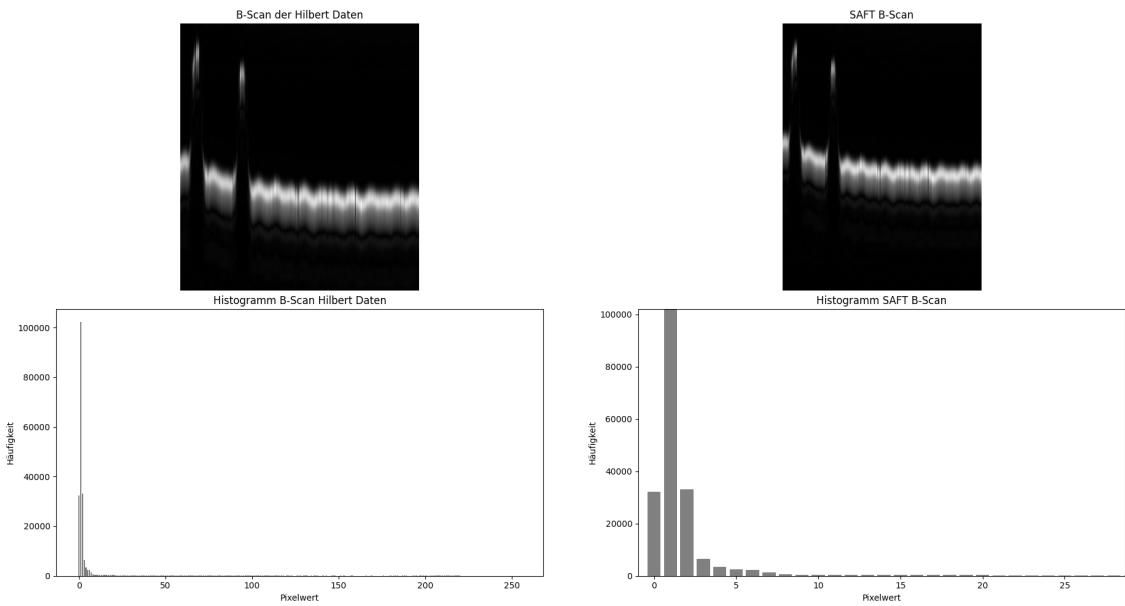


Abbildung 4.4: Auf der linken Seite sieht man den B-Scan in den Hilbert-Daten und rechts den errechneten SAFT B-Scan. Unter den Bildern sind jeweils die zugehörigen Histogramme

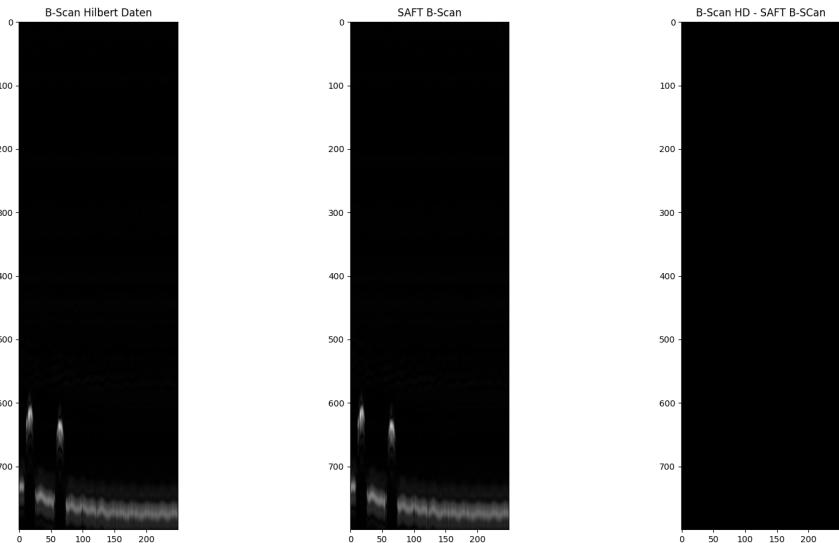


Abbildung 4.5: Links: B-Scan der Hilbert Daten, mitte: SAFT-B-Scan, Rechts: Subtraktion des B-Scans und des SAFT B-Scans

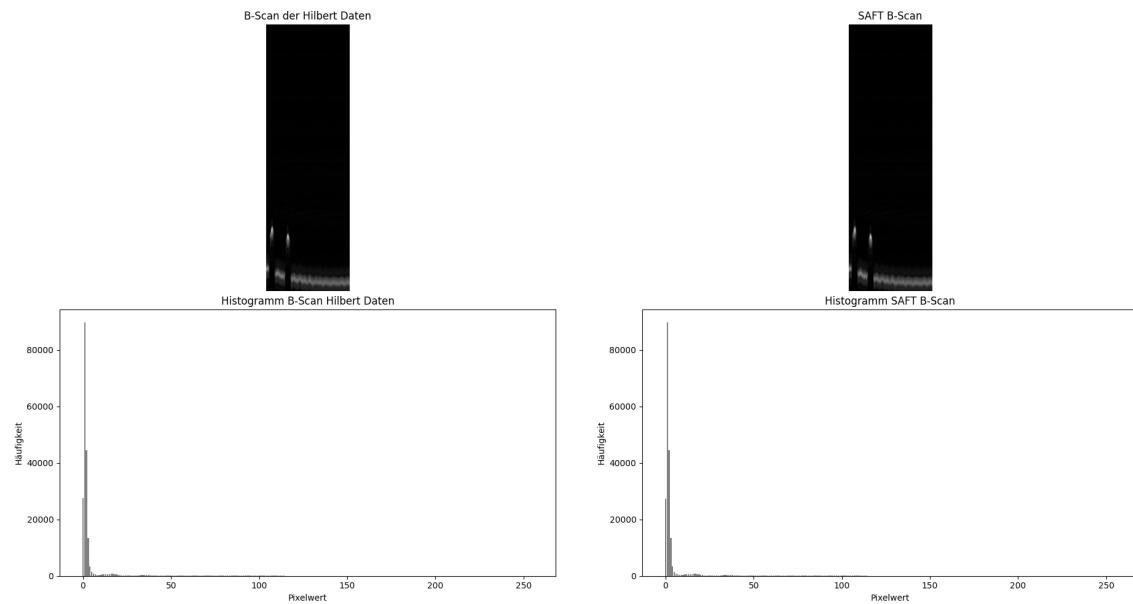


Abbildung 4.6: Auf der linken Seite sieht man den B-Scan aus den Hilbert-Daten und rechts den errechneten SAFT B-Scan. Unter den Bildern sind jeweils die zugehörigen Histogramme

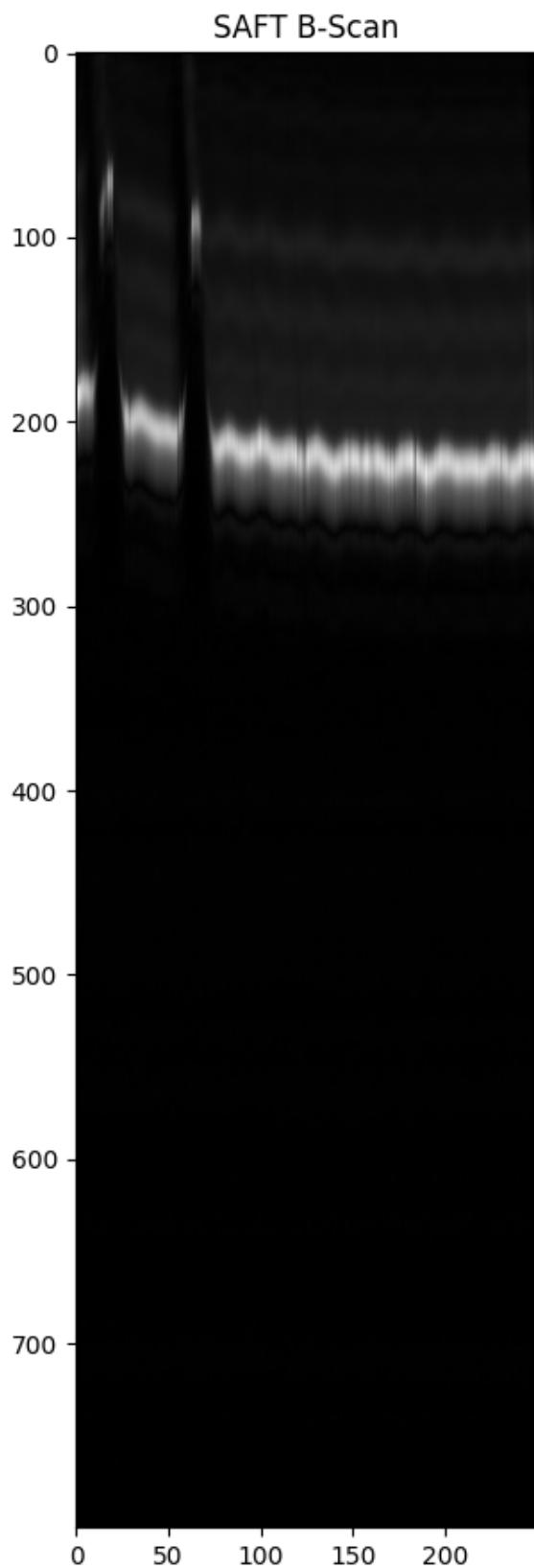


Abbildung 4.7: Abbildung zeigt das Normierungsartefakt, welches sichtbar wird, wenn man über die Anzahl an Pixeln normiert, die für die synthetische Apertur betrachtet wurden

# Kapitel 5

## Diskussion

In diesem Kapitel werden die Ergebnisse aus Kapitel 4 diskutiert und interpretiert. Es wird ein Zusammenhang mit der Aufgabenstellung aufgebaut und ausgewertet ob der Algorithmus wie erwartet funktioniert hat.

### 5.1 Interpretation der Ergebnisse

Zu dem Ergebnis des B-Scans im Nahfeld kann man sagen, dass es auffällige Punkte gibt. In Abbildung 4.3 kann man im rechten Bild nichts erkennen, das bedeutet, dass nichts an dem Bild verändert wurde. Dies beobachtet man auch, wenn man in Abbildung 4.4 die Ausschnitte der beiden B-Scans betrachtet. Auch die Histogramme zeigen, dass keine erkennbaren Veränderungen vorhanden sind.

Die gleichen Beobachtungen kann man für das SAFT-Ergebnis im Fernfeld machen. Im rechten Fenster kann man nur schwarze Pixel erkennen. Dies sagt aus, dass keine Veränderungen durch den Algorithmus entstanden sind. Wie auch im Nahfeld zeigen die Histogramme keinen Unterschied an.

Aus den Ergebnissen lässt sich schließen, dass der Algorithmus keinen Einfluss auf die Qualität des Bildes hat. Artefakte, die bei der Berechnung des Bildes durch den Algorithmus auftreten können sind Normierungsfehler. Normiert man in der Berechnung einer Pixelreihe mit der Anzahl an Punkten, die relevant für sie synthetische Apertur sind, so ist der Unterschied der Helligkeitswerte so groß, dass ein Bild mit sich wiederholenden Mustern zeigt. Betrachtet man einen einfachen B-Scan unverändert, so stellt man fest, dass die einzelnen Pixel, die schwarz dargestellt werden, nie null sind. Die Werte sind meist kleiner 0,1, was dazu führt, dass der Pixel schwarz wird. Das Normierungsartefakt ist in Abbildung 4.7 zu sehen. Man erkennt deutlich einen Schleier oberhalb der Probe, der sich nach oben weiter durchzieht. Dieses Artefakt tritt nicht auf, wenn man die Punkte über die Anzahl aller Pixel normiert.

## 5.2 Auswertung des Algorithmus

Der vollständige Code des Algorithmus wird im Anhang zur Verfügung gestellt(Kapitel A). Die auftretenden Berechnungen können durch den Verlauf des Algorithmus als korrekt eingestuft werden. Durch genaues Betrachten der Variablen während der Durchführung, kann ebenfalls gesagt werden, dass die Ergebnisse aller einzelnen Rechnungen zu den Dimensionen der physikalischen Daten passen.

Eine auffällige Stelle im Algorithmus ist jedoch die Berechnung der Gewichtungen der selektierten Signalanteile. Wie in den Grundlagen erklärt (Kapitel 2.4.1) werden die Gewichtungen mit einem Modell der PSF des Systems berechnet. Dieses Modell setzt sich aus mehreren Gauß-Funktionen zusammen. Die Halbwertsbreite wird mittels einer Variable  $\mu_{xy}$  angegeben. Diese Variable hängt von einem  $\sigma$  ab. In dieser Berechnung muss noch genauer nach Fehlern gesucht werden. Eventuell ist es auch nicht der richtige Ansatz, die PSF zu modellieren.

Außerdem ist eine potentielle Fehlerquelle das Rechnen in verschiedenen Koordinatensystemen. Dieser Punkt kann nicht einfach vernachlässigt werden, dennoch hat er großes Potential für Fehler in den Berechnungen zu sorgen.

Der Algorithmus funktioniert abseits dem Gewichtungsfaktor wie erwartet. Es wird auf die verschiedenen Koordinatensysteme geachtet, die in Kapitel 2.3 vorgestellt werden. Im Vergleich zu den vorgestellten Algorithmen für die synthetische Apertur(Kapitel 3.2) liefert der Algorithmus in dieser Arbeit kein vergleichbares Ergebnis.

Aus den Auswertungen der Ergebnisse kann geschlossen werden, dass es Fehler innerhalb des Algorithmus gibt. Die Fehler treten im Algorithmus 2 auf. Logische Fehler können hier für die vorliegenden Artefakte verantwortlich sein. Auch mathematische Fehler können nicht außer Acht gelassen werden. Um abschließend die Komplexität des Algorithmus zu analysieren, wird noch die  $\mathcal{O}$ -Notation betrachtet.

Die Komplexität des Algorithmus kann mithilfe der sogenannten  $\mathcal{O}$ -Notation beschrieben werden. Sobald der Algorithmus verschachtelte Schleifen hat, spricht das für eine lange Laufzeit. In der  $\mathcal{O}$ -Notation werden verschachtelte Schleifen "multipliziert". In dem Algorithmus dieser Arbeit, wird über eine ganze Reihe an Pixel iteriert. Zudem wird in dieser Schleife, mit zwei weiteren Schleifen, durch einen begrenzten Bereich im Datensatz gelaufen. Daraus ergibt sich eine  $\mathcal{O}$ -Notation von

$$\mathcal{O} = (n * n * n) = \mathcal{O}(n^3). \quad (5.1)$$

Das bedeutet, dass die Laufzeit kubisch mit der Größe der Eingabe anwächst. Der Algorithmus ist daher sehr anfällig für lange Laufzeiten. Da der größte verwendete Datensatz dreidimensional ist und das Format 1800x400x400 aufweist, und für jeden Pixel eine Be-

rechnung durchgeführt wird, die maximal eine Sekunde dauern würde, würde die Ausführung  $1800 \times 400 \times 400 \times 1$  Sekunden benötigen. Dies entspricht 9,1 Jahren. Das Ersetzen der Schleifen würde daher eine enorme Verbesserung der Laufzeit bedeuten. Abschließend lässt sich zum Algorithmus sagen, dass eine Optimierung des Algorithmus (siehe Algorithmus 2) im Hinblick auf wachsende Datensätze unumgänglich ist.

# Kapitel 6

## Fazit

Die Ergebnisse des Algorithmus sind nicht wie erwartet. Nach der Anwendung des Algorithmus, ist kein feststellbarer Unterschied auszumachen. Dennoch kann man im Blick auf die Thematik weiter ansetzen und Variationen im Bezug auf die Gewichtung relevanter Signalanteile einbringen und testen.

Das Vorgehen in der Arbeit ist der richtige Ansatz, doch es müssen noch viele Aspekte optimiert werden. Die Umsetzung der Theorie muss nochmal gründlich getestet werden. Das Verfahren der synthetischen Apertur wird immer ein sehr rechenintensives Verfahren sein, doch mit der Komplexität des Themas muss das Konzept sehr sorgfältig ausarbeitet werden.

Zur Laufzeit und Komplexität des Algorithmus ist zu sagen, dass es noch viel Optimierungsbedarf gibt. Um den Algorithmus einmal komplett laufen zu lassen vergehen im Durchschnitt mehrere Minuten. Dies variiert mit der Größe des Datensatzes. Im Kapitel 3.1.2 wird von einer geeigneten Art gesprochen, den Algorithmus in die Visualisierungsumgebung einzubinden. Mit den vorliegenden Ergebnissen macht es keinen Sinn, diese in eine Visualisierungsumgebung einzubinden.

Um das Problem der Gewichtung zu lösen, können auch Variationen oder Kombinationen der vorgestellten Methoden getestet werden.

Im Algorithmus 2 kommen mehrere Schleifen vor, die die Laufzeit und Komplexität steigern. Mit Numpy ist es möglich solche Schleifen durch Vektorisierung der Aufgaben zu ersetzen. Das ermöglicht zusätzlich, mit Funktionen von Numpy Berechnungen für alle Punkte im Array, gleichzeitig auszuführen. Zudem sind die Funktionen funktional programmiert. Übersetzt man die Struktur in eine Objekt-orientierte Form, so kann man auftretende Redundanzen, der Variablen verhindern. Dies führt zu geringeren Speichernutzungen des Arbeitsspeichers. Diese Verbesserungen könnten dazu beitragen, nicht nur den Algorithmus in vorliegender Form zu verbessern, sondern ermöglichen es auch den Algorithmus auf den kompletten Datensatz anzuwenden. Es würde das Selektieren der Daten obsolet machen.

Die vorgeschlagenen Verbesserungsmöglichkeiten konnten aufgrund zeitlicher Beschränkungen nicht mehr realisiert werden. Ursprünglich war die Algorithmusauswahl anders angesetzt, doch die Einführung neuer Datensätze führte zu unerwarteten Problemen.

## 6.1 Ausblick

Als Ausblick für diese Arbeit und dieses Thema ist zu sagen, dass bekannte Vorgehensweisen zur Optimierung bereits bekannt sind und deshalb schnell umgesetzt werden können. Mithilfe von Numpy-Vektorisierung und weiterer Parallelisierung durch die GPU mit Schnittstellen wie CUDA, sollte es möglich sein die Laufzeit deutlich zu verbessern. CUDA ist eine Schnittstelle, mit der es möglich ist Daten und Berechnungen auf die GPU des Computers zu ziehen. CUDA speziell wird von Nvidia zur Verfügung gestellt und benötigt eine Nvidia Grafikkarte.

Da, vor allem neuartige, GPUs darauf ausgelegt sind, Rechenoperationen auf große Arrays durchzuführen sollte es auch möglich sein, die Berechnung so schnell auszuführen, dass man den Algorithmus in Analysesysteme implementieren kann.

Für das Thema der synthetischen Apertur in der akustischen Mikroskopie gibt es mit dem vorgestellten Ansatz von Gao Rongkang (et. al) aus dem Jahr 2022, gute Ansätze, die eventuell auch in diesem Ansatz übernommen werden können. Darunter die Art, wie die Summe der Signalanteile gewichtet wird [3]. Die CF-Wichtung erzielt gute Ergebnisse und wirkt noch gegen Störungen und Artefakte. Es wäre möglich die CF-Wichtung mit in den Algorithmus aufzunehmen und zu testen ob man sie zusätzlich zur Punktspreizfunktion multipliziert.

# Anhang A

## Anhang

### A.1 Code zum Algorithmus

```
1 import numpy as np
2 from multiprocessing import cpu_count
3 from concurrent.futures import ProcessPoolExecutor as ppe
4 import time
5 from esm_loader import esm_auslesen
6 from matplotlib import pyplot as plt
7 import random
8 import matplotlib.colors as mcolors
9 from matplotlib import gridspec
10 from scipy.fft import fft2, fftshift
11 from scipy.signal import hilbert
```

Abbildung A.1: Importierte Bibliotheken

```

13  def Data2HilbertData(data):
14      ...
15      Berechnet den Hilbertdatensatz.
16      Es wird durch alle Pixel im Raster X-Y Iterriert und zu jedem Punkt alle Z-Werte genommen.
17      Jeder Punkt, der in der For-Schleife behandelt wird, stellt ein akustisches Signal dar.
18      Das Signal wird mit der Hilbert-Transofrmation transformiert.
19
20      Rückgabe der Funktion ist ein Datensatz mit gleichem Format.
21      ...
22
23      # Array Format
24      z, x = np.shape(data)
25
26      # Hilbert array erstellen im gleichem Format wie Rohdatensatz
27      hilbert_data = np.zeros_like(data)
28
29
30      for j in range(x):
31
32          signal = data[:,j]
33          signal = np.abs(hilbert(signal))
34          hilbert_data[:,j] = signal
35
36
37      # # Jedes Signal wird nun transformiert
38      # for i in range(y):
39      #     for j in range(x):
40      #         signal = data[:, i, j]
41      #         #print(len(signal))
42      #         signal = np.abs(hilbert(signal))
43      #         hilbert_data[:, i, j] = signal
44
45      # Ausgabe der Funktion
46      print('Datenformat der transformierten Daten: ', np.shape(hilbert_data))
47      return np.array(hilbert_data)

```

Abbildung A.2: Funktion die die Hilbert-Transformation anwendet

```

def saft_für_eine_reihe_berechnen(tuple_id, position, sample_DK, img_row, selektierte_daten, schallgeschwindigkeit, bereich_um_den_punkt_der_apertur, mu_x, mu_xy, sigma):

    zd, yd, xd = np.shape(selektierte_daten) # für bereich 0 => (800, 1, 100)
    bild_reihe_für_berechnete_x = np.zeros(xd + 1, dtype=np.float64)

    bild_reihe_für_berechnete_x[0] = tuple_id

    samplerate = 2e9
    alpha = 0.01
    fokuslaenge = 72.77e-6
    dampfFing = np.exp(-2* alpha)
    stepsize = 0.05e-3

    hardware_delay_EK = 5000 # später aus dem Datensatz entnehmen
    sample_position_des_ersten_vorlauf_echoes_EK = 3140 # delay hardware = 50000, 1. VE = 3140 Linse

    delay_vom_ersten_vorlaufecho_in_samples_LK = hardware_delay_EK - sample_position_des_ersten_vorlauf_echoes_EK

    fokusentfehrung_in_meter_LK = 72.77e-6
    fokusentfehrung_in_zzeit_LK = fokusentfehrung_in_meter_LK * (schallgeschwindigkeit / 2)
    z_wandler = 0

    You, last week * neuere ansatz der SAFT darstellung ...
    mu_xy = 2 * np.sqrt(2 * np.log(2)) * sigma * 10e-5
    x_Achse = [i * stepsize for i in range(xd)]

    for i, x in enumerate(img_row):
        liste_aller_SAFT_werte_DK = []

        if (i - bereich_um_den_punkt_der_apertur) <= 0 :
            selektierte_daten_mit_x_grenzen = selektierte_daten[:, :, 0: (i + bereich_um_den_punkt_der_apertur + 1)]
        elif (i + bereich_um_den_punkt_der_apertur) > xd:
            selektierte_daten_mit_x_grenzen = selektierte_daten[:, :, (i - bereich_um_den_punkt_der_apertur):]

        elif (i - bereich_um_den_punkt_der_apertur) > 0 and (i + bereich_um_den_punkt_der_apertur) < xd:
            selektierte_daten_mit_x_grenzen = selektierte_daten[:, :, (i - bereich_um_den_punkt_der_apertur):(i+berreich_um_den_punkt_der_apertur+1)]


        bereich_x = int(np.shape(selektierte_daten_mit_x_grenzen)[2] / 2)
        bereich_x = (-berreich_x, berreich_x)
        bereich_y = int(np.shape(selektierte_daten_mit_x_grenzen)[1] / 2)
        bereich_y = (-berreich_y, berreich_y)

```

Abbildung A.3: Python Code zu Algorithmus 2

```

for yw in range(bereich_y[0], bereich_y[1]):
    for xw in range(bereich_x[0], bereich_x[1]):

        # z-Wert in einen Zeitwert umrechnen
        zeitwert_rekonstruktionsebene_DK = sample_DK/ samplerate
        # Vorabberechnung der Positionsunterschiede auf dem x- und y- Raster
        delta_x = (i * stepsize) - ((i + xw) * stepsize)
        delta_y = (position * stepsize) - ((yw + position) * stepsize)
        x_e_funktion = [np.exp(-i - (x*stepsize)**2 / mu_xy) for i in x_Achse]
        # ÜBERGANG IN DAS LINSENKOORDINATESYSTEM
        ## z-Wert im Linsenkoordinatensystem

        zeitwert_rekonstruktionsort_senkrecht_im_wasser_LK = ((zeitwert_rekonstruktionsebene_DK) + (delay_vom_ersten_vorlaufecho_in_samples_LK / samplerate)) # IN SEKUNDEN

        # Zeitwert in einen Abstand umgerechnet
        abstand_rekonstruktionsort_senkrecht_im_wasser_LK = (zeitwert_rekonstruktionsort_senkrecht_im_wasser_LK * schallgeschwindigkeit)/2

        # Abstand zum SAFT-Wandler berechnen
        abstand_im_wasser_zum_SAFT_wandler_LK = np.sqrt(abstand_rekonstruktionsort_senkrecht_im_wasser_LK**2 + (delta_y)**2)

        # Abstand in eine Zeit umrechnen
        zeit_im_wasser_zum_SAFT_wandler_LK = abstand_im_wasser_zum_SAFT_wandler_LK / schallgeschwindigkeit * 2

        # Zeitwert in einen Samplewert umrechnen
        zeit_im_wasser_zum_SAFT_wandler_in_samples_LK = int(zeit_im_wasser_zum_SAFT_wandler_LK * samplerate)

        # Delta t im Linsenkoordinatensystem
        dt_LK = zeit_im_wasser_zum_SAFT_wandler_LK - zeitwert_rekonstruktionsort_senkrecht_im_wasser_LK
        dz_LK = dt_LK * schallgeschwindigkeit / 2

        # Berechnung Wichtung
        gewichtung = np.exp(-(daempfung * abstand_im_wasser_zum_SAFT_wandler_LK**2) * x_e_funktion[int(position + yw)] * x_e_funktion[int(x + xw)])
        zeit_im_wasser_zum_SAFT_wandler_in_samples_DK = zeit_im_wasser_zum_SAFT_wandler_in_samples_LK - delay_vom_ersten_vorlaufecho_in_samples_LK
        # Wichten des Punktes
try:
    saft_wert_DK = selektierte_daten[zeit_im_wasser_zum_SAFT_wandler_in_samples_DK, position_yw, i + xw] * gewichtung
    liste_aller_SAFT_werte_DK.append(saft_wert_DK)
except:
    pass

rekonstruierter_wert_DK = np.sum(liste_aller_SAFT_werte_DK) / (yw * xw)
bild_reihe_für_berechnete_x[i+1] = x + rekonstruierter_wert_DK

return bild_reihe_für_berechnete_x

```

Abbildung A.4: Python Code zu Algorithmus 2  
VI

```

1[0]6    > def Worker_stuff(args) -> np.ndarray:
1[0]7
1[0]8     b_scan_rows = []
1[0]9
1[10]    _25_reihen_ = args
1[11]
1[12]    print(len(_25_reihen_))
1[13]    for i, l in enumerate(_25_reihen_):
1[14]        tuple_id, position, z_zeitwert_der_aktuellen_ahene, img_row, selektierte_daten, schallgeschwindigkeit, bereich_um_den_punkt_der_spatialur, mu_z, mu_xy = l[0], l[1], l[2], l[3], l[4], l[5], l[6], l[7], l[8]
1[15]        b_scan_reihe = soft_für_eine_reihe_berechnen(tuple_id, position, z_zeitwert_der_aktuellen_ahene, img_row, selektierte_daten, schallgeschwindigkeit, bereich_um_den_punkt_der_spatialur, mu_z, mu_xy)
1[16]        b_scan_rows.append(np.asarray(b_scan_reihe))
1[17]
1[18]    # testliste = np.random.rand(40), tolis()
1[19]    # test liste = np.array(test liste)
1[20]    # b_scan_rows.append(test liste)
1[21]    print(f'Reihe: {i} fertig')
1[22]    print(f'Länge der b_Scan_rows: , len(b_Scan_rows))'
1[23]    print(f'Länge des ersten elements der b_Scan_rows: , len(b_Scan_rows[0]))'
1[24]
1[25]    return b_scan_rows

```

Abbildung A.5: Zuteilung der Aufgaben für jeden Prozess

```

def saft_ausführen(data, achse, position, samplerate, stepsize, hardware_delay_samples, lens_vorlaufecho_samples):
    # Konstanten
    c = 1485
    zd, yd, xd = np.shape(data)
    delay_LK_samples = hardware_delay_samples - lens_vorlaufecho_samples
    delay_LK_abstand = delay_LK_samples / samplerate * c / 2

    apertur_durchmesser = 750.87E-6      # meter
    focus_breite = 9.107e-6              # meter
    schallgeschwindigkeit= 1485          # meter pro sekunde
    bereich_um_den_punkt_der_apertur = 7
    samples_DK = [i for i in range(zd)]
    samples_LK = zd + delay_LK_samples

    alle_abstaende_LK = [(i / samplerate) * schallgeschwindigkeit) / 2 for i in range(samples_LK)]
    max_abstand_LK = alle_abstaende_LK[-1]

    x1, y1 = 0, apertur_durchmesser/2
    x2, y2 = max_abstand_LK/2, focus_breite
    x3, y3 = max_abstand_LK, apertur_durchmesser/2

    punkte= [(x1, y1),(x2, y2),(x3, y3)]
    sigma_modell_koeffizienten_LK = fit_parabel(punkte, max_abstand_LK/2)
    print(f'Koeffizienten (a, K): {sigma_modell_koeffizienten_LK}')
    sigma_liste = sigma_berechnen(sigma_modell_koeffizienten_LK, alle_abstaende_LK, x2)

    mu_z = 2 * np.sqrt(2 * np.log(2)) * sigma_liste[-1]
    mu_xy = 0

    if achse == 'x':
        if (position - bereich_um_den_punkt_der_apertur) < 0:
            selektierte_daten = data[:, 0:(position + bereich_um_den_punkt_der_apertur + 1), :]
        elif (position + bereich_um_den_punkt_der_apertur) > yd:
            selektierte_daten = data[:, (position- bereich_um_den_punkt_der_apertur):, :]
        elif (position-bereich_um_den_punkt_der_apertur) > 0 & (position + bereich_um_den_punkt_der_apertur) < yd:
            selektierte_daten = data[:, (position - bereich_um_den_punkt_der_apertur):(position + bereich_um_den_punkt_der_apertur + 1), :]

        img_array = np.zeros((zd, xd))

        argumente_tupel_für_alle_reihen = [] # Liste mit zd Tupeln
        for i, sample_DK in enumerate(samples_DK):
            img_row = img_array[i]
            img_row = data[i, position, :]
            tupel_id = i
            sigma = sigma_liste[i]
            argumente_für_die_funktion = (tupel_id, position, sample_DK, img_row, selektierte_daten, schallgeschwindigkeit, bereich_um_den_punkt_der_apertur, mu_z, mu_xy, sigma)
            #print(argumente_für_die_funktion)
            argumente_tupel_für_alle_reihen.append(argumente_für_die_funktion)

batch_size = int(zd / cpu_count())
argumente_unterteilt_in_batch_listen = [] # 16 Listen mit jeweils 25 Listen mit jeweils 400 Tupeln

```

Abbildung A.6: Python Code zu Algorithmus 1

```

for i in range(cpu_count()):
    batch = argumente_tupel_für_alle_reihen[(i*(batch_size)):((batch_size) * (i+1)) ]
    argumente_unterteilt_in_batch_liste.append(batch)

with ppe(max_workers=cpu_count()) as executor :
    liste_mit_ergebnissen = [] # Liste mit 16 Listen
    i = 0
    for args in argumente_unterteilt_in_batch_liste:

        ergebnis_einer_reihe = executor.submit(Worker_stuff, args)
        i += 1
        ergebnis_einer_reihe.add_done_callback(custom_callback)
        liste_mit_ergebnissen.append(ergebnis_einer_reihe)

    liste_mit_richtigen_ergebnissen = []
    for i in liste_mit_ergebnissen:
        ergebnis = i.result()
        liste_mit_richtigen_ergebnissen.append(ergebnis)

bildreihen = []
for i in range(np.shape(liste_mit_richtigen_ergebnissen)[0]):
    for j in range(np.shape(liste_mit_richtigen_ergebnissen)[1]):

        reihe = liste_mit_richtigen_ergebnissen[i][j][1:]
        bildreihen.append(reihe)

SAFT_bild = np.array(bildreihen)
You, 1 second ago * Uncommitted changes
return SAFT_bild

```

Abbildung A.7: Python Code zu Algorithmus 1

```

183     if __name__ == '__main__':
184
185         # data = np.random.rand(400, 400, 400)
186
187         _, _, data = esm_auslesen(r"D:\\Test Daten Bachelor\\test daten\\fokus_weiter_oben.esm")
188         print('daten ausgelesen')
189         data = data[1300:-100, :, :]
190         print(f'Daten zugeschnitten auf: {np.shape(data)}')
191         samplerate = 2e9
192         start = time.time()
193         saft_data = saft_ausführen(data, 'x', 100, samplerate)
194         ende = time.time() - start
195         print(f'Algorithmus dauert {ende / 60} Minuten.')
196
197
198         b_img = saft_data
199

```

Abbildung A.8: Start des Algorithmus

# Literaturverzeichnis

- [1] Golafsoun Ameri. *Synthetic aperture imaging in acoustic microscopy*. PhD thesis, Ryerson University Toronto, ON, Canada, 2011.
- [2] H Boche. Verhalten der Hilbert-Transformation for quasibandbegrenzte Signale. Technical report, 1998.
- [3] Rongkang Gao, Qiang Xue, Yaguang Ren, Hai Zhang, Liang Song, and Chengbo Liu. Achieving depth-independent lateral resolution in AR-PAM using the synthetic-aperture focusing technique, 6 2022.
- [4] Kibero GmbH. easySAM, 2015.
- [5] Erdal Oruklu, Yufeng Lu, and Jafar Saniie. Hilbert Transform Pitfalls and Solutions for Ultrasonic NDE Applications. *2009 IEEE International Ultrasonics Symposium*, 2009.
- [6] Weiqi Qian, Weiyou Yang, Yan Zhang, Chris R Bowen, and Ya Yang. Piezoelectric Materials for Controlling Electro-Chemical Processes. *Nano-micro letters*, 12(1):149, 7 2020.
- [7] Rüdiger Kramme, Rudolf Götz, and Frank Schön. *Medizintechnik*. Springer Berlin Heidelberg, 2017.
- [8] Markus Schartel. *Signalverarbeitungskonzepte zur Minendetektion mittels drohnengestütztem Ground Penetrating Synthetic Aperture Radar*. PhD thesis, Universität Ulm, 2021.
- [9] Schlegel Wolfgang, Karger Christian P., and Jäkel Oliver. *Medizinische Physik: Grundlagen - Bildgebung - Therapie - Technik*. Springer Spektrum, 2018.
- [10] Zuliang Yu and Siegfried Boseck. Scanning acoustic microscopy and its applications to material characterization. *Reviews of Modern Physics*, 67(4):863, 10 1995.

# Eigenständigkeitserklärung

Hiermit erkläre ich, Philipp Bellia, geboren am 28.04.1999, an Eides statt, dass ich die vorliegende Bachelorarbeit mit dem Titel „Implementierung und Validierung eines Synthetischen Apertur Focusing Technique Algorithmus für die akustische Mikroskopie“ selbstständig und ohne fremde Hilfe angefertigt habe.

Ich habe hierfür keine anderen Quellen und Hilfsmittel als diejenigen herangezogen und benutzt, die in der Bachelorarbeit bei jeder einzelnen Verwendung ausdrücklich und umfassend genannt werden.

Die Arbeit haben in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen. Mir ist bewusst, dass ein Zuwiderhandeln gegen diese Erklärung schwerwiegende Konsequenzen für mein weiteres Studium nach sich ziehen wird.

Saarbrücken, den 27.09.2023

Unterschrift: