Writeup zu „PBL:Einführung in medizintechnische Systeme“, Gruppe 02

# Problem

Die Intravaskuläre Optische Kohärenztomographie (IVOCT) ist eine medizinische Bildgebungsmodalität, die eine kurzkohärente Lichtquelle nutzt um Tiefenprofile von Gewebe zu erstellen. Dafür wird der Lichtstrahl in einen Probe- und Referenzarm aufgespalten und beide an der Probe bzw. dem Referenzspiegel reflektierte Strahlen werden in einem Interferometer verglichen um die Wegstrecke des Probestrahls zu ermitteln. Dabei ist vor allem (ähnlich wie beim Ultraschall) die Reflektivität der einzelnen Gewebeschichten von Bedeutung, diese ist ausschlaggebend für das entstehende Interferenzmuster und damit für das Bild. Die einzelnen 1-dimensionalen Tiefenprofile (A-Scan) werden dabei in Serie entlang einer Gewebeprobe aufgenommen um 2-dimensionale Schichtbilder zu erstellen (B-Scan).

Eins der Haupteinsatzgebiete der in diesem Projekt behandelten IVOCT ist die Bildgebung bei der Behandlung der Koronaren Herzkrankheit, die durch Ablagerungen an den Gefäßwänden entsteht und unbehandelt zu einem lebensgefährlichen Herzinfarkt führen kann. Diese Modalität ermöglicht neben der Computertomographie eine sichere und zuverlässige Navigation und Untersuchung innerhalb der Koronargefäße. Eine der wichtigen Aufgaben der IVOCT bei der Behandlung von Koronarer Herzkrankheit ist die Hilfestellung bei der ursprünglichen Diagnose dieser, wobei das Lumen der Koronargefäße vermessen und somit auf vorhandene Stenosen untersucht wird.

Anhand dieser Grundvoraussetzungen ist die wesentliche Problemstellung des Projektes die Erstellung und Verarbeitung der in einem Gefäßphantom aufgenommenen Rohdaten und die anschließende Ermittlung des Durchmesserverlaufs entlang mehrerer Aufnahmen. Dabei gliedert sich die anfallende Arbeit in die Aufnahme der Rohdaten im Labor, die Berechnung des Bildes über die Rohdatenverarbeitung und die anschließende Bildverarbeitung. Die Bildverarbeitung ist der Kern des Problems, da diese im Gegensatz zu den beiden anderen Komponenten eine Vielzahl an Lösungsansätzen zulässt und somit die mit Abstand größte Herausforderung bei der Problemlösung darstellt.

Die sich daraus ergebende Problemlösung sollte anschließend in der vorliegenden schriftlichen Ausarbeitung festgehalten und vor den anderen Gruppen kurz präsentiert werden.

# Material und Methoden

## Git und die Modul-Aufteilung

Bei diesem Programmier-Projekt mit 4 Teilnehmern und insgesamt ca. 1400 Zeilen Code haben wir uns für Git und für eine Aufteilung des Codes in Module entschieden. Außerdem haben wir diese Module dann bestimmten Gruppenteilnehmern zugeordnet. Dieser Ansatz hat den Autoren individuell viele Freiheiten gelassen, während immer die Zuständigkeit klar war und über die Versionskontrolle sowie auch über den Modul-Aufbau eine gewisse „Code-Sicherheit“ gegeben wurde. Später haben die Autoren sich noch dazu entschieden die Parameter zu jedem Modul am Anfang des jeweiligen Moduls zu deklarieren, was nun auch das Writeup erleichtert.

Angelehnt an die technische Problemstellung beinhaltet unser Projekt 6 Module:

* Main :: Ruft Module nacheinander auf
* Rohdaten-Verarbeitung :: Konvertiert die Rohdaten zu einem M-Scan
* M-Scan to B-Scan :: Segmentiert die einzelnen B-Scans im M-Scan
* Filter Artefacts :: Vorverarbeitung für reibungslose Berechnung des Durchmessers
* DetermineDiameter :: Berechnung des Durchmessers
* GUI ::Graphische Bedienmöglichkeit der Main

## Die Implementierung über Matlab

Der Code ist komplett in Matlab geschrieben, wie von den Tutoren vorgeschlagen. Die Funktionen, die Matlab per default mitliefert waren dabei komplett ausreichend.

Lediglich den Teil, der das kartesische und polare Koordinatensystem hin und zurück konvertiert, haben wir kopiert, dieser wurde ursprünglich von 2007 von Prakash Manandhar ([pmanandhar@umassd.edu](mailto:pmanandhar@umassd.edu)) implementiert.

# Ergebnisse

## Vorverarbeitung

Zuerst wird eine Rohdatenverarbeitung vorgenommen. Dazu wird der Rohdatensatz als 2D-Matrix, der vom Hersteller gegebene Dunkelstrom und die Position der nichtlinearen Stützstellen in den Workspace geladen.

Daraufhin wird von jeder Spalte der Rohdatenmatrix, welche nach der Fourier-Transformation einen A-Scan darstellt, der Dunkelstrom subtrahiert. Der DC-Term wird bestimmt, indem der Mittelwert jeder Zeile der Matrix errechnet wird. Dieser wird dann wiederrum von der Matrix subtrahiert. Ein häufiges Problem bei der Fourier-Transformation ist der Leck-Effekt. Um die resultierenden Seitenbänder zu reduzieren wird jedes Spektrum mit dem Hann-Fenster multipliziert, die sogenannte Apodisation. Des Weiteren werden die nichtlinearen Stützstellen linear in den k-Raum interpoliert. Die Fourier Transformation überführt die Spektren in A-Scans. Da das erhaltene Signal komplex ist, wird der Betrag des Signals genommen und für eine bessere Darstellung logarithmisch komprimiert und in Grauwerte umgewandelt.

Der nun entstandene M-Scan (Aneinanderreihung der A-Scans) wird für die Bildverarbeitung erst einmal in B-Scans aufgeteilt. Ein B-Scan stellt einen Querschnitt der untersuchten Arterie dar. Da der Katheter meist nicht genau mittig in der Arterie liegt entsteht ein Bild welches Maxima und Minima für die Katheter-Artefaktlinien und Arterienwände aufweist. Der Bereich von einem Maximum zum nächsten gibt an, dass der Katheter eine Drehung von 360° gemacht hat und dementsprechend ergibt dieser Bereich einen B-Scan. Um die Maxima ausfindig zu machen wird zunächst das Rauschen durch einen Median-Filter verringert. Die Artefaktlinie des Prismas wird entfernt. Der Prewitt-Kantenfilter, angewendet auf den Bereich des Katheter-Artefaktes, ergibt eine binäre Matrix bestehend aus Nullen und Einsen. Die Einsen repräsentieren hierbei die Artefaktlinien. Da die Katheter-Artefaktlinie direkt unterhalb der Prisma-Artefaktlinie liegt, wird die Zeile unterhalb des entfernten Prisma-Artefakts abgescannt und der Abstand der Einsen in einen Vektor gespeichert. Daraufhin werden kleine Abstände vernachlässigt, der Vektor enthält nun die Positionen der Maxima. Der M-Scan wird genau an diesen Positionen in B-Scans aufgeteilt.

## Verarbeitung

Die Bildverarbeitung erzeugt aus jedem B-Scan ein kartesisches Bild. Zusätzlich wird die innere Wand der Arterie segmentiert und der Durchmesser bestimmt.   
Dazu werden als erstes die Artefakte aus den Scans entfernt indem der Bildbereich oberhalb der zu segmentierenden Arterienwand einem Grauwert gleichgesetzt wird. Mit Hilfe einer Kontrastverstärkung, eines mehrfach angewendeten Median-Filters, einer Transformation in ein schwarz-weißes Bild und einer for-Schleife wird das Artefakt, welches teilweise die Arterie durchzieht, ermittelt und entfernt. Dabei überprüft die for-Schleife, anhand bestimmter Punkte unterhalb und neben den untersuchten Bildpunkten, ob diese zum Artefakt oder zur Arterie gehört.

Eine ähnliche Schleife setzt alle Werte unterhalb der Arterienwand auf schwarz. Die Kante zwischen schwarz und weiß kann nun detektiert und gefiltert werden. Das Bild wird interpoliert und in das kartesische Koordinatensystem transformiert. Der Flächenmittelpunkt der Arterie wird ermittelt und so verschoben, dass dieser mit dem Mittelpunkt des Bildes übereinstimmt. Es erfolgt die Transformation des originalen B-Scans in kartesische Koordinaten, die segmentierte Kante wird hinzu projiziert und zeigt im Bild die Arterienwand an.

Die Bestimmung des Durchmessers erfolgt in Polarkoordinaten anhand der detektierten Kante. Es wird unter der Annahme, dass die Rotation des Katheters konstant erfolgt ist, der Abstand von einem Punkt der Kante und dem Punkt nach einer halben Drehung (halber B-Scan) zur X-Achse gemessen und aufaddiert.

## GUI

Die GUI bietet eine komfortable Ansteuerung der einzelnen Module um diese nacheinander ausführen zu können. Den komplexen Anforderungen einer echten medizinischen Anwendung genügt sie bei Weitem nicht, allerdings ist es ein erster Versuch der Autoren in diese Richtung.

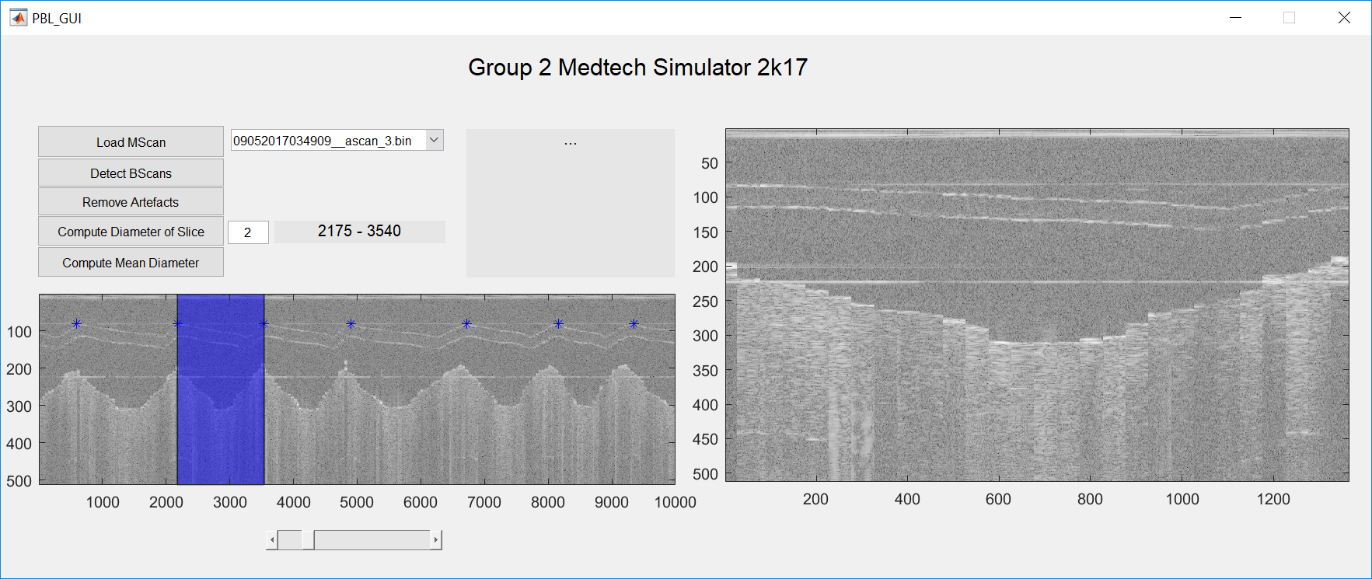
Über den Block an Buttons oben Links hat der Anwender die Möglichkeit aus den Messungen eine auszuwählen und zu laden. Danach können über dem Button darunter „DetectBScans“ die B-Scans segmentiert werden. Aus diesen wird dann direkt einer ausgewählt. In der Übersicht unten links werden alle erkannten B-Scans mit blauen Stern-Markierungen gekennzeichnet, welche über die darunterliegende Scrollbar durchgeschaltet werden können. Der aktuell ausgewählte B-Scan kann sofort in der großen Ansicht in der rechten Hälfte genauer betrachtet werden.

Abbildung 1: GUI nach Laden des M-Scans und Detektieren der B-Scans

Nur nach einem anschließenden Entfernen der Artefakte über den dazugehörigen Button kann für den ausgewählten B-Scan der Durchmesser berechnet werden. Dabei wird die Innenwand segmentiert, welche auf der rechten Hälfte der GUI in rot markiert wird (siehe Abbildung 2: GUI nach Filterung der Artefakte und Berechnung eines segmentierten B-Scans). Der durchschnittliche Durchmesser des B-Scans wurde in diesem Beispiel auf einen Wert von 1.77 umgerechnet.

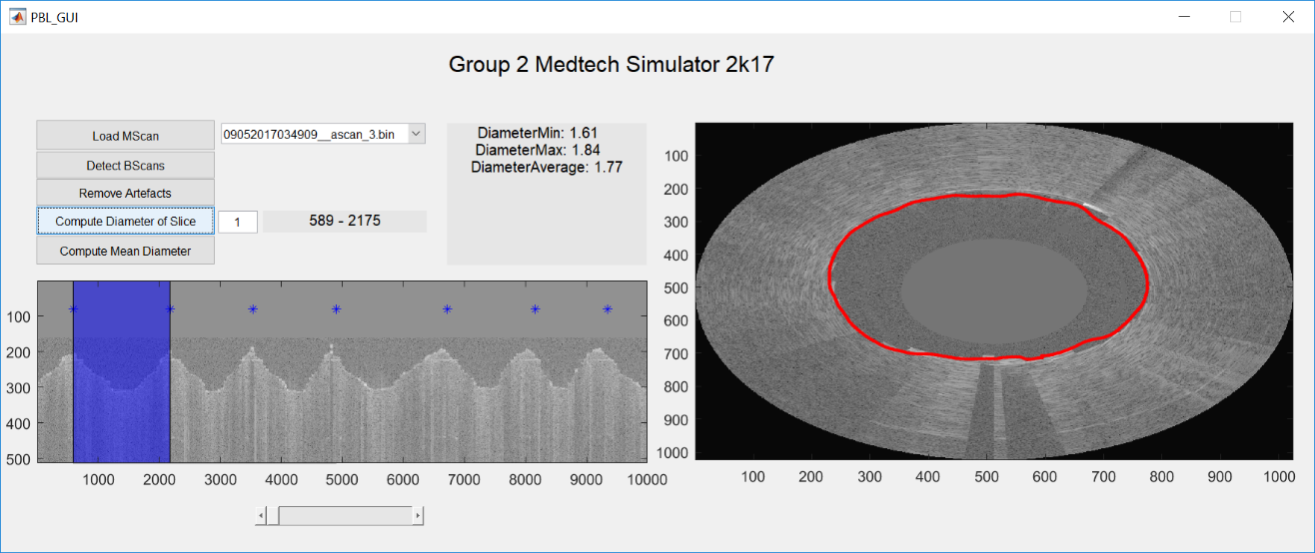


Abbildung 2: GUI nach Filterung der Artefakte und Berechnung eines segmentierten B-Scans

# Diskussion

## Variation der Parameter

Der Projekt-Code beinhaltet eine Vielzahl an Parametern (siehe Abbildung 3). Die verschiedenen Filter zur Segmentierung und Vorverarbeitung so anzuwenden, dass diese wiederum mit dem nächsten Filter zusammenspielen, war eine große Hürde zu Anfang des Projektes. Auch das Fine-Tuning war aufwändig: Das Anpassen eines einzelnen Parameters konnte dazu führen, dass eine Menge an anderen Filtern nicht mehr wie erwartet funktioniert haben. Parameter zu verbessern war dadurch eine sehr umfangreiche Aufgabe. In diesem Teil wird untersucht, wie stark die Funktionalität des erstellten Codes von einer Variation der Parameter abhängig ist.

Abbildung 3: Anzahl an Parametern Pro Modul

Für die Segmentierung des M-Scans in B-Scans ist in Abbildung 4 eine Variation der Parameter aufgeführt. Dabei wurde jeder Parameter einzeln variiert, es wurden also nicht mehrere Parameter gleichzeitig verändert, um deren Einfluss aufeinander erkennen zu können. Die enthaltenden Parameter sind vorwiegend Höhenpositionen bis auf eine Grauwert -„Param Line“ und einen horizontalen Abstandswert „DistanceParam“. Bei der Variierung wurden zum aktuellen Wert -10 bis 10 addiert, wobei das bei vereinzelten Werten durch Matrix-Begrenzungen oder Begrenzungen eines anderen Grenzwertes nicht möglich war. Diese Felder sind in der Abbildung weiß markiert. Der eingetragene Wert in der Tabelle bezieht sich dann auf die segmentierten B-Scans im aktuellen M-Scan, 7 war hier die richtige Anzahl und ist daher entsprechend in Grün markiert. Bereits einen einzigen B-Scan nicht zu erkennen ist im medizinischen Kontext schon enttäuschend, dieser würde dann verloren gehen oder eventuell benachbarte B-Scans verfälschen. Das spiegelt wieder, dass der optimale Wert 7 hier grün und alle darunter gelb bis rot gefärbt sind.



Abbildung 4: Variation der Parameter bei MtoBscan.m

Anhand der Abbildung kann man beispielhaft erkennen, wie wichtig es bei diesem Projekt war die richtigen Parameter herauszufinden.

Anhand von Parametern wie „LBoundPrewitt“ kann man erkennen, wie es nicht unbedingt nur einen richtigen Bereich für einen bestimmten Parameter gibt. Hier werden gute Ergebnisse beim aktuellen Wert erzielt, variiert man diesen um 1 Punkt nach oben oder unten ist dieser weit daneben, variiert man diesen um 9 Punkte nach oben liefert dieser aber wieder bessere Werte.

## Matlab vs. Python

Zu Projektstart hatten die Autoren die Wahl das Projekt in Python oder Matlab zu implementieren. Da Matlab von den Tutoren unterstützt wird und Python nicht, es einen Einführungskurs zu Matlab gab und die Autoren außerdem teilweise fortgeschrittenes Matlab-Wissen hatten, wurde sich dafür entschieden.

Matlab hat allerdings ein entscheidendes Manko, was nun nach Abschluss der PBL zum Tragen kommt: Sollte sich nun nach Abschluss des Studiums einer der Gruppenteilnehmer dazu entscheiden sich das Projekt nochmal anzugucken, könnte er das nur nach Erwerb einer Matlab Lizenz, da Matlab proprietäre Software ist. Aus Sicht der Free Software Foundation ist dies auch zentraler Zweck von proprietärer Software, welche zu Schul- bzw. Studienzwecken kostenfrei ist.[[1]](#footnote-2)Es stellt sich die Frage inwiefern der erstellte Code tatsächlich noch das eigene geistige Eigentum ist, wenn dieser nach Abschluss des Studiums nicht mehr kompiliert werden kann. Bei Python wäre das dadurch, dass Python Open Source und Free Software ist anders.

# Ausblick

Das vorgestellte Programm bietet eine gute Grundlage für eine Weiterentwicklung und Automatisierung der Module. Man könnte mit Verfahren wie zum Beispiel einem adaptiven Threshold die Anzahl der Parameter verkleinern. Die Parameter sollten aus einer Konfigurations-Datei gelesen werden, damit wären müsste das Programm nicht nach jeder Parameter-Anpassung neu kompiliert werden. Die Eingabe der Parameter wäre damit außerdem übersichtlicher und das Abspeichern als Metadaten einfacher. Die Durchmesserbestimmung ist in den ersten und letzten Spalten sehr rauschanfällig. Dies kann verbessert werden, indem man für das Filtern vor die ersten Spalten, die letzten und an die letzten Spalten die ersten hängt. Nach dem Filtern müssen die angehängten Spalten wieder abgeschnitten werden und man erhält eine komplett gefilterte Messung. Beim Pulsieren wird je nach Threshhold der äußere Durchmesser ignoriert oder es wird kein Durchmesser gefunden. Hierfür muss ein neuer Algorithmus mit einem Kantenfilter, der mehr als nur eine Kante findet, implementiert werden.

**Autoren:**

Problem – Alexander Strachatov

Material und Methoden – Felix Brilej

Ergebnisse: Vorverarbeitung und Verarbeitung – Alexandra Eberenz

Ergebnisse: GUI – Felix Brilej

Diskussion – Felix Brilej

Ausblick – Malte Kraienhorst

1. <https://www.gnu.org/education/education.html> [↑](#footnote-ref-2)