Bachelorthesis

Routenminimierung bei der Erfassung mehrdimensionaler Daten am Nahfeldmikroskop für die Nutzbarmachung von Compressed Sensing

---

vorgelegt am 4. August 2020

•

Fachbereich Duales Studium Wirtschaft / Technik
Hochschule für Wirtschaft und Recht Berlin

Name:	Oleg Kornelsen
Ausbildungsbetrieb:	Helmholtz-Zentrum Berlin
Studienbereich:	Technik
Fachrichtung:	Informatik
Studiengang:	Informatik
Studienjahrgang:	2017
Erstgutachter:	Dr. Bernd Kästner
Zweitgutachter:	Prof. Dr. Gerrit Kalkbrenner

KURZFASSUNG

Bachelorthesis im Studiengang Informatik
Hochschule für Wirtschaft und Recht Berlin
Fachbereich Duales Studium Technik
Jahrgang 2017

Oleg Kornelsen

In der Forschung ist es sehr wichtig, diverse Messabläufe zu automatisieren. Auch die Geschwindigkeit des Ablaufs und die Qualität der Messung ist von entscheidender Bedeutung. Da eine vollständige Messung am Nahfeldmikroskop sehr lange dauern kann, soll an der Physikalisch-Technischen Bundesanstalt (PTB) eine Routenminimierung bei der Erfassung der Daten angewendet werden.

Diese Bachelorarbeit beschreibt die Anwendung und den Vergleich der Algorithmen an ausgedünnten Daten, um den kürzesten Pfad in einem 3D-Raum zu finden. Außerdem wird der Entwurf eines Systems behandelt, das entsprechend dem Pfad die Daten an einem Infrarot-Nahfeldmikroskop der Firma neaspec GmbH aufnehmen kann. Ziel ist es nachzuweisen, dass die einzelnen zufälligen Messpunkte mit dem Nahfeldmikroskop entsprechend dem Pfad aufgenommen werden können.

Erstgutachter	Dr. Bernd Kästner
Zweitgutachter	Prof. Dr. Gerrit Kalkbrenner

Berlin, 4. August 2020.

ABSTRACT

Bachelorthesis for Computer Science
Berlin School of Economics and Law
Department of Cooperative Studies
Class 2017

Oleg Kornelsen

Route minimization in the acquisition of multidimensional data on the near field microscope for the utilization of compressed sensing

In research, it is very important to automate the various measurement processes. The speed of the process and the quality of the measurement is also very important. Since a complete measurement using a near-field microscope can take a long time, route minimization should be used at the Physikalisch-Technische Bundesanstalt (PTB) when collecting the data.

This bachelor thesis describes the application and comparison of the algorithms for the thinned out data in order to find the shortest path in a 3D space. In addition, the design of a system is dealt with, which can record the data according to the path on an infrared near-field microscope from neaspec GmbH. The goal is to demonstrate that the individual random measuring points can be recorded with the near field microscope according to the path.

Primary review	Dr. Bernd Kästner
Secondary review	Prof. Dr. Gerrit Kalkbrenner

Berlin, 4. August 2020.

Danksagung

Für die Möglichkeit Informatik studieren zu dürfen, möchte ich mich beim Helmholtz-Zentrum bedanken. Ich möchte auch Olaf-Peter Sauer, Anita Zahr, Marion Schröder, Lutz Rossa und Michael Helmig sehr herzlich danken.

Mein weiterer Dank gilt auch der Physikalisch-Technischen Bundesanstalt, die es mir ermöglicht, meine Bachelorarbeit durchzuführen. Besonders bedanke ich mich bei Dr. Bernd Kästner für die Betreuung und Zusammenarbeit in der Ausarbeitung meiner Bachelorthesis.

Auch danke ich Prof. Dr. Gerrit Kalkbrenner für die hilfreiche Unterstützung, die er geleistet hat.

Inhaltsverzeichnis

Kurzfassung **I**

Abstract **III**

Inhaltsverzeichnis **VIII**

Abbildungsverzeichnis **IX**

Glossar **XI**

Akronyme **XII**

1 Allgemeines **1**

1.1 Motivation 1

1.2 HZB - Helmholtz Zentrum Berlin 1

1.3 PTB - Physikalisch-Technische Bundesanstalt 2

1.4 Einordnung 2

2 Einführung **5**

2.1 Grundlagen 5

2.1.1 neaSNOM Mikroskop 5

2.1.2 Compressed Sensing 7

2.2 Zielsetzung 8

2.3 Anforderungen 10

3 Routenoptimierung **13**

3.1 Daten 13

3.1.1 Relevante Daten 14

3.1.2 Datenreduzierurg 15

3.2 MatLab-Skript-Anpassung 16

3.3 Bewegungszeit 17

3.4 Statische Routen 21

3.5 Optimierte Routen 22

3.5.1 TSP - Traveling Salesman Problem 22

3.5.2 TSP-Mathematica 24

3.5.3 Suche nach einem nahe gelegenen Punkt 25

3.6 Routenvergleich 26

4 Mikroskopsteuerung **29**
4.1 Verbindung und Ablauf 29
4.1.1 Datenaufnahme 29
4.1.2 Progressdarstellung 30
4.2 Datentransformierung 30

5 Implementierung **33**
5.1 Programmiersprache 33
5.2 Programmstruktur 36

6 Ergebnis **39**
6.1 Zusammenfassung 39
6.2 Test 39
6.3 Fazit 42
6.4 Ausblick 42

Webquellen **43**

Literaturverzeichnis **44**

Ehrenwörtliche Erklärung **47**

Abbildungsverzeichnis

1 Teilziele der Thesenprüfung 3

2 NeaSNOM Mikroskop 6
3 Cantilever mit der Spitze 6
4 Modell der Rasterkraftmikroskopie und Spekrtoskopie 7
5 Compressed Sensing 8

6 Vollständige Messung 13
7 3D-Matrix 14
8 Intensität des Spektrums 15
9 Zufällige Stellen 16
10 Rekonstruierte Bilder 18
11 Z-Route 21
12 XY-Route 22
13 Optimierte Route 25
14 Vergleich der Routen - Tabelle 26
15 Vergleich der Routen - Diagramm 27

16 Vor- und Nachteile der Programmiersprachen 33

17 Verlauf der Route in 3D-Raum 40
18 Gemessene und rekonstruierte Daten 41
19 Messergebnis aus rekonstruierte Daten 41

Hinweis

Alle Abbildungen bei denen keine explizite Quelle angegeben ist, sind Schöpfungen des Autors und als solche nicht kennzeichnungspflichtig.

Glossar

Atomic Force Microscopy Die Atomic Force Microscopy (deutsch Rasterkraftmikroskopie) dient der mechanischen Abtastung von Proben. Atomare Wechselwirkungen ändern dabei die Ausrichtung, Amplitude und Frequenz eines Hebelarms (englisch Cantilever). Die Änderung wird mithilfe von Licht von einer 4-Segment Diode aufgenommen. Aus den gemessenen Wechselwirkungen kann die Oberfläche der Probe berechnet werden.

Metrology Light Source MLS ist ein Elektronenspeicherring der Physikalisch-Technischen Bundesanstalt am Forschungsstandort Adlershof, Berlin Deutschland.

NP-Problem In der Informatik bezeichnet NP (für nichtdeterministisch polynomielle Zeit) eine fundamentale Komplexitätsklasse aus dem Bereich der Komplexitätstheorie

Software Development Kit Ein SDK ist eine Sammlung von Methoden, Funktionen und Bibliotheken, die bei der Entwicklung von Programmen helfen sollen.

Akronyme

AFM Atomic Force Microscopy

BESSY Berliner Elektronenspeicherring-Gesellschaft für Synchrotronstrahlung

BMWi Bundesministeriums für Wirtschaft und Energie

CS Compressed Sensing

F-A funktionale Anforderungen

GUI Graphical User Interface

HZB Helmholtz-Zentrum Berlin für Materialien und Energie

MLS Metrology Light Source

N-A nicht-funktionale Anforderungen

PTB Physikalisch-Technische Bundesanstalt

SDK Software Development Kit

THz Terahertz

TSP Traveling Salesman Problem

1 Allgemeines

Die vorliegende Arbeit ist das Ergebnis zu der Bachelorthesis von Oleg Kornelsen. Die Arbeit wurde innerhalb von zehn Wochen an der Physikalisch-Technischen Bundesanstalt (PTB) im Auftrag des Helmholtz-Zentrum Berlin für Materialien und Energie (HZB) durchgeführt.

1.1 Motivation

Heutzutage werden durch Systeme mit ständig wachsender Komplexität und hochmoderner Hardware auch größere Datenmengen gewonnen. Inzwischen wird es immer wichtiger, dass hohe Datenaufkommen zu verarbeiten, um sinnvolle Informationen gewinnen zu können. Da eine Messung sehr lange dauern kann, werden Messstrategien benötigt, um ohne signifikante Informationsverluste die Messung in kürzere Zeit durchzuführen. Entsprechende komprimierte Messverfahren sind Gegenstand aktueller Forschung. Hierfür wird ein Softwareansatz benötigt, um die komprimierten Messabläufe umzusetzen. So haben die Wissenschaftler die Möglichkeit, neuartige Kompressionsstrategien experimentell zu verifizieren.

1.2 HZB - Helmholtz Zentrum Berlin

Helmholtz ist ein eingetragener Verein, dessen Mitglieder rechtlich selbstständige Zentren sind. Diese Zentren mit ihren vielfach vertretenen interdisziplinär und international agierenden Experten sind der wissenschaftliche Motor der Gemeinschaft. Das HZB ist eines von mehreren naturwissenschaftlichen Forschungszentren [Helmholtz]. Gegründet wurde das HZB unter dem Namen Institut für Kernforschung Berlin im Winter 1956/57 und am 14. März 1959 umgetauft zum Hahn-Meitner-Institut für Kernforschung Berlin. Seit der Verschmelzung mit der Berliner Elektronenspeicherring-Gesellschaft für Synchrotronstrahlung (BESSY) am 11. November 2008 trägt das Zentrum seinen aktuellen Namen. Mit ca. 1100 Mitarbeitern ist das HZB heute eines der größten außeruniversitären Forschungszentren in Berlin. Das HZB forscht in der Hauptstadt der Bundesrepublik Deutschland an zwei Standorten, in Wannsee und Adlershof.

Für diese Forschung betreibt das HZB moderne Laboratorien, Infrastrukturen und Instrumente. Viele stehen nicht nur den Wissenschaftlern des HZB zur Verfügung, sondern zugleich einer nationalen und internationalen Forschungsgemeinschaft.

Die am meisten genutzte Infrastruktur ist der Elektronenspeicherring BESSY II, ein Teilchenbeschleuniger, der eine spezielle Art von Röntgenstrahlung liefert. Sie wird als Synchrotronstrahlung bezeichnet. Das HZB kooperiert mit viele anderen Hochschulen und Forschungszentren in Deutschland und auch international. Ein Teil dieser Gemeinschaft ist die PTB.

1.3 PTB - Physikalisch-Technische Bundesanstalt

Die Physikalisch-Technische Bundesanstalt (PTB), das nationale Metrologieinstitut, ist eine wissenschaftlich-technische Bundesoberbehörde im Geschäftsbereich des Bundesministeriums für Wirtschaft und Energie. Als das wichtigste nationale Metrologieinstitut Deutschlands ist die PTB oberste Instanz bei allen Fragen des richtigen und zuverlässigen Messens. Sie ist die technische Oberbehörde des Bundesministeriums für Wirtschaft und Energie (BMWi) und beschäftigt insgesamt rund 1900 Mitarbeiter [PTB]. Weiterhin ist die PTB, wie es auch in vielen anderen größeren Unternehmen üblich ist, in mehreren Abteilungen gegliedert. Bezogen auf das durchzuführende Projekt, wurde dieses in der Abteilung 7 „Temperatur und Synchrotronstrahlung“ durchgeführt.

Die PTB besitzt im Willi-Wien-Laboratorium einen eigenen, kleinen Speicherring, der als Metrology Light Source (MLS) beizeichnet wird [MLS]. Die Synchrotronstrahlung dieses Speicherrings wird für die Nahfeldmikroskopie verwendet, welche auch ein Teil dieses Projektes ist.

1.4 Einordnung

Mittels Terahertz (THz)-Nahfeldmikroskopie werden über ein 2D Raster verteilt THz-Spektren aufgenommen. Diese Methode basiert auf einem Rasterkraftmikroskop und erlaubt dadurch eine räumliche Auflösung im Nanometerbereich, trotz der Verwendung von THz-Strahlung mit um ein Vielfaches größeren Wellenlängen. Die systematische Aufnahme von Spektren an jedem einzelnen Punkt ist jedoch sehr zeitaufwendig. In Zukunft soll eine erhebliche Verringerung der Messzeit durch die

Verwendung von Compressed-Sensing (s. 2.1.2) Methoden erreicht werden. Hierzu muss ein geeigneter Pfad in Raum- und Spektrometer-Koordinaten gefunden werden. Die Umsetzung erfolgt an einem kommerziellen Nahfeldmikroskop des Modells neaSNOM (Neaspec GmbH, Deutschland). Zum Nahfeldmikroskop existiert ein Software Development Kit (SDK), welches in die Programmiersprache Python eingebunden werden kann. Das Ziel der Bachelorarbeit besteht darin, mithilfe der Verwendung des SDK, in Python ein Programm zur Ansteuerung des Nahfeldmikroskops mit einer minimierten Route zu erstellen und an einem konkreten Probensystem zu testen. Folgende These wurde für diese Bachelorarbeit aufgestellt:

> *Es ist möglich, anhand eines vorgegebenen, kompletten Datensatzes einen für nachträgliche Rekonstruktion geeigneter reduzierten Datensatz zu erzeugen, eine minimierte Route zu finden sowie die Vorbereitung zur Anwendung von Compressed Sensing durchzuführen. Unter Anderem ist es auch machbar die einzelnen zufälligen Punkte mit dem Nahfeldmikroskop entsprechend der Route aufzunehmen.*

Um diese These zu überprüfen, wird zunächst ein Ablaufschema formuliert. Dieses Schema beinhaltet die einzelnen Schritte, die als Teilziele betrachtet werden. Damit ist es einfacher das Endziel zu erreichen und die These zu bestätigen oder anderenfalls zu erfahren, an welcher Stelle die Machbarkeit nicht mehr umzusetzen ist. (s. Abb. 1)

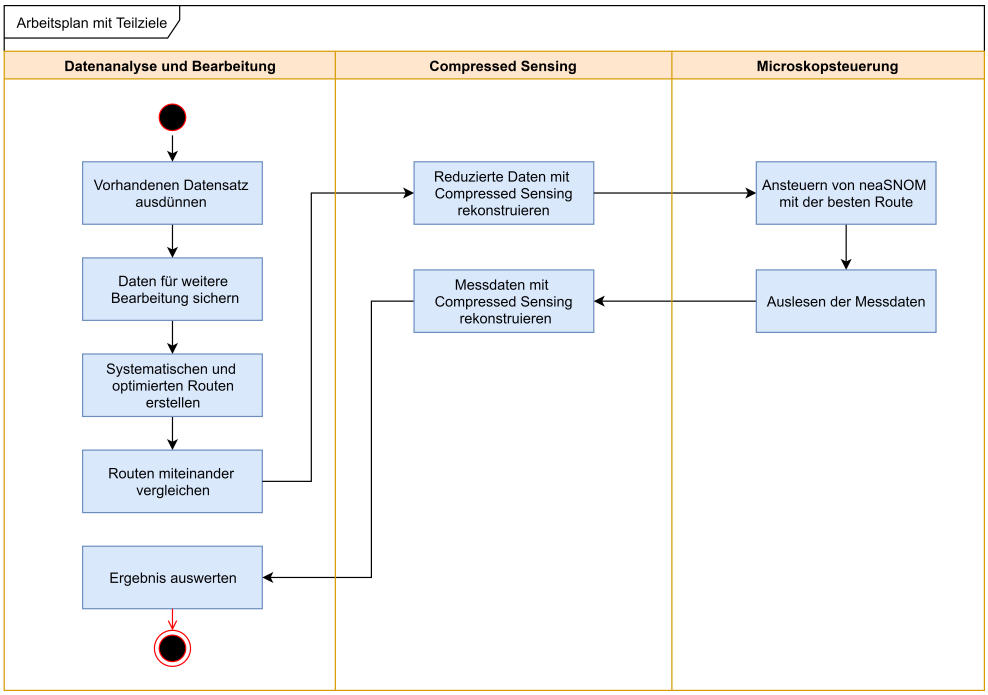


Abbildung 1: Teilziele der Thesenprüfung

2 Einführung

In diesem Abschnitt werden die Voraussetzungen für das Projekt beschrieben. Es wird grob erklärt, wie das Nahfeldmikroskop aufgebaut ist und wie es funktioniert. Weiterhin wird auch umrissen, was Compressed Sensing ist. Aus der Zielsetzung, die vorher bereits formuliert wurde, werden Anforderungen bestimmt, die für die Sicherstellung der Arbeitsqualität dienen sollen. Am Ende dieses Kapitels sollen die Informationen für das Projekt vollständig dargestellt werden. Mit diesen Erkenntnissen kann mit dem Entwurf angefangen werden.

2.1 Grundlagen

Ziel dieses Projekts ist die Routenminimierung bei der Erfassung mehrdimensionaler Daten am Nahfeldmikroskop, welche für die Nutzbarmachung von Compressed Sensing dient. Daher es ist wichtig zu verstehen, wie die Daten am Nahfeldmikroskop aufgenommen werden. Außerdem soll damit aufgezeigt werden, wie groß die Gewichtung für die Lösung dieser Aufgabe durch Compressed Sensing ist.

2.1.1 neaSNOM Mikroskop

Die PTB betreibt am Standort Adlershof ein Infrarot-Nahfeld-Mikroskop. Dieses wird für das Projekt verwendet. Das Nahfeldmikroskop ist unter dem Namen neaSNOM bekannt. Das Gerät wurde von der Firma neaspec GmbH entwickelt. Laut Hersteller basierend auf der revolutionären Technologie von neaspec ist neaSNOM das einzige Mikroskop auf dem Markt, das Bildgebung und Spektroskopie im sichtbaren, infraroten und sogar Terahertz-Spektralbereich bei einer räumlichen Auflösung von nur $10nm$ ermöglicht [neaspec]. Das optische NeaSNOM-Scan-Nahfeldmikroskop basiert auf einer Rasterkraftmikroskopie (Atomic Force Microscopy (AFM)), das im intermittierenden Kontaktmodus betrieben wird. Auf der Abbildung 2 ist der Aufbau des Mikroskops dargestellt. Die von den Wissenschaftlern ausgewählten Proben werden auf dem Probetisch (Sample Positioner) befestigt. Gemessen werden die Proben mit einer sehr kleinen Messspitze, die an einem Hebelarm, der Cantilever genannt wird, befestigt ist, welcher wiederum an einem Prüfkopf (Probing Head) arretiert wird (s. Abb. 3).

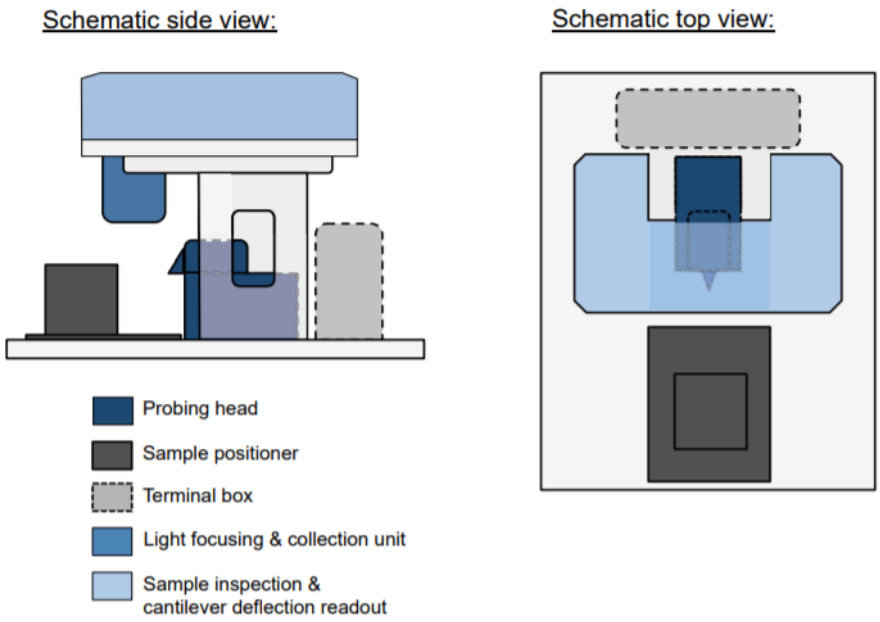
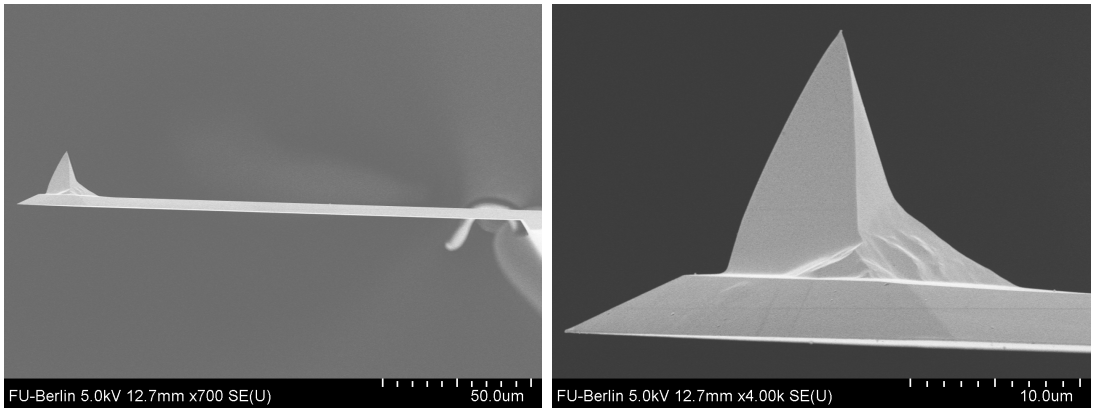


Abbildung 2: NeaSNOM Mikroskop

Die Messspitze spielt während der Messungen die wichtigste Rolle. Diese besteht meistens aus Silizium (Si) und ist ca. $10 - 15\mu m$ lang. Für Spektrometrie ist die Spitze noch zusätzlich mit einer Schicht aus Gold belegt.

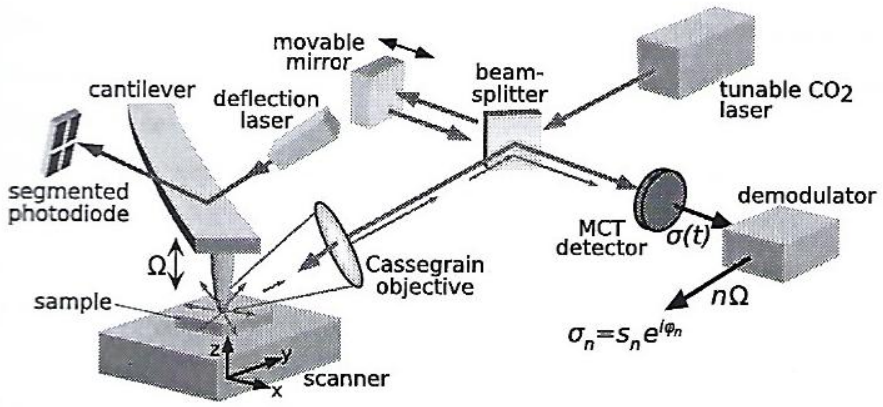


Quelle: Piotr Patoka, Freie Universität Berlin

Abbildung 3: Cantilever mit der Spitze

Wie schon erwähnt wurde, arbeitet NeaSNOM in einem intermittierenden Kontaktmodus. In diesem Modus wird der Cantilever mit Schwingungen angeregt. Die Schwingungen liegen in der Nähe der Resonanz mit einer Amplitude in der Größenordnung von 10 bis $100nm$. Der Cantilever wird an die Oberfläche gebracht, so dass sich in der unteren Hälfte der Schwingungen die Oberfläche der Probe berührt. Beim Scannen einer Probe wird eine Änderung der Amplitude und Phase der Cantilever-Schwingungen aufgezeichnet. Das Erhalten von AFM-Bildern des Oberflächenreliefs ist mit der Registrierung kleiner Biegungen des Cantilevers abgeschlossen. In der Rasterkraftmikroskopie werden zu diesem Zweck häufig optische Verfah-

ren eingesetzt. Das optische AFM-System wird so eingestellt, dass die Strahlung des Halbleiterlasers auf den Cantilever fokussiert wird und der reflektierte Strahl in die Mitte des lichtempfindlichen Bereichs des Fotodetektors fällt. Als positionsempfindliche Fotodetektoren werden vierteilige Halbleiterphotodioden verwendet (s. Abb. 4).



Quelle: neaSNOM

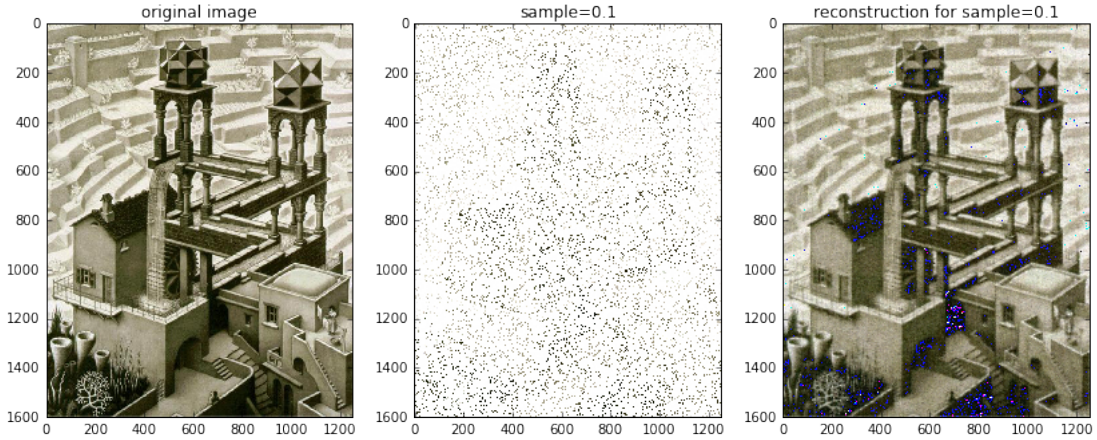
Abbildung 4: Modell der Rasterkraftmikroskopie und Spekrtoskopie

Zusätzlich kann auf dem NeaSNOM eine Spektroskopie durchgeführt werden. Mit der Verwendung von dem Interferometer werden die Spektren an der Position der Messspitze aufgenommen. Damit es möglich ist, wird Synchrotronstrahlung aus dem Speicherring der MLS verwendet. Die Strahlung hat eine hohe Intensität und ein kontinuierliches, breites Spektrum [Hofmann]. Für das Nahfeldmikroskop wird aus der Synchrotronstralung mit Hilfe von diversen optischen Elementen ein Infrarotlicht genutzt. Das Infrarotlicht pariert vor der Verwendung noch durch einen Filter, damit der Wellenlängenbereich zwischen 4 μm und 20 μm liegt.

2.1.2 Compressed Sensing

Compressed Sensing (CS) (deutsch: komprimiertes Abtasten), auch bekannt als *Compressive Sensing* oder *Compressive Sampling*, ist eine neue mathematische Theorie der Datenerfassung und Rekonstruktion, die in der Signal- und Bildverarbeitung Anwendung findet [Davenport]. *CS* ist ein aufregendes, schnell wachsendes Gebiet, das in den Bereichen Elektrotechnik, angewandte Mathematik, Statistik und Informatik große Aufmerksamkeit auf sich gezogen hat. *CS* bietet einen Rahmen für die gleichzeitige Erfassung und Komprimierung von endlichdimensionalen Vektoren, der auf einer linearen Dimensionsreduktion beruht. Es ist ziemlich überraschend, dass aus wenigen hochdimensionalen Daten von unvollständigen Messungen unter Verwendung effizienter Algorithmen, ein vollständiges Bild der kompletten Messung wiederhergestellt werden kann [Eldar].

Für die Verwendung von *CS* müssen die Daten aus dem ganzen Raum der Messung, in gleichmäßiger Verteilung, vorkommen. Idealerweise sollten die Signale zudem auch zufällige Stellen sein. So lässt sich das nachfolgende Bild aus 10 % der Information fast originalgetreu wiederherstellen. Das rekonstruierte Bild dient als beispielhafte Verdeutlichung. (s. Abb. 5).



Quelle: http://www.pyrunner.com/weblog/2016/05/26/compressed-sensing-python/

Abbildung 5: Compressed Sensing

Für die Rekonstruierung der Daten gibt es heute schon relativ viele Methoden und Algorithmen, die für bestimmte fälle eigene Vor- und Nachteile haben. Für diesen Versuch (s. Abb. 5) wurde zum Beispiel der Orthant-wise limited-memory quasi-Newton (OWL-QN) - Algorithmus verwendet.

An der Stelle sei darauf hingewiesen, dass das Thema *CS* verdeutlichen soll, woran der Inhalt dieses Projektes später anknüpft. Die Ausführung weiterer, komplexer Schritte dazu bedürfen deshalb keiner weiteren Erläuterung.

Es steht ein von Wissenschaftlern erstelltes MatLab-Script zur Verfügung, welches 3D-Modelle aus den gemessenen reduzierten Daten rekonstruieren kann.

2.2 Zielsetzung

Das Ziel dieses Projekts ist es, die optimale Route durch die reduzierten zufällig ausgewählte Messstellen für das Mikroskop, zu ermitteln. Dafür soll ein kompletter Messablauf geplant und erstellt werden, was im Folgenden dargelegt wird.

Ein Messablauf kann nicht ohne den Einsatz eines Nutzers durchgeführt werden. Weiterhin muss mindestens eine Probe auf dem Probentisch des neaSNOMs befes-

tigt werden. Der nächste Schritt ist die Probe an die Messspitze anzufahren und somit zu lokalisieren. Da die Probe relativ groß sein kann, muss vom Nutzer auch eine ausreichend groß dimensionierte Testumgebung für die Machbarkeit der Messung gewählt werden. Die gesamte Durchführung erfolgt technisch durch das Graphical User Interface (GUI) des NeaSNOMs, daher ist dies eine unabdingbare Voraussetzung. Weitere Schritte können vom Programm automatisiert übernommen werden. Die Kommunikation über NeaSNOM wurde bereits durch den Bachelor-Studenten Sebastian Sachse vorprogrammiert und kann für das Projekt verwendet werden. Auch die Annäherung an die Messspitze ist fertig implementiert und findet Anwendung in diesem Projekt. Das Programm hat außerdem die Aufgabe, die Daten entsprechend der Route zu sammeln und zu speichern. Abschließend sollen die ermittelten Daten so angeordnet und gehalten werden, dass diese als Grundlage vom zu entwickelnden MatLab-Script verwendbar sind.

2.3 Anforderungen

Aus der Zielsetzung heraus und den damit verbundenen Gesprächen mit den Stakeholdern ergaben sich Voraussetzungen und auch Anforderungen an das System, welches im Laufe dieses Projekts entwickelt werden soll. Im folgenden werden diese Anforderungen entsprechend der Schablone von Rupp formuliert [Rupp]. Die funktionalen Anforderungen lauten:

F-A01: **Filterung** Das System soll in der Lage sein, nur relevante Werte aufzunehmen.

F-A02: **Systematische Routen** Das System soll mindestens eine systematische Route haben.

F-A03: **Optimierte Routen** Das System soll mindestens eine optimierte Route zur Verfügung stellen können.

F-A04: **Zeitberechnung** Das System soll die Zeiten für die Routen anhand der Informationen aus der Dokumentation zum NeaSNOM berechnen.

F-A05: **Analyse** Die Routen sollen miteinander verglichen werden.

F-A06: **Kommunikation** Die Software soll mit dem neaSNOM Mikroskop kommunizieren können.

F-A07: **Probenannäherung** Das System muss in der Lage sein, die Messspitze mit der Probe in Kontakt zu bringen.

F-A08: **Scanning** Die Software muss entsprechend der Route, die Information anhand der ermittelten Punkte aufnehmen.

F-A09: **Progress** Das System soll den Verarbeitungsfortschritt des Anforderungspunktes F-A08, Scanning, liefern. Der Fortschritt kann jeweils anteilig in Prozent oder in Anzahl der Punkte am Gesamtergebnis dargestellt werden.

F-A10: **Datenspeicherung** Das System soll die gesammelten Daten speichern.

F-A11: **Transformierung** Das System muss aus die gespeicherten Daten so verarbeiten, dass diese im Matlab-Script lesbar übernommen werden können.

Die nicht-funktionalen Anforderungen an das System werden im Folgenden beschrieben:

N-A01: **Benutzbarkeit** Das System soll einfach und intuitiv bedienbar sein.

N-A02: **Modularität** Das System soll modular in Routenberechnung und Mikroskopsteuerung aufgeteilt und abgegrenzt werden.

N-A03: **Zufälligkeit** Die Route soll immer aus zufällig ermittelten Punkten generiert werden.

N-A04: **Stabilität** Das System muss stabil laufen und beim Auftreten von Fehlern die belegten Ressourcen wieder freigeben können.

N-A05: **Parametrisierung** Das System soll dem Nutzer die Möglichkeit bieten, leicht die Parameter verändern zu können.

N-A06: **Berechnungszeit** Das System muss essentiell für die Berechnung der optimierten Route eine adäquate Zeit verbrauchen.

N-A07: **Erweiterbarkeit** Die Software soll zukünftig auch jederzeit erweiterbar sein.

3 Routenoptimierung

Dieses Kapitel beschreibt die Ansätze zur Routenoptimierung. Es wird dargestellt, wie aus den Daten relevante Informationen gewonnen werden. Danach wird gezeigt, wie die relevanten Informationen noch weiter reduziert werden können, um eine Optimierung anzuwenden. Danach werden mit weiteren Schritten die vorliegenden statischen und optimierten Routen betrachtet und miteinander verglichen. Zum Schluss wird in diesem Kapitel bestimmt, in welcher Form die Daten an das MatLab-SKript übergeben werden.

3.1 Daten

Um mit den Daten operieren zu können, muss erst verstanden werden in welcher Struktur die Daten dargestellt sind. Dafür steht ein bereits aufgenommener Referenzdatensatz zur Verfügung, der als Voralge dient(s. Abb. 6). Der Datensatz repräsentiert einen sternförmigen Abdruck auf der Oberfläche, die aus *Hexagonal Silicon Carbide* (6H-SiC) besteht.

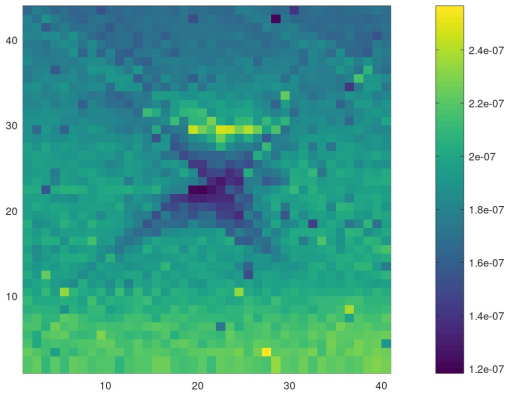


Abbildung 6: Vollständige Messung

Der Datensatz ist als 3D-Matrix (s. Abb. 7) aufgebaut. Die Bezeichnungen auf dem Bild sind im Folgenden zu übersetzen:

„X“: Ist die Position der X-Achse vom NeaSNOM

„Y“: Ist die Position der Y-Achse von NeaSNOM

„Z“: Ist die Position des Interferometers von NeaSNOM

Jedes Kästchen beinhaltet einen Wert, welcher ein gemessenes Spektrum an einem bestimmten Punkt repräsentiert. Die Werte sind als komplexe Zahlen abgelegt. Die Parameter des Datensatzes sind 41x44x1024 (X x Y x Z). Das entspricht 1.847.296 Werten insgesamt. Die Werte wurden analysiert, um die relevanten Daten daraus zu entnehmen.

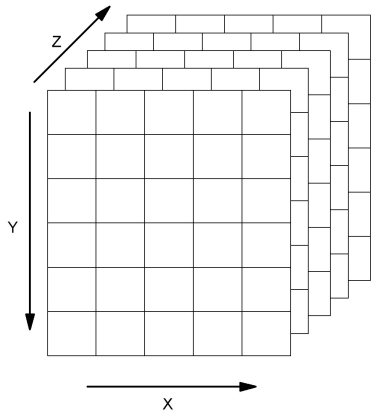


Abbildung 7: 3D-Matrix

3.1.1 Relevante Daten

Bei der Spektroskopie sind die Daten interessant, bei deren der Wert sich zu den anderen Werten stark unterscheidet und die Schwingungen stärker zu sehen sind. Um das visuell zu sehen, sollte aus der 3D-Matrix ein Vektor gezogen werden, der an einem Punkt das ganze Spektrum zeigen kann. Aus 1024 Werten konnte ein Diagramm (s. Abb. 8) erstellt werden. In dem Diagramm lässt sich visuell erkennen, in welchem Raum das Spektrum wichtig wird. Für Wissenschaftler sind alle Werte relevant, die im Raum mit Amplitude ab 1/10 von der maximalen Amplitude liegen [Kästner]. Um die genaue Grenze anzulegen, sollte im Vektor zuerst die maximale Amplitude bestimmt werden. Anhand dieser Information konnten die ersten Werte, die größer als 1/10 des Maximums sind, von beiden Seiten gefunden werden. Auf dem Diagramm sind die Grenzen an den grünen Linien zu erkennen.

Die erste Grenze ist an der Position 562 und die Zweite an der Position 882. Die Differenz zwischen diesen zwei Stellen liegt bei 320 Punkten. An anderen XY-Positionen haben die Vektoren etwas andere Grenzen, daher müssen für die ganze 3D-Matrix eindeutige Grenzwerte angelegt werden. So entstand eine untere Grenze an der Stelle 600 und eine Obere an der Stelle 900. Zwischen diesen zwei Werten liegen die wich-

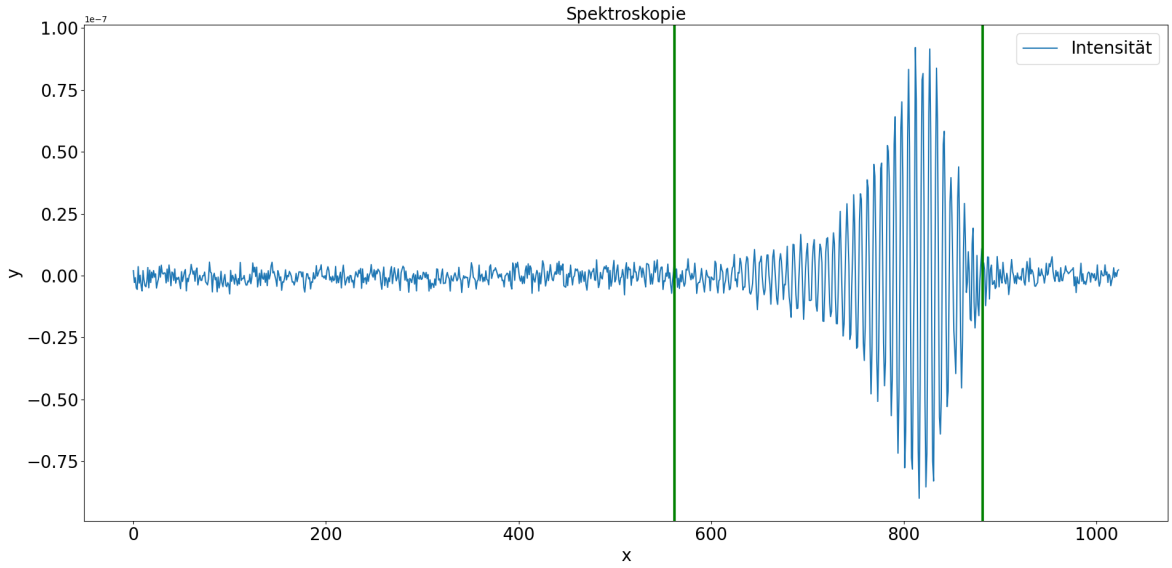


Abbildung 8: Intensität des Spektrums

tigsten Informationen, die von Wissenschaftlern weiter untersucht werden können. Die restlichen Informationen gelten als irrelevant.

Auf diese Weise kann theoretisch die Messdauer reduziert werden, wenn eine Probe nur in eingegrenzten bestimmten Raum gemessen wird. Nach nach diesem Vorgehen kommen insgesamt 41x44x300 = 541.200 Punkte zustande. Für die Bachelorthesen reicht dieser Schritt nicht aus, sodass die Daten noch weiter reduziert werden müssen.

3.1.2 Datenreduzierurg

Das Ziel in dem Abschnitt ist, die verbleibenden relevanten Daten so zu reduzieren, sodass die gesamten Informationen nicht verloren gehen oder rekonstruiert werden können. Für das Rekonstruieren der Daten steht schon das bereits erwähnte Werkzeug der Mathematiker *CS* zur Verfügung.

Laut *CS* reichen für das Rekonstruieren 10% der ursprünglichen Daten aus (s. 2.1.2). Die Daten sollen aber zufällig verteilt werden, um ein effizientes Ergebnis zu bekommen.

Die zufälligen Werte konnten aus jedem Vektor gezogen werden. In dem Diagramm (s. Abb. 9) sind die Stellen mit roten Punkten markiert. Von 300 möglichen Stellen werden nur 30 genommen, was genau 10% entspricht. So bleiben für die weitere Verwendung 54.120 Punkte im 3D-Raum.

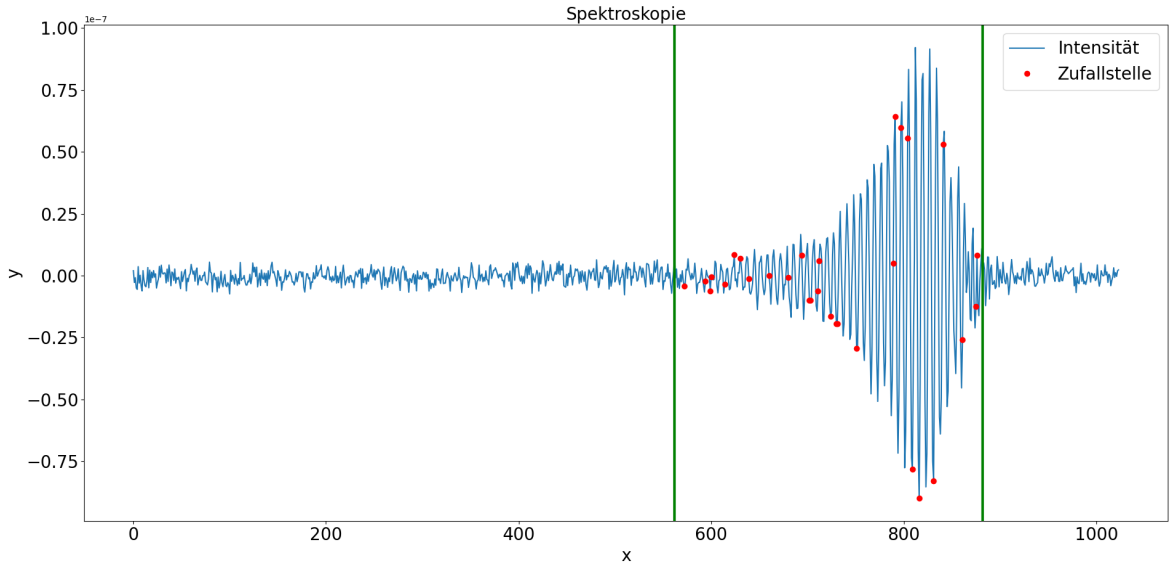


Abbildung 9: Zufällige Stellen

Für die weitere Analyse wurden später auch andere Mengen der Punkte hinzugezogen. Dies ermöglicht dem Nutzer später leichter eine Entscheidung zu treffen, wie stark die Informationen in seiner Messung reduziert werden sollen. Entschieden wird dann zwischen der Messzeit und dem Qualität der Messung.

3.2 MatLab-Skript-Anpassung

Die reduzierten Daten sollten auf Effizienz geprüft werden. Die beste Möglichkeit war dabei die Daten mit dem MatLab-Skript abzuarbeiten. Anhand der Ergebnisse wird die Entscheidung, wie groß die Anzahl der Messpunkte sein muss, einfacher fallen. Damit das Einarbeiten des MatLab-Skripts möglich ist, wurde das bestehende Programm analysiert. Das Ziel war die Bestimmung des Datenformats, welches an das Skript übergeben werden soll.

Nach der Einarbeitung in die Programmiersprache MatLab und die grobe Analyse des Skriptes wurde festgestellt, dass das Programm am Eingang zwei Arrays bekommen soll. Die Arrays sind abhängig voneinander und müssen immer zusammen übergeben werden.

Ein Array beinhaltet die Werte zu den Punkten, die zufällig gewählt wurden. Wenn an einer XY-Position 30 Punkte gewählt werden sollen, dann stehen an dieser Position 30 Werte. Die Werte müssen als komplexe Zahlen dargestellt werden und dürfen in diesem Array unsortiert sein. Die Positionen, wo die Werte stehen sollen sind auch relevant und dafür dient das zweite Array. Im zweiten Array stehen an

jeder XY-Stelle die Positionen für die Werte, an deren das Spektrum aufgenommen wurde. Die Indizes aus beiden Arrays müssen übereinstimmen für den korrekten Skriptablauf.

Für ein besseres Verständnis sind weitere Auszüge von beiden Arrays dargestellt. Als Beispiel wurden die ersten 5 Werte an der Position x=1 und y=1 ausgewählt:

„Data“: [9.40917575e-09+2.30219792e-08j
-1.59237307e-08-4.04732425e-08j
-1.65188764e-08-3.94513170e-08j
1.22786308e-08+2.42531795e-08j
1.15197758e-08+2.61079644e-08j]

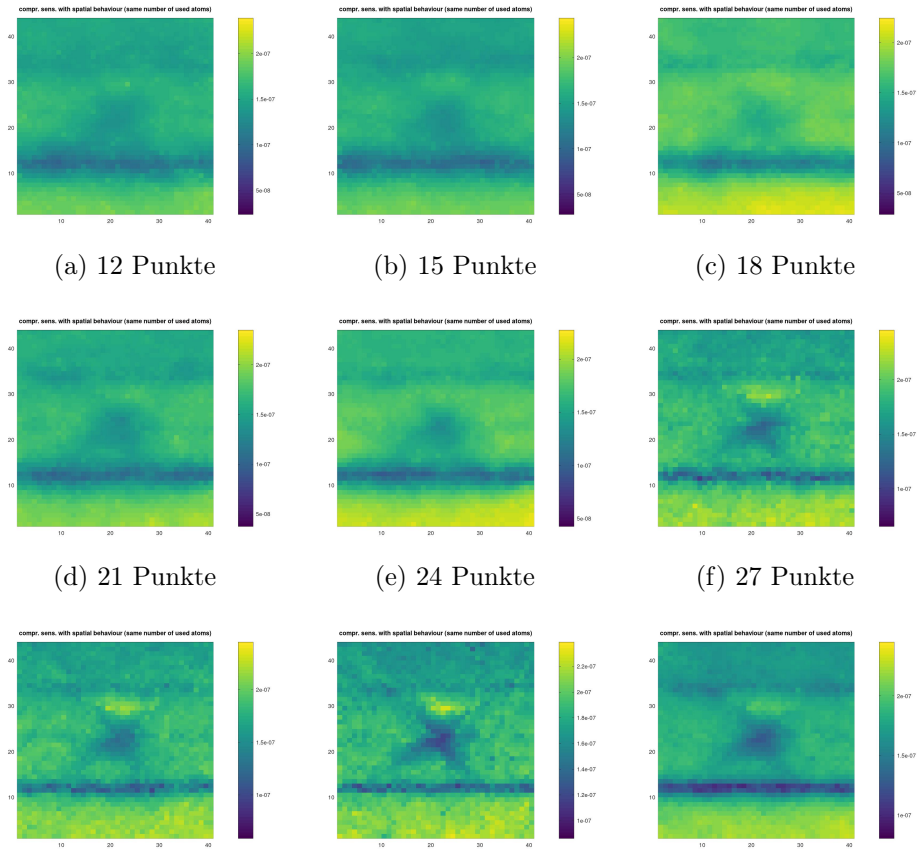
„Position“: [80 131 132 155 162]

Nach der Bestimmung der Datenübergabe wurden verschiedene Datensätze mit einer verschiedenen Anzahl von Punkten abgearbeitet. Da die Berechnung eine lange Zeit dauert, wurden nicht alle möglichen Messungen durchgeführt. Aus den gelieferten Ergebnissen kann visuell erkannt werden, wie die Qualität der Bilder mit steigender Punkteanzahl besser wird. Auf den Bildern (s. Abb. 10), die aus 12, 15, 18, 21 und 24 zufälligen Punkte von 300 Punkten pro Vektor rekonstruiert wurden, ist die Probe zu erkennen, d.h. der Abdruck ist in seiner Stern-Form leicht erkennbar. Die Qualität der Bilder ist sehr schlecht. Erst ab 27 Punkten pro Vektor zeigt sich eine deutlich bessere Qualität. Somit kann die Aussage der Wissenschaftler, über die Möglichkeit aus 10% der Informationen ein Bild zu rekonstruieren, bestätigt werden. Nach der Analyse des Skriptes ist ein gutes Resultat schon mit 9% der Informationen erreicht.

3.3 Bewegungszeit

Für eine Messung ist die Zeit wichtig, die gebraucht wird, um die wichtigen Informationen der Probe zu sammeln. Ein einzelner Messpunkt wird relativ schnell gemessen. Eine komplette Messung kann mehrere tausend und sogar noch mehr Messpunkte enthalten. Allerdings steigt dadurch die Messzeit deutlich, was vermieden werden soll.

Die Messpunkte werden nacheinander gemessen, da keine Möglichkeit besteht die Messungen parallel durchzuführen. Der Ablauf der Messungen ist wie folgt: Eine Probe ist auf dem Probentisch befestigt. Die Probe ist meistens nicht größer als



(a) 12 Punkte (b) 15 Punkte (c) 18 Punkte

(d) 21 Punkte (e) 24 Punkte (f) 27 Punkte

(g) 30 Punkte (h) 45 Punkte (i) 60 Punkte

Abbildung 10: Rekonstruierte Bilder

$1,5cm \times 1,5cm$. Der Probentisch befindet sich in einer „Nullposition“ z. B. dort, wo es eine interessante Stelle für Wissenschaftlern ist. Der Bereich kann innerhalb $2 - 10\mu m$ liegen. Nun gerät der Probentisch so in Bewegung, dass sich das Cantilever an der ersten Messstelle befindet, an der ein Wert aufgenommen werden soll. Gleichzeitig wird von dem Gerät das Interferometer auf eine Position, die auch schon durch den Zufallsgenerator definiert ist, umgestellt. Der Wert wird gemessen und der Probentisch fährt zu einer weiteren Messposition. Das Verfahren wird solange wiederholt, bis alle Messstellen, die durch den Zufallsgenerator bestimmt wurden, gemessen sind. Wenn alle Messstellen durchgegangen wurden, können die Messwerte ausgewertet und für weitere Verwendungen angepasst werden.

Bei der Messzeit der einzelnen Messstellen kann keine Zeit eingespart werden, da alle Stellen definitiv gemessen werden müssen. Eine Beschleunigung der gesamten Messung kann nur durch Verringerung der Leerfahrtzeit erreicht werden. Das ist die Zeit, die das Gerät durch Bewegung des Probentisches und des Interferometers benötigt. Das Ziel muss sein die Summe aller Fahrzeiten, die der Probentisch und das Interferometer beim Wandern von Stelle zu Stelle benötigt, so gering wie möglich

zu halten. Wird die Nullstellung und jede Messstelle als „Planet im Universum“ betrachtet und wird die Zeit als Entfernung zwischen Planet i und Planet j definiert, die das Gerät braucht, um sich von i nach j zu bewegen, so entsteht ein Traveling Salesman Problem (TSP). Das Problem ist symmetrisch, da der Weg von i nach j genauso viel Zeit benötigt wie umgekehrt. Zu dem Problem wird noch ein weiteres Kapitel (s. 3.5.1) behandelt.

Es ist selbstverständlich, dass je nach Anwendung unterschiedliche Größen optimiert werden können. Das aber bei einem konkreten Optimierungsproblem wie die Aufnahme des Spektrums an den bestimmten Stellen der Probe noch ein Entscheidungsspielraum besteht, wie der Abstand zwischen zwei Messpunkten definiert werden soll, ist vielleicht nicht auf den ersten Blick klar. Anders ausgedrückt muss die Frage stellen: Wie lässt sich der zeitliche Abstand zwischen zwei Messpunkten bestimmen? Auf den ersten Blick kommt die Idee auf, die Entfernung mit dem gewöhnlichen, so genannten euklidischen Abstand zu messen. Ungewöhnlich ist für diesen Fall, dass es sich um einen dreidimensionale Raum handelt. Dieser wird mit $c(i, j)$, dem bezeichneten Abstand zwischen zwei Punkten $(x_i, y_i, z_i), (x_j, y_j, z_j)$ folgendermaßen definiert:

$$c(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

X und Y bestimmen in diesem Fall die Position des Probentisches unter dem Cantilever. Z wurde für die Interferometersposition angelegt.

Wenn dieser Abstand mit einem die Geschwindigkeit des Probentisches widerspiegelnden Faktor multipliziert wird, dann ist dies wahrscheinlich eine gute Einschätzung der Fahrzeit. Das ist bei der Spektrenaufnahme aber nicht unbedingt so. Der Probentisch wird von zwei Motoren angetrieben, einer bewegt sich in X-Achse, der andere in Y-Achse, nennen wir sie X- und Y-Motor. Zusätzlich steht noch ein Motor zur Verfügung, der sich um die Interferometerposition kümmert, nennen wir diesen Z-Motor. Soll das Gerät von (x_i, y_i, z_i) nach (x_j, y_j, z_j) fahren, so bewegt sich der X-Motor von x_i nach x_j, der Y-Motor von y_i nach y_j und der Z-Motor von z_i nach z_j. Derjenige Motor, der am längsten braucht, bestimmt die zeitliche Entfernung zwischen den zwei Punkten. Die Bewegung ist jedoch nicht gleichförmig. Sie beginnt mit einer Beschleunigungsphase, dann fährt der Motor mit Maximalgeschwindigkeit, danach kommt die Bremsphase. Zum Schluss muss der Probentisch und das Interferometer noch ausgerichtet werden. Bei kurzen Strecken kommt es gar nicht zur Maximalgeschwindigkeit, weil während der Beschleunigungsphase schon die Bremsphase eingeleitet werden muss. Ferner sind der X-Motor, der Y-Motor und Z-Motor

nicht gleich schnell. Das kann an technischen Gegebenheiten oder unterschiedlicher Abnutzung liegen. Die „echten“ zeitlichen Entfernungen sind also durchaus schwierig zu ermitteln, selbst dann, wenn alle technischen Parameter bekannt sind. In dieser Arbeit ist die Entfernung durch eine Näherungsformel bestimmt. Ein einfacher Ansatz zur Bestimmung der Einstellungszeit der Probe besteht darin, das Maximum der Werte $|x_i - x_j|$, $|y_i - y_j|$ und $|z_i - z_j|$ zur Entfernung zwischen den Punkten (x_i, y_i, z_i) und (x_j, y_j, z_j) zu finden. In der Mathematik wird dieser als max-Abstand bezeichnet.

$$c_{max}(i,j) := max\{|x_i - x_j|, |y_i - y_j|, |z_i - z_j|\}$$

Um zu wissen, welche der Routen optimaler ist, müssen die technischen Charakteristiken des Mikroskops betrachtet werden. Gesucht ist die Information über die Bewegungszeit zwischen den einzelnen Punkten auf den XY-Achsen und die Einstellungszeit des Interferometers. Aus der technischen Dokumentation konnten die Geschwindigkeiten ausgelesen werden. Diese sind:

XY-Bewegung: $5\mu m/s$

Interferometer: $4mm/s$

Die Informationen zum Pixelabstand in der geplanten Messung zwischen zwei Messpunkten beträgt:

DeltaX, DeltaY: $200nm$

Interferometer: $781nm$

Anhand dieser Informationen wird die Bewegungszeit der Motoren zwischen zwei Punkten berechnet. Das ist die Zeit von einem Punkt bis zum anderen nächstgelegenen Punkt, die vom Mikroskop gebraucht wird. Nach folgender physikalischen Grundformel ist die Zeit für die XY-Achse zu berechnen:

$$t = \frac{s}{v} => t = \frac{200nm}{5\mu m/s} = 40ms$$

Entsprechend dieser Formel sollte theoretisch die Umstellung der Interferometerposition zwischen zwei Punkten in der Zeit von $0,2\mu s$ erfolgen. Tatsächlich ist das Interferometer nicht sofort nach der Umstellung verwendbar und braucht relativ viel

Zeit, in Vergleich zur Positionsänderung, für die Kalibrierung. Im Endeffekt ist die Bewegungszeit für das Interferometer als Konstante mit $150ms$ angelegt. Diese bleibt immer gleich, unabhängig vom Abstand der Punkte zueinander.

3.4 Statische Routen

Die Informationen, die davor gesammelt wurden, reichen aus, um nach optimierten Routen zu suchen. Es wird aber nicht klar, ob die optimierten Routen besser oder schlechter werden. Es soll eine Möglichkeit zur Verfügung stehen, einen Vergleich von mehreren Routen durchzuführen. Das Ergebnis wird zeigen, welche der Algorithmen besser für die Messung geeignet ist. Statische Routen können nicht als optimiert bezeichnet werden, jedoch können diese einem anderen Prinzip des Ablaufs zugrunde liegen und dadurch die "Fahrzeiten"beeinflussen.

Z-Achse - Route

Die erste statische Route liest die Probe vektorweise ab (s. Abb. 11). In der 3D-Matrix kann an jeder XY-Stelle die Z-Achse als ein Vektor betrachtet werden. Anders gesagt gehören zu jeder XY-Stelle auch n Positionen an der Z-Achse. Diese Positionen, wie vorher erklärt wurde, sind die des Interferometers. Die Idee ist Folgende: Der Probentisch erreicht erste Messstelle an der XY-Position. An dieser Position werden alle Interferometerpositionen nacheinander durchgegangen und die Werte an jedem Punkt aufgenommen.

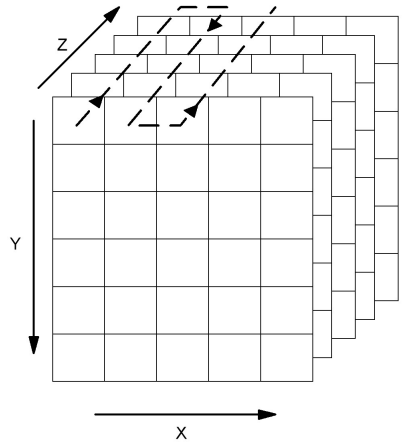


Abbildung 11: Z-Route

XY-Achse - Route

Eine andere Route ist so definiert, dass diese immer nacheinander die Ebenen misst (s. Abb. 12). Zuerst wird eine feste Interferometerposition eingestellt. Wenn es zu dieser Position irgendwelche Messpunkte auf der XY-Ebene gibt, dann fährt der Probentisch zu dem erstem Messpunkt. Wenn diese Stelle gemessen wurde, bewegt sich der Probentisch zur nächsten Stelle. Wenn alle Punkte auf dieser Ebene gemessen sind, dann geht das Gerät zur nächsten Interferometerposition über.

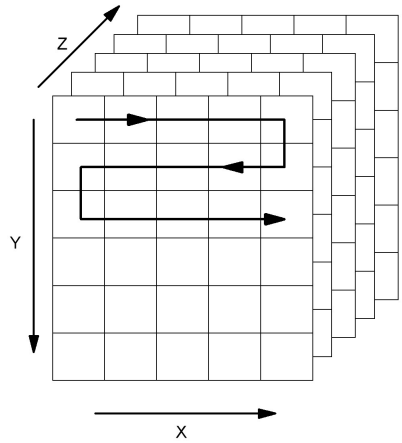


Abbildung 12: XY-Route

3.5 Optimierte Routen

Die Routenoptimierung ist das Hauptziel der Arbeit. Wie schon vorher kurz erläutert wurde (s. 3.3), ergibt sich daraus ein TSP. Bevor die optimierten Routen dargestellt werden, wird sich im folgenden Kapitel mit TSP beschäftigt.

3.5.1 TSP - Traveling Salesman Problem

Das berühmte Problem wurde in den 30er Jahren des 20. Jahrhunderts formuliert und ist immer noch eine der beliebtesten und wichtigsten Aufgaben der Kombinatorik. Der Grund liegt in seiner praktischen Bedeutung. Die Aufgabe besteht darin, die rentabelste Route zu finden, die einmal durch alle angegebenen Städte führt, und dabei zum Startpunkt zurückzukehren.

Die Optimierung dieses NP-Problems ist NP-Schwer. Dieses Problem gehört auch zur Anzahl des transcomputationalen Problems. Bereits bei einer relativ kleinen

Anzahl von Städten (mehr als 65) wird die Lösungsberechnung mehrere Milliarden Jahre dauern.

Viele bestehende Aufgaben können als Problem des Handlungsreisenden formuliert werden: ein Transportproblem, Bohren von Leiterplatten, Röntgenkristallographie, Computernetzwerke usw [Gutin]. Auch messen der Spektroskopie an den einzelnen Stellen der Probe gehört zu dieser Art von Problemen. In den letzten 50 Jahren wandten sich Physiker zunehmend dem TSP zu, insbesondere den stochastischen Versionen des Problems. Die Motivation war immer Eigenschaften zu finden, die auf ein großes Gebiet ungeordneter Systeme angewendet werden können, entweder durch gute Näherungsmethoden oder durch genaue analytische Ansätze. Die Lösung des TSP ist von großem theoretischen Interesse, da so auch andere NP-Probleme in Polynomialzeit gelöst werden können. Anders gesagt, kann das gelöste NP-Problem bei der Lösung der anderen NP-Probleme helfen.

Die Aufgaben des TSP werden durch Anwendung verschiedener Ansätze und Methoden gelöst, die aus theoretischen Berechnungen und Untersuchungen abgeleitet wurden. Alle effektiven Methoden, die die erschöpfende Suche erheblich reduzieren, gehören zur Klasse der heuristischen Methoden [Lawler]. Das Ergebnis der meisten heuristischen Methoden ist nicht die rentabelste Route, sondern nur ihre Annäherung. Oft werden sogenannte Anytime-Algorithmen benötigt, die eine aktuelle Näherungslösung schrittweise verbessern. Es kann in folgenden Gruppen von Methoden zur Lösung von TSP unterscheiden werden, die als einfach eingestuft sind:

Vollständige Suche

Vollständige Suche ist auch als „Brute Force “-Methode bezeichnet. Das ist eine Methode, mit der ein Problem gelöst wird, indem alle möglichen Lösungen aufgezählt werden. Die Komplexität dieser Aufzählung hängt von der Größe der Aufgabe ab. Wenn der Lösungsraum sehr groß ist, kann die Lösung des Problems durch umfassende Suche des Ergebnisses der Ausführung erst nach einigen Jahren und möglicherweise Jahrhunderten und Jahrtausenden gefunden werden. In der Routenoptimierung geht es um eine Route durch mehrere Tausend Messpunkte. Mit einer vollständigen Suche würde es ewig dauern.

Zufällige Suche

Manchmal kann eine finale Entscheidung als eine Folge von mehreren Teilentscheidungen dargestellt werden. Wenn solche Entscheidungen mithilfe eines zufälligen Mechanismus getroffen werden, ist der Zeitaufwand für die Lösung des Problems recht gering. Die Lösung wird wiederholt gesucht und der Mechanismus erinnert sich an die „beste Option“, dass heißt die beste der resultierenden Lösungen. Dieser einfache Ansatz kann erheblich verbessert werden, wenn es möglich ist, die Zufallssuche mit dem heuristischen Ansatz und dem lokalen Suchverfahren zu kombinieren.

Greedy-Algorithmen (gierige Algorithmen)

Diese Algorithmen zeichnen sich dadurch aus, dass sie Schritt für Schritt ohne zurückzusetzen, eine Lösung aufbauen können. Der aktuelle Schritt wählt anhand lokaler Informationen den Folgezustand aus, welcher zum Zeitpunkt der Wahl das beste Ergebnis verspricht. Bei dieser Methode entstehen keine Teillösungen. Die Basis eines Greedy-Algorithmus ist eine bereits vorsortierte Liste. Die Algorithmen sind meist sehr schnell in Vergleich zu der vollständigen oder auch zufälligen Suche. Die Lösung, die von Greedy-Algorithmen angeboten wird ist nicht die Optimalste, aber meistens immer noch eine sehr Gute.

3.5.2 TSP-Mathematica

Es wurde nach Informationen für passende Algorithmen für Routenoptimierung, die mehrere tausend Punkte bearbeiten können und die Berechnungszeit noch angemessen ist, recherchiert. Viele bereits zur Verfügung stehenden Lösungen konnten nicht den Anforderungen entsprechen. Einige Bibliotheken, die für 2-dimensionale Matrizen erstellt wurden, funktionieren nicht mit 3D-Matrizen. Eine gute Lösung hat nur die Programmiersprache Wolfram Mathematica geliefert [Mathematica].

Mathematica ist ein interaktives, benutzerfreundliches Programm zur symbolischen und numerischen Mathematik, das von der Firma Wolfram Research entwickelt wurde. Mathematica kann sowohl zahlenmäßig als auch formelmäßig Ableitungen, Stammfunktionen, Lösungen von algebraischen Gleichungen und Differentialgleichungen ausgeben.

Mathematica bietet auch die Möglichkeit eine optimale Route aus einer Liste mit Daten zu erstellen. Die Berechnung der Route ist sehr schnell. Außerdem können die Daten auch 3-dimensional sein. Welche Methode für die Optimierung der Route verwendet wurde, ist nicht vorgegeben. Das Prinzip der Berechnung liegt an dem euklidischen Abstand und lässt sich nicht ändern. Es kann nur das Ergebnis der Berechnung weiterverwendet werden. Für die optimale Route, die von Matematica erstellt wird, kann die Bewegungszeit der Motoren des Mikroskops berechnen werden. Die Zeit ist kleiner als bei beiden statischen Routen und dadurch kann eine Optimierung von Mathematica angenommen werden. Besser wäre ein Algorithmus zu verwenden, wo Max-Abstand für die Berechnung genutzt wird. Ein Beispiel der optimierten Route kann auf der Abbildung 13 visuell nachvollzogen werden.

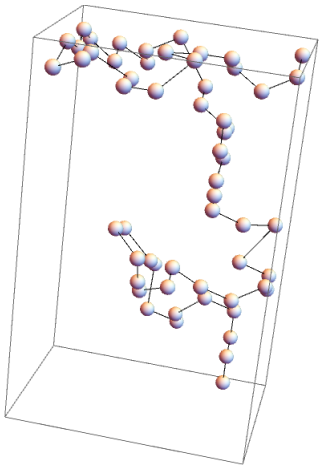


Abbildung 13: Optimierte Route

3.5.3 Suche nach einem nahe gelegenen Punkt

Die zweite optimierte Route wurde vom Autor der Bachelorthesis speziell für Mikroskop NeaSNOM entwickelt. Die Suche erfolgt nach dem *nearest neighbour algorithm*, mit der Berechnung der Entfernung zwischen zwei Messstellen mit Max-Abstand. Der *nearest neighbour algorithm* ist ein Greedy-Algorithmus. Er sucht vom Ausgangspunkt den nächsten Punkt. Alle Punkte sind meist in einer Liste definiert. Um einen Punkt zu finden, wird das System für den aktuellen Punkt den Abstand aller Punkte in der Liste überprüfen. Wenn der kleinste Abstand gefunden ist, wird aus der Liste der aktuelle Punkt gelöscht. Nun wird der Punkt, der die geringste Entfernung hatte, als aktueller Messpunkt genommen. Die Berechnungszeit für diese Methode kann durch eine Formel definieren werden. Die Berechnungszeit lässt sich allerdings noch optimieren. Es wurde berechnet, dass der minimale zeitliche Abstand 40 ms beträgt. Somit kann die Suche in der Liste an der Stelle, wo zwei Messpunkte zu nahe zueinander liegen, unterbrochen werden und in diesem Fall als bestes Ergebnis angenommen werden.

3.6 Routenvergleich

Nachdem alle statischen und optimierten Routen erstellt wurden, müssen diese miteinander verglichen werden. Dadurch wird bestimmt, welche der vier Routen besser für das Mikroskop geeignet ist. Um die Routen zu vergleichen, wurden dieselben Eingangsdaten ausgewählt. Die Daten wurden aus der vorgelegten 3D-Matrix zufällig gezogen. Der Vergleich erfolgte in mehreren Schritten, die jeweils eine andere Anzahl von Punkten enthalten. Auf diese Weise konnten alle vier Routen in unterschiedlichen Situationen analysiert werden. Als Resultat entstand eine Tabelle (s. Abb. 14), die alle Messergebnisse beinhaltet. Die Tabelle basiert auf den ursprüng-

Menge der Punkte			Algorithmus - Zeit des Weges in Sekunden			
Punkte pro Vektor	Insgesamt	Relevant	Z-Route	XY-Route	TSP - Mathematica	Suche nach einem nahe gelegenen Punkt
10	18040	5237	785,2	3344	744	730,6
20	36080	10770	1615	6393	1451	1383
30	54120	15899	2385	8853	2085	1920
40	72160	21260	3189	11063	2728	2417
50	90200	26362	3954	12647	3297	2839
60	108240	31855	4778	14097	3900	3246
70	126280	37193	5579	15238	4476	3620
80	144320	42340	6351	16052	5010	3956
90	162360	47807	7170	16711	5578	4280
100	180400	52891	7933	17243	6094	4573
150	270600	79397	11909	18996	8660	5904
200	360800	105845	15876	19947	11205	7045
250	451000	132916	19937	20437	13715	8113

Abbildung 14: Vergleich der Routen - Tabelle

lich angegebenen Daten und liefert weitere Informationen. Die erste Spalte stellt die Anzahl der Punkte dar, die an jeder XY-Position genommen wurden. Die nächste Spalte zeigt die Anzahl aller Punkte aus der 3D-Matrix. In der dritten Spalte befindet sich die Anzahl der relevanten Punkte, die in *CS* verwendet werden. Die weiteren vier Spalten zeigen die Zeit in Sekunden, die das Mikroskop für Messungen benötigt. Jede Spalte davon gehört zu einer Route, die oben erläutert wurde. Die Zeiten wurden anhand der technischen Information des NeaSNOMs berechnet und können bei der praktischen Anwendung abweichen. Die Abweichung gilt dann für alle vier Routen und wird den gesamten Überblick nicht ändern. Basierend auf den Daten aus dieser Tabelle wurde auch ein Diagramm erstellt, das eine visuelle Wahrnehmung erleichtert. Aus der Tabelle geht hervor, dass mit einer kleinen Anzahl von Punkten alle Routen außer der statischen XY-Straße fast zur gleichen Zeit ausge-

führt werden. Mit zunehmender Punktzahl erhöht sich der Zeitunterschied zwischen den erforderlichen Messungen.

Die Grafik (s. Abb. 15) zeigt, dass optimierte Routen bessere Ergebnisse liefern als statische Routen. Alle Routen auf der Grafik außer der XY-Route sind fast linear. Dadurch es ist möglich anhand der Punktenanzahl die Zeit der Messung mit einer von diesen Routen vorherzusagen.

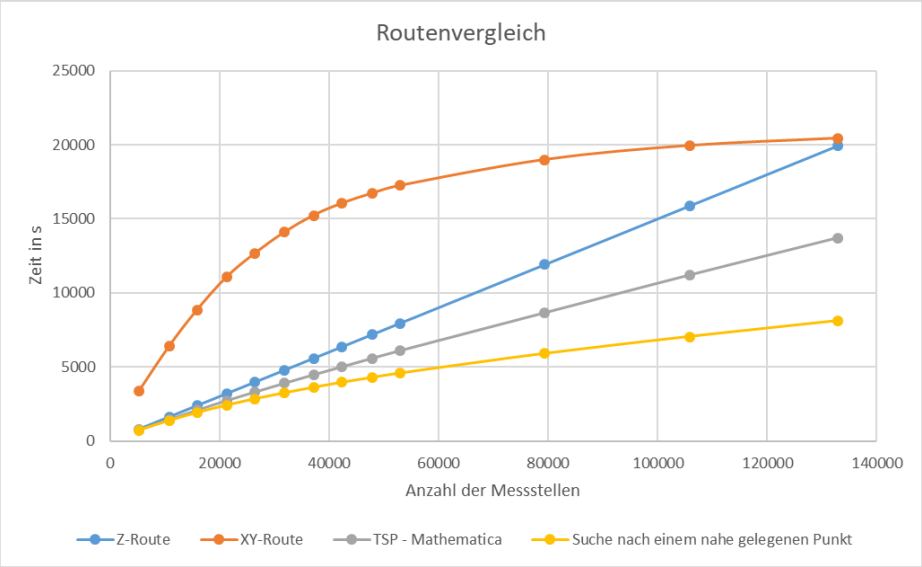


Abbildung 15: Vergleich der Routen - Diagramm

Das beste Ergebnis zeigt die vom Autor entwickelte Route. Um beispielsweise 10% der Informationen, die für *CS* nötig sind, mit dieser Route zu messen, wird fast zweimal weniger Zeit als bei der besten statischen Route gebraucht. Die Berechnung dieser Route dauert jedoch ziemlich lange. Die ungefähre Zeit, um die Route durch 50.000 Punkten zu berechnen, liegt bei etwa einer halben Stunde.

Nach diesem Vergleich konnte die optimierte Route für das Mikroskop angewendet werden. Das Mikroskop selbst muss erst dafür vorbereitet bzw. vorprogammiert werden.

4 Mikroskopsteuerung

In diesem Kapitel wird beschrieben, wie eine Verbindung mit dem Mikroskop NeaSNOM aufgebaut wird und wie eine Messung abläuft. Zusätzlich wird über das Format der Datenaufnahme und über die Konvertierung der Daten für Matlab-SKript berichtet.

4.1 Verbindung und Ablauf

Die Verbindung des Geräts wird mit einem eigenen GUI hergestellt. Das GUI bietet dem Nutzer eine umfangreiche Funktionalität für vielseitige Messungen von diversen Proben. Es gibt keine Möglichkeit in dem GUI der Hersteller eine eigene Route für die Messungen anzulegen. Für solche Fälle steht ein SDK zu Verfügung. Es können viele Funktionen des Mikroskops in eigenen Programmen genutzt werden. Das macht die Nutzung und Steuerung des Mikroskops noch flexibler.

Wenn eine Probe, die gemessen werden soll, vorbereitet ist und sich auf dem Probentisch befindet, kann das Programm die Steuerung übernehmen. Zuerst erfolgt eine Verbindung mit dem Server, der das Mikroskop steuert. Wenn eine Verbindung steht, kann die Spitze des Cantilevers an die Oberfläche der Probe angenähert werden. Dafür dienen die einfachen Funktionsaufrufe auf einer beliebigen Plattform mit einer Programmiersprache, die Bibliotheken des NeaSNOMs anbinden kann.

4.1.1 Datenaufnahme

Sobald das Programm sich mit dem Gerät verbunden hat und das Cantilever angenähert ist, kann die Datenaufnahme gestartet werden. Das NeaSNOM bietet eine große Menge von Kanälen, die jeweils eigene Information liefern. Insgesamt sind es über 20 diverse Kanäle. Das sind mechanische, optische und topografische Informationen. Jeder Kanal repräsentiert dabei eine andere Messmethode, bei der unterschiedliche Signale aufgezeichnet werden [Taubner]. Für diese Arbeit sind davon nur zwei optische Kanäle für die weitere Verwendung wichtig:

„O2A“: Optisches Signal der Amplitude zweiter Ordnung.

„O2P“: Optisches Signal der Phase zweiter Ordnung.

Die optischen und mechanischen Kanäle des Mikroskops werden dabei in verschiedene Ordnungen unterteilt. Mit einer Ordnung ist der Grad der Modulation eines Lock-In Verstärkers gemeint. Ein Lock-In Verstärker kann mithilfe eines Referenzsignals ein verstärktes Messsignal modulieren. Dafür müssen die Phase und die Frequenz des Referenzsignals bekannt sein [Zurich Instr.]. Nach einer Standardmessung einer Area auf der Probe kommt als Ergebnis ein Array mit den Daten heraus. Für die optimierte Route müssen die Daten nach jedem Messpunkt abgelesen und gesichert werden. Die Daten werden in diesem Fall ähnlich zur Standardmessung in einem Array gehalten.

4.1.2 Progressdarstellung

Die Standardmessung bietet dem Nutzer die Möglichkeit den Progress der Messung zu verfolgen. Das ist ein Balken, der die Zeit bis zum Ende der Messung einer Area darstellt. Diese Funktion macht die Kommunikation zwischen dem Nutzer und dem Gerät deterministischer. Dadurch ist es auch einfacher zu erkennen, ob das System noch funktioniert oder abgestürzt ist. Die Abstürzung des Systems kann diverse Ursachen haben. Bei der Messung mit einer optimierten Route kann der Progress auch genutzt werden, aber nur für jeweils einen Messpunkt. Das ist nicht sinnvoll, da die Messung eines Punktes weniger als eine Sekunde dauert. Da die Route offensichtlich mit einer Schleife abgearbeitet wird und die Anzahl der Punkte bekannt ist, kann eine anderere Progressbar eingesetzt werden. Anhand der Information wie lange eine Iteration der Schleife dauert kann die angenäherte Dauer der Messung berechnet werden. Angenähert wird nur, weil jede Iteration verschiedene Zeiten benötigen kann. Nach jeder Iteration kann die Zeit korrigiert werden, um die Messungsdauer genauer darzustellen. Genauso wird der Progress in Prozent angezeigt und nach jeder Iteration aktualisiert. So kann ein Nutzer erkennen wie weit die Messung vorangeschritten ist und wie lange es noch dauern kann.

4.2 Datentransformierung

Nachdem die Messung durchgeführt wurde, entstehen die Daten an beiden Kanäle O2A und O2P. Für jede Stelle ist ein Wert pro Kanal, der als reelle Zahl dargestellt ist, zuständig. Das Mathlab-Skript erwartet als Eingangswerte komplexe Zahlen [Engel]. Aus diesen Gründen ist es nötig, die Werte zu konvertieren. Da der O2A-Kanal die Amplitude und der O2P-Kanal die Phase liefert, sind das bereits

Polarkoordinaten. Entsprechend der trigonometrischen Form können die Werte, also Polarkoordinaten, eingesetzt werden um eine komplexe Zahl zu berechnen. Die trigonometrische Form zur Berechnung liegt vor:

$z = O2A(\cos O2P + i \sin O2P)$

Wenn alle Werte konvertiert sind, müssen nur noch die Arrays für das Matlab-Skript vorbereitet werden.

5 Implementierung

In diesem Abschnitt wird die Implementierung der beschriebenen Algorithmen und Funktionalitäten erklärt. Es wird beschrieben, für welche Programmiersprache sich entschieden wurde und welche Bibliotheken verwendet wurden. Auch die Struktur des Programms wird erklärt, sowie die Methoden und Ergebnisse zum Testen des Codes vorgestellt.

5.1 Programmiersprache

Bevor die Algorithmen implementiert werden können, stand lag Auswahl von Programmiersprachen vor. Eine der Anforderungen der Stakeholder an die Programmiersprache ist, dass sie selbst die Sprache verstehen müssen. Die Wissenschaftler beschäftigen sich hauptsächlich mit Produkten wie Wolfram Mathematica, Matlab und auch Python. Das bedeutet, dass alle anderen Sprachen gar nicht für die aktuelle Bachelorthese berücksichtigt werden. Alle drei Sprachen haben ihre eigenen Nach- und Vorteile. Die Auswahlkriterien sind in der folgenden Tabelle 16 dargestellt.

	Mathematica	Matlab	Python
Kostenpflichtig	Ja	Teilweise	Nein
Erweiterungspakete	Hat nur eigene	Ja, gibt es	Ja, und sehr viel
Eigene Erfahrung	Nein	Nein	Ja
Steuerung der Geräte	Nein	Ja	Ja
Vorinstalliert	Nein	Nein	Ja
Berechnungszeit	Sehr schnell	Schnell	Schnell

Abbildung 16: Vor- und Nachteile der Programmiersprachen

Nach dem Vergleich der Sprachen fiel die Wahl bei der Programmiersprache auf Python. Wichtig bei dieser Entscheidung waren die eigenen Erfahrungen und die Möglichkeit die Sprache auf allen nötigen Plattformen installiert zu haben. Auch die Erweiterung der Sprache durch das einfache Hinzufügen der Bibliotheken ist relevant. Andersrum wurden die anderen zwei Sprachen auch teilweise verwendet. Mathematica für die Berechnung der optimalen Route. Mathlab um zu verstehen, welche Eingangsdaten *CS* braucht.

Python

Python ist eine plattformunabhängige, interpretierte höhere Programmiersprache. Die Sprache gehört der Python Software Foundation. [Weigend]. Python wurde 1991 von Guido van Rossum im Centrum Wiskunde & Informatica in Amsterdam entwickelt. Python ist für die meisten gängigen Betriebssysteme frei erhältlich und bei den meisten Linux-Distributionen im Standardumfang enthalten. Python wird oft in wissenschaftlichen Bereichen als Skriptsprache eingesetzt, unterstützt aber mehrere Programmierparadigmen, wie die objektorientierte, die aspektorientierte und die funktionale Programmierung. Python verwendet die dynamische Typisierung, also eine Typisierung, bei der erst zur Laufzeit auf Variablentypen geprüft wird, statt zur Kompilierung. Das Ziel bei der Entwicklung von Python war die so zu entwerfen, dass sie einfacher und übersichtlicher als andere Sprachen ist. Das soll vor allem durch Reduzierung der Syntax, wie auslassen von geschweiften Klammern und Semikolons am Ende von Anweisungen, und Reduzierung der Schlüsselwörter erreicht werden. Statt mit Klammern und Semikolons zu arbeiten, wie viele andere Programmiersprachen, verwendet Python die Zeilenumbrüche und Einrückungen dafür. Diese Einführung verbessert die Lesbarkeit des Quellcodes. Python verfügt über solche mächtigen Bibliotheken wie *Numpy*, *Scipy* oder *Pandas*, die vor allem für wissenschaftliche Projekte nützlich sind. Dadurch hat diese Sprache das Interesse von vielen Wissenschaftlern geweckt. Diese Pakete bieten viele hoch optimierte mathematische Grundlagen, wie Matrizenmanipulation, Signalverarbeitung, Bildverarbeitung und viele weitere Möglichkeiten. In diesem Projekt sind von diesen Bibliotheken *Numpy* und *Scipy* verwendet worden.

Pakete

Neben den mächtigen Paketen *Numpy* und *Scipy*, die schon vorgestellt wurden, wurden in diesem Projekt noch weitere nicht mehr so mächtige aber immer noch wichtige Bibliotheken angewendet. In folgendem Abschnitt werden diese mit kurze Beschreibung aufgelistet.

sys *Sys* ist Teil der Standardbibliothek von Python und stellt Variablen und Funktion bereit, die in unmittelbarem Zusammenhang mit dem Python-Interpreter selbst stehen. In diesem Projekt steht diese Bibliothek zur Einbindung des SDKs deren Pfad in den Systempfad übernommen wird. So-

mit können diese danach über den Import-Befehl eingebunden werden.

clr Die Abkürzung clr steht für *Common Language Runtime* und ist Bestandteil von Pythonnet. Damit lassen sich .NET Programme in Python einbinden, indem die clr Namespaces als Pythonpakete behandelt werden. Es ist notwendig das Paket Pythonnet installiert zu haben.

time ist ein einfaches Paket und gehört zur Standardbibliothek. Es bietet Manipulationen von Zeiten im Quellcode an und erlaubt einen Zugriff auf die Systemuhr.

os os bietet in Python grundlegende Betriebssystemfunktionen. Gebraucht wird diese Bibliothek, um Pfade zu verwalten und zu erstellen.

random Dieses Paket ermöglicht die Pseudo – Zufallszahlen zu generieren. Es wird Pseudo genannt, weil die scheinbar zufälligen Zahlen in Wirklichkeit berechenbar sind.

math Diese Standardbibliothek erlaubt einfache mathematische Funktionen zu Nutzen. Zum Beispiel Wurzel ziehen oder Sinus berechnen.

tqdm tqdm muss als taqadum gelesen werden und das bedeutet "Progress" in arabischer Sprache. Damit ist es absolut einfach den Progress des Schleifenablaufs darzustellen.

Numpy NumPy (Numeric Python) ist ein Open-Source-Modul für Python, das allgemeine mathematische und numerische Operationen in Form vorkompilierter, schneller Funktionen bietet. Sie werden in hochwertigen Paketen kombiniert. Sie bieten Funktionen, die mit den Funktionen von MatLab verglichen werden können. NumPy bietet grundlegende Methoden zum Bearbeiten großer Arrays und Matrizen an. In diesem Projekt ist Numpy für die Bearbeitung der Messdaten geeignet, da diese als große 3D-Matrizen vorliegen.

Scipy SciPy (Scientific Python) erweitert die Numpy-Funktionalität um eine große Sammlung nützlicher Algorithmen wie Minimierung, Fourier-Transformation, Regression und andere angewandte mathematische Techniken. Außerdem erlaubt SciPy die Dateien mit Daten aus der MatLab zu nutzen. Solche Dateien werden in NumPy-Arrays konvertiert. Andersrum geht es auch und das wurde verwendet um die Daten im MatLab-Format abzuspeichern.

5.2 Programmstruktur

Das Programm besteht insgesamt aus zwei Skripten. Ein Skript ist für die Routenberechnung zuständig und das andere für die Steuerung des Mikroskops anhand einer Route. Diese Unterteilung wurde gemacht, weil beide Fälle relativ viel Zeit für die Berechnung des Ergebnisses brauchen. Es ist sinnvoll die Route separat zu berechnen und wenn diese zur Verfügung steht eine Messung starten.

Routeberechnung

Dieses Skript beinhaltet Funktionen, die anhand eines zufälligen Datensatzes, verschiedene Routen zusammenstellen kann. Die Matrizenparameter werden festgelegt aus der Breite: X-Achse, Höhe: Y-Achse und Tiefe: Interferometerposition. Außerdem ist die Information über die untere und obere Grenze nötig. Diese Informationen werden gebraucht um die zufälligen Punkte zwischen diesen Grenzen auszuwählen. Als weitere Parameter kommen die Anzahl der Punkte, die zwischen beiden Grenzen gesucht sind. Zwei letzte Parameter stellen die Zeit, die die Motoren von einem Punkt bis zu anderem brauchen in ms dar.

Jede Funktion ist für eine eigene Aufgabe zuständig. Eine Funktion stellt die Liste mit zufällig ausgewählten Stellen bereit. Drei Funktionen suchen eigene Routen anhand dieser Liste. Das sind die Routen: XY-Route, Z-Route, und die Route mit der Suche nach einem nahen gelegenen Punkt. Für die TSP-Route ist die Funktion anders implementiert. Diese Route braucht die Liste die mit Wolfram Mathematica bearbeitet wurde. Da der Zugang zu Mathematica limitiert und nicht überall möglich ist, bleibt diese Funktion eher nur als Teil dieses Projektes. Außerdem gibt es noch eine letzte Funktion, die die Zeit für den Routenansatz anhand der vorgegebenen Zeiten berechnet.

Als Ergebnis liefert jede Funktion, die Routen berechnet, die Gesamtzeit. Die Route mit der Suche nach einem nahe gelegenen Punkt liefert zusätzlich die Liste mit dem optimierten Pfad durch alle Punkte.

Mikroskopsteuerung

Unter der Mikroskopsteuerung ist auch die Kommunikation mit dem Gerät gemeint. Die Kommunikation mit dem neaSNOM Mikroskop erfolgt über die SDKs des Herstellers. Diese werden verwendet um die korrekte Verbindung zum Mikroskop herzustellen. Das neaSNOM Mikroskop ist an einem Computer angeschlossen, der als Server dient. Das Skript ist in diesem Fall als ein Client zu sehen, der sich über TCP/IP mit dem Server verbindet. Die IP oder ein Netzwerkname des Mikroskops werden als Parameter in diesem Skript geschrieben. Wenn der Verbindungsaufbau erfolgreich ist, dann wird vom SDK ein Mikroskop-Objekt zurückgegeben. Zu diesem Objekt können viele Funktionen, die auch im SDK vordefiniert sind, angewendet werden. So wird damit die Annäherung der Spitze durchgeführt und schließlich die Spektroskopie gestartet. Die Spektroskopie erfolgt durch eine Schleife. Um einen Scan zu starten, braucht das neaSNOM Mikroskop zunächst noch die Werte der Messung. Die nötigen Parameter dafür sind die Startposition und die Aufnahmezeit des Spektrums. Dann wird die Messung gestartet. Für jede Iteration wird ein Scan-Objekt zurückgegeben. Dieses wird benutzt um die Information der Kanäle O2A und O2P abzulesen und zu den Numpy-Arrays zuzuführen. Diese Arrays werden am Ende der Messung in die Arrays für das MatLab-Script konvertiert und als MatLab-Datensatz abgespeichert.

6 Ergebnis

Dieser letzter Abschnitt stellt das Ergebnis zu dieser Bachelorarbeit vor. Hier werden nachfolgend alle Durchführungen und Erkenntnisse zusammengefasst. Es wird dargelegt, ob diese Arbeit erfolgreich war und ob die Bachelorthese bestätigt werden konnte. Es wird auch beschrieben, wie das Programm und die einzelnen Funktionen getestet wurden. Des Weiteren wird beschrieben, welche Schwierigkeiten während der Arbeit aufgetreten sind und wie diese umgegangen wurden. Am Ende wird berichtet, wie es mit diesem Projekt weitergeht und wie dieses erweitert werden kann.

6.1 Zusammenfassung

Diese Arbeit wurde an der PTB durchgeführt und befasst sich mit der These:

> *Es ist möglich, anhand eines vorgegebenen, kompletten Datensatzes einen für nachträgliche Rekonstruktion geeigneter reduzierten Datensatz zu erzeugen, eine minimierte Route zu finden sowie die Vorbereitung zur Anwendung von Compressed Sensing durchzuführen. Unter Anderem ist es auch machbar die einzelnen zufälligen Punkte mit dem Nahfeldmikroskop entsprechend der Route aufzunehmen.*

Zu diesem Zweck wurde ein Optimierungsalgorithmus für die Routenminimierung entwickelt, der auf dem Nearest neighbour algorithm basiert und zu dem TSP-Problem gehört. Der Algorithmus findet nicht die optimierteste Route, aber dafür verwendet er auch nicht so viel Zeit für die Berechnung. Dieser Algorithmus sowie die Steuerung des Mikroskopes über das SDK der Firma neaspec GmbH wurden mit der Programmiersprache Python implementiert.

6.2 Test

Es wurden standardmäßig die üblichen Software- und Funktionentests während der Implementierung durchgeführt. Diese gehören meistens zu einer technischen Dokumentation und werden hier nicht genauer beschrieben. Es muss aber gesagt werden, dass das Programm durch alle Tests kommt. Es wurde noch ein zusätzlicher

Test durchgeführt, der zu der Bachelorthese gehört und hier dargestellt werden muss.

Dieser Test stellt den kompletten Ablauf der Messung vor. Zuerst wurde eine optimierte Route (s. Abb. 17) entsprechend des implementierten Algorithmus erstellt. Danach wurde anhand dieser Route das Spektrum von der Probe gemessen. Schließlich wurden die davon entstandenen Daten mit dem MatLab-Script abgearbeitet (s. Abb. 18). Auf der Abbildung ist zu erkennen, dass aus den relativ wenigen Punkten, die gemessen wurden, eine akkurate Rekonstruktion entstanden ist. Wird diese verglichen mit der Abb. 9, kann erkannt werden, dass die Rekonstruktion nahe an das Original herankommt. Ziel dieses Tests war, dass ein Ergebnis geliefert wird, welches zu erwarten ist. Dadurch sollte gezeigt werden, ob und wie gut der Algorithmus und die Steuerung des Gerätes funktioniert. Ein solcher Test dauert insgesamt bis zu einem Arbeitstag. Dazu gehört auch die Suche der bestimmten Position auf der Probe und diversen Voreinstellungen vor der Messung, die manchmal länger dauern können als die Messung selbst.

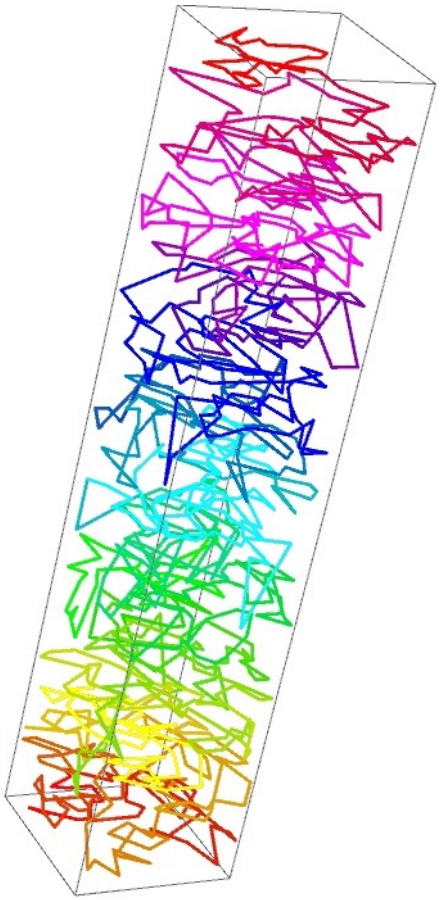


Abbildung 17: Verlauf der Route in 3D-Raum

Während der Messungen sind einige Fehler aufgetreten. Die Fehler konnten nicht beseitigt oder vermieden werden, da diese auf der Server-Seite passierten. Der Ser-

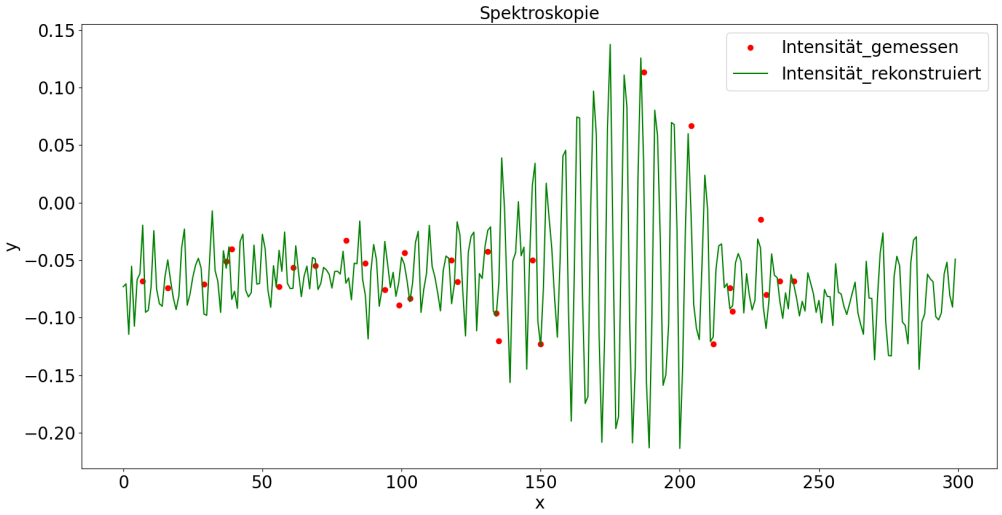


Abbildung 18: Gemessene und rekonstruierte Daten

ver liefert keine Antwort auf die gestellte Anfrage und der Client nimmt an, dass die Messung noch durchgeführt wird. Ein anderes Problem ist die Zeit der Messung. Die Zeit für die Einstellung der Motoren des Mikroskops stimmt mit den berechneten Werte nicht überein. Für die Bewegung von einem Punkt zu einem anderen wird deutlich mehr Zeit gebraucht. Diese Erkenntnis ändert aber nicht die Gesamtübersicht der Routen. Nur die Zeiten müssen mit einem Faktor multipliziert werden. Aus den Logs des Mikroskops, wo alle Befehle protokolliert sind, konnte entnommen werden, dass bei einem Funktionsaufruf die Parameter der Geschwindigkeit bei dem Server nicht berücksichtigt werden. Die Firma neaspec wurde davon in Kenntnis gesetzt. Da sich mit dem SDK für NeaSNOM sich nicht so viele Nutzer beschäftigen, beinhaltet die Software des Herstellers noch nicht viele getestete Fälle. Dadurch können unerwartete Fehler auftreten. Die Firma neaspec hat schnell auf den Report reagiert und wird versuchen die Fehler zu beseitigen.

Das Ergebnis der kompletten Messung ist auf der Abbildung 19 zu sehen. Auf dem Bild ist zu sehen den sternförmigen Abdruck auf der Oberfläche der Probe.

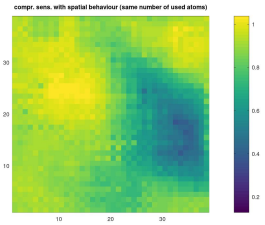


Abbildung 19: Messergebnis aus rekonstruierte Daten

Bei der Auswertung der Messung ist zu erkennen, dass die Messung und vor allem der Ansatz der optimierten Route funktioniert, wie es zu erwarten war.

6.3 Fazit

Nach dem Abschluss der durchgeführten Arbeit kann die am Anfang gestellte These als bestätigt bezeichnet werden. Es wurden alle einzelnen Schritte der Arbeit dargestellt: Minimierung der Route, Erfassung der Daten anhand der Route und *CS* anhand der erfassten Daten. Das Programm funktioniert und steht bereits zur Verfügung. Das System liefert positive Ergebnisse, aber kann immer noch verbessert werden.

6.4 Ausblick

Da die Zeit für dieses Projekt begrenzt war, konnten nicht alle möglichen Ideen für die Optimierung der Route umgesetzt werden. Ein Anfang wurde dafür präsentiert und anhand der Informationen, die in dieser Arbeit beschrieben sind, kann das Projekt erweitert und optimiert werden. Die Optimierungen können sowohl auf der Seite des Herstellers von NeaSNOM vorkommen als auch von der Seite der Entwickler, die für das Mikroskop diverse Software schreiben.

Eine Idee für die weitere Optimierung der Messung ist, die statischen und optimierten Routen zu kombinieren. Die Optimierte Route hat weniger zufällig ausgewählte Messpunkte, aber dafür werden an jeder Stelle nicht nur eine, sondern mehrere Messungen durchgeführt. Die Messungen können mit schon vorgegebenen Funktionen von NeaSNOM gestartet werden. Das soll die Berechnung der Route deutlich beschleunigen. Der Ablauf der Messung in diesem Fall wird auch weniger Zeit verbrauchen im Vergleich zur optimierten Route mit der gleichen Anzahl an Messstellen. Eine andere Idee ist die Verwendung der künstlichen Intelligenz für das Projekt. Es können verschiedene Methoden für verschiedene Schritte des Projektes angesetzt werden. Evolutionäre Algorithmen können für die Optimierung der Routen verwendet werden. Ein neuronales Netz, kann eventuell *CS* ersetzen. Das neuronale Netz muss dann in der Lage sein im Prozess der Messung die Bilder aus schon aufgenommenen Daten herzustellen. Je mehr Daten, desto besser wird die Qualität und der Nutzer kann dann selber bestimmen, wann die Qualität der Messung ausreichend ist.

Webquellen

[Gwyddion] Gwyddion, 2020, *Gwyddion – Free SPM (AFM, SNOM/NSOM, STM, MFM, ...) data analysis software* [online].
Letzter Zugriff am 28.07.2020. Verfügbar unter: http://gwyddion.net/.

[Helmholtz] Helmholtz Gemeinschaft Deutscher Forschungszentren, 2020, *Home - Helmholtz-Gemeinschaft Deutscher Forschungszentren* [online].
Letzter Zugriff am 08.07.2020. Verfügbar unter: https://www.helmholtz.de/.

[HZB] Helmholtz-Zentrum Berlin, 2020, *Helmholtz-Zentrum Berlin (HZB) - Home* [online].
Letzter Zugriff am 08.07.2020. Verfügbar unter: https://www.helmholtz-berlin.de/.

[Mathematica] Wolfram, 2015, *FindShortestTour* [online].
Letzter Zugriff am 23.07.2020. Verfügbar unter: https://reference.wolfram.com/language/ref/FindShortestTour.html.

[neaspec] neaspec GmbH, 2019, *Home - neaspec GmbH* [online].
Letzter Zugriff am 20.07.2019. Verfügbar unter https://www.neaspec.com/.

[PTB] Physikalisch-Technische Bundesanstalt, 2019, *Physikalisch-Technische Bundesanstalt* [online].
Letzter Zugriff am 20.07.2019. Verfügbar unter: https://www.ptb.de/cms/.

Literaturverzeichnis

[Davenport] M. Davenport, M. Duarte, Y. Eldar, G. Kutyniok, 2012, *Compressed sensing: Theory and Applications*, Cambridge University Press, 2012

[Engel] Joachim Engel , Andreas Fest, 2016, *Komplexe Zahlen und ebene Geometrie*, De Gruyter, 2016, ISBN: 978-3-110-40686-3.

[MLS] J. Feikes, M. von Hartrott, M. Ries, P. Schmid, G. Wüstefeld, a. Hoehl, R. Klein, R. Müller, and G. Ulm, 2011 *Metrology Light Source: The first electron storage ring optimized for generating coherent THz radiation*, Phys. Rev. Spec. Top. - Accel. Beams 14, 030705

[Eldar] Y. C. Eldar and G. Kutyniok, 2012 *Compressed Sensing: Theory and Applications*, Cambridge University Press, ISBN: 978-1-107-00558-7

[Green] Green, G., 1828, *An Essay on the Application of Mathematical Analysis to the Theories of Electricity and Magnetism*, Rare-BookClub.com, Peterborough, 2012, ISBN: 978-1231136904.

[Griffiths] Griffiths, P. R., Haseth, J. A. d. und Haseth, J. A. D. 1986, *Fourier transform infrared spectrometry*, Wiley, New York, 1986, ISBN: 978-0-471-09902-4.

[Gutin] Gregory Gutin and Abraham P. Punnen, 2002, *The Traveling Salesman Problem and Its Variations*, Kluwer, Dordrecht, 2002, ISBN: 978-0-306-48213-7

[Hermann] Hermann, P. Kästner, B. Hoehl, A. Kashcheyevs, V. Patoka, P., Ulrich, G., Feikes, J., Ries, M., Tydecks, T., Beckhoff, B., Rühl, E., und Ulm, G., 2017, *Enhancing the sensitivity of nano-FTIR spectroscopy*, Opt. Express, 2017, Seiten 16574–16588, DOI: 10.1364/OE.25.016574.

[Hofmann] Hofmann, A., 2004, *The Physics of Synchrotron Radiation*, Cambridge University Press, Cambridge, 2004, ISBN: 978-0-521-

30826-7.

[Huth] Huth, F., Govyadinov, A. Amarie, S. Nuansing, W., Keilmann, F. und Hillenbrand, R., 2012, *Nano-FTIR absorption spectroscopy of molecular fingerprints at 20 nm spatial resolution.*, Nano letters, 2012, Seiten 3973–8, DOI: 10.1021/nl30 1159v.

[Kästner] Kästner, B., Schmähling, F., Hornemann, A., Ulrich, G., Hoehl, A., Kruskopf, M., Pierz, K., Raschke, M. B., Wübbeler, G. und Elster, C., 2018, *Compressed sensing FTIR nano-spectroscopy and nano-imaging*, Optics Express, 2018, Seite 18115, DOI: 10.1364/OE.26.018115.

[Klapetek] Klapetek, P., 2012, *Quantitative Data Processing in Scanning Probe Microscopy - SPM Applications for Nanometrology*, William Andrew, Zagreb, 2012, ISBN: 978-1-455-73059-9.

[Lawler] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, 1985, *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, Wiley, Chichester, 1985, ISBN 10: 0471904139 ISBN 13: 9780471904137

[neaSNOM] neaspec GmbH, 2015, *neaSNOM System Description*, neaspec GmbH, München, 2015, neaSNOM Handbuch.

[Riley] Riley, K. F., 2006, *Mathematical Methods for Physics and Engineering - A Comprehensive Guide*, Cambridge University Press, Cambridge, 2006, ISBN: 978-1-139-45099-7.

[Rupp] Rupp, C., Simon, M. und Hocker, F., 2009, *Requirements Engineering und Management*, HMD Praxis der Wirtschaftsinformatik, 2009, Seiten 94–103, DOI: 10.1007/BF03340367.

[Taubner] Taubner, T., 2005, *Infrarotspektroskopie im Nahfeld einer Tastspitze*, Shaker, Aachen, 2005, ISBN: 978-3-832-24237-4.

[Ulrich] Ulrich, H. und Weber, H., 2017, *Laplace-, Fourier- und z-Transformation - Grundlagen und Anwendungen*, Springer-Verlag, Berlin Heidelberg New York, 2017, ISBN: 978-3-658-03450-4.

[Weigend] Weigend, M., 2018, *Python 3 - Lernen und professionell anwenden. Das umfassende Praxisbuch*, MITP-Verlags GmbH & Co. KG, Heidelberg, 2018, ISBN: 978-3-958-45793-5.

[Wiesendanger] Wiesendanger, R., 1994, *Scanning Probe Microscopy and Spectroscopy - Methods and Applications*, Cambridge University Press, Cambridge, 1994, ISBN: 978-0-521-42847-7.

[Zurich Instr.] Zurich Instruments, 2016, *White Paper - Principles of lock-in detection and the state of the art*, Zurich Instruments, Zürich, 2016, White Paper.

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich:

1. dass ich meine Bachelor-Thesis selbstständig verfasst habe,

2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe,

3. dass ich meine Bachelor-Thesis bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Ort, Datum	Oleg Kornelsen