



Digitale Medien Semesterarbeit WS 2002/03

Bildskelettierung dendritisch gewachsener Kristalle

Giselher Wichmann, Aurel Thomi

5. Februar 2003

Letzte Änderung: 6. Mai 2003



© Copyright 2003 G.Wichmann & A.Thomi, Switzerland

Ohne vorherige ausdrückliche schriftliche Genehmigung des Autors dürfen keine Auszüge dieses Berichtes vervielfältigt, in einem Informationssystem gespeichert oder in irgendeiner Form weiter übertragen werden. Es wird keine Gewähr übernommen, dass die in diesem Bericht enthaltenen Angaben frei von Patentrechten sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Bericht berechtigt auch ohne besondere Kennzeichnung nicht zur Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürfen. Fehler, Änderungen und Ergänzungen vorbehalten.

Inhaltsverzeichnis

1	Einleitung	4
2	Aufgabenstellung	5
3	Projekttablauf	6
4	Die morphologische Skelettierung	7
4.1	Ziele einer Skelettierung.....	7
4.2	Der Verarbeitungsprozess einer morphologischen Skelettierung.....	7
4.3	Grenzen der morphologischen Skelettierung.....	8
5	Die Medial Axis Transformation (MAT)	9
6	Auswertung der pixelorientierten Verfahren	12
7	Splines	12
8	Dynamisches Programmieren	13
9	Erweiterte LaGrange-Methode	14
10	Detektion der Hauptspitze	15
11	Weiterführende Überlegungen	17
11.1	Physikalische Ansätze	17
11.2	Ansatz Fouriertransformation	17
11.3	Ansatz zur Seitenastbestimmung	19
11.4	Auswertung.....	20
12	Projektauswertung	20
13	Persönliches Fazit	20
14	Quellenverzeichnis	20

1 Einleitung

An der ETH Zürich Höggerberg untersucht die Gruppe um Prof. Dr. J. Bilgram das Wachstum von Xenon-Kristallen. Dazu werden von einem wachsenden Kristall eine Sequenz von digitalen Bildern erstellt.



Abbildung 1: Beispiel eines unaufbereiteten Kristallbildes

Diese Fotografien werden mit einem konturextrahierenden Verfahren in ein Schwarzweiss-Bild konvertiert. Im folgenden wird die Spitze jedes Wachstumszweiges bestimmt und diese im zeitlichen Verlauf ausgewertet. Herr H. Singer hatte nun die Idee, ein einzelnes Bild nach den prädestinierten Wachstumsrichtungen zu analysieren und diese Daten mit der dynamischen Auswertung zu korrelieren. Dies wurde der Gegenstand dieser Semesterarbeit.

2 Aufgabenstellung

Ziel dieser Semesterarbeit ist die Implementation eines für die Praxis im Labor geeigneten Algorithmus zur Skelettierung von binären Bildern (schwarz-weiß) dendritisch gewachsener Kristalle, um die Wachstumsrichtung der Seitenäste aus Einzelbildern zu bestimmen und mit dem bereits existierenden Verfahren zur Bestimmung der Wachstumsrichtung aus aufeinanderfolgenden Bildern zu korrelieren.

Nach unserer Einschätzung sollte die Lösung folgende Bedingungen annähern:

- ⇒ Eine möglichst aussagekräftige Bestimmung der Wachstumsrichtung anhand eines einzelnen Bildes.
- ⇒ Die Anwendung sollte in der Verarbeitungsumgebung von Herrn H. Singer lauffähig und integrierbar sein. Dies impliziert sowohl Schnittstellen wie auch Plattformen.
- ⇒ Die Verarbeitung sollte unabhängig von der Ausrichtung des Kristalls sein. D.h. das Resultat sollte nicht durch eine Drehung beeinflussbar sein.
- ⇒ Das Resultat sollte eine gewisse Stabilität haben. Ein verändertes Pixel darf die Auswertung nicht fundamental verändern.
- ⇒ Innerhalb technischer Grenzen sollte die Skalierung ebenfalls keinen direkten Einfluss auf die Auswertung haben.

3 Projektablauf

Aufgrund der Ungewissheit mit welchen Mitteln das Ziel erreichbar bez. eine Analyse überhaupt zuverlässig möglich ist, sind wir iterativ vorgegangen. Als erstes versuchten wir, Informationen über Bildskelettierung zu finden und wurden auf einen morphologischen Ansatz aufmerksam, den wir im Internet fanden. In diesem Dokument¹ wird beschrieben wie eine iterative morphologische Skelettierung implementiert werden kann (siehe morphologische Skelettierung). Mit diesem Algorithmus stiessen wir allerdings auf Probleme, verursacht durch diskontinuierliche Beschreibung der Kristallkontur.

Weil für die Skelettierung eines Bildes mehrere Möglichkeiten existieren, haben wir nach einer zweiten, vom morphologischen Ansatz unabhängigen, Variante gesucht. Fündig wurden wir im Buch "Digital Image Processing"² indem ein Ansatz zur Skelettierung beschreiben wird, welcher auf der Idee einer "Körper Mittellinie" (medial axis) basiert. Nach erfolgreicher Implementierung und anschliessendem Austesten an Kristallbildern, waren zwar gewisse Probleme des morphologischen Algorithmus nicht erkennbar, allerdings zeigten sich andere Probleme mit diesem Algorithmus (siehe medial axis transformation).

Bedingt dieser zum Teil grundsätzlichen Probleme einer pixelorientierten Auswertung (siehe Auswertung der pixelorientierten Verfahren) haben wir vor den Weihnachtsferien die Entscheidung gefällt, keine weiteren pixelorientierten Algorithmen zu implementieren, sondern uns von der Pixel-Ebene zu lösen und die Kristallkontur rein mathematisch zu beschreiben und auszuwerten. Durch die fortgeschrittene Zeit konzentrierten wir unsere Anstrengungen auf Lösungen mit Matlab, wobei wir von den bereits vorhandenen Funktionen profitierten (Beispiel "links division" welche überbestimmte Gleichungssysteme löst, indem nach dem quadratischen Fehler optimiert wird). Wir wollten uns auf die Evaluation eines Auswertungsansatzes und nicht dessen Implementierung in C++ konzentrieren.

Unser Ziel war es, in der verbleibenden Zeit möglichst konkret zeigen zu können, mit welchem Ansatz die gewünschten Informationen aus den Kristallbildern extrahiert werden können. Dieser Weg führte uns ins Themengebiet der Splines. Mit der Möglichkeit über eine bestimmte Menge von Stützpunkten eine möglichst genaue mathematische Funktion zu legen, wollten wir die Kristallkontur annähern.

Wenn wir eine solche Menge von approximierenden Funktionen von Teilbereichen der Kristallkontur hätten, so könnten wir aus dieser Grundlage die gewünschten Informationen berechnen. Die wichtigste Voraussetzung war, dass die einzelnen Funktionen den Verlauf der Kontur möglichst genau beschreiben, wobei die Funktionen zwischen zwei Seitenästen beginnen und enden sollen, diese Orte nannten wir "Mulden". Für diesen Zweck versuchten wir mittels dynamischem Programmieren eine Art

¹ morphologie.pdf, Seite 19,
URL: <http://www.tnt.uni-hannover.de/org/whois/stud/nikolaos/files/lit/bildverarbeitung/morphologie/morphologie.pdf>, (19.12.2002)

² Digital Image Processing; Rafael C. Gonzalez und Richard E. Woods;
Addison Wesley Publishing Company; 1992; S.491ff

"Viterbi"-Algorithmus zu implementieren, der über eine bestimmten Anzahl von Punkten (in unserem Fall die Kristallkontur) diese mit einer optimalen Anzahl von Splines n-ten Grades approximiert werden. In unserem Fall sollten es kubische Funktionen sein. Was "optimal" heisst, versuchten wir mit dem Ansatz der "erweiterten LaGrange-Methode" (siehe erweiterte LaGrange-Methode) umzusetzen. Dadurch erhofften wir uns, dass die Teilbereiche in den "Mulden" starten bez. enden.

Allerdings führte uns dieser Weg nicht zum erhofften Ziel. Wie beabsichtigt wird die Kontur zwar optimal approximiert, aber optimal heisst nicht zwingend, dass die Teilbereiche von "Mulde" zu "Mulde" gehen. Ausserdem war die benötigte Rechenzeit für eine solche Approximation von einem Bild mit ca. 4000 Punkten zu langsam. Diesen Umstand versuchten wir durch Aufteilung der Punktemenge in zwei Punkte Mengen, je eine auf beiden Seiten der Hauptspitze (siehe Spitzendetektion) die Rechenzeit zu verringern. Trotzdem ist dieser Algorithmus für unsere Zwecke nicht geeignet, wir mussten uns nach einem anderen Weg umschaun.

Das eigentliche Problem der Beschreibung der Kontur wollten wir in zwei Teilprobleme unterteilen. Erstens, finden der "Mulden" und zweitens Approximation von "Mulde" zu "Mulde". Das zweite Teilproblem war bereits gelöst, da die Funktion "SplineN" im Matlab eine Funktion n-ten Grades über eine Punkte Menge legt, wobei die Start und Endpunkte exakt übereinstimmen müssen. Das erste Teilproblem war schwieriger.

Zum Zeitpunkt des Abgabetermins dieser Arbeit, konnten wir nicht zuverlässig die "Mulden"-Punkte bestimmen.

4 Die morphologische Skelettierung

Werden die Spurpunkte der Zweigspitzen über die Konturen des Kristalls gelegt, so liegt eine intuitive Vermutung eines Zusammenhangs zwischen den zu Linien verbundenen Spurpunkten und der Skelettierung der Kontur nahe. Aus diesem Grund wurde bereits in der Definitionsphase des Projektes eine Anwendung der morphologischen Skelettierung favorisiert.

4.1 Ziele einer Skelettierung

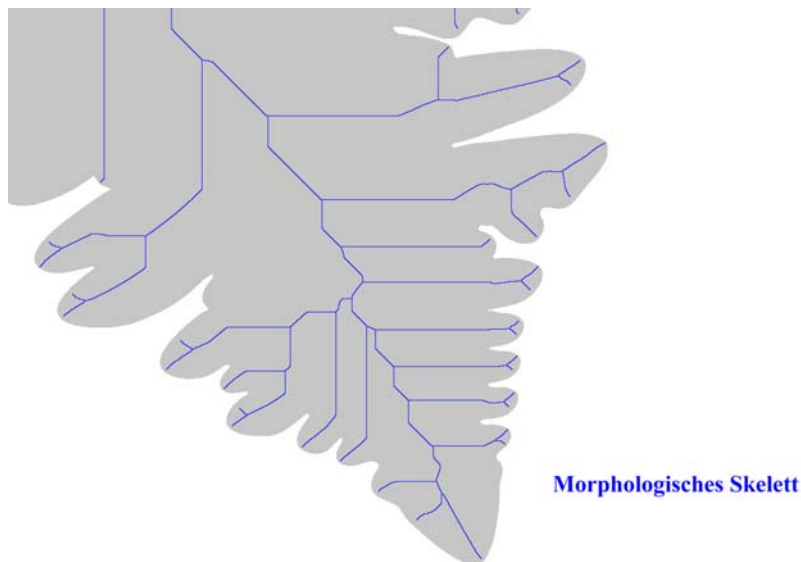
Die Idee der Skelettierung ist die Verdünnung ("thinning") von Segmenten auf eine Skelettlinie. Ein Skelett besteht idealerweise aus Linien mit einer Pixel Breite, liegt genau in der Mitte des Segments, widerspiegelt die ursprüngliche Form eines Bildes, verkürzt keine Linien, ist robust gegen Störungen und hat eine Konvergenz auf eine eindeutige Linie.

4.2 Der Verarbeitungsprozess einer morphologischen Skelettierung

Bei der morphologischen Skelettierung wird das Bild iterativ nach Punkten abge- sucht, welche Elemente von Kanten ausdünnender Flächen sind.

In der praktischen Implementation existieren zur Identifikation der zu entfernenden Bildelemente acht verschiedene Zustandsmöglichkeiten (Masken), welche nicht relevante Elemente ("don't care") enthalten können. Für den Verarbeitungsprozess entscheidend ist, dass nach einer Übereinstimmung mit einer Maske dieses Element nicht sofort entfernt wird, sondern zur Evaluation seiner Nachbarelemente noch seinen ursprünglichen Zustand bis zum Ende der Überprüfung aller Bildpunkte behält und erst anschliessend den Statuswechsel vollzieht. Anderenfalls kann der Algorithmus Kanten "vor-sich-her" schieben und die Skelettlinie wird nicht in der Mitte des Segments zu liegen kommen. Ferner ist zu beachten, dass die Masken nach Möglichkeit die gleiche Priorität haben, um keine unausgeglichene Ausprägungsrichtung zu erhalten.

Der Verarbeitungsprozess ist abgeschlossen, wenn mit dem oben beschriebenen Verfahren keine Pixel mehr lokalisiert werden, welche gelöscht werden können.



den1_085

Abbildung 2: Beispiel eines skelettierten Kristallbildes

4.3 Grenzen der morphologischen Skelettierung

Es existieren technische Grenzen, welche eine Abweichung von der oben beschriebenen Zielsetzung erzwingen. So ist beispielsweise durch die Einschränkungen des Pixelrasters die Mittellinie eines Segments mit einer Breite von einer geradzahigen Pixel Anzahl, nicht exakt in der Mitte sondern dem Raster entsprechend versetzt. Am gravierendsten ist dies bei einer Breite von zwei Pixeln, da dann nur eine gelöscht werden kann und die verbleibende als Skelettlinie übrig bleibt.

Aber von diesen technischen Einschränkungen abgesehen, ergeben sich weitere für diese Anwendung stark negative Auswirkungen durch die Diskretisierung der Bildelemente durch die pixelbezogene Darstellung.

Die zwei Effekte, welche die Aussagekraft der Bildanalyse am meisten beschneiden, sind die unvermeidbare Priorisierung von Richtungen, das bedeutet auch eine uner-

wünschte, starke Abhängigkeit von der Ausrichtung, bzw. Drehung. Zweitens die Auswirkungen von Unregelmässigkeiten der Kontur. Hat ein einzelnes schwarzes Pixel in seiner Umgebung der acht Nachbarn zum Beispiel nur zwei andere schwarze Pixel, so entsteht von diesem Punkt aus zwingend eine ganze Skelettierungslinie. Kleine Änderungen der Ausgangslage können so einen starken Einfluss auf das Endresultat haben. Dies entspricht nicht den Zielsetzungen.

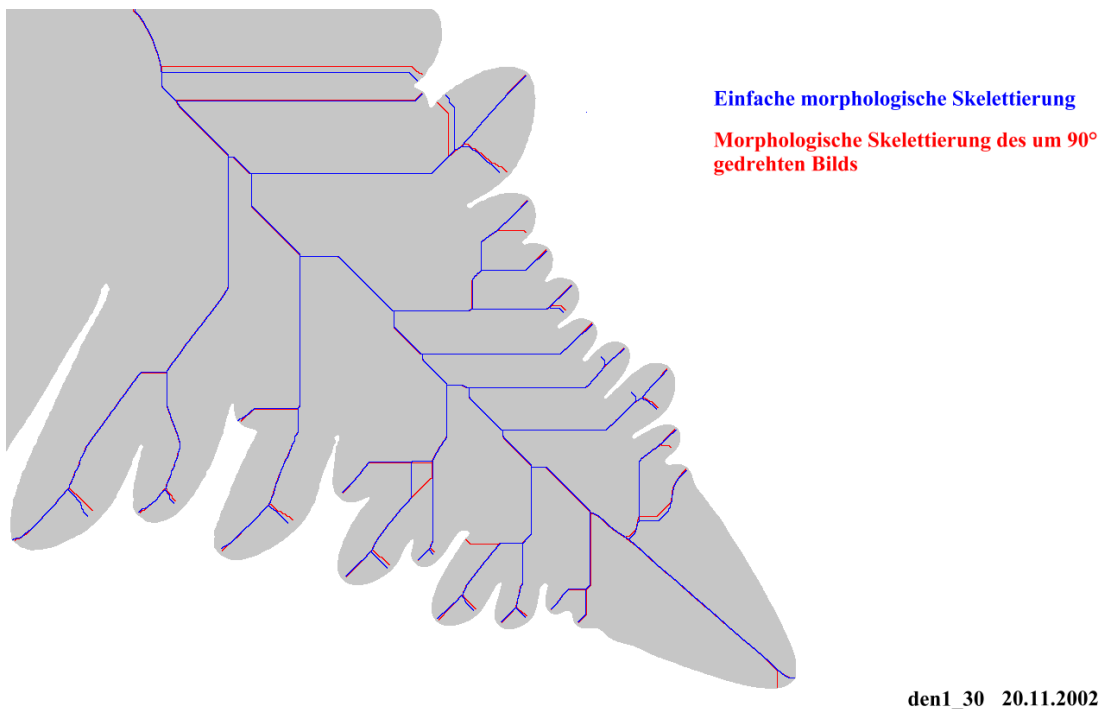


Abbildung 3: Einfluss der Drehung bei der morphologischen Skelettierung

5 Die Medial Axis Transformation (MAT)

Als Alternative zum ersten Algorithmus, der auf der morphologischen Skelettierung basiert, wurden wir auf einen Algorithmus aus dem Buch "Digital Image Processing"³ aufmerksam. Es handelte sich dabei um ein Verfahren, bei welchem eine Körpermitellinie gesucht wird. Gleich wie die morphologische Skelettierung ist der MAT Algorithmus iterativ, besteht aber aus zwei unabhängigen Teilschritten. Die Nachbarschaft eines Punktes p1 wird folgendermassen definiert:

p9	p2	p3
p8	p1	p4
p7	p6	p5

³ Digital Image Processing; Rafael C. Gonzalez und Richard E. Woods; Addison Wesley Publishing Company; 1992; S.491ff

Zum charakterisieren der Umgebung eines Punktes wird die Anzahl "Nicht-Null Nachbarn" $N(p_1)$ und die Anzahl "0-1" Übergänge $S(p_1)$ - beginnend bei p_2 im Uhrzeigersinn - benutzt.

Eine Iteration setzt sich aus zwei Teilen Step1 und Step2 zusammen, welche wiederum aus zwei Teilschritten, dem Markieren der zu löschenden Punkten und dem Löschen der markierten Punkte, bestehen. Um Änderungen an der Struktur der Daten zu vermeiden, werden zuerst alle entsprechenden Punkte als "zu löschen" markiert und erst in einem zweiten Schritt "gelöscht". Als Kriterien zum Löschen der Punkte gelten folgende Bedingungen, wobei der Punkt als "zu löschen" markiert wird wenn alle Bedingungen erfüllt sind:

Step 1	Step 2
(a) $2 \leq N(p_1) \leq 6$	(e) $2 \leq N(p_1) \leq 6$
(b) $S(p_1) = 1$	(f) $S(p_1) = 1$
(c) $p_2 * p_4 * p_6 = 0$	(g) $p_2 * p_4 * p_8 = 0$
(d) $p_4 * p_6 * p_8 = 0$	(h) $p_2 * p_6 * p_8 = 0$

Diese zwei Schritte werden so lange durchgeführt, bis kein Punkt mehr gelöscht werden kann.

Das Resultat ist eine Körpermittellinie, welche die Ränder des Körpers nicht berührt.



Abbildung 4: MAT Beispiel
Grau: Ursprungsbild
Schwarz: Transformiertes Bild

Beispiel eines Transformatierten Bildes eines Kristalls (den1_34.fillcod):

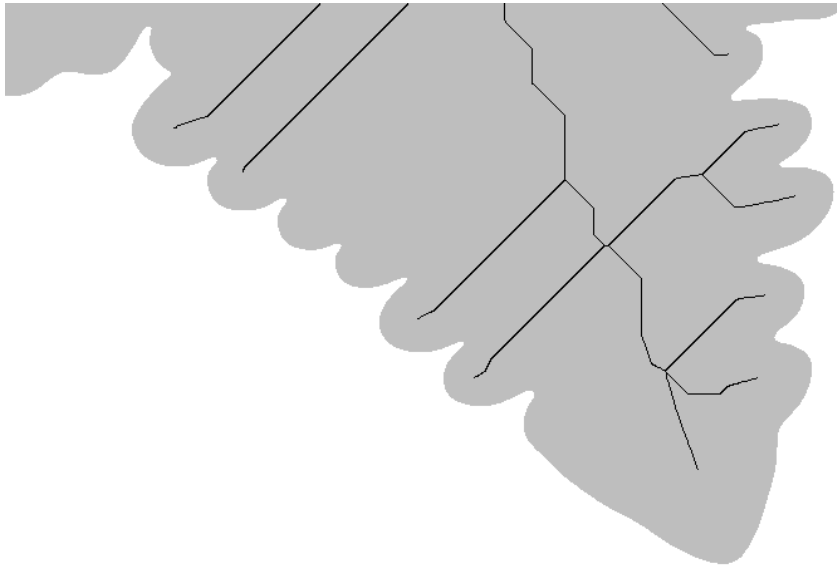


Abbildung 5: den1_34.fillcod
Grau: Ursprungsbild
Schwarz: Transformatiertes Bild

Auffallend ist, dass zwei "Äste" des Kristall-Skeletts fehlen. Der Grund konnte bisher nicht schlüssig nachgewiesen werden. Aber wenn man die Transformation schrittweise betrachtet, so erkennt man, dass sich während der Verdünnung markante rechte Winkel bilden.

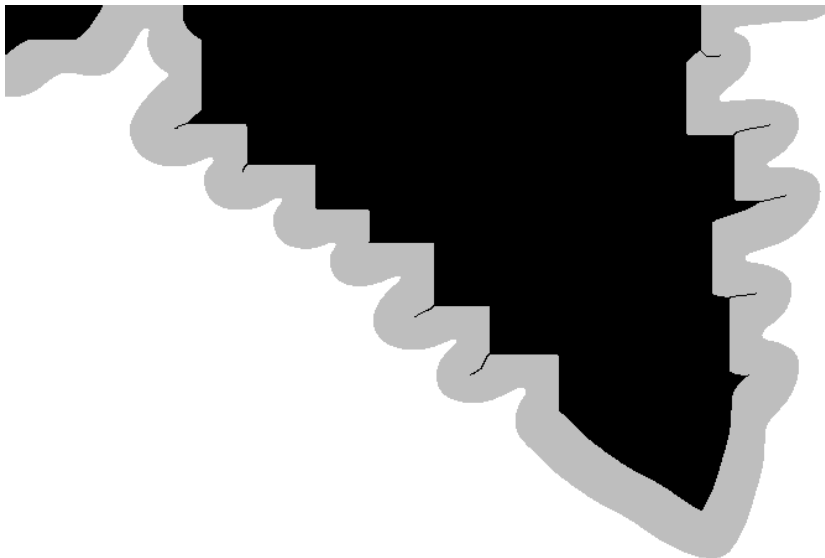


Abbildung 6: MAT Zwischenbild

Wahrscheinlich werden diese Ecken ähnlich den Ecken eines Rechtecks behandelt. Weil nun eine Eigenschaft des MAT Algorithmus ist, dass die Körpermittellinie nicht die Körperländer berührt, verschwindet dieser "Ast". Das würde zwar das mögliche Verschwinden erklären, aber nicht, warum ein Teil der Spitzen trotz des sichtbaren Winkels erkannt werden und andere nicht. Auch kleinere Spitzen, die fast nur im Ansatz zu erkennen sind, gehen durch diese Transformation verloren.

6 Auswertung der pixelorientierten Verfahren

Im direkten Vergleich der beiden pixelbasierenden Verarbeitungen ist zu erkennen, dass der MAT-Algorithmus weniger Verästelungen in den Spitzen hat.

Es können aber unter ungünstigen Umständen bei der gleichen Ausgangslage, ganze Äste verloren gehen, welche mittels der morphologischen Skelettierung gefunden werden.

Beide Verarbeitungen werden durch die Abbildung in einem Pixelfeld beeinflusst. Dies wurde durch die Drehung des Bilds innerhalb des Pixelrasters nachgewiesen.

Es konnte beim morphologischen skelettierten Bild beobachtet werden, dass sich zwei in der Ausgangslage nur durch eine Drehung im Ursprungsbild unterscheidende Skelette ganze Äste als Differenzen aufwiesen. Der weit im Inneren des Kristalls liegende Teil des Skeletts ist bevorzugt in 45° Schritten bezogen auf den Pixelraster ausgerichtet.

Für dieses Einsatzgebiet ist der innere Teil des Skeletts weniger relevant und somit ist die 45°-Diskretisierung der Linien nur teilweise begrenzend.

Der Verlust von Seitenästen ist für diese Anwendung unerwünscht.

7 Splines

Unter einer Spline verstehen wir eine Kurve n-ten Grades, welche eine gegebene Punktmenge approximiert. Um eine Kontur mittels Funktionen n-ten Grades zu beschreiben, teilen wir die X- und Y-Koordinaten in zwei Funktionen auf, wobei beide Funktionen von der gleichen Variablen, in unserem Fall S, abhängig ist:

$$X = f_x(S)$$

$$Y = f_y(S)$$

Die Funktion f(S) ist eine Funktion n-ten Grades mit (n+1) Koeffizienten, eine Spline n-ten Grades in Abhängigkeit von S. Die Variable S ist die unabhängige für die Funktionen der X- und Y-Koeffizienten. Wir bezeichnen diese als "Spur". Diese Spur ist der summierte Abstand vom Startpunkt bis zum n-ten Punkt:

$$S = f(n) = 0 + \sum_{n=2}^N \text{Abstand}(P(n-1), P(n))$$

$$P(n) = \{X_n, Y_n\}$$

Abstand(P_1, P_2): Abstand zweier Punkte in einer Ebene

Mit dieser Funktion lässt sich einer Menge von Punkten, gegeben in X- und Y-Koordinaten, einen Wert S zuordnen, welcher als Argument, die zwei Funktionen für die X- und Y-Koordinaten verwendet wird. Wichtig ist, dass die Menge der Punkte P in der Reihenfolge der Kontur geordnet sind (diese Voraussetzung wurde durch die Rohdaten der ETH bereits erfüllt und musste nicht weiter beachtet werden).

Eine Spline ohne weitere Nebenbedingungen konnte mit einem relativ einfachen Gleichungssystem beschrieben werden.

Für unsere Zwecke stellten wir für die Spline noch eine zentrale Nebenbedingung, die Anfangs- und Endpunkte mussten exakt erfüllt werden, d.h. in diesen Punkten fehlerlos sein. Diese Nebenbedingung führte zu einem modifizierten Gleichungssystem. Die Funktion SplineN() implementiert genau diese erweiterte Eigenschaft.

8 Dynamisches Programmieren

Mittels der SplineN() Funktion waren wir im Stande eine Menge Punkte mit Anfangs und Endpunkt durch eine Spline zu approximieren. Der nächste Schritt war eine Menge von Punkten in eine optimale Anzahl von Splines n-ten Grades zu unterteilen.

Wir bedienten uns einer Idee aus der Mobil-Kommunikation, dem "Viterbi"-Algorithmus, welcher die Eigenschaft hat, aus einer Folge von Kodewörtern die optimale Kodewortfolge zu extrahieren (wobei optimal im Sinne der Mobil-Kommunikation den wenigsten Bitfehler entspricht). Die Implementation stützt sich allerdings nur auf der Idee ab und ist nicht von bekannten Algorithmen abgeleitet oder abgeschrieben, Ähnlichkeiten mit anderen Implementationen sind rein zufällig.

Zur Definition des Optimums bedienten wir uns der Idee der erweiterten LaGrange-Methode (siehe folge Kapitel). Zur Auswertung des Optimums wird ein Faktor L verwendet, welcher als "Kostenfaktor pro Spline" Interpretiert werden kann. Dies führte zur Funktion Trellis().

Da die Berechnung eines Netzes, bestehend aus einzelnen Trellis - wobei ein Trellis eine Folge von Splines ist - auch auf 2 GHz Rechnern beträchtliche Zeit beanspruchte (Komplexität von etwa $0.5 \cdot N^2$, N die Anzahl Punkte der zu approximierenden Kontur oder Teilkontur) und wir nicht für verschiedenen Kostenfaktoren L jedes mal das gesamte Netz neu berechnen wollten, modifizierten wir die Funktion Trellis() und teilten diese in zwei Funktionen auf. In eine Funktion MakeNet(), welche das Netz erstellt und eine Funktion FindTrellis(), welche für einen gegebenen Kostenfaktor L eine optimale Auswahl von aufeinanderfolgenden Splines aus dem Netz herausliest. Erste Resultate entsprachen weitgehend den Erwartungen, allerdings zeigte sich beim Durchrechnen mit verschiedenen Kostenfaktoren, dass der Algorithmus wie erwartet die Kontur optimal approximiert aber einzelne Spline's nicht, wie erhofft, in den "Mulden" beginnen bez. enden, sondern mitten in Seitenspitzen.

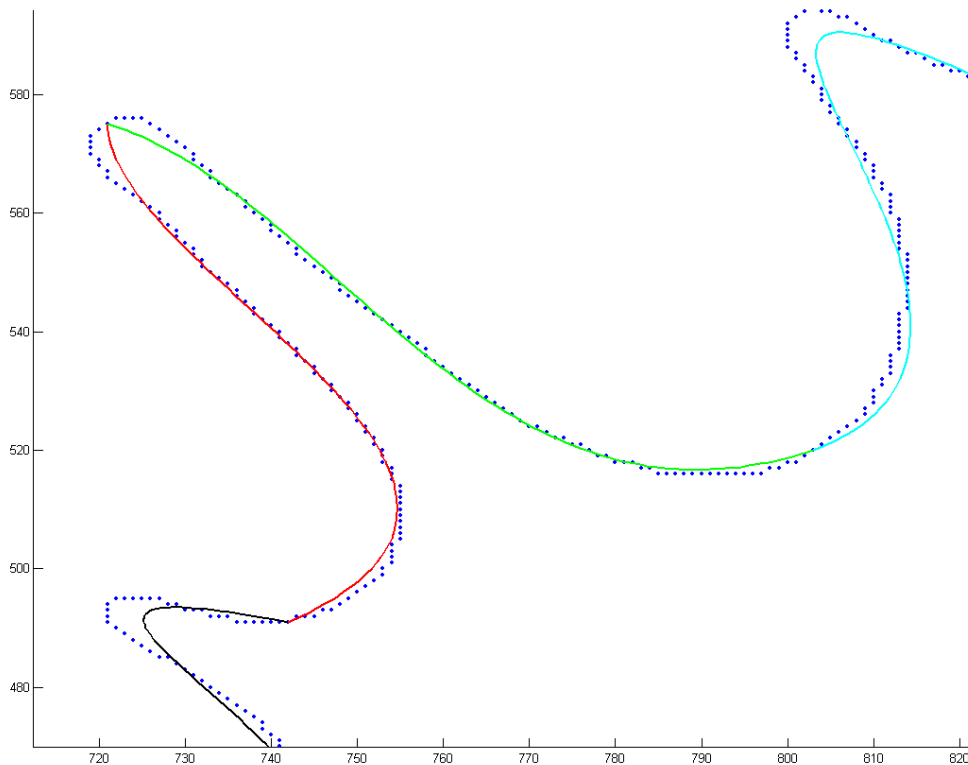


Abbildung 7: Approximation der Kontur mit Splines

9 Erweiterte LaGrange-Methode

"modified lagrange multiplier method"

Diesen Ansatz wählten wir aufgrund einer Besprechung mit Herrn G. Schuster, welcher uns die nötigen Einzelheiten erklärte. Dabei wird versucht, eine Charakteristische Zahl J zu optimieren. Diese Zahl J setzt sich zusammen aus dem Fehler, verursacht durch die Approximation, und den "Kosten" für einen Teilbereich gewichtet mit einem Kostenfaktor L (ursprünglich Lambda, wobei L für die Matlab Funktionen verwendet wurde).

In unserer Implementierung legten wir die Kosten für einen Teilbereich (eine Spline) als 1 fest, wobei die Matlab Funktion `Cost()` so geschrieben wurde, dass man diese Kosten auch von den Punkten abhängig machen kann.

Grundsätzlich gilt der Zusammenhang, je grösser der Kostenfaktor L desto weniger Teilbereiche.

Wir erhofften uns, dass die Funktion Anzahl Teilbereiche in Abhängigkeit des Kostenfaktors eine treppenartige konvexe Kurve ergibt, welche in einem gewissen Bereich des Kostenfaktors eine markante Stufe in der Anzahl der Teilbereiche ergibt. Die Anzahl der Teilbereiche sollte in diesem Bereich der Anzahl der Spitzen des Kristallbildes entsprechen.

Allerdings war das Resultat dieser Untersuchung nicht so eindeutig wie wir das erhofften.

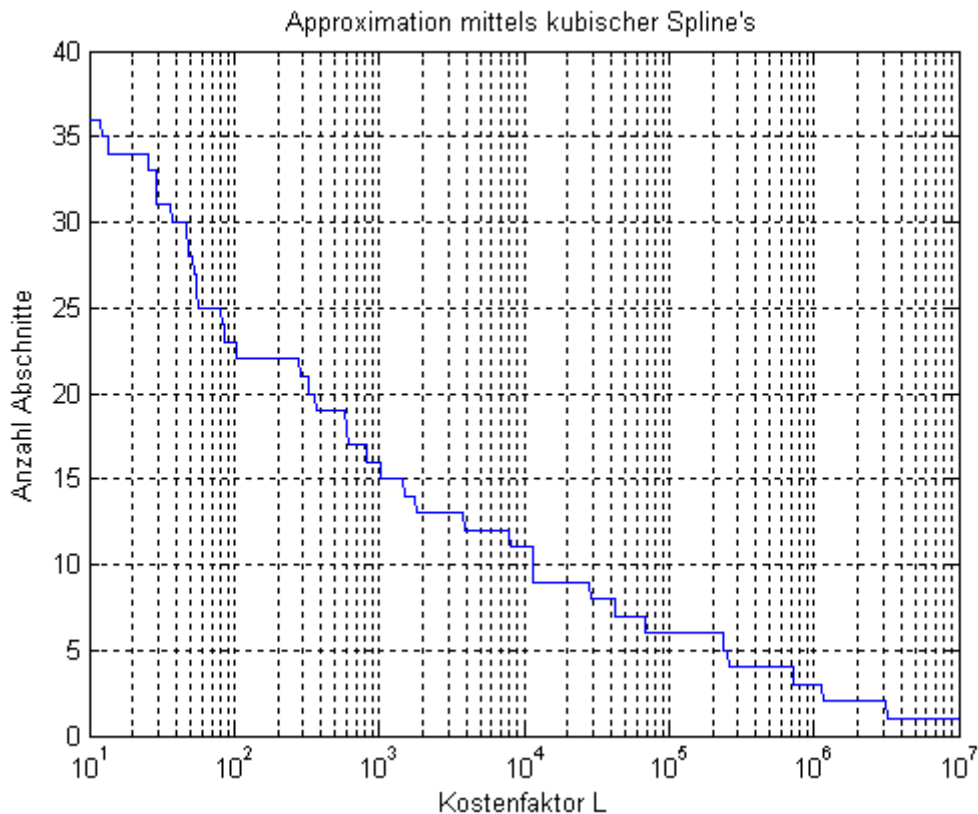


Abbildung 8: Anzahl kubischer Splines in Abhängigkeit des Kostenfaktors L

10 Detektion der Hauptspitze

Als Ausgangslage liegt im Matlab die Kontur in einem fortlaufenden, d.h. hier dem Konturverlauf entsprechend sortierten zweispaltigen Array vor.

In einem iterativen Prozess wird eine Parabel über die Kontur approximiert. An den Anfangs- und Endpunkt dieser Parabel wird eine Tangente gelegt, welche sich in einem Punkt schneiden muss. Die Winkelhalbierende dieser Tangenten wird wieder mit der Kontur geschnitten, d.h. es wird der Punkt bestimmt, welcher den kleinsten Abstand zu dieser Geraden aufweist. Von diesem Punkt aus kann die Auswahl des neuen eingegrenzten Bereichs wiederholt werden.

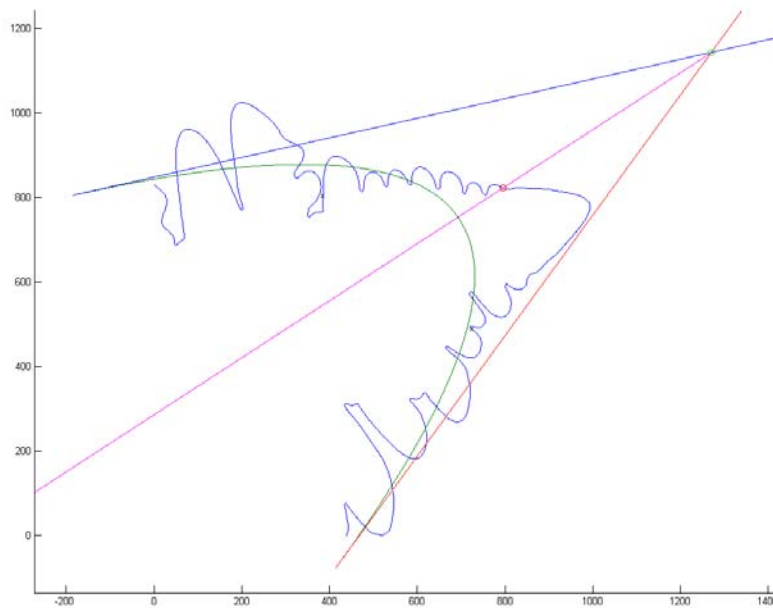


Abbildung 9: Beispiel der Spitzendetektion (erste Iteration)

Die Schwierigkeit liegt in der Bestimmung des neu relevanten Bereichs. Nicht bewährt haben sich eine proportionale Kürzung der durch den Punkt abgegrenzten Bereiche, sowie vom Ursprungsarray ausgehend, eine kleiner werdende absolute Punktezahl.

Die besten Resultate wurden mit dem Kriterium des geometrischen Abstands zu diesem Schnittpunkt erreicht (Kreisausschnitt), welcher schrittweise kleiner angesetzt würde.

Ein weiteres Optimierungsproblem stellt die auf einen Schritt bezogene Verkleinerungsrate dar. Wird der Bereich zu schnell eingeeengt, kann es je nach geometrischer Form eine Fehlinterpretation geben und statt der Hauptspitze wird eine Spitze eines Nebenasts gefunden. Kleinere Schritte erfordern aber mehr Durchläufe für die gleiche Genauigkeit.

Am Ende muss der relative Index wieder in den Index des Ausgangsarray umgerechnet werden.

11 Weiterführende Überlegungen

11.1 Physikalische Ansätze

Aus dem physikalischen Ursprung dieser Bilder legt sich eine Implementation, welche sich auf die Diffusionsgleichung stützt nahe. Jedoch ist dies aufgrund der vielen und nicht immer einfach zu bestimmenden Parametern sehr komplex.

Es stellt sich nun die Frage, ob eine gute Näherung bereits erreicht werden kann, wenn mit einem einfacheren Modell dieser Sachverhalts nachgebildet wird.

Der uns am nächsten erscheinende erste Schritt wäre eine Betrachtung des möglichen Einfallswinkels eines Moleküls, welches in die Kontur einwächst. Entsprechend einer der statistischen Weglängen eines Moleküls zwischen zwei Zusammenstößen gewichteten Integration des Winkels und des Herkunftsraumes, sollte eine erste Aussage über die Auftreff- und somit über die Wachstumswahrscheinlichkeit, beziehungsweise der Wachstumsgeschwindigkeit des Punktes auf der Kontur aussagen.

Die Aussagekraft dieser Methode würde aber - nach unserer Einschätzung - markant verbessert werden, wenn diese Betrachtung zu einem aus der Kontur abgeleiteten Gradienten erweitert wird.

Diese Verfahren sollten nur zur Bestimmung der Relation der Zunahme angewendet werden. Die absolute Flächenzunahme kann aufgrund bekannter Gesetzmässigkeiten abgeschätzt werden.

In Anlehnung an die Modelle der Schneekristallentstehung⁴ könnte man auch die molekulare Ausrichtung des Kristalls in einer Absorbtionswahrscheinlichkeit berücksichtigen. Wir denken hier beispielsweise an die Modelle der QLL-Theorie.

Hier sind aber auch bereits Grenzen sichtbar. In wieweit eine zweidimensionale Projektion eines dreidimensionalen Kristalls unter einer rein retrospektiven und sehr eingeschränkten Analyse möglich ist, bleibt abzuklären. Ausserdem verfügen wir nicht über die nötige Fachkompetenz um abwägen zu können, welche Theorien - z.B. des Wasser - Analogien zum Element Xenon haben.

Neben dem limitierenden Zeitfaktor war auch ein Grund, weshalb wir nicht die hauptsächlich physikalischen Ansätze verfolgt haben, dass wir unsere gelernten Ansätze eines Elektrotechnikstudenten in diese Arbeit einbringen wollten.

11.2 Ansatz Fouriertransformation

Eine der wichtigsten Theorien für die Signalverarbeitung ist die Fouriertransformation. Sie lässt sich auch auf dieses Problem anwenden. Werden die Pixelkoordinaten als komplexe Signalwerte interpretiert, so lässt sich diese Information in einem Frequenzspektrum analysieren. Ein Löschen der höheren Frequenzanteile führt nach einer Rücktransformation zu einer gröberen Struktur. Jedoch erschien uns diese Verfahrensansätze als wenig erfolgsversprechend. Und wurden nicht weiter verfolgt.

⁴ Yoshinori Furukawa, Chemie in unserer Zeit, 31. Jahrgang 1997, Nr.2

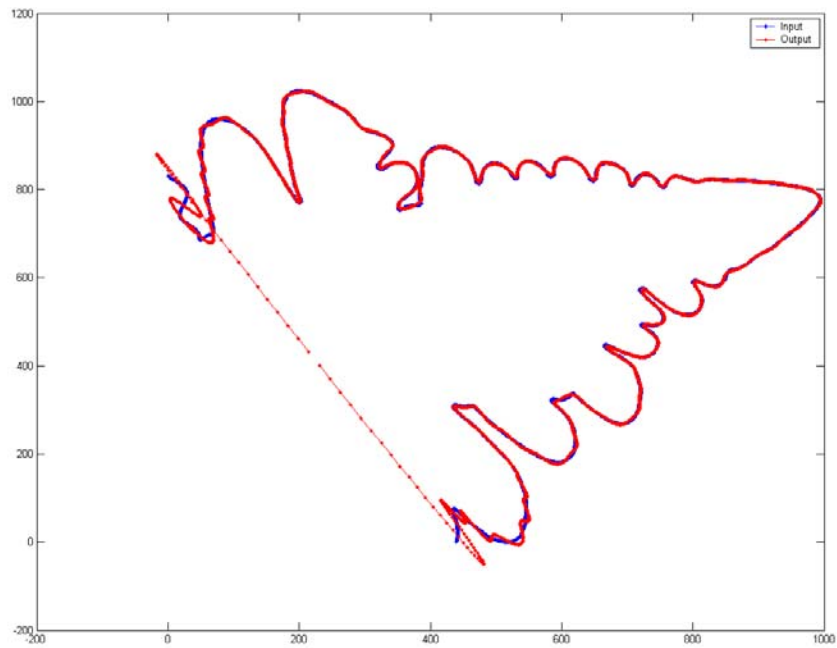


Abbildung 10: Bild vor und nach Fouriertransformation

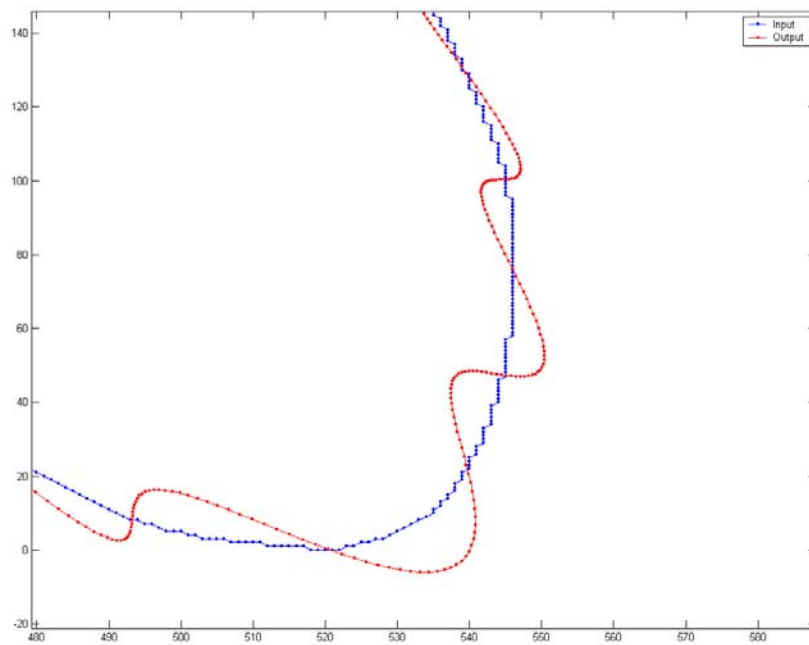


Abbildung 11: Ausschnitt der Fouriertransformation aus Abbildung 10

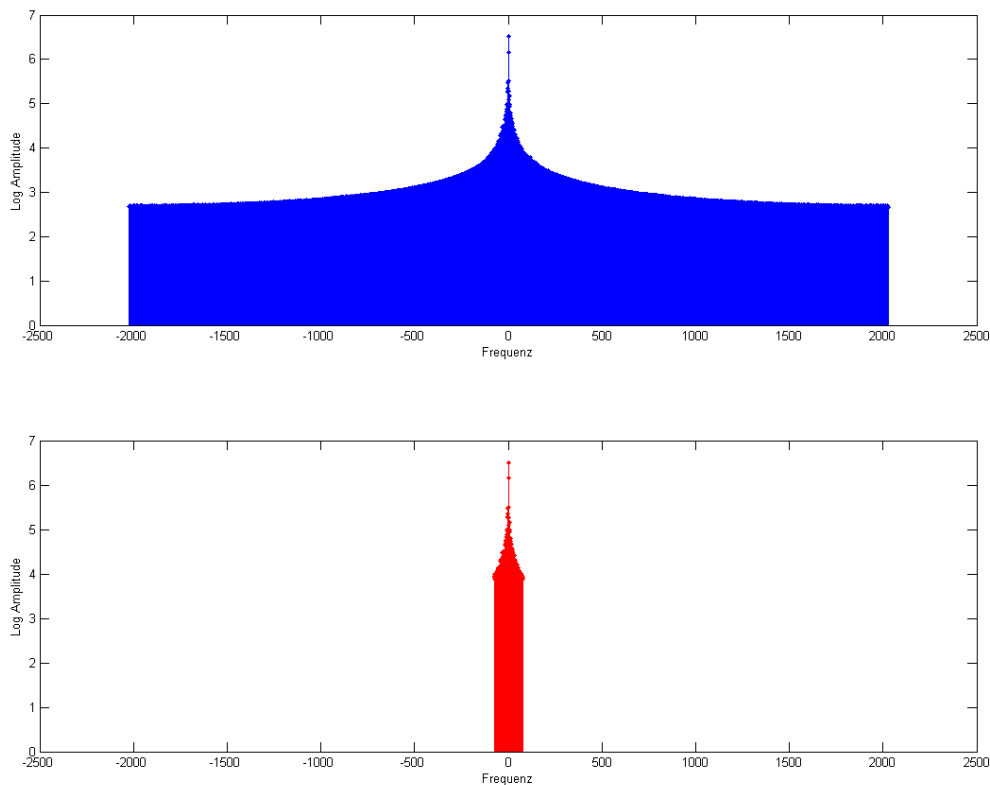


Abbildung 12: Vergleich der Frequenzspektren

11.3 Ansatz zur Seitenastbestimmung

Eine gute Auswertung der Seitenäste erscheint durch folgendes Konzept erreichbar:

Man bestimmt die Winkel, welche die Punkte der Kontur bilden. Ähnlich einer numerischen Ableitung, jedoch muss nicht zwangsweise nur der Winkel benachbarter Punkte betrachtet werden. Bei unserer Auflösung und den Proportionen der Seitenäste des abgebildeten Kristalls erscheint uns der Winkel zwischen dem aktuellen Punkt mit dem dritten vorangehenden und dem dritten nachfolgenden Punkt als angebracht.

Diese Winkelreihe könnte mit einem Tiefpass gefiltert werden und anschliessend nach Extrema untersucht werden. Wobei eine logische Abfolge zwischen einem positiven und einem negativen Extrema zu erwarten ist, weil ein vollständiger Seitenast mit einer Aussenkurve beginnt, mindestens eine Innenkurve aufweist und in einer weiteren Aussenkurve endet. Dieses Verfahren sollte die Aufteilung in grosse Seitenäste ermöglichen.

11.4 Auswertung

Die genau erörterten Äste könnten mit einem Spline, z.B. einer Parabel approximiert werden, welche eine klare mathematische Definition der Mittellinie aufweist. Aus diesen Werten sollten die Richtungsvektoren ersichtlich sein, welche unter Umständen mit den oben beschriebenen Verfahren (Physikalische Ansätze) betragsmässig gewichtet werden könnten.

Zur Bewertung der Aussagekraft können diese Wachstumsvektoren in eine Textdatei exportiert werden und mit den Resultaten der dynamischen Analyse bewertet werden.

12 Projektauswertung

Diese Semesterarbeit hat gezeigt, dass auf der Ebene von Pixelorientierte Algorithmen keine zuverlässigen Vorhersagen für Wachstumstendenzen einzelner Seitenäste aufgrund von Kristallaufnahmen gemacht werden können. Desweiteren haben wir Möglichkeiten gesucht, die Kristallkontur mathematisch zu beschreiben, um uns von der pixelorientierten Analyse zu entfernen und die weiteren Auswertungen auf der Basis von mathematischen Funktionen aufzubauen. Dieser Schritt konnte nicht komplett vollzogen werden. Wir sind auf vielversprechende Ansätze gestossen, die es durchaus Wert wären, diese weiter zu verfolgen. Wir sehen unsere Arbeit als Vorarbeit für weiterführende Arbeiten zu diesem Thema.

13 Persönliches Fazit

Wir haben Einblick in ein Thema erhalten, zu welchem es laufende Forschungsprojekte gibt. Durch unsere von der Elektrotechnik kommende "art"-fremde Sichtweise auf dieses Themengebiet, hofften wir neue Impulse zu bringen und die Probleme mal aus einem anderen Gesichtspunkt zu betrachten. Zudem wollten wir bewährte Lösungsansätze aus der Elektrotechnik auf diese Problemstellung übertragen. Auch wenn wir nicht den durchschlagenden Erfolg erzielten, so war es für uns eine nicht minder erfolgreiche Zeit.

14 Quellenverzeichnis

- morphologie.pdf
URL: <http://www.tnt.uni-hannover.de/org/whois/stud/nikolaos/files/lit/bildverarbeitung/morphologie/morphologie.pdf>
(19.12.2002)
- Digital Image Processing; Rafael C.Gonzalez und Richard E.Woods; Addison Wesley Publishing Company;1992
- Yoshinori Furukawa, Chemie in unserer Zeit, 31.Jahrgang 1997, Nr.2