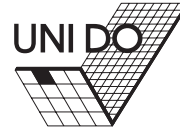


Projektgruppe 484

3D - AugTracer

**LEHRSTUHL 7
FACHBEREICH INFORMATIK
UNIVERSITÄT DORTMUND**

30. November 2006



- 1. Betreuer: Prof. Dr. Heinrich Müller**
- 2. Betreuer: Dipl. Inform. Frank Weichert**

Vorwort

Ein Endbericht ist der Abschluss einer jeden Projektgruppe. Die Teilnahme an einer Projektgruppe ist Bestandteil des Informatik-Hauptstudiums an der Universität Dortmund. Die Projektgruppe erstreckt sich über ein Jahr und hat zum Ziel, typische Arbeitsmethoden der Informatik bei der Bearbeitung umfangreicher Problemstellungen zu vermitteln.

Das in der PG 484 im Wintersemester 05/06 und Sommersemester 06 entworfene Programm wurde „**3D-AugTracer**“ getauft.

An dieser Stelle wollen die Teilnehmer dieser Projektgruppe dem Lehrstuhl für Mechanische Verfahrenstechnik (Fachbereich Bio- und Chemieingenieurwesen) für die gute und produktive Zusammenarbeit danken. Weiterhin danken sie allen Mitarbeitern am Lehrstuhl VII, Graphische Systeme, für ihre Unterstützung, insbesondere den Projektgruppenbetreuern Prof. Dr. Heinrich Müller und Dipl. Inform. Frank Weichert für ihre umfangreiche Hilfestellungen und ihre wertvollen Anregungen.

Dortmund, September 2006

Christian Asche

Björn Bönninghoff

Dominik Bourquain

Hong Chen

Carsten Dierig

Andreas Fischbach

Nicole Gimbel

Benjamin Jungbluth

Anne Kunert

Adrian Schyja

Patrick Strootmann

Jérôme Triebel

Maurice Ullrich

Mathematische Notation

Im Folgenden werden die unten aufgeführten mathematischen Notationen benutzt.

- Matrizen werden in Großbuchstaben und fett dargestellt:

$$\mathbf{A} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix}$$

\mathbf{A}^T bezeichnet die transponierte \mathbf{A} Matrix.

\mathbf{A}^{-1} bezeichnet die invertierte \mathbf{A} Matrix.

\mathbf{A}^{-T} bezeichnet die Inverse der transponierten \mathbf{A} Matrix.

- Vektoren werden in Kleinbuchstaben und dick dargestellt.
Im Text werden Spaltenvektoren ebenso wie Punkte der Übersichtlichkeit halber als transponierte Zeilenvektoren geschrieben: $\mathbf{p} = (x, y, z)^T$
- $\mathbf{t} \times \mathbf{z}$ bezeichnet das Kreuzprodukt der beiden Vektoren \mathbf{t} und \mathbf{z} .
- $\|\mathbf{v}\|$ bezeichnet die euklidische Norm des Vektors \mathbf{v} .

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgabenstellung an die Projektgruppe	1
1.2	Versuchsaufbau	2
1.3	Stand der Forschung	4
1.4	Aufbau des Endberichts	4
2	Grundlagen der Verfahrenstechnik - Zerstäubung	7
2.1	Übersicht	7
2.2	Grundlagen	7
2.2.1	Aufgaben und Einsatzgebiete	8
2.2.2	Aufbau und Funktionsweise einer Hohlkegeldüse	9
2.3	Strömungsverhalten	10
2.3.1	Struktur und Zerfallsprozess der Lamelle	10
2.3.2	Merkmale für die charakteristische Beschreibung einer Lamelle	11
2.3.3	Beispielhafte Untersuchung unterschiedlicher Stoffe	12
2.4	Interne Messverfahren	12
2.5	Externe Messverfahren	13
2.5.1	Particle Image Velocimetry	14
3	Bildverbesserung	15
3.1	Einleitung	15
3.2	Kamera-Kalibrierung	15
3.2.1	Einleitung	15
3.2.2	Realisierung	17
3.2.3	Anmerkungen	19
3.3	Dewarping	20
3.3.1	Einleitung	20
3.3.2	Vorgehensweise	20
3.3.3	Algorithmus	21
3.4	Denoising	22
3.4.1	Einleitung	22
3.4.2	Denoising-Algorithmus	22
3.4.3	Parameter	24
3.4.4	Anmerkungen	25
3.5	Fazit	25
4	Segmentierung	27
4.1	Einleitung	28
4.2	Lamellen-Segmentierung	28
4.2.1	Einleitung	28
4.2.2	Histogrammausgleich	28

4.2.3	Sobeloperator	31
4.2.4	Binarisierung	31
4.2.5	Schwellenwertverfahren nach Otsu	32
4.2.6	Template Matching	33
4.2.7	Kantentracking	34
4.2.8	Fehldetektionen	34
4.2.9	Ausgleichsgerade	36
4.2.10	Sprühwinkelbestimmung	39
4.2.11	Lamellenabriss	41
4.2.12	Zusammenfassung, Ausblick	41
4.3	Frequenzanalyse	42
4.3.1	Einleitung	42
4.3.2	Auswertung der Silhouette	42
4.3.3	Amplitudendarstellung	43
4.3.4	Metrik	45
4.3.5	Gemittelte Amplitudendarstellung	45
4.3.6	Eingesetzte Analyse-Methoden	46
4.3.7	Fourier-Transformation	46
4.3.8	Gefensterter Fourier-Transformation	46
4.3.9	Wavelet-Analyse	48
4.3.10	Frequenz-Spektrogramm	50
4.3.11	Wavelet-Spektrogramm	51
4.3.12	Amplitudenverdoppelung	52
4.3.13	Amplitudenvervierfachung	53
4.3.14	Kumulierte Amplituden	54
4.3.15	Kumulierte Frequenzen	55
4.4	Tropfenanalyse	56
4.4.1	Einleitung	56
4.4.2	Hough-Transformation	56
4.4.3	Template Matching	57
4.4.4	Tropfenerkennung im Stereo-Modus	57
4.4.5	Formfaktoranalyse	58
4.4.6	Effizienzsteigerungen	59
4.4.7	Weitere Ansätze	59
4.4.8	Ergebnisse	60
4.5	Wavelet-Segmentierung	61
4.5.1	Grundlagen	61
4.5.2	Verfahren zur Segmentierung	63
4.5.3	Energieberechnung	63
4.5.4	Hölder-Exponent	63
4.5.5	Scanline Verfahren	64
4.6	Haralick	64
4.7	Fazit	66
5	Tracking	69
5.1	Einleitung	69
5.2	KLT-Tracking	70
5.2.1	Triangulierung	71
5.3	Blockmatching	72
5.4	Partikel-Tracking	73
5.4.1	Initiale Ansätze	73
5.4.2	Berechnung des Trackings	74
5.4.3	Erstellen von synthetischen Partikeln	75

5.4.4	Berechnung der Wavelet-Transformation	75
5.4.5	Vorbereiten des nächsten Iterationsschritts	77
5.4.6	Skelettierung und Vektorfeldbestimmung	78
5.5	Fazit	79
6	3D-Rekonstruktion	81
6.1	Einleitung	81
6.2	Berechnung der Fundamentalmatrix	81
6.3	Das Korrespondenzproblem	83
6.4	Berechnung der 3D Punkte	84
6.5	Anmerkungen	85
6.6	Metrik	85
6.7	Fazit	86
7	Visualisierung	89
7.1	Einleitung	89
7.2	Datenvisualisierung	89
7.3	Stereovisualisierung	90
7.4	VTK	92
7.5	Fazit	92
8	Objektorientierte Realisierung	93
8.1	Einleitung	93
8.2	UML-Modelle	93
8.3	Klassenbeschreibungen wichtiger Klassen	96
8.3.1	Kalibrierung	96
8.3.2	Denoising	96
8.3.3	Dewarping	96
8.3.4	Segmentierung	97
8.3.5	Tropfenanalyse	97
8.3.6	Tracking	97
8.3.7	3D-Rekonstruktion	98
8.3.8	Visualisierung	98
8.3.9	Frequenzanalyse	98
8.4	Werkzeuge und verwendete Bibliotheken	98
9	Persistenz	111
9.1	Einleitung	111
9.2	Datenexport	112
9.2.1	Excel-Export	112
9.2.2	Video-Export	113
10	Auswertung eines Versuchs	115
10.1	Einleitung	115
10.2	Bildverbesserung	116
10.2.1	Kalibrierung	116
10.2.2	Denoising/Dewarping	117
10.3	Segmentierung der Lamelle	117
10.4	Tropfenanalyse	117
10.5	Frequenzanalyse	117
10.6	Tracking	118
10.7	3D-Rekonstruktion	118
10.8	Fazit	118

11 Zusammenfassung und Ausblick	123
11.1 Projektresumé	123
11.2 Ausblick	123
A Handbuch	125
A.1 Installation	125
A.1.1 Qt-Installation	125
A.1.2 MySQL-Server-Installation	125
A.1.3 3D-AugTracer-Installation	126
A.2 Kalibrierung	126
A.3 3D-AugTracer	129
A.3.1 Neues Projekt	129
A.3.2 Projekt laden	135
A.3.3 Anzeigen im Hauptfenster	136
A.3.4 Menü	137
B Pflichtenheft	143
B.1 Zielbestimmung	143
B.2 Produkteinsatz	144
B.3 Produktübersicht	145
B.4 Produktfunktionen	145
B.5 Produktdaten	146
B.6 Benutzungsoberfläche	146
B.7 Nichtfunktionale Anforderungen	146
B.8 Technische Produktumgebung	146
B.9 Anforderung an Entwicklungsumgebung	147
Abbildungsverzeichnis	152
Tabellenverzeichnis	152
Literaturangabe	159
Sachregister	160

Kapitel 1

Einleitung

Inhalt

1.1	Aufgabenstellung an die Projektgruppe	1
1.2	Versuchsaufbau	2
1.3	Stand der Forschung	4
1.4	Aufbau des Endberichts	4

1.1 Aufgabenstellung an die Projektgruppe

Die in den letzten Jahren aufkommende Debatte über die Reduzierung des Kraftstoffverbrauchs von PKWs sowie die zusätzliche Reduzierung des Schadstoffausstoßes haben die Entwicklung moderner Direkteinspritzmotoren vorangetrieben. Besonders die Entwicklung auf dem Gebiet der direkten Dieseleinspritzung, allgemein bekannt als DDI oder TDi Motoren, hat eine rasante Entwicklung genommen und führte zu einer Kraftstoffeinsparung von ca 45% [Sta03].

Zurückzuführen ist dies auf das gewachsene Verständnis über die Vorgänge des Zerfalls von Flüssigkeiten zu Tropfen definierter Größe und Verteilung während des Austritts aus einer Düse (siehe Abb. 1.1). Gewonnen werden konnten diese Erkenntnisse durch eine stark verbesserte Messtechnik, die es ermöglicht, Anregungs- und Zerfallsprozesse besser analysieren zu können [FLW04].

In Zusammenarbeit mit dem Lehrstuhl für Mechanische Verfahrenstechnik (Fachbereich Bio- und Chemieingenieurwesen) wird die Grundlagenforschung zum Verständnis von Zerfallsprozessen in Hohlkegeldüsen im Hinblick auf praktische Anwendungsmöglichkeiten (z.B. Motoren, medizinische Sprühgeräte, Lackspritzanlagen) untersucht. Die an der Düsenmündung austretende Flüssigkeit wird dabei üblicherweise als Lamelle bezeichnet (siehe Abb. 1.2).

Für weitergehende Untersuchungen, die zu einer Verbesserung der bisherigen Modelle führen sollen, ist eine exakte dreidimensionale Visualisierung der Flüssigkeit und ihres Verhaltens nach dem Austritt aus der Düse notwendig. Hier setzt die Aufgabe der Projektgruppe an. Innerhalb dieser wird die Form und Bewegung der Lamelle in einer erweiterten dreidimensionalen Realität (Augmented Reality, kurz AR) visualisiert [Azu97]. Augmentierte Wirklichkeit bedeutet in diesem Zusammenhang, dass Videodaten (reale Daten) der ausströmenden Flüssigkeit mit virtuellen, rechnergenerierten Daten in einer „Welt“ dargestellt werden. Durch diesen Ansatz ist es möglich, Informationen über den Zerfall der Lamelle, den Verlauf der Bewegung (Vektorfelder und Stromlinien) und die Geschwindigkeit innerhalb der Lamelle, die direkt aus den Vektorfeldern ableitbar ist, einzublenden. Zur Verstärkung des immersiven Eindrucks und damit auch zum besseren Verständnis dieser komplexen Vorgänge kann für die Darstellung stereographisches Visualisierungssystem (z.B. Stereoprojektor, Rückprojektionswand oder Head-Mounted-Display), welches „echte“ Dreidimensionalität ermöglicht, eingesetzt werden.



Abbildung 1.1: Darstellung der Lamelle

Es existieren zahlreiche theoretische Studien, die den Zerfallsmechanismus von flüssigen Filmen/Lamellen beschreiben. Im Allgemeinen ist der Wachstumsprozess mit Hilfe einer linearen Stabilitätsanalyse beschrieben worden. Bereits 1953 begann Squire [Squ] mit Untersuchungen von antisymmetrischen Oszillationen auf Lamellen mit konstanter Dicke, aus denen erste Berechnungsansätze für Wellenlängen und Wellenwachstum resultieren. Eine weiterführende Arbeit von Dombrowski [Dom] stellt Ansätze für Lamellen radial abnehmender Dicke auf und versucht den Einfluss der Viskosität darzustellen. Mit dem Ergebnis, dass die Lamellenausbreitung eine deutliche Abhängigkeit dieser Parameter zeigt. Der Zerfallsprozess infolge aerodynamischen Zerwellens wird über die Bildung von Fragmenten definiert. Erreicht die Welle eine kritische Amplitude, bilden sich aus der halben Wellenlänge Fragmente aus, die aufgrund der Oberflächenspannung in Ligamente und schließlich zu Tropfen zerfallen [Dom]. Diese Abrisspunkte bzw. -bereiche können mit Hilfe des in dieser Projektgruppe entwickelten Programms ermittelt sowie eine Segmentierung der Tropfen durchgeführt werden.

Primäres Ziel ist die Analyse der Silhouette vom Düsenaustritt bis vor den Abriss der Lamelle in Fragmente bzw. Tropfen. So ist es schließlich möglich, den Winkel des Sprühkegels als wichtige Kenngröße direkt zu bestimmen. Darüberhinaus erhält man Informationen über die Tropfenverteilung und eine Extraktion der Silhouette der Lamelle.

Es folgen Untersuchungen bezüglich der sich auf der Lamelle einstellenden Frequenzen, Amplituden, Wellenlänge und Wachstumsraten. Diese Kenngrößen lassen sich direkt aus der Frequenzanalyse ablesen.

3D-AugTracer ermöglicht außerdem die Berechnung weiterer spezieller Kenngrößen wie die Reynoldszahl, Ohnesorgezahl und Weberzahl, welche in Kapitel 2 näher erläutert werden.

1.2 Versuchsaufbau

Im Folgenden wird die Versuchsanordnung und -durchführung beschrieben, die zur Aufnahme der Bilder verwendet wurden. Diese bilden die Grundlage der Berechnungen in **3D-AugTracer**. Die zu untersuchende Flüssigkeit wird in einem Vorlagebehälter gesammelt, und mit Hilfe einer Kreiselpumpe im Kreislauf gefahren. Für jede Messung werden die Flüssigkeitstemperatur, der Volumenstrom und der Düsenvordruck aufgenommen, aus denen sich in Kombination mit der Lamellengeschwindigkeit und dem Sprühwinkel weitere Werte, wie z.B. die Druck-Reynolds-Zahl berechnen lassen (vgl. Kapitel 2.3.2).

Die Aufnahmen der Flüssigkeitslamelle werden mit Hilfe zweier horizontal und parallel zueinander positionierter CCD-Kameras (Typ *pixelfly* der Firma CPO [Dat]) im Durchlichtverfahren realisiert. Durch die Verwendung eines Objektivs mit langer Brennweite kann die Beleuchtung soweit hinter



Abbildung 1.2: Stroboskopische Darstellung des Sprühkegels

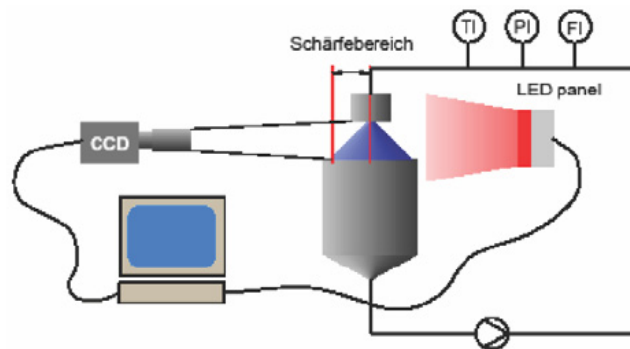


Abbildung 1.3: Schematischer Versuchsaufbau

das Düsenmodell und somit außerhalb des Schärfbereichs der Kamera platziert werden, dass die Leuchtdioden zu einer einheitlichen Fläche verschwimmen.

Die Kameras werden im Double-Shutter-Modus betrieben. Das bedeutet, dass jeweils zwei Bilder in sehr kurzer zeitlicher Abfolge aufgenommen werden können. Die maximale Dauerfrequenz einer Kamera beträgt 12,5 Hz, was einem zeitlichen Abstand von 80 ms entspricht. Sie werden untereinander und mit der LED-Beleuchtung synchronisiert, so dass sie immer gleichzeitig ein Bild aufnehmen. Die Kameras werden so positioniert, dass die Düse im oberen Bildbereich zu erkennen bleibt und die Lamelle etwa in der unteren Bildhälfte zu Tropfen zerfällt. Die Kameras liefern Serien von Einzelbildern, die für die rechte und linke Kamera in getrennten Ordnern gespeichert werden. Mit diesen Bildern und den Bildern aus der Kalibrierung werden die Berechnungen in **3D-AugTracer** durchgeführt.

Nach jedem Neuausrichten der Kameras müssen diese kalibriert werden (vgl. Kapitel 3.2.1). Dazu wird bei ausgeschalteter Pumpe ein genau bekanntes Kalibriermuster in unterschiedlichen Positionen unter die Düse gehalten und von beiden Kameras gleichzeitig aufgenommen. Die erste Aufnahme muss direkt unter der Düse und in senkrechter Ausrichtung zur Blickrichtung der Kameras

erfolgen. Danach folgen mindestens zwei weitere Aufnahmen, wobei das Kalibriermuster jedesmal gedreht und gekippt wird. Je mehr dieser Kalibrierungsaufnahmen durchgeführt werden, um so besser sind die Ergebnisse der späteren Berechnungen. Als Versuchsflüssigkeiten kamen Wasser

Gemische	Viskosität (mPas)	Dichte (kg/m ³)
Wasser	1,0	1000
Glycerin 1	4,9	1118
Glycerin 2	28,4	1188

Tabelle 1.1: Stoffdaten bei T=20°C

und zwei verschiedene Glycerin/Wasser-Gemische zum Einsatz. Die Stoffdaten sind in Tabelle 1.1 aufgeführt. Die Flüssigkeitsdrucke lagen zwischen $\Delta p = 0,3$ bis 1,6 bar.

1.3 Stand der Forschung

Was das Verhalten von Flüssigkeiten innerhalb von Hohlkegeldüsen betrifft, so lässt sich sagen, dass die Strömungseigenschaften weitgehend erforscht sind. Ganz anders verhält es sich jedoch außerhalb der Düsen im Sprühkegel. Über die dortigen Strömungsverhältnisse herrscht bislang wenig Klarheit, was zweifelsfrei daran liegt, dass die Charakteristika der Lamelle nur sehr schwierig zu analysieren sind.

Heutige Verfahren zur Messung von Gas-Feststoff-Strömungen, wie z. B. die PIV oder das Sensor-Array (vgl. Kapitel 2.5.1), basieren auf optischen Ansätzen. Dies hat zum Einen den Vorteil, dass der experimentelle Aufwand und damit verbunden auch der finanzielle Aufwand nicht all zu hoch sind. Auf der anderen Seite ist die Auswertung am Computer sehr viel einfacher. Strömungsphasen müssen nicht aufwendig durch Kamerafilter oder besondere Flüssigkeiten getrennt werden, sondern lassen sich durch geeignete Methoden per Computer erkennen. Neben dem finanziellen Aspekt spielt auch der Zeitgewinn dabei eine entscheidende Rolle.

In Bezug auf das Ziel dieser Projektgruppe ergeben sich in diesem Zusammenhang jedoch einige Fragen. Da der Sprühkegel mit zwei Kameras aufgenommen werden soll, stellt sich die Frage nach dem Informationsgewinn. Für die Messung des Sprühkegels ist ebenfalls eine Kombination mehrerer Verfahren denkbar. So ließen sich z. B. PIV und Sensor-Array kombinieren. Dadurch entsteht die Frage: Wie genau lassen sich beide Verfahren kombinieren? Kann man die Messergebnisse sinnvoll zusammen nutzen? Welche weiteren Vorteile ergeben sich durch die Verwendung zweier Kameras? Auch die Möglichkeit der Verwendung von Tracer-Partikeln wirft einige Fragen auf: Kann man sie innerhalb der Lamelle überhaupt erkennen? Verhalten sich die Tracer-Partikel innerhalb der Lamelle anders als in einer anderen Phase?

1.4 Aufbau des Endberichts

Die folgenden Kapitel dokumentieren die Arbeit der Projektgruppe **3D-AugTracer**. Der Aufbau dieses Endberichts orientiert sich dabei primär an der Ausführungsreihenfolge der Berechnungsschritte in dem entwickelten Programm (siehe Abb. 1.4).

Zunächst werden in Kapitel 3 die Methoden zur Bildverbesserung (Kamera-Kalibrierung, Rauschreduzierung und Entzerrung) erläutert. Bedingt durch die Kameraoptik kommt es zu Verzerrungen innerhalb der Aufnahmen, wodurch diese nicht direkt verwendbar sind. Unter Verwendung von eingebrachten Markierungsgittern kann zunächst eine Kalibrierung der Kameras erfolgen (siehe Kap. 3.2.1) [Zha00] und basierend auf diesen Daten eine Entzerrung (Dewarping) der Videodaten (siehe Kap. 3.3) [RM95], sowie eine Rauschreduzierung (siehe Kap. 3.4).

Nach der Verbesserung der Bilddaten können im nächsten Schritt nun die Lamelle (vgl. Kapitel 4.2), Ligamente und Tropfen (vgl. Kapitel 4.4) segmentiert werden. Darüber hinaus ist es möglich,

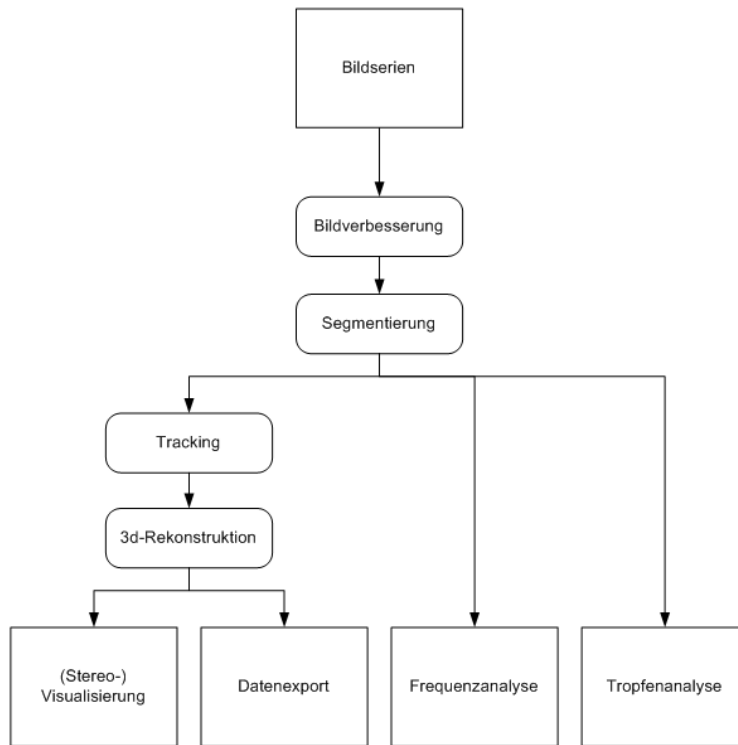


Abbildung 1.4: 3D-AugTracer - Übersicht

eine Frequenzanalyse durchzuführen (vgl. Kapitel 4.3).

Um Aussagen über die Bewegung des Sprühkegels und die Geschwindigkeit innerhalb der Lamelle zu erhalten, können Verfahren eingesetzt werden, die das Verfolgen von Objekten oder markanten Punkten über eine Bildsequenz ermöglichen. Als Ergebnis erhält man Vektorfelder und Stromlinien. Näheres zu diesen eingesetzten Trackingverfahren findet man in Kapitel 5.

Für die im vorhergehenden Schritt segmentierten Tropfen und die durch die Trackingverfahren erhaltenen Vektorfelder ist nun die tatsächliche räumliche Lage bzw. Richtung zu bestimmen. Dieses ist möglich, da zu jedem Zeitpunkt zwei Aufnahmen unter einem vorgegebenen Winkel vorliegen und die korrelierenden Aufnahmen in einem geometrischen Zusammenhang (epipolare Geometrie) stehen [Zha98]. Kapitel 6 geht auf die 3D-Rekonstruktion der gewonnenen Daten ein. Konform zu den dreidimensionalen Daten kann die Darstellung auf einer Rückprojektionswand erfolgen, die durch eine dreidimensionale stereographische Projektion eine immersive Visualisierung erlaubt. Kapitel 7 gibt eine Übersicht über die Visualisierung der Daten und die Stereoprojektion zum Beispiel mit einem HMD (head mounted display).

Kapitel 8 geht auf die objektorientierten Realisierung des Projekts ein. Es enthält eine Übersicht über die wichtigsten Klassen inkl. deren Beschreibungen sowie eine Übersicht der eingesetzten Softwarebibliotheken.

Das darauf folgende Kapitel 9 enthält Informationen zur Persistenz und Datenhaltung. Es wird dargestellt, wie die gewonnenen Daten exportiert werden können (Excel- und Videoexport) und wie sie dauerhaft gespeichert werden können.

Im Kapitel 10 wird eine Auswertung eines Versuchs exemplarisch wiedergegeben. Abschließend wird in Kapitel 11 ein Fazit zur Projektarbeit gezogen.

Im Anhang findet man zudem das Handbuch für die Installation, Inbetriebnahme und Benutzung von **3D-AugTracer** sowie das zu Beginn der Projektgruppe erstellte Pflichtenheft.

Kapitel 2

Grundlagen der Verfahrenstechnik - Zerstäubung

Inhalt

2.1	Übersicht	7
2.2	Grundlagen	7
2.2.1	Aufgaben und Einsatzgebiete	8
2.2.2	Aufbau und Funktionsweise einer Hohlkegeldüse	9
2.3	Strömungsverhalten	10
2.3.1	Struktur und Zerfallsprozess der Lamelle	10
2.3.2	Merkmale für die charakteristische Beschreibung einer Lamelle	11
2.3.3	Beispielhafte Untersuchung unterschiedlicher Stoffe	12
2.4	Interne Messverfahren	12
2.5	Externe Messverfahren	13
2.5.1	Particle Image Velocimetry	14

2.1 Übersicht

Das einführende Kapitel behandelt die allgemeinen chemietechnischen Grundlagen und die spezielle Darstellung des zu untersuchenden Problems. Es werden zunächst allgemeine Zerstäuber beschrieben und im speziellen die eingesetzten Hohlkegeldüsen klassifiziert.

Darauf aufbauend wird dargestellt, wie sich die unterschiedlichen Komponenten im Versuchsaufbau auf das Endergebnis auswirken. Die ermittelten Kenntnisse tragen zum Verständnis der Lamellenausprägung bei, deren Merkmalcharakterisierung im Anschluss erfolgt. Danach werden verschiedene Messverfahren vorgestellt und kurz erleutert.

2.2 Grundlagen

Der Aufbau einer Hohlkegeldüse bestimmt weitestgehend den austretenden Sprühkegel. Zusätzlichen Einfluss haben die eingesetzten Flüssigkeiten, deren Merkmale zunächst dargestellt werden. Die folgenden Absätze basieren größtenteils auf dem Buch „Atomization and Sprays“ von Arthur H. Lefebvre [Lef89b], das als Standardwerk für die Untersuchung von Düsensystemen gilt. Weitere Literaturquellen sind [Wal01] und [Koc06]

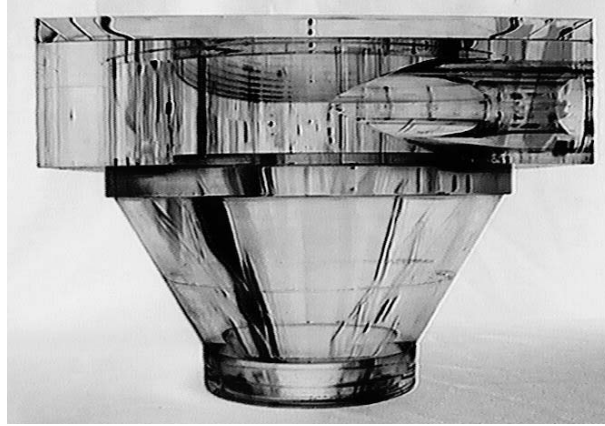


Abbildung 2.1: Hohlkegeldüse

2.2.1 Aufgaben und Einsatzgebiete

Das Zerstäuben von Flüssigkeiten und Suspensionen spielt in vielen Bereichen der Technik eine wichtige Rolle. Je nach Aufgabenstellung ergeben sich unterschiedliche Anforderungen an das herzustellende Spray. Dabei lassen sich als wesentliche Auslegungsgrößen die Tropfengröße, die Tropfengrößenverteilung, die Sprayform oder der Durchsatz aufzählen. Desweiteren spricht man von Einstoff- bzw. Zweistoff-Düsen, um den Unterschied zwischen dem Versprühen eines einzelnen Stoffes (Flüssigkeiten¹) und dem Vermischen eines flüssigen Stoffes mit einem zusätzlich eingeführten Gas herauszustellen.

Weit verbreitete Anwendung finden Zerstäuberdüsen in unterschiedlich realisierbaren Größen in vielen Bereichen und Verfahren. Ein klassisches Beispiel ist der Verbrennungsmotor, in dem Einspritz-Düsen eingesetzt werden um in der Zylinderkammer eine möglichst schnelle und auch gleichmässige Treibstoff-Zufuhr zu gewährleisten. Dadurch ergeben sich optimale Leistungsraten bei der anschließenden Zündung der Treibstoff-Gas-Gemische. In größerer Ausführung finden Zerstäuber in Lackierverfahren oder Produktionsprozessen Anwendung, da sie eine gute räumliche und vor allem gleichmäßige Ausbreitung der versprühten Flüssigkeiten garantieren können und somit z.B. die Feinheit und Ebenmäßigkeit einer Lackierung möglich wird.

Eine Einteilung, der in der Praxis eingesetzten Zerstäubertypen, lässt sich im Allgemeinen nach der Art der Zufuhr der Zerstäubungsenergie vornehmen. Es ergeben sich vier Gruppen von Zerstäubertypen:

- Einstoffdruckzerstäuber
- Pneumatische Zerstäuber
- Mechanische Zerstäuber
- Sonderzerstäuber

Die im Rahmen dieser Arbeit verwendeten Hohlkegeldüsen (HKD) gehören in den Bereich der sog. Einstoffdruckdüsen auf deren Funktionsweise im nächsten Abschnitt explizit eingegangen wird. Eine generelle Übersicht über die unterschiedlichen Zerstäubertypen ist z.B. in [Wal82] oder in [Wal01] zu finden.

¹Genaugenommen werden Flüssigkeiten in diesem Zusammenhang als Feststoffe bezeichnet.

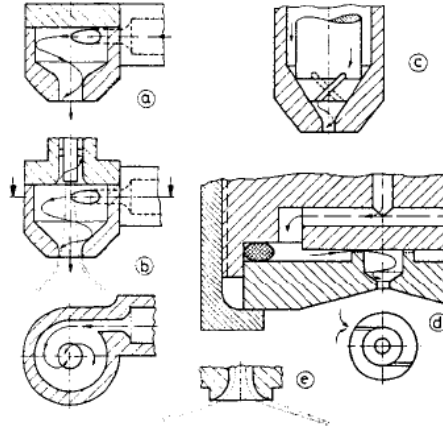


Abbildung 2.2: Bauarten von Hohlkegeldüsen

Bezeichnung	Geometrie
Drallkammerdurchmesser	50mm
Eintrittsöffnung	2x7mm
Austrittsdurchmesser	7x15mm

Tabelle 2.1: Geometriedaten der Hohlkegeldüse

2.2.2 Aufbau und Funktionsweise einer Hohlkegeldüse

Hohlkegeldüsen zählen aufgrund ihrer positiven Eigenschaften zu den effektivsten Einstoffdruckdüsen im Bereich der sog. Einstoffdüsen. Sie erzeugen bei Flüssigkeiten mit niedrigen Viskositäten und vorgegebener Druckdifferenz sehr kleine Tropfengrößen im Vergleich zum Düsendurchmesser und weisen aufgrund der relativ großen freien Querschnitte eine geringe Verstopfungsneigung auf. Hohlkegeldüsen sind in einer Vielzahl von Geometrien verfügbar (vgl. Abb. 2.2), aber im Allgemeinen ist nur ein begrenztes Wissen über deren Effizienz vorhanden, so dass sich die Auslegung eines solchen Zerstäubers für technische Anwendungen schwierig gestaltet. Der Zerfall von sich ausdünnenden Lamellen wird zumeist auf aerodynamisch angeregte Wellen zurückgeführt und ist dabei mit exponentiellem Wachstum der Amplituden entlang der Stromlinien verbunden.

Da sich Größe und Funktion der Hohlkegeldüsen je nach Einsatzgebiet stark verändern können, wird an dieser Stelle ein recht einfacher Ansatz für den Aufbau beschrieben. Die in den Versuchen eingesetzte Hohlkegeldüse besteht aus einem modularen Aufbau und ermöglicht somit die Untersuchung eines breiten Geometriespektrums zusammensetzbarer Komponenten, die es ermöglichen, beinahe jede beliebige Geometrie in kurzer Zeit aufzubauen (siehe Abb. 2.3). Es sollte erwähnt sein, dass die untersuchte Hohlkegeldüse nur einen Modellaufbau darstellt, der den in der Praxis um einiges kleineren Ablauf anschaulich darstellen soll.

- Eintrittsöffnung – die Flüssigkeit wird tangential in die Drallkammer eingeleitet. Um ein homogenes Sprühbild zu erhalten, sind mindestens zwei Eintrittsöffnungen erforderlich.
- Drallkammer – die Flüssigkeit legt sich aufgrund der Zentrifugalkraft an die innere Wand der Drallkammer an. Bei ausreichendem Differenzdruck bildet sich im Zentrum der Düse eine freie Grenzfläche aus, im Allgemeinen als Gaskern bezeichnet.
- Düsenmündung – den Abschluss der Hohlkegeldüse bildet die Mündung. Beim Überströmen der scharfkantigen Düsenmündung entsteht ein Kegel in Form eines Rotationshyperboloids dessen Dicke mit zunehmender Düsenentfernung abnimmt (vgl. Abb. 2.4).

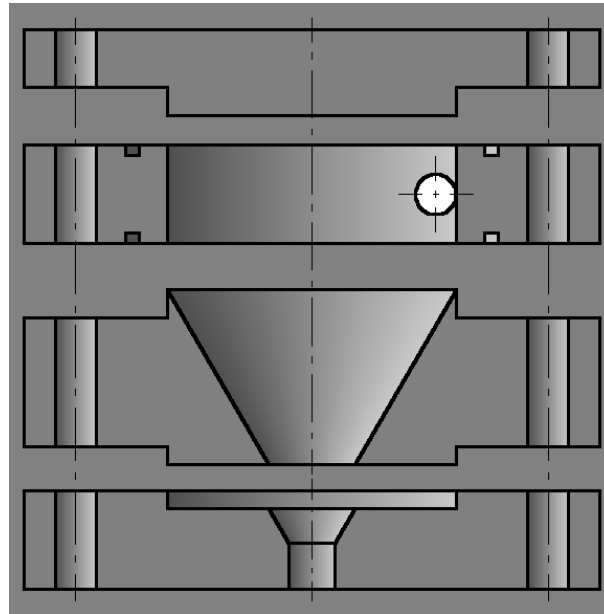


Abbildung 2.3: Modell einer Hohlkegeldüse

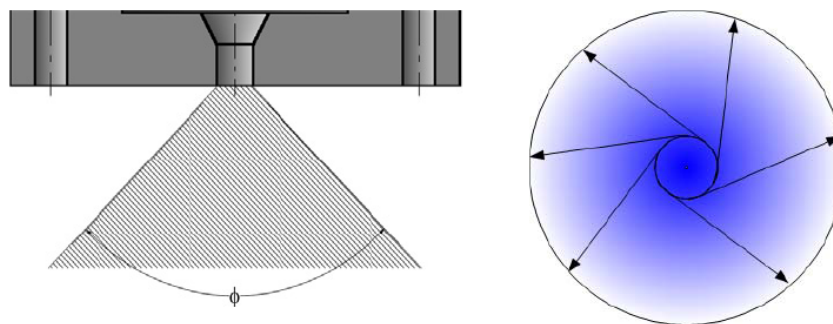


Abbildung 2.4: Rotationshyperboloid (Seitenansicht und Aufsicht)

2.3 Strömungsverhalten

2.3.1 Struktur und Zerfallsprozess der Lamelle

Für den Zerfall von Flüssigkeitslamellen können drei verschiedene Zerfallsmechanismen unterschieden werden:

- Zerfall durch Randkontraktion
- Zerfall infolge aerodynamischer Wellenbildung
- Turbulenter Zerfall

Die größte technische Bedeutung hat der Zerfall durch aerodynamische Wellenbildung und turbulenten Zerfall, auf den aber im Weiteren nicht näher eingegangen wird. Die Aufgabe der Projektgruppe bestand darin, den Einfluss durch aerodynamische Wellenbildung zu untersuchen. Aufgrund

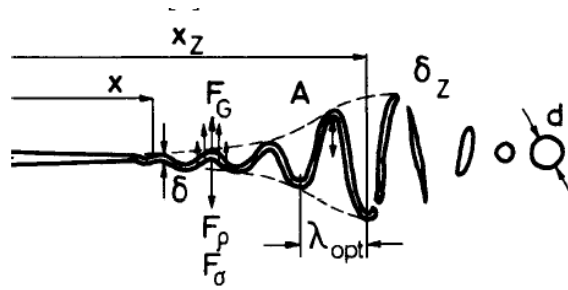


Abbildung 2.5: Aerodynamische Wellenbildung

von zufälligen Störungen bilden sich antisymmetrische, sinusförmige Wellen auf der Lamelle aus. Durch die Relativgeschwindigkeit zwischen Gas und Flüssigkeit entsteht an den Wellenbergen ein Unterdruck, der das Anwachsen der Ausbuchtungen der Wellen weiter verstärkt (vgl. Abb. 2.5). Im Allgemeinen wird die Annahme getroffen, dass sich die Wellen bzw. die Amplitude exponentiell entlang der Lauflänge entwickeln und die Laufgeschwindigkeit der Welle gleich der Relativgeschwindigkeit zwischen Gas und Flüssigkeit ist. Bei einem kritischen Amplitudenverhältnis zerfällt die Lamelle in ein Netz aus Fragmenten. Diese Fragmente ziehen sich aufgrund der Oberflächenspannung zu Flüssigkeitsligamenten zusammen, aus denen letztendlich nach dem Rayleighzerfall Tropfen entstehen.

2.3.2 Merkmale für die charakteristische Beschreibung einer Lamelle

Zur Charakterisierung der Lamelle bzw dessen Aufbruchverhalten ist die Kenntnis von Stoffeigenschaften (Dichte, Oberflächenspannung, etc) und der Geometrieparameter der Hohlkegeldüse zwingend erforderlich. Ziel der Projektgruppe war die Bestimmung von verschiedenen Merkmalen des Zerfallsprozesses:

- Tropfengrößenverteilung – es gibt eine Reihe von Untersuchungen und Verfahren zur Bestimmung der Tropfengröße. Aufgrund der statistischen Größenverteilung der Tropfen untereinander (Faktor 1-100) nimmt man als Merkmal häufig die mittlere Tropfengröße oder eine Verteilung über verschiedene Größen-Intervalle.
- Zerfallslänge – Die Zerfallslänge beschreibt die Länge der zusammenhängenden Lamelle oder charakterisiert den Punkt (bzw. Bereich) an dem keine eindeutige Zuordnung zwischen Tropfen und Lamelle erfolgen kann.
- Kegelwinkel – der Winkel der sich öffnenden Lamelle, je nach Düsensystem können bis annähernd 180° erreicht werden.
- Frequenzen der Oberflächenwellen – Diese charakterisiert die orthogonalen, in Bezug auf die Düsenöffnung senkrechten, Schwingungen der Oberflächenwellen. Die Untersuchungen der Projektgruppe bezogen sich auf die Extraktion der Schwingungen auf dem Kegel (Frequenzanalyse und Amplitudenbestimmung).

- Geschwindigkeitszahl φ

$$\varphi = \frac{v_{eff}}{u} \left[\frac{\frac{m}{s}}{\frac{m}{s}} \right] \text{ mit } u = \sqrt{2\Delta\rho_l/\rho_l} \left[\frac{bar}{m^3} \right]$$

wobei v_{eff} die effektive Geschwindigkeit und u die Potentialgeschwindigkeit ist.

Die Geschwindigkeitszahl φ beschreibt den Dissipationsverlust der Strömung in der Düse. Die effektive Geschwindigkeit soll durch die Bewegung der Lamelle bestimmt werden. Dieses Problem wurde von der PG-Gruppe gelöst, indem durch die Bewegung des Vektorfeldes die Geschwindigkeit analysiert wird. Die Potentialgeschwindigkeit wird durch die Dichte bzw.

die Änderung der Dichte bestimmt. Da die Dichte für bestimmte Substanzen konstant ist, konnten die Kennzahl unmittelbar bestimmt werden.

- Reynoldszahl Re

$$Re = \frac{vD\rho}{\eta} \left[\frac{\frac{m}{s} mm \frac{kg}{m^3}}{Pas} \right]$$

wobei v die Geschwindigkeit, D die Düsendurchmesser, ρ die Dichte und η die Viskosität sind.

Die Reynoldszahl ist eine dimensionslose Kennzahl. Sie beschreibt das Verhältnis von Trägheitskraft zu Zähigkeitskraft. v hat hier dieselbe Bedeutung wie oben dargestellte effektive Geschwindigkeitszahl v_{eff} . Den Düsendurchmesser(D) bzw. die Dichte(ρ) und die Viskosität(η) eines Stoffes kann man direkt aus existierenden Daten erhalten.

- Ohnesorgezahl Oh

$$Oh = \frac{\eta}{\sqrt{\sigma\rho D}} \left[\frac{Pas}{\frac{N}{m} \frac{kg}{m^3} mm} \right]$$

wobei η die Viskosität, σ die Oberflächenspannung, ρ die Dichte und D der Durchmesser der Ligamente sind.

Die Ohnesorgezahl ist eine Kennzahl, die nur aus Stoff- und Geometriedaten besteht. Sie beschreibt das Verhältnis von Zähigkeitskräften zu Oberflächenkräften bei der Deformation von Ligamenten. Hier soll der Durchmesser von Ligament betrachtet werden.

- Weberzahl We

$$We = \frac{v^2 D \rho}{\sigma} \left[\frac{\frac{m^2}{s^2} mm \frac{kg}{m^3}}{\frac{N}{m}} \right]$$

wobei v die effektive Geschwindigkeit, D die Lamellendicke, ρ die Dichte und σ die Oberflächenspannung bezeichnen.

Die Weberzahl beschreibt das Verhältnis der auf den Tropfen wirkenden Trägheitskraft zu den Oberflächenkräften. Sie soll hier in zwei Teilen berechnet werden. Der erste Teil bezieht sich auf den oberen Teil der Lamelle. Hier wird der Düsendurchmesser als die Lamellendicke D benutzt. Im zweiten Teil wird die Dicke der Ligamente als Lamellendicke benutzt. Weil die Daten der Ligamente nicht bekannt sind, wird nur die Weberzahl für den oberen Teil berechnet.

2.3.3 Beispielhafte Untersuchung unterschiedlicher Stoffe

Unterschiedliche Flüssigkeiten können bei identischen Umgebungsbedingungen stark variierende Eigenschaften haben, welche sich stark auf den Zerfallsprozess der Lamelle auswirken, Abb. 2.6 zeigt dies visuell, einige Eigenschaften sind in Tabelle 2.2 aufgelistet.

Flüssigkeit	Temperatur (K)	Viskosität (kg/ms)	Dichte (kg/m ³)	Oberflächenspannung (kg/s ²)
Aceton	293	0.00032	792	0.0237
Glycerin	273	12.1	1260	0.0630
Glycerin	293	0.622		0.0630
Mercurium	293	0.00153	13550	0.480
Wasser	300	0.00085	1000	0.0717

Tabelle 2.2: Eigenschaften ausgewählter Flüssigkeiten

2.4 Interne Messverfahren

Die Untersuchung der Strömungsprozesse innerhalb der Hohlkegeldüse ist nicht Bestandteil der Projektgruppe. Der Vollständigkeit halber wird an dieser Stelle nur kurz erwähnt, dass es Möglichkeiten gibt, diese Strömungsdaten durch Simulationen zu bestimmen, da die Vorgänge relativ

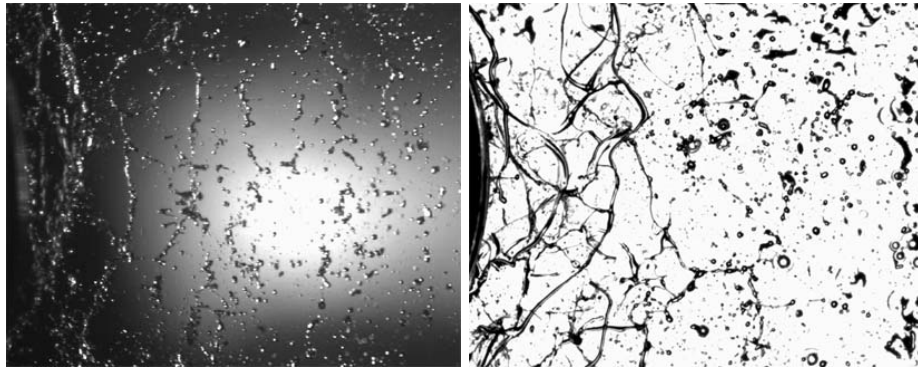


Abbildung 2.6: Zerfallsprozess für Wasser und Wasser-Glycerin-Gemisch

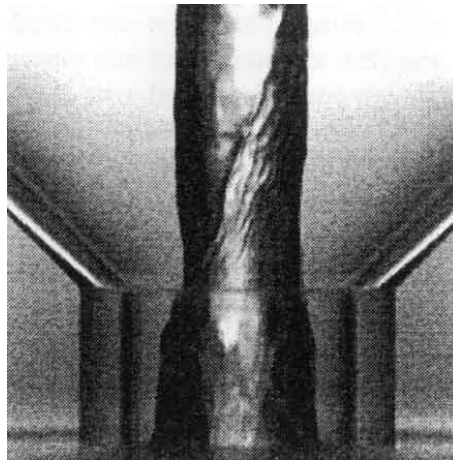


Abbildung 2.7: Luftkern

genau bekannt sind.

Weitere Untersuchungen betrachten den Luftkern, der sich im Zentrum der Hohlkegeldüse bildet. Dieser wird durch einen Laserstrahl abgescannt, so dass sich ein klares Bild seiner Verteilung ergibt (siehe Abb. 2.7). Weitere Informationen über die Untersuchung interner Prozesse finden sich z.B. in [DEML03].

2.5 Externe Messverfahren

Die externen Messverfahren dienen zur Charakterisierung des Strömungsverhaltens außerhalb der Düse. Grundsätzlich werden dabei zwei verschiedene Ansätze unterschieden.

Zu den nicht optischen Messverfahren gehören die Tomographie, bei der radioaktive Tracer-Partikel in die Strömung eingebracht werden, sowie die Messung mit Röntgenstrahlung. Diese Verfahren werden immer dann eingesetzt, wenn die zu untersuchende Strömung optisch nicht klar ist.

Liegt eine klare Strömung vor, kann auf die optischen Messverfahren zurückgegriffen werden. Zu diesen zählen die Holographie, die Laser-Doppler-Anemometrie (LDA), das Sensor-Array sowie die Lichtschnitt-Verfahren Particle Tracking Velocimetry (PTV), Particle Image Velocimetry (PIV) und Laser Speckle Velocimetry (LSV).

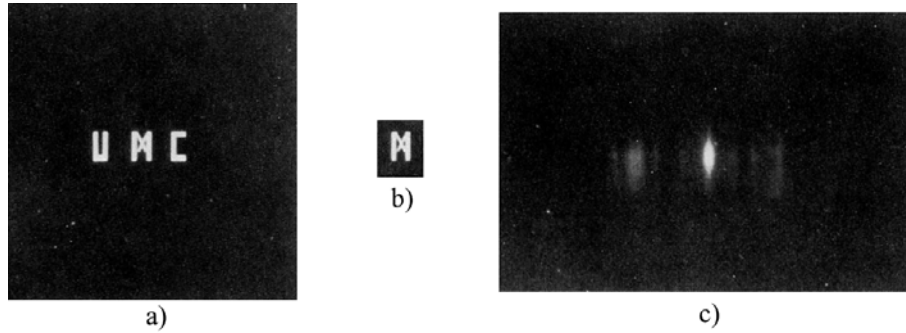


Abbildung 2.8: Beispiel: a) Bild f , b) Muster h , c) Darstellung der Matrix ϕ

2.5.1 Particle Image Velocimetry

Die Particle Image Velocimetry (PIV) gehört ebenso wie die Particle Tracking Velocimetry (PTV) und die Laser Speckle Velocimetry (LSV) zu den Lichtschnitt-Verfahren. Bei diesen Verfahren handelt es sich um flächige Messverfahren zur Geschwindigkeitsbestimmung von Partikeln innerhalb einer Gas-Feststoff-Strömung. Die Beschreibungen zum Aufbau und zur Auswertung sind ausführlich in [Gro02] beschrieben. Ziel der PIV ist es, anhand von Tracer-Partikeln, die in die Strömung eingebracht werden, die Geschwindigkeit der einzelnen (Tracer-)Partikel zu bestimmen.

In einem weiteren Schritt müssen nun die Partikel innerhalb der Testsubstanz detektiert werden, dies geschieht über visuellen Vergleich. Setzt man zwei Variablen zueinander in Beziehung, so wird dies allgemein als Korrelation bezeichnet. Im Zusammenhang der Bildverarbeitung wird die Korrelation dazu benutzt, ein Muster h in einem Bild f aufzufinden (siehe Abb. 2.8).

Aus der Sicht der Signaltheorie entspricht die Korrelation der Faltung zweier Signale. Besitzen die Signale f und h die Dimensionen $M \times N$, so ergibt sich für die Korrelationsmatrix:

$$\phi_{fh}(m, n) := f(m, n) \circ h(m, n) := \sum_{\xi=0}^{M-1} \sum_{\eta=0}^{N-1} f(\xi, \eta) h(\xi - m, \eta - n) \quad (2.1)$$

Da die allgemeine Korrelation ein weitgehend häufig eingesetztes Mittel in der Bildverarbeitung darstellt, wird darauf hingewiesen, dass sie in vielen der innerhalb der PG eingesetzten Verfahren benötigt wird. So werden z.T. veränderte Korrelations-Berechnungen in der Epipolar-Berechnung (vgl. Kapitel 6) oder den unterschiedlichen Tracking-Verfahren (vgl. Kapitel 5) eingesetzt.

Kapitel 3

Bildverbesserung

Inhalt

3.1	Einleitung	15
3.2	Kamera-Kalibrierung	15
3.2.1	Einleitung	15
3.2.2	Realisierung	17
3.2.3	Anmerkungen	19
3.3	Dewarping	20
3.3.1	Einleitung	20
3.3.2	Vorgehensweise	20
3.3.3	Algorithmus	21
3.4	Denoising	22
3.4.1	Einleitung	22
3.4.2	Denoising-Algorithmus	22
3.4.3	Parameter	24
3.4.4	Anmerkungen	25
3.5	Fazit	25

3.1 Einleitung

Bedingt durch die Kameraoptik kommt es zu Verzerrungen innerhalb der Aufnahmen, wodurch diese nicht direkt verwendbar sind. Unter Verwendung von eingebrachten Markierungsgittern kann zunächst eine Kalibrierung der Kameras erfolgen und basierend auf diesen Daten eine Entzerrung (Dewarping) der Videodaten. Darüber hinaus werden die Parameter der Kamera-Kalibrierung für die 3D-Rekonstruktion benötigt. Es wird zudem eine Rauschreduzierung auf den Aufnahmen durchgeführt, um später genauere Ergebnisse zu erhalten.

Insgesamt wird eine effiziente algorithmische Umsetzung angestrebt.

3.2 Kamera-Kalibrierung

3.2.1 Einleitung

Um eine spätere Rekonstruktion von 3D-Daten aus den 2D-Bildern zu ermöglichen, muss eine Kamerakalibrierung vorgenommen werden. Hierbei werden die intrinsischen und extrinsischen Parameter der beiden Kameras bestimmt.

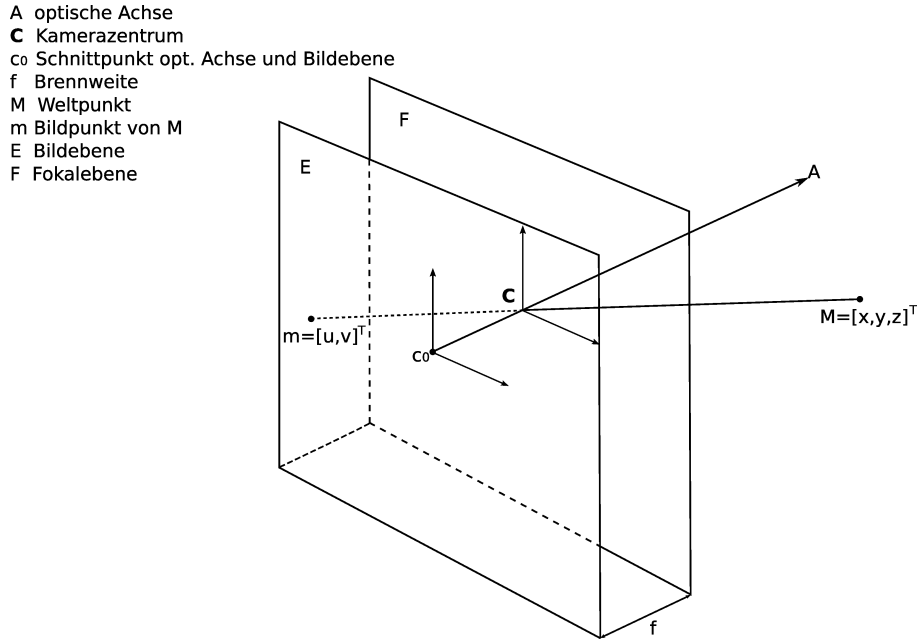


Abbildung 3.1: Projektion des Weltpunktes auf den Kamerapunkt

Zu den intrinsischen Parametern gehören der Bildhauptpunkt (u_0, v_0) , der leicht verschoben sein kann, die horizontalen und vertikalen Skalierungsfaktoren k_u und k_v und die effektive Brennweite f . Diese Parameter hängen von der Kamera ab und verändern sich nicht.

Die allgemeine Kamera-Kalibrierungsmatrix beschreibt die projektive Transformation von homogenen 3D-Weltkoordinaten in normalisierte homogene 2D-Bildkoordinaten:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.1)$$

Im Gegensatz zu den intrinsischen beschreiben die extrinsischen Parameter das Lageverhältnis zwischen Kamera- und Weltkoordinatensystem, sodass sich diese je nach Aufnahme verändern können. Sie setzen sich aus der Translation \mathbf{t}

$$\mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (3.2)$$

sowie der Rotation \mathbf{R} zusammen:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (3.3)$$

Diese ergibt sich aus der Verknüpfung der drei Rotationsmatrizen:

$$\mathbf{R} = \mathbf{R}_x^\alpha \circ \mathbf{R}_y^\alpha \circ \mathbf{R}_z^\alpha$$

mit

$$\mathbf{R}_x^\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}, \mathbf{R}_y^\alpha = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}, \mathbf{R}_z^\alpha = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

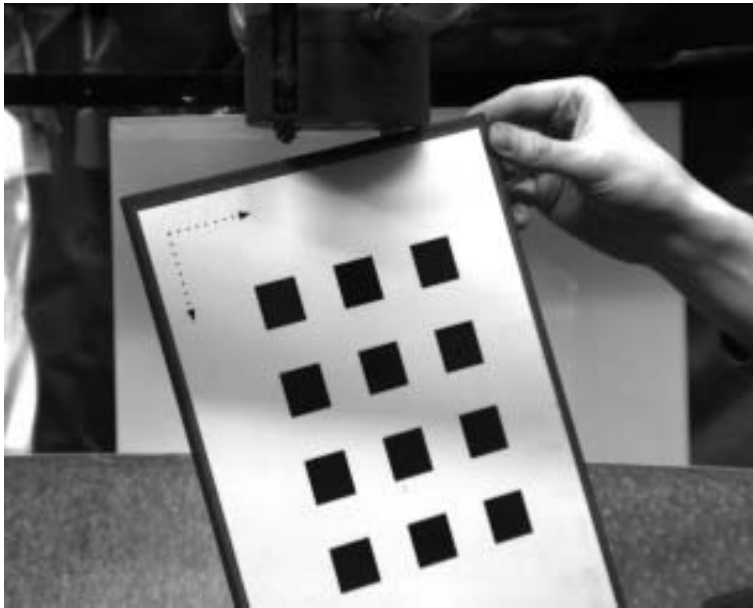


Abbildung 3.2: Aufnahme des Kalibrierungsgitters unterhalb der Düse

wobei \mathbf{R}_i^λ eine Rotation um die Achse i mit dem Winkel λ beschreibt.

Als Lösungsansätze bieten sich Methoden der direkten Messung an, die aber üblicherweise aufwendig oder gar unmöglich sind. Ein alternatives und gängigeres Verfahren, das auch hier verwendet wird, ist die Verwendung von Kalibrierpunkten, deren räumliche Lage bekannt ist. Es wurde das Verfahren nach Zhang [Zha00] gewählt, da es sich flexibler als die anderen Verfahren erweist. Hierbei wird auf ein Muster zurückgegriffen, das in verschiedenen Orientierungen (mindestens zwei) gezeigt wird. Dabei kann entweder die Kamera oder das Kalibrierungsgitter bewegt werden, wobei die Art der Bewegung nicht bekannt sein muss.

Auf das ausgewählte Verfahren von Zhang wird weiterhin eine Levenberg-Marquardt-Optimierung zur Verbesserung der Ergebnisse angewandt.

3.2.2 Realisierung

Um das Verhältnis von Kamera- und Weltkoordinatensystem zu berechnen, werden zunächst Referenzpunkte benötigt, deren Koordinaten bekannt sind. Um diese zu erhalten, wird ein Kalibrierungsgitter verwendet (siehe Abb. 3.2). Dieses besteht aus schwarzen Quadraten auf weißem Hintergrund, die Koordinaten der Eckpunkte sind bekannt und werden in einer Textdatei gespeichert. Da die Kameras fest installiert sind, wird das Gitter bewegt, mit unterschiedlichen Ausrichtungen unter die Düse gehalten und von beiden Kameras aufgenommen.

Um die Parameter zu errechnen wurde ein Kalibrierungstool entwickelt, das unabhängig vom Programm **3D-AugTracer** gestartet werden kann. Da das Verfahren von Zhang verwendet wird, werden mindestens drei verschiedene Aufnahmen des Kalibrierungsgitters benötigt. Da die Berechnung der Parameter über eine Schätzung einer Homographie aus den Bildern heraus geschieht, wird die Berechnung umso genauer, desto mehr Bilder man in unterschiedlichen Positionen aufnimmt.

Diese Bilder werden paarweise in das Kalibrierungstool geladen. Mit der Maus markiert der Benutzer in jedem Bild einen viereckigen Bereich, in dem die schwarzen Quadrate des Referenzgitters liegen. Danach erfolgt eine automatische Suche nach den Eckpunkten. Die gefundenen Punkte werden mit den gegebenen Referenzpunkten verglichen und nach der Methode von Zhang wird eine Homographie geschätzt.

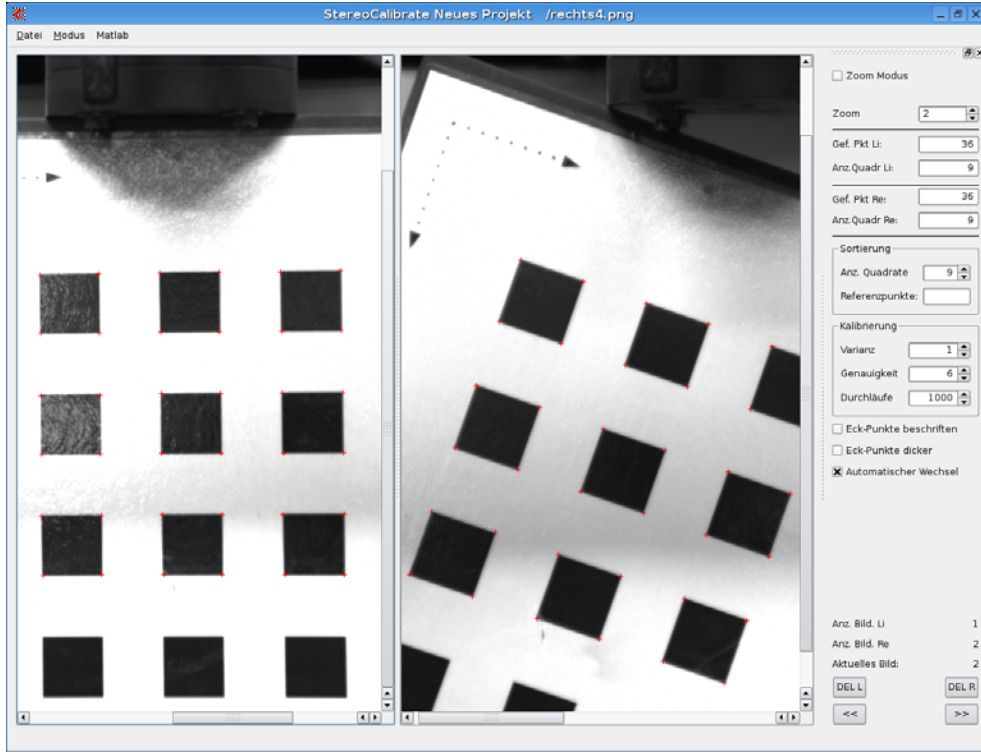


Abbildung 3.3: Bildschirmabdruck des Kalibrierungstools

So werden zunächst die intrinsischen Parameter ermittelt, aus denen die extrinsischen berechnet werden können. Danach folgt eine Verfeinerung der erhaltenen Werte.

Zunächst gilt: ein Modellpunkt $\tilde{M} = [X, Y, 1]^T$ und der zugehörige Bildpunkt $\tilde{m} = [u, v, 1]^T$ lassen sich über eine Homographie $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3] = \mathbf{A}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]$ wie folgt verknüpfen:

$$s\tilde{m} = \mathbf{H}\tilde{M} \quad (3.4)$$

Da \mathbf{r}_1 und \mathbf{r}_2 orthonormal zueinander sind, ergibt sich (\mathbf{A}^{-T} entspricht der inversen und transponierten Matrix \mathbf{A} (vgl. Kapitel Mathematische Notationen)):

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 = 0 \quad (3.5)$$

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 \quad (3.6)$$

Dies sind zwei Nebenbedingungen für die intrinsischen Parameter. Da es bei der Homographie acht Freiheitsgrade gibt, aber schon sechs extrinsische Parameter besetzt werden müssen (drei für Rotation, drei für Translation) bleiben nur diese zwei Bedingungen für die intrinsischen Parameter. Um das Kamerakalibrierungsproblem zu lösen wird mit einer analytischen Lösung gestartet.

Sei

$$\mathbf{B} = \mathbf{A}^{-T} \mathbf{A}^{-1} \equiv \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{pmatrix} \quad (3.7)$$

$$= \begin{pmatrix} \frac{1}{\alpha^2} & -\frac{c}{\alpha^2\beta} & \frac{cv_0 - u_0\beta}{\alpha^2\beta} \\ -\frac{c}{\alpha^2\beta} & \frac{c^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{c(cv_0 - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{cv_0 - u_0\beta}{\alpha^2\beta} & -\frac{c(cv_0 - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(cv_0 - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{pmatrix} \quad (3.8)$$

\mathbf{B} ist symmetrisch und über einen 6D-Vektor \mathbf{b} definiert:

$$\mathbf{b} = [B_{11}B_{12}B_{22}B_{13}B_{23}B_{33}]^T \quad (3.9)$$

Der i -te Spaltenvektor von \mathbf{H} sei $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$. Dann gilt:

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b} \quad (3.10)$$

mit $\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$. Also können die beiden Nebenbedingungen (3.5) und (3.6) aus der gegebenen Homographie als zwei homogene Gleichungen in \mathbf{b} geschrieben werden:

$$\begin{pmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{pmatrix} \mathbf{b} = 0 \quad (3.11)$$

Bei n Bildern des Modellgitters erhält man

$$\mathbf{V} \mathbf{b} = 0 \quad (3.12)$$

mit \mathbf{V} als $2n \times 6$ Matrix. Die Lösung zu (3.12) kann über Singularwertzerlegung errechnet werden. Es ist der Eigenvektor von $\mathbf{V}^T \mathbf{V}$ mit dem kleinsten Eigenwert.

Sobald \mathbf{b} geschätzt wurde, kann die interne Matrix \mathbf{A} bestimmt werden. Es gilt:

$$v_0 = (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2) \quad (3.13)$$

$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})] / B_{11} \quad (3.14)$$

$$\alpha = \sqrt{\lambda / B_{11}} \quad (3.15)$$

$$\beta = \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \quad (3.16)$$

$$c = -B_{12}\alpha^2\beta/\lambda \quad (3.17)$$

$$u_0 = cv_0/\alpha - B_{13}\alpha^2/\lambda \quad (3.18)$$

$$(3.19)$$

Wenn \mathbf{A} bekannt ist, so können für jedes Bild die externen Parameter errechnet werden:

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1} \mathbf{h}_1 \quad (3.20)$$

$$\mathbf{r}_2 = \lambda \mathbf{A}^{-1} \mathbf{h}_2 \quad (3.21)$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \quad (3.22)$$

$$\mathbf{t} = \lambda \mathbf{A}^{-1} \mathbf{h}_3 \quad (3.23)$$

mit $\lambda = 1 / \|\mathbf{A}^{-1} \mathbf{h}_1\| = 1 / \|\mathbf{A}^{-1} \mathbf{h}_2\|$.

Die Ungenauigkeiten bei der Berechnung der intrinsischen Parameter werden dadurch ausgeglichen, dass sie über die verschiedenen Bilder gemittelt werden.

Die extrinsischen Parameter unterscheiden sich bei den unterschiedlichen Orientierungen voneinander, also setzt man jeweils paarweise eine Kamera in den Ursprung und berechnet die Lage der anderen Kamera dazu. Da sich die Kameras während der Aufnahmen nicht bewegen, kann man diese Werte ebenfalls mitteln, um eine genauere Angabe der extrinsischen Parameter zu erhalten.

3.2.3 Anmerkungen

Als Einstieg wurde das Kalibrierungstool der vorherigen PG464 betrachtet. Auch diese hat die Methode von Zhang aufgegriffen, die grundlegenden Methoden sind also ähnlich. Für das vorliegende Projekt wurden allerdings einige Änderungen übernommen, so wird auch auf einem anderen Kalibrierungsgitter gearbeitet.

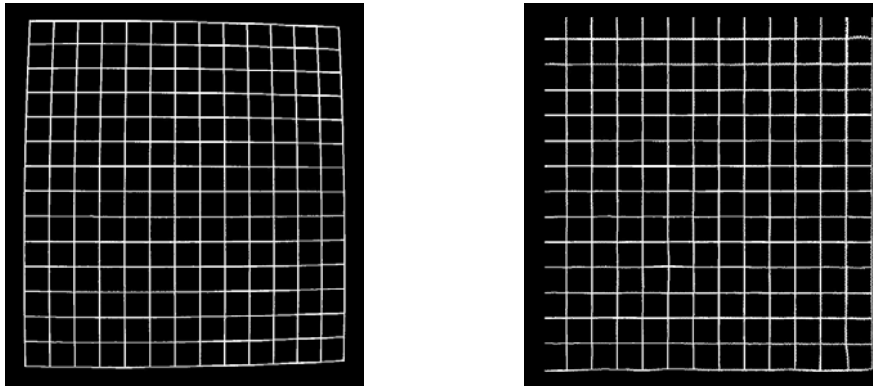


Abbildung 3.4: Dewarting-Beispiel, links: verzerrt, rechts: entzerrt

Für die Matrizenberechnung wurde auf die Bibliothek `newmat` zurückgegriffen, über die auch Singularwertzerlegungen berechnet werden können.

Die erhaltenen Ergebnisse wurden mit Hilfe des Programms `Matlab` überprüft, zu dem auch ein Kalibrierungstool von Zhang existiert. Mit Hilfe dieses Tools konnte man feststellen, dass die Kalibrierung ein akzeptables Ergebnis liefert, solange alle Eckpunkte der Quadrate des Kalibrierungsgitters exakt gefunden werden. Aus wenigen Pixeln Verschiebung kann jedoch eine grobe Ungenauigkeit der errechneten Parameter resultieren. Auch die Levenberg-Marquardt-Optimierung kann in diesem Fall nur bedingt den Fehler verringern.

Die Kalibrierung ist ein eigenständiges Tool und kann unabhängig gestartet werden.

3.3 Dewarting

3.3.1 Einleitung

Die aufgenommenen Bilder weisen den so genannten Barrel-Effekt auf (siehe Abb. 3.4, links). Das bedeutet, dass die Bilder vor allem an den Seiten verzerrt sind. Gründe hierfür sind Abweichungen in der Linse der Kamera und Beeinflussung durch das Magnetfeld der Erde. Diese Verzerrungen des Bildes führen zu möglicherweise erheblichen Abweichungen bei der Segmentierung und beim Tracking. Daher müssen die Bilder mit entsprechenden Dewarting-Algorithmen entzerrt werden (siehe Abb. 3.4, rechts) [RM95].

3.3.2 Vorgehensweise

Für das Entzerrern der Bilder ist es zunächst erforderlich, ein Kalibrierungsgitter (siehe Abb. 3.2) aufzunehmen und in der Aufnahme die Gitterpunkte zu ermitteln. Die so festgestellten Gitter-Eckpunkte sind natürlich auch verzerrt (siehe Abb. 3.5, links). Sie werden mit \mathbf{p}_i bezeichnet, die korrekten (entzerrten) Gitterpositionen mit \mathbf{r}_i (siehe Abb. 3.5, rechts). Im nächsten Schritt werden die Korrekturvektoren $\mathbf{t}_i = \mathbf{p}_i - \mathbf{r}_i$ ermittelt (siehe Abb. 3.6). Gesucht ist nun zunächst eine Interpolationsfunktion $f(p)$, welche alle Stützpunkte p_i des Gitternetzes auf die zugehörigen Korrekturvektoren t_i abbildet. Hierzu eignen sich radiale Basisfunktionen auf Basis der Hardy'schen Multiquadriken (s.u.).

Nun kann das Bild wie folgt neu berechnet werden: Für je vier benachbarte Pixel wird die korrigierte Position durch Addieren des Korrekturvektors, welcher mit der Interpolationsfunktion ermittelt wurde, bestimmt. Der eventuell entstandene Raum zwischen den Pixeln im Zielbild wird mittels bilinearer Interpolation aufgefüllt, damit keine Lücken entstehen [Pum06].

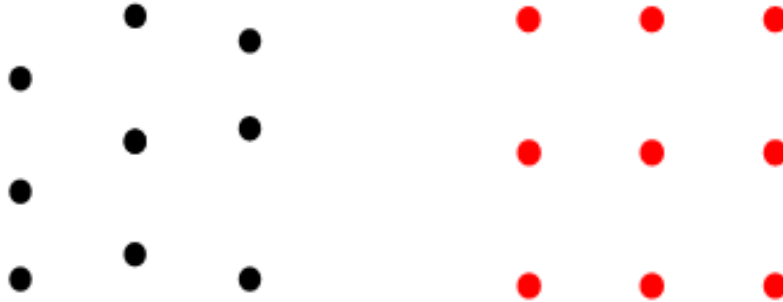


Abbildung 3.5: Gitterpunkte, links: verzerrt, rechts: entzerrt

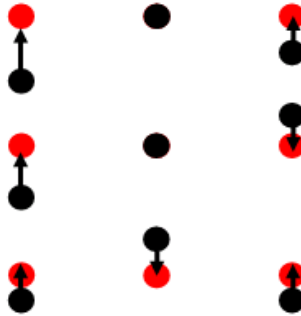


Abbildung 3.6: Verschiebevektoren für die Gitterpunkte

3.3.3 Algorithmus

Das Prinzip des Dewarping-Algorithmus ist folgendermaßen:

- Bestimmung der verzerrten Punkte des Gitters (vgl. Kapitel 3.2.1).
- Bestimmung der korrespondierenden Referenzgitterpunkte (vgl. Kapitel 3.3.3.1).
- Berechnen der Verschiebevektoren der Gitterpunkte zu den Punkten des Referenzgitters (vgl. Kapitel 3.3.3.2).
- Für jeden Pixel wird entsprechend der Verschiebevektoren der Gitterpunkte ein Verschiebevektor durch Interpolation berechnet (vgl. Kapitel 3.3.3.3).
- Die einzelnen Pixel werden nun gemäß der vorher berechneten Verschiebevektoren verschoben und es wird so ein neues, unverzerrtes Bild aufgebaut.

3.3.3.1 Bestimmung der korrespondierenden Referenzgitterpunkte

Bei den Bildern kann man davon ausgehen, dass das Bild in der Mitte nur gering verzerrt ist. Daher ist es möglich, aus den Gitterabständen der mittigsten Punkte des Gitters das restliche Referenzgitter zu rekonstruieren. Hierzu wird zunächst der Mittelpunkt des Bildes berechnet und ausgehend davon die vier Gitterpunkte, die den geringsten Abstand zu dem Mittelpunkt haben, bestimmt. Anschließend wird für je zwei dieser vier Punkte der Abstand d berechnet. So ist es möglich, ein Referenzgitter aufzubauen, dessen Gitterabstände genau d entspricht. Man erhält für jeden der n Punkte \mathbf{p}_i des verzerrten Gitters einen Referenzpunkt \mathbf{r}_i (Stützstelle).

3.3.3.2 Berechnen des Verschiebevektoren der Gitterpunkte

Hierbei wird lediglich eine einfache Abstandsberechnung der Punkte zu ihren zugewiesenen Referenzpunkten durchgeführt. Mit dem Gitterpunkt \mathbf{p}_i und dem Referenzgitterpunkt \mathbf{r}_i wird der Verschiebevektor \mathbf{v}_i mit $i = 1, \dots, n$ folgendermaßen berechnet:

$$\mathbf{v}_i = (\mathbf{p}_i - \mathbf{r}_i) \quad (3.24)$$

3.3.3.3 Radiale Basisfunktion

Zur Berechnung der Verschiebevektoren der einzelnen Pixel werden radiale Basisfunktionen verwendet. Hierbei wird die Interpolationsfunktion aus einer Linearkombination von Basisfunktionen konstruiert, um anschließend die Koeffizienten α_i der Basisfunktionen

$$f(x) = \sum \alpha_i R(d_i(x)) \quad (3.25)$$

zu ermitteln. Die Werte der Funktion \mathbf{R} hängen nur von den Abständen der Punkte (Pixel) zu den Stützstellen des Gitters ab, daher sind die einzelnen Basisfunktionen \mathbf{R}_i radialsymmetrisch. Die Koeffizienten α_i werden durch Einsetzen der Stützstellen und das darauf folgende Lösen des so entstehenden Gleichungssystems, berechnet. Sehr bekannte radiale Basisfunktion sind die der Hardy'schen Multiquadriken:

$$R(d_i(x)) = (d_i^2 + r_i^2)^\mu \quad (3.26)$$

Bei der Wahl des Exponenten schlägt Hardy einen Wert von $\mu = 1$ vor, aber auch $\mu = -1$ liefert gute Ergebnisse. Aus verschiedenen Quellen [RM92] [RM95] wurde ersichtlich, dass das Berechnen von individuellen \mathbf{r}_i für jeden Datenpunkt \mathbf{p}_i notwendig ist. Hierbei ist \mathbf{r}_i der kürzeste Abstand von \mathbf{p}_i zu seinem nächsten Nachbarn:

$$\mathbf{r}_i = \min_{i \neq j} d_i(x_j). \quad (3.27)$$

3.4 Denoising

3.4.1 Einleitung

Obwohl die Kameras von sehr guter Qualität sind, können die Bilder Diskontinuitäten, wie z.B. Rauschen, enthalten. Um diese Fehlerquelle in der nachfolgenden Bearbeitung der Bilder möglichst gering zu halten, muss zunächst das Rauschen reduziert werden.

3.4.2 Denoising-Algorithmus

Das aufgenommene Bild verhält sich zum reellen Bild nach der folgenden Gleichung:

$$\tilde{F}(u, v) = F(u, v) * H(u, v) + R(u, v) \quad (3.28)$$

In dieser Gleichung steht $\tilde{F}(u, v)$ für das aufgenommene Bild, $F(u, v)$ das Originalbild und $H(u, v)$ repräsentiert die sogenannte Modulationsübertragungsfunktion. Das reelle Bild wird durch Linsenverzerrung, nichtideale Sensoren, etc. verändert. Diese Veränderung kann berechnet werden und dementsprechend objektiv rückgängig gemacht werden. Dies geschieht durch das Dewarping (vgl. Kapitel 3.3).

Die Funktion $R(u, v)$ hingegen repräsentiert ein Signalrauschen, welches überall in der Natur vorkommt und insbesondere auch den von der Kamera aufgenommenen Signalen additiv überlagert ist. Da keine Referenz eines Signals (Bildes) ohne Rauschen existiert, ist es nicht möglich dieses

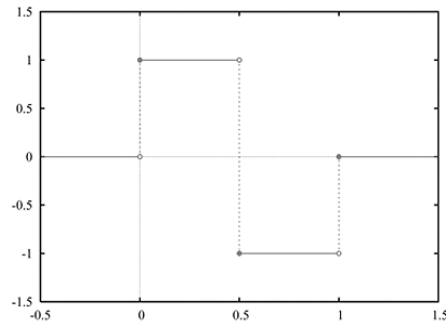


Abbildung 3.7: Haar-Wavelet. Die Skalierung ist variabel.

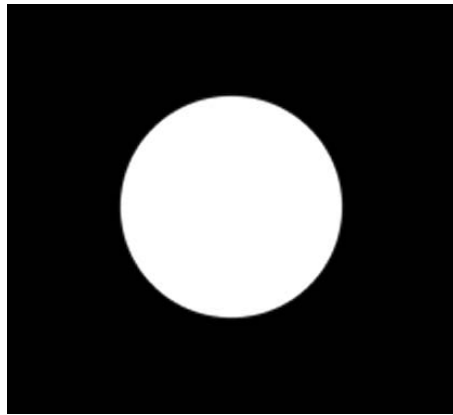


Abbildung 3.8: Ein idealer Tiefpassfilter. Der weiße Kreis im Inneren enthält den jeweiligen Frequenzbereich des transformierten Bildes.

Rauschen zu berechnen und folglich auch unmöglich eine objektive Verbesserung des Signals zu erwirken. Es ist lediglich möglich, das Rauschen anhand subjektiver Kriterien zu reduzieren. Dabei kann das Signal auch grundlegend verändert werden. Die Kriterien für ein subjektiv verbessertes Bild können Einfluss z.B. auf eine spätere Segmentierung haben.

Beim Denoising wird die sogenannte Wavelet-Transformation verwendet (vgl. Kapitel 4.3.9), die ein Bild mit Hilfe einer Waveletfunktion in den Frequenzbereich transformiert.

Die diskrete eindimensionale Wavelet-Transformation einer diskreten Funktion $f : \{0, \dots, n - 1\} \rightarrow \mathbb{R}$ ist folgendermaßen definiert:

$$W(j, s) = \sum_{k=0}^{n-1} f(k) \cdot \psi\left(\frac{k-j}{s}\right), j = 0, \dots, N - 1, 1 \leq s \leq S \quad (3.29)$$

mit gegebener Basisfunktionen („Mutterwavelet“) $\psi : \mathbb{R} \rightarrow \mathbb{R}$.

Die Basisfunktion ist beliebig aus einer Fülle von Funktionen wählbar. In Abb. 3.7 ist als Beispiel das sogenannte Haar-Wavelet gezeigt.

Das Rauschen wird im Frequenzbereich durch einen Tiefpassfilter entfernt. Ein Tiefpassfilter, wie er z.B. in Abb. 3.8 abgebildet ist, wird dabei mit dem Bild im Frequenzbereich multipliziert. Dadurch werden hohe Frequenzen, die das Rauschen größtenteils enthalten, entfernt. Beim Tiefpassfilter kann ein Grenzwert angegeben werden, der anschaulich die Größe des Radius des Kreises in Abb. 3.8 variiert. Außerdem kann angegeben werden, ob der Grenzwert vom Typ Hard- oder Softthreshold sein soll (vgl. 3.4.3).

Abb. 3.9 zeigt einen Teil des Lamellenrandes eines Sprühkegels, wobei das linke Bild noch Rauschen

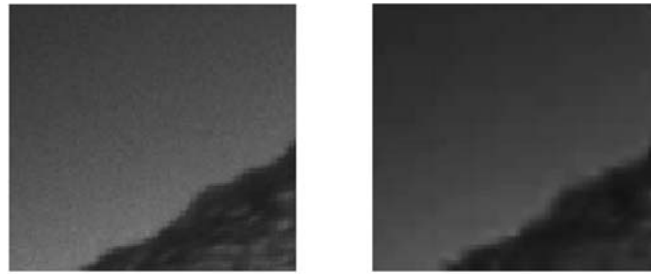


Abbildung 3.9: Lamellenrand (kontrastverstärkt), links mit Rauschen, rechts rauschreduziert

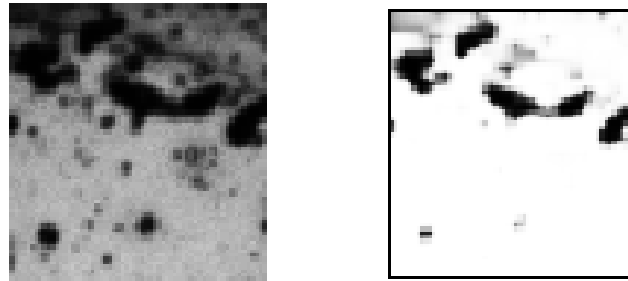


Abbildung 3.10: Tropfengebiet (kontrastverstärkt), links mit Rauschen, rechts rauschreduziert

enthält. Auf diesem Bild wurde mit der Denoising-Funktion von **3D-AugTracer** eine Rauschreduzierung durchgeführt. Das Ergebnis ist auf dem rechten Bild von Abb. 3.9 zu sehen.

Genauso wurde auch bei Abb. 3.10 vorgegangen. Die Abbildung zeigt ein Tropfengebiet unterhalb der Lamelle des Sprühkegels. Auf das linke Bild wurde die Rauschreduzierung angewandt, wodurch sich als Ergebnis das rechte Bild ergab. Man sieht deutlich, dass der Hintergrund in beiden Resultatbildern gleichmäßiger ist als in den Originalbildern.

Sowohl in Abb. 3.9 als auch in Abb. 3.10 erkennt man subjektiv eine Verbesserung des Bildes nach der Rauschreduzierung. Die Struktur im Hintergrund ist gleichmäßiger als vor der Bearbeitung. Es ist möglich, dass dadurch die weitere Bearbeitung der Bilder bessere Ergebnisse liefern wird.

3.4.3 Parameter

Für die Ausführung der Denoisingfunktion werden einige Parameter benötigt. Diese Parameter können vom Benutzer variiert werden, um Experimente zu ermöglichen und die Anwendung der Rauschreduzierungsfunktion auf unterschiedlichen Eingabebildern zu ermöglichen. Diese Parameter werden im Folgenden erläutert.

Wavelet Es stehen zwei Wavelettypen zur Auswahl: das Haar (1,1)-Wavelet (siehe Abb. 3.7) und das CDF (2,2)-Wavelet (siehe Abb. 3.11).

Transformationstiefe Die Transformationstiefe gibt an, wie oft die Vorwärts- und Rückwärts-Transformation mit der Wavelet-Transformation durchgeführt werden soll. Sinnvolle Werte liegen hierbei zwischen 1 und 5.

Threshold Der Threshold ist die obere Grenze, bis zu dem alle Werte im Frequenzbereich auf null gesetzt werden.

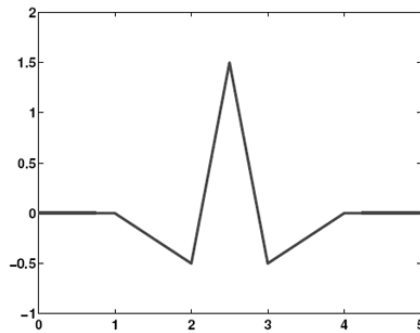


Abbildung 3.11: Cohen-Daubechies-Feauveau-Wavelet(2,2) (CDF). Die Skalierung ist variabel.

Thresholdtyp

- Hard-Thresholding: beim Hard-Thresholding werden alle Werte, die kleiner als der Threshold sind, auf null gesetzt.
- Soft-Thresholding: beim Soft-Thresholding wird von dem Betrag der Werte, die nicht auf Null gesetzt wurden, der Threshold subtrahiert.

3.4.4 Anmerkungen

Zu Beginn der Entwicklung der Denoisingfunktion wurde auf die WAILI-Bibliothek der Universität Leuven (Belgien) zurückgegriffen. Diese Bibliothek war zunächst leider nur für Linux verfügbar. Alle Versuche, die Bibliothek auch unter Windows nutzbar zu machen, schlugen fehl. Daraufhin haben die Entwickler den WAILI-Quellcode, der in der Programmiersprache C vorlag, an die für die PG genutzte Sprache C++ angepasst. Somit war es möglich, für die Rauschreduzierung der Bilder sämtliche Funktionen der WAILI-Bibliothek, sowohl unter Windows als auch unter Linux, zu verwenden.

3.5 Fazit

Es zeigte sich, dass das Entzerren der aufgenommenen Bilder nur wenige Änderungen ergaben, wodurch die Qualität der Kameras bestätigt wird. Die Rauschreduzierung hingegen ergab deutliche Verbesserungen, so dass mit den verbesserten Aufnahmen eine genauere Berechnung weiterer Daten erzielt werden kann.

Kapitel 4

Segmentierung

Inhalt

4.1	Einleitung	28
4.2	Lamellen-Segmentierung	28
4.2.1	Einleitung	28
4.2.2	Histogrammausgleich	28
4.2.3	Sobeloperator	31
4.2.4	Binarisierung	31
4.2.5	Schwellenwertverfahren nach Otsu	32
4.2.6	Template Matching	33
4.2.7	Kantentracking	34
4.2.8	Fehldetektionen	34
4.2.9	Ausgleichsgerade	36
4.2.10	Sprühwinkelbestimmung	39
4.2.11	Lamellenabriss	41
4.2.12	Zusammenfassung, Ausblick	41
4.3	Frequenzanalyse	42
4.3.1	Einleitung	42
4.3.2	Auswertung der Silhouette	42
4.3.3	Amplitudendarstellung	43
4.3.4	Metrik	45
4.3.5	Gemittelte Amplitudendarstellung	45
4.3.6	Eingesetzte Analyse-Methoden	46
4.3.7	Fourier-Transformation	46
4.3.8	Gefensterter Fourier-Transformation	46
4.3.9	Wavelet-Analyse	48
4.3.10	Frequenz-Spektrogramm	50
4.3.11	Wavelet-Spektrogramm	51
4.3.12	Amplitudenverdoppelung	52
4.3.13	Amplitudenvervierfachung	53
4.3.14	Kumulierte Amplituden	54
4.3.15	Kumulierte Frequenzen	55
4.4	Tropfenanalyse	56
4.4.1	Einleitung	56
4.4.2	Hough-Transformation	56

4.4.3	Template Matching	57
4.4.4	Tropfenerkennung im Stereo-Modus	57
4.4.5	Formfaktoranalyse	58
4.4.6	Effizienzsteigerungen	59
4.4.7	Weitere Ansätze	59
4.4.8	Ergebnisse	60
4.5	Wavelet-Segmentierung	61
4.5.1	Grundlagen	61
4.5.2	Verfahren zur Segmentierung	63
4.5.3	Energieberechnung	63
4.5.4	Hölder-Exponent	63
4.5.5	Scanline Verfahren	64
4.6	Haralick	64
4.7	Fazit	66

4.1 Einleitung

Die korrigierten Bilder (vgl. Kapitel 3) stehen nun zur weiteren Bearbeitung, wie der Segmentierung, zur Verfügung. Die Segmentierung stellt den Hauptbereich des Programms dar. Der Sprühkegel wird unter verschiedenen Gesichtspunkten analysiert. Zunächst erfolgt die Erkennung der Lamellenkanten und des Markerpunktes, um aus den gewonnenen Daten Sprühwinkel und Sprühkegel für die Visualisierung zu erhalten. Durch die Trackinganalyse kann die Ausbreitungsrichtung der Flüssigkeit erkannt werden. Zudem liefert eine Tropfenanalyse die Häufigkeitsverteilung in Bezug auf den Tropfenradius. Diese Kernbereiche und Analysemethoden werden in den folgenden Kapiteln beleuchtet.

4.2 Lamellen-Segmentierung

4.2.1 Einleitung

Die Aufgabe der Lamellensegmentierung ist es die Lamellenkanten zu extrahieren. Diese Lamellenkanten werden als Funktion zu ihrer Ausgleichsgeraden an die Frequenzanalyse (vgl. Kapitel 4.3) übergeben. Desweiteren werden bei der Lamellensegmentierung die Sprühwinkel bestimmt. Diese werden durch den Winkel der Ausgleichsgeraden zur Senkrechten durch den Düsenmittelpunkt bestimmt. Grundlage für die Lamellensegmentierung bilden die erstellten Bildsequenzen des Flüssigkeitskegels. Die sich auf den Aufnahmen abzeichnenden Silhouetten der Lamelle, die „Lamellenkanten“, werden mit Methoden der digitalen Bildverarbeitung extrahiert.

In dem hier vorliegenden Kapitel werden die Methoden zur Segmentierung der Kanten beschrieben. Die denoisten und dewarpten Bilder werden zunächst einem Histogrammausgleich unterzogen, danach wird der Sobeloperator angewendet um die Kanten im Bild hervorzuheben. Im Anschluss werden die Bilder binarisiert. Auf dem binarisierten Bildern wird ein Template Matching zur Lamellenkantenfindung durchgeführt. Nachdem ein Teilstück der Lamellenkante durch das Template Matching gefunden wurde, wird die Kante durch ein Kantentracking vervollständigt. Durch die Lamellensegmentierung werden die Sprühwinkel und Ausgleichsgeraden der Lamelle bestimmt.

4.2.2 Histogrammausgleich

Ziel des Histogrammausgleichs ist es, im Bild eine Kontrastverstärkung zu erreichen. Erst dann ist ein vernünftiges Ergebnis bei der Berechnung eines Schwellenwertes zu erwarten, da das hier verwendete Verfahren nach Otsu mit statistischen Mitteln das Histogramm analysiert um einen Schwellenwert zu berechnen.

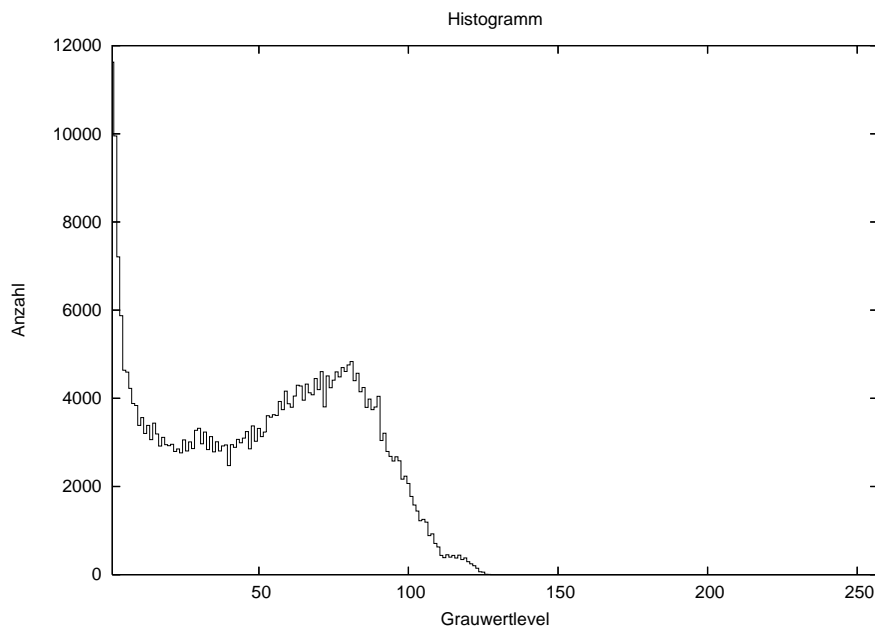
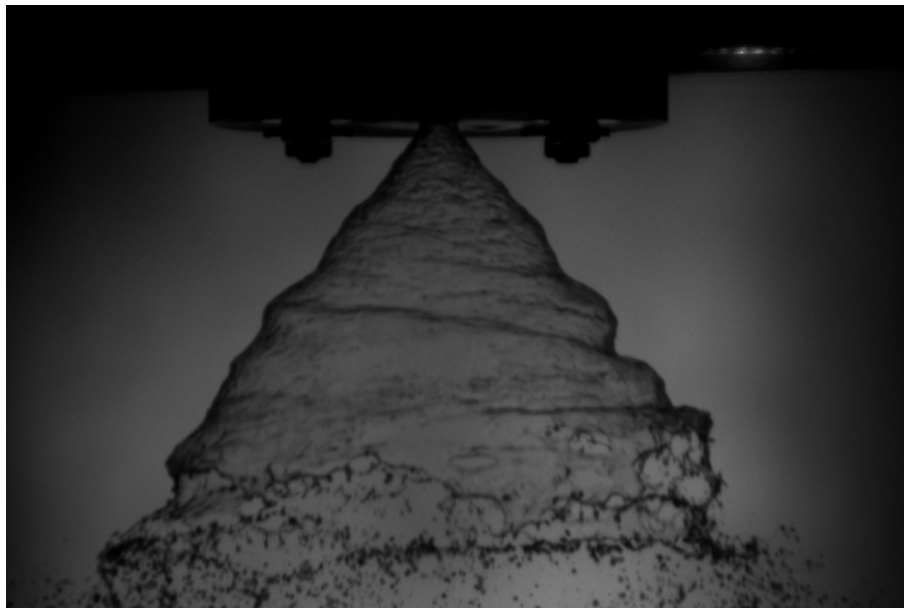


Abbildung 4.1: Bildausschnitt mit Histogramm vor dem Histogrammausgleich

Beim Histogrammausgleich wird das Bild rechnerisch derart umgewandelt, dass alle in ihm vorgefundenen Helligkeitswerte nach der Transformation nicht nur etwa gleich häufig vorhanden sind, sondern darüber hinaus über das gesamte Spektrum der durch die Farbtiefe eines Bildes verfügbaren Werte (hier 256 Graustufen) neu verteilt sind.

Das Histogramm der aufgenommenen Bilder weist einige Charakteristika auf, insbesondere liegen an den hellsten Stellen im Bild Intensitätswerte vor, die nur im mittleren Bereich des Grauwertspektrums liegen (vgl. Abb. 4.1). In dem untersten Grauwertbereich ist ein Peak zu erkennen, der durch die Fülle an Bildpunkten die den Hintergrund repräsentieren zu erklären ist (in diesem Bereich liegen die Häufigkeiten der Intensitäten von 12000 Pixel bis 4000 Pixel), dann fallen die Häufigkeiten mit steigendem Grauwert im Histogramm ab und pendeln sich bei 3000 Pixel ein.

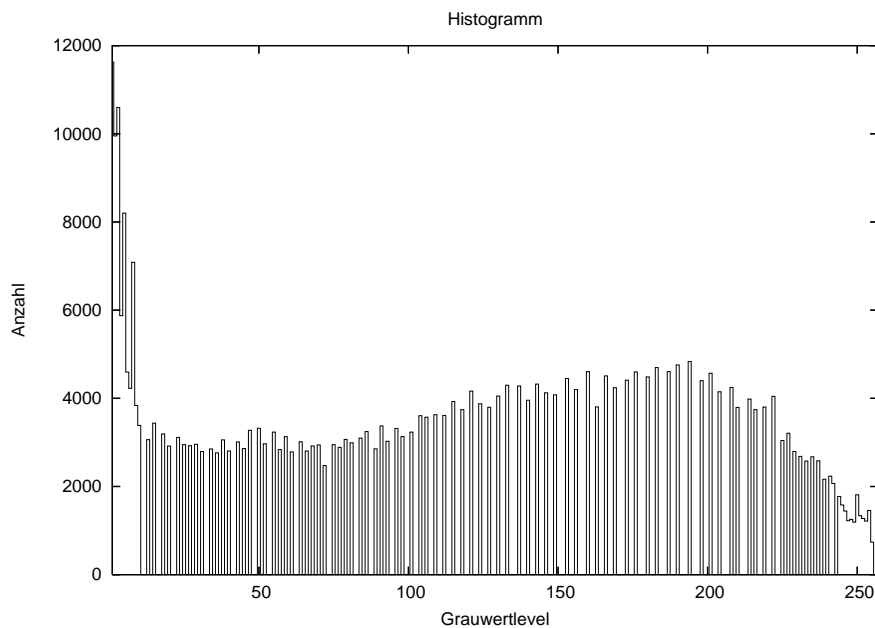
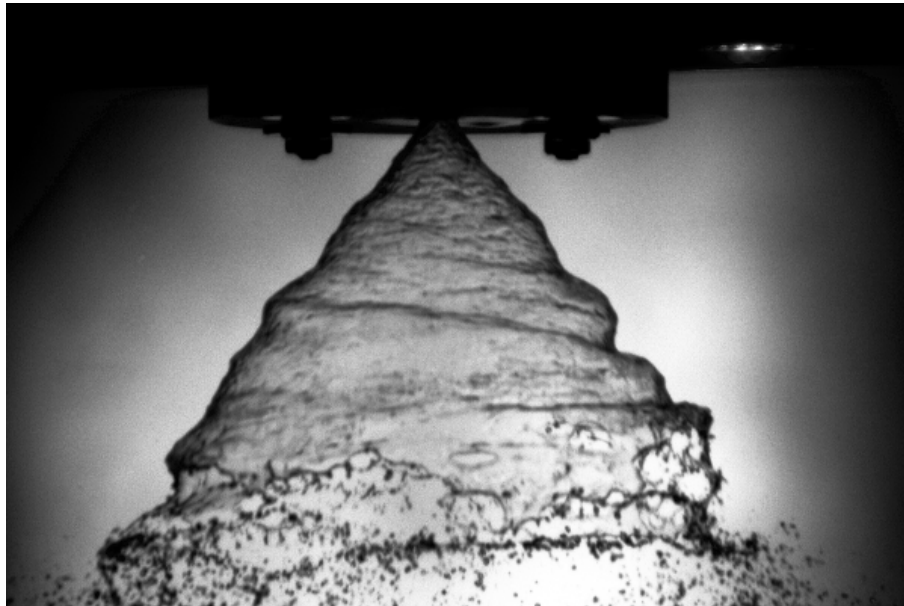


Abbildung 4.2: Bildausschnitt mit Histogramm nach dem Histogrammausgleich

Auffällig ist, dass das Bild ab Intensitäten von ca 125 keine Pixel mit diesen Intensitätswerten mehr aufweist.

Um nun nicht nur den Kontrast zu verstärken, sondern möglichst eine Gleichverteilung der Grauwerte zu erreichen, damit der gesamte zur Verfügung stehende Wertebereich optimal ausgenutzt wird, wird folgendermaßen vorgegangen:

1. Berechnung des Grauwert-Histogramms H (Gibt für jeden Grauwert die Anzahl seines Auftretens im Bild an)

2. Berechnung des kumulativen Histogramms

$$H_k(g) = \sum_{i=0}^g H(i), 0 \leq g \leq 255 \quad (4.1)$$

3. Normalisieren von H_k

$$H_k(g) = \frac{255}{N} \cdot H_k(g), 0 \leq g \leq 255, N = \text{GesamtAnzahlPixel} \quad (4.2)$$

4. Verwenden von H_k als Look-up-Table für die Grauwerte im Bild. Das Ergebnis ist das Ziel-Bild

Nach der Transformation ist fast der gesamte Grauwertbereich abgedeckt (vgl. Abb. 4.2) und eine Gleichverteilung im Histogramm konnte approximiert werden. Tatsächlich wird eine Gleichverteilung in der Praxis beinahe niemals erreicht.

4.2.3 Sobeloperator

Das Ziel ist es aus dem Graustufenbild Kanteninformationen zu extrahieren. Das charakteristische Merkmal einer Kante ist eine lokale starke Kontraständerung über eine relativ kurze Distanz. Diese kann man als Funktion verstehen. Die Änderungsrate einer Funktion bezogen auf die Distanz ist ihre erste Ableitung. Daher werden Kantendetektionsverfahren basierend auf der ersten Ableitung als gradientenbasierte Verfahren bezeichnet. Ableitungen sind ein geeignetes Mittel zur Erkennung von Kanten. Allerdings ist die Verwendung eines einfachen Gradientenoperators anfällig gegenüber Bildrauschen. Es werden häufig zufällige Bildstörungen als Kanten erkannt. Eine Bildglättung unterbindet zufälliges Rauschen. Beim Sobeloperator kommt dafür eine Mittelwertberechnung über die jeweils benachbarten Zeilen bzw. Spalten zum Einsatz. Die Glättung wird orthogonal zur Filterrichtung vorgenommen, d.h. bei Berechnungen des horizontalen Filters wird über die benachbarten Zeilen, beim vertikalen über die Spalten geglättet. Jedoch werden hier die zentrale Filterzeile bzw. Filterspalte doppelt so stark gewichtet wie die übrigen Zeilen bzw. Spalten. Die Filtermasken \mathbf{H} ergeben sich wie folgt:

$$\mathbf{H}_x = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (4.3)$$

$$\mathbf{H}_y = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (4.4)$$

(vgl. Abbildung 4.3, Bild 2 zu Bild 3).

4.2.4 Binarisierung

Die Binarisierung der Bilder dient zur Vereinfachung der Extraktion der Kanten im Bild. Durch die Binarisierung soll den Kanten der Wert Eins (weiß) zugewiesen und der Hintergrund auf Null (schwarz) gesetzt werden. Mit Hilfe eines Schwellenwertverfahrens, hier das Schwellenwertverfahren nach Otsu (vgl. Kapitel 4.2.5), soll jeder Bildpunkt zu einem Segment durch den Vergleich des Grauwerts mit einem Schwellenwert zugeordnet werden. Durch den Schwellenwert werden zwei Segmente, Hintergrund und Objekt, gebildet. Hier wird durch den Schwellenwert das Bild in Hintergrund (schwarz) und Kanten (weiß) eingeteilt, (vgl. Abbildung 4.3, Bild 4). Im nun folgenden Unterkapitel (vgl. Kapitel 4.2.5) wird im Detail erläutert, wie der Schwellenwert bestimmt wird.

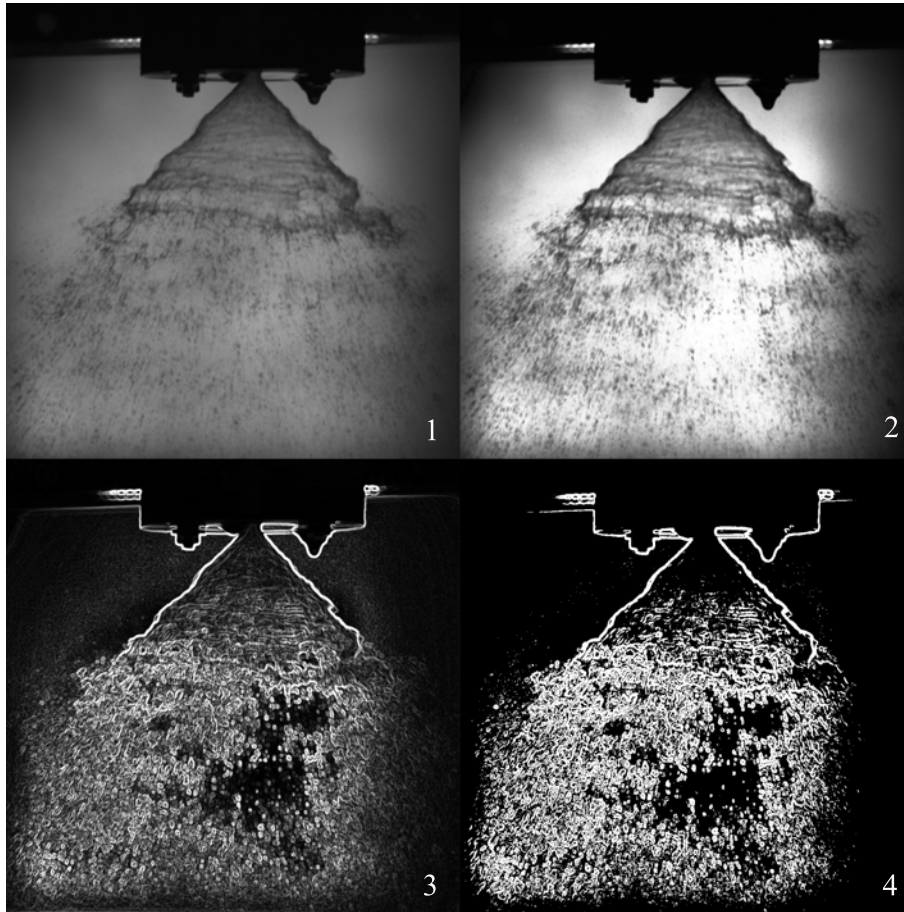


Abbildung 4.3: Bildbearbeitungspipeline. Bild 1 ist das Originalbild, Bild 2 ist kontrastverstärkt, Bild 3 zeigt die Anwendung des Sobeloperators auf das kontrastverstärkte Bild, Bild 4 ist das binarisierte Sobelbild.

4.2.5 Schwellenwertverfahren nach Otsu

Das Schwellenwertverfahren nach Otsu verwendet das statistische Mittel der Varianz und ist ein histogrammbasiertes Verfahren (vgl. Kapitel 4.2.2). Die Beschreibung des Verfahrens orientiert sich direkt an „A threshold selection method from gray level histograms“ von N.Otsu [Ots79]. Ziel des Verfahrens ist es, einen möglichst guten Schwellenwert zu finden, der das Bild in zwei Grauwertklassen, üblicherweise repräsentiert eine Grauwertklasse den Hintergrund und die andere das Objekt, aufteilt. In einem gegebenen Bild werden die N Pixel durch L Grauwertstufen $[1, 2, \dots, L]$ repräsentiert. Durch einen Schwellenwert t , lässt sich jeder Pixel nun jeweils einer von zwei Klassen C_0 und C_1 (Hintergrund oder Objekt) zuordnen. Die Klasse C_0 repräsentiert dabei alle Pixel der Grauwertstufen $[1 \dots t]$ und die Klasse C_1 entsprechend die Pixel mit den Grauwertstufen $[t + 1 \dots L]$. Um einen guten Schwellenwert zu erreichen, wird die Varianz innerhalb der beiden Klassen minimiert, und die Varianz zwischen den beiden Klassen maximiert. Hieraus ergibt sich der Quotient

$$Q(t) = \frac{\sigma_{zw}^2(t)}{\sigma_{in}^2(t)} \quad (4.5)$$

wobei t der gefundene Schwellenwert (Threshold), $\sigma_{zw}^2(t)$ die Varianz zwischen den Klassen und $\sigma_{in}^2(t)$ die Varianz innerhalb der Klassen ist. Die Varianz innerhalb der Klassen wird gebildet aus den Summen der Produkte der Auftretswahrscheinlichkeiten der Grauwertklassen und ihren

Varianzen:

$$\sigma_{in}^2(t) = P_0(t) \cdot \sigma_0^2(t) + P_1(t) \cdot \sigma_1^2(t) \quad (4.6)$$

Die Auftrittswahrscheinlichkeit P_0 ergibt sich durch Aufsummierung der Auftrittswahrscheinlichkeiten aller Grauwerte von 1 bis t :

$$P_0(t) = \sum_{g=1}^t p(g) \quad (4.7)$$

Da ein Pixel zu einer der beiden Klassen gehören muss, ist $P_1(t)$ die Umkehrwahrscheinlichkeit zu $P_0(t)$. Die Auftrittswahrscheinlichkeit für die einzelnen Grauwertstufen $p(g)$ wird ermittelt aus dem Histogrammwert zum Grauwert g geteilt durch die Gesamtzahl N an Pixeln. Die Varianzen $\sigma_i^2(t)$ werden durch Aufsummierung des Produktes der quadratischen Abweichung der Grauwerte vom Mittelwert und der Auftrittswahrscheinlichkeit der Grauwertstufe berechnet:

$$\sigma_0^2(t) = \sum_{g=1}^t (g - \bar{g}_0)^2 p(g) \quad (4.8)$$

$$\sigma_1^2(t) = \sum_{g=t+1}^L (g - \bar{g}_1)^2 p(g) \quad (4.9)$$

wobei \bar{g}_i der Mittelwert der Klasse i ist. Als Mittelwert wird in allen Fällen das arithmetische Mittel der Grauwerte genutzt. Analog wird die Varianz zwischen den Grauwertklassen berechnet:

$$\sigma_{zw}^2(t) = P_0(t) \cdot (\bar{g}_0 - \bar{g})^2 + P_1(t) \cdot (\bar{g}_1 - \bar{g})^2 \quad (4.10)$$

wobei \bar{g}_i der Mittelwert der Klasse i und \bar{g} der Gesamtmittelwert ist. Somit kann für jeden Grauwert t der Quotient $Q(t)$ berechnet werden. Der gesuchte Schwellenwert befindet sich am Maximum von $Q(t)$.

4.2.6 Template Matching

Der Grundgedanke des Template Matching Verfahrens besteht darin, zunächst ein Template zu definieren, dessen Maße typischerweise deutlich kleiner als die des zu untersuchenden Bildes sind. Im folgenden Schritt wird dieses Template auf jedes Pixel des Bildes gesetzt und die Übereinstimmung anhand einer Korrelationsfunktion verglichen. Das bedeutet, je ähnlicher der Ausschnitt des Originalbildes mit dem Template ist, desto besser ist die Korrelation. Der Korrelationswert wird an der entsprechenden Stelle in einem neuen Bild gespeichert. Wurde das Verfahren auf allen Pixeln des Originalbildes angewandt, ergibt sich somit ein Bild, dem man Bereiche hoher Übereinstimmung leicht entnehmen kann. Nun gilt es, einen Schwellenwert zu finden, der das Ergebnisbild binarisiert und für jedes Pixel entscheidet, ob die Übereinstimmung hoch genug ist oder nicht.

Hier ist das Ziel des Template Matching einen Punkt der Silhouettenkante der Lamelle zu finden. Auf dem binarisierten Bild wird das Template Matching mit vorgenerierten Kantentemplates durchgeführt.

Im ersten Schritt werden die Kantentemplates erzeugt. Die Templates sind 20 x 20 Pixel groß und sind Binärbilder. Eine Umgebung von vierzig Kantentemplates, d.h. zwanzig Templates für die Auffindung der linken Lamellensilhouettenkante und zwanzig für die Auffindung der rechten Lamellensilhouettenkante, erwiesen sich in verschiedenen Testläufen als robust. Die Silhouettenkante der Lamelle weist gegenüber den anderen detektierten Kanten im Bild eigene Merkmale auf. Die Silhouettenkante verläuft, auch bei verschiedenen Viskositäten und Druck, diagonal über eine längere Distanz im Bild. Diese Eigenschaft wird sich für die Generierung der Kantentemplates zu Nutze gemacht. Es werden Kantentemplates mit schwarzen Hintergrund und weißer Kante mit verschiedenen Steigungen generiert (vgl. Abb. 4.4). Durch die aus der Waveletsegmentierung (vgl. Kapitel 4.5) gewonnenen Informationen über die Markerkoordinate kann man das Bild in linke und rechte Hälfte unterteilen. Die zwanzig verschiedenen Kantentemplates für linke und für die rechte

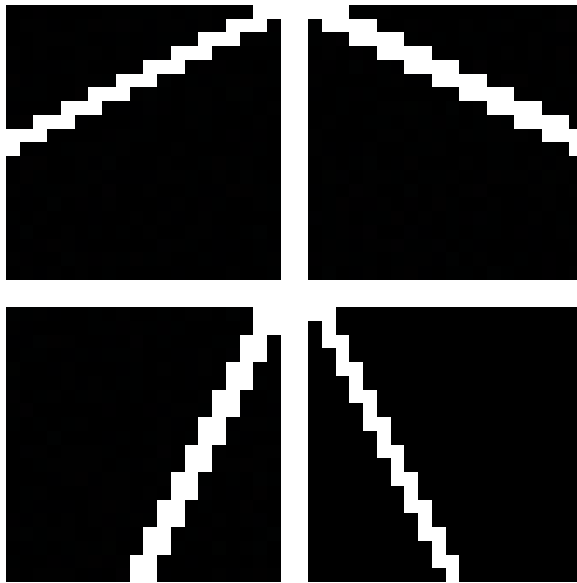


Abbildung 4.4: Vier vergrößerte Kantenemplates mit der jeweiligen höchsten und geringsten Steigung. Die anderen generierten Kantenemplates haben eine Steigung dazwischen.

Lamellenkante werden so über die jeweilige Bildhälfte und unterhalb des Makers geschoben. So werden für jede Position im Bildausschnitt das Abstandsmaß zum Kantenemplate bestimmt. Das Abstandsmaß wird als absolute Differenz gemessen, da auf binarisierten Bilder gearbeitet wird. Die Differenz wird lediglich für die Kante, d.h. weiße Pixel, auf dem Template aufaddiert. Die Position des Templates auf dem Bild mit der geringsten Differenz wird als Startpunkt an das Kantentracking übergeben. Dieses wird in dem nun folgenden Kapitel genauer beschrieben.

4.2.7 Kantentracking

Durch das im vorigen Unterkapitel (vgl. Kapitel 4.2.6) beschriebene Template Matching wird auf den beiden Lamellenkanten ein Startpunkt, der sogenannte „Seed“, für das Kantentracking ermittelt. Ziel des Kantentracking ist es, von diesen Startpunkten, die als 2D-Koordinaten vorliegen, die Silhouettenkanten der Lamelle nach Oben und Unten zu vervollständigen. Dabei wird berücksichtigt, dass nur eine durchgehend verlaufende Kante gute Ergebnisse bei den auf die Lamellensegmentierung aufbauenden Verfahren (Berechnung der Ausgleichsgeraden, Sprühwinkelbestimmung, Frequenzanalyse) ermöglicht.

Die zu extrahierende Silhouettenkante verläuft im Bild diagonal von oben nach unten. Die Kante besteht aus adjazenten Kantenzeilen und diese wiederum bestehen aus einem oder mehreren nebeneinander liegenden weißen Pixeln.

Daraus folgt der Algorithmus um für einen Pixel der Kante die Kantenzeile zu ermitteln, der in Pseudocode beschrieben ist (vgl. Algorithmus 1).

Mithilfe des Algorithmus zur Berechnung einer Kantenzeile ausgehend von einem Seed-Pixel ergibt sich ein rekursiver Algorithmus um die gesamte Kante einer Seite auf der Lamelle zu verfolgen, indem für diese Kantenzeile die in der darunter- und darüberliegenden Zeile anliegenden Pixel als Seed-Koordinaten für neue Kantenzeilen ermittelt werden (vgl. Algorithmus 2).

Daraus ergeben sich die Silhouettenkanten (vgl. Abb. 4.5).

4.2.8 Fehldetektionen

Die aus dem Kantentracking resultierenden Silhouettenkanten weisen Fehldetektionen auf. So werden Teile der Düse und kleine Kanten die von der Silhouettenkante in das Lamelleninnere abzweigen

Algorithmus 1 : berechneKantenzeile(seedX, seedY)

```

Input : seedX, seedY
Output : kantenzeile
begin
  // Suche nach Links
   $x \leftarrow \text{seedX}$ ;
  while pixelwert an position(x, seedY) ist weiß do
    füge pixel(x, seedY) kantenzeile hinzu;
    markiere Pixel(x, seedY) als untersucht;
     $x \leftarrow (x - 1)$ ;
  end
  // Suche nach Rechts
   $x \leftarrow (\text{seedX} + 1)$ ;
  while pixelwert an position(x, seedY) ist weiß do
    füge Pixel(x, seedY) kantenzeile hinzu;
    markiere Pixel(x, seedY) als untersucht;
     $x \leftarrow (x + 1)$ ;
  end
end

```

Algorithmus 2 : Berechne Kante aus Kantenzeile

```

Input : kantenzeile
Output : kante
begin
  foreach pixel aus kantenzeile do
    // Suche anliegende Pixel in der darüberliegenden Zeile
    foreach oberhalb anliegende pixel do
      kantenzeile  $\leftarrow$  berechneKantenzeile (x-Koordinate von pixel, y-Koordinate von
      pixel);
      kantenzeile zu kante hinzufügen;
      berechneKante (kantenzeile);
    end
    // Suche anliegende Pixel in der darunterliegenden Zeile
    foreach unterhalb anliegende pixel do
      kantenzeile  $\leftarrow$  berechneKantenzeile (x-Koordinate von pixel, y-Koordinate von
      pixel);
      kantenzeile zu kante hinzufügen;
      berechneKante (kantenzeile);
    end
  end
end

```

mit extrahiert (vgl. Abb. 4.6). Das Ziel der Lamellensegmentierung ist es nur die Silhouettenkante zu segmentieren. Um die Fehldetektionen beheben zu können, wird eine Baumstruktur pro Lamellenkante aufgebaut, damit gezielt fehldetektierte Kantenstücke gelöscht werden können. Während des Kantentrackings (vgl. Kapitel 4.2.7) wird eine Baumstruktur für jede Lamellenkante aufgebaut. Die Baumstruktur wird so aufgebaut, dass der durch das Template Matching (vgl. Kapitel 4.2.6) gefundene Startpunkt, der sogenannte Seed, Wurzel des Baumes ist. Wird während des

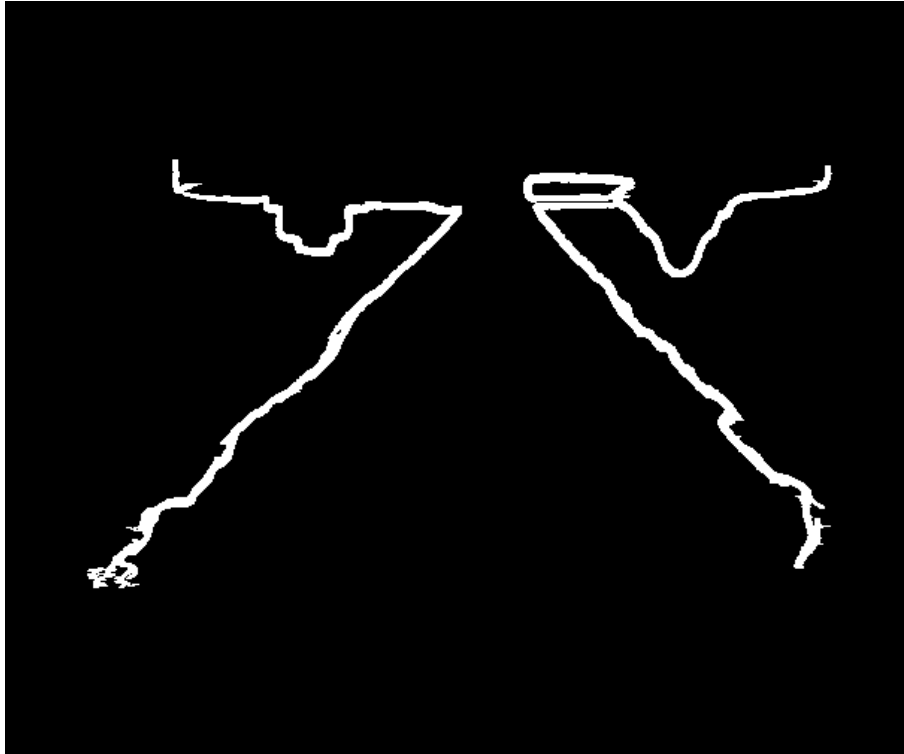


Abbildung 4.5: Die aus dem Kantentracking resultierenden Silhouettenkanten.

Kantentrackings (vgl. Kapitel 4.2.7) in einer Richtung an einer Kantenzeile mehr als nur ein Seed gefunden, so wird für jeden gefundenen Seed ein neuer Ast in den Baum gehängt. Dies wird analog fortgesetzt bis das Kantentracking abgeschlossen ist (vgl. Abb. 4.7). Die Äste werden schon während des Kantentrackings mit Nummern versehen, so dass man sie später gezielt ansprechen kann. In einem ersten Schritt werden die detektierten Inneren Äste gelöscht. Dies sind Äste im Baum, die keine Nachfolger haben.

Die komplexeren Fehldektionen, wie z.B. die Düsenkanten, werden über eine Winkelbestimmung erkannt und eliminiert. Die Lamellenkante weist als charakteristisches Merkmal auf, dass diese Diagonal übers Bild verläuft. Die Kante der Düse verläuft horizontal im Bild. Durch die Winkelbestimmung der Äste, d.h. der Kantenstücke zur y-Achse, kann man die Düsenkante durch einen Winkel um ca. 90° zur y-Achse detektieren und diese mit ihren Nachfolgern löschen. Das Ergebnis sind die Silhouettenkanten der Lamelle (vgl. Abb. 4.8), anhand derer die Ausgleichsgeraden und Sprühwinkel bestimmt werden.

4.2.9 Ausgleichsgerade

Aus den durch das Kantentracking erhaltenen Kantenkoordinaten der Lamelle wird nun eine Ausgleichsgerade in Laufrichtung der Lamelle berechnet. Diese Ausgleichsgeraden, die für jedes Bild der Videosequenz, und zwar für die linke und für die rechte Lamellenkante, getrennt berechnet werden, werden für die Frequenzberechnung (vgl. Kapitel 4.3.2) benötigt.

Die Koordinaten liegen im Array P als 2D Punkte vor: $P(x[k], y[k])$ wobei $x[k]$ die x-Koordinate und $y[k]$ die y-Koordinate des k-ten Punktes im Array ist. Anhand der Koordinaten $x[k]$ und $y[k]$ ($k = 0 \dots n - 1$) ist eine Ausgleichsgerade nach der Methode der kleinsten Fehlerquadrate (die sogenannte Regressionsgerade) zu berechnen:

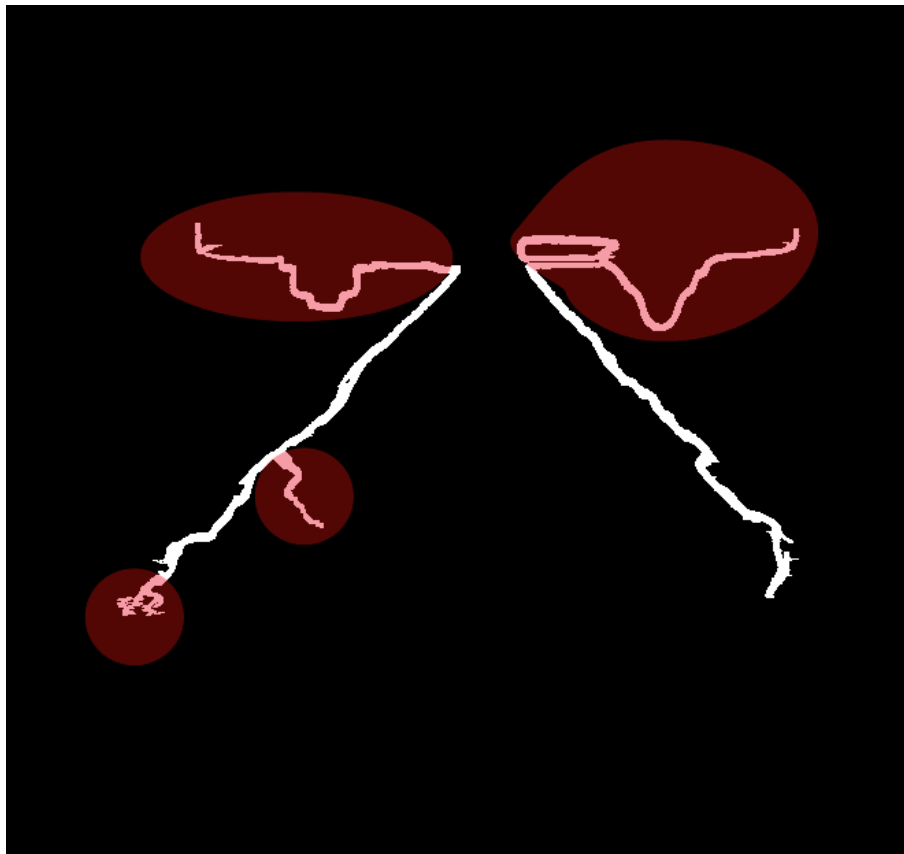


Abbildung 4.6: Ein typisches Beispiel für entstandene Fehldetektionen beim Kantentracking. Die Fehldetektionen sind rot markiert.

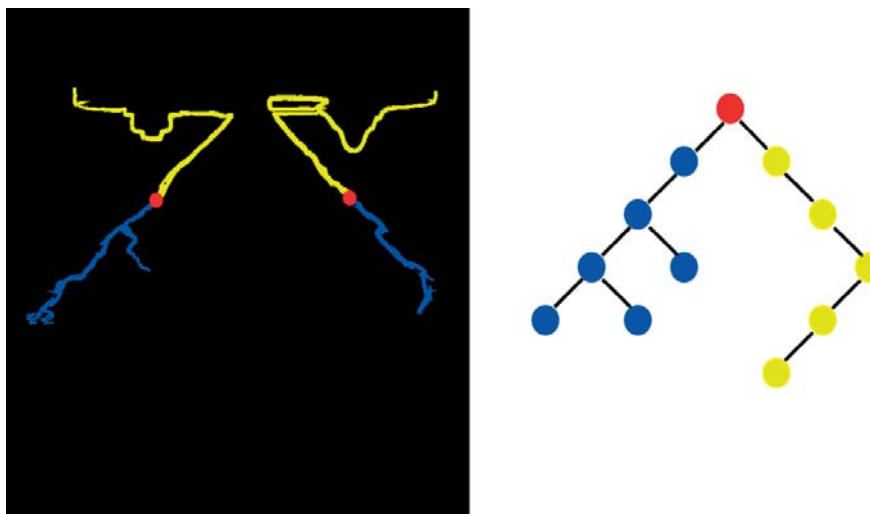


Abbildung 4.7: Beispiel der Baumstruktur der linken Lamellenkante. Der rote Knoten repräsentiert den Seed, die blauen Knoten die Lamellenkantestücke, die nach unten verfolgt werden und die gelben Knoten die Lamellenkantenstücke, die nach oben verfolgt werden.

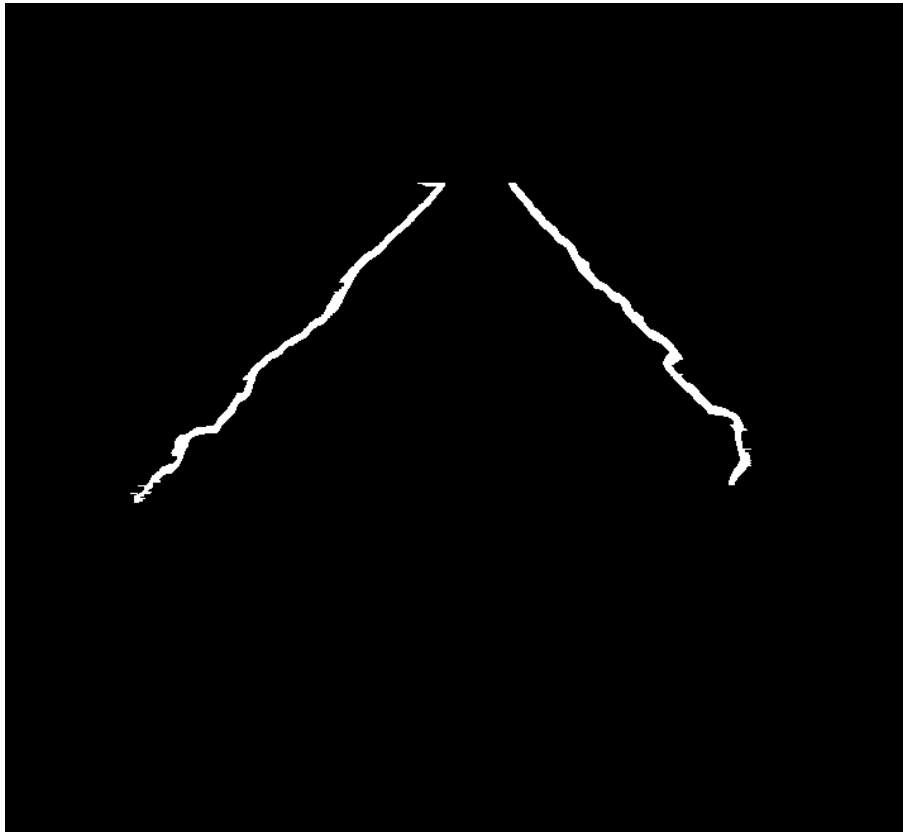


Abbildung 4.8: Die resultierenden Silhouettenkanten nach der Behebung der Fehldetektionen.

1. Zuerst wird das arithmetische Mittel aller x-Koordinaten berechnet

$$xmean = \frac{1}{n} \sum_{k=0}^{n-1} x[k] \quad (4.11)$$

2. Als nächstes bestimmt man das arithmetische Mittel aller y-Koordinaten

$$ymean = \frac{1}{n} \sum_{k=0}^{n-1} y[k] \quad (4.12)$$

3. Im nächsten Schritt bildet man die Summe der Quadrate aller x-Koordinaten

$$ssqx = \sum_{k=0}^{n-1} x[k]^2 \quad (4.13)$$

4. Danach wird die Summe der Produkte der x- und y-Koordinaten berechnet

$$sxy = \sum_{k=0}^{n-1} x[k] \cdot y[k] \quad (4.14)$$

5. Nun erhält man den Anstieg der Regressionsgeraden

$$a = \frac{sxy - n \cdot xmean \cdot ymean}{ssqx - n \cdot xmean^2} \quad (4.15)$$

Algorithmus 3 : Anpassung Ausgleichsgerade

```

Input : ausgleichsgerade, alterStartpunkt, neuerStartpunkt
Output : angepasste Ausgleichsgerade
begin
  neuerStartpunkt als startpunkt für ausgleichsgerade festlegen;
  xDiff ← abstand alterStartpunkt zu neuerStartpunkt in x-Richtung;
  // 3 als Wert für die Konstante C hat sich als ausreichend erwiesen
  berechnen des suchbereich für neuerEndpunkt:
  alterStartpunkt.x - C · xDiff ... alterStartpunkt.x + C · xDiff;
  foreach endpunkt aus suchbereich do
    | neueAusgleichsgerade mit neuerEndpunkt konstruieren;
  end
  foreach neueAusgleichsgerade do
    | zeilenweise abstände zur x-Koordinate der reellen Lamellenkante aufsummieren;
  end
  angepasste Ausgleichsgerade ← neueAusgleichsgerade mit minimaler abweichung zur
  Lamellenkante;
end

```

6. Durch Einsetzen erhält man das Absolutglied (Ordinatenschnittpunkt) der Regressionsgeraden

$$b = ymean - a \cdot xmean \quad (4.16)$$

Die Funktion der Ausgleichsgeraden liegt nun in Form $f(x) = a \cdot x + b$ vor und es lassen sich die benötigten Funktionswerte durch Einsetzen berechnen.

Jetzt können also für jedes Frame aus der Videosequenz für die linke bzw. rechte Lamellenkante die Ausgleichsgerade berechnet werden. Für die Frequenzanalyse ist es von Bedeutung, dass die Amplitudenfunktion (vgl. Kapitel 4.3.3) für jede Lamellenkante auf ein und derselben Ausgleichsgerade berechnet wird. Zu diesem Zweck müssen die Ausgleichsgeraden gemittelt werden. Vorher wird dazu für alle Ausgleichsgeraden ein gemeinsamer Startpunkt berechnet. Dazu wird in jedem Frame der Schnittpunkt der linken und rechten Ausgleichsgerade als Startpunkt der linken und rechten Ausgleichsgeraden festgelegt (dieser Schnittpunkt liegt aufgrund der Konturen der Lamellenkanten ungefähr am Düsenaustritt). Dadurch erhält man in jedem Frame einen Schnittpunkt, der nun noch gemittelt werden muss. Dies geschieht mit dem arithmetischen Mittel der x- und y-Koordinate des Schnittpunktes. Als Alternative wurde der Median der Schnittpunkte berechnet (um mögliche Ausreißer zu unterdrücken), dies brachte aber keine signifikante Verbesserung.

Nun gilt es für jede Lamellenkante eine neue Ausgleichsgerade zu bestimmen, die im berechneten Startpunkt beginnt und die möglichst nah an der berechneten Regressionsgerade liegt. Dies wird im Algorithmus „Anpassung der Ausgleichsgerade“ beschrieben (vgl. Algorithmus 3).

4.2.10 Sprühwinkelbestimmung

Der Sprühwinkel ist ein von den Chemietechnikern benötigter Parameter um das Verhalten von Lamellen zu analysieren. In einem Vorverarbeitungsschritt wird der gemittelte Schnittpunkt aller Ausgleichsgeraden der Bildserie bestimmt. Es wird zunächst ein Gesamtsprühwinkel zwischen linker und rechter Lamellenkante, der Sprühwinkel der linken Lamellenkante und der rechten Lamellenkante zur Senkrechten durch den vorher bestimmten gemittelten Schnittpunkt pro Bild ermittelt. Zudem wird über die Bildserie der gemittelte Gesamtsprühwinkel bestimmt. Weiterhin werden über die Bildserie der minimale und der maximale Gesamtsprühwinkel und der minimale und maximale

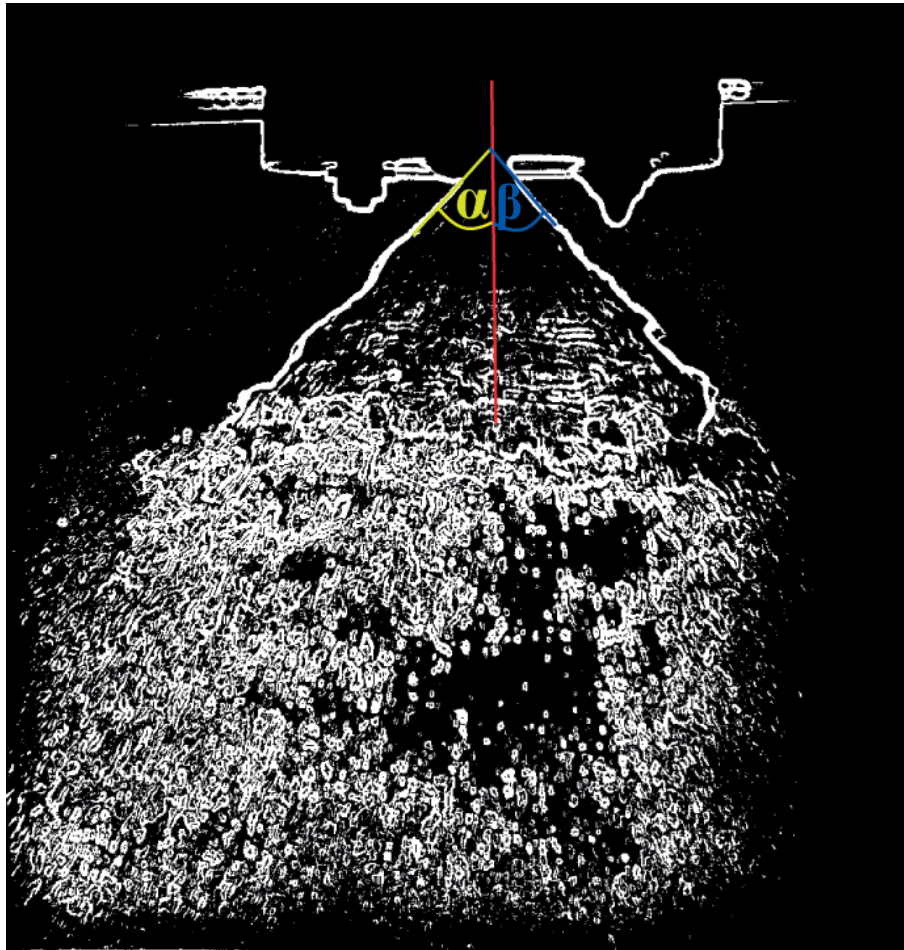


Abbildung 4.9: Winkelberechnung zur Senkrechten durch den Düsenmittelpunkt (rot)

Sprühwinkel pro Lamellenkantenseite ermittelt. Der Sprühwinkel wird aufgrund der Ausgleichsgeraden (vgl. Kapitel 4.2.9) der Lamellenkante zur Senkrechten durch den vorher ermittelten gemittelten Schnittpunkt bestimmt. Mit der trigonometrischen Funktion wird in den so entstehenden rechtwinkligen Dreiecken der Winkel berechnet (vgl. Abb. 4.9).

Der Sprühwinkel α ist der Winkel zwischen der linken Lamellenkante und der Senkrechten durch den gemittelten Schnittpunkt. Der Sprühwinkel β ist der Winkel zwischen der rechten Lamellenkante und der Senkrechten durch den gemittelten Schnittpunkt.

Der Gesamtsprühwinkel γ ergibt sich aus $\alpha + \beta$.

Der gemittelte Sprühwinkel A für die linke Lamellenkante:

$$A = \frac{\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_n}{n} \quad (4.17)$$

dabei ist n die Anzahl der Bilder.

Der gemittelte Sprühwinkel B für die rechte Lamellenkante:

$$B = \frac{\beta_1 + \beta_2 + \beta_3 + \dots + \beta_n}{n} \quad (4.18)$$

dabei ist n die Anzahl der Bilder.

Der gemittelte Gesamtsprühwinkel:

$$\Gamma := \frac{\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_n + \beta_1 + \beta_2 + \beta_3 + \dots + \beta_n}{n} \quad (4.19)$$

dabei ist n die Anzahl der Bilder.

Der minimale Sprühwinkel der linken Lamellenkante A_{min} zur Senkrechten durch den gemittelten Schnittpunkt:

$$A_{min} = \min(\alpha_1, \alpha_2, \dots, \alpha_n) \quad (4.20)$$

Der minimale Sprühwinkel der rechten Lamellenkante B_{min} zur Senkrechten durch den gemittelten Schnittpunkt:

$$B_{min} = \min(\beta_1, \beta_2, \dots, \beta_n) \quad (4.21)$$

Der minimale Gesamtsprühwinkel Γ_{min} :

$$\Gamma_{min} = \min((\alpha_1 + \beta_1), (\alpha_2 + \beta_2), \dots, (\alpha_n + \beta_n)) \quad (4.22)$$

Der maximale Sprühwinkel der linken Lamellenkante A_{max} zur Senkrechten durch den gemittelten Schnittpunkt:

$$A_{max} = \max(\alpha_1, \alpha_2, \dots, \alpha_n) \quad (4.23)$$

Der maximale Sprühwinkel der rechten Lamellenkante B_{max} zur Senkrechten durch den gemittelten Schnittpunkt:

$$B_{max} = \max(\beta_1, \beta_2, \dots, \beta_n) \quad (4.24)$$

Der maximale Gesamtsprühwinkel Γ_{max} :

$$\Gamma_{max} = \max((\alpha_1 + \beta_1), (\alpha_2 + \beta_2), \dots, (\alpha_n + \beta_n)) \quad (4.25)$$

4.2.11 Lamellenabriss

Der Lamellenabriss ergibt sich durch die Endpunkte der Lamellenkanten. Aus diesem Grunde ist auch der Schwellenwert sehr wichtig (vgl. Kapitel 4.2.12). Wird dieser zu niedrig gewählt, wird auch über die schwach zusammenhängenden Kanten, das sich durch ein Überlappen der Kante und Tropfen ergeben kann, die Kante detektiert. In den weiteren Verfahren (vgl. Kapitel 4.4) ist es ggfs. möglich den relevanten zu untersuchenden Bereich (region of interest) der Lamelle initial auszuwählen um z.B. die Laufzeit zu reduzieren. Oberhalb des Lamellenabrisspunktes sind sehr wenige Tropfen zu finden, deshalb lässt sich anhand der Lamellenkanten-Endpunkte und der Ergebnisse der Tropfenanalyse eine genauere Aussage über den Abrisspunkt treffen.

4.2.12 Zusammenfassung, Ausblick

In den vorangegangenen Unterkapiteln wurde beschrieben, wie die Videosequenz analysiert wird und welche Informationen benötigt werden um richtige Schlussfolgerungen zu ziehen und das Verständnis über den Zerfallsprozess des Sprühkegels bei Hohlkegeldüsen zu verbessern. Nun soll erläutert werden unter welchen Aspekten nochmals eine Verbesserung der Ergebnisse der Lamellen-Segmentierung zu erwarten ist.

Es wurde festgestellt, dass mit dem verwendeten Verfahren (vgl. Kapitel 4.2.7) die Lamellenkante vom Seed-Punkt aus zusammenhängend verfolgt werden können, dabei aber auch Fehldetektionen vorgenommen werden (vgl. Kapitel 4.2.8), die in einem Nachbesserungs-Schritt korrigiert werden müssen. In diesem Zusammenhang könnten Aktive Konturen eine Rolle bei der Verbesserung der Kantendetektion spielen.

Die Objektkontur wird hierbei durch eine parametrische Kurve beschrieben. Nach einer (manuellen) Initialisierung wird die Form der Kontur abhängig von sogenannten internen und externen Energien korrigiert. Zur Berechnung der externen Energien wird der Bildinhalt im Bezug zur Position der Kontur genutzt und durch einen Minimierungsalgorithmus wird die Form der Kontur berechnet, bei der die Summe aller Energien minimal ist. Dieses Verfahren ist sehr rechenintensiv, da praktisch durch Ausprobieren die optimale Kontur bestimmt wird.

Ein solches Verfahren würde die Lamelle aber auch über schwierige Stellen im Bild verfolgen (dort wo, z.B. durch einen zu hohen Schwellenwert, die Lamelle nicht mehr durchgehend gezeichnet ist) können.

Desweiteren ist es sehr wichtig einen guten Schwellenwert zu bestimmen. Das Verfahren nach Otsu liefert einen initial guten Schwellenwert. Häufig jedoch ist dieser Schwellenwert noch zu niedrig, so dass durch das Kantentracking die Lamellenkontur auch in das „innere“ der Lamelle verfolgt wird. Dies bedeutet, dass der Schwellenwert überprüft und gegebenenfalls manuell nachgebessert werden muss.

4.3 Frequenzanalyse

4.3.1 Einleitung

Hohlkegeldüsen erzeugen aufgrund ihrer Geometrie und der Parameter der eingesetzten Stoffe unterschiedliche Lamellenausprägungen. Einige Merkmale sind bereits mit bloßem Auge ersichtlich, wie z.B. die ungefähren Abrisspunkte der Zerfallsprozesse. Für andere Aspekte werden Standfotografien oder Zeitlupen-Videoaufnahmen aus Hochgeschwindigkeitskameras analysiert.

Es wird angenommen, dass die radialen Ausprägungen der Lamelle symmetrisch um den Sprühkegel verlaufen. Dies erlaubt die Auswertung der Lamellenkontur, so dass diese Ergebnisse stellvertretend für den gesamten Kegel dienen können. Die Extraktion der Lamellenkontur lässt sich mit Mitteln der digitalen Bildverarbeitung erreichen und ist Aufgabe der oben beschriebenen Segmentierung (vgl. Kapitel 4.2). In Bezug auf eine durchlaufende Ausgleichsgerade lässt sich die Lamellenkontur als diskrete Funktion über dem Zeitintervall¹ interpretieren. Die abweichenden Konturverläufe ergeben somit die Amplituden der Verlaufsfunktion.

Aus Sicht der Chemietechnik ist es von besonderem Interesse, die auftretenden Frequenzen zu analysieren um somit markante Frequenzbereiche zu identifizieren, die für die Zerfallsprozesse von Bedeutung sein könnten. Ausgehend von der Kontursegmentierung und den dort eingesetzten Bilderkennungsverfahren fallen unter Umständen weitere störende Frequenzen in das Signal ein, deren Auswirkungen möglichst unterdrückt werden sollen. Weiterhin soll untersucht werden, ob sich der Lamellenverlauf über die Zeit verändert. Deshalb werden gesonderte Verfahren eingesetzt, die den sich ausdehnenden Verlauf der Lamelle berücksichtigen.

In diesem Kapitel wird nun die Berechnung der Amplitudenfunktion dargestellt. Im Anschluss erfolgt eine Beschreibung und Diskussion der eingesetzten Analyse-Methoden. Teile des vorliegenden Kapitels orientieren sich an der Diplomarbeit „Bildbasierte Analyse der Silhouetten von Hohlkegeldüsen“ von Thomas Umpfenbach [Ump06].

4.3.2 Auswertung der Silhouette

Die Segmentierung der Lamelle ergibt pro Video-Frame eine topologisch geordnete Punktmenge, die den Verlauf der Kontur repräsentiert. Zusätzlich muss die Ausgleichsgerade so definiert sein, dass sie einen optimalen Schnitt durch den Konturverlauf bildet. Da die Gerade durch einen fixen oberen Punkt verläuft, bleibt somit nur ihr Winkel bezüglich einer senkrechten Geraden als freier

¹Definiert als örtlicher Verlauf der Kontur. Sofern die Video-Frames gemeint sind, wird an entsprechender Stelle darauf hingewiesen

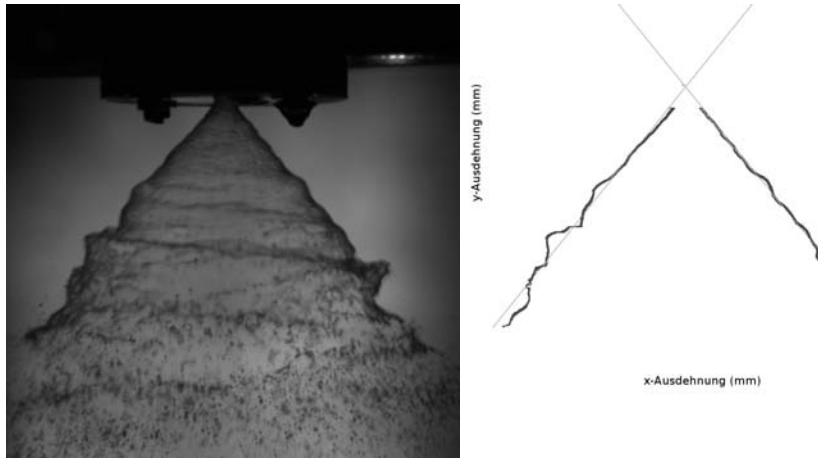


Abbildung 4.10: Sprühkegel und Konturen mit Ausgleichsgeraden

Parameter. Ein Beispiel für einen Sprühkegel und die daraus resultierenden Konturen findet sich in Abb. 4.10.

Es ist anzunehmen, dass der Hohlkegel aufgrund seiner Rotation symmetrisch aufgebaut sein müsste. Die Meßresultate haben diese Annahme widerlegt und zeigen auf jeder Seite leicht unterschiedliche Winkel der Ausgleichsgeraden. Daher ist eine gesonderte Betrachtung der einzelnen Seiten notwendig. Eine weitere Alternative besteht in der Mittelung der jeweiligen Konturverläufe.

4.3.3 Amplitudendarstellung

Aus der Zwei-Punkt-Darstellung der Ausgleichsgeraden folgt ihre Vektordarstellung. Sind zwei Punkte \mathbf{p} , \mathbf{q} gegeben, dann folgt:

$$g : \mathbf{p} + \lambda(\mathbf{q} - \mathbf{p}) = \mathbf{p} + \lambda\mathbf{e} \quad (4.26)$$

Da die Punkte \mathbf{p} , \mathbf{q} in Koordinatendarstellung bekannt sind, lässt sich der Normalenvektor der Ausgleichsgeraden \mathbf{n} berechnen:

$$\mathbf{n} = \begin{pmatrix} -e_y \\ e_x \end{pmatrix} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix} \quad (4.27)$$

normiert:

$$\bar{\mathbf{n}} = \begin{pmatrix} \frac{n_x}{\sqrt{n_x^2 + n_y^2}} \\ \frac{n_y}{\sqrt{n_x^2 + n_y^2}} \end{pmatrix} \quad (4.28)$$

bzw. analog für den Richtungsvektor der Geraden

$$\bar{\mathbf{e}} = \begin{pmatrix} \frac{e_x}{\sqrt{e_x^2 + e_y^2}} \\ \frac{e_y}{\sqrt{e_x^2 + e_y^2}} \end{pmatrix} \quad (4.29)$$

Sei $X = \{(x, y) | (x, y) \text{ bezeichnen einen Punkt im binarisierten Eingabebild}\}$ nun die Menge der Lamellenpunkte. Für X gelten folgende Bedingungen:

1. X hat einen Punkt \mathbf{p} mit kleinster y -Koordinate, so dass \mathbf{p} Anfangspunkt der Kontur ist
2. X hat einen Punkt \mathbf{q} mit größter y -Koordinate, so dass \mathbf{p} Endpunkt der Kontur ist

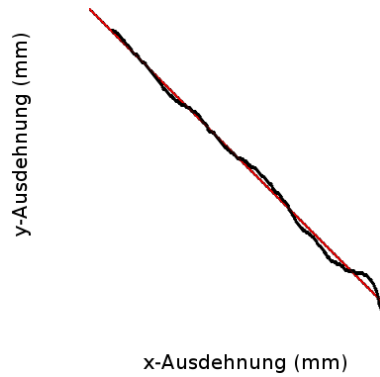


Abbildung 4.11: Kontur mit Ausgleichgerade

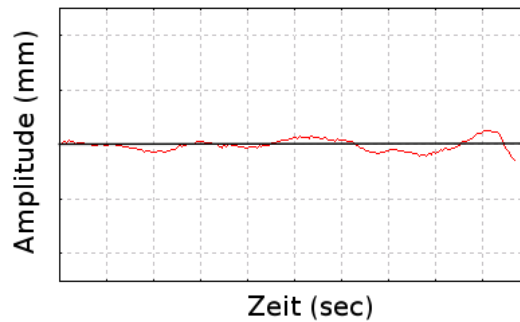


Abbildung 4.12: Korrespondierende Amplitudenfunktion $h(t)$

3. Es gibt für jede Zeile zwischen den Punkten \mathbf{p} und \mathbf{q} genau einen einzigen Punkt, somit werden Hinterschnitte vermieden und weiterhin ist das Zusammenhängen der Konturlinie gesichert.
4. X ist streng topologisch sortiert

$\mathbf{x} \in X$ ist nun ein gegebener Punkt im Eingabebild, für den der lotrechte Abstand h zur Geraden $g : \mathbf{p} + \lambda \mathbf{e}$ zu berechnen ist. Für die Lage von \mathbf{x} gilt somit:

$$\begin{aligned} \mathbf{x} &= \mathbf{p} + \lambda \mathbf{e} + h \mathbf{n} \\ \Leftrightarrow h \mathbf{n} &= \mathbf{x} - \mathbf{p} - \lambda \mathbf{e} \end{aligned} \tag{4.30}$$

beziehungsweise in Gleichungsform

$$\begin{aligned} -he_y &= x_x - p_x - \lambda e_x \\ he_x &= x_y - p_y - \lambda e_y \\ \Rightarrow \lambda &= \frac{x_x - p_x + he_y}{e_x} \end{aligned} \tag{4.31}$$

Einsetzen und Auflösen ergibt

$$h = \frac{x_y - p_y - \frac{x_x e_y}{e_x} + \frac{p_x e_y}{e_x}}{e_x + \frac{e_y^2}{e_x}} \tag{4.32}$$

da $e_x^2 + e_y^2 = 1$ folgt nun für die Höhenwerte $h(\mathbf{x})$

$$h = (x_y - p_y)e_x - (x_x - p_x)e_y \tag{4.33}$$

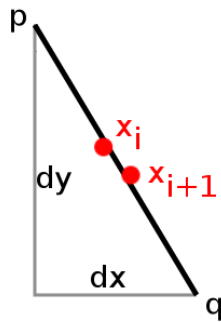


Abbildung 4.13: Skizzierung der Winkelverzerrung

Die Amplitudenfunktion h ermittelt nun zu jedem gegebenen Konturpunkt seinen Abstand zur entsprechenden Ausgleichsgeraden² (vgl. Abb. 4.12).

4.3.4 Metrik

Für die Berechnung der korrekten Messwerte (Wellenlängen, Amplituden, etc.) wird ein Umrechnungsfaktor zwischen Pixel-Distanzen und metrischem Maß benötigt. Die Berechnung dieses Faktors wird in Kapitel 6.6 beschrieben. Da sich die analysierte Lamellenkontur am äußeren Rand des Sprühkegels befindet, kann die vereinfachte Metrik benutzt werden, die auf der zur Blickrichtung orthogonalen Ebene überall gleich ist.

Zusätzlich zu dem metrischen Faktor muss ein Faktor für die Winkelverzerrung berücksichtigt werden. Die zur Eingabe verwendeten Längenangaben beziehen sich zunächst nur auf die Pixelabstände, für die die Ausgleichsgerade das Bezugssystem darstellt. Da die Konturpunkte nur als eindimensionales Array vorliegen, muss das Steigungsverhältnis der Ausgleichsgeraden benutzt werden, um die metrische Verzerrung der Pixellängen zu korrigieren (vgl. Abb. 4.13).

Es gilt somit für den waagerechten metrischen Abstand zweier Punkte aus der Amplitudenfunktion $h(t)$:

$$d(\mathbf{x}_i, \mathbf{x}_{i+1}) = \sqrt{1 + \left(\frac{\Delta x}{\Delta y}\right)^2} \quad (4.34)$$

4.3.5 Gemittelte Amplitudendarstellung

Da für den Lamellenverlauf eine gewisse Regelmäßigkeit und Symmetrie angenommen wird, wurde die Idee aufgegriffen, eine gemittelte Amplitudendarstellung zu wählen. So lassen sich einseitige Abweichungen der Lamelle auffangen.

Prinzipiell wird für die Darstellung des gemittelten Konturverlaufs lediglich der jeweilige Durchschnittswert der Amplitude gewählt. Die Gesamtlänge der Eingabedaten richtet nach der zahlenmäßig kleineren der beiden Punktemengen.

Es wird darauf hingewiesen, dass die Berechnung mit der gemittelten Amplitudendarstellung fehlerbehaftet ist. Da, wie erwähnt, die Ausgleichsgeraden beider Lamellenseiten unterschiedliche Winkel haben, ergeben sich aus der Berechnung der Metrik ebenfalls unterschiedliche Werte (vgl. Kapitel 6.6). Dies führt zu einer Verzerrung, da die gemittelte Kontur nicht mit einer korrekten Metrik arbeiten kann.

²die Daten wurden aus **3D-AugTracer** exportiert und über **Gnuplot** visualisiert

4.3.6 Eingesetzte Analyse-Methoden

Im Programm **3D-AugTracer** sind einige gängige Verfahren zur Signalanalyse realisiert. Die meisten Verfahren beziehen sich jeweils auf eine wählbare Kontur der Lamelle (z.B. für die Fourier-Transformation oder die Wavelet-Analyse). Diese wird bestimmt durch den entsprechenden Video-Frame, die Auswahl der Kamera (links bzw. rechts im Stereo-Fall) und die Seite der Kontur (linker/rechter Rand des Sprühkegels). Andere Verfahren arbeiten auf einer Serie von Konturen, diese werden lediglich durch die Kamera und die Seite (Definitionen wie oben) bestimmt.

Im Folgenden wird auf den mathematischen Hintergrund der Analyse-Methoden eingegangen und es werden ihre jeweiligen Resultate in Bezug auf die Frequenzanalyse diskutiert.

4.3.7 Fourier-Transformation

Die diskrete eindimensionale Fourier-Transformation (DFT) wird definiert durch:

$$F(k) := \sum_{m=0}^{M-1} f(m) \cdot \exp(-2\pi i \frac{km}{M}) \quad (4.35)$$

Die Berechnung der DFT lässt sich effizient mit dem Fast-Fourier-Transformation-Algorithmus durchführen. Die Implementierung dieser Operationen wird durch die externe FFTW-Bibliothek [FFT] unterstützt.

Anschaulich zerlegt die DFT ein gegebenes Signal aus M komplexen Abtastwerten in ihr diskretes Frequenzspektrum von 0 bis $M-1$ (vgl. Abb. 4.15). Für reelle Eingabedaten wie Intensitätswerten aus Bildern ergeben sich jedoch nur für die ersten $\frac{M-1}{2}$ Frequenzen Werte $\neq 0$.

Zur Verdeutlichung der Methoden wird in Abbildung 4.14 ein synthetisch erzeugtes Signal, bestehend aus überlagerten und zeitlich variierenden Sinus-Schwingungen, dargestellt. Die generierten Amplitudenwerte werden fortan für die weiteren Beispiele verwendet.

Anhand der Ausgabe lassen sich nun dominante Frequenzbereiche erkennen, die für die Zerfallsprozesse relevant sein könnten. Die höheren Frequenzbereiche stellen Rauschen dar, das immanent aus der Konturextraktion mittels Bildverarbeitung folgt.

4.3.8 Gefensterete Fourier-Transformation

Die gefensterete Fourier-Transformation (auch „Short-time Fourier-Transformation“) berücksichtigt im Vergleich zu der normalen Fourier-Transformation zeitliche Veränderungen im Signal stärker. Eine wesentlich stärkere Differenzierung erreichen lediglich Wavelet-Analysen (s.u.). Der Ansatz besteht nun darin, ein gegebenes Signal auf ein Intervall einzuschränken und die Fourier-Transformation (bzw. die DFT) lediglich auf diesem kürzeren Intervall durchzuführen.

Die diskrete Short-Time Fourier-Transformation ist definiert durch:

$$X(m, \omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-m)e^{-j\omega n} \quad (4.36)$$

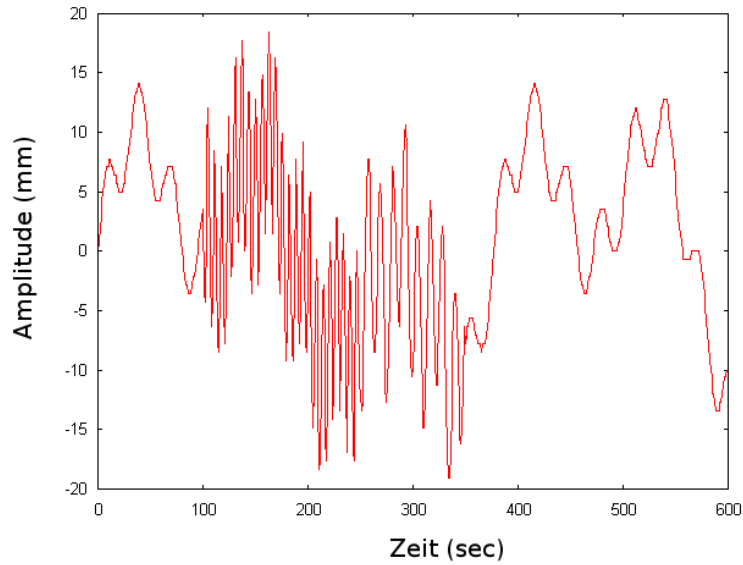
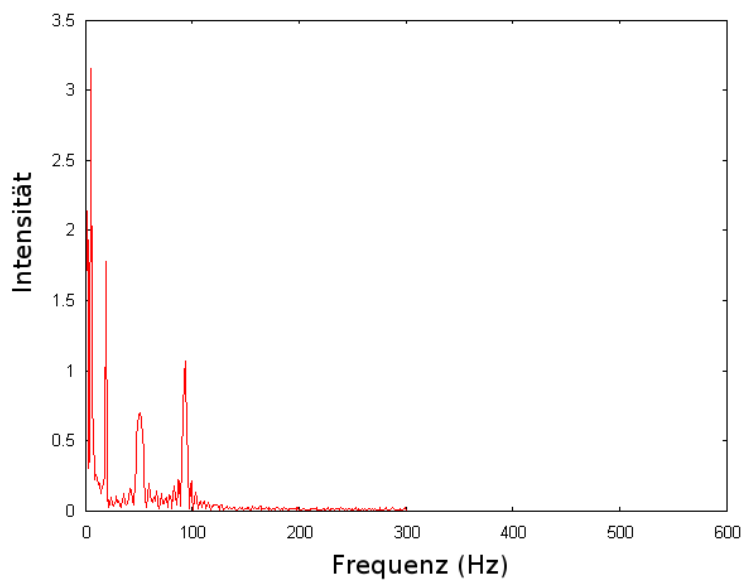
Dabei sind $x(n)$ das Eingangssignal, $w(n)$ die spezifische Fensterfunktion und ω die Frequenz.

Die Einschränkung auf das genaue Intervall erfolgt durch die Definition einer Fensterfunktion, mit der das Signal multipliziert wird. In **3D-AugTracer** sind zwei unterschiedliche Fensterfunktionen $w(t)$ implementiert (vgl. Abb. 4.16):

1. Rechteck-Funktion mit Intervallgröße $[-\alpha; \alpha]$:

$$w(t) = \begin{cases} 1 & \Leftrightarrow \|t\| \leq \alpha \\ 0 & \text{sonst} \end{cases} \quad (4.37)$$

α gibt dabei die links- bzw. rechtsseitige Intervalllänge an, die das Fenster umschließt.

Abbildung 4.14: Amplitudenfunktion $h(t)$ Abbildung 4.15: Fourier-Transformation $F(h(t))$

2. Gauß-Funktion³ mit variabler Standardabweichung σ :

$$w(t) = e^{-\frac{t^2}{2\sigma^2}} \quad (4.38)$$

ω ist dabei die Standard-Abweichung der Gauss-Normalverteilungs-Kurve.

³in **3D-AugTracer** wird für die Gauß-Fensterung zusätzlich noch eine Rechteck-Grenze definiert

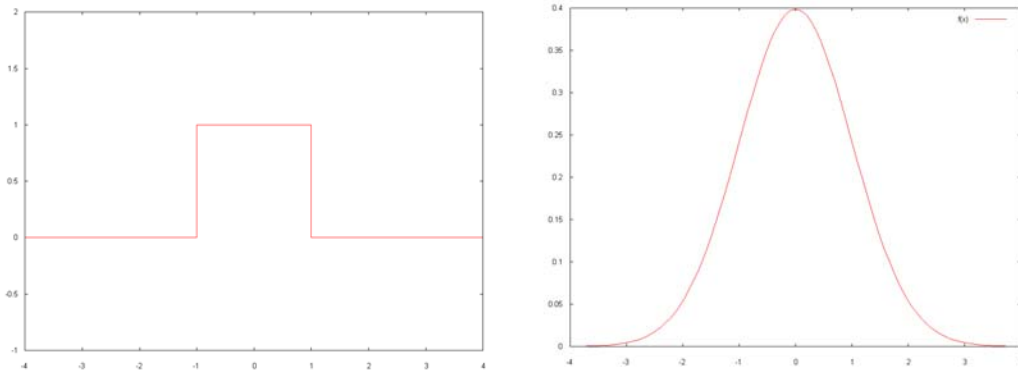


Abbildung 4.16: Rechteck- und Gauß-Fensterfunktion

Die gefensterterte Fourier-Transformation wird nun über einen Teil des Eingangssignals mit Länge L durchgeführt. Abhängig von der Anzahl der Unterteilungen k entstehen somit Intervalle der Länge l_1, l_2, \dots, l_k mit $\sum l_i = L$. Somit ergeben sich jeweils Frequenzbereiche von 0 bis $\frac{2}{l_i}$. In **3D-AugTracer** lassen sich Intervalle gleicher oder unterschiedlicher Länge über dem Eingangssignal definieren. Die resultierenden Spektren werden auf gleiche Länge skaliert und zu einem Gesamtspektrum aufsummiert. In der Ausgabe lassen sich nun die Frequenzen ersehen, die (abhängig von den Intervalldimensionen) in allen Intervallen gleichmäßig, jedoch in stärkerem Ausmaß, hervortreten.

4.3.9 Wavelet-Analyse

Der Nachteil der Fourier-Transformation besteht darin, dass ein Signal stets über die gesamte Zeit analysiert wird. Dadurch entfallen wichtige Informationen über zeitliche Veränderungen des Eingangssignals. Auf der anderen Seite erlaubt die gefensterterte Fourier-Transformation eine feinere Auflösung im Zeitbereich, verliert aber Informationen aus dem Frequenzbereich. Deutlich wird dies, indem man sich die Charakteristik eines Dirac-Impulses betrachtet. Dieser ermöglicht eine unendlich feine Bestimmung des Ortes (der Dirac-Impuls hat eine singuläre, gegen Null konvergierende Ausdehnung), aber er ermöglicht keine Bestimmung der lokal auftretenden Frequenzen. Je größer und ungenauer nun der Zeitbereich gesetzt wird, desto detaillierter lassen sich die Frequenzen bestimmen.

Wavelet-Analysen ermöglichen es nun, ein Signal über variable Längen zu betrachten und geben so ein deutlich besseres Verständnis der zeitlichen Charakteristik wieder. Weitere Grundlagen werden im Kapitel „Wavelet-Segmentierung“ (vgl. Kapitel 4.5) behandelt.

Die Analyse eines Signals mittels Wavelets erfolgt durch die diskrete eindimensionale Wavelet-Transformation einer diskreten Funktion $f : \{0, \dots, n-1\} \rightarrow \mathbb{R}$

$$W(s) = \sum_{k=0}^{n-1} f(k) \cdot \psi\left(\frac{k}{s}\right), 1 \leq s \leq S \quad (4.39)$$

mit gegebener Basisfunktionen („Mutterwavelet“) $\psi : \mathbb{R} \rightarrow \mathbb{R}$ und Skalierung s .

Prinzipiell besteht die Wavelet-Transformation aus einer Faltung des Eingangssignals mit der gewünschten Basisfunktion. Diese wird grundsätzlich durch ihre jeweilige Skalierung s und Translation j beschrieben. In **3D-AugTracer** lassen sich, abhängig vom Wavelet-Typ, weitere Parameter einstellen, auf diese wird explizit in den Wavelet-Beschreibungen eingegangen. Zwei Beispiele für Wavelet-Transformationen sind in Abb. 4.18 und 4.19 ersichtlich. Es lässt sich erkennen, dass die eingesetzte Wavelet-Transformation für die Skalierung $s = 2.5$ weiterhin durch die hochfrequente Rauschüberlagerung gestört wird. Unter der Skalierung $s = 7.5$ entfällt das Rauschen und das dominierende Nutzsignal wird deutlich.

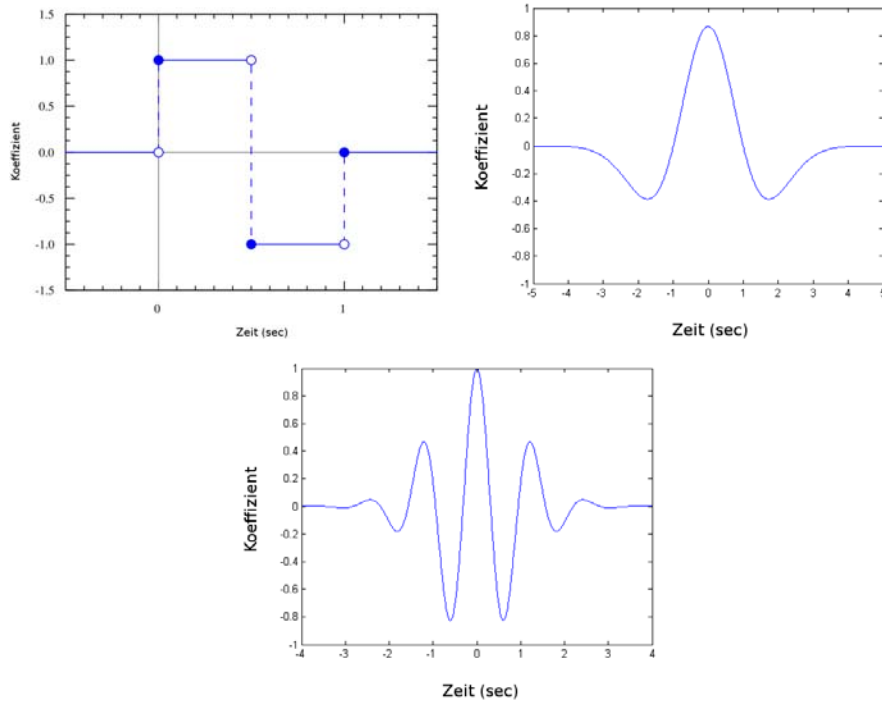


Abbildung 4.17: Wavelet-Typen: a) Haar-Wavelet, b) Mexican-Hat-Wavelet, c) Morlet-Wavelet

3D-AugTracer unterstützt bislang drei Wavelet-Typen (vgl. Abb. 4.17)

1. Haar-Wavelet, $\psi : \mathbb{R} \rightarrow \mathbb{R}$

$$\psi(t) = \begin{cases} 1 & \Leftrightarrow 0 \leq t < \frac{1}{2} \\ -1 & \Leftrightarrow \frac{1}{2} \leq t < 1 \\ 0 & \text{sonst} \end{cases} \quad (4.40)$$

2. Mexican-Hat-Wavelet, $\psi : \mathbb{R} \rightarrow \mathbb{R}$

$$\psi(t) = \frac{2}{(9\pi)^{1/4}} \cdot (1 - t^2)e^{-t^2/2} \quad (4.41)$$

3. Morlet-Wavelet, $\psi : \mathbb{R} \rightarrow \mathbb{C}$ mit gegebener Gauß-Standardabweichung σ

$$\begin{aligned} \psi(t) &= c_\sigma \pi^{-\frac{1}{4}} e^{-\frac{1}{2}t^2} (e^{i\sigma t} - \kappa_\sigma) \\ \kappa_\sigma &= e^{-\frac{1}{2}\sigma^2} \\ c_\sigma &= (1 + e^{-\sigma^2} - 2e^{-\frac{3}{4}\sigma^2})^{-\frac{1}{2}} \end{aligned} \quad (4.42)$$

Die Faltung des Signals mit dem Wavelet im Ortsbereich lässt sich effizient als Multiplikation im Frequenzbereich durchführen. Hierfür wird die FFTW-Bibliothek für die benötigten Fourier-Transformationen eingesetzt. Die nun entstehenden Koeffizienten-Diagramme heben bestimmte Frequenzbereiche im Signal hervor. Somit ergibt sich ein deutlicheres Bild über die charakteristischen Merkmale der Lamellenkontur. Sofern diese ausreichend erkennbar sind, lassen sich unter Umständen bei direkter Betrachtung bereits periodische Frequenzen erkennen.

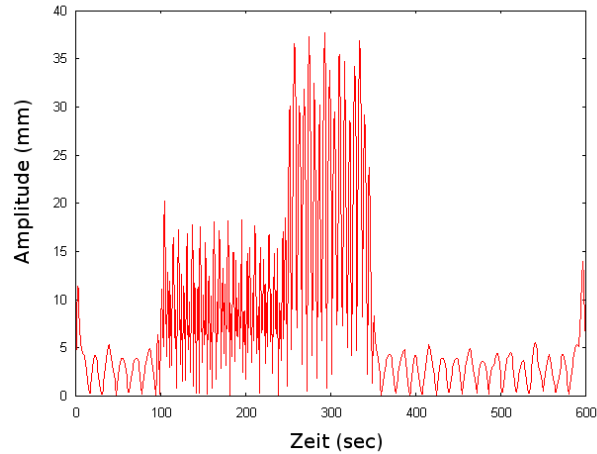


Abbildung 4.18: Wavelet-Transformation mit Mexican-Hat-Wavelet für $s = 2.5$

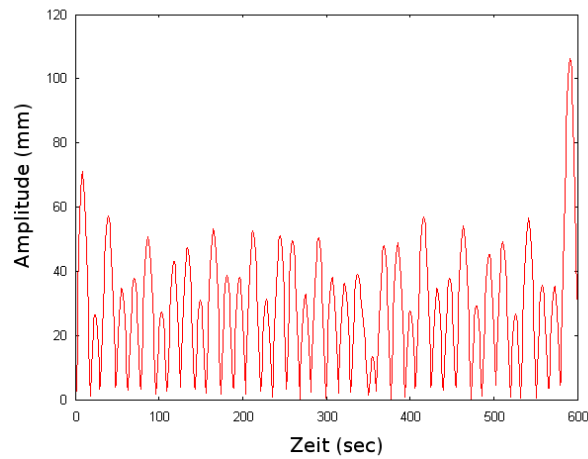


Abbildung 4.19: Wavelet-Transformation mit Mexican-Hat-Wavelet für $s = 7.5$

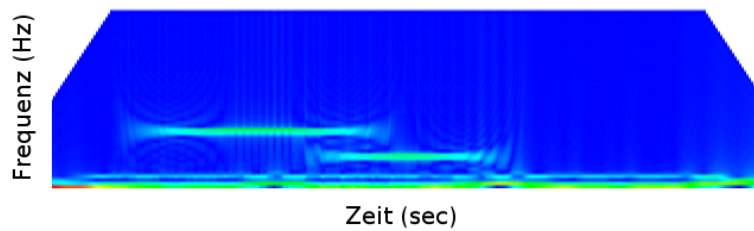
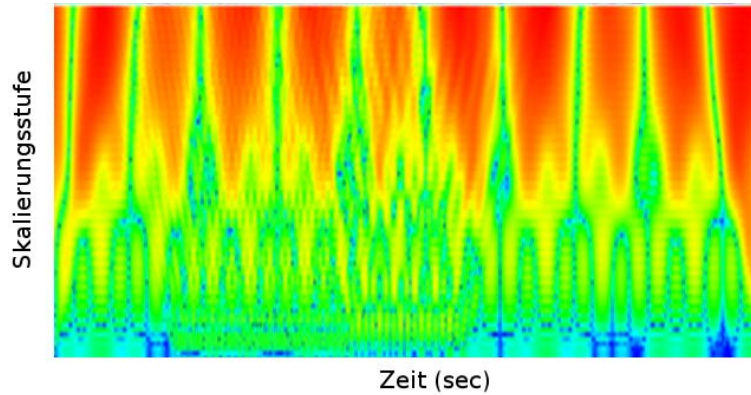


Abbildung 4.20: Frequenzspektrogramm mit $\alpha = 50$ und $\tau = 1$

4.3.10 Frequenz-Spektrogramm

Ein Spektrogramm visualisiert ein Signal kombiniert im Orts- und Frequenzbereich. Es erlaubt die spektrale Darstellung eines Signals über seinen Zeitbereich in einer einzelnen Grafik. Die horizontale Komponente verläuft über das Zeitfenster, jede vertikale Linie in einem Spektrogramm entspricht einem Spektrum zu diesem Zeitpunkt. Der Pegel wird durch Farben oder Graustufen gekennzeichnet.

Abbildung 4.21: Waveletspektrogramm für $j = 0, \dots, 100, \Delta j = 1$

In **3D-AugTracer** werden Frequenz-Spektrogramme über die (einseitige) Intervallgröße α und Schrittweite τ definiert. Bei einer gegebenen Amplituden-Funktion $h(t)$ wird im Prinzip eine gefensterte Fourier-Transformation im Rechteck-Intervall $[x - \alpha; x + \alpha]$ durchgeführt, das entstehende Spektrum wird im Spektrogramm in der Vertikalen an der Stelle t aufgetragen. τ bezeichnet die Schrittweite für die z -Abstände. Im Anschluss werden die Intensitätswerte in Bezug auf die höchste auftretende Intensität skaliert und auf den vorhandenen Farbraum umgerechnet. Eine logarithmische Farb-Umrechnung ist zusätzlich möglich. Ein Beispiel ist in Abb. 4.20 dargestellt. Hier lassen sich deutlich die tiefen, durchgehenden Frequenzen mit hoher Intensität erkennen (unterer Bereich). Zusätzlich erscheinen die zeitweise überlagerten Stör-Frequenzen als waagrecht verlaufender Balken.

Anhand des Spektrogramms lassen sich nun deutlich spektrale Zusammenhänge erkennen. Es ermöglicht die Analyse, ob bestimmte Frequenzbereiche durchgehend über die gesamte Signaldauer erscheinen oder ob diese nur störenden Charakter besitzen.

Es ist zu beachten, dass die Intervallgrößen theoretisch beliebig groß definiert sein dürfen, die auftretenden Randartefakte jedoch mit zunehmender Länge überragen. In **3D-AugTracer** werden längere Intervalle an den Signalrändern abgeschnitten, somit werden an dieser Stelle kürzere Frequenzbereiche ermittelt, die sich bzgl. der ermittelten Wellenlängen nicht mit den korrekten Intervallen vergleichen lassen können.

4.3.11 Wavelet-Spektrogramm

Wavelet-Spektrogramme spiegeln den Verlauf der Koeffizienten der Wavelet-Transformation wieder. Auf der horizontalen Komponente wird analog zu den Frequenz-Spektrogrammen der zeitliche Verlauf des Signals erfasst. Die vertikale Achse entspricht den variablen Skalierungsstufen j der Wavelet-Transformation (vgl. Kapitel 4.3.9).

3D-AugTracer ermöglicht es, eine untere bzw. obere Grenze und die gewünschte Schrittweite für die Skalierungsstufen anzugeben. Im Anschluss werden die Koeffizienten in Bezug auf den höchsten auftretenden Wert skaliert und auf den vorhandenen Farbraum umgerechnet. Eine logarithmische Farb-Umrechnung ist zusätzlich möglich. Ein Beispiel ist in Abb. 4.21 dargestellt. In dem Spektrogramm ist deutlich die Periodizität des Nutzsymbols zu erkennen, diese zeichnet sich durch eine regelmässige waagerechte Strukturierung in den oberen Skalierungsstufen aus. Im unteren Bereich, d.h. für die niedrig eingestellten Skalierungsstufen sind noch keine regelmässigen Signalcharakteristiken erkennbar.

Die Randartefakte am oberen Bereich erscheinen bei zu hoher Wahl der Skalierungsstufen. Die Ursache liegt darin, dass das benutzte Haar-Wavelet einen sehr starken Tiefpass-Charakter besitzt, das auf hohen Stufen ein schwach varriierendes Koeffizienten-Bild ergibt. Desweiteren kann es zu Blockartefakten aufgrund numerischer Ungenauigkeiten kommen.

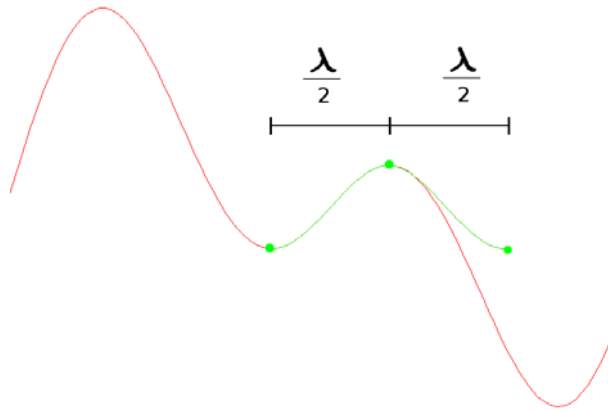


Abbildung 4.22: Prinzip der Amplitudenverdoppelung

4.3.12 Amplitudenverdoppelung

Dieses Verfahren versucht ebenfalls Frequenzen zu bestimmen, die innerhalb eines Signals auftreten. Im Gegensatz zu den oben erwähnten Transformationen, werden einzelne diskrete Frequenzen über Funktionsanalyse und geometrische Interpretation ausgelesen. Die jeweiligen Ergebnisse sind im Vergleich zu den Transformations-Berechnungen deutlicher zu erkennen, da starke Störsignale mit geringem Aufwand eliminiert werden können. Nachteilig hingegen ist, dass die ermittelten Frequenzen nicht notwendigerweise in dem Eingangssignal erscheinen müssen, ausserdem werden große Teile des Spektrums ignoriert. Es wird jedoch angenommen, dass die Ergebnisse relativ ähnlich zu den korrekten Frequenzverteilungen ausfallen.

Zu Beginn der Analyse wird die Punktemenge X der Lamellenkontur auf einige wenige Punkte reduziert. Diese haben die Eigenschaft, dass sie lokale Extrema der diskreten Amplitudenfunktion darstellen.

Sei X die Menge der Konturpunkte, $h(\mathbf{x})$ mit $\mathbf{x} \in X$ die diskrete Amplitudenfunktion.

Dann wird die Differenzfunktion $h'(\mathbf{x})$ definiert durch

$$\begin{aligned} h'(\mathbf{0}) &:= 0 \\ h'(\mathbf{x}) &:= h(\mathbf{x}) - h(\mathbf{x}-\mathbf{1}), \quad x > 0 \end{aligned} \quad (4.43)$$

Analog zu der gewöhnlichen Funktionsanalyse wird die Menge der Extrempunkte E nun als die Menge der korrespondierenden Punkte ermittelt, die die Nullstellen der Differenzfunktion $h'(x)$ darstellen.

Sind \mathbf{p}' , \mathbf{p}'' mit $d(\mathbf{p}', \mathbf{p}'') = 1$ aus der Amplitudenfunktion gegeben und sind $h'(\mathbf{p}') < 0$ und $h'(\mathbf{p}'') \geq 0$, dann gibt es ein \mathbf{p} mit $h'(\mathbf{p}) = 0$. Die Amplitude $h(\mathbf{p})$ entspricht somit einen Extrempunkt im Konturverlauf (für \mathbf{p}' und \mathbf{p}'' einen Tiefpunkt). Der symmetrische Fall mit vertauschtem \mathbf{p}' und \mathbf{p}'' ermittelt die Hochpunkte.

Aus den Abständen der Extrempunkte wird nun die Wellenlänge errechnet. Für den naiven Fall einer einfachen Sinus-Schwingung entspricht die Wellenlänge genau dem Abstand zweier Hoch- bzw Tiefpunkte. Die geometrische Frequenzestimmung benutzt nun jeweils den Abstand zwischen einem Hoch- und einem unmittelbar folgenden Tiefpunkt aus der Menge der Extrempunkte E (bzw. im Anschluss den Abstand zwischen dem letzten Tief- und seinem folgenden Hochpunkt). Nun wird exemplarisch angenommen, dass dieser gemessene Abstand d der halben Wellenlänge einer im Signal auftretenden Schwingung entspricht (vgl. Abb. 4.22).

Sei l die Länge des Eingangssignals, für die ermittelte Frequenz f gilt:

$$f = \frac{l}{2d} \quad (4.44)$$

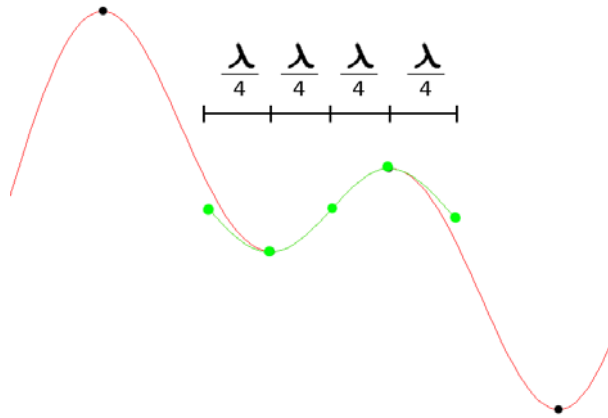


Abbildung 4.23: Prinzip der Amplitudenvervierfachung

Somit wird zu jedem Extrempunkt aus der Lamellenkontur eine Frequenz bestimmt. Die gesammelten Frequenzen werden in **3D-AugTracer** als Häufigkeitsdiagramm dargestellt. Sofern Frequenzen dominieren, erscheinen sie auch im Diagramm als Peaks.

Hinweis: Um das Verfahren besser darstellen zu können, sollte von der Annahme ausgegangen werden, dass eine Sinus-Schwingung periodisch für $\lambda/2$ bzw. λ jeweils erneut 0 ergibt. d entspricht somit den Abständen der Null-Durchgänge.

Als Modifikationen lassen sich zwei Parameter definieren:

1. Amplituden-Schwellenwert $\hat{\alpha}$:
Sind zwei Extrempunkte \mathbf{p} , \mathbf{q} gefunden, so definiert $\hat{\alpha}$ die minimale Amplitudendifferenz die \mathbf{q} erfüllen muss, also $\|h(\mathbf{q}) - h(\mathbf{p})\| \geq \hat{\alpha}$.
2. Wellenlängen-Schwellenwert $\hat{\lambda}$:
Ist eine Wellenlänge λ anhand der Amplitudenverdoppelung bestimmt worden, so muss gelten $\lambda \geq \hat{\lambda}$. Aus praktischer Sicht kann in **3D-AugTracer** $\hat{\lambda}$ nicht kleiner als 2 Pixel sein.

4.3.13 Amplitudenvervierfachung

Die Amplitudenvervierfachung beruht auf dem Verfahren der Amplitudenverdoppelung und entspricht diesem weitestgehend (vgl. Kapitel 4.3.12). Da die ermittelten Frequenzen aus weitaus „unsaubereren“ Methoden bestimmt werden, besitzt dieses Verfahren eine größere Fehlerrate. Positiv ist, dass es im Vergleich zu der Amplitudenverdoppelung feinere Auflösungen erreicht bzw. mehr Frequenzen ermittelt.

Wie beschrieben, wird die Menge der Extrempunkte E ermittelt. Zusätzlich wird die Menge M der lokalen „Nullstellen“ von X berechnet (\mathbf{x} liegt genau mittig zwischen den Extrempunkten).

Sind \mathbf{p}' , \mathbf{p}'' aus der Menge der Extremstellen gegeben, dann gibt es ein \mathbf{p} mit $h(\mathbf{p}) = \frac{h(\mathbf{p}') + h(\mathbf{p}'')}{2}$. Aus den Abständen zwischen diesen Punkten wird ebenfalls die Wellenlänge errechnet. Im Unterschied zu der Amplitudenverdoppelung werden die einzelnen Abstände nun jeweils zwischen einem Extrempunkt und seinem folgenden Nullpunkt bzw. zwischen einem Nullpunkt und seinem folgenden Extrempunkt ermittelt. Der jeweilige Abstand entspricht somit einer Viertel-Wellenlänge (vgl. Abb. 4.23).

Sei l die Länge des Eingangesignals, für die ermittelte Frequenz f gilt:

$$f = \frac{l}{4d} \quad (4.45)$$

Die gesammelten Frequenzen werden in **3D-AugTracer** ebenfalls als Häufigkeitsdiagramm dargestellt.

Als Modifikationen lassen sich analog zu der Amplitudenverdoppelung zwei Parameter definieren:

1. Amplituden-Schwellenwert $\hat{\alpha}$:
Sind zwei Extrempunkte \mathbf{p} , \mathbf{q} gefunden, so definiert $\hat{\alpha}$ die minimale Amplitudendifferenz die q erfüllen muss, also $\|h(\mathbf{q}) - h(\mathbf{p})\| \geq \hat{\alpha}$.
2. Wellenlängen-Schwellenwert $\hat{\lambda}$:
Ist eine Wellenlänge λ anhand der Amplitudenvervierfachung bestimmt worden, so muss gelten $\lambda \geq \hat{\lambda}$. Aus praktischer Sicht kann in **3D-AugTracer** $\hat{\lambda}$ nicht kleiner als 2 Pixel sein.

4.3.14 Kumulierte Amplituden

Um sich einen Überblick über das Amplitudenwachstum der Lamelle zu machen, stellt die kumulierte Amplituden-Darstellung den Verlauf der maximalen Lamellenschwingungen dar. Die untersuchten Video-Aufnahmen lassen ein deutliches Amplitudenwachstum über die Zeit erkennen, d.h. mit zunehmender Entfernung vom Düsenaustritt steigt die Schwingung der zusammenhängenden Lamellenkontur an, bis sie ihren Abrisspunkt erreicht und in Tropfen zerfällt.

Die Berechnung der kumulierten Amplituden erfolgt in mehreren Schritten, die sowohl für jedes einzelne Video-Frame durchgeführt und im Anschluss in einem einzelnen Graph für die gesamte Serie zusammengefasst werden. Aus diesem Grund benötigt die kumulierte Darstellung lediglich die Auswahl einer Kamera und der gewünschten Kontur-Seite (vgl. Kapitel 4.3.2).

1. Seien M_i die Mengen der Konturpunkte, die aus dem Video-Frame i extrahiert wurden. Es wird zu Beginn die maximale Mengengröße m bestimmt, d.h. $m = \max(|M_i|)$. m bestimmt die Länge der Ausgabe.
2. Es sind $E = \{\}$ die kumulierte Punktmenge und H ein Array, $|H| = m$, für die Amplitudenwerte.
3. Berechne für jedes Video-Frame i die Menge der Extrempunkte E_i .
4. Füge alle Mengen E_i in die Menge E ein, so dass E stets topologisch⁴ sortiert bleibt.
5. Sei $j = 0, \dots, m - 1$. Bestimme für jedes j die Menge $\widetilde{E}_{j,\cdot}$ der Punkte aus E , die die gleiche x-Koordinate j besitzen.
6. Berechne den durchschnittlichen Höhenwert der Amplituden

$$H[j] = \frac{\sum_{\forall \mathbf{p} \in \widetilde{E}_{j,\cdot}} h(\mathbf{p})}{|\widetilde{E}_{j,\cdot}|} \quad (4.46)$$

für jede Menge $\widetilde{E}_{j,\cdot} \neq \{\}$, sonst definiere $H[j] := 0$.

In Schritt 3 wird die Menge der Extrempunkte berechnet. Dies erfolgt annähernd gleich zu der Berechnung der Extrempunkte für die Amplitudenverdoppelung (vgl. Kapitel 4.3.12). In **3D-AugTracer** lässt sich für die Berechnung der kumulierten Amplituden ebenfalls ein Amplituden-Schwellenwert $\hat{\lambda}$ definieren, die Bedeutung entspricht der Amplitudenverdoppelung.

Zusätzlich lässt sich eine funktionale Modifikation mitgeben, so dass sich der initiale Amplituden-Schwellenwert $\hat{\lambda}$ über die Zeit verändert. Dazu stehen zwei Optionen zur Auswahl:

1. lineares Wachstum mit Parameter δ , also

$$\hat{\lambda}(t) = \hat{\lambda}(0) + \delta t \quad (4.47)$$

⁴ $p < q \Leftrightarrow p_x < q_x \vee (p_x = q_x \wedge p_y < q_y)$

2. exponentielles Wachstum mit Parameter ϵ , also

$$\hat{\lambda}(t) = \hat{\lambda}(0) \cdot e^t. \quad (4.48)$$

Die Ergebnismenge H ergibt nun das Bild der durchschnittlich maximalen Amplitude definiert über jeden Zeitpunkt des Eingabesignals. Die hiermit resultierenden Konturen sind relativ robust gegenüber einem einzelnen Video-Frame, allerdings steigt die Qualität der Darstellung erst bei größeren Serien. Andernfalls ergeben sich bereits übermäßig viele Nullstellen in der kumulierten Punktmenge, eine Höhenwert-Bestimmung ist somit nicht durchgängig durchführbar.

4.3.15 Kumulierte Frequenzen

Eine weitere Analyse, die über eine Bilder-Serie läuft, stellt die Visualisierung der kumulierten Frequenzen dar. Im Unterschied zu den kumulierten Amplituden, werden hierbei die erscheinenden Frequenzen betrachtet. Das Verfahren ermöglicht es, die Frequenzen so zu präsentieren, dass nicht nur die Frequenzbereiche eines einzelnen Video-Bildes, sondern eben die Verteilung aus einer ganzen Bilder-Serie berücksichtigt wird. Die Frequenzen werden über das Verfahren der Amplitudenverdoppelung (vgl. Kapitel 4.3.12) ermittelt, um sich auf markante Peaks konzentrieren zu können. Auf die Berechnung der rauschanfälligeren Fourier-Transformation des Signals wird verzichtet.

Die Berechnung der kumulierten Frequenzen erfolgt in mehreren Schritten, die für jedes einzelne Video-Frame durchgeführt werden. Aus diesem Grund benötigt die kumulierte Darstellung lediglich die Auswahl einer Kamera und der gewünschten Kontur-Seite (vgl. Kapitel 4.3.2).

1. Seien M_i die Mengen der Konturpunkte, die aus dem Video-Frame i extrahiert wurden. Es wird zu Beginn die maximale Mengengröße m bestimmt, d.h. $m = \max(|M_i|)$.
 m bestimmt die Länge der Ausgabe. Da Wellenlängen für die Frequenzberechnung einbezogen werden, folgt somit, dass die maximale Wellenlänge nur $\frac{m}{2}$ betragen kann (die obere Hälfte der Ausgabe somit =0).
2. Es sind $E = \{\}$ die kumulierte Punktmenge und F ein Array, $|F| = m$, als Ergebnis-Zähler
3. Berechne für jedes Video-Frame i die Menge der Extrempunkte E_i .
4. Bestimme bei einem Video-Frame i den x-Abstand zwischen jeweils zwei Extrempunkten \mathbf{p}_j und $\mathbf{p}_{j+1} \in E_i$, $j = 0, \dots, |E_i| - 2$, also

$$\hat{d} = d(\mathbf{p}_j, \mathbf{p}_{j+1}) \quad (4.49)$$

Berechne daraus die Frequenz

$$\lambda_j = \frac{|E_i|}{2\hat{d}} \quad (4.50)$$

5. Erhöhe $F[|\lambda_j|]$ um 1 für alle $j = 0, \dots, |E_i| - 2$.
6. Wiederhole nun Schritt 4 und 5 für alle weiteren Video-Frames.

In Schritt 4 werden die Frequenzen analog zu der Frequenzberechnung in der Amplitudenverdoppelung berechnet. Aus diesem Grund werden auch für die Berechnung der kumulierten Frequenzen der Amplituden-Schwellenwert $\hat{\alpha}$ und der Wellenlängen-Schwellenwert $\hat{\lambda}$ definiert (vgl. Kapitel 4.3.12). F repräsentiert nun die gesamte Spektralzerlegung der Video-Serie. Bei genügend hoher Bilderzahl lässt sich somit ein deutlicheres Bild über die Frequenzen auf der Lamellenoberfläche machen.

4.4 Tropfenanalyse

4.4.1 Einleitung

Die Tropfenanalyse als Teil der Segmentierung befasst sich mit der Erkennung und Klassifikation der Tropfen unterhalb der Lamelle (Abb. 4.24). Unter Klassifikation wird verstanden, dass für jeden Tropfen der entsprechende Radius zugewiesen wird. Das Ziel besteht darin, eine Verteilung der Tropfenradien für die aktuellen Flüssigkeitsparameter (Substanz und Druck) zu erhalten und durch Veränderung und Auswertung der Einstellungen eine Kombination zu finden, bei der die Verteilung aus chemietechnischer Sicht optimal ist. Vor dem graphischen Hintergrund ist die Erkennung der Tropfen nicht trivial. Tropfen sind selten kreisförmig und unterscheiden sich in Helligkeit, Größe und Sichtbarkeit. Es wurden verschiedene Ansätze zur Lösung der Problematik in Betracht gezogen und analysiert. Zum Schluss werden Ergebnisse vorgestellt und weitere mögliche Ansätze angerissen.

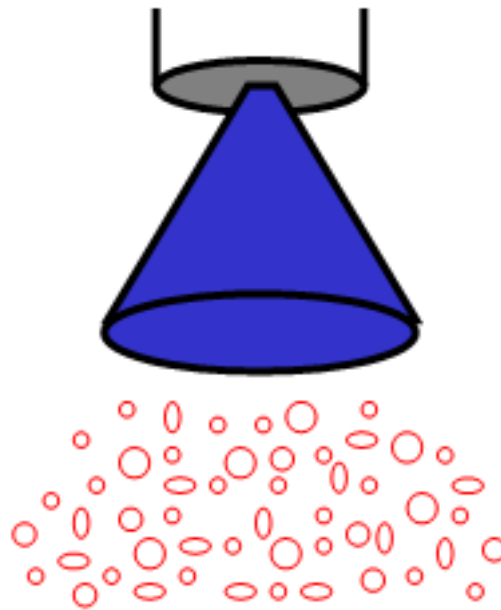


Abbildung 4.24: Tropfen unterhalb der Lamelle

4.4.2 Hough-Transformation

Die Hough-Transformation ist ein robustes globales Verfahren zur Erkennung von Geraden, Kreisen oder beliebigen anderen parametrisierbaren geometrischen Figuren in einem binären Gradientenbild, also einem Schwarz-Weiß-Bild, nach einer Kantenerkennung. Zur Erkennung von geometrischen Objekten wird ein Dualraum erschaffen (speziell: Parameterraum, Hough-Raum), in den für jeden Punkt im Bild, der auf einer Kante liegt, alle möglichen Parameter der zu findenden Figur im Dualraum eingetragen werden. Jeder Punkt im Dualraum entspricht damit einem geometrischen Objekt im Bildraum. Bei einem Kreis der Mittelpunkt und Radius. Danach wertet man den Dualraum aus, indem man nach Häufungen sucht, die dann der gesuchten Figur entsprechen.

In Bezug auf die Tropfenerkennung bestand die Hoffnung, in dem Bild die Tropfen-Umrisse als Kreise zu erkennen. Nach ersten Versuchen stellte sich jedoch heraus, dass das Verfahren für grosse Kreise optimiert ist. Es konnte auch nach unterschiedlichen Einstellungen keine Tropfen erkannt werden, so dass dieser Ansatz verworfen wurde.

4.4.3 Template Matching

4.4.3.1 Template

Das in Kapitel 4.2 erläuterte Template Matching -Verfahren wird nun auch hier angewandt, indem hier als Templates Tropfen herangezogen werden und ein Verfahren entwickelt wird, das für einen gegebenen Radius realistisch aussehende Tropfen generiert. Aufgrund der Helligkeitsunterschiede im Bild durch grösseren Abstand zur Lichtquelle oder teilweise Verdeckungen werden zudem Tropfen mit unterschiedlicher Intensität bestimmt.

In dem implementierten Verfahren werden zunächst grosse Tropfen gesucht, um auszuschließen, einen grossen Tropfen fälschlicherweise als mehrere kleine Tropfen zu identifizieren. Bis zum kleinsten zu untersuchenden Radius wird die Tropfen-Grösse nach jedem Vergleichsschritt vermindert. Um während der Analyse mit unterschiedlichen Grössen keine Tropfen an einer identischen Stelle im Bild mehrfach erkennen zu können, wird eine Matrix verwaltet, die für jedes Pixel bestimmt, ob an dieser Stelle schon ein Tropfen gefunden wurde.

Für möglichst genau generierte Templates wird zunächst der zu untersuchende Bereich auf minimale und maximale Intensität untersucht, innerhalb dessen diese für eine authentische Übereinstimmung generiert werden. Um den Übergang vom dunklen Tropfeninneren zur Hintergrundfarbe zu simulieren, wird ein RadialGradient genutzt, der von Qt zur Verfügung gestellt wird und gute Ergebnisse liefert.

Je nach Tropfengröße wird der zusätzliche Abstand vom Tropfen-Rand zum Template-Rand angepasst, damit auch die Möglichkeit besteht, sich leicht überschneidende Tropfen zu erkennen.

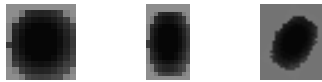


Abbildung 4.25: Templates: Original ; skaliert; rotiert

4.4.3.2 Gradienten-Bild

Eine interessante Variation des Template-Matching-Ansatzes in Bezug auf die Tropfenanalyse wäre es, statt mit den Original-Bildern mit Gradientenbildern zu arbeiten und entsprechende Gradienten-Templates zu erstellen. Ein Gradientenoperator erkennt alle Kanten in einem Bild. Mit dessen Hilfe können Tropfenränder festgestellt werden. Ein großer Vorteil besteht darin, dass die Kanten von der Farbe oder Intensität unabhängig sind, so dass Schwarz-Weiß-Übergänge äquivalent zu Weiß-Schwarz-Übergängen betrachtet werden. Dies hat in Bezug auf die Tropfenanalyse den Vorteil, dass auch unausgeleuchtete Tropfen ausserhalb des Lichtkegels erkannt werden können. Diese sind weiß vor schwarzem Hintergrund, wobei die gut ausgeleuchteten Exemplare schwarz vor weißem Hintergrund koloriert sind. Es kristallisiert sich jedoch ein gravierender Nachteil heraus. Es wird ein Häufigkeitsschwerpunkt gefundener Tropfen mit 500 Mikrometer Radius erwartet. Das Gradientenverfahren erweist sich für Tropfen dieser Dimension als ungeeignet, da derart kleine Kanten als Tropfenumriß nicht erkannt und der Tropfen als Rauschen geglättet und somit nicht mehr erkannt werden kann.

Aus diesen nachteiligen Gründen wird der Ansatz des Gradientenoperators verworfen und durch eine Erweiterung des Template Matching-Verfahrens ersetzt. Um dennoch Tropfen mit unterschiedlichen Intensitäten erkennen zu können, wird der Umweg über modifizierte Templates mit variierender Helligkeit beschritten.

4.4.4 Tropfenerkennung im Stereo-Modus

Im Mono-Modus fehlt bei der Tropfenerkennung ein ganz entscheidender Faktor: die Tiefe der Tropfen und somit deren Position im Raum unter der Düse. Genau diese Information erhalten wir im Stereo-Modus mit Hilfe der 3D-Rekonstruktion (vgl. Kapitel 6). Es wird angenommen, dass im

Mono-Modus viele Tropfen nicht erkannt werden, da diese durch andere Tropfen im Vordergrund verdeckt werden oder im Schatten liegen. Daher erweist es sich als sinnvoll, die 3D-Information zu nutzen, indem man unterhalb der Düse virtuell eine zweidimensionale Ebene spannt und die Tropfen, die hinter dieser Ebene liegen, aus den Ergebnissen entfernt. (Abb. 4.26)

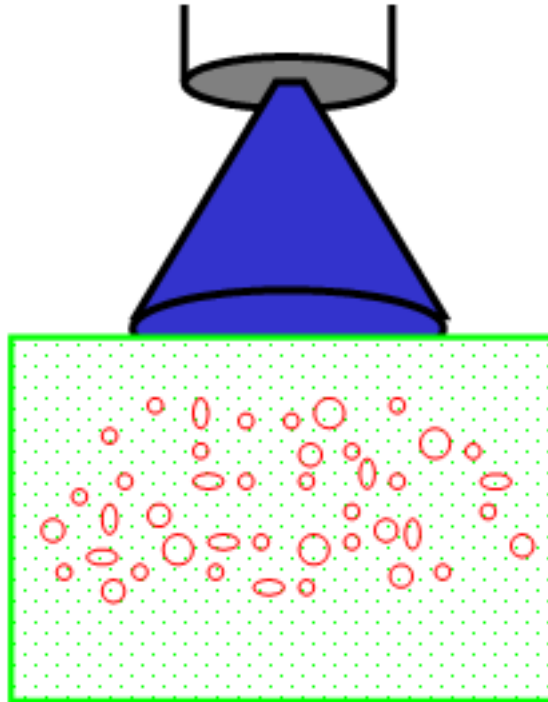


Abbildung 4.26: Tropfen hinter Ebene entfernen

Somit erhalten wir nur die Tropfen, die im Vordergrund vor der Düsenebene liegen und könnten diese Ergebnisse verdoppeln, um ein komplettes und korrektes Bild der Tropfenverteilung haben zu können.

4.4.5 Formfaktoranalyse

Im Rahmen der Formfaktoranalyse wird zunächst eine Intensitätsanalyse vorangestellt. Hierbei wird die kleinste und höchste Intensität des Tropfenbereichs festgestellt. Unterschreitet die Differenz beider Ergebnisse einen Schwellwert, stellt das Verfahren fest, dass es sich mit hoher Wahrscheinlichkeit um eine Fehlererkennung handelt und verwirft den Tropfen, ohne die Analyse fortzuführen, und beginnt mit dem nächsten Kandidaten. Wenn im Gegenteil dazu die Intensitätsdifferenz hoch ist, wird der Tropfen ohne weitere Kontrolle behalten. Für alle weiteren Fälle setzt hier die Formfaktoranalyse ein.

Der Formfaktor setzt sich aus Fläche und Umfang des Objekts zusammen und ist ein Indiz für seine Form. Er berechnet sich aus dem quadrierten Umfang geteilt durch die Fläche, normiert auf die Fläche des Kreises.

Beträgt der Faktor den Wert 1, lässt die Anzahl gefundener Flächen- und Objektpixel auf die Form eines Kreises schließen. Um diesen Formfaktor einfach berechnen zu können, wird nun jeder Tropfen binarisiert und als Binarisierungsschwellwert die Mitte zwischen niedrigster und höchster Intensität gewählt. Eine Auswertung dieser binären Form ist nun im Rahmen der Formfaktoranalyse zu bewerkstelligen und es kann entschieden werden, ob die gefundene Form im gewissen Rahmen

einem Kreis entspricht und der Tropfen behalten wird, oder die Form ein undefinierbares Gebilde darstellt und der Tropfen entfernt wird.

4.4.6 Effizienzsteigerungen

Dem Nutzer bietet sich die Möglichkeit, die Effizienz des Verfahrens zu steuern. Als schnellste Methode wird das bekannte Template-Matching angeboten. Es prüft für jedes in dem gegebenen Radius-Intervall generierte Template in zwei unterschiedlichen Helligkeitsstufen den markierten Bereich der Lamelle. Für die Güte der Ergebnisse wird zudem das Formfaktor-Verfahren ausgeführt. Werden präzisere Ergebnisse erwünscht, können zusätzlich Template-Verzerrungen zur Untersuchung herangezogen werden. Hierbei wird jedes Template um dessen Radius in der Höhe verzerrt. Aufgrund der Tropfen-Geschwindigkeit verformt sich der Tropfen häufig in eine in vertikale Richtung gestreckte Form. Da sich die Tropfen-Flugrichtung zum Lamellenrand hin um einen horizontalen Anteil verstärkt, kann in einem letzten Verbesserungsschritt der verzerrte Tropfen rotiert werden.

Damit sind alle Möglichkeiten des Template Matching ausgeschöpft. Alle Templates werden immer in den unterschiedlichen Helligkeitsstufen generiert.

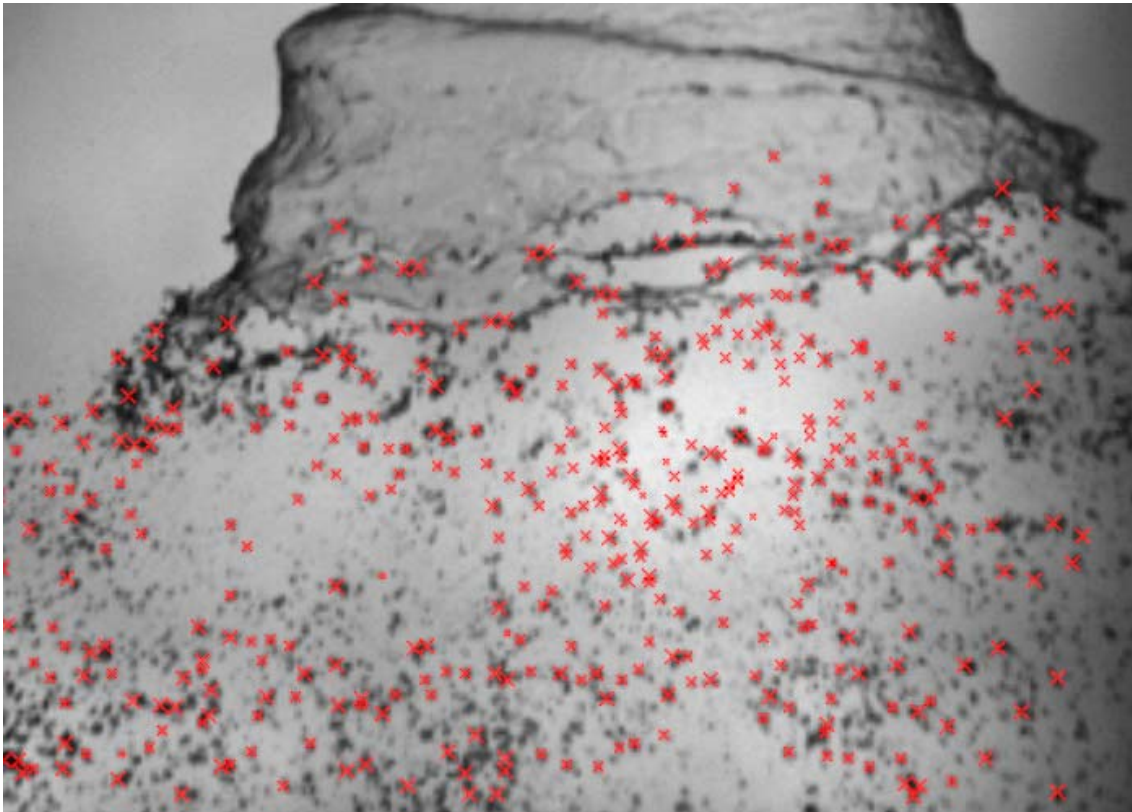


Abbildung 4.27: Erkannte Tropfen

4.4.7 Weitere Ansätze

Snakes sind aktive Konturen, die zur Bestimmung einer Objektkontur angewandt werden. Das Verfahren ist zur Erkennung des Umrißes weniger Objekte entwickelt worden und für Tropfen nicht unbedingt geeignet. Es müssten zu viele Initial-Snakes in dem Bild verteilt werden, die nicht klein genug sein können, um sehr kleine Tropfen zu erkennen. Ein anderer Ansatz in Bezug auf

Snakes wäre es jedoch, diese zu verwenden, um Bereiche auszuklammern, in denen sich sicher keine Tropfen befinden. Im folgenden könnte ein anderes Verfahren die übrigen Stellen untersuchen.

Die im Rahmen der Frequenzanalyse implementierte Ringprojektion (vgl. Kapitel 5.4.4.1) wurde testweise mit unterschiedlichen Parametern auf die Tropfenanalyse angewandt, brachte jedoch keine akzeptablen Ergebnisse. Rotationsinvarianz verbessert Effizienz, da keine rotierten Tropfen mehr separat gesucht werden müssen.

Generell sollte eine Histogrammglättung vorgenommen werden. Das Histogramm des Kamerabildes zeigt, dass die Intensitäten alle im Bereich 0 - 120 liegen, so dass bei einer Glättung auf den gesamten Intensitätsbereich (0 - 255) eine deutlich bessere Extraktion der Tropfen möglich ist. Tests haben ergeben, dass sogar Tropfen zum Vorschein kommen, die mit bloßem Auge gar nicht erkennbar waren, da sie im Schatten lagen und die Intensitätsdifferenz für das Auge nicht extrahierbar war.

4.4.8 Ergebnisse

Testweise wurden 20 Bilder im Mono-Modus in den unterschiedlichen Gütestufen untersucht und folgende Ergebnisse festgestellt:

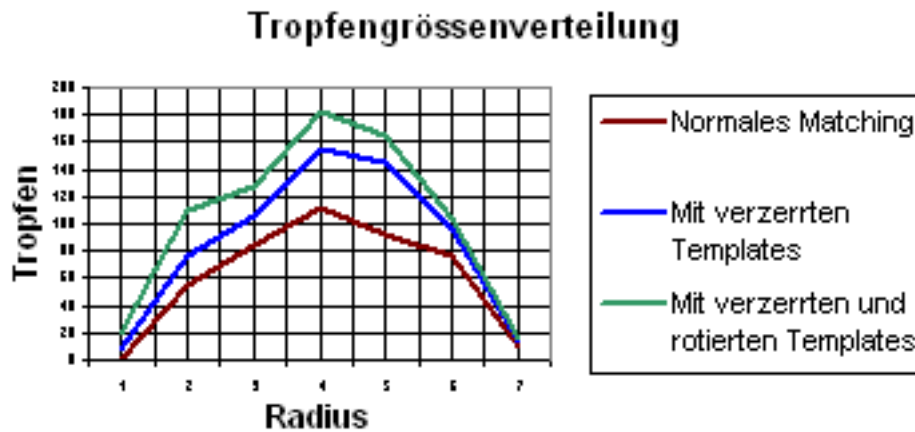


Abbildung 4.28: Tropfenverteilung

Es ist ersichtlich, dass bei zusätzlicher Erkennung verzerter und rotierter Templates zu fast allen Tropfengrößen eine höhere Zahl Tropfen zu finden ist, wobei schon in der einfachsten Methode des Template Matching eine korrekte Verteilung visualisiert werden kann, die später durch Hinzunahme weiterer Templates bestätigt wird.

Es kann bei einem manuell erstellten Testbild mit unterschiedlichen Tropfen festgestellt werden, dass alle Tropfen korrekt erkannt und klassifiziert wurden. Die Güte des Ergebnisses ist stark von den Lichtverhältnissen und Rauschen im Bild abhängig, so dass ein Fehlerbereich schwer bestimmbar ist. Generell kann man sagen, dass sich in verschwommenen und homogenen Bereichen, in denen man mit bloßem Auge nur Tropfen vermuten kann, eine Erkennung als fast unmöglich erweist. Zudem verhält es sich so, dass bei kleinen Tropfen nur wenige Pixel über die Tropfenfrage entscheiden und eine zufällig Tropfen-ähnliche Pixelanordnung schon einen Tropfen falsch identifizieren kann. Je größer der Tropfen, desto wahrscheinlicher ist dessen korrekte Erkennung.

4.5 Wavelet-Segmentierung

4.5.1 Grundlagen

In Kapitel 4.2 und Kapitel 4.4 wurden verschiedene Segmentierungsverfahren vorgestellt, um Tropfen sowie die Lamelle zu segmentieren. Die vorgestellten Verfahren haben gemeinsam, dass die Merkmale bzw. die benötigten Informationen eines Bildausschnitts durch Funktionen auf den dort vorkommenden Grauwerte ermittelt werden.

Alternativ hierzu bietet sich an, das Bildsignal zuerst in einer anderen Form darzustellen, beispielsweise durch eine Orthogonaltransformation. Die Bildinformation ist dort in den Koeffizienten der Entwicklung nach dem Orthogonalsystem enthalten. Eine der bekanntesten Transformationen ist die bereits in Kapitel 4.3 vorgestellte Fourier-Transformation, welche auf dem Orthogonalsystem der Sinus- und Cosinus-Funktionen beruht und die Abbildung des Signals in den Frequenzbereich ermöglicht (Siehe auch [Ach85]).

Eine solche Transformation ermöglicht die Segmentierung nach Merkmalen, welche in der Abbildung enthalten sind.

Die Fouriertransformation liefert jedoch keine Informationen, welche ausschliesslich einen lokalen Bildbereich betreffen. Dies ist dadurch bedingt, dass die Sinus- und Cosinus-Funktionen nur in der Frequenz, jedoch nicht in den örtlichen bzw. zeitlichen Ausdehnung begrenzt sind. Eine Analyse mithilfe der Fourier-Transformation kann also nur Informationen über den Frequenzbereich des vollständigen Bildes liefern.

Um die lokale Ausprägung von Frequenzen im Bild zu ermitteln und für die Segmentierung zu nutzen, kann man alternativ Basisfunktionen verwenden, welche Lokalität sowohl im Orts- als auch im Frequenzbereich aufweisen. Die Wavelet-Transformation [Bla98] nutzt solche Funktionen, die sogenannten Wavelets.

Ein Wavelet ψ ist eine Funktion mit geringer Varianz und kompaktem Träger, deren Integral 0 ergibt. Des weiteren muss die durch Skalierung und Verschiebung entstehende Funktionenmenge das gesuchte Orthonormalsystem bilden.

Ein Beispiel für ein solches Wavelet ist das sog. Haar-Wavelet (siehe Abb. 4.29):

$$\psi_{Haar}(x) = \begin{cases} 1 & 0 \leq x < 1/2 \\ -1 & 1/2 \leq x < 1 \\ 0 & \text{sonst} \end{cases} \quad (4.51)$$

Mit Hilfe dieser Funktion werden nun die gesuchten Waveletfunktionen erzeugt:

$$\psi_{r,k}(t) = 2^{-r/2} \psi_{Haar}\left(\frac{t - k \cdot 2^r}{2^r}\right) \quad r, k \in \mathbb{Z} \quad (4.52)$$

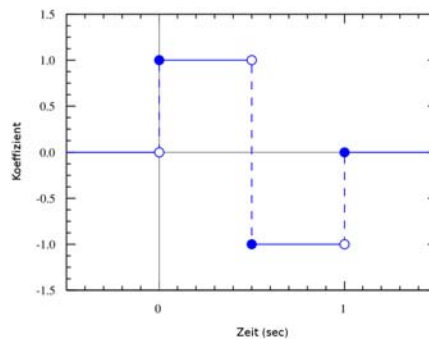


Abbildung 4.29: Haar Wavelet

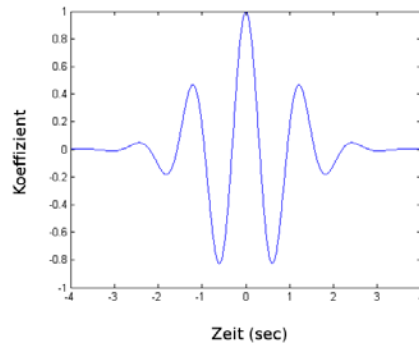


Abbildung 4.30: Morlet-Wavelet

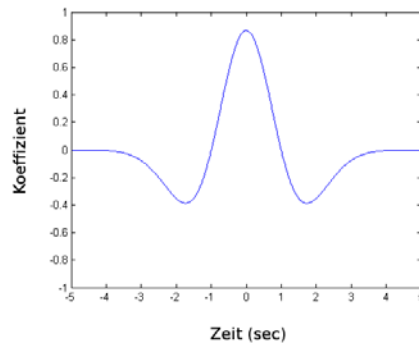


Abbildung 4.31: Mexican Hat Wavelet

Weitere Wavelets sind beispielsweise Morlet-Wavelets (siehe Abb. 4.30) und das Mexican Hat Wavelet (siehe Abb. 4.31).

Die (kontinuierliche) Wavelet-Transformation selber ist über folgende Formel definiert [Bla98]:

$$[W_\psi f](a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \overline{\psi\left(\frac{x-b}{a}\right)} f(x) dx \quad (4.53)$$

Hierbei bezeichnet a die Skalierung und b die Translation des Wavelets. Da die Skalierung des Wavelets auch die Ausdehnung der Basisfunktion innerhalb des Ortsbereichs kontrolliert, wird gleichzeitig zu der untersuchten Frequenz auch die Größe des Fensters bestimmt, in welchem die Transformation stattfindet.

Die diskrete Wavelet-Transformation (DWT) zeichnet sich durch Skalierungen und Translationen aus einem diskreten Wertebereich aus. Durch den diskreten Definitionsbereich wird sie ausserdem über die Summe gebildet:

$$W(j, s) = \sum_{k=0}^{n-1} f(k) \cdot \psi\left(\frac{k-j}{s}\right), j = 0, \dots, N-1, 1 \leq s \leq S \quad (4.54)$$

Üblicherweise werden die Werte j und s so gewählt, dass bei einem Transformationsschritt die Skalierung verdoppelt wird und die Translation halbiert. Bei einem endlichen Signalausschnitt erhöht sich somit die Anzahl der Koeffizienten in Zweierpotenzen. Die Überschneidung der Wavelets wird hierbei verhindert bei Beibehaltung der Rekonstruierbarkeit. Die DWT eignet sich daher für Kompressionsverfahren sowie für das Denoising von Bildern (vgl. Kapitel 3.4).

Im **3D-AugTracer** Projekt wird die Wavelet-Transformation innerhalb der Segmentierung durch

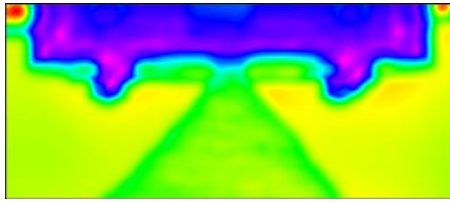


Abbildung 4.32: Hölder-Exponenten an der Düse

Faltung des Wavelets mit dem Bild errechnet. Für eine festgelegte Skalierung erhält man somit ebensoviele Koeffizienten wie Bildpunkte, da der Vorgang einer Translation von einem Pixel entspricht.

4.5.2 Verfahren zur Segmentierung

Das eingesetzte Verfahren nutzt die Wavelet-Transformation zur Ermittlung des Hölder-Exponenten in jedem Bildpunkt (Siehe [VVBF04]). Der Begriff stammt aus der fraktalen Analyse und sagt aus, wie oft ein Signal an einem Punkt stetig differenzierbar ist.

Da die Bilder im unausgeleuchteten Hintergrund sowie auf dem Düsenapparat durch das Gegenlicht sehr wenig Schwankungen im Signal haben, ergeben sich dort im Vergleich zur eher unruhigen Lamelle sowie dem ausgeleuchteten Hintergrund vergleichsweise hohe Hölder-Exponenten (siehe Abb. 4.32).

Durch das Schwellwertverfahren nach Otsu (vgl. Kapitel 4.2.5) kann der Bereich von Werten bestimmt werden, in dem sich die Hölder-Exponenten in diesen Bildteilen befinden.

Da für die Lamellensegmentierung ein Punkt oberhalb der Düse als Ausgangspunkt für die Suche nach Kanten benötigt wird, hat es sich angeboten, einen relativ zusammenhängenden Bereich mit Hölder-Exponenten oberhalb des Schwellwerts zu suchen, der die Breite des Düsenapparates besitzt. Hier kommt ein Scanline-Verfahren zum Einsatz.

Im Folgenden werden die einzelnen Schritte detailliert betrachtet.

4.5.3 Energieberechnung

Zuerst wird auf einem Bild einer Serie die Wavelet-Transformation mittels Faltung vorgenommen. Diese geschieht durch pixelweise Multiplikation des Bildes mit dem Wavelet im Frequenzraum. Das verwendete Wavelet ist das Mexican Hat Wavelet, da es rotationssymmetrisch und daher für zweidimensionale Signale geeignet ist. Als Ergebnis erhält man die Koeffizienten für ein Bild mit einheitlicher Skalierung und einer Translation von jeweils einer Pixelbreite in X- und Y-Richtung. Die Energie berechnet sich für einen beliebigen Bildausschnitt aus dem quadratischen Mittel der Koeffizienten:

$$E = \sqrt{c_1^2 + c_2^2 + \dots + c_n^2} \quad (4.55)$$

Da für die Berechnung der lokalen Differenzierbarkeit nur ein Ausschnitt des Bildes benötigt wird, wird die Energie für jedes Pixel für eine Umgebung von 9 mal 9 Pixeln berechnet.

4.5.4 Hölder-Exponent

Der Hölder Exponent α steht mit der Skalierung r des Wavelets als Basis in exponentiellen Zusammenhang mit der berechneten Energie für $r \rightarrow 0$:

$$r^\alpha \sim E_r \quad (4.56)$$

Der Hölder-Exponent ergibt sich somit aus der Steigung der Ausgleichsgeraden der Punkte $(\log r, \log E_r)$. Für die Berechnung wird die Energie für mehrere Skalierungen wie zuvor beschrieben ermittelt.

4.5.5 Scanline Verfahren

Nachdem die Hölder-Exponenten für jedes Pixel bestimmt sind, wird mit dem Verfahren nach Otsu ein Schwellwert gesucht, um den Düsenapparat von dem Rest des Bildes trennen zu können. Innerhalb der Düse befinden sich die Hölder-Exponenten oberhalb des Schwellwertes, und die Breite der Düse ist durch den Versuchsaufbau sowie die Kamerakalibrierung bekannt. Um die Position der Düse zu bestimmen, muss also ein Bereich gefunden werden, der ungefähr die zu erwartende Breite der Düse in Bildpunkten besitzt, eine rechteckige Ausdehnung hat und grösstenteils mit hohen Hölder-Exponenten gefüllt ist. Um diesen Bereich zu ermitteln, kann ein Scanline Verfahren benutzt werden. Global wird hierzu eine Liste angelegt, welche Datensätze der folgenden Form enthält:

p_l : linke Kante
 p_r : rechte Kante
 p_s : Anzahl der Pixel über dem Schwellwert.

In diesen Datensätzen werden Bereiche gespeichert, welche die Breite der Düse haben. Zu jedem Eintrag wird ein Zähler gehalten, der die Anzahl der Zeilen des Bereichs enthält. Daraufhin wird iterativ von oben nach unten durch die Zeilen gegangen. Für jede Zeile wird zuerst folgender Ablauf vorgenommen:

1. Suche von Links nach Rechts die erste Überschreitung des Schwellwerts.
2. Suche zur rechten Kante des Bereichs, Speichere die Koordinaten, beginne bei 1.

Mit diesem Verfahren werden alle einzelnen Bereiche innerhalb einer Zeile gefunden, welche vollständig über dem Schwellwert liegen. Sie werden in einer Liste identisch zu der globalen Liste abgelegt, wobei die Anzahl der Pixel aus der Differenz der Kanten ermittelt wird. Als zweiter Schritt werden die Bereiche nun so konkateniert, dass Bereiche der gesuchten Breite gebildet werden. Existieren diese, werden sie mit den bereits vorhandenen der globalen Liste verglichen. Existiert der Bereich schon in der vorherigen Zeile, wird er weiter übernommen und sein Zeilenzähler erhöht. Ist er nicht existent, wird ein neuer Eintrag in der globalen Liste eingetragen. Bereiche der vorherigen Zeile, die nicht weiterführen, werden gelöscht. Hierbei wird ebenfalls global der Bereich mit der grössten Zeilenzahl gespeichert. Nach Durchlaufen des Bilds entspricht der Mittelpunkt dieses grössten Bereichs der Mitte des Düsenapparats.

4.6 Haralick

Mit den sogenannten Haralick-Merkmalen war die Hoffnung verbunden, die Löcher der Lamelle segmentieren zu können. Diese Hoffnung wurde leider enttäuscht, da durch die resultierenden Bilder insgesamt nicht eindeutig auf Ort und Größe von Löchern geschlossen werden konnte. Es wurden hierfür folgende fünf der Haralick-Merkmalen verwendet:

- Energie
- Korrelation
- Trägheit
- lokale Homogenität
- Entropie.

Alle Merkmalsmaße basieren auf der sogenannten *Co-occurrence Matrix*. Da hier nur Grauwertbilder mit 256 Grauwertstufen betrachtet werden, hat diese quadratische Matrix eine Größe von 256x256. In jedem Matrixelement wird nun jeweils die Anzahl der in dem Bild vorkommenden

Paare von Grauwerten, bzgl. eines Abstands, gespeichert. Hier wurde lediglich mit dem Abstand von einem Pixel gearbeitet. Es wird also z.B. in dem Matrixelement (x,y) die Anzahl der im Bild nebeneinander vorkommenden Pixel eingetragen, wobei der eine Pixel den Grauwert x und der nebenliegende (Abstand ein Pixel) den Grauwert y hat. Nachfolgend die genaue Definition.

Definition 4.1. Co-occurrence Matrix

Die Co-occurrence Matrix C ist eine $G \times G$ Matrix wobei G die Anzahl der Grauwertstufen des Bildes der Größe $i \times j$ ist. Die Matrix M ist ebenso groß. c_1 und c_2 sind Abstandsparameter.

Dann ist C gegeben durch:

$$C = M + M^T$$

mit:

$$M(m,n) = \text{card}\{g(i,j) | g(i,j) = m \wedge g(i+c_1, j+c_2) = n\}, m, n \in \{0, \dots, G-1\}, c_1, c_2 \in \{0, 1\}$$

$g(i,j)$ gibt den Grauwert des Bildes an der Stelle (i,j) an.

$\text{card}(x)$ bezeichnet die Kardinalität der Menge x .

In der Implementierung werden vier Matrizen für je eine Richtung eines jeden Pixels betrachtet. Über diese vier Matrizen wird der Durchschnitt gebildet und diese Matrix danach noch einmal normalisiert.

Definition 4.2. Energie

Basierend auf der Co-occurrence Matrix C ist die Energie E folgendermaßen definiert:

$$E = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (C(i,j))^2$$

Definition 4.3. Korrelation

Basierend auf der Co-occurrence Matrix C ist die Korrelation K folgendermaßen definiert:

$$K = \frac{1}{\sigma_x \sigma_y} \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - \mu_x)(j - \mu_y) C(i,j)$$

mit:

$$\mu_x = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} i C(i,j)$$

$$\mu_y = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} j C(i,j)$$

$$\sigma_x = \sqrt{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - \mu_x)^2 C(i,j)}$$

$$\sigma_y = \sqrt{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (j - \mu_y)^2 C(i,j)}$$

Definition 4.4. Trägheit

Basierend auf der Co-occurrence Matrix C ist die Trägheit T folgendermaßen definiert:

$$T = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 C(i,j)$$

Definition 4.5. lokale Homogenität

Basierend auf der Co-occurrence Matrix C ist die lokale Homogenität L folgendermaßen definiert:

$$L = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1+(i-j)^2} C(i,j)$$

Definition 4.6. Entropie

Basierend auf der Co-occurrence Matrix C ist die Entropie H folgendermaßen definiert:

$$H = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} C(i,j) \ln C(i,j)$$

Die Bilder in Abb. 4.33 sind durch Kachelung des gesamten Bildes in kleinere Rechtecke entstanden. Die Rechtecke wurden jeweils als eigenständiges Bild behandelt und entsprechend die Haralick-Maße berechnet und in Relation zueinander gesetzt. Auf dieser Grundlage ist es nicht gelungen, Löcher der Lamelle eindeutig zu bestimmen. Die verschiedenartige Einfärbung der Haralick-Merkmalmaße lässt keine sicheren Rückschlüsse auf Löcher zu.

Die Bilder in Abb. 4.34 sind auf einem Ausschnitt des von der Kamera aufgenommenen Bildes entstanden. Auch die Hoffnung, eine Verbesserung durch das Eliminieren von nicht relevanten Werten des Teils des Bildes ohne Lamelle zu erzielen, schlug fehl. Auch hier sind Löcher nicht eindeutig und sicher zu identifizieren.

Die folgende Tabelle gibt die berechneten Werte für das Ausgangsbild aus Abbildung 4.33 an.

Energie	0,0243905
Korrelation	0,997192
Traegheit	8,69848
lokale Homogenität	0,577947
Entropie	-5,96445

4.7 Fazit

Die Lamellensegmentierung funktioniert nicht optimal, da der genaue Abrisspunkt nicht bestimmt werden kann. Dieses Problem kann vermutlich mit Hilfe von Snake-Verfahren behoben werden.

Die Qualität der Frequenzanalyse hängt sehr von den Segmentierungs-Ergebnissen ab. Eine Analyse von Testbildern ergab korrekte Ergebnisse in Bezug auf die Frequenzbestimmung.

Gute Ergebnisse nach der Tropfenanalyse sind unabhängig von der Segmentierung, jedoch stark abhängig von den Helligkeitsverhältnissen innerhalb des Tropfenbereichs unterhalb der Lamelle.

Die Löchersegmentierung wurde nicht weiter verfolgt und auch nicht in das Programm *3D-AugTracer* implementiert, da die Resultate wenig Erfolg aufweisen.

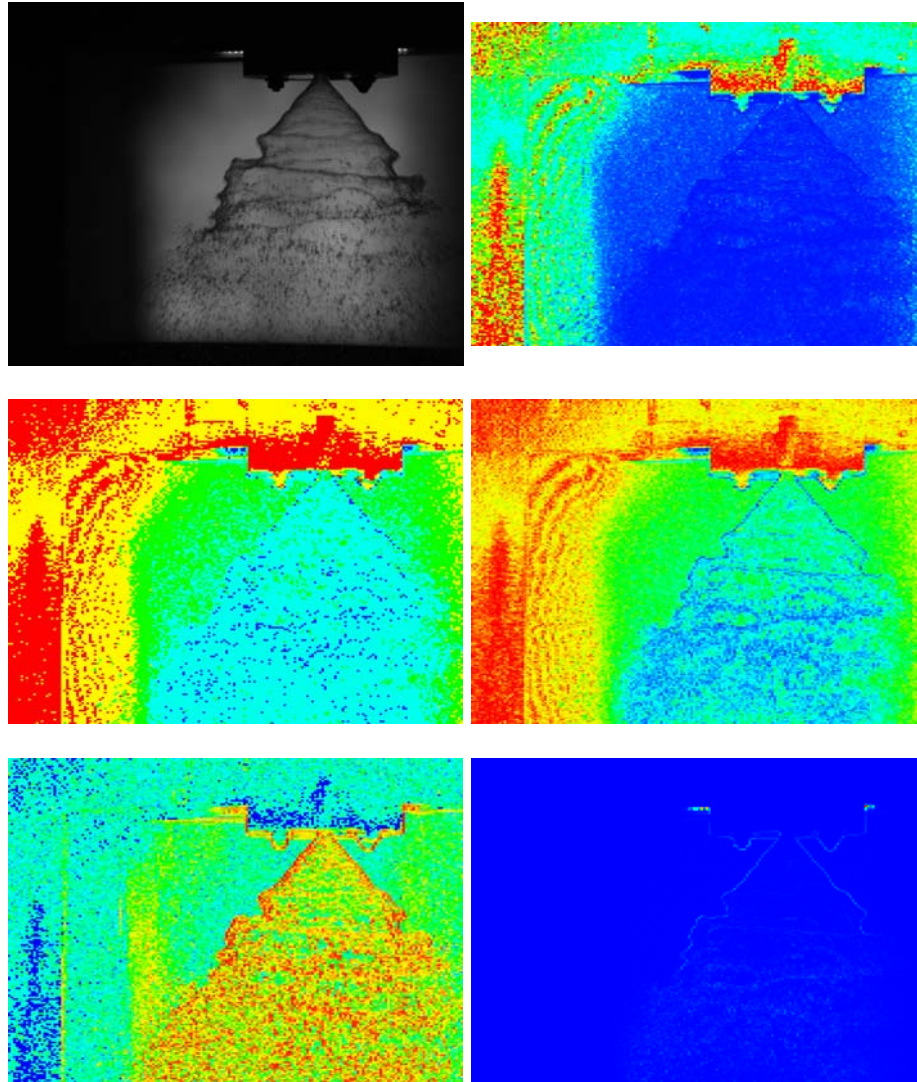


Abbildung 4.33: Ausgangsbild (o.l.), Beispielbilder für Energie (o.r.), Entropie (m.l.), lokale Homogenität (m.r.), Korrelation (u.l.) und Trägheit (u.r.) mit je 180x180 Kacheln

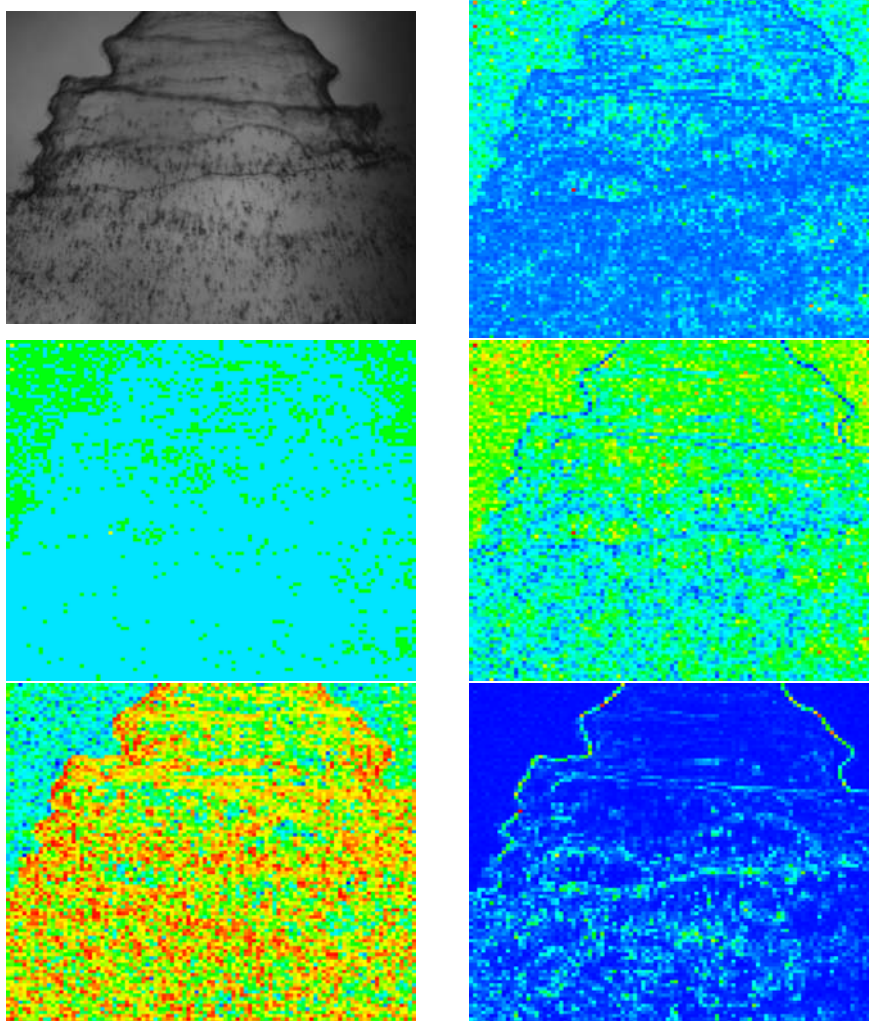


Abbildung 4.34: Lamellenausschnitt (o.l.), Ausschnitt für Energie (o.r.), Entropie (m.l.), lokale Homogenität (m.r.), Korrelation (u.l.) und Trägheit (u.r.) mit je 100x100 Kacheln

Kapitel 5

Tracking

Inhalt

5.1	Einleitung	69
5.2	KLT-Tracking	70
5.2.1	Triangulierung	71
5.3	Blockmatching	72
5.4	Partikel-Tracking	73
5.4.1	Initiale Ansätze	73
5.4.2	Berechnung des Trackings	74
5.4.3	Erstellen von synthetischen Partikeln	75
5.4.4	Berechnung der Wavelet-Transformation	75
5.4.5	Vorbereiten des nächsten Iterationsschritts	77
5.4.6	Skelettierung und Vektorfeldbestimmung	78
5.5	Fazit	79

5.1 Einleitung

Unter Tracking versteht man das Verfolgen von Objekten oder markanten Punkten über eine Bildsequenz. Es kann eingesetzt werden, um zum Beispiel Personen oder Fahrzeuge mit einer Kamera zu überwachen. Eine andere Möglichkeit ist es, aus der Beobachtung einer festen oder nur langsam sich ändernden Umgebung die Bewegung der Kamera zurückzurechnen.

Da im Programm **3D-AugTracer** die Positionen der Kameras nach der Kalibrierung (vgl. Kapitel 3.2.1) bekannt sind und sich über eine Bildserie nicht ändern, kann hier die Bewegung der Flüssigkeit, die aus der Düse strömt, überwacht werden. Aus den ermittelten Daten lassen sich Richtung und Geschwindigkeit sowohl der Lamelle als auch der Tropfen bestimmen. Dabei wurden zwei verschiedene Ansätze verfolgt:

Der erste Ansatz besteht darin, die Unterschiede der sichtbaren Strukturen zwischen zwei aufeinanderfolgenden Bildern zu bestimmen und als Bewegung zu interpretieren. Aufgrund der hohen Geschwindigkeit der Flüssigkeit kann dies mit den verwendeten Kameras nur im so genannten Doubleshuttermodus geschehen. Dazu werden Kameras so eingestellt, dass je zwei Bilder in einem sehr kurzen zeitlichen Abstand zueinander aufgenommen werden. Danach wird ein längerer Zeitraum benötigt, in dem der Bildspeicher ausgelesen wird und die Daten an den angeschlossenen Computer übertragen werden. In dieser Zeit können keine weitere Bilder aufgenommen werden. Anschließend werden wieder zwei Aufnahmen kurz hintereinander gemacht. Mit diesem Verfahren ist es möglich pro Sekunde etwa 20 Doubleshutter-Paare aufzunehmen. Aus dem bekannten Zeitabstand zwischen den Bildern lässt sich anschließend die Geschwindigkeit der Flüssigkeit berechnen.

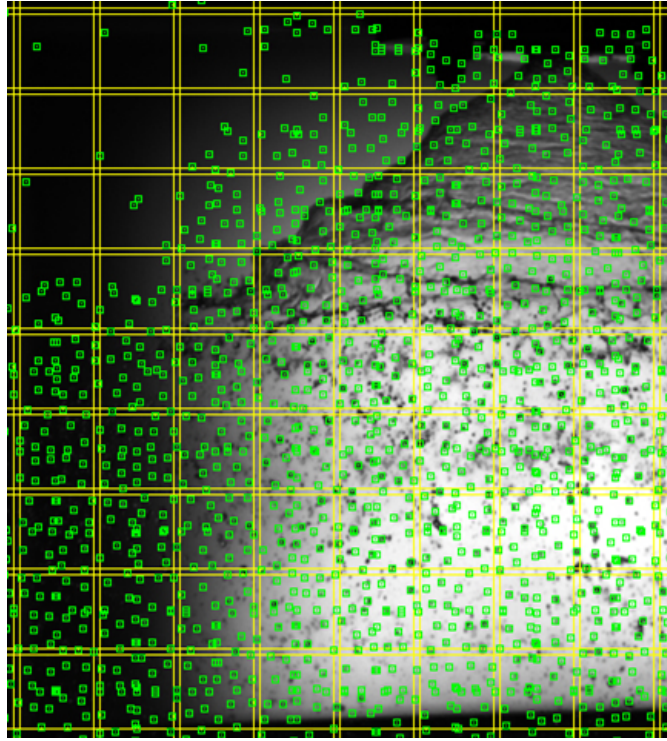


Abbildung 5.1: Selektierte Features mit Fensterung beim KLT-Tracking

Für den zweiten Ansatz werden fluoreszierende Tracerpartikel in die Flüssigkeit eingebracht, die sich mit der Lamelle mitbewegen. Hier ergeben sich aus der bekannten Form und Größe der Partikel weitere Möglichkeiten der Verfolgung. So können aufgrund der Verzerrung eines Tracerpartikels durch eine längere Belichtungszeit auch Bewegungen in einem einzelnen Frame erkannt werden. Ansatz eins wird vom Kanade-Lucas-Tomasi-Tracking sowie vom Blockmatching verwendet. Das Partikel-Tracking verwendet, wie der Name schon sagt, die zweite Möglichkeit. Alle drei Verfahren werden im Folgenden vorgestellt.

5.2 KLT-Tracking

Der KLT-Tracker (Kanade-Lucas-Tomasi) ist ein automatischer Feature-Tracking Algorithmus, der als Bibliothek in **3D-AugTracer** eingebunden wurde. KLT sucht selbständig nach den n am besten zu erkennenden Features im ersten Frame und versucht diese über die Bildsequenz zu verfolgen. Die gewünschte Anzahl n kann dabei vom Benutzer vorgegeben werden. Sollten während des Trackings Features verloren gehen, weil sie z. B. aus dem Bild herauslaufen oder sich zu stark verändern, so werden neue Features gesucht, bis eine Gesamtzahl von n wieder erreicht ist. Um eine möglichst gleichmäßige Verteilung der Features sicherzustellen, wird in **3D-AugTracer** jedes Bild in mehrere leicht überlappende Fenster unterteilt und der Algorithmus jeweils auf den sich entsprechenden Fenstern in den Frames der Bildserie ausgeführt (siehe Abb. 5.1).

Um gute Features zu bestimmen, berechnet KLT an jeder Bildposition die Eigenwerte der lokalen Intensitäts-Variations-Matrix

$$\mathbf{Z} = \begin{pmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{pmatrix}. \quad (5.1)$$

Dabei ist $g(x, y)$ die Intensitätsfunktion. Ein Feature wird als gut betrachtet, wenn beide Eigenwerte λ_1 und λ_2 über einem definierten Schwellenwert liegen. Die so erkannten Features werden nach dem

jeweils kleineren der Eigenwerte sortiert und die n besten für das Tracking ausgewählt.

5.2.1 Triangulierung

Um eine effiziente Speicherung und einen schnellen Zugriff auf die ermittelten Bewegungsvektoren zu ermöglichen, wurde sich für ein Raster entschieden, bei dem nur die Bewegungen an den Stützpunkten abgespeichert werden. Die Vektoren zwischen den Stützpunkten ergeben sich durch lineare Interpolation der umliegenden Werte. Da aber beim KLT-Tracking die verfolgten Features an unregelmäßigen, nicht vorherzusagenden Positionen liegen, müssen die Daten erst in dieses regelmäßige Muster überführt werden. Die Grundidee besteht dabei darin, die Featurepositionen zu triangulieren und für jeden Rasterpunkt das Dreieck zu finden, in das er fällt. Der abgespeicherte Bewegungsvektor ergibt sich dann aus Interpolation der Werte an den drei umliegenden Eckpunkten. Der Fehler, der sich durch die zweimalige Interpolation ergibt, wird dabei in Kauf genommen, in der Hoffnung, dass er durch Mitteln über die Vektorfelder vieler Bildpaare minimal wird.

Die Speicherung und Verwaltung der Triangulierung geschieht auf Grundlage der Half-Edge-Datenstruktur [Opea]. Dabei wird jede Kante durch zwei Halbkanten in entgegengesetzter Richtung repräsentiert. Jede Halbkante enthält einen Verweis auf den Knoten, auf den sie gerichtet ist und auf die Facette, die links von ihr liegt (in Richtung der Orientierung). Außerdem noch einen Verweis auf die entgegengerichtete Halbkante und zwei Verweise auf die Vorgänger- und Nachfolgerhalbante entlang der Umrundung der adjazenten Facette. Jede Facette speichert den Verweis auf eine angrenzende Halbkante und jeder Knoten eine Halbkante, die von diesem Knoten ausgeht. Mit diesen Informationen ist es möglich, sich entlang der Halbkanten durch die gesamte Struktur zu „hangeln“.

Bei der Art der Triangulierung wurde die Delaunay-Triangulierung gewählt, weil die dabei auftretenden Dreiecke die Eigenschaft haben, möglichst gleichmäßig verteilt zu sein. Insbesondere entstehen keine unnötig langen und spitzen Dreiecke, so dass die Interpolation immer nur auf einem kleinen, lokal beschränkten Bereich geschieht. Eine Triangulierung erfüllt genau dann die Delaunay-Bedingung, wenn alle Dreiecke einen leeren Umkreis haben, d.h. dass keine weiteren Punkte innerhalb des Umkreises liegen. Solch eine Triangulierung ist immer möglich und eindeutig, wenn nicht mehr als 3 Punkte auf einem Kreis liegen.

Die Datenstruktur für die Triangulierung ist so angelegt worden, dass beim Einfügen neuer Knoten das schon vorhandene Netz so umgeformt wird, dass wieder eine gültige Delaunay-Triangulierung vorliegt (vgl. [Pri]). Dazu wird die Ebene zunächst mit den Eckpunkten eines Rechtecks, welches den Bereich des gesamten Bildes knapp umfasst, initialisiert und die Eckpunkte auf triviale Weise trianguliert. Knoten, die außerhalb dieses Rechtecks eingefügt werden sollen, werden ignoriert. Dadurch kann man sich auf den Fall beschränken, dass ein neuer Knoten innerhalb einer schon vorhandenen Facette liegt. Das Einfügen eines Knotens läuft in den folgenden Schritten ab (vgl. Abb. 5.2):

- a: Zuerst muss die Facette gesucht werden, die die Koordinaten des neuen Knotens enthält. Falls der Knoten auf einer Kante liegt, kann eine der beiden angrenzenden Facetten gewählt werden. Wenn ein Knoten mit den selben Koordinaten schon vorhanden ist, wird das Einfügen sofort abgebrochen.
- b: Von der gefundenen Facette aus werden rekursiv nach außen alle Dreiecke gesucht, in deren Umkreis der neue Punkt fällt. Hat ein Dreieck einen leeren Umkreis, so haben auch die nach außen angrenzenden Dreiecke leere Umkreise und die Rekursion kann abgebrochen werden. Außerdem bricht die Rekursion ab, wenn man beim Überprüfen über die zuvor festgelegte Größe der zu triangulierenden Fläche hinausläuft.
- c: Als nächstes werden alle Facetten die im vorherigen Schritt gefunden wurden gelöscht. Zusätzlich werden alle Kanten entfernt, die durch das Löschen der beiden angrenzenden Facetten überflüssig geworden sind.

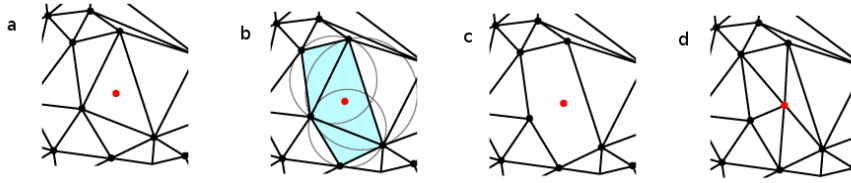


Abbildung 5.2: Ablauf der Triangulierung beim Einfügen eines neuen Punktes

- d: Zum Schluss wird der neue Knoten eingefügt und Kanten zwischen allen Knoten auf dem Rand des entstandenen „Lochs“ und dem neuen Knoten erstellt. Zwischen den Kanten werden neue Facetten eingefügt.

Um zu zeigen, dass die entstandene Triangulierung wirklich der Delaunay-Bedingung genügt, reicht es zu zeigen, dass alle neuen Dreiecke einen leeren Umkreis haben. Dies ist erfüllt, wenn nach dem Hinzufügen eines Knotens v jede der neuen Kanten eine Delaunay-Kante ist, d.h. einen leeren Umkreis besitzt. Alle alten Kanten sind Delaunay-Kanten, da mindestens ein angrenzendes Dreieck die Umkreisbedingung erfüllt, andernfalls wäre sie von dem Algorithmus entfernt worden.

Angenommen, es gäbe eine solche Kante k , die keine Delaunay-Kante ist. Für eine gültige Delaunay-Triangulierung müsste stattdessen eine Kante k' eingefügt werden, die nicht v als einen der Endpunkte hat. Eine solche Kante hätte aber auch schon vor Einfügen von v einen leeren Umkreis, also wäre sie wegen der Eindeutigkeit der Delaunay-Triangulierung schon vorher vorhanden gewesen¹. Dann hätte aber eines der Dreiecke an k' auch nach Einfügen von v einen leeren Umkreis und k' wäre nicht von dem Algorithmus entfernt worden. Das ist ein Widerspruch zur Folgerung, dass k' nicht in der entstandenen Triangulierung liegt. Daher ist die Annahme falsch und es liegt tatsächlich eine Delaunay-Triangulierung vor.

5.3 Blockmatching

Bei diesem Verfahren werden aus zwei aufeinanderfolgenden Bildern jeweils ein kleines quadratisches Fenster um die selbe Position herum ausgeschnitten und miteinander verglichen. Das Fenster im ersten Bild bildet das Template, das etwas größere Suchfenster wird aus dem zweiten Bild ausgeschnitten. Es wird davon ausgegangen, dass ein Feature im Bild zwar seine Position, nicht aber sein grundsätzliches Aussehen ändert. Daher kann das Template t über das größere Suchfenster f verschoben werden und die Übereinstimmung zwischen beiden Fenstern mittels der Korrelation

$$c(u, v) = \sum_{i,j} f(i, j)t(i - u, j - v) \quad (5.2)$$

berechnet werden. Da diese einfache Formel bei starken Helligkeitsschwankungen im Bild schlechte Ergebnisse liefern kann, wird die Korrelation durch eine Normierung verbessert:

$$\gamma(u, v) = \frac{\sum_{i,j} [f(i, j) - \bar{f}_{u,v}] [t(i - u, j - v) - \bar{t}]}{\left\{ \sum_{i,j} [f(i, j) - \bar{f}_{u,v}]^2 [t(i - u, j - v) - \bar{t}]^2 \right\}^{0,5}} \quad (5.3)$$

Dabei ist \bar{t} der Mittelwert des Templates und $\bar{f}_{u,v}$ der Mittelwert des Suchfensters unter dem Template. Durch die Normierung liegen alle Werte in dem Intervall $[-1, 1]$ (vgl. [Lew]).

¹In dem Fall, dass mehr als drei Punkte auf einem Kreis liegen, kann es mehrere solche Kanten geben, die zu einer gültigen Delaunay-Triangulierung gehören. Mindestens eine davon war vor Einfügen von v vorhanden. Nenne diese k'

Die Werte u und v , für die die höchste Korrelation bestimmt wurde, werden als Verschiebung in x - und y -Richtung des Punktes in der Mitte Templates angenommen. Sollte sich ein Feature so schnell bewegen, dass es im zweiten Bild nicht mehr innerhalb des Suchfensters liegt oder sich so verändern, dass es nicht mehr wiederzuerkennen ist, so wird trotzdem ein Punkt mit maximaler Korrelation bestimmt, der aber keine Bedeutung mehr hat. Daher werden alle Ergebnisse, die unterhalb eines anpassbaren Schwellenwertes liegen, verworfen. Dieser Schwellenwert ist standartmäßig auf 0,5 voreingestellt.

Um die Bewegungen im gesamten Bild zu erfassen, wird diese Berechnung um mehrere Punkte herum bestimmt. Dazu wird ein quadratisches Gitter mit einstellbarer Seitenlänge über das Bild gelegt und an jedem Schnittpunkt das Suchfenster bzw. das Template in den beiden Frames ausgeschnitten. Da somit die Position der verfolgten Features beim Blockmatching bekannt ist, entfällt der rechenaufwändige Schritt der Triangulierung und der Interpolation der Messwerte (vgl. Kapitel 5.2.1). Die ermittelten Verschiebevektoren können unmittelbar abgespeichert werden.

Die Fenstergrößen des Suchfensters und des Templates können angepasst werden um bessere Ergebnisse zu erhalten oder die Suchzeit einzuschränken, wenn eine maximale Bewegung zwischen den Frames bekannt ist.

5.4 Partikel-Tracking

Ursprünglich war in der Projektgruppe angedacht, die Oberflächenstrukturen eines resultierenden Sprühkegels anhand von fluoreszierenden Partikeln zu identifizieren. Die eingesetzten Partikel werden der Testflüssigkeit zugeführt und sind auf den Videoaufnahmen als nachleuchtende Punkte oder Streifen (bedingt durch die Belichtungsdauer) erkennbar. Ein Beispiel für eine solche Partikelspur ist in Abb. 5.3 sichtbar².

Das Problem des Partikeltrackings besteht hauptsächlich in der Segmentierung der Spuren. Da sich diese nur schwach von den Strukturen des Sprühkegels unterscheiden (bzw. nicht immer als solche erkennbar sind), hat sich das Tracking-Verfahren als deutlich komplexer herausgestellt. Einige erfolglose Ansätze werden im folgenden Kapitel beschrieben. Im Anschluß erfolgt die Darstellung der effektiven Tracking-Berechnung.

5.4.1 Initiale Ansätze

Das Partikel-Tracking unterscheidet sich von den anderen Tracking-Verfahren, da es im Vergleich zu den Doubleshutter-Aufnahmen lediglich auf Einzelbildern einer Kamera arbeitet (vgl. Kapitel 1.2). Die sichtbaren Partikel-Spuren geben bereits Richtungen und Geschwindigkeiten der Partikel-Bewegung an. Die somit resultierenden Größen können über die Berechnungen aus einer gesamten Bilderserie verfeinert werden.

Zunächst wurden offensichtliche Ideen verfolgt, um die sichtbaren Partikel-Spuren zu segmentieren.

- Die direkte Bestimmung der Partikelspuren durch Kantendetektion (z.B. Sobel-Operator) hat zu einer Übermenge an Informationen geführt, die letztlich keinen Gewinn eingebracht hat.
- Eine Verfeinerung der Ergebnisse konnte durch Korrelation mit fixen Kanten-Templates erreicht werden. Dazu wurden exemplarische Ausschnitte des Bildes gewählt und mit dem gesamten Video-Bild gefaltet. Die ausgeschnittene Partikelbahnen in der Größe von ca 20x20 Pixeln haben jedoch zu keinen hochwertigen Ergebnissen geführt. Dies lag hauptsächlich an der Verwendung von einseitig gedrehten Partikel-Templates, so dass sich Korrelationen (wenn überhaupt) nur auf der Seite der Lamelle ergaben, aus der das Template ausgeschnitten wurde.

²auf den Aufnahmen ist eine UV-Lampe zur Aktivierung der Fluoreszenz erkennbar, die z.T. in das sichtbare und von der Kamera aufgenommene Lichtspektrum einstrahlt

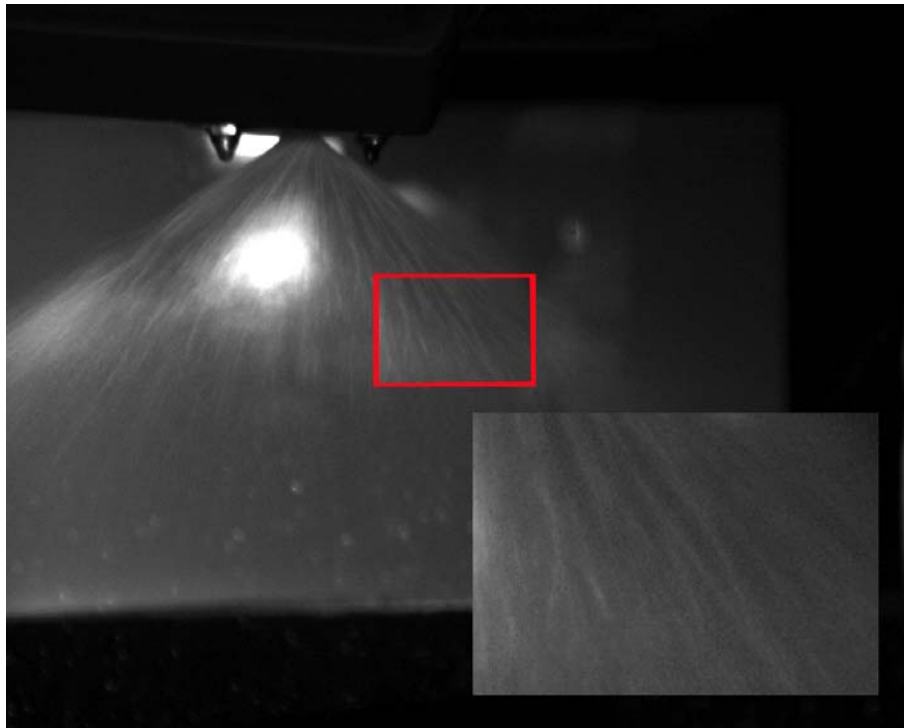


Abbildung 5.3: Gesamtansicht mit Partikelausschnitt

- Ein weiterer Ansatz bestand in der Merkmals-Identifizierung durch Wavelets. Obwohl das Ergebnis einer einfachen Wavelet-Transformation keine eindeutige Identifizierung der Bahnen ermöglicht hat, wurden Teile dieser Berechnungen für die Lösung übernommen.

Die Erkenntnisse dieser Ansätze wurden dann für die finale Methodik vereinzelt übernommen.

5.4.2 Berechnung des Trackings

Die bestmögliche Tracking-Berechnung folgte aus dem Aufsatz "Rotation-invariant pattern matching using wavelet decomposition" von Tsai und Chen [TC02]. Die Autoren beschreiben dort ein Verfahren zur Texturklassifikation anhand bestimmter rotationsunabhängiger Merkmale. Die Bestimmung der bildweisen Partikelpuren orientiert sich an folgendem Grobverlauf:

1. Zunächst wird ein synthetisches Partikelbild generiert.
2. Die gefilterten Bilder werden nach Pyramiden-Schema und unter Einsatz einer Multiskalen-Zerlegung auf eine wählbare Anzahl J von Stufen herunterskaliert.
3. Auf der untersten Stufe werden mittels Ringprojektion (vgl. Kapitel 5.4.4.1) und Korrelation die markanten Punkte für die Partikelpuren bestimmt.
4. Die entstehende Karte wird binarisiert, Median-gefiltert und auf die nächste Stufe hochskaliert.
5. Aus dieser Maske werden nur noch die relevanten Bereiche für die nächsthöhere Stufe entnommen.
6. Schritt 3 und 4 werden jeweils für die nächsthöhere Skalierungsstufe des Bildes wiederholt.
7. Die überzeichneten Pfadstrukturen aus der höchsten Stufe werden zuletzt durch eine mediale Hauptachsen-Transformation verdünnt und in einzelne Pfade umgewandelt.

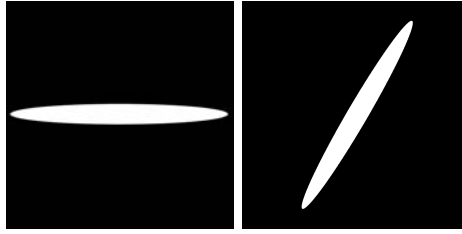


Abbildung 5.4: synthetische Partikel: normal und rotiert

i	Analyse-Filter		Synthese-Filter	
	TP-filter $h_L(i)$	HP-Filter $h_H(i)$	TP-Filter $g_L(i)$	HP-Filter $g_H(i)$
0	0.60294	1.11508	1.11508	0.60294
+/- 1	0.26686	-0.59127	0.59127	-0.26686
+/- 2	-0.07822	-0.05754	-0.05754	-0.07822
+/- 3	-0.01686	0.09127	-0.09127	0.01686
+/- 4	0.02674			0.02674

Tabelle 5.1: Daubechies-9/7-Filter

5.4.3 Erstellen von synthetischen Partikeln

Aus der impliziten Darstellung für allgemeine Ellipsen mit Breite a und Höhe b wird ein quadratisches Partikelbild der Größe $M_p \times M_p$ erzeugt, so dass ein synthetisches Partikel einer gefüllten Ellipse mit starker Längsstreckung entspricht (vgl. Abb. 5.4). Diese Form der Partikel konnte durch Betrachten der Testbilder angenähert werden.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (5.4)$$

5.4.4 Berechnung der Wavelet-Transformation

Zunächst wird das Eingabebild $f(x, y)$ jeweils mit einem Hochpass $h_H(i)$ bzw. Tiefpass $h_L(i)$ in x -Richtung gefaltet.

$$\tilde{f}_H(x, y) = f(x, y) * h_H(x) := \sum_{\xi=-3}^3 f(\xi, y) h_H(x - \xi) \quad (5.5)$$

$$\tilde{f}_L(x, y) = f(x, y) * h_L(x) := \sum_{\xi=-4}^4 f(\xi, y) h_L(x - \xi) \quad (5.6)$$

Die verwendeten Daubechies-9/7-Filter $h_H(i)$ und $h_L(i)$ für die Analyse bzw. Synthese sind in Tabelle 5.4.4 ersichtlich und sind in [ANSE00] nachlesbar.

Im Anschluss werden die Bilder auf halbe Breite skaliert, also gilt:

$$f_H(x', y) := \tilde{f}_H(2x', y), 0 \leq x' \leq \left\lfloor \frac{x}{2} \right\rfloor \quad (5.7)$$

$$f_L(x', y) := \tilde{f}_L(2x', y), 0 \leq x' \leq \left\lfloor \frac{x}{2} \right\rfloor \quad (5.8)$$

Somit entstehen aus einem Ausgangsbild mit Dimension $M \times N$ je ein hoch- bzw. tiefpassgefiltertes Ergebnisbild mit Dimension $\frac{M}{2} \times N$. Auf den entstehenden Bildern $f_H(x, y)$ bzw. $f_L(x, y)$ wird die

Faltung und Skalierung erneut in y -Richtung durchgeführt.

$$\tilde{f}_{HH}(x, y) = f_H(x, y) ** h_H(y) := \sum_{\zeta=-3}^3 f_H(x, \zeta) h_H(x, y - \zeta) \quad (5.9)$$

$$\tilde{f}_{HL}(x, y) = f_H(x, y) ** h_L(y) := \sum_{\zeta=-4}^4 f_H(x, \zeta) h_L(x, y - \zeta) \quad (5.10)$$

$$\tilde{f}_{LH}(x, y) = f_L(x, y) ** h_H(y) := \sum_{\zeta=-3}^3 f_L(x, \zeta) h_H(x, y - \zeta) \quad (5.11)$$

$$\tilde{f}_{LL}(x, y) = f_L(x, y) ** h_L(y) := \sum_{\zeta=-4}^4 f_L(x, \zeta) h_L(x, y - \zeta) \quad (5.12)$$

und

$$f_{HH}(x, y') := \tilde{f}_{HH}(x, 2y'), 0 \leq y' \leq \left\lfloor \frac{y}{2} \right\rfloor \quad (5.13)$$

$$f_{HL}(x, y') := \tilde{f}_{HL}(x, 2y'), 0 \leq y' \leq \left\lfloor \frac{y}{2} \right\rfloor \quad (5.14)$$

$$f_{LH}(x, y') := \tilde{f}_{LH}(x, 2y'), 0 \leq y' \leq \left\lfloor \frac{y}{2} \right\rfloor \quad (5.15)$$

$$f_{LL}(x, y') := \tilde{f}_{LL}(x, 2y'), 0 \leq y' \leq \left\lfloor \frac{y}{2} \right\rfloor \quad (5.16)$$

Das Teilbild HH (high/high) beinhaltet nun höchsten Koeffizienten-Details des Bildes, da es die Informationen der doppelten Hochpassfilterung enthält. In den beiden Teilbildern HL (high/low) und LH (low/high) sind die mittelbandigen Informationen enthalten. Das grösste Teilbild LL (low/low) entspricht dem stark tiefpass-gefilterten Originalbild. Sobald die Pyramiden-Skalierung J -mal durchgeführt wird, entstehen somit 4^J Einzelbilder der Grösse $\frac{M}{2^J} \times \frac{N}{2^J}$.

5.4.4.1 Verfahren der Ringprojektion

Da die Tracerpartikel unterschiedliche Lage besitzen, muss auf ein Identifikations-Merkmal zurückgegriffen werden, dass die Ähnlichkeit eines Partikels zu den synthetischen Ellipsen-Partikeln rotationsunabhängig bestimmen kann. Die rechenzeit-intensive Bestimmung der Ringprojektion ermöglicht dies. Im Wesentlichen werden die folgenden Berechnungen zweifach hintereinander durchgeführt. Der erste Durchgang ermittelt auf dem Teilbild f_{LL}^J (tiefpassgefiltertes Eingabebild auf Stufe J) die Koeffizienten wie folgt beschrieben.

Für jeden Punkt $f(x, y)$ wird nun der Vektor der Ringprojektion $p(r)$ bestimmt. Ein gegebener kartesischer Punkt $f(x, y)$ wird zunächst in Polarkoordinaten umgewandelt:

$$f(x, y) = f(r \cos \theta, r \sin \theta) \quad (5.17)$$

Die Funktion der Ringprojektion $p(r)$ ergibt sich aus dem Integral:

$$p(r) = \frac{1}{2\pi r} \int_0^{2\pi} f(r \cos \theta, r \sin \theta) d\theta \quad (5.18)$$

Das Ergebnis besteht aus einem Vektor der Länge r_{max} . Da die Grösse der bestmöglichen Ringprojektion sinnvollerweise durch die Grösse des synthetischen Partikels $M_p \times M_p$ begrenzt wird, ergibt sich ein $r_{max} = \frac{M_p}{2}$.

Wie erwähnt beansprucht die lokale Berechnung der Ringprojektion den anteilmässig grössten Rechenaufwand des Algorithmus. In **3D-AugTracer** wird das Integral $\int_0^{2\pi} f(\dots) d\theta$ lediglich mit einer begrenzten Anzahl diskreter Stellen ausgewertet. Da sich aus der Anwendung heraus nur geringe

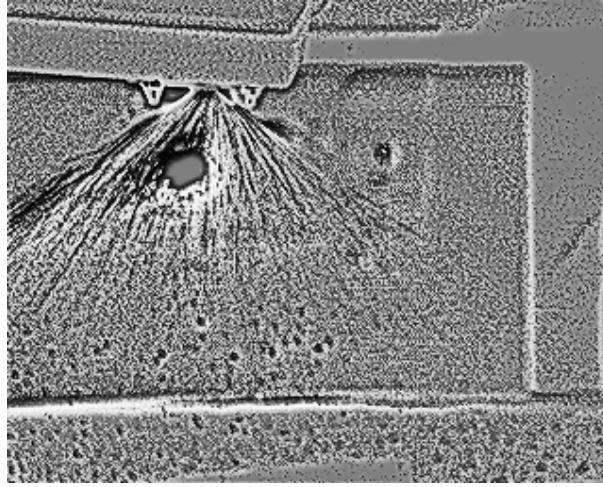


Abbildung 5.5: Korrelationskoeffizienten

Partikelgrößen ergeben, ergibt sich für r_{max} i.d.R. ein Wert zwischen 10 und 20. Die Anzahl der addierten Integral-Summanden kann durch eine grobe Schätzung der überdeckten Pixel verkürzt werden (ausgehend vom Punkt $f(x, y)$ liegen maximal $8r$ Pixel auf dem Umkreis mit Radius r).

5.4.4.2 Korrelation auf der Ringprojektion

Es sind nun der vorberechnete Ringprojektions-Vektor $P_s = (p(0), p(1), \dots, p(r_{max}))$ für das synthetische Partikel und der lokale Ringprojektionsvektor $P_l = (p'(0), p'(1), \dots, p'(r_{max}))$ gegeben, zusätzlich noch μ_s bzw. μ_l als Mittelwerte von P_s bzw. P_l . Dann definiert sich die normierte Korrelation $\bar{\rho}$ zwischen den Vektoren als

$$\bar{\rho} = \frac{\sum_{r=0}^{r_{max}} (p(r) - \mu_s)(p'(r) - \mu_l)}{\sqrt{\sum_{r=0}^{r_{max}} (p(r) - \mu_s)^2 \cdot \sum_{r=0}^{r_{max}} (p'(r) - \mu_l)^2}}, \bar{\rho} \in [-1; 1] \quad (5.19)$$

Ein Korrelations-Koeffizient von 1 beweist dabei einen vollständig linearen Zusammenhang, -1 beweist einen vollständig reziproken Zusammenhang. Ein Koeffizient mit einem Betrag zwischen 0 und 1 gibt den Grad der Übereinstimmung an. Falls der Wert 0 beträgt, dann liegt kein Zusammenhang vor. Das Resultat einer solchen Korrelationbewertung gibt Abb. 5.5 wieder.

5.4.5 Vorbereiten des nächsten Iterationsschritts

Das Ziel dieses Schrittes besteht darin, aus den Korrelationsdaten eine binäre Maske für die nächstgrößere Auflösungsstufe $J-1$ zu bestimmen. Dazu wird ein Schwellenwert $\bar{\rho}_{min}$ für die Korrelation definiert, so dass die korrespondierenden 2×2 Pixel der nächstgrößeren Maske M_J genau dann aktiviert werden, wenn $\bar{\rho} \geq \bar{\rho}_{min}$. Die Berechnung der Ringprojektionen auf dieser Stufe erfolgt dann nur noch für die markierten Pixel.

Ein Ausschnitt der Berechnung ist in Abb. 5.6 dargestellt. Es zeigt sich jedoch, dass die binären Daten keine zusammenhängenden Bereiche abdecken. Zur Ausbesserung wird ein Median-Filter wählbarer Größe $m \times m$ benutzt.

Zu diesem Zeitpunkt werden die drei verbleibenden Teilbilder der Stufe J berücksichtigt. Aus den einzelnen Teilbildern $f_{HH}^J(x, y)$, $f_{HL}^J(x, y)$ und $f_{LH}^J(x, y)$ wird mittels Wavelet-Synthese ein kumuliertes Teilbild $f_d^J(x, y)$ mit Größe $\frac{M}{2^J} \times \frac{N}{2^J}$ konstruiert.

$$f_d^J(x, y) = |f_{LH}^J(x, y)| + |f_{HL}^J(x, y)| + |f_{HH}^J(x, y)| \quad (5.20)$$



Abbildung 5.6: Korrelationskoeffizienten: Grauwerte, binarisiert und Median-gefiltert (3x3)

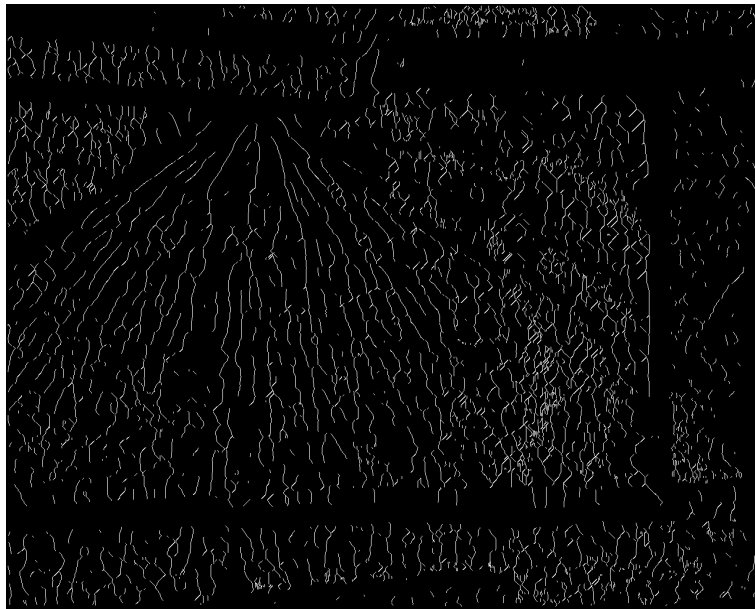


Abbildung 5.7: Partikelpfade

Eine weitere Korrelationsberechnung erfolgt nun auf diesem Teilbild. Es wird die Binärmaske aus $f_{LL}^J(x, y)$ benutzt, um die markanten Pixel einzuzugnen.

5.4.6 Skelettierung und Vektorfeldbestimmung

Nach der mehrfachen Anwendung der vorherigen Schritte definiert die Binärmaske M_0 nun die überzeichneten Tracerbahnen in der höchsten Detailstufe. Aus dem binären Maskenbild werden nun die Pfade extrahiert. Da sich bereits die groben Tracerspuren abzeichnen, werden gängige Verfahren zur Berechnung der Medialachsentransformation benutzt [Mül05]. Als Ergebnis ergeben sich Pfadbilder mit ausgedünnten Spuren der Tracer-Partikel (vgl. Abb. 5.7). Der letzte Schritt besteht in der Bestimmung der Vektorpfade aus dem Ergebnisbild. Ausgehend von jedem Pixel $p(x, y)$ wird in folgender Reihenfolge untersucht, ob es ebenfalls ein Pfadpixel unterhalb ($p(x, y+1)$), links unterhalb ($p(x-1, y+1)$) oder rechts unterhalb ($p(x+1, y+1)$) gibt. Sofern dies zutrifft, wird das Vektorfeld an der Stelle $v(x, y)$ auf den entsprechenden normierten Vektor zu diesem Pixel

gesetzt, also:

$$v(x, y) = \begin{cases} \left(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^T & \text{sofern } p(x-1, y+1) \text{ existiert} \\ (0, 1)^T & \text{sofern } p(x, y+1) \text{ existiert} \\ \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^T & \text{sofern } p(x+1, y+1) \text{ existiert} \end{cases} \quad (5.21)$$

5.5 Fazit

Die Verfahren Blockmatching und KLT-Tracking liefern in allen getesteten Bildserien vergleichbare Ergebnisse, die auch im Rahmen der Erwartungen bzgl. der Dimensionen der berechneten Kenngrößen (z.B. Geschwindigkeiten) liegen.

Die Ausarbeitung des Partikel-Trackings konnte nur auf einer einzelnen Testreihe durchgeführt werden. Es hat sich gezeigt, dass unter diesen Bedingungen durchaus akzeptable Ergebnisse erzielt werden können. Allerdings wurde im Laufe der Projektgruppe der Ansatz der Tracerpartikel-Detektion nicht weiter verfolgt, so dass keine weiteren Testbilder zur Verfügung standen. Das Verfahren hat somit nur experimentellen Status und seine Funktionalität in **3D-AugTracer** ist keineswegs sichergestellt.

Kapitel 6

3D-Rekonstruktion

Inhalt

6.1	Einleitung	81
6.2	Berechnung der Fundamentalmatrix	81
6.3	Das Korrespondenzproblem	83
6.4	Berechnung der 3D Punkte	84
6.5	Anmerkungen	85
6.6	Metrik	85
6.7	Fazit	86

6.1 Einleitung

Um die Ergebnisse der Lamellenextraktion (vgl. Kapitel 4.2) sowie der Tropfenanalyse (vgl. Kapitel 4.4) im 3-dimensionalen Raum zu visualisieren, müssen aus den gewonnenen 2-dimensionalen Informationen 3-dimensionale Informationen errechnet werden. Diese werden über die Punktepaare aus beiden Bildern berechnet. Um diese zu erhalten benötigt man zunächst die Ergebnisse der Kamerakalibrierung, da die intrinsischen und extrinsischen (vgl. [Zha98]) Parameter für die Berechnung einer Fundamentalmatrix verwendet werden.

Über diese Matrix kann man für jeden Punkt in einem Kamerabild eine Epipolarlinie zu dem entsprechenden Punkt in dem anderen Bild berechnen. Dies dient der Einschränkung des Suchbereiches, da nun nur entlang der Linie nach dem korrespondierenden Punkt gesucht werden muss.

Nachdem dieses Korrespondenzproblem gelöst wurde, kann man den 3D Punkt errechnen, indem man eine Ebene durch die Punkte legt und den Schnittpunkt der optischen Linien berechnet.

Für die korrekte Bestimmung von Längenangaben wird ein Metrik-Faktor benötigt, der die ermittelten Pixel-Abstände in Millimeter umwandelt.

6.2 Berechnung der Fundamentalmatrix

Die Epipolare Geometrie besagt, dass, wenn z.B. zwei Bilder aus verschiedenen Kamerapositionen C_1 und C_2 existieren und man die Projektion m_1 (das ist der Schnittpunkt der Linie $C_1\bar{M}$ mit der Bildebene) des Bildes vom Objektpunkt \mathbf{M} kennt, der korrespondierende Punkt im anderen Bild auf der sogenannten Epipolar-Linie liegt. Diese Linie wird durch die Spur gebildet, die entsteht, wenn man die Ebene, die von den Projektionszentren der Kameras C_1 und C_2 und den Objektpunkt \mathbf{M} gebildet werden, mit den Bildebenen schneidet (siehe Abb. 6.1).

Für die Berechnung von korrespondierenden Punkten deren Rücktransformation in den euklidischen Raum muss aus den 2D-Aufnahmen der epipolare Zusammenhang bestimmt werden.

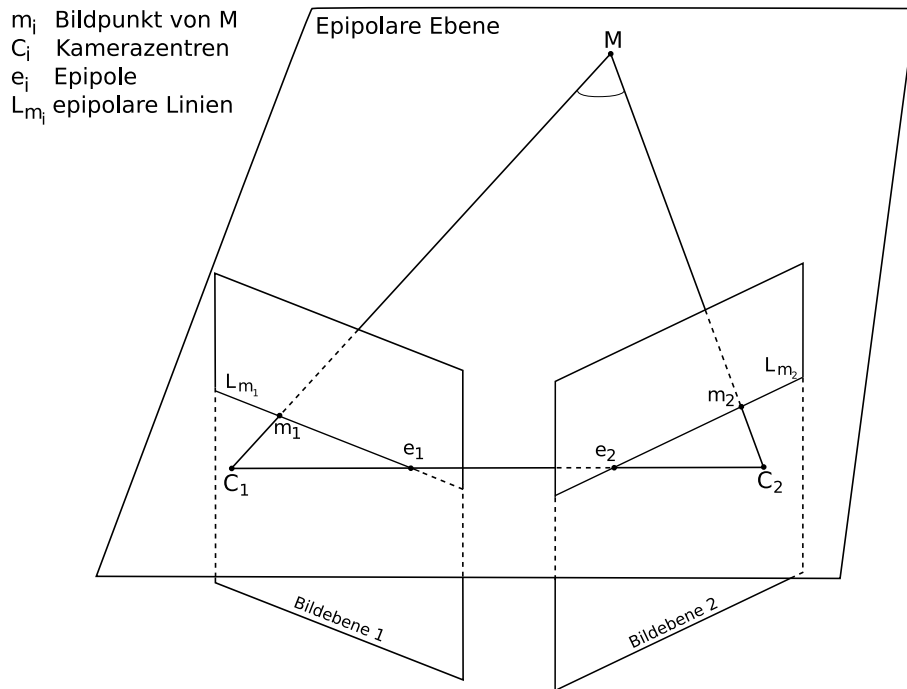


Abbildung 6.1: Epipolare Geometrie

Die Fundamentalmatrix beschreibt den Zusammenhang zweier Bilder einer Szene. Bei gegebenem Punkt \mathbf{m}_1 in der ersten Bildebene B_1 muss der korrespondierende Punkt in der zweiten Bildebene B_2 auf der sogenannten epipolaren Linie \mathbf{l}_{m_2} von \mathbf{m}_1 liegen. Die Linie \mathbf{l}_{m_2} von \mathbf{m}_1 ergibt sich als Schnitt der epipolaren Ebene E mit der zweiten Bildebene B_2 . Diese Gerade wird die epipolare Gerade genannt.

Allgemeiner gesagt: für einen Bildpunkt \mathbf{m}_i der ersten Kamera liegen alle 3D-Punkte, die auf diesen Bildpunkt abgebildet werden, auf einer Geraden durch diesen Bildpunkt und der ersten Kamera. Der korrespondierende Punkt im Bild der zweiten Kamera muss auf dem Bild der zweiten Geraden bzgl. der zweiten Kamera liegen. Alle epipolaren Geraden schneiden sich in einem Punkt, dem Epipol (hier: $\mathbf{e}_1, \mathbf{e}_2$), der Bildpunkt des optischen Zentrums der zweiten Kamera. Die Fundamentalmatrix beschreibt diesen Zusammenhang und bestimmt die epipolaren Geraden.

Seien \mathbf{A}_1 und \mathbf{A}_2 die intrinsischen Parameter der ersten, bzw. zweiten Kamera. Dann ist die Fundamentalmatrix \mathbf{F} :

$$\mathbf{F} = \mathbf{A}_2^{-T} \mathbf{T} \mathbf{R} \mathbf{A}_1^{-1} \quad (6.1)$$

\mathbf{T} ist dabei nichts anderes als eine antisymmetrische Matrix, definiert durch \mathbf{t} , mit der Bedingung $\mathbf{T}z = \mathbf{t} \times z$ für alle 3D-Vektoren:

$$\mathbf{T} = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix}$$

Es wurden zwei verschiedene Verfahren implementiert, um die Fundamentalmatrix zu berechnen. Zunächst wurde die einfache Methode von Zhang (vgl. [ZL01]) genutzt, bei der über die Projektionsmatrizen der beiden Kameras die Matrix berechnet wird. Ein gängigeres Verfahren bietet die lineare Methode ebenfalls nach Zhang (8-point-algorithm [Zha98]). Hiermit lassen sich die Fundamentalmatrix und die beiden Epipole (Eigenvektor zum Eigenwert Null der Fundamentalmatrix bzw. deren Transponierten) aus mindestens acht korrespondierenden Punktpaaren schätzen. Diese Punktpaare erhält man über die Referenzpunkte des Kalibrierungsgitters.

Seien $\tilde{\mathbf{m}}_{1k} = [u_{1k}, v_{1k}, 1]^T$ und $\tilde{\mathbf{m}}_{2k} = [u_{2k}, v_{2k}, 1]^T$ die korrespondierenden Punktepaare in den beiden Bildern und $\mathbf{F} = (\mathbf{F}_{ij})_{1 \leq i, j \leq 3}$ die gesuchte Fundamentalmatrix, so gilt :

$$\tilde{\mathbf{m}}_k^T \mathbf{F} \tilde{\mathbf{m}}_k = 0$$

Diese Gleichung kann als lineare und homogene Gleichung aufgeschrieben werden:

$$\mathbf{u}^T \mathbf{f} = 0$$

mit

$$\begin{aligned} \mathbf{u} &= [u_1 u_2, v_1 u_2, u_2, u_1 v_2, v_1 v_2, v_2, u_1, v_1, 1]^T \\ \mathbf{f} &= [F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}, 1]^T \end{aligned}$$

Gesucht ist ein neundimensionaler Vektor \mathbf{f} mit $\mathbf{u}^T \mathbf{f} = 0$ unter der Bedingung $\|\mathbf{f}\| = 1$ (um die triviale Lösung zu vermeiden). Unter dieser Bedingung wird $\|\mathbf{U}\mathbf{f}\|$ minimiert:

$$\sum_i (\tilde{\mathbf{m}}_{2i}^T \mathbf{F} \tilde{\mathbf{m}}_{1i})^2 \longrightarrow \min \quad (6.2)$$

und lässt sich vereinfacht darstellen durch:

$$\|\mathbf{U}\mathbf{f}\|^2 \longrightarrow \min \quad (6.3)$$

mit

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_n^T \end{pmatrix} = \begin{pmatrix} u_{11}u_{21} & v_{11}u_{21} & u_{21} & u_{11}v_{21} & v_{11}v_{21} & v_{21} & u_{11} & v_{11} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{1n}u_{2n} & v_{1n}u_{2n} & u_{2n} & u_{1n}v_{2n} & v_{1n}v_{2n} & v_{2n} & u_{1n} & v_{1n} & 1 \end{pmatrix} \quad (6.4)$$

Die Lösung für das gesuchte Minimierungsproblem

$$\min_f \|\mathbf{U}\mathbf{f}\|^2 \quad \text{unter } \|\mathbf{f}\| = 1$$

entspricht dem Eigenvektor zum kleinsten Eigenwert λ von $\mathbf{U}^T \mathbf{U}$. Da die Fundamentalmatrix eine Rang-2-Matrix ist und dieses Verfahren (im allgemeinen) eine Rang-3-Matrix liefern würde, gilt es die zu \mathbf{f} bzgl. der Frobeniusnorm nächste Rang-2-Matrix \mathbf{F} zu bestimmen. Diese lässt sich durch die Singulärwertzerlegung $\mathbf{F} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ mit $\mathbf{S} = \text{diag}(\sigma_1, \sigma_2, \sigma_3), \sigma_1 \geq \sigma_2 \geq \sigma_3$ (Diagonalmatrix) bestimmen - σ_i ist der *i*-te Singulärwert.

$$\mathbf{F} = \mathbf{U}\mathbf{S}'\mathbf{V}^T$$

liefert die gesuchte Rang-2-Matrix mit $\mathbf{S}' = \text{diag}(\sigma_1, \sigma_2, 0)$. Indem man den kleinsten Singulärwert σ_i auf Null setzt, erhält man die gesuchte Rang-2-Matrix. Der Epipol des ersten Bildes entspricht dabei der *i*-ten Spalte und der Epipol des zweiten Bildes entspricht der *i*-ten Zeile aus der Matrix \mathbf{V} . Vorteil dieses Verfahrens ist die direkte analytische Schätzung von \mathbf{f} , allerdings entspricht das zu minimierende Kriterium nicht dem euklidischen Abstand.

6.3 Das Korrespondenzproblem

Um nun für alle Punkte die korrespondierenden Punkte im anderen Bild zu berechnen, bedient man sich der Korrelation.

Mathematisch gesehen misst die Korrelation positive und negative lineare Zusammenhänge von Zufallsvariablen - "[...]der Durchschnitt der Abweichungsquadrate vom Durchschnitt eines statistischen Merkmals."¹

¹<http://de.wikipedia.org/wiki/Varianz>

Normalerweise werden dazu markante (Eck)Punkte in beiden Aufnahmen berechnet (mit einem Kantendetektor). Hier bietet sich an, die gefundenen Tropfen als markante Punkte zu verwenden. Das Vorgehen ist dann wie folgt: in beiden Aufnahmen werden zwei gleich große Korrelationsfenster definiert. Diese werden miteinander verglichen und für jedes Pixel wird ein Verschiebungsvektor d (Disparität) berechnet, der eine Ähnlichkeitsfunktion c maximiert. Dies kann in beide Richtungen durchgeführt werden: suche für ein festes Fenster in der linken Aufnahme ein passendes Fenster in der rechten Aufnahme und umgekehrt. Um diesen Korrelationswert zu berechnen, der zwischen -1 und $+1$ liegen kann, werden die Intensitäten (Grauwerte) beider markanten Punkte verglichen. Da nun die Fundamentalmatrix bekannt ist, kann der Suchraum bei der Korrespondenzfindung von der ganzen Bildebene auf die zugehörige epipolare Gerade eingeschränkt werden. Es muss also nur entlang dieser Geraden nach der höchsten Korrelation gesucht werden.

Seien $\mathbf{m}_1 = (u_1, v_1)$ sowie $\mathbf{m}_2 = (u_2, v_2)$ die markanten Punkte in der ersten bzw. zweiten Aufnahme und $(2n+1)$ sowie $(2m+1)$ die Breite und Höhe des Korrelationsfensters um das Zentrum des Punktes \mathbf{m}_1 . Dann wird die Korrelationsoperation ausgeführt, indem ein rechteckiger Suchbereich $(2d_u+1) \times (2d_v+1)$ auf den zweiten markanten Punkt \mathbf{m}_2 gelegt wird und die Intensitäten im Korrelationsfenster des Punktes \mathbf{m}_1 mit den Intensitäten im Suchbereich des zweiten Punktes \mathbf{m}_2 verglichen werden.

Das Korrelationsmaß $c : \mathbb{Z} \times \mathbb{Z} \rightarrow [-1, +1]$ an der Position $(\mathbf{m}_1, \mathbf{m}_2)$ ist definiert durch:

$$c(\mathbf{m}_1, \mathbf{m}_2) = \frac{\sum_{i=-n}^n \sum_{j=-m}^m [I_1(u_1+i, v_1+j) - \overline{I_1(u_1, v_1)}] \cdot [I_2(u_2+i, v_2+j) - \overline{I_2(u_2, v_2)}]}{(2n+1)(2m+1)\sqrt{\sigma^2(I_1) \cdot \sigma^2(I_2)}} \quad (6.5)$$

wobei:

- $\overline{I_k(u, v)} = \sum_{i=-n}^n \sum_{j=-m}^m \frac{I_k(u+i, v+j)}{[(2n+1)(2m+1)]}$, $k \in 1, 2$ - das arithmetische Mittel (ein Schätzer für den Erwartungswert)
- $\sigma^2(I_k) = \sqrt{\frac{\sum_{i=-n}^n \sum_{j=-m}^m I_k^2(u, v)}{(2n+1)(2m+1)} - \overline{I_k(u, v)^2}}$, $k \in 1, 2$ - ein Schätzer für die Varianz

Das Korrelationsmaß lässt sich weiter umformen zu:

$$c(\mathbf{m}_1, \mathbf{m}_2) = \frac{s \sum_{i=-n}^n \sum_{j=-m}^m \prod_{k=1}^2 I_k(u_k+i, v_k+j) - \prod_{k=1}^2 \sum_{i=-n}^n \sum_{j=-m}^m I_k(u_k+i, v_k+j)}{\sqrt{\prod_{k=1}^2 \left(s \cdot \sum_{i=-n}^n \sum_{j=-m}^m I_k^2(u_k+i, v_k+j) - \left(s \sum_{i=-n}^n \sum_{j=-m}^m I_k(u_k+i, v_k+j) \right)^2 \right)}} \quad (6.6)$$

Die Umformung zeigt die algorithmische Berechnung des Korrelationswertes $c(\mathbf{m}_1, \mathbf{m}_2)$: Die Intensitäten müssen im Korrelationsfenster aufsummiert (Zähler), quadratisch aufsummiert (Nenner) sowie gemischt multipliziert werden (Nenner).

6.4 Berechnung der 3D Punkte

Eine einfache Methode um den 3D Punkt zu rekonstruieren ist über die Projektionsmatrizen zweier korrespondierender Punkte.

Seien \mathbf{P}_1 und \mathbf{P}_2 die beiden Kameraprojektionsmatrizen, d.h. sie beschreiben das Abbildungsverhalten einer Lochkamera, $\hat{\mathbf{m}}_1 = (u_1, v_1, 1)$ und $\hat{\mathbf{m}}_2 = (u_2, v_2, 1)$ korrespondierende Bildpunkte, d.h. sie sind Bilder eines 3D-Punktes $M = (X, Y, Z)$, so gilt :

$$s_1 \tilde{\mathbf{m}}_1 = \mathbf{P}_1 \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (6.7)$$

$$s_2 \tilde{\mathbf{m}}_2 = \mathbf{P}_2 \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad \text{mit } s_1, s_2 \in \mathbb{R} \quad (6.8)$$

Nach weiteren Rechenschritten erhält man zusammengefasst:

$$\underbrace{\begin{pmatrix} p_{111} - u_1 p_{131} & p_{112} - u_1 p_{132} & p_{113} - u_1 p_{133} & p_{114} - u_1 p_{134} \\ p_{121} - v_1 p_{131} & p_{122} - v_1 p_{132} & p_{123} - v_1 p_{133} & p_{124} - v_1 p_{134} \\ p_{211} - u_2 p_{231} & p_{212} - u_2 p_{232} & p_{213} - u_2 p_{233} & p_{214} - u_2 p_{234} \\ p_{221} - v_2 p_{231} & p_{222} - v_2 p_{232} & p_{223} - v_2 p_{233} & p_{224} - v_2 p_{234} \end{pmatrix}}_{A(\mathbf{m}_1, \mathbf{m}_2)} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (6.9)$$

Gesucht ist ein Eigenvektor zum Eigenwert Null, um die triviale Lösung zu vermeiden. Da $A(\mathbf{m}_1, \mathbf{m}_2)$ Rang 4 haben wird, erhält man die nächste (bezüglich der Frobeniusnorm) Rang-3-Matrix durch Singulärwertzerlegung

$$A(\mathbf{m}_1, \mathbf{m}_2) = \mathbf{U} \begin{pmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & d_3 & 0 \\ 0 & 0 & 0 & d_4 \end{pmatrix} \mathbf{V}^T \quad (6.10)$$

D.h. \mathbf{U} und \mathbf{V} sind orthogonale Matrizen und die Singulärwerte d_i sind größer gleich Null. Der kleinste Singulärwert wird auf Null gesetzt. Sei z.B. $d_j = \min(d_1, d_2, d_3, d_4)$, dann ist $(v_{1j} \ v_{2j} \ v_{3j} \ v_{4j})$ (j-te Spalte von \mathbf{V}) ein Eigenvektor zum Eigenwert Null und liefert den rekonstruierten 3D-Punkt, indem die vierte Koordinate zu 1 normiert wird und dann die ersten drei Koordinaten betrachtet werden.

6.5 Anmerkungen

Zum Testen der 3D-Rekonstruktion wurde ein eigenständiges Tool erstellt. Hierbei müssen zunächst ein Bildpaar und die dazugehörigen Kalibrierungsdaten geladen werden.

Es besteht nun die Möglichkeit, verschiedene Punkte in einem Bild zu markieren und die korrespondierenden Punkte in dem anderen Bild zu finden. Dabei werden jeweils auch die zugehörigen epipolaren Linien zur Kontrolle im anderen Bild angezeigt. Sind alle korrespondierenden Punktepaare gefunden, lässt sich die 3D-Rekonstruktion durchführen. In einem weiteren Registerreiter werden die Ergebnisse davon angezeigt, in der 2D-Darstellung wird weiterhin eine Delaunay-Triangulierung mit eingefügt.

6.6 Metrik

Für die Bestimmung von Geschwindigkeiten oder von Abständen wird eine Metrik benötigt, die eine Umrechnung von Pixeln in mm vornimmt. Zur korrekten Bestimmung muss während der Kalibrierung das erste Bild direkt unterhalb der Düse aufgenommen werden. Die Metrik ergibt sich dann aus dem Vergleich der Größe eines Kalibrierungsquadrates in Pixeln und der bekannten realen Seitenlänge in mm. In dem Fall, dass ein gegebener Punkt in der Ebene der Düse liegt, z. B. auf der Lammelenkante, liefert diese einfache Methode den korrekten Wert der Metrik. In einer zweiten

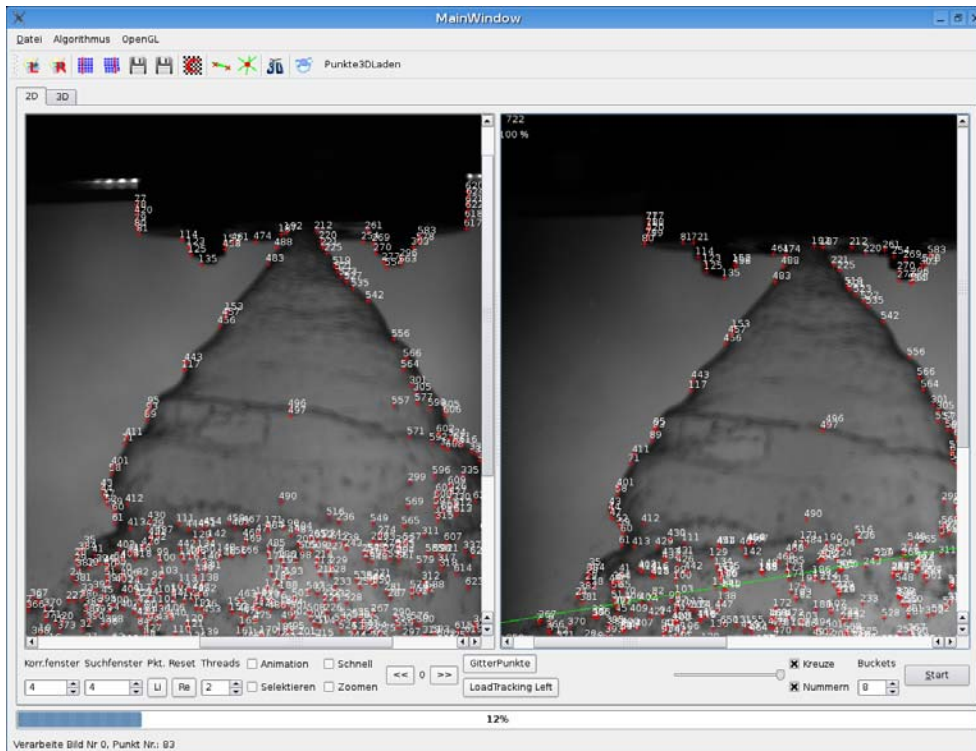


Abbildung 6.2: Tool zum Testen der 3D-Rekonstruktion

Methode wird davon ausgegangen, dass der Punkt auf dem Sprühkegel liegt. Aufgrund des aus der Segmentierung bekannten Sprühwinkels und des Austrittspunktes der Lamelle wird der Abstand des gewählten Punktes von der Kamera geschätzt und damit die reale Breite des Pixels berechnet.

6.7 Fazit

Anhand des Tools zum Testen der 3D-Rekonstruktion konnte man feststellen, dass die 3D-Rekonstruktion sehr stark von der Qualität der Kalibrierdaten abhängt, da bei ungenauen Angaben eine fehlerhafte Fundamentalmatrix berechnet wird, sodass die Epipolarlinien nicht exakt gefunden werden. Dadurch werden natürlich öfter falsche Korrespondenzen gefunden. Sind die Daten jedoch gut berechnet, liefert die 3D-Rekonstruktion ein akzeptables Ergebnis (vgl. Abb. 6.3).

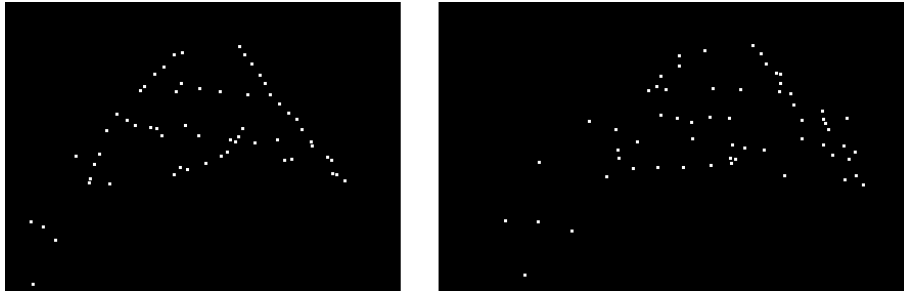


Abbildung 6.3: Rekonstruierte Punkte

Kapitel 7

Visualisierung

Inhalt

7.1	Einleitung	89
7.2	Datenvisualisierung	89
7.3	Stereovisualisierung	90
7.4	VTK	92
7.5	Fazit	92

7.1 Einleitung

Nachdem die Ergebnisse der Segmentierung (vgl. Kapitel 4), des Trackings (vgl. Kapitel 5) und der 3D-Rekonstruktion (vgl. Kapitel 6) berechnet wurden, müssen diese visualisiert werden. Hierzu werden Methoden von OpenGL [Shr05] benutzt. Außerdem wurde eine Visualisierung mit Hilfe des Visualization ToolKit (VTK) [Sch00] prototypisch realisiert.

Die Ergebnisse werden als synthetische Daten auf das zuvor erstellte Video gezeichnet. Hierbei ist es dem Benutzer überlassen, welche Daten angezeigt werden sollen. Die Auswahl erfolgt über das Visualisierungswidget im Hauptfenster von **3D-AugTracer**.

Da es möglich sein sollte, zwei ausgewertete Versuche (Projekte) miteinander zu vergleichen, können die Ergebnisse beider Projekte gleichzeitig angezeigt werden. Um die Ergebnisse beider Projekte unterscheiden zu können, werden zur Visualisierung verschiedene Farben genutzt. Die Visualisierung kann sowohl bei Mono- als auch bei Stereo-Projekten erfolgen.

7.2 Datenvisualisierung

Im Wesentlichen werden die Ergebnisse der Segmentierung und des Trackings visualisiert. Aus dem Berechnungsschritt der Segmentierung wurden zwei Ausgleichsgeraden pro Bild, jeweils für beide Seiten, errechnet. Sie liegen im Datencontainer als Typ `QLine` vor. Der Typ `QLine` speichert eine Gerade in Form von zwei Punkten, die jeweils vom Typ `QPoint` sind. Ein `QPoint` hat jeweils eine X- und Y-Koordinate. Zwischen den beiden Punkten der Geraden wird dann mit `GL_LINE` eine Linie gezeichnet [Shr05].

Darüber hinaus erhält man eine linke und rechte Ausgleichsgerade, die über alle Bilder gemittelt wird. Der Schnittpunkt dieser beiden gemittelten Geraden ist der Startpunkt für die Visualisierung dieser beiden Geraden. Alle diese Informationen werden im Datencontainer gespeichert.

Der Endpunkt dieser Ausgleichsgeraden ergibt sich durch die horizontalen und vertikalen Komponenten ausgehend vom Startpunkt. Der Endpunkt ist lediglich zum Zeichnen wichtig. Er hat

keine chemietechnische Bedeutung. Auch die gemittelten Ausgleichsgeraden werden mit `GL_LINE` visualisiert [Shr05].

Ein weiteres Ergebnis der Segmentierung sind Sprühwinkel (vgl. Kapitel 4.2). Diese setzen ebenfalls beim Schnittpunkt der beiden gemittelten Ausgleichsgeraden an. Der Winkel wird ausgehend vom Lot als Linie mittels `GL_LINE` dargestellt, das Lot selbst wird dabei nicht angezeigt.

Aus den Ergebnissen der Segmentierung lässt sich nun auch ein 3-dimensionaler Kegel errechnen und zeichnen. Die Spitze des Kegels liegt genau in dem Schnittpunkt der beiden gemittelten Ausgleichsgeraden, der Radius ergibt sich aus dem Wert gemittelten Sprühwinkels. Der Kegel wird dann noch transparent angezeigt.

Aus der Segmentierung ergeben sich auch die Lamellenkoordinaten, welche ausgehend von einem Startpunkt mittels `GL_LINE` stückweise gezeichnet werden können.

Aus der Tropfenanalyse werden Tropfen im Datencontainer gespeichert. Dabei wird ihr Durchmesser und ihre Koordinaten gespeichert. Diese Informationen wird wahlweise auf einem 2D-Bild oder im 3D-Raum angezeigt. Die unterschiedlichen Durchmesser der Tropfen werden zusätzlich durch unterschiedliche Farben repräsentiert. Die 3-dimensionale Darstellung der Tropfen basiert auf den Berechnungsergebnissen der 3D-Rekonstruktion (vgl. Kapitel 6). Sowohl die 2D- als auch die 3D-Visualisierung der Tropfen erfolgt mit Hilfe der OpenGL Utility Library `GLU`. Mit `gluSphere` wird eine Kugel an die Koordinate des Tropfens gezeichnet. Der Radius für jede Kugel ergibt sich ebenfalls aus der Tropfenanalyse. Kugeln unterschiedlicher Radien werden mit unterschiedlichen Farben dargestellt. Für die 3D-Visualisierung wird zudem die z-Achse skaliert.

Ein weiteres Ergebnis der Berechnungen sind Vektorfelder, welche durch Anwendung der Trackingverfahren berechnet werden (vgl. Kapitel 5). Hierbei stehen Ergebnisse des Blockmatching-, KLT- und Partikeltrackings im Datencontainer zur Verfügung. Die Vektorfelder werden durch Pfeile dargestellt, welche mittels der `GLU`-Bibliothek realisiert werden. Für jeden Pfeil wird ein individueller Winkel berechnet. Da die Tracking-Verfahren Verschiebevektoren liefern (in x- und y-Richtung), ist die Berechnung der Winkel einfach durchzuführen. Die Länge ist hingegen bei allen Pfeilen gleich.

Aus den Verschiebevektoren lassen sich auch Stromlinien zu den durchgeführten Verfahren darstellen. Hierbei beginnt man an einem Punkt und wendet den Verschiebevektor für diesen Punkt an. Zwischen Start- und Zielpunkt wird nun eine Linie gezeichnet (`GL_LINE`). Dieses Vorgehen wendet man anschließend iterierend auf den Zielpunkt an, wodurch man eine durchgängige Linie erhält. Dies muss für weitere Startpunkte wiederholt werden. Da die Stromlinien auch Geschwindigkeiten darstellen, werden sie mit unterschiedlichen Farben für unterschiedliche Geschwindigkeiten visualisiert.

Die Trackingverfahren berechnen in allen Fällen 2-dimensionale Vektorfelder. Durch die 3D-Rekonstruktion stehen jedoch auch 3-dimensionale Vektorfelder zur Visualisierung zur Verfügung (vgl. Kapitel 6). Hierbei werden für jeden darzustellenden Pfeil zwei Winkel vom Lot aus berechnet.

7.3 Stereovisualisierung

Die Stereovisualisierung stellt eine Erweiterung zur normalen Datenvisualisierung dar. Unter der Voraussetzung, dass in **3D-AugTracer** ein Stereoprojekt erstellt wurde, ist es möglich, die aufgenommene Versuchsanordnung holographisch über ein HMD (head mounted display) oder eine geeignete Projektionsanlage (z.B. mit dem Infitec-Verfahren [Inf]) anzuzeigen. Zur Umsetzung wurden zwei verschiedene Ansätze ausprobiert, die Toe-In-Methode und die Off-Axis-Methode [LH97].

Bei der Toe-In Methode werden die Sichtpyramiden leicht zueinander geneigt, so dass sich die Projektionsebenen jeweils in ihrer Mitte schneiden (siehe Abb. 7.1). Die Sichtpyramiden bleiben symmetrisch. Das Bild ist jedoch durch diesen Aufbau mit einem Fehler behaftet. Ein Punkt, der bei der Betrachtung genau in Entfernung der Projektionsfläche liegt, müsste innerhalb der modellierten Szene auf der winkelhalbierenden Ebene zwischen den Projektionsebenen liegen. Punkte dort werden ausser in der Schnittgeraden jedoch auf zwei Punkte auf den Projektionsebenen mit

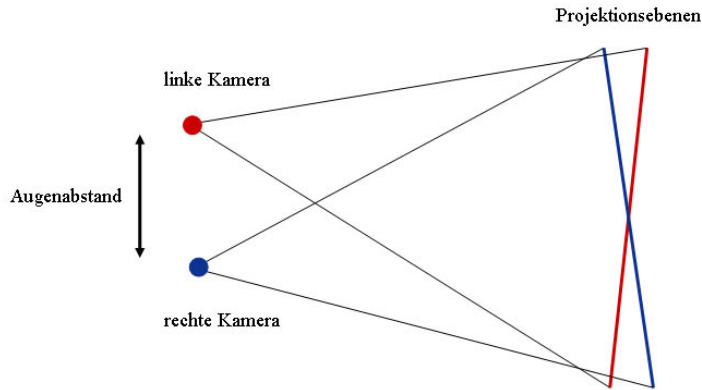


Abbildung 7.1: Toe-In Methode

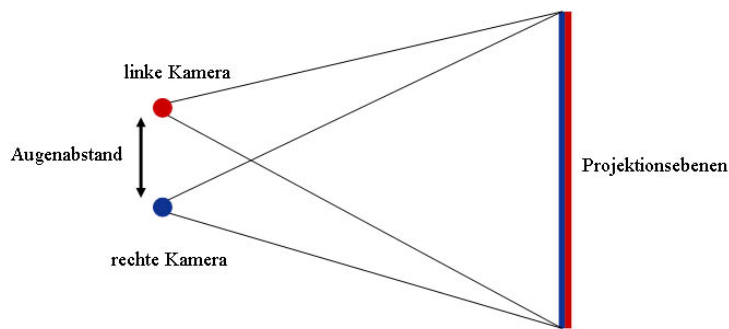


Abbildung 7.2: Off-Axis Methode

nichtidentischen Koordinaten projiziert. Frühe Versuche in der Projektgruppe zeigten, dass dieses Verfahren Unbehagen bei längerer Betrachtung auslöste. Grund dafür ist der Projektionsfehler, der von der Bildmitte nach außen hin immer gravierender zum tragen kam.

Aus diesem Grund wurde der Toe-In Ansatz nicht weiter verfolgt, und die Off-Axis Methode ausprobiert. Diese Methode behebt den Projektionsfehler, indem asymmetrische Sichtpyramiden bei paralleler Blickrichtung verwendet werden. Die Projektionsebenen sind daher identisch, und durch die Asymmetrie der Sichtpyramiden wird erreicht, dass sie für beide Kameras dieselbe Begrenzung in vertikaler Richtung haben (siehe Abb. 7.2). Dieser Aufbau entspricht daher auch eher dem Wiedergabesystem, welches ebenfalls nur eine Leinwand nutzt. Da diese Methode weniger Unbehagen erzeugt, wird sie der Toe-In Methode vorgezogen. Die erforderlichen asymmetrischen Sichtpyramiden werden von OpenGL unterstützt.

Die endgültige Umsetzung in **3D-AugTracer** bedient sich der Off-Axis Methode mit der Modifizierung, dass die Sichtpyramiden der beiden Kameras nicht asymmetrisch eingeschränkt werden, sondern symmetrisch bleiben. Dies hat zwar den Effekt, dass im linken und rechten Bildrand die Bildinformationen nur von einer Kamera dargestellt werden. Das ist aber bewusst so beabsichtigt, da sonst wichtige Bildbereiche abgeschnitten und somit wichtige Informationen fehlen würden.

Zur Darstellung wurde passives Stereo gewählt. Die Bilder für das linke und das rechte Auge werden also nicht per Farb- oder Zeitmultiplex ausgegeben, sondern parallel nebeneinander dargestellt. Da OpenGL aber nur eine Kamera unterstützt, muss dies auch nach einander geschehen. Dazu wird bei einem Stereoprojekt zuerst die Kamera um den halben Augenabstand nach links verschoben. Die vom Projekt erzeugten Videos werden dabei nach einander als Textur auf ein Rechteck (`GL_QUADS`) gelegt, welches soweit nach links bzw. rechts verschoben wird, dass hinterher beim Zusammenführen

der beiden Bilder ein sinnvolles Bild entsteht. Die synthetischen Daten (Sprühkegel, 3D-Tropfen, usw.) werden dann als dreidimensionale Objekte zwischen Kamera und Rechteck positioniert. Nachdem die Szene in einem Bild festgehalten wurde, wird die Kamera nun für das Bild des rechten Auges veroben und die Prozedur beginnt von vorne.

7.4 VTK

Für die Stereowiedergabe wurde zunächst eine Implementierung mit der Bibliothek VTK (Visualization Toolkit) durchgeführt. VTK bot zwar die vorgefertigten Möglichkeiten der Darstellung von Pfeilen, Stromlinien, etc. an, jedoch ließ es sich nicht ohne erheblichen Aufwand in QT integrieren. Letztendlich war die unzureichende Dokumentation der Bibliothek der ausschlaggebende Grund, weshalb auf eine Implementierung mit VTK verzichtet wurde. Statt dessen wird die freie Bibliothek OpenGL verwendet.

7.5 Fazit

Bei der Implementierung der Visualisierung zeigte sich, dass sich die 2-dimensionale Darstellung der errechneten Daten relativ leicht realisieren ließ. Die größten Probleme lagen hierbei bei der korrekten Berechnung der Winkel zur Darstellung der Sprühwinkel, des Kegels, der Ausgleichsgeraden für jeden Frame und der Pfeile für die Vektorfelder, da die Achsen des (3-dimensionalen) Koordinatensystems zur Darstellung der einzelnen Ergebnisse teilweise gedreht werden mussten. Die 3-dimensionale Visualisierung der Daten war hingegen schwieriger bzw. wurde letztendlich aufgegeben, weil die Rekonstruktion der jeweiligen 3D-Koordinaten keine zufriedenstellenden Ergebnisse lieferte.

Die Benutzung des Visualization ToolKit (VTK) wurde auf Grund der fehlenden Unterstützung für die Stereovisualisierung nicht weiter verfolgt.

Kapitel 8

Objektorientierte Realisierung

Inhalt

8.1	Einleitung	93
8.2	UML-Modelle	93
8.3	Klassenbeschreibungen wichtiger Klassen	96
8.3.1	Kalibrierung	96
8.3.2	Denoising	96
8.3.3	Dewarping	96
8.3.4	Segmentierung	97
8.3.5	Tropfenanalyse	97
8.3.6	Tracking	97
8.3.7	3D-Rekonstruktion	98
8.3.8	Visualisierung	98
8.3.9	Frequenzanalyse	98
8.4	Werkzeuge und verwendete Bibliotheken	98

8.1 Einleitung

Die oben erwähnten Verfahren wurden objektorientiert implementiert. In diesem Kapitel wird näher auf die objektorientierte Programmierung sowie die Klassenhierarchie eingegangen.

Bei der objektorientierten Programmierung liegt eine Aufteilung der zu beschreibenden Welt in Objekte mit ihren Eigenschaften und Operationen zugrunde. Ergänzt wird dies durch das Konzept der Klasse, bei dem Objekte aufgrund ähnlicher Eigenschaften zusammengefasst werden. Die Struktur eines Objekts wird durch die Attribute (auch Eigenschaften) seiner Klassendefinition festgelegt. Das Verhalten des Objekts wird von den Methoden der Klasse bestimmt.

Bei dem vorliegenden Projekt wurde zunächst grob in GUI- und Work-Klassen unterteilt. Letztere wurden in die einzelnen Arbeitsschritte, die in den vorherigen Kapiteln beschrieben wurden, eingeteilt. Jeder einzelne Bereich wurde nach individuellen Kriterien weiter verfeinert. Zusätzlich gibt es Klassen, die einerseits für die Datensicherung zuständig sind und andererseits das Zusammenspiel aller Klassen steuern (siehe Abb. 8.1).

8.2 UML-Modelle

Um den Datenfluss innerhalb des Projekts besser überblicken zu können wurde ein grobes Diagramm erstellt, welches die Abhängigkeiten der einzelnen Bereiche untereinander verdeutlicht (siehe

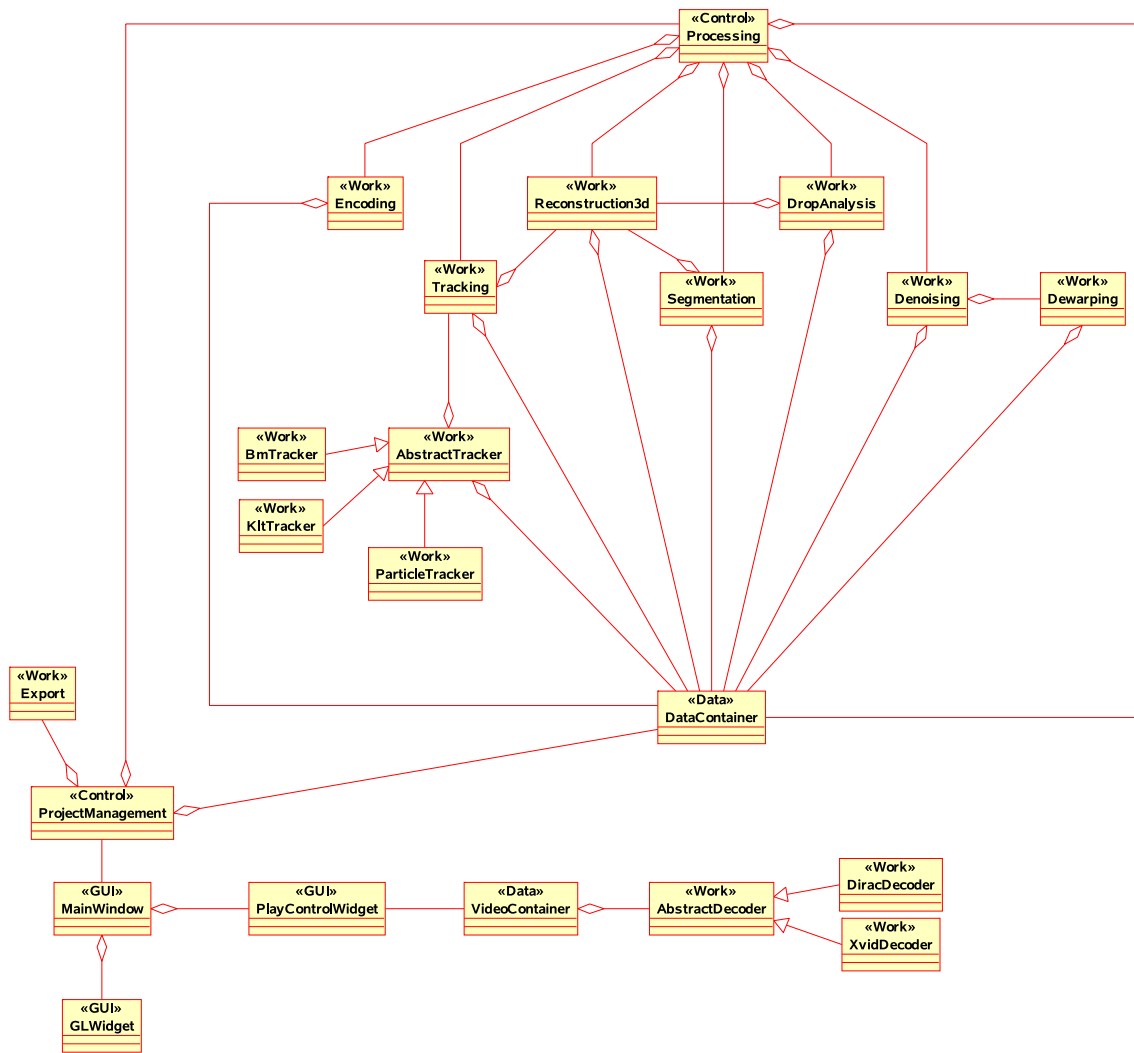


Abbildung 8.1: Klassendiagramm

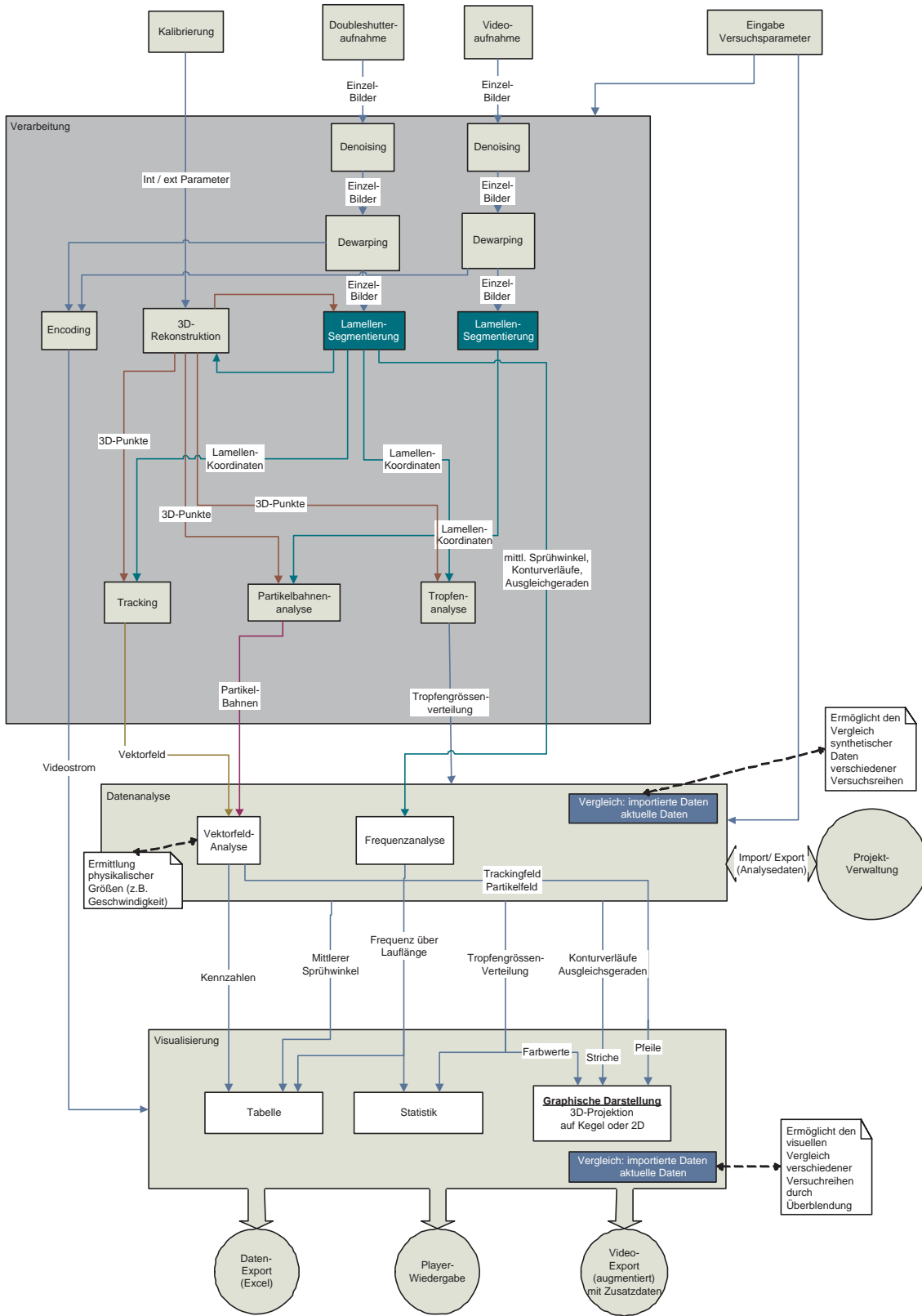


Abbildung 8.2: Abhängigkeitsdiagramm

Abb. 8.2). Dieses ist unterteilt in die Bereiche der Dateneingabe, der Verarbeitung und Visualisierung. Am Ende steht die Datensicherung und der Export der ermittelten Ergebnisse.

Bei der Dateneingabe handelt es sich um die aufgenommenen Bilder und die Kalibrierungsdaten, die unabhängig vom Hauptprogramm ermittelt werden. Außerdem müssen Angaben zu den Versuchsparametern vom Benutzer gemacht werden.

Die Verarbeitung startet in jedem Fall mit der Aufbereitung der Bilder durch Denoising und Dewarping. Daraufhin werden wahlweise die weiteren Verfahren durchgeführt, wobei die Segmentierung eine zentrale Rolle spielt, da sie die Voraussetzung für gute Ergebnisse der anderen Berechnungen ist.

Die Visualisierung der berechneten Werte erfolgt zum einen in Form von Tabellen, zum anderen graphisch als Projektion auf den Kegel, wahlweise ist auch eine stereographische Darstellung möglich.

Sämtliche Daten können in Form von Excel-Tabellen exportiert werden, zusätzlich ist es möglich, sich ein Video der Lamelle mit den darauf projizierten graphischen Ergebnissen ausgeben zu lassen. Um die Schritte beim Anlegen eines neuen Projektes zu verdeutlichen, wurde ein Sequenzdiagramm erstellt.

8.3 Klassenbeschreibungen wichtiger Klassen

8.3.1 Kalibrierung

8.3.1.1 Klasse *Calibrate*

Die Klasse zur Kalibrierung der Kamera. In dieser Klasse sind die Methoden zur Kalibrierung implementiert, einschließlich Levenberg-Marquardt-Methode zur Optimierung.

8.3.1.2 Klasse *ProjectParameter*

Über diese Klasse wird ein XML-File mit allen Projektparametern erstellt, das u. a. die intrinsischen und extrinsischen Kameraparameter enthält. Dieses File wird später in **3D-Augtracer** eingelesen.

8.3.2 Denoising

8.3.2.1 Klasse *Denoising*

Diese Klasse stellt die Rauschreduzierungsfunctionalität zur Verfügung und sendet die bearbeiteten Bilder an das Dewarping weiter.

8.3.3 Dewarping

8.3.3.1 Klasse *Dewarping*

Diese Klasse ist die Hauptklasse der Dewarping-Functionalität. Sie stellt alle benötigten Methoden zur Erzeugung eines unverzerrten Bildes zur Verfügung.

8.3.3.2 Klasse *Refpoint*

Diese Klasse dient der Berechnung von unverzerrten Gitterpunkten, die eine wichtige Grundlage für den Dewarping-Algorithmus darstellt.

8.3.3.3 Klasse *ImageCenter*

Diese Klasse stellt grundlegende Methoden für die Durchführung von Berechnungen auf Bildern zur Verfügung.

8.3.4 Segmentierung

8.3.4.1 Klasse *Segmentation*

Bei dieser Klasse handelt es sich um die Hauptklasse der Lamellen-Segmentierung, es werden die nötigen Verfahren (Sobel, Otsu, Kantentracking, Template Matching) gesteuert.

8.3.4.2 Klasse *ConeEdge*

Die beim Kanten-Tracking erhaltene Information wird in dieser Klasse gespeichert.

8.3.4.3 Klasse *WaveletSegmentation*

Die Klasse WaveletSegmentation enthält Methoden zur Berechnung des Hölder Exponenten und sucht den Düsenapparat mit der Klasse Scanline.

8.3.4.4 Klasse *Scanline*

Die Klasse Scanline wird von der Klasse WaveletSegmentation aufgerufen, um den Mittelpunkt des Düsenapparates zu finden.

8.3.4.5 Klasse *Container*

Auf Grund der grossen Datenmengen, die bei der Waveletsegmentierung anfallen, stellt diese Klasse ein Interface für Dateiauslagerung zur Verfügung. Es bestehen mehrere Implementierungen für verschiedene Datentypen.

8.3.5 Tropfenanalyse

8.3.5.1 Klasse *Dropanalysis*

Diese Klasse steuert die Tropfenanalyse, generiert entsprechende Templates und wertet die Ergebnisse aus bzw. berechnet die Tropfenverteilung.

8.3.5.2 Klasse *Drop*

In dieser Klasse ist das eigentliche Template Matching-Verfahren implementiert, um zu einem gegebenen Template die besten Übereinstimmungen zu finden. Die Anpassung des Schwellenwertes erfolgt automatisch.

8.3.6 Tracking

8.3.6.1 Klasse *Tracking*

Aus dieser Klasse heraus werden die jeweiligen Tracking-Verfahren initialisiert und gestartet.

8.3.6.2 Klasse *VectorField*

Ein Objekt vom Typ Vectorfield ist das Ergebnis jeden Trackings. In einem Raster mit einstellbarer Rasterweite werden die berechneten Verschiebungsvektoren an jedem Rasterpunkt gespeichert. Werden Werte zwischen diesen Stützstellen benötigt, so werden die Vektoren aus den vier umliegenden Punkten linear interpoliert.

8.3.6.3 Klasse *HalfEdge*

Die Klasse *HalfEdge* stellt eine 2D-Half-Edge-Datenstruktur zur Verfügung, die beim Einfügen neuer Punkte automatisch so neu vernetzt wird, dass jederzeit eine Delaunay-Triangulierung vorliegt. Diese Art der Triangulierung wird vom KLT-Tracking verwendet, um die berechneten Verschiebungsvektoren für die rasterartige Speicherung im *VectorField* aufzubereiten (vgl. Kapitel 5.2).

8.3.7 3D-Rekonstruktion

8.3.7.1 Klasse *Epipolarline*

Diese Klasse stellt die Methoden zur Berechnung der Fundamentalmatrix zur Verfügung. Auch die Rekonstruktion des Punktes im 3D-Raum wird hier durchgeführt.

8.3.7.2 Klasse *Correlation*

In dieser Klasse werden die verschiedenen Verfahren zur Korrelation implementiert.

8.3.8 Visualisierung

8.3.8.1 Klasse *GLWidget*

Diese Klasse dient zum Anzeigen der aufgenommenen Videos und stellt Methoden zur Visualisierung der synthetischen Daten zur Verfügung.

8.3.9 Frequenzanalyse

8.3.9.1 Klasse *FrequencyAnalysisCore*

Die Kern-Klasse stellt alle benötigten Methoden statisch zur Verfügung, die für die Frequenzanalyse benötigt werden. Für die Berechnung der Analyse-Verfahren wird zum Teil auf die *WaveletLib*-Klasse zurückgegriffen.

8.3.9.2 Klasse *WaveletLib*

Da einige Berechnungen bzgl. Fourier-Transformationen und Wavelet-Transformationen vereinheitlicht werden sollen, stellt diese Klasse die benötigten Methoden bereit. Zu großen Teilen handelt es sich dabei um Wrapper-Methoden, die auf die FFTW-Bibliothek abgebildet werden.

8.4 Werkzeuge und verwendete Bibliotheken

Verwendete Bibliotheken:

- **newmat**
Die *newmat*-Bibliothek[New] wird für die Matrizenberechnung vor allem bei der Kamera-Kalibrierung (vgl. Kapitel 3.2.1) und der 3D-Rekonstruktion (vgl. Kapitel 6) verwendet. Sie stellt u.a. auch Methoden für die Singularwertzerlegung und Eigenvektorberechnung zur Verfügung.
- **Videocodecs**
Es werden die beiden Codecs *XviD*[Xvi] und *Dirac*[Dir] zum Erstellen der Videos verwendet. Da die Darstellung der Videos bei Verwendung des *Dirac*-Codecs jedoch nicht zufriedenstellend ist, sollten Videos hauptsächlich mit dem *XviD*-Codec erstellt werden.
- **fftw**
Die Bibliothek[FFT] stellt die für die Berechnung von Wavelets (vgl. Kapitel 4.5) und auch bei der Frequenzanalyse (vgl. Kapitel 4.3) benötigten Fourier-Transformationen zur Verfügung.

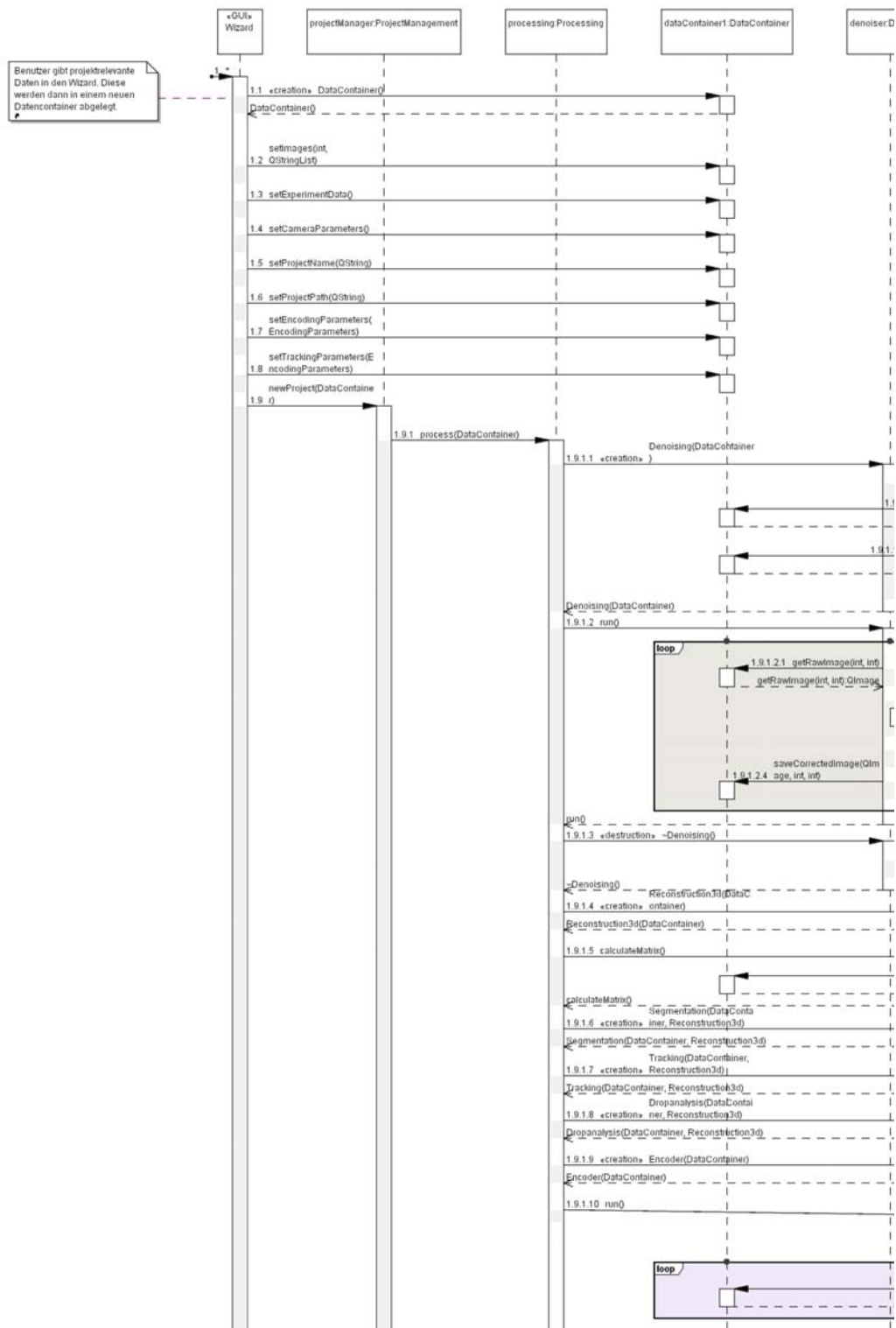


Abbildung 8.4: Sequenzdiagramm zum Erstellen eines neuen Projektes (1/9)

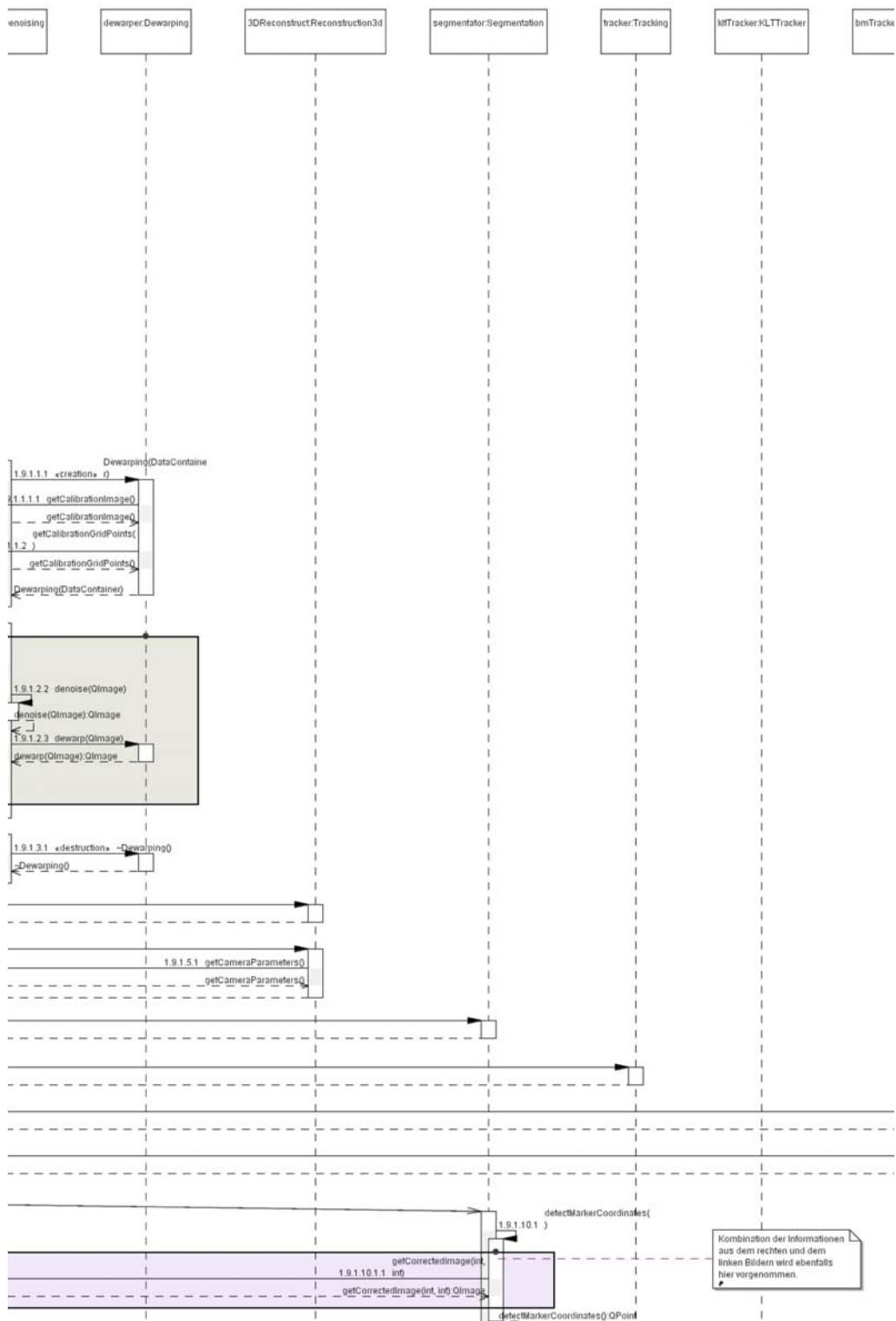


Abbildung 8.5: Sequenzdiagramm zum Erstellen eines neuen Projektes (2/9)

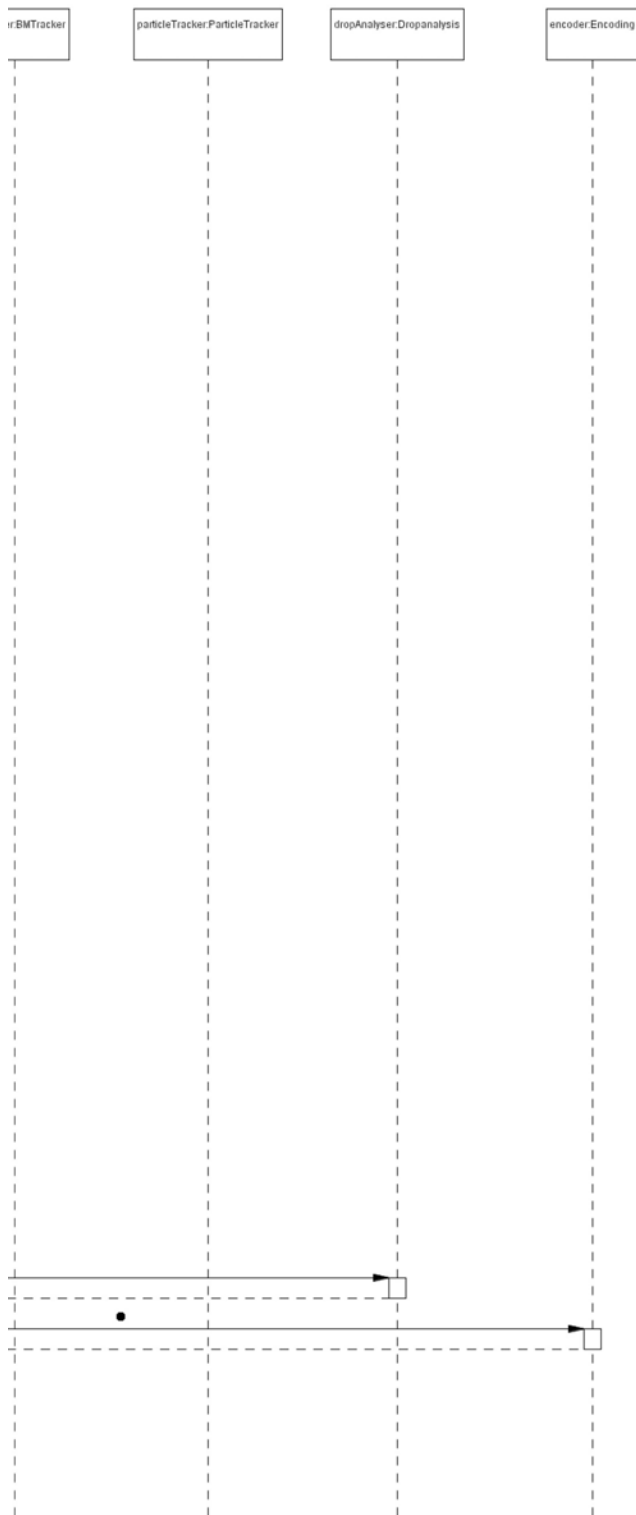


Abbildung 8.6: Sequenzdiagramm zum Erstellen eines neuen Projektes (3/9)

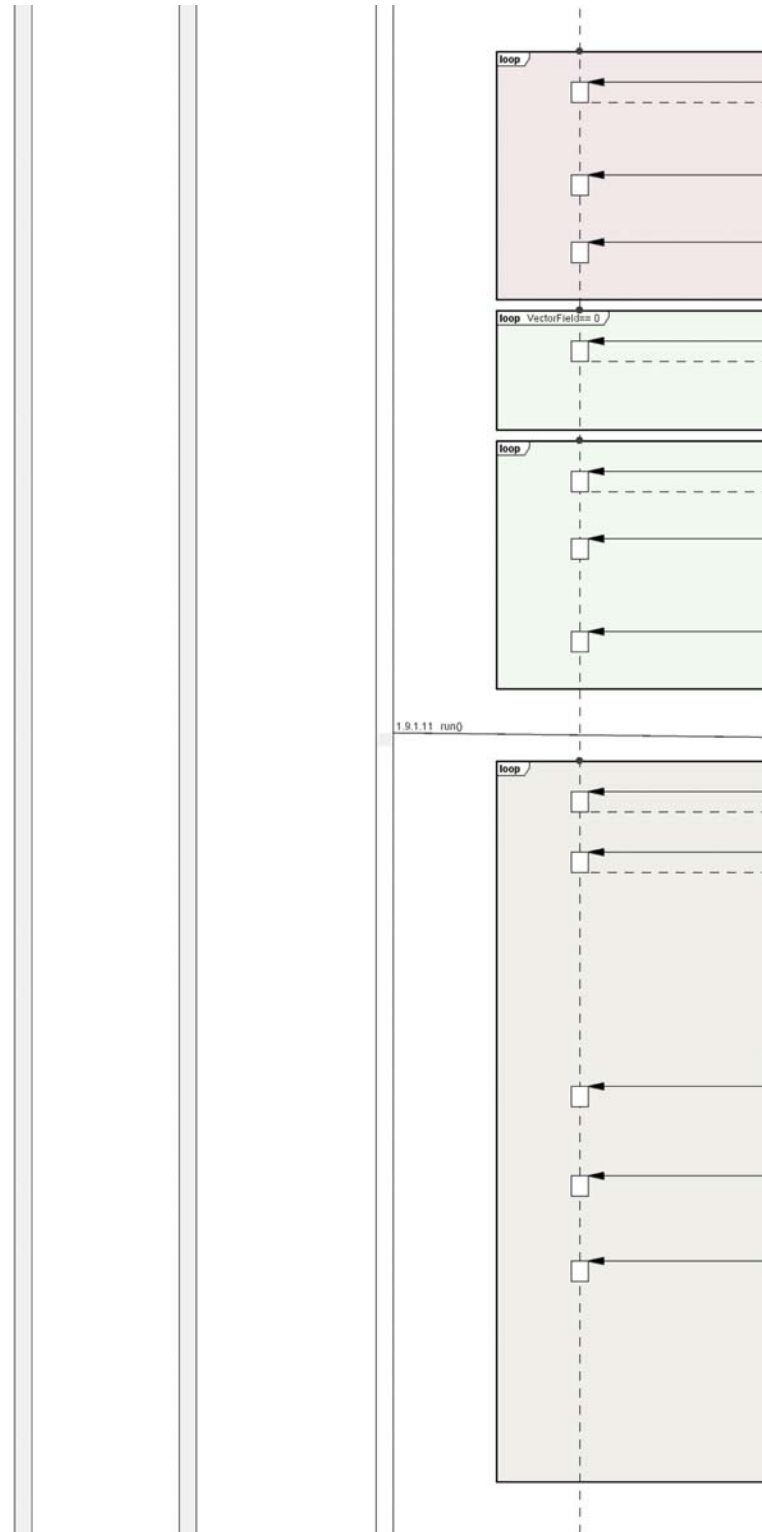


Abbildung 8.7: Sequenzdiagramm zum Erstellen eines neuen Projektes (4/9)

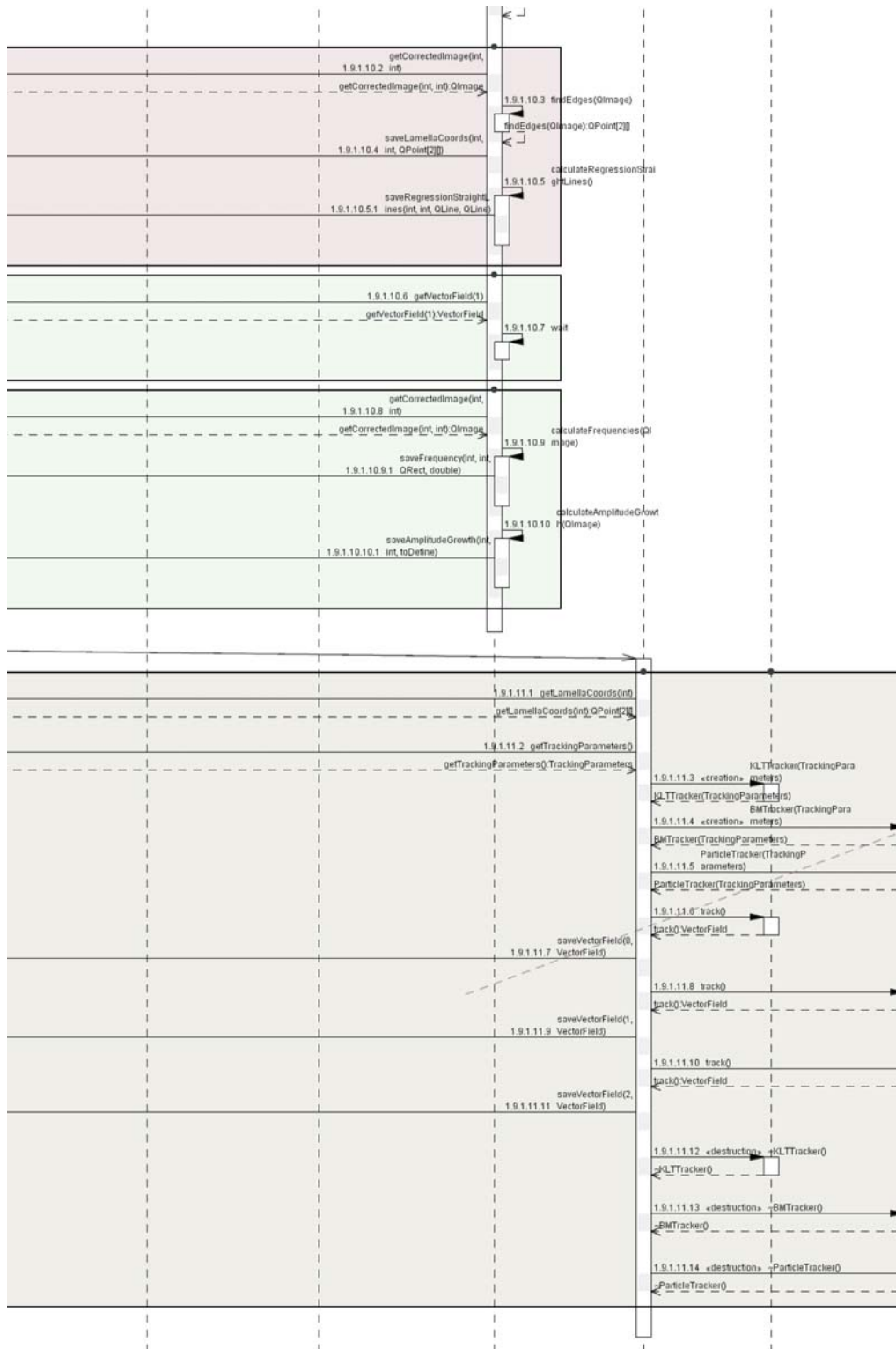


Abbildung 8.8: Sequenzdiagramm zum Erstellen eines neuen Projektes (5/9)

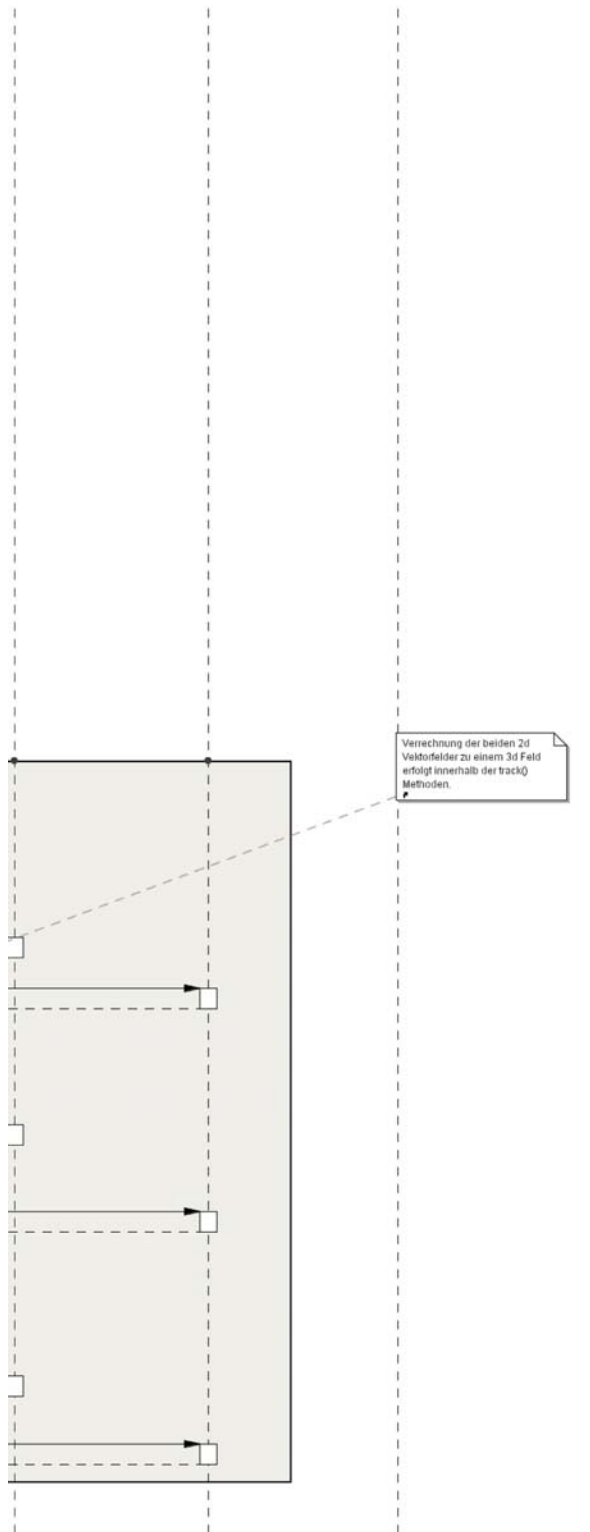


Abbildung 8.9: Sequenzdiagramm zum Erstellen eines neuen Projektes (6/9)

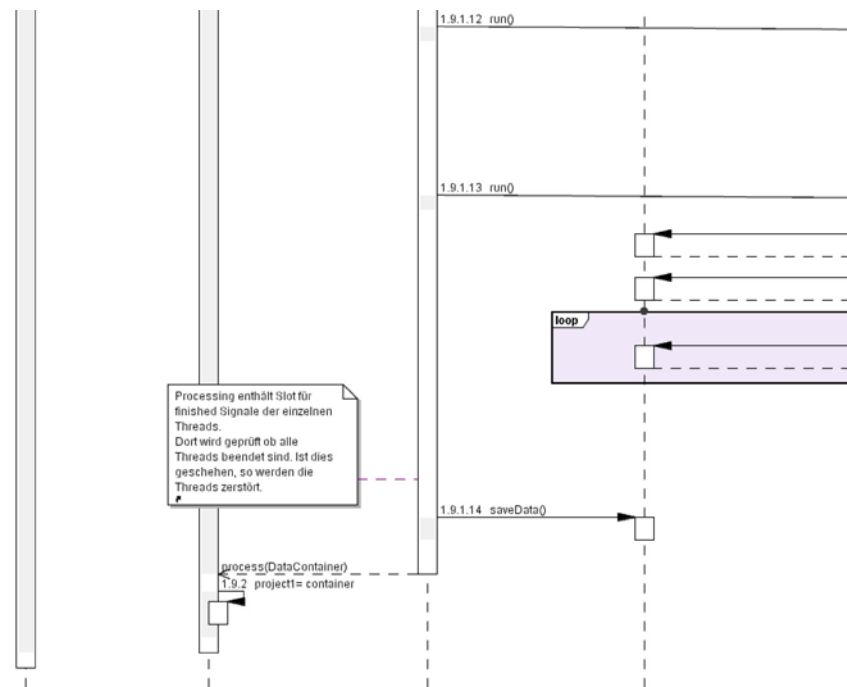


Abbildung 8.10: Sequenzdiagramm zum Erstellen eines neuen Projektes (7/9)



Abbildung 8.11: Sequenzdiagramm zum Erstellen eines neuen Projektes (8/9)

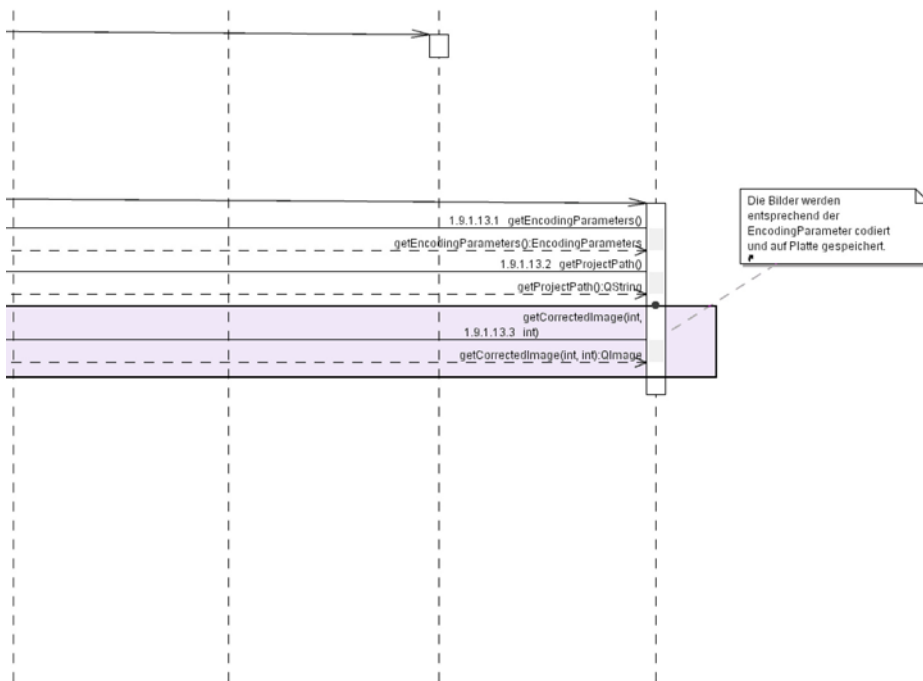


Abbildung 8.12: Sequenzdiagramm zum Erstellen eines neuen Projektes (9/9)

- **OpenGL**

Die gesamte graphische Visualisierung wurde mit OpenGL[Opeb] realisiert. So werden z.B. die Vektorfelder mit Hilfe von Pfeilen und die Tropfen als kleine Kugeln dargestellt (vgl. Kapitel 7).

- **kl**

Diese Bibliothek[KLT] realisiert eines der verwendeten Trackingverfahren (vgl. Kapitel 5.2).

- **wali**

Diese Bibliothek[WAI] stellt die diskrete Wavelettransformation zur Verfügung, über die auch das Denoising realisiert wird (vgl. Kapitel 3.4).

- **mysql**

Die Datenspeicherung erfolgt innerhalb einer MySQL Datenbank[MyS] (vgl. Kapitel 9.1), welche einen einfachen und effizienten Zugriff auf die ermittelten Daten ermöglicht.

Kapitel 9

Persistenz

Inhalt

9.1	Einleitung	111
9.2	Datenexport	112
9.2.1	Excel-Export	112
9.2.2	Video-Export	113

9.1 Einleitung

Für die dauerhafte Sicherung der Daten wird eine MySQL-Datenbank genutzt. In dieser werden für jeden Berechnungsschritt eine oder mehrere Tabellen angelegt, in denen die erhaltenen Ergebnisse verwaltet werden. Beim Anlegen eines neuen Projektes wird als letzter Schritt nach der Berechnung die Datenbank für das Projekt erstellt. Im Einzelnen gibt es diese Tabellen mit dem folgenden Inhalt:

- Allgemeine Parameter
Hierunter fallen der Projektname sowie die Eigenschaften und Anzahl der aufgenommenen Bilder.
- Thread Parameter
In dieser Tabelle werden Information darüber gespeichert, welche Berechnungen bereits ausgeführt wurden.
- Kamera Parameter
Hier stehen die aus dem Kalibrierungstool erhaltenen extrinsischen und intrinsischen Parameter der linken und rechten Kamera.
- Versuchsdaten
Hier werden die allgemeinen Eigenschaften des durchgeführten Experiments gespeichert, wie z. B. die verwendete Flüssigkeit und der Abstand der Kameras zur Düse.
- Encoding Parameter
Es werden die Angaben zum verwendeten Videocodec abgelegt.
- Dewarping Parameter
Es werden die Daten des zum Dewarping benötigten Gitters gespeichert.
- Denoising Parameter
Einzelne Angaben wie der verwendete Threshold oder der Typ des Wavelets für das Denoising werden in diese Tabelle geschrieben.

- Segmentierungs-Parameter
Die gemittelten Daten sowie die Sprühwinkel und die Ausgleichsgeraden eines jeden Bildes stehen in dieser Tabelle zur Verfügung. Weiterhin werden hier Angaben wie die Koordinaten des Markers angeführt.
- Tracking Parameter
Unter die Tracking Parameter fallen die Art des genutzten Trackings, die Bereiche, in denen getrackt wurde, und die Angaben zu den Optionen für die einzelnen Verfahren. Außerdem werden die sich ergebenden Vektorfelder gesichert.
- Rekonstruktions-Parameter
Für die Rekonstruktion werden die Koordinaten der 3D-Punkte in den einzelnen Bildern benötigt, weiterhin die Angaben über z. B. den Korrelationswert und das Clipping-Rechteck. Auch die 3D-Vektorfelder werden abgespeichert.
- Tropfenanalyse Parameter
Zu den gespeicherten Tropfen-Parametern zählen zum einen die Koordinaten eines Tropfens und sein Radius, zum anderen, zu welchem Frame und welcher Kamera er gehört. Auch der Bereich, in denen nach Tropfen gesucht wurde, wird mit angegeben. Die errechnete Tropfenverteilung wird ebenfalls gesichert.

Es wird eine neue Datenbank pro Projekt angelegt, wobei für noch nicht berechnete Daten leere Tabellen erstellt werden, die nachträglich gefüllt werden können.

9.2 Datenexport

Um die ermittelten Daten auch außerhalb von **3D-AugTracer** verwenden zu können, stehen Möglichkeiten zum Datenexport zur Verfügung.

9.2.1 Excel-Export

Der Excel-Export geschieht in Form eines XML-Dokuments, das in Excel als Tabelle importiert werden kann. Der XML-Header enthält unter anderem die folgenden wichtigen Einträge:

```
<?xml version="1.0"?>
<?mso-application progid="Excel.Sheet"?>
<DocumentProperties xmlns="urn:schemas-microsoft-com:office:office">
<Author>PG484-3DAugtracer-2006-Uni-Dortmund</Author>
<LastAuthor>Everybody</LastAuthor>
<Created>2006-07-12T18:16:29Z</Created>
<Version>11.5606</Version>
</DocumentProperties>
<ExcelWorkbook xmlns="urn:schemas-microsoft-com:office:excel">
<WindowHeight>9210</WindowHeight>
<WindowWidth>15195</WindowWidth>
<WindowTopX>480</WindowTopX>
<WindowTopY>105</WindowTopY>
<ActiveSheet>0</ActiveSheet>
</ExcelWorkbook>
```

Zu den abgespeicherten Werten gehören

- die Projektparameter
- sämtliche Einstellungen, die beim Erstellen eines neuen Projektes angegeben werden können

- eine Liste mit allen verarbeiteten Bildern
- die berechneten Kennzahlen
- die Position und Größe aller segmentierter Tropfen
- die Ergebnisse der Frequenzanalyse

Aufgrund der großen Anzahl der Lamellenkoordinaten lassen sich diese nur als Plaintext ausgeben. Der Benutzer kann dabei wählen, ob alle Kanten in einer Datei gespeichert werden sollen, oder ob eine einzelne Datei pro Frame angelegt wird.

9.2.2 Video-Export

Es besteht die Möglichkeit, alle im Programm angezeigten Daten in einen Videostream zu exportieren. Der Benutzer wählt zuerst die Optionen aus, die er exportieren möchte, z. B. stereo oder mono, eingblendete Ausgleichsgeraden, Vektorfelder o. Ä. Dann wird das externe Tool *mencoder*[MP1] aufgerufen, das ein Video im avi-Format mit dem XviD-Codec[Xvi] erstellt.

Kapitel 10

Auswertung eines Versuchs

Inhalt

10.1 Einleitung	115
10.2 Bildverbesserung	116
10.2.1 Kalibrierung	116
10.2.2 Denoising/Dewarping	117
10.3 Segmentierung der Lamelle	117
10.4 Tropfenanalyse	117
10.5 Frequenzanalyse	117
10.6 Tracking	118
10.7 3D-Rekonstruktion	118
10.8 Fazit	118

10.1 Einleitung

In diesem Kapitel wird anhand eines Beispieldatensatzes exemplarisch eine Analyse und Bewertung der einzelnen Berechnungen des Programms **3D-AugTracer** vorgenommen. Der Datensatz behandelt ein Stereoprojekt mit jeweils 181 Bildern im Double-Shutter-Modus.

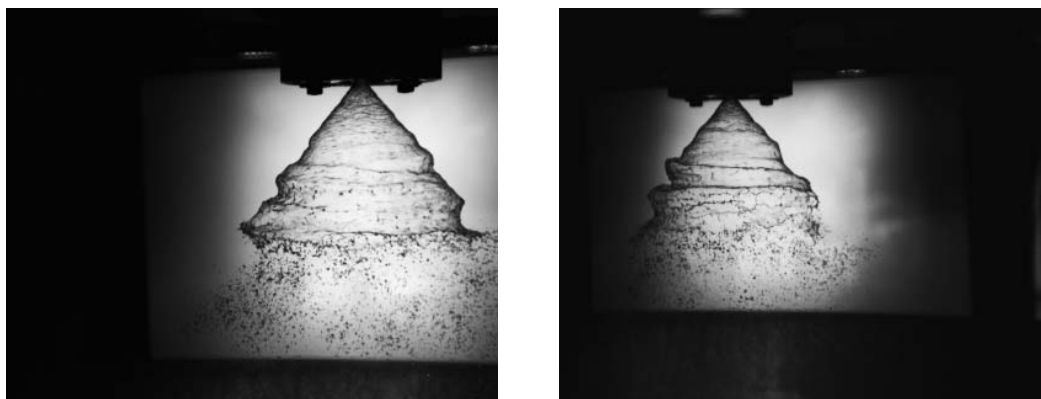


Abbildung 10.1: Darstellung des Kegels des Beispieldatensatzes: Das linke Bild entspricht der linken Kamerasicht, das rechte Bild entspricht der rechten Kamerasicht.

Die wichtigsten Versuchsparameter lauten:

- Flüssigkeit: Wasser
- Druck: 0.3 Bar
- Abstand Düse - Kamera: 138,5 cm
- Abstand linke - rechte Kamera: 7,3 cm

10.2 Bildverbesserung

10.2.1 Kalibrierung

Zunächst müssen beide Kameras kalibriert werden. Dazu wurde das entworfene Kalibrierungsprogramm mit einem vorgegebenem Kalibrieremuster verwendet. Es werden automatisch markante Eckpunkte gefunden, die für die weiteren Berechnungen nötig sind (siehe Abb. 10.2).

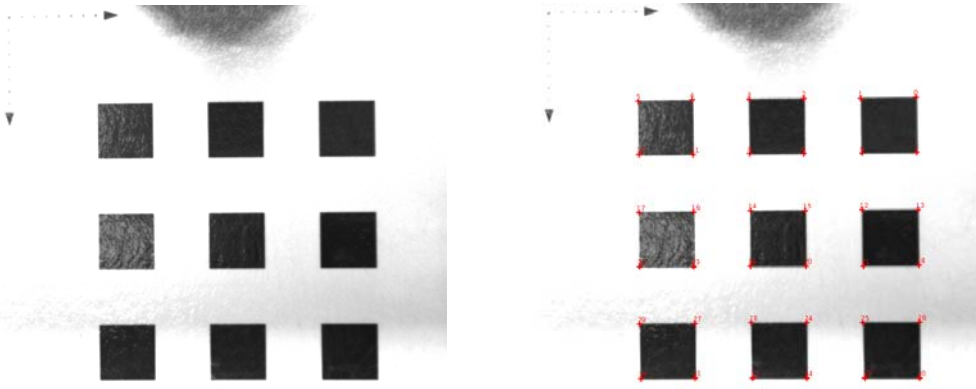


Abbildung 10.2: Kalibrieremuster (links) sowie gefundene markante Eckpunkte (rechts).

Neben den gefundenen Eckpunkten werden auch die Positionen sowie Rotationen der beiden Kameras zueinander berechnet - ausgedrückt in Form von Rotations- und Translationsmatrizen. Dabei wird davon ausgegangen, dass die linke Kamera im Ursprung des (imaginären) Koordinatensystems liegt. Das Kalibrierungsprogramm berechnet die Translation und Rotation der rechten Kamera. In den Versuchsreihen standen die Kameras parallel zueinander, so dass die Rotation nur minimal ist. Dies entspricht der Einheitsmatrix als Rotationsmatrix (siehe Abb. 10.3).

Verrechnete Parameter, Mittelwerte									
Kamera Links					Kamera Rechts				
Rotation			Translation		Rotation			Translation	
1	0	0	X	0	0.895688	0.217973	0.281716		688.13
0	1	0	Y	0	-0.242578	0.916987	-0.00303157		421.736
0	0	1	Z	0	-0.211265	-0.117882	0.9482		451.828

Abbildung 10.3: Kalibrierergebnis: Die Rotation entspricht annähernd der Einheitsmatrix (kaum Rotation vorhanden).

10.2.2 Denoising/Dewarping

Um eine optimale Berechnung der weiteren Parameter zu gewährleisten, müssen Fehler wie Rauschen oder Verzerrungen, die bei der Aufnahme durch die Kamera entstehen, entfernt werden. Dies geschieht über das Denoising (Entfernung des Rauschens) bzw. des Dewarping (Entzerren des aufgenommenen Bildes). Die weiteren Berechnungen werden auf den so verbesserten Bildern durchgeführt (siehe Abb. 10.4).



Abbildung 10.4: Denoising: (von li. nach re.) Original Aufnahme, Verbesserte Aufnahme (Rauschen entfernt), normalisiertes Differenzbild.

10.3 Segmentierung der Lamelle

Für fast alle Berechnungen ist es zunächst notwendig, die Lamelle zu segmentieren. Dabei wird der Rand des Sprühkegels verfolgt und die Ausgleichsgeraden werden berechnet (siehe Abb. 10.5). Bei dem verwendeten Datensatz wurde ein Threshold von 247 gewählt, der eine fast optimale Segmentierung liefert. Der errechnete über alle Bilder gemittelte Lamellenwinkel liegt bei $39,9^\circ$ für den linken Teilkegel sowie $38,1^\circ$ für der rechten Teilkegel. Dies entspricht einem Gesamtwinkel von 78° .

10.4 Tropfenanalyse

Es wurde nach Tropfen der Radiengröße 6-8 Pixel in einem eingeschränkten Bereich unterhalb des Lamellenabrisses gesucht. Eine Verteilung der gefundenen Tropfen ist in Abbildung 10.6 angegeben. Dadurch dass die Tiefeninformationen in den Bildern verloren gehen und sich Tropfen überlagern, werden weiter entfernte Tropfen bzw. im Hintergrund liegende Tropfen nicht immer erkannt.

10.5 Frequenzanalyse

Für die Analyse der Oberflächenwellen-Frequenzen wird zunächst die Darstellung der Amplitudenfunktion dargestellt (Abb. 10.7). Im Anschluss folgen die Darstellungen der Fourier-Transformierten des Signals und einer Wavelet-Analyse mit dem Mexican-Hat-Wavelet bei Skalierung $j = 3$ (Abb. 10.8 und 10.9). Weiterhin werden exemplarische Spektrogramme der Frequenzverteilungen und der Wavelet-Koeffizienten abgebildet (Abb. 10.10 und 10.11). Abgeschlossen wird die Analyse mit den Darstellungen der kumulierten Amplituden und der kumulierten Frequenzen (Abb. 10.12 und 10.13).

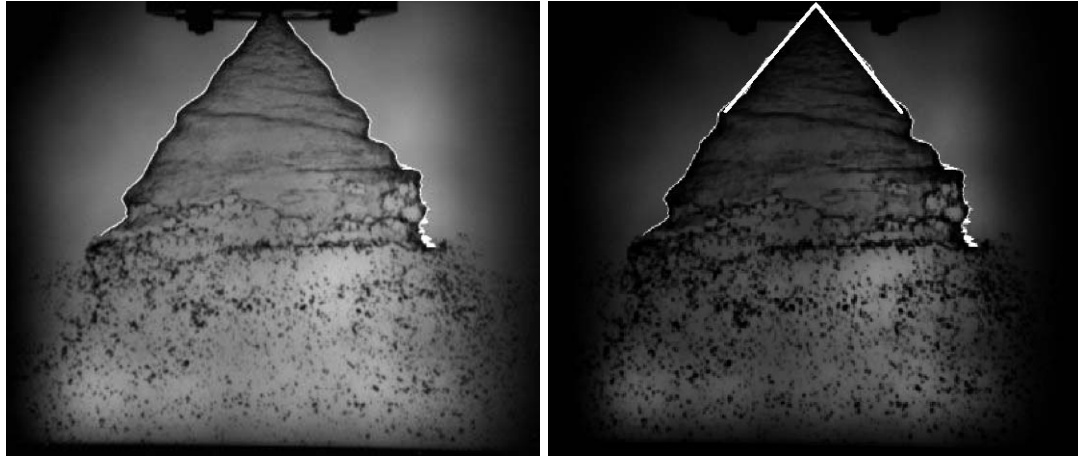


Abbildung 10.5: Darstellung der segmentierten Lamelle und den Ausgleichsgeraden

10.6 Tracking

Von den drei wählbaren Tracking-Verfahren liefert das Block-Matchin-Tracking die besten Ergebnisse. Bei diesem Versuch wurden folgende Parameter eingestellt:

- Templategröße 10
- Suchgröße 25
- Gitterabstand 30
- minimale Korrespondenz 0,0

Das dadurch erzeugte Vektorfeld zeigt Abbildung 10.14. Anhand der Stromlinien lässt sich die Bewegung der Lamelle verfolgen (siehe Abb. 10.15), nach Doppelklick auf einen Bereich wird die entsprechende Geschwindigkeit angegeben.

10.7 3D-Rekonstruktion

Anhand von in beiden Bildern bekannten Punkten wie den gefundenen Tropfen kann eine Rekonstruktion des entsprechenden 3D-Punktes erstellt werden. Diese Berechnung wird nur auf die Tropfen angewandt. Eine entsprechende Visualisierung der 3D-Tropfen bei dem verwendeten Beispieldatensatz ist in Abbildung 10.16 zu sehen.

10.8 Fazit

Um eine hohe Genauigkeit der errechneten Daten zu gewährleisten, sollte ein möglichst großer Datensatz verwendet werden. Vor allem bei der Berechnung von Vektorfeldern sind die Ergebnisse bei einem kleinen Datensatz nicht sehr aussagekräftig.

Weiterhin sind die Veränderungen der Lichtverhältnisse und die unterschiedlichen Einstellungen der Kameras bezüglich Helligkeit und Zoom ein Problem, das insbesondere in der Bildbearbeitung zu Berechnungsfehlern führen kann. Um dies zu kompensieren, besteht die Möglichkeit, für die Kameras getrennte Thresholds bei der Segmentierung einzugeben. Durch die verschiedenen Größeneinstellungen der Kameras ist es jedoch nur schwer möglich, bei der späteren 3D-Projektion eine optimale Visualisierung des 3D-Kegels zu erlangen.

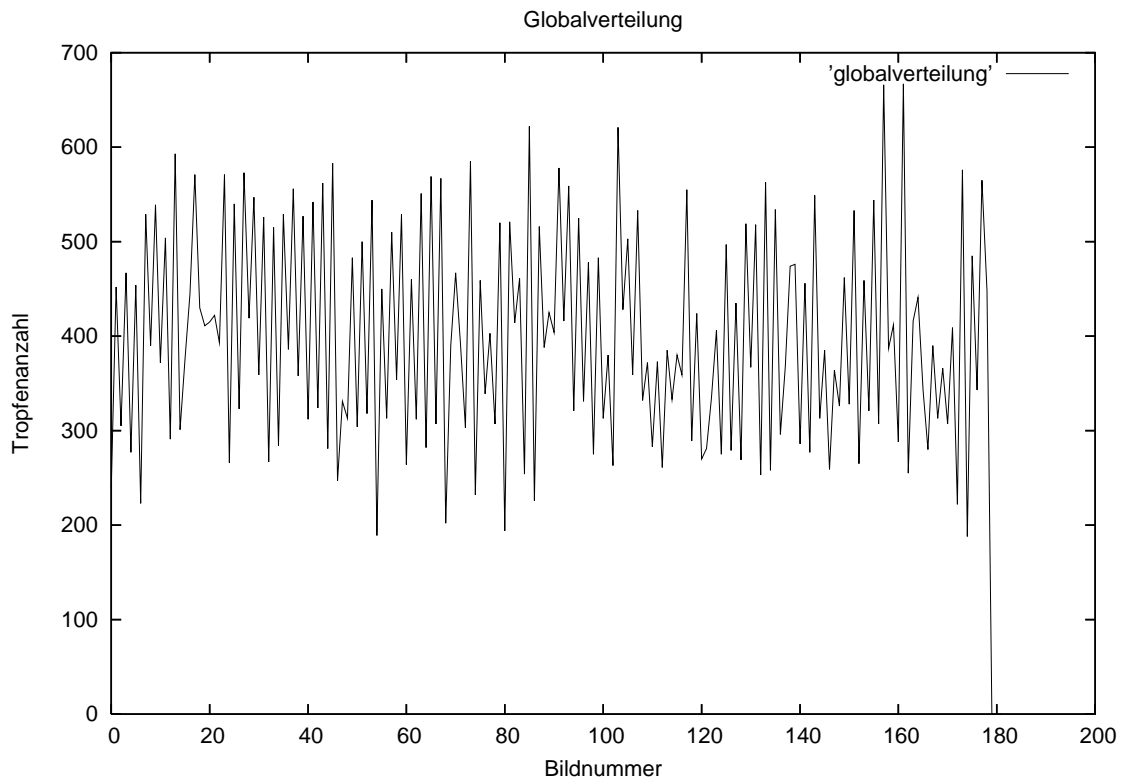


Abbildung 10.6: Globale Tropfenverteilung

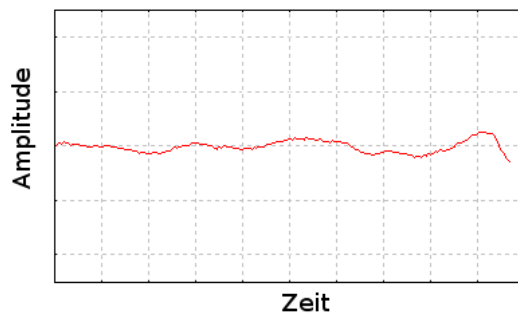


Abbildung 10.7: Amplitudenfunktion

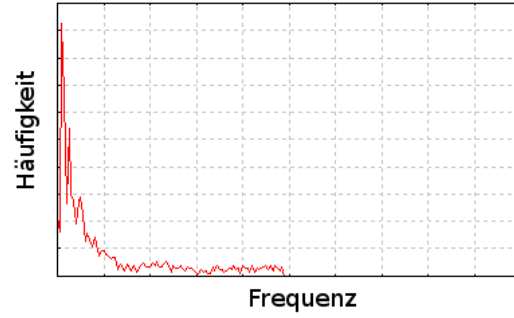


Abbildung 10.8: Fourier-Analyse

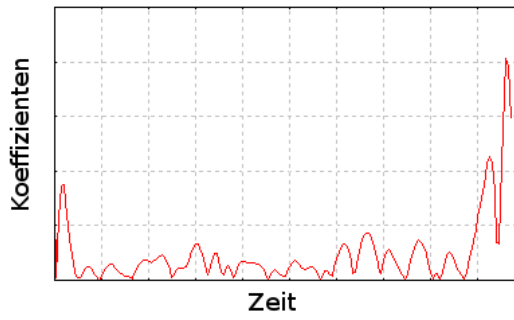


Abbildung 10.9: Waveletanalyse

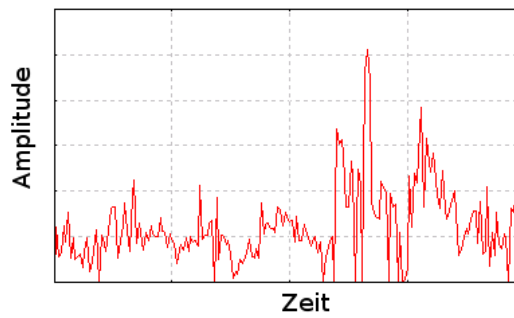


Abbildung 10.10: kumulierte Amplituden

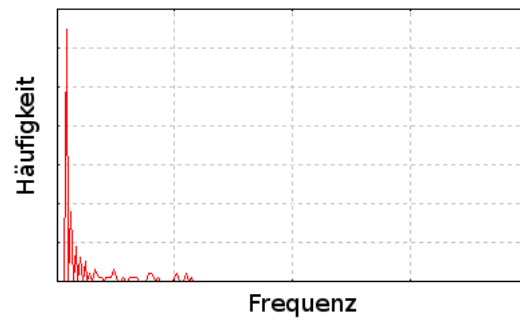


Abbildung 10.11: kumulierte Frequenzen

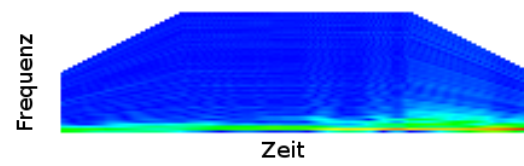


Abbildung 10.12: Frequenzspektrogramm

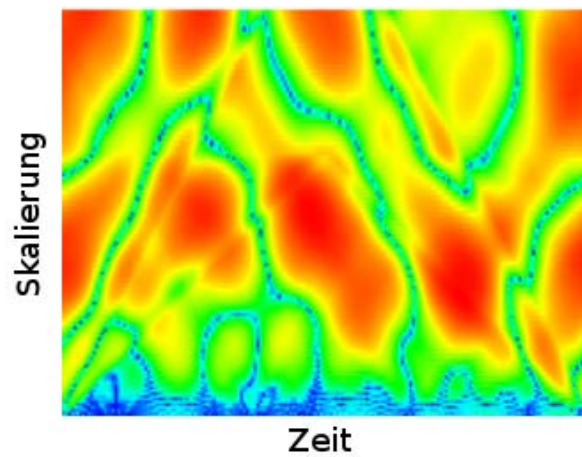


Abbildung 10.13: Waveletspektrogramm

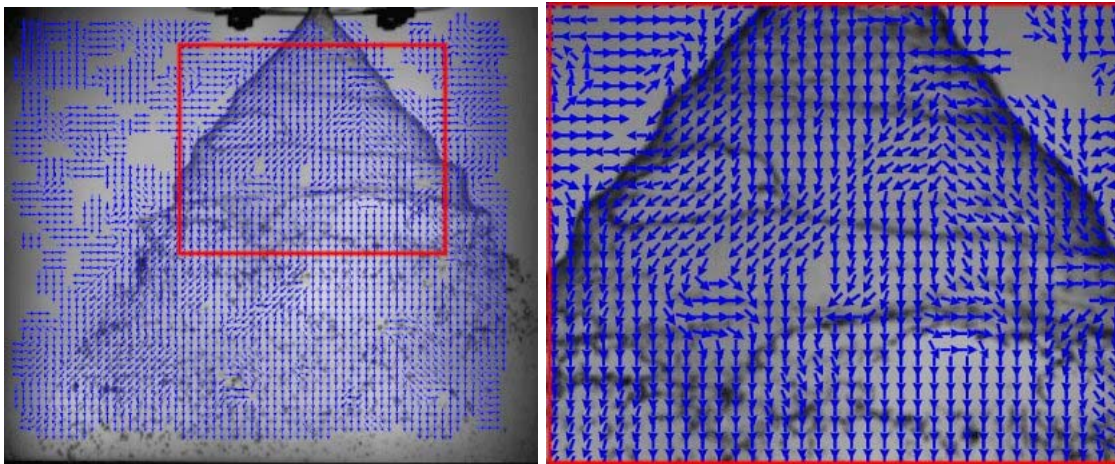


Abbildung 10.14: Block-Matching Vektor-Feld

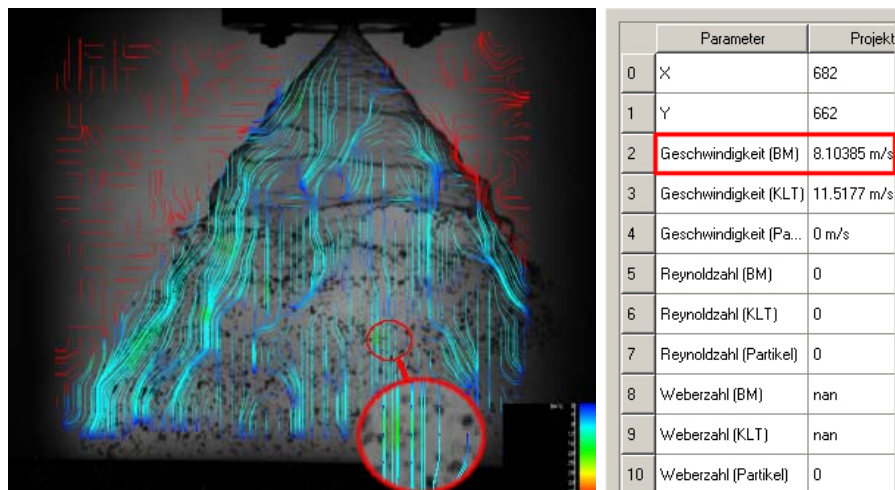


Abbildung 10.15: Block-Matching Vektor-Stromlinien (links) sowie die ausgewählte Geschwindigkeit (rechts)

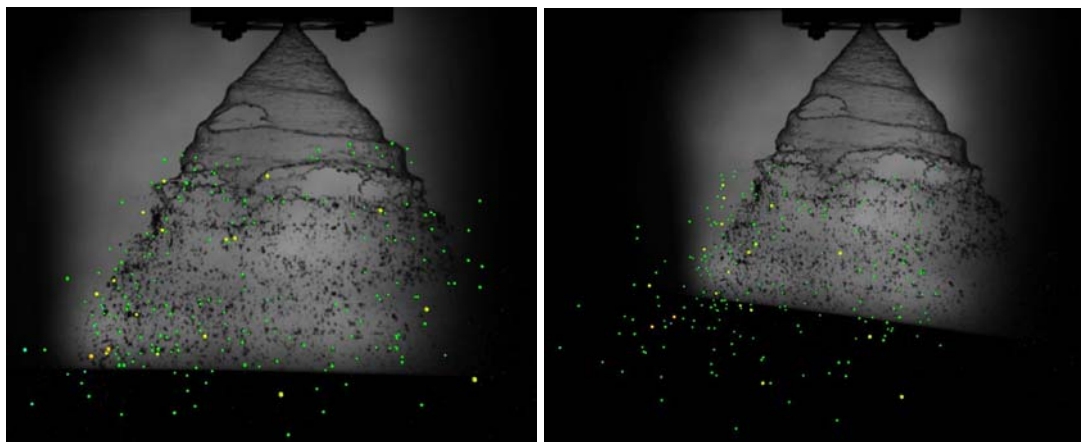


Abbildung 10.16: Darstellung der 3D-Tropfen nach der Rekonstruktion: Frontalansicht (links), perspektivische Ansicht (rechts)

Kapitel 11

Zusammenfassung und Ausblick

Inhalt

11.1 Projektresumé	123
11.2 Ausblick	123

Insgesamt wurde in dieser Projektgruppe ein System entworfen, welches in der Lage ist, die aufgenommenen Bilder bzw. Videos von Sprühkegeln einer Hohlkegeldüse mittels Segmentierungs- und Trackingverfahren zu bearbeiten. Dem voraus geht eine generelle Verbesserung der Bilddaten mittels Rauschreduzierung und Entzerrung sowie eine Kalibrierung der Kameras. Die durch die Segmentierung und das Tracking gewonnenen Daten können letztendlich mittels OpenGL visualisiert werden.

Es stellte sich in der Entwicklungszeit von **3D-AugTracer** heraus, dass nicht alle Verfahren, an denen Projektteilnehmer gearbeitet haben, die gewünschten Ergebnisse erzielen konnten. So war eine 3D-Visualisierung der gewonnenen Daten nicht zufriedenstellend, und auch Löcher im Sprühkegel konnten mittels Haralick-Segmentierung nicht mit Sicherheit erkannt werden.

11.1 Projektresumé

Es darf nicht unerwähnt bleiben, dass jedes Mitglied der Projektgruppe bei der Arbeit an dem Programm wertvolle Kenntnisse und Erfahrungen im Bereich der Bildverarbeitung, des Software-designs und der Projektarbeit in einem größeren Team gewonnen hat. Diese Erfahrungen werden in hohem Maße zur Erfüllung der Anforderungen beitragen, die bei der Bearbeitung der Diplomarbeit und in der Berufspraxis an Informatiker gestellt werden.

11.2 Ausblick

Im Anschluss an diese Projektgruppe 484 - **3D-AugTracer** beginnt eine neue Projektgruppe (PG 498 - **3D-LaserNAV**) zur Laserpointer-gestützten multimodalen Interaktion in einer dreidimensionalen stereographischen Realität. Der Prozess der Entwicklung von der Nutzung des Laserpointers zur manipulierten, erweiterten virtuellen 3D-Realität lässt sich in die gleichen Phasen aufteilen, die auch die Aufgabe dieser Projektgruppe prägte. So werden auch in der PG 498 nach anfänglicher Aufbereitung der Daten mittels digitaler Bildverbesserung (Rauschreduzierung, Entzerren, etc.) Segmentierungs- und Trackingverfahren eingesetzt werden. Die Ergebnisse sollen anschließend 2- und 3-dimensional visualisiert werden.

Anhang A

Handbuch

Inhalt

A.1 Installation	125
A.1.1 Qt-Installation	125
A.1.2 MySQL-Server-Installation	125
A.1.3 3D-AugTracer-Installation	126
A.2 Kalibrierung	126
A.3 3D-AugTracer	129
A.3.1 Neues Projekt	129
A.3.2 Projekt laden	135
A.3.3 Anzeigen im Hauptfenster	136
A.3.4 Menü	137

A.1 Installation

Um das **3D-AugTracer**-Programm auf einem beliebigen Computer mit Windows-Betriebssystem zu installieren, werden folgende zusätzliche Komponenten benötigt

1. Qt
2. MySQL-Server

A.1.1 Qt-Installation

Auf der Homepage des Herstellers Trolltech lässt sich die aktuelle Qt-Version kostenlos beziehen <http://www.trolltech.com/developer/downloads/qt/windows>. Zum Zeitpunkt der Erstellung dieses Handbuchs ist die Version 4.1.4 aktuell, jedoch wird die Installation einer neueren Version empfohlen, sobald diese veröffentlicht wird.

Während der Installation muss die Option „Set Environment Variables“ aktiviert werden. Zudem ist es empfehlenswert, die Zuweisung der .ui - Dateierweiterung zu Qt zu aktivieren.

A.1.2 MySQL-Server-Installation

Unter <http://dev.mysql.com/downloads/mysql/5.0.html> kann durch Auswahl des entsprechenden Betriebssystems die MySQL 5.0 Community Edition bezogen werden. Alle Programmtests wurden mit dieser Version ausgeführt, so dass wenn möglich nur diese Version verwandt werden sollte.

Zusätzlich wird empfohlen, das Paket Gui-Tools¹ zu installieren, um den enthaltenen MySQL Administrator für die komfortable Datenbank-Administration nutzen zu können.

A.1.3 3D-AugTracer-Installation

Nachdem die nötigen Programme sich nun auf der Festplatte befinden, kann der komplette lib-Ordner von der Installations-CD zusammen mit der Datei augtracer.exe an eine beliebige Stelle kopiert werden. Der lib-Ordner beinhaltet alle für das Programm nötigen Bibliotheken. Nach einem Neustart wird **3D-AugTracer** durch Doppelklick auf augtracer.exe gestartet.

A.2 Kalibrierung

Um für den Sprühkegel eine 3D-Rekonstruktion durchführen zu können, müssen die Koordinaten einzelner Punkte in den aufgenommenen Bildern bekannt sein. Dazu müssen die Kameras aufeinander abgestimmt, also kalibriert werden. Dies wird vor dem Erstellen eines neuen Projekts ausgeführt. Auch für die Berechnung der Metrik werden die Daten der Kalibrierung benötigt. Hierfür muss man ein Referenzgitter aufnehmen, dessen Punkte bekannt sind und in einer Referenzdatei des Formats .cmod gespeichert wurden. Ein entsprechendes Gitter kann über das Kalibrierungsprogramm erstellt und ausgedruckt werden.

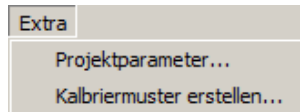


Abbildung A.1: Menüpunkt Extra

Für optimale Ergebnisse werden Aufnahmen dieses Gitters in mehreren Orientierungen (mindestens 3) von beiden Kameras benötigt. **Das erste dieser Bilder muss eine Aufnahme des gerade unter die Düse gehaltenen Referenzgitters sein, da dies für die weitere Berechnung im Projekt 3D-AugTracer nötig ist.**

Wenn ein neues Projekt angelegt wird, besteht die Möglichkeit die einzelnen Parameter mit anzugeben, wie beispielsweise die verwendete Flüssigkeit wie Wasser oder Glycerin (siehe Abb. A.2). Diese Parameter werden mit den gespeicherten Ergebnissen exportiert, können jedoch auch später noch im Programm **3D-AugTracer** verändert werden. Um nun die Kalibrierung durchzuführen, müssen zunächst die Bilder mit dem aufgenommenen Kalibrierungsgitter geladen werden. Dabei ist darauf zu achten, dass die Bildpaare links und rechts in der gleichen Reihenfolge geladen werden. Über den Menüpunkt „Modus“ werden die einzelnen folgenden Schritte ausgewählt:

1. Eckpunkte setzen

Als nächstes müssen die Grenzen des Gitters definiert werden. Klicken Sie dazu in jedem Bild die Ecken eines imaginären Vierecks an, das das Gitter umgibt. Dabei wird mit jedem Klick eine weitere Kante des Vierecks angezeigt (siehe Abb. A.4). Ist dieses vollständig, springt das Programm automatisch zum nächsten Bild der Serie weiter und Sie können den Vorgang wiederholen. Sind alle Bilder einer Kamera bearbeitet, muss derselbe Schritt für die andere Kamera ebenfalls durchgeführt werden.

2. Gitterpunkterkennung

Durch Klick auf diesen Menüpunkt sucht das Programm automatisch nach den Eckpunkten der Quadrate im Referenzgitter. Diese werden durch kleine Kreuze dargestellt. Weiterhin wird die Anzahl der gefundenen Punkte in jedem Bild angezeigt und es ist darauf zu achten,

¹<http://dev.mysql.com/downloads/gui-tools/5.0.html>

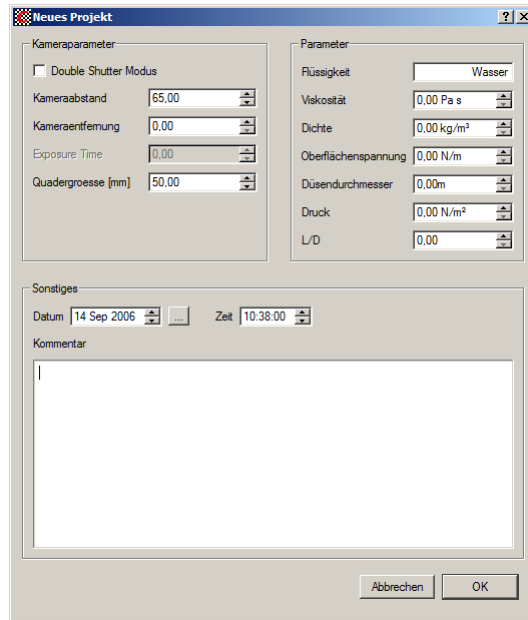


Abbildung A.2: Projektparameter eines neuen Projektes angeben

Modus	Extra
1. Eckpunkte Setzen	Ctrl+1
2. Gitterpunkterkennung	Ctrl+2
3. Sortieren	Ctrl+3
4. Referenz-Modell Laden	Ctrl+4
5. Kalibrieren	Ctrl+5
Ergebnisse Ansehen	Ctrl+6

Abbildung A.3: Menüpunkte Modus

dass die Zahl der gefundenen Punkte mit der Zahl der angegebenen Punkte im Referenzgitter übereinstimmt. Falsch gefundene Punkte können mit einem Rechts-Klick gelöscht, neue mit einem Links-Klick hinzugefügt werden. Um dies zu vereinfachen, können die Kreuze, die die Punkte markieren, durch Auswählen des Punktes „Eckpunkte dicker“ vergrößert und über „Eckpunkte beschriften“ nummeriert werden (siehe Abb. A.5). Gehen Sie alle Bilder der linken und rechten Kamera durch und vergewissern Sie sich, dass die Eckpunkte korrekt erkannt wurden.

3. Sortieren

Um die Referenzpunkte mit den ermittelten Punkten zu vergleichen, werden diese durchnummeriert. Die Nummerierung startet in der linken oberen Ecke und wird horizontal weitergeführt. Sollte noch ein weiterer falsch gesetzter Punkt gefunden und korrigiert werden, kann die Sortierung jederzeit wiederholt werden.

4. Referenz-Modell laden

Sind alle Eckpunkte definiert und nummeriert, kann das entsprechende Referenzmodell geladen werden. Es erscheint ein Dateidialog, in dem eine Datei des Formats .comod ausgewählt werden kann.

5. Kalibrieren

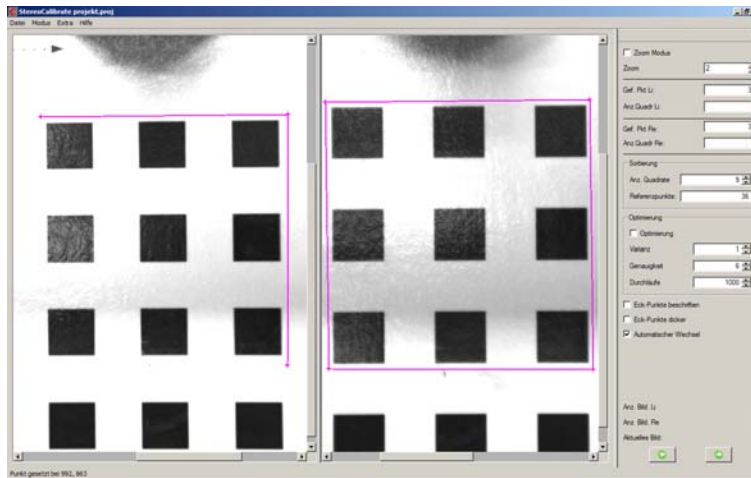


Abbildung A.4: Eckpunkte des Referenzgitters bestimmen

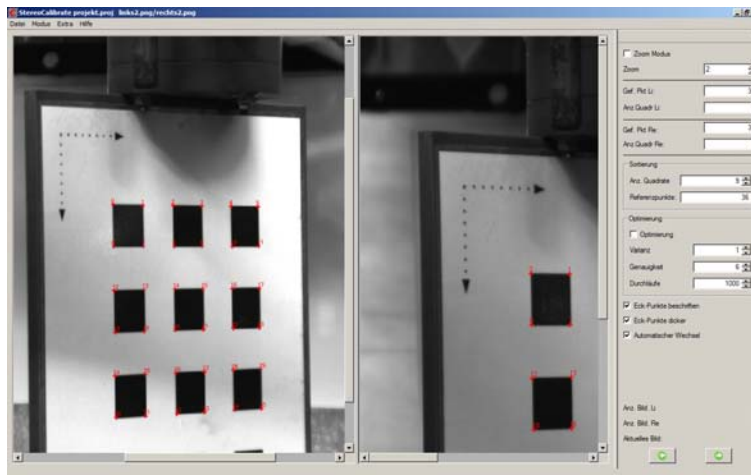


Abbildung A.5: Gitterpunkte setzen

Die Kalibrierung kann nun durchgeführt werden. Dieser Prozess dauert einen Moment. Ist er erfolgreich abgeschlossen, können Sie sich die Ergebnisse anzeigen lassen (siehe Abb. A.7). Möchten Sie zu den Bildern zurückkehren, entfernen Sie den Haken bei dem Menüpunkt „Ergebnisse anzeigen“.

Sie haben die Möglichkeit, geladene Bilder und eventuell bereits gefundene Punkte oder Ergebnisse als Projekt abzuspeichern. Ein entsprechendes Projekt kann natürlich auch geladen werden. Das Speichern oder Exportieren der Ergebnisse ist notwendig für das Erstellen eines neuen Projektes in **3D-AugTracer**. Hierfür wird eine XML-Datei mit den ermittelten Daten benötigt.

In der Leiste rechts neben den angezeigten Bildern stehen weitere Optionen zur Verfügung (siehe Abb. A.8). Unter anderem wird hier die Anzahl der gefundenen Punkte und der Referenzpunkte angezeigt, die in jedem Bild übereinstimmen müssen. Weiterhin kann man eine Optimierung auswählen, die während der Kalibrierung durchgeführt werden soll. Hier können Varianz, Genauigkeit und die Anzahl der Durchläufe beeinflusst werden. Je höher Genauigkeit und Durchläufe gewählt werden und je geringer die Varianz eingestellt wird, desto genauer ist das Gesamtergebnis. Dafür wird die Berechnung der Kalibrierdaten längere Zeit in Anspruch nehmen. Die Standardwerte von 1, 6 und 1000 sind ein guter Kompromiss zwischen benötigter Berechnungszeit und erhaltener

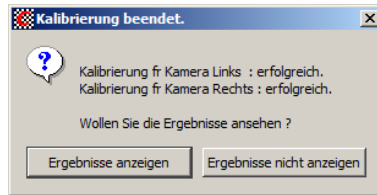


Abbildung A.6: Erfolgreiche Kalibrierung

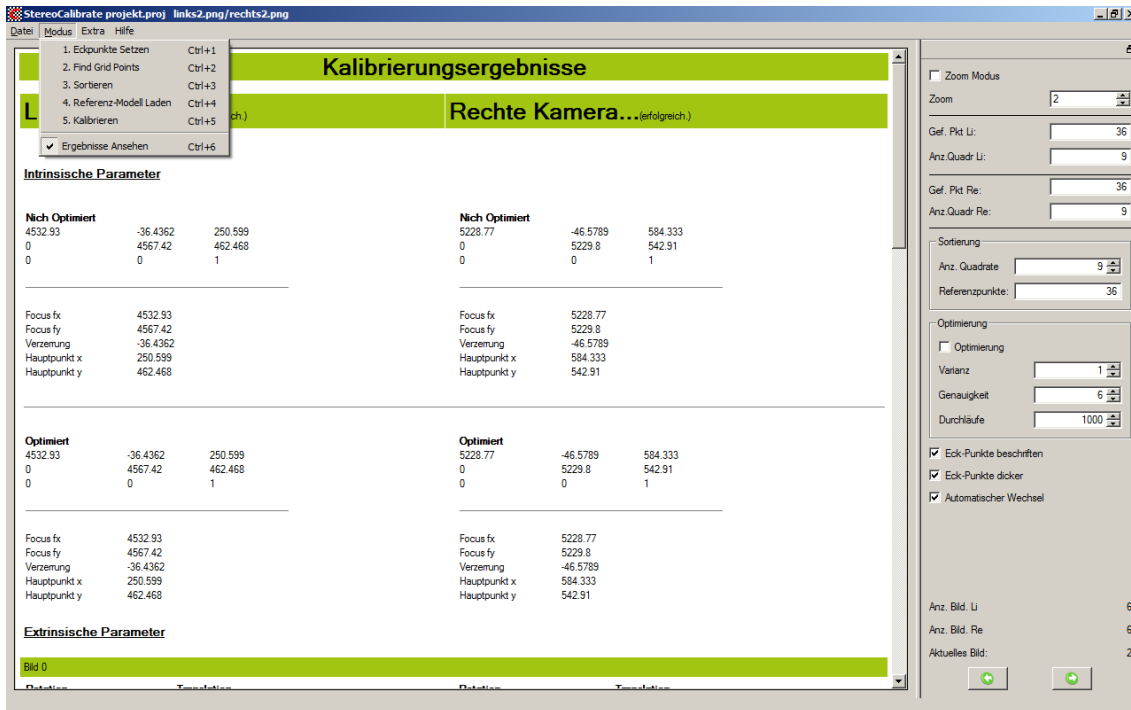


Abbildung A.7: Kalibrierungsergebnisse anzeigen

Genauigkeit.

A.3 3D-AugTracer

3D-AugTracer ist das Hauptprogramm, in dem alle Berechnungen durchgeführt werden und die Ergebnisse angezeigt werden können. Die folgenden Abschnitte sind in der Reihenfolge aufgeführt, wie sie typischerweise nach der Installation genutzt werden oder möglich sind.

A.3.1 Neues Projekt

Bevor Sie in dem Programm **3D-AugTracer** ein neues Projekt anlegen können, müssen Sie zunächst die Kalibrierung der gewünschten Bildserie vornehmen (vgl. Kapitel A.2). Zum Erstellen klicken Sie auf den Button „Neues Projekt“ oder wählen Sie aus dem Menüpunkt „Datei“ den entsprechenden Eintrag aus. Dadurch öffnet sich der Assistent, der Sie durch die einzelnen Schritte führt.

Geben Sie zuerst einen Namen für Ihr neues Projekt ein und bestimmen Sie den Ort, an dem

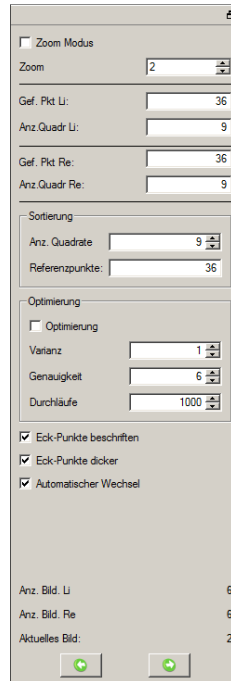


Abbildung A.8: Optionen für die Kalibrierung

es gespeichert werden soll (siehe Abb. A.9)². Nach einem Klick auf „Weiter“ öffnet sich die Seite Projektparameter. Es wird automatisch die Kalibrierungsdatei geladen, falls sie sich in dem zum Speichern des Projekts angegebenen Ordner befindet. Ansonsten müssen Sie diese manuell laden. Klicken Sie dazu auf den Button „Durchsuchen“ und wählen die entsprechende Datei aus. Es werden alle gespeicherten Versuchsparameter angezeigt. Sie haben die Möglichkeit diese zu bearbeiten und einen Kommentar zu dem Projekt hinzuzufügen. Klicken Sie anschließend auf „Weiter“ (siehe Abb. A.10).

Sie gelangen nun zu der Seite „Dateiauswahl“ (siehe Abb. A.11). Hier wählen Sie zunächst aus, ob Sie ein Mono- oder Stereo-Projekt erstellen möchten. Danach muss ein beliebiges Bild aus der Serie ausgewählt werden. **3D-AugTracer** fügt automatisch alle Bilder im selben Ordner zu der Serie hinzu. Haben Sie ein Stereo-Projekt gewählt, müssen die Bilder der rechten und der linken Kamera selektiert werden. Über den Button „Vorschau“ lassen sich alle Bilder anzeigen und Sie können durch einen Klick auf das gewünschte Bild nur bestimmte für das neue Projekt auswählen. Achten Sie darauf, dass Sie (im Stereomodus) für linke und rechte Kamera die sich entsprechenden Bilder markieren.

Durch einen Klick auf „Weiter“ gelangen Sie zu der Seite mit der Auswahl der durchzuführenden Berechnungen (siehe Abb. A.12). Es stehen folgende Möglichkeiten zur Verfügung:

- Videoerstellung
- Lamellensegmentierung
- Tracking
- Tropfenanalyse

Nur im Stereomodus:

²**3D-AugTracer** benötigt zum Speichern aller berechneten Daten zusätzlich den Zugriff auf eine MySQL-Datenbank, den Sie in einem der folgenden Schritte konfigurieren

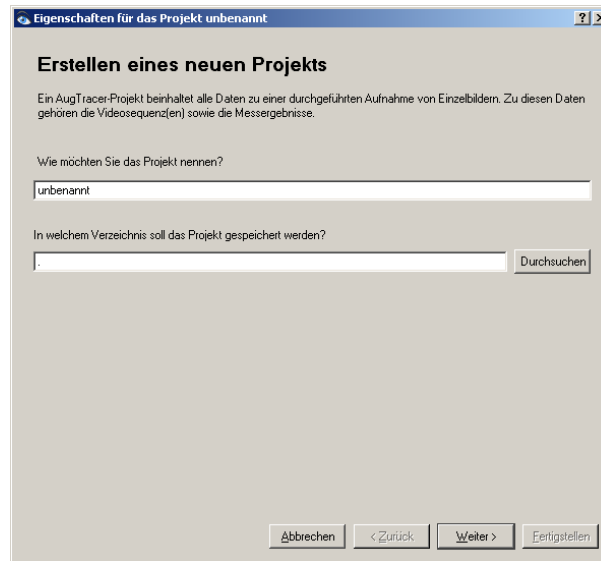


Abbildung A.9: Neues Projekt - Assistent - Titelseite

- 3D Tropfenrekonstruktion
- 3D Vektorfeldrekonstruktion

Auf den weiteren Seiten werden Standardwerte die in jedem Projekt benötigt werden und die Optionen der einzelnen ausgewählten Berechnungsschritte eingestellt. Im Folgenden werden alle Seiten beschrieben:

Datenbankanbindung

Wählen Sie hier Ihren Datenbankserver und den verwendeten Port aus, geben Sie außerdem Benutzernamen und Ihr Passwort ein. Der Server dient der Sicherung all Ihrer Berechnungen. Sie können die angegebene Verbindung durch einen Klick auf den Button „Verbindung testen“ überprüfen (siehe Abb. A.13).

Denoising

Über das Denoising wird das Rauschen in den einzelnen Bildern reduziert. Es besteht die Möglichkeit einen weichen oder harten Thresholdtyp auszuwählen. Beide Typen wählen als Threshold den oben eingestellten Wert, der von 5 bis 30 variieren kann. Weiterhin kann man die verwendete Basisfunktion einstellen sowie die Transformationstiefe (siehe Abb. A.14). Alle gewählten Einstellungen lassen sich über den Button „Vorschau“ anzeigen.

Angaben zum Video

Zunächst können Sie zwischen den beiden Formaten XviD und Dirac wählen (siehe Abb. A.15). Möchten Sie weitere Funktionen einstellen, klicken Sie auf die „Experteneinstellungen“, die im nächsten Schritt angezeigt werden.

Markerbestimmung

Der Marker kennzeichnet den ungefähren Mittelpunkt der Düse und wird für die weitere Berechnung benötigt (siehe Abb. A.16). Wählen Sie die automatische Markerfindung oder die manuelle Suche aus. Für letzteres wählen Sie über die Button „vorheriges Bild“ und „nächstes Bild“ einen

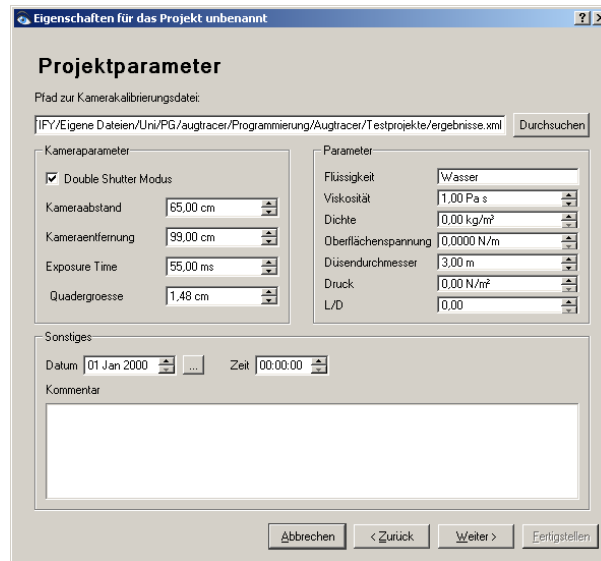


Abbildung A.10: Neues Projekt - Assistent - Projektparameter

Frame aus in dem der Marker deutlich zu erkennen ist und markieren Sie diesen mit einem Klick auf die linke Maustaste. Die automatische Suche findet den Marker nach einigen Momenten selbstständig. Sollten Sie ein Stereoprojekt erstellen, wird diese Seite für die rechte und die linke Kamera getrennt angezeigt.

Lamellensegmentierung

Der Aufbau dieser Seite dauert einen Moment, da der Threshold automatisch ermittelt wird. Das Ergebnis wird rechts unter *Vorschau* angezeigt. Sind Sie mit dem gefundenen Wert nicht zufrieden, können Sie diesen manuell verändern. Über den Button „Vorschau“ können Sie sich für die eingegebene Anzahl von Frames die segmentierten Bilder ansehen. Auch diese Seite erscheint bei Auswahl eines Stereo-Projektes zweimal, jeweils für die linke und rechte Kamera (siehe Abb. A.17).

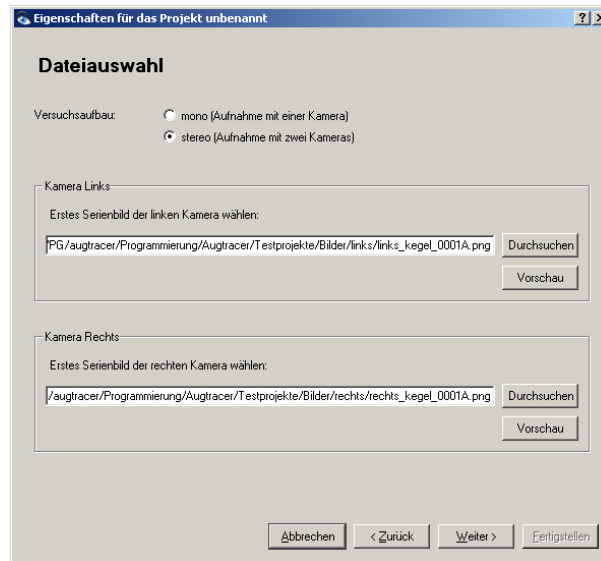


Abbildung A.11: Neues Projekt - Assistent - Dateiauswahl

Tracking

Das Tracking soll Strukturen im Bild über mehrere Frames verfolgen und daraus die Bewegung der Flüssigkeit bestimmen. Wählen Sie zunächst aus, welche Verfahren Sie verwenden möchten. Im unteren Bereich können Sie die Parameter der Verfahren einstellen.

- **Blockmatching** (siehe Abb. A.18)
 - Templategröße
Die Templategröße gibt die Seitenlänge des ausgeschnittenen Quadrats im ersten Bild fest, das im Suchfenster des zweiten Bildes wiedererkannt werden soll.
 - Suchgröße
Die Suchgröße legt die Seitenlänge des quadratischen Suchfensters im zweiten Bild fest.
 - Gitterabstand
Template und Suchfenster werden in regelmäßigen Abständen aus den Bildern ausgeschnitten. Der Gitterabstand legt die Distanz zwischen den Quadratmittelpunkten fest.
 - minimale Korrespondenz
Gibt an, wie gut die Übereinstimmung des Templates mit dem Fenster im nächsten Bild sein muss, damit der Verschiebevektor übernommen wird.
- **KLT-Tracking** (siehe Abb. A.19)
 - Fenstergröße
Das zu untersuchende Bild wird in Fenster gleicher Größe aufgeteilt. Die Fenstergröße legt die Seitenlänge des Fensterquadrates fest.
 - Features pro Fenster
Die Feature-Anzahl gibt an, wie viele Merkmale pro Fenster wiedererkannt werden sollen.
- **Partikeltracking**
 - Skalierungsfaktor
Der Skalierungsfaktor gibt die Rekursionstiefe für die Bildunterteilung an. Mit steigender Tiefe reduziert sich die Rechenzeit, allerdings sinkt auch die Genauigkeit.

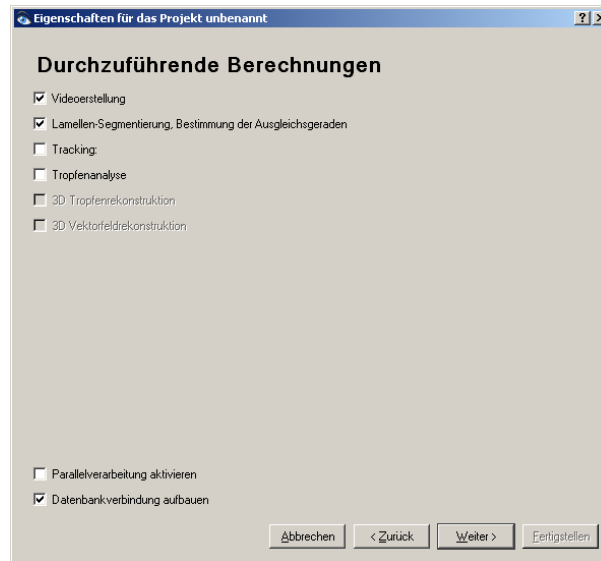


Abbildung A.12: Neues Projekt - Assistent - Auswahl der durchzuführenden Berechnungen

- Dimension für den Medianfilter
Der Medianfilter wird zur Glättung von Zwischenergebnissen verwendet. Größere Fenster erzeugen bessere Ergebnisse, allerdings auf Kosten der Rechenzeit.
- Korrelationsschwellwert
Definiert den Schwellwert für die normierte Korrelation zur Klassifikation von Pfadpunkten.

Sie können den Bereich in dem das Tracking ausgeführt werden soll bestimmen, indem Sie die entsprechende Option aktivieren und dann den Button „Festlegen“ betätigen. Über die Schieber können Sie mit der Maus den Bereich festlegen (siehe Abb. A.20).

Tropfenanalyse

Die Tropfenanalyse soll alle Tropfen im Bild erkennen. Sie können drei Genauigkeitsstufen auswählen, wie nach den Tropfen gesucht wird. Dabei enthalten die höheren Stufen die Berechnungen der niedrigeren (siehe Abb. A.21).

- Stufe 1
Es wird ein Template-Matching-Verfahren genutzt, das eine Helligkeitsanpassung vornimmt.
- Stufe 2
Die Verzerrungen von Tropfen und eine Formfaktorberechnung werden berücksichtigt.
- Stufe 3
Es wird eine rotationsunabhängige Suche vorgenommen.

Weiterhin können Sie einen Bereich wählen, in dem der Tropfenradius der Templates variiert wird. Ähnlich wie beim Tracking können Sie den Suchbereich des Bildes einschränken (siehe Abb. A.20).

3D Rekonstruktion

Wenn ein Stereo-Projekt erstellt wird, besteht die Möglichkeit sich die ermittelten Daten wie Tropfen und Vektoren dreidimensional anzeigen zu lassen. Dazu müssen die Daten der linken und rechten Bildserie miteinander verrechnet werden (siehe Abb. A.22).

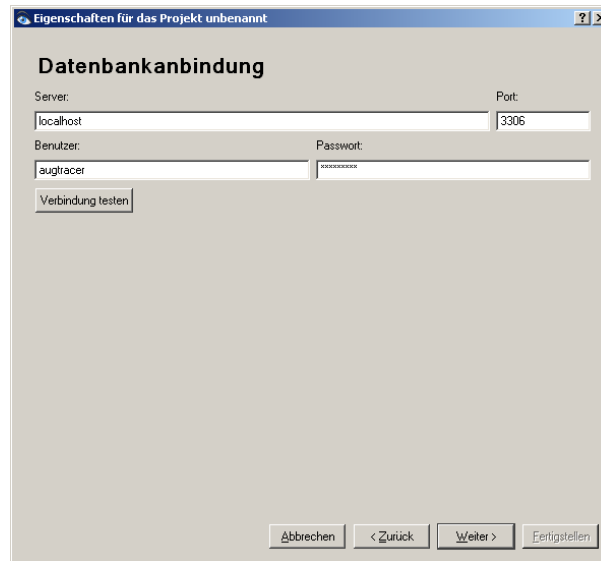


Abbildung A.13: Neues Projekt - Assistent - Datenbankanbindung

- Delta Abstand
Zur Berechnung werden aus den Kalibrierungsaufnahmen Epipolar-Linien ermittelt (vgl. Kapitel 6). Diese werden anhand von zwei korrespondierenden Punkten im linken und rechten Bild überprüft. Liegt die Epipolarlinie des linken Punktes im rechten Bild weiter als der Delta-Abstand vom rechten Punkt entfernt, so wird diese für alle weiteren Berechnungen verworfen.
- Suchfenstergröße
Dieser Wert beschreibt die Größe des Fensters, in dem auf den epipolaren Geraden nach Korrespondenzen gesucht wird.
- Korrelationsfenstergröße
Das Korrelationsfenster wird im Suchfenster bewegt und es werden zur Korrespondenzbestimmung die Intensitäten in diesem Fensterbereich verglichen. Bei verrauschten Bildern ist ein kleinerer Wert von Vorteil.

Beginn der Berechnungen

Haben Sie alle Eingaben getätigt, klicken Sie auf „Weiter“ um die Berechnungen zu starten. Sollten Sie noch etwas ändern wollen, können Sie über den Button „Zurück“ zu den vorherigen Seiten zurückkehren.

Berechnungen werden durchgeführt

Hier sehen Sie im oberen Bereich die Ausgaben der aktuell durchgeführten Verfahren. Über den Berechnungsfortschritt informieren Sie die Statusbalken am Fuß der Seite. Nach erfolgreich abgeschlossener Berechnung klicken Sie auf „Fertigstellen“ und Sie gelangen zurück in das Hauptfenster, indem nun das erste Bild der Serie angezeigt wird (siehe Abb. A.23).

A.3.2 Projekt laden

Um ein bereits erstelltes Projekt zu laden, wählen Sie den Menüpunkt „Projekt öffnen“ oder klicken Sie auf den entsprechenden Button in der Toolleiste. Es erscheint ein Dialog, in dem Sie das gespei-

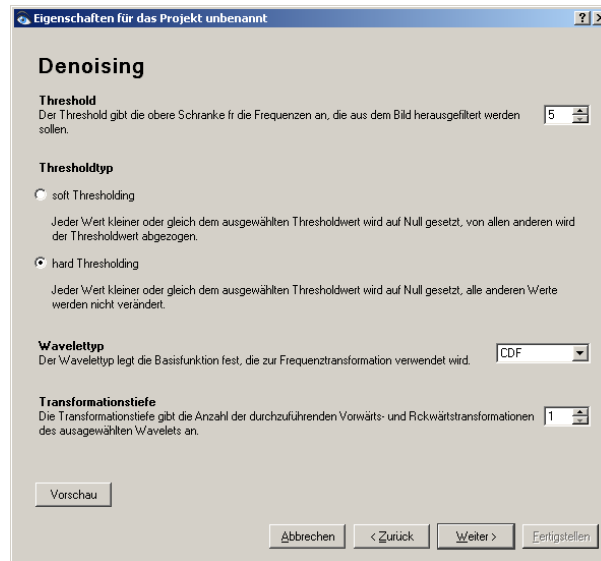


Abbildung A.14: Neues Projekt - Assistent - Parameter für Denoising

cherte Projekt auswählen. Durch Klick auf „Öffnen“ wird das Projekt mit allen bisher ermittelten Daten geladen und das erste Bild der Serie wird angezeigt.

A.3.3 Anzeigen im Hauptfenster

Den größten Teil des Fensters nimmt der Anzeigebereich ein, in dem zunächst das Video angezeigt wird. Im rechten Bereich finden sich Optionen zur Anzeige der ermittelten Daten.

- Visualisierung
Wählen Sie hier das Projekt aus und alle Visualisierungen, die im Anzeigebereich gezeigt werden sollen.
- Globalparameter
Es werden die Parameter des aktuellen Projekts, sowie die einzelnen gewählten Optionen der Berechnungen in Tabellenform angezeigt.
- Frame X
Für das im Anzeigebereich angezeigte Frame X werden die einzelnen errechneten Parameter in Tabellenform angezeigt.
- Geschwindigkeit/Kennzahlen
Wenn Sie im Anzeigebereich einen Doppelklick auf der Lamelle durchführen, erscheinen in diesem Fensterbereich die zugehörigen Geschwindigkeiten und physikalischen Werte an diesem Punkt.

Über die Karteireiter im unteren Bereich kommt man zu folgenden Optionen:

- Wiedergabe
Die gegebenenfalls zu einem Video verarbeiteten Bilder können hier mit den gewählten Visualisierungen wiedergegeben werden.
- Frequenzanalyse
Hier werden die berechneten Frequenzen auf der Lamelle der Düse dargestellt. Sie können zwischen verschiedenen Darstellungsarten wählen, die unterschiedliche Analysemethoden ermöglichen.

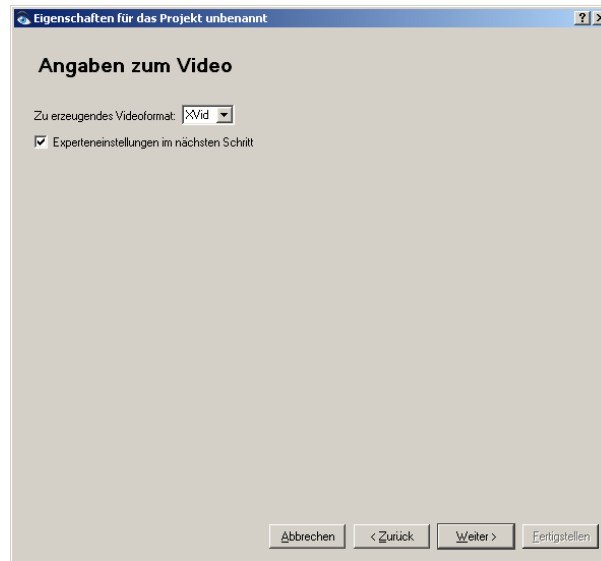


Abbildung A.15: Neues Projekt - Assistent - Auswahl des Videoformats

- Tropfen
In dieser Ansicht wird die Tropfenverteilung angezeigt. Sie können auswählen, ob die Ergebnisse pro Frame oder für die gesamte Serie angezeigt werden sollen. Nach rechts wird die Größe der Tropfen aufgetragen und nach oben die jeweilige Anzahl. Im Fall der gesamten Serie wird die Anzahl über alle Bilder gemittelt.

A.3.4 Menü

Im Folgenden werden die einzelnen Menüeinträge kurz beschrieben

- Menü Datei
 - Neues Projekt
Öffnet den Wizzard, mit dem Sie ein neues Projekt erstellen können (vgl. Kapitel A.3.1).
 - Projekt öffnen
Öffnet einen Dateidialog, in dem Sie ein zuvor erstelltes Projekt laden können.
 - Video öffnen
Öffnet einen Dateidialog, in dem Sie ein zuvor exportiertes Video laden können. Sie können auswählen, ob Sie ein Video (Mono) oder zwei Videos (Stereo) laden möchten.
 - Projekt vergleichen
Öffnen Sie ein zweites Projekt, um die berechneten Daten beider geöffneten Projekte visuell zu vergleichen.
 - Video exportieren
Wählen Sie aus, ob sie ein Mono- oder Stereovideo erstellen möchten und geben Sie einen Dateinamen an. Danach wird ein Video mit allen aktuell gewählten Visualisierungen erstellt.
 - Daten exportieren
Hier können Sie sämtliche berechneten Daten in externe Dateien speichern. Die Lamellenkoordinaten werden dabei als Textdatei abgelegt, wahlweise in eine einzige Datei oder jeweils eine Datei für jeden Frame. Alle übrigen Daten werden als Excel-Dokument gespeichert.



Abbildung A.16: Neues Projekt - Assistent - Markerbestimmung

- Datenbankdump exportieren
Sie haben die Möglichkeit, die Datenbankeinträge zu einem erstellten Projekt in einer Datei zu speichern, um die Daten kopieren zu können und auf einem anderen Datenbankserver zu verwenden. Wählen Sie einen Dateinamen, unter dem der Datenbankdump gespeichert werden soll.
- Datenbankdump importieren
Sie können hier einen zuvor gespeicherten Datenbankdump laden, um ihn dem aktuellen Datenbankserver hinzuzufügen. Wählen Sie dazu die Dumpdatei aus und geben Sie den Pfad zu den ursprünglich verwendeten Bilddaten an.
- Beenden
Hiermit wird das Programm **3D-AugTracer** beendet.
- Menü Berechnungen
Hier können sämtliche Verfahren, die schon unter dem Punkt *Neues Projekt* (vgl. Kapitel A.3.1) vorgestellt wurden.
- Menü Fenster
 - Wiedergabe
Zeigt das Hauptfenster an, das auch über den Karteireiter „Wiedergabe“ erreicht werden kann.
 - Frequenz-Auswertung
Zeigt Frequenzanalyse an, die auch über den Karteireiter „Frequenzanalyse“ erreicht werden kann.
 - Tropfen-Auswertung
Zeigt die Tropfenanalyse an, die auch über den Karteireiter „Tropfen“ erreicht werden kann.
 - Player verbergen / anzeigen
Blendet die Videosteuererelemente aus bzw. ein.



Abbildung A.17: Neues Projekt - Assistent - Lamellensegmentierung

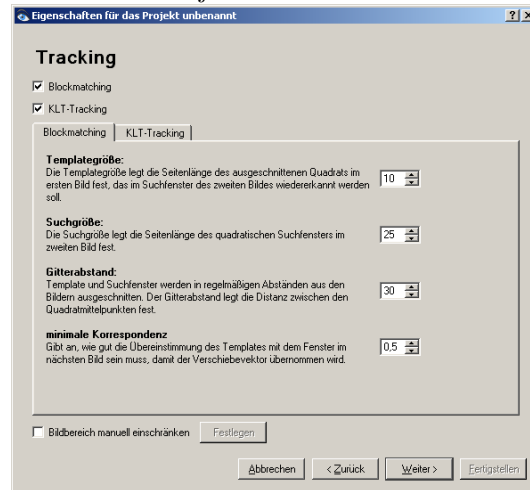


Abbildung A.18: Neues Projekt - Assistent - Block-Matching

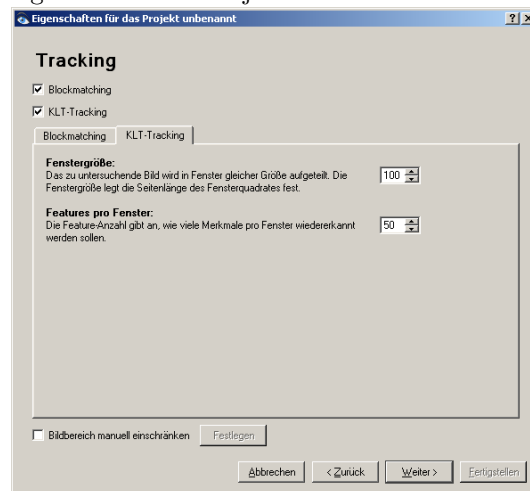


Abbildung A.19: Neues Projekt - Assistent - KLT-Tracking

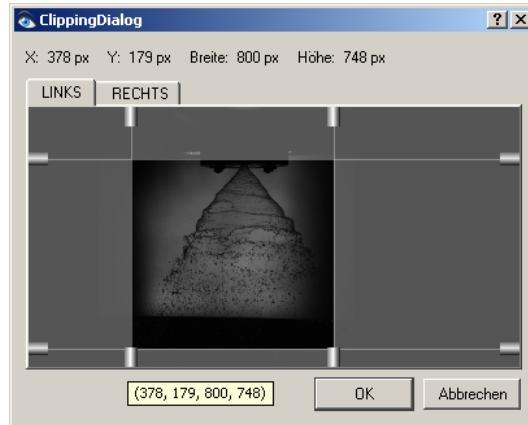


Abbildung A.20: Neues Projekt - Assistent - Clipping

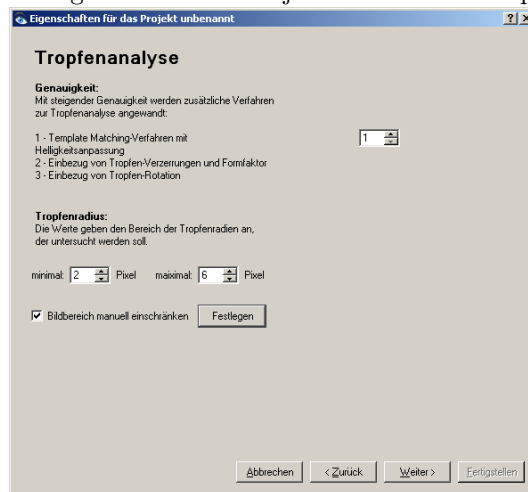


Abbildung A.21: Neues Projekt - Assistent - Tropfenanalyse



Abbildung A.22: Neues Projekt - Assistent - 3D-Rekonstruktion

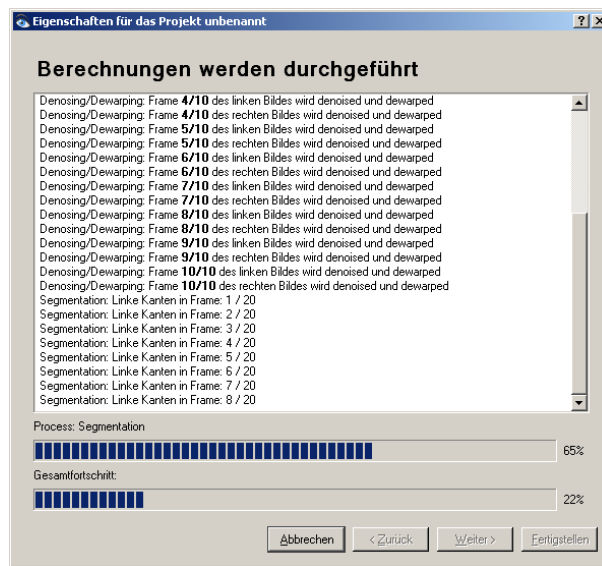


Abbildung A.23: Neues Projekt - Assistent - Ausführung der Berechnungen

Anhang B

Pflichtenheft

Inhalt

B.1 Zielbestimmung	143
B.2 Produkteinsatz	144
B.3 Produktübersicht	145
B.4 Produktfunktionen	145
B.5 Produktdaten	146
B.6 Benutzungsoberfläche	146
B.7 Nichtfunktionale Anforderungen	146
B.8 Technische Produktumgebung	146
B.9 Anforderung an Entwicklungsumgebung	147

Zu Beginn der Projektgruppe wurde ein Pflichtenheft erstellt, welches die Zielsetzung des zu erstellenden Softwareprodukts festlegen sollte. Die Chemietechniker äußerten ihre Anforderungen und Wünsche, die sie an das Programm stellten und die von den Projektteilnehmern umgesetzt werden sollten. In den nachfolgenden Abschnitten werden die Muss- und Wunschkriterien, die Abgrenzungskriterien sowie die Produktfunktionen, Produktdaten und weitere Informationen zu dem zu erstellenden Produkt dargestellt.

B.1 Zielbestimmung

Erstellen von 2 Komponenten:

1. Aufzeichnung
2. Bildverarbeitung & Visualisierung

Musskriterien für 1):

- Bereitstellung von unkomprimierten bzw. komprimierten Rohdaten durch Chemietechniker
- 10 fps, 1024x768, Kamera: Pixelfly
- Möglichkeit zur Eingabe von Versuchs- und Kameraparametern
- siehe Auftrag (a):
 - Akquisition

- Datenexport (Berechnung der Amplitude, Frequenz, Position und Strömungsrichtung und Geschwindigkeit der Tracerpartikel)

Wunschkriterien 1):

- Lieferung der Daten in verlustfreier und komprimierter Qualität
- 100 fps, 1280x1024

Musskriterien für 2):

- Stereographische Projektion
- Echtzeit Rendering
- Möglichkeit zur Eingabe von Versuchs- und Kameraparametern
- siehe Auftrag (b-f):
 - Visualisierung der erweiterten virtuellen Welt
 - Datenaufbereitung
 - Segmentierung
 - Rekonstruktion der synthetischen Daten
 - Kalibrierung und Synthese der AR-Szene

Wunschkriterien 2):

- Stereographisches Video
 1. Polarisationsfilter
 2. Interferenzfilter
- verfolgen von Strukturen (Strombahnen, Löcher) über die gesamte Lamelle
- Strömungsprofil
- Rekonstruktion der Lamelle, möglichst Verfolgen eines rundlaufenden Partikels
- Möglichkeit zum Vergleich zweier Videos
- Excel-Export o.A. der gemessenen Daten

Abgrenzungskriterien:

- keine Darstellung zweier überlagerter Videos
- Echtzeitverarbeitung der Videodaten (Tracer)

B.2 Produkteinsatz

- Anwendungsbereiche: Chemietechniker möchten den Zerfall der Hohlkegeldüsen untersuchen.
- Zielgruppen: Chemietechniker in Grundlagenforschung
- Betriebsbedingungen: <90% Luftfeuchtigkeit, 10-40°C Temperatur für Kameras, Kameras müssen spritzwassergeschützt sein

B.3 Produktübersicht

- Aufnahme, Lage und Bewegungsbestimmung der Tracerpartikel, Visualisierung
- Trennung zwischen Videoaufnahme und Videoverarbeitung/Visualisierung.

B.4 Produktfunktionen

- **Akquisition:**

Der Sprühkegel wird von zwei Hochgeschwindigkeitskameras als Folge von Einzelbildern aufgezeichnet, die digital an den Rechner übertragen werden. Aus den paarweisen, synchronisierten Folgen von Einzelbildern sind zwei zugehörige Videostreams zu erzeugen.

- **Datenaufbereitung:**

Bedingt durch die Kameraoptik kommt es zu Verzerrungen innerhalb der Aufnahmen, wodurch diese nicht direkt verwendbar sind. Unter Verwendung von eingebrachten Markierungsgittern kann zunächst eine Kalibrierung der Kameras erfolgen und anhand der Parameter des Versuchsaufbaus wird eine Entzerrung der Videodaten berechnet. Obwohl die resultierende augmentierte Darstellung offline erfolgen soll, wird eine effiziente algorithmische Umsetzung angestrebt.

- **Segmentierung:**

In den Videosequenzen sind nun die fluoreszierenden Tracerpartikel zu segmentieren.

- **Rekonstruktion der synthetischen Daten:**

Für die im vorhergehenden Schritt segmentierten Tracerpartikel ist nun die tatsächliche räumliche Lage zu bestimmen. Dieses ist möglich, da zu jedem Zeitpunkt zwei Aufnahmen unter einem vorgegebenen Winkel vorliegen und die korrelierenden Aufnahmen in einem geometrischen Zusammenhang (epipolare Geometrie) stehen. Aus den dreidimensionalen Informationen, zu verschiedenen Zeitpunkten, kann anschließend die Flugbahn der Tracerpartikel berechnet werden.

- **Kalibrierung und Synthese der AR-Szene:**

Zur konsistenten Darstellung der realen Videodaten und zusätzlicher synthetischer (computergenerierter) Daten innerhalb einer dreidimensionalen Darstellung ist es notwendig, die jeweiligen Koordinatensysteme der beteiligten Komponenten zueinander in Beziehung zu setzen. Da nicht garantiert ist, dass die Tracerpartikel in allen Frames zu detektieren sind (Verdeckung), muss eine Kalibrierung der Videosequenzen ohne explizite Marker erfolgen. Hierzu sind automatische Feature-Tracking Algorithmen, welche sich beispielsweise an prägnanten Farb- und Kontrastveränderungen orientieren, an die Problemstellung anzupassen. Unter Verwendung der identifizierten Marker kann eine projektive Rekonstruktion der realen und künstlichen Daten zu einer dreidimensionalen Darstellung erfolgen (Merging-Based Projection Reconstruction).

- **Visualisierung der erweiterten virtuellen Welt:**

Konform zu den dreidimensionalen Daten soll die Darstellung primär stereotauglich sein, z.B. HMD.

- Videowiedergabe, Vorlauf, Rücklauf, Zoom, Zeitlupe.
- Mögliche Visualisierungen: Dash-Tubes, Glyphs, Vektorfeld
- Einbettung zweier getrennter Videoplayer in das Programm
- mögliches Zusatzfeature: Frameskipping für effizientere aber ungenauere Berechnung

B.5 Produktdaten

- Tabelle, Tracerpartikelbewegung und zugehörige Videos
- Datenaustausch zw. den beiden Anwendungen durch eine XML-Config Datei, in der z.B. die internen/externen Kameraparameter stehen, Düsengeometrie, Viskosität, (...etc)

B.6 Benutzungsoberfläche

Trennung in 2 Programme:

1. Programm zur Berechnung der Videodaten
2. Programm für die Visualisierung des Videos

Zu 1:

- Einstellung der Codeception und Tracing-Sensitivität
- Möglichkeit zur Eingabe der benutzten Kameraparameter zur authentischen Berechnung

Zu 2:

- Eingebettetes Video mit intuitiven Bedienelementen zur:
 - Auswahl des Datensatzes
 - Art der Visualisierung mit Preview
 - Ausgabeart (Monitor mit/ohne Stereo)
 - Videoplayer-Abspielfunktionen (Abspielen, Pause, Vor-/Rücklauf, Zeitlupe)

B.7 Nichtfunktionale Anforderungen

- Anstreben einer plattformunabhängigen Lösung. Generelle Möglichkeit der
- Nutzung unter Windows

B.8 Technische Produktumgebung

Software:

- Windows
- (Linux)
- OpenGL

Hardware:

- 2 Kameras, Abstand: 6.5cm, Winkel: 3°, Abstand zum Sprühkegel: 1m, max. erfassbare Breite: 40cm
- 1 Computer mit einer schnellen Festplatte zum Aufzeichnen
- 1 Computer mit 8GB Arbeitsspeicher (≥ 3 Festplatten (80GB OS, min. 2x250GB für Videos, 250GB Daten) zum Verarbeiten der Daten)
- Zugriff auf Stereoprojektionsmöglichkeit

B.9 Anforderung an Entwicklungsumgebung

- Qt
- Visual Studio.NET / KDevelop
- UML-Tool: Together
- ARToolkit
- CVS
- diverse Codecs

Abbildungsverzeichnis

1.1	Darstellung der Lamelle	2
1.2	Stroboskopische Darstellung des Sprühkegels	3
1.3	Schematischer Versuchsaufbau	3
1.4	Übersicht des Versuchsaufbaus	5
2.1	Beispiel einer Hohlkegeldüse	8
2.2	Bauarten von Hohlkegeldüsen	9
2.3	Modell einer Hohlkegeldüse	10
2.4	Rotationshyperboloid (Seitenansicht und Aufsicht)	10
2.5	Aerodynamische Wellenbildung	11
2.6	Zerfallsprozess für Wasser und Wasser-Glycerin-Gemisch	13
2.7	Luftkern einer Hohlkegeldüse	13
2.8	Beispiel einer Korrelation sowie der Korrelationsmatrix	14
3.1	Projektion des Weltpunktes auf den Kamerapunkt	16
3.2	Aufnahme des Kalibrierungsgitters unterhalb der Düse	17
3.3	Bildschirmabdruck des Kalibrierungstools	18
3.4	Dewarping-Beispiel, links: verzerrt, rechts: entzerrt	20
3.5	Gitterpunkte, links: verzerrt, rechts: entzerrt	21
3.6	Verschiebevektoren für die Gitterpunkte	21
3.7	Haar-Wavelet	23
3.8	Ein idealer Tiefpassfilter	23
3.9	Lamellenrand (kontrastverstärkt)	24
3.10	Tropfengebiet (kontrastverstärkt)	24
3.11	Cohen-Daubechies-Feauveau-Wavelet	25
4.1	Bildausschnitt mit Histogramm vor dem Histogrammausgleich	29
4.2	Bildausschnitt mit Histogramm nach dem Histogrammausgleich	30
4.3	Bildbearbeitungspipeline	32
4.4	Kantentemplates	34
4.5	Die aus dem Kantentracking resultierenden Silhouettenkanten.	36
4.6	Fehldetektionen beim Kantentracking	37
4.7	Beispiel der Baumstruktur	37
4.8	Die resultierenden Silhouettenkanten nach der Behebung der Fehldetektionen.	38
4.9	Winkelberechnung zur Senkrechten durch den Düsenmittelpunkt (rot)	40
4.10	Sprühkegel und Konturen mit Ausgleichgeraden	43
4.11	Kontur mit Ausgleichgerade	44
4.12	Korrespondierende Amplitudenfunktion $h(t)$	44
4.13	Skizzierung der Winkelverzerrung	45
4.14	Amplitudenfunktion $h(t)$	47
4.15	Fourier-Transformation $F(h(t))$	47

4.16	Rechteck- und Gauß-Fensterfunktion	48
4.17	Wavelet-Typen: a) Haar-Wavelet, b) Mexican-Hat-Wavelet, c) Morlet-Wavelet	49
4.18	Wavelet-Transformation mit Mexican-Hat-Wavelet für $s = 2.5$	50
4.19	Wavelet-Transformation mit Mexican-Hat-Wavelet für $s = 7.5$	50
4.20	Frequenzspektrogramm mit $\alpha = 50$ und $\tau = 1$	50
4.21	Waveletspektrogramm für $j = 0, \dots, 100, \Delta j = 1$	51
4.22	Prinzip der Amplitudenverdoppelung	52
4.23	Prinzip der Amplitudenvervierfachung	53
4.24	Tropfen unterhalb der Lamelle	56
4.25	Templates (Original, skaliert, rotiert)	57
4.26	Tropfen hinter Ebene entfernen	58
4.27	Erkannte Tropfen	59
4.28	Tropfenverteilung	60
4.29	Haar Wavelet	61
4.30	Morlet-Wavelet	62
4.31	Mexican Hat Wavelet	62
4.32	Hölder-Exponenten an der Düse	63
4.33	Veranschaulichung der Segmentierungsverfahren	67
4.34	Veranschlichung der Segmentierungsverfahren an der Lamelle	68
5.1	Features beim KLT-Tracker	70
5.2	Triangulierungsablauf	72
5.3	Gesamtansicht mit Partikelausschnitt	74
5.4	Synthetische Partikel (normal und rotiert)	75
5.5	Korrelationskoeffizienten	77
5.6	Korrelationskoeffizienten (Grauwerte, binarisiert und Median-gefiltert)	78
5.7	Partikelpfade	78
6.1	Epipolare Geometrie	82
6.2	Tool zum Testen der 3D-Rekonstruktion	86
6.3	Rekonstruierte Punkte	87
7.1	Toe-In Methode	91
7.2	Off-Axis Methode	91
8.1	Klassendiagramm	94
8.2	Abhängigkeitsdiagramm	95
8.3	Übersicht für das Sequenzdiagramm zum Erstellen eines neuen Projektes	99
8.4	Sequenzdiagramm zum Erstellen eines neuen Projektes (1/9)	100
8.5	Sequenzdiagramm zum Erstellen eines neuen Projektes (2/9)	101
8.6	Sequenzdiagramm zum Erstellen eines neuen Projektes (3/9)	102
8.7	Sequenzdiagramm zum Erstellen eines neuen Projektes (4/9)	103
8.8	Sequenzdiagramm zum Erstellen eines neuen Projektes (5/9)	104
8.9	Sequenzdiagramm zum Erstellen eines neuen Projektes (6/9)	105
8.10	Sequenzdiagramm zum Erstellen eines neuen Projektes (7/9)	106
8.11	Sequenzdiagramm zum Erstellen eines neuen Projektes (8/9)	107
8.12	Sequenzdiagramm zum Erstellen eines neuen Projektes (9/9)	108
10.1	Versuchsauswertung - Darstellung des Kegels des Beispieldatensatzes	115
10.2	Kalibriermuster sowie gefundene markante Eckpunkte	116
10.3	Versuchsauswertung - Kalibrierergebnis	116
10.4	Versuchsauswertung - Denoising	117
10.5	Versuchsauswertung - Lamellensegmentierung	118

10.6	Versuchsauswertung - Globale Tropfenverteilung	119
10.7	Frequenzanalyse - Amplitudenfunktion	119
10.8	Frequenzanalyse - DFT	120
10.9	Frequenzanalyse - Wavelet	120
10.10	Frequenzanalyse - kumulierte Amplituden	120
10.11	Frequenzanalyse - kumulierte Frequenzen	121
10.12	Frequenzanalyse - Frequenzspektrogramm	121
10.13	Frequenzanalyse - Waveletspektrogramm	121
10.14	Versuchsauswertung - Block-Matching Vektor-Feld	122
10.15	Versuchsauswertung - Block-Matching Vektor-Stromlinien	122
10.16	Versuchsauswertung - Darstellung der 3D-Tropfen	122
A.1	Menüpunkt Extra	126
A.2	Projektparameter eines neuen Projektes angeben	127
A.3	Menüpunkte Modus	127
A.4	Eckpunkte des Referenzgitters bestimmen	128
A.5	Gitterpunkte setzen	128
A.6	Erfolgreiche Kalibrierung	129
A.7	Kalibrierungsergebnisse anzeigen	129
A.8	Optionen für die Kalibrierung	130
A.9	Neues Projekt - Assistent - Titelseite	131
A.10	Neues Projekt - Assistent - Projektparameter	132
A.11	Neues Projekt - Assistent - Dateiauswahl	133
A.12	Neues Projekt - Assistent - Auswahl der durchzuführenden Berechnungen	134
A.13	Neues Projekt - Assistent - Datenbankanbindung	135
A.14	Neues Projekt - Assistent - Parameter für Denoising	136
A.15	Neues Projekt - Assistent - Auswahl des Videoformats	137
A.16	Neues Projekt - Assistent - Markerbestimmung	138
A.17	Neues Projekt - Assistent - Lamellensegmentierung	139
A.18	Neues Projekt - Assistent - Block-Matching	139
A.19	Neues Projekt - Assistent - KLT-Tracking	139
A.20	Neues Projekt - Assistent - Clipping	140
A.21	Neues Projekt - Assistent - Tropfenanalyse	140
A.22	Neues Projekt - Assistent - 3D-Rekonstruktion	140
A.23	Neues Projekt - Assistent - Ausführung der Berechnungen	141

Tabellenverzeichnis

1.1	Stoffdaten bei $T=20^{\circ}\text{C}$	4
2.1	Geometriedaten der Hohlkegeldüse	9
2.2	Eigenschaften ausgewählter Flüssigkeiten	12
5.1	Daubechies-9/7-Filter	75

Literaturverzeichnis

- [3iv] 3ivx - mpeg-4 audio and video compression. <http://www.3ivx.com>. [Last Ref. 2006].
- [Abm94] Wolfgang Abmayr. *Einführung in die digitale Bildverarbeitung*. BG Teubner Verlag, Stuttgart, 1994.
- [Ach85] Dietmar Achilles. *Die Fourier-Transformation in der Signalverarbeitung*. Springer-Verlag, 1985.
- [AMI05] AMI. D2.2 the ami multimodal meeting database - infrastructure, data and management. Technical report, 2005.
- [ANSE00] C. A. Christopoulos A. N. Skodras and T. Ebrahimi. Jpeg2000: The upcoming still image compression standard. 2000.
- [ARTa] Artoolkit. <http://www.hitl.washington.edu/artoolkit>, [Last Ref. 2006].
- [ARTb] How does artoolkit work?
<http://www.hitl.washington.edu/artoolkit/documentation/userarwork.htm>, [Last Ref. 2006].
- [Azu97] Ronald T. Azuma. A survey of augmented reality. <http://citeseer.ist.psu.edu/315409.html>, August 19 1997.
- [Bal98] Helmut Balzert. *Lehrbuch der Software-Technik*. Spektrum Akademischer Verlag, Heidelberg, 1998.
- [BBC05] BBC. Dirac video codec. Technical report, BBC, 2005.
- [Bei83] Boris Beizer. *Software Testing Techniques*. Coriolis Group, 1983.
- [Bla98] Christian Blatter. *Wavelets - Eine Einführung*. Friedr. Vieweg und Sohn, 1998.
- [Bre97] M. Breig. Entwicklung einer bewegungsdetektion und -vorhersage für den einatz zweier beweglichen kameras bei menschlichen bewegungen. Master's thesis, 1997.
- [Bro76] D. Brown. The bundle adjustment - progress and prospect. In *XIII Congress of the ISPRS*, Helsinki, 1976.
- [Bud04] F. Budszuhn. *Visual C++*. Addison-Wesley, 2004.
- [Bur97] R. Burkhardt. *UML - Unified Modeling Language*. Addison-Wesley-Longman, 1997.
- [CVS] Version management with cvs. <http://www.cvshome.org/docs/manual>, [Last Ref. 2006].
- [Dal] Dalsa pantera 1m60 camera.
http://vfm.dalsa.com/selector/prodinfo.asp?lboxProd__ID=DS-21-01M60, [Last Ref. 2006].

- [Dat] Datenblatt pco pixelfly camera.
http://www.pco.de/download/?url=/data/BR_PFE_0505.pdf, [Last Ref. 2006].
- [DEML03] David Donjat, Jean-Luc Estivalezes, Marion Michau, and Gerard Lavergne. Phenomenological study of the pressure swirl atomizer internal flow. In *ICLASS 2003*, Sorrento, Italy, Germany, 2003.
- [DF96] Frederic Devernay and Olivier D. Faugeras. From projective to euclidean reconstruction. In *CVPR*, page 264, 1996.
- [Dir] Dirac video codec.
<http://dirac.sourceforge.net/>. [Last Ref. 2006].
- [Div] Official site of divx video codec. <http://www.divx.com>. [Last Ref. 2006].
- [Dom] N. Dombrowski. The aerodynamic instability and disintegration of liquid sheets.
- [DQ03] Ye Duan and Hong Qin. 2.5D active contour for surface reconstruction. In *VMV*, pages 431–439, 2003.
- [Dus03] Elfriede Dustin. *Effective Software Testing*. Addison Wesley, 2003.
- [FFT] Fftw.
<http://www.fftw.org>. [Last Ref. 2006].
- [FG98] A. Fuhrmann and E. Groller. Real-time techniques for 3D flow visualization. In *IEEE Visualization '98 (VIS '98)*, pages 305–312, Washington - Brussels - Tokyo, October 1998. IEEE.
- [FLW04] D. Feggeler, F. Landwehr, and Peter Walzel. Experimental investigation on surface waves at the exit of hollow cone nozzles. *ILASS*, 2004.
- [Fra] Fraktale Bildkompression - Wikipedia.
http://de.wikipedia.org/wiki/Fraktale_Bildkompression. [Last Ref. 2006].
- [FZ98] Andrew W. Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV (1)*, pages 311–326, 1998.
- [GCH⁺02] Simon Gibson, Jonathan Cook, Toby Howard, Roger J. Hubbard, and Daniel Oram. Accurate camera calibration for off-line, video-based augmented reality. In *ISMAR*, pages 37–46, 2002.
- [Göh02] Daniel Göhring. Digitalkameratechnologien: Ccd kontra cmos. Technical report, Humboldt Universität zu Berlin, 12.08.2002.
- [Gre02] Kate Gregory. *Visual C++ .NET*. Markt + Technik, 2002.
- [Gro02] Beate Grotta. *Erweiterung der Particle Image Velocimetry zur Untersuchung verdünnter Gas-Feststoff-Strömungen*. Shaker Verlag, 2002.
- [GW02a] R. C. Gonzalez and R. E. Woods. *Digital Image Processing, Second Edition*. Prentice Hall, 2002.
- [GW02b] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (2nd Edition)*. Prentice Hall, January 2002.
- [Hab87] P. Haberäcker. *Digitale Bildverarbeitung*. Hanser, München, 1987.
- [Hor] Horita avg-50. <http://www.horita.com/smptevitc.htm#AVG-50>. [Last Ref. 2006].

- [Hor86] Berthold Klaus Paul Horn. *Robot Vision*. MIT Press, Cambridge, Massachusetts, 1986.
- [Huf] Huffuyv video codec. <http://neuron2.net/www.math.berkeley.edu/benrg/huffuyv.html>. [Last Ref. 2006].
- [IM98] Blake A Isard M. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
- [Inf] Infitec. <http://www.infitec.net>. [Last Ref. 2006].
- [Jäh97] Bernd Jähne. *Digitale Bildverarbeitung*. Springer Verlag, 1997.
- [JJ93] Ivins Jim and Porrill John. Everthing you always wanted to know about snakes (but were afraid to ask). *ATVRU Technical Memo,86*, 1993.
- [Kal60] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [Kaz80] H. Kazmierczak. *Erfassung und maschinelle Verarbeitung von Bilddaten*. Springer, Berlin, 1980.
- [KL03] Y. Joo K. Lee, C. Lee. Application technique of particle image velocimetry and entropy analysis to investigate spray structure for gasoline direct injection injector. pages 953–964. *Meas. Sci. Technol.*, 2003.
- [KLT] Klt: An implementation of the kanade-lucas-tomasi feature tracker. www.ces.clemson.edu/~stb/klt/. [Last Ref. 2006].
- [KM88] Terzopoulos D. Kass M., Witkin A. Snake: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.
- [Koc06] Stephan Koch. *Frequenzanalyse von Oberflächenwellen beim Austritt aus einer Hohlkegeldüse mittels CCD-Aufnahmen*. PhD thesis, Universität Dortmund, 2006.
- [Kos05] Rainer Koschke. Softwaretechnik. Vorlesungsfolien SS05, Universität Bremen, 2005.
- [Kra01] Ralph Krause. CVS: An introduction. *Linux Journal*, 87:72, 74, 76, July 2001.
- [Kru04] G. Krueger. *Handbuch der Java-Programmierung*. Addison-Wesley, 2004.
- [KS02] T. Elze K. Stoehr, S. Junick. Verarbeitung von mpeg datenstroemen unter ms windows und linux. Technical report, Motion Picture Experts Group, 2002.
- [KSW97] Frank Knobloch and M. Seeboerger-Weichselbaum. *Das Einsteigerseminar Java*. BHV, Germany, 1997.
- [KV98] Kiriakos N. Kutulakos and James R. Vallino. Calibration-free augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 4(1), January 1998.
- [Lab02] Frank Labatz. *Projektmanagement und Projektsteuerung, Seminar*, 2002.
- [Lan04] Frank Landwehr. *Entwicklung eines Lichtleiterverfahrens zur Charakterisierung von Mehrphasenströmungen*. PhD thesis, Universität Dortmund, 2004.
- [Lef89a] A. H. Lefebvre. Properties of sprays, particle and particle systems characterization, 1989.
- [Lef89b] Arthur H. Lefebvre. *Atomization and Sprays*. Hemisphere Publishing Corporation, 1989.

- [Lew] J. P. Lewis. Fast normalized cross-correlation.
<http://www.idiom.com/~zilla/Work/nvisionInterface/nip.html>. [Last Ref. 2006].
- [LH97] M. Dulberg L. Harrison, D. McAllister. Stereo computer graphics for virtual reality. 1997.
- [Lig04] LightWorld. Endbericht der Projektgruppe 383, Informatik VII, 2004. Universität Dortmund.
- [Lit01] Dave Litwiller. Ccd vs. cmos: Facts and fiction. Technical report, Dalsa, 2001.
- [LL93] F. F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):617–634, June 1993.
- [MBM01] Edward M. Mikhail, James S. Bethel, and J. Chris McGlone. *Introduction to modern photogrammetry*. John Wiley and Sons, New York, NY, USA; London, UK; Sydney, Australia, 2001.
- [Mei02] K. Meissner. *Projektmanagement, Qualitaet, Entwicklungsprozess*, 2002.
- [Mül05] Heinrich Müller. *Vorlesung Digitale Bildverarbeitung*. 2005.
- [mor] more3d.
<http://www.more3d.de>. [Last Ref. 2006].
- [Mor78] J. J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In *G.A. Watson, Lecture Notes in Mathematics 630*, pages 105–116. Springer-Verlag, Berlin, 1978. Cited in Åke Björck’s bibliography on least squares, which is available by anonymous ftp from [math.liu.se](ftp://math.liu.se) in `pub/references`.
- [MPE] Mpeg homepage. <http://www.chiariglione.org/mpeg/>. [Last Ref. 2006].
- [MP1] Mplayer - the movie player.
www.mplayerhq.hu/. [Last Ref. 2006].
- [MS99] R. Boyle M. Sonka, V. Hlavac. Image, processing, analysis and machine vision. PWS Publishing, 1999.
- [MT95] Terzopoulos D. McInerney T. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis. *Comp. Med. Imag. Graph.*, 19(1):69–83, 1995.
- [MyS] Mysql - database server.
www.mysql.org/. [Last Ref. 2006].
- [New] Newmat c++ matrix library.
www.robertnz.net/nm_intro.htm. [Last Ref. 2006].
- [Nie81] H. Niemann. *Pattern analysis*. Springer, Berlin, 1981.
- [NK03] M. Sumida N. Kawahara, E. Tomita T. Nakayama. Microscopic observation of primary spray structure of high-pressure swirl injector for gasoline direct injection engine. In *The 9th Int. Conf. on Liquid Atomization and Spray Systems*, ICLASS, September 2003.
- [OMSS01] Thomas Ottmann, Rainer Müller, G. Seitz, and Ch. Steinert. Video in Vorlesungsaufzeichnungen mit informatikfernen Inhalten am Beispiel Archäologie. *Informatica Didactica*, 2, 2001.

- [Opea] Open mesh: The halfedge data structure.
www-i8.informatik.rwth-aachen.de/OpenMesh/html/libdoc/mesh_hds.html.
[Last Ref. 2006].
- [Opeb] OpenGL - the industry's foundation for high performance graphics.
www.opengl.org/. [Last Ref. 2006].
- [Ots79] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, March 1979.
- [OXM02] Charles B. Owen, Fan Xiao, and Paul Middlin. What is the best fiducial? In *The First IEEE International Augmented Reality Toolkit Workshop*, pages 98–105, Darmstadt, Germany, September 2002.
- [Pol04] Marc Pollefeys. Visual 3d modeling from images. In *VMV*, page 3, 2004.
- [Pri] Sjaak Priester. Delaunay triangles.
http://www.codeguru.com/Cpp/data/mfc_database/misc/article.php/c8901/.
[Last Ref. 2006].
- [PS00] Rob Pooley Perdita Stevens. *UML*. Addison Wesley, 2000.
- [Pum06] Dietmar Pum. *Digitale Bildverarbeitung in der Lebensmittel- und Biotechnologie*. Zentrum für NanoBiotechnologie, Universität für Bodenkultur Wien, 2006.
- [Ree01] Phil Rees. Synchronisation and smpte timecode.
<http://www.philrees.co.uk/articles/timecode.htm>, [Last Ref. 2006], 2001.
- [Rie00] Harald Rieser. Diplomarbeit - pure-java h.263 encoder. Master's thesis, FH Salzburg, 2000.
- [RK98] Erich Rome and Marina Kolesnik. 3D-Szenenrekonstruktion aus Bilddaten. Einführung in den Stand der Technik. Technical report, GMD, Sankt Augustin, Germany, 1998.
- [RL83] P. Robert and C. Labit. *Comparative Study Between Intra and Interframe Predictions Schemes*, pages 307–313. 1983.
- [RM92] D. Ruprecht and H. Muller. Image warping with scattered data interpolation. 1992.
- [RM94] Detlef Ruprecht and Heinrich Müller. Image warping with scattered data interpolation methods. Technical Report Aug16-5, Technical University of Munich, August 16, 1994.
- [RM95] D. Ruprecht and H. Muller. Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications*, 15(2):37–43, March 1995.
- [Roy70] Dr. Winston W. Royce. Managing the development of large software systems. 1970.
- [RZ96] Projet Robotvis and Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. <http://citeseer.ist.psu.edu/316750.htm>, July 04 1996.
- [Sah00] Subhasis Saha. Image compression - from dct to wavelets : A review. Technical report, 2000.
- [Sch00] William J. Schroeder. *Vtk User's Guide*. Kitware Inc, 2000.
- [SG04] Tobias Hanning und Robert Wagner Simone Graf. Projektive rekonstruktion - stereo-bildverarbeitung ohne nebenwissen. 0.8.12.2004.
- [Shr05] Dave Shreiner. *OpenGL Programming Guide*. Addison Wesley, 2005.

- [smp] Society of motion picture television engineers. <http://www.smpete.org/>. [Last Ref. 2006].
- [Sof] Pco pixelfly software camware. http://www.pco.de/data/SW_CAMWAREWIN_213.zip, [Last Ref. 2006].
- [Spe93] Peter Spellucci. *Numerische Verfahren der nichtlinearen Optimierung*. Birkhaeuser, Basel, 1993.
- [Spi01] Andreas Spillner. *Prozess-Modelle für die Softwareentwicklung*. Hochschule Bremen, 2001.
- [Squ] H. B. Squire. Investigation of the instability of a moving liquid film.
- [ST94] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 593–600, Los Alamitos, CA, USA, June 1994. IEEE Computer Society Press.
- [Sta03] Cornel Stan. *Direct Injection Systems: The Next Decade in Engine Technology*. Society of Automotive Engineers, 2003.
- [Sto01] Josef Stoeger. Technischer Projektablauf bei MAGNA STEYR Heavy Stamping. Master's thesis, FH Graz, 2001.
- [Str00] Bjarne Stroustrup. *Die C++ Programmiersprache*. Addison-Wesley, Reading, 2000.
- [TC02] D.M Tsai and C.H. Chen. Rotation-invariant pattern matching using wavelet decomposition. 2002.
- [Tiz98] Hamid R. Tizhoosh. *Fuzzy-Bildverarbeitung*. 1998.
- [Tri96] Bill Triggs. Factorization methods for projective structure and motion. In *CVPR*, pages 845–851, 1996.
- [Tsa87] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Trans. Robotics and Automation*, 3(4):323–344, 1987.
- [TZ00] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, April 2000.
- [uJRuIJ99] G. Booch und J. Rumbaugh und I. Jacobson. *Das UML Benutzerhandbuch*. Addison-Wesley, 1999.
- [Ump06] Thomas Umpfenbach. Bildbasierte Analyse der Silhouetten von Hohlkegeldüsen. Master's thesis, Universität Dortmund, 2006.
- [VVBF04] M. Voorons, Y. Voirin, G. B. Bénéié, and K. Fung. Very high spatial resolution image segmentation based on the multifractal analysis. 2004. 20th congress ISPRS. Istanbul, Turkey.
- [Wah89] Friedrich M. Wahl. *Digitale Bildsignalverarbeitung*. Springer Verlag, Berlin, Germany, 1989.
- [WAI] Waili - wavelets with integer lifting. www.netlib.org/a/wavelet/waili.pdf. [Last Ref. 2006].
- [Wal82] Peter Walzel. *Auslegung von Einstoff-Druckdüsen*, 1982.

- [Wal01] Peter Walzel. Abschätzen von Tropfengrößen an Lamellendüsen - Ein Vergleich verschiedener Ansätze. 2001.
- [WB02] Peter Walzel and Peter Broll. Lamella disintegration at sheet forming nozzles, estimates for drop sizes. In *ILASS Europe 2002*, pages 573–577, Zaragoza, Spain, September 2002.
- [WL03] Peter Walzel and Fank Landwehr. Faseroptisches Sensorarray zur Untersuchung von Gas/Flüssig-Mehrphasenströmungen. In *Chemie Ingenieur Technik*, pages 218–221, Freiberg, Germany, March 2003.
- [Xvi] Home of xvid-codec. <http://www.xvid.org/>. [Last Ref. 2006].
- [YI00] T. Nakajima Y. Ikeda, N. Yamada. Multi-intensity-layer particle-image velocimetry for spray measurement. pages 617–626. *Meas. Sci. Technol.*, 2000.
- [ZDFL95] Zhengyou Zhang, Rachid Deriche, Olivier D. Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artif. Intell*, 78(1-2):87–119, 1995.
- [Zha98] Z.Y. Zhang. Determining the epipolar geometry and its uncertainty: A review. *IJCV*, 27(2):161–195, March 1998.
- [Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell*, 22(11):1330–1334, 2000.
- [ZL77] Ziv and Lempel. A universal algorithm for sequential data compression. *IEEETIT: IEEE Transactions on Information Theory*, 23, 1977.
- [ZL01] Z. Zhang and C. Loop. Estimating the fundamental matrix by transforming image points in projective space. *Computer Vision and Image Understanding*, 82(2):174–180, May 2001.
- [ZSH96] Malte Zöckler, Detlev Stalling, and Hans-Christian Hege. Interactive visualization of 3D-vector fields using illuminated streamlines. In Roni Yagel and Gregory M. Nielson, editors, *Proceedings of the Conference on Visualization*, pages 107–114, Los Alamitos, October 27–November 1 1996. IEEE.

Sachregister

- 3D-Rekonstruktion, 129
- Abrisspunkt, 53
- Amplitude, 41
- Amplituden-Schwellenwert, 52–54
- Amplitudenfunktion, 38
- Amplitudenverdoppelung, 51, 53
- Amplitudenvervierfachung, 52
- Ausgleichsgerade, 41, 87
- Auswertung, 111

- Barrel-Effekt, 19
- Basisfunktion, 22, 47
- Baumstruktur, 34
- Bildglättung, 30

- Clippingrectangle, 129
- Codec, *siehe* Video

- Daubechies-9/7-Filter, 75
- Denoising, 22–24, 127
- Dewarping, 19–22
- DFT, *siehe* Fourier-Transformation
- Doubleshutterbild, 73
- Doubleshuttermodus, 2, 69
- Drall
 - kammer, 9
- Düsen
 - mündung, 9

- Eintrittsöffnung, 9
- Ellipse, 74
- Energie, 64
- Entropie, 64
- Epipol, 79
- Epipolar-Linie, 79
- exportieren, 133
- Extrempunkt, 53, 54

- Feature, 69, 70, 72
- FFT, *siehe* Fourier-Transformation
- Filter, 30
- Formfaktor, 57
- Fourier-Transformation, 47
 - DFT, 45
 - Fast-Fourier-Transformation, 45
 - Gefenstert, 45, 47, 50
- Frequenz, 41
- Frequenz-Spektrum, 49
- Frequenz-Spektrum, 45
- Frequenzanalyse, 11
- Fundamentalmatrix, 79

- Gauß-Funktion, 46
- gemittelte Amplituden, 44
- Geschwindigkeitszahl, 11
- Gitterpunkte, 20
- Grauwertklassen, 30
- Grauwertstufen, 31

- Halbkante, *siehe* Half-Edge
- Half-Edge, 70
- Handbuch, 121
- Hinterschnitt, 43
- Histogramm, 28
- Histogrammausgleich, 27
- Histogrammglättung, 59
- Hochpass, 75
- Hochpunkt, 51
- Hough-Transformation, 55

- importieren, 132, 133
- Installation, 121
 - 3D-AugTracer, 121
 - MySQL-Server, 121
 - Qt, 121
- Intervallgröße, 50

- Kalibrierung, 15–19, 122
- Kalibrierungsgitter, *siehe* Referenzgitter
- Kantendetektion, 73
- Kantentracking, 33
- Kantenzeile, 33
- Kegel, 87
- Kegelwinkel, 11
- Korrelation, 13, 64, 72, 74, 76
- Korrelationsfunktion, 32
- kumulierte Amplituden, 53
- kumulierte Frequenzen, 54

- Lamellenkante, 27
- Lamellenkontur, 41
- Lamellenkoordinaten, 88
- Lamellensegmentierung, *siehe* Segmentierung
- Laser Speckle Velocimetry, 13
- lokales Extremum, 51
- LSV, *siehe* Laser Sepckle Velocimetry

- Marker, 127
- Markerbestimmung, *siehe* Marker
- Medialachsentransformation, 78
- Median-Filter, 77
- Metrik, 44
- Mexican Hat, 61
- Mittelwert, 76
- mittlere Tropfengröße, 11
- Multiquadriken, 20
- MySQL, 121

- Neues Projekt anlegen, 124
- newmat, 19
- normierte Korrelation, 76
- Normierung, 72

- Oberflächenwellen, 11
- Off-Axis Methode, 88
- Ohnesorgezahl, 11
- OpenGL, 89

- Particle Image Velocimetry, 13
- Particle Tracking Velocimetry, 13
- Partikel, 73, 74
- Partikel-Tracking, 73
- Pflichtenheft, 139
- PIV, *siehe* Particle Image Velocimetry13
- PTV, *siehe* Particle Tracking Velocimetry13

- Randartekfakt, 50
- Rauschen, 22
- Rauschreduzierung, 23
- Rechteck-Funktion, 45
- Referenzgitter, 16
- Referenzgitterpunkte, 21
- Regressionsgerade, 35
- Rekonstruktion, *siehe* 3D-Rekonstruktion
- Reynoldszahl, 11
- Ringprojektion, 59, 76
- Ringprojektions-Vektor, 76
- ROI, 40

- Schrittweite, 50
- Schwellenwert, 24, 31, 70, 72, 77
- Schwellenwertverfahren, 30
- Seed, 33

- Segmentierung, 61, 73, 127
- Short-time Fourier-Transformation, 45
- Sichtpyramide, 88
- Simulation, 12
- Skelettierung, 78
- Snake, 58
- Sobel-Operator, 73
- Sprühwinkel, 38, 87
- Sprühkegel, 7
- Stromlinien, 88
- Suchfenster, 72

- Template, 32, 56, 72
- Template-Matching, 32
- Threshold, *siehe* Schwellenwert
- Tiefpass, 50, 75
- Tiefpunkt, 51
- Toe-In Methode, 89
- Trägheit, 64
- Tracerpartikel, 69
- Tracking, 69, 73, 128
 - Blockmatching, 128
 - KLT, 69, 128
 - Patrikeltracking, 128
- Triangulierung, 70
 - Denaunay, 71–72
- Tropfen, 53, 88
- Tropfenanalyse, 88, 132
- Tropfengröße, 11

- Varianz, 30
- Vektorfeld, 70, 78, 88
- Vektorpfad, 78
- Verbrennungsmotor, 7
- Verschiebevektoren, 21
- Verzerrung, 20
- Video, 127, 131–133

- Wavelet, 23
 - Analyse, 47
 - Spektrum, 50
 - Transformation, 22, 47, 50, 61, 75
- Weberzahl, 12
- Wellenlänge, 44
- Wellenlängen-Schwellenwert, 52–54
- Winkelbestimmung, 35
- Winkelverzerrung, 44

- Zerfallslänge, 11
- Zerstäuber, 8