

**Johann Wolfgang Goethe-Universität
Frankfurt am Main**

**Diplomarbeit:
Videobasierte Klassifikation von Eiskristallen unter
Verwendung eines 3D Geometriemodells**

Von Ralph Kuhnert

*Durchgeführt am Institut für Informatik, FB 15 Biologie und Informatik
in Kooperation mit dem Institut für Meteorologie und Geophysik, FB 11
Geowissenschaften*

Prüfer:

Prof. Dr.-Ing. Detlef Krömker
Professur für Graphische Datenverarbeitung
Prof. Dr. Ulrich Schmidt
Professur für Physik der Atmosphäre

Betreuer:

Dr. Ralf Dörner,
Dipl.-Inform. Paul Grimm,
Dr. Heinz Bingemer,
Dr. Ulrich Bundke

15. September 2003

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass die vorliegende Diplomarbeit ohne unzulässige Hilfe und ausschließlich unter Verwendung der angegebenen Literatur angefertigt wurde.

Frankfurt am Main, den 15. September 2003

Ralph Kuhnert

Hintergrund des Themas

Objekterkennung und Objektklassifikation durch die Analyse von Videobildern ist eine Kernaufgabe der Computer Vision, die mit dem heutigen Stand der Technik in der Bildverarbeitung nicht allgemein gelöst werden kann. Trotzdem wurden in diesem Gebiet eine Reihe von Algorithmen und Softwaresysteme entwickelt, die es erlauben, für spezifische Anwendungsfälle Lösungen zu realisieren. Zur Objekterkennung wird dabei häufig ein anwendungstypisches 3D Modell eingesetzt, um daraus einen modell-gestützten Filter zu konstruieren. Die Bildung von Eiskristallen an Eiskeimen unter bestimmten Messbedingungen ist in diesem Kontext eine interessante Aufgabe für die Objekterkennung und Objektklassifikation, da es bereits mathematische Beschreibungen von Teilen des Geometriemodells gibt. Dabei ist die videobasierte Objektklassifikation aus Sicht der Anwendung in der Meteorologie mit folgenden Zielsetzungen verbunden:

Die Anzahl-Konzentration und Struktur primärer Eiskristalle, die in der Atmosphäre auf heterogenem Wege an sog. Eiskeimen entstehen, sind für die Beschreibung der in kalten Wolken über mehrere und teilweise recht verwickelte Sekundärprozesse ablaufenden Niederschlagsentwicklung von großer Bedeutung.

1. Eine in situ-Klassifikation der Formen neu gebildeter Eiskristalle im luftgetragenen Zustand wurde bisher noch nicht vorgenommen. Bei gegebenen Bedingungen in der Messaperatur (Übersättigung von Wasserdampf bzgl. Eis von einigen Prozent) ist neben der Bildung von Eiskristallen ist auch mit der Bildung von unterkühlten Tröpfchen zu rechnen.
2. Ziel dieser Arbeit ist zunächst, im Videobild des „aktivierten“ Aerosols zwischen unterkühlten Tröpfchen und Eiskristallen zu unterscheiden, und deren Anzahlkonzentration im Probenahmefluss zu bestimmen.
3. Eine Zuordnung der beobachteten Eiskristalle zu vorgegebenen Strukturklassen und eine Bestimmung von zugehörigen Strukturparametern der Klasse (z.B. Länge und Durchmesser von hexagonalen Stäbchen) soll in einem zweiten Teil angestrebt werden.
4. Das zu entwickelnde Analyseprogramm soll zunächst an einem vorgegebenen, und noch zu erstellenden Bilddatensatz getestet werden. Zu einem späteren Zeitpunkt soll dann das Analyseprogramm an realen Bildern der Messaperatur trainiert werden können.

Aufgabenstellung

Ziel der Arbeit soll es sein, konstruktive Aussagen über die Durchführbarkeit einer spezifischen bildbasierten Klassifikation in dem oben beschriebenen Anwendungsfall zu treffen. Dabei sind zunächst relevante Ansätze zum einen hinsichtlich Computer Vision, zum anderen hinsichtlich Klassifizierungsalgorithmen aus der Intellektik zu identifizieren und auf ihre Verwendbarkeit für den Anwendungsfall zu bewerten. Weiterhin ist ein Softwaresystem zu entwerfen und prototypisch zu implementieren, das Bilddaten einlesen, Objekte darin erkennen und klassifizieren, sowie die Resultate geeignet präsentieren können soll. Dieses System soll im Rahmen der Arbeit adaptierte Algorithmen aus der Computer Vision implementieren. Die Arbeit ist am Ende kritisch zu evaluieren. Dazu können neben Realaufnahmen auch aussagekräftige künstlich generierte Bilddaten verwendet werden. Die Evaluation soll statistische Aussagen über die zu erwartende Güte der Klassifikation enthalten. Daneben sollen Empfehlungen für den Aufbau eines Messgeräts für den Anwendungskontext, das sich der videobasierten Objektklassifikation bedient, gegeben und mögliche Alternativen diskutiert werden.

Betreuer: Dr. Ralf Dörner, Dipl.-Inform. Paul Grimm, Dr. Heinz Bingemer, Dr. Ulrich Bundke

Prof. Dr.-Ing. Detlef Krömker
Professur für Graphische Datenverarbeitung

Prof. Dr. Ulrich Schmidt
Professur für Physik der Atmosphäre

Abstract

Die Arbeitsgruppe für Chemie und Physik der Atmosphäre am Institut für Meteorologie und Geophysik der Johann Wolfgang Goethe-Universität Frankfurt befasst sich unter anderem mit der Entwicklung einer Continuous Flow Diffusion Chamber zur Erfassung und Klassifikation von CCN und IN. Diese Partikel besitzen eine Größe im Mikrometerbereich und sind somit nicht leicht zu erfassen und zu unterscheiden. Bei vergleichbaren Versuchen beschränkte sich bisher die automatische Auswertung auf die Anzahl der Partikel. Es gibt noch kein Verfahren, welches eine Klassifikation in CCN und IN videobasiert vornehmen kann.

Es lag ebenfalls kein reales Bildmaterial vor, welches zu Testzwecken für die Klassifikation geeignet gewesen wäre. Basierend auf den physikalischen und meteorologischen Grundlagen wurde mittels Raytracing ein künstlicher Bilddatensatz mit kleinen Eiskristallen und Wassertröpfchen unter verschiedenen Betrachtungsverhältnissen erstellt.

Anhand dieses Bilddatensatzes wurde dann ein Verfahren zur Klassifikation entwickelt und prototypisch implementiert, welches dies mittels Methoden aus der graphischen Datenverarbeitung und durch Berechnung der Momente vornimmt.

Es war notwendig, Verfahren aus der Kameratechnik zu betrachten, die später in der realen Anwendung mit sehr kurzzeitiger Belichtung, geeigneter Optik und hochauflösender CCD-Kamera detaillierte Bilder von Objekten in der Größe von einigen $10\mu\text{m}$ liefern können.

Inhaltsverzeichnis

0	Einleitung.....	1
1	Grundlagen.....	3
1.1	Meteorologie und Physik.....	3
1.1.1	Homogenes Kondensationswachstum / Depositionsgefrieren	3
1.1.2	Heterogenes Kondensationswachstum / Depositionsgefrieren.....	3
1.1.3	Cloud Condensation Nuclei und Ice Nuclei.....	4
1.2	Kristallwachstum.....	6
1.2.1	Bilder von Eiskristallen.....	7
1.3	Anwendungshintergrund Continuous Flow Diffusion Chamber.....	8
1.3.1	Bilder aus der existenten CFDC	10
1.3.2	Geplanter Umbau zum Eiskeimzähler.....	11
2	Konstruktion von Testbildern.....	13
2.1	Anforderungen aus Anwendung.....	13
2.2	Verfahren zur Erzeugung der Testbilder.....	14
2.2.1	Auswahl des Verfahrens.....	14
2.2.2	Auswahl des Raytracers.....	15
2.3	Erstellung der Testbilder.....	15
2.3.1	Szene.....	15
2.3.1.1	Modellierung der Kamera.....	16
2.3.1.2	Beleuchtung.....	16
2.3.1.3	Objekte.....	18
2.3.1.4	Aufbau der Szene.....	20
2.4	Die Güte der Testbilder.....	21
2.5	Die verschiedenen Testbilder.....	22
2.5.1	Die einzelnen Formen mit Beleuchtung von der Seite.....	22
2.5.2	Bilder mit Beleuchtung von vorne.....	24
2.5.3	Bilder mit Beleuchtung von hinten.....	26
3	Algorithmen und Implementierung.....	28

3.1 Grundlagen der Implementierung.....	28
3.1.1 Programmiersprache.....	28
3.1.2 Verwendung von Bibliotheken zur Objekterkennung.....	29
3.1.2.1 Gandalf.....	29
3.1.2.2 VXL.....	29
3.1.2.3 Open Source Computer Vision Library.....	30
3.1.2.4 Entscheidung für OpenCV	30
3.2 Algorithmen und Programmaufbau.....	30
3.2.1 Stufe 1: graphische Benutzeroberfläche.....	31
3.2.2 Stufe 2: Einlesen des Bildes als Bitmap und Vorbereiten des Bildes.....	32
3.2.3 Stufe 3: Vorverarbeitung des Bildes.....	32
3.2.3.1 Mittelwert-Filter.....	32
3.2.3.2 Gauss-Filter.....	33
3.2.3.3 Medianfilter.....	34
3.2.3.4 Histogramm und Thresholding.....	34
3.2.3.5 Konstante Wahl des Thresholds.....	35
3.2.3.6 Wahl einer Hysterese für den Threshold.....	35
3.2.3.7 Berechnung des Thresholds mit dem Otsu-Verfahren.....	35
3.2.3.8 Alternative Methode zur Berechnung des Thresholds.....	37
3.2.3.9 Weitere Effekte des Thresholdings.....	37
3.2.4 Stufe 4A: Bildverarbeitung	38
3.2.4.1 Verwendung des Sobel-Filters zur Detektion der Kanten.....	38
3.2.4.2 Anwendung des Canny-Filters zur Detektion der Kanten.....	39
3.2.5 Stufe 4B: Schritt 2 Identifikation und Klassifikation.....	40
3.2.5.1 Trennen der Objekte und Finden der Konturen.....	40
3.2.5.2 Einzelne Konturen finden.....	40
3.2.5.3 Freeman Chain Code.....	41
3.2.5.4 Polygonale Approximation nach Teh-Chin.....	41
3.2.5.5 Verwendung von Ellipse-Fitting.....	42
3.2.5.6 Verwendung von Momenten zur Beschreibung der Form eines Objektes.....	43
3.2.5.7 Objekterkennung.....	44
4 Ergebnisse und Evaluierung.....	46

4.1 Frontalbeleuchtung.....	46
4.2 Hintergrundbeleuchtung.....	47
4.3 Beleuchtung senkrecht zur Kameraachse.....	49
4.4 Variation der Größe.....	52
4.5 Folgerung.....	53
5 Kamertechnik.....	54
5.1 Bedingungen für ein mögliches Aufnahmeverfahren.....	54
5.2 vorhergehendes CFDC Verfahren	54
5.3 PATRICIA.....	55
5.4 PIV / PTV.....	56
5.5 Kamertest mit Digitalkamera.....	57
5.6 Empfehlungen für Kamertechnik.....	57
6 Zusammenfassung und Ausblick.....	59
7 Anhang.....	60
7.1 Literaturverzeichnis.....	60
7.2 Internetquellen.....	61
7.3 Emails.....	65
7.4 Anlagen.....	65

0 Einleitung

Die Arbeitsgruppe für Chemie und Physik der Atmosphäre am Institut für Meteorologie und Geophysik der Johann Wolfgang Goethe-Universität Frankfurt befasst sich unter anderem mit der Erfassung und messtechnischen Bearbeitung von Wolkenkondensationskernen. Hierzu gehört auch die Entwicklung eines passenden Messgerätes. Diese Apparatur soll Wolkenkondensationskeime (videobasiert) erfassen und dahingehend klassifizieren, ob sie zur Bildung von Eiskristallen oder Wassertropfen führen. Bisher beschränkte sich bei vergleichbaren Versuchen die automatische Auswertung auf die Anzahl der Partikel – es gibt jedoch noch kein Verfahren, welches diese Klassifikation videobasiert vornehmen kann.

Das Ziel dieser Arbeit ist die Erkennung und In-Situ-Klassifikation von neugebildeten Eiskristallen und Wassertröpfchen im luftgetragenen Zustand im Videobild in Wasser und Eis. Dies soll mittels geeigneter Verfahren aus der graphischen Datenverarbeitung geschehen, die auch prototypisch implementiert werden sollen. Dabei soll auch eine Aussage getroffen werden, welche Verfahren dafür geeignet sind.

Ein Problem hierbei ist, dass es sich um kleine Teilchen mit einer Größe im Mikrometerbereich handelt, die nicht leicht zu erfassen und zu unterscheiden sind. Dazu war es auch notwendig, Verfahren aus derameratechnik zu betrachten, die später in der realen Anwendung, geeignete Bilder liefern können

Es gab noch kein Bildmaterial, welches zu Testzwecken und zur Entwicklung eines Verfahrens zur Klassifikation geeignet wäre. Es musste zuerst ein Satz an Testbildern generiert werden. Darum war es erforderlich, die Objekte anhand der Grundlagen aus Kristallographie und Physik modellhaft zu beschreiben, deswegen ist für die Wahl eines passenden Verfahrens ein entsprechender Wissenshintergrund in Meteorologie unerlässlich. Hierzu ist die Kenntnis der Kondensationskerne wesentlich, deswegen werde ich deren Eigenschaften zu Beginn näher erläutern.

Ein Problem dieser Arbeit war, dass sich die Voraussetzungen dafür im Laufe der Arbeit geändert haben – am Anfang war noch von Partikelverfolgung die Rede, was zum Beispiel die Anwendung von Kalmanfiltern zur Folge gehabt hätte. ... auf einem Sichtfeld mit einer Größe von wenigen Millimetern. Im weiteren Verlauf der Projektplanung des experimentellen Aufbaus, zeigte sich dass Partikelgeschwindigkeiten von $1 - 5 \text{ms}^{-1}$ im Bereich der Kamera notwendig sind. Eine Partikelverfolgung schied somit aus. Somit ist es schon schwierig genug, die Partikel überhaupt auf ein Bild zu bringen.

Die Arbeit besitzt folgenden Aufbau – im ersten Kapitel werden zunächst die benötigten Grundlagen aus Physik und Meteorologie erläutert. Danach folgen die Grundlagen aus der Kristallographie, um das Wachstum der Eiskristalle zu erklären. Dazu gehört auch die Beschreibung des Anwendungshintergrundes mit der der Continuous Flow Chamber und der Erklärung ihrer Funktionsweise.

Anhand der beschriebenen Grundlagen erfolgt im zweiten Kapitel die Beschreibung des Verfahrens zur Erstellung der Testbilder mittels Raytracing. Es enthält den Aufbau der Szene und die Modellierung der Eiskristalle und Wassertropfen.

Nach der Erstellung der Testbilder, erfolgt im dritten Kapitel die Beschreibung zur Entwicklung und Implementierung und der dabei verwendeten Algorithmen zur Filterung und Bearbeitung der Bilder in den einzelnen Schritten der Bildverarbeitung. Dabei wird ein Verfahren vorgestellt, welches die Klassifikation anhand der Berechnung der Momente einer zweidimensionalen Abbildung von Wassertropfen und Eiskristallen vornimmt.

Daran schließt sich im vierten Kapitel die Evaluierung der Ergebnisse des angewendeten Verfahrens an – inwieweit wirken sich die Richtung der Beleuchtung, die Größe der Teilchen und das verwendete Verfahren auf die Qualität der Klassifikation aus.

Aus den daraus abgeleiteten Ergebnissen werden dann im fünften Kapitel Verfahren aus der Kamertechnik vorstellen, die geeignete Bilder von Wassertropfen und Eiskristallen für das vorgestellte Verfahren liefern können.

Schließen möchte ich mit dem sechsten Kapitel mit der Zusammenfassung und einem Ausblick, wie es weitergehen könnte.

Diese Arbeit hat sich als eine sehr interessante Aufgabe für mich erwiesen – hatte ich doch dadurch die Möglichkeit, Wissen aus meinem Fach Informatik und meinem Nebenfach Meteorologie anzuwenden und miteinander zu kombinieren.

Bedanken möchte ich mich bei Prof. Dr.-Ing. Detlef Krömker und Dr. Ulrich Schmidt, die diese Diplomarbeit ermöglicht haben und bei allen, die mich mit Rat und Ermutigungen bei dieser Arbeit unterstützt haben.

1 Grundlagen

1.1 Meteorologie und Physik

Meines Erachtens ist es notwendig, einen kurzen Überblick über die meteorologischen und physikalischen Grundlagen zu geben, die zur Kondensation von kleinen Tröpfchen oder Eiskristallen führen, da sie schließlich der Grund für meine Arbeit waren.

Eine Voraussetzung für die Kondensation ist, dass der Wasserdampfpartialdruck der Umgebungsluft größer als der Wasserdampfdruck über der Partikeloberfläche ist. Der Sättigungsdampfdruck ist hierbei über runden Oberflächen, wie zum Beispiel einem Wassertropfen, größer als der über einer ebenen Wasserfläche. Zudem ist der Sättigungsdampfdruck über Eis geringer als der über Wasser [Seinfeld98]. Somit ist die Übersättigung über Eis größer als über Wasser und aus diesem Grund wachsen Eiskristalle schneller als Wassertropfen. Bei der Kondensation, unterscheidet man zwei unterschiedlichen Prozessen.

1.1.1 Homogenes Kondensationswachstum / Depositionsgefrieren

Hierbei findet die Kondensation ohne die Hilfe von weiteren Teilchen, den so genannten Kondensationskernen, statt. Sobald die notwendige Übersättigung erreicht ist, kommt es spontan zur Tröpfchenbildung [Mason62]. Diese Methode erfordert allerdings Übersättigungen von mehreren 100%. Solch hohe Übersättigungen kommen in der Atmosphäre im Normalfall nicht vor. Üblich sind Übersättigungen zwischen 1 - 10% oder noch geringer [Pruppacher97]. Daher spielt die homogene Kondensation in der Atmosphäre keine Rolle.

Unter Nukleation versteht man die Entstehung der „Keime“, die durch Kondensations- / Depositionswachstum zu Wolkentropfen / Eiskristallen anwachsen.

1.1.2 Heterogenes Kondensationswachstum / Depositionsgefrieren

Der wesentlich wichtigere Prozess in der Atmosphäre ist die heterogene Kon [Pruppacher97]. Hier bilden sich die Tropfen an Aerosolpartikeln – den sogenannten Kondensationskeimen. Aerosolpartikel, welche die Bildung von Tropfen verursachen, nennt man „Cloud Condensation Nuclei“ kurz CCN. Alle Aerosolpartikel können schließlich Tropfenbildung verursachen wenn die Übersättigung groß genug ist. Deswegen setzt man oft die Anzahlkonzentration der Tropfen bei einer Übersättigung von mehreren 100% mit der Anzahl der Kondensationskerne gleich. Diese heißen dann einfach

„Condensation Nuclei“ – CN. Üblicherweise beginnt das Gefrieren von Wolken bei Temperaturen, die zu warm sind für die homogene Nukleation von Gefrierkernen von Wasser (bei etwa $-3,5^{\circ}\text{C}$). Die Aerosolpartikel, die für heterogenes Depositionswachstum verantwortlich sind, nennt man „Ice Nuclei“ – kurz IN.

1.1.3 Cloud Condensation Nuclei und Ice Nuclei

Die Anzahl der Cloud Condensation Nuclei steigt mit zunehmender Übersättigung, da dann durch den Kelvin-Effekt immer kleinere Aerosolpartikel zur Tropfenbildung aktiviert werden [Seinfeld98]. Die Konzentration und chemische Zusammensetzung von CCN ist variabel. Auch ist die Kondensation von Wasser an Aerosolpartikeln mit einer wasserlöslichen Komponente von der Masse und der chemischen Zusammensetzung der wasserlöslichen Komponente abhängig. Schon bei einer relativen Feuchte von unter 100% werden die löslichen Komponenten flüssig (Deliquescenz). Erhöht man sukzessive die relative Feuchtigkeit, dann erfolgt Gleichgewichtswachstum durch Diffusion bis zu einer kritischen Übersättigung, danach erfolgt die Aktivierung und instabiles Wachstum zu einem Wolkentropfen.

Bei den Ice Nuclei werden vier Arten unterschieden [Pruppacher97], [URL 20]. Es gibt den so genannten „Deposition Mode“. Hier lagert sich Wasser bei Temperaturen von unter 0 Grad Celsius auf dem Aerosolpartikel an und bildet dort Eis. Diese Kerne nennt man „Deposition Nuclei“ oder „Sorptions Nuclei“.

Die nächste Variante ist der „Condensation Freezing Mode“. Das Wasser lagert sich an dem Kondensationskeim an und es bildet sich ein Tropfen, der irgendwann während der Kondensation gefriert. Diesen Kondensationskeim nennt man „Condensation Freezing Nuclei“.

Eine andere Möglichkeit ist der „Immersion Mode“. Der IN ist schon bei Temperaturen über 0 Grad Celsius im Tropfen enthalten. Der Tropfen gefriert sobald die Temperaturen gering genug sind. Diese Kerne nennt man „Immersion Nuclei“.

Die vierte und letzte Variante ist der „Contact Mode“. Das unterkühlte Tröpfchen, das eine Temperatur geringer 0°C hat, gefriert, sobald es von dem Eiskeim berührt wird. Dieser wird „Contact Nuclei“ genannt.

Für einen Aerosolpartikel von gegebener chemischer Zusammensetzung und Größe ist es möglich, auf jede dieser vier Arten zu reagieren. Für jede der 4 Arten ist die Temperatur charakteristisch.

Vergleicht man die Gesamtzahl der Aerosolpartikel mit der Anzahl der IN, kann man erkennen, dass nur eine geringe Untermenge der Aerosolpartikel als IN dient. Es wird somit deutlich, dass es an einen Eiskeim verschiedene Anforderungen gibt, die es ihm ermöglichen, als Eiskeim zu dienen.

Nach Untersuchungen von [Pruppacher97] soll ein IN im allgemeinen wasserunlöslich sein, denn sonst würde er sich ja bei Kontakt mit einem flüssigen Tropfen im Wasser auflösen und könnte nicht im „Immersion Mode“ und „Condensation Freezing Mode“ als Eiskeim dienen.

Weiterhin sollte der IN eine gewissen kritische Größe besitzen [Pruppacher97], denn die Effizienz, als Eiskeim zu dienen, nimmt mit fallender Größe stark ab und wird immer stärker von der Temperatur abhängig. Bei gegebener Temperatur und Übersättigung gibt es eine kritische Größe, die der IN mindestens haben muss. Üblicherweise geht man von einer minimalen Größe von $0,1\mu\text{m}$ aus.

Wichtig sind auch die chemischen Bindungen auf der Oberfläche des IN. Diese sollten nämlich in Typ, Polarität und Stärke den chemischen Bindungen von Eis entsprechen [Pruppacher97]. Es ist von Vorteil wenn sie rotationssymmetrisch sind.

Die passenden chemischen Bindungen sind alleine aber noch nicht ausreichend, es gibt auch noch die kristallographischen Anforderungen. Die Kristallstrukturen von Eis und IN sollten ähnlich sein, hinsichtlich der geometrischen Anordnung der Bindungen. Je ähnlicher diese der von Eis ist um so einfacher kann das Eis dort anknüpfen.

Als letztes gibt es noch das Active Site Requirement. Das bedeutet, dass an den Aerosolpartikeln Stellen vorhanden sein müssen, an denen die Eisbildung ansetzen kann [Pruppacher97]. Gewöhnlich sind dies Stellen am Partikel, an denen Wasser angelagert wird und dort beginnt auch der Gefrierprozeß.

Aufgrund dieser Anforderungen bestehen IN zum Beispiel oft aus Tonmineralien oder Seesalz [Ebert00] und haben eine Größe zwischen $0,1$ und $15\mu\text{m}$, ihre Anzahl ist sehr variabel.

Man erkennt CCN und IN erst daran, dass sie zu Wolkentropfen beziehungsweise Eiskristallen werden. Es gibt zur Zeit keine andere Eigenschaft, die sie als CCN oder IN identifizierbar macht.

1.2 Kristallwachstum

Die Eiskristalle wachsen durch Deposition, das heißt Wasserdampfmoleküle lagern sich durch Diffusion an der Oberfläche des Eiskristalls ab.

Große Eiskristalle, wie zum Beispiel Schneeflocken, haben eine Vielzahl von Formen [URL 40], [URL 41], [URL 42], jedoch vom kristallographischen Standpunkt aus gesehen, haben Eiskristalle eine Basisform, nämlich die eines hexagonalen Prismas [Wang02], mit zwei Basisebenen (vom Typ $[0001]$) und sechs Seitenflächen (vom Typ $[10\bar{1}0]$) [Pruppacher97]. Flächen anderen Typs kommen nur sehr selten vor, da sie metastabil sind. Das Verhalten eines Eiskristalls wird durch die Oberflächen bestimmt, die am langsamsten wachsen. Metastabile Oberflächen wachsen am schnellsten und werden zu Kanten und Ecken eines Eiskristalls, die stabilen Oberflächen dagegen wachsen nur langsam und bilden somit die Begrenzungsflächen. Das Verhältnis der Wachstumsraten der Seitenflächen zu dem der Basisflächen variiert in einer charakteristischen Weise mit der Temperatur und der Übersättigung. Dadurch entstehen auch die verschiedenen Formen der Eiskristalle. So kann man trotz der vielen Formen, die es bei großen Eiskristallen gibt, davon ausgehen, dass die Formen kleiner Eiskristalle nahe an der Basisform sind. Beispiel für die Form kleiner Eiskristalle finden sich in Abbildung 1 und 2..

1.2.1 Bilder von Eiskristallen

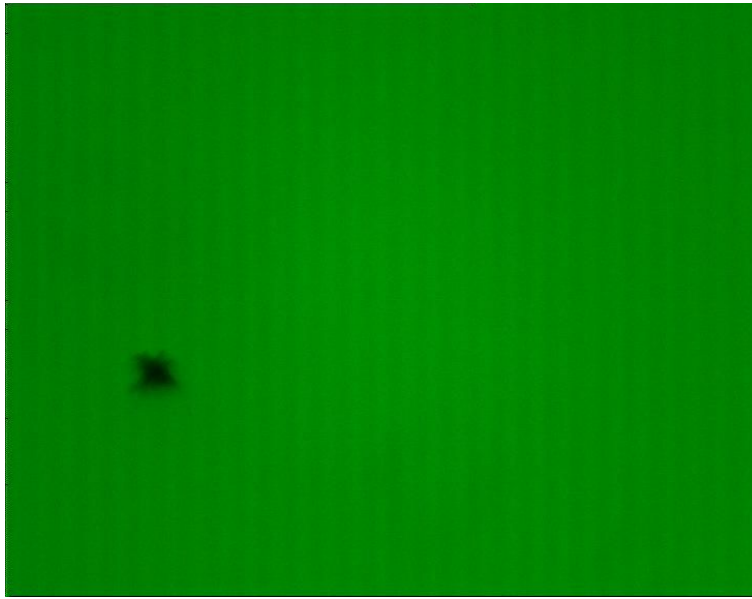


Abbildung 1 Bild eines Eiskristalls der Größe $35\mu\text{m}$ aus PATRICIA [URL 29]

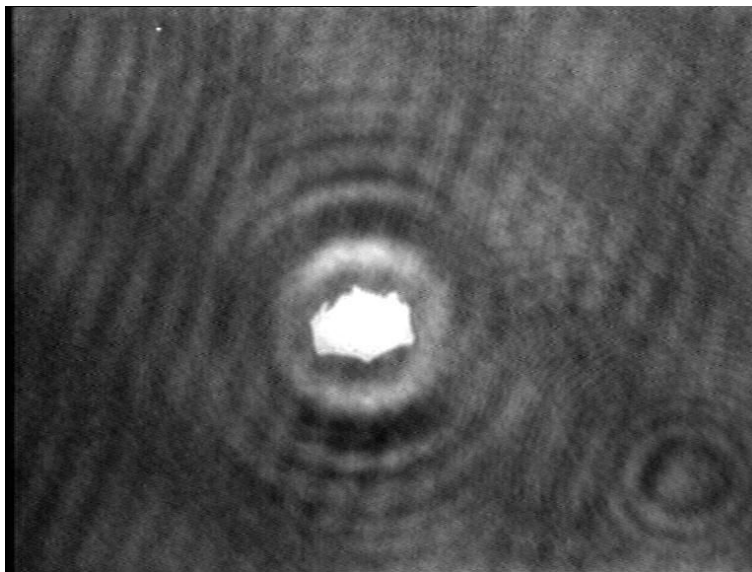


Abbildung 2 Hologramm eines Eiskristalls in der Form eines hexagonalen Plättchens der Größe 2mm aus HODAR [Vössing98] , [Email 2]

1.3 Anwendungshintergrund Continuous Flow Diffusion Chamber

Das Ziel dieser Arbeit ist die Auswertung der Daten einer sogenannten „Continuous Flow Diffusion Chamber“ kurz CFDC. Dies ist ein schon mehrfach verwendetes Messinstrument zur Messung von Wolkenkondensationskernen z. B. die CFDC von der Colorado State University [URL 21] in CRYSTAL-FACE [URL 6] oder FIRE ACE [URL 11]. Hierzu eine kurze Erklärung der Funktionsweise einer CFDC [Rogers98], [Otto02]. Eine im Institut für Meteorologie und Geophysik bereits eingesetzte CFDC (siehe Abbildung 3) besteht aus zwei vertikal angeordneten Platten. Diese beide Platten werden feucht gehalten, jedoch werden beide unterschiedlich temperiert. Dadurch entsteht zwischen diesen beiden Platten eine Zone, die mit Wasserdampf übersättigt ist. Diese Übersättigung ist nur abhängig von den Temperaturen der beiden Platten. Ein Luftstrom, welcher die in der Probenluft enthaltenen Kondensationskerne enthält, strömt nun zwischen diesen Platten hindurch. Wenn die Kondensationskerne durch die mit Wasserdampf übersättigte Zone strömen, kommt es dort zur Kondensation des Wasserdampfs an den Kondensationskernen und diese wachsen dann durch Diffusion von Wasserdampf zu kleinen Tröpfchen an. Diese werden anschließend von einer Videokamera oder einem Optical Particle Counter erfasst und können ausgewertet werden. Das Institut für Meteorologie und Geophysik plant nun eine Modifikation dieses Gerätes mit der auch bei tiefen Temperaturen IN nachgewiesen werden können. Da bei der angestrebten maximalen Eisübersättigung von 160% auch eine Übersättigung über Wasser besteht, ist mit der Bildung von unterkühlten Tröpfchen zu rechnen.

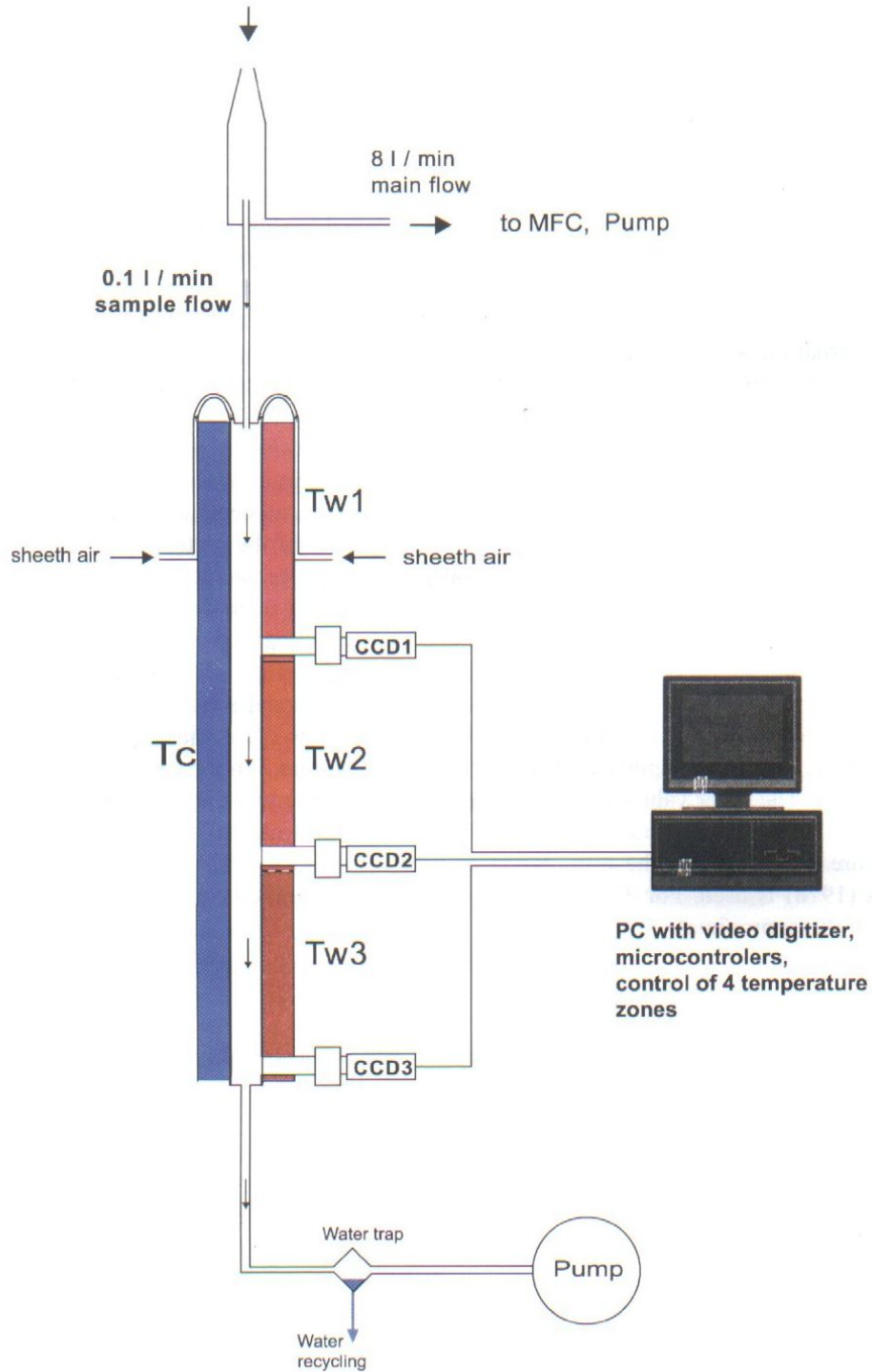


Fig. 1. Flow diagram and layout of the CCN spectrometer. The cold plate (left) is at $T_c = 20^\circ\text{C}$, the temperature of the warm plates (right) increases from top (T_{w1}) to bottom (T_{w3}).

Abbildung 3 Zeichnung der existenten CFDC aus [Otto02]

1.3.1 Bilder aus der existenten CFDC

Zum Vergleich sind hier ein paar Bilder [Email 1] aus der existenten CFDC [Otto02].
Diese zeigen allerdings nur Wassertropfen.

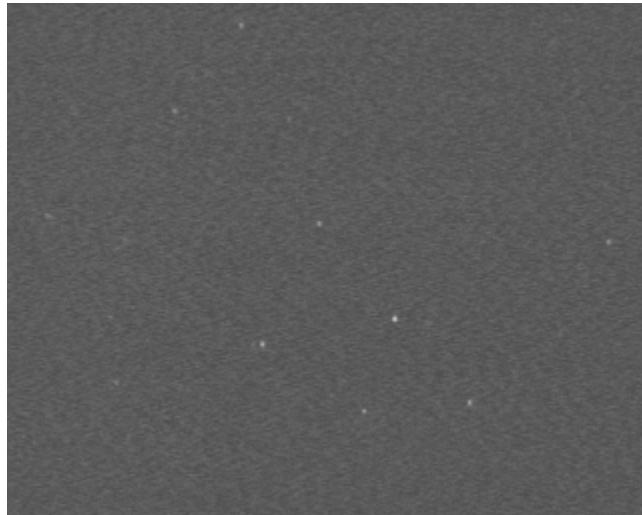


Abbildung 4 Bild 1 aus der existenten CFDC



Abbildung 5 Bild 2 aus der existenten CFDC

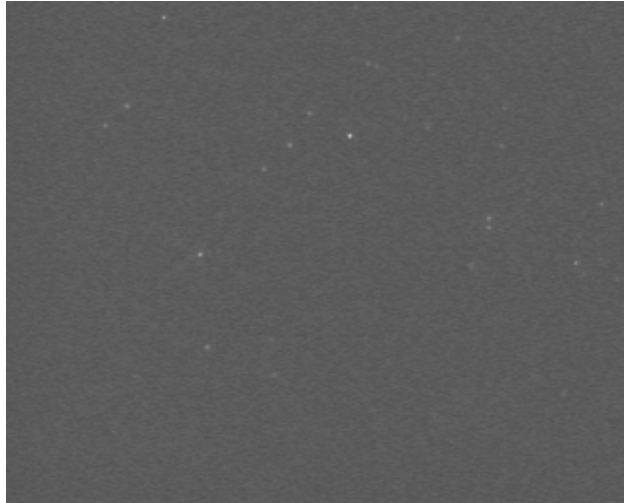


Abbildung 6 Bild 3 aus der existenten CFDC

1.3.2 Geplanter Umbau zum Eiskeimzähler

Zuerst war geplant, dass die Kamera einen konstanten Strom von Bildern liefert. Dabei gibt es aber einige Probleme. Zum einen rechnet man mit sehr wenigen Kondensationskernen – nicht jedes Bild würde überhaupt ein Objekt enthalten und zum anderen werden diese Teilchen voraussichtlich mit einer relativ hohen Geschwindigkeit von etwa $1 - 5\text{ms}^{-1}$ sich durch das Sichtfeld der Kamera bewegen, welches eine Größe im Millimeterbereich hat – im vorhergehenden Versuch waren es $12 \times 15\text{mm}$. Für den aktuellen Versuch gibt es Überlegungen für ein Sichtfeld von $4 \times 4\text{mm}$. Ein weiteres Problem ist, wenn man einen Strom von Bildern mit relativ hoher Auflösung hat und diesen speichern und verarbeiten will, kommt man ganz schnell an die Grenzen für Datentransfer und Speicherplatz. Zum Beispiel hat eine unkomprimiertes Bild in der für die Testbilder verwendeten hohen Auflösung von 2272×1704 Pixel in Graustufen eine Größe von etwa 3,7 Megabyte – bei 20 Bilder pro Sekunde ergibt dies 74MBs^{-1} , was zum Beispiel für USB 2.0 zu viel ist, mit welchem eine Datentransferrate 60MBs^{-1} (480Mbits^{-1} [URL 50]) erreichbar ist.

Aus diesem Grund gibt es mittlerweile ein neues Konzept zur Lösung dieses Problems. Die Partikel werden, bevor sie die Kamera passieren, von einem Optical Particle Counter mit einem Laser erfasst und gezählt. Er sorgt dann dafür, dass, wenn sich der Partikel im Aufnahmebereich der Kamera befinden, eine Blitzlampe aufleuchtet und die Kamera das Bild aufnehmen kann. Hierdurch kommt es aber auch zu einer Mehrfachbelichtung der Bilder. Man hat so eine Reduktion der Anzahl der Bilder. Durch die kurze Belichtungszeit

mit Blitzlampe, erhält man kaum Eigenbewegung der Partikel. Da der Optical Particle Counter die Blitzlampe auslöst, wird auch jedes Partikel im Bild erfasst.

2 Konstruktion von Testbildern

Nachdem nun die Gegebenheiten, die zur Entstehung der Wassertropfen und Eiskristalle führen, bekannt sind, kann hier die Beschreibung über die Erzeugung der Testbilder folgen, an denen später die Algorithmen zur Bildverarbeitung getestet werden sollen.

2.1 Anforderungen aus Anwendung

Die Testbilder sollen die Gegebenheiten des Versuches möglichst genau darstellen. Die Objekte mit ihren Eigenschaften, also Form und Größe der Eiskristalle und Wassertropfen und ihre physikalischen Eigenschaften wie Reflexion und Brechung, sollen so exakt wie möglich wiedergegeben werden.

Die darzustellenden Eiskristalle und Wassertropfen sind klein – ihre Größe wird in der Realität einige μm betragen und bestenfalls einige 100μ sein. Betrachtet man die Bilder *Abbildung 4*, *Abbildung 5* und *Abbildung 6* in Kapitel *Anwendungshintergrund Continuous Flow Diffusion Chamber* kann man erkennen, wie klein die beobachteten Partikel sind, wenn man sie auf einem Blickfeld von $12\text{mm} \times 15\text{mm}$ betrachtet.

Die Größe wird dadurch bestimmt, welche CCN und IN vorhanden sind und wie lange sie sich in der CFDC aufhalten und welche relative Feuchte dort herrscht. Aufgrund der physikalischen Gegebenheiten wachsen Eiskristalle schneller als Wassertropfen und eine höhere Übersättigung fördert ein schnelleres Wachstum. (siehe Kapitel Meteorologie und Physik)

Desweiteren geht man nur von einer begrenzten Anzahl von Objekten pro Bild aus, da die Anzahlkonzentration der aktivierten CCN und IN gering ist.

Diese Objekte befinden sich in einem Luftstrom, in welchem sie sich an der Kamera vorbeibewegen. In diesem Luftstrom können die Partikel rein zufällig im Raum verteilt und orientiert sein.

Die Qualität der Bilder ist stark abhängig von der in der Kamera verwendeten Elektronik und Optik. Sie bestimmen, mit welcher Auflösung die Bilder dargestellt werden und mit wie vielen Pixeln ein Objekt beschrieben wird. Die Optik beeinflusst wie stark die Objekte vergrößert werden, aber auch in welchem Bereich die Kamera scharf sieht. Ein Nebeneffekt der Kamera ist auch das auf dem Bildern sichtbare Rauschen, welches durch den CCD-Sensor hervorgerufen werden kann.

2.2 Verfahren zur Erzeugung der Testbilder

Es stellte sich zunächst die Frage nach einem geeigneten Verfahren zur Erzeugung von Testbildern, da es noch keine realen Bilder gab an denen das Verfahren getestet werden konnte. Auch gibt es bis jetzt noch keine vergleichbaren Experimente, die solche Bilder liefern könnten. Man kann zwar relative gute Bilder von Eiskristallen mit dem „Cloud Particle Imager“ - kurz CPI bekommen [Heymsfield02], doch leider sind diese Bilder nicht vergleichbar mit den zu erwartenden, da die Aufnahmebedingungen komplett verschieden sind – bei CPI schlagen die Eiskristalle beziehungsweise Partikel auf einer Oberfläche auf und werden dann fotografiert – insofern sind auch keine Bilder von Wassertropfen möglich. Ähnliches gilt für Hologramme [Vössing98] und Elektronenmikroskopie [URL 7].

Bei der geplanten CFDC fliegen Eiskristalle und Wassertropfen frei an der Kamera vorbei und werden im Flug fotografiert. Somit blieb für die Testbilder nur die künstliche Bilderzeugung und es stellte sich die Frage nach einem geeigneten Verfahren.

In der Computergrafik gibt es verschiedene Verfahren zur Erzeugung möglichst realistischer Bilder.

2.2.1 Auswahl des Verfahrens

Es gibt die Möglichkeit solche Bilder einfach mit einem Grafikeditor, wie zum Beispiel PaintShop Pro oder Adobe Photoshop, quasi zu malen. Diese Methode ist jedoch nicht geeignet, da dies physikalisch ungenau ist, denn es erfolgt eine rein subjektive Gestaltung. Außerdem wäre die manuelle Erstellung eines jeden Testbildes erforderlich. Für die Modellierung auf physikalischen Grundlagen bietet sich ein anderes Verfahren an – das Raytracing. Es basiert auf physikalischen Grundlagen, und zwar auf dem Prinzip der Strahlverfolgung. Hier wird die Ausbreitung von Licht durch Transmission, Brechung und Reflexion an den Objekten simuliert. In der Realität geht das Licht geradlinig von den Lichtquellen aus, beleuchtet die Objekte und erreicht über Reflexion, Streuung und Brechung den Beobachter beziehungsweise die Kamera. Da schließlich nur ein geringer Teil der Lichtstrahlen den Beobachter erreicht, geht man beim Raytracing in umgekehrter Reihenfolge vor. Hier verfolgt man modellhafte Lichtstrahlen von der Kamera aus, diese werden dann an den Objekten der Szene ideal reflektiert oder ideal gebrochen. Sobald ein Lichtstrahl ein Objekt trifft, wird dort eine lokale Beleuchtungsrechnung durchgeführt und

der Weg des Lichtstrahls weiterverfolgt. Dies geschieht solange, bis die auf dem Strahl transportierte Energie zu gering wird, der Strahl die Szene verlässt oder auf eine Lichtquelle trifft. Aufgrund dieses Funktionsprinzips werden vor allem Effekte wie Reflexion und Brechung korrekt berücksichtigt und dargestellt. Allerdings wird keine diffuse Beleuchtung berechnet. Dies kann nur durch globale Parameter oder weitere Verfahren wie zum Beispiel Radiosity ergänzt werden.

2.2.2 Auswahl des Raytracers

Als Raytracer habe ich POV-Ray 3.5 von www.povray.org [URL 31] ausgewählt. POV-Ray bietet eine ganze Reihe von Vorteilen. Es ist als Freeware erhältlich und verfügt über eine große Vielfalt an Funktionen wie zum Beispiel Kameraunschärfe und verschiedene Beleuchtungsmodelle. Die Gestaltung photorealistischer Bilder wird ermöglicht und weiter fortgeschrittene Techniken wie Photonmapping sind schon implementiert.

Es gibt zwar halboffizielle Weiterentwicklungen von POV-Ray wie zum Beispiel MEGAPOV, in denen einzelne Funktionen noch erweitert und verbessert wurden. Dennoch habe ich mich für POV-Ray entschieden, da es über eine sehr gute Dokumentation verfügt. Weiterhin gibt es auch eine ganze Reihe frei verfügbarer Erweiterungen, mit denen fehlende Features und Objekte ergänzt werden können, zum Beispiel über www.povray.org/resources/links/ [URL 35] oder www.f-lohmueller.de [URL 33].

2.3 Erstellung der Testbilder

Mit dem ausgewählten Verfahren folgt nun die Erstellung der erforderlichen Testbilder mittels POV-Ray.

2.3.1 Szene

Zunächst wurde eine Szene mit POV-Ray modelliert, welche die Gegebenheiten des Versuchs darstellen soll. Eine Szene kann in POV-Ray aus der (virtuellen) Kamera, den Objekten, dazu gehören auch die Lichtquellen, den Effekten, den globalen Parametern und den „Language Directives“ bestehen [URL 34].

2.3.1.1 Modellierung der Kamera

Wichtig für eine genaue Abbildung der Szene war die Modellierung der Kamera. Das Standardmodell, der in POV-Ray verwendeten Kamera, ist eine Lochkamera mit Punktblende und perspektivischer Projektion [URL 34]. Diese Lochkamera bildet in Grundeinstellung alle Objekte scharf ab. Dagegen besitzt eine reale Kamera mit abbildenden Objektiv nur eine begrenzte Tiefenschärfe, die von Optik und Blende abhängig ist.

POV-Ray bietet jedoch die Möglichkeit, die begrenzte Tiefenschärfe einer solchen realen Kamera zu simulieren. Dies geschieht, indem die Apertur der virtuellen Lochkamera, die in Grundeinstellung die Größe 0 hat, vergrößert wird. Dann wird ein Blickpunkt definiert, an welchem die virtuelle Kamera (mit begrenzter Tiefenschärfe) scharf sieht. Je größer die Apertur, in desto geringerer Entfernung von diesem Blickpunkt beginnt sich der Unschärfefeffekt auszuwirken. POV-Ray berechnet diesen Effekt, indem es unter Berücksichtigung der angegebenen Apertur mehrere Lichtstrahlen für jedes Pixel aussendet und dann für das Endergebnis den Durchschnitt bildet.[URL 34]

Diese Methode liefert zwar Ergebnisse, welche den Effekt der Tiefenschärfe in den generierten Bilder realistisch aussehen lassen, sie hat jedoch den Nachteil, dass sich den Werten für die Apertur keine realen Tiefenschärfebereiche zuordnen lassen.

Seit den vorangegangenen Versuchen mit der CFDC ist mittlerweile auch die Kameratechnik weiterentwickelt worden. Inzwischen sind Kameras verfügbar, welche eine höhere Auflösung als die damals verwendeten 320 x 256 Pixel haben. Zum Beispiel in dem Experiment PATRICA [URL 29] wird eine Kamera mit 732 x 573 Pixel verwendet und Bröder/Sommerfeld [URL 8] verwenden eine Kamera mit 1280 x 1024 Pixel.

Auf Anregung von Herr Dr. Bundke wurde zum Rendern eine Kamera mit einer noch höheren Auflösung verwendet. Kameras mit einer Pixelzahl von etwa 4 Millionen Pixel sind in manchen Bereichen schon üblich. Somit wurden die Bilder in einer Auflösung von 2272 x 1704 Pixel gerendert – diese Auflösung wurde einfach von einer solchen Kamera übernommen.

2.3.1.2 Beleuchtung

Eine möglichst realistische Beleuchtung der Szene ist wichtig. Üblicherweise werden beim Raytracing und somit auch bei POV-Ray Punktlichtquellen verwendet. Reale Lichtquellen dagegen sind nicht punktförmig, sondern besitzen eine gewisse flächige Ausdehnung.

Punktlichtquellen haben den Nachteil, dass die Schatten zu scharfe unrealistische Kanten haben, reale Flächenlichtquellen dagegen verursachen einen Schatten mit weichen Kanten. Das betrifft hier vor allem den Schatten, den die Eiskristalle und Wassertropfen auf andere Partikel oder sich selbst werfen.

Dies ist hier von Bedeutung, da die Eiskristalle und Wassertropfen in Realität kleiner als die Lichtquelle sind. Wenn man nun für die Testbilder eine Punktlichtquelle verwenden würde, würde nur die direkt beleuchtete Seite der Objekte entsprechend ausgeleuchtet, die andere Seite würde dann total im Schatten liegen. Wird dagegen eine Flächenlichtquelle verwendet, gelangt auch noch etwas Licht am Objekt vorbei und somit auf die Schattenseite des Objektes. Man hat somit noch eine schwach beleuchtete Zone mit Halbschatten.

POVRay ermöglicht auch die Verwendung von Flächenlichtquellen. Diese wurde aus den genannten Gründen für die Erstellung der Szene verwendet. Für die Größe dieser Lichtquelle wurde zunächst ein realistische Maß von 5 x 5mm angenommen.

Die nächste Technik die angewendet wurde um das Beleuchtungsmodell realistischer zu gestalten, ist das Photonmapping [URL 18]. Im Normalfall verfolgt man beim Raytracing Lichtstrahlen von der virtuellen Kamera durch die Bildebene auf die Szene, bis sie schließlich an den Lichtquellen oder im Hintergrund ankommen. Beim Photonmapping geht man dagegen umgekehrt vor. Wie in der Realität, gehen die Photonen von den Lichtquellen aus und erreichen über Brechung und Reflexion an den Objekten in der Szene schließlich die Kamera.

Photonmapping ist vor allen interessant für spiegelnde und transparente Objekte, dazu gehören auch Wassertropfen und Eiskristalle. Mittels dieser Technik können sich Objekte ineinander spiegeln und auch durch Brechung entstehenden Effekte („reflective and refractive caustics“ [URL 32]) werden vom Prinzip her so berechnet, wie sie in Realität zustande kommen. Auf diese Art und Weise erhält man erheblich realistischere Reflexionen und Brechungseffekte.

Hierzu gehört auch die Definition des Interiors der Objekte. Reflexion und Brechung der Objekte sind wie in der Realität abhängig vom ihrem Brechungsindex. Somit wurden Bilder, abhängig vom Brechungsindices für Eis und für Wasser, berechnet, um die dadurch entstehende Brechungs- und Reflexionseffekte korrekt zu berücksichtigen.

2.3.1.3 Objekte

Zur einfacheren Handhabung, um die geometrische Modellierung von der Beleuchtung zu trennen, wurden die Objekte nicht direkt in der Szene modelliert, sondern es wurde in POV-Ray eine Vorlagendatei (crystals.inc) mit den unterschiedlichen Objekten erstellt.

In dieser Vorlage wurden vier Basisobjekte definiert. Das einfachste dieser Objekte ist das Wassertröpfchen. Es wird in Näherung als Kugel modelliert (siehe Abbildung 7). Wassertröpfchen dieser Größe haben aufgrund der Oberflächenspannung auch in der Realität Kugelgestalt [Pruppacher97].

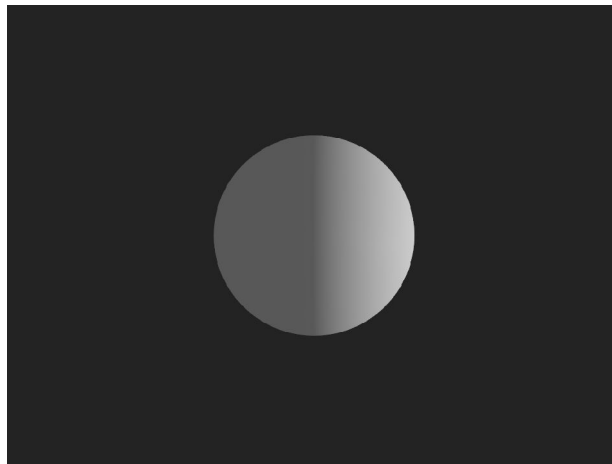


Abbildung 7 mit POV-Ray modelliertes Wassertröpfchen

Die nächsten drei Objekte modellieren kleine Eiskristalle. Wie oben schon erwähnt, ist eine typische Form eines kleinen Eiskristalls das hexagonale Plättchen. Diese Form ist jedoch als Basisform in POV-Ray nicht vorhanden. Deshalb wurde sie einer im Internet frei erhältlichen Vorlagendatei entnommen und zwar aus der auf http://www.f-lohmueller.de/pov_tut/pov__ger.htm [URL 33] zu findenden Datei shapes_lo.inc. In dieser Vorlagendatei ist als Objekt ein Hexagon definiert, welches als Basis für die Vorlage des hexagonalen Plättchen dient. (siehe Abbildung 8)

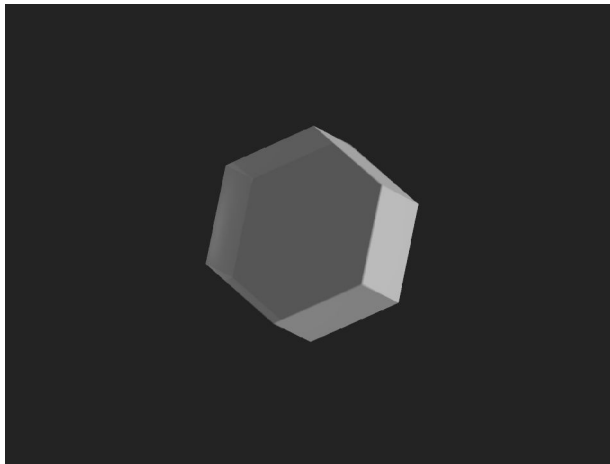


Abbildung 8 mit POVRay modelliertes hexagonales Plättchen

Eine andere einfache Form ist die hexagonale Säule. (siehe Abbildung 9) Diese lässt sich ähnlich einfach modellieren. Sie sieht dann folgendermaßen aus:

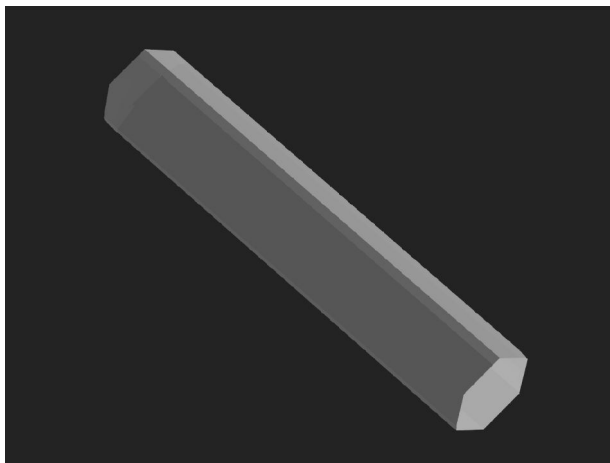


Abbildung 9 mit POVRay modellierte hexagonale Säule

Das letzte Objekt, welches noch in der Vorlage modelliert wurde, ist eine hohle hexagonale Säule, wie man sie zum Beispiel in [Mossop85] sehen kann. Hier ist die Modellierung etwas aufwendiger und deswegen habe ich sie mit Hilfe von Constructive Solid Geometry in POVRay vorgenommen. Zunächst wurde eine halbe hexagonale Säule ähnlich dem vorhergehenden Objekt definiert. Von dieser wurde dann ein sechsseitiger Pyramidenstumpf abgezogen, der auch in shapes_lo.inc enthalten ist. Das erhaltene Objekt

wurde am Nullpunkt gespiegelt und mit dem Originalobjekt zusammengesetzt. (siehe Abbildung 10)

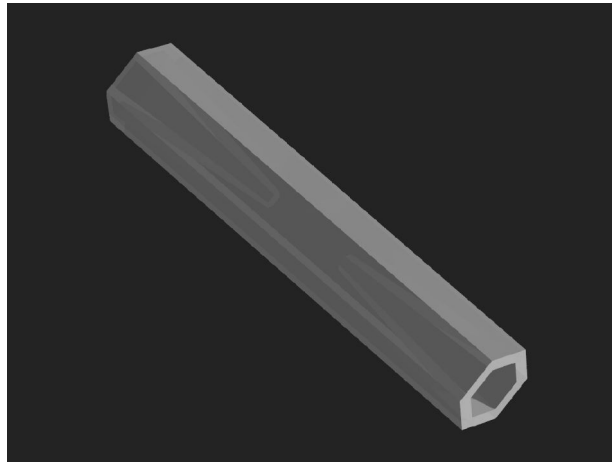


Abbildung 10 mit POV-Ray modellierte hohle hexagonale Säule

2.3.1.4 Aufbau der Szene

Bei dem Aufbau der Szene habe ich mich an dem Paper von Otto et al. [Otto02] orientiert. Hier strömen die Partikel durch eine Kammer mit 40 bzw. 35cm Länge, 20cm Breite und 14mm Tiefe. Die Begrenzungen wurden in der Szene durch Halbebenen modelliert. In der CFDC beobachtete eine CCD-Kamera die vorbei strömenden Partikel auf einen Bereich von 12 x 15 x 2mm. Allerdings besaß diese CCD-Kamera nur eine Auflösung von 320 x 256 Pixel. Dies hatte zur Folge, dass die Größe der beobachteten Objekte nur wenige Pixel betrug und somit keine Details erkennbar waren. Aus dem Grund war auch keine weitere Klassifikation möglich. In diesem Volumen sind die einzelnen Partikel vollkommen zufällig verteilt. Auch ihre Orientierung im Raum ist rein zufällig. Die Beleuchtung erfolgt von der Seite, also senkrecht zu dem Partikelstrom und der Kameraachse. Es wurden auch Testbilder mit Beleuchtung von vorne und Beleuchtung von hinten erstellt. Allerdings zeigte sich später bei deren Auswertung, dass sich hier Eiskristalle und Wassertropfen wesentlich schlechter voneinander unterscheiden ließen. Folglich wurden die meisten Bilder mit seitlicher Beleuchtung erstellt.

2.3.1.5 Nachbearbeitung

Nachdem die Bilder mit POV-Ray fertig gerendert waren, wurde noch eine Nachbearbeitung mit dem Programm Image Analyzer vorgenommen. Betrachtet man die Bilder aus der Veröffentlichung über den vorhergehenden Aufbau einer CFDC, kann man dort ein Rauschen erkennen. Dieses Kamerarauschen sollte auch für die Testbilder modelliert werden. Allerdings lässt sich dieses Rauschen nicht mit POV-Ray erzeugen. Hierfür gibt es kommerzielle Lösungen zum Beispiel als Filter für Adobe Photoshop. Allerdings wurde hierfür das auf <http://meesoft.logicnet.dk/Analyzer/> [URL 22] zu findende Programm Image Analyzer verwendet. Dieses hat den Vorteil, dass es als Freeware erhältlich ist. Zudem besitzt es noch weitere Funktionen wie Histogrammerstellung und ähnliches, die später bei der Entwicklung des Programms hilfreich waren. Mit Hilfe von Image Analyzer wurden dann den mit POV-Ray erstellten Bildern White Noise [URL 26] (in Stärke 7) hinzugefügt und damit der Rauscheffekt berücksichtigt.

2.4 Die Güte der Testbilder

Natürlich stellt sich auch die Frage nach der Güte der Testbilder. Das größte Problem bei der Erstellung war, dass es keine realen Bilder gab, die man als genaue Vorlage für die Testbilder oder sogar als Testbild selbst hätte verwenden können. Der einzige mögliche Ansatz war, dass ich mich an den physikalischen Gegebenheiten orientiere und diese so genau wie möglich modelliere.

Ein Problem der Bilder ist die Tiefenschärfe – POV-Ray stellt die begrenzte Tiefenschärfe einer realen Kamera so dar, dass sie realistisch aussieht, jedoch lassen sich den Parametern in POV-Ray, keine Parameter einer realen Kamera wie zum Beispiel die Blende zuordnen. Zudem sind auch die Kameraparameter der realen Kamera noch unbekannt. Folglich wurden die Bilder mit verschiedenen Werten der Apertur für die Unschärfe berechnet.

Eine weitere Problematik ist die Wahl der Parameter in POV-Ray. Bestimmte Funktionen verfügen über eine große Vielfalt von Parametern, mit denen sich viele Effekte parametrisieren lassen und deren Effekt nicht unbedingt klar ersichtlich ist, wie zum Beispiel die Anzahl der verwendeten Photonen beim Photonmapping. Deswegen wurden dann in vielen Fällen die vorgegebenen Standardparameter verwendet. Jedoch hat sich auch herausgestellt, dass sich wegen der geringen Größe der Eiskristalle und Wassertropfen, die Wirkung der Effekte durch Änderungen an ihren Parametern kaum sichtbar änderte.

Es wurden sehr einfache Formen von Eiskristallen modelliert, da man davon ausgeht, dass die Eiskristalle nur sehr wenig Zeit zum Wachsen haben und somit nur eine geringe Größe mit einem einfachen Formenspektrum – hexagonale Prismen oder Plättchen. Trotzdem können eventuell in der Realität auch andere Formen von Eiskristallen vorkommen.

Sobald man jedoch reale Bilder hat, kann man diese mit vorhandenen Bildern vergleichen. Etwaige Unterschiede sollten sich auch angesichts der vielen Parametrisierungsmöglichkeiten in POV-Ray leicht beheben lassen um dann noch bessere und realistischere Testbilder zu erzeugen. Aufgrund dieser vielen Möglichkeiten wurden auch verschiedene Varianten der Testbilder gerendert, um die entsprechende Bandbreite abzudecken.

Die Bilder wurden auf einem Rechner mit 1.9GHz PIV mit 768MB RAM und Windows XP Professional gerendert. Die Programmversion war POV 3.5.icl.win32 und das Rendern dauerte pro Bild etwa 22 Minuten (1363,7Sekunden).

2.5 Die verschiedenen Testbilder

Zur Veranschaulichung folgt hier eine Überblick über die einzelnen Testbilder mit verschiedenen Objekten in verschiedenen Beleuchtungsvarianten.

2.5.1 Die einzelnen Formen mit Beleuchtung von der Seite



Abbildung 11 Wassertropfen mit Beleuchtung von der Seite



Abbildung 12 hexagonale Plättchen mit Beleuchtung von der Seite

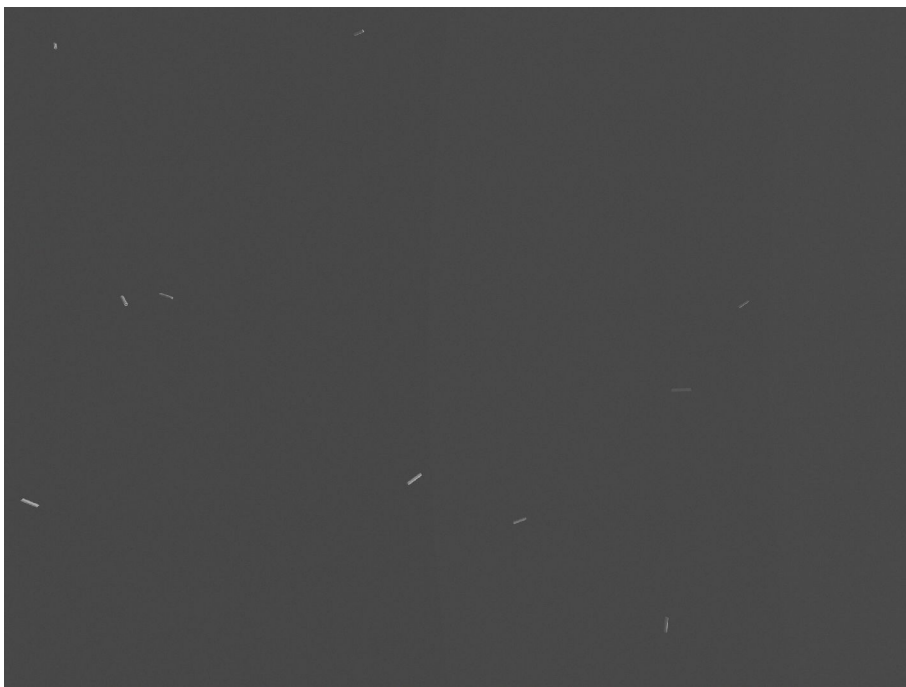


Abbildung 13 hexagonale Säulen mit Beleuchtung von der Seite

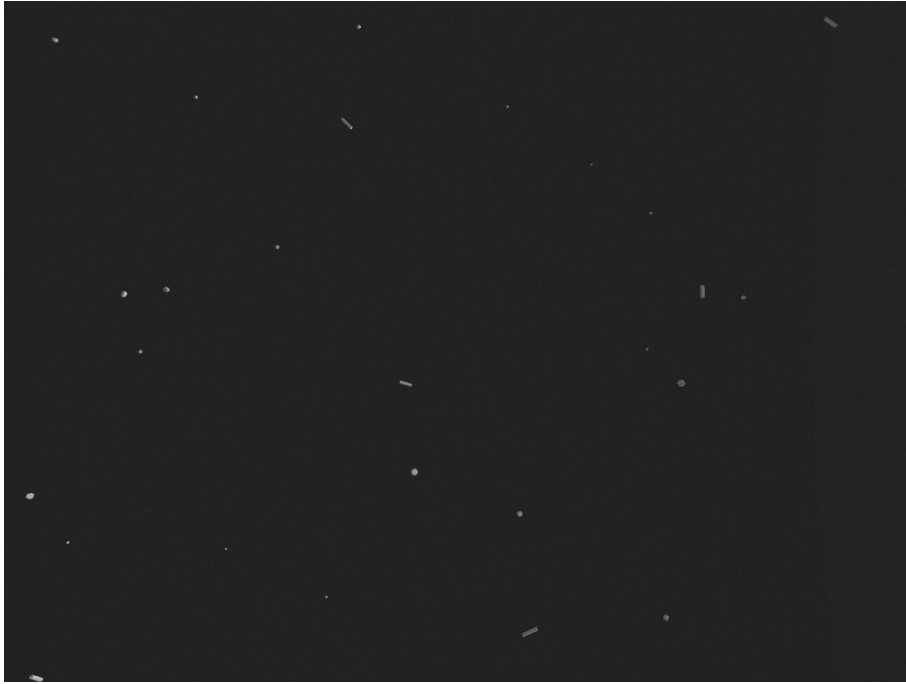


Abbildung 14 alle Formen mit geringerer Tiefenschärfe

2.5.2 Bilder mit Beleuchtung von vorne



Abbildung 15 Tropfen mit frontaler Beleuchtung

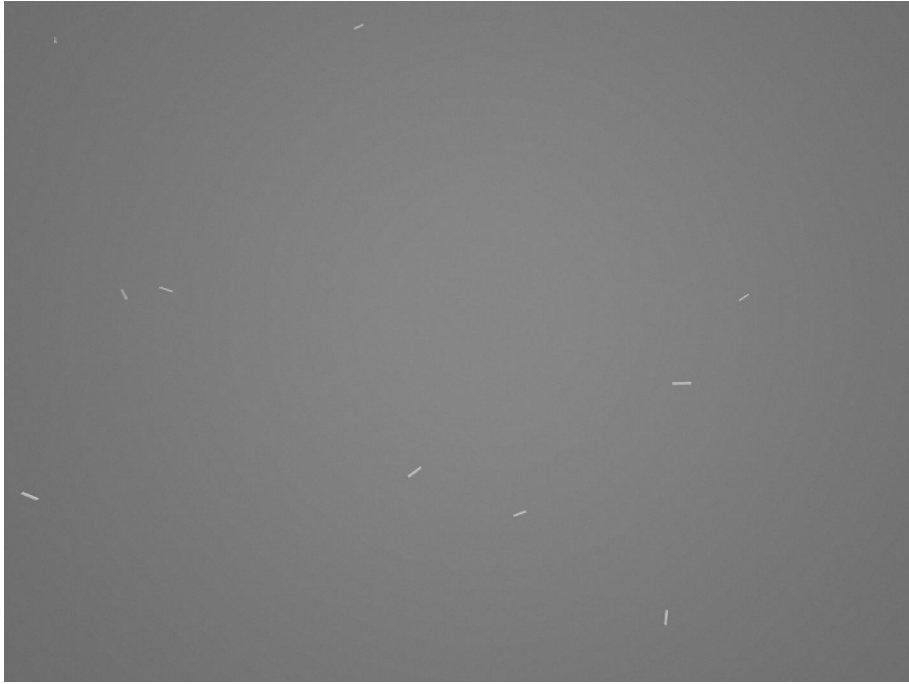


Abbildung 17 hexagonale Säulen mit frontaler Beleuchtung



Abbildung 18 alle Formen mit frontaler Beleuchtung

2.5.3 Bilder mit Beleuchtung von hinten



Abbildung 19 Tropfen mit Hintergrundbeleuchtung



Abbildung 20 hexagonale Plättchen mit Hintergrundbeleuchtung

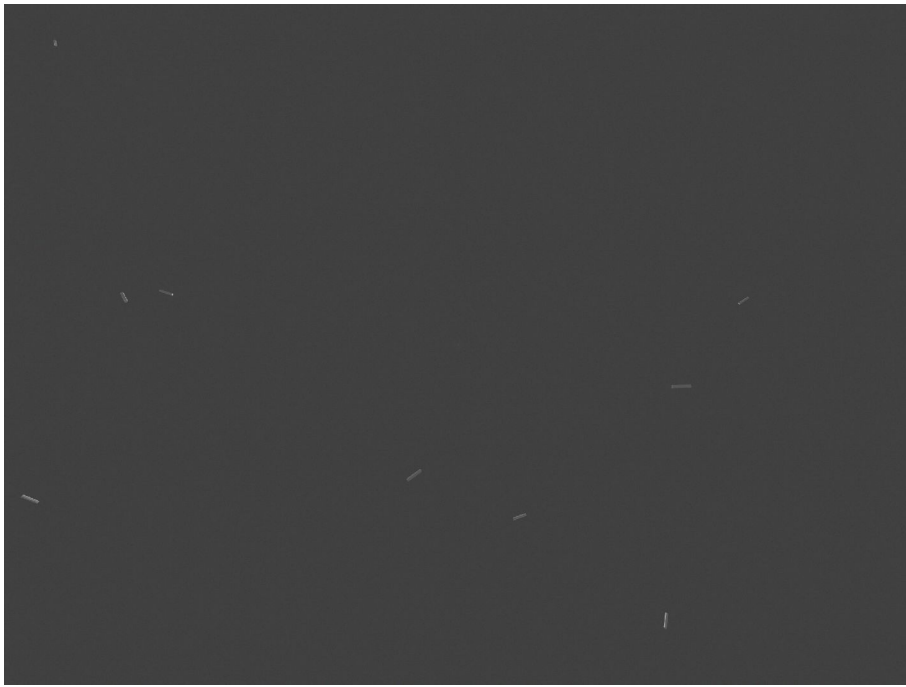


Abbildung 21 hexagonale Säulen mit Hintergrundbeleuchtung

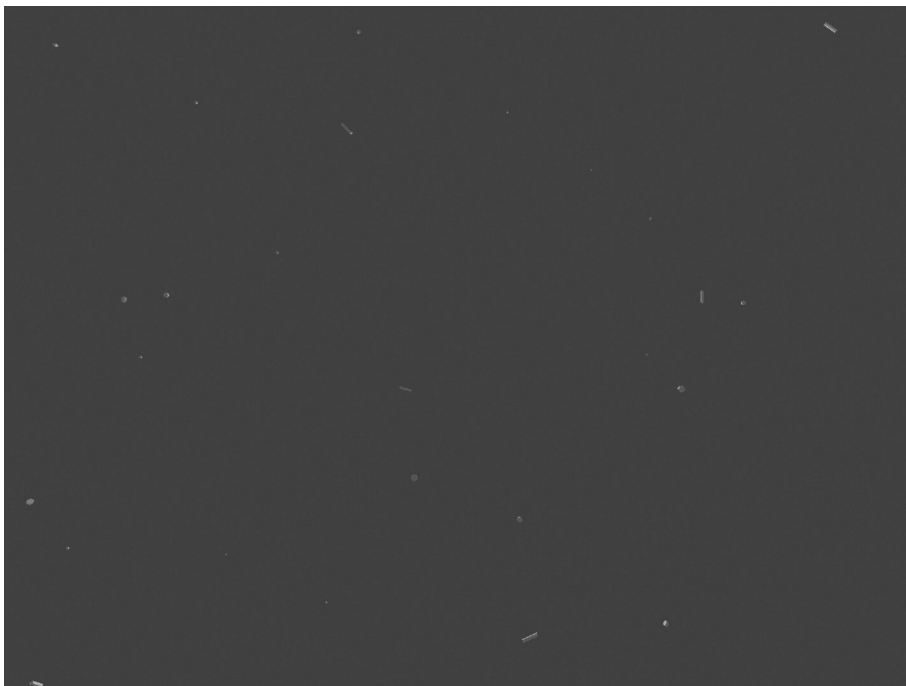


Abbildung 22 alle Formen mit Hintergrundbeleuchtung

3 Algorithmen und Implementierung

Nachdem im vorherigen Kapitel die Testbilder erstellt worden sind, möchte ich hier die Grundlagen und die Entwicklung der Implementierung darstellen und die dabei auf ihre Verwendbarkeit untersuchten Algorithmen beschreiben.

3.1 Grundlagen der Implementierung

3.1.1 Programmiersprache

Es stellte sich die Frage welche Programmiersprache für die Implementierung der Algorithmen gewählt werden soll. Zunächst wurde Perl in Erwägung gezogen, nach näherer Überlegung hat sich Perl jedoch als ungeeignet erwiesen. Perl ist für Skripte und kleinere Anwendungen gedacht und wird nicht kompiliert sondern nur interpretiert. Somit ist die Performance von Perl für Anwendungen, die schnell oder in realtime ablaufen sollen, nicht ausreichend. Auch habe ich keine Bibliotheken für Perl gefunden, welche die für Computer Vision benötigten Funktionen enthalten.

Java ist eine beliebte Programmiersprache, jedoch verfügt auch Java über ähnliche Nachteile wie Perl – es wird auch nur interpretiert und für schnelle oder Real-Time-Anwendungen ist seine Performance nicht ausreichend und somit wurde Java ebenfalls nicht verwendet.

Schließlich habe ich mich für die Verwendung der Programmiersprache C++ entschieden. Im Gegensatz zu Perl und Java verfügt C++ über eine wesentlich bessere Performance. Hiermit lassen sich schnelle Anwendungen und für realtime geeignete Anwendungen programmieren. Zudem gibt es eine umfangreiche Auswahl an Bibliotheken zur Bildbearbeitung und Computervision für C++.

Als Entwicklungsumgebung für die Implementierung des Programms wurde Microsoft Visual C++ 6.0 gewählt. Damit lassen sich auch komfortable Programmoberflächen gestalten.

3.1.2 Verwendung von Bibliotheken zur Objekterkennung

Um nicht sämtliche Funktionen selber programmieren zu müssen, empfiehlt sich die Verwendung vorhandener Bibliotheken. Dazu habe ich im Internet zum Beispiel über die „Computer Vision Homepage“ [URL 47] nach verfügbaren Bibliotheken [URL 5] und Programmen gesucht, welche die entsprechenden Funktionen und Algorithmen enthalten. Einige von diesen, wie KUIM [URL 48] RobotVision CAD [URL 37] SUSAN [URL 46], haben sich aufgrund mangelnder Funktionalität als nicht geeignet erwiesen. Drei Bibliotheken, welche über einen geeigneten Funktionsumfang verfügen, nämlich Gandalf, VXL und OpenCV wurden auf ihre Verwendbarkeit im konkreten Anwendungsfall untersucht. Kriterien hierfür waren Performance, Verfügbarkeit, Vorhandensein der benötigten Funktionen und einer guten Dokumentation. Folgend die Beschreibungen der betrachteten Bibliotheken.

3.1.2.1 Gandalf

Gandalf [URL 13], [URL 14] besteht aus vier einzelnen Programmpaketen. Es gibt zunächst ein Paket mit Basisstrukturen und Routinen wie zum Beispiel Memory Allocation und Error Handling, die von den andern Paketen verwendet werden. Das zweite Paket enthält ein Funktionen für Lineare Algebra mit Matrix- und Vektoroperationen. Das nächste Paket enthält die Datenstruktur und Lowlevel-Bildbearbeitungsfunktionen. Und schließlich gibt es noch das Paket für Bildverarbeitung, welches die Geometrie- und die Computer Vision-Funktionen enthält. Nach Angaben der Entwickler wurde Gandalf so entwickelt, dass es schnell und speichereffizient arbeitet. Es gibt auch kommerzielle Anwendungen im Bereich Erkennung von Fälschungen und „Camera Motion Control“, die mit Gandalf geschrieben wurden. Gandalf wurde in C und nicht in C++ erstellt, um durch einen einfacheren Aufbau Geschwindigkeitsvorteile zu auszunutzen.

3.1.2.2 VXL

Die zweite Bibliothek, die ich in Erwägung gezogen habe, ist VXL – dies steht für „the Vision-something-Libraries“ [URL 51], [URL 53]. Es gibt auch hier ein sehr umfangreiches Angebot an Funktionen. Die „Core Libraries“ enthalten Funktionen für Numerische Algorithmen, einfache Bildverarbeitungsfunktionen, Geometriefunktionen, I/O-Funktionen und selbstverständlich passenden Datenstrukturen. Diese werden ergänzt durch weitere Bibliotheken für Bildverarbeitung, Koordinatensysteme, Kamerageometrie, Stereoverarbeitung, Videomanipulation, Struktur aus Bewegung,

Wahrscheinlichkeitsmodellierung, GUI-Design, Klassifikation und Schätzverfahren und noch viele weitere. VXL ist in ANSI/ISO C++ geschrieben und auf verschiedenen Plattformen lauffähig.

3.1.2.3 Open Source Computer Vision Library

Die dritte Bibliothek, die ich näher betrachtet habe, ist die Intel Open Source Computer Vision Library – kurz OpenCV genannt[URL 15], [URL 17]. OpenCV ist gedacht für realtime Computervision und sie ist vom Funktionsumfang sehr umfangreich – viele Bildbearbeitungs- und Bildverarbeitungsfunktionen, Datenstrukturen für Bilder, Konturfunktionen, Thresholding, Kamerakalibration, Schätzung mit Kalman und Condensation, Histogramm, Gestenerkennung, Stereobasierte Verfahren, Matrixoperationen und noch viele weitere Vorlagen und Funktionen. OpenCV ist ebenfalls in C geschrieben und für verschiedene Plattformen geeignet.

3.1.2.4 Entscheidung für OpenCV

Alle drei Bibliotheken haben gemeinsam, dass sie Open Source Projekte sind, ihre Quellen sind über sourceforge.net verfügbar. Schließlich habe ich mich entschieden, die OpenCV Bibliotheken zu verwenden. Diese bieten ein sehr vielfältiges Angebot an Funktionen, vor allen Dingen auch an Mid- und Highlevelfunktionen, wie zum Beispiel das Finden von Konturen und Berechnen von Momenten. Ein großer Vorteil von OpenCV ist seine gute und leicht verständliche Dokumentation, die wesentlich besser als die von VXL und Gandalf ist. Die Dokumentation gibt es einerseits als Pdf-Datei direkt von Intel [URL 28], allerdings nicht auf den neusten Stand, und zum anderen für die aktuelleren Versionen [URL 36], diesen beiliegend oder direkt online. Das Ganze wird ergänzt durch eine sehr aktive Mailingliste [URL 19], durch die es auch einen guten Support gibt. Zudem gibt es viele gute Beispiele zum Beispiel über [URL 54] auf denen man aufbauen kann.

3.2 Algorithmen und Programmaufbau

Das Programm ist in mehrere Stufen unterteilt, um die einzelnen Verarbeitungsschritte besser voneinander abzugrenzen. Im Überblick gestaltet sich sein Aufbau folgendermaßen. Da die graphische Benutzeroberfläche das erste ist, was dem Benutzer begegnet, bildet sie die erste Stufe. Sie zeigt einen Teil der Parameter an, ermöglicht es ein Bild zur Verarbeitung auszuwählen und diese zu starten.

Die zweite Stufe liest das zu bearbeitende Bild in den Speicher und bereitet es dort für die weitere Verarbeitung vor.

In der dritten Stufe wird das Bild für die eigentliche Verarbeitung vorbereitet. Zunächst wird ein Filter zur Reduzierung des Rauschens angewendet, danach wird das Histogramm des Bildes erstellt und daraus ein passender Wert für den Threshold errechnet. Mit der Durchführung des Thresholdings endet die Bearbeitung in dieser Stufe.

Mit der vierten Stufe erfolgt dann die eigentliche Verarbeitung des Bildes. Zunächst werden die Kanten detektiert und aus ihnen die Konturen der einzelnen Objekte bestimmt. Anhand dieser Konturen werden die Momente der einzelnen Objekte berechnet und damit werden sie dann erkannt und klassifiziert.

3.2.1 Stufe 1: graphische Benutzeroberfläche.

Stufe 1 beschreibt die graphische Benutzeroberfläche (siehe Abbildung 23). Aus wissenschaftlicher Sicht enthält sie keinerlei Besonderheiten. Sie dient nur zur Steuerung des Programms. Man kann ein Bild öffnen und dessen Verarbeitung starten sowie das Programm beenden. Zudem werden die Grenzwerte für die Exzentrizität der Objekte angezeigt und nach Verarbeitung des Bildes auch der berechnete Wert für den Threshold.

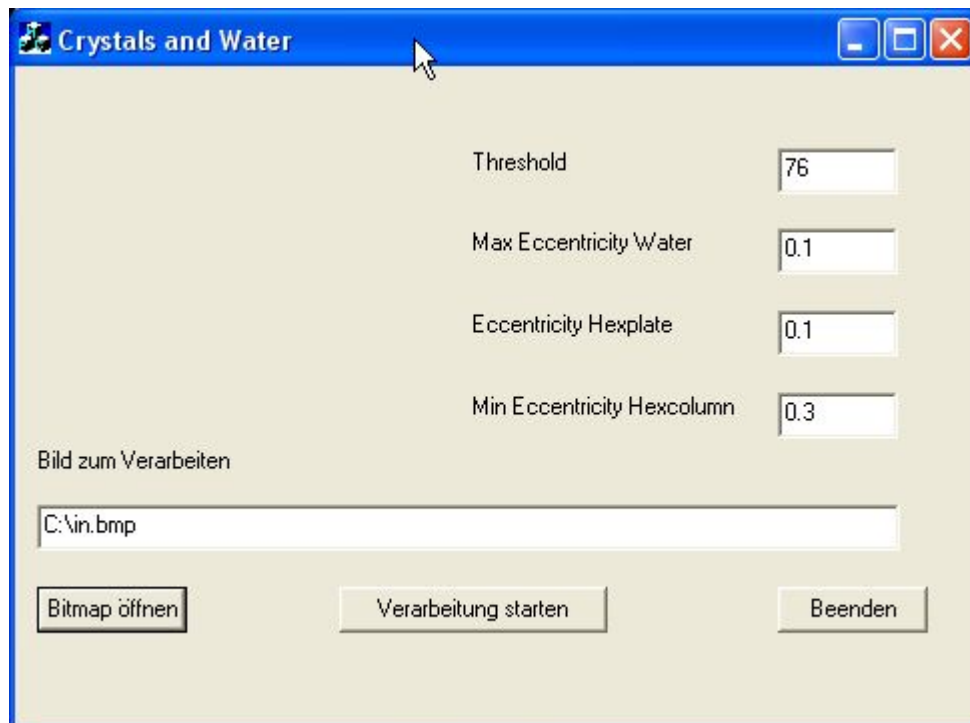


Abbildung 23 Programmfenster

3.2.2 Stufe 2: Einlesen des Bildes als Bitmap und Vorbereiten des Bildes

Startet man aus der ersten Stufe heraus die Bildverarbeitung wird in der zweiten Stufe zunächst das ausgewählte Bild geladen und in einem neuen Fenster angezeigt. Danach wird zur Vorbereitung der Bildverarbeitung noch ein schwarzer Rahmen mit der Breite von 1 Pixel um das Bild gelegt. Es hat sich herausgestellt, dass man fehlerhafte Konturen bekommt, wenn die betrachteten Objekte den Rand des Bildes berühren. Der Grund hierfür ist, dass die später verwendete Funktion `cvFindContours` fehlerhafte Ergebnisse liefert, wenn ein Objekt den Bildrand berührt. Denn statt die komplette Kontur des Objektes zu bestimmen, zersplittert `cvFindContours` die Kontur des betrachteten Objektes in unvollständige Bruchstücke. Diese sind erkennbar als nicht geschlossene Kanten, die für eine weitere Verarbeitung nicht mehr geeignet sind.

Das Ergebnis dieser Operationen wird sowohl als Datei als auch im Bildfenster ausgegeben. Danach wird die eigentliche Bildverarbeitung aufgerufen, die zunächst das Bild in Graustufen konvertiert und dann mit der Vorverarbeitung des Bildes beginnt.

3.2.3 Stufe 3: Vorverarbeitung des Bildes

Die Vorverarbeitung des Bildes läuft folgendermaßen ab. Zunächst wird das Bild mit einem Glättungsfilter bearbeitet um das Rauschen zu entfernen oder zumindest zu reduzieren. Nach Anwendung dieses Glättungsfilters wird wiederum das bearbeitete Bild als Bitmap und im Fenster ausgegeben. Als Glättungsfilter wurden folgende Filter getestet.

3.2.3.1 Mittelwert-Filter

Der einfachste Filter um ein Bild zu glätten, ist ein „Mittelwert-Filter“. Hier wird der Wert eines Pixels durch den Wert des Durchschnittes seiner Nachbarn, inklusive des Pixels selber, ersetzt [URL 44]. Der Mittelwert-Filter ist ein linearer Filter und wird durch eine Faltung berechnet. Diese Faltung wird durch eine Matrix, den sogenannten Filterkern beschrieben. Üblicherweise verwendet man für einen Mittelwert-Filter folgende 3x3-Matrix.

$$\begin{array}{ccc} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{array} \quad (1)$$

Durch Verwendung eines größeren Filterkerns, zum Beispiel 5x5 oder 7x7, erhält man eine stärkere Glättung aber dann wirken sich die Nachteile des Filters noch stärker aus – Objektkanten werden verwischt, da hier die Werte von Hintergrundpixeln und Objektpixeln zu neuen Werten vermischt werden. Ein weiterer Nachteil ist, dass sich ein einzelnes unrepräsentatives Pixel auf eine ganze Region auswirkt.

Das Hauptproblem der Anwendung des Mittelwert-Filters war die zunehmende Unschärfe an den Kanten der Eiskristalle, die dadurch eher runder wurden. Viel schlimmer waren jedoch die Auswirkungen auf die noch kleineren Wassertropfen, deren dunkle Stellen nun noch stärker mit dem Hintergrund verschmolzen.

3.2.3.2 Gauss-Filter

In OpenCV wurde in einem der Beispielprogramme auch die Verwendung eines Gauss-Filters zur Glättung der Bilder vorgeschlagen, der auch getestet wurde. Der Gauss-Filter arbeitet ähnlich wie der Mittelwert-Filter basierend auf Faltung. Er verwendet einen anderen Filterkern, der eine Gauss'sche Glockenkurve und ihre Eigenschaften repräsentiert [URL 43]. Eine Gauss-Verteilung hat in 2D die Form

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Damit lässt sich der Filterkern berechnen – hier ein Beispiel der Größe 5 x 5

$$\frac{1}{273} \begin{array}{ccccc} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{array} \quad (3)$$

Dieser wird ähnlich dem Mittelwert-Filter angewendet. Jedoch wird in diesem Fall ein gewichteter Mittelwert verwendet – Pixel, die näher in der Nachbarschaft des zentralen Pixels sind, werden stärker gewichtet als die weiter entfernten. Deswegen ergibt sich eine weichere Glättung und eine bessere Erhaltung der Karten.

Der Gauss-Filter hat sich beim Testen in diesem Fall nicht bewährt, da sein Einfluss auf das Rauschen ziemlich gering war, gerade die Stellen an denen das Rauschen ein Maximum aufwies, blieben erhalten. Er war folglich für diesen Zweck nicht verwendbar.

3.2.3.3 Medianfilter

Als dritte Variante wurden zur Vermeidung der Nachteile der vorher geprüften Glättungs-Filter zur Reduzierung des Rauschens, der Medianfilter getestet. Im Gegensatz zum Mittelwert-Filter ist der Medianfilter ein nichtlinearer Filter [Jähne02], [URL 45]. Ähnlich wie beim Mittelwert-Filter werden wieder die Pixel in Nachbarschaft des Pixels betrachtet. Aber hier wird der Grauwert eines Pixels, durch den Median der Grauwerte der Pixel in der Nachbarschaft ersetzt. Der Medianfilter hat zum einen den Vorteil, dass er im Gegensatz zu dem Mittelwert-Filter, ein einzelnes unrepräsentatives Pixel den Median nicht wesentlich beeinflusst und zum anderen, da ein bereits vorhandener Wert eines Pixels gewählt wird, werden keine neuen Werte geschaffen und Kanten dadurch besser erhalten.

Es wurden Medianfilter mit Filtermasken der Größe 3 x 3, 5 x 5 und 7 x 7 getestet. Der Medianfilter mit der Größe der Filtermaske von 3 x 3 wurde gewählt und im endgültigen Programm eingesetzt. Mit größeren Filtermasken wird das Rauschen zwar besser entfernt, allerdings wirkt es sich auch stärker auf die Objekte aus. Denn auch hier ist die Wirkung auf die kleinen Wassertropfen am stärksten – sie werden dadurch zum Teil entfernt.

3.2.3.4 Histogramm und Thresholding

Der nächste Schritt in der Vorverarbeitung ist die Berechnung eines Histogramms des Bildes. Daran schließt sich die Wahl des Wertes für den Threshold an. Die Wahl des Wertes für den Threshold ist entscheidend für die weitere Verarbeitung des Bildes. Denn Thresholding dient dazu, Pixel auszublenden, die nicht in einem bestimmten Bereich fallen, da sie zum Beispiel zu dunkel sind oder um ein Graustufenbild in ein Bi-Level- oder Schwarzweißbild umzuwandeln. Für die Wahl des Thresholds gibt es verschiedene Methoden – man kann den Threshold konstant wählen oder einen Bereich für den Threshold vorgeben. Man kann auch einen optimalen Wert berechnen lassen – hierfür gibt es verschiedene Verfahren, die üblicherweise auf dem Histogramm des Bildes arbeiten. Dafür habe ich folgende Varianten betrachtet.

3.2.3.5 Konstante Wahl des Thresholds

Die einfachste Variante ist die manuelle Wahl eines konstanten Thresholds für die zu bearbeiteten Bilder. Dies ist schnell eine schlechte Wahl, da ein Wert für den Threshold für mehrere Bilder kritisch ist – man hat schnell einen unpassenden Wert. Führt man das Thresholding mit einem falschen Wert durch, erhält man schlechte bis gar keine Ergebnisse [URL 23]. Zum Beispiel hatte ich zunächst als Threshold für die seitlich beleuchteten Bilder einen Wert von 78 gewählt. Allerdings wenn ich dann versucht habe, diesen Wert auf ein frontal beleuchtetes Bild anzuwenden, wurde dieses durch Anwendung dieses Thresholds komplett weiß. Der für das frontal beleuchtete Bild passende Threshold von 138 führte bei einem seitlich beleuchteten Bild wiederum dazu, dass ein Teil der Objekte ausgeblendet wurde. Für die automatische Bearbeitung der Bilder ist diese Methode nicht geeignet.

3.2.3.6 Wahl einer Hysterese für den Threshold

Eng damit verwandt ist die Wahl des Thresholds anhand einer Hysterese-Kurve[URL 52] . Hier gibt man zwei Werte vor. Alle Werte, die größer als der höhere Wert sind, werden als zugehörig zur der Klasse der Objekte klassifiziert. Alle Werte, die kleiner als der niedrigere Wert sind, werden als zugehörig zum Hintergrund klassifiziert. Liegt der Wert eines Pixels zwischen diesen Werten, wird das Pixel als zugehörig zum Objekt erkannt, wenn es benachbart zu anderen Pixeln eines Objektes liegt.

Auch dieses Verfahren hat sich als nicht geeignet erwiesen, denn es war wiederum eine manuelle Wahl der oberen und unteren Schranke für den Threshold notwendig, die nicht für alle Bilder geeignet war.

3.2.3.7 Berechnung des Thresholds mit dem Otsu-Verfahren

Eine geeignete und aufwendigere Methode zur Berechnung des passenden Thresholds für ein Bild ist das Otsu-Verfahren [URL 23]. Diese Methode setzt voraus, dass im Histogramm eine bimodale Verteilung der Grauwerte vorhanden ist [URL 52], das heißt, es gibt im Histogramm zwei Peaks – der eine enthält die Pixel des Hintergrunds und der andere die Pixel der Objekte. Diese Voraussetzung ist hier im Anwendungsfall gegeben, wobei der Peak der Hintergrundpixel wesentlich stärker ausgeprägt ist als der Peak der Objektpixel.

Die Berechnung läuft dann wie folgt. Ein Maß für die Homogenität einer Region, also auch eines Peaks, ist die Varianz - eine homogene Region hat eine geringe Varianz. Man

berechnet den Threshold, indem die Varianz innerhalb der beiden Klassen minimiert wird, die durch den Threshold getrennt werden. (Die Varianz ist das Quadrat der Standardabweichung)

Genauer ausgedrückt berechnet man zunächst

$$\sigma_g^2 = \text{Varianz gesamt} \quad (4)$$

und

$$\sigma_i^2 = \text{Varianz innerhalb einer Klasse} \quad (5)$$

und die Variation der Mittelwerte gegenüber dem gesamten Mittel der Intensität für jede Klasse definiert die Varianz zwischen den Klassen

$$\sigma_z^2 = \text{Varianz zwischen den Klassen} \quad (6)$$

man minimiert die Varianz in einer Klasse zu der Varianz gesamt

$$\eta(t) = \frac{\sigma_z^2}{\sigma_i^2} \quad (7)$$

Der Wert von t der den kleinsten Wert von $\eta(t)$ erzeugt ist der passende Threshold σ_b^2 und den Mittelwert μ_g berechnet man aus dem Bild. Die Varianz zwischen den Klassen berechnet sich dann

$$\sigma_z^2 = w_0 w_1 (\mu_0 - \mu_1)^2 \quad (8)$$

mit

$$w_0 = \sum_{i=0}^t p_i \quad (9)$$

und

$$w_1 = 1 - w_0 \quad (10)$$

p_i ist der Wert im Histogramm an Stelle i , dividiert durch die Gesamtzahl der Pixel

$$\mu_0 = \frac{\mu_t}{w_0} \quad (11)$$

$$\mu_1 = \frac{\mu_g - \mu_t}{1 - w_0} \quad (12)$$

$$\mu_t = \sum_{i=0}^t i * p_i \quad (13)$$

$\eta(t)$ wird für alle Werte von t berechnet und das t mit dem kleinsten $\eta(t)$ ist der optimale Threshold.

Beim Testen wurde festgestellt, dass die Otsu-Methode für einen Teil der Bilder versagt, nämlich dann, wenn die Fläche, die von Wassertropfen und Eiskristallen belegt wird, also die Menge der hellen Objekte, zu klein ist. Dann wird für den Threshold ein Wert berechnet, der zu gering ist und im Bereich des Peaks des Rauschens liegt. Somit liefert in diesem Fall die Anwendung des Thresholds ein falsches Ergebnis, da das Rauschen nicht ausgeblendet wird.

3.2.3.8 Alternative Methode zur Berechnung des Thresholds

Um diesen Fehler abzufangen, wird der Threshold mit dem Durchschnitt des Grauwertes des Bildes verglichen. Bei einer inkorrekten Berechnung mit der Otsu-Methode liegt dieser nämlich unter dem Durchschnitt der Grauwerte des Bildes. Tritt dieser Fall ein, wird eine zweite Methode angewendet, in Anlehnung an die Methode von Christine Antoine, Mike D. Lloyd and Jacques Antoine [URL 1]. Dort wird um den Threshold zu berechnen, entweder das Minimum zwischen linkem Maximum, also nach dem Peak der Hintergrundpixel, und rechtem Maximum, dem Peak der Objektpixel, gesucht oder, wenn sich dies nicht finden lässt, da zum Beispiel kein eindeutiger Peak der Objektpixel existiert, ein passender Wendepunkt der Kurve gesucht.

Der mit Otsu gefundene Threshold wird mit dem Durchschnitt verglichen, ist er größer als der Durchschnitt, wird er für das Thresholding verwendet. Ist er kleiner als der Durchschnitt, wird das Maximum im Peak des Hintergrundrauschens genommen und das nächste lokale Minimum als Wert für den Threshold.

Mit diesem Wert wird dann das Thresholding durchgeführt und diese Vorgehensweise hat sich für die Testbilder gut bewährt. Das Hintergrundrauschen wurde komplett ausgeblendet und sämtliche Objekte wurden gut erhalten. Deshalb wurden dann auch keine weiteren Verfahren mehr, wie zum Beispiel eine lokale Wahl oder adaptive Berechnung des Thresholds [URL 23], geprüft.

3.2.3.9 Weitere Effekte des Thresholdings

Das Thresholding des Bildes hat noch weitere Auswirkungen. Dadurch werden auch die Helligkeitsunterschiede in den Objekten, die Objektkanten anzeigen können, ausgeblendet. In Versuchen hatte sich herausgestellt, dass unter Umständen diese Innenkanten eines Objektes dazu führen können, dass es in mehrere Objekte aufgeteilt werden kann. Dabei gehen Informationen über die innere Struktur des Objektes verloren, zum Beispiel können jetzt hexagonale Prismen und hohle hexagonale Prismen nicht mehr voneinander

unterschieden werden. Sowohl nach Anwendung des Medianfilters als auch nach Anwendung des Thresholds wird das Bild am Bildschirm angezeigt und als Bitmap gespeichert.

3.2.4 Stufe 4A: Bildverarbeitung

Nachdem das Bild vorbereitet und vorverarbeitet wurde, geschieht nun die eigentliche Bildverarbeitung. Zunächst erfolgt die Detektion der Kanten des Bildes mit Hilfe des Canny-Filters .

Daran schließt sich das Trennen der einzelnen Objekte anhand der gefundenen Kanten und das Finden ihrer Konturen an. Darauf erfolgt die Berechnung der Momente der einzelnen Objekte. Mittels dieser geschieht dann die Identifikation der Objekte und schließlich die Ausgabe der Ergebnisse.

3.2.4.1 Verwendung des Sobel-Filters zur Detektion der Kanten

Kanten sind die Grenzen von Regionen mit unterschiedlicher Helligkeit oder Farbe. Diese trennen oft verschiedene Objekte eines Bildes voneinander. Ein einfaches Verfahren um Kanten zu erkennen ist der sogenannte Sobel-Operator [URL 2]. Dieser basiert auf Faltung mit zwei 3x3Filterkernen

$$G_x \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (14)$$

$$G_y \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (15)$$

Der Filterkern G_x reagiert auf horizontale Kanten und G_y reagiert auf vertikale Kanten. Beide Filterkerne können einzeln auf die Eingabe angewendet werden, um Kanten in x-Richtung oder y-Richtung zu finden. Man wendet sie gemeinsam an, um die absolute Größe und Orientierung des Gradienten an jedem Punkt des Bildes zu errechnen. Die Größe des Gradienten ist gegeben durch

$$|G| = \sqrt{G_x^2 + G_y^2} = |G_x| + |G_y| \quad (16)$$

und die Orientierung des Gradienten lässt sich beschreiben durch

$$\theta = \arctan(G_y/G_x) \quad (17)$$

Allerdings reagiert der Sobel-Operator empfindlich auf Rauschen und wurde aus diesem Grund nicht verwendet.

3.2.4.2 Anwendung des Canny-Filters zur Detektion der Kanten

In den Beispielen von OpenCV wird meistens der Canny-Filter zur Detektion der Kanten verwendet [URL 9], [URL 28]. Dieser arbeitet mit einem mehrstufigen Verfahren. Als Eingabe wird hier ein Graustufenbild verwendet. Zunächst wird das Bild mit einem Gauss-Filter geglättet. Danach wird das geglättete Bild differenziert bezüglich der Richtungen x und y. Dies geschieht unter Verwendung des bereits beschriebenen Sobel-Operators. Aus den Gradientenwerten für x und y wird die Größe und der Winkel des Gradienten berechnet. Nach der Berechnung des Gradienten kann man die Kanten dort finden, wo die Größe des Gradienten ein lokales Maximum aufweist. Dies geschieht durch die Unterdrückung von „Nicht-Maxima“, also Punkte, die kein lokales Maximum sind. Jedoch müssen vor allem „Nicht-Maxima“ unterdrückt werden, die senkrecht zu einer Kante stehen, und nicht die „Nicht-Maxima“ in Richtung der Kante, um die Kontinuität einer Kante zu gewährleisten. Der Algorithmus startet damit, dass er den Winkel des Gradienten auf einen der vier Sektoren (Abbildung 24) reduziert.

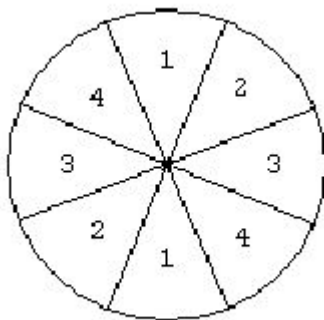


Abbildung 24 Sektoren für Canny-Filter
(aus Anleitung OpenCV [URL 28])

Der Algorithmus übergibt die 3x3-Nachbarschaft durch das Größen-Array. An jedem Punkt wird das Element in der Mitte entlang der durch den Gradienten gegebenen Linie mit seinen zwei Nachbarn verglichen. Ist der Wert in der Mitte nicht größer als seine Nachbarn, ist er ein „Nicht-Maximum“ und wird unterdrückt.

Schließlich verwendet der Canny-Operator ein sogenanntes „Hysterese-Thresholding“, um Kanten zu finden. Es hat den Vorteil, dass Kanten eher durchgängig sind, da sonst Kanten durch ein Pixel mit einem zu geringen Wert unterbrochen werden können.

Ein Segment einer Linie wird sofort akzeptiert, wenn der Wert über dem oberen Threshold liegt, und sofort verworfen, wenn sein Wert unter dem unteren Threshold liegt. Segmente/Punkte, welche zwischen den beiden Thresholdwerten liegen, werden dann als zugehörig betrachtet, wenn sie mit einer Linie verbunden sind.

Der Canny-Filter hat sich für diese Anwendung gut bewährt - die Kanten der Objekte wurden zuverlässig erkannt. Somit habe ich keine weiteren Verfahren mehr getestet. Nach Anwendung des Canny-Filters wird das Bild sowohl als Datei ausgegeben als auch am Bildschirm angezeigt.

3.2.5 Stufe 4B: Schritt 2 Identifikation und Klassifikation

3.2.5.1 Trennen der Objekte und Finden der Konturen

Einer der wesentlichsten Schritte ist das Trennen der einzelnen Objekte und das Finden der Konturen. OpenCV bietet hierfür die Funktion `cvFindContours`[URL 28], welches die einzelnen Konturen findet und erkennt. Für diese Funktion gibt es zwei Möglichkeiten Optionen einzustellen. Zum einen welche Konturen berücksichtigt werden und in welcher Hierarchie sie sortiert sind – nur äußere oder auch innere Konturen und wie diese sortiert werden sollen. Hier wurden nur die äußeren Konturen bestimmt, da es wegen der Anwendung des Thresholds keine inneren mehr geben kann. Die anderen Optionen sind für die polygonale Approximation der Kontur.

3.2.5.2 Einzelne Konturen finden

Die gefundenen Kanten liegen noch in Punktdarstellung vor - für die weitere Verarbeitung wird jedoch eine Vektordarstellung benötigt. Die Funktion `cvFindContours` in OpenCV berechnet diese Darstellung mit folgendem Algorithmus nach Suzuki und Abe [URL 28].

Der Algorithmus betrachtet das Binärbild zeilenweise. Findet er einen Punkt, der zu einer neuen Kante/Rand gehört, beginnt eine Prozedur, welche der Kante folgt und den Umriss im Kettencode (Freeman) speichert. Die besuchten Pixel werden dabei mit bestimmten positiven oder negativen Werten gekennzeichnet. Ist der rechte Nachbar des betrachteten Pixels ein Pixel mit Wert 0 und ist zur gleichen Zeit das Pixel mit Wert 0 auf der rechten Seite der Kante, wird das Pixel auf der Kante mit einem negativen Wert markiert. Im anderen Fall wird der Punkt mit dem gleichen aber positiven Wert markiert, so als ob der Punkt noch nicht besucht worden wäre. Man kann dies leicht bestimmen, da eine Kante sich selber kreuzen oder eine andere Kante berühren kann.

3.2.5.3 Freeman Chain Code

Für die Beschreibung der Konturen gibt es verschiedene Methoden. Eine einfache Methode Konturen darzustellen ist der „Freeman chain code“ [URL 12]. Anstelle von Koordinatenangaben der Pixel werden hier die Nachbarn eines Pixels P auf folgende Art und Weise nummeriert:

$$\begin{array}{ccc} 3 & 2 & 1 \\ 4 & P & 0 \\ 5 & 6 & 7 \end{array} \quad (18)$$

Eine Folge von Punkten in 8er-Nachbarschaft kann somit durch die Angabe der Koordinaten des Startpixels, gefolgt von den Codes, die den Ort des folgenden Pixels relativ zum vorhergehenden Pixel angeben.

3.2.5.4 Polygonale Approximation nach Teh-Chin

Eine andere Möglichkeit ist die Darstellung durch eine polygonale Approximation. Es gibt verschiedene Algorithmen, die diese aus dem Kettencode berechnen. Die Idee dahinter ist das Finden von markanten Punkten im Kettencode. Ein Beispiel hierfür ist die „Teh-Chin chain approximation“ [URL 28]. Hierbei kann man zwei unterschiedliche Beschreibungen von Krümmungen verwenden.

Eine sehr einfache Variante ist die L1-Krümmung. Sie berechnet den Unterschied aufeinander folgender Kettencodes

$$c_i^{(1)} = ((f_i - f_{i-1} + 4) \bmod 8) - 4 \quad (19)$$

Ein Wert von 0 entspricht hier einer geraden Linie und ein Wert von 4 einem scharfen Winkel – also einer Richtungsänderung in die entgegengesetzte Richtung.

Eine etwas komplexere Variante ist die Berechnung der k-Kosinus Krümmung. Hier wird für ein Pixel (x_i, y_i) die Nachbarschaft im Radius m_i ausgewählt. Der Radius m_i kann entweder konstant sein oder für jeden Punkt neu berechnet werden. Für alle Paare (x_{i-k}, y_{i-k}) und (x_{i+k}, y_{i+k}) mit $k = 1 \dots m$ wird berechnet

$$c_{ik} = \frac{(a_{ik} \cdot b_{ik})}{|a_{ik}| |b_{ik}|} = \cos(a_{ik}, b_{ik}) \quad (20)$$

mit

$$a_{ik} = (x_{i-k} - x_i, Y_{i-k} - y_i) \quad (21)$$

und

$$b_{ik} = (x_{i+k} - x_i, Y_{i+k} - y_i) \quad (22)$$

Dann wird ein Index h_i gesucht, so dass

$$c_{im} < c_{im-1} < \dots < c_{ih_i} \geq c_{ih_i-1} \quad (23)$$

Der Wert von c_{ih_i} betrachtet man als Wert der Krümmung am Punkt i . Dieser Wert schwankt zwischen dem Wert -1 für eine gerade Linie und +1 für die Richtungsänderung genau in die entgegengesetzte Richtung.

Bei Verwendung der L1-Krümmung arbeitet der Teh-Chin-Algorithmus wie folgt. Es erfolgen mehrere Durchgänge über die gegebene Kurve und bei jedem Durchgang werden einige Punkte entfernt. Zuerst werden alle Punkte mit $c_i^{(1)}=0$ entfernt und für alle anderen Punkte $c_i^{(1)}$ und m_i berechnet. Danach erfolgt eine Unterdrückung der Nicht-Maxima, es werden alle Punkte entfernt für die die Bedingung gilt, dass für $c_i^{(1)}$ das Maß h_i auf m_i gesetzt wird.

Am Ende werden Paare von zwei aufeinander folgenden Punkten durch einen einzigen Punkt ersetzt und Gruppen von drei oder mehr aufeinander folgenden Punkten durch den ersten und den letzten Punkt dieser Gruppe.

Die Berechnung unter Verwendung der k-Kosinus-Krümmung erfolgt analog.

Für die nachfolgende Berechnung hat sich die Auswahl der polygonalen Approximation nach Teh-Chin unter Verwendung der k-Kosinus-Krümmung als beste Variante erwiesen.

3.2.5.5 Verwendung von Ellipse-Fitting

Ein grundlegendes Verfahren der Bildverarbeitung ist das Ellipse-Fitting. Hier wird eine Ellipse an das zu untersuchende Objekt angepasst. Anhand der Eigenschaften dieser Ellipse wie Größe, Orientierung und ihre Radien lassen sich Aussagen über das Objekt treffen.

Somit war mein erster Ansatz um eine Klassifikation der Objekte in Eis und Wasser vorzunehmen die Verwendung von Ellipse-Fitting. Die Idee dahinter war, dass eine Ellipse, welche an einen perfekt runden Wassertropfen angepasst wird, sich als Kreis darstellt – ihre beiden Radien werden nahezu gleich sein. Dagegen wird eine Ellipse, welche an einem eckigen Eiskristall angepasst wird, eher einer normalen Ellipse als einem Kreis entsprechen, das heißt die beiden Radien werden unterschiedlich sein.

In Versuchen hat sich dann aber gezeigt, dass es auch möglich ist an einem Eiskristall eine Ellipse anzupassen, welche die Form eines Kreises hat. Somit hat sich dieses Verfahren als nicht geeignet zur Klassifikation von Eis und Wasser erwiesen.

3.2.5.6 Verwendung von Momenten zur Beschreibung der Form eines Objektes.

In der Computervision gibt es eine ganze Reihe unterschiedlich komplexer Verfahren zur Erkennung von Objekten, wie Neuronale Netze[URL 3], [URL 4], [URL 10], lineare Objektklassenmodelle [URL 24], Spherical Harmonics [Funkhouser03], Active Appearance Modelle[URL 16], [URL 27] und Momente [URL 25] . Dennoch möchte ich als nächstes Verfahren zur Beschreibung der Form von Objekten, die Momente der Objekte betrachten, da sie für mich eine relativ unkomplizierte und übersichtliche Lösung darstellten. Die Eigenschaften der Form eines Objektes lassen sich über dessen Momente [URL 39] beschreiben. Momente sind abhängig von der Funktion der Grauwerte $f(x,y)$ eines Objektes definiert. Somit ist ein Moment der Ordnung $p + q$ definiert als:

$$m_{p,q} = \iint x^p * y^q f(x, y) dx dy \quad (24)$$

Hat man statt einem Grauwertbild ein Binärbild erhält man stattdessen

$$f(x, y) = b(x, y) = \left\{ \begin{array}{l} 1 \text{ Punkt gehört zu Objekt} \\ 0 \text{ Punkt gehört nicht zu Objekt} \end{array} \right\} \quad (25)$$

Somit ergibt sich das Moment der nullten Ordnung als

$$m_{0,0} = \iint b(x, y) dx dy \quad (26)$$

Dieses Moment beschreibt die Fläche des Objektes. Für die Flächen-Momente der ersten Ordnung erhält man somit:

$$m_{1,0} = \iint x b(x, y) dx dy \quad (27)$$

$$m_{0,1} = \iint y b(x, y) dx dy \quad (28)$$

Mit Hilfe der Momente nullter und erster Ordnung lässt sich dann der Schwerpunkt des Objektes berechnen. Für dessen Koordinaten gelten:

$$x_c = \frac{m_{1,0}}{m_{0,0}} \quad (29)$$

$$y_c = \frac{m_{0,1}}{m_{0,0}} \quad (30)$$

Aus den Flächen-Momenten $m_{p,q}$ und dem Schwerpunkt (x_c, y_c) lassen sich die Zentral-Momente $cm_{p,q}$ ableiten. Sie errechnen sich aus den Flächen-Momenten, indem diese durch den Schwerpunkt reduziert werden.

$$cm_{p,q} = \iint (x - x_c)^p * (y - y_c)^q f(x, y) dx dy \quad (31)$$

Das Zentral-Moment nullter Ordnung entspricht dem Flächen-Moment nullter Ordnung.

$$m_{0,0} = cm_{0,0} \quad (32)$$

Somit ergibt sich aber auch für die Zentral-Momente erster Ordnung

$$cm_{1,0} = cm_{01} = 0 \quad (33)$$

Die Zentral-Momente sind invariant zu Translationen des Objektes. Allerdings sind die Zentral-Momente noch abhängig von der Größe des Objektes. Deswegen kann man aus den Zentral-Momenten die normalisierten Zentral-Momente $ncm_{p,q}$ berechnen, indem diese mit der Fläche des Objektes skaliert werden.

$$ncm_{p,q} = \frac{cm_{p,q}}{m_{0,0}^{\frac{p+q}{2}+1}} \quad (34)$$

Aus den Zentral-Momenten zweiter Ordnung lässt sich dann die Exzentrizität [[Jähne02] ecc des Objektes berechnen.

$$ecc = \frac{(cm_{2,0} - cm_{0,2})^2 + 4cm_{1,1}^2}{(cm_{2,0} + cm_{0,2})^2} \quad (35)$$

Die Exzentrizität hat einen Wertebereich von 0 bis 1, je runder das Objekt um so geringer ist sie. Ein Kreis hat eine Exzentrizität von 0 – dagegen hat ein lang gestrecktes linienförmiges Objekt eine Exzentrizität nahe 1.

Es gibt noch Momente höherer Ordnung und je höher die Ordnung der Momente umso präziser beschreiben sie die Form des Objektes. Es gibt auch noch weitere Momente wie zum Beispiel Legendre-Momente oder Hu-Momente, welche die Form noch weitergehender beschreiben und auch oft zum Beispiel zur Schriftzeichenerkennung verwendet werden. [URL 25]. Im vorliegenden Fall war dies jedoch nicht notwendig.

Somit erfolgt im nächsten Schritt das Berechnen der Momente der einzelnen Konturen. Es werden folgende Momente berechnet, zunächst das Flächenmoment nullter Ordnung, dann die Zentral-Momente zweiter Ordnung und schließlich die normalisierten Zentral-Momente zweiter und dritter Ordnung. Aus diesen wird dann die Exzentrizität des Objektes errechnet

3.2.5.7 Objekterkennung

An die Berechnung der Momente schließt sich die eigentlich Objekterkennung an. Die Objekte werden anhand ihrer Exzentrizität und des Flächen-Moments nullter Ordnung unterschieden. Die Grenzwerte hierfür habe ich empirisch bestimmt. Die geringste Exzentrizität – nämlich unter 0,1, besitzen die Wassertropfen, im Bereich von 0,1 bis 0,3 sind die hexagonalen Plättchen angesiedelt und alles was eine Exzentrizität größer als 0,3 hat wird als hexagonales Prisma identifiziert. Da ein kleiner Teil der hexagonalen

Plättchen eine zu geringe Exzentrizität aufweisen kann und manche Tröpfchen eine zu hohe Exzentrizität aufweisen, wird zusätzlich noch das Flächenmoment nullter Ordnung als zusätzliches Kriterium betrachtet – dies wegen den physikalischen Gegebenheiten da Eiskristalle größer als die Tröpfchen sind. Wird dieses Kriterium für die letztendliche Entscheidung verwendet, bekommt das identifizierte Partikel noch einen zusätzlichen Vermerk – als un rundes Tröpfchen oder als rundes hexagonales Plättchen.

Sämtliche berechneten Ergebnisse werden in einer Textdatei ausgegeben, sie enthält die Objekt-Nr, Anzahl der Konturpunkte, Exzentrizität, Flächen-Moment nullter Ordnung, Zentral-Momente bis dritte Ordnung, die normalisierten Zentral-Momente bis dritte Ordnung, Produkt aus Fläche und Exzentrizität und schließlich die Art des identifizierten Partikels – Wassertropfen, unrunder Wassertropfen, rundes hexagonales Plättchen, hexagonales Plättchen und oder hexagonale Säule.

Schließlich werden die erkannten Objekte zur Kontrolle für die Ausgabe noch markiert, indem ihre Konturen mit verschiedenen Farben im Originalbild nachgezeichnet werden. Die Konturen der erkannten Wassertropfen werden in rot nachgezeichnet, die Konturen der erkannten hexagonalen Plättchen in grün gezeichnet und schließlich die Konturen der hexagonalen Prismen in blau.

Bei Wassertropfen und hexagonalen Plättchen, bei denen das Flächenmoment nullter Ordnung die endgültige Entscheidung mitverursacht, wird ein dunklerer Farbton verwendet. Das ganze wird dann auch als Bitmap ausgegeben. So kann man in der Ergebnisdatei visuell nachvollziehen, als was welches Partikel klassifiziert worden ist.

4 Ergebnisse und Evaluierung

Nachdem das Verfahren zur Klassifikation der Partikel im vorhergehenden Kapitel entwickelt wurde, wird jetzt das Verfahren unter verschiedenen Bedingungen untersucht. Die Beleuchtung erweist sich als ein entscheidender Faktor für Objekterkennung und Klassifikation. Zusätzlich zu der in [Otto02] verwendeten seitlichen Beleuchtung habe ich getestet, ob sich mit Frontalbeleuchtung oder Hintergrundbeleuchtung bessere Ergebnisse erzielen lassen. Dafür wurden drei Szenen erstellt, die erste zeigte nur Wassertropfen, die zweite nur hexagonale Plättchen, und die dritte nur hexagonale Säulen. Die Anordnung der Objekte war jeweils gleich, nur die Beleuchtung wurde variiert. Die Implementierung lieferte dann folgende Ergebnisse.

4.1 Frontalbeleuchtung

Für die frontal beleuchteten künstlichen Testbilder war die Lichtquelle direkt über der Kamera angebracht.

Frontal beleuchtet werden die Wassertropfen eindeutig identifiziert. Ihre Exzentrizität liegt immer unter dem von mir empirisch ausgewählten Grenzwert von 0,1. Die maximale Exzentrizität eines Wassertropfens beträgt 0,097161. Die Größe der Wassertropfen wurde nicht für diese Entscheidung benötigt.

Problematischer gestaltet sich dagegen die Identifikation der hexagonalen Plättchen. Diese werden nicht eindeutig identifiziert. Bei dem verwendeten Testbild wurden nur 4 von 10 abgebildeten Plättchen über die Exzentrizität erkannt. Weitere 5 wurden als runde hexagonale Plättchen identifiziert, das heißt von der Exzentrizität her entsprach der Wert einem Wassertropfen und nur die Größe verursachte schließlich die Identifikation als hexagonales Plättchen. Das letzte der 10 Objekte wurde ebenfalls auf Grund der Größe als unrunder Wassertropfen identifiziert.

Die Identifikation der hexagonale Säulen verläuft besser – es werden auf dem Testbild 9 von 10 eindeutig identifiziert. Jedoch wurde eine in zwei Teile aufgeteilt, die als unrunder Wassertropfen erkannt wurden. Betrachtet man das Testbild genauer, kann man sehen, dass dieser Fehler darin begründet ist, dass der Kontrast zwischen Objekt und Hintergrund zu gering ist und somit ein Teil des Objektes als Hintergrund betrachtet wird. Man erhält dadurch zwei unzusammenhängende Bruchstücke, die jeweils der Größe eines Wassertropfens entsprechen.

Als Schlussfolgerung ergibt sich daraus, dass die Frontalbeleuchtung keinen besonderen Vorteil bei der Betrachtung der Objekte bringt. Zwischen den hexagonalen Plättchen und

den Tropfen kommt es zu einer starken Überschneidung bezüglich der Exzentrizität. Man erhält mit Frontalbeleuchtung zudem ein helles Bild, welches einen zu geringen Kontrast zwischen Objekten und Hintergrund aufweist. Das kann eine fehlerhafte Identifikation der Objekte verursachen und damit auch eine fehlerhafte Klassifikation in Wasser und Eis. Deswegen wurden auch keine weiteren Testbilder mehr mit Frontalbeleuchtung gerendert.

4.2 Hintergrundbeleuchtung

Als nächste Variante habe ich eine Hintergrundbeleuchtung der Szene getestet. Die Lichtquelle gegenüber der Kamera auf der gleichen Höhe angebracht. Auch hier wurden die Wassertropfen eindeutig identifiziert. Ihre Exzentrizität war immer eindeutig unter Grenzwert 0,1 – denn die maximale Exzentrizität eines Wassertropfens beträgt 0,079236. Allerdings findet sich auf dem Testbild auch ein Artefakt. Dieses erweist sich bei genauer Betrachtung des Testbildes als eine „Bildstörung“, die nicht mit der Beleuchtung zusammenhängen muss.

Noch problematischer als bei der Frontalbeleuchtung gestaltet sich dagegen die Identifikation der hexagonalen Plättchen. Sie wurden nicht eindeutig identifiziert. Von den betrachteten hexagonalen Plättchen werden nur 2 von 10 über ihre Exzentrizität erkannt. Die restlichen 8 von 10 als runde hexagonale Plättchen identifiziert – also wieder nur über die Größe und nicht über die Exzentrizität. Es findet sich auf dem Bild wieder ein Artefakt als Produkt des Hintergrundrauschens. Bei genauerer Betrachtung des Artefaktes, kann man jedoch erkennen, dass eine Anhebung der Größenschranke für Artefakte diesen Fehler beheben würde.

Die Situation bei den hexagonalen Säulen ist wieder besser, denn diese wurden alle eindeutig identifiziert. Aber auch hier entstand wiederum ein Artefakt, welches als unrunder Wassertropfen identifiziert wurde. Betrachtet man auch dieses Artefakt genauer, erkennt man, dass wie schon beim Bild zuvor eine Anhebung der Größenschranke für Artefakte diesen Fehler beheben würde.

Insgesamt erweist sich die Hintergrundbeleuchtung noch problematischer als die Frontalbeleuchtung. Die hexagonalen Plättchen sind ähnlich rund wie Wassertropfen – für die Wassertropfen hat man einen Durchschnitt der Exzentrizität von 0,035070 und für die hexagonalen Plättchen einen Durchschnitt von 0,038702. Hinzu kommt noch, dass sich auf den Bildern Artefakte finden, die durch die Kombination Hintergrundbeleuchtung und Rauschen erzeugt werden.

Somit erweist sich die Hintergrundbeleuchtung als ungeeignete Variante, die zu einer fehlerhaften Klassifikation in Wasser und Eis führt. Aus diesem Grund wurden auch keine weiteren Testbilder mit Hintergrundbeleuchtung mehr gerendert.

4.3 Beleuchtung senkrecht zur Kameraachse

Testbilder mit Beleuchtung senkrecht zur Kameraachse wurden gleich bei der Programmentwicklung verwendet. Deshalb wurden für diese Beleuchtungsvariante auch mehr Testbilder erstellt.

Die Wassertropfen auf den Testbilder wurden eindeutig identifiziert. Von 58 betrachteten Wassertropfen wurden die meisten anhand ihrer Exzentrizität identifiziert. Nur 3 der betrachteten wurden als unrunde Wassertropfen identifiziert. Dies kommt daher, dass bei seitlicher Beleuchtung, die dem Licht abgewandte Seite wesentlich dunkler sein kann und darum nur ein Teil des Objektes sichtbar ist. Folglich wird nicht der volle runde Wassertropfen betrachtet und man erhält eine größere Exzentrizität.

Allerdings zeigte sich auch hier ein Problem des Verfahrens, nämlich dass ein Paar zusammenklebender Wassertropfen als hexagonale Säule identifiziert wurde. Betrachtet man den entsprechenden Ausschnitt des Bildes (Abbildung 25), kann man sehen, dass die beiden Tropfen so zusammenkleben, dass sich eine Kontur ergibt, die der einer hexagonalen Säule gleicht und eine hohe Exzentrizität besitzt.



Abbildung 25 Zusammenklebende
Wassertropfen

Folglich entspricht auch der Wert der Exzentrizität von 0,505207, dem Wert einer hexagonalen Säule und wird somit als solche klassifiziert. Das Problem ist, dass zusammenklebende oder sich überschneidende Objekte nicht richtig erkannt werden. Dies ist unabhängig von der Form und Beleuchtung der einzelnen Objekte.

Als nicht ganz einfach erweist sich wiederum die Klassifikation der hexagonalen Plättchen. Von diesen wurden 17 vollkommen eindeutig über die Exzentrizität, weitere 40 der betrachteten 60 werden als runde hexagonale Plättchen und zwei Objekte werden als

hexagonale Säulen identifiziert. Das kommt daher, dass die Kamera auf die Längsseite der hexagonalen Plättchen schaut und diese so gesehen als hexagonale Säulen erscheinen (Abbildung 26).

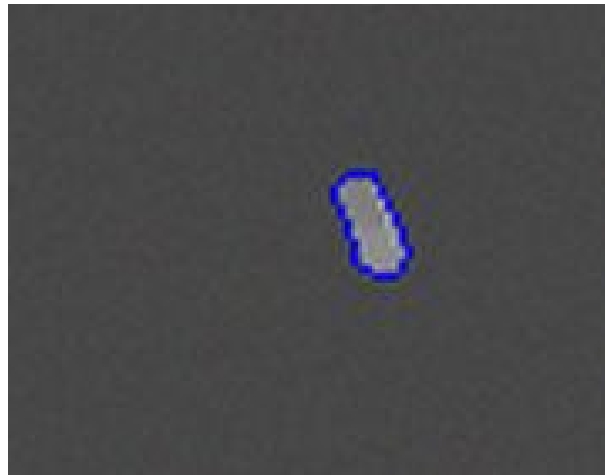


Abbildung 26 als hexagonale Säule identifiziertes hexagonales Plättchen

Eine einziges hexagonales Plättchen wurde auch als unrunder Wassertropfen identifiziert, das passierte weil es zu weit weg von der Kamera entfernt war und somit viel zu klein erschien.

Bei den hexagonale Säulen verhielt sich das Ganze relativ unproblematisch – sie werden eindeutig identifiziert. Von 60 betrachteten hexagonalen Säulen wurden 59 eindeutig als solche identifiziert. Nur eine einzige wurde als hexagonales Plättchen identifiziert. Dies liegt daran, dass die Kamera vor allem auf die Grundplatte der hexagonalen Säule schaute und diese so einem hexagonalen Plättchen sehr ähnlich war. (Abbildung 27).



Abbildung 27 also hexagonales Plättchen identifizierte Hexagonale Säule

Aus diesem Blickwinkel sind hexagonale Säulen und Plättchen von der Kontur her fast nicht zu unterscheiden, allerdings wirkt sich dieser Fehler nicht auf die Klassifikation in Eis und Wasser aus, da eine eindeutige Klassifikation als Eis erfolgt.

Alle Testbilder besitzen schon eine leichte Unschärfe. Bei der seitlichen Beleuchtung habe ich jedoch Testbilder mit unterschiedlicher Stärke der Unschärfe erstellt. Dabei stellte sich heraus, dass sich eine leichte Unschärfe unproblematisch verhält, selbst auf einem komplett scharfen Bild waren die Ergebnisse nicht besser. Verstärkt man die Unschärfe der Bilder treten zwei Effekte auf: Objekte, die zu weit aus dem Schärfebereich entfernt sind, werden nicht mehr erfasst – da sie zu unscharf sind und mit dem Hintergrund verschmelzen. Dies betrifft vor allem die kleinen Wassertropfen. Der andere Effekt ist, dass die Grenze zwischen Tropfen und hexagonalen Plättchen weiter verwischt wird, da die durch die Unschärfe verwischten Tröpfchen größer erscheinen.

4.4 Variation der Größe

Um den Einfluss der Größe zu untersuchen, wurden weitere Bilder mit den Eiskristallen und Wassertröpfchen in verschiedenen Größen gerendert. Dazu wurde auch die Berücksichtigung des Flächen-Moments nullter Ordnung im Programm annäherungsweise korrigiert. Zunächst wurden auf diesen Testbildern die Objekte auf 125%, 150% und 200% vergrößert und diese dann mit dem Programm ausgewertet, um zu sehen, ob man durch die Vergrößerung eine Verbesserung der Ergebnisse erhält.

Eine Vergrößerung der Objekte hat vor allem zur Folge, dass die Konturen durch mehr Punkte genauer beschrieben werden. Hierdurch ergibt sich in den meisten Fällen eine genauere Berechnung der Momente und es lassen sich bei größeren Objekten auch neue präzisere Grenzwerte für die Exzentrizität wählen. Allerdings hat sich die bisherige Größe – im Durchschnitt für hexagonale Plättchen 159 Pixel, durchschnittliche Größe hexagonale Säulen 258 Pixel und durchschnittliche Größe Wassertropfen 39 Pixel – als ausreichend für die Klassifikation der Eiskristalle und Wassertropfen erwiesen. Auf den Testbildern entsprachen dies Größen bei den Wassertropfen von 30 - 50 μm , bei den hexagonalen Plättchen von 60 – 100 μm und bei den hexagonalen Säulen von 180 – 300 μm .

Danach wurde untersucht, wie sich eine Verkleinerung der einzelnen Objekte auf die Klassifikation auswirkt. Das größte Interesse lag darin, wie klein die einzelnen Objekte minimal sein können, so dass noch eine einwandfreie Klassifikation möglich ist. Dazu wurden auch Bilder in einer geringeren Auflösung von 1024 x 768 Pixel gerendert.

Die Erkennung der Wassertropfen verhält sich relativ robust, selbst bei kleinen Größen erhält man noch zuverlässige Ergebnisse. Das Problem hierbei ist eher, dass eine Kontur sehr leicht rund oder zu exzentrisch sein kann, weil sie einfach durch viel zu wenige Punkte unpräzise beschrieben wird.

Bei großen Wassertropfen wird die Kontur wesentlich präziser beschrieben und somit auch die Exzentrizität präziser berechnet. Folglich erhält man für diese einen deutlich niedrigeren Wert für die Exzentrizität – bis hin zu dem perfekten Wert von 0.

Als sehr problematisch erweisen sich wiederum die hexagonalen Plättchen. Sie wurden bei fallender Größe oft sehr rund beschrieben, da einfach zu wenige Punkte für eine präzise Beschreibung, vor allem der Ecken, vorhanden waren. Dies führte zu schlechten Ergebnissen bei geringer Auflösung der Kamera.

Die Exzentrizität erweist sich auch als abhängig von der Größe – sowohl bei sehr großen Hexagonalen Plättchen als auch bei sehr großen Wassertropfen wird eine wesentlich geringere Exzentrizität berechnet, womit dann geringere Grenzwerte notwendig sind.

Betrachtet man zum Beispiel in Realität [Heymsfield02] Eiskristalle in Größe einige 100µm, kann man sehen, dass ihre Form oft unregelmäßiger als auf den künstlich generierten Testbildern ist. Dadurch ergibt sich eine höhere Exzentrizität und sie sollten in dieser Form leichter zu identifizieren sein.

Hexagonale Säulen lassen sich unabhängig von der Größe wieder gut identifizieren. Sowohl bei großen als auch bei kleineren Größen lassen sie sich noch gut erkennen. Dies funktioniert zuverlässig bis zu einer durchschnittlichen Größe von 40 - 50 Pixel, wobei die Exzentrizität bei diesen kleinen Größen wieder abnimmt. Erst unter dieser Größe wird die Identifikation problematisch. Problematisch ist es hier nur wenn der Blick wieder vor allem auf die Grundplatte der hexagonalen Säule fällt - ist diese zu klein ist, verhält sie sich ähnlich wie ein hexagonales Plättchen.

4.5 Folgerung

Eine Unterscheidung zwischen Wasser und Eis ist gut möglich, allerdings sollte die Größenverteilung zwischen Eiskristallen und Wassertropfen bekannt sein. Am problematischsten für die Klassifikation sind die hexagonalen Plättchen, ihre Werte für die Exzentrizität überschneiden sich stark mit denen der Wassertropfen. Dies ist auch von der Beleuchtung abhängig - eine ungünstige Beleuchtung verschlimmert diesen Effekt noch zusätzlich. Zudem kann eine ungünstige Beleuchtung zur Bildung von Artefakten auf den Bildern führen. Dieses Problem wird in der Realität geringer sein, da die Eiskristalle wahrscheinlich von der Form her nicht ganz so regelmäßig sind (siehe URL 38, [Hobbs90] oder [Kumai62]), wodurch sich das Problem abschwächen sollte

Mit dem bisher verwendeten Verfahren lässt sich kein Unterschied zwischen hohlen hexagonalen Säulen und normalen hexagonalen Säulen feststellen.

Weitere Formen, die sich äußerlich unterscheiden, sollten sich jedoch anhand der Momente erkennen lassen, wobei auch die Verwendung weiterer Momente notwendig sein kann.

5 Kameratechnik

Anhand der bisher erarbeiteten Ergebnisse werden hier Verfahren der Kameratechnik vorgestellt, die für die Verarbeitung geeignete Bilder liefern können. Denn zur Klassifizierung der Objekte in Eis und Wasser ist eine ausreichend gute Bildqualität notwendig. Wie im vorherigen Kapitel zu sehen ist, müssen die einzelnen Objekte deutlich auf den Bildern erkennbar sein. Dies ist abhängig von der verwendeten Kamera – sowohl von deren Optik als auch der verarbeitenden Elektronik.

5.1 Bedingungen für ein mögliches Aufnahmeverfahren

Eine wichtige Bedingung ist, dass das Aufnahmeverfahren ein In-Situ-Verfahren sein muss, da sich die Objekte vollkommen berührungsfrei am Aufnahmegerät vorbei bewegen. Somit würde zum Beispiel CPI nicht funktionieren. Die Objekte werden sich relativ schnell – man rechnet mit Geschwindigkeiten von bis zu 10ms^{-1} – an der Kamera vorbei bewegen. Folglich ist eine kurze Belichtungszeit notwendig, damit die Objekte scharf genug abgebildet werden.

Weiterhin müssen die Objekte ausreichend groß abgebildet werden, das heißt abhängig von der Größe der betrachteten Objekte müssen sie entsprechend vergrößert werden. Hierzu ist es ebenfalls gut wenn nur wenig Rauschen vorhanden ist damit kleine Objekte besser berücksichtigt werden können.

Immerhin reicht beim verwendeten Verfahren die Beschränkung auf eine Kamera aus. Das Verfahren sollte auch nicht zu aufwendig und damit teuer sein, da die CFDC eventuell mehrere Stufen hat, die gleichzeitig beobachtet werden müssen.

5.2 vorhergehendes CFDC Verfahren

Hier folgt nochmal eine kurze Beschreibung des Aufnahmeverfahrens der CFDC aus [Otto02]. Die Wassertropfen wurden von einer Xenon-Lampe oder Laser-Diode beleuchtet. Dann wurden die Objekte in einem Volumen von $12 \times 15 \times 2\text{mm}$ mit einer CCD-Kamera mit der Auflösung 320×256 Pixel als 8Bit Grauwertbild aufgenommen. Das Problem war die zu geringe Größe der Objekte in Kombination mit der geringen Auflösung der Kamera – die beobachteten Objekte waren nur wenige Pixel groß. Dies war ausreichend um die Objekte zählen zu können, aber Objektdetails sind auf den Bildern nicht mehr zu erkennen. Eine Klassifikation der Objekte ist so unmöglich. Auch ist das Rauschen auf den Bildern sehr ausgeprägt.

5.3 PATRICIA

Ein Experiment an dem man sich bezüglich der Kameratechnik orientieren kann ist PATRICIA – hiermit wurden schon sehr kleine Eiskristalle beobachtet (Abbildung 1). PATRICIA wurde entwickelt zur gleichzeitigen Analyse von Partikeln und ihrer Bewegung [URL 29], [[Poppe02]]. Es ist für die Untersuchung von Partikeln bis zu einer Größe von wenigen Mikrometern gedacht. Diese Partikel werden berührungsfrei gemessen. Dabei werden gleichzeitig zwei Messungen vorgenommen, nämlich ein Abbild des Partikels und die Partikelbahn.

Die Abbildung der Partikel wird mit Hellfeldmikroskopie mit einer CCD-Kamera (Sony DX9100P) der Auflösung 732 x 573 Pixel vorgenommen. Hierzu wird ein Mikroskop (Firma Thalheim-Spezialoptik) mit langen Fokalabstand verwendet – zur Zeit sind dies 80mm. Hierfür gibt es je nach Größe der zu betrachteten Objekte drei verschiedenen Optiken mit unterschiedlicher Vergrößerung. Damit ergibt sich auch ein unterschiedlich großes Sichtfeld. Die lange Seite hat je nach Vergrößerung entweder eine Länge 3,89mm oder 2,28mm und bei der stärksten Vergrößerung hat die lange Seite eine Länge von 0,78mm. Dadurch ergibt sich eine Auflösung von 5,3 μ m, 3,1 μ m oder 1,1 μ m pro Pixel. Entsprechend verhält sich die Tiefenschärfe – sie beträgt 0,8mm für die geringste Vergrößerung, 0,5mm für die mittlere und 0,2mm für die stärkste Vergrößerung.

Die Beleuchtung erfolgt durch eine Blitzlampe (Hamamatsu L4634) mit einer kurzen Beleuchtungsdauer von weniger als 0,5 μ s. Sie ist als Hintergrundbeleuchtung angebracht. Abhängig von ihr und der gewählten Vergrößerung ergibt sich dann auch die maximale Geschwindigkeit der beobachteten Objekte. Bei der für die CFDC gedachten maximalen Partikelgeschwindigkeit von 5ms⁻¹ bewegt sich ein Objekt innerhalb der Blitzdauer von 0,5 μ s um 2,5 μ m. Dies entspricht bei der geringsten Vergrößerung 0,5 Pixeln und bei der stärksten Vergrößerung sogar 2,5 Pixeln, was ein unscharfes Bild ergeben würde.

Die Verfolgung der Partikelbahnen ist kein primäres Ziel der CFDC, aber integraler Bestandteil von PATRICA, deswegen möchte ich der Vollständigkeit halber kurz auf das Verfahren zur Untersuchung der Partikelbahn eingehen - sie wird stroboskopisch aufgenommen. Die Beleuchtung erfolgt durch eine Laserdiode mit einer Frequenz bis zu 20MHz. Das am Objekt gestreute Licht wird mit einem langreichweitigen Mikroskop mit einer CCD-Kamera aufgenommen. Dabei wird das Eindringen von direktem Laserlicht durch einen blinden Fleck vor der Mikroskopoptik verhindert. Man erhält somit Bilder der Trajektorien der Partikel und dies für Partikelgeschwindigkeiten bis zu 110ms⁻¹ und

Partikelbahnen von Partikeln der Größe bis einige 100nm. Das besondere an PATRICA ist, dass Partikel und Partikelbahn gleichzeitig auf zwei verschiedenen Farbkanälen aufgenommen werden – nämlich auf Grün wird das Partikel abgebildet und auf Rot dessen Bewegung.

Aber das interessante und wichtige an PATRICIA ist, dass hiermit ein Verfahren vorliegt, dass Bildern von solch kleinen Objekten liefern kann, wie sie für die Auswertung benötigt werden.

5.4 PIV / PTV

Ein anderes Verfahren, das als Orientierung diene, war das von Bröder und Sommerfeld in „Visual Study of bubble-bubble interactions with a high speed camera“ [URL 8] vorgeschlagene System zur „Particle Image Velocimetry“ - kurz PIV, und „Particle Tracking Velocimetry“ kurz PTV. Hier handelt es sich um einen ähnlichen Versuch aber mit größeren Objekten und mit quasi umgekehrten Vorzeichen – hier ging es um Luftblasen in einer Wassersäule, statt um Eiskristall und Wassertröpfchen in Luft. Eine einzige CCD-Kamera (PCO SensiCam) mit Makrooptik und einer Auflösung 1280 x 1024 Pixel beobachtet Luftblasen mit einer Größe von 2 - 4mm und Tracer-Partikel mit einem Durchmesser von 40 - 60µm in einer Wassersäule. Der Tiefenschärfebereich ist kleiner als 4mm. Die Beleuchtung erfolgt durch ein gegenüber der Kamera angebrachtes LED-Array mit einer Pulsdauer von 200µs. Diese längere Zeit ist gerechtfertigt aufgrund der langsameren Geschwindigkeiten der in Wasser aufsteigenden Luftblasen von bis zu etwas 0,36ms⁻¹.

Die Abbilder der kleinen Tracer-Partikel werden jedoch nur in Bezug auf die Geschwindigkeit mittels PIV ausgewertet – für die weitere Verarbeitung werden sie ebenso wie das Rauschen mit einem Median-Filter entfernt. Es werden allerdings nur die größeren scharf abgebildeten Blasen von der Form her ausgewertet – es ging hier um ihre Koaleszenzraten.

Er hat aber auch als Anregung gedient selbst einen Test mit Makrooptik durchzuführen. Zunächst bin ich davon ausgegangen, das Verfahren der PIV und PTV auch bei der Partikelverfolgung hilfreich sein können, dies hat sich aber in diesem Anwendungsfall als nicht möglich erwiesen.

5.5 Kamerateest mit Digitalkamera

Um zu sehen, welche Bildqualität eine konventionelle CCD-Kamera mit Makrooptik liefert, habe ich einige Testbilder gemacht, um zu sehen, welche Größe die kleinsten Strukturen, die noch aufgelöst werden, haben. Dazu habe ich die Kamera Ricoh Caplio RR1 verwendet. Sie hat eine Auflösung von 2272 x 1704 Pixel und eine Zoomoptik mit einer Brennweite von 7,3 - 21,9mm, die auch Makroaufnahmen ermöglicht.

Dann habe ich aus einer Distanz von etwa 15mm Bilder von einem Testbild für die Druckvorstufe aus [FOGRA93] gemacht. Die Brennweite des Objektivs war 10,7mm. Dieses Testbild enthält Mikrolinien und Spalten mit einer Breite von 4µm bis 40µm.

Die kleinsten Linien, welche auf dem Testbild noch einwandfrei erkennbar waren, hatten eine Größe von 15µm. Linien mit einer Breite von 40µm wurden auf dem Bild noch mit einer Breite von 5 Pixel dargestellt. Dies entspricht noch den kleinsten noch erkennbaren hexagonalen Säulen mit 40 - 50 Pixeln. Das Rauschen war auf den Bildern nur schwach ausgeprägt. Die Tiefenschärfe auf den Bildern war leider gering, aber ich konnte immerhin erkennen, dass man mit einer Makrooptik Objekte dieser Größe noch abbilden kann.

5.6 Empfehlungen für Kameratechnik

Aus diesem Versuchen/Gegebenheiten leiten sich für mich folgende Schlüsse ab.

Die Partikel müssen aufgrund ihrer Geschwindigkeit mit einer kurzen Belichtungszeit aufgenommen werden, um noch ein scharfes Bild erhalten zu können. Diese ist abhängig von der Geschwindigkeit der Partikel – für 5ms^{-1} sind schon Zeiten von $0,5\mu\text{s}$ empfehlenswert. Deswegen erscheint mir die Beleuchtung mittels einer entsprechenden Blitzlampe oder eines Lasers als unabdingbar.

Die Verwendung einer hochauflösende Kamera zum Beispiel von PCO [URL 30] ist absolut notwendig. Eine höher auflösende Kamera ermöglicht auch ein größeres Sichtfeld. Bei relativ großen Partikeln mit Größen über $100\mu\text{m}$ ist die Verwendung von Makrooptik mit einer hochauflösende Kamera möglich. Man kann sie hierbei an den Gegebenheiten bei der Erstellung der Testbilder orientieren, das heißt die Verwendung einer Kamera mit einer ähnlichen Auflösung wie die virtuelle Kamera der Testbilder.

Für kleinere Partikel ist eine Mikroskopoptik vergleichbar PATRICIA eher angemessen, um eine entsprechende große und detaillierte Abbildung der Partikel zu erreichen, wenn deren Größe nur wenige $10\mu\text{m}$ beträgt.

Betrachtet man die Bilder aus PATRICIA – der 35μ Eiskristall wurde auf einem Blickfeld von $0,75 \times 0,59\text{mm}$ mit einer Auflösung von 732×575 Pixel aufgenommen – dies

entspricht $1\mu\text{m}$ pro Pixel. Strebt man bei Beibehaltung dieses Wertes ein größeres Blickfeld wie die für die neue CFDC angedachten $4 \times 4\text{mm}$ an, ist die Verwendung einer entsprechend höher auflösenden Kamera notwendig. Hierfür bietet sich zum Beispiel eine Kamera wie die pco.4000 von pco.imaging [URL 30] mit einer Auflösung 4032×2688 Pixel an. Allerdings ist hier auch ein Blickfeld mit Seitenverhältnis 4 zu 3 gemäß der Kamera statt einem quadratischen $4 \times 4\text{mm}$ sinnvoll.

Es gab die Idee auch in Realität analog wie im Raytracing ebenfalls eine Lochkamera statt einer normalen Kameraoptik zu verwenden. Lochkamera haben den Vorteil einer unendlichen Tiefenschärfe, allerdings ist ihre Auflösung durch die Blendenöffnung [URL 49] begrenzt. Leider habe ich auch nach intensiver Suche keine Quellen gefunden, wo man in Versuchen ähnlich kleine Objekte mit einer Lochkamera aufgenommen hat.

6 Zusammenfassung und Ausblick

Eine Klassifikation der Partikel anhand eines einzelnen Bildes mit einem momentbasierten Verfahren ist gut möglich, allerdings ist die Verwendung einer hochauflösenden Kamera zwingend. Am besten funktionierte die Klassifikation bei Tropfen und bei hexagonalen Säulen bis zu einer minimalen Abbildungsgröße der Objekte von 40 Pixel. Als äußerst problematisch haben sich dagegen die hexagonalen Plättchen erwiesen, denn sie lassen sich leicht mit den Wassertropfen verwechseln. Die Größe hat sich hier als ein wichtiges und oft notwendiges Entscheidungskriterium erwiesen. Allerdings wird mit steigender Größe die Exzentrizität der Wassertropfen besser berechnet, diese nähert sich dann dem idealen Wert von 0 an.

Deswegen hat sich als Herausforderung nicht die große Formenvielfalt der Eiskristalle erwiesen, sondern das Hauptproblem lag vielmehr darin, Wassertropfen von ziemlich runden Eiskristallen zu unterscheiden.

Die verwendete Bibliothek OpenCV bietet die für die Verarbeitung des Bildes notwendigen Funktionen – sollten später noch weitere Formen erkannt werden, ist dies damit möglich. Zudem bietet OpenCV auch noch die Möglichkeit der direkten Kamerasteuerung.

Die Geschwindigkeit der Partikel ist vor allem ein Problem der Kamera- und Beleuchtungstechnik. Nach den Erkenntnissen, vor allem aus PATRICIA, kann man 10ms^{-1} als obere Grenze für die Partikelgeschwindigkeit ansetzen – schneller sollten sie auf keinen Fall sein, da man sonst keine scharfen Bilder mehr erhält.

Die Überlegungen einen kleinen Beobachtungsbereich der Größe $4 \times 4\text{mm}$ zu verwenden, sind gut, sicher besser als $12 \times 16\text{mm}$. Es wäre auch gut, wenn die Partikel sich in einem kleinen Tiefenbereich befinden – die Verwendung aerodynamischer Linsen, um dies zu unterstützen, wäre sinnvoll.

Die Beleuchtung von der Seite, die schon bei der vorhergehenden CFDC eingesetzt wurde, hat sich auch hier als geeignetes Verfahren erwiesen.

Zur Auswertung der Bilder wurde wiederum ein Rechner mit 1.9GHz PIV mit 768MB RAM und Windows XP Professional verwendet. Das angewendete Verfahren arbeitet sehr schnell auf diesem Rechner – für die Verarbeitung eines Bildes mit 1024×768 Pixel wurde 1 Sekunde benötigt und für die Bilder mit 2272×1706 Pixel 5 Sekunden. Durch den Verzicht auf die Ausgabe auf dem Monitor und als Datei sollte sich noch zusätzlich Zeit sparen lassen.

7 Anhang

7.1 Literaturverzeichnis

- [Ebert00] M. Ebert, S. Weinbruch, P. Hoffmann and H. M. Ortner (2000) Chemical Characterization of North Sea Aerosol Particles, Elsevier Science BV, J. Aerosol Sci. Vol.31, No. 5, 613-632
- [FOGRA93] FOGRA Forschungsgemeinschaft Druck e. V. (1993) Kontrollmittel, FOGRA Forschungsgemeinschaft Druck e. V.
- [Funkhouser03] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, David Jacobs (2003) A Search Engine for 3D Models, ACM, Transactions on Graphics, Vol. 22, No. 1, 83-105
- [Heymsfield02] Andrew J. Heymsfield et al. (2002) Observations and Parameterizations of Particle Size Distributions in Deep Tropical Cirrus and Stratiform Precipitating Clouds: Results from In Situ Observations in TRMM Field Campaigns, American Meteorological Society, Journal of the Atmospheric Sciences Vol. 59, 3457-3491
- [Hobbs90] Peter V. Hobbs, Arthur L. Rangno (1990) Rapid Development of High Ice Particle Concentrations in Small Polar Cumuliform Clouds, Journal of the Atmospheric Sciences Vol. 47, 2710-2722
- [Jähne02] Bernd Jähne (2002) Digitale Bildverarbeitung, 5. überarbeitete und erweiterte Auflage, Springer Verlag,
- [Kumai62] Motoi Kumai, Karl E. Francis (1962) Nuclei in Snow and Ice Crystals on the Greenland Ice Cap under Natural and Artificially Stimulated Conditions, Journal of the Atmospheric Sciences Vol. 19, 474-481
- [Mason62] B. J. Mason (1962) Clouds, Rain & Rainmaking, Cambridge University Press
- [Mossop85] S. C. Mossop (1985) The Origin and Concentration of Ice Crystals in Clouds, American Meteorological Society, Bulletin American Meteorological Society Vol. 66, No. 3, 264-273
- [Otto02] Peter Otto, Hans-Walter Geoghegan, Heinz Bingemer (2002) A new three stage continuous flow CCN-Counter, Elsevier Science BV, Atmospheric Research 61, 299-310
- [Poppe02] Torsten Poppe, Gerhard Wurm, Rene Krieg (2002) Optical particle and particle motion analysis with PATRICIA, Institute of Physics Publishing, Measurement Science and Technology, 796-802
- [Pruppacher97] Hans R. Pruppacher, James D. Klett (1997) Microphysics of Clouds and Precipitation -2nd revised and enlarged edition, Kluwer Academic Publishers, Atmospheric and Oceanographic Sciences Library Volume18,

[Rogers98] David C. Rogers (1998) Development of a Continuous Flow Thermal Gradient Diffusion Chamber for Ice Nucleation Studies, Elsevier Science BV, Atmospheric Research 22, 149-181

[Seinfeld98] John. H. Seinfeld, Spyros N. Pandis (1998) Atmospheric Chemistry and Physics: From air pollution to climate change, A Wiley-Interscience publication

[Vössing98] Hermann-Josef Vössing, Stefan Borrmann, Ruprecht Jaenicke (1998) In-line holography of cloud volumes applied to the measurement of raindrops and snowflakes, Elsevier Science BV, Atmospheric Research 49, 199-212

[Wang02] Pao K. Wang (2002) Shape and Microdynamics of Ice Particles and Their Effects in Cirrus Clouds, Academic Press, Advances in Geophysics Volume 45, 1-43

7.2 Internetquellen

URL 1 Christine Antoine, Mike D. Lloyd, Jacques Antoine A Robust Thresholding Algorithm For Halftone Dots <<http://www.supelec.fr/ecole/radio/JPPS01.pdf>> (03.07.2003)

URL 2 James Matthews An Introduction to Edge Detection: The Sobel Edge Detector <<http://www.generation5.org/content/2002/im01.asp>> (05.08.2003)

URL 3 Beleg Neuronale Netze <<http://www.informatik.htw-dresden.de/~iwe/Belege/Hahn/documentation/beleg.htm>> (01.06.2003)

URL 4 Bildverarbeitung, Objekterkennung und Neuronale Netze. <<http://inf33-www.informatik.unibw-muenchen.de/research/vision.html>> (26.05.2003)

URL 5 Computer Vision Software <<http://www-2.cs.cmu.edu/afs/cs/project/cil/ftp/html/v-source.html>> (20.08.2003)

URL 6 CRYSTAL-FACE Instruments <<http://cloud1.arc.nasa.gov/crystalface/instruments.html>> (10.03.2003)

URL 7 Electron Microscopy Unit Snow Page <www.lpsi.barc.usda.gov/emusnow/default.htm> (10.03.2003)

URL 8 D. Bröder, M. Sommerfeld Experimental studies of bubble interaction and coalescence in a turbulent flow by an imaging PIV/PTV system <http://www-mvt.iw.uni-halle.de/articles/lisabon_2002_v0.pdf> (26.05.2003)

URL 9 Feature Detectors – Canny Edge Detector <<http://www.dai.ed.ac.uk/HIPR2/canny.htm>> (02.08.2003)

URL 10 Feed-Forward-Netze <<http://kik.informatik.fh-dortmund.de/SR/Pro+Dipl/neuronalenetze/kurs2.html>> (01.06.2003)

URL 11 FIRE III ACE Archive
<http://eosweb.larc.nasa.gov/project/fire/table_fire3_ace.html> (02.03.2003)

URL 12 Freeman chain code
<<http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u31.html#OUT31.3.2>>
(04.08.2003)

URL 13 Gandalf Home Page <<http://gandalf-library.sourceforge.net/>> (27.05.2003)

URL 14 Philip F. McLauchlan Gandalf: The Fast Computer Vision and Numerical Library
<<http://gandalf-library.sourceforge.net/tutorial/report.pdf>> (27.05.2003)

URL 15 [http--sourceforge.net-projects-opencvlibrary-](http://sourceforge.net/projects/opencvlibrary/)
<<http://sourceforge.net/projects/opencvlibrary/>> (08.07.2003)

URL 16 [http--www.imm.dtu.dk/~aam-introduction.html](http://www.imm.dtu.dk/~aam-introduction.html)
<<http://www.imm.dtu.dk/~aam/introduction.html>> (30.05.2003)

URL 17 [http--www.intel.com-research-mrl-research-opencv-](http://www.intel.com-research-mrl-research-opencv-)
<<http://www.intel.com/research/mrl/research/opencv/>> (08.07.2003)

URL 18 [http--www.shit-chaos.net-am-user-Photon_mapping-dts-1.0.pdf](http://www.shit-chaos.net-am-user-Photon_mapping-dts-1.0.pdf) <http://www.shit-chaos.net/am-user/Photon_mapping-dts-1.0.pdf> (09.08.2003)

URL 19 [http--www.yahoogroups.com-messages-OpenCV](http://www.yahoogroups.com-messages-OpenCV)
<<http://www.yahoogroups.com/messages/OpenCV>> (08.07.2003)

URL 20 Gabor Vali Ice nucleation - theory. A tutorial Presentation at the 1999
NCAR/ASP Summer Colloquium. <http://www-das.uwyo.edu/~vali/nucl_th.pdf>
(20.08.2003)

URL 21 Ice Nuclei Research - Continuous Flow Diffusion Chamber
<<http://lamar.colostate.edu/%7Epdemott/cfdc/cfd.html>> (10.03.2003)

URL 22 Image Analyzer <<http://meesoft.logicnet.dk/Analyzer/>> (13.07.2003)

URL 23 Michael A. Wirth Image Segmentation
<<http://www.uoguelph.ca/~mwirth/CIS6320/lecture4.pdf>> (02.07.2003)

URL 24 Thomas Vetter Maschinelle Bildanalyse und Bildsynthese von Gesichtern
<<http://www.billingpreis.mpg.de/hbp97/vetter.pdf>> (30.05.2003)

URL 25 Frank-Michael Schleich Momentbasierte Methoden der Schriftzeichenerkennung
<<http://dol.uni-leipzig.de/pub/2002-33>> (17.06.2003)

URL 26 Noise sources in a CCD
<http://www.mssl.ucl.ac.uk/www_detector/ccdgroup/opttheory/darkcurrent.html>
10.09.2003

URL 27 Adam Smyczek Objekterkennung mit Active Appearance Modellen
<<http://www.rad.rwth-aachen.de/~kohnen/teaching/smy99.pdf>> (18.05.2003)

URL 28 Intel Corporation Open Source Computer Vision Library Reference Manual
<<http://prdownloads.sourceforge.net/opencvlibrary/OpenCVReferenceManual.pdf?download>> (08.07.2003)

URL 29 Partikelanalysator PATRICIA <http://www.astro.uni-jena.de/Users/poppe/patricia_neu/index.html> (26.04.2003)

URL 30 pco.imaging - specialized cameras <http://www.pco.de/php/products/index_1-en__01030401.html> (30.08.2003)

URL 31 POV-Ray - The Persistence of Vision Raytracer <<http://www.povray.org/>> (27.03.2003)

URL 32 POV-Ray questions and tips - Language
<<http://www.students.tut.fi/~warp/povQandT/languageQandT.html>> (13.04.2003)

URL 33 POV-Ray Tutorial - Einführung, deutsche Beschreibungen, Anleitungen und Beispiele zum Raytracer POVRAY zu 3D-Grafik mittels Ray <http://www.f-lohmueller.de/pov_tut/pov__ger.htm> (25.03.2003)

URL 34 POV-Ray: Documentation <<http://www.povray.org/documentation/>> (27.03.2003)

URL 35 POV-Ray: Resources: Links: <www.povray.org/resources/links/> (23.05.2003)

URL 36 Project: Open Computer Vision Library: File List
<http://sourceforge.net/project/showfiles.php?group_id=22870> (08.07.2003)

URL 37 psksvp's xperiment home page <<http://www.ccs.neu.edu/home/psksvp/lg.htm>> (01.06.2003)

URL 38 Replicator Data - Index <<http://box.mmm.ucar.edu/science/cirrus/>> (25.05.2003)

URL 39 Johannes Kilian Simple Image Analysis By Moments Version 0.2
<<http://groups.yahoo.com/group/OpenCV/files/Documentation/Simple%20Image%20Analysis%20by%20Moments.pdf>> (11.06.2003)

URL 40 Snow Crystal Photography <<http://www.microscopy-uk.org.uk/mag/artmay01/snowcrystal.html>> (10.03.2003)

URL 41 SNOW CRYSTALS <<http://radar.sci.hokudai.ac.jp/crystal/gallery.html>> (10.03.2003)

URL 42 Snowflakes and Snow Crystals
<<http://www.cco.caltech.edu/~atomic/snowcrystals/>> (28.01.2003)

URL 43 Spatial Filters – Gaussian Smoothing
<<http://www.dai.ed.ac.uk/HIPR2/gsmooth.htm>> (02.08.2003)

URL 44 Spatial Filters – Mean Filter <<http://www.dai.ed.ac.uk/HIPR2/mean.htm>>
(02.08.2003)

URL 45 Spatial Filters – Median Filter <<http://www.dai.ed.ac.uk/HIPR2/median.htm>>
(02.08.2003)

URL 46 SUSAN Low Level Image Processing
<<http://www.fmrib.ox.ac.uk/~steve/susan/susan/susan.html>> (27.05.2003)

URL 47 The Computer Vision Homepage <<http://www-2.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html>> (20.08.2003)

URL 48 The KUIM Image Processing System
<<http://www.ittc.ku.edu/~jgauch/research/kuim/html/index.html>> (27.05.2003)

URL 49 Matt Young The Pinhole Camera
<<http://www.pinhole.com/resources/articles/Young/>> (08.09.2003)

URL 50 USB 2.0-Karten für Highspeed-Verbindungen
<<http://www.monitor.co.at/index.cfm?storyid=4131>> (02.09.2003)

URL 51 The VXL Homepage <<http://vxl.sourceforge.net/>> (27.05.2003)

URL 52 George Bebis Thresholding
<<http://www.cs.unr.edu/~bebis/CS791E/Notes/Thresholding.pdf>> (03.07.2003)

URL 53 VXL <http://paine.wiau.man.ac.uk/pub/doc_vxl/> (27.05.2003)

URL 54 Yahoo! Groups: OpenCV Files <<http://groups.yahoo.com/group/OpenCV/files/>>
(08.07.2003)

7.3 Emails

Email 1 Heinz Bingemer <Bingemer@meteor.uni-frankfurt.de> 21.02.2003
„Tropfenbilder“ [Enthält Bilder aus der CFDC], persönliche Email, (erhalten 21.02.2003)

Email 2 Heinz Bingemer <Bingemer@meteor.uni-frankfurt.de> 25.03.2003 „Bilder“
[Enthält Bilder der Hologramme von Wassertropfen und Eiskristallen aus HODAR von H-
J. Vössing], persönliche Email, erhalten (25.03.2003)

Email 3 Marc Cymontkowski <cymontkowski@imageprocessing.de> 20.06.2003 „RE:
[OpenCV] Adding lines and columns“, Email an Mailingliste
OpenCV@yahoogroups.com, (erhalten 20.06.2003)

Email 4 RZ Liu <liurz@assuredigit.com> 02.07.2003 „Re: Re: Re: Re: [OpenCV] Finding
optimal Threshold value“ [Enthält Otsu Algorithmus], Email an Mailingliste
OpenCV@yahoogroups.com, erhalten 02.07.2003

7.4 Anlagen

Als Anlagen befinden sich auf der beiliegenden DVD

- Die in POV-Ray verwendeten Dateien
 - die Vorlagendatei crystals.inc
 - die Szenenbeschreibungen
 - die generierten Testbilder
- Programm
 - FindCircles
 - Sourcecode
 - ausführbare Datei
 - CrystalsandDrops
 - Sourcecode
 - ausführbare Datei
- Abbildungen
- Ergebnisse