Technische Universität
München
Fakultät für Informatik

Diplomarbeit

Modellbasierte, computergestützte Analyse von Daten aus dem DPIV-Verfahren

Mario Gleirscher

Technische Universität
München
Fakultät für Informatik

Diplomarbeit

Modellbasierte, computergestützte Analyse von Daten aus dem DPIV-Verfahren

Modell based, computer-aided Analysis of Data provided by the DPIV-Method

Mario Gleirscher

Aufgabensteller:
Prof. Dr. Rüdiger Westermann, Lehrstuhl für „Computer Graphics & Visualization“

Betreuer:
Prof. Dr. Rüdiger Westermann
Dipl. Inf. Thomas Schiwietz, Lehrstuhl für „Computer Graphics & Visualization“

Abgabedatum: 15. September 2004

Erklärung

Ich versichere, dass ich diese Ausarbeitung der Diplomarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 20. September 2004 Mario Gleirscher

„Man empfindet die Langeweile nicht,
wenn man niemals ordentlich arbeiten gelernt hat."
(Friedrich Nietzsche, 1883)

Zusammenfassung

Schon im Jahre 1904 hatte einer der damals bekanntesten Fluidmechaniker Ludwig PRANDTL die Idee eines Wasserzirkulationstunnels verwirklicht, in dessen strömendes Medium er winzige Partikel eingestreut und direkt mit freiem Auge beobachtet hat. Die Wissenschaft der Fluidmechanik in Kombination mit der modernen Technik hat diese Art der Strömungsvisualisierung bis heute hin in vielfältiger Art und Weise weiterentwickelt. Stand der Technik sind hochauflösende, digitale Bilderfassungssysteme, leistungsstarke Datenverarbeitungssysteme und statistische Methoden zur Rekonstruktion von Vektorfeldern aus hauptsächlich digitalen Bilddaten. Die bisher entwickelten Verfahren, darunter vor allem die „Particle Image Velocimetry" (PIV), sind mitterweile in vielen Anwendungsfällen sehr genau und effizient. Problematisch wird es bei mikroskopischen (μ-PIV) oder turbulenten, kompressiblen Strömungen mit stark vom Ideal abweichenden Aufnahmen, aber auch bei stark vom Standard differierenden Partikelbildcharakteristiken. Zur Erkennung von Strömungseigenschaften basierend auf finiten Differenzen, wie z. B. der Wirbelstärke, der Deformation und damit der Strömungstopologie, sind genaue Ergebnisse in der Rekonstruktion unerlässlich.

Hier wird nun versucht, Methoden und Ideen aus der digitalen Bildverarbeitung — **Kantendetektionsverfahren** im Speziellen — in geeigneter Weise anzuwenden, um die genannten Merkmale in den Aufnahmen zu erhalten, unerwünschte Bildinformationen jedoch zu eliminieren. Eine Betrachtung dieser Signale im Orts- und Frequenzbereich führte letztendlich zu weiteren Gedankengängen. Es wurde versucht, den negativen Einfluss von verrauschten Bildern auf diverse PIV-Methoden nachzuweisen. Tests haben gezeigt, dass es nötig ist, genauere frequenzanalytische Überlegungen anzustellen, um signifikant bessere Ergebnisse zu erzielen; z. B. die Anpassung von Wavelets auf die betrachtete Bild- und Rauschklasse abhängig von den für die PIV-Analyse wichtigen Merkmalen.

Schlussfolgernd ist ein Vorwissen von globalen Parametern wie z. B. der Geschwindigkeitsverteilung sehr wichtig für einen initialen, gut schätzenden Schritt einer iterativen PIV-Analyse. Vorgeschlagen wird ferner eine Methode zur Gewinnung dieses Vorwissens. Das Endergebnis dieser Arbeit zeigt sich einerseits durch eine überschauliche, bewertende und referenzierte Übersicht zahlreicher moderner Verfahren und Ideen dieser Disziplin, welche teils sehr unterschiedlich motiviert entstanden sind. Andererseits beinhaltet das Gesamtresultat auch eine ausführliche software- bzw. modellbasierte Validierung einiger Ansätze aus der digitalen Bildverarbeitung. Als Hilfsmittel diente dabei die dafür entworfene Testsoftware *PIVA*. Die Ergebnisse zeigen den Einfluss verschiedener Systemparameter auf die Analyse.

Stichwörter:

adaptiv, Ansätze, digitale Bildverarbeitung, DPIV, Implementierung, iterativ, Methoden, multiskalar, Überblick, Verfahren, Wavelets.

Abstract

Very early in 1904, one of the most popular fluid-mechanicians, Ludwig PRANDTL, had the idea of a tunnel containing circulating water, which he inspected by simply watching seeded particles. Today physical science and state-of-the-art techniques have lead this principle to a very sophisticated set of methods. With the help of high-resolution digital imaging and processing systems and via statistical methods, it became possible to gain vector field data out of exposures. In many use cases the current techniques make it feasible to get very accurate results at rather low cost. Increasingly higher usage of Particle Image Velocimetry (PIV) methods with microscopic, turbulent, compressible flows like in μ-PIV showed some problems when processing the low-quality images with non-standard particle image characteristics.

To achieve acceptable results for vorticity and deformation, furthermore, to analyse the topology of flows, various approaches, such as some methods for **edge detection**, out of the digital image processing discipline have been revisited. The features have to remain but the unwanted information have to be removed from the exposures. Observations of the signals in the spatial and frequency domains lead to further thoughts, which have to be investigated in detail during further derivative work. In addition, one of the tasks was to show the negative influence of noised images on various PIV methods. By now, the results of the test experiments pointed out, that it is important to optimize the analysis of the frequency information of the image by using appropriate wavelets for the respective image class. This must be done to obtain much better results, i. e. to reduce noise and to strengthen the power of the features wanted in the PIV analysis procedure.

One final conclusion was, that the previous knowledge of some global parameters such as the overall velocity distribution is fundamental to get initial estimators in order to start an iterative process leading into valid and highly accurate results. Methods will be discussed. The first part of the overall results of this work is an evaluative overview of variously motivated approaches out of the digital image processing discipline. As a second part a validation of the new approaches has been made via the implementation of a specific testing software. The results of the tests show the significance and the effect of the various system parameters on the analysis.

Keywords:

adaptive, approaches, digital image processing, DPIV, implementation, iterative, methods, multiscale, overall view, wavelets.

Vorwort

Mein Interesse an der Physik und im Speziellen an der Strömungsmechanik hat mich dazu bewogen, mich für das Thema dieser Diplomarbeit, „Modellbasierte, computergestützte Analyse von Daten aus dem DPIV-Verfahren“, zu entscheiden. Die Mitarbeit an einem interdisziplinären Projekt im Bereich der numerischen Strömungslöser am Lehrstuhl von Prof. Zenger im Institut für Informatik der TU München bot mir einige Grundlagen aus der Fluidmechanik. Für mich als Informatiker war es anfangs ungewohnt, da die Technik der PIV-Analyse aus einer ganz anderen Richtung an das Problem herangeht als die simulative Berechnung von Strömungsverhältnissen mittels NAVIER-STOKES-Gleichungen. Statistische Korrelationsmethoden und die digitale Bildverarbeitung mit der Signalverarbeitung als Grundlage führten zu einer ausgedehnten Einarbeitungsphase, in der ich sehr viel aus den verschiedenen Bereichen gelernt habe. Dazu zählen vor allem Recherche, wissenschaftliches Arbeiten und die Entwicklung, Implementierung und Dokumentation von datenverarbeitenden Systemen sowie die Erstellung einer umfassenden, strukturierten Ausarbeitung.

Ich hoffe, den Erwartungen der Leser dieser Arbeit zu einem genügenden Teil gerecht zu werden und sie zu animieren, sich in das Thema mit den hier angegebenen Hilfestellungen einzuarbeiten, meine Gedankengänge zu verfolgen und, wo gewünscht, auszubauen.

Danksagung. Hiermit möchte ich allen danken, die mich auf dem Weg durch das Informatikstudium und auf meinem Lebensweg begleitet und mir in schwereren Zeiten geholfen haben, vor allem meiner Familie. Meinen Freunden und Studienkollegen danke ich herzlich für die zahlreichen Gespräche, Fragestunden und Diskussionen, aus denen ich stets viel gelernt habe, allen voran Christian Müller, Franz Strasser und Sebastian Kienzl. Entsprechender Dank geht auch an meine Betreuer Prof. Dr. Rüdiger Westermann und Dipl. Inf. Thomas Schiwietz, welche mir bei der Erstellung dieser Diplomarbeit praktische Hilfestellung geboten haben.

Mario Gleirscher
<gleirsch@in.tum.de>

München, den 20. September 2004

Hilfsmittel. Diese Diplomarbeit wurde mit Leslie LAMPORT's LaTeX2ε [Kop02] basierend auf dem leistungsstarken Textsatzsystem TeX von Donald E. KNUTH [Knu86] gesetzt. Die Abbildungen u. a. im Abschnitt 8 wurden mit dem freien Daten- und Funktionsdarstellungswerkzeug *gnuplot* generiert [WK04]; basierend auf den Ergebnissen erzeugt durch *PIVA* (siehe Abschnitt 7).

Inhaltsübersicht

1 Einführung 5

Das Fachgebiet der Strömungsvisualisierung hat bereits eine lange Geschichte, ist breit gefächert und bietet mittlerweile eine Fülle an Lösungsansätzen

2 Erweiterte Ansätze zur Verbesserung der DPIV 17

Die digitale Analyse von PIV-Daten hat sich im letzten Jahrzehnt stark weiterentwickelt und hält für spezielle Anwendungen mit hohen Anforderungen optimierte Verfahrensvarianten parat

3 Erkennung von Merkmalen in Strömungen 35

Im Gegensatz zu den Merkmalen in Bildern betrachtet man als Strömungsmechaniker vor allem die aus den Vektordaten berechenbaren Strömungsmerkmale

4 Methodische Hilfsmittel aus der digitalen Bildverarbeitung 39

Die Detektion von brauchbaren Merkmalen in Bildern hängt sehr stark mit der Anwendung passender digitaler Filter zusammen

5 Hierarchische Ansätze in der DPIV 49

Eine hierarchische, adaptive Analyse von Daten eröffnet neue Möglichkeiten der Rekonstruktion und steigert die Effizienz der Verfahren

6 Betrachtung frequenzbasierter Ansätze zur Merkmalsextraktion 57

Bis dato werden in der PIV leistungsfähige, digitale Filtermethoden zur Behandlung von qualitativ schlechten, verrauschten Aufnahmen kaum verwendet

7 Beschreibung der Implementierung 65

Zur Evaluation der DPIV-Verfahren bedarf es eines flexiblen Softwaresystems zur parametrisierbaren Analyse von Bildern und Bildsequenzen

8 Testexperimente und Ergebnisse 85

Die Validierung des Einflusses verschiedener Verfahrensparameter auf die Ergebnisse liefert wichtige Aussagen über die Qualität der PIV-Analyse

9 Ausblick 109

Motto: „Niemals ist ein Thema als erschöpft oder eine Aufgabe als entgültig abgeschlossen anzusehen.“

Inhaltsverzeichnis

Vorwort i

Inhaltsübersicht iii

Ausführliches Inhaltsverzeichnis vii

Abbildungsverzeichnis ix

Tabellenverzeichnis x

Über den inhaltlichen Aufbau dieser Arbeit 1

I Theorie 3

1 Einführung 5
- 1.1 Geschichte der Strömungsvisualisierung . . . 5
- 1.2 DPIV und OPIV genauer betrachtet . . . 6
  - 1.2.1 Das Prinzip der PIV . . . 6
  - 1.2.2 Technische Entwicklung und Anwendung . . . 7
  - 1.2.3 Wichtige Systemparameter . . . 7
  - 1.2.4 OPIV . . . 8
  - 1.2.5 DPIV . . . 8
  - 1.2.6 Particle Image Velocimetry — Verfahrensmodi . . . 8
- 1.3 Einführung in die DPIV — Verwendung in Theorie und Praxis . . . 9
  - 1.3.1 Kurze Beschreibung der direkten Kreuzkorrelation . . . 11
  - 1.3.2 Kreuzkorrelation via „Fast FOURIER Transform“ . . . 13
  - 1.3.3 Vergleich der PIV mit der direkten numerischen Strömungssimulation 13
- 1.4 Ziel und Zweck dieser Diplomarbeit . . . 14

2 Erweiterte Ansätze zur Verbesserung der DPIV 17
- 2.1 Betrachtungen zur Genauigkeit der PIV und deren Verbesserung . . . 17
  - 2.1.1 Detektion von Spitzen in der Korrelationsebene . . . 17
  - 2.1.2 Erweiterung des Suchfensters . . . 19
  - 2.1.3 Deformation bzw. Verschiebung von Abfrage- und Suchfenster . . . 20
  - 2.1.4 Die kontinuierliche Fensterverschiebung (CWS) . . . 22
  - 2.1.5 Der „peak-locking“-Effekt als wichtigster Fehler der PIV . . . 22
  - 2.1.6 Erhöhung der Vektorfeldauflösung und iterative Verfahren . . . 23
  - 2.1.7 Eine spezielle D-NKK zur Vorbereitung von PIV-Bildern . . . 25

2.1.8 Erkennung von Ausreißern oder „outliers“ 25
2.2 Optimierung auf Geschwindigkeit 27
2.2.1 Kreuzkorrelation via FFT — Vor- und Nachteile 27
2.2.2 Eine schnelle Variante der D-NKK 28
2.2.3 Kombination von FFT-KK und D-NKK — ein hybrider Ansatz 28
2.3 Alternative Ansätze und Anwendungen 29
2.3.1 Ideen aus der PIV-Challenge ’03 29
2.3.2 Ermittlung der Korrelationsspitze mit Hilfe einer Gradientenmethode 30
2.3.3 Eine PIV-Methode mit einer Genauigkeit 2. Ordnung 31
2.3.4 Weitere Ansätze mit hoher Genauigkeit 32
2.3.5 PIV-Analyse durch Merkmalsverfolgung 33
2.3.6 Angepasste Methoden in der μ-PIV 33

3 Erkennung von Merkmalen in Strömungen 35
3.1 Bestimmung der Wirbelstärke 35
3.2 Bestimmung der Deformation 35
3.3 Weitere Merkmale von Strömungen 36

4 Methodische Hilfsmittel aus der digitalen Bildverarbeitung 39
4.1 Hoch- und Tiefpassfilter im Allgemeinen 39
4.2 Der GAUSS’sche Unschärfefilter 40
4.3 Filteroperatoren für die Kantendetektion in Bildern 41
4.3.1 Der SOBEL-Operator 41
4.3.2 Eine durch Wavelets generalisierte, multiskalare Form des CANNY-Operators 42
4.4 Die GAUSS’sche Pyramide 46

5 Hierarchische Ansätze in der DPIV 49
5.1 Hierarchische Verfeinerung des Abfragerasters 49
5.1.1 Optimierung der Effizienz durch Vergröberung bzw. Kompression von Bilddaten 50
5.2 Adaptive Verfeinerung des Abfragerasters 51
5.2.1 Entscheidungskriterien zur Adaption 51
5.3 Probleme und Überlegungen bei multiresoluten Lösungen 54

6 Betrachtung frequenzbasierter Ansätze zur Merkmalsextraktion 57
6.1 Hoch- und Tiefpassfilter zur Extraktion von Merkmalen 57
6.1.1 Das Konzept der Filterhierarchie 58
6.2 Entfernung des Rauschens zur Verbesserung der Korrelation 58
6.3 Effizienzsteigernde, merkmalserhaltende Vergröberung von Bildern 60
6.4 Eine Kurzübersicht über die Verfahrenskomplexitäten 61

II Praxis 63

7 Beschreibung der Implementierung 65
7.1 Definition der Problemstellung . . . 65
7.2 Anforderungsanalyse . . . 66
7.2.1 Szenarienanalyse und Definition von Anwendungsfällen . . . 66
7.2.2 Schnittstellenspezifikation — Das Ausgabeformat . . . 67
7.2.3 Festlegung nichtfunktioneller Anforderungen . . . 67
7.2.4 *PIVA* Anwenderdokumentation . . . 68
7.3 Systemaufbau . . . 73
7.3.1 Allgemeine Überlegungen . . . 74
7.3.2 Überblick aller Systemklassen . . . 74
7.4 Schematische Beschreibung der Kernkomponenten und Algorithmen . . . 79
7.4.1 Die Klasse `PIVAnalysisControl` . . . 79
7.4.2 Die Klasse `FieldReconstructorControl` . . . 81
7.4.3 Die Klasse `CrossCorrelationControl` . . . 81

8 Testexperimente und Ergebnisse 85
8.1 Definition allgemeiner Vergleichsmetriken . . . 85
8.2 Testexperiment 1: Computergenerierte Aufnahmen . . . 87
8.2.1 Informationen zu den Testbildern und Systemparameter . . . 87
8.2.2 Ergebnisse und Bilder . . . 88
8.3 Testexperiment 2: Reale Aufnahmen von mikroskopischen, biologischen Strukturen . . . 94
8.3.1 Informationen zu den Testbildern und Systemparameter . . . 95
8.3.2 Ergebnisse und Bilder . . . 95

9 Ausblick 109
9.1 Verbesserung der Implementierung . . . 109
9.2 Gelerntes . . . 111
9.3 Schlussfolgerungen . . . 113

A Glossar 115

B Erklärung der Symbole und Operatoren 119

C Wichtige Internetadressen 121

Literaturverzeichnis 123

Index 129

Abbildungsverzeichnis

1.1 Das Verfahrensprinzip der DPIV-Technik 6
1.2 YOUNG's Streifen als Ergebnis der OPIV 8
1.3 Doppelt belichtete Einzelaufnahme 9
1.4 Einfach belichtete Doppelaufnahme 9
1.5 Eine Autokorrelationsebene 11
1.6 Eine Kreuzkorrelationsebene 12

2.1 Das Suchfenster relativ zum Abfragefenster vergrößern 19
2.2 Eine Fensterverschiebemethode 0. Ordnung 21
2.3 Die Fensterdeformation 1. Ordnung 24
2.4 Die iterative, hierarchische WIDIM-Methode 24
2.5 Eine diskrete, zweidimensionale Integralfunktion 29
2.6 Die „Central Difference Image Correction"-Methode (CDIC) 32

3.1 Die drei Strömungstypen 37

4.1 Die kontinuierliche GAUSS'sche Glockenkurve 41
4.2 Diskreter GAUSS'scher 5×5-Konvolutionskern $\hat{G}$ $(N = 2)$ 41
4.3 SOBEL-Operator S_X für vertikale Kanten 42
4.4 SOBEL-Operator S_Y für horizontale Kanten 42
4.5 Eine Wavelet-Glättungsfunktion 44
4.6 Die Wavelet-Gradientenfunktion 44
4.7 Eine schematische Darstellung der GAUSS'schen Pyramide 47

5.1 Eine Zone lokaler, adaptiver Verfeinerung 51

6.1 Die Frequenzbänder von PIV-Aufnahmen 59
6.2 Eine Hoch-/Tiefpasshierarchie von Aufnahmen 59
6.3 Die schematische Erkennung von Rauschspitzen 60

7.1 Die gesamte Systemarchitektur 75
7.2 Die Kernarchitektur 79
7.3 Das Sequenzdiagramm für `PIVAnalysisControl` 80

7.4 Das Sequenzdiagramm für `FieldReconstructorControl` 82
7.5 Das Sequenzdiagramm für `CrossCorrelationControl` 83

8.1 Aufnahmepaar Testexperiment 1 87
8.2 Das Referenzvektorfeld des „PIV Standard Image“-Projektes 87
8.3 Das beste Ergebnisvektorfeld in Testexperiment 1 89
8.4 Das Differenzvektorfeld zu Abbildung 8.2 89
8.5 Die Ergebnisse der Testreihe 1 90
8.6 Ergebnis der CWS in Testreihe 1 91
8.7 Das Histogramm zu Testexperiment 1 91
8.8 Die Kantenbildhierarchie in Testexperiment 1 mit Ansatz A 92
8.9 Die Kantenbildhierarchie in Testexperiment 1 mit Ansatz B 93
8.10 Aufnahme in Testexperiment 2 94
8.11 Ein Ergebnisvektorfeld von *PIVA* in der Testreihe 2. ohne Filterung 96
8.12 Ein Ergebnisvektorfeld von *PIVA* in der Testreihe 2. nach Ansatz A 97
8.13 Das Differenzvektorfeld zu Abbildung 8.12 98
8.14 Das beste Ergebnisvektorfeld von *PIVA* in der Testreihe 2. nach Ansatz B . . . 99
8.15 Das Histogramm zu Testexperiment 2 100
8.16 Die Ergebnisse der Testreihe 2 100
8.17 Ein Ergebnisvektorfeld von *PIVA* in Testreihe 2. mit CWS-Methode 101
8.18 Ergebnis der CWS in Testreihe 2 102
8.19 Die Kantenbildhierarchie in Testexperiment 2 mit Ansatz A 105
8.20 Die Kantenbildhierarchie in Testexperiment 2 mit Ansatz B 106
8.21 Ein Vergleich von Bildern mit und ohne Rauschfilterung 107

Tabellenverzeichnis

6.1 Komplexität der PIV-Verfahrensvarianten 62

7.1 Die Kommandozeilenoptionen von *PIVA* 73

8.1 Vorschläge für Testmetriken zur Validierung der Vektorfelder 86
8.2 Die Systemparameter für Testexperiment 1 88
8.3 Die Systemparameter für Testexperiment 2 95
8.4 Testexperiment 2: Die Stichprobenergebnisse bzgl. der Metrik M_2 104

B.1 Erklärung der Symbole und Operatoren 120

Über den inhaltlichen Aufbau dieser Arbeit

Um es dem Leser dieses Schriftstücks zu erleichtern, zu den Stellen seines Interesses so schnell wie möglich zu gelangen, möchte ich kurz auf den roten Faden dieser Arbeit aufmerksam machen.

Abschnitt 1 — Dort findet man eine grundsätzliche Einführung in das Fachgebiet der PIV-Analyse. Es sollen vor allem Informatiker, die sich u. a. mit der digitalen Bildverarbeitung befassen, relativ schnell verstehen, worum es denn aus der Sicht eines Fluidmechanikers und Physikers geht. Um dem Leser die Ideen und Methoden der partikelbasierten Strömungsanalyse näher zu bringen, war es nötig, die aus einer umfangreichen Recherche zusammengetragenen Informationen zu bündeln und im Hinblick auf diese Diplomarbeit strukturiert niederzuschreiben.

Abschnitt 2 — Dieser Teil der Arbeit bietet eine themenbezogene Übersicht etablierter Verfahren. Zum besseren Verständnis dieses Abschnittes ist ein mathematisches und vor allem statistisches Hintergrundwissen sehr von Vorteil.

Abschnitt 3 — Über die Rekonstruktion von Vektorfeldern aus Bilddaten hinausgehend, behandelt dieser Abschnitt die Nutzung von Vektordaten zur Analyse von Strömungsmerkmalen. Die hier erklärte Thematik dürfte vor allem für Strömungsmechaniker sehr interessant sein, geht jedoch nicht so sehr in die Tiefe.

Abschnitt 4 — Die PIV-Analyse ist sehr stark mit der digitalen Bildverarbeitung verbunden. Die Kapitel 4 bis 6 eignen sich für Leser aus dem Fachgebiet der digitalen Bildverarbeitung. Deshalb wurde Kapitel 4 eingeführt, um die digitale Filterung von Bilddaten kurz zu beschreiben. Dieses Kapitel kann ohne weiteres übersprungen werden, sofern man über die Grundlagen der digitalen Bildverarbeitung genügend informiert ist.

Abschnitt 5 — Auch bei der Verarbeitung von PIV-Daten sind in den letzten Jahren zunehmend hierarchische Methoden in den Vordergrund gerückt. In dieser Diplomarbeit wird daraus Nutzen gewonnen, welcher in Abschnitt 5 erklärt wird.

Abschnitt 6 — Ein interessanter Aspekt bei der Verarbeitung von digitalisierten Aufnahmen sind die darin enthaltenen Merkmale, welche vorwiegend bestimmte Frequenzbänder in der Regel jenseits des Rauschens belegen. Im Kapitel 6 wird auf die Beeinflussung der Merkmale durch Rausch-, Hoch- und Tiefpassfilter eingegangen.

Abschnitt 7 — Ein ebenfalls zentraler Teil der Diplomarbeit war die Implementierung eines auf der direkten Kreuzkorrelation basierenden, flexiblen Softwaresystems zur Analyse von PIV-Daten. Dieser Abschnitt wurde hauptsächlich für Informatiker geschrieben und erklärt die Systemarchitektur, die verwendeten Algorithmen und die Zusammenarbeit der Systemkomponenten. Für Programmentwickler sei auf eine gesonderte,

nicht in diese Schrift integrierte, „*doxygen*“-Entwicklerdokumentation (siehe [Hee04]) verwiesen.

Abschnitt 8 — Um die Ergebnisse der Überlegungen und der Implementierung zu veranschaulichen und der Diplomarbeit auch einen validierenden Charakter zu verleihen, werden in Kapitel 8 einige Testszenarien abgebildet.

Abschnitt 9 — Im Zuge der Arbeit und der Implementierung sind Probleme aufgetaucht und diverse Überlegungen haben neue Ideen mit sich gebracht. Abschnitt 9 bespricht diese Themen abschließend. Eine kleine Zahl von weiterführenden Gedankengängen wurde aus Gründen des Zusammenhangs nicht in dieses Kapitel integriert, sondern in den entsprechenden Abschnitten belassen und mit der Markierung **[Idee]** gekennzeichnet. Der Leser mag diese Gedanken je nach Bedarf als Anstoß für eine Verbesserung und Erweiterung der bestehenden Arbeit nutzen oder in eigene Projekte einbauen. Zusätzlich habe ich in Kapitel 9.3 einen kurzen Rückblick angefügt, in dem die wichtigsten Thematiken aus eigener Sicht noch einmal erläutert werden.

Anhang — Abkürzungen und fachspezifische Begriffe werden nicht immer vollständig im Text erklärt bzw. nur bei ihrer ersten Verwendung eingeführt. Bei Unklarheiten verweise ich auf das ausführliche **Glossar** im Anhang. Ferner habe ich für den raschen Überblick eine **Symbol- und Operatorentabelle** angefügt. Der Leser findet eine ausführliche Liste mit **Internetadressen** und **Literaturreferenzen** am Ende dieser Schrift.

Teil I

Theorie

1 Einführung

Das Fachgebiet der Strömungsvisualisierung hat bereits eine lange Geschichte, ist breit gefächert und bietet mittlerweile eine Fülle an Lösungsansätzen

---

Um auf die Thematik dieser Diplomarbeit ausreichend hinzuführen, folgt nun ein kurzer Überblick über die Entwicklung der Methoden zur Strömungsvisualisierung zusammengefasst aus [RWK98]. Dieses Buch behandelt den Fachbereich der PIV-Analyse gesamtheitlich. Die folgenden Teilabschnitte befassen sich mit einer spezifischen Einführung in dieses Gebiet. Einem Leser, welcher in diesem Fachbereich nur geringfügige Kenntnisse hat, empfehle ich, diesen Abschnitt durchzulesen.

1.1 Geschichtliche Entwicklung von partikelbasierten Methoden zur Strömungsvisualisierung und -rekonstruktion

Schon im Jahre 1904 hatte einer der damals bekanntesten Fluidmechaniker, Ludwig PRANDTL, die Idee eines Wasserzirkulationstunnels verwirklicht, in dessen strömendes Medium er winzige Partikel eingestreut und direkt beobachtet hat. Doch in den folgenden neunzig Jahren wurde diese Idee aufgrund mangelnder technischer Grundausstattungen nur mäßig weiterentwickelt. In den Dreißiger Jahren des 20. Jahrhunderts entwickelte man erste Verfahrensansätze. Eine praktikable Umsetzung dieser initialen Idee zur Erzielung qualitativ und quantitativ verwendbarer Ergebnisse geschah aber erst in den letzten 15 bis 20 Jahren. Mit entsprechend hochtechnisierten Anlagen konnte in den Achzigern eine Visualisierung und Analyse komplexer Strömungsverhältnisse, wie sie schon in jedem einfacheren Experiment auftreten, ermöglicht werden.

Frühe Herangehensweisen waren sogenannte Einzelpunkt-Messmethoden wie z. B.

- die Laser-Doppler-Anemometrie (LDA) oder die
- Heißdraht-Anemometrie (HWA).

Diese Verfahren und davon abgeleitete Varianten ermöglichen keine besonders gute räumliche und zeitliche Auflösung von Geschwindigkeitsdaten. Ein weit besserer Ansatz, abgeleitet von den einstigen Ideen von PRANDTL, wurde mit der sogenannten „Particle Image Velocimetry“ — im Folgenden kurz **PIV** und zu Deutsch „Partikelbild-Geschwindigkeitsmessung“ — Anfang der 80er Jahre eingeführt. Anfangs hat sich die vollständig optische Verfahrensvariante „Optical PIV“ oder OPIV etabliert. Mit der Weiterentwicklung der digitalen Bildverarbeitung hat sich die „Digital PIV“ oder DPIV immer mehr in den Vordergrund geschoben. Im Abschnitt 1.2 wird auf diese beiden Verfahrensvarianten näher eingegangen.

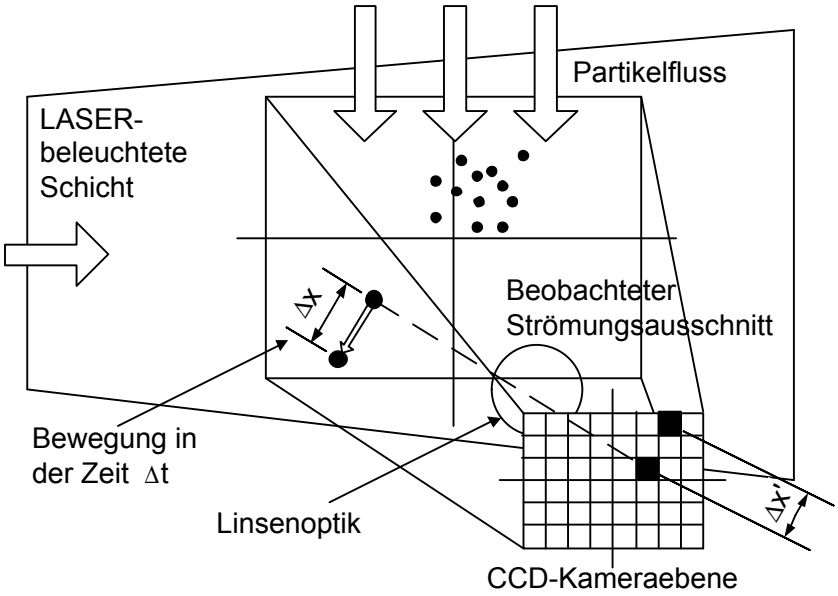

Abbildung 1.1: Das Verfahrensprinzip der DPIV-Technik

Mit der Arbeit von R. J. ADRIAN erhält man eine detaillierte Übersicht über die geschichtliche Entwicklung der PIV; siehe [Adr96]. Einen einführenden Überblick über die damaligen Verfahren der digitalen Bildverarbeitung in der Fluidvisualisierung bietet [Hes88]. Die Doktorarbeit von J. WESTERWEEL [Wes93] betrachtet das aktuelle Thema aus der theoretischen Sicht, beginnt aber mit einer umfassenden Einführung.

1.2 DPIV und OPIV genauer betrachtet

1.2.1 Das Prinzip der PIV

In der Abbildung 1.1 findet man den schematischen PIV-Versuchsaufbau. In ein strömendes Medium werden Partikel eingestreut. Diese bewegen sich idealerweise in Richtung der Stromlinien. An der zu beobachtenden Stelle (z. B. einer Stufe) wird das in gewissem Maße transparente Medium mit einem kohärenten Laser-Lichtvorhang pulsiert beleuchtet. Synchron mit dem Laser-Puls nimmt eine senkrecht auf den Laser-Vorhang gerichtete Bilderfassungseinrichtung (z. B. eine CCD-Kamera) die momentanen Positionen der das Laserlicht reflektierenden Partikel innerhalb des dünnen Lichtvorhangs auf. Die Partikel werden stroboskopisch — also gepulst — beleuchtet, da sonst während der Aufnahme verzerrte Partikelbilder entstehen. Durch eine zeitlich festgelegte Folge von Aufnahmen erhält man Informationen von Partikelbildmustern über Ort und Zeit. Diese Aufnahmen können mit statistischen Methoden wie der Auto- oder Kreuzkorrelation (siehe Abschnitt 1.3.1) untersucht werden. Als Ergebnis lassen sich gewisse Teile der Bilder miteinander korrelieren bzw. einander zuordnen, und zwischen diesen lässt sich eine relative Verschiebung bestimmen. In Folge — Ort und Zeit sind bekannt — erhält man eine flächendeckende Geschwindigkeitsverteilung in der beobachteten Region. Bei der herkömmlichen PIV können allerdings nur Verschiebungen von zusammengehörenden Partikelbildmustern in den zwei Richtun-

gen der Ebene innerhalb des Lichtvorhangs registriert werden. PIV zählt zu den nichtberührenden Verfahren, d. h. die aus den Beobachtungen ermittelten Daten werden i. A. durch das Verfahren selbst nicht verzerrt.

Es haben sich inzwischen auch Verfahren mit mehreren Bilderfassungsgeräten entwickelt (z. B. Stereo-PIV mit zwei CCD-Kameras), die es ermöglichen, auch die dritte räumliche Verschiebungskomponente zu registrieren; siehe [HSK+01].

1.2.2 Technische Entwicklung und Anwendung

Anfang der Achziger Jahre hat der Fluidmechaniker Roland MEYNART vom Karman-Institut in Belgien gezeigt, dass man PIV praktisch anwenden kann. PIV wird heute hauptsächlich in der Fluidmechanik für die Beobachtung von Wasser- und Luftströmungen verwendet. Zur genaueren Analyse von Flüssigkeiten ist dieses Verfahren schon seit längerer Zeit ausgereift. Für aerodynamische Anwendungen und komplizierte Strömungsverhältnisse ließ es sich zunächst jedoch kaum nutzen.

Doch mit der Anwendung von Nd:YAG-Lasern und hochfrequenten Pulsraten wurden auch bei Messungen von Luftströmungen mit sehr kleinen Partikeln und hohen Partikeldichten zunehmend bessere Ergebnisse erzielt. Bessere Streutechniken, die Anwendung der Kreuzkorrelation anstatt der Autokorrelation (siehe Abschnitt 1.3.1) und immer leistungsstärkere (digitale) Bildverarbeitungssysteme zur Verarbeitung großer Datenmengen führten zum entgültigen Durchbruch der PIV. Früher wurde vorwiegend intuitiv mittels „Trial & Error“-Methoden vorgegangen, während die theoretischen Hintergründe und Zusammenhänge heute weitgehend bis ins Detail erforscht sind; siehe [Wes93].

1.2.3 Wichtige Systemparameter

Die folgende Auflistung erklärt die wichtigsten Systemparameter, welche einen signifikanten Einfluss auf die Qualität und Quantität der Ergebnisse einer PIV-Analyse haben:

- Die *zeitliche Auflösung* der Aufnahmen ermöglicht die Beobachtung feiner Veränderungen der Strömungsstrukturen.
- Die *räumliche Auflösung* der Analyse der Aufnahmen entscheidet über die Skalengröße der beobachtbaren Strukturen und den Bereich der erfassbaren Partikelmusterverschiebungen.
- Die *Partikelstreudichte* und der *Partikeldurchmesser* kann negativen Einfluss auf die Dynamik der Strömung haben.
- Die *Beleuchtungsstärke und -dauer* des Lasers hat Einfluss auf die Bildqualität und folglich auf die Genauigkeit der ermittelten Verschiebungen und Geschwindigkeiten.
- Eine zu geringe *Auflösung der Bilderfassungseinrichtung* beschränkt die Genauigkeit der Analyse.
- Die *digitale Bildqualität* bzw. der Anteil an *Bildmerkmalen* und *Rauschen* entscheiden über die Detektierbarkeit von Strömungsmerkmalen.

Im später behandelten praktischen Teil II werden u. a. einige dieser Faktoren durch die implementierte Software namens *PIVA* abgebildet.

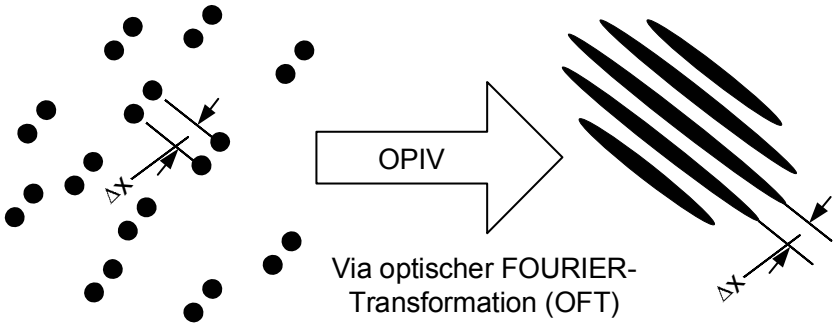

Abbildung 1.2: YOUNG's Streifen als Ergebnis der OPIV

1.2.4 OPIV

Die bereits oben erwähnte, rein optische Verfahrensvariante OPIV beruht auf dem Prinzip der optischen FOURIER-Transformation. Die in Abbildung 1.1 gezeigte Einrichtung erzeugt Negativaufnahmen. Diese werden mittels Laser-Licht durchleuchtet und fungieren als sogenanntes Beugungsgitter, um dadurch auf einer hinter der PIV-Aufnahme liegenden Ebene helle und dunkle Streifen (siehe Abbildung 1.2) abzubilden. Diese Streifen werden auch „YOUNG's Streifen" genannt und geben Aufschluss über die Richtung und Größe von Musterverschiebungen. Einfache Messungen direkt auf den Streifen sind zwar sehr schnell, führen aber leider zu ungenauen Ergebnissen. Deshalb hat man diese Streifenbilder digitalisiert und mittels schneller FOURIER-Transformation (FFT) analysiert, um entsprechende Verschiebungsinformationen aus Spitzen im Frequenzbereich zu lesen. Damit geht der Vorteil einer sehr schnellen optischen Analyse wieder verloren. [Wes93] und [RWK98] beschreiben dieses Verfahren genauer.

1.2.5 DPIV

Um die Nachteile der OPIV und den zusätzlichen Aufwand der kosten- und zeitintensiven Fotoentwicklung zu kompensieren, ging man dazu über, eine FFT-Analyse direkt auf digitalen Aufnahmen der Partikel — erzeugt durch eine CCD-Kamera — zu machen. Im Abschnitt 1.3 wird auf das DPIV-Verfahren detailliert eingegangen.

1.2.6 Particle Image Velocimetry — Verfahrensmodi

Je nach Streudichte der Partikel und der damit resultierenden Charakteristika der Aufnahmen unterscheidet [Wes93] drei Verfahrensvarianten, deren Begriffe hier nicht ins Deutsche übersetzt werden:

Particle Tracking Velocimetry (PTV): Die Partikeldichte ist sehr gering, sodass es zum einen keine Überschneidungen zwischen den Partikelbildern gibt und zum anderen möglich ist, einzelne **Partikel** in ihrer Laufbahn zu **verfolgen**. Dieser Modus ist laut [RWK98] wegen der geringen Partikeldichte für Luftströmungsmessungen gut geeignet. Des weiteren werden aufgrund der nötigen hohen Bildauflösungen vielfach analoge Bilderfassungssysteme verwendet.

Particle Image Velocimetry (PIV): Hier kann eine Überlappung der Partikelbilder vorkommen, und deshalb eigenen sich **statistische Methoden** gut zur Geschwindigkeitsmessung. PIV hat in manchen Anwendungen Vorteile gegenüber PTV, da hier keine einzelnen Partikel beobachtet werden müssen.

Laser Speckle Velocimetry (LSV): Die Partikeldichte ist bei der LSV sehr hoch und die Beleuchtung führt zu Interferenzeffekten auf den Aufnahmen, sodass einzelne Partikel nicht mehr erkennbar sind. Verschiebungen können somit nur mehr **musterbasiert** („speckle") durch angepasste Methoden aus der PIV bestimmt werden.

1.3 Einführung in die DPIV — Verwendung in Theorie und Praxis

Bis jetzt haben wir uns nur recht oberflächlich mit der Ermittlung von Verschiebungen und Geschwindigkeitsmessungen auf der Grundlage von digitalisierten Aufnahmen von Partikelbildern befasst. Es gibt grundsätzlich mehrere Arten von Partikelaufnahmen in Strömungen (siehe Abbildungen 1.3 und 1.4).

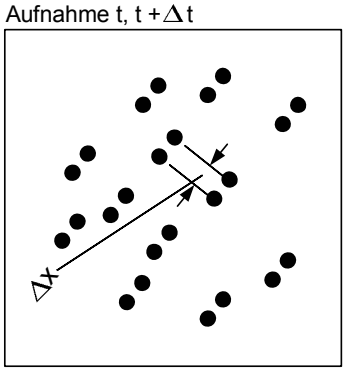

Abbildung 1.3: Doppelt belichtete Einzelaufnahme zur Autokorrelation

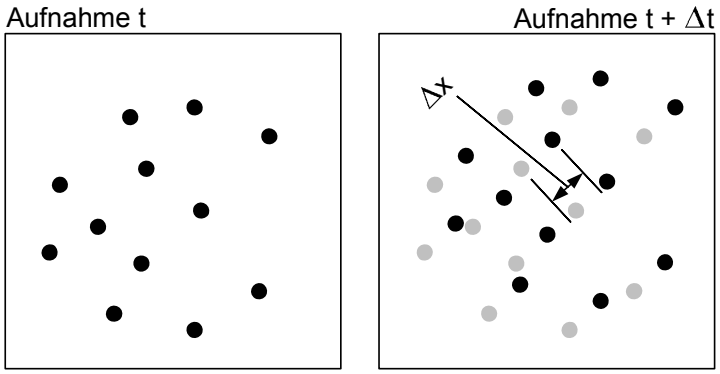

Abbildung 1.4: Einfach belichtete Doppelaufnahme (Aufnahmepaar) zur Kreuzkorrelation

Typ A: Zwei- oder auch mehrfach belichtete Einzelaufnahmen beinhalten mehrere Abbilder derselben Partikel zu verschiedenen Zeitpunkten, eine sogenannte *Partikelspur*.

Typ B: Einfach belichtete Doppel- oder Mehrfachaufnahmen bestehen aus mehreren Einzelbildern und enthalten nur die momentanen Positionen von Partikeln zu einer bestimmten Zeit.

Im Folgenden wird eine Aufnahme des Typs A als Leuchtintensitätsfeld $I(x,y)$ und Aufnahmen des Typs B werden als Leuchtintensitätsfelder $I(x,y)$ und $S(x,y)$ [1] bezeichnet und durch die Definitionsmenge $\mathcal{B} = \{(x,y)|0 \leq x < H \wedge 0 \leq y < B \wedge H,B \in \mathbb{N}\}$ und die Abbildung $I,S : \mathcal{B} \mapsto \mathbb{R}^2$ definiert, wobei H bzw. B die Höhe bzw. Breite der diskreten, digitalisierten

[1] In dieser Arbeit werden nur Doppelaufnahmen behandelt.

Aufnahmen bezeichnen. Im Falle des Typs B liefern I und S also für ein $(x,y) \in \mathcal{B}$ den zugehörigen reellen Helligkeitswert der Aufnahme an der Stelle (x,y). In der Praxis handelt es sich beim Helligkeitswert lediglich um einen diskreten 8-Bit Grauwert im Bereich $[0,255]$.

Anschaulich erklärt in [Bro99] geht es nun darum, einzelne Bereiche in diesen Bildern zu analysieren und zu vergleichen. Man teilt eine Aufnahme o. B. d. A. in einen regelmäßigen Raster der Größe $X \times Y$ mit $X \cdot N = B$ und $Y \cdot N = H$ ein. Für Paare $(\hat{I}, \hat{S})$ von Rasterzellen mit $\hat{I} \subset I$ und $\hat{S} \subset S$ und der Definitionsmenge $\hat{\mathcal{B}} = \{(m,n) | 0 \leq m < N \wedge 0 \leq n < N \wedge N = 2 \cdot M + 1 \wedge N, M \in \mathbb{N}\} \subset \mathcal{B}$ mit $N \times N$ Pixeln [2] wird eine Analyse vollzogen.

Für beide Typen von Aufnahmen gibt es zwei ähnliche statistische Methoden, um korrelierende[3] Teile von Bildern zu erkennen. Diese Methoden basieren auf der Berechnung von sogenannten Korrelationskoeffizienten und werden häufig in der digitalen Bildverarbeitung[4] verwendet. Dabei wird zwischen der

- **Autokorrelation** für Bilder des Typs A — eine Rasterzelle $\hat{I}_{X,Y} \subset I$, wobei X und Y die Stelle im Raster bezeichnen, wird mit sich selbst korreliert — und der

- **Kreuzkorrelation** für Aufnahmen des Typs B — die Rasterzelle $\hat{I}_{X,Y}$ des ersten Bildes (im Folgenden bezeichnet als **Abfragefenster**) wird mit der Rasterzelle $\hat{I}_{X,Y}$ des zweiten Bildes (im Folgenden bezeichnet als **Suchfenster**) korreliert —

unterschieden.

Wir werden im Rest dieser Schrift nur mit der Kreuzkorrelationsmethode arbeiten. Die Berechnung eines Korrelationskoeffizienten (siehe Gleichung 1.1[5]) erfolgt an jeder Stelle $(m,n) \in \hat{\mathcal{B}}$ einer Rasterzelle $\hat{I}_{X,Y}$. Somit wird eine sogenannte Korrelationsebene der Größe N^2 berechnet, welche mit hoher Wahrscheinlichkeit an einer einzigen Stelle ein eindeutiges, globales Maximum besitzt. [Wes93] hat Schätzer für diese Wahrscheinlichkeiten abhängig von diversen Parametern errechnet. WESTERWEEL's generelle Annahme besteht darin, dass sich das Partikelbildmuster innerhalb des Suchfensters homogen bewegt. Hier wird aber nicht weiter auf diese Problematik eingegangen.

Auf die Autokorrelation möchte ich hier ebenfalls nicht weiter eingehen, da diese nicht Bestandteil der Thematik dieser Diplomarbeit ist. Die Autokorrelation hat den Nachteil, dass es nicht nur eine Spitze in der Korrelationsebene gibt, sondern wie in Abbildung 1.5 ersichtlich, eine Autokorrelationsspitze im Zentrum der Ebene und zwei symmetrisch zur Mitte liegende Spitzen, welche die quantitative Verschiebung zwischen beiden Belichtungen kennzeichnen. Damit erhält man eine Zweideutigkeit und kann folglich die Richtung des Geschwindigkeitsvektors nur mit Vorkenntnis der Strömungsrichtung eindeutig bestimmen. Der Vorteil der Autokorrelation bestand früher darin, dass man weniger Aufnahmen machen musste und somit Entwicklungszeit bzw. Speicher sparen konnte. Genaueres zu den beiden statistischen Methoden findet man auch in der analytischen Schrift [KA92], welche die Kreuzkorrelation zusammenfassend als besser geeignet für PIV ansieht.

In [HFW92a] wird die Kreuzkorrelation in der sogenannten „Particle-Image-Pattern-Matching"-Technik (PIPM) angewendet.

[2] Hier werden o.B.d.A. nur quadratische Regionen betrachtet.
[3] zusammengehörende
[4] Pattern Matching
[5] im Falle der Kreuzkorrelation

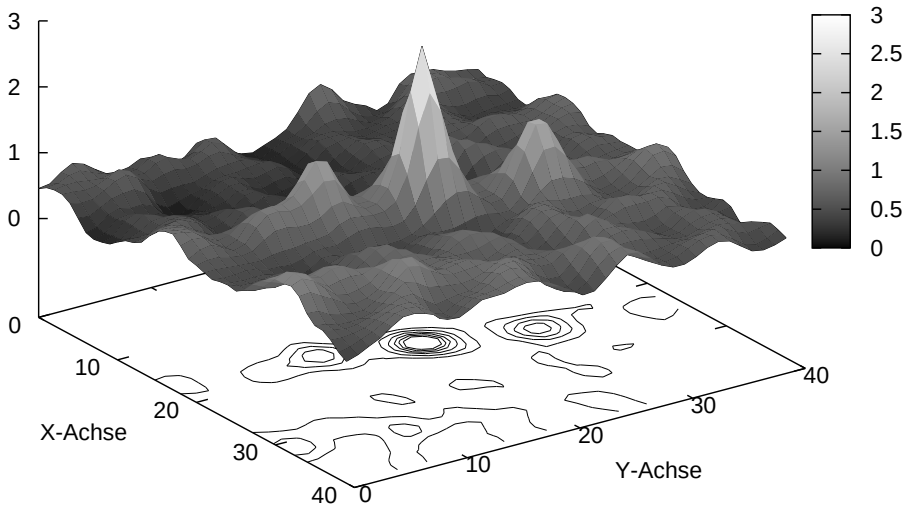

Abbildung 1.5: Eine Autokorrelationsebene mit einer zentralen Autokorrelationsspitze und zwei symmetrisch gegenüber liegenden Spitzen in der Entfernung der Partikelbildverschiebung. Dies führt zu einer Richtungszweideutigkeit. Auf der Z-Achse wird $R_{\hat{I}\hat{S}}(m,n)$ abgebildet. Der Ursprung $(0,0)$ des Kreuzkorrelationstupels (m,n) liegt hier bei $(20,20)$.

1.3.1 Kurze Beschreibung der direkten Kreuzkorrelation

Wie bereits erwähnt, erhält man durch die Kreuzkorrelation (KK) zwischen zwei einfach belichteten Aufnahmen eine eindeutige Spitze in der Korrelationsebene (siehe Abbildung 1.6). Die Position dieser in der Korrelationsebene relativ zum Zentrum kann direkt als Verschiebung des Partikelbildmusters interpretiert werden. Hierbei ist zu beachten, dass in der Korrelationsebene nur Werte an diskreten Stellen bekannt sind und vorerst nur ganzzahlige Verschiebungen ermittelt werden können.

Die Berechnung eines normalisierten Kreuzkorrelationskoeffizienten zwischen Abfragefenster $\hat{I}_{X,Y}$ und Suchfenster $\hat{S}_{X,Y}$ auf der Position (X,Y) im Raster der Aufnahme und an der Stelle $(m,n) \in \hat{\mathcal{B}}$ zwischen den Signalen im einfachen diskreten Fall gestaltet sich wie folgt (siehe [WG91]) [6]:

$$R_{\hat{I}\hat{S}}(m,n) = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \hat{I}(i,j)\hat{S}(i-m,j-n)}{\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \hat{I}^2(i,j) \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \hat{S}^2(i,j)}} \tag{1.1}$$

Hier ist $(m,n) \in \{(m,n) | -M+1 \leq m,n \leq N+M-1 \wedge N = 2 \cdot M+1\}$ und es gilt $\hat{S}(i,j) = 0$ wenn $i,j > N-1 \vee i,j < 0$. Der Normalisierungsterm als Nenner erzeugt den Wertebereich $R_{\hat{I}\hat{S}}(m,n) \in [0,1]$. Auf die genaue Berechnung der Verschiebung und der Geschwindigkeit wollen wir in Abschnitt 2.1.1 näher eingehen.

[6] Die Rasterindizes (X,Y) für $\hat{I}$ und $\hat{S}$ wurden zur Abkürzung weggelassen.

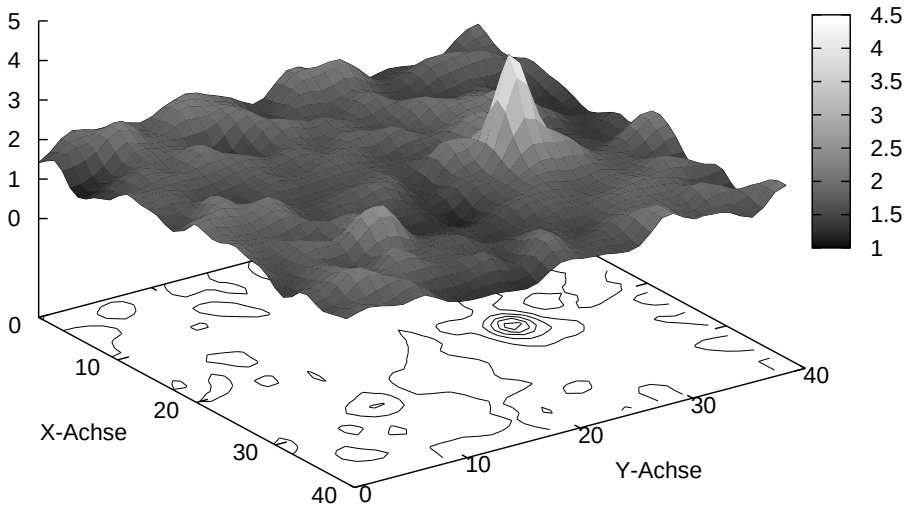

Abbildung 1.6: Eine Kreuzkorrelationsebene mit eindeutiger Spitze $\max_{m,n}\{R_{\hat{I}\hat{S}}(m,n)\}$. Auf der Z-Achse wird $R_{\hat{I}\hat{S}}(m,n)$ abgebildet. Der Ursprung $(0,0)$ des Kreuzkorrelationstupels (m,n) liegt hier bei $(20,20)$.

Ein kurzer Statistikexkurs

Zur Erklärung der bereits öfter verwendeten Begriffe aus der mathematischen Statistik erscheint mir hier ein kurzer Exkurs passend. Die Bezeichnungen in der Literatur sind teils inkonsistent. [Wes93] spricht ausschließlich von der Auto- und Kreuzkovarianz. Diese Begriffe stellen den Grad des Zusammenhangs bzw. der Korrelation zwischen zwei nicht notwendig verschiedenen Zufallsvariablen A und B dar. $\hat{A}$ und $\hat{B}$ bezeichnen im Folgenden Mengen von Zufallsvariablen. Wie in den meisten Abhandlungen über PIV verwende ich diese Terme in der vorliegenden Schrift wie folgt:

- Die *Kovarianz* wird berechnet durch

  $$\mathrm{cov}\{A,B\} = \mathrm{E}\{(A-\mu_A)(B-\mu_B)\}$$

  Hier sind A bzw. B die Zufallsvariablen und $\mu_A = \mathrm{E}\{A\}$ bzw. $\mu_B = \mathrm{E}\{B\}$[7] die Mittelwerte. $\mathrm{E}\{A\}$ ist der sogenannte Erwartungswert einer Zufallsvariablen. Bei $\mathrm{cov}\{A,B\} = 0$ sind A und B unkorreliert. Ist $\mathrm{cov}\{A,B\} > 0$, so tendiert B zu wachsen, wenn A wächst bzw. gilt $\mathrm{cov}\{A,B\} < 0$, so tendiert B zur Abnahme, wenn A abnimmt.

- Als *Kreuzkovarianz* (D-KK) wird die Berechnung von $cov\{A_i,B_j\}$ für alle möglichen Paare $(A_i,B_j) \in \{(A_i,B_j)|A_i \in \hat{A} \wedge B_j \in \hat{B}\}$ bezeichnet.

- Die *Autokovarianz* kann damit durch $\mathrm{cov}\{A_i,A_j\}$ ausgedrückt werden, mit $A_i,A_j \in \hat{A}$.

- Die *Korrelation* gilt hier als eine **normalisierte Form der Kovarianz** und ist definiert durch

  $$\mathrm{cor}\{A,B\} = \frac{\mathrm{cov}\{A,B\}}{\sqrt{\mathrm{var}\{A\}\cdot\mathrm{var}\{B\}}} \equiv \frac{\mathrm{cov}\{A,B\}}{\sigma_A\cdot\sigma_B} \quad \text{mit} \quad -1 \le \mathrm{cor}\{A,B\} \le 1.$$

[7] Eine empfehlenswerte, theoretische Einführung in die mathematische Statistik und Wahrscheinlichkeitsrechnung findet man z. B. unter http://mathworld.wolfram.com/topics/ProbabilityandStatistics.html.

- Als *Kreuzkorrelation* (D-NKK) wird die Berechnung von $cor\{A_i, B_j\}$ für alle möglichen Paare $(A_i, B_j) \in \{(A_i, B_j) | A_i \in \hat{A} \wedge B_j \in \hat{B}\}$ bezeichnet. Dabei nennt man $cor\{A_i, B_j\}$ den Kreuzkorrelationskoeffizienten zwischen den Zufallsvariablen A_i und B_j.
- Die *Autokorrelation* ist somit definiert durch $\mathrm{cor}\{A_i, A_j\}$, mit $A_i, A_j \in \hat{A}$.

1.3.2 Kreuzkorrelation via „Fast FOURIER Transform“

Die Idee der FOURIER-Transformation aus der OPIV kann in der DPIV in Form der FFT wieder aufgegriffen werden, um das Zeitproblem $O(N^4)$ der oben erklärten direkten Kreuzkorrelation auf $O(N^2 + N^2 \log_2 N^2)$ zu reduzieren. N sei hier die Größe des Abfragefensters und eine 2er-Potenz. Mit Hilfe des Kreuzkorrelationstheorems[8] lässt sich die Kreuzkorrelation mittels FOURIER-Transformation wie folgt berechnen:

$$R_{\hat{I}\hat{S}} = \mathcal{F}^{-1}\{\mathcal{F}(\hat{I}) \cdot \mathcal{F}^*(\hat{S})\} \tag{1.2}$$

$\mathcal{F}$ bezeichnet die FOURIER-Transformation, $\mathcal{F}^{-1}$ deren Inverses und $\mathcal{F}^*$ deren komplex-konjugiertes Ergebnis. Dieses Theorem ist folgendermaßen zu verstehen: Bildet man die inverse FOURIER-Transformierte einer komplex-konjugierten Multiplikation der FOURIER-transformierten Bildsignale, so erhält man direkt die Kreuzkorrelationsfunktion.

Wir werden auf diese Variante der Kreuzkorrelation — von jetzt an bezeichnet als FFT-KK — in Abschnitt 2.2.1 noch genauer eingehen. Die direkte Kreuzkorrelation wird von nun an mit D-NKK abgekürzt, ohne Normalisierungsterm durch D-KK.

1.3.3 Vergleich der PIV mit der direkten numerischen Strömungssimulation

Eigenschaften von PIV:

+ Quantitative Visualisierung und Analyse großer Datenmengen mittels digitaler Bildverarbeitung möglich.
+ Die Analyse von komplexen, realen und sogar turbulenten Strömungsverhältnissen ist mittlerweile möglich.
– Der Aufbau von teuren Versuchsanlagen mit kostenintensiven Geräten ist nötig.

Eigenschaften von DNS:

+ Es gibt mittlerweile eine große Zahl von schnellen und spezialisierten Strömungslösern.
+ Der Versuchsaufbau ist weniger kostspielig und die Qualität der Visualisierung sehr hoch.
– Eine direkte numerische Simulation auf Grundlage der die physikalischen Fluideigenschaften beschreibenden NAVIER-STOKES-Gleichungen ist trotz der heutigen sehr leistungsfähigen Rechenanlagen nur beschränkt anwendbar. Zum einen muss selbst bei der Betrachtung von kleineren Strukturen ein relativ großes Gebiet berechnet werden,

[8] Siehe http://mathworld.wolfram.com/Cross-CorrelationTheorem.html.

was je nach Auflösung des Berechnungsgitters aufwändig sein kann. Zum anderen sind die Randbedingungen nicht immer passend annehmbar.

– Ein sehr viel gravierenderes Problem bei einer hochwertigen Simulation, im Gegensatz zu simplen, laminaren Stufenströmungen, stellen die nicht hinreichend gut simulierbaren, turbulenten Strömungscharakteristika in verzweigten Gebieten vor allem bei hohen Reynoldszahlen dar. Randbedingungen können oft nicht genügend realistisch vereinbart werden.

1.4 Ziel und Zweck dieser Diplomarbeit

Der erste Schritt bestand darin, die bisherigen Methoden und Ansätze im Fachbereich PIV hinsichtlich Genauigkeit und Effizienz zu beleuchten. Eine umfangreiche Recherche brachte eine stattliche Anzahl von verschiedenen Ansätzen und Fallstudien aus der Fluidmechanik ans Tageslicht, von denen die für dieses Diplomprojekt wichtig Erscheinenden in den folgenden Abschnitten kurz behandelt werden. In den diversen Teilen dieser Schrift findet man Hinweise auf den Einfluss der jeweiligen Ansätze auf die Genauigkeit und die Effizienz der PIV-Analyse. Ferner wurden auch Vorschläge für eine günstige Kombination von Verfahrensvarianten und -ansätzen ausgearbeitet, welche der Leser in den verschiedenen Kapiteln wieder findet, insbesondere in Abschnitt 9.

Als erstes Ziel dieser Arbeit galt die Entwicklung und Implementierung einer möglichst flexiblen, parametrisierten Software *PIVA*[9] zur Rekonstruktion der Geschwindikeitsvektorfelder aus den PIV-Aufnahmen. Die Software soll aktuelle Ansätze aus dem PIV-Bereich sinnvoll kombinieren, um so diverse Szenarien testen zu können. Des weiteren soll sie hierarchische, adaptive Ansätze realisieren (siehe Abschnitt 5) und hinsichtlich noch nicht realisierter Methoden erweiterbar sein. Benötigte Verfahren der digitalen Bildverarbeitung sollen ebenfalls im System integriert sein.

Ein zweites Ziel dieser Arbeit war es, ein möglichst genaues, jedoch schnelles Verfahren für qualitativ schlechte Aufnahmen ohne die für PIV charakteristischen Partikelbildmuster zu finden. Im Gegensatz zu den auf Standardaufnahmen[10] (siehe [ONSK00]) sehr gut funktionierenden Methoden gibt es z. B. bei Aufnahmen aus dem medizinisch-biologischen Bereich gravierende Probleme mit der Genauigkeit[11]. In diesen Bildern sind keine gut extrahierbaren Partikelbilder vorhanden, und zusätzlich tritt in den Bildsignalen oft unerwünschtes Rauschen auf. Das erschwert die genaue Ermittlung von Verschiebungen und Geschwindigkeiten. Diese Aufnahmen müssen folglich vorbereitet bzw. durch spezielle Verfahren behandelt werden, ohne dabei signifikante Merkmale durch die Entfernung bzw. Filterung des Rauschens zu verlieren. Das Wort „modellbasiert“ im Titelthema dieser Diplomarbeit bezieht sich also auf die Spezialisierung der Thematik auf die eben beschriebene Bildklasse und die Disziplin der μ-PIV.

Es stellte sich also die **Frage**: Kann man hinreichend genaue Ergebnisse mit effizienten Methoden erzielen und/oder gehen durch das (Rausch-)Filtern wichtige Merkmale der beobachteten Strukturen verloren? Hierbei handelt es sich um Strukturen, welche für eine

[9] Particle Image Velocimetry Analyser

[10] Künstliche Versionen von klassischen PIV-Aufnahmen sind vom „PIV Standard Image“-Projekt unter `http://vsj.or.jp/piv/` erhältlich und können dort auch generiert werden.

[11] Hier gibt es vor allem im Bereich des μ-PIV einige Forschungsprojekte.

gewinnbringende Analyse der Topologie einer Strömung aus den rekonstruierten Vektorfeldern wichtig sind.

Mit hierarchischen Methoden, dem kombinierten Einsatz von Hoch- und Tiefpassfiltern und kontinuierlichen Verfahren aus der Bildverarbeitung (Sub-Pixel-Interpolation) wurden diverse Tests vollzogen und Ergebnisse hinsichtlich der Qualität der Geschwindigkeitsdaten notiert (siehe Abschnitt 8).

2 Erweiterte Ansätze zur Verbesserung der DPIV

Die digitale Analyse von PIV-Daten hat sich im letzten Jahrzehnt stark weiterentwickelt und hält für spezielle Anwendungen mit hohen Anforderungen optimierte Verfahrensvarianten parat

---

Die einführende Erklärung des PIV-Grundprinzips geht von einer einfachen, intuitiven Methode zur Ermittlung der Verschiebungen aus. Um allerdings in der Praxis verwendbare Ergebnisse zu erzielen, sind weiterführende Überlegungen nötig. Ich möchte darauf hinweisen, dass sich auch hier eine lange Reihe von Wissenschaftlern vor allem aus dem Bereich der Fluidmechanik Gedanken gemacht hat und diese in zahlreichen Artikeln und Forschungsberichten veröffentlicht hat. Die folgenden Abschnitte geben eine Übersicht dieser Ansätze und beleuchten vor allem die Vor- und Nachteile im Hinblick auf die Zielvorstellungen dieser Diplomarbeit. Es sei vorweg genommen, dass die hier erklärten Methoden aus Gründen des zeitlichen Aufwandes nicht alle vollständig in der Testsoftware *PIVA* implementiert worden sind. Im Folgenden wird aber in Kombination mit einer entsprechenden Erklärung gesondert hingewiesen. Ebenso erhebt dieser Abschnitt keinen Anspruch auf Vollständigkeit bzgl. der existierenden Lösungsansätze im Bereich der PIV. Im praktischen Teil der Diplomarbeit (siehe Teil II), in welchem auf einige Details der Implementierung eingegangen wird, finden sich entsprechende Verweise auf die nun gezeigten Methoden.

Im Teilabschnitt 2.1 werden direkte und indirekte Ansätze zur Verbesserung der Genauigkeit der Ergebnisse der Kreuzkorrelation erklärt. Der Abschnitt 2.2 bietet Überlegungen zur effizienteren Gestaltung der Berechnungen unter Berücksichtigung der Erhaltung der Genauigkeit.

2.1 Betrachtungen zur Genauigkeit der PIV und deren Verbesserung

2.1.1 Detektion von Spitzen in der Korrelationsebene

Wie bereits in Abschnitt 1.3.1 erklärt, können wir aus der Kreuzkorrelationsebene mit hoher Wahrscheinlichkeit ein globales Maximum ermitteln, das sich aber noch an einer diskreten Stelle befindet. Da jedoch in der Realität kaum diskrete Ergebnisse vorkommen, bietet sich ein interpolierender Ansatz zur Ermittlung des reellen Anteils der Verschiebung, d. h. zur genaueren Bestimmung dieser Spitze an. Nun nehmen wir an, dass eine vorläufige Spitze $R_{\hat{f}\hat{S}}(m_0, n_0)$ an der Stelle $(m_0, n_0) \in \mathbb{N}^2$ der Kreuzkorrelationsebene mit Hilfe einer zweidimensionalen $O(N^2)$-Suche in der gesamten Korrelationsebene gefunden wurde und definieren

nun die Hilfsvariable $R_m \equiv R_{\hat{I}\hat{S}}(m_0+m,n_0)$ zur abkürzenden Beschreibung der horizontalen[1] Nachbarn von $R_{\hat{I}\hat{S}}(m_0,n_0)$ mit $m \in \{-1,0,1\}$. Somit setzt sich die reelle Verschiebung $\vec{r} \stackrel{!}{=} (r_D,s_D)^T$ aus

$$r_D = m_0 + \varepsilon_m \quad \text{und} \quad s_D = n_0 + \varepsilon_n \tag{2.1}$$

zusammen, wobei die Tupel (m_0,n_0) bzw. $(\varepsilon_m,\varepsilon_n)$ den ganzzahligen bzw. reellen Anteil der Partikelbildverschiebung bezeichnen.

Neben anderen, ähnlichen Methoden betrachtet [Wes93] zur Bestimmung von $-0.5 < \varepsilon_m,\varepsilon_n \leq 0.5$ drei verschiedene 3-Punkt-Schätzermethoden, nämlich

- den „center-of-mass"-Schätzer,
- den „parabolic-fit"-Schätzer und
- den „Gaussian-fit"-Schätzer.

Zwei dieser Schätzer wurden in der Software *PIVA* realisiert, wobei der „center-of-mass"-Schätzer dem „Gaussian-fit"-Schätzer vorgezogen wurde, wenn Zweiter nicht anwendbar war (siehe Gleichung 2.3). O.B.d.A. betrachten wir die Berechnungen nur in horizontaler Richtung (m_0). Der „center-of-mass"-Schätzer ist universell einsetzbar und definiert durch

$$\varepsilon_{C_m} = \frac{R_{+1} - R_{-1}}{R_{-1} + R_0 + R_{+1}}. \tag{2.2}$$

Der „Gaussian-fit"-Schätzer, der eine GAUSS'sche Glocke unter die drei Punkte setzt und von [Wes93] als Bester bewertet wird, hat die beiden Einschränkungen, dass $R_0 > R_{\pm 1}$ und $R_i > 0$ für $i = -1,0,1$ und ist definiert durch

$$\varepsilon_{G_m} = \frac{\ln R_{+1} - \ln R_{-1}}{2(\ln R_{-1} + \ln R_{+1} + 2\ln R_0)}. \tag{2.3}$$

Mit dieser Methode hat man nun eine relativ genaue Bestimmung der Partikelbildverschiebung erreicht.

Ermittlung der Geschwindigkeit

Mit der Kenntnis der genauen Partikelbildverschiebung $\vec{r}$, der Vergrößerungskonstante M des Bilderfassungssystems und des zeitlichen Unterschiedes Δt der beiden Aufnahmen setzt sich die betragsmäßige Strömungsgeschwindigkeit laut [RWK98] innerhalb einer bestimmten Rasterzelle $\hat{I}_{X,Y}$ wie folgt zusammen:

$$|\vec{U}_{X,Y}| = \frac{|\vec{r}_{X,Y}|}{M \cdot \Delta t} + \frac{\varepsilon_{resid}}{M \cdot \Delta t} \quad \text{mit} \quad \lim_{\Delta t \to 0} \frac{\varepsilon_{resid}}{M \cdot \Delta t} = \infty. \tag{2.4}$$

Der in der Rekonstruktion verbleibende Fehler ε_{resid} zwischen realer und ermittelter Geschwindigkeit ist i. d. R. natürlich nicht bekannt. Aus diesem Ausdruck ist ersichtlich, dass ein kleines Δt die Komponente mit dem Fehler ε_{resid} verstärkt, ein großes Δt aber den Rauschanteil der Aufnahmesignale erhöht, d. h. den Wertebereich der Geschwindigkeiten nach unten skaliert und dessen Unterschied zum Rauschen vermindert.

[1] Analog dazu lässt sich auch eine Hilfsvariable für die vertikalen Nachbarn definieren.

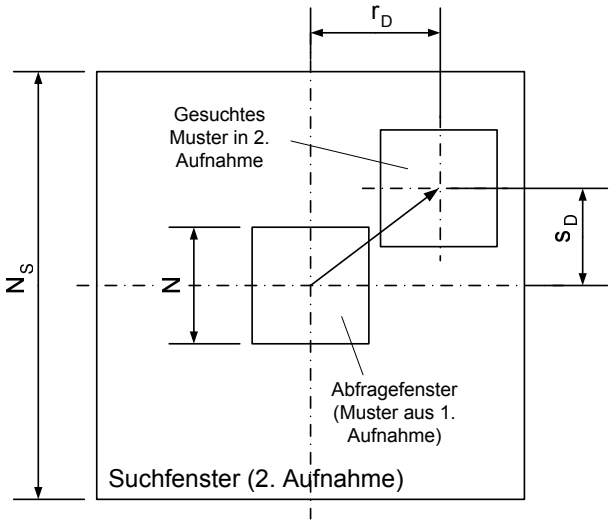

Abbildung 2.1: Das Suchfenster relativ zum Abfragefenster vergrößern

2.1.2 Erweiterung des Suchfensters

Bisher wurde angenommen, dass Abfrage- und Suchfenster die gleichen Abmessungen besitzen. Da ja bei der Kreuzkorrelation das Abfragefenster an jede Stelle des Suchfensters geschoben und dann eine Form der Konvolution berechnet wird, haben wir immer nur eine teilweise Überlappung zwischen den beiden Fenstern und folglich auch nur eine unvollständige Kreuzkorrelation. Somit nehmen also nur die im teils sehr viel kleineren Überlappungsbereich befindlichen Signalwerte an der Berechnung des Korrelationskoeffizienten teil, da alle Werte außerhalb dieses Bereichs nur als „0-Summanden“ in die Gleichung 1.1 einfließen. Dieser Tatbestand führt zu einer oft ungewollten, pyramidenförmigen Gewichtung der Kreuzkorrelationskoeffizienten relativ zum Zentrum der Korrelationsebene (siehe [Wes93]). Ist die Verschiebung der Partikelbildmuster zwischen erster und zweiter Aufnahme größer als $N/4$, wenn N der Durchmesser des Abfragefensters ist, so erhält man zunehmend schlechtere Ergebnisse infolge des sogenannten „in-plane“-Partikelpaarverlustes (siehe [HFW92a] und [KA90]). Das Abfragefenster zu vergrößern ist mit erheblichem Zeitaufwand verbunden. [Wes93] berichtet, dass Größen von mehr 64×64 Pixel zu keiner Verbesserung der Ergebnisse führen und schlägt 32×32 Pixel als Standard vor.

Eine sehr naheliegende, aber kostspielige Methode wurde sehr bald von [KA92] vorgeschlagen und von [HFW92a] genauer beschrieben. Man vergrößert das Suchfenster unter dem Abfragefenster auf das mindestens Doppelte der Ursprungsgröße $N_s \approx [2,x] \cdot N$ und lässt das Abfragefenster nicht mehr über das Suchfenster hinausgleiten (siehe Abbildung 2.1). Dadurch erhalten wir eine von den Größenverhältnissen zwischen Abfrage- und Suchfenster abhängige Korrelationsebene verbunden mit einer entsprechenden Erweiterung der Bandbreite der entdeckbaren Verschiebungen. Jeder Koeffizient $R_{\hat{I}\hat{S}}(m,n)$ wird unter vollständiger Überlappung der beiden Fenster berechnet. Da wir für jedes $R_{\hat{I}\hat{S}}(m,n)$ nun nur mehr den unter dem Abfragefenster liegenden Bereich des Suchfensters an der Stelle (m,n) betrachten, wandeln wir den zweiten Term im Nenner von Gleichung 1.1 in $\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\hat{S}^2(i-m,j-n)$ um. Dieser Ansatz wird in der Software *PIVA* verwendet.

2.1.3 Deformation bzw. Verschiebung von Abfrage- und Suchfenster

Um jedoch nicht zu ineffizienteren Methoden übergehen zu müssen, wurden neue Verfahrensansätze geschaffen und erprobt, angetrieben durch die Ideen von [KA92] und [HFW92a]. Im Prinzip beruhen alle Gedankengänge darauf, durch bereits in vorhergehenden (einfachen) PIV-Methoden errechneten Schätzern für die lokale Partikelbildverschiebung, relativ zueinander liegende, nicht notwendig deckende, Positionen zwischen Abfrage- und Suchfenster festzulegen. Dementsprechend gibt es mittlerweile eine Reihe von Methoden n-ter Ordnung.

Wiederum von [KA92] vorgeschlagen und durch [HFW92a] ausgebaut wurde eine Methode 0. Ordnung, welche weitläufig unter den Namen „Discrete Window Offset Method“ und „Window Displacement Method“ (Fensterverschiebemethode) verwendet wird. Wie in Abbildung 2.2 ersichtlich wird die Position des Partikelmusters in $\hat{I}_{X,Y}$ mit dem Ursprung (X,Y) im Gesamtbild in der zweiten Aufnahme im Bereich $\hat{S}_{X+\lfloor r_1 \rfloor, Y+\lfloor s_1 \rfloor}$ an der Stelle $(X+\lfloor r_1 \rfloor, Y+\lfloor s_1 \rfloor)$ vermutet, wobei die Verschiebung (r_1, s_1) an dieser Stelle der Aufnahme in einer vorhergehenden Rekonstruktionsiteration ermittelt worden ist. Von dort aus kann eine zur ersten Verschiebung bzw. zu diesem ersten Schätzervektor relative Zweite (s_2, r_2) ermittelt werden. Diese Verschiebung ist aufgrund der Optimierung der Korrelationsspitze genauer und wird nun vektoriell zur Ersten addiert. Diese Methode erfordert folglich keine derart großen Suchfenster mehr wie in Abschnitt 2.1.2. Wir sehen nun daraus, dass ein solches Vorgehen iterativ, also in mehreren Schritten, gelöst werden muss, bis ein annehmbares Ergebnis zu erwarten ist. Näheres zu diesen iterativen Verfahren wird im folgenden Abschnitt 2.1.6 angeführt. Lediglich der Ansatz 0. Ordnung ist in *PIVA* integriert und erzielt wie erwartet gute Ergebnisse jedoch klarerweise unter der Bedingung, dass $(r_1, s_1) \gg (1,1)$ ist. Genauer analysiert wurde diese Methode von [WDG97].

Schon [HFW92b] hat eine genaue, aber zeitintensive (weil $O(N^2)$) Technik namens „Particle Image Distortion“ (PID) entwickelt. Diese berücksichtigt

- neben der **linearen Verschiebung** des Suchfensters noch
- die **Verzerrung** n-ter Ordnung[2] des Partikelmusters unter dem Abfragefenster — entstehend durch verschiedene Geschwindigkeiten in unterschiedlichen Strömungsschichten — und
- dessen **Rotation** — auch entstehend durch Wirbelstrukturen.

Der Teil der ersten Aufnahme unter dem Abfragefenster wird also entsprechend den örtlichen Werten der Strömungskrümmung verzerrt. Dann wird dieses verzerrte Muster mit dem bereits verschobenen Suchfenster in der zweiten Aufnahme korreliert. Eine sehr gute Zusammenfassung u. a. dieser Methoden findet man auch in dem aktuelleren Artikel [Sca02]. Dieser weist allerdings darauf hin, dass eine Fensterdeformation höherer Ordnung zwar die Korrelationswerte verbessert und damit die Anzahl der Ausreißer vermindert, jedoch nicht unbedingt die Sub-Pixel-Genauigkeit erhöht. [JJDAF95] schlägt eine Verbesserung der PID-Methode vor, um diese Schwachstellen auszumerzen und präsentiert gute Ergebnisse. [WG03] beschreibt ein Verfahren 1. Ordnung ähnlich der PID unter Einbeziehung der Verdrehung eines Partikelmusters anschaulich und visualisiert entscheidend bessere Testergebnisse im Gegensatz zur Methode 0. Ordnung. *PIVA* ist dafür konzipiert, um mit einem Ansatz höherer Ordnung erweitert werden zu können.

[2] Der entsprechende Artikel beschreibt lediglich eine Methode 1. Ordnung, siehe Abbildung 2.3.

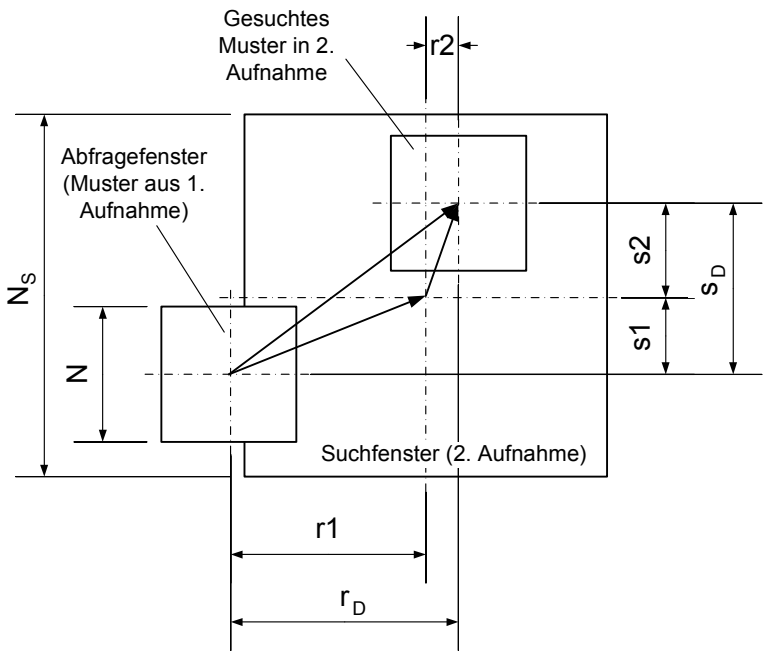

Abbildung 2.2: Eine Fensterverschiebemethode 0. Ordnung. Die zweite Verschiebung (r_2, s_2) (ermittelt in der 2. Iteration) wird nachher zur Ersten (r_1, s_1) (ermittelt in der 1. Iteration) addiert.

An dieser Stelle sei noch auf die Dissertation [Bro99] verwiesen, in der zusätzlich zur Fensterverschiebetechnik eine die lokale Wirbelstärke berechnende Finite-Differenzen-Methode angewendet wird. Mit dieser Information kann ähnlich der PID-Technik eine Verdrehung des Partikelmusters kompensiert werden. Um höher aufgelöste, jedoch trotzdem genaue Vektorfelder zu rekonstruieren, wird der in einer früheren Iteration $i-1$ gewonnene Schätzer oder Verdrehungswinkel für die lokale Wirbelstärke dazu verwendet, um in der Iteration[3] i mittels Variation dieses Winkels eine möglichst vollständige Übereinstimmung der Partikelmuster zu erhalten. Eine Überlappung der Abfragefenster wird dabei ausgelassen.

Ein ähnlicher Ansatz zur Bestimmung der Verdrehung zwischen zwei Mustern bzw. der Wirbelstärke stammt von [RSY01] und wird als „Direct Measurement of Vorticity“ (DMV) bezeichnet. Die Wirbelstärke wird also nicht aus den Vektordaten sondern aus den Bilddaten direkt gewonnen. Im ersten Schritt wird eine herkömmliche PIV-Methode mit Fensterverschiebetechnik für die Zuordnung der zusammengehörenden Muster in beiden Bildern angewendet. Im zweiten Schritt betrachtet man Kreisausschnitte dieser Partikelmusterpaare, indem sie vom kartesischen Koordinatensystem in ein Polares transformiert werden. Per D-(N)KK oder FFT-KK wird dann der Versatz im Polaren berechnet welcher zurücktransformiert einen Verdrehungswinkel zwischen den Mustern und die entsprechende Winkelgeschwindigkeit ergibt. Eine Umrechnung von Kartesisch auf Polar ist allerdings $O(N^2)$-aufwändig.

[3] i repräsentiert hier die nächste Verfeinerungsstufe mit der doppelten Vektorfeldauflösung.

2.1.4 Die kontinuierliche Fensterverschiebung (CWS)

Angestoßen durch die in der PIV-Analyse auftretenden, verschiedenen Fehler erklärt u. a. [GW02] die Technik der kontinuierlichen (oder auch „Sub-Pixel"-) Suchfensterverschiebung im Gegensatz zum bisher verwendeten, ganzzahligen Anteil von $\vec{r}$ zur Suchfensterverschiebung. D. h. das Partikelmuster aus der ersten Aufnahme wird nun an einer reellen Position der zweiten Aufnahme vermutet. Auf eine der wichtigsten Arten solcher Fehler wird im nachfolgenden Abschnitt 2.1.5 eingegangen.

Da man natürlich Informationen nur an diskreten Stellen (Grauwertverteilung durch diskrete Pixel) kennt, ermittelt man — entsprechend der reellen Verschiebung aus vorangegangenen Iterationen — z. B. mit bilinearer oder quadratischer Spline-Interpolation der Grauwerte ein temporäres Partikelmuster $\hat{S}'(x,y)$ aus der zweiten Aufnahme S auf Grundlage der reellen Partikelbildverschiebung $\vec{r}$. Aus der Definitionsgleichung 2.1 kennen wir die ganzzahligen und reellen Teile von $\vec{r}$.

Für $\hat{S}'$ als Ersatz für $\hat{S}$ in der D-(N)KK ergibt sich damit durch

$$\begin{aligned}\hat{S}'(x,y) \quad = \quad & (1-|\varepsilon_m|)\cdot(1-|\varepsilon_n|)\cdot\hat{S}(x+m_0,y+n_0) \\ & +|\varepsilon_m|\cdot(1-|\varepsilon_n|)\cdot\hat{S}(x+m_0+d_m,y+n_0) \\ & +|\varepsilon_n|\cdot(1-|\varepsilon_m|)\cdot\hat{S}(x+m_0,y+n_0+d_n) \\ & +|\varepsilon_m|\cdot\varepsilon_n\cdot\hat{S}(x+m_0+d_m,y+n_0+d_n).\end{aligned} \tag{2.5}$$

ein bilineares Interpolationsschema, wobei

$$\vec{d} \equiv \begin{pmatrix} d_m \\ d_n \end{pmatrix} = \begin{pmatrix} \mathrm{sign}(\varepsilon_m) \\ \mathrm{sign}(\varepsilon_n) \end{pmatrix}.$$

[Sca02] erklärt diese Interpolationsmethode ebenfalls. Wie man allerdings leicht sehen kann, handelt es sich hierbei um einen zusätzlichen $O(N^2)$-Algorithmus, der allerdings laut den Ergebnissen einiger Artikel eine signifikante Verbesserung der Genauigkeit nach sich zieht. Bilineare Interpolation sollte aber für eine noch bessere Annäherung durch Interpolationsschemata höherer Ordnung ersetzt weren. *PIVA* implementiert die oben erklärte Technik bereits. Ergebnisse können in Abschnitt 8 eingesehen werden.

2.1.5 Der „peak-locking"-Effekt als wichtigster Fehler der PIV

Bei der PIV-Analyse treten erwartungsgemäß Fehler verschiedener Herkunft auf. Eine wichtige Art von Fehlern, welche mit herkömmlichen Methoden kaum zu vermeiden sind, nennt man den sogenannten „peak-locking"-Effekt. Er entsteht durch die Digitalisierung (siehe [Wes93]), hat ein periodisches Muster der Wellenlänge von einem Pixel bzw. einer Abtastung und verzerrt die ermittelten Verschiebungen in Richtung des nächsten ganzzahligen Wertes. D. h. dieser Fehler lässt sich indirekt in einem Histogramm der Partikelbildverschiebungen als periodische Kurve ähnlich einer harmonischen Schwingung visualisieren, wobei die lokalen Maxima dieser Kurve bei den ganzzahligen Werten liegen, siehe z. B. Bilder in Kapitel 4 von [GW02] bzw. Histogramme im späteren Abschnitt 8.

[SR00] berichten, dass man diesen Fehler mit der kontinuierlichen Fensterverschiebung aus Abschnitt 2.1.4 nahezu vollständig kompensieren kann. [GW02] untersucht diesen Effekt

ebenfalls sehr genau und kommt zum gleichen Ergebnis. Ferner wird der „peak-locking"-Effekt laut den Untersuchungen in [RWK98] bei einem durchschnittlichen Partikelbilddurchmesser von ≈ 2 Pixel minimal. Der Grund hierfür wurde nicht angegeben. Vor allem im Bereich der μ-PIV (siehe Tests in Abschnitt 8.3) erwartet man sich damit genauere Ergebnisse, da die Partikelbildverschiebungen dort oft $\ll 1$ Pixel sind.

PIVA implementiert momentan wahlweise die oben propagierte Sub-Pixel-Fensterverschiebemethode, um dem „peak-locking"-Effekt entgegen zu wirken.

2.1.6 Erhöhung der Vektorfeldauflösung und iterative Verfahren

Bis jetzt haben wir uns noch keine Gedanken über die Größe N des Abfragefensters gemacht. Denn von dieser hängt die Auflösung des in Kapitel 1.3 eingeführten Rasters und damit auch die Auflösung des rekonstruierten Vektorfeldes ab. Je nach Anwendung kann dieser Parameter unterschiedlich festgelegt werden. Es stellt sich aus vorangegangenen Betrachtungen heraus, dass je nach lokaler Geschwindigkeitsverteilung eine untere Schranke für diese Größe besteht, um vernünftige Ergebnisse zu erzielen. Das ist aber ungenügend, denn hat man kleine Strukturen wie z. B. Wirbel in einer Strömung, möchte man diese auch im Vektorfeld wiedererkennen. Schon von [Wes93] wurde eine Überlappung der Abfragefenster von bis zu 90% bei gleichbleibender Größe N vorgeschlagen. Aufgrund der somit gut korrelierten Ergebnisse benachbarter Abfragefenster führt dies zu einem gewissermaßen geglätteten Vektorfeld. Nachteile der Überlappung bestehen laut [RWK98] darin, dass benachbarte Verschiebungen aus teils denselben Partikelmustern ermittelt werden. Die schon erwähnte Korrelation benachbarter Verschiebungsvektoren führe dann zu verzerrten Ergebnissen nach der Weiterverarbeitung (z. B. Differentiation) der Vektorfelder.

Des weiteren gibt es effizientere hierarchische Ansätze. Wir haben bereits gesehen, dass man mit der Fensterverschiebetechnik genauere Werte für PIV-Ergebnisse erreichen kann. Da diese Methode ohnehin eine iterative Vorgangsweise voraussetzt, kann man die in einer Iteration $i-1$ ermittelten Daten nutzen, um diese in der Iteration i direkt oder via bilinearer Interpolation zwischen den Nachbarvektoren (siehe [SR99]) als Schätzer in einem feineren Abfrageraster mit Abfragefenstern der Größe $N/2$ o.B.d.A. zu verwenden (siehe Abbildung 2.4). Hat man erst einen guten Schätzervektor[4], so kann man die Suchfenstergröße ebenso drastisch verkleinern. Man sieht sofort, dass die direkte Kreuzkorrelation mit nunmehr $(N/2)^4 = N^4/16$ Schritten schneller wird bei jetzt doppelter Auflösung des Vektorfeldes. Details beschreibt [SR99] mit der dort eingeführten WIDIM-Technik und berichtet weiter, dass es i. d. R. weniger als fünf Iterationen bedarf, um mit der Fensterverschiebetechnik zu einer Konvergenz der Partikelverschiebung zu kommen. Das Konvergenzintervall oder -kriterium wird mit einer geeigneten oberen Fehlerschranke festgelegt. Dabei kann man auch mehrere Iterationen innerhalb einer Auflösungsebene machen, um genauere Schätzer für feinere Auflösungsstufen zu erhalten. [SR00] hat, wie in Abbildung 2.3 ersichtlich, weitergehend den Ansatz 1. Ordnung von [HFW92b] (siehe Abschnitt 2.1.3) aufgegriffen und auf seine WIDIM-Technik angewendet. Weitere Informationen zur Verbesserung der Auflösung in Vektorfeldern gibt es in Kapitel 5.

PIVA verwendet diesen Ansatz, um sehr hohe Auflösungen des Vektorfeldes bei gering bleibenden Anteilen von Ausreißern[5] zu erreichen, vererbt Schätzervektoren aus höheren

[4] Solche werden erreicht durch nachbearbeitende Methoden, wie z. B. die Suche und Ersetzung von fehlerhaften Vektoren bzw. Ausreißern. Details dazu werden später in Abschnitt 2.1.8 beschrieben.

[5] Hierbei handelt es sich um fehlerhaft rekonstruierte Vektoren, genauer erklärt im späteren Abschnitt 2.1.8.

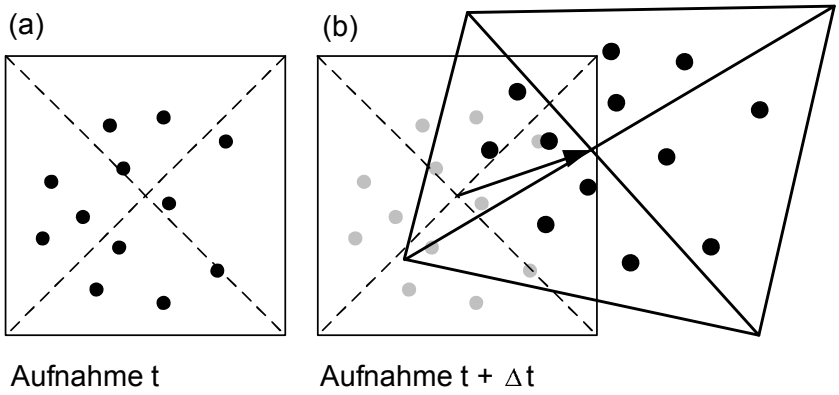

Abbildung 2.3: Die Fensterdeformation 1. Ordnung, siehe [SR00]. In (a) befindet sich das Ausgangspartikelmuster. (b) zeigt ein durch Translation, biaxiale Scherung und Rotation deformiertes Partikelmuster.

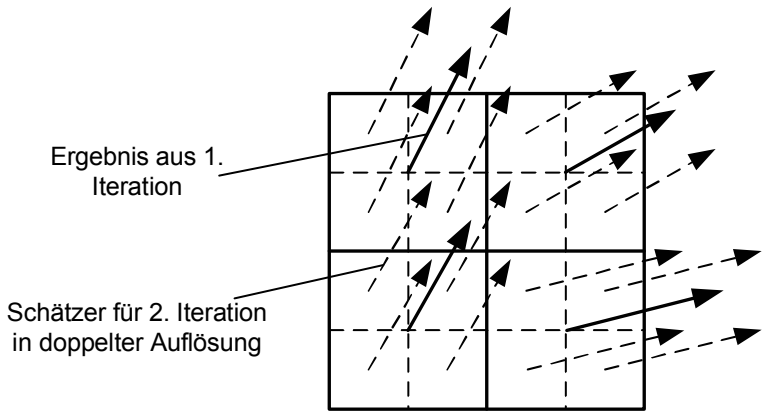

Abbildung 2.4: Ein Beispiel aus der WIDIM-Methode von [SR99]. Schätzer in der ersten Iteration werden 1:1 in die nächste Verfeinerungsebene vererbt, um in einer zweiten Iteration mittels Fensterverschiebetechnik genauere Ergebnisse zu erzielen.

Ebenen bisher aber nur direkt, d. h. 1 : 1. Die oben erwähnte bilineare Interpolation von neuen Schätzervektoren wäre allerdings besser.

2.1.7 Eine spezielle D-NKK zur Vorbereitung von PIV-Bildern

[RWK98] und [Lew95] verwenden eine spezielle Form der normalisierten Kreuzkorrelation (D-NKK), in der sie $\hat{I} \equiv \hat{I}_{X,Y}$ bzw. $\hat{S} \equiv \hat{S}_{X,Y}$ um die durchschnittliche Helligkeit $\langle \hat{I} \rangle$ bzw. $\langle \hat{S} \rangle$ der Aufnahmen in den gerade betrachteten Bereichen vermindern, sodass die Aufnahme zu einem um Null gemittelten Zufallssignal wird. Die Gleichung 1.1 für die Kreuzkorrelationskoeffizienten in der Rasterzelle (X,Y) gestaltet sich damit um zu

$$R_{\hat{I}\hat{S}}(m,n) = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[\hat{I}(i,j) - \langle \hat{I} \rangle][\hat{S}(i-m,j-n) - \langle \hat{S} \rangle(m,n)]}{\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[\hat{I}(i,j) - \langle \hat{I} \rangle]^2 \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[\hat{S}(i-m,j-n) - \langle \hat{S} \rangle(m,n)]^2}}, \tag{2.6}$$

wobei $\langle \hat{S} \rangle(m,n)$ das arithmetische Mittel des betrachteten $N \times N$-Teilbereichs des Suchfensters an der Stelle (m,n) relativ zum Suchfenster ist.

[GME+02] berichtet von schärferen Korrelationsspitzen und damit genaueren Ergebnissen. [Wes93] schlägt eine ähnliche Methode namens Kontrastnormalisierung vor, um damit Helligkeitsunterschiede innerhalb und zwischen verschiedenen Aufnahmen zu kompensieren und so die Korrelation zu verbessern.

PIVA kann per Kommandozeilenschalter wahlweise mit oder ohne dieser Form der Kreuzkorrelation arbeiten. Anstatt die kompliziertere Gleichung 2.6 direkt zu implementieren, könnte man das gesamte Bild auch in einem Schritt vor der Korrelationsberechnung normalisieren. Die Regionalität der **Bildnormalisierung** geht dabei allerdings zu einem gewissen Grad verloren. Die Datenstrukturen für das so erhaltene Null-gemittelte Zufallssignal müssen somit auch **negative Werte** unterstützen. Bekannterweise kann hier nicht ausschließlich der „Gaussian-fit"-Schätzer verwendet werden, da negative Werte für $R_{\hat{I}\hat{S}}(m,n)$ in Gleichung 2.3 nicht zugelassen sind. An solchen Stellen können die betrachteten Korrelationswerte um

$$|\min_{i,j}\{R_{\hat{I}\hat{S}}(m+i,n+j)\}| \quad \text{mit} \quad (i,j) \in \{(0,-1),(-1,0),(0,0),(1,0),(0,1)\}$$

ins Positive verschoben werden, was das Ergebnis dieses Schätzers verzerren kann. Sicherer und für solche Fälle auch in *PIVA* realisiert, ist die Zuhilfenahme des „center-of-mass"-Schätzers aus Gleichung 2.2.

2.1.8 Erkennung von Ausreißern oder „outliers"

Nach einer PIV-Analyse kann es abhängig von Prozessparametern wie z. B. der Abfragefenstergröße, der lokalen Geschwindigkeitsverteilung, etc. zu Messfehlern kommen, den sogenannten Ausreißern oder „outliers". Z. B. WESTERWEEL (siehe [Wes94]) beschreibt drei statistische Methoden, um Ausreißer zu erkennen. Seine umfangreiche, von vielen Parametern abhängige, statistische Analyse schließt mit einer oberen Grenze r_c^2 zur Entscheidung zwischen „gültigem Vektor" und „Ausreißer", welches sich für unsere versuchsweise Anwendung zu

$$r_c^2 = C \cdot \mathrm{var}\{\vec{V}_{i,j}\} \tag{2.7}$$

mit einer empirisch wählbaren Konstante $C \approx 2...3$ und der Varianz der Vektoren $\vec{V}_{i,j}$ mit $(i,j) \in M$ vereinfachen lässt. Im Folgenden wollen wir mit r den skalaren Abstand eines Vektors $\vec{V}_{i,j}$ zu einem entsprechenden Referenzvektor $\vec{V}_{ref}$ bezeichnen.

Die Umgebung M gestaltet sich je nach Wahl einer der folgenden Methoden:

- Der „Global-mean"-Test nimmt das gesamte Vektorfeld über der Aufnahme in die Menge M auf und berechnet darüber das arithmetische Mittel $\vec{V}_{ref} \equiv \langle \vec{V}_{i,j} \rangle$, erzeugt daher große Varianzen und führt zu einem schlechten Kriterium für die Erkennung von Ausreißern.

- Der „Local-mean"-Test funktioniert gleich wie der „global-mean"-Test, beobachtet aber nur eine beschränkte (z. B. 3×3-) Nachbarschaft als M und erzielt nur unter der Bedingung, dass benachbarte Vektoren keine Ausreißer sind, gute Ergebnisse. Eine Kombination zwischen erster und dieser Methode wäre zwar besser aber ineffizient.

- Der „Local-median"-Test liefert diesbzgl. die besten Ergebnisse und wird deshalb auch von *PIVA* implementiert. Er betrachtet eine 3×3- oder 5×5-Nachbarschaft als Umgebung M. Hier wird aber der komponentenweise Median der Vektoren aus M inklusive des betrachteten Vektors $\vec{V}_{i,j}$ berechnet.

$$[\vec{V}]_{i,j} \equiv \vec{V}_{ref} = \begin{pmatrix} \text{median}\{\vec{V}^{(x)}_{i+k,j+l}\} \\ \text{median}\{\vec{V}^{(y)}_{i+k,j+l}\} \end{pmatrix} \quad \text{mit} \quad k,l \in \{-1,0,+1\} \tag{2.8}$$

  und $\vec{V}^{(x)}$ bzw. $\vec{V}^{(y)}$ sind die beiden Komponenten x bzw. y von $\vec{V} = (x,y)^T$.

Ist also der quadrierte, skalare Unterschied des betrachteten Vektors $\vec{V}_{i,j}$ zum Median mit

$$r^2 = \|[\vec{V}]_{i,j} - \vec{V}_{i,j}\|^2 \tag{2.9}$$

größer als r_c^2, so wird dieser als Ausreißer verworfen. Als Ersatz kann z. B. wie in *PIVA* ein neuer Vektor durch bilineare Interpolation genommen werden oder auch das arithmetische Mittel aller Nachbarvektoren.

[RWK98] schlägt weitere Methoden namens „Global histogram operator" und „Dynamic mean value operator" vor, auf die hier aber nur verwiesen wird. Des weiteren existiert ein funktionsreiches Softwarepaket namens *CleanVec*[6] zum Erkennen und Ersetzen von Ausreißern.

Prinzipiell gilt es jedoch, sich genau zu überlegen, wie oft man eine Validierung des Vektorfeldes durchführt (z. B. nach jeder Verfeinerung oder sogar nach jeder Iteration). Das oben vereinfachte Kriterium r_c^2 wird hier zu Versuchszwecken verwendet. Es empfiehlt sich, dieses mit Hilfe der Überlegungen von [Wes94] auf seine eigenen Bedürfnisse anzupassen. Bei turbulenten Strömungen mit großer Wirbelstärke kann es je nach Wahl von r_c^2 zu häufigen Fehlern 2. Art kommen, und es werden gültige Vektoren als falsch deklariert. Man erhält dadurch eine übermäßig starke Glättung des Vektorfeldes. *PIVA* führt für den Zweck dieser Diplomarbeit eine Erkennung und Ersetzung in der ersten Iteration durch. Hier wäre zu überlegen, ob man eine Validierung in Abhängigkeit der lokalen Strömungskrümmung und Vektorfeldauflösung durchführt oder auslässt. Weitere Ideen zur Validierung von Vektorfeldern und Ersetzung von Ausreißern werden in [NLR97] verfolgt.

[6] Informationen dazu gibt es unter http://ltcf.tam.uiuc.edu/Downloads/Software/CleanVec/cleanVec.html.

2.2 Optimierung auf Geschwindigkeit

Eine Steigerung der Effizienz der Methoden ist auch bei den heute sehr leistungsstarken Rechenanlagen unerlässlich. Deshalb möchte ich nun auf einige Verbesserungsmöglichkeiten eingehen. Im späteren Abschnitt 5.1.1 werden im Zuge der Erklärung adaptiver Verfahrensvarianten zusätzliche Optimierungsmöglichkeiten bzgl. der Geschwindigkeit besprochen.

2.2.1 Kreuzkorrelation via FFT — Vor- und Nachteile

Die schnelle Kreuzkorrelation via FFT (FFT-KK) wurde bereits in Abschnitt 1.3.2 eingeführt. Ich möchte hier noch einmal auf die Vor- und Nachteile eingehen. [Pus00] hat sich mit dem Vergleich zwischen direkter Kreuzkorrelation (D-NKK) und FFT-KK eingehend befasst und gelangt zu zwei naheliegenden, aber auch fundamentalen Aussagen:

1. Für qualitativ gute Ergebnisse ist stets die D-NKK vorzuziehen.

2. Die Algorithmen sind unbedingt auf realen Daten zu Testen, da es hier tiefgreifende Unterschiede zu künstlichen Aufnahmen gibt.

Neben der **hohen Effizienz** als wichtigstem **Vorteil** der FFT-basierten Kreuzkorrelation herrschen folgende

Nachteile:

– Es besteht die Bindung der Abfragefenstergröße an eine 2er-Potenz. Somit muss man bei arbiträren Größen freien Platz mit Nullen oder sich wiederholenden Daten auffüllen. Allerdings haben [GM98] eine Methode entwickelt, um die FFT-KK auch mit arbiträren Abfragefenstergrößen sinnvoll zu verwenden. Eine nähere Betrachtung sei dem Leser an dieser Stelle überlassen.

– Die FOURIER-Transformation (FT) stützt sich einerseits auf einen unendlich großen Signalbereich. Das Abfragefenster ist andererseits nach allen Seiten begrenzt. In einer „beschränkten“ FT kann man also einen als N-periodisch auftretend angenommenen Teil der Länge N des Signales betrachten. Folglich erzwingt man die Annahme, dass die PIV-Bilddaten N-periodisch seien, was i. A. aber falsch ist und deshalb zu verzerrten, ungenauen Ergebnissen führt (für Details siehe [RWK98]).

– Die FFT-KK hat im Gegensatz zur D-NKK keinen Normalisierungsterm im Nenner. Durch die Normalisierung sind die Ergebnisse weniger empfindlich gegenüber Helligkeitsschwankungen [GME+02]. Ein zusätzlicher Normalisierungsterm in der FFT-KK ist laut [RWK98] schwieriger realisierbar als im Ortsbereich und führt zu niedriger Effizienz.

– Die auf Seite 19 angesprochene pyramidenförmige Gewichtung des Signales in der Korrelationsebene bleibt durch die FFT implizit erhalten, da ja hier kein vergrößertes Suchfenster verwendet wird.

2.2.2 Eine schnelle Variante der D-NKK

Wie bereits erwähnt, nutzt die FFT-KK keine Normalisierung, was die Ergebnisse zusätzlich verschlechtern kann. Will man die Eigenschaften einer Normalisierung nutzen, gelangt man wieder zur D-NKK, da sich die FFT-KK mit einer solchen Eigenschaft laut [Lew95] nicht ebenso effizient realisieren lässt.

Aus Gleichung 2.6 kennen wir den zweiten Hauptterm $\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[\hat{S}(i-m,j-n)-\langle\hat{S}\rangle(m,n)]^2$ im Nenner. Dieser erweitert sich ja zu

$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\hat{S}^2(i-m,j-n) \quad - \quad 2\langle\hat{S}\rangle(m,n)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\hat{S}(i-m,j-n) \quad + \quad N^2\langle\hat{S}\rangle^2(m,n), \tag{2.10}$$

hängt von (m,n) ab und müsste im Gegensatz zum ersten Hauptterm für jedes (m,n) in $O(N^2)$ berechnet werden. Hier sei erwähnt, dass $\langle\hat{S}\rangle(m,n) = 1/N^2 \cdot \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\hat{S}(i-m,j-n)$ gilt. Abhilfe schafft nun der Ansatz von [Lew95], welcher vorschlägt, für die Terme[7] $\sum\sum\hat{S}$ und $\sum\sum\hat{S}^2$ einmalig diskrete Intergrale auf $N_s \times N_s$ durch

$$\begin{aligned} s(u,v) &= \hat{S}(u,v)+s(u-1,v)+s(u,v-1)-s(u-1,v-1) \quad \text{und} \\ s^2(u,v) &= \hat{S}^2(u,v)+s^2(u-1,v)+s^2(u,v-1)-s^2(u-1,v-1) \end{aligned}$$

zu berechnen und diese als zweidimensionale Integralfunktion auszuwerten. Dabei gilt $s(u,v) = s^2(u,v) = 0$, wenn $u,v < 0$. Bei diesen Integralen handelt es sich um diskrete, fortlaufende, zweidimensionale Summen (siehe Abbildung 2.5).

Die Summe der Helligkeitsquadrate $\hat{S}$ im Suchfenster unter dem auf (m,n) positionierten $N \times N$-Abfragefenster lässt sich somit in $O(1)$ durch

$$\begin{aligned} e_{\hat{S}^2}(m,n) &= s^2(m+N-1,n+N-1) \\ &- s^2(m-1,n+N-1) \\ &- s^2(m+N-1,n-1) \\ &+ s^2(m-1,n-1) \end{aligned} \tag{2.11}$$

berechnen und analog die einfachen Helligkeiten $\hat{S}$.

2.2.3 Kombination von FFT-KK und D-NKK — ein hybrider Ansatz

[Idee] Da wir ja mit der eingeführten (kontinuierlichen) Fensterverschiebetechnik ohnehin bereits einen iterativen Algorithmus haben, bietet es sich an, grobe Schätzer bei größeren Abfragefenstern mittels FFT-KK zu berechnen und in feineren Vektorfeldauflösungen mit kleineren Abfrage- und Suchfenstern auf die D-NKK umzusteigen. Zusätzlich können wir die schnelle Version der D-NKK aus Abschnitt 2.2.2 verwenden. Damit haben wir eine sogenannte **hybride Variante**.

PIVA ist in dieser Hinsicht erweiterbar, nutzt momentan aber nur die D-NKK für Berechnungen. Eine effiziente Implementierung[8] kann leicht integriert werden.

[7] Parameter wurden hier zur Abkürzung ausgelassen.
[8] Die freie Bibliothek „Fastest Fourier Transform in the West“ gibt es unter `http://www.fftw.org`.

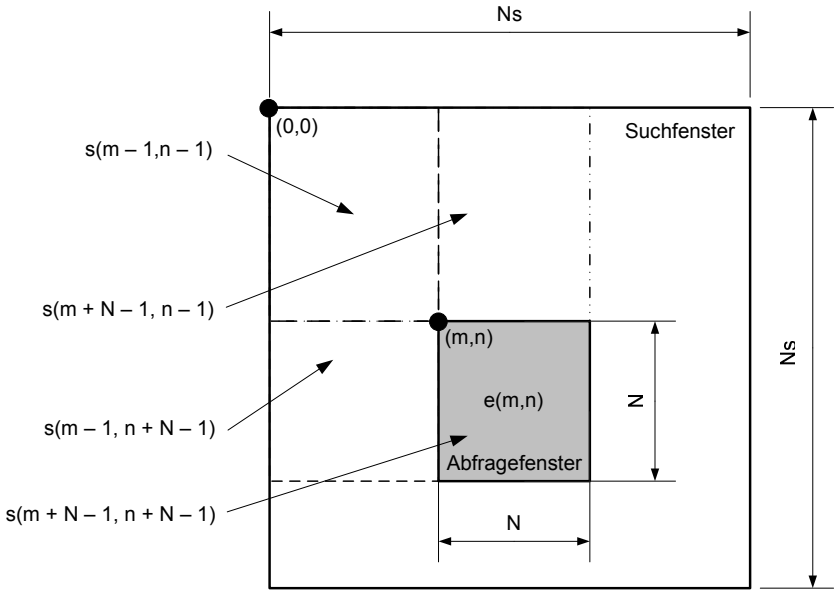

Abbildung 2.5: Eine diskrete, zweidimensionale Integralfunktion (in [Lew95] auch als Laufsumme bezeichnet), hier am Beispiel der einfachen Helligkeit $\hat{S}$. Alle gezeigten Rechtecke für s beginnen in der linken, oberen Ecke $(0,0)$ und überlappen sich entsprechend, was direkt auf die Gleichung 2.11 schließen lässt.

2.3 Alternative Ansätze und Anwendungen

Abgesehen von den bisher behandelten Methoden gibt es noch eine lange Reihe weiterer Ansätze, von denen ich hier nur einige kurz anschneiden werde, um den Anforderungen einer Übersicht von PIV-Methoden einigermaßen gerecht zu werden. Diese behandeln hauptsächlich die Verbesserung der Genauigkeit der PIV-Analyse und beschäftigen sich teilweise auch mit der gleichzeitigen Erhöhung der Effizienz. Für detaillierte Informationen sei aber jeweils auf die angegebene Literatur verwiesen. [Sca02] beinhaltet eine aktuelle, gut erklärte Zusammenfassung moderner Techniken im Bereich der DPIV. Für den Zweck dieser Diplomarbeit war es nicht nötig, die hier angedachten Ideen durch *PIVA* in Software abzubilden. Ferner hätte der dafür nötige Aufwand einen vertretbaren Rahmen für ein „1-Mann-Projekt“ gesprengt.

2.3.1 Ideen aus der PIV-Challenge '03

In dem im Jahre 2003 ausgetragenen „Wettkampf“ zur Evaluierung der neuesten PIV-Methoden — interessant und anschaulich zusammengefasst in [SOK03] — haben einige Forschungsgruppen die verschiedenen Methoden in vordefinierten, vergleichenden Tests ausgewertet. [SOK03] berichtet u. a. von der Forschungsgruppe *LEGI* der Universität von Grenoble (Frankreich). *LEGI* erzielt mit D-KK und Sub-Pixel-Fensterverschiebung sehr gute Ergebnisse unter der Akzeptanz eines langsameren Verfahrens. Es wird eine Partikelmusterdeformation im kontinuierlichen Bereich verwendet. Anstatt der herkömmlichen D-KK wird in der letzten Iteration des hierarchisch verfeinernden Verfahrens die Methode der kleinsten Quadrate zwischen kontinuierlich deformiertem Abfragefenster und diskretem Suchfenster

angewendet, um den „peak-locking“-Effekt zu minimieren.

Andere Teilnehmer wie z. B. der Hersteller eines kommerziellen Produktes *LAVISION* erreichen mittels FFT-KK annehmbare Ergebnisse. Es wurde ferner auch berichtet, dass manche Forschungsgruppen zu einer Überbestimmung der Verfahrensgenauigkeit gelangten, was ja bei der Wirkung der PIV als Tiefpassfilter eher fragwürdig erscheint.

Schlussfolgernd wurde angegeben, dass die heutigen PIV-Algorithmen im Gesamten auch in problematischeren Fällen wie z. B. turbulenten Strömungen annehmbar gut seien. Im Unterschied dazu hat sich bei den PTV-Methoden herausgestellt, dass hier sehr wohl entscheidende Qualitätsunterschiede zwischen den Algorithmen herrschen (Näheres hierzu in [SOK03]). Diese Aussage wird allerdings noch nicht auf die Verfahrenstechnik der mikrosopischen PIV-Analyse (μ-PIV) ausgeweitet.

2.3.2 Ermittlung der Korrelationsspitze mit Hilfe einer Gradientenmethode

[SNOO00] hat sich einen gesamtheitlichen, komplizierten Ansatz überlegt, um den reellen Teil der Partikelmusterverschiebung genauer zu bestimmen. Die bisher besprochene iterative Methode per Kreuzkorrelation wird lediglich bis zur letzten Iteration angewendet. Sobald die in fortschreitenden Iterationen zusätzlich gewonnenen Verschiebungsanteile kleiner als ± 1 Pixel sind, wird die bisher propagierte „Gaussian-fit“-Methode durch eine Gradientenmethode mit einer Kombination aus einer räumlichen und einer zeitlichen Ableitung der Helligkeitsverteilungsfunktion[9] ersetzt. Als Annahme gilt allerdings, dass sich alle Pixel innerhalb des Abfragefensters in etwa homogen in dieselbe Richtung bewegen (siehe (u,v) ab Gleichung 2.12). Der Ansatz wendet die TAYLOR-Reihenentwicklung und die LAGRANGE-Ableitung — hier also die ersten partiellen Ableitungen der Helligkeit in zwei Raumrichtungen $\hat{I}_x$ bzw. $\hat{I}_y$ und unter Kenntnis von Δt die erste partielle Ableitung nach der Zeit $\hat{I}_t$ — auf eine die Partikelbildverschiebung beschreibende Formel

$$\tilde{I}(x+\Delta x, y+\Delta y, t+\Delta t) - \tilde{I}(x,y,t) = \Delta\tilde{I}(x,y,t,\Delta x,\Delta y,\Delta t)$$

an und erhält somit

$$\tilde{I}_t + u\tilde{I}_x + v\tilde{I}_y = e. \tag{2.12}$$

Details zu dieser Berechnung würden hier zu weit führen und sind in [SNOO00] nachzulesen. Hierbei ist $\tilde{I}$ die räumlich-zeitliche Helligkeitsverteilungsfunktion über alle Aufnahmen und natürlich nur zu diskreten Raum- und Zeitschritten verfügbar. e wird als Helligkeitsveränderung ebenfalls abhängig von Raum und Zeit in Form weißen Rauschens angenommen.

Gleichung 2.12 sollte nun an jeder Stelle des Abfragefensters ausgewertet werden. Da wir $\vec{v} = (u,v)^T$ und e nicht kennen, wenden wir das lineare Quadratmittelproblem an und erhalten ein Minimierungsproblem der Form

$$\nabla \sum_{\hat{\mathcal{B}}} (\tilde{I}_t + u\tilde{I}_x + v\tilde{I}_y)^2 \stackrel{!}{=} \vec{0} \tag{2.13}$$

mit $\hat{\mathcal{B}}$ gleich der Menge aller Pixel im Abfragefenster. e kürzt sich hier durch die Differentiation. Wie in [SNOO00] erhalten wir mittels entsprechender Auflösung und Umformung

$$\begin{pmatrix} \sum_{\hat{\mathcal{B}}} \tilde{I}_x^2 & \sum_{\hat{\mathcal{B}}} \tilde{I}_x\tilde{I}_y \\ \sum_{\hat{\mathcal{B}}} \tilde{I}_x\tilde{I}_y & \sum_{\hat{\mathcal{B}}} \tilde{I}_y^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum_{\hat{\mathcal{B}}} \tilde{I}_x\tilde{I}_t \\ \sum_{\hat{\mathcal{B}}} \tilde{I}_y\tilde{I}_t \end{pmatrix}. \tag{2.14}$$

[9] Diese Funktion gibt die Helligkeit eines Partikelbildes zu einer gewissen Zeit an einem gewissen Ort an.

Vor- und Nachteile:

– Somit kann der reelle Anteil der Verschiebung $\delta x, \delta y < 0.5$ bzw. der Geschwindigkeit $\vec{v}$ basierend auf den Informationen aus dem gesamten Abfragefenster ermittelt werden, was aber sehr ineffizient ist.

+ Nichtsdestotrotz erreicht man damit Genauigkeiten im Bereich 0.01 Pixel im Gegensatz zu den 0.04 Pixeln mit der herkömmlichen „Gaussian-fit"-Methode.

– Ferner kann diese Methode nur unter bestimmten, in [SNOO00] angeführten Bedingungen entsprechende Genauigkeiten liefern.

2.3.3 Eine PIV-Methode mit einer Genauigkeit 2. Ordnung

Der bisherigen PIV-Analyse liegt ja mit den Aufnahmen zum Zeitpunkt t und Δt eine Vorwärtsdifferenzenmethode zugrunde, da man das Vektorfeld zum Zeitpunkt t bestimmen will. Die Fensterverschiebetechnik beobachtet bekannterweise einen um einen bereits bestimmten Schätzervektor verschobenen Bereich der Aufnahme zur Zeit Δt in Form des Suchfensters.

[WM01] haben diese von ihnen als „Forward Difference Interrogation" (FDI) bezeichnete Methode durch einen eigenen Ansatz namens „Central Difference Interrogation" (CDI) — jedoch auch mit der KK als Grundlage — ausgetauscht. Zusätzlich wird die in 2.1.4 erwähnte Technik der kontinuierlichen Fensterverschiebung eingesetzt. Die allgemein bekannten zentralen Differenzen haben einen Fehler 2. Ordnung und sind daher um einiges genauer.

Das Abfragefenster für die Rasterzelle $\hat{I}_{X,Y}$ wird nun ebenfalls verschoben, und zwar um $-\vec{r}/2$ des ermittelten Schätzervektors $\vec{r}$. Das Suchfenster auf der zweiten Aufnahme zur Zeit Δt wird analog um $\vec{r}/2$ verschoben (siehe Abbildung 2.6).

Zusätzlich wird die PID-Methode nach [HFW92b] mit einer schnelleren 4- bzw. 9-Punkt-Methode ebenfalls 2. Ordnung realisiert, um Partikelmuster zu korrigieren (Verdrehung) und damit eine intensivere Übereinstimmung der beiden Aufnahmen zu erhalten. Bei der 4-Punkt-Methode werden vier Vektoren zur Bestimmung der Musterverzerrung in 1. Ordnung hinzugezogen, die 9-Punkt-Methode ermöglicht mit neun benachbarten Vektoren eine Verzerrungsberechnung 2. Ordnung. Die 4-Punkt-Methode ist in der Praxis allerdings vorzuziehen. Details sind im entsprechenden Artikel nachzulesen.

Bei guter Implementierung ist der *CDIC*-Ansatz mit der CDI als Grundlage ebenso effizient wie die herkömmliche FDI-Methode. Die Effizienz der *PIVA*-Standardmethoden wird allerdings nicht erreicht, da hier in jedem Fall Bildinterpolation der Zeit $O(N^2)$ auftritt. In dem späteren Artikel [WG03] findet sich ein experimenteller Vergleich zwischen FDI und der CDIC-Methode mit dem erwartungsgemäßen Ergebnis, dass CDIC entschieden besser als FDI ist. CDIC schneidet hier auch gut gegenüber dem „Minimum Quadratic Difference"-Ansatz (MQD) von [GM96] ab. MQD ist im Prinzip kaum vom D-KK-Ansatz zu unterscheiden, mit dem Unterschied der Suche nach einem Minimum im Gegensatz zur Suche eines Maximum. Ferner wird dabei auch eine nicht einheitliche Verteilung der Partikelbilder bzw. der Beleuchtungsintensität berücksichtigt[10]. Wie in [GM00] gezeigt, kann die MQD ähnlich der FFT-KK auch via FFT effizient berechnet werden.

[10] Ähnlich wie bei der D-NKK.

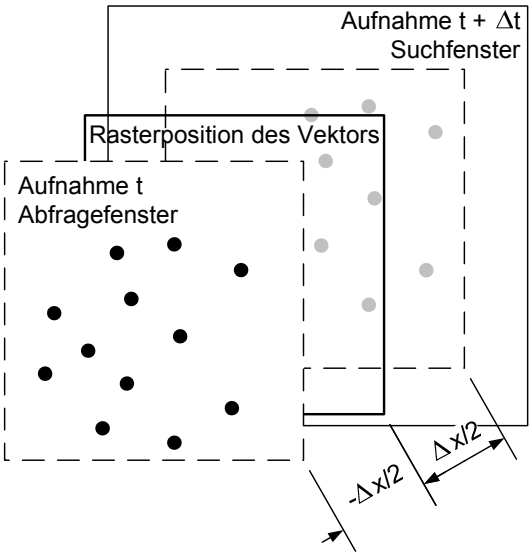

Abbildung 2.6: Die „Central Difference Image Correction"-Methode (CDIC)

2.3.4 Weitere Ansätze mit hoher Genauigkeit

Auf folgende Artikel bin ich im Zuge meiner Recherche gestoßen und möchte es dem Leser selbst überlassen, sich nach Bedarf entsprechend einzuarbeiten.

[WG03] vergleicht jeweils korrelationsbasiertes PTV und PIV mit bzw. ohne der CDIC-Methode und der kontinuierlichen Fensterverschiebung. PIV mit CDIC schneidet davon am besten ab. [Sca02] beschreibt diese CDI-Technik im Vergleich zur FDI-Technik ebenfalls recht anschaulich. Weitere Informationen über das PTV-Verfahren und einen mit PIV kombinierten Ansatz zur Erreichung hoher Vektorfeldauflösungen kann man in [KAZ95] nachlesen.

Von [TD95] wird die sogenannte „Correlation Image Velocimetry" (CIV) erwähnt, welche mit einer Partikelmusterdeformation höherer Ordnung sehr gute Ergebnisse liefert, aber dementsprechend ineffizient ist. Laut [Sca02] sind DMV und CIV durchaus brauchbar, jedoch bzgl. den Ergebnissen bei der Ermittlung der Wirbelstärke noch nicht genügend getestet worden.

In der von [FD00] vorgeschlagenen Methode *PASTIS*, einer Erweiterung des CIV-Algorithmus, wird die Technik des größeren Suchfensters verwendet neben einer Sub-Pixel-Fensterverschiebung.

Die Methode der orthogonalen dynamischen Programmierung (ODP) wurde von [Qué92] in der PIV-Analyse (ODP-PIV) eingeführt und in [QPK98] erweitert angewendet. Diese Technik leitet sich ursprünglich aus der Spracherkennung ab und fungiert als Erweiterung der Sub-Pixel-Fensterverschiebetechnik. Die Bereiche der beiden Aufnahmen im Abfrage- und Suchfenster werden streifenweise verglichen.

Eine ausführliche Betrachtung der bei PIV-Analysen auftretenden Fehler mit deren Ursachen und Wirkungen findet man in [HDG97]. [Har98b] schreibt ferner über die Vermeidung von Fehlern bei der Analyse. Ähnliche Untersuchungen hinsichtlich der Genauigkeit von PIV-Methoden zusammen mit sehr anschaulichen Bildern bietet der technische Report [Bol99].

2.3.5 PIV-Analyse durch Merkmalsverfolgung

Ein grundsätzlich zu unterscheidender Ansatz wird von [Che01] vorgeschlagen. Kurz erklärt, geht es um die Anwendung von sogenannten „*Feature Tracking*"-Methoden als Ersatz zu traditionellen, korrelationsbasierten Methoden. Eingegangen wird auf zwei merkmalbasierende Verfolgungstechniken: dem „KLP-Tracker" und dem „IPAN-Tracker". Merkmale (z. B. Kanten) — damit sind hier Bildmerkmale im Gegensatz zu Strömungsmerkmalen gemeint — werden an beliebigen Stellen extrahiert und in verschiedenen Bildern verglichen. Positive Vergleiche führen zu Verschiebungsvektoren. Die Ausreißer werden wie in 2.1.8 erklärt nach einem „Local-median"-Test erkannt und gelöscht. Um mit diesen Ergebnissen weiterarbeiten zu können, werden die irregulär verteilten Vektoren auf ein reguläres Positionsgitter[11] interpoliert. Jeder Vektor $\vec{v}$ auf dem regulären Gitter wird durch seine irregulären Nachbarn — exponentialverteilt im Abstand zu $\vec{v}$ gewichtet — interpoliert. Tests werden auch auf den künstlichen Aufnahmen von [ONSK00] durchgeführt.

2.3.6 Angepasste Methoden in der μ-PIV

Gegenstand aktueller Forschung in den letzten Jahren ist das sogenannte *μ-PIV* oder *Micro-PIV*. Aus dem Namen lässt sich schon folgern, dass es sich hier um die Beobachtung von mikroskopischen Strömungen und Strukturen handelt. Dort herrschen dementsprechende Verhältnisse wie in [WPL02] beschrieben. Derart kleine Strukturen, die mittels starker Vergrößerung beobachtet werden, sind meist von der BROWN'schen Molekularbewegung[12] beeinflusst. Um diesen Effekt und eine hier oft sehr niedrige Bildintensität zu kompensieren wird eine sogenannte „*Ensemble*"-Korrelation verwendet. D. h. die Kreuzkorrelation an einer bestimmten Stelle in der ersten Aufnahme wird auch in 10 bis 20 aufeinanderfolgenden Aufnahmen berechnet. Das daraus erhältliche Stichprobenmittel aller Korrelationsebenen bietet ein entscheidend größeres (besseres) Signal/Rauschen-Verhältnis bzw. genauere Verschiebungsvektoren. In konventionellen PIV-Methoden würde dieser Ansatz kaum Verbesserungen mit sich bringen, da hier die auftretenden Merkmale sich schneller verändern als die Frequenz der Aufnahmen und daher ein Stichprobenmittel verzerrte Werte mit sich bringen würde. Jedoch in vielen mikroskopischen Anwendungen findet man meist stationäre Strömungen, welche als Vorbedingung der „*Ensemble*"-Korrelation gelten. Dazu sei aber erwähnt, dass man diese Aussage nicht für biologische Strukturen mit u. U. Eigenleben treffen kann (siehe Abschnitt 8.3).

Die Artikel [DWD+99] und [MWS00] beschreiben die Technik der „*Ensemble*"-Korrelation im Detail. Weitere Anwendungen in der Disziplin *μ-PIV* findet man in [KLS04].

[11] Ein Gitter mit gleichen horizontalen und vertikalen Abständen der Gitterlinien und der Zuordnung eines Vektors zu jeder Kreuzung dieser Linien.

[12] thermisch getriebene Eigenbewegung der Moleküle

3 Erkennung von Merkmalen in Strömungen

Im Gegensatz zu den Merkmalen in Bildern betrachtet man als Strömungsmechaniker vor allem die aus den Vektordaten berechenbaren Strömungsmerkmale

Die PIV-Analyse dient in erster Linie der Rekonstruktion von Geschwindigkeitsinformationen aus zeitlich aufeinanderfolgenden Aufnahmen. Hat man nun ein mit Hilfe der im letzten Abschnitt erklärten Methoden rekonstruiertes Vektorfeld zur Verfügung, kann man sich der eigentlichen Analyse der Strömungsmerkmale widmen. Die extrahierten Vektordaten können nun genauso wie per direkter, numerischer Simulation (DNS) berechnete Daten ausgewertet werden. In den folgenden Teilabschnitten werden einige für Strömungsanalysen wichtige Merkmale eingeführt. Folgende Informationen basieren auf der Dissertation [Wes93]. Strömungsmerkmale sind einer der wichtigsten Bestandteile der Strömungsvisualisierung und -analyse. Es sei hier erwähnt, dass der Schwerpunkt dieser Diplomarbeit auf der Datenrekonstruktion lag und die Themen dieses Abschnittes daher nur oberflächlich behandelt werden.

3.1 Bestimmung der Wirbelstärke

Die sogenannte Wirbelstärke (Rotationsvektor) $\vec{\omega}$ (engl. „vorticity vector") beschreibt die Zirkulation der Strömung an einer bestimmten Stelle (i, j). Er ist definiert durch die Rotation des Vektorfeldes $\vec{\omega} = \nabla \times \vec{v}_{i,j}$ mit dem dreidimensionalen Geschwindigkeitsvektor $\vec{v}_{i,j} = (u, v, w)^T$ und $\vec{\omega} = (\omega_X, \omega_Y, \omega_Z)^T$ und kann in unserem ausschließlich zweidimensionalen Fall auf die Komponente

$$\omega_Z = \frac{\partial v}{\partial X} - \frac{\partial u}{\partial Y}$$

bzw. auf diskreten Daten durch die Anwendung zentraler Differenzen 1. Ordnung auf

$$\omega_Z \sim \frac{v_{i+1,j} - v_{i-1,j}}{2\Delta X} - \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta Y} \tag{3.1}$$

reduziert werden. Um die Verstärkung des Rauschens im Vektorfeld durch die Differentialgleichung 3.1 zu vermindern, sollte man dieses voher mit einem geeigneten Tiefpassfilter glätten.

3.2 Bestimmung der Deformation

Großes Interesse kann auch am Zusammenhang zwischen verschiedenen Regionen einer Strömung und an der dort ausgetauschten kinetischen Energie oder Scherspannung bestehen. Deshalb untersucht man häufig die *Topologie einer Strömung*, d. h. die Klassifizierung

ihrer Strömungsmerkmale in bestimmte *Strömungstypen*. Dazu kann man ein Strömungsfeld je nach den Werten der zweiten Invariante des Deformationstensors II, des Druckes p und der absoluten Geschwindigkeit $\|\vec{v}\|$ relativ zu deren Effektivwerten[1] in drei Strömungstypen zerlegen.

Die zweite Invariante des Deformationstensors lässt sich durch

$$II = \sum_{i,j\in\{1,2,3\}} \frac{\partial v_i}{\partial X_j}\frac{\partial v_j}{\partial X_i} = \sum_{i,j\in\{1,2,3\}} \left(E_{i,j}^2 - \frac{1}{2}\omega_i^2\right) \tag{3.2}$$

bestimmen, wobei $\vec{v} = (v_1, v_2, v_3)$, $\vec{X} = (X_1, X_2, X_3)$ und der Verformungstensor $E_{i,j}$ gegeben ist durch

$$E_{i,j} = \frac{1}{2}\left(\frac{\partial v_i}{\partial X_j} + \frac{\partial v_j}{\partial X_i}\right).$$

Mit nur zwei vorhandenen (ebenen) Geschwindigkeitskomponenten in X_1- und X_2-Richtung lässt sich Gleichung 3.2 vereinfachen und mittels Vorwärtsdifferenzen 1. Ordnung der Komponenten $v_1 \equiv u$ bzw. $v_2 \equiv v$ an einer diskreten Stelle (i, j) des Vektorfeldes umwandeln in

$$II_{i,j} \sim \left(\frac{u_{i+1,j} - u_{i,j}}{\Delta X_1}\right)^2 + \left(\frac{v_{i,j+1} - v_{i,j}}{\Delta X_2}\right)^2 + 2\left(\frac{v_{i+1,j} - v_{i,j}}{\Delta X_1}\right)\left(\frac{u_{i,j+1} - u_{i,j}}{\Delta X_2}\right). \tag{3.3}$$

Dem nicht fachkundigen Leser empfehle ich an dieser Stelle eine Einführung in die Strömungsmechanik wie z. B. [OB02]. Eine für Informatiker und Numeriker geeignete detaillierte Behandlung der Themen dieses Abschnittes erhält man mit [GDN95].

Wir kommen nun wieder zurück zur Klassifikation in die drei folgenden Strömungstypen (siehe Abbildung 3.1):

Wirbel (**E**) sind Zonen mit großer Wirbelstärke und werden bei Erfüllung von $II_{i,j} < -2II_{rms}$ und $p < -0.2p_{rms}$ also solche deklariert.

Konvergenzzonen (**C**) sind Bereiche, welche nicht rotieren und starke konvergente bzw. divergente Stromlinien aufweisen. Sie müssen den Bedingungen $II > II_{rms}$ und $p > p_{rms}$ genügen.

Strömungszonen (**S**) sind Regionen mit kaum deformierten, geradförmigen Stromlinien und können durch $v_i^2 > v_{i_{rms}}^2$ für $i \in \{1,2\}$ und $|II| < II_{rms}$ entdeckt werden.

Als grundsätzliches Problem bei der Analyse dieser Strömungstypen besteht die Verstärkung des Rauschens im Vektorfeld durch finite Differenzen 1. Ordnung. Bei schlechten Ergebnissen sollte man auf Differenzen 2. Ordnung umsteigen bzw. wie in Abschnitt 3.1 vorgehen.

3.3 Weitere Merkmale von Strömungen

[RWK98] behandelt das Thema der Datenreduktion bzw. Extraktion von Merkmalen nach einer Rekonstruktion des Vektorfeldes vielfältig und erwähnt neben den bereits von [Wes93] behandelten Strömungsmerkmalen — Wirbelstärke und Deformation — auch noch folgende Kennwerte, auf welche hier aber nicht näher eingegangen wird:

[1] Im Englischen „Root-Mean-Square“ (RMS) genannt, geben sie die Quadratwurzel des arithmetischen Mittels der Quadrate an und werden hier mit dem Subskript „*rms*“ gekennzeichnet.

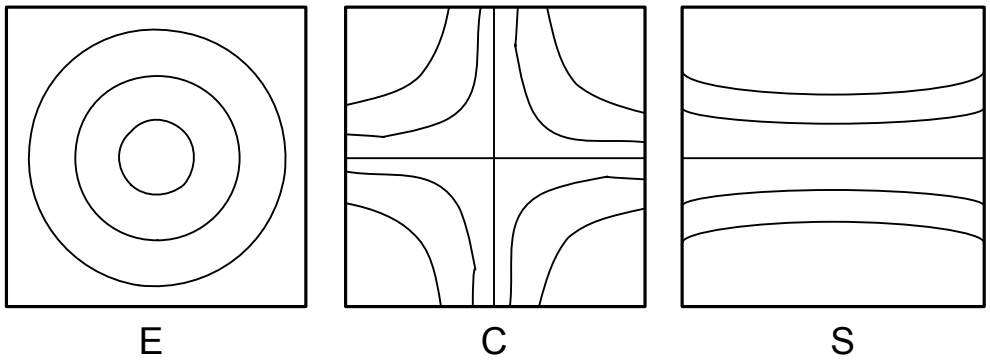

Abbildung 3.1: Die drei Strömungstypen nach [Wes93]: E = Wirbel, C = Konvergenzzonen und S = Strömungszonen.

- die Divergenz und
- die Rotation des Vektorfeldes (Wirbelstärke),
- das globale Mittel der Geschwindigkeiten,
- die Zirkulation bzw.
- der Massenfluss entlang einer Kurve (via Kurvenintegral),
- der Fluss durch eine bzw.
- das Potential auf einer Fläche (via Flächenintegral) und
- das Druckfeld berechnet mittels NAVIER-STOKES-Gleichungen.

Ferner berichtet [RWK98] von der Weiterverarbeitung der PIV-Vektordaten in neuronalen Netzen (siehe [KSK+02]) oder per „Proper Orthogonal Decomposition“[2] (POD) und von Vergleichen mit per NAVIER-STOKES-Gleichungen simulierten Daten. Es werden Finite-Differenzen-Methoden verschiedener Ordnung wie z. B. die Vorwärts- und Rückwärtsdifferenzen, die zentralen Differenzen sowie die Methode der kleinsten Quadrate und die RICHARDSON-Extrapolation angesprochen. Letztere hat von allen den Fehlerterm höchster Ordnung und erbringt somit beste Ergebnisse. Sie wird deshalb bei der Differentiation im Rahmen eines adaptiven Verfahrens zur genauen Berechnung der Wirbelstärke von [LK95] verwendet.

[2] Siehe Glossar für nähere Erklärung bzw. [Cha00].

4 Methodische Hilfsmittel aus der digitalen Bildverarbeitung

Die Detektion von brauchbaren Merkmalen in Bildern hängt sehr stark mit der Anwendung passender digitaler Filter zusammen

Ein nicht zu unterschätzender Teil der PIV-Analyse bis hin zu den rekonstruierten Vektordaten besteht aus Verfahren der digitalen Bildverarbeitung. Ich möchte dem allgemeinen Leser auch hier etwas Hintergrundwissen vermitteln und daher die im Zuge der Diplomarbeit verwendeten Methoden kurz erklären. Es handelt sich hier um häufig verwendete Algorithmen zur Hoch- bzw. Tiefpassfilterung von Signalen. In diesem Teil wird auch die Kantendetektion in Bildern zur Merkmalsextraktion besprochen. In den späteren Abschnitten wird jedoch an den nötigen Stellen auf das hier Besprochene verwiesen, sodass dieser Abschnitt vorerst übersprungen und nach Bedarf als Nachschlageteil eingesehen werden kann. Ich verweise auf das anschauliche, im Internet verfügbare Nachschlagewerk [FPWW04] und auf die Einführung in die digitale Bildverarbeitung von [Jäh02].

4.1 Hoch- und Tiefpassfilter im Allgemeinen

Betrachtet man ein Bild, sei es kontinuierlich oder diskret, als ein entsprechend periodisches Signal, so kann man dieses mit der bekannten FOURIER-Transformation vom sogenannten Ortsbereich in den Frequenzbereich transformieren und erhält dadurch die Anteile der im Bildsignal vorkommenden Frequenzen[1]. Den Frequenzbereich kann man in verschiedene Bänder beliebiger Breite (Frequenzteilbereiche) unterteilen und analysieren. Aus der Forschung weiß man schon seit langem, dass gewisse Frequenzbänder je nach Art des Signals eine spezielle Bedeutung im Ortsbereich haben. Zu diesen Bändern zählt z. B. das sogenannte Rauschen, ein i. d. R. als Fehler definierter Anteil eines Signales. Rauschen kann durch verschiedene Ursachen entstehen, dementsprechend gibt es in elektronischen Signalen z. B.

- das *weiße Rauschen*, ein über dem gesamten Frequenzbereich vorhandener Zufallsfehler[2],
- das *Funkelrauschen*, welches nur zwischen 0 und 100 Hz liegt, außerhalb aber vom weißen Rauschen überlagert wird,

[1] FOURIER hat ja einst gezeigt, dass man ein beliebiges T-periodisches Signal durch eine Summe von Anteilen T-harmonischer Frequenzen (Kreisfrequenzen der Form $\omega = 2\pi n/T$ mit $n \in [0, \infty)$) darstellen kann.

[2] Die spektrale Dichte ist frequenzunabhängig. Je nach Filterung kann sich das weiße Rauschen zu einem farbigen — z. B. rosa oder blauen — Rauschen verändern. Die „Farbe“ des Rauschens entspricht der Zuordnung des Rauschfrequenzbereiches zum herkömmlichen Farbspektrum.

- das *Stoßrauschen* oder auch *Impulsrauschen*, das keine bestimmte Frequenz einnimmt, und
- das *Schrotrauschen* oder auch *Widerstandsrauschen* mit einem dem weißem Rauschen ähnlichen Frequenzverlauf.

Eine mathematische Betrachtung des Rauschens in Bilddaten ist z. B. in [Kun02] zu finden. Ich kann dieses Vorlesungsskript wegen seiner guten Qualität wärmstens empfehlen.

Rauschen führt oft zu verzerrten oder falschen Ergebnissen und ist daher meistens unerwünscht. An dieser Stelle kommen die sogenannten *Filter* zum tragen, welche mittels geeigneter *Filterfunktion* gewisse Frequenzbänder aus dem Gesamtsignal dämpfen oder sogar löschen. Da man aber im Frequenzbereich nicht die häufig nötigen Informationen über den Ort oder die Zeit der vorkommenden Frequenzen hat, möchte man Filter verstärkt im Ortsbereich anwenden. Dort kann man sich einen Filter als räumlich begrenzte gewichtete Funktion vorstellen, welche mittels Konvolution[3] auf dem Signal angewendet wird. Filter können z. B. mit der sogenannten *gefensterten* FOURIER- oder auch mit der *Wavelet*-Transformation verwirklicht werden (siehe [Gra95] bzw. [Mal99]).

4.2 Der GAUSS'sche Unschärfefilter

Im Rahmen dieser Diplomarbeit wird der sehr weit verbreitete GAUSS'sche Glättungs- oder Unschärfefilter mehrfach verwendet. Der hier betrachtete zweidimensionale, kreissymmetrische Konvolutionsoperator ist definiert durch die zweidimensionale GAUSS'sche Normalverteilung[4]

$$G(x,y) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.1}$$

mit der Standardabweichung σ und dem Mittelwert $(0,0)$.

Ein an diskreten stellen ausgewerteter Konvolutionskern $\hat{G}$ der Größe $(2N+1)\times(2N+1)$ mit dem halben Kerndurchmesser $N=2$ und der Standardabweichung $\sigma=1$ hat eine Form wie in Abbildung 4.2 gezeigt. Er stellt eine diskrete, begrenzte Filterfunktion für die Faltung mit einem Bildsignal dar. Der GAUSS'sche Filter kann effizient durch Dekomposition der Funktion 4.1 in x- und y-Richtung implementiert werden (siehe [FPWW04]).

Dieser diskrete Konvolutionskern $\hat{G}$ wird nun auf ein Bild I angewendet, in dem für jede Stelle (x,y) des Bildes der Bereich $I(x-N,y-N)$ bis $I(x+N,y+N)$ mit $\hat{G}$ durch

$$\tilde{I}(x,y) = \sum_{-N\leq i,j\leq N} I(x+i,y+j)\hat{G}(i,j) \tag{4.2}$$

gefaltet wird und $\tilde{I}$ das geglättete Bild darstellt. In die Berechnung eines Pixels $\tilde{I}(x,y)$ fließen also die normalverteilten Anteile der innerhalb des Wirkungsbereichs von $\hat{G}$ benachbarten Pixel von $I(x,y)$ ein. Hier gilt $(x,y)\in\{(x,y)|N\leq x<(D_X-N)\wedge N\leq x<(D_Y-N)\}$, wobei D_X und D_Y die Abmessungen des Bildes in x- und y-Richtung darstellen. Es empfiehlt sich, die Bilder um einen Rand von N auf allen Seiten zu erweitern, damit man den Konvolutionskern auf jeden Pixel des Eingabebildes schieben kann. In den Randbereichen kann es zu verzerrten Glättungsergebnissen kommen, da hier i. d. R. mit Nullen erweitert wird.

[3] Auch Integralfaltung genannt, siehe Abschnitt B.
[4] Auch als Glockenfunktion bezeichnet.

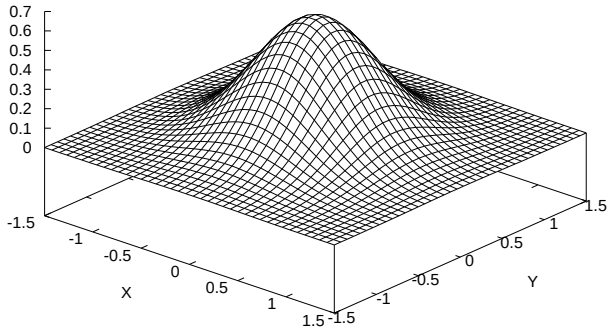

Abbildung 4.1: Die kontinuierliche GAUSS'sche Glockenkurve um $(0,0)$ mit $\sigma = 1$. Die Z-Achse kann als Gewichtung interpretiert werden.

0.004	0.015	0.026	0.015	0.004
0.015	0.059	0.095	0.059	0.014
0.026	0.095	0.150	0.095	0.026
0.015	0.059	0.095	0.059	0.014
0.004	0.015	0.026	0.015	0.004

Abbildung 4.2: Diskreter GAUSS'scher 5×5-Konvolutionskern $\hat{G}$ $(N = 2)$

4.3 Filteroperatoren für die Kantendetektion in Bildern

Im Gegensatz zu einem Glättungsfilter, der hohe Frequenzen mit Rauschanteilen heraus filtern soll, geht es hier um die örtliche Detektion von hohen Frequenzanteilen, in Bildern schlichtweg als Kanten bezeichnet. Das Filterprinzip per Konvolution wurde bereits in Abschnitt 4.2 erklärt und kann hier analog angewendet werden. Als Konvolutionskern können nun sogenannte Kantendetektionsoperatoren verwendet werden. Dabei gibt es eine ganze Reihe von verschiedenen Detektoren. Im Rahmen dieser Diplomarbeit sollen hier aber nur Operatoren basierend auf dem Helligkeitsgradienten, speziell der SOBEL- und eine allgemeine Form des CANNY-Operators, genauer betrachtet werden. In den Testexperimenten in Abschnitt 8 kann die Qualität der Ergebnisse der PIV-Analyse mit beiden Kantendetektoren eingesehen werden.

4.3.1 Der SOBEL-Operator

Der SOBEL-Operator arbeitet hier auf Schwarz-Weiß-Bildern, reagiert auf Helligkeitsveränderungen dem Gradienten entsprechend und kennzeichnet auf diese Weise örtlich hohe Frequenzen in seiner Ausgabe. In dieser Arbeit wird er intuitiverweise als Hochpassfilter bezeichnet, obwohl er nicht nur Frequenzen filtert, sondern auch verändert. Er besteht aus den zwei[5] 3×3-Teiloperatoren (gezeigt in den Abbildungen 4.3 und 4.4), die getrennt mit dem Eingabebild konvolviert werden. S_X reagiert am stärksten auf vertikale Kanten, S_Y dagegen auf horizontale. Aus den per S_X bzw. S_Y richtungsabhängig kantengefilterten Bildern $\tilde{I}_X$ bzw. $\tilde{I}_Y$ kann durch

$$|\tilde{I}(x,y)| = \sqrt{\tilde{I}_X^2 + \tilde{I}_Y^2} \tag{4.3}$$

das in beiden, linear unabhängigen Raumrichtungen X und Y gefilterte Kantenbild $\tilde{I}_{X,Y}$ ermittelt werden. Der SOBEL-Operator wirkt in gewissem Maße glättend auf das Eingabebild und ist daher im Gegensatz zu anderen Kantendetektoren relativ unempfindlich gegenüber Rauschen.

[5] Es gibt auch eine approximierende 1-Komponentenversion dieses Operators, auf welche hier aber nicht weiter eingegangen wird (siehe [FPWW04]).

-1	0	+1
-2	0	+2
-1	0	+1

Abbildung 4.3: SOBEL-Operator S_X für vertikale Kanten

-1	-2	-1
0	0	0
+1	+2	+1

Abbildung 4.4: SOBEL-Operator S_Y für horizontale Kanten

Des weiteren empfiehlt es sich, eine umfassende Einführung in die Merkmalsextraktion wie z. B. [NA02] zur Hand zu nehmen.

4.3.2 Eine durch Wavelets generalisierte, multiskalare Form des CANNY-Operators

Eine Verbesserung gegenüber dem älteren und funktional beschränkten SOBEL-Operator stellt der CANNY-Operator dar, dessen Testresultate wir später in Abschnitt 8 sehen werden. Entwickelt und definiert wurde der CANNY-Operator in [Can86]. Der ursprüngliche Gedanke des CANNY-Operators war, die relativ unflexiblen, älteren Kantenfilter durch einen kontinuierlichen, skalierbaren Filter zu ersetzen.

Für dieses Kapitel empfehle ich das Buch von [Jäh02], welches eine grundlegende Einführung in die Kantendetektion in digitalisierten Bildern bietet. Zum tieferen Verständnis der folgenden Ausführungen und deren Herleitung verweise ich zusätzlich auf zwei Artikel. [MZ92] bietet eine problembezogene Einführung in die Theorie der Wavelets, generalisiert den CANNY-Operator durch ein Paar von parametrisierten, hierarchischen (multiskalaren) Wavelets und geht etwas tiefer in die Mathematik. Ein genaueres mathematisches Verständnis dieser Theorien ist für die Betrachtung des hier Erklärten nicht zwingend notwendig, da hauptsächlich auf die Implementierung eingegangen wird. [Chr04] erklärt eine Form der praktischen Implementierung und Anwendung eines solchen Kantendetektors. Es sei hier noch erwähnt, dass wir in dieser Arbeit nur die Dekomposition von Bildern vornehmen. Zur möglichen Rekonstruktion der originalen Bilddaten aus den multiskalaren Kantenbildern verweise ich auf die eben genannten Artikel.

Der ursprüngliche CANNY-Operator

Der CANNY-Operator ist genauso wie der SOBEL-Operator ein regularisierter Kantendetektor, da er eine Glättung des Bildsignals beinhaltet. Optimale Kantendetektoren müssen auf den Frequenzbereich eingestellt werden, bei dem das maximale Signal/Rausch-Verhältnis vorliegt.

Genauer definiert stellt der CANNY-Detektor einen Projektionsoperator $G_{\vec{n}}$ dar, welcher mit einem Bild $I(x,y)$ mit $(x,y) \in \mathcal{B}$ gefaltet wird. Die Nulldurchgänge der resultierenden Funktion $G_{\vec{n}} * I(x,y)$ können dann als Kanten von I betrachtet werden.

Damit der Projektionsoperator $G_{\vec{n}}$ die beschriebene Eigenschaft erlangt, muss er zwei Bedingungen genügen. Zum Ersten muss er das Signal/Rausch-Verhältnis maximieren und zum Zweiten eine maximale Lokalisation ermöglichen. CANNY hat demzufolge als gute Approximation für diesen Operator die erste Ableitung der GAUSS'schen Glocke errechnet. Die

erste Richtungsableitung dieser in Gleichung 4.1 beschriebenen, zweidimensionalen Funktion nach dem Normalenvektor $\vec{n}$ zur Kantenrichtung gestaltet sich durch

$$G_{\vec{n}} = \frac{\partial G}{\partial \vec{n}} = \vec{n} \cdot \nabla G. \tag{4.4}$$

Da wir die Richtung des Kantenvektors nicht a priori kennen, verwenden wir hier die Approximation $\vec{n} \approx \frac{\nabla(G*I)}{|\nabla(G*I)|}$. Das ist also der normierte Gradientenvektor der Faltung von G und I. Die Nulldurchgänge $G_{\vec{n}} * I = 0$ müssen damit also durch $\frac{\partial}{\partial \vec{n}}(G_{\vec{n}} * I) = 0$ berechnet werden. Durch Einsetzen von Gleichung 4.4 erhalten wird dann

$$\frac{\partial^2}{\partial \vec{n}^2}(G * I)(x,y) = 0. \tag{4.5}$$

Gleichung 4.5 reagiert damit auf die Wendepunkte der Kanten eines GAUSS-geglätteten Bildes I.

Da wir nun nur die Standardabweichung σ in $G(x,y)$ anpassen können, um bessere Ergebnisse der Detektion zu erhalten, liegt es nahe, eine multiskalare Version des CANNY-Operators mit Wavelets zu verwenden, welche einen Skalierungsparameter als weiteren Freiheitsgrad zur Verfügung stellt.

Der Wavelet-Ansatz zur Kantendetektion

[MZ92] hat den CANNY-Detektor generalisiert, indem er für die Glättungsfunktion $G(x,y) = \theta(x,y)$ ein Paar von Wavelets

$$\psi_j^x = -2^j \frac{\partial \theta_j(x,y)}{\partial x}, \quad \psi_j^y = -2^j \frac{\partial \theta_j(x,y)}{\partial y} \tag{4.6}$$

verwendet hat (siehe Abbildung 4.5 und 4.6). ψ^x und ψ^y werden mit $j \in \mathbb{N}_0$ dyadisch dilatiert[6], um eine Skalierung der Form $\theta_j(x,y) = 2^{-j}\,\theta(2^{-j}x, 2^{-j}y)$ zu erhalten. j stellt damit den zweiten, einstellbaren Parameter der Kantendetektion dar.

Die Wavelet-Transformation (WT) der j-ten Skala in einer Richtung $k \in \{x,y\}$ ist damit durch

$$W_j^k I(x,y) = (I * \psi_j^k)(x,y) \tag{4.7}$$

gegeben. Mit Gleichung 4.6 können wir die gesamte WT als

$$\begin{pmatrix} W_j^x I(x,y) \\ W_j^y I(x,y) \end{pmatrix} = 2^j\, \nabla(I * \theta_j)(x,y) \tag{4.8}$$

schreiben.

Da diese WT proportional zum Gradienten des durch θ_j geglätteten Bildes I ist, sei schon vorweggenommen, dass man die Wendepunkte des Bildes I (die Kanten) nun über die Maxima des Moduls[7] der WT finden kann.

[6] Eine dyadische Dilatation eines Wavelets ψ^k mit $k \in \{x,y\}$ ist eine Skalierung in 2-er Potenzen, d. h. die Wavelet-Funktion wird mit einem Faktor 2^j zentral in beide Achsenrichtungen von k gedehnt oder gestaucht.

[7] Auf die genaue Berechnung des Moduls werden wir später mit Gleichung 4.13 eingehen.

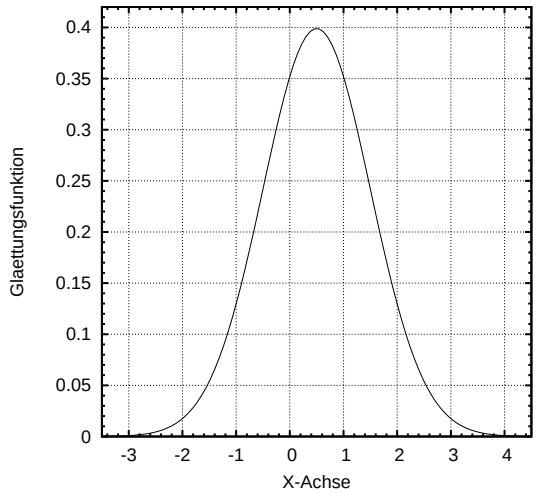

Abbildung 4.5: Eine konkrete Glättungsfunktion gewählt mit $\theta(x,y) = G(x,y)$ für $y = 0$

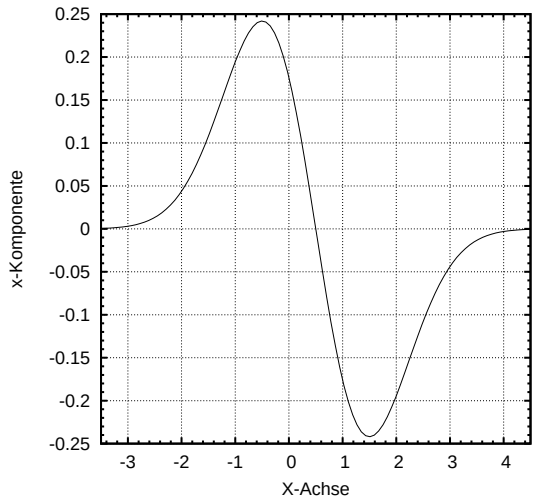

Abbildung 4.6: Die x-Komponente ψ^x als Wavelet der einfach abgeleiteten und dilatierten Glättungsfunktion $G(x,y)$ für $y = 0$

Implementierung im diskreten Bildraum

Im Falle eines diskreten Signals bzw. eines digitalisierten Bildes kann man die Glättungsfunktion aus Gleichung 4.1 und die Wavelets aus Gleichung 4.6 durch die folgenden Filterbänke[8] für die Dekompositionsfilter ersetzen:

- Die Filterbank für die multiskalare[9] Glättungsfunktion wird aus dem diskreten quadratischen „Spline"-Filter $h_0[n] = \{1, [3], 3, 1\} \cdot \sqrt{2}/4$ mit dem gekennzeichneten $h_0[0] = 3$ durch sogenanntes „Upsampling"[10] mit

$$h_j[n] = (h_0 \uparrow j)[n] \tag{4.9}$$

  generiert.

- Die Filterbank für den normalisierten Finite-Differenzen-Filter zur Approximation des Helligkeitsgradienten erhalten wir mit

$$g_j[n] = 0,\ g_j[\mp 2^{j-1}] = \pm\sqrt{2}/2 \quad \text{und} \quad g_0[0] = -\sqrt{2}/2,\ g_0[1] = \sqrt{2}/2. \tag{4.10}$$

- Der diskrete, eindimensionale DIRAC-Filter (oft auch als KRONECKER-Symbol bezeichnet) ist definiert durch

$$\delta[t] = \begin{cases} 1, \text{wenn} & t = 0 \\ 0, \text{sonst} & \end{cases} .$$

[8] Eine Filterbank ist hier ein Satz von zusammengehörenden Filtern (Konvolutionskernen), wobei die einzelnen Filter der Bank meist skalierte Versionen eines bestimmten Filters darstellen.
[9] in mehreren Skalen vorhandene
[10] Siehe Symboltabelle B.

Es gilt $j \in \mathbb{N}_0$ und $n \in \mathbb{Z}$.

Mit Hilfe der soeben erklärten Filterbänke kann diese diskrete Wavelet-Transformation (DWT) mit einer Kaskade von separierbaren Konvolutionen[11] durchgeführt werden, in jeder Skala 2^j mit den

$$\begin{aligned} &\text{Bildkoeffizienten} && a_{j+1}(x,y) = [a_j * (h_j, h_j)](x,y) \\ &\text{und den Detailkoeffizienten} && \begin{cases} d^x_{j+1}(x,y) = [a_j * (g_j, \delta)](x,y) \\ d^y_{j+1}(x,y) = [a_j * (\delta, g_j)](x,y) \end{cases} . \end{aligned} \tag{4.11}$$

Für diese und folgende Berechnungen muss das Bild I normalisiert werden, sodass $\forall (x,y) \in \mathcal{B} : I(x,y) \in [0,255] \mapsto I'(x,y) \in [0,1]$. Es gilt dann $a_0(x,y) \equiv I'(x,y)$. $h_j[n]$ kann hier als Tiefpassfilter und $g_j[n]$ als Hochpassfilter in jeweils skalierbarer Form angesehen werden.

Wie erwähnt, können wir die gesuchten Kanten nun über die lokalen Maxima des Moduls und des Argumentes[12] der DWT in der Skala 2^j bestimmen. Damit berechnen sich das Modul und das Argument in vereinfachter Schreibweise durch

$$M_j a_0(x,y) = \sqrt{|d^x_j|^2 + |d^y_j|^2} \quad \text{und} \quad A_j a_0(x,y) = arg(d^x_j + \mathrm{i} \cdot d^y_j). \tag{4.12}$$

In der eindimensionalen Nachbarschaft in der Richtung gegeben durch das Argument $A_j a_0$ können wir nun die Menge $\mathcal{P} = \{\vec{u}_{j,p}\}_{p \in \mathbb{N}_0}$ der Positionen der lokalen Maxima des Moduls in der Skala 2^j durch

$$M_j a_0(\vec{u}_{j,p}) > M_j a_0(\vec{u}_{j,p} + \vec{\varepsilon}) \quad \text{und} \quad M_j a_0(\vec{u}_{j,p}) > M_j a_0(\vec{u}_{j,p} - \vec{\varepsilon}) \tag{4.13}$$

bestimmen mit

$$\vec{\varepsilon} = \begin{cases} \vec{\varepsilon}_v = (0,1)^T, \text{wenn} & |A_j a_0(\vec{u}_{j,p}) - A_j a_0(\vec{u}_{j,p} + \vec{\varepsilon}_h)| < |A_j a_0(\vec{u}_{j,p}) - A_j a_0(\vec{u}_{j,p} + \vec{\varepsilon}_v)| \\ \vec{\varepsilon}_h = (1,0)^T, \text{sonst} \end{cases}$$

und $\vec{u}_{j,p} = (u_{j,p}, v_{j,p})^T$. Die Komplexität des bisherigen Algorithmus kann mit $O(N^2 log_2 N)$ beschrieben werden, wenn N die Abmessung eines quadratischen Bildes ist. Die Auflösung der Bilder bleibt über die Filterhierarchie hinweg gleich, da die Aufgabe dieser Prozedur ja nicht in der Kompression der Daten besteht.

Festlegung eines korrelationsbasierten Grenzbereichs zur Kantenelimination

Selbst bei nahezu rauschfreien Aufnahmen können je nach Skalierungsebene „falsche“ Kanten in das Kantenbild mitaufgenommen werden. Mit einer Eingrenzung[13] der Modulmaxima der DWT mit der Lage $u_{j,p}$ auf den Bereich $[T,1]$ können unerwünschte Kanten kaum signifikanter Merkmale entfernt werden.

In der Skala 2^j bezeichnen wir diesen lokalisierten Bereich mit $T(x,y,j)$. Zur Berechnung schlägt [Chr04] folgenden Algorithmus vor, welcher in *PIVA* auch implementiert wird. Eine ausführlichere Beschreibung des Gegenstandes findet man im selbigen Artikel.

[11] Mit $A * (Z,S)$, der Matrix A und den beiden Vektoren Z und S wird die separierbare, schnelle, zweidimensionale Konvolution gekennzeichnet, welche sich mathematisch durch $A * (Z \cdot S^T)$ berechnet. Dieses Vorgehen ist nur bei separierbaren Filtermatrizen $F \equiv Z \cdot S^T$ möglich.

[12] Auf die Ausdrücke „Modul“ und „Argument“ (Winkel) kommt man durch die Betrachtung der beiden Detailkoeffizienten aus Gleichung 4.11 als komplexe Zahl.

[13] Gemeinhin bezeichnet mit dem englischen Wort „Threshold“.

1. Setze $j = 0$, für $j \in [0, J]$ mit J Dekompositionsebenen.

2. Berechne $T(x,y,j) = 1/\Theta_{x,y}(U_j, U_{j+1})$ bzw. $T(x,y,j) = 1/\Theta_{x,y}(U_j, U_0)$ für $j = J$ und $\Theta_{x,y}(\cdot,\cdot) > 0$ mit der Matrix

$$U_j(x,y) = \begin{cases} 1, \text{ wenn } (x,y)^T \in \mathcal{P} \\ 0, \text{ sonst} \end{cases}$$

   der Kantenpunkte $\vec{u}_{j,p}$ der Ebene j. Die Matrix $\Theta(U_j, U_{j+1})$ enthält dann die Korrelationskoeffizienten $\Theta_{x,y} = U_j(x,y) \cdot U_{j+1}(x,y)$ mit 0 bzw. 1 als Ausprägungen der Zufallsvariablen U_j.

3. Falls $\Theta_{x,y}(U_j, U_{j+1}) = 0$, dann setze $T(x,y,j) = \beta$, wobei $\beta \leq 1$ einen weiteren Parameter für die Kantendetektion darstellt und empirisch zu ermitteln ist. Bei $\beta = 1$ würden alle Kantenpunkte, welche in der nächsten Ebene U_{j+1} nicht mehr vorhanden sind, auch aus M_j mittels Korrelation gelöscht.

4. Grenze die gefundenen Kantenpunkte $\vec{u}_{j,p}$ mit den nun berechneten $T(u_{j,p}, v_{j,p}, j)$ aus, wenn $M_j a_0(\vec{u}_{j,p}) < T(u_{j,p}, v_{j,p}, j)$.

5. Erhöhe $j = j + 1$ und fahre bei Schritt 1. fort.

Ziel dieses Algorithmus sind besser definierte — glatte und rauschfreie — Kantenverläufe. Als Alternative gilt die einfachere Methode, den Threshold T einfach konstant zu setzen, was mit einer empirischischen Ermittlung verbunden ist. Um exakte, scharfe Kantenverläufe zu erhalten, kann man auch alle Punkte aus dem Bild filtern, welche keine lokalen Modulmaxima darstellen. Gute Ergebnisse sind ab der 2. Filterebene erzielbar.

4.4 Die GAUSS'sche Pyramide

Mit Hilfe der GAUSS'schen Pyramide (GP) kann man eine mehrstufige Vergröberung der Auflösung von Bildern erreichen. Ein Originalbild I der Größe $B(0) \times H(0)$ mit $B(l) = M_B \cdot 2^{N-l} + 1$, $H(l) = M_H \cdot 2^{N-l} + 1$, $l \in [0, N]$ und frei wählbarem M_B und M_H kann in l Stufen zu einem Bild der Größe $B(l) \times H(l)$ vergröbert werden. Bilder die nicht diesen Abmessungen entsprechen, müssen zuerst um entsprechende Pixelspalten und -zeilen erweitert werden. Um alle Pixel des Bildes I_l der Ebene l in das neue Bild I_{l+1} der Ebene $l+1$ einfließen zu lassen, wird wie in [BA83] erklärt ein $K \times K$-Gewichtungskern[14] w von GAUSS'scher Glockenform verwendet (siehe Abschnitt 4.2). Mit der Vorschrift

$$I_{l+1}(i,j) = \sum_{-K \leq m,n \leq K} w(m,n)\, I_l(2i+m, 2j+n) \tag{4.14}$$

gelangt man somit von der Ebene l in die nächstgröbere Ebene $l+1$, wobei I_0 das Originalbild und I_N das gröbste 1×1 Pixel große Bild darstellt und $(i,j) \in B(l+1) \times H(l+1)$. Das soeben Erklärte wird schematisch in Abbildung 4.7 gezeigt.

[14] Im Artikel [BA83] konzentriert man sich o.B.d.A. auf einen 5×5-Kern.

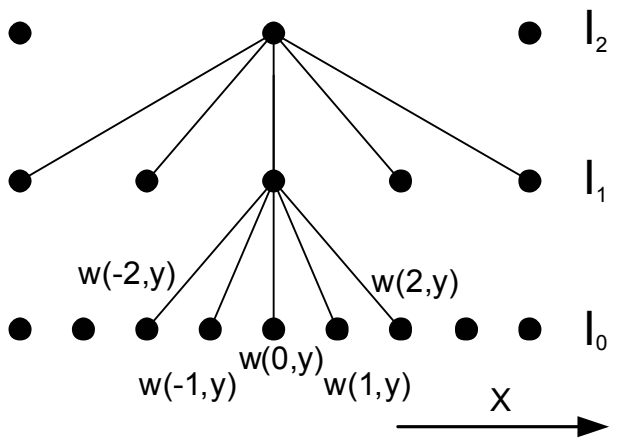

Abbildung 4.7: Eine schematische Darstellung der GAUSS'schen Pyramide im Falle eines 5×5-Gewichtungskernes. Die um 1 verminderte Auflösung halbiert sich pro Ebene.

5 Hierarchische Ansätze in der DPIV

Eine hierarchische, adaptive Analyse von Daten eröffnet neue Möglichkeiten der Rekonstruktion und steigert die Effizienz der Verfahren

In einigen, bereits besprochenen Ansätzen wurde die Möglichkeit der Verwendung verschiedener Auflösungsstufen der Vektorfelder betrachtet. Wir wollen hier noch einmal etwas mehr ins Detail gehen und betrachten einfache hierarchische Techniken und einen adaptiven Ansatz zur lokalen Anpassung der Auflösung der Geschwindigkeitsdaten.

5.1 Hierarchische Verfeinerung des Abfragerasters

Eine gesamtheitliche Verfeinerung des Vektorfeldes wurde bereits in Kapitel 2.1.6 eingehend erklärt. Die dort erläuterten Ideen von [SR99] und [SR00] sind allerdings mit einigen Problemen verbunden.

PIVA implementiert der Einfachheit halber bisher nur eine 1 : 1-Vererbung von Schätzervektoren in die nächstfeinere Hierarchieebene. Dies ist bei linearen Strömungen des Typs S[1] zulässig. Bei Strömungen des Typs E und C empfiehlt es sich jedoch in jedem Fall, eine bilineare Interpolation der Vektoren vorzunehmen, um bessere Ergebnisse in der Korrelationsebene zu erhalten.

Ferner ist zu überlegen, in welcher Art und Weise man das Verhältnis von Abfrage- und Suchfenstergröße N/N_s über die Iterationen und Verfeinerungen hinweg pflegt, um effizient zu bleiben, d. h. so rasch wie möglich mit kleinen Fenstern zu arbeiten. Idealerweise sollte man $\lim_{i \to A_i} N_i/N_{s_i} \approx 1$ erreichen, wenn i die i. Iteration und $A_i \in \{3, 4, \ldots\}$ (üblicherweise) die Anzahl der Iterationen bezeichnen und N_i bzw. N_{s_i} Teil einer entsprechenden Reihe mit $N_{A_i} = N_{s_{A_i}} - 1$ sind. Eine untere Schranke für die Größe von Abfragefenstern von üblicherweise 8×8 Pixeln sollte allerdings eingehalten werden, da für die durchschnittliche Partikelbildgröße d_t meist $d_t \geq 3$ gilt und laut [Wes93] eine Mindestanzahl von 4 Partikelbildpaaren innerhalb eines Abfragefensters existieren sollten, um eine Verschiebung mit hoher Wahrscheinlichkeit richtig zu bestimmen.

Üblicherweise und auch in *PIVA* wird angenommen, dass man nach jeder Iteration näher an die reale Verschiebung herankommt. Mit dieser Vorkenntnis und der (diskreten) Fensterverschiebetechnik kann man das Verhältnis N/N_s nach jeder Iteration etwas mehr in Richtung 1 stellen bei gemeinsamer Verkleinerung von Abfrage- und Suchfenster.

In lokalen Bereichen mit großen Geschwindigkeiten relativ zum Suchfenster und vor allem im Randbereich von Bildern kann man diese Anpassung des Verhältnisses nicht einheitlich bzw. uneingeschränkt vornehmen, da man Korrelation verliert und somit eine höhere Wahrscheinlichkeit für Ausreißer bestehen würde.

[1] Siehe Strömungsklassifikation in Abschnitt 3.2.

[Idee] Ein Ansatz, der nicht von *PIVA* implementiert wird, wäre ein adaptives Verfahren, um das Verhältnis von aktuell ermittelter Verschiebung und aktueller Suchfenstergröße als Kriterium für eine über mehrere Iterationen individuelle Abfrage- und Suchfenstergrößenreihe zu verwenden. Gedanken zum Verhältnis der Suchfenstergröße und der dazu maximal messbaren Verschiebung bietet [SR99].

5.1.1 Optimierung der Effizienz durch Vergröberung bzw. Kompression von Bilddaten

[RWK98] erwähnt in seinem hierarchischen PIV-Algorithmus eine Vergröberung der Auflösung der Aufnahmen in frühen Iterationen zur Ermittlung von Schätzervektoren. Unter der Voraussetzung, dass bei einer gröberen Auflösung genügend tiefe Frequenzinformationen erhalten bleiben, um die für PIV wichtigen Merkmale zu verfolgen, kann man die Effizienz erheblich steigern. Als grobe Richtlinie für diese Voraussetzung gilt der durchschnittliche Partikelbilddurchmesser d_t. Diesbzgl. wären allerdings noch genauere Analysen notwendig.

[Lew95] schlägt im Detail vor, eine sehr effiziente Verfolgung von Merkmalen mit hierarchischem Ansatz via Wavelet-Transformation[2] durchzuführen. In *PIVA* wurde der Ansatz der GAUSS'schen Pyramide zur Hochpassfilterung und Vergröberung[3] von Bildern gewählt.

Die in Abschnitt 4.4 beschriebene Methode von [BA83] wird nun angewendet. Es werden jedoch nur so viele Vergröberungsebenen der Pyramide ausgehend vom Originalbild verwendet wie zulässig. Zulässig sind $l_{max} = \max_l\{d_t/2^l \geq 1\}$ Ebenen mit dem Partikelbilddurchmesser d_t. l_{max} muss kleiner als die Anzahl der Verfeinerungsstufen bis zur entgültigen Vektorfeldauflösung sein, um mindestens in dieser feinsten Auflösungsstufe das Originalbild als Grundlage der Kreuzkorrelation verwenden zu können. Dieses liefert i. A. natürlich die beste Grundlage für gute Ergebnisse. Die somit gewonnene, effiziente Methode wird in Abschnitt 8.2 getestet. Aus der Abbildung 8.5 wird dann ersichtlich, dass die Vergröberung zwar signifikant kürzere Rechenzeiten erzielt, jedoch nicht die Genauigkeit einer herkömmlichen D-(N)KK erreicht.

Am Rande sei noch der Artikel [Har00] erwähnt. Hier wird ebenfalls ein Ansatz für die schrittweise Verfeinerung der Auflösung des Vektorfeldes ähnlich der in *PIVA* angewandten Variante erklärt. Im Zuge einer hierarchischen Verfeinerung der Abfrage- und Suchfenstergrößen mit vererbten Schätzern wird aber eine andere Methode zur Beschleunigung der Korrelation eingesetzt, welche als „kompressionsbasierte Korrelation" bezeichnet wird und von [Har96] eingeführt bzw. von [Har98a] erweitert wurde. Die Kompression beruht auf dem Prinzip, für die Korrelation unwichtige Pixel (mit niederen Graustufen) aus dem Bild zu eliminieren und nur Pixel, welche einen wichtigen Anteil an Partikelbildern haben in einem separaten „dünnen" Feld zu speichern und in die Berechnungen einfließen zu lassen. Um nach der Ermittlung der groben Schätzervektoren in einer finalen Iteration keine Korrelationsdaten zu verlieren, ist die Kompressionsrate bis auf 1:1 (Originalbild) verminderbar. Ferner wird noch eine PIV-Analyse zweiter Ordnung ähnlich wie in Abschnitt 2.3.3 integriert. Somit lassen sich genaue Ergebnisse bis zu Vektorfeldauflösungen in Partikelgröße erreichen; laut diesem Artikel bei besserer Effizienz als mit einer herkömmlichen FFT-KK.

[2] Informationen zur Wavelet-Transformation gibt es in [Gra95] oder unter `http://www.amara.com/IEEEwave/IEEEwavelet.html`.

[3] Im Englischen als „Coarsification", „Downsampling" oder auch „Subsampling" bezeichnet.

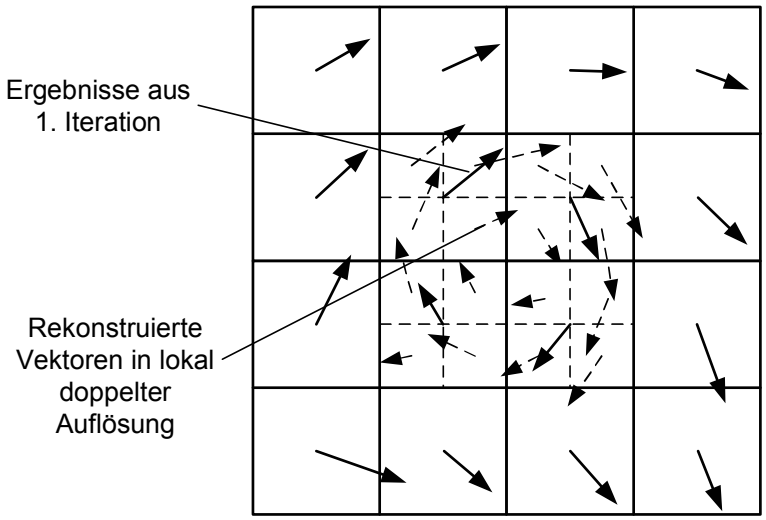

Abbildung 5.1: Eine Zone lokaler, adaptiver Verfeinerung in Abhängigkeit der Wirbelstärke bzw. der Krümmung der Strömungslinien.

Dem Leser empfehle ich daher, diese Idee aufzugreifen und hinsichtlich ihrer Nutzbarkeit weiterzuverfolgen.

5.2 Adaptive Verfeinerung des Abfragerasters

Im Gegensatz zur in Abschnitt 5.1 besprochenen vollständigen Verfeinerung der Vektorfeldauflösung kann man auch den aus vielen Anwendungen bekannten Ansatz der adaptiven Verfeinerung nutzen, um sich Berechnungen und Analysen an denjenigen Stellen zu sparen, wo sich eine lokale Verfeinerung kaum oder nicht mehr auf die Genauigkeit auswirkt. Dies ist z. B. ein gängiges Verfahren bei numerischen Strömungslösern in der Fluidmechanik. In unserem Fall ist eine genauere Betrachtung bei linearen Stromlinien (Typ S aus Abschnitt 3.2) mit nur sehr geringer Wirbelstärke weniger interessant als bei kleinen turbulenten Vortexstrukturen. Realisiert wird die Adaption durch eine regionale Variation der Abfragefensterdichte. Ferner ist auch zu berücksichtigen, dass man bei hohen lokalen Strömungsgeschwindigkeiten zumindest initial größere Abfragefenster benötigt [SR99]. Um alle diese Regionen ausfindig zu machen, sind entsprechende Kriterien notwendig, auf welche nun näher eingegangen wird.

5.2.1 Entscheidungskriterien zur Adaption

Im Folgenden werden zwei Kriterien für die Entscheidung einer adaptiven Verfeinerung der Vektorfeldauflösung bzw. einer adaptiven Abfragefenstergröße besprochen. Das **erste Kriterium** liegt der fundamentalen Strömungseigenschaft *Wirbelstärke* (Rotation) zugrunde, das **Zweite** entstand aus eigenen Überlegungen und nimmt an Stellen mit *Ausreißern* eine lokale Variation einiger Systemparameter vor.

Kriterium 1: Lokale Krümmung der Strömung

[Sca02] schlägt vor allem bei extrem inhomogenen, kompressiblen (Scher-)Strömungen eine adaptive Betrachtung vor. Dazu wird lediglich der lokale Krümmungsradius $r(x,y)$ der Strömung betrachtet, welcher die Geschwindigkeitsveränderung $\Lambda(x,y)$ angemessen beschreibt. Der lokal minimale Krümmungsradius $r_{min}(x,y)$ lässt sich berechnen durch den inversen maximalen Eigenwert[4] der HESSE-Matrix

$$H_u(x,y) = \begin{pmatrix} u_{xx} & u_{xy} \\ u_{yx} & u_{yy} \end{pmatrix} \tag{5.1}$$

mit den zweiten[5], partiellen Ableitungen der ersten Komponente $u(x,y)$ des Geschwindigkeitsvektors $\vec{v}(x,y) = (u(x,y), v(x,y))^T$. Die Ableitungen $u_{\{x|y\}^2}$ lassen sich einfach mittels zentraler oder vorwärts gerichteter, finiter Differenzen 1. Ordnung berechnen. Das Eigenwertproblem liefert dann

$$\lambda_{1,2} = \frac{u_{xx}+u_{yy}}{2} \pm \sqrt{\left(\frac{u_{xx}+u_{yy}}{2}\right)^2 - u_{xy}^2} \tag{5.2}$$

und der minimale Krümmungsradius ergibt sich aus

$$r(x,y) = \frac{1}{\max\{\lambda_1, \lambda_2\}}. \tag{5.3}$$

Die von der Geschwindigkeitsveränderung $\Lambda = 2\pi\sqrt{r}$ und vom Überlappungsfaktor OF abhängige (neue) Abfragefenstergröße N lässt sich durch

$$N = \frac{a+b\Lambda}{1-OF} \equiv \frac{a+2\pi b\sqrt{r}}{1-OF} \tag{5.4}$$

ermitteln, mit z. B. $OF \in \{...,50\%,...,75\%,...\}$. Dabei gilt für die von der Partikelstreudichte und von d_t abhängige minimale Abfragefenstergröße $a \in [8,32]$ und für die modulationsbeeinflussende Konstante $b < 1$. Ich erinnere noch einmal an die Existenz von mindestens 4 Partikelbildpaaren innerhalb eines Abfragefensters.

[Sca02] beschreibt folgenden in seinen Experimenten verwendeten iterativen, adaptiven Algorithmus namens AR-WIDIM:

1. Ein Zufallsgenerator erzeugt eine Anzahl s_0 von beliebigen irregulären Stellen innerhalb einer Aufnahme. Die initiale Fenstergröße ist N_0 und $i = 0$.

2. An den s_i Punkten wird nun eine Korrelationsanalyse vorgenommen.

3. Die so ermittelten Geschwindigkeiten werden mit einer zweidimensionalen Minimum-Quadrat-Methode auf ein reguläres Gitter interpoliert (siehe auch [Che01]).

4. $r(x,y)$ wird dann aus diesen regulären Vektordaten berechnet und Gleichung 5.4 bietet eine Richtlinie zur adaptiven Anpassung der Fenstergröße.

[4] Für eine Wiederholung des speziellen Eigenwertproblems aus der linearen Algebra (Matrizenkalkül) empfiehlt sich z. B. das Werk [BSMM00].

[5] und auch wechselweisen

5. An den Stellen (x,y) des regulären Gitters mit kleinerem $r(x,y)$ (also stärkerer Krümmung) werden nun von neuem mehr Abfragepunkte s_{i+1} zufallsverteilt als an den Stellen mit großem Krümmungsradius $r(x,y)$. Dabei gilt nun $s_{i+1} \gg s_i$ für diese adaptive Verteilung der Abfragepunkte.

6. Nun kann der Algorithmus mit $i = i+1$ und N_{i+1} ab Schritt 2 wiederholt oder bei Erreichung

   - eines Konvergenzkriteriums $\|\vec{v}_{i+1} - \vec{v}_i\|^2 < \varepsilon$,
   - einer bestimmten Anzahl von Iterationen i_{max} oder
   - der minimalen Abfragefenstergröße a

   beendet werden.

Der frühere Ansatz WIDIM wurde mit dem iterativen, adaptiven Ansatz verglichen und erbringt bei genannten Strömungstypen bessere Ergebnisse. Anstatt des Kriteriums 1 könnte auch der in [Wes93] beschriebene Wert für die Wirbelstärke verwendet werden (siehe Abschnitt 3.1).

PIVA verwendet zwar bereits das Kernkriterium in Gleichung 5.3 zur lokalen regulären Verfeinerung, verfolgt aber nicht den oben beschriebenen, zufallsbasierten Ansatz auf irregulären Gittern. Das System lässt sich aber um einen adaptiven Ansatz erweitern. Die Grundfunktionalität bzgl. der Kreuzkorrelation bleibt bestehen. Lediglich das System zur Ermittlung der Vektorfelder muss geringfügig verändert oder erweitert werden. Nähere Informationen gibt es später im Abschnitt 7. Momentan unterstützt *PIVA* lediglich eine ganzheitliche Verfeinerung des Vektorfeldes.

Kriterium 2: Lokale Detektion von Ausreißern

[Idee] Ein weiteres Kriterium, eine Anpassung der lokalen Abfrage- bzw. Suchfenstergröße vorzunehmen, wäre z. B. an Stellen, wo *Ausreißer* gefunden werden. Ausreißer treten u. a. mit sehr hoher Wahrscheinlichkeit dort auf, wo das Verhältnis der Suchfenstergröße zur Verschiebung zu klein wird (siehe Abschnitt 2.1.2). An solchen Stellen erhält man folglich schlechte Korrelationssignale ohne eindeutig interpretierbare Spitze. Eine Adaption bestünde darin, das Suchfenster hier entscheidend größer zu machen und/oder bei Annahme einer größeren, lokalen Wirbelstärke[6] die Vektorfeldauflösung/Abfragefenstergröße zusätzlich zu verkleinern, um innerhalb eines Abfragefensters homogene Strömungsverhältnisse zu erreichen. Als Entscheidungshilfe müsste man benachbarte, bereits bestehende Vektoren verwenden, um damit die Wirbelstärke bzw. die lokale Durchschnittsgeschwindigkeit zu berechnen. Anstatt des gefundenen Ausreißers kann nun zur Berechnung je nach Bedarf z. B. ein bilinear interpolierter Ersatz genommen werden.

Ferner könnte man eine durch das **Kriterium 1** bedingte Vererbung von Schätzervektoren in feinere Ebenen vornehmen. Bei zu kleinem Krümmungsradius unterlässt man die Vererbung und vergrößert an dieser Stelle das Suchfenster, um in feineren Auflösungsstufen neue, bessere Ergebnisse zu erzielen.

Es wäre interessant, in dieser Richtung weitere Nachforschungen und Überlegungen anzustellen. Leider kann im Rahmen dieser Diplomarbeit nicht weiter in die Tiefe gegangen werden. *PIVA* implementiert diesen Ansatz nicht.

[6] Dabei handelt es sich um kleine wirbelige Strömungsstrukturen.

5.3 Probleme und Überlegungen bei multiresoluten Lösungen

Bei hierarchischen Schemata in der PIV zur Näherung an eine möglichst genaue Lösung sollte man einiges bedenken. Abhängig von der Art der PIV-Aufnahmen und den Systemparametern kann es entsprechend viel oder wenig Sinn machen auf vergröberten Bildern zu arbeiten. In vergröberten Aufnahmen sollten Abfragefenster nicht vergrößert werden. Ich erinnere hier an die Annahme, dass die Strömungsverhältnisse innerhalb eines Abfrage- bzw. Suchfensters möglichst homogen sein sollten, um initiale Schätzer gut zu bestimmen und die Rate der Ausreißer gering zu halten (siehe [Wes93]).

Bei stark wirbeligen, inkompressiblen und/oder turbulenten Strömungen mit kleinen Strömungsmerkmalen macht es von vornherein kaum Sinn, große Abfragefenster zu verwenden. Der Vergröberungsansatz von [RWK98] dürfte für derartige Strömungen recht problematisch sein. In jedem Fall bleibt es dem Leser überlassen, sich in einschlägige Lektüre für die PIV-Analyse spezieller Strömungsszenarien einzulesen.

Aussage 5.1 *PIV-Analysen sind aber in vielen Fällen mit einer intensiven, empirischen Heransgehensweise zur Ermittlung idealer Parameter für eine individuelle Beobachtung verbunden.*

[Idee] Eine weitere Idee basierend auf einer Kombination der beiden Kriterien aus Abschnitt 5.2.1 und der Problembeschreibung aus Kapitel 5.3 wäre eine **Adaption der Abfrage- und Suchfenstergrößen in beide Richtungen**. Auf der Grundlage des Vorwissens der lokalen Geschwindigkeitsverteilung könnte man nach Gleichung 5.4 und der Richtlinie $\|\vec{r}\| < N/4$ aus [KA90] bzw. [KA92] ein größeres Suchfenster einsetzen, um lokal den Bereich der erfassbaren Geschwindigkeiten zu erhöhen. Zusätzlich könnte man in diesen Regionen die Vektorfeldauflösung verfeinern. Über mehrere Iterationen innerhalb einer Verfeinerungsebene kann das Suchfenster mit der iterativen Fensterverschiebetechnik wieder verkleinert und so die Genauigkeit wieder erhöht werden. Dem Leser sei es überlassen, diesen Ansatz weiter zu verfolgen.

Zu große Geschwindigkeitsgradienten können demnach auch Probleme bereiten. Einen Artikel, der sich mit der Optimierung von PIV-Methoden für Strömungen mit hohen regionalen Geschwindigkeitsunterschieden und -bereichen befasst, findet man in [Bol00].

Probleme mit zu großen Suchfenstern

Die Größe des Suchfensters ist je nach erwartetem Bereich der Geschwindigkeiten und der Art der Bilder **nicht zu groß** zu wählen. Bei künstlichen Bildern hat man eindeutige Merkmale (Partikel). Beispielsweise biologische Bilder beinhalten oft Muster, die sich in ähnlicher Weise örtlich wiederholen, sodass bei kleiner Vektorfeldauflösung und sehr kleinen Geschwindigkeiten ein zu großes Suchfenster falsche Werte liefern kann. Die richtige Wahl der Suchfenstergrößen und damit die optimale Ermittlung der Schätzervektoren in einer ersten Rekonstruktionsiteration bilden eine sehr wichtige Grundlage für folgende Iterationen mit der „Window Offset"-Methode (siehe Kapitel 2.1.3). Von ihr hängt es dann ab, ob in den folgenden Schritten überhaupt ein hinreichend korrektes Geschwindigkeitsfeld rekonstruiert werden kann.

- **[Idee]** Ein Ansatz bestünde darin, in einem „rohen" ersten Durchlauf über das gesamte Bild die Bandbreite W der Geschwindigkeiten zu ermitteln, wenn W nicht schon vorher

aus anderen Informationsquellen bekannt ist. Mit dem Wert W kann das anfängliche Verhältnis zwischen Such- und Abfragefenster günstig festlegt werden.

- **[Idee]** Eine weitere Möglichkeit zur Ermittlung einer initialen Abfrage- und Suchfenstergröße basierend auf der Geschwindigkeitsverteilung wäre z. B. eine schnelle PIV-Analyse mit der FFT-KK in zwei Durchläufen mit zwei verschiedenen (empirischen) Abfrage- und Suchfenstergrößen $N_{1,2}$ und $N_{S_{1,2}}$. Das Paar $(N_?, N_{S_?})$ mit weniger Ausreißern im Ergebnisvektorfeld kann dann als $\vec{N}_{init} = (N_?, N_{S_?})$ für weitere Iterationen verwendet werden, mit $? \in \{1, 2\}$.

PIVA erhält empirisch ermittelte Anfangswerte für Abfrage- und Suchfenstergröße über Kommandozeilenparameter und verfolgt diese beiden Ideen noch nicht.

6 Betrachtung frequenzbasierter Ansätze zur Merkmalsextraktion

Bis dato werden in der PIV leistungsfähige, digitale Filtermethoden zur Behandlung von qualitativ schlechten, verrauschten Aufnahmen kaum verwendet

---

Um die Methoden der PIV-Analyse auch auf weit vom Standard abweichende Aufnahmen gewinnbringend anwenden zu können, kann man Ansätze der Merkmalsextraktion, einem wichtigen Bereich der digitalen Bildverarbeitung, einfließen lassen. Der folgende Abschnitt ist im Kontext dieser Diplomarbeit Gegenstand der Forschung und enthält hauptsächlich Gedankengänge, welche es noch gilt, mathematisch nachzuvollziehen. Die hier behandelten Teilabschnitte können auch als theoretische Hintergedanken für die entsprechenden Tests im späteren Abschnitt 8 interpretiert werden.

6.1 Hoch- und Tiefpassfilter zur Extraktion von Merkmalen

Grundsätzliche Annahme ist hier, dass eine geeignete Aufnahme entsprechende Merkmale enthält, die sich verfolgen lassen. I. d. R. unterscheiden sich diese Merkmale signifikant vom sogenannten **Rauschen**, welches in vielen Bildern auftritt und von diversen Faktoren wie z. B. der verwendeten PIV-Hardware und des gesamten Aufnahmesystems abhängig ist.

Man sieht das Rauschen als Zufallssignal und folglich manifestiert sich dieses in den beiden, betrachteten Aufnahmen verschieden. Daraus leite ich a priori einen negativen Einfluss auf das Kreuzkorrelationssignal her, was ebenso als Motivation für diesen Abschnitt gilt. Die dazu nötigen Überlegungen werden wir nun behandeln. Die Aufgabenstellung für die späteren Tests in Abschnitt 8 besteht darin, diesen negativen Einfluss nachzuweisen.

Eine weitere Annahme ist, dass wir hier ein über das gesamte Frequenzband der Aufnahmen verteiltes „weißes“ Rauschen vorfinden (siehe [Kun02] für weitere Details).

Nun wird versucht, das Bildsignal mit zwei verschiedenen Methoden zu zerlegen, um Rauschen und Bildmerkmale voneinander gewinnbringend zu trennen:

Ansatz A — Im diesem ersten, einfacheren Ansatz verwenden wir den in Teilkapitel 4.2 beschriebenen GAUSS'schen Tiefpass- oder Unschärfefilter und den in Teilabschnitt 4.3 erklärten SOBEL-Operator als Hochpass- oder Kantenfilter. Dieser Vorgang wird mehrmals wiederholt, sodass eine Hierarchie von hochpass- und tiefpassgefilterten Bildern entsteht, wie sie in Abbildung 6.2 gezeigt wird. Dieser Operator ist kein Hochpassfilter im eigentlichen Sinne, da er lediglich auf starke Helligkeitsgradienten und damit auf Anteile hoher Frequenzen besonders reagiert. Diese intuitive Bezeichnung wollen wir dennoch weiterverwenden. Es sei am Rande erwähnt, dass als Alternativen für

den SOBEL-Operator als Hochpassfilter auch der „ROBERT'S-Cross"-Operator oder der „PREWITT-Gradient-Edge"-Detektor verwendet werden könnten.

Ansatz B — Aus den Tests in Abschnitt 8 wird ersichtlich, dass sich die Ergebnisse nicht in dem gewünschten Maße verbessern. Deshalb wurden in einem zweiten Anlauf die Hoch- und Tiefpassfilter durch ein *Wavelet*-basiertes Kantendetektionsverfahren abgeleitet vom CANNY-Operator ersetzt, welchen wir von nun an als „Wavelet"-Operator bezeichnen. Theoretische Details können in Abschnitt 4.3.2 nachgelesen werden. Testergebnisse dazu sind in Kapitel 8.3 einsehbar. Die Hierarchie aus Abbildung 6.2 bleibt jedoch dieselbe.

Merkmale befinden sich meist in einem höheren Frequenzbereich und bleiben somit in den hochpassgefilterten Bildern — naheliegenderweise bezeichnet als „Kantenbilder" K_i — erhalten. Nachteilig in den Kantenbildern ist das zu einem je nach Detektionsoperator vorhanden bleibende **Rauschen**. Der SOBEL-Operator an sich hat mit seinem 3×3 Kern bereits eine glättende Wirkung zur Verminderung des Rauschens. Der Wavelet-Operator bewirkt dasselbe mit seiner Skalierbarkeit auf verschiedenen Ebenen in flexiblerer Weise.

6.1.1 Das Konzept der Filterhierarchie

Hinzuzufügen ist zum Ansatz A, dass sich die Frequenzbereiche des GAUSS-Filters und des SOBEL-Operators überschneiden. In den tiefpassgefilterten Bildern T_i bleiben also die wichtigen Merkmale erhalten, das Rauschen im hohen Frequenzbereich wird jedoch gefiltert und verkümmert somit zu einem sogenannten farbigen oder „rosa" Rauschen. Eine erneute Anwendung des SOBEL- bzw. des Wavelet-Operators auf einem tiefpassgefilterten Bild T_{i+1} liefert die hochpassgefilterten Merkmale K_{i+1} jedoch dieses Mal mit geringerem Rauschen im hohen Frequenzbereich. Mehrmalige Anwendung erzeugt die in Abbildung 6.2 gezeigte Hierarchie der Tiefe N mit $0 \leq i \leq N$. Der **Vorteil von Ansatz B** liegt darin, dass neben dem Tiefpassfilter auch der Hochpassfilter entsprechend skaliert wird, was zu einer besseren Detektion der Lage und der Form der Kanten führt. Das Prinzip der kombinierten Filterung (Hochpass- und Tiefpassfilter) aus Ansatz A bleibt im Ansatz B erhalten.

Nun kann man die Kantenbilder K_i mit den Merkmalen für eine D-NKK-Analyse verwenden. Im Abschnitt 8.3 sieht man entsprechende Ergebnisse bei Aufnahmen biologischer Strukturen mit beiden Kantenfiltervarianten.

6.2 Entfernung des Rauschens zur Verbesserung der Korrelation

Die Kantenbildhierarchie K_i aus Abbildung 6.2 kann im Falle von Ansatz A dazu verwendet werden, das durch den SOBEL-Operator zwar abgeschwächte, jedoch noch existierende „weiße" Rauschen im ersten Kantenbild K_0 zu eliminieren.

Folgender Gedanke abgeleitet aus [MZ92] bildet die Grundlage dieses Ansatzes:

Aussage 6.1 *Merkmale bleiben in allen Kantenbildern K_i mit abnehmender Intensität erhalten, Rauschspitzen jedoch fallen mit zunehmend tieferer Filterebene i weg bzw. verändern ihre örtlichen Positionen zufällig. Vorbedingung ist die Verwendung einer bestimmten Klasse von Wavelets, hier ein entsprechender Kantendetektionsoperator wie z. B. der modifizierte* CANNY*-Filter aus Abschnitt 4.3.2 (Ansatz B).*

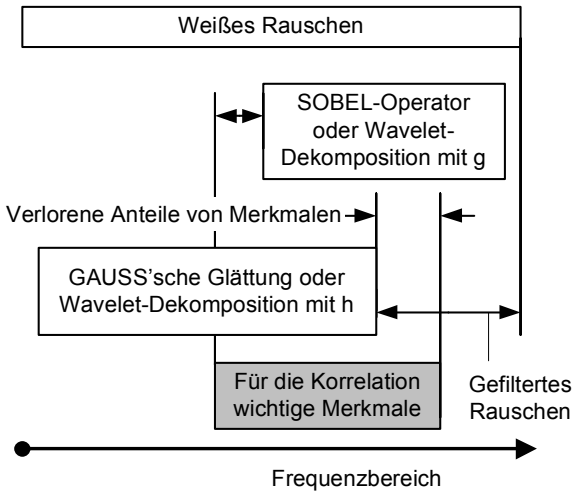

Abbildung 6.1: Die grobe, schematische Einteilung des Frequenzbereichs und die Lage des Wirkungsbereichs der eingesetzten Filter aus Ansatz A und B.

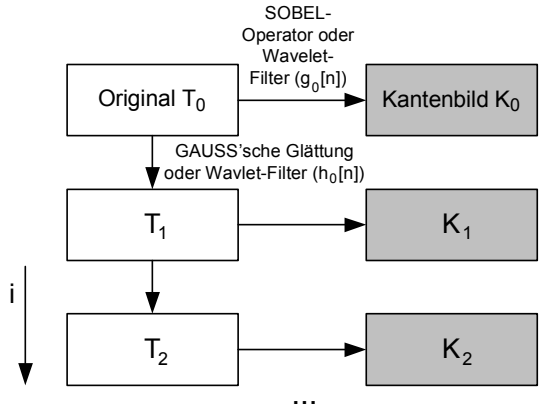

Abbildung 6.2: Eine Hierarchie von hoch- und tiefpassgefilterten Versionen (T_i bzw. K_i) einer Originalaufnahme mit den Filtern aus Ansatz A oder Ansatz B.

Ein Vergleich der in der ersten Ebene K_0 gefundenen Spitzen mit den darunterliegenden Ebenen ermöglicht also eine Elimination von in tieferen Ebenen nicht korrespondierenden Spitzen (siehe Abschnitt 6.3).

Das Vorgehen entspricht folgendem Algorithmus, welcher auch in *PIVA* implementiert wird:

1. Setze $i = 0$.

2. $\forall$ Pixel $K_0(x,y)$: Prüfe, ob es ein lokales Maximum der Art

$$\max_{lokal}(i,x,y) = \max\{K_i(x-1,y), K_i(x,y-1), K_i(x,y), K_i(x+1,y), K_i(x,y+1)\} \tag{6.1}$$

   gibt.

3. Wenn $\max_{lokal}(0,x,y) = K_0(x,y)$,

   a) setze $i = 1$ und

   b) suche Korrespondenten $\max_{lokal}(i,x,y)$ der Spitze $K_0(x,y)$ in allen $i \times i$-Umgebungen $U(i,x,y) = \{K_i(x+m,y+n) | -i \leq m,n \leq i\}$ der Hierarchie $\forall i \in [1,N]$.

   c) Wenn in keiner Ebene K_i mit $i > 0$ ein entsprechendes lokales Maximum gefunden wird, dann ersetze die nun als Rauschen deklarierte Spitze $K_0(x,y)$ durch

$$\begin{aligned} K_{0_{neu}}(x,y) &= \frac{\tilde{K}_{0_{neu}}(x,y)}{2} \quad \text{mit} \\ \tilde{K}_{0_{neu}}(x,y) &= \max\{\max_{-1\leq i\leq 1}\{K_0(x+i,y-1)\} + \max_{-1\leq i\leq 1}\{K_0(x+i,y+1)\}, \\ &\qquad \max_{-1\leq i\leq 1}\{K_0(x-1,y+i)\} + \max_{-1\leq i\leq 1}\{K_0(x+1,y+i)\}\}. \end{aligned} \tag{6.2}$$

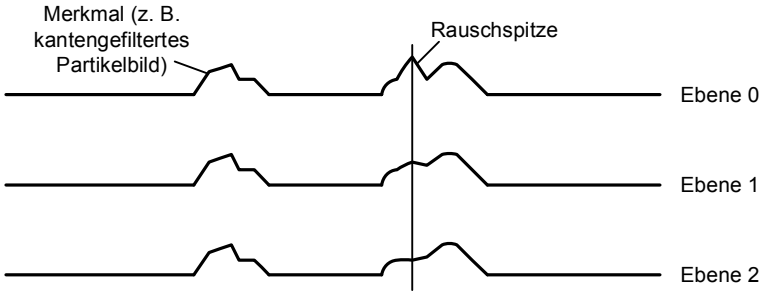

Abbildung 6.3: Die schematische Erkennung von Rauschspitzen in den verschiedenen Ebenen i der Kantenbilder K_i. Die im Originalbild gefundenen Kanten im Kantenbild K_0 werden hinsichtlich nicht-hierarchisch auftretender Rauschspitzen untersucht, um die Korrelationsebene somit rauschfreier zu gestalten.

Gleichung 6.2 stellt eine strukturerhaltende Form der bilinearen Interpolation dar und wirkt glättend auf das Bild K_0. Nach dieser Prozedur, welche im schlimmsten Falle $o(B \cdot H \cdot (N^3 + N^2))$ Schritte benötigt, mit den Bildabmessungen $B \times H$ und i. d. R. $N \in [2,5]$, erhält man ein leicht geglättetes Bild K_0 mit vermindertem Rauschen. Ein evaluierender Test kann in Abschnitt 8 eingesehen werden.

Dieser Algorithmus wird nur zusammen mit dem Ansatz A verwendet. Ansatz B verwirklicht das hier Erklärte mit dem korrelationsbasierten „Threshold“ von [Chr04], beschrieben ab Seite 45.

6.3 Effizienzsteigernde, merkmalserhaltende Vergröberung von Bildern

Wie schon in Kapitel 5.1.1 beschrieben, kann durch eine Vergröberung der Bildauflösung eine effizientere Korrelationsanalyse erreicht werden, da ja die Anzahl der zu betrachtenden Pixel auf einen Bruchteil reduziert wird.

Die GAUSS'sche Pyramide erzeugt einerseits durch den Glättungsoperator und andererseits durch die Halbierung der Anzahl der diskreten Werte tiefpassgefilterte Bilder. Dem NYQUIST-SHANNON'schen Abtasttheorem zufolge entfernt man mit der Verdopplung des Abtastintervalls Δd_i (Bildauflösung der Vergröberungsstufe i) die Frequenzen (u. a. Merkmale) mit einer Wellenlänge $l < 2\,\Delta d_{i+1}$ mit $\Delta d_{i+1} = 2\,\Delta d_i$ aus dem ursprünglichen Signal.

Wichtig ist hierbei also, dass man die Annahme treffen kann, dass in den betrachteten Bildern Merkmale entsprechend tiefer Frequenzen vorhanden sind. Ferner müssen diese Merkmale direkt mit der Strömung gekoppelt sein.

Unter diesen Vorbedingungen ist es dann möglich, Bilder entsprechend zu vergröbern. Wir haben diesbzgl. schon in Abschnitt 5.1.1 den durchschnittlichen Partikelbilddurchmesser als Richtlinie erwähnt. Zu diesem Thema finden sich in Abschnitt 8 ebenfalls einige Testergebnisse.

Die GAUSS'sche Pyramide kann mit einem Konvolutionskern der Größe $N \times N$ erzeugt werden [BA83]. Der 5×5-Kern benötigt mehr Rechenzeit als der 3×3-Kern (auch genannt „Triangular“-Kern). In der PIV-Analyse erscheint ein 3×3-Kern besser geeignet, da er einen geringeren Unschärfeeffekt als der 5×5-Kern besitzt und Partikelbilder somit besser erhal-

ten bleiben.

Anstatt einer GAUSS'schen Glocke zur Interpolation der Pixel im Vergröberungsschritt können auch einfachere Methoden angewendet werden. Der arithmetische Mittelwert von jeweils vier Pixeln könnte verwendet werden. Dieser schnelle Ansatz hat den Nachteil einer signifikanten Verzerrung des Signals.

[Idee] Ein allgemeinerer Vorschlag wäre die Verwendung einer flexiblen Wavelet-Transformation. Im Hochfrequenzbereich würden über mehrere Ebenen laut Aussage 6.1 keine Merkmale (Spitzeninformationen) verloren gehen. Das Rauschen jedoch vermindere sich durch die wechselweise, hierarchische Filterung in zwei Frequenzbändern. Ansatz B aus Kapitel 6.1 verfolgt diesen Gedanken jedoch bisher ohne die Halbierung der Abtastrate zwischen den hierarchischen Skalen, und entsprechende Tests werden in Abschnitt 8 gezeigt.

An dieser Stelle möchte ich darauf hinweisen, dass es vorteilhaft wäre, tiefer in das Fachgebiet der *Digitalen Filter* einzutauchen, um auf dessen Grundlagen Ansätze aus der Bildverarbeitung gezielt anwenden zu können bzw. die Verarbeitung von Bildsignalen abzielend auf die PIV-Analyse zu optimieren. Als Einstiegsreferenz kann ich dazu das Buch von [Göt98] angeben.

6.4 Eine Kurzübersicht über die Verfahrenskomplexitäten

Dieses Teilkapitel dient lediglich als kurzer Überblick zum Vergleich der Komplexitäten einiger, in den letzten Abschnitten besprochenen Verfahrensvarianten und Ideen und enthält sich einer besonderen Informationstiefe. Ferner sei darauf hingewiesen, dass Komplexitäts- bzw. Speicherplatzangaben natürlich vom verwendeten Algorithmus bzw. von der Implementierung abhängen. Abweichungen von diesen als grobe Richtlinien geltenden Angaben und bessere Komplexitäten durch andere Verfahrenskombinationen bzw. geschicktere Implementierung sind durchaus möglich. Der Artikel [MM02] befasst sich detailliert mit der Laufzeitanalyse einiger, gängiger Methoden aus dem Fachgebiet der PIV.

In der Tabelle 6.1 findet man die Kurzbeschreibung der Ansätze zusammen mit der Angabe des relativ zur einfachen D-KK zusätzlichen Speicherbedarfs, der Laufzeitkomplexität des verwendeten Algorithmus und einer Referenz auf das beschreibende Kapitel. Die Abkürzungen für die Verfahrensbegriffe sind im Glossar auf Seite 115 nachzulesen.

Ansatz	**Zusätzliche Speicherkomplexität**	**Laufzeitkomplexität**	**Referenz**
Einfache D-KK	keine	$O(N^4)$	1.3.1
Einfache D-NKK	keine	$O(N^4)$ (jedoch mehr Rechenaufwand als bei der D-KK!)	1.3.1
Spezielle D-NKK	keine	D-NKK $\cdot O(N^2)$	2.1.7
Schnelle, spezielle D-NKK	$O(N_s^2)$ Platz für zwei Integraltabellen	$O(N_s^2 + N^4)$ ohne die mehrmalige Berechnung der variablen Mittelwerte und Normalisierungsterme	2.2.2
FFT-KK	keine	$O(N^2 + N^2 \log_2 N^2)$	1.3.2
Größeres Suchfenster	keine	$O(N^2 \cdot N_s^2)$	2.1.2
PID	$O(N_s^2)$ Platz für Suchfenster	D-KK[a] $+O(N_s^2)$ zur Berechnung des verzerrten Musters	2.1.3
CWS	$O(N_s^2)$ Platz für Suchfenster	D-KK $+O(N_s^2)$ zur Berechnung der interpolierten Bilddaten	2.1.4
CDIC	$O(N_s^2)$ Platz für Suchfenster	D-KK $+O(N_s^2)$ zur Berechnung der interpolierten Bilddaten	2.3.3
Iterative Verfahren	keine	D-KK $\cdot O(i)$ i ... Anzahl der Iterationen	2.1.6
Adaptive Verfahren	keine	$O(\sum_{n=1}^{a} N_n^4)$ im Falle der einfachen D-KK, a ... Anzahl der adaptiven Verfeinerungen	5.2
Filtern von Bildern	$O(e \cdot B \cdot H)$ für gefilterte Bilder (-hierarchien) e ... Anzahl der Filterebenen	D-KK $+O(e \cdot B \cdot H \cdot F_B \cdot F_H)$ für die Filter- und Faltungsoperationen F_B, F_H ... Abmessungen des Filterkernes	4.3

[a] An allen folgenden Stellen natürlich ersetzbar durch die D-NKK oder die FFT-KK.

Tabelle 6.1: Komplexität der PIV-Verfahrensvarianten

Teil II

Praxis

7 Beschreibung der Implementierung

Zur Evaluation der DPIV-Verfahren bedarf es eines flexiblen Softwaresystems zur parametrisierbaren Analyse von Bildern und Bildsequenzen

Dieser Abschnitt befasst sich eingehend mit dem praktischen Teil dieser Diplomarbeit. In der Hauptsache bestand dieser aus der Entwicklung und Implementierung eines flexiblen Datenverarbeitungssystems zur Validierung und Evaluierung der im theoretischen Teil erklärten Methoden und Ansätze.

Ich möchte jedoch vorwegnehmen, dass ich hier nicht auf Details der Programmierung eingehen werde. Der Leser findet einen oberflächlichen Einstieg in die Systemarchitektur und die Zusammenarbeit der Kernkomponenten, um sich in kurzer Zeit in die Materie einarbeiten zu können. Die nötigen Informationen zur Weiterentwicklung des Systems findet man in der Entwicklerdokumentation von *PIVA*; siehe [Gle04].

Das System wurde mit Hilfe von Methoden aus dem Bereich des Software Engineering entwickelt. Dies wäre unter Berücksichtigung des Systemumfangs nicht unbedingt notwendig gewesen, aber im Hinblick auf spätere Erweiterbarkeit können wertvolle Vorteile daraus gezogen werden. An dieser Stelle möchte ich auf [BD04] verweisen, welcher einen modernen Einstieg in die Welt der objektorientierten Softwareentwicklung bietet. Eine Einführung in die Modellierungssprache UML[1] findet man dort ebenso.

Als Referenzsystem im Bezug auf die PIV-Funktionalität galt das Programm *PIV View*[2] der *PivTec GmbH*, welches eine große Zahl von Einstellungen zulässt, um Analyseergebnisse zu optimieren. Im Gegensatz zu *PIVA*, welches schwerpunktmäßig zum Testen neuer Verfahren entwickelt worden ist, findet man mit *PIV View* ein kommerzielles, benutzerfreundliches System entwickelt für den industriellen Einsatz.

7.1 Definition der Problemstellung

Dieses Kapitel beschreibt den Schritt „*problem statement*“ im Softwareentwicklungsprozess.

Die Aufgabe bestand in der Implementierung eines PIV-Analysewerkzeugs zur Rekonstruktion von Vektordaten bzw. Geschwindigkeitsdaten aus digitalisierten Aufnahmen von partikelbestreuten Strömungen. Die Analysemethode soll auf der D-KK basieren — viele Bildvergleichstechniken basieren auf der statistischen Korrelation. Die im theoretischen Teil besprochenen Methoden und Ansätze zur Verbesserung der Genauigkeit und der Verarbeitungsgeschwindigkeit sollen, soweit wie möglich, kombinier-, ein- und/oder ausschaltbar sein. Ein hierarchischer, adaptiver Ansatz zur Erhöhung der Vektorfelddichte soll ebenfalls Teil der Architektur werden. Zum System gehört ferner eine Schnittstelle zur Konfiguration

[1] Die „Unified Modeling Language“; die UML 2.0 Spezifikation ist unter `http://www.uml.org` erhältlich.
[2] Siehe `http://www.pivtec.com/pivview.html`.

und Bedienung. Ziel des Systems ist die Evaluation der Ausgabedaten bzgl. der Qualität. Echtzeitanforderungen werden nicht gestellt, sodass auf eine optimale Datenverarbeitungsgeschwindigkeit nur bedingt geachtet werden muss.

7.2 Anforderungsanalyse

Dieses Kapitel beschreibt in etwa den Schritt „*requirements elicitation*“ im Softwareentwicklungsprozess nach [BD04].

Nachdem man das prinzipielle Problem formell definiert hat, kann man in einer gezielten Analyse die für die Lösung nötigen Anforderungen ableiten. Für umfangreichere Softwareentwicklungsprojekte würde sich dieses Kapitel in weitaus mehr Abschnitte gliedern, die sich laut einem bestimmten Vorgehensmodell mit einer feinstrukturierten Problemanalyse und -lösung befassen. Da viele der generischen Analyseschritte aufgrund des geringen Umfangs des Projekts keinen besonderen Wissensgewinn erbracht haben, d. h. auf das vorliegende Problem nicht sinnvoll anwendbar waren, wurden diese hier ausgelassen.

7.2.1 Szenarienanalyse und Definition von Anwendungsfällen

Als Benutzer des Systems kommen lediglich solche in Betracht, welche die implementierten PIV-Analysemethoden testen wollen.

Als grundsätzliche Aufgaben werden angegeben:

- Das Laden einer oder mehrerer Aufnahmen in einem einfachen Grafikformat[3],
- die Übergabe von Systemparametern, siehe Abschnitt 7.2.4,
- die Ausgabe von Vektordaten in einem einfachen Format zur Eingabe in ein Visualisierungssystem.
- die Ausgabe von Bilddaten zur Visualisierung der Hierarchie der hochpassgefilterten Bilder.

Als Grundlage für den Aufbau der Systemarchitektur kann man folgenden Anwendungsfall mit dem Ziel der Vektordatenrekonstruktion voraussetzen:

1. Der Benutzer übergibt dem System $N \geq 2$ einzelbelichtete PIV-Aufnahmen.
2. Nun spezifiziert er die passenden Systemparameter, um die Art der Analyse und Verarbeitung festzulegen. Als einfach zu implementierende Schnittstelle wird die Kommandozeile gewählt.
3. Das System erzeugt die entsprechenden Vektordaten (siehe Kapitel 7.2.2) und die zugrundeliegenden, gefilterten Bilddaten.

[3] Hier wurde das Bitmap-Format (BMP oder DIB) von Microsoft Windows gewählt. Eine Spezifikation des Formats findet man z. B. unter `http://www.audre.org/GraphicFAQ/`.

4. Der Benutzer nimmt diese Ausgabedaten, übergibt sie einem Visualisierungssystem[4], welches diese Informationen interpretieren kann. Die Ausgabe besteht aus $N-1$ Vektorfeldern. Als Vektorvisualisierungssystem dient eine am Lehrstuhl entwickelte Software spezialisiert auf die interaktive Darstellung von Vektorfeldern. Ersatzweise kann auch das flexible Werkzeug *gnuplot* (siehe [WK04]) verwendet werden.

5. Nun stehen dem Benutzer folgende Möglichkeiten zur Verfügung:
   - Vergleich der Vektordaten mit einem Referenzvektorfeld durch Subtraktion,
   - Ermittlung von statistischen Daten aus einem Differenzvektorfeld,
   - Skalierung des Vektorfeldes zur besseren Visualisierung,
   - Nachbearbeitung des Vektorfeldes hinsichtlich vorkommender Ausreißer.

6. Abschließend erhält der Benutzer eine Visualisierung der beobachteten Strömungsverhältnisse in Form von zweidimensionalen Vektorfelddaten.

7.2.2 Schnittstellenspezifikation — Das Ausgabeformat

Als Schnittstelle zwischen *PIVA* und anderen Anwendungen dient die Ausgabe des Systems in Form eines sehr einfachen Rohtextformates, welches zur Haltung von Vektorfelddaten verwendet werden kann. In bekannter EBNF-Schreibweise[5] wird dieses Format folgendermaßen definiert:

```
NL          = "\n"
EOF         = ""
VEKTOR      = X_KOMPONENTE "," Y_KOMPONENTE
POSITION    = X_POSITION "," Y_POSITION
FELDELEMENT = POSITION ";" VEKTOR NL
VEKTORFELD  = BILDDIM_X NL
              BILDDIM_Y NL
              {FELDELEMENT}
              EOF
```

`BILDDIM_X`, `BILDDIM_Y`, `X_POSITION` und `Y_POSITION` sind Nonterminale vom Typ `int` und `X_KOMPONENTE` bzw. `Y_KOMPONENTE` vom Typ `double`. Die zeilenweise Ordnung der `FELDELEMENT`e ist beliebig. Werden zwei Vektorfelder V_1 und V_2 verglichen bzw. wird ein Differenzvektorfeld berechnet, muss für jeden Vektor $\vec{v}_1(x,y) \in V_1$ genau ein entsprechender Vektor $\vec{v}_2(x,y) \in V_2$ an deckungsgleicher Position (x,y) existieren. Die Vektoren werden, wenn nötig, programmintern nach ihrer Position sortiert, sodass deren Reihenfolge in der Datei irrelevant ist.

7.2.3 Festlegung nichtfunktioneller Anforderungen

Eine Aufstellung von nichtfunktionellen Anforderungen war nötig, um die Flexibilität des Systems im Voraus abschätzen zu können.

- Die Bedienung soll ausschließlich über die Kommandozeile erfolgen und trotzdem noch möglichst einfach sein.

[4] Ein System zur Darstellung von Vektordaten und/oder ein Bildverarbeitungssystem wie z. B. *GIMP* unter http://www.gimp.org.

[5] Erweiterte BACKUS-NAUR-Form (EBNF), siehe [Rec02].

- Als Bilddaten sind Aufnahmen im 8-Bit-Graustufen BMP-Format erlaubt mit beliebigen Bildabmessungen.
- Die Ausgabe von Bilddaten soll im formlosen RAW-Format erfolgen.
- Die Überlappung von Abfragefenstern und die hierarchische Rekonstruktion (siehe Abschnitt 5) sollen kombiniert werden können.
- Das System wird in C++ entwickelt, ist portabel zu gestalten und wird daher soweit möglich nur die auf allen Systemen verfügbaren Standardbibliotheken (C/C++ Standard Bibliothek inkl. STL-Erweiterung) nutzen[6]. Deshalb wird z. B. keine Nebenläufigkeit bzw. Grafikausgabe integriert.
- Die Software wird mit Hilfe des Quelltextdokumentationssystems *doxygen* dokumentiert, um anderen Entwicklern den Einstieg zu erleichtern; siehe [Hee04].
- Das System erhält eine Ausnahmebehandlung, welche Fehler in den Eingabeparametern abfängt und dem Benutzer entsprechende Fehlerstatusmeldungen liefert.
- Es bestehen keine Echtzeitanforderungen an das System.
- Ein Vergleich mit den Referenzdaten aus [ONSK00] soll gute Ergebnisse hervorbringen.
- Das Programm wird aus erweiterbaren und austauschbaren Komponenten zusammengesetzt. Die Systemarchitektur soll entsprechend flexible Eigenschaften aufweisen. Auf Wiederverwendbarkeit wird Wert gelegt, dabei hilft vor allem das Konzept von generischen Quelltextteilen (z. B. C++-Templates und Vererbung).
- Auf Sparsamkeit hinsichtlich des Speicherplatzverbrauchs soll geachtet werden. Dies ist aber nur Nebensache, da die Datenmengen in Relation zu den heutigen Hauptspeichergrößen gering sind.

Diese Anforderungen führen zu einem System, wie es in Kapitel 7.3 beschrieben wird.

7.2.4 PIVA Anwenderdokumentation

Dieser Abschnitt befasst sich eingehend mit der Benutzung der Kommandozeile und mit dem Einfluss der dort möglichen Parameter auf das System. Es sei hier erwähnt, dass es Teile in *PIVA* gibt, welche aufgrund unangebrachter Zweckmäßigkeit noch nicht implementiert wurden und für diese Diplomarbeit weniger interessante Parameter daher noch nicht ihre vollständige Funktionalität besitzen. Die *PIVA*-Kommandozeile (Synopsis) zur Vektordatenrekonstruktion gestaltet sich wie folgt:

```
piva [-a] [-b beta] [-B scales] [-C accuracy] [-d density] [-D depth]
     [-e] [-f outbase] [-g] [-G] [-{h|?}] [-H steps] [-i iterations]
     [-I size] [-l scalefactor] [-m maskfile] [-M magnification] [-n]
     [-N] [-o overlap] [-O outbase] [-p interval] [-P method] [-r size]
     [-s method] [-S size] [-t type] [-T] [-v] [-V] [-w type] [-Y]
     aufnahme_1.bmp aufnahme_2.bmp [aufnahme_3.bmp [...]]
```

[6] Eine umfassende ANSI-C/C++-Referenz findet man im Werk von [Lou00].

Der Aufruf von *PIVA* zur Berechnung eines Differenzvektorfeldes sieht folgendermaßen aus:

```
piva [-{VT}] [-t type] -O outbase
     -u vectorfelddatei_1.dat vectorfelddatei_2.dat
```

Ein Vektorfeld kann mit der nun angegebenen Kommandozeile skaliert werden:

```
piva [-{VT}] [-t type] [-l scalefactor] [-R thres] -O outbase
     -k vectorfelddatei.dat
```

Eine ausführbare Version der Software *PIVA* und/oder die zugehörige Entwicklerdokumentation [Gle04] sind bei Bedarf beim Autor dieser Schrift erhältlich.

Es folgt ein alphabetisch geordneter, tabellarischer Überblick aller Kommandozeilenoptionen.

Option	**Bedeutung**	**Standardwert oder -aktion**
`-a`	Mit dieser Option wird die adaptive Verfeinerung des Vektorfeldes aktiviert; siehe 5.2. Diese Stellen werden mittels Kriterien gefunden, welche momentane Werte der Wirbelintensität bzw. Deformation auf dem aktuellen Gitter berücksichtigen. Diese Option impliziert `-r`.	
`-b beta`	Dieser Parameter gibt den oberen „Threshold"-Faktor zur Elimination von Kanten in den verschiedenen Hierarchieebenen an. Üblicherweise ist `beta` empirisch zu ermitteln und $\ll 1$ bis ≈ 1.	0.1
`-B scales`	Den multiskalaren Wavelet-Kantendetektor aktivieren. `scales` gibt die Anzahl der Hierarchiefilterebenen (Wavelet-Skalen) an, siehe Abschnitt 4.3.2. Siehe auch Option `-H`.	0 (deaktiviert)
`-C accuracy`	Kriterium zur Erkennung der Konvergenz. Wenn sich der Vektor gegenüber der letzten Iteration um weniger als `accuracy` geändert hat, gilt dies als Erfüllung des Konvergenzkriteriums.	0.01
`-d density`	`density` bezeichnet die Partikeldichte = mittlere Partikelanzahl / cm^2 oder / Pixel.	200
`-D depth`	`depth` gibt an, bis in welche Baumtiefe beim Schreiben der Vektordaten in eine Datei gestiegen werden soll.	-1 = vollständiger Baum
`-e`	Diese Option wendet einen speziellen Rauschfilter an, der Signalspitzen über die Kantenbilder-Hierarchie hinweg vergleicht und Rauschspitzen entfernt; siehe Kapitel 6.2.	

Tabelle 7.1: Die Kommandozeilenoptionen von *PIVA* (Fortsetzung auf nächster Seite)

Option	Bedeutung	Standardwert oder -aktion
`-f outbase`	Mit dieser Option wird das Speichern aller Bilder einer Hochpassfilterhierarchie aktiviert. Die Bilder werden unter Dateinamen laut dem Muster `outbase`*pn*{a|b}_*level*`.raw` gespeichert. *pn* bezeichnet die Stelle des Bilderpaares (a,b) in der Eingabereihenfolge und *level* die Hochpassfilterebene.	deaktiviert
`-g`	Damit werden die Bilder im Zuge der Verfeinerungsschritte je nach Partikeldurchmesser entsprechend durch die GP vergröbert. D. h. im 1. Schritt wird das gröbstmögliche Bild verwendet, im 2. das nächstfeinere, usw. bis zum Bild mit der originalen Auflösung; siehe 6.3.	
`-G`	Nach dem Ende der Rekonstruktion des Vektorfeldes wird mit dieser Option abschließend geglättet. Dies geschieht durch die Bildung des arithmetischen Mittels der acht benachbarten Vektoren (8er-Nachbarschaft) und die unbedingte Ersetzung des betrachteten Vektors durch den jeweiligen Mittelwert.	
`-h,-?`	Die Kommandozeilenhilfe anzeigen.	
`-H steps`	Legt die Anzahl `steps` der zu durchlaufenden SOBEL-Hochpassfilterschritte zur Erstellung der Hierarchie aus Abschnitt 6.1 fest. Siehe auch Option `-B`.	0 = kein SOBEL-Filterschritt
`-i iterations`	Dieser Schalter gibt die Anzahl der maximal zu durchlaufenden Iterationen bis zur nächsten Verfeinerung der Vektorfeldauflösung bzw. bis zur Termination der Iteration an. Es werden nur nicht konvergierte Vektoren bis zur letzten Iteration in die Analyse miteinbezogen.	2
`-I size`	Bezeichnet die anfängliche Größe der quadratischen Abfragefenster. Vorzugsweise empfehlen sich 2er-Potenzen für `size`, ebenso für die Bildabmessungen.	32
`-k`	Diese Option hat keinen Einfluss auf das PIV-Analysesystem. Wenn angegeben, dann wird in den *Skalierungsmodus* der Software umgeschaltet, in dem eine auf der Kommandozeile übergebene Vektorfelddatei zur besseren Visualisierung skaliert wird.	
`-l scalefactor`	Mit diesem Schalter kann man alle Vektoren des Vektorfeldes durch Multiplikation mit `scalefactor` skalieren. Diese Option kann auch im Skalierungsmodus `-k` verwendet werden.	1

Tabelle 7.1: Die Kommandozeilenoptionen von *PIVA* (Fortsetzung auf nächster Seite)

Option	Bedeutung	Standardwert oder -aktion
`-m maskfile`	Aktiviert die Maskierung der Eingabebilder und übergibt den Dateinamen `maskfile` des einheitlichen S/W-Maskierungsbildes. Damit können bestimmte Bereiche der PIV-Bilder aus der Beobachtung genommen werden.	
`-M magnification`	`magnification` spezifiziert die Vergrößerung M des fotografierten Bereichs gegeben durch $M = z_0/Z_0$, wobei Z_0 den Abstand des Objektes und z_0 den Abstand der Belichtungsebene jeweils zur Linsenoptik darstellen.	10
`-n`	Spezielle normalisierte Kreuzkorrelation verwenden; siehe 2.1.7. Bei unterschiedlich stark belichteten Aufnahmen ist diese Option hilfreich.	
`-N`	Alle Vektoren der Vektorfelder bei der Ausgabe in Dateien normieren. Dies kann zu einer Verbesserung der Visualisierung von Vektorfeldern beitragen. Es gilt dann $\|\vec{v}_N(x,y)\| = 1$	
`-o overlap`	Überlappungsfaktor OF der Abfragefenster; siehe 2.1.6. `overlap` ist eine ganzzahlige Prozentangabe im Wertebereich 0 bis 99 (vorzugsweise 50% oder 75%).	0 = keine Überlappung.
`-O outbase`	Basisname (Präfix) `outbase` der Vektorfeldausgabedateien.	`./vektorfeld`
`-p interval`	Zeitlicher Abstand `interval` (Δt) zwischen zwei Belichtungen/Aufnahmen in Millisekunden zur Berechnung und Ausgabe von Geschwindigkeiten in Pixel/s.	deaktiviert oder 0 = direkte Pixelverschiebungen ausgeben
`-P method`	Dieser Parameter legt die Art der Sub-Pixel-Spitzendetektion fest; siehe 2.1.1. Zur Zeit sind nur die Methoden 1 und 2 vollständig implementiert! Gültige Werte für `method` sind: 1 ... „Gaussian-fit“-Schätzer (Standard) 2 ... „Parabolic-fit“-Schätzer 3 ... „Center-of-mass“-Schätzer	1
`-r size`	Verfeinert die Vektorfeldauflösung in einem oder mehreren Schritten bis zur Erreichung von `size`; siehe Kapitel 5. Empfohlen werden hierfür 2er-Potenzen. Die Verfeinerung wird im gesamten Aufnahmebereich vorgenommen, wenn der Parameter `-a` nicht angegeben wird.	

Tabelle 7.1: Die Kommandozeilenoptionen von *PIVA* (Fortsetzung auf nächster Seite)

Option	Bedeutung	Standardwert oder -aktion
`-R thres`	Im Skalierungsmodus `-k` gibt man hier eine obere Grenze für die Länge der Vektoren an. Zu lange Vektoren werden damit aus dem Datensatz gelöscht.	0 = keinen Threshold.
`-s method`	Die Methode zur Erkennung von Ausreißern festlegen; siehe Kapitel 2.1.8. Zur Zeit ist nur die (empfohlene) Methode 2 vollständig implementiert! Gültige Werte für `method` sind: 0 ... keine Validierung verwenden 1 ... „Detectability“ 2 ... „Local-median“-Test (Standard) 3 ... „Local-mean“-Test 4 ... „Global-mean“-Test	2
`-S size`	Die Verwendung großer Suchfenster wird damit aktiviert; siehe 2.1.2. D. h. das Suchfenster wird, wenn möglich (nicht an Randbereichen!), immer um einiges grösser als das Abfragefenster angelegt. Die initiale Suchfenstergröße N_s berechnet sich dann durch $N_s = 2 \cdot \texttt{size} + N$.	deaktiviert
`-t type`	Ausgabeformat für Vektorfelddaten festlegen. Gültige Werte für `type` sind: 1 ... Eigenes Ausgabeformat 2 ... „gnuplot“-Format (Standard) 3 ... „TecPlot“-Format (Noch nicht unterstützt!)	2
`-T`	Damit werden Informationen über die Verarbeitungszeit genauer angezeigt.	
`-u`	Diese Option hat keinen Einfluss auf das PIV-Analysesystem. Wenn angegeben, dann wird der *Vergleichsmodus* des Systems aktiviert, in dem zwei auf der Kommandozeile übergebene Vektorfelddateien verglichen und ein Differenzvektorfeld ausgegeben wird.	
`-v`	Versionsinformationen anzeigen.	
`-V`	Laufzeitinformationen genauer anzeigen; das sogenannte „verbose output“.	

Tabelle 7.1: Die Kommandozeilenoptionen von *PIVA* (Fortsetzung auf nächster Seite)

Option	Bedeutung	Standardwert oder -aktion
`-w type`	Aktiviert/Deaktiviert die Fensterverschiebetechnik; siehe Teilabschnitt 2.1.3. Im Randbereich von Aufnahmen kommt diese Option nicht zur Anwendung. Gültige Werte für `type` sind: 0 ... Fensterverschiebung deaktivieren (Standard) 1 ... Diskrete Fensterverschiebung anwenden 2 ... Kontinuierliche Fensterverschiebung anwenden	deaktiviert
`-Y`	Y-Achse des Koordinatensystems spiegeln. Standardmäßig ist der Ursprung $(0,0)$ festgelegt in der linken oberen Ecke.	

Tabelle 7.1: Die Kommandozeilenoptionen von *PIVA*

Weitere Bemerkungen zur Verwendung von PIVA

Die PIV-Bilder müssen natürlich in der Reihenfolge ihrer Aufnahme angegeben werden, damit die zeitliche Zuordnung eindeutig möglich ist. Ferner müssen alle Bilder dieselben Abmessungen haben und im 8-Bit-Grauwertmodus des Windows BMP-Formates gespeichert sein.

Ein Beispiel

Der Aufruf von *PIVA*

```
piva -I 32 -r 8 -w 1 -Y -V -S 50 -G -O vf  bild_01.bmp bild_02.bmp
```

erzeugt die Vektorfelddatei `vf_001.dat` mit einer finalen Auflösung eines 8×8 Pixel Rasters. Mit dem Befehl

```
piva -u -O vf_001 reference.dat vf_001.dat
```

kann das Vektorfeld mit einem Referenzvektorfeld `reference.dat` verglichen werden, welches dieselbe Anzahl von Vektoren an den gleichen Positionen besitzen muss! Das Differenzvektorfeld befindet sich nacher in der Datei `vf_001.diff`. Mit der hier impliziten Option `-t 2` wird zusätzlich eine Datei namens `vf_001.gp` erzeugt, welche Anweisungen für eine automatische Visualisierung mit *gnuplot* beinhaltet. In zwei weiteren Dateien `vf_001.metric` und `vf_001.metric5` werden statistische Daten ausgegeben, welche in Abschnitt 8.1 erklärt werden.

7.3 Systemaufbau

Dieser Abschnitt erklärt nun die Architektur des gesamten Systems. Die wichtigsten Klassen werden separat behandelt.

7.3.1 Allgemeine Überlegungen

Das Gesamtsystem lässt sich oberflächlich betrachtet in zwei große Subsysteme unterteilen.

Rekonstruktionssubsystem — Dieses übernimmt alle Funktionen zur Rekonstruktion, Verarbeitung und Ausgabe von Vektordaten.

Bildverarbeitungssubsystem — In diesem Teilsystem werden Aufgaben wie das Laden, Bearbeiten (Filtern, etc.), Speichern und die interne Verwaltung von Bilddaten erledigt.

Zur dauerhaften Datenhaltung werden herkömmliche Dateien verwendet. Eingabebilddaten werden im BMP-Format gespeichert, Ausgabebilddaten in dem RAW-Format[7]. Die Ausgabevektorfelder werden unter dem im Abschnitt 7.2.2 spezifizierten Format gespeichert.

In der Applikation wird aus Gründen der Simplizität der bewährte nicht-nebenläufige, prozedurale Kontrollfluss eingesetzt. Dementsprechend gestaltet sich der Ablauf stapelartig in jeweils einem Durchgang per PIV-Analyse.

Auf Erweiterbarkeit wurde wie gefordert geachtet. Ebenso wurde zur Unterstützung dieser Eigenschaft eine ausführliche Entwicklerdokumentation erstellt. Die Einarbeitung eines Entwicklers und die Behebung von Softwarefehlern sollten daher in kurzer Zeit möglich sein.

Das System hat einen geringen Funktionsumfang und daher keine ausgeprägten, abstrakten Klassenstrukturen. Dennoch konnte man die Flexibilität der Architektur erhöhen, in dem man Teile der Klassenstruktur auf sogenannten Entwurfsmustern basieren ließ; siehe [GHJV95]. Zwei einfache Muster, nämlich das „Bridge“- und das „Composite“-Muster konnten gewinnbringend eingesetzt werden.

7.3.2 Überblick aller Systemklassen

Die folgende Liste gibt eine kurze Beschreibung der an der Systemarchitektur beteiligten Klassen. Eine detaillierte Beschreibung mit einer Auflistung der Methoden und Felder der einzelnen Klassen und eine gut lesbare Dokumentation der Systemkomponenten (verknüpftes HTML) findet man in [Gle04].

In Abbildung 7.1 findet man eine Auflistung der Klassen der Systemarchitektur eingeteilt in die oben genannten Subsysteme. Der Fokus dieser Dokumentation liegt auf der Beschreibung der Kernfunktionalität, deshalb folgt nun ein Überblick über die Klassen des PIV-Analysesystems:

CrossCorrelationControl — Diese Klasse verfügt über die Funktionalität zur Berechnung der statistischen Kreuzkorrelation zwischen zwei Signalen. Als Signal interpretiert werden hier zweidimensionale rechteckige Bereiche aus den beiden betrachteten Bilddatenfeldern. Als Werte der Zufallsvariablen gelten jeweils die niedrigwertigsten 8 Bit[8] eines Pixel-RGB-Wertes der internen Repräsentation, da die Aufnahme als Graustufenbild gedeutet wird. Als Ergebnis wird eine Korrelationsebene in Form eines `Array2D<double>` zurückgeliefert. Die Klasse bietet weiterhin Methoden für die

[7] Das RAW-Format spezifiziert eine formlose, zeilenweise Sequenzialisierung von z. B. $B \times H$ 24-Bit/Pixel RGB-Werten, mit B bzw. H gleich der Bildbreite bzw. -höhe. Beim Laden solcher Bilddaten in Bildverarbeitungssystemen müssen entsprechende Parameter vom Benutzer manuell angegeben werden.

[8] `0xBB` mit `B` für Blau.

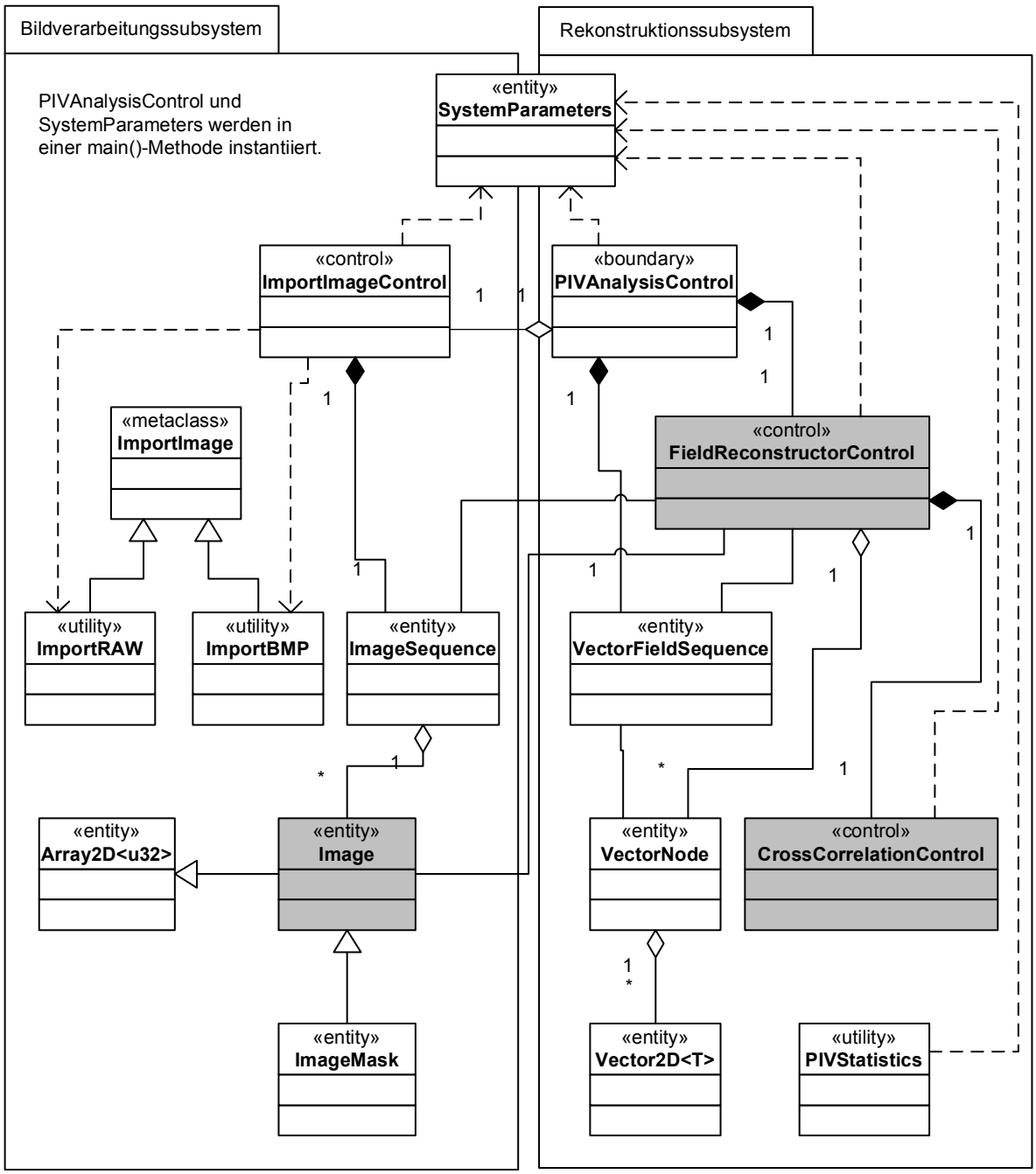

Abbildung 7.1: Die Klassen der Systemarchitektur und ihre Einteilung in die beiden Subsysteme

nötigen statistischen Berechnungen und für die Ermittlung der Korrelationsspitze. Mit dem Aufruf der Methode `setup()` wird die Berechnung eines Verschiebungsvektors an einer bestimmten Position der Aufnahmen vorbereitet und mit `doDirectCC()` durchgeführt.

FieldReconstructorControl — Das Herzstück der PIV-Analyse wird durch diese Klasse abgebildet. Sie vollzieht alle im Zusammenhang mit der Rekonstruktion von Vektoren aus den gegebenen Bildern nötigen Aufgaben und enthält folglich Methoden für einen Großteil der in Abschnitt 2 beschriebenen Verfahren. Nach der Verarbeitung eines Bilderpaares befindet sich das rekonstruierte Vektorfeld in einer Baumstruktur mit `VectorNode` und kann gespeichert werden. `FieldReconstructorControl` unterstützt nur quadratische Abfragefenster, was zu einem in beide Richtungen äquidistanten Abfrageraster und damit auch Vektorgitter führt. Die Eingabebilder müssen nicht notwendig quadratisch sein. Je nach Überlappungsfaktor OF und Größe N wird das Abfragefenster so oft wie möglich in den Bildbereich gepasst; bei arbiträren Bildgrößen bleiben dadurch Randbereiche unbetrachtet. Die Größe $B_R \times H_R$ des Abfragerasters lässt sich für die Breite B der Aufnahme durch

$$B_R = (B - N) \text{ div } \left\lfloor N \cdot \frac{OF - 100}{100} \right\rfloor + 1$$

berechnen, analog dazu H_R für die Höhe H der Aufnahme. Zu den Hauptaufgaben zählen ein iteratives, adaptives Berechnen von Schätzern durch die D-KK, eine Erhöhung der Genauigkeit durch die wahlweise diskrete oder kontinuierliche Fensterverschiebetechnik etc. unter Zuhilfenahme der Methoden der Klasse `CrossCorrelationControl`.

Image — Diese Klasse ist eine Ableitung der Vorlageninstanziierung `Array2D<unsigned long int>` und enthält eine interne Repräsentation der Bilddaten unabhängig vom Format der einzulesenden Bilddatei. Die interne Repräsentation besteht aus einem 32-Bit RGB-Wert pro Pixel in der hexadezimalen Form `0x00RRGGBB`. Ferner implementiert diese Klasse die im System verwendete Funktionalität zum Hoch- und Tiefpassfiltern von Bildern mit den Methoden `gaussian_pyramid()`, `doGaussianBlur()`, `doSobelEdgeFilter()` und `doMultiScaleDWT()`. Dazu gehören der SOBEL-Operator und der generalisierte CANNY-Filter, ein multiskalarer Wavelet-Filter. Hier verarbeitete Bilder sind einzelbelichtete Paare von Aufnahmen zur Analyse mittels Kreuzkorrelation.

PIVAnalysisControl — Die Schnittstelle zum gesamten PIV-Analysesystem wird durch diese Klasse implementiert. In der Methode `run()` werden sequentiell Bilderpaare aus einer `ImageSequence`-Instanz mit den Methoden aus `FieldReconstructorControl` verarbeitet. Für jedes Paar werden Daten initialisiert. Mit der Methode `save()` werden am Schluss der Bearbeitungsphase alle Vektorfelder aus einer angelegten `VectorFieldSequence`-Instanz in Dateien ausgelagert.

PIVStatistics — Um statistische Aussagen über ein Vektorfeld zu treffen oder dieses nachzubearbeiten, wurde diese Klasse geschrieben. Der Vektorfeldvergleichs- und auch der Skalierungsmodus von *PIVA* werden damit implementiert. Im Zuge dieser Aufgaben werden ebenso diverse Metriken, wie z. B. die durchschnittliche, prozentuale Differenz zwischen den Vektoren zweier Vektorfelder, berechnet.

SystemParameters — Diese Klassenstruktur verfügt über Felder, welche konstante und über die Kommandozeile übergebene Parameter speichern. Damit wird eine Gesamtkonfiguration für eine PIV-Analyse festgelegt. Alle Einstellungen sind direkt schreibend und lesend abrufbar und die meisten werden von der Methode `readParameters()` eingelesen, welche die GNU `getopt()`-Funktionalität[9] nutzt, um die Kommandzeileneingaben zu interpretieren.

Zu den internen Hilfsstrukturen gehören folgende Klassen- und Klassenvorlagen:

Array2D<T> — Diese C++-Klassenvorlage („Class Template") implementiert eine einfache Version eines zweidimensionalen „Random Access"-Feldes (Matrix) des Datentyps `T`. Ferner unterstützt sie auch ein sogenanntes Subindizieren von Teilfeldern. Ein Zugriff auf die Daten einer Instanz ist möglich mit dem Indizierungsoperator `[]`, z. B. durch `<Instanz>[<Zeile>][<Spalte>]`. Aus Zeitgründen wird allerdings nur der Zeilenindexbereich auf Gültigkeit geprüft.

ImageMask — Diese Klasse leitet sich von `Image` ab und enthält eine für alle in der PIV-Bildsequenz befindlichen Bilder gültige Maske zum Filtern von nicht zu analysierenden Bildbereichen. Eine gültige PIV-Bildmaske ist ein mit den PIV-Aufnahmen deckungsgleich großes BMP-Bild mit reinem Weiß (`0x00FFFFFF`) für zu maskierende Pixel (-bereiche) und reinem Schwarz (`0x00000000`) für in die Korrelation einzubeziehende Pixel (-bereiche).

ImportBMP — Eine Konkretisierung von `ImportImage` für das „Microsoft Windows Bitmap"-Format (BMP) findet man mit dieser Klasse. Sie stellt Methoden zur Verfügung, um Bilder aus einer Datei zu laden. Es werden zur Zeit allerdings nur unkomprimierte BMP-Dateien unterstützt.

ImportImage — Diese abstrakte Klasse implementiert das bereits genannte „Bridge"-Muster und dient als Schnittstelle für `ImportImageControl`. Sie stellt eine Abstraktion für alle importierbaren Bildformate dar. „Importklassen" werden von `ImportImage` abgeleitet.

ImportRAW — Eine weitere Konkretisierung von `ImportImage` implementiert `ImportRAW` und bietet die Funktionalität, ein „rohes" RGB-Bild in eine Datei zu schreiben. Da diese Klasse nur Testzwecken diente, wurde das Einlesen von RAW-Bilddaten bisher nicht implementiert.

ImportImageControl — Zum Einlesen und zur Handhabung von Bildsequenzen mit mehr als einem Bild findet man Methoden in dieser Klasse.

ImageSequence — Durch eine simple Ableitung der STL-Vorlage `deque<Image *>` kann die Klasse `ImageSequence` eine Liste beliebig vieler PIV-Aufnahmen zur Analyse verwalten. Alle Bilder der Sequenz müssen dieselben Dimensionen haben.

[9] Die Funktion `getopt()` ist direkt nur in der GNU Standard C/C++ Library verfügbar, durch die Einbindung von `unistd.h`. Auf anderen Systemen muss evtl. eine z. B. frei erhältliche Version verwendet werden.

Vector2D<T> — Diese Klassenvorlage repräsentiert einen zweidimensionalen Vektor des Typs `T`. Auf die Komponenten kann direkt lesend und schreibend zugegriffen werden. Um alle Methoden der Vorlage sinnvoll nutzen zu können, kommen als Felddatentypen nur `char`, `[short|[long] long] int` in den `signed|unsigned`-Versionen und `float` bzw. `[long] double` in Frage.

VectorFieldSequence — Durch eine simple Ableitung der STL-Vorlage `deque<Vectornode *>` kann `ImageSequence` eine Liste beliebig vieler Vektorfelder als Ergebnis der Rekonstruktionsprozedur verwalten.

VectorNode — Diese Klasse implementiert das oben genannte „Composite"-Muster und bildet das generierende Element der als Grundlage der Vektorfeldrekonstruktion benutzten Baumstruktur. Eine `VectorNode`-Instanz ist ein Blatt des Baumes, wenn sie keine Zeiger auf Kinder enthält. Eine `VectorNode`-Instanz ist ein Knoten, wenn sie ein Kinderknotenfeld mit mindestens einem gültigen Zeiger ($\neq$ `NULL`) auf ein Kind hat.

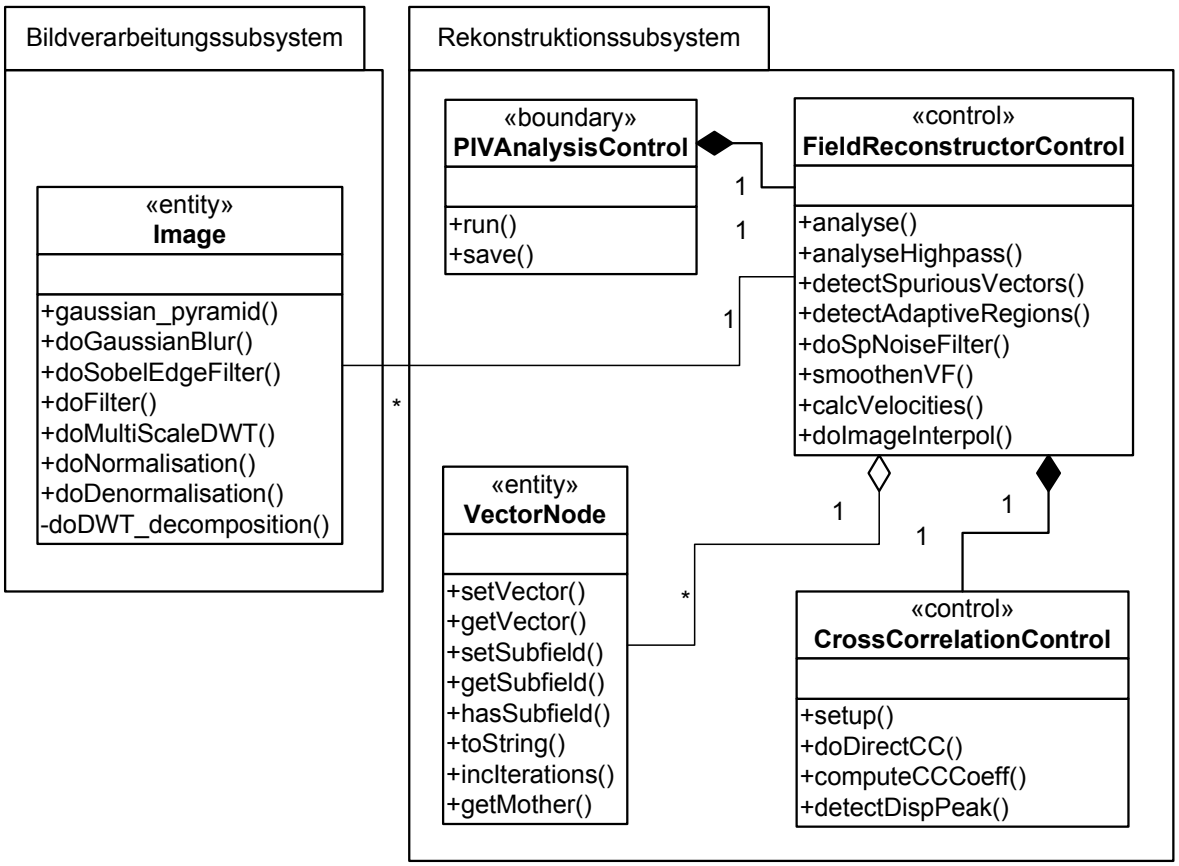

Abbildung 7.2: Die Kernklassen der Systemarchitektur mit den wichtigsten Methoden zur PIV-Analyse eines Paares von Aufnahmen.

7.4 Schematische Beschreibung der Kernkomponenten und Algorithmen

Die Abbildung 7.2 gibt Aufschluss über die Zusammenarbeit der Kernkomponenten und veranschaulicht die Aufteilung in die beiden in Abschnitt 7.3 genannten Subsysteme. Wir werden uns nun die Algorithmen in den Kernkomponenten des Systems veranschaulichen. Dazu eignet sich eine übersichtliche Darstellung mit Hilfe von UML-Sequenzdiagrammen. Die folgende Beschreibung steigt von den groben Strukturen der Anwendung hinab in die Feinen.

7.4.1 Die Klasse `PIVAnalysisControl`

Das UML-Sequenzdiagramm in 7.3 erklärt die hauptsächliche Aufgabe der Klasse `PIVAnalysisControl`. Für jedes verfügbare Paar von Aufnahmen werden die entsprechenden Daten initialisiert, d. h. die Grundstruktur des Vektorfeldbaumes wird aufgebaut. Danach wird die Analyse mit `FieldReconstructorControl` gestartet.

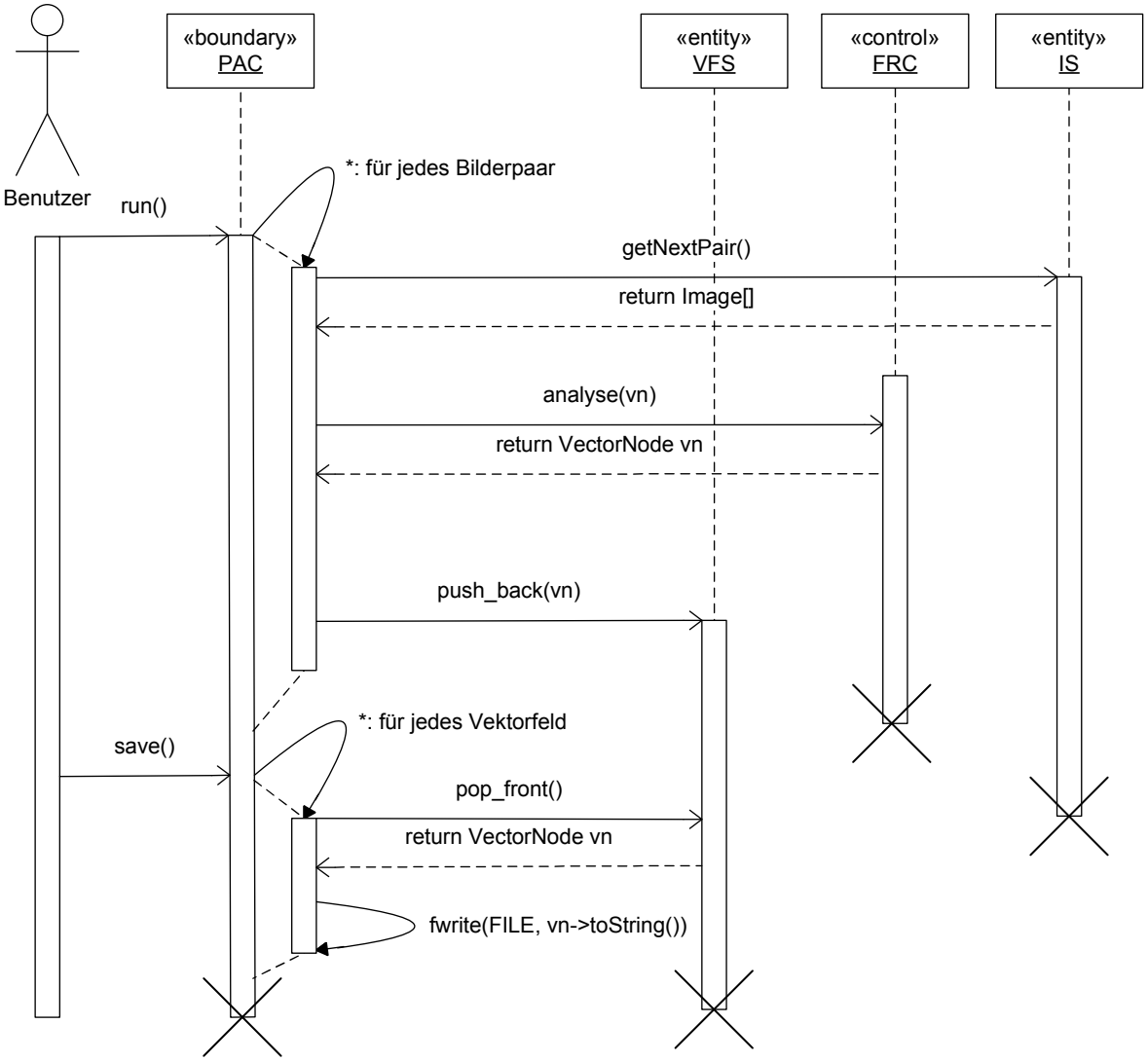

Abbildung 7.3: Die schematische Arbeitsweise der Klasse `PIVAnalysisControl` als UML-Sequenzdiagramm. Die Objektinstanz-Klassenbeziehung: PAC → `PIVAnalysisControl`, IS → `ImageSequence`, FRC → `FieldReconstructorControl`, VFS → `VectorFieldSequence`

7.4.2 Die Klasse `FieldReconstructorControl`

Wie bereits oben beschrieben, läuft die Hauptarbeit der PIV-Analyse in der Klasse `FieldReconstructorControl` ab. Die Funktionalität ist entsprechend groß. Es handelt sich aber um rekursive Methoden, die einheitlich auf dem in vorhergehenden Iterationen erstellten Vektorfeldbaum arbeiten bzw. diesen erweitern. Das Sequenzdiagramm 7.4 zeigt den prinzipiellen Ablauf mit den wichtigsten Schritten.

7.4.3 Die Klasse `CrossCorrelationControl`

`CrossCorrelationControl` ist zuständig für die effiziente bzw. genaue Berechnung der Kreuzkorrelation. Für die Ermittlung eines Verschiebungsvektors müssen das zu observierende Bilderpaar, die Positionen von Abfrage- und Suchfenster und der bereits ermittelte Anteil der Verschiebung durch die Methode `setup()` festgelegt werden. Parameter zur Auswahl der Verfahrensvariante werden aus einer globalen Instanz von `SystemParameters` entnommen. Der Ablauf wird in Abbildung 7.5 gezeigt.

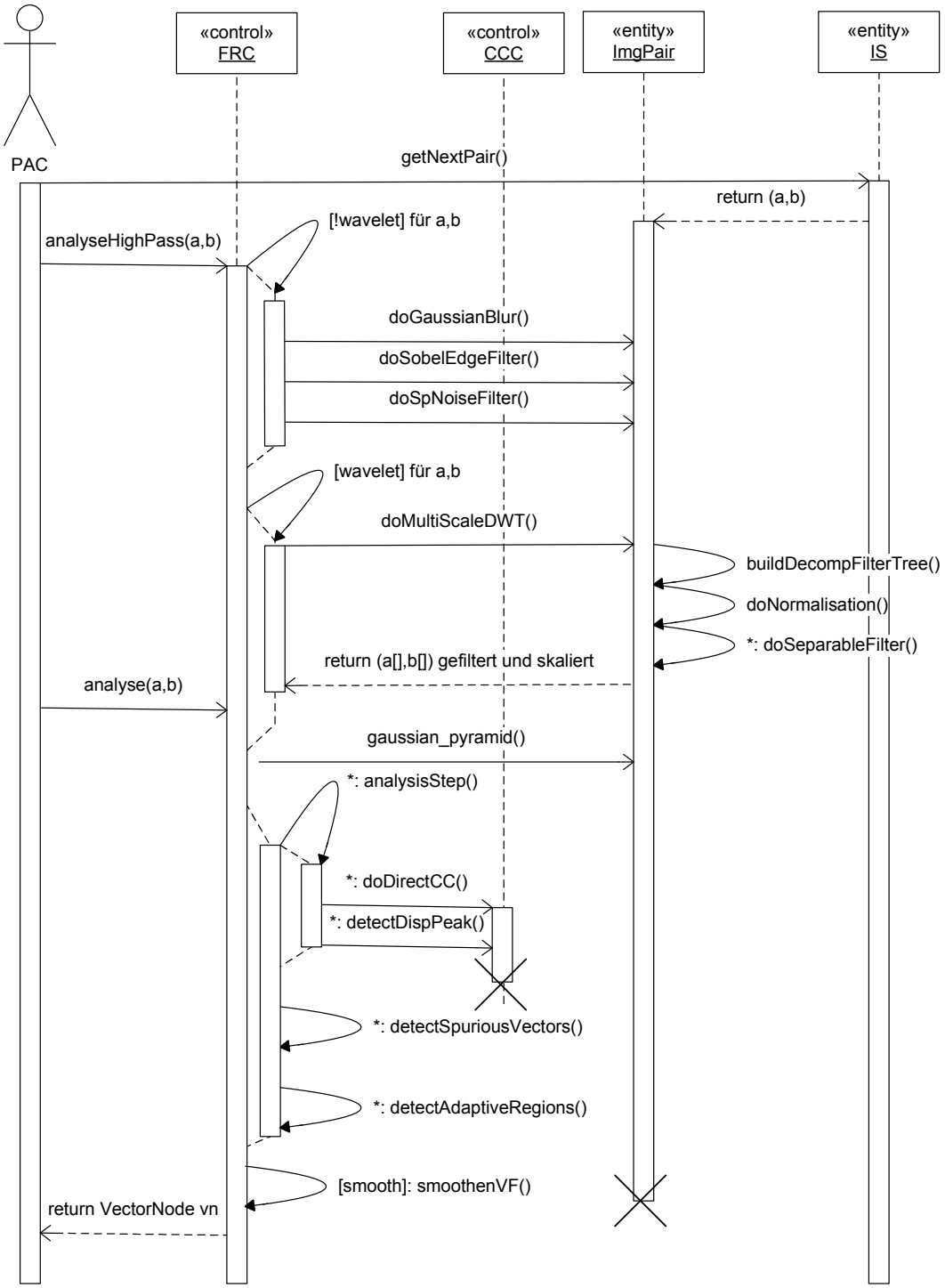

Abbildung 7.4: Die schematische Arbeitsweise der Klasse `FieldReconstructorControl` als UML-Sequenzdiagramm. Die Objektinstanz-Klassenbeziehung: PAC → `PIVAnalysisControl`, IS → `ImageSequence`, FRC → `FieldReconstructorControl`, ImgPair → `Image[]`, CCC → `CrossCorrelationControl`

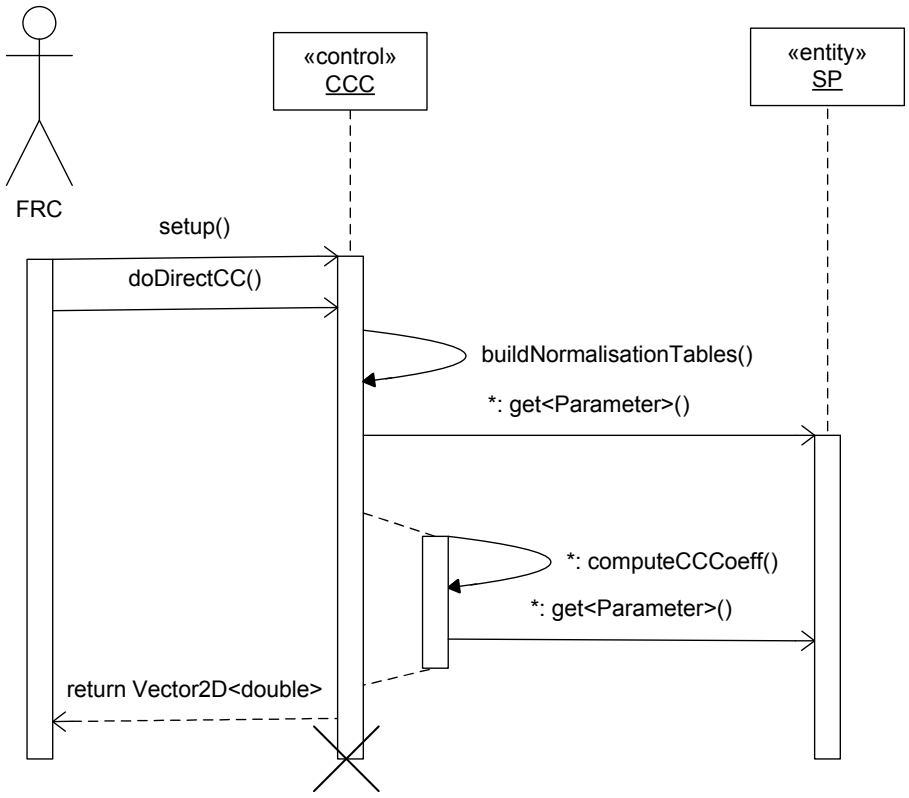

Abbildung 7.5: Die schematische Arbeitsweise der Klasse `CrossCorrelationControl` als UML-Sequenzdiagramm. Die Objektinstanz-Klassenbeziehung: FRC → `FieldReconstructorControl`, CCC → `CrossCorrelationControl`, SP → `SystemParameters`

8 Testexperimente und Ergebnisse

Die Validierung des Einflusses verschiedener Verfahrensparameter auf die Ergebnisse liefert wichtige Aussagen über die Qualität der PIV-Analyse

---

Zur Validierung der implementierten Ansätze ist es unerlässlich, praktische Testexperimente durchzuführen. Testreihen wurden mit künstlich erzeugten Partikelbildaufnahmen und mit realen Aufnahmen biologischer Strukturen durchgeführt. Für frei konstruierbare Experimente stand leider keine eigene PIV-Hardware zur experimentellen Betrachtung von Strömungen zur Verfügung.

Alle betrachteten Bilder in beiden Testreihen liegen wie schon in Abschnitt 7.2.3 erwähnt in dem 8-Bit-Graustufen BMP-Format vor. Hier sei vorweggenommen, dass *PIVA* noch keine kontinuierliche Fensterverschiebetechnik beherrscht und daher der „peak-locking"-Effekt in den Ergebnissen nicht kompensiert wird. Dies ist jedoch für die Kernaussage der Testexperimente nicht problematisch, da die Fehlerbereiche in den verglichenen Verfahrensvarianten i. d. R. größer als der durch diesen Effekt verursachte periodische Fehler sind.

8.1 Definition allgemeiner Vergleichsmetriken

Als Vergleichsmetrik wird hier ein bewertendes Kriterium zum Vergleich zweier Vektorfelder bzw. darauf berechneter statistischer Werte bezeichnet.

Für die Tabelle 8.1 gilt: Die Funktion $\vec{v}_l(i,j) : \mathrm{V} \rightarrow \mathbb{R}^2$ liefert einen reellen Verschiebungs- oder Geschwindigkeitsvektor an der Stelle (i,j) eines einheitlichen Vektorgitters V der Abmessungen $B_V \times H_V$. Es gilt $l = T$ für das zu testende Vektorfeld und $l = R$ für das jeweilige Referenzfeld.

In der Tabelle 8.1 findet man einige Vorschläge für Metriken zur vergleichenden Validierung, u. a. basierend auf statistischen Momenten.

Metrik	Definition	Idealwert
M_1 berechnet das arithmetische Mittel der prozentualen Abweichung zum Referenzfeld mit der Norm $\|\cdot\|$ (siehe Tabelle B.1).	$$M_1 \stackrel{!}{=} \frac{100}{B_V \cdot H_V} \sum_{\substack{0 \le i < B_V \\ 0 \le j < H_V}} \frac{\|\vec{v}_R(i,j) - \vec{v}_T(i,j)\|}{\|\vec{v}_R(i,j)\|}$$	0
M_2 berechnet das arithmetische Mittel der Vektoren des zu testenden Vektorfeldes.	$$M_2 \stackrel{!}{=} \langle \vec{v}_T \rangle \equiv \frac{1}{B_V \cdot H_V} \sum_{\substack{0 \le i < B_V \\ 0 \le j < H_V}} \vec{v}_T(i,j)$$	$\langle \vec{v}_R \rangle$
M_3 berechnet die Varianz der Vektoren des zu testenden Vektorfeldes mit Hilfe der Norm $\|\cdot\|$.	$$M_3 \stackrel{!}{=} \mathrm{var}\{\|\vec{v}_T\|\} \equiv \frac{1}{B_V \cdot H_V} \sum_{\substack{0 \le i < B_V \\ 0 \le j < H_V}} \|\vec{v}_R(i,j) - \langle \vec{v}_T \rangle\|^2$$	$\mathrm{var}\{\|\vec{v}_R\|\}$
M_4 zählt die nach einem „local-median"-Test gefundenen Ausreißer; wie in 2.1.8 beschrieben.	$$M_4 \stackrel{!}{=} \sum_{\substack{0 \le i < B_V \\ 0 \le j < H_V}} \mathrm{median}\{\vec{v}_T(i,j)\} \quad \text{mit}$$ $$\mathrm{median}\{\vec{v}_T(i,j)\} = \begin{cases} 1, & \text{Ausreißer} \\ 0, & \text{sonst} \end{cases}$$	0
M_5 erzeugt ein Histogramm H mit l Intervallen zur Veranschaulichung der Verteilung der x- bzw. y-Komponenten der Vektoren. Eine Betrachtung der Vektorlänge $\|\vec{v}_T\|$ wäre hier eine weitere Möglichkeit. Wahlweise Approximation des Histogramms mittels Hüll- oder Schwerpunktskurve.	$$M_5 \stackrel{!}{\equiv} H_x(l) = \sum_{\substack{0 \le i < B_V \\ 0 \le j < H_V}} \begin{cases} 1, & (l-1)\varepsilon < \vec{v}_T^{(x)}(i,j) \le l\varepsilon \\ 0, & \text{sonst} \end{cases}$$ $$\text{mit} \quad l \in \left[1 \ldots \left\lceil \frac{\max_{i,j}\{\vec{v}_T^{(x)}(i,j)\} - \min_{i,j}\{\vec{v}_T^{(x)}(i,j)\}}{\varepsilon} \right\rceil \right]$$ und passend gewähltem ε, z. B. 0.1. Analog $H_y(l)$ für $\vec{v}_T^{(y)}$.	kein sichtbarer „peak-locking"-Effekt

Tabelle 8.1: Vorschläge für Testmetriken zur Validierung der Vektorfelder

8.2 Testexperiment 1: Computergenerierte Aufnahmen

Die folgenden Testreihen mit PIV-Analysen wurden auf computergenerierten Bildern gemacht, welche basierend auf einer Monte-Carlo-Simulation[1] erzeugt wurden. [ONSK00] erklärt die Anforderungen an künstliche Bilder und bietet eine Reihe von Referenzbild- und Vektordaten, mit welchen eine große Anzahl der zweidimensionalen PIV-Verfahrensvarianten hinreichend evaluiert werden können. Ferner beschreibt auch [Bol99] die Verwendung einer Monte-Carlo-Simulation zur Erstellung von künstlichen Aufnahmen zur Validierung der PIV-Analyse.

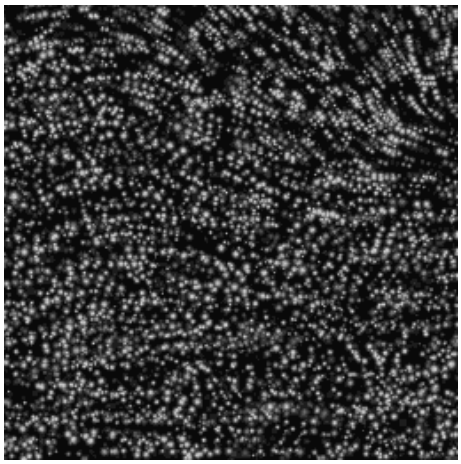

Abbildung 8.1: Zur Veranschaulichung überlagertes, künstliches Aufnahmepaar (zwei getrennte Aufnahmen) zum Zeitpunkt t und $t+\Delta t$

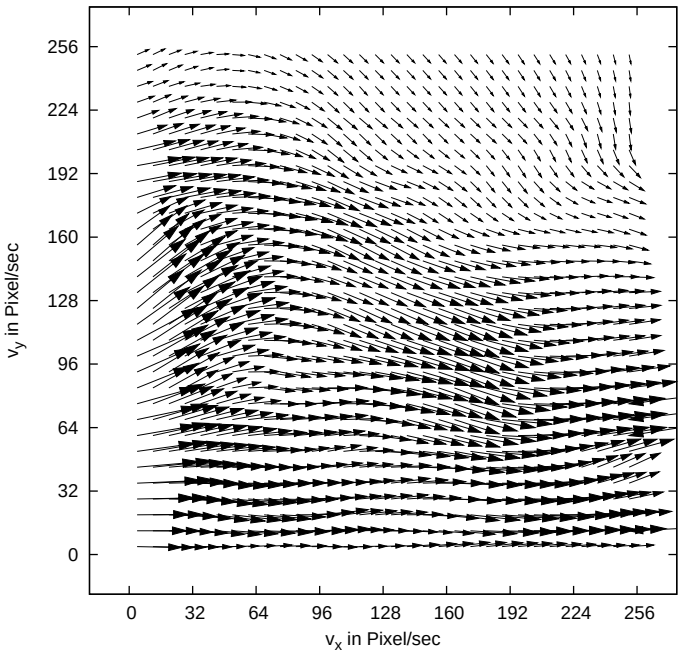

Abbildung 8.2: Das Referenzvektorfeld `#01` von der Homepage des „PIV Standard Image"-Projektes. Die Skalen geben hier und in allen folgenden Vektorfeldern im Unterschied zu den Beschriftungen die Position der Vektoren im Bild in Pixeln an.

8.2.1 Informationen zu den Testbildern und Systemparameter

Als Testbilder wurde die Reihe `#01` der auf der Homepage von [ONSK00][2] angebotenen Bilderreihen verwendet, welche die Referenzpartikelbewegung in einer zweidimensionalen Wandscherströmung darstellt (siehe Abbildungen 8.1 und 8.2). Entsprechende Referenzvektordaten gegeben in einem 8×8-Raster in Pixel/Δt für den Vergleich sind dort ebenso erhältlich[3]. Für das Experiment geltende Parameter sind in der Tabelle 8.2 aufgelistet.

[1] Ein Monte-Carlo-Algorithmus ist ein randomisierter, also auf Zufallszahlen basierender, Algorithmus, der allerdings mit einer gewissen (meist sehr geringen) Wahrscheinlichkeit ein falsches Ergebnis liefern kann. Der tragende Vorteil dieses Ansatzes liegt in seiner hohen Effizienz.

[2] Siehe `http://vsj.or.jp/piv/`.

[3] Siehe `http://piv.vsj.or.jp/piv/java/tmp/vel.dat`.

Systemparameter	Wert	Einheit
Bildabmessungen $B \times H$	256×256	Pixel
Aufnahmeintervall Δt	33	msec
Durchschnittliche Partikelbildgeschwindigkeit $\|\langle\vec{v}\rangle\|$	7.5	Pixel/Δt
Maximale Partikelbildgeschwindigkeit $\|\vec{v}_{max}\|$	15.0	Pixel/Δt
Durchschn. normal. „out-of-plane"-Geschwindigkeit $\|\langle w\rangle\|$	0.017	
Anzahl der Partikelbilder P	4000	
Durchschnittlicher Partikelbilddurchmesser $\langle D\rangle$	5.0	Pixel
Finale Abfragefenstergröße N_f	8×8	Pixel

Tabelle 8.2: Die wichtigsten Systemparameter für Testexperiment 1; entnommen aus [ONSK00].

Für den Test wurde folgender Verfahrenmodus gewählt bzw. wurden folgende Verfahrensparameter und damit *PIVA*-Optionen variiert:

- initiale Abfragefenstergröße N_i (Optionen `-I` und `-S`),
- Anzahl der Hochpassfilterebenen h (Option `-H`),
- mit und ohne speziellem Rauschfilter (Option `-e`),
- mit Vergröberung der Aufnahmen (Option `-g`),
- mit diskreter und kontinuierlicher Fensterverschiebung (Option `-w`).

8.2.2 Ergebnisse und Bilder

Die Ergebnisse waren meist sehr zufriedenstellend. In Abbildung 8.3 kann man das Ergebnis einer sehr guten Rekonstruktion der Vektordaten sehen. Das Bild 8.4 zeigt den Unterschied zur Referenz 8.2 in Form von Differenzvektoren. Mit dem hierarchischen, iterativen Verfahren können sogar bei Abfragefenstergrößen von 4×4-Pixeln brauchbare Ergebnisse erzielt werden.

Das Diagramm 8.5 bietet eine Übersicht mehrerer Testreihen mit der Variation verschiedener Systemparameter. Ersichtlich ist daraus, dass bei den künstlichen Bildern mit den angedachten Filtermethoden aus der digitalen Bildverarbeitung keine Verbesserung erzielt werden kann. Die üblichen Verfahrensvarianten der PIV erreichen hier ideale Ergebnisse. Damit ist vor allem die kontinuierliche Fensterverschiebetechnik gemeint, siehe Diagramm 8.6. Diese Resultate bzgl. der Metrik $\mathbf{M}_1$ sind erwartungsgemäß um einiges besser als alle bisherigen. Die Qualität des Wavelet-Kantendetektors sticht zwar durch gute Werte heraus, erübrigt sich aber bei den künstlichen Aufnahmen ebenfalls.

Das Histogramm 8.7 zeigt die Verteilung der Vektorlängen laut Metrik $\mathbf{M}_5$ ohne Beseitigung des „peak-locking"-Effektes. Der **Standardfehler von etwa 6%** rührt daher, dass am Rand der Aufnahmen keine spezielle Behandlung in der Software stattfindet und somit in ihrer Länge stark nach unten verzerrte Vektoren zustande kommen. Am Rand wird die

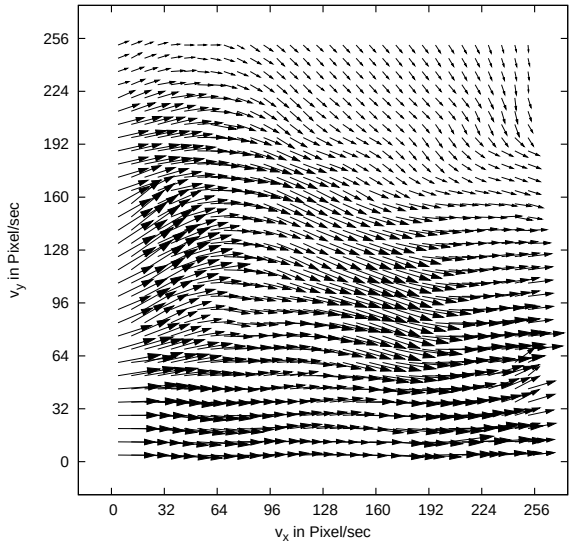

Abbildung 8.3: Das beste Ergebnisvektorfeld von *PIVA* nach drei Iterationen mit diskreter Fensterverschiebetechnik und einer initialen Abfragefenstergröße von 16×16 Pixel.

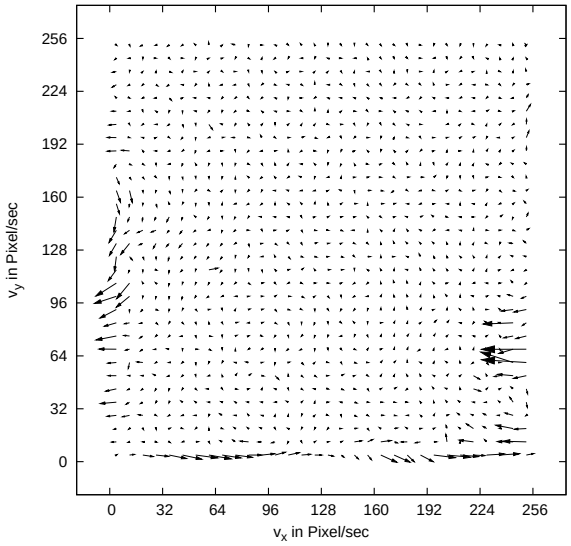

Abbildung 8.4: Das Differenzvektorfeld berechnet aus der Abbildung 8.3 und der Referenz 8.2. Die normalverteilten Fehler sind hier aufskaliert. Die überproportionalen Ausschläge am Rand entstehen durch eine in der Software fehlende Spezialbehandlung der Randbereiche der Aufnahmen und können unberücksichtigt bleiben.

einfachste PIV-Analysemethode, u. a. ohne Fensterverschiebetechnik, verwendet. Ergebnistrends werden durch diese Tatsache aber nicht beeinflusst.

Eine Vergröberung der Aufnahmen durch das GP-Schema (Option `-g`) führt hier aufgrund der genügend großen Partikelbilder bei viel kürzeren Bearbeitungszeiten zu guten Ergebnissen. Die GP wirkt als Tiefpassfilter, was bei den praktisch rauschfreien, künstlichen Bildern anscheinend keinen negativen Einfluss hat. Da die künstlichen Bilder eben keinen hier relevanten Rauschanteil besitzen, bleiben die Ergebnisse der Testreihen mit den Filterhierarchien und der GP-Vergröberung jedoch bestenfalls gleich bzw. werden sogar schlechter.

Eine Glättung des gesamten Vektorfeldes als letzter Schritt der Rekonstruktion führt naheliegenderweise zu einer Verschlechterung der Genauigkeit, jedoch trotzdem zu einer qualitativ schöneren Visualisierung.

In den Abbildungen 8.8 und 8.9 sind die Unterschiede der Kantenbilder zwischen dem SOBEL-Operator (Ansatz A) und dem Wavelet-Operator (Ansatz B) ersichtlich.

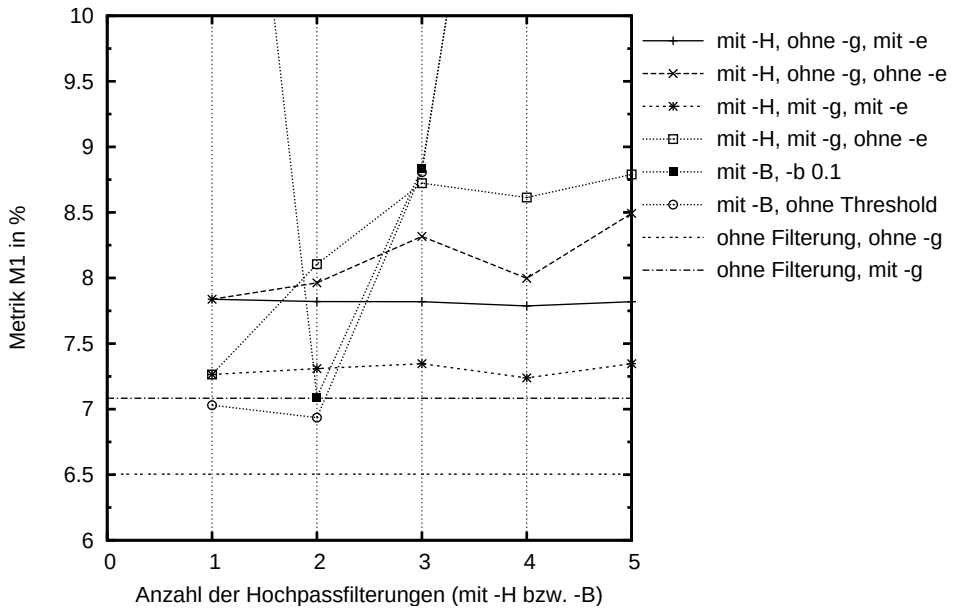

Abbildung 8.5: Die Aufzeichnung der Testreihe in Testexperiment 1 mit Hilfe der in Kapitel 8.1 definierten Metrik $\mathbf{M}_1$. Die besten Ergebnisse werden jedoch ohne digitale Filter erzielt. Der Standardfehler von 6% wird dadurch erzeugt, dass in den Randbereichen der Aufnahmen nur einfache PIV-Methoden angewendet werden.

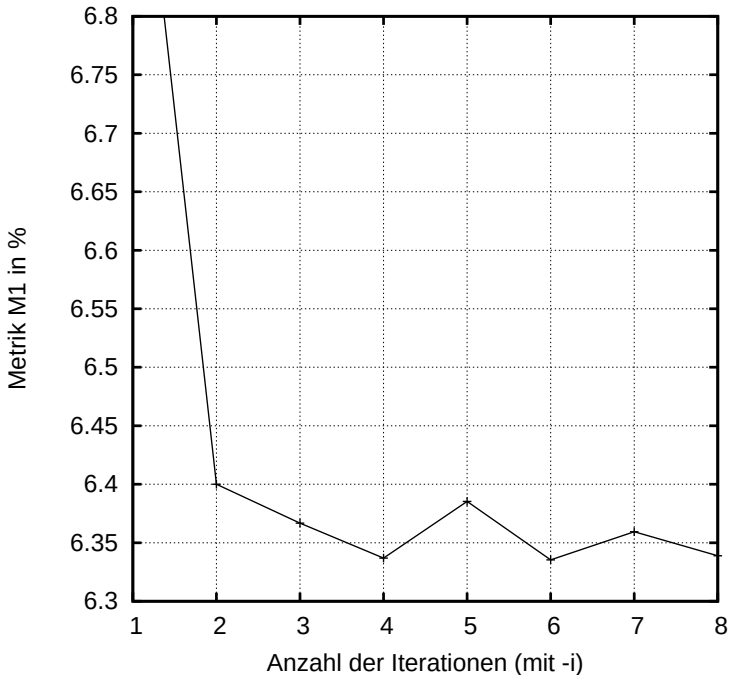

Abbildung 8.6: Ergebnis der kontinuierlichen Fensterverschiebetechnik auf den künstlichen Aufnahmen. Die Metrik $\mathbf{M}_1$ schwingt abklingend, konvergiert dann gegen einen Wert um $\approx 6.34\%$ und erzielt erwartungsgemäß die besten Ergebnisse. Die weitgehende Elimination des „peak-locking"-Effektes und die besseren Ergebnisse sind im Vergleich zu Diagramm 8.5 ersichtlich.

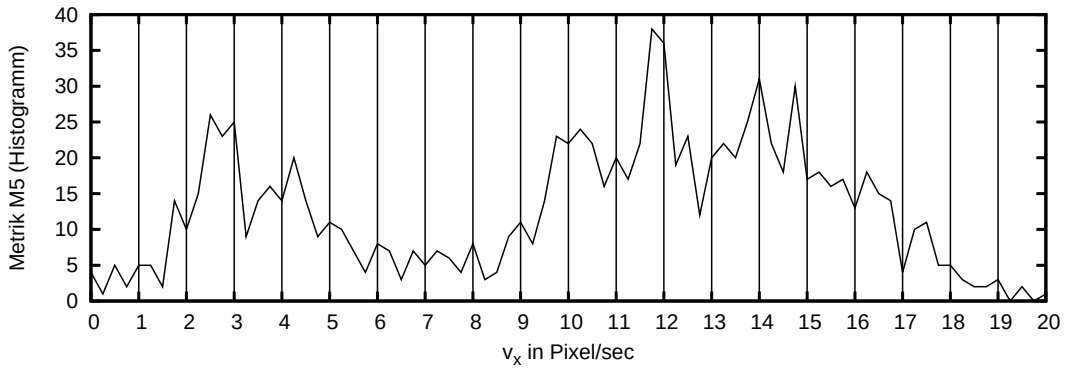

Abbildung 8.7: Das Histogramm über die x-Komponentenlängen des Vektorfeldes aus Abbildung 8.3 mit $\varepsilon = 0.25$, siehe $\mathbf{M}_5$ in Tabelle 8.1. Ein leichter „peak-locking"-Effekt ist sichtbar.

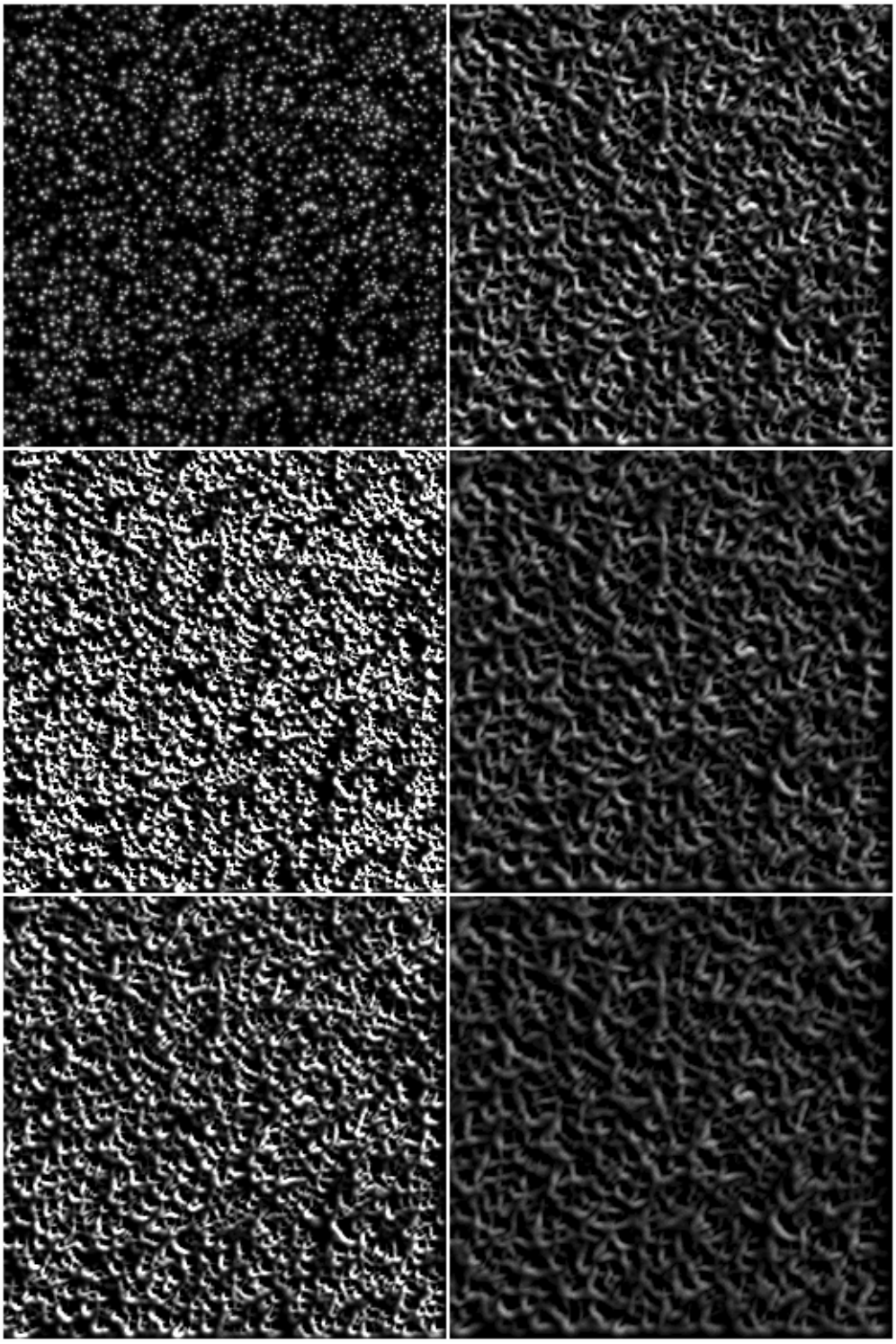

Abbildung 8.8: Gezeigt wird die Kantenbildhierarchie in Testexperiment 1 mit Ansatz A. **Links oben** befindet sich das Original (ohne Option `-H` oder mit `-H 0`). In den hierarchischen Kantenbildern kann man die Kantenstrukturen nach 1–5 Glättungen (`-H 1` bis `-H 5`) des Originals erkennen. Aus den Bildern ist eine einseitige Kantendetektion ersichtlich.

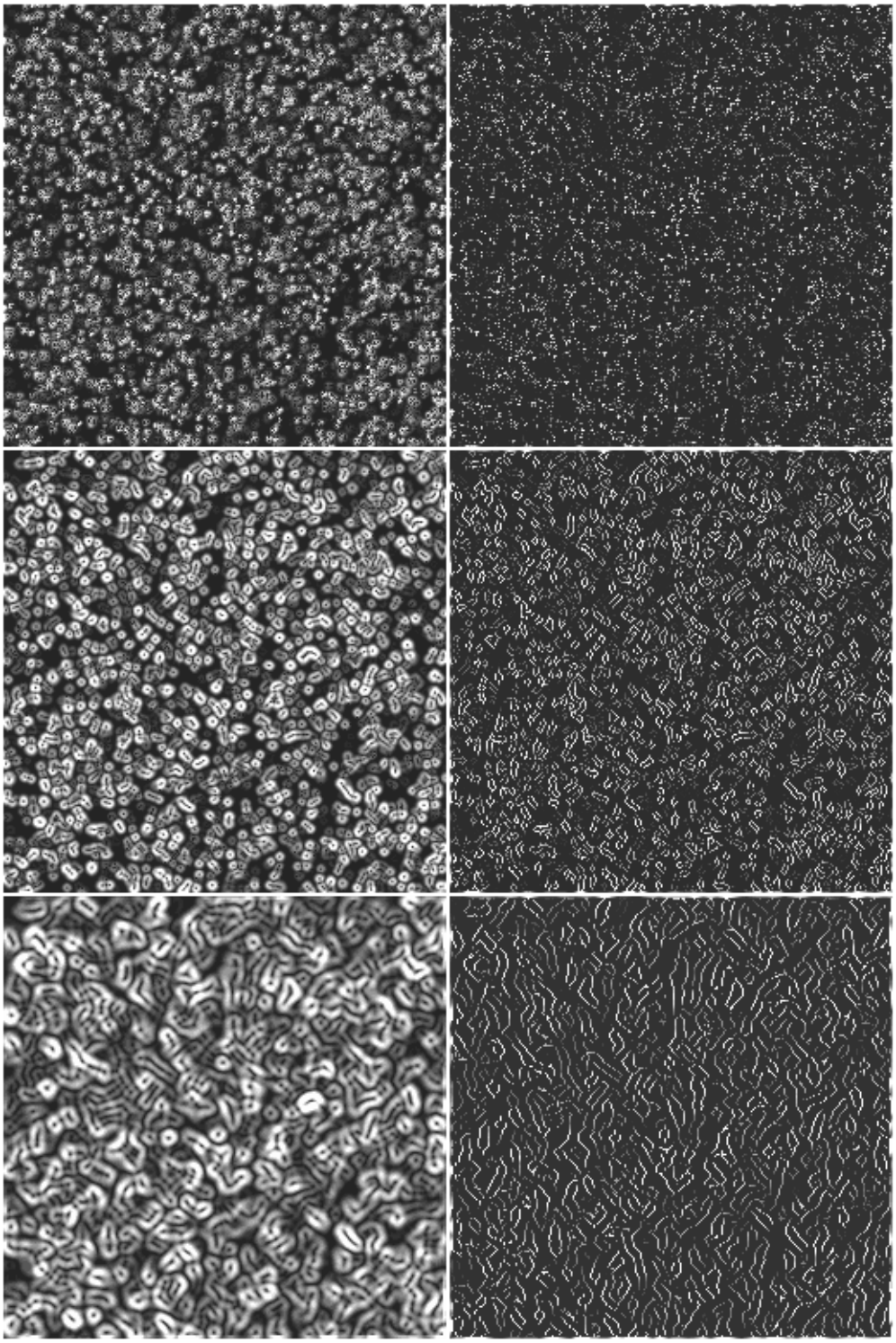

Abbildung 8.9: Gezeigt wird die Kantenbildhierarchie in Testexperiment 1 mit Ansatz B. In den hierarchischen Kantenbildern mit den Moduli der DWT in der **linken Spalte** kann man die Kantenstrukturen nach 1–3 Glättungen (`-B 1` bis `-B 3` und `-b 0`) des Originals aus Abbildung 8.1 erkennen. Die **rechte Spalte** enthält die dazu gehörenden Modulmaxima, welche scharfe Kantenverläufe darstellen.

8.3 Testexperiment 2: Reale Aufnahmen von mikroskopischen, biologischen Strukturen

Diesem Test lagen zwei Aufnahmen von biologischen Zellstrukturen wie in Abbildung 8.10 gezeigt vor, welche sich in mikroskopischen Skalen bewegen und Strömungen mit zahlreichen kleinen Wirbelstrukturen erzeugen. Verfahren aus dem Bereich des μ-PIV kommen hier also auch in Frage. Zum Vergleich der Ergebnisse gab es hier leider keine Referenzvektordaten. Deshalb wurde ein relativer und visueller Vergleich zwischen den verschiedenen Ergebnissen der Testreihe gemacht, um sich somit an ein Referenzergebnis empirisch zu nähern. Nach dieser Prozedur kann der Einfluss verschiedener Parameter auf das System besser beobachtet werden. Anstatt der Partikelbildgeschwindigkeit $\vec{v}$ wurde hier die Partikelbildverschiebung $\vec{r}$ verwendet, da der zeitliche Abstand der beiden Aufnahmen nicht bekannt war. Ferner wurden Ergebnisse visuell und soweit möglich statistisch mit einer frei erhältlichen Demonstrationsversion der Software *PIV View 2C (Version 2.2)* der Firma *PIVTec GmbH* verglichen, um grobe Unstimmigkeiten von vornherein beseitigen zu können. Mit dieser Version der funktionsreichen Software konnten allerdings keine Vektordaten zur Weiterverarbeitung (z. B. zum rechnerischen Vergleich) gespeichert werden.

Abbildung 8.10: Die erste Aufnahme der biologischen Strukturen zum Zeitpunkt t

Systemparameter	Wert	Einheit
Bildabmessungen $B \times H$	512×512	Pixel
Durchschnittliche Partikelbildverschiebung $\langle \vec{r} = (r_D, r_S)^T \rangle$	$(0.034, -0.0096)$	Pixel/Δt
Maximale Partikelbildgeschwindigkeit $\|\vec{r}_{max}\|$	2.589	Pixel/Δt
Minimale Partikelbildgeschwindigkeit $\|\vec{r}_{min}\|$	0.00095	Pixel/Δt
Finale Abfragefenstergröße N_f	8×8	Pixel
Anzahl der Vektoren	4096	
Initiale Suchfenstergröße	14×14	Pixel
Anzahl der Iterationen	1 (ohne `-w`)	

Tabelle 8.3: Bekannte und durch Versuche ermittelte Systemparameter für Testexperiment 2

8.3.1 Informationen zu den Testbildern und Systemparameter

Aus Tabelle 8.3 können die für diesen Test bekannten Parameter entnommen werden. Dabei ist zu erwähnen, dass hier im Gegensatz zu den künstlichen Bildern weitaus weniger Vorwissen existiert, um die Qualität der durchgeführten Tests zu evaluieren.

Vorangegangene Überlegungen haben die Rolle des Vergleichs zwischen diskreter und kontinuierlicher Fensterverschiebetechnik (u. a. mit *PIV View*) in den Tests sehr gestärkt, sodass im Folgenden auch dieser Aspekt in den Ergebnissen berücksichtigt wird. In beiden Anwendungen *PIVA* und *PIV View* wurde eine Abfragefenstergröße von 8×8 und keine Überlappung ($OF = 0$) eingestellt und am Schluss der Analyse eine lokale Mittelwertglättung des gesamten Vektorfeldes vorgenommen. *PIVA* wurde mit den Kommandozeilenparametern

```
piva -i 1 -I 8 -S 3 -e -H <level> -G -s 2
     -O <ausgabe> <bild1>.bmp <bild2>.bmp
piva -O <differenz> -u <referenz>.dat <ausgabe>.dat
```

ausgeführt. *PIV View* wurde wie bereits angedeutet zusätzlich mit einem iterativen Sub-Pixel-Fensterverschiebeverfahren ausgeführt. Das damit erhaltene Ergebnis wurde für spätere Vergleiche bzgl. $\mathbf{M}_2$ als Referenz verwendet.

8.3.2 Ergebnisse und Bilder

Das Hauptaugenmerk lag bei diesem Test auf der Frage, ob man mit Hochpass- und Rauschfilterung der Aufnahmen die Anzahl der Ausreißer bzw. das Rauschen in der Korrelationsebene vermindern kann.

In Abbildung 8.11 kann man gute Ergebnisse mit der herkömmlichen D-KK sehen. Strömungen und Wirbelstrukturen sind im Vektorfeld gut nachvollziehbar, obwohl hier noch keine Sub-Pixel-Fensterverschiebung angewendet worden ist. In Bereichen mit keiner nachvollziehbaren Bewegung (z. B. obere Mitte im Bild 8.10) bilden sich Zufallsvektorfelder sehr kleinen durchschnittlichen Betrages. Aus Abbildung 8.12 sind zwar visuell gute Ergebnisse mit der SOBEL-Hochpass- und Rauschfilterung ersichtlich, bei betragsmäßig großen Verschiebungen kann jedoch meist eine Verzerrung der Vektoren nach oben festgestellt

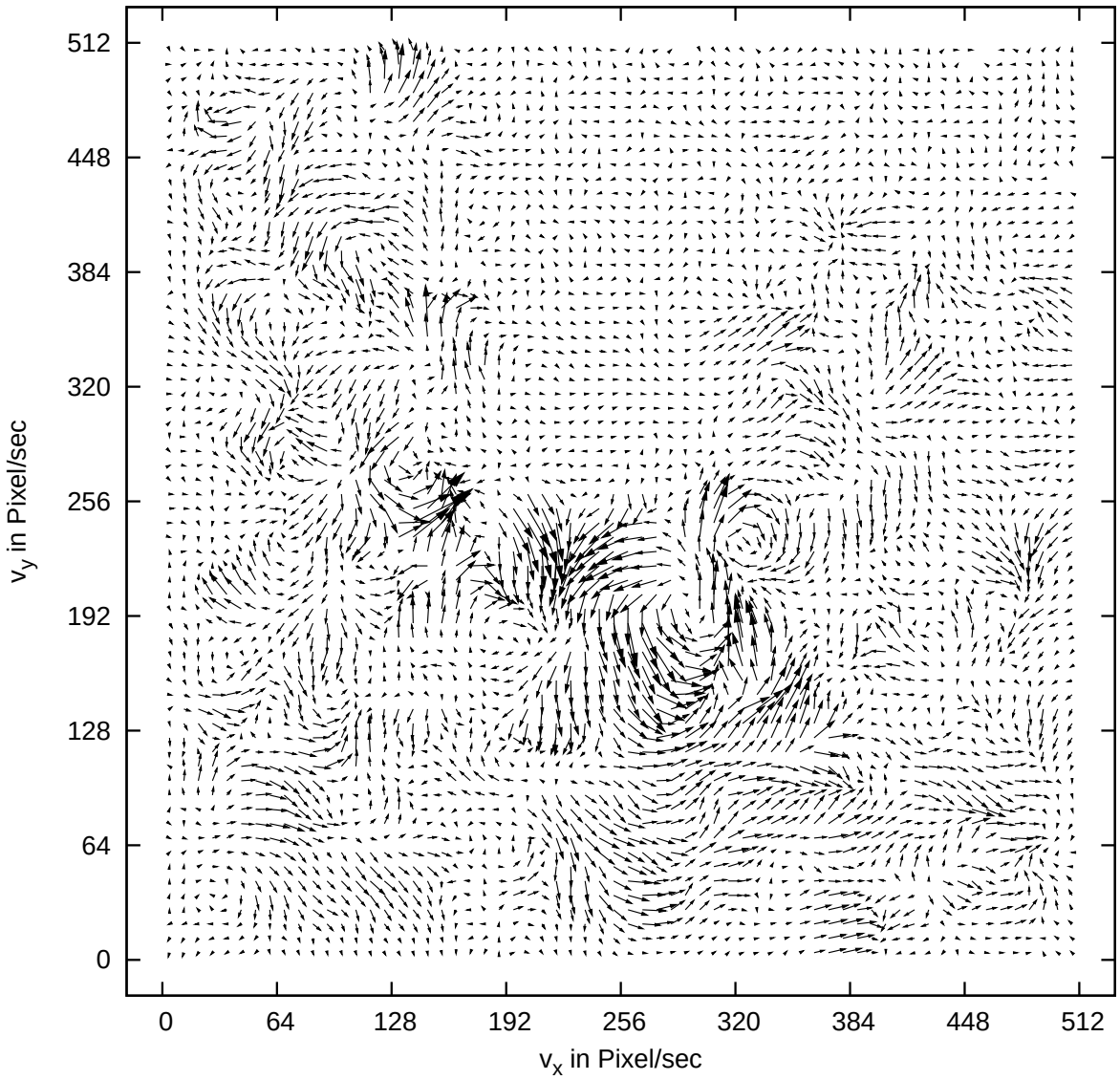

Abbildung 8.11: Ein Ergebnisvektorfeld von *PIVA* in der Testreihe 2 mit ersetzten Ausreißern. SOBEL-Hochpass- und Rauschfilter wurden hier **nicht** angewendet!

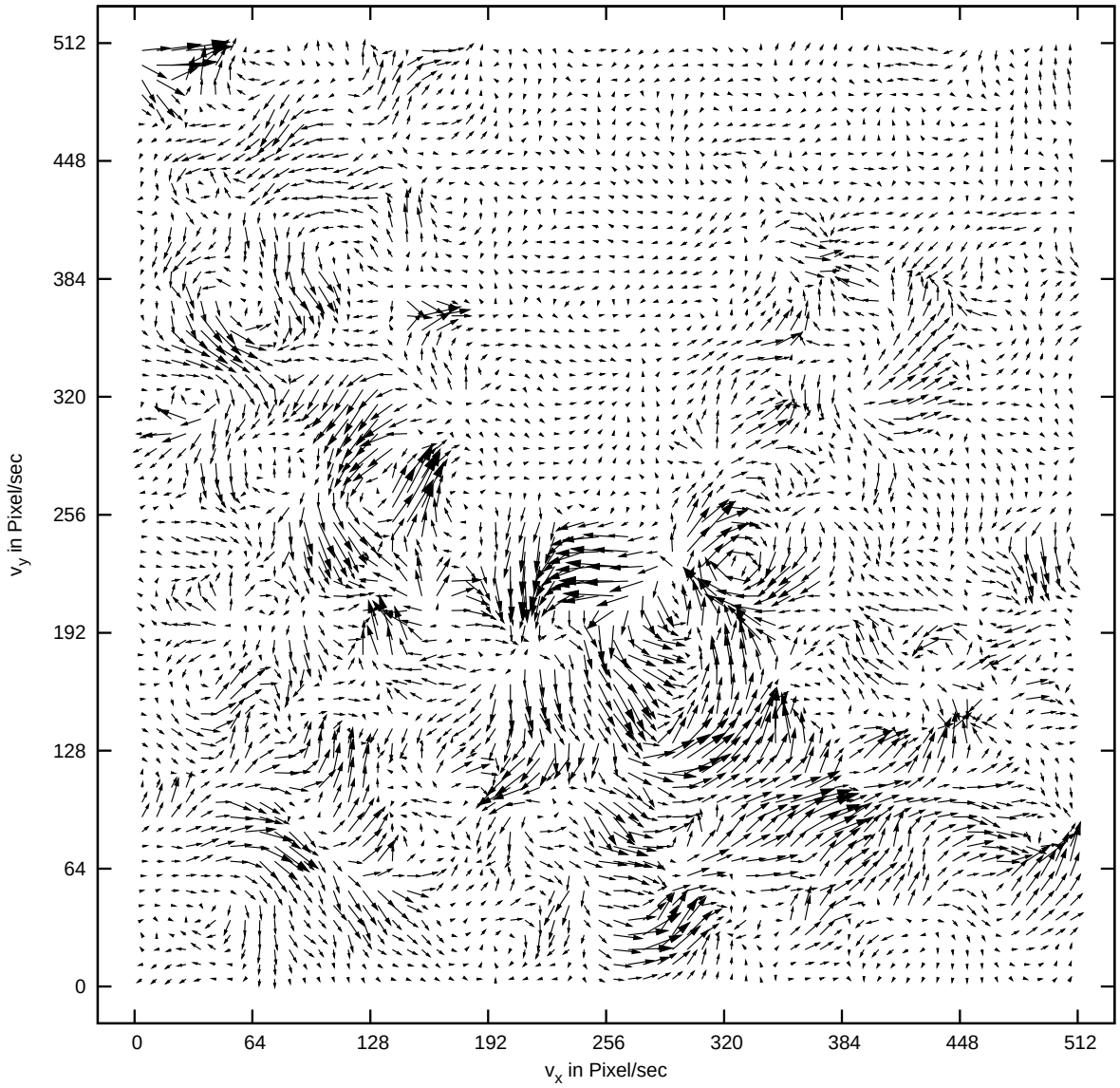

Abbildung 8.12: Ein Ergebnisvektorfeld von *PIVA* in der Testreihe 2 mit ersetzten Ausreißern. SOBEL-Hochpass- und Rauschfilter (Ansatz A) führen hier zu einer signifikanteren Ausprägung der Wirbelstrukturen im Gegensatz zur Analyse von ungefilterten Aufnahmen.

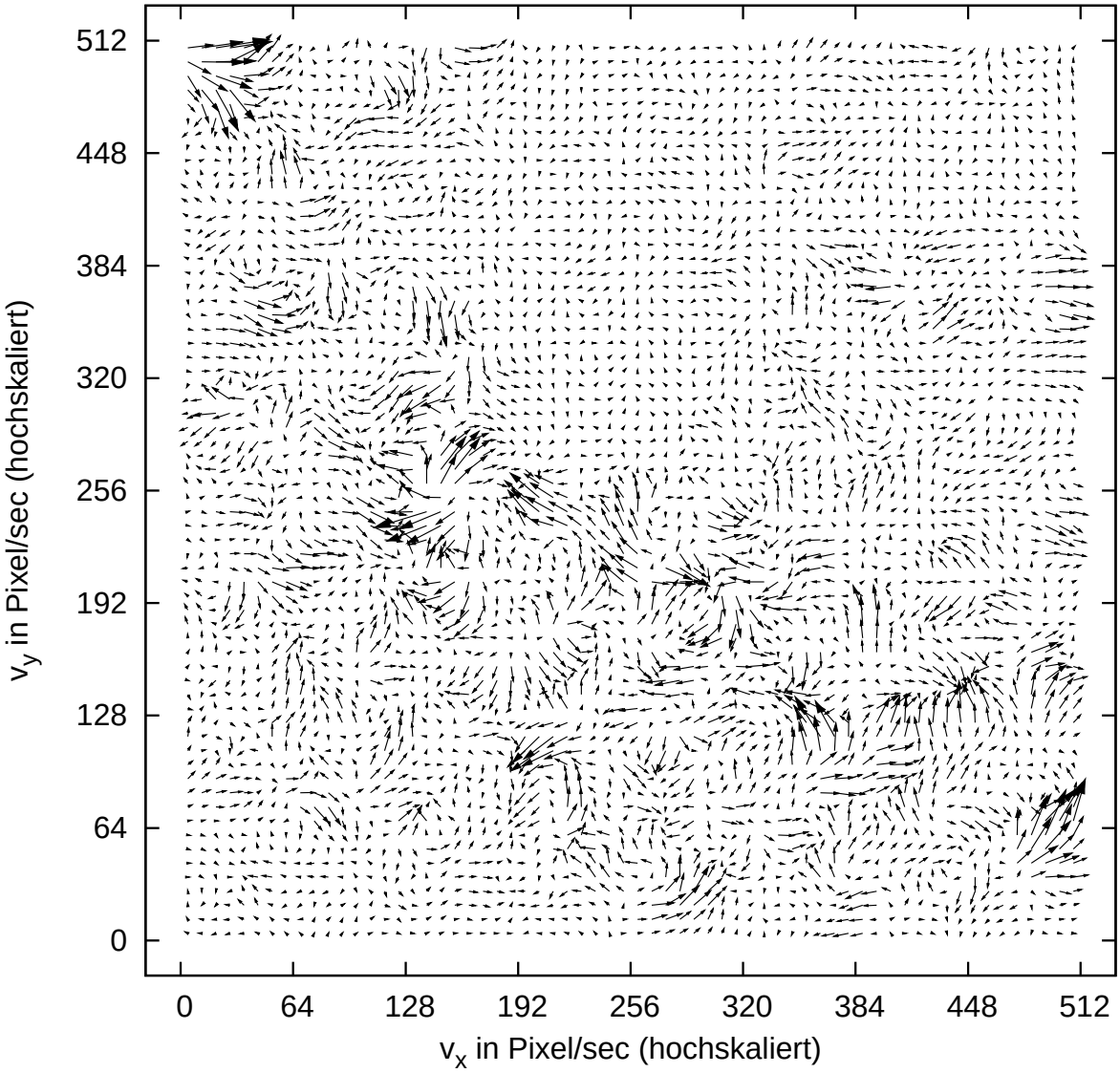

Abbildung 8.13: Das Differenzvektorfeld berechnet aus den Abbildungen 8.12 und 8.11. Die normalverteilten Fehler wurden hier aufskaliert. Die überproportionalen Ausschläge am Rand entstehen durch eine in der Software *PIVA* fehlende Spezialbehandlung der Aufnahmen am Rand und können unberücksichtigt bleiben.

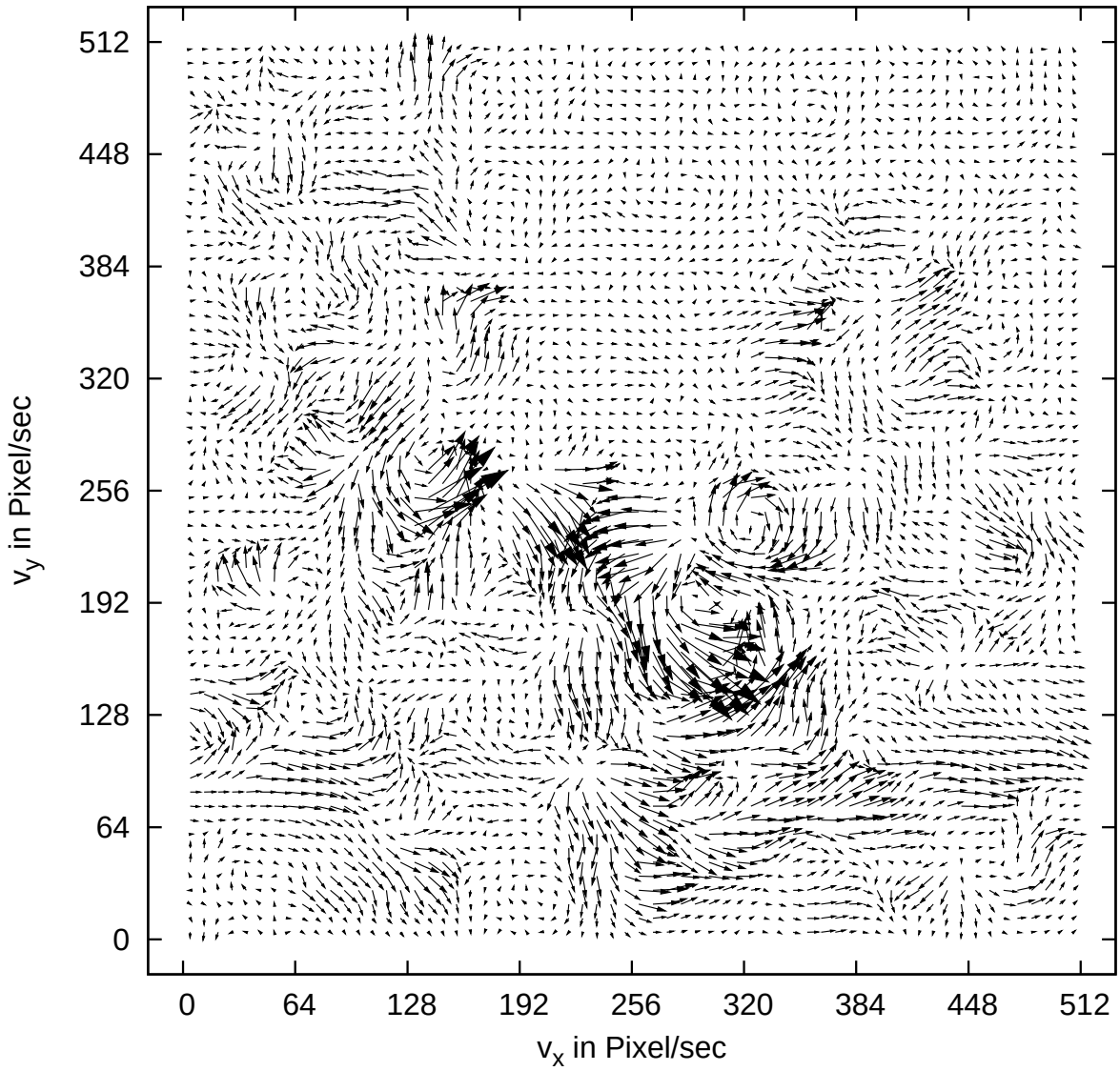

Abbildung 8.14: Das beste Ergebnisvektorfeld von *PIVA* in der Testreihe 2 mit ersetzten Ausreißern. Der Wavelet-Hochpassfilter (Ansatz B) führt hier zu einer stärkeren, formschöneren Ausprägung der Wirbelstrukturen im Gegensatz zur Analyse von ungefilterten und nach Ansatz A gefilterten Aufnahmen.

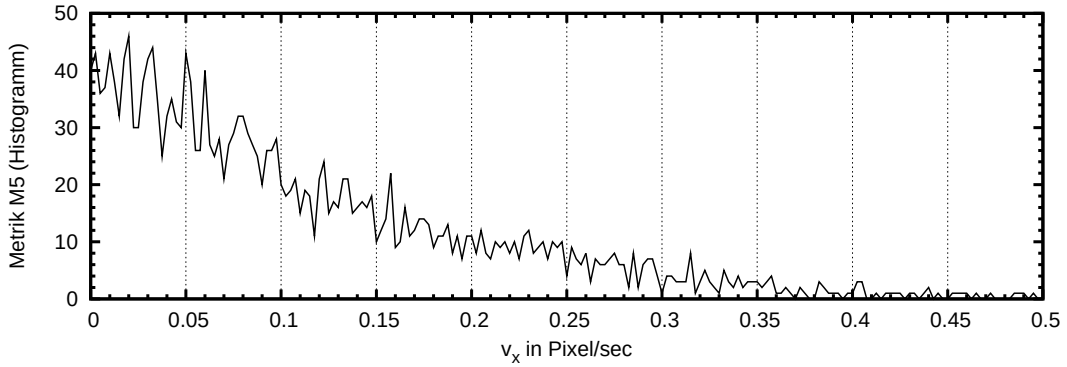

Abbildung 8.15: Das Histogramm über die x-Komponentenlängen des Vektorfeldes aus Abbildung 8.12 mit $\varepsilon = 0.125$, siehe M_5 in Tabelle 8.1. Der „peak-locking"-Effekt ist hier nicht direkt sichtbar. Eine Verzerrung aller Partikelbildverschiebungen gegen 0 ist jedoch anzunehmen.

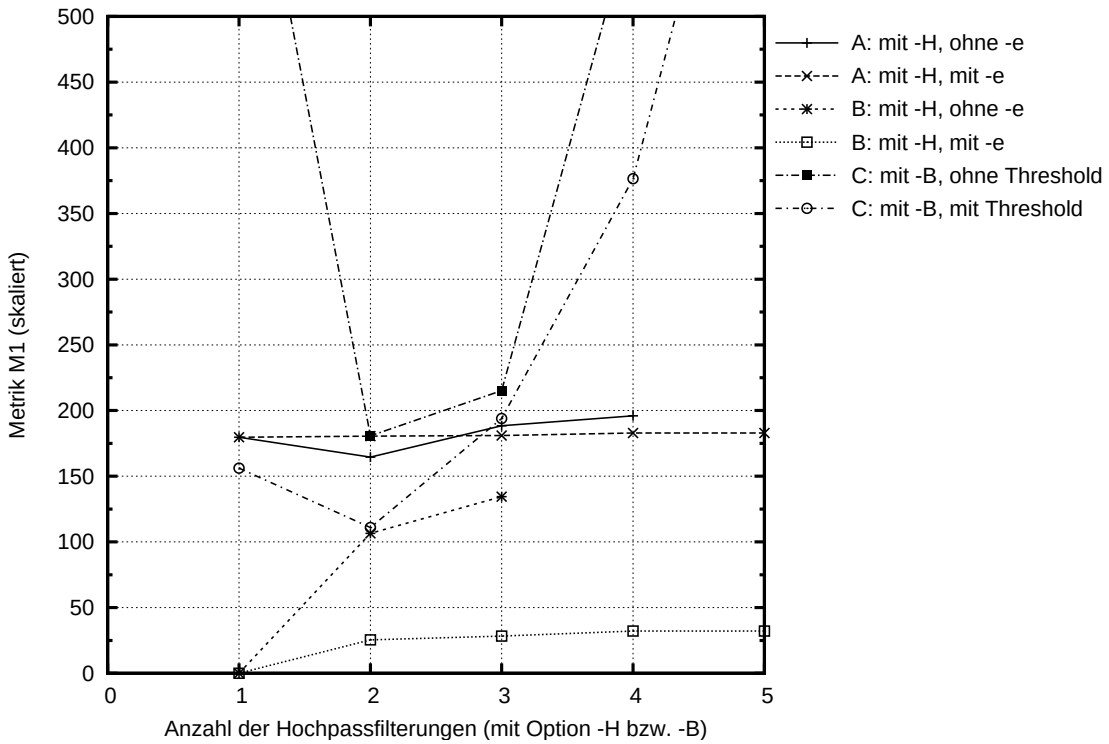

Abbildung 8.16: Die Aufzeichnung der Testreihe mit Hilfe der in Kapitel 8.1 definierten Metrik $\mathbf{M}_1$. Die mit **A** bezeichneten Kurven zeigen Vergleiche verschiedener Filterstufen mit einem herkömmlich erzeugten Referenzvektorfeld, die Kurven **B** zeigen Vergleiche mit einer einmalig hochpassgefilterten Referenz. Die Ergebnisse sind allerdings nur relativ zu betrachten, da kein absolutes Referenzvektorfeld vorhanden war. Die stark unterschiedlichen Wertebereiche der Kurven dürfen daher unberücksichtigt bleiben. Die nachvollziehbar geringsten Abweichungen zeigen nahezu alle Tests nach zweifacher Kantenfilterung. Das Kurvenpaar **C** zeigt die Anwendung des Wavelet-Kantenfilters (Ansatz B).

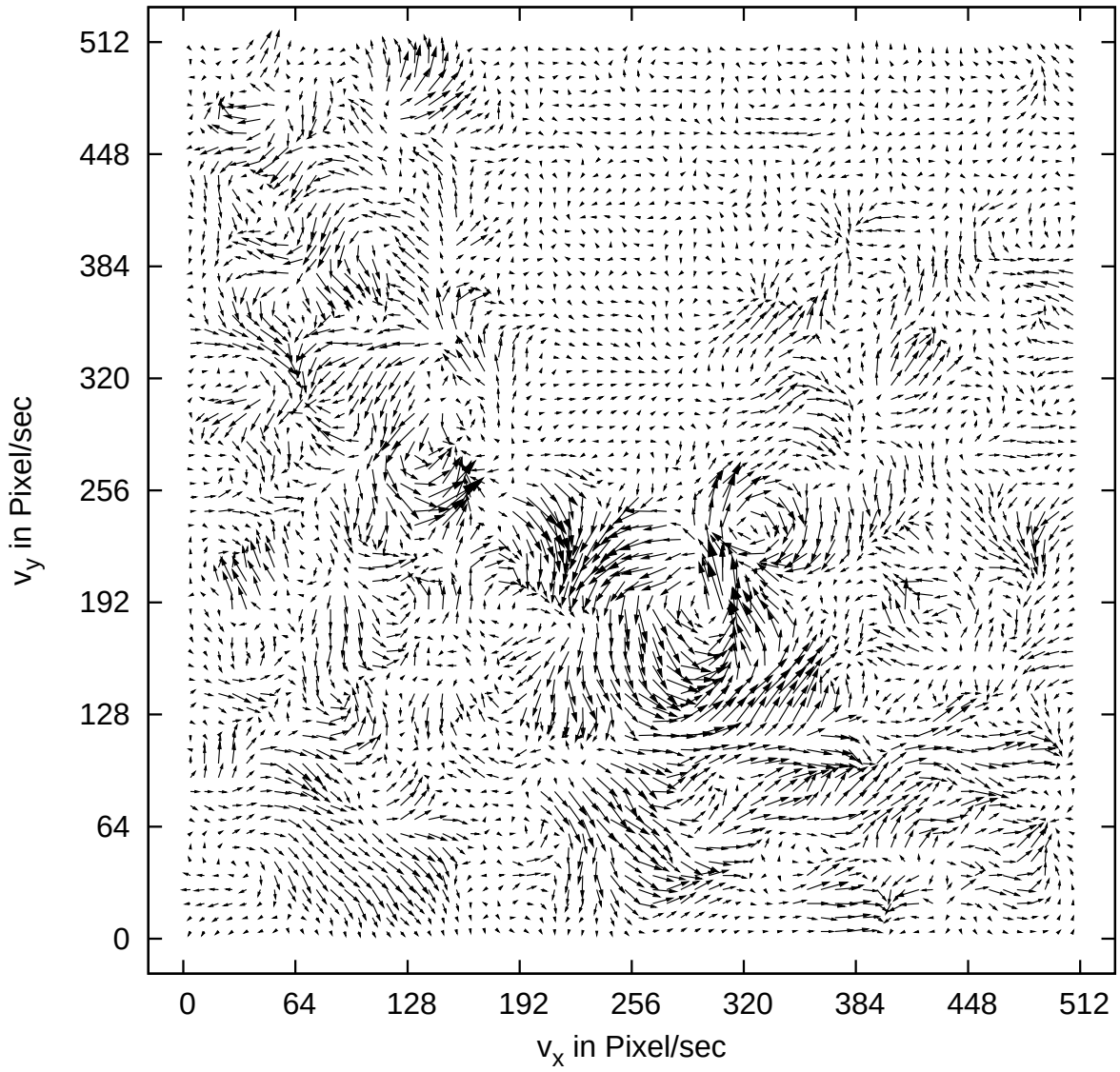

Abbildung 8.17: Ein Ergebnisvektorfeld von *PIVA* in der Testreihe 2 mit der iterativen CWS-Methode und durch Glättung ersetzten Ausreißern. Der Charakteristik der Wirbelstrukturen lässt sich sehr schön nachvollziehen, ebenso die im Gegensatz zu Abbildung 8.11 leicht größeren Vektoren als Resultat der CWS-Methode.

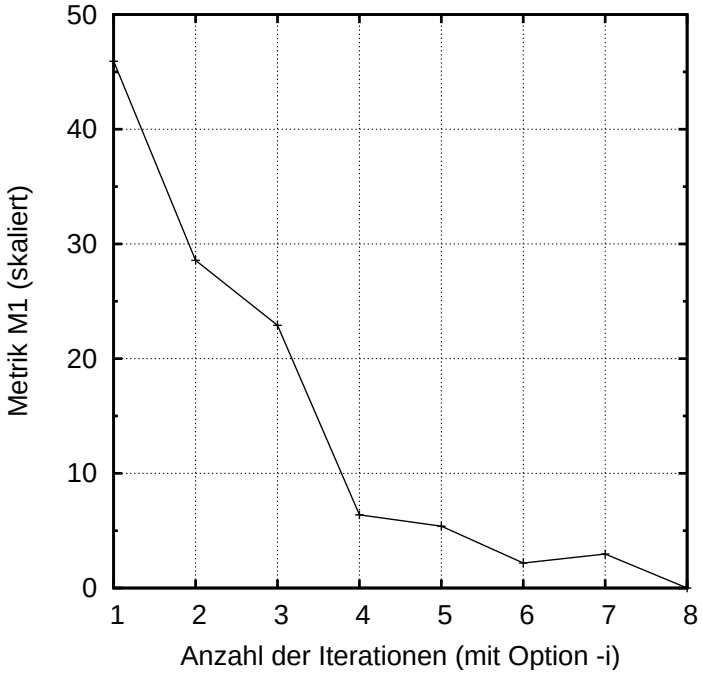

Abbildung 8.18: Ergebnis der kontinuierlichen Fensterverschiebetechnik auf den Aufnahmen biologischer Strukturen. Die Metrik $\mathbf{M}_1$ schwingt abklingend und konvergiert in einem kleinen Lösungsbereich um 0. Aus dem Histogramm 8.15 geht hervor, dass die durchschnittliche Verschiebung ≈ 0.04 Pixel beträgt und somit selbst bei Verwendung der CWS signifikante Fehler auftreten können. Ein Vergleich mit den Ergebnissen von *PIV View* liefert grundsätzlich einheitliche Ergebnisse. In diesen Skalen kann hier jedoch nicht eindeutig beobachtet werden, dass die CWS besser als die Kantenfilterung sei.

werden, was zu einer stärkeren Ausprägung der beobachteten Strukturen führt. Das Differenzvektorfeld 8.13 (= Abb. 8.12 – Abb. 8.11) zeigt den Unterschied zwischen diesen beiden Abbildungen noch einmal genauer. Die Strukturen und Strömungsmerkmale des Vektorfeldes 8.11 sind markant und zeigen die Flussrichtungen deutlich. Das Vektorfeld 8.14 auf Grundlage der Wavelet-Kantenfilterung zeigt betragsmäßig differierende Ergebnisse, wobei die beobachteten Wirbelstrukturen stets erhalten bleiben. Aus Tabelle 8.4 wird allerdings ersichtlich, dass der Vergleich der Wavelet-Filterung mit den Ergebnissen von *PIV View* die **besten Resultate** liefert, unter der Annahme, dass diese Software mit der kontinuierlichen Fensterverschiebung exakt arbeitet. Aus dieser Tabelle ist auch das global etwas bessere Ergebnis des eingesetzten Rauschfilters zu erkennen.

Das Diagramm 8.15 zeigt die Verteilung der Vektorlängen laut Metrik $\mathbf{M}_5$. Der „peak-locking"-Effekt ist daraus nicht ersichtlich. Es wird jedoch angenommen, dass diese Verteilung in Richtung 0 verzerrt ist. Aus dem Diagramm 8.16 kann man die Ergebnisse der Testreihen mit verschiedenen Systemparametern ablesen. Alle Werte sind relativ zueinander zu betrachten. Wie bei den künstlichen Bildern erreicht man auch hier die besten Resultate mit einmaliger Kantendetektion.

Da die durchschnittliche Verschiebung bei den Aufnahmen biologischer Strukturen weit unter einem Pixel liegt (≈ 0.04 Pixel), ist eine kontinuierliche Fensterverschiebetechnik (CWS) empfehlenswert, um genauere Ergebnisse erzielen zu können, welche nicht durch den „peak-locking"-Effekt verzerrt sind. In Abbildung 8.18 wurde dieser Ansatz ohne jegliche Filterung getestet. Dabei wurde die Metrik $\mathbf{M}_1$ auf eine Reihe von Ergebnisvektorfeldern bzgl. einer selbst erstellten Referenz (Systemparameter aus der Tabelle 8.3 mit der zusätzlichen Option `-i 8`) angewendet. Aus diesem Diagramm ist ersichtlich, dass das Ergebnis mit zunehmender Anzahl von Iterationen mit abklingender Schwingung einen eingeschränkten Lösungsbereich anstrebt. Signifikant bessere Ergebnisse konnten jedoch auch nicht erreicht werden, da selbst mit der CWS Genauigkeiten von **nur** ≈ 0.01 Pixel erreicht werden können (vergleiche dazu Histogramm 8.15). `PIV View` zeigt, dass sowohl die Ergebnisse einer CWS-Methode (siehe Abbildung 8.17) als auch die Filterung bei diesen Aufnahmen ähnlich gute Ergebnisse liefern, wobei jedoch im allgemeinen die CWS-Methode zu bevorzugen ist, da diese auf dem vollständigen Bildsignal arbeitet.

In der Abbildung 8.16 kann man ebenso sehen, dass der hier angewendete Rauschfilter einen leicht steigenden Einfluss auf die Abweichungen von der Referenz hat (Metrik M_1). Da keine absolute Referenz verfügbar ist, ist es nötig in diese Richtung noch weitere Überlegungen anzustellen bzw. Testexperimente durchzuführen.

In den Abbildungen 8.19 und 8.20 werden die durch die beiden in Kapitel 6.1 besprochenen Ansätze für Kantendetektoren erzeugten und als Grundlage für die PIV-Analyse verwendeten „Kantenbilder"-Hierarchien gezeigt. In der rechten Spalte kann man die Modulmaxima erkennen, in der linken Spalte sieht man die gesamtheitliche Aufzeichung der Helligkeitsgradienten.

Im Bild 8.21 kann man ferner die Eingriffsstellen des vorgeschlagenen Rauschfilters aus Abschnitt 6.2 sehen. Diese Stellen — gekennzeichnet durch schwarze Punkte — treten hier in relativ kleiner Zahl und Dichte auf, sodass sich die Rauschfilterung als nicht sehr effektiv gestaltet bzw. in den Bildern weniger Rauschen als angenommen auftritt. Die Ergebnisse mit Rauschfilterung liegen wie oben erwähnt trotzdem näher an den Ergebnissen von *PIV View*, sodass ein negativer Einfluss des Filters nicht beobachtet werden kann.

Als Gemeinsamkeit der beiden Hochpassfilter kann die guten Ergebnisse nach zweimaliger Kantenfilterung und die nachvollziehbar schlechteren Ergebnisse in gröberen Skalen

Testkonfiguration	$\langle \vec{r}_T^{(x)} \rangle$	$\langle \vec{r}_T^{(y)} \rangle$
Modi mit *PIVA*		
ohne Filterung	0.0249	-0.0165
mit Hochpass- und Rauschfilterung (Ansatz A)		
`-e -H 2`	0.0289	-0.0069
`-e -H 3`	0.0303	-0.0059
mit Wavelet-Filterung (Ansatz B)		
`-B 3, -S 4` ohne „Threshold“[a]	0.0351	-0.0094
`-B 3` mit „Threshold“	0.0350	-0.0051
mit (iterativer) kont. Fensterversch. `-w 2 -i 6`	0.0336	-0.0149
Modi mit *PIV View*		
mit kont. Fensterversch.	0.0340	-0.0096

[a] Hierbei handelt es sich um das quantitativ beste Resultat mit dem Einsatz von Filtern relativ zum Ergebnis von *PIV View*.

Tabelle 8.4: Die Ergebnisse bzgl. der Metrik M_2 ($\langle \vec{r}_T^{(x)} \rangle$ und $\langle \vec{r}_T^{(y)} \rangle$) im Vergleich zu den Ergebnissen von *PIV View*, welches die qualitativ hochwertigere Sub-Pixel-Fensterverschiebetechnik (CWS) einsetzt. Vor allem die Wavelet-Filterung zeigt relativ zu *PIV View* gute Ergebnisse, allerdings unter der Hypothese, dass *PIV View* selbst richtige Resultate liefert. Auch die Technik der kontinuierlichen Fensterverschiebung kann hier im Gegensatz zu den künstlichen Bildern keine besseren Ergebnisse erzielen.

nennen. Es zeigte sich, dass der von [Chr04] vorgeschlagene Eingrenzungsbereich („Threshold“) aus Teilabschnitt 4.3.2 für den korrelationsbasierten Bildvergleich nicht geeignet ist, da er für die Korrelation wichtige Informationen aus dem Bild entfernt. Dem entsprechend schneiden die Tests mit dem Wavelet-Filter ohne diese Eingrenzung besser ab. Die tendentiell besseren Ergebnisse mit dem Wavelet-Filter sind jedoch noch genauer nachzuprüfen. Ferner stellt sich die Frage, ob es sich lohnt, Aufnahmen überhaupt zu filtern oder stattdessen andere PIV-Methoden einzusetzen, um ähnlich gute Ergebnisse zu erzielen. Eine Erhöhung der Auflösung bei der Produktion der Aufnahmen, sofern möglich, würde die Bandbreite der registrierbaren Geschwindigkeiten ebenfalls vergrößern.

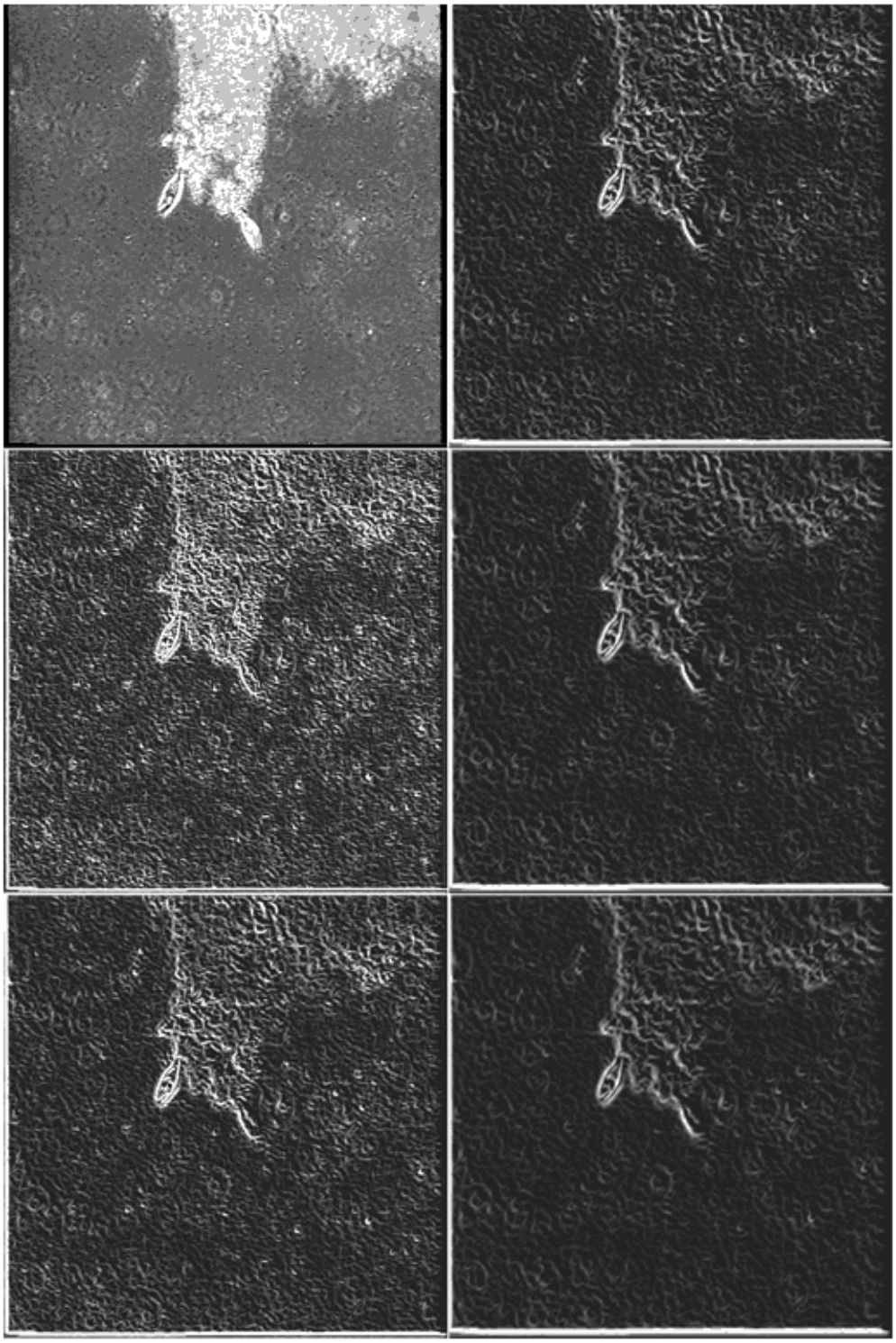

Abbildung 8.19: Die Kantenbildhierarchie mit Ansatz A in Testexperiment 2. **Links oben** befindet sich das Original (Option `-H 0`). In den hierarchischen Kantenbildern in den **beiden Spalten** kann man die Kantenstrukturen nach 1–5 Glättungen (`-H 1` bis `-H 5`) des Originals erkennen.

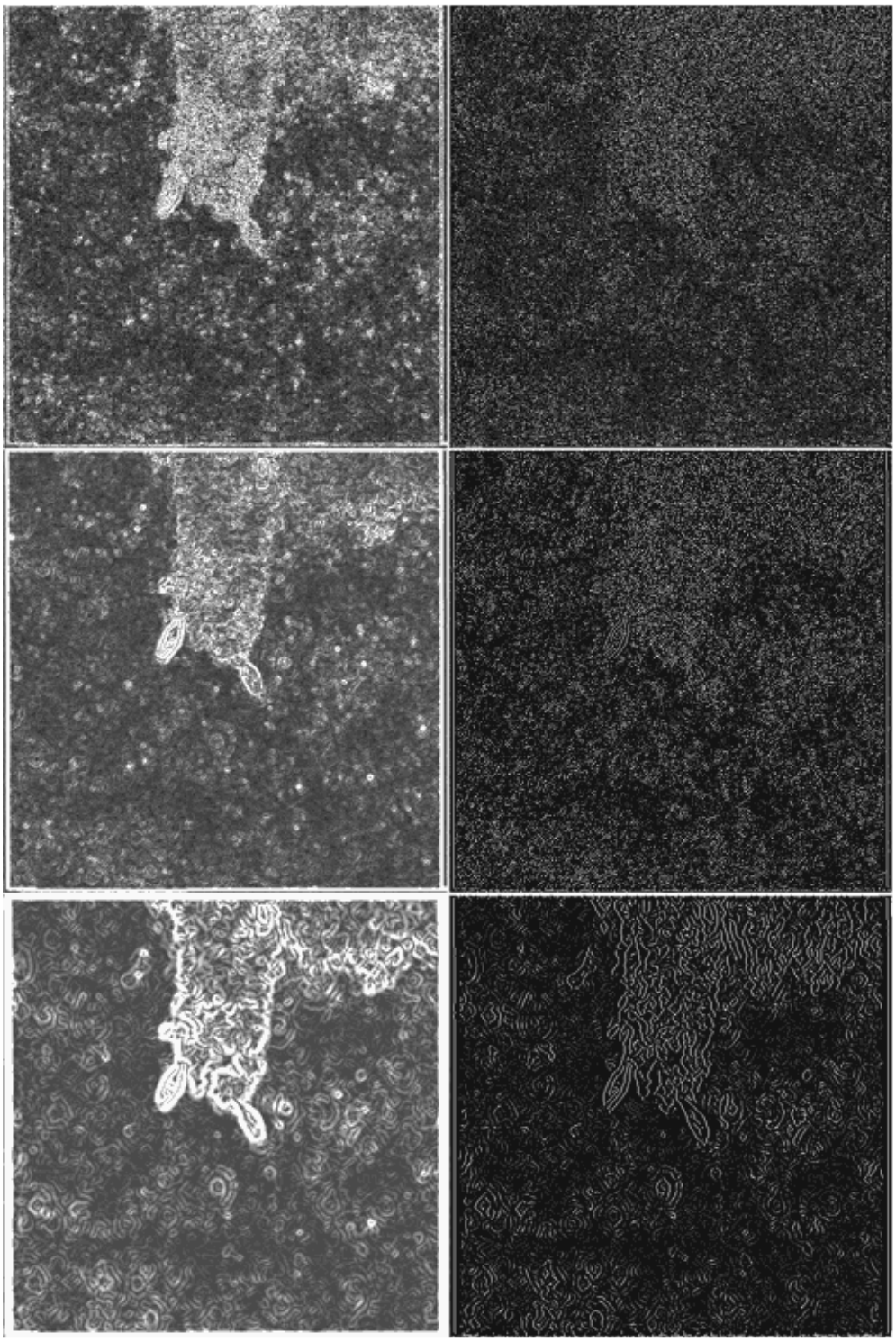

Abbildung 8.20: Die Kantenbildhierarchie mit Ansatz B in Testexperiment 2. In den hierarchischen Kantenbildern mit den Moduli der DWT in der **linken Spalte** kann man die Kantenstrukturen nach 1–3 Glättungen (`-B 1` bis `-B 3` und `-b 0`) des Originals aus Abbildung 8.10 erkennen. Die **rechte Spalte** enthält die dazu gehörenden Modulmaxima, welche scharfe Kantenverläufe darstellen.

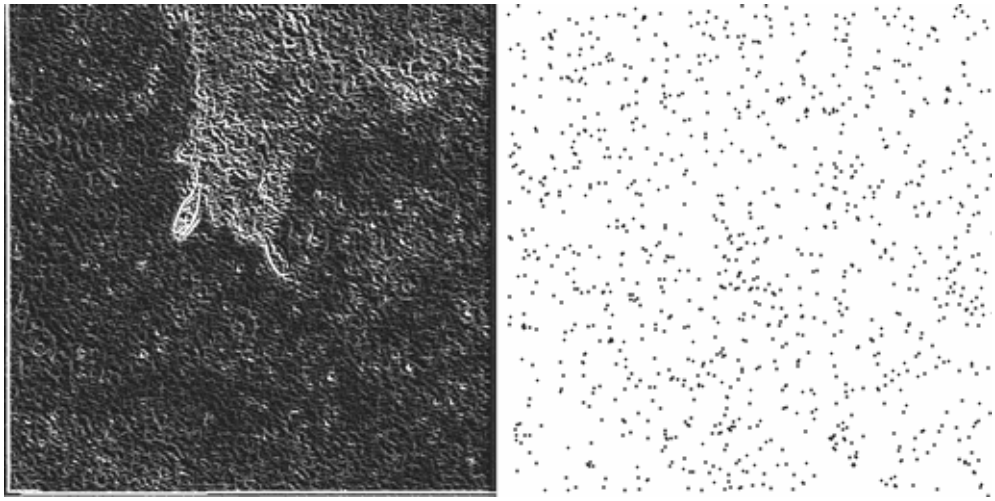

Abbildung 8.21: Gezeigt wird ein Vergleich von hochpassgefilterten Kantenbildern mit und ohne Rauschfilterung aus Teilabschnitt 6.2 in Testexperiment 2. **Links** wird die einmal hochpassgefilterte Aufnahme gezeigt. **Rechts** davon werden die Stellen gezeigt, wo jene Aufnahme auf der Grundlage einer dreischichtigen Kantenbildhierarchie rauschgefiltert wurde. Die als solche angenommenen Rauschspitzen zeigen sich wie erwartet nahezu homogen verteilt, tragen aber wie schon aus den Tests ersichtlich, aufgrund ihrer niedrigen Quantität kaum zu einer Verbesserung der Ergebnisse bei (siehe rechtes Bild).

9 Ausblick

Motto: „Niemals ist ein Thema als erschöpft oder eine Aufgabe als entgültig abgeschlossen anzusehen.“

Im Laufe der Bearbeitungszeit taucht man immer tiefer in Materie ein, befasst sich horizontal mit vielen Nebenproblemen aber auch vertikal mit Detailproblemen. Durch diesen Prozess entstehen meist Gedanken und Ideen, welche aber leider nicht alle bis ans Ende verfolgt werden können oder, um im Rahmen des Themas zu bleiben, auch nicht sollen. Darunter befinden sich auch nachträglich erkannte verbessernde und erweiternde Ansätze. Dennoch kann das hier Zusammengetragene für zukünftige Arbeiten wichtig sein. Ideen für den praktischen Teil der Diplomarbeit findet man in Teilabschnitt 9.1. Gedanken zum theoretischen Teil, sofern sie nicht schon in Kapitel 6 behandelt wurden, sind in Abschnitt 9.2 aufgelistet.

9.1 Verbesserung der Implementierung

Die folgende Auflistung behandelt grundsätzliche Vorschläge zur Verbesserung der entwickelten Software hinsichtlich der Erhöhung der PIV-Analysequalität aber auch im Hinblick auf internes Datenmanagement der Anwendung. Das Vorwissen eines Informatikers ist aufgrund der Fachausdrücke für die folgenden teils implementierungstechnischen Absätze sehr von Vorteil.

Datenstruktur für Vektorfelder — Aus den vielen Möglichkeiten wurde eine, wie schon erwähnt, für adaptive Verfahren geeignete, intuitive Baumstruktur ähnlich den „Quadtrees“ eingesetzt. Der große Unterschied zu den sogenannten „Quadtrees“ besteht darin, dass die Anzahl der Kinderknoten eines Knotens in jeder Baumebene unterschiedlich sein kann. Vorteile dieser Entscheidung sind

- effiziente Algorithmen auf geordneten Baumstrukturen, wie z. B. das Suchen, Löschen, Einfügen und
- ein signifikant geringerer Speicherbedarf durch eine nur regionale Verfeinerung der Vektorfeldauflösung.

Nachteile ergeben sich z. B.:

- Bei der Implementierung eines Algorithmus zur Ermittlung der 4er- oder 8er-Nachbarschaft von Knoten in adaptiven Bäumen. Selbst mit der Einschränkung, dass zwei Knoten derselben Baumebene gleich viele Kinder haben müssen, ge-

staltet sich eine Bestimmung der Nachbarschaft schwierig und in $O(T)$ mit der Baumtiefe T[1].

- Bei vielen anderen Algorithmen auf Bäumen.

Im Zuge genauerer Betrachtungen der PIV-Verfahren und der Implementierung hat es sich herausgestellt, dass der Speicherbedarf selbst bei großen Datenmengen im Gegensatz zu anderen Anwendungen relativ gering ist und von den meisten Rechnern mühelos bewältigt werden kann.

Der Vorschlag besteht darin, als Datenstruktur eine einfache Hierarchie von zweidimensionalen Feldern zu verwenden (z. B. `Array2D<Vector2D<double> >`).

Nachteile gestalten sich durch eine geringere Flexibilität bei der Auswahl der Fenstergröße in einer Verfeinerungsebene. Intelligente Implementierung kann diesen Nachteil aber kompensieren. Die **Ermittlung der Nachbarschaft** im hier betrachteten Sinne würde auf dieser Datenstruktur rechnerisch sehr einfach realisierbar sein (Modulo-Rechnung), jedoch im „Worst-Case“ auch $O(T)$ benötigen.

Verwendung von R-Bäumen — Wenn man aus bestimmten Gründen bei einer Baumstruktur bleiben will, könnte man über den Einsatz sogenannter **R-Bäume** nachdenken, die in Datenbanksystemen häufig Anwendung finden[2]. Für Bäume dieser Art gibt es umfangreiche Algorithmen-Bibliotheken.

Indizieren auf Bildern — Die interne Linearisierung der PIV-Aufnahmen könnte anstatt der einfachen zeilenweisen Linearisierung mittels raumfüllenden Kurven wie z. B. HILBERT- oder PEANO-Kurven erfolgen. Vorteile von Indizierungschemen mit raumfüllenden Kurven sind die im Speicher dichter beeinanderliegenden Werte für im Bild benachbarte Pixel (Regionalisierung). Dadurch erreicht man eine weit bessere „cache“-Nutzung, ebenso die Möglichkeit der Parallelisierung im Hinblick auf Echtzeitverarbeitung auf mehreren Prozessoren. Bei einer zeilenweisen Linearisierung ist die Anzahl der „cache misses“ und damit auch die Datenzugriffszeit je nach Systemarchitektur um einiges höher. Die Eingabebilder müssen für die Abbildung der raumfüllenden Kurven Abmessungen von 2er bzw. 3er-Potenzen besitzen oder dahingehend erweitert werden. Eine Lineariserung kostet anfangs Rechenzeit, muss aber für jedes Bild nur einmal berechnet werden. Auf die linearisierten Daten kann mit einer entsprechenden Funktion zugegriffen werden, welche kartesische Koordinaten (x, y) auf den jeweiligen Index i der raumfüllenden Kurve abbildet, was in $O(1)$ möglich ist. Diese Technik wird heute im Bereich der numerischen Strömungslöser schon rege angewendet.

FFT-KK — Die schon öfter erwähnte Möglichkeit der Verwendung der FFT zur Kreuzkorrelation in einem hybriden Algorithmus ist unerlässlich für ein effizientes Analysesystem. Die Nutzung einer externen Bibliothek vereinfacht die Implementierung.

Standardisiertes Ausgabeformat — Nebensächlich für die Thematik dieser Diplomarbeit aber wichtig für die Schnittstelle zu einer qualitativ hochwertigen Visualisierung ist

[1] Eine Recherche für Graphenbibliotheken spezialisiert auf Bäume wäre hier angebracht.

[2] Unter http://gi.cebitec.uni-bielefeld.de/teaching/2003winter/raumindexstrukturen/ findet man eine Übersicht über Indizierungsstrukturen für zwei- und n-dimensionale Räume, darunter auch R-Bäume, zusammen mit zahlreichen Literaturangaben.

das Ausgabeformat. Das TecPlot-Format[3] bietet hier Möglichkeiten zur strukturierten Speicherung von Vektordaten. Das in *PIVA* realisierte Format kann aber auch ohne Veränderung mit einem entsprechend konfigurierten *gnuplot* gelesen werden.

Verbesserte Analyse in den Randbereichen der Aufnahmen — Eine sehr einfache und naheliegende Möglichkeit wäre, ein Bild entsprechend (z. B. um eine Suchfenstergröße N_s) zu erweitern. Dadurch wäre eine unkomplizierte Fensterverschiebetechnik ohne Spezialfälle und ein vergrößertes Suchfenster auch am Rand möglich.

Erkennung von Ausreißern — Der implementierte „local-median"-Schätzer findet zwar Ausreißer, ersetzt jedoch fälschlicherweise oft gültige Vektoren in Bereichen wo die Varianz der Vektoren sehr klein wird. Das Kriterium zur Entscheidung zwischen Ausreißer und gültigem Vektor muss mit den Betrachtungen von [Wes93] genauer implementiert werden. Zur Erreichung einer schönen Visualisierung würde sich aber auch eine schlichte Glättung des Vektorfeldes durch Mittelwertbildung in 9er-Blöcken eignen, z. B. nach jeder Iteration mit Ausnahme der Finalen.

Intelligentere Vererbung von Schätzervektoren — Beim adaptiven Verfahrensmodus sollte man die Schätzervektoren einer gröberen Hierarchieebene nicht direkt (1 : 1) sondern z. B. durch bilineare Interpolation in die nächstfeinere Adaptionsstufe vererben. Dadurch gelangt man schneller zu einem guten Ergebnis und vermindert den Anteil der Ausreißer.

Weitere Probleme in der aktuellen Version der Implementierung

Das Kriterium für adaptive Verfeinerung muss hinsichtlich der Überlegungen in Abschnitt 5.3 reimplementiert werden. Die bisherige Implementierung ist noch relativ instabil und abhängig von den beobachteten Strömungsverhältnissen. Hinsichtlich eines vollständig frei wählbaren Überlappungsfaktors OF von Abfragefenstern müssen einige Teile der hierarchischen Implementierung noch flexibilisiert werden. Wenn Korrelationspitzen am Rand auftreten, wird dies bereits vom System erkannt. Reagiert wird bisher lediglich mit einem warnenden Hinweis. Es bietet sich an, das Suchfenster an diesen Stellen der Abfrage (wieder) zu vergrößern, um falsch ermittelte Werte für Verschiebungen zu vermeiden.

9.2 Gelerntes

Unabhängig von der Implementierung gibt es weitere Überlegungen, die im Laufe der Bearbeitungszeit entstanden. Aus Zeitgründen konnten diese aber nicht bis ins Detail verfolgt werden. Es sei dem Leser überlassen, diese Ideen zu beurteilen und, wenn nötig, weiter zu verfolgen oder zu begründen.

Beschleunigung der Suche der Korrelationsspitze — Das Suchen nach der höchsten Spitze in der Korrelationsebene könnte effizienter gestaltet werden, in dem man die Suche auf einen gewissen Bereich einschränkt, z. B. Suche in einem 3×3- anstatt eines 16×16-Bereichs. Die Größe des Bereichs kann man vom in der letzten Iteration gewonnenen, zusätzlichen Anteil des Verschiebungsvektors abhängig machen. Ist also $\vec{r} < 1$, so wird

[3] Ein offenes rohtextbasiertes Format zur Beschreibung von Geometriedaten; siehe `www.tecplot.com`.

sich die Korrelationsspitze sehr wahrscheinlich in einer 3×3-Umgebung im Zentrum der Korrelationsebene befinden. Damit kann auch die Detektion von falschen Spitzen bei schlechten Systemparametern eliminiert und damit der Anteil von Ausreißern verkleinert werden.

Adaptive Anpassung der Suchfenstergröße — Bei erwartungsgemäß kleinen Verschiebungen kann auch das Suchfenster verkleinert werden, analog verhält es sich umgekehrt; siehe [SR99]. Ebenso sollte man verfahren, wenn eine Korrelationspitze am Rand der Korrelationsebene gefunden wird. Nun kann es sein, dass es sich um eine falsche Spitze handelt, das gesuchte globale Maximum aber außerhalb des Korrelationsbereichs liegt. Folglich muss das Suchfenster vergrößert werden.

Selektive Wavelets zur gezielten Filterung der Aufnahmen — Ein weites Spektrum an Möglichkeiten bietet die Verwendung von Wavelets als digitale Filter und Dekompositionsfilternetze. Im Speziellen geht es um die Beseitigung des Rauschens in den Aufnahmen. Dazu ist es notwendig, die Charakteristik des Rauschens z. B. in biologischen Aufnahmen auch abhängig von den Aufnahmeeinrichtungen zu analysieren und einen entsprechenden Filter zu konstruieren. Es würde sich lohnen, in dieser Richtung mehr Forschungsaufwand zu betreiben. Die erzielten Ergebnisse mit dem in dieser Arbeit verwendeten Wavelet-Ansatz versprechen eine Verbesserung bei (extrem) verrauschten Bildern. Ob gegenüber herkömmlichen PIV-Methoden (z. B. Sub-Pixel-Fensterverschiebetechnik) entscheidende Verbesserungen erzielt werden können, muss jedoch noch genauer überprüft werden. Bei den Aufnahmen, welche in den Tests verwendet wurden, hat man mit herkömmlichen PIV-Verfahren Ergebnisse erzielt, die durch die beschriebene Signalfilterung nicht mehr maßgeblich verbessert werden konnten.

Glättung des Vektorfeldes zur Elimination von Ausreißern — Diese bereits in 9.1 beschriebene Idee muss näher beleuchtet werden. Eine Glättung führt je nach Art zur Veränderung einer großen Anzahl oder sogar aller Vektoren. Während die Änderung in der Regel gering bzw. sogar vorteilhaft sein kann, muss ein solches Vorgehen bei der Wahl der Abfragefenstergröße bzw. der Vektorfeldauflösung berücksichtigt werden. Zusätzlich ist ein geeignetes Glättungsverfahren zu wählen und zu untersuchen, ob eine intensive Glättung in späteren Iterationen ebenso vorteilhaft ist, wie in Früheren. Als Richtwert für ein deterministisches Verfahren könnte hier der von [Wes93] genannte „Valid Data Yield“ Γ genommen werden, welcher von der Partikeldichte N_I, vom „out-of-plane“- und vom „in-plane“-Partikelverlust (F_I bzw. F_O) abhängt.

Ein adaptiver Modus zur Verfeinerung des Vektorfeldes — Eine weitere Variante der adaptiven Verfeinerung wäre Folgende: In jedem Schritt verfeinert man auf dem gesamten Gebiet. Schätzervektoren aus höheren Ebenen des Baumes werden aber nur dort weitervererbt, wo laut Adaptivitätskriterium aus Teilabschnitt 5.2.1 die Wirbelstärke kleiner bzw. die Krümmung größer als ein bestimmter Wert sind. Dann wird die Suchfenstergröße für die Kinderknoten in der neuen Ebene entsprechend angepasst. Wenn Vektoren an Stellen linearer Strömung vererbt werden, dann kann das Suchfenster entsprechend verkleinert werden, da man dort mit der Fensterverschiebetechnik arbeiten kann. An den Stellen mit großer Wirbelstärke weiß man Schätzer und damit die für

Bestimmung der Suchfenstergröße wichtige Geschwindigkeitsverteilung. Über die Effizienz entscheiden die Implementierung und die gewünschten Ergebnisse.

Kontinuierliche Fensterverschiebung — Die Ergebnisse der Tests zeigen, dass in den betrachteten Aufnahmen nicht soviel störendes Rauschen enthalten ist wie anfangs angenommen. Deshalb liegt es nahe, eine kontinuierliche Fensterverschiebetechnik mit geeigneter Interpolation der diskreten Grauwerte anzuwenden, da die Werte der Partikelbildverschiebungen der Aufnahmen in Abschnitt 8.3 für eine herkömmliche PIV-Analyse unüblich klein sind (durchschnittlich ≈ 0.04 Pixel). Diese Empfehlung kann bei Aufnahmen medizinisch-biologischer Strukturen in der μ-PIV gemacht werden.

Optimierung des Geschwindigkeitsbereichs — Es hat sich auch herauskristallisiert, dass zu kleine Geschwindigkeiten, die teilweise sogar unter der erreichbaren Genauigkeit der PIV (mit guten Methoden $\approx 0.04(-0.01)$ Pixel) liegen (vergleiche Histogramm 8.15), grundsätzlich schlechtere Analyseergebnisse liefern. Oft wird eine Reproduktion der Aufnahmen — eine Wiederholung des (gesamten) Experiments — mit möglichst hohen Auflösungen bzw. besserer Qualität unvermeidbar sein.

Partikelmusterdeformation höherer Ordnung — Aus Zeitgründen konnten die in Kapitel 2.1.3 veranschaulichten Ansätze nicht mehr in *PIVA* implementiert werden. Nichtsdestotrotz sind diese Techniken bei entsprechenden Geschwindigkeitsgradienten und Strömungsverhältnissen maßgeblich für gute Ergebnisse und sollten in guten Analysesystemen zum Einsatz gelangen. Ob ähnliche Qualitätszunahmen auch in μ-PIV-Geschwindigkeitsskalen ($\|\vec{r}\|$ oft $\ll 1$ Pixel) erreicht werden können, muss noch genauer untersucht werden.

Über den Rauschanteil in digitalen Aufnahmen — Die Ermittlung des Rauschanteils in digitalen Bildern ist oft schwierig, da dieser von vielen Faktoren abhängt. Vorerst gilt es das störende Rauschen zu definieren. Die in den letzten Abschnitten besprochenen PIV-Methoden sollten in weiteren Testreihen mit extrem verrauschten Bildern getestet werden. Dabei würde der Vergleich einer PIV-Analyse von Aufnahmen mit künstlich generiertem Rauschen (z. B. per Zufallsgenerator) und durch schlechte Aufnahmeeinrichtungen erzeugtem Rauschen ein interessantes Ergebnis darstellen.

9.3 Schlussfolgerungen

In der langen und ausführlichen Recherche hat sich der Eindruck vermittelt, dass Techniken der heute weit fortgeschrittenen digitalen Signal- und Bildverarbeitung nur selten in einschlägigen Artikeln einbezogen werden. Ähnlich ist es mit hierarchischen, adaptiven Algorithmen zur regional optimierten Rekonstruktion von Vektorfeldern. Motiviert durch diesen Tatbestand erscheint es notwendig, in diese Richtung weiterzuforschen. Diese Arbeit bietet mit ihren Ergebnissen hinsichtlich der Verbesserung der Analysequalität durch das Filtern von Rauschen im Bildsignal und der gezielten Extraktion und Beobachtung von PIV-Merkmalen eine Grundlage dafür. Da der Schwerpunkt auf der Zusammenfassung von PIV-Methoden und der Implementierung eines Analysesoftwaresystems lag, konnte im Fachbereich digitale Bildverarbeitung nicht hinreichend in die Tiefe gegangen werden. Folglich ist eine genauere Betrachtung der Bildsignalklassen und der Zusammenhang der

verschiedenen Rauschcharakteristiken mit den für die PIV wichtigen Merkmalen nötig, um von vornherein einen entsprechenden, neuen Analyseansatz entwerfen zu können.

Die relativ schlechten Ergebnisse in den Aufnahmen aus Testexperiment 8.3 sind vor allem darauf zurückzuführen, dass der Anteil des störenden Rauschens überschätzt wurde und der Geschwindigkeitsbereich unüblich klein für eine PIV-Analyse ist. Tests haben gezeigt, dass es nötig ist, genauere frequenzanalytische Überlegungen anzustellen, um signifikant bessere Ergebnisse zu erzielen; z. B. die Konstruktion von merkmalserhaltenden Wavelet-Filtern passend zur betrachteten Bild- und Rauschklasse. Zusätzlich empfehle ich weitere Experimente mit gezielt verrauschten Aufnahmen zur einwandfreien Validierung der konstruierten Filter.

Des weiteren zeigt sich der Fachbereich der partikelbasierten Strömungsvisualisierung als sehr weit verzweigt, sodass es angebracht erschien, einen Überblick der gängigsten Methoden zu erstellen. In vielen Artikeln wird in teils sehr unterschiedliche Richtungen gearbeitet. Dies führt u. a. dazu, dass schon lang etablierte Methoden etwas in den Hintergrund geraten, obwohl diese einmal grundlegend durchdacht worden sind. Mit dieser Schrift liegt nun ein einführender, globaler und bewertender Überblick über gängige PIV-Verfahren und ein Vorschlag für eine flexible Architektur zur Entwicklung eines softwarebasierten Analysesystems vor. Vielleicht kann diese Schrift als Anhaltspunkt dazu beitragen, die Festlegung einer kleinen Anzahl von möglichst generischen PIV-Standardmethoden und Verfahrensvarianten zu erleichtern.

Ich möchte noch einmal etwas tiefer in die Materie eintauchen und festhalten, dass folgender Tatbestand durch andere Quellen (siehe z. B. [Pus00]) bestätigt wird: Eine FFT-KK zur Erreichung hoher Geschwindigkeiten in der Analyse ist nicht mit der Genauigkeit einer D-NKK zu vergleichen. Kontinuierliche Fensterverschiebung und Fensterdeformation n-ter Ordnung tragen unter entsprechenden Bedingungen (z. B. passende Bandbreite der Partikelbildverschiebungen) ebenfalls entscheidend zur Verbesserung der Genauigkeit bei. Dennoch ist eine gut implementierte, mit den hier behandelten Ansätzen kombinierte FFT-KK ausreichend für gute Ergebnisse in der hier geforderten Qualität der Visualisierung. PIV-Analysen sind in den meisten Fällen mit einer empirischen Herangehensweise zur Ermittlung idealer Parameter für eine individuelle Beobachtung verbunden. So kann mit entsprechendem Vorwissen — weitgehend unabhängig von der Methodik — die Qualität der Ergebnisse entscheidend verbessert bzw. bei Unkenntnis drastisch verschlechtert werden.

Einige Literaturvorschläge zur Vertiefung und Fortsetzung der Arbeit

Dem interessierten Leser kann ich nach meinen Recherchen einige Artikel nennen, die meiner Meinung nach die Richtung vertiefender Nachforschungen angeben. Als Referenz für die kontinuierliche Fensterverschiebetechnik sei der Artikel [GW02] empfohlen. Die Analysemethode von [Har00] solle die PIV sehr effizient gestalten. Dieser Artikel stützt sich auf die früheren Arbeiten [Har96] und [Har98a]. Ein genauere Beleuchtung dieser Methoden wäre zu erwägen. Ein Vergleich der Ergebnisse mit denen der *PASTIS*-Methode aus [FD00] wäre ebenso interessant. Zur Vertiefung der hierarchischen und adaptiven Ansätze der PIV möchte ich die beiden Artikel [NLRR02] und [Use99] empfehlen.

A Glossar

Dieser Glossar enthält zu einem großen Teil Akronyme und Erklärungen aus der englischen Sprache. Eine Übersetzung in das Deutsche ist hier nicht notwendig und teils auch nicht sinnvoll, da in einschlägigen Fachkreisen hauptsächlich englische Begriffe verwendet werden.

AR-WIDIM — Die WIDIM-Methode mit dem Ansatz der „**A**daptive **R**esolution“.

Bildmerkmal — Eine Farb- und/oder Helligkeitsveränderung in einem bestimmten Bereich, oft als Kante bezeichnet. Im Gegensatz dazu gibt es das →**Strömungsmerkmal**.

CCD — Ein „**C**harge **C**oupled **D**evice“ ist ein Element zur elektronischen Erfassung von Lichtstrahlen und wird häufig in digitalen Bilderfassungssystemen verwendet.

CDI — **C**entral **D**ifference **I**nterrogation (Scheme).

CDIC — „**C**entral **D**ifference **I**mage **C**orrection“, verwendet →**CDI** als Grundlage.

CFD — **C**omputational **F**luid **D**ynamics.

CIV — **C**orrelation **I**mage **V**elocimetry.

CWS — „**C**ontinuous **W**indow **S**hift“-Methode; →**Fensterverschiebetechnik**.

DFT — Diskrete FOURIER-Transformation.

D-KK — →**D-NKK**, lediglich ohne Normalisierungsterm und im Rahmen dieser Arbeit als Kreuzkovarianz bezeichnet (siehe Statistikexkurs auf Seite 12 und →**KK**).

D-NKK — **D**irekte, **n**ormalisierte **K**reuz**k**orrelation. Die →**D-KK** wäre dann eine nicht normalisierte Kreuzkorrelation, auch bezeichnet als Kreuzkovarianz.

DMV — **D**irect **M**easurement of **V**orticity.

DNS — „**D**irect **N**umerical **S**imulation“ oder direkte numerische Simulation.

DPIV — **D**igital Particle Image Velocimetry.

(D)WT — (**D**iskrete) **W**avelet-**T**ransformation.

FDI — **F**orward **D**ifference **I**nterrogation (Method).

Fensterverschiebetechnik — Meist als „Window Offset Method“, „Window Displacement Method“ oder „Window Shift Technique“ bezeichnet; →**WIDIM**.

FFT — Schneller rekursiver Algorithmus zur Berechnung der →**DFT**.

FFT-KK — **K**reuz**k**orrelation basierend auf der →**FFT**.

GP — Eine GAUSS'sche **P**yramide ist eine mehrstufige Vergröberung der Auflösung von Bildern.

HPV — **H**igh resolution **P**article **V**elocimetry.

HWA — **H**ot-**W**ire **A**nemometry.

IPAN — **I**mage and **P**attern **An**alysis group.

KK — **K**reuz**k**orrelation.

(Kreuz-)Korrelationssignal — Diese zweidimensionale Funktion setzt sich aus allen Funktionswerten der →**KK** (Kreuzkorrelationskoeffizienten) zusammen, welche durch die Verschiebung des Abfragefensters über dem Suchfenster erzeugt werden.

LDA — **L**aser **D**oppler **A**nemometry.

LSV — **L**aser **S**peckle **V**elocimetry.

MQD — **M**inimum **Q**uadratic **D**ifference.

NYQUIST-SHANNON **Abtasttheorem** — Aus [Wik] lässt sich folgende kurze und anschauliche Erklärung zitieren: „Das NYQUIST-SHANNON'sche Abtasttheorem besagt, dass ein kontinuierliches Signal mit einer Maximalfrequenz f_{max} mit einer Frequenz größer als $2 \cdot f_{max}$ abgetastet werden muss, damit man aus dem so erhaltenen zeitdiskreten Signal das Ursprungssignal ohne Informationsverlust wieder rekonstruieren kann."

ODP — **O**rthogonal **D**ynamic **P**rogramming.

OPIV — **O**ptical Particle Image Velocimetry.

Partikel — In eine Flüssigkeit eingestreute Materie, z. B. Öl- oder Wasserstaub, feines Kunststoffgranulat.

Partikelbild — Hierbei handelt es sich um das (digitale bzw. diskretisierte) Abbild eines einzigen beleuchteten →**Partikel**s auf einer Aufnahme einer Bilderfassungseinrichtung. Der Partikel selbst kann dabei in der Realität eine andere Form und Größe besitzen.

Partikelbilddichte — Anzahl der →**Partikelbild**er innerhalb einer Flächeneinheit, gemittelt über einem größeren Gebiet.

Partikelbilddurchmesser — Mittlerer Durchmesser der →**Partikelbild**er.

Partikelbildmuster — Räumlicher Zusammenschluss einer Gruppe von →**Partikelbild**ern.

Partikelbildspur — Zeitlicher Zusammenhang mehrerer Abbilder desselben →**Partikel**s.

Partikelbildverschiebung — Durch die →**KK** aus zwei Aufnahmen ermittelte Verschiebung zweier, korrespondierender →**Partikelbildmuster**.

PASTIS — **P**eak **a**nti-aliasing **s**pline **t**ransformed **i**nterrogation **s**cheme.

PID — **P**article **I**mage **D**istortion.

PIPM — **P**article **I**mage **P**attern **M**atching.

PIV — „Particle Image Velocimetry“ oder auch „**P**article **I**maging **V**elocimetry“.

PIVA — „Particle Image Velocimetry **A**nalyser“; im Rahmen dieser Diplomarbeit entwickeltes Softwaresystem zum Testen und Vergleichen verschiedener Ansätze aus der PIV.

POD — Bei der „**P**roper **O**rthogonal **D**ecomposition“ handelt es sich um eine vielfach in der Fluidmechanik (-datenanalyse) angewendete, elegante Methode zur Reduktion eines komplexen N-dimensionalen Problems auf ein Einfacheres der Dimension $n \ll N$. Die POD-Methode wird auch als „Principal Component Analysis“ (Hauptachsenanalyse) bezeichnet. Eine gute Einführung findet sich unter [Cha00].

PTV — **P**article **T**racking **V**elocimetry.

Stereo-PIV — →**PIV** mit zwei oder mehreren Bilderfassungsgeräten zur Ermittlung der dritten Geschwindigkeitskomponente.

Strömungsmerkmal — Im Gegensatz zum →**Bildmerkmal** gelten in einer strömenden Flüssigkeit z. B. Wirbel, Turbulenzen und Grenzschichtströmungen als wichtige Merkmale.

STL — Die „**S**tandard **T**emplate **L**ibrary“ befindet sich seit langem in der C++ Standard Bibliothek und bietet eine Sammlung von generischen Kontainerstrukturen verwendet als Grundlage in nahezu allen in C++ programmierten Systemen. Entwickelt wurde diese Klassenbibliothek von Silicon Graphics (SGI); siehe [Sil04].

WIDIM — **W**indow **D**isplacement **I**terative **M**ultigrid.

B Erklärung der Symbole und Operatoren

Die folgende Tabelle erklärt die wichtigsten in dieser Schrift verwendeten Symbole und Operatoren.

Symbol/ Operator	Erklärung
$\uparrow p$	Der sogenannte „Upsampling"-Operator fügt Nullen in eine Sequenz $(x)_q$ ein und erzeugt somit nach der Vorschrift $$(x \uparrow p)_n = \begin{cases} x_q, & \text{wenn} \quad n/p = q \in \mathbb{Z} \\ 0\ , & \text{wenn} \quad n/p \notin \mathbb{Z} \end{cases}$$ die Sequenz $(x)_n$ mit $n \in [0,N]$ bzw. $q \in [0...Q]$, dem q-ten Element x_q der Sequenz und den Sequenzlängen $N = \lvert(x)_n\rvert$ bzw. $Q = \lvert(x)_q\rvert$.
$\langle\rangle$	Das arithmetische Mittel aller Werte
$\lvert\cdot\rvert$	Üblicherweise der Betrag einer Zahl
$\lVert\cdot\rVert$	EUKLID'sche Vektornorm berechnet sich durch $\lVert\vec{v}\rVert = \sqrt{v_x^2 + v_y^2}$ mit $\vec{v} = (v_x, v_y)^T$
$\lVert\cdot\rVert^2$	Quadrat der EUKLID'schen Vektornorm
δ	KRONECKER-Symbol oder DIRAC-Filter, oft auch δ_{ij}
$\vec{\omega}$	Wirbelstärke, Rotationsvektor oder „vorticity vector"
σ	Standardabweichung
σ^2	Varianz
$\vec{0}$	Der generische Null-Vektor
$\arg z$	Das Argument einer komplexen Zahl $z = a + \mathrm{i} \cdot b$ ist ihr Winkel zur Horizontalen mit $-\pi < \arg z \leq \pi$ und lautet $$\arg z = \begin{cases} \arccos(\frac{a}{\lvert z\rvert}), & \text{wenn} \quad b \geq 0 \\ -\arccos(\frac{a}{\lvert z\rvert}), & \text{wenn} \quad b < 0 \end{cases}.$$
d_t	Der mittlere Partikelbilddurchmesser in einer Aufnahme
$\mathcal{B}$	Menge aller Pixelpositionen in einer Aufnahme

Tabelle B.1: Erklärung der Symbole und Operatoren (Fortsetzung auf nächster Seite)

Symbol/ Operator	Erklärung
$\hat{\mathcal{B}}$	Menge aller Pixelpositionen innerhalb einer Rasterzelle (eines Abfragefensters)
$f * g$	Die Konvolution oder das Faltungsintegral berechnet sich durch $$(f*g)(t) \equiv \int_0^t f(\tau)g(t-\tau)d\tau.$$ Im Diskreten berechnet man die Faltung mit $$(f*g)(t) \equiv \sum_{0\leq i\leq t} f(i)\cdot g(t-i).$$
$\mathcal{F}$	FOURIER-Transformation
I	Erste Aufnahme zur Zeit t, Bilddatenfeld mit 8-Bit Grauwerten
$\hat{I}$	Abfragefenster („interrogation window“)
II	Deformationstensor
N	Größe des quadratischen Abfragefensters
N_s	Größe des quadratischen Suchfensters
$o(f(n))$	LANDAU-Symbol für die Menge aller Funktionen mit einer Komplexität echt kleiner $f(n)$.
$O(f(n))$	LANDAU-Symbol für die Menge aller Funktionen mit einer Komplexität kleiner oder gleich $f(n)$.
$\vec{r}$	Oft als reeller Verschiebungsvektor verwendet
$R_{\hat{I}\hat{S}}$	Kreuzkorrelationskoeffizient
S	Zweite Aufnahme zur Zeit $t+\Delta t$
$\hat{S}$	Suchfenster („search window“)
$\mathrm{sign}(x)$	Die Signum- oder „Vorzeichen“-Funktion berechnet sich durch $$\mathrm{sign}(x) = \begin{cases} 1, \text{ wenn } & x \geq 0 \\ -1, \text{ wenn } & x < 0 \end{cases}.$$
∇U	Der Nabla-Operator, hier o. B. d. A. zweidimensional für eine Funktion U gleichzusetzen mit dem Gradienten grad $U = \begin{pmatrix} \partial U/\partial x \\ \partial U/\partial y \end{pmatrix}$
$\vec{v}$	Meist als reeller Geschwindigkeitsvektor verwendet
z_{rms}	Der diskrete Effektivwert bzw. „Root-Mean-Square“ von z berechnet sich durch $z_{rms} = \sqrt{\frac{1}{N}\sum_{z_i\in Z} z_i^2}$

Tabelle B.1: Erklärung der Symbole und Operatoren

C Wichtige Internetadressen

Die folgende Liste enthält aus der Recherche hervorgegangene Adressen aus dem Internet, die sich teils eingehend und anschaulich mit der „Particle Image Velocimetry“ befassen. Die Aktualität dieser Referenzen kann sich seit der Erstellung dieser Schrift allerdings verschlechtert haben.

- Projekt **PIV Standard Image** der *Visualization Society of Japan*, http://vsj.or.jp/piv/
- Projekt **PIV-Challenge**, http://www.pivchallenge.org
- Projekt **PivNet 2**, http://pivnet.sm.go.dlr.de
- Projekt **EUROPIV 2**, http://www.univ-lille1.fr/europiv/
- Forschungsgruppe **URAPIV**™, http://urapiv.tripod.com
- Forschungsgruppe im Bereich **DPIV** des Von-Karman-Institutes für Fluiddynamik in Belgien, http://www.vki.ac.be/research/themes/mt/dpiv.html
- Forschungsgruppe **μ-PIV** an der Universität von Florida, http://grove.ufl.edu/~jnchung/piv/micropiv.html
- Forschungsgruppe für Mikrofluidmechanik und **μ-PIV** an der NTNU (Norwegen), http://www.mtf.ntnu.no/PEOPLE/MICHALM/microfluidic_research.htm
- **MatPIV** – PIV-Softwarepaket für MATLAB, http://www.math.uio.no/~jks/matpiv/
- **GPIV** – Ein freies PIV-Softwareprojekt, http://gpiv.sourceforge.net/gpiv_home.html
- **PIV View (2C)** – Kommerzielles Softwarepaket der Firma *PivTec GmbH* für die 2D-PIV-Analyse, http://www.pivtec.com/. Dort gibt es auch eine freie Demo-Version.
- **EDPIV** – Kommerzielle Software zur digitalen PIV-Evaluation, http://eo.yifan.net/users/l/lcgui/Edpiv_intro.htm. Eine Studentenversion ist ebenso erhältlich.
- **LAVISION** – Hersteller für Hardware zur PIV-Analyse, http://www.piv.de
- **DANTEC Dynamics** – Etablierter Anbieter für PIV-Komplettlösungen, http://www.dantecdynamics.com

Literaturverzeichnis

[Adr96] ADRIAN, RONALD J.: *Bibliography of Particle Image Velocimetry Using Imaging Methods, 1917-1995*. TAM Report 817 UILU-ENG-96-6004, University of Illinois, 1996. 6

[BA83] BURT, PETER J. und EDWARD H. ADELSON: *The Laplacian Pyramid as a Compact Image Code*. IEEE Transactions on Communications, 3l(4):532–540, April 1983. 46, 50, 60

[BD04] BRUEGGE, BERND und ALLEN H. DUTOIT: *Object-Oriented Software Engineering: Using UML, Patterns and Java*. Prentice Hall, Upper Saddle River, NJ, 2. Auflage, Jänner 2004. 65, 66

[Bol99] BOLINDER, JONAS: *On the accuracy of a digital Particle Image Velocimetry system*. Technischer Bericht ISRN LUTMDN/TMVK – 3186 – SE, Division of Fluid Mechanics, Lund Institute of Technology, Lund University, Juni 1999. 32, 87

[Bol00] BOLINDER, JONAS: *Optimization of PIV in flows with strong velocity gradients*. Internet: `http://www.piv.de/papers/bolinderlundpivopt.pdf`, 2000. 54

[Bro99] BROCK, AXEL: *Beitrag zur verbesserten Auswertung der Particle Image Velocimetry (PIV) in turbulenten Strömungen*. Doktorarbeit, Universität von Stuttgart, 1999. 10, 21

[BSMM00] BRONSTEIN, ILJA N., KONSTANTIN A. SEMENDJAJEW, GERHARD MUSIOL und HEINER MÜHLIG: *Taschenbuch der Mathematik*. Verlag Harry Deutsch, 5. Auflage, 2000. 52

[Can86] CANNY, J.: *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, November 1986. 42

[Cha00] CHATTERJEE, ANINDYA: *An introduction to the proper orthogonal decomposition*. Current Science, 78(7):808–817, April 2000. 37, 117

[Che01] CHETVERIKOV, DMITRY: *Particle Image Velocimetry by Feature Tracking*. In: *Computer Analysis of Images and Patterns - 9th International Conference, CAIP Warsaw*, Band LNCS 2124, Seiten 325–332, Poland, September 2001. Springer Verlag. 33, 52

[Chr04] CHRISTOV, IVAN: *Multiscale Image Edge Detection*. Class Research Project, MIT, Mai 2004. 42, 45, 60, 104

[DWD+99] DELNOIJ, E., J. WESTERWEEL, N.G. DEEN, J.A.M. KUIPERS und W.P.M. VAN SWAAIJ: *Ensemble correlation PIV applied to bubble plumes rising in a bubble column*. Chem. Eng. Sci., 54:5159–5171, 1999. 33

[FD00] FINCHAM, A. M. und G. DELERCE: *Advanced optimization of correlation imaging velocimetry*. Experiments in Fluids, [Suppl.], 29:S013–22, 2000. 32, 114

[FPWW04] FISHER, ROBERT, SIMON PERKINS, ASHLEY WALKER und ERIK WOLFART: *Hypermedia Image Processing Reference 2 (HIPR2)*. Internet: `http://homepages.inf.ed.ac.uk/rbf/HIPR2/`, 2004. 39, 40, 41

[GDN95] GRIEBEL, MICHAEL, THOMAS DORNSEIFER und TILMAN NEUNHOEFFER: *Numerische Simulation in der Strömungsmechanik*. Vieweg Verlag, 1995. 36

[GHJV95] GAMMA, ERICH, RICHARD HELM, RALPH JOHNSON und JOHN VLISSIDES: *Design Patterns: elements of reusable object-oriented software*. Addison-Wesley, 1995. 74

[Gle04] GLEIRSCHER, MARIO: *PIVA — Entwicklerdokumentation*, Juli 2004. 65, 69, 74

[GM96] GUI, L. und W. MERZKIRCH: *Phase-separation of PIV digital mask technique*. In: *European Research Community On Flow, Turbulence And Combustion: Bulletin*, Band 30, Seiten 45–48, 1996. 31

[GM98] GUI, L. und W. MERZKIRCH: *Generating arbitrarily sized interrogation windows for correlation-based analysis of Particle Image Velocimetry recordings*. Experiments in Fluids, 24:66–69, 1998. 27

[GM00] GUI, L. und W. MERZKIRCH: *A comparative study of the MQD method and several correlation-based PIV evaluation algorithms*. Experiments in Fluids, 28:36–44, 2000. 31

[GME+02] GRAESSLIN, I., M. MITTELBACH, H. EGGERS, T. SCHAEFFTER, P. BÖRNERT und O. LANGE: *MR Real-Time Coronary Imaging using the Normalized Cross-Correlation*. In: *5. Jahrestagung – Deutsche Sektion der ISMRM e.V.*, Hamburg, Oktober 2002. International Society for Magnetic Resonance in Medicine. 25, 27

[Gra95] GRAPS, AMARA: *An Introduction to Wavelets*. IEEE Computational Science and Engineering, 2(2):50–61, Sommer 1995. 40, 50

[GW02] GUI, L. und S. T. WERELEY: *A correlation-based continuous window shift technique for reducing the peak locking effect in digital PIV image evaluation*. Experiments in Fluids, 32:506–17, 2002. 22, 114

[Göt98] GÖTZ, HERMANN: *Einführung in die digitale Signalverarbeitung*. Teubner Verlag, Stuttgart, 3. Auflage, September 1998. 61

[Har96] HART, D.P.: *Sparse Array Image Correlation*. In: *8th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, 1996. 50, 114

[Har98a] HART, D.P.: *High-Speed PIV Analysis Using Compressed Image Correlation*. Journal of Fluids Engineering, 1998. 50, 114

[Har98b] HART, D.P.: *The Elimination of Correlation Errors in PIV Processing*. In: *9th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, Juli 1998. 32

[Har00] HART, DOUGLAS P.: *Super-Resolution PIV by Recursive Local-Correlation*. Journal of Visualization, 3(2):187–194, 2000. 50, 114

[HDG97] HUANG, H., D. DABIRI und M. GHARIB: *On errors of digital particle image velocimetry*. Meas. Sci. Technol., 8:1427–1440, Dezember 1997. 32

[Hee04] HEESCH, DIMITRI VAN: Doxygen — *Ein Quelltextdokumentationssystem*. Internet: `http://www.doxygen.org`, 2004. 2, 68

[Hes88] HESSELINK, LAMBERTUS: *Digital Image Processing in Flow Visualization*. Ann. Rev. Fluid Mech., 20:421–85, 1988. 6

[HFW92a] HUANG, H. T., H. E. FIEDLER und J. J. WANG: *Limitation and Improvement of PIV. Part I: Limitations of Conventional Techniques due to Deformation of Particle Image Patterns.* Experiments in Fluids, 15(3):168–174, 1992. 10, 19, 20

[HFW92b] HUANG, H. T., H. E. FIEDLER und J. J. WANG: *Limitation and Improvement of PIV. Part II: Particle image distortion, a novel technique*. Experiments in Fluids, 15(3):263–273, 1992. 20, 23, 31

[HSK+01] HU, HUI, TETSUO SAGA, TOSHIO KOBAYASHI, NOBUYUKI TANIGUCHI und MASASHI YASUKI: *Dual-plane Stereoscopic PIV: System Setup and its Application on Lobed Jet Mixing Flow*. Experiments in Fluids, 31(3):277–293, 2001. 7

[JJDAF95] JAMBUNATHAN, K., X. Y. JU, B. N. DOBBINGS und S. ASHFORTH-FROST: *An improved cross correlation technique for Particle Image Velocimetry*. Meas. Sci. Technol., 6:507–14, 1995. 20

[Jäh02] JÄHNE, BERND: *Digitale Bildverarbeitung*. Springer-Verlag, 5. Auflage, 2002. 39, 42

[KA90] KEANE, R. D. und R. J. ADRIAN: *Optimization of particle image velocimeters. Part I: Double-pulsed systems*. Meas. Sci. Technol., 1:1202–1215, 1990. 19, 54

[KA92] KEANE, R. D. und R. J. ADRIAN: *Theory of cross-correlation analysis of PIV images*. Appl. Sci. Res., 49:191–215, 1992. 10, 19, 20, 54

[KAZ95] KEANE, R.D., R.J. ADRIAN und Y. ZHANG: *Super-resolution Particle Image Velocimetry*. Meas. Sci. Technol., 6:754–768, 1995. 32

[KLS04] KIM, B. J., Y. Z. LIU und H. J. SUNG: *Micro PIV measurement of two-fluid flow with different refractive indices*. Meas. Sci. Technol., 15:1097–1103, Juni 2004. 33

[Knu86] KNUTH, DONALD E.: *Computers and Typesetting: A - The TEX book, B - TEX: The Program, C - The METAFONT book, D - METAFONT: The Program, E - Modern Typefaces*. Addison-Weseley, Reading, MA, 1984-86. i

[Kop02] KOPKA, HELMUT: *LATEX - Band 1: Einführung*. Pearson Studium. Addison-Wesley, 3. Auflage, 2002. i

[KSK+02] KIMURA, ICHIRO, YASUNORI SUSAKI, R. KIYOHARA, AKIKAZU KAGA und YASUAKI KUROE: *Gradient-based PIV Using Neural Networks*. Journal of Visualization, 5(4), 2002. 37

[Kun02] KUNZ, DIETMAR: *Skript zur Vorlesung „Digitale Bildverarbeitung", WS 2001/02, Kapitel 6. Rauschen*. Internet: http://www.fo.fh-koeln.de/dozenten/dkunz/skripte/bv/bv06.pdf, 2002. 40, 57

[Lew95] LEWIS, J. P.: *Fast normalized cross-correlation*. In: *Vision Interface*, Seiten 120–123, Quebec City, Mai 1995. 25, 28, 29, 50

[LK95] LOURENÇO, L. und A. KROTHAPALLI: *On the Accuracy of Velocity and Vorticity Measurements with PIV*. Experiments in Fluids, 18:421–428, 1995. 37

[Lou00] LOUIS, DIRK: *C/C++*. Das Kompendium. Markt&Technik, München, 2000. 68

[Mal99] MALLAT, S.: *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, USA, 2. Auflage, 1999. 40

[MM02] MCKENNA, S. P. und W. R. MCGILLIS: *Performance of digital image velocimetry processing techniques*. Experiments in Fluids, 32:106–115, 2002. 61

[MWS00] MEINHART, C.D., S.T. WERELEY und J.G. SANTIAGO: *A PIV algorithm for estimating time-averaged velocity*. Journal of Fluids Engineering, 122:285–289, 2000. 33

[MZ92] MALLAT, S. und S. ZHONG: *Characterization of Signals from Multiscale Edges*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(7):710–732, Juli 1992. 42, 43, 58

[NA02] NIXON, MARK S. und ALBERTO S. AGUADO: *Computer Vision and Image Processing: Feature Extraction*. Newnes, Jänner 2002. 42

[NLR97] NOGUEIRA, J., A. LECUONA und P. A. RODRÍGUEZ: *Data validation, false vectors correction and derived magnitudes calculation on PIV data*. Meas. Sci. Technol., 8:1493–1501, 1997. 26

[NLRR02] NOGUEIRA, J., A. LECUONA, U. RUIZ-RIVAS und P. A. RODRÍGUEZ: *Analysis and alternatives in two-dimensional multigrid particle image velocimetry methods: application of a dedicated weighting function and symmetric direct correlation*. Meas. Sci. Technol., 13:963–974, 2002. 114

[OB02] OERTEL JR., HERBERT und MARTIN BÖHLE: *Strömungsmechanik – Grundlagen, Grundgleichungen, Lösungsmethoden, Softwarebeispiele*. Studium Technik. Vieweg, 2. Auflage, März 2002. 36

[ONSK00] OKAMOTO, K., S. NISHIO, T. SAGA und T. KOBAYASHI: *Standard images for particle-image velocimetry*. Meas. Sci. Technol., 11:685–691, 2000. 14, 33, 68, 87, 88

[Pus00] PUST, O.: *PIV: Direct Cross-Correlation compared with FFT-based Cross-Correlation*. In: *Proceedings of the 10th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, 2000. 27, 114

[QPK98] QUÉNOT, G. M., J. PAKLEZA und T. A. KOWALEWSKI: *Particle Image Velocimetry with optical flow*. Experiments in Fluids, 25:177–189, 1998. 32

[Qué92] QUÉNOT, G. M.: *The „orthogonal algorithm" for optical flow detection using dynamic programming*. In: *Proceedings of IEEE ICASSP, III*, Seiten 249–252, 1992. 32

[Rec02] RECHENBERG, PETER (Herausgeber): *Informatik-Handbuch*. Hanser-Verlag, München, 3. Auflage, 2002. 67

[RSY01] RUAN, X., X. SONG und F. YAMAMOTO: *Direct measurement of the vorticity field in digital particle images*. Experiments in Fluids, 30:696–704, 2001. 21

[RWK98] RAFFEL, M., C. WILLERT und J. KOMPENHANS: *Particle Image Velocimetry – a Practical Guide*. Springer, Berlin, 1998. 5, 8, 18, 23, 25, 26, 27, 36, 37, 50, 54

[Sca02] SCARANO, F.: *Iterative image deformation methods in PIV*. Meas. Sci. Technol., [Review] 13:R1–R19, Jänner 2002. 20, 22, 29, 32, 52

[Sil04] SILICON GRAPHICS: *Standard Template Library (STL) — Programmer's Guide*. Internet: `http://www.sgi.com/tech/stl/`, 1993-2004. 117

[SNOO00] SUGII, YASUHIKO, SHIGERU NISHIO, TAKETOSHI OKUNO und KOJI OKAMOTO: *A highly accurate iterative PIV technique using a gradient method*. Meas. Sci. Technol., 11:1666–1673, Dezember 2000. 30, 31

[SOK03] STANISLAS, M., K. OKAMOTO und C. KÄHLER: *Main results of the First International PIV Challenge*. Meas. Sci. Technol., [Review] 14:R63–R89, 2003. 29, 30

[SR99] SCARANO, F. und M. L. RIETHMULLER: *Iterative multigrid approach in PIV image processing with discrete window offset*. Experiments in Fluids, 26(6):513–523, 1999. 23, 24, 49, 50, 51, 112

[SR00] SCARANO, F. und M. L. RIETHMULLER: *Advances in iterative multigrid PIV image processing*. Experiments in Fluids, [Suppl.], 29:S51–S60, 2000. 22, 23, 24, 49

[TD95] TOKUMARU, P. T. und P. E. DIMOTAKIS: *Image Correlation Velocimetry*. Experiments in Fluids, 19:1–15, 1995. 32

[Use99] USERA, GABRIEL: *Adaptive Algorithms for PIV Image Analyzing*. In: *Grupo de Trabajo Sobre Hidromecánica, 5ª Reunión*, Montevideo - Uruguay, Mai 1999. 114

[WDG97] WESTERWEEL, J., D. DABIRI und M. GHARIB: *The effect of a discrete window offset on the accuracy of cross-correlation analysis of digital PIV recordings*. Experiments in Fluids, 23:20–28, 1997. 20

[Wes93] WESTERWEEL, JERRY: *Particle Image Velocimetry – Theory and Application*. Doktorarbeit, Delft University, 1993. 6, 7, 8, 10, 12, 18, 19, 22, 23, 25, 35, 36, 37, 49, 53, 54, 111, 112

[Wes94] WESTERWEEL, J.: *Efficient detection of spurious vectors in Particle Image Velocimetry data sets*. Experiments in Fluids, 16:236–247, 1994. 25, 26

[WG91] WILLERT, C. E. und M. GHARIB: *Digital Particle Image Velocimetry*. Experiments in Fluids, 10:181–193, 1991. 11

[WG03] WERELEY, S.T. und L. GUI: *A correlation-based central difference image correction (CDIC) method and application in a four-roll mill flow PIV measurement*. Experiments in Fluids, 34:42–51, 2003. 20, 31, 32

[Wik] WIKIPEDIA — *Freie, mehrsprachige, multidisziplinäre und sehr umfangreiche Online-Enzyklopädie*. Internet: `http://de.wikipedia.org/wiki/Hauptseite`. 116

[WK04] WILLIAMS, THOMAS und COLIN KELLEY: gnuplot — *An Interactive Plotting Program*. Internet: `http://www.gnuplot.info`, April 2004. i, 67

[WM01] WERELEY, S. T. und C. D. MEINHART: *Second-order accurate Particle Image Velocimetry*. Experiments in Fluids, 31:258–268, 2001. 31

[WPL02] WESTERWEEL, J., C. POELMA und R. LINDKEN: *Two-point ensemble correlation method for μ-PIV applications*. In: *11th Int. Symp. on Applications of Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, Juli 2002. 33

Index

FOURIER-Transformation, 13
NAVIER-STOKES-Gleichungen, 13
YOUNG's Streifen, 8

Abfragefenster, 11
Abfragefensterdichte, 51
Abfrageraster, 10, 11
Adaption, 51
Ausreißer, 25

Bildauflösung
  Vergröberung, 60

CCD-Kamera, 8

Deformationstensor, 36
detection
  peak, 17
digitale Filter, 39

exposure
  double, 9
  multiple, 9
  single, 9

Filter
  Glättungs-, 40
  Unschärfe-, 40
frame
  double, 9
  single, 9
Frequenzband, 40

Geschwindigkeitsgradienten, 54
Geschwindigkeitsverteilung, 54
Gewichtungskern, 47

Kantendetektion, 41
  multiskalar, 42
  Wavelets, 42

Kompression
  verlustbehaftet, 50
Konvolutionskern, 40
Korrelation, 12
Korrelationskoeffizient, 10
Kovarianz, 12
Kreuzkorrelation, 6, 11
Kreuzkorrelationsebene, 13
Kreuzkorrelationskoeffizient, 25

Leuchtintensitätsfeld, 9

Merkmalsverfolgung, 33
Messfehler, 25

OPIV, 8
outlier, 25

Partikel
  Streudichte, 8
Partikelbilder, 6
Partikelbildverschiebung, 18
peak-locking, 22
PIV
  effiziente, 28
  Geschichte, 5
  Gradientenmethode, 30
  hybride, 28
  theoretische Einführung, 6
  Verfahrensmodi, 8
PIVA
  Anforderungen, 66
    nichtfunktionell, 67
  Anwenderschnittstelle, 67
  Anwendungsfall, 66
  Klassen, 74
  Kommandozeilenoptionen, 69
  Systemarchitektur, 73, 74

Quadtrees, 109

Rauschen, 39

Signalfilterung, 39
Softwareentwicklungsprozess, 65
Standardabweichung, 40
Strömung
    Topologie, 35
    Typen, 36
    Zirkulation, 35
Suchfenster, 11, 19
    Deformation, 20
    Probleme, 54
    Verschiebung
        diskret, 20
        kontinuierlich, 22

Vektorfeld
    Verfeinerung, 49
Vektorfeldauflösung, 23
vorticity vector, 35

zentrale Differenzen, 31
Zufallssignal, 25

Urheberrechtsvermerk

© September 2004, das Urheberrecht dieser Ausarbeitung und der in diesem Rahmen erstellten Software liegt beim Autor Mario Gleirscher.

Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet. Im Text vorkommende Firmen- oder Produktnamen u. ä. sind geschützte Bezeichnungen ihrer Urheber.

Der Autor ist damit einverstanden, dass auf diese Arbeit in wissenschaftlichen Veröffentlichungen Bezug genommen wird. Der Technischen Universität München (TUM), vertreten durch den Lehrstuhl für „Computer Graphics & Visualization“, wird ein (nicht ausschließliches) Nutzungsrecht an dieser Arbeit eingeräumt. Dieses Nutzungsrecht schließt jedoch die Weitergabe und (gewinnbringende) Nutzung der in Zusammenhang mit dieser Arbeit erstellten Software und Dokumentation (einschließlich dieser Ausarbeitung) für Verwendungszwecke außerhalb des nichtkommerziellen Lehr- und Forschungsbetriebes aus. Über ein gesondertes Nutzungsrecht ist mit dem Autor und nur diesem selbst zu verhandeln.