

Development of an Advanced Method for a Micro Particle Tracking Velocimetry Application

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Technische Mathematik

eingereicht von

Christian Ratzenböck

Matrikelnummer 0626873

an der

Fakultät für Informatik der Technischen Universität Wien

Betreuung: a.o.Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig

Mitwirkung: Dipl.-Ing. Dr.techn. Christian Maszl (Austrian Institute of Technology)

Dipl.-Ing. Dr.techn. Johann Emhofer (Austrian Institute of Technology)

Wien, 11. Oktober 2013

(Unterschrift Verfasser)

(Unterschrift Betreuung)

Development of an Advanced Method for a Micro Particle Tracking Velocimetry Application

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Technical Mathematics

by

Christian Ratzenböck

Registration Number 0626873

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: a.o.Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Assistance: Dipl.-Ing. Dr.techn. Christian Maszl (Austrian Institute of Technology)
Dipl.-Ing. Dr.techn. Johann Emhofer (Austrian Institute of Technology)

Vienna, October 11, 2013

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Christian Ratzenböck
Simmeringer Hauptstraße 68-74 2/8, 1110 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasser)

Abstract

The research and development of novel fluids for high-efficient absorption heat pump applications require a detailed analysis of microscopic liquid flows. For this purpose, a Single Aperture Defocusing Micro Particle Tracking Velocimetry system has been implemented within the scope of this work. Due to the lack of commercial software for the considered application, an image processing software is developed which computes spatially resolved velocity data by means of flow-traced particles in recorded image sequences. As the particle position along the optical axis determines its shape in the image, the application of a particle matching algorithm is required for a full, three-dimensional characterization of the flow.

The method introduced for particle detection is based on a modified Circular Hough Transform operating on the binary edge map of a particle image denoised with the aid of a nonlinear diffusion filter. This approach permits the detection of partially overlapping particles and is robust to variations in particle shape, image contrast and noise. Furthermore, a matching technique using normalized radial intensity profiles is developed and a tracking algorithm is presented which is customized for planar flows and able to compensate for potentially imprecise matching results.

The evaluation shows, that the matching technique proposed achieves a considerably increased accuracy along with a reduction in computation time in comparison to conventional cross-correlation based matching methods. Moreover, the particle detection method as well as the particle tracking method individually perform almost error-free. In total, promising results on manually annotated experimental data as well as automatically generated synthetic data demonstrate the applicability of the software.

Kurzfassung

Das Verhalten mikroskopisch dünner Fallfilme spielt eine wichtige Rolle in der Erforschung und Entwicklung neuartiger Flüssigkeiten für den Einsatz in modernen, hocheffizienten Absorptionswärmepumpen. Für die detaillierte Erfassung der räumlich aufgelösten Strömungsgeschwindigkeit wurde im Rahmen dieser Arbeit ein Single Aperture Defocusing Micro Particle Tracking Velocimetry System installiert. Dabei werden lokale Geschwindigkeitsdaten mit Hilfe von Mikropartikeln, die in aufgenommenen Bildsequenzen verfolgt werden, berechnet. Da sich die Tiefenposition der Partikel im Film in ihrer Form im Bild widerspiegelt, ist die Entwicklung eines Matching-Algorithmus zur vollständigen Beschreibung des dreidimensionalen Systems notwendig.

Die in dieser Arbeit beschriebene Bildverarbeitungssoftware wertet die Aufnahmen des Systems aus. Zunächst werden die Bilder durch Anwendung eines nichtlinearen Diffusionsfilters entrauscht. Anschließend werden die Partikel durch eine modifizierte Circular Hough Transform, die sich auf ein binäres Kantenbild stützt, detektiert. Dieser Ansatz ermöglicht auch die Erfassung von (teilweise) überlappenden Teilchen im Bild und ist robust gegenüber Abweichungen in Partikelform, Bildkontrast und Rauschen. Der vorgestellte Matching-Algorithmus basiert auf der Beschreibung der Partikel durch normalisierte, radiale Helligkeitsprofile und ist, wie die Auswertung beweist, präziser und schneller als konventionelle Kreuzkorrelationsmethoden. Die Verknüpfungen der Teilchen zwischen zwei Frames werden durch einen Tracking-Algorithmus berechnet, der speziell auf den vorliegenden Fall eines ebenen Flusses zugeschnitten ist und eventuelle Ungenauigkeiten im Matching ausgleichen kann.

Eine ausführliche Evaluierung der vorgeschlagenen Methode anhand eines manuell annotierten, experimentellen Datensatzes sowie eines automatisch generierten, synthetischen Datensatzes beweist die erfolgreiche Einsetzbarkeit der entwickelten Software.

Contents

1	Introduction	1
1.1	Motivation	6
1.1.1	Scope of Discussion	6
1.1.2	Objective	10
1.1.3	Contribution	10
1.2	Results	11
1.3	Outline	12
2	State of the Art	13
2.1	Image Pre-Processing Methods	13
2.2	Particle Detection and Particle Matching Methods	17
2.3	Particle Tracking Methods	19
2.4	Summary	21
3	Methodology	23
3.1	Step 1: Image Pre-Processing	23
3.1.1	Linear diffusion filtering	25
3.1.2	Nonlinear diffusion filtering	26
3.1.2.1	The Perona-Malik model	26
3.1.2.2	The continuous CLMC equation	26
3.1.2.3	The discrete CLMC equation	28
3.1.2.4	Anisotropic Diffusion Filtering	34
3.1.3	Proposed modifications and impact on Micro PTV data	34
3.2	Step 2: Particle Detection	38
3.2.1	Edge Detection - The Canny Algorithm	39
3.2.2	Hough Transforms	40
3.2.2.1	Standard Hough Transform	41
3.2.2.2	Circular Hough Transform	42
3.2.2.3	Generalized Hough Transform	44
3.2.3	Proposed modifications and impact on Micro PTV data	45
3.3	Step 3: Particle Matching	52
3.3.1	Normalized Cross-Correlation	55
3.3.2	Proposed Method: Radial intensity profiles	57

3.4	Step 4: Particle Tracking	58
3.4.1	Linear Programming	59
3.4.2	Integer Programming	61
3.4.3	Proposed Implementation	63
3.5	Summary	67
4	Evaluation and Results	69
4.1	Experiments Overview	69
4.2	Experimental Dataset	73
4.3	Synthetic Dataset	82
4.4	Summary	85
5	Conclusion and Outlook	87
	Bibliography	89
	List of Acronyms	97

Introduction

“Fluid mechanics is that discipline within the broad field of applied mechanics that is concerned with the behavior of liquids and gases at rest or in motion” [47]. As listed by [47], the investigation of fluid mechanics has concerned scientists throughout history: from early writings of Archimedes (287-212 B.C.), who first expressed the principles of hydrostatics and flotation, to contributions associated with famous names such as Newton, Euler, Navier and Stokes up to the pioneering work of Ludwig Prandtl, a German professor in the early twentieth century, who is *“generally considered to be the father of present-day fluid mechanics”* [47]. The task of determining the velocity fields of liquid flows, however, is still challenging to this day [61, 63].

Particle Image Velocimetry (PIV) and Particle Tracking Velocimetry (PTV) are closely related optical methods for visualizing velocity fields in liquid and gas flows [61]. The principal strategy of “classical” PIV [3, 61] is as follows:

Firstly, tiny spherical tracer particles (“seeding particles”), which are assumed to follow the stream lines without influencing the local flow velocity, are added to the flow. In the second step, a 2-dimensional region of interest (“light sheet”) is illuminated by a coherent light source. Finally, the scattered light by the particles is recorded on a camera, positioned such that the light sheet is located near to its focal plane. The velocity field of the flow in the light sheet is subsequently deduced with the aid of the particle displacements and the knowledge of time between the two frames. By repeatedly relocating the light sheet along the optical axis, the velocity field of a 3-dimensional interrogation volume can be determined. However, all particular velocity components are 2-dimensional as solely motions within the light sheets are captured. Figure 1.1 demonstrates this procedure.

PIV and PTV differ solely by the strategy how the seeding particle displacements are calculated. In the case of PTV, the corresponding particles between two consecutive frames are associated and tracked, whereas PIV uses a statistical evaluation based on cross-correlation techniques. If a “medium” particle density is provided in the image, i.e. single particles can be distinguished while *“it is no longer possible to identify image pairs by visual inspection of the recording”* [61], PIV is applied. For “low” particle densities, PTV is the preferred method, while in the case of “high” particle density, where individual particles cannot be identified any

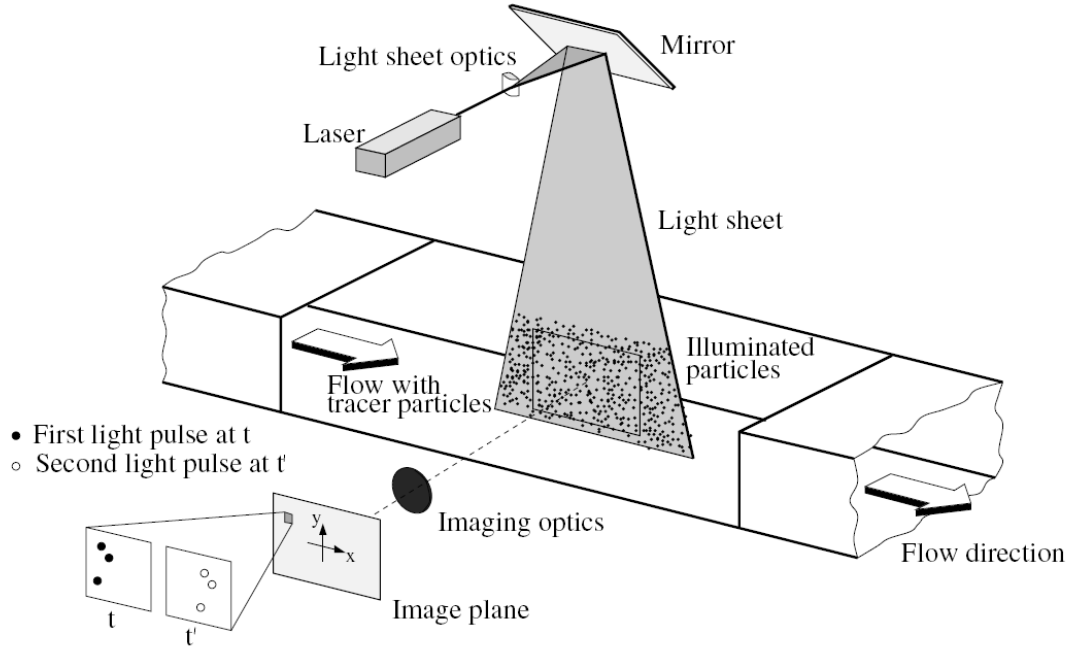


Figure 1.1: Experimental arrangement for a PIV measurement. Figure taken from [61].

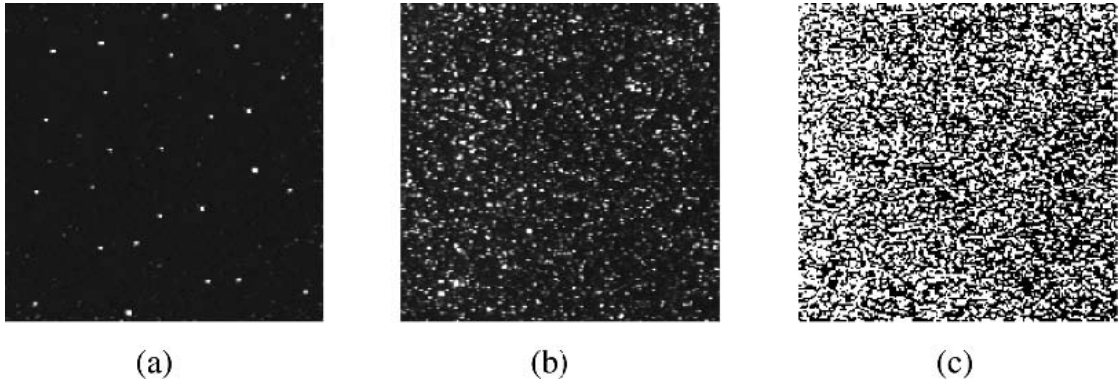


Figure 1.2: Three modes of particle image density: (a) low (PTV), (b) medium (PIV), (c) high (LSV) image density. Figure taken from [61].

more, so called Laser Speckle Velocimetry (LSV) is applied [4, 61]. In Figure 1.2, examples of the three particle image density modes (low, medium, high) are shown. Due to this close relation of PIV and PTV, for the rest of this chapter, if PTV is not explicitly mentioned, the term “PIV” shall represent both PIV and/or PTV.

The choice of size and material of the seeding particles is influenced by different physical conditions, e.g. fluid density and viscosity. For a detailed description of the method of selection, we refer to [4] and [61].

The main advantage of PIV methods is their non-intrusivity [63], i.e. the flow is not disturbed by the measurement. Additionally, “*aside from Doppler Global Velocimetry [...], which is a new technique particularly appropriate for medium to high-speed air flows, and Molecular Tagging Velocimetry [...] all other techniques for velocity measurements only allow the measurement of the velocity of the flow at a single point*” [61] whereas PIV results provide global flow velocity information.

When dealing with flows where the spatial scale of the fluid motion is approximately in the range of

$$10^{-4} \text{ to } 10^{-7} \text{ m,}$$

the so-called μ PIV (or micro-PIV) measurement technique is applied [84]. The basic concept of μ PIV coincides with the PIV approach, whereas the small length scales constitute “*considerably different optical and mechanical constraints*” [84]. Therefore, μ PIV is regarded as a separate technique, which - since its first introduction by Santiago et al. in 1998 [61, 64] - has become a standard measurement technique for the investigation of micro-scale flows [59]. According to [61], there are three major differences which distinguish μ PIV from macroscopic PIV:

- the seeding particles are small compared to the wavelength of the illuminating light,
- the seeding particles are small enough that the effects of Brownian motion must be considered,
- the illumination source is typically not a light sheet but rather an illuminated volume of the flow due to limited optical access,

whereby volume illumination depicts the fundamental difference [42]. This implicates the requirement of a different approach since the 3-dimensional position information of the particles is not constituted by the location of a light sheet but rather has to be identified in another manner. Furthermore, the standard PIV method assumes a planar flow and is not capable of computing the out-of-plane velocity component which can lead to substantial errors [61]. Hence, advanced, existing PIV and newly developed μ PIV techniques have recently been presented and assessed with respect to a successful application concerning microscale flows [40, 61, 84, 85]. In the following, six common techniques are summarized according to their description in [40, 84, 85]:

- **Standard μ PIV** extracts and processes particles from a measurement plane, similar to the previously described standard macroscopic PIV approach. The difference is that in the microscopic framework, the measurement plane - the focal plane of the camera - is distinguished by characteristics of the optical system rather than by illumination. As pointed out by Wereley and Meinhart [84], this is made possible on account of the “*sharply defined objective planes*” of microscopic objective lenses, enabling particles to transition “*quickly from being in focus to being out of focus*”. As with macroscopic PIV, a (3-dimensional) interrogation area is scanned by altering the position of the optics and a 2-component velocity field is constructed (3D-2C method).

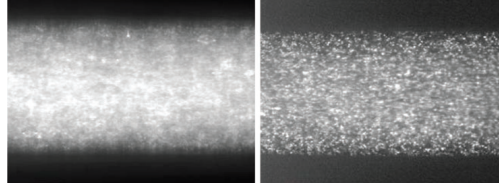


Figure 1.3: Particle images acquired using (a) standard imaging and (b) confocal imaging. Figure taken from [84].

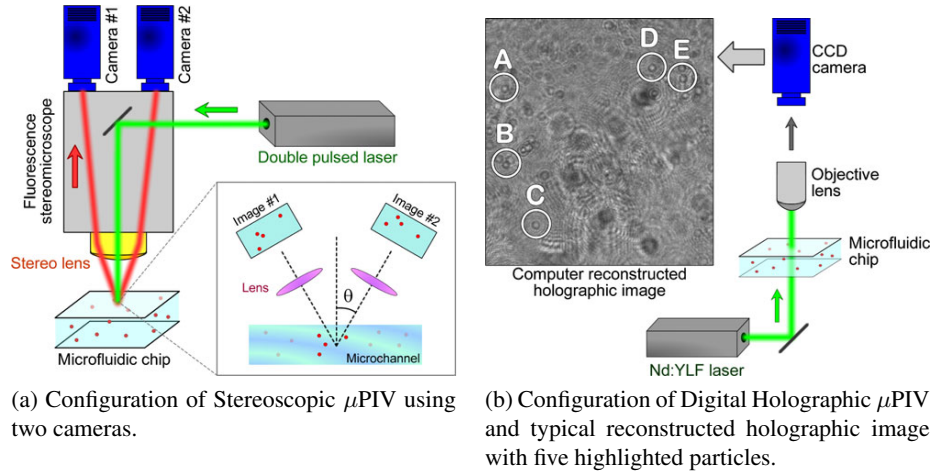


Figure 1.4: Two different μ PIV variants. Both figures taken from [85].

- **Confocal μ PIV** applies the technique of focusing the illumination and the recording optics on a single coincidental spot. As a result, the image plane has to be scanned point by point in order to acquire one image. In return, interfering background light from out-of-focus images is reduced and the spatial resolution along the optical axis is increased. Otherwise, this method is identical to the standard approach. Figure 1.3 demonstrates the superior image contrast of this technique.
- **Stereoscopic μ PIV** uses multiple cameras to display the same interrogation area from different viewing angles, thus allowing the reconstruction of the 3-component velocity field, as indicated in Figure 1.4(a). The depth of the measurement volume, however, is “restricted to a relatively small depth-of-focus region which is determined by the objective lens” [40]. Therefore, Stereoscopic μ PIV is generally referred to as a 2D-3C method [40, 61, 84, 85].
- **Digital Holographic μ PIV:** A hologram image is generated by the interference pattern that is created by the scattered light by the particles and the unaffected laser beam. Both amplitude and phase information of the objects are saved instead of traditional intensity data. The digital hologram can be numerically reconstructed and 3D particle information

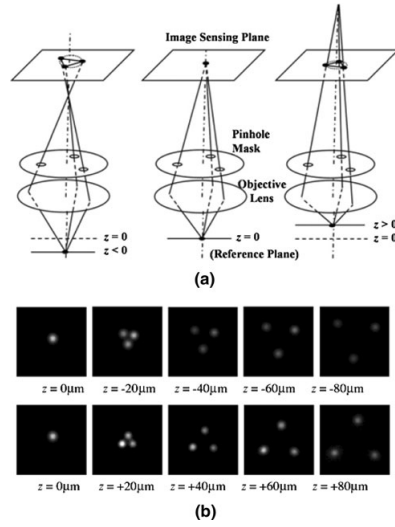


Figure 1.5: (a) Defocusing concept using a three-hole-aperture mask, (b) Experimentally recorded particle images. Figure taken from [85].

is obtained. The application of a 3D tracking algorithm yields the 3-component velocity field (3D-3C). Figure 1.4(b) shows a holographic image acquired by this method as well as its schematic configuration.

- **Defocusing μ PIV:** In literature, two distinct defocusing techniques are mentioned [40,84, 85]. The first method uses a **multi-hole aperture** mask on a single camera recording the frames. Figure 1.5 indicates the scenario of a three-hole aperture where the size and orientation of the triangular particle image pattern provide information on a particle's distance to the focal plane.

The second defocusing method is applied to **single-hole aperture** imaging systems. The 3-dimensional particle position is encoded in the characteristic structure of defocused particle images. As will be motivated at the beginning of Section 1.1.1, this approach, also applied by Paschke et al. [56] and Park and Kihm [54], is followed in this thesis.

Table 1.1 gives an overview of these μ PIV methods. The first column contains the method's name, Column 2 states the dimension of the image field (i.e. if an interrogation plane or volume is treated) and Column 3 indicates the number of components of the calculated velocity vectors (i.e. if the particle velocities are restricted to a plane). In the last column, the authors and the year of publication of the first successful application of the method in the field of microfluidics are listed.

This chapter will first give the motivation for the thesis, followed by the scope of discussion where further physical background is elucidated. In the subsequent two sections, the objective and the main contribution of this thesis are illustrated. Section 1.2 provides a summary of the results and an outline of the thesis concludes this chapter.

Table 1.1: Overview of advanced μ PIV techniques.

Method	ID ^a	VC ^b	Original Source
“Standard” μ PIV	3D	2C	Santiago et al. [64] in 1998
Confocal μ PIV	3D	2C	Park et al. in 2004
Stereoscopic μ PIV	2D	3C	Lindken et al. in 2006
Holographic μ PIV	3D	3C	Satake et al. in 2005
Multi-hole Aperture Defocusing μ PIV	3D	3C	Yoon and Kim [88] in 2006
Single Aperture Defocusing μ PIV	3D	3C	Park and Kihm [54] in 2006

^a Image Dimension^b Velocity Components

1.1 Motivation

This thesis is part of a project realized in cooperation of the Austrian Institute of Technology (AIT) with the German chemical company Evonik. One main aim of the project is the development of a Single Aperture Defocusing μ PTV system in order to analyze the efficiency of falling film absorbers.

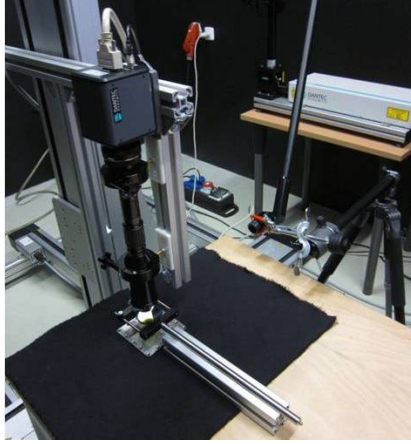
Due to their high heat and mass transfer rates, falling films are used in energy and process technology for example for heating, cooling, evaporation and condensation purposes [6]. Common devices using the falling film technology are falling film evaporators and heat exchangers [60]. Moreover, the range of application includes the chemical industry, the food industry, the paper industry and refrigeration technology [6].

At the AIT, falling films are primarily applied in the field of absorption heat pumps. Considering the development of more efficient devices, there is an emphasis on the research, development and optimization of novel fluids in use (the so-called *working pairs*) in order to replace the currently conventional working pairs LiBr–H₂O (Lithium-Bromide and Water) and H₂O–NH₃ (Water and Ammoniac) [68]. The reasons for this are their detrimental characteristics, such as corrosivity (LiBr–H₂O and H₂O–NH₃), toxicity (H₂O–NH₃) and risk of crystallization (LiBr–H₂O).

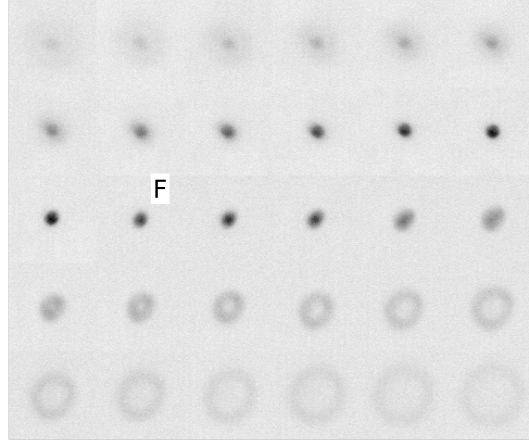
The company Evonik is in a worldwide leading position regarding the development and investigation of new working pairs for absorption heat pumps [78] that include hyperbranched polymeres, ionic liquids and water. As such fluids do not possess the above-mentioned disadvantages, they constitute potential replacements for the working pairs in use, similar heat and mass transfer rates provided. Since heat and mass transfer are, inter alia, influenced by the local flow velocity [66], it is required to examine the velocity field in order to acquire information about the suitability of the liquid for an application in the field of absorption heat pumps.

1.1.1 Scope of Discussion

A film thickness in the range of approximately 400 to 600 μ m is expected for the given application and therefore, the installation of a μ PIV system is required. In order not to omit potential velocity components, the implementation of a 3C technique is necessary, cf. Table 1.1.



(a) Setup for reference image extraction.



(b) 30 experimentally obtained reference images, ordered by increasing distance to backplate, row-wise from left to right. The focused particle is denoted by "F". Gray values are inverted and the contrast is artificially enhanced.

Figure 1.6: Reference image measurement.

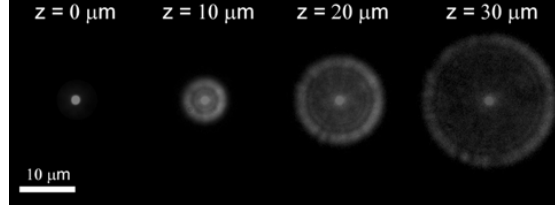
At the time of the draft of this work, the experiments have been conducted on liquid flows under atmospheric conditions. In further consequence, however, it is desired to investigate flows inside specific devices under vacuum conditions. In this context, the visual accessibility is constrained, hence disqualifying the Stereoscopic μ PIV technique where the installation of multiple cameras is required. As indicated in Figure 1.4(b), the non-transparency of the backplate additionally complicates the application of holographic methods. Apparently, the implementation of a Defocusing μ PIV technique represents the superior choice for the existing situation. Since multiple apertures involve the segmentation problem of overlapping particle patterns [85], the decision has been taken in favor of the Single Aperture Defocusing μ PIV method. An experimental setup was installed at the AIT, a detailed description of which is provided below (see Figure 1.8).

The chosen Single Aperture Defocusing μ PIV technique requires the classification of differently shaped particles. In order to facilitate the particle recognition process, a comparatively low density of seeding particles in the investigated liquid is established (resulting in approximately 60-100 particles per image). The velocity calculations therefore are conducted by means of a tracking system, i.e. μ PTV.

The essence of the proposed method is to compare the detected particles to reference images pre-recorded at variable distances. The standard reference image measurement process is executed as follows (setup shown in Figure 1.6(a)): Initially, (at least) one particle is placed in a drop of the investigated liquid and the camera is positioned such that the designated particle is located in its focal plane. After recording the particle, the optical system is repeatedly shifted by 20 μ m along the optical axis (in both directions) in order to obtain multiple reference images.



(a) Particle images studied in [86]. Particle diameter: $1\ \mu\text{m}$, shift from focus: $z=-30, -20, -10, 0, 10, 20, 30\ \mu\text{m}$ respectively, $20\times$ objective lens. Image taken from [86].



(b) Particle images studied in [59]. Particle diameter: $3\ \mu\text{m}$, shift from focus denoted as z , $60\times$ water immersion lens. Image taken from [85].

Figure 1.7: Reference particles from different applications in literature.

Particles near to the focal plane are recorded as small, bright spots on the image, while particles behind the focal plane seem to grow a diffuse corona and particles closer to the optical system adopt a larger, annular shape. Figure 1.6(b) displays the results of a reference image compilation with 13 camera shifts towards the particle and 16 shifts away from it, respectively. Starting from the top left corner, the recorded particles are shown row by row in ascending order of distance to the optical system. The second particle from the left in the third row is situated in the camera focus.

As stated by [59], the “*large number of microscope and μPIV system configurations (e.g., numerical aperture, magnification, Fresnel number, particle size, objective immersion medium, etc) makes a general analysis of the [...] pattern produced by a spherical seed particle on the image plane cumbersome*”. Figure 1.7 shows reference images originating in different optical setups.

Rings in the defocused imaging plane can be caused by both diffraction and spherical lens aberration, as cited by Wu et al. [86]. For that matter, they [86] investigated the contribution of those optical phenomena to the image both experimentally and numerically in a similar setup. It is demonstrated that spherical lens aberration is the decisive factor for the characteristic particle shapes.

Ultimately, the actual location of a particle is determined by the reference image with the highest congruence. In the papers by Park and Kihm [54] and Peterson et al. [59], the focal plane is placed behind the flow containing the particles. Out-of-focus particles therefore are ring-shaped exclusively and allow the characterization by the radius of the outermost ring. It is evident from Figure 1.6 and 1.7 that the Signal-to-Noise Ratio (SNR) of reference images, a measure of signal strength relative to background noise, decreases with increasing distance to focus. Therefore, the positioning of the optical system such that the focal plane is located approximately in the middle of the flow, as implemented by Paschke et al. [56], yields an exten-

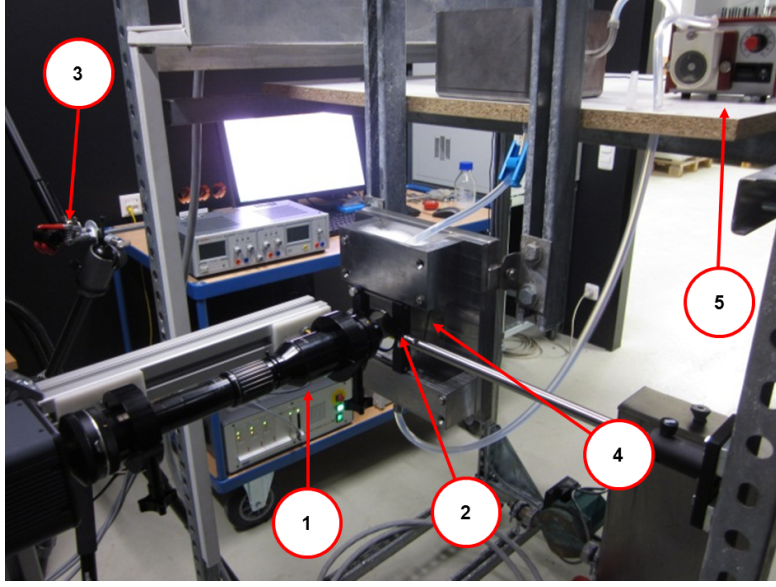


Figure 1.8: Experimental setup: (1) long distance microscope installed on CCD camera, (2) optical band pass filter, (3) laser, (4) backplate of falling film, (5) tube pump.

sion of the examined flow volume. Due to the distinguished shape of behind-the-focus particles, a different matching technique is required. In [56], a cross-correlation based approach is used which also is adopted and evaluated in this work. Alternatively, an algorithm based on radial intensity profiles is proposed in Section 3.3.2.

The considered μ PTV system, as shown in Figure 1.8, consists of the following components:

- **Laser:** DualPower 200-15. Laser medium: Nd:YAG, Beam diameter: 6.5 mm, Repetition rate: max. 15 Hz, Energy: max. 200 mJ per pulse, Wavelength: 532 nm.
- **Seeding particles:** Material: melamine formaldehyde (MF), Diameter: $10.2\ \mu\text{m}$ (standard deviation: $0.17\ \mu\text{m}$), Fluorescent dye: Rhodamine B.
- **Optical system:** Consists of
 - **Digital CCD Camera:** Hamamatsu Photonics C9300-501. Resolution: 2048x2048 pixels.
 - **Long distance microscope:** InfiniMax MX-6. Working distance: 66 mm, Magnification: 2.90, Depth of field: $24\ \mu\text{m}$, Numerical aperture: 0.15, Resolution: $22\ \mu\text{m}$, Field of view: 2.2 mm.
 - **Filter:** Optical band pass filter. Transmitted wavelength: 580 nm. Subsequent to the excitation of the fluorescent seeding particles by the laser (at 532 nm), the wavelength of the emitted light changes [61]. With the aid of the band pass filter, solely the relevant wavelength is transmitted and the influence of perturbing factors such as background light and reflections from the plate is minimized such that (almost) exclusively the particles are visible in the image.

- **Metal backplate**
- **Pump:** The pump ensures that the backplate is constantly wetted as uniformly as possible.
- **Liquids:** The experiments have been conducted with the following liquids:
 - Soap water
 - A LiBr/H₂O solution
 - A recently developed ionic liquid, called “LAYER-1”, diluted with water

1.1.2 Objective

The aim of this thesis is the development and the implementation of a software tool which is capable of extracting the spatially resolved particle velocities with the use of the μ PTV system existing at the AIT. In order to ensure comparability of the results to existing fluid motion theories (e.g. Nusselt’s Film Theory [49]) and therefore guarantee a practically relevant validation of the software, the tracking algorithm initially has been designed to solely detect planar movement. Due to the low Reynolds numbers of the considered falling films, this assumption agrees with the physical properties of laminar flows.

The goal is to extract film velocity statistics that are significant and allow an accurate evaluation of the investigated fluid. For this purpose, over 100 image pairs are evaluated in each measurement and for each considered layer, mean and median particle velocity are calculated (see Chapter 4).

Prospectively, the software will have to deal with various types of experimental setups:

- different liquids with distinct consequences for the optical system (e.g. refractive index, fluorescent effects of the liquid itself)
- different backplate structures with distinct consequences for the optical system (microchannels in the plate can leave artifacts on the images)

Furthermore, illumination changes within two images of a subsequent recording appear. Potential reasons for this include the superposition of both laser pulses and an imperfect justification of both lasers. Though the laser energy can be manually adjusted as compensation, in practice it has been found that perfect brightness alignment of both recordings of an image pair is unlikely to achieve.

Therefore, in conclusion, the algorithms used are required to be robust and adaptable towards fluctuations in image brightness, contrast and noise in the raw data.

1.1.3 Contribution

Within the scope of this thesis, automatic particle detection algorithms as well as particle tracking algorithms are studied with regard to their application in a μ PTV framework. In particular, an advanced shape detection method customized for images acquired by the Single Aperture Defocusing μ PTV technique is introduced. As the number of successfully evaluated particles

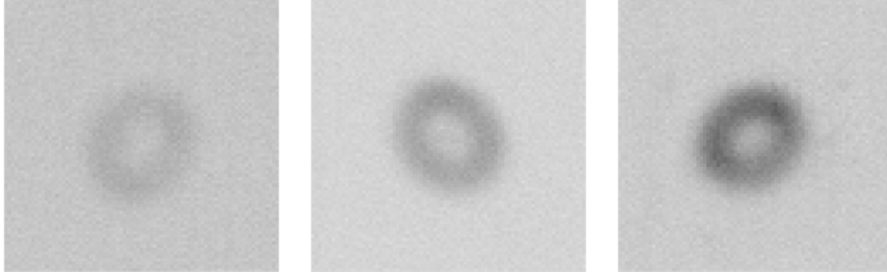


Figure 1.9: Particles from different μ PTV measurements. Left: Particle from reference image, Middle and Right: Particles from (distinct) raw images. Gray values inverted.

per image pair is desired to be maximal, the approach not only takes separated particles into account but also permits the dissection of particle clusters. Furthermore, a particle matching method independent from image contrast has been developed and physical properties of the flow are exploited by the particle tracking algorithm presented.

Employing a simplified representation of the rotationally symmetrical optical system in use, Wu et al. verified the theoretical circular particle symmetry in numerical simulations [86]. In the practical situation, however, this symmetry can be disturbed by various impacts such as an uneven gas-liquid interface, asymmetric aberrations of the optical system and camera noise. Figure 1.9 shows examples of non circular symmetric particles extracted from experimental μ PTV measurements and indicates that the occurring particle structures can be distorted. In addition, different contrast conditions are illustrated. Thorough investigation of the available raw image data suggests that the shape distortions of particles belonging to the same layer are (almost) equivalent except for rotations, cf. Figure 1.9. Therefore, the features used in the matching algorithm developed are chosen to be rotationally invariant.

1.2 Results

The methodology introduced in this thesis is evaluated by means of manually annotated experimental data as well as automatically generated synthetic data. For both datasets, the matching algorithm proposed is compared to standard methods based on cross-correlation. Furthermore, the particle linking process (on image pairs) is evaluated separately from the particle detection algorithm (on single images). The performance metrics employed to quantify their accuracy are precision, recall and a weighed mean of these two values, the F-score.

For the experimental dataset, the algorithm proposed reaches an F-score of 0.907 for particle detection and 0.901 for the whole μ PTV algorithm. In contrast, the F-score for the standard cross-correlation approach is 0.855 for particle detection and 0.853 for the whole algorithm.

Naturally, the results for the synthetic dataset are superior. In addition, the discrepancy between both techniques is minimal: for the algorithm proposed, F-scores of 0.986 (particle detection) and 0.985 (whole algorithm) are achieved while the corresponding values for the cross-correlation technique are 0.982 and 0.977, respectively.

Furthermore, the average computation time of the whole μ PTV algorithm is more than twice as short with the new particle matching method.

1.3 Outline

This chapter depicted the purpose and motivation of the thesis. The remainder of the thesis is structured as follows:

Chapter 2 - State of the Art provides an overview of existing methods used in different automatic particle detection and tracking applications. According to the methodology of the algorithm developed, first, image pre-processing techniques are discussed. Then, traditional approaches for particle detection and matching are outlined and classified into two different sorts of algorithms. Finally, common two-frame tracking schemes are reviewed.

Chapter 3 - Methodology gives background information about existing methods employed in this thesis. For image pre-processing, the field of (nonlinear) diffusion filters is introduced and theoretical and numerical properties are analyzed. As for the particle detection process, a summary of the Canny edge detector is provided and the concepts of different Hough Transforms are described in detail. Ultimately, the particle matching section outlines the Normalized Cross-Correlation coefficient while the fourth section of this chapter deals with Linear and Integer Programming problems. Each section concludes with an explanation of the proposed adaptations to the described method(s) for the use in the current μ PTV application.

Chapter 4 - Evaluation and Results presents the experiments based on an experimental and a synthetic μ PTV dataset, respectively. Subsequent to the introduction of the performance metrics and the methods compared, the evaluation and analysis for each dataset are listed.

Chapter 5 - Conclusion and Outlook concludes this thesis and gives an outlook to potential improvements and extensions of the methods proposed.

State of the Art

In this chapter, state-of-the-art methods for the main components of the software tool developed are reviewed: image pre-processing, particle detection and matching as well as particle tracking in gray-scale images. In conventional (micro-)PIV/PTV applications, the shape of the relevant particles is not subject to change [61]. Due to the special characteristics of the Single Aperture Defocusing μ PTV technique applied in the current project, the successful recognition and classification of various particles types is crucial, as outlined in Chapter 1. Therefore, the methods discussed in the following not only refer to pertinent PIV-related literature, but rather originate in different technical applications where particle detection is required. The diversity of particle shapes and sizes occurring in the reviewed sources is demonstrated in Figure 2.1(a)-(d).

In this chapter, first an overview of image pre-processing approaches in particle detection applications is given.

In Section 2.2, selected methods for automatic particle detection applied in diverse technical domains, are characterized. In addition, their potential contributions to a successful particle matching are contemplated where possible.

The final component of the algorithm developed is to associate detected particles in two frames. Therefore, a short summary of established PTV/ μ PTV tracking algorithms is presented in Section 2.3.

Table 2.1 lists the sources surveyed in this chapter. For each referenced work, it is specified where methods concerning parts of the algorithm developed (pre-processing, particle detection/matching and particle tracking) are explicitly described. Furthermore, the application field of the source is given.

2.1 Image Pre-Processing Methods

In literature, miscellaneous pre-processing strategies are applied concerning automatic particle detection. In the following, a selection of commonly performed approaches is listed. The classification into four categories, according to the size of the pixel neighborhood that is used for the calculation of a pixel's new brightness value, has been adopted from [73]:

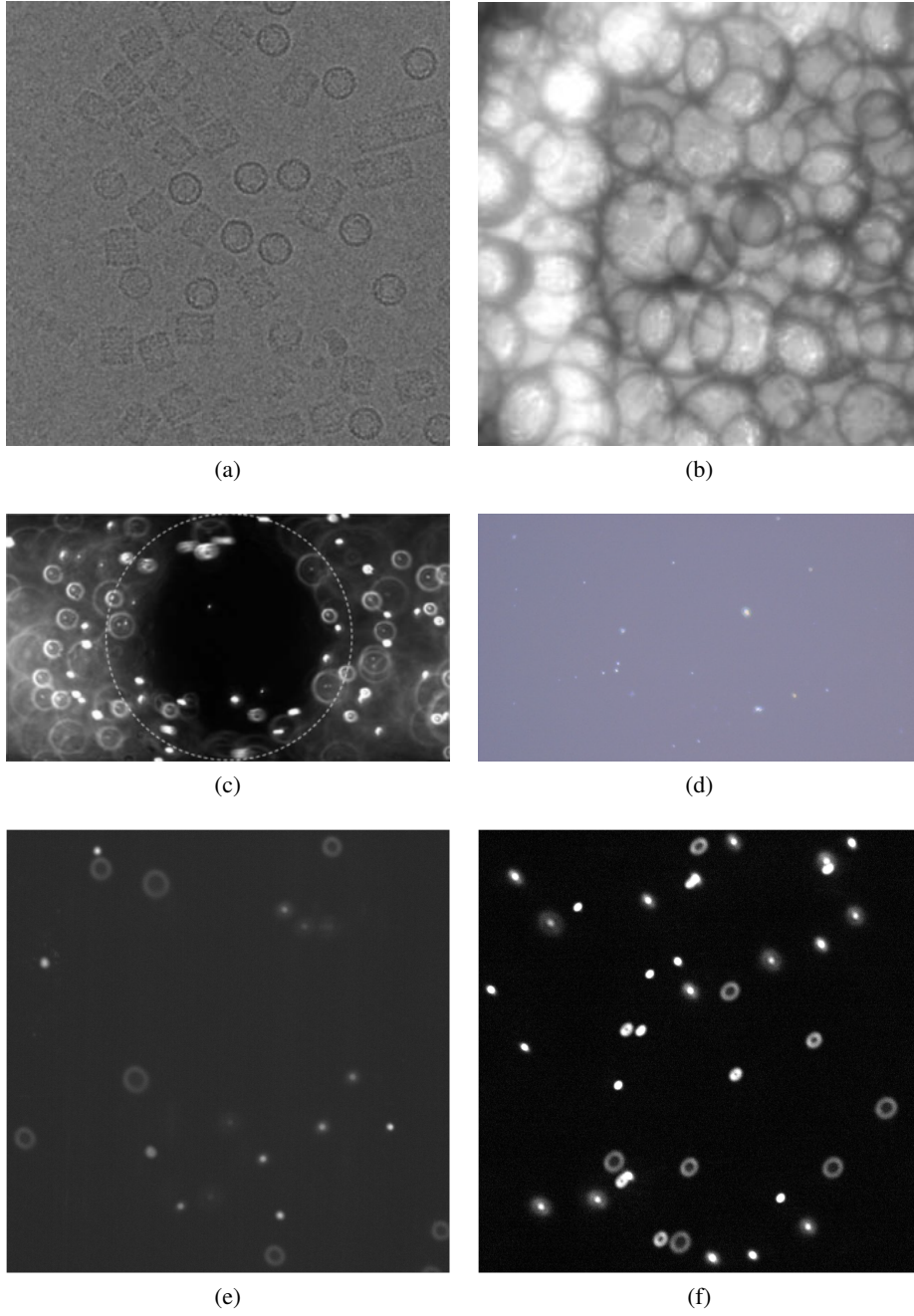


Figure 2.1: Comparison of processed images: (a) Particle image from dataset used in [74,90,91], (b) Particle image from [44], (c) Particle image from [54], (d) Particle image from [30], (e) Experiment in LAYER-1 liquid, (f) Experiment in soap water.

Table 2.1: Overview of surveyed sources with respect to potential contribution to current application.

Source(s)	PP ^a	PD/PM ^b	PT ^c	Goal
[54, 56, 59]	✓	✓	✓	Single Aperture Defocused μ PTV
[4, 61]	✓	✓	✗	PIV textbook
[1, 2, 33, 45, 48, 70, 74, 89–91]	✓	✓	✗	Cryo-Electron Microscopy
[44]	✓	✓	✗	Automatic drop detection
[30, 38]	✓	✓	✗	Asbestos detection analysis
[65]	✓	✓	✓	Tracking Algorithm in cell biology
[63]	✗	✗	✓	Variational Fluid Motion Estimation
[31, 50, 53, 57, 75]	✗	✗	✓	Tracking Algorithms for PTV

^a Pre-Processing

^b Particle Detection/Matching

^c Particle Tracking

1. **Pixel brightness transformations** compute each pixel’s new brightness based on a given transformation of the original brightness values.
2. **Geometric transformations** map each pixel to a new position, e.g. rotations, translations and scalings, including brightness interpolations.
3. **Local pre-processing methods** use a (small) neighborhood of each pixel to create its new brightness value.
4. **Image restoration** processes the nature of the whole image for the purpose of a reconstruction of the original image.

Paschke et. al [56] enhance the image contrast in their μ PTV application with the aid of histogram stretches, a histogram equalization and a Gamma correction (all class 1). Other pre-processing methods applied in PIV/PTV systems include background removal [4], either by subtracting a previously recorded background image (class 1), or a uniform-/median-/minimum filtered image (class 3). In general, the majority of publications addressing automatic particle detection listed in Table 2.1 use a type of background subtraction prior to the detection algorithm.

Adiga et al. [1,2] additionally propose the use of a nonlinear filter (class 4) which is based on the solution of a partial differential equation and “*has been proved to be effective in achieving the desired reduction in noise*” [2]. A similar, further developed filter using the so-called Diffusion Tensor (DT) has recently been applied by Joubert et al. [33]. This method originates in the famous Perona-Malik model [58] and is closely related to the filter of Catté, Lions, Morel and Coll (CLMC) [14]. Extensive theoretical and numerical investigation concerning the last two mentioned filters has been conducted by Weickert et al. [80, 81, 83].

In the context of diffusion filters (further described in Section 3.1), the smoothing of an image by convolution with a 2-dimensional Gaussian (with variable parameter σ) represents the

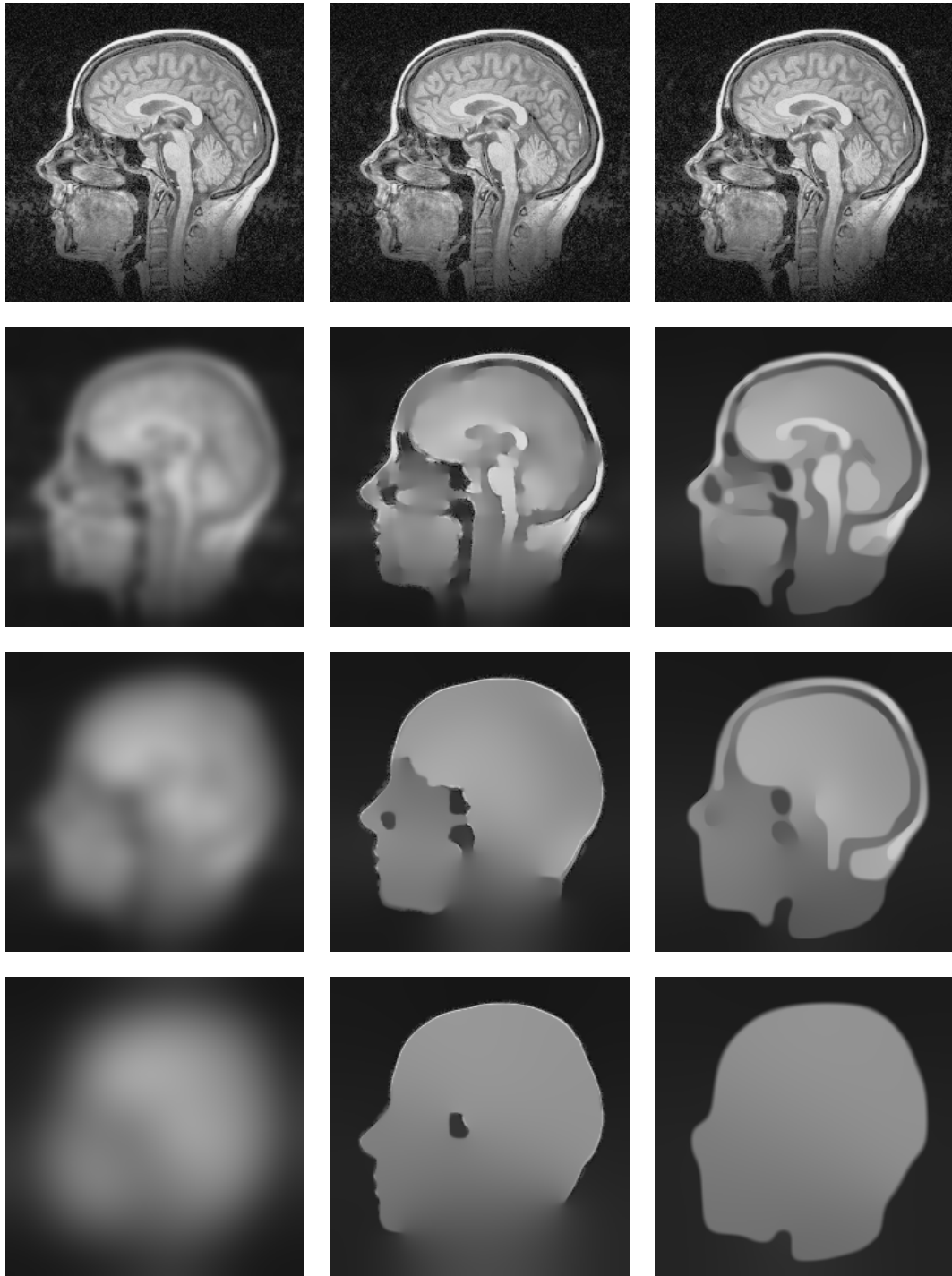


Figure 2.2: Evolution of an MRI slice under different diffusion filters. Top: Original image, Left Column: Image evolution under LD filter, Middle Column: Image evolution under CLMC filter, Right Column: Image evolution under DT filter. Images taken from [80].

Linear Diffusion (LD) approach [80], see Section 3.1.1. Figure 2.2 displays different outcomes in the processing of an MRI slice with the LD, CLMC and DT filter respectively. The top row contains versions of the original raw image and the images below accrue from varying parameter choices. It is apparent that the first method is inferior to the latter two and the parameter selection process is crucial for the amount of structural information preserved in the image.

Other pre-processing techniques successfully applied in the literature listed in Table 2.1 include the self quotient image method where the raw image is pointwise divided by a Gaussian smoothed version of itself [44] and Bayesian wavelet denoising algorithms [74].

2.2 Particle Detection and Particle Matching Methods

Manual particle selection becomes an exhausting task when dealing with hundreds of images. Therefore, designing problem-fitted algorithms for automatic particle detection constitutes a major challenge in those applications (cf. Table 2.1).

In the scope of Cryo-Electron Microscopy (Cryo-EM), Nicholson and Glaeser [48] give a review of methods for automatic particle picking. The techniques presented are classified as follows:

- Methods based on template matching
- Methods based on edge-detection
- Methods based on intensity comparisons
- Texture-based methods
- Methods based on neural networks

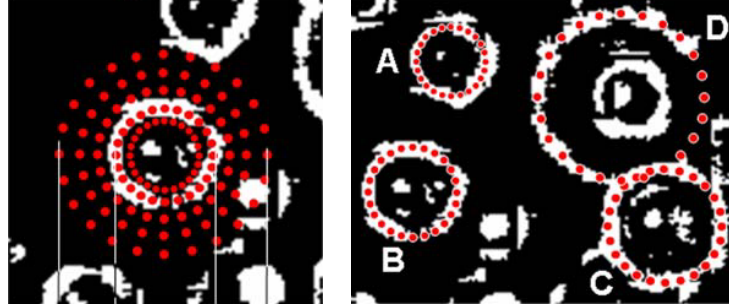
Another overview from this area of application is done by Zhu et al. [91] who present the results of a program dedicated to the development of fully automated particle selection in the Cryo-EM field. In this *Multidisciplinary Workshop on Automatic Particle Selection for cryoEM*, the algorithms of 10 research groups were evaluated on a common dataset. The techniques are grouped into two more general classes:

- Methods based on template matching (Class 1)
- Feature-based methods (Class 2)

Shortly summarized, algorithms belonging to Class 1 use generated template images of particles for the detection process by applying a correlation technique while Class 2 methods extract particle locations from an image by recognizing salient local or global features [91]. The latter therefore potentially need an additional matching process if a further particle classification is required. It should be noted that techniques can be considered as being “between” those two classes [74]. In this context, Mallick et al. [45] and Sorzano et al. [74] introduced a third class for their machine learning-based methods.



(a) Detected area of asbestos particles. Images taken from [38].



(b) Identification of particles on a binarized μ PTV image. Images taken from [54].

Figure 2.3: Particle detection algorithms by (a) Kuba et al. [38] and (b) Park and Kihm [54].

The algorithm presented by Maass et al. [44] provides an example for a template matching method (Class 1) in the field of automatic drop detection in multiphase systems. Drop locations with a specific radius are extracted as local probability maxima in a cross-correlation image with the sought-after structure.

Feature-based methods (Class 2) presented in [91] for instance apply efficient Hough Transforms [90] or Voronoi Diagrams and distance transforms [89] for the detection of specific geometric shapes. Another Class 2 approach is the method developed by Short [70]: reference objects are described by elementary properties such as the radius of gyration and intensity sum/variance on circular sectors and concentric rings, respectively. Subsequently, the image area is scanned with a moving square window and the congruence of the corresponding features indicates the presence of a particle searched for.

Another learning-based approach where the features are trained by a random forest classifier is studied by Joubert et al. [33].

In the field of asbestos detection analysis, microscope images of building material are examined for asbestos particles. Since their size, shape and color pattern are not a-priori known [38], the goal is to automatically detect and count all occurring structures. Both reviewed techniques [30,38] initially divide the image into smaller subregions (30×30 in [30], 10×10 in [38]) which are then classified into particle and background areas, either by clustering the RGB variance values [30] or by application of a One-Class Support Vector Machine on the gray-scale histograms of the regions [38]. Naturally, the particle areas are refined in the end, e.g. by a thresholding operation (see Figure 2.3(a)).

In their Single Aperture Defocused μ PTV application, Park and Kihm [54] determine the

flow velocity field over a $95\text{ }\mu\text{m}$ diameter sphere inside a $100\text{ }\mu\text{m}^2$ square channel. For the detection of ring-shaped particles, the brightness values in the original image, as shown in Figure 2.1(c), initially are subject to adaptive thresholding. A location is then identified as a particle center if the ratio of foreground pixels of the binarized image on a sampled, surrounding circle exceeds a certain threshold. Figure 2.3(b) indicates this procedure.

In a similar way, Paschke et al. [56] implement binarization by performing Canny edge detection. Subsequent to the calculation of particle center coordinates, matching with reference images is performed by means of a cross-correlation approach.

2.3 Particle Tracking Methods

In this section, particle linking algorithms for the tracking application in PTV/ μ PTV images are addressed. An extensive description of the classical PIV approach which determines discrete flow velocity fields based on a statistical cross-correlation technique is provided in the PIV textbooks by Adrian and Westerweel [4] and Raffel et al. [61]. Those methods do not involve the extraction of particle coordinates but rather operate directly on gray-scale images.

Pereira et al. [57] state that in “*conventional PTV, most methods use three or four consecutive flow images (multi-frame methods) for accurate velocity measurements*”, e.g. the algorithms described in [26, 52]. However, since the laser in the current system (cf. Section 1.1.1) solely allows two consecutive pulses per recording, a two-frame tracking algorithm is required. This implies that, aside from theoretically estimated flow characteristics, no knowledge concerning the expected particle motion is available from previous frames. Nevertheless, a cross-correlation based technique can yield a rough estimate of the mean particle velocity between both images [57].

A basic two-frame tracking scheme is the nearest neighbor method, which establishes links between particles such that their distances in both frames are minimal. In literature, different variations have been proposed, two of which are presented and evaluated by Pereira et al. [57] with regard to the implementation in a 3D Defocusing PTV/ μ PTV application. In the first method, originally presented by Labonté [39], a neural network acts in an iterative, self-organizing manner as a “*pre-sorting routine for the nearest neighbour method*” [57], providing a positive impact on the standard technique. Secondly, the so-called *relaxation method*, first employed by Baek and Lee [10], postulates continuity in the velocity vector field by formulating the so-called *quasi-rigidity condition*: particles in a common neighborhood are subject to similar displacements in the flow. According to this assumption, particle link probabilities are computed and iteratively updated until convergence.

The evaluation of those two methods has been conducted on a specified synthetic 3D Burgers’ vortex flow field, sampled on a total of $N_{cl} = 5000$ vectors, as shown in Figure 2.4(a). The top image in Figure 2.4(b) displays the recovery ratio $\eta_r = M_{cl}/N_{cl}$, where M_{cl} denotes the number of correctly tracked particle pairs, for the different investigated methods. The tracking parameter Φ on the x -axis is inversely proportional to the time (and in further consequence, the particle displacement) between both frames. Finally, the bottom image in Figure 2.4(b) shows the mismatch rate $\eta_m = M_{sl}/M_l$ which represents the ratio of the number of incorrect links M_{sl} divided by the number of overall identified links M_l .

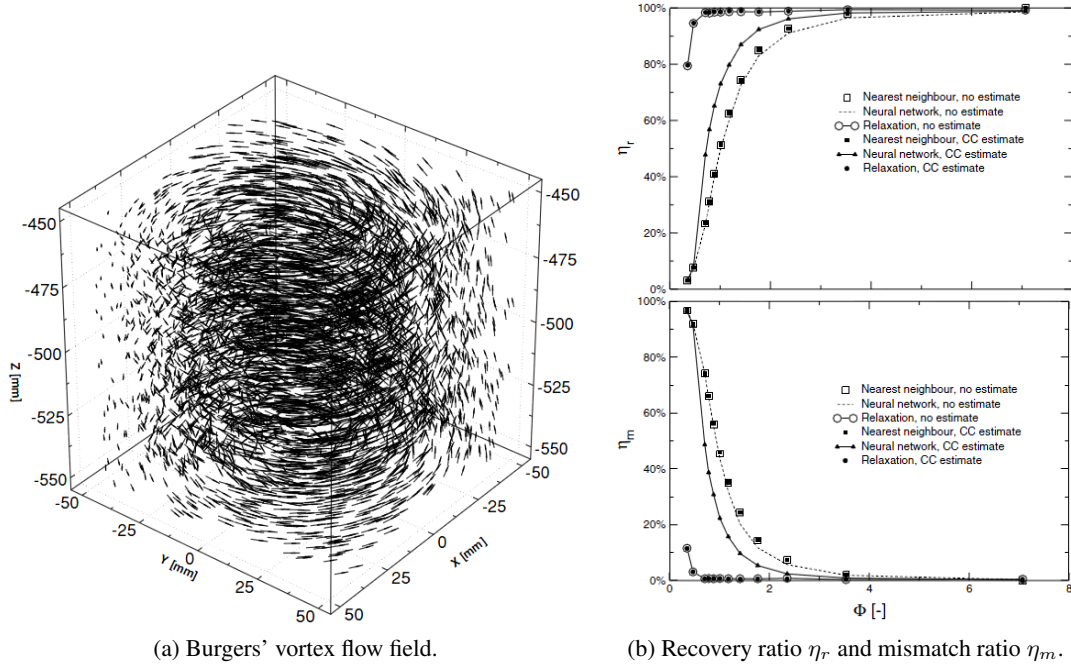


Figure 2.4: Evaluation of PTV tracking algorithms. Figures taken from [57].

Different alterations to the relaxation method have been proposed, as mentioned by Jia et al. [31], who themselves very recently introduced two beneficial modifications for a 2D framework: firstly, the identification of the neighborhood in the quasi-rigidity condition is based on the Delaunay Tessellation of the set of particle positions. As a result, the calculation efficiency is enhanced, especially for increasing particle numbers. Secondly, their bidirectional algorithm developed enhances the computation of the no-match probability, thus improving the quality of the particle pairing process.

Ruhnau [63] was the first to apply variational methods in a PIV/PTV framework. The whole velocity field is estimated by the minimization of a functional composed of a local data term and a global regularization term. While its results are shown to surpass the four-frame method described in [26] and the classical relaxation method [10], its practical use is limited due to the high computational cost [31].

In 1996, Dorigo et al. [20] introduced the ant colony optimization algorithm as a “*viable new approach to stochastic combinatorial optimization*”. It is inspired by the behavior of a group of ants determining the shortest path between a food source and their nest. According to [50, 53], the method was first applied in the PTV particle association process by Takagi [75] in 2007. In the last three years, the algorithm has been improved by Ohmi et al. [50] by adopting a quasi-rigidity condition and extended by Panday et al. [53] for the use of image data acquired by Stereoscopic PTV measurements.

As pointed out by [31, 50, 57], other common tracking approaches include the binary cross-

correlation method [76] which is related to the classical PIV method and the application of different cost functions [25, 36] and genetic algorithms, respectively [51, 69].

2.4 Summary

This chapter gave a review of state-of-the-art literature related to the main components needed in a Defocusing μ PTV software. First, pre-processing methods which improve the image quality for the further processing steps are illustrated. There is no ambition to summarize the broad range of existing image pre-processing techniques (for an overview, see the respective chapter in [73]) since the approaches used in the literature discussed in this chapter (cf. Table 2.1) are entirely sufficient for the application on μ PTV recordings. Subsequently, different particle detection algorithms are discussed and two algorithm classification schemes are introduced. Section 2.3 finally is focused on particle tracking methods which are actually implemented in 2- or 3-dimensional PTV/ μ PTV applications.

Methodology

The Micro Particle Tracking Velocimetry approach presented in this work consists of four components: image pre-processing, particle detection, particle matching and particle tracking. In this chapter, the concepts of the basic algorithms used and the reasons and consequences of any modifications applied are illustrated.

3.1 Step 1: Image Pre-Processing

The principal goal of image pre-processing is the minimization of any influences that obstruct the qualitative and/or temporal performance of the subsequently applied algorithms. For the current μ PTV image data, this primarily involves the removal of camera noise and background disturbances. The image background area (e.g. arising from reflections or adherent particles on the backplate from previous measurements) is labeled on an empty recording of the scene in the absence of fluid flow and ignored in the detection process. In addition, in order to create contrast alignment between both recordings of a μ PTV measurement, histogram matching is performed. This is a simple routine where the cumulative image histograms of two images are equalized [73]. For noise elimination, the nonlinear diffusion filter proposed by Catté, Lions, Morel and Coll [14] is employed. Previously, similar techniques have been used to denoise low-quality raw images in automatic particle detection applications by [1, 33].

This section introduces the concept of diffusion filters and in further consequence provides insight into the most common variants presented in literature [14, 58, 80, 81, 83]. Crucial theoretical results of the implemented method are presented in Section 3.1.2.2 while example images, which demonstrate its actual performance on a μ PTV dataset, are shown in Section 3.1.3.

For the sake of completeness and clarity, several notations occurring in the remaining course of this work are specified.

Definition 1. Let $n \in \mathbb{N}$, $\Omega \subset \mathbb{R}^n$ and $1 \leq p < \infty$. Then we define the function spaces

$$\begin{aligned} C(\Omega) &:= \{f : \Omega \rightarrow \mathbb{R} \mid f \text{ continuous}\} \\ C^\infty(\Omega) &:= \{f : \Omega \rightarrow \mathbb{R} \mid f \text{ infinitely often continuously differentiable}\} \\ L^p(\Omega) &:= \left\{f : \Omega \rightarrow \mathbb{R} \mid \int |f(x)|^p dx < \infty\right\} \\ L^\infty(\Omega) &:= \{f : \Omega \rightarrow \mathbb{R} \mid \text{ess sup}_\Omega f < \infty\} \text{ where} \\ &\quad \text{ess sup}_\Omega f := \inf \{c > 0 \mid |f(x)| \leq c \text{ for almost every } x \in \Omega\}. \end{aligned}$$

Moreover, unless stated otherwise, the norm of a vector $x = (x_1, \dots, x_n)^T$ denotes its l^2 -norm

$$\|x\| := \|x\|_2 := \sqrt{\sum_{i=1}^n x_i^2}.$$

Apart from physics and engineering sciences, Partial Differential Equations (PDEs) have been successfully applied in image processing and computer vision for image smoothing and restoration purposes for over two decades [9, 80]. Diffusion filters represent a PDE-based approach used to enhance low-quality image data. Weickert [80] explains the term *diffusion* as “a physical process that equilibrates concentration differences without creating or destroying mass.” Mathematically, this is expressed by Fick’s law:

$$j = -D \cdot \nabla u. \quad (3.1)$$

“This equation states that a concentration gradient ∇u causes a flux j which aims to compensate for this gradient” [80] while the *diffusivity* D , in general a positive definite symmetric matrix, describes the relation between ∇u and j . The conservation of mass over time is characterized by the *continuity equation*

$$u_t + \text{div } j = 0, \quad (3.2)$$

where $u_t := \frac{\partial u}{\partial t}$ denotes the time derivative of u . By combining (3.1) and (3.2), the *diffusion equation*

$$u_t = \text{div}(D \cdot \nabla u) \quad (3.3)$$

is derived. In the context of image processing, the values of the concentration u are identified with the pixel brightness values of a gray-scale image. For the rest of this section, let $\Omega := (0, a_1) \times (0, a_2) \subset \mathbb{R}^2$ with $a_1, a_2 > 0$ be the rectangular domain of an image f , which is described as a bounded, scalar function $f \in L^\infty(\Omega)$. Typical gray-scale pixel representations are unsigned 8-bit integers (values between 0 and 255) and floating point (values between 0 and 1) [9]. A diffusion filtered image is then obtained by solving the diffusion equation (3.3) forward in time until a fixed stopping time τ , with the original image f as initial condition at $t = 0$. As will be motivated below, τ plays the role of a “scale parameter: larger values lead to simpler image representations” [83].

The classification of diffusion filters is based on the form of the diffusivity: if D is depending on the evolving image u itself, the filter is called nonlinear; otherwise we speak of a linear filter [80]. All diffusion filter methods described in this work are designed for gray-value images. It should be noted, however, that nonlinear diffusion filters have also been successfully extended to vector-valued images such as RGB or multi- and hyperspectral images [21, 80, 82].

3.1.1 Linear diffusion filtering

This section is focused on the homogeneous, linear diffusion process, i.e. where the diffusivity is constant over the whole image domain. Here, the scalar diffusivity $D = 1$ is considered. Evans [24] shows that for $f \in C(\mathbb{R}^n) \cap L^\infty(\mathbb{R}^n)$, the resulting heat equation

$$\begin{aligned} u_t &= \Delta u, & \text{on } \mathbb{R}^n \times (0, T), \\ u(x, 0) &= f(x) & \text{on } \mathbb{R}^n, \end{aligned} \quad (3.4)$$

possesses the solution

$$u(x, t) = (K_{\sqrt{2t}} * f)(x), \quad (3.5)$$

where K_ρ denotes the Gaussian with standard deviation ρ :

$$K_\rho(x) := \frac{1}{2\pi\rho^2} \exp\left(-\frac{\|x\|^2}{2\rho^2}\right). \quad (3.6)$$

Additionally, the uniqueness of the solution is ensured under the condition that solely functions satisfying the growth estimate $|u(x, t)| \leq Ae^{a\|x\|^2}$ (for some $a, A > 0$) are tolerated [24]. Moreover, due to $K_\rho \in C^\infty(\mathbb{R}^n)$ and the differentiation property of the convolution operator, the function $u(\cdot, t)$ is infinitely often differentiable for all positive t . As a consequence of (3.5), linear diffusion filtering with stopping time τ is equivalent to the convolution with a Gaussian with standard deviation $\rho = \sqrt{2\tau}$.

In *scale-space theory*, an image f is represented as a one-parameter family $\{T_t f : t \geq 0\}$ of gradually simplified (smoothed) versions of itself. This simplification property is expressed in different axioms in literature [23, 43, 80]. T_t is denoted as the *scale-space operator* generating a scale space. The family generated by

$$T_t := f \mapsto K_t * f$$

is called the *Gaussian scale-space* and is related to the solution to the heat equation described in (3.5). It fulfils the desired axioms [23, 80] and represents the “*historically first and best investigated scale-space*” [80]. Convolution with Gaussian kernels represents an important class of operators in the field of image processing, as it “*can be used as basis to solve a large variety of visual tasks, including feature detection, feature classification, stereo matching, motion descriptors, shape cues, and image-based recognition*”. [43].

A major drawback of linear diffusion filtering in the context of image pre-processing is the fact that not only noise is smoothed but also information features such as edges are blurred [80]. Moreover, edges detected at a coarse scale can be dislocated with respect to their location at a finer scale [80]. Those problems can be overcome by the use of a nonlinear diffusion filter.

3.1.2 Nonlinear diffusion filtering

The basic idea of nonlinear diffusion filtering is to prevent the diffusion process from smoothing across edges, where the “important” image information is stored. Edges are subjectively perceived as image locations where the intensity function u changes abruptly [73]. The diffusivity in (3.3) is therefore chosen to be dependent on the image gradient: $D = D(\nabla u)$.

3.1.2.1 The Perona-Malik model

The first nonlinear diffusion approach was published by Perona and Malik [58] in 1990. In their model, edge locations are indicated by high values of $\|\nabla u\|^2$. The diffusivity is given by a positive, monotonically decreasing, scalar function $D = g(\|\nabla u\|^2)$, which satisfies the conditions

$$g(0) = 1, \lim_{s \rightarrow \infty} g(s) = 0,$$

yielding the famous Perona-Malik equation

$$u_t = \operatorname{div}(g(\|\nabla u\|)\nabla u). \quad (3.7)$$

The following diffusivity functions have been originally proposed [58, 83] ($\lambda > 0$):

$$g(s^2) := \frac{1}{1 + \frac{s^2}{\lambda^2}}, \quad (3.8)$$

and ($\lambda > 0$, $c = 3.315$)

$$g(s^2) := \begin{cases} 1 & s^2 = 0, \\ 1 - \exp\left(-\frac{c}{(s/\lambda)^8}\right) & s^2 > 0. \end{cases} \quad (3.9)$$

In both variants, λ may be called a “contrast parameter” [83]: image regions where $\|\nabla u\| < \lambda$ are smoothed while $\|\nabla u\| > \lambda$ indicates the presence of an edge which may even be enhanced in the diffusion process [79].

Although the numerical results are described as “*visually very impressive*” [80], two problems emerge: firstly, evidence was found that the Perona-Malik process is theoretically ill-posed, i.e. unstable with respect to perturbations of the initial image [80]. Secondly, large gradient magnitudes in the original image can also be caused by noise which leads to the fact that the respective location is “misinterpreted” as an edge and therefore preserved instead of being smoothed.

These characteristics give rise to the following regularizing modification of (3.7).

3.1.2.2 The continuous CLMC equation

Catté, Lions, Morel and Coll [14] propose a spatial regularization by replacing the term $g(\|\nabla u\|^2)$ by a Gaussian smoothed version $g(\|\nabla u_\sigma\|^2)$ where

$$u_\sigma = \tilde{u} * K_\sigma \quad (3.10)$$

with $\sigma > 0$ and \tilde{u} is an extension of u from Ω to \mathbb{R}^2 in order for the convolution to be well defined. For this, the image is mirrored on its boundary which mathematically results in the adoption of a homogeneous Neumann boundary condition. In conclusion, this leads to the problem

$$\begin{aligned} u_t &= \operatorname{div}(g(\|\nabla u_\sigma\|^2)\nabla u) \quad \text{on } \Omega \times (0, T], \\ u(x, 0) &= f(x) \quad \text{on } \Omega, \\ \partial_n u &= 0 \quad \text{on } \partial\Omega \times (0, T], \end{aligned} \tag{3.11}$$

where $\partial_n u$ denotes the normal derivative of u . Analogous to [83], (3.11) henceforth will be called the CLMC equation. There exists a unique solution to (3.11) (in the distributional sense) which is infinitely often differentiable on $\bar{\Omega} \times (0, T)$ [14] and depends continuously on the initial data [80]. In addition, Weickert [80] proves that the solution $u(x, t)$ of (3.11) fulfils several scale-space properties, thus generates its own nonlinear scale-space framework. In the following, the principal results are listed.

Let T_t be the scale-space operator which maps an image f to the family of images created by the nonlinear diffusion process described by the CLMC equation (3.11):

$$T_t f := u(\cdot, t), \quad t \geq 0. \tag{3.12}$$

Then the following scale-space properties hold:

Gray-level shift invariance: Let $C \in \mathbb{R}$, then

$$T_t(f + C) = T_t(f) + C \quad \forall t \geq 0.$$

Conservation of average gray value: Let $\mu = \frac{1}{|\Omega|} \int_{\Omega} f(x) dx$ be the average gray level of f , then

$$\frac{1}{|\Omega|} \int_{\Omega} (T_t f)(x) dx = \mu \quad \forall t \geq 0.$$

Isometry invariance: Let $R \in \mathbb{R}^{2 \times 2}$ be an orthogonal transformation and define $Rf(x) := f(Rx)$, then

$$T_t(Rf) = R(T_t f) \quad \forall t \geq 0.$$

Extremum principle:

$$\operatorname{ess\,inf}_{\Omega} f \leq T_t f(x) \leq \operatorname{ess\,sup}_{\Omega} f \quad \forall x \in \Omega \quad \forall t > 0$$

Nonenhancement of local extrema: Let u be the unique solution of (3.11), $t_0 > 0$ and x_0 a local extremum of $u(\cdot, t_0)$ with nonvanishing Hessian, then

$$\begin{aligned} \partial_t u(x_0, t_0) &< 0 \quad \text{if } x_0 \text{ is a local maximum,} \\ \partial_t u(x_0, t_0) &> 0 \quad \text{if } x_0 \text{ is a local minimum.} \end{aligned}$$

Image simplification for increasing t : Let M be the image with constant gray value μ , then

$$\lim_{t \rightarrow \infty} \|T_t f - M\|_{L^p(\Omega)} = 0 \quad \text{for } p \in [1, \infty).$$

Furthermore, the function

$$t \mapsto \|T_t f\|_{L^p(\Omega)} \quad \text{for } p \in [2, \infty)$$

is decreasing in t . As a consequence, the *image energy* $\|T_t f\|_{L^2(\Omega)}^2 = \int_{\Omega} |u(x, t)|^2 dx$ is reduced by the progressing diffusion process. Therefore, Parseval's theorem, which states that the L^2 -norm of a square integrable function is equal to the L^2 -norm of its Fourier transform, indicates that for $t \rightarrow \infty$, low frequency components are dominating in the processed image $T_t f$. In addition, the *image entropy*

$$t \mapsto S[T_t f] := - \int_{\Omega} u(x, t) \ln(u(x, t)) dx \quad \text{if } \text{ess inf}_{\Omega} f > 0,$$

which represents a measure of uncertainty and missing information, is increasing in t [80].

3.1.2.3 The discrete CLMC equation

Due to the discrete nature of digital images, it is desirable that the numerical discretization of (3.11), with respect to the spatial dimensions as well as the scale parameter t , fulfils similar scale-space properties as the continuous framework listed above. In this context, Weickert [80] established a discrete scale-space theory for a broad class of image filtering processes, as specified in Definition 2. In the following, the substantial results are sketched.

Instead of its “natural” characterization as a matrix $B \in \mathbb{R}^{m \times n}$, a discrete image is represented by a vector $F \in \mathbb{R}^N$, $N = m \cdot n$ whose elements f_j , $j \in J := \{1, \dots, N\}$ describe the gray values at each pixel. For this purpose, consider the index transformation (see Figure 3.1):

$$(i, j) \mapsto \begin{cases} (i-1)n + j, & i \text{ odd}, \\ in + 1 - j, & i \text{ even}. \end{cases} \quad (3.13)$$

Definition 2 (The discrete filter class (P_d)). *Let $f \in \mathbb{R}^N$. Calculate a sequence $(u^{(k)})_{k \in \mathbb{N}_0}$ of processed versions of f by means of*

$$\begin{aligned} u^{(0)} &= f, \\ u^{(k+1)} &= Q(u^{(k)})u^{(k)}, \quad \forall k \in \mathbb{N}_0, \end{aligned} \quad (3.14)$$

where $Q = (q_{ij})$ has the following properties:

(D1) *continuity in its argument:* $Q \in C(\mathbb{R}^N, \mathbb{R}^{N \times N})$

(D2) *symmetry:* $q_{ij}(v) = q_{ji}(v) \quad \forall i, j \in J, \forall v \in \mathbb{R}^N$

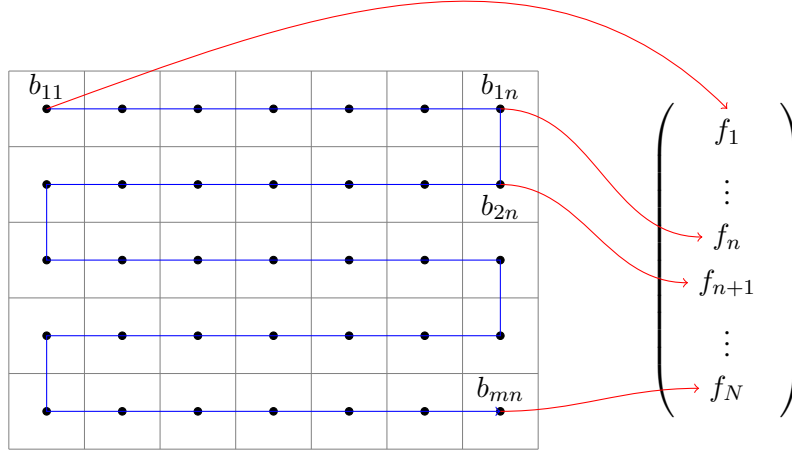


Figure 3.1: Image notation according to (3.13).

(D3) unit row sum: $\sum_{j=1}^N q_{ij}(v) = 1 \quad \forall i \in J, \forall v \in \mathbb{R}^N$

(D4) nonnegativity: $q_{ij}(v) \geq 0 \quad \forall i, j \in J, \forall v \in \mathbb{R}^N$

(D5) irreducibility for all $v \in \mathbb{R}^N$, i.e. for any $i, j \in J$ there exist $k_0, \dots, k_r \in J$ with $k_0 = i$ and $k_r = j$ such that $q_{k_p k_{p+1}} \neq 0$ for $p = 0, \dots, r-1$

(D6) positiv diagonal: $q_{ii}(v) > 0 \quad \forall i \in J, \forall v \in \mathbb{R}^N$

The system (3.14) combined with the conditions (D1)-(D6) is referred to as filter class (P_d) .

For each $f \in \mathbb{R}^N$, a filter associated to the class (P_d) creates a distinct sequence $(u^{(k)})_{k \in \mathbb{N}_0}$. In Theorem 1, the beneficial effects of such filters for the purpose of image pre-processing are demonstrated.

Theorem 1. Let $f \in \mathbb{R}^N$, $\mu := \frac{1}{N} \sum_{j=1}^N f_j$ be the average grey level of f and $(u^{(k)})_{k \in \mathbb{N}_0}$ be a sequence of images fulfilling the requirements of the filter class (P_d) . Then the following statements hold:

(i) For every finite k , $u^{(k)}$ depends continuously on f .

(ii) Conservation of average gray value:

$$\frac{1}{N} \sum_{j=1}^N u_j^{(k)} = \mu \quad \forall k \in \mathbb{N}_0$$

(iii) Extremum principle:

$$\min_{j \in J} f_j \leq u_i^{(k)} \leq \max_{j \in J} f_j \quad \forall i \in J, \forall k \in \mathbb{N}_0$$

(iv) *Image simplification for increasing k :*

$$\lim_{k \rightarrow \infty} u^{(k)} = (\mu, \mu, \dots, \mu)^T.$$

Furthermore, the following functions are decreasing in k :

$$\begin{aligned} k &\mapsto \|u^{(k)}\|_{\ell^p} \quad \forall p \geq 1 \\ k &\mapsto \sum_{j=1}^N u_j^{(k)} \ln(u_j^{(k)}) \quad \text{if } \min_{j \in J} f_j > 0. \end{aligned}$$

Analogous to the continuous setting, the results are interpreted as decreasing energy and increasing entropy for $k \rightarrow \infty$.

Proof. The proof follows the lines of [80].

(i) This fact is a direct consequence of prerequisite (D1).

(ii) Due to the symmetry (D2) of Q , not only the row sums (D3) but also the column sums fulfil $\sum_{i=1}^N q_{ij}(v) = 1$ for all $j \in J$ and $v \in \mathbb{R}^N$. This leads to

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N u_i^{(k+1)} &= \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N q_{ij} \left(u_j^{(k)} \right) u_j^{(k)} = \frac{1}{N} \sum_{j=1}^N \left(\sum_{i=1}^N q_{ij} \left(u_j^{(k)} \right) \right) u_j^{(k)} \\ &= \frac{1}{N} \sum_{j=1}^N u_j^{(k)} \stackrel{k=0}{=} \mu \quad \forall k \in \mathbb{N}_0. \end{aligned}$$

(iii) The following inequalities hold for all $i \in J$ and $k \in \mathbb{N}_0$:

$$\begin{aligned} u_i^{(k+1)} &= \sum_{j=1}^N q_{ij} \left(u_j^{(k)} \right) u_j^{(k)} \stackrel{(D4)}{\leq} \max_{m \in J} u_m^{(k)} \sum_{j=1}^N q_{ij} \left(u_j^{(k)} \right) \stackrel{(D3)}{=} \max_{m \in J} u_m^{(k)} = \max_{j \in J} f_j, \\ u_i^{(k+1)} &= \sum_{j=1}^N q_{ij} \left(u_j^{(k)} \right) u_j^{(k)} \stackrel{(D4)}{\geq} \min_{m \in J} u_m^{(k)} \sum_{j=1}^N q_{ij} \left(u_j^{(k)} \right) \stackrel{(D3)}{=} \min_{m \in J} u_m^{(k)} = \min_{j \in J} f_j, \end{aligned}$$

where the last equality in both lines follows from induction in k .

(iv) *Idea:* It is shown that for any convex function $r \in C([\min_j f_j, \max_j f_j], \mathbb{R})$, the sequence

$$V^{(k)} := \Phi \left(u^{(k)} \right) := \sum_{i=1}^N r \left(u_i^{(k)} \right), \quad k \in \mathbb{N}_0$$

is a so-called *Lyapunov sequence*:

$$\begin{aligned} \Phi(u^{(k)}) &\geq \Phi((\mu, \mu, \dots, \mu)) \quad \forall k \in \mathbb{N}_0, \\ V^{(k+1)} &\leq V^{(k)} \quad \forall k \in \mathbb{N}_0. \end{aligned} \tag{3.15}$$

The proof of convergence to the constant image with gray value μ is then conducted with the aid of Lyapunov stability theory and requires the prerequisites (D5) and (D6). The monotonicity of the energy and entropy functionals respectively is a consequence of (3.15) with specific convex functions r . See [80] for details.

■

Due to the properties (i)-(iv), Theorem 1 suggests the designation that each filter from the class (P_d) creates a discrete scale space [80]. The next goal is to discretize the CLMC equation

$$u_t = \operatorname{div}(g(\|\nabla u_\sigma\|^2)\nabla u) = \sum_{l=1}^2 \partial_{x_l} \left(g(\|\nabla u_\sigma\|^2) \partial_{x_l} u \right). \quad (3.16)$$

For the implementation, the diffusivity function g listed in (3.9) has been selected.

In conformity with [83], the position of pixel i with value f_i is denoted as x_i , h is the (uniform) grid size and $t_k := k\tau$ are the discrete times (τ is the uniform time step size). The approximations to $u(x_i, t_k)$ and $g(\|\nabla u_\sigma(x_i, t_k)\|^2)$ are labeled u_i^k and g_i^k respectively, e_l is the unit vector in the l -th coordinate direction and $i+$ and $i-$ represent the indices of the closest pixels to x_i in positive and negative direction e_l respectively. The time derivative is approximated by a simple forward difference and a discretization of the summands in (3.16) is deduced as follows:

$$\begin{aligned} g \left(\left\| \nabla u_\sigma \left(x_i + \frac{h}{2} e_l, t_k \right) \right\|^2 \right) \partial_{x_l} u \left(x_i + \frac{h}{2} e_l, t_k \right) &\approx \frac{g_i^k + g_{i+}^k}{2} \frac{u_{i+}^k - u_i^k}{h} \Rightarrow \\ \partial_{x_l} \left(g(\|\nabla u_\sigma(x_i, t_k)\|^2) \partial_{x_l} u(x_i, t_k) \right) &\approx \frac{(g_i^k + g_{i+}^k)(u_{i+}^k - u_i^k)}{2h^2} - \\ &\quad \frac{(g_{i-}^k + g_i^k)(u_i^k - u_{i-}^k)}{2h^2}. \end{aligned}$$

In matrix-vector notation, this finally yields the *explicit* iteration scheme (3.18) for the discretization of Equation (3.16)

$$\frac{u^{k+1} - u^k}{\tau} = A(u^k)u^k \quad (3.17)$$

$$\Rightarrow u^{k+1} = \left(I + \tau A(u^k) \right) u^k \quad (3.18)$$

with $A(u^k) = \sum_{l=1}^2 A_l(u^k)$, where $A_l(u^k) = \left(a_{ij}^l(u^k) \right)$ and

$$a_{ij}^l(u^k) = \begin{cases} \frac{g_i^k + g_j^k}{2h^2}, & j \in \mathcal{N}_l(i) \\ -\sum_{n \in \mathcal{N}_l(i)} \frac{g_i^k + g_n^k}{2h^2}, & j = i \\ 0 & \text{else.} \end{cases} \quad (3.19)$$

Here, $\mathcal{N}_l(i)$ describes the set of the two neighbors of pixel i in direction $\pm e_l$ (boundary pixels only have one neighbor). In image processing, the spatial step size usually fulfils $h = 1$ [83].

Theorem 2 demonstrates that the useful application of the explicit scheme (3.18) is restricted to small time steps τ . This motivates the investigation of different iteration schemes.

By approximating the spatial derivations of u - represented by the right side of Equation (3.17) - at time t_{k+1} , the *implicit* iteration scheme (3.21) is deduced:

$$\frac{u^{k+1} - u^k}{\tau} = A(u^{k+1})u^{k+1} \quad (3.20)$$

$$\Rightarrow \left(I - \tau A(u^{k+1}) \right) u^{k+1} = u^k \quad (3.21)$$

Due to the nontrivial structure of the matrix A , which depends nonlinearly on u , solving the system of equations (3.21) is a complicated process [83]. Theorem 2 shows that the *semi-implicit* discretization scheme (3.23), where A is evaluated in an explicit way and the linear remainder in an implicit manner, is applicable without limitation. It requires to solve a system of linear equations in order to obtain the solution u^{k+1} :

$$\frac{u^{k+1} - u^k}{\tau} = A(u^k)u^{k+1} \quad (3.22)$$

$$\Rightarrow \left(I - \tau A(u^k) \right) u^{k+1} = u^k \quad (3.23)$$

In the following, both the explicit and the semi-implicit scheme are examined.

Theorem 2. *The explicit scheme (3.18) of the CLMC equation creates a discrete scale space subject to the time step size restriction $\tau < \frac{1}{4}$ while the semi-implicit scheme (3.23) permits arbitrarily large time steps.*

Proof. According to Theorem 1, the properties (D1)-(D6) need to be verified. The explicit scheme is given by $Q_1(u^k) := (q_{ij}^1(u^k)) := I + \tau A(u^k)$, implying $q_{ij}^1(u^k) = \delta_{ij} + \tau \sum_{l=1}^2 a_{ij}^l$ where δ_{ij} denotes the Kronecker delta and a_{ij}^l is defined in (3.19). The continuity condition (D1) follows directly from the continuity of the diffusivity g , cf. (3.9). Due to the symmetry of δ_{ij} and a_{ij}^l (due to $j \in \mathcal{N}_l(i) \Leftrightarrow i \in \mathcal{N}_l(j)$), Q_1 is symmetric and therefore fulfils (D2). From (3.19) it is also clear that $\sum_{j=1}^N a_{ij}^l = 0$ for all $i \in J$ and $l = 1, 2$, which proves (D3). The nonnegativity of q_{ij}^1 for $i \neq j$ results from the positivity of g_j^k for all $j \in J$ and $k \in \mathbb{N}_0$. In order to fulfil (D4) and (D6), all diagonal elements q_{ii}^1 must be positive. As a consequence of $g \leq 1$, $h = 1$ and $|\mathcal{N}_l| \leq 2$, the inequality

$$q_{ii}^1 = 1 - \tau \sum_{l=1}^2 \sum_{n \in \mathcal{N}_l(i)} \frac{g_i + g_n}{2h^2} > 0 \quad \forall i \in J$$

is satisfied in particular, if

$$\tau < \frac{1}{4}. \quad (3.24)$$

Finally, from Figure 3.1 it is obvious that the index transformation (3.13) ensures that $\{i-1, i+1\} \subset \cup_{l=1}^2 \mathcal{N}_l(i)$. Therefore, the tridiagonal elements of Q_1 are non-zero:

$$\begin{aligned} q_{i,i-1}^1 &> 0 \quad \forall i = 2, \dots, N \\ q_{i,i+1}^1 &> 0 \quad \forall i = 1, \dots, N-1. \end{aligned} \quad (3.25)$$

Now let $s_1, s_2 \in J$. If $s_1 \leq s_2$, (3.25) guarantees

$$q_{s_1, s_1+1}^1 > 0, \quad q_{s_1+1, s_1+2}^1 > 0, \quad \dots, \quad q_{s_2-1, s_2}^1 > 0.$$

For $s_1 > s_2$, we have

$$q_{s_1, s_1-1}^1 > 0, \quad q_{s_1-1, s_1-2}^1 > 0, \quad \dots, \quad q_{s_2+1, s_2}^1 > 0.$$

Hence, Q_1 is irreducible and (D5) is proven.

The implicit scheme depends on the inversion of

$$B(u^k) := (b_{ij}(u^k)) := I - \tau A(u^k).$$

Due to $|b_{ii}| = 1 + \sum_{j \neq i} |b_{ij}|$ for all $i \in J$, B is strictly diagonally dominant, i.e. $|b_{ii}| > \sum_{j \neq i} |b_{ij}|$ for all $i \in J$. Then, results from linear algebra state that $Q_2 := B^{-1}$ exists and fulfils (D1)-(D6) for all time step sizes $\tau > 0$ (see [83] for details). ■

Despite its unconditional stability, the major drawback of the semi-implicit scheme is its high computational cost accruing from the solution of the linear system $B(u^k)u^{k+1} = u^k$. For this purpose, a modification of (3.23) is introduced:

$$u^{k+1} = \frac{1}{2} \sum_{l=1}^2 \left(I - 2\tau A_l(u^k) \right)^{-1} u^k. \quad (3.26)$$

The system (3.26) is called *Additive Operator Splitting (AOS) Scheme* since the diffusion term is splitted into one-dimensional processes along the coordinate axes. Subject to an adequate pixel numbering, the matrices

$$B_l(u^k) := I - 2\tau A_l(u^k)$$

possess tridiagonal form and the resulting linear systems can be solved in linear complexity with the aid of the so-called *Thomas algorithm* [83]. Furthermore, the following results can be proven [80, 83]:

- The scheme (3.26) satisfies the conditions (D1)-(D6) and therefore creates a discrete scale space.
- The Taylor expansions of all three presented schemes (3.18), (3.23) and (3.26) have the same approximation order $\mathcal{O}(\tau + h^2)$ and therefore are consistent to the original equation.

Table 3.1 outlines the results obtained in this section. Section 3.1.3 demonstrates the impact of the discretized CLMC equation on raw μ PTV data.

Table 3.1: Finite difference schemes creating a discrete nonlinear diffusion scale space. Table content taken from [79].

Scheme	Stability	Costs per iteration
Explicit	$\tau < \frac{1}{4}$	very low
Semi-implicit	$\tau < \infty$	high
AOS	$\tau < \infty$	low

3.1.2.4 Anisotropic Diffusion Filtering

The approaches illustrated up to this point used a scalar diffusivity $D = g(\|\nabla u\|^2)$, implicating that the flux $j = -g\nabla u$ is always parallel to ∇u . Such models are known as *isotropic* diffusion filters. *“Nevertheless, in certain applications it would be desirable to bias the flux towards the orientation of interesting features. These requirements cannot be satisfied by a scalar diffusivity anymore, a diffusion tensor leading to anisotropic diffusion filters has to be introduced”* [80]. For this purpose, different suggestions for the construction of the tensor D have been released, as for instance described in [18,80,81]. Weickert [80] proves the existence of a continuous as well as a discrete scale space (in the sense that was introduced above) for a broad class of diffusion tensors, thus extending the results from the isotropic CLMC equation (Theorem 1 and 2).

It should be noted that the notation in literature is not consistent concerning the terms “(an)isotropic”. Some authors refer to homogeneous diffusion processes (constant diffusivity) as isotropic while calling any inhomogeneous filter anisotropic, even if they use a scalar-valued diffusivity (e.g. [16,58]).

3.1.3 Proposed modifications and impact on Micro PTV data

Let us now illustrate the behaviour of the discrete CLMC diffusion filters on μ PTV images, using the diffusivity g from (3.9). For the implementation of the algorithm, the *Nonlinear Diffusion Toolbox* by Frederico D’Almeida [19] from the MATLAB Central File Exchange server has been thoroughly examined and adopted. For the calculations, MATLAB R2010a on a 64-bit Windows 7 operating system with an Intel Core 2 Duo T9400 processor (2.53 GHz) and 4 GB RAM was used.

Initially, the influence of the contrast parameter λ (appearing in the representation (3.9) of the diffusivity) and the regularization parameter σ (appearing in the CLMC equation (3.16) with (3.10)) shall be visualized, regarding both the explicit discretization (3.18) and the AOS scheme (3.26). For this, a 140×140 window containing four distinct particles has been selected as the reference image and the stopping time is chosen as $T = 4$. For the sake of visibility, the contrast is increased by means of a linear pixel brightness transformation to the gray value range $[0, 255]$. The explicit scheme is applied with time step size $\tau = 0.1$ ($k_{\max} = 40$ iterations) while the solution of the AOS scheme is computed in $k_{\max} = 2$ steps with $\tau = 2$.

Subjective visual inspection of the resulting images suggests that $\lambda \in [5, 10]$ and $\sigma \in [1, 5]$ represent reasonable choices. In Figure 3.2, it can be seen that for a lower parameter selection,

image noise remains and a higher selection leads to a slightly oversimplified outcome (the holes in the ring-shaped particles are diminishing).

Due to the scale space properties of the considered diffusion filter class, the major influence on the morphological image structure certainly is constituted by the stopping time $T = k_{\max}\tau$. Figure 3.3 compares the temporal evolution of the AOS scheme to a stable ($\tau < 1/4$) and an unstable ($\tau > 1/4$) version of the explicit scheme. Furthermore, the universal blurring effect of a linear diffusion filter (Gaussian filtering) is displayed.

Finally, the accuracy as well as the temporal performance shall be evaluated more quantitatively. “*Since no analytical solution to the CLMC equation is known, [...] a good numerical approximation to a test example as a standard for comparison*” has to be used, as Weickert [83] points out. Analogous to [83], the explicit scheme with a small step size (in our case: $\tau = 0.1$) provides the filtered reference image v . The relative ℓ^2 -error of an approximation u is calculated as

$$\text{error}_C(u) = \frac{\|u - v\|}{\|v\|}, \quad (3.27)$$

where $\|v\| = (\sum_i v_i^2)^{\frac{1}{2}}$ again denotes the ℓ^2 -norm (of the vectorized image v). The subscript C indicates that this error is derived from the continuous image representations of u and v . Originally, the images are given in 8-bit format, i.e. the gray values are contained in the set $\{0, 1, \dots, 255\}$. The images resulting from the diffusion process, however, are real-valued and in the range $[0, 255]$ due to the extremum principle. Naturally, by rounding those values to the closest number in \mathbb{N}_0 , the data format of the filtered image is adjusted. In the following, this (component-wise) rounding operation is denoted by square brackets $[\cdot]$. The ℓ^2 -error of the discretized image is therefore computed as

$$\text{error}_D(u) = \frac{\|[u] - v\|}{\|v\|}. \quad (3.28)$$

Table 3.2 lists the processing times and ℓ^2 -errors of AOS and explicit scheme using the stopping time $T = 200$ and various time steps τ (or equivalent: a various number of iterations $k_{\max} = \frac{T}{\tau}$). Two different images are evaluated: Image 1 refers to the 140×140 pixel image from Figure 3.2 and 3.3 while Image 2 represents the whole 2048×2048 recording from which Image 1 is cut out. The lower error rates for Image 2 result from the existence of large image regions without particles, where no significant information needs to be processed. By comparison, Image 1 possesses a relatively high information density. Furthermore, it is demonstrated that the average processing time per iteration of a regular 2048×2048 μ PTV image amounts to approximately 2.9 seconds for the explicit scheme and 6.4 seconds for the AOS scheme. Considering the explicit scheme with time step τ_e and the AOS scheme with number of iterations k_a , we conclude that (for the configuration in use) the processing time of the latter falls below that of the former for all stopping times

$$T \geq \lceil 2.2k_a \rceil \tau_e,$$

where $\lceil c \rceil$ denotes the lowest natural number greater than c .

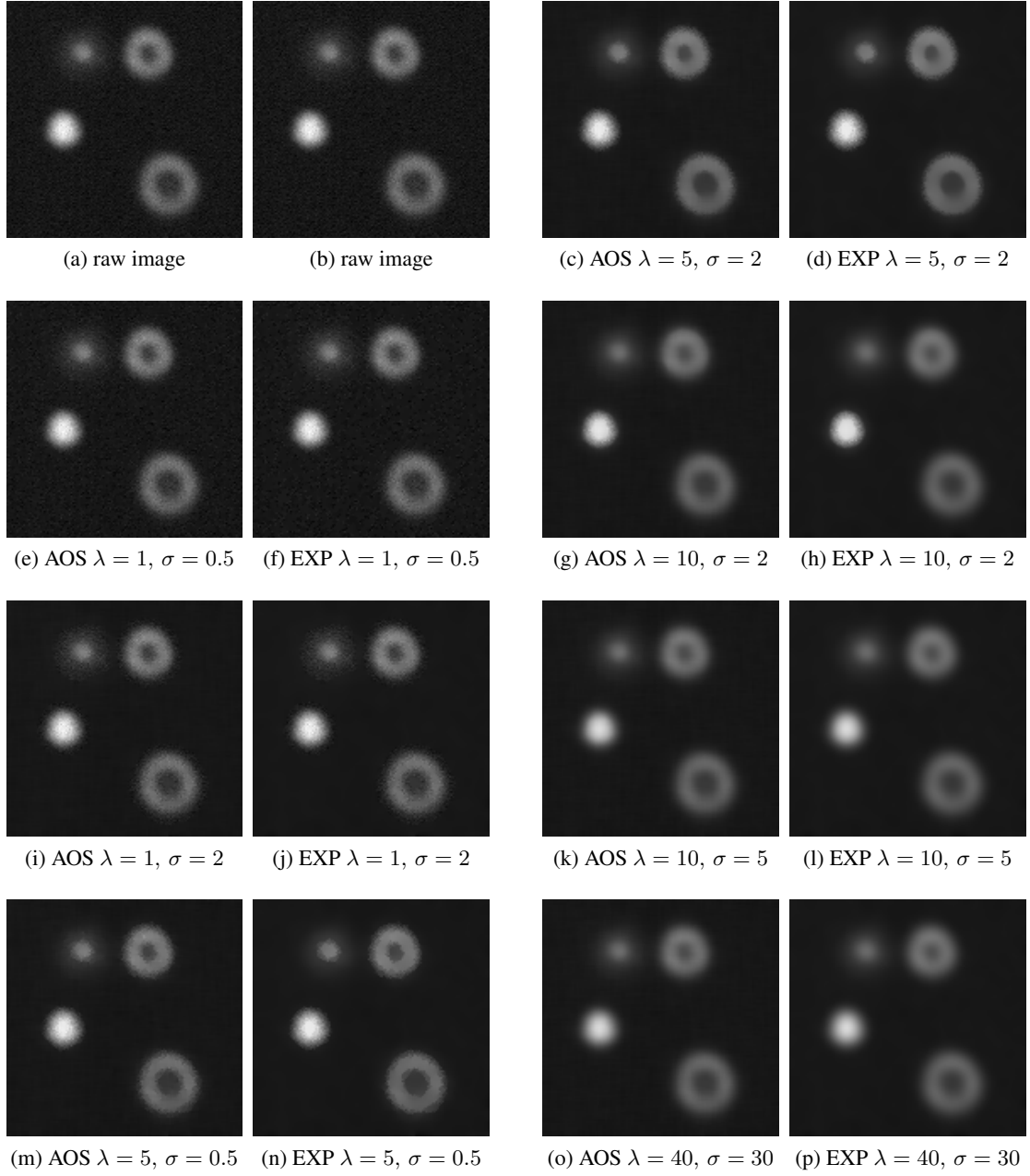


Figure 3.2: Evolution of a cut out μ PTV image under the discretized CLMC equation (3.11), with varying parameters λ [from diffusivity (3.9)] and σ . The explicit scheme EXP (3.18) was evaluated in $k_{\max} = 40$ time steps with $\tau = 0.1$, the AOS scheme (3.26) in $k_{\max} = 2$ steps with $\tau = 2$ respectively.

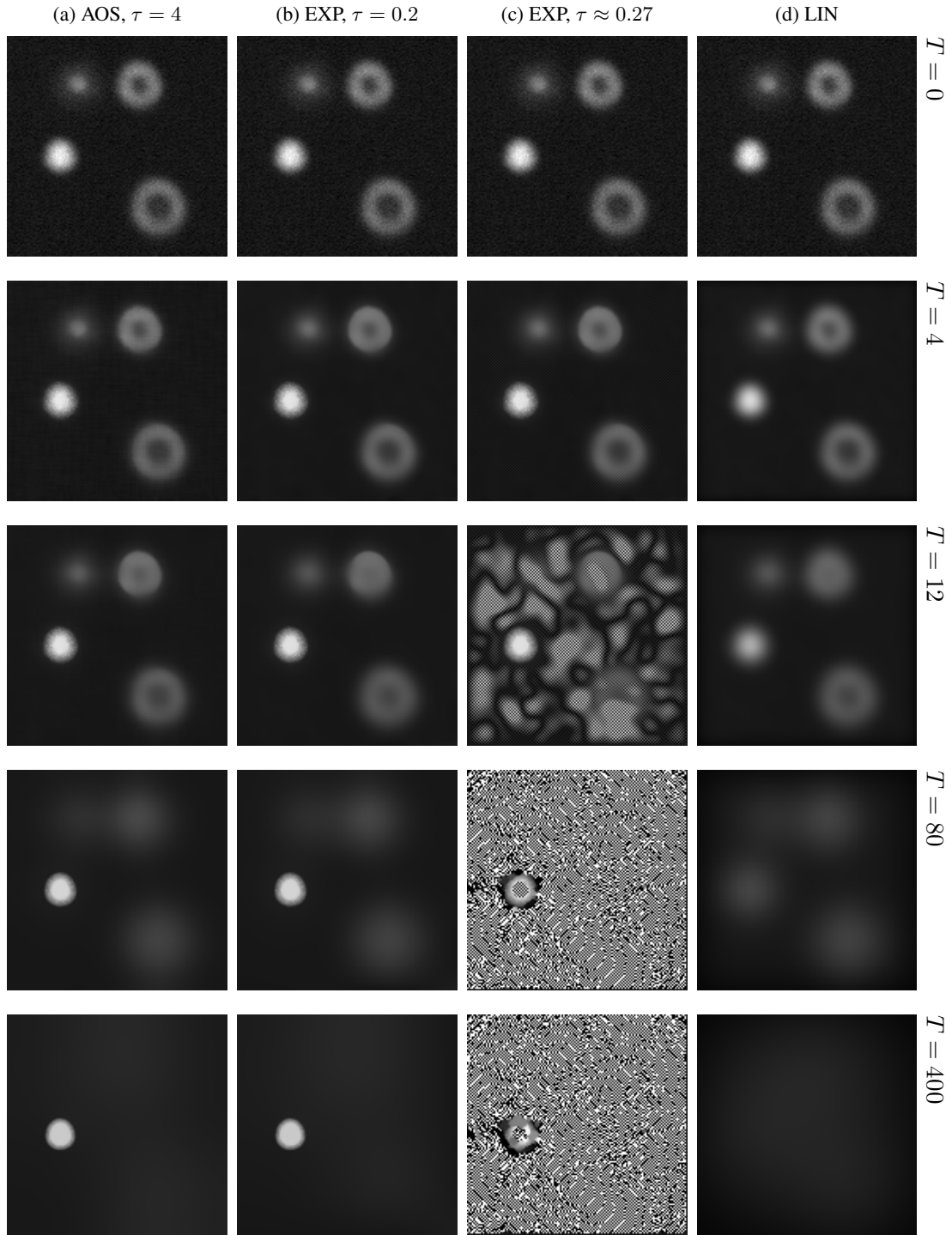


Figure 3.3: Evolution of a cut out μ PTV image under diffusion filtering with varying stopping time T . Parameter selection: $\lambda = 8$, $\sigma = 2$. Column (a): CLMC with AOS scheme. Column (b): CLMC with stable, explicit scheme. Column (c): CLMC with unstable, explicit scheme. Column (d): Linear Diffusion approach (Gaussian smoothing).

Table 3.2: Comparison of nonlinear diffusion schemes with respect to processing times and accuracy. Parameter selection: $\lambda = 8$, $\sigma = 2$, $T = 200$.

Scheme	τ	k_{\max}	Image 1			Image 2		
			CPU time	error _C	error _D	CPU time	error _C	error _D
explicit	0.1	2000	15.08 s	0 %	0.76 %	5836 s	0 %	1.43 %
	0.2	1000	7.56 s	0.003 %	0.76 %	2922 s	0.002 %	1.43 %
	0.25	800	6.06 s	0.23 %	0.80 %	2339 s	0.41 %	1.49 %
AOS	0.25	800	6.70 s	5.65 %	5.69 %	5109 s	1.08 %	1.79 %
	1	200	1.64 s	5.64 %	5.69 %	1275 s	1.08 %	1.79 %
	5	40	0.35 s	5.65 %	5.70 %	255.3 s	1.09 %	1.79 %
	20	10	0.10 s	6.22 %	6.26 %	63.84 s	1.32 %	1.94 %
	50	4	0.05 s	8.89 %	8.92 %	25.53 s	2.30 %	2.70 %
	100	2	0.03 s	13.91 %	13.92 %	12.89 s	3.77 %	4.04 %
	200	1	0.02 s	21.66 %	21.68 %	6.39 s	5.99 %	6.16 %

To sum up, with regard to the accuracy, the explicit scheme is superior to the AOS scheme. Especially for the processing of hundreds of large images, however, the AOS scheme represents a powerful tool due to the enormous temporal performance advantage combined with a merely moderate loss of accuracy: Table 3.2 shows that in the given framework, the AOS scheme with $k_{\max} = 10$ iterations is over 45 times faster than the explicit scheme with step size $\tau = 0.2$. Nevertheless, the corresponding discretized ℓ^2 -errors are in a close range (1.43 % and 1.94 %, respectively) such that the resulting images are indistinguishable with the naked eye, cf. the related situation in Figure 3.3.

For the practical use on μ PTV images, we propose a stopping time of $T \approx 4$. In this case, the AOS scheme with $k_{\max} = 5$ is about twice as fast as the explicit scheme with $\tau = 0.2$ while the image quality of the latter is slightly superior (as indicated in Table 3.2). Both methods represent justified pre-processing techniques and ultimately, the method of choice depends on the priorities of the user.

3.2 Step 2: Particle Detection

The upcoming section describes the algorithms implemented for the purpose of automatic particle detection. The procedure is composed of two stages: initially, the edge locations of the gray valued raw image are computed by means of the well-known Canny algorithm [13]. Subsequently, the particle positions, indicated by the presence of circular shapes, are extracted from this binary edge image with the aid of a Circular Hough Transform.

The principles of the Canny algorithm are explained in Section 3.2.1, followed by an overview of different types of Hough Transforms in Section 3.2.2. Finally, Section 3.2.3 outlines the modifications applied to both techniques due to the considered framework and depicts the effect of various parameter selections.

3.2.1 Edge Detection - The Canny Algorithm

Despite the fact that its introduction dates back to the year 1986, the Canny Algorithm still represents an established edge detection technique [72]: “*Although research into reliable edge-detection algorithms continues, the Canny method is generally acknowledged as the best ‘all-round’ edge detection method developed to date*”. Canny postulates three performance criteria for the edge detection problem [13, 73]:

(EDP1) Detection criterion: important edges should not be missed, non-edge pixels should not be falsely marked.

(EDP2) Localization criterion: the distance between actual and detected edge position should be minimal.

(EDP3) One response criterion: there must be only one response to a single edge.

Actually, (EDP3) is implicitly covered by (EDP1). However, it is required since Canny’s mathematical expression for (EDP1) does not address the multiple response characteristic. The edge detector is formulated as a convolution filter determined to locate step edges disturbed by white Gaussian noise. The task is to identify the one filter that optimizes (within the scope of calculus of variation) the three criteria (EDP1)-(EDP3). For a detailed description of the optimization problem, we refer to the original paper [13]. While the result proved to be too complex to be solved analytically, it is shown that the convolution of the image f with a two-dimensional Gaussian K_σ (cf. (3.6)), followed by the computation of the normal derivative,

$$\frac{\partial}{\partial n} (K_\sigma * f), \quad (3.29)$$

constitutes an efficient approximation [13] to the optimal edge detector. Note that the differentiation and convolution operator commute, which speeds up computation in practice since the normal derivative of a Gaussian is known in advance. The Gaussian parameter σ is associated with the scale at which the edges are computed (cf. the Gaussian scale-space in Section 3.1.1). The edge normal n of the image f is estimated as

$$n = \frac{\nabla(K_\sigma * f)}{\|\nabla(K_\sigma * f)\|}. \quad (3.30)$$

An edge location is then situated at a local maximum of (3.29) in direction n , yielding the necessary condition

$$\frac{\partial^2}{\partial n^2} (K_\sigma * f) = 0, \quad (3.31)$$

and the strength of an edge is given by the magnitude η :

$$\eta := \left| \frac{\partial}{\partial n} (K_\sigma * f) \right| = \|\nabla (K_\sigma * f)\|. \quad (3.32)$$

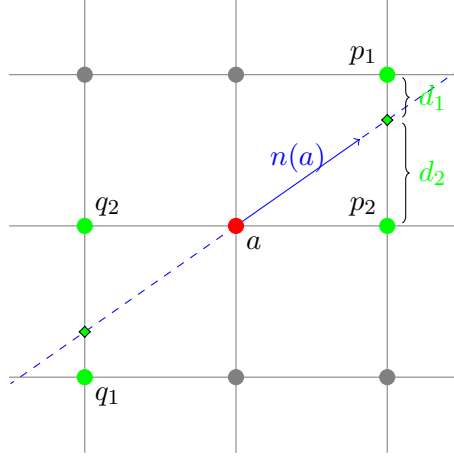


Figure 3.4: Non-maximum suppression technique. The magnitudes along the edge normal are interpolated from the adjacent pixels.

In the (discrete) practical case, Equation (3.31) is realized as the condition that the strength of an edge pixel dominates the edge magnitude of its neighbors along the gradient direction. Since in general, the orientation of the gradient vector is not a multiple of $\pi/4$ (i.e. pointing exactly at a neighbor pixel), it is common to use linear interpolation [55]. As illustrated in Figure 3.4, we denote the four neighbors of a pixel a which are closest to the line spanned by the normal vector $n(a)$ as p_1 , p_2 (positive direction) and q_1 , q_2 (negative direction) respectively; their respective distances on the grid are labeled d_1 , d_2 . Assuming unit pixel distance, the two magnitude conditions for the edge candidate a are then written as

$$\begin{aligned}\eta(a) &> (1 - d_1)\eta(p_1) + (1 - d_2)\eta(p_2), \\ \eta(a) &> (1 - d_1)\eta(q_1) + (1 - d_2)\eta(q_2).\end{aligned}$$

As a consequence, it is ensured that the thickness of each detected edge does not exceed one pixel. In literature, this operation is referred to as *non-maximal suppression* [73].

Eventually, the number of edge candidates is further reduced by thresholding the edge magnitude (3.32). In order to avoid streaking, i.e. the breaking up of edge contours due to magnitude fluctuations, so-called *hysteresis thresholding* is deployed. Thereby, two magnitude thresholds t_0 , $t_1 > 0$ are applied: all edge responses, where the magnitude exceeds the higher threshold t_1 , constitute the definite, strong image edges while weaker responses with a magnitude between t_0 and t_1 are solely considered as edges, if they are connected to a strong edge. Candidates with magnitudes smaller than t_0 are discarded.

A complete C code of the Canny algorithm is provided in [55]. Details and results for an application on μ PTV data are illustrated in Section 3.2.3.

3.2.2 Hough Transforms

Originally designed for the identification of lines in binary images in 1962, a diversity of modifications have made Hough Transforms (HT) a well-established tool for the detection of specific

shapes in images [71]. In order to detect a binary object that is described by an equation with n appearing parameters, it is necessary to construct an n -dimensional so-called accumulator space. This is done by starting a voting process where each image edge point casts votes for that set in the parameter space which corresponds to all possible objects that contain this point. By accumulating all votes for each n -parameter-tupel, the accumulator space is built. The parameters of potential objects eventually can be extracted as the positions of local maxima of this accumulator space.

In their survey [29], Illingworth and Kittler discuss the HT concept in general and provide details for a successful implementation. As cited by [73], they conclude that Hough Transforms fulfil several beneficial properties:

- The HT is very robust in the presence of additional data such as other objects or noise.
- Partial occlusion or a slight deformation does not impede the recognition of an object.
- Multiple occurrences of a shape can be detected in one processing step.
- Since each image point is treated separately, the implementation of a parallel processing algorithm is feasible.

In the first two subsections, two well-known representatives of the HT are presented: the algorithm for the detection of straight lines (henceforth denoted as the Standard Hough Transform (SHT)) as well as the Circular Hough Transform (CHT). Subsequently, an introduction to the Generalized Hough Transform (GHT), which allows the recognition of arbitrary, non-parametric shapes, is given.

3.2.2.1 Standard Hough Transform

In the original work by Hough [27], lines are characterized by the slope-intercept form

$$y = kx + d. \quad (3.33)$$

A reorganization of (3.33) instantly yields that the voting set for a fixed, arbitrary edge point (x, y) again is described by a line in the (k, d) plane

$$d = -xk + y. \quad (3.34)$$

Due to the fact that vertical lines are not covered by the representation (3.33), the use of the normal form of a line is preferred [22]:

$$\rho = x \cos \theta + y \sin \theta, \quad (3.35)$$

where $\theta \in [-\pi, \pi]$ is the angle of its normal and $\rho \geq 0$ is its perpendicular distance to the origin, see Figure 3.5(a). Figure 3.5(b) and (c) show the votes of three points p, q, r (collinear on line l) by means of the transformations (3.34) and (3.35) respectively. The parameters of l correspond to the point of intersection of their voting sets.

A complete C code of the SHT again is provided in [55].

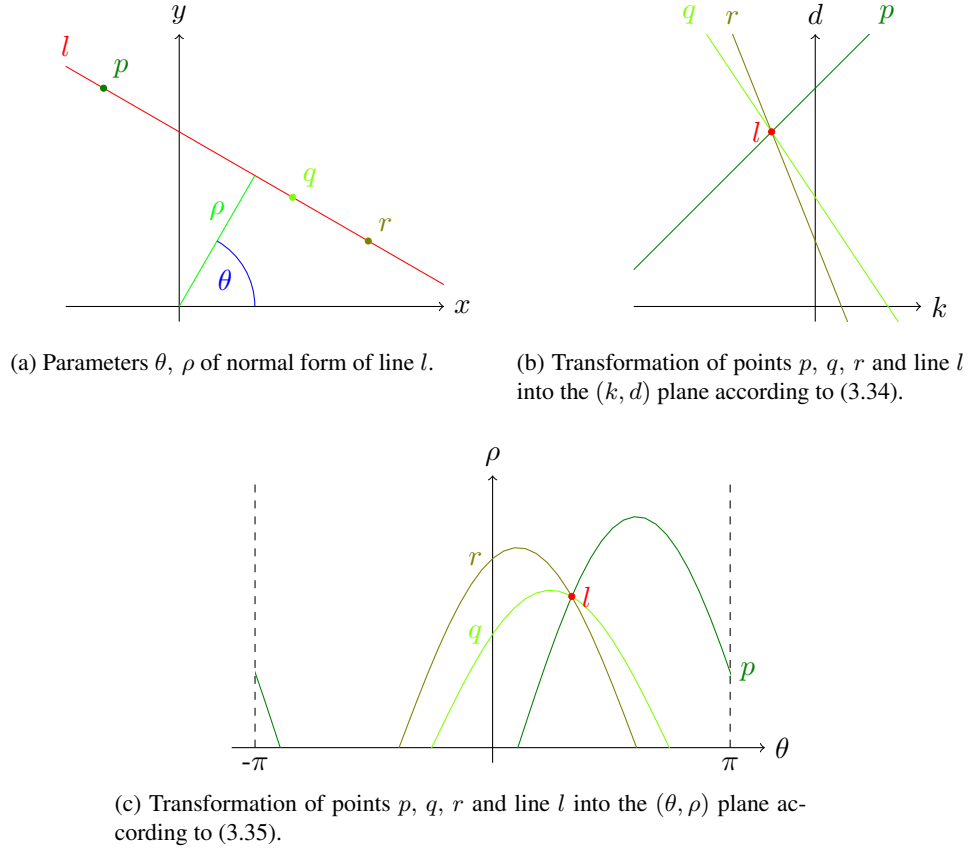


Figure 3.5: Standard Hough Transform voting schemes.

3.2.2.2 Circular Hough Transform

The CHT is intended to find circular objects in images, characterized by their center coordinates and their (a priori unknown) radii, thus establishing a 3-dimensional accumulator space. We denote the circle with radius r and center coordinates (a, b) as

$$B_r(a, b) := \{(c, d) \in \mathbb{R}^2 : (c - a)^2 + (d - b)^2 = r^2\}.$$

For an edge pixel (x, y) and a fixed search radius r , the set of center candidates $V^r(x, y)$ where the corresponding circle with radius r contains (x, y) on the border is constituted by the circle with center (x, y) and radius r :

$$\begin{aligned} V^r(x, y) &= \{(a, b) \in \mathbb{R}^2 : (x, y) \in B_r(a, b)\} \\ &= \{(a, b) \in \mathbb{R}^2 : (x - a)^2 + (y - b)^2 = r^2\} \\ &= \{(a, b) \in \mathbb{R}^2 : (a - x)^2 + (b - y)^2 = r^2\} \\ &= B_r(x, y). \end{aligned}$$

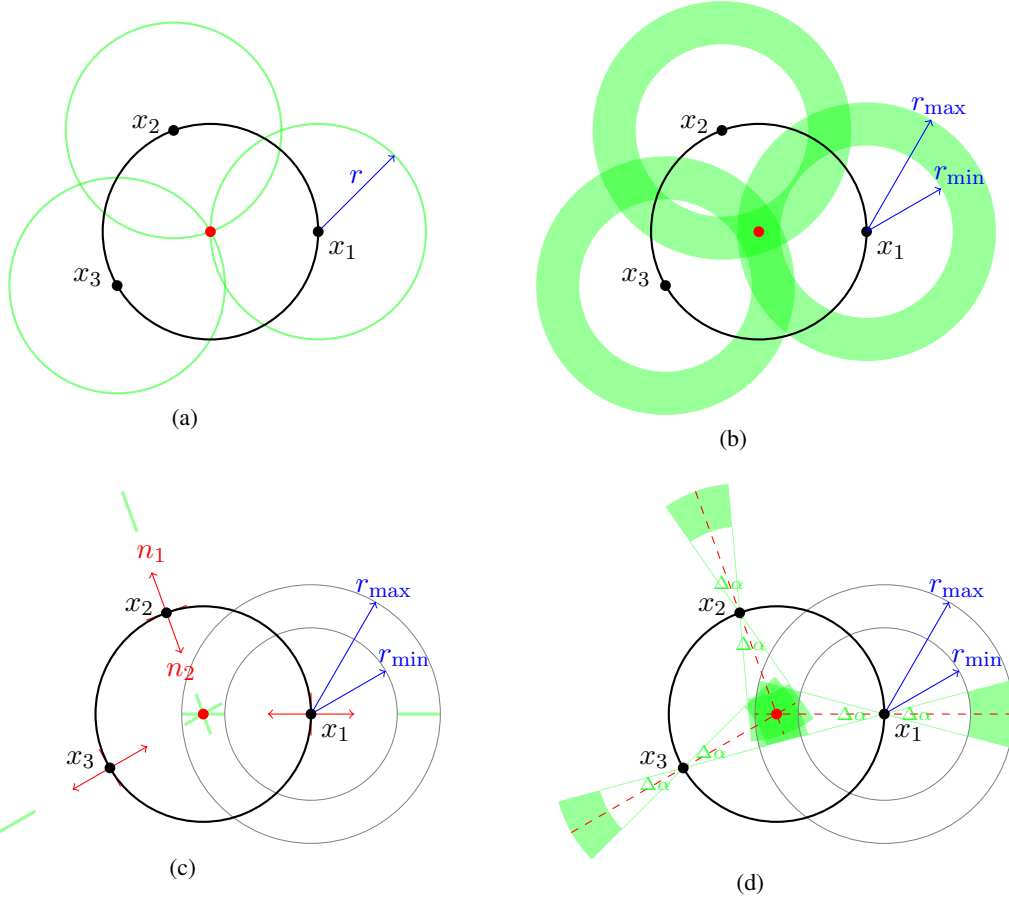


Figure 3.6: Voting schemes: (a) standard CHT, (b) CHT with multiple radii in a 2D parameter space, (c) gradient oriented CHT with multiple radii in a 2D parameter space, (d) gradient oriented CHT with multiple radii and orientation uncertainty $\Delta\alpha$. Figure inspired by [71].

Figure 3.6(a) indicates the voting scheme of three edge points on a circular edge with known radius. Passing through all possible radii in a search range $[r_{\min}, r_{\max}]$, the three-dimensional voting set $V(x, y)$ for the edge point (x, y) is given by a frustum:

$$V(x, y) = \{(x_1, y_1, r) \in \mathbb{R}^3 : r \in [r_{\min}, r_{\max}], (x_1 - x)^2 + (y_1 - y)^2 = r^2\}.$$

In order to reduce computational and memory requirements and improve the detection performance, different modifications to the standard CHT have been introduced, as discussed for instance in [8, 29, 71, 73]. Firstly, it is sufficient to consider a flat, two-dimensional accumulator space such that an edge point increments a comprising annulus of points in an accumulator plane. The minimum/maximum radius of the voting annulus corresponds to the minimum/maximum radius of the circle searched, as seen in Figure 3.6(b). The local maxima in this accumulator array are strong candidates for circle centers. The circle radii have to be determined in an ad-

ditional step, for instance with the aid of the so-called radial histogram technique. Thereby, each potential radius r is represented by a histogram bin and its value is given by the number of edge pixels along the circle with radius r surrounding the detected center. The peak location eventually yields the radius of the circle.

In addition, the direction of the edge normals can be used to constrain the number of votes per pixel due to the fact that for each point on a perfect circle, the vector pointing towards the center is perpendicular to the edge direction. According to each search radius, solely those elements in the accumulator array which are positioned in direction of one of the two edge normals are incremented (see Figure 3.6(c)). In order to compensate for shape irregularities, however, it is beneficial to vote for an arc with uncertainty angle $\Delta\alpha$ instead of only one point per normal direction [71, 73]. According to these modifications, the voting area of an edge pixel is composed of two annular sectors as shown in Figure 3.6(d) which provides a superior performance for the CHT [71].

If edge magnitude information is available, “another heuristic that has a beneficial influence on the curve search is to weight the contributions to accumulator cells [...] by the edge magnitude” [73]. Alternatively, other modifications have been proposed, such as adaptive algorithms [15, 28], randomized approaches [17, 87], the application of a scale invariant kernel operator [8] or the implementation of a complex accumulator array, where the radius is encoded in the phase of the vote [7]. An overview of existing parametric HT variations as well as a comparison with respect to speed and memory usage is given in [34]. In general, a problem-based adjustment of the voting strategy can significantly improve the obtained results, cf. Section 3.2.3.

3.2.2.3 Generalized Hough Transform

There are applications where a parametric representation of the structure searched for in an image is not available. For this purpose, Ballard [11] introduced the GHT - an extension to the HT algorithm which enables the detection of patterns similar to a reference object by exploiting edge direction information. As a first step, a reference point x^R inside the region is arbitrarily selected (usually the centroid [35]). For each boundary point x of the reference object, its edge orientation $\phi(x)$ is computed and its distance r and orientation α towards x^R are stored in a reference table (a so-called *R-table*) as a function of ϕ . An edge pixel $x_e = (e_1, e_2)$ in the considered image, whose edge orientation $\phi(x_e)$ corresponds to the R-table value ϕ_j , then votes for the set $V(x_e)$ of all potential reference points, for which x_e is located on the respective object:

$$V(x_e) = \bigcup_{l=1}^{n_l} \left\{ \left(e_1 + r_j^l \cos(\alpha_j^l), e_2 + r_j^l \sin(\alpha_j^l) \right) \right\}.$$

Accordingly, an accumulator space A is established in a similar manner to the previously described Hough Transforms and the positions of the reference points of the objects searched are indicated by local maxima of A . Figure 3.7 illustrates the geometry of the R-table construction and displays the general form of an R-table.

If the scale S and the degree of rotation τ of the object are not known in advance, the accumulator space becomes 4-dimensional. This leads to a memory requirement of $N^2 S_q \tau_q$ where

the edges detected by the Canny algorithm superimposed on the original image. In the center column, the resulting accumulator arrays as well as the highlighted positions of its identified local maxima are displayed. The third column of images eventually shows the extracted particle locations and their associated radii. A characterization of the specific coloring used for the edge components and the local maximum tags in the plots is provided below, at the passage of the respective algorithm description.

Determination of the parameters for the Canny algorithm

Perona and Malik [58] state that edge detection subsequent to nonlinear diffusion filtering is superior to linear Gaussian filtering used in the classical Canny approach. Hence, the Gaussian smoothing operation is not necessarily required. However, the scale of the detected edges can be further adjusted by the selection of a “moderate” $\sigma \in [1, 3]$ (depending on the image quality) for the Gaussian K_σ . Apart from that, the AOS scheme produces slightly perceptible horizontal and/or vertical strips for a small and practically relevant number of time steps (e.g. $k_{\max} \in \{2, 3, 4\}$), which can be observed in a close examination of Figure 3.2 and 3.3 (note that due to the small image size, this behaviour is hardly visible in a printed version of this document). The reason for this is the numerical realization, as the AOS scheme induces a decomposition of the diffusion process into two one-dimensional processes along the coordinate axes [80]. From a heuristic point of view, this causes the elimination of the stripes while the strong image edges, which “survived” the application of the nonlinear diffusion filter, are largely preserved.

The specification of the hysteresis thresholds t_0 and t_1 for a whole raw image series of over 100 image pairs is a non-trivial task. In this approach, they are chosen as constants based on manually selected values for the reference matrix images. The occurring contrast inconsistencies between distinct raw images imply the requirement of relative thresholds (with respect to the maximum appearing absolute gradient value). In this case, constant thresholds are only legitimate for those images where the maximum gradient value descends from a particle in the same distance to the camera focus as in the reference image. In other words, this scenario requires that the layer of the brightest particle from each image is consistent with the layer of the brightest particle from the reference image. For an ordinary μ PTV dataset (70-100 particles per image), however, this condition has proven to be satisfied in general.

Alternatively, the thresholds could be selected separately for each image pair, e.g. via a specified, fixed value for the amount of strong edge candidates. Due to the diversity of particle shapes, the border length of individual particles is widely different: the computed thresholds for an image mainly filled with ring particles (long border length) and an image filled with the same number of sharp particles (short border length) respectively would vary significantly, even under similar contrast conditions. Hence, the former mentioned constant threshold approach has been adopted.

The success of the whole detection algorithm is influenced by the parameter selection: on the one hand, too low thresholds result in a large number of useless edges which impede the performance of the CHT and give rise to the potential detection of false positives. On the other hand, too high thresholds eliminate the edges of diffuse low-contrast particles, leading to a large category of false negatives. It is not the purpose of the μ PTV software, however, to omit the recognition of whole particle classes. Therefore, the path taken is to choose the thresholds in the

lower range which permits the detection of abundant edges. In return, prior to the construction of the accumulator array, the technique described below removes spurious edges in order to minimize perturbing votes.

Elimination of disturbing edges and the CHT voting mechanism

In the present μ PTV images, particles are characterized as bright spots (high gray value) on a dark background (low gray value) such that the gradient vectors on the outer edge of a particle exclusively point towards the particle center. Therefore, the vectors which determine the edge normals in the Canny algorithm (3.30) represent a valid estimation of the inner edge normals of a circle. Thus, in contrast to the scheme depicted in Figure 3.6(d), the accumulator array votes are cast solely into one annular sector instead of two, cf. Figure 3.8.

As a further improvement for the performance of the CHT algorithm on μ PTV data, spurious edges are removed from the edge image in order to minimize perturbing votes prior to the construction of the accumulator array. On that note, a connected edge component is declared as a spurious edge if it does not contribute to (but rather disturb) the detection of one true particle center, i.e. at least one of the following conditions is fulfilled:

- (E1) The component is solely a short edge fragment, e.g. consists of 8 pixels at most.
- (E2) The gradient vectors of its elements, which determine the voting areas, do not intersect frequently enough and therefore do not allow constructive accumulation. In order to keep computational requirements at a minimum, at most 10 edge points are randomly selected from the component. The rays originating from the corresponding gradient vectors are subject to pairwise comparisons and if less than a certain percentage (e.g. 30%) of them intersect, the component is rejected. Especially interfering votes from the inner circle of ring-shaped particles, where the gradient vectors point towards the outside of the circle, are prevented by this procedure.

The first column of Figure 3.11 reveals the detected edges with varying Canny thresholds in the template image. Those edge components permitted to cast votes in the CHT are displayed in green while the spurious, dismissed components are colored in red. The assertion from above that low thresholds are superior is confirmed since the elimination algorithm effectively removes the majority of redundant edges (second row). In contrast, high Canny thresholds are responsible for the failed recognition of the diffuse particle due to the deficient detection of relevant edges (first row). In addition, selected gradient vectors occurring in condition (E2) are drawn in Figure 3.11(c).

Figure 3.12(a) indicates the consequences of the omission of the edge elimination process: whereas the overall particle detection quality is hardly affected (exclusively all four true particles are identified in the present case), the temporal performance of the algorithm is deteriorated due to the large number of non-zero elements in the accumulator array.

Additionally, specific voting weights are introduced. In order to compensate for the higher number of votes originating from large circles, the weighting factor $\frac{1}{r}$ is used. Furthermore, a dependency of the voting weights on the angular distance to the oriented edge normal is proposed. In total, for an edge point (x, y) with gradient orientation ϕ_{xy} and uncertainty $2\Delta\alpha$, we

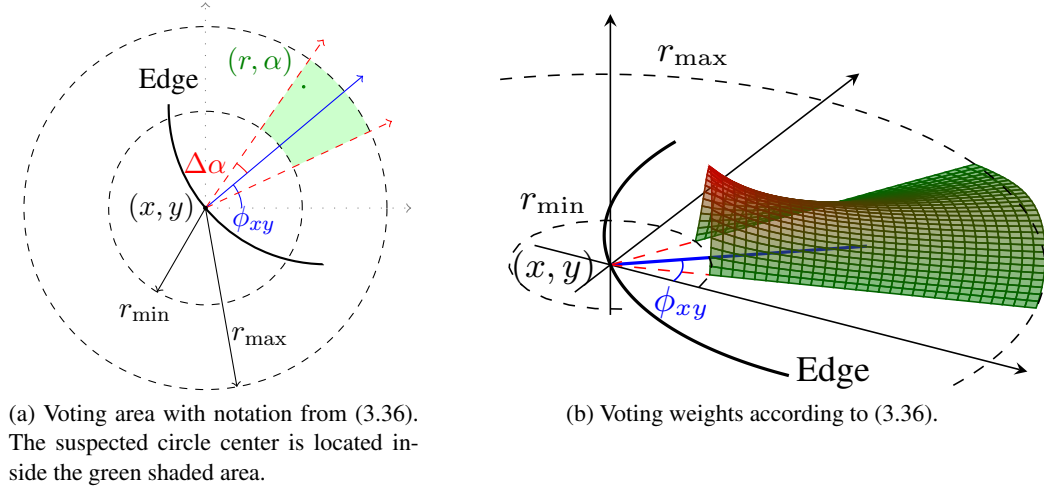


Figure 3.8: CHT voting mechanism for an edge pixel (x, y) .

define the weight function

$$W_{xy}(r, \alpha) := \frac{1}{r} \left(\exp \left(-\frac{(\alpha - \phi_{xy})^2}{\Delta\alpha^2} \right) - \frac{1}{e} \right), \quad (3.36)$$

where (r, α) is given in polar coordinates with origin (x, y) and

$$r \in [r_{\min}, r_{\max}], \quad \alpha \in [\phi_{xy} - \Delta\alpha, \phi_{xy} + \Delta\alpha].$$

In accordance with the size of the reference particles at hand, we select $r_{\min} = 5$ and $r_{\max} = 40$.

Figure 3.8(a) illustrates the voting area in conformity with the notation from above, while the distribution of the voting weights according to (3.36) is shown in Figure 3.8(b). The beneficial impact of this method on the accumulator array construction and local maximum extraction is demonstrated in Figure 3.12(b) where the prerequisites coincide with those from Figure 3.11(c) except for the weights $W(r, \alpha)$ which have been replaced by

$$\widetilde{W}(r, \alpha) := \frac{1}{r}, \quad \text{for } r \in [r_{\min}, r_{\max}], \quad \alpha \in [\phi_{xy} - \Delta\alpha, \phi_{xy} + \Delta\alpha]. \quad (3.37)$$

In this case, the detection of both ring-shaped particles fails due to the inferior design of the accumulator array.

Local maximum detection in the CHT

In the traditional sense, an element of a two-dimensional array is referred to as a local maximum if its value dominates that of its immediate (eight) neighbor pixels. For the sake of robustness, this condition is tightened: in order for a position (x, y) in the accumulator array A to be considered as a candidate for a circle center, it must be the strict maximum in a 7×7 square neighborhood (with 48 neighbor pixels):

$$A(x, y) > A(x + i, y + j) \quad \forall i, j \in \{-3, -2, -1, 1, 2, 3\}. \quad (3.38)$$

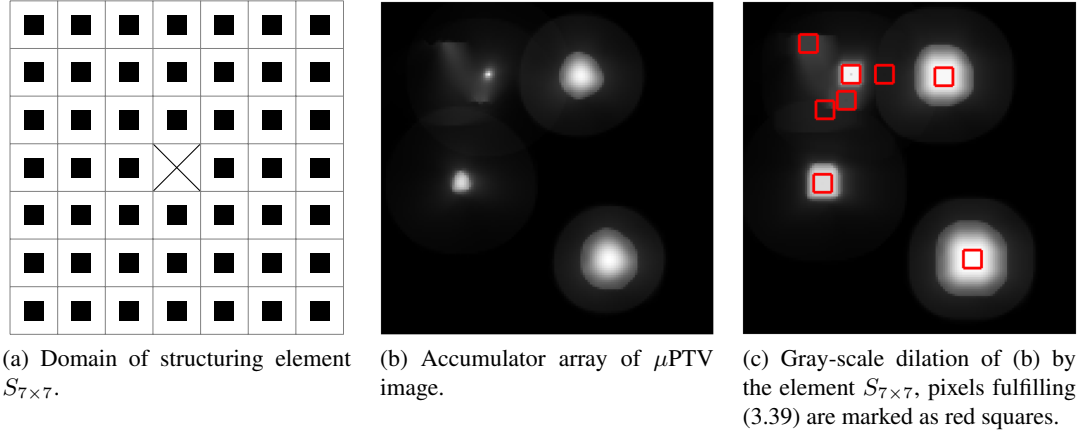


Figure 3.9: Local maximum extraction.

Within the scope of mathematical morphology, the gray-value *dilation* and *erosion* operators \oplus and \ominus , respectively, of a gray-scale image B by a structuring element S are defined as

$$(B \oplus S)[x, y] = \max_{(i, j) \in D_S} \{B(x - i, y - j) + S(i, j) : [x - i, y - j] \in D_B\},$$

$$(B \ominus S)[x, y] = \min_{(i, j) \in D_S} \{B(x - i, y - j) - S(i, j) : [x - i, y - j] \in D_B\},$$

where D_S and D_B denote the domains of S and B respectively. The condition (3.38) therefore can be expressed as

$$(A \oplus S_{7 \times 7})[x, y] > A(x, y), \quad (3.39)$$

where $S_{7 \times 7}$ denominates the flat (zero valued) object shown in Figure 3.9(a). Points belonging to its domain are denoted as small black squares, the origin with coordinates (0,0) is marked as a diagonal cross. Since the standard morphological operations are provided in the MATLAB Image Processing Toolbox, (3.39) constitutes an elegant way of local maximum extraction. The impact of gray-value dilation on the accumulator array image from Figure 3.9(b) is demonstrated in Figure 3.9(c). Pixels fulfilling (3.39) are highlighted by red squares.

Dilation and erosion are dual transformations [55], in the sense that

$$(X \ominus Y)^c = X^c \oplus \check{Y},$$

where $X^c = \{x : x \notin X\}$ denotes the set complement and $\check{X} = \{-x : x \in X\}$ is the reflection of X . For a list of other properties and the introduction of further morphological operations, we refer to [55, 73].

In order to reduce the number of erroneously detected particle center candidates (false positives), in addition to (3.39), further restrictions are established: the first two conditions (M1) and (M2) are based on accumulator array properties while (M3) and (M4) refer to the detected shapes in the particle image.

- (M1)** Depending on the extent of the radius search range, the accumulator array is thresholded with the value of a p -quantile of its non-zero elements (e.g. $p = 0.85$). Due to performance issues, this step is performed prior to the dilation operation.
- (M2)** Two local maxima must be situated at a minimum distance d apart from each other (e.g. $d = 10$ pixels). Initially, the candidates are sorted by descending accumulator values, yielding a list of coordinates m_i . Let $S_d(m_i) := \{x \in \mathbb{R}^2 : \|x - m_i\| \leq d\}$ be the circular neighborhood of pixel m_i with radius d . In the case of multiple particle occurrences in immediate vicinity, the weakest candidates with lower accumulator values $\{m_j : m_j \in S_d(m_i) \text{ for any } i < j\}$ are dismissed. This constraint is reasonable since it emerged that (indistinguishable) particle clusters constitute the major source of error (see Chapter 4). The remaining candidates are subject to a strict local maximum condition in the extended neighborhood

$$S^i := S_d(m_i) \cap \left(\bigcup_{j < i} S_d(m_j) \right)^c.$$

The exclusion of the neighborhoods of stronger candidates from the local maximum search area benefits the detection of weak, true particle centers which are between d and $2d$ pixels apart from a strong particle center. As with the 7×7 square neighborhood, the constraint

$$A(m_i) > A(x) \quad \forall x \in S^i$$

is realized in terms of a dilation by the appropriate structuring element according to S^i .

- (M3)** The variance of the normalized image brightness values of the pixels along each circle line (with varying radius) surrounding the center candidate must fall below a certain threshold $t_{nv} \in [0, 1]$. As a result, centers of eminently deformed objects are eliminated. However, the threshold value for this condition is not selected in a strict manner (e.g. $t_{nv} = 0.25$) since minor particle shape imperfections are common in the available μ PTV images and should not be penalized.
- (M4)** The ratio of valid gradient vectors of the pixels along at least one surrounding circle line must exceed a certain threshold t_v . This condition will be clarified in the forthcoming description of the radius determination process.

Finally, the remaining local maximum locations in the accumulator array are refined by the computation of local centroid positions.

In the accumulator array images (center column) in Figure 3.11 and 3.12(a)-(b), all locations where (3.39) is fulfilled are tagged. The scenario of Figure 3.12(c) uses the respective $S_{3 \times 3}$ “standard” element instead of $S_{7 \times 7}$. In this case, the number of false center candidates more than doubles; due to the radius determination process. Red coloring signifies that at least one of the conditions (M1) and (M2) is not satisfied, i.e. the accumulator votes for the local maximum candidate are not decisive enough. Blue colored markers imply that both (M1) and (M2) are

fulfilled whereas (M3) or (M4) does not pertain, i.e. characteristics in the original image do not suggest a circular object at the position in question. Green tags constitute the particle centers satisfying all constraints (M1)-(M4).

Radius determination in the CHT

The reference matrix image displayed in Figure 1.6(b) motivates that the strongest edge of a μ PTV particle is not necessarily located at its outer visible boundary. Thus, the outer edge of diffuse particles is often insufficiently captured by the Canny algorithm, as seen in Figure 3.11. Therefore, the implementation of the radius histogram technique mentioned in Section 3.2.2.2 is not optimal. Instead, the behavior of the gradient vectors on circle lines surrounding the considered particle center candidate is observed, as described in condition (M4). The largest number R within the radius search range fulfilling the following two conditions is chosen as the particle radius:

- (R1) The ratio v_R of valid gradient vectors on the circle line with radius R exceeds a certain threshold: $v_R \geq t_v$ (e.g. $t_v = 0.5$, black dashed line in Figure 3.10). A gradient vector is called valid, if
 - its magnitude exceeds a threshold t_m (which is in the magnitude of the lower Canny threshold t_0),
 - the deviation of its orientation to the orientation of the optimal vector (pointing towards the center) is in a moderate range (e.g. lower than $\pi/6$).
- (R2) The three previously computed smaller circle lines do not contain a considerably higher number of valid gradient vectors, i.e.: $v_R \geq 0.95 \cdot \max(v_{R-1}, v_{R-2}, v_{R-3})$. In this way, the selection of a significantly too large radius due to unfitting (too low) thresholds t_m, t_v is avoided.
- (R3) The circular image region with radius R around the considered particle must not be occluded by more than 50 % by the circular regions of other particles. The background for the formulation of this condition is the fact that the particle matching algorithms described in the next section solely operate on the pixel area not occluded by other particles.

As mentioned above, if no radius fulfilling those conditions is found, the potential particle center is tagged as a false positive and rejected (blue squares in the accumulator array images). Figure 3.10 displays the ratio of valid gradient votes for the particles detected in Figure 3.11(c), according to condition (R1). The radius chosen based on (R2) is marked by a diamond.

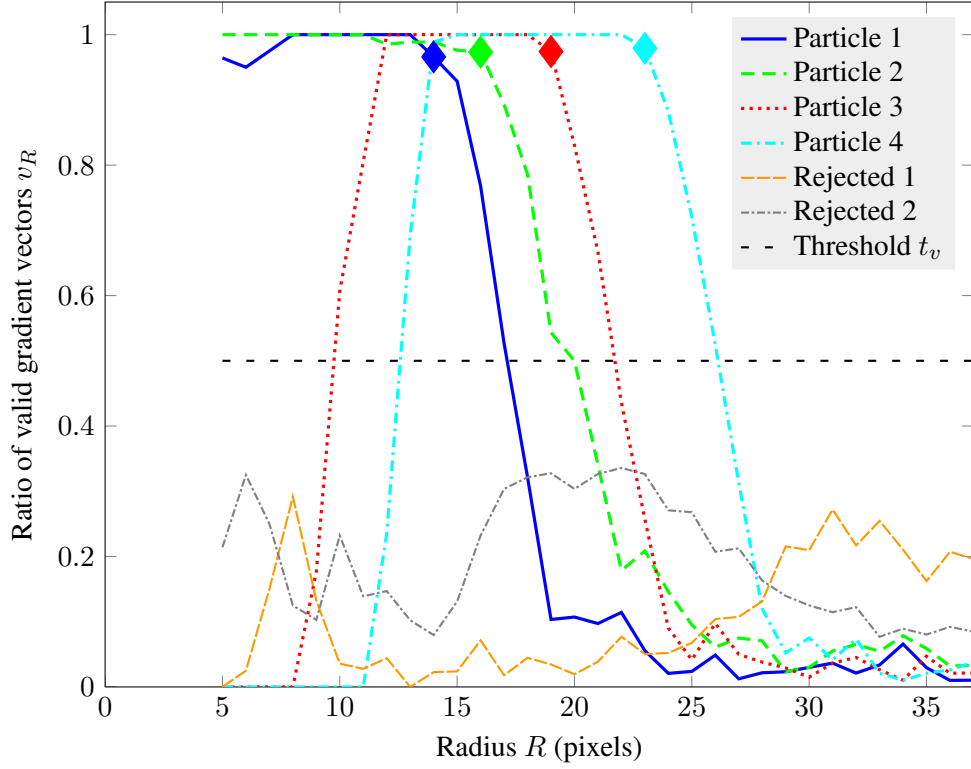


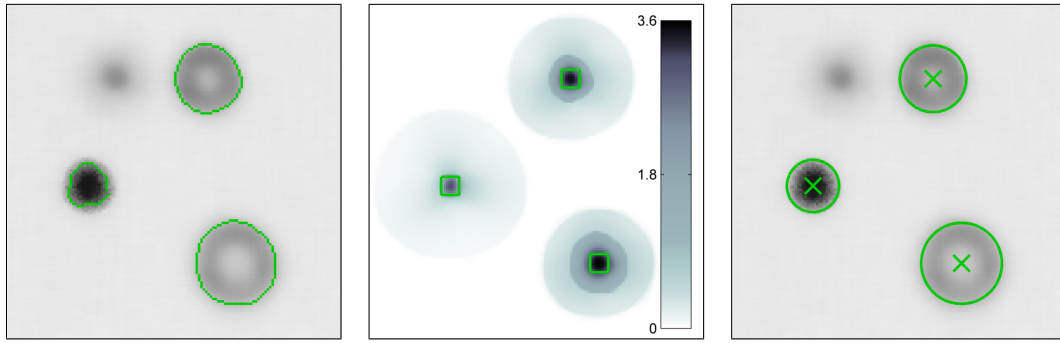
Figure 3.10: Radius determination process of the scenario shown in Figure 3.11(c). The selected particle radii are marked as diamonds.

3.3 Step 3: Particle Matching

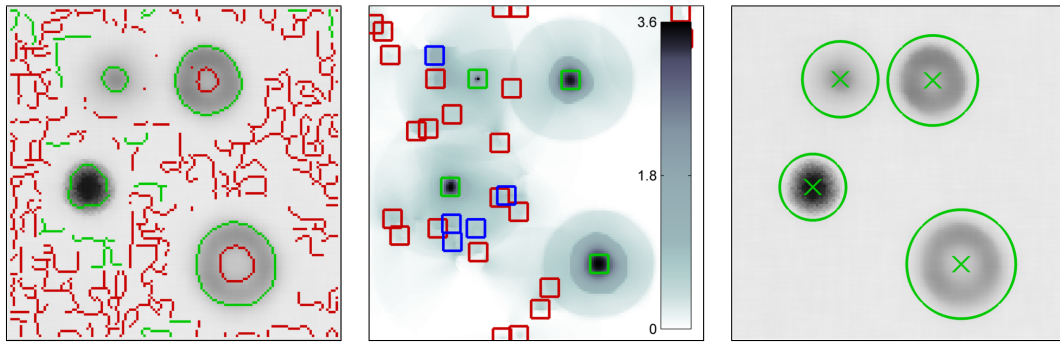
Subsequent to the successful detection of a particle, its position in the film is determined by locating the most resembling reference image. For this purpose, in their μ PTV application, Paschke et al. [56] implemented a cross-correlation based technique. The concept of cross-correlation matching is described in Section 3.3.1. As an alternative, an improved approach based on normalized radial intensity profiles (brightness profiles) is presented in Section 3.3.2.

A descriptive feature of a circular particle is given by its radius. Therefore, consistence of the radius values (e.g. maximum difference: 2 pixels) is used as a necessary condition for two particles to match. The radius determination process implemented in the current application is described in Section 3.2.3.

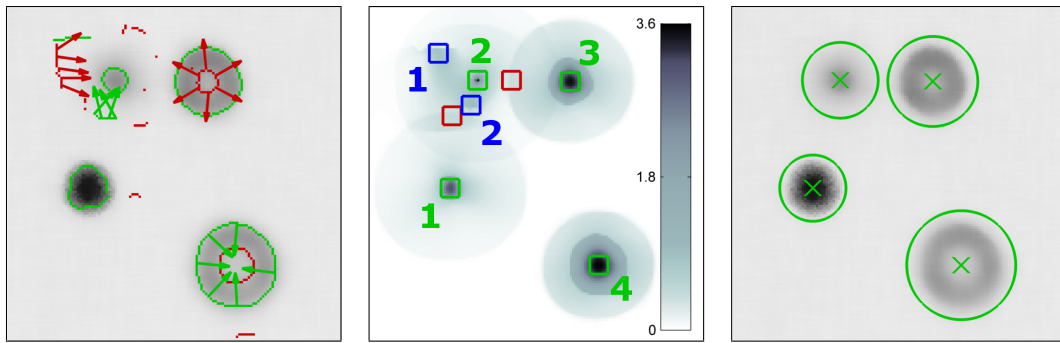
The results in Section 4 demonstrate that the proposed technique based on radial intensity profiles not only provides the expected improvement concerning temporal performance, but also yields a superior reliability in the matching process compared to cross-correlation based algorithms.



(a) Canny thresholds selected too high.

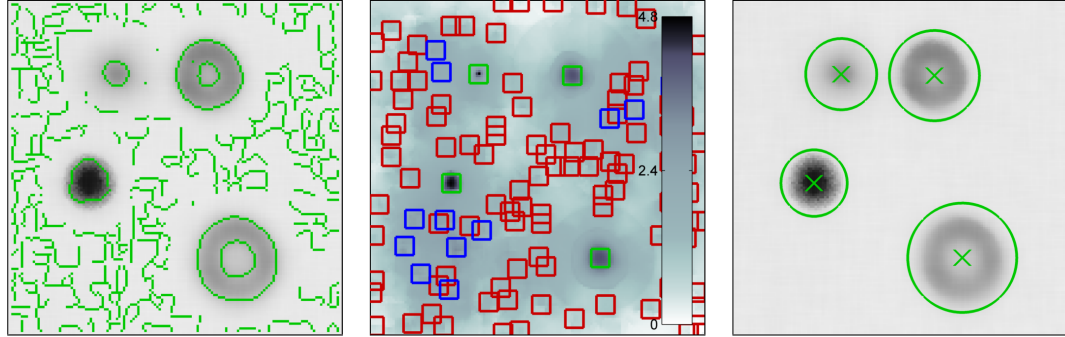


(b) Canny thresholds selected too low.

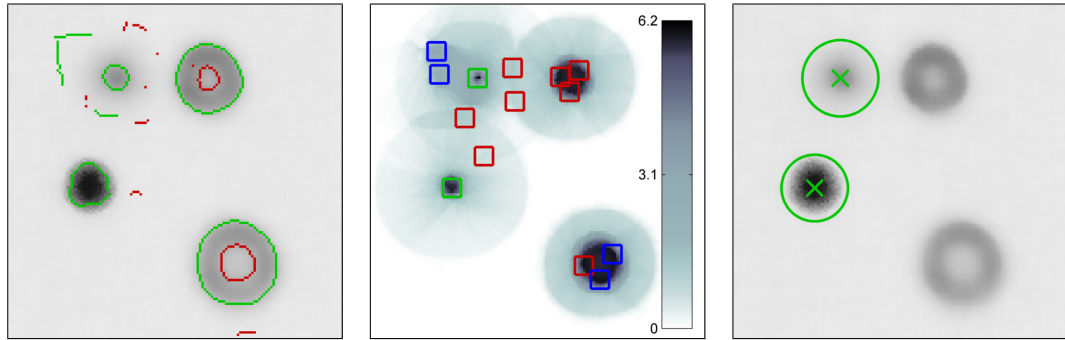


(c) Canny thresholds selected appropriately.

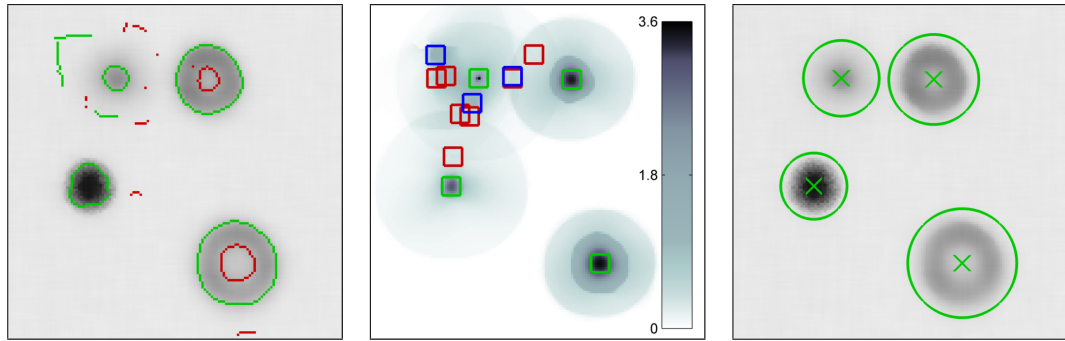
Figure 3.11: Particle detection algorithm, different Canny thresholds. Left Column: inverted, pre-processed images with detected edges (eliminated edges colored in red), the arrows in (c) represent the gradient vectors of selected edge pixels. Center Column: corresponding accumulator arrays with tagged local maxima, the labels in (c) are used as reference in Figure 3.10 and 3.13. Right Column: eventually detected particles on inverted original image.



(a) No elimination of disturbing edges.



(b) Voting weights constant for each search radius.



(c) Local maximum search in traditional 3×3 mask.

Figure 3.12: Particle detection algorithm, without certain modifications. Left Column: inverted, pre-processed images with detected edges (eliminated edges colored in red). Center Column: corresponding accumulator arrays with tagged local maxima. Right Column: eventually detected particles on inverted original image.

3.3.1 Normalized Cross-Correlation

Cross-correlation constitutes a standard approach for feature detection in the field of image processing, as cited by Lewis [41]. For a square, discrete template image $t \in \mathbb{R}^{N \times N}$, which is to be localized in an image $f \in \mathbb{R}^{M \times M}$, $M \geq N$, the cross-correlation values $c(u, v)$ of f and t are given by

$$c(u, v) := (f \star t)(u, v) := \sum_{x, y} f(x, y) t(x - u, y - v), \quad (3.40)$$

where the sum is computed in the region under the (u, v) -translated, $N \times N$ -sized window. The best matching position of t within the image f is then determined by the offset (u, v) which maximizes $c(u, v)$.

Template matching using standard cross-correlation is not robust since (3.40) strongly depends on the local image brightness in the image f : brighter image areas yield higher correlation values. To eliminate this disadvantage, both the image f and the template t are normalized by subtracting the mean and dividing by the standard deviation, yielding the Normalized Cross-Correlation (NCC) values $\gamma(u, v) \in [-1, 1]$

$$\gamma(u, v) = \frac{\sum_{x, y} [(f(x, y) - \overline{f_{u, v}})(t(x - u, y - v) - \bar{t})]}{\sqrt{\sum_{x, y} [(f(x, y) - \overline{f_{u, v}})^2] \sum_{x, y} [(t(x - u, y - v) - \bar{t})^2]}}, \quad (3.41)$$

where \bar{t} is the mean brightness of t and $\overline{f_{u, v}}$ corresponds to the mean brightness of f in the window corresponding to the offset (u, v) .

However, the direct computation of $\gamma(u, v)$ via (3.41) is computationally expensive: although the normalized version of the template t

$$t'(x - u, y - v) := \frac{t(x - u, y - v) - \bar{t}}{\sqrt{\sum_{x, y} [(t(x - u, y - v) - \bar{t})^2]}}$$

can be computed in advance (in only $5N^2 + 2$ arithmetic operations), the computational complexity of the rewritten NCC term

$$\gamma(u, v) = \frac{\sum_{x, y} [(f(x, y) - \overline{f_{u, v}}) t'(x - u, y - v)]}{\sqrt{\sum_{x, y} [(f(x, y) - \overline{f_{u, v}})^2]}} \quad (3.42)$$

is

$$\mathcal{O}(M^2 N^2), \quad (3.43)$$

both for numerator γ^{num} and denominator γ^{denom} of (3.42) [12, 41]. Due to the correlation theorem (3.44)

$$\begin{aligned} \mathcal{F}(g \star h) &= \mathcal{F}(g) \cdot \mathcal{F}^*(h), \quad \text{or equivalently} \\ g \star h &= \mathcal{F}^{-1}(\mathcal{F}(g) \cdot \mathcal{F}^*(h)) \quad \text{if } \mathcal{F}^{-1} \text{ well defined,} \end{aligned} \quad (3.44)$$

where \mathcal{F} is the Fourier transform and \mathcal{F}^* its complex conjugate, γ^{num} can be computed by making use of the frequency domain since (3.42) implies that $\gamma^{\text{num}} = (f - \overline{f_{u,v}}) \star t'$. The complexity of this operation amounts to $\mathcal{O}(M^2 \log_2(M))$ [41], which in particular is an improvement to the complexity of the direct computation (3.43) for (relatively) large template widths N . A different, faster approach for the calculation of γ^{num} is presented by Briechle and Hanebeck [12] who approximate γ^{num} with the aid of special basis functions at the cost of an emerging approximation error.

For an efficient computation of the denominator γ^{denom} , a simple expansion yields

$$\begin{aligned} \sum_{x,y} (f(x,y) - \overline{f_{u,v}})^2 &= \sum_{x,y} [f(x,y)^2] - 2 \sum_{x,y} [f(x,y) \overline{f_{u,v}}] + \sum_{x,y} [\overline{f_{u,v}}^2] \\ &= \sum_{x,y} [f(x,y)^2] - \frac{2}{N} \left(\sum_{x,y} [f(x,y)] \right)^2 + \frac{1}{N} \left(\sum_{x,y} [f(x,y)] \right)^2 \\ &= \sum_{x,y} [f(x,y)^2] - \frac{1}{N} \left(\sum_{x,y} [f(x,y)] \right)^2. \end{aligned}$$

As demonstrated by Lewis [41], the remaining two summation terms can be efficiently computed from running sum tables $s(u, v)$, $s^2(u, v)$, defined by

$$\begin{aligned} s(u, v) &= f(u, v) + s(u-1, v) + s(u, v-1) - s(u-1, v-1), \\ s^2(u, v) &= f^2(u, v) + s^2(u-1, v) + s^2(u, v-1) - s^2(u-1, v-1), \end{aligned}$$

with $s(u, v) = s^2(u, v) = 0$ when either $u, v < 0$. With this notation,

$$\begin{aligned} \sum_{x=u}^{u+N-1} \sum_{y=v}^{v+N-1} f(x, y) &= s(u+N-1, v+N-1) - s(u-1, v+N-1) \\ &\quad - s(u+N-1, v-1) + s(u-1, v-1), \\ \sum_{x=u}^{u+N-1} \sum_{y=v}^{v+N-1} f^2(x, y) &= s^2(u+N-1, v+N-1) - s^2(u-1, v+N-1) \\ &\quad - s^2(u+N-1, v-1) + s^2(u-1, v-1), \end{aligned}$$

which implies that subsequent to the initialization of s and s^2 , consuming $\mathcal{O}(M^2)$ arithmetic operations, solely three additions/subtractions per offset (u, v) are required to evaluate the sums over f and f^2 .

The approach presented above is exclusively designed for rectangular template regions, i.e. the sums over x, y in the NCC representations are computed for a rectangular domain. In the

current particle matching application, however, it is desired to reduce the (perturbing) influence of nearby particles, which motivates the use of circular template regions. Moreover, the exclusion of the area of intersecting particles from the template region increases the reliability of particle cluster dissection, as demonstrated in Chapter 4.

In practice, the computation of the whole correlation matrix is redundant since solely the NCC value for that offset (u_0, v_0) , which associates the known particle center locations in both images, is relevant. Due to potentially imprecise center coordinates, the calculation of $\gamma(u, v)$ is performed in a small $(3 \times 3$ or $5 \times 5)$ neighborhood of (u_0, v_0) , straight forward via (3.41) or (3.42), respectively.

Each detected particle from a μ PTV recording is then matched to all reference particles and that with the highest NCC score then determines the allocated position in the flow. In order to impede the detection of false positives, this score must exceed a predefined threshold, otherwise the detected particle is discarded. Since large, ring-shaped particles are more prone to shape imperfections than small, bright particles, this threshold is adjusted with regards to the considered layer. In practice, it was found that the values 0.9 (for the first, diffuse particle), 0.97 (for the focused, sharp particle), and 0.7 (for the last, ring-shaped particle), with linear combinations of those values for the layers in between, represent an adequate parameter choice.

In any case, the calculation of the NCC represents the most time-consuming component of the presented algorithm as will be shown in Chapter 4.

3.3.2 Proposed Method: Radial intensity profiles

In this work, we propose an improved matching technique based on normalized radial intensity profiles (brightness profiles). Radiating from the particle center, the profiles are computed by averaging the intensities on surrounding circle lines, followed by a scaling to $[0, 1]$ which ensures contrast invariance. Here, the surrounding circle line of the center pixel c with radius r consists of all image pixels x which fulfil

$$r - \frac{1}{2} \leq \|x - c\| \leq r + \frac{1}{2}.$$

In order to increase the accuracy of the profiles, they are considered with half a pixel distance between two adjacent grid points, i.e. the surrounding circle lines are computed for $r = 1, 1.5, 2, 2.5, \dots, r_{\max}$.

Interfering areas from other detected particles in close vicinity are excluded from the profile calculation process. Figure 3.13 displays the radial intensity profiles from the particles detected in Figure 3.11(c).

The sum of the pointwise, absolute distances between a reference profile and the considered particle profile then indicates their dissimilarity. Therefore, that reference particle with the lowest dissimilarity score represents the best match. Similar to the cross-correlation matching approach, the lowest dissimilarity score must undercut a predefined (layer index dependent) threshold for the particle not to be discarded.

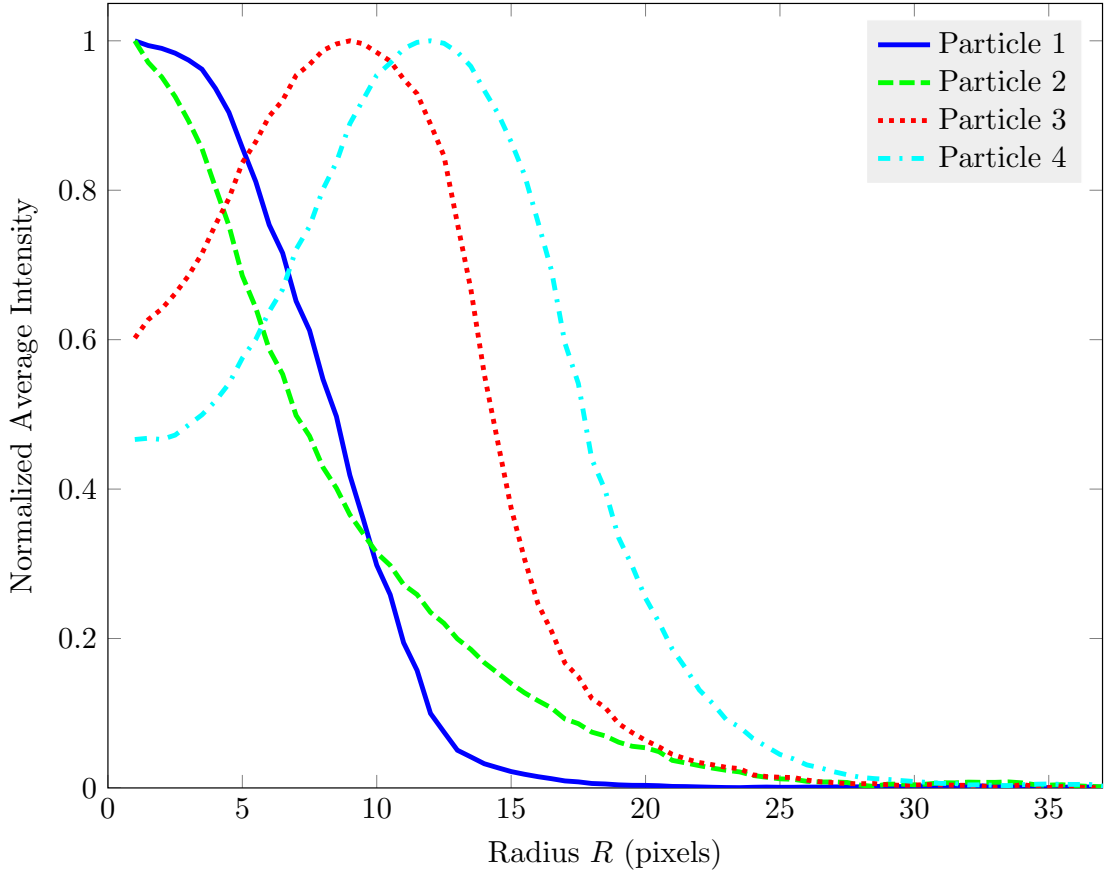


Figure 3.13: Normalized radial intensity profiles of the particles labeled in Figure 3.11(c).

3.4 Step 4: Particle Tracking

Due to the comparatively low particle density in the existing μ PTV recordings (approximately 60-100 particles, diameter between 10 and 80 pixels on 2048×2048 pixel images), the use of a standard tracking algorithm is sufficient. The algorithm implemented searches for the most plausible particle links in two frames by minimizing a linear cost function and was inspired by the work of Sbalzarini and Koumoutsakos [65]. This particular method was selected since a-priori knowledge about the particle motion can be easily included into the formulation of the tracking problem and the tools to solve the problem are available in the MATLAB Optimization Toolbox (version R2010a).

In order to provide theoretical background, next, a short overview of Linear Programming and Integer Programming techniques is given, followed by an explanation of the precise method implemented in the μ PTV application.

3.4.1 Linear Programming

In Linear Programming, a linear objective function is optimized with respect to linear equality and/or inequality constraints. The standard form of the general Linear Programming (LP) problem is given by [77]

$$\begin{aligned} & \text{maximize } c^T x \\ & \text{subject to } Ax \leq b, \\ & \quad x \geq 0. \end{aligned} \tag{3.45}$$

Here, $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ is a given matrix of coefficients, $c = (c_j) \in \mathbb{R}^n$ and $b = (b_i) \in \mathbb{R}^m$ are given vectors of coefficients and the values of the vector $x = (x_j) \in \mathbb{R}^n$ are to be determined. Note that each minimization problem can be viewed as a maximization problem after multiplying the vector c by -1 . Furthermore, it is clear that an equality constraint

$$\sum_{j=1}^n a_{ij} x_j = b_i$$

can be converted to standard form by writing it as a pair of inequalities:

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq b_i, \\ \sum_{j=1}^n -a_{ij} x_j &\leq -b_i. \end{aligned}$$

Finally, if a variable x_j is not supposed to be nonnegative, the substitution

$$x_j = x_j^+ - x_j^-$$

with $x_j^+ \geq 0$, $x_j^- \geq 0$ allows the introduction of unconstrained variables to the standard form [37].

Each of the $2n$ inequality constraints in (3.45) describes a half-space in \mathbb{R}^n which must contain the solution. The intersection of these half-spaces therefore constitutes the set of the so-called *feasible solutions* for the LP problem. For a maximum number of $n = 3$ variables, this geometric interpretation permits a graphical illustration. An example where $n = 2$ and

$$A = \begin{pmatrix} -1 & 3 \\ 1 & 1 \\ 2 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 12 \\ 8 \\ 10 \end{pmatrix}, \quad c = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

is displayed in Figure 3.14(a). In addition, two level sets of the objective function $c^T x$ are shown, which indicate that $x = (x_1, x_2)^T = (6, 2)^T$ represents the optimal solution.

As stated by Matoušek and Gärtner [46], “*tens of different algorithms have been suggested for linear programming over the years*”. In the textbooks on Linear Programming reviewed for

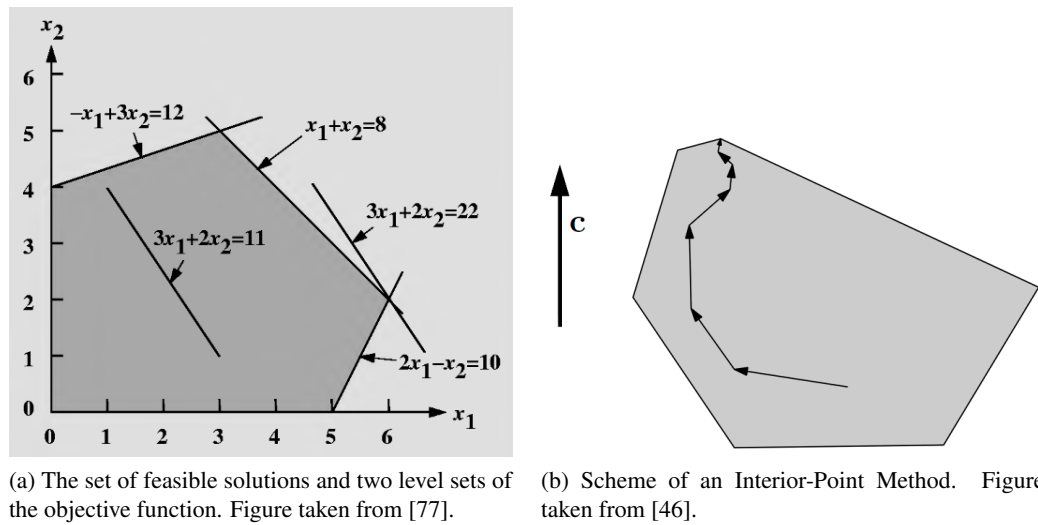


Figure 3.14: Graphical interpretation of a Linear Programming problem.

this section [37, 46, 77], the algorithms relevant for a practical application are essentially classified into two groups: techniques based on the **Simplex Method** and **Interior-Point Methods**. It can be shown [46] that if an optimal solution to (3.45) exists, it is reached at a “vertex” of the set of feasible solutions, cf. Figure 3.14(a). The basic idea of the Simplex Method is to find such a feasible starting solution and to gradually enhance it by moving along the edges from one vertex to another such that the value of the objective function grows until the optimum is obtained. In contrast, Interior-Point Methods “walk through the interior of the set of feasible solutions toward an optimum, carefully avoiding the boundary. Only at the very end, when they get almost to an optimum, they jump to an exact optimum by a rounding step” [46]. This behaviour is indicated in Figure 3.14(b).

The analytical procedure of both approaches is rather lengthy and best described with the aid of a detailed example. For this purpose, we refer to the above-mentioned literature [37, 46, 77] which provides an introduction into the field of Linear Programming. Moreover, dozens of tutorials and solvers (commercial software as well as freeware) have been made available online due to the popularity of those methods¹.

A comparison of the performance of techniques based on the Simplex Method and Interior-Point Methods leads to the discovery that each approach has its advantages, depending on the particular problem [46, 77].

¹The first two of the following links address the Simplex Method, the latter deal with Interior-Point Methods (all accessed July 21, 2013):

http://people.hofstra.edu/stefan_waner/realworld/tutorialsf4/frames4_4.html

<http://www.zweigmedia.com/RealWorld/simplex.html>

<http://www.caam.rice.edu/~zhang/lipsol/>

http://www.diku.dk/OLD/undervisning/2006-2007/2006-2007_b2_426/interior.pdf

3.4.2 Integer Programming

A standard Integer Programming (IP) problem differs from the standard LP problem by the constraint that the variables x_j are required to be integers:

$$\begin{aligned} & \text{maximize } c^T x \\ & \text{subject to } Ax \leq b, \\ & \quad x \geq 0, \\ & \quad x \in \mathbb{Z}^n. \end{aligned} \tag{3.46}$$

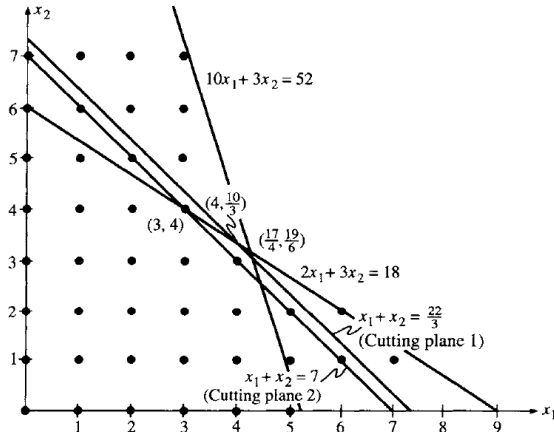
In general, two classes of algorithms dealing with (3.46) are mentioned [37, 46, 67]: **Cutting Plane Methods** and **Branch-and-Bound Methods**. Initially, both classes apply so-called *LP relaxation*, where the integrality constraint in (3.46) is omitted, i.e. a standard LP problem is solved (e.g. by means of the Simplex Method). It can be proven that if the LP relaxation has at least one feasible solution, then (3.46) also has at least one optimal solution [46]. If by chance an optimal solution \tilde{x} of the LP relaxation fulfils $\tilde{x} \in \mathbb{Z}^n$, then \tilde{x} obviously is an optimal solution of (3.46). In the general case $\tilde{x} \notin \mathbb{Z}^n$, the value of $c^T \tilde{x}$ represents an upper bound for the objective function of the IP problem and further investigation is necessary.

Cutting Plane Methods successively add new constraints which do not affect the set of feasible integral solutions but “cut off” a part of the non-integral solutions, including the optimal solution \tilde{x} . This procedure is repeated until - after a finite number of iterations - the updated optimal solution is integral. An explicit representation for the creation of such cutting planes is available (*Gomory cuts*) [37, 46]. Figure 3.15(a) demonstrates this process for the problem

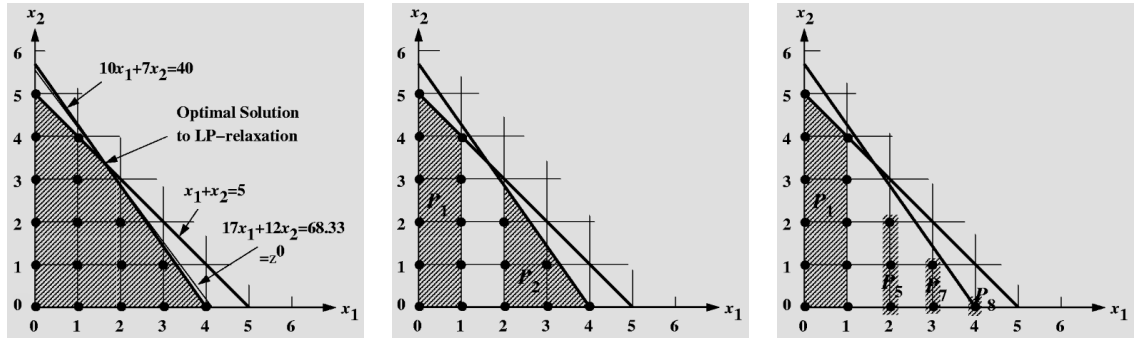
$$\begin{aligned} & \text{maximize } z = 5x_1 + 6x_2 \\ & \text{subject to } 10x_1 + 3x_2 \leq 52, \\ & \quad 2x_1 + 3x_2 \leq 18 \\ & \quad x_1 \geq 0, x_2 \geq 0, \text{ integers.} \end{aligned} \tag{3.47}$$

The optimal solution of the LP relaxation of (3.47) is $\tilde{x} = (x_1, x_2) = (\frac{17}{4}, \frac{19}{6})$, with objective value $z = 40.25$. After cutting off non-integral solutions with the aid of Cutting plane 1, the updated optimal solution $(x_1, x_2) = (4, \frac{10}{3})$ is still non-integral (with objective value $z = 40$). The second iteration finally yields the optimal solution of (3.47) as $(x_1, x_2) = (3, 4)$, with maximum objective value $z = 39$.

Branch-and-Bound Methods select a non-integral component of the optimal solution of the LP relaxation \tilde{x}_j and split the integer program (3.46) into two “smaller” integer programs (branching step): The first one is subject to the (additional) constraint $\lfloor x_j \rfloor \leq \tilde{x}_j$, for the second one the constraint $\lceil x_j \rceil \geq \tilde{x}_j$ is added. Subsequently, both problems are solved and the splitting process is repeated until (at least) one of the subproblems possesses an integral optimal solution, whose objective value then is a lower bound for the optimal objective value. In further consequence, only those subproblems where the maximum objective value exceeds this lower bound need to be considered (bounding step). “*In the worst case the bounding step may not prune any subtree, but in many practical applications it results in enormous savings and allows for solving*



(a) Principle of a Cutting Plane Method. Figure taken from [37].



(b) Iteration process of a Branch-and-Bound Method. Figures taken from [77].

Figure 3.15: Approaches for the Integer Programming problem.

large integer programs” [46]. In Figure 3.15(b), the problem

$$\begin{aligned}
 &\text{maximize } z = 17x_1 + 12x_2 \\
 &\text{subject to } 10x_1 + 7x_2 \leq 40, \\
 &\quad x_1 + x_2 \leq 5 \\
 &\quad x_1 \geq 0, x_2 \geq 0, \text{ integers}
 \end{aligned} \tag{3.48}$$

is discussed. In the first frame, the set of feasible solutions of the LP relaxation is shaded. Its optimal solution is $\tilde{x} = (x_1, x_2) = (\frac{5}{3}, \frac{10}{3})$ with maximum objective value $z = 68.33$, and the thin line represents the corresponding level set of the objective function. In the first branching step, the set of feasible solutions is splitted along the x_1 -axis, thus generating the two feasible subregions P_1 and P_2 , see the second frame of Figure 3.15(b). The optimal solution of P_1 at $(x_1, x_2) = (1, 4)$ with an objective value of $z = 65$ represents a first feasible solution to (3.48). However, since the optimal solution of P_2 at the (non-integral) point $(x_1, x_2) = (2, \frac{20}{7})$ possesses the higher objective value $z = 68.29$, P_2 requires the application of further branching

steps. In the end, it is found that $(x_1, x_2) = (4, 0)$ with objective value $z = 68$ is the optimal solution to (3.48), marked as P_8 in the third frame of Figure 3.15(b).

A famous task which can be formulated as an IP problem is the **Traveling Salesman Problem** where a salesman needs to visit a number of cities (with known pairwise distances). The goal is to minimize the total traveling distance under the constraint that each city is visited exactly once and that the salesman returns to the home city in the end (see [77] for more details).

Another example is the so-called **Knapsack Problem**: consider a hiker who can carry k kilograms of equipment at most. Each item is described by its weight a_j and its value c_j . The goal is to decide which items to carry in order to maximize the total value of all packed items, in accordance with the weight limitation. Let $x_j = 1$ if the j th item is chosen and $x_j = 0$ if it is not chosen. Then the mathematical formulation is

$$\begin{aligned} & \text{maximize} \quad \sum_{j=1}^n c_j x_j \\ & \text{subject to} \quad \sum_{j=1}^n a_j x_j \leq k, \\ & \quad \quad \quad x_j \in \{0, 1\}, \quad j = 1, \dots, n. \end{aligned}$$

The special case of IP where the variables x_j are restricted to the set $\{0, 1\}$ is referred to as zero-one programming problem [37].

3.4.3 Proposed Implementation

In this section, the implementation of a special Integer Programming problem for the current μ PTV application is motivated and described in detail.

The goal is to determine the - in some sense most plausible - set of particle links between both frames of a μ PTV recording. Let $\{p_i : i = 1, \dots, m\}$ and $\{q_j : j = 1, \dots, n\}$ be the set of particles detected in Frame 1 and Frame 2, respectively. We define the binary, so-called association matrix $X = (x_{ij})$ as

$$x_{ij} = \begin{cases} 1 & \text{if } p_i \text{ and } q_j \text{ are produced by the same particle,} \\ 0 & \text{otherwise.} \end{cases}$$

Due to entrance and exit of particles near the image border or detection errors, it is possible that the same physical particle which appears on one frame is not detected on the other. On that account, the association matrix is augmented with a column $x_{i,n+1}$ (for particles disappearing in Frame 2) and a row $x_{m+1,j}$ (for particles newly appearing in Frame 2), containing links to so-called “dummy particles”. Thus, each particle must be either linked to a particle from the

other frame or to a dummy particle, leading to the constraints for the matrix X :

$$\begin{aligned} \sum_{j=1}^{n+1} x_{ij} &= 1 \quad \forall i = 1, \dots, m, \\ \sum_{i=1}^{m+1} x_{ij} &= 1 \quad \forall j = 1, \dots, n. \end{aligned} \quad (3.49)$$

To find an optimal set of links (x_{ij}) , the following linear cost functional Φ is minimized:

$$\Phi = \sum_{i=1}^{m+1} \sum_{j=1}^{n+1} c_{ij} x_{ij} \rightarrow \min, \quad (3.50)$$

where c_{ij} represents the cost of associating the particle p_i with particle q_j . The cost of linking a particle to one of the dummy particles is set to a predefined constant threshold C . Associations with a cost larger than this value are impossible and do not need to be considered in the optimization problem. By denoting the 3-dimensional physical position (determined in the detection and matching process) of the particles p_i, q_j in the flow as P_i, Q_j , the selection $c_{ij}^n := \|Q_j - P_i\|$ represents the nearest-neighbor method which minimizes the total sum of particle distances between both frames.

In practice, however, it is beneficial to include a-priori physical knowledge about the fluid motion. Let $E(p_i)$ be a reliable estimation of the displacement vector of particle p_i , then the use of

$$c_{ij}^p := \|Q_j - (P_i + E(p_i))\|$$

constitutes an appropriate adjustment. In the following, a method of computation for $E(p_i)$ for the current application, based on a physical background, is presented. As a matter of fact, this method is applicable for all planar flows, i.e. where all flow velocity vectors can be described in a two-dimensional coordinate system.

In his investigations of the condensation process, Nusselt [49] deduced the following formula for the velocity field of a (laminar) falling film on a vertical plate:

$$u(x, y, z) = \frac{g}{v} \left(\delta z - \frac{z^2}{2} \right) e_x, \quad (3.51)$$

where $g = 9.81 \frac{m}{s^2}$ is Earth's gravitational constant, v denotes the kinematic viscosity of the considered liquid, δ is the film thickness (in practice estimated or measured with a probe), z represents the distance to the backplate and e_x is the unit vector in flow direction x . Obviously, (3.51) describes a planar flow (in the (x, y) -planes) whose cross-sectional velocity profile is shown in Figure 3.16.

Since the recording camera is positioned perpendicular to the flow - cf. the picture of the experimental setup in Figure 1.8 on page 9 and the sketch in Figure 3.16 - the optical appearance of a particle is not expected to change between two frames. As a consequence, two particles p_i and q_j linked by the tracking algorithm must belong to the same layer. In practice, the strong

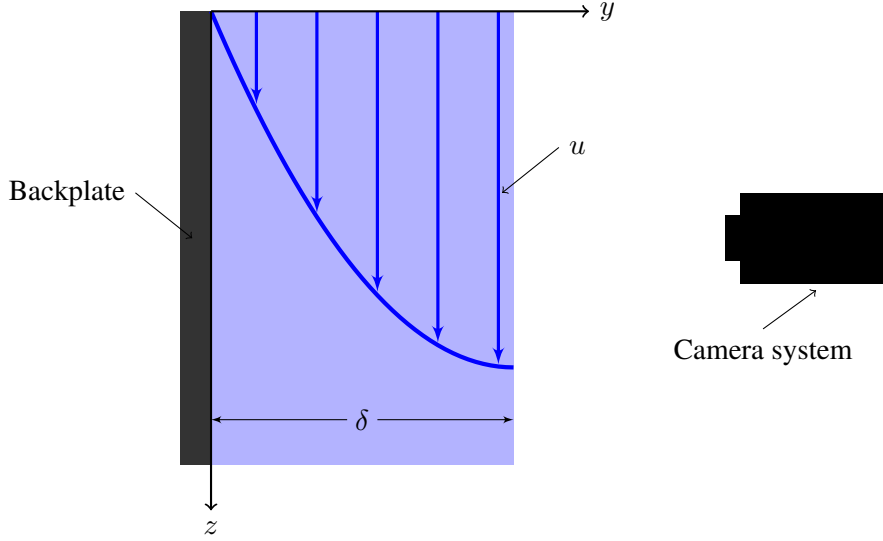


Figure 3.16: Velocity profile of a falling film according to Nusselt [49]. Figure inspired by [47].

resemblance of specific reference images (cf. Figure 1.6(b) on page 7: the reference particles near to the focal plane are hardly distinguishable) as well as image noise impede the accurate particle classification. This statement is confirmed by the fact that the manual classification of an image from a synthetic dataset where cut-out reference images were randomly distributed (see Section 4.3 for a detailed description of this dataset) resulted in solely 71% exactly recovered matches whereas 29% of the associations were off by one layer. Moreover, the finite number of reference images (recorded at variable camera distances) accounts for variations of the shapes of flow particles between those camera distances.

Therefore, the condition of coinciding layer indices is loosened and the planar motion is computed in the “best” common layer l_{ij} of both particles. As a result, small deviations in the matching result of (one or both) particles can be compensated and the erroneous rejection of the particle link is prevented.

For this purpose, the set of possible layer indices is observed, i.e. the indices of those reference images for which the matching value exceeds (correlation matching) or undercuts (profile matching) the corresponding threshold for two considered particles p_i and q_j . In the case that this set is empty, a best common layer does not exist and the link $p_i \leftrightarrow q_j$ is impossible, i.e. $x_{ij} = 0$ and the variable x_{ij} is removed from the optimization problem (3.50) with side conditions (3.49). Otherwise, for cross-correlation matching, l_{ij} is chosen as that possible layer index where the sum of both NCC matching values is maximal; for profile matching, the index, where the sum of the profile matching values is minimal, is considered. In conclusion, the best common layer l_{ij} permits the computation of the distance y between the moving particle and the backplate. Equation (3.51) then yields the estimated displacement vector $E(p_i) = E(l_{ij})$.

In order to impede the association of two particles where the matching algorithm assigned different main layers (that with the best matching value), an additional cost term c_{ij}^d , quantifying

the shape dissimilarity of p_i and q_j , is used:

$$c_{ij} := c_{ij}^p + c_{ij}^d.$$

We set

$$c_{ij}^d := \frac{d_{\max}}{10} \cdot (S_{p_i}(l_{ij}) + S_{q_j}(l_{ij}) - 2), \quad (3.52)$$

where $S_{p_i}(l_{ij})$ is the position of l_{ij} in a list of (sorted from best to worst) matching layers of particle p_i (analogous for q_j). Thus, connections where both particles have the same main layer are not penalized by this term, since c_{ij}^d equals zero in this case. The value d_{\max} is the maximum allowed displacement of a particle in the second frame from its predicted position by means of $E(l_{ij})$. In particular, we set $d_{\max} = \sqrt{d_p^2 + d_n^2}$ where d_p and d_n describe the maximum allowed displacement in the primary flow direction and its normal, respectively. This means that each potential link $p_i \leftrightarrow q_j$ necessarily must fulfil $c_{ij}^p \leq d_{\max}$, otherwise it is declared impossible and $x_{ij} = 0$ (analogous to above). The factor $\frac{d_{\max}}{10}$ in (3.52) therefore adjusts the magnitude of c_{ij}^d to that of c_{ij}^p .

The choice of the parameters d_p and d_n is crucial for the successful execution of the tracking algorithm: on the one hand, a “too high” selection can cause a large number of possible particle links. As a result, the dimensionality of the optimization problem (3.50) with (3.49) increases which may affect the temporal performance of the algorithm. On the other hand, a “too low” selection can eliminate a correct link x_{ij} due to the inaccuracy of $E(l_{ij})$. In practice, choosing d_p approximately as the maximum displacement vector according to (3.51) and $d_n = \frac{d_p}{2}$ yields satisfying results: it avoids the omission of correct links but in general eliminates over 95% of the incorrect links.

Since $c_{ij} = c_{ij}^p + c_{ij}^d$, the cost C for a dummy link is composed of maximum acceptable values for c_{ij}^p and c_{ij}^d :

$$C = d_{\max} + 3 \frac{d_{\max}}{10} = 1.3 d_{\max}.$$

In this work, the binary linear optimization problem (3.50) combined with side conditions (3.49) is efficiently solved by the MATLAB Optimization Toolbox routine `bintprog` which applies a Branch-and-Bound Method, supported by the Simplex Method (see Section 3.4.1 and 3.4.2). For the given low particle density, more than two iteration steps have never been required to identify the optimal association matrix X .

Figure 3.17 illustrates the variables introduced above in a schematic image pair where one particle p_1 in the first frame is confronted with four potential linking particles q_1, q_2, q_3, q_4 in the second frame. Suppose that cross-correlation matching is selected and the matching threshold is $t = 0.95$. For the sake of visibility, we assume that the orientation of the displacement estimation vectors of two different layers is variable which is not the case in our application due to (3.51). Table 3.3 lists the (fictional) NCC matching values and the corresponding layer indices of all five particles appearing in the scene.

Table 3.4 indicates how the best common layer is computed. Moreover, it becomes clear that q_1 and q_2 remain as the only possible partners for p_1 since q_3 has no common layer with p_1 and

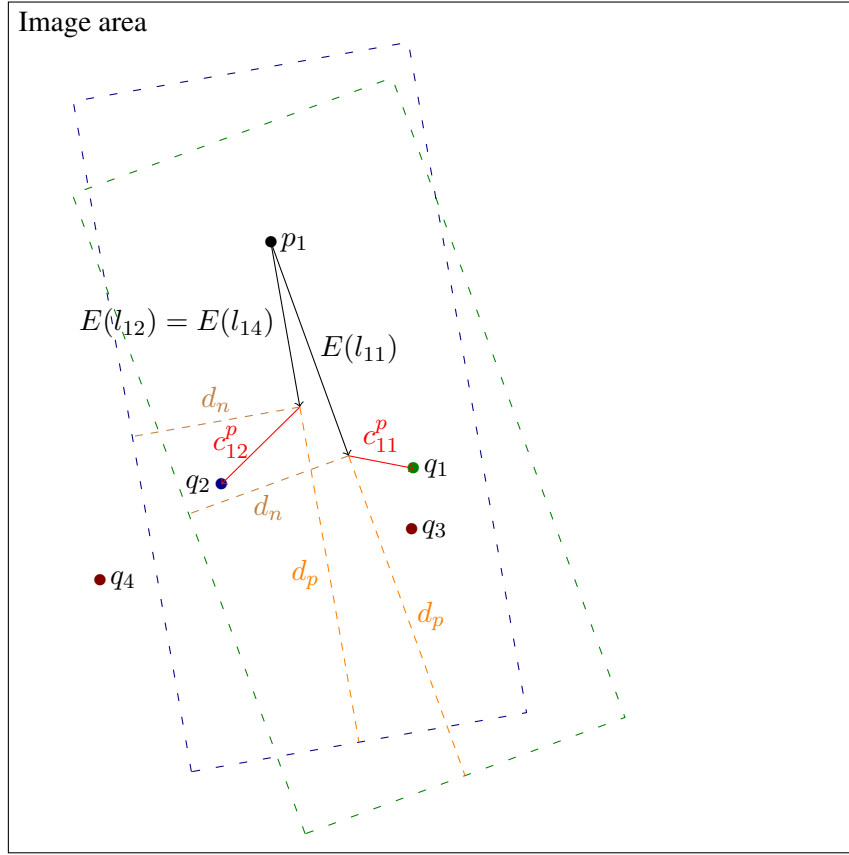


Figure 3.17: Illustration of variables used for particle tracking.

q_4 is too distant from the estimated particle position. The fifth column shows that $c_{11}^d < c_{12}^d$ and from Figure 3.17, it is evident that $c_{11}^p < c_{12}^p$. Therefore, the link $p_1 \leftrightarrow q_1$ is the most favorable in this situation.

3.5 Summary

This chapter gave an overview of the methods applied in this thesis. The whole approach is divided into four major tasks: image pre-processing, particle detection, particle matching and particle tracking. For each task, first the theoretical background was specified, followed by a detailed description of the specific algorithms implemented.

Concerning image pre-processing, the field of nonlinear diffusion filtering was introduced. Based on the fundamental work of Perona and Malik [58], the advanced nonlinear diffusion filter by Catté, Lions, Morel and Coll (CLMC) [14] was derived and its theoretical numerical stability properties were analyzed. In particular, two discretization schemes were assessed on μ PTV data: an explicit scheme and a semi-implicit variant (AOS scheme), where the former turned out to be somewhat more precise at the expense of a higher computational cost.

Table 3.3: (Fictional) particle matching properties of example situation in Figure 3.17.

Particle	MV ^a	CLI ^b
p_1	0.995, 0.99, 0.97, 0.96, 0.955	12, 10, 9, 8, 13
q_1	0.995, 0.98, 0.97, 0.96	10, 11, 12, 9
q_2	0.97, 0.96, 0.955	6, 8, 7
q_3	0.995, 0.97	20, 21
q_4	0.99, 0.97, 0.96	8, 7, 9

^a Matching Values (NCC) larger than threshold t , sorted in a decreasing manner

^b Corresponding Layer Indices

Table 3.4: Properties of potential particles links $p_1 \leftrightarrow q_j$.

Particle	SML ^a	CCLI ^b	l_{1j}	$S_{p_1}(l_{1j}) + S_{q_j}(l_{1j})$	d_p ok? ^c	d_n ok? ^d
q_1	1.985, 1.965, 1.93	10, 12, 9	10	$2 + 1 = 3$	✓	✓
q_2	1.92	8	8	$4 + 2 = 6$	✓	✓
q_3	—	—	—	—	—	—
q_4	1.95, 1.93	8, 9	8	$4 + 1 = 5$	✓	✗

^a Sum of Matching Values of common layers with p_1 , sorted in a decreasing manner

^b Corresponding Common Layer Indices with p_1

^c Distance to estimated displacement vector in primary flow direction within range?

^d Distance to estimated displacement vector normal to primary flow direction within range?

In order to detect particles on a CLMC filtered image, a modified Circular Hough Transform is used for the extraction of circular shapes from a (Canny) edge image. As for the practical realization, especially the method of parameter selection was illustrated and modifications for a superior performance of the method presented were proposed. These include an algorithm which eliminates disturbing edges from the edge image and the adoption of specific voting weights for the CHT.

Subsequently, two different particle matching algorithms were reviewed: the first, based on the Normalized Cross-Correlation coefficient, represents a (computationally expensive) standard matching technique already in use in a μ PTV application [56]. As an alternative, a simple method which compares radial particle brightness profiles is proposed. Promising results in the next chapter demonstrate the applicability of this matching technique.

The last section deals with particle tracking, i.e. searching for the correct association of detected and matched particles from two consecutive frames. For this purpose, a linear optimization problem was formulated as the minimization of a special cost function. Subsequent to a short overview of Linear and Integer Programming techniques, which gave an impression of how to solve such problems, the derivation of the approach implemented was founded on physical background knowledge according to Nusselt's Film Theory [49].

Evaluation and Results

In this chapter, the evaluation process of the μ PTV system implemented is described and results are provided. The influence of different particle matching techniques on the performance of the algorithm is analyzed and situations where the method proposed fails are revealed. In particular, the particle detection algorithm is evaluated separately from the whole μ PTV algorithm.

Two datasets describing planar velocity fields have been used to evaluate the method introduced in this thesis. The first one consists of 5 experimental μ PTV image pairs in a 50% solution of the LAYER-1 testing liquid with water, containing a total of 693 particles (344 linked pairs, 5 single particles). In this case, the ground truth data was manually labelled. The second dataset involves 5 synthetic image pairs containing a total of 662 randomly situated reference particles (323 linked pairs, 16 single particles), superimposed with Gaussian white noise.

The algorithms in this work were implemented in MATLAB R2010a, making use of both Image Processing Toolbox and Optimization Toolbox. The software was tested on a 64-bit Windows 7 operating system with an Intel Core 2 Duo T9400 processor (2.53 GHz) and 4 GB RAM. In order to reduce computation time, the code is vectorized where possible.

The next section outlines the experimental settings as well as the statistical methods applied in the evaluation. Section 4.2 and 4.3 in detail characterize both datasets and present the results obtained. In addition, the physical relevance of the extracted velocity data is discussed.

4.1 Experiments Overview

For image pre-processing, the AOS scheme of the CLMC equation is implemented (see Section 3.1.2.3). As motivated in Section 3.1.3, the parameter selection $\lambda = 8$, $\sigma = 2$, $T = 4$, $\tau = 2$ guarantees proper noise removal in a comparatively short computation time.

In the forthcoming performance analysis, first, the particle detection algorithm (in combination with matching) is analyzed at single-particle level. This means that all frames are regarded separately and the detection performance is evaluated by means of the particle layer indices allocated by the algorithm.

As a second step, the whole algorithm is evaluated, i.e. the particle pairs associated by the tracking algorithm are compared with labelled, correct particle links.

For these two purposes, the standard measures *precision* and *recall* [62] are applied. In this context, the number of *true positives* t_p is declared as the quantity of correctly identified particles (or particle links, respectively, for the evaluation of the whole algorithm). Erroneously detected particles (particle links) not listed in the ground truth are denoted as *false positives* f_p (Type I error), while the *false negatives* f_n are represented by those entries in the ground truth data which have not been discovered by the algorithm (Type II error). Those particles (particle links), where an inaccurate layer index has been assigned, therefore contribute to both false positives and false negatives.

Then, the precision p and the recall r are defined by

$$p = \frac{t_p}{t_p + f_p}, \quad r = \frac{t_p}{t_p + f_n}. \quad (4.1)$$

Another accuracy measure is given by the *F-score* F_β

$$F_\beta = \frac{(1 + \beta^2) \cdot p \cdot r}{\beta^2 \cdot p + r}, \quad (4.2)$$

which represents a weighted average of precision and recall. In our application, the correctness of the results is more important than the amount of extracted data per image. Therefore, we use $\beta = 0.5$, thus weighing the precision twice as much as the recall.

In the ground truth data, each particle is characterized by the coordinates of its center and its layer index. All entries for which a corresponding detected particle exists, are labelled true positives (of the particle detection algorithm). To be precise, two conditions must be fulfilled:

- the (Euclidian) distance between the detected and the true center coordinates must be small. Due to potential inaccuracies in the manually annotated ground truth data (especially for large, low-contrast ring-shaped particles, where the center location is not apparent), a maximum allowed deviation of 4 pixels is introduced. This corresponds to approximately 6.5 % of the diameter of the largest occurring reference particle (62 pixels).
- their layer indices must be equal. As will be motivated below, for an increased relevance of the results, this requirement is relaxed to a maximum allowed difference in the layer index of up to ± 3 .

In the case of multiple detected particles fulfilling both conditions, solely the one nearest to the true center is selected as the corresponding true positive. With the current data, however, this situation never occurred.

The influence of five different matching variants on the outcome of the algorithm is compared: the first one applies the radial intensity profile technique as described in Section 3.3.2.

As for the second method, further robustness towards the profile matching process is created by merging the most resembling reference images into one class according to a previously defined matching value threshold t . In the experiments below, each pair of reference images,

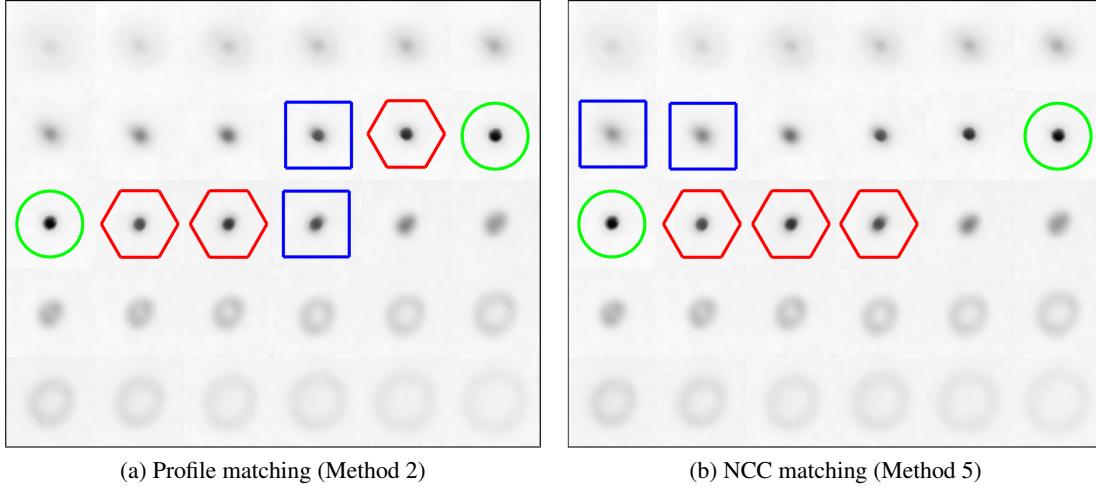


Figure 4.1: All 30 reference images with highlighted particles which are merged into three classes. Gray values inverted, contrast enhanced.

where the sum of absolute distances between their intensity profiles undercuts the value $t = 0.6$, is merged into the same layer class. Since the intensity profiles are evaluated on two grid points per pixel (cf. Section 3.3.2), the matching value 0.6 corresponds to an approximate area of $\frac{0.6}{2} = 0.3$ between both intensity profile curves. In Figure 4.1(a), the three particle classes emerging in the current application are marked by blue squares, green circles and red hexagons, respectively. In the further process, one representative particle per class is selected and the others are discarded, i.e. in the current situation, from 30 original reference images 26 distinct layer classes are remaining. Due to the reduced amount of layers, a locally decreased spatial depth-resolution of the flow is the consequence. Figure 4.1(a) discloses a potentially occurring problem of the introduction of layer classes for the practical application: the physical significance of the velocity data is diminished if the layer indices of the same class are not adjacent (here: blue squares and red hexagons). The selection of a stricter threshold t prevents this situation; nevertheless, incorrect matching results cannot be ruled out due to the remaining resemblance of the corresponding reference images.

The third method applies NCC matching with rectangular template windows with the aid of Fourier transforms according to the approach presented by Lewis [41], see Section 3.3.1.

Ultimately, Methods 4 and 5 compute the NCC directly in a circular template region, using a 5×5 neighborhood around the corresponding circle centers (see Section 3.3.1). Analogously to Method 2, Method 5 combines layers with similar particle shapes into one class. To be precise, a maximum NCC value of 0.99 is allowed for two reference images not to be merged into the same class. Figure 4.1(b) shows the resulting layer classes for this method.

Furthermore, the origins of the (Type I and Type II) errors are investigated more concretely. For both particle detection as well as the whole μ PTV process, six potential reasons for the emergence of false negatives can be identified while there are three reasons for the occurrence of false positives, a detailed description of which is provided in Tables 4.1-4.4.

Table 4.1: Reasons for the detection of false positives in the particle detection algorithm.

Reason	Description
1	wrong layer index assigned (too large difference from ground truth value)
2	particle would have been correctly detected, but center coordinates are inaccurate (more than 4 pixels, less than 8 pixels away)
3	none of the reasons above (e.g. spurious detection due to background)

Table 4.2: Reasons for the detection of false negatives in the particle detection algorithm.

Reason	Description
1	particle center not detected since no local maximum in the accumulator array was found (conditions (M1) and (M2) on page 50)
2	local maximum in the accumulator array found, but no corresponding particle radius (conditions (R1), (R2), (R3) on page 51)
3	particle radius found but variance condition along surrounding circle lines not fulfilled (condition (M3) on page 50)
4	particle radius found and variance condition fulfilled but particle erroneously dismissed due to poor matching value (threshold)
5	matching value accepted but wrong layer index assigned (too large difference from ground truth value)
6	particle would have been correctly detected, but center coordinates are inaccurate (more than 4 pixels, less than 8 pixels away)

Table 4.3: Reasons for the detection of false positives in the whole μ PTV algorithm.

Reason	Description
1	wrong layer index assigned (too large difference from ground truth value), error passed on from incorrect particle detection (Table 4.1 Reason 1)
2	particle link would have been correctly detected, but center coordinates of at least one particle were inaccurate (Table 4.1 Reason 2)
3	other erroneous links produced by tracking algorithm

Table 4.4: Reasons for the detection of false negatives in the whole μ PTV algorithm.

Reason	Description
1	particle link impossible to find; 0 particles detected
2	particle link impossible to find; 1 particle detected
3	particle link impossible to find; both particles detected but a wrong layer index was assigned to at least one particle (Table 4.2 Reason 5)
4	particle link impossible to find; both particles correctly detected (layer index ± 3) but no common layer index exists
5	particle link would have been correctly detected, but center coordinates of at least one particle were inaccurate (Table 4.2 Reason 6)
6	particle link possible to find; erroneous link produced by tracking algorithm

4.2 Experimental Dataset

For the experimental dataset, ground truth data is not a-priori available and therefore created through manual particle annotation in the original, non pre-processed raw images. As the flows considered in this work are assumed to be parallel to a backplate, planar velocity vectors are postulated, cf. (3.51). This physical assumption leads to an increased reliability in the particle labelling process as it allows the simultaneous classification of the same particle in both frames with varying contrast conditions. Nevertheless, uncertainty in the specified ground truth data remains: first of all, the finite number of reference images accounts for new, “in-between” types of occurring particle shapes. Moreover, the similarity of specific reference images of different layers (cf. Figure 4.1: reference particles around and behind the focal plane) as well as image noise impede the accurate particle classification. This statement is confirmed by the fact that the manual annotation of an image from the synthetic dataset, where ground truth data is available, resulted in solely 71% exactly matched particles whereas 29% of the associations were off by one layer. As a result, in the case of the experimental image data, it is not meaningful to demand the exact recognition of the manually characterized ground truth reference label. Instead, a deviation of ± 2 or ± 3 layers, respectively, is accepted and considered as a true positive.

Tables 4.5 and 4.6 display the performance of our profile matching algorithm in contrast to the cross-correlation based approaches outlined in Section 3.3 for the experimental dataset. Precision, recall and $F_{0.5}$ -score of the particle detection algorithm as well as the whole μ PTV algorithm are listed for a maximum allowed matching difference of 0, 1, 2 and 3 layer indices, respectively. Note that for the evaluation of the detection algorithm, the layer index of a particle is computed based on the best matching value (highest NCC score or lowest profile dissimilarity score, respectively). Concerning a particle link in the whole μ PTV algorithm, this layer index is recomputed (a common layer index for both particles is found, see Section 3.4.3) and therefore possibly differs from the previous value. Furthermore, the average computation time of the whole algorithm per image pair is shown. The most relevant results (profile matching vs. NCC matching, allowed deviation ± 2 layers, no layer classes) are highlighted in bold.

Table 4.5: Results of particle detection algorithm on real data, ground truth manually annotated

Matching method	Precision				Recall			
	± 0	± 1	± 2	± 3	± 0	± 1	± 2	± 3
1: Profiles	0.445	0.831	0.924	0.945	0.407	0.760	0.846	0.864
2: Profiles, merged layers	0.478	0.872	0.957	0.971	0.433	0.789	0.866	0.879
3: NCC Lewis	0.401	0.803	0.862	0.900	0.323	0.648	0.696	0.726
4: NCC mask	0.397	0.792	0.868	0.904	0.368	0.734	0.805	0.838
5: NCC mask, merged layers	0.498	0.865	0.908	0.931	0.459	0.798	0.837	0.859

Matching method	$F_{0.5}$ -score			
	± 0	± 1	± 2	± 3
1: Profiles	0.437	0.816	0.907	0.928
2: Profiles, merged layers	0.469	0.854	0.937	0.951
3: NCC Lewis	0.382	0.766	0.823	0.859
4: NCC mask	0.391	0.779	0.855	0.890
5: NCC mask, merged layers	0.489	0.851	0.893	0.916

Table 4.6: Results of whole μ PTV algorithm on real data, ground truth manually annotated

Matching method	Precision				Recall			
	± 0	± 1	± 2	± 3	± 0	± 1	± 2	± 3
1: Profiles	0.466	0.844	0.932	0.949	0.398	0.721	0.797	0.811
2: Profiles, merged layers	0.495	0.871	0.962	0.972	0.413	0.727	0.802	0.811
3: NCC Lewis	0.432	0.803	0.875	0.905	0.331	0.616	0.672	0.695
4: NCC mask	0.431	0.803	0.875	0.908	0.381	0.709	0.773	0.802
5: NCC mask, merged layers	0.523	0.877	0.924	0.937	0.459	0.770	0.811	0.823

Matching method	$F_{0.5}$ -score				Processing Time
	± 0	± 1	± 2	± 3	
1: Profiles	0.451	0.816	0.901	0.918	39.6 s
2: Profiles, merged layers	0.476	0.838	0.925	0.935	39.1 s
3: NCC Lewis	0.407	0.757	0.825	0.854	101.5 s
4: NCC mask	0.420	0.782	0.853	0.885	91.6 s
5: NCC mask, merged layers	0.509	0.854	0.899	0.912	86.3 s

It is concluded that the proposed approach using radial intensity profiles is superior to the NCC-based techniques reviewed, with respect to the evaluation measures examined as well as the temporal performance. The merging of layers into classes results in an increase of the precision of approximately 3 – 5 %.

Unfortunately, a direct comparison to existing literature where a related Defocusing μ PTV framework is addressed [54,56,86] is not possible since in those publications, image processing components are either analyzed insufficiently or not at all. Solely Paschke et al. [56], who implemented a cross-correlation based technique, report a maximum recall value of approximately 0.2. Table 4.6 indicates that in this regard, the algorithm presented achieves superior results. However, note that the conditions are related but not identical as the actual performance of the method is influenced by different factors such as the given particle density in the considered recordings, the number of reference particles or the present contrast conditions attributable to the preferences of the optical system in use.

Figures 4.2 and 4.3 display the distribution of false positives and false negatives, respectively, according to the error classification in Tables 4.1-4.4. The plots on the left refer to the standard profile matching algorithm (Method 1), the right side is dedicated to NCC matching (Method 4). In Figure 4.2, where the error statistics of the particle detection algorithm are depicted, the blue bars stand for the particles which are partially occluded. This concerns particles which are either clipped by the image border or part of a cluster, i.e. where other particles are located in close proximity.

In case of the **particle detection algorithm**, we observe that

- the distribution of false positive reasons is similar for both matching techniques: over 80 % of Type I errors originate in erroneous matching results (f_p Reason 1), with an insignificant negative influence of particle clusters.
- the majority of the false negatives are also attributable to poor matching results (f_n Reasons 4 and 5). Detection errors due to the modified CHT (f_n Reasons 1,2,3) are rare and primarily result from the indistinguishability of optically coalesced particle clusters. To be precise, the main reason for a false negative in profile matching is the erroneous rejection due to the matching threshold selected. Regarding NCC matching, however, detected particles with an incorrectly assigned layer index are the dominant factor. Note that this result is dependent on the explicit selection of the parameters appearing in the implementation. A “loosened” matching threshold reduces the number of incorrectly rejected particles, but also causes a higher number of wrongly allocated layer indices. In the notation from above, this would induce a decline of f_n Reason 4 errors, combined with an increase in errors due to f_n Reason 5 and f_p Reason 1.

As for the **whole μ PTV algorithm**,

- as a logical consequence of the results from above, again the vast majority (this time over 90 % for both matching techniques) of false positives is due to wrong layer indices.
- more than half of the false negative links are due to the unsuccessful detection of the corresponding particle in both frames.

- all particle links, where the corresponding particle was successfully detected in both frames, are recovered by the tracking algorithm, as the frequency of fn Reason 6 is zero (for both matching techniques).

Figures 4.4-4.6 show real image cut-outs containing examples for the different error scenarios listed in Tables 4.1-4.4. The colors of the circles highlighting the particles indicate the detection status (green = successful, yellow = wrong layer, red = unsuccessful). The different shades of green indicate the difference to the layer index listed in the ground truth (bright $\leftrightarrow \pm 0$, darker $\leftrightarrow \pm 1$, dark $\leftrightarrow \pm 2$). Analogously, the color of the arrows describes the tracking status of the whole μPTV algorithm. Further details are provided in the respective image captions.

To conclude this section, the physical relevance of the results obtained is evaluated. The theoretical velocity profile of a falling film on a flat, vertical plate was specified in the previous chapter in Equation (3.51) on page 64, see also Figure 3.16. The currently considered measurements, however, were conducted on a structured plate with engraved microchannels; a sketch is provided in Figure 4.7(a). Due to the inhomogeneous thickness of the flow, it is obvious that the resulting particle velocities cannot be expected to follow a clear profile. Nevertheless, the hypothetical velocity profiles with flow thickness d_1 (in a channel) and d_2 (on a ridge), as indicated in Figure 4.7(a), can be used as reference curves for the extracted data. Note that the exact values inserted for d_1 , d_2 and the kinematic viscosity of the liquid ν were estimated based on the velocity data. This data is visualized in Figure 4.7(b) with the aid of a violin plot, which generates multiple histograms side-by-side. In this way, the particle velocity distributions for all flow layers are shown in a well-organized manner. For the creation of the graphic, a toolbox downloaded from the MATLAB Central File Exchange was used [32]. For each layer, a red cross marks the mean particle velocity. It can be seen that the particle velocities (except for a few errors) are limited by the hypothetical profiles and their distribution shows good agreement with the parabolic profile shape, especially for the deeper and shallower regions of the flow. The high variation of the particle velocities of the layers in the middle of the flow (near to the ridges in the plate) is due to the high variation of possible particle distances to the nearest plate surface where the velocity is assumed to be zero (no-slip boundary condition).

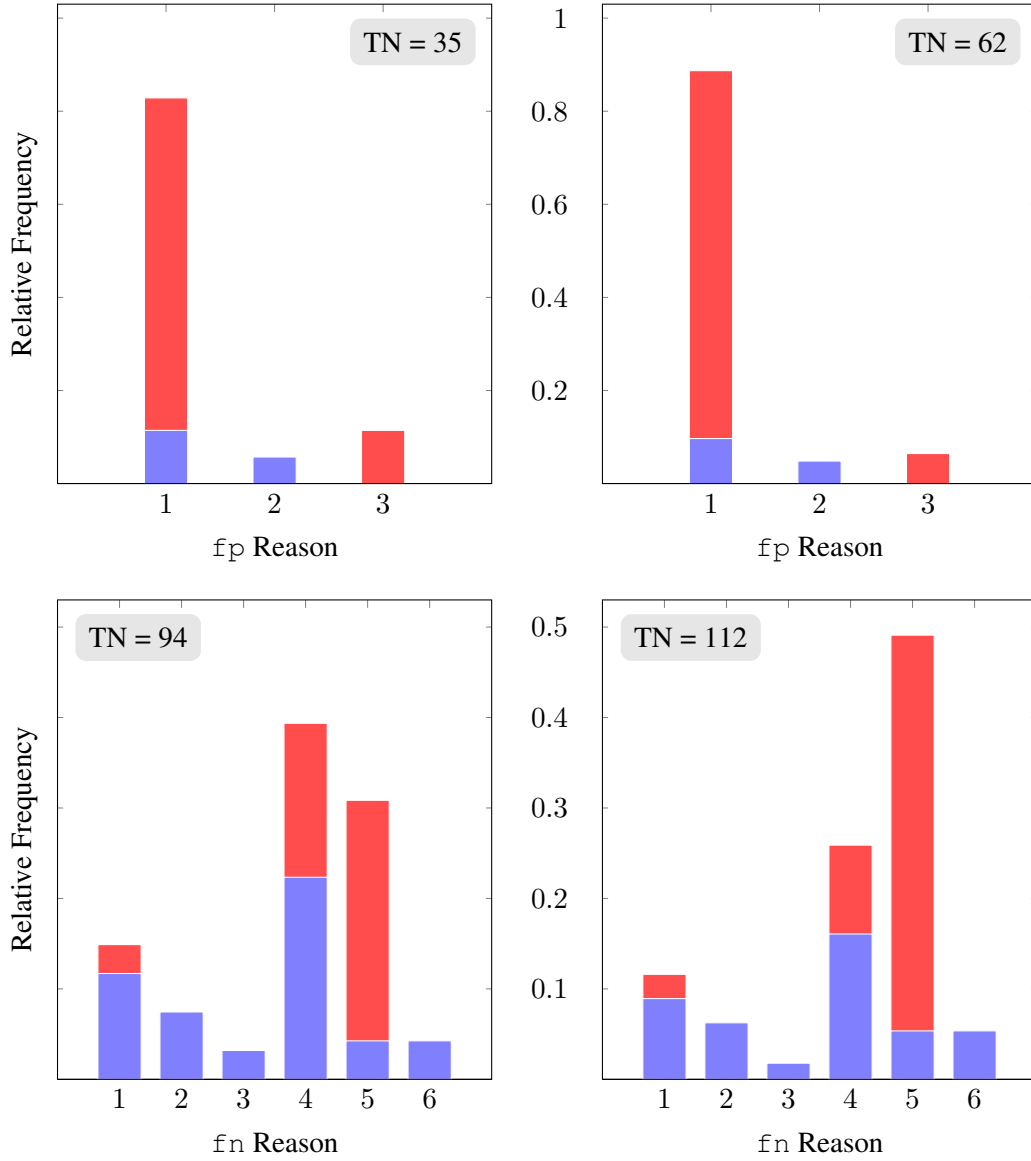


Figure 4.2: False positive and false negative error distribution of **particle detection algorithm** according to reasons listed in Tables 4.1 and 4.2. TN = Total Number which relative frequency refers to. Left: profile matching (Method 1), Right: NCC matching (Method 4). Above: False positives fp (see Table 4.1). Below: False negatives fn (see Table 4.2). The blue coloring indicates the fraction of particles which are not entirely visible, either due to particle clustering or clipping by the image border.

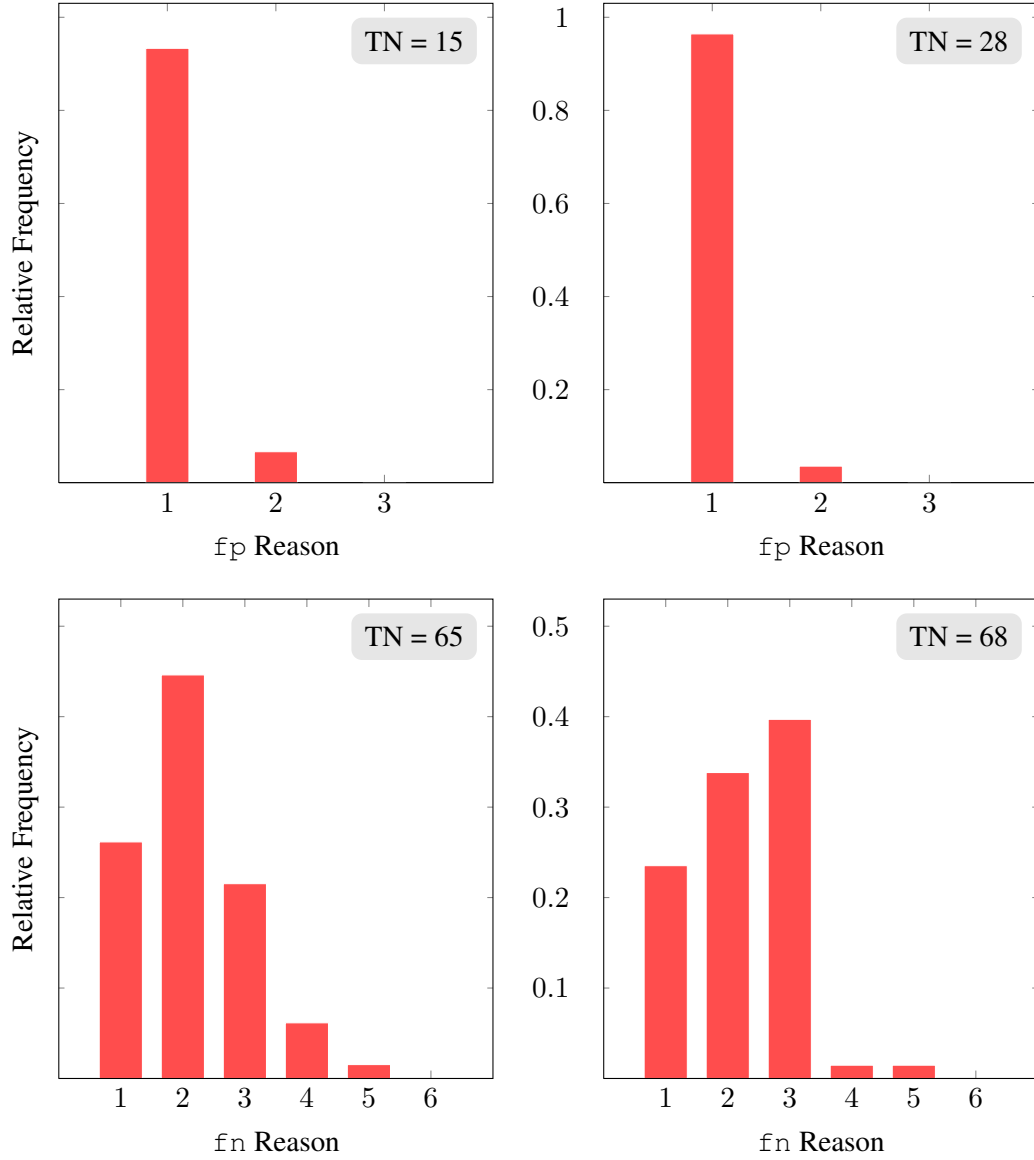
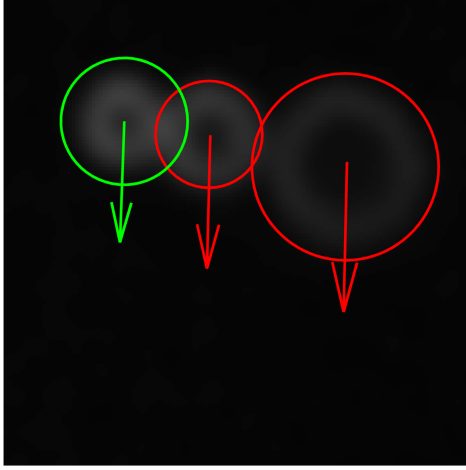
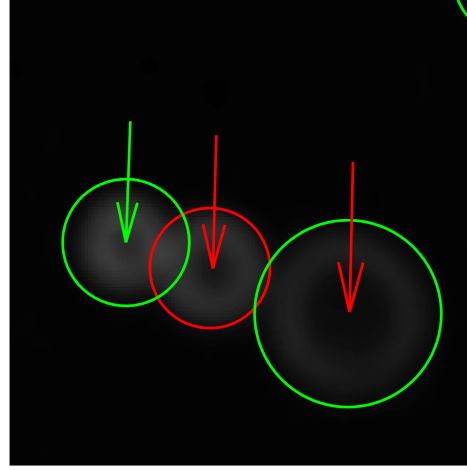


Figure 4.3: False positive and false negative error distribution of **whole μ PTV algorithm** according to reasons listed in Tables 4.3 and 4.4. TN = Total Number which relative frequency refers to. Left: profile matching (Method 1), Right: NCC matching (Method 4). Above: False positives fp (see Table 4.3). Below: False negatives fn (see Table 4.4).



(a) Detection error scenario 1a: the left particle is correctly identified (layer index ± 0), the other two particles are detected by the CHT but discarded since no matching value (profile matching) undercuts the matching threshold (Table 4.2 Reason 4).

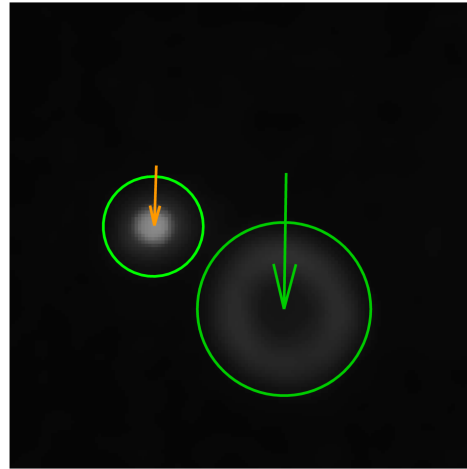


(b) Detection error scenario 1b: the left and right particles are correctly identified (layer index ± 0), the particle in the middle is detected by the CHT but discarded since no matching value (profile matching) undercuts the matching threshold (Table 4.2 Reason 4).

Figure 4.4: Tracking error scenario 1: the left particle is correctly tracked (common layer index ± 0). The particle in the middle cannot be tracked since it was not correctly detected in both frames (Table 4.4 Reason 1). The right particle cannot be tracked since it was only correctly detected in one frame (Table 4.4 Reason 2).

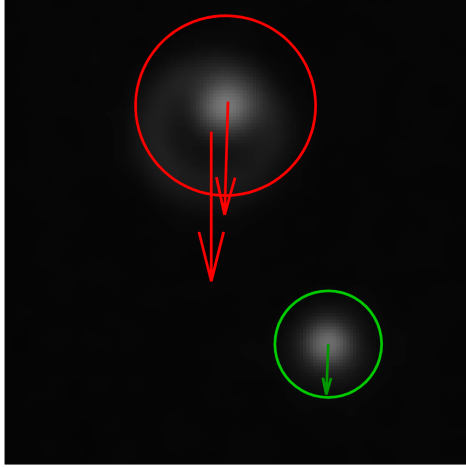


(a) Detection error scenario 2a: the left particle is detected but assigned to a wrong layer (Table 4.1 Reason 1 and Table 4.2 Reason 5). The right particle is correctly identified (layer index ± 1).

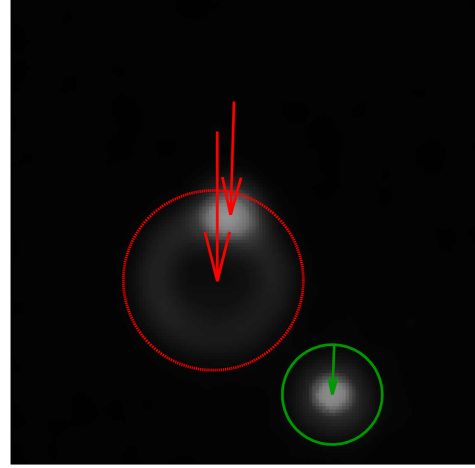


(b) Detection error scenario 2b: here, the left particle is correctly identified (layer index ± 0) as well as the right particle (layer index ± 1).

Figure 4.5: Tracking error scenario 2: the left particle is tracked in a wrong common layer (Table 4.3 Reason 1 and Table 4.4 Reason 3). Observe the difference in the particle shapes in both frames, which impedes the discovery of a meaningful common layer index! The right particle is correctly tracked (common layer index ± 1).

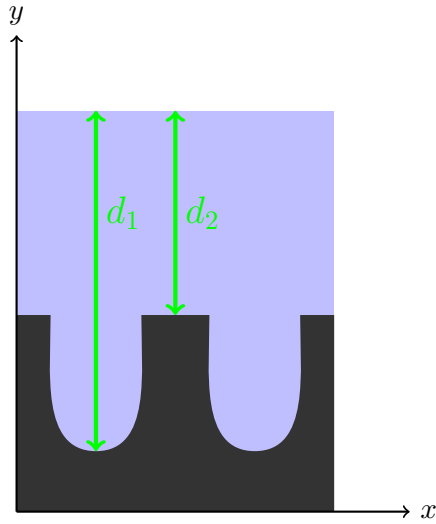


(a) Detection error scenario 3a: in the top half of the image, a ring-shaped particle coalesces with a bright particle. The ring particle is not detected at all (Table 4.2 Reason 1) while the sharp particle is associated with a too large radius. In the particle matching step, this spurious detection is discarded (Table 4.2 Reason 4). The third particle in the scene is correctly identified (layer index ± 1).

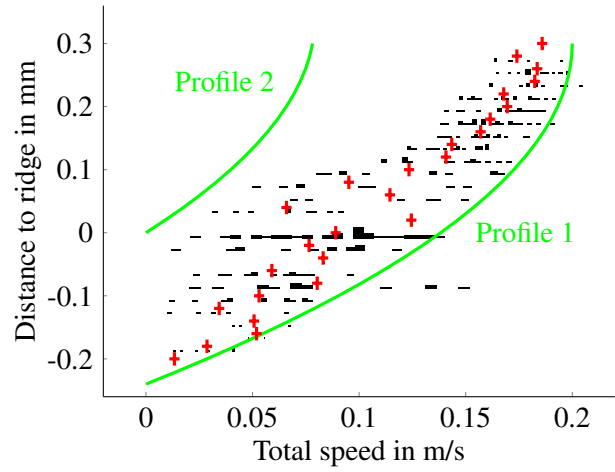


(b) Detection error scenario 3b: in this frame, all three particle centers are detected as local maxima in the accumulation array. However, the ring particle does not fulfil the variance condition (Table 4.2 Reason 3) and the bright particle in the cluster is discarded as a result of the radius condition (R3), Table 4.2 Reason 2. The third particle is correctly identified (layer index ± 2).

Figure 4.6: Tracking error scenario 3: both particles in the cluster are not tracked since they have not been successfully detected on both frames (Table 4.4 Reason 1). The third particle link is correctly recognized (common layer index ± 2).



(a) Cross-section segment of backplate used in experiments.



(b) Violin plot of the absolute particle velocities extracted from the experimental dataset.

Figure 4.7: Physical relevance of the results.

In addition, a different, larger dataset of 1142 image pairs containing a total of 71158 particles is investigated for a more meaningful description of the present flow behavior. Figure 4.8 displays the average vertical (x -component) flow velocity u_x , spatially resolved on a horizontal cross-section across the (y, z) -plane. The extracted velocity data indicates the actual geometry of the backplate and we observe that the actual depth of the microchannels located on the approximately 1.8×1.8 mm large region recorded by the camera is varying.

As a consequence of the no-slip boundary condition, it is expected that with increasing distance to the plate surface, the flow velocity also increases. This is consistent with the results shown in Figure 4.8 where the deeper microchannel (on the right-hand side of the graphic) accounts for higher flow velocities near to the flow surface (i.e. where $zn^{-1} \approx 0.2$) than the shallower microchannel (in the middle of the graphic). Here, zn^{-1} represents the *optical* distance to focus as n is the (unknown) refractive index of the liquid.

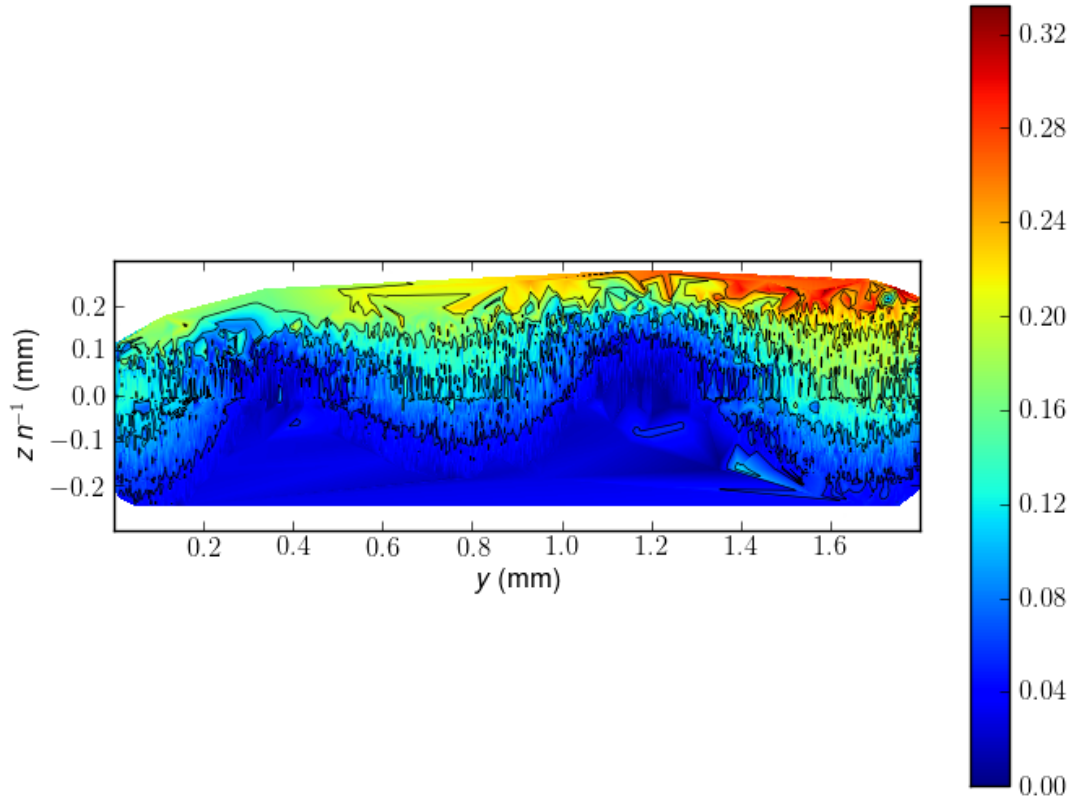


Figure 4.8: Average vertical flow velocity u_x in m/s, spatially resolved in (y, z) -plane where zn^{-1} denotes the optical distance to the focal plane (sample of over 70000 particles).

4.3 Synthetic Dataset

In the following, the μ PTV software developed is tested on synthetic raw data. In order to achieve meaningful comparability to the results presented in the previous section, the image structure should be as similar as possible to that of an experimental image. For this purpose, initially, the creation process of the synthetic raw image data in use is outlined.

1. **Random particle positioning:** In the end, 2048×2048 pixel images are desired. Since particle motion into/out of the frame should be possible, a 200 pixels wide “frame” is added outside the image area prior to particle positioning. The row and column coordinates, respectively, of the particle centers of the first frame are then calculated by generating 100 uniformly distributed random integers in the range $[-199, 2248]$. Moreover, for each coordinate, a random layer index between 1 and 30 is assigned. For convenience, a parabolic velocity profile (depending on the layer index) is postulated. As for the particle’s position in the second frame, the corresponding displacement vector from the velocity profile as well as a small random deviation in both directions (Gaussian distributed, $\mu = 0$, $\sigma = 5$ pixels) are added to the coordinates of the first frame.
2. **Determination of noise model:** As the particles inserted into the synthetic image are (circular) cut-outs from reference recordings, the noise distribution in the particle image should be consistent with the noise distribution from the reference recordings. To this end, an empty image region (i.e. only background, no particles) is cut out from all 30 reference recordings. The distribution of the corresponding total gray value data, displayed in Figure 4.9, suggests the application of a Gaussian distribution. In order to identify its parameters, the maximum likelihood estimates are computed ($\mu_0 = 55.9$, $\sigma_0 = 3.9$, scaled density function plotted in red).
3. **Creation of images:** Each image is initialized with constant gray value μ_0 . Subsequently, the cut-out particle images are plugged in at the calculated positions. For a realistic display of particle clusters, the pixel brightness at the intersection of p particles is selected as the maximum of all p potential gray values. Finally, additive white Gaussian noise according to the noise model ($\mu = 0$, $\sigma = \sigma_0$) is added and the images are clipped to the range $[1, 2048] \times [1, 2048]$.

Analogous to the experimental dataset, the particle detection and the whole μ PTV algorithm are evaluated separately. A first glance at the results presented in Tables 4.7 and 4.8 causes astonishment: firstly, the precision values of exact particle detection and tracking (layer index ± 0) are “only” in the range between 0.8 and 0.85 although the ground truth labels are accurate with certainty. Secondly, the combination of the most resembling layers into one class does not conspicuously improve the quality of the algorithm, in some cases precision/recall even deteriorate.

The distribution of the detected particles, where the difference to the correct layer index does not equal zero, provides the answer to both phenomena: 68.1 % belong to Layer 1-8, only 10.3 % are part of Layer 9-28 (where the most resembling reference particles are located, see Figure 4.1) and 21.6 % are counted among Layer 29-30. Therefore, for the synthetic dataset,

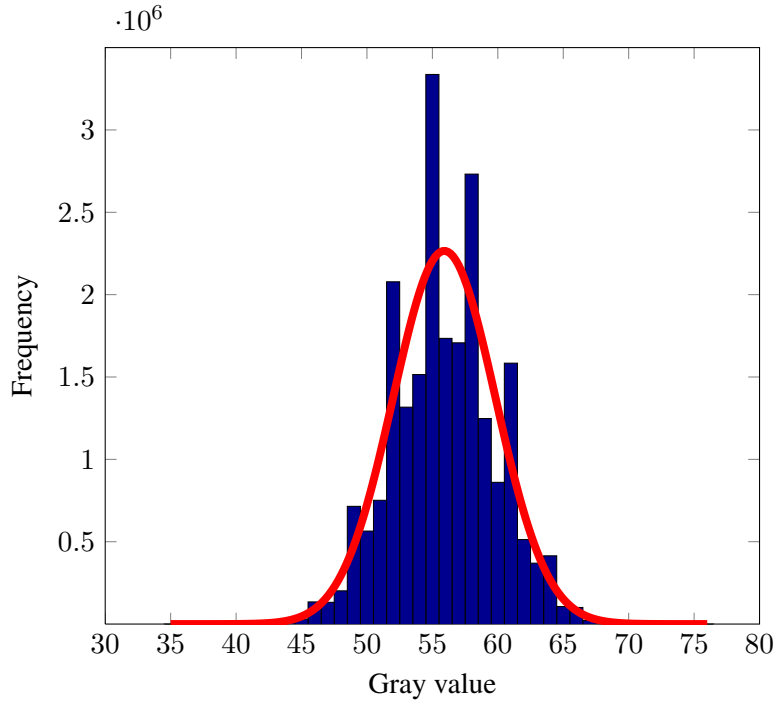


Figure 4.9: Gray value histogram of image background area from 30 reference recordings. The red line indicates the corresponding Gaussian distribution.

primarily the low-contrast particles are affected by imprecise matching. The reason for this is the increasing perturbing influence of the superimposed additive Gaussian noise with decreasing contrast of the particle image. In other words, the artificially added noise in some cases causes the unintended optical “transition” of a particle into a different layer. The impact of this effect on the particle velocity distribution, however, is marginal since a precision value of over 0.96 is reached for an allowed deviation of the layer index of ± 1 , regardless of the matching technique selected.

In general, the profile matching technique again is (slightly) more precise than the NCC-based methods; as a matter of fact, no false positive particle links are encountered for an allowed variation of the layer index of ± 2 . The error distribution analysis (not shown as a graphic) reveals that over 95 % of the false negatives in particle detection are due to the deficient separation of particle clusters, for both profile matching and NCC matching. Furthermore, the unsuccessful detection of both particles is the exclusive reason for a false negative in particle tracking. In return, each false positive particle detection arises from a particle clipped on the image border, where the particle’s center is not located in the image range and therefore not listed in the ground truth data.

Table 4.7: Results of particle detection algorithm on synthetic data, ground truth available.

Matching method	Precision				Recall			
	± 0	± 1	± 2	± 3	± 0	± 1	± 2	± 3
1: Profiles	0.811	0.972	0.989	0.989	0.796	0.955	0.971	0.971
2: Profiles, merged layers	0.816	0.974	0.989	0.989	0.799	0.953	0.968	0.968
3: NCC Lewis	0.846	0.967	0.983	0.985	0.769	0.879	0.894	0.896
4: NCC mask	0.844	0.969	0.985	0.985	0.832	0.956	0.971	0.971
5: NCC mask, merged layers	0.836	0.969	0.986	0.986	0.822	0.953	0.970	0.970

Matching method	$F_{0.5}$ -score			
	± 0	± 1	± 2	± 3
1: Profiles	0.808	0.969	0.986	0.986
2: Profiles, merged layers	0.813	0.970	0.985	0.985
3: NCC Lewis	0.829	0.948	0.964	0.966
4: NCC mask	0.841	0.967	0.982	0.982
5: NCC mask, merged layers	0.833	0.966	0.983	0.983

Table 4.8: Results of whole μ PTV algorithm on synthetic data, ground truth available.

Matching method	Precision				Recall			
	± 0	± 1	± 2	± 3	± 0	± 1	± 2	± 3
1: Profiles	0.833	0.973	1	1	0.774	0.904	0.929	0.929
2: Profiles, merged layers	0.836	0.977	1	1	0.771	0.901	0.923	0.923
3: NCC Lewis	0.851	0.967	0.993	0.993	0.728	0.827	0.848	0.848
4: NCC mask	0.851	0.967	0.990	0.990	0.799	0.907	0.929	0.929
5: NCC mask, merged layers	0.848	0.964	0.990	0.990	0.796	0.904	0.929	0.929

Matching method	$F_{0.5}$ -score				Processing Time
	± 0	± 1	± 2	± 3	
1: Profiles	0.821	0.959	0.985	0.985	37.1s
2: Profiles, merged layers	0.822	0.960	0.983	0.983	36.8s
3: NCC Lewis	0.823	0.936	0.960	0.960	89.6s
4: NCC mask	0.840	0.954	0.977	0.977	81.5s
5: NCC mask, merged layers	0.837	0.951	0.977	0.977	75.5s

4.4 Summary

In this chapter, the evaluation of the method introduced in this thesis was presented. First, an overview of the experiments was given and the statistical measures employed to indicate the performance of the methods evaluated were described. The metrics used were precision, recall and $F_{0.5}$ -score.

A classification system was introduced which distinguishes between Type I and Type II errors occurring in the particle detection process from Type I and Type II errors emerging in the particle tracking process. Graphical examples which illustrate the error classification were provided.

Subsequently, the datasets were characterized and the corresponding results were given. The main outcome is that the proposed technique based on radial intensity profiles not only provides the expected improvement concerning temporal performance, but also yields a superior reliability in the matching process compared to NCC algorithms. The image processing is performed more than two times faster with our suggested algorithm.

In addition, the physical relevance of the extracted velocity data was investigated.

Conclusion and Outlook

In this thesis, image processing algorithms customized for the implementation in a Single Aperture Defocusing μ PTV application are presented. Their task is to localize (particle detection), classify (particle matching) and associate (particle tracking) corresponding particles in paired image recordings in order to determine the flow velocity field of thin falling films.

Due to camera noise in the μ PTV recordings, the implementation of an image pre-processing routine is required. Owing to its property to smooth noise while largely preserving image edges, the CLMC filter, a nonlinear diffusion filter, represents the method of choice. The procedure involves the numerical solution of a nonlinear partial differential equation and two discretization schemes are discussed. In the end, the so-called *AOS scheme* is preferred to the *explicit scheme* due to its significantly shorter computation time at the cost of an acceptable loss of accuracy.

The core of the particle detection algorithm implemented is a modified Circular Hough Transform which operates on a binary edge image and exploits the circular symmetry of the particles. The fundamental modifications introduced include a technique which reduces computation time by eliminating disturbing edges prior to the application of the CHT. In addition, the application of certain voting weights for the CHT and a radius determination method specifically developed for μ PTV image data are proposed. As Hough Transforms in general are known to be robust algorithms, moderate imperfections in the particle shape caused by optical aberrations do not impede a successful detection. Moreover, in contrast to elementary detection methods, the CHT allows the correct dissection of clusters of overlapping particles instead of a-priori rejecting them as non-circular objects. In practice, however, the number of further processed particles is reduced since a relevant matching result can solely be ensured for merely partially (50 % at most) occluded particles. Therefore, the exact evaluation results are depending on the particle density and the actual spatial particle distribution in the considered image dataset.

In the majority of related literature, the appearing particle shapes allow a classification based on the outer visible radius of a ring structure. However, due to the different setup used in the current approach, which captures a wider region of the investigated flow, other distinct particle shapes emerge such that a different matching technique is required. For this purpose, a simple

method based on radial intensity profiles is proposed, which is proven to reach a higher accuracy and is more than twice as fast as classical cross-correlation methods.

The tracking algorithm implemented is formulated as a minimization problem which is solved efficiently for the data evaluated in this work. The physical assumption of a planar flow is included as a condition which increases the reliability of the results for the considered falling films.

The evaluation is performed on two different datasets: for the *experimental dataset* consisting of real μ PTV recordings, ground truth data is manually annotated while the *synthetic dataset* is automatically generated by means of randomly positioned reference particles on empty images which are subsequently disturbed by additive white Gaussian noise. Both datasets use a total of 30 different classes of matched reference particles.

The whole approach is shown to yield promising results on both datasets and for the experimental dataset, the physical validity of the results is also ensured.

Future Work

A detailed error analysis reveals that incorrect particle matching causes the majority of incorrect detections. One reason for this is that the matching thresholds, which determine if a particle detection is discarded or assigned to a specific layer, are sensitive parameters and both a too high or a too low selection have negative influence on the performance of the whole approach. Therefore, future work could deal with the (optimized) automatization of the parameter selection process, which also would enhance the usability of the software.

Moreover, from a physical point of view, the relevance of the results depends on the refraction properties of the liquid. To this end, the thickness of the drop in the reference image measurement (cf. Section 1.1.1) or potential fluctuations in the film thickness influence the optical appearance of the recorded particles and must be taken into account if the scale along the optical axis is to be determined exactly.

Bibliography

- [1] U. Adiga et al., *A binary segmentation approach for boxing ribosome particles in cryo EM micrographs*, Journal of Structural Biology **145** (2004), 142–151.
- [2] U. Adiga et al., *Particle picking by segmentation: A comparative study with SPIDER-based manual particle picking*, Journal of Structural Biology **152** (2005), 211–220.
- [3] R. J. Adrian, *Twenty years of particle image velocimetry*, Experiments in Fluids **39** (2005), 159–169.
- [4] R. J. Adrian and J. Westerweel, *Particle Image Velocimetry*, vol. 30, Cambridge University Press, 2011.
- [5] A. S. Aguado et al., *Bias Error Analysis of the Generalised Hough Transform*, Journal of Mathematical Imaging and Vision **12** (2000), 25–42.
- [6] F. Al-Sibai, *Experimentelle Untersuchung der Strömungscharakteristik und des Wärmeübergangs bei welligen Rieselfilmen*, Ph.D. thesis, RWTH Aachen University, 2004.
- [7] T. J. Atherton and D. J. Kerbyson, *Using phase to represent radius in the coherent circle Hough transform*, IEE Colloquium on the Hough Transform (London), vol. 5, 1993, pp. 1–4.
- [8] T. J. Atherton and D. J. Kerbyson, *Size invariant circle detection*, Image and Vision Computing **17** (1999), no. 11, 795–803.
- [9] G. Aubert and P. Kornprobst, *Mathematical problems in image processing: partial differential equations and the calculus of variations*, vol. 147, Springer Science+ Business Media, 2006.
- [10] S. J. Baek and S. J. Lee, *A new two-frame particle tracking algorithm using match probability*, Experiments in Fluids **22** (1996), 23–32.
- [11] D. H. Ballard, *Generalizing the Hough Transform to Detect Arbitrary Shapes*, Pattern Recognition **13** (1981), no. 2, 111–122.
- [12] K. Briechle and U. D. Hanebeck, *Template matching using fast normalized cross correlation*, Proceedings of SPIE: Optical Pattern Recognition XII, vol. 4387, 2001, pp. 95–102.

- [13] J. Canny, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986), no. 6, 679–698.
- [14] F. Catté et al., *Image selective smoothing and edge detection by nonlinear diffusion*, SIAM Journal on Numerical Analysis **29** (1992), 182–193.
- [15] J. Cauchie et al., *Optimization of an Hough transform algorithm for the search of a center*, Pattern Recognition **41** (2008), 567–574.
- [16] S.-M. Chao and D.-M. Tsai, *Anisotropic diffusion with generalized diffusion coefficient function for defect detection in low-contrast surface images*, Pattern Recognition **43** (2010), no. 5, 1917–1931.
- [17] K.-L. Chung et al., *Efficient sampling strategy and refinement strategy for randomized circle detection*, Pattern Recognition **1** (2012), 252–263.
- [18] G.-H. Cottet and L. Germain, *Image processing through reaction combined with nonlinear diffusion*, Mathematics of Computation **61** (1993), no. 204, 659–673.
- [19] F. D’Almeida, *Nonlinear Diffusion Toolbox*, MATLAB Central File Exchange (<http://www.mathworks.com/matlabcentral/fileexchange/3710-nonlinear-diffusion-toolbox>), 2003, Retrieved March 19, 2013.
- [20] M. Dorigo et al., *The ant system: optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **26** (1996), no. 2, 29–41.
- [21] J. M. Duarte-Carvajalino et al., *Comparative study of semi-implicit schemes for nonlinear diffusion in hyperspectral imagery*, IEEE Transactions on Image Processing **16** (2007), no. 5, 1303–1314.
- [22] R. O. Duda and P. E. Hart, *Use of the Hough transform to detect lines and curves in pictures*, Communications of the ACM **15** (1972), no. 1, 11–15.
- [23] R. Duits et al., *On the Axioms of Scale Space Theory*, Journal of Mathematical Imaging and Vision **20** (2004), 267–298.
- [24] L. C. Evans, *Partial Differential Equations*, vol. 19, American Mathematical Society, 1998.
- [25] I. Grant and X. Pan, *An investigation of the performance of multi-layer neural networks applied to the analysis of PIV images*, Experiments in Fluids **19** (1995), no. 3, 159–166.
- [26] Y. Hassan and R. Canaan, *Full-field bubbly flow velocity measurements using a multiframe particle tracking technique*, Experiments in Fluids **12** (1991), 49–60.
- [27] P. V. C. Hough, *Method and means for recognizing complex patterns*, U.S. Patent 3069654, 1962.

- [28] J. Illingworth and J. Kittler, *The adaptive Hough transform*, IEEE Transactions on Pattern Analysis and Machine Intelligence **9** (1987), no. 5, 690–698.
- [29] J. Illingworth and J. Kittler, *A Survey of the Hough Transform*, Computer Vision, Graphics and Image Processing **44** (1988), 87–116.
- [30] K. Ishizu et al., *Image Processing of Particle Detection for Asbestos Qualitative Analysis Support Method*, 10th International Conference on Control, Automation, Robotics and Vision, Hanoi, Vietnam, 17-20 December 2008, IEEE, 2008, pp. 868–873.
- [31] P. Jia et al., *Improvement in the independence of relaxation method-based particle tracking velocimetry*, Measurement Science and Technology **24** (2013), no. 5, 055301.1–055301.13.
- [32] Jonas, *Violin Plots for plotting multiple distributions*, MATLAB Central File Exchange (<http://www.mathworks.com/matlabcentral/fileexchange/23661-violin-plots-for-plotting-multiple-distributions-distributionplot-m>), 2009, Retrieved March 19, 2013.
- [33] P. Joubert et al., *Automatic particle picking using diffusion filtering and random forest classification*, Proceedings of 6th Workshop on Microscopic Image Analysis with Applications in Biology, Heidelberg, Germany, 2011.
- [34] H. Kälviäinen et al., *Probabilistic and non-probabilistic Hough transforms: overview and comparisons*, Image and Vision Computing **13** (1995), no. 4, 239–252.
- [35] A. A. Kassim et al., *A comparative study of efficient generalised Hough transform techniques*, Image and Vision Computing **17** (1999), 737–748.
- [36] M. Knaak et al., *A Hopfield neural network for flow field computation based on particle image velocimetry/particle tracking velocimetry image sequences*, Proceedings of the IEEE International Conference on Neural Networks, 1997, pp. 48–52.
- [37] B. Kolman and R. E. Beck, *Elementary Linear Programming with Applications*, Elsevier Science & Technology Books, 1995.
- [38] H. Kuba et al., *Automatic Particle Detection and Counting by One-Class SVM from Microscope Image*, Advances in Neuro-Information Processing (Berlin, Heidelberg) (Mario Köppen, Nikola Kasabov, and George Coghill, eds.), Springer, 2009, pp. 361–368.
- [39] G. Labonté, *On a neural network that performs an enhanced nearest-neighbour matching*, Pattern Analysis and Application **3** (2000), 267–278.
- [40] S. J. Lee and S. Kim, *Advanced particle-based velocimetry techniques for microscale flows*, Microfluid Nanofluid **6** (2009), 577–588.
- [41] J. P. Lewis, *Fast normalized cross-correlation*, Industrial Light and Magic, 1995.
- [42] F.-C. Li and K. Hishida, *Particle Image Velocimetry Techniques and its Applications in Multiphase Systems*, Advances in Chemical Engineering **37** (2009), 87–147.

- [43] T. Lindeberg, *Scale-Space*, Wiley Encyclopedia of Computer Science and Engineering (B. Wah, ed.), John Wiley & Sons, 2009, pp. 2495–2504.
- [44] S. Maaß et al., *Automated drop detection using image analysis for online particle size monitoring in multiphase systems*, Computers and Chemical Engineering **45** (2012), 27–37.
- [45] S. P. Mallick et al., *Detecting Particles in Cryo-EM Micrographs using Learned Features*, Journal of Structural Biology **145** (2004), 52–62.
- [46] J. Matoušek and B. Gärtner, *Understanding and Using Linear Programming*, Springer, 2007.
- [47] B. R. Munson et al., *Fundamentals of Fluid Mechanics*, 6 ed., John Wiley & Sons, 2010.
- [48] W. V. Nicholson and R. M. Glaeser, *Review: Automatic Particle Detection in Electron Microscopy*, Journal of Structural Biology **133** (2001), 90–101.
- [49] W. Nusselt, *Die Oberflächenkondensation des Wasserdampfes*, Z. VDI **60** (1916), 541–546.
- [50] K. Ohmi et al., *Particle tracking velocimetry with an ant colony optimization algorithm*, Experiments in Fluids **48** (2010), 589–605.
- [51] K. Ohmi and S. P. Panday, *Particle tracking velocimetry using the genetic algorithm*, Journal of Visualization **12** (2009), no. 3, 217–232.
- [52] N. T. Ouellette et al., *A quantitative study of three-dimensional Lagrangian particle tracking algorithms*, Experiments in Fluids **40** (2006), 301–313.
- [53] S. P. Panday et al., *An ant colony optimization based stereoscopic particle pairing algorithm for three-dimensional particle tracking velocimetry*, Flow Measurement and Instrumentation **22** (2011), 86–95.
- [54] J. S. Park and K. D. Kihm, *Three-dimensional micro-PTV using deconvolution microscopy*, Experiments in Fluids **40** (2006), 491–499.
- [55] J. R. Parker, *Algorithms for Image Processing and Computer Vision*, 2 ed., John Wiley & Sons, 2011.
- [56] S. Paschke, J.-U. Repke, and G. Wozny, *A New Approach for Measuring the Local 3D Velocity Field of Thin Film Flows With a Common μ PIV Application*, Chemie Ingenieur Technik **80** (2008), 1477–1485.
- [57] F. Pereira et al., *Two-frame 3D particle tracking*, Measurement Science and Technology **17** (2006), no. 7, 1680–1692.
- [58] P. Perona and J. Malik, *Scale-Space and Edge Detection using Anisotropic Diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990), no. 7, 629–639.

- [59] S. D. Peterson et al., *Three-dimensional particle tracking using micro-particle image velocimetry hardware*, Measurement Science and Technology **19** (2008), no. 11, 115406.1–115406.8.
- [60] J. Pospisil et al., *Falling film heat exchange and backsplash on horizontal tube bundles*, International Journal of Energy **3** (2009), 35–42.
- [61] M. Raffel et al., *Particle Image Velocimetry: A Practical Guide*, 2nd ed., Experimental Fluid Mechanics, Springer, 2007.
- [62] C. J. van Rijsbergen, *Information Retrieval*, 2nd ed., ch. Evaluation, pp. 112–140, Butterworth, 1979.
- [63] P. Ruhnu, *Variational Fluid Motion Estimation with Physical Priors*, Ph.D. thesis, University of Mannheim, 2006.
- [64] J. G. Santiago et al., *A micro particle image velocimetry system*, Experiments in Fluids **25** (1998), 316–319.
- [65] I. F. Sbalzarini and P. Koumoutsakos, *Feature point tracking and trajectory analysis for video imaging in cell biology*, Journal of Structural Biology **151** (2005), 182–195.
- [66] G. Schnabel and E. U. Schlünder, *Wärmeübergang von senkrechten Wänden an nicht-siedende und siedende Rieselfilme*, Verfahrenstechnik **14** (1980), no. 2, 79–83.
- [67] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, 1998.
- [68] M. Seiler, *Hyperbranched polymers: Phase behavior and new applications in the field of chemical engineering*, Fluid Phase Equilibria **241** (2006), no. 1-2, 155–174.
- [69] J. Sheng and H. Meng, *A genetic algorithm particle pairing technique for 3D velocity field extraction in holographic particle image velocimetry*, Experiments in Fluids **25** (1998), 461–473.
- [70] J. M. Short, *SLEUTH - a fast computer program for automatically detecting particles in electron microscope images*, Journal of Structural Biology **145** (2004), 100–110.
- [71] M. Smereka and I. Duleba, *Circular Object Detection Using a Modified Hough Transform*, Applied Mathematics and Computer Science **18** (2008), no. 1, 85–91.
- [72] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*, John Wiley & Sons, 2011.
- [73] M. Sonka et al., *Image processing, analysis and machine vision*, 3rd ed., Thomson, 2008.
- [74] C. O. S. Sorzano et al., *Automatic particle selection from electron micrographs using machine learning techniques*, Journal of Structural Biology **167** (2009), no. 3, 252–260.

- [75] T. Takagi, *Study on particle tracking velocimetry using ant colony optimization [in Japanese]*, Journal of Visualization Society of Japan **27** (2007), 89–90.
- [76] T. Uemura et al., *A high speed algorithm of image analysis for real time measurement of two-dimensional velocity distribution*, Flow Visualization ASME FED **85** (1989), 129–134.
- [77] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, 2 ed., Kluwer Academic Publishers, 2001.
- [78] P. Wasserscheid and M. Seiler, *Leveraging gigawatt potentials by smart heat-pump technologies using ionic liquids*, ChemSusChem **4** (2011), no. 4, 459–463.
- [79] J. Weickert, *A review of nonlinear diffusion filtering*, Scale-space theory in computer vision (B. ter Haar Romeny et al., ed.), Lecture Notes in Computer Science, vol. 1252, Springer, Berlin, 1997, pp. 3–28.
- [80] J. Weickert, *Anisotropic diffusion in image processing*, vol. 1, Teubner Stuttgart, 1998.
- [81] J. Weickert, *Coherence-Enhancing Diffusion Filtering*, International Journal of Computer Vision **31** (1999), 111–127.
- [82] J. Weickert, *Coherence-enhancing diffusion of colour images*, Image and Vision Computing **17** (1999), no. 3, 201–212.
- [83] J. Weickert et al., *Efficient and reliable schemes for nonlinear diffusion filtering*, IEEE Transactions on Image Processing **7** (1998), no. 3, 398–410.
- [84] S. T. Wereley and C. D. Meinhart, *Recent Advances in Micro-Particle Image Velocimetry*, Annual Review of Fluid Mechanics **42** (2010), 557–576.
- [85] S. J. Williams et al., *Advances and applications on microfluidic velocimetry techniques*, Microfluid Nanofluid **8** (2010), 709–726.
- [86] M. Wu et al., *Three-dimensional fluorescent particle tracking at micron-scale using a single camera*, Experiments in Fluids **38** (2005), 461–465.
- [87] L. Xu et al., *A new curve detection method: Randomized Hough transform*, Pattern Recognition Letters **11** (1990), 331–338.
- [88] S. Y. Yoon and K. C. Kim, *3D particle position and 3D velocity measurement in a microvolume via the defocusing concept*, Measurement Science and Technology **17** (2006), 2897–2905.
- [89] Z. Yu and Bajaj C., *Detecting circular and rectangular particles based on geometric feature detection in electron micrographs*, Journal of Structural Biology **145** (2004), 168–180.
- [90] Y. Zhu et al., *Automatic Particle Detection Through Efficient Hough Transforms*, IEEE Transactions on Medical Imaging **22** (2003), no. 9, 1053–1062.

- [91] Y. Zhu et al., *Automatic particle selection: results of a comparative study*, Journal of Structural Biology **145** (2004), no. 1, 3–14.

List of Acronyms

μ PIV	Micro Particle Image Velocimetry
μ PTV	Micro Particle Tracking Velocimetry
AOS	Additive Operator Splitting
CHT	Circular Hough Transform
CLMC	Catté, Lions, Morel, Coll
Cryo-EM	Cryo-Electron Microscopy
DT	Diffusion Tensor
GHT	General Hough Transform
HT	Hough Transform
IP	Integer Programming
LD	Linear Diffusion
LP	Linear Programming
LSV	Laser Speckle Velocimetry
NCC	Normalized Cross-Correlation
PDE	Partial Differential Equation
PIV	Particle Image Velocimetry
PTV	Particle Tracking Velocimetry
SHT	Standard Hough Transform