On the Development of a Volumetric Velocimetry Technique using Multiple Plenoptic Cameras

by

Timothy W. Fahringer Jr

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 15, 2018

Keywords: 3D PIV, light field imaging, plenoptic camera, tomography, volume reconstruction

Approved by

Brian Thurow, Chair, Associate Professor of Aerospace Engineering
Stephen Nichols, Assistant Professor of Aerospace Engineering
Stanley Reeves, Professor of Electrical and Computer Engineering
David Scarborough, Assistant Professor of Aerospace Engineering

Abstract

Plenoptic PIV was recently introduced as a viable three-dimensional, three-component velocimetry technique based on light field cameras. One of the main benefits of this technique is its single camera configuration allowing the technique to be applied in facilities with limited optical access. The main drawback of this configuration is decreased accuracy in the out-of-plane dimension. This dissertation presents a solution with the addition of a second plenoptic camera in a stereo-like configuration. A framework for reconstructing volumes with multiple plenoptic cameras including the volumetric calibration and reconstruction algorithms are presented. It is shown that the addition of a second camera doubles the reconstruction quality and removes the 'cigar'-like elongation associated with the single camera system. In addition, it was found that adding a third camera provided minimal benefit for the reconstruction quality of sparse particle fields. Further metrics of the reconstruction quality are quantified in terms of particle density, number of cameras, camera separation angle, voxel size, and the effect of common image noise sources. In addition, a synthetic Gaussian ring vortex is used to compare the accuracy of the single and two camera configurations. It was determined that the addition of a second camera reduces the RMSE velocity error from 0.85 to 0.23 voxels. Finally, the technique is applied experimentally on a ring vortex and comparisons are drawn from the four presented reconstruction algorithms.

The trade-off between spatial and angular resolution is the main consideration when designing a plenoptic camera. This dissertation provides guidelines for the selection of the microlens array using theoretical analysis as well as synthetic and experimental data for validation. It was determined that the optimal selection of the microlens size depends heavily on the desired volume depth and a good rule-of-thumb is the span of the volume should be ~ 1.1 DoF$_p$ (single pixel, or perspective, depth-of-field). It was also determined that while

this is the optimal selection, the robustness of the cross-correlation algorithm mitigates the effect of sub-optimal microlens selection allowing for a single configuration to be used in a wide variety of situations.

Acknowledgments

This dissertation represents the culmination of 7 years of work, sleepless nights, and endless debugging. Given the nature of the thesis, only my name is listed as an author; however many people have directly and indirectly led to the development of this work through their support, advice, and friendship. Unlike the people who have contributed scientifically to work on which this thesis is based, these people are not mentioned in the thesis itself. This section is dedicated to them.

To my PhD advisor Dr. *Brian Thurow*, thank you for taking a chance on me, and giving me the opportunity to work on what would become the hallmark of your research. The early days were some of the most exciting from building the first camera and taking the first images (I'm still mad we couldn't publish the Star Wars Lego images) to the hours of discussion on tomographic reconstruction, PIV, and more broadly light-field imaging it truly felt like a collaborative effort. Your ability to guide your students without micromanaging, allowing the organic growth of ideas through exploration is the reason I stayed for a PhD. The research environment you have cultivated has led from the early days when it was just an assistant professor, a first year PhD student, and a plenoptic camera; to a department chair with a team of PhD students and two post-doc's processing data taken from around the country working with some of the most prestigious institutions and labs in the world. For allowing me to be a part of your lab, travel the world, and for all the opportunities you have provided me, I thank you.

To Dr. *Stanley Reeves* thank you for taking the time to teach me the fundamentals of tomographic reconstruction in our weekly meetings at the beginning of my graduate studies. In addition to Dr. Reeves, I would like to thank Dr. *Stephen Nichols* and Dr. *David Scarborough* for being on my PhD committee.

To *Kyle Lynch* and *Zach Reid*, thank you for showing me the ropes of laser alignment and PLIF as an undergrad, as well as putting up with me asking an untold number of questions while you were trying to graduate. In addition, I would like to thank Kyle for laying the groundwork for plenoptic-PIV, and providing useful nuggets of knowledge about tomographic reconstruction and PIV which often proved to be profound.

To *Blake Melnick* I will always remember the long nights trying to acquire plenoptic-PIV measurements of a turbulent boundary layer, making the entire wind tunnel look like a drug den, and the bond we formed over nervousness to present at conferences. For this and showing that it was, in fact, possible to graduate with a PhD I thank you.

To *Harris Haynes* our competitions to create the best looking homework were legendary. Remembering the night where you locked yourself out of your office, while we were all up late working on conference papers, and we had to climb through my ceiling to unlock your door (that stick is still in the office) brings back tears of laughter. Your work ethic and attention to detail was an inspiration, your ability to function on less than four hours of sleep was extraordinary, and your love for Moe's BBQ was ridiculous. For you friendship I thank you.

To *Bryan Brock* thank you for the endless entertainment you provided as an office mate, from your ridiculous antics (including spilling coffee all over your monitors and keyboard, RIP) to the friendly discussion of politics, your friendship has always been appreciated. From games of Diablo, Catan, Battlestar Galactica, and The Resistance with Steven, Winston, and Todd to leaving work early to play golf in the summer we had a great time, possibly at the expense of work, during our time together in Auburn. Along with Steven and Stewart we have formed a friendship that has lasted much longer than our time at Auburn for this I thank you.

To *Steven Ritz* thank you for your constant friendship over the past 7 years. The early bonds we formed bitching at Bryan for getting all the legendaries in Diablo have lasted for the past 5 years even though you have moved to the other side of the country. Through the endless nights of gaming our friendship has remained, at the expensive of sleep. My yearly

excursions to LA for Blizzcon remains the highlight of my year, and the extra trip in 2017 to be the best man at your wedding was one of the great joys in my life. Thank you.

To *Kyle Johnson* thank you for your friendship over the past 5 years. From late nights working on math finals in between coats of paint, to disk golf, drunken nights (incl. the emergence of Wedding Tim), and car rides with Tucker your friendship has been greatly appreciated. I am still waiting for the call to go work at In-N-Out, and I will always remember our time sharing an office as it was a unique unicorn-like experience.

To *Elise Hall* my only dream is that you move past our rocky start filled with animosity, eye-rolls, and grumpiness, to the greener pastures of the bitter sarcasm that is our friendship. There were good times at tailgates, Christmas parties, conferences, and waiting for Hans-Werner to get scared by the paper-towel dispenser, and bad times with that damn globe experiment. For you bitter honesty, friendship, and planning all the conference trips I thank you.

To *Jenna Klemkowsky* thank you for bringing positivity to an office otherwise filled with anxiety. The memories of our first conference together, riding the slingshot, was a highlight of my graduate career even though you couldn't make it. More great times were had at numerous baseball, gymnastics, and football games as well as simply wasting time talking with you and Elise in your office. One day I will get you to purchase raw denim, but until then I thank you for your continued friendship and support in my old age.

To *Chris Clifford* thank you for your help in the design and fabrication of the vortex generation facility as well as the help conducting the experiments used in this work. In addition, your implementation of Dragon on the cluster has greatly aided in speeding up the data processing time for which we are all thankful. I would also like to acknowledge our heroic efforts trying to save the world from a pandemic (RIP Asia) and the equally heroic effort trying to carry you out of silver, for which I may need you to return the favor... For being a great friend and DM, I thank you.

During my tenure as a PhD student at Auburn I was given the opportunity to work for several months at NASA Langley research center. To my NASA LaRC mentor *Tony Humphreys* thank you for the opportunity and guidance you provided me during my time as an intern. To *Scott Bartram* and *Mark Fletcher* thank you for doing all the hard work setting up the experiments and laser systems. To *Brett Bathel* thank you for words of guidance and the opportunity to work with you on the plenoptic BOS experiments. Finally, I would like to thank *Paul Danehy* for giving me the opportunity to work in your lab and allowing my to finish my PhD in the meantime.

Finally, we come to the people whom I moved away at the start of this Journey 11 years ago: my family.

To my Mother, *Sally Parkhurst*, thank you for always believing, even when you probably shouldn't have, and instilling a since of ambition in me. It feels impossible to capture what a mother means to their child in a few words, but from my earliest memories living in the gatehouse watching countless VHS tapes and having to play outside in my smurf rain boots while they exterminated the fleas, to this moment you have constantly pushed for me to have a better life and reach my goals. For constantly pushing me to be a better student, often up a very steep hill, and reach my potential I thank and love you.

To my Father, *Tim Fahringer*, thank you for instilling in me a great work ethic, which proved invaluable to finish this dissertation. Even though you and my mother were separated, you never missed a big moment. You never missed a soccer, baseball, or football game. When I broke my collarbone you arrived at the hospital before my mother and me and on my move to Auburn, when my car broke down, you dropped everything to haul my car and me to Auburn, then immediately drove back home to work the next day. The impact of these moments and many more are immeasurable. Working with you during the summers, where we would start early to leave early (this never happened), prepared me for the long nights I would endure in graduate school, albeit without the constant yelling of "shift". For always being there, supporting me, and pushing me I thank and love you.

To my step-parents, *Don Parkhurst* and *Shirley Fahringer*, the two of you have been a great blessing in my life. For as long as I can remember you been apart of my life, and it is impossible to imagine life without you. Because of you, my world opened up to a whole new set of experiences, from trips to Ocean City, the Outer Banks, the Caribbean, countless family reunions, and much much more. There are a lot of people who wish for two loving parents, I am lucky enough to have four. For your love and support, I thank and love you.

To my siblings *Michael Gardner*, *Kelly Gardner*, *Michelle Gardner*, *Olivia Parkhurst*, and *Grant Parkhurst* thank you for the support over the years; hopefully now that this dissertation is complete we will be able to spend more time together. To my Grandparents *Maw*, *Mom-mom* and *Pop-pop*, *Grandma* and *Grandpa*, and *Tinka* and *Big Don* thank you for the unending love and support you have given me throughout my life. To the remainder of my family, which is far to numerous to mention, know that your love, support, and kindness has not gone unnoticed and I thank and love you.

To the faculty, staff, and the Auburn family thank you.

-War Damn Eagle.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Throughout history, the curiosity of our species has constantly pushed the boundaries of exploration, expediting innovation in the design and construction of aerodynamic vehicles. From sea-faring vessels which allowed our ancestors to sail around the world to the Saturn V which propelled man to the moon exploration is truly our hallmark. Unfortunately the design of these vehicles has often been the result of trial-and-error experimentation and incomplete mathematical models as our fundamental understanding of fluid mechanics has lagged behind the ingenuity of the engineers and the bravery of the explorers. Mathematical descriptions of fluid mechanics have existed for centuries with an inviscid/incompressible solution given by Euler in the mid 1700's, termed the Euler equations, and a mathematically complete description given by Claude-Louis Navier and George Gabriel Stokes in the early 1800's, termed the Navier-Stokes equations. Derived by applying the principles of conservation of mass, momentum, and energy to a fluid element, the Navier-Stokes equations yield an exact expression for fluid motion; however in the following two-hundred years no general solution has been found; in fact, there is not proof that a solution always exists, or if a solution does exist that it would be void of mathematical singularities. This lack of an analytical solution has led to two complimentary fields of study: experimental fluid dynamics (EFD) and computational fluid dynamics (CFD).

The field of computational fluid dynamics focuses on modeling the Navier-Stokes equations such that they can be solved numerically with a computer. Since a computer can only operate on discrete elements, these computations are, typically, performed at particular points in a Eulerian (not changing with time) mesh, with some predefined spacing. The most

accurate model, which can resolve the full temporal and spatial scales of the turbulence, involves a full numerical simulation of the Navier-Stokes equations termed the direct numerical simulation (DNS). While very accurate, this solution is, unfortunately, very computationally expensive requiring the smallest scales of the flow to be resolved by the computational mesh and the time integration. Both of these issues scale poorly with Reynolds number, limiting this technique to low Reynolds number flows. The two most common models for use with real-world applications are the large eddy simulation (LES), and Reynolds averaged Navier-Stokes equations (RANS). The RANS model is created by decomposing the instantaneous Navier-Stokes equations into time-averaged and fluctuating components. The solution given by the RANS model is approximately the time-averaged solution to the Navier-Stokes equations. Alternatively, the LES model is able to provide simulations of instantaneous flow-fields and their topology as a function of time. This method seeks to reduce the computational burden of DNS by removing the smallest turbulent scales from the simulation via filtering, effectively yielding a low-pass filter effect on the solution. It is noted, that the strict removal of these scales will affect the solution since the energy cascade from large eddies to smaller eddies, which is fundamental to turbulence, is artificially broken. Therefore modeling of these effects is typically added, and is a very active area of research.

1.1 Survey of Velocity Measurements Techniques

Experimental fluid dynamics is the direct study of fluid mechanics via a real-world experiment. A lot of research effort, including this work, has been done to develop techniques to accurately measure the properties of the fluid flow. One of the most sought after properties is the vector velocity of the fluid, from which derivative quantities such as the Reynolds stress tensor and vorticity vector can be calculated. The earliest of these techniques utilized physical probes such as the pitot tube and hot wire. More recently, non-intrusive, laser based measurements have become the norm and a lot of work has been done to extend these methods to two and three dimensions.

A pitot tube is able to measure the velocity of a gas by measuring the stagnation pressure and the static pressure simultaneously. Using Bernoulli's equation a simple relationship between the velocity and the difference of these quantities can be found for incompressible flows. A more complex relationship exists for compressible flows known as Rayleigh's pitot tube formula. Several of these devices form a pitot-static system which can be found on most aircraft and are used to determine airspeed, Mach number, altitude, and altitude trend. The major limitation for this system is its inability to measure very low speed flows and velocity fluctuations to a high precision.

Hot-wire anemometry is a technique which utilizes a very thin wire (several microns) that is electrically heated. Since the electrical resistance of most metals is temperature dependent, as the flow moves around the wire, cooling the wire, a relationship between the electrical resistance and the flow velocity can be determined. These instruments tend to be very delicate, but provide high frequency response and spatial resolution of the flow velocity for one component of the velocity field. Additional components can be determined by adding additional wires. One such configuration is the so-called 'x' wire, where two wires are placed orthogonal forming an 'x' shape allowing for two components of velocity to be determined. This technique is still used extensively in turbulence studies, and the validation of other techniques due to its high frequency response and spatial resolution. The main limitations of this technique are its inability to measure reverse flow as well as its delicate nature.

These probes, while still used today, create issues due to their physical presence in the flow-field causing disturbances in the flow making measurements near or behind the device inaccurate. This limits the amount of these devices that can be used simultaneously resulting in long test times if more than a single point of data is desired. Non-intrusive, laser-based diagnostics have been developed to augment these devices and are typically based on inferring the motion from illuminated tracer particles or atomic/molecular species undergoing excitation.

The first class of non-intrusive laser-diagnostics for velocimetry are those based on the imaging of particles. The most direct of these approaches is *particle image velocimetry* (PIV, Adrian and Yao [7]) which directly measures the motion of the individual tracer particles or small groups of particles from two images taken with a small time delay. This method, like all particle-based techniques, assume that the tracer particles accurately follow the flow. This technique has seen widespread use do to its ability to measure a 2D plane of velocity with a relatively simple experimental arrangement and straightforward data processing. Recently, there has been a concerted effort in the community to extend this technique to 3D, including this work, allowing for the full velocity field to be captured. A more complete survey of PIV is given in the following chapter.

The other major set of particle-based velocimetry techniques are based on measuring the frequency shift that occurs when light is scattered off of a moving particle. This frequency shift, caused by the Doppler effect, can be directly related to the particle velocity. There are several techniques which measure this shift. The first, a point measurement, known as *laser Doppler anemometry/velocimetry* (LDA/LDV, Tropea *et al.* [8]) is based on crossing two coherent laser beams, which are focused at the point of intersection, generating a set of straight fringes forming the probe volume. As a particle crosses into the probe volume, the light is scattered onto a photo detector where the resultant fringe pattern (spacing) can be analyzed to determine the Doppler-shifted frequency. Then by comparing this frequency to the frequency of the incident laser light the Doppler shift and thus the particle's velocity can be determined. *Planar Doppler velocimetry* (PDV), or *Doppler global velocimetry* (DGV, Komine *et al.* [9]), is able to measure this Doppler shift in a 2D plane. This technique, however, measures the Doppler shifted frequency using light-frequency discriminators which absorb light as a function of frequency. Typically, these systems employ a reference camera which simply images the scattered light from the particles, and a signal camera which measures the scattered light after it has been filtered via a light-frequency discriminator, typically an iodine vapor cell. The difference between these two images can be related to

the absorption characteristics of the iodine cell which is a function of frequency. From this relationship the Doppler shift, and therefore velocity, can be derived.

The second class of methods is based on exciting atoms/molecules, either already present in or added to the flow, then tracking their motion. One such method is termed *molecular tagging velocimetry* (MTV, Hiller *et al.* [10]). This technique utilizes a 'write' laser to excite or 'tag' lines of molecules turning them into long lifetime tracers. These lines are then recorded by a camera at two successive times after excitation. The motion of the lines between the two frames yields the motion of the fluid. The 2D velocity can be determined by using two write lasers forming crossing lines. The major benefit to this method is that, often, the molecules are already present in the flow allowing this technique to be used in facilities where adding particles is problematic. The drawback is the limited amount of velocity information that can be obtained in a single snapshot, since you need to write discrete lines and thus the information in-between the lines is not resolved. Another method which uses excitation is called *femtosecond laser electronic excitation tagging* (FLEET, Michael *et al.* [11]). This technique focuses a femto-second laser to a point in the flow-field disassociating Nitrogen molecules into Nitrogen atoms, which when they recombine fluoresce. This fluorescence can be recorded for multiple frames allowing for the velocity to be determined. The benefit of this technique is its experimental simplicity requiring a single laser, camera, and focusing optics; however FLEET is only able to measure a point or line and is typically limited to flows with a very high, or even pure, Nitrogen concentration.

1.2 Roadmap

This work focuses on the adaptation of tomographic reconstruction to the multiple plenoptic camera solution. Adding a second plenoptic camera increases the experimental complexity of the overall system; however, this increase is only marginal when compared to other multi-camera techniques such as tomo-PIV (3-5 cameras) or synthetic aperture PIV (9 cameras). Moreover, additional plenoptic cameras provide an increase in robustness and

accuracy to the technique, but are not explicitly necessary unlike the aforementioned techniques. This flexibility allows the plenoptic PIV system to be easily tailored to a given experiment whether experimental complexity/optical access or accuracy is more important. The focus of this dissertation is on the development of a general framework for the reconstruction of 3D intensity fields from multiple plenoptic cameras, in this case for 3D PIV, but it could also be used for the development of other techniques such as volumetric scalar imaging. During the completion of this work software was necessarily developed including: an implementation of the VODIM cross-correlation based PIV algorithm, development and implementation of the refocusing/perspective image synthesis capabilities unique to light-field cameras, development of a tool used to simulate plenoptic images, and the development and adaptation of reconstruction algorithms suitable for plenoptic PIV. In addition, smaller but significant software programs were developed for calibration, image processing, and data post-processing among many others. The majority of this work focuses on the development and implementation details; as such, an emphasis is placed on synthetic data, with a known solution, allowing quantitative assessments on the accuracy of the technique. Further development is done on the design on plenoptic cameras, specifically for particle imaging applications, and an experimental validation is performed.

Chapter 2 presents an overview of the traditional particle image velocimetry technique. Including detailed information on the experimental considerations (tracer particles, laser illumination) and the post-processing algorithms which were implemented for this work. Validation of the algorithm's implementation is provided using sample data from the PIV Challenge database. Finally, a brief survey of the current 3D/3C PIV methods are presented.

Chapter 3 provides an overview of light field imaging, and, in particular, the plenoptic camera. The algorithms used to decode the plenoptic image sensor into a light field as well as the algorithms used to synthesize images from the light field data are detailed. In addition, a tool used to generate synthetic plenoptic images, used for testing the plenoptic PIV algorithms, is presented.

Chapter 4 begins with a description of the necessary steps for processing a raw plenoptic image into a 3D/3C vector field. Two of those steps, volumetric calibration and volumetric reconstruction, are then described in detail; including three different reconstruction techniques: filtered refocusing, multiplicative refocusing, and tomographic reconstruction. A numerical assessment of the plenoptic-PIV technique including the effect of many experimental configurations as well as noise sources on the reconstruction quality is performed. In addition, a simulated Gaussian ring vortex is used to determine the benefit of a second camera on the 3D/3C velocity field. Finally, an experimental validation is performed using a ring vortex generated in a purpose-built water tank.

Chapter 5 describes the effect that the microlens size has on the reconstructed particle volume. First, the two depths-of-field associated with plenoptic cameras are related to the spatial blur and particle elongation effects common to the single-camera configuration. Then, a theoretical analysis was performed which relates the theoretical particle reconstruction error with the optimal microlens size for an arbitrary configurations. Synthetic particle reconstructions and vector fields are then used to validate this model. Finally, experimental data is presented using a modified version of the experiment from Chapter 4.

Chapter 6 provides concluding remarks.

Chapter 2

Particle Image Velocimetry

Particle image velocimetry (PIV) refers to a measurement technique, used in experimental fluid mechanics, that infers the instantaneous vector velocity of a flow field by capturing the motion of tracer particles immersed in the working fluid. Historically, the roots of PIV can be traced to laser speckle velocimetry (LSV, Meynart [12]) which measures high concentrations of tracer particles such that observing individual particles is impossible; instead the speckle pattern created by coherent light reflecting off the dense particle field is measured. Using the same operating principles, with reduced seeding density, a new technique known as pulsed laser velocimetry was developed by Adrian and Yao [7] which would later that year be referred to as particle image velocimetry (Adrian [13]). A modern traditional or planar-PIV system consists of a double-pulsed laser, light-sheet-forming optics, tracer particles, a digital camera capable of frame-straddling, and a computer for image post-processing. Each aspect of this system has received a great deal of research over the past 30 years and will be touched on briefly in this work. For a more detailed overview the reader is referred to Adrian and Westerweel [14], Raffel *et al.* [15], and Tropea *et al.* [8].

2.1 Working Principle of PIV

The process of extracting the velocity of a flow field using PIV is shown in Figure 2.1. To visualize the motion of the flow field, tracer particles are immersed in the working fluid (air, water, etc...). These particles must be small enough such that they closely follow the motion of the fluid, but large enough such that when illuminated they scatter enough light to be recorded by a camera. A double pulsed laser is used to illuminate the particles twice such that two images are recorded: one for each laser pulse. Then, the motion of the particles

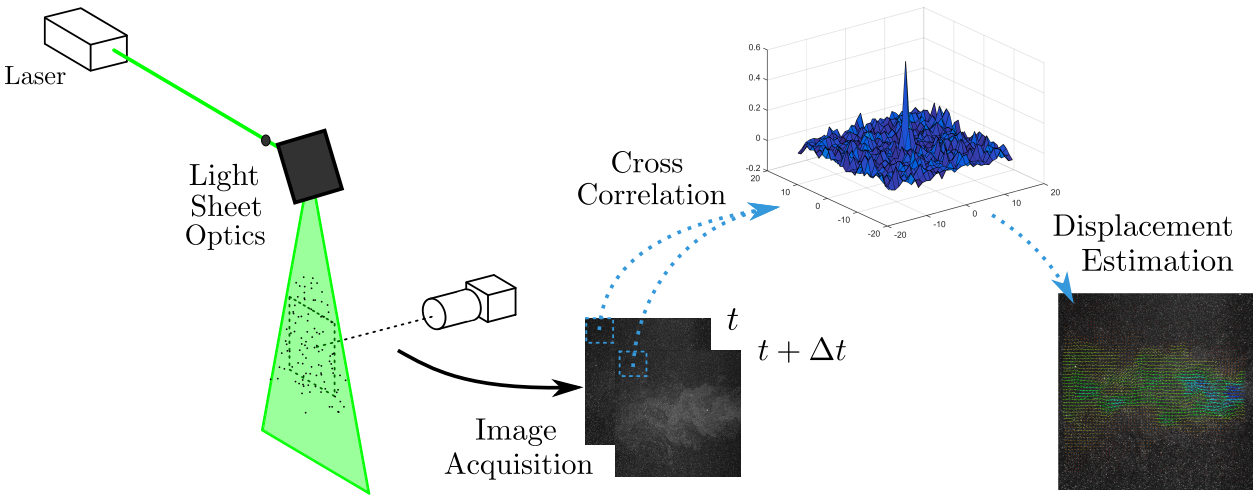


Figure 2.1: Schematic illustrating the working principle of PIV.

between the two images is determined using cross-correlation algorithms yielding the final velocity field.

Fundamentally the measurement acquired by a PIV system is the displacement of small groups of particles inferred from the acquisition of images at, typically, two distinct instances in time, t and $t + \Delta t$ where Δt is the known as the pulse separation. From the observed displacement and the pulse separation the velocity can be determined from its fundamental definition; however since there are only two data points the extracted velocity is temporally averaged over the pulse separation as expressed in equation 2.1.

$$V = \frac{\Delta x}{\Delta t} = \frac{1}{\Delta t} \int_{t}^{t+\Delta t} V(t)\ dt \tag{2.1}$$

In addition to the temporal averaging, the velocity, V, is spatially averaged since the data was extracted from a group of particles instead of the individual particles directly. These properties represent the fundamental assumptions in PIV: small groups of particles have the same, or very similar, velocities and that with a small enough pulse separation the motion of the particles is accurately approximated as a straight line. Additional assumptions and sources of error will be detailed as each part of the measurement technique is discussed.

2.2 Tracer Particles

One of the fundamental assumptions used in PIV is that the motion of the particles accurately represents the motion of the fluid. The equations of motion governing a particle immersed in a fluid are very complex and require assumptions to solve. One such approximation is known as the Boussinesq-Ossen equation, whose derivation can be found in Crowe *et al.* [16], which is predicated on the approximation of small spherical particles at low particle Reynolds numbers. For more information on the equations of motion governing tracer particles the reader is referred to Mei [17], for a solution to this equation in the context of turbulent flows, and Melling [18] for a detailed discussion on tracer particle tracking in general. In addition a broad review of tracers can be found in Adrian and Westerweel [14] and Tropea *et al.* [8].

The fidelity of a tracer particle can be characterized by the difference between its velocity ($\mathbf{u_p}$) and the fluid velocity ($\mathbf{u_f}$), known as the slip velocity ($\mathbf{u_s}$). Due to the use of very small particles in PIV applications the particle motion is dominated by Stokes drag. Neglecting the other terms in the Boussinesq-Oseen equation, the slip velocity can be calculated as

$$\mathbf{u_s} = \mathbf{u_p} - \mathbf{u_f} = d_p^2 \frac{(\rho_p - \rho_f)}{18\mu} \frac{d\mathbf{u_p}}{dt} \tag{2.2}$$

where d_p is the diameter of tracer particle, ρ_p and ρ_f are the densities of the particle and fluid respectively, and μ is the dynamic viscosity of the fluid. From equation 2.2 the most readily apparent way to reduce the slip velocity is to use neutrally buoyant particles ($\rho_p = \rho_f$) which would reduce the slip velocity to zero. For liquid flows, this condition can be easily met allowing for large ($\sim$50 μm) particles to be used. Unfortunately for gas flows the particle density is generally much larger ($\mathcal{O}(10^3)$) than the density of the gas. Therefore smaller particles must be used to minimize the slip velocity ($d_p < 5$ μm).

For a specific flow field, the ability of the tracer to follow the flow can be determined by comparing the response time of the tracer (τ_p) to the smallest time scales of the flow (τ_k). In

particular the tracers response time must be less than the smallest times scales of the flow in order to capture their motion. For the case of gas flows ($\rho_p \gg \rho_f$) the particle response to a step change in the flow velocity can be modeled by exponential decay and is given by

$$\tau_p = \frac{(\rho_p - \rho_f)d^2}{18\mu} \tag{2.3}$$

The smallest times scales of the flow can be easily calculated as the Kolmogorov time scale ($\tau_k = \ell/u$) of the relevant flow features. Typically the fidelity of a tracer particle in turbulent flows is presented as the Stokes number (St), which is defined as the ratio of the particle response time and the characteristic flow times scale as shown in equation 2.4. As a guide, acceptable tracer accuracy ($< 1\%$ error) can be met using the following condition: $St < 0.1$ [8].

$$St = \frac{\tau_p}{\tau_k} \tag{2.4}$$

The preceding analysis suggests to use very small particle to decrease the particle response time; however, the particles must also scatter enough light such that they can be recorded by a camera. Micrometer sized particles scatter light in the Mie regime and therefore the ability to scatter light is a function of particle diameter, wavelength of light, and relative refractive index of the particle with respect to the refractive index of the fluid. In this regime the intensity of scattered light is roughly proportional to d_p^2 and therefore careful particle selection is required such that the particle properly follows the flow and scatters enough light to be recorded.

2.3 Illumination

General considerations for illumination in PIV applications are the ability to shape the light, a narrow pulse width, and enough intensity to image the scattered light of particles. The ability to shape light is important since, for a 2D/2C PIV system, only a single plane or slice of the flow field is desired. As shown in Figure 2.1, the laser light is formed into a

thin sheet, such that all illuminated particles are in the same plane. This is required since the measurement is inherently 2D; if there is motion in the third dimension it will create a bias in the 2D/2C vector field. The narrow pulse width is required to create sharp particle images. If the light source has a large pulse and the particles are illuminated for the entire exposure of the camera, they would blur and create streaks. This would significantly hinder the cross-correlation analysis and lead to spurious vectors. These requirements can be easily met with the use of double pulsed Nd:YAG lasers which create coherent light that can be shaped using appropriate optics, have a narrow pulse width ($\sim$10 ns), and have ample energy (typically 10-400 mJ/pulse).

2.4 Image Processing

The image processing algorithms used in this work are implementations of the WIDIM algorithm (Scarano and Reithmuller [19]) and their 3D extension VODIM (Scarano and Poelma [20]). Figure 2.2 shows the steps in the predictor-corrector iteration of the WIDIM algorithm. This algorithm relies on iteratively updating the velocity estimation (predictor, v_p^k) using a cross-correlation correction v_c^k. For the first iteration large windows are used such that the 1/4 window rule is observed. In subsequent iterations the images are deformed such that the particle displacement should be zero allowing for smaller windows to be used. This process repeats for a desired number of iterations or until a convergence criteria is met.

2.4.1 Cross Correlation

The industry standard for determining the displacement of particles for high density PIV images is cross correlation. The use of correlation based analysis was originally developed using spatial correlation for double-exposed photographs by Adrian [13] and later for digital image pairs using cross correlation by Willert and Gharib [21]. For digital cross-correlation, each image is first subdivided into interrogation windows, then the corresponding windows between each frame are processed using a normalized 2D cross correlation. Mathematically,

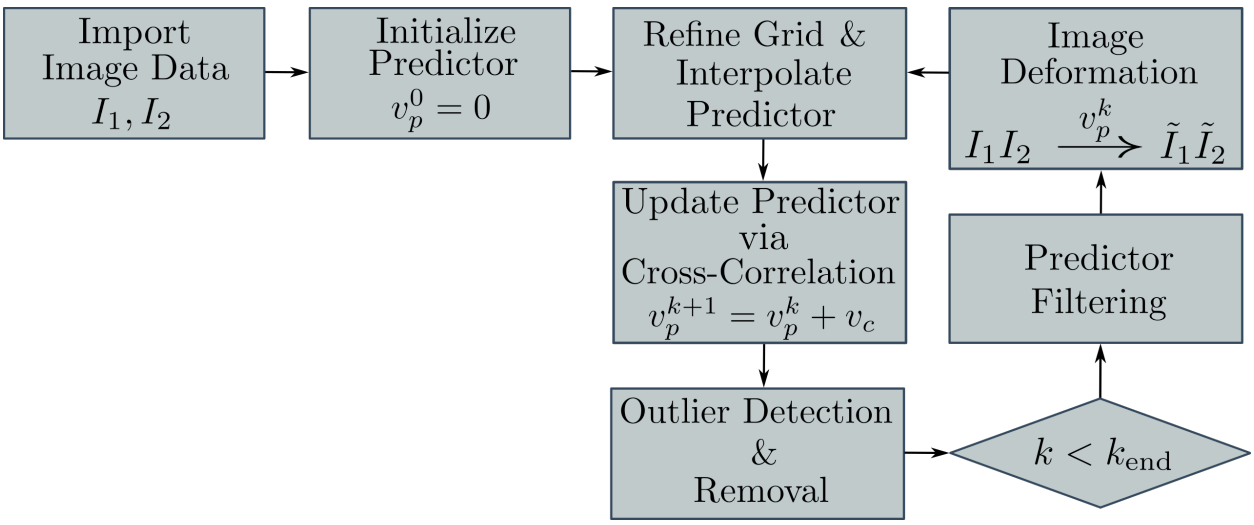


Figure 2.2: Flow chart showing the processing steps of the WIDIM/VODIM algorithm.

the normalized 2D cross correlation coefficient is shown in equation 2.5

$$\phi_{fg}(m,n) = \frac{\sum_{i,j}\left(f(i,j)-\overline{f}\right)\cdot\left(g(i-m,j-n)-\overline{g}\right)}{\sqrt{\sum_{i,j}\left(f(i,j)-\overline{f}\right)^2\cdot\sum_{i,j}\left(g(i-m,j-n)-\overline{g}\right)^2}} \tag{2.5}$$

where f and g are the interrogation windows of image I_1 and I_2 respectively. Note that $\overline{g}$ is the average of the shifted window. This function has values between 1 and -1 where a value of 1 corresponds to $f = g$ and -1 to $f = -g$. An example correlation map (ϕ_{fg}) and the interrogation windows that produced it are shown in Figure 2.3. The flow field in this example is a linear shift imaged by noiseless synthetic images which is shown to generate a very strong correlation peak ($\phi_{fg} \approx 0.9$) whose location is the estimate of the particle displacement.

While the direct cross correlation (DCC) approach provides an accurate method of determining the displacement of a group of particles it is computationally expensive. Computationally, the operation shown in equation 2.5 takes $\mathcal{O}(d_w^4)$ operations where d_w is the size of the interrogation window in pixels. An alternative approach is to do the correlation in the frequency domain utilizing the convolution theorem, which states that convolution in the spatial/time domain is multiplication in the frequency domain. This new formulation relies

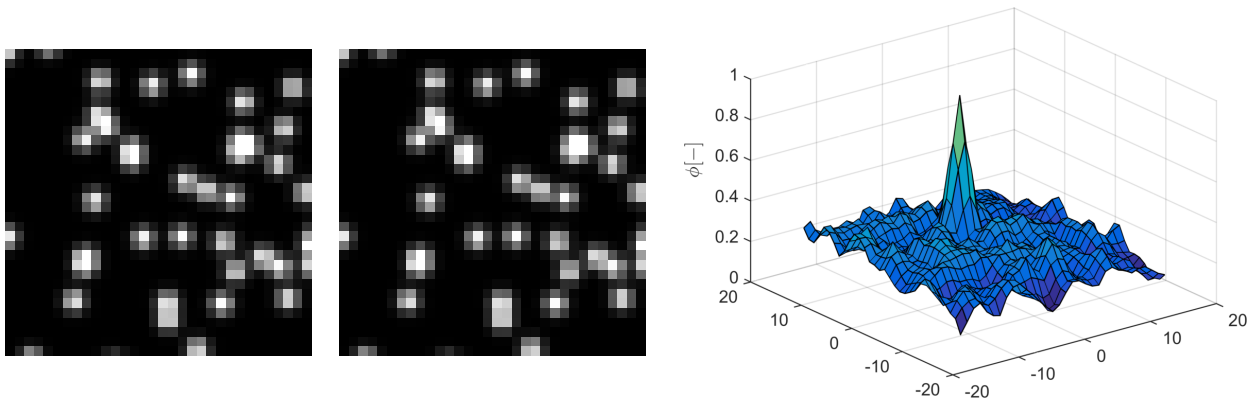


Figure 2.3: Example correlation map ϕ_{fg} (*right*) produced from 32 x 32 pixel interrogation windows extracted from f (t, *left*) and g ($t + \Delta t$, *middle*)

on computing the fast Fourier transform (FFT) on each interrogation window, multiplying one times the complex conjugate of the other (turning convolution into cross correlation), then taking the inverse FFT to return to the spatial domain. This process is mathematically given by

$$\phi_{fg} = \mathcal{F}^{-1}\left\{\mathcal{F}^{*}(f) \cdot \mathcal{F}(g)\right\} \tag{2.6}$$

where $\mathcal{F}$ is the FFT operator and $\mathcal{F}^{*}(f)$ is the complex conjugate of $\mathcal{F}(f)$. Due to the properties of the FFT this process only takes $\mathcal{O}(d_w^2 \log_2 d_w^2)$ operations. For a typical window size of 32 x 32 pixels this corresponds to a 100 times speedup over using the DCC algorithm; because of this, the FFT-based approach is used in this work.

Once the correlation map has been calculated the estimation of the displacement is calculated from the location of the tallest peak in the map. This is typically done by, first, finding the location of the largest value in ϕ_{fg}, then fitting a function around this element to determine the location to sub-pixel accuracy. The performance of some typical peak-fit functions (centroid, parabola, and a Gaussian) were evaluated by Scarano and Reithmuller [22] where they found that the use of a 3-point Gaussian fit yields the best result. In addition if the proper sub-pixel window shift is applied the bias error for the Gaussian fit is reduced to 0.011 pixels which is significantly less than expected random error [22]. The 3-point Gaussian peak-fit function is given in equation 2.7 where the subscripts -1, 0, and $+1$ indicate the

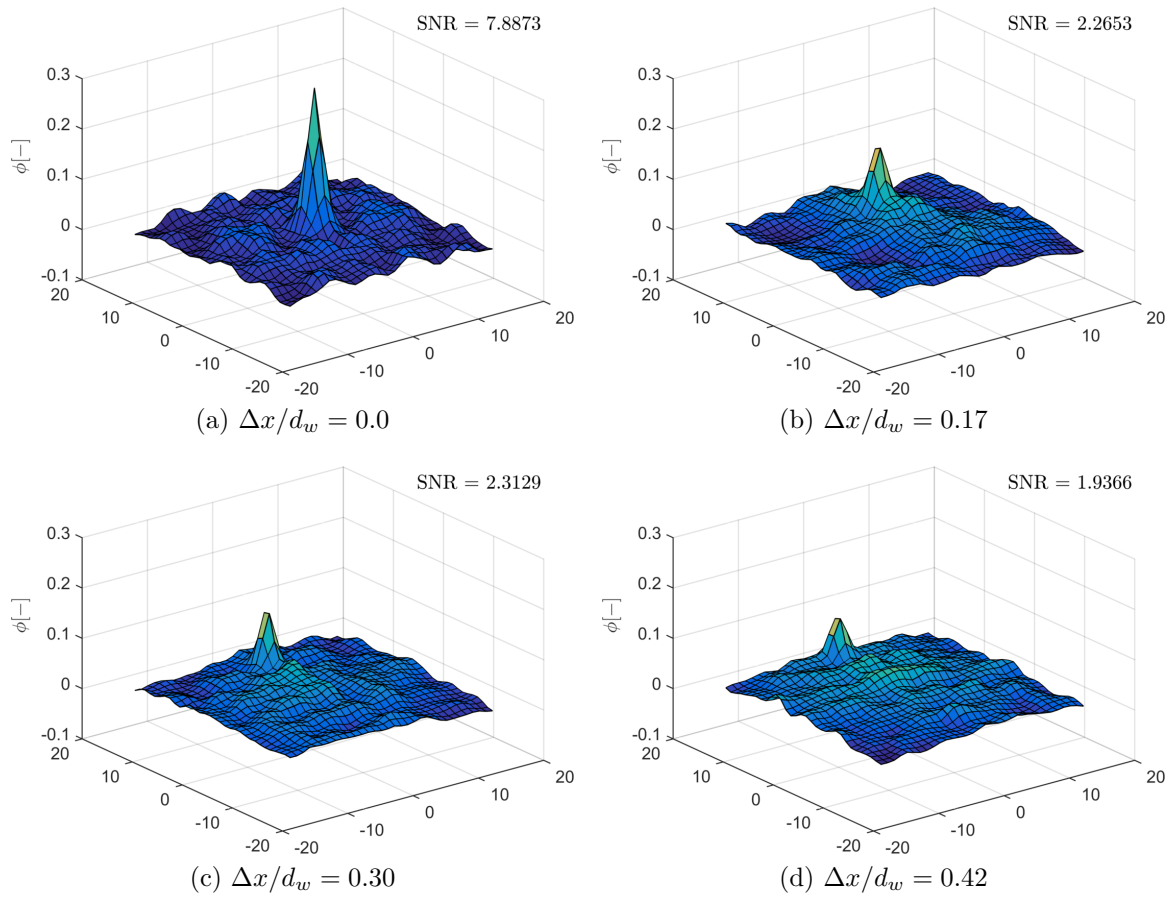
position in the correlation map relative to the maximum value. It is noted that this function is applied to the vertical and horizontal directions independently.

$$\epsilon = \frac{\ln \phi_{-1} - \ln \phi_{+1}}{2 \ln \phi_{-1} - 4 \ln \phi_0 + 2 \ln \phi_{+1}} \tag{2.7}$$

In addition to the primary peak, several weaker secondary peaks are shown throughout the map in Figure 2.3. One method of characterizing the quality of the correlation is the ratio between the primary peak and the largest secondary peak, known as the signal-to-noise ratio (SNR). Typically SNR values above 1.5 produce an accurate particle displacement estimate (Keane and Adrian [23]).

As outlined in Keane and Adrian [23] and Adrian and Westerweel [14] the three major factors which reduce the detectability of the correlation peak are in-plane displacement, out-of-plane displacement, and the displacement gradient within the interrogation window. The loss-of-correlation due to in-plane displacement F_I is shown in Figure 2.4 where the SNR is shown to reduce as the in-plane displacement Δx increases. This source of loss of correlation is particularly problematic when using the FFT-based correlation since FFT's are explicitly defined for periodic signals. To alleviate this the interrogation windows are typically zero-padded to at least twice their original size reducing the likelihood that the correlation signal wraps around to the other size of the interrogation window. In addition, a rule-of-thumb is to select window sizes such that the particle displacement is less that 1/4 of the window size, termed the quarter-window rule (Adrian and Westerweel [14]). This source of error can also be mitigated by using the DCC approach with large window sizes or using multi-pass method such as WIDIM which will be discussed in a later section.

The out-of-plane loss-of-correlation, F_O, is due to the particles moving out of the laser sheet in between the two frames such that a particle appears in one frame and not the other. The effect of this error is shown in Figure 2.5 where four correlation maps are shown with increasing out-of-plane motion given as the ratio of the z-displacement Δz and the laser sheet



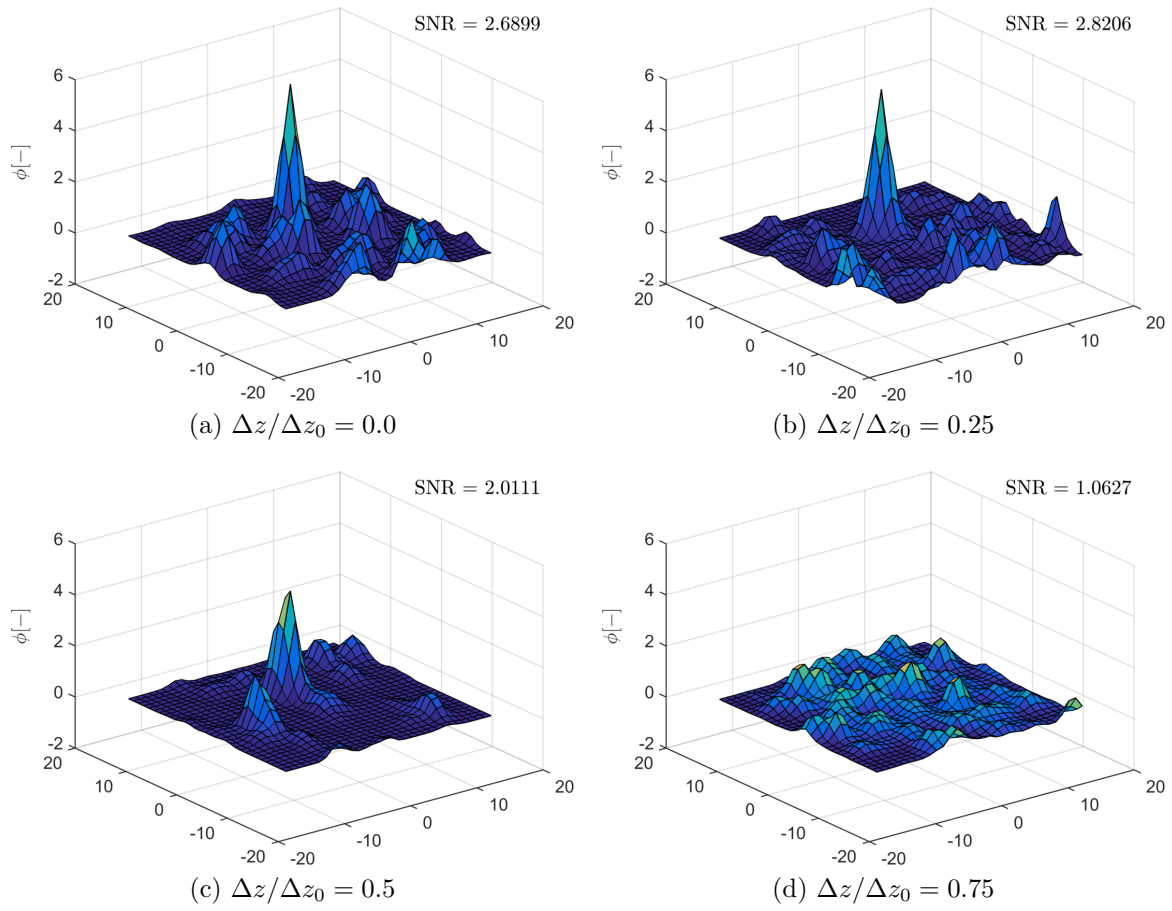
(a) $\Delta x/d_w = 0.0$ (b) $\Delta x/d_w = 0.17$

(c) $\Delta x/d_w = 0.30$ (d) $\Delta x/d_w = 0.42$

Figure 2.4: Correlation map for increasing in-plane displacement error, F_I, for FFT based cross-correlation.

thickness Δz_0. It is shown that as this ratio increases the SNR of the correlation decreases and at significantly high out-of-plane displacements the correlation peak is impossible to determine. This is a hard limitation of 2D/2C PIV measurements that cannot be solved with post-processing.

One inherent assumption made when using cross-correlation is that the motion within the window is constant. Unfortunately, real world flows have large-scale in-plane and out-of-plane velocity gradients which will cause a loss-of-correlation. Figure 2.6 shows the loss-of-correlation due to the spatial velocity gradient within the interrogation window, F_Δ. It is shown that as the gradient increases the amplitude of correlation peak reduces while the width broadens and the SNR decreases. It was shown in Adrian and Westerweel [14]



(a) $\Delta z/\Delta z_0 = 0.0$ (b) $\Delta z/\Delta z_0 = 0.25$

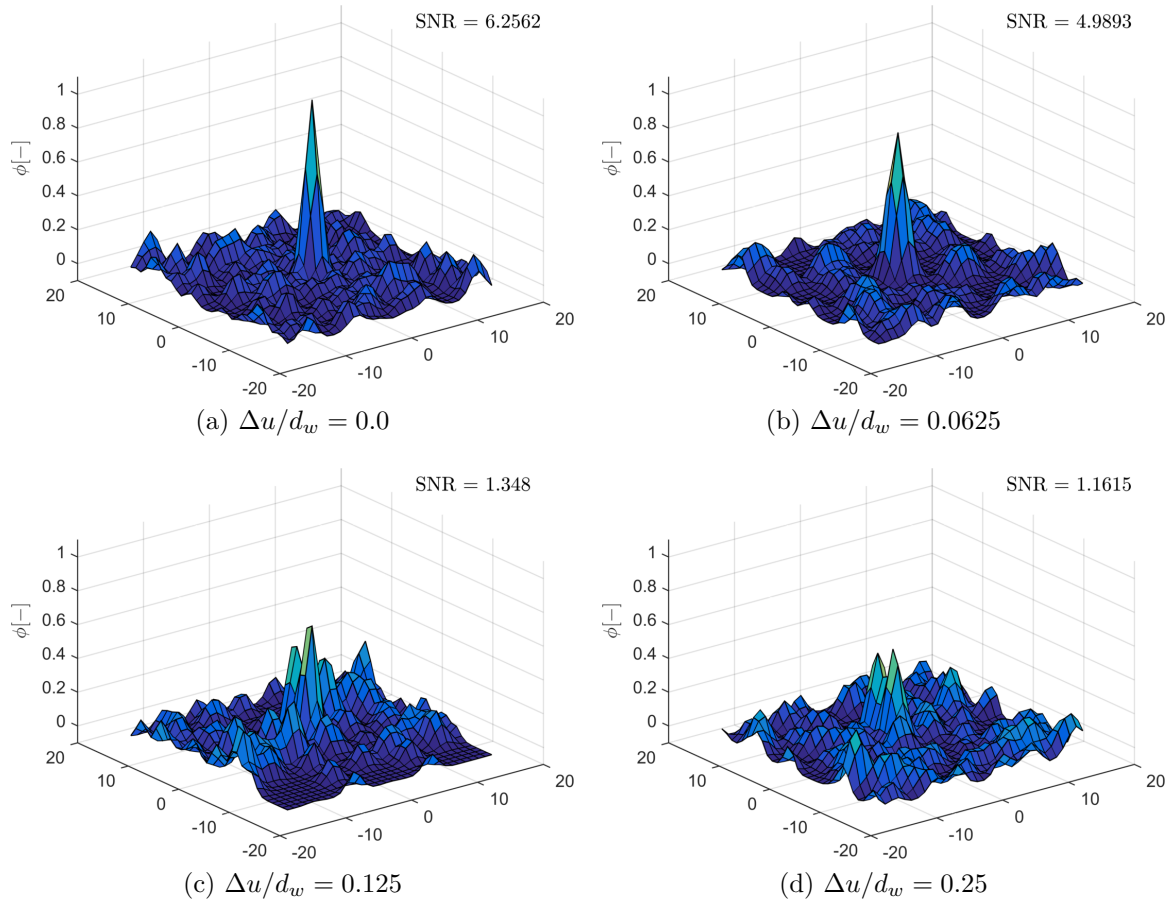
(c) $\Delta z/\Delta z_0 = 0.5$ (d) $\Delta z/\Delta z_0 = 0.75$

Figure 2.5: Correlation map for increasing out-of-plane displacement error, F_O, for FFT based cross-correlation.

that if the velocity gradient Δu is less than the particle diameter the loss-of-correlation is negligible. For stronger gradients window deformation methods have been developed to warp the interrogation regions to match the local gradients and are the topic of a later section.

2.4.2 Outlier Detection and Removal

Validation of the velocity prediction is performed every iteration in an attempt to mitigate spurious correlation due to image artifacts affecting subsequent iterations. This is particularly important in the WIDIM algorithm since the images for the next iteration are deformed based on the previous iteration. This process is broken up into two steps: outlier detection and outlier removal.



(a) $\Delta u/d_w = 0.0$ (b) $\Delta u/d_w = 0.0625$

(c) $\Delta u/d_w = 0.125$ (d) $\Delta u/d_w = 0.25$

Figure 2.6: Correlation map for increasing spatial gradient within the interrogation window, F_Δ, for FFT based cross-correlation.

To detect spurious vectors the universal outlier detection developed by Westerweel and Scarano [24] is applied. This method compares the median residual of each vector's velocity relative to a robust estimate of the local velocity variation. The method is considered universal in the fact that a single threshold value can be used to detect spurious vectors for any flow-field. For a displacement vector, v_0, the median of its KxK neighborhood is given by $u_m = \text{median}\{u_1, u_2, ..., u_{2K-1}\}$ where K is the kernel size (for this work $K = 2$). For each neighboring vector, u_i, the residual $r_i = |u_i - u_m|$ is calculated and the median of the residual $r_m = \text{median}\{r_1, r_2, ..., r_{2K-1}\}$ is determined. This value is then used to normalize

the residual of u_0, r_0, which is given by

$$r_0 = \frac{|u_0 - u_m|}{r_m} \tag{2.8}$$

For areas of low turbulence intensities (i.e. uniform flows) this value can become artificially high as $r_m \to 0$. To mitigate this effect a minimum normalization level, ε is added to equation 2.8 giving

$$r_0^* = \frac{|u_0 - u_m|}{r_m + \varepsilon} \tag{2.9}$$

It was determined by Westerweel and Scarano [24] that a suitable value of $\varepsilon = 0.1$ pixels representing typical cross correlation fluctuation levels. Once r_0^* has been calculated it is compared to a predetermined threshold, whose value is typically $\sim$2, which removes the largest 10% of residuals [24] and if the value is less than the threshold the vector is retained; otherwise the vector is invalid and replaced.

Once a spurious displacement vector is identified, two methods of replacement are used in this work. The first is to simply try to use an alternate peak from the correlation map. For this process the normalized residual test is applied to the displacement determined via the second and third highest peaks. The alternate peaks are considered in order (i.e if they both pass the test the second highest peak will be chosen) and if one of the alternate peak passes the test it is used in lieu of the first and the vector is labeled valid. If all three of the highest peaks are invalid, the vector displacement is determined by calculating the weighted average of its valid neighboring vectors.

2.4.3 High-Order Correlation via Image Deformation

Mitigation of the loss of correlation associated with large in-plane displacement, F_I and the displacement gradient within an interrogation window, F_Δ is accomplished by applying the preceding algorithm iteratively. The method referred to as window displacement iterative

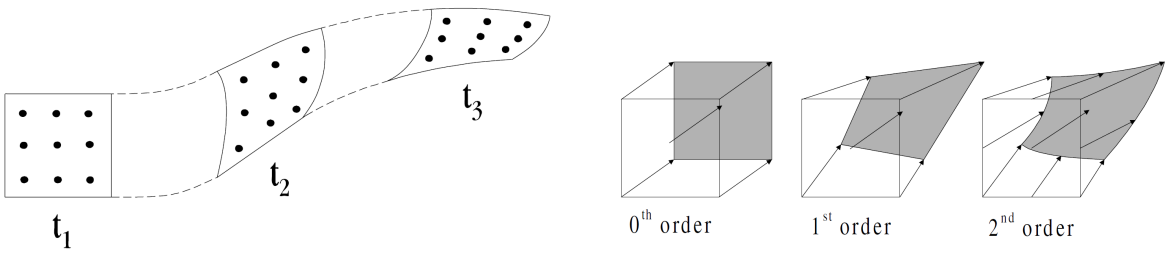


Figure 2.7: Schematics showing the motion of a fluid element and tracers (*left*) the effect of truncation order on the window deformation (*right*). From Scarano [1]

multigrid (WIDIM, Scarano and Riethmuller [22]) has become the defacto standard algorithm using this approach. The WIDIM algorithm starts with large interrogation windows (observing the 1/4 window rule) to initialize the predictor v_p^k. For each subsequent iteration the interrogation regions in the two input images I_1 and I_2 are shifted/deformed using the predictor such that, if correct, the images will be identical. In effect, this shifts the correlation peak to the center of the correlation map eliminating the effect of F_I in subsequent iterations. Furthermore, moving the correlation peak to the center of the correlation map allows for much smaller windows to be used and in general each iteration reduces the size of the windows and the spacing between them. The process of mitigating F_Δ by deforming the interrogation windows, as detailed in Scarano [1], is the focus for the rest of this section.

Figure 2.7 (*left*) shows the motion of a cluster of particles tracing a fluid element through three time instants. For each instant in time, t_1, t_2, t_3, there exists some displacement $d(x, y, t)$ that describes the motion to the next instant in time. Using the cross-correlation algorithm described earlier, the displacement can be estimated, in a general sense, with the following expression

$$\max_d \int_W f(x,y)g(x+\Delta x, y+\Delta y)\,dx\,dy \tag{2.10}$$

where the displacement, d, is determined by maximizing the correlation between f and g, displaced by $(\Delta x, \Delta y)$, over the interrogation region W. The displacement given from this formula represents the most probable displacement of the particle field within the interrogation region. If the displacement $(\Delta x, \Delta y)$ is a uniform shift, it is obvious that the

displacement would yield an exact match. The more general formulation, shown in equation 2.11, is when the applied shift varies as a function of position within the interrogation region.

$$\max_{\Delta x, \Delta y} \int_W f(x,y) g\left[x + \Delta x(x,y), y + \Delta y(x,y)\right] \, dx \, dy \tag{2.11}$$

This formula suggests that in order to obtain the correct displacement we must consider the velocity gradient across the interrogation window; otherwise the estimated displacement will be some weighted average of the particle ensemble. To obtain a more accurate estimate, the images can be deformed prior to the cross-correlation analysis using a prediction of $\Delta x(x,y)$ and $\Delta y(x,y)$. Mathematically, this is shown in equation 2.12 where the displacement distribution over the finite interrogation region can be estimated using a Taylor series

$$\begin{aligned} u(x,y) = u(x_0,y_0) &+ \left(\frac{\partial u}{\partial x}\right)(x-x_0) + \left(\frac{\partial u}{\partial y}\right)(y-y_0) + \dots \\ &+ \frac{1}{2!}\left[\left(\frac{\partial^2 u}{\partial x^2}\right)(x-x_0)^2 + \left(\frac{\partial^2 u}{\partial x \partial y}\right)(x-x_0)(y-y_0)\right. \\ &\left. + \left(\frac{\partial^2 u}{\partial y^2}\right)(y-y_0)^2\right] + \dots \\ &+ o(x-x_0)^3 \end{aligned} \tag{2.12}$$

where $x \in [x_0 - 0.5W, x_0 + 0.5W]$, $y \in [y_0 - 0.5W, y_0 + 0.5W]$, and (x_0, y_0) denotes the center of the interrogation region. Since the cross-correlation algorithm can only yield the zeroth order term in the Taylor series, image deformation techniques are used to compensate for the higher order terms such that the correction made by the cross-correlation algorithm should be a simple uniform shift. Therefore, the cross-correlation function in equation 2.5 can be rewritten using the velocity predictor as

$$\phi_{fg}(m,n) = \frac{\sum_{i,j}\left(f(i - \frac{u_p \Delta t}{2}, j - \frac{v_p \Delta t}{2}) - \overline{f}\right) \cdot \left(g(i - m + \frac{u_p \Delta t}{2}, j - n + \frac{u_p \Delta t}{2}) - \overline{g}\right)}{\sqrt{\sum_{i,j}\left(f(i - \frac{u_p \Delta t}{2}, j - \frac{v_p \Delta t}{2}) - \overline{f}\right)^2 \cdot \sum_{i,j}\left(g(i - m + \frac{u_p \Delta t}{2}, j - n + \frac{u_p \Delta t}{2}) - \overline{g}\right)^2}} \tag{2.13}$$

where u_p and v_p are the predicted spatial velocity determined from the previous iteration. If the prediction was correct the resultant correlation peak would be located at the origin of the correlation map, indicating zero additional shift, and would have a value of approximately one.

In order to apply equation 2.13 a predictor field must be built using some number of the parameters given from the Taylor series expansion. The effect of the first three terms are given in Figure 2.7 (*right*). It is shown that using a zeroth order predictor is equivalent to a uniform shift, a first order predictor is equivalent to a piecewise-linear interpolation, and so on. Several methods have been developed utilizing varying orders of the Taylor series expansion focusing mostly on the zeroth order and first order approximations as the higher order approximation add significant computational complexity for minimal gain.

For brevity, only the linear displacement predictor used in the WIDIM algorithm is considered here. For a full review of the deformation methods the reader is referred to Scarano [1]. For the first order deformation methods the velocity predictor is interpolated at each pixel using bilinear interpolation. Typically this is done using the central difference interrogation such that the deformed images are expressed as $\tilde{I}_1(x,y) = I_1(x - u_p^k/2, y - v_p^k/2)$ and $\tilde{I}_2(x,y) = I_2(x + u_p^k/2, y + v_p^k/2)$. The advantage of using the central difference over the forward difference is that it is second order time accurate (FD is 1st order), which improves the accuracy when using large pulse separations.

Algorithmically, each pixel has a unique predicted displacement interpolated from the surrounding vectors. Then, using this displacement, the irradiance value for that pixel is determined from the original image using interpolation. Due to the high-frequency content in PIV images the use of linear interpolation is not appropriate; instead the pixel intensities are interpolated using the cardinal function interpolation formula [1] which is given by

$$\tilde{I}(x,y) = I(x',y') = \sum_{i=-\infty}^{i=+\infty} \sum_{j=-\infty}^{j=+\infty} f(i,j) \times \frac{\sin[\pi(i-x')]}{\pi(i-x')} \frac{\sin[\pi(j-y')]}{\pi(j-y')} \tag{2.14}$$

where (x', y') is the sub-pixel location determined from the bilinear interpolation scheme. Computationally, it is impractical to consider all pixels from negative to positive infinity. As such, only a limited number of pixels (a 7x7 kernel is used in this work) are considered. One side effect of truncating the interpolation domain is an excessive rippling in the pass-band. One method of mitigating this effect is to multiply the truncated-sinc with a window function to smooth the response. For more information on the use of weighting windows with image deformation the reader is referred to Nogueira *et al.* [25] and Astarita [26].

An alternative approach is to filter the velocity predictor or corrector as shown in Schrijer and Scarano [27]. They showed the use of several filter techniques and identified critical values for the filter strength. In this work the use of the non-linear regression predictor filter is used. This can be shown to augment the predictor update to the following form

$$v_p^{k+1} = \mathbf{F}_{regr}\left(v_p^k\right) + r_c\left[v_0 - \mathbf{F}_{regr}\left(v_p^k\right)\right] \tag{2.15}$$

where $\mathbf{F}_{regr}\left(v_p^k\right)$ is the velocity field filtered with a two-dimensional least squares regression, r_c is the characteristic response of the window filter ($\mathrm{sinc}(d_w/\lambda)$ here), and v_0 is the exact particle displacement. Computationally this step is performed by computing the 2D least-square regression on a k by k region around each vector, then setting the velocity of each vector to the zeroth order coefficient.

2.5 Example Data

In order to provide additional validation for the particular implementation of the WIDIM algorithm used in this work it has been applied to several test cases from the PIV Challenge (pivchallenge.org) database. The PIV challenge database provides a set of well-designed test cases designed to test specific aspects, challenges, and limitations of the PIV processing algorithms. A small subset of the database has been chosen, focusing on 2D/2C PIV, to be presented in this work.

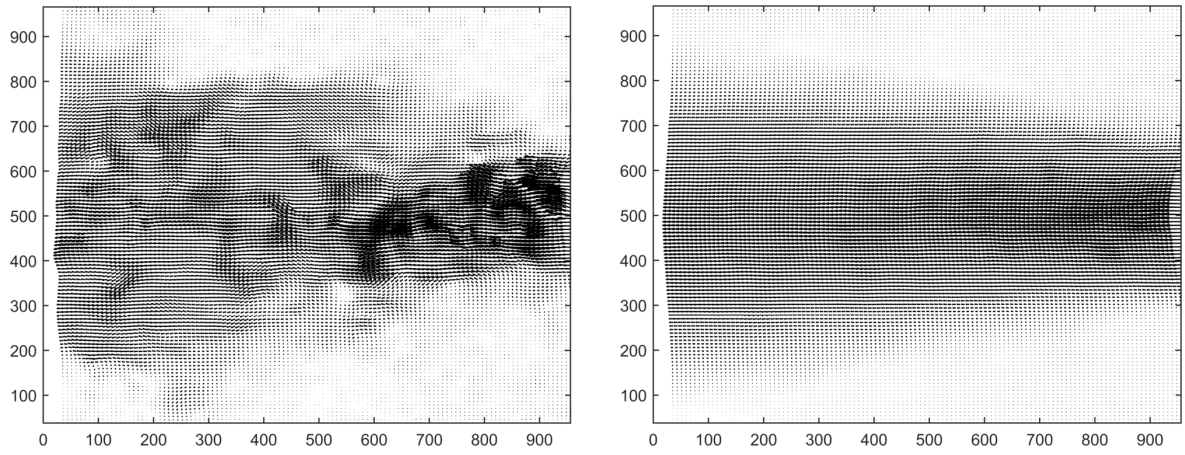


Figure 2.8: Results from processing 2003 PIV Challenge (Case A) data. Shown are an example instantaneous velocity field (image pair A100, *left*) and the ensemble average (*right*).

2.5.1 2003 PIV Challenge: Case A

The first test case presented is an experimental data set from a self-similar turbulent jet (case A from the 2003 PIV Challenge, Stanislas *et al.* [2]). This data set was designed to test the performance of the different algorithms relative to each other in non-ideal imaging conditions. Contained in the dataset are 100 image pairs of size 992 x 1004 pixels. An example instantaneous vector field is shown in Figure 2.8 (*left*) and an average of all 100 pairs is shown in Figure 2.8 (*right*). For more information on the experimental arrangement and to see the original results the reader is referred to Stanislas *et al.* [2].

This analysis was performed with initial window sizes of 64 x 64 and final windows of 32 x 32 with 50% overlap after 4 iterations using FFT based cross-correlation. For comparison, the data presented in Figure 2.8 (*right*) is compressed into a single one-dimensional profile. This is done by fitting, in a least-squares sense, the 2D mean displacement field to

$$U(x-x_0, y-y_0) = U_c(x)\exp\left[-\frac{(y-y_0)^2}{\lambda(x)^2}\right] \text{ with: } \begin{cases} U_c(x) = \frac{A}{(x-x_0)} \\ \lambda(x) = B\cdot(x-x_0) \end{cases} \tag{2.16}$$

Table 2.1: Parameters for analysis of 2003 PIV Challenge Case A.

Resolution	Num. Vectors	x_0	y_0	A	B
32x32	1357200	2163.5	500	-7703	0.1031

where (x_0, y_0) is the virtual jet origin, $U_c(x)$ is the mean centerline velocity, $\lambda(x)$ is the mean jet width, and B is the jet spreading rate. First the virtual jet origin was determined, for x_0, using the 1/5 rule and, for y_0, using the center of the streamwise displacement of the jet at $x = 956$ pixels. Then the fitted values A and B could be determined and are listed in Table 2.1. The data is then presented as a scaled mean velocity (u/U_c) as a function of the reduced radial coordinate $(\eta = (y - y_0)/\lambda(x))$ and is shown in Figure 2.9 where each data point is the average of all downstream locations. The mean axial velocity (*top, left*), mean radial velocity (*top, middle*) and the rms axial velocity fluctuations (*top right*) are presented in Figure 2.9. Comparing to the results published in Stanislas *et al.* [2], presented in the bottom row of Figure 2.9 for convenience, the mean axial velocity profile matches well. The mean radial velocity shows the same general trends, but due to the increased scatter between the methods its hard to obtain any notion of accuracy. For the rms axial fluctuations a close match between this implementation and the presented PIV data is observed.

2.5.2 2003 PIV Challenge: Case B

The second test case, from the same PIV Challenge, is a synthetic data set generated from a DNS simulation of a turbulent open channel flow. Unlike the first test, this test was designed to test each algorithm compared to a known solution not simply their performance relative to each other. The dataset contains 100 image pairs of size 1536 x 512 pixels. Sample results using 32 x 32 pixel windows with 75% offset are shown in Figure 2.10. This figure shows an example instantaneous pair (image pair B001) with Figure 2.10 (*top*) showing the instantaneous streamwise displacement contours and Figure 2.10 (*middle*) showing the instantaneous wall-normal displacement contours. In addition, the average streamwise

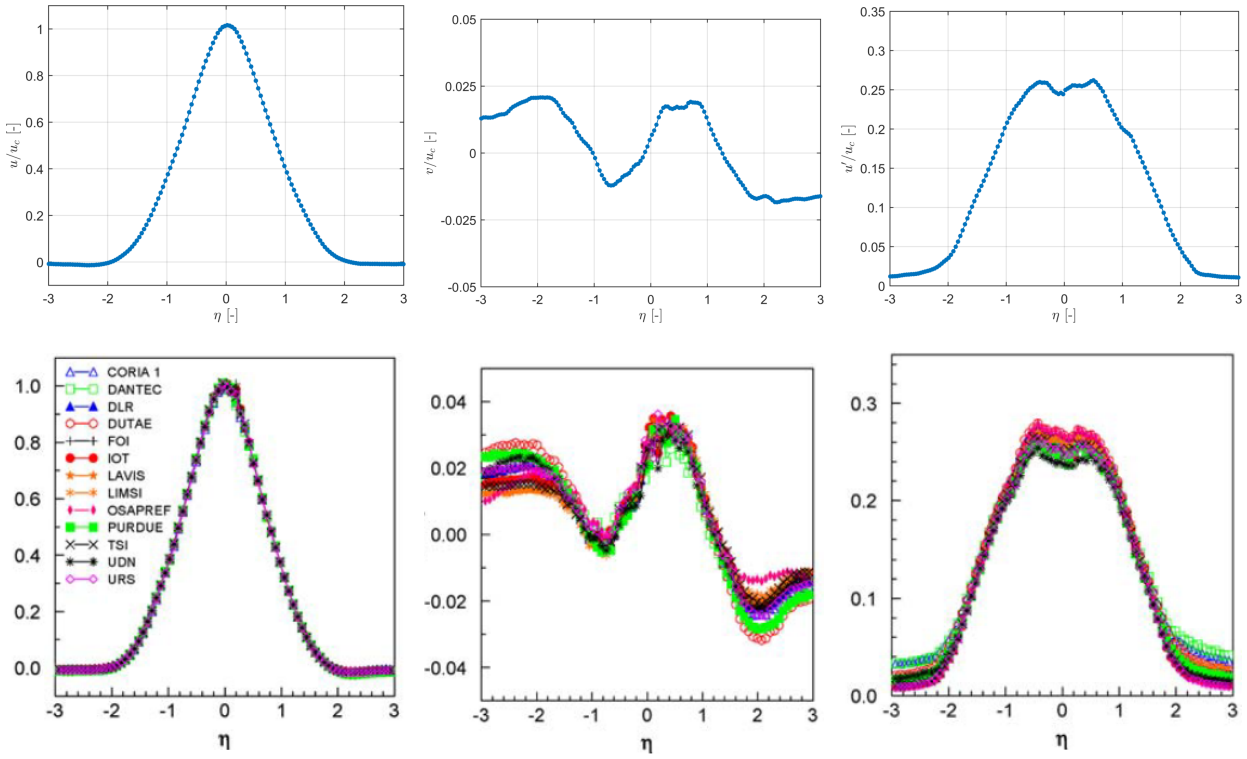


Figure 2.9: Results from processing 2003 PIV Challenge (Case A) data. Top row is data processed using the algorithms implemented for this work and bottom row is from the 2003 PIV Challenge [2]. Shown are the scaled results (as a function of the reduced radial coordinate $\eta = (y - y_0)/\lambda(x)$) for the mean axial velocity (*top left*), mean radial velocity (*top right*), rms axial velocity fluctuations (*bottom left*). A detail of the rms axial velocity fluctuations for $-4 < \eta < -2$ is shown in (*bottom right*).

displacement contours are shown in Figure 2.10 (*bottom*). From Figures 2.10 (*top & bottom*) the stratification of the boundary layer is easily identified.

In addition to the contour maps, the data is presented as mean line plots (averaged in x) which are shown in Figure 2.11. In Figure 2.11 (*top left*) the mean streamwise velocity profile showing the turbulent boundary layer profile. Figures 2.11 (*top middle & right*) show the streamwise and wall-normal turbulence intensities (u' & v') respectively. Looking at the data presented in Stanislas *et al.* [2], presented in the bottom row of Figure 2.11, the profiles presented here match to the expected results.

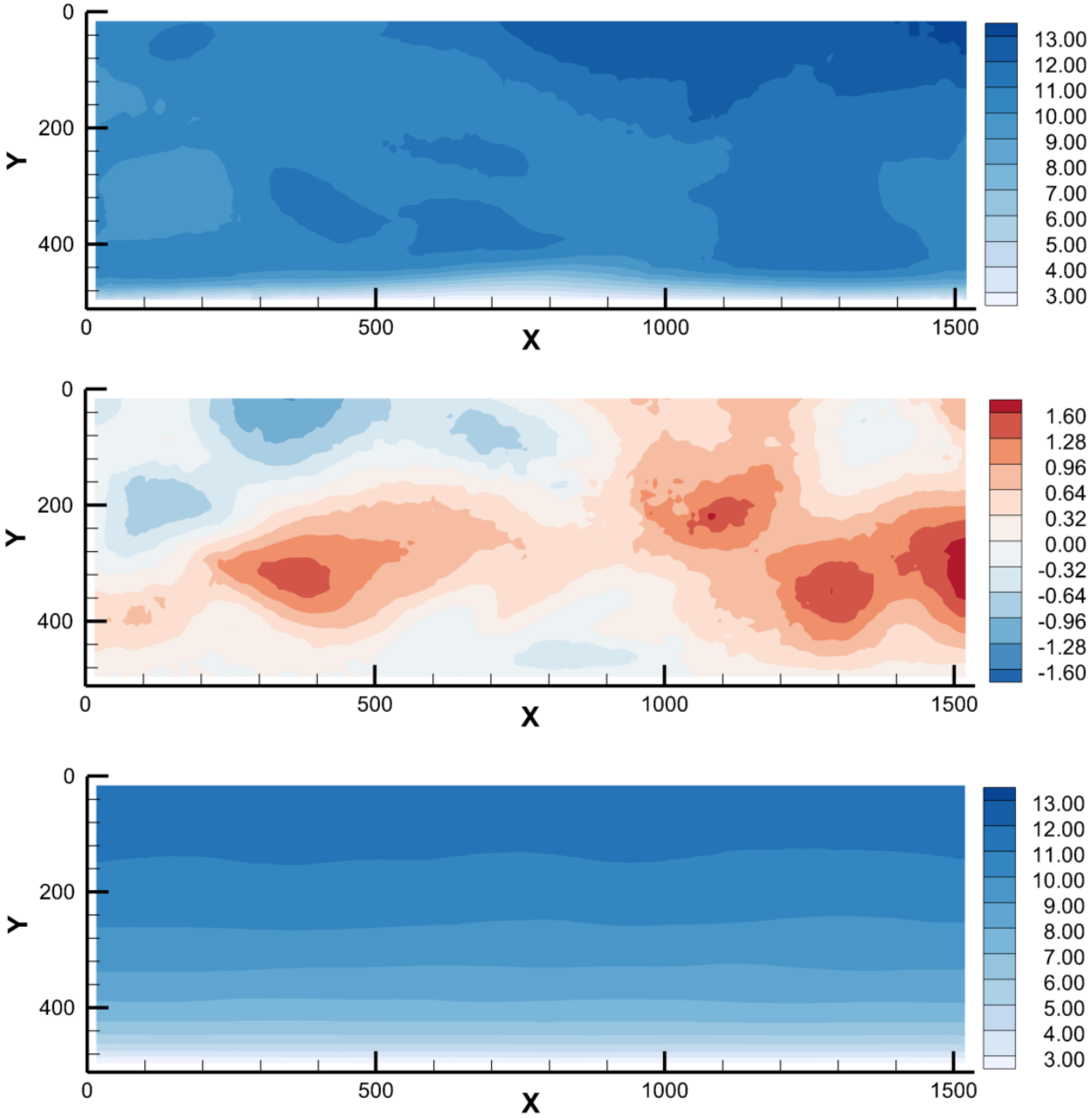


Figure 2.10: Results from processing 2005 PIV Challenge (Case B) data. Shown are the instantaneous streamwise component u (*top*) and the instantaneous wall-normal component v (*middle*) for image pair B001. In addition the ensemble average streamwise component is shown (*bottom*).

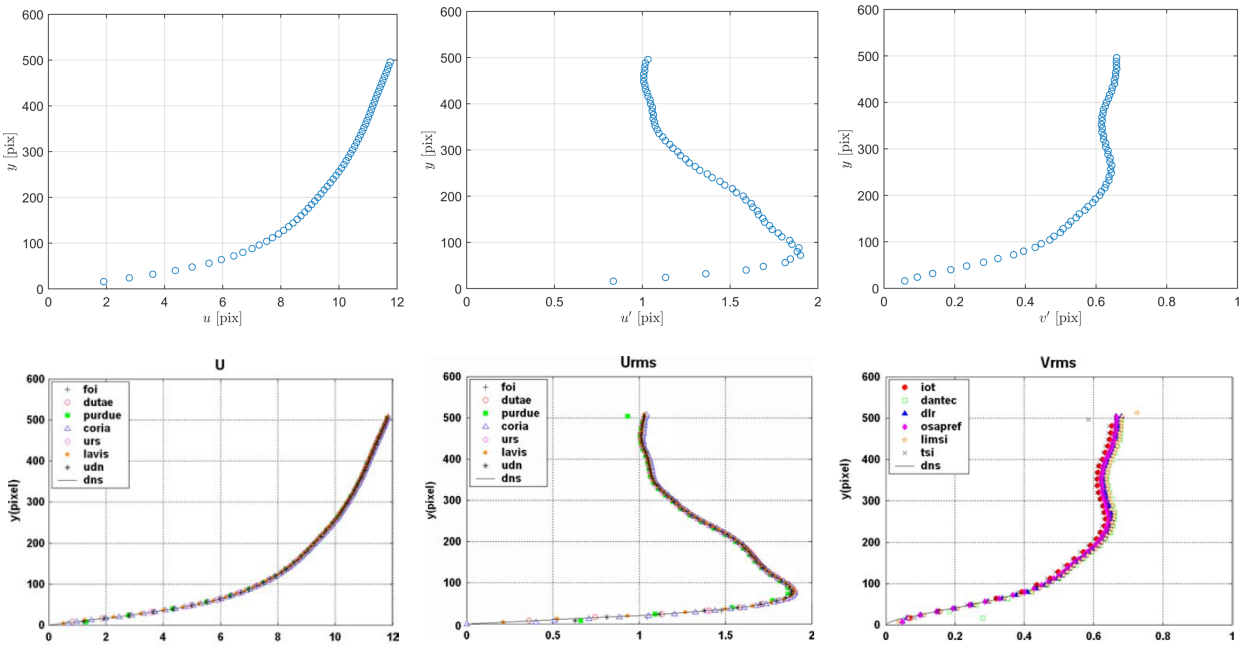


Figure 2.11: Results from processing 2003 PIV Challenge (Case B) data. Top row is data processed using the algorithms implemented for this work and bottom row is from the 2003 PIV Challenge [2]. Shown are the mean streamwise velocity profile (*left*), mean streamwise turbulence intensity (*middle*), and mean wall-normal turbulence intensity (*right*).

2.6 Stereoscopic PIV

The move to obtaining a more complete picture of the measured flow-field started with stereoscopic PIV (Arroyo and Greated [28], Willert [3]) which adds a second camera in order to resolve the third (out-of-plane) component of velocity. A typical camera configuration is shown in Figure 2.12 where the two cameras are imaging the same plane using the Scheimpflug condition. By enforcing the Scheimpflug condition, which forces the object plane, aperture plane, and image plane to intersect at a single point, both cameras are able to focus across the same object plane.

Once the images are recorded, the 2D/2C vector fields are processed separately for each camera using the traditional PIV algorithms described earlier. To reconstruct the 3C vector field the vectors from each camera are combined using the geometric relationship between

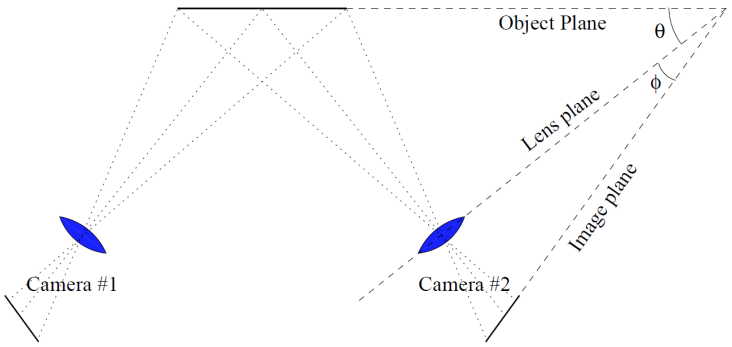


Figure 2.12: Typical stereoscopic PIV configuration applying the Scheimpflug condition. From Willert [3].

the two cameras. The equations used to reconstruct the 3C vector field are given by

$$\Delta x = \frac{B(\Delta x_1 + \Delta x_2) - x(\Delta x_1 - \Delta x_2)}{2B + \Delta x_1 - \Delta x_2} \tag{2.17}$$

$$\Delta y = \frac{B(\Delta y_1 + \Delta y_2) - y(\Delta x_1 - \Delta x_2)}{2B + \Delta x_1 - \Delta x_2} \tag{2.18}$$

$$\Delta z = \frac{H(\Delta x_1 - \Delta x_2)}{2B + \Delta x_1 - \Delta x_2} \tag{2.19}$$

where $2B$ is the distance between the two cameras, H is the distance from the cameras to the object plane, $(\Delta x_1, \Delta y_1)$ and $(\Delta x_2, \Delta y_2)$ are the 2D/2C vector fields from camera 1 and camera 2 respectively, and (x, y) is the position of the particular vector.

The addition of the second camera to resolve the third component of velocity adds new sources of error. In particular the error associated with the angle between the two cameras and the registration error from matching the two cameras to each other. As shown in Prasad [29], for shallow angles $\theta < \pm 20°$ the out-of-plane error is significantly higher (> 3 times) than the in-plane error. As the separation angle increases to $\theta = \pm 45°$ the out-of-plane error becomes equivalent to the in-plane error; however at these high angles the geometric reconstruction error increases.

To mitigate the error associated with geometrically matching both camera images, an empirical calibration procedure is required. This procedure generally involves the simultaneous imaging of a calibration target (dot card) by both cameras. By using an empirical

approach all distortions between the image sensor and object plane can be corrected at once. Two different methods are typically used to apply the warping; the first is to fit a second or third-order polynomial to the calibration points (Soloff *et al.* [30]) and the second is to use functions derived from the camera pinhole model (Willert [3]). One issue with these techniques is that it is necessary to position the calibration target at the exact same plane as the light sheet. Due to this, correction schemes, known as self-calibration, have been developed by Willert [3], Coudert and Schon [31], and Wieneke [32].

2.7 Extension to 3D

To apply the preceding algorithms to the data generated from volumetric particle reconstructions (see Chapter 4) they must be adapted to 3D. This is done, simply, by extending the domain of each step to include a third dimension (i.e. adding an additional for loop) or, when applicable, by using the 3D version of existing algorithms (i.e using a 3D FFT in lieu of the 2D version). The algorithms and processing steps are the same, but due to these requirements, the code base for the 3D VODIM is separate from the 2D WINDIM code and therefore requires additional testing. To test the 3D algorithms a 3D vector field was generated using the analytical function of a Gaussian ring vortex. This test will be used extensively in Chapter 4 to evaluate the multi-camera plenoptic PIV technique as a whole so the discussion of this vortex will be left to section 4.3.4. For this test, the volume of particles will be artificially generated, bypassing any reconstruction artifacts, by placing 15,000 particles (3 x 3 x 3 Gaussian spheres) within a discretized volume (30 x 30 x 30 mm, 260 x 260 x 260 voxels) and displacing them via the analytical equation. Then, the 3D VODIM algorithm is used to determine the displacement of these particles. In this case, a 4 pass scheme with final windows of size 16 x 16 x 16 voxels with 75% overlap was used. The volumes used, and the resultant vector field are shown in Figure 2.13 where each particle is shown as an iso-surface for frame A (Figure 2.13, *left*) and frame B (Figure 2.13, *middle*). The 3D vector field (Figure 2.13, *right*) is shown as a 3D iso-surface of vorticity magnitude (0.2

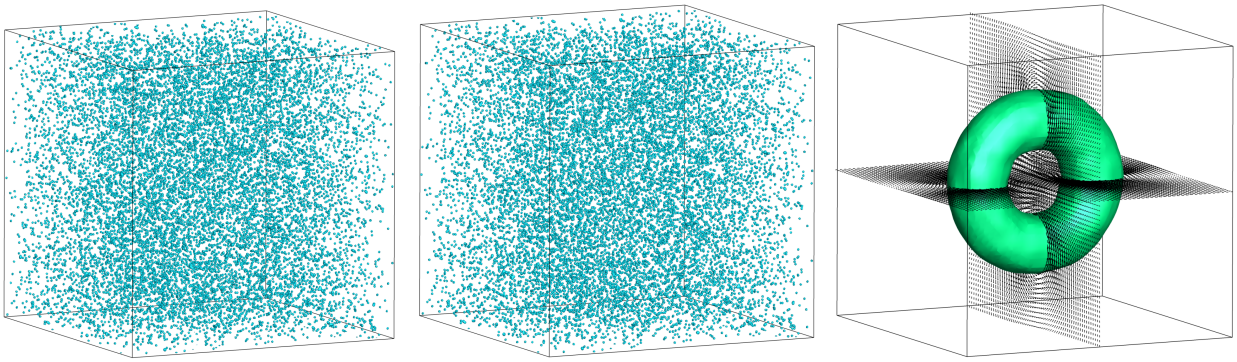


Figure 2.13: Artificial particle volumes frame A (*left*) and frame B (*middle*) processed with 3D VODIM algorithm (*right*) showing 3D iso-surface of vorticity magnitude (0.2 voxels/voxel) and two slices of velocity vectors.

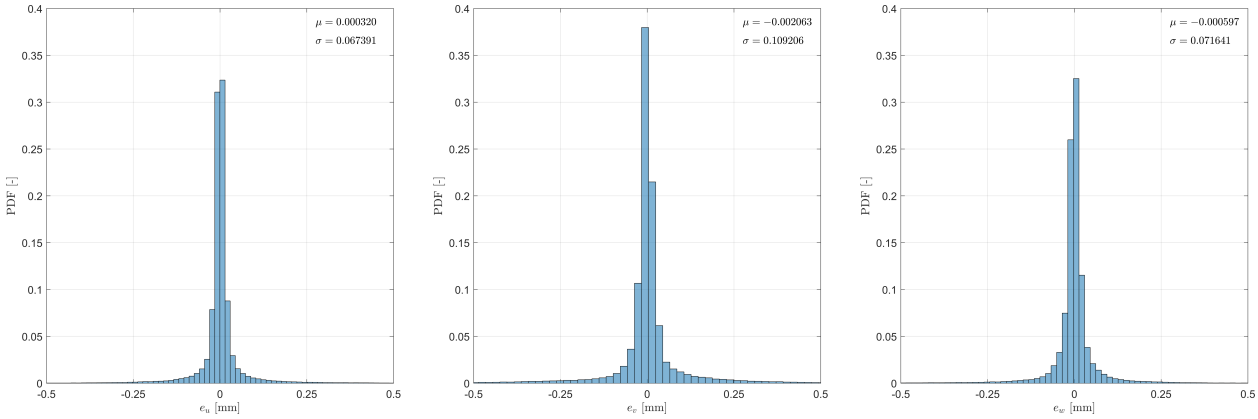


Figure 2.14: Histograms of displacement error between the 3D VODIM solution and the analytical solution for the Gaussian ring vortex. From left to right are the error in each of the velocity components e_u, e_v, and e_w respectively.

voxels/voxel) with two slices of velocity vectors. For comparison, the analytical displacement was calculated at each vector location, and the error between the analytical and measured displacement (e_u, e_v, e_w) was calculated and is shown in Figure 2.14. It is shown that the average error for each component was roughly zero (< 0.002 voxels) with uncertainties of 0.067, 0.109, and 0.071 voxels for u, v, and w respectively. These errors are at or below the expected value of 0.1-0.2 voxels [14].

Looking closer it can be shown that the areas of high error are concentrated around in areas of large velocity gradients (i.e. the core of the vortex ring). Figure 2.15 shows the displacement error extracted from slices of the volume presented as colormaps overlayed

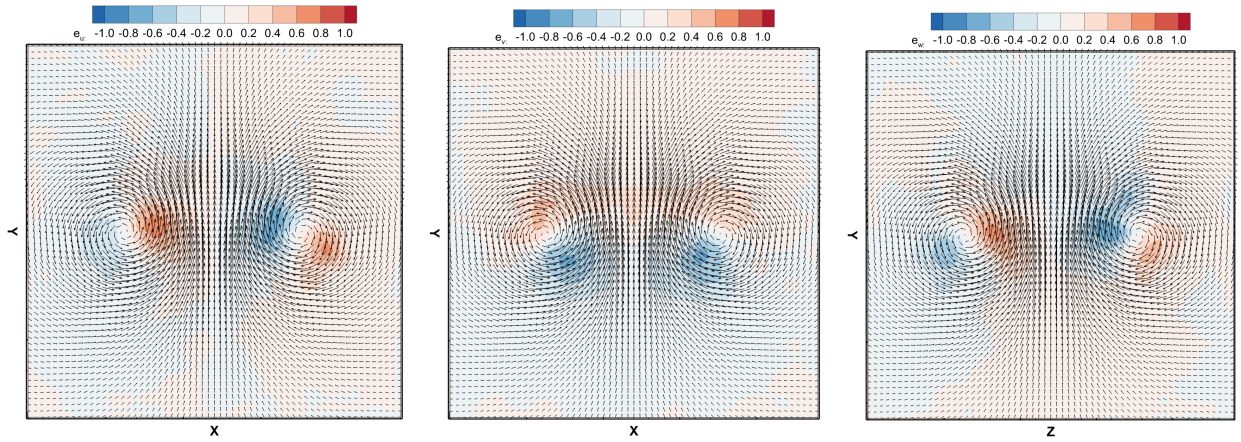


Figure 2.15: Slices from Gaussian ring vortex processed with 3D VODIM. Vectors show the measured velocity and colormap shows the error in each component u (*left*), v (*middle*), and w (*right*).

with the measured velocity vectors. It is shown the for each of the components in areas of constant velocity (i.e away from the vortex) the error is low, while in areas of high velocity gradients the error becomes larger. This is likely due the windowing effect of PIV where each vector is actually the average of its window.

2.8 Current 3D PIV techniques

Due to PIV being an image based technique, the measurements obtained have been traditionally limited to two dimensions. Consequently, traditional PIV is not capable of capturing the full three dimensional (3D), three component (3C) velocity field instantaneously, which is important for quantifying the topology and extent of flow structures which pervade most turbulent flows. Moreover, turbulence is inherently 3D in nature, and a full description requires a measurement of the 3D velocity field and derivative quantities such as the stress tensor and vorticity vector.

The desire to capture the full 3D/3C nature of turbulence has led to the development of several techniques with a visualization of the capabilities of these efforts shown in Figure 2.16. It is noted that advances such as stereoscopic-PIV (Arroyo and Greated [28], Willert [33]), as described in the previous section, as well as dual-plane stereoscopic PIV (Kähler

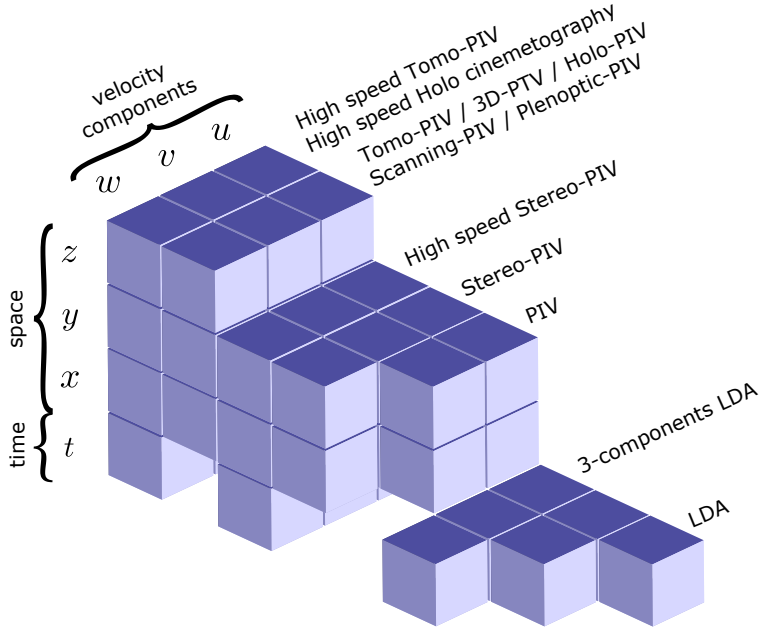


Figure 2.16: Measured components and domain of laser based velocimetry techniques (adapted from Scarano [4])

and Kompenhans [34]) capture the three components of velocity albeit in a 2D plane and are therefore not true 3D measurements. The remainder of this section will focus on five different 3D PIV techniques: scanning PIV (Brücker [35]), defocusing PIV (Willert and Gharib [33], Pereira *et al.* [36]), holographic PIV (Hinsch [37], Herrman and Hinsch [38]), tomographic PIV (Elsinga *et al.* [39]), and synthetic aperture PIV (Belden *et al.* [40]).

Scanning PIV extends traditional or stereoscopic PIV to three dimensions by recording and processing multiple planes of the traditional methods. The key to this technique is the use of high-repetition-rate lasers, high-frame-rate cameras, and high-speed motion-controlled mirrors to quickly scan and image a 2D light sheet at multiple locations in the third dimension. This is done, similar to a traditional PIV system, by forming a laser into a thin light sheet using appropriate optics; however, for scanning PIV, this light sheet will be deflected by an oscillating mirror such that each pulse will image a slightly different depth plane creating a 'stack' of image pairs which are then processed using traditional cross-correlation. If the images are sampled finely enough (observing the Nyquist criterion) the full 3D vector field can be reconstructed. This technique was demonstrated by Brücker [35] on the flow

around a short cylinder and to obtain the third component of velocity Brücker used the concept of continuity between neighboring slices to obtain the out-of-plane component. It is noted that if a stereoscopic PIV system is used the out-of-plane component could be directly measured. The fundamental limitation of this technique is the repetition/acquisition rate of the laser/camera system. The scanning system must be able to scan significantly faster than the characteristic time scales of the flow in order to 'freeze' the flow such that the first and last image in the scan are imaging the same instant in time. As laser technology, in particular repetition rate, improves this technique becomes more viable; however, even with kHz rate laser systems scanning PIV is typically limited to low speed applications. The use of MHz-rate laser systems (Lynch and Thurow [41], Thurow *et al.* [42]) has potential to improve scan rates; however, the complexity and expense of the laser and camera systems are currently too prohibitive for broad application.

Defocusing PIV attempts to image the 3D position of the particles directly by using a modified aperture. For the single camera configuration, presented by Willert and Gharib [33], a mask was placed in front of a conventional lens with 3 pinholes arranged in a triangular pattern. If a particle is located at the focal plane a single particle image is recorded; however, if the particle is located elsewhere three particle images are created forming an equilateral triangle. The 3D position of the particle can then be calculated based on the location of the encoded pattern on the image sensor as well as the distance between each of the particle images using simple geometry. The major benefit to this technique is the use of a single camera that can intrinsically measure the 3D location of particles, which drastically reduces the experimental complexity when compared to a multi-camera technique. One drawback is the use of pinhole apertures which significantly reduce the amount of light imaged by the camera and therefore increase the power requirement of the laser. In addition, since each particle is imaged as a particle triplet, the sensor can become overcrowded at higher particle densities making the triplets difficult to identify. A solution to this problem was given by Pereira *et al.* [36] where they used three separate lens/camera systems to image

the particle triplet pattern. Furthermore by using three cameras the parallax of the imaging system (separation of the pinhole apertures) was increased, increasing the accuracy of the 3D particle tracking. This system, however, has the obvious requirement of precise alignment of three camera systems which drastically increases the experimental complexity.

Holographic PIV utilizes the concept of optical holography to image the amplitude and phase of the scattered particle light (objective wave) when it is superimposed with a second wave termed the reference wave. This image, called an interference pattern or hologram, can then be used to reconstruct the particle field by illuminating it with a replica of the reference wave. The two main configurations for the recording of a hologram are the so called in-line holography (Sheng *et al.* [43]) and off-axis holography (Zhang *et al.* [44]). The fundamental trade-off between the two techniques are the amount of scattered light and the resolution of the interference pattern. For off-axis holography the camera is placed normal to the laser generating an objective wave via side scattering. In contrast, in-line holography places the camera, as its name suggests, in line with the laser such that the object wave is created using forward scattering. Since PIV particles are used, which scatter light in the Mie regime, in-line holography is able to capture a significant amount more light than the off-axis configuration. The main benefit to off-axis holography is the increased resolution due to increased angular range of the recorded light. Historically, holographic PIV has been performed using photographic plates due to their high resolution when compared to digital CCD/CMOS cameras. In particular, when a hologram is created in the off-axis configuration the narrow fringe spacing requires photographic plates. As such, the move to digital cameras has focused on the so called digital in-line holography (DiH) method. In particular, the work of Sheng *et al.* [43] is noted for the development of a DiH microscope for 3D particle tracking. Due to the resolution requirements to resolve the fringe patterns, holographic PIV is typically limited to small-scale laboratory experiments.

Tomographic PIV represents the current standard of the 3D PIV techniques. From its inception (Elsinga *et al.* [39]) it has seen rapid development and widespread adoption in the

scientific community. A more complete survey of the technique is given by Scarano [4] where numerous experimental studies are presented. Briefly, tomographic PIV is a technique that utilizes 3 or more (typically 4) standard CCD/CMOS PIV cameras to image a very thick laser sheet ($\sim$ 10 mm). Then using tomographic reconstruction the set of images corresponding to the two distinct instances in time are reconstructed into two discretized intensity fields. These reconstructed volumes are then processed using 3D cross-correlation algorithms resulting in a 3D/3C vector field. Much of the current work is focused on using kHz rate cameras to obtain sequences of time-resolved images. Using this additional information (the knowledge of particles over time) the accuracy of the technique can be improved through changes to the tomographic reconstruction algorithms (Novara *et al.* [45], Lynch and Scarano [46]) or by directly tracking their motion throughout the sequence (Schanz *et al.* [47]). The major limitation of this technique are the large optical access needed to image with four cameras at large angular separation, high laser power due to the use of small apertures, complexity of the experimental arrangement, and the overall expense of the system. Nonetheless, tomo-PIVs success in obtaining 3D/3C velocity measurements in a multitude of facilities is notable and has revitalized recent research in 3D flow diagnostics.

Synthetic aperture PIV (SAPIV) is a multi-camera technique based on similar principles (light field imaging) to the work presented here. Typical applications of light field imaging use a dense sampling of the angular space to reconstruction the light field (in this case the scattered light from particles). To do this a large multi-camera array (Belden *et al.* [40] uses 8 cameras) is used. In contrast, the work presented here utilizes plenoptic cameras which are able to densely sample the angular space ($\sim$ 100 views) albeit with a much smaller angular range. In SAPIV, the reconstructed particle volumes are computed using the map-shift-average algorithm which allows for the particle images to be computationally refocused throughout the volume. Since in-focus particles form sharp peaks and out-of-focus particles blur over multiple voxels, a simple intensity threshold can be applied to remove the out-of-focus particles. The reconstruction can then be processed using cross-correlation yielding a

3D/3C vector field. The number of cameras and their associated experimental complexity (alignment, optical access, cost, etc.) is the main limitation of this technique.

Chapter 3

Light Field Imaging

The field of light field imaging has experienced significant growth over the last couple decades and has evolved into a rich and active area of research. In this chapter, a basic overview of the history and fundamental concepts of light field imaging are presented; however, the reader is encouraged to consult other sources, such as Adelson *et al.* [48, 49], Levoy *et al.* [50, 5], Ng *et al.* [6], and Lumsdaine and Georgiev [51], for more detailed information.

Historically, the notion of a light field is over a century old with its roots outlined in Lippman [52]. The modern definition of a light field comes from Adelson and Bergen [48] where space is described as being filled with a dense array of light rays of varying intensities. These light rays contain information about our world and can be described in a systematic manner using the plenoptic function. The plenoptic function refers to the parameterization of the light field, where each light ray is represented by its 3D position in space (x, y, z) and its angle of propagation (θ, ϕ), thus forming a 5D function [1] representing all light rays traveling through space. Assuming constant intensity, or more precisely irradiance, of a light ray along its path of propagation, the plenoptic function is typically reduced to a 4D function, denoted as $L(x, y, \theta, \phi)$. In this context, a conventional photograph or image can be thought of as a 2D projection of the 4D light field where the angular dimensions have been integrated out at the sensor plane.

Adelson and Wang [49] utilized this concept to estimate the depth and shape of objects by measuring the plenoptic function with a single camera, referred to as a plenoptic camera.

[1] In a general sense, one can also include the wavelength, polarization and time dependency of light in space such that the full light field may be considered as an 8D function. This is known as the radiance function.

The camera utilized a specialized optical design to encode both the spatial (x, y) and angular (θ, ϕ) components of the light field onto a 2D image sensor. In a conventional camera, a main lens collects light across a range of input angles bounded by the size of the aperture and focuses the light directly onto the image sensor, which records the total light intensity at each pixel regardless of the angle of incidence. In contrast, in a plenoptic camera the main lens focuses the entire angular distribution of light onto an array of microlenses. Each microlens covers a small number of pixels on the image sensor and can be thought of as forming a macropixel. In this configuration, the microlenses capture the spatial information contained in the light field, while the pixels contained under the microlens record the angular distribution. This relationship will be described in greater detail later in this chapter. Adelson and Wang's version of the plenoptic camera utilized a 500 x 500 pixel CCD with a microlens array of 100 x 100 microlenses. This results in a camera with a spatial resolution of 100 x 100 pixels with an angular sampling of 5 x 5.

Capturing and altering the light field is not limited to using a plenoptic camera. Levoy [50, 5] describes several methods of obtaining the light field in order to computationally generate an image or rendering of an object. One method places the object of interest at the center of a sphere, then, using a spherical gantry, thousands of images can be taken at different positions along the spheres surface. The resulting collection of 2D images taken at discrete angles is a representation of the 4D light field. Another method is to mount multiple cameras, Levoy [5] used 128, in an array allowing an instantaneous light field to be acquired. These techniques utilize multiple 2D images to build the 4D light field. In this vein, it is noted that defocusing-PIV, tomo-PIV, and SAPIV are implicitly measuring the light field, albeit with relatively low angular resolution. In contrast, the plenoptic camera directly captures the 4D light field on a single image sensor in a single snapshot, with a fairly dense angular sampling over a limited angular range.

Figure 3.1: Computationally refocused images generated from a single exposure, focused: (*left*) on an alarm clock that is in front of the nominal focal plane, (*center*) at the nominal focal plane, and (*right*) on a student behind the nominal focal plane.

As camera and microlens technology has improved, the interest in plenoptic cameras has grown. Of the more recent developments the work of Ng [53, 54] who designed a hand-held plenoptic camera for digital photography is of particular note. The camera consisted of a modified DSLR with a 16 megapixel image sensor and a microlens array of 296 x 296 microlenses. Ng's research focused on computationally rendering conventional 2D images from the light field data collected by the plenoptic camera in a single snapshot. They demonstrated the ability to computationally generate, after the fact, photographs with a different focal position or a shift in the perspective. Examples of refocused images acquired with our plenoptic camera (described later) are shown in Figure 3.1. The three images represent the focus shifted toward the camera, stationary, and shifted away from the camera relative to the nominal focal plane. In Figure 3.2, the perspective of the observer is shifted with one image showing a left view and the other showing a right view. These images serve to illustrate the unique information obtained by a plenoptic camera and how it can be used for computational imaging. Recently, commercial variants of plenoptic cameras have become available. For consumer photography, Lytro (Founded by Ng) offers a point-and-shoot plenoptic camera with built in refocusing capabilities. In the field of machine vision Raytrix offers a plenoptic 2.0 camera that offers a similar ability to change the perspective of an image after the fact.

More recently, Levoy et al. [55, 56] has developed a light field microscope based on the plenoptic camera. The fundamental principle remains the same; however, their work

Figure 3.2: Computationally rendered image where the viewpoint of the observer has been changed to (*left*) the left side of the aperture and (*right*) the right side of the aperture.

focused on additional challenges associated with microscopic imaging. For one, wave optics and diffraction must be considered in a microscopic environment whereas geometrical optics is sufficient for macroscopic imaging. In addition, a typical microscope objective functions differently than a normal camera lens, producing orthographic rather than perspective views. Next, most objects in microscope images are partially transparent whereas the previous effort had focused on scenes with opaque objects.

3.1 The Plenoptic Camera

As alluded to earlier, the plenoptic camera differs from a conventional camera with its ability to not only capture the spatial information about a scene, but also capture the angular information. In a conventional camera this information is integrated at the image sensor. This is shown schematically in Figure 3.3. The left most section of the figure shows light emanating from a point (x, z) on the nominal focal plane of the camera. An objective lens captures this light and focuses it, for the conventional camera, onto an image sensor, which then records a single value. Thus, the angular information is integrated and therefore lost. In contrast, for a plenoptic camera, the objective lens focuses the incoming light onto a microlens array. This microlens array then distributes the light onto several pixels on the

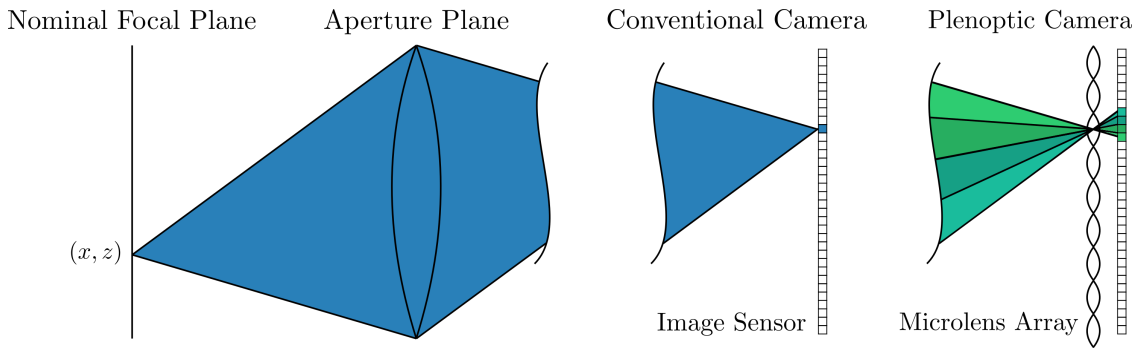


Figure 3.3: Illustration of the differences between a conventional camera and a plenoptic camera in how they sample the light field.

image sensor. As indicated by the different colors, each pixel represents a different section of the angular distribution.

3.1.1 Prototype Camera

As part of the development of the plenoptic PIV technique, a prototype plenoptic camera was constructed using an Imperx Bobcat ICL-4820 camera, which uses a Kodak KAI-16000 image sensor, that was the highest resolution interline CCD available at the time of construction. The microlens array was fabricated by Adaptive Optics Associates, a subsidiary of Northrup Grumman. Specifically, the microlenses are manufactured using a proprietary process where an epoxy filled mold is used to print the microlenses onto the glass surface. The primary challenge faced in constructing the prototype camera was fabricating a custom mounting device for the microlens array to position it accurately over the sensor. A custom mount was designed by Light Capture, Inc. and manufactured in-house. The mount consists of a series of positioning screws to adjust the height of the microlens array above the sensor and to adjust the orientation of the array with respect to the sensor. Due to the availability of a higher resolution interline image sensor, a second generation camera was designed and constructed. This camera was based on the Imperx Bobcat B6620 camera, which uses a Kodak KAI-29050 image sensor. The microlens array is manufactured by Jenoptik Inc using gray-scale photo-lithography on a fused silica substrate and was mounted using a modified

Table 3.1: List of fixed (unmodifiable once constructed) plenoptic camera parameters.

Parameter	Symbol	16 Mp	29 Mp
Microlens Array Layout	-	Rectangular	Hexagonal
Microlens Pitch (flat to flat)	p_μ	0.125 mm	0.077 mm
Microlens Focal Length	f_μ	0.500 mm	0.308 mm
Number of Microlenses: X-direction	n_{μ_x}	289	471
Number of Microlenses: Y-direction	n_{μ_y}	193	362
Pixel Pitch	p_p	0.0074 mm	0.0055 mm
Number of Pixels: X-direction	n_{p_x}	4820	6600
Number of Pixels: Y-direction	n_{p_y}	3280	4400
Microlens Array Material	-	BK7/Epoxy	Fused silica

version the original microlens mount. In this camera the microlenses are packed hexagonally instead of rectilinearly to decrease the wasted pixels in between the sub-aperture images. The full list of the camera parameters for the both the original (16 Mp) and second generation (29 Mp) cameras are listed in Table 3.1.

3.2 Building the Light Field

As mentioned earlier, the information stored in the light field recorded by the plenoptic camera can be manipulated in post-processing to create synthetic images whose focus or perspective has been changed. In order to do this the raw image data must be molded into a more computationally friendly format. This process, termed *building the light field* requires a precise calibration of the microlens and pixel positions and is detailed herein.

3.2.1 Two-Plane Parameterization

The preceding discussion parameterizes a light ray by its position on the world focal plane and angle of propagation. An alternative, and often times more convenient, way to parameterize the light field is known as the two-plane parameterization. The discussion herein is derived from Levoy [50]. Figure 3.4 (*left*) describes a light ray by its position (x, z) and its angle of propagation θ. Figure 3.4 (*right*) shows a light ray that is defined by pairs of points, u and s, located on two planes separated by a known distance. These two

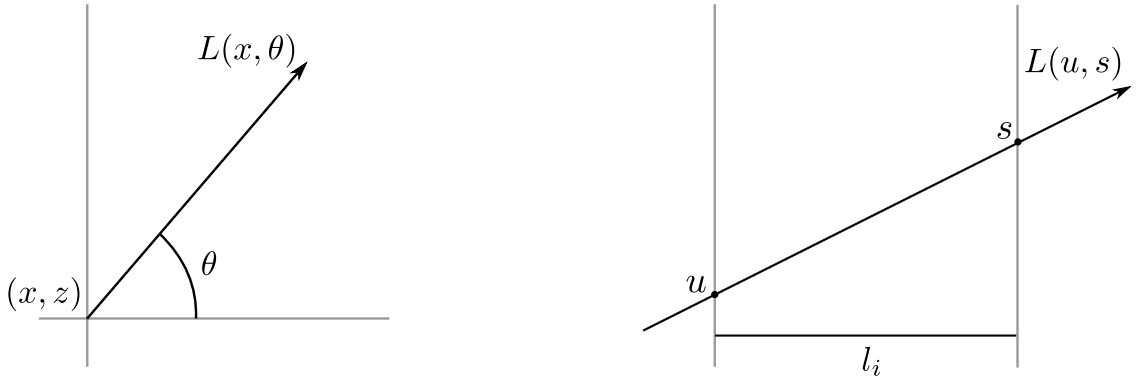


Figure 3.4: Two geometric representations of a light ray. The first parameterizes the light ray by its position and angle of propagation (*left*) and the second parameterizes the same light ray by a pair of points on two planes (*right*). Adapted from Levoy [5]

descriptions of the light ray are equivalent, since they can be derived from each other using simple trigonometric relations.

The plenoptic camera lends itself to this type of parameterization due to it inherently having two primary planes that light rays intersect: the microlens plane and the aperture plane, separated by a fixed distance, l_i. As discussed previously, the microlenses are responsible for discretizing the spatial location of all incoming light rays. The second plane, the aperture, represents the angular information where each microlens is effectively forming an image of the aperture on the image sensor. Therefore each pixel of the image sensor is associated with a discretized point on the microlens plane (s, t coordinate) as well as a point on the aperture plane (the u, v coordinate) separated by the image distance of the main lens.

The two-plane parameterization offers a more straightforward and convenient representation of the light field as the upper and lower bounds of the aperture plane are fixed and constant for every microlens. This is in contrast to the angular parameterization, where the range of sampled angles varies with each microlens.

3.2.2 Microlens Registration

Using the aforementioned two-plane parameterization, the recorded light field can be fully described through determination of the (u, v, s, t) position of each pixel. For experimentally obtained images, the exact locations of the microlenses relative to the image sensor are not known. As such, a registration procedure was developed to determine the positions of the microlenses, and the pixels beneath them. This procedure begins by taking a registration image. This image is obtained by minimizing the aperture of the camera (i.e. increasing the f-stop to its maximum value) and imaging a uniformly illuminated white surface, such as a piece of paper, while keeping the focal position of the camera constant. The last statement is very important as the positions of the microlens images on the CCD shift depending on the main lens configuration. A sample registration image is shown in Figure 7, *left*. The white dots are the centers of the reduced aperture image formed by each microlens. In terms of the two-plane parameterization these dots represent the center of the aperture (u_0, v_0, s, t). Since the aperture is not closed to a perfect point and the center of a microlens may not fall directly on a single pixel, the exact location of each microlens is calculated to sub-pixel accuracy using a simple centroid fit. An example of the centroid fit is shown in Figure 7, *right* where the center of each group of pixels is shown as a green "x".

This registration procedure results in a list of microlens positions in image coordinates. For ease of calculation these are converted to physical units using a priori knowledge of the microlens array, specifically the layout of the array (rectangular or hexagonal) and the microlens pitch. For this conversion the center of the CCD is considered to be the origin of the coordinate system. Mathematically, the conversion is given in equations 3.1 and 3.2.

$$s = s_{pix} \cdot p_p - 0.5(n_{p_x} \cdot p_p) \tag{3.1}$$

$$t = t_{pix} \cdot p_p - 0.5(n_{p_y} \cdot p_p) \tag{3.2}$$

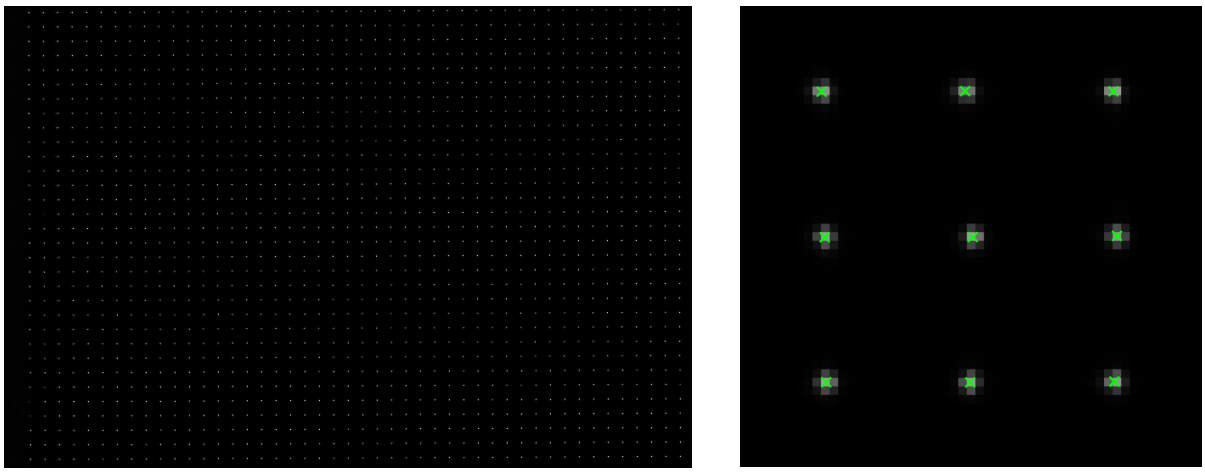


Figure 3.5: Subset of an experimental registration image (*left*) and corresponding centroid fit (*right*).

3.2.3 Pixel-Aperture Registration

Corresponding to each microlens (s, t) there are an associated set of pixels beneath it. Using the two-plane parameterization each of these pixels represents light that entered the camera through a specific section of the aperture. Determining the exact location of this section for each pixel requires knowledge of the image distance. The image distance can be determined from the nominal magnification of the imaging system as well as the focal length of the main lens. Experimentally, this can be calculated by simply imaging a ruler that is located at the focal plane. Then using the definition of the magnification and the thin lens equation the image distance can be determined. Using similar triangles, the distance between each pixel's position and the center of the microlens can then be converted to a position on the aperture plane. This assumes that the origin of the (u, v) coordinate system is the center of the main lens. This conversion is given in equations 3.3 and 3.4 where the subscript i represents the current pixel.

$$u_i = (s - x_i)\frac{p_p l_i}{f_\mu} \tag{3.3}$$

$$v_i = (t - y_i)\frac{p_p l_i}{f_\mu} \tag{3.4}$$

Since the microlens spacing does not necessarily line up with the image sensor (i.e. one microlens is not an integer number of pixels in size) each set of pixels beneath different microlenses image different chunks of the aperture, making interpolation more complicated. Therefore the (u, v) data will be resampled onto a uniform grid bounded by the minimum and maximum u's and v's imaged by any pixel. This is done using a simple bilinear interpolation on each set of pixels behind each microlens. At this point the building of the 4D light field data structure is complete.

3.3 Image Synthesis

The images shown in Figures 3.1 and 3.2 are illustrations of synthetically generating images from a recorded light field. Mathematically, these images are the result of simulating the imaging process on a synthetic light field, L', which is defined by a synthetic aperture plane (u', v') and a synthetic film plane (s', t'). This can be expressed as

$$E(s', t') = \iint L'(u', v', s', t')A(u', v')\ du\ dv \tag{3.5}$$

where $A(u', v')$ is an aperture function (1 inside the aperture, 0 otherwise). In order to create an actual image, however, this synthetic light field must be written in terms of the recorded light field. The relationship between L' and L is shown in Figure 3.6 where α and β are implicitly defined and γ and δ are defined as

$$\gamma = \frac{\alpha + \beta - 1}{\beta} \tag{3.6}$$

$$\delta = \frac{\alpha + \beta - 1}{\alpha} \tag{3.7}$$

The blue line in this schematic is representative of a single light ray, which is defined by its position on the synthetic aperture plane (u', v'), its position on the synthetic film plane (s', t'), and the distance seperating them $(\alpha + \beta + 1)l_i$. Projecting the ray onto the recorded

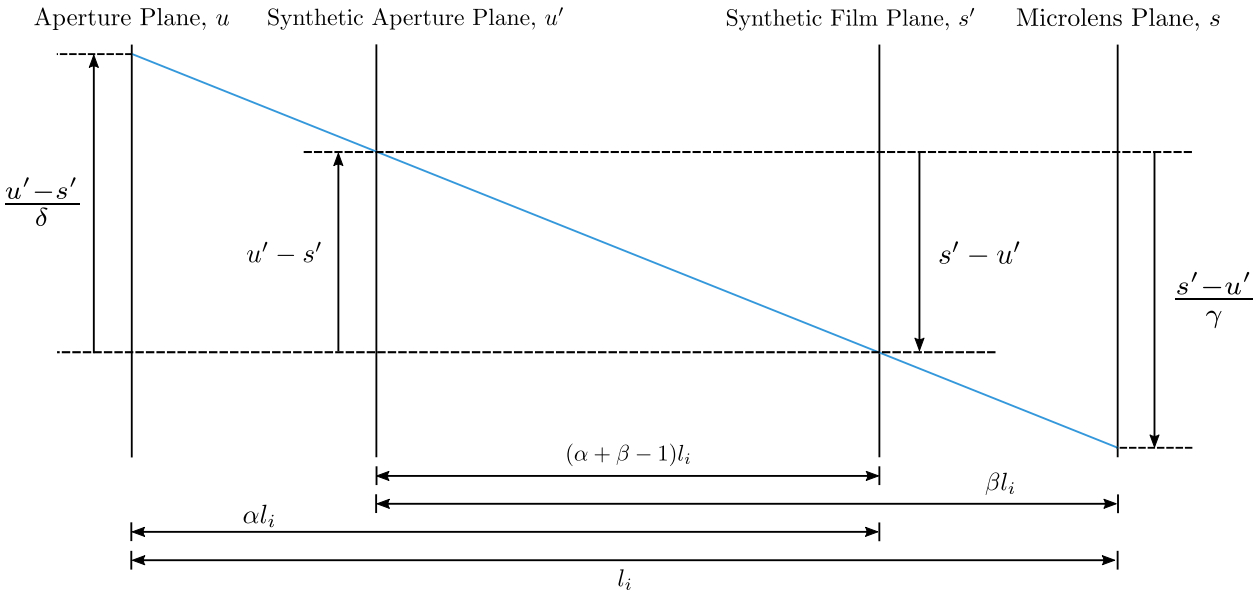


Figure 3.6: Schematic showing the relationship between the desired synthetic light field L' and the recorded light field L. Adapted from Ng *et al.* [6]

aperture and film planes allows for the synthetic light ray $L'(u', v', s', t')$ to be rewritten as

$$L'(u', v', s', t') = L\left(s' + \frac{u'-s'}{\delta}, t' + \frac{v'-t'}{\delta}, u' + \frac{s'-u'}{\gamma}, v' + \frac{t'-v'}{\gamma}\right) \tag{3.8}$$

Rewriting equation 3.5 in terms of the recorded light field yields the synthetic imaging equation:

$$E(s', t') = \iint L\left(s' + \frac{u'-s'}{\delta}, t' + \frac{v'-t'}{\delta}, u' + \frac{s'-u'}{\gamma}, v' + \frac{t'-v'}{\gamma}\right) A(u', v')\, du\, dv \tag{3.9}$$

Computationally probing the recorded light field at arbitrary locations requires interpolation since the original light field was recorded at discrete locations. To make this process easier the 4D interpolation procedure is broken up into two 2D interpolation steps. Given an arbitrary light ray, (u_2, v_2, s_2, t_2), where the subscript 2 refers to the point of interpolation in interpolation space, the contribution of the neighboring microlenses (s, t) is considered first and is separated due to the different microlens layouts (hexagonal or rectangular). The rectangular layout will be used here for illustration; however, the formulation required for

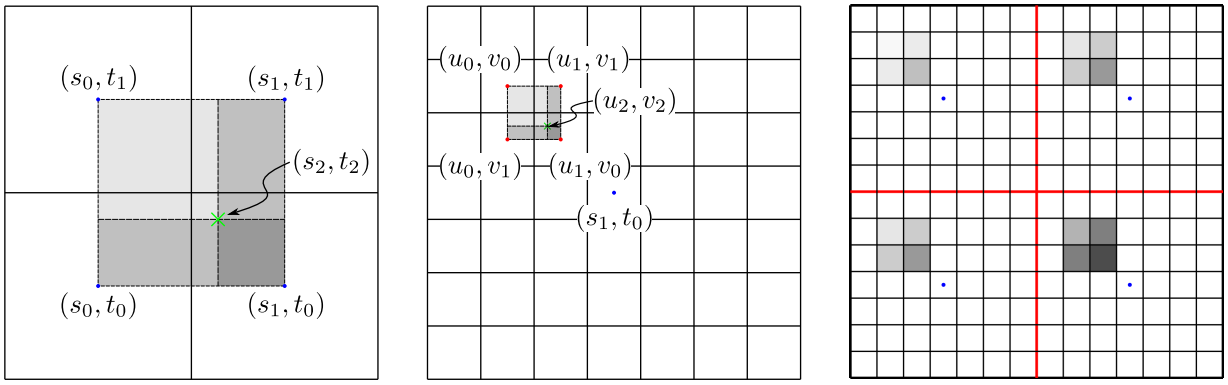


Figure 3.7: Interpolation of the light field data structure. (*left*) 2D bilinear interpolation on rectilinear (s, t) data. (*center*) 2D bilinear interpolation on (u, v) data. (*right*) Illustration of 4D interpolation for the rectangular microlens array.

the hexagonal layout is given in Appendix 7. The second interpolation is on the (u, v) space which is independent of the layout of the microlens array. This process will result in 16 coefficients that are used to interpolate the irradiance of the light ray from the measured light field.

First, we consider the intersection of the light ray with the (s, t) plane to determine the distribution of the light ray on the nearest four microlenses. This is represented schematically in Figure 3.7 (*left*) where the green "x" is the point where the projection strikes the microlens plane, the blue dots represent the center of each microlens, and the shaded area enclosed by the dotted lines is the interpolation domain. In this representation, each ray is implicitly assumed to have a finite width equal to the size of one microlens, which is consistent with the physical function of the microlenses within the camera. The surrounding microlens positions are determined, in microlens coordinates, by using the floor and ceiling operators, where the subscript 0 is associated with the floor operator and the subscript 1 with the ceiling operator. This allows the relative position of the light ray to the neighboring microlens centers to be easily calculated, and it has the benefit of auto-normalizing the coefficient since the separation is equal to one (i.e. $\text{ceil}(s_2) - \text{floor}(s_2) = 1$). Once the interpolation coefficient for the four microlenses have been calculated the u, v interpolation can take place.

Figure 3.7 (*center*) shows the discretization of the aperture plane as viewed from the pixel behind microlens (s_1, t_0). The green "x" refers to where the projection strikes the aperture plane. The red dots represent the centers of each (u, v) location on the aperture. As with the (s, t) interpolation (u_2, v_2) is expressed in terms of pseudo-pixel coordinates using the floor/ceiling operators.

Once the sixteen locations for which we need to calculate a coefficient have been found, the value of the coefficient must be determined. To do this we employ a simple linear interpolation scheme in which the coefficient is a combined value of the (s, t) and (u, v) interpolation steps. The distance from the $(0, 0)$ point in both interpolation schemes is all that is needed to calculate the coefficient. The relative distances, d, are given by

$$d_s = s_2 - s_0 \quad d_t = t_2 - t_0 \quad d_u = u_2 - u_0 \quad d_v = v_2 - v_0 \tag{3.10}$$

Using these and simple geometry the sixteen coefficients can be calculated. The interpolation coefficients, N_{uvst}, have subscripts that represent their location relative to the voxel to be interpolated. For example N_{0000}, is the coefficient for point (u_0, v_0, s_0, t_0). The coefficients are calculated by using the normalized distances and are shown to be

$$\begin{aligned} N_{0000} &= (1 - d_u)(1 - d_v)(1 - d_s)(1 - d_t) \\ N_{0001} &= (1 - d_u)(1 - d_v)(1 - d_s)(d_t) \\ &\vdots \\ N_{1111} &= (d_u)(d_v)(d_s)(d_t) \end{aligned} \tag{3.11}$$

The result of this procedure can be seen in Figure 3.7 (*right*) where the red boarder represents the four microlenses shown in Figure 3.7 (*left*) with the (u, v) distribution behind it. The sixteen interpolation coefficients are shown as the shaded squares with intensity depending on their weight (white = 0, black = 1). In other words, Figure 3.7 (*right*) shows the relative

distribution of intensity on the image sensor that results from a single light ray. The final irradiance value associated with the light ray would be then be determined by

$$L'(u', v', s', t') = \sum_{u=u_0}^{u_1} \sum_{v=v_0}^{v_1} \sum_{s=s_0}^{s_1} \sum_{t=t_0}^{t_1} N_{uvst} \cdot L(u, v, s, t) \tag{3.12}$$

3.3.1 Computational Refocusing

A simple introduction into manipulating a light field is to resample the light field at a new synthetic focal plane. This process, termed computational refocusing, has been adapted from the work of Ng [53] and relies on the two-plane parameterization of the light field. For computational refocusing the synthetic imaging equation is restricted by only moving the synthetic film plane (i.e. $\beta = 1$) and using the full aperture (i.e $A(u', v') = 1$). The composite terms γ and δ then reduce to $\gamma = \alpha$ and $\delta = 1$, which simplifies the synthetic imaging equation to the synthetic refocusing equation

$$E(s', t') = \iint L\left(u, v, u\left(1 - \frac{1}{\alpha}\right) + \frac{s'}{\alpha}, v\left(1 - \frac{1}{\alpha}\right) + \frac{t'}{\alpha}\right)\, du\, dv \tag{3.13}$$

A simplified version of the light field representation is shown in Figure 3.6 is shown in Figure 3.8. This schematic shows the rendering of a synthetically refocused image at s'. Each light-ray (blue line) is projected from a position on the aperture plane u through s' and onto the original film plane s. Then using the interpolation procedure detailed earlier, the light ray is assigned an irradiance value. For the point (s', α) the final intensity is the sum of all the projections, whose intensity is indicated by $L(u, s')$. Therefore if each of the gray values indicates a value of one, and the white values hold a value of zero, the final intensity of (s', α) would be seven. This process is repeated for every (s', t') location to create the full image.

An additional consideration is how the aperture plane (u, v) is sampled. The simplest method would be to use the same number of samples as the original recording (number

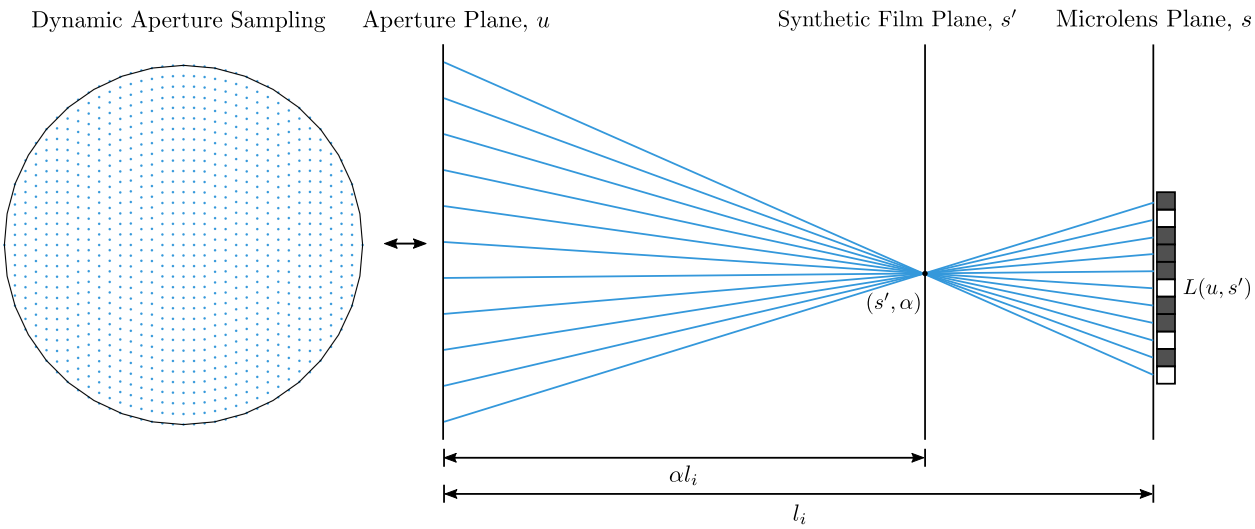


Figure 3.8: Schematic showing projection of x' and u onto original light field.

of pixels beneath a microlens). This method, however, fails at large distances away from the original film plane due to undersampling of the (s, t) plane. In essence, the distance between two neighboring projections becomes greater than a single microlens which results in a loss of information, and creates banding in the image. To overcome this an algorithm has been developed to dynamically super-sample the aperture ensuring that the maximum difference between two neighboring projections is less than a microlens (shown in Figure 3.8). In addition, the algorithm does not sample values outside the circular aperture. More information on this algorithm is available in Appendix 8.

3.3.2 Computational Perspective Generation

Another benefit of capturing the entire light field, is the ability to change perspective of the scene, or in other words to change the angle at which the scene is observed. These images are generated by only considering a single angle (i.e. aperture position) in the light field. Similar to the refocused image, a single value is used to represent a microlens however, instead of summing the angular information into a single value, a specific angle (u, v) is chosen and that value is used. Mathematically, the synthetic imaging is restricted by only considering a single angle (i.e. we do not integrate over the aperture). As such the position of

the focal plane is irrelevant and we therefore set $\alpha = 1$. The resulting synthetic perspective equation is

$$E(s',t') = L\left(s' + \frac{u'-s'}{\beta}, t' + \frac{v'-t'}{\beta}, s', t'\right) \tag{3.14}$$

For the work presented here, $\beta = 1$ is chosen such that we are sampling the original aperture plane. Therefore, we can generate perspectives where the viewer is located at different points across the aperture. Some sample images of this effect are shown in Figure 3.2.

3.4 Particle Image Simulation

While the previous discussion about manipulating the light field is useful for understanding its unique capabilities, it does not directly apply to 3D fluid velocimetry measurements. To develop this technique synthetic data is needed to test the overall accuracy of the particle reconstruction algorithms. Specifically, synthetic data allows the reconstructed volumes to be compared against a known solution, whereas experimental data does not allow for such a comparison. To do this a plenoptic camera simulator has been developed and is detailed herein.

As mentioned previously the optical configuration for a plenoptic camera differs from a conventional camera with the addition of a microlens array. In order to construct the simulator, the following variables and relationships are defined in Figure 3.9. Due to the nature of the simulation all parameters are measured relative to the optical axis in both the x and y directions. The origin of the z axis is defined at the nominal focal plane of the camera with positive z pointing away from the camera.

Particle positions are defined by their position relative to the center of a volume positioned at the nominal focal plane of the main lens, where the main lens is modeled as a thin lens with focal length, f_m, and an aperture with diameter, d_a. Similarly, the microlenses are defined by their focal length, f_μ, and pitch, p_μ. The physical image sensor is defined by a pixel pitch, p_p, which denotes the size of a pixel. The distances separating the elements are

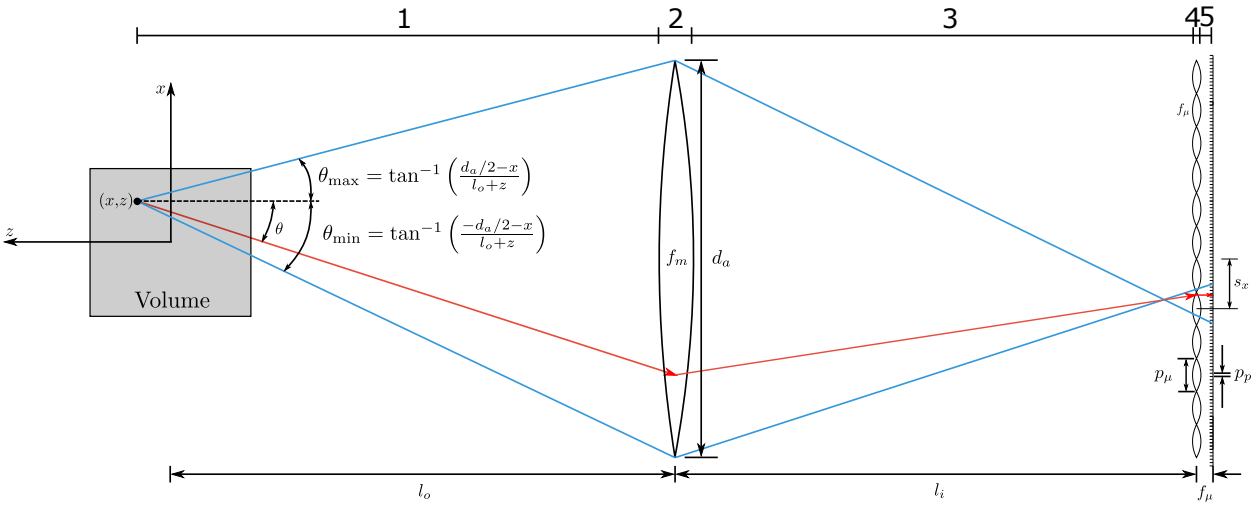


Figure 3.9: Schematic of ray-tracing process for a plenoptic camera.

the object distance, l_o, which separates the focal plane of the camera and the main lens and the image distance, l_i, which separates the main lens and the microlens array. The image and object distances are related by the thin lens equation, shown in equation 3.15, which makes the assumption that the thickness of the lens is negligible relative to the length of the optical system itself.

$$\frac{1}{l_i} + \frac{1}{l_o} = \frac{1}{f_m} \tag{3.15}$$

We note that modern camera lenses, which typically contain multiple lens elements, can be approximated by a thin lens where l_i and l_o are measured relative to the principle planes of the lens. While not considered here, the present framework also allows for more detailed modeling of these additional lens elements. l_i and l_o are related to the magnification of the imaging system through equation 3.16.

$$M = -\frac{h_i}{h_o} = \frac{l_i}{l_o} \tag{3.16}$$

In combination with eq. 3.15, this equation allows for the calculation of l_i and l_o knowing only the magnification, which can be obtained by imaging a ruler, and focal length of the main lens.

The optical parameters are now divided into two categories: variable and fixed parameters. The variable parameters can change with each experiment and include the main lens focal length, aperture diameter, and magnification. The object and image distances are also variable, however, as shown previously, they are dependent on the main lens focal length and magnification. The second class of parameters are set through hardware design and cannot be modified once the camera has been assembled. These include microlens pitch, microlens focal length, pixel pitch, and the number of pixels. These parameters can be modified in the simulator to accommodate testing and camera design, but are not varied in this work.

One consequence of the microlens parameters being fixed is a forced condition known as f-number matching. This condition, recognized by Ng *et al.* [53], states that the image-side f-number of the main lens must be equal to or greater than the f-number of the microlenses. This condition prevents any overlap between adjacent microlens images which would otherwise cause ambiguity in the light field parameterization. The equation for calculating the image-side f-number, as described by Ng [6], is shown in equation 3.17 where, f is the focal length, and $f/\#$ is the f-number, which is defined as the focal length divided by the size of the aperture.

$$(f/\#)_m = (f/\#)_\mu/(1 - M) \tag{3.17}$$

In this work, we simulate a nominal 1:1 imaging magnification such that $h_i = h_o$ and $M = -1$. In the future, the parameterization of the plenoptic camera as a function of magnification needs to be considered, however in order to keep the number of variables used in this work manageable only a single magnification is used. The fixed parameters used in the present simulation are shown in Table 3.1 and are used throughout this work unless otherwise noted. The input or variable parameters used throughout this work are shown in Table 3.2.

In this regards, it is worth commenting that the degree of parallax observed in the perspective views is limited by the size of the lens aperture used to form the image and

Table 3.2: Variable parameters for plenoptic camera simulation.

Parameter	Symbol	Value
Main Lens Focal Length	f_m	50 mm
Main Lens F-number	$(f/\#)_m$	2
Magnification	M	-1
Object Distance	l_o	100 mm
Image Distance	l_i	100 mm

the objects location relative to the main lens. Ultimately, the aperture size is limited by the requirement that the f-number of the microlenses must be matched to the image-side f-number of the main imaging lens. In the work described herein, we focus on 1:1 imaging with $f/4$ microlenses. Under these conditions, the f-stop of the imaging lens is set to $f/2$ with a nominal working distance equal to $2f$ (i.e. 1:1 magnification is achieved at a working distance equal to twice the focal length of the imaging lens).

The process of the ray-tracing simulation is shown schematically in Figure 3.9. For each synthetically generated particle, represented as a point source located at (x, y, z), a large number of rays (typically $>$ 10,000) are used to simulate the light emanating from that point. Each light ray is given an initial position, determined from the particles location, as well as an initial angle. The angle is generated as a random number between θ_{min} and θ_{max}, which are determined based on the distance to the lens and the aperture size. In Figure 3.9 the maximum angles are shown as the outermost blue rays, and the expressions for the maximum and minimum angles are given. From this initial state the ray is propagated to the main lens, labeled as 1. This is done using the simple trigonometric relations given by

$$x_1 = x + (l_o + z) * \tan\theta \tag{3.18}$$

$$y_1 = y + (l_o + z) * \tan\phi \tag{3.19}$$

Then the ray's angle needs to be deflected at the main lens, labeled as 2. To do this the thin lens approximation is strictly enforced, such that the analysis in the previous chapter

is held true. The new angles of propagation are given by

$$\theta_2 = \tan^{-1}\left(\frac{x_1 - x}{l_o + z} - \frac{x_1}{f_m}\right) \tag{3.20}$$

$$\phi_2 = \tan^{-1}\left(\frac{y_1 - y}{l_o + z} - \frac{y_1}{f_m}\right) \tag{3.21}$$

The ray is then propagated to the microlens array, labeled as 3, using the following trigonometric relations

$$x_3 = x_1 + l_i * \tan\theta_2 \tag{3.22}$$

$$y_3 = y_1 + l_i * \tan\phi_2 \tag{3.23}$$

Once the ray is at the microlens plane, the particular microlens struck by the light ray needs to be determined. Then the distance from the optical axis of the main lens to the optical axis of the microlens, termed s_x, can be determined. The reference frame can then be shifted to the optical axis of the microlens for the calculation of the deflection of the ray's angles. The new angle of propagation for the light ray is given by the following expression.

$$\theta_4 = \tan^{-1}\left(\frac{x_1 - x_3}{l_i} - \frac{x_3 - s_x}{f_\mu}\right) \tag{3.24}$$

$$\phi_4 = \tan^{-1}\left(\frac{y_1 - y_3}{l_i} - \frac{y_3 - s_x}{f_\mu}\right) \tag{3.25}$$

Finally, the ray is propagated to the image sensor using the following expression.

$$x_5 = x_3 + f_\mu * \tan\theta_4 \tag{3.26}$$

$$y_5 = y_3 + f_\mu * \tan\phi_4 \tag{3.27}$$

At the image sensor, the individual pixel the ray is incident on can be determined and a count can be added. This process is repeated for all rays for all particles.

It should be noted that the simulator takes into account diffraction effects by randomizing the spatial coordinate of each light ray at the microlens plane and sensor plane through a normally distributed random number generator, set in a manner that the standard deviation is equal to the diffraction-limited spot size. For both the microlens array as well as the main lens the diffraction-limited spot size is 5.2 μm. Analysis at the condition presented here indicates that diffraction does not result in a substantial change in the simulator results. This is due to the large f-number of the main lens and the microlenses where the diffraction limited spot size is smaller than the characteristic spatial dimensions (microlens and pixel pitch) of the camera.

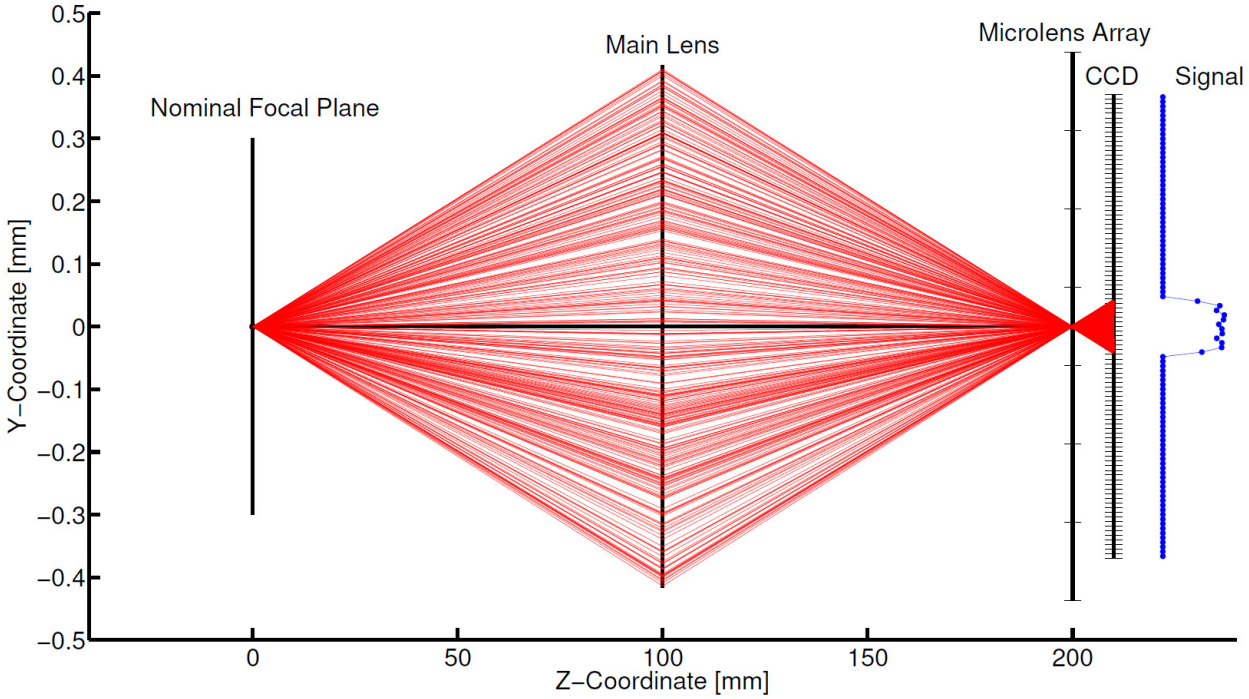
3.4.1 1D Simulations

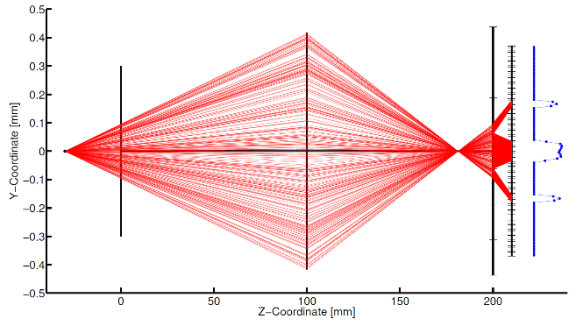
A 1D simulator was constructed as a simple means to evaluate basic camera concepts without requiring a full image simulation, and is far easier to visualize. A detailed description of the simulator construction in Lynch [57], but results are shown here to illustrate the ray tracing process. Figure 3.10 (*top*) shows a particle simulated at the focal point of the optical system. The red lines represent the ray propagation from the particles position through the entire optical system culminating at the image sensor. The blue line, shown behind the CCD, is the integrated signal resulting from the ray tracing procedure. In this case the rays converge onto a single microlens, then spread out onto the image sensor. In Figure 3.10 a-d, the particle is moved in the volume illustrating the unique signal patterns formed by the plenoptic camera. In Figure 3.10a, all of the light rays converge in front of the microlens plane in a manner that is consistent with the image plane moving closer to the main lens as the object plane moves further away. After passing through this focal point, the rays spread out and intersect several microlenses. Depending on the incident angle, the microlens redirect the incident light to different pixels on the image sensor forming a unique image pattern corresponding to the particle positions. Conversely, in Figure 3.10b, the light rays are intersected by the microlens array prior to reaching their focal point forming a distinctly

different image pattern. Figures 3.10c and 3.10d show the effect of shifting the particles position in the y-direction. The effect shown is that the signal is simply shifted.

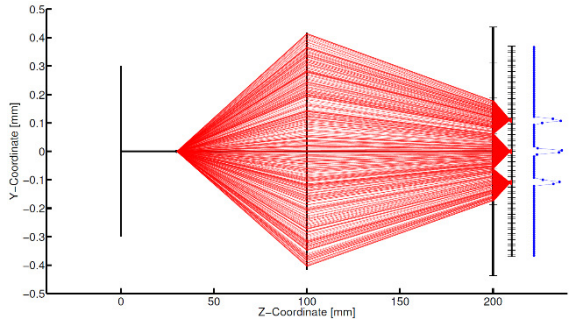
3.4.2 2D Simulations

A sample of the full 2D simulator is shown in Figure 3.11 (*top*). The image provided is a subset of a full image, whose size is set in accordance with the KAI16000 image sensor to 4872 x 3248 pixels. This image was generated using a particle volume ranging from z= -10 mm to +10 mm and a particle density of 0.5 particles per microlens (ppm) or 0.0017 particles per pixel (ppp) resulting in a particle concentration of 2.32 part/mm^3. Upon visual inspection of the image, particles that lie near the focal plane produce nearly circular images that stand out from the rest of the field. The remaining particle images are distributed across multiple microlenses and are difficult to distinguish. As a comparison an experimental image taken with the prototype plenoptic camera is provided in Figure 3.11 (*bottom*).

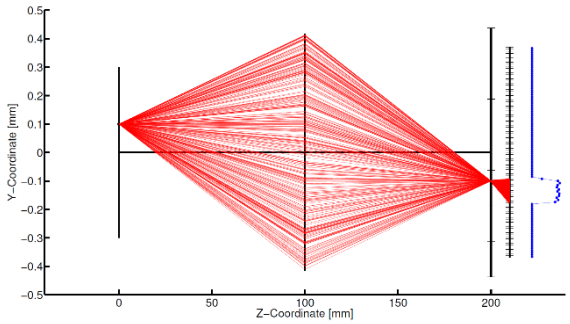




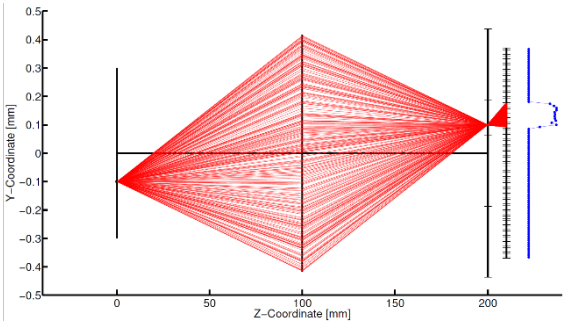
(a) dz = + 30 mm



(b) dz = - 30 mm



(c) dy = + 0.1 mm



(d) dy = - 0.1 mm

Figure 3.10: 1D simulations at different lateral positions. 1 out of every 100 rays shown. Integrated signal shown in blue.

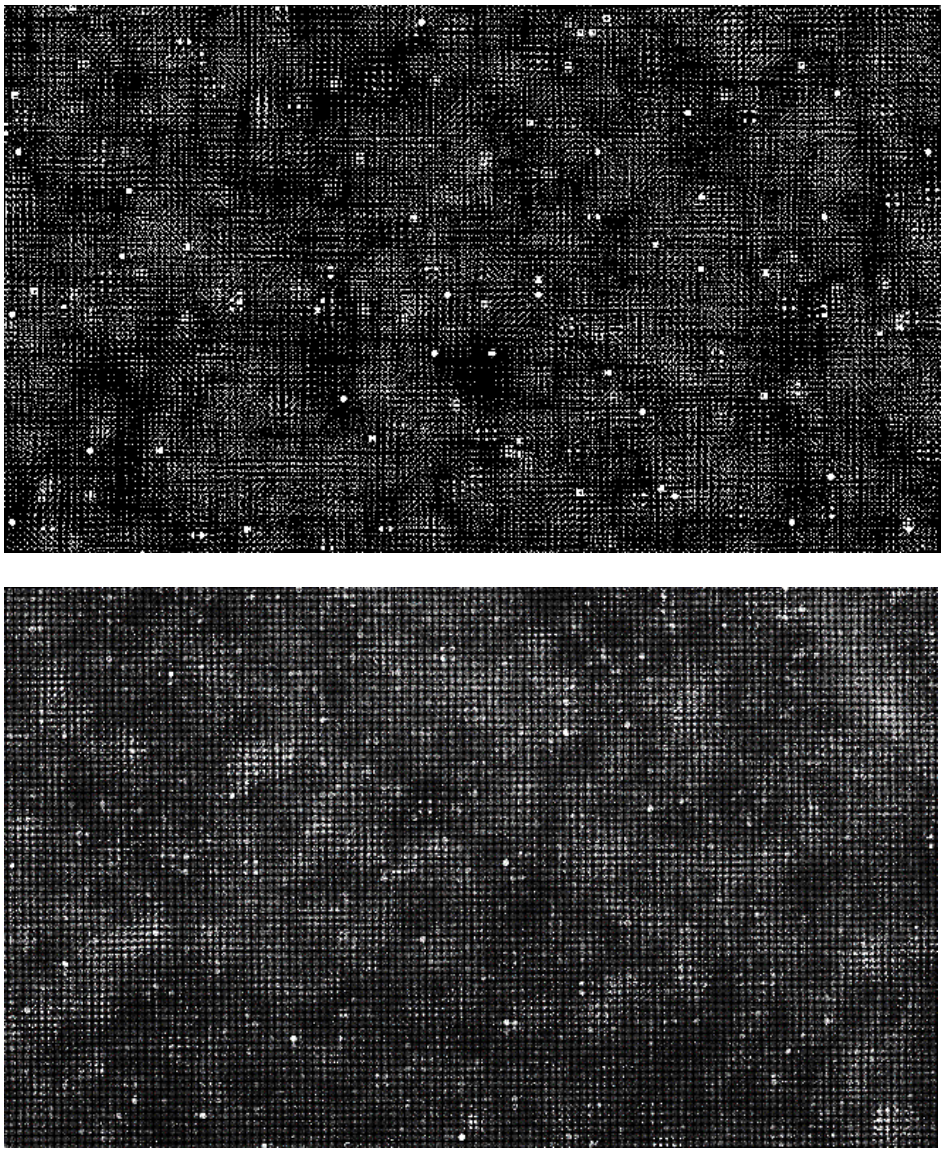


Figure 3.11: Example plenoptic image generated using the ray-tracing simulator (*top*) as well as an experimental image taken with the prototype plenoptic camera (*bottom*).

Chapter 4

Plenoptic Particle Image Velocimetry

The development of plenoptic particle image velocimetry is the focus of this dissertation; in particular, using multiple plenoptic cameras to obtain a 3D/3C vector field. Previous work by Lynch [57] and Fahringer *et al.* [58] has focused on the development of a 3D/3C measurement technique using a single plenoptic camera. This effort has included both development of the hardware necessary to capture image pairs in high-speed flows and the corresponding volume reconstruction algorithms. Both aspects have been explored using a combination of synthetic and experimental data with the basic concept demonstrated in a variety of flow fields ranging from low speed boundary layer flows to heated supersonic jets.

Other notable works with plenoptic-PIV include Skupsch and Brücker [59] who performed multi-plane PIV using a plenoptic camera applied to a convective flow. They utilized a refocusing and thresholding approach to mitigate the effect of out-of-focus particles similar to SAPIV. Ostmann et al.[60] utilized a 3x3 doublet microlens array in combination with a shadowgraphy technique to track the 3D path of particles in a swirling flow. Additionally, La Foy and Vlachos [61] demonstrated the capabilities of using multiple plenoptic cameras to reconstruct synthetic particle volumes. Specifically, they noticed a substantial improvement in depth resolution with the addition of a second camera. Cenedese *et al.* [62] focused specifically on using the plenoptic camera to determine the 3D position of a particle. To do this they generated a series of refocused images and found the image (corresponding to a specific depth location) in which the particle had the smallest area. In addition to the work done using the traditional plenoptic concept, other groups have utilized the plenoptic 2.0 [51], or focused plenoptic, concept to obtain 3D/3C velocimetry data. [63, 64, 65]

The focus of this chapter is on the reconstruction algorithms used to generate the particle volumes which are processed with cross-correlation analysis to yield a 3D/3C vector field. The working principle is described and the steps necessary for processing experimental reconstructions are detailed. The unique aspects of this process, volumetric calibration and reconstruction, are discussed in detail and a parametric study of their performance is given including particle density, number of plenoptic cameras, grid resolution, and various sources of experimental error. In addition a detailed discussion of the reconstruction artifacts known as ghost particles is given and the influence of the particle reconstruction on the final 3D/3C vector field is evaluated via a synthetic 3D Gaussian vortex. Finally, modified versions of the MART algorithm, specifically designed to decrease reconstruction time, are presented as well as a reconstruction of a scalar field showing the potential of this technique beyond PIV applications.

The process of obtaining a 3D/3C vector field from a set of plenoptic images is shown in Figure 4.1. Much like 2D PIV, as discussed in Chapter 2, the working fluid is seeded with tracer particles which are illuminated by a pulsed light source, generally an Nd:YAG laser. The scattered light is then recorded by 2 plenoptic cameras in double-frame mode, requiring a double pulsed laser. Once the images are acquired they are preprocessed in order to mitigate the effects of background noise associated with reflections and the image sensor. In addition a volumetric calibration is needed in order to map multiple cameras to a common grid. Therefore a volumetric calibration algorithm was developed where each camera is calibrated to a single set of object points. Using the calibration the images are reconstructed into a volume of particles which is fed into a 3D cross-correlation technique based on the VODIM algorithm to obtain the final vector field.

In order to facilitate the reconstruction of particle volumes from multiple cameras, a common coordinate system must be used. An illustration of the convention used in this work is presented in Figure 4.2 where three configurations (1-3 cameras) are presented. For each configuration, regardless of the number of cameras or their angles of separation, the

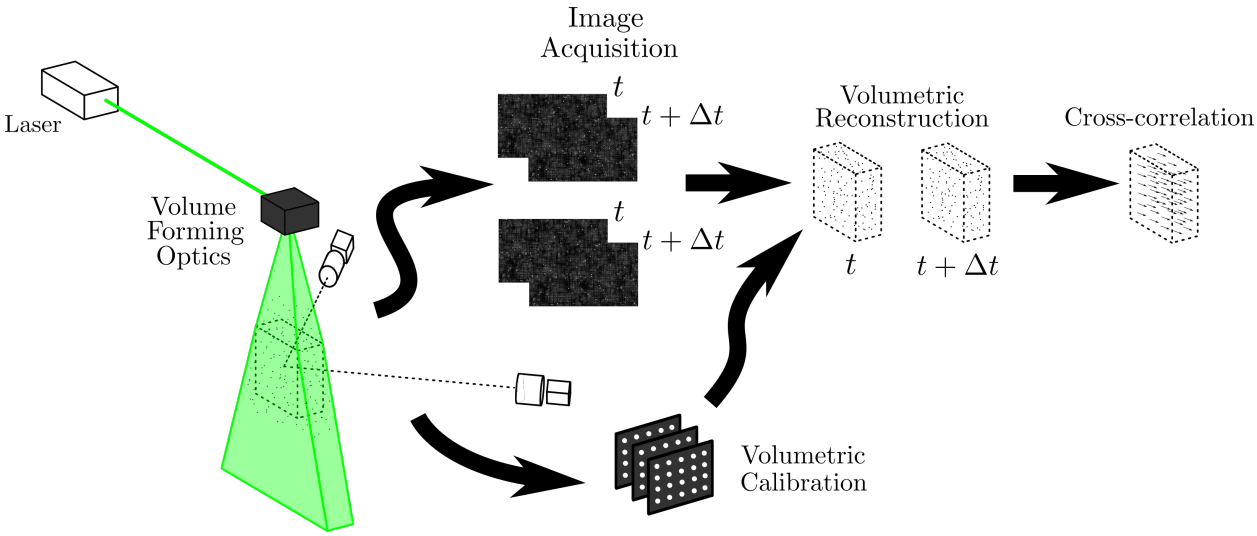


Figure 4.1: Working principle of plenoptic-PIV

world coordinate system (x, y, z) is aligned to the single-camera configuration (or in the case of two cameras, a virtual single-camera configuration which bisects the angle of separation, θ). This means that for the multi-camera configurations a relationship must be developed to relate each camera to the common coordinate system and is the focus of the following section. The remainder of the light field coordinates (u, v, s, t) are defined in the same way as the single-camera configuration and are unique to each camera as illustrated in Figure 3.6.

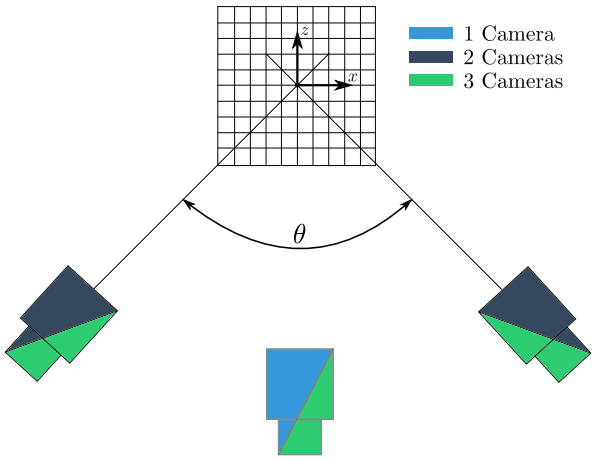


Figure 4.2: Illustration of plenoptic-PIV reconstruction process with multiple cameras.

4.1 Direct Light Field Calibration

The volumetric calibration procedure has two purposes: one to describe the imaging process of the camera and two to ensure both cameras reconstruct the same volume. One such calibration procedure was described earlier (in chapter 3) analytically, where each point in object space (x, y, z) was mapped into the camera to (s', t', α) using the thin lens approximation and the magnification relation. Then, by isolating a particular aperture position (u, v), the location in the original light field that the point in object space was recorded can be determined using a simple linear projection operator as given by

$$s = u\left(1 - \frac{1}{\alpha}\right) + \frac{s'}{\alpha} \qquad t = v\left(1 - \frac{1}{\alpha}\right) + \frac{t'}{\alpha} \tag{4.1}$$

Thus providing a map from $(x, y, z, u, v) \rightarrow (s, t)$. Using this simple calibration volumetric reconstruction can be performed; however, due to the assumptions of a perfect lens in both the thin-lens approximation and linear projection the presence of any optical distortions will cause errors in the reconstruction. In addition, this formulation assumes that the coordinate system is aligned with the camera (z is along the optical axis) making it impossible for two cameras to use a common coordinate system. A more advanced calibration technique which can correct for these assumptions was developed based on empirically imaging a known calibration target, derived from a similar technique developed for stereo-PIV (Soloff *et al.* [30]). The result of the calibration procedure, termed direct light field calibration (DLFC) as outlined in Hall *et al.* [66], is the replacement of the $(x, z, u) \rightarrow s' \rightarrow s$ and $(y, z, v) \rightarrow t' \rightarrow t$ projections in the synthetic light field with a third-order polynomial. Using this formulation equation 3.13 can be rewritten for an arbitrary location (x, y, z) as

$$E(x, y, z) = \iint L\left(u, v, \mathscr{P}_s(x, y, z, u, v), \mathscr{P}_t(x, y, z, u, v)\right)\, du\, dv \tag{4.2}$$

where these polynomials are defined as

$$\begin{aligned}\mathscr{P}(x,y,z,u,v) = {} & a_0 + a_1x + a_2y + a_3z + a_4u + a_5v + a_6x^2 + a_7xy + a_8xz + a_9xu \\ & + a_{10}xv + a_{11}y^2 + a_{12}yz + a_{13}yu + a_{14}yv + a_{15}z^2 + a_{16}zu + a_{17}zv \\ & + a_{18}u^2 + a_{19}uv + a_{20}v^2 + a_{21}x^3 + a_{22}x^2y + a_{23}x^2z + a_{24}x^2u + a_{25}x^2v \\ & + a_{26}xy^2 + a_{27}xyz + a_{28}xyu + a_{29}xyv + a_{30}xz^2 + a_{31}xzu + a_{32}xzv \\ & + a_{33}xu^2 + a_{34}xuv + a_{35}xv^2 + a_{36}y^3 + a_{37}y^2z + a_{38}yz^2 + a_{39}yzu \\ & + a_{40}yzv + a_{41}y^2u + a_{42}yu^2 + a_{43}yuv + a_{44}y^2v + a_{45}yv^2 + a_{46}z^3 \\ & + a_{47}z^2u + a_{48}zu^2 + a_{49}zuv + a_{50}z^2v + a_{51}zv^2 + a_{52}u^3 + a_{53}u^2v \\ & + a_{54}uv^2 + a_{55}v^3 \end{aligned} \tag{4.3}$$

where $a_0 \rightarrow a_{55}$ are the calibration coefficients (note that there are two sets of calibration coefficients: one for s and one for t). The process for obtaining the coefficients for the polynomial is given by the following procedure which is performed independently for each camera.

1. Acquire images of calibration target (x, y) at multiple depth locations (z) spanning the volume of interest.

2. Generate perspective views $E_{uv}(s, t)$ of each calibration image where each pixel in every perspective view is a unique sample of the light field $L(u, v, s, t)$

3. Locate every calibration point in each perspective view

4. Build vectors $(\mathbf{s}, \mathbf{t})$ containing the locations of the calibration points found in step 3 using the following expressions:

$$\mathbf{s} = [s_0(u_0, v_0), s_1(u_0, v_0), \cdots, s_n(u_0, v_0), s_0(u_1, v_0), s_1(u_1, v_0), \cdots, s_n(u_1, v_0), \cdots, s_n(u_m, v_m)]^T$$

$$\mathbf{t} = [t_0(u_0, v_0), t_1(u_0, v_0), \cdots, t_n(u_0, v_0), t_0(u_1, v_0), t_1(u_1, v_0), \cdots, t_n(u_1, v_0), \cdots, t_n(u_m, v_m)]^T$$

where n is the number of calibration points, and m is the number of perspective views in each direction.

5. Build a matrix ($\mathbf{A}$) from the known location of the calibration points as defined by the manufacturing of the calibration target and the depth location the image was acquired at using the following expression:

$$A = \begin{pmatrix} 1 & x_0(u_0,v_0) & y_0(u_0,v_0) & z_0(u_0,v_0) & u_0 & v_0 & x_0(u_0,v_0)^2 & \cdots & v_0^3 \\ 1 & x_1(u_0,v_0) & y_1(u_0,v_0) & z_1(u_0,v_0) & u_0 & v_0 & x_1(u_0,v_0)^2 & \cdots & v_0^3 \\ 1 & x_2(u_0,v_0) & y_2(u_0,v_0) & z_2(u_0,v_0) & u_0 & v_0 & x_2(u_0,v_0)^2 & \cdots & v_0^3 \\ & & \vdots & & & & & & \\ 1 & x_0(u_1,v_0) & y_0(u_1,v_0) & z_0(u_1,v_0) & u_1 & v_0 & x_0(u_1,v_0)^2 & \cdots & v_0^3 \\ 1 & x_1(u_1,v_0) & y_1(u_1,v_0) & z_1(u_1,v_0) & u_1 & v_0 & x_1(u_1,v_0)^2 & \cdots & v_0^3 \\ 1 & x_2(u_1,v_0) & y_2(u_1,v_0) & z_2(u_1,v_0) & u_1 & v_0 & x_2(u_1,v_0)^2 & \cdots & v_0^3 \\ & & \vdots & & & & & & \\ 1 & x_n(u_m,v_m) & y_n(u_m,v_m) & z_n(u_m,v_m) & u_m & v_m & x_n(u_m,v_m)^2 & \cdots & v_m^3 \end{pmatrix}$$

6. Solve the system of equations $\mathbf{a}_s = \mathbf{A}/\mathbf{s}$ and $\mathbf{a}_t = \mathbf{A}/\mathbf{t}$ in a least square sense to obtain the calibration coefficients $a_0 \rightarrow a_{55}$ for both $\mathscr{P}_s$ and $\mathscr{P}_t$.

For synthetic data, the same procedure is repeated, but using a modified version of the synthetic image generation tool outlined in Fahringer *et al.* [58] to complete a Monte-Carlo simulation. This is necessary due to the calibration's presence in reconstruction process as discussed in the following section. In addition, using the calibration in the synthetic tests makes the processing more analogous to real-world experiments. This modified procedure is given by:

1. A large number (1,000 in this study) of synthetic particles are generated at random locations (x, y, z) in the volume.

2. A number of light rays (100 in this study) with random angles are generated and propagated from each particle to the main lens, where their location (u, v) is recorded and then to the microlens array where there location is recorded (s, t).

3. Build vectors $(\mathbf{s}, \mathbf{t})$ using the recorded microlens location for each ray of each particle.

4. Build a matrix $(\mathbf{A})$ from the locations of each particle (x, y, z) and the location on the main lens (u, v) of each light ray from each particle.

5. Solve the system of equations $\mathbf{a}_s = \mathbf{A}/\mathbf{s}$ and $\mathbf{a}_t = \mathbf{A}/\mathbf{t}$ in a least square sense to obtain the calibration coefficients $a_0 \rightarrow a_{55}$ for both $\mathscr{P}_s$ and $\mathscr{P}_t$.

It is noted that an increased number of simulated calibration points and angles was tested with inconsequential change in the coefficients.

4.2 Volumetric Reconstruction Algorithms

The biggest difference between 3D PIV techniques and the traditional planar-PIV methods is the need to resolve the 3D particle field. In this work the individual particle locations are not determined; instead the particle distribution is reconstructed using one of two sets of techniques described herein. The first set of techniques, are based on the computational refocusing ability of plenoptic cameras, with some modifications. The second technique, the Multiplicative Algebraic Reconstruction Technique (MART), is based on tomographic reconstruction of a 3D particle field from its 2D projections which has been used with great success in tomo-PIV [39].

Figure 4.2 illustrates the geometry of the reconstruction problem. Regardless of the number of cameras or their angle of separation, θ, the voxel grid is defined with respect to the single camera configuration. Using the DLFC calibration method described previously we can relate each camera to this grid explicitly.

The reconstruction process begins, for all algorithms discussed herein, by building the light field for each camera (i.e. assigning a u, v, s, t value to each pixel). Then a sweep of

perspective images are generated for each camera fixing the (u, v) sampling for the reconstruction process. Cycling through each voxel, the contributions from each perspective view are determined by projecting that voxel onto the imaging plane (s, t) at the angle (u, v) given by the perspective view using the DLFC data. Then using a simple bilinear interpolation the perspective views contribution can be evaluated by the particular algorithm. This is repeated for all perspective views for all cameras for a given voxel.

4.2.1 Integral refocusing

To illustrate the effects of the modified refocusing-based reconstruction algorithms, the baseline un-modified case is also presented. The computational framework for this algorithm was presented previously. In short, the original radiance of a location in space can be estimated by integrating over the angular space recorded by the light field at a particular location. For volumetric reconstruction the angular space is limited by the generated perspective views for each camera. This is shown in the modified computational refocusing equation.

$$E(x, y, z) = \sum_{c} \sum_{uv} L\left(u, v, \mathscr{P}_s(x, y, z, u, v), \mathscr{P}_t(x, y, z, u, v)\right) \tag{4.4}$$

4.2.2 Filtered Refocusing

The filtered refocusing algorithm uses the same formulation as the standard integral based refocusing, with the addition of a post reconstruction filter. The filter is based upon the principle that if a particle exists inside of a voxel, then the pixel whose intensity contributes to that voxel's refocused intensity will be non-zero. Instead of only calculating the refocused intensity $E(x, y, z)$ during the reconstruction process, a second value $V(x, y, z)$ is also calculated. This value is given as the percentage of projections whose value is above a SNR threshold. Schematically this can be shown in Figure 3.8 where if we take the eleven projections through point (s', α) we can see that seven of them yield a non-zero value from the original light field $L(u, s')$. Therefore the percentage of valid (projections above zero)

for this location $V(x, y, z) = 0.63$. In equation form this is given by

$$V(x, y, z) = \frac{n_{proj,valid}}{n_{proj,total}} \tag{4.5}$$

Therefore all voxels are represented as two values, their intensity and the percentage of valid projections. To get the final reconstructed volume, all voxel's that have less than a desired percentage of valid projections are set to zero. Mathematically, the final intensity is determined by

$$E(x, y, z) = \begin{cases} E(x, y, z) & \text{if } V(x, y, z) > \text{desired percentage} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

In theory, if a particular location is occupied by a particle, then all the light rays passing through that point should have a finite irradiance value such that a threshold of 100% can be set as the pass/fail condition for whether a particle occupies that position in space. In practice, however, there are several reasons why this threshold needs to be lowered in order to yield a useful reconstruction. The first reason is due to the discrete nature of the refocusing procedure and the grid upon which the refocusing procedure is applied. In the current implementation of the refocusing algorithm, a uniform grid of points is created and all light rays passing through those points are determined. Nominally, the spacing between grid points is one microlens diameter as each microlens samples all light rays incident upon it. The true position of a particle, however, is not expected to match up perfectly with this grid. As such, when the recorded light field is interrogated at this position, some of the interpolated irradiance values might fall below the threshold due to subgrid positioning of the particle relative to the grid.

A second reason for relaxing the threshold is due to practical elements in construction of the camera and image sensor. Pixel defects or the presence of dust on the image sensor can lead to reduced or non-existent signal levels for particular pixel locations. The sensitivity

of each pixel can depend on the incident angle of light rays striking that pixel. In addition, the alignment of the microlens images with the image sensor is imperfect such that one pixel might sample only a small portion of the main lens aperture leading to a reduced signal. In the case of coherent illumination, such as with a laser, constructive and destructive interference could affect the signal measured at the image sensor surface. These effects have yet to be explored in detail, but could represent practical reasons to reduce the pass/fail threshold below 100%.

4.2.3 Multiplicative Refocusing

Another adaptation of the refocusing algorithm, first introduced by La Foy and Vlachos [61], is multiplicative refocusing, which is based, conceptually, on the multiplied-line-of-sight (MLOS, Atkinson and Soria [67]) algorithm. This method attempts to remove the blur associated with out-of-focus particles with the scaling of multiplication. Effectively, the reconstructed signal is equivalent to the projection intensity raised to the power of the number of views, n_{uv}. If the irradiance generated from particles is significantly higher than the background noise, the refocused signal, generated by multiplicative refocusing, will be significantly higher. (i.e. $1000^{n_{uv}} >> 10^{n_{uv}}$). Due to this scaling, some consideration must be made to ensure that a computer can store a very large number. If this goes untreated voxels will erroneously contain infinite radiance. Appropriate signal attenuation is obtained by raising each projection by the inverse of the number of views. Additionally to mitigate a similar effect due to a large number of cameras the contribution from each camera is raised by the inverse of the number of cameras, n_c. The multiplicative refocusing equation is given by:

$$E(x,y,z) = \prod_{c} \left(\prod_{uv} \left[L\left(u,v,\mathscr{P}_s(x,y,z,u,v),\mathscr{P}_t(x,y,z,u,v)\right) \right]^{1/n_{uv}} \right)^{1/n_c} \tag{4.7}$$

4.2.4 Tomographic Reconstruction

Tomographic reconstruction of plenoptic-PIV particle fields is in general an ill-posed problem whose system of equations is underdetermined leading to ambiguity in the solution. A special class of reconstruction algorithms are better suited for these problems and are known as algebraic methods as described by Herman and Lent [68]. These methods rely on iteratively solving a system of linear equations which model the imaging system. The problem can be stated as the projection of the volumetric intensity distribution $E(x, y, z)$ onto a pixel located at (s, t) yields the known intensity of that pixel, $I(s, t)$. In equation form this is given by

$$\sum_j w_{i,j} E_j = I_i \tag{4.8}$$

where N_j represents the number of voxels in the line-of-sight of the ith pixel. The weighting function $w_{i,j}$ describes the relationship between the recorded image (ith pixel) and the 3D volume of interest (jth voxel), and is detailed later in this section. In order to solve this set of equations, iterative techniques have been developed that update the current solution for E based on the previous solution. For additive techniques such as the algebraic reconstruction technique (ART [68]) the update is based on the difference between the image intensity data and the projection of the volume such that when they are equal the update added to the solution is zero. For multiplicative techniques such as the multiplicative algebraic reconstruction technique (MART [68]) the update is based on the ratio of the image intensity data to the projection of the volume such that when they are equal the update multiplied to the solution is unity.

The algorithm used in this work is the standard MART algorithm, which was shown by Elsinga *et al.* [39] to work well in multi-camera tomo-PIV. Starting from an initial guess of the volume $E^0 = 1$ MART is updated via the following expression

$$E_j^{k+1} = E_j^k \left(\frac{I_i}{\sum_j w_{i,j} E_j^k} \right)^{\mu w_{i,j}} \tag{4.9}$$

Where k is the number of iterations and μ is a relaxation parameter which must be less than or equal to two. The exponent restricts updates to parts of the volume affected by the ith pixel by raising the argument to 0, therefore multiplying the current voxel by 1, if the voxel is not affected by the ith pixel.

In order to use tomographic reconstruction, a weighting function describing the unique relationship between the plenoptic camera and the volume must be determined. In techniques such as tomo-PIV, the weighting function is based on a straight line projection of a pixel through the volume. The weighting coefficients are calculated as the overlapping volume between the pixels line-of-sight and the voxel elements normalized by the volume of a voxel. This weighting function works well when the entire volume is in focus, such that the line-of-sight of the pixel is a decent approximation for the formation of the image. Due to the unique point spread function of the plenoptic camera as well as the fact that the volume is ideally out-of-focus (at least in a conventional sense), this method of calculating the weights is not applicable. With this in mind, a method for determining the weighting function was developed in Fahringer *et al.* [58] by considering the unique nature of the plenoptic camera. Due to the complex nature of the plenoptic camera this was a complex process that resulted in a large weighting matrix (roughly 360 GB). An alternate method for calculating the weighting matrix is to simply use the DLFC data as it fundamentally describes the imaging process of the plenoptic camera. In essence, the DLFC data provides a model which relates arbitrary locations in the volume, E_j, to their imaged positions in perspective views I_i. Therefore for each E_j the location on the image sensor for each perspective view is determined (s, t). Since this value will, most likely, not align with a single pixel the four nearest pixels are considered. The weights are generated as the interpolation coefficients from a bilinear interpolation scheme for each pixel, and the contribution of the four pixels are processed with the MART equation. Similar approaches have been taken in tomo-PIV. This simpler approach is used in the remainder of the thesis.

4.3 Synthetic test results

In order to facilitate a parametric study of the multi-camera plenoptic system, synthetic images are used such that the accuracy of the reconstruction can be compared to a known solution. For more information on the specifics of generating synthetic particle images used in this work the reader is referred to Fahringer *et al.* [58]. To reduce the number of permutations in the parametric study, some of the variables will be held constant. First, the simulated cameras are the same as the 29 Mp second generation plenoptic cameras used in the experimental portion of this work. Second, the multi-camera system will operate at 1:1 imaging conditions using a 50 mm focal length lens to mirror the single camera work done previously [58, 69]. In addition, Table 4.1 gives a list of all the default simulation parameters used in this section. Unless otherwise stated these are the values used.

4.3.1 Qualitative analysis

To visually demonstrate the volumetric reconstruction capabilities of these techniques, a volume containing 319 synthetically generated particles was imaged by both one and two plenoptic camera systems. Each case was then reconstructed with the four techniques. Figure 4.3 (*left*) shows the projection (summation of all signal) of the 3D volume, created with two-camera MART, along the y-direction. The colormap starts at white (0 counts) and increases to black (max counts) while the red circles indicate the actual location of the particles. This figure shows that the MART algorithm, using two cameras, reconstructs the positions and shape (spherical) of the particles accurately. A subset of the larger volume was extracted for ease of visualization and comparison of the 8 different camera and reconstruction combinations. For integral refocusing, each particle spans the entire volume (along depth) creating the black streaks visible in the insets. In the two camera configuration, the intersection of these streaks creates a clear 'x' shape at each particle location. The two modified refocusing techniques perform markedly better, removing most of the blur associated with traditional refocusing. In the single-camera configuration, both techniques

Table 4.1: Default simulation parameters. Parameter types: 'Fixed' parameters set based on the manufacturing of the current plenoptic cameras available for this study, 'Constant' parameters remain unchanged throughout the simulations, but could be changed in future work, and 'Variable' are changed throughout this work. Unless otherwise specified these are the parameters used in every study.

Parameter	Type	Symbol	Value
Number of pixels	Fixed	-	6600 x 4400
Pixel pitch	Fixed	p_p	0.0055 mm
Microlens packing	Fixed	-	Hexagonal
Number of microlenses	Fixed	-	471 x 362
Microlens pitch	Fixed	p_μ	0.077 mm
Microlens focal length	Fixed	f_μ	0.308 mm
Main lens focal length	Constant	f_m	50 mm
Main lens f-number	Constant	$(f/\#)_m$	2
Magnification	Constant	M	-1
Reconstructed volume size	Constant	-	20 x 20 x 20 mm^3
Number of cameras	Variable	N_c	2
Camera separation angle	Variable	θ	90°
Reconstructed grid density	Variable	-	260 x 260 x 260
Reconstruction algorithm	Variable	-	MART
Relaxation parameter	Variable	μ	1.0
Number of iterations	Variable	-	5
s, t sampling	Variable	-	$2.0p_\mu$
u, v sampling	Variable	-	$1.75\Delta u$
Particle density	Variable	N_p	0.0125 ppμ

perform very similarly clearly showing the 'cigar'-like particle elongation associated with the single-camera configuration. Similar performance between the algorithms is noted in the two-camera configuration, with multiplicative refocusing showing more noise throughout the volume. The MART algorithm shows smaller particle reconstructions in both the one and two camera configurations, which can be attributed to MART's iterative nature.

4.3.2 Quantitative analysis

For dense particle fields, the presence of other particles makes calculating the error of the particle positions difficult. Therefore, to determine the accuracy of the reconstruction process, a statistical measure, known as the reconstruction quality factor, is used. This work utilizes the zero-mean reconstruction quality factor Q^*, as defined in La Foy and Vlachos

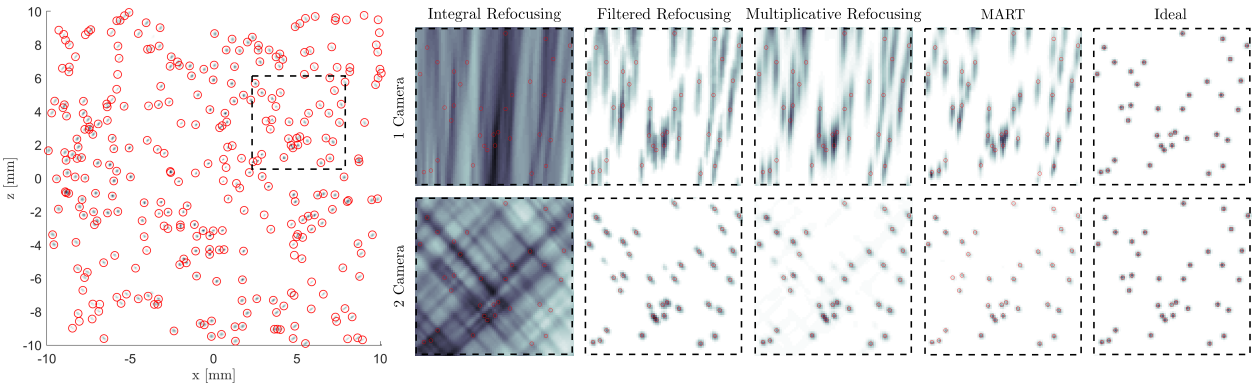


Figure 4.3: Visual comparison of volumetric reconstruction techniques: integral refocusing, filtered refocusing, multiplicative refocusing, and MART. Image on left is a projection (summation of all signal along y-direction) of the full reconstructed volume with two camera MART with insets showing the different reconstruction techniques with 1 or 2 cameras. The colormap starts at white (0 counts) scaling to black (max counts) and the red circles indicate the location of the particles.

[61], where the term zero-mean specifies that the volumes have a mean of zero, which is done by subtracting the mean from the original volume. They demonstrated that as the particle density increased the zero-mean reconstruction quality factor became a more accurate measure than the standard reconstruction quality factor defined in Elsinga *et al.* [39]. The zero-mean quality factor is defined as:

$$Q^* = \frac{\sum \tilde{E}(x,y,z) \cdot \tilde{E}_0(x,y,z)}{\sqrt{\sum \tilde{E}(x,y,z)^2 \cdot \sum \tilde{E}_0(x,y,z)^2}} \tag{4.10}$$

Where $\tilde{E}(x,y,z)$ and $\tilde{E}_0(x,y,z)$ are the zero-mean reconstructed intensity field and the zero mean exact intensity field respectively. The exact intensity volume was created using 3x3x3 voxel Gaussian blobs consistent with the shape of a spherical particle.

Particle density is one of the main parameters to be tested throughout the remainder of this work. Typically, this parameter is given in terms of particles-per-pixel; however due to the unique multiplexing of spatial and angular information onto the plenoptic image sensor, this is better expressed in units of particles-per-microlens (ppμ) since the microlenses

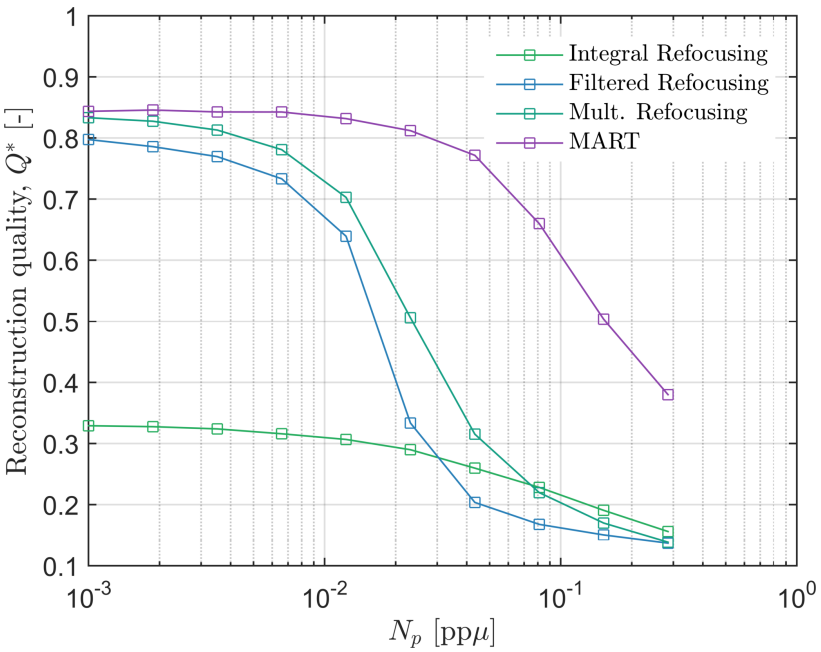


Figure 4.4: Comparison of volumetric reconstruction techniques: integral refocusing, filtered refocusing, multiplicative refocusing, and MART.

govern the spatial sampling in plenoptic cameras. More importantly, this formulation allows for a more direct comparison to conventional imaging techniques, such as tomo-PIV, where the volumetric particle density (particles-per-mm^3) would typically be higher, but the performance of the two techniques in terms of particles-per-pixel/microlens is similar.

A comparison of the four proposed algorithms, using the Q^*-factor criterion, is shown in Figure 4.4 where each algorithm is compared as a function of particle density in the two-camera configuration with 90° of separation. It is clear that integral refocusing does not yield viable accurate particle reconstructions due to reconstructed particles blurring across the entire volume as shown previously. The other two refocusing techniques perform similarly to each other with multiplicative having the slight edge. The MART algorithm is the best performer in the group especially at higher particle densities. Assuming $Q^* > 0.75$ (Elsinga *et al.* [39]) as the cutoff MART is able to yield viable reconstructions at six times the particle density of filtered and multiplicative refocusing. This can be attributed to the iterative nature of MART allowing for corrections to the initial projection. Using this conclusion MART will be the algorithm of choice for the remainder of this work.

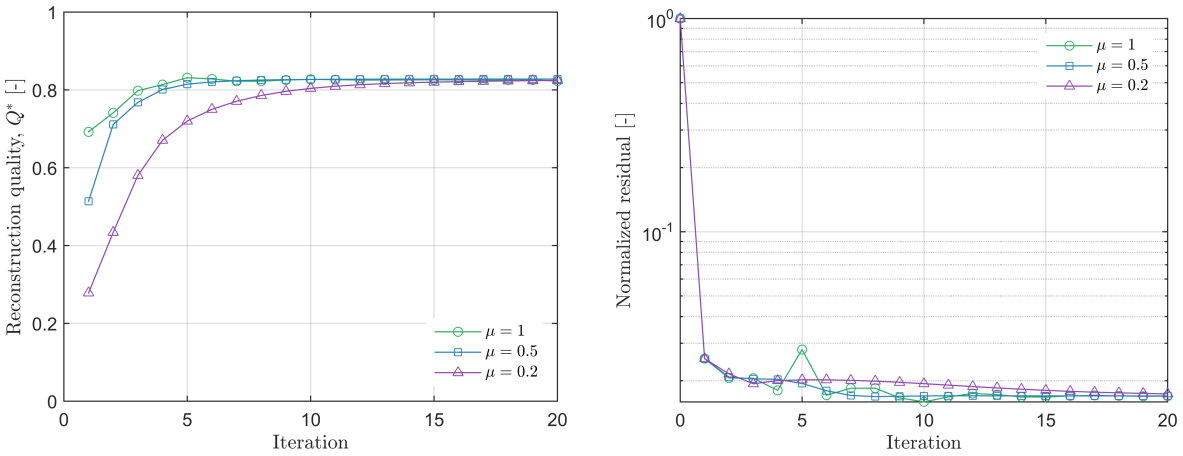


Figure 4.5: Effect of the relaxation parameter and number of iterations on MART reconstruction quality (*left*) and normalized residual (*right*).

The first set of parametric studies is on the tunable parameters for the MART algorithm. Due to the iterative nature of the MART algorithm it is necessary to test that the algorithm converges to the optimal solution. The first test, shown in Figure 4.5, *left*, shows how the solution converges with three different relaxation parameters. It is shown that for $\mu = 0.5$ or 1 the solution converges quickly (~ 5 iterations) and does not diverge from that solution. For the $\mu = 0.2$ case the solution converges slower (~ 15 iterations) but to the same solution. The fact that MART converges to a solution does not mean it necessarily converges to the optimal solution. This can be tested using the normalized residual (or root mean square difference) of the pixel intensity compared to the volume projection (i.e. $E_{uv_i} - \sum_j w_{i,j} E_j$). MART is shown to converge to the same solution after 20 iterations for all three relaxation parameters. For $\mu = 1$ the solution oscillates briefly around iteration 5 but quickly converges thereafter.

In addition to the relaxation parameter and the number of iterations, the effect of sampling density of the perspective views used in the reconstruction process needs to be determined. It is noted that the sampling density (*i.e.* number of views $\Delta u, \Delta v$ and pixel density $\Delta s, \Delta t$) used to extract data from the light field can be arbitrarily chosen using interpolation. Nominally, the choice for Δs and Δt is the size of a microlens and Δu and

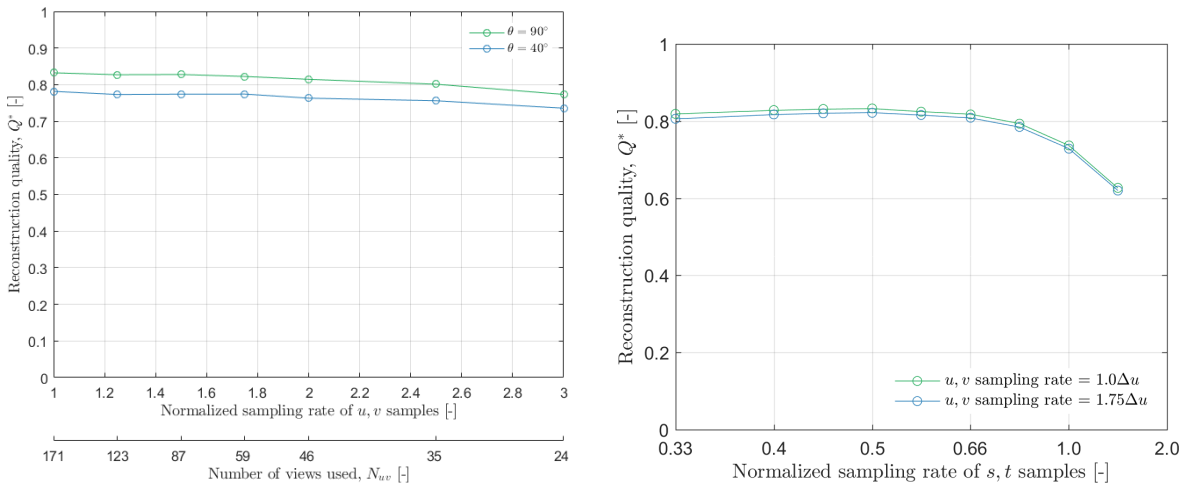


Figure 4.6: Effect of u, v sampling (*left*) and s, t sampling (*right*) on reconstruction quality.

Δv are chosen based on the number of pixels behind each microlens. In Figure 4.6, *left* the effect that the number of views of a single camera has on the reconstruction quality is shown. In this figure the number of views is expressed as the sampling rate normalized by nominal sampling rate ($\Delta u = l_i p_p / f_\mu$). In addition, the total resulting number of views used in the reconstruction is given on a second axis. It is shown that for both separation angles shown here reducing the sampling rate to $1.75\Delta u$ has minimal effect on the quality of the reconstruction, but has the benefit of a near 2x speed up in the reconstruction time. Figure 4.6 (*right*) shows the effect of the resolution of the perspective views on the reconstruction quality for both the $1.0\Delta u$ and $1.75\Delta u$ cases illustrating the compound effect of reducing both resolutions. It is shown that super-sampling the perspective views by 2 times (decreasing the normalized sampling rate of the s, t samples) increases Q^* by 0.1 (13% increase) and that the decrease in quality due to the u, v sampling is minimal.

Experimental configurations are examined and shown in Figure 4.7. Figure 4.7 (*top left*) shows the effect of the number of cameras on the reconstruction quality. It is shown that the addition of a second camera is a significant improvement over a single camera, and the addition of a third camera provides some improvement, particularly at higher particle densities, but there are diminishing returns with the addition of the third camera. The effect that the angle of separation, θ as referenced in Figure 4.2, has on the reconstruction

is shown in Figure 4.7 (*top right*). The maximum quality is shown to be approximately 75°; however the acceptable range ($Q^* > 0.75$) is shown to be between 40° and 120°. The resolution of the reconstructed voxel grid is explored in Figure 4.7 (*bottom right*) by holding the pixel (in this case the nominal pixel size of the perspective views i.e. one microlens) size constant and varying the size of the voxels. The comparison volume, in this case, was created with identical particle sizes equal to the nominal 3 x 3 x 3 voxel Gaussian shape from the 1:1 case. It is shown that values between 1.0 and 2 are acceptable with the best solution being a 1:1 voxel to pixel ratio. In real experiments the DLFC calibration data may not yield a perfect calibration. To test the effect of errors in the calibration procedure, after the images were recorded, one camera was displaced along its x-axis during the calculation of the calibration coefficients. From Figure 4.7 (*bottom right*) it is shown that the maximum acceptable calibration error is 0.75 microlenses. It was shown that the DLFC calibration method was able to reduce the calibration errors to less than 0.3 microlenses [66].

The effect of noise on the MART reconstructions is shown in Figure 4.8. The first type of noise is the addition random Gaussian noise on the raw plenoptic image. The noise was generated with a mean of zero and a variance equal to a percent of the peak image intensity (shown as the x-axis of Figure 4.8 (*left*)). This noise was added to the noise-free raw images and all resulting negative values were set to 0 prior to reconstruction. It is shown that left untreated image noise is highly detrimental to the reconstruction quality. After 1-2% the data falls below the allowable quality. To mitigate this effect a sliding mean subtraction was added to the perspective views generated for the reconstruction process. The results, using a 32x32 pixel window, are shown in Figure 4.8 (*left*). The addition of the sliding mean subtraction mitigates the effect of the noise allowing for accurate particle reconstruction even up to 90%.

The second type of noise, shown in Figure 4.8, *right*, is the addition of background particles. For this simulation the background particles are added in a 1 mm region around the reconstructed volume and for comparison the addition of the same amount particles

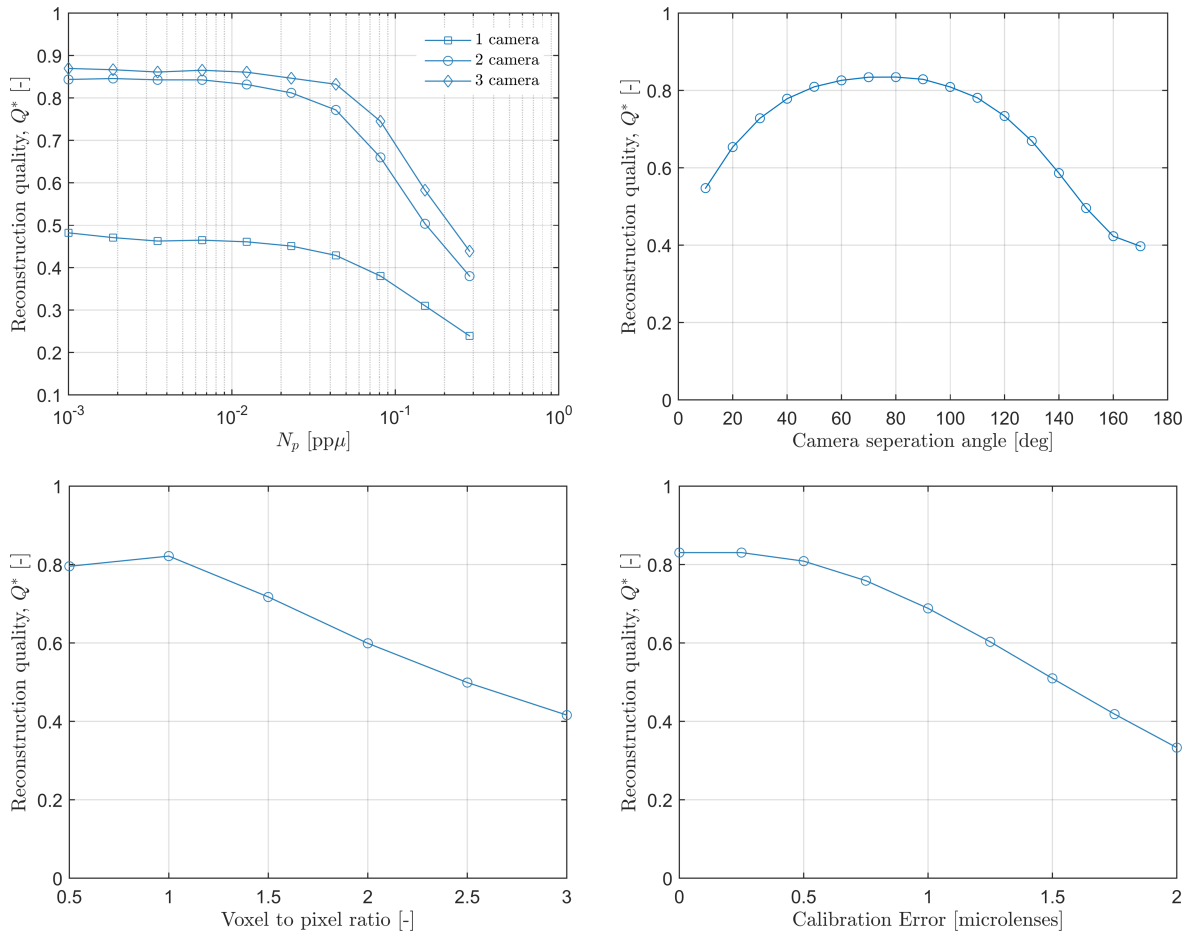


Figure 4.7: Reconstruction quality as a function of number of cameras (*top left*), camera separation angle (*top right*), voxel-pixel ratio (*bottom left*), and calibration error (*bottom right*).

inside the volume (equivalent to increasing the particle density) is considered. It is shown that particles outside the volume reduce the quality of the reconstruction more than ones in the volume. This is due to the fact that the irradiance from the out-of-volume particles is still used in the reconstruction process and that intensity must be placed somewhere inside the volume. Having a large amount of particles outside the bounds of the volume increase the noise floor in the reconstruction essentially adding a low intensity smear across the volume.

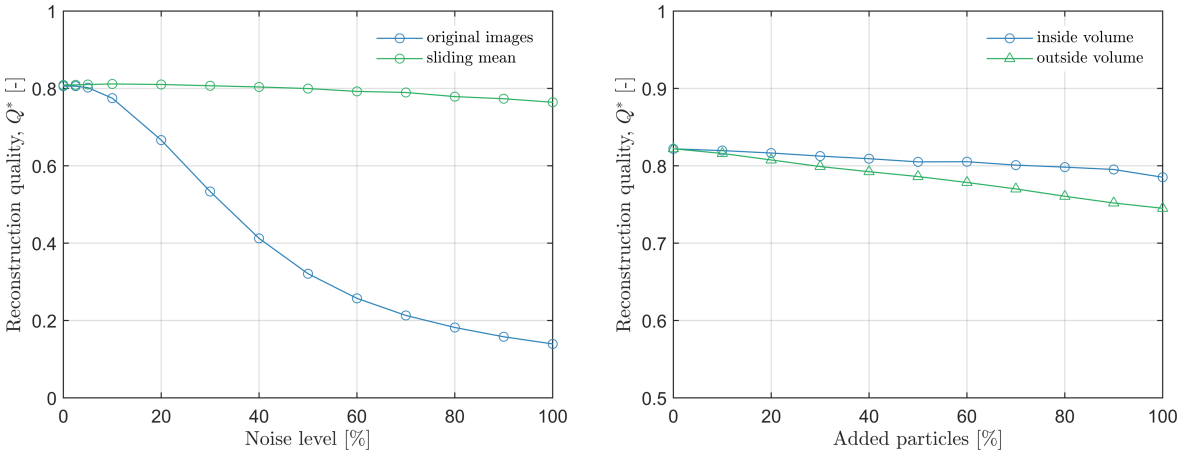


Figure 4.8: The effect of two type of noise on the MART reconstruction: white image noise (*left*) and additional particles (*right*).

4.3.3 Ghost Particles

Due to the fact that the reconstruction process is undetermined there exists multiple intensity fields, E, that solve the system of equations given by equation 4.8. This results in the formation of reconstruction artifacts known as *ghost particles*, which were first discovered in the context of 3D particle triangulation by Maas *et al.* [70] for 3D PTV and tomographic reconstruction by Elsinga *et al.* [39] for tomo-PIV. For single camera plenoptic-PIV the formation of discrete ghost particles was not observed. This was a direct result of the elongation of the particle reconstructions. In essence, at the location where a ghost particle would have formed the elongated particle reconstructions from the actual particles smeared together. For the two camera configuration, discrete ghost particles are shown to exist. The presence and number of ghost particles can be determined, for synthetic data, by counting the peaks of intensity in the reconstructed volume. For this analysis, true reconstructed particles are considered to be intensity peaks located within 1 voxel of a simulated particle and all other peaks (with a value greater than 6 in arbitrary units) are considered ghosts. Several metrics are presented in Figure 4.9 to show the amount of these artifacts as well as their effect on the particle reconstructions.

Plotted in the top row of Figure 4.9 is the amount of detected true particles (*left*) and ghost particles (*right*) as a function of particle density. It is shown that as the particle density increases the number of detected true particles decreases for all algorithms with MART and multiplicative refocusing performing the best for the majority of particle densities. This decrease as a function of particle density is caused by the ghost particles stealing intensity from real particles, and in some cases, overwhelming them entirely. It is also shown that as particle density increases the number of detected ghost particles generally increases with MART outperforming the refocusing based techniques. It is noted that the refocusing based methods deviate from this trend, with the number of detected ghost particles decreasing after 0.05 ppμ for both filtered and multiplicative refocusing. In addition, all three curves are shown to merge toward the same solution as was shown in Figure 4.4 for the reconstruction quality factor. Essentially filtered and multiplicative refocusing are smearing all the intensity across the whole volume *ala* integral refocusing such that a discernible peak does not occur. In addition, the number of ghost particles detected using integral refocusing is constantly decreasing, starting at a value of 400. Since all particles span the entire volume, the volume is filled with signal even at low particle densities. As the number of true particles increase, the voxels available to contain a ghost decrease resulting in the curve presented here. Just as important as the number of detected true/ghost particles is their relative intensity. Ideally, the true particles would have significantly higher intensities such that the cross-correlation algorithms would favor the true particles. This ratio is presented in Figure 4.9 (*bottom left*) and it is shown that MART performs better than the other algorithms for all particle densities and the ghost-to-true intensity ratio is always less than 0.6. The refocusing based methods are shown to merge to the same solution as described earlier, and perform significantly worse after ppμ = 0.05. The effect of ghost particles on the reconstruction is shown in Figure 4.9 (*bottom right*) as the RMSE of the true reconstructed particle locations is presented as a function of particle density. It is shown that all algorithms perform similarly with multiplicative refocusing yielding better results for low particle densities and MART for

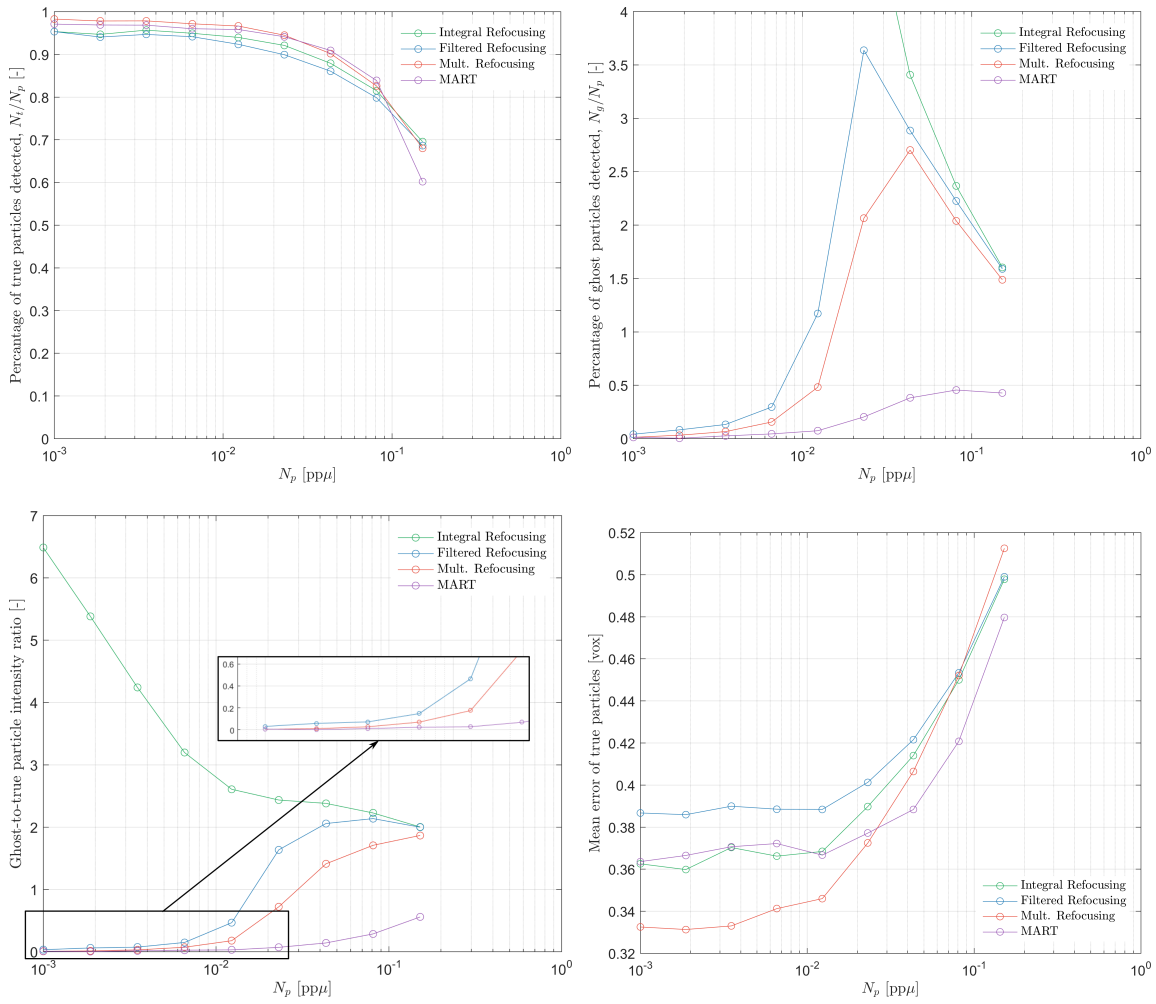
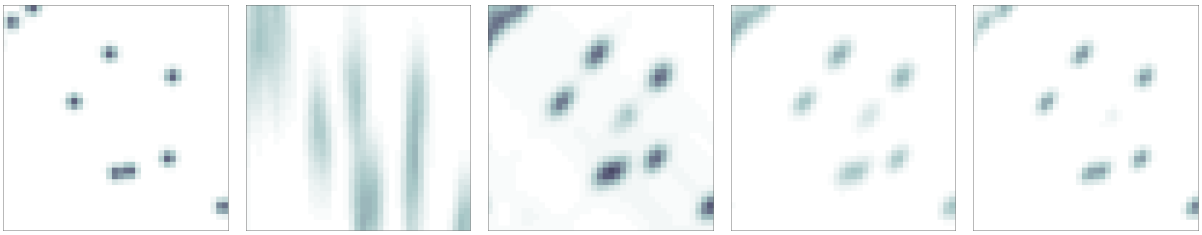


Figure 4.9: Effect of ghost particles on the two-camera reconstruction in terms of: (*top left*) percentage of true particles detected, (*top right*) percentage of ghost particles detected, (*bottom left*) ghost-to-true particle intensity ratio, and (*bottom right*) mean error of true particles as a function of particle number density.

high densities. This figure along with figure 4.9 (*top right*) illustrate the influence of ghost particles has on the true particle reconstructions. For the refocusing based algorithms, these ghosts tend to merge with true particles directly shifting the peaks causing this error. While this effect may occur at high densities for MART, the main source of error is the algorithm shifting the intensities of true particles such that the volume can satisfy equation 4.8.

As an visual example of ghost particles, the formation and suppression of a ghost particle is shown in Figure 4.10. This figure is a zoomed in version of Figure 4.3 at $x = -4, z = 8.5$ where a ghost particle is located surrounded by actual particles. Figure 4.10a shows the



(a) truth (b) 1 camera (c) Mult. Ref. (d) MART 1 iter (e) MART 5 iter

Figure 4.10: The effect of iterations on ghost particle suppression. From left to right: the truth, single camera MART reconstruction, two camera multiplicative refocusing, two camera MART after 1 iteration, and two camera MART after 5 iterations.

exact particle distribution for comparison. Figure 4.10b shows the reconstruction by a single camera configuration illustrating the lack of a discrete ghost particle due to the overlap of the elongated particle reconstructions. Using a non-iterative method like multiplicative refocusing (Figure 4.10c) yields a very strong ghost particle whose peak is 50% of surrounding particles. This is also true of the first iteration of the MART reconstruction (Figure 4.10d), but further iterations reduce the intensity of the ghost particle and at 5 iterations (Figure 4.10e) the peak of the ghost particle is 14% of the surrounding particles.

4.3.4 Gaussian Ring Vortex

The final test is a synthetic experiment designed to test the two-camera plenoptic PIV systems ability to produce an accurate velocity field. This test provides an upper limit on the accuracy of the technique as the images are generated noise free, with zero calibration error. As such, the only sources of error are those created in the reconstruction process. The two cameras ($\theta = 90°$) are imaging a 20 x 20 x 20 mm^3 volume randomly seeded with 8500 particles ($N_p = 0.05$ ppμ). Using the analytical equations for a Gaussian ring vortex these particles are displaced and the image pair is generated, using a synthetic image generation tool (Fahringer *et al.* [58]). The MART algorithm (5 iterations, $\mu = 1$, 260 x 260 x 260 voxels) was then applied in the following configurations: only camera A, only camera B, and both cameras. The resulting volumes were then processed using 3D cross-correlation

software with final window sizes of 32 x 32 x 32 voxels with 75% overlap. As a means of comparison, a pair of volumes was generated using the exact position of the particles (identical to the comparison volumes used for the Q-factor) and were processed using the same cross-correlation software.

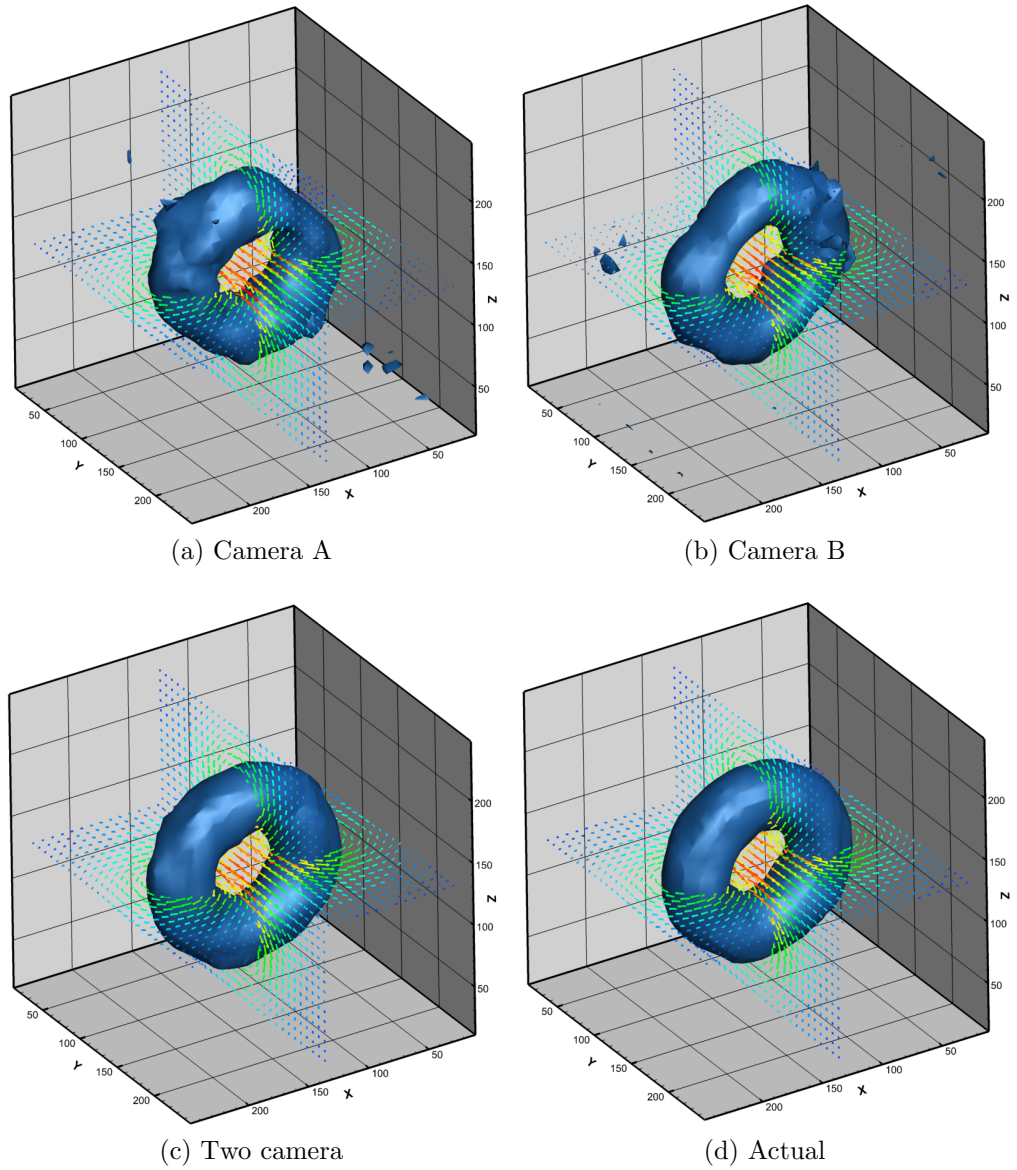
Three-dimensional visualizations of the results for each configuration are shown in Figure 4.11 with an iso-surface of vorticity magnitude at 0.2 voxels/voxel and 2 slices of the velocity field colored by velocity magnitude. The results of the single camera cases are presented in Figure 4.11a for Camera A and 4.11b for Camera B. When compared to the actual velocity (Figure 4.11d) there is noticeable distortion of the vorticity iso-surface. Each case has two noticeably thinner sections which correspond to the location of the nominal focal plane of each camera. This effect is due to the depth of field of the main lens, where all particles image identically onto the image sensor regardless of the z-position in this zone. Note that the coordinate system is defined at a 45 degree angle between both cameras. Additionally, there are areas other than the vortex ring that show vorticity, which is related to noise in the reconstructed velocity field. For the two camera configuration (Figure 4.11c) the vorticity iso-surface does not have the same distortions as each single camera case and closely resembles the actual solution. In addition, there are no other sources of vorticity present indicating a mitigation of the noise seen in the single camera cases.

Slices can be extracted from the full 3D velocity field for easier comparison. Figure 4.12 shows slices of u, v, and w components (rows) for all cases (columns). In the u and w components of velocity the benefit of the second camera is obvious. The first two columns (single camera configurations) show a fair amount of noise in the solutions as well as some asymmetries which are not present in either the two-camera configuration (third column) or the actual solution (fourth column). In contrast, The v component looks remarkably similar between all of the cases, at least visually. This is due to the depth direction of both single cameras being a combination of the x and z axis, which is known to be a problematic component for the single camera configuration.

The quantitative benefit of the second camera can be analyzed by presenting the error of the vector field when compared to the actual solution as shown in Figure 4.13. This figure plots the absolute error of the w-component versus the u-component for all three cases. The single camera cases are shown to exhibit large uncertainties along a particular axis, corresponding to the depth direction for each camera as shown in Fahringer *et al.* [58]. By combining both cameras the effect of the out-of-plane error is mitigated leaving only the combination of the in-plane error for each camera. To further emphasize the gain of a second camera the root mean square error of the velocity field is presented in Table 4.2 for all cases and all velocity components. For the u and w components the error was reduced from 0.6 to 0.14 voxels and 0.16 to 0.09 voxels for the v component, resulting in an overall reduction in RMSE of 0.85 to 0.23 voxels. In addition Table 4.2, shows the RMSE for the other three reconstruction methods. It is noted that due to the poor reconstruction performance at high particle densities the images for the refocusing based methods were generated with a $N_p = 0.012$ ppμ. It is shown that MART performs better than the three refocusing based methods for all configurations. For the single camera configuration, multiplicative performs better than filtered which performs better than integral refocusing. In the two camera configuration, multiplicative and integral refocusing perform very similarly and both out-perform filtered refocusing. The ability for integral refocusing to accurately resolve the 3D vector field is likely due to the noise-free recordings used in this study, where the cross-correlation algorithm is able to accurately correlate the peaks created by the crossing lines-of-sight as if they were spherical particles. In the presence of noise, such as the experimental work shown in the next sections, the other algorithms would likely perform better.

Table 4.2: RMSE for Gaussian vortex velocity field reconstructed with 3 camera configurations.

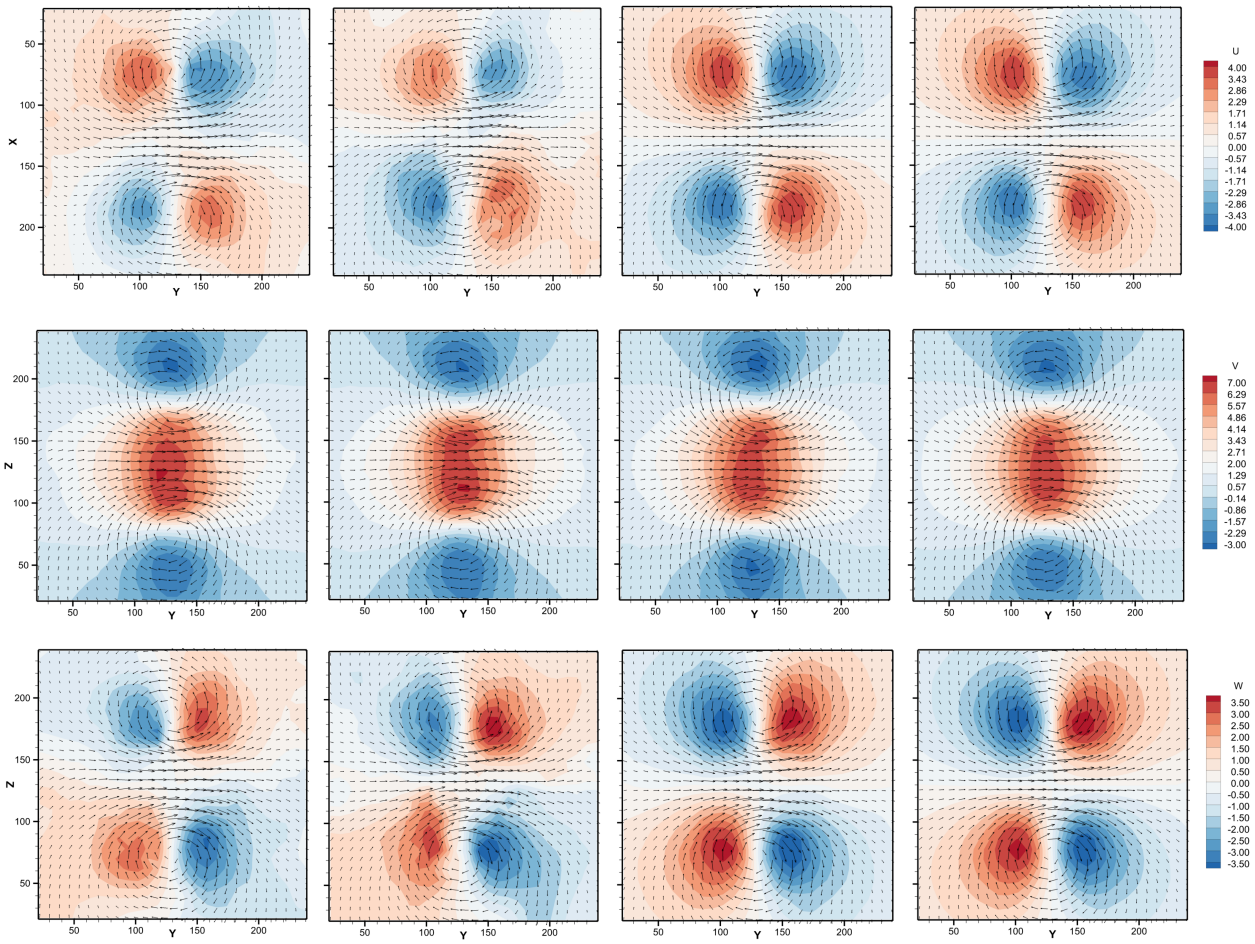
Configuration	u RMSE [vox]				v RMSE [vox]				w RMSE [vox]			
	IR	FR	MR	MART	IR	FR	MR	MART	IR	FR	MR	MART
Camera A	0.72	0.63	0.57	0.61	0.23	0.32	0.22	0.16	0.73	0.63	0.58	0.59
Camera B	0.77	0.71	0.67	0.60	0.25	0.41	0.24	0.16	0.77	0.70	0.66	0.59
Two Camera	0.16	0.19	0.16	0.14	0.16	0.19	0.16	0.09	0.15	0.19	0.15	0.14



(a) Camera A (b) Camera B

(c) Two camera (d) Actual

Figure 4.11: Gaussian ring vortex velocity fields reconstructed with individual cameras as well as the two camera system. Each figure shows vector field slices (colored by velocity magnitude) and a vorticity magnitude iso-surface (0.2 voxels/voxel).



(a) Camera A (b) Camera B (c) Two camera (d) Actual

Figure 4.12: Slices of Gaussian ring vortex velocity fields reconstructed with individual cameras as well as the two camera system. Each row shows a different velocity component: (*top*) u-component in the YX plane (Z = 130), (*middle*) v-component in the YZ plane (X = 130), and (*top*) w-component in the YZ plane (X = 130). Each column shows a different reconstruction configuration from left to right: only camera A, only camera B, both cameras, and the actual answer.

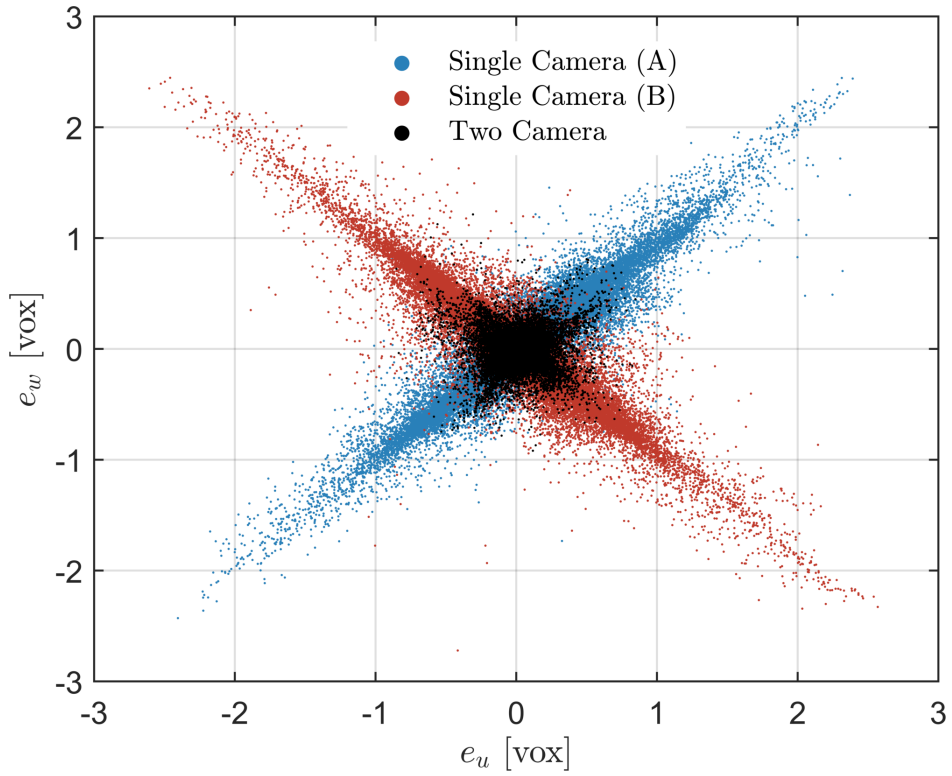


Figure 4.13: Absolute error in reconstructed Gaussian ring vortex velocity field for each individual camera as well as the two camera system.

4.4 Experimental assessment

The experimental validation of the proposed technique was conducted in a 25L purpose built water tank (based on Gharib *et al.* [71]) in which a ring vortex was generated as shown in Figure 5.20. To generate the vortex a pulse triggers the opening of a solenoid which releases a slug of water (L/D = 4) through a nozzle mounted on the bottom of the tunnel. Driving the flow is a constant head pressure tank system, consisting of two tanks and a pump, used to generate a constant 0.1 m head. For this experiment a nozzle with a diameter of 19 mm and a non-dimensional length of L/D = 12 was used. Illumination was provided by a Quantel Evergreen HP (400 mJ/pulse, 10 Hz, running at ∼30% max energy) which scattered light off of silver-coated hollow glass spheres (mean diameter of 10 microns at an estimated particle density of 0.03 ppμ), onto two plenoptic cameras at a 90° angular separation. The camera sensors and microlens arrays have the same specifications as used in the simulation section as defined in Table 4.1 by the fixed parameters; however, these cameras were equipped with 60 mm lenses operating at an f/# = 2.8 and a magnification of -0.4 which results in an effective depth-of-field of 55 mm. Two-hundred image pairs were acquired with a pulse separation of 9 ms. The center of each camera's field of view was focused to a point roughly 2.6 diameters downstream of the nozzle at which the origin of the reconstructed volume is defined where the y-axis is aligned along the axis of the nozzle and the z-axis is aligned between the two cameras.

The raw images were reconstructed using all four of the algorithms in the two camera configuration. All algorithms were processed using mean subtractions and, specifically, for filtered refocusing an intensity threshold of 0 (with the use of the mean subtraction filter) and filter threshold of 0.925 was used, and for MART a relaxation parameter of 1.0 and 5 iterations were used. The reconstructed volumes were 60x60x60 mm^3 in physical size and were discretized into 300x300x300 voxels. To obtain the 3D vector field each volume pair was processed using the cross-correlation scheme utilizing 4 passes starting with 64^3 windows and

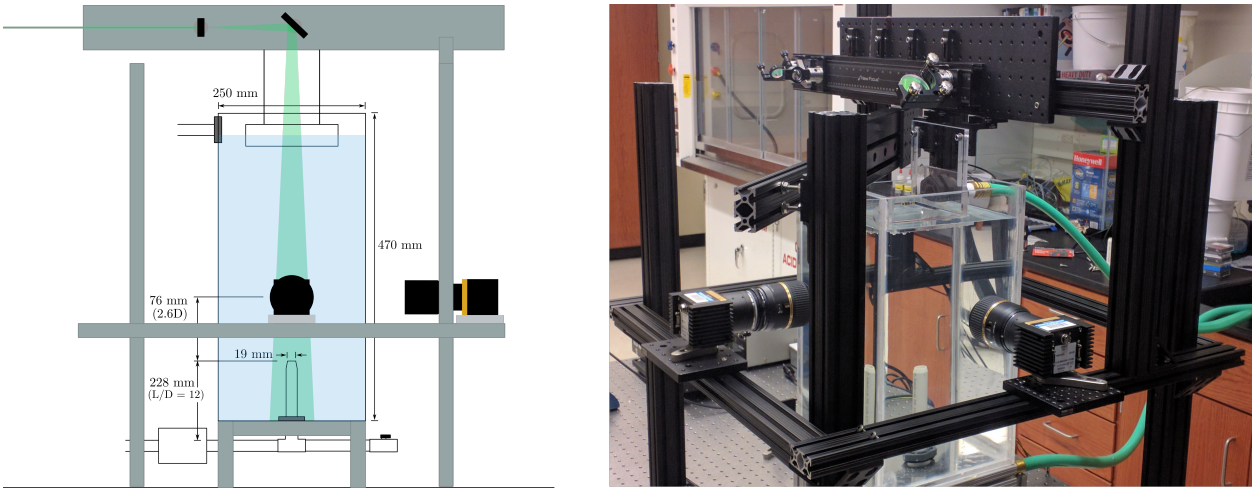
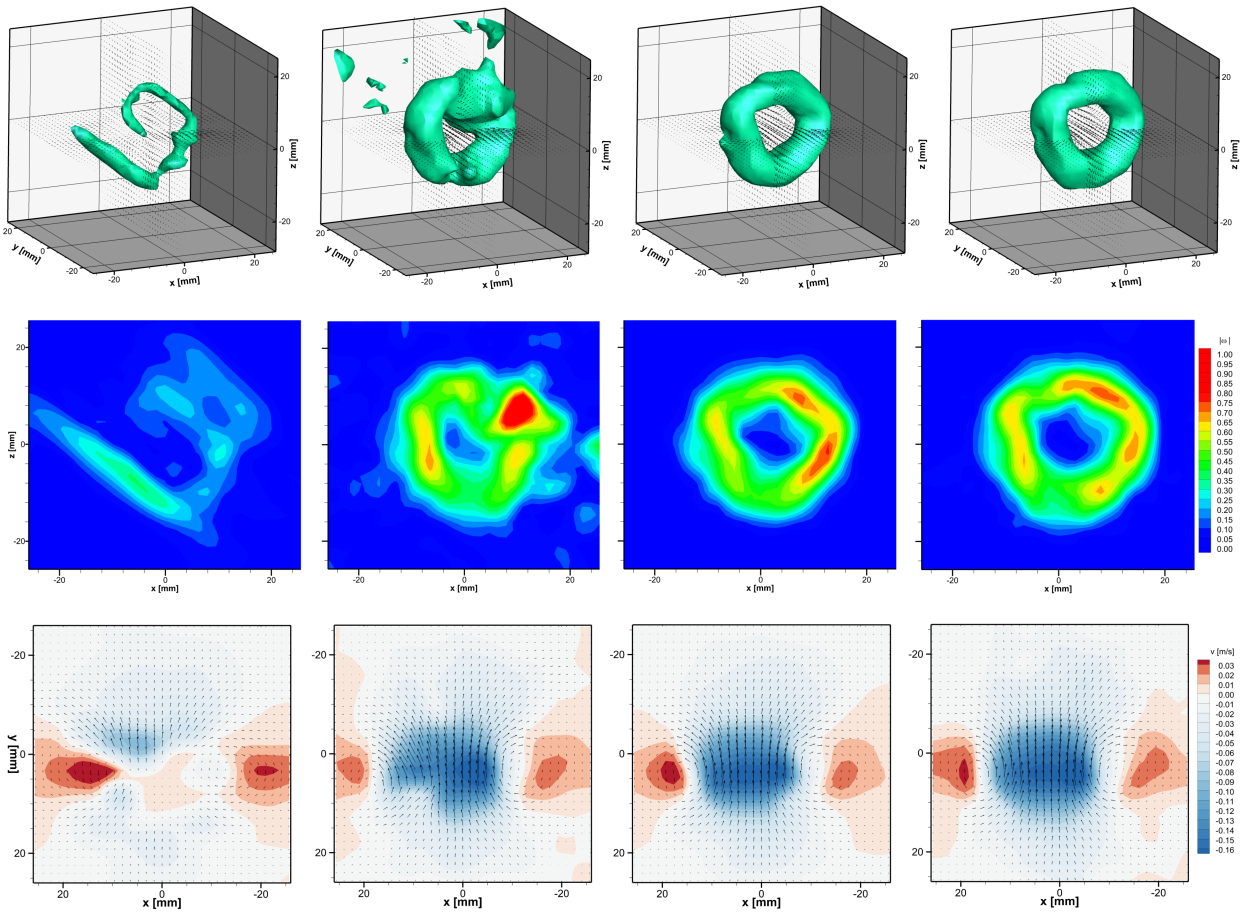


Figure 4.14: Experimental Arrangement.

finishing with 32^3 windows with 50% overlap resulting in 33^3 vectors with a spatial resolution of 3.2 mm/vector.

Results from an instantaneous image pair are shown in Figure 4.15. Each column in this figure depicts a different reconstruction technique, from left to right: integral refocusing, filtered refocusing, multiplicative refocusing, and MART. Each column shows a different visualization of the same data, from top to bottom: 3D vorticity magnitude iso-surface (20 s^{-1}), streamwise summation of vorticity magnitude ($\int \omega dy$) shown in the cross-stream plane, and v-velocity contour with vectors. For integral refocusing, the 3D vorticity iso-surface does not form a well defined ring as expected. Instead,the core appears as a rectangle aligned with the directions of each camera. In addition, the two planar slices show severe distortions in the vortex core, particularly in the v component. Filtered refocusing shows an improvement over the standard integral refocusing. The 'donut' vorticity structure created by ring vortex is clearly visible in the 3D iso-surface. The slices also illustrate the improvement, showing a more defined vortex pair. It is noted, that there is a significant amount of secondary vorticity present in the reconstructed flow-field which is not expected. Multiplicative refocusing shows a very clean flow field, in the 3D iso-surface plot, with the only vorticity present in the ring vortex itself. In addition, the slices show a much smoother velocity field than filtered or integral refocusing. The MART reconstruction is similar to the multiplicative refocusing



(a) IR (b) FR (c) MR (d) MART

Figure 4.15: Instantaneous vector field generated using four reconstruction algorithms: (a) integral refocusing, (b) filtered refocusing, (c) multiplicative refocusing, and (d) MART. Top row shows 3D vorticity magnitude iso-surface (20 s^{-1}), middle row shows streamwise summation of vorticity magnitude shown in the cross-stream plane in arbitrary units, and bottom row shows v-velocity contour.

results with a clean flow field with the only vorticity present in the vortex ring itself. The slices show more variation than the multiplicative results, but they both have a very clear 3D vortex ring structure with features that match the expected results.

The relative precision, or uncertainty, of each method can be determined by comparing the consistency of multiple instantaneous vector fields via the standard deviation of the vector fields. Unfortunately, averaging multiple instances of the vortex is non-trivial due to instabilities in the vortex ring causing the vortex to be captured at a different position and orientation (tilt) for each run of the experiment. Therefore, a process was developed to

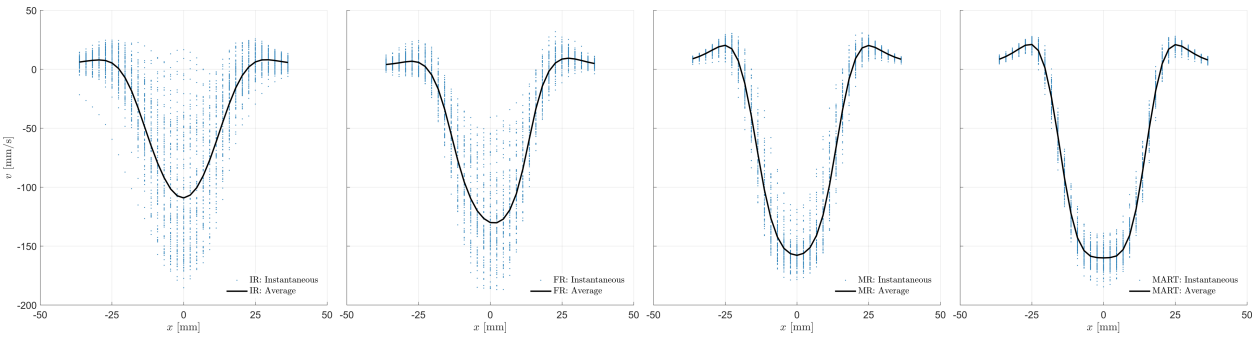


Figure 4.16: Streamwise velocity profiles extracted from 200 instantaneous vector fields with the average shown as the black line for all four methods.

extract a profile through the core of the vortex. First the vorticity magnitude was calculated and a slice was extracted at a specified angle using rotation matrices. Then a line was fitted through the core of the vortex using the two vorticity magnitude peaks (the core of the vortex ring) as tie points and a streamwise velocity profile was extracted. This process is repeated for 360 degrees and the results are averaged. The resulting profiles for all 200 runs are shown in Figure 4.16 as the blue dots and the average for all 200 runs is shown as the black line. The standard deviations for the four methods was calculated to be 23.62, 16.41, 11.12, and 6.97 mm/s for integral refocusing, filtered refocusing, multiplicative refocusing, and MART respectively. The results indicate that the MART algorithm has the highest precision by a significant margin, followed by multiplicative refocusing.

For comparison, a velocity field was generated from a single camera using the MART algorithm (5 iterations, $\mu = 0.2$) for the same image pair and is shown in Figure 4.17. The other algorithms are not presented as the same conclusions from the two camera results can be drawn for the single camera case. It is shown that the single camera technique captures the overall flow field; however, the second vortex (located at $x = 15$mm) is shown to be compressed along the x-axis. This is likely due to the x-component being a function of the depth direction for the single camera. The v-component shows a more accurate flow-field as expected for the in-plane velocity component. In addition, the iso-surface of vorticity magnitude is shown to shrink along the focal plane of the camera, much like the synthetic data.

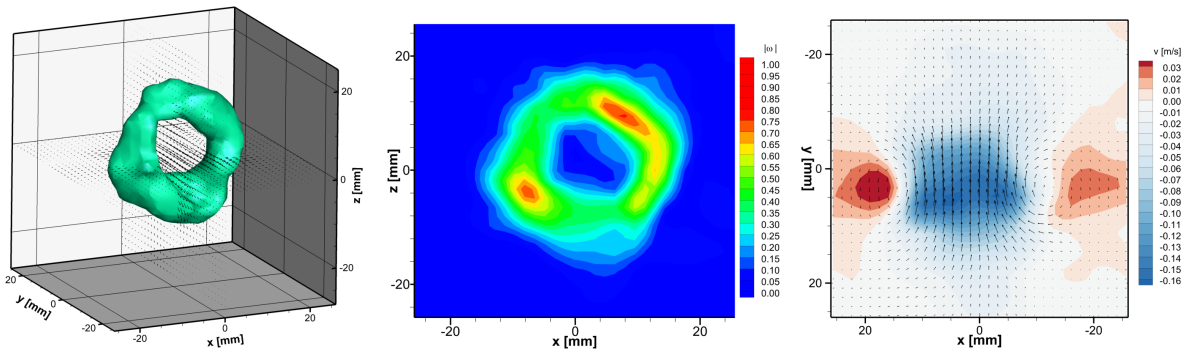


Figure 4.17: Instantaneous vector field generated from single camera configuration using the MART algorithm. left image shows 3D vorticity magnitude iso-surface (20 s^{-1}), middle shows the streamwise summation of vorticity magnitude shown in the cross-stream plane in arbitrary units, and right shows v-velocity contour.

A more detailed comparison of the one versus two camera configuration is presented in Figure 4.18 where two slices are extracted from each data set. These slices are taken from a coordinate system aligned with the single camera (i.e. the x, y axis are coplanar with the focal plane and the z-axis is aligned with the optical axis of the camera), such that the improvement to the out-of-plane velocity component, which serves as the main motivation for this work, can be tested. Figure 4.18 (*top*) shows a slice extracted from the focal plane of this rotated coordinate system on which the in-plane velocity vectors and vorticity magnitude contours are plotted. It is shown that the single camera configuration closely matches the two camera solution. The slices extracted along the optical axis (Figure 4.18, *bottom*) paints a different picture with the single camera configuration showing a severely damped out-of-plane velocity component, clearly illustrating the benefit of adding a second camera.

Streamwise velocity profiles extracted from 200 instantaneous vector fields of the single camera configuration are shown in Figure 4.19. In comparison to the two camera results (Figure 4.16), the single camera reconstructions, using the MART algorithm, yields a standard deviation of 12.31 which is slightly worse than the two camera multiplicative refocusing, but significantly worse (∼2x) than the two-camera MART results.

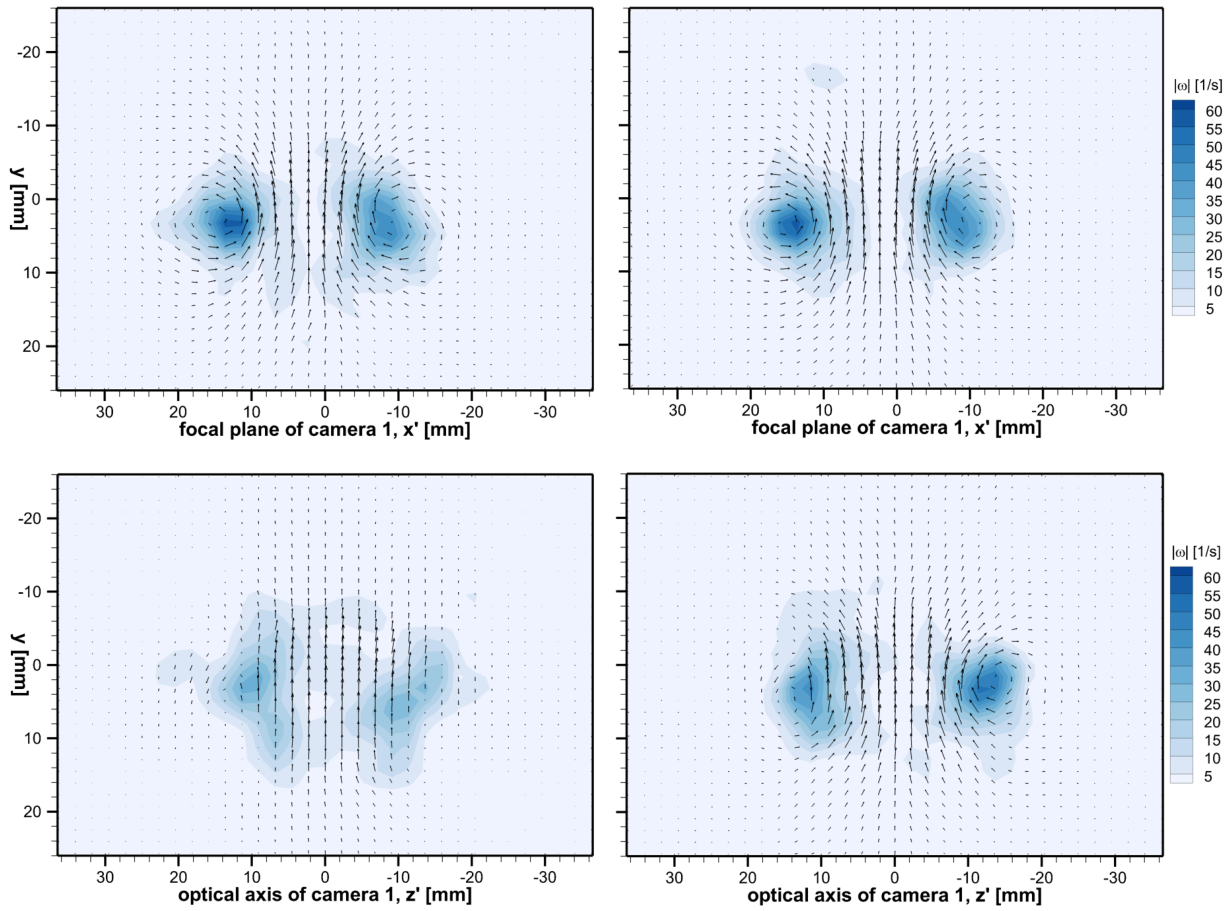


Figure 4.18: Instantaneous vector field generated from single camera configuration (*left*) and two camera configuration (*right*) using the MART algorithm. Data is rotated to be aligned with the single camera configuration. Top row shows the nominal 'x' axis of the single camera (aligned with the focal plane) and bottom row shows data extracted along the optical axis of camera 1. Contours show vorticity magnitude in units of s^{-1}.

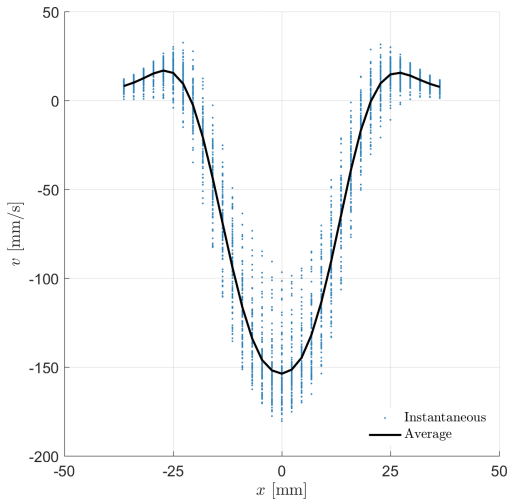


Figure 4.19: Streamwise velocity profiles extracted from 200 instantaneous vector fields with the average shown as the black line for the single camera configuration.

4.5 Acceleration of the MART Algorithm

Due to the iterative processing in the MART algorithm it can take several minutes to compute a single volume. As an example for the default parameters given in Table 4.1 5 iterations of MART would take 6.5 minutes where the refocusing based methods take less than one minute. A trivial addition to the algorithm that can decrease the reconstruction time is to set a lower threshold such that if a voxel's intensity value falls below the threshold it is no longer processed. More advanced methods have been developed in the tomo-PIV community to accelerate the MART processing including multiplied first guess (MLG) by Worth and Nicols [72] and multiplied line-of-sight (MLOS) by Atkinson and Soria [67] which act as a pre-processing step allowing MART to converge faster. Additionally Discetti and Astarita [73] applied a multi-grid approach starting with a coarse grid for the first iteration and refining the voxel grid on each subsequent iteration.

For this work three accelerated versions of MART are considered. The first is simply applying a lower threshold of 1e-5 while using the standard MART algorithm. The other two are based on the MLG/MLOS algorithms where the first iteration of MART is replaced with, in this case, the filtered refocusing and multiplicative refocusing algorithms. Figure

4.20 show the cumulative time to compute a reconstruction for the default test case for all 4 algorithms. It is shown that the basic MART algorithm takes 386 seconds to complete, with 13 seconds of pre-calculations (common to all algorithms), the first iteration taking 98 seconds, and each subsequent iteration takes approximately 70 seconds. The first iteration takes longer because the denominator in the update equation (equation 4.9) must be pre-calculated since we iterate over the voxels instead of the pixels. Simply adding the lower threshold reduces the reconstruction time to 255 seconds. It is shown that until the third iteration the algorithm performs identically to the traditional MART algorithm due to no voxels falling below the threshold. On the third and later iterations the time to complete each iteration is reduced to 23 seconds.

With the replacement of the first iteration with multiplicative/filtered refocusing the cost of that iteration reduces to 30 seconds. In the case of MR-MART the voxels are not below the threshold after the first 4 iterations such that iterations 2-4 take 70 seconds like the regular MART, but the final iteration takes just 3 seconds. For FR-MART ($V = 0.75$) iterations 2-4 take 11 seconds each and the final iteration only takes 0.7 seconds. This is because filtered refocusing sets all voxels less than the validity threshold to zero. It is also shown that the accelerated versions of MART do not produce a lower quality reconstruction with each algorithms final volume yielding a $Q^* \approx 0.82$.

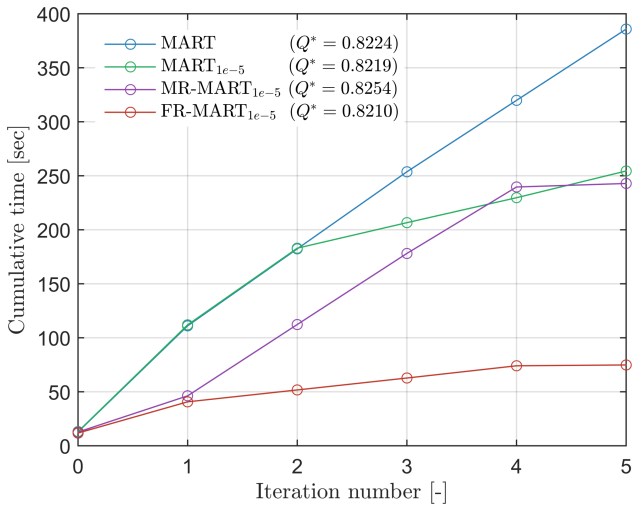


Figure 4.20: Comparison of MART with accelerated variations including MART with a threshold, MR-MART, and FR-MART.

4.6 Scalar Field Reconstruction

Looking more broadly, plenoptic imaging has the potential to be adapted for 3D variations of a wide variety of optical flow diagnostics including non-particle based measurements such as laser induced fluorescence and background oriented Schlieren imaging (Klemkowsky [74]). In general, most other multi-camera techniques represent a sparse angular sampling of the light field such that specific assumptions (e.g. sparse particle field) are needed in order to produce a 3D reconstruction of the volume of interest. Due to these assumptions, these methods cannot easily be extended to provide equivalent 3D information using other types of measurement. Plenoptic cameras, on the other hand, provide a dense sampling of the angular space that allows for the development and implementation of novel algorithms that arent subject to the same constraints and can thus be more easily adapted to provide 3D information for other types of measurements. Computational refocusing is a clear example of one such algorithm that might be exploited for such a purpose such that plenoptic cameras are likely to find a multitude of applications in the future.

As an example, a simple experiment was conducted using two plenoptic cameras imaging the flame produced by Bunsen burner. The experimental setup can be seen in Figure 4.21

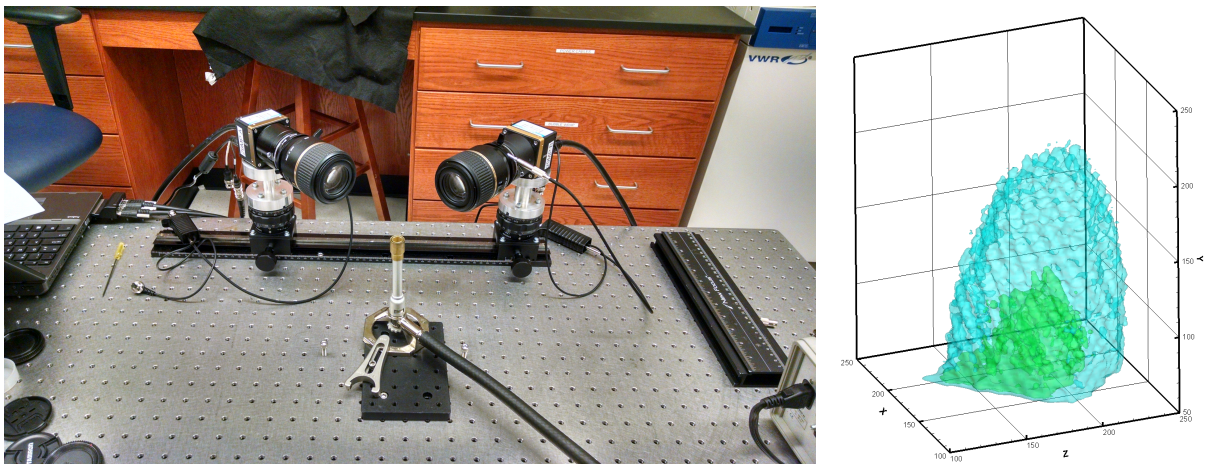


Figure 4.21: MART reconstruction of a 3D flame produced by a Bunsen burner.

where a 90° degree separation between the two cameras is used. Each camera was equipped with a 60 mm macro lens whose nominal magnification was -0.6. A sample reconstruction is shown in this figure that was processed using the MART algorithm ($\mu = 1$, 5 iterations). It is shown that the reconstruction produces a laminar flame as expected; however some artifacts such as the square shape and the roughness of the reconstruction are shown. The square shape is caused by only having two views (the flat sides of the views are aligned with the cameras) and the roughness is attributed to the MART algorithm forcing a maximum entropy solution. Alternative algorithms exist which do not enforce this condition and are designed to handle smooth data. Nevertheless, this simple experiment shows promising results and requires further investigation.

Chapter 5

Design and optimization of a plenoptic camera for PIV applications

This chapter attempts to provide insight into the inherent trade-off between angular and spatial resolution associated with plenoptic cameras. As mentioned previously, the spatial and angular resolution of a plenoptic camera is governed by the size of a microlens; in particular, the physical size of the microlens dictates the spatial resolution, and the number of pixels imaging through each microlens controls the angular resolution. The purpose of this chapter is to establish rules-of-thumb for the selection of a particular microlens size for a given experiment in the context of particle reconstruction. Previous work by Deem *et al.* [75] explored this relationship using geometric optics where they derived a relationship between the size of a reconstructed particle and the relative microlens size; however this relationship was only valid at the nominal focal plane of the camera. In addition, they used the ray-space representation of the light field to indirectly infer the accuracy as a function of depth. This work aims to augment their original analysis to any arbitrary depth using numerical simulations and in doing so provide a guide in the design of a plenoptic camera for particle imaging.

The first part of this chapter describes a fundamental relationship between the camera's parameters and the size/shape of reconstructed particles. This analysis begins with a discussion of the multiple depths-of-field associated with plenoptic imaging which is shown as an accurate model of the blur and elongation of a reconstructed particle. Following this analysis is a theoretical extension of the aforementioned ray-space analysis by Deem *et al.* where a prediction of the reconstruction accuracy at an arbitrary depth is given. Then a series of synthetic experiments are presented to validate this model and finally, an experimental validation is provided.

5.1 Depth of Field

The plenoptic camera offers the unique ability to generate two types of images (refocused and perspective) as described in Chapter 3. One of the major differences between these two images is the level of blurring that occurs throughout the image. For the refocused images (Fig. 3.1) the majority of the image is out-of-focus; whereas in the perspective images (Fig. 3.2) the entire field of view is seemingly in-focus. This range, termed depth of field (DoF), can be calculated for any given imaging system from the near/far depth limits as described by Kingslake [76] and are given by

$$z_n = \frac{d_a l_o}{d_a + c_o} \tag{5.1}$$

$$z_f = \frac{d_a l_o}{d_a - c_o} \tag{5.2}$$

where d_a is the size of the aperture, l_o is the object distance, and c_o is the circle of confusion in object space. The DoF can then be calculated by taking the difference between the far and near field depth limits as given by

$$\text{DoF} = z_f - z_n = \frac{2 d_a l_o c_o}{d_a^2 + c_o^2} \tag{5.3}$$

To obtain the two unique depths of field associated with refocused and perspective views one must change the aperture diameter accordingly. For refocused views the entire aperture (summation of all u's and v's) is used, similar to traditional imaging, and therefore $d_a = p_m$. Perspective views, on the other hand, use only a single pixel (single u,v value), most similar to a pin-hole camera, and therefore utilize a reduced aperture where $d_a = \Delta u = l_i p_p / f_\mu$. The effect of this reduction in aperture can be seen visually in Figure 5.1 where the blue shaded region is the full aperture imaging condition, and the green shaded region is the reduced aperture imaging condition. The depth limits are defined, in this figure, as the intersection of the field of view (blue/green shaded regions) with the image of the circle of confusion

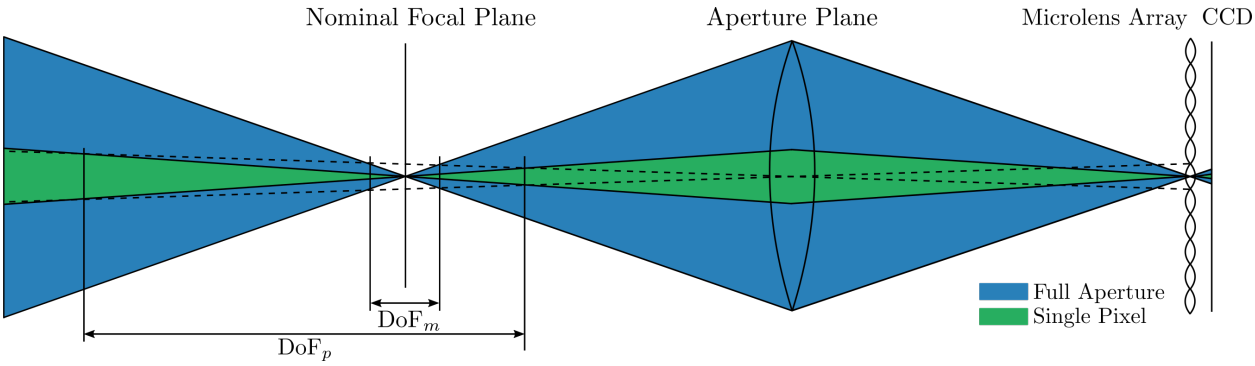


Figure 5.1: Schematic illustrating the two unique depths of field using the full aperture (blue) as well as a single pixel/view (green).

(dotted line). It is shown that the reduced aperture DoF, DoF$_p$, extends far beyond the full aperture DoF, DoF$_m$, which is consistent with the images shown in Figures 3.1 and 3.2.

In the context of particle reconstructions, both the full and reduced aperture DoF have an effect. First, since the particle reconstructions are based on the perspective views, as detailed in Chapter 4, the reduced aperture DoF represents a limit on how deep a volume can be accurately reconstructed. When the volume extends beyond this range it is expected that the quality of the particle reconstruction will decrease as the reconstructed particles blur/elongate. This is similar to other multi-camera techniques such as tomo-PIV where the entire volume is required to be inside the DoF, which is accomplished by increasing the f-number, decreasing the aperture diameter, and extending the depth of field. By doing this, these multi-camera techniques limit the amount of light collected by the imaging system. For each increase of the f-stop the light intensity drops by a factor of two. For plenoptic imaging, this effect is mitigated somewhat since the size of the microlenses increase and therefore the collection area is increased. This would be analogous to increasing the pixel size in a multi-camera system. In addition, the total amount of collected light remains the same as the physical f-number is unchanged. Therefore the signal levels in the reconstructed volumes should be unaffected. Second, the full aperture DoF represents, to some degree, the elongation of the particle reconstructions as well as producing a region around the nominal focal plane at which the particle images will look the same, and therefore create identical

particle reconstructions. This leads to an ambiguous solution in this area for the single-camera configuration.

In general, a decreased full-aperture DoF and an increased reduced-aperture DoF is desired for optimal particle reconstructions. There are two options for decreasing the full-aperture DoF: first is to increase the diameter of the aperture, by decreasing the f-number, and second is to increase the magnification. The first is limited by available lenses and the requirement for f-number matching as discussed in Chapter 3. The second can be done, but at the expense of the size of the field of view/reconstructed volume. To increase the reduced-aperture DoF the reduced apertured diameter and/or the magnification needs to be decreased. Decreasing the magnification has the side effect of increasing the field of view and reducing the resolution (microlenses/mm) of the imaging system and requires a trade-off between DoF_m and DoF_p. The diameter of the aperture, however, can be increased easily by increasing the size of the microlenses. This represents the fundamental trade-off that will be discussed in this Chapter: small microlenses (greater resolution and smaller circle of confusion, small DoF_p) vs. large microlenses (lesser resolution and larger circle of confusion, large DoF_p).

Figure 5.2 shows the effect of changing the microlens size, presented here as number of pixels per microlens $N_{pxp\mu} = p_\mu/p_p$ assuming a constant pixel size, on the single pixel depth of field. The figure on the left shows lines plotted along the near and far field limits with the shaded area in-between indicating the area that is inside the DoF. The figure on the right shows the total DoF for each magnification. As predicted as the size of the microlens, and therefore the circle of confusion, increases or the magnification decreases the single pixel depth of field increases.

5.1.1 Generalized Formulation

The previous formulation is only valid for the nominal focal plane as it was derived in the context of conventional imaging. Since the plenoptic camera can refocus to other planes

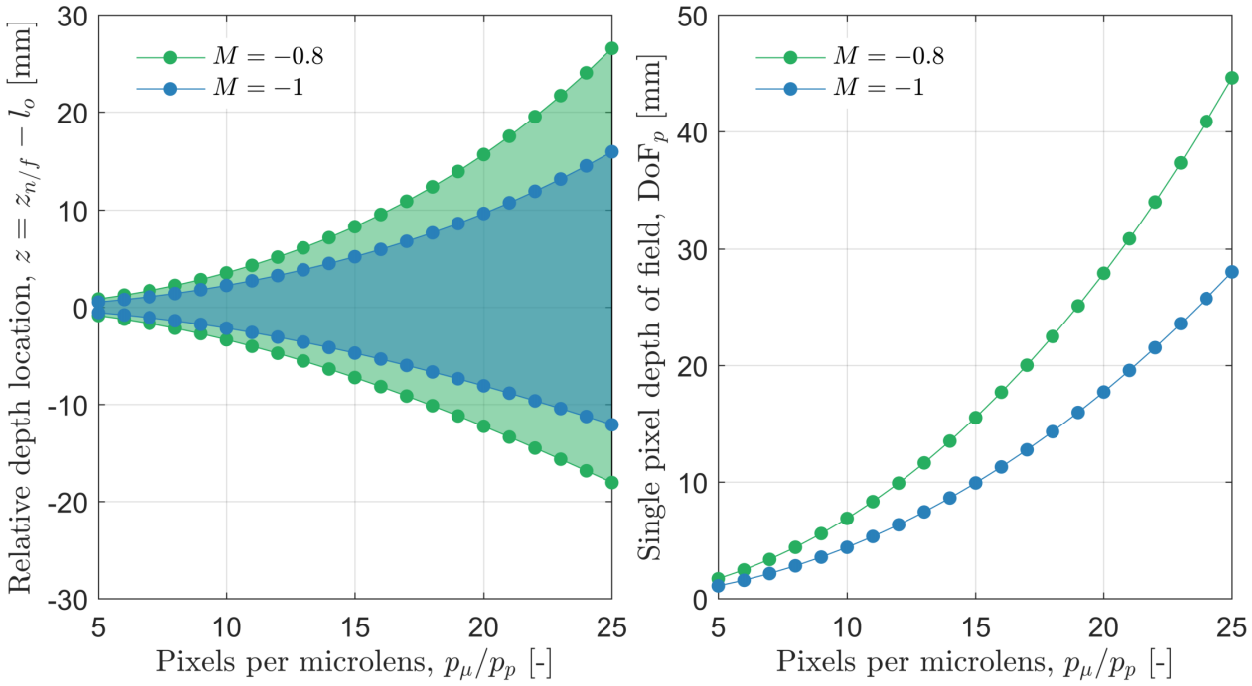


Figure 5.2: Single pixel DoF as a function of microlens size given in pixels per microlens for two magnifications.

a more general framework is needed. This idea has been previously explored by Perwaß and Wietzke [77] from which the following analysis is derived. In particular, they noted that, the circle of confusion varies with depth as shown in Figure 5.3. It is shown that as the object moves further away from the nominal focal plane the image blurs to a larger area resulting in a larger circle of confusion (if it blurs outside the size of a microlens). By calculating the size of this blur spot, a more general circle of confusion can be defined as

$$c_i(\alpha) = \max\left(p_\mu, \Delta u \left|\frac{1}{\alpha} - 1\right|\right) \tag{5.4}$$

where $\alpha = l_i'(z)/l_i$ is the non-dimensional image space location of the object (z). It is noted that within the depth of field this value will be less than the size of a microlens, which much like the pixel in the conventional imaging scenario is the minimum allowable circle of confusion. Therefore any value $c_i(\alpha)$ that is less than the size of a microlens is set to the

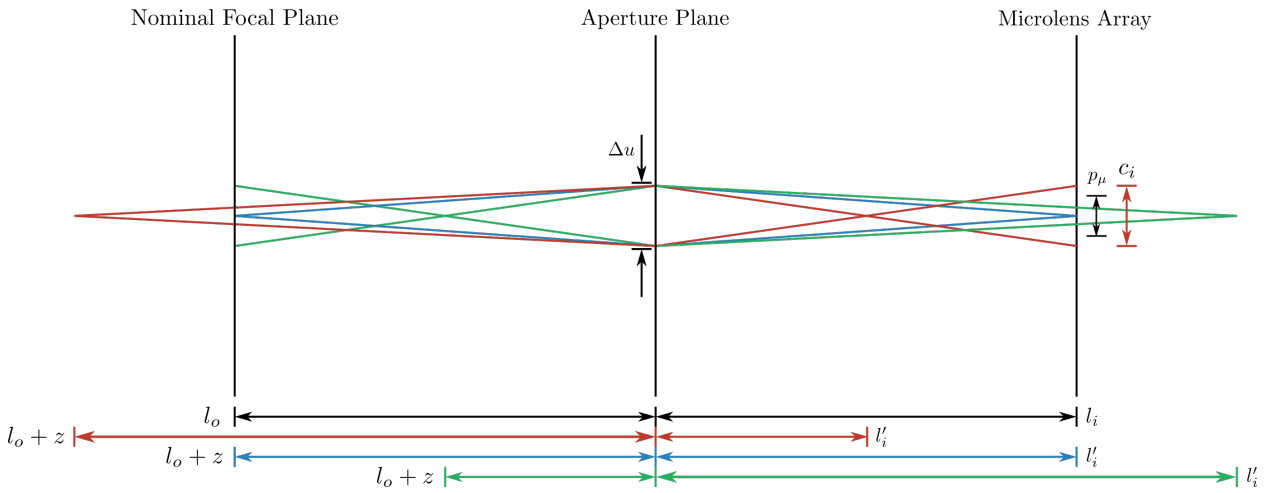


Figure 5.3: Schematic generalizing the calculation of the circle of confusion in *image* space for a perspective view. The colors represent the near field (green), the focal plane (blue), and the far field (red).

size of a microlens. The *object* space CoC can then be defined as

$$c_o(\alpha) = -c_i(\alpha)/M'(\alpha) \tag{5.5}$$

where

$$M'(\alpha) \equiv (l_i + z)/(l_o + z) = \alpha(1 - M) - 1 \tag{5.6}$$

Using this formulation it is possible to calculate the DoF at any arbitrary refocused plane. To illustrate how the circle of confusion operates as a function of depth, the theoretical expression for c_i is plotted against depth z in Figure 5.4. In addition, perspective views of particle simulations (described in the next paragraph) are added for visualization at 7 different depth locations: z = -10, -5, -4, 0, 4, 5, and 10 respectively. This data was calculated using microlenses of size 14 pxpμ and a nominal magnification of -1. It is shown that around the focal plane from approximately -4.1 to 4.5 the circle of confusion remains constant and is given by the size of a microlens. Outside this region the circle of confusion increases due to the particles being outside the reduced aperture DoF, DoF$_p$. This is clearly

evident in the perspective view images where z = -4, 0, and 4 have the same size, but z = -5 and 5 have increased image sizes.

A numerical simulation was performed to test the accuracy of the variable circle of confusion and full aperture depth of field model by simulating the imaging process on synthetic data. The basis for the simulation, and all simulations in this chapter, is the same as the in the previous chapter as defined in Table 4.1, but only using a single camera. The data was generated via the simulation of 1500 individual particles, creating 1500 individual images, using the plenoptic simulation tool discussed in Chapter 3 for three different microlens sizes: p_μ = 10, 14, and 20 pxpμ. Then two separate analysis were performed: the verification of the circle of confusion model and testing if the full-aperture depth of field is a reliable predictor for particle elongation. The circle of confusion model was tested by extracting a perspective view ($u = 0,\ v = 0$) from each particle image, then summing that image along the x-axis creating an image that is a single pixel wide as shown in Figure 5.5. The final image is then created by inserting each individual image, at its respective depth location, into a single continuous image whose boundary should match the variable circle of confusion model. To test the depth of field as a predictor for particle elongation, each of the 1500 images were reconstructed into volumes using the MART algorithm. Then following a similar procedure to the circle of confusion, as illustrated in figure 5.5, the volumes are compressed into a single pixel image and inserted into a continuous image at their respective depth locations.

The results of the circle of confusion test are shown in Figure 5.6 with the y-axis showing the blur in the perspective view and the x-axis showing the particles location in depth for three different microlens sizes of 10, 14, and 20 pxpμ from top to bottom. It is shown that the circle of confusion illustrated by these plots is greater than what the model predicts, which is the result of two factors: the discretization of the microlens array as well as using interpolation in the generation of the perspective views. The discretization error is created when the incident light does not coincide exactly with a microlens. If a cone of light incident on the microlens array is smaller than a microlens (nominal circle of confusion) and is centered

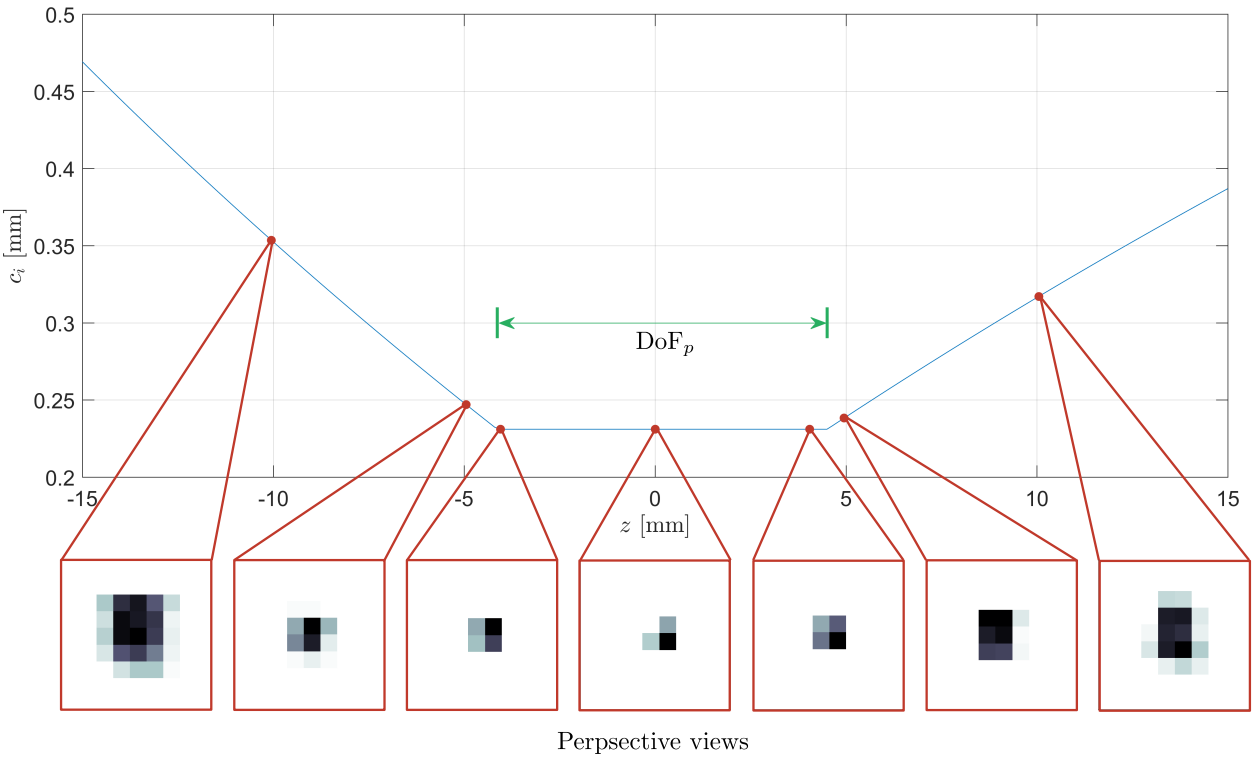


Figure 5.4: Illustration of the generalized circle of confusion, which is constant inside the DoF$_p$ and expands outside of it. Schematic shows c_i plotted against volume depth, z and 7 perspective views, from left to right, located at z = -10, -5, -4, 0, 4, 5, and 10 respectively.

on a single microlens only a single pixel on the perspective view will have signal and the circle of confusion will be equal to one microlens; however, if the cone is centered on the edge of a microlens there will be signal on multiple microlenses and therefore the circle of confusion will be two microlenses. In addition, the effect of interpolation in the generation of the perspective view will be to blur the image on the order of a single pixel. To account for both of these effects the range of circle of confusion's we expect to see is between the theoretical value, defined by equation 5.4, and the theoretical value plus two pixels (in this case a pixel is equivalent to a microlens). The red line in Figure 5.6 shows the model's prediction plus two microlenses and is shown to bound the synthetic data for all cases.

Figure 5.7 shows the results of the particle elongation tests with the y-axis showing particle elongation and the x-axis showing the particles location in depth for three different microlens sizes of 10, 14, and 20 pxpμ from top to bottom. In these figures the red lines

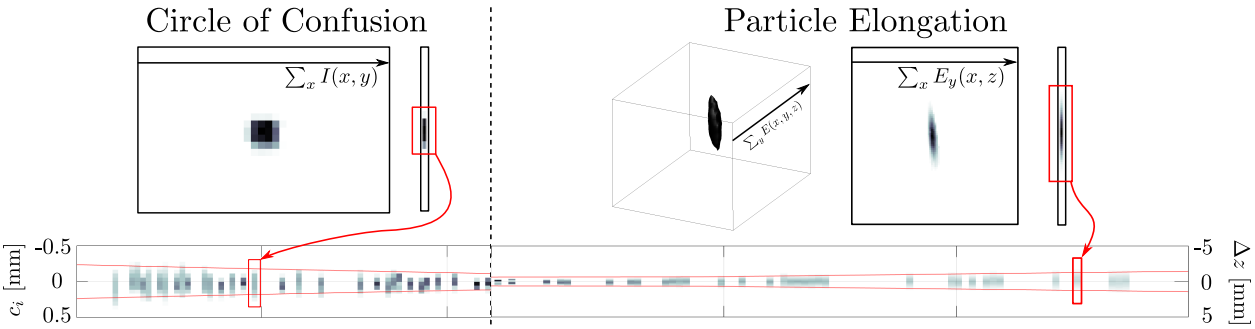


Figure 5.5: Schematic showing the construction of Figure 5.6 (Circle of Confusion) and Figure 5.7 (Particle Elongation).

indicate the full aperture depth of field calculation using the average circle of confusion generated from the previous analysis: two microlens diameters. It is shown that the full aperture depth of field is a good predictor for the particle elongation for all cases.

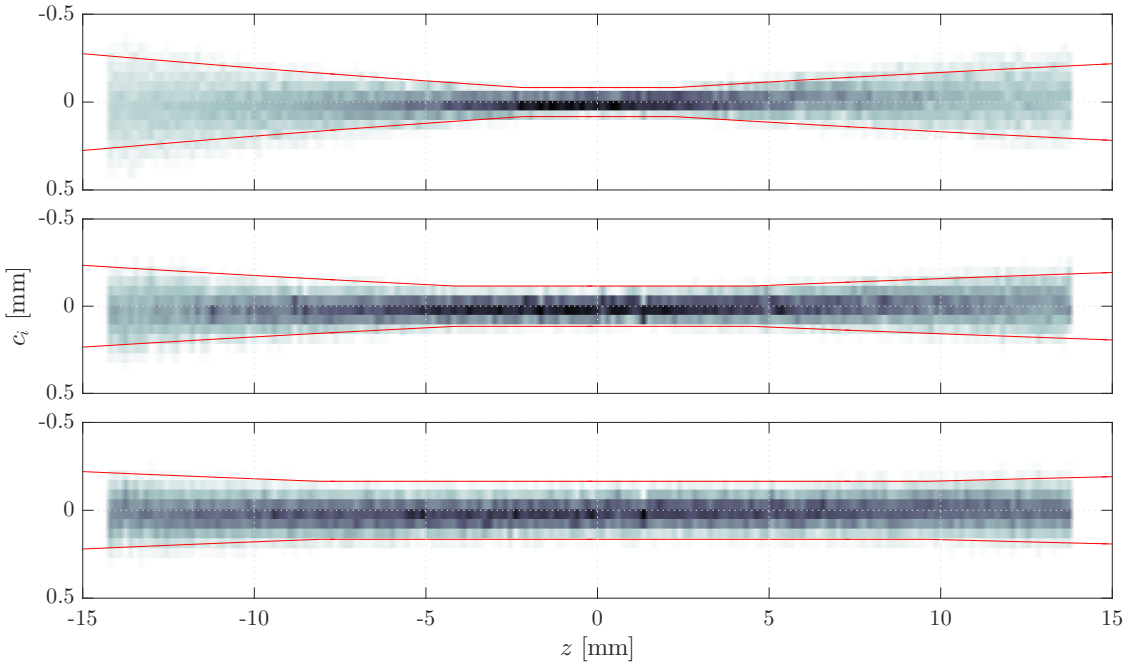


Figure 5.6: Illustration of image space circle of confusion generated from perspective views of synthetically generated particle images for three different microlens sizes. Red lines show the bounds of the predicted theory with a two microlens buffer as an estimate of the combined effect of discretization/interpolation.

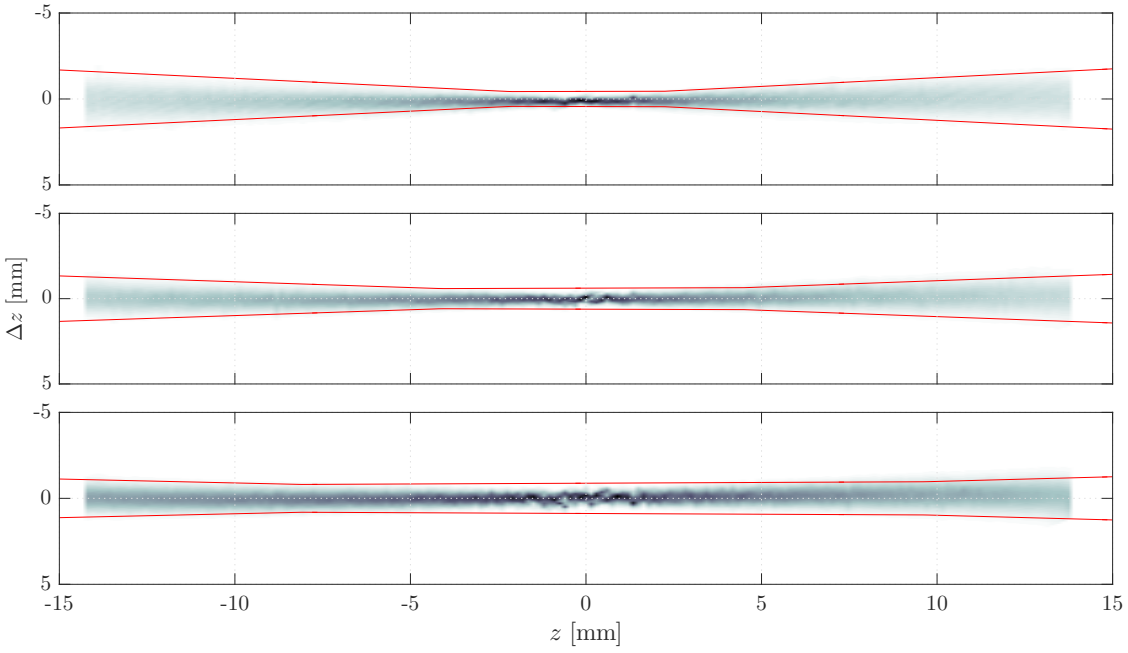


Figure 5.7: Illustration of reconstructed particle elongation as a function of volume depth for three different microlens sizes. Red lines show the bounds of the predicted theory of the reduced aperture depth of field with a one microlens buffer for the circle of confusion.

5.2 Effect on theoretical depth accuracy

The preceding analysis focused on the effect of the camera parameters, in particular the microlens size, on the size/shape of reconstructed particles. This section shifts the focus to the effect that the microlenses have on the theoretical positional accuracy of the particle reconstructions. In order to analyze this effect, a simplified two-dimensional light field is considered. By reducing the parameterization a simplified model for extracting an objects position in space directly from the light field can be derived. This model is based on the ray-space diagrams, introduced by Levoy and Hanrahan [50], which plot the two remaining light field coordinates against each other: s v. u. This process is shown in Figure 5.8, where two light rays (blue/green lines) emanate from a common point p, pass through the main lens at points u^- and u^+, and then strike the same microlens, s. In a ray-space diagram these two light rays are depicted as points (u^-, s) and (u^+, s) as shown in Figure 5.8 (*right*). All light rays captured by the light field can be shown in this manner, allowing for the direct analysis of the captured light field.

The ray-space diagram shown in Figure 5.8 illustrated the concept with a continuous sampling of the light field; however, when a light field is recorded by an actual plenoptic camera, it is recorded in discrete samples. This is illustrated in Figure 5.9 where each rectangle represents one such discrete sample of size: $\Delta s = p_\mu$ by $\Delta u = l_i p_p / f_\mu$. Additionally,

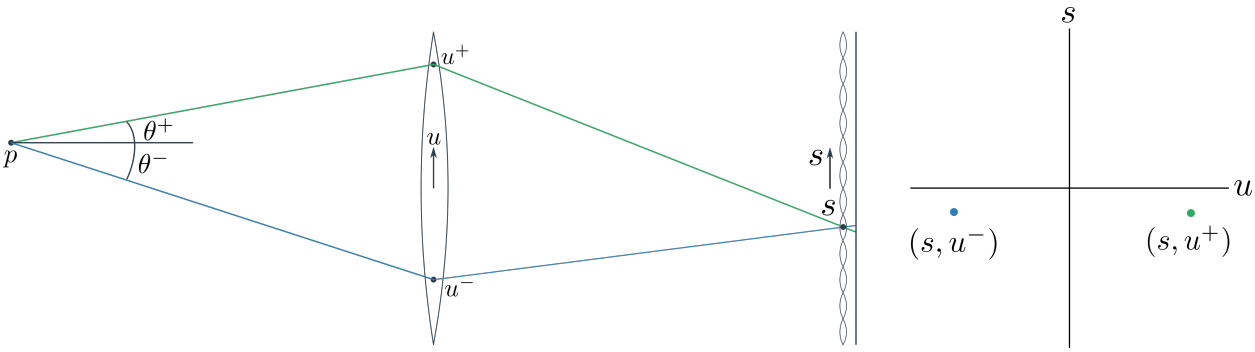


Figure 5.8: Schematic illustrating the calculation of ray space diagrams. (*left*) shows the projection of point p through the u plane onto the s-plane and (*right*) shows the resultant ray-space diagram.

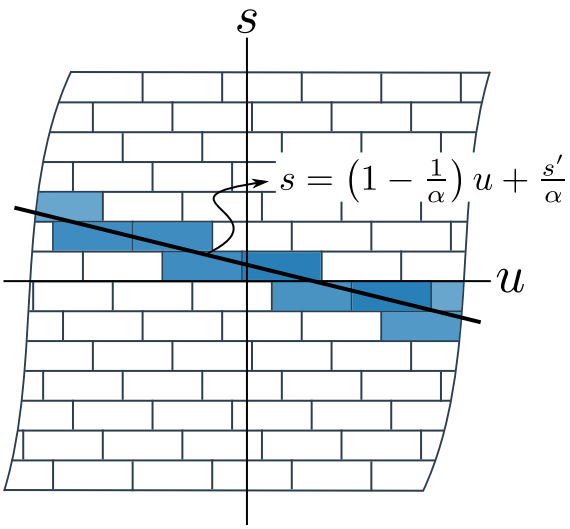


Figure 5.9: Ray-space diagram of a particle on a discretized light-field based on the microlens array.

this figure depicts a light field that contains all light rays emanating from a single point (*i.e.* the entire u distribution), instead of just the two shown previously. This image of a single object, in ray-space, is shown to form a continuous line spanning the u-coordinates whose equation is shown to be

$$s = \left(1 - \frac{1}{\alpha}\right) u + \frac{s'}{\alpha} \tag{5.7}$$

where s' is the image-space spatial location of point p. Both the slope and y-intercept of this equation are based on the position of the object. Therefore, if the slope and y-intercept can be determined, so too can the original position of the object. Using all locations (u, s) in the light field containing signal, the light field data can be fit, in a least-square sense, weighted by the intensity of the light field. Then from the measured slope the depth location, α, of the object can be determined followed by its spatial position using the y-intercept.

To test the performance of this technique, a simulation of the light field recording where the signal is artificially applied to the ray-space diagram was performed. This is done, by first defining the ray-space as a rectilinear grid with spacing $(\Delta u, \Delta s)$. Then using equation 5.7 a line is written from an initial $u_0 = -D/2$ to a final $u_0 = D/2$ position. At each position (10,000 in total) the (u, s) sample containing the position is found and its irradiance value

increased by 1. In addition, Poisson shot-noise is applied to ray-space image to add variation in the signal. The position can then be estimated via the procedure detailed earlier.

By repeating the simulation and evaluation of the ray-space model the plenoptic camera's ability to resolve the depth of an object can be determined for a range of depths/microlens sizes. As an illustration of this, Figure 5.10 shows the measured depth location versus the actual depth location from -15 to 15 mm (relative to the nominal focal plane) for 6 different microlens sizes at 1:1 magnification. It is shown that all microlens sizes appear to predict the position accurately; however upon closer inspection, near the focal plane, it is shown that smaller microlenses perform better, and, conversely, further away from the focal plane larger microlenses yield a more accurate solution. Moreover, the shape of each line appears to have the same shape, but scaled based on the microlens size. Therefore, it can be hypothesized that, the optimal microlens configuration is dependent on the depth of volume under investigation.

More information can be obtained by looking at the average error over varying volume spans, S_z, where S_z is the range of all depth positions, z between $-S_z/2$ and $S_z/2$, for each of the different microlens sizes. The depth error as a function of volume span is plotted in Figure 5.11 (*left*), which is constructed by calculating the mean of Figure 5.10 for different volume spans. Initially, the error increases as a function of volume span at the same rate for each microlens size. This region corresponds to the region inside the full-aperture DoF and corresponds to the 'flat' location around the optical axis illustrated in Figure 5.10. Once outside the DoF_m, the error decreases to a minimum and begins to steadily increase. Each microlens size is shown to have a unique volume span where it has the minimum average error; indicating that the selection of the microlens size is depth dependent. Earlier, it was hypothesized that the maximum volume depth would be a function of the reduced-aperture DoF. Figure 5.11 (*right*) shows the mean depth error plotted versus normalized volume span, where each line was normalized by it's microlenses DoF_p. It is shown that all lines collapse to a single curve whose minimum is roughly $S_z/DoF_p = 1.1$.

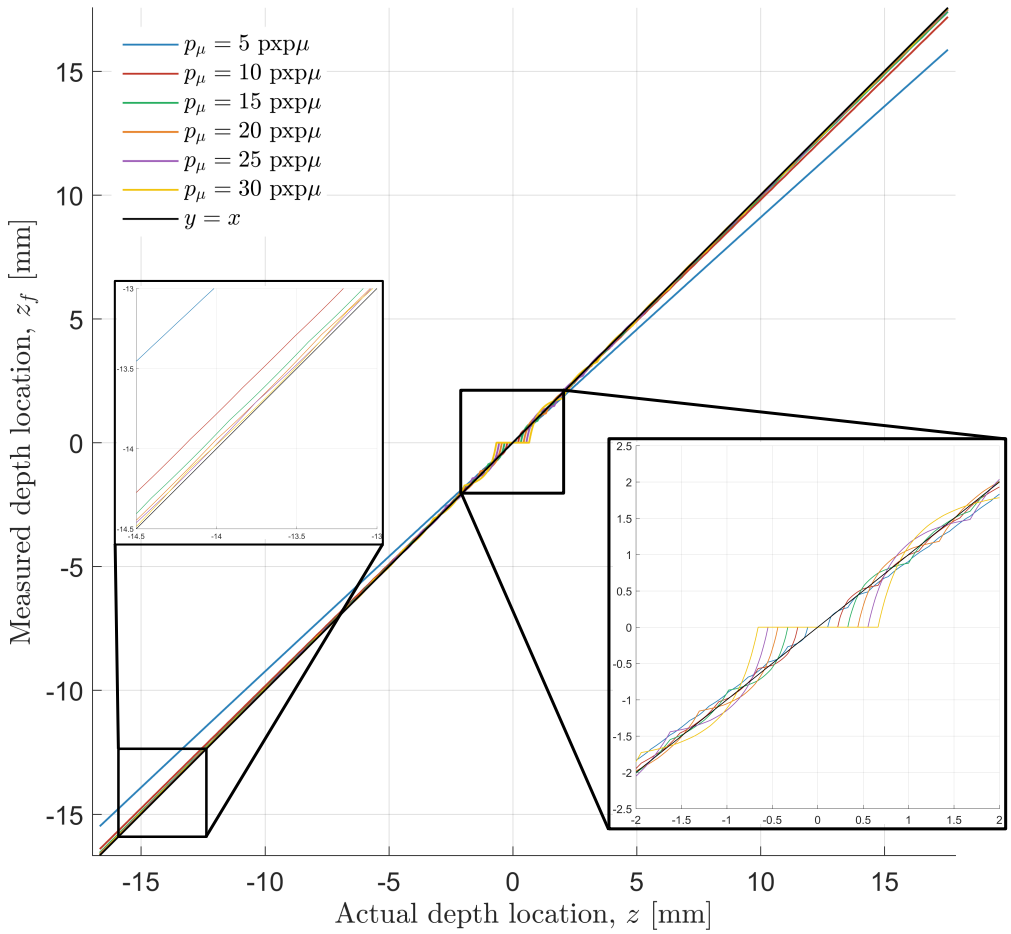


Figure 5.10: Measured depth location versus actual depth location for 6 different microlens sizes calculated via numerical simulation of ray-space particle triangulation.

Looking at the measured lateral x, y location versus volume span, plotted in Figure 5.12 (*left*), shows that the theoretical error for the in-plane components is on the order of 10^{-4}. It is noted that this data only includes the effect of shot noise and is otherwise void of real-world noise and bit-depth effects which would likely increase this error. In addition, the error associated with converting the plenoptic image into a ray-space diagram, as well as the effect of multiple particles, is completely ignored. If applied to real data this conversion would likely be the main source of error. However, even without these error sources it is shown, in Figure 5.12 (*left*), that the error increases as the particle's depth location moves away from the nominal focal plane. This is more evident in Figure 5.12 (*right*) which shows the mean error as a function of volume span. Where it is shown that each microlens size has

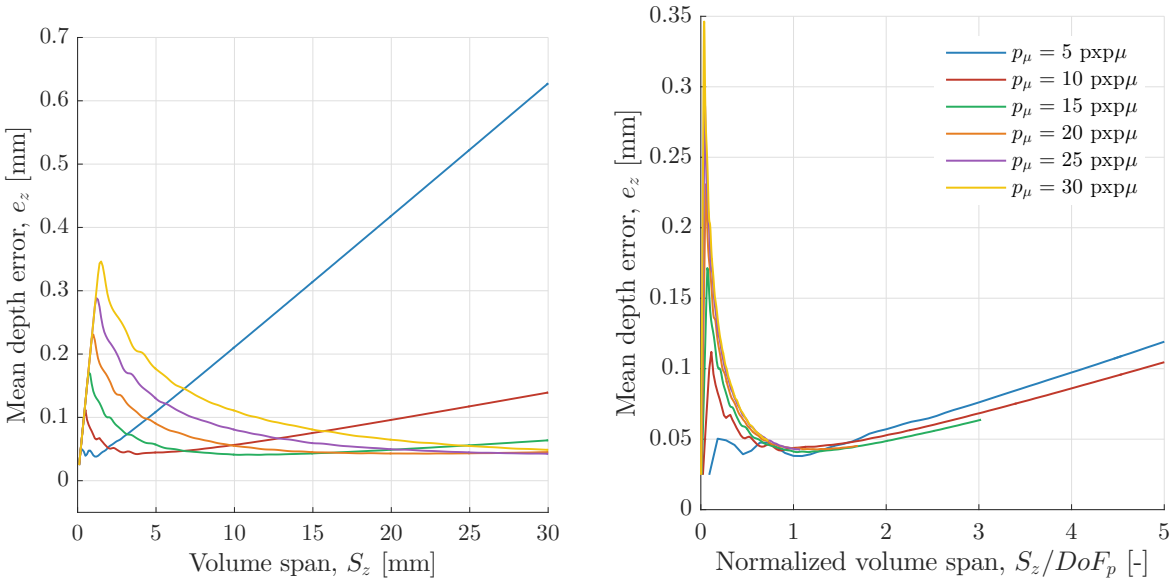


Figure 5.11: Mean depth error (*left*) plotted versus volume span, calculated by extracting the slope and intercept from simulated ray-space data. Mean depth error versus normalized volume span shown to indicate depth of field dependence (*right*).

an initial region of very high accuracy, corresponding to the region inside DoF_m, and then increases for the remainder of the volume spans. It is shown that all microlens sizes perform similarly with the exception of the smallest case. When compared to the lateral error, it can be inferred that, for the single-camera configuration, the depth error is dominant and can be used as the sole consideration for microlens size selection.

To further test the depth dependence, a wide range of microlens sizes were tested from 5 to 45 pxpμ at a variety of volume spans of 10 to 30 mm. For each span the following process was used to determine the optimal microlens size. First the error between the measured and the actual depth positions were calculated and averaged over the total volume span for each of the different microlens sizes. Then these values were fit using a rational expression of form

$$e(N) = \frac{aN^2 + bN + c}{N + d} \tag{5.8}$$

where *a*-*d* are coefficients determined via the fitting process. An example of the fitting is shown in Figure 5.13 (*left*). The optimal microlens can then be determined by finding the

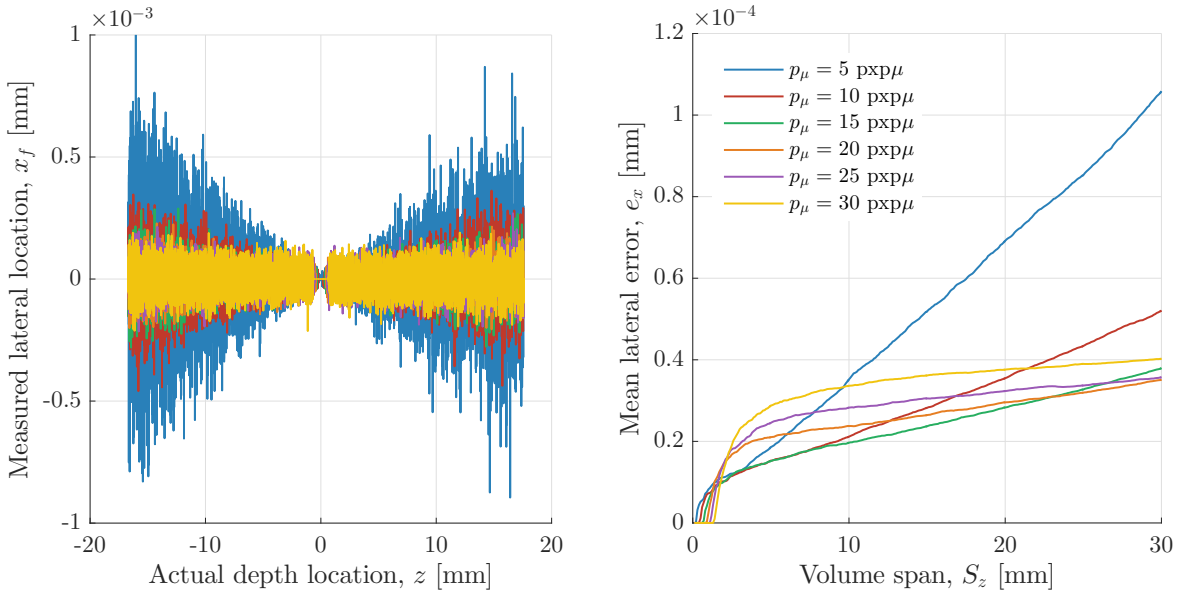


Figure 5.12: Measured lateral location versus actual depth location (*left*) and mean lateral error versus volume span (*right*).

N value corresponding to the minimum of equation 5.8 which is given by

$$N = \frac{-2da + \sqrt{(2da)^2 - 4a(bd - c)}}{2a} \tag{5.9}$$

While this value represents the absolute optimum it is noted that the difference between this optimum and the surrounding values is minimal, indicating the average depth error has a low sensitivity to the microlens size. The cumulative result of this process is shown in Figure 5.13 (*right*) where the optimal microlens size is plotted versus total volume span for four different magnifications. It is shown that as the volume span increases, the optimal microlens size also increases for all cases. In addition it is shown that as the magnification increases the optimal microlens size also increases.

As shown previously, the reduced aperture DoF plays a role in the maximum volume span over which yields an accurate reconstruction. By normalizing the x-axis of Figure 5.13 (*right*) by each points respective DoF_p the effect of this parameter can be identified. Figure

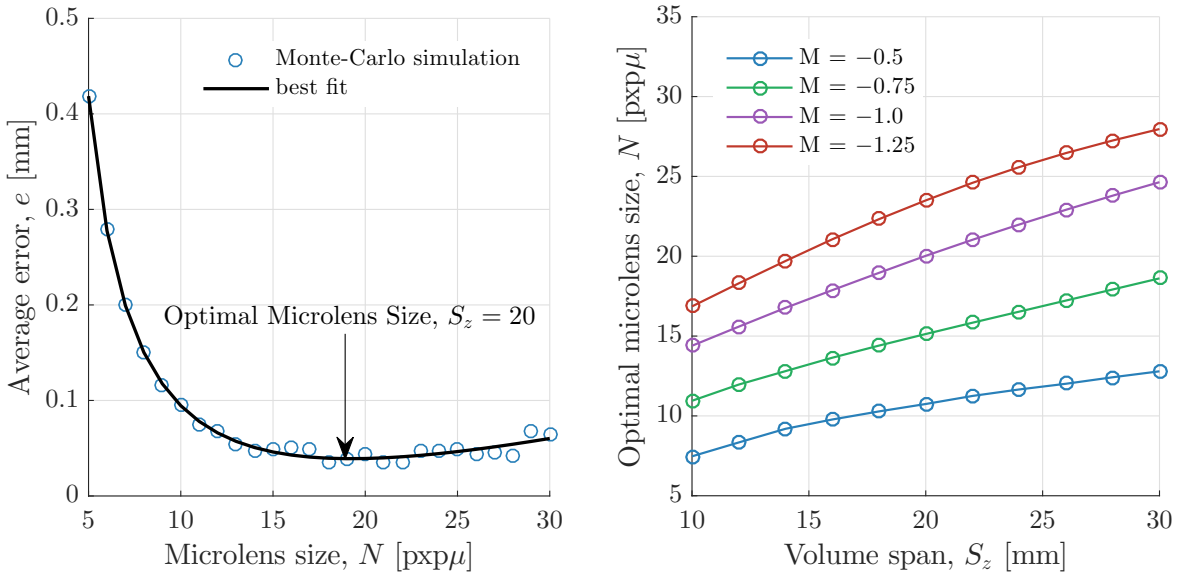


Figure 5.13: Least-square fitting of rational expression to particle reconstruction error as a function of microlens size for a volume span, $S_z = 20$ (*left*) and the optimal microlens size for a range of volume depths for 4 different magnifications (*right*).

5.14 shows this normalization and it is clear that the microlens which yields the optimal reconstruction is the one in whose DoF_p is 1 - 1.1 times the span of the volume in depth.

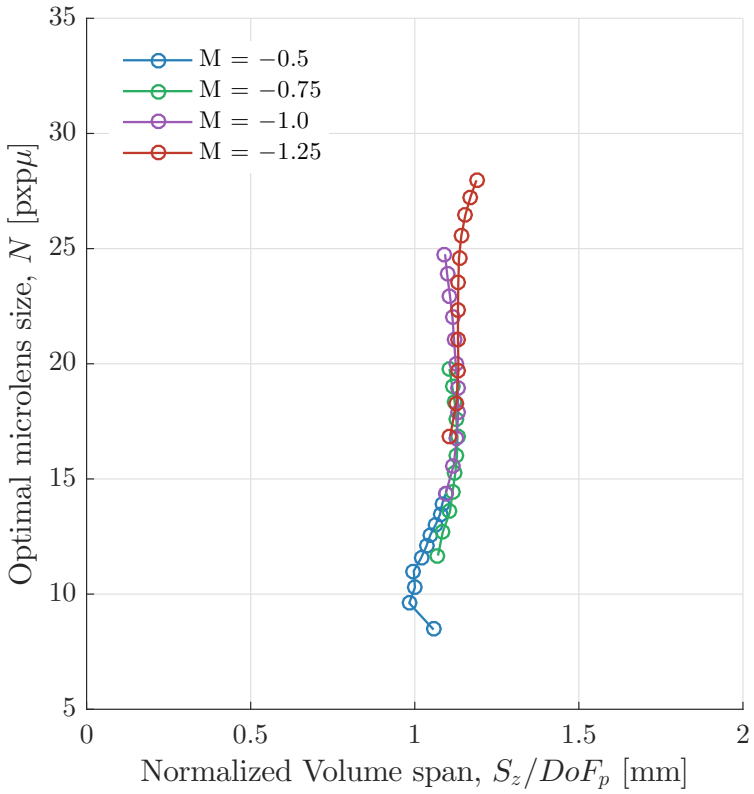


Figure 5.14: Optimal microlens size for a range of volume spans normalized by the reduced aperture DoF for 4 different magnifications.

5.3 Effect on Reconstruction Quality

To test the effect of microlens size on the quality of the particle reconstruction, the reconstruction quality factor, defined in Chapter 4, is used. Three different magnifications (M = -0.5, -0.75, and -1), and therefore volume sizes ($S_x = S_y = S_z$ = 40, 26, and 20 mm), are tested with varying microlens sizes at a constant particle density (N_p = 3000). Two volumes were generated at each condition using the MART algorithm (5 iterations, $\mu = 1$) and every volume was held to a constant 260x260x260 voxel grid. Adapting the grid to each different microlens was tested, for a single magnification, and yielded the same trends, but required a significant amount more computational time. Therefore only the constant voxel case is presented here. The second volume was displaced using the Gaussian ring vortex equations as in Chapter 4 and the final quality factor was calculated as the average of the two. To qualitatively illustrate the effect that the microlens size has on the particle

reconstructions, slices are extracted from each of the volumes from the $M = -1$ case and presented in Figure 5.15. Each sub-image represents a different microlens size starting from $N = 10$ ppμ in the top left and ending with $N = 29$ ppμ in the bottom right as indicated above each image. Two major trends can be seen from these images. The first is that the smaller microlenses yield smaller reconstructed particles near the focal plane of the camera, corresponding to $z = 0$ mm. The second is that near the edges of the volumes the smaller microlenses have more elongated reconstructed particles. It is hypothesized that the volume which will yield the most accurate vector field is the one with the best balance of these effects, that is the one with the most uniform particle sizes.

Figure 5.16 (*left*) shows the reconstruction quality factor as a function of microlens size for all three magnifications. It is shown that all three curves yield different optimal microlens sizes with the total difference between the best and worst solution being about 0.1 for all three cases. When compared to the theoretical solution shown in Figure 5.13 (*right*) it is shown that the predicted solutions are a close match to the synthetic cases. For $M = -0.5$ and $S_z = 40$ the theoretical result is 16.4, via extrapolation, and the synthetic result is 17.4. Similarly for $M = -0.75$ and $S_z = 26$ the theoretical result is 18.3 and the synthetic result is 18.3 and finally, for $M = -1.0$ and $S_z = 20$ the theoretical result is 20.0 and the synthetic result is 19.3. Further insight can be gained from Figure 5.16 (*right*) where the x-axis is now the normalized volume depth. It is shown that the optimal solutions for all three cases fall between 1.0 and 1.1 depths-of-field with the $M = -0.5$ case being closer to 1 which is the same as the synthetic result shown in Figure 5.14.

5.3.1 Synthetic Vector Field

The final synthetic test was designed to illustrate the effect of microlens size on the final vector field. Using the volumes from the previous section, which were displaced via the Gaussian ring vortex equation, and the cross-correlation analysis presented in Chapter 2 the final vector field can be determined. It is noted that the Gaussian ring vortex was scaled

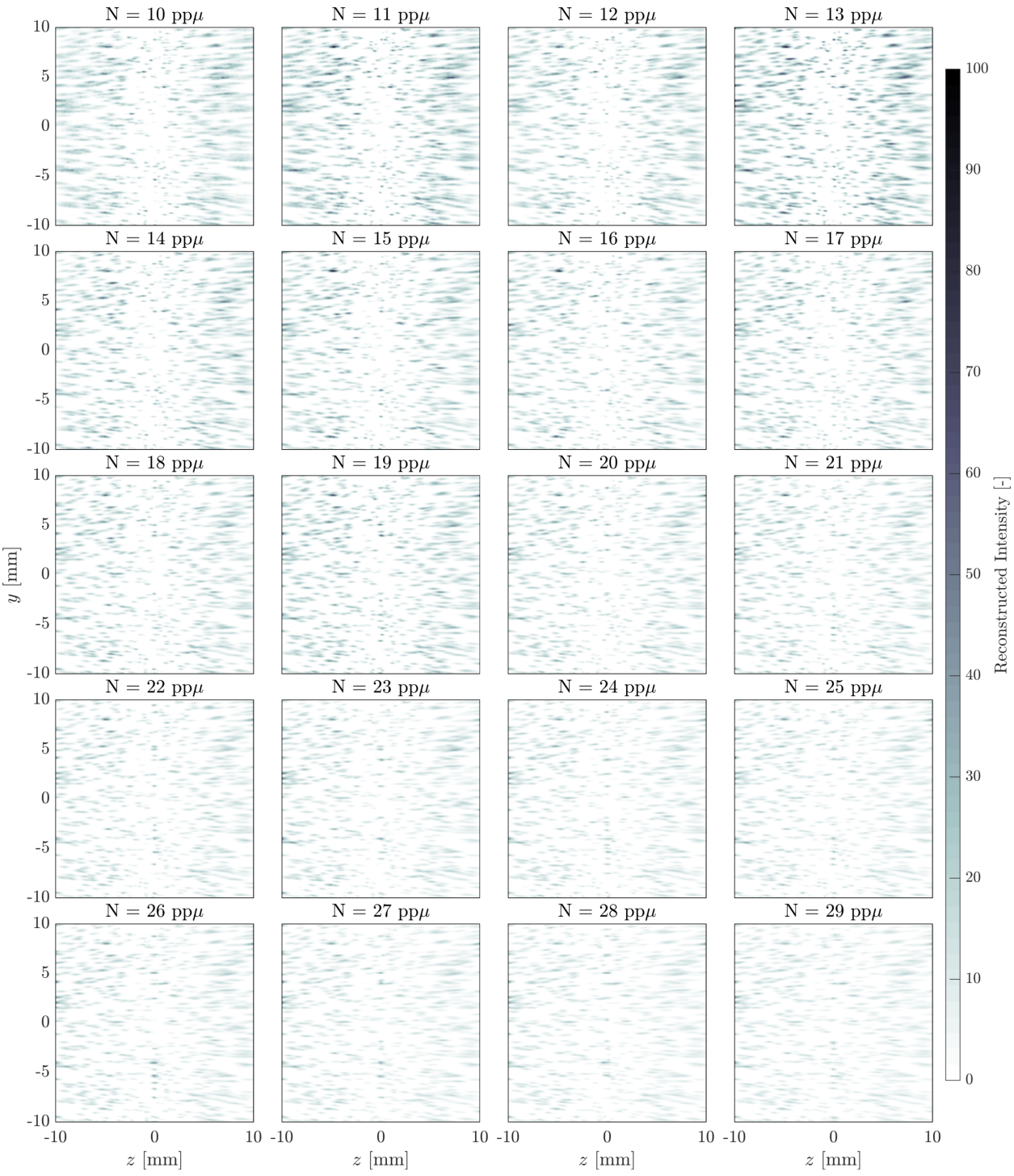


Figure 5.15: Slices extracted from reconstructed volumes ($M = -1$) showing the effect of microlens size on the particle reconstructions. Microlens size indicated above each image increasing from top left to bottom right.

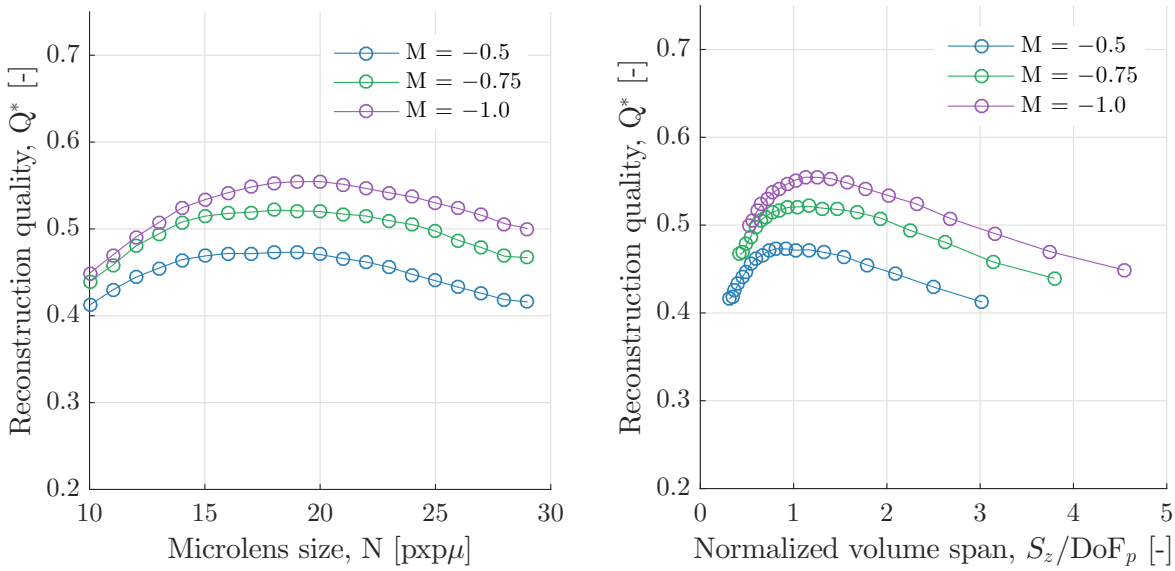


Figure 5.16: Reconstruction quality factor vs. microlens size (*left*) and normalized volume depth (*right*).

with magnification such that the vortex was roughly the same relative size for each magnification. This vector field was then compared to a "known" solution which was generated by creating two volumes using the exact position of the particles and processing them using the same cross-correlation software exactly as was done in Section 4.3.4. The volumes were processed using 4 passes of the cross-correlation algorithm starting with 64x64x64 windows and finishing with 32x32x32 windows with 75% overlap. Slices of the vector fields are presented in Figure 5.17 showing the v component of velocity and Figure 5.18 showing vorticity magnitude. Unlike the reconstructed volumes it is difficult to notice any trends with respect to the size of the microlenses. Some volumes show more spurious vectors than others but they occur randomly regardless of the microlens size.

The total vector velocity error when compared to the "known" solution, using RMSE, is shown in Figure 5.19 (*left*) as a function of microlens size and Figure 5.19 (*right*) as a function of normalized volume depth. It is shown that the minimum error is somewhat correlated to the minimum reconstruction accuracy; however they are not an exact match and there is more noise in the curves, including multiple minimums for the $M = -0.5$ and $M = -0.75$

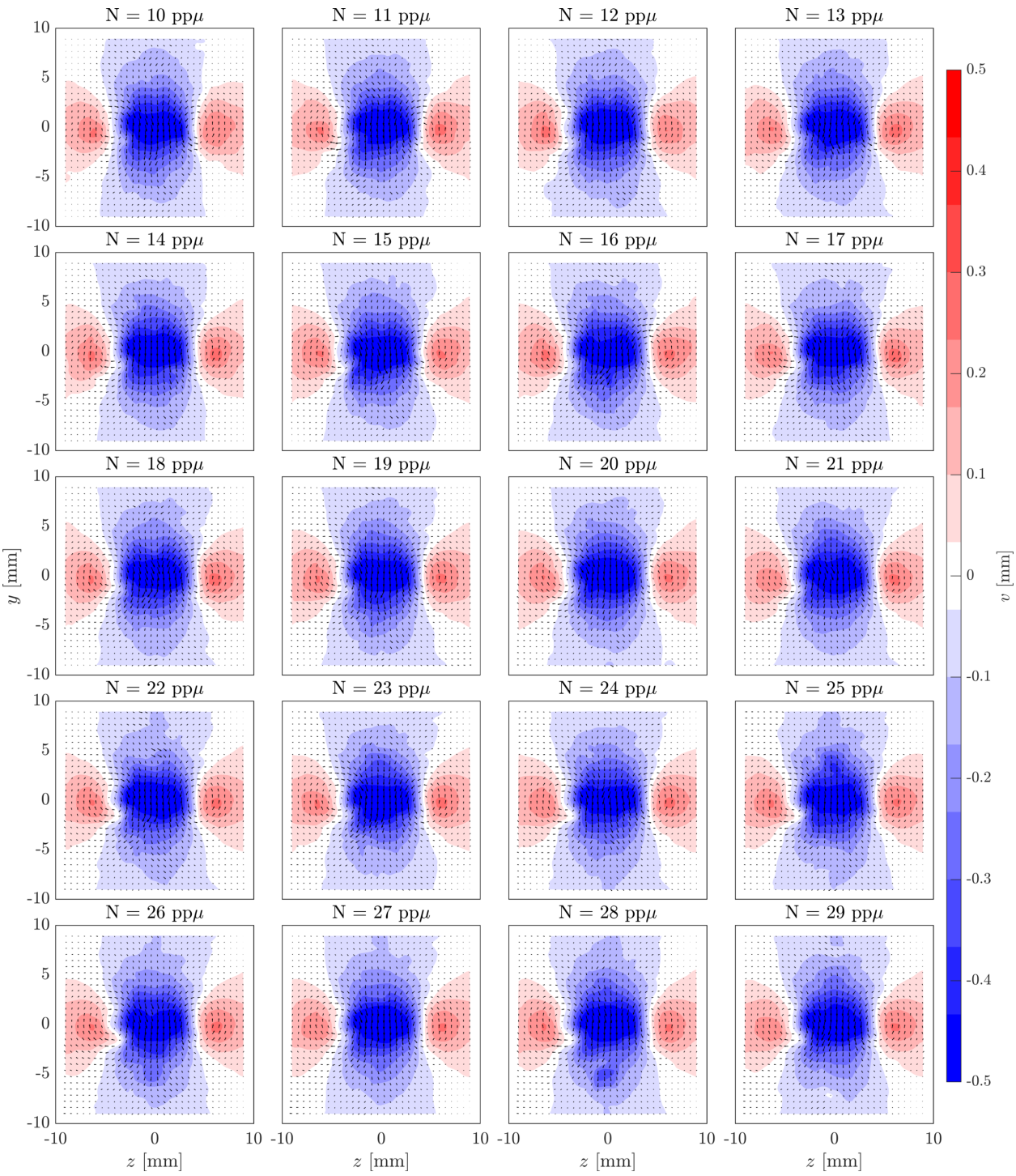


Figure 5.17: Slices extracted from reconstructed vector fields ($M = -1$) showing the effect of microlens size on the v-component of velocity. Microlens size indicated above each image increasing from top left to bottom right.

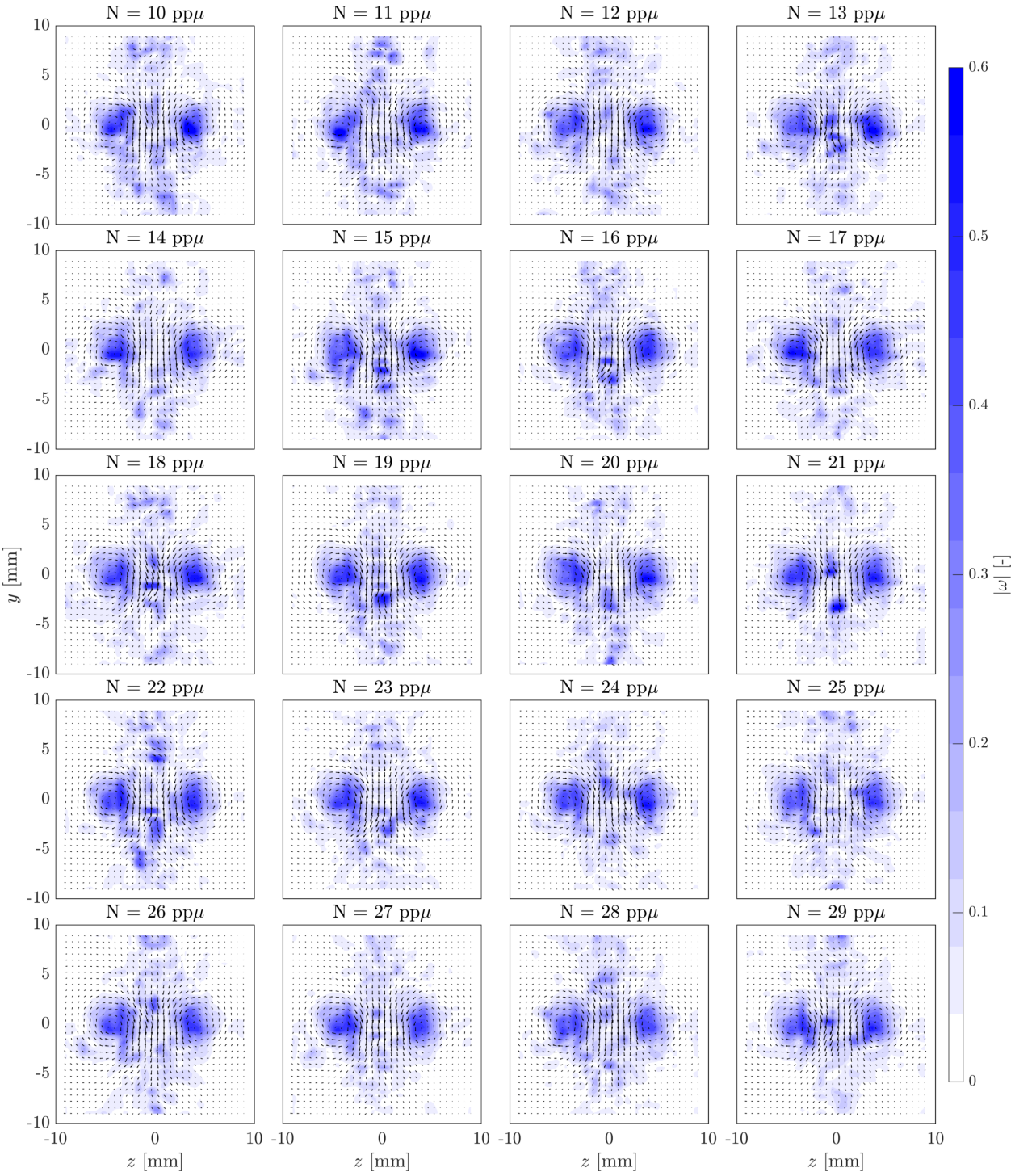


Figure 5.18: Slices extracted from reconstructed vector fields ($M = -1$) showing the effect of microlens size on the vorticity magnitude. Microlens size indicated above each image increasing from top left to bottom right.

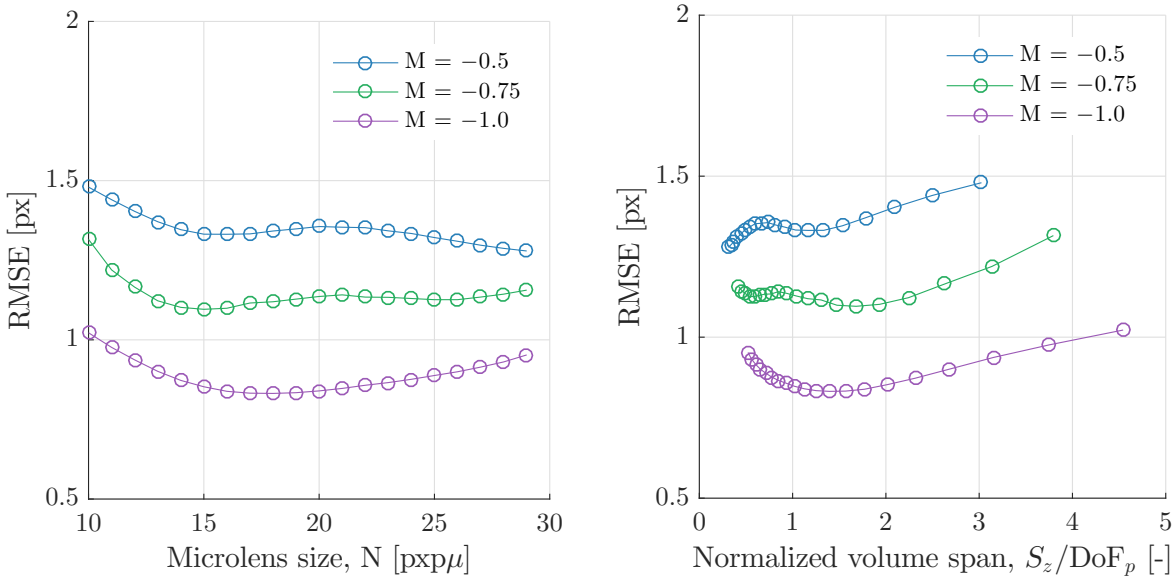


Figure 5.19: RMSE of vector field vs. microlens size (*left*) and normalized volume depth (*right*).

cases. As the quantitative data suggested the difference between the different microlens sizes is minimal with roughly a 0.2 pixel difference between the minimum and maximum errors for each case. This is likely due to the robustness of the cross-correlation algorithm being able to mitigate the effect of the particle elongation. Additionally it is shown that the minimum for each case is between 1 and 2 depths-of-field which is consistent with the reconstruction quality data.

5.4 Experimental Test

The experimental validation of the optimal microlens size theory was conducted in the same facility as discussed in Chapter 4 using the same laser, and flow conditions. The major difference is the addition of 2 cameras placed on the remaining sides of the square tank as shown in Figure 5.20. These cameras differ from the two used previously by having different sizes microlenses (pixels-per-microlens) relative to their CCD. Two different plenoptic camera models were available for testing, the first was the prototype 16 Mp camera used in

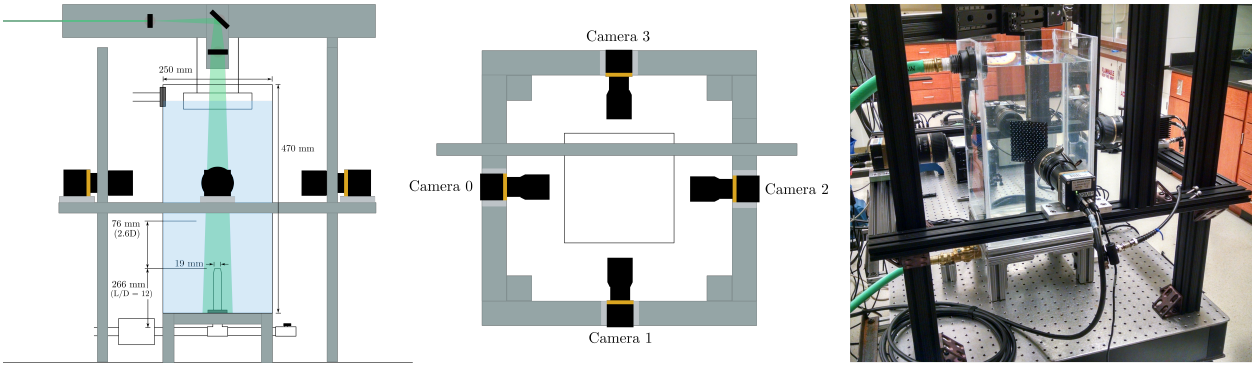


Figure 5.20: Experimental Arrangement.

Fahringer *et al.* [58] and the second is the second generation 29 Mp camera discussed previously. In addition to the image sensor differences, these cameras also use different microlens arrays. Therefore, it is possible to swap the microlens arrays and yield two new camera configurations. For this experiment, three different configurations (microlens sizes) will be tested: N = 10, 14, and 20 pxpμ where the 10 and 20 pxpμ were created by swapping the microlens arrays from the prototype and second generation cameras and the 14 pxpμ is the standard second generation camera. The full parameters of these cameras are given in Table 5.1. To create the validation measurement a two-camera plenoptic system will be used, as described in Chapter 4, using Camera's 2 and 3. This allows for the comparison of three different configurations on instantaneous data instead of a purely statistical comparison. In addition, it allows for the particle reconstruction quality to be determined

5.4.1 Reconstruction Analysis

The raw images were reconstructed using the MART algorithm with $\mu = 0.2$ for 5 iterations. The reconstructed volumes were 100x60x100 mm^3 in physical size and were discretized into 500x300x500 voxels. Two metrics are used to determine the best microlens size to create an accurate reconstruction. The first is the reconstruction quality factor, Q^*, as defined in equation 4.10, with the comparison volume being generated from the two-camera data. The second method is known as the normalized intensity variance, NIV, which is a measure of the sparsity of the reconstruction as detailed by Novara and Scarano [78]. A

Table 5.1: Camera Parameters

	Camera 0	Camera 1	Camera 2	Camera 3
Base Camera	Imperx B4820	Imperx B6620	Imperx B6640	Imperx B6640
Number of Pixels	4904 x 3280	6600 x 4400	6600 x 4400	6600 x 4400
Number of Microlenses	471 x 362	289 x 193	471 x 362	471 x 362
Microlens Packing	Hexagonal	Rectangular	Hexagonal	Hexagonal
Pixel Pitch	0.0074 mm	0.0055 mm	0.0055 mm	0.0055 mm
Microlens Pitch	0.077 mm	0.125 mm	0.077 mm	0.077 mm
Pixels-Per-Microlens	10	20	14	14
Microlens F-number	4	4	4	4
Main Lens Focal Length	60 mm	60 mm	60 mm	60 mm
Main Lens F-number	2.8	2.8	2.8	2.8
Nominal Magnification	-0.4	-0.4	-0.4	-0.4
DoF_p	38 mm	118 mm	55 mm	55 mm

higher NIV indicates a more sparse reconstruction with higher-amplitude peaks; whereas a low NIV indicates a reconstruction the opposite: a dense particle field with low-amplitude peaks which is characteristic of ghost particles. This value is calculated the variance of the intensity within the volume and normalizing by the mean as given by

$$\text{NIV} = \frac{\sqrt{\frac{1}{N}\sum_j^N (E_j - \bar{E})^2}}{\bar{E}} \tag{5.10}$$

This metric is particularly useful for experimental analysis since it does not rely on a known quantity, and will allow for the comparison between the single camera and two camera reconstructions.

5.4.2 Velocity Analysis

To obtain the 3D vector field each volume pair was processed using the cross-correlation scheme utilizing 4 passes starting with 64x64x64 windows and finishing with 32x32x32 windows with 75% overlap resulting in 58x33x58 vectors with a spatial resolution of 6.4 mm/vector. Analysis of the vector fields was performed with two metrics. First the root mean square error (RMSE) of the velocity and vorticity field was calculated taking the two camera data

Table 5.2: Results of the experimental vortex reconstruction quality and vector field accuracy analysis. Values shown are the average of 200 volumes and 100 vector fields.

	Camera 0	Camera 1	Camera 2	Camera 3	Cameras 2/3
Q^*	0.019	0.148	0.243	0.265	-
NIV	14.44	14.73	13.98	14.04	31.08
e_{vel}	0.846	0.912	0.781	0.797	-
e_{vort}	0.077	0.079	0.071	0.069	-
e_{∇}	0.251	0.290	0.281	0.287	0.212

as the truth measurement. The second metric allows for the vector fields to be evaluated without a comparison field by estimating the velocity error from the error in the divergence. Since this flow is incompressible, any measured velocity divergence is attributed to measurement error. For such a flow, the velocity error can be related to the standard deviation of the divergence field by

$$e_{\nabla} = \frac{\sigma_{\nabla}}{\sqrt{\frac{3}{2h^2}}} \tag{5.11}$$

where σ_{∇} is the standard deviation of the divergence of the velocity and h is the vector spacing as described by Atkinson *et al.* [79].

5.4.3 Experimental Results

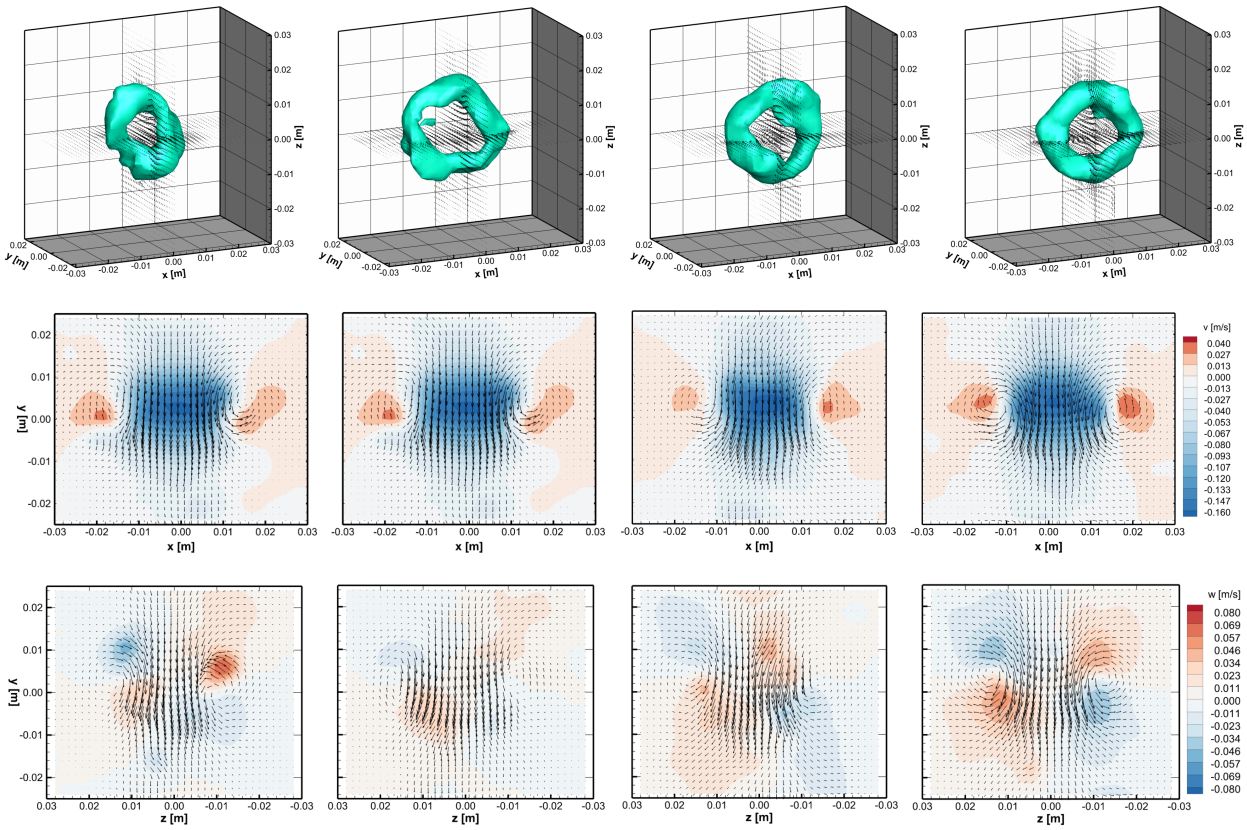
The results of the reconstruction quality tests are shown in Table 5.2 for the four single camera configurations as well as the two camera configuration. The tabulated values indicate the average values for all 200 volume for each case. For the experimental reconstruction quality factor cameras 2 and 3 have the highest values of around 0.25 followed by camera 1 with 0.15, then camera 0 with 0.02. There is little difference in the normalized intensity variance results with all of the single camera cases having values within 0.75 of eachother, while the two camera configuration resulted in a NIV of greater than two times the single camera configurations. This is expected since the two camera reconstructions do not exhibit the same particle elongation as the single camera configuration. Overall, the results indicate that the camera 2/3 reconstructions are more accurately reconstructing the two camera

particle field (it is noted that cameras 2 and 3 are used to create the two camera particle field) and all the single camera configurations produce roughly the same NIV values with cameras 1 and 0 holding a slight edge over cameras 2/3 with the two-camera configuration being clearly superior.

An example vector field is presented in Figure 5.21 for 4 cases: camera 0, 1, 3, and the two camera configuration. Presented in the top row is two slices of the vector field along with a vorticity magnitude iso-surface at 20 s^{-1}. The middle row shows an xy slice of the vector field with streamwise velocity contours, and the bottom row shows a zy slice with cross-stream velocity contours. The results show that cameras 1 and 3 reconstruct a more similar vortex structure to the two-camera result than camera 0 which is compressed along its depth direction. The slices show similar features across the board, with the camera 3 results showing slightly less spurious vectors; however all the single camera cases show a compressed vortex core region when compared to the two camera results.

Further analysis is given in Table 5.2 with the velocity and vorticity RMS as well as the error based on divergence. It is shown that the vector fields generated by cameras 2 and 3 have lower RMS values (0.79 for velocity and 0.07 for vorticity) than the camera 0 data (0.846 and 0.077) which is better than the camera 1 data (0.912 and 0.079). The error based on divergence indicates that the camera 0 data has the best results of 0.251 followed by the camera 2/3 data with 0.284 and then the camera 1 data with a value of 0.29. The two camera case has an error based on divergence of 0.212. As expected, the camera 2/3 results show the best qualitative and quantitative vector results; however the results of all 4 single camera cases are similar which is expected given the results of the synthetic data.

Overall, both the reconstruction quality and vector field analysis indicate that the camera 2/3 single camera reconstructions produce the most accurate reconstruction and vector fields, which is consistent with the preceding analysis since cameras 2/3 had the most optimal microlens size based on their single-pixel depth of field. The other two cases, camera 0 and camera 1, were very sub-optimal in terms of their microlens size having $\mathrm{DoF}_p = 1.6S_z$



(a) Camera 0 (b) Camera 1 (c) Camera 3 (d) Two Camera

Figure 5.21: Example instantaneous velocity field shown as 3D vorticity magnitude iso-surface (*top*), slices showing velocity vectors with streamwise velocity contours (*middle*), and slices showing cross-stream velocity contours (*bottom*) for four different camera configurations.

and $0.5S_z$ respectively. In spite of this, these configuration were able to produce results comparable to the more optimal case indicating that while there is an optimal microlens size the resulting vector fields are not very sensitive to it. Therefore, a single plenoptic camera design, with a static microlens size, should be a robust system that can handle a wide variety of different volume sizes with relatively little change in performance.

5.5 Microlens Selection Summary

This chapter provided theory, numerical analysis, and experimental data in an attempt to determine the optimal microlens size for any given experiment. Using this information

the following guide provides a step-by-step method for determining which size microlenses should be selected for a given experiment. As an example the experimental configuration given in Table 5.1 will be used

1. Determine the size of volume S_x, S_y, and S_z you wish to reconstruct.

$$S_x = S_y = S_z = 60 \text{ mm}.$$

2. Using S_x and S_y and the size of you CCD you can determine the magnification of the imaging system.

$$M = -25.4/60 \approx -0.4$$

3. From the magnification and some knowledge of the working distance restrictions the appropriate focal length lens can be determined using the thin lens equation (3.15) and magnification relation (3.16). For our case we had approximately 200 mm between the edge of the lens and the center of our water tank.

$$f_m = \frac{l_o M}{M-1} = \frac{200(-0.4)}{-0.4-1} = 57.1 \approx 60 \text{ mm}.$$

4. A microlens f-number should be selected such that f-number matching can be achieved (Equation 3.17). Since our microlenses were all fixed at $f/4$ we chose a main lens f-number of 2.8. The reverse of this math would be:

$$(f/\#)_\mu = (f/\#)_m(1-M) = 2.8(1--0.4) = 3.92 \approx 4.$$

5. Finally solve equation 5.3 for the reduced-aperture DoF ($d_a = \Delta u = l_i p_p / f_\mu$) setting $\text{DoF}_p = 1.1 S_z$. This was performed by fitting a series of microlens sizes to a 5th order polynomial where $x = N_{pxp\mu}$ and $y = DoF_p - 1.1S_z$, with the real root of this polynomial yielding the optimal microlens size of $15.68 \approx 16$.

Chapter 6

Conclusion

In order to mitigate the increased error along the optical axis of a single camera plenoptic-PIV system a second camera was added in a stereoscopic configuration. The framework for adding the second camera including the modification of the reconstruction algorithms and the camera calibration was presented. Four reconstruction algorithms were tested: three based on the refocusing capabilities of the plenoptic cameras and one based on tomographic reconstruction. It was found that the MART algorithm was, in general, the best performer especially at higher particle densities. The stability of the MART algorithm was tested, concluding that it was stable for a wide range of relaxation parameters with the only difference being the rate of convergence.

Several tests were performed to determine the optimal configuration with the results showing a 2-camera configuration with angular separation between 40-120 degrees reconstructed with a 1:1 voxel to pixel ratio to be the ideal configuration. In addition the effect of calibration error was determined to be highly detrimental, especially when the error was > 1 microlens where the reconstruction quality drops below the desired 0.75. In addition, it was found that particles added outside the volume are slightly more detrimental that ones added inside the volume and that using sliding mean subtraction can effectively mitigate image noise.

The presence and effect of reconstruction artifacts, known as ghost particles, was tested. It was shown that as particle density increases, the number of detected true particles decreased, the number of detected ghost particles increased, and the ratio of ghost-to-true

particle intensity increased. In addition, the RMSE of particle position was presented showing similar performance for all four algorithms, with the error generally increasing as a function of particle density.

Velocity errors were examined using a synthetic Gaussian ring vortex as the flow field. It was found that the addition of a second camera decreased the RMS velocity error from 0.61, 0.16, and 0.59 to 0.14, 0.09, and 0.14 for the u, v, and w components respectively. The algorithms performed similar to each other with MART and integral refocusing yielding the lowest RMSE. Additionally, the technique was tested experimentally on a ring vortex generated in a purpose built water tank where an instantaneous velocity field produced the expected ring vortex vorticity structure for the modified refocusing methods as well as the MART algorithm where it was shown that the MART algorithm performed best in terms of relative uncertainty. In addition, comparison were made to the single camera configuration where it was shown that the addition of a second camera drastically improved the out-of-plane velocity accuracy of the technique.

Overall, the addition of a second plenoptic camera drastically increased the reconstruction quality (by a factor of 2) and decreased the vector velocity error by a factor of 4, by increasing the overall experimental and computational complexity. With the addition of the second camera more optical access is required as well as precise alignment of the two cameras; however, their is no need for Scheimpflug mounts therefore the experimental complexity is on the order of a stereoscopic PIV system. The computational cost increases linearly as the number of cameras increase; thus for particularly large data sets or dense reconstruction grids the, faster, refocusing-based methods may become more desirable. Even so, the benefit of the second camera to produce accurate, 3D/3C velocity measurements is evident and should be the standard for plenoptic-PIV provided the optical access requirements can be met.

Following the analysis on the addition of a second plenoptic camera, the effect of the microlens size was studied. It was determined through numerical analysis that the optimal

microlens for a given experiment was one in which the single-pixel depth of field spanned 1.1 times the volume depth. Synthetic tests and an additional experiment were performed to validate this result. It was found that while the optimal microlens size was the same as the numerical analysis predicted, the performance loss for using a non-optimal microlens was minimal. Therefore a single microlens array, with a fixed size, should be able to be used for a multitude of experiments with little loss in performance.

The plenoptic camera, and its ability to perform quantitative 3D/3C velocimetry measurements was tested in this thesis. The future of the technology looks bright as it offers fundamentally unique features, such as the ability to obtain 3D information behind a single aperture. Unfortunately, this technology has not seen widespread adoption throughout the fluid diagnostic community, which is likely tied to the inability to purchase a scientific grade version of this camera. For example, in this work, we made our own camera by designing the microlens assembly and sourcing the parts ourselves. Most labs, especially ones more focused on the fluid physics, are not willing to manufacture the cameras and write all the necessary code for data processing. Therefore in order for plenoptic-PIV, or other spin-off technologies, to see wide-spread adoption a commercially available turn-key solution must become available. For this to happen the technology must be mature enough, or present a unique capability, for a company to invest the time and money to develop the product. In my estimation, this will require the development of a time-resolved Lagrangian particle tracking algorithm, ala shake-the-box [47], for single-camera plenoptic-PIV. This technology would provide capabilities, by only using a single camera, that other competing technologies can not match and allow access to facilities where multi-camera systems could not make measurements. A second or third camera could then be added as a value option for increased accuracy, or to provide the ability for scalar-field reconstruction.

Chapter 7

Hexagonal Interpolation

Mathematically, the interpolation algorithm for the hexagonally arranged microlens array is more complex than the rectilinear case. The procedure detailed here finds the nearest four microlenses to the point of interest and uses a weighted distance calculation to determine each microlenses contribution. Schematically this problem is represented in Figure 7.1 where point (x_2, y_2) represents the location of interpolation and (x_0, y_0) is the nearest microlens. For the hexagonal layout there are two types of rows the point can fall on an *offset* and a *not offset* row. These rows are defined based on the position of the first microlens in the row relative to the row above and below it. An offset row starts to the right $(+x)$ of the rows adjacent to it. Due to this effect assigning an indexed value is non-trivial and requires careful treatment. The convention used in this work treats the array as rectilinear where each element's index is simply the numbered row where it resides, and the number of microlenses preceding it in its row. To correct for this indexing a matrix of offsets are used during the interpolation process.

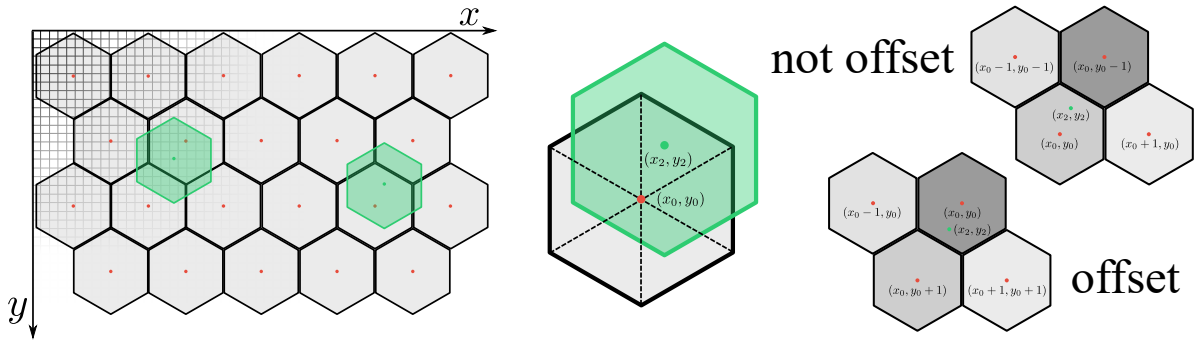


Figure 7.1: Schematic of interpolation on a hexagonally arranged microlens array. Illustration on *left* shows two point to be interpolated with one being on an offset row, *center* shows the different sections corresponding to different groups of neighbors, and *right* shows the two interpolants from the *left* illustration and their corresponding neighbors with indicies.

Given the point of interest (x_2, y_2) and arrays containing the coordinates of the microlens positions X and Y the algorithm is as follows. First we determine if the row is offset by checking the following expression $Y(x_0, y_0) > Y(x_0, y_0 + 1)$ with $x_0 = \text{Round}(x_2)$ and $y_0 = \text{Round}(y_2)$. If true, then the row is offset. Then we determine which section, illustrated as dashed lines in Figure 7.1, *center*, the point falls into. This is done by the following relations.

$$\Delta x = x_2 - x_0$$

$$\Delta y = y_2 - y_0$$

$$\theta = \text{atan2}(dy, dx) + 30\pi/180;$$

After converting to degrees the section can be determined by.

$$\text{section} = \lfloor \theta_d / 60 \rfloor$$

Once the section and whether or not the row is offset is computed the four microlenses to be interpolated can be found. The relative positions for each of the four microlenses are shown in table 7.1 as relative coordinates for all possible configurations.

Table 7.1: Relative location of the four nearest microlenses to point (x_0, y_0) for both not-offset and offset rows. The offset section only shows the ones that change.

	Not-Offset								Offset		
Section	x_0	y_0	x_1	y_1	x_2	y_2	x_3	y_3	x_1	x_2	x_3
1	0	0	0	1	1	0	0	-1	1	1	1
2	0	0	1	0	0	-1	-1	-1	1	1	0
3	0	0	0	-1	-1	-1	-1	0	1	0	-1
4	0	0	-1	-1	-1	0	-1	1	0	-1	0
5	0	0	-1	0	-1	1	0	1	-1	0	1
6	0	0	-1	1	0	1	1	0	0	1	1

We can now determine the distance between each microlens center and the point (x_2, y_2).

$$
\begin{aligned}
D_{x_0} &= x_0 - x_2 \\
D_{x_1} &= x_0 + |0.5 \cdot (y_1 - y_0)| \cdot (-1 \cdot \text{isOffset}) - x_2 \\
D_{x_2} &= x_0 + |0.5 \cdot (y_2 - y_0)| \cdot (-1 \cdot \text{isOffset}) - x_2 \\
D_{x_3} &= x_0 + |0.5 \cdot (y_3 - y_0)| \cdot (-1 \cdot \text{isOffset}) - x_2
\end{aligned}
$$

$$
\begin{aligned}
D_{y_0} &= (y_0 - y_2) \cdot 0.5 \cdot \sqrt{3} \\
D_{y_1} &= (y_1 - y_2) \cdot 0.5 \cdot \sqrt{3} \\
D_{y_2} &= (y_2 - y_2) \cdot 0.5 \cdot \sqrt{3} \\
D_{y_3} &= (y_3 - y_2) \cdot 0.5 \cdot \sqrt{3}
\end{aligned}
$$

$$
\begin{aligned}
D_0 &= \sqrt{D_{x_0}^2 + D_{y_0}^2} \\
D_1 &= \sqrt{D_{x_1}^2 + D_{y_1}^2} \\
D_2 &= \sqrt{D_{x_2}^2 + D_{y_2}^2} \\
D_3 &= \sqrt{D_{x_3}^2 + D_{y_3}^2}
\end{aligned}
$$

Then using the distances to each microlens center the final weighting value can be computed as the overlapping area of two circles as a function of distance, with $r = 0.5$ being the radius

of the circle.

$$A_0 = \left(2 \cdot r^2 * \cos^{-1}(D_0/(2r)) - (D_0/2) \cdot \sqrt{4 \cdot r^2 - D_0^2}\right) /(\pi r^2)$$
$$A_1 = \left(2 \cdot r^2 * \cos^{-1}(D_1/(2r)) - (D_1/2) \cdot \sqrt{4 \cdot r^2 - D_1^2}\right) /(\pi r^2)$$
$$A_2 = \left(2 \cdot r^2 * \cos^{-1}(D_2/(2r)) - (D_2/2) \cdot \sqrt{4 \cdot r^2 - D_2^2}\right) /(\pi r^2)$$
$$A_3 = \left(2 \cdot r^2 * \cos^{-1}(D_3/(2r)) - (D_3/2) \cdot \sqrt{4 \cdot r^2 - D_3^2}\right) /(\pi r^2)$$

Then we normalize the area's such that $\sum_0^3 A_i = 1$ These four values are then used as the weights in lieu of the $d_{x,y}$ pairs as follows

$$A_0 \approx (1 - d_x)(1 - d_y) \qquad A_1 \approx (1 - d_x)d_y \qquad A_2 \approx d_x(1 - d_y) \qquad A_3 \approx d_x d_y$$

Chapter 8

Dynamic Aperture Sampling

Dynamic aperture sampling is used in the refocusing algorithm to avoid banding due to under-sampling the s, t plane. This effect is caused when the separation between two adjacent projections (u, s') and $(u + \Delta u, s')$, Δs is greater than the original sampling of one microlens pitch. To illustrate this effect and provide the procedure used to overcome it, the following illustration and subsequent discussion is given.

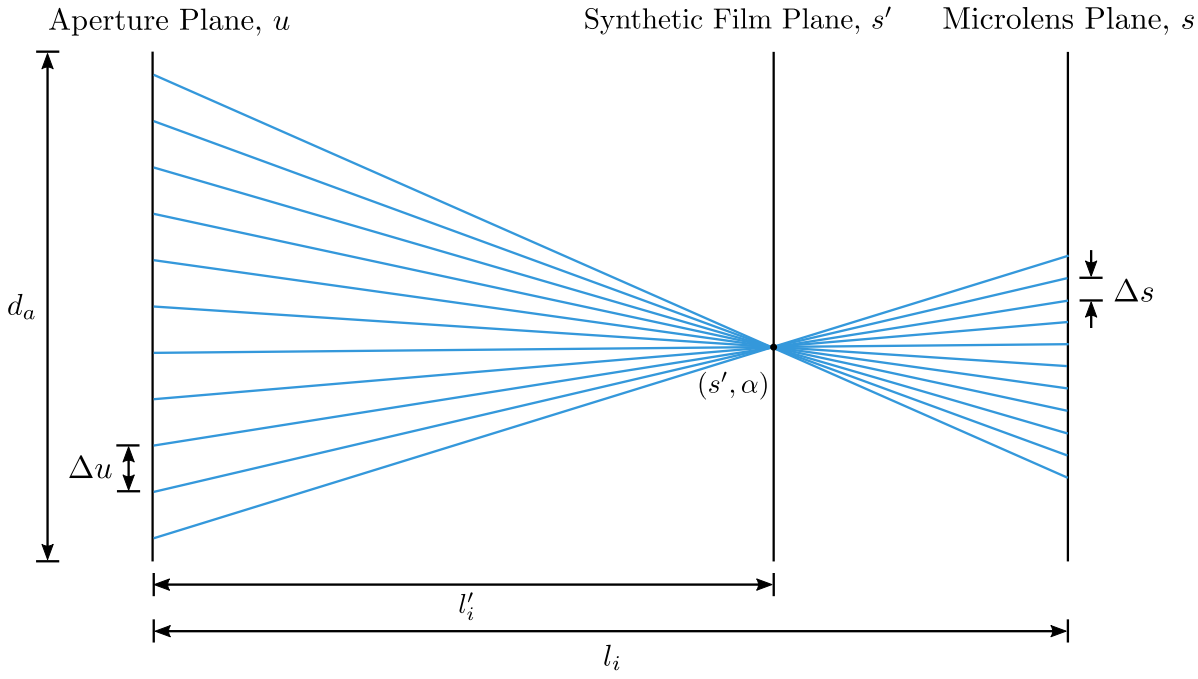


Figure 8.1: Dynamic aperture sampling.

The size of a pixel on the main lens can be calculated as

$$\Delta u = \frac{l_i p_p}{f_\mu}$$

using similar triangles. Then using the projection given by equation 3.13, the difference between two adjacent projections ($s_1 = f(u, s')$ & $s_2 = f(u + \Delta u, s')$) can be calculated as

$$\begin{aligned}
|s_2 - s_1| &= \left|(u + \Delta u)\left(1 - \frac{1}{\alpha}\right) + \frac{s'}{\alpha} - \left(u\left(1 - \frac{1}{\alpha}\right) + \frac{s'}{\alpha}\right)\right| \\
&= \left|(u + \Delta u)\left(1 - \frac{1}{\alpha}\right) - u\left(1 - \frac{1}{\alpha}\right)\right| \\
&= \left|\Delta u\left(1 - \frac{1}{\alpha}\right)\right| \\
\Delta s &= \Delta u \frac{|\alpha - 1|}{\alpha}
\end{aligned}$$

Using this separation we can determine if it is larger than a microlens (i.e. will it skip a microlens and cause banding.) If this is the case we recalculate Δu to avoid this.

$$\Delta u = \begin{cases} \Delta u & \text{if } \Delta s < p_\mu \\ p_\mu \frac{\alpha}{|\alpha - 1|} & \text{otherwise} \end{cases}$$

Once the separation has been determined the number of samples in u can be determined

$$n_u = \lceil d_a / \Delta u + 1 \rceil$$

Then the separation between each point is given by

$$\delta_u = \frac{u_f - u_i}{n_u - 1}$$

The remainder of the algorithm is presented below is pseudocode.

for $i = 0; i < n_u$ **do**
    $u' = u_i + \delta_u \cdot i$
    if $u' > u_i$ **then**
        $u' = u_i$

else if $u' < u_f$ **then**

$u' = u_f$

end if

$n_v = \lceil 2\sqrt{u_i^2 - u'^2}/\Delta u + 1 \rceil$

$d_{1/2} = \sqrt{(d_a/2)^2 - u'^2}$

for $j = 0; j < n_v$ **do**

if $n_v == 1$ **then**

$v' = 0$

else

$v' = d_{1/2} - 2 \cdot j \cdot d_{1/2}/(n_v - 1)$

end if

end for

end for

Chapter 9

Lens Simulation Derivation

This derivation provides the framework for the handling of the lenses in the plenoptic simulation tool. In contrast to using ray-transfer matrices, this implementation directly enforces the thin-lens equation. It was discovered that the small angle approximations used in the ray-transfer matrices created non-negligible errors in the propagation through a lens. Figure 9.1 shows the imaging process of a lens.

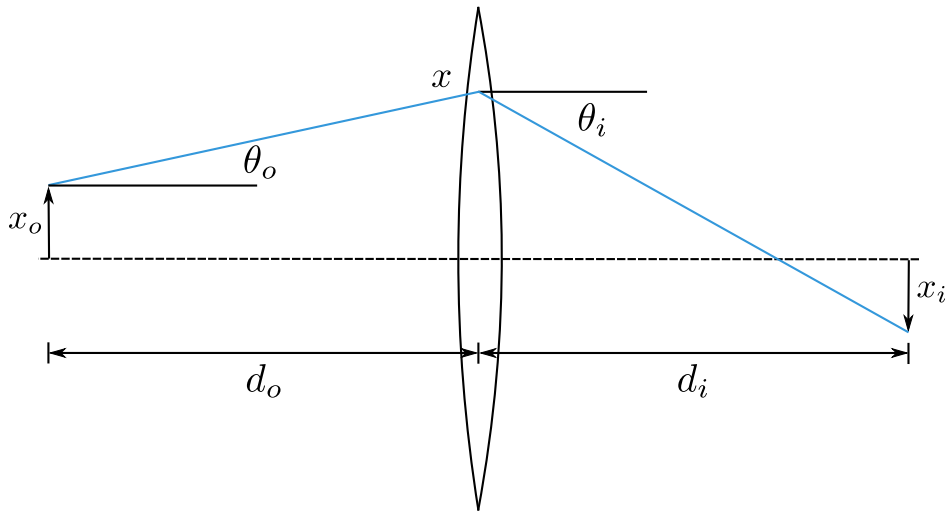


Figure 9.1: Schematic of a lens

First we define the image distance as

$$d_i = \left(\frac{1}{f} - \frac{1}{d_o}\right)$$

where f is the focal length of the lens. Then from the definition of magnification

$$M = -\frac{d_i}{d_o}$$

$$x_i = x_o \cdot M$$

The purpose of this lens function is to determine the angle of propagation leaving the lens. This angle is defined as

$$\theta_i = \tan^{-1}\left(\frac{x_i - x}{d_i}\right)$$

If we substitute for the known variables

$$\theta_i = \tan^{-1}\left(\frac{x_o\dot{M} - x}{d_i}\right)$$

$$\theta_i = \tan^{-1}\left(\frac{-x_o\frac{d_i}{d_o} - x}{d_i}\right)$$

$$\theta_i = \tan^{-1}\left(-\frac{x_o}{d_o} - \frac{x}{d_i}\right)$$

$$\theta_i = \tan^{-1}\left(-\frac{x_o}{d_o} - x\left(\frac{1}{f} - \frac{1}{d_o}\right)\right)$$

$$\theta_i = \tan^{-1}\left(\frac{x - x_o}{d_o} - \frac{x}{f}\right)$$

Chapter 10

Dragon Software Quick Reference

Dragon is the name of the software suite which implements the algorithms presented in this dissertation. This appendix describes the purpose of the functions as well as where they are discussed in this dissertation for reference. A more complete documentation has been left to the AFDL, which is expected to be continually updated, alongside the software itself. What is presented here is a snapshot of the code at the time of my graduation and represents my contribution.

10.1 External Functions

This set of functions represents the API boundary of the Dragon.dll library. All of these functions are exposed to the user to call at will from C++ as well as through the MATLAB API.

void DRG_initFromJson(**char*** settingsFileName)

Purpose:	To load in the settings file for the current process.
Relevant Section:	none.
Inputs:	**char*** settingsFileName Name of the *.json settings file.
Outputs:	1 if *.json was read successfully, 0 otherwise.
MATLAB Call:	DragonInitFromSettingsFile.m

void DRG_quit(**void**)

Purpose:	Deallocates the globalLightField, created by DRG_buildRadiance, variable if it exists.
Relevant Section:	none.
Inputs:	none.
Outputs:	none.
MATLAB Call:	DragonQuit.m

int DRG_simulatePlenopticCamera(...)

Purpose:	Simulates the imaging process of a plenoptic camera on a particle field.	
Relevant Section:	Section 3.4	
Inputs:	**int** nParticles	Number of particles to be simulated.
	float* xPos	x-position of the particles.
	float* yPos	y-position of the particles.
	float* zPos	z-position of the particles.
	float* intensity	Intensity of the particles.
	float* imageData	Output image data.
	char* calibrationDir	Directory to output microlens calibration file.
	float rotAngleX	Camera's angle of rotation about the x-axis.
	float rotAngleY	Camera's angle of rotation about the y-axis.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonSimulateImage.m	

int DRG_buildRadiance(...)

Purpose:	Builds the global light field array from raw image and microlens calibration data.	
Relevant Section:	Section 3.2	
Inputs:	**int** nMicroX	Number of microlenses in microlens calibration along x-axis.
	int nMicroY	Number of microlenses in microlens calibration along y-axis.
	float* cLocX	Microlens locations (via calibration) in x-dir.
	float* cLocY	Microlens locations (via calibration) in y-dir.
	float* imageData	Input raw image data.
	int nImages	Number of raw images.
	float* threshImage	Image used for the intensity threshold value in filtered refocusing.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonBuildRadiance.m	

int DRG_buildRadianceFromBOSVectors(...)

Purpose:	Builds the global light field array for Background Oriented Schlieren via the displacement vectors.	
Relevant Section:	See Klemkowsky *et al.* [74]	
Inputs:	**char*** vectorDir	Directory containing the processed vector fields.
	float* xVec	Vector defining s positions of radiance array, whose length is defined by the nPixelsX setting.
	float* yVec	Vector defining t positions of radiance array, whose length is defined by the nPixelsY setting.
	float* uVec	Vector defining u positions of radiance array.
	int numU	Number of u samples.
	float* vVec	Vector defining v positions of radiance array.
	int numV	Number of v samples.
	int bosRadianceType	0 - u-component, 1 - v-component, and 2 - magnitude.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonBuildRadianceFromBOSVectors.m	

int DRG_refocusLightFieldToImage(...)

Purpose:	Refocus global light field to an image with alpha scaling.	
Relevant Section:	Section 3.3.1	
Inputs:	**float** alpha	Normalized depth location.
	int nVoxelsX	Number of output pixels x-dir.
	int nVoxelsY	Number of output pixels y-dir.
	float* refImage	Output refocused image.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonRefocusToImage.m	

int DRG_refocusLightFieldToFocalStack(...)

Purpose:	Refocus global light field to a series of images with alpha scaling.	
Relevant Section:	Section 3.3.1	
Inputs:	**float*** alphaVector	Normalized depth locations of images in focal stack.
	int nAlphas	Number of images in focal stack.
	int nVoxelsX	Number of output pixels x-dir.
	int nVoxelsY	Number of output pixels y-dir.
	float* focalStack	Output refocused images.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonRefocusToFocalStack.m	

int DRG_generateSinglePerspectiveView(...)

Purpose:	Creates a perspective view at a given u, v value.	
Relevant Section:	Section 3.3.2	
Inputs:	**float**∗ xPrimeVector	x-locations of output pixels.
	int nVoxelsX	Number of output pixels x-dir.
	float∗ yPrimeVector	y-locations of output pixels.
	int nVoxelsY	Number of output pixels x-dir.
	float uPrime	u-position.
	float vPrime	v-position.
	float∗ perImage	Output perspective image.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonGenerateSinglePerspectiveView.m	

int DRG_generatePerspectiveViews(...)

Purpose:	Creates a sweep of perspective view at given u, v values.	
Relevant Section:	Section 3.3.2	
Inputs:	**float**∗ xPrimeVector	x-locations of output pixels.
	int nVoxelsX	Number of output pixels x-dir.
	float∗ yPrimeVector	y-locations of output pixels.
	int nVoxelsY	Number of output pixels y-dir.
	float∗ uPrimeVector	u-positions.
	float∗ vPrimeVector	v-positions.
	int nVoxelsY	Number of perspective views to output.
	float∗ perImageSweep	Output perspective images.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonGeneratePerspectiveViews.m	

int DRG_reconstructLightFieldToImage(...)

Purpose:	Creates a reconstructed image at a given z location.	
Relevant Section:	none.	
Inputs:	**float** zPosition	Depth location of image.
	float∗ xVector	x-locations of output pixels.
	int nVoxelsX	Number of output pixels x-dir.
	float∗ yVector	y-locations of output pixels.
	int nVoxelsY	Number of output pixels y-dir.
	float∗ xCoeff	x-coefficients for light field calibration.
	float∗ yCoeff	y-coefficients for light field calibration.
	float∗ filterThreshold	Image used for the validation threshold value in filtered refocusing.
	float∗ refImage	Output reconstructed image.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonReconstructToImage.m	

int DRG_lightFieldReconstruction(...)

Purpose:	Reconstructs raw images into volumes.	
Relevant Section:	Chapter 4	
Inputs:	**float*** xVector	x-locations of output voxels.
	int nVoxelsX	Number of output voxels x-dir.
	float* yVector	y-locations of output voxels.
	int nVoxelsY	Number of output voxels y-dir.
	float* zVector	z-locations of output voxels.
	int nVoxelsZ	Number of output voxels z-dir.
	float* camAngles	Angles of cameras. Only used for synthetic data.
	int nCameras	Number of cameras.
	int* imageNumbers	Image numbers to be reconstructed. Its assumed that the images have the 0000.tif naming convention.
	int nImagesPerCamera	Number of images to be reconstructed per camera.
	char* image_folder	Folder containing raw images. Images must be in subfolders named based on their camera index (i.e. /camera0/)
	char* mcal_folder	Folder containing microlens calibration file. Must be named after camera number (i.e. camera0.drg-mcal)
	char* lfcal_folder	Folder containing light field calibration file. Must be named after camera number (i.e. camera0.drg-lfcal)
	char* iThresh_folder	Folder containing intensity threshold for filtered refocusing. Must be named after camera number (i.e. camera0.tif)
	char* fThresh_folder	Folder containing validation threshold for filtered refocusing. Must be named filter0.bin and filter1.bin for first and second frame.
	char* vol_folder	Volume output folder. Volume names will be the same as the image names.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonLightfieldReconstruction.m	

int DRG_runCrossCorrelation2D(...)

Purpose:	Calculates the 2D vector field from a PIV image pair	
Relevant Section:	Chapter 2	
Inputs:	**float**∗ imageA	Image data from frame A.
	float∗ imageB	Image data from frame B.
	float∗ imageMask	Image Mask.
	int nPasses	Number of passes in the WIDIM scheme.
	int∗ windowSizesX	Size of the windows in x-direction.
	int∗ windowSizesY	Size of the windows in y-direction.
	int∗ windowOffsetsX	Offsets between windows in x-direction.
	int∗ windowOffsetsY	Offsets between windows in y-direction.
	char∗ outputFileName	Vector output file.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonRunCrossCorrelation2D.m	

int DRG_runCrossCorrelation3D(...)

Purpose:	Calculates the 3D vector field from a PIV volume pair	
Relevant Section:	Chapter 2	
Inputs:	**float**∗ volumeA	Volume data from frame A.
	float∗ volumeB	Volume data from frame B.
	float∗ volumeMask	Volume Mask.
	int nPasses	Number of passes in the VOLDIM scheme.
	int∗ windowSizesX	Size of the windows in x-direction.
	int∗ windowSizesY	Size of the windows in y-direction.
	int∗ windowSizesZ	Size of the windows in z-direction.
	int∗ windowOffsetsX	Offsets between windows in x-direction.
	int∗ windowOffsetsY	Offsets between windows in y-direction.
	int∗ windowOffsetsZ	Offsets between windows in z-direction.
	char∗ outputFileName	Vector output file.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonRunCrossCorrelation3D.m	

int DRG_runCorrelationAveraging3D(...)

Purpose:	Calculates the 3D vector field using correlation averaging from a series of PIV volume pairs	
Relevant Section:	none.	
Inputs:	**char*** volumeFolderName	Folder containing volumes with frames in sequential order.
	int* imageNumbers	Volume numbers to be processed.
	int nImagePairs	Number of volume pairs to be processed.
	float* volMask	Volume mask.
	int nPasses	Number of passes in the VOLDIM scheme.
	int* windowSizesX	Size of the windows in x-direction.
	int* windowSizesY	Size of the windows in y-direction.
	int* windowSizesZ	Size of the windows in z-direction.
	int* windowOffsetsX	Offsets between windows in x-direction.
	int* windowOffsetsY	Offsets between windows in y-direction.
	int* windowOffsetsZ	Offsets between windows in z-direction.
	char* outputFileName	Vector output file.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	DragonRunEnsembleCrossCorrelation3D.m	

int DRG_calibrateSyntheticCamera(...)

Purpose:	Calculates the light field calibration for a synthetic camera	
Relevant Section:	Section 4.1	
Inputs:	**int** nPoints	Number of points used in Monte-Carlo simulation.
	float* xPos	x-location of points used in simulation.
	float* yPos	y-location of points used in simulation.
	float* zPos	z-location of points used in simulation.
	float rotAngleX	Camera's angle of rotation about the x-axis.
	float rotAngleY	Camera's angle of rotation about the y-axis.
	char* outputFileName	Calibration output file.
Outputs:	Error code: 1 if successful.	
MATLAB Call:	none.	

10.2 Internal Functions

These functions are helper functions and not accessible outside the library. They are listed by the file they are in.

10.2.1 Dragon.cpp

Main compilation file, contains mostly basic math type functions as well as functions used throughout the library.

isIndeterminate(x):	Determines if the variable is a real number.
isInfinite(x):	Determines if the variable is infinite.
idx2(dims,d0,d1):	Converts 2D array index to linear index.
idx3(dims,d0,d1,d2):	Converts 3D array index to linear index.
idx4(dims,d0,d1,d2,d3):	Converts 4D array index to linear index.
idx5(dims,d0,d1,d2,d3,d4):	Converts 5D array index to linear index.
nnInterp(A,dimX,xi,yi):	2D nearest neighbor interpolation.
bilerp(A,dimX,xi,yi):	2D linear interpolation.
bicubic(A,dimX,xi,yi):	2D cubic interpolation.
clamp(val,min,max):	Clamps val between min and max.
randInRange(min,max):	Returns random value between min and max.
min(x,n):	Returns minimum value of x.
max(x,n):	Returns maximum value of x.
mean(x,n):	Returns mean of x.
median(x,n):	Returns median of x.
dot(x,y,n):	Returns dot product of x and y.
slidingMinFilter(...):	Filter which subtracts minimum of window from image.
slidingMeanFilter(...):	Filter which subtracts mean of window from image.
slidingNormalization(...):	Filter which normalizes image with mean of window.
turtle(nTotalSamplesUB,x,y):	Turtle marching function.
imageDistanceConversion(zPos):	Calculates l_o, l_i, M', and α.
apertureFunction(...):	Dynamic aperture function. (See Appendix 8).
applyLightFieldCalibration(...):	Converts (x, y, z, u, v) to (s, t) (Section 4.1).
gaussianSmoothingFilter(...):	Smooths volume with a Gaussian function.

10.2.2 DragonSIM.cpp

File containing code related to the plenoptic simulation tool.

translate(ray,distance):	Function used to translate the light ray (Section 3.4).
lens(...):	Function used to propagate the light ray through a lens (Section 3.4).

10.2.3 DragonLFT.cpp

File containing code related to manipulation of the light field including: refocusing and perspective view generation.

getNearestMicrolensIndex(...):	Finds the nearest microlens index to point (x2,y2).
lightFieldInterpLinearInternal(...):	Returns the 4 microlens coordinates to (x2,y2) and their interpolation coefficients. (Section 3.3).
lightFieldInterpLinear(...):	Calculates the radiance of the light field at (x2,y2,u2,v2) via linear interpolation.
lightFieldInterpSINC(...):	Calculates the radiance of the light field at (x2,y2,u2,v2) via cardinal interpolation.
lightFieldInterpIntensityThreshold(...):	Calculates the intensity threshold of the light field at (x2,y2,u2,v2) via linear interpolation.
refocusLightFieldAtPoint(...):	Calculates the refocused value at a point (xPrime, yPrime, alpha).
perspectiveShiftAtPoint(...):	Calculates the perspective value at a point (xPrime, yPrime, uPrime, vPrime).
buildRadiance(...):	builds the radiance of the input lightfield.

10.2.4 DragonRCN.cpp

File containing code related to volumetric reconstructions.

monteCarloDLFC(...):	Calculates the light field calibration for synthetic data.
perspectiveShiftAtPointIntThresh(...):	Calculates the perspective view of the intensity threshold for filtered refocusing.

10.2.5 DragonPIV.cpp and DragonPIV3D.cpp

File containing code related to 2D and 3D cross-correlation.

prepareImage(...):	Masks and buffers raw image.
generateGrid(...):	Calculates locations of vectors for current pass.
generatePredictor(...):	Calculates velocity predictor.
interpolateDeformation(...):	interpolation functions for image/volume deformation (Section 2.4.3).
imageDeformation(...):	Deforms images (Section 2.4.3).
performCrossCorrelation(...):	Calculates cross-correlation for entire image/volume (Section 2.4.1).
findCorrelationPeaks(...):	Calculates the correlation peak for each window (Section 2.4.1).
vectorValidation(...):	Validates vector field based on normalized median test (Section 2.4.2).
vectorReplacement(...):	Replaces invalid vectors using interpolation (Section 2.4.2).
regressionFiltering(...):	Filters velocity data with linear regression (Section 2.4.3).
exportVectorFieldToFile(...):	Exports vector data in Tecplot format.

Bibliography

[1] Fulvio Scarano. Iterative image deformation methods in PIV. *Measurement Science and Technology*, 13:R1–R19, 2001.

[2] M. Stanislas, K. Okamoto, C. J. Kähler, and J. Westerweel. Main results of the Second International PIV Challenge. *Experiments in Fluids*, 39(2):170–191, 2005.

[3] Christian Willert. Stereoscopic digital particle image velocimetry for application in wind tunnel flows. *Measurement Science and Technology*, 8:1465–1479, 1999.

[4] Fulvio Scarano. Tomographic PIV: principles and practice. *Measurement Science and Technology*, 24:012001, 2013.

[5] Marc Levoy. Light fields and computational imaging. *Computer*, 39:46–55, 2006.

[6] Ren Ng, Marc Levoy, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light Field Photography with a Hand-held Plenoptic Camera. *Informational*, pages 1–11, 2005.

[7] Ronald J. Adrian and Chung Sheng Yao. Development of pulsed laser velocimetry (PLV) for measurement of turbulent flow. In *Proc Symp Turbul*. Univ of Missouri-Rolla, 1984.

[8] Cameron Tropea, A.L. Yarin, and J.F. Foss. Springer Handbook of Experimental Fluid Mechanics. *Young*, 1:959–1042, 2007.

[9] Hiroshi Komine, Stephen J Brosnan, Allen B Litton, and Eddy A Stappaerts. Real-Time, Doppler Global Velocimetry. In *Conference on Lasers and Electro-Optics*, page CPD20. Optical Society of America, 1990.

[10] B Hiller, R A Booman, C Hassa, and R K Hanson. Velocity visualization in gas flows using laserinduced phosphorescence of biacetyl. *Review of Scientific Instruments*, 55(12):1964–1967, 1984.

[11] James B Michael, Matthew R Edwards, Arthur Dogariu, and Richard B Miles. Femtosecond laser electronic excitation tagging for quantitative velocity imaging in air. *Appl. Opt.*, 50(26):5158–5162, sep 2011.

[12] R Meynart. Instantaneous velocity field measurements in unsteady gas flow by speckle velocimetry. *Applied optics*, 22(4):535–540, 1983.

[13] Ronald J Adrian. Scattering particle characteristics and their effect on pulsed laser measurements of fluid flow: speckle velocimetry vs particle image velocimetry. *Applied optics*, 23(11):1690–1691, 1984.

[14] Ronald J. Adrian and Jerry Westerweel. *Particle Image Velocimetry*. Cambridge University Press, 2011.

[15] Markus Raffel, CE Willert, Steven T Wereley, and Jürgen Kompenhans. *Particle image velocimetry: a practical guide*. Springer Berlin, 2007.

[16] Clayton Crowe, John Schwarzkopf, Martin Sommerfeld, and Yutaka Tsuji. *Multiphase Flowes with Droplets and Particles*. CRC Press, 2012.

[17] R. Mei. Velocity fidelity of flow tracer particles. *Experiments in Fluids*, 22(1):1–13, 1996.

[18] a Melling. Tracer particles and seeding for particle image velocimetry. *Measurement Science and Technology*, 8(12):1406–1416, 1997.

[19] F. Scarano and M. L. Riethmuller. Iterative multigrid approach in PIV image processing with discrete window offset. *Experiments in Fluids*, 26:513–523, 1999.

[20] Fulvio Scarano and Christian Poelma. Three-dimensional vorticity patterns of cylinder wakes. *Experiments in Fluids*, 47:69–83, 2009.

[21] C. E. Willert and M. Gharib. Digital particle image velocimetry. *Experiments in Fluids*, 10(4):181–193, 1991.

[22] F. Scarano and M. L. Riethmuller. Advances in iterative multigrid PIV image processing. *Experiments in Fluids*, 29(7):S051–S060, 2000.

[23] Richard D. Keane and Ronald J. Adrian. Theory of cross-correlation analysis of PIV images. *Applied Scientific Research*, 49(3):191–215, 1992.

[24] Jerry Westerweel and Fulvio Scarano. Universal outlier detection for PIV data. *Experiments in Fluids*, 39(6):1096–1100, 2005.

[25] J. Nogueira, a. Lecuona, and P. a. Rodríguez. Local field correction PIV: on the increase of accuracy of digital PIV systems. *Experiments in Fluids*, 27(2):107–116, 1999.

[26] T. Astarita. Analysis of velocity interpolation schemes for image deformation methods in PIV. *Experiments in Fluids*, 45(2):257–266, 2008.

[27] F. F J Schrijer and Fulvio Scarano. Effect of predictor-corrector filtering on the stability and spatial resolution of iterative PIV interrogation. *Experiments in Fluids*, 45:927–941, 2008.

[28] M.P. Arroyo and C.A. Greated. Stereoscopic particle image velocimetry. *Measurement Science and Technology*, 2:1181–1186, 1991.

[29] a. K. Prasad. Stereoscopic particle image velocimetry. *Experiments in Fluids*, 29(2):103–116, 2000.

[30] S M Soloff, R J Adrian, and Z-C Liu. Distortion compensation for generalized stereoscopic particle image velocimetry. *Measurement Science and Technology*, 8(12):1441–1454, 1997.

[31] Sebastien J M Coudert and Jean-Paul Schon. Back-projection algorithm with misalignment corrections for 2D3C stereoscopic PIV. *Measurement Science and Technology*, 12(9):1371–1381, 2001.

[32] B. Wieneke. Stereo-PIV using self-calibration on particle images. *Experiments in Fluids*, 39(2):267–280, 2005.

[33] C.E. Willert and M. Gharib. Three-dimensional particle imaging with a single camera. *Experiments in Fluids*, 12(6), apr 1992.

[34] C. J. Kahler and J. Kompenhans. Fundamentals of multiple plane stereo particle image velocimetry. *Experiments in Fluids*, 29:S070–S077, 2000.

[35] Ch. Brucker. Digital-Particle-Image-Velocimetry (DPIV) in a scanning light-sheet: 3D starting flow around a short cylinder. *Experiments in Fluids*, 19(4), aug 1995.

[36] F. Pereira, M. Gharib, D. Dabiri, and D. Modarress. Defocusing digital particle image velocimetry: a 3-component 3-dimensional DPIV measurement technique. Application to bubbly flows. *Experiments in Fluids*, 29:S078–S084, 2000.

[37] K D Hinsch. Holographic particle image velocimetry. *Measurement Science & Technology*, 13:R61–R72, 2002.

[38] Sven F Herrmann and Klaus D Hinsch. Light-in-flight holographic PIV (LiFH-PIV) for wind- tunnel applications : Off-site reconstruction of deep- volume real particle images. *Measurement Science & Technology*, 15:1–9, 2004.

[39] G. E. Elsinga, Fulvio Scarano, B. Wieneke, B. W. Oudheusden, and B. W. Van Oudheusden. Tomographic particle image velocimetry. *Experiments in Fluids*, 41:933–947, 2006.

[40] Jesse Belden, Tadd T Truscott, Michael C Axiak, and Alexandra H Techet. Three-dimensional synthetic aperture particle image velocimetry. *Measurement Science and Technology*, 21:125403, 2010.

[41] Kyle P. Lynch and Brian S. Thurow. 3-D flow visualization of axisymmetric jets at Reynolds number 6,700 and 10,200. *Journal of Visualization*, 15:309–319, 2012.

[42] Brian Thurow, Kyle Lynch, Steven Williams, and Micahel Melnick. 3-D flow imaging using a MHz rate puls burst Laser system. In *15th Int Symp on Applications of Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, 2010.

[43] Jian Sheng, Edwin Malkiel, and Joseph Katz. Digital holographic microscope for measuring three-dimensional particle distributions and motions. *Applied optics*, 45(16):3893–3901, 2006.

[44] J. Zhang, B. Tao, and J. Katz. Turbulent flow measurement in a square duct with hybrid holographic PIV. *Experiments in Fluids*, 23(5):373–381, 1997.

[45] Matteo Novara, Kees Joost Batenburg, and Fulvio Scarano. Motion tracking-enhanced MART for tomographic PIV. *Measurement Science and Technology*, 21(3):035401, 2010.

[46] K. P. Lynch and F. Scarano. An efficient and accurate approach to MTE-MART for time-resolved tomographic PIV. *Experiments in Fluids*, 56(3):1–16, 2015.

[47] Daniel Schanz, Sebastian Gesemann, and Andreas Schröder. Shake-The-Box: Lagrangian particle tracking at high particle image densities. *Experiments in Fluids*, 57(5):1–27, 2016.

[48] Edward H. Adelson and Jr Bergen. The plenoptic function and the elements of early vision. *Computational models of visual ...*, pages 3–20, 1991.

[49] Edward H. Adelson and John Y a Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):99–106, 1992.

[50] Marc Levoy and Pat Hanrahan. Light field rendering. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, pages 31–42, 1996.

[51] Andrew Lumsdaine and Todor Georgiev. The Focused Pleoptic Camera, ICCP. In *IEEE International Conference on.*, 2009.

[52] G Lippman. La photographie intégrale. *Adadémie des Sciences*, 146:446–551, 1908.

[53] Ren Ng. Digital light field photography. pages 1–203, 2006.

[54] Ren Ng. Fourier slice photography. *ACM Transactions on Graphics*, 24:735, 2005.

[55] Marc Levoy, Ren Ng, Andrew Adams, Matthew Footer, and Mark Horowitz. Light field microscopy. *ACM Transactions on Graphics*, 25(3):924, 2006.

[56] M. Levoy, Z. Zhang, and I. McDowall. Recording and controlling the 4 D light field in a microscope using microlens arrays. *Journal of Microscopy*, 235(November 2008):144–162, 2009.

[57] Kyle Lynch. *Development of a 3-D Fluid Velocimetry Technique based on Light Field Imaging*. PhD thesis, Auburn University, 2011.

[58] Timothy W Fahringer, Kyle P Lynch, and Brian S Thurow. Volumetric particle image velocimetry with a single plenoptic camera. *Measurement Science and Technology*, 26(11):115201, 2015.

[59] Christoph Skupsch and Christoph Brücker. Multiple-plane particle image velocimetry using a light-field camera. *Optics Express*, 21(2):1726–40, 2013.

[60] Stefan Ostmann, Humberto Chaves, and Christoph Brücker. 3D path tracking of particles in a swirling flow using a light-field camera. *17th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, pages 7–10, 2014.

[61] R.R. La Foy and P. Vlachos. Multi-Camera Plenoptic Particle Image Velocimetry. *PIV13; 10th International Symposium on Particle Image Velocimetry, Delft, The Netherlands, July 1-3, 2013*, jul 2013.

[62] Antonio Cenedese, Claudia Cenedese, Francesco Furia, Marco Marchetti, Monica Moroni, and Luca Shindler. 3D particle reconstruction using light field imaging. *16th Int Symp on Applications of Laser Techniques to Fluid Mechanics*, pages 9–12, 2012.

[63] T Nonn, J Kitzhofer, D Hess, and Christoph Brücker. Measurements in an IC-engine flow using light-field volumetric velocimetry. *16th International Symposium . . .*, i:9–12, 2012.

[64] Christoph S. Garbe, Björn Voss, and Julian Stapf. Plenoptic Particle Streak Velocimetry (pPSV): 3D3C fluid flow measurement from light fields with a single plenoptic camera. *16th Int Symp on Applications of Laser Techniques to Fluid Mechanics*, (1):9–12, 2012.

[65] Tatsuya Kawaguchi, Takushi Saito, and Isao Satoh. Volumetric reconstruction of particulate dispersions from a light field image by means of a plenoptic camera. *17th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, i:7–10, 2014.

[66] Elise M. Hall, Timothy W. Fahringer, Brian S. Thurow, and Daniel R. Guildenbecher. Volumetric calibration of a plenoptic camera. In *55th AIAA Aerospace Sciences Meeting*, Grapevine, Texas, jan 2017. American Institute of Aeronautics and Astronautics.

[67] Callum Atkinson and Julio Soria. An efficient simultaneous reconstruction technique for tomographic particle image velocimetry. *Experiments in Fluids*, 47:553–568, 2009.

[68] G T Herman and A Lent. Iterative reconstruction algorithms. *Computers in biology and medicine*, 6(4):273–94, oct 1976.

[69] Timothy W Fahringer and Brian S Thurow. Filtered refocusing: a volumetric reconstruction algorithm for plenoptic-PIV. *Measurement Science and Technology*, 27(9):94005, 2016.

[70] H G Maas, a Gruen, and D Papantoniou. Particle tracking velocimetry in 3-dimensional flows. 1. Photogrammetric determination of partacle coordinates. *Experiments in Fluids*, 15(2):133–146, 1993.

[71] M Gharib, E Rambod, and K Shariff. A universal time scale for vortex ring formation. *J. Fluid Mech.*, 360:121–140, 1998.

[72] N. a. Worth and T. B. Nickels. Acceleration of Tomo-PIV by estimating the initial volume intensity distribution. *Experiments in Fluids*, 45:847–856, 2008.

[73] Stefano Discetti and Tommaso Astarita. A fast multi-resolution approach to tomographic PIV. *Experiments in Fluids*, 52(3):765–777, 2012.

[74] Jenna Klemkowsky. *Plenoptic BOS: Combining a Plenoptic Camera with the Background Oriented Schlieren Technique*. PhD thesis, Auburn University, 2016.

[75] Eric A Deem, Louis N Cattafesta, Timothy W Fahringer, and Brian S Thurow. On the resolution of Plenoptic PIV. *11th International Symposium on Particle Image Velocimetry*, 084003:84003, 2015.

[76] R Kingslake. Depth of Field. In *Optics in Photography*, pages 84–88. 1992.

[77] Christian Perwass and Lennart Wietzke. Single lens 3D-camera with extended depth-of-field. *Proc.SPIE*, 8291:8921–8921–15, 2012.

[78] Matteo Novara and Fulvio Scarano. Performances of motion tracking enhanced Tomo-PIV on turbulent shear flows. *Experiments in Fluids*, 52(4):1027–1041, 2012.

[79] Callum Atkinson, Sebastien Coudert, Jean-Marc Foucaut, Michel Stanislas, and Julio Soria. The accuracy of tomographic particle image velocimetry for measurements of a turbulent boundary layer. *Experiments in Fluids*, 50:1031–1056, 2011.