On the Development of Defocusing Digital Particle Image Velocimetry with Full Characterization

Thesis by

Emilio Castaño Graff

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

2007
(Defended May 25, 2007)

The contents presented herein are proprietary and are not to be viewed, copied, or distributed without explicit and exclusive written consent from Emilio Castaño Graff from the date of submission until 16 January 2018.

Acknowledgements

This research was supported by the Office of Naval Research, under grants ONR-N00014-00-1-0110, ONR-N0014-07-0138, and ONR-MURI-N00015-06-1-0730.

Special thanks go to Francisco Pereira, Pavel Svitek, and Mory Gharib.

Abstract

Defocusing Digital Particle Image Velocimetry is the first volumetric, three-dimensional PIV method ever put into practice. This manuscript contains the details of its development, a detailed analysis of its performance (both through simulation and real measurements), and a series of experimental demonstrations of the capability of the technique. The system is capable of resolving upwards of 7,000 vectors per pair with an absolute error on the order of 0.03% of the volume size.

MATRIX

Contents

List of Figures

List of Tables

List of Terms

mDnC — Refers to the dimensionality of the measurement in a particular PIV technique. m is the dimensions of the volume of measurement (2 for light sheets and 3 for volumes) and n is the number of velocity components extracted. Naturally, $m \leq n$.

X, Y, Z — Coordinates denoted by upper-case letters are space-coordinates—that is, coordinates in the real world (mm). These are also used to designate the coordinate system. For a defocusing camera, the origin is at the intersection of the optical axis and the aperture plane. Z goes along the optical axis, Y is “height”, and X is “width”.

s — The side length of an individual pixel.

x, y — Coordinates denoted by lower-case letters are pixel coordinates and usually refer only to the position of an object on the sensor or in an image. In the calculations, image coordinates are usually in millimeters, which can be converted to pixels knowing the pixel size of the sensor in question.

L — The distance between the aperture plane and the reference plane of a defocusing camera (mm).

l — The distance between the aperture plane and the image plane in the pinhole-optics model of a defocusing camera (mm).

f — The focal length of a lens (mm)—not to be confused with the f-number, denoted as f/.

c, d — The coordinates of an aperture on the faceplate in the three-dimensional pinhole-optics model of a defocusing camera (mm).

s_{ij} — The aperture separation in the three-dimensional pinhole model.

a — The [short] side length of the characteristic probe volume.

B
The sensitivity coefficient—the camera design parameters' contribution to the sensitivity of a camera. $B = Ms_{ij}$.

R
The planar resolution at the reference plane in pixels per millimeter. $R = M/s$ where s is the dimension of one pixel.

aperture plane
In the pinhole-optics model of the defocusing camera, this is the plane on which all the apertures lie. Here $Z = 0$.

camera
In this document, *camera* refers to a defocusing camera assembly: that is, three sensors with three lenses. (*Example*: the Ian *Camera*)

characteristic probe volume
This is the rectangular prism whose shortest cross-section is the largest square inscribable into the cross-section of the mappable region.

CL160
Framegrabbers from IO Industries which include an on-board SCSI RAID-0 interface to bypass the PCI bus when recording images from digital cameras directly to disk.

DDPIV
Defocusing Digital Particle Image Velocimetry. Note that `DDPIV` refers to the processing software whereas DDPIV refers to the concept.

defocusing
The concept by which a multiple-aperture optical system can be used to measure position in space, introduced by Willert and Gharib [1992].

defocusing camera
A camera built employing the defocusing concept. Also referred to as *DDPIV camera* or *3D camera*.

dewarping
The multi-plane scheme for software correction of optical aberration and assembly error used in DDPIV, whereby a grid of dots is imaged at several Z positions.

double frame
A double frame image contains the first frame of a pair on the top half of the image and the second frame on the bottom half.

double frame
A double frame image contains the first frame of a pair on the top half of the image and the second pair on the bottom frame.

focal plane
Equivalent to the reference plane, this is the plane at which the fields of view of all sensors in a defocusing camera coincide with each other.

ghost particle
A particle which is generated from a particle image group whose particle images did not all come from the same particle in space.

group

When referring to a group of particle images, a group is the collection of particle images which are necessary to reconstruct a particle's position. In three-aperture cameras, then, groups consist of three particle images and are sometimes referred to as *triplets*.

image plane

This is the plane on which the sensors of a defocusing camera lie.

mappable region

The region in a defocusing camera which is the intersection of the fields of view of all the sensors. The location of any point within this region can be measured (withholding any optical effects). The region in practice is usually smaller and cut short compared to the theoretical one due to depth of field, manufacturing error, deviation from pinhole optics, etc.

matching tolerance

See *pixel tolerance*.

optical axis

This is a line that traverses through the centerline of an optical system. In the case of a defocusing camera, the optical axis is perpendicular to the aperture plane and goes through the point of intersection of the sensor axes. The origin of the coordinate axes is defined as the intersection of the optical axis and the aperture plane.

optical footprint

The minimum size of the optical window of a tank placed at a given Z coordinate so that the mappable region is unobstructed.

outlier

A velocity vector which is clearly wrong when compared to its neighbors.

package

In this document, *package* refers to the actual ceramic package of the imager chip. (*Example*: Kodak KAI-2001)

particle

A light-scattering flow marker (such as a microscopic particle or bubble) or a dot pattern which is tracked by DDPIV. In terms of processing, *particle* refers to the three-dimensional representation of a particle image group.

particle image

The image of a particle or the light scattered off a particle on an imaging sensor.

particle image triplet

A triplet of particle images which corresponds to a particle in space. For a camera with more than three apertures, the correct term would be "particle image group".

pixel tolerance

The number of pixels deviation from the aperture layout pattern that a set of matching particle images is allowed to be; it is also called the *matching tolerance*. In `DDPIV` there are two pixel tolerances, a coarse tolerance for initial matching and a fine tolerance used in the final reconstruction. Frequently the pixel tolerance is simply referred to as the "fine tolerance" or the "coarse tolerance".

`Python`

This is a very powerful scripting language, available at http://www.python.org. Video Savant can be controlled via `Python` scripts, which in turn can run programs from the command line, making it possible to automate procedures such as multi-plane dewarping or systematic experiments.

quality

When used as "the quality of the reconstruction", this refers to the proportion of ghosts and real particles in a reconstructed point cloud; when used as "the quality of the vector field", it refers to the relative proportion of outlier vectors.

RAID

Redundant Array of Independent Disks. RAID exists in various modes. The CL160 framegrabbers from IO Industries employ RAID-0, commonly referred to as "striping", where a chunk of data that must be written is split into as many "stripes" as there are drives in the array and then written to all drives simultaneously. Therefore, not counting overhead, a RAID-0 array with n identical drives is both n times larger than one drive and n times faster than one drive.

real particle

A particle whose corresponding particle image group contains only particle images from one specific particle in space.

reference plane

Equivalent to the focal plane, this is the plane at which the fields of view of all sensors in a defocusing camera coincide with each other.

resolution

In terms of defocusing cameras, the resolution has two main components: the pinhole-optics sensitivity of the camera and planar resolution of the sensor.

SCSI

Small Computer System Interface. This is an interface for hard drives which is high-speed and reliable. The CL160 framegrabbers from IO Industries have an on-board SCSI RAID-0 interface for storing images during recording.

sensitivity
This refers to the derivative $\frac{db}{dZ}$, which expresses how much particle image separation varies with respect to the forming particle's Z coordinate for a given setup. Note that to be mathematically correct the sensitivity is a partial derivative but considering that the sensitivity coefficient is dependent entirely on camera design parameters once a camera is built the sensitivity is only a function of Z.

sensitivity coefficient
The coefficient B of the sensitivity $\frac{\partial b_{ij}}{\partial Z_P} = -\frac{BL}{Z_P^2}$ which indicates the contribution of the optical and geometric layout of two apertures to the sensitivity.

sensor
In this document, *sensor* does not refer to the sensor chip itself but rather the entire imager (camera). Thus each *camera* is built with three *sensors*. (*Example*: ImperX 2M30L)

sensor axis
The line connecting the center of the sensor to the center of its corresponding aperture in a defocusing camera. For proper alignment, the sensor axes of all the sensors in the camera should intersect at one point.

triplet
The three particle images the constitute a single particle in a three-aperture camera.

usable probe volume
This is a measurement of the actual Z coordinate range which can be mapped using a camera. The range is found by performing a dewarping calibration and then imaging a grid target at intervals in the range of Z's enclosed by the mappable region and processing them to see at what pixel tolerance they can be found.

Video Savant
Software from IO Industries used to capture images from digital cameras to disk.

voxel
This refers to the three-dimensional element of a three-dimensional grid—used in `FILTERPART` for the particle statistics and in `FINDFLOW` for cross-correlated flow calculation. The name from the analogy "pixel" is to "picture element" as "voxel" is to "volume element".

yield
The yield of a point cloud is roughly the number of particles divided by the average number of particle images, and is an indicator of the performance of the hardware, illumination, seeding, and calibration together. The yield of the vector field is the number of vectors divided

by the average number of particles, and depends also on the parameter Φ (when performing tracking).

Part I

Introduction

Chapter 1

Three-dimensional particle image velocimetry

1.1 Particle Image Velocimetry

1.2 Motivation

The development of DPIV in the early 90's was a revolution in that it allowed researchers to more easily quantify flow features that were previously only visualized with smoke, dye, or particle streaks. Its limitations to a two-dimensional planar domain spurred development of three-dimensional alternatives such as Stereo DPIV (SPIV) almost immediately as soon as the technology allowed for it. Fully volumetric, 3D3C techniques such as holography were also developed, but remained too difficult to implement practically.

Even with its relative simplicity SPIV did not become as common as standard DPIV. Attempts to produce some volumetric information doubled the cost of the system by using two SPIV setups in tandem imaging close planes of the flow (so that spatial derivatives could be calculated in all directions). The cost and complication of using four cameras with two lasers (often of different wavelength) prohibited Dual SPIV from progressing much further.

Researchers settled on using DPIV to take slices of the flow and inferring three dimensional behavior from these multiple planes (which varied in position, orientation, or both).

It is well known that the vorticity generated in a flow is very closely linked to the forces and thus momentum in the flow, especially in biomimetic-type propulsion[1]. The research mentioned in this literature, as well as others, all aims to understand the flow generated by dynamic mechanisms and correlate that with the forces, because only by understanding the flow structures can engineers learn to harness them for propulsion or other real uses.

[1] see Saffman [1992], Anderson, Streitlien, Barrett, and Triantafyllou [1998], Bartol, Gharib, Weihs, Webb, Hove, and Gordon [2003], Birch and Dickinson [2001, 2003], Birch, Dickson, and Dickinson [2004], Drucker and Lauder [1999], Wang, Birch, and Dickinson [2004], Dickinson [1999], among others

But without 3D3C data, many gaps exist in which the researchers have no alternative other than to speculate. Vorticity, for example, cannot even be fully calculated with even SPIV measurements. Although all three velocity components are available (U, V, and W), calculating full vorticity also requires a gradient in the velocities in each of the three directions, and SPIV cannot natively provide the $\partial/\partial z$ derivative.

$$\omega_x = \frac{\partial W}{\partial y} - \frac{\partial V}{\partial z} \qquad \omega_y = \frac{\partial W}{\partial x} - \frac{\partial U}{\partial z} \qquad \omega_z = \frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} \tag{1.2-1}$$

Although vortical structures can be inferred from multiple sets of 2D data, it is easy to think of how much detail is potentially lost: a corkscrew vortex will look the same as a bent tube unless so many slices are taken that the measurement approximates a volumetric one. Several real examples for the need of a comprehensive three-dimensional flow data exist. Drucker and Lauder [1999] infers the flow structure generated by a fish fin from 3 mutually orthogonal planes of DPIV data. Ringuette [2004] observed with dye how rapidly (almost instantly[2]) the flow around an impulsively started plate develops three-dimensionality . Johansson and Norberg [2003] proposed a method of locomotion for swimming birds from qualitative observations of dye around a mechanically-actuated rigid plate. In Noca, Shiels, and Jeon [1997], the discrepancy in amplitude between the force measurement and calculation is attributed partly to the three-dimensionality of the flow.

Birch and Dickinson [2001] observed that there is a substantial span-wise flow (which may aid in the prolonged attachment of the leading-edge vortex) on a model of a fruit fly wing from perpendicular DPIV planes . They predict that the spanwise flow drains energy from the leading edge vortex. However, when they interrupt the spanwise flow by putting fences along two chords on the wing, the vortex decreased in strength, disproving their hypothesis. With a full three-dimensional map of the flow around the wing, the morphology of this leading edge vortex and its interaction with the tip vortex would revealed.

These and other studies show the importance of studying the full flow structure near the edges of these bodies and the evolution and morphology of the vortical structures formed in the flow. In short, a fully volumetric three-dimensional version of DPIV would aid such research in that such flows could be mapped fully and instantaneously in three dimensions, providing measured values of $\mathbf{U}$, ω, and Γ, thus removing one more variable from the quest to understand the physical fluid phenomena.

[2] Ringuette [2004] indicates that, for aspect ratio 6, three-dimensionality ensues just after a formation time of 0.5, where formation time equals $\frac{\bar{U}t}{c}$ where c is the plate chord.

1.3 Methods available

Particle Image Velocimetry (PIV) was born in the late 1970's, first receiving its name in 1984 [Adrian, 2005]. It was evident in these early years that PIV would be an immensely powerful experimental flow analysis tool. However, the method at the time was difficult to implement, requiring complex optical setups and meticulous image analysis. The method was greatly simplified once both digital acquisition and image analysis was shown to be feasible [Willert and Gharib, 1991]. The simplification of the method together with the rapid advances in digital camera technology over the years has made Digital PIV (DPIV) commonplace in fluid mechanics laboratories.

In either its analog or digital form, PIV was limited to a two-dimensional measurement averaged over the thickness of the illumination volume, which was, for this reason, made to be as thin as possible. In DPIV the velocity of a section of the field of view is measured by cross-correlating the same section of two exposures. The average displacement of the contents of this section of the image (images of the flow markers) divided by the time elapsed between the exposures is taken to be the velocity of the fluid there.

Because standard PIV measures two components of velocity in one plane (a two-dimensional volume), recent convention in the field is to refer to the method is referred as a "2D2C" measurement.

1.3.1 Stereo PIV

Perhaps the most direct extension of PIV into the third dimension is Stereo PIV (SPIV), whereby, in essence, the single digital camera is replaced by two cameras separated by a considerable space. Perhaps the earliest publication of the currently popular method is that of Prasad and Jensen in Prasad and Jensen [1995]. The two points of view are used to extract the third component of velocity from the illuminated volume by parallax. Thus it is still a measurement of the average velocity within the illumination volume which, as in standard PIV, is made to be thin. Stereo PIV is thus a 2D3C measurement, although much of the literature, even recently, refers to it as 3D PIV. The velocity is measured exactly as in standard PIV, with the third component reconstructed from the two 2D2C measurements.

In an attempt to create a 3D3C system based on Stereo PIV, some researchers have attempted to combine two or more independent SPIV setups to create a multi-plane measurement. In essence, multiple systems operating at optically separable wavelengths acquire simultaneous images of the experiment. One obvious drawback is the need for multiple wavelengths of light and an ever increasing number of cameras and computers. Although some published results exist, the painstaking setup procedure has limited the technique.

1.3.2 Photogrammetry

Photogrammetry is the science of measuring from photographs. It has a long and rich history associated with cartography, aerial photography, and surveying. In the machine vision world, photogrammetry came to mean the reconstruction of a three-dimensional domain from several points of view. Multiple cameras are synchronized spatially through a calibration [Tsai, 1987] which also builds a mathematical pinhole model for each camera containing some distortion correction parameters. Typically the position of points in the volume is ascertained by tracing rays backwards from the image plane through the camera model into space. The result is a discrete set of points, or a point cloud. In the PIV field, this type of 3D3C is typically referred to as Particle Tracking Velocimetry (PTV) or 3D-PTV, though it can also refer to 2D2C measurements where seeding is too low to perform image correlation as is done in PIV. It has been used primarily for domains of considerable size, such as quarter-scale wind tunnels. Because typically the location of the flow markers must be reconstructed, seeding concentration is relatively low simply because all the particles in the depth of the volume must be visible in the two-dimensional space that is the camera sensor. In other words, spatial resolution is limited by the depth of the volume keeping in mind that the maximum number of flow markers that can be reliably reconstructed is in the tens of thousands (though common results are in the neighborhood of 3000 tracers per frame). Although typically these systems should be as mobile as a stereo PIV setup, they usually stay in once place, that is, they are built around the test facility.

1.3.3 Holographic PIV

Holographic PIV (HPIV) generates holograms rather than images at different instances in time, thus it has the potential to overcome the spacial resolution systems of photogrammetric systems. Unforunately this comes at the expense of a very complex and costly setup that is more an installation than an instrument. The HPIV method is reminiscent of the early days of PIV, when both the acquisition and reconstruction required a lot of skill and specialized equipment. Measurements with orders of magnitude higher seeding density than in photogrammetry are possible. A beautiful example of this technique—showing both its potential and complexity—is Barnhart, Adrian, and Papen [1994].

1.3.4 Tomographic PIV

Unveiled in Elsinga, Scarano, Wieneke, and van Oudheusden [2005], Tomographic PIV systems consist of four cameras in a bi-axial Scheimpflug arrangement aimed at the volume of interest, which is in the shape of a slab (depth is considerably shorter than the width or the height). This is an interesting 3D3C approach which could very well be considered the direct extension of SPIV into the third dimension. The images acquired are fed into a feedback loop that uses them to reconstruct a volumetric image of the domain iteratively. The seeding particles are never strictly identified; instead, the reconstruction result is a three-dimensional intensity map consisting of three-dimensional pixels. This is similar to the result of true tomographic imaging systems, such as confocal microscopy or CAT scans, where individual image slices are joined together to generate a volumetric image.

Velocity is then obtained by performing a three-dimensional image correlation. Spatial resolutions are reported to be considerably higher than those of photogrammetric systems, presumably because higher seeding densities are acceptable since reconstruction errors do not propagate past the cross-correlation stage.

1.3.5 Defocusing DPIV

Defocusing DPIV (DDPIV) can be interpreted as a subset of photogrammetry, the main advantage over which being that the reconstruction is much simpler due to the optical layout. Like photogrammetry, it can track discrete particles in place, which gives it an advantage over all the other methods in that it can also track the movement of surfaces in three dimensions. The single assembly makes it a very compact system, and, being fully digital, is much simpler to operate than HPIV. The calibration method, derived from that of SPIV, indirectly provides a more accurate camera model than the one used in photogrammetry. Its optimal arrangement requires three sensors, and thus it is he least expensive volumetric setup, with both photogrammetry and Tomo PIV requiring at least 4 sensors.

Chapter 2

Defocusing DPIV

2.1 Introduction

Defocusing DPIV (DDPIV), at its core, is a special subset of photogrammetry. Three sensors are assembled into a common faceplate such that they are coplanar and their fields of view intersect in a predetermined region called the *mappable region*. In general terms, DDPIV can estimate the depth of features based on the relative location of their images on the three sensors. In the case of fluid measurements, images of the flow markers are used to reconstruct discrete point clouds for instants in time. Velocity is computed either through tracking algorithms or three-dimensional cross-correlation on the discrete clouds.

The primary difference between the defocusing technique and photogrammetry lies in the optical layout and the calibration scheme. The optical layout affords a reconstruction method simpler than backwards ray-tracing—if the sensors are parallel, the position of the image of a point is decoupled relative to the point's Z and X, Y position.

The calibration scheme used in DDPIV is based loosely on that developed for SPIV and thus is a completely different paradigm than that used in photogrammetry (commonly based on the methods of Tsai [1987]). Rather than generating a pinhole model of the sensors with coefficients to correct for distortion, *multi-plane dewarping*, as DDPIV's calibration is called, is given the pinhole model and is made to correct the location of the images so that they fit this model. This yields more precise result than typical Photogrammetry methods, especially for volumes on the order of the instrument size, because it also accounts for differences between the pinhole model and the optical reality that do not propagate themselves as measurable distortion.

2.2 Capabilities of DDPIV

As with many of its competitors, the capabilities of DDPIV are closely linked to the processing software, but they can be categorized as capabilities of the optical system itself and limitations imposed by the software.

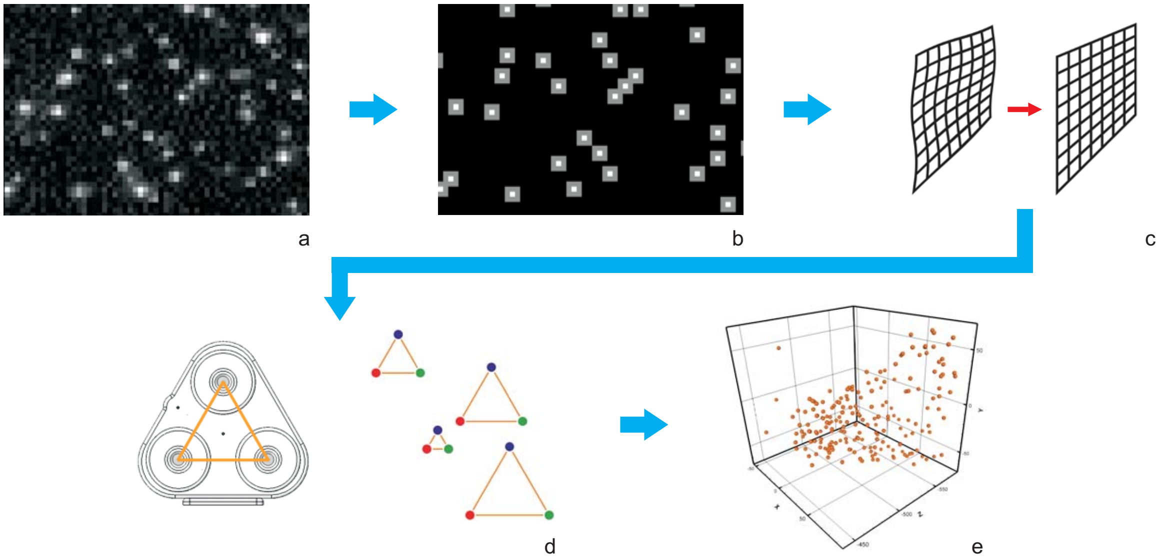
2.2.1 Capabilities of the optical system

The optical arrangement of the DDPIV camera allows for the measurement of the position of a point in space that is visible in all of the camera's sensors—as opposed to, for example, a tomographic system which images depth in a slicing manner (such as the typical ultrasound scanners used in the medical field). Thus DDPIV is able to map the surfaces of optically opaque bodies or reconstruct point clouds of suspended markers in a transparent volume.

In either case, the object in question must be surrounded by a medium of stable index of refraction. For example, in experiments involving compressible flow, the changes in the index of refraction of the fluid must be reproducible during the calibration procedure, so that they can simply be "removed" as if they were distortions in the optical path. As another example, if it is desired to map both surfaces of a thin membrane, then the index of the fluid must match that of the material of the membrane for the measurements to be accurate.

The physical result of the measurement is that any point within the mappable region will be visible in all three sensors. The images are spaced relative to each other as if the camera was a single lens and the sensors were portions of a single imager near the three off-axis apertures. Thus

Figure 2.2-1: Due to the optical arrangement of the DDPIV camera, the processing of images into three-dimensional point clouds is relatively simple. First, each raw image (a)—one for each aperture—is optionally preprocessed with blurring, normalization, etc., and then a Gaussian fitting algorithm locates each particle image and finds its center to sub-pixel accuracy (b). The list of particle image coordinates for each aperture are then dewarped using the multi-plane dewarping coefficients (c). The actual reconstruction step (d) consists only of looking for particle images that, across all the aperture images, form the same pattern as does the aperture layout. The distance between the particle that generated the particle images and the reference (focal) plane of the camera is proportional to the size of the matching pattern. At the end of the reconstruction step, a pointcloud (e) contains discrete three-dimensional particles. Two such point clouds, reconstructed from image sets exposed at different times, can be used to estimate particle velocities.



a point on the reference plane will have three images which have no relative offset between them, whereas any point off the reference plane will have three offset images as if they were portions of a single blurred image of that point.

The location of these images is of course dependent on the path that the light rays took from the object to the image. The ability to distinguish relative offset between them correlates directly with the final precision of the position measurement and depends in an indirect sense in the ability to reconstruct the path of the rays. Thus the measurement accuracy is closely related to the multi-plane dewarping calibration.

2.2.2 Capabilities of the software

Generically speaking, the software must be able to perform two tasks: first, it must identify the matching particle images of each single point in space, and second, it must be able to measure the distance between these images, taking into account the optical path of the image-forming rays.

The methods employed in performing these tasks is in some sense dependent on the application. Because DDPIV was developed primarily for fluid flow, `DDPIV` assumes the points to be reconstructed (tracer particles) have small, Gaussian-like images[1]. For a successful measurement, the sensor images must consist of distinguishable particle images of at least 4-pixel-diameter so that their sub-pixel locations can be estimated. Because these images are featureless, the only information available to match corresponding images together is the pattern formed by a particle image triplet relative to the pattern formed by the aperture layout of the camera itself. The location of the images of the triplet is corrected using multi-plane dewarping to take into account the real optics of the experimental medium and the lenses. The size of the resulting corrected pattern corresponds to the depth location of the particle with those three images.

`DDPIV` then requires any image to be processed to look like a field of particles, so when the experiment includes surface mapping, the surface must be tagged with small dots.

This, in essence, is a limitation imposed by `DDPIV` and not by the method itself. The three dimensional measurement is an optical one, thus any object generating distinguishable images can be mapped. For example, if a surface has a texture of its own, a cross-correlation-based algorithm could be used to identify and simultaneously measure the spacing of the three images of an area of the surface. Such a depth map would resemble those seen in Kanade, Yoshida, Oda, Kano, and Tanaka [1996].

[1] Technically, it is assumed that they are point sources of light and thus their images are the point spread function, which can be approximated as a Gaussian.

2.3 Requirements and limitations of DDPIV

There are some restrictions to DDPIV that result directly from the definition of the concept and construction of the instrument:

1. The probe volume size and location is fixed at design time. Although it can be adjusted somewhat during multi-plane dewarping, this usually comes at a cost of sensitivity. Fixing the measurement domain at design time allows for a solid, single-instrument construction which makes the camera essentially rigid in the sub-pixel scale—modular multi-sensor systems may require calibration after each power cycle, whereas the modern defocusing camera can hold its calibration even in the event of substantial physical shock (such as impact from a hammer), certainly surviving transportation of the camera from one lab to another.

2. When performing experiments that require imaging through thick surfaces of different media (such as into a water-filled tank), the faceplate of the camera should be positioned parallel to the surfaces (the tank wall). Multi-plane dewarping is designed to correct for aberrations in the system including tank walls, so slight bulging or inhomogeneity in a tank wall can be corrected. This is a requirement of the optical model in the software, though a considerable amount of misalignment should be removable through multiplane dewarping.

3. Seeding density can heavily affect the quality of the resulting point cloud. To establish the distance between the reference plane and a given particle, one must know the distance between the images of said particle. But once the particles are photographed, there is no information that links one particle image to another other than their relative location. Because of optical aberrations and multi-plane dewarping's inability to completely correct for these errors, there must be some tolerance in how the particle images are matched together. The more particles that are imaged, the more likely that images will be mismatched, thus generating particles that never existed—these are referred to as "ghosts". One must keep in mind that a seeding density that would create a heavily populated image in laser-sheet illumination for standard DPIV is much larger than the density required to fill an image when a substantial depth is illuminated. Some ghosts can be tolerated as they are often eliminated during the velocity estimation, nevertheless care should be taken to keep their population to a minimum. Normally images that result in a light background or "hazy" look indicate over-seeding. Unlike image correlation techniques which can also work with "blobs" or "speckles", DDPIV requires distinct particle images to work. In the end, it is the density of particle images in the exposure frames that dictate how well `DDPIV` will be able to identify real particles. It is difficult to quantify exact spatial densities for a variety of volume depths, especially considering that the illumination will also change.

4. Because the illumination must now be spread into a much larger volume (relative to a thin sheet as in DPIV), the power per unit area (luminous intensity) decreases considerably and thus small particles used in standard DPIV may not even be visible (or too dim) in DDPIV lighting. Ideally particles should be of 100-micron diameter or more for a 120 mJ per pulse laser (and a 1:1:1 volume). One should take care to illuminate the volume using a slowly diverging cone (or a completely collimated expanded beam) so that illumination intensity does not drop rapidly within the probe volume and so that the option is available to use mirrors to "reuse" the light.

Chapter 3

The Definition of Defocusing

3.1 Definition of Focus

It is important to understand that in the defocusing concept the **blur of the particle images is irrelevant in the determination of the Z position of particles**. In fact, DDPIV cameras are built to have large depths of field. Photographic objectives usually don't offer a good depth range of only slight defocus blur and thus particles would quickly disappear[1] were the depth of field not large.

In a mathematical world, a definition of focal plane would be the Z coordinate which, if a point source of light were placed there, its rays that entered a perfect lens would be bent by such a lens so that they intersect at exactly one point on the other side. In reality, there are two facts that require that this definition be expanded: first, no lens is perfect—no lens can bend rays entering it so that they intersect at one point (because of what is cumulatively referred to as "aberrations"), and second, images are formed by discrete photoreceptors—in the case of digital sensors, these units are pixels; in the case of film they are the individual clusters of "grains" (which in general are smaller for lower ISO speeds).

If we take just the first fact into account—the fact that lenses are not perfect—then we can only define "best focus" as the point at which the rays intersect at the image in the smallest possible area (since it will never intersect at just one point). This area is called the circle of least confusion (see figure 3.1-2).

If we now add the fact that sensors have a minimum discernible image element, then it is obvious that for a given sensor, a lens can only be "so good", because once the circle of least confusion is smaller than one image element it is impossible to increase the resolution of the image.

If we move the point source along Z then the location of its image also moves. If the detector

[1] As the blur on a particle image increases, its intensity decreases, since the same amount of light is now made to cover a larger area. The mathematical limit of defocus is a decrease in contrast. A good example is a dirty lens on a camera—the dust on the lens is not visible in the photographs; it is "so out of focus" that the dust propagates itself as a constant background level in the photograph.

Figure 3.1-1: A "perfect" lens, generated by optimizing with all surface, thickness, and glass parameters as variables, no constraints, and minimum spot size as the only target. Rays from a point source of light (left) 50 mm from the stop converge to exactly one "point" on the image (right). Surfaces are even aspherics; optimization resulted in a focal length of around 29 mm with a spot size in the nanometers at f/4 on-axis.

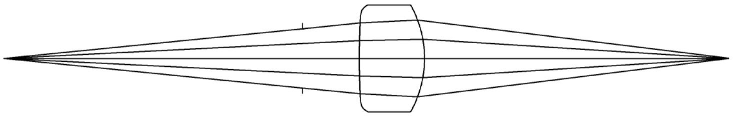
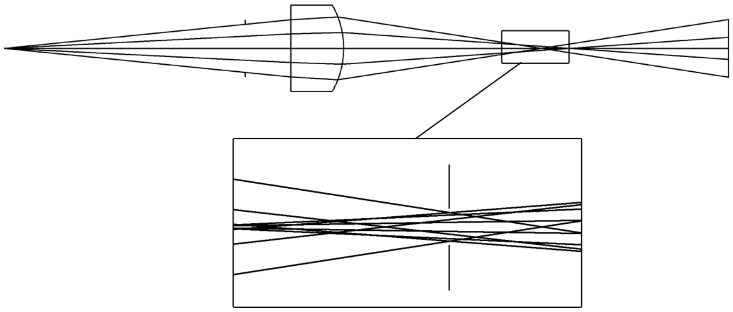


Figure 3.1-2: This lens was generated by reducing the parameters of the lens in figure 3.1-1 so that the surfaces were spherical and a real glass was being used, resulting in an imperfect lens. It has been purposely defocused so that the circle of least confusion (between the vertical lines in the inset) becomes apparent—the rays no longer intersect at a single point.



location is fixed, then the size and shape of the image of the point on the detector changes. Most notably, it changes size—this is defocus blur. To first order, the focal plane of a lens does not change with the aperture diameter[2]. Thus decreasing the diameter of the aperture has the effect that the angle of the outer (marginal) rays with the optical axis decreases. Even though the Z location of the circle of least confusion moves relative to the detector when the point source moves along Z, there will be a region in space where, due to the angle of the rays as they approach the detector, moving the point source will not increase the size of its image appreciably—this is the depth of field at the given aperture. Thus the smaller the aperture, the lower the angle and thus the longer (deeper) the depth of field.

One can imagine specifically that if the circle of least confusion at best focus is small relative to the minimum image element and the aperture is very small then there may be a large range in space where the area of intersection of the rays on the image is not larger than an image element and thus as far as the image is concerned the optical system is perfectly in focus within this entire range. In reality this is usually not possible because few lenses are sharp enough to have a focused

[2] In reality the circle of least confusion may change shape and/or position with the diameter of the aperture

image substantially smaller than a pixel at best focus and because as the aperture diameter decreases diffraction starts to affect the image as well. In DDPIV a superbly sharp system would have negative consequences, because the microscopic seeding particles are in effect “point sources” of light and if they were imaged only in a single pixel it would be impossible to obtain the sub-pixel center of these images.

In photography, the depth of field is generally defined with more arbitrary constraints based on experience and opinion (and sometimes can be as specific as to consider the final size of the photograph). For DDPIV, “in focus” refers to a *region in which the particle images do not vary appreciably in size (no more than 1 or 2 pixels) and intensity (no more than a 20% change).*

3.2 Diffraction and Its Importance to DDPIV

The image of a point source of light is not just a point but a point with diffraction rings around it, partly because as the light passes by the edge of the aperture inside the lens it is actually bent by the aperture. The exact shape of this image is called the point spread function for a particular optical system and varies with the aperture diameter. At large apertures, there is so much more light coming from the center of the lens that this increase in intensity at the edge of the aperture is not visible, but as the aperture decreases in diameter, it becomes more and more prominent until the center spot of the image is of the same intensity and effectively the image gets blurred. If the intensity and diameter of diffracted rays grows enough to affect more than just one image element on the sensor, the blur will become apparent in the resulting image. The image of the principal peak of the point spread function is called the Airy disc.

Diffraction and the ray angles play somewhat counteracting roles in defining the depth of field and sharpness of a lens at a specified aperture diameter. This is part of the reason it is common to hear photographers say that lenses are sharpest at mid-range aperture size (between fully open and minimum aperture)—at some point the point spread function will grow due to diffraction past the size of the circle of least confusion at a larger aperture and eventually past the size of the smallest image element[3]. Equation 3.2-1 can be used to estimate the Airy disc diameter for a lens at a given f-number; λ is the wavelength of the light, f is the f-number of the lens, and d is the diameter of the Airy disc in the units of the wavelength.

$$d \approx 2.44\lambda f \tag{3.2-1}$$

[3] For a good demonstration of diffraction in photography, visit http://www.cambridgeincolour.com/tutorials/diffraction-photography.htm. From there you can go back to the main tutorial link which has great explanations to common questions such as pixel size, depth of field, etc.

Figure 3.2-1: 40-micron particles imaged at f/5.6. A laser cone illuminates particles in a tank of water, which are imaged through a 45 mm focal length lens at f/5.6 onto a 7.4-micron-pixel CCD. The image shown here is 100 by 100 pixels. The short depth of field is evident as few particles look sharp. Just outside the focal region the particle images are blurred blobs. Outside this region the particles are imaged as rings of the same shape as the aperture diaphragm. This is due to diffraction concentrating more light near the edges of the ray bundle than in the center.

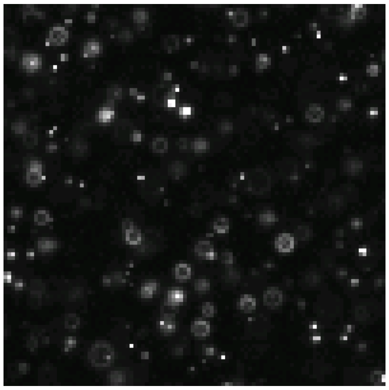


Figure 3.2-2: The same volume as in figure 3.2-1, but imaged at f/8. There are less ring-like images and more blob-like images as the depth of field expands due to the decrease in aperture size.

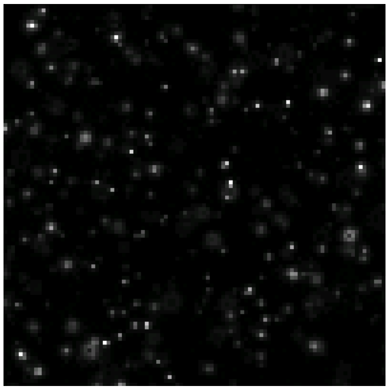


Figure 3.2-3: The same volume as in figure 3.2-1, but imaged at f/11. Now there are no visible ring images, but note how the particle images in sharpest focus occupy primarily only one pixel.

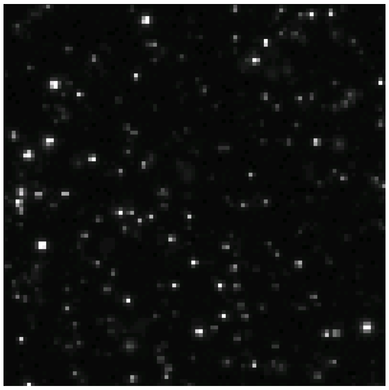


Figure 3.2-4: The same volume as in figure 3.2-1, but imaged at f/16. Sharpest focus images have increased in size a bit due to diffraction.

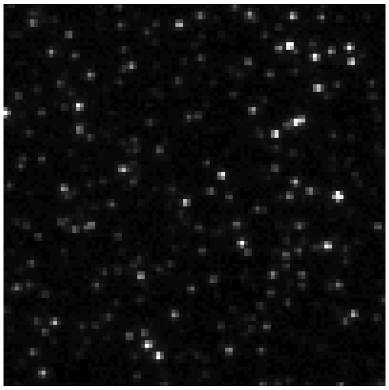


Figure 3.2-5: The same volume as in figure 3.2-1, but imaged at f/22. Diffraction is clearly evident, as now all visible particle images are more or less the same size—indicating a long depth of field and blurring due to diffraction. (There is an increase in the noise of the image because the gain had to be increased to accommodate the higher f-number.)

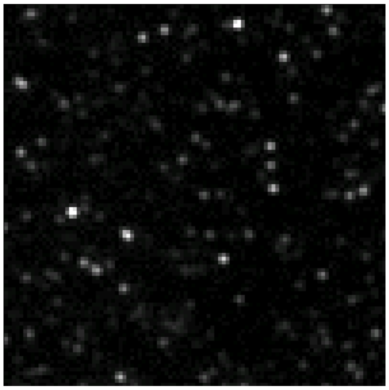


Figure 3.2-6: This logarithmic contour plot shows the calculated point spread function (image of a focused point source of light) for the Edmund Optics 50 mm Double-Gauss lens (stock number U54-690) at f/4. The central, bright part of the spot (the Airy disc) is 4 microns in diameter.

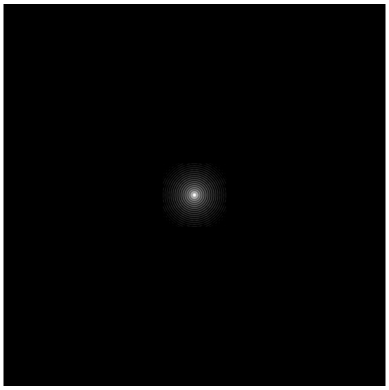
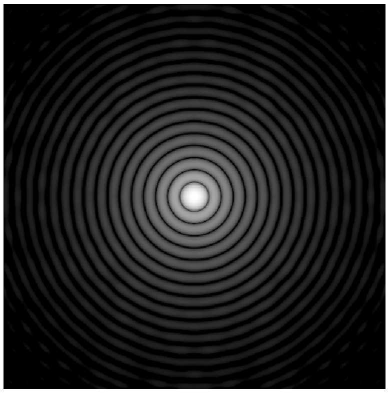


Figure 3.2-7: This logarithmic contour plot shows the calculated point spread function for the lens of figure 3.2-6 at f/18 (the minimum aperture). The Airy disc is 24 microns in diameter—several pixels on most digital sensors today.

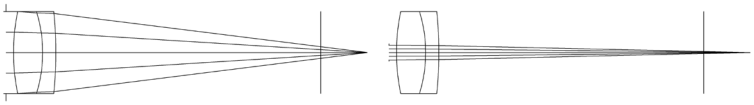


It is imperative to note that simply adding more light will, in general, not expand the size of the particle image on the sensor[4]. Without physical blurring there is practically no way to increase the image size of a particle.

3.3 Definition of Defocusing

Now picture the rays coming out of a lens with a large aperture as they approach the image, as in figure 3.3-1.

Figure 3.3-1: Shown is an Edmund 25 mm Achromat (stock number U32-305) defocused at f/4 (left) and f/22 (right). Notice how with the smaller aperture, the rays converge at a more horizontal slope. This shows qualitatively that the depth of field will be greater at f/22 because as the point source is moved toward the lens (and its image moves to the right) the distance between the marginal (outer) rays at the image (the vertical line) will grow more slowly than at f/4.



A small change in the Z position of the point source will increase the image size—the larger the aperture, the steeper the angle of the rays, and the more quickly the image will grow with a change in Z.

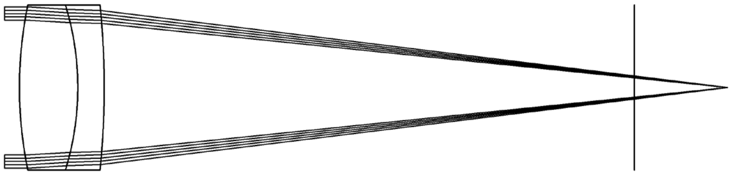
One can see that it would be possible to estimate the Z location of the point source (relative to the focal plane) from the size of its image on the sensor, but the more sensitive the system is made to be (the larger the aperture), the larger the images will be, and the more they will intersect with each other. Moreover there is the added problem of intensity—the particles will reflect an equal

[4] Unless perhaps the increase in light is substantial, such as going from volume illumination to a sheet

amount of light regardless of the system aperture size, thus the intensity of a particle image is (more or less) inversely proportional to the diameter of the image squared.

The defocusing concept was born by "cropping" these images so that the depth of field of the portions is long, thus allowing for images of more consistent size and intensity (see figure 3.3-2). By replacing the large centered aperture with a small off-axis one (which only allows the marginal rays through), the depth information carried by the blur is still available. Multiple apertures are needed to decouple the X, Y of the particle from its Z. In practice, three apertures were used as there is too much ambiguity in matching particle images together when only two apertures are present. An equilateral triangle centered about the axis of the lens was the logical choice as the three images formed for each particle moved at proportionally equal rates with respect to Z[5]. This was the work of the original paper, Willert and Gharib [1992].

Figure 3.3-2: Here the single f/4 aperture on the left of figure 3.3-1 has been replaced with two smaller apertures. Note how instead of one large image there are now two images of the same point source. This is the heart of the defocusing technique.



With the long depth of field of a given aperture, the individual images are small throughout a large range in Z, but their position relative to each other would change in the same way the diameter of the blurred image from a single aperture would change. Thus when three images are clearly discernible[6] when imaging one point source the lens is said to be "defocused" *even though the individual images by themselves are in focus by our definition of depth of field.* This is the most demonstrative description of the defocusing technique.

The X, Y coordinate of a given particle is then the centroid of the pattern formed by its multiple images, and its distance from the focal plane is proportional to the distance between its images b. The *sensitivity* is the rate of change of the separation with Z: $\frac{db}{dZ}$.

Because the images themselves can be in focus over a substantial range of Z's, the focal plane (the Z at which the three images coincide) is sometimes referred to as, more specifically, the reference plane. This also clarifies discussion of the pinhole model of the defocusing camera (see chapter 4)

[5] A triangle of apertures also has the advantage that it allows to separate particles behind the focal plane from those in front of it as the image pattern flips across that plane.

[6] they may still be overlapping

as by definition pinhole optics imply an infinite depth of field.

3.4 Evolution of Defocusing

Keeping the lens focal length and sensor geometry (resolution and size) constant, the ratio between the aperture separation and the distance from the apertures to the reference plane defines the sensitivity (and thus precision) in Z. The larger this ratio, the faster the three particle images grow apart with a change in Z.

The closer the focal plane is to a lens, the shorter the depth of field[7]. Thus a limitation arises because the depth of field must be large enough to encompass the desired measurement volume.

The closer the apertures are to the edge of the lens, the lower the image quality. Lenses are normally manufactured with spherical surfaces to make them cheaper and simpler. This introduces aberrations. *Chromatic* aberrations arise from the fact that the speed of light in a medium varies with the wavelength of the light, thus the paths through different media are different for different wavelengths of light. The *dispersion* of a glass is the relative variation of the index of refraction versus wavelength. Both the dispersion and the thickness of an element define how much it “separates” the light going through it. In terms of PIV, chromatic aberration is technically not important because the illumination is usually from a single-wavelength laser.

Other aberrations arise from the fact that the focal plane is not really a plane but rather a curved surface, and as the field of view becomes larger angularly, distortion can appear—straight lines will be imaged as curved lines off-center from the lens. Other higher-order aberrations can also appear—these include astigmatism and coma, which distort a circular point so that its image is no longer a circular dot. Astigmatism is a difference in the focal location of perpendicular ray bundles and will distort the image into an ellipse. Coma can generate images that look like little comets (hence its name).

Traditional lenses with spherical components are designed, in essence, so that aberrations from one element help to counteract those of others. Even with the computerized design methods employed today, lenses are designed so that aberrations are *balanced* along the radius of its image; in other words, sometimes performance at the center of the lens is sacrificed to prevent the outer edges from getting too bad. So even though chromatic aberration is not a factor in PIV it may become a factor in that this is one of the aberrations which is “balanced” in commercial lenses.

Even then, in general, the outer portion of a lens performs much worse than the center. So when the aperture diameter is decreased to a mid-range diameter as described above, rays going through the “bad” portions of the lens are removed from the image, increasing the quality of the image.

So by moving the apertures off center as in the defocusing technique, *only* these “bad” rays are

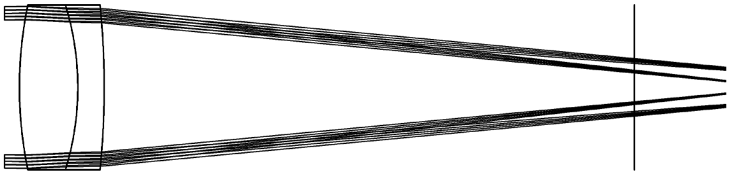
[7] Depth of field should be thought of as approximately depending only on magnification.

Figure 3.4-1: On the left are rays through a 4-millimeter-diameter centered aperture and the lens in figure 3.3-1; on the right the same aperture has been shifted 2 millimeters radially. Note that not only is the spot size larger, but the rays are no longer symmetrically arranged, indicating that the resulting image will not be a circular dot.

used to form the image. A limit is thus reached quickly with how far apart the apertures can be placed. The limit also arises from the physical size of the lens. A longer focal length lens can be made with a large diameter (at exponentially increasing expense) but to get a reasonable size field of view the focal plane must be moved far from the lens, automatically reducing the sensitivity. To get a reasonable field of view close to the lens, the focal length must be short. This requires a small radius of curvature in the elements which increases the spherical aberrations and limits the diameter of the lens (before it simply becomes a sphere).

In the end, the limits imposed by depth of field, diffraction, aberrations, and lens geometry limit a single-lens system to dissatisfying sensitivity levels.

Figure 3.4-2: The aperture arrangement of figure 3.3-2 is shown imaging a point source 300 millimeters away (inner rays) and another 100 millimeters away (outer rays). Notice the image separation has grown by just about the image diameter and even with these small (0.5-millimeter-diameter) apertures the image size increases quite a bit between the two. This is an extreme example of the lack of sensitivity in single-lens systems as the lens is only 6 millimeters in diameter.



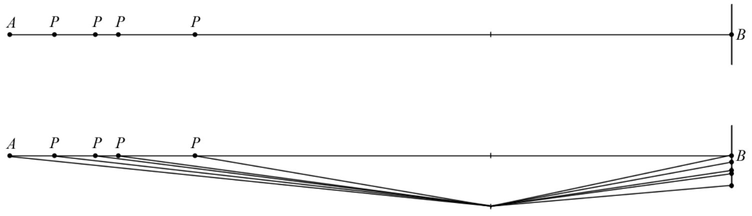
A secondary problem is that of image crowding: with three apertures, each point will now generate three images, and the more images that appear the harder it is to properly match triplets based only on the aperture pattern. If each aperture had its own sensor, then not only could in theory the particle density increase by a factor of three, but the particle image matching would benefit from the additional knowledge of knowing which particle image came from which aperture.

This problem was tackled early on in the development of DDPIV. Initial attempts included color-masking each aperture, then using a color-separating prism (as found in 3-CCD camcorders) to divert each color (thus each aperture) to a separate sensor. This required white light for illumination and reintroduced chromatic aberration into the list of problems (though conceivably they could be calibrated out). A second attempt separated the images by simply diverting the rays through each aperture laterally out to a sensor via a mirror. Aside from manufacturing difficulties, the geometry and aberration-induced aperture separation limits still had not been tackled and thus the sensitivity was still low. None of these cameras ever worked satisfactorally.

3.4.1 Separation into Three Lenses

Regardless of the aperture arrangement—whether there are apertures off axis or a single one on the axis—in a camera with an infinite depth of field (pinhole optics), the image of a point is the locus of all points on the ray which forms this image. Said differently, all points P between point A on the focal plane and the center of an aperture will have the same image at point B on the image plane—thus a single aperture system does not store depth information in an image since it is impossible to know which point P actually formed the image (see figure 3.4-3, top).

Figure 3.4-3: In pinhole optics, multiple points along the same ray (points P) will form an image in the same exact place (point B) as that of a point A at the "focal" plane (top). If a second aperture is added off-axis, both apertures will image point A at point B but all other points P will have different images through the second aperture as they are no longer on the same ray through that aperture. Thus with two apertures, depth information can be recovered independently of blur.

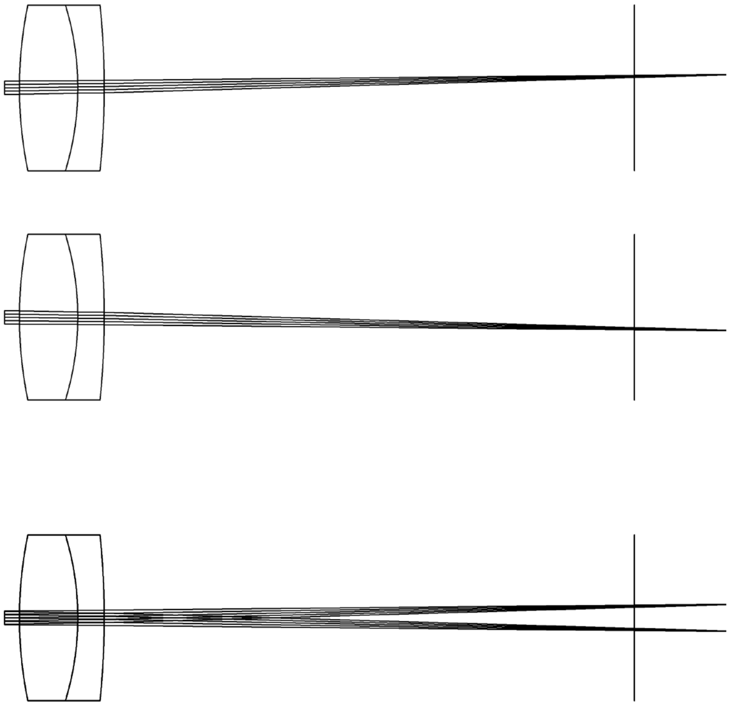


However the if there are multiple apertures in the system the depth information can be recovered because each image (from each aperture) will represent a different ray in space which, if each image comes from the same particle, should intersect at exactly one point.

Looked at it another way, a point cannot move along two different directions at once and thus there will always be depth information as long as there are at least two apertures.

This depth information can be obtained as in the original defocusing concept (three apertures in one lens) or as in the present implementation (one aperture in each of three lenses—see figure 3.4-4). As long as the three lenses do not share an optical axis (which they could only share if they at the exact same location in space) the "defocusing" effect would still be present. This is even true if the axes of the lenses are parallel (which is the case of DDPIV cameras). In this sense the sensitivity of the system can be interpreted as how far particles can be from the optical axis of each lens versus the distance from the aperture which conceptually is the same measurement as the aperture separation versus distance to the reference plane was for the sensitivity of a single-lens system.

Figure 3.4-4: The top of this figure shows the same point imaged through two lenses that are offset from each other. The bottom shows the two lenses' ray traces superimposed on each other, showing how two offset lenses can obtain depth information exactly as a multiple-aperture single-lens setup would.



Separating the system into three lenses allows for much larger separations while conserving image quality and simultaneously separating the apertures each to one sensor[8]. Limits of sensitivity are now imposed only by the illumination and geometry of the image space of the lenses chosen. In theory then photographic objectives intended for larger-format photography could yield much higher sensitivity than smaller formats, however, in practice it seems the limiting factor is the light fall-off. Lenses experience light fall-off as a function of radial distance from the axis, and this effect seems to be amplified by the linearity of CCD exposure[9]. In the end, image quality is no longer a concern in these cases as the images will be too dark to discern well before the edge of the realm of unacceptable aberrations is reached.

The three-lens arrangement was introduced by Pereira at the ONR workshop in San Diego in February 1999 and subsequently at PIV '99 (see Pereira, Gharib, Modarress, and Dabiri [1999b]).

[8] which also has the added advantage that at the reference plane three distinct points can still be measured.

[9] Some argue that the microlenses implemented in interline-transfer CCD's also contribute to the light fall-off.

The paper from the latter was published in 2000 (see Pereira, Gharib, Dabiri, and Modarress [2000a]). A second paper in 2002 analyzed the method in more detail (see Pereira and Gharib [2002]).

Chapter 4

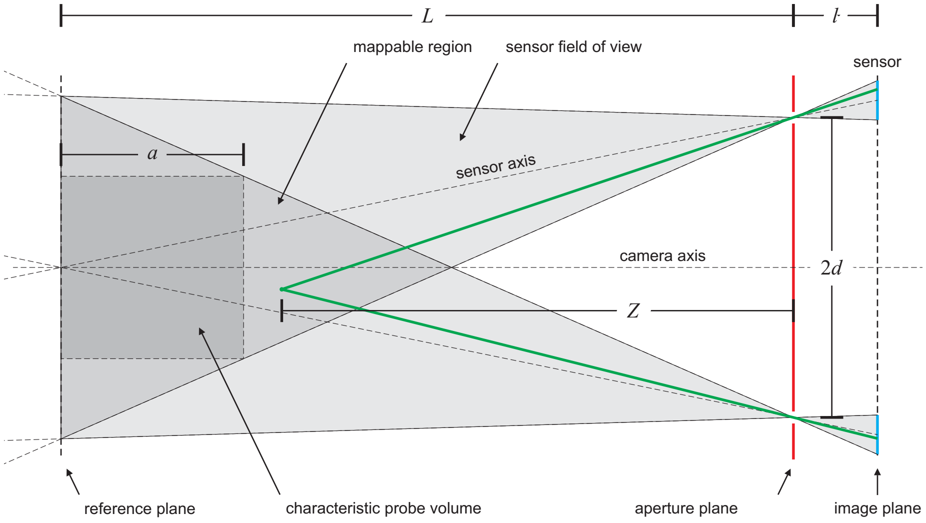
Pinhole Optics Approach

4.1 Introduction

The mathematical explanation of defocusing cameras and the basis of the algorithms of the processing software are based on pinhole optics. In pinhole optics, there are no lenses, and apertures represent points through which only single rays of light can pass. All relationships are then based on similar geometry.

The key quantities discussed are exemplified in the 2-aperture depiction in figure 4.1-1.

Figure 4.1-1: Pinhole-optics diagram of a two-aperture defocusing arrangement.



L is the distance from the aperture plane to the reference plane, l is the distance between the aperture plane and the image plane, $2d$ is the distance between the two apertures, a is the side length of the characteristic probe volume, and Z is the distance between a particle and the aperture plane. In this depiction the apertures are centered about the optical axis of the camera so that the distance between an aperture and the center of the aperture plane is d.

The fields of view of the sensors are shown here as shaded in light grey (they of course continue past the reference plane but are not shaded past it); they are constructed by taking lines from the edges of the sensors and passing them through the center of the respective apertures. The *mappable region* is the region where the fields of view intersect, and in theory any point in this region is measurable by the camera. It is shown here as the darker grey triangle. The *characteristic probe volume* (sometimes referred to as just probe volume) is the rectangular prism whose short cross section is the largest inscribable square in the cross section of the mappable region. In this two-dimensional case, then, it is just a square.

4.2 Developing a Ray Tracing Model

4.2.1 Introduction

To derive the equations of the pinhole model of a defocusing camera it is necessary to be able to trace rays in the pinhole-optics sense, which is typically just a matter of using geometric similarity to measure unknown lengths. Here we present a more tactile method of physically finding intersection points of lines and planes. This makes it very easy to extend the theory into three dimensions (as is done here).

The following derivations follow closely in concept those of Willert and Gharib [1992] and Pereira and Gharib [2002], where the analysis was presented in a two-dimensional case for simplification, and will be compared to that in the latter reference from time to time. An alternative, though less general analysis is performed in Kajitani and Dabiri [2005]. The standard formulas for planes and intersections below are taken from Tuma and Walsh [1997].

4.2.2 Pinhole-Optics Requirements for Defocusing Cameras

The rules that govern alignment for defocusing are simple. They can be divided into requirements essential to the measurement of deep volumes and requirements typical of optimized defocusing arrangements.

The essential requirements are as follows:

- The sensors should all be coplanar (on the image plane).
- The apertures (considered to be points in space) should all be coplanar (on the aperture plane).
- The image plane should be parallel to the aperture plane.

- The fields of view of the sensors intersect in a common region—the *mappable region*—the position of any point in which can be measured by the camera.

These constitute the core of the defocusing method. These can be extended, along with some definitions, to a more optimized setup:

- The *sensor axes* connect the center of the sensors to their corresponding apertures and should all intersect at exactly one point.
- The *optical axis* of the camera is perpendicular to the aperture plane and passes through the point of intersection of the sensor axes.
- The reference plane is the plane perpendicular to the optical axis at the point where the sensor axes and optical axis intersect.

Note that in the pinhole-optics model there is no requirements for anything dealing with lenses, including lens focal length, aperture size, and focal plane location[1]. The above requirements have some consequences:

- The sensors all have the same magnification.
- The fields of view of all the sensors coincide exactly at the reference plane.
- The mappable region's location and size are governed by the distance from the aperture plane to the reference plane, the distance from the aperture plane to the image plane, and the aperture layout pattern.

These rules reduce the necessary parameters to define a camera to chosen values for L, f (focal length of the lens), the aperture layout pattern, and the size of the sensors.

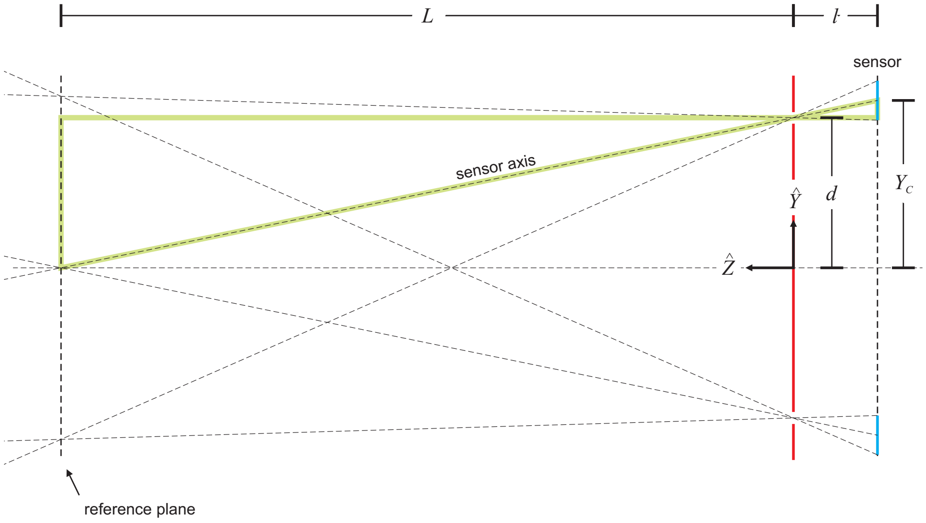
In the following discussion, refer to figure 4.2-1. The two brown triangles in the figure are similar. They are formed by drawing a line parallel to the optical axis through the aperture and linking the ends of the sensor axis to this line at the image plane and reference plane. Thus their proportions are established by the distances along the Z direction L and l.

The assumption that was made in Pereira and Gharib [2002] is that the focal plane should coincide with the reference plane. Here we will do the same, though this is not necessary. If we place an imaginary lens of focal length f at the aperture, then we can relate l, L, and f together by the thin lens equation (equation 4.2-1). This is the only place where any lens parameter enters the derivation.

$$\frac{1}{L} + \frac{1}{l} = \frac{1}{f} \tag{4.2-1}$$

[1] In pinhole optics, the depth of field is infinite. It can be thought of as an optical system where only one ray can travel through an aperture, thus there are no other rays to form a blurred image of any point in space.

Figure 4.2-1: Schematic of aperture-sensor layout.



If we call the center of the sensor point $\boldsymbol{C} = (X_C, Y_C, Z_C)$, then based on similar triangles in the defocusing arrangement we arrive at the following proportionality:

$$\frac{Y_C - d}{d} = \frac{l}{L} \tag{4.2-2}$$

An equivalent quantity exists involving X_C. Proportionalities of this type (that is, corresponding sides of similar triangles) are equal to the *magnification* for that particular Z.

Z_C is simply $-l$ (but by focusing the lens at the reference plane, we can replace it with its equivalent $fL/(L-f)$ from equation 4.2-1 since this form includes only camera design parameters), and thus the coordinate of the center of the sensor in question is

$$\begin{aligned} X_C &= c\left(1 + \frac{f}{L-f}\right) \\ Y_C &= d\left(1 + \frac{f}{L-f}\right) \\ Z_C &= -\frac{fL}{L-f} \end{aligned} \tag{4.2-3}$$

which simplifies to

$$\begin{aligned} X_C &= \quad \frac{cL}{L-f} \\ Y_C &= \quad \frac{dL}{L-f} \\ Z_C &= - \frac{fL}{L-f} \end{aligned} \tag{4.2-4}$$

4.2.3 Equations of Planes

In preparation of our ray-tracing method, we must repeat some definitions from Tuma and Walsh [1997]. The equation of a plane in space in general form is

$$AX + BY + CZ + D = 0 \tag{4.2-5}$$

The equation of a plane in space that passes through three points (X_i, Y_i, Z_i), (X_j, Y_j, Z_j), and (X_k, Y_k, Z_k) has coefficients

$$\begin{aligned} A &= \begin{vmatrix} Y_i & Z_i & 1 \\ Y_j & Z_j & 1 \\ Y_k & Z_k & 1 \end{vmatrix} \qquad B = \begin{vmatrix} Z_i & X_i & 1 \\ Z_j & X_j & 1 \\ Z_k & X_k & 1 \end{vmatrix} \\ C &= \begin{vmatrix} X_i & Y_i & 1 \\ X_j & Y_j & 1 \\ X_k & Y_k & 1 \end{vmatrix} \qquad D = - \begin{vmatrix} X_i & Y_i & Z_i \\ X_j & Y_j & Z_j \\ X_k & Y_k & Z_k \end{vmatrix} \end{aligned} \tag{4.2-6}$$

The equation of a plane in space through a point $\boldsymbol{X_0} = (X_0, Y_0, Z_0)$ with normal $\boldsymbol{n} = (n_X, n_Y, n_Z)$ is

$$\boldsymbol{n} \cdot (\boldsymbol{X} - \boldsymbol{X_0}) = 0 \tag{4.2-7}$$

We can expand this expression to get the quantities necessary to write the equation in the form of equation 4.2-5 to get

$$n_X X + n_Y Y + n_Z Z + (-n_X X_0 - n_Y Y_0 - n_Z Z_0) = 0 \tag{4.2-8}$$

and thus our coefficients are

$$\begin{gathered} A = n_X \quad B = n_Y \quad C = n_Z \\ D = -(n_X X_0 + n_Y Y_0 + n_Z Z_0) \end{gathered} \tag{4.2-9}$$

The intersection of three distinct planes is a point. If the three planes have equations in the standard form as in equation 4.2-5, then the point $\boldsymbol{R} = (X_R, Y_R, Z_R)$ at which the three intersect (assuming they do, of course) is

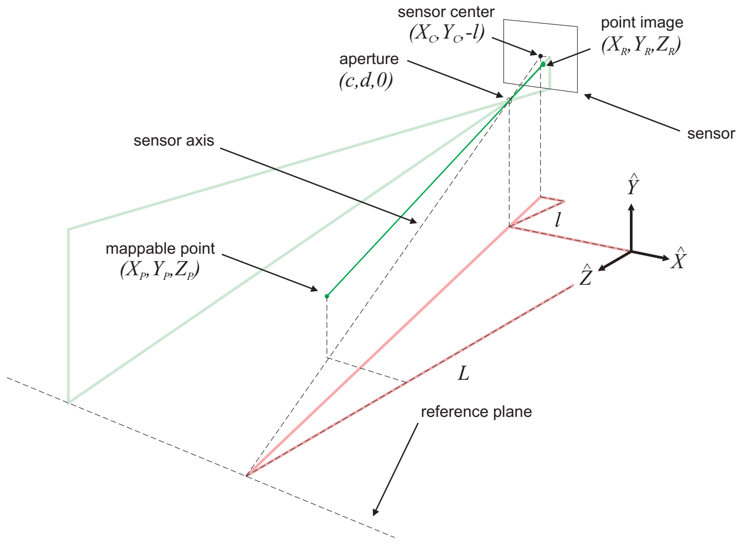
$$X_R = -\frac{\begin{vmatrix} D_1 & B_1 & C_1 \\ D_2 & B_2 & C_2 \\ D_3 & B_3 & C_3 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{vmatrix}} \quad Y_R = -\frac{\begin{vmatrix} A_1 & D_1 & C_1 \\ A_2 & D_2 & C_2 \\ A_3 & D_3 & C_3 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{vmatrix}} \quad Z_R = -\frac{\begin{vmatrix} A_1 & B_1 & D_1 \\ A_2 & B_2 & D_2 \\ A_3 & B_3 & D_3 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{vmatrix}} \tag{4.2-10}$$

In general, two planes written in the form of equation 4.2-5 will intersect as long as

$$A_1 : B_1 : C_1 \neq A_2 : B_2 : C_2 \tag{4.2-11}$$

4.2.4 Ray-Tracing with Planes

Figure 4.2-2: Schematic for 3D pinhole optics, showing a point in space (X_P, Y_P, Z_P), an aperture at $(c, d, 0)$, the point's image on the sensor at (X_R, Y_R, Z_R), and the sensor's center $(X_C, Y_C, -l)$. Two sets of similar triangles are marked: the pink one allows for calculation of the lateral offset of the sensor relative to the aperture, and the green one can be used to calculate the field of view at the reference plane.



To find the pixel coordinates of a point in space, three planes will be defined. The first two planes' intersection defines the light ray passing through the aperture. The third plane is that of the sensor itself.

The first two planes will be defined by three points and thus equation 4.2-6 will be used to calculate their coefficients. Using the standard axis layout for DDPIV (Z being the camera axis direction, X being horizontal, and Y being vertical, with the origin at the intersection of the aperture plane with the optical axis), given a point $\boldsymbol{P}$ in space (X_p, Y_p, Z_p), and given that the aperture in question is located at $(c, d, 0)$ (see figure 4.2-2), then for the first ray plane we choose the points

$$(X_p, Y_p, Z_p) \quad (c, d, 0) \quad (c-1, 0, 0) \tag{4.2-12}$$

Note that the third point is arbitrary—we are only interested that the plane contain the aperture and the point in question. Thus to generate a second plane, we can change the arbitrary point; here we take

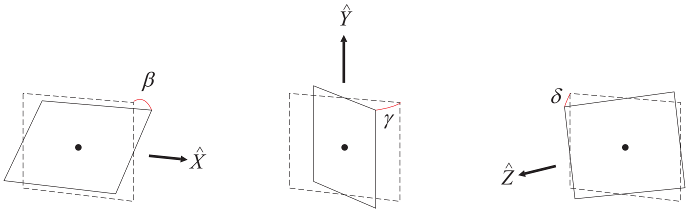
$$(X_p, Y_p, Z_p) \quad (c, d, 0) \quad (0, d-1, 0) \tag{4.2-13}$$

To define a general sensor plane, we will allow for deviation from the ideal defocusing arrangement—misalignment. We will call the linear misalignments ΔX, ΔY, and ΔZ, so that the quantities in equation 4.2-4 become

$$\begin{aligned} X_C = & \quad \frac{cL}{L-f} + \Delta X \\ Y_C = & \quad \frac{dL}{L-f} + \Delta Y \\ Z_C = & - \frac{fL}{L-f} + \Delta Z \end{aligned} \tag{4.2-14}$$

The angular misalignments will be represented as angles defined as follows: β is rotation about X, with 0 being perfectly vertical, γ is rotation about Y, with 0 coinciding with the image plane, and δ is the rotation about Z, with 0 being the horizontal position[2]. Commonly when dealing with cameras these angles are referred to as tilt, pan, and roll, respectively. The angular misalignments are factored in by including them in the definition of the plane in the sensor, for which we will use the normal-point version of the general form (equation 4.2-9).

Figure 4.2-3: Convention for a rotated sensor, showing the angles in order of rotation.



Compound three-dimensional rotations are easy to deal with mathematically if the angles are defined as Euler angles (meaning that order of rotation matters). Consider the basis of the normal

[2] In reality, these are treated as Euler angles, with the order of rotation as listed, so only the first rotation is about the real axes; the other two are about the newly rotated axes. Refer to figure 4.2-3

space coordinate system XYZ, $\hat{\boldsymbol{e}}^i$. If we take a transformation (for example, a rotation) $\boldsymbol{T}^{(1)}$ and apply it to this basis, we will get a new basis $\hat{\boldsymbol{f}}^i$. Mathematically, we can write the transformation in summation form as

$$\hat{\boldsymbol{f}}^i = \sum_{\alpha} \boldsymbol{T}^{(1)}_{\alpha i} \hat{\boldsymbol{e}^i} \tag{4.2-15}$$

Now we can define another transformation based on the space of $\hat{\boldsymbol{f}}^i$, call it $\boldsymbol{T}^{(2)}$, yielding a basis $\hat{\boldsymbol{g}}^i$ which in turn can be transformed by $\boldsymbol{T}^{(3)}$ to yield $\hat{\boldsymbol{h}}^i$. Thus we can combine the three transformations by substitution to get a relation between $\hat{\boldsymbol{g}}^i$ and $\hat{\boldsymbol{e}}^i$:

$$\begin{gathered}\hat{\boldsymbol{f}}^i = \sum_{\alpha} \boldsymbol{T}^{(1)}_{\alpha i} \boldsymbol{e}^{\hat{\alpha}} \quad \hat{\boldsymbol{g}}^j = \sum_{\eta} \boldsymbol{T}^{(2)}_{\eta j} \hat{\boldsymbol{f}}^{\eta} \quad \hat{\boldsymbol{h}}^k = \sum_{\zeta} \boldsymbol{T}^{(3)}_{\zeta k} \hat{\boldsymbol{g}^{\zeta}} \\ \Rightarrow \hat{\boldsymbol{h}}^k = \sum_{\zeta,\eta,\alpha} \boldsymbol{T}^{(3)}_{\zeta k} \boldsymbol{T}^{(2)}_{\eta\zeta} \boldsymbol{T}^{(1)}_{\alpha\eta} \boldsymbol{e}^{\hat{\alpha}}\end{gathered} \tag{4.2-16}$$

Remembering that matrix multiplication can be written as

$$[\boldsymbol{AB}]_{ij} = \sum_{k} \boldsymbol{A}_{ik} \boldsymbol{B}_{kj} \tag{4.2-17}$$

we can rewrite our sum as

$$\hat{\boldsymbol{h}}^k = \sum_{\alpha} \left[\sum_{\zeta} \left(\sum_{\eta} \boldsymbol{T}^{(1)}_{\alpha\eta} \boldsymbol{T}^{(2)}_{\eta\zeta} \right) \boldsymbol{T}^{(3)}_{\zeta k} \right] \boldsymbol{e}^{\hat{\alpha}} \tag{4.2-18}$$

and thus

$$\hat{\boldsymbol{h}}^k = \boldsymbol{T}^{(1)} \boldsymbol{T}^{(2)} \boldsymbol{T}^{(3)} \boldsymbol{e}^{\hat{\alpha}} \tag{4.2-19}$$

Now that we have the basis after our transformation, we can calculate the i^{th} component of $\hat{\boldsymbol{h}}^k$ by dotting it with the appropriate basis vector from the reference coordinate space:

$$\hat{\boldsymbol{h}}^k \cdot \hat{\boldsymbol{e}}^{\alpha}_i = [\boldsymbol{T}^{(1)} \boldsymbol{T}^{(2)} \boldsymbol{T}^{(3)}]_{ik} \tag{4.2-20}$$

Again, because our angles are defined as Euler angles, our transformations are rotations about a single axis, so we can write them as

$$
\begin{aligned}
\boldsymbol{T}^{(1)} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{pmatrix} \\
\boldsymbol{T}^{(2)} &= \begin{pmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{pmatrix} \\
\boldsymbol{T}^{(3)} &= \begin{pmatrix} \cos\delta & \sin\delta & 0 \\ -\sin\delta & \cos\delta & 0 \\ 0 & 0 & 1 \end{pmatrix}
\end{aligned}
\tag{4.2-21}
$$

For reference, the complete 3-axis rotation matrix is

$$
\boldsymbol{T}^{(t)} = \begin{pmatrix} \cos\delta\cos\gamma & \cos\gamma\sin\delta & \sin\gamma \\ -\cos\beta\sin\delta - \cos\delta\sin\beta\sin\gamma & \cos\beta\cos\delta - \sin\beta\sin\delta\sin\gamma & \cos\gamma\sin\beta \\ \sin\beta\sin\delta - \cos\beta\cos\delta\sin\gamma & -\cos\delta\sin\beta - \cos\beta\sin\delta\sin\gamma & \cos\beta\cos\gamma \end{pmatrix}
\tag{4.2-22}
$$

In the new basis $\hat{\boldsymbol{h}}^i$ defined by this transformation, $\hat{\boldsymbol{h}}^1$ is the sensor's x axis, $\hat{\boldsymbol{h}}^2$ is its y axis (corresponding to an image's horizontal and vertical axes, respectively), and $\hat{\boldsymbol{h}}^3$ is the direction normal to the sensor plane. We can calculate these vectors using equation 4.2-20, which tells us that they are simply the columns of the matrix in equation 4.2-22:

$$
\begin{aligned}
\hat{\boldsymbol{h}}^1 &= (\cos\delta\cos\gamma, -\cos\beta\sin\delta - \cos\delta\sin\beta\sin\gamma, \sin\beta\sin\delta - \cos\beta\cos\delta\sin\gamma) \\
\hat{\boldsymbol{h}}^2 &= (\cos\gamma\sin\delta, \cos\beta\cos\delta - \sin\beta\sin\delta\sin\gamma, -\cos\delta\sin\beta - \cos\beta\sin\delta\sin\gamma) \\
\hat{\boldsymbol{h}}^3 &= (\sin\gamma, \cos\gamma\sin\beta, \cos\beta\cos\gamma)
\end{aligned}
\tag{4.2-23}
$$

Using point $\boldsymbol{C}$ as defined in equation 4.2-14 and the normal $\hat{\boldsymbol{h}}^3$ with equation 4.2-9 we can calculate the coefficients of the sensor plane so that it is in the general form of equation 4.2-5 and obtain

$$
\begin{aligned}
A_{CCD} = \sin\gamma \qquad B_{CCD} &= \cos\gamma\sin\beta \qquad C_{CCD} = \cos\beta\cos\gamma \\
D_{CCD} &= -\left(\tfrac{cL}{L-f} + \Delta X\right)\sin\gamma \\
&\quad + \left(\tfrac{dL}{L-f} + \Delta Y\right)\cos\gamma\sin\beta \\
&\quad - \left(\tfrac{fL}{L-f} + \Delta Z\right)\cos\beta\cos\gamma
\end{aligned}
\tag{4.2-24}
$$

And, using equation 4.2-6 with the points we defined in equations 4.2-12 and 4.2-13, we can calculate the coefficients of the two planes that define the light ray:

$$\begin{aligned} A_1 =& - dZ_P & A_2 =& - Z_P \\ B_1 =& Z_P & B_1 =& cZ_P \\ C_1 =& d(X_P - c + 1) - Y_P & C_2 =& - c(Y_P - d + 1) + X_P \\ D_1 =& dZ_P(c-1) & D_2 =& - cZ_P(d-1) \end{aligned} \tag{4.2-25}$$

Note that if different points are used than those in equations 4.2-12 and 4.2-13, the planes will be different (and correspondingly the coefficients in equation 4.2-6), but the result should be the same—by picking $\boldsymbol{P}$ and the aperture location as two of the points, you are ensuring that the intersection will be the light ray from the particle through the aperture. However one must be careful in choosing the third points by defining them relative to the aperture location so to avoid two points coinciding. For example, setting the third point to $(0, 0, 0)$ will make the calculation fail if the aperture is located at $c = 0, \;\; d = 0$ because then two of the definition points will coincide. Thus it is safest to choose third points which differ from the second point in at least one component by an additive term.

The intersection of these three planes will yield the space coordinates of point $\boldsymbol{R}$, which must be converted to the coordinate plane of the sensor by taking the inner product of the vector $\boldsymbol{R} - \boldsymbol{C}$ with the axes vectors $\hat{\boldsymbol{h}}^1$ (to yield x) and $\hat{\boldsymbol{h}}^2$ (to yield y). The coordinates of $\boldsymbol{R}$ are calculated with equation 4.2-10, at which point they are so long they are not worth mentioning (this whole process can be programmed in the exact steps shown above). To finalize,

$$\begin{aligned} x =& (\boldsymbol{R} - \boldsymbol{C}) \cdot \hat{\boldsymbol{h}}^1 \\ y =& (\boldsymbol{R} - \boldsymbol{C}) \cdot \hat{\boldsymbol{h}}^2 \end{aligned} \tag{4.2-26}$$

There are some special cases of these expressions that *are* worth looking at. First, in the case of a perfectly aligned sensor,

$$\begin{aligned} x =& \frac{f}{L-f} \frac{c(L - Z_P) - LX_P}{Z_P} \\ y =& \frac{f}{L-f} \frac{d(L - Z_P) - LY_P}{Z_P} \end{aligned} \tag{4.2-27}$$

Note that the relationship between x and X_P is linear, as is that of y and Y_P, but they are not linear with respect to Z_P. This is expected, as the sensors are coplanar with the XY plane. More appropriately put, the X and Y directions are perpendicular to the normal of the sensor planes. *This is a really important condition that makes the defocusing arrangement unique.*

The quantity $\frac{f}{L-f}$ is the optical magnification M of the system. As mentioned above, it is the proportion between any two corresponding sides of the similar triangles depicted in figure 4.2-1. For example,

Figure 4.2-4: The view of a flat dewarping target from a sensor that has -27° of tilt, 15° of pan, and 10° of roll. (Design parameters for Ian Camera.)

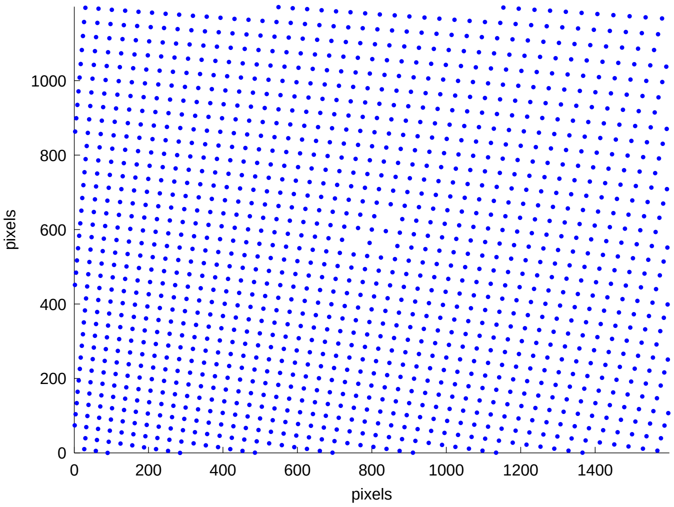
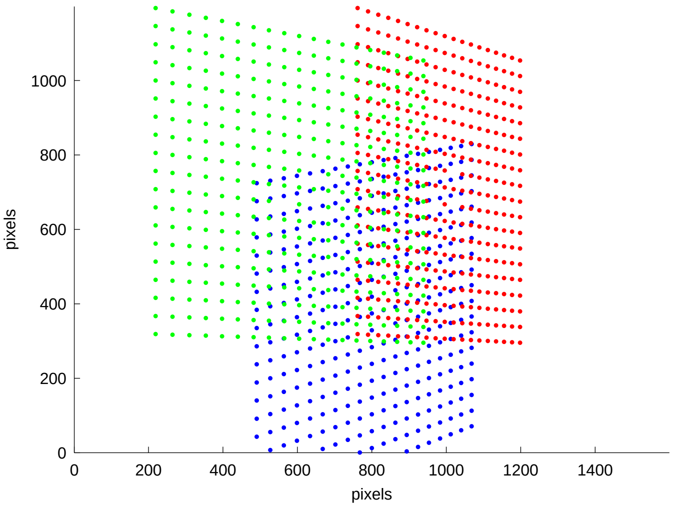


Figure 4.2-5: The view of a flat target at 45° to the camera and centered at $0.8L$ from the aperture plane as seen by the three apertures with perfectly aligned sensors. (Design parameters for Ian Camera.)



$$\frac{l}{L} = \frac{\left(\frac{1}{f} - \frac{1}{L}\right)^{-1}}{L} = \frac{\frac{Lf}{L-f}}{L} = \frac{f}{L-f} = M \tag{4.2-28}$$

A subset of the perfectly aligned condition is that of two apertures on the X axis symmetrically spaced by a distance $2d$ (so that the aperture locations are $(0, d)$ and $(0, -d)$). In this case we have the exact condition under which the equations were originally presented in Pereira and Gharib [2002] (with the exception that he called the *total* separation d)[3]. Setting $c = 0$, we arrive at their results:

$$\begin{aligned} x &= -M\frac{LX_P}{Z_P} \\ y &= \quad M\frac{d(L - Z_P) - LY_P}{Z_P} \end{aligned} \tag{4.2-29}$$

Another interesting case is that for which we allow linear misalignment in X and Y. After some simplification, equations 4.2-27 become

$$\begin{aligned} x =& M\frac{c(L - Z_P) - LX_P}{Z_P} - \Delta X \\ y =& M\frac{d(L - Z_P) - LY_P}{Z_P} - \Delta Y \end{aligned} \tag{4.2-30}$$

which is expected—the linear misalignment should affect nothing but the final location of the particle image. They can thus be measured (most likely through calibration) by imaging a single known point and removed to recover a perfect alignment mapping.

If we now also add the possibility of a linear misalignment in Z,

$$\begin{aligned} x =& M\frac{c(L - Z_P) - LX_P}{Z_P} - \Delta X + \Delta Z\frac{X_P - c}{Z_P} \\ y =& M\frac{d(L - Z_P) - LY_P}{Z_P} - \Delta Y + \Delta Z\frac{Y_P - d}{Z_P} \end{aligned} \tag{4.2-31}$$

Now let's go back to a case where the Z alignment is perfect, but instead include a rotation by the angle δ about the Z axis, still maintaining all the sensors on the image plane. Then we arrive at:

$$\begin{aligned} x =& \left(M\tfrac{c(L-Z_P)-LX_P}{Z_P} - \Delta X\right)\cos\delta - \left(M\tfrac{d(L-Z_P)-LY_P}{Z_P} - \Delta Y\right)\sin\delta \\ &= x_{\Delta X,\Delta Y}\cos\delta - y_{\Delta X,\Delta Y}\sin\delta \\ y =& \left(M\tfrac{c(L-Z_P)-LX_P}{Z_P} - \Delta X\right)\sin\delta + \left(M\tfrac{d(L-Z_P)-LY_P}{Z_P} - \Delta Y\right)\cos\delta \\ &= x_{\Delta X,\Delta Y}\sin\delta + y_{\Delta X,\Delta Y}\cos\delta \end{aligned} \tag{4.2-32}$$

where $x_{\Delta X,\Delta Y}$ and $y_{\Delta X,\Delta Y}$ are the x, y coordinates as defined by equation 4.2-30.

[3] Do not confuse the use of d to equate to previous results and d as the general Y coordinate of an aperture in the derivations in this document.

This is pure rotation about the Z axis, as expected. This type of rotation is easily measurable using a single calibration plane with a minimum of two points.

Now if all misalignments are zero except for γ we have

$$\begin{aligned} x =& M\frac{c(L-Z_P)-LX_P}{Z_P\cos\gamma+(X_P-c)\sin\gamma} \\ y =& M\frac{(d(L-Z_P)-LY_P)\cos\gamma+(cY_P-dX_P)\sin\gamma}{Z_P\cos\gamma+(X_P-c)\sin\gamma} \end{aligned} \tag{4.2-33}$$

Nonzero values for β and γ can be interpreted as changing the magnification locally on a sensor—that is, the magnification becomes a function of x and/or y. A sensor with a different ΔZ as the others experiences the same problem—its magnification differs from that of the other sensors (though in this case it is constant throughout the sensor).

One of the most noticeable types of image distortion in lenses, especially wide angle lenses, is barrel distortion. Mathematically it can be defined by

$$\Delta r = Q_B r^3 \tag{4.2-34}$$

where r is the distance from a particular pixel to the origin of the distortion (so that if the center of the sensor lies on the optical axis of the lens then $r=\sqrt{x^2+y^2}$) and Q_B is a coefficient of distortion. If Q_B is negative, the distortion is called "barrel" because the edges of an imaged rectangle bulge beyond the corners, whereas if it is positive, it is called "pincushion" because the corners extend beyond the edges. By the logic above, then, distortions such as pincushion and barrel are of the same type as misalignments like ΔZ, β, and γ—they are, in essence, local changes in magnification.

Figure 4.2-6: Pincushion distortion.

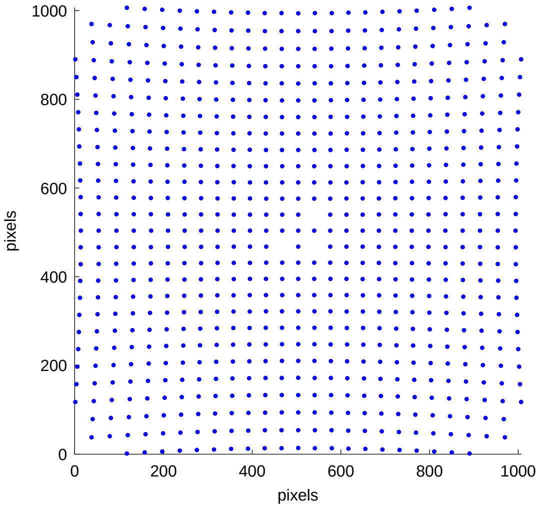


Figure 4.2-7: Barrel distortion.

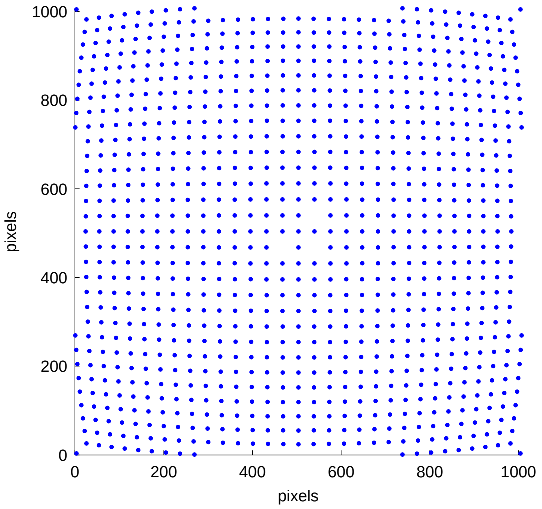
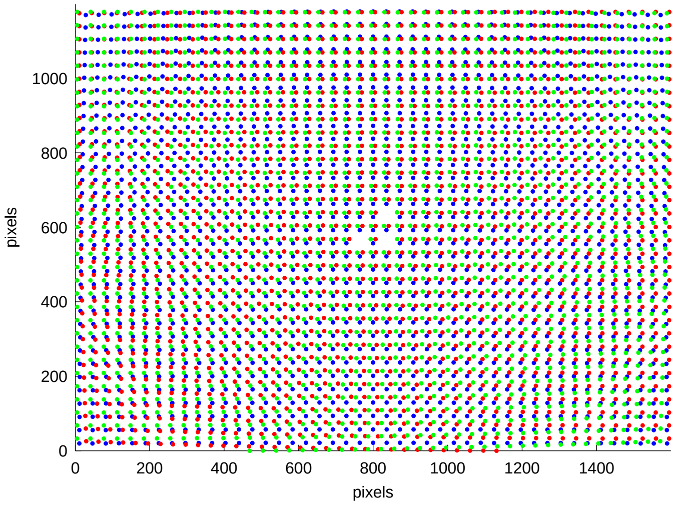


Figure 4.2-8: An example of how lenses with barrel distortion can affect the images. Here Ian's Camera is simulated with barrel distortion approximately equal to the real distortion induced by its 28 mm lenses. The image is that of the dewarping target at the reference plane, which should result in the three sensors' points mapping directly on top of each other. However the triangle arrangement enforces a different amount and orientation of barrel distortion at each sensor and so the points no longer line up.



4.3 Application to Defocusing

The ray tracing model described above allows for all degrees of freedom in the position of the sensors. At that point it becomes tedious to deal with mathematically but is well suited for implementation into a simulator. In this section we will continue to follow the theoretical discussion as presented in Pereira and Gharib [2002] by applying the above model.

The characteristic quantity in DDPIV is the particle image separation—that is, the distance between the image of the same particle on two different sensors for a given particle location $\boldsymbol{P}$. In the past it has been defined by simple subtraction but more generally it can be written as

$$\begin{aligned} b_{ij} &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\ &= M \frac{L - Z_P}{Z_P} \sqrt{(c_i - c_j)^2 + (d_i - d_j)^2} \end{aligned} \tag{4.3-35}$$

In words, this says that the separation is the distance between the apertures rescaled by a quantity involving the Z coordinate of the particle being imaged and the optical parameters of the system. This proves that the multiple aperture-multiple sensor arrangement is equivalent to the multiple aperture-single sensor. The fact that image separation is only a function of the Z coordinate of the particle is what makes defocusing a special case of photogrammetry.

Equation 4.3-35 can be simplified by the introduction of certain quantities. We define the aperture separation s_{ij} as the physical distance between two apertures,

$$s_{ij} = \sqrt{(c_i - c_j)^2 + (d_i - d_j)^2} \tag{4.3-36}$$

To follow Pereira and Gharib [2002], we define the factor K so that

$$K = \frac{1}{M s_{ij} L} \tag{4.3-37}$$

and thus equation 4.3-35 simplifies to

$$b_{ij} = \frac{1}{K} \left(\frac{1}{Z_P} - \frac{1}{L} \right) \tag{4.3-38}$$

as in Pereira and Gharib [2002].

From this single quantity it is possible to calculate the space coordinates of an imaged point. For a given pair of apertures, equation 4.3-38 can be solved for Z_P. Without using equation 4.3-37 (which was included just for consistency), we can arrive at a form of Z_P which shows very clearly how the performance of the camera has a geometrical aspect (in the inclusion of s_{ij}) and an optical aspect (in the ratio b_{ij}/M):

$$Z_{P_{(ij)}} = \frac{L}{1 + \frac{b_{ij}}{M}\frac{1}{s_{ij}}} \tag{4.3-39}$$

Thus we have, mathematically, that the minimum number of apertures to establish the position of an unknown point is two[4].

Z_P in practice is calculated using the average separations $\bar{b}_{ij}$ and $\bar{s}_{ij}$.

$$Z_P = \frac{L}{1 + \frac{\bar{b}_{ij}}{M}\frac{1}{\bar{s}_{ij}}} \tag{4.3-40}$$

Once Z_P is known, the values for X_P and Y_P can be calculated from equation 4.2-27. In practice the values obtained from each aperture are averaged so that

$$\begin{aligned} X_P &= \bar{c}\Big(1 - \frac{Z_P}{L}\Big) - \bar{x}M\frac{Z_P}{L} \\ Y_P &= \bar{d}\Big(1 - \frac{Z_P}{L}\Big) - \bar{y}M\frac{Z_P}{L} \end{aligned} \tag{4.3-41}$$

where the bar indicates arithmetic mean. If the optical axis coincides with the geometric center of the aperture layout pattern, then $\bar{c} = \bar{d} = 0$ by definition of the origin.

The sensitivity of a pair of apertures to the third dimension is defined as the derivative of b with respect to Z_P:

$$\frac{\partial b_{ij}}{\partial Z_P} = -\frac{ML}{Z_P^2}s_{ij} \tag{4.3-42}$$

Two of the three quantities on the right side of equation 4.3-42 depend on the camera design: M and s_{ij}. Recall that

$$M = \frac{l}{L} = \frac{f}{L - f}$$

and that s_{ij} is the separation between two apertures. So sensitivity increases as the focal length of the lenses increase, the distance to the reference plane decreases, and the aperture separation increases.

One can interpret this in an alternative, perhaps more intuitive, way: given a particular L, increasing l in pinhole optics is equivalent to increasing the focal length thus decreasing the field of view of the sensor. What remains then is the ratio s_{ij}/L. In a two-aperture symmetrical layout $\tan(s_{ij}/L)$ is twice the angle between a sensor axis and the optical axis.

So the sensitivity of a camera can be increased by increasing the focal length, bringing the reference plane closer (shrinking the mappable region), or spreading the apertures farther apart,

[4] It should be reiterated, however, that with only two apertures the ambiguity in the particle image matching is large and thus three is considered to be the minimum practical number.

which is exactly what we had concluded.

As a guide in characterizing cameras we can take the aperture-specific part of equation 4.3-42 and call it the *sensitivity coefficient* B:

$$B_{ij} = Ms_{ij} \tag{4.3-43}$$

Note that the sensitivity coefficient could be different for each pair of apertures. For equilateral triangle cameras, then, the average sensitivity coefficient $\bar{B}_{ij}$ is the same as any of the three B_{ij}'s. For non-equilateral triangle cameras, The B_{ij}'s can be maximized by pairing apertures that are farthest apart together.

4.4 Summary

The pinhole model for defocusing cameras is very useful in understanding the defocusing concept and is the model followed by the processing software. As will be exposed in chapter 6, there are many reasons why the pinhole model fails in the practical case.

It is important to note that the sensitivity of a defocusing camera as derived here (equation 4.3-42) does *not* take into account any sensor parameters, thus keeping all else constant the *resolution* of the system will increase as the pixel density increases because the planar resolution (see section 6.2) will increase. The *resolution* of a camera is the final precision of actual measurements. This can be evaluated in many different ways, especially when precision flat targets are available, but can also be estimated by imaging very slow flows and looking at the jitter in the resulting particle tracks.

Chapter 5

Performance Analysis

5.1 Introduction

Before delving into the pitfalls of the pinhole optics approach, we can continue into a detailed performance analysis of the defocusing method using the derivations from chapter 4. As an approximation, errors from Gaussian fitting and dewarping can be lumped together and the camera performance can be analyzed for the first time from the point of view of the actual software algorithms involved.

5.2 Error Analysis

Returning to equation 4.3-35, we can estimate the contribution of pixel error in the algorithm which locates the particle images to the determination of the coordinates of the particle. Without assuming the errors in x and y are independent for a single particle image (in most cases it is not, as will be shown in chapter 12), then for a given aperture pair i, j we have

$$\begin{aligned}\sigma_{b_{ij}}^2 =&\sigma_{x_i}^2\left(\frac{\partial b_{ij}}{\partial x_i}\right)^2+\sigma_{x_j}^2\left(\frac{\partial b_{ij}}{\partial x_j}\right)^2+\sigma_{y_i}^2\left(\frac{\partial b_{ij}}{\partial y_i}\right)^2+\sigma_{y_j}^2\left(\frac{\partial b_{ij}}{\partial y_j}\right)^2\\&+\sigma_{x_iy_i}^2\left(\frac{\partial b_{ij}}{\partial x_i}\frac{\partial b_{ij}}{\partial y_i}\right)+\sigma_{x_jy_j}^2\left(\frac{\partial b_{ij}}{\partial x_j}\frac{\partial b_{ij}}{\partial y_j}\right)\end{aligned} \tag{5.2-1}$$

and if we further assume that all the errors are independent, and that the sensors and lenses are all equal, so that the magnitudes of the errors in x and y in each aperture is the same, we set $\sigma_{x_i} = \sigma_{x_j} = \sigma_x$ and $\sigma_{y_i} = \sigma_{y_j} = \sigma_y$ which reduces the equation to

$$\begin{aligned}\sigma_{b_{ij}}^2 &= 2\sigma_x^2\left(\frac{\partial b_{ij}}{\partial x_i}\right)^2+2\sigma_y^2\left(\frac{\partial b_{ij}}{\partial y_i}\right)^2\\&=2\frac{\sigma_x^2(\Delta x)^2+\sigma_y^2(\Delta y)^2}{(\Delta x)^2+(\Delta y)^2}\end{aligned} \tag{5.2-2}$$

where $\Delta x = x_i - x_j$ and $\Delta y = y_i - y_j$. The error in the separation is then

$$\sigma_{b_{ij}} = \sqrt{2}\frac{\sqrt{\sigma_x^2(\Delta x)^2 + \sigma_y^2(\Delta y)^2}}{b_{ij}} \tag{5.2-3}$$

In fact, equation 5.2-3 is really a middle-ground approximation, because it assumes that the errors in x and y are independent *and* of different magnitude. To arrive at a simple estimate of the error, it would be inconsequential at this point to not throw out the latter assumption and arrive at

$$\sigma_{b_{ij}} = \sqrt{2}\sigma_{\text{img}} \tag{5.2-4}$$

where σ_x and σ_y have been replaced by σ_{img} as an estimate of the Gaussian fitting error magnitude for the sake of simplification.

For a given particle, we can rewrite equation 4.3-40 as

$$Z_P = \frac{LM\bar{s}_{ij}}{M\bar{s}_{ij} + \bar{b}_{ij}} \tag{5.2-5}$$

so that the quantity that depends on software, $\bar{b}_{ij}$, is easy to isolate. According to our assumptions, $\sigma_{b_{ij}}$ depends only on the error of the particle image location σ_{img}, so for a camera with N apertures[1],

$$\sigma_{\bar{b}_{ij}} = \sqrt{\frac{2}{N}}\sigma_{\text{img}} \tag{5.2-6}$$

and thus the error in Z_P is

$$\sigma_{Z_P} = \sqrt{\frac{2}{N}}\frac{LM\bar{s}_{ij}}{\left(\bar{b}_{ij} + M\bar{s}_{ij}\right)^2}\sigma_{\text{img}} \tag{5.2-7}$$

We can simplify this further using the second line in equation 4.3-35 to substitute for $\bar{b}_{ij}$ to arrive at[2]

$$\sigma_{Z_P} = \sqrt{\frac{2}{N}}\frac{Z_P^2}{LM\bar{s}_{ij}}\sigma_{\text{img}} \tag{5.2-8}$$

Now recall equation 4.3-41:

$$\begin{aligned} X_P &= \bar{c}\left(1 - \frac{Z_P}{L}\right) - \bar{x}M\frac{Z_P}{L} \\ Y_P &= \bar{d}\left(1 - \frac{Z_P}{L}\right) - \bar{y}M\frac{Z_P}{L} \end{aligned}$$

Dealing only with X_P for conciseness, we can rewrite the above as

[1] Note that `DDPIV` defines $\bar{b}_{ij} = \frac{1}{N}\left(\sum_{i=2}^{N} b_{i(i-1)} + b_{N,1}\right)$.

[2] Note that we can substitute after taking the derivative, because the partial derivative is in fact the partial derivative *evaluated* at a particular point. The distinction is subtle but important.

$$X_P = \bar{c} - \frac{Z_P}{L}\left(\bar{c} + M\bar{x}\right) \tag{5.2-9}$$

The two quantities that have measurement error are Z_P and $\bar{x}$. Thus we have,

$$\sigma_{X_P}^2 = \left(\frac{\partial X_P}{\partial \bar{x}}\right)^2 \sigma_{\bar{x}}^2 + \left(\frac{\partial X_P}{\partial Z_P}\right)^2 \sigma_{Z_P}^2 \tag{5.2-10}$$

The first derivative equals

$$\left(\frac{\partial X_P}{\partial \bar{x}}\right)^2 = \frac{M^2 Z_P^2}{L^2} \tag{5.2-11}$$

and the second

$$\left(\frac{\partial X_P}{\partial Z_P}\right)^2 = \frac{\left(\bar{c} + M\bar{x}\right)^2}{L^2} \tag{5.2-12}$$

From before (equation 5.2-6), we know that

$$\sigma_{\bar{b}_{ij}} = \sqrt{\frac{2}{N}}\sigma_{\text{img}}$$

and since $\bar{x}$ is just the average of the individual particle image x-coordinates,

$$\sigma_{\bar{x}} = \sqrt{\frac{1}{N}}\sigma_{\text{img}} \tag{5.2-13}$$

To summarize, the errors in the reconstructed position of a given particle is

$$\begin{aligned}
\sigma_{X_P} &= \sigma_{\text{img}} \frac{1}{\sqrt{N}} \frac{Z_P}{L^2} \frac{1}{\bar{B}_{ij}} \sqrt{L^2 M^2 \bar{B}_{ij}^2 + 2\left(\bar{c} + M\bar{x}\right)^2 Z_P^2} \\
\sigma_{Y_P} &= \sigma_{\text{img}} \frac{1}{\sqrt{N}} \frac{Z_P}{L^2} \frac{1}{\bar{B}_{ij}} \sqrt{L^2 M^2 \bar{B}_{ij}^2 + 2\left(\bar{d} + M\bar{y}\right)^2 Z_P^2} \\
\sigma_{Z_P} &= \sigma_{\text{img}} \sqrt{\frac{2}{N}} \frac{Z_P^2}{L\bar{B}_{ij}}
\end{aligned} \tag{5.2-14}$$

Alternatively, we can perform a similar treatment to the one used during the analysis of the Z_P error, we can replace $\bar{x}$ according to equation 4.2-27 to arrive at new expressions for the error in X_P and Y_P that depend only on particle coordinates:

$$\begin{aligned}
\sigma_{X_P} &= \sigma_{\text{img}} \frac{1}{\sqrt{N}} \frac{Z_P}{L^2} \frac{1}{\bar{B}_{ij}} \sqrt{L^2 M^2 \bar{B}_{ij}^2 + 2\left(L M^2 X_P + \bar{c}\left(M^2\left(Z_P - L\right) - Z_P\right)\right)^2} \\
\sigma_{Y_P} &= \sigma_{\text{img}} \frac{1}{\sqrt{N}} \frac{Z_P}{L^2} \frac{1}{\bar{B}_{ij}} \sqrt{L^2 M^2 \bar{B}_{ij}^2 + 2\left(L M^2 Y_P + \bar{d}\left(M^2\left(Z_P - L\right) - Z_P\right)\right)^2}
\end{aligned} \tag{5.2-15}$$

In addition, we can use equation 4.3-42 to see how the errors vary with the camera's sensitivity:

$$\begin{aligned}
\sigma_{X_P} &= -\sigma_{\text{img}} \frac{1}{\sqrt{N}} \left(\frac{\partial \bar{b}_{ij}}{\partial Z_P}\right)^{-1} \frac{1}{Z_P L} \sqrt{L^2 M^2 \bar{B}_{ij}^2 + 2\left(LM^2 X_P + \bar{c}\left(M^2\left(Z_P - L\right) - Z_P\right)\right)^2} \\
\sigma_{Y_P} &= -\sigma_{\text{img}} \frac{1}{\sqrt{N}} \left(\frac{\partial \bar{b}_{ij}}{\partial Z_P}\right)^{-1} \frac{1}{Z_P L} \sqrt{L^2 M^2 \bar{B}_{ij}^2 + 2\left(LM^2 Y_P + \bar{d}\left(M^2\left(Z_P - L\right) - Z_P\right)\right)^2} \qquad (5.2\text{-}16) \\
\sigma_{Z_P} &= -\sigma_{\text{img}} \sqrt{\frac{2}{N}} \left(\frac{\partial \bar{b}_{ij}}{\partial Z_P}\right)^{-1}
\end{aligned}$$

It is interesting to note then that the ratio $\sigma_{Z_P}/\sigma_{X_P}$ does *not* depend on the number of apertures or the sensitivity specifically, but still decreases, all other things constant, as the sensitivity coefficient increases.

Up until this point, the error analysis is more or less equivalent to that of Kajitani and Dabiri [2005], with the primary exception that here we do not assume an axisymmetric, three-aperture layout. If we apply this assumption to the equations in the form of those in equation 5.2-15, we arrive at

$$\begin{aligned}
\sigma_{X_P} &= \sigma_{\text{img}} \frac{1}{\sqrt{3}} \frac{M Z_P}{L} \sqrt{1 + 2\frac{X_P^2}{d^2}} \\
\sigma_{Y_P} &= \sigma_{\text{img}} \frac{1}{\sqrt{3}} \frac{M Z_P}{L} \sqrt{1 + 2\frac{Y_P^2}{d^2}} \qquad (5.2\text{-}17) \\
\sigma_{Z_P} &= \sigma_{\text{img}} \sqrt{\frac{2}{3}} \frac{Z_P^2}{LMd}
\end{aligned}$$

by substituting $\bar{s}_{ij} = d$ and $N = 3$. We arrive at these expressions, which are very similar to those in Kajitani and Dabiri [2005], by an oversimplification of the error in $\bar{b}_{ij}$. Take, for example, the case of a three-aperture camera. `DDPIV` defines the average image separation as

$$\bar{b}_{ij(3)} = \frac{1}{3}\left(b_{12} + b_{23} + b_{31}\right) \qquad (5.2\text{-}18)$$

which, as mentioned before, uses each particle image twice. If we calculate the error in this quantity directly, assuming that the error in the particle image location is of equal magnitude in all apertures and x and y (and that x error and y error are independent) as before, we arrive at

$$\begin{aligned}
\sigma_{\bar{b}_{ij}} = \sigma_{\text{img}} \frac{1}{3} \\
&\left[\left(\frac{\Delta x_{12}}{b_{12}} - \frac{\Delta x_{31}}{b_{31}}\right)^2 + \left(\frac{\Delta x_{23}}{b_{23}} - \frac{\Delta x_{12}}{b_{12}}\right)^2 + \left(\frac{\Delta x_{31}}{b_{31}} - \frac{\Delta x_{23}}{b_{23}}\right)^2 + \right. \qquad (5.2\text{-}19) \\
&\left.\left(\frac{\Delta y_{12}}{b_{12}} - \frac{\Delta y_{31}}{b_{31}}\right)^2 + \left(\frac{\Delta y_{23}}{b_{23}} - \frac{\Delta y_{12}}{b_{12}}\right)^2 + \left(\frac{\Delta y_{31}}{b_{31}} - \frac{\Delta y_{23}}{b_{23}}\right)^2\right]^{\frac{1}{2}}
\end{aligned}$$

where $\Delta x_{ij} = x_i - x_j$ and $\Delta y_{ij} = y_i - y_j$, which, upon replacing x and y according to equation 4.2-27 turns into

$$
\begin{aligned}
\sigma_{\bar{b}_{ij}} = \sigma_{\text{img}} \frac{1}{3} \\
&\left[\frac{\left(s_{12}(c_1 - c_3) + s_{13}(c_1 - c_2)\right)^2}{s_{12}^2 s_{13}^2} + \frac{\left(s_{12}(c_2 - c_3) + s_{23}(c_2 - c_1)\right)^2}{s_{12}^2 s_{23}^2} \right. \\
&+ \frac{\left(s_{31}(c_2 - c_3) + s_{23}(c_1 - c_3)\right)^2}{s_{31}^2 s_{23}^2} + \frac{\left(s_{12}(d_1 - d_3) + s_{31}(d_1 - d_2)\right)^2}{s_{12}^2 s_{31}^2} \\
&\left. + \frac{\left(s_{12}(d_2 - d_3) + s_{23}(d_2 - d_1)\right)^2}{s_{12}^2 s_{23}^2} + \frac{\left(s_{31}(d_2 - d_3) + s_{23}(d_1 - d_3)\right)^2}{s_{31}^2 s_{23}^2} \right]^{\frac{1}{2}}
\end{aligned}
\tag{5.2-20}
$$

This difference arises from the fact that the aperture images are counted twice so it is not possible to derive the error of the average particle image separation from the error on the separation of an individual pair of particle images. As will be seen in chapter 14, the distinction is quite important—the error in the average separation is minimized if the aperture order is such that nearest apertures are connected together rather than in such a way as to maximize the average separation $\bar{s}_{ij}$[3].

5.3 Reconstruction Quality

As discussed above, only two apertures are necessary to calculate the three-dimensional position of a point in space. However, because the matching of particle images during reconstruction is only according to their relative placement, mismatches can easily occur with just two apertures (since then the matching condition is "any two points which lie on a segment parallel to the aperture separation direction"). The *quality* of the reconstruction refers to how many ghost, or mismatched, particles are added to the reconstructed cloud.

Ghosts exist in two types: the first is due to more than one particle images being within a matching tolerance of each other in a single aperture image; the second occurs from random chance that the distribution of particles in the clouds generates false matches within the matching tolerance.

Chapter 14 details a set of simulations that were used to analyze the precision and quality of point cloud reconstruction as a function of a variety of factors. The point clouds used in these simulations were very thin sheets of particles on the XY plane (so that the Z thickness of the particle distribution is below 1 µm). In this way, the aperture images are identical, and one is simply shifted relative to another along the direction connecting the two apertures by an amount corresponding to the sheet's Z-coordinate.

[3] Of course this only matters for cameras with four or more apertures, since four is the minimum number of apertures for which there is a choice in the order of connecting two apertures at at time using each aperture only twice.

5.3.1 Clumping Ghosts

The first type of ghost is the *clumping ghost*, and is extremely important as the number of apertures increases. Given a particle seeding density ρ in particle images per pixels squared[4] of randomly distributed particle images, we can model the probability distribution of n-ghosts with a Poisson distribution of the form

$$P(k,\lambda) = \frac{e^{-\lambda}\lambda^k}{k!} \tag{5.3-21}$$

We will follow exactly the very famous example of Clarke [1946]. First, because our point cloud is essentially a plane in XY, and if we assume that it is completely within the field of view of all the camera's apertures, then we can say that ρ is the same for all apertures and is equal to

$$\rho = \frac{m}{A} \tag{5.3-22}$$

with m denoting the number of particles in the cloud and A the pixel area which they cover at this given Z on the image. Our area of interest is the area enclosed by the matching tolerance δ, defined in `DDPIV` as

$$a = \pi\delta^2 \tag{5.3-23}$$

Thus our image space is divided into M regions with

$$M = \frac{A}{a} = \frac{m}{\rho a} \tag{5.3-24}$$

and our parameter λ is

$$\lambda = \rho a \tag{5.3-25}$$

such that the expected number of regions containing k particle images equals

$$\begin{aligned} MP(k,\lambda) &= \frac{m}{\lambda}\frac{e^{-\lambda}\lambda^k}{k!} \\ &= m\frac{e^{-\lambda}\lambda^{k-1}}{k!} \end{aligned} \tag{5.3-26}$$

Some values of this are shown in table 5.3-1 compared to measurements taken on the point clouds from the simulations[5].

[4] Note that defining the density in pixel space means it depends on the magnification and thus the Z coordinate.

[5] The comparison between calculation and measurement is not completely equivalent. The measurement is that of the number of particles which contain $k-1$ neighbors within the tolerance δ, whereas the calculation is the number of regions of size a which contain k particles. Thus the probabilities in the measured quantities would be normalized by the number of particles whereas the calculated probabilities are normalized by the number of areas, or A/a.

Table 5.3-1 shows how that the Poisson distribution fits the data from the clouds[6] used in the simulations reasonably well, albeit not as nicely as in Clarke [1946].

In our simulated data, there are no lost particle images, and since the point cloud is very thin in Z, it looks exactly the same as projected through each aperture. Thus, in dealing with a camera which has N apertures, k particles within a matching tolerance δ will generate g_c clumping ghosts as defined by

$$g_c = k^N - k \tag{5.3-27}$$

This means that two particle images within δ of each other in an 8-aperture camera will generate 254 ghosts, and 3 particle images will generate a whopping 6,558 ghosts. Estimating with the measured data in table 5.3-1, the simulated 8-aperture camera, at a density of 3.82 particle images per 100 pixels squared will generate $172 \times 254 + 2 \times 6,558 = 56,804$ ghosts. In the simulation, 41,115 were generated.

In real experiments the numbers are not as high as indicated here, since a particle image takes up space on the image plane and thus can become indistinguishable from another if the two are close enough.

5.3.2 Random Ghosts

The second type of ghost occurs much more frequently. First we start our analysis with the simple case of two horizontally opposed apertures, such that the search criteria for matching is simply "any particle image in image 2 that is near (within tolerance of) a horizontal line emanating from a particle image in image 1". From this it should be evident why, in practice, two-aperture cameras are unusable.

First, let us assume that the sheet is at the reference plane[7], so that the images from aperture 1 and 2 will coincide exactly. Given a particle in aperture 1 with a certain x coordinate, all the particles in aperture 2 that match it at a tolerance δ are in a rectangle with dimensions 2δ by $w - x$, so that its area is[8]

$$a(x) = 2\delta\,(w - x) \tag{5.3-28}$$

As before, the particle images are spread randomly over an area A, so that we can expect that within this matching rectangle there are

[6] with ρ at the reference plane

[7] if it is not, then the corresponding shift between images only increases the width of the search area

[8] In reality, the matching rectangle would be one with two semicircular ends with the particle of aperture 1 at the center of the left semicircle.

$$g_{r,pp}(x) = m\frac{a(x)}{A} - 1 = \frac{2m\delta\,(w-x)}{wh} - 1 \tag{5.3-29}$$

ghosts per particle image (at least one particle image must be the real match). We then assume that the particles are evenly distributed in width, so that there are m/w particle images with this x coordinate and thus the total number of ghosts is

$$\begin{aligned} g_{r,2} &= \int_0^w g_{r,pp}(x)\frac{m}{w}dx \\ &= \frac{m}{w}\left[\int_0^w \frac{2m\delta}{hw}(w-x) - 1dx\right] \\ &= \frac{m^2\delta}{h} - m \end{aligned} \tag{5.3-30}$$

For the highest density case, in a two-aperture camera, this estimate yields 68,652 ghosts. The simulation yielded 99,976. The discrepancy is in small part due to the fact that we are only counting random ghosts (whereas the simulation counts both types together) and because we have indirectly assumed that random clumping does not affect the random ghost—the 174 particle images that exist in clumps in fact "share" a search strip, generating even more random ghosts.

Once a third aperture is added to the camera, the matching criterium becomes much more restrictive. Along with the first two particle images lying within some tolerance along some direction, the third particle image must now be along some prescribed direction *and* a specified distance from aperture two so that the distance between particle images is in proportion to the distance between corresponding apertures. This essentially limits the probability of finding the third particle image in the right place to generate a ghost to a mere[9]

$$P_g = \frac{4\delta^2}{wh} \tag{5.3-31}$$

However this probability is only valid within the region A. If we assume for simplicity that the camera has three apertures horizontally arranged (so that the distance between apertures 1 and 2 is the same as that between 2 and 3), then it is easy to see that once the distance between the matching particle images in image 1 and 2 exceeds $w/2$, the probability P_g is 0 because the matching criteria require that the third particle image be at a distance greater than w from the first. Thus only half the prospective ghosts could possibly form complete mismatched triplets, so that the expected number of ghosts in a three-aperture camera becomes is $g_{r,2}/2 \times mP_r$. Using the high-density case as an example again, this yields an estimated 2,952 ghosts. The simulation of a three-aperture, horizontally arranged camera yielded 4,295 total ghosts; again, the extra ghosts are due to clumping effects.

[9] The actual search area in the software is circular; here we use a rectangle to maintain with the previous approximation.

The assumption that our point cloud was a sheet at the reference plane was simply to facilitate the calculation of the strip dimensions (and because it matches the situation in the simulations of chapter 14). In general the widths w and h can be that of the entire image and the arguments should still hold.

We can extend the argument above to an N-aperture camera to arrive at a rough estimate of the number of random ghosts generated at a given seeding density:

$$\begin{aligned} g_{r,N} &= \frac{g_{r,2}}{N-1}\left(mP_g\right)^{N-1} \\ &= \frac{1}{N-1}\left(\frac{m^2\delta}{h}-m\right)\left(m\frac{4\delta^2}{wh}\right)^{N-1} \end{aligned} \tag{5.3-32}$$

5.3.3 Ghosts versus Number of Apertures

Equation 5.3-32 is the estimated number of random ghosts for an N-aperture camera. The estimated number of clump ghosts is

$$\begin{aligned} g_{c,N} &= \sum_{k=1}^{3} g_c MP(k,\lambda) \\ &= m\sum_{k=1}^{3}\left[\left(k^N-k\right)\frac{e^{-m\frac{4\delta^2}{wh}}\left(\frac{4\delta^2}{wh}\right)^{k-1}}{k!}\right] \end{aligned} \tag{5.3-33}$$

assuming that the probability of clumps of four particles is minute. Equations 5.3-32 and 5.3-33 were used to generate figures 5.3-1 and 5.3-2. Evidently three is the optimum number of apertures from the point of view of ghost generation.

5.4 Summary

The analysis of ghost generation presented here should be taken as an order of magnitude estimate only. It is based solely on probability, not taking into account that clumped particle images are hard to come by in images where the particle image actually takes up a finite amount of space (and thus clumped particle images are recorded as indistinguishable blobs) and the point cloud is not a thin sheet. In sheet experiments, most of the time it is also possible to filter a large percentage of the clump ghosts with a simple population density threshold on the obtained particle cloud.

Figure 5.3-1: Estimated number of ghosts for a particle image density of 3.82 particles images per 100 pixels^2 and a matching tolerance of 0.75 pixels as a function of number of apertures.

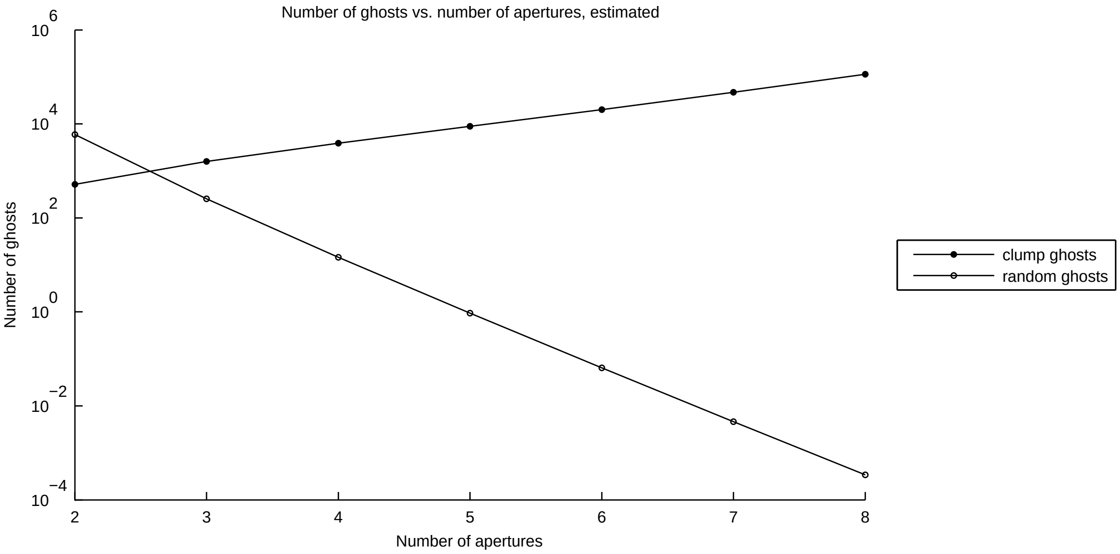


Figure 5.3-2: Estimated proportion of ghosts for a particle image density of 3.82 particles images per 100 pixels^2 and a matching tolerance of 0.75 pixels as a function of number of apertures.

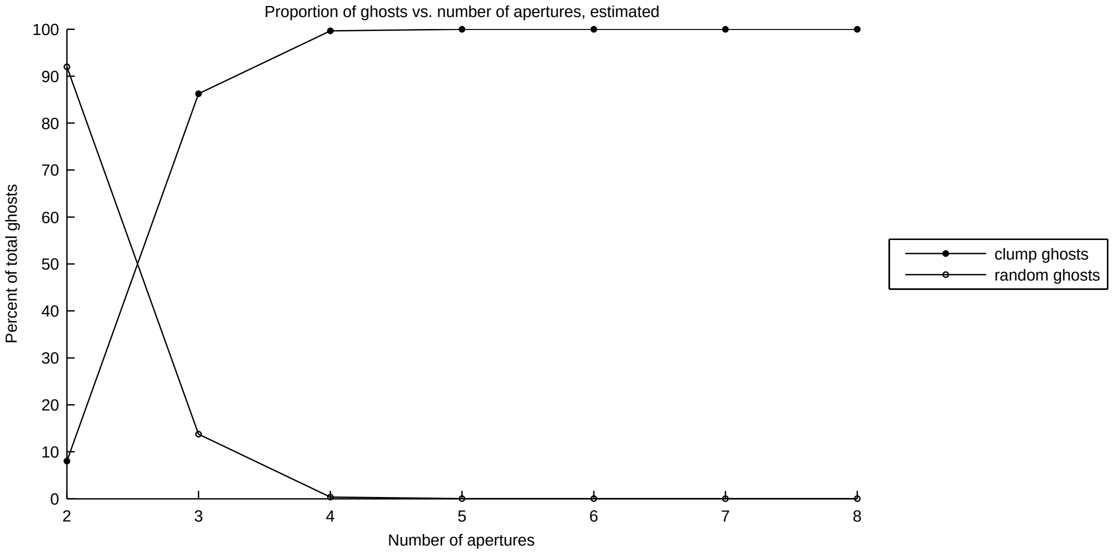


Table 5.3-1: Table of typical values for the Poisson probability of finding k particle images in a small region of the image as described in equation 5.3-26. δ is the matching tolerance which defines $a = \pi\delta^2$. m is chosen so that A is constant.

ρ	m	δ	a	k	$MP(k, \lambda)$	Measured
0.48/100	750	0.50	0.79	1	747	749
0.48/100	750	0.50	0.79	2	1	1
0.48/100	750	0.50	0.79	3	0	0
0.48/100	750	0.75	1.77	1	743	749
0.48/100	750	0.75	1.77	2	3	1
0.48/100	750	0.75	1.77	3	0	0
0.96/100	1500	0.50	0.79	1	1488	1496
0.96/100	1500	0.50	0.79	2	6	4
0.96/100	1500	0.50	0.79	3	0	0
0.96/100	1500	0.75	1.77	1	1474	1493
0.96/100	1500	0.75	1.77	2	12	7
0.96/100	1500	0.75	1.77	3	0	0
1.44/100	2250	0.50	0.79	1	2225	2237
1.44/100	2250	0.50	0.79	2	13	13
1.44/100	2250	0.50	0.79	3	0	0
1.44/100	2250	0.75	1.77	1	2193	2228
1.44/100	2250	0.75	1.77	2	28	22
1.44/100	2250	0.75	1.77	3	0	0
1.91/100	3000	0.50	0.79	1	2955	2977
1.91/100	3000	0.50	0.79	2	22	23
1.91/100	3000	0.50	0.79	3	0	0
1.91/100	3000	0.75	1.77	1	2900	2961
1.91/100	3000	0.75	1.77	2	49	39
1.91/100	3000	0.75	1.77	3	0	0
2.87/100	4500	0.50	0.79	1	4400	4453
2.87/100	4500	0.50	0.79	2	50	47
2.87/100	4500	0.50	0.79	3	0	0
2.87/100	4500	0.75	1.77	1	4277	4404
2.87/100	4500	0.75	1.77	2	108	96
2.87/100	4500	0.75	1.77	3	2	0
3.82/100	6000	0.50	0.79	1	5822	5922
3.82/100	6000	0.50	0.79	2	87	78
3.82/100	6000	0.50	0.79	3	1	0
3.82/100	6000	0.75	1.77	1	5608	5826
3.82/100	6000	0.75	1.77	2	189	172
3.82/100	6000	0.75	1.77	3	4	2

Chapter 6

Deviation from Pinhole Optics

6.1 Introduction

In the derivations of chapter 4, there was no mention of pixel size, sensor resolution, or lenses; it was purely a geometrical analysis. Sensor resolution, lens quality, and other factors play an important role in the performance of a defocusing camera.

6.2 The Particle Image

In a real defocusing camera, the position of particles in space is found by first searching an image for particle images and then searching through the particle images to find the ones that match the pattern of the aperture layout. This requires that the x, y coordinate of the particle images be known to good accuracy because this error propagates directly into the calculated X, Y, Z position and then velocity of the particle. A real sensor is a “discretized” space in that it is made of pixels. They are the smallest image element and nothing smaller than a pixel can be resolved. In determining the sub-pixel position of a particle image, a two-dimensional Gaussian is fitted over the particle images in a least-squares sense; the position of the peak of the resulting Gaussian is taken as the location of the particle image. Thus it is critical that *particle images occupy more than one pixel* on the sensor to have any hope of exacting its position. Of course if the particle images are too big then overcrowding is an issue. Simulations show that the accuracy in determining the sub-pixel location of particle images peaks with particle images whose radius (measured as the distance to the point where the intensity drops to $1/e^2$ of maximum) is 2 pixels[1].

For a given seeding particle size, the two principal factors that affect the size of the image is the “sharpness” of the lens and the sensor planar resolution. For the purposes of this discussion, we define *planar resolution* as the number of pixels per millimeter of field of view at the reference plane. By the definition of magnification, the field of view dimension W at the reference plane of a

[1] See chapter 12 for tests of the algorithm.

sensor of dimension w is

$$M = \frac{w}{W} \Rightarrow W = \frac{w}{M}$$

If s is the dimension of one pixel, then $w = Ns$ where N is the number of pixels along that dimension, and the planar resolution will be

$$R = N/W = N\frac{M}{w} = \frac{M}{s} \tag{6.2-1}$$

The sharpness of the lens is critical because if a particle is treated as a small sphere whose image must cover four pixels in diameter then for a given defocusing camera the minimum particle size would be $4/R$. In the case of the Emilio Camera, this would correspond to just under 400 microns. In actuality, particle images will be larger than this geometric estimate; the lens will blur the image slightly. Once the effects of diffraction from the aperture are considered, given enough light particles much smaller than this prediction can be imaged. In the case of the Emilio Camera, tests indicate that particles about 100 microns in diameter yield good images at f/22 when illuminated by a 200 mJ-per-pulse laser expanded to illuminate the intended probe volume (see figure 3.2-5). In effect then it is best *not* to have an extremely sharp lens as any physical blurring device independent of Z coordinate (such as diffraction) is useful in expanding the particle images over enough pixels so that its sub-pixel position can be measured more accurately.

Typical blurring or soft-focus filters available for photography are inadequate because they are essentially a pseudo-random (or worse—a structured) pattern of micro-prisms used to scatter the rays before they enter the lens. They may work well with incoherent light and large entrance pupils, but when imaging microscopic particles illuminated by a laser at high f-number they make the particle images "jump" around in the image[2] rather than move continuously, thus any advantage gained by physical blurring (which is minimal in the case of these filters) is obliterated by the "increased discretization" of the image domain.

More sophisticated blurring filters exist—those which are meant to be mounted on the aperture plane. The advantage of having the blurring element at the aperture is that this is the only place in a lens where the rays from any point in space pass through the exact same location and thus would be altered equally. These are not cheap, and are very rare for 35-mm-format lenses since the lens would have to be disassembled to install the filter. Rarely lenses are available in "soft focus" versions that implement such filters; realistically these are not an option for DDPIV cameras because they are so rare. There is potential for a custom-designed blurring element to be made for installation together with the replacement aperture (see chapter 9), such as those depicted in Palum [2001].

[2] This is exactly what these filters intend to do: move adjacent point sources of light around on the image so that edges are made to be fuzzy.

These difficulties in imaging and reconstructing the particle image location imply that a tolerance must be used when matching images together. This tolerance is called the *pixel tolerance* and corresponds to how many pixels away from the predicted pattern the particle images are allowed to be. The pixel tolerance must be increased to accommodate for errors resulting from the Gaussian fit (the quality of the particle images) and errors introduced by multi-plane dewarping. However it should also be minimized to eliminate as many ghost particles as possible. With the Ian Camera this tolerance is typically around 0.75 for actual experiments on seeded flows, whereas with the Emilio Camera a tolerance of 0.50 pixels shows good results.

6.3 The Optics

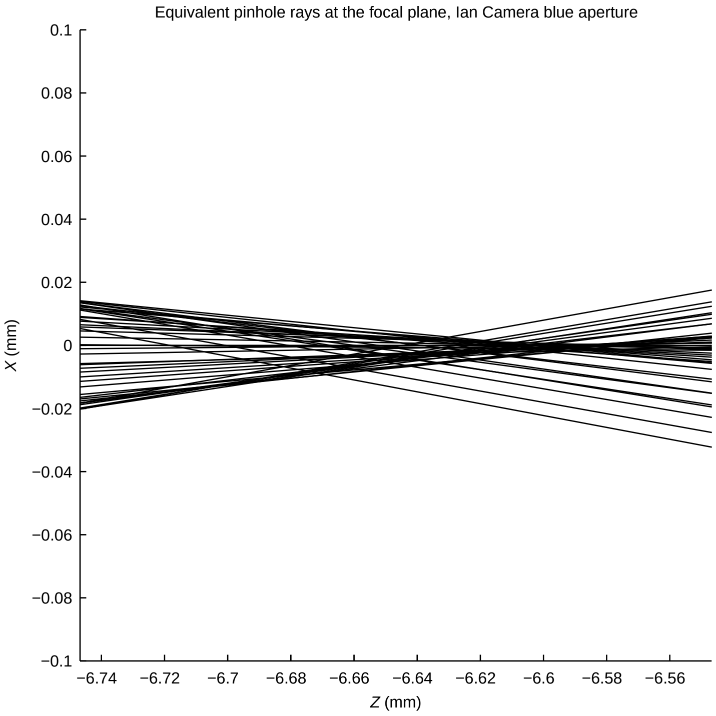
The pinhole optics approximation is useful as a “black box” but diverges from the real case as the lenses grow in complexity. In designing photographic objectives, the aberrations and distortions are minimized by, in short, manipulating the path of the incoming rays. As mentioned before, the aperture is the only place that *all* rays must pass through. As one looks at the ray paths away from the aperture (that is, forwards or backwards within the lens assembly), rays emanating from different field points take different paths. The image of the aperture as seen from the front of the lens is called the *entrance pupil* and it is possible that it changes position and size as the observer moves through the field. In fact, lens designers have been known to purposefully distort the entrance pupil as a function of field position in wide angle lenses to attempt to decrease light fall-off. As the field decreases in width and height this movement is decreased but still existent, so that *there is absolutely no guarantee that a given lens has an equivalent “pinhole” system.* In other words, if an exact ray trace is performed for a given lens and points in space are connected by straight lines to their respective images the lines will not intersect at exactly one point, and, most importantly, the Z coordinate of the intersection region will be a function of the Z coordinates of the points. For a given sensor size, the effect will be amplified as the focal length of the lens decreases and barrel distortion begins to appear (remember this type of distortion can be seen as a local change in magnification which can be viewed as a local change in the location of the pinhole). In other words, a lens that generates measurably perfect images may have an “equivalent pinhole” (within some tolerance) for a given Z coordinate, but the location of this pinhole will move with Z.[3]

Figures 6.3-3 and 6.3-4 show the measured location of the equivalent pinhole as a function of Z for the Ian Camera and the Emilio Camera, respectively. The measurement is done by taking a dewarping set in air, connecting the field points (dewarping target dots) to their images with straight lines, and finding the average intersection[4] in space of these lines for each dewarping plane.

[3] This is essentially because the pinhole optics model ignores the fact that real lenses have distinct entrance and exit pupil planes.

[4] Since the rays are not guaranteed to all intersect with each other, the “intersection” point is in fact the average

Figure 6.3-1: The region of intersection for the "pinhole equivalent" rays emanating from the central row of dots in the dewarping target at the reference plane for the Ian Camera's blue aperture. The rays resemble those of a point source through a lens with spherical aberration because of the barrel distortion. Note that the pattern is asymmetrical because the sensor is not on-axis with the lens.



A sample of these ray bundles is shown in figure 6.3-1 for the Ian Camera and figure 6.3-2 for the Emilio Camera. The barrel distortion propagates itself as a change in the Z coordinate of the intersection of the rays of a single dewarping plane (directly analogous to spherical aberration of the rays from a point source), which is clearly visible in the Ian Camera ray plot (the focal length of the lenses in the Ian Camera is 28 mm). The Emilio Camera, on the other hand, has a much more rectilinear image, both because the lens is longer focal length and because the sensor-lens offset is not as large as in the Ian Camera. Thus its rays seem to intersect much more neatly at a single point for a single plane.

of the midpoints of the shortest segment connecting any two rays.

Figure 6.3-2: The region of intersection for the "pinhole equivalent" rays emanating from the central row of dots in the dewarping target at the reference plane for the Emilio Camera's blue aperture. The smaller sensor-lens shift and longer focal length contribute to a much "cleaner" intersection bundle.

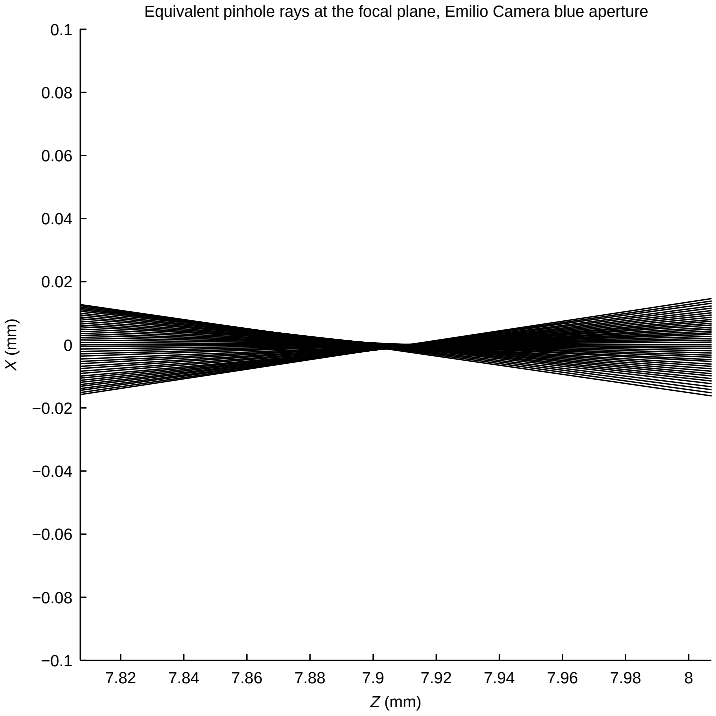


Figure 6.3-3: Measured location of the equivalent pinhole for each aperture of the Ian Camera.

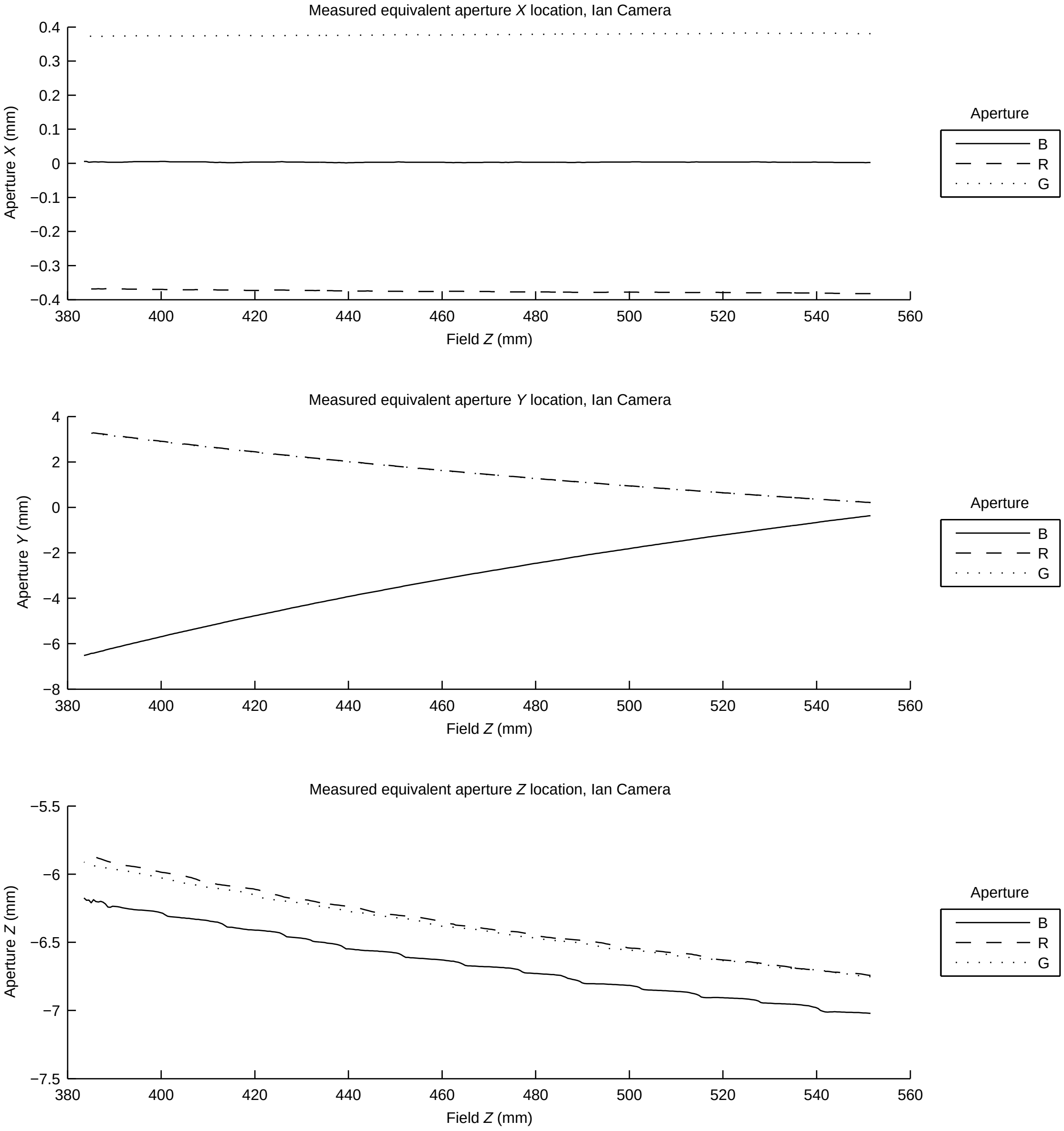
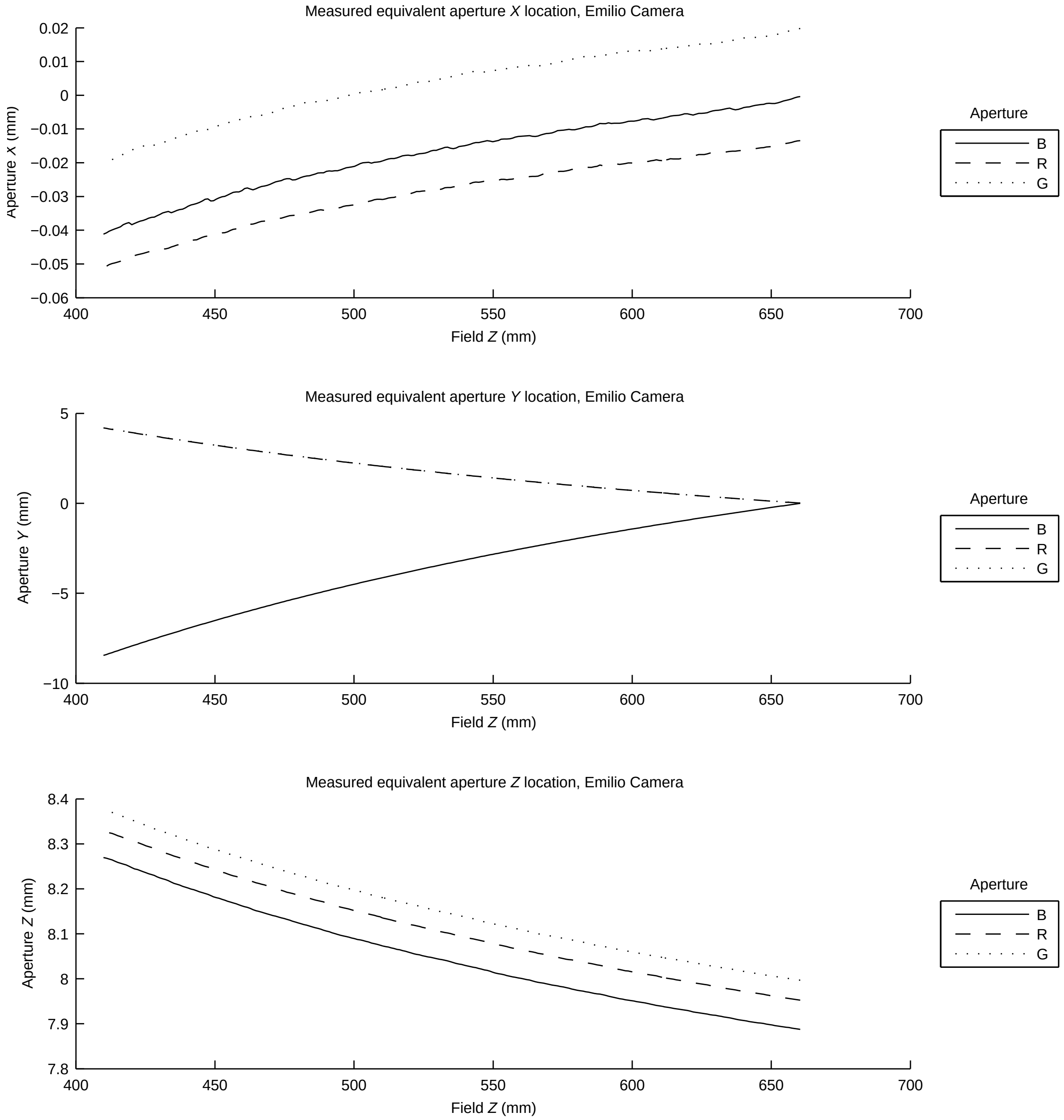


Figure 6.3-4: Measured location of the equivalent pinhole for each aperture of the Emilio Camera. The slight variation in X is due to misalignment of the Z traverse with the optical axis.



Part II

Details of the Instrument

Chapter 7

History of Defocusing Cameras

7.1 Defocusing Camera “Concepts” and “Generations”

Several concepts were explored to some depth during the development of DDPIV. Of these, only two were considered for hardware implementation. “Concept 1” refers to single-lens, multiple-aperture cameras such as the one in Willert and Gharib [1992]. “Concept 5” was formulated in April of 1998 and is the model introduced in chapter 4; it is the arrangement used for all modern defocusing cameras.

Within Concept 5, there are three generations: “first-generation” cameras had straight, simple lenses and alignment stages, “second-generation” cameras had tilted lenses and for the most part relied in some way or another on sensor alignment, and “third-generation” cameras feature photographic objectives and no sensor alignment.

To date, 10 cameras have been built under the Concept 5 model. Two of these were built by Viosense, eight were built at Caltech; five were for use by the Gharib group, two for use the Hornung group, and one went to Dr. Ian Bartol of Old Dominion University. Of these, 1 is first-generation, 7 are second-generation, and two are third-generation. Of the original, one-lens design only two were built with the intention of use for measurement—Concept 1 was quickly abandoned for Concept 5.

7.2 Introduction

All but one of the cameras built to date feature equilateral triangle aperture layouts. This is done so that the sensitivity between all apertures is equal. This is certainly not a requirement.

Traditionally, the aperture on top is referred to as the “blue” aperture, the one on the starboard side (bottom right if looking at the camera from the back) is the “red” aperture, and the port side (bottom left if looking at the camera from the back) is the “green”. The names have been kept for tradition’s sake, but have been referred to as “1”, “2”, and “3” also.

As sensors and sensor interfaces have evolved the signal quality and planar resolution has improved (and cost has gone down). But the choice of lens is perhaps the most influential aspect of final image quality. Initially, for the Silver Camera the lenses were chosen to simply allow for the use of a larger aperture while still "keeping" the pinhole-optics aspect of the setup intact. Because of the offset between the sensor center and the lens axis required for defocusing alignment the image quality was terrible since the center of the lens field was not in use. Subsequent cameras had the lenses tilted so that the lens axis would be closer to the center of the sensors. These had slightly better overall image quality by balancing the aberrations with a tilted focal plane[1]. Inevitably the next step would be to use photographic objectives which would provide far superior image quality at even larger sensor-lens offsets since, for example, 35-mm-format objectives are designed to cover at least a 24 by 36 mm rectangle (the size of a 35-mm-format negative). The factor limiting offsets at this point became the light fall-off experienced as the radial distance from the lens axis grows[2]. The larger field allowed the lenses to be mounted straight-on as in the Silver Camera.

As the image quality increased it became easier to identify one of the most critical problems in building a consistent camera—the heat generated by the sensors. During operation some sensors can get very hot—for example, those based on the KAI-20XX chip can reach temperatures of up to +30°C over ambient. The heat is transferred to the sensor body and to some extent whatever conductor is touching it. The mismatch in thermal expansion of the different materials (for example, aluminum camera body and ceramic CCD package) can cause several pixels of shift in the image. In the older cameras, when dewarping was young and misalignments thought to be fatal, the sensors were mounted on stages, so the heat movement was never noticed because cameras were not often reused without having to adjust the stages somehow and recalibrate. With the Taiwan Camera, which had custom-made lockable stages that could resist several pounds of force (in an attempt to minimize the recalibrations required), the heat movement stuck out like a sore thumb. Tests were then run with every available sensor and camera to try to find out exactly what it was that was moving.

Test showed that every sensor[3] firmly bolted to an optical table, experienced several pixels of shift as they reached a stable operating temperature. The Kodak ES1.0 units, which were remote-head types, were the slowest to stabilize, taking over 90 minutes until no shift was detectable. Others, like the UNIQ's, reached a stable position in a few minutes. The problem was not so much the movement but the hysteresis of the movement. After one power cycle, the new "cold" position was not equal to the previous one, and once the sensor was turned on again, the new "hot" position was not the same as the previous one. Sensors that were held only by the circuit board (so the chip was only supported

[1] Since the f-number is usually high so is the depth of field, so a tilted focal plane does not imply a view-camera style blur.

[2] The light fall-off of lenses may not be apparent with most film cameras because of the nonlinear exposure characteristics of film and because the microlenses commonly found on digital sensors add an extra bit of loss as a function of angle of incidence.

[3] At the time the Kodak ES1.0, UNIQ UP-1030, and Pulnix TM-6300 were tested.

by its own pins) exhibited even more unstable movement. The only setup that had no hysteresis was the one in the Revelle camera where the chips were bolted to small custom-made aluminum frames which were then epoxied onto holders bolted to the faceplate. It was then believed that the source of the hysteresis was stresses that would build between the sensor body and whatever it was bolted to would not allow a continuous expansion. Whether the source was the force exerted by the screws or the mismatch in material between the sensor body and the screws is unclear, but when the Ian Camera was built by bolting (and gluing) the chips directly to the faceplate tests concluded that this arrangement had no hysteresis. The movement still exists thus there is a warm-up time for cameras. The voluminous faceplates of the new cameras make the warm up times longer but also stabilizes the system against local and sudden changes in temperature.

7.3 Camera History—Concept 1

7.3.1 The "Tube Camera"

The first camera of all, put together by Willert, was simply a Texas Instruments tube camera with a three-aperture mask put inside a Navitron 25mm lens. It showed very nicely how the triangles form due to each aperture's image, and in moving the video even the eye starts to interpret the 3-D information provided by the triangles—truly fascinating, and an interesting toy to play with. This is the camera used in Willert and Gharib [1992].

Figure 7.3-1: The lens, with the three-pinhole mask, used by Willert for Willert and Gharib [1992].

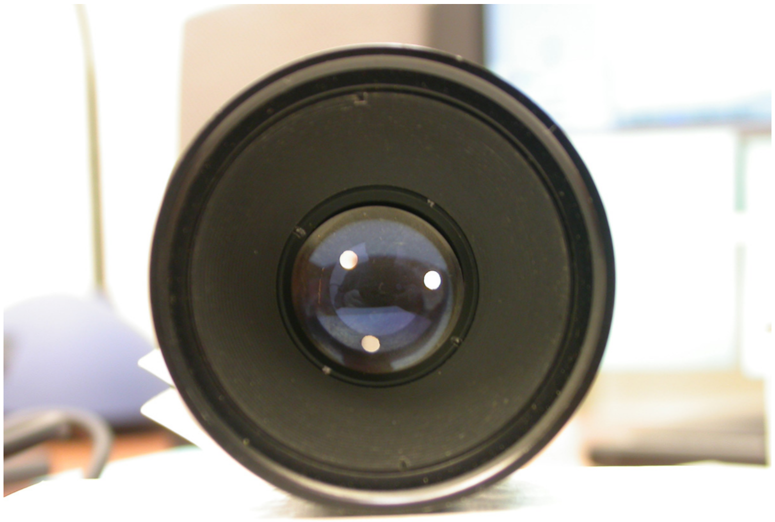
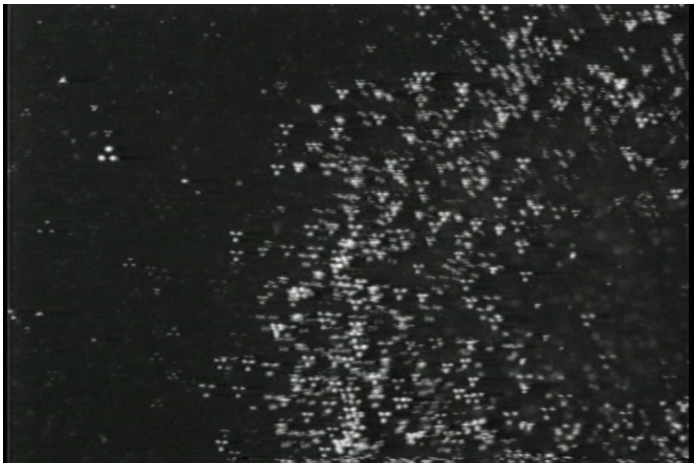


Figure 7.3-2: The first demonstration of defocusing—viewing a vortex ring through Willert's lens from Willert and Gharib [1992].

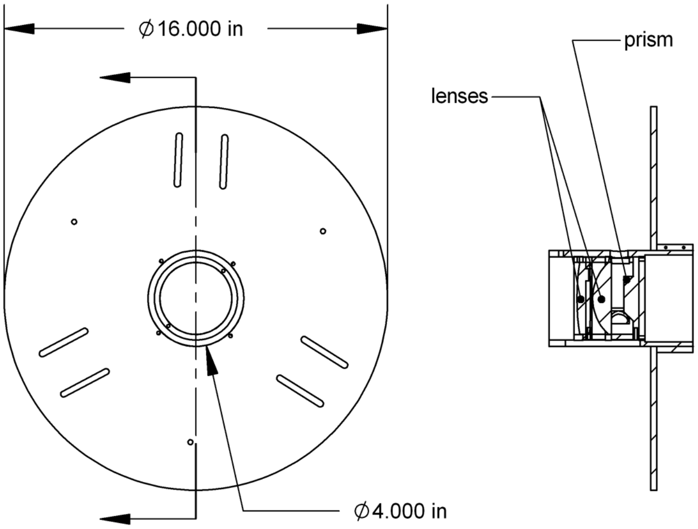


7.3.2 The "Color Splitter"

This is a camera that is featured in the patents—it consisted of a large lens with a color-masked apertures, after which color splitting prisms were used to send each pinhole's image to a separate sensor. With this arrangement, white light was needed for illumination and there was a substantial loss in image intensity from the color splitting. It never produced anything.

7.3.3 The "Prototype Camera"

Figure 7.3-3: The Prototype camera.



This one is also featured in the patents. The design consisted of two large lenses with a three-pinhole mask and mirrors behind these which were used to separate each pinhole's image into a separate sensor. The sensors were mounted radially, facing the camera axis. This camera was used to demonstrate the defocusing concept but proved to be impractical for flow measurement. It looked like a large flying saucer when assembled, and was quite bulky.

7.4 Camera History—Concept 5

7.4.1 Introduction

Concept 5 cameras were the only ones to ever produce data. Table 7.4-1 is a summary of all the cameras built to date and their properties. All the parameters are for the camera in air.

Table 7.4-1: Summary of camera design parameters for Concept 5 cameras. All lengths are in millimeters. For Revelle Camera, B is according to the equilateral triangle with the same base dimension.

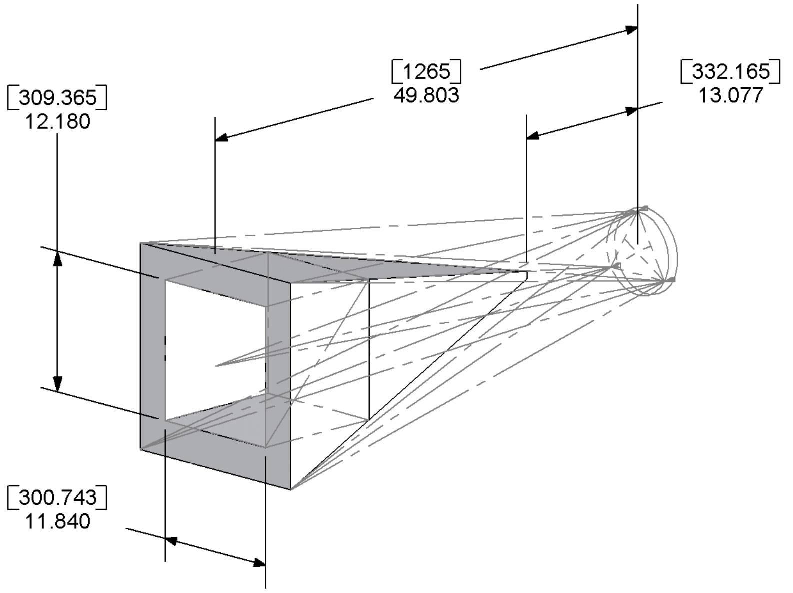
Camera	s_{ij}	L	f	a	$\bar{B}_{ij}$	R
Silver	160.226	1265	25	300	3.230	2.240
Black	73.61	558.28	25	100	3.450	6.997
Kumar	108.25	531.24	60	40	13.78	19.00
Ludwieg	72.17	624.93	60	50	7.665	15.85
Lunchbox	174.6	4780	20	700	0.734	0.568
Revelle	153.98	778.72	50	80	10.56	10.24
Taiwan	160.22	1058.78	50	140	7.941	6.698
Ian	220	551.5	28	100	11.77	7.228
Emilio	170	640	45	130	12.86	10.22

7.4.2 The “Silver Camera”

Table 7.4-2: Important parameters of the Silver Camera.

Original sensors	Kodak ES 1.0
Probe volume (a)	300 mm
Distance to reference plane (L)	1265 mm
Type of lens	Achromatic doublet with 1 mm pinhole
Focal length of lens (f)	25 mm
f-number of lens	f/25
Aperture triangle side (s_{ij})	160.226 mm
Angle of lens	0°
Sensitivity coefficient (B)	3.23 mm
Planar resolution (R)	2.24 pixels/mm

Figure 7.4-1: 3D pinhole model of the Silver Camera.



This was the first Concept 5 camera. Originally, it featured Kodak ES 1.0 sensors which were positioned using New Focus Picomotor-actuated six-degree-of-freedom stages. This camera was large, and heavy, and suffered from poor image quality. The stages were incredibly expensive and require a driver (which is now obsolete) and were most likely complete overkill for the amount of precision that is needed in alignment of the sensors[4]. They also had a very limited range of motion, so aligning the sensors involved disassembling the sensor brackets and repositioning them on the

[4] At this time the movement due to heat had not been discovered and dewarping had not yet been implemented.

stages various times until they were in a position in which the range of motion of the stages would allow for them to be positioned correctly.

This camera was used to execute the first successful flow experiment: the camera was used to simultaneously perform size measurement and speed of bubbles flowing past a model ship propeller in a water tank (see Pereira et al. [2000a]).

However the camera suffered from heavy distortion toward the edges—most notably in the red and green apertures—and also from severe light fall-off.

In the summer of 2000 David Shaack, optical designer, had consulted on the matter and, in a detailed report, recommended that the lenses be tilted so that their axis is collinear with the sensor axis. The next several cameras employed this recommendation and suffered a lot less aberration and light fall-off.

In 2003 the camera was rehabilitated, replacing the Kodak ES 1.0 sensors with the UNIQ UP 1030's. Another minor modification was the addition of bellows between the sensor and the faceplate since the camera cover was always in the way and thus never mounted. However the UNIQ sensors suffered severe heat movement problems and the camera never produced anything in this form.

The Silver Camera is also known as "C5", after the concept. It is the only first-generation camera.

Figure 7.4-2: A photograph of the Silver Camera in 2003. At this point, it included bellows between the sensors and faceplate, and was equipped with the UNIQ UP-1030's.

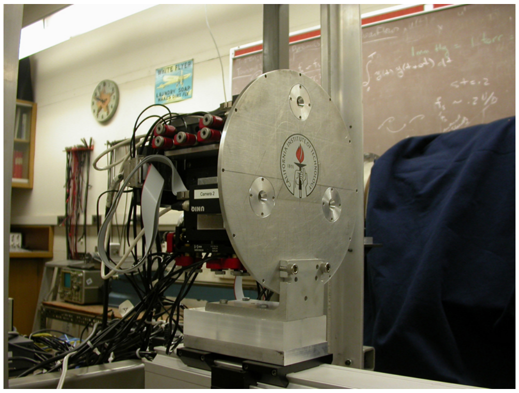
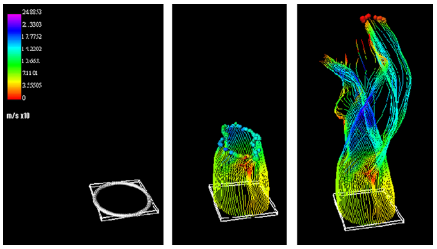


Figure 7.4-3: The most famous of all the DDPIV data sets, from the Silver Camera.

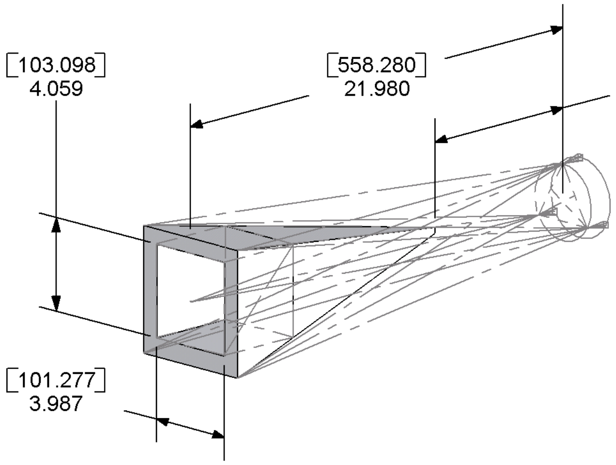


7.4.3 The “Black Camera”

Table 7.4-3: Important parameters of the Black Camera.

Original sensors	UNIQ UP-1030 with double-exposure (PIV-mode) modification
Probe volume (a)	100 mm
Distance to reference plane (L)	558.28 mm
Type of lens	Achromatic doublet with 2 mm pinhole
Focal length of lens (f)	25 mm
f-number of lens	f/12.5
Aperture triangle side (s_{ij})	73.61 mm
Angle of lens	4.35°
Sensitivity coefficient ($\bar{B}_{ij}$)	3.45 mm
Planar resolution (R)	7.00 pixels/mm

Figure 7.4-4: 3D pinhole model of the Black Camera.

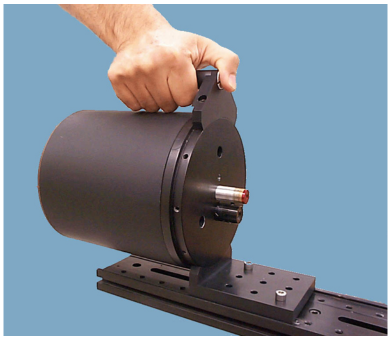


The Black camera was a very small camera originally designed for use underwater inside a tow tank facility. The camera was to fit inside of a torpedo-shaped enclosure and perform measurements of bubble size and velocity near the hull of a model ship. Due to the size constraints, it is very tightly packed—the blue sensor has no degrees of freedom, whereas the other two could only be adjusted in Z using small 40×80 mm Opto-Sigma stages. The idea was that by relying on machining precision and flat head screws combined with counterbores, the yaw and pitch angles that are critical in alignment could be fixed and assumed to be correct; by then a rudimentary single-plane dewarping had been implemented. The magnification on the red and green sensors would simply be matched to the top one. This is also the first model to incorporate a laser diode at the optical axis which could be used to align the camera to targets.

Although never used underwater, the image quality is far superior to the Silver Camera's due to the angled-lens design. This camera played an integral role in the development surge of 2005 when multi-plane dewarping and particle tracking were implemented, and provided various demonstrative data sets.

The Black Camera is also known as the "6-inch Camera" though the diameter of its faceplate is 6.25 inches. This marked the beginning of the second-generation cameras, all of which had tilted achromatic doublets for lenses.

Figure 7.4-5: An infamous photograph of the Black Camera.



7.4.4 The "Kumar Camera"

Table 7.4-4: Important parameters of the Kumar Camera.

Original sensors	UNIQ UP-1030 with double-exposure (PIV-mode) modification
Probe volume (a)	40 mm
Distance to reference plane (L)	531.24 mm
Type of lens	Achromatic doublet with 2 mm pinhole
Focal length of lens (f)	60 mm
f-number of lens	f/30
Aperture triangle side (s_{ij})	108.25 mm
Angle of lens	6.17°
Sensitivity coefficient ($\bar{B}_{ij}$)	13.78 mm
Planar resolution (R)	19.00 pixels/mm

Figure 7.4-6: Data set of a vortex ring (near bottom right) from a piston cylinder (visible in the top left) traveling through a bubble plume in a tank full of plastic particles, measured with the Black Camera. Measured particle relative size is shown here by color and size of marker; the plastic particles were consistently smaller than the bubbles, which came from an aquarium air stone. This is an example of all three functions possible with a defocusing camera: solid surface tracking, particle sizing, and velocity measurement (not shown here). This data set was used in Pereira et al. [2006c].

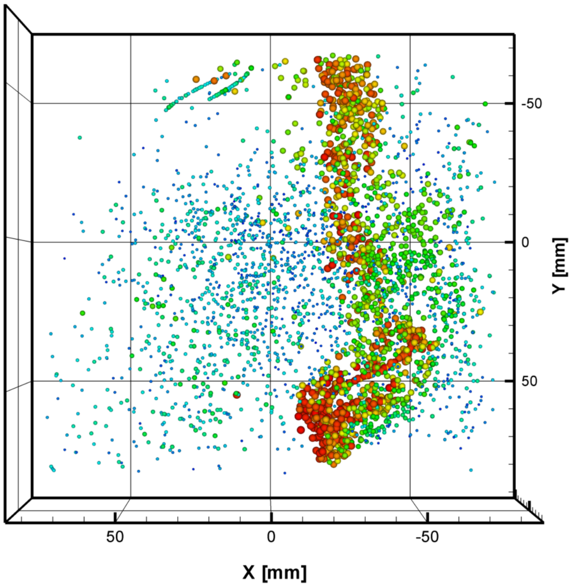
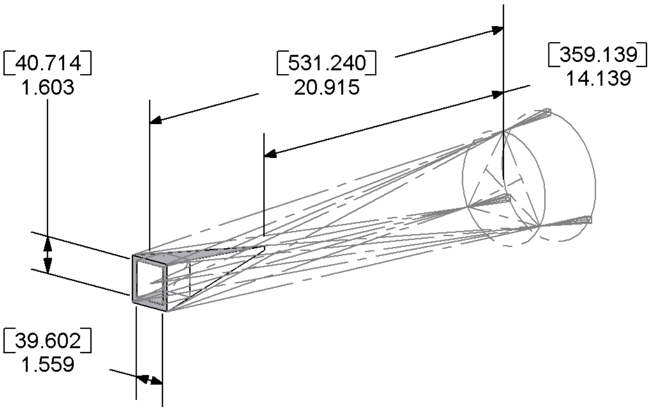


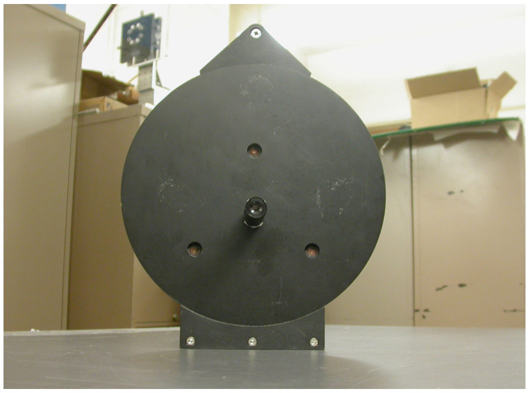
Figure 7.4-7: 3D pinhole model of the Kumar Camera.



This camera was originally built for Kumar Bobba, but was never used by him. It is claimed to have performed one successful experiment, though the results of the velocity measurements are dubious. Even though it has the highest sensitivity coefficient and planar resolution, it is not known to have produced anything substantial.

Originally it was built with a six-degree-of-freedom stack of Opto-Sigma stages (by using one Z stage, one XY stage, and one tip-tilt stage), however, the tip-tilt stage was not lockable and not

Figure 7.4-8: A photograph of Kumar's Camera.



meant to be used with any weight off-center and was soon removed, again to rely on the straightness of the pixel plane on the ceramic package.

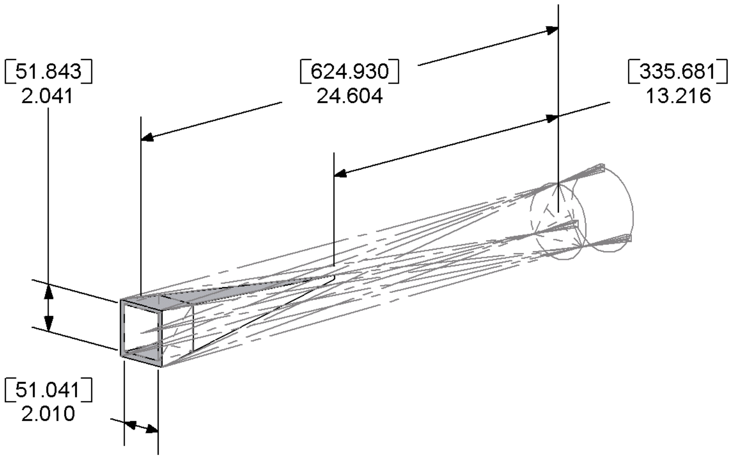
Kumar's Camera is also known as the "David Camera".

7.4.5 The "Ludwieg Camera"

Table 7.4-5: Important parameters of the second Ludwieg Camera.

Original sensors	UNIQ UP-1830 with double-shot (PIV-mode) modification
Probe volume (a)	50 mm
Distance to reference plane (L)	624.93 mm
Type of lens	Achromatic doublet with 2 mm pinhole
Focal length of lens (f)	60 mm
f-number of lens	f/30
Aperture triangle side (s_{ij})	72.17 mm
Angle of lens	3.89°
Sensitivity coefficient ($\bar{B}_{ij}$)	7.67 mm
Planar resolution (R)	15.85 pixels/mm

Figure 7.4-9: 3D pinhole model of the Ludwieg Camera.



This camera was a slight modification of the faceplate of the first Ludwieg Camera. It has produced some preliminary data under the name of "3DPTV" (3-Dimensional Particle Triangulation Velocimetry), namely a map of a doll's face (but only interpolated data is shown, not raw data). A paper describing in more detail what exactly makes it a different system was submitted to Measurement Science and Technology in 2003. It included the same stages as Kumar's Camera and the First Ludwieg Camera.

7.4.6 The "Lunchbox Camera"

Table 7.4-6: Important parameters of the Lunchbox Camera.

Original sensors	Pulnix TM-6300
Probe volume (a)	700 mm
Distance to reference plane (L)	4780 mm
Type of lens	Achromatic doublet with 2 mm pinhole
Focal length of lens (f)	20 mm
f-number of lens	f/10
Aperture triangle side (s_{ij})	174.6 mm
Angle of lenses	1.21°
Sensitivity coefficient ($\bar{B}_{ij}$)	0.73 mm
Planar resolution (R)	0.57 pixels/mm

Figure 7.4-11: A photograph of the Lunchbox Camera. Note that the apertures are upside-down compared to the standard cameras.

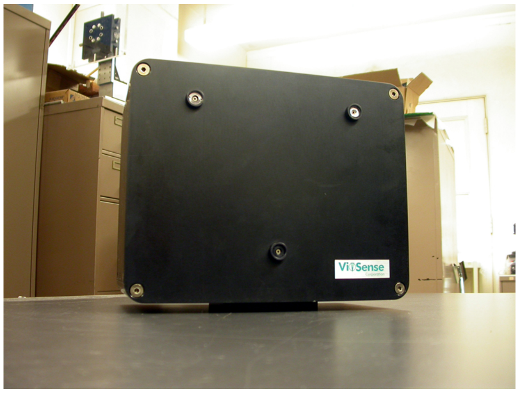
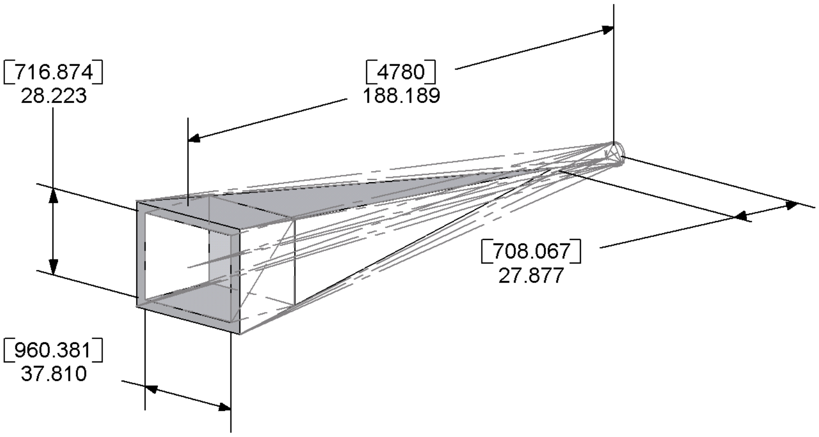


Figure 7.4-10: 3D pinhole model of the Lunchbox Camera.



This was the first camera designed by Pavel Svitek for Viosense. It was also the first camera built with the aperture triangle upside-down, and to date is the one with the largest volume by far. It was built with low-resolution sensors to cut costs and an enormous probe volume to perform demonstrations for use in profilometry of solid objects (it was never intended as a flow-measuring tool). Several successful data sets were taken, but at first, due to the apertures being mounted “upside-down”, the 3D data looked “inside-out”. Regardless several quick demos showed the system’s ability to image solid objects and track the surface of moving fish and a paper flag (both of which had already been done with the Silver Camera years before).

The alignment stages were designed by Svitek, and are lockable, six-degree-of-freedom spring-screw platforms. The alignment was performed by Brad Dooley and has survived several airplane flights in the cargo hold—in fact the casing of the camera has a few dings but the alignment is still functional. It was shown later that this assembly, too, suffered the heat movement problem, but it amounted to less than one pixel due to the low planar resolution of the camera.

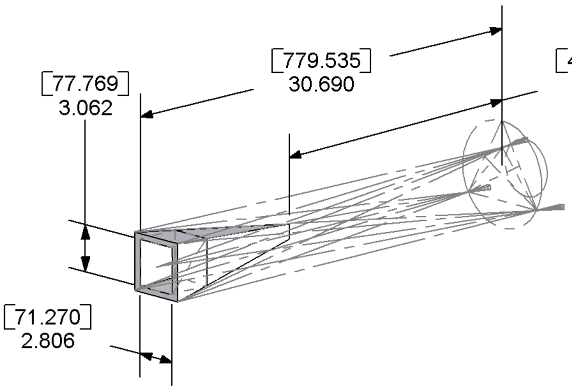
The Lunchbox Camera is also known as the “Low-Res Camera”.

7.4.7 The “Revelle Camera”

Table 7.4-7: Important parameters of the Revelle Camera.

Original sensors	JAI M4+ CL
Most recent sensors	UNIQ UP-1030 with double-exposure (PIV-mode) modification
Probe volume (a)	80 mm
Distance to reference plane (L)	778.72 mm
Type of lens	Achromatic doublet with 2 mm pinhole
Focal length of lens (f)	50 mm
f-number of lens	f/25
Aperture triangle side[5] (s_{ij})	153.98 mm
Angle of lens	6.51°
Sensitivity coefficient ($\bar{B}_{ij}$)	10.56 mm
Planar resolution (R)	10.24 pixels/mm

Figure 7.4-12: 3D pinhole model of the Revelle Camera.



The Revelle Camera was the first designed by Pavel Svitek for Caltech. It was designed for use underwater behind a full-size ship, replacing the original intention of performing measurements on model ships in a tow tank (which is what the Black Camera was designed for). The goal of the experiment was to measure bubble size, population, and velocity.

Due to the size constraints, the camera was not designed as an equilateral triangle but as an isosceles one in which the blue aperture is placed half the distance from the centroid of the equilateral aperture triangle, (so its height is 2/3 that of an equilateral triangle with the same base). The sensors were taken apart and mounted so that the chip and socket were glued to the back of the faceplate after being aligned with external stages and the bodies of the sensors were mounted on the faceplate laterally. The entire assembly had to be water-tight, so there was a lot of care (and epoxy) in the design and assembly of the lens windows and the joint between the faceplate and the "torpedo". Cables were run to the torpedo from the computer (atop the deck of the ship) inside of a large-diameter Tygon tube which connected to the torpedo with a custom sealing flange and hose clamps.

The stages used were the same Opto-Sigma stages used in the above cameras. During alignment for the Revelle experiment (summer of 2003), the tip-tilt stages were used. The alignment was poor and the camera was never proven to work, as the entire mast and light delivery system failed on board the ship after a swell hit the mast while it was being mounted and put a kink in it. Some of the welds on the mast were also prone to leaking.

During preparations for the Athena test (summer of 2004), the JAI sensors were replaced with the UNIQ's from Kumar's camera because as it turned out the JAI's were a pair of "Revision A" and one "Revision B" and it was impossible to synchronize them perfectly. With the UNIQ cameras the alignment was tested and the camera was shown to have a usable probe volume about 100 mm deep. However the Opto-Sigma stages have quite a bit of play, and noise in the sensors forced the use of plastic screws for attaching the sensors to the stages, thus the alignment was not perfect and pixel accuracies below 1 pixel could not be used reliably to identify triangles (using single-plane dewarping). By now the heat movement had already been discovered and thus the alignment and gluing was performed without turning the cameras off. It was discovered that the hysteresis in the sensor power-cycle movement is removed by gluing the sensors as in this camera, but the movement still exists—that is, a calibration is only valid once the sensors temperature has stabilized.

For the Athena test the mast was replaced by custom-built aluminum space frame built by Total Structures, and the setup was mounted sideways, with the camera on the port side and the illumination coming from starboard. The "laser bucket", as the underwater housing for the laser optics was named, had several problems with leaking and the mirror was never mounted correctly so only about half the probe volume was illuminated. The laser, on loan from LaBest, suffered some damage on the first outing as it was left unrestrained on board the ship. As a result, only one laser was firing, and there were several problems with noise in the synchronization signals.

Only three runs of measurements were performed. This data was processed successfully and yielded decent sizing data, though there is potentially quite a bit of noise in the results clouding the population information for the smallest bubbles. Also several interesting phenomena were observed (namely a difference in bubble population size distribution between runs in a bay and runs in the Gulf of Mexico), but the lack of repeated runs made most of the data inconclusive.

Figure 7.4-13: The Revelle Camera (here with dummy plate) assembly just before the second leak test.

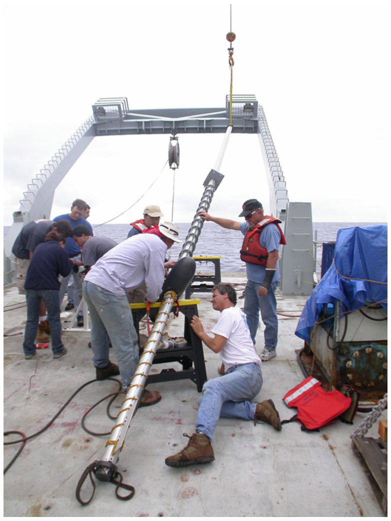


Figure 7.4-14: The Revelle Camera just prior to being mounted on the transom of the R.V. Revelle.

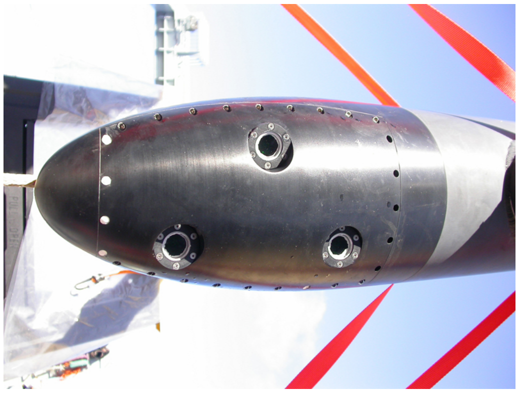


Figure 7.4-15: The Revelle Camera during alignment after the sensors were replaced with UNIQ UP-1030's, seen from above the faceplate.

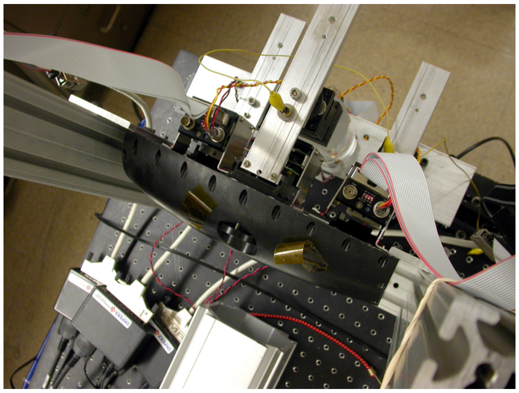


Figure 7.4-16: For the Revelle Camera, the chips were mounted on custom frames (unfinished aluminum) which were glued into receptacles (black-anodized aluminum) on the faceplate. This removed the hysteresis in the heat movement (though the movement had not yet been discovered at this time). Shown here is the blue sensor's frame held in place within the receptacle by stages during alignment (top right) ready to accept the epoxy, which would fill the gap between the frame and receptacle. The red sensor is visible on the top left, out of focus.

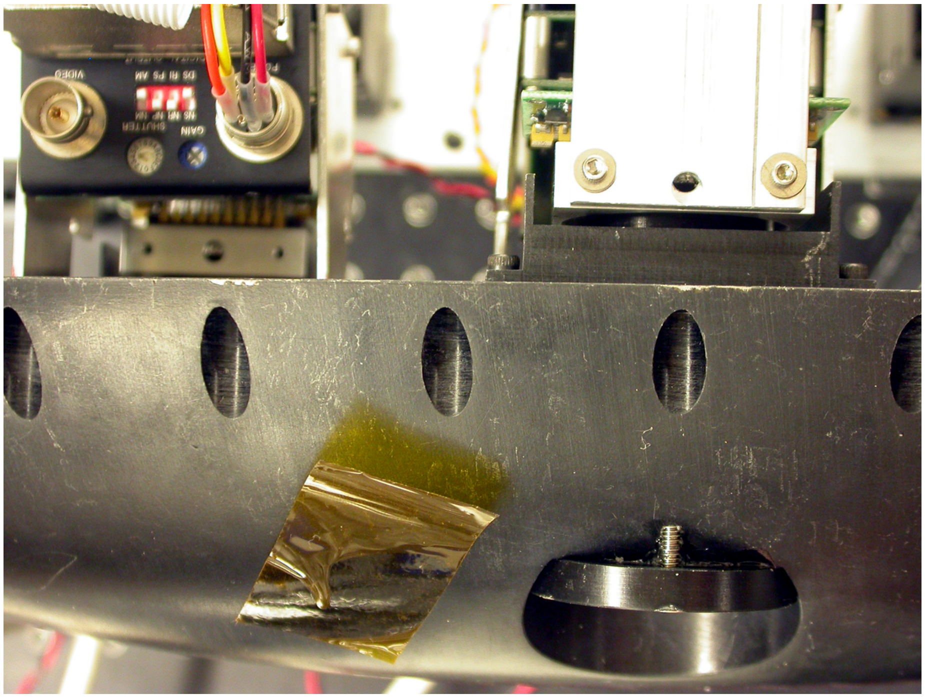


Figure 7.4-17: The Revelle Camera during alignment, showing the faceplate from the back.

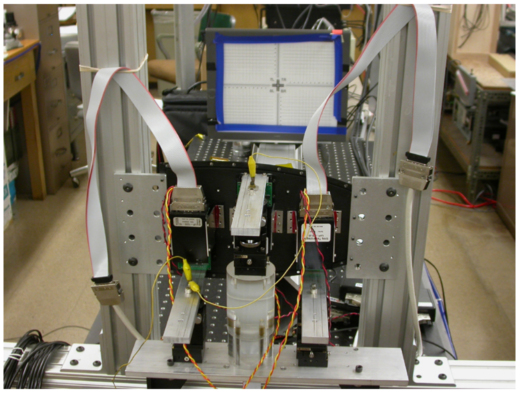


Figure 7.4-18: The space frame, replacing the long mast from the Revelle test, mounted on the transom of the R.V. Athena.

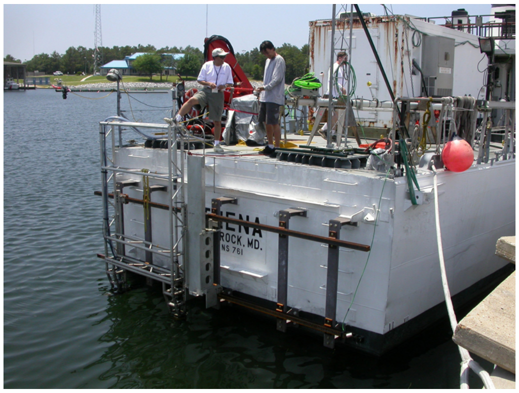
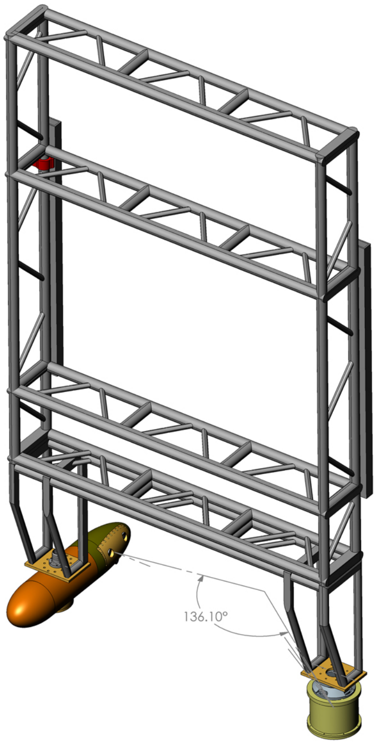


Figure 7.4-19: CAD model of the Revelle Camera assembly for the Athena test showing the forward-scatter angle.

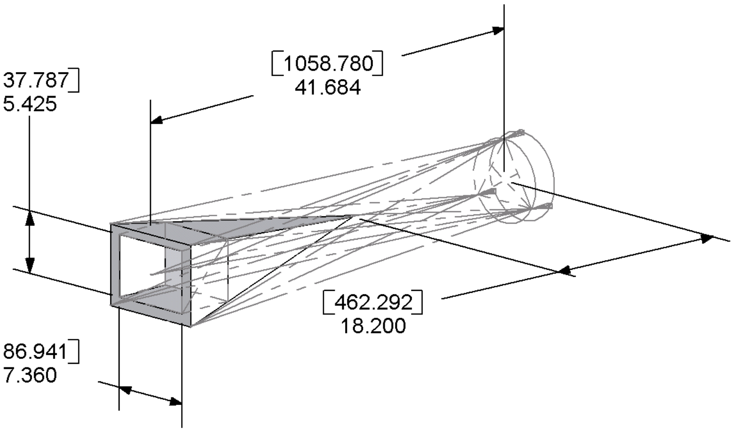


7.4.8 The “Taiwan Camera”

Table 7.4-8: Important parameters of the Taiwan Camera.

Original sensors	JAI M4+CL
Most recent sensors	JAI M2-CL
Probe volume (a)	140 mm
Distance to reference plane (L)	1058.78 mm
Type of lens	Achromatic doublet with 2 mm pinhole
Focal length of lens (f)	50 mm
f-number of lens	f/25
Aperture triangle side (s_{ij})	160.22 mm
Angle of lens	4.99°
Sensitivity coefficient ($\bar{B}_{ij}$)	7.94 mm
Planar resolution (R)	6.70 pixels/mm

Figure 7.4-20: 3D pinhole model of the Taiwan Camera.



This was the first serious attempt at commercializing DDPIV. It was built for a customer with the intention of being a full-on DDPIV system for flow measurement to be delivered in early 2004. The original sensors were replaced after the discovery of the problem with the the Revelle Camera sensors. In this camera, they were replaced by JAI M2-CL, which come from the manufacturer with a “PIV mode” (double-exposure mode).

This camera had potential; it was by far the most precise (highest resolution) camera ever built. The thin faceplate (to save weight) amplified the heat movement problem, and everyone was convinced that this was the source of the movement until it was shown that the movement also existed even in the Revelle Camera (whose faceplate was 30 pounds of solid aluminum). Although

Figure 7.4-21: Data processed from the Taiwan Camera of an image of the flat dewarping target, taken 20 mm from the focal plane. Notice the curve in the transparency—about 1 mm tall—which was found, after inspection of the target, to be caused by sagging of the transparency, which was only held on the lightbox by tape on its edges. The data was processed with 0.5 pixel accuracy and no ghost particles were found.

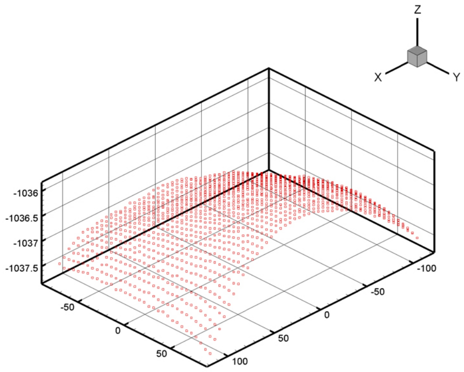
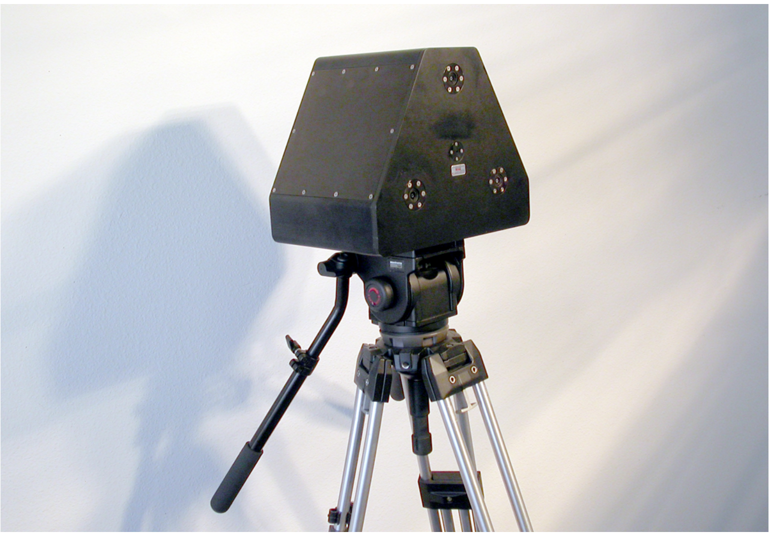


Figure 7.4-22: Photo of the Taiwan Camera.



the phenomenon had been observed before, it is with this camera that it became evident that the source of the movement was indeed the heat from the chips and not a mechanical problem in the stages. It was proven systematically that the movement had severe hysteresis and that it resulted in out-of-plane misalignment. Tests were also done with different numbers of dots on dewarping targets, but the software has changed much since then and the tests are no longer valid.

The camera took several beautiful images of solid targets, and with some of the many alignments that were performed, pixel accuracies down to 0.5 were being used reliably. It featured an improved version of the “Svitek Stages” found in the Lunchbox Camera.

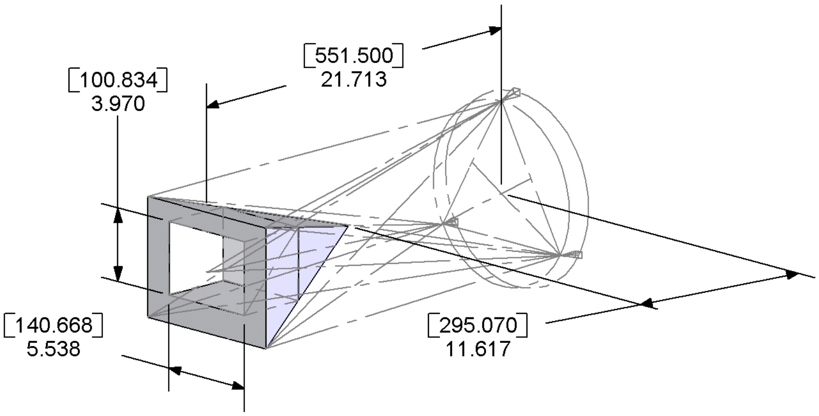
The Taiwan Camera is also known as the “Pitotech Camera” after the company that had ordered it. It no longer exists as it was disassembled to return the majority of the parts for a refund, since the project was abandoned once the heat movement was discovered.

7.4.9 The “Ian Camera”

Table 7.4-9: Important parameters of the Ian Camera.

Original sensors	ImperX 2M30L
Probe volume (a)	100 mm
Distance to reference plane (L)	551.5 mm
Type of lens	Photographic objective (Tokina SL28)
Focal length of lens (f)	28 mm
f-number of lens	f/16
Aperture triangle side (s_{ij})	220 mm
Angle of lens	0°
Sensitivity coefficient ($\bar{B}_{ij}$)	11.77 mm
Planar resolution (R)	7.23 pixels/mm

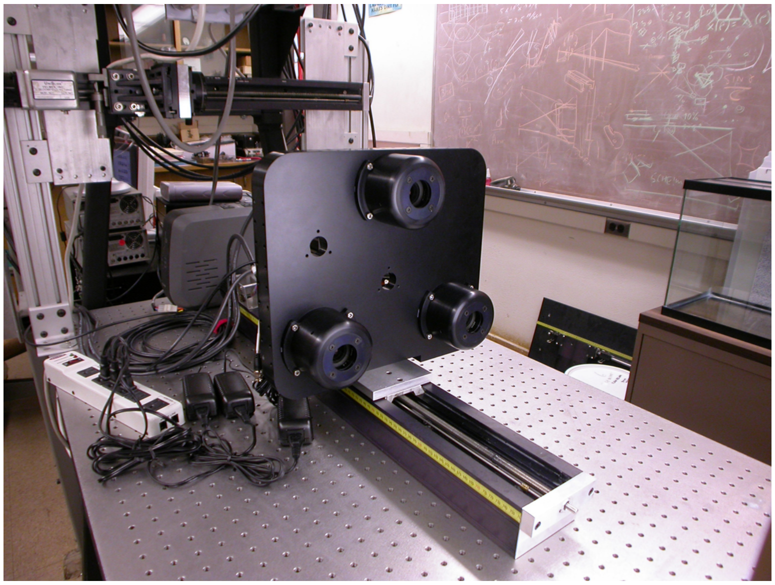
Figure 7.4-23: 3D pinhole model of the Ian Camera.



The Ian Camera provided a push in defocusing camera development. Several years had been spent with the Silver Camera, the Black Camera, and the Revelle Camera, and much had been learned. This camera featured key information engraved into the back plate and two laser diodes used for aiming which would cross more-or-less at the reference plane.

The highest resolution sensors affordable were targeted, but they were physically larger than any other sensor (except the JAI M2-CL, which had the same chip) and image quality looked to be an issue. Up until this time, the lenses were tilted so that the distance between the axis of the lens and the corner of the image was minimized, thus reducing aberrations and light fall-off. Even then, with such large sensors the sensitivity would be limited as even on-axis an achromatic doublet does not provide a satisfactory image at the corners of this chip.

Figure 7.4-24: A photograph of the Ian Camera during assembly.



So 35-mm-format photographic objectives were chosen since they should provide ample coverage when used straight-on for the image size plus the required offset. This camera had by far the largest sensitivity because of this. Its construction was concurrent with the development of multi-plane dewarping, which was essential in the operation of this camera, as discussed in section 6.3.

The sensors were taken apart and assembled directly onto the faceplate so that they could be bolted and glued to the same contiguous piece of aluminum in hopes that this would replicate the hysteresis-less Revelle Camera arrangement (in which the chips were bolted and glued to an aluminum frame which was then glued to the faceplate). The assembly was a success, and moreover proved that the Kodak chips have well-aligned pixel planes.

Dealing with photographic objectives proved a challenge at first as if the camera was to hold its calibration (by solving the heat-movement problem) then the lenses had to be fixed somehow. Originally the lens housings were designed to hold the lenses in space via padded set screws, but that method became questionable during assembly because it would link the lenses to any impact on the outside. The decision was made then to take the lenses completely apart, clean the grease off the focusing mechanism, and glue every moving part together with liquid epoxy.

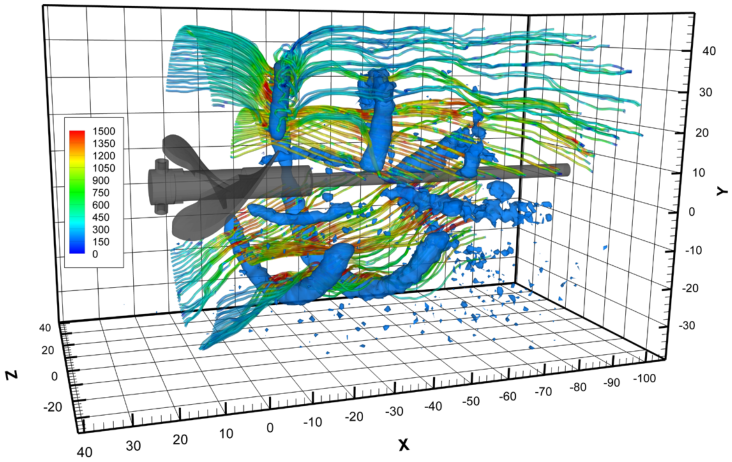
Once the lenses were cleaned, they were reassembled and mounted to the camera. The camera was turned on and left to warm up. The lenses were focused by checking the magnification—not image sharpness—by using the software to measure the sub-pixel distance between four dots on a target at the reference plane. Once they were focused the glue was injected with a syringe into every accessible thread, and it was left to set with the camera on. The assembly was then tested over several power cycles with single-plane dewarping and showed to hold. Impact tests were conducted which the camera failed—this included hitting the camera with a hammer (with considerable force) and dropping it from a four inch height onto a wooden surface. It was thought that the aperture diaphragm was moving, so all visible joints on that were glued but the problem persisted[6].

The final reference plane engraved on the back of the camera for reference was taken as the distance at which the images of the three sensors best matched—so it is slightly different than the design distance.

This camera produced what is by far the largest, most precise, and most impressive data set of all. In homage to the original propeller experiments it was used to map the flow around a propeller in a water tunnel seeded with bubbles. The resulting data set, a phase-average of 200 pairs every 5 degrees of rotation, contains velocity and bubble population information for a half-rotation of the two-blade propeller with data points every 1 mm in each direction.

This camera was the first of the third-generation cameras. Appendix B of Graff [2007a] is a step-by-step report of the assembly of this camera.

Figure 7.4-25: One phase station of the propeller data set taken with the Ian Camera. The blue blobs are population concentrations of bubbles, clearly showing the tip vortex and a second vortex which seems to come off the trailing end of the propeller hub. The colored tubes are instantaneous streamlines of velocity color-coded by speed.



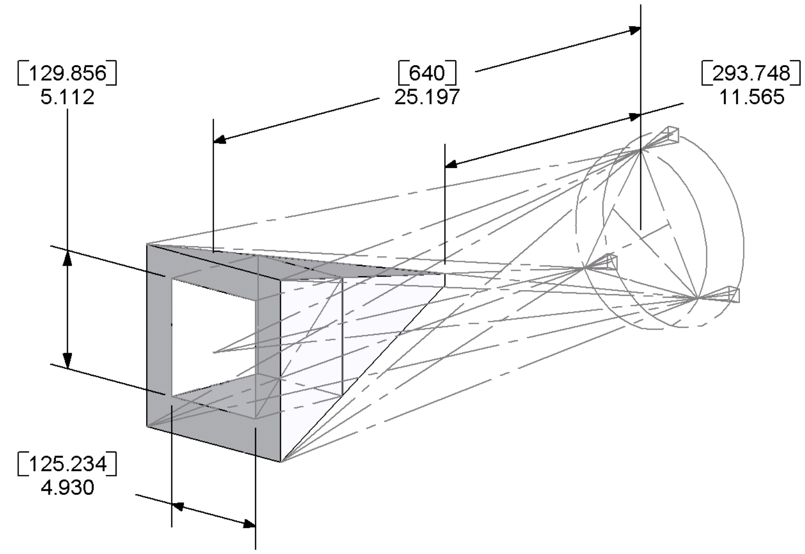
[6] In these lenses it was impossible to remove (or glue) the aperture leaves themselves so it is thought that this is the source of the problem.

7.4.10 The “Emilio Camera”

Table 7.4-10: Important parameters of the Emilio Camera.

Original sensors	ImperX 4M15L
Probe volume (a)	130 mm
Distance to reference plane (L)	640 mm
Type of lens	Photographic objective (Nikkor 45 mm)
Focal length of lens (f)	45 mm
f-number of lens	f/22
Aperture triangle side (s_{ij})	170 mm
Angle of lens	0°
Sensitivity coefficient ($\bar{B}_{ij}$)	12.86 mm
Planar resolution (R)	10.22 pixels/mm

Figure 7.4-26: 3D pinhole model of the Emilio Camera.



The Emilio Camera was built to become the group workhorse after the Ian Camera was delivered. It was the first camera completely designed and assembled by Emilio Graff, from concept, to CAD, to clean room (machining was performed by the GALCIT machine shop). It was clear by now that the mappable region size is restricted more by laser power than anything else, as a result higher sensitivities than the Ian Camera could be achieved in a smaller package with the 4-megapixel sensors.

Design targets included simplifying the mechanical layout even further and making the camera impact-proof. Moreover, the flat field correction feature of the sensors was used to combat the light fall-off.

The aperture leaves were replaced with a precision pinhole (as in the older cameras) reinforced and glued in place (to see if the camera calibration could withstand impact). The camera fares much better than the Ian Camera, but hammer impact still has a measurable effect. The pinhole size was chosen to correspond to f/22 to exploit diffraction as a physical blurring tool, which has allowed the use of a larger variety of seeding particles. The camera is also designed to require minimal effort if it ever needs to be sealed against moisture; the assembly consists of a minimum number of parts and most are sealed.

Appendix A of Graff [2007a] is a summary of the assembly of this camera.

Figure 7.4-27: A rendering of the Emilio Camera CAD model.

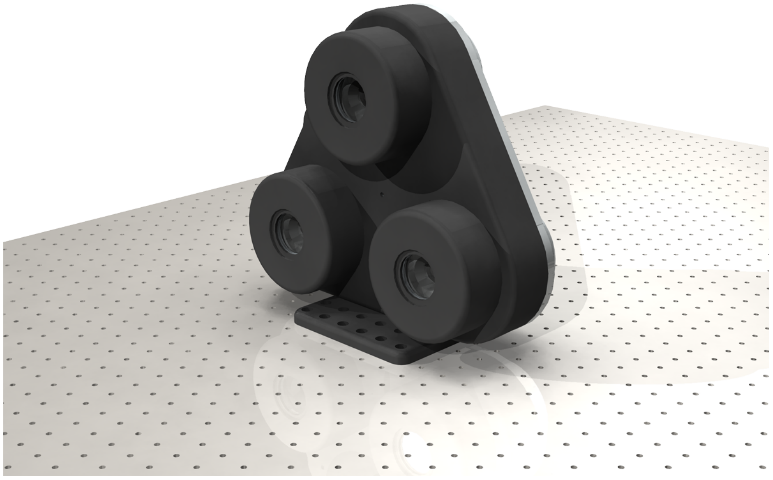


Figure 7.4-28: The Emilio Camera during the flapper experiments.

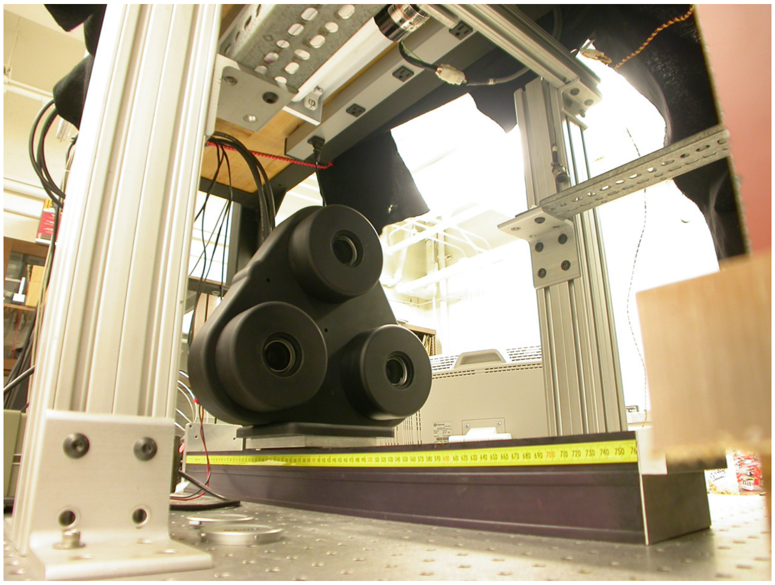
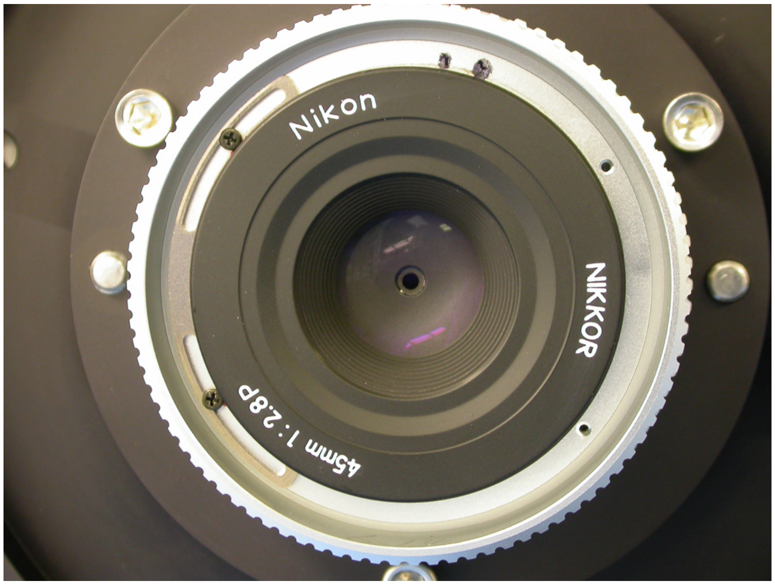


Figure 7.4-29: View of reinforced precision pinhole used in the lenses of the Emilio Camera.



Chapter 8

The Modern Defocusing Camera System

8.1 Introduction

The modern defocusing camera system has four components: the camera, the computer, the calibration setup, and the processing software. The acquisition system is the only part of the system that is commercially available. The rest is the result of in-house design and construction from off-the-shelf components.

8.2 Camera

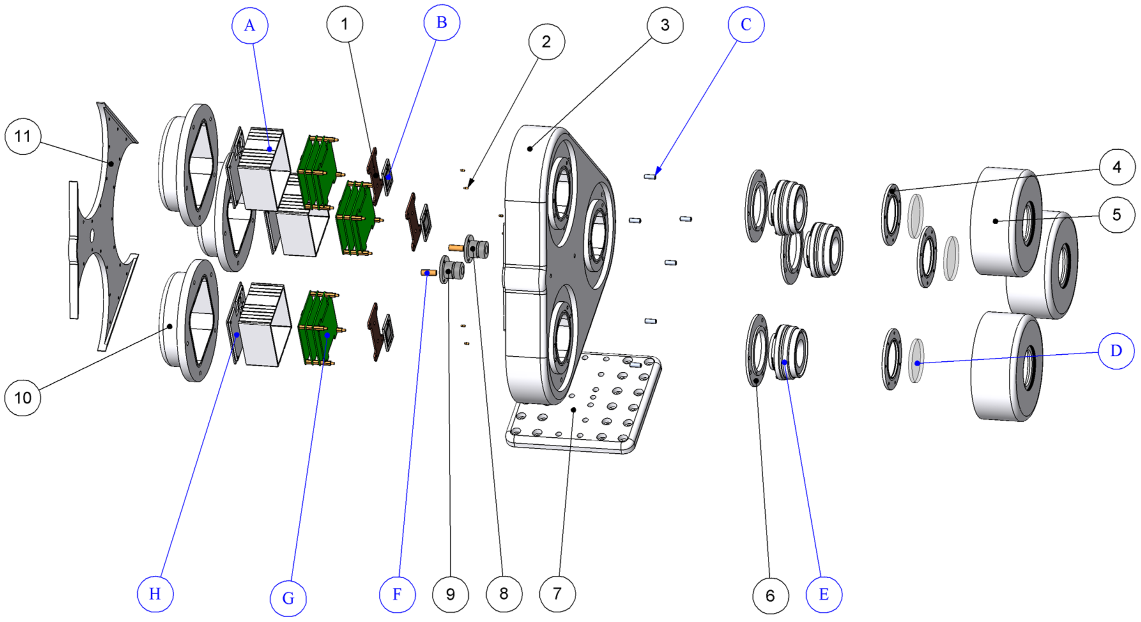
The features of the modern (third-generation) defocusing camera are summarized below.

- High-resolution sensors for increasing camera resolution.
- 35-mm-format lenses to accommodate larger sensor offsets and maximize sensitivity.
- Two aiming diodes—one aligned with the optical axis, and another one on the cone of the lens axes for quick estimation of the location of the reference plane in multi-medium experiments.
- Solid, single-faceplate construction for removing the hysteresis in the heat movement.
- Rigid assembly for minimizing the need to generate multi-plane dewarping sets.

The modern defocusing camera is a simple device. The assembly is centered around a massive faceplate which locates the lenses and sensors relative to each other. The mass in the faceplate is a safety measure, particularly against localized temperature differences which can render a system useless. The key in the layout is that all three sensors are coplanar—this is a defining characteristic of defocusing, as it makes the multi-sensor systems behave as a single-sensor system which in turn simplifies the reconstruction process relative to other systems such as photogrammetry.

Lenses are commercially available, manual-focus 35-mm-format objectives with the aperture diaphragm replaced by a solid pinhole. Sensors are commercially available units, of the "machine vision" type. Figure 8.2-1 shows an exploded view of the Emilio Camera, distinguishing between commercial and custom-made components.

Figure 8.2-1: The Emilio Camera, exploded view. Lettered items were purchased, numbered ones were made in the GALCIT machine shop. Total cost for the hardware of the camera, including machining, was $36,947.93.



8.3 Computer

The most hardware-critical function of the computer is acquisition. Kodak-based sensors normally output at a bandwidth of 60 megabytes per second. Since defocusing cameras have three sensors, the computer must be able to cope with nearly 200 megabytes per second of data.

Although modern computer buses like PCI Express can far exceed this requirement (and so can RAM), recording to hard disk is another story, and machine vision hardware is usually behind the times. Moreover, any data traveling through the bus could be prone to interruptions from the operating system. One extremely reliable solution is the CL160 framegrabber from IO Industries. The card is a 32-bit PCI card, but features an on-board SCSI RAID-0 interface. Recording to disk is not a problem because it is easy to surpass 60 megabytes per second writing speed with just two SCSI drives in a stripe array. Since the interface is on the framegrabber, this data never goes through any computer buses and thus the system is nearly computer independent. The PCI interface is then only used for controlling the card and viewing images stored on the drive arrays.

Coupled with IO Industries' own Video Savant software, three CL160's with four drives connected to each provide unsurpassed reliability and performance. Reliability comes at a price—the acqusition hardware costs a minimum of $4,000 per sensor.

Video Savant is scriptable via `Python` which is a great advantage because it allows for automatization of tedious processes. This has enabled calibration to become fully automated. Automatization also facilitates phase-averaging experiments such as the flapper experiment in chapter 18.

8.4 Multi-plane Dewarping

"Multi-plane dewarping", the name given to the DDPIV method's calibration technique, is intended to adjust for the difference between pinhole optics and real optics as well as correct for aberrations generated by the experimental setup and errors in the manufacturing of the sensors, lenses, and the camera itself. In photogrammetry it is standard for the calibration to calculate directly relevant optical parameters of the system and layout so that it can be replaced by a "black box" in the processing. *Multi-plane dewarping* differs in methodology in that it "forces" the camera into interpreting pixel coordinates correctly, that is, as if it were a true pinhole camera.

As camera hardware and construction evolved, more and more was expected out of them. In the early cameras, with their simple lenses and "one-time use", it was adequate to perform dewarping only at the reference plane and expand the pixel tolerance some to allow for factors that related to the Z coordinate. But in developing them as reliable, precise research tools, the calibration had to become accurate and reusable. To maximize the mappable particle density the pixel tolerance had to shrink and thus the Z discrepancies could no longer be ignored.

The increased precision in the modern defocusing camera construction and hardware have made the experimental facility the weakest link in the optical train. If the optical access to the experimental enclosure is of high quality (optical window), the dewarping set may be generated in a separate tank with similar optical access. Aquarium glass, for example, is guaranteed to require in-situ calibration, even for each location of the camera relative to the tank.

For experiments in air with the third-generation cameras, multi-plane dewarping could be considered a one-time correction for misalignment, optical path, and manufacturing tolerance. However for multi-medium experiment it becomes a much more frequent necessity, since the optical path will change due to the index of the fluid and the tank containing it, and, as mentioned above, the walls of the tank themselves can add aberrations that must be corrected for.

Because of the heat movement, the calibration is also sensitive to tempartture. Most experiments have been done in typical lab environment, with small differences in the ambient temperature. However if the experimental environment is drastically different than the laboratory (for example, if a camera is submerged in frigid water) it may be necessary to compensate for the temperature difference by simulating the experimental environment during calibration.

8.4.1 The Calibration Hardware: Multi-plane Dewarping Rig and Target

The procedure of creating a multi-plane dewarping set is simple and is very similar to that used in standard stereo-PIV systems today. It consists of imaging a precisely-constructed grid of dots at several different Z coordinates throughout the desired mappable region. Because the dewarping set will be used to “force” the camera to behave as expected, it is a bit more constrained than large-scale photogrammetry-style calibrations.

Targets were originally transparencies printed on a laser printer. The Ian Camera proved these to be inadequate and thus the new targets are made by burning off the mirror coating on a standard (second-surface) mirror with the grid pattern at a laser engraving shop. The accuracy of the grid can be increased with more accurate processes such as higher-resolution laser systems, photochemical etching, etc., however, the targets must be large enough to cover the field of view of the camera at the reference plane, which makes the more accurate methods substantially more expensive.

The Emilio Camera then showed that these mirror targets were inadequate for *it*. A photochemically etched target (10 micron precision) was made and the standard deviation of the dewarping functions fell by a factor of 5 for a “hot” camera and up to 12 for a cold camera relative to the old mirror target.

In either case they are illuminated from the back by diffuse light, be it from fluorescent bulbs, a strobe, or a diffused laser, since this facilitates even illumination of all the dots.

The grid of dots is a square grid, the center of which is distinguished by the fact that three of the four diagonal points of the center point are omitted. This pattern is automatically recognized by `gridfind` to be the origin of the grid. The dot spacing is chosen so that up to 2000 dots are visible at the reference plane (since it is easier to skip dots in software rather than have to make a new target). The diameter of the dot is chosen so that at the reference plane they resemble a Gaussian with a $1/e^2$ radius of 2 pixels (see chapter 12).

The software assumes that the target is parallel to the faceplate of the camera and that the Z coordinate of each location where the target is imaged is well known. Other degrees of freedom, such as X, Y, and δ need not be quantified but must remain constant throughout the calibration.

A typical dewarping rig is shown in figure 8.4-3. More information can be found in chapter 3 of Graff and Pereira [2007].

8.4.2 Multi-plane Dewarping Procedure

The multi-plane dewarping procedure is quite simple. The idea is to photograph a precision grid of dots with a known spacing at several planes throughout the probe volume, starting with the reference plane. Three processing steps in the software generate a mapping between the imaged grid and a perfect one, and this mapping is used during the reconstruction of the point clouds in

Figure 8.4-1: The dewarping target, as imaged by the one sensor during an actual dewarping run with the Ian Camera. This is the image at the focal plane so the origin (the dot isolated by the three diagonal spaces) is very close to the center of the image.

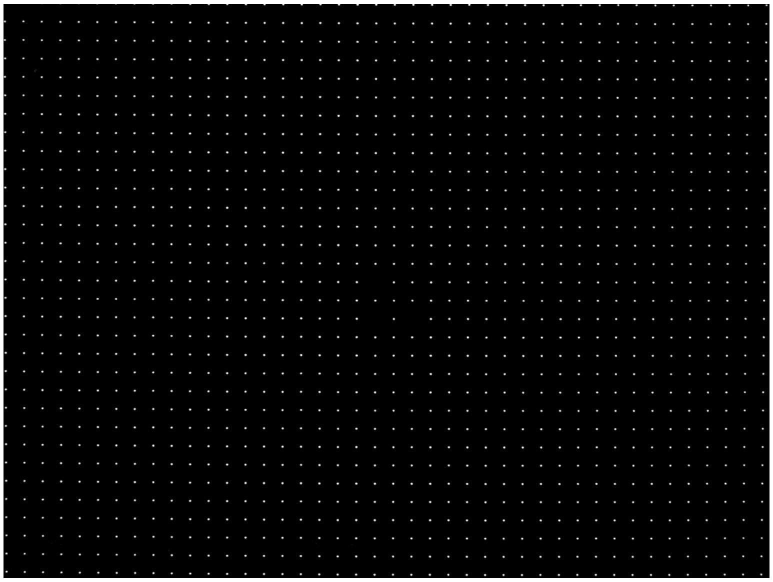
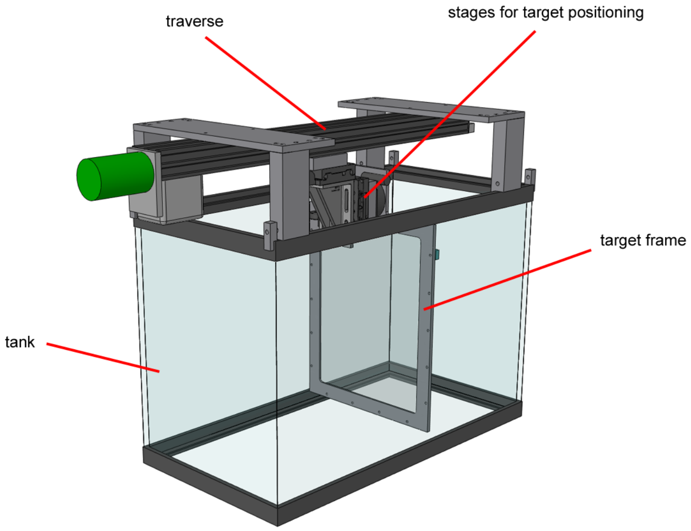


Figure 8.4-2: The dewarping target, as an RGB-composite of all three apertures' images (from the Ian Camera). This is the image at the focal plane so any mismatch between the three apertures is due to lens distortion and manufacturing error.

Figure 8.4-3: Typical multi-plane dewarping setup for calibration in a fluid other than air. The target frame holds the grid target and diffuser. The traverse moves the target through the probe volume; the stages are used to align the target. There are X and Y stages for positioning the origin and a rotation stage for aligning the grid to the horizontal. Alignment of the target face to the camera faceplate is achieved by shimming between the bracket and the traverse.



processing of experimental data.

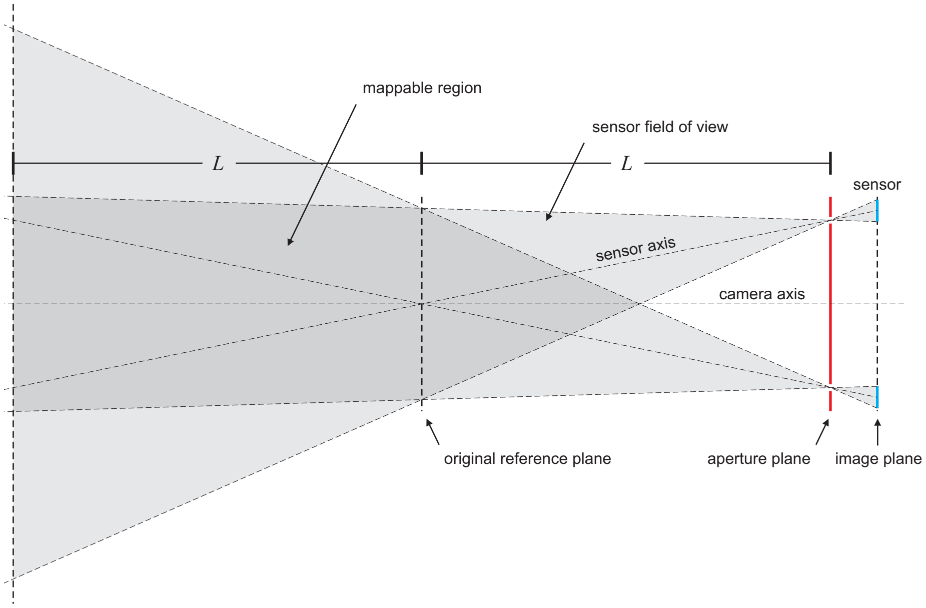
Because the internal pinhole model is forced to fit the images of the dewarping grid, any errors in traversing the grid during calibration are propagated to experimental data directly. Thus it is absolutely key to align the target so that it is parallel to the faceplate of the camera and that the Z traverse be aligned with the optical axis of the camera, and that at each station, the Z coordinate be known to satisfactory precision. Thus the dewarping setup is carefully aligned and components are bolted in a repeatable manner. The frame that holds the target is made to be relatively thick to minimize warping or vibration.

The first step in creating a dewarping set is to turn on the camera and let it warm up. Typically the warm up period takes about 30 minutes since the faceplates for the two third-generation cameras weigh around 25 pounds. The beauty of the phenomenon is that it is easy to check for, since a calibration done at the wrong temperature will yield a poorly reconstructed particle field.

The central laser diode in the third-generation cameras is carefully aligned during assembly to be perpendicular to the faceplate, so by placing a flat mirror at the target and checking for the reflection on the facplate it is easy to precisely align the target parallel to the faceplate. The outer diode is aimed to emanate from the camera's aperture separation so that it traverses toward the optical axis at the same angle as the sensor axis. Because of this, it will be refracted similarly through media as would the light scattered by the seeding particles. The crossing of the two beams provides a quick way to “find” the reference plane in multi-medium experiments.

Note that since multi-plane dewarping forces the parameters to fit reality the reference plane is really an arbitrary location chosen during the design of the camera to maximize the mappable region at a particular working distance and characterize the camera there. However it is possible to simply start the calibration at a different Z location and thus "move" the reference plane, allowing some flexibility in the working distance and/or mappable region size for a particular camera. The geometry of the high-sensitivity cameras, exemplified in figure 8.4-4, makes it impossible to increase the size of the mappable region, and if the working distance is extended it is done so at the expense of resolution and sensitivity.

Figure 8.4-4: Anywhere in space where the fields of view intersect is in theory mappable—including the region behind the reference plane. In this particular layout (the same of figure 4.1-1 but showing the mapping to a distance $2L$ from the aperture plane) the mappable region does not necessarily grow in size. In a "more sensitive" camera, the mappable region may shrink in size as the distance from the reference plane grows, whereas a "less sensitive" camera will exhibit a growing mappable region.



Images should be taken through the desired mapping region at every 5 to 10 millimeters in Z. Too many planes and the error introduced by dewarping may become destructive in the reconstruction; too few and the set may not be able to properly correct for the optical path difference between planes to a satisfactory level.

The grid images are processed in three steps. In the first, `DDPIV` is used to perform Gaussian fitting on the dot images themselves to build a list of the sub-pixel locations of the grid dots as imaged. `gridfind` is then used to "walk" the grid, assigning each dot image a "perfect grid" location. `dewarpC` takes the correspondence list and performs a least squares fit of a second, third, and fourth order function that can generate the perfect grid point locations from the imaged ones, thus "dewarping" the image. Each plane is processed separately—that is, the correction is independent of Z location.

Chapter 9

Defocusing Camera Design

9.1 Introduction

The design of a defocusing camera happens in two parts. First, because characteristics such as the mappable region size and position and the sensitivity of the camera depend on the geometric layout of the apertures, the parameters are finalized by iterating with the pinhole relationships of chapter 4. `ECG Designer` is a Mathematica program written just for that. The second part involves mechanical design to try to approximate the results of `ECG Designer` or adjustments thereof in a real camera. The two can be linked if the CAD software used supports the definition of relations between sketch segments and individual parts.

Detailed information about DDPIV camera design and construction can be found in part III: Design and Construction of Graff [2007a].

9.2 Procedure for Designing a Defocusing Camera

`ECG Designer` is an implementation of the equations of chapter 4 in Mathematica. Based on six parameters, it will compute several factors pertinent to the camera, such as the position and size of the mappable region in air and through two different indices (such as a glass wall and then water), the planar resolution R, necessary sensor offset, etc. By iterating on the six parameters according to the requirements of the desired camera, a satisfactory set can be found and then applied to the CAD model.

The six parameters are:

- “L”—the distance to the reference plane L. Choosing this depends mainly on the physical space for which the camera is intended. If it must be able to map at least half of a given water tunnel, for instance, then the water tunnel wall thickness and index refraction of the fluid can be entered to see how far into the test section the mappable region will extend if the camera is right at the tunnel wall.

- "sij"—the aperture separation s_{ij}. Although the full pinhole relations are programmed in, `ECG Designer` assumes the camera will have an equilateral triangle aperture layout.
- "h"—the height of the sensor in millimeters. Sensor dimensions are usually chosen before design begins based on budget and sometimes features.
- "w"—the width of the sensor in millimeters.
- "pmm"—the value of pixels per millimeter on the sensor, or $1/s$.
- "f"—the focal length of the lens f. This can be a very limiting factor in that it must be commercially available unless custom lenses will be made.

The four secondary parameters to perform calculations for multi-medium experiments are:

- "TankZ"—the distance between the outside tank wall and the aperture plane.
- "Tankt"—the thickness of the tank wall.
- "Tankn"—the index of refraction of the wall material.
- "Watern"—the index of refraction of the fluid inside the tank.

The output of the calculations is extensive. It includes:

- The value of the focal distance l, used when constructing the CAD model.
- The c, d coordinates of the apertures on the faceplate, along with the ratios c/L and d/L, the horizontal and vertical angles of the sensor axes $\arctan(c/L)$ and $\arctan(d/L)$, the necessary X, Y shift of the sensor center relative to the aperture, and the maximum diagonal distance for each sensor (the distance to the corner farthest from the axis of the lens). A diagram is shown depicting a 35-mm-format frame and the circle enclosing it[1] as well as the equivalent circle for a lens with a maximum of 12.4 mm of shift (such as the Nikkor 85 mm shifting lens).
- The closest measurable Z coordinate, equivalent to the tip of the mappable region closest to the camera. In most cases, measurements will never be performed here, because even at high f number rarely does the depth of field allow the dewarping target to be reliably used so close to the camera.
- The field of view at the focal plane. In rare cases, the cost and/or availability of a dewarping target of the necessary size is important.
- The planar resolution R.

[1] Presumably the image quality of a 35-mm-format lens should be acceptable inside a circle circumscribing the 36 mm×24 mm frame size.

- The sensitivity at the reference plane $\partial b/\partial Z|_{Z=L}$.
- The side length a of the characteristic probe volume in air and the fluid.
- The *optical footprint*—the size of the clear area necessary at a given depth to prevent clipping of the mappable region—at the front of the characteristic probe volume in air and water.
- The sensitivity at the front of the characteristic probe volume $db/dZ|_{Z=L-a}$.
- The image separation b at the front of the characteristic probe volume (in pixels).
- The X, Y, Z coordinate of the center of the characteristic probe volume in air and water.
- The Z coordinate of the front face of the characteristic probe volume.
- The optical footprint at the tank wall.
- The X, Y coordinate of the center of the optical footprint.
- The distance to the reference plane in the fluid, measured both from the aperture plane and the inside of the tank wall.
- 3D and 2D diagrams of the fields of view in air and fluid showing the characteristic probe volume.
- Plots of the image separation b vs. Z and the sensitivity $\partial b/\partial Z$ vs. Z for the entire mappable region.

Appendix F in Graff [2007a] contains more detail about `ECG Designer` and the calculations it performs.

It should be noted that many of the geometrical parameters are calculated for both air and a fluid (such as the distance to the reference plane, the size of the probe volume, etc.). Other quantities, such as the sensitivity, are not calcaulted, although they also change. If the fluid index is greater than 1, then the sensitivity goes down because the sensor axes get kinked to an angle closer to horizontal. Another way of looking at it is that the reference plane moves farther away from the camera while the aperture separation remains the same. The change is not calculated simply because it is irrelevant in that the sensitivity is, effectively, a rating parameter used to compare one design to another—the higher the sensitivity, the better potential precision the design will exhibit.

The geometrical parameters are of course estimates with respect to the real result, both because `ECG Designer` is based on pinhole optics and because slight misalignments in the assembly will also affect parameters like the location of the reference plane and the size of the mappable region. But they are the main part of the designer, allowing to quickly iterate on parameters based on the requirements.

In the case of the Ian Camera, for example, Prof. Ian Bartol wanted to use it for two different water tunnels. He requested an a value of 100 mm in water. The first tunnel had a test section 152.4 mm tall and 152.4 mm wide with a Plexiglas wall of thickness 12.7 mm. Maximum length for side access was 1000 mm; for bottom access it was 890 mm. The second tunnel was larger, with a test section width and height equal to 300 mm. Access dimensions were much larger and were not a constraint. Since he wanted to map the entire test section of the large tunnel from one side, one constraint on the camera (mainly on L) was that the mappable region extend to the back wall of the tunnel while the camera sits outside the test section at some distance.

Table 9.2-1: Difference between the measured field of view and the calculated, pinhole-optics field of view for several lenses that were considered for defocusing cameras.

Lens	Measured (mm)	Pinhole (mm)	%
Nikkor 20mm, $L = 800$mm	257	264	-2.76
Nikkor 20mm, $L = 400$mm	122	133	-7.78
Nikkor 28mm Shift, $L = 800$mm	179	185	-3.24
Nikkor 28mm Shift, $L = 400$mm	82	90	-9.03
Nikkor 24-120mm Zoom @ 24mm, $L = 800$mm	220	218	0.74
Nikkor 50mm, $L = 800$mm	100	97	2.53
Tokina 28mm, $L = 800$mm	121	124	-2.60

The basic proportions of the camera and the location of the mappable region are constrained primarily by the flow facilities for which it is intended. The choice of the size of the mappable region, namely a, is a function of the intended seeding particle, light source, and planar resolution R (and of course intended experiment). A smaller volume can be lit more brightly with less power, and most likely can be used with smaller seeding particles. If the seeding particles are bubbles, the resolution R can probably decrease substantially keeping the illumination constant. Tests with the Emilio Camera show that with 100-micron-diameter Kodak fluorescent particles and a 200 mJ per pulse Nd:YAG laser this upper limit is around 120 mm if the laser is collimated into a cylinder, which corresponds to about 8.8 J/m^2.[2]

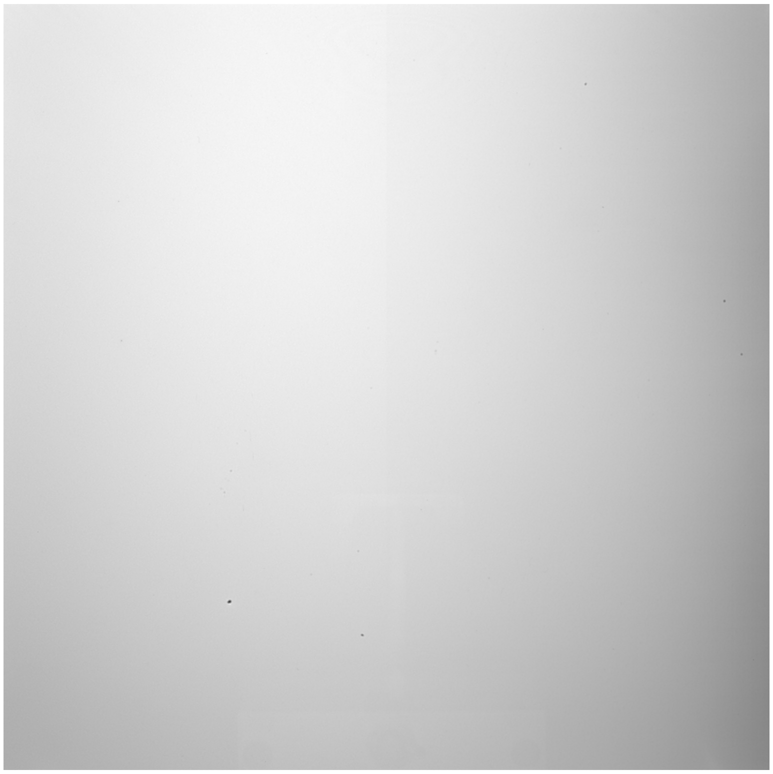
The aperture separation should be maximized. The upper limit is of course when the sensor-lens shift is so high that the light fall-off becomes considerable at the edges of the image. This can be checked quickly if a sensors and a few lenses are available, and should be checked with the final components before finalizing the decision. By putting the sensor on a stage it can be shifted relative to the lens while imaging a target with small dots (such as a dewarping target). Such a setup is shown in figure 9.2-4.

The light fall-off for the Emilio Camera can be considered the upper limit, though for the Ian Camera it was considerably worse (and that camera is, of course, still functional even in the corners). Images of a white light box with four diffusive layers taken by the Emilio Camera is shown in

[2] This is of course a very rough figure, since it depends on the f-number used, the distance to the probe volume, and, most importantly, the seeding particle size and reflectivity.

figures 9.2-1, 9.2-2, and 9.2-3. The slight difference between sensors R and G is accentuated by small differences in the gain.

Figure 9.2-1: Light fall-off in aperture B of the Emilio Camera. The difference between the darkest pixel and the lightest is about 100 (of a possible 255).



However one should also take care to consider the fact that the higher the sensitivity, the higher the angle between the sensor axes and the optical axis is, sometimes making illuminating an experiment in forward scatter a bit challenging (as one of the sensors is bound to be almost out of the forward-scatter regime).

None of the parameters are independent of each other. Exploration is the best way to realize this, but table 9.2-2 describes some of the basic relationships.

The values for “L” and “f” are most often used to control the mappable region size and location given a specific sensor. Although it has important effects, “pmm” is rarely a parameter that can be varied. The value of “sij” is used to maintain the desired balance between the sensitivity and the sensor-lens offset. Alternatively it will also define the possibilities of using a given camera for a mappable region farther away than designed for (as in depicted in figure 8.4-4 and discussed in section 3.5 of Graff [2007a]).

Figure 9.2-2: Light fall-off in aperture R of the Emilio Camera. The difference between the darkest pixel and the lightest is about 140 (of a possible 255).

9.3 Mechanical and Optical Design Considerations

9.3.1 Introduction

The modern (third-generation) defocusing camera comprises relatively few parts:

- A faceplate onto which the sensors are mounted.
- Lens covers (with windows) so that the lenses can be protected from impact and dirt.
- Sensor covers (or a backplate) to protect sensor assemblies from impact.
- A "foot" or base to help stabilize the camera. The center of mass will be within the faceplate since it is by far the most massive part.
- Two laser diodes for aiming the camera. The beam of one should coincide with the optical axis, the other one should form an angle with the second equivalent to the sensor axis angle and they should cross at the reference plane.

Figure 9.2-3: Light fall-off in aperture G of the Emilio Camera. The difference between the darkest pixel and the lightest is about 170 (of a possible 255).

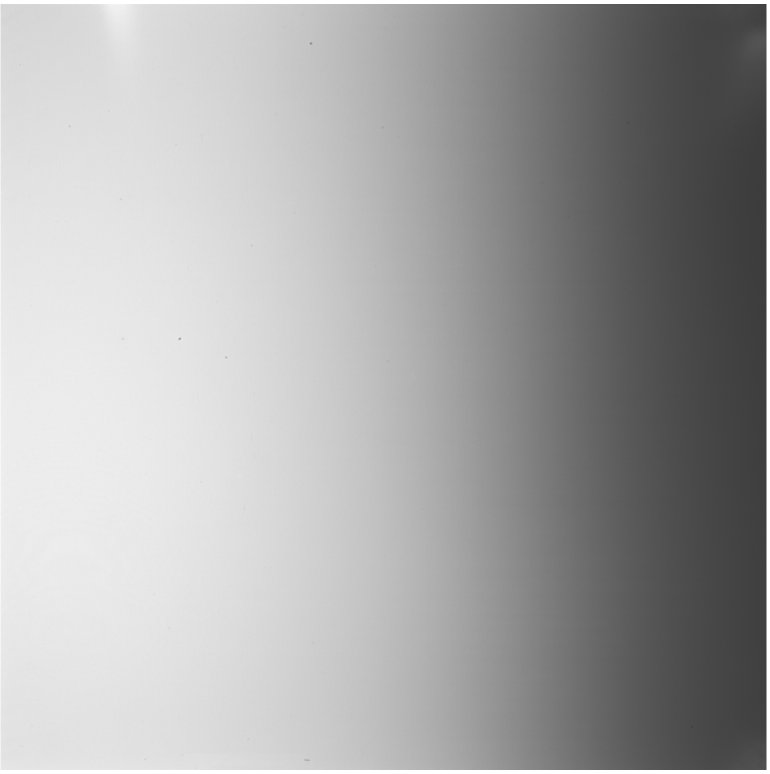


Figure 9.2-4: CAD model of a sensor shifting setup to test for light fall-off and verify output from `ECG designer`. The lens is mounted on a plate with the appropriate mount which is placed at the correct register distance with respect to the sensor which sits on a linear stage mounted perpendicular to the axis of the lens.

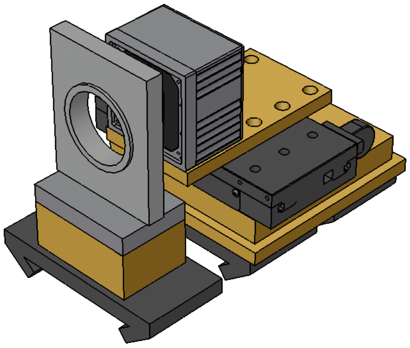


Table 9.2-2: Basic effects of camera parameters.

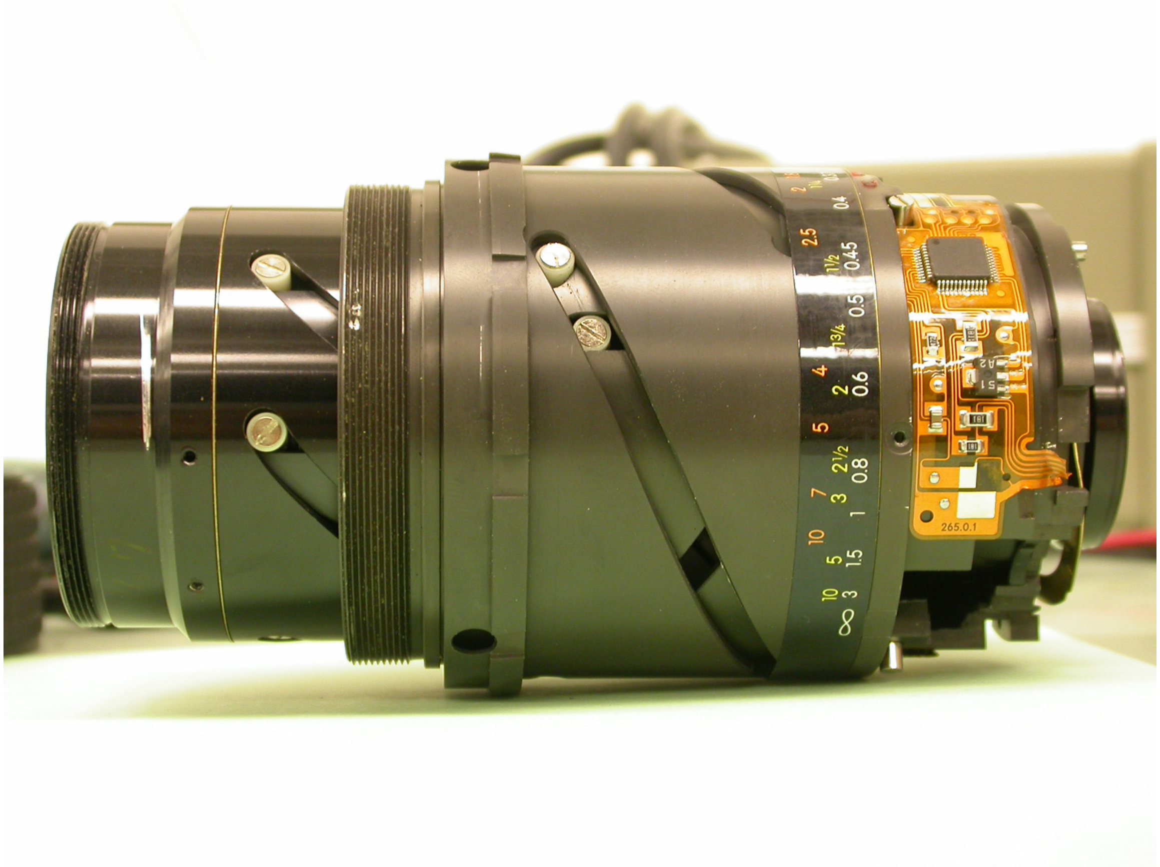
	When increased...	When decreased...
"L"	Increases mappable region size, decreases sensitivity and resolution, and decreases sensor offset; requires longer f or smaller sensor size to maintain mappable region and larger s_{ij} to maintain sensitivity.	Decreases mappable region size, increases sensitivity and resolution, and increases sensor offset; requires shorter f or larger sensor size to maintain mappable region and shorter s_{ij} to control sensor offset.
"sij"	Decreases mappable region depth, increases sensitivity, and increases sensor offset and light fall-off.	Increases mappable region depth, decreases sensitivity, and decreases sensor offset and light fall-off.
"h", "w"	Increases mappable region size and increases maximum sensor diagonal and light fall-off; requires longer f to maintain mappable region; probably decreases framerate.	Decreases mappable region size and decreases maximum sensor diagonal and light fall-off; requires shorter f to maintain mappable region; probably increases framerate.
"pmm"	Increases resolution and decreases f-number needed for diffraction blurring.	Decreases resolution and increases f-number needed for diffraction blurring.
"f"	Decreases mappable region size, increases sensitivity and resolution; requires longer L to maintain mappable region.	Increases mappable region size, decreases sensitivity and resolution, increases [barrel] distortion; requires shorter L to maintain mappable region.

9.3.2 The Lens

Aside from having the desired focal length, the lens construction has to be considered. Any minute movement of a lens or its components will render a multi-plane dewarping set useless. As rigid as a focusing mechanism may seem to the touch, there will be several pixels of play. So the lens should be locked in place as rigidly as possible. In the modern cameras this is done by cleaning the focusing threads off the lenses (which are chosen purposefully to be of the manual-focus variety) and the assembly is glued with liquid epoxy once the lenses are focused during assembly. Although it could possibly be easier to simply remove the lens unit from the barrel and build a custom holder, this has the advantage that it is easy to focus the lenses during assembly.

Manual focus lenses have the advantage that the assembly is relatively simple and the lens moves as a unit while focusing. Autofocus lenses, on the other hand, can have quite complex focusing movements. The Micro-Nikkor 105mm autofocus lens has a complex, four-layer system of curved tracks for focusing in which one lens group stays stationary (relative to the camera), the next moves in two different directions through the focusing track, and the remaining two move in the same direction but at different rates. Fixing such a lens would be impossible, unless careful measurements of the inter-group distances were taken and a custom, rigid barrel was made. The probability of this being successful is close to zero, as the only hope of doing so would require the entire mechanism to be modeled in a CAD program since there is no access to the elements themselves while the lens is still in a state where the focusing mechanism is functional.

Figure 9.3-1: The Micro-Nikkor 105mm autofocus lens with two of the four layers of curved tracks that comprise its focusing mechanism exposed.



9.3.3 The Sensor

To remove the hysteresis of the heat movement of the sensors there are two alternatives: the Revelle Camera approach or the Ian Camera approach.

The Revelle Camera method may be used, whereby the chips are mounted to custom-made aluminum frames. The faceplate is assembled with lenses in place and the sensors are aligned to the faceplate by external stages, verifying the alignment by continuously imaging a target. The faceplate should have “receptacles” in which the frames sit with a gap between the frames and the walls of the receptacles. Once the sensors are in place the gap is filled with epoxy.

Although the alignment of the sensor is overkill, especially with the relatively well-aligned assembly of the Kodak chips[3], by mounting the chips to a small frame the warm-up time is reduced

[3] At least the KAI 2001’s in the Ian Camera have pixel planes well aligned to the package.

if the epoxy is not a good heat conductor. The Revelle Camera has a massive faceplate but only 4 minutes elapse before the movement in the sensors has stabilized.

The Ian Camera approach entails duplicating the part onto which the chips are mounted in the original sensors into the back of the faceplate and then mounting the chips directly to the faceplate. A bit of epoxy was used in the corners of the chip for safety in the Ian Camera but this was omitted in the Emilio Camera. The direct-mounting approach has the added advantage that no external stages are necessary, but since the entire faceplate is now the heat sink of the sensors, warm-up times can be considerably longer. The warm-up time on the Ian Camera is conservatively estimated at 30 minutes.

Many sensors today include heat spreaders in their design—usually a simple plate that extends along the back of the chip and bolts to the sensor body somehow. Some sensor manufacturers use this as the structural mount for the package, that is, the package is bolted to the heat spreader which is then bolted to the sensor faceplate. In the modern defocusing cameras the opposite approach is used: the package is bolted to the faceplate and the heat spreader is only connected to the package via thermal transfer grease (no rigid mechanical link). It has been proposed to machine the heat spreader as part of the faceplate (all one single piece) and mount the sensor to this thin strip faceplate, however, it is not clear what thermo-mechanical effects this would have.

Placement of the protective glass on chips can be quite arbitrary and this reduces further the available front contact area; of the three KAI 2001's in the Ian Camera, one had the glass well over 10° off horizontal. The specifications for the KAI 40XX, on the other hand, call for much more precise glass placement (because the glass is bigger relative to the package). The faceplate must be designed with these tolerances in mind unless the intention is to remove the glass altogether or reposition it with better alignment.

It should be *emphasized* here that the image orientation on the sensor should be checked prior to design. In the case of the KAI-40XX, the image is upright when the long end of the package is vertical, that is, the chip is "sideways". This is indicated in the full specification of the chip but *not* in the mechanical drawings, where the chip is shown with its long side on the horizontal like the drawing for all other Kodak chips.

Tolerances on mechanical aspects of the ceramic package can be surprisingly loose, especially for the dowel holes meant for alignment during assembly. These must be treated carefully in the design of the camera. One of the alignment holes is usually a slot so concentricity is not a big problem, but diameter can be. The alignment pins should be manufactured separately and *not* "built into" the faceplate (as was done with the Ian Camera) or made to be "press fit" so they can be replaced in the case that they do not fit a specific chip. In the case of the Emilio Camera, the dowels that were made are exact replicas of the ones made for ImperX.

It is best to design the camera such that the sensor body (minus the front, which is replaced by the faceplate) can be fully assembled and fully encloses the sensors so that even without the back part of the camera assembly the sensors remain independently housed units.

9.3.4 Optical Channel

In professional photography cameras, glare from light that is off the image can cause a loss of contrast in the images. Many manufacturers adopt devices such as lens hoods or baffles in the lens-film space to reduce or completely block stray light. In custom machining these may be expensive.

Both the Ian Camera and the Emilio Camera have simple cavities aft of the lenses. The aluminum faceplate is anodized black but no other precautions are taken. The Emilio Camera was bead-blasted before the anodization so that a more matte finish was achieved, but whereas the Ian Camera has no glare problems, the Emilio Camera has slight spots of glare when the probe volume is illuminated laterally (as is done normally). Simple masks could be made to mount in the lens cover window that simply block all light except that in the field-of-view cone of each sensor, but these would be tedious to align and the glare spots are small enough to be of minimal concern.

The glare spots can be seen as hazy spikes of brightness, clearly visible in figures 9.2-2 and 9.2-3.

Chapter 10

Software

10.1 Introduction

Completing the system is a custom software suite written to process raw images into three-dimensional particle and vector fields.

The suite can be divided into two parts: calibration software, comprised of `gridfind` and `dewarpC`, and the processing software, called `DDPIV`, written by Francisco Pereira and Emilio Graff.

Extensive detail of the operation of the processing software necessary to operate a DDPIV camera can be found in part II of Graff and Pereira [2007].

10.2 The Calibration Software: `gridfind` and `dewarpC`

Once the images of the grid are acquired as described in section 8.4.2, the position of the images of the dots in each aperture is found using `DDPIV`'s Gaussian fitting algorithm. `gridfind` then creates a list which links the imaged grid points to perfect grid points, and `dewarpC` obtains the coefficients to second, third, and fourth order functions which map the imaged points to perfect points. The application of the dewarping set in processing is deeply intertwined with the particle image matching routine in `DDPIV`. Both of these are command-line programs.

10.2.1 `gridfind` version 2.0.1d

The purpose of `gridfind` is to "walk" the grid and determine how many grid units away from the origin a certain point is so that each imaged point is matched to its location on a perfect grid.

First, `gridfind` searches near the center of the image for two adjacent dots to establish an initial guess of the grid spacing. The origin is found by traversing through the entire grid looking for the set pattern of a dot with three of its four diagonals missing, using this initial spacing guess to create a "mask". Because of this search method, if for any reason dots missing on the imaged grid make it look as though more than one dot was the origin a malformed grid will be created.

Once the origin is identified the average distance to the four closest points is used as the final spacing estimate. The perfect grid will have its dots spaced by this number averaged over all the apertures for a given plane (as one of the requirements of defocusing alignment is that all sensors have the same magnification). Because the physical spacing of the dots is known (from printing the grid) the correlation between pixels and space coordinates can be established. With the spacing established, `gridfind` proceeds to walk along columns and rows, matching the next dot on the grid as the dot closest to the expected location on the image. The expected location is based on the spacing and the angle between the last two dots identified with a tolerance of a fraction of the spacing. Because of this, it is somewhat robust against a few extra points being identified in the Gaussian fitting step, such as dirt on the grid, bubbles on the tank, or dirt/bad pixels on the sensor. It is also able to reconstruct the grid of severely curved images (such as those from lenses with high barrel distortion). The "walk" is done four times; overlapping the grid by first walking vertically looking horizontally and then walking horizontally looking vertically so that in most cases even isolated points in the corners will be counted.

In the end `gridfind` outputs a text file which assigns each measured dot a place in the perfect grid.

10.2.2 `dewarpC` version

The purpose of `dewarpC` is to calculate the dewarping (correction) coefficients. The mappings are of the form

$$\begin{aligned} x_d &= \frac{\sum_{i=0}^{o}\sum_{j=0}^{o-i} A_{ij} x_u^i y_u^j}{\sum_{i=0}^{o}\sum_{j=0}^{o-i} B_{ij} x_u^i y_u^j} \\ y_d &= \frac{\sum_{i=0}^{o}\sum_{j=0}^{o-i} C_{ij} x_u^i y_u^j}{\sum_{i=0}^{o}\sum_{j=0}^{o-i} D_{ij} x_u^i y_u^j} \end{aligned} \tag{10.2-1}$$

where x_d and y_d are the corrected ("dewarped" or perfect) coordinates of a point and x_u and y_u are its measured ("undewarped" or imaged) coordinates. The coefficients A_{ij}, B_{ij}, C_{ij}, and D_{ij} are obtained by minimizing the sum of the square of the error between the x_d and y_d and the respective perfect grid location for all the measured points using the `lmdif` function of the `MINPACK` library. Coefficients are calculated for mappings of second order ($o = 2$), third order ($o = 3$), and fourth order ($o = 4$). Processing allows the option to use an average of any two or all three mappings, a specified order for all planes, or the best performing order for each plane. The order that performs the best is that which has the lowest standard deviation of the square of the error.

`dewarpC` can be told to warn the user if the standard deviation of the error exceeds a particular value; this can be used to help detect "hiccups" which can happen when the minimization gets stuck at a local minimum and yields an incorrect mapping.

10.3 DDPIV version 4.6.0c R14 IPP

DDPIV is the main data processing software. The original version was written by Francisco Pereira in the late 90's and was a collection of independent DOS programs, each performing a task in the overall processing. In homage to this, the principal processing steps are named after these programs.

- FINDPART is the first step, which takes the images and reconstructs the point cloud.
- FILTERPART performs basic transformations and filtering on point clouds and also calculates particle statistics.
- FINDFLOW takes two point clouds as input and calculates velocity fields.
- FILTERFLOW performs basic transformations and filtering on vector fields.
- FLOWSTAT calculates vector statistics.

IMAGEGEN is another large part of the processing software that is not a processing step but rather a pinhole camera simulator. It has been essential to the development of algorithms as a testbed.

The software is command-line capable, complementing the scriptability of Video Savant in creating a completely automateable package.

10.3.1 FINDPART

FINDPART, as mentioned above, has two main functions: to detect particle images, and to find particles in space. It includes routines for image pre-processing, particle image detection, application of multi-plane dewarping, converting particle images to particles in space, and the correction of particle Z coordinates due to refraction from a multi-medium experimental setup.

10.3.1.1 Image Pre-processing

Depending on the selected options, DDPIV may perform some pre-processing steps on the raw images before the particle search begins. This can be divided into two categories—image manipulation, which consists of preparing image files for processing (such as splitting a multi-sensor image), and image alteration, where actual pixel values are changed. These steps proceed in order by aperture. If the images are double frame, then both frames of an aperture are processed before the next aperture is processed. Regardless of the image layout, the files should be 8-bit BMP files.

After the images are sorted into single-frame, single-sensor images, the image in memory is altered according to the selected options. There are 5 possible pre-processing functions: background subtraction happens first, then de-interlace, invert, blur, and normalize, in that order. Those functions that benefit from the optimization are implemented using Intel's Image Processing Primitives (IPP) library.

- Background subtraction allows for an image to be subtracted from all experimental images. Because the Gaussian fitting step simply searches for bright spots in the image, illuminated background elements can contribute a lot to the ghost particle population, so it is important to get an image which has *only* particle images.

- De-interlacing takes the rows and then the columns of the image and multiplies them by some constant so that the average pixel value of two adjacent ones is the same. It is useful for analog sensors, such as the ones in the Lunchbox Camera.

- Invert simply inverts the pixel values of the image. `FINDPART` expects the particle images to be lighter than the background; in the case of a light surface marked with dark dots it is necessary to invert the image first.

- Blur is an implementation of Gaussian blur using a 13 by 13 pixel kernel. Normally the blur radius will not exceed 3 pixels. Blur can be used when the images are extremely noisy, although in normal circumstances, tests with dewarping target images show no discernable increase in the precision of the Gaussian fit with blur turned on. However, the current implementation of the Gaussian fitting routine fails with saturated particle images (see below) and if the particle images are big enough, blurring them substantially is the only way to recover any data from such images.

- Normalize takes the pixel value histogram of the image and stretches it so that there is at least one pixel with value of 0 and one with value of 255.

`FINDPART` also includes an image masking function, but it is not implemented on the pixel values themselves but rather during the Gaussian fitting step.

10.3.1.2 Particle Search

After the images are prepared, pixel threshold values are used to mark possible particle images. The user defines a particle image maximum radius, which should be large enough to cover all the particle images of interest, and `FINDPART` searches areas of this size pixel-by-pixel through the entire image. The mask is implemented at this step—if a particular image coordinate has value 0 in the mask, then `FINDPART` continues without checking the pixel values there. Once a non-masked image coordinate is found, the pixel values within the search radius of that image coordinate are checked. If the maximum pixel value inside this radius is larger than the threshold set by the user for this aperture then that pixel coordinate is checked against the maximum allowable overlap, also set by the user. Typically this is 50%, so if the particle images have a clear maximum pixel and values sloping down around it, the pixel-by-pixel check will not mark multiple “bright spots” per particle image. If a maximum pixel value passes the mask, threshold, and overlap tests, then the set of pixels around it becomes the kernel of pixels for a Gaussian fit. If a heavily saturated particle image is passed through this procedure, then normally `FINDPART` will mark multiple possible particle images per saturated particle image because the first 255-valued pixel is taken as the maximum for a particular search area, thus marking the edge of a particle image for fitting. Even if the overlap is set to 0 this will likely result in multiple particle images being tagged instead of the single saturated one. By blurring the entire image as mentioned above, the saturated particle images lose their plateau shape and a single, centrally located pixel becomes the maximum value so the particle image search will be successful.

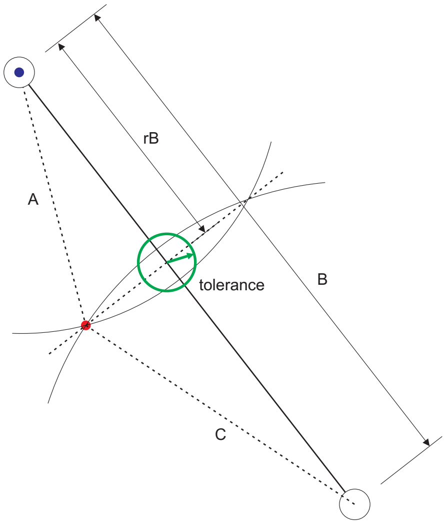
For each prospective particle image detected by threshold, a two-dimensional Gaussian is fitted over the pixel values, and the center of this Gaussian is taken to be the location of the particle image. The fit is done by minimizing the difference between the pixel values and the Gaussian within the kernel defined as above using the `lmdif` function of `MINPACK`. (This function minimizes the sum of the squares of n nonlinear functions using a modified Levenberg-Marquardt algorithm and is publicly available.) If for some reason the fit fails, the software defaults to taking the centroid of the kernel as the particle image location.

At this point, the images are discarded, and `FINDPART` proceeds with a list of x, y pixel coordinates of particle images for each aperture. Internally, the software takes all these coordinates and dewarps them according to the coefficients obtained with `dewarpC`. All dewarping planes are interpreted independently. By default, the dot on the origin of the grid is mapped to $0, 0$ for each plane, and all the other dots are mapped so that they form a perfect grid. At the reference plane this is correct, but for all other planes the particle image coordinates must be shifted according to the theoretical pinhole-optics separation between apertures as defined in equation 4.3-35.

Particle image matching continues with a coarse tolerance first, and using the dewarping coefficients from the reference plane. Figure 10.3-1 is an exaggerated view of the matching procedure. Two of the camera's apertures are shown as white circles; they are separated by a distance B. The particle image from the first aperture, shown as a blue circle, is superimposed onto the aperture location since everything is measured relative to this point. The particle image in the second aperture, shown as a red circle, is being checked for the possibility of a prospective match. The distance between the two particle images is A, and the distance between the second particle image to its aperture is C. The circles traced by these two radii intersect and form a line segment which cuts the segment between the apertures a distance rB from the blue particle image. If this were a perfect pinhole camera with no error, then the two circles would intersect at one point, and rB would be exactly the separation predicted by equation 4.3-35. In the case shown in figure 10.3-1,

$$\mathsf{rB} = \frac{\mathsf{A} + (\mathsf{B} - \mathsf{C})}{2} \tag{10.3-2}$$

Figure 10.3-1: Exaggerated view of the particle image matching criteria used in `FINDPART`.



In a real camera, the second (red) particle image is allowed to be within a certain tolerance of this perfect location. The tolerance is shown as a green circle in figure 10.3-1. At the first stage, where all particle images are dewarped using the reference plane coefficients, the tolerance is usually large (1.5 pixels for the Ian and Emilio Cameras) and is known as the “coarse” tolerance. If the second particle image is a match with the first using the coarse tolerance[1] then a preliminary Z is calculated using rB as the separation. The particle image coordinates are replaced with a linear interpolation of the coordinates given by the dewarping coefficients of the two dewarping planes on either side of this preliminary Z or as the coordinate given by the last dewarping plane if the preliminary Z is beyond (closer to the camera than) the last dewarping plane.

The calculation depicted in figure 10.3-1 is repeated with the new particle image coordinates and using the “fine” tolerance (typically 0.5 pixels for the Emilio Camera). If the two particle images still match then the next particle image is sought in the next aperture using the fine tolerance, always updating the preliminary Z at each new aperture, until finally N particle images are matched in N apertures and a final Z is calculated using the separations between $N-1$ apertures.

After a particle is created its Z coordinate is corrected for the refraction of the different indexes in a multi-medium experiment. Because the dewarping coefficients are calculated from images taken in a similar multi-medium setup, the correction is a simple rescaling in Z by as defined by

$$Z_{true} = (Z_{apparent} + L_1 + L_2/n_{12})\, n_{13} - L_2 - L_1 \tag{10.3-3}$$

where 1, 2, and 3 are the three media from camera to particle location (typically air, wall material, and experiment fluid, respectively), L_i are the distances in those mediums, and n_{1i} are the indexes

[1] Note that figure 10.3-1 shows a pair of particle images that *do not* match.

of refraction of the medium relative to air. This is a simplified version of the correction depicted in equation 20 of Pereira and Gharib [2002], since by doing the dewarping in the fluid all X and Y dependency of the correction is removed.

FINDPART is the key step in the processing because its result is the reconstructed point cloud corresponding to the locations in space of the particles that were imaged. Typically vector fields are obtained using particle tracking algorithms and thus the error introduced in this step propagates directly to those results. Great care has been taken to ensure that the calibration is accurate and that the hardware design is such that once a calibration is obtained the results are consistent from power cycle to power cycle. Chapters 12, 13, and 14 examine in detail the simulated and experimental precision of the key parts of the reconstruction performed in FINDPART.

10.3.2 FILTERPART

FILTERPART's three primary functions are to alter the point cloud results from FINDPART through geometric transformations (translation, rotation) and population density filtering, to calculate particle field statistics in an averaged data set, and to apply the particle sizing algorithms[2].

The population density filter uses a simple neighborhood density criterion. If a certain particle has less than a threshold value of neighbors within a user-defined radius, it is removed. The filter can be applied iteratively, ensuring that all remaining particles meet the criterion. The filter can be reversed such that the remaining particles are the ones in the higher density regions. This method of filtering is very useful in removing solid surfaces that may have been illuminated by the laser during a flow experiment. It is also convenient when mapping surfaces marked with a wide range of dot sizes, since then it is likely that the particle image search radius in FINDPART will be too small for the large particle images resulting in a single large particle image being identified as a clump of small ones which, combined with the tolerance during matching will yield a clump of particles. The population density filter will effectively remove all but one of these particles, though the one that remains has no correlation to the original particle image since it is picked according to its order in the data structure as the recursive filter runs.

10.3.3 FINDFLOW

FINDFLOW is responsible for obtaining a vector field from two point clouds. There are two methods: a discrete three-dimensional cross-correlation, designated “PIV”, and three different algorithms for particle tracking, commonly designated “PTV”.

[2] Particle sizing will not be discussed here. Its implementation is described in Pereira and Gharib [2004].

10.3.3.1 PIV

The cross-correlation scheme first divides the domain, or flow cell, into a three-dimensional grid of volume elements called voxels. Typically they are laid out such that the overlap between voxels is 50% of the dimension on each side.

The correlation itself is computed by creating a three dimensional Gaussian for each pair of particles whose X, Y, Z position is the midpoint between the two particles' position—that is, if there are M particles in the voxel for the first frame and N particles for the second, then $M \times N$ Gaussians are formed. During FINDPART, each particle is assigned a radius equal to the average of the $1/e^2$ radii of the particle images that collectively matched to form that particle. During FILTERPART this radius can be converted to real units using a sizing calibration and application of the Mie-scattering-based sizing algorithm. The pair Gaussians formed for the PIV calculation have standard deviation equal to $\sqrt{r_i^2 + R_j^2}$, where r_i is the radius of the particle in the first frame and r_j that of the one in the second. The intensity of the pair Gaussian is set as e^{-a^2}, where $a = (r_j - r_i)/r_i \times 100$.

Once all the pair Guassians are defined, they are added together yielding a three-dimensional correlation space, and the lmqn function from the PDL nonlinear optimization library is used to find the X, Y, Z coordinate of the maximum of this sum. The resulting velocity for the voxel has a vector pointing from the center of the voxel to the location of the maximum of the correlation space and a confidence defined as the maximum value of the correlation divided by the sum of all the pair Gaussian peak values divided by the square root of the number of pair Gaussians.

The PIV method of calculating velocity has been more or less abandoned because it is extremely sensitive to ghost particles and requires several particles per voxel to yield a good vector, which typically implies only 1,000 or so vectors per pair. Moreover rarely are the vector fields presentable without outlier correction (see below) and smoothing.

10.3.3.1.1 Outlier Correction An erroneous vector in a calculated vector field is known as an outlier. They can be caused by several factors, such as the voxel size being too small relative to the displacement, the number of particles in the voxel being too low, or too much noise being generated by ghost particles.

One of the simplest methods for finding such vectors in 2D PIV is to compare each vector the average of its eight immediate neighbors. If the vector deviates more than a certain threshold from this mean, it is deemed an outlier, and can be removed (the preferred route) or replaced by the average.

With three dimensions there is the added advantage that now each vector has 26 immediate neighbors. In DDPIV, outliers are replaced by the weighted average of their 26 neighbors. If the vector checked is assigned i, j, k coordinates of $0, 0, 0$, then neighbors that have $|i| = 1$ and $j, k = 0$, $|j| = 1$ and $i, k = 0$, or $|k| = 1$ and $i, j = 0$ (that is, the 6 next-door neighbors, or centers of the faces of the 3 by 3 cube) are given a weight of 1. Those with $|i|, |j| = 1$ and $k = 0$, $|j|, |k| = 1$ and $i = 0$, or $|i|, |k| = 1$ and $j = 0$ (the 12 in-plane diagonal neighbors, or the centers of the edges of the 3 by 3 cube) are given a weight of $\sqrt{2}/2$, and those with $|i|, |j|, |k| = 1$ (the 8 remaining corners of the 3 by 3 cube) are given a weight of $\sqrt{3}/3$.

This correction can be applied in a multiple-pass fashion, however, it is dangerous and can make any random noise vector field resemble turbulent free-stream flow.

If outlier correction is turned on there is the further option of calculating a refined flow estimate, also known in 2D PIV as window shifting. In this scheme, the first cross-correlation is used as an initial guess. For the second pass, the voxel is shifted in the second frame by the amount indicated by the first vector and the cross-correlation is performed again. The final velocity is the vector sum of these two vectors. The idea behind it is that in all flows it is likely that particles are leaving the voxel between frames, so by shifting the voxel the second cross-correlation should be much stronger than the first.

10.3.3.2 PTV

A recent addition to the software were three particle tracking algorithms originally written by Heinrich Stüer and implemented directly by Francisco Pereira as detailed in Pereira et al. [2006c].

In particle tracking, the idea is to finish with each particle in the first frame connected to one in the second frame. The three algorithms differ in the way they decide which particle pair is a correct match. However they all have in common that the match of a given particle in the second frame must be within some search radius R_s centered about an initial guess which can be 0, the center of mass of the particles in the voxels, the vector obtained through a coarse-grid PIV calculation. R_s is designated as a percentage of the voxel size, thus allowing the user to bias the search by making the longest dimension of the voxel correspond to the direction of the mean flow in the flow cell. This can be very helpful in cases of very directional flow, such as a jet.

10.3.3.2.1 Nearest Neighbor This is the simplest of all the algorithms, and simply chooses the destination particle as the one which is closest in space to the source particle. Its implementation is not quite that simple because it does verify that the chosen vector is the best possible vector for *both* particles.

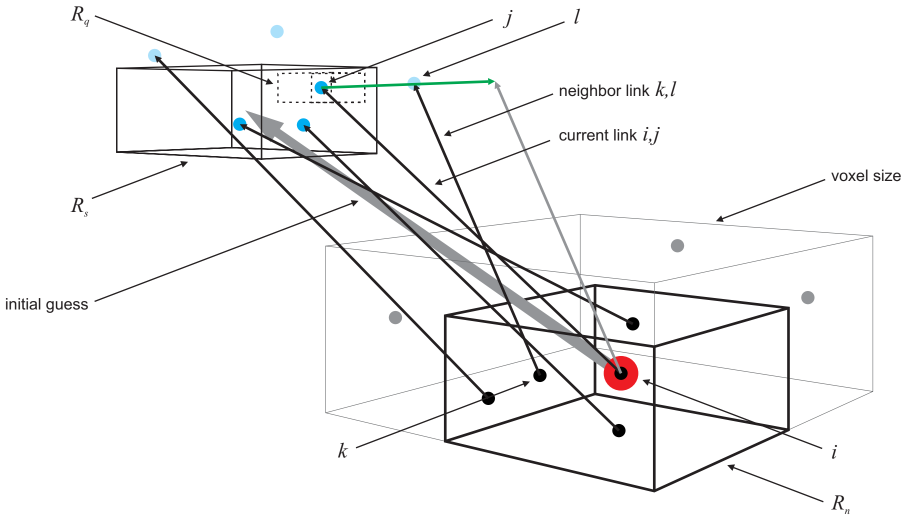
10.3.3.2.2 Neural Network The neural network implementation is based on that of Labonté [1999] with the modification that it is used as a first step in a modified nearest-neighbor scheme. It is computationally intensive and its results did not show exuberant promise in initial testing and thus it is not currently used. It is compared to the other two schemes in Pereira et al. [2006c].

10.3.3.2.3 Relaxation method [3]

The relaxation method is the method of choice, it being nearly as fast as nearest neighbor in most cases but much less susceptible to ghost particles. The key of this method is that neighboring particles are assumed to have similar displacements between frames.

The user designates a neighborhood, R_n, which will enclose all the particles in the first frame whose vectors will be compared according to a semi-rigidity condition. As with the other algorithms, R_s is the search area for destination particles in the second frame, centered about the source particle plus the initial guess vector. The relaxation method also adds the parameter R_q, which is the magnitude of the maximum allowable difference between vectors emanating from particles that are within R_n of each other. As with R_s, R_n and R_q are defined as percentages of the voxel size. Figure 10.3-2 depicts this arrangement in space. The current source particle i, shown with a thick red outline, marks the center of the neighborhood R_n. The N_n particles in the first frame within R_n (black circles) will provide a reference point for the semi-rigidity condition. The initial guess (thick grey vector) shifts the center of the region R_s. The N_s particles in the second frame that fall within R_s (cyan circles) can be possible links for the current source particle. Note that there are particles like particle l in the second frame *not* within R_s (light cyan circles) which may be linked to particles in the first frame that *are* within R_n—these are simply links from a previous iteration on those particles.

Figure 10.3-2: A graphical representation of the relaxation method. See the text for a detailed description.



In the figure, the link labeled "neighbor link" between particles k and l is shown again in grey translated so that its tail coincides with the tail of the current link between particles i and j. Thus the green vector shows the difference between links k, l and i, j. Centered about the tail of this

[3] The discussion presented here follows closely that of Pereira et al. [2006c] and uses similar notation.

difference is the volume R_q, against which the difference vector is checked. In this case the green vector clearly exceeds the bounds of R_q thus the pair of links being checked is "bad". For each pair of links checked, weights are assigned such that

$$Q_{i,j,k,l} = \begin{cases} 1 & \text{if } \|\mathbf{d}_{i,j} - \mathbf{d}_{k,l}\| < R_q \\ 0 & \text{otherwise} \end{cases} \tag{10.3-4}$$

where $\boldsymbol{d}_{i,j}$ is the vector of the link i, j $\boldsymbol{d}_{k,l}$ is that for the link j, k. In the implementation it is a component check—that is, if any component of the vector exceeds that dimension of R_q it is considered "bad". An additional weight can be assigned by comparing the link i, j to the flow estimate:

$$F_{i,j} = \begin{cases} 1 & \text{if } \|\mathbf{d}_{i,j} - \mathbf{u}_{c,i}\Delta t\| < R_q \\ 0 & \text{otherwise} \end{cases} \tag{10.3-5}$$

where $\mathbf{u}_{c,i}$ is the velocity estimate for particle i. The actual value of $\mathbf{u}_{c,i}$ is computed by taking a Gaussian-weighted average of the estimated velocity field (provided by center-of-mass or cross-correlation on a voxelized grid, as mentioned above) where the center of the Gaussian is at particle i and the standard deviation is the largest diagonal of R_s.

These weights are used to favor links that are "good" in a probability assigned to a link which indicates the likeliness that it is the best link for a given particle i within a neighborhood R_n.

For iteration a, the probability is updated according to equation 10.3-6, which is a modified version of the probability presented in Barnard and Thompson [1980]:

$$\tilde{P}^{(a)}_{i,j} = P^{(a-1)}_{i,j} \left[A + B \left(\sum_{k=1}^{N_n} \sum_{l=1}^{N_{p_k}} P^{(a-1)}_{k,l} Q_{i,j,k,l} \right) + CF_{i,j} \right] \tag{10.3-6}$$

where N_{p_k} is the number of possible links of a neighbor particle k to the second frame (the number of k, l links). The constants are fixed as $A = 0.3$ and $B = 3.0$ in Barnard and Thompson [1980] and the constant C is defined in Pereira et al. [2006c] to be $C = 1.0$. In words, if the link i, j does not satisfy the conditions set in equations 10.3-4 and 10.3-5 then it is severely punished. The strongest reward comes from the probabilities of particle i's neighbors' links which satisfy the semi-rigidity condition of equation 10.3-4.

The distinction between $\tilde{P}$ and P is that $\tilde{P}$ is not normalized, that is, $\tilde{P}$ does not satisfy

$$P^{\star}_i + \sum_{j=1}^{N_{p_i}} P_{i,j} = 1 \tag{10.3-7}$$

for a given iteration. That is, the sum of the probabilities of each link emanating from particle i plus the probability that there is no link from particle i must equal 1. $\tilde{P}$ is normalized at iteration

a by

$$P_{i,j}^{(a)} = \frac{\tilde{P}_{i,j}^{(a)}}{P_i^{\star(a-1)} + \sum_{j=1}^{N_n} \tilde{P}_{i,j}^{(a)}} \tag{10.3-8}$$

and $P^\star$ is set by

$$P_i^{\star(a)} = \frac{P_i^{\star(a-1)}}{P_i^{\star(a-1)} + \sum_{j=1}^{N_n} \tilde{P}_{i,j}^{(a)}} \tag{10.3-9}$$

Initially the probabilities are set to

$$P_{i,j}^{(0)} = P_i^{\star(0)} = \frac{1}{N_{p_i} + 1} \tag{10.3-10}$$

10.3.4 FILTERFLOW

`FILTERFLOW` serves a similar purpose to `FILTERPART`. It allows for cropping the flow cell and filtering the vector field by minimum and maximum velocities.

10.3.5 FLOWSTAT

`FLOWSTAT` is the statistical side of the flow processing. The mean is calculated first, then RMS and cross-RMS are computed concurrently. For flows calculated using PTV, `FLOWSTAT` includes a fast sorting algorithm to divide the vectors into voxels.

If $\mathbf{U} = (U, V, W)$, the formula for RMS vector in voxel k is

$$\mathbf{U}_{k_{RMS}} = \sum_i^N (\mathbf{U}_{k_i} - \overline{\mathbf{U}}_k)^2 / N \tag{10.3-11}$$

where $\overline{\mathbf{U}}_k$ is the mean over the frames of the vector and N is the number of vectors in voxel k. For the cross-RMS, the formula is

$$\begin{aligned} U_{k_{\times RMS}} &= \sum_i^N (V_{k_i} - \overline{V})(W_{k_i} - \overline{W})/N \\ V_{k_{\times RMS}} &= \sum_i^N (U_{k_i} - \overline{U})(W_{k_i} - \overline{W})/N \\ W_{k_{\times RMS}} &= \sum_i^N (U_{k_i} - \overline{U})(V_{k_i} - \overline{V})/N \end{aligned} \tag{10.3-12}$$

where the barred quantities, again, denote the mean quantities. `FLOWSTAT` also keeps track of the number of vectors and the standard deviation in each voxel and includes the capability of filtering vectors based on deviation from the mean to complement `FILTERFLOW`'s filtering capabilities and to provide an alternative to outlier correction in PTV flows.

Part III

Performance of the System

Chapter 11

Testing Methods

11.1 Introduction

The performance of the DDPIV system was tested extensively both with simulations to test the algorithms of the software and with real-world tests of the actual system in practice (using both the Ian Camera and the Emilio Camera).

Testing was divided into the basic functions of the software: particle cloud reconstruction and velocity calculation. The particle cloud reconstruction is further divided into its three principal components: Gaussian fitting (chapter 12), multi-plane dewarping (chapter 13), and particle image matching (chapter 14).

There are two contributors to what is collectively called the "performance" of an algorithm. The first is loosely referred to as the "quality" of the reconstruction, and categorizes how much of the available information an algorithm was able to extract—for example, the number of particle images that were properly identified by the Gaussian fitting algorithm or the ratio of real to ghost particles in a reconstructed point cloud. The second is the "precision" of the reconstruction, which is the error in the correctly reconstructed elements—such as the error in the center of the fitted Gaussian or the error in the position of a reconstructed real particle. Both aspects are important in DDPIV.

Generally speaking, the quality of a reconstruction is more sensitive than the precision of the result. In practice, it is sometimes possible to relinquish quality to obtain higher overall precision, but in the cases of instantaneous dynamic events (such as non-repeating flow) it is imperative to be able to obtain a high-quality result at an acceptable precision.

11.2 Simulations

The simulated data in most cases was generated to test very specific aspects of the algorithms. In testing the components of `FINDPART`, the same point cloud was used for all the tests. These point clouds were generated pseudo-randomly through `IMAGEGEN` so that all the particles are confined within a 1 µm thickness, making it easy to isolate ghosts. The point clouds used are shown in section B.1 of Graff [2007b].

The tests were designed from the point of view of the end user, and in many cases the results can be used to compare directly with real data results to get an estimate of how well a particular experiment is going.

Each of the following chapters further detail the testing methods pertinent to the algorithms they cover.

Chapter 12

Gaussian Fitting Algorithm

12.1 Introduction

At the heart of the defocusing technique is the identification of the location in space of the particles. This is done in two steps: first, the particle images themselves must be identified in each aperture's image; second, these particle images must be matched between apertures using the multi-plane dewarping calibration.

The Gaussian fit in fact depends first on a substep—that of marking potential particle images with a pixel value threshold as discussed in section 10.3.1.2. A kernel of pixels is taken centered about the bright point of a particle image obtained by the threshold and these pixels are used in the fitting step.

This Gaussian fit yields four quantities: the sub-pixel center, major and minor radii, and the amplitude. The amplitude and radii are used in the particle size measurement algorithm (and indirectly in the cross-correlation algorithm, see section 10.3.3.1). Only the sub-pixel center is used in the reconstruction of the point cloud.

The three factors that affect the Gaussian fit are the size (radius), brightness, and quality (that is, how truly Gaussian it is) of the particle image and the noise in the image. This chapter presents a discussion of the results of various test conditions with example results. The unabridged results are in Graff [2007d].

12.2 Simulation Details

To ascertain the performance of the Gaussian fitting algorithm, several simulations were performed using the `IMAGEGEN` routine in `DDPIV`. The point cloud used is the one of figure B.1-1 in Graff [2007d], and it is displaced 149 times, one millimeter in Z each time. For the tests of the Gaussian fitting this particle density was chosen because it was the one with the highest number of particles with still only a minimal number of overlapping particle images. All 150 images were processed and the statistics presented are for all 150 images in two horizontally opposed apertures (so a total of 300 images). This corresponds to $750 \times 300 = 225{,}000$ particle images generated.

Each particle image is a perfect Gaussian, generated at the sub-pixel scale, value-sampled to discretize it into pixel space. The radius of the particle image is defined as the number of pixels between the center and the point at which the value of the Gaussian is $1/e^2$ of the maximum.

Noise in the image is pseudo-random in a normal distribution, whose magnitude is the maximum amplitude. Sample images showing the appearance of the particle images and the noise can be found in section A.1 of Graff [2007d].

The selection of the kernel, or the set of pixels over which the Gaussian is fit, is important. Too small a kernel will yield pixel values with not much change in slope, while a kernel that extends well beyond the particle image may not converge even if the particle image is completely isolated (no other particle images are within the kernel). This behavior seems to be consequence of the `lmdif` function itself, however, rarely in a real case will particle images be so isolated that a large kernel would not encompass more than one particle image.

A “good” size for the kernel ranges from one which covers all “visible” pixels of a particle image (meaning all pixels which, to the end user, seem to be affected by the particle image) to one slightly larger than the particle image (has at most a one-pixel-wide border of zero-pixel values). When there is a lot of noise in the image, it is beneficial to have the kernel at the larger end of this range in case the noise affects the initial guess for the center of the particle image. In the tests of the algorithm, the kernel was chosen to be the smaller one in this range.

The threshold was chosen so that all particle images were detected. This may mean that some of the noise was mistakenly identified as a particle image, especially in the simulations with the highest noise.

To simulate real data processing, the maximum permissible particle image overlap was set to 50%. This means that the location between two adjacent peaks may be no higher than half the kernel radius.

12.2.1 Measuring the Results

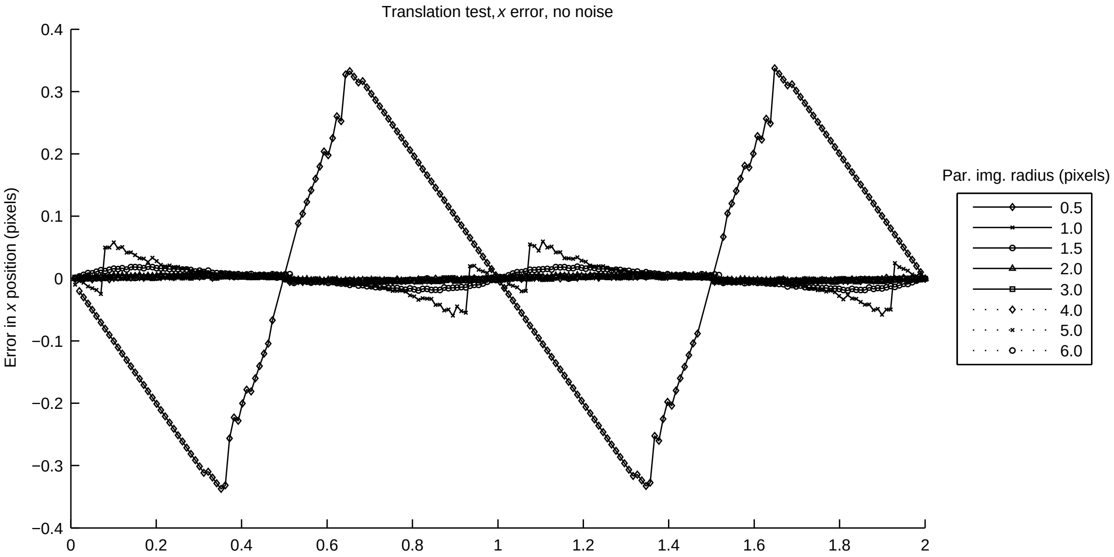
To interpret the results, the fitting error for the sub-pixel center was confined to be less than the kernel radius. That is, if a particle image was detected at some point x_p, y_p, then the closest real particle image within the kernel radius was assumed to be the one that generated that fit. If there were no particle images in this range, then the detected particle image was labeled as an “extra” particle image. If there were more than one particle image within this radius, then the detected particle image was labeled as a “double” particle image—this happens when two particle images overlap, in which case the error is usually larger than for an isolated particle image, and in some cases, can result in one particle image detected for two actual particle images. Any particle images that were never detected are labeled as “lost”. The accuracy in the fit is checked only for those detected particle images that do not fit any of these categories, that is, they are solid fits on isolated particle images. These are labeled as “real” particle images.

12.3 Effect of a Discretized Domain

Typical particle images in experiments occupy no more than 2 pixels in radius. Even though the results presented in section 12.4 show that at this radius the fitting algorithm performs well, a second test was performed to see if there are any effects due to the fact that the image is a discretized domain made of square pixels.

This test consisted of looking at the error in the sub-pixel center of the Gaussian fit with respect to systematic fractional pixel shifts in the generated image: a particle image was generated so that its center was a whole number in the image coordinate system. Subsequent images were created with the same size particle image with its center shifted in x by precisely one hundredth of a pixel in one direction only. If indeed there is no “pixel locking”, then the sub-pixel centers of the fits should also be evenly distributed with only some random error.

Figure 12.3-1: Error in the x coordinate of the sub-pixel center of the Gaussian fit in a systematic translation of 1/100 of a pixel per step for images with no noise.

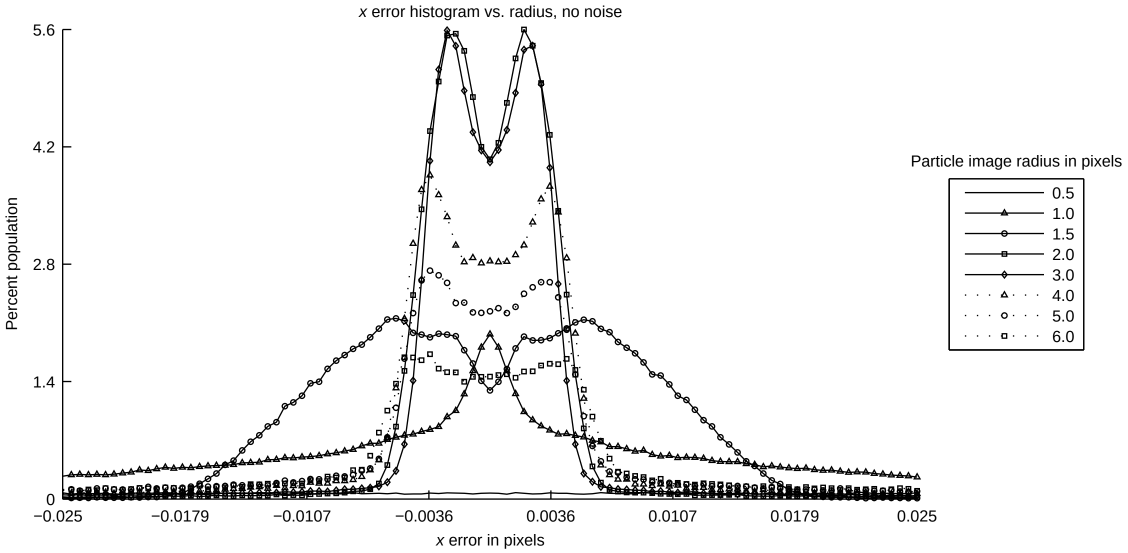


As figure 12.3-1 shows, there is bias in the error at all particle image sizes, but particularly those below 2 pixels in radius. The 0.5-pixel particle images show the most susceptibility to this, which is obvious since at that size they are no more than at most two non-zero pixels. Still, the error is not evenly distributed.

As can be seen by the figures in part I of Graff [2007d], the magnitude of the error decreases greatly as the particle image radius approaches 2.0 after which it remains nearly the same. The bias varies in shape but at these sizes it is much smaller than any of the errors introduced by other factors, such as image noise. Subsequent plots also show that the errors in x and y in the translation test are independent, as the tests with no noise exhibit 0 error in y.

This slight bias is also visible in the pseudo-random point cloud tests. The histograms in figure 12.3-2 shows that, starting at a radius of 2.0, the error is extremely well confined but exhibits two population peaks symmetrical about 0.

Figure 12.3-2: Histogram of the pixel error in the x coordinate of the sub-pixel coordinate of the center of recovered particle images for the case of no noise in the image.



The fact that the discretized domain of the image introduces a slightly non-random error can even be seen in the results of the point cloud reconstruction tests of chapter 14.

12.4 Effect of Image Noise and Particle Image Radius

The size of the particle image in pixels is an important factor in the final quality of the fit. Obviously, if the particle image is smaller than one pixel, then at most only four pixels can possibly contain information about said particle image and thus the fit is almost arbitrary (no sub-pixel resolution can be expected). As the size grows both the time needed to perform each fit[1] and the chance of overlap increases.

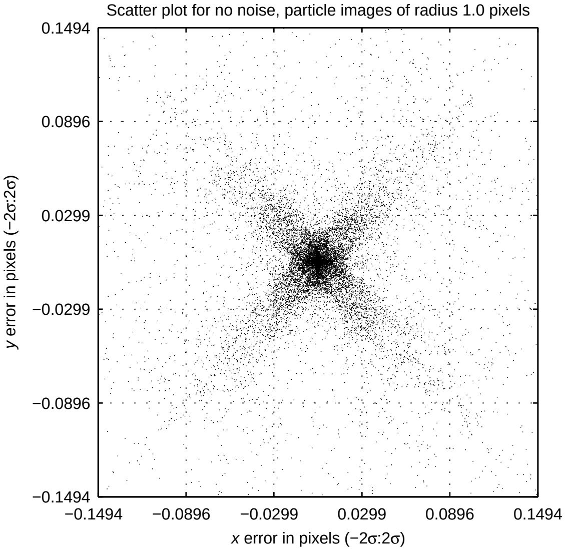
For this test, all particle images had a magnitude equivalent to “bright” (as described in section 12.5), which means that at most one pixel on each particle image had a value of 255.

[1] The computational time is of course dependent on the size of the kernel, which in turn is about the same as the size of the particle image.

For reference, enlarged samples of the images generated can be seen in section A.1 of Graff [2007d].

The case of perfect, noiseless images will be examined first. In this situation, the discretization error is evident, as was seen above. Scatter and average error plots show that the smaller the particle image, the more dependency between x and y error that appears.

Figure 12.4-1: Scatter plot of the error in the sub-pixel coordinate of the center for at most 15,000 recovered particle images for the case of no noise in the image and particle images of radius 1.0.



The scatter plot of the error for 1-pixel-radius particle images of figure 12.4-1, for example, shows some very interesting patterns that are a consequence of the particle image being too small relative to the pixels. Dependency artifacts then increase a bit as the particle image size increases, which can be seen as spikes jutting out from the central distribution in the scatter plots—but as the particle image size increases, so does the magnitude of the error.

The 1.5-pixel-radius mark is also a transition for the behavior of the average x and y errors per image.

Between figures 12.4-3, 12.4-4, and 12.4-5, it can be seen that the average error drops sharply, but that by the time the particle images are 2 pixels in radius, there is a difference between the average error in x and y. At this point, it is impossible to separate the contributions of the optimization itself, the pseudo-random position of the particle images, and the discretized domain. The discretization effect can be clearly seen in the scatter plot of figure 12.4-6.

Note also that the average error remains relatively constant for the small particle images, but in figure 12.4-5 it is clear that it decreases as the images are moved closer and closer to the apertures. This shows that at this particle image size, particle image overlap is already affecting the quality of the fit. The translation tests plot of section 12.3 is a more pure measurement of the fitting accuracy

Figure 12.4-2: Scatter plot of the error in the sub-pixel coordinate of the center for at most 15,000 recovered particle images for the case of no noise in the image and particle images of radius 4.0.

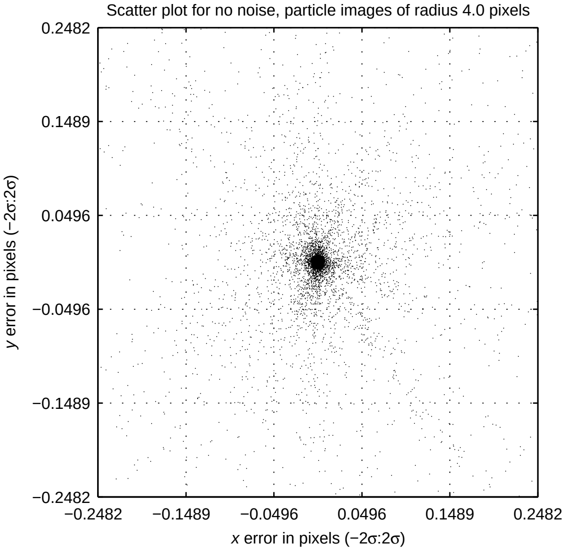


Figure 12.4-3: Plot of average magnitude of the pixel error in the sub-pixel coordinate of the center of recovered particle images for the case of no noise in the image and particle images of radius 1.0.

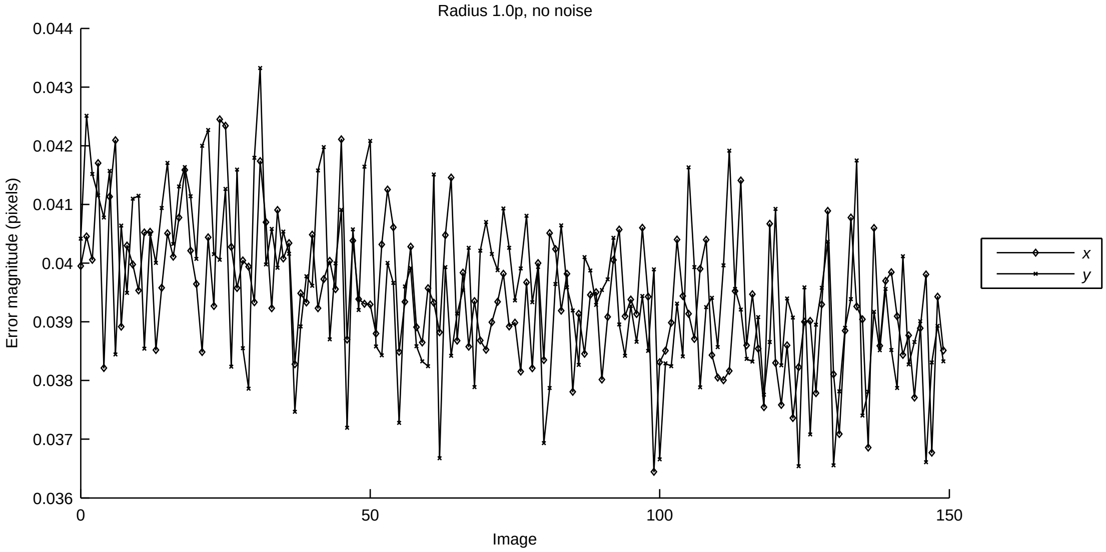


Figure 12.4-4: Plot of average magnitude of the pixel error in the sub-pixel coordinate of the center of recovered particle images for the case of no noise in the image and particle images of radius 1.5.

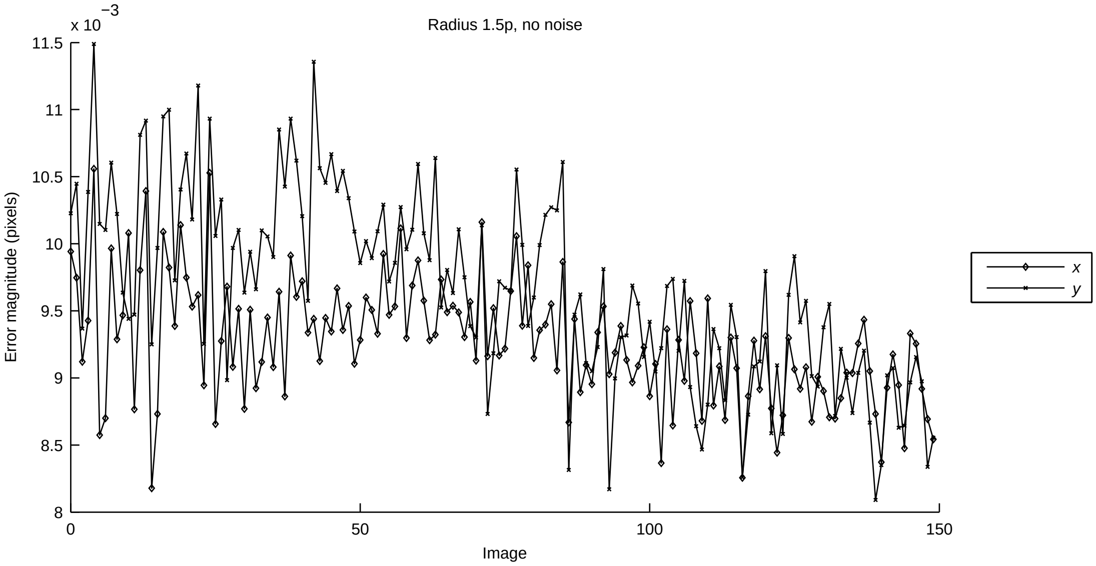


Figure 12.4-5: Plot of average magnitude of the pixel error in the sub-pixel coordinate of the center of recovered particle images for the case of no noise in the image and particle images of radius 2.0.

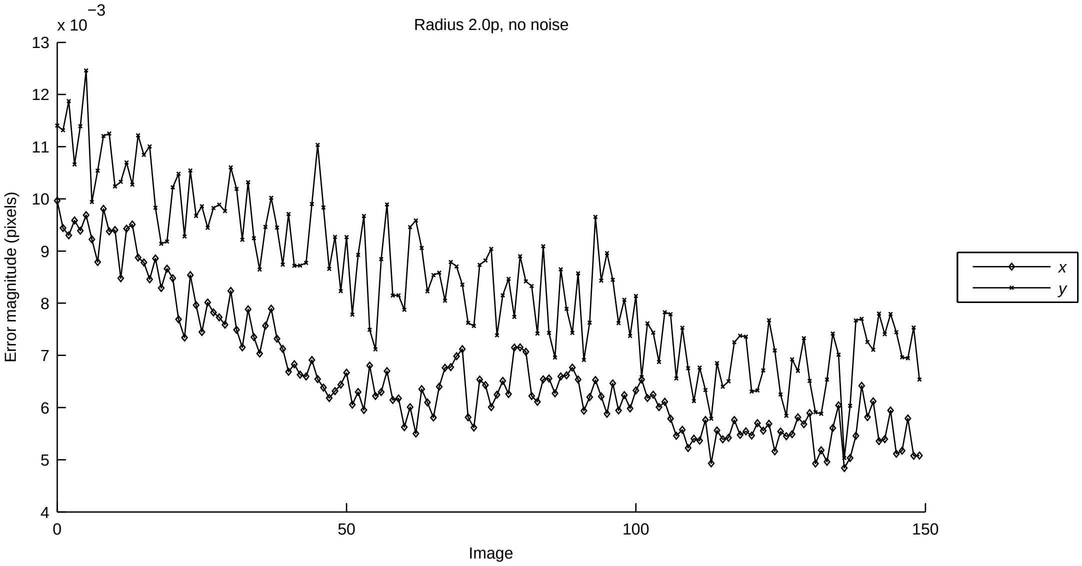
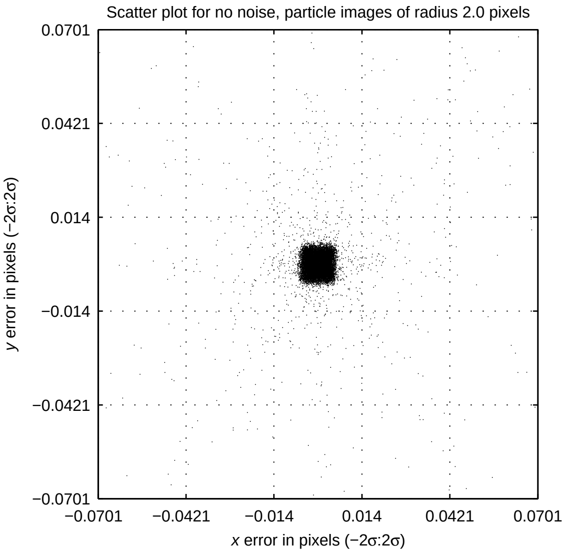


Figure 12.4-6: Scatter plot of the error in the sub-pixel coordinate of the center for at most 15,000 recovered particle images for the case of no noise in the image and particle images of radius 2.0.



because there is no overlap in those particle images. The data presented in this section through the scatter plots and average error plots, however, is more realistic, since they more closely approximate experimental situations, so the evidence presented by those plots can be more readily applied to estimating experimental conditions.

The precision of the Gaussian fit is not the only performance benchmark. As shown in figure 12.4-7, the 0.5-pixel-radius particle image is completely inadequate since a substantial portion of them are lost. Moreover the histograms of figure 12.3-2 show that the error for those that are recovered is substantial, which should be expected since no sub-pixel accuracy is possible with particle images that do not affect more than a single pixel.

On the other end of the spectrum, the large particle images take up so much room that the overlap becomes a considerable problem, as shown in figure 12.4-8.

Adding some slight noise to the image greatly affects the results. With magnitude 5 noise, the histogram spreads over a width nearly 8 times larger than the case with no noise, and any bias due to discretization is overwhelmed, as is obvious from comparison of figures 12.4-6 and 12.4-10.

The evidence can also be seen in the results of the systematic translation test of figures 12.3-1 and 12.4-11.

For each increase in the magnitude of the noise, the error distribution increases in width by the same factor. The increased noise also diminishes the number of correctly identified particle images. Not only do extra particle images begin to appear, but they contribute to the error of properly identified particle images in two ways: first, the fit is obviously affected, though if the particle image is large enough, the effect should be minimized. But there is also added uncertainty in identifying

Figure 12.4-7: Population statistics for recovered particle images for the case of no noise in the image and particle images of radius 0.5.

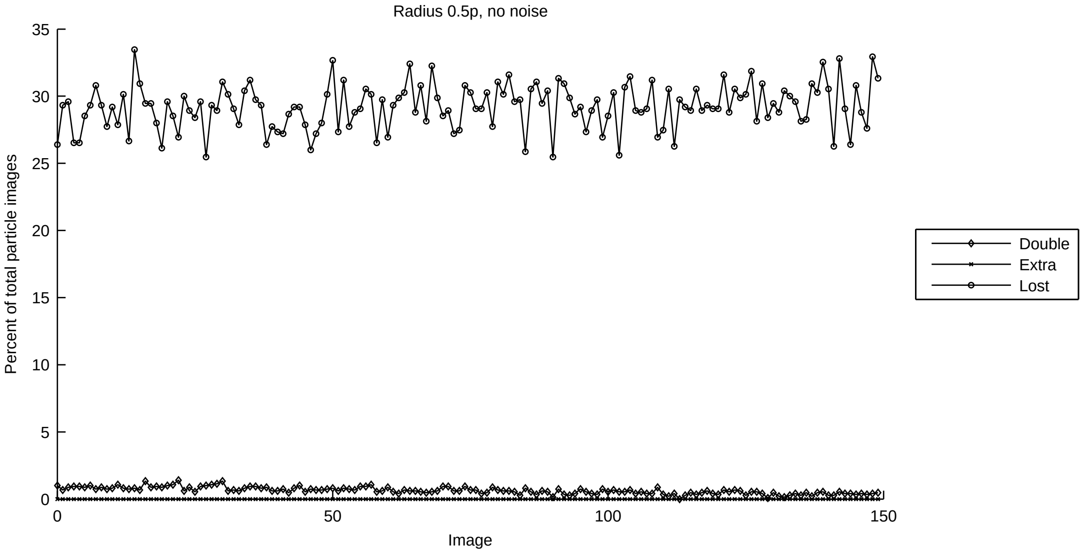


Figure 12.4-8: Population statistics for recovered particle images for the case of no noise in the image and particle images of radius 6.0.

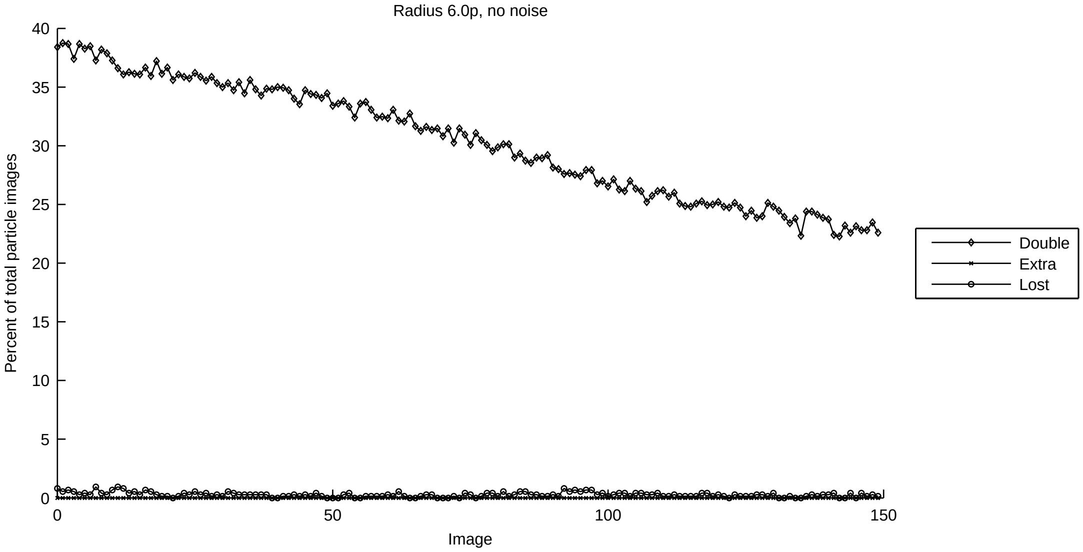


Figure 12.4-9: Histogram of the pixel error in the x coordinate of the sub-pixel coordinate of the center of recovered particle images for the case of noise of magnitude 5 in the image.

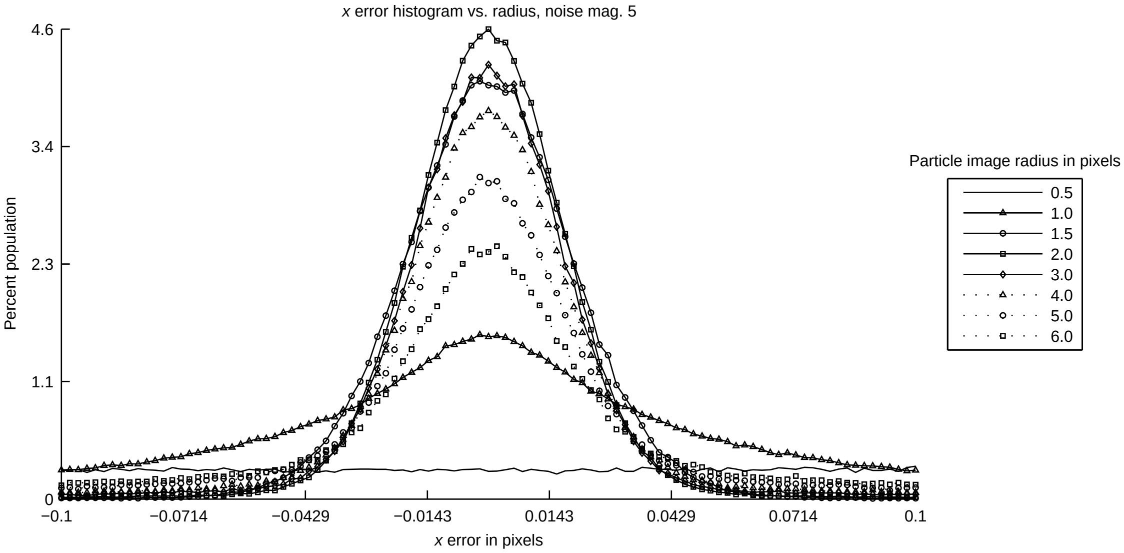


Figure 12.4-10: Scatter plot of the error in the sub-pixel coordinate of the center for at most 15,000 recovered particle images for the case of noise of magnitude 5 in the image and particle images of radius 2.0.

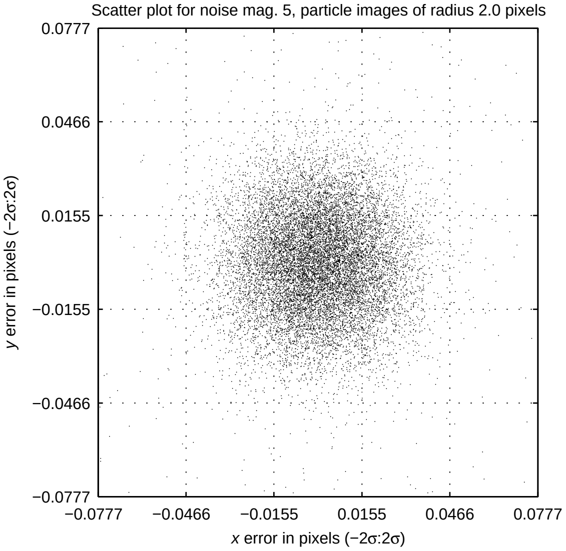
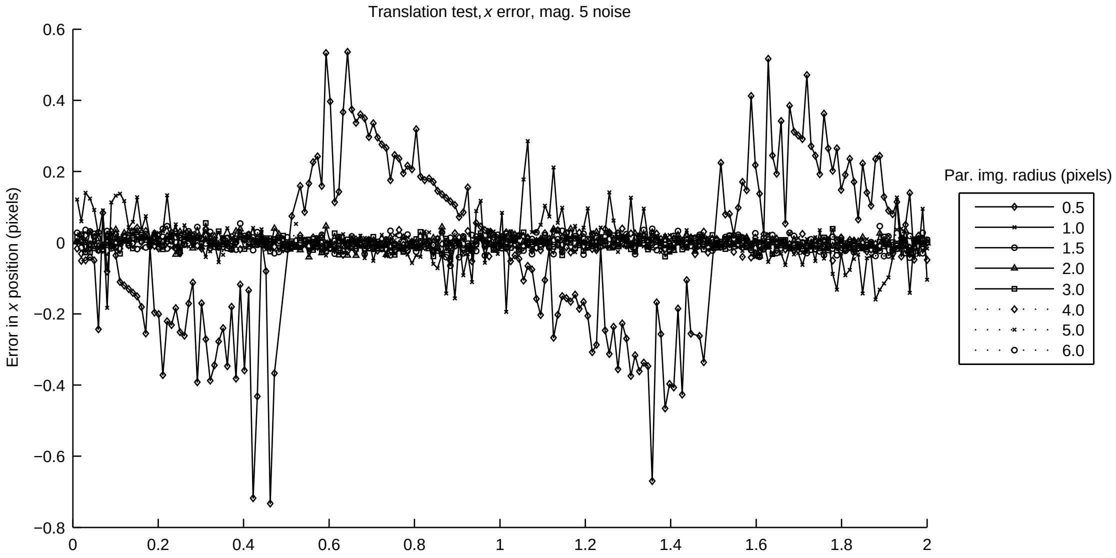


Figure 12.4-11: Error in the x coordinate of the sub-pixel center of the Gaussian fit in a systematic translation of 1/100 of a pixel per step for images with noise of magnitude 5.



the potential particle image peaks, about which the kernel for fitting is centered. The cumulative result is that at extremely high noise levels, there is little advantage to particle image size from the point of view of fit accuracy, as can be seen in figure 12.4-12. (Note that a noise of magnitude 50 is well beyond what can be achieved during a real experiment.) There is, however, still an advantage from the point of view of the number of particle images recovered, as seen in figure 12.4-13.

Figure 12.4-12: Histogram of the pixel error in the x coordinate of the sub-pixel coordinate of the center of recovered particle images for the case of noise of magnitude 50 in the image.

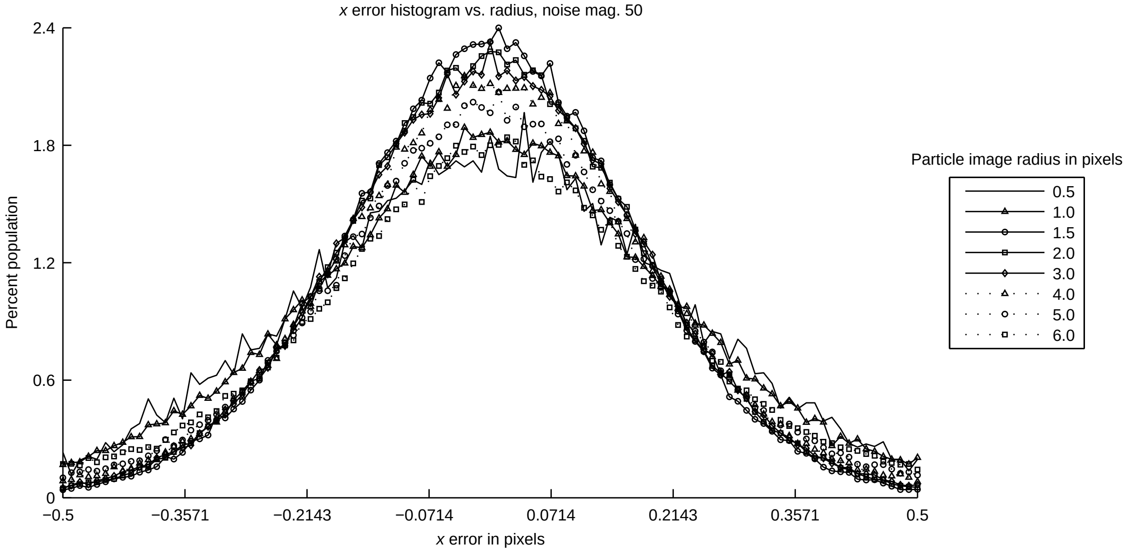
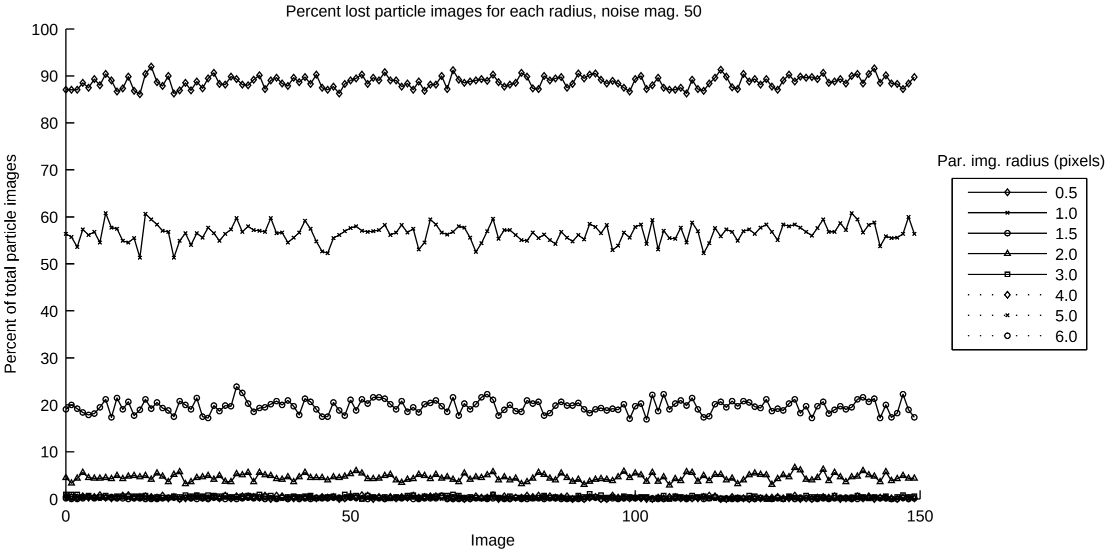


Figure 12.4-13: Percent lost particle images for the case of noise of magnitude 50 in the image for particle images of each radius.



As will be seen in section 12.5, the susceptibility of the accuracy to noise is much lower than that to image intensity. Between perfect images (figure 12.4-14) and those with absurd amounts of noise (figure 12.4-15), the 99th percentile grows by less than 50% for most particle image radii, though, as mentioned above, the increase in noise decreases the difference between the performance at different particle image radii.

Figure 12.4-14: Percentile of the radial pixel error magnitude in the sub-pixel coordinate of the center of recovered particle images for the case of no noise in the image.

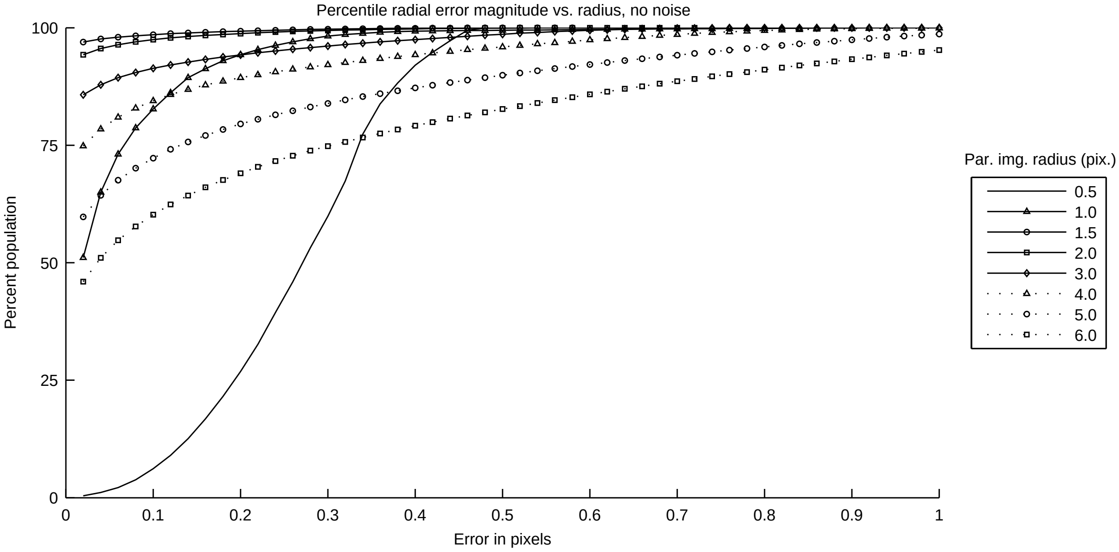
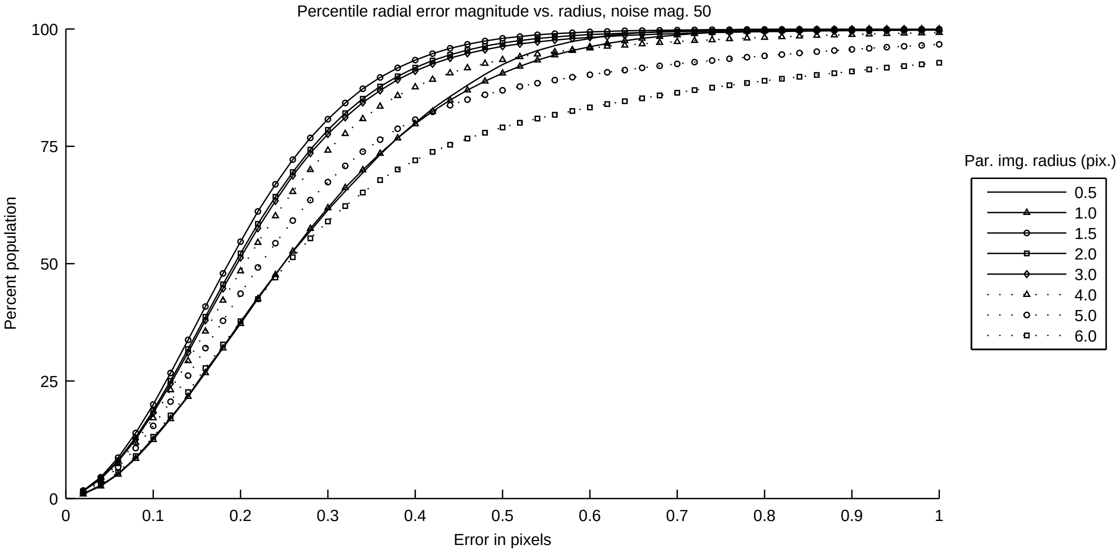


Figure 12.4-15: Percentile of the radial pixel error magnitude in the sub-pixel coordinate of the center of recovered particle images for the case of noise of magnitude 50 in the image.



12.5 Effect of Particle Image Intensity and Radius

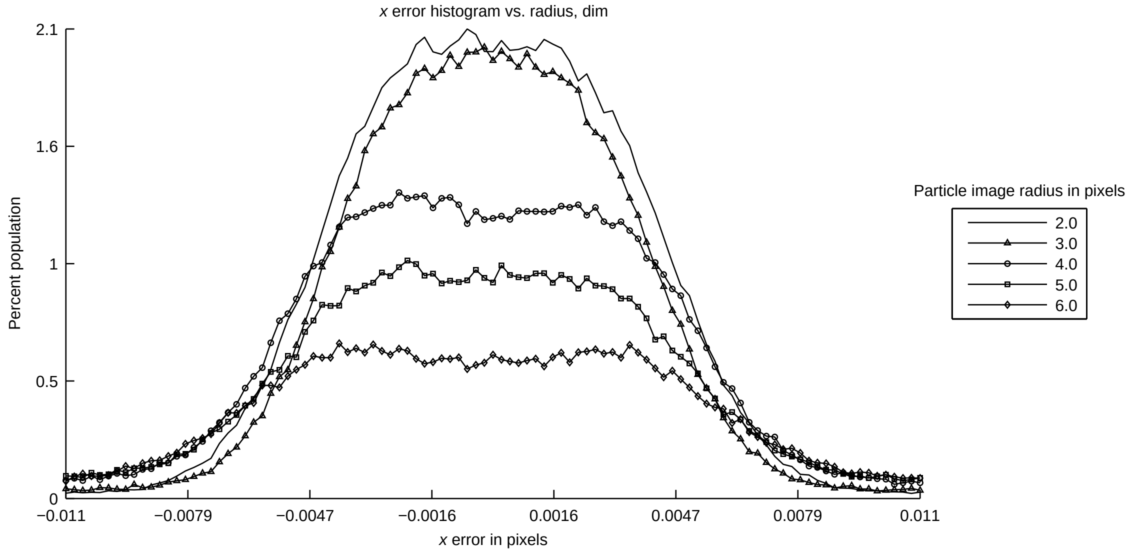
Four brightnesses were tested. For the “really dim” case the particle images were no brighter than a pixel value of 53; for “dim” the maximum is 211; in “bright” particle images contained up to 20% saturated pixels (so maximum brightness is 255); in “saturated” several pixels per particle image had a value of 255. The particle images used in section 12.4 with no image noise fall between “dim” and “bright”. For this case, the particle image radii of 0.5, 1.0, and 1.5 were not evaluated, since they were not visible in the dimmest setting.

For reference, enlarged samples of the images generated can be seen in section A.2 of Graff [2007d].

The first thing to notice is that the result for a dim particle image is not equivalent to the perfect particle image of smaller diameter. This can be seen by comparing figures 12.3-2 and 12.5-1 which have the same number of bins and bin dimensions. The case of dim particle images performs much worse, even though the peak intensity per particle image is on average only 25% lower. This indicates that the fitting sequence is very sensitive to the amount of slope in the Gaussian particle image.

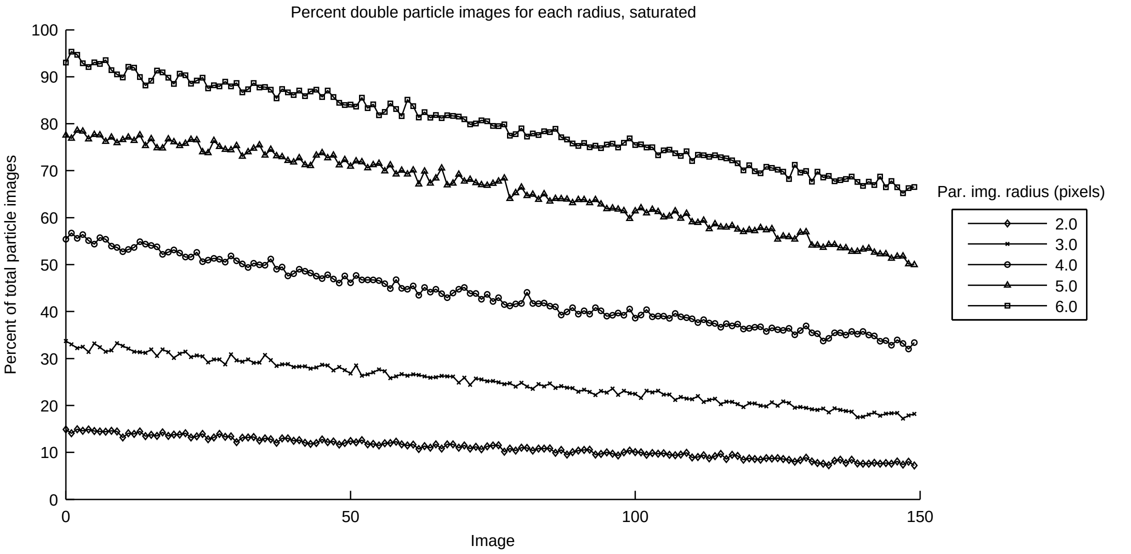
It would be expected, then, that the algorithm would fare well in fitting over saturated particle images, which is extremely important because it is often difficult to control the seeding and the illumination in such a way to arrive at a constant particle image intensity throughout the volume, and evidently dim particle images perform quite badly. The difficulty arrives in the identification of the potential particle images during thresholding. Figure 12.5-2 confirms that with the saturated

Figure 12.5-1: Histogram of the pixel error in the x coordinate of the sub-pixel coordinate of the center of recovered particle images for the case of dim particle images.



particle images, the Gaussian fitting kernel is misplaced to begin with—shown by the fact that the number of double particle images increases directly with particle image size. Particle image recovery is otherwise essentially unaffected with respect to intensity.

Figure 12.5-2: Percent double particle images for the case of saturated particle images of each radius.



Comparing the confinement of the error on equal scales, the difference between really dim images and dim images is a two-fold increase in precision (as shown by figures 12.5-3 and 12.5-4). These two cases depict very clearly the dependence of accuracy on the slope—a higher slope yielding

Figure 12.5-3: Percentile of the radial pixel error magnitude in the sub-pixel coordinate of the center of recovered particle images for the case of really dim particle images.

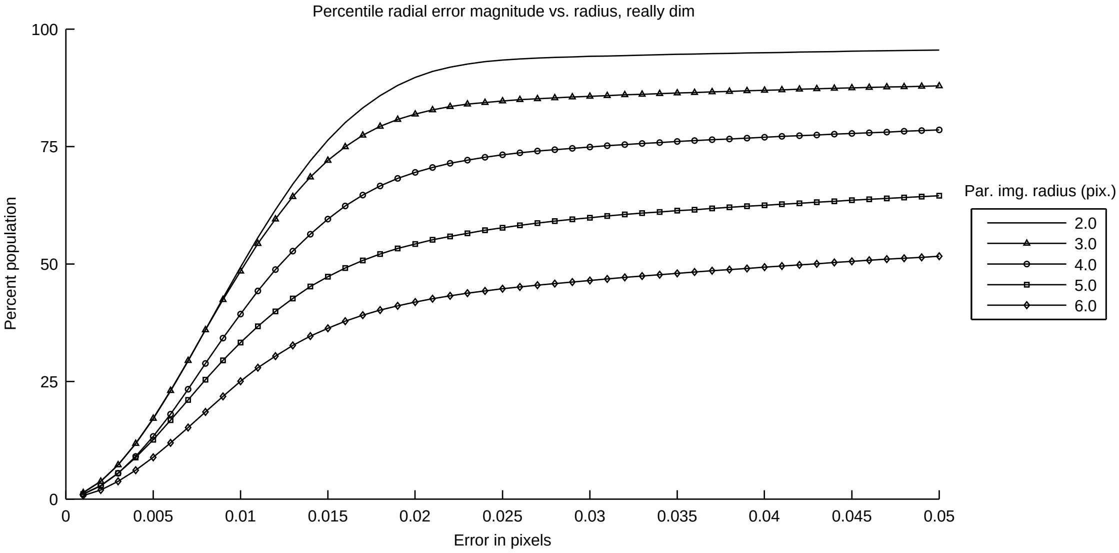


Figure 12.5-4: Percentile of the radial pixel error magnitude in the sub-pixel coordinate of the center of recovered particle images for the case of dim particle images.

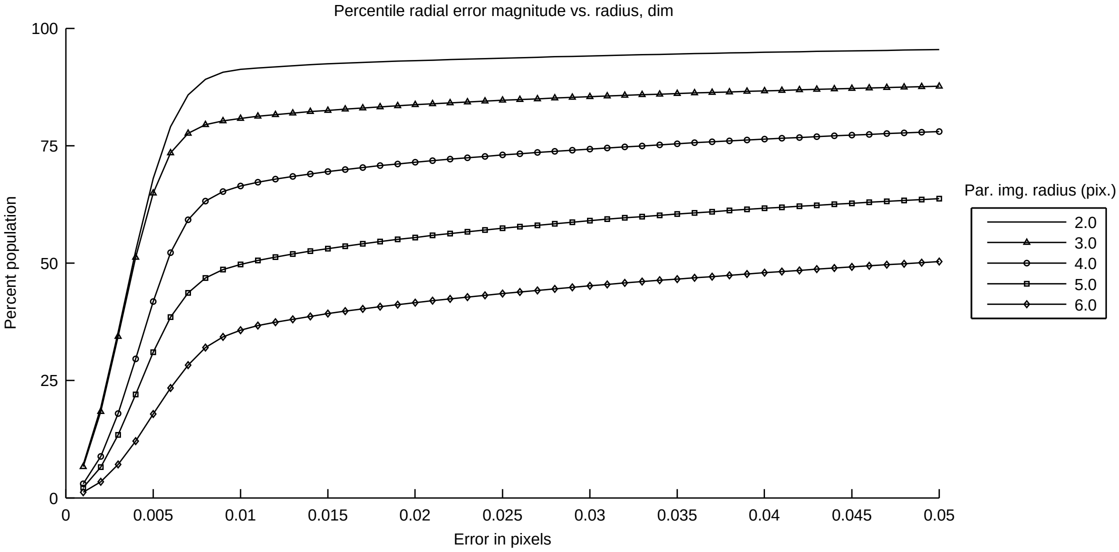
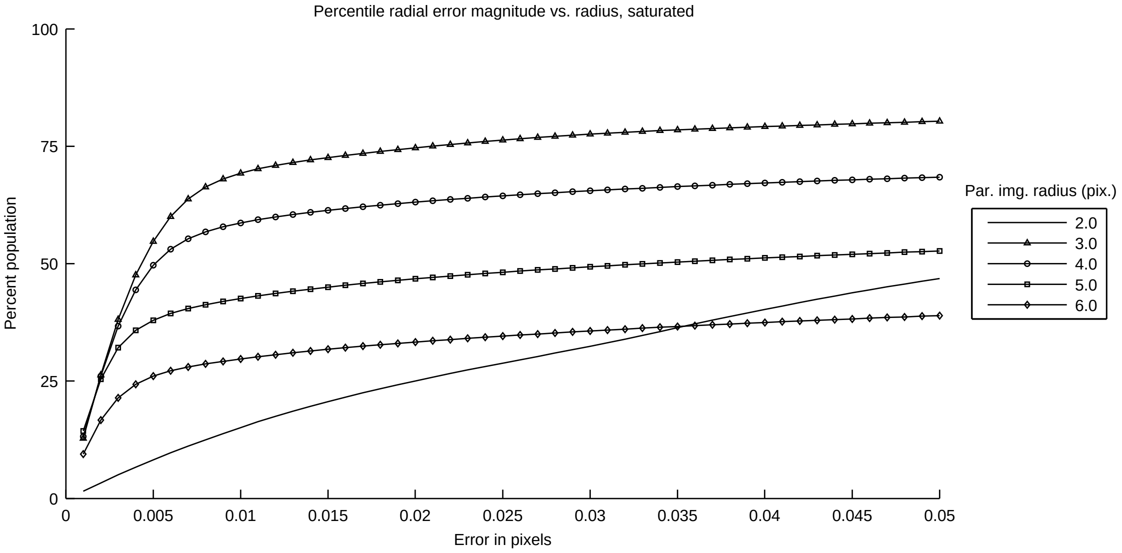


Figure 12.5-5: Percentile of the radial pixel error magnitude in the sub-pixel coordinate of the center of recovered particle images for the case of saturated particle images.



better results—since the precision is very clearly separated by particle image diameter and, since the intensities are constant, a larger diameter indicates a more gradual slope. Saturated particle images (figure 12.5-5) show high accuracy only for those whose kernels had the opportunity to be in the correct place, but none can rival the quality of the fit obtained using perfect particle images (figure 12.4-14).

12.6 Conclusion of Simulations

Simulations show that the Gaussian fitting algorithm is most sensitive to the amount of available sloping pixel values in the particle image. This is shown by a peak in accuracy at 2.0-pixel-radius particle images in perfect conditions, with falloffs both as the particle image size decreases and increases, and a rapid decrease in fitting performance as the intensity drops.

Noise also affects the Gaussian fit accuracy, but, not considering the performance of saturated particle images, has a more pronounced effect on the recoverability of particle images in the extreme, that is, it affects the placement of the kernel in the threshold substep.

Particle image saturation essentially inhibits proper placement of the fitting kernel, which is a direct consequence of the algorithm used and not a measurement limitation by any means; future versions of the software will correct this pitfall.

In balancing recoverability and fitting accuracy, it seems clear that when the choice is available, the user should increase the gain of the sensors, thereby increasing the noise while increasing the particle image brightness until most particle images are just below saturation.

12.6.1 Error Statistics

In section 5.2 it was assumed that the error in x and y was both of equal magnitude and independent (that is, there is no σ_{xy} term. The simulations presented here afford a large sample set and can help to verify this assumption. Table 12.6-1 shows the values of the variance and covariance for the tests of Gaussian fitting accuracy versus noise in the image. The first thing to notice is that, in the perfect case, the lowest variances are at pixel radii of 1.5 and 2.0, but already at this size the covariance is only one order of magnitude smaller than the variance. In the case of the 2.0-pixel radius, the variance in y is nearly an order of magnitude greater than that of x. Indeed the Gaussian fitting process is not one where the error in x and y is independent or the same magnitude. As the noise increases, the data fits more closely the assumptions made in section 5.2.

With respect to particle image brightness (table 12.6-2), the dimmer particle images have variances in the same order of magnitude, but saturation, as seen in the other evidence, destroys any accuracy in the method.

12.7 Evidence of Performance in Real Data

To test the performance of the Gaussian fitting algorithm with real particles, an experiment was setup in still water using fluorescent particles approximately 100 µm in diameter manufactured by Kodak. These particles are matched excellently to be neutrally buoyant in fresh water and will remain suspended for several days if undisturbed.

The tank of water was seeded so that around 12,000 particle images were detected per image. A sequence of 100 images with the water perfectly still was taken with the Emilio Camera, and the results of the Gaussian fitting algorithm were then clumped within the sequence so that given a starting image coordinate (from image 0), any particle image within 0.5 pixels of this point in each subsequent image was considered to be the same particle image in the next frame. In essence, this is a very simple nearest-neighbor tracking algorithm with a very strict search limit (0.5 pixel radius). This radius was chosen because this is the typical matching tolerance used with the Emilio Camera and, moreover, since the particle image kernel was set to 2 pixels in radius, this prevents particles detected overlapping with each other to be counted as a single particle. For each clump or sequence of a single particle image, the error was taken to be the standard deviation of the sequence of coordinates for that clump. Because of the massive number of particle images, the standard deviation was calculating by computing a running estimate of the standard deviation rather than amassing the entire data set first and then calculating the precise deviation. The algorithm used was from Knuth [1998]; it computes the exact mean and an estimated standard deviation in a running fashion according to

Table 12.6-1: Error statistics for image noise vs. particle image radius simulations.

Condition	N	μ_x	μ_y	σ_x^2	σ_y^2	σ_{xy}^2
no noise, 0.5p	78,687	-3.48×10^{-4}	-5.91×10^{-4}	4.22×10^{-2}	4.25×10^{-2}	1.45×10^{-5}
no noise, 1.0p	314,340	-4.83×10^{-5}	1.74×10^{-5}	5.62×10^{-3}	5.54×10^{-3}	-5.71×10^{-6}
no noise, 1.5p	108,970	1.55×10^{-4}	-9.71×10^{-5}	5.44×10^{-4}	6.07×10^{-4}	-2.90×10^{-5}
no noise, 2.0p	106,329	1.44×10^{-4}	7.70×10^{-5}	6.84×10^{-4}	1.24×10^{-3}	-3.62×10^{-5}
no noise, 3.0p	98,038	3.36×10^{-4}	-8.21×10^{-4}	6.26×10^{-3}	5.38×10^{-3}	-1.14×10^{-5}
no noise, 4.0p	88,016	-1.55×10^{-3}	-1.77×10^{-3}	1.47×10^{-2}	1.54×10^{-2}	-8.62×10^{-4}
no noise, 5.0p	77,051	1.32×10^{-3}	5.29×10^{-4}	3.88×10^{-2}	4.51×10^{-2}	2.15×10^{-3}
no noise, 6.0p	65,135	-1.95×10^{-3}	-8.97×10^{-3}	8.12×10^{-2}	8.99×10^{-2}	2.74×10^{-3}
5 noise, 0.5p	79,498	2.28×10^{-4}	-1.05×10^{-3}	5.87×10^{-2}	5.88×10^{-2}	1.77×10^{-4}
5 noise, 1.0p	315,070	4.05×10^{-5}	6.40×10^{-5}	1.56×10^{-2}	1.54×10^{-2}	-2.00×10^{-5}
5 noise, 1.5p	108,972	2.20×10^{-4}	-9.85×10^{-7}	8.66×10^{-4}	9.24×10^{-4}	-3.57×10^{-5}
5 noise, 2.0p	106,313	1.32×10^{-4}	1.79×10^{-4}	9.48×10^{-4}	1.51×10^{-3}	-4.75×10^{-5}
5 noise, 3.0p	98,018	4.11×10^{-4}	-7.96×10^{-4}	6.53×10^{-3}	5.68×10^{-3}	9.48×10^{-6}
5 noise, 4.0p	87,987	-1.45×10^{-3}	-1.74×10^{-3}	1.48×10^{-2}	1.58×10^{-2}	-5.05×10^{-4}
5 noise, 5.0p	77,019	1.68×10^{-3}	4.47×10^{-4}	3.95×10^{-2}	4.69×10^{-2}	2.01×10^{-3}
5 noise, 6.0p	65,116	-2.59×10^{-3}	-9.29×10^{-3}	7.81×10^{-2}	9.16×10^{-2}	-2.83×10^{-4}
10 noise, 0.5p	52,040	7.50×10^{-4}	5.57×10^{-4}	5.27×10^{-2}	5.34×10^{-2}	5.24×10^{-5}
10 noise, 1.0p	205,697	-7.76×10^{-5}	1.69×10^{-4}	1.09×10^{-2}	1.07×10^{-2}	7.88×10^{-6}
10 noise, 1.5p	108,969	2.35×10^{-4}	8.34×10^{-5}	1.72×10^{-3}	1.77×10^{-3}	-4.41×10^{-5}
10 noise, 2.0p	106,290	1.82×10^{-4}	2.93×10^{-4}	1.74×10^{-3}	2.29×10^{-3}	-5.74×10^{-5}
10 noise, 3.0p	97,980	4.62×10^{-4}	-7.85×10^{-4}	7.41×10^{-3}	6.58×10^{-3}	-8.54×10^{-7}
10 noise, 4.0p	87,994	-1.32×10^{-3}	-1.83×10^{-3}	1.60×10^{-2}	1.71×10^{-2}	-4.88×10^{-4}
10 noise, 5.0p	76,945	2.21×10^{-3}	-7.39×10^{-5}	4.16×10^{-2}	4.87×10^{-2}	1.79×10^{-3}
10 noise, 6.0p	65,106	-1.82×10^{-3}	-1.06×10^{-2}	8.23×10^{-2}	9.63×10^{-2}	-6.09×10^{-4}
20 noise, 0.5p	37,731	1.07×10^{-3}	1.83×10^{-3}	5.36×10^{-2}	5.37×10^{-2}	-1.57×10^{-4}
20 noise, 1.0p	147,634	1.37×10^{-4}	2.24×10^{-4}	1.98×10^{-2}	2.00×10^{-2}	-7.65×10^{-5}
20 noise, 1.5p	108,970	3.84×10^{-4}	1.89×10^{-4}	5.15×10^{-3}	5.18×10^{-3}	-7.50×10^{-5}
20 noise, 2.0p	106,315	4.05×10^{-4}	4.31×10^{-4}	5.13×10^{-3}	5.76×10^{-3}	-8.10×10^{-5}
20 noise, 3.0p	97,903	5.30×10^{-4}	-6.80×10^{-4}	1.07×10^{-2}	9.77×10^{-3}	-9.48×10^{-5}
20 noise, 4.0p	87,920	-8.89×10^{-4}	-2.26×10^{-3}	1.95×10^{-2}	2.09×10^{-2}	-5.73×10^{-4}
20 noise, 5.0p	76,765	3.10×10^{-3}	-2.11×10^{-3}	4.65×10^{-2}	5.30×10^{-2}	1.68×10^{-3}
20 noise, 6.0p	64,993	4.36×10^{-4}	-1.62×10^{-2}	9.13×10^{-2}	1.05×10^{-1}	1.01×10^{-3}
50 noise, 0.5p	12,510	-6.69×10^{-4}	1.51×10^{-3}	4.97×10^{-2}	5.09×10^{-2}	6.19×10^{-5}
50 noise, 1.0p	48,462	1.22×10^{-4}	-2.34×10^{-4}	5.25×10^{-2}	5.24×10^{-2}	-4.15×10^{-4}
50 noise, 1.5p	87,309	7.01×10^{-5}	8.75×10^{-4}	3.04×10^{-2}	3.04×10^{-2}	-1.06×10^{-4}
50 noise, 2.0p	101,999	5.98×10^{-4}	9.53×10^{-4}	3.44×10^{-2}	3.48×10^{-2}	-9.25×10^{-5}
50 noise, 3.0p	97,779	2.35×10^{-3}	-2.54×10^{-3}	3.96×10^{-2}	3.81×10^{-2}	-4.82×10^{-4}
50 noise, 4.0p	87,799	3.69×10^{-3}	-1.01×10^{-2}	5.06×10^{-2}	5.51×10^{-2}	-1.63×10^{-4}
50 noise, 5.0p	76,570	1.12×10^{-2}	-2.44×10^{-2}	8.88×10^{-2}	9.79×10^{-2}	4.10×10^{-3}
50 noise, 6.0p	64,730	1.31×10^{-2}	-4.93×10^{-2}	1.46×10^{-1}	1.57×10^{-1}	3.65×10^{-4}

Table 12.6-2: Error statistics for particle image brightness vs. particle image radius simulations.

Condition	N	μ_x	μ_y	σ_x^2	σ_y^2	σ_{xy}^2
really dim, 2.0p	106,311	1.42×10^{-4}	8.50×10^{-5}	7.54×10^{-4}	1.33×10^{-3}	-3.97×10^{-5}
really dim, 3.0p	97,991	4.57×10^{-4}	-7.93×10^{-4}	6.45×10^{-3}	5.58×10^{-3}	-2.95×10^{-5}
really dim, 4.0p	87,904	-1.52×10^{-3}	-2.14×10^{-3}	1.43×10^{-2}	1.59×10^{-2}	-6.79×10^{-4}
really dim, 5.0p	76,744	2.56×10^{-3}	-9.38×10^{-4}	3.80×10^{-2}	4.54×10^{-2}	1.38×10^{-3}
really dim, 6.0p	64,825	-9.17×10^{-5}	-1.36×10^{-2}	7.32×10^{-2}	8.96×10^{-2}	-2.76×10^{-3}
dim, 2.0p	106,317	1.38×10^{-4}	7.29×10^{-5}	7.05×10^{-4}	1.27×10^{-3}	-3.77×10^{-5}
dim, 3.0p	98,034	3.53×10^{-4}	-8.02×10^{-4}	6.39×10^{-3}	5.53×10^{-3}	-1.23×10^{-5}
dim, 4.0p	87,948	-1.36×10^{-3}	-2.03×10^{-3}	1.40×10^{-2}	1.56×10^{-2}	-7.34×10^{-4}
dim, 5.0p	76,881	1.75×10^{-3}	1.01×10^{-4}	3.71×10^{-2}	4.49×10^{-2}	1.30×10^{-3}
dim, 6.0p	64,926	-2.60×10^{-3}	-9.91×10^{-3}	7.14×10^{-2}	8.63×10^{-2}	-3.10×10^{-3}
bright, 2.0p	106,204	-3.45×10^{-4}	7.71×10^{-5}	1.67×10^{-3}	2.58×10^{-3}	-9.52×10^{-5}
bright, 3.0p	97,608	7.76×10^{-4}	-6.01×10^{-4}	1.26×10^{-2}	1.09×10^{-2}	1.21×10^{-3}
bright, 4.0p	87,829	-1.87×10^{-3}	-5.16×10^{-3}	3.66×10^{-2}	4.24×10^{-2}	1.07×10^{-3}
bright, 5.0p	77,517	5.06×10^{-3}	-4.77×10^{-3}	1.17×10^{-1}	1.47×10^{-1}	2.01×10^{-2}
bright, 6.0p	65,776	1.05×10^{-2}	-1.72×10^{-2}	2.76×10^{-1}	3.10×10^{-1}	2.57×10^{-2}
saturated, 2.0p	274,811	-7.19×10^{-3}	2.47×10^{-5}	3.31×10^{-2}	3.55×10^{-2}	8.27×10^{-4}
saturated, 3.0p	287,827	2.78×10^{-3}	-1.77×10^{-3}	1.05×10^{-1}	9.73×10^{-2}	8.16×10^{-4}
saturated, 4.0p	263,368	6.32×10^{-3}	-1.29×10^{-2}	3.35×10^{-1}	3.02×10^{-1}	1.27×10^{-2}
saturated, 5.0p	234,590	2.63×10^{-2}	-3.93×10^{-2}	8.35×10^{-1}	7.51×10^{-1}	5.25×10^{-2}
saturated, 6.0p	194,389	3.45×10^{-2}	-6.29×10^{-2}	1.75	1.44	9.37×10^{-2}

$$\begin{aligned} \mu_{\text{new}} &= \mu_{\text{old}} + \frac{x_{\text{new}} - \mu_{\text{old}}}{n+1} \\ s^2_{\text{new}} &= \frac{(n-1)\, s^2_{\text{old}} + (x_{\text{new}} - \mu_{\text{new}})\,(x_{\text{new}} - \mu_{\text{old}})}{n} \end{aligned} \tag{12.7-1}$$

where x_{new} is the current value being added to the sequence.

The first thing to notice in figures 12.7-1 and 12.7-2 is the fact that the error distribution is different for x and y. As had been observed in section 12.4, the Gaussian fitting algorithm seems to perform differently in one direction than the other. Still, the upper bound of the percentile is nearly equal, so it is not the extremes of the error that are different, but rather the distribution. In this real case, taking this x and y error difference too literally is dangerous because, even though the flow was completely quiescent, the laser beam does change the temperature of the water and slight vertical movement can be observed in some particles. The 0.5-pixel-radius matching area limits this, but if there were movement in some particles, it would certainly contribute to the non-zero peak in the histogram of the y error, like the one in figure 12.7-3.

The second thing to notice is that the error is well controlled. The error percentile is within that of the case of really dim particle images and also that of the case of magnitude 10 image noise. From the sample images of figures 12.7-4 and 12.7-5 it is clear that the noise level of the image is

Figure 12.7-1: Percentile plot of the pixel error in the x coordinate of the sub-pixel center of recovered particle images from the Emilio Camera's "B" aperture using Kodak 100 µm fluorescent particles.

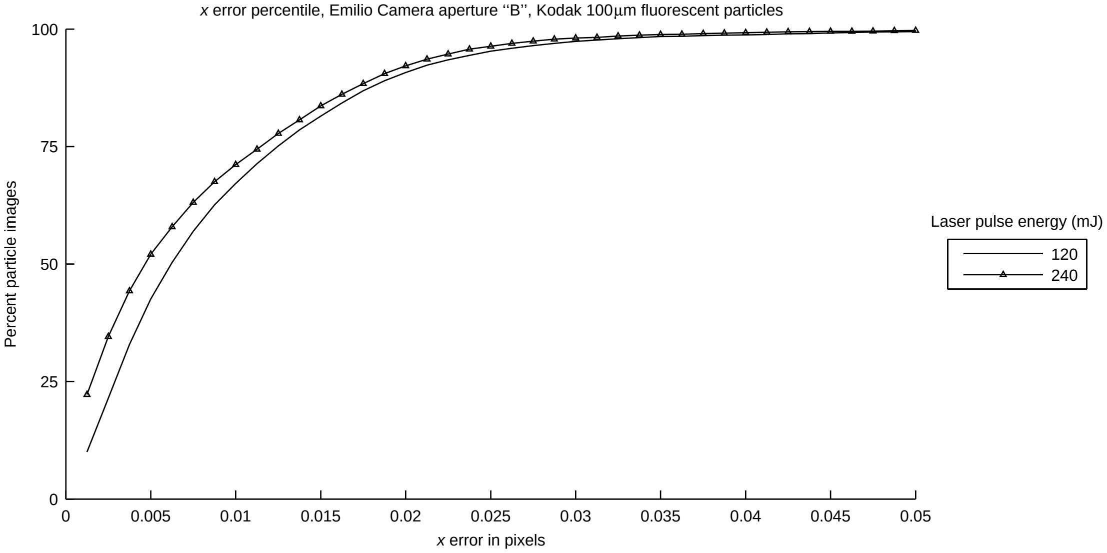


Figure 12.7-2: Percentile plot of the pixel error in the y coordinate of the sub-pixel center of recovered particle images from the Emilio Camera's "B" aperture using Kodak 100 µm fluorescent particles.

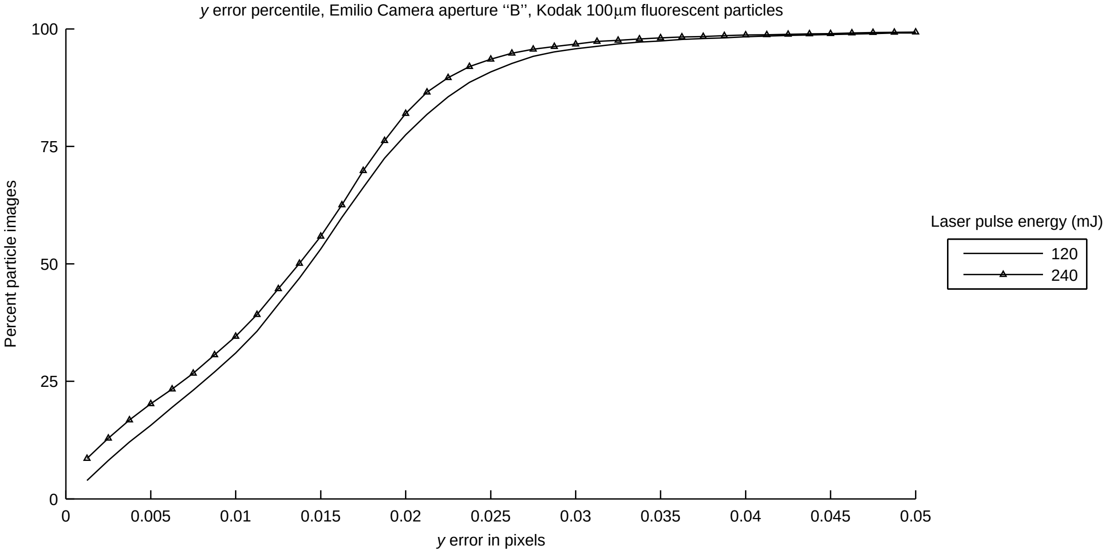
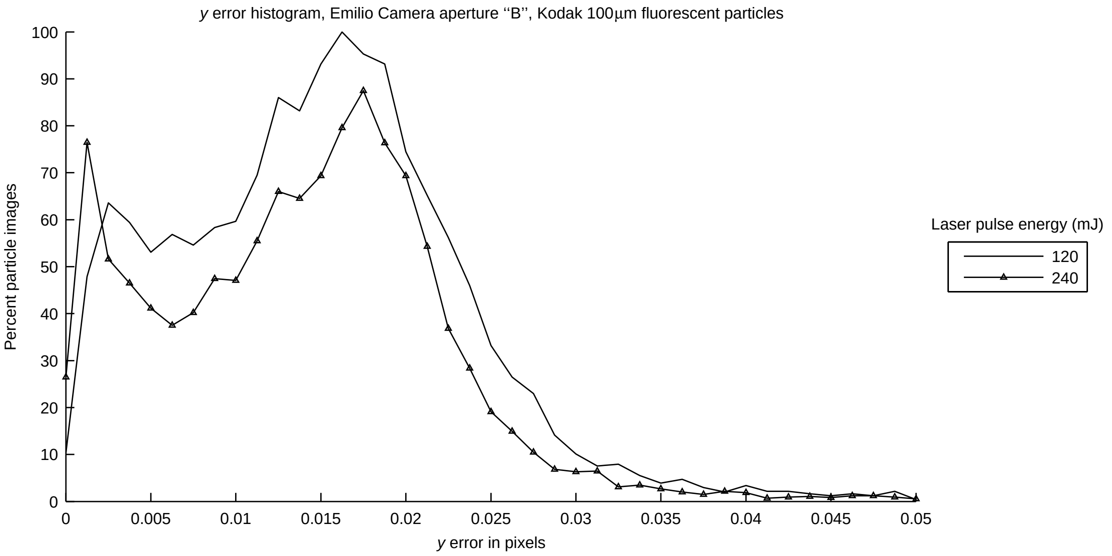


Figure 12.7-3: Histogram of the pixel error in the *y* coordinate of the sub-pixel center of recovered particle images from the Emilio Camera's "B" aperture using Kodak 100 µm fluorescent particles.



very low, but, as expected, there is a considerable range of brightnesses.

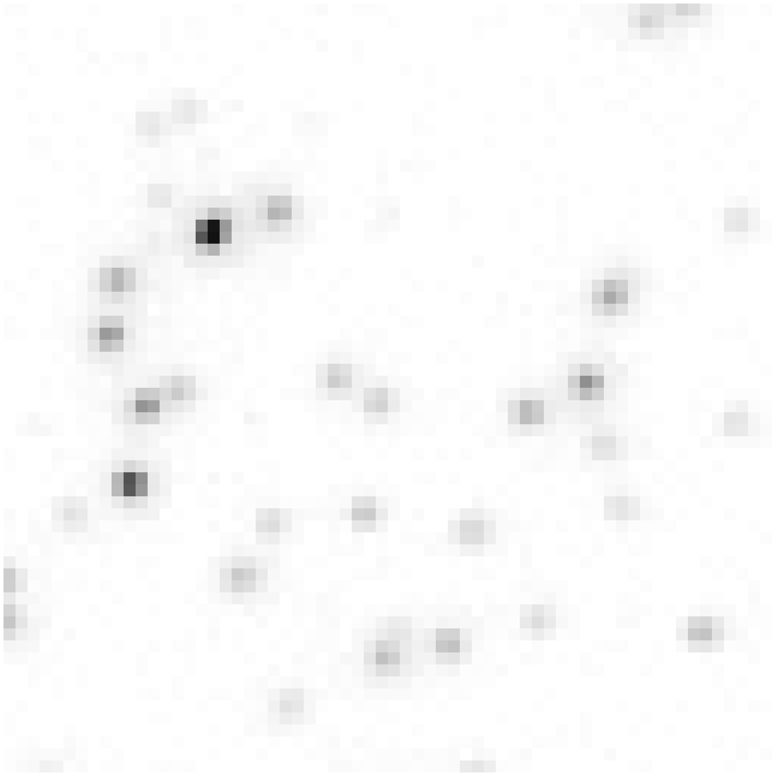
In real situations it is, of course, impossible to separate the error induced by the particles themselves and that of the algorithm. Even in this test scenario, performed with particles that are neutrally buoyant, one cannot draw too strong a conclusion about the results since it is impossible to keep the particles from moving. In most cases, particles are not neutrally buoyant, and performing such a tests is difficult, if not impossible. Particles tend to not be spherical, and thus even movement of the particle while in-place relative to the tank could affect the performance of the fit as the brightness changes. As can be seen from the sample images, it is also impossible to obtain a uniform brightness, both because of the size distribution of the particles and because the Nd:YAG lasers used sacrifice beam quality for intensity and contain "hot" and "cold" spots.

In real flow situations, the intensity variations within the beam can be quite problematic because particles may be visible in one frame and nearly disappear in the next as they travel from hot spot to cold spot. As particle size increases, this effect is alleviated, which can be seen that the number of particles in the reconstruction (as a percentage of the average number of particle images in the apertures) increases as the size increases. For example, with the Ian Camera, using near-millimeter-diameter bubbles for seeding and illuminating in a forward-scatter arrangement, yields of 85% are common, where as typical results of the Emilio Camera with the Kodak particles peaked at 70%. This is also due to the fact that if a particle is not spherical (and the Kodak ones are *not* perfectly so) then it is possible that its brightness is so much a function of the observation angle that at instants it may disappear in one or more apertures—one can imagine a faceteded, n-hedron-like shape, where the blue and green apertures both receive light reflected off a particular facet but it is quite possible

Figure 12.7-4: Enlarged section of one image from the sequence recorded with the Emilio Camera illuminated with a single 120 mJ laser pulse.

that the red aperture is entirely in a "shadow" of the particle. Typically experiments are illuminated with the laser at 90° to the optical axis; problems of this sort could be alleviated by putting the laser more in a back-scatter arrangement but usually this proves difficult in multi-medium experiments because of reflections off the tank walls. The optical limitations of DDPIV make it impossible to illuminate from the center of the camera, which would be the ideal case if the angle between the sensors and the optical axis could be made large enough.

Figure 12.7-5: Enlarged section of one image from the sequence recorded with the Emilio Camera illuminated with a signle 240 mJ laser pulse.



Chapter 13

Multi-plane Dewarping

13.1 Introduction

Once the Gaussian fitting is complete, before the particle image matching can occur the locations of the particle images must be corrected using multi-plane dewarping. In this chapter, the effects of different types of error and distortion are analyzed.

These are primarily a test of the ability of the dewarping coefficients in equation 10.2-1 to converge since it was not possible to generate simulations using lens raytracing.

13.2 Simulation Details

A camera with the same parameters as the Ian Camera (table 7.4-9) was simulated and all rays traced according to pinhole optics. 200 dewarping grids with a dot spacing of 5 mm were generated starting at the reference plane and traversing 1 mm in Z at a time. Appendix A.1 in Graff [2007c] contains samples of how the dewarping grids looked after the induced errors.

The point cloud used to test the resulting dewarping is the one of figure B.1-1 in Graff [2007b], traversed 1 mm at a time to cover a depth of 150mm as in the tests of the Gaussian fit. No images were formed, that is, the point cloud was input as a list of x, y particle image coordinates, thus completely bypassing any behavior due to Gaussian fitting. The dewarping was applied using all planes, skipping 5, skipping 20, and in single-plane fashion. The point cloud reconstruction (particle image matching) was done with a fine tolerance of 0.75 pixels, which was typical for experiments done with the Ian Camera.

To analyze the results, a `MATLAB` script traversed through all the particles that were found and linked them to the closest real particle, deeming the distance between them the error in the reconstruction. Ghost particles were automatically detected during processing with `DDPIV`.

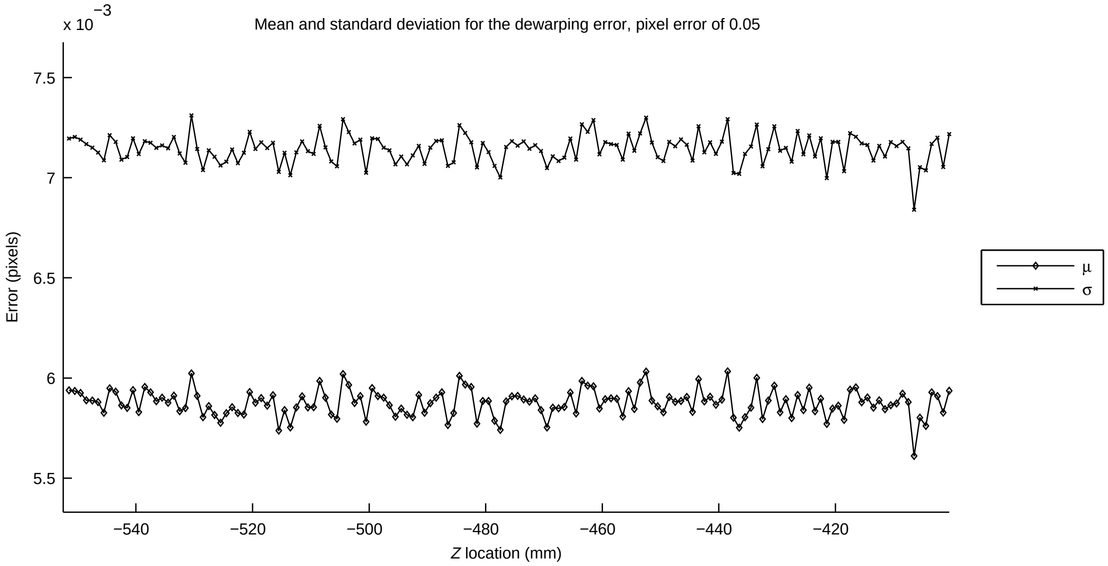
All in-plane errors are characterized by the error in X alone; it should be understood that in the simulated cases not involving bitmapped images, this is perfectly equivalent to the Y error.

13.3 Effect of Pseudo-Random Noise in the Particle Image Coordinates

The first test is to add pseudo-random noise to the particle image coordinates of otherwise perfect pinhole-optics images. This is one way to include error that would originate in the Gaussian fitting step, but tests the ability of the dewarping coefficients to converge properly without any global image distortion. Noise magnitudes of 0.05, 0.10, 0.20, 0.50, and 1.00 pixels were tested.

The first indication of the quality of a dewarping set is the standard deviation of the error between the actual fit of the dewarped grid and the perfect grid. In experiments, a high standard deviation is a sure indication of bad convergence at that specific plane. This quantity scales linearly with the noise in the particle image coordinates.

Figure 13.3-1: Average and standard deviation for the pixel error in the dewarping mapping for dewarping done with pixel error of 0.05.



The X and Z errors in the resulting reconstruction also scales linearly with the magnitude of the noise. The bottom of the plots in figures 13.3-3 and 13.3-4 show the locations of the planes used for each case. The error is independent of the number of planes skipped, which indicates that the dewarping coefficients successfully smooth out the noise (the resulting fit is independent of the noise). This correlates well with the linear relationship between noise and dewarping error, since if the dewarping fits are all perfect, then the dewarping error is a measure of the noise.

The number of ghosts and recovered particles is independent of the noise, indicating that this particle layout is not dense enough that noise in the particle image coordinates would increase the number of ghosts. Once the noise exceeds the tolerance of 0.75 pixels, the number of real particles found decreases, as expected. If the error is normally distributed, one can guess that only about 75%

Figure 13.3-2: Average and standard deviation for the pixel error in the dewarping mapping for dewarping done with pixel error of 0.50.

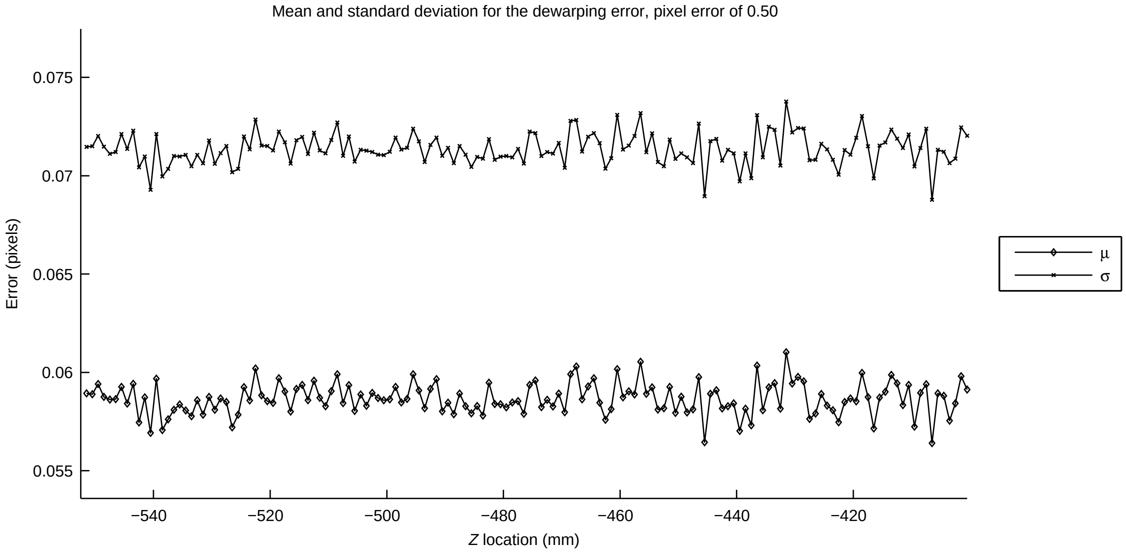


Figure 13.3-3: Error in X for dewarping done with pixel error of 0.05. The bottom plot shows the actual locations of the dewarping planes.

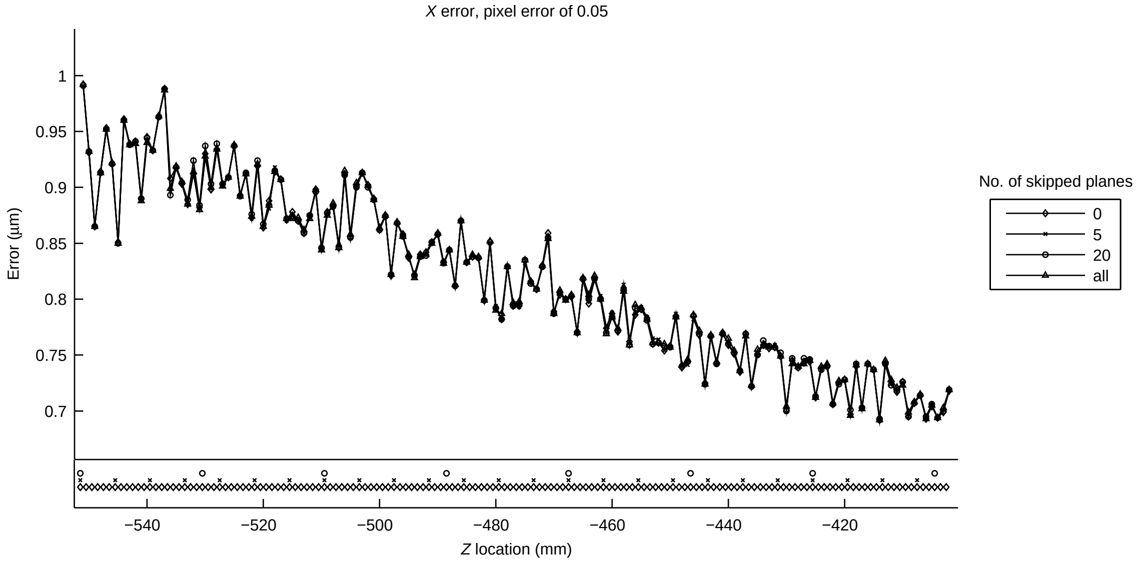
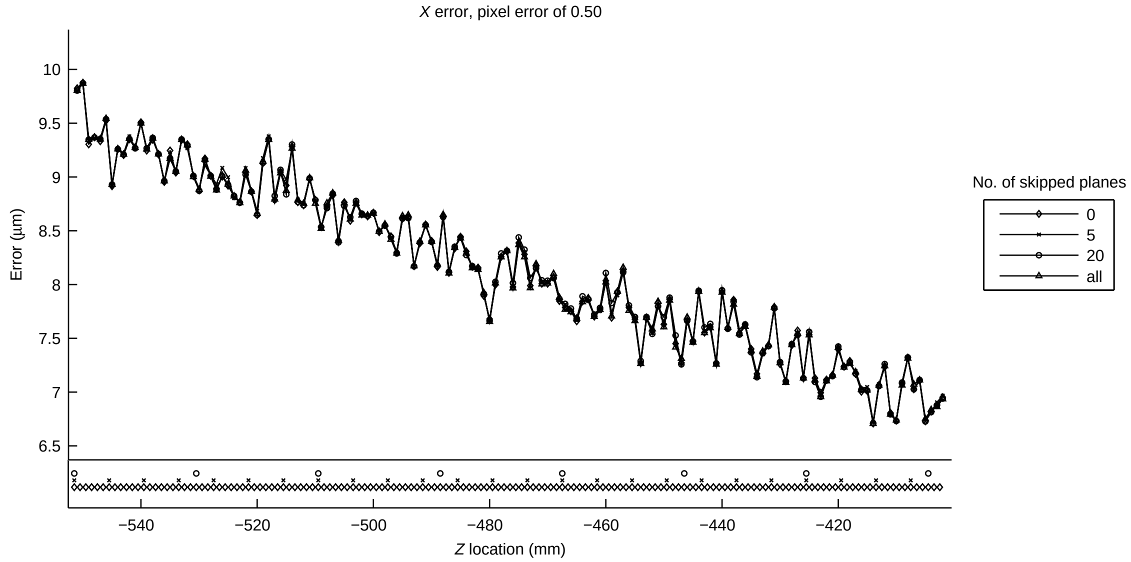
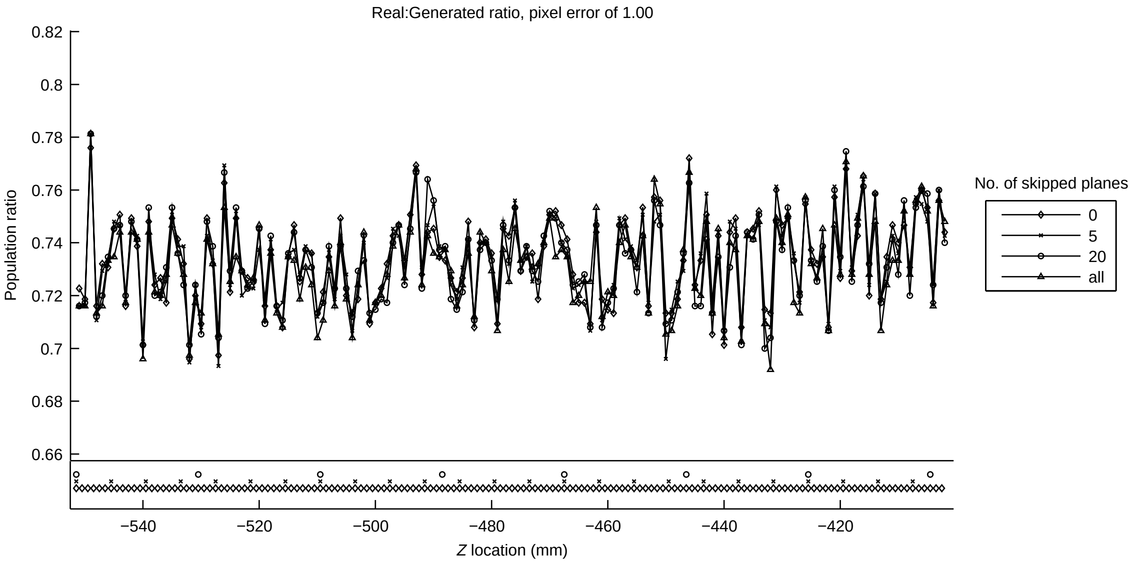


Figure 13.3-4: Error in X for dewarping done with pixel error of 0.50. The bottom plot shows the actual locations of the dewarping planes.



of the particles will remain within tolerance if the noise is 1.0 pixels. This is verified in figure 13.3-5.

Figure 13.3-5: Real to generated ratio for dewarping done with pixel error of 1.00. The bottom plot shows the actual locations of the dewarping planes.



13.4 Effect of Artificial Barrel

The artificial barrel distortion is generated with equation 4.2-34, using -1^{-9}, -10^{-9}, -15^{-9}, -20^{-9}, and -25^{-9} for Q_B. Barrel distortion brings out the worst in dewarping—specific planes where the dewarping error is much higher than in others, indicating that the optimization was not able to converge on the best values for the coefficients. As the amount of distortion increases, the frequency of failed fits, and the severity of their error, increase, as is evident in the difference between figures 13.4-1 and 13.4-2.

Figure 13.4-1: Average and standard deviation for the pixel error in the dewarping mapping for dewarping done with pincushion coefficient of -1^{-9}.

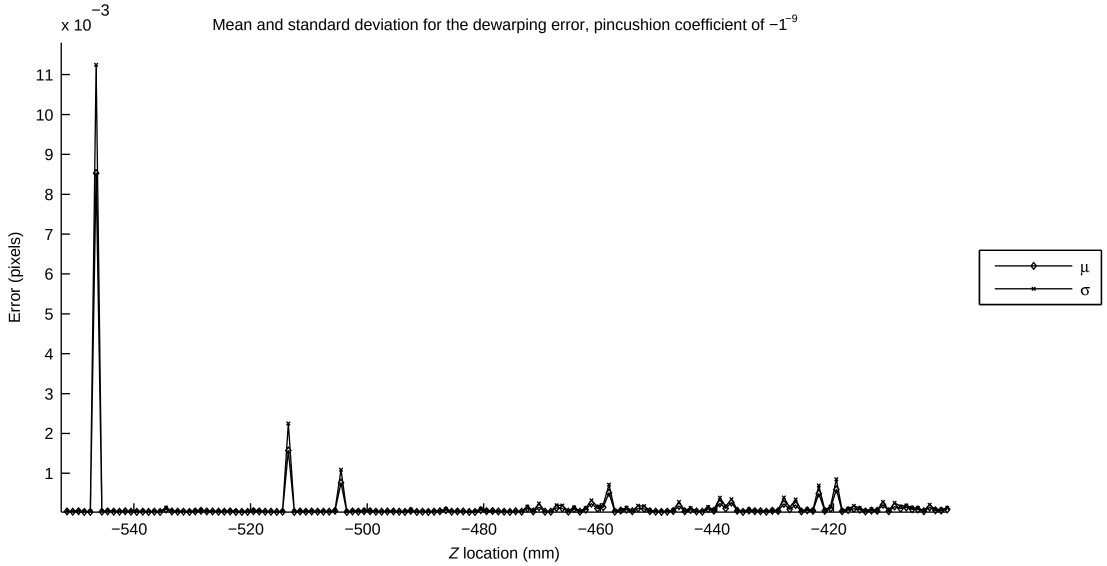
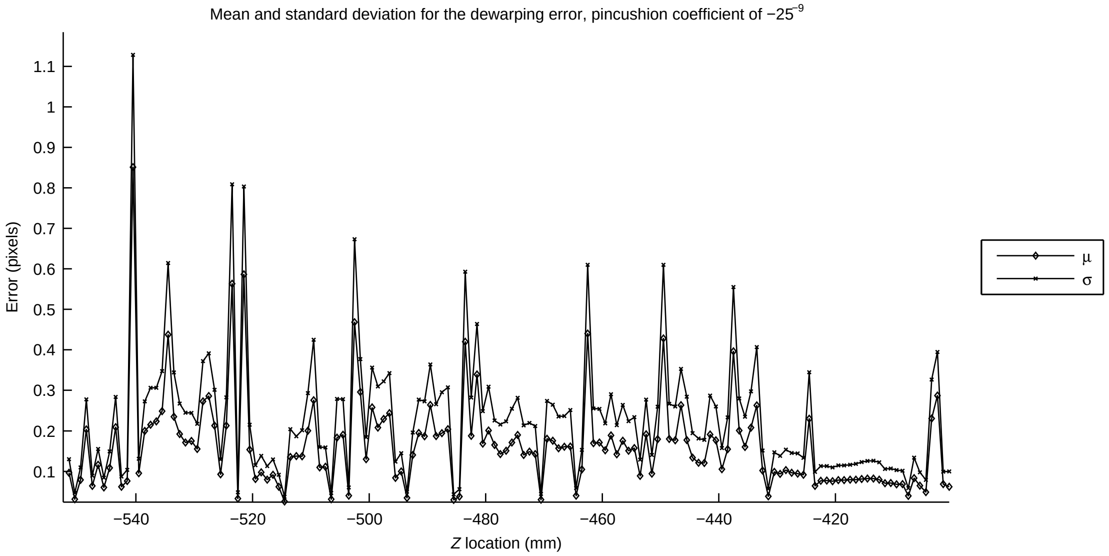


Figure 13.4-2: Average and standard deviation for the pixel error in the dewarping mapping for dewarping done with pincushion coefficient of -25^{-9}.



The dangers of using too many dewarping planes becomes evident in looking at the case of

Figure 13.4-3: Error in X for dewarping done with pincushion coefficient of -25^{-9}. The bottom plot shows the actual locations of the dewarping planes.

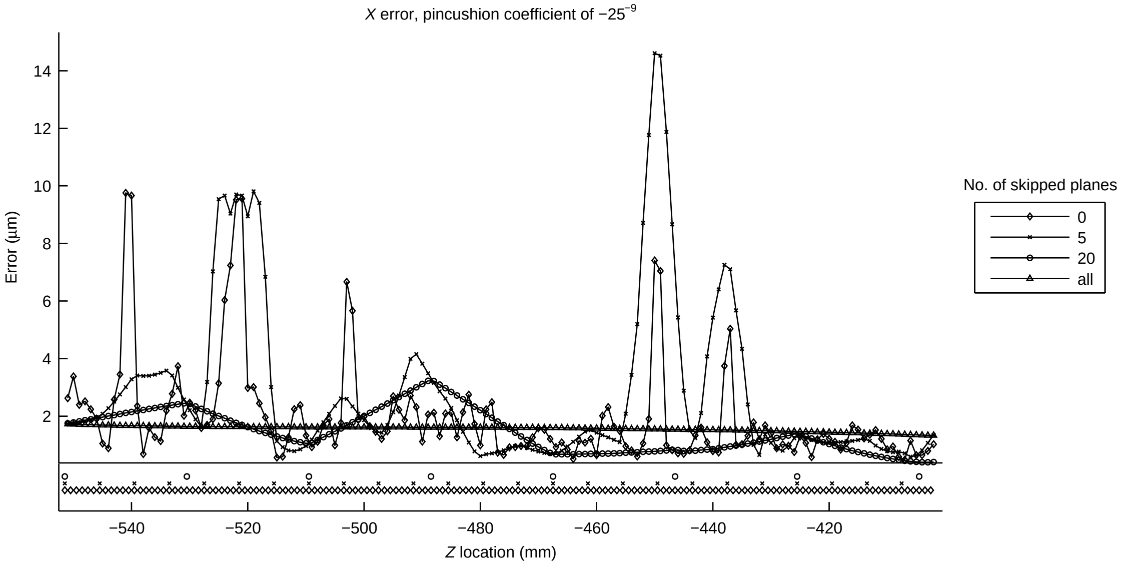


Figure 13.4-4: Real to generated ratio for dewarping done with pincushion coefficient of -25^{-9}. The bottom plot shows the actual locations of the dewarping planes.

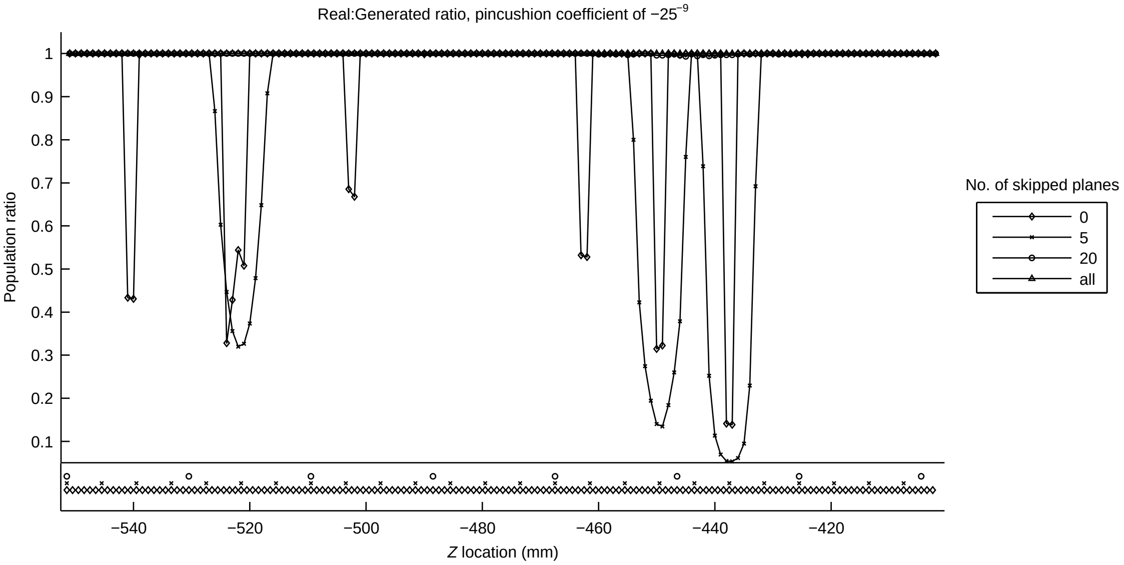
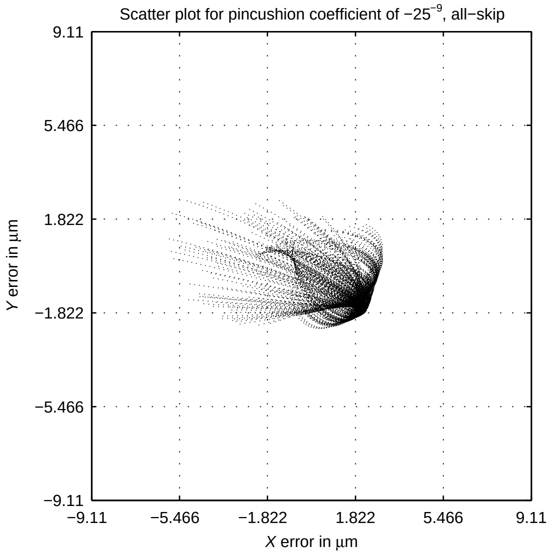


figure 13.4-2 in detail. Figure 13.4-3 shows how by indiscriminantly skipping even a predetermined number of planes dewarping errors can be smoothed substantially. As the planes get closer and closer to the camera, the error in the dewarping fit becomes more and more severe because the magnification is increasing—a pixel corresponds to a smaller spatial dimension. Comparing the error in X in figure 13.4-3 to the dewarping error of figure 13.4-2, the location of the most of the peaks of error in X coincide with the peak dewarping errors, however, the magnitude of the correlation is also related to the actual distribution and orientation of the error within the dewarping plane (which is lost in the statistics of figures like 13.4-2). As before, it is important to look not only at the error of the recovered particles, but also at the quality of the recovery itself (figure 13.4-4) which shows that some of these failed dewarping planes completely eliminate the possibility of recovering anything there.

Figure 13.4-5: Scatter plot of X versus Y error for the case of pincushion coefficient of -25^{-9}, all-skip.



In viewing the errors of the -25^{-9} case from the point of view of a scatter plot, it becomes evident that, even though the barrel distortion is simulated (and not due in fact to different ray paths), there is a residual error that remains when using single plane dewarping, as shown in figure 13.4-5.

In the case of these pinhole optics simulations, it seems logical to use single-plane dewarping in light of this, since it literally smooths the bumps that come from the occasional failed convergence. However in the real case, using single-plane dewarping will greatly diminish the Z accuracy as one moves toward the camera. The user is responsible for removing dewarping planes which do not converge properly or tweaking them until a better convergence is obtained rather than skipping a large number of them.

13.5 Effect of Tilt in Target

Tilting the target has completely different consequences. Tilts of 0.1, 0.5, 1.0, 2.0, and 5.0 degrees were evaluated. Because dewarping forces the images to appear as perfectly flat, perfectly aligned, and perfectly imaged, if the dewarping target is misaligned, this exact misalignment will be imparted to the data as well. In the case of tilt, a perfectly vertical sheet of particles will be recovered as having a tilt coinciding to that of the target.

There is a perhaps subtle, however important, difference between tilting the target and tilting the sensors. Following the discussion at the end of section 4.2.4, by tilting the target we are changing the position and thus angle of the rays that form the image, thus single-plane dewarping cannot possibly cope. Multi-plane dewarping, assuming that the dewarping functions converge properly (they do in this case) should guarantee a 100% recovery of the point cloud, however, as mentioned above, the error in the absolute position of the particles will depend directly on the deformation of the target.

Figure 13.5-1 shows exactly this. At a target tilt of 1°, single plane dewarping begins to fail at about 50 mm from the reference plane, indicating that, from the point of view of reconstructing the point cloud at 0.75 pixel accuracy, the reconstruction alone could cope with maybe just two or three dewarping planes covering a 150-mm-deep volume.

Figure 13.5-1: Real to generated ratio for dewarping done with target tilt of 1.0°. The bottom plot shows the actual locations of the dewarping planes.

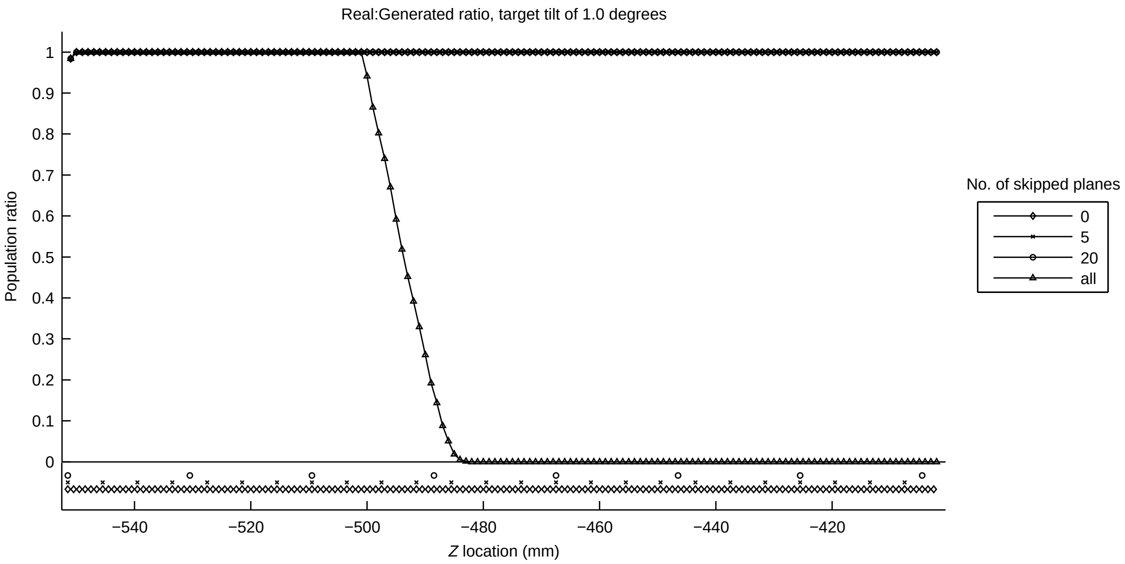
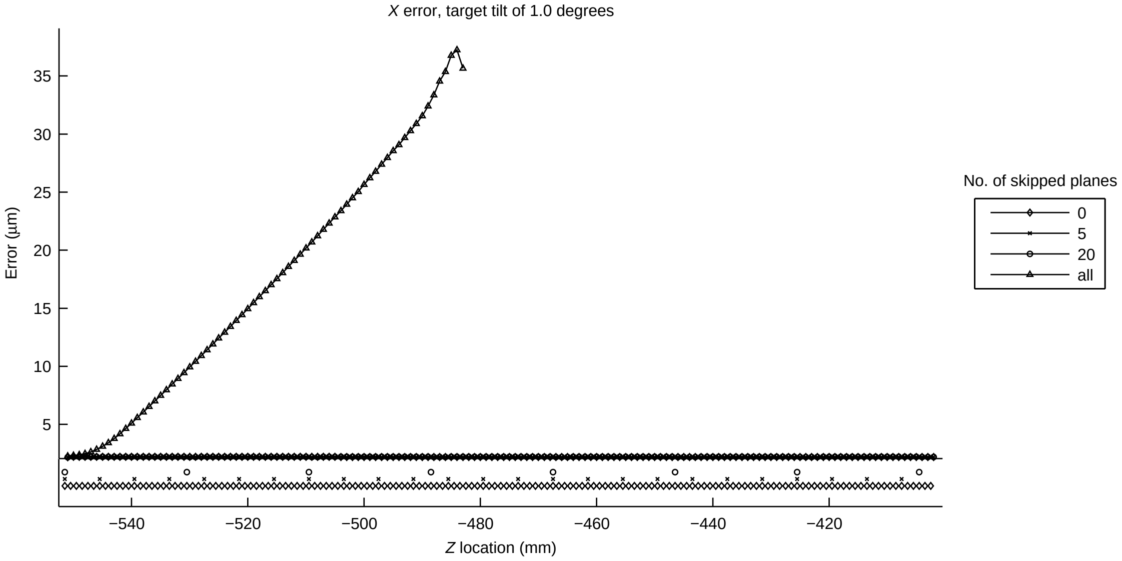


Figure 13.5-2 shows how multi-plane dewarping will all but completely eliminate *average* X error, but single-plane cannot. One should take care to properly interpret the plots as average errors per plane; the average X error is extremely low because it is nearly symmetric across the tilt axis and

thus nearly cancels out. Again, since tilting the target corresponds to changing the image-forming rays themselves it would be impossible to do so. One could imagine that, if 40 µm of error is tolerable, two or three dewarping planes would be sufficient; the interpolation between planes would keep the error under control.

A scatter plot of the error distribution across the volume proves that this error is just a rotation of the data's Z-plane. The Z error is larger simply due to the axis of rotation (Y). The case of a 5°-tilt shows this nicely (figure 13.5-3).

Figure 13.5-2: Error in X for dewarping done with target tilt of 1.0°. The bottom plot shows the actual locations of the dewarping planes.



This type of error makes this case a great chance to see the effects of interpolation between planes when the error is very smooth. The very slight tilt case of 0.1°, when viewed from the point of view of the ratio between the out-of-plane and in-plane error (figure 13.5-4), shows this exactly. Since a physical shift in Z corresponds to a larger pixel shift in x as one approaches the camera, the error in the spaces between dewarping planes increases. (In this figure the error ratio seems staggered because of numerical precision; the average X error is extremely small.)

Even at this slight tilt, though, the average Z error is a whopping 25 µm. Absolute position measurements are extremely sensitive to the target quality and alignment. But it is important to realize also that velocity is a relative position measurement, and thus the velocity vectors obtained with tracking, though misplaced in space, will have nearly correct components, because the displacement between frames of the particles forming them will be small and thus the relative error induced by a bad or ill-positioned target between them will also be very small.

At this time it is appropriate then to analyze the method by which the target is positioned relative the camera. Third-generation cameras are equipped with two laser diodes—one which in

Figure 13.5-3: Scatter plot of X versus Z error for the case of target tilt of 5.0°, 0-skip.

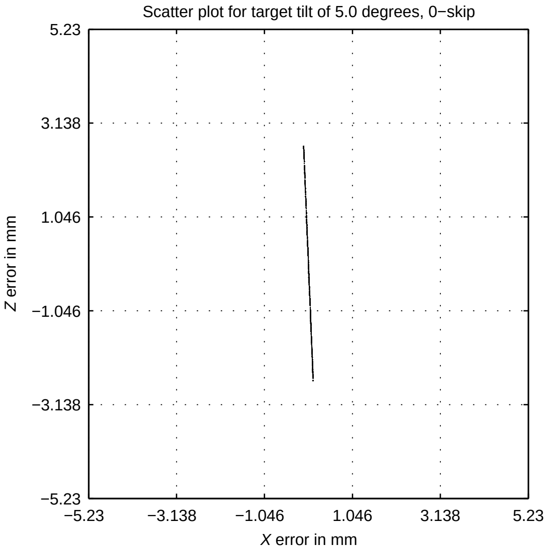
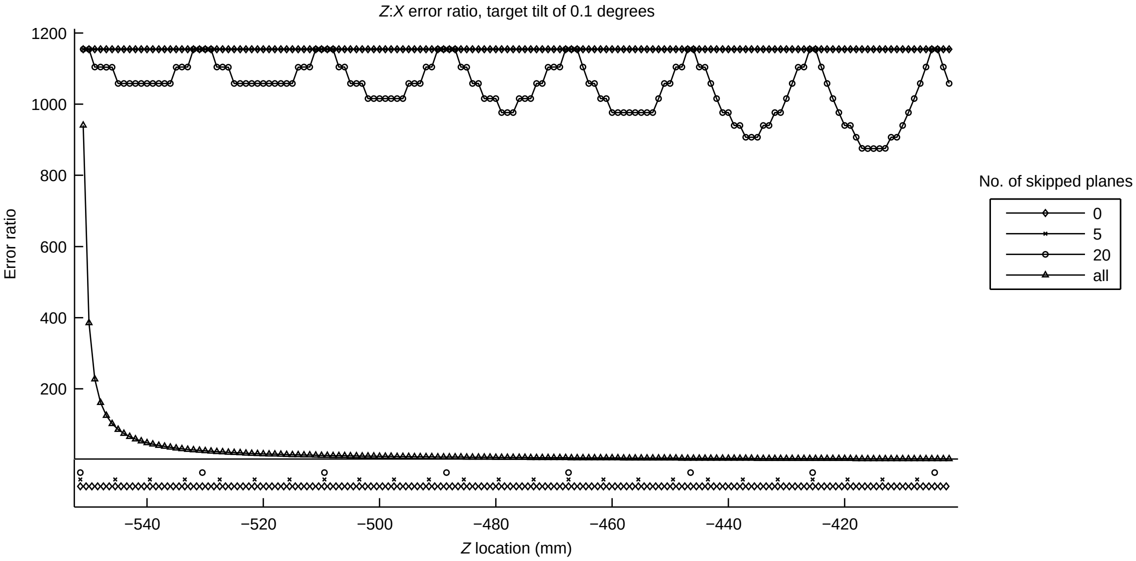


Figure 13.5-4: Z to X error ratio for dewarping done with target tilt of 0.1°. The bottom plot shows the actual locations of the dewarping planes.



theory corresponds to the optical axis, and another one projecting a beam at an angle to the axis equal to the angle of the sensor axis so that the reference plane location can be quickly found in multi-medium experiments. Both are glued into fixtures which are then bolted to the cameras. Alignment is done in a jig, with holes that offer a snug fit to the fixtures and a target mounted on a stage. The target is a metal plate with a small dimple machined into it so that its height (Y-coordinate) coincides to the Y-coordinate of the center of the fixture hole to within machining precision of the plate, the stage on which it sits, and the jig. The X-coordinates are aligned only to mounting precision on an optical table; this is done by applying pressure to the jig and the stage in the same direction during bolting. Thus the accuracy is dependent on the thread quality of the screws used.

The fixture must have a hole large enough for the diode to fit with enough space around it so that epoxy can be applied once it is in place, thus the diameter of the hole is nearly twice that of the diode. The diode is held in place with a 6-axis stage sandwich. It's position is adjusted so that three criteria are simultaneously met. First, the diode must be concentric to its hole in the fixture; this is checked by eye only. Second, the beam must land on the dimple in the target; this is evaluated by observing the brightness of the beam from an off-axis location—since the dimple is conical, the spot looks much brighter off-axis once it is within the dimple. The dimple has diameter of less than 400 µm, and the diode's beam is focused on the target surface, so at worst case the laser spot is some 200 µm larger than the dimple. The third, and most important criterion, is that when a precisely flat first-surface mirror is rested on the target face, the beam should reflect back "into" the laser diode.

The procedure obviously does not provide much X or Y precision, but the angular precision is high. Assuming the mirror is perfectly flat and of even thickness, and that the target plate is straight within mounting precision (which could not be worse than a few thousands of an inch concentricity with average-quality screws), and that it is possible to evaluate the "beam going back into the diode" well within two millimeters, and, keeping in mind that the reference plane distance (L) for the Emilio Camera is 640 mm, then the angle of the beam of the diode cannot be worse than 0.09°. Moving the diode fixture from the jig to the camera should introduce no angular misalignment since all surfaces are relatively small (1-inch-diameter) and are all machined flat.

When positioning the target, the same method of watching the reflection of the center diode beam is used to check the angle of the target relative to the camera—so the same level of accuracy can be expected. The pinhole optics tests show then that the best-case scenario of image quality coupled with the worst-case scenario for target alignment should yield an average Z error of around 20 to 30 µm for these third-generation cameras. Again, it should be reinforced that, to this day, no measurement performed with any defocusing camera was intended to provide absolute position, and the error introduced by target misalignment do not affect relative measurements perceptively.

13.6 Effect of Misaligned Target Motion

In this simulation, the movement of the target along Z was misaligned at 0.01 $\mathrm{mm}_X/\mathrm{mm}_Z$, 0.05 $\mathrm{mm}_X/\mathrm{mm}_Z$, 0.10 $\mathrm{mm}_X/\mathrm{mm}_Z$, and 0.30 $\mathrm{mm}_X/\mathrm{mm}_Z$, but the images are otherwise perfect, so there is no error in the dewarping.

Here we see a similar situation to the one of a tilted target, because the result is that the data is "sheared" just as the movement of the target is, so the end result is an artificially induced X velocity. In the displacements that can be expected in real experiments, this induced velocity would be nearly unmeasurable though, as the case above, the resulting vector's position would suffer from error.

In a real situation, even an error of 0.01 $\mathrm{mm}_X/\mathrm{mm}_Z$ would be hard to miss, since it is very easy to check target traverse alignment by simply aligning the center diode to a particular point on the grid and, after traversing the grid for a substantial length, seeing if the laser dot has moved relative to the grid. The estimate of the straightness of the center diode beam above indicates the error can be at most 0.003 $\mathrm{mm}_X/\mathrm{mm}_Z$, which translates to an estimated 60 µm error in X relative to the depth of the volume.

13.7 Effect of Incorrect Target Z

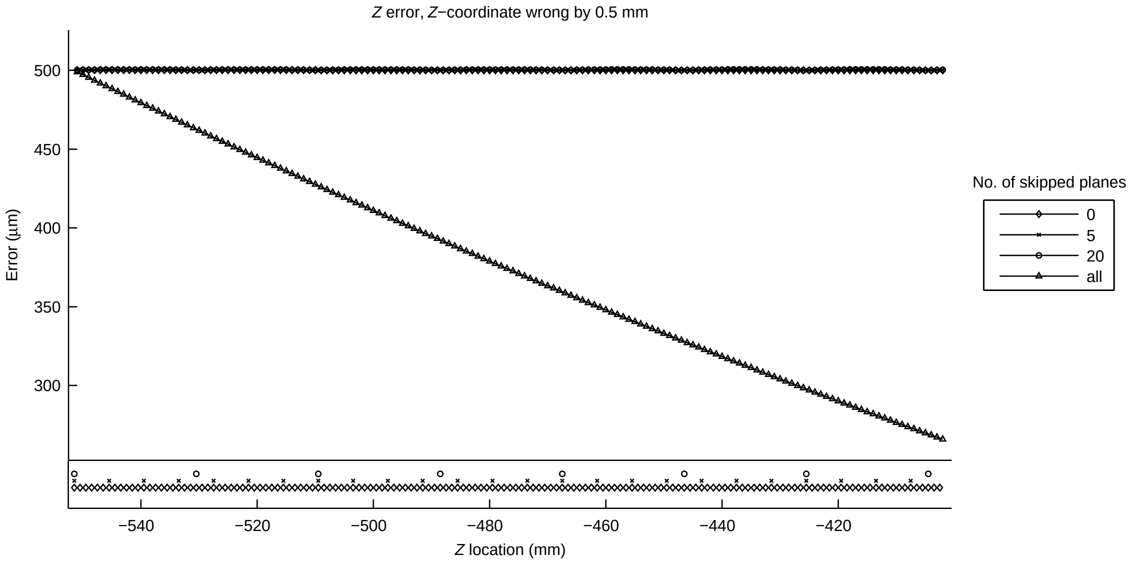
As a third example of an improperly performed dewarping set, the effect of assigning an incorrect Z coordinate to the target is analyzed.

Currently the dewarping procedure requires that the user provide a distance between the camera and the target at each dewarping plane. Commonly, with the target moved by a motorized traverse, this is done by measuring the distance to the reference plane target from the face plate of the camera with a tape measure, then doing the same for the last plane in the sequence, and assuming that the stepper motor is not slipping and thus provides exactly the same interval between planes.

As examples of situations where the Z coordinate assigned to the target is wrong, four situations were tested. In the first, the assigned Z is 0.5 mm off the actual Z. The second situation has a position error varying linearly from +0.5 mm at the reference plane to −0.5 mm at the last plane (200 mm ahead of the reference plane)—this is like a user measuring the reference plane and last plane distances incorrectly and then interpolating as described above. The remaining two cases are of random error at each plane, first within 0.5 mm and then within 2.0 mm, simulating a setup where the plane locations had to be measured at each location.

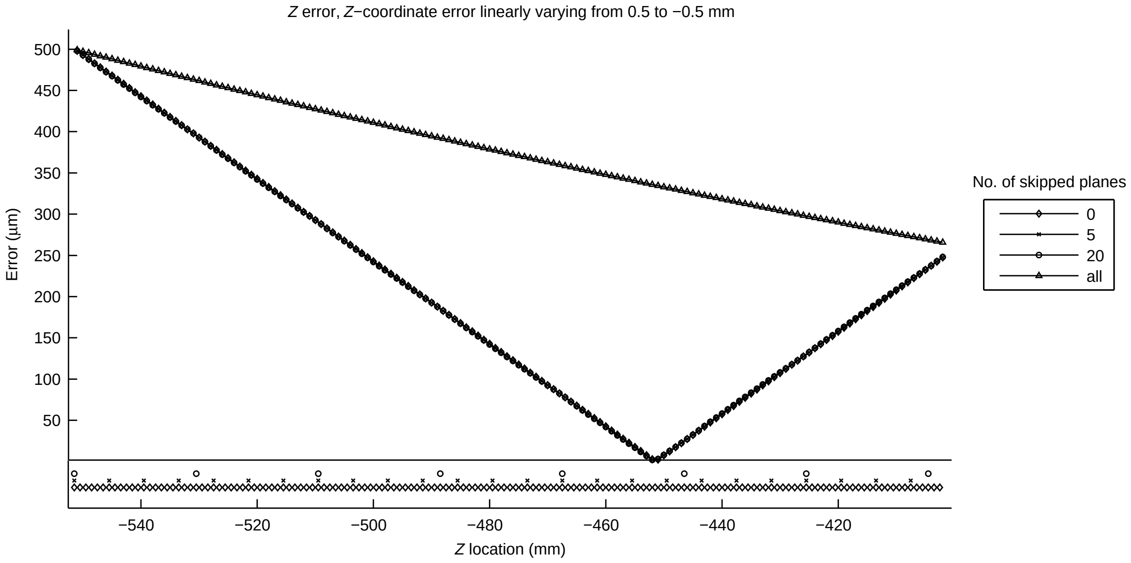
In the first case, that of a constant error, the Z error (figure 13.7-1) when using multi-plane dewarping equals the error in the assigned Z coordinates, much like in the other two cases of misaligned target motion—and is really just a shift in Z nearly equivalent to shifting the origin in the final data set.

Figure 13.7-1: Error in Z for dewarping done with Z-coordinate wrong by 0.5 mm. The bottom plot shows the actual locations of the dewarping planes.



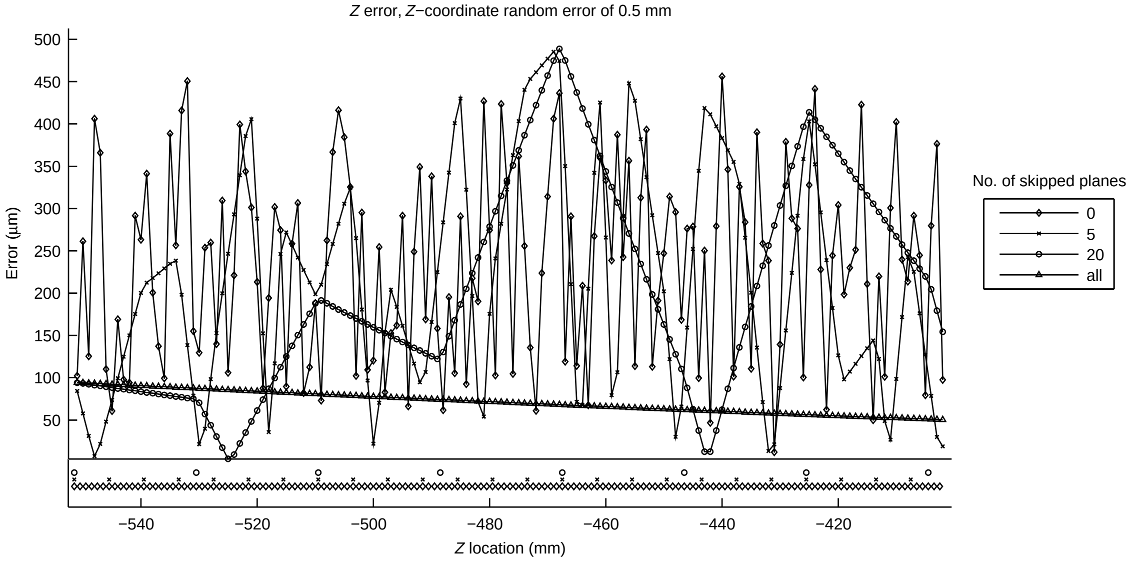
In the case of a linearly varying error, there is a point where the assigned Z coordinate is correct and thus the Z error is 0, as shown in figure 13.7-2 at around -450 mm.

Figure 13.7-2: Error in Z for dewarping done with Z-coordinate error linearly varying from 0.5 to -0.5 mm. The bottom plot shows the actual locations of the dewarping planes.



In the case of random error for each plane, the error is directly transfered to the data, which is a much more dangerous situation than measuring one plane incorrectly and calculating the position thereafter based on a traverse's precise movement. Whereas calculating based on a wrong measurement will induce an absolute error which is irrelevant from the point of view of velocity, error at each dewarping plane will induce a relative error as well and thus *will* affect the velocity.

Figure 13.7-3: Error in Z for dewarping done with Z-coordinate random error of 0.5 mm. The bottom plot shows the actual locations of the dewarping planes.



13.8 Conclusion of Simulations

Multi-plane dewarping is extremely robust in aligning images affected by commonly encountered problems. Most simulated problems can be taken care of successfully with just a few planes. Still, none of these simulations account for the differing ray paths in a real glass lens, so the analysis is more a test of the convergence of the dewarping function and gives an indication of the result of certain problems.

The tests show that it is very difficult to create problems in relative measurements such as velocity. The largest possibility for error lies in user operation of the dewarping setup, such as misaligning the target's traverse or the target itself. These prove to cause very little relative error if any, although for absolute measurements, the effects can be substantial.

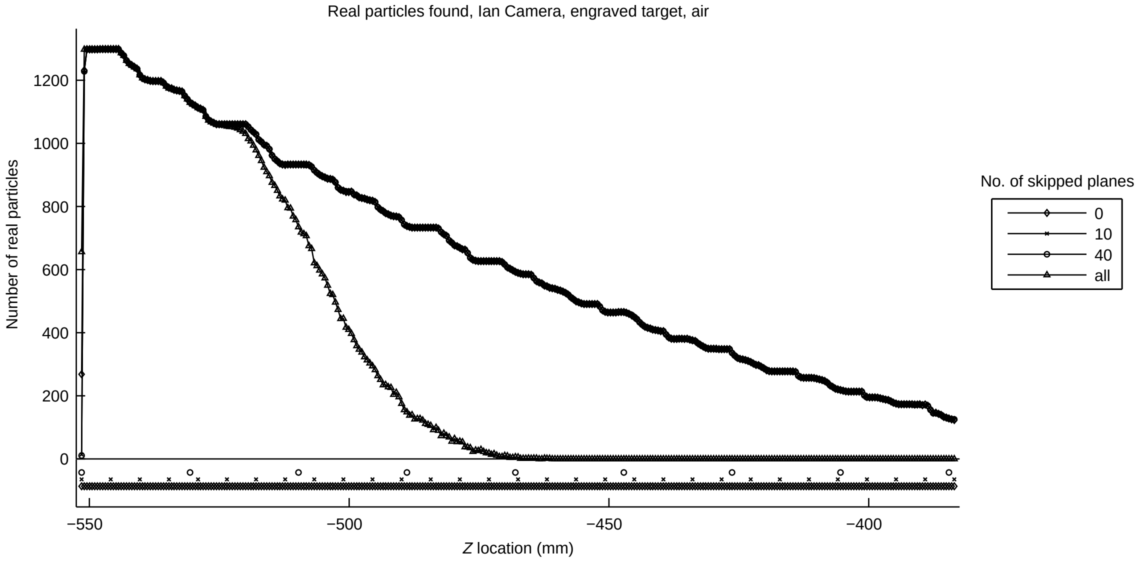
Analysis—especially the case of incorrectly assigned Z coordinates—points to the importance of positioning the target *consistently* if not correctly. A traverse with high relative precision, such as one with an encoder or one driven by a strong stepper motor, is a must for positioning the target in Z. Alternatively, if a camera shows to need few dewarping planes (such as the Emilio Camera), bars machined to specific lengths can be used as spacers to position the target at precise distances from the camera.

13.9 Evidence of Performance in Real Data

With data from real cameras, the effect of a real lens can be seen. The data presented here is from actual dewarping sets used either for testing camera performance or as calibrations for actual experiments. The dewarping images were processed with the dewarping coefficients (generated from the same images) and the resulting grid point coordinates were compared to the coordinates of a perfect grid at the same Z coordinate. Dewarping images are acquired as a pixel-by-pixel average of at least 30 images to reduce image noise; noise is further reduced by pre-processing the images with a 1-pixel-radius Gaussian blur. For Ian Camera data, the images were processed with a 0.5 pixel matching tolerance, since successive tests show that this usually covers the entire volume when processing dewarping target images. For the Emilio Camera, the tolerance was 0.25 pixels.

As mentioned in section 6.3, the difference between an optical system with a real lens and one based on pinhole optics can be interpreted as the lens causing a movement of the pinhole as a function of X, Y, and Z. Real distortion causes a variation in the equivalent pinhole location as a function of radial lens coordinate (X, Y, and Z in space). This is the case of the Ian Camera, and, as far as DDPIV is concerned, it causes a change in the shape of the pattern that matching particle images generate as a function of radial lens coordinate. It is expected then that, for the Ian Camera, single-plane dewarping is unable to reconstruct the entire volume as is evident from figure 13.9-1.

Figure 13.9-1: Real particles found for dewarping done with the Ian Camera, engraved target, in air. The bottom plot shows the actual locations of the dewarping planes.



The Emilio Camera, on the other hand, has very little distortion in the image, and thus there is no appreciable movement of the equivalent aperture with the radial lens coordinate, which means that in space, there is no movement with X and Y for a given Z. Because of this, single-plane dewarping can reconstruct the entire volume, as shown by figure 13.9-2, but, because there is still movement of the equivalent pinhole with respect to Z, the Z coordinate obtained with single-plane dewarping has more and more error as one approaches the camera (figure 13.9-3).

Note that the number of grid points reconstructed ("real particles") will decrease as one approaches the camera since the grid spacing remains constant but the mappable region shrinks in X, Y (it is pyramid-shaped), as can be seen in figure 7.4-26.

Figure 13.9-2: Real particles found for dewarping done with the Emilio Camera, etched target, in air. The bottom plot shows the actual locations of the dewarping planes.

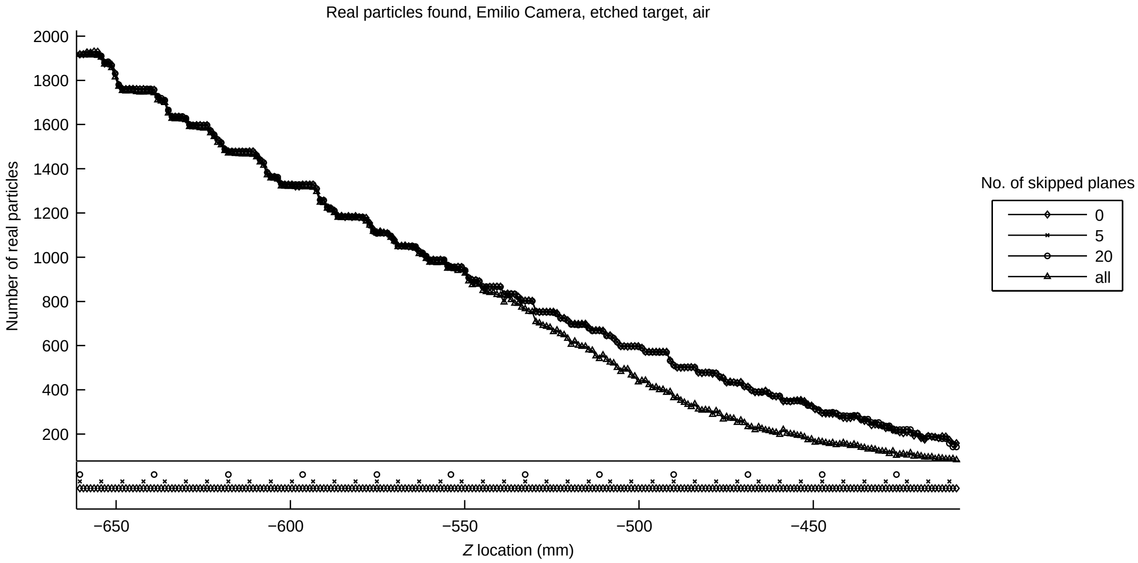
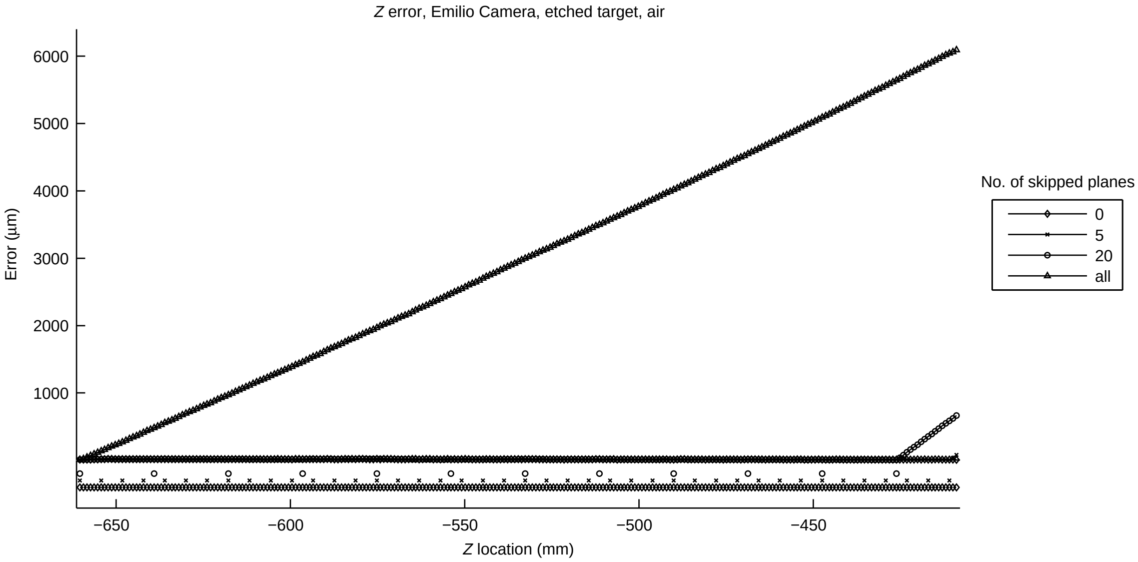
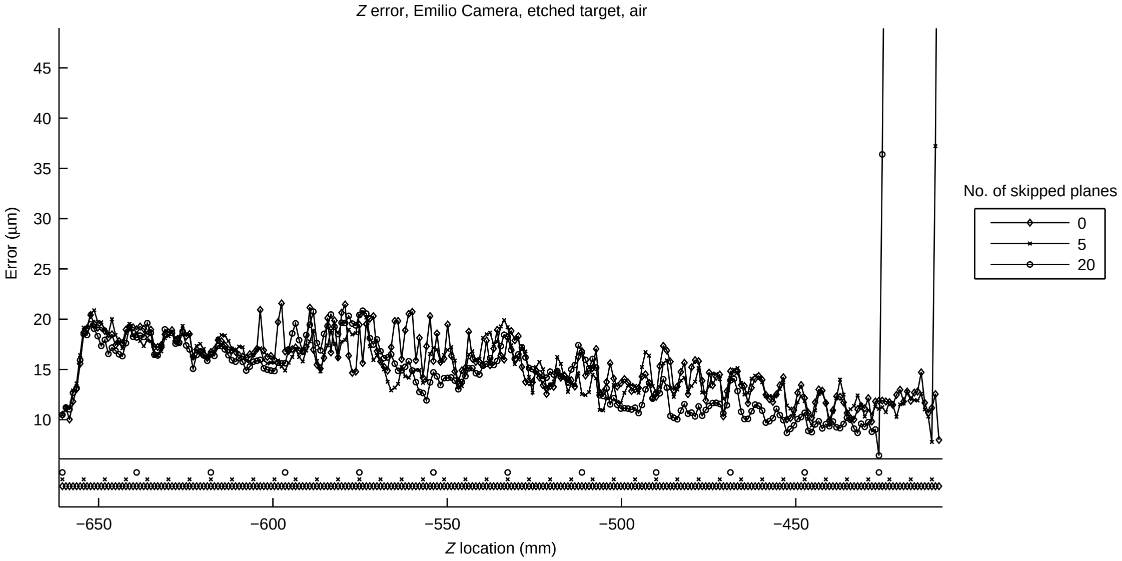


Figure 13.9-3: Error in Z for dewarping done with the Emilio Camera, etched target, in air. The bottom plot shows the actual locations of the dewarping planes.



Looking more closely at the case of multi-plane dewarping, it is evident that even with less than one dewarping plane every 20 mm the error in Z is well in check (figure 13.9-4).

Figure 13.9-4: Error in Z for dewarping done with the Emilio Camera, etched target, in air. The bottom plot shows the actual locations of the dewarping planes.



The Ian Camera was always used with a dewarping target made by laser-engraving the grid of dots onto the mirror surface of a quarter-inch-thick second-surface mirror (so that the mirror backing served as a mask when backlit). The mirror substrate was thick, which provided sufficient flatness, and the target was always imaged with the substrate facing back so that there was no scaling problems due to refraction. Because the laser used for engraving was a moving-head CO_2 laser, the accuracy of the engraving is not great, and even under visual inspection at moderate magnification, it is clear that the dots are elongated in the direction of the movement of the laser head.

For the Emilio Camera, a photo-chemical etching process was used to create a much more accurate version of the engraved grid. The principal differences are that the substrate is checked for flatness, the etching process itself has a resolution of 10 microns, and that the etched target is 30 times more expensive. It is natural to want to compare the two.

The dewarping error is significantly higher in the case of the less expensive target, especially closer to the camera, where it quickly gets out of control as the imperfections in the target are magnified. As can be seen in comparing figures 13.9-5 and 13.9-6, the error is at least three times higher with the engraved target, but in a volume up to about 140 mm deep, it is within a factor of 4 of the error obtained using the etched target.

Comparing the Z error of figure 13.9-7 to that of figure 13.9-4, it is evident that even though there is an increased precision with the etched target, it is doubtful if in practice the enormous cost increase is justifiable, since the Z accuracy obtained with the cheaper target is still within 0.02%

Figure 13.9-5: Average and standard deviation for the pixel error in the dewarping mapping for dewarping done with the Emilio Camera, etched target, in air.

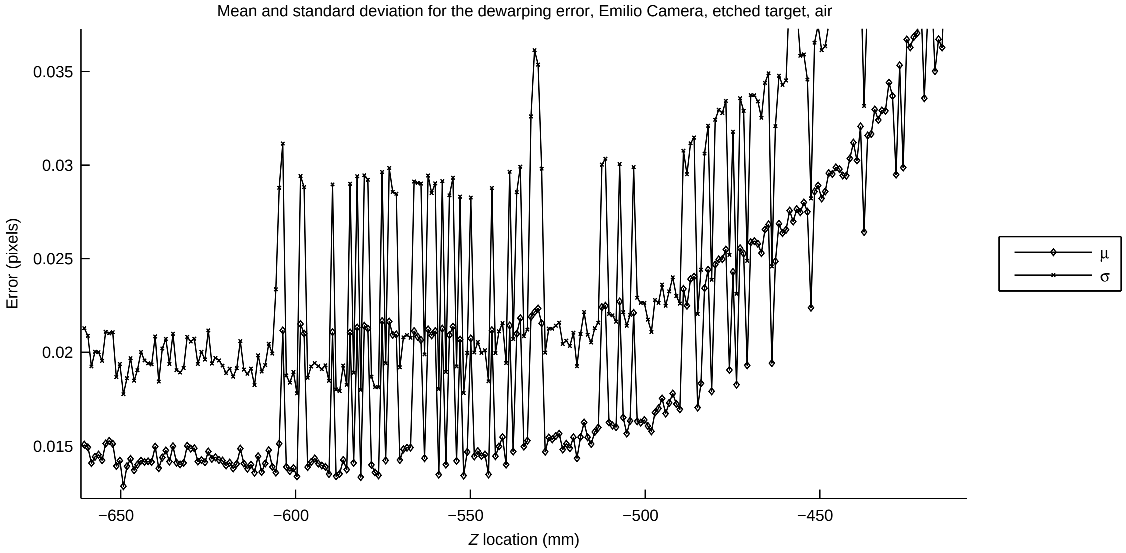
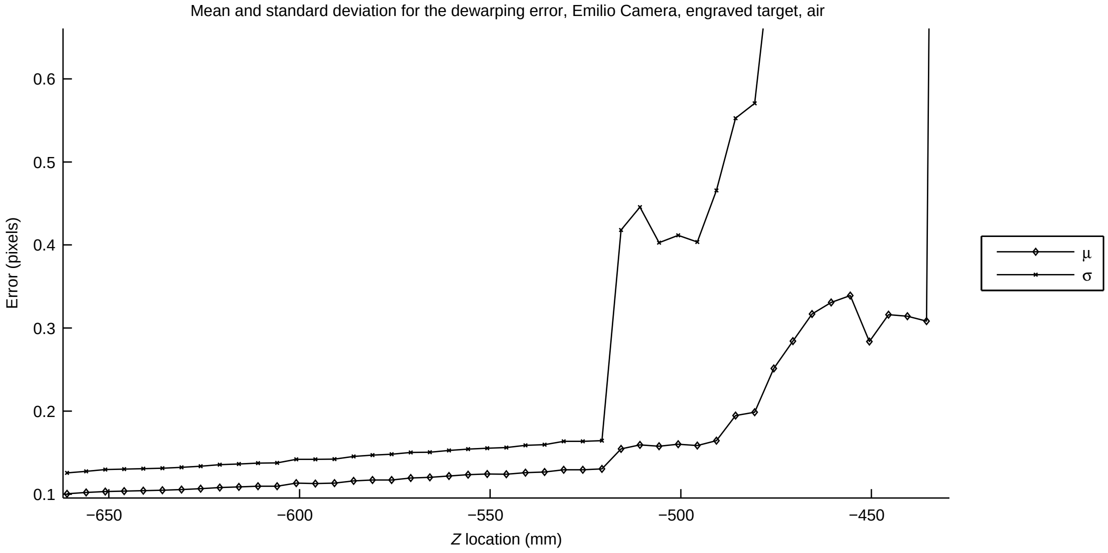
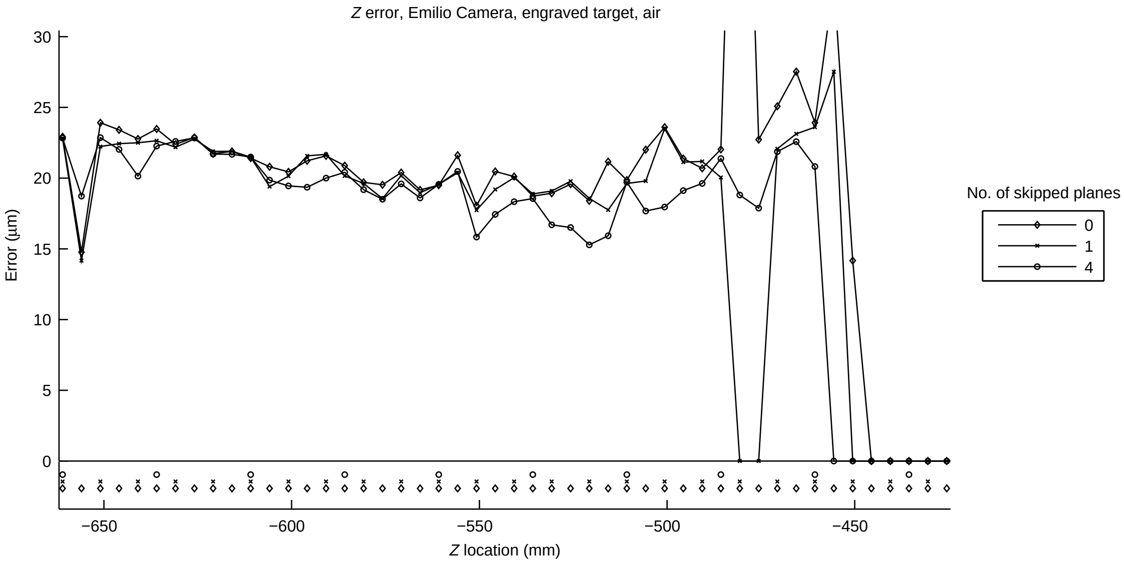


Figure 13.9-6: Average and standard deviation for the pixel error in the dewarping mapping for dewarping done with the Emilio Camera, engraved target, in air.



of the volume depth. Of course the Z error does not actually depend on the volume depth, so for thin, light-sheet-style measurements, or measurements of small surface deformations, it may still be justifiable to use a more accurate target.

Figure 13.9-7: Error in Z for dewarping done with the Emilio Camera, engraved target, in air. The bottom plot shows the actual locations of the dewarping planes.



Chapter 14

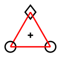
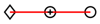
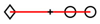
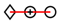
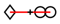
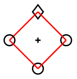
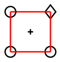
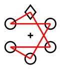
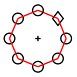
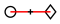
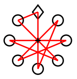
Reconstruction of the Point Cloud

14.1 Introduction

The only information available for matching the particle images between apertures is their position in the image. The sub-pixel coordinates are corrected by multi-plane dewarping to account for the optics of the camera and the experimental setting. An initial Z estimate is calculated using equation 4.3-40 once a pair of particle images from apertures 1 and 2, whose coordinates are corrected with the dewarping coefficients of the reference plane, are matched using a coarse tolerance. This Z estimate is then used to correct the particle image coordinates with an interpolation of the correction between the nearest two dewarping planes and particle images in subsequent apertures are checked for a match using a fine tolerance. Once all the particle images belonging to a particle are collected, the final X, Y, and Z coordinates of the particle are obtained using equations 4.3-41 and 4.3-40. This matching procedure is described in detail in section 10.3.1.2. A perfomance analysis of multi-plane dewarping is in chapter 13.

14.2 Simulation Details

Table 14.2-1: Summary of the aperture layouts used in the tests of point cloud reconstruction. Figures are to scale; circles represent apertures and crosses represent the optical axis. The red lines show the aperture pairs which contribute to the value of Z_P, that is, these are the lengths s_{ij}. The aperture shaped like a diamond is the first on the order.

Camera	Apertures	Aperture layout	$\bar{B}_{ij}$
"A"	3		11.77
"B"	3		15.69
"C"	3		15.69
"D"	3		7.845
"E"	3		7.845
"F"	4		11.77
"G"	4		11.77
"H"	6		10.11
"I"	8		6.362
"J"	2		11.77
"K"	8		14.88

There are four main factors that affect the performance of the point cloud reconstruction. One is the quality of the Gaussian fit, which in real cases is also related to the seeding particles themselves. The seeding density also affects the final reconstruction—not just from the point of view of particle image overlap but also because as the seeding density increases, so does the chance of matching ghosts. The quality of the dewarping coefficients also matters since a bad convergence at a given plane can obliterate the reconstruction in a neighborhood of this plane. The aperture layout of the camera itself, and the number of apertures, affects both the precision of the reconstruction and the quality (number of ghosts).

We have already gained some knowledge about the Gaussian fit in chapter 12. To eliminate the question of dewarping while analyzing the effect of the aperture layout, we analyze 11 different cameras in a simulation. Within this simulation we look at different image types in seven different conditions and 6 different seeding densities. First are 2-pixel-radius particle images, which proved to be the optimum particle image size. Images were generated with both perfect images and also magnitude 5 pseudo-random noise. This noise is enough to rid the Gaussian fitting algorithm of any bias due to discretization, but increases the magnitude of the error of the fit (see section 12.4). To remove dependency on the Gaussian fitting algorithm, different aperture layouts were also tested using a list of x, y particle image coordinates as input. First these were processed with 0.01 pixel matching tolerance, which gives an indication of how prone to ghosts a particular point cloud is. They were also processed with 0.75 pixel matching tolerance, since in real experiments this was the tolerance typically used with the Ian Camera (on which all these simulated cameras are based). The same text (`PTS`) input was also processed at 0.75 pixel tolerance with pseudo-random noise of magnitude 0.05, 0.10, and 0.20 pixels artificially added. This was done to compare random noise to the type of error in the Gaussian fitting algorithm.

Table 14.2-1 shows the aperture layouts that were considered. Camera “A” is the layout of the Ian Camera (see table 7.4-9). Camera “B” is formed by maintaining the separation between adjacent apertures to be equal to that of the Ian Camera, except the apertures are all in a straight line (and one is on the optical axis). Camera “C” is the same, except the central aperture is moved to the midpoint between the optical axis and one of the outer apertures. Cameras “D” and “E” follow the same concept, except the total width is constrained to that of the base length of the Ian Camera aperture triangle. Cameras “A” through “E” all have three apertures.

Cameras “F” and “G” each have four apertures. To date, all cameras built have had three sensors, but some (like Elsinga et al. [2005]) argue that adding apertures only increases the accuracy of 3D systems. The aperture spacing is such that adjacent apertures are separated by the same length as in the Ian Camera. The difference is in the orientation; for camera “F”, the shifts in particle image coordinate as a function of Z are aligned with the pixel array, whereas for “G” they are at 45°.

Camera “H” is a double Ian Camera, that is, two Ian Cameras mounted so that their optical axes coincide. Camera “I” has 8 apertures (the maximum number currently supported by the software) and will serve as an upper limit of performance versus number of apertures. Finally, camera “J”, the same as Camera “D” but without the central aperture, will prove that even though it is sufficient to have two apertures to obtain 3D information, in practice it generates too much noise.

Camera "K" is identical to Camera "I", with the exception that the aperture order is such that it maximizes $\bar{B}_{ij}$. This is in light of the fact that the error depends on the sensitivity (see equation 5.2-16). Note that unlike cameras "F" and "G", which have different orientations and thus the point clouds were re-traced for each camera, cameras "I" and "K" use *identical* images, thus in the case of PTS files with pseudo-random noise added, the particle images for camera "I" and "K" each have identical error applied to their position whereas the ones for "F" and "G" do not.

The point clouds were generated to be only 1 µm thick in Z so that ghosts were easy to identify. The different densities can be seen in the appendix, in figures B.1-1 through B.1-6 in Graff [2007b]. As in the dewarping tests, the clouds were traversed 1 mm in Z at a time to form each image.

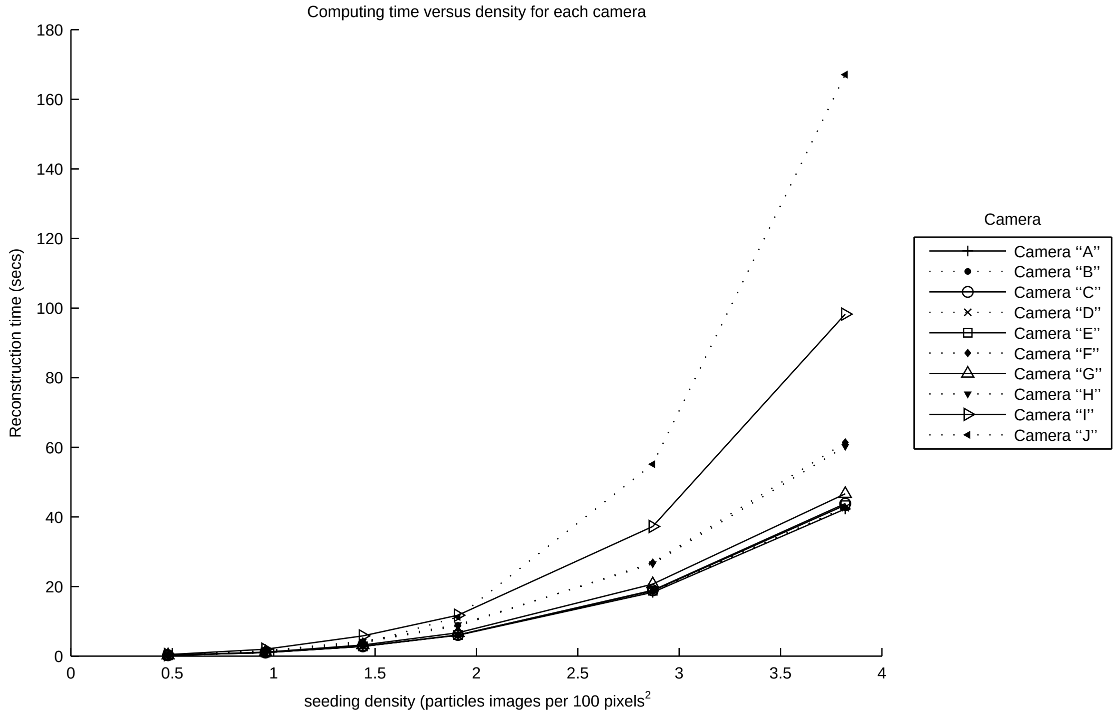
To analyze the results, a MATLAB script traversed through all the particles that were found and linked them to the closest real particle, deeming the distance between them the error in the reconstruction. Ghost particles were automatically detected during processing with DDPIV for the text input; bitmap input required ghost detection based solely on final location (relative to the generated point cloud). The cutoff for detecting ghosts was chosen to be 70 µm; that is, if a particle found is not within 70 µm of a generated particle, it is a ghost. This value was chosen as it is the worst-case estimate of the Z error in the Ian Camera (determined from experiment). Double particles are those which are within this distance of *more than one* generated particle; generally speaking these should occur only when two particle images are so close together that they are well within the processing tolerance such that the resulting particles are so close together that it is hard to distinguish whether their final position indicates a ghost or high error in a real particle.

14.3 Computing Time

The point cloud reconstruction step is usually the one that consumes the most computing time. It is a function of the seeding density and the number of apertures, the latter in two different ways.

The seeding density obviously affects computing time since as it increases, there are more particle images to check for matches. The number of apertures affects the computing time because as it increases the search is complicated slightly as a full match requires finding particle images in more lists (one for each aperture). The proportion of ghost particles recovered can heavily affect computing time because for each match made, all the quantities (position, brightness, etc.) must be computed for that particle. As will be made clear below, the two-aperture case of Camera "J" has an exhorbitant proportion of ghosts, and thus as the seeding density increases it is the one that consumes the most computing time to reconstruct a cloud. The three-aperture cameras always take the minimum amount of time because of a low number of apertures combined with a low proportion of ghosts. As the aperture numbers increase, so does slightly the proportion of ghosts because (as

Figure 14.3-1: Time to reconstruct reference plane point cloud as a function of seeding density for each camera.



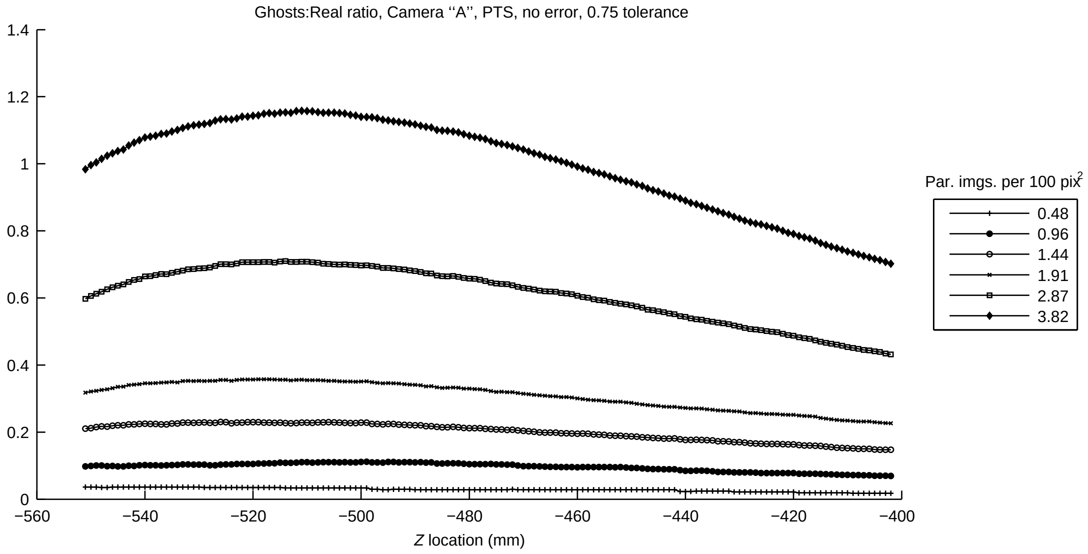
will be seen below) as the number of apertures increase so do the ghosts resulting from two particle images being close together in one aperture image. This coupled with the fact that now there are also more lists of particle images to access increases computing time, hence Camera "I" is the second-most resource-intensive arrangement.

14.4 Effect of Seeding Density

The seeding density can affect the performance of the point cloud reconstruction in two principal ways. First, the higher the seeding density, the more likely that particle images will overlap, which can lower the precision of the Gaussian fitting algorithm and thus affect both the reconstruction quality and precision. Second, the higher the seeding density, the higher the proportion of ghosts that can result. This loss of quality completely overpowers the loss in precision from particle image overlap.

As an example, figure 14.4-1 shows the effect on the ratio of ghost to real particles as a function of seeding density for the simulated Ian Camera (camera "A") for `PTS` data at 0.75-pixel tolerance. The number of ghost particles varies with Z since it is a function of the probability that the arrangement of the point clouds will line up to generate mismatches within a 0.75-pixel radius.

Figure 14.4-1: Ghosts to real ratio for camera "A" and PTS, no noise, 0.75 tol.



To give more meaning to the seeding densities, a density of 1.44 particle images per 100 pixels squared, at which figure 14.4-1 shows there are at least 20% ghosts in the recovered point cloud, corresponds to a 1600×1200 pixel image containing 27,648 particle images, which is considered "dense" in practice[1]. Most experiments are performed around 0.96 particle images per 100 pixels squared (18,432 particle images in a 1600×1200 pixel image) or below.

The effect of overlapping particle images can be seen as an increase in the position error as the seeding density increases for bitmap inputs. Figure 14.4-2 shows how for the perfect Gaussian case, the error in X increases more or less linearly with the seeding density. Conversely the same plot for a PTS input with pseudo-random pixel error of 0.2 pixels magnitude (figure 14.4-3) shows the expected invariance in the X error with density. It should also be noted that the X error at the maximum seeding density for perfect bitmap input is higher than that for 0.2-pixel artificial error, even though in the Gaussian fitting tests it was shown that this is more than twice the value of the 99th percentile for perfect 2.0-pixel-radius Gaussian particle images.

The effect of seeding density on quality is also strong, as can be seen in figure 14.4-4. The higher the density, the lower the proportion of correctly recovered particles—but the *higher* the *absolute number* of particles recovered. The increased number of particles recovered with density will most likely not justify the increase processing time, and, once the increase in error and especially the number of ghosts is factored in, the higher densities evidently do not yield good data in volumetric domains where the ghosts are intermixed with the real data and it is thus impossible to separate

[1] Although here the densities are enumerated in pixel space, the seeding density for simulations is in fact fixed in real space, not pixel space. Thus the values in particle images per 100 pixels squared are only correct for the reference plane, decreasing as the particles approach the camera with the optical magnification.

Figure 14.4-2: Error in X for camera "A" and 2.0p radius, no noise

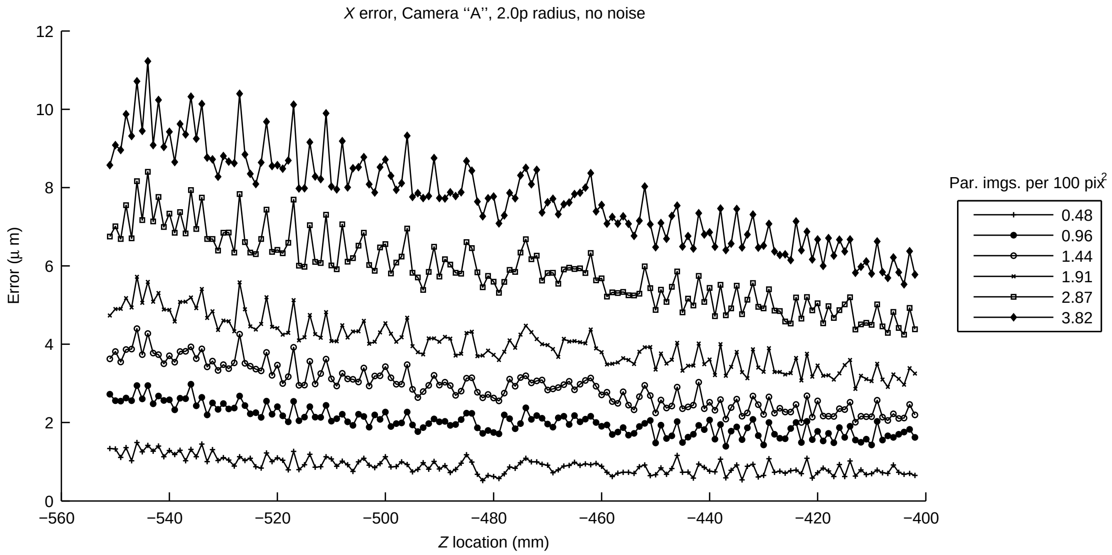
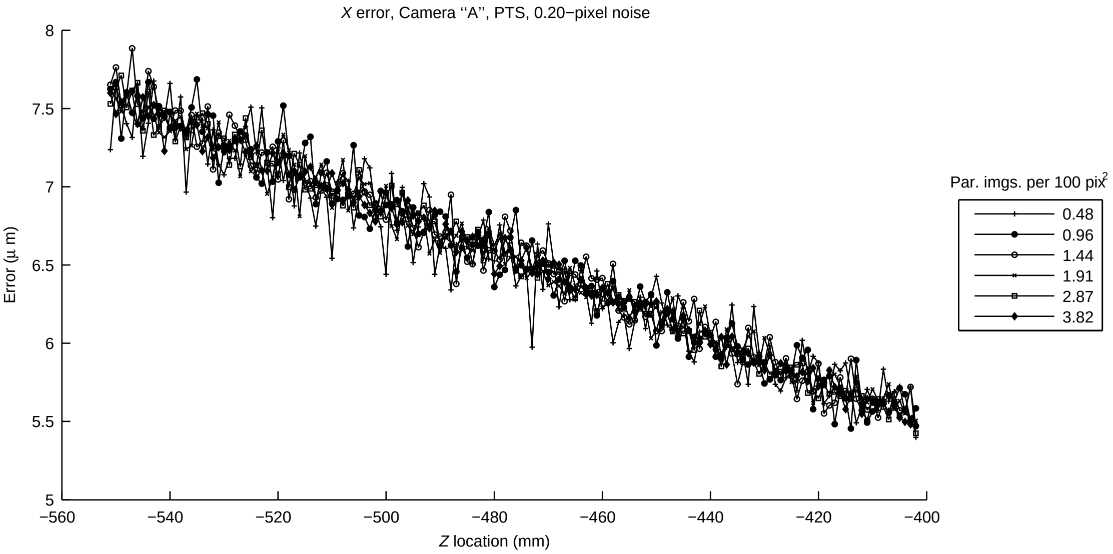


Figure 14.4-3: Error in X for camera "A" and PTS, 0.20p noise



any of them.

Figure 14.4-4: Real to generated ratio for camera “A” and 2.0p radius, no noise

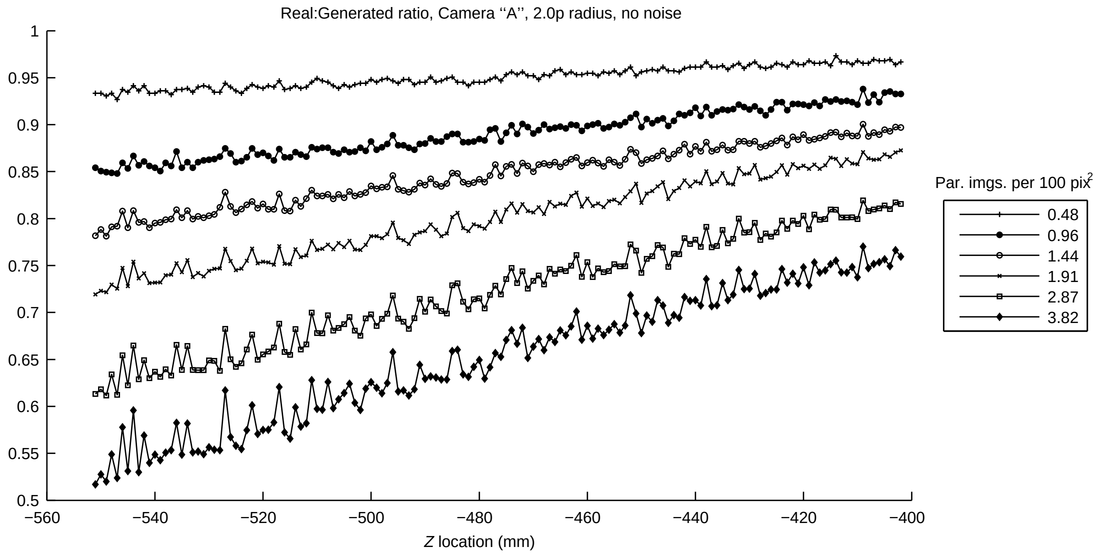


Figure 14.4-5: Ghost distribution for camera “A” and PTS, no noise, 0.75 tol.

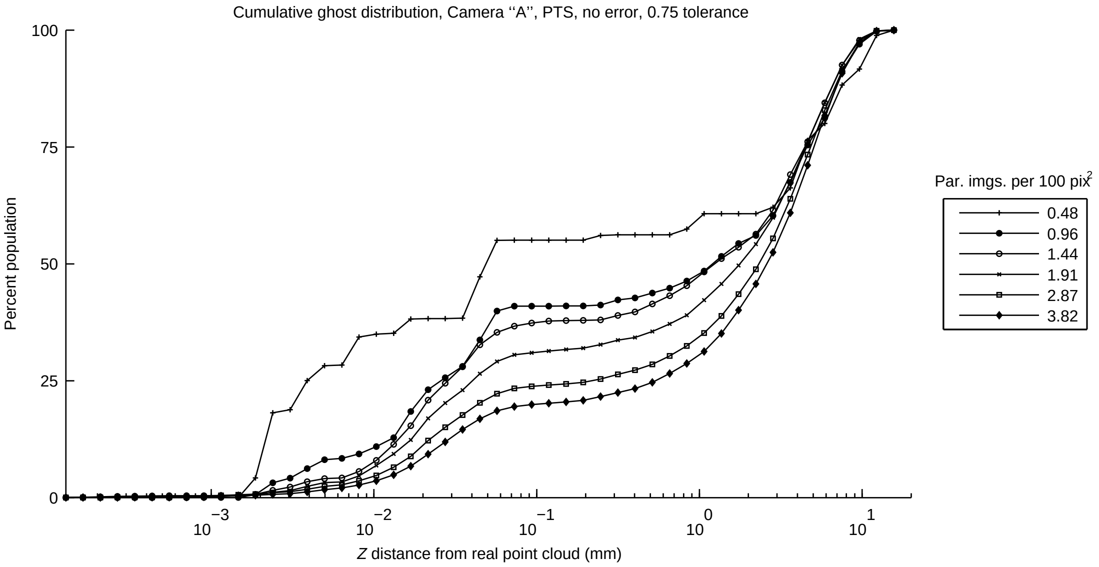
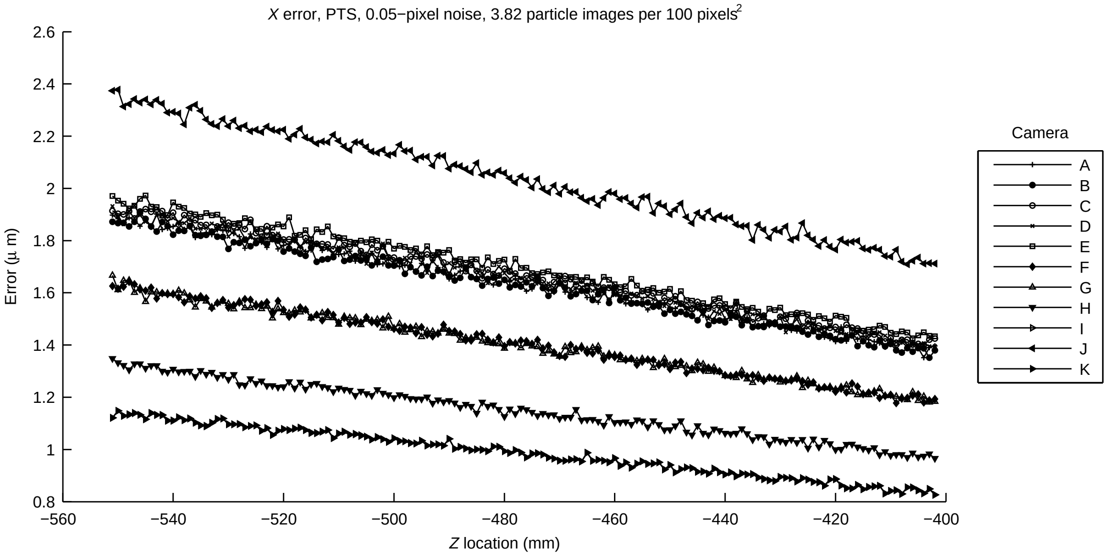


Figure 14.4-5 shows that as the density increases, so does the proportion of random ghosts relative to clump ghosts; if we choose (arbitrarily) 10 µm as the cut-off distance between clump ghosts and random ghosts, the population of clump ghosts drops by nearly 35% in proportion to the total ghosts.

14.5 Effect of Number of Apertures and Aperture Layout

As concluded in section 5.2, the reconstruction accuracy is dependent both on the average sensitivity of the aperture pairs and the number of apertures (and the coordinate of the point in question, of course).

Figure 14.5-1: Error in X for PTS, 0.05p noise and 3.82 particles images per 100 pixels^2



The aperture layouts can be separated into clumps according to the X error, as shown in figure 14.5-1. The first group, at the bottom of the graph (with the lowest mean error magnitude), consists of cameras “I” and “K”. Recall that these cameras have the exact same aperture layout, just a different aperture order so that their values of $\bar{s}_{ij}$ are not equal. As far as the X error goes, they are indistinguishable. Substituting the correct values for the constants in equation 5.2-15 we arrive at

$$\begin{aligned}\sigma_{X_P,\text{ “I”}} &= 1.83\times10^{-7}\sigma_{\text{img}}Z_P\sqrt{35286.6+4.97836X_P^2}\\ \sigma_{Y_P,\text{ “I”}} &= 1.83\times10^{-7}\sigma_{\text{img}}Z_P\sqrt{35286.6+4.97836Y_P^2}\end{aligned}\tag{14.5-1}$$

for camera “I” and

$$\begin{aligned}\sigma_{X_P,\text{ “K”}} &= 7.81\times10^{-8}\sigma_{\text{img}}Z_P\sqrt{192702+4.97836X_P^2}\\ \sigma_{Y_P,\text{ “K”}} &= 7.81\times10^{-8}\sigma_{\text{img}}Z_P\sqrt{192702+4.97836Y_P^2}\end{aligned}\tag{14.5-2}$$

for camera “K”. To ensure that the point clouds always fit in the field of view of the cameras, they were constrained to lie between (-30, 30) in X and (-25, 25) in Y. At these extremes, the error ratio in the two cameras is

$$\begin{aligned}\frac{\sigma_{X_P,\text{ “I”}}}{\sigma_{X_P,\text{ “K”}}} &= 1.04946\\ \frac{\sigma_{Y_P,\text{ “I”}}}{\sigma_{Y_P,\text{ “K”}}} &= 1.03484\end{aligned} \tag{14.5-3}$$

The next clump, with an error near 1.6 µm at $Z = -551.5$, consists of the two four-aperture cameras “F” and “G”. The reason the curves don’t like exactly atop each other, as they do in the case of Cameras “I” and “K”, is that, as mentioned above, the artificial noise induced in the input for cameras “F” and “G” is *not* identical.

The next, more loosely gathered clump consists of all the three-aperture cameras. On average the one with the highest error is camera “E”, which, surprisingly, has consistently higher error than camera “D” (which has the same $\bar{s}_{ij}$ but a different layout). The same relationship exists between “B” and “C”. This is due to a small subtlety visible in the error formula as seen in equation 5.2-14: the error in X is actually influenced by the average aperture X-coordinate $\bar{c}$. Both cameras “B” and “D” are symmetric across X, whereas “C” and “E” are not. Camera “A” lies between cameras “B” and “C” because it is symmetric along X and its $\bar{s}_{ij}$ is higher than that of cameras “D” and “E”.

Figure 14.5-2: Error in Y for PTS, 0.05p noise and 3.82 particles images per 100 pixels^2

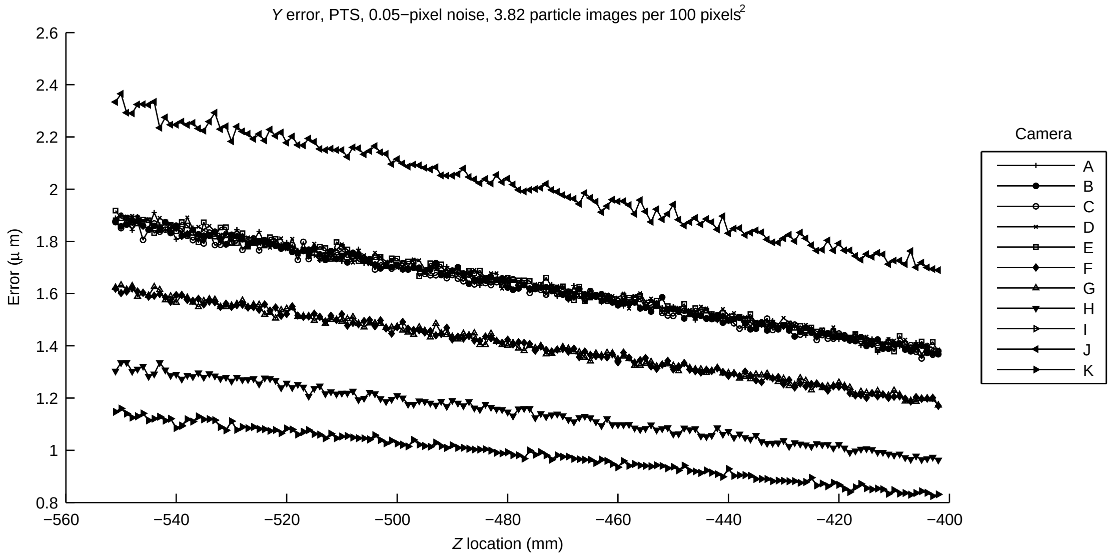


Figure 14.5-2 shows that the Y error is clumped similarly. Cameras “A” through “E” are now almost indistinguishable. In fact, Camera “A” is the only camera not symmetric on both axis, but its $\bar{d}$ value is only 0.0003, so it may as well be.

The plot of the average Z error magnitude (figure 14.5-3) reveals the pitfalls in the assumptions of Kajitani and Dabiri [2005] and the ones initially used in section 5.2. Even though the aperture order for Camera “K” results in a much higher average sensitivity coefficient than that of Camera

"I", the error in Z is greater for Camera "K" than Camera "I" with identical induced error in the input images. If we write the error in the average separation as we did for a three-aperture camera in equation 5.2-20 for these two cameras and take the ratio, we get

$$\frac{\sigma_{\bar{b}_{ij},\text{ ``K''}}}{\sigma_{\bar{b}_{ij},\text{ ``I''}}} \approx 2.471 \tag{14.5-4}$$

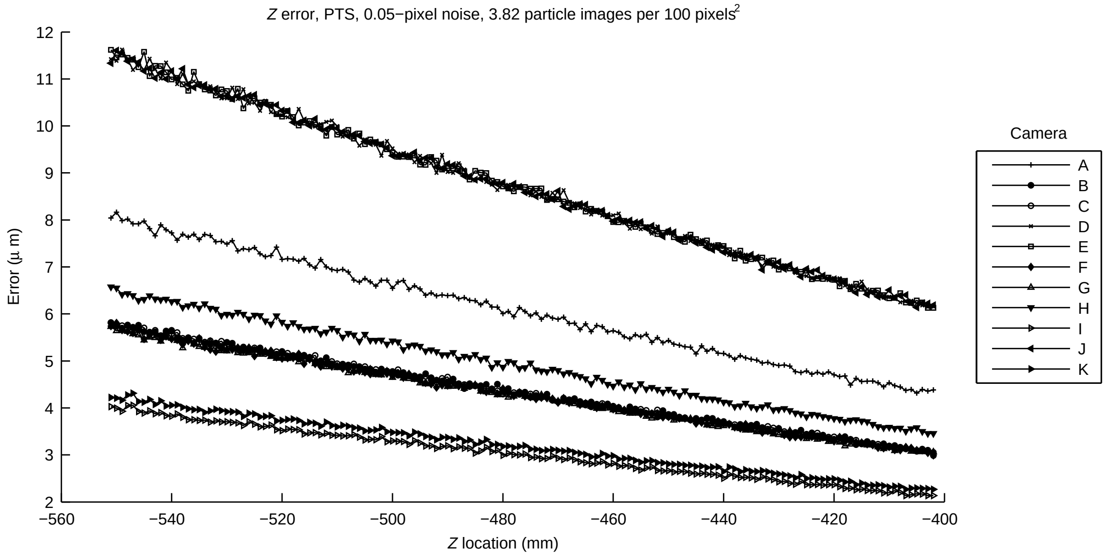
and thus

$$\frac{\sigma_{Z_P,\text{ ``K''}}}{\sigma_{Z_P,\text{ ``I''}}} = \frac{\sigma_{\bar{b}_{ij},\text{ ``K''}}}{\sigma_{\bar{b}_{ij},\text{ ``I''}}}\frac{\bar{s}_{ij},\text{ ``I''}}{\bar{s}_{ij},\text{ ``K''}} \approx 2.471 \times 0.4279 = 1.057 \tag{14.5-5}$$

which matches the simulations exactly.

Still, those two cameras lie at the bottom with the least error. The ones with the most error are cameras "J", "D", and "E", with "J" having only two apertures and "D" and "E" having the lowest average sensitivity coefficients of the three-aperture cameras. Next in order of decreasing error lie Camera "A" and Camera "H". The latter has twice the number of apertures as the former, yet the error is at most only 20% less. The next group contains all the remaining cameras, which are the "wide" three-aperture arrangements and the two four-aperture cameras (the latter, in the case of artificially induced error in PTS input, are identical to each other).

Figure 14.5-3: Error in Z for PTS, 0.05p noise and 3.82 particles images per 100 pixels^2



As far as reconstruction quality, the number of apertures has the greatest effect on the results; since ghost particles arise from random chance alone there is no dependency on aperture layout or ordering other than related to the proportions of the point cloud. For example, if the point cloud is a long, thin rectangle, a camera with its apertures laid out along the long side of the rectangle will generate more ghosts than one arranged along the short side.

The *total* number of ghosts is also affected by the layout because different layouts will produce different mappable regions, and of course no ghosts can be generated outside this region. This does not, however, have anything to do with the process that generates the ghosts; it only acts as an envelope to limit their possibility in space.

Figure 14.5-4: Ghosts to real ratio for PTS, no noise, 0.75 tol. and 3.82 particles images per 100 pixels^2

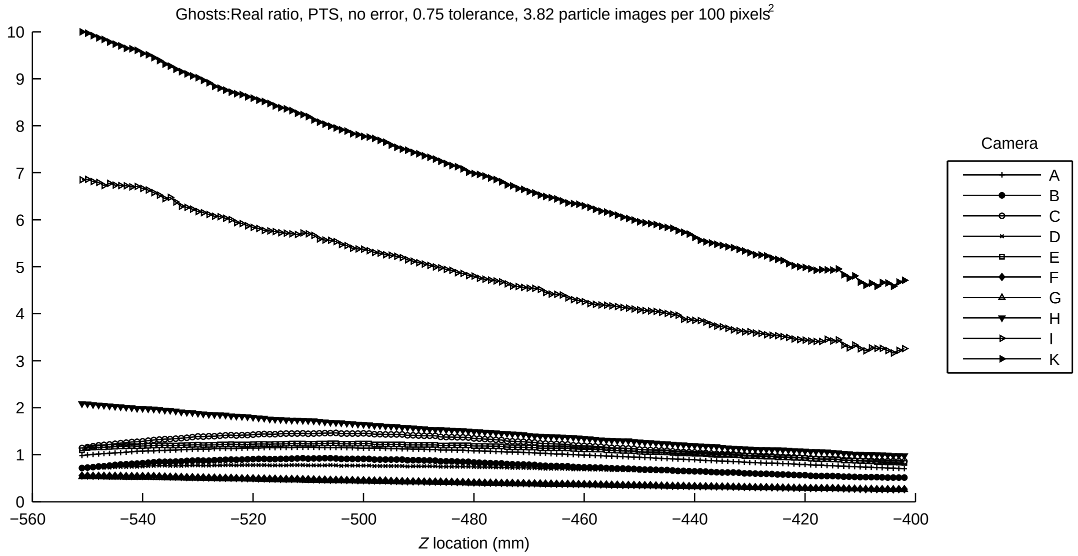


Figure 14.5-5: Ghosts to real ratio for camera “J” and PTS, no noise, 0.75 tol.

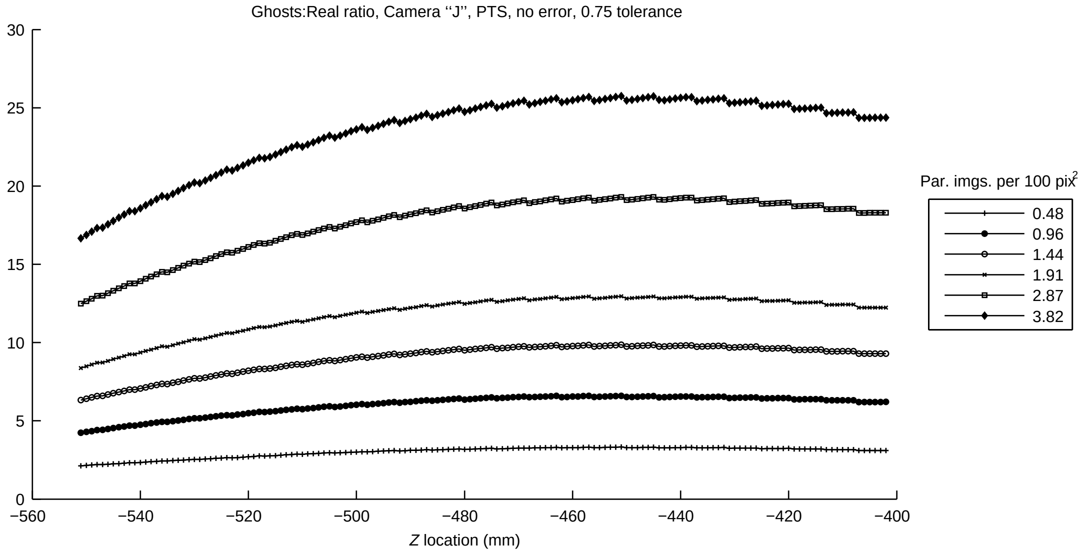


Figure 14.5-4 shows the proportion of ghosts generated at the highest density for perfect PTS

files. Camera “J” is pictured separately in figure 14.5-5 since the proportion is so much greater. Recall from section 5.3 that there are two types of ghosts: those that arise from the particle images being packed too densely (clump ghosts), and those that arise from random chance alone (which is, of course, also affected by density). The analysis presented for clump ghosts is close to the data. In the range of simulations performed, we cover both ends of the ghost spectrum: with Camera “J” we present situation in which the probability of random ghosts is high since the matching criterium is simply “two particle images along the direction connecting the two apertures”. Camera “I” (and “K”) represent the other end, where the chance of random ghosts is infinitessimal (due to the cameras’ having 8 apertures) but the number of clump ghosts that can be generated is immense (according to the exponential estimate presented in section 5.3).

Although the proportion of ghosts is a usable quantity, the distribution of the ghosts in space is more important. Still, the data should be interpreted carefully, because the simulations are of thin sheets of particles, whereas most experiments of interest are in volumes. The exponential rule for estimating clump ghosts (equation 5.3-27) is only valid for thin sheet clouds which have the same relative particle image arrangement in each aperture. If the cloud is volumetric, then the relative particle image arrangement will be different in each aperture (since the shift of matching particle images is related to the Z-coordinate of the particle that generated them), so a clump of k particle images within the matching tolerance in one aperture is most likely to form only $k-1$ ghosts which will have a similar value of $\bar{b}_{ij}$, $\bar{x}$, and $\bar{y}$ to the real particle and thus be close to it in the final cloud.

The ghost distribution as a function of camera apertures, as predicted by figure 5.3-2, indicates that with two apertures, almost all the ghosts are random while with three apertures the proportion almost completely reverses, and already at 4 apertures there are almost no random ghosts. The reality of a two-aperture system is trivial, that is, it is inconceivable that the random ghosts not dominate. This is shown in figure 14.5-6; the distribution is nearly invariable to density.

One three-aperture case was already shown in figure 14.4-5. The symmetric in-line arrangements are surprisingly susceptible to clump ghosts (using the cut off at 10 µm as before). In fact, at the lowest density, both cameras “B” and “D” exhibit a sort of “dead-man’s zone” between 10 µm and 1 mm, with no additional ghosts, that divides the ghost population almost evenly into two halves (see figure 14.5-7). As the density increases the transition is more gradual, but there is still a surprisingly non-diminutive percentage of ghosts lying within 1 µm of the sheet. One should keep in mind that ghost distribution is also dependent (and heavily so) on the actual arrangement of the particle images and on the relationship between the image geometry and the aperture layout because the intersection between the search path and the available particle image space will vary in length as the aperture layout varies.

The four-aperture case of figure 14.5-8 (although there is some difference between the distribution of cameras “F” and “G” due to relative orientation) supports the conclusion from figure 5.3-2 that

Figure 14.5-6: Ghost distribution for camera “J” and PTS, no noise, 0.75 tol.

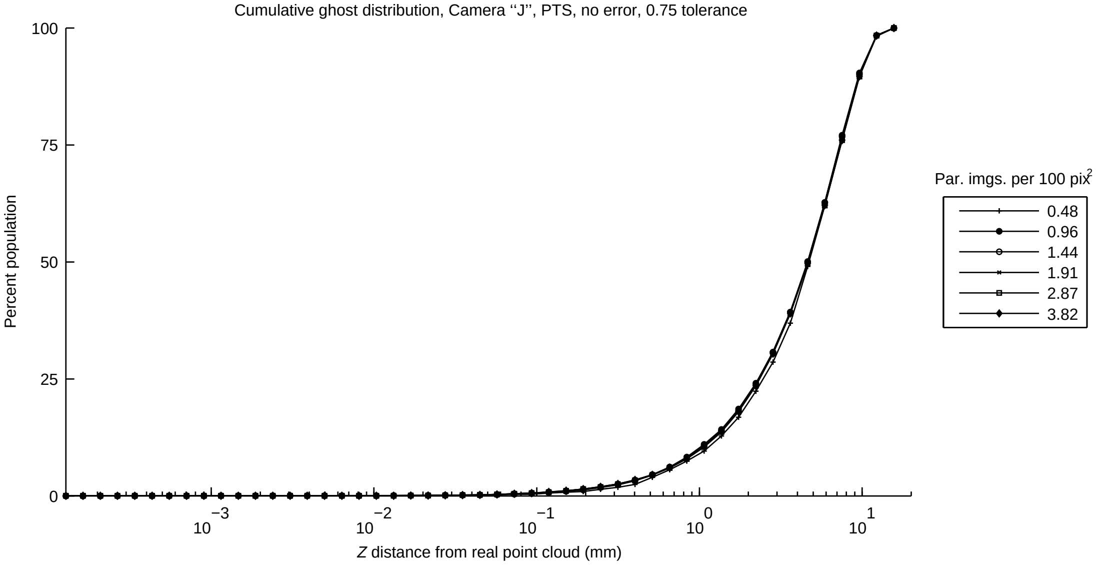
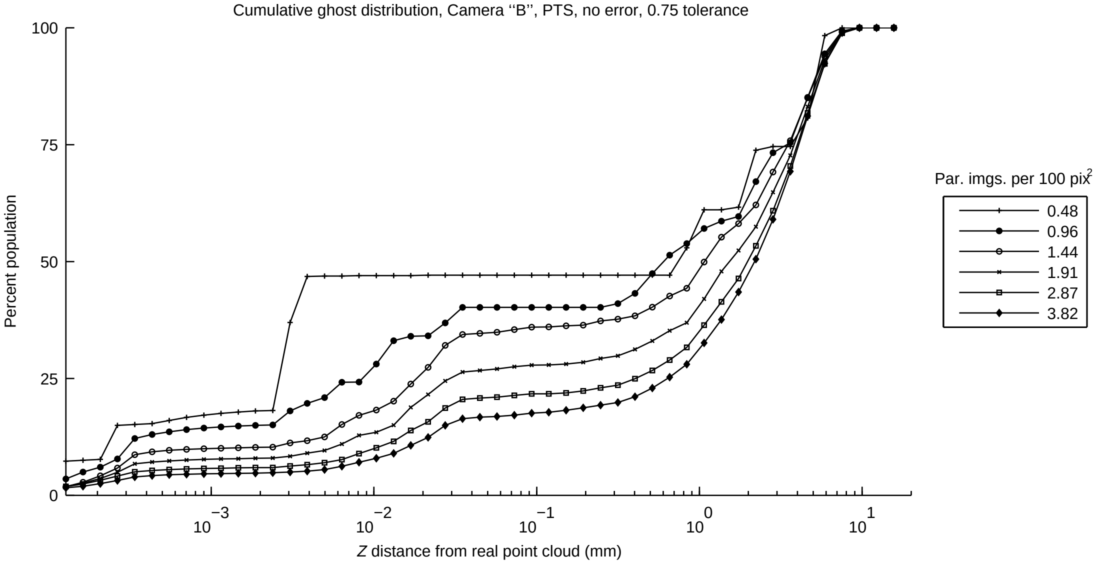
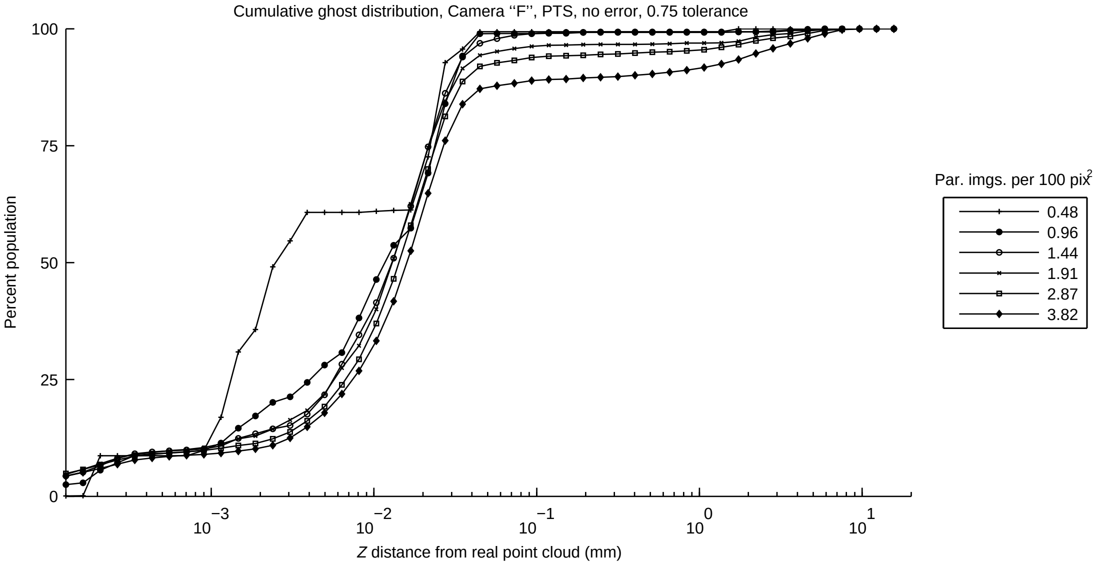


Figure 14.5-7: Ghost distribution for camera “B” and PTS, no noise, 0.75 tol.



with four apertures most of the ghosts are due to clumps.

Figure 14.5-8: Ghost distribution for camera "F" and PTS, no noise, 0.75 tol.



14.6 Effect of Gaussian Fitting Algorithm

The simulations presented above did not involve the Gaussian fitting algorithm as all the input "images" were just PTS files. By using actual bitmaps as input, particle images that occupy a finite amount of space are introduced (so that clumping is less likely) as are any effects due to the Gaussian fitting algorithm—such as dependent error in x and y, discretization effects, and differences due to the orientation of the particle image separation direction with respect to the pixel grid.

Figure 14.6-1 shows the X error in Camera "A" at 1.44 particle images per 100 pixels squared and figure 14.6-2 shows the same results at double the density. The average magnitude of the error in the PTS inputs remains the same since the particle images in that case are discrete points and thus impervious to overlap issues. The two bitmap image cases are obviously affected by overlap, as the error doubles between the two cases. The error also fluctuates a lot more between planes; this is evidently due to the effect of a discretized domain. Comparing the histograms in figures 12.3-2 and 12.4-9, it seems that the zero-noise case has a substantial advantage in terms of the error distribution, but this advantage is all but lost at the lower density of figure 14.6-1 and completely inconsequential at the density of figure 14.6-2.

The error statistics in section 12.6.1 suggest that the Gaussian fitting algorithm exercised on 2.0-pixel-radius particle images (which were deemed to provide the most consistent performance) does not result in independent error in x and y as assumed in section 5.2. This is also visible in

Figure 14.6-1: Error in X for camera “A” and 1.44 particle images per 100 pixels^2

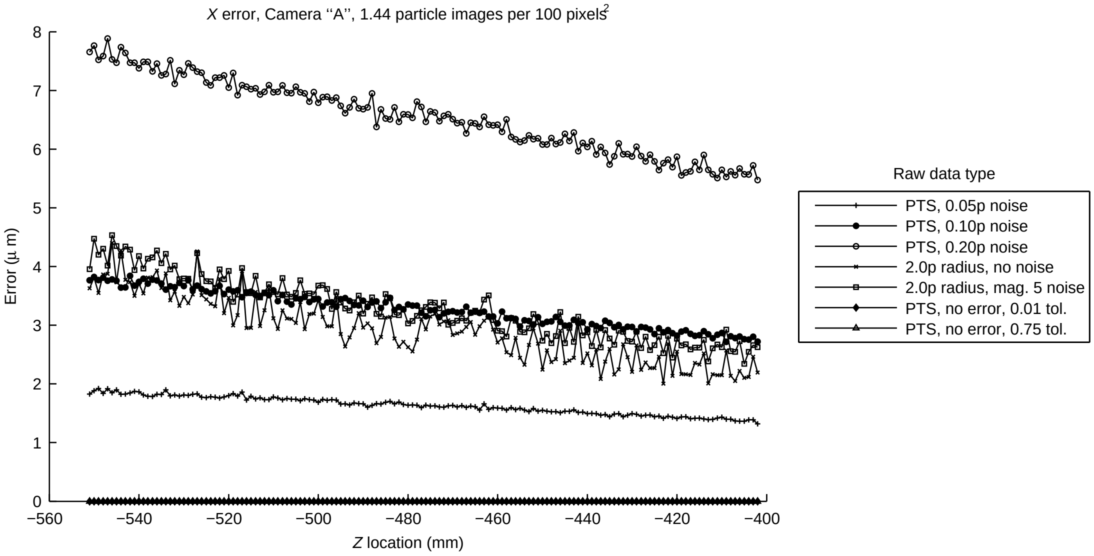
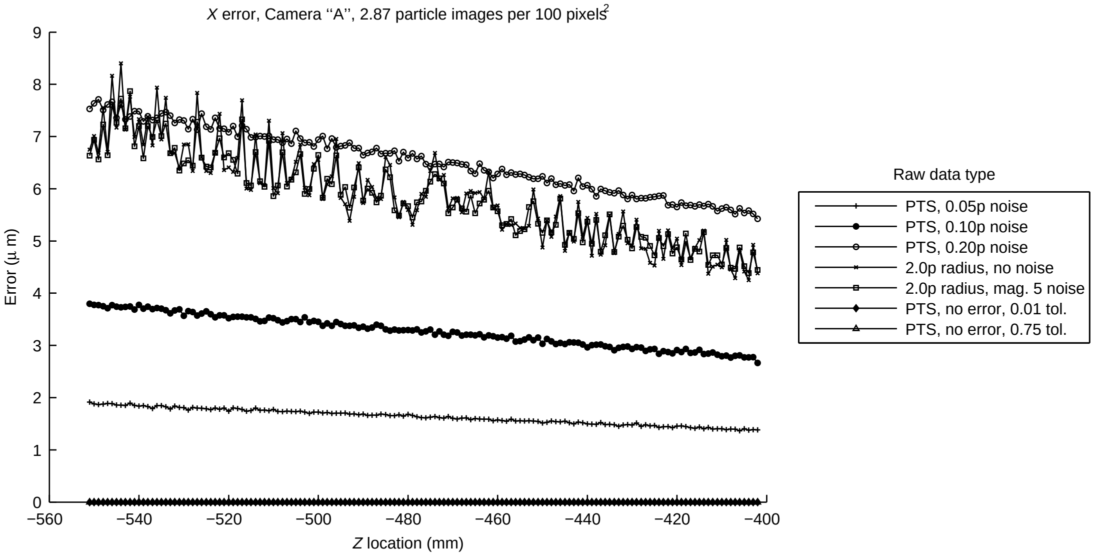


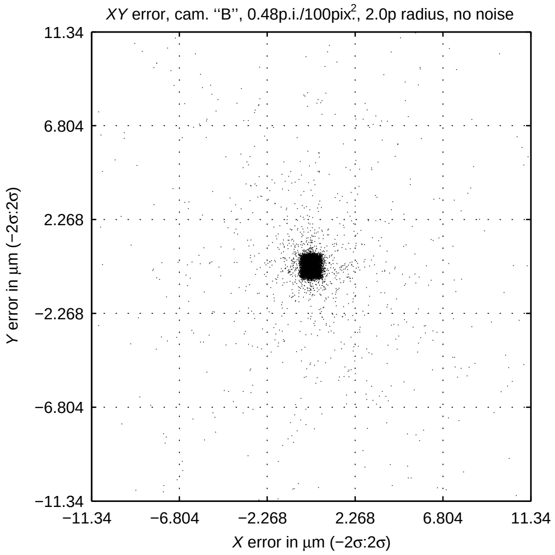
Figure 14.6-2: Error in X for camera “A” and 2.87 particles image per 100 pixels^2



the scatter plot of figure 12.4-6. This is a manifestation of the discretization effect, and so the Gaussian fitting error will manifest itself differently in cameras with different sensor orientations and arrangements.

The ideal test case for this is a comparison of cameras “B” and “C”, which are identical except for the asymmetrical arrangement of Camera “C”. This asymmetry helps to misalign any discretization effects so that the error biases do not coincide at particular Z coordinates. The scatter plot shows how Camera “C” (figure 14.6-4) has a noticeably narrower X error distribution than does camera “B” (figure 14.6-3).

Figure 14.6-3: X versus Y error for camera “B”, 0.48 particles images per 100 pixels^2, 2.0p radius, no noise



Even a minute amount of noise will all but destroy the bias in the error of the Gaussian fitting algorithm (figure 12.4-10) and this propagates directly into a camera’s X and Y error distribution.

The cumulative distribution of the error most obviously shows the effects of the Gaussian fitting algorithm. Using `PTS` files as input, with pseudo-random error introduced, the cameras rank as expected, with those with the highest number of apertures having the narrowest error distributions (figures 14.6-6 and 14.6-7).

With bitmaps as input, the situation changes. The biased error produced by the zero-noise images completely eliminates any advantage with number of apetures in the in-plane error(figure 14.6-8) and completely re-arranges the ranking of the cameras in terms of Z error—with several three-aperture arrangements performing better than the eight-aperture ones (figure 14.6-9). As the density increases and the bias is coupled with overlap, the six and eight-aperture arrangements become the worst performers, with Camera “B” showing a distinguishable superiority over all other arrangements.

Adding even slight noise to the image brings the simulated performance closer to predictions, and

Figure 14.6-4: X versus Y error for camera "C", 0.48 particles images per 100 pixels^2, 2.0p radius, no noise

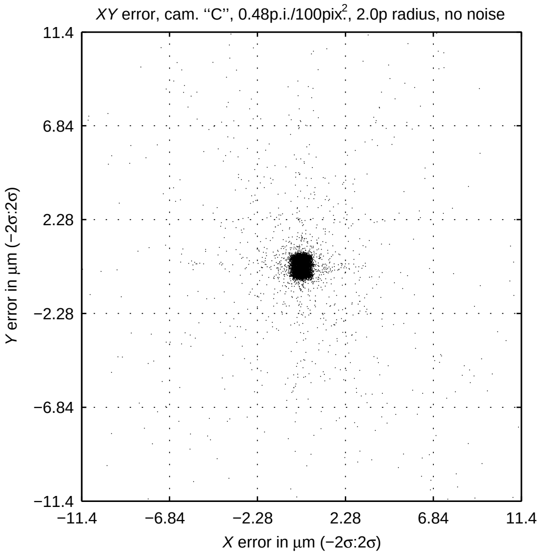


Figure 14.6-5: X versus Y error for camera "G", 0.48 particles images per 100 pixels^2, 2.0p radius, magnitude 5 noise

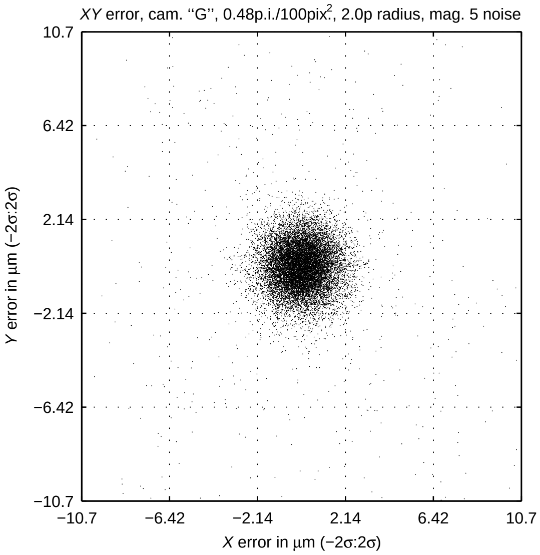


Figure 14.6-6: *X* error percentile for PTS, 0.05p noise and 0.48 particles images per 100 pixels^2

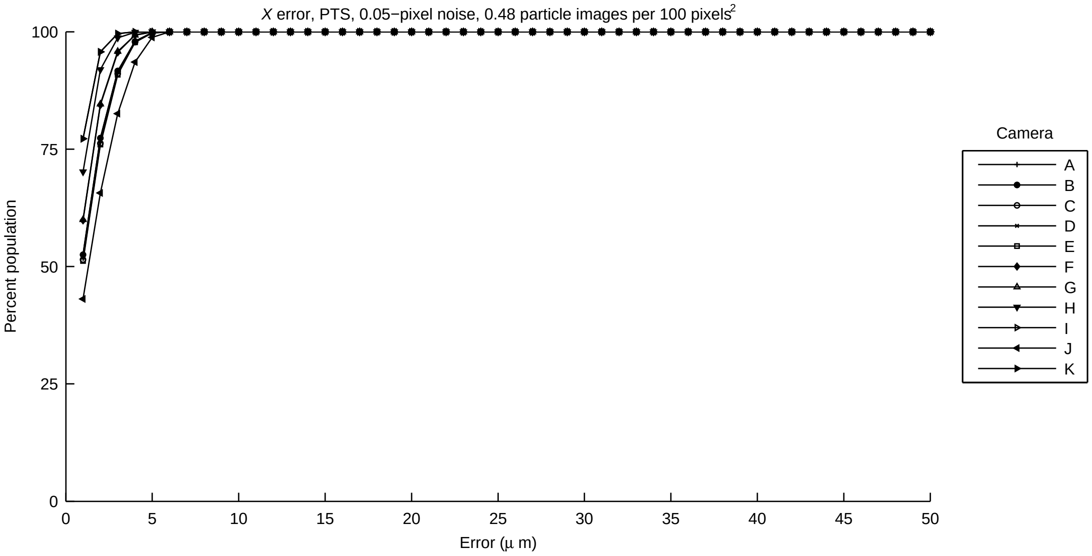


Figure 14.6-7: *Z* error percentile for PTS, 0.05p noise and 0.48 particles images per 100 pixels^2

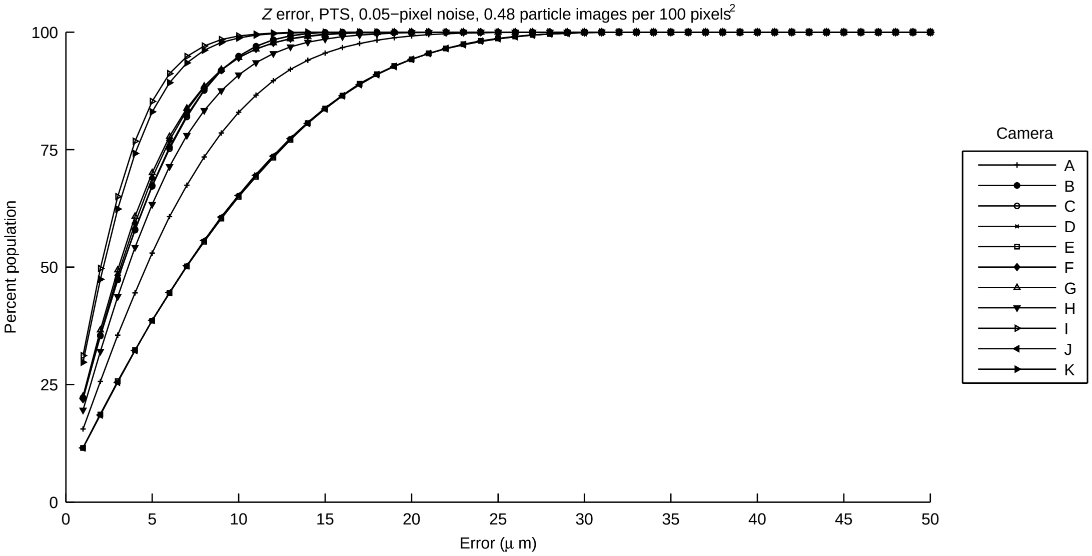


Figure 14.6-8: X error percentile for 2.0p radius, no noise and 0.48 particles images per 100 pixels^2

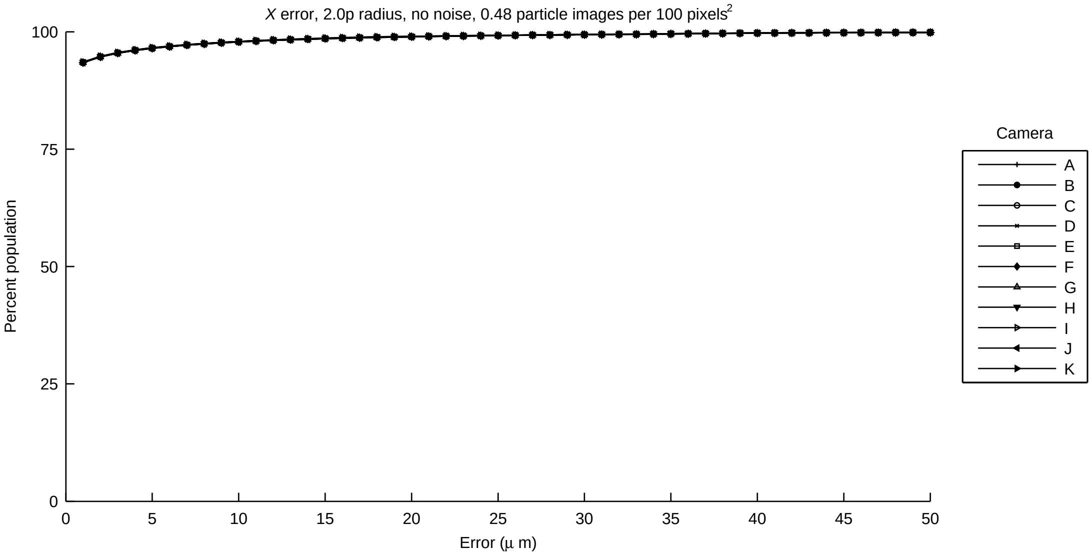
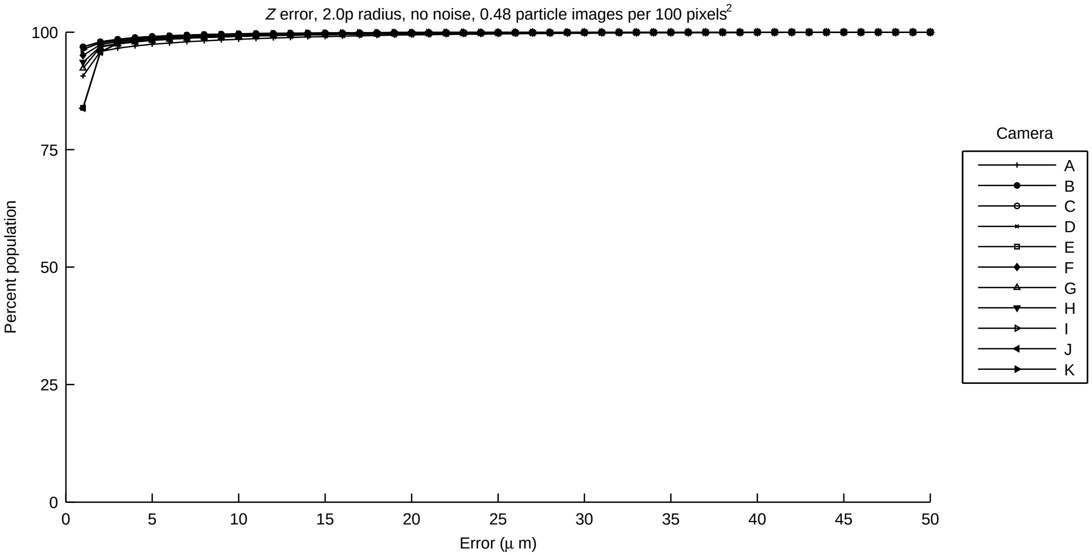


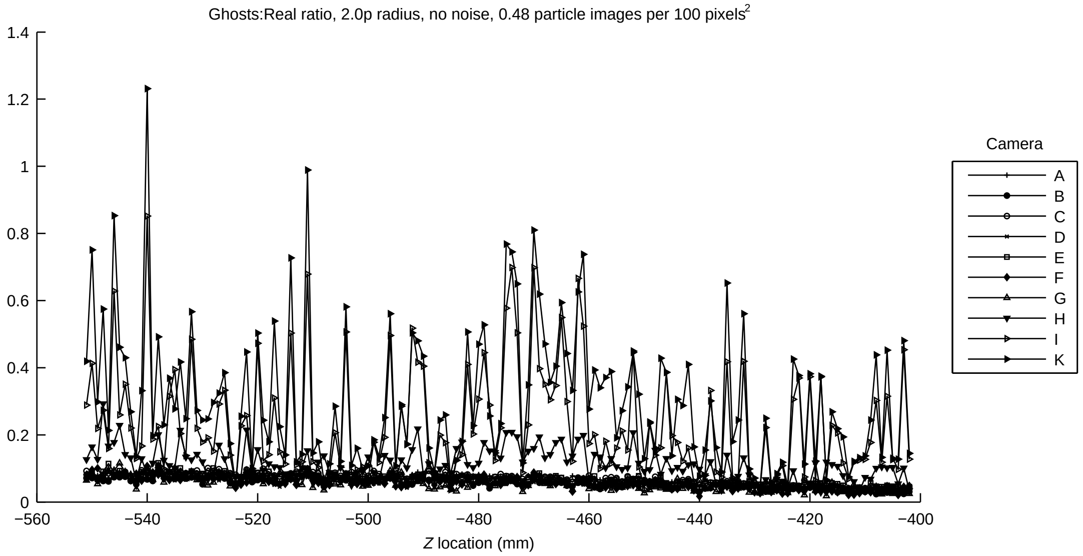
Figure 14.6-9: Z error percentile for 2.0p radius, no noise and 0.48 particles images per 100 pixels^2



the error distribution in Z is substantially narrower across the board than that of artificially-induced 0.05-pixel noise.

Ghost production with bitmap inputs is much less absurd than in the PTS case. As mentioned before, when particle images occupy finite space there is a reduced chance of clumping. There is little difference in ghost production between images with no noise and those with slight noise added (though the difference increases with density, see figures 14.6-11 and 14.6-12), but across all densities there is an obvious disadvantage to having eight apertures; even though the possibility of clumping is lower, its effects are still strong. At the lower densities, the six and eight-aperture arrangements clearly produce more ghosts (figure 14.6-10).

Figure 14.6-10: Ghosts to real ratio for 2.0p radius, no noise and 0.48 particle images per 100 pixels^2



At the high extreme of density, the performance of the arrangements is a bit more separated (figure 14.6-11); the six and eight-aperture arrangements are not the worst performers but are still the most irregular. Perhaps somewhat surprising is that the best performers are the two four-aperture arrangements, cameras “F” and “G”, both in the no-noise and slight-noise cases. This could be explained by the fact that any bias in the error of the Gaussian fitting is perfectly aligned in these cameras because the particle image separation magnitude and direction relative to the pixel plane is symmetric across both axes. The six and eight-aperture cameras are the next best performers *most of the time*, however they frequently perform worse than other, more consistent arrangements. Also surprising is that the three-aperture arrangements are split into two groups, between which there is a substantial difference in ghost production. The better group consists of cameras “B” and “D”—the two symmetric arrangements—and the worse group contains cameras “A”, “C”, and “E”, with Camera “A” performing best most of the time. The two groups are separated by symmetry and thus it is likely that the difference in performance is due to alignment of the bias.

Figure 14.6-11: Ghosts to real ratio for 2.0p radius, no noise and 3.82 particle images per 100 pixels^2

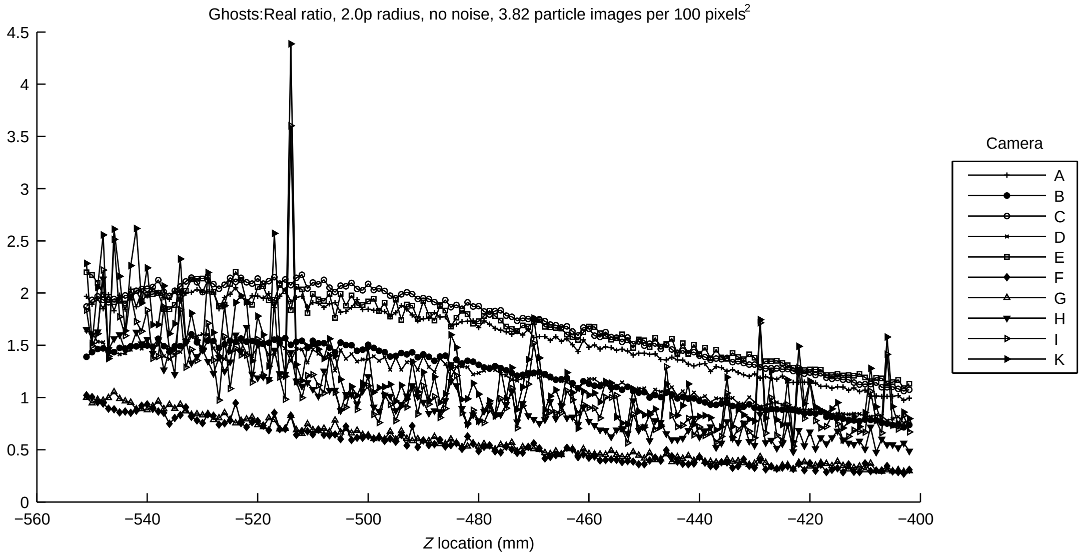


Figure 14.6-12: Ghosts to real ratio for 2.0p radius, magnitude 5 noise and 3.82 particle images per 100 pixels^2

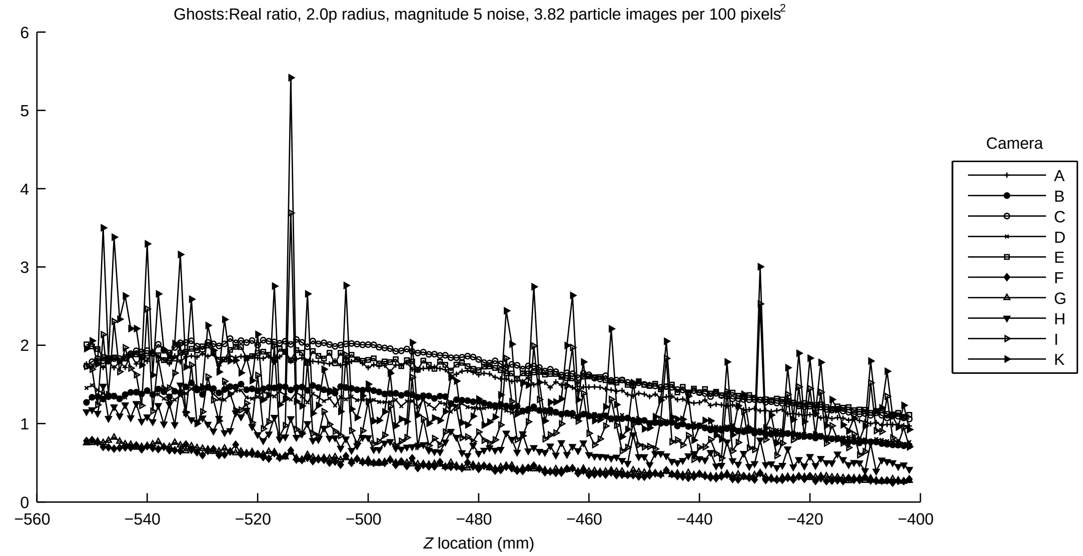
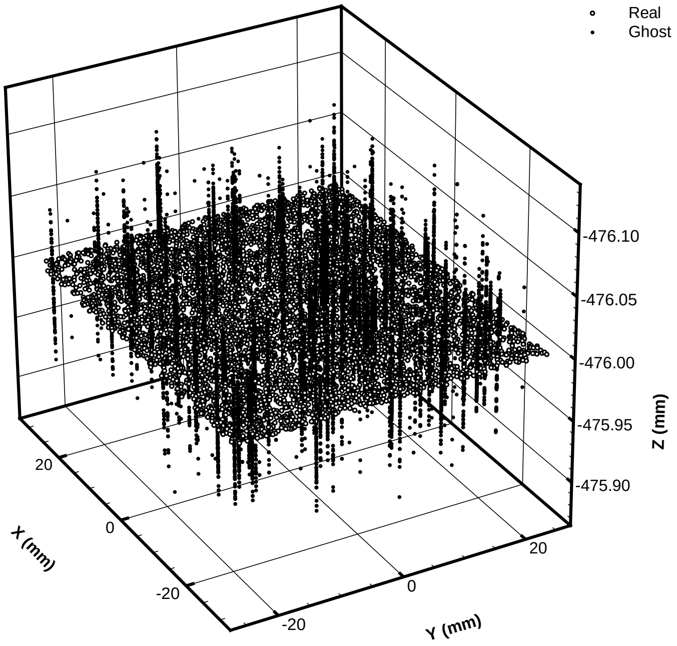


Figure 14.6-13 shows exactly what is meant by clump ghosts. Near two or three real particles that are really close together, a "shotgun blast" of ghosts is formed. Because the original particle images were within tolerance of each other, the resulting X, Y, Z-coordinates of the ghosts are extremely close to that of the real particle. To contrast, the same frame is shown as a result of bitmap input in figure 14.6-14. With particle images occupying finite space, it is harder for two Gaussian-fitting

results to lie within tolerance of each other[2], so the ghosts are a bit more separated. In the PTS case the density of these ghosts is so high that they can be easily filtered out by keeping only the particles that have k neighbors or less within some region R. In the bitmap case it is not as clear-cut, and filtering in this way could remove a considerable number of real particles, too.

Figure 14.6-13: Point cloud sample, Camera "I", perfect PTS input, 3.82 particle images per 100 pixels^2. Note the layout of the clump ghosts (Z axis is magnified 200 times).



The seemingly drammatic difference between figures 14.6-13 and 14.6-14 appears much more insignificant when viewed in figure 14.6-15. Any sort of noise in the particle image locations, whether artificially induced as in the PTS case or as a result of Gaussian fitting, works to reduce the number of extremely close clump ghosts. The effect can also be seen in the case of the inline cameras, which on perfect images show an enormous number of close clump ghosts, but with added noise show a more reasonable distribution.

14.7 Conclusion of Simulations

The simulations verified some subtleties that arose from the error analysis of section 5.2, principally that in-plane error is minimized for symmetrical cameras and that the Z error depends on aperture

[2] With real images, the primary source of clump ghosts are saturated particle images, which may result in several "particle images" due to the tresholding step of the Gaussian fitting algorithm.

Figure 14.6-14: Point cloud sample, Camera "I", perfect bitmap input, 3.82 particle images per 100 pixels^2. Note the seeming lack of clump ghosts (Z axis is magnified 200 times).

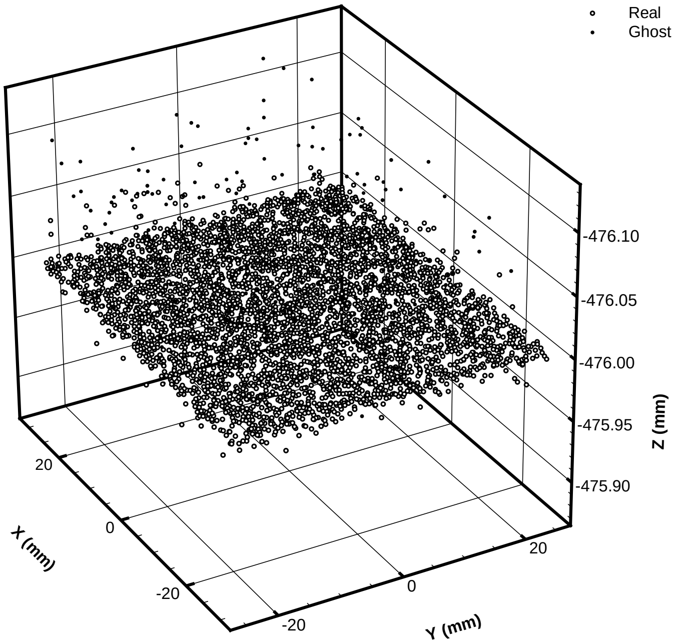
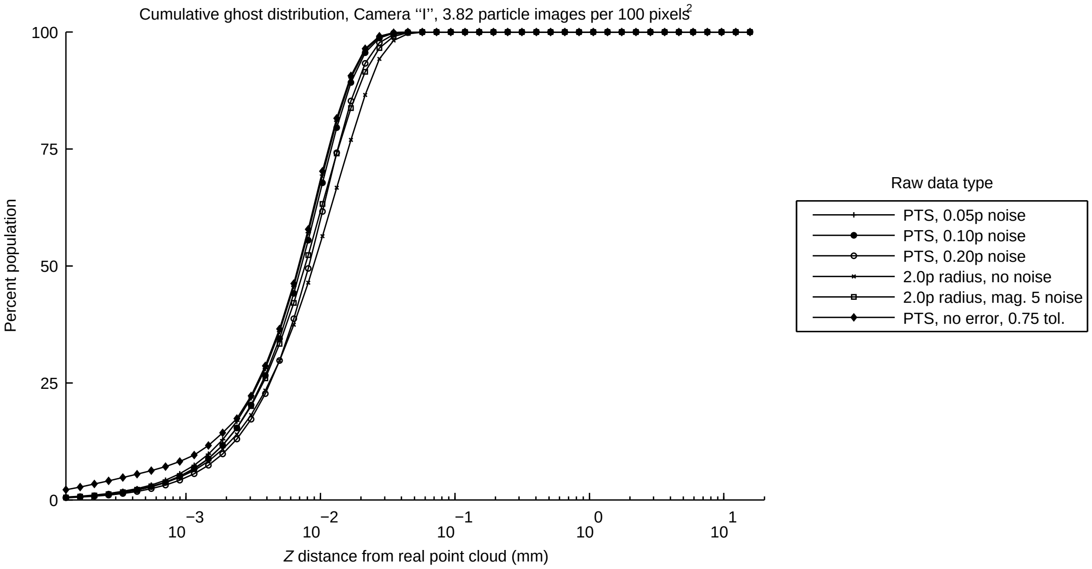


Figure 14.6-15: Ghost distribution for camera "I" and 3.82 particles images per 100 pixels^2



ordering during processing. The error introduced by the Gaussian fitting algorithm is non-trivial in that it can overpower the theoretical advantage of one aperture arrangement over another.

The simulations were performed with thin-sheet point clouds, so the reconstruction quality results cannot directly be applied to the volume case. The proportion of random and clump ghosts was verified, but it should be reiterated that clump ghosts, as described here, are really a phenomenon of sheet clouds and, in general, can either be filtered out by a population density filter or, if left alone during the particle tracking step, will probably generate only small amounts of error in the velocity (since clump ghosts are physically close to real particles). In light of this, the three-aperture arrangement may in fact not be the ideal camera in terms of ghosts generated for the general experimental situation—a four aperture camera will offer a better reconstruction quality—but it should be considered that this would come at a 33% increase in cost and all the issues involved in adding more data—the need for bigger storage, faster networks, and faster processors.

It should also be noted that all the simulations were performed using pinhole optics, and that in practice, cameras "'B", "C", and possibly "F" and "G" would not be feasible because the required offset between lens and sensor would introduce too much distortion and too much light fall-off. The in-line cameras are optically superior because the sensor shift is in one direction only and thus the distance between the center of the lens and the farthest corner of the sensor is minimized.

14.7.1 Error Statistics

Table 14.7-2 shows the mean and variance for each condition simulated. The last column is frequently mentioned in the literature; it is the square of the ratio of the out-of-plane to the in-plane error. For all PTS test cases it agrees well with the theory, and the least expensive way to decrease the ratio is by spreading the apertures apart as in cameras "B" and "C". It should be noted that for bitmap input, claims that the cameras are more precise in-plane than out-of-plane are evidently false; this is presumably due to the biased errors introduced by the Gaussian fitting algorithm, which is also supported by the fact that the means of the errors for bitmap inputs are often orders of magnitude greater than their PTS counterparts. This conjecture should be interpreted with care because with bitmap input there is no way to classify particles as being real or ghosts other than by their final location relative to the generated cloud[3].

[3] In fact, the method failed for the eight-aperture cameras with perfect bitmap input, which produced many ghosts well within the 70 µm separation criterium and resulted in many of these being counted as real particles.

Table 14.7-2: Error statistics for the reconstruction simulations.

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
"A", PTS-0.05p, 0.48	112,500	4.34×10^{-6}	-7.30×10^{-7}	1.56×10^{-5}	4.08×10^{-6}	4.06×10^{-6}	5.89×10^{-5}	7.24
"A", PTS-0.10p, 0.48	112,500	-1.61×10^{-5}	-1.02×10^{-5}	1.59×10^{-5}	1.63×10^{-5}	1.61×10^{-5}	2.35×10^{-4}	7.28
"A", PTS-0.20p, 0.48	112,498	-2.62×10^{-5}	5.66×10^{-6}	1.03×10^{-4}	6.51×10^{-5}	6.46×10^{-5}	9.35×10^{-4}	7.23
"A", 2.0p-0, 0.48	106,956	-4.73×10^{-5}	-9.39×10^{-5}	1.10×10^{-5}	1.83×10^{-5}	2.80×10^{-5}	9.65×10^{-6}	1.77×10^{-1}
"A", 2.0p-5, 0.48	106,829	-5.00×10^{-6}	-5.16×10^{-5}	1.72×10^{-5}	1.82×10^{-5}	2.81×10^{-5}	2.83×10^{-5}	5.39×10^{-1}
"A", PTS-0.05p, 0.96	225,000	3.13×10^{-7}	-5.04×10^{-6}	-2.31×10^{-5}	4.07×10^{-6}	4.07×10^{-6}	5.85×10^{-5}	7.19
"A", PTS-0.10p, 0.96	225,000	3.71×10^{-6}	1.31×10^{-5}	-5.89×10^{-5}	1.63×10^{-5}	1.62×10^{-5}	2.34×10^{-4}	7.22
"A", PTS-0.20p, 0.96	224,992	1.08×10^{-6}	6.88×10^{-6}	4.62×10^{-5}	6.53×10^{-5}	6.47×10^{-5}	9.41×10^{-4}	7.25
"A", 2.0p-0, 0.96	200,381	-1.65×10^{-4}	-1.84×10^{-4}	1.80×10^{-5}	5.35×10^{-5}	5.83×10^{-5}	2.82×10^{-5}	2.35×10^{-1}
"A", 2.0p-5, 0.96	199,179	-7.21×10^{-5}	-1.09×10^{-4}	1.70×10^{-5}	4.90×10^{-5}	5.59×10^{-5}	4.34×10^{-5}	3.92×10^{-1}
"A", PTS-0.05p, 1.44	337,500	-2.14×10^{-6}	4.03×10^{-7}	-2.85×10^{-6}	4.06×10^{-6}	4.06×10^{-6}	5.86×10^{-5}	7.23
"A", PTS-0.10p, 1.44	337,500	5.31×10^{-6}	-9.19×10^{-6}	-6.37×10^{-7}	1.63×10^{-5}	1.62×10^{-5}	2.34×10^{-4}	7.22
"A", PTS-0.20p, 1.44	337,491	-3.53×10^{-6}	1.24×10^{-5}	9.28×10^{-6}	6.55×10^{-5}	6.51×10^{-5}	9.37×10^{-4}	7.17
"A", 2.0p-0, 1.44	285,482	-6.47×10^{-5}	-2.24×10^{-4}	3.02×10^{-5}	8.22×10^{-5}	7.86×10^{-5}	3.80×10^{-5}	2.30×10^{-1}
"A", 2.0p-5, 1.44	283,253	5.94×10^{-6}	-1.39×10^{-4}	2.54×10^{-5}	7.42×10^{-5}	7.38×10^{-5}	5.36×10^{-5}	3.59×10^{-1}
"A", PTS-0.05p, 1.91	450,000	-4.36×10^{-8}	-1.19×10^{-7}	2.89×10^{-6}	4.07×10^{-6}	4.05×10^{-6}	5.86×10^{-5}	7.22
"A", PTS-0.10p, 1.91	450,000	7.11×10^{-6}	-4.70×10^{-6}	-9.98×10^{-6}	1.63×10^{-5}	1.63×10^{-5}	2.34×10^{-4}	7.20
"A", PTS-0.20p, 1.91	449,986	-4.61×10^{-6}	1.77×10^{-5}	1.02×10^{-4}	6.52×10^{-5}	6.47×10^{-5}	9.40×10^{-4}	7.24
"A", 2.0p-0, 1.91	360,063	-1.28×10^{-4}	-2.52×10^{-4}	2.97×10^{-5}	1.18×10^{-4}	1.03×10^{-4}	5.37×10^{-5}	2.41×10^{-1}
"A", 2.0p-5, 1.91	355,456	-1.42×10^{-5}	-1.30×10^{-4}	2.86×10^{-5}	1.05×10^{-4}	9.40×10^{-5}	6.61×10^{-5}	3.37×10^{-1}
"A", PTS-0.05p, 2.87	675,000	-2.41×10^{-6}	4.81×10^{-7}	1.05×10^{-5}	4.07×10^{-6}	4.05×10^{-6}	5.87×10^{-5}	7.24
"A", PTS-0.10p, 2.87	675,000	3.24×10^{-6}	-2.55×10^{-6}	-7.30×10^{-6}	1.63×10^{-5}	1.62×10^{-5}	2.35×10^{-4}	7.24
"A", PTS-0.20p, 2.87	674,953	-8.94×10^{-6}	2.35×10^{-6}	1.80×10^{-5}	6.53×10^{-5}	6.47×10^{-5}	9.37×10^{-4}	7.21
"A", 2.0p-0, 2.87	487,654	-2.70×10^{-4}	-2.18×10^{-4}	2.71×10^{-5}	1.78×10^{-4}	1.62×10^{-4}	8.67×10^{-5}	2.57×10^{-1}
"A", 2.0p-5, 2.87	476,711	-1.18×10^{-4}	-8.80×10^{-5}	5.70×10^{-5}	1.59×10^{-4}	1.44×10^{-4}	9.46×10^{-5}	3.19×10^{-1}
"A", PTS-0.05p, 3.82	900,000	-1.05×10^{-6}	-1.18×10^{-7}	3.77×10^{-6}	4.08×10^{-6}	4.05×10^{-6}	5.86×10^{-5}	7.21
"A", PTS-0.10p, 3.82	900,000	-8.16×10^{-6}	-9.25×10^{-7}	1.33×10^{-5}	1.63×10^{-5}	1.62×10^{-5}	2.34×10^{-4}	7.22
"A", PTS-0.20p, 3.82	899,897	5.07×10^{-6}	5.77×10^{-6}	5.51×10^{-5}	6.53×10^{-5}	6.49×10^{-5}	9.37×10^{-4}	7.18

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
“A”, 2.0p-0, 3.82	586,732	-2.12×10^{-4}	-2.61×10^{-4}	5.52×10^{-5}	2.43×10^{-4}	2.29×10^{-4}	1.26×10^{-4}	2.65×10^{-1}
“A”, 2.0p-5, 3.82	564,356	-8.77×10^{-5}	-9.90×10^{-5}	6.44×10^{-5}	2.12×10^{-4}	1.96×10^{-4}	1.26×10^{-4}	3.09×10^{-1}
“B”, PTS-0.05p, 0.48	110,819	9.33×10^{-6}	-9.39×10^{-6}	-1.04×10^{-5}	4.08×10^{-6}	4.02×10^{-6}	2.95×10^{-5}	3.63
“B”, PTS-0.10p, 0.48	110,819	9.40×10^{-6}	-6.41×10^{-6}	3.14×10^{-5}	1.62×10^{-5}	1.61×10^{-5}	1.18×10^{-4}	3.66
“B”, PTS-0.20p, 0.48	110,701	2.16×10^{-6}	1.46×10^{-5}	7.36×10^{-5}	6.45×10^{-5}	6.48×10^{-5}	4.75×10^{-4}	3.65
“B”, 2.0p-0, 0.48	105,329	-3.84×10^{-5}	-7.22×10^{-5}	-2.90×10^{-6}	1.89×10^{-5}	3.22×10^{-5}	1.98×10^{-6}	3.34×10^{-2}
“B”, 2.0p-5, 0.48	105,212	-5.39×10^{-6}	-2.90×10^{-5}	-5.76×10^{-7}	1.83×10^{-5}	3.16×10^{-5}	1.21×10^{-5}	2.14×10^{-1}
“B”, PTS-0.05p, 0.96	221,629	4.12×10^{-6}	-8.48×10^{-6}	-4.88×10^{-6}	4.05×10^{-6}	4.05×10^{-6}	2.96×10^{-5}	3.64
“B”, PTS-0.10p, 0.96	221,629	-9.36×10^{-6}	-3.73×10^{-6}	6.59×10^{-5}	1.62×10^{-5}	1.62×10^{-5}	1.18×10^{-4}	3.64
“B”, PTS-0.20p, 0.96	221,346	-2.86×10^{-5}	-2.33×10^{-6}	1.08×10^{-4}	6.46×10^{-5}	6.48×10^{-5}	4.73×10^{-4}	3.66
“B”, 2.0p-0, 0.96	198,402	-1.89×10^{-4}	-1.76×10^{-4}	1.34×10^{-5}	5.65×10^{-5}	7.30×10^{-5}	1.53×10^{-5}	1.13×10^{-1}
“B”, 2.0p-5, 0.96	196,988	-7.04×10^{-5}	-1.07×10^{-4}	1.29×10^{-5}	5.19×10^{-5}	6.98×10^{-5}	2.06×10^{-5}	1.63×10^{-1}
“B”, PTS-0.05p, 1.44	332,276	5.36×10^{-6}	6.67×10^{-7}	1.67×10^{-5}	4.07×10^{-6}	4.04×10^{-6}	2.95×10^{-5}	3.64
“B”, PTS-0.10p, 1.44	332,276	8.03×10^{-6}	6.10×10^{-6}	3.64×10^{-5}	1.62×10^{-5}	1.62×10^{-5}	1.18×10^{-4}	3.64
“B”, PTS-0.20p, 1.44	331,845	-1.36×10^{-5}	5.94×10^{-6}	3.06×10^{-5}	6.47×10^{-5}	6.48×10^{-5}	4.75×10^{-4}	3.67
“B”, 2.0p-0, 1.44	282,896	-8.21×10^{-5}	-2.46×10^{-4}	2.75×10^{-5}	8.84×10^{-5}	9.78×10^{-5}	1.90×10^{-5}	1.01×10^{-1}
“B”, 2.0p-5, 1.44	280,188	-2.04×10^{-5}	-1.19×10^{-4}	2.23×10^{-5}	7.80×10^{-5}	9.04×10^{-5}	2.54×10^{-5}	1.50×10^{-1}
“B”, PTS-0.05p, 1.91	443,105	1.47×10^{-6}	4.97×10^{-7}	-1.28×10^{-6}	4.06×10^{-6}	4.04×10^{-6}	2.95×10^{-5}	3.63
“B”, PTS-0.10p, 1.91	443,105	1.01×10^{-5}	-4.20×10^{-7}	9.63×10^{-6}	1.62×10^{-5}	1.62×10^{-5}	1.19×10^{-4}	3.65
“B”, PTS-0.20p, 1.91	442,446	-7.21×10^{-7}	7.90×10^{-6}	3.52×10^{-6}	6.50×10^{-5}	6.46×10^{-5}	4.73×10^{-4}	3.65
“B”, 2.0p-0, 1.91	358,869	-2.13×10^{-4}	-2.04×10^{-4}	3.37×10^{-5}	1.31×10^{-4}	1.30×10^{-4}	2.91×10^{-5}	1.11×10^{-1}
“B”, 2.0p-5, 1.91	352,678	-6.20×10^{-5}	-9.09×10^{-5}	3.06×10^{-5}	1.12×10^{-4}	1.16×10^{-4}	3.39×10^{-5}	1.50×10^{-1}
“B”, PTS-0.05p, 2.87	664,406	-7.57×10^{-7}	-3.27×10^{-6}	4.33×10^{-6}	4.07×10^{-6}	4.05×10^{-6}	2.95×10^{-5}	3.64
“B”, PTS-0.10p, 2.87	664,405	-7.18×10^{-6}	-7.43×10^{-7}	-3.75×10^{-6}	1.62×10^{-5}	1.62×10^{-5}	1.18×10^{-4}	3.65
“B”, PTS-0.20p, 2.87	663,337	-7.60×10^{-6}	-3.57×10^{-7}	1.81×10^{-5}	6.50×10^{-5}	6.48×10^{-5}	4.74×10^{-4}	3.65
“B”, 2.0p-0, 2.87	488,892	-3.36×10^{-4}	-2.52×10^{-4}	7.62×10^{-5}	1.95×10^{-4}	2.03×10^{-4}	4.74×10^{-5}	1.20×10^{-1}
“B”, 2.0p-5, 2.87	475,592	-1.67×10^{-4}	-7.19×10^{-5}	5.45×10^{-5}	1.69×10^{-4}	1.80×10^{-4}	5.01×10^{-5}	1.47×10^{-1}

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
“B”, PTS-0.05p, 3.82	885,830	9.06×10^{-7}	-4.51×10^{-7}	8.78×10^{-6}	4.05×10^{-6}	4.06×10^{-6}	2.96×10^{-5}	3.65
“B”, PTS-0.10p, 3.82	885,830	-1.95×10^{-6}	1.39×10^{-6}	-9.81×10^{-8}	1.62×10^{-5}	1.61×10^{-5}	1.18×10^{-4}	3.66
“B”, PTS-0.20p, 3.82	884,278	-4.90×10^{-6}	-3.26×10^{-6}	6.79×10^{-5}	6.49×10^{-5}	6.48×10^{-5}	4.73×10^{-4}	3.65
“B”, 2.0p-0, 3.82	594,324	-2.23×10^{-4}	-3.25×10^{-4}	9.37×10^{-5}	2.65×10^{-4}	2.84×10^{-4}	7.08×10^{-5}	1.28×10^{-1}
“B”, 2.0p-5, 3.82	566,454	-1.19×10^{-4}	-7.87×10^{-5}	2.98×10^{-5}	2.25×10^{-4}	2.42×10^{-4}	6.82×10^{-5}	1.47×10^{-1}
“C”, PTS-0.05p, 0.48	110,819	-2.99×10^{-6}	2.67×10^{-6}	-1.40×10^{-6}	4.22×10^{-6}	4.04×10^{-6}	2.96×10^{-5}	3.59
“C”, PTS-0.10p, 0.48	110,819	3.89×10^{-6}	7.54×10^{-6}	2.02×10^{-5}	1.68×10^{-5}	1.61×10^{-5}	1.18×10^{-4}	3.56
“C”, PTS-0.20p, 0.48	110,819	-3.71×10^{-6}	2.15×10^{-5}	4.02×10^{-5}	6.75×10^{-5}	6.49×10^{-5}	4.71×10^{-4}	3.55
“C”, 2.0p-0, 0.48	105,346	-4.62×10^{-5}	-7.74×10^{-5}	-2.85×10^{-6}	1.91×10^{-5}	3.25×10^{-5}	2.09×10^{-6}	3.50×10^{-2}
“C”, 2.0p-5, 0.48	105,217	-9.80×10^{-7}	-3.41×10^{-5}	4.35×10^{-7}	1.85×10^{-5}	3.15×10^{-5}	1.22×10^{-5}	2.17×10^{-1}
“C”, PTS-0.05p, 0.96	221,629	2.95×10^{-6}	-4.23×10^{-6}	-8.05×10^{-6}	4.20×10^{-6}	4.04×10^{-6}	2.97×10^{-5}	3.60
“C”, PTS-0.10p, 0.96	221,629	3.47×10^{-6}	-5.98×10^{-6}	-1.08×10^{-5}	1.68×10^{-5}	1.62×10^{-5}	1.18×10^{-4}	3.58
“C”, PTS-0.20p, 0.96	221,628	-4.83×10^{-6}	5.44×10^{-6}	7.80×10^{-5}	6.72×10^{-5}	6.49×10^{-5}	4.73×10^{-4}	3.57
“C”, 2.0p-0, 0.96	198,905	-2.27×10^{-4}	-1.69×10^{-4}	2.05×10^{-5}	5.77×10^{-5}	7.43×10^{-5}	1.64×10^{-5}	1.18×10^{-1}
“C”, 2.0p-5, 0.96	197,258	-8.68×10^{-5}	-9.88×10^{-5}	5.40×10^{-6}	5.25×10^{-5}	7.03×10^{-5}	2.16×10^{-5}	1.68×10^{-1}
“C”, PTS-0.05p, 1.44	332,276	7.33×10^{-6}	-2.75×10^{-6}	3.48×10^{-6}	4.22×10^{-6}	4.04×10^{-6}	2.95×10^{-5}	3.57
“C”, PTS-0.10p, 1.44	332,276	4.49×10^{-6}	-2.48×10^{-6}	-6.46×10^{-7}	1.69×10^{-5}	1.62×10^{-5}	1.18×10^{-4}	3.58
“C”, PTS-0.20p, 1.44	332,271	-1.68×10^{-5}	-5.96×10^{-6}	9.44×10^{-5}	6.76×10^{-5}	6.48×10^{-5}	4.73×10^{-4}	3.57
“C”, 2.0p-0, 1.44	283,828	-1.27×10^{-4}	-2.34×10^{-4}	2.52×10^{-5}	9.10×10^{-5}	9.99×10^{-5}	2.06×10^{-5}	1.07×10^{-1}
“C”, 2.0p-5, 1.44	280,809	-2.68×10^{-5}	-1.15×10^{-4}	1.25×10^{-5}	8.04×10^{-5}	9.18×10^{-5}	2.69×10^{-5}	1.55×10^{-1}
“C”, PTS-0.05p, 1.91	443,105	-1.81×10^{-6}	2.44×10^{-6}	6.95×10^{-6}	4.22×10^{-6}	4.04×10^{-6}	2.95×10^{-5}	3.58
“C”, PTS-0.10p, 1.91	443,105	8.32×10^{-6}	-1.16×10^{-5}	-2.05×10^{-6}	1.69×10^{-5}	1.62×10^{-5}	1.18×10^{-4}	3.58
“C”, PTS-0.20p, 1.91	443,093	-8.16×10^{-6}	-9.77×10^{-6}	1.11×10^{-4}	6.76×10^{-5}	6.47×10^{-5}	4.73×10^{-4}	3.57
“C”, 2.0p-0, 1.91	360,177	-2.54×10^{-4}	-2.06×10^{-4}	2.97×10^{-5}	1.34×10^{-4}	1.32×10^{-4}	3.14×10^{-5}	1.17×10^{-1}
“C”, 2.0p-5, 1.91	353,656	-1.03×10^{-4}	-9.93×10^{-5}	2.59×10^{-5}	1.14×10^{-4}	1.17×10^{-4}	3.57×10^{-5}	1.54×10^{-1}
“C”, PTS-0.05p, 2.87	664,406	2.59×10^{-8}	3.17×10^{-7}	6.42×10^{-6}	4.23×10^{-6}	4.05×10^{-6}	2.95×10^{-5}	3.57
“C”, PTS-0.10p, 2.87	664,406	3.68×10^{-6}	5.21×10^{-6}	2.78×10^{-5}	1.69×10^{-5}	1.62×10^{-5}	1.19×10^{-4}	3.58
“C”, PTS-0.20p, 2.87	664,371	-1.04×10^{-5}	-4.96×10^{-6}	3.57×10^{-5}	6.75×10^{-5}	6.47×10^{-5}	4.74×10^{-4}	3.58

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
“C”, 2.0p-0, 2.87	492,860	-3.81×10^{-4}	-2.50×10^{-4}	8.22×10^{-5}	2.01×10^{-4}	2.09×10^{-4}	5.20×10^{-5}	1.28×10^{-1}
“C”, 2.0p-5, 2.87	478,162	-1.82×10^{-4}	-7.23×10^{-5}	5.83×10^{-5}	1.73×10^{-4}	1.84×10^{-4}	5.37×10^{-5}	1.54×10^{-1}
“C”, PTS-0.05p, 3.82	885,830	2.57×10^{-6}	1.03×10^{-6}	5.07×10^{-6}	4.22×10^{-6}	4.04×10^{-6}	2.96×10^{-5}	3.58
“C”, PTS-0.10p, 3.82	885,830	-2.72×10^{-6}	4.83×10^{-6}	2.88×10^{-6}	1.69×10^{-5}	1.62×10^{-5}	1.18×10^{-4}	3.57
“C”, PTS-0.20p, 3.82	885,777	7.41×10^{-6}	-5.02×10^{-6}	5.57×10^{-5}	6.75×10^{-5}	6.46×10^{-5}	4.72×10^{-4}	3.58
“C”, 2.0p-0, 3.82	602,394	-2.15×10^{-4}	-2.88×10^{-4}	1.04×10^{-4}	2.73×10^{-4}	2.94×10^{-4}	7.78×10^{-5}	1.36×10^{-1}
“C”, 2.0p-5, 3.82	571,784	-1.15×10^{-4}	-7.27×10^{-5}	3.60×10^{-5}	2.32×10^{-4}	2.48×10^{-4}	7.41×10^{-5}	1.55×10^{-1}
“D”, PTS-0.05p, 0.48	112,500	-1.29×10^{-5}	4.45×10^{-6}	-2.30×10^{-5}	4.13×10^{-6}	4.10×10^{-6}	1.17×10^{-4}	1.43×10^{1}
“D”, PTS-0.10p, 0.48	112,500	1.10×10^{-6}	-4.24×10^{-6}	1.15×10^{-4}	1.65×10^{-5}	1.65×10^{-5}	4.68×10^{-4}	1.42×10^{1}
“D”, PTS-0.20p, 0.48	112,362	-1.99×10^{-5}	-1.79×10^{-5}	3.06×10^{-4}	6.65×10^{-5}	6.56×10^{-5}	1.87×10^{-3}	1.41×10^{1}
“D”, 2.0p-0, 0.48	106,998	-3.60×10^{-5}	-7.30×10^{-5}	3.31×10^{-6}	1.73×10^{-5}	3.21×10^{-5}	5.86×10^{-6}	1.05×10^{-1}
“D”, 2.0p-5, 0.48	106,879	3.92×10^{-6}	-2.86×10^{-5}	8.93×10^{-6}	1.68×10^{-5}	3.12×10^{-5}	4.40×10^{-5}	8.34×10^{-1}
“D”, PTS-0.05p, 0.96	225,000	-1.05×10^{-6}	-9.50×10^{-7}	2.23×10^{-5}	4.15×10^{-6}	4.11×10^{-6}	1.17×10^{-4}	1.42×10^{1}
“D”, PTS-0.10p, 0.96	225,000	5.28×10^{-6}	-4.45×10^{-6}	1.16×10^{-4}	1.66×10^{-5}	1.65×10^{-5}	4.66×10^{-4}	1.41×10^{1}
“D”, PTS-0.20p, 0.96	224,725	3.07×10^{-6}	1.57×10^{-6}	1.52×10^{-4}	6.64×10^{-5}	6.54×10^{-5}	1.88×10^{-3}	1.43×10^{1}
“D”, 2.0p-0, 0.96	200,535	-1.44×10^{-4}	-1.72×10^{-4}	2.51×10^{-6}	4.87×10^{-5}	6.94×10^{-5}	1.87×10^{-5}	1.50×10^{-1}
“D”, 2.0p-5, 0.96	199,314	-6.58×10^{-5}	-8.68×10^{-5}	-1.19×10^{-5}	4.33×10^{-5}	6.50×10^{-5}	5.65×10^{-5}	4.98×10^{-1}
“D”, PTS-0.05p, 1.44	337,500	2.02×10^{-6}	-2.38×10^{-6}	1.77×10^{-5}	4.16×10^{-6}	4.11×10^{-6}	1.17×10^{-4}	1.42×10^{1}
“D”, PTS-0.10p, 1.44	337,500	-1.45×10^{-6}	4.74×10^{-6}	6.02×10^{-5}	1.66×10^{-5}	1.64×10^{-5}	4.67×10^{-4}	1.42×10^{1}
“D”, PTS-0.20p, 1.44	337,053	5.47×10^{-6}	-3.02×10^{-6}	1.09×10^{-4}	6.66×10^{-5}	6.57×10^{-5}	1.88×10^{-3}	1.42×10^{1}
“D”, 2.0p-0, 1.44	285,477	-6.25×10^{-5}	-2.18×10^{-4}	1.44×10^{-5}	7.48×10^{-5}	9.10×10^{-5}	2.65×10^{-5}	1.58×10^{-1}
“D”, 2.0p-5, 1.44	283,197	-2.70×10^{-5}	-1.14×10^{-4}	1.06×10^{-6}	6.57×10^{-5}	8.44×10^{-5}	6.47×10^{-5}	4.27×10^{-1}
“D”, PTS-0.05p, 1.91	450,000	-5.53×10^{-8}	3.89×10^{-7}	1.50×10^{-5}	4.15×10^{-6}	4.11×10^{-6}	1.17×10^{-4}	1.41×10^{1}
“D”, PTS-0.10p, 1.91	450,000	-5.68×10^{-6}	-6.41×10^{-6}	4.87×10^{-5}	1.66×10^{-5}	1.64×10^{-5}	4.68×10^{-4}	1.42×10^{1}
“D”, PTS-0.20p, 1.91	449,368	2.77×10^{-6}	-7.67×10^{-6}	2.25×10^{-4}	6.65×10^{-5}	6.58×10^{-5}	1.88×10^{-3}	1.42×10^{1}
“D”, 2.0p-0, 1.91	360,573	-1.64×10^{-4}	-2.04×10^{-4}	4.43×10^{-5}	1.12×10^{-4}	1.20×10^{-4}	4.14×10^{-5}	1.78×10^{-1}
“D”, 2.0p-5, 1.91	355,298	-4.76×10^{-5}	-6.85×10^{-5}	3.79×10^{-5}	9.48×10^{-5}	1.07×10^{-4}	7.83×10^{-5}	3.91×10^{-1}

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
"D", PTS-0.05p, 2.87	675,000	1.68×10^{-6}	-8.15×10^{-7}	3.92×10^{-6}	4.15×10^{-6}	4.11×10^{-6}	1.17×10^{-4}	1.42×10^{1}
"D", PTS-0.10p, 2.87	675,000	1.37×10^{-6}	-6.67×10^{-6}	2.38×10^{-5}	1.66×10^{-5}	1.64×10^{-5}	4.69×10^{-4}	1.42×10^{1}
"D", PTS-0.20p, 2.87	673,898	-9.88×10^{-6}	1.07×10^{-5}	1.13×10^{-4}	6.64×10^{-5}	6.56×10^{-5}	1.87×10^{-3}	1.42×10^{1}
"D", 2.0p-0, 2.87	488,098	-2.89×10^{-4}	-2.34×10^{-4}	6.90×10^{-5}	1.69×10^{-4}	1.87×10^{-4}	6.37×10^{-5}	1.81×10^{-1}
"D", 2.0p-5, 2.87	475,764	-1.27×10^{-4}	-6.88×10^{-5}	3.90×10^{-5}	1.43×10^{-4}	1.62×10^{-4}	9.99×10^{-5}	3.35×10^{-1}
"D", PTS-0.05p, 3.82	900,000	7.19×10^{-7}	-5.71×10^{-7}	1.80×10^{-5}	4.15×10^{-6}	4.11×10^{-6}	1.17×10^{-4}	1.42×10^{1}
"D", PTS-0.10p, 3.82	900,000	1.45×10^{-6}	-2.38×10^{-6}	5.62×10^{-5}	1.66×10^{-5}	1.64×10^{-5}	4.69×10^{-4}	1.42×10^{1}
"D", PTS-0.20p, 3.82	898,345	-9.48×10^{-7}	-6.06×10^{-7}	1.62×10^{-4}	6.64×10^{-5}	6.57×10^{-5}	1.88×10^{-3}	1.42×10^{1}
"D", 2.0p-0, 3.82	587,030	-1.83×10^{-4}	-3.35×10^{-4}	1.08×10^{-4}	2.27×10^{-4}	2.59×10^{-4}	9.23×10^{-5}	1.88×10^{-1}
"D", 2.0p-5, 3.82	562,468	-7.46×10^{-5}	-8.79×10^{-5}	5.10×10^{-5}	1.90×10^{-4}	2.20×10^{-4}	1.24×10^{-4}	3.04×10^{-1}
"E", PTS-0.05p, 0.48	112,500	-8.74×10^{-6}	9.30×10^{-6}	-2.17×10^{-5}	4.28×10^{-6}	4.13×10^{-6}	1.18×10^{-4}	1.39×10^{1}
"E", PTS-0.10p, 0.48	112,500	1.04×10^{-5}	3.72×10^{-5}	-1.50×10^{-5}	1.70×10^{-5}	1.64×10^{-5}	4.73×10^{-4}	1.41×10^{1}
"E", PTS-0.20p, 0.48	112,500	2.18×10^{-5}	7.53×10^{-6}	3.80×10^{-4}	6.80×10^{-5}	6.53×10^{-5}	1.89×10^{-3}	1.41×10^{1}
"E", 2.0p-0, 0.48	106,985	-3.62×10^{-5}	-7.23×10^{-5}	-2.03×10^{-6}	1.72×10^{-5}	3.15×10^{-5}	5.98×10^{-6}	1.08×10^{-1}
"E", 2.0p-5, 0.48	106,857	-7.66×10^{-6}	-3.15×10^{-5}	4.07×10^{-6}	1.64×10^{-5}	3.09×10^{-5}	4.42×10^{-5}	8.52×10^{-1}
"E", PTS-0.05p, 0.96	225,000	4.32×10^{-6}	-1.42×10^{-6}	1.57×10^{-5}	4.30×10^{-6}	4.10×10^{-6}	1.17×10^{-4}	1.39×10^{1}
"E", PTS-0.10p, 0.96	225,000	4.92×10^{-6}	-6.66×10^{-6}	9.95×10^{-5}	1.72×10^{-5}	1.64×10^{-5}	4.69×10^{-4}	1.40×10^{1}
"E", PTS-0.20p, 0.96	224,999	-3.49×10^{-6}	1.14×10^{-5}	2.94×10^{-4}	6.88×10^{-5}	6.54×10^{-5}	1.88×10^{-3}	1.39×10^{1}
"E", 2.0p-0, 0.96	200,908	-1.75×10^{-4}	-1.71×10^{-4}	6.59×10^{-6}	4.98×10^{-5}	6.92×10^{-5}	1.99×10^{-5}	1.60×10^{-1}
"E", 2.0p-5, 0.96	199,505	-7.99×10^{-5}	-1.00×10^{-4}	-5.81×10^{-6}	4.44×10^{-5}	6.46×10^{-5}	5.76×10^{-5}	5.10×10^{-1}
"E", PTS-0.05p, 1.44	337,500	-2.63×10^{-6}	1.08×10^{-5}	-1.83×10^{-5}	4.28×10^{-6}	4.12×10^{-6}	1.17×10^{-4}	1.40×10^{1}
"E", PTS-0.10p, 1.44	337,500	-1.01×10^{-5}	1.46×10^{-5}	6.93×10^{-5}	1.73×10^{-5}	1.65×10^{-5}	4.69×10^{-4}	1.39×10^{1}
"E", PTS-0.20p, 1.44	337,493	1.67×10^{-5}	1.50×10^{-6}	1.72×10^{-4}	6.90×10^{-5}	6.57×10^{-5}	1.87×10^{-3}	1.39×10^{1}
"E", 2.0p-0, 1.44	286,190	-9.12×10^{-5}	-2.12×10^{-4}	1.42×10^{-5}	7.70×10^{-5}	9.25×10^{-5}	2.82×10^{-5}	1.65×10^{-1}
"E", 2.0p-5, 1.44	283,524	-3.12×10^{-5}	-1.24×10^{-4}	-1.75×10^{-6}	6.62×10^{-5}	8.46×10^{-5}	6.60×10^{-5}	4.32×10^{-1}
"E", PTS-0.05p, 1.91	450,000	5.33×10^{-6}	-2.62×10^{-6}	1.19×10^{-5}	4.30×10^{-6}	4.11×10^{-6}	1.17×10^{-4}	1.40×10^{1}
"E", PTS-0.10p, 1.91	450,000	1.68×10^{-6}	-8.69×10^{-6}	1.63×10^{-5}	1.73×10^{-5}	1.64×10^{-5}	4.68×10^{-4}	1.39×10^{1}
"E", PTS-0.20p, 1.91	449,995	1.38×10^{-5}	-7.45×10^{-7}	1.71×10^{-4}	6.88×10^{-5}	6.56×10^{-5}	1.87×10^{-3}	1.39×10^{1}

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
“E”, 2.0p-0, 1.91	361,509	-1.82×10^{-4}	-2.12×10^{-4}	5.03×10^{-5}	1.14×10^{-4}	1.21×10^{-4}	4.36×10^{-5}	1.85×10^{-1}
“E”, 2.0p-5, 1.91	355,957	-4.42×10^{-5}	-7.30×10^{-5}	3.44×10^{-5}	9.60×10^{-5}	1.09×10^{-4}	8.00×10^{-5}	3.95×10^{-1}
“E”, PTS-0.05p, 2.87	675,000	5.10×10^{-7}	-1.02×10^{-6}	4.11×10^{-6}	4.32×10^{-6}	4.10×10^{-6}	1.17×10^{-4}	1.39×10^{1}
“E”, PTS-0.10p, 2.87	675,000	8.85×10^{-6}	-2.33×10^{-6}	7.40×10^{-5}	1.72×10^{-5}	1.64×10^{-5}	4.69×10^{-4}	1.39×10^{1}
“E”, PTS-0.20p, 2.87	674,977	-6.63×10^{-6}	-1.57×10^{-5}	1.76×10^{-4}	6.91×10^{-5}	6.56×10^{-5}	1.88×10^{-3}	1.39×10^{1}
“E”, 2.0p-0, 2.87	491,253	-3.10×10^{-4}	-2.16×10^{-4}	8.79×10^{-5}	1.73×10^{-4}	1.92×10^{-4}	6.80×10^{-5}	1.88×10^{-1}
“E”, 2.0p-5, 2.87	477,517	-1.42×10^{-4}	-5.17×10^{-5}	5.15×10^{-5}	1.46×10^{-4}	1.65×10^{-4}	1.02×10^{-4}	3.37×10^{-1}
“E”, PTS-0.05p, 3.82	900,000	9.88×10^{-7}	1.75×10^{-6}	3.60×10^{-5}	4.32×10^{-6}	4.10×10^{-6}	1.17×10^{-4}	1.39×10^{1}
“E”, PTS-0.10p, 3.82	900,000	-8.67×10^{-7}	-1.12×10^{-6}	3.43×10^{-5}	1.73×10^{-5}	1.64×10^{-5}	4.69×10^{-4}	1.39×10^{1}
“E”, PTS-0.20p, 3.82	899,943	-2.06×10^{-7}	3.80×10^{-6}	2.32×10^{-4}	6.90×10^{-5}	6.56×10^{-5}	1.87×10^{-3}	1.39×10^{1}
“E”, 2.0p-0, 3.82	593,441	-1.90×10^{-4}	-2.80×10^{-4}	1.37×10^{-4}	2.35×10^{-4}	2.67×10^{-4}	9.85×10^{-5}	1.95×10^{-1}
“E”, 2.0p-5, 3.82	566,306	-8.38×10^{-5}	-7.99×10^{-5}	7.21×10^{-5}	1.94×10^{-4}	2.25×10^{-4}	1.29×10^{-4}	3.08×10^{-1}
“F”, PTS-0.05p, 0.48	111,069	-3.60×10^{-7}	-2.90×10^{-6}	6.40×10^{-6}	3.06×10^{-6}	3.04×10^{-6}	2.96×10^{-5}	4.84
“F”, PTS-0.10p, 0.48	111,069	6.73×10^{-6}	2.05×10^{-5}	3.69×10^{-5}	1.22×10^{-5}	1.22×10^{-5}	1.18×10^{-4}	4.84
“F”, PTS-0.20p, 0.48	111,038	6.32×10^{-8}	-1.39×10^{-5}	8.98×10^{-5}	4.88×10^{-5}	4.86×10^{-5}	4.74×10^{-4}	4.87
“F”, 2.0p-0, 0.48	105,377	-5.99×10^{-5}	-8.11×10^{-5}	5.68×10^{-6}	1.93×10^{-5}	2.91×10^{-5}	3.36×10^{-6}	5.93×10^{-2}
“F”, 2.0p-5, 0.48	105,269	-1.39×10^{-5}	-1.99×10^{-5}	2.05×10^{-6}	1.86×10^{-5}	2.82×10^{-5}	1.35×10^{-5}	2.55×10^{-1}
“F”, PTS-0.05p, 0.96	222,079	-3.03×10^{-7}	-3.07×10^{-6}	9.82×10^{-7}	3.06×10^{-6}	3.03×10^{-6}	2.95×10^{-5}	4.82
“F”, PTS-0.10p, 0.96	222,079	-2.97×10^{-6}	-2.87×10^{-6}	1.08×10^{-5}	1.22×10^{-5}	1.21×10^{-5}	1.18×10^{-4}	4.82
“F”, PTS-0.20p, 0.96	221,994	-1.51×10^{-6}	2.14×10^{-5}	-1.21×10^{-6}	4.89×10^{-5}	4.84×10^{-5}	4.73×10^{-4}	4.86
“F”, 2.0p-0, 0.96	199,443	-3.02×10^{-4}	-1.47×10^{-4}	-1.44×10^{-6}	5.99×10^{-5}	6.81×10^{-5}	2.46×10^{-5}	1.82×10^{-1}
“F”, 2.0p-5, 0.96	196,911	-9.37×10^{-5}	-6.75×10^{-5}	3.79×10^{-6}	5.14×10^{-5}	6.18×10^{-5}	2.55×10^{-5}	2.14×10^{-1}
“F”, PTS-0.05p, 1.44	333,015	-2.91×10^{-6}	-3.13×10^{-6}	-1.47×10^{-5}	3.05×10^{-6}	3.05×10^{-6}	2.96×10^{-5}	4.86
“F”, PTS-0.10p, 1.44	333,015	-5.60×10^{-7}	-9.46×10^{-6}	1.80×10^{-5}	1.22×10^{-5}	1.21×10^{-5}	1.19×10^{-4}	4.88
“F”, PTS-0.20p, 1.44	332,927	-6.26×10^{-7}	-1.09×10^{-6}	8.02×10^{-5}	4.88×10^{-5}	4.87×10^{-5}	4.73×10^{-4}	4.83
“F”, 2.0p-0, 1.44	285,299	-1.14×10^{-4}	-2.63×10^{-4}	-1.01×10^{-5}	1.01×10^{-4}	9.60×10^{-5}	3.12×10^{-5}	1.58×10^{-1}
“F”, 2.0p-5, 1.44	280,330	-1.75×10^{-5}	-1.43×10^{-4}	-1.76×10^{-6}	8.24×10^{-5}	8.38×10^{-5}	3.12×10^{-5}	1.86×10^{-1}

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
“F”, PTS-0.05p, 1.91	444,093	2.35×10^{-6}	1.61×10^{-6}	4.04×10^{-6}	3.05×10^{-6}	3.04×10^{-6}	2.95×10^{-5}	4.86
“F”, PTS-0.10p, 1.91	444,088	7.58×10^{-6}	-3.41×10^{-6}	1.74×10^{-6}	1.22×10^{-5}	1.21×10^{-5}	1.18×10^{-4}	4.86
“F”, PTS-0.20p, 1.91	443,931	4.84×10^{-6}	-1.63×10^{-5}	1.32×10^{-4}	4.88×10^{-5}	4.87×10^{-5}	4.74×10^{-4}	4.86
“F”, 2.0p-0, 1.91	363,963	-3.21×10^{-4}	-2.65×10^{-4}	1.35×10^{-5}	1.46×10^{-4}	1.28×10^{-4}	4.92×10^{-5}	1.76×10^{-1}
“F”, 2.0p-5, 1.91	352,922	-8.93×10^{-5}	-9.85×10^{-5}	1.93×10^{-5}	1.16×10^{-4}	1.06×10^{-4}	4.19×10^{-5}	1.89×10^{-1}
“F”, PTS-0.05p, 2.87	665,946	-2.56×10^{-6}	-2.68×10^{-6}	-6.53×10^{-6}	3.04×10^{-6}	3.03×10^{-6}	2.95×10^{-5}	4.86
“F”, PTS-0.10p, 2.87	665,942	8.39×10^{-7}	2.39×10^{-6}	2.55×10^{-5}	1.22×10^{-5}	1.21×10^{-5}	1.18×10^{-4}	4.86
“F”, PTS-0.20p, 2.87	665,656	-2.50×10^{-5}	-1.11×10^{-5}	4.84×10^{-5}	4.87×10^{-5}	4.85×10^{-5}	4.72×10^{-4}	4.85
“F”, 2.0p-0, 2.87	501,241	-5.65×10^{-4}	-1.69×10^{-4}	3.82×10^{-5}	2.19×10^{-4}	2.09×10^{-4}	8.00×10^{-5}	1.88×10^{-1}
“F”, 2.0p-5, 2.87	475,941	-2.08×10^{-4}	-7.04×10^{-5}	3.36×10^{-5}	1.76×10^{-4}	1.67×10^{-4}	6.36×10^{-5}	1.89×10^{-1}
“F”, PTS-0.05p, 3.82	887,585	-8.34×10^{-7}	3.16×10^{-6}	8.30×10^{-8}	3.05×10^{-6}	3.04×10^{-6}	2.95×10^{-5}	4.83
“F”, PTS-0.10p, 3.82	887,579	1.38×10^{-6}	6.79×10^{-6}	9.74×10^{-6}	1.22×10^{-5}	1.22×10^{-5}	1.18×10^{-4}	4.85
“F”, PTS-0.20p, 3.82	887,141	8.63×10^{-6}	3.09×10^{-6}	5.68×10^{-5}	4.87×10^{-5}	4.86×10^{-5}	4.72×10^{-4}	4.85
“F”, 2.0p-0, 3.82	618,818	-3.90×10^{-4}	1.94×10^{-5}	8.68×10^{-6}	3.00×10^{-4}	3.00×10^{-4}	1.22×10^{-4}	1.99×10^{-1}
“F”, 2.0p-5, 3.82	569,052	-1.29×10^{-4}	1.64×10^{-6}	1.78×10^{-5}	2.38×10^{-4}	2.31×10^{-4}	9.10×10^{-5}	1.93×10^{-1}
“G”, PTS-0.05p, 0.48	112,500	-3.56×10^{-6}	7.85×10^{-7}	1.26×10^{-5}	3.03×10^{-6}	3.04×10^{-6}	2.91×10^{-5}	4.82
“G”, PTS-0.10p, 0.48	112,500	-4.27×10^{-6}	-7.01×10^{-6}	2.29×10^{-5}	1.21×10^{-5}	1.21×10^{-5}	1.18×10^{-4}	4.87
“G”, PTS-0.20p, 0.48	112,469	-1.66×10^{-5}	1.33×10^{-5}	7.68×10^{-5}	4.85×10^{-5}	4.84×10^{-5}	4.73×10^{-4}	4.91
“G”, 2.0p-0, 0.48	106,962	-3.79×10^{-5}	-6.49×10^{-5}	6.24×10^{-6}	1.88×10^{-5}	2.96×10^{-5}	4.87×10^{-6}	8.59×10^{-2}
“G”, 2.0p-5, 0.48	106,850	-1.57×10^{-5}	-2.27×10^{-5}	-4.03×10^{-6}	1.85×10^{-5}	2.86×10^{-5}	1.46×10^{-5}	2.76×10^{-1}
“G”, PTS-0.05p, 0.96	225,000	6.56×10^{-7}	2.41×10^{-6}	1.26×10^{-5}	3.03×10^{-6}	3.03×10^{-6}	2.92×10^{-5}	4.81
“G”, PTS-0.10p, 0.96	225,000	1.21×10^{-5}	1.11×10^{-6}	-1.71×10^{-5}	1.21×10^{-5}	1.21×10^{-5}	1.17×10^{-4}	4.82
“G”, PTS-0.20p, 0.96	224,933	-2.13×10^{-5}	-8.58×10^{-6}	2.94×10^{-5}	4.88×10^{-5}	4.86×10^{-5}	4.70×10^{-4}	4.83
“G”, 2.0p-0, 0.96	202,124	-2.62×10^{-4}	-1.81×10^{-4}	2.70×10^{-5}	5.89×10^{-5}	6.58×10^{-5}	2.61×10^{-5}	1.97×10^{-1}
“G”, 2.0p-5, 0.96	199,803	-8.31×10^{-5}	-1.22×10^{-4}	1.74×10^{-5}	5.13×10^{-5}	5.97×10^{-5}	2.70×10^{-5}	2.30×10^{-1}
“G”, PTS-0.05p, 1.44	337,500	3.41×10^{-6}	-3.81×10^{-6}	1.08×10^{-5}	3.06×10^{-6}	3.02×10^{-6}	2.94×10^{-5}	4.86
“G”, PTS-0.10p, 1.44	337,500	-5.59×10^{-6}	1.29×10^{-5}	1.06×10^{-5}	1.22×10^{-5}	1.21×10^{-5}	1.18×10^{-4}	4.86
“G”, PTS-0.20p, 1.44	337,393	9.69×10^{-6}	-5.26×10^{-6}	1.15×10^{-4}	4.86×10^{-5}	4.84×10^{-5}	4.69×10^{-4}	4.83

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
“G”, 2.0p-0, 1.44	289,974	-1.11×10^{-4}	-3.08×10^{-4}	2.66×10^{-5}	1.02×10^{-4}	9.35×10^{-5}	3.51×10^{-5}	1.77×10^{-1}
“G”, 2.0p-5, 1.44	285,158	-1.32×10^{-6}	-1.64×10^{-4}	2.30×10^{-5}	8.27×10^{-5}	8.18×10^{-5}	3.54×10^{-5}	2.13×10^{-1}
“G”, PTS-0.05p, 1.91	450,000	-2.18×10^{-6}	1.95×10^{-6}	-4.37×10^{-6}	3.04×10^{-6}	3.02×10^{-6}	2.93×10^{-5}	4.84
“G”, PTS-0.10p, 1.91	450,000	2.28×10^{-6}	1.35×10^{-6}	1.44×10^{-5}	1.21×10^{-5}	1.21×10^{-5}	1.17×10^{-4}	4.83
“G”, PTS-0.20p, 1.91	449,849	1.57×10^{-6}	-9.33×10^{-7}	6.87×10^{-5}	4.86×10^{-5}	4.84×10^{-5}	4.69×10^{-4}	4.84
“G”, 2.0p-0, 1.91	370,126	-2.52×10^{-4}	-2.96×10^{-4}	5.54×10^{-5}	1.45×10^{-4}	1.26×10^{-4}	5.59×10^{-5}	2.04×10^{-1}
“G”, 2.0p-5, 1.91	358,996	-6.64×10^{-5}	-1.37×10^{-4}	1.62×10^{-5}	1.16×10^{-4}	1.05×10^{-4}	4.70×10^{-5}	2.13×10^{-1}
“G”, PTS-0.05p, 2.87	675,000	3.21×10^{-6}	-6.91×10^{-7}	8.00×10^{-6}	3.03×10^{-6}	3.02×10^{-6}	2.93×10^{-5}	4.84
“G”, PTS-0.10p, 2.87	675,000	-3.17×10^{-7}	5.82×10^{-7}	2.63×10^{-6}	1.22×10^{-5}	1.21×10^{-5}	1.17×10^{-4}	4.84
“G”, PTS-0.20p, 2.87	674,717	3.37×10^{-6}	2.39×10^{-7}	6.04×10^{-5}	4.87×10^{-5}	4.83×10^{-5}	4.70×10^{-4}	4.85
“G”, 2.0p-0, 2.87	509,593	-4.73×10^{-4}	-2.09×10^{-4}	8.43×10^{-5}	2.16×10^{-4}	2.03×10^{-4}	8.69×10^{-5}	2.09×10^{-1}
“G”, 2.0p-5, 2.87	484,981	-1.64×10^{-4}	-4.68×10^{-5}	5.22×10^{-5}	1.76×10^{-4}	1.63×10^{-4}	7.12×10^{-5}	2.14×10^{-1}
“G”, PTS-0.05p, 3.82	900,000	1.23×10^{-7}	-6.68×10^{-7}	1.82×10^{-5}	3.04×10^{-6}	3.03×10^{-6}	2.94×10^{-5}	4.85
“G”, PTS-0.10p, 3.82	899,999	-2.65×10^{-6}	-2.53×10^{-6}	1.95×10^{-5}	1.22×10^{-5}	1.21×10^{-5}	1.17×10^{-4}	4.82
“G”, PTS-0.20p, 3.82	899,544	3.57×10^{-6}	-6.87×10^{-7}	2.67×10^{-5}	4.85×10^{-5}	4.85×10^{-5}	4.71×10^{-4}	4.84
“G”, 2.0p-0, 3.82	630,595	-2.92×10^{-4}	-3.40×10^{-5}	7.48×10^{-5}	2.96×10^{-4}	2.95×10^{-4}	1.32×10^{-4}	2.20×10^{-1}
“G”, 2.0p-5, 3.82	579,550	-1.01×10^{-4}	-6.48×10^{-6}	4.17×10^{-5}	2.34×10^{-4}	2.26×10^{-4}	9.97×10^{-5}	2.16×10^{-1}
“H”, PTS-0.05p, 0.48	112,500	-1.12×10^{-6}	2.50×10^{-6}	3.91×10^{-6}	2.05×10^{-6}	2.02×10^{-6}	3.84×10^{-5}	9.42
“H”, PTS-0.10p, 0.48	112,500	3.45×10^{-6}	8.33×10^{-6}	7.44×10^{-6}	8.22×10^{-6}	8.13×10^{-6}	1.55×10^{-4}	9.42
“H”, PTS-0.20p, 0.48	112,490	-2.57×10^{-5}	1.99×10^{-6}	5.22×10^{-5}	3.29×10^{-5}	3.26×10^{-5}	6.17×10^{-4}	9.38
“H”, 2.0p-0, 0.48	106,877	-2.55×10^{-5}	-6.19×10^{-5}	5.08×10^{-6}	1.69×10^{-5}	2.99×10^{-5}	4.85×10^{-6}	8.59×10^{-2}
“H”, 2.0p-5, 0.48	106,731	-1.22×10^{-6}	-2.65×10^{-5}	5.05×10^{-6}	1.60×10^{-5}	2.80×10^{-5}	1.76×10^{-5}	3.55×10^{-1}
“H”, PTS-0.05p, 0.96	225,000	-8.07×10^{-7}	-4.31×10^{-6}	1.49×10^{-5}	2.05×10^{-6}	2.03×10^{-6}	3.84×10^{-5}	9.42
“H”, PTS-0.10p, 0.96	225,000	8.14×10^{-6}	5.24×10^{-6}	-8.31×10^{-6}	8.22×10^{-6}	8.19×10^{-6}	1.54×10^{-4}	9.36
“H”, PTS-0.20p, 0.96	224,971	-7.03×10^{-6}	5.44×10^{-6}	1.43×10^{-4}	3.28×10^{-5}	3.25×10^{-5}	6.17×10^{-4}	9.46
“H”, 2.0p-0, 0.96	208,158	-7.22×10^{-4}	-1.57×10^{-4}	1.96×10^{-5}	6.58×10^{-5}	7.39×10^{-5}	6.14×10^{-5}	4.20×10^{-1}
“H”, 2.0p-5, 0.96	200,357	-1.46×10^{-4}	-7.98×10^{-5}	6.37×10^{-6}	4.95×10^{-5}	6.10×10^{-5}	3.64×10^{-5}	3.13×10^{-1}

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ^2_X	σ^2_Y	σ^2_Z	$\frac{\sigma^2_Z}{\sigma^2_X+\sigma^2_Y+2\sigma^2_{XY}}$
“H”, PTS-0.05p, 1.44	337,500	-1.92×10^{-6}	-2.86×10^{-6}	9.97×10^{-6}	2.05×10^{-6}	2.04×10^{-6}	3.86×10^{-5}	9.43
“H”, PTS-0.10p, 1.44	337,500	2.46×10^{-6}	-3.71×10^{-6}	2.35×10^{-5}	8.22×10^{-6}	8.13×10^{-6}	1.54×10^{-4}	9.42
“H”, PTS-0.20p, 1.44	337,416	-1.40×10^{-5}	-6.53×10^{-7}	8.74×10^{-5}	3.27×10^{-5}	3.26×10^{-5}	6.16×10^{-4}	9.45
“H”, 2.0p-0, 1.44	301,513	-2.47×10^{-4}	-3.19×10^{-4}	2.04×10^{-5}	1.42×10^{-4}	1.16×10^{-4}	7.12×10^{-5}	2.82×10^{-1}
“H”, 2.0p-5, 1.44	285,830	-3.97×10^{-5}	-1.44×10^{-4}	2.61×10^{-5}	8.30×10^{-5}	8.47×10^{-5}	4.50×10^{-5}	2.66×10^{-1}
“H”, PTS-0.05p, 1.91	450,000	-1.45×10^{-6}	1.09×10^{-7}	1.04×10^{-5}	2.04×10^{-6}	2.03×10^{-6}	3.85×10^{-5}	9.45
“H”, PTS-0.10p, 1.91	450,000	1.03×10^{-5}	-3.71×10^{-6}	5.62×10^{-6}	8.19×10^{-6}	8.12×10^{-6}	1.54×10^{-4}	9.46
“H”, PTS-0.20p, 1.91	449,856	2.44×10^{-6}	8.22×10^{-6}	9.38×10^{-5}	3.28×10^{-5}	3.26×10^{-5}	6.15×10^{-4}	9.40
“H”, 2.0p-0, 1.91	402,879	-8.40×10^{-4}	-2.73×10^{-4}	-7.54×10^{-5}	2.08×10^{-4}	1.67×10^{-4}	1.33×10^{-4}	3.40×10^{-1}
“H”, 2.0p-5, 1.91	361,839	-1.62×10^{-4}	-7.76×10^{-5}	-3.36×10^{-5}	1.18×10^{-4}	1.07×10^{-4}	6.67×10^{-5}	2.94×10^{-1}
“H”, PTS-0.05p, 2.87	675,000	-1.15×10^{-6}	1.64×10^{-6}	-1.11×10^{-6}	2.05×10^{-6}	2.04×10^{-6}	3.85×10^{-5}	9.40
“H”, PTS-0.10p, 2.87	674,997	-3.10×10^{-6}	2.10×10^{-6}	-8.64×10^{-6}	8.24×10^{-6}	8.15×10^{-6}	1.54×10^{-4}	9.41
“H”, PTS-0.20p, 2.87	674,700	9.96×10^{-6}	2.98×10^{-6}	9.20×10^{-5}	3.28×10^{-5}	3.25×10^{-5}	6.17×10^{-4}	9.42
“H”, 2.0p-0, 2.87	582,170	-1.76×10^{-3}	-4.00×10^{-5}	-3.26×10^{-5}	2.96×10^{-4}	2.93×10^{-4}	1.95×10^{-4}	3.24×10^{-1}
“H”, 2.0p-5, 2.87	492,240	-4.38×10^{-4}	9.89×10^{-6}	1.19×10^{-5}	1.77×10^{-4}	1.77×10^{-4}	9.91×10^{-5}	2.84×10^{-1}
“H”, PTS-0.05p, 3.82	900,000	-1.42×10^{-6}	1.79×10^{-6}	1.85×10^{-5}	2.05×10^{-6}	2.03×10^{-6}	3.86×10^{-5}	9.42
“H”, PTS-0.10p, 3.82	899,993	-2.98×10^{-6}	-1.43×10^{-6}	1.83×10^{-5}	8.22×10^{-6}	8.14×10^{-6}	1.54×10^{-4}	9.41
“H”, PTS-0.20p, 3.82	899,443	9.15×10^{-6}	-9.08×10^{-6}	7.34×10^{-5}	3.28×10^{-5}	3.26×10^{-5}	6.16×10^{-4}	9.41
“H”, 2.0p-0, 3.82	779,361	-4.69×10^{-4}	7.64×10^{-4}	1.17×10^{-4}	4.09×10^{-4}	4.41×10^{-4}	2.75×10^{-4}	3.15×10^{-1}
“H”, 2.0p-5, 3.82	596,476	-1.41×10^{-4}	2.50×10^{-4}	4.88×10^{-5}	2.45×10^{-4}	2.57×10^{-4}	1.38×10^{-4}	2.74×10^{-1}
“I”, PTS-0.05p, 0.48	111,069	2.79×10^{-7}	4.52×10^{-6}	1.26×10^{-5}	1.53×10^{-6}	1.52×10^{-6}	1.47×10^{-5}	4.81
“I”, PTS-0.10p, 0.48	111,068	1.72×10^{-5}	-7.36×10^{-6}	8.49×10^{-5}	6.10×10^{-6}	6.10×10^{-6}	5.89×10^{-5}	4.84
“I”, PTS-0.20p, 0.48	110,795	-1.85×10^{-5}	3.28×10^{-6}	2.31×10^{-4}	2.43×10^{-5}	2.43×10^{-5}	2.38×10^{-4}	4.87
“I”, 2.0p-0, 0.48	105,219	-4.27×10^{-5}	-7.19×10^{-5}	6.16×10^{-6}	1.77×10^{-5}	2.70×10^{-5}	2.71×10^{-6}	5.02×10^{-2}
“I”, 2.0p-5, 0.48	105,047	-1.19×10^{-5}	-2.33×10^{-5}	2.91×10^{-6}	1.60×10^{-5}	2.49×10^{-5}	7.45×10^{-6}	1.60×10^{-1}
“I”, PTS-0.05p, 0.96	222,079	-3.94×10^{-6}	8.40×10^{-7}	2.13×10^{-5}	1.53×10^{-6}	1.52×10^{-6}	1.48×10^{-5}	4.84
“I”, PTS-0.10p, 0.96	222,078	3.77×10^{-6}	3.43×10^{-6}	6.02×10^{-5}	6.10×10^{-6}	6.07×10^{-6}	5.91×10^{-5}	4.85
“I”, PTS-0.20p, 0.96	221,458	-3.37×10^{-6}	2.40×10^{-6}	2.34×10^{-4}	2.45×10^{-5}	2.43×10^{-5}	2.37×10^{-4}	4.87

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
"I", 2.0p-0, 0.96	218,611	-1.75×10^{-3}	-7.14×10^{-5}	1.03×10^{-4}	9.35×10^{-5}	8.31×10^{-5}	9.76×10^{-5}	5.65×10^{-1}
"I", 2.0p-5, 0.96	198,111	-2.43×10^{-4}	-4.78×10^{-5}	6.09×10^{-5}	5.44×10^{-5}	5.73×10^{-5}	2.58×10^{-5}	2.24×10^{-1}
"I", PTS-0.05p, 1.44	333,015	-2.65×10^{-7}	2.64×10^{-6}	1.08×10^{-5}	1.53×10^{-6}	1.52×10^{-6}	1.47×10^{-5}	4.85
"I", PTS-0.10p, 1.44	333,013	-2.60×10^{-8}	2.21×10^{-7}	4.23×10^{-5}	6.10×10^{-6}	6.07×10^{-6}	5.89×10^{-5}	4.84
"I", PTS-0.20p, 1.44	332,059	-1.04×10^{-5}	1.85×10^{-5}	2.45×10^{-4}	2.44×10^{-5}	2.43×10^{-5}	2.37×10^{-4}	4.88
"I", 2.0p-0, 1.44	336,165	-1.81×10^{-4}	-9.80×10^{-4}	1.32×10^{-4}	3.16×10^{-4}	1.80×10^{-4}	1.16×10^{-4}	2.69×10^{-1}
"I", 2.0p-5, 1.44	284,053	-1.22×10^{-4}	-2.12×10^{-4}	1.90×10^{-5}	1.05×10^{-4}	8.61×10^{-5}	3.26×10^{-5}	1.75×10^{-1}
"I", PTS-0.05p, 1.91	444,093	-9.15×10^{-7}	1.97×10^{-6}	1.76×10^{-5}	1.52×10^{-6}	1.52×10^{-6}	1.47×10^{-5}	4.83
"I", PTS-0.10p, 1.91	444,088	5.44×10^{-7}	4.41×10^{-6}	3.76×10^{-5}	6.08×10^{-6}	6.06×10^{-6}	5.91×10^{-5}	4.87
"I", PTS-0.20p, 1.91	442,822	6.45×10^{-6}	-2.88×10^{-6}	2.28×10^{-4}	2.43×10^{-5}	2.43×10^{-5}	2.37×10^{-4}	4.88
"I", 2.0p-0, 1.91	514,210	-1.55×10^{-3}	-1.02×10^{-3}	6.07×10^{-4}	3.93×10^{-4}	2.77×10^{-4}	2.26×10^{-4}	3.34×10^{-1}
"I", 2.0p-5, 1.91	367,773	-3.22×10^{-4}	-1.12×10^{-4}	1.39×10^{-4}	1.51×10^{-4}	1.19×10^{-4}	6.22×10^{-5}	2.28×10^{-1}
"I", PTS-0.05p, 2.87	665,945	1.67×10^{-6}	-1.06×10^{-6}	8.84×10^{-6}	1.52×10^{-6}	1.52×10^{-6}	1.48×10^{-5}	4.84
"I", PTS-0.10p, 2.87	665,932	1.65×10^{-6}	-4.08×10^{-6}	5.56×10^{-5}	6.10×10^{-6}	6.08×10^{-6}	5.91×10^{-5}	4.85
"I", PTS-0.20p, 2.87	663,812	-5.78×10^{-6}	-4.25×10^{-6}	2.28×10^{-4}	2.45×10^{-5}	2.43×10^{-5}	2.38×10^{-4}	4.87
"I", 2.0p-0, 2.87	826,154	-3.59×10^{-3}	-5.69×10^{-4}	8.71×10^{-4}	5.00×10^{-4}	4.84×10^{-4}	2.81×10^{-4}	2.72×10^{-1}
"I", 2.0p-5, 2.87	510,381	-5.84×10^{-4}	1.69×10^{-4}	1.30×10^{-4}	2.22×10^{-4}	2.06×10^{-4}	9.15×10^{-5}	2.18×10^{-1}
"I", PTS-0.05p, 3.82	887,547	-1.54×10^{-6}	1.24×10^{-6}	1.28×10^{-5}	1.52×10^{-6}	1.52×10^{-6}	1.47×10^{-5}	4.85
"I", PTS-0.10p, 3.82	887,503	-3.08×10^{-6}	-2.88×10^{-6}	5.44×10^{-5}	6.10×10^{-6}	6.07×10^{-6}	5.91×10^{-5}	4.85
"I", PTS-0.20p, 3.82	884,397	-1.59×10^{-6}	7.94×10^{-7}	2.39×10^{-4}	2.44×10^{-5}	2.43×10^{-5}	2.37×10^{-4}	4.87
"I", 2.0p-0, 3.82	1,298,476	2.51×10^{-4}	3.04×10^{-3}	7.34×10^{-4}	6.27×10^{-4}	7.74×10^{-4}	3.50×10^{-4}	2.35×10^{-1}
"I", 2.0p-5, 3.82	628,046	-1.60×10^{-4}	8.13×10^{-4}	1.44×10^{-4}	3.03×10^{-4}	3.24×10^{-4}	1.19×10^{-4}	1.86×10^{-1}
"J", PTS-0.05p, 0.48	112,500	-2.35×10^{-6}	9.09×10^{-6}	6.52×10^{-5}	6.13×10^{-6}	6.13×10^{-6}	1.18×10^{-4}	9.61
"J", PTS-0.10p, 0.48	112,500	-1.01×10^{-5}	3.83×10^{-6}	3.32×10^{-5}	2.46×10^{-5}	2.44×10^{-5}	4.70×10^{-4}	9.58
"J", PTS-0.20p, 0.48	112,500	1.78×10^{-5}	-3.75×10^{-5}	4.38×10^{-6}	9.84×10^{-5}	9.74×10^{-5}	1.88×10^{-3}	9.57
"J", 2.0p-0, 0.48	107,075	-2.93×10^{-5}	-7.48×10^{-5}	-1.41×10^{-6}	1.85×10^{-5}	3.24×10^{-5}	6.13×10^{-6}	1.07×10^{-1}
"J", 2.0p-5, 0.48	106,971	-5.72×10^{-6}	-3.23×10^{-5}	1.66×10^{-5}	1.86×10^{-5}	3.28×10^{-5}	4.41×10^{-5}	7.80×10^{-1}

Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
“J”, PTS-0.05p, 0.96	225,000	-4.17×10^{-6}	3.29×10^{-6}	2.79×10^{-5}	6.13×10^{-6}	6.12×10^{-6}	1.18×10^{-4}	9.66
“J”, PTS-0.10p, 0.96	225,000	1.49×10^{-5}	-9.54×10^{-6}	8.23×10^{-6}	2.46×10^{-5}	2.45×10^{-5}	4.70×10^{-4}	9.60
“J”, PTS-0.20p, 0.96	225,000	1.70×10^{-5}	-2.36×10^{-5}	1.96×10^{-5}	9.86×10^{-5}	9.74×10^{-5}	1.87×10^{-3}	9.54
“J”, 2.0p-0, 0.96	200,419	-1.20×10^{-4}	-1.76×10^{-4}	2.73×10^{-6}	5.05×10^{-5}	6.85×10^{-5}	1.78×10^{-5}	1.41×10^{-1}
“J”, 2.0p-5, 0.96	199,692	-5.37×10^{-5}	-1.01×10^{-4}	2.35×10^{-5}	4.73×10^{-5}	6.70×10^{-5}	5.66×10^{-5}	4.73×10^{-1}
“J”, PTS-0.05p, 1.44	337,500	3.73×10^{-7}	2.42×10^{-6}	8.30×10^{-6}	6.16×10^{-6}	6.12×10^{-6}	1.17×10^{-4}	9.57
“J”, PTS-0.10p, 1.44	337,500	-5.99×10^{-6}	1.04×10^{-5}	1.04×10^{-4}	2.46×10^{-5}	2.44×10^{-5}	4.69×10^{-4}	9.58
“J”, PTS-0.20p, 1.44	337,500	-1.36×10^{-5}	6.12×10^{-6}	1.30×10^{-4}	9.83×10^{-5}	9.75×10^{-5}	1.87×10^{-3}	9.57
“J”, 2.0p-0, 1.44	285,344	-5.44×10^{-5}	-2.08×10^{-4}	1.37×10^{-5}	7.46×10^{-5}	9.00×10^{-5}	2.59×10^{-5}	1.54×10^{-1}
“J”, 2.0p-5, 1.44	283,798	-6.96×10^{-6}	-1.26×10^{-4}	2.23×10^{-5}	6.91×10^{-5}	8.60×10^{-5}	6.50×10^{-5}	4.12×10^{-1}
“J”, PTS-0.05p, 1.91	450,000	3.11×10^{-6}	-1.46×10^{-6}	3.95×10^{-6}	6.17×10^{-6}	6.10×10^{-6}	1.17×10^{-4}	9.57
“J”, PTS-0.10p, 1.91	450,000	5.73×10^{-6}	1.26×10^{-6}	-4.88×10^{-5}	2.45×10^{-5}	2.44×10^{-5}	4.68×10^{-4}	9.56
“J”, PTS-0.20p, 1.91	450,000	1.25×10^{-5}	7.06×10^{-6}	-3.52×10^{-5}	9.87×10^{-5}	9.73×10^{-5}	1.87×10^{-3}	9.56
“J”, 2.0p-0, 1.91	359,269	-1.03×10^{-4}	-2.41×10^{-4}	3.70×10^{-5}	1.08×10^{-4}	1.16×10^{-4}	3.87×10^{-5}	1.72×10^{-1}
“J”, 2.0p-5, 1.91	356,261	-2.61×10^{-5}	-1.01×10^{-4}	3.58×10^{-5}	9.82×10^{-5}	1.09×10^{-4}	7.86×10^{-5}	3.83×10^{-1}
“J”, PTS-0.05p, 2.87	675,000	-2.71×10^{-6}	-1.56×10^{-6}	9.40×10^{-6}	6.13×10^{-6}	6.10×10^{-6}	1.17×10^{-4}	9.56
“J”, PTS-0.10p, 2.87	675,000	3.59×10^{-6}	-7.60×10^{-6}	6.98×10^{-6}	2.46×10^{-5}	2.44×10^{-5}	4.69×10^{-4}	9.57
“J”, PTS-0.20p, 2.87	675,000	1.22×10^{-5}	-4.30×10^{-6}	4.84×10^{-5}	9.83×10^{-5}	9.79×10^{-5}	1.87×10^{-3}	9.54
“J”, 2.0p-0, 2.87	485,424	-2.01×10^{-4}	-2.65×10^{-4}	5.16×10^{-5}	1.66×10^{-4}	1.80×10^{-4}	6.04×10^{-5}	1.77×10^{-1}
“J”, 2.0p-5, 2.87	478,083	-1.05×10^{-4}	-8.70×10^{-5}	3.50×10^{-5}	1.50×10^{-4}	1.66×10^{-4}	1.01×10^{-4}	3.29×10^{-1}
“J”, PTS-0.05p, 3.82	900,000	-7.69×10^{-7}	9.08×10^{-7}	-2.40×10^{-6}	6.14×10^{-6}	6.10×10^{-6}	1.17×10^{-4}	9.59
“J”, PTS-0.10p, 3.82	900,000	-3.71×10^{-6}	-2.69×10^{-6}	2.14×10^{-5}	2.46×10^{-5}	2.45×10^{-5}	4.70×10^{-4}	9.55
“J”, PTS-0.20p, 3.82	900,000	-8.90×10^{-6}	-1.12×10^{-5}	6.72×10^{-5}	9.86×10^{-5}	9.74×10^{-5}	1.87×10^{-3}	9.55
“J”, 2.0p-0, 3.82	580,320	-1.63×10^{-4}	-3.90×10^{-4}	6.80×10^{-5}	2.20×10^{-4}	2.47×10^{-4}	8.65×10^{-5}	1.84×10^{-1}
“J”, 2.0p-5, 3.82	565,614	-7.16×10^{-5}	-1.34×10^{-4}	4.88×10^{-5}	1.97×10^{-4}	2.22×10^{-4}	1.26×10^{-4}	3.01×10^{-1}
“K”, PTS-0.05p, 0.48	111,069	2.63×10^{-7}	4.53×10^{-6}	7.41×10^{-7}	1.53×10^{-6}	1.52×10^{-6}	1.64×10^{-5}	5.38
“K”, PTS-0.10p, 0.48	111,069	1.75×10^{-5}	-7.36×10^{-6}	4.12×10^{-5}	6.11×10^{-6}	6.11×10^{-6}	6.59×10^{-5}	5.41
“K”, PTS-0.20p, 0.48	111,069	-2.04×10^{-5}	2.89×10^{-6}	5.46×10^{-5}	2.43×10^{-5}	2.43×10^{-5}	2.63×10^{-4}	5.38

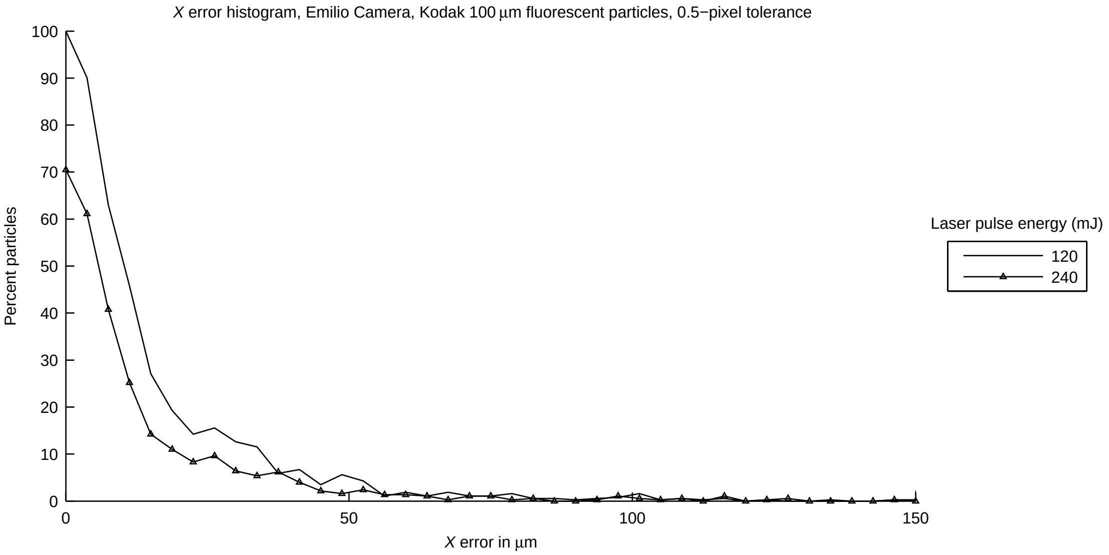
Error statistics for the reconstruction simulations (continued).

Condition	N	μ_X	μ_Y	μ_Z	σ_X^2	σ_Y^2	σ_Z^2	$\frac{\sigma_Z^2}{\sigma_X^2+\sigma_Y^2+2\sigma_{XY}^2}$
"K", 2.0p-0, 0.48	105,242	-4.44×10^{-5}	-7.47×10^{-5}	3.93×10^{-6}	1.77×10^{-5}	2.80×10^{-5}	2.69×10^{-6}	4.95×10^{-2}
"K", 2.0p-5, 0.48	105,094	-1.19×10^{-5}	-2.06×10^{-5}	-3.07×10^{-6}	1.60×10^{-5}	2.62×10^{-5}	8.02×10^{-6}	1.69×10^{-1}
"K", PTS-0.05p, 0.96	222,079	-3.84×10^{-6}	9.07×10^{-7}	1.18×10^{-5}	1.53×10^{-6}	1.52×10^{-6}	1.65×10^{-5}	5.40
"K", PTS-0.10p, 0.96	222,078	3.94×10^{-6}	3.49×10^{-6}	1.01×10^{-5}	6.11×10^{-6}	6.07×10^{-6}	6.60×10^{-5}	5.40
"K", PTS-0.20p, 0.96	222,070	-3.98×10^{-6}	2.45×10^{-6}	6.38×10^{-5}	2.45×10^{-5}	2.43×10^{-5}	2.64×10^{-4}	5.41
"K", 2.0p-0, 0.96	236,217	-2.72×10^{-3}	-4.20×10^{-5}	3.42×10^{-5}	1.09×10^{-4}	9.79×10^{-5}	1.59×10^{-4}	8.17×10^{-1}
"K", 2.0p-5, 0.96	200,700	-4.05×10^{-4}	-4.50×10^{-5}	-1.29×10^{-5}	5.86×10^{-5}	6.48×10^{-5}	3.74×10^{-5}	3.03×10^{-1}
"K", PTS-0.05p, 1.44	333,015	-2.36×10^{-7}	2.58×10^{-6}	-1.55×10^{-6}	1.53×10^{-6}	1.52×10^{-6}	1.65×10^{-5}	5.41
"K", PTS-0.10p, 1.44	333,015	-9.56×10^{-8}	3.15×10^{-7}	-4.40×10^{-6}	6.10×10^{-6}	6.08×10^{-6}	6.58×10^{-5}	5.40
"K", PTS-0.20p, 1.44	332,981	-1.02×10^{-5}	1.75×10^{-5}	5.35×10^{-5}	2.44×10^{-5}	2.43×10^{-5}	2.64×10^{-4}	5.42
"K", 2.0p-0, 1.44	364,697	-3.96×10^{-4}	-9.16×10^{-4}	9.73×10^{-5}	3.61×10^{-4}	2.06×10^{-4}	1.71×10^{-4}	3.53×10^{-1}
"K", 2.0p-5, 1.44	288,403	-2.38×10^{-4}	-1.74×10^{-4}	-6.22×10^{-5}	1.16×10^{-4}	9.75×10^{-5}	4.53×10^{-5}	2.22×10^{-1}
"K", PTS-0.05p, 1.91	444,093	-9.01×10^{-7}	1.97×10^{-6}	6.84×10^{-6}	1.52×10^{-6}	1.52×10^{-6}	1.65×10^{-5}	5.40
"K", PTS-0.10p, 1.91	444,091	5.72×10^{-7}	4.46×10^{-6}	-1.84×10^{-5}	6.09×10^{-6}	6.07×10^{-6}	6.61×10^{-5}	5.44
"K", PTS-0.20p, 1.91	443,975	6.31×10^{-6}	-2.95×10^{-6}	4.93×10^{-5}	2.43×10^{-5}	2.43×10^{-5}	2.64×10^{-4}	5.42
"K", 2.0p-0, 1.91	604,522	-2.19×10^{-3}	-1.35×10^{-3}	4.41×10^{-4}	4.38×10^{-4}	3.11×10^{-4}	3.22×10^{-4}	4.12×10^{-1}
"K", 2.0p-5, 1.91	379,917	-5.84×10^{-4}	-1.25×10^{-4}	3.80×10^{-5}	1.70×10^{-4}	1.35×10^{-4}	8.86×10^{-5}	2.84×10^{-1}
"K", PTS-0.05p, 2.87	665,946	1.68×10^{-6}	-1.13×10^{-6}	-1.25×10^{-6}	1.53×10^{-6}	1.52×10^{-6}	1.65×10^{-5}	5.41
"K", PTS-0.10p, 2.87	665,940	1.63×10^{-6}	-4.06×10^{-6}	6.78×10^{-6}	6.11×10^{-6}	6.08×10^{-6}	6.59×10^{-5}	5.41
"K", PTS-0.20p, 2.87	665,696	-6.47×10^{-6}	-3.93×10^{-6}	4.74×10^{-5}	2.45×10^{-5}	2.43×10^{-5}	2.64×10^{-4}	5.42
"K", 2.0p-0, 2.87	1,012,889	-5.50×10^{-3}	-1.22×10^{-3}	4.34×10^{-4}	5.56×10^{-4}	5.40×10^{-4}	3.81×10^{-4}	3.19×10^{-1}
"K", 2.0p-5, 2.87	534,016	-1.04×10^{-3}	1.66×10^{-4}	-2.85×10^{-5}	2.51×10^{-4}	2.40×10^{-4}	1.23×10^{-4}	2.52×10^{-1}
"K", PTS-0.05p, 3.82	887,585	-1.48×10^{-6}	1.20×10^{-6}	2.46×10^{-6}	1.53×10^{-6}	1.52×10^{-6}	1.65×10^{-5}	5.41
"K", PTS-0.10p, 3.82	887,582	-3.27×10^{-6}	-2.83×10^{-6}	4.07×10^{-6}	6.11×10^{-6}	6.08×10^{-6}	6.60×10^{-5}	5.41
"K", PTS-0.20p, 3.82	887,139	-1.90×10^{-6}	8.73×10^{-7}	5.55×10^{-5}	2.44×10^{-5}	2.43×10^{-5}	2.64×10^{-4}	5.42
"K", 2.0p-0, 3.82	1,712,271	4.16×10^{-4}	3.14×10^{-3}	2.49×10^{-5}	7.22×10^{-4}	8.32×10^{-4}	4.43×10^{-4}	2.67×10^{-1}
"K", 2.0p-5, 3.82	677,558	-2.03×10^{-4}	1.05×10^{-3}	-4.67×10^{-5}	3.43×10^{-4}	3.79×10^{-4}	1.68×10^{-4}	2.25×10^{-1}

14.8 Evidence of Performance in Real Data

The data set used in section 12.7 was processed into clouds with a 0.5-pixel matching tolerance. The clouds were then processed with the relaxation method for particle tracking, establishing a link between particles in different frames. By following the resulting tracks (and subtracting the minute velocity due to the small amount of particle drift) it is possible to estimate the standard deviation of the position of the reconstructed particle. This is the result presented in figures 14.8-1 and 14.8-2. Most striking is that the Z error distribution is not significantly wider than that of the X error, following exactly the error statistics of the simulations.

Figure 14.8-1: Histogram of the error in the X coordinate of recovered particles from the Emilio Camera using Kodak 100 µm fluorescent particles.

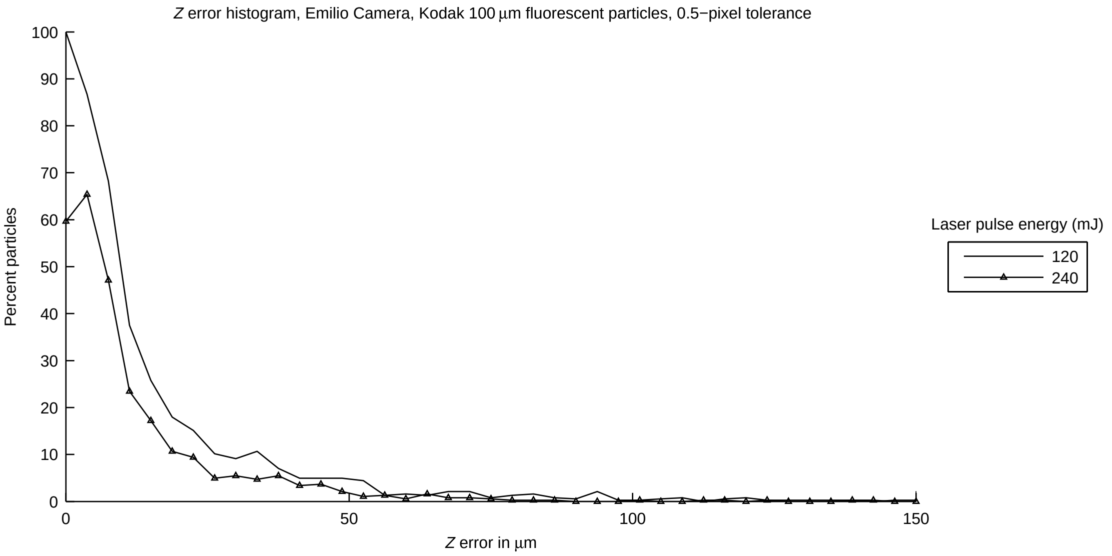


Automatic tracking of particles for the purpose of estimating the error is not very robust. If a particle is missing in a frame for whatever reason, the track is interrupted and quantifying the error over several frames becomes difficult. Still, it is important to look at the error from actual particles rather than just printed precision targets because in practice it is clear that performance using a precision target is much higher than using unevenly scattered dots or particles. Since automatic analysis is difficult and perhaps unreliable, it must be accompanied by direct observation.

The example from the Ian Camera (figure 14.8-3) is thought to be a good one based on the fact that the yield of particles (number of particles divided by average number of particle images) was high. The principal velocity component is obviously particle drift (the test was done in water and the particles are heavier than water). Evidently the jitter in Z for the track is less than 50 µm, and probably less than 20 in X.

The example from the Emilio Camera (figure 14.8-4), performed with neutrally buoyant particles

Figure 14.8-2: Histogram of the error in the Z coordinate of recovered particles from the Emilio Camera using Kodak 100 µm fluorescent particles.



(and thus most of the movement in the track is due to error rather than drift), shows slightly better performance as most of the Z vectors are less than 40 µm long. Still, considering that the Emilio camera should be substantially more precise than the Ian Camera due to the fact that it has a larger $\bar{B}_{ij}$ and pixel count, there must be error arising from the particle/illumination combination, which is also evident from the yield (which is much lower than that of the test with the Ian Camera).

Evidently the error statistics in both the simulations and these real tests conceal an immense variation in the precision of particle image identification, because an individual particle track clearly shows a difference between the X and Z precision, but the performance of an entire point cloud together does not reflect this.

Understanding the effects of seeding and illumination combinations experimentally is very difficult due to the relative lack of selection in seeding particles (they must be 5 to 10 times bigger than typical PIV particles) and the time expense of performing and analyzing reliable tests. In practice, the yield remains as the quickest estimate of the efficiency of a particle/illumination combination.

Figure 14.8-3: Measured particle streaks for two particles using the Ian Camera. Particles are ≈ 200 µm diameter transparent plastic spheres; illumination is from a strobe in forward scatter. The Z to X error ratio is evident in the jitter of the tracks.

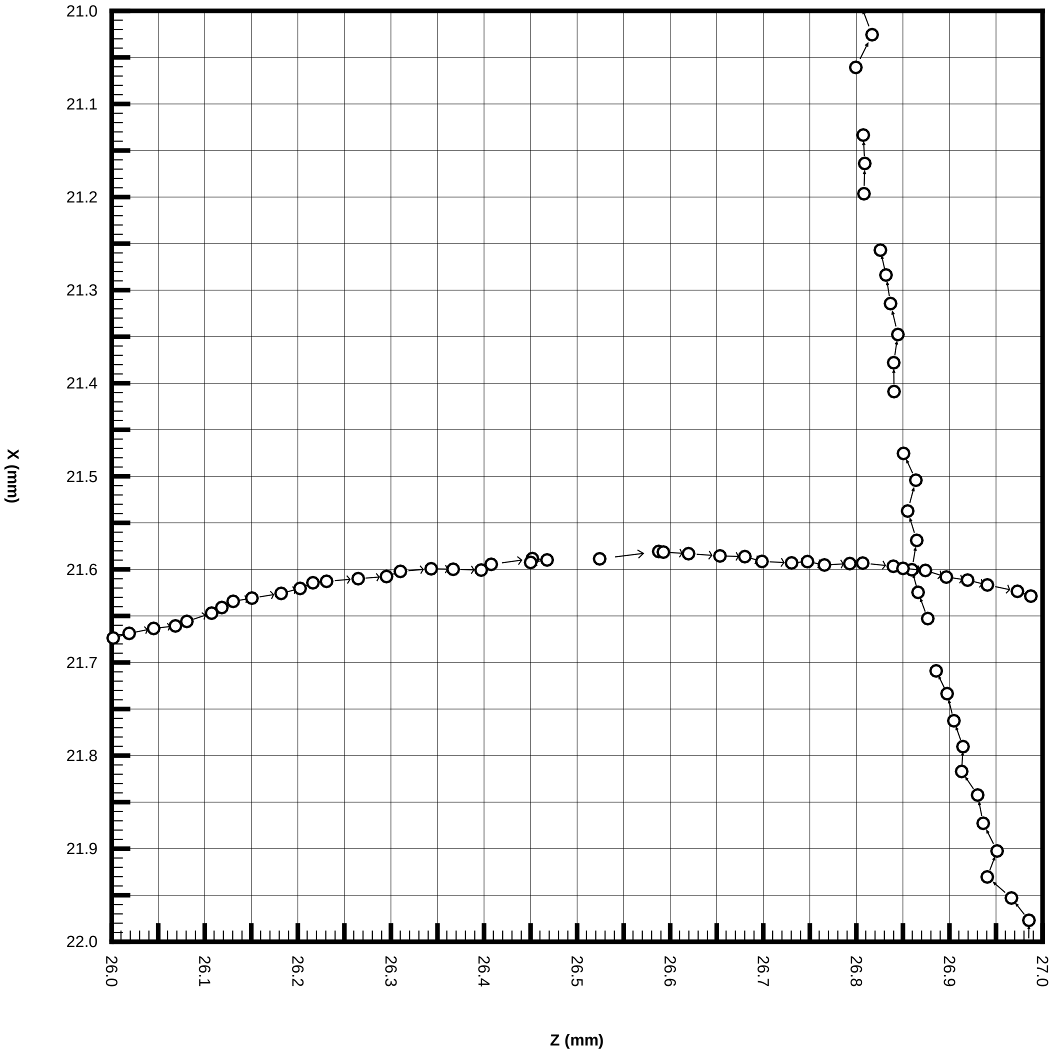
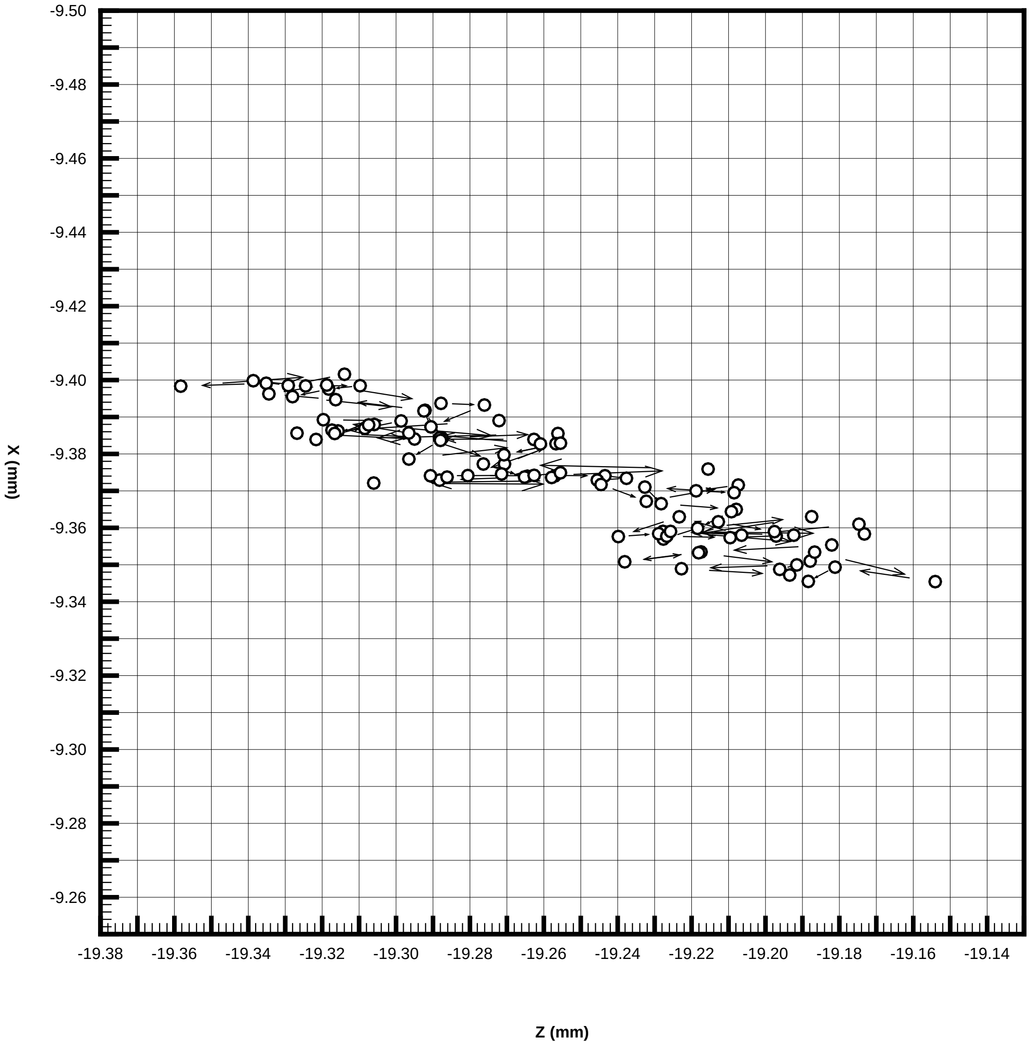


Figure 14.8-4: Measured particle streaks for a single particle using the Emilio Camera. Particles are Kodak 100 µm diameter fluorescent particles; illumination is from a single 120 mJ laser pulse. The Z to X error ratio is evident in the jitter of the track.



Chapter 15

Velocity Calculation

15.1 Introduction

After the point clouds have been reconstructed, the velocity can be calculated in a variety of ways. The preferred method is by particle tracking, in particular, a method called “relaxation”. Detail of the methods and some performance analysis was done in Pereira et al. [2006c]; here the tests are extended to include the effect of ghost particles, but they do not include tests of the neural network method, which has proven difficult to use in practice.

15.2 Details of the Simulation

Three different flows were simulated. The first is the multiple-vortex flow used in Pereira et al. [2006c]. The second and third are a simple translation (with velocity in the XZ plane) and a step flow (where the veloctiy is a simple translation in X but switches direction abruptly at the mid-plane of the volume). Each of these was simulated at two seeding densities—0.002 and 0.010 particles per cubic millimeter. The density of the multiple vortex case is 0.020 particles per cubic millimeter. The lowest density corresponds to 1,024 particles in an 80×80×80 mm volume, the middle one to 5,120 particles in the same volume, and the highest to 20,000 particles in a 100×100×100 mm volume. Real experiments typically lie in a region just higher than the middle density.

The two methods analyzed, nearest neighbor and relaxation method, were tested with no initial guess, since it has been shown in practice that the added computation time yields little, if any, gain.

In Pereira et al. [2006c], the performance tests were performed with synthetic fields only, which means there are no ghosts. Here, the particle fields were ray-traced through the Ian Camera (in the method of the `PTS` input used in chapter 14) using the typical 0.75-pixel matching tolerance. Since there was no artificial error and no Gaussian fitting, the accuracy is perfect, but the quality[1] varies with the tracking parameter Φ and the number of ghosts produced.

[1] Here quality refers to the number of correct links established.

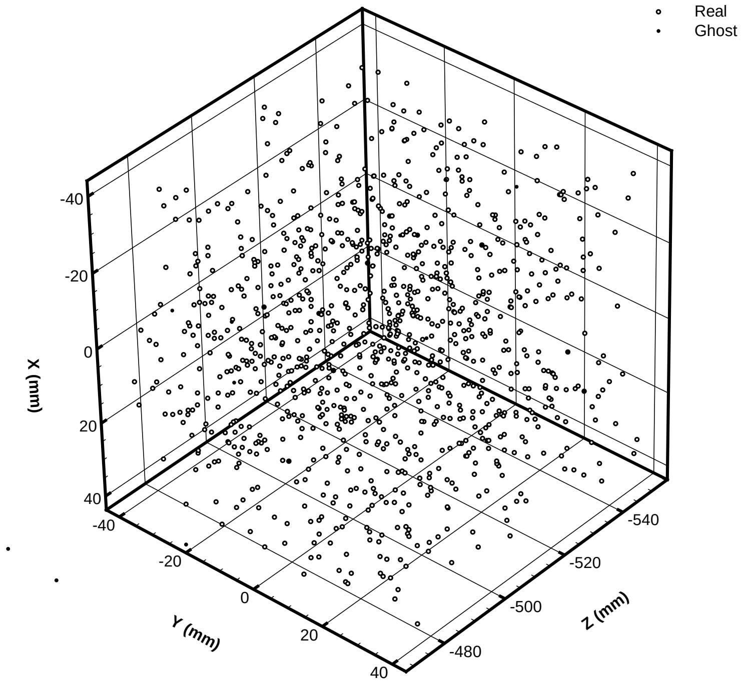
The tracking parameter Φ is defined in Pereira et al. [2006c] as

$$\Phi = \frac{d_0}{|\boldsymbol{X}|_{\max}} = \sqrt[3]{\frac{3}{4\pi\rho}} \frac{1}{|\boldsymbol{U}|_{\max}\Delta t} \tag{15.2-1}$$

where ρ is the particle density and $|\boldsymbol{U}|_{\max}$ is the expected maximum velocity. A higher value for Φ for a given condition indicates that the particles displaced less between two frames—that is, the Δt is shorter. Because these were simulations, the actual value of $|\boldsymbol{U}|_{\max}$ was used; obviously in an experiment it is usually an estimate. The value of ρ used was always that measured with synthetic particles only, that is, it does not account for ghosts. The search volume was defined to be much greater than the actual velocity to account for the fact that in real experiments this must normally be the case to be able to handle strong gradients.

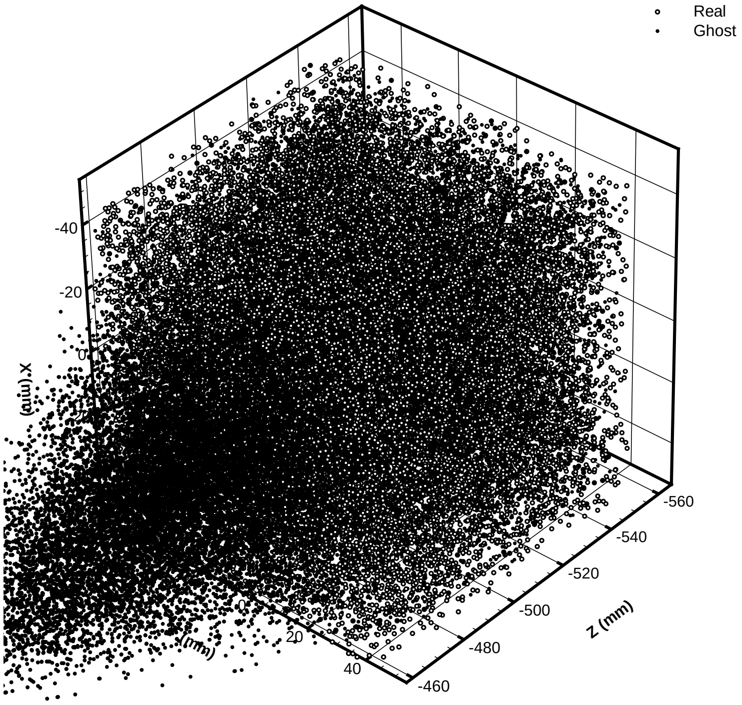
As these are now volume clouds, not sheets, clump ghosts are rare and random ghosts dominate. Still, in the case of the translation and step flows, it is possible for ghosts to "persist" between frames, especially when the displacement between frames is low (corresponding to a low Δt). In the multiple-vortex case, this is less likely, so tracks involving ghosts are most likely terribly incorrect.

Figure 15.2-1: Real and ghost particles for one frame of the 0.002 particles per mm^3 case. The cloud contains 1,024 real particles and 29 ghosts. Axes are adjusted to include only the real particles.



Tracks were categorized according to the particles they link. "Correct" tracks are, of course, between two real particles that actually represent the same particle after a displacement. "Mis-

Figure 15.2-2: Real and ghost particles for one frame of the 0.020 particles per mm^3 case. The cloud contains 19,934 real particles and 45,980 ghosts (most of which are well outside the real particle volume). Axes are adjusted to include only the real particles.



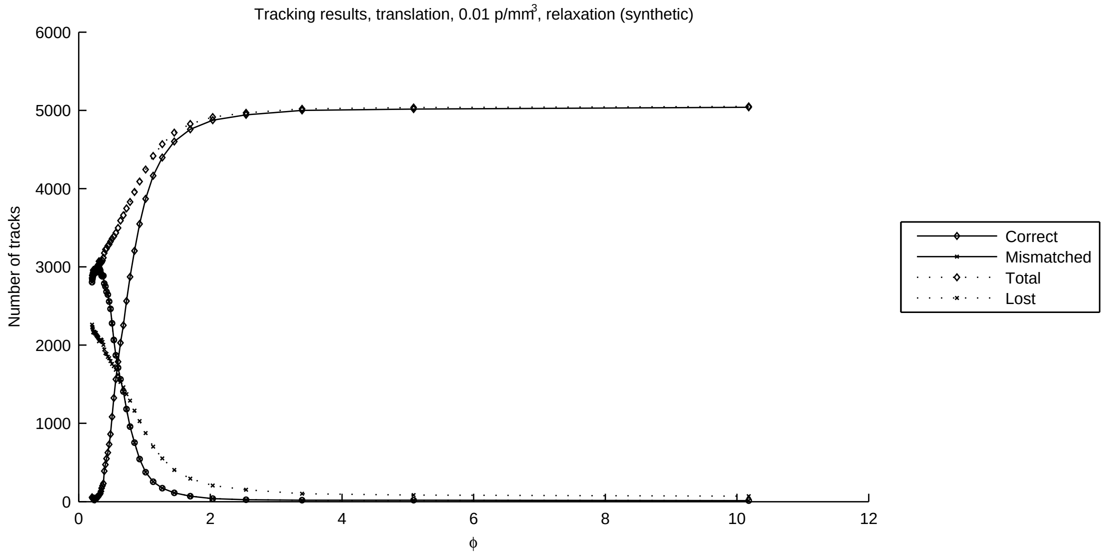
matched" are tracks between two real particles that *do not* represent the same particle after a displacement, and thus are wrong. "Lost" tracks are neither correct or mismatched, that is, they represent particles real particle pairs that were not linked at all. Once ghosts are introduced, there are three more classes: "ghost-to-real", "real-to-ghost", and "ghost-to-ghost", each denoting the type of particle in each frame that forms the link. Correct links are the only good ones; mismatched and all three ghost links introduce noise into the vector field, and missing or lost links are wasted information.

15.3 Effect of Seeding Density and Δt

As shown in Pereira et al. [2006c], as the displacement between frames is increased, the number of successful links decreases. This is true in general, and is shown in figure 15.3-1 for the case of the medium-density translation flow using the relaxation method. With synthetic data, links are either correct, incorrect, or missing. Even in this relatively simple case (with no velocity gradients) the number of missing links at the highest Φ is not zero.

In the lower density case the situation is the same. It shows more clearly though the effect that

Figure 15.3-1: Results for generated flow case of translation, 0.01 particles per mm^3 using relaxation method (from synthetic particles)



the step flow has on decreasing the overall performance: the minimum Φ at which the result is almost 100% perfect increases from nearly 0.75 in the translation case to around 2 for the step case. For any values above that, the performance does not increase appreciably. However once ghosts are factored in (figure 15.3-2), the number of missing links increases substantially, and, on top of it, a few of the real particles are incorrectly linked to ghosts. Most of the links involving ghosts are between two ghosts. (This is purely an informative statistic since all links involving ghosts will most likely be drastically different than the actual flow.)

The nearest neighbor method (figure 15.3-3) actually show better performance at the higher Φ values, presumably because every particle within the search volume receives a link regardless of anything else. The relaxation method, with its iterative probability contest, may actually remove good links due to the influence of nearby incorrect ones.

If the density increases, so does the number of ghosts and the strength of their influence. The relaxation method again produces less correct links, however, it most importantly produces less incorrect links. Figure 15.3-4 shows that at the higher values of Φ, the relaxation method produces about 33% less wrong tracks. Most important is its performance in the lower values of Φ—it produces a minimum number of wrong tracks at a value of $\Phi = 2$, at which point it produces nearly half the number of wrong tracks as does nearest neighbor. The exact position of this minimum varies but is between 2.5 (highest density) and 4 (lowest density) for the cases tested, and may not exactly coincide (but is close to) the point of peak good to wrong track ratio.

The increase in performance in this respect of relaxation versus nearest neighbor is augmented with increasing density; in the highest density case relaxation produces around 65% less wrong tracks

Figure 15.3-2: Results for generated flow case of translation, 0.002 particles per mm^3 using relaxation method

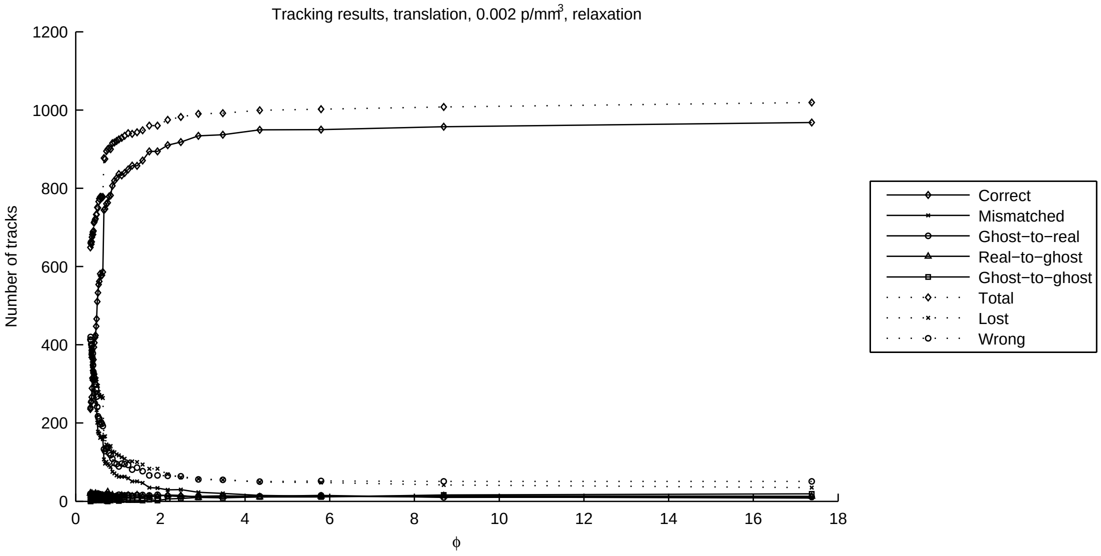


Figure 15.3-3: Results for generated flow case of translation, 0.002 particles per mm^3 using nearest neighbor method

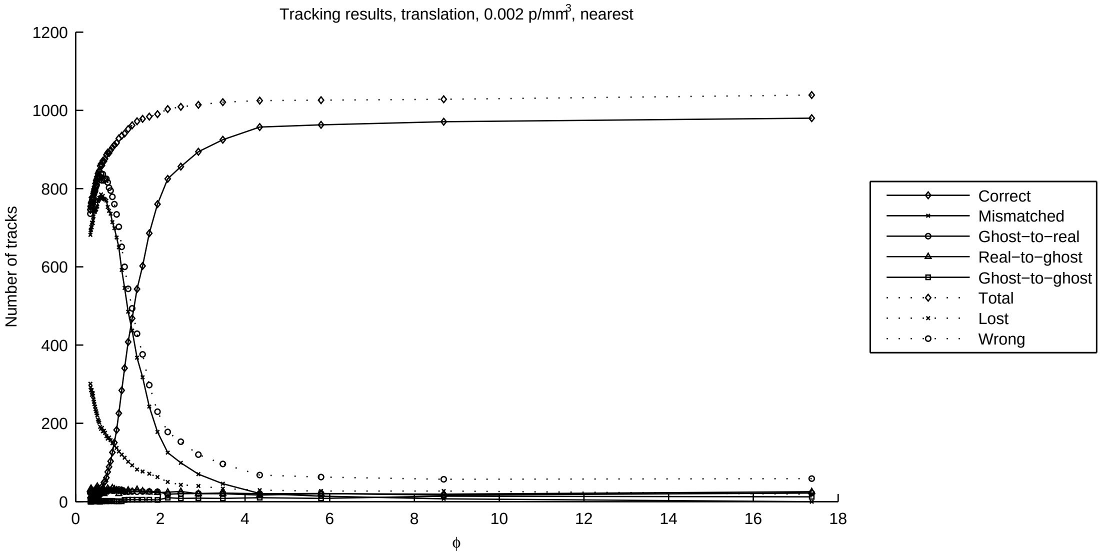
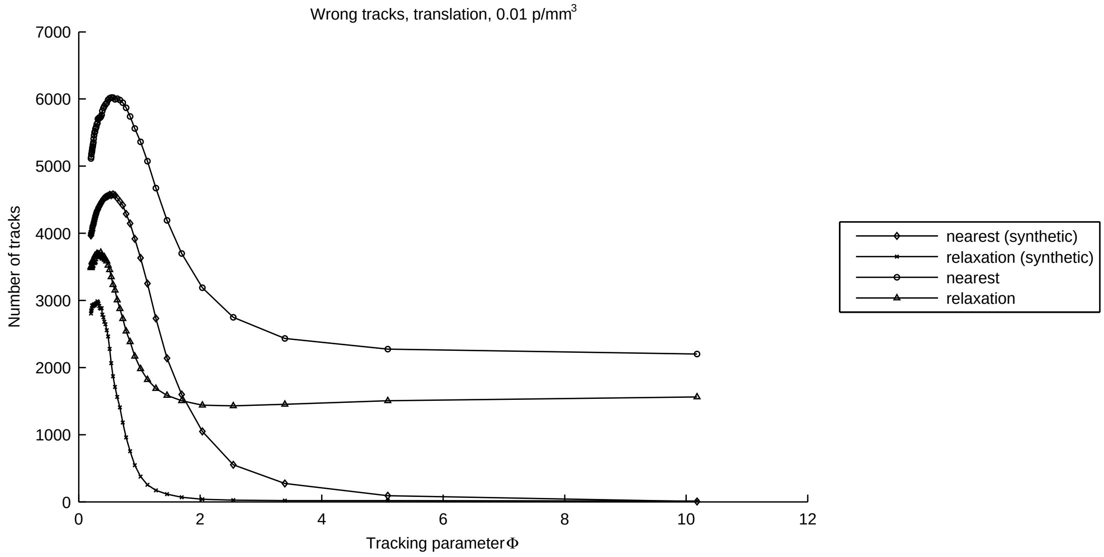


Figure 15.3-4: Wrong tracks for generated flow case of translation, 0.01 particles per mm^3



than does nearest neighbor. The most important aspect of this is not that there is a minimum at a low Φ but rather that the performance is good there at all—since after the links are formed the velocity is simply a vector subtraction of the locations of the particles involved, the error in the reconstructed position of the particle is added unaltered to the velocity measurement. At high values of Φ, where tracking is easy, the reconstruction error may well be on the order of the displacement between frames, resulting in a vector field that, even though it contains mostly correct links, the precision of the velocity is awful. The displacement between frames should always be maximized to reduce this effect, but of course the linking performance has to be accounted for. In this respect, the relaxation method is far superior to nearest neighbor.

Figures 15.3-5 and 15.3-6 show the same section of the vector field of the multiple vortex case (with ghosts) calculated with each method. Even with the nearest neighbor method, the flow is very obvious, however, any spurious vectors, or “outliers”, make it difficult to perform any automatic analysis on the flow. Due to the relatively recent implementation of the PTV algorithms into the software (late 2005) there is currently no automatic, robust way to remove outliers from PTV data; simply checking vector magnitude or direction of a vector with respect to its neighbors is useless in the cases where there are substantial velocity gradients because correct vectors at the boundaries will also fail the test. Currently the most reliable method is to remove vectors that fall outside of a user-selected variance interval during averaging, which of course does nothing for the instantaneous flow case.

Figure 15.3-5: Enlarged section of the vector field for the multiple-vortex case at $\Phi \approx 2.46$, for which the good to wrong track ratio peaks for the relaxation method at ≈ 7.20 and the mean particle displacement is $\approx 416\mu m$, using the nearest neighbor method.

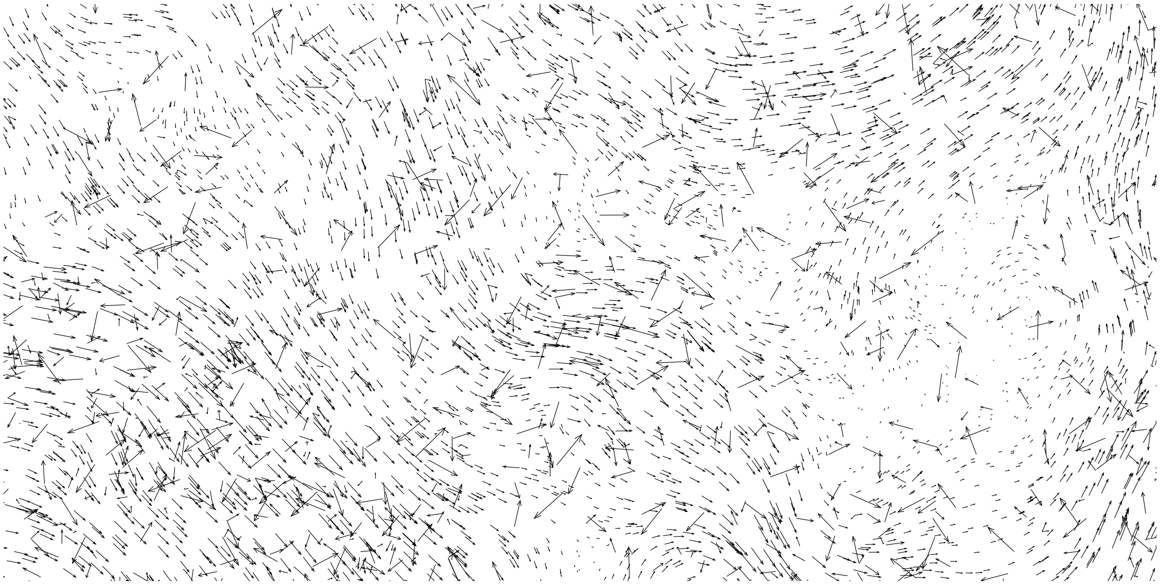
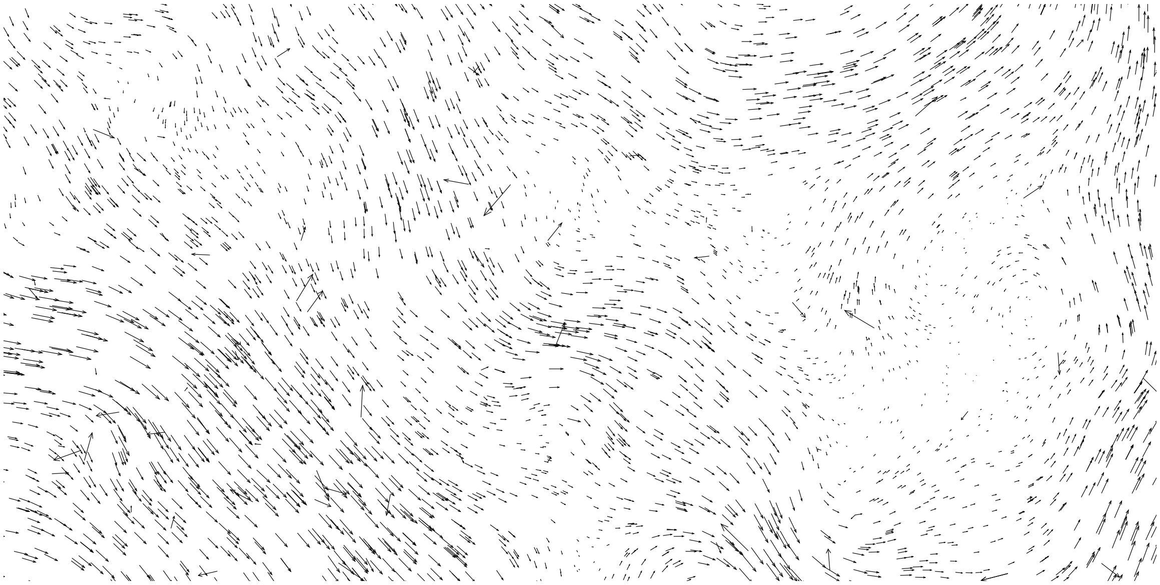


Figure 15.3-6: Enlarged section of the vector field for the multiple-vortex case at $\Phi \approx 2.46$, for which the good to wrong track ratio peaks for the relaxation method at ≈ 7.20 and the mean particle displacement is $\approx 416\mu m$, using the relaxation method.



15.4 Evidence of performance in real data

The two examples presented here span the bulk of the development of DDPIV by the author.

The first is a demonstration of a vortex ring in water using the Black Camera (see section 7.4.3) before multi-plane dewarping or particle tracking (July of 2004). As a result the accuracy is miserable; a 1.5-pixel tolerance had to be used to generate any results at all. Of the nearly 5,000 particles recovered, it is suspected that probably around 2,000 are real. After processing, to arrive at a relatively "clean" vector field required filtering by cropping the maximum velocity magnitude to 35 mm per second. The final vector field contained 1,414 vectors for a yield of 31% (which is about half of what can be expected today). Using this for the value of $|\boldsymbol{U}|_{\max}$ yields a value of $\Phi \approx 23.09$ which, according to the simulations is beyond the region where the performance of nearest neighbor and relaxation is distinguishable. This is verified in this test, which resulted in the same vector field using both methods. At this value of Φ, it is almost guaranteed that all wrong links are due to ghosts, since the chance of two real particles being close enough together to generate a mismatch is minute.

The enormous value of Φ arises because this demonstration was a hasty test and the Δt was just 6 milliseconds, which is definitely too slow for a slow-moving vortex ring about the size of a fist. With an average displacement of just 44 µm, the argument presented above about the signal-to-noise ratio of PTV measurements becomes clear. Figure 15.4-1 shows a view along Z of one resulting vector field (with no filtering other than velocity cropping). Although the data is not perfect, the ring is clearly visible. If the same data set is rotated 90°, so that the view is along Y, the data seems completely hopeless, as the minute displacement is smaller than the error in Z position.

The second example is that of a flapping flow in water, recorded with the Emilio Camera in January of 2007. With calibration performed using the super-accurate photo-etched target in-situ, a matching tolerance of 0.5 pixels resulted in an extremely accurate particle field. With such a strong cloud, the yield of the "clean" vector field was nearly 80%. Visual inspection of the vector field indicates that the maximum velocity was around 500 mm per second, which results in $\Phi \approx 2.28$ and a mean displacement of ≈ 263 µm. Assuming that the error in Z is at worst 40 µm (as in the sample particle track of figure 14.8-4, we can estimate a signal-to-noise ratio of ≈ 6.58 in Z (and better in X). This, along with the fact that the primary flow direction is along X results in a beautiful vector field. Figures 15.4-3 and 15.4-4 show the vector field which, again, contains no filtering other than velocity cropping.

Figure 15.4-1: *XY* view of a vortex ring demonstration performed with the Black Camera in 2004. With just a few outliers, it is obvious where the ring is.

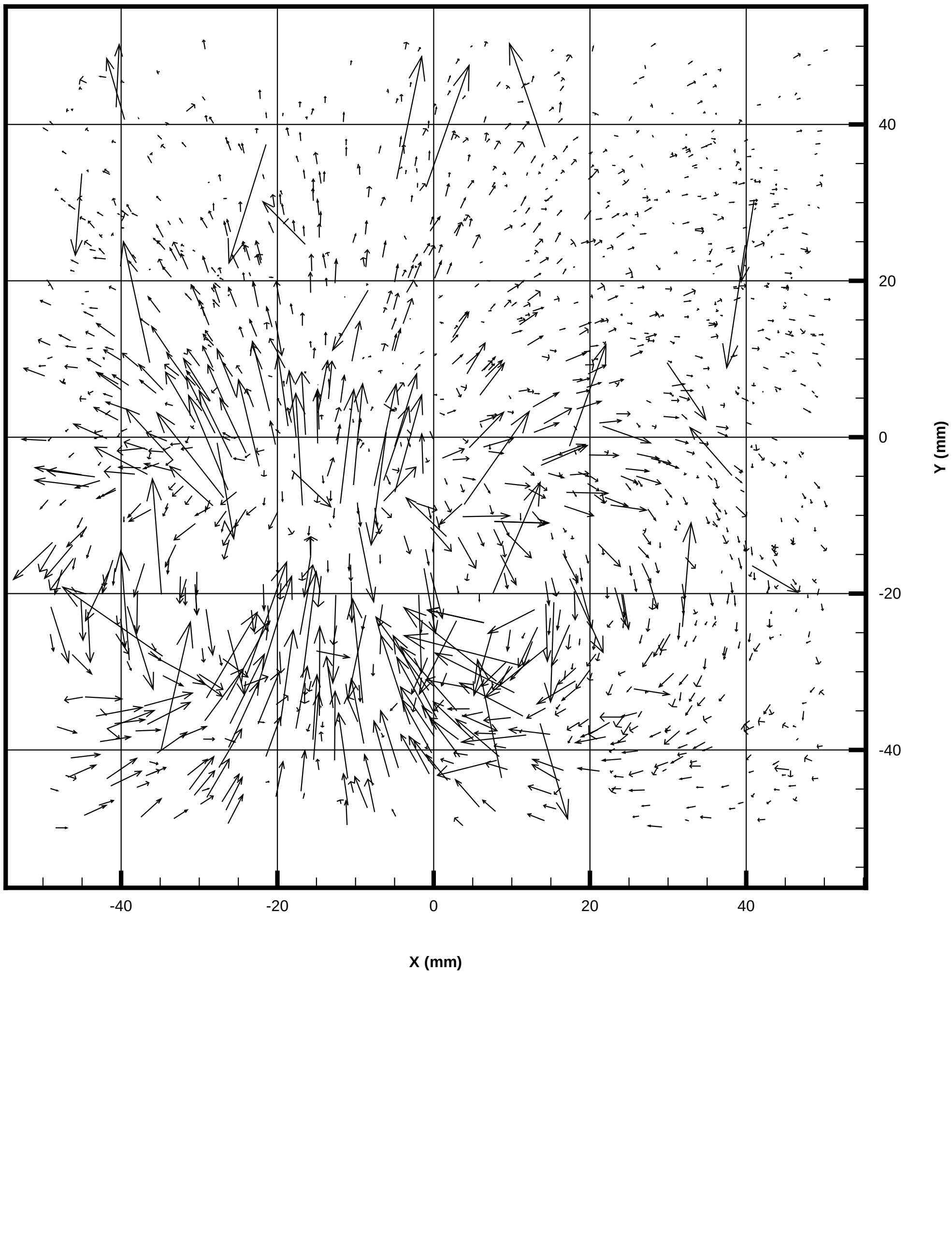


Figure 15.4-2: XZ view of a vortex ring demonstration performed with the Black Camera in 2004. The ring is completely lost due to the fact the displacements are mostly smaller than the average error.

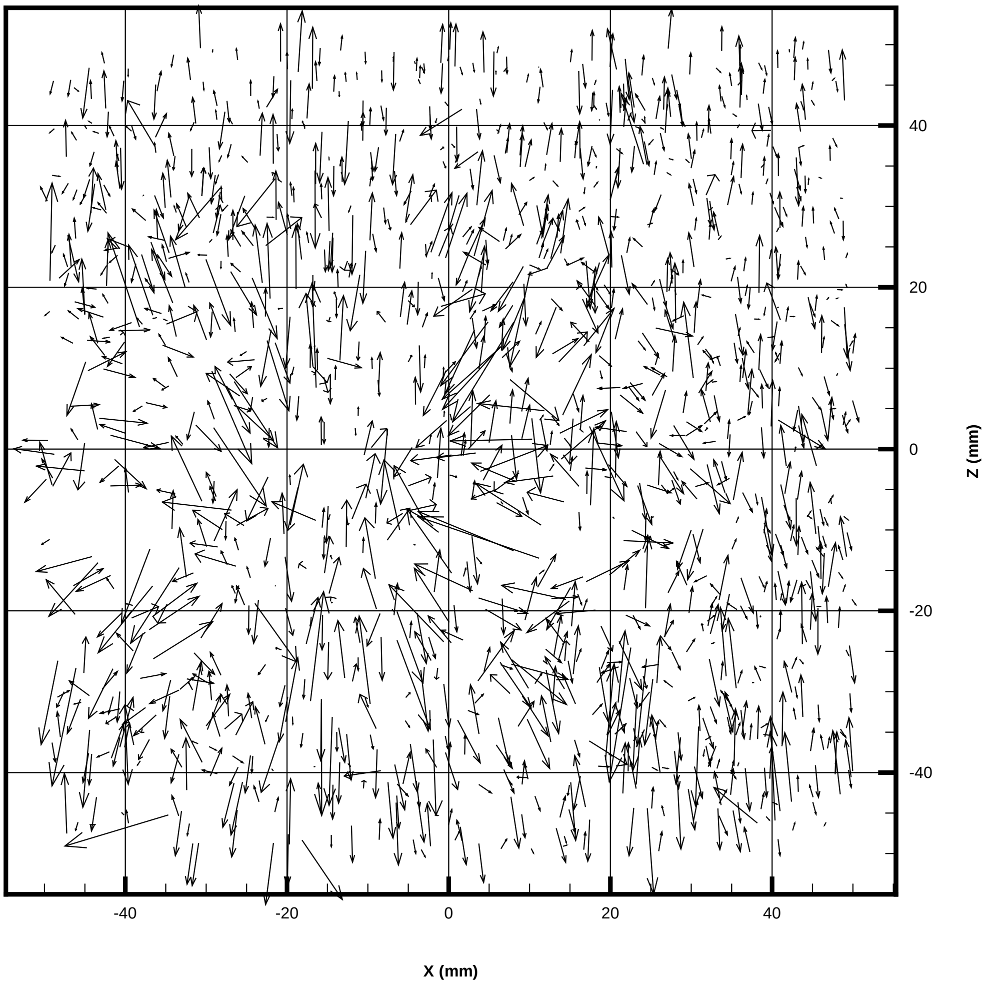


Figure 15.4-3: XY view of the flow induced by a moving flap recorded with the Emilio Camera in 2007.

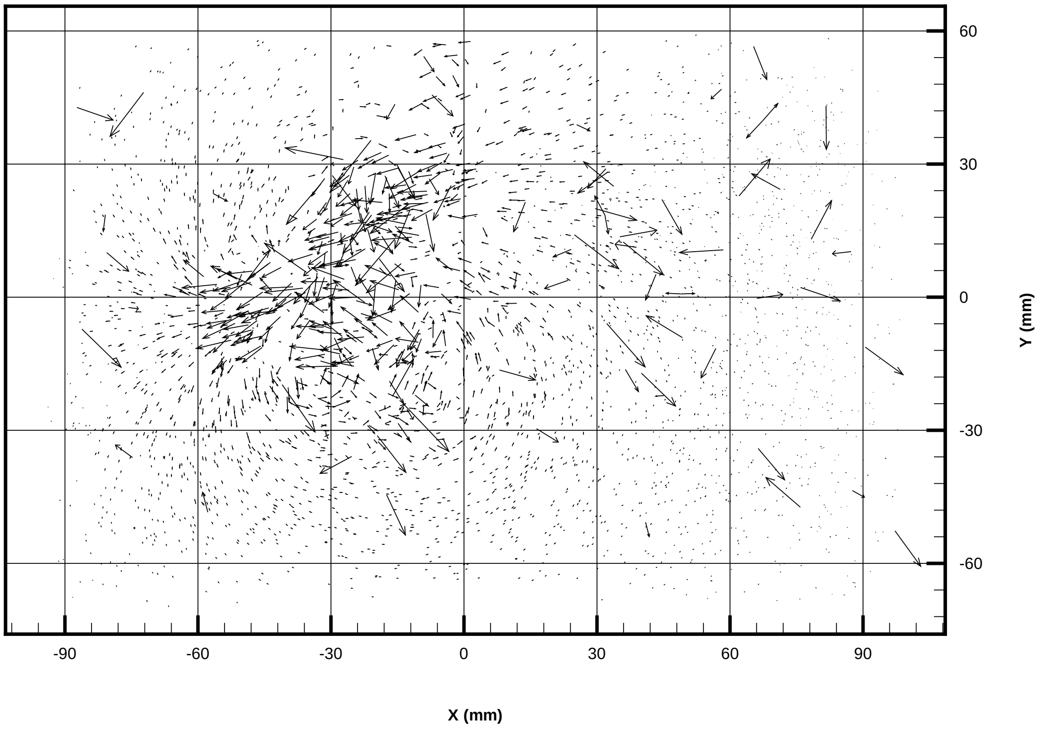
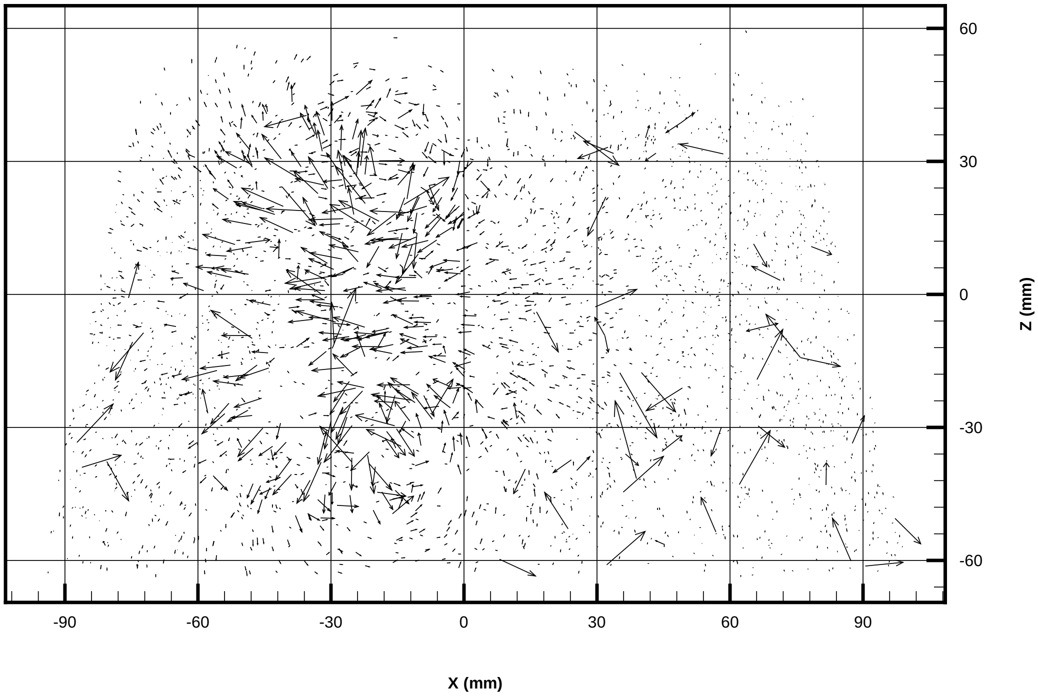


Figure 15.4-4: *XZ* view of the flow induced by a moving flap recorded with the Emilio Camera in 2007. Note the quality of even the quiescent regions, due to the high quality of the raw data and calibration.



Part IV

Experimental Demonstrations

Chapter 16

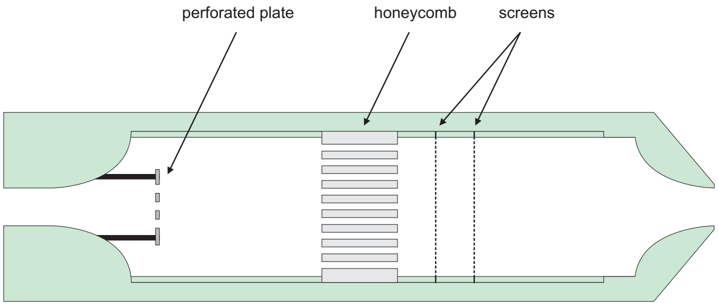
Jet Flow

16.1 Experiment Details

The jet assembly of figure 16.1-1 was placed at the center of a water tank approximately 500×350×300 mm facing down (towards $Y-$), with the jet nozzle about 30 mm below the free surface (centered at $Y \approx 55$ in the figures). A Rainbow Lifegard Quiet One 1200 aquarium pump feeds the jet from one corner of the tank. Kodak 100 µm fluorescent particles were used for seeding, and images were acquired at 7 pairs per second. The Reynolds number, based on the nozzle diameter and exit speed, is $\approx 1.2 \times 10^4$.

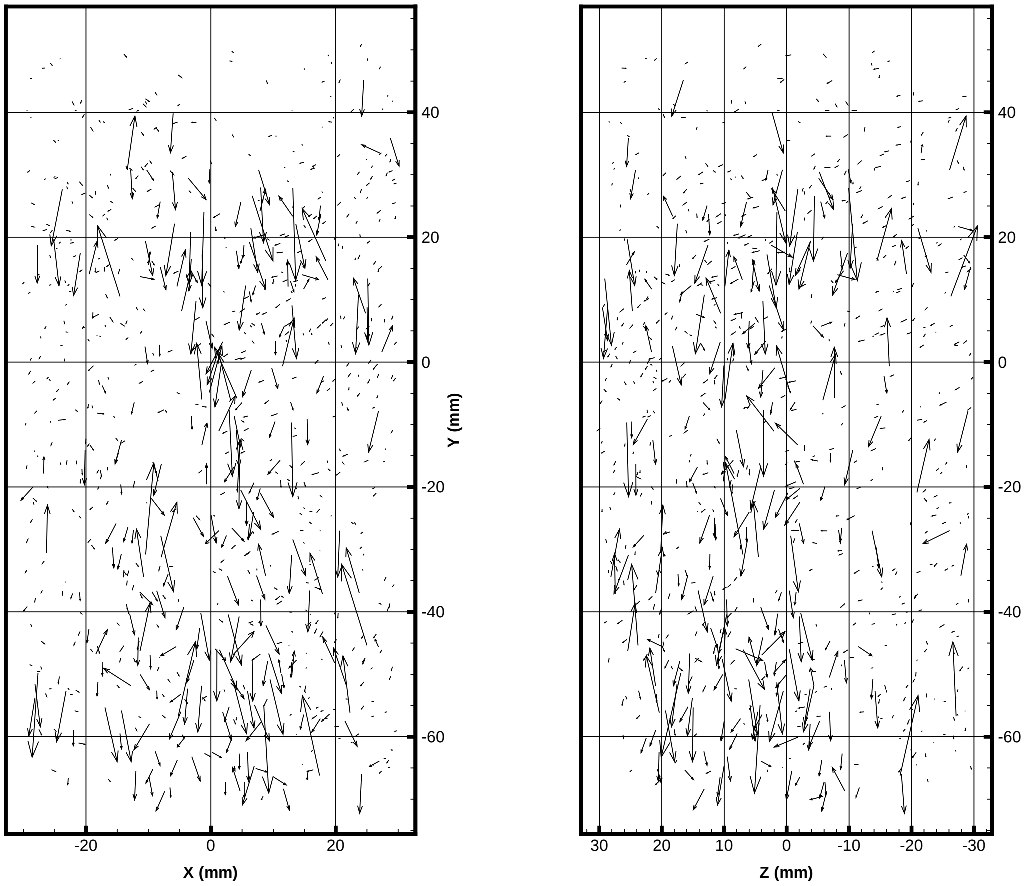
Each point cloud had an approximate volume of 60×124×60 mm ($X \times Y \times Z$) with about 1200 particles (density of 0.0027 particles per cubic mm). A total of 1,024 point cloud pairs were processed. After velocity cropping, about 800 vectors remained (yield of 67%). With a Δt of 8 milliseconds, the experiment was performed at a $\Phi \approx 0.56$ (the maximum displacement is nearly 8 mm), which, according to figure 15.3-2, puts it in an unfavorable region for the relaxation method (and a prohibitive region for the nearest neighbor method), which explains the low yield. The

Figure 16.1-1: Schematic of the jet used in the experiment, 50% actual size. Flow passes from left to right. Outlet nozzle diameter is 10 mm.



average displacement is around 1 mm, however, the average Z displacement is only 120 µm and thus the signal-to-noise ratio in Z is relatively low[1].

Figure 16.1-2: XY and YZ view of the vector field from one pair of frames in the jet experiment, showing the dismal performance of the relaxation method ($\Phi \approx 0.56$).



Vector fields produced by particle tracking are averaged by dividing the domain into a grid of overlapping volume elements, each of which has a vector assigned to it equal to the average of all the vectors which fall within that element. These elements, in `DDPIV`, are called “voxels”, though in most other cases a voxel is a volumetric pixel in a three-dimensional bitmap image. Typically the voxel overlap is set to be 50% in each direction. In the case of the jet experiment, the voxels were 4×4×4 mm and so there is one data point ever 2 mm in each direction (approximately 68,000 vectors).

When averaging a flow in `DDPIV`, the user is given an option to filter the vectors by removing those vectors in each voxel which fall beyond a user-defined multiple of the standard deviation of all

[1] There is a clear advantage in the volumetric capabilities of DDPIV. Because the domain is a volume and the orientation is irrelevant, in a proper vessel the experiment can be reoriented relative to the camera so that the signal-to-noise ratio is better balanced between the axes.

the vectors within that voxel. The algorithm is multi-pass, that is, it filters each voxel a multiple number of times so that each time a "bad" vector is removed the average is recalculated (so the distribution should narrow and approach a normal one as the iterations proceed).

In the case of the jet experiment, averaging all vector fields resulted in an average of 105 vectors per voxel without filtering and 55 vectors per voxel with filtering set to remove all vectors beyond 2 standard deviations.

16.2 Results

In comparing the filtered and unfiltered averages, the first thing to notice is that the maximum velocity magnitude of the unfiltered vector field is much lower than that of the filtered one (figures 16.2-1 and 16.2-2). One effect of having too low a value of Φ is that in such a directional flow is that mismatched particles can often result in vectors that are facing the wrong direction, thus the resulting average is much lower than it should be. The occurrence of this is directly proportional to the velocity displacement, and so in the unfiltered data the jet core is almost completely gone. Filtering the vectors during averaging greatly reduces this pitfall, as is evidenced by the fact that the standard deviation drops drastically after filtering in the high-speed section of the jet (figure 16.2-4 compared to figure 16.2-3). This, of course, only works if the initial matching was successful enough that the magnitudes between the wrong vectors and right vectors differ enough and that there are more correct vectors than incorrect ones. Figure 16.2-6 shows, agreeingly, that the high speed section is that which ends up with the least number of vectors in each average. In the unfiltered vector field (figure 16.2-5), there are still less vectors in the center than on the outer edges. This may be caused by uneven energy distribution in the laser beam, or by this region's combination of seeding density and displacement having an undesired consequence in the relaxation algorithm.

One way to ascertain the validity of the data is to calculate the mass flux in different portions of the jet. Any volume not including the nozzle should have zero net mass flux, since there are no sources or sinks anywhere. Figure 16.2-7 shows this quantity for the filtered data set. Each point on the plot shows the net mass flux through a slice (XZ plane) as a percentage of the mass in-flux at the nozzle. The slices are 29×1/2×29 voxels in dimension, so the flux of the jet itself dominates over the flux due to entrainment. The fact that there is systematic error (that is, the net flux is positive) indicates that the measurement is not good. Performing the same analysis on the unfiltered data yields an even higher bias to the error in the mass flux, thus it is safe to conclude this error is in some ways a measure of the amount of mismatching that occurred during tracking, since, as mentioned before, in this low-Φ, directional flow, errors in matching frequently result in vectors facing the opposite direction of the flow (and thus decreasing the average magnitude). It seems then that filtering by standard deviation improves the quality of the flow qualitatively but

Figure 16.2-1: XY slice through the center of the jet. The colors show the magnitude of the vectors in each voxel (in mm per second).

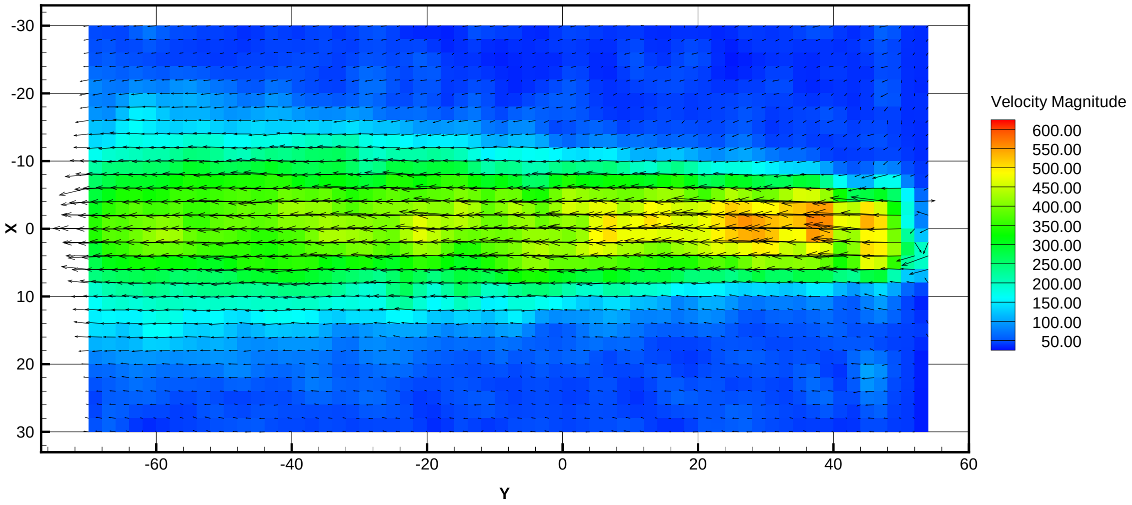


Figure 16.2-2: XY slice through the center of the jet, filtered by standard deviation during averaging. The colors show the magnitude of the vectors in each voxel (in mm per second).

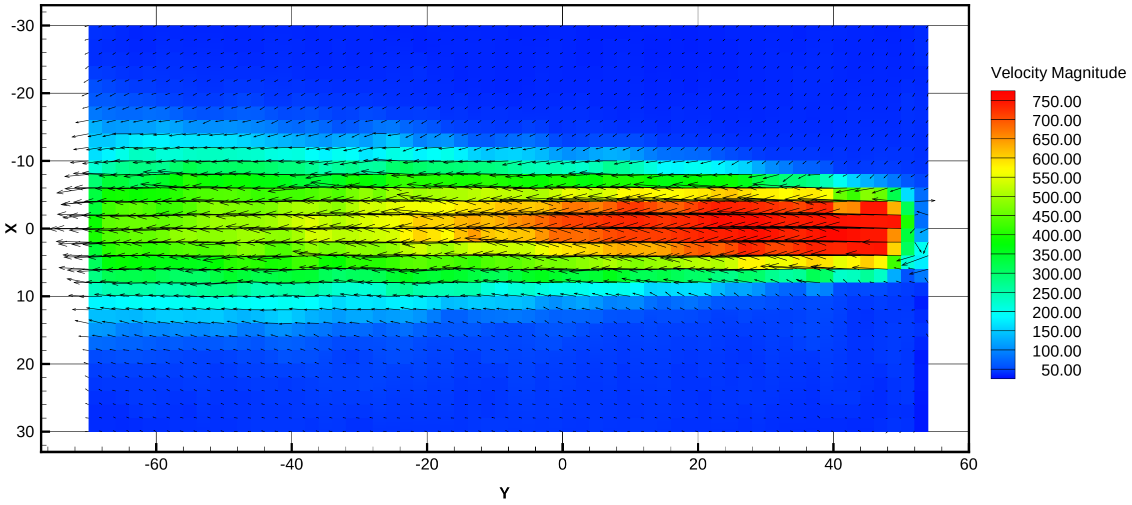


Figure 16.2-3: *XY* slice through the center of the jet. The colors show the standard deviation of the velocity (in mm per second).

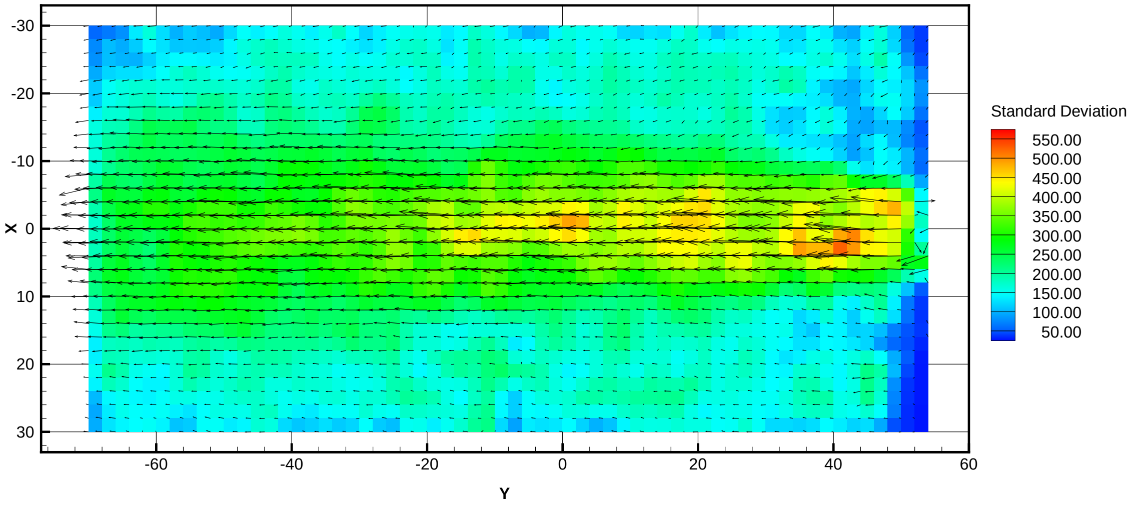


Figure 16.2-4: *XY* slice through the center of the jet, filtered by standard deviation during averaging. The colors show the standard deviation of the velocity (in mm per second).

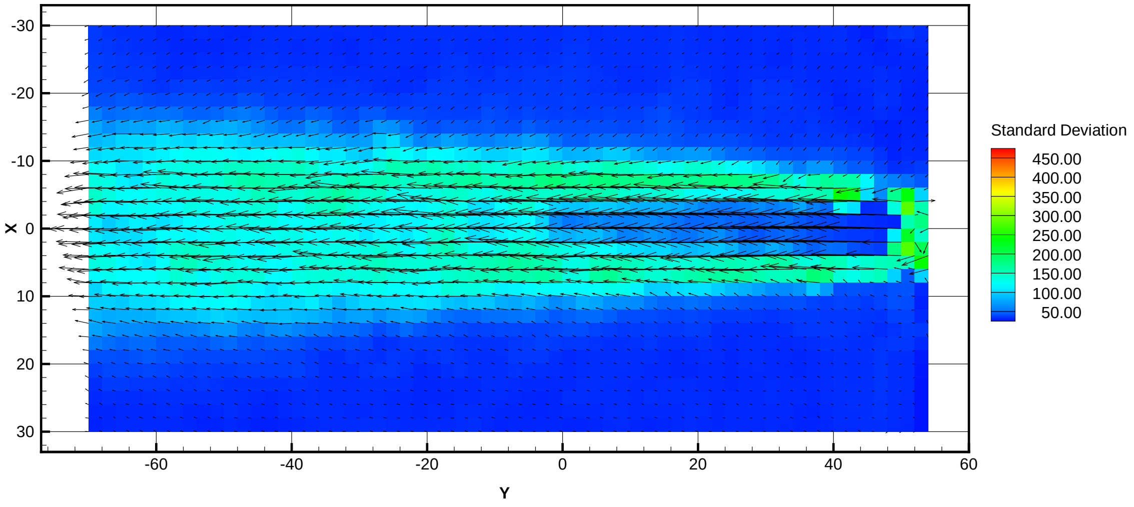


Figure 16.2-5: XY slice through the center of the jet. The colors show the number of vectors in each voxel.

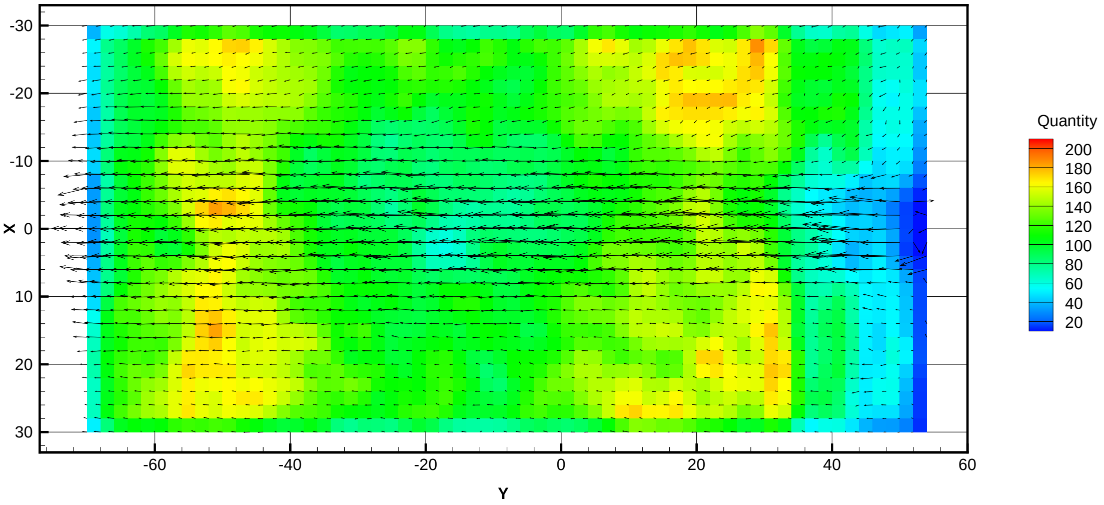
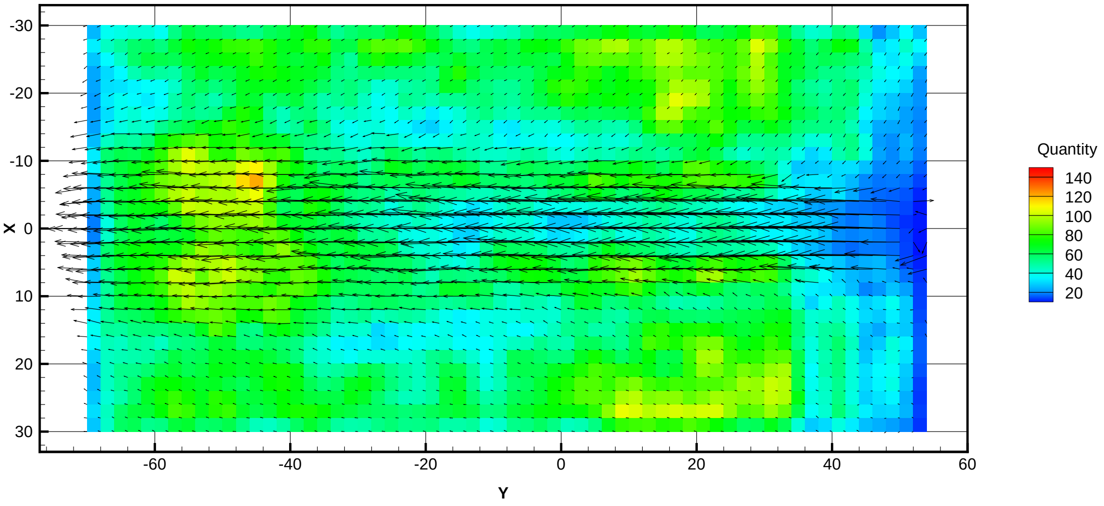


Figure 16.2-6: XY slice through the center of the jet, filtered by standard deviation during averaging. The colors show the number of vectors in each voxel.



does not rid it of any quantitative problems.

Figure 16.2-7: Net mass flux as a percent of influx at the nozzle for XZ slices at different distances from the nozzle.

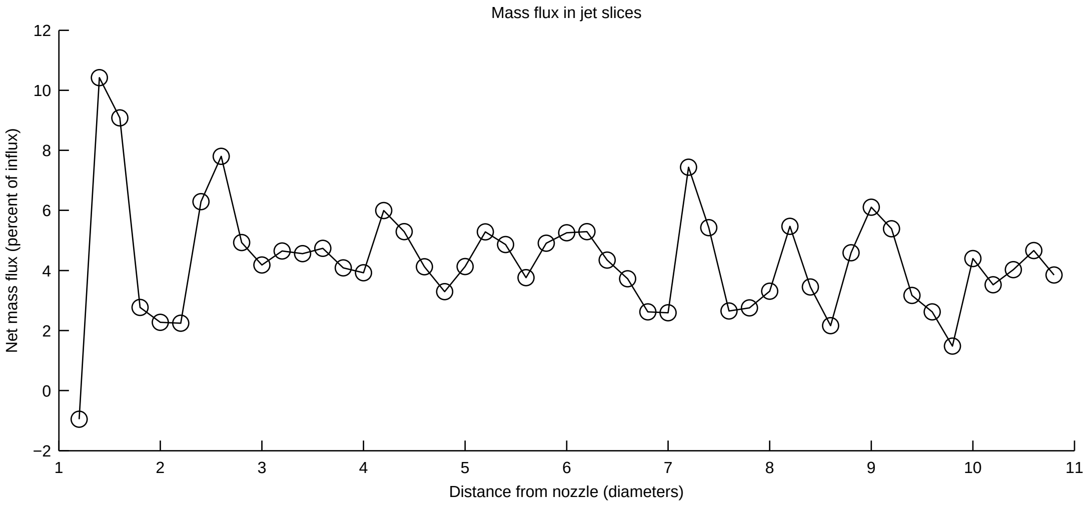
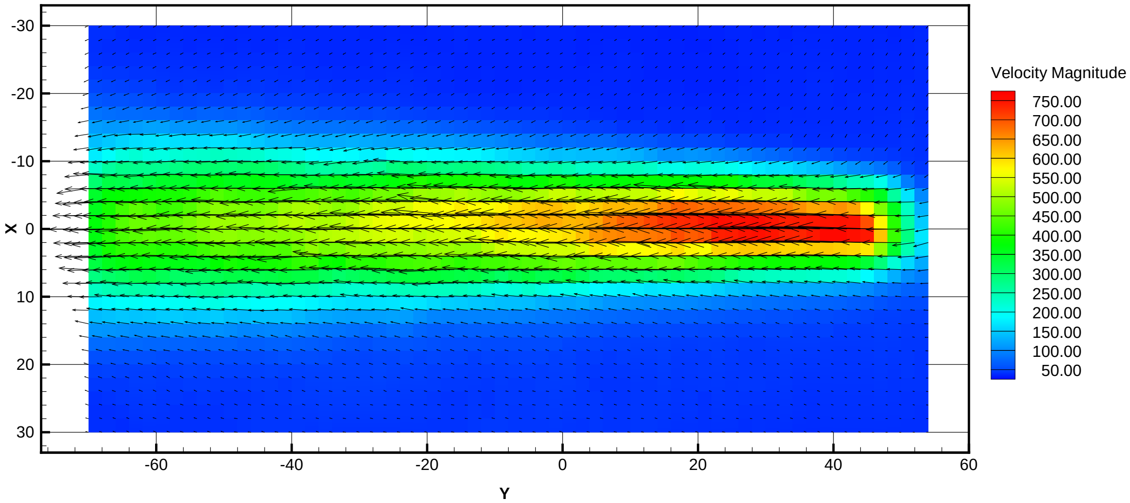


Figure 16.2-8: XY slice through the center of the jet, filtered by standard deviation during averaging and then smoothed by correcting the outliers. The colors show the magnitude of the vectors in each voxel (in mm per second).



16.3 Conclusion

Even though it was performed in undesirable conditions, the data can be used to at least get an idea of the full capabilities of a DDPIV result. Because we have a fully volumetric data set, we can calculate a complete vorticity vector field. Figure 16.3-2 shows isosurfaces (three-dimensional contours) of vorticity magnitude, sliced at the midplane of the jet and just after the nozzle to show the cross-section.

Figure 16.3-1: Velocity field of the jet in four XZ slices. Contours show the velocity magnitude of the cross section, showing the evolution from a top-hat profile to a Gaussian shape, and the vectors are the components tangent to the slices. Note the apparent circulation around the jet.

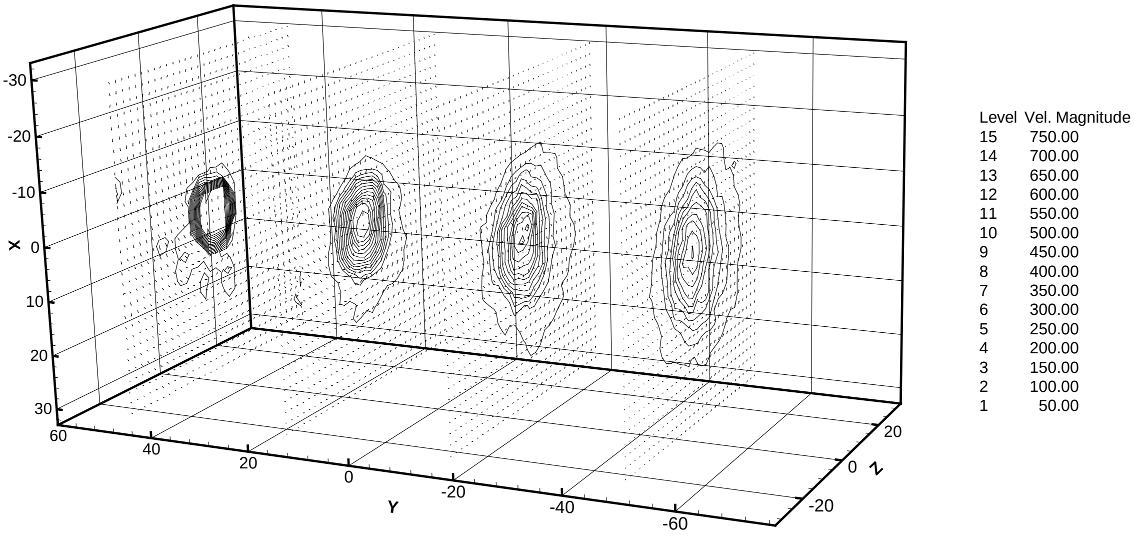
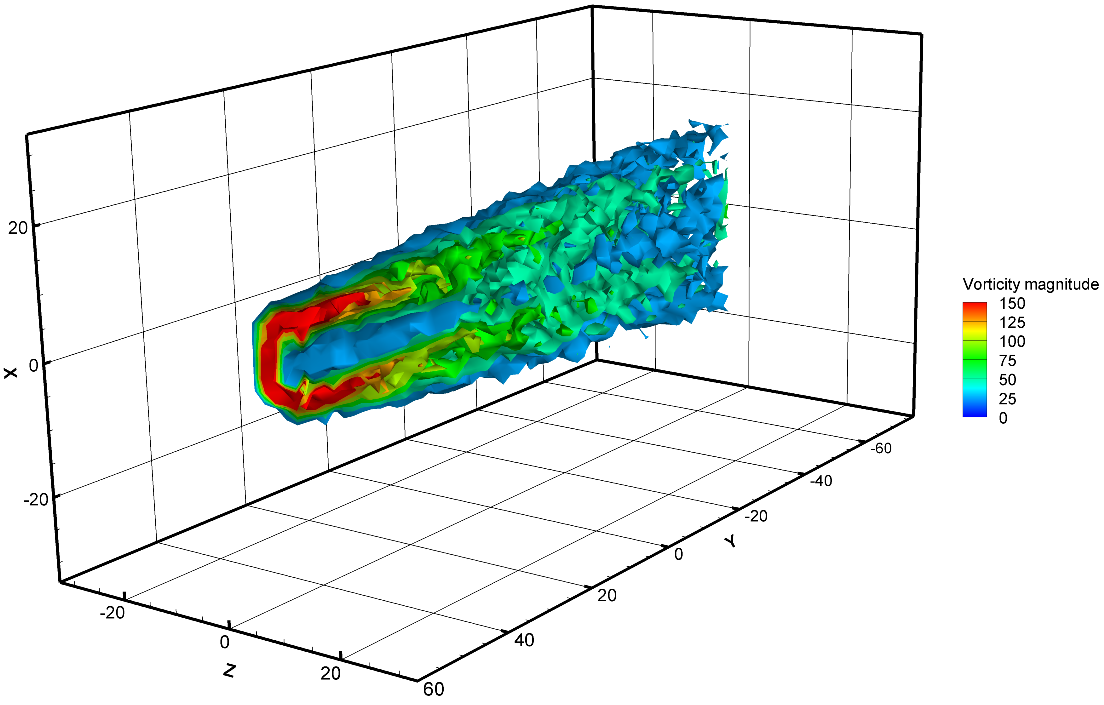


Figure 16.3-2: Vorticity field of the jet (filtered but *not* smoothed), calculated by finite difference.



Chapter 17

Propeller in a Bubbly Flow

17.1 Experimental Details

The purpose of this demonstration was to produce the largest ever DDPIV data set, both as a test of the Ian Camera itself and of the software prior to its delivery. The idea was to track the flow of bubbles through a model ship propeller in a water tunnel.

The water tunnel test-section, with cross-sectional dimensions of 150×150 mm, had been bent into a slight "s" shape from years of improper moving of the tunnel from one lab to another. The bend was slight (just a few millimeters over the almost one-meter length) but was enough to necessitate in-situ dewarping. Ten planes were imaged in the tunnel, which corrected the optical distortion of the bent test section without a problem. The contraction in the tunnel was also slightly deformed and there was a small (≈5 mm) backward-facing step where the test section did not meet the tunnel at the right place. Since there was little hope of obtaining a clean flow without rebuilding the tunnel, the bubble generator was placed right at the entrance to the test section (providing almost 1/3 area

Figure 17.1-1: CAD model of the propeller setup (flow is from right to left). The propeller is driven via a belt at a 1:2 ratio. The plate mounted to the axle of the stepper motor runs through a photo-interrupter used to phase-lock acquisition.

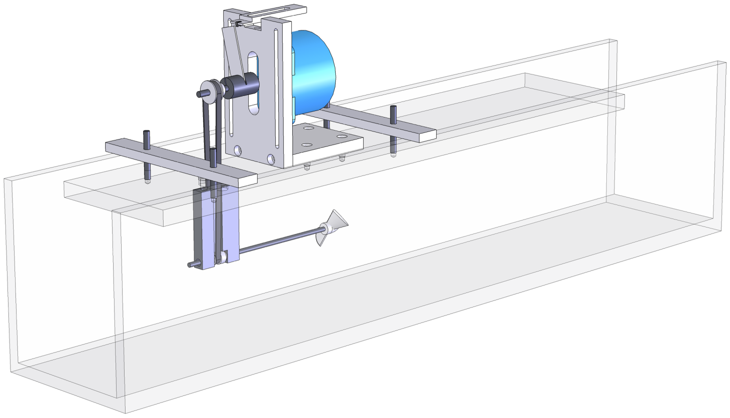
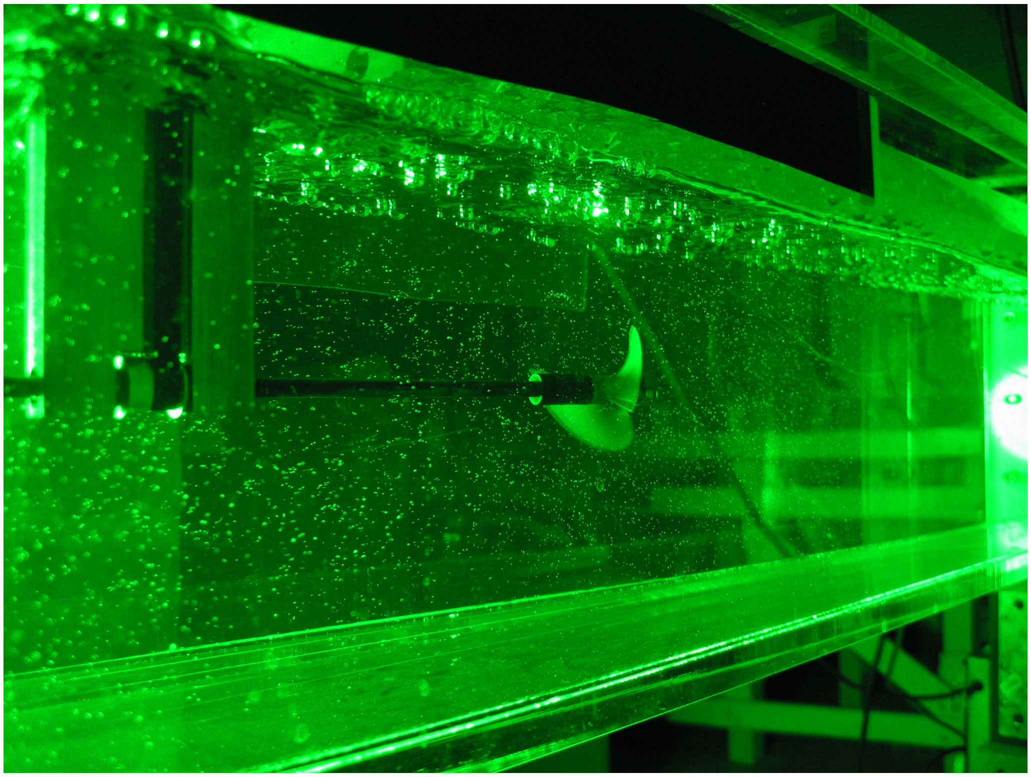


Figure 17.1-2: Photograph of the propeller and bubbles under laser illumination..

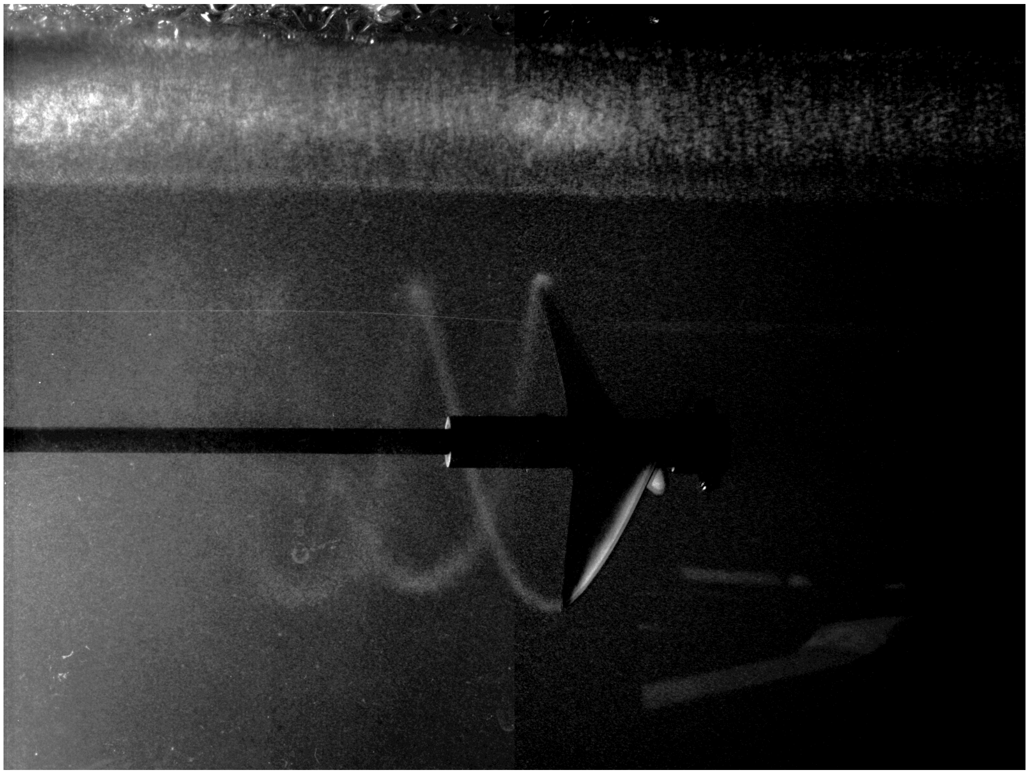


obstruction), guaranteeing a lot of swirl. The bubble generator was a porous air stone mounted as the large face of a cylindrical brass chamber fed by compressed air. By the time the bubbles reached the propeller (almost 50 cm downstream), extremely large bubbles had already surfaced, and what remained was a semi-stratified mixture of small to medium-sized bubbles.

The two-blade model propeller, with approximate tip-to-tip diameter of 65 mm, was mounted in a pull configuration and driven by a stepper motor via timing belt and pulleys. The pulleys were in the ratio of 1:2, that is, the propeller spun twice as fast as the motor. A C-shaped photo-interrupter module was mounted above the motor, and a plate attached to the motor shaft ran between the emitter and receiver once per turn to generate the phase synchronization pulse. Two hundred pairs of images were acquired at 37 phases, 5° apart, to resolve the 180° span between blades. The phase delay was calculated based on the angular speed of the motor and was added to the phase pulse from the photo-interrupter to generate a master synchronization pulse off which the camera and laser timing was referenced. For the phase-resolved data set, the speed of the propeller was approximately 15 Hz (3,000 RPM) which gives a Reynolds number of ≈210,000 based on the propeller diameter. The free stream speed was approximately 450 mm per sec.

Because the bubbles on average were relatively large (on the order of 500 µm diameter), the laser illumination was set up with a 120-mJ Nd:YAG laser in a forward-scatter arrangement. Typically

Figure 17.1-3: Average of 400 raw images at one phase of the propeller experiment. The concentration of bubbles in the blade-tip vortices is obvious. (This is an average of two sets of pairs, so on close inspection, the blade movement from frame 1 to 2 is visible.)



if translucent seeding is small enough, illumination can come from any angle, but if it is too large, certain angles can generate two particle images per bubble which is, of course, undesirable. Forward-scatter illumination is very tricky with DDPIV cameras since each aperture will form a different angle with the light cone, and thus in the measurements the bubble density appears to be higher in certain places than in others because there is a substantial intensity difference for each aperture throughout the volume which is hard to balance and still avoid saturation.

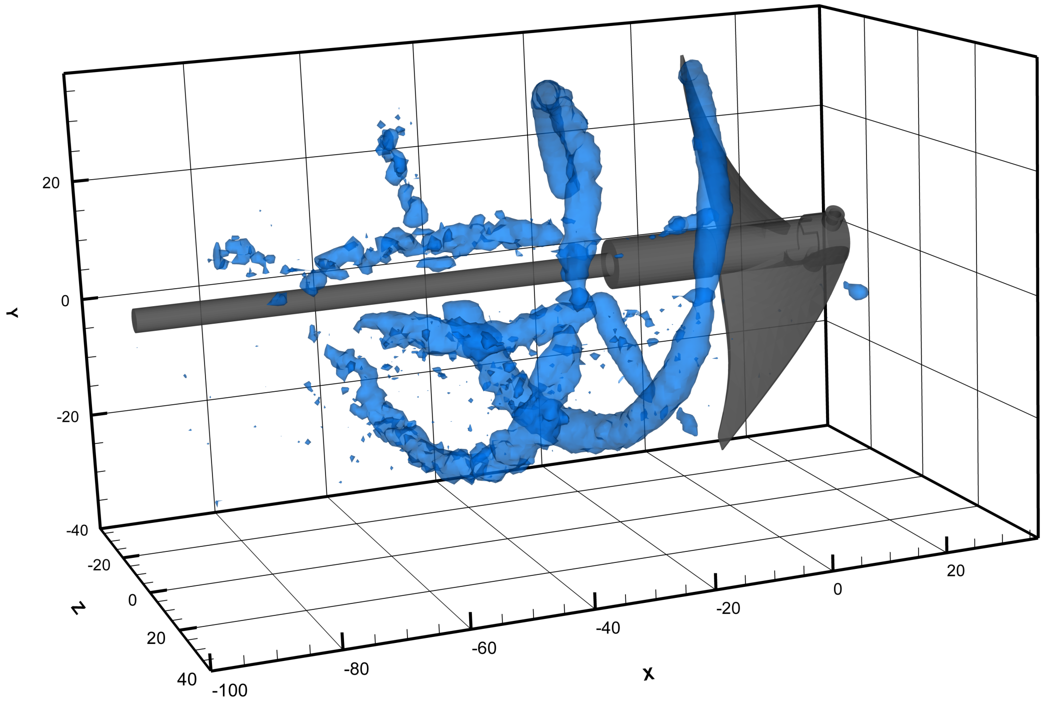
The results were phase-averaged to obtain both velocity data and population statistics every 1 mm in each direction (for a total of nearly 800,000 data points per phase). The seeding density, on average, was approximately 0.005 particles per cubic mm; with a Δt of 750 µs, the experiment was recorded at $\Phi \approx 1.41$. Although not optimal (the yield, on average, was 65%), it is certainly better than in the jet experiment. The average displacement comes to about 500 µm, the average signal-to-noise ratio is good.

17.2 Results

It should be understood that the bubbles in the experiment were generated by the air stone and *not* cavitation. Moreover, they are definitely buoyant; there is a vertical component in the free-stream velocity within the measurement volume—they were not intended to be flow tracers. The propeller was mounted backwards to the configuration of most ships; that is, since the propeller is pulling (and its shaft is in its wake) there is no typical hub vortex.

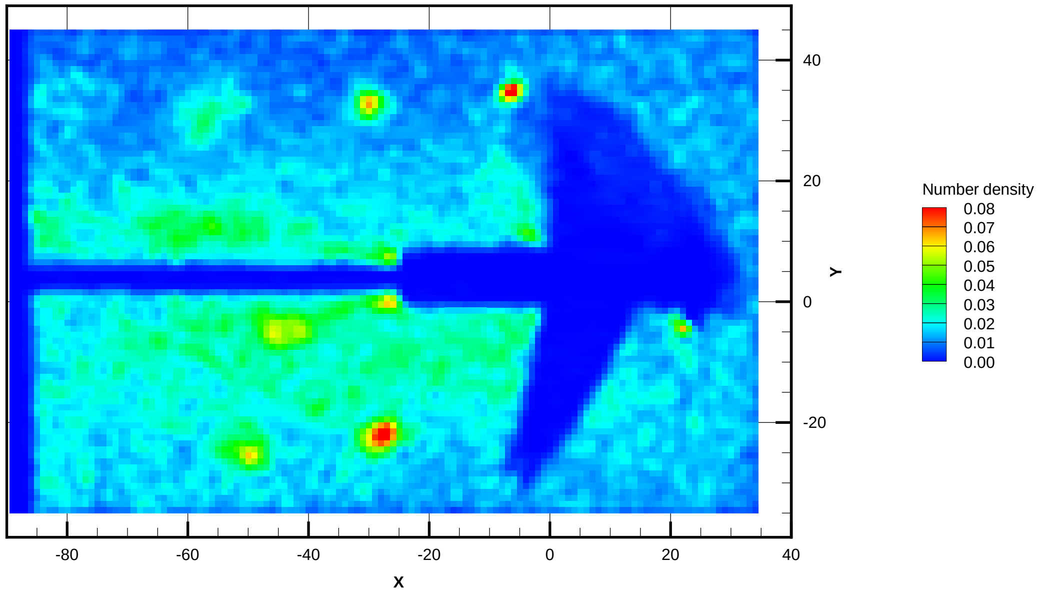
As figure 17.1-3 shows, the collection of bubbles on the tip vortices is easy to see. After processing, the vortices can be visualized in the same way, by forming isosurfaces at a designated population level. In the case of figure 17.2-1, isosurfaces are drawn at a level 3 times that of the free-stream population. This reveals a second vortex pair which hugs the shaft of the propeller which is not visible in the raw images. (It is thought that these vortices emanate from the trailing edge of the propeller hub, which is a larger diameter than the shaft.)

Figure 17.2-1: Propeller vortices visualized by bubble concentration. The blue blobs are isosurfaces of concentration set at approximately 3 times the number of bubbles as the free stream. Interruptions are caused by "shadows" in the data. There are three "shadows" since there are three apertures (these are of course not really shadows but rather obscuration of the fields of view) plus a real shadow since the propeller was illuminated from behind ($Z+$). The propeller shown here is not exactly the shape of the one used; it is meant to serve as a correct-scale schematic.



It is clear from figure 17.2-2 that the tip vortices are clearly discernable by population density up to at least 180° (helix angle) from the blade tip. It is possible, because of the full three-dimensiontality of the processe data, to track the evolution of the vortices in terms of the number of bubbles being trapped in their cores.

Figure 17.2-2: Contour plot of average number density (defined as number of bubbles in a voxel divided by the number of frames divided by the number of voxels) in an XY slice cutting through he axis of the propeller.



By compressing all the phase data sets to a slice one voxel thick in X, and averaging together, the circular base of the almost-perfect helix is exposed, and by fitting a circle to the concentrations of bubbles, the "exact" center of rotation of the propeller can be obtained. Each phase set was combined into a concatenated particle cloud of all 400 images, rotated about this YZ rotation axis every 5 degrees, then cropped into a volume only one voxel thick in Z (centered about the rotation axis) to obtain the population statistics of a thin slab 2 mm thick (similar to figure 17.2-2). Each phase data set was thus transformed into 73 slabs which represented the cross-section of the vortices at intervals of 5°. From one such set of slabs it is possible to construct a bubble population history (as a function of angular distance from the blade tip). Because the four obscurations in the volume are constant in space, they are at different angular locations (relative to the propeller blades) for different phase sets, thus all the data sets combined should overlap to create a complete averaged population history.

Automatic detection of the vortex cores was performed on the slabs themselves by scanning them along common X-coordinates (columns). Scanning in this fashion the cores could be identified simply as the point of largest bubble population along a single X-column either above or below the propeller shaft. The average population of the entire slab was used as a threshold (that is, any points with population lower than the average were ignored as possible cores). Any vortices not near the tip of the blades are discarded (the hub vortex is not too strong and hard to detect automatically).

After scanning in this fashion it is of course possible to find “multiple” cores next to each other (in adjacent columns) since the strong parts of the vortices frequently span several columns. The list of potential cores was then filtered by assuming that no two cores are within 10 mm of each other, and any potential cores that fall within this distance will be discarded and replaced with the one that has the highest population concentration. The points within this region were sampled to generate a list of population versus radial distance from the center for each vortex core. The center was taken as the point of highest concentration (that is, no “smoothing” by surface fitting or similar methods was performed). When calculating the radial distance, the X coordinate had to be corrected by the approximate helix angle of 16.8°, since the helical shape causes the vortex to have its axis at an angle to the YZ plane (but lies on the XZ plane).

If there are four vortex cores, then it is easy to ascertain which is which by designating them as “bottom right”, “top left” (vortex 1) and “top right”, “bottom left” (vortex 2), in order of decreasing intensity. If there were more than four cores in a slice (which at this point would only occur if an extraneous point or the hub vortex passed the threshold test) then the four strongest would be taken to be the tip vortices. If three cores were detected (as long as two of them were “top” or “bottom”) they can still be sorted, but any less (which would occur in the shadows) and the slice is discarded. Once the cores are sorted, it is only necessary to calculate the angular distance between them and the blade tip. Since we are averaging the two vortices together and this is a two-blade propeller, then at every rotation $\mod 180^\circ$ we are back at 0. Any vortex cores on the “left” side must have 180° added to the angle since they are the same vortex crossing the slice on the opposite side of the propeller in the opposite direction.

Each core was processed in this manner for each of the 73 rotations of each of the 37 phase sets. The goal was to average all the sets and the two vortices together. Close examination of the individual phase data sets indicated that there was simply too much of a difference in the population distribution between sets. Figure 17.2-3 shows the data for the same angular distance from the tip for all phase sets. Even though the shape is similar, the magnitude of the concentration varies substantially. This is due to several reasons. First, each phase set is recorded at a different propeller position, but the seeding and illumination, with all its variances, is of course fixed to the lab frame. Thus at different phase angles, the bubble population going around the blades can vary substantially, since the larger ones rise faster. More importantly, since the intensity of the bubble images varies with x and y, so does their detectability, since the software uses a constant threshold to mark possible particle images. Moreover the pressure in the air feeding the bubble generator varied with time the actual seeding density was not constant between phases (nor was the size of the bubbles)—the average density varied by up to a factor of two from phase to phase. Lastly some phase sets have certain angular distances obscured by the four shadows present in the measurement. There is no reason to believe that the concentration of bubbles at the vortex is linearly dependent on any or all of these factors, so correction of the data sets is impossible.

Figure 17.2-3: Bubble population distribution, 35°from tip, all phase sets. Various factors contribute to the fact that, especially near the center, there is a huge variation in bubble population from phase set to phase set.

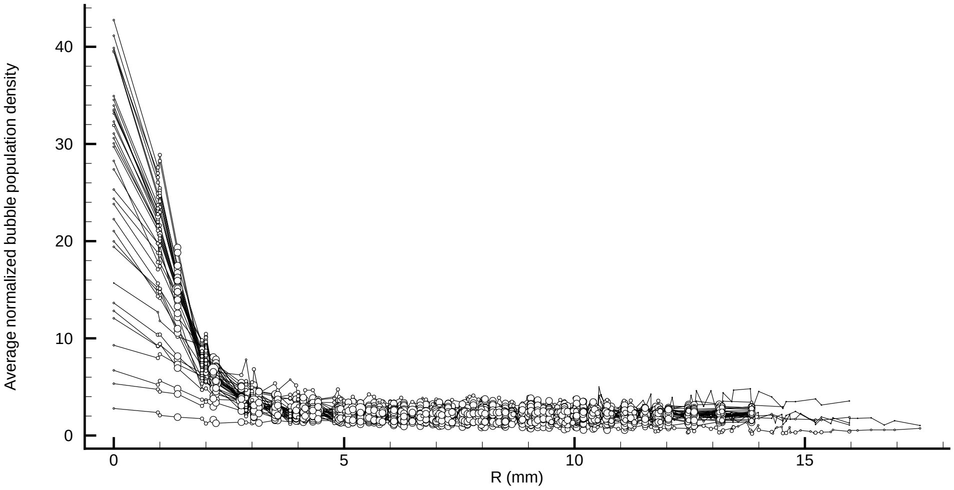
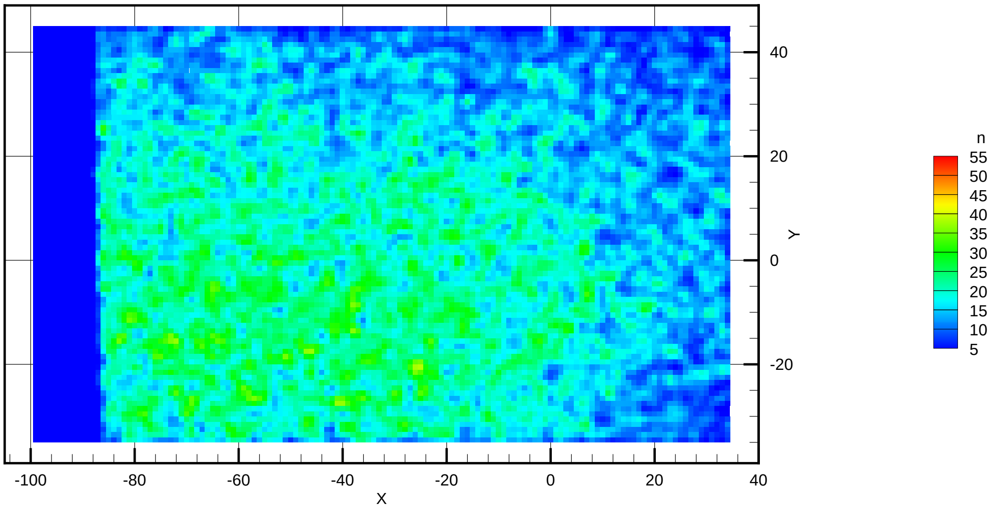
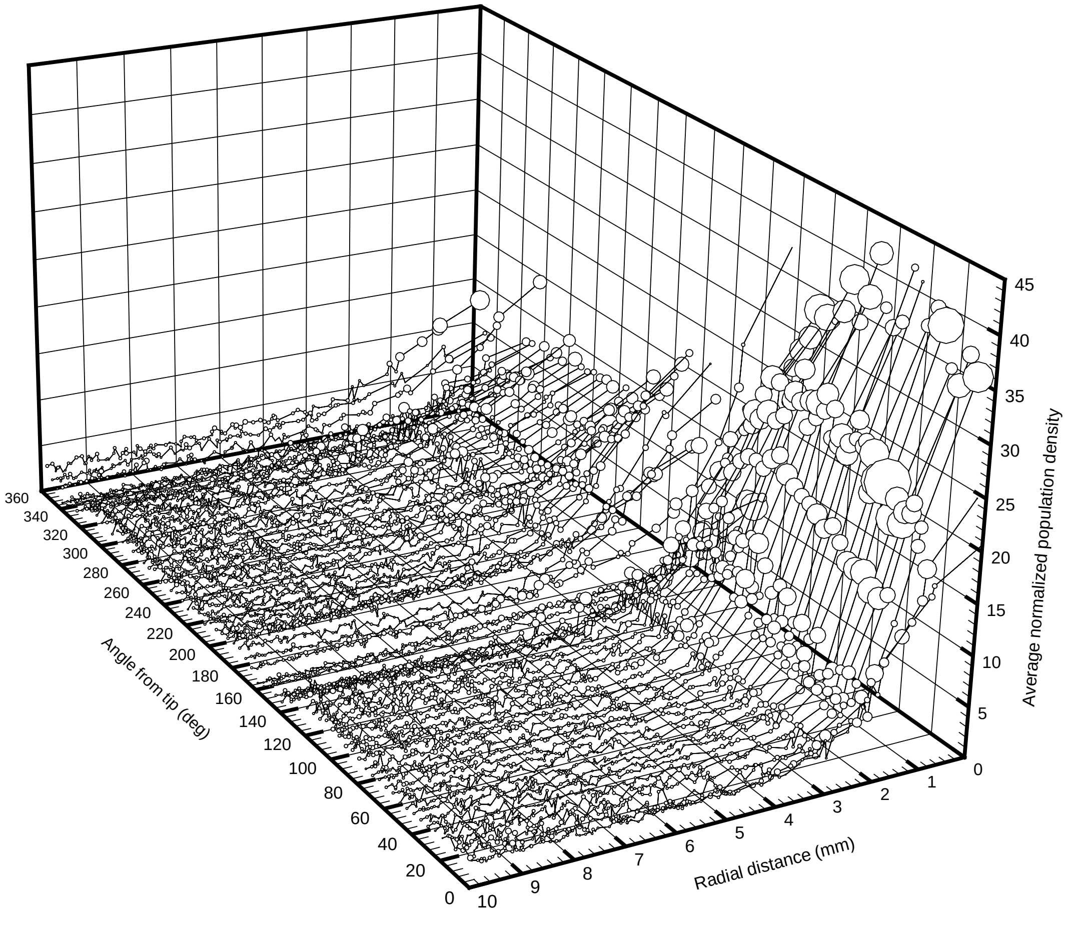


Figure 17.2-4: Contour plot of a slice in the mid-plane of the test section in the region where the propeller was placed for the freestream. There is a clear variation in detected seeding density with space due to both physical bubble distribution and the illumination setup.



We cannot examine the population distribution across phases, but the basic shape of the distribution can be obtained by looking at it for any particular phase. One of the most complete sets (that is, one of the ones that was least affected by the four shadows) is that of the 5° phase, shown in figure 17.2-5. At this phase angle the cross section of propeller was minimized on the image. Aside from physical obscuration as mentioned before, data can be shadowed also by reflections of the laser off the blades themselves, which will generate a speckle pattern that will become indistinguishable from particles and must be removed physically from the image to avoid generating ghosts. If the speckle is bright enough, after removal nothing will be left on the image there, so the resulting point cloud has a sectional "hole" through the Z axis which has a cross-section the shape of the blade reflection. Thus minimizing the cross section on the image maximizes the areas where particle images are undisturbed by the removal of the blade reflection.

Figure 17.2-5: Bubble population density as a function of radial distance from the vortex axis and angular distance from the blade tip for the 5° phase set. This is the most complete set, since with the propeller lying at an angle of minimum cross-section in the image, reflections of the laser off the blades interfered the least with the data set. The size of the point is equivalent of the error bar (scale of 1:1), calculated as the standard deviation of the average at that point.



Bands of no data in the figure correspond to the four shadows (obscurations) mentioned before. The first few angular distances (to 10°) are disturbed by the image of the blades themselves. The population concentration peaks between 45° and 60°, a trend that can be seen in the other phase sets. At an angular distance around 220°, the peak population density in the vortex becomes nearly flat as a function of angular distance. If the vortex diameter is characterized by a threshold population density (as with the isosurfaces), the plot shows clearly that the vortex appears to maintain a constant "size".

The bubble velocity field was phase-averaged on the same voxel set so that the population and velocity data would coincide. Figure 17.2-6 shows the contribution of the vertical component of velocity to the magnitude; it is clear that the bubbles were large enough that they do not follow the flow of the water (the freestream measurement is also an average of 200 pairs and the voxels are the same size as the propeller measurements).

Figure 17.2-6: Slice at the ceneter of the test section showing the vertical component of velocity as a percentage of the total magnitude (unfiltered vector field). Flow is from left to right.

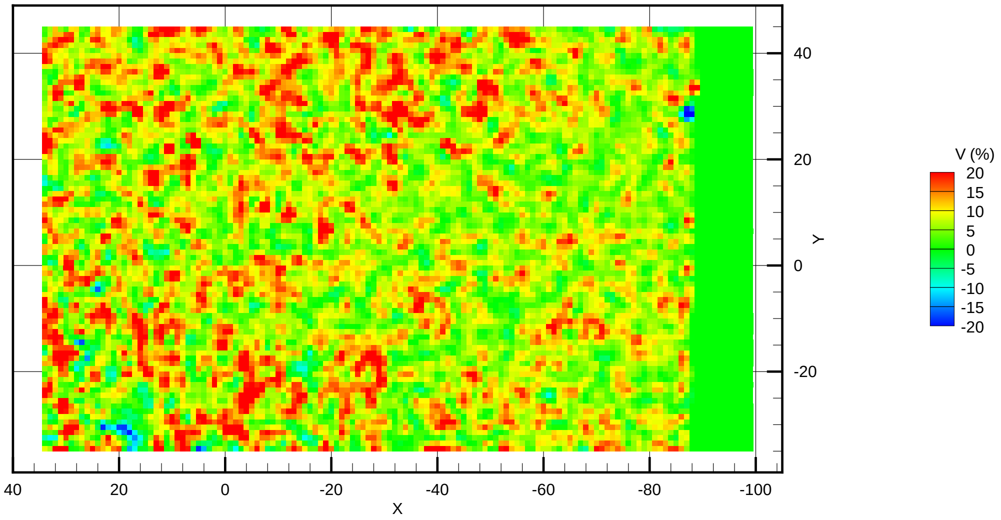


Figure 17.2-7 is analogous to what would result had the measurement been performed with SPIV. With the full volumetric measurement of DDPIV, figures like 17.2-8 and 17.2-9 are possible.

Figure 17.2-7: XY slice of propeller bubble velocity field through the propeller axis, 5° phase. These three-component vectors in a single depth slice are the equivalent of an SPIV measurement.

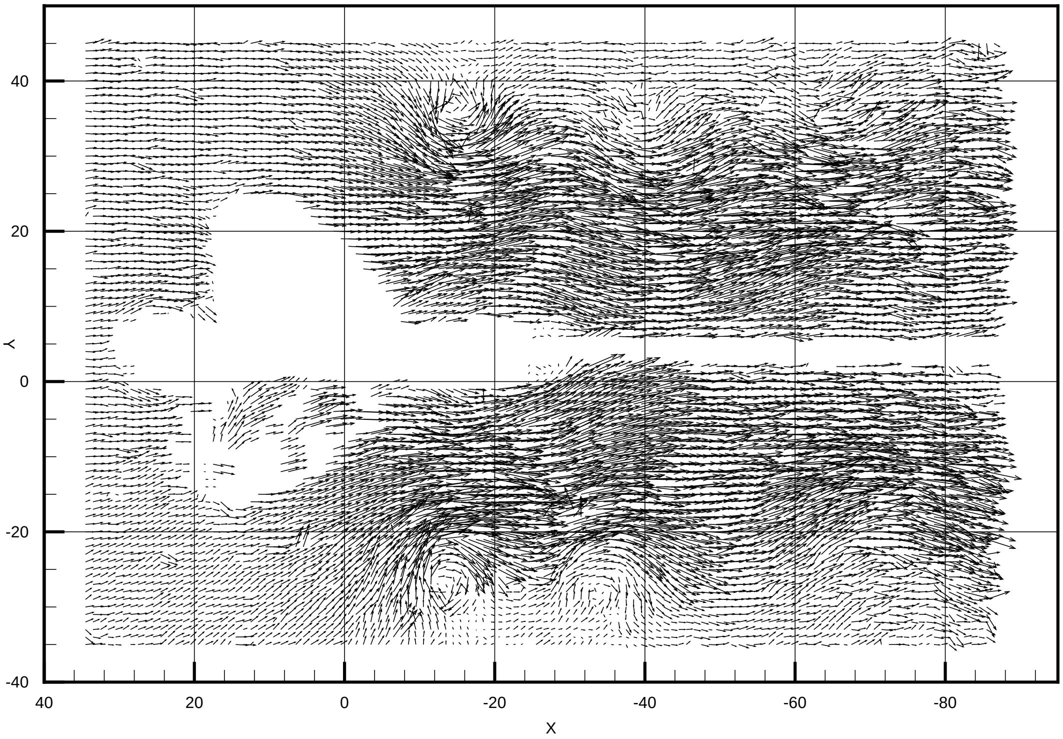


Figure 17.2-8: Velocity and population data for the 5° phase of the propeller experiment. The magenta blobs is an isosurface of population density highlighting the vortices. The three-dimensional streamlines are color coded with velocity magnitude.

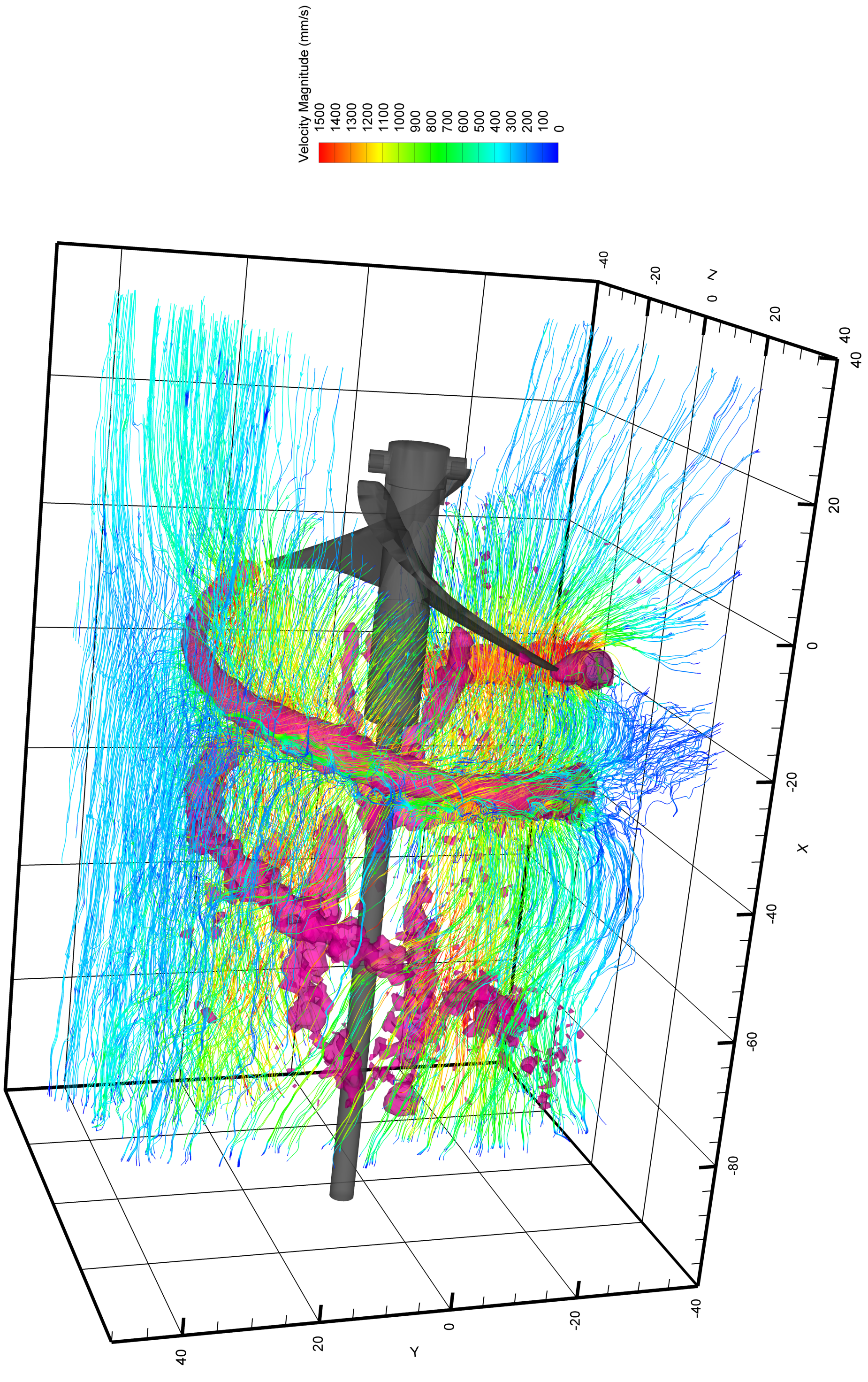
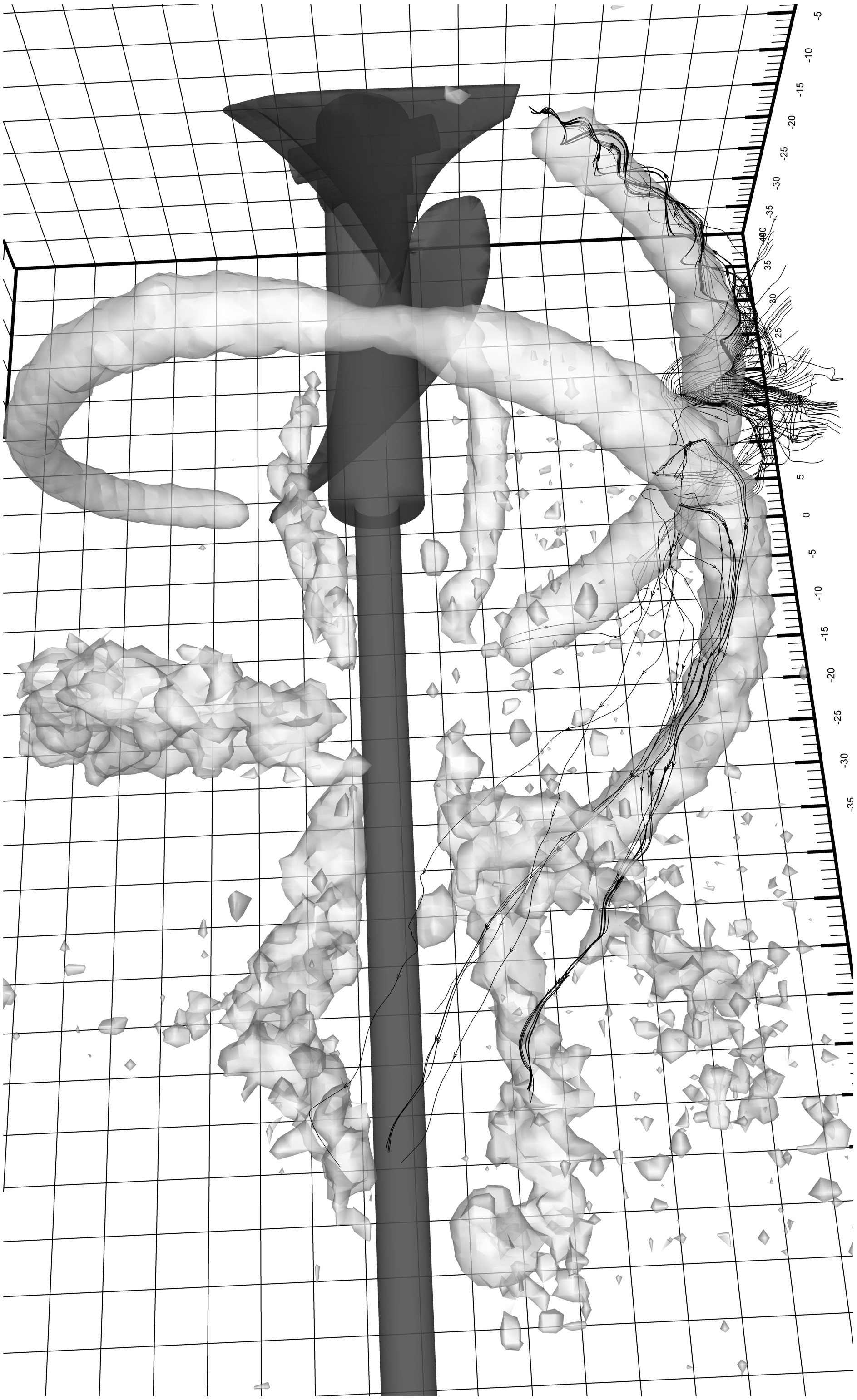


Figure 17.2-9: Detail of bubble streamlines showing a spiraling flow into the tip vortex of the propeller blade.



Chapter 18

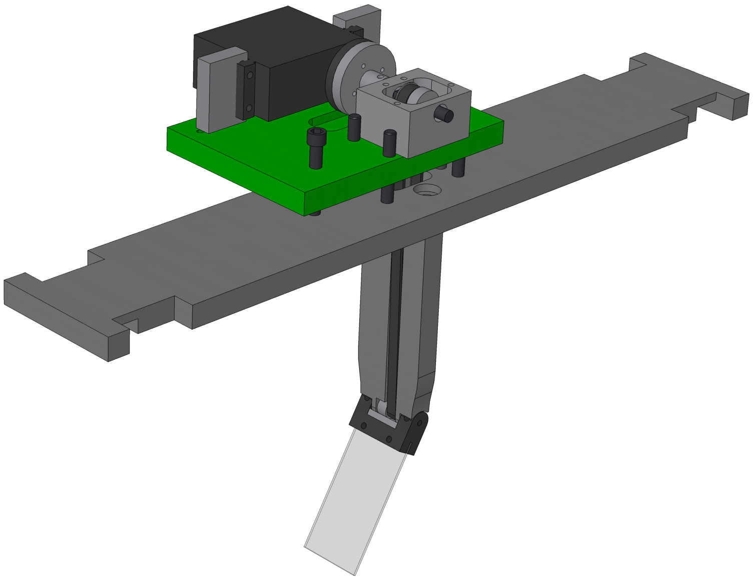
Flapping Flow

18.1 Experimental Details

In this demonstration, the objective was to map the flow generated by the flapping motion of a rectangular plate. To increase resolution, the flow was measured in a phase-locked manner, averaging 30 sets together per phase. As mentioned in section 15.4, the measurement was performed at $\Phi \approx 2.28$. The tank and seeding were identical to that of the jet experiment. The voxels used for averaging are 5 mm to a side and 2.5 mm apart in each direction; there is no filtering in the average but there is a slight smoothing.

As much a test of the Emilio Camera, this was a test of the ability to run an automated, repeatable

Figure 18.1-1: CAD model of the flapper setup. The flap itself is a recangular piece of glass (to minimize shadows) 1.2 mm thick, 38 mm wide, and 76 mm long. The platform on which the servo is mounted could be raised or lowered via the two screws to adjust the belt tension. The area around the flapper's pulley could be covered by a thin sheet of plastic (like a transparency) fitting snuggly between the flapper mount's countour and the pulley chamber to remove any influence of the belt on the flow field.



experiment. The manual acquisition of each phase set in the propeller experiment required over 3 hours of user time. Because the objective here was to measure a start-up flow, several minutes had to pass between runs, and depending on a human user would be too inconvenient.

The flapper itself was driven via timing belt from a digital servo-motor (to allow for almost arbitrary velocity programs). A belt is by no means optimal in this case, since it is hard to account for its compliance, but using a plastic belt (rather than rubber) at high tension—albeit destructive in the long term—provided adequate repeatability in the movement; there was no option to move the encoder from the motor to the flapper itself. More favorable design variations would have been extremely difficult to implement in such a low-profile design impervious to being partially submerged considering the time available. In the end, although the motion was not as consistent as in the propeller experiment (and it was a bit sticky at the slower speeds), fluctuations were well less than half of the voxel spacing (2.5 mm).

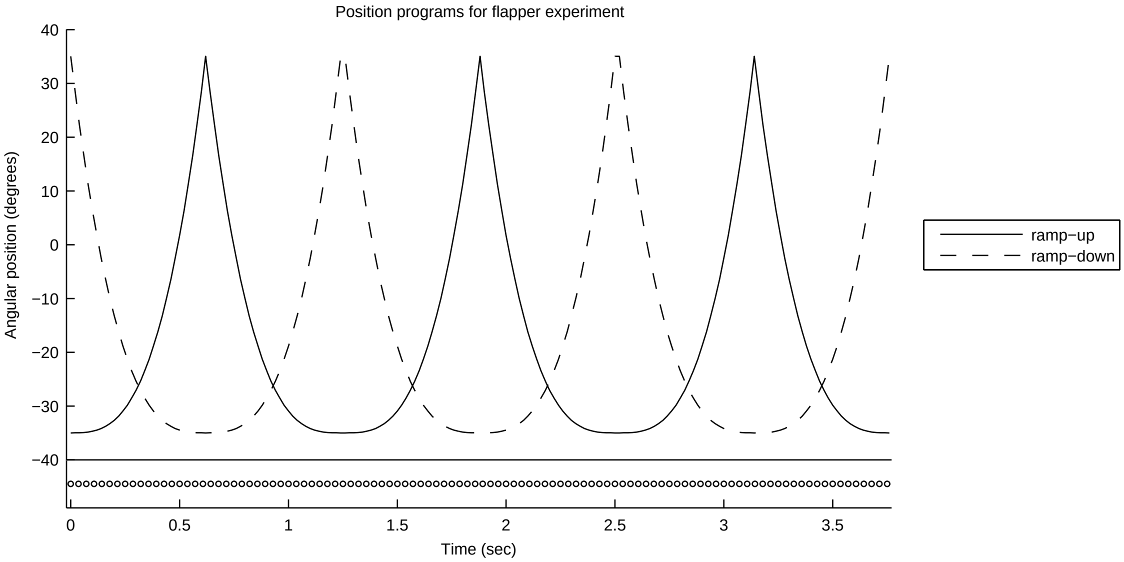
To assure accurate timing, the control of the servo-motor was accomplished by programming of custom Firmware on an Atmel AT90USBKEY demonstration board. The actual motor control was via pulse-width modulation on one of the microcontroller's two 16-bit counters. Another counter was used to generate a "synch-out" pulse with a precise delay between the leading edge of this pulse and the start of the motion. The controller was made accessible to the acquisition computer via a virtual COM port (on the USB bus). The Visual Basic program used to communicate with the motor controller for the dewarping traverse (which makes automatic multi-plane dewarping possible) was modified to communicate with this new custom servo-motor controller.

Laser and camera timing was, as usual, accomplished via a National Instruments PCI-6602 digital counter board. For the flapper experiment, a new timing program was written that allowed control of the pulse generators on the card from the command-line (among a host of other functions).

Video Savant, the software used for image acquisition, is capable of running `Python` scripts for acquisition or playback. With all the components of the experiment now controllable from the acquisition computer via command line, the entire experiment could be run from a single `Python` script which recorded a set (3 cycles), shook bubbles off the flap (by moving it between extremes at maximum speed), exported the images, and waited a set amount of time (10 minutes) for the flow to come to a stand-still, changing the phase delay as necessary to capture the different phases. `Python` scripts on other computers in the laboratory watched for exported images and moved them to processing drives via the network, freeing up resources on the acquisition computer so as to leave the experiment uninterrupted. These scripts would also invoke `DDPIV` via command line and process the images essentially as they were recorded using a configuration fine-tuned by the user using test images.

The recording `Python` script output a log which was made visible via HTTP on a dedicated Linux server, allowing the progress of the acquisition to be monitored remotely. A webcam aimed at the

Figure 18.1-2: Position programs for the flapper experiment. The circles represent the approximate times at which an image pair was acquired.



experiment placed images every two seconds during acquisition onto the same server to allow the user to check for any physical problems—although these images leave much to be desired (they were either completely black or blasted with green laser light), this is how it was discovered about halfway through that the first belt had broken. Because the phases were not recorded sequentially, this belt failure, which necessitated that the flapper assembly be moved relative to the camera, precluded the different phase sets from being shown together as one continuous animation. Although the flapper position and orientation is well-known by the very DDPIV measurement, there is no automatic way to determine the rotation/translation combination for each phase that would allow them to be combined again, and as time was now at a premium (laser malfunction had caused months of delay prior to this), they were left as separate. In the end, acquisition was an almost continuous 14-day affair, processing required an additional month or so on several computers simultaneously (including some reprocessing as new features were being added to the software concurrently).

Seeding was by no means constant due to the long wait times between cycles. It was replenished approximately once per day. Some phases had to be repeated due to over or under-seeding. In the end, four phases of each of the two velocity programs were satisfactorily recorded and processed.

The two velocity programs were actually the same cycle out-of-phase by 180° and were based on a linear acceleration (or deceleration) program (cubic position) from an angle -35° to 35° off vertical[1]. The program with increasing velocity from start to mid-cycle was nicknamed “ramp-up” and the one which started quickly and slowed as it approached the mid-cycle was nicknamed “ramp-down”.

[1] The values presented here are the programmed values. The actual motion is not exactly symmetrical; for the ramp-up program the position at the fast end was closer to 29° before the flap changed direction as a consequence of the motor being too slow.

The Reynolds number is around 20,000, taking the maximum tip speed and tip width as parameters.

18.2 Results

Figure 18.2-1: Mid-plane slice of the vector field at the end of the first forward stroke in the ramp-up program.

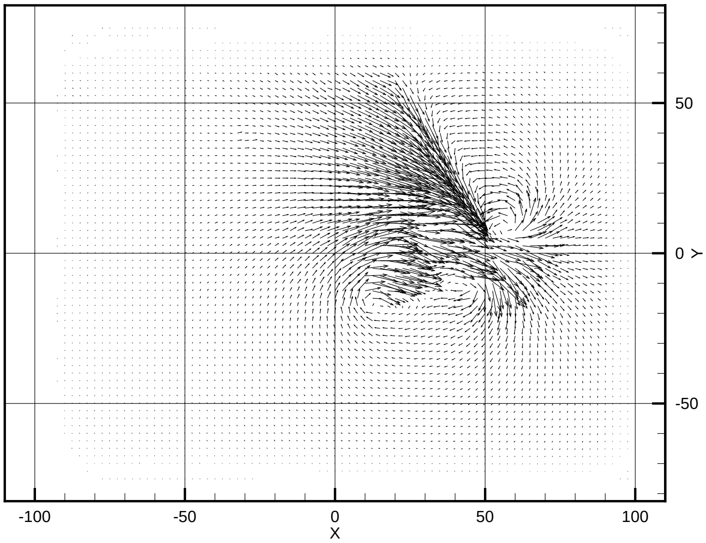
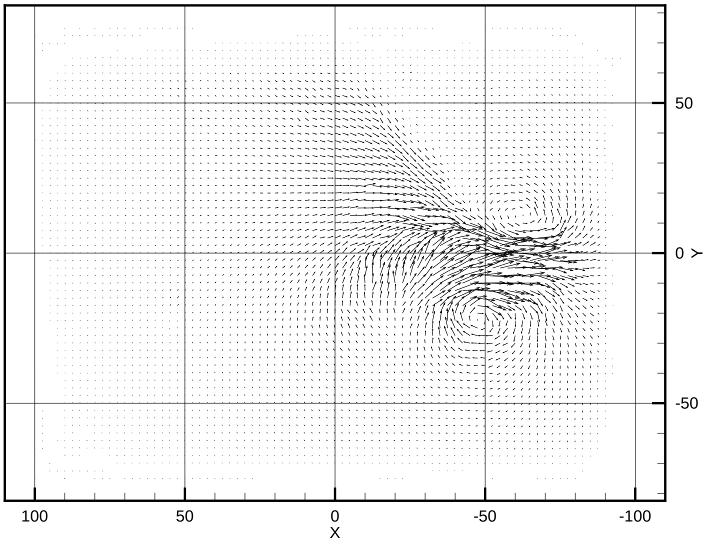


Figure 18.2-2: Mid-plane slice of the vector field at the end of the first forward stroke in the ramp-down program, seen from the opposite side to match the orientation of figure 18.2-1.



Figures 18.2-1 and 18.2-2 show slices through the mid-plane of the flap of the vector field for the ramp-up and ramp-down position programs, respectively, at the end of the first forward stroke.

Figure 18.2-3: Mid-plane slice of the vector field at the end of the second forward stroke in the ramp-up program.

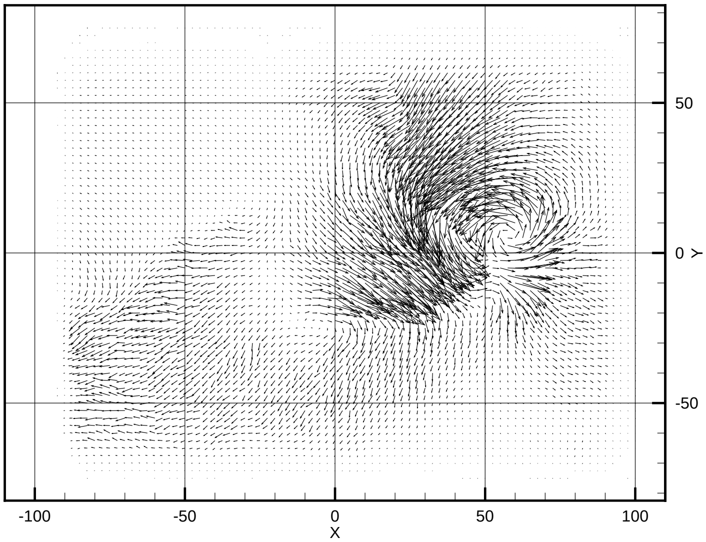
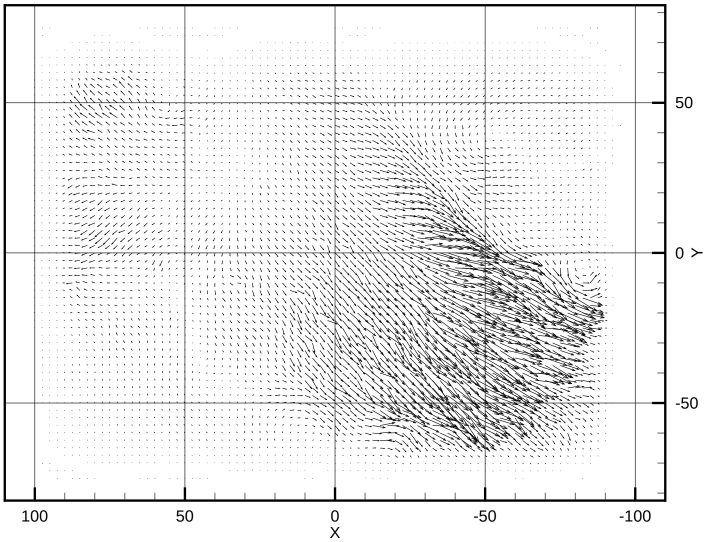


Figure 18.2-4: Mid-plane slice of the vector field at the end of the second forward stroke in the ramp-down program, seen from the opposite side to match the orientation of figure 18.2-1.



Because the two programs differ only by phase, it is the first stroke that is the most different. The ramp-down flow case is much “cleaner”, that is, the stroke ends with a coherent vortical structure being ejected from the plate’s edge, pushed by the wake flow generated during the stroke. In the next few frames, this structure sheds off almost completely from the plate edge, traveling almost exclusively in the $X-$ direction. One can visualize it by imagining that if the flap starts moving quickly and progressively slows down, the fluid that it originally dragged behind it is allowed to “catch up” to the flap and so the structures remain coherent. By the time the flap starts moving in the other direction, the main wake structure has already moved out of the plate’s reach.

The ramp-up case is completely opposite. At the end of the first forward stroke, the plate suddenly changes direction at the point of its highest speed, when the fluid is desperately trying to follow it. In figure 18.2-1 there is a vortex near $X = 15$ which was shed off the plate’s edge as it was traveling to the right. A second vortex is formed at the flap’s edge as it switches direction (seen in figure 18.2-1 at $X = 55$, but the first vortex has not moved far enough away to be immune to the action of the impending backstroke. In other words, the vorticity generated during the stroke is not shed at the same time as that generated at the end of the stroke in the ramp-up program, whereas the two are shed in sync in the ramp-down program. Moreover, in the ramp-down program, this shed vorticity has time to escape, whereas in the ramp-up case there is not enough time before the flap returns and thus the structures are annihilated.

At the end of the back stroke, the situations are reversed (since the two programs are 180° out of phase), but now the starting condition is the flow of the forward stroke (rather than no flow). Figures 18.2-5, 18.2-6, 18.2-7, and 18.2-8 show the evolution of the flow during the first forward and back strokes. In the ramp-up program, the first vortex is shed as the plate passes $X = 0$. The second vortex is generated by the plate’s rapid turn-around, rather than the flow simply wrapping around the edge of the plate (as in the case of the ramp-down program), and both are annihilated by the back stroke. At the end of the back stroke the plate is at its slowest and thus it acts like a deflector to divert the flow downward. This is exactly the behavior seen at the end of the second forward stroke of the ramp-down program; once the still starting condition has been affected the only difference between the two flows is the phase difference in the programs.

Figure 18.2-5: Mid-plane slice sequence for ramp-up program; images go from left to right with an interval of ≈ 143 milliseconds.

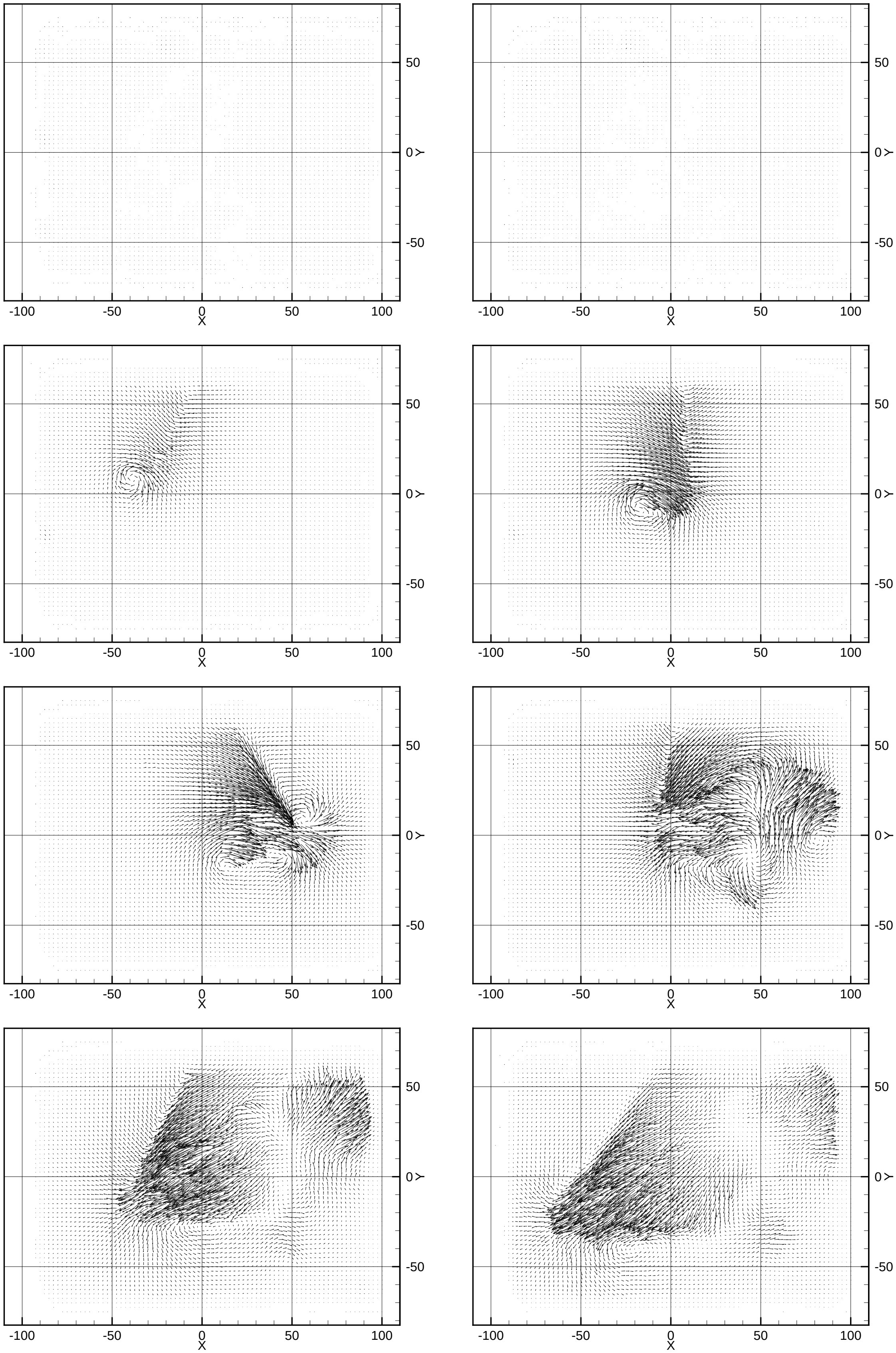


Figure 18.2-6: Mid-plane slice sequence for ramp-up program (continued); images go from left to right with an interval of ≈ 143 milliseconds.

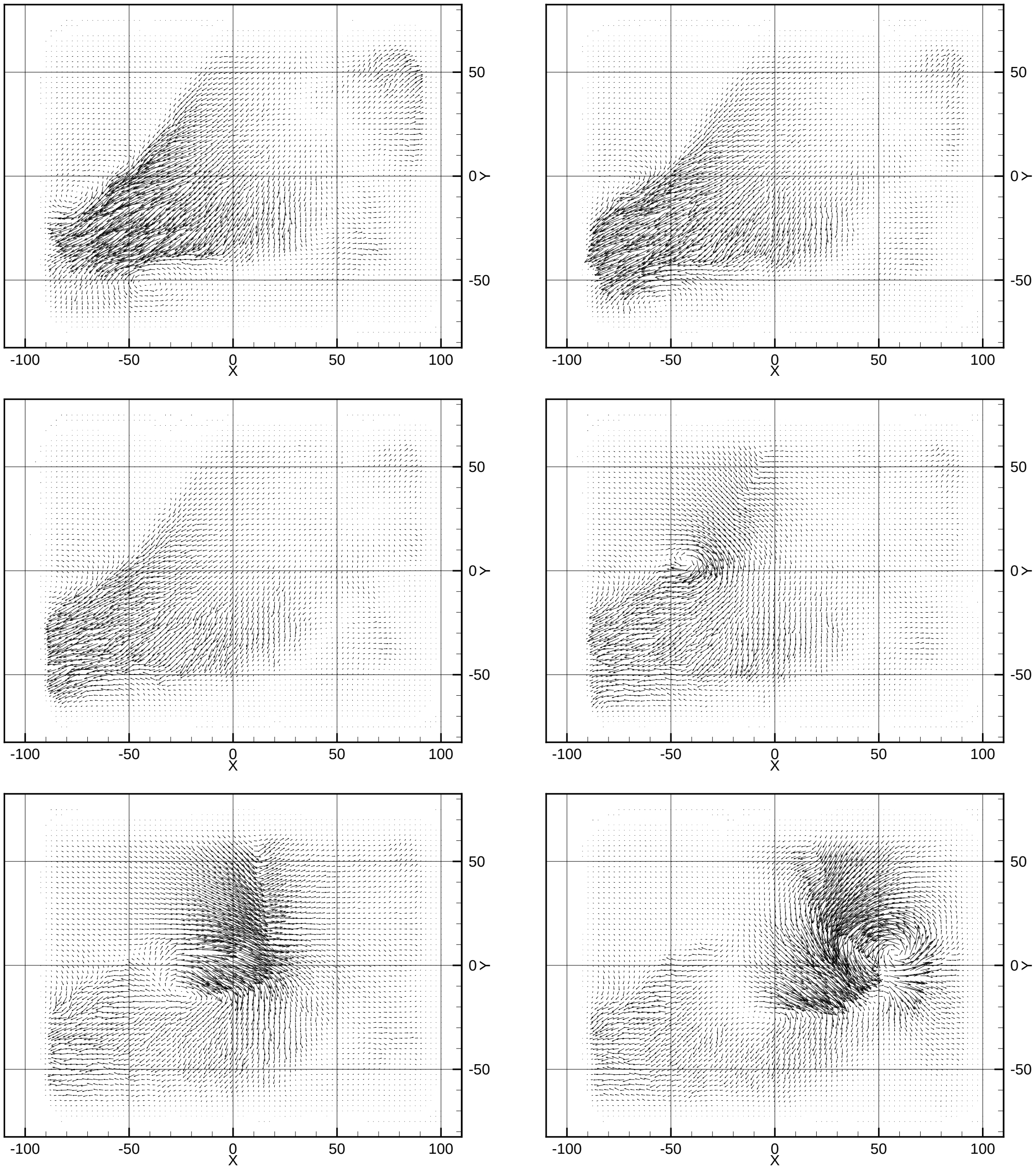


Figure 18.2-7: Mid-plane slice sequence for ramp-down program; images go from left to right with an interval of ≈ 143 milliseconds.

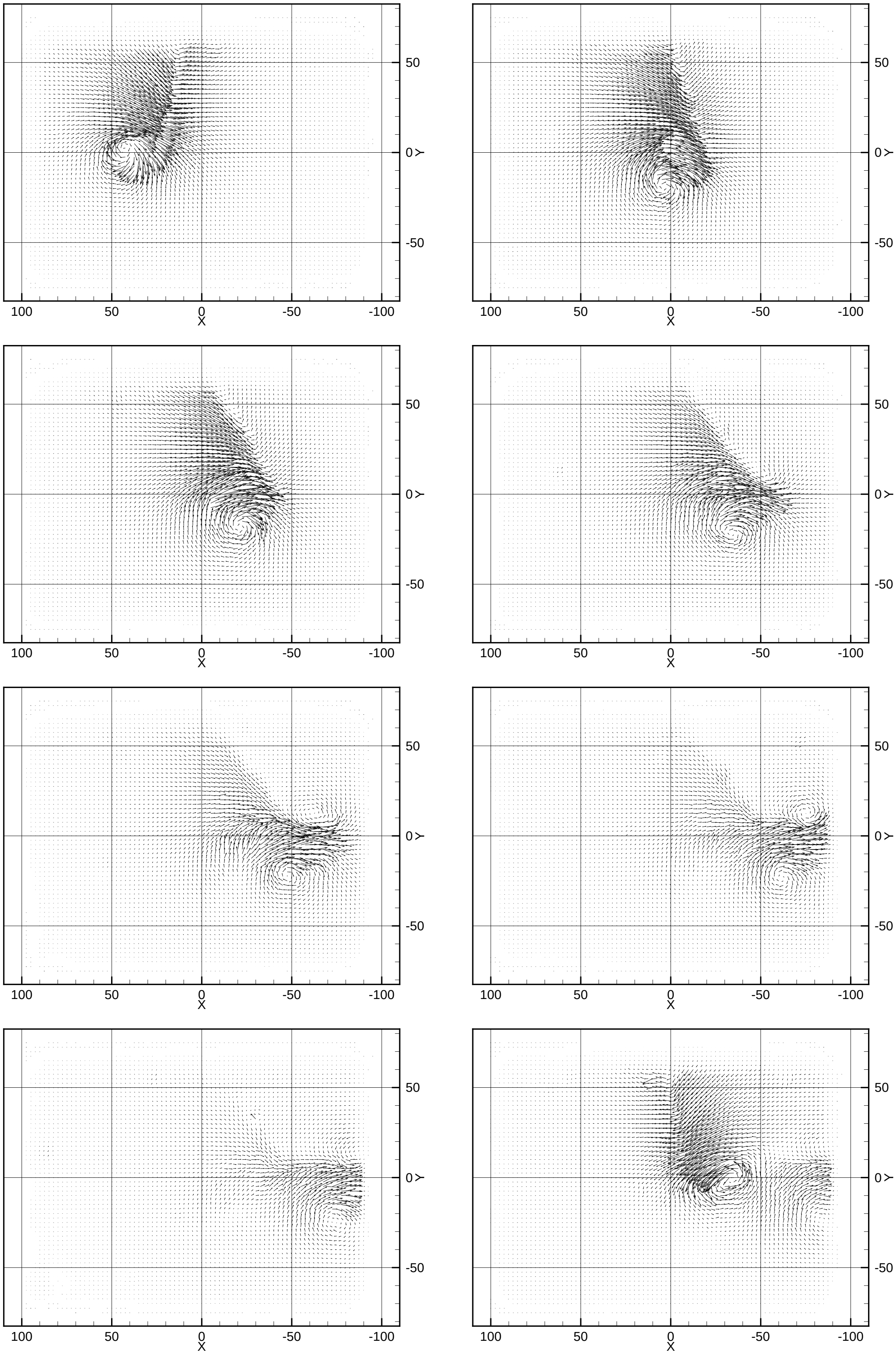
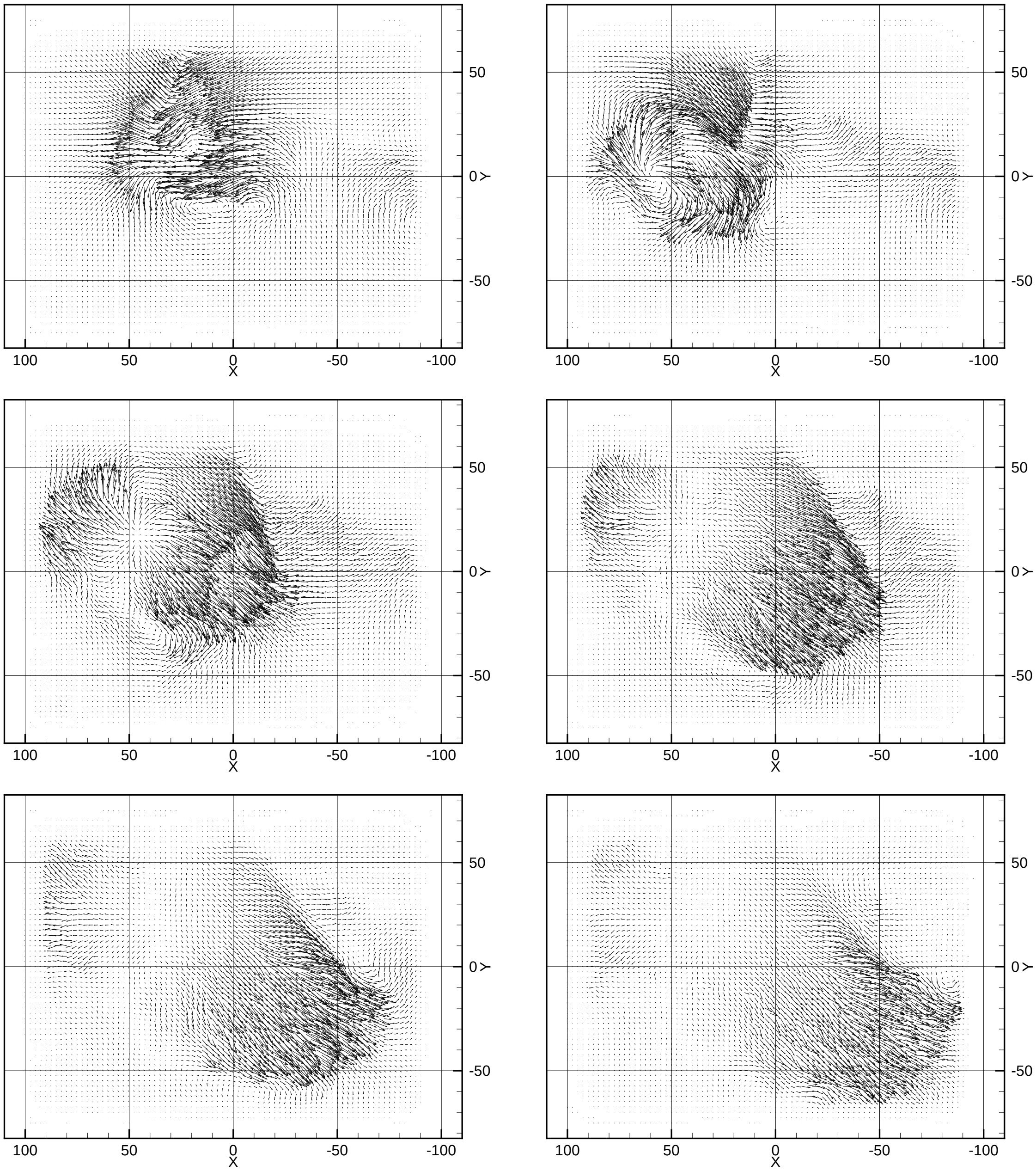


Figure 18.2-8: Mid-plane slice sequence for ramp-down program (continued); images go from left to right with an interval of ≈ 143 milliseconds.

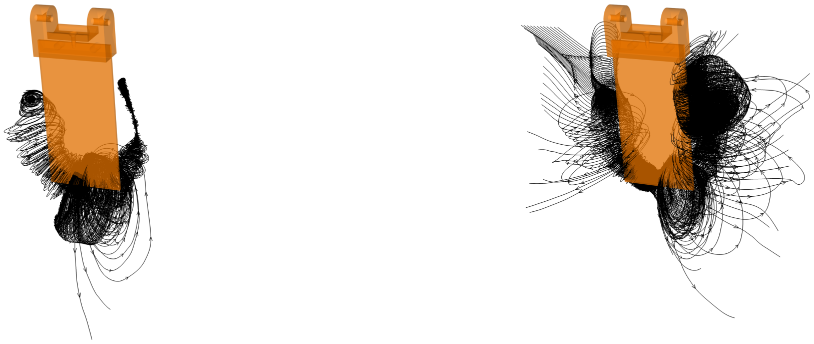


Again, this style of figure does not do the measurement justice since we have at our disposal a volumetric data set. When studying the basic shape of the flow, three-dimensional instantaneous streamlines are indispensable. From the point of view of visualization, they have the disadvantage that they are extremely sensitive to noise. For the flapper experiment, using 5 mm voxels is already pushing the envelope of the averaging, as there is an average of 11 vectors contributing to each voxel (6 with aggressive filtering). By enlarging the voxels to 7.5 and enabling aggressive filtering (filter by standard deviation and perform outlier removal) the number jumps to 16 vectors per voxel. This makes streamlines (and derivative quantities like vorticity) much smoother and easier to interpret schematically. Thus filtering aggressively has an important function in visualization, *but it should be interpreted as an atypical exaggeration of the quality of the results.* The use of smoothing vector fields by outlier correction is rampant in all PIV literature, but it should not be taken lightly when testing the instrumentation itself.

This being said, the remaining figures were obtained after aggressive filtering (2 standard deviation removal during averaging and a multi-pass, 10% threshold outlier correction). We will concentrate on the ramp-down program and analyze the structure of the coherent vortical mass that is ejected at the end of the first forward stroke.

Visualization of the velocity field by instantaneous streamlines indicates that there are two phases in the forward stroke. In the first phase, the plate is moving faster than its wake, and it can be seen that the flow field curves from the front of the plate to the back along all three edges (figure 18.2-9).

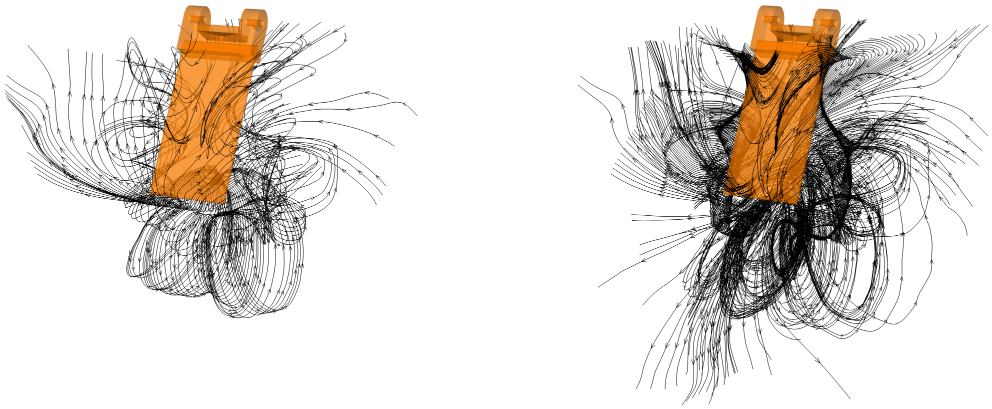
Figure 18.2-9: Instantaneous velocity streamlines for the ramp-down program, frame 1. On the left are the streamlines passing through the bottom edge of the flap; on the right are streamlines passing through the two side edges of the flap.



At some stage approximately halfway through the forward stroke, with the wake now moving faster than the decelerating plate, the flow starts to wrap around the plate from the back to the front (figure 18.2-10). The flows through the sides and the bottom edge are still connected, and the side vortices start to transform into a cork-screw shape.

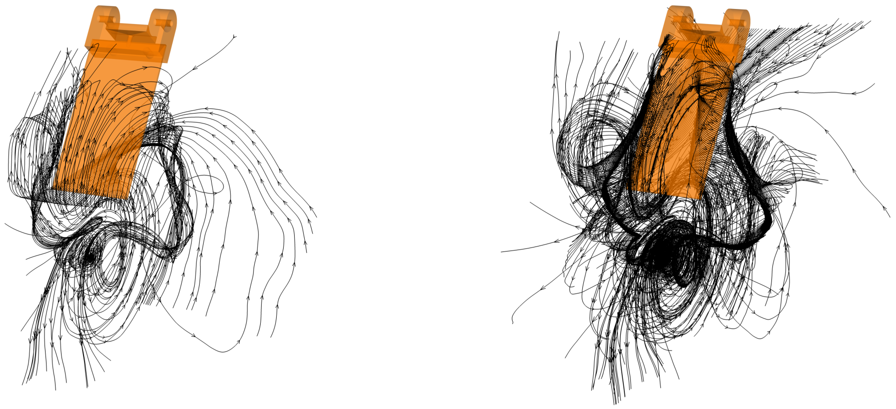
By the fourth frame in the sequence (figure 18.2-11), the flap has all but stopped. The original vortex at the bottom of the plate pushes flow around the bottom edge, which curls up and meets the plate on the forward side. It then splits in two, flowing around the sides and joining the original side vortices, which are now primarily axial flow. This is the beginning of the separation of the flow

Figure 18.2-10: Instantaneous velocity streamlines for the ramp-down program, frame 3 (+246 ms). On the left are the streamlines passing through the bottom edge of the flap; on the right are streamlines passing through the two side edges of the flap.



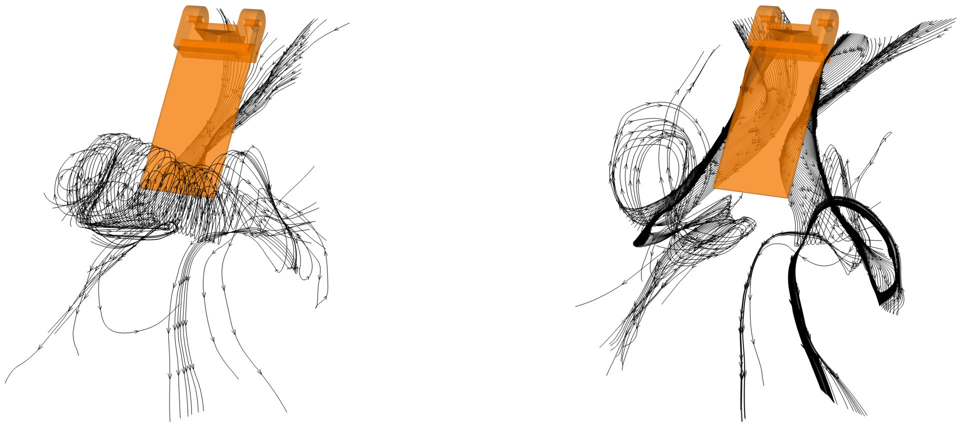
into the flow around the bottom edge and the flow from the sides. Note how the edge flow starts disconnecting from the original vortex.

Figure 18.2-11: Instantaneous velocity streamlines for the ramp-down program, frame 4 (+390 ms). On the left are the streamlines passing through the bottom edge of the flap; on the right are streamlines passing through the two side edges of the flap.



In the fifth frame the separation is obvious (figure 18.2-12). Flow around the bottom edge has formed its own vortex (the one at the top in the slices of figures 18.2-7 and 18.2-8). The original bottom vortex does not seem to be interacting with the flow through the sides, which is pushed away from the corners of the plate. In the next frame the region behind the plate contributing to the top vortex has narrowed further, and the flow around the sides has started to spiral farther from the corners. The picture is incomplete without looking at the vorticity to search for clues as to how the vortical structures are connected.

Figure 18.2-12: Instantaneous velocity streamlines for the ramp-down program, frame 4 (+433 ms). On the left are the streamlines passing through the bottom edge of the flap; on the right are streamlines passing through the two side edges of the flap.



In the velocity data presented above, the streamlines were placed at every millimeter along the perimeter of the flap. This was done arbitrarily based on visual inspection of the flow field (and also because the three-dimensionality of the corners is of interest). For the vorticity data, we place small bundles of streamlines (10 of them in a 2-mm-diameter circle) at the strongest cores of vorticity, found by taking slices of the flow on different planes and identifying the areas of highest out-of-plane vorticity in each slice.

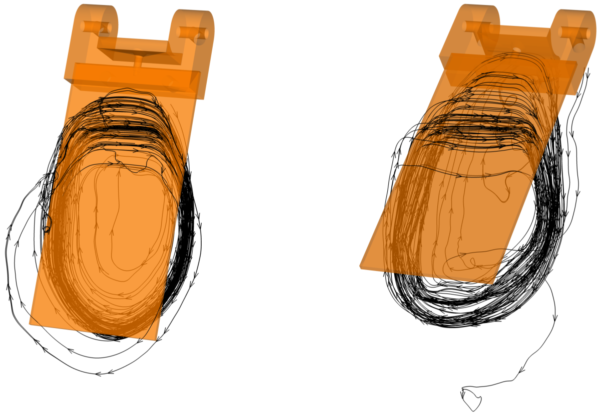
Before delving into the data, it is useful to have a few simple examples in mind to understand the behavior of "vorticity streamlines". If the vorticity field is that of a perfect vortex ring, then any streamline passing through the core will trace a circle of constant diameter, so that if we imagine the streamline as being parametrized with θ, then it will return to itself whenever θ is a multiple of 2π. So a bundle of streamlines placed in a circle to coincide with the vorticity contour of a cross-section of the ring will form a circular ring.

If there is any in-plane vorticity at any cross section of the ring, then a streamline passing through it will diverge off its perfect circle. If the disturbances are noise, then on the whole, the bundle of streamlines will spiral in and out of itself, so that the ring is not represented by a perfect circular ring of streamlines but rather a tightly wound, somewhat erratic spiral. As θ extends toward infinity the spiral will appear more and more dense. This does not indicate anything specifically, but if the spacing between a bundle of streamlines (at the same θ) changes then it can be though of as indicating relative rotational speed of the fluid in the vortex.

If we now imagine a tornado-type vortex, a bundle of streamlines placed at a plane near the "ground" will slowly diverge as they move up toward the "sky" (they will also diverge very close to the ground). Quantitatively, if the bundle is placed at a contour outline at a particular cross section, then the shape of this bundle at other cross sections coincides with the contour on those planes.

In the data presented below, the vorticity field is too noisy to interpret streamline density as relative vortex intensity unless the vortex is extremely strong (as in the case of the vortex generated during the forward stroke). The bundle of streamlines are arbitrarily placed at the strongest cores of Z-vorticity at the mid-plane (coinciding with the plane of rotation) of the flap, so vortex strength is more confidently obtained from iso-surfaces or contours at slices of the field. The splitting and trajectory of the streamlines does however paint a very informative picture of the shape and interaction of multiple vortex structures—something that the magnitude of the vorticity alone can shed no light on.

Figure 18.2-13: Instantaneous vorticity streamlines for the ramp-down program, frames 1 and 2.



As figure 18.2-13 shows, in frames one and two the vortex formed by the flap moving through the fluid is still attached to the top (lower speed) part of the flap. The vortex is strongest at this point. As mentioned before, in frame 3 some of the wake is already started to move around the flap. Figure 18.2-14 suggests that the vortex formed at the bottom edge of the flap at this time loops around and connects to the edge of the main vortex formed during the forward stroke, resulting in an oblong ring vortex surrounding the bottom of the original U-shaped vortex (the bottom of the "U" actually dips through the center of the ring).

In frame 4 (figure 18.2-15), the ring vortex has taken some of the vorticity from the U-shaped vortex as more of the wake spills over the flap. The sides of the U-shaped vortex, staying nearly stationary next to the flap, are convecting the back part of the ring into the center of the flap, pinching it into a kidney shape. In the fifth frame the pinching continues, and the bottom of the "U" starts to twist around as well.

Figure 18.2-14: Instantaneous vorticity streamlines for the ramp-down program, frame 3.

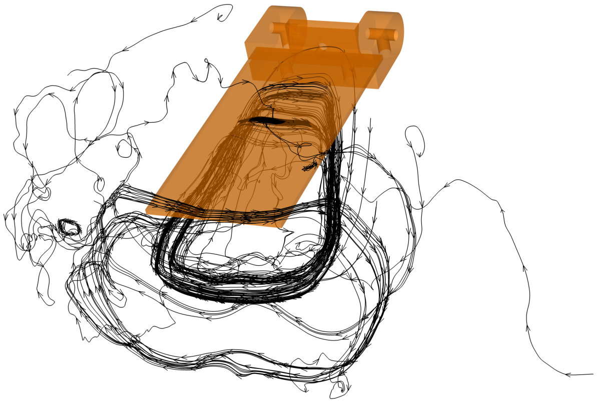
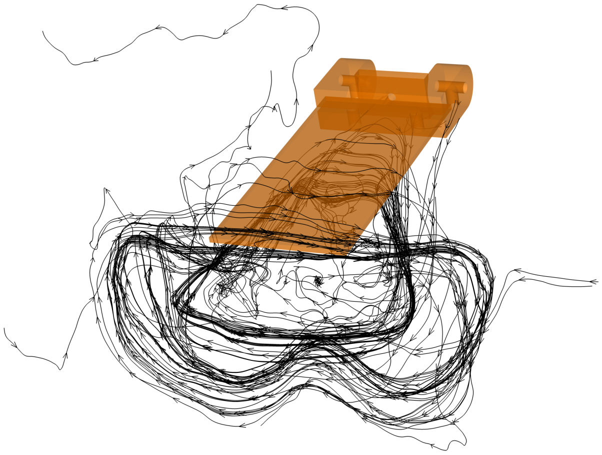
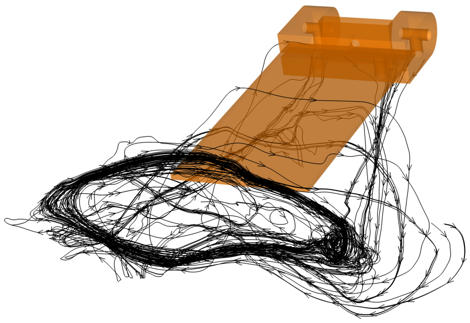


Figure 18.2-15: Instantaneous vorticity streamlines for the ramp-down program, frame 4.



In frame 6 (figure 18.2-16), just before the plate starts moving the other way, the U-vortex and the ring have joined together at the back. The ring connects the two opposing vortices seen in the vertical slices above. This coincides with the flow around the bottom edge and the sides of the flap being the most separate.

Figure 18.2-16: Instantaneous vorticity streamlines for the ramp-down program, frame 6.



In the next frame the flap has started to move back, and the weak starting vortex forming around the flap remains connected to the ring left behind from the forward stroke, forming exactly the structure proposed by Drucker and Lauder [1999]. Since now the flap is accelerating, this vortex quickly grows in strength, and the original vortex continues to convect out of the frame—the new U-shaped vortex becomes much stronger than the left-over ring.

Visualizations of this type would have been impossible without a volumetric measurement. Even if slices had been recorded on different planes (as in Drucker and Lauder [1999], the actual shape of the vortices can only be inferred—if optical access for the necessary slices are even available. Placing the vorticity streamlines based on slices of the vorticity field was educational; once there are more than two cores in the field of view it is not obvious how they are connected—even in this demonstration there were a few surprises.

The type of visualization afforded by DDPIV can only be approximated with carefully executed dye visualizations, and only if recorded in stereo could there be any hope of actually communicating the structures as they were observed. Even then, the data set would not be actually three-dimensional, and could not be viewed from any viewpoint other than the one from which it was recorded—not to mention that the quantitative information in these data sets could be used either as verification of CFD calculations or as input in a sort of "hybrid" method.

Part V

Conclusion

Chapter 19

Summary and Future Work

19.1 Conclusion

DDPIV is the first successful practical implementation of high-density, three-dimensional PIV. The hardware has achieved a turn-key status—the camera is capable of being mounted on a heavy-duty video tripod and minimizes the need for re-calibration. The software is capable of handling enormous data sets, and is fast enough to make such dense measurements practical. The experimental demonstrations presented here should give a clear glimpse at the possibilities available with the DDPIV technique, and the simulations have established numerical expectations for the confidence of the results.

19.2 Future Work

Most of the time has been spent on developing the hardware and the particle cloud reconstruction, which is fundamental to all applications of DDPIV. Still, it is not understood how there can be such a large variation in yield (from particle images to particles) as a function of the seeding particles. The options for calculating velocity are also relatively limited; relaxation tracking still has problems of finding completely impossible matches in quiescent flow which indicates that its implementation has to be looked at more closely. Although some optimization has begun on the algorithms, the point cloud reconstruction needs a performance boost if large data sets are to be the norm.

The simulations presented here used to test the algorithms are nearly exhaustive, but the experimental verifications need to be extended. To date no experiment has been performed under optimal conditions, that is, a proper water tunnel, a tank with optical-grade walls, and carefully constructed standardized experiments with known results. The data generated by DDPIV has only recently had competition in Tomo-PIV, but it can still be partially verified using different, widely accepted instrumentation.

Lastly, the issue of seeding must be solved. As mentioned before the performance with respect to different seeding has not been fully explained, and this is primarily because there is not a large selection of seeding to pick from. Most of the standard seeding used in PIV is too small to image volumetrically unless an extremely powerful light source is available. Relying only on laser light also makes it impossible to ascertain whether the coherence in the light has an effect or not, and the available strobes require seeding even larger than that required by the 200 mJ Nd:YAG laser. A real effort should be made to procure a variety of different particles and attempt to quantify their performance in DDPIV.

Bibliography

R. J. Adrian. Twenty years of particle image velocimetry. *Experiments in Fluids*, 39:159–169, 2005. 4

J. M. Anderson, K. Streitlien, D. S. Barrett, and M. S. Triantafyllou. Oscillating foils of high propulsive efficiency. *Journal of fluid mechanics*, 360:41–72, 1998. 2

S. Barnard and W. Thompson. Disparity analysis of images. *Transactions on Pattern Analysis and Machine Intelligence*, 2:333–340, 1980. 115

D. H. Barnhart, R. J. Adrian, and G. C. Papen. Phase-conjugate holographic system for high-resolution particle image velocimetry. *Applied Optics*, 33:7159–7170, 1994. 5

I. K. Bartol, M. Gharib, D. Weihs, P. W. Webb, J. R. Hove, and M. S. Gordon. Hydrodynamic stability of swimming in ostraciid fishes: role of the carapace in the smooth trunkfish lactophrys triqueter (teleostei: Ostraciidae). *The journal of experimental biology*, 206(4):725, 2003. 0022-0949. 2

J. M. Birch and M. H. Dickinson. Spanwise flow and the attachment of the leading-edge vortex on insect wings. *Nature*, 412(6848):729–733, 2001. 0028-0836. 2, 3

J. M. Birch and M. H. Dickinson. The influence of wing-wake interactions on the production of aerodynamic forces in flapping flight. *The journal of experimental biology*, 206(Pt 13):2257–2272, 2003. 0022-0949. 2

J. M. Birch, W. B. Dickson, and M. H. Dickinson. Force production and flow structure of the leading edge vortex on flapping wings at high and low reynolds numbers. *The journal of experimental biology*, 207(7):1063–1072, 2004. 2

R. Clarke. An application of the poisson distribution. *Journal of the Institute of Actuaries*, 72:481, 1946. 46, 47

D. Dabiri, M. Gharib, D. Jeon, and B. Dooley. Comparison of defocusing dpiv with stereo dpiv. In *APS fluid mechanics division conference*, 2001.

M. H. Dickinson. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284: 1954–1960, 1999. 2

E. G. Drucker and G. V. Lauder. Locomotor forces on a swimming fish: three-dimensional vortex wake dynamics quantified using digital particle image velocimetry. *The journal of experimental biology*, 202(Pt 18):2393–2412, 1999. 0022-0949. 2, 3, 253

G. E. Elsinga, F. Scarano, B. Wieneke, and B. W. van Oudheusden. Tomographic particle image velocimetry. In *6th International Symposium on Particle Image Velocimetry*, 2005. 5, 165

M. Gharib, D. Modarress, D. Dabiri, F. Pereira, and F. Taugwalder. Development and application of a defocusing three dimensional dpiv technique for the mapping of two-phase bubbly flows. In *9th International Symposium on the Application of Laser Techniques to Fluid Mechanics*, 1998a.

M. Gharib, D. Modarress, F. Pereira, D. Dabiri, and F. Taugwalder. Three-dimensional mapping of bubbly flows in tow tanks. In *25th American Towing Tank Conference*, 1998b.

M. Gharib, D. Dabiri, and D. Modarress. Aperture coded camera for three dimensional imaging. *United States Patent Number 6,278,847*, 2001.

M. Gharib, F. Pereira, and E. C. Graff. Applications of DDPIV to studies associated with road vehicles. In R. McCallen, F. Browand, and J. Ross, editors, *The aerodynamics of heavy vehicles: trucks, buses, and trains*, volume 19 of *Lecture notes in applied and computational mechanics*, pages 131–141. Springer, Berlin, 2002.

M. Gharib, F. Pereira, and E. C. Graff. Defocusing 3-D PIV technique and its performance with respect to stereographic systems. In *5th International Symposium on Particle Image Velocimetry*, 2003.

E. C. Graff. *DDPIV System Designer's Manual.* Unpublished, 2007a. 82, 84, 94, 96, 98

E. C. Graff. *Characterization of Defocusing Digital Particle Image Velocimetry Performance, Volume III: Performance of Point Cloud Reconstruction.* Unpublished, 2007b. 120, 144, 166

E. C. Graff. *Characterization of Defocusing Digital Particle Image Velocimetry Performance, Volume II: Performance of Multi-Plane Dewarping.* Unpublished, 2007c. 144

E. C. Graff. *Characterization of Defocusing Digital Particle Image Velocimetry Performance, Volume I: Performance of Gaussian Fitting.* Unpublished, 2007d. 121, 122, 123, 125, 133

E. C. Graff and F. Pereira. *DDPIV System User's Manual.* Unpublished, 2007. 89, 105

E. C. Graff, F. Pereira, and M. Gharib. Applications of defocused digital particle image velocimetry to simultaneous dynamic three-dimensional mapping of solid surfaces and their induced flow. In *APS fluid mechanics division conference*, Seattle, 2004.

E. C. Graff, F. Pereira, M. Milano, and M. Gharib. The capabilities of defocused digital particle image velocimetry. In *6th international symposium on particle image velocimetry*, Pasadena, California, USA, 2005.

E. C. Graff, F. Pereira, and M. Gharib. Visualization of bubbly propeller flow with defocusing digital particle image velocimetry. In *12th International Symposium on Flow Visualization*, 2006.

D. Jeon, F. Pereira, and M. Gharib. Application of DDPIV to bubbly flow measurement. In *11th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, 2002.

D. Jeon, F. Pereira, and M. Gharib. Applications of defocusing DPIV to bubbly flow measurement. *Particle & Particle SYstems Characterization*, 20:193–198, 2003.

L. C. Johansson and R. A. Norberg. Delta-wing function of webbed feet gives hydrodynamic lift for swimming propulsion in birds. *Nature*, 424(6944):65–68, 2003. 0028-0836. 3

L. Kajitani and D. Dabiri. A full three-dimensional characterization of defocusing digital particle image velocimetry. *Measurement science and technology*, 16:790–804, 2005. 25, 44, 172

T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996. 9

D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 3rd Edition*. Addison-Wesley, 1998. 137

G. Labonté. A new neural network for particle-tracking velocimetry. *Experiments in Fluids*, 26: 340–346, 1999. 113

D. Modarress, M. Gharib, D. Dabiri, F. Taugwalder, F. Pereira, and S. Haase. 3-D flow mapping of bubbly flows using defocusing technique. In *ONR Workshop on Free-Surface Turbulence and Bubbly Flows*, 1998.

D. Modarress, M. Gharib, F. Pereira, D. Dabiri, and F. Taugwlader. 3-D flow diagnostics for bubbly two-phase flows. In *ONR Workshop on Free-Surface Turbulence and Bubbly Flows*, 1999.

F. Noca, D. G. Shiels, and D. Jeon. Measuring instantaneous fluid dynamic forces on bodies, using only velocity fields and their derivatives. *Journal of fluids and structures*, 11:345–350, 1997. 3

R. J. Palum. Optical blur filter having a four-feature pattern. *United States Patent Number 6,326,998*, 2001. 53

F. Pereira and M. Gharib. Defocusing digital particle image velocimetry and the three-dimensional characterization of two-phase flows. *Measurement Science and Technology*, 5(13):683–694, 2002. 23, 25, 26, 35, 38, 111

F. Pereira and M. Gharib. A method for three-dimensional particle sizing in two-phase flows. *Measurement Science and Technology*, 15(10):2029–2038, 2004. 111

F. Pereira, M. Gharib, D. Dabiri, and D. Modarress. An introductory application of defocusing dpiv to the study of bubbly shear flows. In *APS fluid mechanics division conference*, 1999a.

F. Pereira, M. Gharib, D. Modarress, and D. Dabiri. Defocusing DPIV: A 3-component 3-D DPIV measurement technique. application to bubbly flows. In *3rd International Symposium on Particle Image Velocimetry*, 1999b. 22

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(7):S78–S84, 2000a. 0723-4864. 23, 66

F. Pereira, M. Gharib, D. Dabiri, and D. Modarress. Instantaneous whole field measurement of velocity and size of air microbubbles in two-phase flows using DDPIV. In *10th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, 2000b.

F. Pereira, M. Gharib, D. Dabiri, and D. Modarress. Size and velocity of air bubbles around a model propeller. In *ONR Workshop on Free-Surface Turbulence and Bubbly Flows*, 2000c.

F. Pereira, M. Gharib, D. Modarress, and D. Dabiri. Implementation of defocusing DPIV and application to the bubbly flow around a propeller. In *9th International Symposium on Flow Visualization*, 2000d.

F. Pereira, E. C. Graff, and M. Gharib. Bubble interaction with a propeller flow. In *26th ONR Symposium on Naval Hydrodynamics*, 2006a.

F. Pereira, D. Modarress, M. Gharib, D. Dabiri, and D. Jeon. Aperture coded camera for three dimensional imaging. *United States Patent Number 7,006,132*, 2006b.

F. Pereira, H. Stüer, E. C. Graff, and M. Gharib. Two-frame 3D particle tracking. *Measurement Science and Technology*, 17(7):1680–1692, 2006c. 69, 113, 114, 115, 204, 205, 206

F. Pereira, J. Lu, E. C. Graff, and M. Gharib. Microscale 3d flow mapping with µDDPIV. *Experiments in Fluids*, 2007.

A. K. Prasad and K. Jensen. Scheimpflug stereocamera for particle image velocimetry in liquid flows. *Applied Optics*, 34:7092–7099, 1995. 4

M. J. Ringuette. *Vortex formation and drag on low aspect ratio, normal flat plates*. PhD thesis, California Institute of Technology, 2004. 3

P. G. Saffman. *Vortex dynamics*. Cambridge University Press, Cambridge, 1992. 2

R. Tsai. A versatile camera calibration t echnique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE journal of robotics and automation*, 3:323–344, 1987. 5, 7

J. J. Tuma and R. A. Walsh. *Engineering Mathematics Handbook*. McGraw-Hill Professional, New York, 4 edition, 1997. 25, 28

Z. J. Wang, J. M. J. M. Birch, and M. H. M. H. Dickinson. Unsteady forces and flows in low reynolds number hovering flight: two-dimensional computations vs robotic wing experiments. *The journal of experimental biology*, 207(3):449, 2004. 0022-0949. 2

C. E. Willert and M. Gharib. Digial particle image velocimetry. *Experiments in Fluids*, 10: 181–193, 1991. 4

C. E. Willert and M. Gharib. Three-dimensional particle imaging with a single camera. *Experiments in fluids*, 12:353–358, 1992. xxii, 18, 25, 60, 62, 63