5-2012

Autocorrelation-Based Estimate of Particle Image Density in Particle Image Velocimetry

Scott O. Warner
Utah State University

AUTOCORRELATION-BASED ESTIMATE OF PARTICLE IMAGE DENSITY IN PARTICLE IMAGE VELOCIMETRY

by

Scott O. Warner

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

Dr. Barton L. Smith Major Professor	Dr. Heng Ban Committee Member
Dr. Robert E. Spall Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2012

Abstract

Autocorrelation-Based Estimate of Particle Image Density in Particle Image Velocimetry

by

Scott O. Warner, Master of Science

Utah State University, 2012

Major Professor: Dr. Barton L. Smith
Department: Mechanical and Aerospace Engineering

In Particle Image Velocimetry (PIV), the number of particle images per interrogation region, or particle image density, impacts the strength of the correlation and, as a result, the number of valid vectors and the measurement uncertainty. Therefore, any a-priori estimate of the accuracy and uncertainty of PIV requires knowledge of the particle image density. An autocorrelation-based method for estimating the local, instantaneous, particle image density is presented. Synthetic images were used to develop an empirical relationship based on how the autocorrelation peak magnitude varies with particle image density, particle image diameter, illumination intensity, interrogation region size, and background noise.

This relationship was then tested using images from two experimental setups with different seeding densities and flow media. The experimental results were compared to image densities obtained through using a local maximum method as well as manual particle counts and are found to be robust. The effect of varying particle image intensities was also investigated and is found to affect the particle image density.

(89 pages)

Public Abstract

Autocorrelation-Based Estimate of Particle Image Density in Particle Image Velocimetry

by

Scott O. Warner, Master of Science

Utah State University, 2012

Major Professor: Dr. Barton L. Smith
Department: Mechanical and Aerospace Engineering

Particle Image Velocimetry (PIV) is an optical method for measuring the speed of liquids and gases. As part of PIV, the flow is seeded with small particles. Images of the particles are captured at time intervals and the movement of the particles between images is used to calculate the speed and direction of the fluid flow. The ability of PIV to accurately measure the flow velocity is a function of many parameters including the particle image density, or number of particles contained within an image. Therefore, a knowledge of the particle image density can be used to estimate the accuracy and uncertainty of the PIV measurements.

A method for estimating the particle image density is presented. With the use of synthetic (computer generated) images, a formula was developed that relates the particle image density to a function called an autocorrelation, as well as the particle image diameter, average particle intensity, and the interrogation region size. The relationship was then tested on PIV images from two experimental setups. Particle image density estimates from the experimental images were compared to results acquired by using a local maximum method as well as manual particle counts and are found to be robust. The effect of varying the particle image intensity was also investigated and found to affect the particle image density.

To my wife, Rachel, who supported me each step of the way.

Acknowledgments

I would like to acknowledge my friends and colleagues at the Experimental Fluid Dynamics Laboratory, particularly Dr. Barton Smith for the guidance and encouragement he has offered throughout this process.

I would like to express gratitude to the Nuclear Science & Technology Directorate at Idaho National Laboratory for their support of this research effort. The work was supported through the U.S. Department of Energy, Laboratory Directed Research & Development grant under DOE Contract 122440 (Project Number: 12-045).

I would also like to acknowledge my dear wife, Rachel, for her patience and support.

Scott O. Warner

Contents

List of Tables

List of Figures

Notation

Roman symbols

a	gradient parameter
A	interrogation domain area
D_a	aperture diameter
d_D	correlation peak width
D_I	interrogation region size
d_τ	particle image diameter
d_p	actual particle diameter
d_s	diffraction-limited spot diameter
f	focal length
$f^{\#}$	ratio of f to D_a
IA_1	interrogation area
I_{max}	minimum image intensity
I_{min}	maximum image intensity threshold
$\overline{I_p}$	average intensity of the pixel centered particles
$\mathcal{J}$	laser intensity
M	image magnification
n	integer value
N	number of image density measurements
N_I	particle image density
$N_{I,true}$	true particle image density of the synthetic images
R_h	relative autocorrelation peak height
R_{min}	minimum value of the autocorrelation
R_p	highest autocorrelation peak height
$R(r,s)$	discrete autocorrelation

Greek symbols

ϵ	absolute error
η	relative error
λ	light sheet wavelength
μ_p	mean particle intensity
σ_p	standard deviation of the particle intensities
σ	standard deviation of the Gaussian fit

Acronyms

ABD	Autocorrelation-Based Density
CFD	Computational Fluid Dynamics
DWO	discrete window offsets
EM	Empricial Density
FFT	Fast Fourier Transform
LM	Local maximum
MC	Manual Count
PIV	Particle Image Velocimetry

Chapter 1

Introduction

Particle image velocimetry (PIV) is a non-intrusive measurement technique used in experimental fluid mechanics to acquire spatially resolved velocity fields. PIV is capable of producing two- or three-component velocity fields which other flow measurement techniques, such as hot wire anemometry and acoustic Doppler velocimetry, cannot. PIV is applied to a wide variety of flow problems, ranging from measuring flows in turbomachinery [1] to determining Reynolds stresses in artificial heart valves [2], and is often used as a validation technique for numerical simulations, including Computational Fluid Dynamics (CFD).

As with any measurement technique, it is important to be aware of the parameters that contribute to the measurement uncertainty. Being able to estimate these values allows for a better understanding of how a given parameter effects the measurement and provides a means of quantifying the uncertainty [3]. The remainder of this chapter discusses the fundamentals of PIV and the parameters that contribute to its uncertainty.

1.1 Particle Image Velocimetry

The method of PIV involves introducing neutrally buoyant seeding particles in a flow field and illuminating them with a double-pulsed laser sheet. Image-pairs of the illuminated particles are acquired at discrete time intervals using a high-speed CCD camera. The images are then subdivided into interrogation regions and processed using a cross-correlation-based algorithm to obtain the associated particle displacements and, thus, velocity fields. A diagram of the PIV method is shown in Fig. 1.1. Articles by Prasad [4], Adrian [5], and Westerweel [6] provide a comprehensive description of the technique.

Since its beginning, PIV has undergone substantial improvements in measurement accuracy and reliability [8]. The early technique of analog film recording and evaluation have

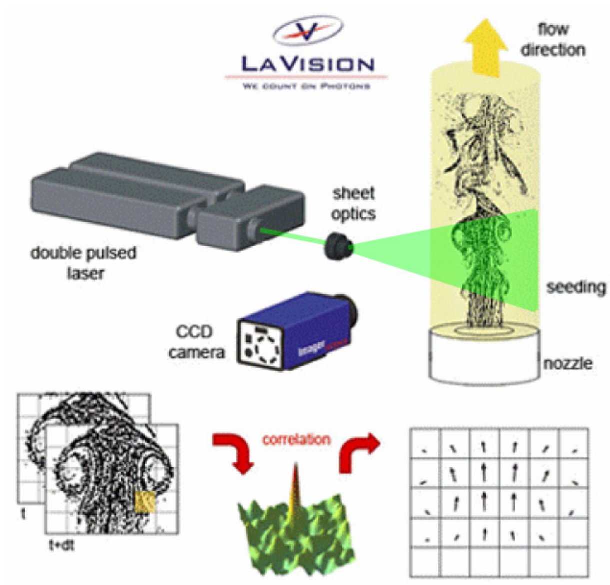


Fig. 1.1: A Particle Image Velocimetry System [7]. A double-pulsed laser illuminates seeding particles that are distributed in a flow. A CCD camera captures images of the seeding particles separated by discrete time intervals. The images are then subdivided into interrogation regions and processed using a cross-correlation-based algorithm to produce a velocity vector field.

been replaced with digital camera techniques. The introduction of sub-pixel interpolation resulted in accuracy values reported as low as 0.1 pixel units [9]. The use of adaptive evaluation algorithms, such as discrete window offsets (DWO) [10] and iterative image deformation methods [11] have been shown to reduce uncertainty levels. A comprehensive report of the historical development of PIV is provided by Adrian [8].

1.2 Uncertainty in PIV

The uncertainty in PIV measurements is a function of many parameters [12]. These parameters include, but are not limited to, particle image diameter, displacement gradients, sub-pixel displacements, and particle image density. Each of these parameters are discussed in the following sections.

1.2.1 Particle Image Diameter

The particle image diameter, d_τ, is the diameter (in pixel units) of the particle as it appears in the recording medium and is proportional to the width of the correlation peak [13,14]. The correlation peak can be generated by either implementing a cross-correlation or an autocorrelation. The cross-correlation method involves correlating two separate images together, while an autocorrelation involves correlating a single image with itself. Many different values have been suggested for optimum particle image diameter ranging from just under 2 pixel units [15] to over 6 pixel units [16]. In general, the optimum particle image diameter ranges from 2 to 4 pixel units [17]. The variation in optimum diameter stems from the method used to estimate the sub-pixel displacements (e.g. three-point Gaussian, three-point parabolic, and centroid fits).

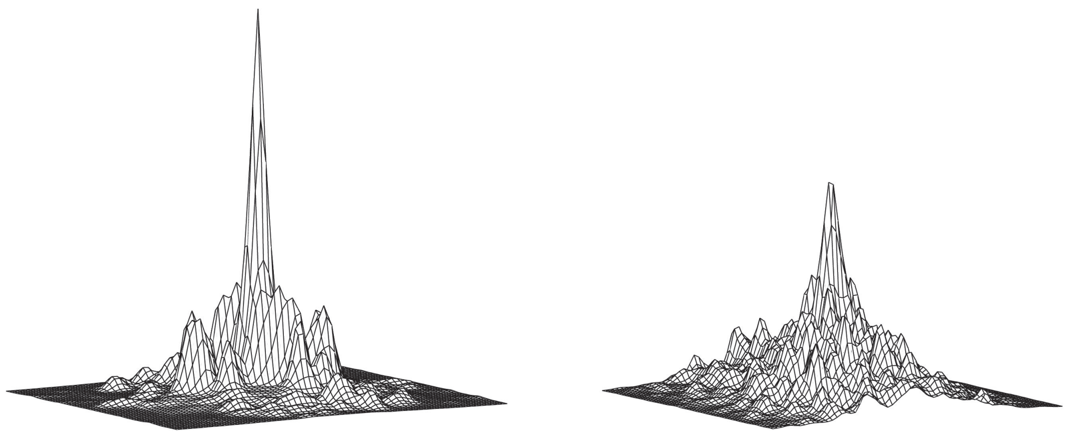
When particle images become too small (≤ 1 pixel) the uncertainty can increase due to "peak locking," or pixel locking. Peak locking results in displacements that are biased toward integer pixel values [12]. The presence of peak locking can be detected by making a histogram plot of PIV displacements. If peak locking is present, the histogram will have distinct peaks at integer displacement values. Image preprocessing can reduce the peak locking effect.

The uncertainty also increases when particles become too large (for a fixed camera resolution and 3-point sub-pixel estimation method). As the particle image diameters increase, the correlation peak becomes more wide, and the center of the correlation peak (used to estimate sub-pixel displacement) becomes more difficult to locate [17].

1.2.2 Displacement Gradient

Displacement gradients, or flow velocity gradients, diminish the amplitude of the correlation peak and broaden its width. As the correlation peak decreases and broadens, the peak becomes less detectable, which results in increased uncertainty. The effect of the displacement gradient on the correlation peak can be seen in Fig. 1.2, where Fig. 1.2a and Fig. 1.2b are cross-correlation maps obtained from synthetic images with no displacement gradient and a 0.2 pixels/pixel displacement gradient, respectively.

The displacement gradient can be obtained using various techniques [18], including finite difference methods [19], biquadratic polynomial with least-squares interpolation [20],



(a) (b)

Fig. 1.2: A demonstration of the effects of displacement gradients on the correlation peak. The correlation of synthetic images with constant parameters show the difference between applying (a) no gradient and (b) a 0.2 pixels/pixel gradient.

and radial basis functions [21, 22].

1.2.3 Sub-Pixel Interpolation

Originally, the resolution of PIV measurements was limited to integer pixel values. To improve the resolution and accuracy, sub-pixel interpolation methods are used in PIV. These methods fit a curve to the correlation peak profile and estimate the location of the correlation peak.

Many methods exist for estimating the sub-pixel displacement, including the peak centroid method [23], Gaussian interpolation [9], sinc interpolation [24], and polynomial interpolation [25]. It has been suggested that the optimal sub-pixel fit method may be a function of the particle image diameter [26]. The uncertainty of the sub-pixel displacement depends on the interpolation method used and how well it fits the correlation peak profile.

1.2.4 Particle Image Density

Another important parameter that effects the PIV measurement uncertainty is the particle image density, N_I. The particle image density is the mean number of particles per interrogation region. It is recommended to have N_I greater than 10 particles [27]. The PIV measurement uncertainty is impacted by N_I in at least two ways:

1. Increasing the number of particle image pairs within an interrogation window increases the probability of a valid displacement vector [28] and decreases the measurement uncertainty [12, 29].

2. In regions of shear, small N_I can result in increased random uncertainty [3].

Due to its influence on uncertainty, it is important to know the local instantaneous particle image density. The particle image density may vary in space and time to due to variations in seeding levels or illumination issues. An estimation of the particle image density can provide the ability to rapidly scan a PIV dataset for these issues, determine the quality of a PIV dataset, and eliminate bad images prior to vector computation.

Few methods for estimating N_I are available. For sufficiently low N_I, one may attempt to simply count the number of particle images by hand. However, the task is time consuming and subject to user bias. It is also difficult to account for dim and overlapping particles within an interrogation domain.

Another method of determining N_I involves using a local maximum routine to find particles. A threshold is needed to separate low level peaks generated by noise and those created by actual particles. This method is also unable to account for overlapping particles and tends to underestimate N_I as the particle image density increases.

Previously, as part of a method for estimating the instantaneous local uncertainty, an estimate of the particle image density N_I has been made by applying a binary threshold to an interrogation domain, summing the binary values of the image, and then dividing by the approximate pixel area of a single particle [3]. This method was applied to computer generated PIV images called "synthetic images." This density estimate is not robust in that it requires a correction factor and threshold value unique to each image set and it also cannot account for overlapping particles.

The autocorrelation peak magnitude is proportional to the particle image density, particle image diameter, average particle image intensity, and the size of the interrogation domain. With the use of synthetic images and estimates of the particle image diameter and average particle image intensity, an empirical relationship between these parameters was determined; and a method to estimate the particle image density was developed.

The next chapter lists the objectives of this research. In chapter 3, the methods for calculating the autocorrelation peak magnitude, particle image diameter, and average particle intensity are discussed along with preprocessing techniques. Chapter 4 covers synthetic image generation and both experimental setups. Chapter 5 discusses the development of the empirical equation used in the autocorrelation-based density (ABD) method followed by the results of the ABD method on synthetic images with noise as well as images from the two experimental setups.

Chapter 2

Objectives

The objectives of the research are as follows:

- Develop a method for estimating the average particle intensity to be used in the autocorrelation-based density (ABD) method.
- Generate synthetic images that contain various particle image diameters, densities, and intensities.
- Develop an empirical relationship relating the particle image density to the correlation peak height, particle image diameter, average intensity, and interrogation region size.
- Determine the effect of noise on method by introducing synthetic noise derived from low speed and high speed PIV cameras.
- Determine the effect of particle image intensity on the particle image density estimation by varying the lens aperture.
- Estimate the particle image density of the experimental data using the ABD method and compare results those obtained by the local-maximum method and manual particle counting.

Chapter 3

Approach

The relative height of the autocorrelation peak R_h for an interrogation region in a PIV image is a function of particle image diameter d_τ, particle image density N_I, interrogation domain area A, and average particle intensity $\overline{I_p}$. As each of these parameters increase, so does the height of the autocorrelation peak (see Fig. 3.1). In other words,

$$R_h = f(d_\tau, N_I, A, \overline{I_p}). \tag{3.1}$$

A method is presented to extract the particle image density from the autocorrelation peak height by quantifying the effects from the other contributing parameters. Before the density estimation takes place, noise due to the image background is removed and the images are normalized. Each of the parameters in Eq. 3.1 are discussed in detail, with exception to A, as its calculation is trivial, followed by the pre-processing techniques used.

3.1 Relative Autocorrelation Peak Height

An autocorrelation map for a single interrogation area IA_1 can be generated by computing the discrete autocorrelation

$$R\,(r,s) = \sum_{i=0}^{D_I-1} \sum_{j=0}^{D_I-1} IA_1\,(i,j)\, IA_1\,(i+r, j+s) \tag{3.2}$$

where D_I is the interrogation cell size and $r, s = -D_I/2, ..., D_I/2 - 1$ [30].

Alternatively, a frequency domain based correlation may be used in place of Eq. 3.2 by applying the Wiener-Khinchin theorem [12]. Using this theorem, the autocorrelation is

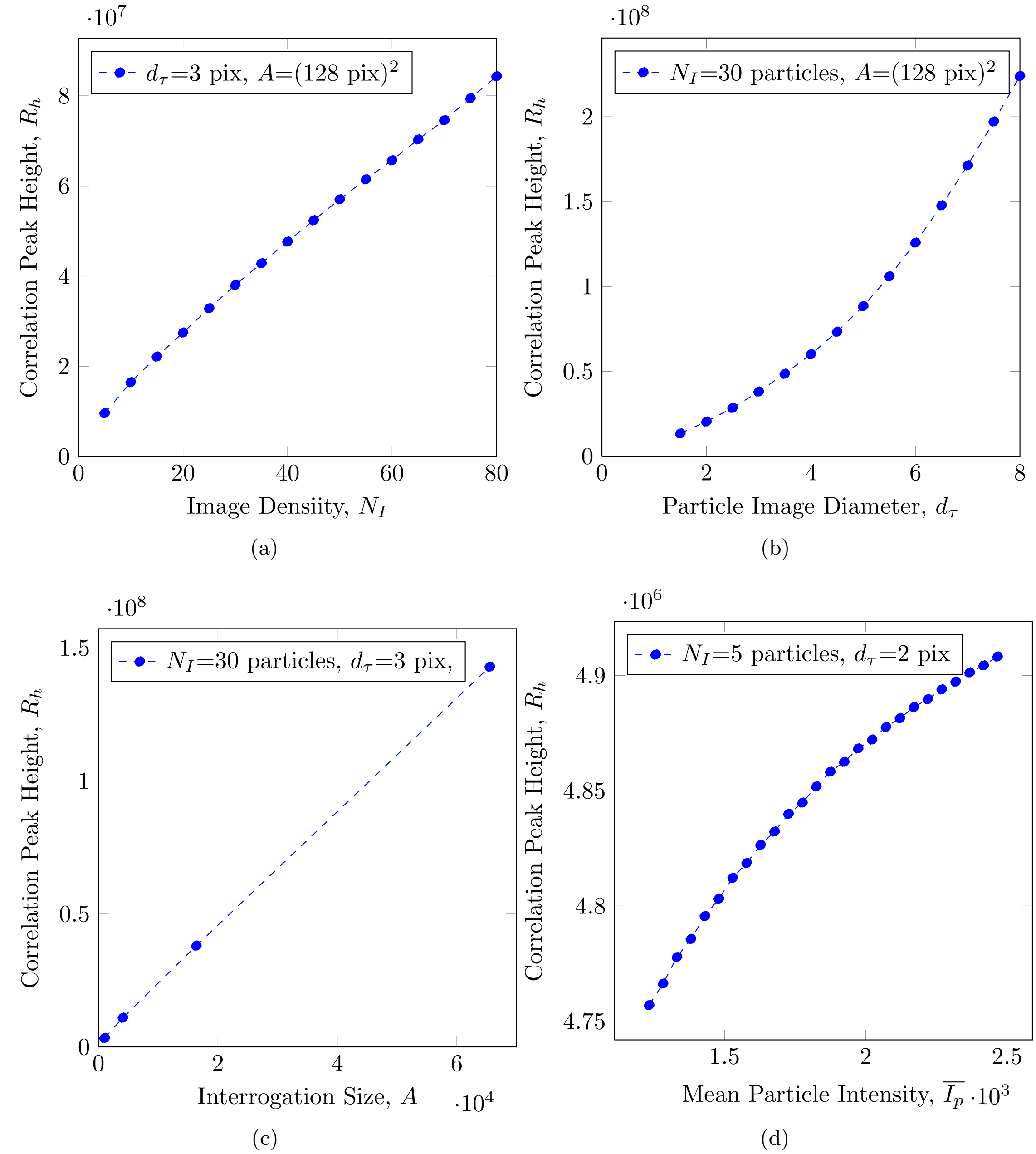


Fig. 3.1: The relative height of the autocorrelation peak R_h as a function of (a) N_I, (b) d_τ, (c) A, and (d) I_P. These figures were generated by autocorrelating synthetic images with known particle image diameters, densities, and intensities.

computed with Fourier transforms as

$$R(r, s) = \mathrm{Re}\left[FFT^{-1}\left\{FFT^{*}\left(IA_1\right) FFT\left(IA_1\right)\right\}\right] \tag{3.3}$$

where FFT denotes a Fast Fourier Transform, * denotes the complex conjugate and the Re operator returns the real part of the complex number [12]. The use of FFTs in Eq. 3.3 to compute the autocorrelation requires less computation time compared to the discrete method in Eq. 3.2 [30]. In order to use Eq. 3.3, the interrogation size must be $D_I = 2^n$, where n is an integer value.

As a result of using Eq. 3.3 to correlate IA_1 with itself, an autocorrelation map is generated such as shown in Fig. 3.2a. The relative peak height R_h is defined as

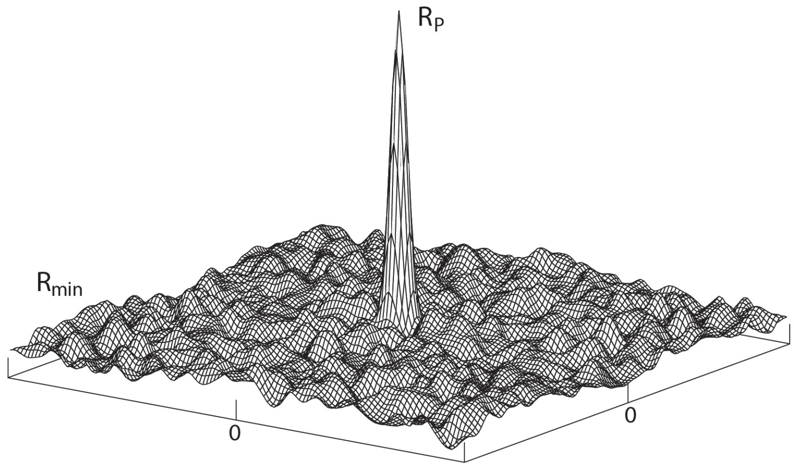
$$R_h = R_p - R_{min}. \tag{3.4}$$

where R_p is the height of the highest correlation peak and R_{min} is the lowest value in the correlation plane (Fig. 3.2b). R_h is calculated relative to the correlation background to help copmpensate for a general image background level or background noise.

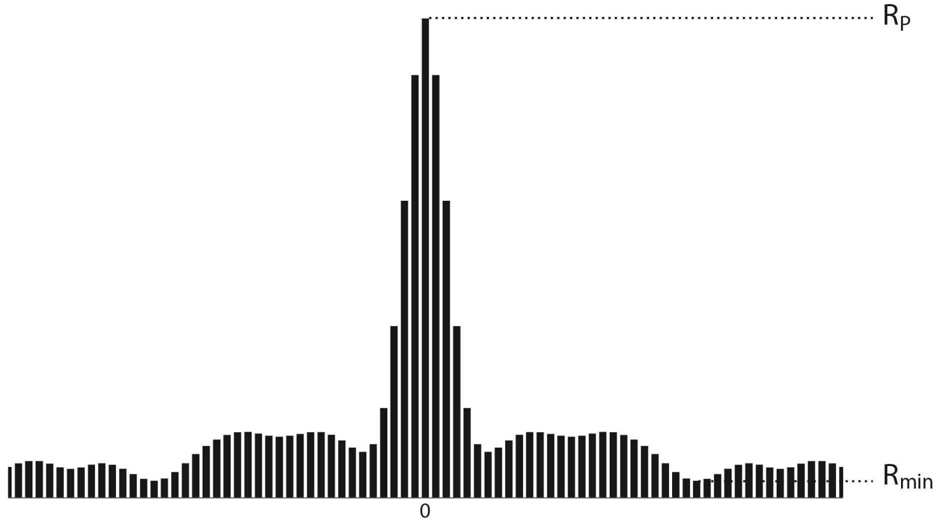
3.2 Average particle intensity

Since the autocorrelation peak magnitude depends on the average particle image intensity, a means to quantify this value based on the images is required. The average particle intensity is calculated by first locating obvious particles using a function that locates local maximum values in the image. Since the pixel intensity is an average of the light intensity incident on the pixel, the maximum intensity of a particle is highest and closest to the true maximum intensity when the particle is centered on a pixel. Therefore, the average particle intensity estimate was based on the particle images that were well aligned with the pixel grid. Figure 3.3 shows the difference in the particle image intensity for an off-center particle image and the preferred pixel-centered particle.

To determine if a particle (represented by a local maximum) is pixel-centered, the standard deviation of the intensity of the four adjacent pixels is computed. If the standard deviation is beneath a specified threshold value, it is deemed pixel-centered and contributes to the average intensity calculation. All of the values of the pixel-centered particles are averaged together to estimate the average particle intensity that contributes to the particle



(a)



(b)

Fig. 3.2: Preforming an autocorrelation using Eq. 3.3 produces (a) an autocorrelation map. A cross section of the autocorrelation map (b) shows the highest and lowest value of the autocorrelation map, R_P and R_{min} repectively. The difference between these values is the relative autocorrelation peak height.

image density calculation.

The average particle intensity estimation method is based solely on local maximum intensities and not the average intensity of the actual individual particles. When two or

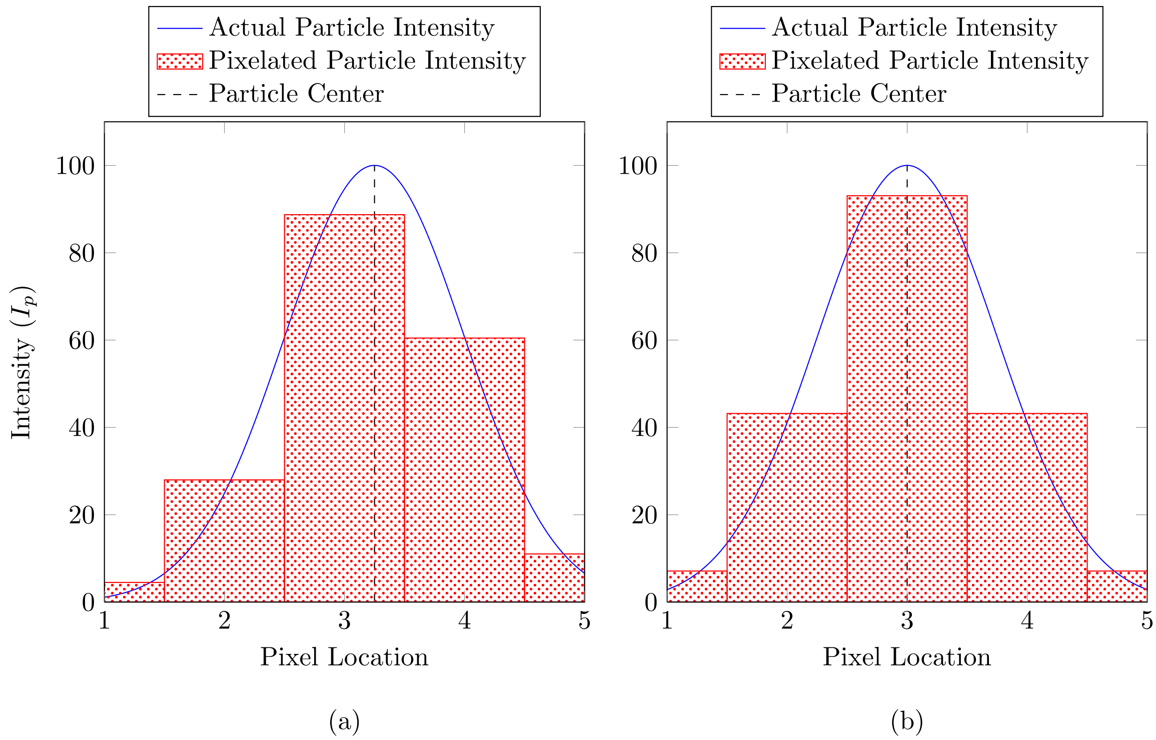


Fig. 3.3: The maximum particle intensity depends on the sub-pixel location of the particle. A particle with a 0.25 pixel offset (a) has a reduced maximum pixel intensity and a larger standard deviation in surrounding pixels. A particle that is pixel centered (b) provides the best estimate of the actual maximum particle intensity. If a particle is pixel centered, the four adjacent pixels will have nearly identical intensity values and a low standard deviation.

more particles overlap, the intensities of the particles are summed together, which will increase the average particle intensity estimate. As a results the average particle intensity estimate will tend to overestimate the true average particle intensity as the density increases.

3.3 Particle Image Diameter Estimation

The particle image diameter d_τ and the width of the displacement-correlation peak d_D are related by

$$d_D \cong \sqrt{2d_\tau^2 + \frac{4}{3}a^2}, \tag{3.5}$$

where the gradient parameter a can be neglected when d_D is obtained through autocorrelation [17]. The autocorrelation peak width is commonly calculated using the e^{-2} width, which is four times the standard deviation for a Gaussian distribution. In order to find the standard deviation from the autocorrelation peak, a 3-point Gaussian fit [12] is applied to a cross section of the peak (Fig. 3.4).

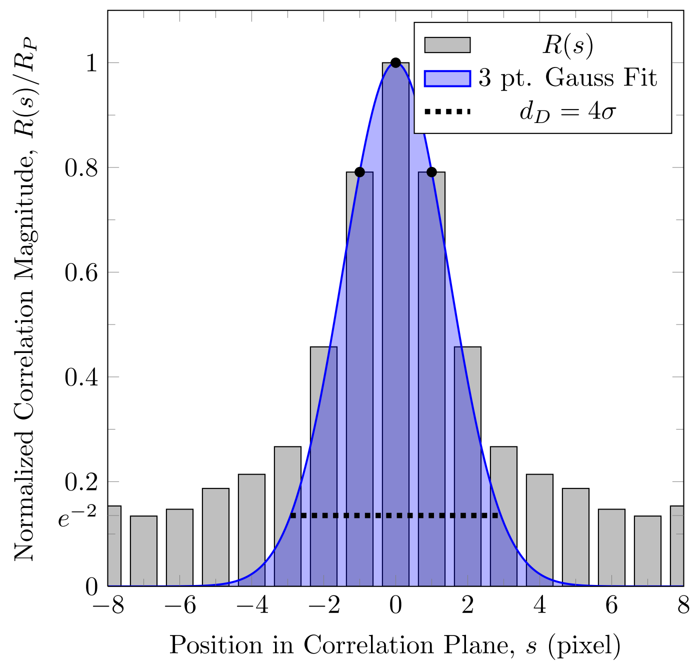


Fig. 3.4: A 3-point Gaussian is applied to the cross section of the correlation peak using the three largest values.

Having obtained the standard deviation, and thereby the autocorrelation peak width, equation 3.5 is solved directly for the particle image diameter

$$d_\tau \cong 2\sqrt{2}\sigma, \tag{3.6}$$

where σ is the standard deviation of the Gaussian fit.

3.4 Particle Image Density Estimation

Under favorable conditions, the particle image density can be approximated by counting particles within an interrogation region. As previously mentioned, the task is time consuming and subject to user bias. Also, it is difficult to estimate the number of particles when particles have low intensity or are overlapping.

The counting method for estimating the particle image density can also be automated by locating and counting local maxima above a specified threshold intensity. This is done by comparing the intensity of each pixel to the intensity of its 8 nearest neighbors. If the center pixel intensity is larger than its neighboring pixels, then it is deemed a local maximum and, therefore, contributes to the particle image density.

This local maximum method is able to provide somewhat accurate results when particle image diameter and density are low. As the particle image diameter and density increase, more particle images overlap, which decreases the number of local maximums and results in an under estimate of the particle image density. To demonstrate this trend, synthetic images were generated with known particle image diameters and image densities, the densities were estimated using the local maximum method. The results are shown in Fig. 3.5. As the density and diameter increase, the estimated value of the density decreases and the error increases.

3.5 Image Preprocessing

One advantage of using digitally acquired images is the ability to remove non-ideal aspects. Under ideal circumstances images would contain brightly and uniformly illuminated particles against a perfectly dark background. However, this idealized scenario is rarely achieved in the experimental world. Generally, the image background is not perfectly dark due to background noise and particles may vary in intensity or have low contrast. This section will discuss methods for removing background noise and enhancing image contrast.

3.5.1 Background Image Removal

Background noise can limit the ability to estimate the particle image diameter and particle image density. In some situations, noise may also cause error in the PIV measurements. Background noise results from many things, including the zero-level noise of the camera sensor, environmental lighting, non-uniform lighting, and laser reflections from

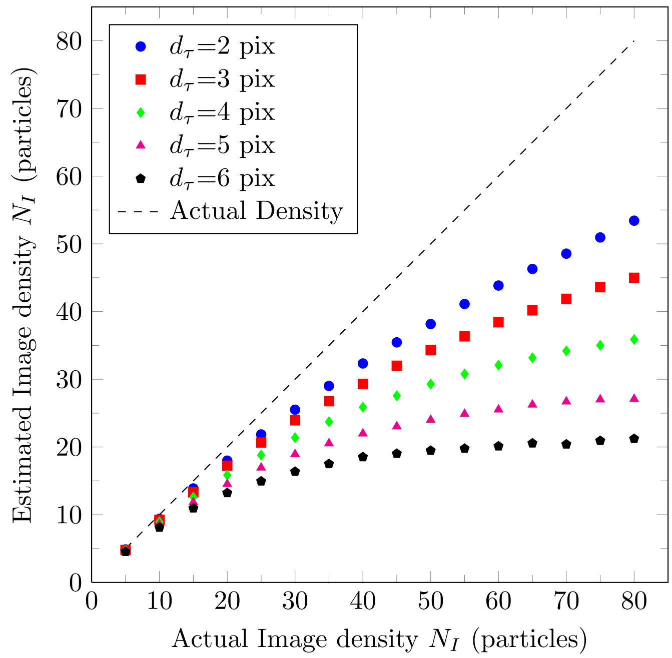


Fig. 3.5: Particle image density estimation using a the local maximum method to locate and count local maximums (particles) in synthetic images with known particle image diameters and image densities.

stationary objects in the flow [31]. Noise from such sources should be avoided; however, this is not always possible. Alternatively, preprocessing can be use to remove non-ideal aspects from the images and is generally beneficial [32, 33]. One technique used to remove noise is to subtract a background image from the original image.

Some of the most common methods of determining the background image include:

1. Recording an image of an illuminated flow without tracer particles.

2. Using the average or local minimum of all images within the image set [34, 35].

3. Applying a low-pass/uniform or a median filter [17].

In the present work, the local minimum method was used to generate the background image. The entire image set was analyzed, and the minimum value for each pixel location was used for the background [26]. Fig. 3.6 shows an original image, the calculated background using this approach, and the image with the background subtracted. Without the use of background removal, stationary objects and ambient lighting contribute to the density estimate, resulting in significantly inflated results.

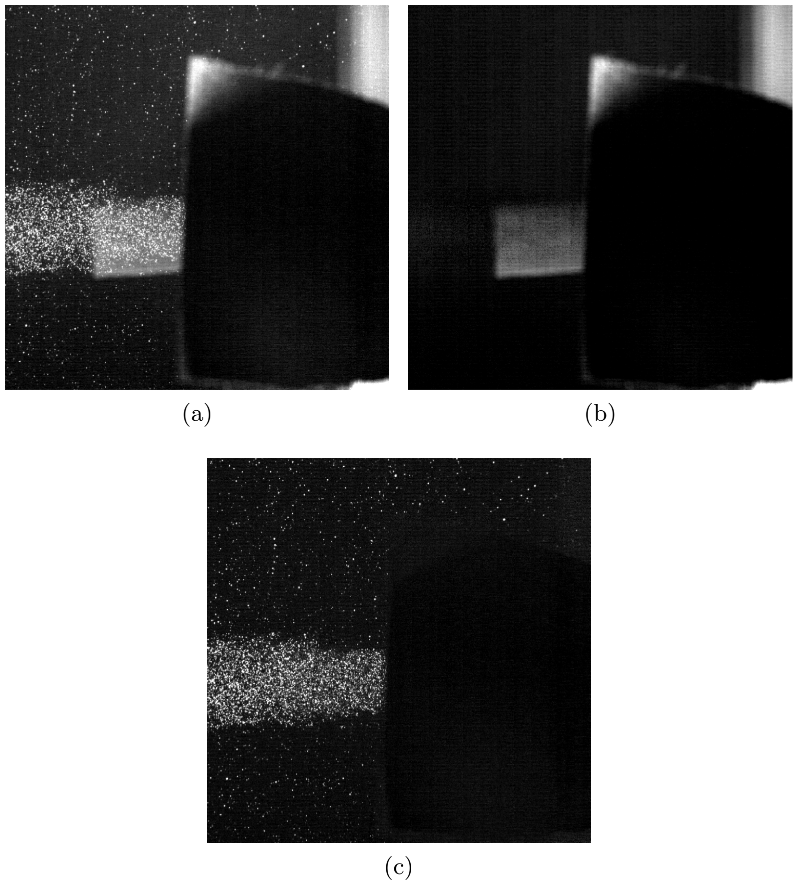


Fig. 3.6: Demonstration of the impact of background removal: (a) the original image (b) the background image based on the local minimum value from each pixel, and (c) the processed image.

3.5.2 Image Normalization

The range of pixel intensities within an image set may vary due to the laser intensity, lens focal number, and bit depth of the camera sensor. By adjusting these parameters, particle intensities can range from values below the noise floor to the saturation threshold. Local variations in intensity can occur due to variations in the size of the tracer particles, nonuniform lighting, and reflections [17]. Preconditioning methods are available to optimize the contrast levels and provide uniform intensity across the full image. Some of these methods include histogram equalization and min-max filtering [17].

The presence of unequal illumination between image sets, and interrogation regions within those sets, can result in inaccurate estimates of the density. Therefore, the images are normalized to ensure a consistent range in pixel intensities. Normalization occurs by subtracting off the smallest intensity value I_{min} in the image and then dividing by I_{max}, where

$$I_{max} = \mu_p + 4\sigma_p, \tag{3.7}$$

and where μ_p and σ_p are the mean and standard deviation of the particle intensities respectively. Any pixel intensity greater than I_{max} is set to I_{max} in order to remove the effects of particles with intensities far from the mean. As a result, images from different cameras with different bit-depths, appear more similar and provide more consistent density estimates.

Chapter 4

Synthetic Image Generation and Experimental Setups

The autocorrelation-based density method was developed using synthetic images with known particle image diameters, image densities, particle image intensities, and interrogation region sizes. The method was then verified using images from two experimental setups. This chapter discusses the synthetic image generator and the experimental setups used to supply images to develop and verify the autocorrelation-based density method.

4.1 Synthetic Image Generation

The particle image density estimation method that has been developed employs an empirical relationship between the autocorrelation peak height and the other parameters previously discussed to estimate the particle image density. The relationship was developed using synthetic data with known particle image diameter, density, and intensity. The synthetic image generator that was used is described by Timmins *et al.* [3]. The simulated particles are randomly distributed throughout the interrogation domain area and within the width of the light sheet. The intensity distributions of the particles are represented by Gaussian functions and the particle image diameter is four times the standard deviation of the Gaussian distribution. The laser sheet intensity distribution is Gaussian and the maximum intensity of any given particle is determined by the particle's position within the light sheet. Flow properties, such as displacement and shear, have no effect on the autocorrelation and are not accounted for.

In order to determine the effect of noise on the density estimation, two levels of background noise were added to the synthetic images to simulate the noise generated by the cameras used in acquiring PIV measurements. To approximate the background noise, 100 images pairs were acquired using two different cameras with the lens caps on. The cameras that were used are the PCO Sensicam QE 12-bit 1376 × 1040 CCD and the Photron Fast-Cam APX RS 10-bit CMOS camera. From these images, the mean and standard deviation of the noise intensity were calculated and applied to the synthetic images.

4.2 Experimental Setups

In order to determine the robustness of the autocorrelation-based density method, testing the method on actual experimental data is important. This was done by applying the method on data from two separate experimental setups with different seeding particles and flow media. The image densities were varried by introducing more seed into the fluid. Each experimenta setup is discussed in detail in the following sections.

4.2.1 Laminar Jet

The first experimental setup consisted of a high-Reynolds-number, large shear, laminar rectangular jet submerged in ambient air (Fig. 4.2). Images were acquired using the PCO camera described above with a 105 mm lens and a New Wave dual-cavity 50 mJ / pulse Nd:YAG laser. This system was controlled using DaVis 7.2 from LaVision [7] and seeded with olive-oil droplets formed in a Laskin nozzle and added at the blower inlet.

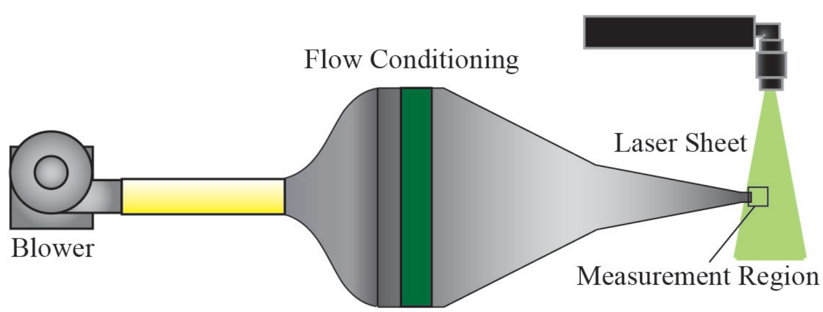


Fig. 4.1: The schematic for the lamnar jet setup. The air drawn in by the blower is seeded, conditioned, and then exits to the measurement region.

Six evenly spaced values of the volume flow rate through a Laskin nozzle were used to adjust the seeding density and, thereby, the particle image density. The values through

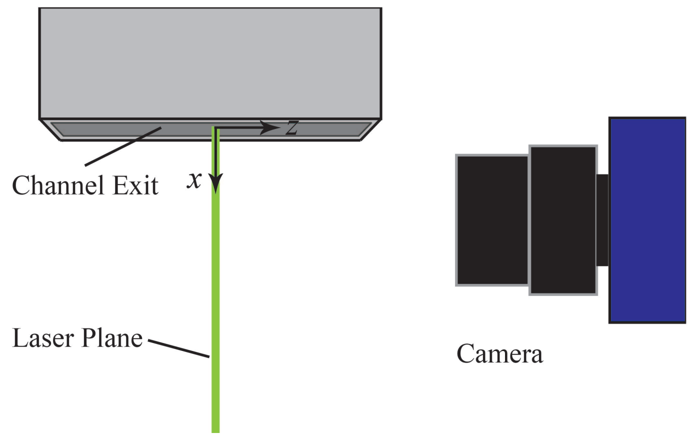


Fig. 4.2: Data is taken at the channel exit. The camera is positioned perpendicular to the flow and normal to the laser plane.

the Laskin nozzle ranged from 200 to 450 standard liters per minute (SLPM) in increments of 50 SLPM. The input air volume flow rate and the output seed mass flow rate have been shown to behave somewhat linearly [36]. For each case, 1000 images were acquired and analyzed using the autocorrelation-based density method. Density estimates were also made by applying the local maximum method discussed earler, as well as manually counting particles from several interrogation regions.

4.2.2 Aquarium

The second experimental setup consisted of a 10-gallon aquarium filled with water and seeded using Potters 110P8 hollow glass microspheres, which have the characteristics shown in Table 4.1. The same laser, lens, and camera were used in conjunction with DaVis 7.2 to acquire the images. A schematic of the aquarium setup is found in Fig. 4.3. As part of this experiment, the effect of particle image diameter and image intensity on the density estimation was investigated.

The particle image diameter can be varied by adjusting the f-number (or aperture) of the lens. The particle image diameter varies as a function of the f-number through the following approximation [12]:

$$d_\tau = \sqrt{(Md_p)^2 + d_s^2}, \tag{4.1}$$

Table 4.1: The properties of Potters hollow microspheres used as seed. [1]Density is measured by gas displacement pycnometer. [2]Bulk Density is the weight as measured in a container and includes the interstitial air. [3]Data represents percent volume distribution measured using laser light scatter technique.

Properties of Potters 110P8 Microspheres	
Density[1], g/cc	1.10 ± 0.05
Bulk Density[2], g/cc	0.49
Size Distribution[3] (μm)	
10%	5
50%	10
90%	21
97%	25

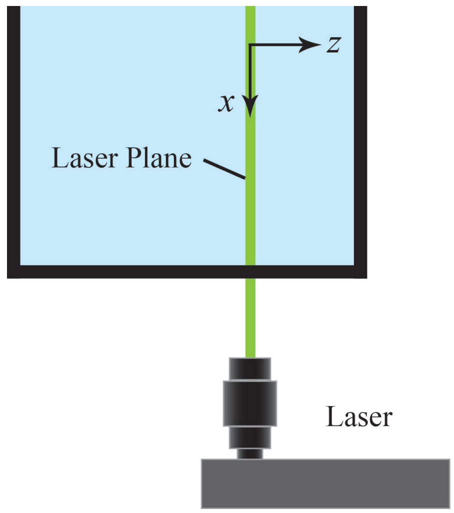
where M denotes the magnification of the image, d_p is the actual particle diameter, and d_s is the diffraction-limited spot diameter,

$$d_s = 2.44\,(1 + M)\,f^{\#}\lambda, \tag{4.2}$$

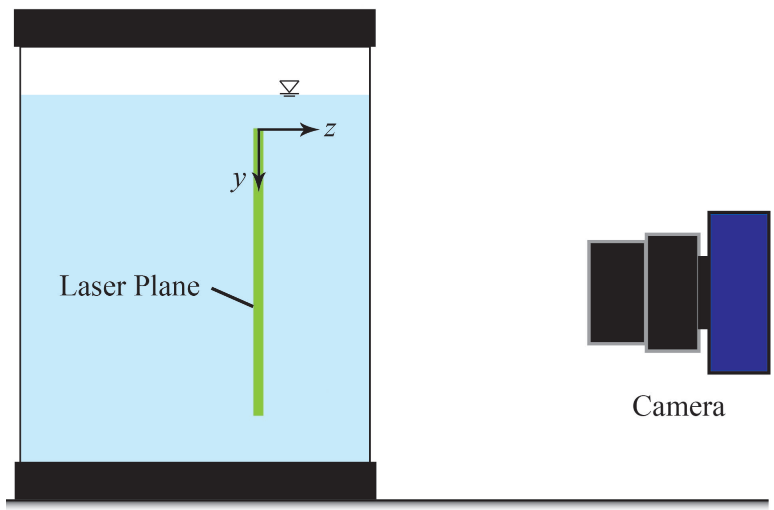
where λ is the wavelength of the light sheet, and $f^{\#}$ is the ratio of the lens focal length, f, and aperture diameter, D_a.

The amount of light that reaches the CCD array of the camera is a function of the laser intensity and the aperture of the lens. By reducing the camera lens aperture while maintaining a constant laser intensity, the effective amount of light incident on the CCD array of the camera will decrease. This decrease in the amount of light captured by the camera will result in fewer recognizable particle images.

To better illustrate why this decrease particle image density occurs, consider seeding particles that have a size distribution that is Gaussian (the exact shape of the distribution is not important to this discussion). The intensity of the light reflected from the particles is a strong function of particle diameter [12, 17] and their location in the Gaussian-profiled light sheet. Therefore, the intensities of the reflections off the particles will also have a wide distribution. Meanwhile, the camera sensor can only capture particle intensities that are above its floor value. A decrease in the image intensity, weather due to the laser intensity or lens aperture size, will result more particles dropping below the noise floor.



(a)



(b)

Fig. 4.3: The schematics for the aquarium setup. Views include (a) an overhead and (b) side view of the setup. The camera is positioned normal to the laser plane.

The situation is illustrated in Fig. 4.4. As the aperture size decreases (Fig. 4.4a to Fig. 4.4c), for a fixed laser intensity and number of actual particles, the average particle reflection intensity decreases and the reflections from smaller particles, as well as particles near the edge of the laser sheet, are lost in the noise floor. Therefore, even though the number of particles inside the field of view remains the same, the particle image density (from the camera's point of view, which is what is important to PIV) decreases. Clearly, the extent to which this is observed will depend on the width of the particle size distribution.

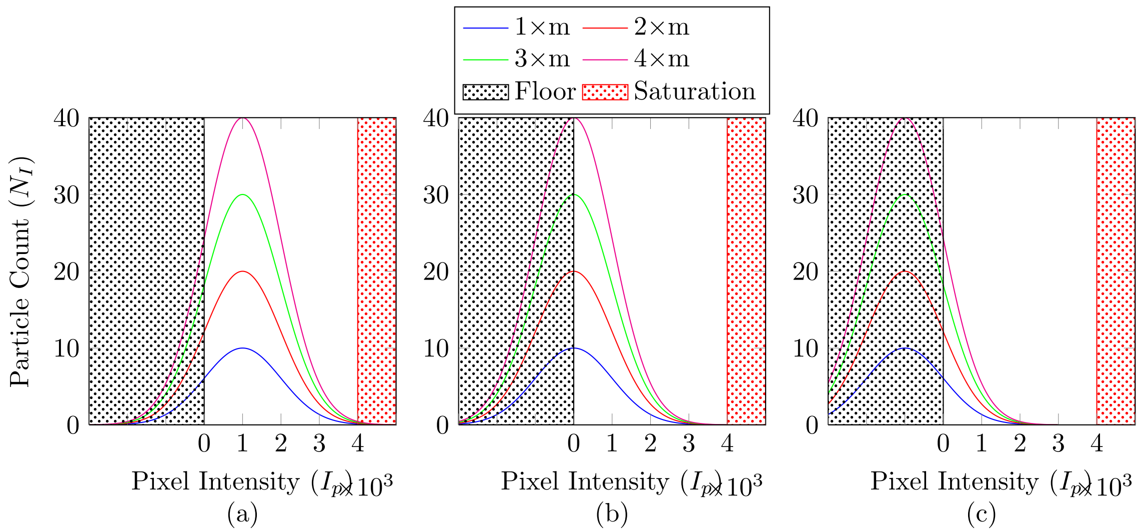


Fig. 4.4: Effect of laser intensity and camera aperture on the estimated particle image density. Three apertures are depicted: (a) large, (b) medium and (c) small. For each case, four values of total seed mass m are depicted. The black and red shades represent the camera noise floor and saturation level, respectively.

The area under any curve and outside of the shaded regions in Fig. 4.4 represents the number of visible particles in the interrogation region. When the amount of seed is doubled, the density will double. As more seed is added, the density will continue to increase linearly, but with different slopes for each case. This trend may also occur for a case with a constant aperture and decreasing laser intensity.

Five sets of data, each with twelve density levels, were acquired using a PCO Sensicam QE 12-bit 1376 $\times$ 1040 CCD with a Nikon 105mm $f/2.8$D AF Micro-Nikkor Lens. The f-number (which is inversely proportional to aperture size) and laser intensity were varied in each set to determine the effect of particle image diameter and image intensity on the density estimation.

The first three image sets had f-numbers of 11, 16, and 22. The laser intensity $\mathcal{J}$ for the first three sets was adjusted such that few particles reached the camera saturation level.

The final two image sets, $f/16$ ($\mathcal{J}_{f11}$) and $f/22$ ($\mathcal{J}_{f11}$), were acquired using the higher two f-numbers while using the same laser intensity used for the $f = 11$ image set.

The seeding density was varied by adding known masses of seed to the flow medium. The mass of each seed sample was measured using a highly accurate analytical balance. For each measured mass of seed, 500 images were acquired and analyzed using the autocorrelation-based density method. Again, density estimates were obtained by applying the local maximum method, as well as manually counting particles from several interrogation regions.

Chapter 5

Results

This chapter discussed the derivation of the empirical equation for estimating the particle image density. The autocorrelation-based density (ABD) method is then applied to synthetic data with known parameters in order to demonstrate the effect of particle image diameter and particle image density on the ABD estimate. The effect of noise on the ABD estimate is then investigated by adding artificial noise to the synthetic images. Lastly, the ABD method is applied to experimental data from the two different experimental setups. Results from the experimental data are discussed, including the effect of varying the particle image diameter and image intensity.

5.1 Empirical Relationship

The ABD method for estimating the particle image density is based on relationship between the autocorrelation peak height and the other parameters discussed previously. The relationship was developed using synthetic data with known particle image diameter and particle image density. The parameter space for the synthetic images used are shown in Table 5.1. While the particle image diameter and density were specifically specified, the average particle intensity was allowed to vary as a function of the other parameters. As the particle image diameter and density increase, so does the number of overlapping particles and, therefore, the estimated average particle intensity.

Table 5.1: Synthetic image parameter space.

Synthetic Image Parameter	Lower Limit	Upper Limit	Step size
Particle Image Diameter (pixels)	1.5	8.0	0.5
Particle Image Density (particles/32×32)	5	80	5

For each combination of parameters, 1000 interrogation sized synthetic images were generated. Interrogation sizes included 32×32, 64×64, 128×128, and 256×256. The relative autocorrelation peak height, average particle intensity, and particle image diameter were calculated for each synthetic image as previously described. Values for these parameters were averaged and are shown in Fig. 5.1 where the particle image density N_I is a function of the particle image diameter d_τ and the multiplication of R_h, A, and $\overline{I_P}$.

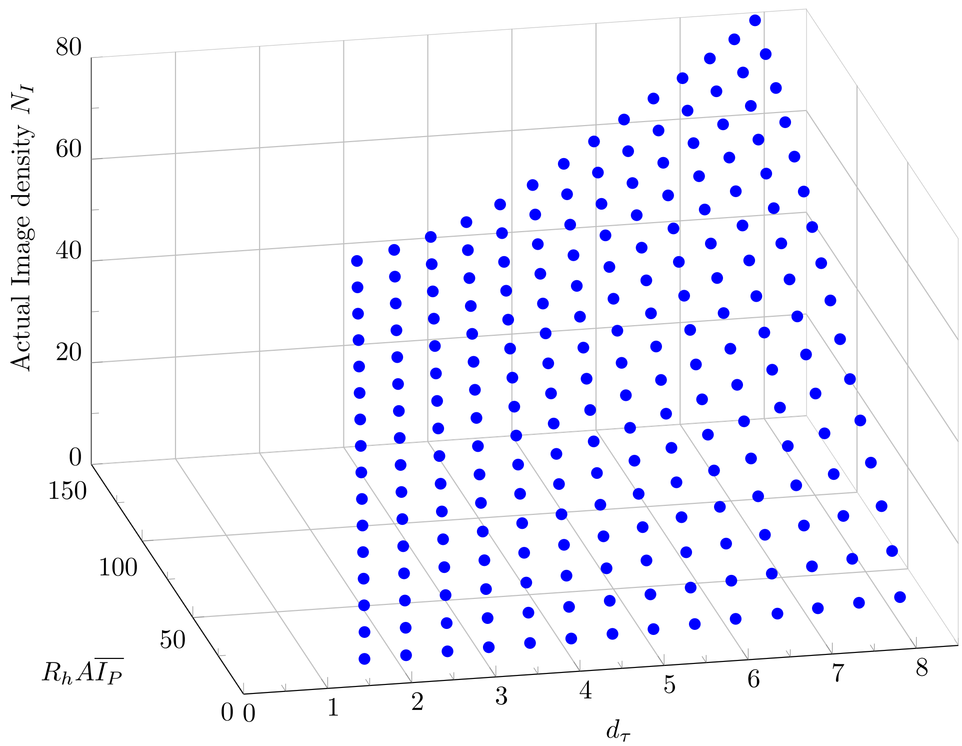


Fig. 5.1: Averaged values of the particle image diameter, relative autocorrelation peak height, interrogation area, and average particle intensity obtained from synthetic images are plotted as a function of the true particle image density. Values for R_h, A, and $\overline{I_P}$ are combined through multiplication in order to reduce the 5-D surface to a 3-D surface for viewing purposes.

The relationship between the parameters and the particle image density was modeled using a power-law function of the form

$$N_I \cong a \left(R_h\right)^b \left(d_\tau\right)^c \left(\overline{I_p}\right)^d \left(A\right)^e, \tag{5.1}$$

where the coefficients a, b, c, d, and e were determined using a nonlinear least-squares fit with a trust-region algorithm. Values for the coefficients are found in Table 5.2 with their corresponding goodness of fit value.

Table 5.2: The coefficients for Eq. 5.1 determined using a least-squares fit and the corresponding goodness of fit value.

a	b	c	d	e	R-square
178	1.40	-2.03	-2.05	-1.42	0.993

Applying the coefficients from Table 5.2 to Eq. 5.1 yields

$$N_I \cong 178 \frac{(R_h)^{1.40}}{(d_\tau)^{2.03} \left(\overline{I_p}\right)^{2.05} (A)^{1.42}}. \tag{5.2}$$

All of the synthetic images defined by the parameter space were processed using Eq. 5.2. For each parameter set, the mean particle image density $\overline{N_I}$ was calculated as

$$\overline{N_I} = \frac{1}{N} \sum_{i=1}^{N} (N_I)_i \tag{5.3}$$

where N is the number of measurements $(N_I)_i$. The absolute error ϵ and the relative error η between the mean particle image density values and the actual density values for a 128×128 interrogation region are shown in Fig 5.2a-b. The absolute and relative errors are defined as

$$\epsilon = \left| N_{I,true} - \overline{N_I} \right| \tag{5.4}$$

and

$$\eta = \frac{\left| N_{I,true} - \overline{N_I} \right|}{\left| N_{I,true} \right|} \tag{5.5}$$

where $N_{I,true}$ is the actual density of the synthetic images.

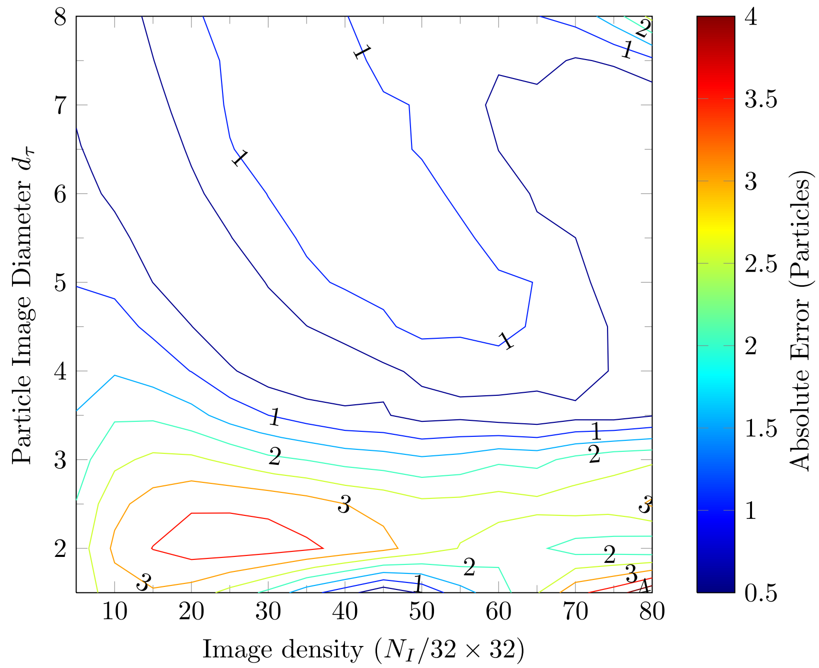
While using a 128×128 interrogation region, the ABD method produces a maximum absolute error of 4.5 particles at a particle image density of 80 particles/(32×32) and a particle image diameter of 1.5 pixels which corresponds to a relative error of 5.6%. The maximum relative error of 44.1% occurs at a particle image density of 5 particles and a 2.0 pixel particle image diameter with a corresponding absolute error of 2.2 particles/(32×32) region.

For the 256×256 interrogation region, the absolute error (Fig. 5.3a) reaches a maximum of 4.8 particles/(32×32) at an image density of 80 particles/(32×32) and particle image diameter of 1.5 pixels which corresponds to a relative error of 6.0%. The relative error (Fig. 5.3b) reaches a maximum of 47.0% at a particle image density of 5 particles/(32×32) and particle image diameter of 2 pixels which corresponds to an absolute error of 2.3 particles/(32×32).

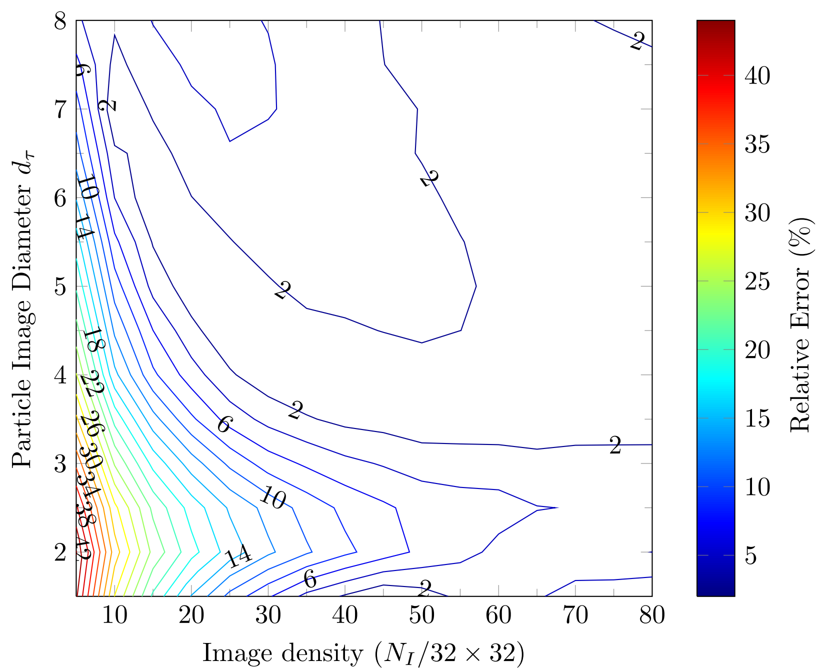
When using a 64×64 interrogation region, the absolute error (Fig. 5.4a) reaches a maximum of 5.2 particles/(32×32) at a particle image density of 80 particles/(32×32) and particle image diameter of 1.5 pixels which corresponds to a relative error of 6.4%. The relative error (Fig. 5.4b) reaches a maximum of 40.6% at a particle image density of 5 particles/(32×32) and particle image diameter of 2 pixels which corresponds to an absolute error of 2.0 particles/(32×32).

Lastly, a 32×32 interrogation region provides a maximum absolute error of 6.9 particles/(32×32) at a particle image density of 80 particles/(32×32) and particle image diameter of 1.5 pixels which corresponds to a relative error of 8.6% (Fig. 5.5a). The maximum relative error for a 32×32 interrogation region is 31.9% at a particle image density of 5 particles/(32×32) and particle image diameter of 2 pixels which corresponds to an absolute error of 1.6 particles/(32×32) (Fig. 5.5b).

The average absolute and relative errors for each interrogation region size are shown in Table 5.3. The lowest errors on average are obtained by using the 64×64 interrogation region. For all cases, the relative error is largest at an image density of 5 particles/(32×32) and a particle image diameter of 2 pixels. The absolute error at this location is relatively small, however, the value for particle image density is below the value recommended for PIV and should be avoided. For all cases the aboslute error is highest at a particle image density of 80 particles and a particle image diameter of 1.5 pixels.

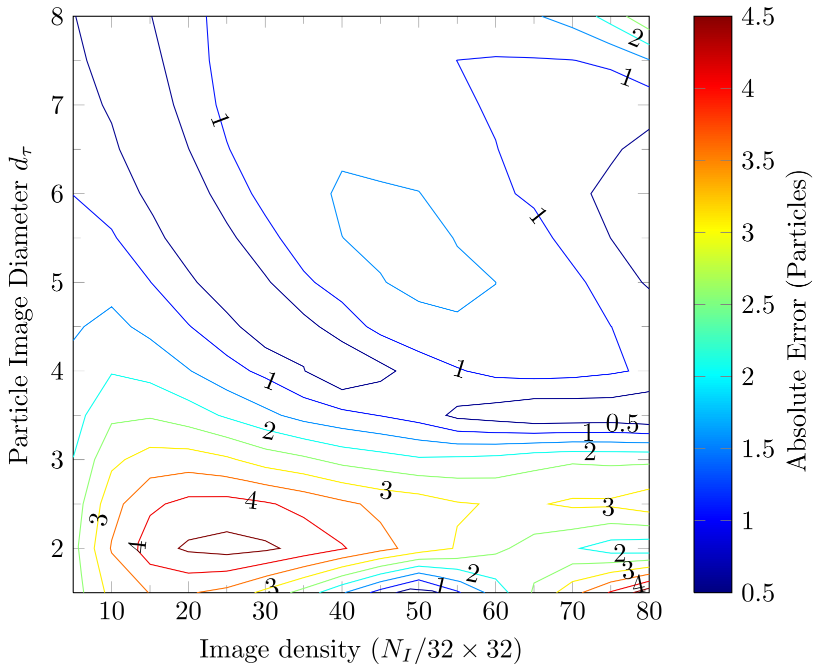


(a)

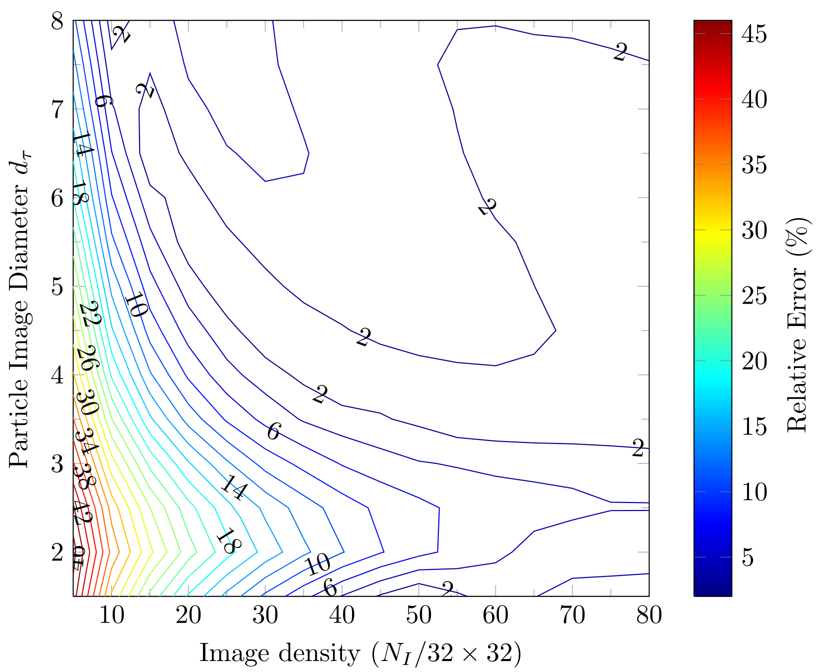


(b)

Fig. 5.2: The relative and absolute errors of the average particle image density calculated using the ABD method. Estimated results are based on the average N_I from 1000 synthetic images with known densities. A 128×128 interrogation region was used.

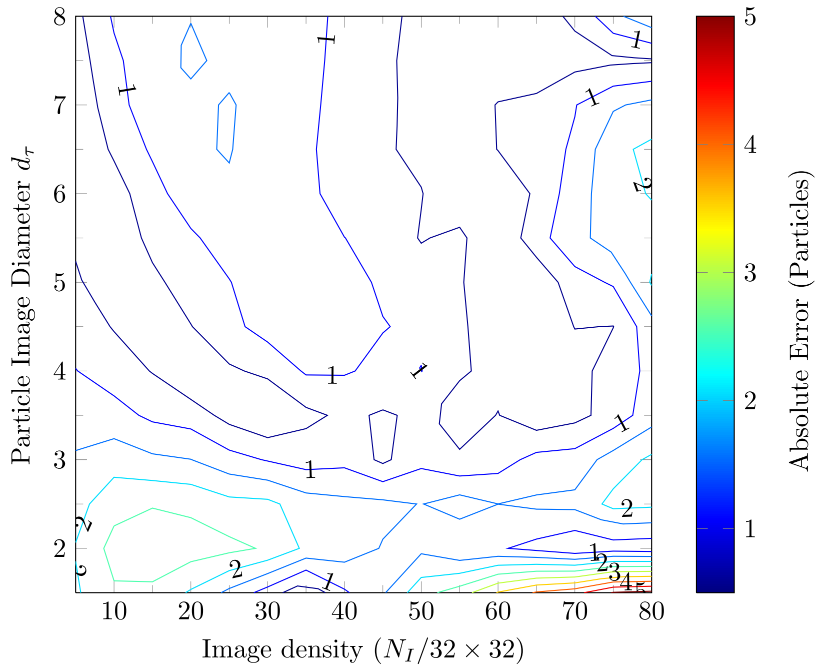


(a)

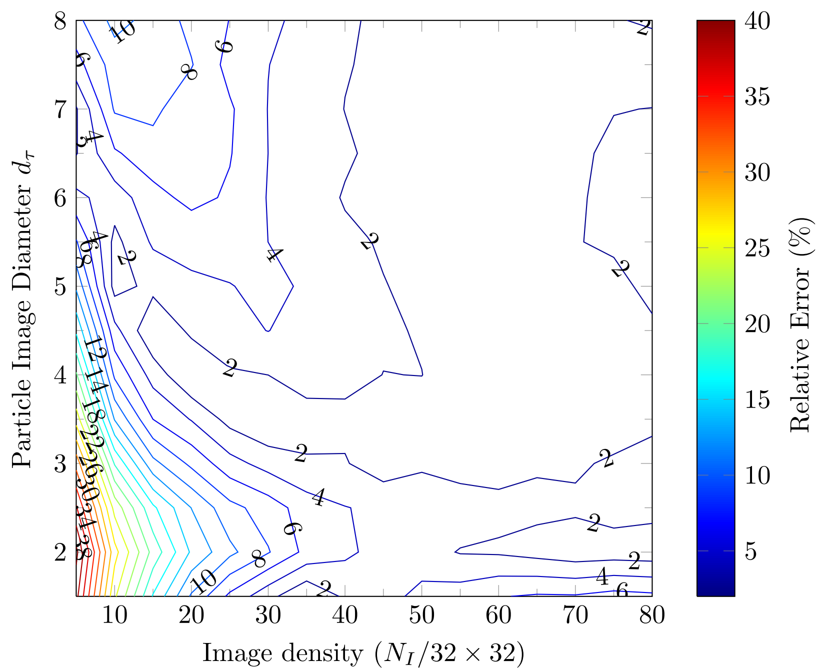


(b)

Fig. 5.3: The relative and absolute errors of the average particle image density calculated using the ABD method. Estimated results are based on the average N_I from 1000 synthetic images with known densities. A 256×256 interrogation region was used.

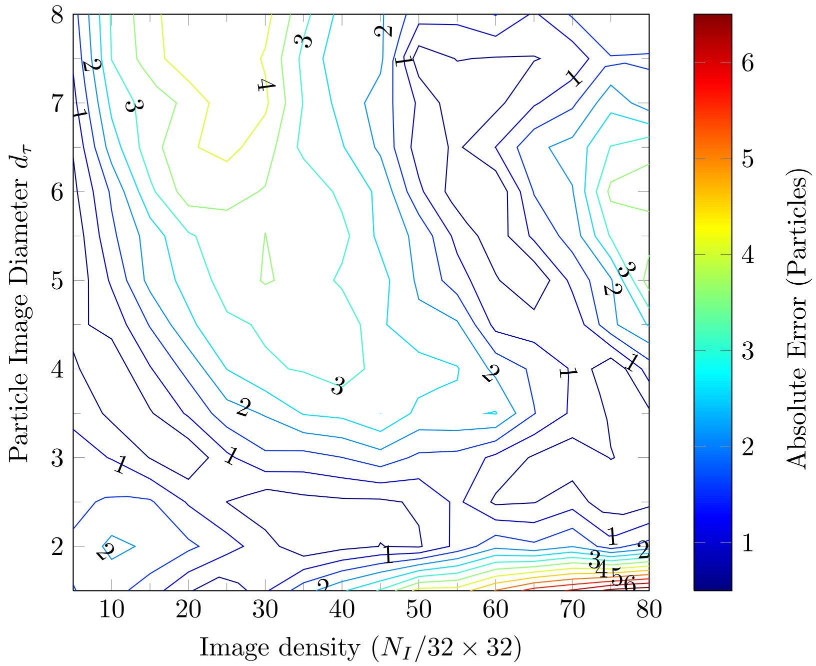


(a)

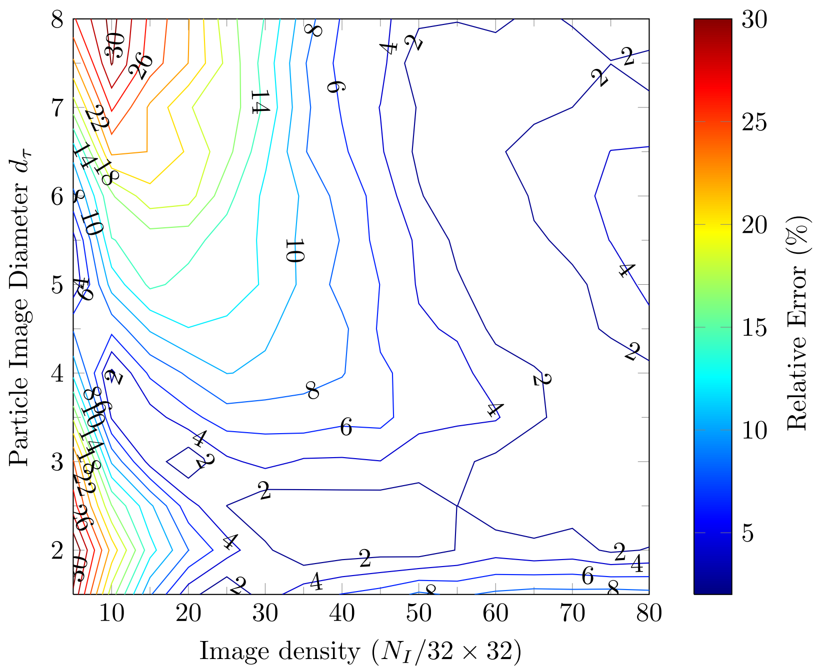


(b)

Fig. 5.4: The relative and absolute errors of the average particle image density calculated using the ABD method. Estimated results are based on the average N_I from 1000 synthetic images with known densities. A 64×64 interrogation region was used.



(a)



(b)

Fig. 5.5: The relative and absolute errors of the average particle image density calculated using the ABD method. Estimated results are based on the average N_I from 1000 synthetic images with known densities. A 32×32 interrogation region was used.

Table 5.3: The mean absolute errors $\bar{\epsilon}$ and the mean relative errors $\bar{\eta}$ for the ABD method obtained for each interrogation size. Errors were obtained by averaging results from the synthetic images used to develop Eq. 5.2

A	$\bar{\epsilon}$ (particles/(32 × 32))	$\bar{\eta}$ (%)
32 × 32	1.91	7.13
64 × 64	1.09	4.50
128 × 128	1.24	5.31
256 × 256	1.57	6.67

5.2 Test with Synthetic Images

While the autocorrelation-based density method was developed using synthetic data, testing it further on synthetic data makes it possible to determine how precision uncertainty varies with the given image parameters and to investigate the ABD method's sensitivity to image noise. The autocorrelation-based density method previously discussed was coded into MATLAB and is shown in the Apendix. The ABD method was applied to synthetic images with known particle image diameters and densities. The parameter space used may be found in Table 5.1.

The results for the following cases were obtained by calculating the particle image density for 100 interrogation regions using both the autocorrelation-based density method and the local maximum method previously discussed. The sample mean $\overline{N_I}$, defined in Eq. 5.3, and sample standard deviation s_{N_I}, defined by

$$s_{N_I} = \left[\frac{1}{N-1} \sum_{i=1}^{N} \left((N_I)_i - \overline{N_I} \right)^2 \right]^{1/2} , \tag{5.6}$$

were calculated for each case. The mean particle image density was plotted for each case and the standard deviation was used to calculate uncertainty bands. All uncertainty bands contained in the following figures represent the 95% confidence interval ($1.96 s_{N_I}/\sqrt{N}$ [37]) of the mean particle image density.

5.2.1 Effect of Density

In the first case, synthetic images were generated with varying densities ranging from

5 to 80 particles per 32×32 interrogation region, particle image diameters of 3 pixels, and no noise. Figure 5.6 shows the calculated particle image density as a function of the actual particle image density. The particle image density was calculated using the ABD method as well as the local maximum method for interrogation sizes of 32×32, 64×64, 128×128, and 256×256.

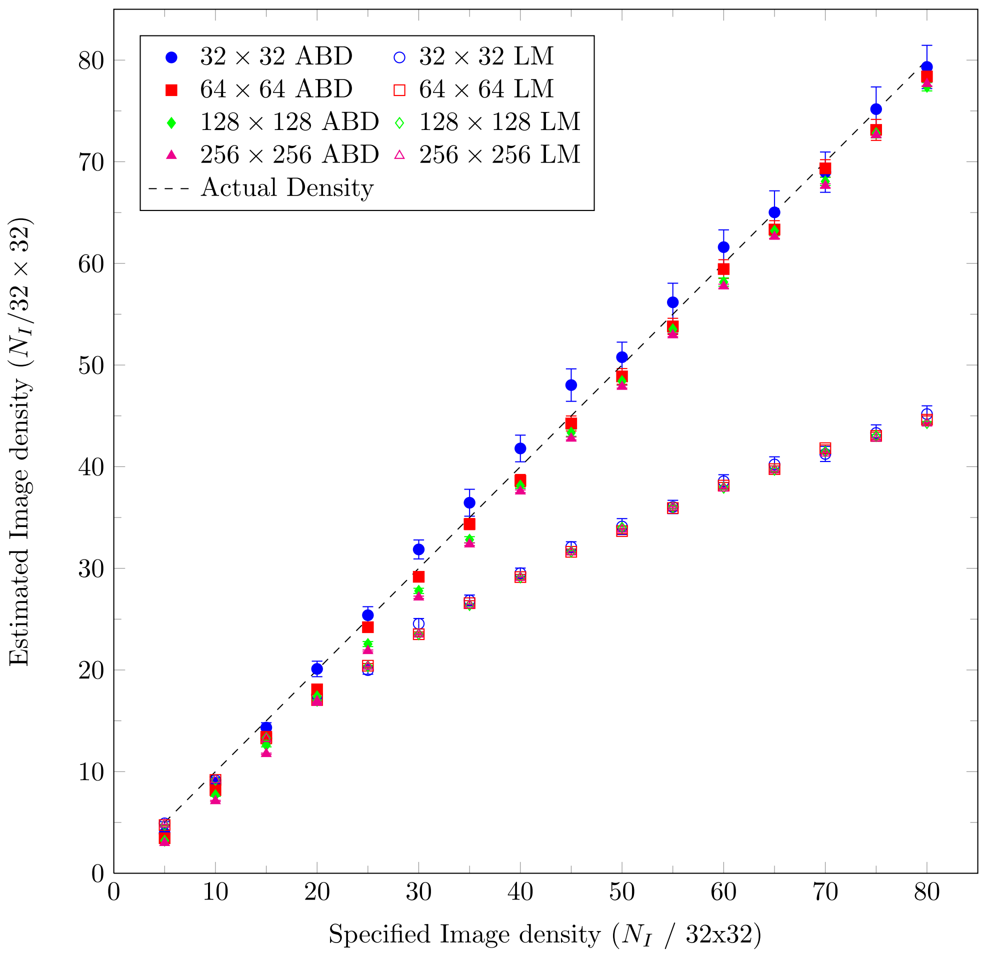


Fig. 5.6: The effect of particle image density on the particle image density estimation. Synthetic images were used with known densities, particle image diameter of 3 pixels, and no noise. The particle image density was estimated using the autocorrelation-based density (ABD) and local maximum (LM) methods.

The ABD method provides similar averaged results for all interrogation sizes. The local maximum method underestimates the particle image density as the actual density increases due to an increase in overlapping particle images. Increasing the density also increases the the precision uncertainty (Fig. 5.7). However, as the interrogation region size increases, the effect of increased density on the precision uncertainty diminishes due to spatial averaging. Larger interrogation regions allow for more spatial averaging and less variation in the estimated particle image density.

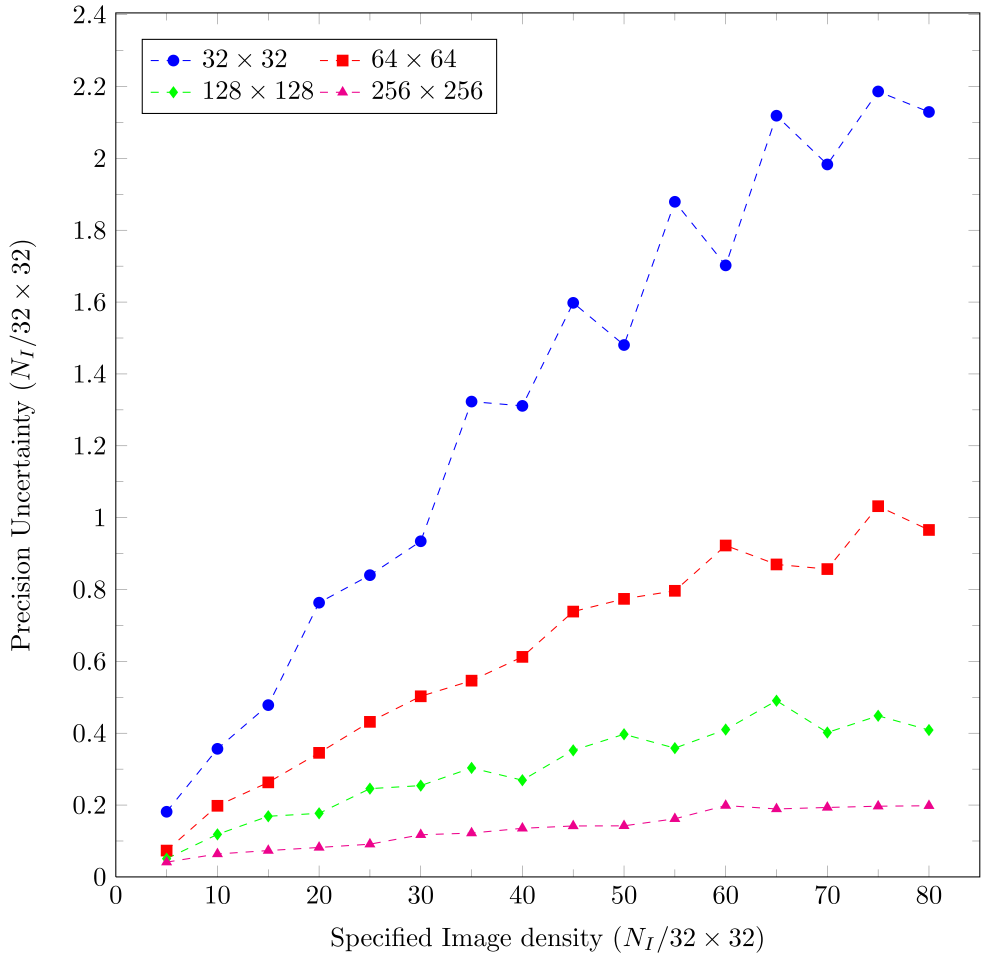


Fig. 5.7: Precision uncertainty of the particle image density estimation results in Fig. 5.6. The folowing values represent the 95% confidence interval ($1.96 s_x/\sqrt{N}$) for each interrogation region size.

5.2.2 Effect of Diameter

The particle image diameter also affects the performance of the ABD method as shown in Fig. 5.8. Images were generated with varying particle image diameters ranging from 1.5 to 8 pixels, constant density of 40 particles per 32×32 region, and no synthetic noise. Again the particle image density was calculated using the autocorrelation-based density method as well as the local maximum method for interrogation sizes of 32×32, 64×64, 128×128, and 256×256.

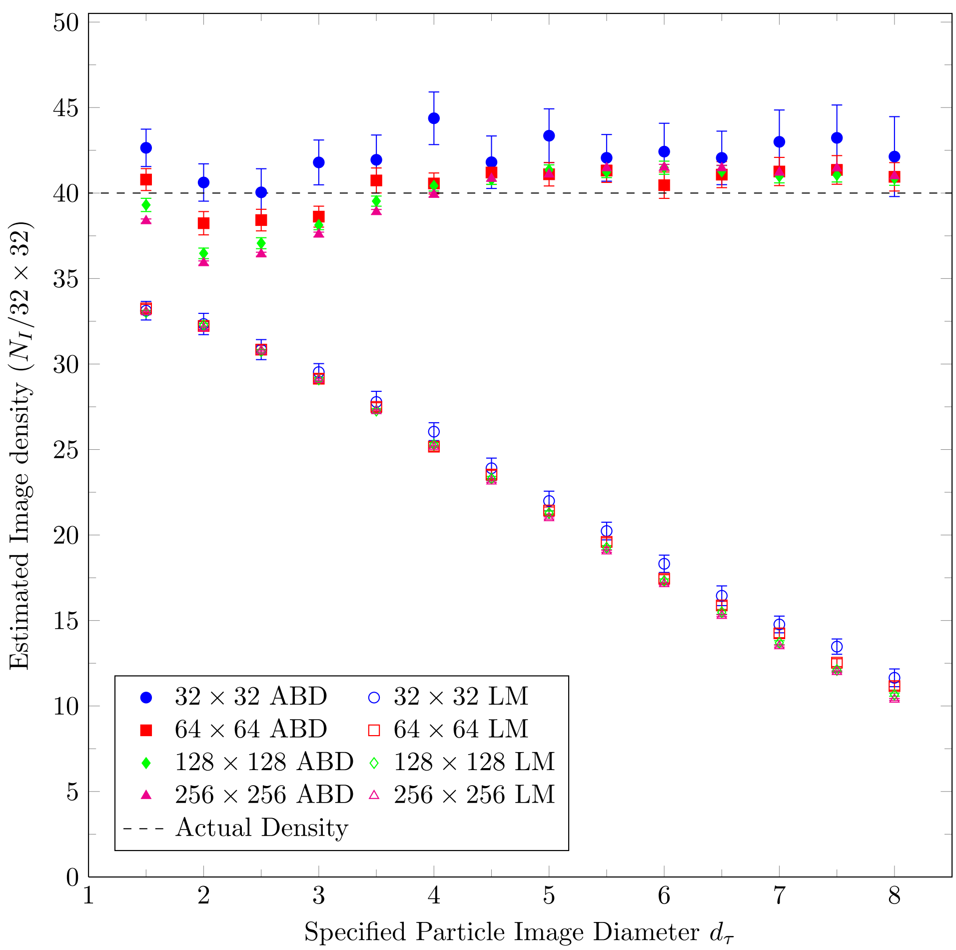


Fig. 5.8: The effect of particle image diameter on the particle image density estimation. Synthetic images with a particle image density of 40 particles per 32×32 region and no noise were used. The particle image density was estimated using the autocorrelation-based density (ABD) and local maximum (LM) methods.

The mean ABD estimates vary between interrogation sizes due to the goodness of the nonlinear least squares fit. In general, the particle image density estimates decrease with increased interrogation size. For all cases the ABD estimates are significantly more accurate than the results obtained by the local maximum method. The local maximum method underestimates the particle image density as the particle image diameter increases. Increasing the particle image diameter increases the amount of particle image overlap and decreases the number of distinct particle images. For all interrogation sizes, the precision uncertainty remains fairly constant as the diameter increases (Fig. 5.9). However, as the interrogation region size increases, the effect of diameter on the precision uncertainty diminishes due to spatial averaging.

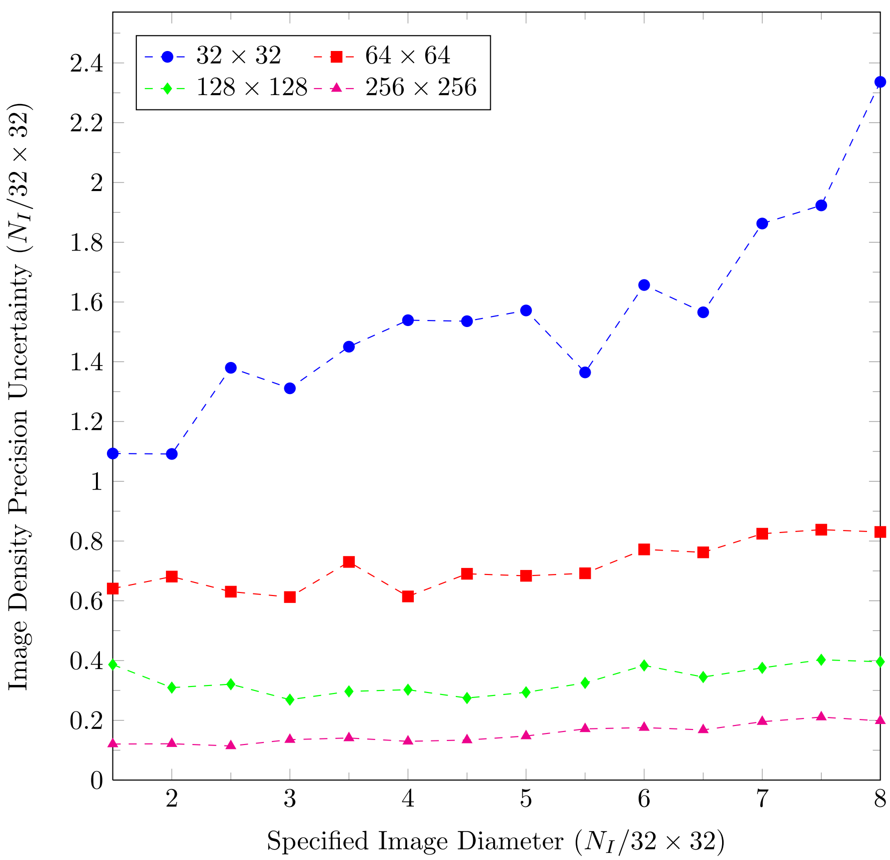


Fig. 5.9: Precision uncertainty of the particle image density estimation results in Fig. 5.6. The folowing values represent the 95% confidence interval, $1.96 s_x/\sqrt{N}$.

5.2.3 Effect of Noise

Two levels of background noise were added to the synthetic images to simulate the noise generated by the PCO Sensicam and Photron FastCam cameras with lens caps in place. The mean and standard deviation of the noise intensity was calculated from 100 images pairs and are found in Table 5.4. The artificial noise was applied to synthetic images randomly using a normal distribution. The synthetic images had particle images with 3-pixel diameters. The average image densities for 1000 images were calculated with the autocorrelation-based density method and the local maximum method using 32×32 and 128×128 interrogation regions.

Table 5.4: The mean and standard deviation of noise from two PIV cameras.

Camera	Bit Depth	Mean	Standard Deviation
PCO Sensicam	12-bit	41.411	2.441
FastCam APX RS	10-bit	2.902	14.541

The results in Fig. 5.10 show that the particle image density estimation is mostly unaffected by the synthetic background noise for a 32×32 interrogation region. As the density increases the particle image density estimates from the camera 2 noise synthetic images drop below the particle image density estimates from the images with no noise and camera 1 noise. The same trend is seen in the precision uncertainty shown in Fig. 5.11.

The results in Fig. 5.12 show that the density estimation is mostly unaffected by the synthetic background noise for a 128×128 interrogation region and produces similar results for all cases. However, as the particle image density increases, the precision uncertainty increases similarly for the no noise and camera 1 noise cases (Fig. 5.13), while the precision uncertainty for the camera 2 noise case somewhat drops off.

Overall, the addition of noise to the synthetic images has little effect on the autocorrelation-based density method. Most of the negative effects of the synthetic noise are removed by the background image removal process discussed earlier in this paper. In order to demonstrate the effect of background image subtraction, the image densities of the synthetic images

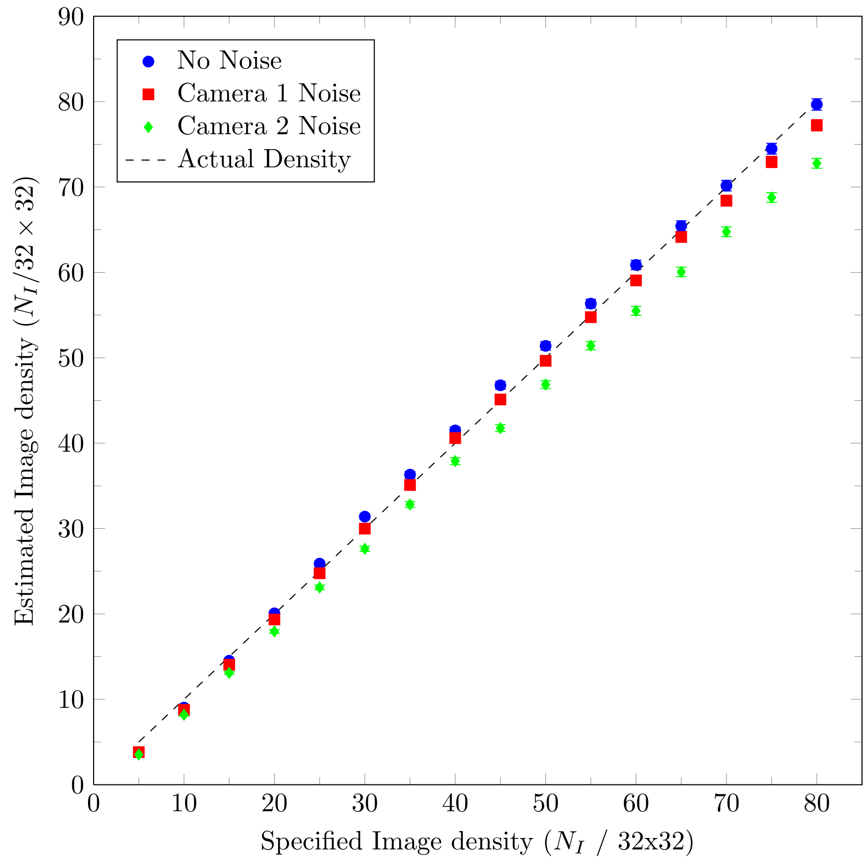


Fig. 5.10: Effect of introducing noise into the synthetic images on the particle image density estimation for a 32×32 interrogation region. The noise generated simulates the noise introduced using two different types of cameras.

with noise from camera 1 and 2 were reprocessed without background removal. Results from both the autocorrelation-based density method and the local maximum method for a 128×128 interrogation region are shown in Fig. 5.14.

Without the use of background subtraction, both the autocorrelation-based density method and the local maximum method are unable to provide acceptable estimates of the particle image density. At low image densities both methods severely over estimate the particle image density due to the random fluctuations of the noise. For the local max method, the random noise fluctuations form peaks, or “false particles,” that contribute to the particle image density. These false particles have a relatively low intensity compared to true particles. At low image densities the false particles dominate the number of true particles

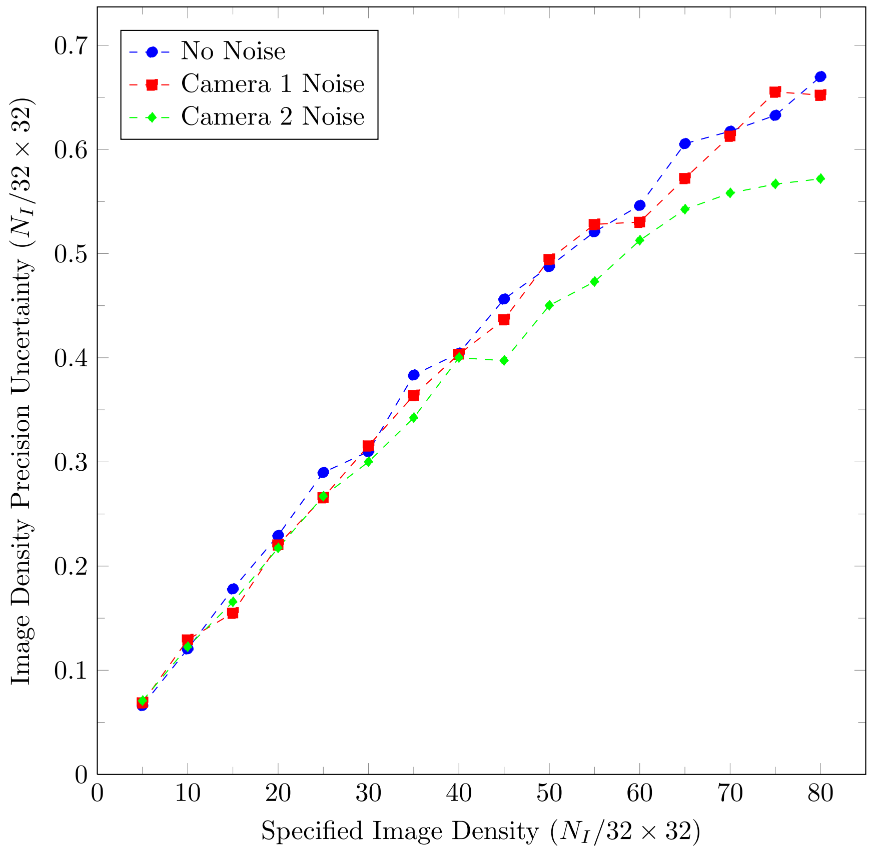


Fig. 5.11: The effect of noise on the precision uncertainty of the density estimation results in Fig. 5.10.

and, therefore, reduce the average intensity which, in turn, causes the autocorrelation-based density method to overestimate the particle image density. As the density increases, the particle image density estimates begin to be dominated by true particle images (rather than false particles), and the slopes of the noisy data converge to the slope of the no noise particle image density estimates. Background image subtraction should always be performed when using the autocorrelation-based density method to estimate the particle image density.

5.3 Experimental Verification

Experimental verification of the synthetic image results requires a PIV setup with adjustable particle image density. This requirement was realized by incrementally increasing the amount of seed particles within the fluid. Images with increasing density were collected

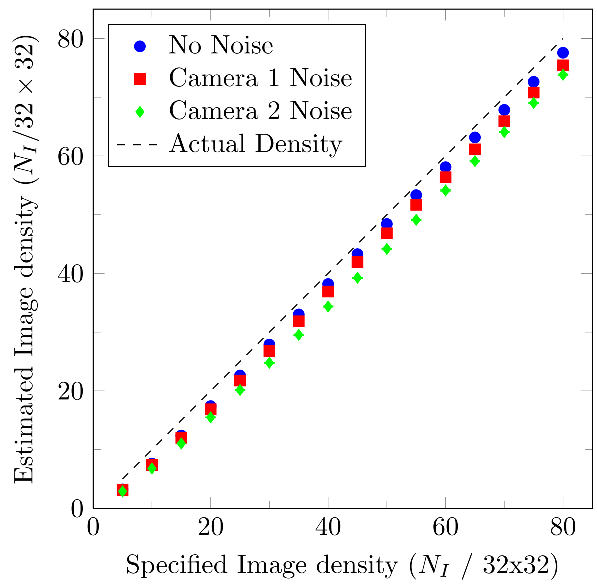


Fig. 5.12: Effect of introducing noise into the synthetic images on the particle image density estimation for a 128×128 interrogation region. The noise generated simulates the noise introduced using two different types of cameras.

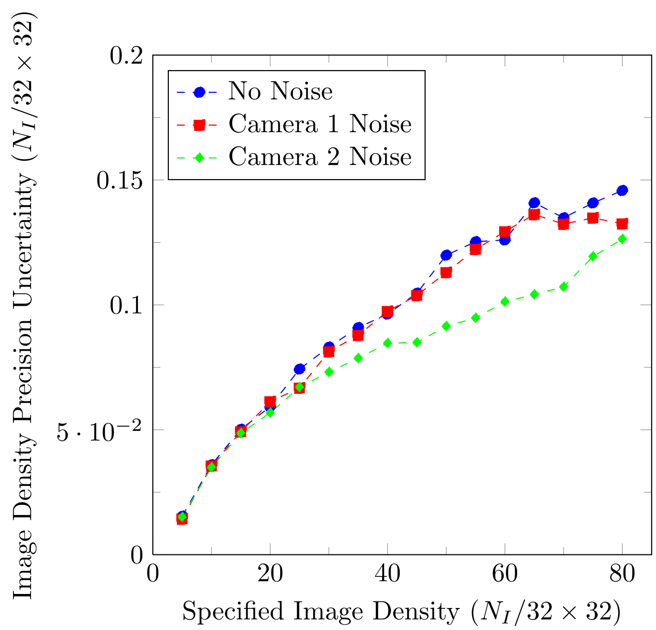


Fig. 5.13: The effect of noise on the precision uncertainty of the density estimation results in Fig. 5.12.

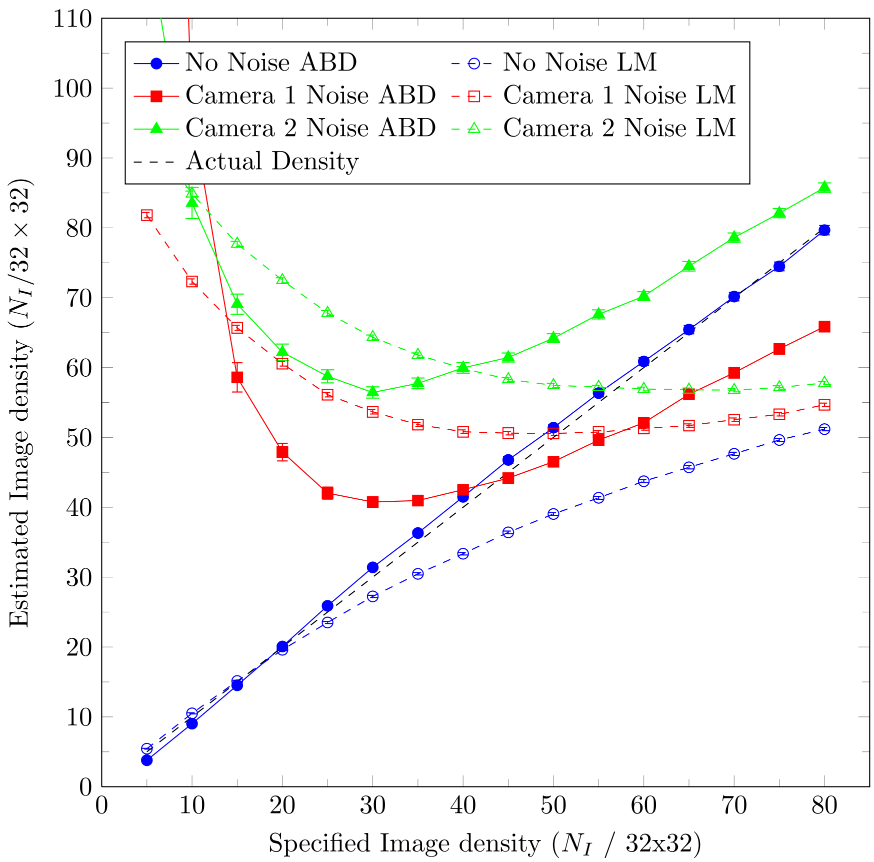
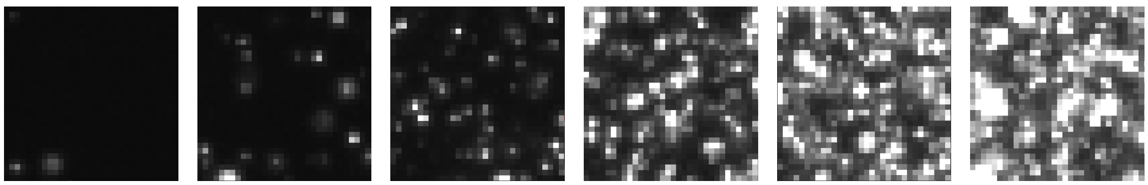


Fig. 5.14: Effect of introducing noise into the synthetic images on the particle image density estimation for a 128×128 interrogation region when background image subtraction is not used. The noise generated simulates the noise introduced using two different types of cameras. The particle image density was estimated using the autocorrelation-based density (ABD) and local maximum (LM) methods.

from two separate experimental setups with different flow media and seed particles.

5.3.1 Laminar Jet Tests

The first experimental setup was a submerged rectangular jet in ambient air seeded with olive-oil droplets as previously described in section 4.2.1. For each of the six evenly-spaced values of the volume flow rate through the Laskin nozzle, 1000 images were acquired and analyzed using the ABD method. Interrogation regions (32×32 pixels) from the images acquired using the laminar jet setup are shown in Fig. 5.15. Particle image density estimates were also made using the local maximum method and by manually counting particles from several interrogation regions. The average of the manually counted densities with their accompanying 95% confidence intervals are plotted along side the results from the ABD and local maximum methods in Fig. 5.16. Confidence intervals for the ABD method and local maximum method are not shown as they are insignificant for such a large sample size.



(a) 200 slpm (b) 250 slpm (c) 300 slpm (d) 350 slpm (e) 400 slpm (f) 450 slpm

Fig. 5.15: Interrogation regions (32×32 pixels) from images acquired using the jet setup. The particle image density was adjusted by increasing the volume flow rate (SLPM) into a Laskin Nozzle.

The results show a nearly linear relationship between flow rate into the Laskin nozzle and the estimated particle image density as anticipated. The local maximum method and manual count results resemble the ABD results until the particle density exceeds 40 particles, after which the local maximum results and manual count start to fall below the ABD methods results. This difference in the results is expected. As the particle image density increases, particle images become lost due to particle overlap. In effect, the rate of particle image density per increase in flow rate decreases for the methods based on local maximums.

5.3.2 Aquarium Tests

The second experimental setup, described in section 4.2.2, consisted of an aquarium filled with water and seeded with hollow glass spheres. Known masses of seeding particles were added to increase N_I. The purpose of this experiment was to further explore the autocorrelation-based density method's ability to estimate the particle image density when subject to variations in particle image density and image intensity.

Twelve values of particle image density were obtained by adding measured masses of seed, whose values are shown in Table 5.5, to the aquarium setup. In attempt to evenly distribute the particles and avoid clumping, the seed samples were first blended into a

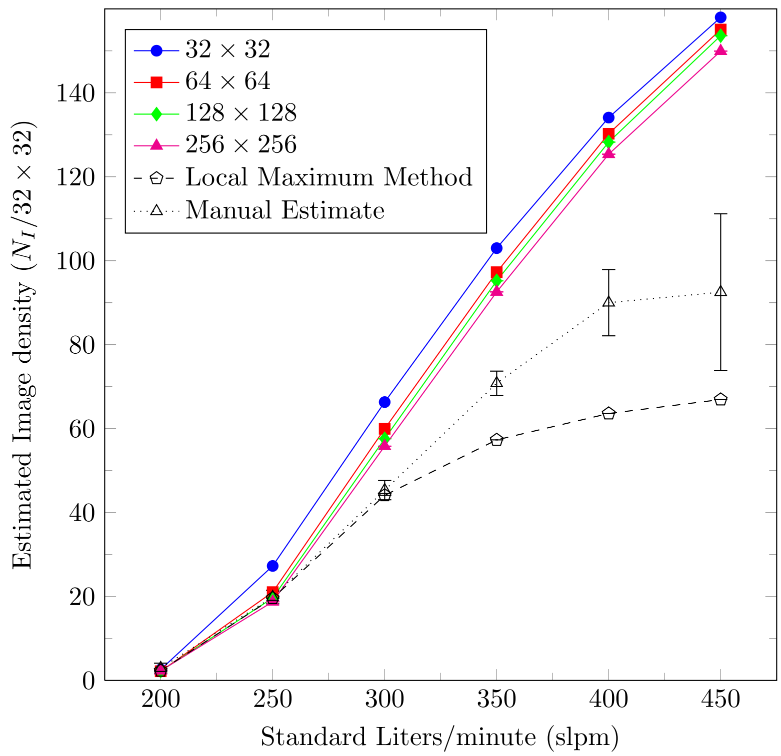


Fig. 5.16: Particle image density estimates as a function of the mass flow rate to the seeder for the rectangular jet case. Results were acquired implementing the autocorrelation-based density method (solid lines and symbols) as well as the local maximum method and manually counting particles from several interrogation regions. Error bars represent the 95% confidence interval.

portion of the flow media being added to the remaining fluid within the aquarium. Sets of 500 images were acquired for each diameter and density. Interrogation regions (32×32 pixels) from the images acquired using the aquarium setup are shown in Fig. 5.17.

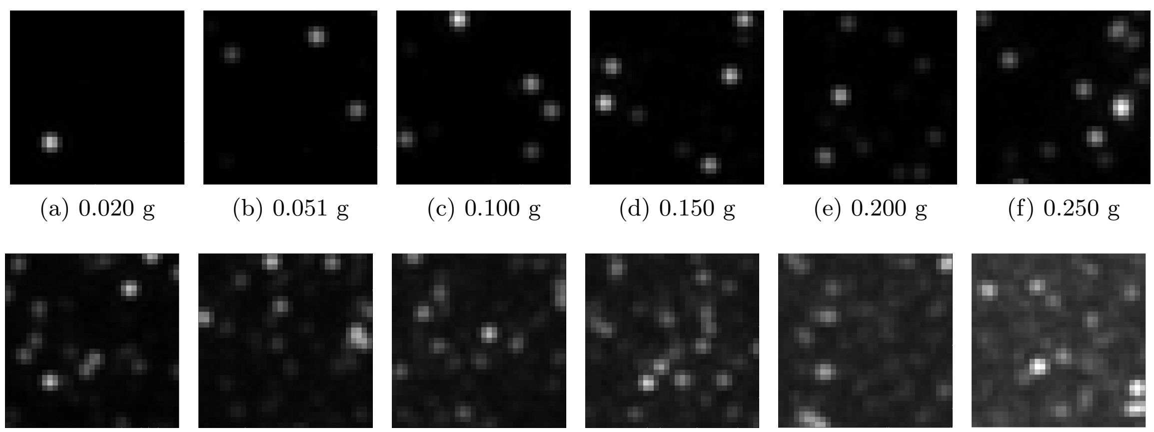
5.3.3 Effect of Particle Image Density

The first aim of this experiment was to determine the effect of the particle image diameter on the autocorrelation-based density method. The diameter was altered by by adjusting the f-number (or aperture) of the lens. Using f-numbers of 11, 16, and 22, produced particle image diameters of 2.4, 3.0, and 4.0 pixels, respectively.

As the f-numbers increases, the aperture decreases, and the amount of light captured by the camera decreases. To compensate for the variation in captured light, the laser intensity

Table 5.5: The samples of hollow glass spheres mixed into the water of the aquarium experiment. The mass of each sample was measured using an analytical balance.

Sample	Mass (g)	Total Mass (g)
1	0.02037	0.02037
2	0.03016	0.05053
3	0.04959	0.10012
4	0.04962	0.14974
5	0.05012	0.19986
6	0.05009	0.24995
7	0.15047	0.40042
8	0.20017	0.60059
9	0.20141	0.80200
10	0.20017	1.00217
11	0.39960	1.40177
12	0.39823	1.80000



(a) 0.020 g (b) 0.051 g (c) 0.100 g (d) 0.150 g (e) 0.200 g (f) 0.250 g

(g) 0.400 g (h) 0.601 g (i) 0.802 g (j) 1.002 g (k) 1.402 g (l) 1.800 g

Fig. 5.17: Images with varying densities acquired from the aquarium setup. The density was increased by adding known masses of hollow glass spheres. Images shown are for the $f/22$ case.

was adjusted in order to obtain similar average image intensities and few saturated particle images for each f-number. The average particle image density values estimated using the autocorrelation-based density method, local maximum method, and manual counting are shown in Fig. 5.18.

The results in Fig. 5.18 show a linear relationship between the cumulative mass of the seeding particles and the density estimated by the ABD method. The data for $f/11$

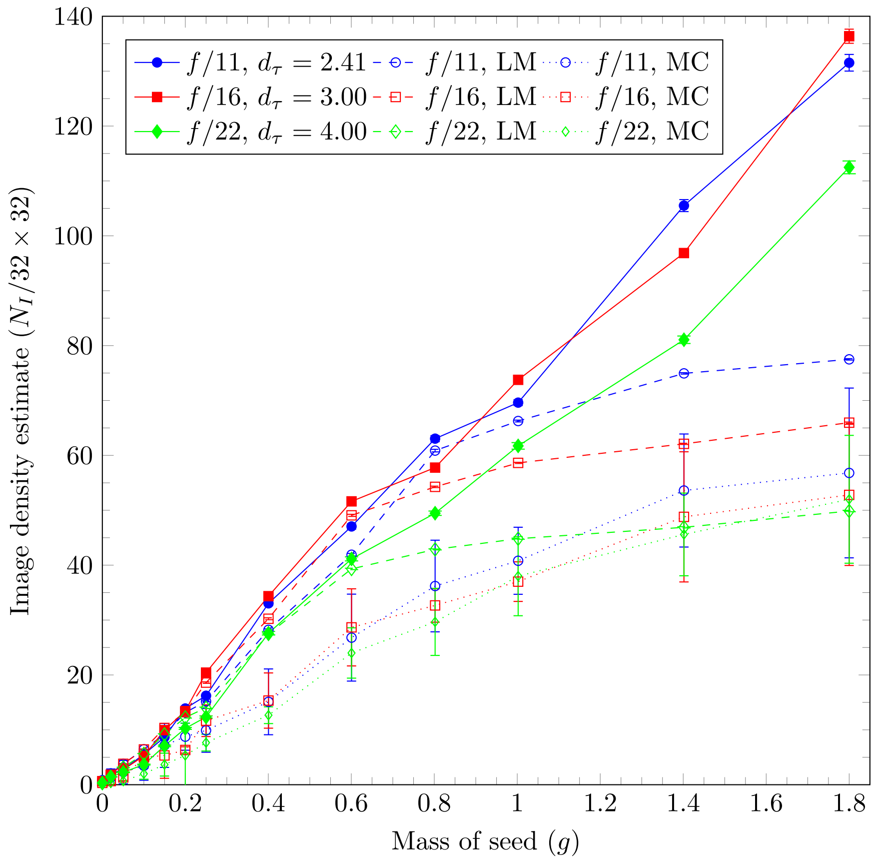


Fig. 5.18: Estimated seeding density as a function of the total mass of seed particles added to the water. Values are shown using three different particle image diameters. The solid lines with the solid symbols represent the average particle image density calculated using the autocorrelation-based density method. The dashed and dotted lines with the open symbols represent the average particle image density estimates from the local maximum (LM) method and manual counting (MC), respectively.

and $f/16$ are very similar because the average image intensities were nearly identical. The maximum available laser intensity for the $f/22$ case was not sufficient to compensate for the decrease in aperture, and thus the intensity of the images was noticeably decreased, leading to a slight loss in recognizable particles and a slight decrease in slope.

The density estimates obtained by the local maximum method and manually counting particles follow the same linear trend at lower densities. As the density increases, the local maximum and manual counting methods are unable to account for overlapping particles and, therefore, underestimates the particle image density. The difference between the local maximum particle image density estimates is due to the increase in particle image diameter. As the diameter increases, more particle overlap occurs, which reduces the particle image density estimate.

5.3.4 Effect of Image Intensity

In the second part of the this experiment, the effect of image intensity was investigated. The image intensity was varied by adjusting the f-number while maintaining a constant laser intensity $\mathcal{J}$. Two image sets were acquired ($f/16$ ($\mathcal{J}_{f11}$) and $f/22$ ($\mathcal{J}_{f11}$)) with the same laser intensity used for the $f/11$ image set. The average particle image density values estimated using the autocorrelation-based density method, local maximum method, and manual counting are shown in Fig. 5.19.

Although the slopes for each f-number is different, the results in Fig. 5.19 still show a linear relationship between the cumulative mass of the seeding particles and the density estimated by the ABD method. For smaller apertures (larger f numbers) the particle image density is smaller as less light reaches the CCD sensor. As before, the local maximum method and manually counting are unable to maintain a linear particle image density relationship as the mass of seed increases due to particle overlap.

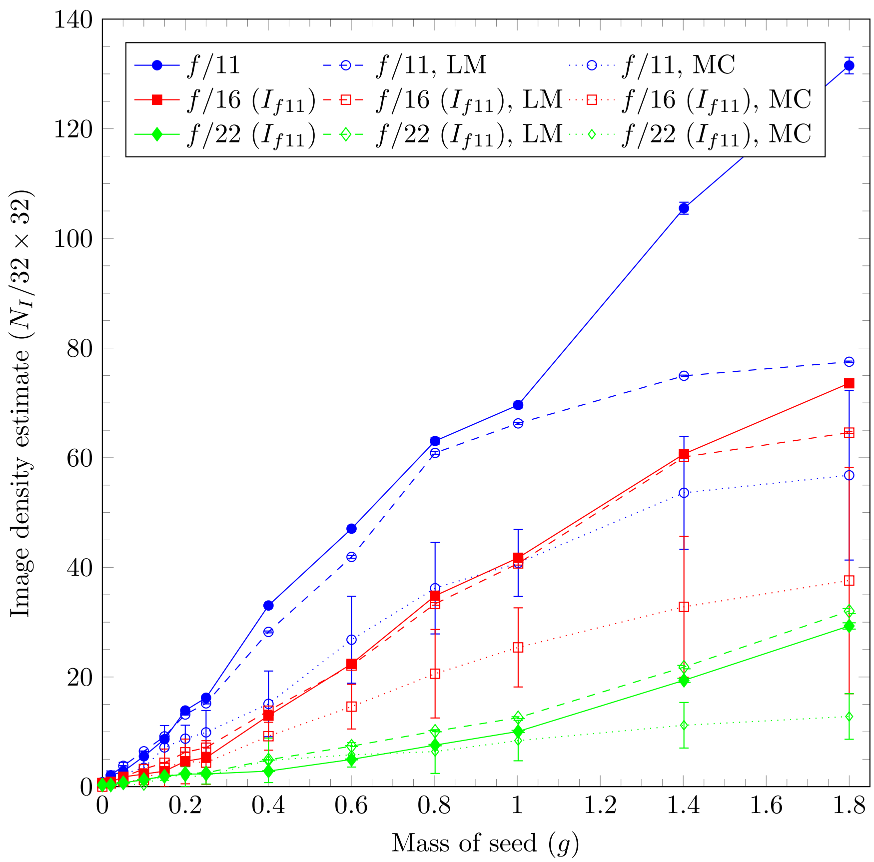


Fig. 5.19: Estimated seeding density as a function of the total mass of seed particles added to the water. Values are shown using three different image intensities. The image intensity was adjusted by using different f-numbers while maintaining a constant laser intensity. The solid lines with the solid symbols represent the average particle image density calculated using the autocorrelation-based density method. The dashed and dotted lines with the open symbols represent the average particle image density estimates from the local maximum (LM) method and manual counting (MC), respectively.

Chapter 6

Conclusions & Future Work

6.1 Conclusions

An autocorrelation-based (ABD) method for estimating the particle image density has been develop. The ABD method is based on the relationship between the relative autocorrelation peak magnitude and the parameters that contribute to its magnitude, namely the particle image diameter, particle image density, interrogation region size, and average particle intensity. The magnitude of each of these parameters was found to be directly proportional to the autocorrelation peak magnitude. Established methods for quantifying the relative autocorrelation peak height and particle image diameter, as well as a new method for estimating the average particle intensity were presented.

Synthetic images with know image parameters were generated. The known image parameters were estimated using the methods previously discussed. A least squares fit was implemented to develop an empirical relationship between the known particle image density and the estimated parameters from interrogation regions of size 32×32, 64×64, 128×128, and 256×256 pixels. The error between the known particle image density and the estimated particle image density from the autocorrelation-based density method was presented. The mean errors were minimized when using the 64×64 interrogation region size having mean absolute and relative errors of 1.09 particles/(32×32) and 4.50%, respectively.

The ABD method was tested further on synthetic images in order to investigate the effect of image parameters on the precision uncertainty and the effect of noise on the particle image density estimate. In general the precision uncertainty decreases with interrogation size, increases with particle image density, and relatively unaffected by particle image diameter. Two levels of noise were added to the synthetic images based on the noise generated by PIV cameras. The effect of the synthetic noise was found to have little influence on the particle image density estimation and precision uncertainty. The effect of background image subtraction on noisy data was also investigated. In the absence of background image subtraction, the error of the particle image density estimates from the noisy data was highly elevated for both the ABD method and the local maximum method. The measurement error was especially high at low particle image density.

The ABD method was then applied to experimental data acquired from two experimental setups. The first setup consisted of a laminar rectangular jet submerged in air and seeded with olive oil droplets. The density of the flow was increased by increasing the volume flow rate to the Laskin Nozzle. The particle image density was estimated with the ABD method, the local maximum method, and by manually counting particles. As anticipated, results from the ABD method provided a near linear increase in particle image density while the estimates of the other two methods eventually leveled off as the volume flow rate increased.

The second experimental setup consisted of an aquarium filled with water. The density was increased by adding known amounts of seed to the medium. As part of this experiment, the effect of particle image diameter and image intensity was investigated by adjusting the f-number and laser intensity. As the particle image diameter increased little change was observed in the particle image density estimates from the ABD and local maximum methods. The image intensity, however, was found to affect the particle image density estimation substantially. As the image intensity decreased (due to an increase in f-number) so did the slope of particle image density results. It is suggested that this difference in the particle image density is due, at least in part, to the amount of light captured by the camera. As the aperture decreases, the intensity of light captured by the camera decreases, which results in dim particles becoming more dim and becoming lost amidst the noise floor.

For both experimental studies, the ABD method was able to provide a near linear increase in particle image density as a function of volume flow rate (for the laminar jet) and mass (for the aquarium) as expected. The local maximum method was able to match the ABD method when the particle image density was low, however, as the density increased the slop of the particle image density estimate began to decay as a result of particle overlap. Similar results were seen by manually counting particle images. The consistent linear results obtained with the autocorrelation-based density method demonstrates its ability to account for overlapping particles.

6.2 Future Work

Future work into the development of the autocorrelation-based estimate of the particle image density may include developing a different method to estimate the average particle intensity, using an alternate image normalization technique, and refining and expanding the image parameter space.

The average particle intensity estimation used in the ABD method is based on a local maximum method. When calculating the average particle intensity no distinction is made between individual and overlapping particles. When particles overlap their intensities sum, which results in an overestimation of the average particle intensity. The development of an alternate method to estimate the average particle intensity that is able to account for particle overlap may provide more accurate results. Alternatively, the influence of the average particle intensity on the ABD method may possibly be removed or lessened through the implementation of a different image normalization method.

The normalizing method used in the image intensity estimation subtracts the lowest intensity from the entire image and then normalizes the image intensity with respect to the highest intensity values within a given interrogation region. Other methods for image normalization are available, including a min-max filtering method presented by Adrian [17]. The min-max filter is a nonlinear filter that, in effect, determines the upper and lower envelope on the image signal. The lower and upper envelops are smoothed using a uniform filter. The lower envelop is subtracted from the image and the upper envelop is used to normalize the image. As a result the image particles have a more uniform intensity level across the entire image and between image sets. The min-max method for normalizing the image particle intensity may remove the need to quantify the average particle intensity, or at least decrease the average particle intensities influence on the ABD method.

Finally, refining and expanding the image parameter space may provide the means to generate a more accurate fit capable of providing higher accuracy results over a wider range of densities. An expansion of the particle image diameter may be done; however, most PIV algorithms implement 3-point fits to estimate the sub-pixel displacements which perform poorly on particles with large image diameters.

References

[1] Wernet, M. P., "Development of Digital Particle Imaging Velocimetry for Use in Turbomachinery," *Experiments in Fluids*, Vol. 28, 2000, pp. 97–115.

[2] Lim, W. L., Chew, Y. T., Chew, T. C., and Low, H. T., "Pulsatile Flow Studies of a Porcine Bioprosthetic Aortic Valve in Vitro: PIV Measurements and Shear-Induced Blood Damage," *Journal of Biomechanics*, Vol. 34, 2001, pp. 1417–1427.

[3] Timmins, B. H., Smith, B. L., and Vlachos, P. P., "A Method for Automatic Estimation of Instantaneous Local Uncertainty in Particle Image Velocimetry Measurements," *Experiments in Fluids*, Vol. 53, 2012, pp. 1133–1147.

[4] Prasad, A. K., "Particle Image Velocimetry," *Current Science*, Vol. 79, No. 1, Jul. 10 2000, pp. 51–60.

[5] Adrian, R. J., "Particle-Imaging Techniques for Experimental Fluid-Mechanics," *Annual Review of Fluid Mechanics*, Vol. 23, 1991, pp. 261–304.

[6] Westerweel, J., "Fundamentals of Digital Particle Image Velocimetry," *Measurement Science & Technology*, Vol. 8, 1997, pp. 1379–1392.

[7] LaVision, "DaVis® version 7.2," 2006, LaVision GmbH, Göttingen, Germany.

[8] Adrian, R. J., "Twenty Years of Particle Image Velocimetry," *Experiments in Fluids*, Vol. 39, 2005, pp. 159–169.

[9] Willert, C. E. and Gharib, M., "Digital Particle Image Velocimetry," *Experiments in Fluids*, Vol. 10, No. 4, 1991, pp. 181–193.

[10] Westerweel, J., Dabiri, D., and Gharib, M., "The Effect of a Discrete Window Offset on the Accuracy of Cross-Correlation Analysis of Digital PIV Recordings," *Experiments In Fluids*, Vol. 23, No. 1, May 1997, pp. 20–28.

[11] Scarano, F., "Iterative Image Deformation Methods in PIV," *Measurement Science & Technology*, Vol. 13, No. 1, Jan. 2002, pp. R1–R19.

[12] Raffel, M., Willert, C., and Kompenhans, J., *Particle Image Velocimetry*, Springer-Verlag, Berlin Heidelberg, 1998.

[13] Adrian, R. J., "Statistical Properties of Particle Image Velocimetry Measurements in Turbulent Flow," *Laser Anemometry in Fluid Mechanics - III*, 1988, pp. 115–129.

[14] Westerweel, J., "Theoretical Analysis of the Measurement Precision in Particle Image Velocimetry," *Experiments In Fluids*, Vol. 29, 2000, pp. S3–S12.

[15] Prasad, A., Adrian, R., Landreth, C., and Offutt, P., "Effect of Resolution on the Speed and Accuracy of Particle Image Velocimetry Interrogation," *Experiments in Fluids*, Vol. 13, 1992, pp. 105–116.

[16] Fincham, A. and Spedding, G. R., "Low Cost, High Resolution DPIV for Measurement of Turbulent Fluid Flow," *Experiments in Fluids*, Vol. 23, 1997, pp. 449–462.

[17] Adrian, R. J. and Westerweel, J., *Particle Image Velocimetry*, Cambridge University Press, New York, 2011.

[18] Karri, S., Charonko, J. J., and Vlachos, P. P., "Robust Wall Gradient Estimation Using Radial Basis Functions and Proper Orthogonal Decomposition (POD) for Particle Image Velocimetry (PIV) Measured Fields," *Measurement Science & Technology*, Vol. 20, 2009, pp. 1–14.

[19] Luff, J. D., Drouillard, T., Rompage, A. M., Linne, M. A., and Hertzberg, J. R., "Experimental Uncertainties Associated with Particle Image Velocimetry (PIV) Based Vorticity Algorithms," *Experiments in Fluids*, Vol. 26, 1999, pp. 36–54.

[20] Fouras, A. and Soria, J., "Accuracy of Out-Of-Plane Vorticity Measurements Derived from In-Plane Velocity Field Data," *Experiments in Fluids*, Vol. 25, 1998, pp. 409–430.

[21] Sinha, S. K. and Kuhlman, P. S., "Investigating the Use of Stereoscopic Particle Streak Velocimetry for Estimating the Three-Dimensional Vorticity Field," *Experiments in Fluids*, Vol. 12, 1992, pp. 377–384.

[22] Spedding, G. R. and Rignot, E. J. M., "Performance Analysis and Application of Grid Interpolation Techniques for Fluid Flows," *Experiments in Fluids*, Vol. 15, 1993, pp. 417–430.

[23] Morgan, J. S., Slater, D. C., Timothy, J. G., and Jenkins, E. B., "Centroid Position Measurements and Subpixel Sensitivity Variations with the MAMA Detector," *Applied Optics*, Vol. 28, No. 6, Mar 1989, pp. 1178–1192.

[24] Lourenco, L. and Krothapalli, A., "On the Accuracy of Velocity and Vorticity Measurements with PIV," *Experiments in Fluids*, Vol. 18, 1995, pp. 421–428.

[25] Chen, J. and Katz, J., "Elimination of Peak-Locking Error in PIV Analysis Using the Correlation Mapping Method," *Measurement Science and Technology*, Vol. 16, No. 8, 2005, pp. 1605.

[26] Cowen, E. A., Monismith, S. G., Cowen, E. A., and Monismith, S. G., "A Hybrid Digital Particle Tracking Velocimetry Technique," *Experiments in Fluids*, Vol. 22, 1997, pp. 199–211.

[27] Keane, R. D. and Adrian, R. J., "Optimization of Particle Image Velocimeters. I. Double Pulsed Systems," *Measurement Science and Technology*, Vol. 1, No. 11, 1990, pp. 1202.

[28] Keane, R. D. and Adrian, R. J., "Theory of Cross-Correlation Analysis of PIV Images," *Applied Scientific Research*, Vol. 49, No. 3, 1992, pp. 191–215.

[29] Charonko, J. J. and Vlachos, P. P., "Estimation Of Uncertainty Bounds from Cross Correlation Peak Ratio for Individual PIV Measurements," ASME, Paper FEDSM2012-72475, 2012.

[30] Bolinder, J., "On the Accuracy of a Digital Particle Image Velocimetry System," Lund Institute of Technology, Technical Report ISSN 0282 - 1990, Lund, Sweden, Jun. 1999.

[31] Honkanen, M. and Nobach, H., "Background Extraction from Double-Frame PIV Images," *Experiments in Fluids*, Vol. 38, 2005, pp. 348–362.

[32] Westerweel, J., *Digital Particle Image Velocimetry: Theory and Application*, Ph.D. thesis, Delft University of Technology, 1993.

[33] Hesselink, L., "Digital Image-Processing in Flow Visualization," *Annual Review of Fluid Mechanics*, Vol. 20, 1988, pp. 421–486.

[34] Gui, L., Wolfgang, M., and Shu, J., "Evaluation of Low Image Density Recordings with the MQD Method and Application to the Flow in a Liquid Bridge," *Flow Visualization and Image Processing*, Vol. 4, 1997, pp. 333–343.

[35] Wereley, W. T., Gui, L., and Meinhart, C. D., "Advanced Algorithms for Microscale Particle Image Velocimetry," *AIAA Journal*, Vol. 40, No. 6, Jun 2002, pp. 1047–1055.

[36] Echols, W. H. and Young, J. A., "Studies of Air-Operated Aerosol Generators," US Naval Research Laboratory, NRL report 5929, Jul. 1963.

[37] Coleman, H. W. and Steele, W. G., *Experimentation, Validation, and Uncertainty Analysis for Engineers*, John Wiley and Sons, Hoboken, NJ, 3rd ed., 2009.

Appendix

This appendix contains the MATLAB code for the autocorrelation-based density method used to process data from synthetic and experimental data. A graphical user interface (GUI) is also available, but is not included.

```
%Version updated 10/9/12 PIVdiaden v.4.4a

% USE THIS CODE BY RUNNING THE GUI 'PIVdiadenGUI.m' WITH
% 'PIVdiadenGUI.fig' IN THE SAME LOCATION AS THIS CODE.
%
% TO USE THIS CODE WITHOUT THE GUI THE STRUCTURE 'Data' MUST
% BE CREATED TO RUN 'PIVdiaden(Data)'.
%{
      Data.imdirec = 'C:\Users\EFDL\Downloads\repivdiadencode';
      Data.imbase  = 'B';
      Data.imzeros = '5';
      Data.imext   = 'tif';
      Data.imcstep = '0';
      Data.imfstep = '1';
      Data.imfstart= '1';
      Data.imfend  = '6';
      Data.imNum   = '1';
      Data.DiaGS = '2';
      Data.DenGS = '2';
      Data.dens = '';
      Data.diam = '';
      Data.hardDens = '0';
      Data.hardDia = '0';
      Data.Plot = '0';
      Data.outbase = 'B00001';
      Data.direcout= 'C:\Users\EFDL\Desktop\';
      Data.par     = '0';
      Data.parprocessors = '6';
      PIVdiaden(Data)
%}

function PIVdiaden(Data)
% Uncertainty analysis tool based upon input structure "Data"
% generated using the PIVuncertainty.m and PIVuncertainty.fig GUI file.
%
% Required GUI inputs include:
%     -Image files (including DaVis *.im7 files)
% Data Structure includes:
%     -imdirec - directory where images are located
%     -imbase  - base letter(s) of image (ie. 'B' for 'B00001.im7')
%     -imzeros - number of number places in file(ie 'B00001.im7' has 5)
%     -imext   - extension on image file (ie. *.im7, *.png, etc.)
%     -imcstep - step from first image to second image
%     -imfstep - step between first and third images
%     -imfstart- first image number
%     -imfend  - last image number
%     -imNum   - chose first or second image (ie. *.im7 file)
%     -DiaGS   - Diameter Grid size (ie. 128x128, 64x64)
%     -DenGS   - Density Grid size (ie. 128x128, 64x64)
```

```
%     -dens    - Image density if hard coded (leave as '' otherwise)
%     -diam    - Image diameter if hard coded (leave as '' otherwise)
%     -hardDens- Hard code density '0' = no, '1' = yes
%     -hrdDia  - Hard code diameter '0' = no, '1' = yes
%     -Plot    - Plot results '0' = no, '1' = yes
%     -outbase - output file name (ie. B00001)
%     -outdirec- directory where results are saved
%     -par     - run in parallel
%     -parprocessors - number of processors to use
%     -bkgd    - minimum pixel value over all images
%     -imgMean - mean pixel value over all images
%     -imgStd  - pixel standard deviation over all images
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Convert Grid Size selection to pixel values
if Data.DiaGS=='1'
    Data.GSdia=32;
elseif Data.DiaGS=='2'
    Data.GSdia=64;
elseif Data.DiaGS=='3'
    Data.GSdia=128;
elseif Data.DiaGS=='4'
    Data.GSdia=256;
end

if Data.DenGS=='1'
    Data.GSden=32;
elseif Data.DenGS=='2'
    Data.GSden=64;
elseif Data.DenGS=='3'
    Data.GSden=128;
elseif Data.DenGS=='4'
    Data.GSden=256;
end

%Set grid resolution to match interrogation size
Data.gridres=Data.GSden;

if Data.bkgd =='1'
%% Calculate BackGround:
%{
    Noise is introduced through stationary objects, reflections, and a
    non-zero backgrond. The background is calculated by comparing all
    images in the set and taking the minimum value for each pixel.
%}
fprintf('\n----------- Calculating Background Image  ------------\n')
%Image file location and base (ie. C:\Folder\File-prefix)
if ispc
    imbase=[Data.imdirec '\' Data.imbase];
else
    imbase=[Data.imdirec '/' Data.imbase];
end

%Image sets
```

```
I1 = str2double(Data.imfstart):str2double(Data.imfstep):str2double(...
    Data.imfend);
I2 = I1+str2double(Data.imcstep);

if strcmp(Data.imext,'im7')==1 || strcmp(Data.imext,'IM7')==1
    %Load single image to determin image size
    Davis=readimx([imbase sprintf(['%0.' Data.imzeros 'i.' Data.imext],I1(1))...
        ]);
    im2=double(Davis.Data(:,1:Davis.Ny))';
else
    %All other image files (Same method for DaVis *im7 files [above])
    im2=double(imread([imbase sprintf(['%0.' Data.imzeros 'i.' Data.imext],I2...
        (1))]));
end

%Pre-allocate matrices for the background and for the mean/stdv
imgSum = zeros(size(im2));
imgSum2 = zeros(size(im2));
bkgd = ones([size(im2),2])*(2^16);

%Load every image pair
for q=1:length(I1)
    if strcmp(Data.imext,'im7')==1 || strcmp(Data.imext,'IM7')==1
        %DaVis Image File
        Davis=readimx([imbase sprintf(['%0.' Data.imzeros 'i.' Data.imext],I2...
            (q))]);
        %Select image (DaVis image pairs are stored in a single matrix,
        %side by side)
        if Data.imNum == '1'
            im2=double(Davis.Data(:,1:Davis.Ny))';
        elseif Data.imNum == '2'
            im2=double(Davis.Data(:,Davis.Ny+1:2*Davis.Ny))';
        elseif Data.imNum == '3'
            im2=double(Davis.Data(:,2*Davis.Ny+1:3*Davis.Ny))';
        elseif Data.imNum == '4'
            im2=double(Davis.Data(:,3*Davis.Ny+1:4*Davis.Ny))';
        end

        %Assign current image to 2nd slot in matrix
        bkgd(:,:,2) = im2;
        %Compare the current min the the new image, keep lowest value
        bkgd(:,:,1) = min(bkgd,[],3);
        %Values for mean and stdv
        imgSum = imgSum + im2;
        imgSum2 = imgSum2 + im2.^2;
    else
        %All other image files (Same as DaVis *im7 files [above])
        if Data.imNum == '1'
        im2=double(imread([imbase sprintf(['%0.' Data.imzeros 'i.' Data.imext...
            ],I1(q))]));
        elseif Data.imNum == '2'
        im2=double(imread([imbase sprintf(['%0.' Data.imzeros 'i.' Data.imext...
            ],I2(q))]));
        end
        bkgd(:,:,2) = im2;
        bkgd(:,:,1) = min(bkgd,[],3);
```

```
        imgSum = imgSum + im2;
        imgSum2 = imgSum2 + im2.^2;
    end
end
%Store background in Data structure for future use
Data.bkgrd = bkgd(:,:,1);
%Calculate and store mean
Data.imgMean = (imgSum./q);
%Calculate and store standard deviation using:
%http://en.wikipedia.org/wiki/Computational_formula_for_the_variance
Data.imgStd = sqrt(imgSum2./(q-1)-(q/(q-1))*((imgSum./q).^2));
%{
%Uncomment to make the following plots:

map = gray(2^10);
%Shows Original Image
figure(1)
imshow(im2,map)

%Shows time minimum background
figure(2)
imshow(Data.bkgrd,map)

%Shows Figure (1) without background
figure(3)
Temp2=im2-Data.bkgrd;
Temp2(Temp2<0)=0;
imshow(Temp2,map)
%}
%End Calculage Backgroud
end

%% Run Jobs in Parallel
%{
    If the "Run in Parallel" box is checked in the PIVuncertainty GUI,
    this section opens matlabpool which enables parallel processing.
    Images are devided between the total number of processors.
%}
if str2double(Data.par)
    mkdir('TempFolder')
    fprintf('\n--- Initializing Processor Cores for Parallel Job ----\n')
    poolopen=1;

    %Don't open more processors than there are image pairs
    if length(str2double(Data.imfstart):str2double(Data.imfstep):str2double(...
            Data.imfend)) < str2double(Data.parprocessors)
        Data.parprocessors=num2str(length(str2double(Data.imfstart):...
            str2double(Data.imfstep):str2double(Data.imfend)));
    end

    try
        %Open Matlab Pool
        matlabpool('open','local',Data.parprocessors);
    catch
        try
            %If First attepmt doesn't work, close and retry
```

```
            matlabpool close
            matlabpool('open','local',Data.parprocessors);
        catch
            %If Second attempt doesn't work, continue with single processor
            beep
            disp('Error Running Job in Parallel - Defaulting to Single ...
                Processor')
            poolopen=0;
            fprintf('\n-------------- Processing Dataset ------------------\n...
                ')
            Data.par = '0';
            diadenprocessing(Data)
            fprintf('---------------- Job Completed ---------------------\n')
        end
    end
    if poolopen
        %Pool successfully open, divide sets
        I1=str2double(Data.imfstart):str2double(Data.imfstep):str2double(...
            Data.imfend);
        I2=I1+str2double(Data.imcstep);

            fprintf('\n--------------- Processing Dataset ...
                -------------------\n')
            spmd
                verstr=version('-release');
                if str2double(verstr(1:4))>=2010
                    I1dist=getLocalPart(codistributed(I1,codistributor('1d'...
                        ,2)));
                    I2dist=getLocalPart(codistributed(I2,codistributor('1d'...
                        ,2)));
                else
                    I1dist=localPart(codistributed(I1,codistributor('1d',2),'...
                        convert'));
                    I2dist=localPart(codistributed(I2,codistributor('1d',2),'...
                        convert'));
                end
                diadenprocessing(Data,I1dist,I2dist);
            end
            fprintf('----------------- Job Completed ----------------------\n...
                ')

        if poolopen
            matlabpool close
        end
    end
    %Combine solution from the parallel processors
    DiadenCombine(Data)
else
    %"Run in Parallel" not selected - Default to single processor
    fprintf('\n-------------- Processing Dataset ------------------\n')
    Data.par = '0';
    diadenprocessing(Data)
    fprintf('---------------- Job Completed ---------------------\n')
end
```

```
function diadenprocessing(Data,I1,I2)

tic
% Image file location/base (ie. C:\Folder\File-prefix) and
% output directory
if ispc
    imbase=[Data.imdirec '\' Data.imbase];
    tempout = ['TempFolder\' Data.imbase];
    Data.outdirec= [Data.outdirec '\'];
else
    imbase=[Data.imdirec '/' Data.imbase];
    tempout = ['TempFolder/' Data.imbase];
    Data.outdirec= [Data.outdirec '/'];
end

if nargin<3 %(Not runing in parallel)
%Image indices
I1 = str2double(Data.imfstart):str2double(Data.imfstep):str2double(...
    Data.imfend);
I2 = I1+str2double(Data.imcstep);
end


%% EVALUATE IMAGE PAIRS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for q=1:length(I1)

    %Load image pair and flip coordinates
    if strcmp(Data.imext,'im7')==1 || strcmp(Data.imext,'IM7')==1
        %DaVis Image File
        Davis=readimx([imbase sprintf(['%0.' Data.imzeros 'i.' Data.imext],I2...
            (q))]);
        if Data.imNum == '1'
            im2=double(Davis.Data(:,1:Davis.Ny))';
        elseif Data.imNum == '2'
            im2=double(Davis.Data(:,Davis.Ny+1:2*Davis.Ny))';
        elseif Data.imNum == '3'
            im2=double(Davis.Data(:,2*Davis.Ny+1:3*Davis.Ny))';
        elseif Data.imNum == '4'
            im2=double(Davis.Data(:,3*Davis.Ny+1:4*Davis.Ny))';
        end

    else
        %All other image files
        if Data.imNum == '1'
        im2=double(imread([imbase sprintf(['%0.' Data.imzeros 'i.' Data.imext...
            ],I1(q))]));
        elseif Data.imNum == '2'
        im2=double(imread([imbase sprintf(['%0.' Data.imzeros 'i.' Data.imext...
            ],I2(q))]));
        end
    end
    %Subtract background and 20 to compensate for an low level noise
    %(Effects on the image densitsy are minimal (tipically 0.5% of max)
    if Data.bkgd =='1'
        im2 = im2-Data.bkgrd-20;
```

```
    end

    %Any negative values set to zero
    im2(im2<0)=0;

    %Flip image
    im2=flipud(im2);

    %ESTIMATE THE FOLLOWING VALUES:
    %   P_Dia       - Particle image diameter
    %   P_Dens      - Image density
    %   DiaRatio    - (dia_x/dia_y) to detect elliptical particles
    %   PeakDen     - Image density using peak counting (8-neighbor)
    %   PeakDen2    - Image density using peak counting (20-neighbors)
    %   Iavg        - Average particle intensity
    [P_Dia, P_Dens, DiaRatio, PeakDen PeakDen2 Iavg]=DiaDen(im2,Data);

    %Convert to single precision to save memory
    P_Dia=single(P_Dia);
    P_Dens=single(P_Dens);
    DiaRatio=single(DiaRatio);
    PeakDen=single(PeakDen);
    PeakDen2=single(PeakDen2);
    Iavg=single(Iavg);

    %If running in parallel, save individual files (combined ln 248)
    if str2double(Data.par)
        save([tempout sprintf(['%0.' Data.imzeros 'i.mat'],I1(q))],...
            'P_Dia','P_Dens','DiaRatio','PeakDen','PeakDen2','Iavg');
    else
    %Otherwise save results for each image to matrix
        avgI(:,:,q)=Iavg;
        Dia(:,:,q)=P_Dia;
        Den(:,:,q)=P_Dens;%/(32*32); %Convert to particles/pixel
        Rat(:,:,q)=DiaRatio;
        PkDen(:,:,q)=PeakDen;%/(32*32);
        PkDen2(:,:,q)=PeakDen2;%/(32*32);
    end
    %Output current step
    fprintf(['Completed (' num2str(q) '/' num2str(length(I1)) ')\n'])
end
%If not running in parallel, continue to stats subfunction
if Data.par == '0'
    DiaDenMeanStdev(Data,Dia,Den,Rat,PkDen,PkDen2,avgI)
end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [DIA PD R PkD PkD2 Iavg] = DiaDen(IM,Data)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Code written to estimate the particle diameter and density for an    %
% image. A minimum grid size of 64x64 should be used. Results below    %
% this grid size become significantly less accurate.                   %
```

```
%                                                                      %
%   Inputs:     IM  - the full image                                   %
%               Data- structure containing all other inputs            %
%   Output:     dia - Diameters for each interrogation region          %
%               PD  - Particle Density for each interrogation region   %
%               R   - Diameter ratio                                   %
%               PkD - Peak counting (8-neigbors)                       %
%               PkD2- Peak counting (20-neighbors)                     %
%               Iavg- Average Particle intensity                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[ysize xsize]=  size(IM);
gridres = Data.gridres;
S=[ysize xsize]./gridres;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                       Begin Diameter Estimation                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if str2double(Data.hardDia)==0
k=0;
%Generate sub-images
GS = Data.GSdia; %Grid size (pix)
inc=GS/gridres;
for i = 1:GS:floor(ysize/GS)*GS
    l=0;
    k = k+inc;
    for j = 1:GS:floor(xsize/GS)*GS
        l=l+inc;
        img=IM(i:i+(GS-1),j:j+(GS-1));
        [Diam,Ratio] = Diameter(img,GS);
        DIA(k-inc+1:k,l-inc+1:l) = Diam;
        R(k-inc+1:k,l-inc+1:l) = Ratio;
    end
end

%Caluclate diameter for portion not withing the GSxGS regions
%----------------x edge----------------
k=0;
for i = 1:GS:floor(ysize/GS)*GS
    k = k+inc;
    j = xsize-GS+1:xsize;
    img=IM(i:i+(GS-1),j);
    [Diam,Ratio] = Diameter(img,GS);
    DIA(k-inc+1:k,l+1:ceil(xsize/gridres)) = Diam;
    R(k-inc+1:k,l+1:ceil(xsize/gridres)) = Ratio;
end
%----------------y edge----------------
i = ysize-GS+1:ysize;
l=0;
for j = 1:GS:floor(xsize/GS)*GS
    l=l+inc;
    img=IM(i,j:j+(GS-1));
    [Diam,Ratio] = Diameter(img,GS);
    DIA(k+1:ceil(ysize/gridres),l-inc+1:l) = Diam;
    R(k+1:ceil(ysize/gridres),l-inc+1:l) = Ratio;
end
```

```
%-----------------corner---------------
img=IM(ysize-GS+1:ysize,xsize-GS+1:xsize);
[Diam,Ratio] = Diameter(img,GS);
DIA(k+1:ceil(ysize/gridres),l+1:ceil(xsize/gridres)) = Diam;
R(k+1:ceil(ysize/gridres),l+1:ceil(xsize/gridres)) = Ratio;

%--------------------------------------

else %Hard-Code box is checked
   DIA=str2double(Data.diam)*ones(ceil(S(1)),ceil(S(2)));
   R=ones(ceil(S(1)),ceil(S(2)));
end
DIA(isnan(DIA))=0;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%                    Estimate Density                       %%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if str2double(Data.hardDens)==0

GS =Data.GSden;
inc=GS/gridres;
k=0;

for i = 1:GS:floor(ysize/GS)*GS
    l=0;
    k = k+inc;
    for j = 1:GS:floor(xsize/GS)*GS
        l=l+inc;
        % Inerrogation region size (GS x GS)
        IA1=IM(i:i+(GS-1),j:j+(GS-1));
        % Particle diameter for region
        Dia = mean(mean(DIA(k-inc+1:k,l-inc+1:l)));
        % Calculate the Density
        [Dens pkDen pkDen2 avgI] = Density(IA1, Dia, GS,pkcut,thresh);
        % Store Solutions
        pd(k-inc+1:k,l-inc+1:l) = Dens;
        pkd(k-inc+1:k,l-inc+1:l) =pkDen;
        pkd2(k-inc+1:k,l-inc+1:l) =pkDen2;
        Iavg(k-inc+1:k,l-inc+1:l) =avgI;
    end
end

%-----------------x edge---------------
    k=0;
for i = 1:GS:floor(ysize/GS)*GS
    k = k+inc;
    IA1=IM(i:i+(GS-1),xsize-GS+1:xsize);
    Dia = mean(mean(DIA(k-inc+1:k,ceil(xsize/gridres)...
        -inc+1:ceil(xsize/gridres))));
    [Dens pkDen pkDen2 avgI] = Density(IA1, Dia, GS,pkcut,thresh);
    pd(k-inc+1:k,l+1:ceil(xsize/gridres)) = Dens;
    pkd(k-inc+1:k,l+1:ceil(xsize/gridres)) = pkDen;
    pkd2(k-inc+1:k,l+1:ceil(xsize/gridres)) = pkDen2;
```

```
    Iavg(k-inc+1:k,l+1:ceil(xsize/gridres)) = avgI;
end

%-----------------y edge----------------
    l=0;
for j = 1:GS:floor(xsize/GS)*GS
    l=l+inc;
    IA1=IM(ysize-GS+1:ysize,j:j+(GS-1));
    Dia = mean(mean(DIA(ceil(ysize/gridres)...
        -inc+1:ceil(ysize/gridres),l-inc+1:l)));
    [Dens pkDen pkDen2 avgI] = Density(IA1, Dia, GS,pkcut,thresh);
    pd(k+1:ceil(ysize/gridres),l-inc+1:l) = Dens;
    pkd(k+1:ceil(ysize/gridres),l-inc+1:l) = pkDen;
    pkd2(k+1:ceil(ysize/gridres),l-inc+1:l) = pkDen2;
    Iavg(k+1:ceil(ysize/gridres),l-inc+1:l) = avgI;
end

%-----------------corner----------------
IA1=IM(ysize-GS+1:ysize,xsize-GS+1:xsize);
Dia = mean(mean(DIA(ceil(ysize/gridres)-inc+1:ceil(ysize/gridres),...
    ceil(xsize/gridres)-inc+1:ceil(xsize/gridres))));
[Dens pkDen pkDen2 avgI] = Density(IA1,Dia,GS,pkcut,thresh);
pd(k+1:ceil(ysize/gridres),l+1:ceil(xsize/gridres)) = Dens;
pkd(k+1:ceil(ysize/gridres),l+1:ceil(xsize/gridres)) = pkDen;
pkd2(k+1:ceil(ysize/gridres),l+1:ceil(xsize/gridres)) = pkDen2;
Iavg(k+1:ceil(ysize/gridres),l+1:ceil(xsize/gridres)) = avgI;
%---------------------------------------
pd(pd<0)=0;
pkd(pd<0)=0;
pkd2(pd<0)=0;
PD=pd;
PkD=pkd;
PkD2=pkd2;
else %Hard Code box is checked
   PD=str2double(Data.dens)*ones(ceil(S(1)),ceil(S(2)));
   PkD=str2double(Data.dens)*ones(ceil(S(1)),ceil(S(2)));
   PkD2=str2double(Data.dens)*ones(ceil(S(1)),ceil(S(2)));
   Iavg=ones(ceil(S(1)),ceil(S(2)));
end

%(un)comment to supress or plot images of diameter/density distribtions
%{
figure(10);surf(DIA,'DisplayName','DIA','edgecolor','none');
view(0,90);axis square;colorbar;title('Particle Image Diameter')

figure(11); surf(PD,'DisplayName','pd','edgecolor','none');
view(0,90);axis square;colorbar;
title('Particle Density')

figure(12);map = gray(2^8);imshow(flipud(IM),map);title('Raw Image')
%}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [Diam,Ratio] = Diameter(img,GS)
% The relationship between the particle image diameter and the width of
% the correlation peak is discussed in Adrian (8.126)
%Subtract the minimum value of the image
img=img-min(img(:));
%Auto-correlate
A=fftn(img);
C2=real(ifftn(conj(A).*A));
%Correlation Peak at Center of image
C2(1:GS/2,1:GS/2)=rot90(C2(1:GS/2,1:GS/2),2);
C2(1:GS/2,GS/2+1:GS)=rot90(C2(1:GS/2,GS/2+1:GS),2);
C2(GS/2+1:GS,1:GS/2)=rot90(C2(GS/2+1:GS,1:GS/2),2);
C2(GS/2+1:GS,GS/2+1:GS)=rot90(C2(GS/2+1:GS,GS/2+1:GS),2);
if max(C2(:))>0
    %Locate Correlation Maximum
    C2=C2-min(C2(:));
    [ymaxval yind] = max(C2,[],1);
    [~, xind] = max(ymaxval);
    yind = yind(xind);
    x=0.5:size(C2,1)-0.5;
    %Number of points used in Gauss Fit (N = val*2+1)
    val = 1;
    xData=x(yind-val:yind+val);
    yData=C2(yind-val:yind+val,xind);
    %Gauss Fit
      % [fitresult, ~] = fit( xData', yData, 'gauss1');
      % Diam = 2*fitresult.c1;
    %The function "fit" above requires curve fitting toolbox.
    %Alternate method:
    p=polyfit(xData',log(yData),2);
    sigma=sqrt(-1/(2*p(1)));
    %mu=p(2)*sigma^2; A=exp(p(3)+mu^2/(2*sigma^2));
    %Particle image diameter in y-direction
    DiamY = 2*sqrt(2)*sigma;

    %In the Y direction
    y=0.5:size(C2,2)-0.5;
    yData2 = y(xind-val:xind+val);
    xData2 = C2(yind,xind-val:xind+val);
    p2=polyfit(yData2,log(xData2),2);
    sigma2=sqrt(-1/(2*p2(1)));
    %Particle image diameter in x-direction
    DiamX = 2*sqrt(2)*sigma2;
    %Average particle image diameter
    Diam = (DiamX+DiamY)/2;
    %Calculate the diameter ratio
    if DiamY~=0
        Ratio=DiamX/DiamY;
    else
        Ratio=0;
    end

else
    Diam = 0;  Ratio = 0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Dens pkDen pkDen2 avgI] = Density(IA1, Dia, GS)
%If particle counting has (dens < pkcut), the value for particle
%ct is used
pkcut=3;
%(image intensities < thresh) are set to 0 (accounts for noise)
thresh=1;

IA3=IA1;                              % Store unalterd image
IA1=sqrt(IA1);                        % Makes for better solution

%Locate peaks
BWI = Peak8(IA3);%imregionalmax(IA1);
PeaksI = IA3.*BWI;
peaksI = PeaksI(PeaksI~=0);
%Take the average and standard deviation of peaks (for normalizing)
if isempty(peaksI)
imavg = 0;
imstd = 0;
else
imavg = mean(peaksI);
imstd = std(peaksI);
end
%Values to normailze interrogation region
IMmin = sqrt(min(IA3(:)));
IMmax = sqrt(imavg+4*imstd);

%If the above value is larger than the maximum value in the image
if IMmax > max(IA1(:))
    IMmax = max(IA1(:));
end

IA1=IA1-IMmin;                        % Normalize image
IA1 = IA1*255/(IMmax-IMmin);          % Normalize image
IA1(isinf(IA1)|isnan(IA1))=0;
IA1(IA1>255)=255;                     % Any value above IMmax = IMmax
%-----------------------------------------------------------
%Autocorrelate
A=fftn(IA1);
C3=real(ifftn(conj(A).*A));
%Correlation Peak at Center of image
C3(1:GS/2,1:GS/2)=rot90(C3(1:GS/2,1:GS/2),2);
C3(1:GS/2,GS/2+1:GS)=rot90(C3(1:GS/2,GS/2+1:GS),2);
C3(GS/2+1:GS,1:GS/2)=rot90(C3(GS/2+1:GS,1:GS/2),2);
C3(GS/2+1:GS,GS/2+1:GS)=rot90(C3(GS/2+1:GS,GS/2+1:GS),2);
%-----------------------------------------------------------
%Count particles by finding local maximums
IA4 = IA1;
IA4(IA4<thresh)=0;
BW2 = Peak8(IA4);
Peaks2 = IA4.*BW2;
peaks2 = Peaks2(Peaks2~=0);
pk2=length(peaks2);
```

```
pkDen =(32*32)*(pk2)/((length(IA1)-2)^2);

%Consider using peaks2 from above instead of peaks for avg intensity
BW = Peak(IA4);
Peaks = IA4.*BW;
peaks = Peaks(Peaks~=0);
pk3=length(peaks);
pkDen2 =(32*32)*(pk3)/((length(IA1)-4)^2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AVERAGE INTENSITY                                                       %
%    Calculate average intensity by taking peak locations and stdev of%
%    four points around them. If stdev is low then particle images is %
%    centered on a pixel and give a better representation of actual   %
%    particle peak intensity                                          %
%-------------------------------------------------------------------------%
    %Pre-allocate
    I=zeros([size(IA1) 4]);
    %Create 3D matrix with 3rd dimension containing the four
    %surrounding points. (ie for pixel IA1(2,2) in image I(2,2,:)
    %contains: IA1(1,2) IA1(3,2) IA1(2,1) IA1(2,3)
    I(:,2:GS,1)=IA3(:,1:GS-1);
    I(:,1:GS-1,2)=IA3(:,2:GS);
    I(2:GS,:,3)=IA3(1:GS-1,:);
    I(1:GS-1,:,4)=IA3(2:GS,:);
    %Only keep values where peaks are located
    I(:,:,1)=(I(:,:,1)./IA3).*BW;
    I(:,:,2)=(I(:,:,2)./IA3).*BW;
    I(:,:,3)=(I(:,:,3)./IA3).*BW;
    I(:,:,4)=(I(:,:,4)./IA3).*BW;
    %Remove any Nan
    I(isnan(I))=0;
    %Compute standard deviation of four values surronding peaks
    Var=std(I,0,3);
    %Calculate the max of the four values surrounding peaks
    Imax=max(I,[],3);
    %Create mask
    Tmp1 = ones(size(IA1));
    %Remove saturate particles (ie peak value = surround value)
    Tmp1(IA3==Imax)=0;
    %Remove peaks where surround values have large standard deviations
    Tmp1(Var>=0.34)=0;
    %Remove values where there are no peaks
    Tmp1(BW==0)=0;
    %Apply mask
    Pks = Tmp1.*IA1;
    %Collect peak values
    Temp=Pks(Pks~=0);

    %Average Intensity:
    if size(Temp,1)>1
    avgI = mean(Temp);
    elseif length(peaks)>=1
        pks=peaks(peaks>=25);
    avgI = mean(pks);
    else
```

```
    avgI = max(IA3(:));
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Correlation Peak Value
CorrPeak=max(C3(:));

%Parameters for empirical density obtained through Least Squares fit
% Goodness of fit:
%   R-square: 0.9999

a= 12.440818554697394;
b=  1.252907533572718;
c= -2.017253413241566;
d= -1.268448437457117;
e= -1.271518390712386;

%Calculate image density
Dens=a*(CorrPeak^b)*(Dia^c)*((GS*GS)^d)*(avgI^e);

%If density is below 3 particles, resort to peak counting.
if pkDen < pkcut
    Dens = pkDen;
end
Dens(isnan(Dens))=0;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Peak Finder                                                         %
%   This function locates local maximums by comparing each location  %
%   (i,j) to the surrounding 20 points                                %
%                            _ _ _                                    %
%                          _|_|_|_|_                                  %
%                         |_|_|_|_|_|                                 %
%                       i |_|_|x|_|_|                                 %
%                         |_|_|_|_|_|                                 %
%                           |_|_|_|                                   %
%                              j                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Matrx = Peak(IMG)

ny = size(IMG,2);
nx = size(IMG,1);
Matrx = zeros(size(IMG));
for i = 3:nx-2
    for j = 3:ny-2
        if         IMG(i,j)>IMG(i,j-2)  && IMG(i,j)> IMG(i,j-1)...
                && IMG(i,j)>IMG(i,j+1)  && IMG(i,j)> IMG(i,j+2)...
                && IMG(i,j)>IMG(i-1,j-1) && IMG(i,j)> IMG(i-1,j)...
                && IMG(i,j)>IMG(i-1,j+1) && IMG(i,j)> IMG(i-2,j)...
                && IMG(i,j)>IMG(i+1,j-1) && IMG(i,j)> IMG(i+1,j)...
                && IMG(i,j)>IMG(i+1,j+1) && IMG(i,j)> IMG(i+2,j)...
                && IMG(i,j)>IMG(i-1,j-2) && IMG(i,j)> IMG(i+1,j-2)...
                && IMG(i,j)>IMG(i+2,j-1) && IMG(i,j)> IMG(i+2,j+1)...
                && IMG(i,j)> IMG(i-1,j+2)&& IMG(i,j)>IMG(i+1,j+2)...
```

```
                && IMG(i,j)> IMG(i-2,j+1)&& IMG(i,j)> IMG(i-2,j-1)
            Matrx(i,j) = 1;
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Peak Finder 8                                                       %
%   This function locates local maximums by comparing each location   %
%   (i,j) to the surrounding 8 points                                 %
%                              - - -                                  %
%                             |-|-|-|                                 %
%                           i |-|x|-|                                 %
%                             |-|-|-|                                 %
%                                j                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Matrx = Peak8(IMG)
ny = size(IMG,2);
nx = size(IMG,1);
Matrx = zeros(size(IMG));
for i = 2:nx-1
    for j = 2:ny-1
        if         IMG(i,j)> IMG(i,j-1) && IMG(i,j)> IMG(i,j+1)...
                && IMG(i,j)> IMG(i-1,j-1) && IMG(i,j)> IMG(i-1,j)...
                && IMG(i,j)> IMG(i-1,j+1) && IMG(i,j)> IMG(i+1,j-1)...
                &&IMG(i,j)> IMG(i+1,j) && IMG(i,j)> IMG(i+1,j+1)
            Matrx(i,j) = 1;
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% When code PIVdiaden is run in parallel individual image solutions   %
% are stored and then read in by this function. The individual results%
% are stored in a temporary file that is deleted at the end of this   %
% function.                                                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function DiadenCombine(Data)

%Determine PC or Mac
if ispc
    tempout = ['TempFolder\' Data.imbase];
    Data.outdirec= [Data.outdirec '\'];
else
    tempout = ['TempFolder/' Data.imbase];
    Data.outdirec= [Data.outdirec '/'];
end

%Number of Images processed
```

```matlab
N = length(str2double(Data.imfstart):str2double(Data.imfstep):str2double(...
    Data.imfend));

%Open first mat file to preallocate size
fname = [tempout sprintf(['%0.' Data.imzeros 'i.mat'],str2double(...
    Data.imfstart))];
load(fname)
%Preallocate matrices
Dia = zeros(size(P_Dia,1),size(P_Dia,2),N,'single');
Den = zeros(size(P_Dens,1),size(P_Dens,2),N,'single');
Rat = zeros(size(P_Dia,1),size(P_Dia,2),N,'single');
PkD = zeros(size(P_Dens,1),size(P_Dens,2),N,'single');
PkD2 = zeros(size(P_Dens,1),size(P_Dens,2),N,'single');
avgI = zeros(size(P_Dens,1),size(P_Dens,2),N,'single');
%Load all *.mat files and store in three dimensional array
n = 1;
for i = str2double(Data.imfstart):str2double(Data.imfstep):str2double(...
    Data.imfend)
    fname = [tempout sprintf(['%0.' Data.imzeros 'i.mat'],i)];
    load(fname)
    Dia(:,:,n) = P_Dia;
    Den(:,:,n) = P_Dens;
    Rat(:,:,n) = DiaRatio;
    PkD(:,:,n) = PeakDen;
    PkD2(:,:,n)= PeakDen2;
    avgI(:,:,n)= Iavg;
    n = n + 1;
end
%Calculate Mean and Standard deviation
DiaDenMeanStdev(Data,Dia,Den,Rat,PkD,PkD2,avgI)
%Delete Temporary Folder
rmdir('TempFolder','s')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This portion calculates the mean and standard deviation of the      %
% entire image set. Used for both parallel and serial cases.          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function DiaDenMeanStdev(Data,Dia,Den,Rat,PkD,PkD2,avgI)
%Remove NaNs
Dia(isnan(Dia))=0;
Den(isnan(Den))=0;
Rat(isnan(Rat))=0;
PkD(isnan(PkD))=0;
PkD2(isnan(PkD2))=0;
avgI(isnan(avgI))=0;
%Calculate Mean and Standard Deviation
meanDia = mean(Dia,3);
meanDen = mean(Den,3);
meanRat = mean(Rat,3);
meanPkD = mean(PkD,3);
meanPkD2 = mean(PkD2,3);
meanIavg = mean(avgI,3);
```

```
sigDen=std(Den,0,3);
sigDia=std(Dia,0,3);
sigRat=std(Rat,0,3);
sigPkD=std(PkD,0,3);
sigPkD2=std(PkD2,0,3);
%Reduce solution to single value per interrogation region
ii = size(Dia,1);
jj = size(Dia,2);
VarDia = Data.GSdia/Data.gridres;
stdDia = sigDia(1:VarDia:ii,1:VarDia:jj);
muDia  = meanDia(1:VarDia:ii,1:VarDia:jj);
Diam   = Dia(1:VarDia:ii,1:VarDia:jj,:);
stdRat = sigRat(1:VarDia:ii,1:VarDia:jj);
muRat  = meanRat(1:VarDia:ii,1:VarDia:jj);
Ratio  = Rat(1:VarDia:ii,1:VarDia:jj,:);

ii = size(Den,1);
jj = size(Den,2);
VarDen = Data.GSden/Data.gridres;
muDen  = meanDen(1:VarDen:ii,1:VarDen:jj);
stdDen = sigDen(1:VarDen:ii,1:VarDen:jj);
Dens   = Den(1:VarDen:ii,1:VarDen:jj,:);
muPkD  = meanPkD(1:VarDen:ii,1:VarDen:jj);
stdPkD = sigPkD(1:VarDen:ii,1:VarDen:jj);
muPkD2  = meanPkD2(1:VarDen:ii,1:VarDen:jj);
stdPkD2 = sigPkD2(1:VarDen:ii,1:VarDen:jj);
PkDen   = PkD(1:VarDen:ii,1:VarDen:jj,:);
PkDen2  = PkD2(1:VarDen:ii,1:VarDen:jj,:);
muIavg  = meanIavg(1:VarDen:ii,1:VarDen:jj);
Iavg    = avgI(1:VarDen:ii,1:VarDen:jj);
%Save Data to specified location
save([Data.outdirec Data.outbase],'Diam','Dens','Ratio','muDia',...
    'muDen','muRat','stdDia','stdDen','stdRat','muPkD','stdPkD',...
    'PkDen','muPkD2','stdPkD2','PkDen2','Iavg');

%Plot if desired
if Data.Plot == '1'
figure(100);
surf(double(meanDia),'DisplayName','DIA','edgecolor','none');
view(0,90);axis square;colorbar;title('Particle Image Diameter')
figure(101);
surf(double(meanDen),'DisplayName','pd','edgecolor','none');
view(0,90);axis square;colorbar;
title('Particle Density')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```