

3D Particle Tracking Velocimetry Method: Advances and Error Analysis

N F Ponchaut, C A Mouton, H G Hornung, and D Dabiri*

Graduate Aeronautical Laboratories
California Institute of Technology
Pasadena, CA 91125

GALCIT Report FM2005.004

August 2005

*University of Washington, Seattle, WA 98195, USA

Abstract

A full three-dimensional particle tracking system was developed and tested. By using three separate CCDs placed at the vertices of an equilateral triangle, the three-dimensional location of particles can be determined. Particle locations measured at two different times can then be used to create a three-component, three-dimensional velocity field. Key developments are: the ability to accurately process overlapping particle images, offset CCDs to significantly improve effective resolution, allowance for dim particle images, and a hybrid particle tracking technique ideal for three-dimensional flows when only two sets of images exist. An in-depth theoretical error analysis was performed which gives the important sources of error and their effect on the overall system. This error analysis was verified through a series of experiments, which utilized a test target with 100 small dots per square inch. For displacements of 2.54 mm the mean errors were less than 2% and the 90% confidence limits were less than $5.2\ \mu\text{m}$ in the plane perpendicular to the camera axis, and $66\ \mu\text{m}$ in the direction of the camera axis. The system was used for flow measurements around a delta wing at an angle of attack. These measurements show the successful implementation of the system for three-dimensional flow velocimetry.

Contents

1	Introduction	4
2	Principle	5
2.1	3DPTV Geometric Description	7
2.1.1	Alignment	7
2.1.2	2D Projections	7
2.1.3	CCD Offsets	8
2.2	Current Camera Configuration	8
3	Data Processing Program	8
3.1	Image Pre-Processing	9
3.2	Peak Search	9
3.3	Calibration	12
3.4	Epipolar Search and Ray Tracing	13
3.5	Peak Level	14
3.6	Particle Tracking	14
4	Theoretical Error Analysis	15
4.1	Calibration Error	15
4.2	Peak Searching Error	16
4.3	Combined Error	18
5	Results	18
5.1	Peak Searching	19
5.2	Experimental Verification	20
5.3	Delta Wing Experiment	26
6	Conclusions and Recommendations	26
7	Acknowledgments	30

1 Introduction

The ability to accurately map fluid velocity fields in three-dimensions would provide researchers with a much greater understanding of a wide variety of flows. A particularly promising technique for velocity measurements is particle velocimetry. With this technique, images of particles are taken at different times and based their displacements and the time separation, the velocity of these particles can be calculated. Unfortunately, three-component three-dimensional velocimetry systems have lagged substantially behind their two-dimensional counterparts. To remedy this situation, some researchers have developed methods using a single charge coupled device (CCD) [1] or several CCDs (generally 3 or 4) using either the epipolar technique [2, 3, 4, 5] or ray tracing [6, 7] for finding corresponding peaks of intensity in particle images. Others are using a holographic system, which although complicated, appears promising [8, 9]. Of particular interest is the work done by Pereira *et al* [3, 10] in developing a digital defocusing particle image velocimetry (DDPIV) system. Their system uses three individual CCDs placed at the vertices of an equilateral triangle, with all three CCDs having overlapping fields of view. The test section can be placed anywhere in this overlap region. An image of every particle in the test section is projected onto each CCD. When the CCD images are overlayed, the three images of a particle will be located at the three corners of an equilateral triangle. One thing that sets DDPIV apart from many other techniques is that the entire camera system is encased in a single unit and can therefore be moved from facility to facility without any need for re-alignment or re-calibration. The DDPIV system uses the size and position of the image triangle to determine the three-dimensional position of the corresponding particle. Sub-pixel accuracy of a particle image location is obtained by having the particle image span more than one pixel. With sub-pixel accuracy the overall resolution of the system is drastically improved.

The current authors first revisited the design choices made by Pereira *et al* and then improved upon some of them. Also, new and robust software package was developed. The software written by the current authors is significantly different and improved from that of Pereira *et al*, and is discussed in detail in section 3. In particular, a ray tracing rather than a defocusing algorithm is implemented in order to find the three-dimensional position of the particles. These improvements provide increased accuracy of results. Furthermore, while Pereira *et al* used a three-dimensional correlation over voxels, the current authors pursue direct particle tracking using a hybrid technique. The system developed by the current authors is referred to as three-dimensional particle triangulation velocimetry (3DPTV) [5, 11].

It is essential for any experiment to have an estimate of the errors that can arise from the system, both from the hardware and from the software. This article will present both theoretically and experimentally the errors one can expect when using 3DPTV in sections 4 and 5. First, a thorough theoretical error analysis will be performed, followed by a comparison of these error estimates with experimental results.

Lastly, in section 5, are the results of experimental measurements of the flow around a delta wing at an angle of attack. The results show the ability of the 3DPTV system to make measurements of interesting flows with a relatively large range of velocities.

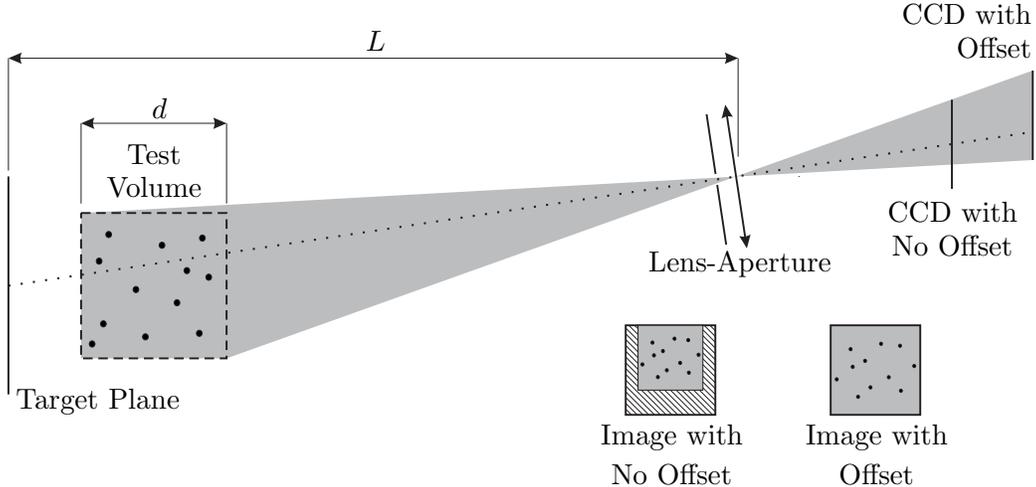


Figure 1: Maximum CCD utilization using offsets. The dashed out area of the CCD image with no offset shows the area of the CCD not utilized.

2 Principle

Both the hardware setup of Pereira *et al* [3, 10] for DDPIV and the hardware setup of 3DPTV consist of a camera comprising three separate CCDs, which all view the same test volume from different angles. The front plate of the camera consists of a mask, three lenses, and three apertures. Behind each lens-aperture combination, there is a CCD. The need for more than two CCDs and suggestion that with three cameras, an equilateral triangle is the optimum configuration, is discussed by Maas *et al* [2]. The setup of Pereira *et al* is such that the center of a target plate is projected onto the center of each CCD. Figure 1 shows this for one of the CCDs. In this arrangement, the image of the test volume does not fill the frame of the CCD; this is clearly sub-optimal. In order to make full use of all the pixels of the CCD, 3DPTV uses offsets, in which the CCDs are moved away from the lens and away from the camera axis as shown in figure 1. Since with these offsets, the image of the test volume covers more pixels, the system will have a higher overall accuracy and improved resolution.

The purpose of the lens in 3DPTV is to give an acceptable range of particle image blur; therefore, different focal lengths are used for the offset and no offset configurations. Since 3DPTV does not rely on the defocusing principle, the restriction that the target plane be in focus, as with DDPIV [3, 10], is not required.

It is important to note that the lens is not parallel to the target plane or to the CCDs. The reason for that is to avoid lens aberrations as much as possible. Future camera designs should have the lens axis intersect the center of the test volume to further reduce coma aberration in the test volume.

In practice it is impossible to place the CCDs at their exact theoretical locations; therefore, careful alignment followed by calibration is required. In order to do this, a well designed target is needed in addition to calibration software. The current authors designed a target with 100 equally spaced dots per square inch. This target is placed at the target plane, the CCDs should be aligned as carefully as possible. The calibration software, discussed further in section 3.3, then compares the measured and known dot locations and creates a mapping to correct for errors in CCD placement as well as lens aberrations and other effects. The benefits of this calibration technique is the simplicity and speed at which

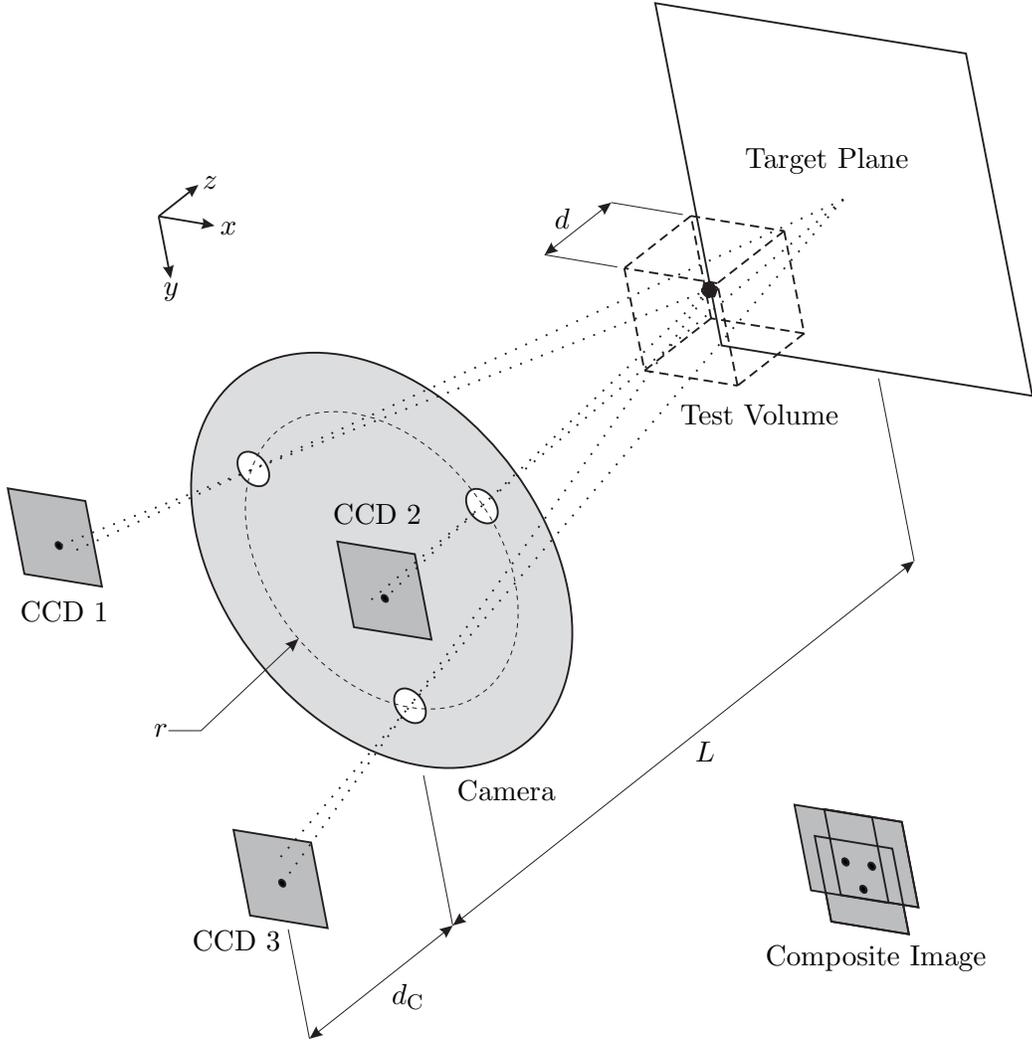


Figure 2: Projection of a particle onto the CCDs. The composite image is formed by overlaying the CCD images with correct offsets. The origin of coordinates is at the centroid of the three lenses.

an accurate calibration can be made.

Once the camera has been aligned and calibrated, the corrected images of a particle in the test volume will appear at the vertices of an equilateral triangle on the overlaid CCD images. The formation of triangles on the overlaid images is shown in figure 2. It is only a matter of ray tracing, as discussed in section 3.4, to transform a triangle on the CCDs to a particle location.

Particle tracking is done using a hybrid technique. Once all of the three-dimensional positions are known, all that is required to determine the three components of the velocity of the particles, is to find corresponding particles in two successive frames separated by a known time interval. This is discussed in detail in section 3.6. The remainder of this section will present the geometry of the 3DPTV system and the author's current camera system.

2.1 3DPTV Geometric Description

The design of the 3DPTV system involves many related parameters. Choosing the best parameters for a set of particular requirements is truly a design exercise. The governing equations used in this design are presented in this section. These equations suppose that the medium surrounding the camera has a constant refractive index.

2.1.1 Alignment

As explained before, the camera is composed of three CCDs placed at the vertices of an equilateral triangle. The distance between the CCDs and the plane of the lenses, d_C , and placement of each CCD are chosen so that the projection of the test volume fills the entire CCD. Let the coordinates of the projection of the center of the target plane onto the i^{th} CCD be \mathbf{P}_i^c . Each of the CCDs views the test volume through its own lens, which is located at \mathbf{P}_i^l . Note that the focal length of the lenses does not enter into any equations presented in this paper and there is no constraint requiring that the target plane be in focus. The focal length of the lenses will only change the degree to which particle images are blurred on the CCDs and therefore should be chosen so that images of particles within the test volume appear on the CCDs with an acceptable range of blur radii. Three-dimensional vector positions \mathbf{P} are measured with respect to the geometric center of the three lenses.

One possible lens configuration is

$$\mathbf{P}_1^l = (0, r, 0), \quad (1)$$

$$\mathbf{P}_2^l = \left(\frac{\sqrt{3}r}{2}, \frac{-r}{2}, 0 \right), \quad (2)$$

$$\mathbf{P}_3^l = \left(-\frac{\sqrt{3}r}{2}, \frac{-r}{2}, 0 \right), \quad (3)$$

where r is the radius of the circle on which the centers of the lenses lie. If the camera is perfectly aligned, the position of the projections of the target plane center onto the CCDs for this lens configuration will be

$$\mathbf{P}_1^c = \left(0, \frac{r}{L} (L + d_C), -d_C \right), \quad (4)$$

$$\mathbf{P}_2^c = \left(\frac{\sqrt{3}}{2L} (L + d_C), -\frac{r}{2L} (L + d_C), -d_C \right), \quad (5)$$

$$\mathbf{P}_3^c = \left(-\frac{\sqrt{3}}{2L} (L + d_C), -\frac{r}{2L} (L + d_C), -d_C \right). \quad (6)$$

2.1.2 2D Projections

Suppose that a particle is at position (x, y, z) with respect to the geometric center of the three lenses. Using geometric optics it is possible to find the pixel coordinates, \mathbf{I}_i , of the

particle image on CCD i , given \mathbf{P}^l and \mathbf{P}^c :

$$\mathbf{I}_1 = \left(-\frac{x}{S_z}d_C, \frac{r(L-z) - Ly}{S_zL}d_C \right), \quad (7)$$

$$\mathbf{I}_2 = \left(\frac{-2Lx + \sqrt{3}r(L-z)}{2S_zL}d_C, \frac{-r(L-z) - 2Ly}{2S_zL}d_C \right), \quad (8)$$

$$\mathbf{I}_3 = \left(\frac{-2Lx - \sqrt{3}r(L-z)}{2S_zL}d_C, \frac{-r(L-z) - 2Ly}{2S_zL}d_C \right), \quad (9)$$

where S is side length of a square pixel. \mathbf{I} is measured in pixels in the plane of each individual CCD with respect to the point where the center of the target plane is projected onto the respective CCD. Using these coordinates, the side length, s , of the equilateral triangle formed by corresponding particle images on the overlaid CCD images is given by

$$s = \frac{\sqrt{3}r(L-z)}{S_zL}d_C. \quad (10)$$

2.1.3 CCD Offsets

Offsetting the CCDs significantly improves the accuracy of the 3DPTV system. The increase in the useful area can be calculated by looking at the additional area covered in the offset configuration, as seen in figure 1. This calculation will produce a ratio of useful CCD surface area between the systems with and without offsets that is given by

$$\frac{s_{\text{offset}}}{s_{\text{nooffset}}} = \left(\frac{2[Ld + 2r(L-z)](d+z)}{Ld[d + 2(r+z)]} \right)^2. \quad (11)$$

This formula supposes that the test volume is a cube with side-length d and it assumes a limited range of values. For other value ranges similar formulas can be found. Note that by offsetting the CCDs, the projection of the center of the target plane onto the CCDs no longer corresponds to the middle pixels, as is seen in figure 2. Clearly, this increased useful area will result in an improved accuracy over that without offsets.

2.2 Current Camera Configuration

The current camera is designed to investigate a $50 \times 50 \times 50$ mm³ test volume. Each sensor is composed of 1024×1024 px, each pixel is square with a side length of $6.45 \mu\text{m}$. Three lenses are placed at the corners of an equilateral triangle inscribed in a circle of radius of 42.5 mm. The target plate used for alignment was placed 625 mm from the camera. The z -range of the test volume is 517 mm to 567 mm from the camera. Equation 11 shows that because of offsets there is a 58% increase in the number of pixels used relative to the setup without the offsets.

3 Data Processing Program

The data processing program was written to convert a set of three images into 3D positions and when required, to compute a velocity field from these results. To achieve this, several computational routines are performed. First, the images are pre-processed to remove noise.

Next, a peak searching algorithm looks for peaks in each image by performing a least-squares curve fit. This least-squares fitting allows for both overlapping particle images and sub-pixel accuracy. Once the peaks in each image have been found, an epipolar search is employed to find corresponding peaks. The locations of the peaks are then used to calculate the three-dimensional particle position by ray tracing; whereas, DDPIV uses the defocusing principle [3, 10]. With particle positions known in two successive frames, particle tracking can be performed.

3.1 Image Pre-Processing

The images have both systematic noise, that is to say noise at specific spatial frequencies, and random noise. To remove the systematic noise, the images are Fourier transformed, spatial frequencies of noise inherent to the camera system are removed, after which an inverse Fourier transform is performed. From this point on, the images are stored as a matrix of double precision numbers so that minimum additional discretization error is introduced. In order to remove random noise, a smoothing filter Gaussian blurs the images. By performing a Gaussian blur a particle image with a Gaussian distribution will remain Gaussian; whereas, the random noise will be reduced. Because of the details of the current 3DPTV program the additional Gaussian blur reduces the error associated with the algorithm, this is discussed further in section 3.2.

3.2 Peak Search

The two most critical parts of the 3DPTV program are the peak search, which finds the location of each particle image, and the calibration subroutines. In fact, since they are the only two subroutines that deal with actual images and noise, the accuracy of the results depends on the robustness of these two subroutines. Searching for peaks can be very time consuming when the particle density is high. Because of the computation time and importance of this subroutine, it must be both robust and efficient.

When investigating a three-dimensional volume, the total number of illuminated particles should be high. However, as more particles are introduced into the volume, finding individual particles becomes increasingly difficult, especially when the particle images are blurred. Three possible solutions to this problem are listed here. All three involve surface fitting of a function in order to obtain sub-pixel accuracy. One solution to this problem is to reduce the total number of particles to a point where virtually no particle images overlap, in which case all that is required is to find all the local maxima and fit a single surface around each maximum. This solution gives excellent image location accuracy, but produces only a small number of particle images. A second solution is to have a relatively high number of particles, while still just finding local maxima and fitting a single surface around each maximum. This solution essentially ignores particle image overlapping and therefore results in some particles being located with excellent accuracy, while others are located with virtually no accuracy; however, this provides for a larger number of total particle images than the first solution. The third solution, which is implemented by the current authors, is to have a relatively high number of particles, but to not rely on local maxima. This solution calls for a least-squares optimization of a number of surfaces and provides very good accuracy for a large number of particles. The reason that a least-squares optimization is the best solution is that when the particle density is high, two distinct particle images often overlap. In this case, two particle images do not necessarily create two local maxima and the local

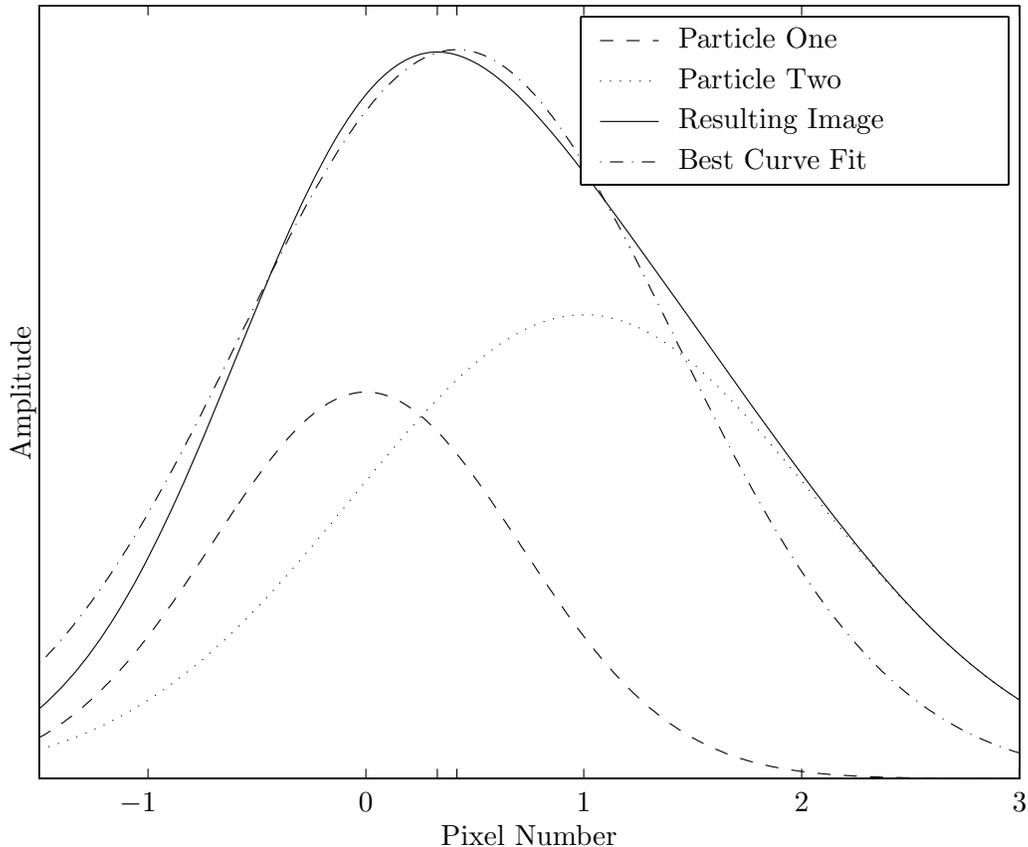


Figure 3: Overlapping particles.

maximum that is formed is not, in general, at a particle position. The example in figure 3 shows that using local maxima as a criterion, as in the second solution, will often produce very inaccurate results. In the figure, one particle is centered at 0 px and one at 1 px, while the best Gaussian curve fit is centered at 0.42 px. This clearly shows that when particle images overlap, the number of local maxima is not the number of particles and that the position of a local maximum is not directly related to particle positions.

From this discussion, we see that we cannot draw many conclusions from the number of local maxima. For the purposes of the current theoretical and experimental results, a Gaussian intensity distribution is used. If there are N particle images, the intensity distribution on the CCD, A_t , is

$$A_t(x, y) = \sum_{i=1}^N A_i F(\bar{r}), \quad (12)$$

where

$$\bar{r}^2 = \frac{(x - x_i)^2 + (y - y_i)^2}{r_{b_i}^2}. \quad (13)$$

In this expression, A_i and (x_i, y_i) are the intensity maximum and the center position of the i^{th} particle image on the CCD, respectively. The blur radius, r_{b_i} , is defined as the radius

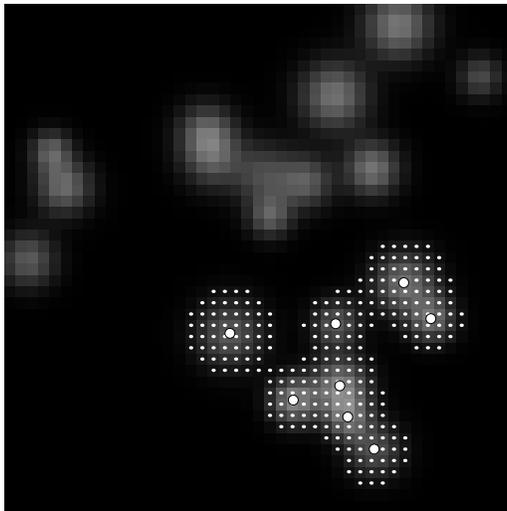


Figure 4: Blob selection. The white dots represent the blob pixels and the white dots with black circles show the centers of the particle images.

at which the intensity reaches 10% of its maximum value. In experiments, the intensity distribution is known and the variables subscripted with i are unknown. This forms a system of size $4N$ that needs to be solved, with N also being unknown. Typically, N is on the order of 10,000, which gives about 40,000 variables to solve for simultaneously, while solving for N itself. The purpose of the peak search algorithm is to solve for N , x_i , y_i , A_i , and r_{b_i} , where, for most applications, x_i , y_i , and N are the variables of interest. By adopting this method, the particle image positions of two or more overlapping particle images can be calculated; whereas, solution two would simply find one incorrect particle image position.

Since, in fact, real particle images do not have an exact Gaussian intensity distribution, the fitting function, $F(\bar{r})$, can easily be changed from a Gaussian to a more complicated function. However, for most applications simply using a Gaussian function is sufficient since particle images are near Gaussian.

To increase the speed of the program, the problem is split into a set of uncoupled smaller problems. To do that, the concept of a blob is introduced. In this article, a blob is defined as the set of all pixels which have an intensity level higher than a given threshold and which are connected (*i.e.* are neighbors). If this threshold is sufficiently small, blobs can be considered as separate problems (*i.e.* particles in one blob do not influence the intensity of other blobs). The concept of a blob is illustrated by example in figure 4. In this figure, all the pixels marked with a white dot belong to the same blob, and the white dots with black circles correspond to particle image locations. The definition of a blob used by the current authors varies from that used Stürer *et al* [12] and Kieft *et al* [4] in the fact that the current authors allow a blob to contain more than one particle image. Maas *et al* [2] allow for multiple particle images in a blob; however, the method they use to find the particle image centers is significantly different.

Inside each blob, the program tries to find the exact number and the position of the particle images. For a given blob, there are N_b particle images, for which the initial guess is based on the number of local maxima above a certain threshold, N_m , and the position of

these local maxima. It is clear that if the noise has a relatively small amplitude, $N_m \leq N_b$. The program was written so that the guessed number of particles, N_g , can only increase, if needed, to reach N_b . Therefore, the subroutine must ensure that the guessed number of particles never becomes higher than N_b (*i.e.* $N_m \leq N_g \leq N_b$).

Particles are added if the least-squares surface fit of the sum of N_g surfaces leads to relatively large errors. A particle is added where the error is a local maximum. However, to keep $N_m \leq N_b$, two particles cannot be added at the same iteration if they are closer than a given distance, usually 3 times the maximum blur radius.

Because it is so critical to ensure that the number of local maxima is indeed below the number of true particle images in a given blob, it is important to remove local maxima associated with noise. To do this, a Gaussian blur is applied during the image pre-processing. The Gaussian blur is useful because local maxima due to noise are leveled out, while Gaussian image distributions remain Gaussian.

To further decrease the processing time for large particle image densities, large blobs are split into smaller overlapping ones. The purpose of this splitting is not to isolate individual particle images, but rather to decouple the problem into a set of smaller problems. Therefore, this splitting is unrelated to the blob splitting of Maas *et al* [2]. To avoid inaccuracies due to cropped data, only the particle images that are found in the center part of the split blobs are recorded. The split blobs are overlapped in such a way as to ensure that their center parts will cover the entire blob.

To increase the computation speed even further, regions of a blob where the error is very low are locked and only the other regions of the blob where the error is still important are recalculated. Again, a minimum distance is used, that is to say, the error surrounding a particle must not only be low at the particle location but also low within this given minimum distance.

3.3 Calibration

The calibration process is a very important part of the program, as it drastically reduces the errors in the measurements. All the relations that were derived in previous sections assume that the camera is exactly aligned. For example, for the alignment error to be of the same order as the error associated with peak searching, each CCD, in our current camera configuration and without calibration would have to have pitch and yaw angles of $0^\circ \pm 0.5^\circ$, a roll angle of $0^\circ \pm 0.0025^\circ$ and a position in z and in x, y that is accurate within $1 \mu\text{m}$ and 100 nm , respectively. This accuracy is not feasible in practice. Furthermore, the lens aberrations create errors that are 100 times larger than the errors in peak searching. For all these reasons, a calibration correction is essential.

The concept of the calibration is that using a target placed at a well known position, a mapping from the actual CCD locations to virtual CCD locations can be created. Since the positions of all the dots on the target are assumed to be exactly known, the exact image positions on the virtual CCDs can be computed using ray tracing with theoretical lens and CCD locations. By identifying all the dots on the real CCDs, a mapping is created from each real CCD to each virtual CCD.

Typically, the target is a flat plate containing a grid of more than 500 dots. The mapping is created by fitting functions of several variables. These functions are chosen to be able to represent any kind of rotation, and translation, and to correct for the main errors that appear due to lens aberrations.

For the camera misalignment, 3 possible rotations and 3 possible displacements have

to be corrected. Only 6 variables are therefore needed [13, 6]; however, it is much easier to compute the calibration mapping by introducing 2 additional non-physical fitting parameters. The 6 variable mapping function [13, 6] can then be written as

$$X = \frac{A_1 + A_2x + A_3y}{1 + A_7x + A_8y}, \quad (14)$$

$$Y = \frac{A_4 + A_5x + A_6y}{1 + A_7x + A_8y}, \quad (15)$$

where X , Y denote the mapped location, and the x , y the actual locations. Note that the individual coefficients are not directly related to those of the previous authors [13, 6]. In addition to this, the authors assumed radial distortions due to lens aberrations. Two additional constants, not related to A_1 through A_8 , are introduced to locate the projection of the optical center of the lens onto the CCD, and six to create a mapping based on radius. Similar mapping functions have been used by other researchers [13, 6].

3.4 Epipolar Search and Ray Tracing

The three CCDs are equally spaced along the circumference of a circle and view the same volume. This means that one particle is projected onto a different location on each of the CCDs, since each CCD images the particle from a different location. If the three CCD images are overlaid, as shown in figure 2, the three particle images will form the corners of a near-equilateral triangle [3, 10]. Complications will clearly arise when the image contains many thousands of particles, and it is no longer obvious which three peaks correspond to a specific particle. In order to find these corresponding peaks, a search algorithm using epipolar lines is employed [2, 3, 4], then ray tracing is used to find the point of closest intersection of the three rays and the RMS distance from the rays to this point. Consider a ray originating from the i^{th} CCD, which is given by $\mathbf{P}_i = \mathbf{P}_i^0 + \lambda_i \mathbf{e}_i$, where \mathbf{P}_i^0 is the center of i^{th} lens, \mathbf{e}_i is the unit vector pointing from the particle image to the lens center of i^{th} CCD and λ_i is a scalar. The estimated position, \mathbf{P}^{P} , of the particle is given by minimizing the function

$$\sum_{i=1}^3 (\mathbf{P}_i - \mathbf{P}^{\text{P}})^2, \quad (16)$$

over all possible λ_i , which gives an optimum $\lambda_i = (\mathbf{P}^{\text{P}} - \mathbf{P}_i^0) \cdot \mathbf{e}_i$. The RMS error, E_{r} , associated with this λ_i and the three rays is

$$E_{\text{r}}^2 = \frac{1}{3} \sum_{i=1}^3 \left\{ (\mathbf{P}^{\text{P}} - \mathbf{P}_i^0)^2 - [(\mathbf{P}^{\text{P}} - \mathbf{P}_i^0) \cdot \mathbf{e}_i]^2 \right\}. \quad (17)$$

Minimizing the error with respect to the particle position, \mathbf{P}^{P} , produces a linear system

$$\left(\sum_{i=1}^3 \mathbf{M}_i \right) \mathbf{P}^{\text{P}} = \left(\sum_{i=1}^3 \mathbf{M}_i \mathbf{P}_i^0 \right), \quad (18)$$

where

$$\mathbf{M}_i = \mathbf{e}_i \otimes \mathbf{e}_i - \mathbf{I}, \quad (19)$$

\mathbf{I} being the identity matrix and the symbol \otimes representing the outer product. The point of closest intersection of the three rays, \mathbf{P}^{P} , can then easily be solved for. Substituting this point back into equation 17 gives the error associated with the three rays.

Since any one particle image may have multiple possible corresponding particle images, the combination that has the lowest RMS error, *i.e.* the one for which the three rays come closest to intersecting, is considered the correct one. In addition, no particle image may be part of more than one set of corresponding particle images.

3.5 Peak Level

A key enhancement of the system is the ability to include hard-to-detect particle images. These are images that are so weak that the signal-to-noise ratio of the particle image intensity is close to one. Rather than ignoring these peaks, as they may indeed correspond to particles, they are recorded as weak peaks. Peaks which are above the given threshold are simply recorded as strong peaks. A peak may also be recorded as a weak peak if the blob to which it belongs is very crowded. Because weak peaks may not correspond to real particles, they may only belong to an equilateral triangle whose other vertices are formed by strong peaks. Since the sub-pixel accuracy of weak peaks is in general much less than that of a strong peak, they are not considered when converting from triangles to 3D positions. In this case, the particle location is simply the point of closest intersection of the rays emanating from the two strong peaks.

3.6 Particle Tracking

The DDPIV technique developed by Pereira *et al* [3, 10] performs 3D cross correlations on interrogation voxels in order to construct the velocity field. This technique was not chosen for two reasons. The primary reason is that the density of particles is very low in 3D, even with 8,000 particles in the test volume, very large correlation volumes would be required. To illustrate this, in 2D, 8,000 particles would represent about 90 particles in each direction; whereas in 3D, it is only 20 particles in each direction. In addition, PIV produces an average velocity in interrogation regions; therefore, flow features can be lost in regions of high velocity gradients. This and other benefits of PTV are discussed by Kim *et al* [14]. One of the general limitation of PTV is that it requires accurately locating individual particle images; however, in 3DPTV this does not present any additional problems since these particle image locations must be computed in order to find the three-dimensional particle positions.

In the current data processing program, the final velocity field is constructed using a particle tracking method. Since only one set of particle pairs is considered, trajectories can not be used to help find corresponding pairs as is done by Malik *et al* [15], Stüer *et al* [12] and Willneff [16]. The current particle tracking algorithm uses a hybrid process similar to the two-dimensional process used by Kim *et al* [14]. First, a rough velocity field is built using a 3D correlation. The domain is divided into small correlation voxels that overlap. The correlation voxel from one frame is offset by the local displacement in the second frame. The resulting vector is the one that gives the best correlation between the original correlation box and the offset box. At the same time, simple direct particle tracking is performed for particles that have only one possible corresponding particle in the other frame. A weighted average between the correlation and the simple direct particle tracking is then evaluated. Appropriate weights are assigned to all the vectors so that the correlation approach has more influence where it is more accurate (*i.e.* where the particle density is high) and the simple particle tracking has more influence where the density is low. This hybrid approach is only used to form a first guess of the actual velocity field. Using

this average field as a first guess, a complete particle tracking is performed. The program iterates the velocity field several times. In each iteration bad vectors are suppressed and good ones are added. Specifically, a vector is considered good if it is within a certain number of standard deviations of the surrounding vectors. If it is outside this range, the vector is considered bad. Details of which vectors are considered good and bad are beyond the scope of this paper and will be presented in future publications. It is important to note that the final velocity field is formed only by particle tracking.

4 Theoretical Error Analysis

Without a thorough understanding of the errors associated with 3DPTV, the results have very little meaning. Because of this, a detailed error analysis was performed, both theoretically and experimentally. The error can be broken up into two uncorrelated parts. The first part is the error due to inaccuracies in calibration and in construction. The second part is the error due to the fact that peak searching is performed on discretized images. In this section the theoretical error analysis is presented. All calculations have been done assuming that the measurements have a Gaussian distribution.

4.1 Calibration Error

As discussed earlier, calibration is a key step in 3DPTV. Without calibration the CCDs must be placed with an unattainable accuracy. Of course, the calibration is not exact either. Errors in calibration arise from several sources. These include residual lens aberrations and the imperfect placement of the test target. There are also errors due to imprecision in the placement of the lenses. Since these errors can not be measured, it is not necessary to derive the exact effect of these errors on velocity measurements. Rather, it is useful to understand the form of the measurement errors.

The errors in calibration will produce errors in the position of a particle, ΔP_i^c . We can then write the errors in velocities due to calibration errors as

$$\Delta U_i^c = \frac{\partial \Delta P_i^c}{\partial x} U_x + \frac{\partial \Delta P_i^c}{\partial y} U_y + \frac{\partial \Delta P_i^c}{\partial z} U_z, \quad (20)$$

where i is either x , y , or z . This can be written more simply as

$$\Delta U_i^c = |U| F_i \left(\frac{U_x}{U_z}, \frac{U_y}{U_z}, x, y, z, \Lambda \right), \quad (21)$$

where Λ is the set of all parameters that describe the camera and the calibration setup. F_i is a dimensionless function that describes the effect of the imprecision of the camera construction and of the calibration setup on the measured velocities. It is important to note that the error on measured velocities is proportional to the magnitude of velocity.

In order to compare with experiments, which were done using a plate in the x - y plane, the error, ΔU_i^c , is integrated over a x - y plane in the test volume to form the plate-averaged error, $\overline{\Delta U_i^c}$. The bar denotes a plate-averaged value defined as

$$\overline{g} = \frac{1}{d^2} \int_{-d/2}^{d/2} \int_{-d/2}^{d/2} g \, dx \, dy, \quad (22)$$

where d is the side length of the test volume. The experiments were done by translating a plate. This means that the velocities were the same for all points on the plate. We can therefore write the plate-averaged error, $\overline{\Delta U_i^c}$, as

$$\mu_i^c = \overline{\Delta U_i^c} = |\mathbf{U}| \overline{F_i} \left(\frac{U_x}{U_z}, \frac{U_y}{U_z}, d, z, \Lambda \right). \quad (23)$$

Since F_i is a function of x and y , we can measure the amount that individual points vary from the plate-averaged value, $(\Delta U_i^c - \overline{\Delta U_i^c})$. We then define σ_i^c to be the range in which 90% of the variances fall. From the definition of σ_i^c , we see that it will also be proportional to $|\mathbf{U}|$.

4.2 Peak Searching Error

A 2D error analysis for DDPIV was done by Pereira *et al* [3, 10]; however, since his analysis is only two-dimensional it can not be applied directly to the actual three-dimensional system. A three-dimensional error analysis was done by Kajitani *et al* [17], which assumes that the particle location is calculated using a best linear unbiased estimator.

We analyze the effect of particle image location errors on the three-dimensional particle position when the ray tracing algorithm, discussed in section 3.4, is used to calculate the particle position. To start with, the peak searching algorithm has a limited accuracy with which it can find the centroid of a blurred particle. This causes an error in the image position. In this paragraph, the 90% confidence limit of particle image location is $\langle \epsilon \rangle_{90}$ in both the x - and y -directions. In order to find the errors in the particle position due to errors in particle image positions, a particle in 3D space is projected onto each CCD. Errors are then added to the x and y placement of these images. This is to say, using the equations in Section 2.1.2, we have

$$\mathbf{I}_1 = (\xi_1 x + \Delta x_1, \xi_1 y + \xi_2 + \Delta y_1), \quad (24)$$

$$\mathbf{I}_2 = \left(\xi_1 x + \frac{\sqrt{3}}{2} \xi_2 + \Delta x_2, \xi_1 y - \frac{\xi_2}{2} + \Delta y_2 \right), \quad (25)$$

$$\mathbf{I}_3 = \left(\xi_1 x - \frac{\sqrt{3}}{2} \xi_2 + \Delta x_3, \xi_1 y - \frac{\xi_2}{2} + \Delta y_3 \right), \quad (26)$$

where

$$\xi_1 = -\frac{d_C}{S_z}, \quad (27)$$

$$\xi_2 = \frac{r(L-z)}{S_z L} d_C. \quad (28)$$

Δx_i and Δy_i have the same 90% confidence interval ($\langle \Delta x_i \rangle_{90} = \langle \Delta y_i \rangle_{90} = \langle \epsilon \rangle_{90}$). Then the particle images, with error added, are projected back into three-dimensional space and the point of closest ray intersection is considered the particle position. The error in particle position is then the distance from the original particle position to the particle position found after errors were added. To simplify the results obtained, several assumptions were made: $r^2 \ll z^2$, $x^2 \ll z^2$ and $y^2 \ll z^2$. For our system, the ratios $\frac{r^2}{z^2}$, $\frac{x^2}{z^2}$, and $\frac{y^2}{z^2}$ are less

than 1%. To first order, this gives the following errors

$$\Delta P_x^{\text{P}} = \frac{-zS}{3d_{\text{C}}} \left(\Delta x_{\text{s}} + \frac{\sqrt{3}x}{2r} \Delta x_{\text{d}} + \frac{x}{2r} \Delta y_{\text{d}} \right), \quad (29)$$

$$\Delta P_y^{\text{P}} = \frac{-zS}{3d_{\text{C}}} \left(\Delta y_{\text{s}} + \frac{\sqrt{3}y}{2r} \Delta x_{\text{d}} + \frac{y}{2r} \Delta y_{\text{d}} \right), \quad (30)$$

$$\Delta P_z^{\text{P}} = \frac{-z^2S}{3d_{\text{C}}r} \left(\frac{\sqrt{3}}{2} \Delta x_{\text{d}} + \frac{1}{2} \Delta y_{\text{d}} \right), \quad (31)$$

where

$$\Delta x_{\text{s}} = \Delta x_1 + \Delta x_2 + \Delta x_3, \quad (32)$$

$$\Delta y_{\text{s}} = \Delta y_1 + \Delta y_2 + \Delta y_3, \quad (33)$$

$$\Delta x_{\text{d}} = \Delta x_2 - \Delta x_3, \quad (34)$$

$$\Delta y_{\text{d}} = 2\Delta y_1 - \Delta y_2 - \Delta y_3. \quad (35)$$

The distribution of the particle position errors, ΔP_i^{P} , is unbiased, *i.e.* $\mu_i^{\text{P}} = 0$. Again, i is either x , y , or z . The 90% confidence limits of particle locations can then be evaluated and are

$$\langle \Delta P_x^{\text{P}} \rangle_{90} = \frac{z\sqrt{r^2 + x^2}}{\sqrt{3}d_{\text{C}}r} S \langle \epsilon \rangle_{90}, \quad (36)$$

$$\langle \Delta P_y^{\text{P}} \rangle_{90} = \frac{z\sqrt{r^2 + y^2}}{\sqrt{3}d_{\text{C}}r} S \langle \epsilon \rangle_{90}, \quad (37)$$

$$\langle \Delta P_z^{\text{P}} \rangle_{90} = \frac{z^2}{\sqrt{3}d_{\text{C}}r} S \langle \epsilon \rangle_{90}. \quad (38)$$

In the case of velocimetry, the 90% confidence limits of velocities are simply:

$$\langle \Delta U_x^{\text{P}} \rangle_{90} = \frac{\sqrt{2}}{t_{\text{s}}} \langle \Delta P_x^{\text{P}} \rangle_{90} = \frac{z\sqrt{2(r^2 + x^2)}}{\sqrt{3}t_{\text{s}}d_{\text{C}}r} S \langle \epsilon \rangle_{90}, \quad (39)$$

$$\langle \Delta U_y^{\text{P}} \rangle_{90} = \frac{\sqrt{2}}{t_{\text{s}}} \langle \Delta P_y^{\text{P}} \rangle_{90} = \frac{z\sqrt{2(r^2 + y^2)}}{\sqrt{3}t_{\text{s}}d_{\text{C}}r} S \langle \epsilon \rangle_{90}, \quad (40)$$

$$\langle \Delta U_z^{\text{P}} \rangle_{90} = \frac{\sqrt{2}}{t_{\text{s}}} \langle \Delta P_z^{\text{P}} \rangle_{90} = \frac{z^2\sqrt{2/3}}{t_{\text{s}}d_{\text{C}}r} S \langle \epsilon \rangle_{90}. \quad (41)$$

These results agree with those of Kajitani *et al*, meaning that when the above assumptions are true, the current ray tracing algorithm produces a best linear unbiased estimate of the particle position. It is assumed that there is negligible error associated with the time separation, t_{s} , which for modern timing systems is true. The ratios of these confidence limits take a very simple form:

$$\frac{\langle \Delta U_z^{\text{P}} \rangle_{90}}{\langle \Delta U_x^{\text{P}} \rangle_{90}} = \frac{\langle \Delta P_z^{\text{P}} \rangle_{90}}{\langle \Delta P_x^{\text{P}} \rangle_{90}} = \frac{z}{\sqrt{r^2 + x^2}}, \quad (42)$$

$$\frac{\langle \Delta U_z^{\text{P}} \rangle_{90}}{\langle \Delta U_y^{\text{P}} \rangle_{90}} = \frac{\langle \Delta P_z^{\text{P}} \rangle_{90}}{\langle \Delta P_y^{\text{P}} \rangle_{90}} = \frac{z}{\sqrt{r^2 + y^2}}. \quad (43)$$

If we integrate the errors in a x - y plane contained in the test volume, which will be useful for comparisons with experiments, we find that the 90% confidence limit of plate-averaged velocities due to peak searching, σ^P , are

$$\sigma_x^P = \sigma_y^P = \left\langle \overline{\Delta U_x^P} \right\rangle_{90} = \left\langle \overline{\Delta U_y^P} \right\rangle_{90} = \frac{z\sqrt{12r^2 + d^2}}{3\sqrt{2}t_s d_C r} S \langle \epsilon \rangle_{90}, \quad (44)$$

$$\sigma_z^P = \left\langle \overline{\Delta U_z^P} \right\rangle_{90} = \frac{z^2\sqrt{2/3}}{t_s d_C r} S \langle \epsilon \rangle_{90}, \quad (45)$$

where d is the side length of the test volume. The ratios of these confidence intervals are

$$\frac{\sigma_z^P}{\sigma_x^P} = \frac{\sigma_z^P}{\sigma_y^P} = \frac{2\sqrt{3}z}{\sqrt{12r^2 + d^2}}. \quad (46)$$

These ratios are typically of order 15. The errors in positions and ratios of errors in Equations 44 through 46 will be assumed to be the errors over the entire x - y plane for simplicity. This assumption will result in a maximum relative difference between Equations 42 and Equation 46 in the test volume of

$$1 - \sqrt{\frac{12r^2 + d^2}{12r^2 + 3d^2}}, \quad (47)$$

which for the current authors is approximately 9%.

One can expect the error to decrease if the time separation increases. However t_s is limited because the particles must not leave the test volume between frames and because, if the corresponding particles are separated by a large distance, it can be difficult to find these corresponding particles back, and if we do, the velocity obtained will just be a time-integrated average.

4.3 Combined Error

The error due to peak searching and calibration can then be joined to form a combined error. From section 4.2 we see that the expected value of the velocity error due to peak searching algorithm is zero; therefore, the expected value of the total error, μ_i , is just that due to calibration error, *i.e.* $\mu_i = \mu_i^C$. Since calibration error is proportional to velocity, the combined mean error will also be proportional to velocity. Both calibration and peak searching contribute to the deviation error. Because the errors due to peak searching and calibration are uncorrelated, the combined 90% confidence limit of plate-averaged velocity, σ_i , is

$$\sigma_i = \sqrt{(\sigma_i^C)^2 + (\sigma_i^P)^2}, \quad (48)$$

where i is x , y , or z .

If the medium surrounding the camera does not have a constant index of refraction, the combined error is slightly different. However, the error is of the same order.

5 Results

In this section, several results will be presented. The first of these are theoretical results to examine the highest accuracy that can be obtained using our data analysis program. A movable plate is then examined to experimentally determine the errors in displacements. Finally, measurements of the vortex generated by the tip of a delta wing conducted in a water tunnel are presented.

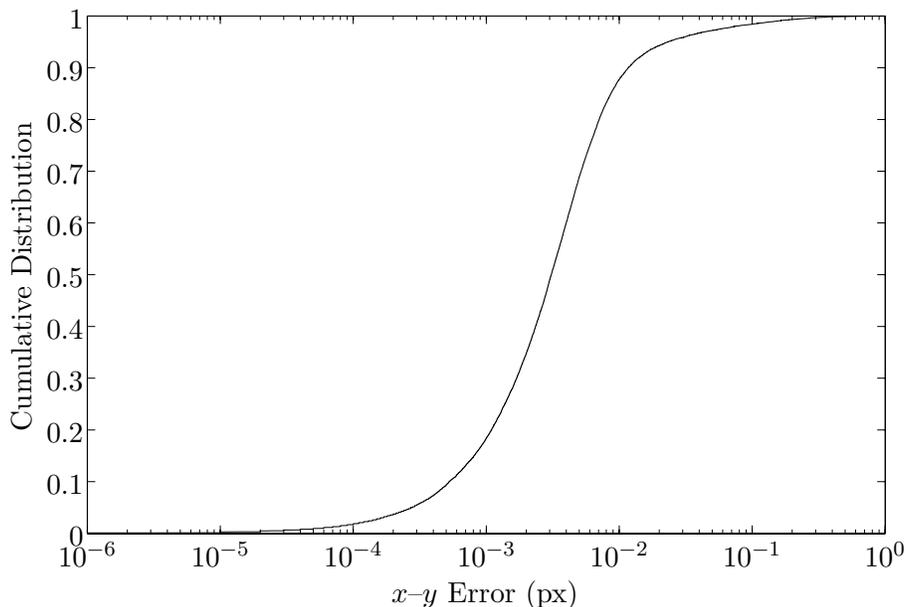


Figure 5: Cumulative distribution function of particles.

5.1 Peak Searching

A synthetic image containing 10,000 particles with known positions was created. Each particle had a Gaussian intensity distribution. In this paper, a particle image radius is defined as the radius at which the intensity of the image reaches 10% of its maximum. The particle image radii in the test of the peak searching algorithm varied between 2 px and 4 px. The intensity, position and radius of each of the particle images were completely random, within the constraint range. Because there were 10,000 particle images distributed over 1 Mpx many of the particle images overlapped.

The program successfully identified 9,911 particles out of 10,000. The missing particles are mostly those with neighbors within 0.5 px, that is to say, they are so close that they are virtually indistinguishable. For each of the particles that were found, the closest real particle was selected in order to evaluate the errors. These results are shown in figure 5. The error is smaller than 0.01 px in each direction for most of the peaks (87.7%) and for the vast majority (94.3%), is closer than 0.02 px to their actual location. The 90% confidence interval is 0.0117 px (which corresponds to $\langle \epsilon \rangle_{90}$ in equations 44 and 45), while the mean error is 0.00926 px. It is also interesting to note that only 8,446 local maxima exist on the image. This means that routines using simple local maxima fitting would find at most 6,892 correct particle images. The remaining local maxima will contain two or more particle images, which can not be computed correctly using local maxima fitting. For these theoretical results, a Gaussian fitting function was used in the peak searching routine and each particle image was given a Gaussian distribution; therefore, these results represent the best possible accuracy of the peak searching algorithm.

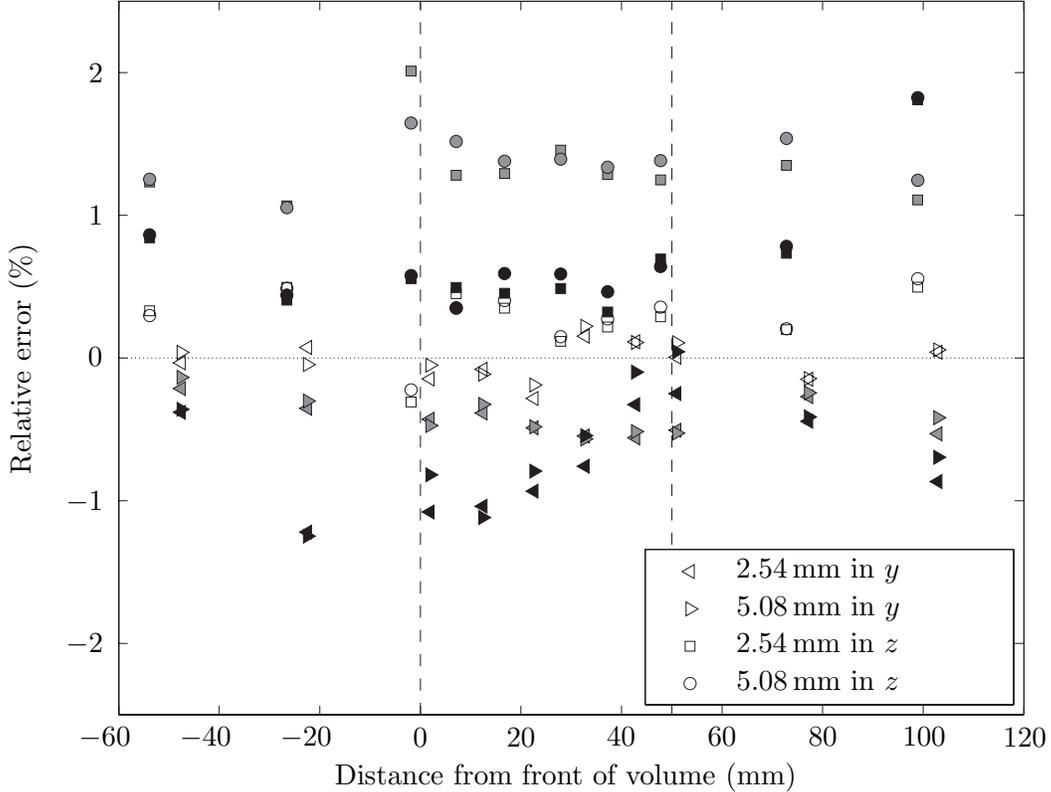


Figure 6: Plate-averaged displacement errors. White, grey, and black symbols refer to the errors in the x -, y -, and z -directions respectively. The 2.54 mm and the 5.08 mm cases are the average of 3 and 2 data points, respectively.

5.2 Experimental Verification

A movable plate test case was performed using a flat plate with an array of white dots on a black background. There were 100 dots per square inch. The plate was placed at z -positions of 469, 494, 519, 529, 539, 549, 559, 569, 594, and 619 mm from the camera. The plate was then moved in three equal increments of 2.54 mm in the y -direction for each of the z -locations, this resulted in a total of 6 displacement vectors for each z -position (3 displacements of 2.54 mm, 2 displacements of 5.08 mm and 1 displacement of 7.62 mm). Similarly the plate was moved in three equal increments of 2.54 mm in the z -direction for each z -position.

First, the mean error was evaluated, and as expected is proportional to the displacement as shown in equation 23. Figure 6 shows the relative mean error obtained for 2.54 mm and 5.08 mm displacements, each data point is an average of 3 and 2 values, respectively. In this plot, the 7.62 mm results are not shown since they correspond to the average of the three 2.54 mm results.

Even though the mean errors are very small, all within 2%, they are larger than expected. One reason for some of the inaccuracy, besides the camera system itself, is lack of precision of the testing setup. In particular, the precision with which the testing target could be moved was on the order of the mean errors.

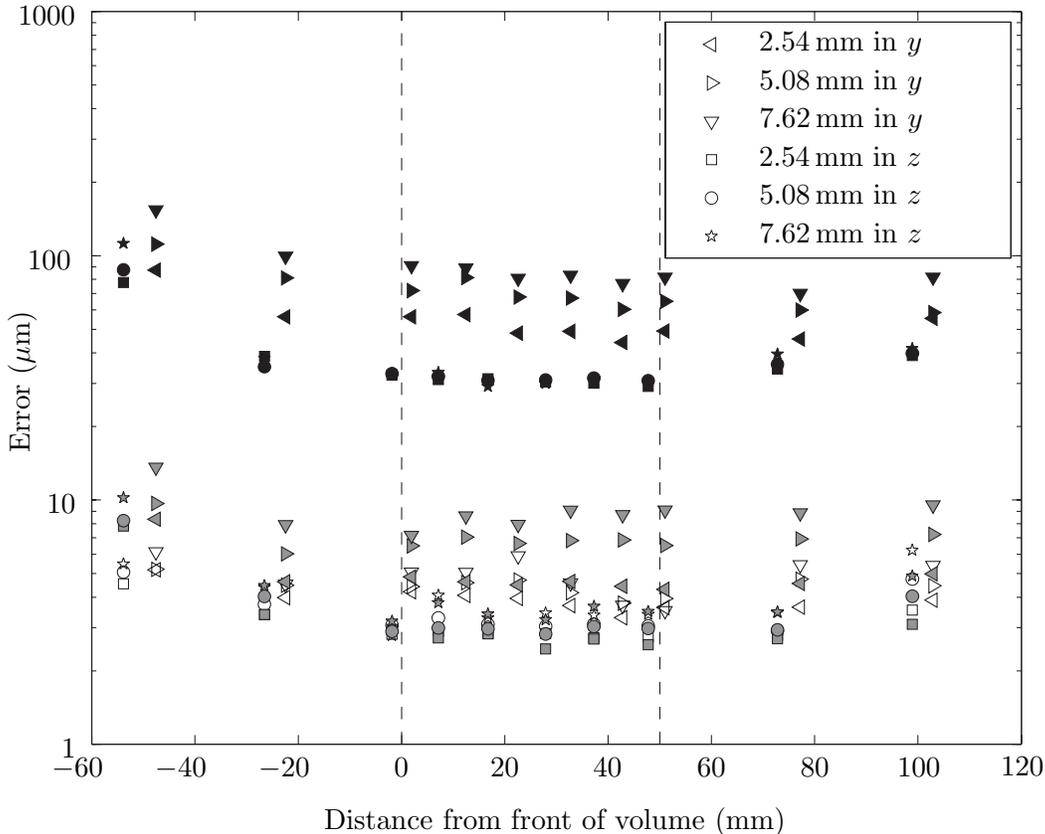


Figure 7: 90% confidence limits of plate-averaged displacements. White, grey, and black symbols refer to the errors in the x -, y -, and z -directions respectively. The 2.54 mm and the 5.08 mm cases are the average of 3 and 2 data points, respectively.

The 90% confidence limit of the displacements was also evaluated. The results are plotted in figure 7. Each symbol is the average of the 90% confidence limits from three and two data sets in the 2.54 mm and the 5.08 mm cases, respectively.

One can directly note that the results depend on the displacement distance of the plate. Our theory tells us that error due to peak searching 4.2 does not depend on the displacement; whereas, the error due to calibration 4.1 is proportional to the velocity (or, for a given time separation, the displacement). As equation 48 shows, the combined 90% confidence limit of plate-averaged displacement, $\langle \delta D_i \rangle_{90}$, can be written as

$$\langle \delta D_i \rangle_{90} = t_s \sqrt{(\sigma_i^c)^2 + (\sigma_i^p)^2}, \quad (49)$$

Since, for most flow measurements, the displacement of the particles will be smaller than 2.54 mm, it is interesting to analyze the results for this particular displacement, as this represents the worst case scenario. Figure 8 therefore gives an upper bound for the 90% confidence limit one can expect with the 3DPTV setup. According to these results, the 90% confidence limit in the test volume remains under $5.2 \mu\text{m}$ and under $66 \mu\text{m}$ in the x -, y -directions and in the z -direction, respectively.

To be able to compare with some of the results obtained previously, it is necessary

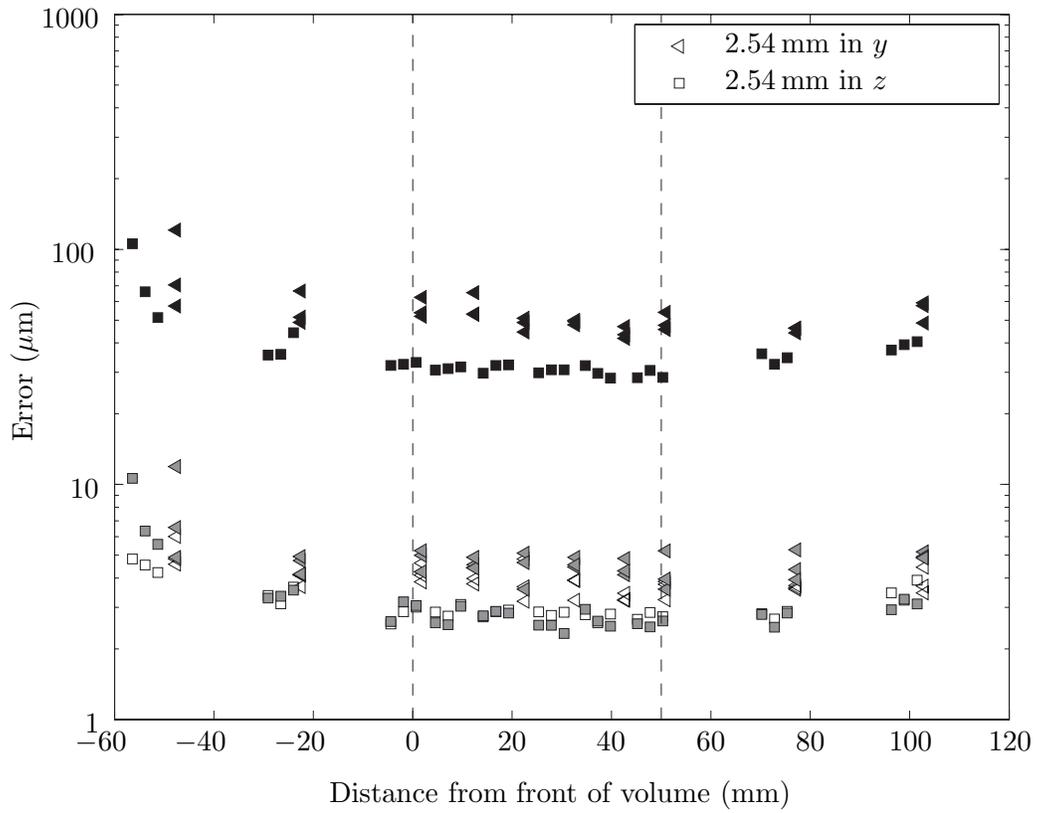


Figure 8: 90% confidence limits of plate-averaged displacements. White, grey, and black symbols refer to the errors in the x -, y -, and z -directions, respectively.

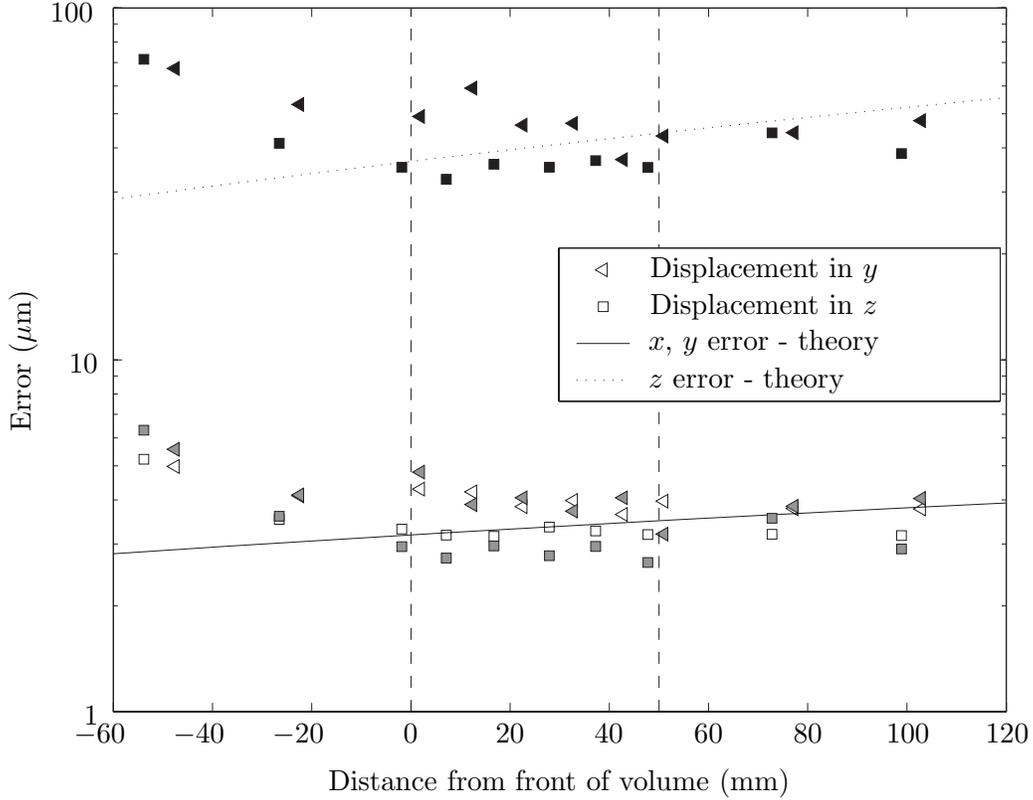


Figure 9: Fixed part of 90% confidence limits of plate-averaged displacements. White, grey, and black symbols refer to the errors in the x -, y -, and z -directions, respectively.

to separate the constant part of the 90% confidence limit from its varying part, as in equation (49). Using the results shown in figure 7, one can obtain these two terms.

The constant part of the 90% displacement error can be obtained theoretically from equations (44) and (45). Both the theoretical values and the experimental data are shown in figure 9. For the theoretical curves, the $\langle \epsilon \rangle_{90}$ value was found by performing a single least-squares fit to all the data inside the volume. This fit produced an $\langle \epsilon \rangle_{90}$ value of 0.069 px. This value is a bit higher than the one obtained theoretically in section 5.1. The discrepancy in the results for low z -position comes from the fact that the dots on the plate appear very blurred and that the amplitude of the peaks becomes very low.

Another important result is the ratio of the constant parts of the error. According to theory, the ratio between the x or y constant part of the 90% confidence limit, and the z constant part of the 90% confidence limit is given by equation (46). The theoretical, parameter-free curve, as well as the experimental values for this ratio are shown in figure 10.

The varying part of the 90% confidence limit is presented in figure 11. These errors remain quite small ($1 \mu\text{m}$ per mm in x or y and $10 \mu\text{m}$ per mm in z); however, part of the error may be due to lack of robustness of the test setup, in particular the placement and precision of the testing target.

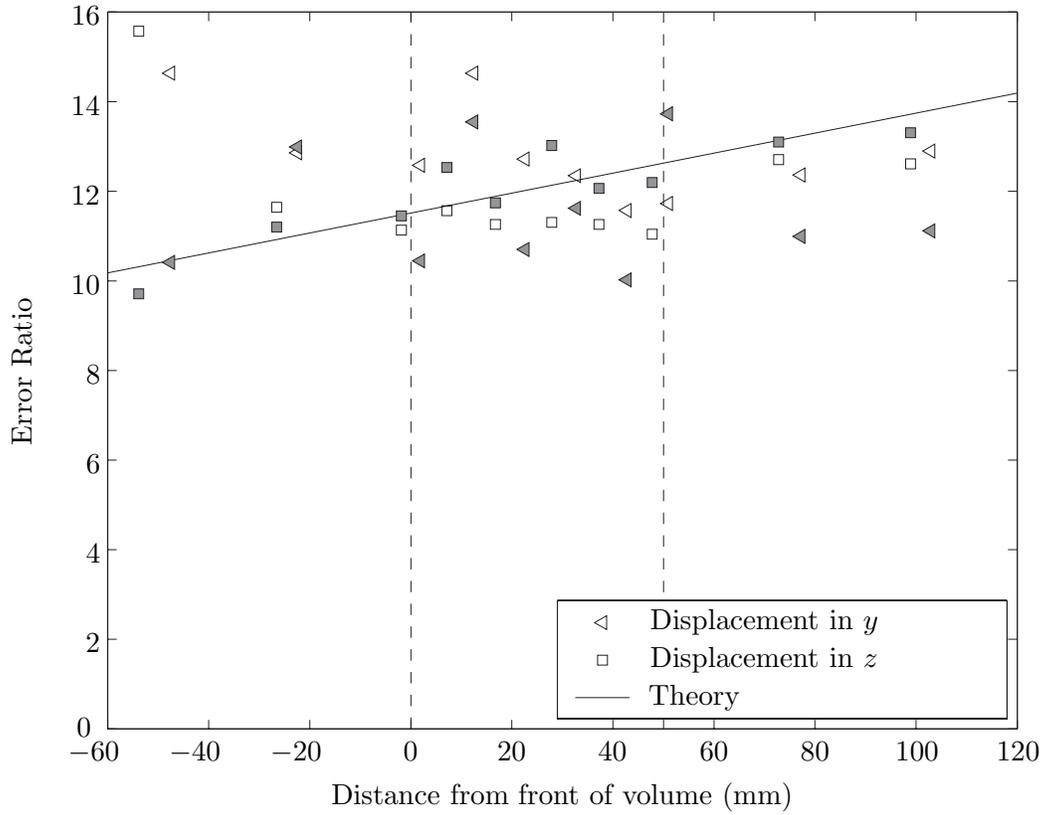


Figure 10: Ratio of fixed part of 90% confidence limits of plate-averaged displacements. Empty and grey symbols refer to the ratio of the constant part of the 90% confidence limit in the x - and y -directions, respectively, and the constant part of the 90% confidence limit in the z -direction.

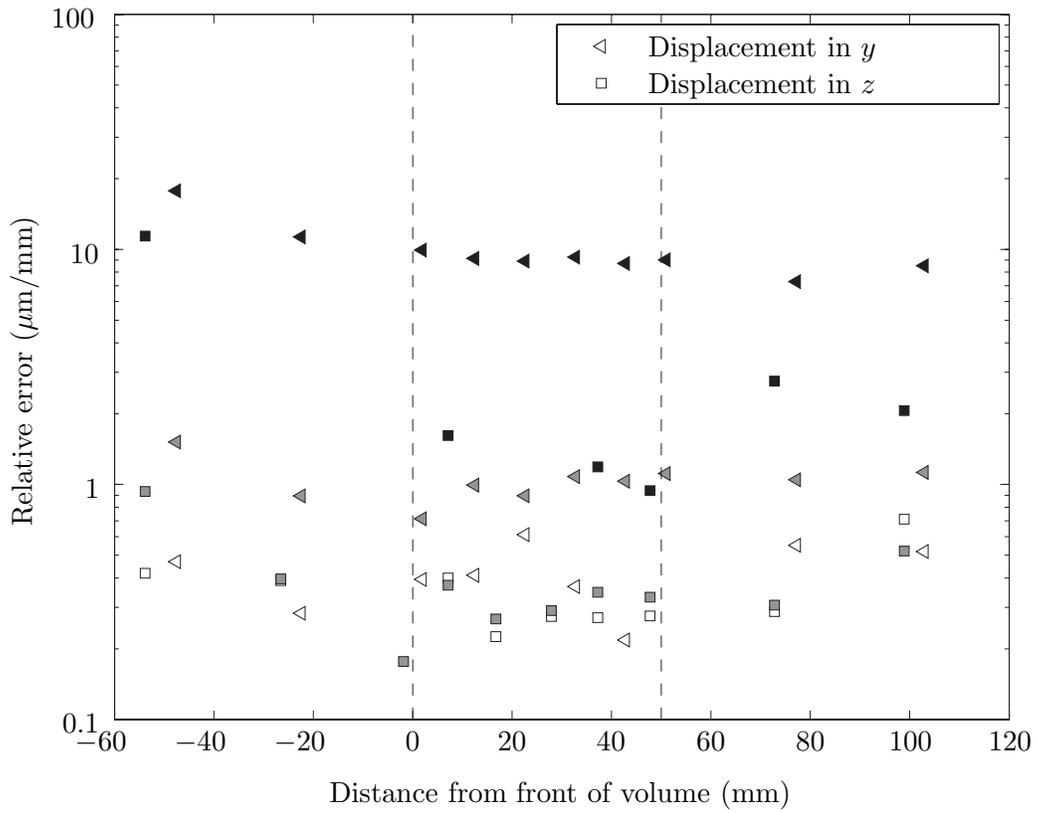


Figure 11: Variable part of 90% confidence limits of plate-averaged displacements. White, grey, and black symbols refer to the errors in the x -, y -, and z -directions, respectively.

5.3 Delta Wing Experiment

Measurements of the vortex generated by a delta wing were conducted in a water tunnel at the University of Washington. A delta wing with a 65.5° sweep angle was placed at a 25° angle of attack. The free-stream speed of the tunnel was 10 cm/s. The results presented below were obtained by time-averaging multiple vector fields.

Figure 12 shows the experimental setup. In both views the location of the laser beam with respect to the delta wing is shown. The side view shows the placement of the camera and the glass of the water tunnel. The coordinate system, i , is centered at the tip of the delta wing. i_1 is along the centerline of the delta wing, i_2 is in the plane of the delta wing and perpendicular to the centerline, and i_3 is perpendicular to the surface of the delta wing.

Three slices of the velocity field are shown in figure 13. The color represents the magnitude of the velocity. The contours correspond to increments of 0.02 cm/s. Starting at an i_1 of 110 mm we see a high speed region, which corresponds to the vortex core. We see that at $i_1 = 180$ mm vortex breakdown has occurred as there is no longer a high speed core. The low speed region corresponds to the secondary vortex. The slices have been cropped at the delta wing.

An in-plane vector field plot at $i_1 = 145$ mm is shown in figure 14. The vortex is clearly seen in the figure. Since the vector field created by 3DPTV is not on a regular grid, the data was interpolated onto a uniform grid. $i_3 = 0$ mm corresponds to the bottom of the delta wing, which is shown as an outline in the figure.

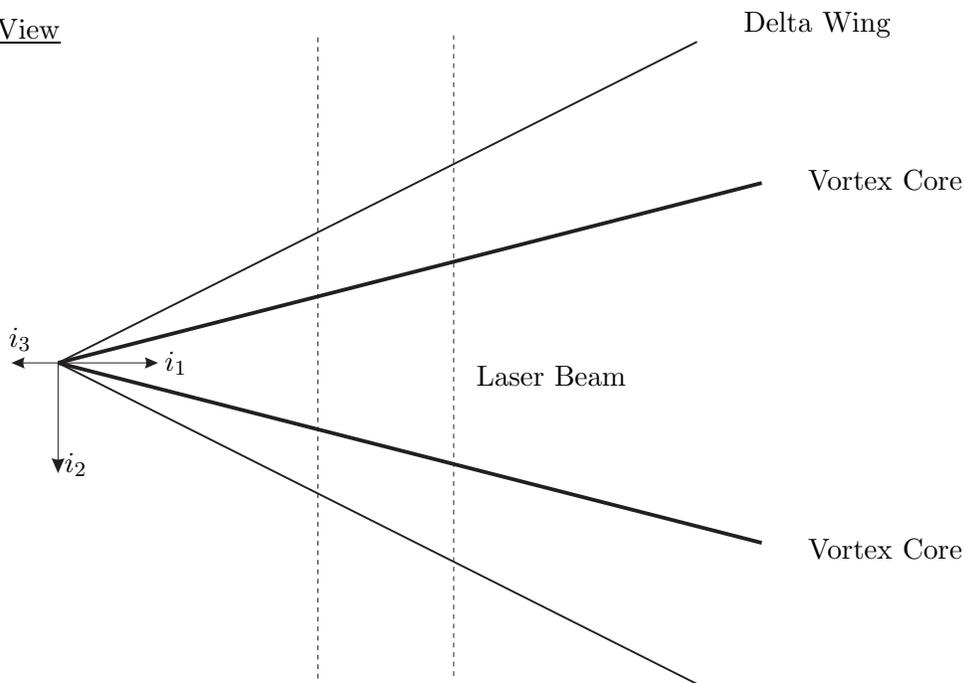
6 Conclusions and Recommendations

In conclusion, the 3D particle triangulation velocimetry technique that was implemented has been shown to produce excellent results in terms of accuracy. The errors were of the order of the accuracy that can be achieved with the translation stages used. The upper bound estimate for the mean error is only 2% of the displacement and the 90% confidence interval is less than $5.2 \mu\text{m}$ and less than $66 \mu\text{m}$ in the x -, y -directions and in the z -direction, respectively. These errors agree fairly well with the theoretical error analysis.

Several improvements can be made to increase the accuracy of the camera or to broaden its use. First, a substantial accuracy improvement in z can be obtained by increasing the radius of the lens positions, r , while still keeping the CCDs in a compact unit. One way of obtaining large r is by physically separating the cameras as many researchers do [2, 12, 16, 4]; however, physically separating the cameras is not recommended because of additional calibration required and lack of convenience when moving experiments. The camera of the current authors maintained its calibration for over a year even though the camera was moved and used in several different test setups. Furthermore, preliminary experiments show that the accuracy in all directions can also be increased by focusing the camera inside the test volume (by reducing L and so increasing d_C). At the same time, the blur radius will be reduced and the number of overlapping particles will be significantly decreased, allowing for either a larger number of particles or for a lower processing time. Because of lens limitation, even these focused images will span multiple pixels. To minimize lens aberrations the axes of the lenses should intersect the center of the test volume.

The 3DPTV system was designed to be able to measure a wide variety of fluid flows as well as solid surfaces. In order to determine three-dimensional locations, individual particles must be found, this lends itself to direct particle tracking. In addition, particle tracking, as opposed to a statistical method, works well even when there are low densities

Bottom View



Side View

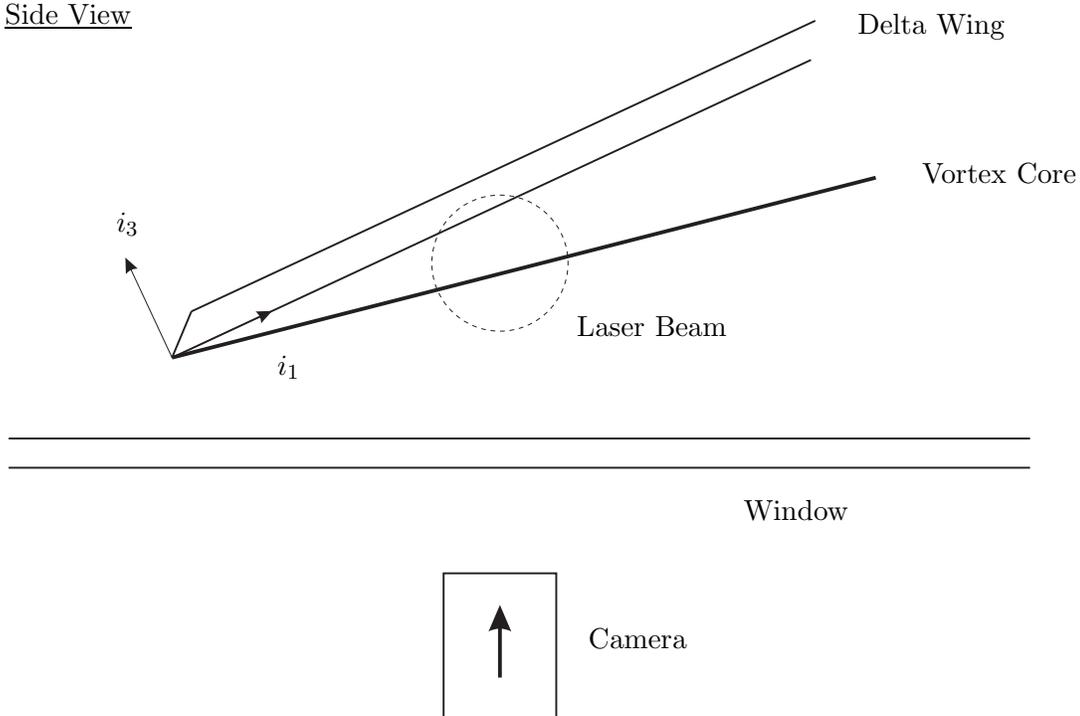


Figure 12: Delta wing setup for three-dimensional velocity measurements of a tip generated vortex.

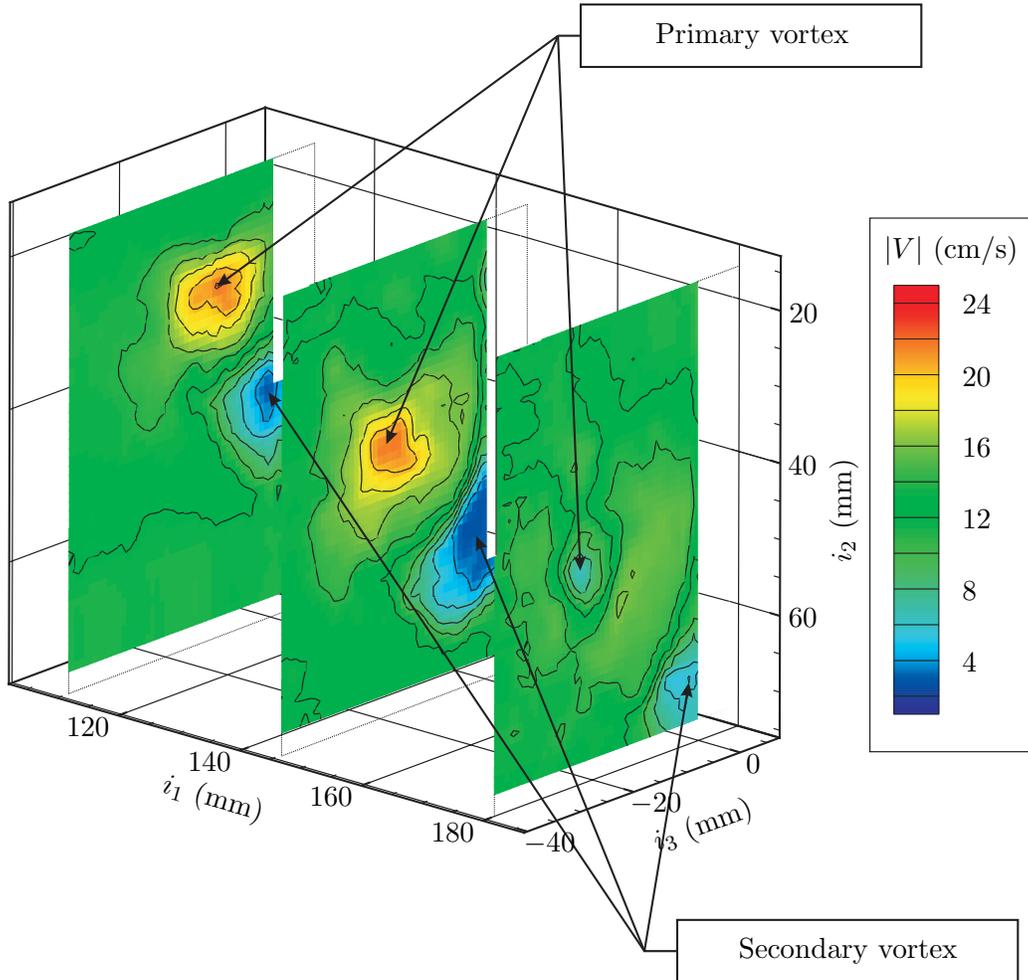


Figure 13: Three-dimensional slices (at an i_1 of 110 mm, 145 mm and 180 mm) showing the breakdown of a vortex generated by a delta wing. The color represents the magnitude of the flow velocity. Note that the slices are cropped at the plate location.

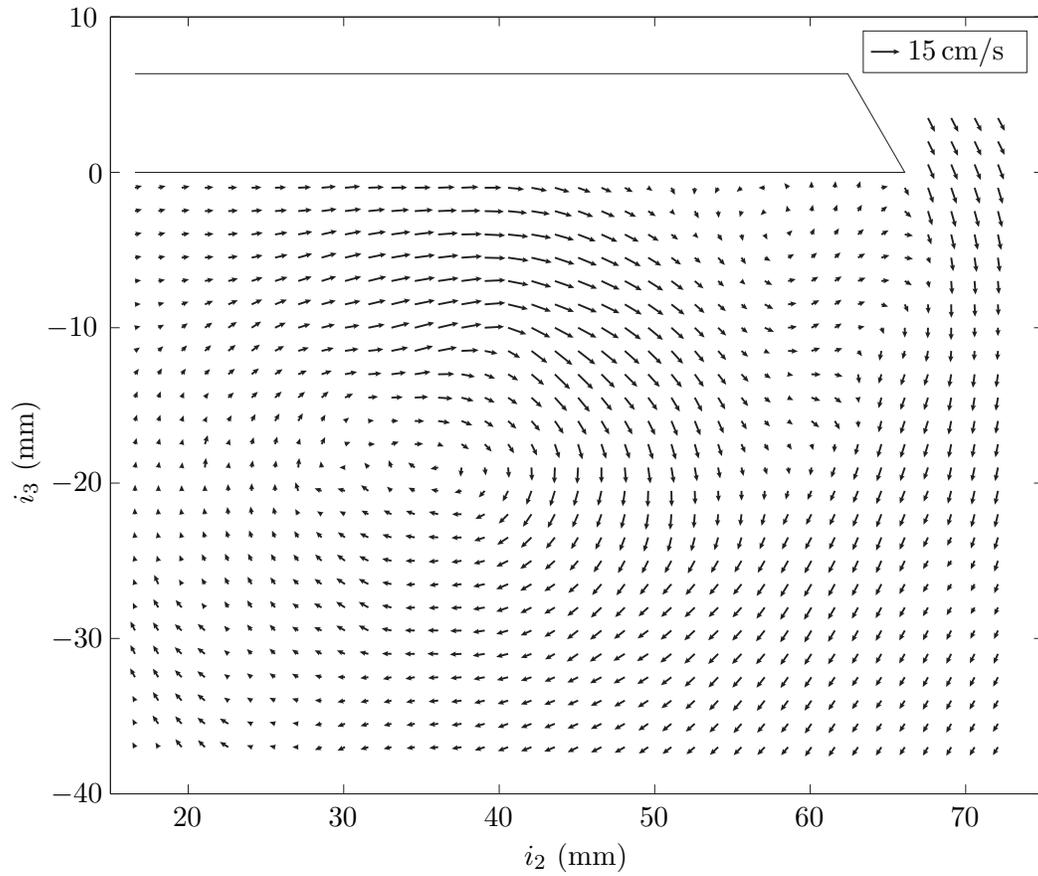


Figure 14: Vector field at $i_1 = 145$ mm showing the vortex generated by the tip of a delta wing.

or large gradients. Since the velocity field is formed directly from the particle positions and not by averaging over several particles, it is important that these particles be found with a great deal of accuracy. In order to achieve high accuracy, a calibration process is essential. This calibration process corrects for the imperfect placement of the CCDs, lens aberrations, and other effects. Since the particle images are blurred they will often overlap on the CCD images. Because of this, a simple search for local maxima will produce very inaccurate results; therefore, the current software performs a least-squares optimization of a fitting function, which produces very good results.

7 Acknowledgments

The authors would like to thank Bahram Valiferdowski for his generous and invaluable assistance in the design and implementation of this research. The project was partially funded by AFOSR (Dr. J. Schmisser) and DURIP. Christopher Mouton's studies were primarily funded by the National Defense Science and Engineering Grant.

References

- [1] C. E. Willert and M. Gharib. Three-dimensional particle imaging with a single camera. *Experiments in Fluids*, 12:353–358, 1992.
- [2] H.-G. Maas, A. Gruen, and D. Papantoniou. Particle tracking velocimetry in three-dimensional flows. *Experiments in Fluids*, 15:133–146, 1993.
- [3] F. Pereira, M. Gharib, D. Dabiri, and D. Modarress. Defocusing digital particle image velocimetry: a 3-component 3-dimensional DPIV measurement technique. application to bubbly flows. *Experiments in Fluids*, Supplement:S78–S84, 2000.
- [4] R. N. Kieft, K. R. A. M. Schreel, G. A. J. van der Plas, and C. C. M. Rindt. The application of a 3D PTV algorithm to a mixed convection flow. *Experiments in Fluids*, 33:603–611, 2002.
- [5] N. F. Ponchaut, C. A. Mouton, and H. G. Hornung. Development of three dimensional particle tracking velocimetry for supersonic flow. In *42nd AIAA Aerospace Sciences Meeting*, Reno, NV, 2004.
- [6] N. Nishino, N. Kasagi, and M. Hirata. Three-dimensional particle tracking velocimetry based on automated digital image processing. *ASME, J. Fluid Engrg.*, 111:384–391, 1989.
- [7] Y. Suzuki and N. Kasagi. Turbulent air-flow measurement with the aid of 3-D particle tracking velocimetry in a curved square bend. *Flow, Turbulence and Combustion*, 63:415–442, 1999.
- [8] Y. Pu, X. Song, and H. Meng. Off-axis holographic particle image velocimetry for diagnosing particulate flows. *Experiments in Fluids*, Suppl.:S117–S128, 2000.
- [9] H. Meng, G. Pan, and S. H. Woodward. Holographic particle image velocimetry: from film to digital recording. *Measurement Science and Technology*, 15:673–685, 2004.

- [10] F. Pereira and M. Gharib. Defocusing digital particle image velocimetry and the three-dimensional characterization of two-phase flows. *Measurement Science and Technology*, 13:683–694, 2002.
- [11] N. F. Ponchaut. *Part I: Three Dimensional Particle Tracking Velocimetry: Advances and Error Analysis*. PhD thesis, California Institute of Technology, 2005.
- [12] H. Stüer, H.-G. Maas, M. Virant, and J. Becker. A volumetric 3D measurement tool for velocity field diagnostics in microgravity experiments. *Measurement Science and Technology*, 10:904–913, 1999.
- [13] S. Murai, H. Nakamura, and Y. Suzuki. Analytical orientation for non-metric camera in the application to terrestrial photogrammetry. In *Int’l Arch. Photogrammetry XXIII, Commission V*, pages 516–525, 1980.
- [14] H.-B. Kim and S.-J. Lee. Performance improvement of two-frame particle tracking velocimetry using a hybrid scheme. *Measurement Science and Technology*, 13:573–582, 2002.
- [15] N. A. Malik, Th. Dracos, and D. A. Papantoniou. Particle tracking velocimetry in three-dimensional flows. part ii: Particle tracking. *Experiments in Fluids*, 15:279–294, 1993.
- [16] J. Willneff. 3D particle tracking velocimetry based on image and object space information. In *ISPRS Commission V Symposium*, Korfu, Greece, 2002.
- [17] L. Kajitani and D. Dabiri. A full three-dimensional characterization of defocusing digital particle image velocimetry. *Measurement Science and Technology*, 16:790–804, 2005.