VALIDATION OF A THREE DIMENSIONAL PARTICLE TRACKING VELOCIMETRY SOFTWARE

A Thesis

by

HRISTO B GOUMNEROV

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Yassin Hassan
Committee Members,	Nagamangala Anand
	William H. Marlow
Head of Department,	Andreas A. Polycarpou

August 2014

Major Subject: Mechanical Engineering

ABSTRACT

In the present research, the three dimensional particle tracking software OpenPTV is validated with synthetic images from the Standard PIV project by the Visualization Society of Japan, and with experimental data from the twin-jet facility at the Laser Diagnostics Multiphase Flow Laboratory in Texas A&M University. OpenPTV is an open source software, initially developed at ETH Zurich and now used among what has become the OpenPTV consortium, a collection of academic institutions with interest in experimental fluid mechanics who want to develop a better software for everyone.

The software is tested with regard to particle detection, particle position and velocity reconstruction in three dimensional space, as well as individual particle trajectory reconstruction. For the experimental assessment, the OpenPTV results are compared with a Laser Doppler Velocimetry study, as well as results from the in-house two dimensional particle tracking velocimetry software. As contributions to the consortium, the author wrote a calibration code to be implemented in the OpenPTV software which will facilitate processing and improve performance as well as a post-processing script which allows ensemble averaging and computes various fluid mechanics metrics.

DEDICATION

To people from all places and backgrounds who would like to contribute to humanity and are in pursuit of finding the ultimate truth.

ACKNOWLEDGEMENTS

I would like to thank my parents, Roumiana Geipel and Boyan Goumnerov for providing me with the opportunity to pursue my dream of higher education and their support throughout my academic study. I would like to thank my family for their perpetual support and guidance in these early years of my life.

I would like to thank all my friends for the positive reinforcement and belief in my abilities as an academic scholar; I thank Rohit and Akshay Singh for helping me take my mind off of work when needed, I thank Jonathan Raftery for being a supportive and understanding roommate, I thank Mike Gorman and Fatih Sarikurt who have made working in the lab a joy, and all my new friends whom I met in College Station.

I would like to thank the OpenPTV consortium, and in particular Dr. Alex Liberzon from Tel Aviv University for the extensive help with the OpenPTV software.

I would like to thank Dr. Elvis Dominguez and Dr. Carlos Estrada-Perez for their help not only with respect to PTV, but also post-processing and technical writing.

I would like to thank my advisor Dr. Yassin Hassan for giving me the opportunity to perform research in the Laser Diagnostics Multiphase Flow Laboratory, as well as my graduate committee members Dr. William H. Marlow and Dr. Nagamangala Anand. Last but not least, I would like to thank all of my instructors and staff members in the Mechanical Engineering department, and the Texas Engineering Experiment Station for positively contributing to my experience at Texas A&M University – College Station.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Page

1. INTRODUCTION

"The essence of engineering is the utilization of the resources and laws of nature to benefit humanity."

1.1 *Various experimental fluid mechanics techniques*

Fluid mechanics is an integral part of nearly all mechanical systems. It is imperative to have a fundamental knowledge of internal and external fluid motion in order to completely characterize and understand a system. An example of such a system is the flow of water along a fuel rod array within a typical PWR reactor. To obtain thermal power parameters, i.e. the critical heat flux in order to assess the heat transfer within the system, requires comprehensive flow data [1].

Traditionally, scholars have studied systems involving some type of flow by conducting an experiment using one or more flow measurement techniques, although more recently the emergence of Computational Fluid Dynamics (CFD) has become a viable alternative. Performing CFD simulations is often significantly more practical and inexpensive, however researchers still conduct fluid flow experiments to study complex systems and to validate simulation results.

Several different operable techniques have been developed for flow measurement distinguished by their applicability, performance and the kind of results they deliver. Flow measurements can be performed intrusively by inserting probes into the flow or non-intrusively. Among today's flow measurement techniques, the most common are classified to these two types in Table 1. The results are velocities in one, two or three dimensions and/or trajectories over time.

Table 1: Overview of different flow measurement techniques [2]

Classification	Method	Type of Measurement
intrusive	Hot Wire Anemometry (HWA) / Constant Temperature Anemometry (CTA)	Single probe location, thermo-electric
	Pulsed Wire Anemometry (PWA)	Single probe location, thermo-electric
non-intrusive	Laser Doppler Anemometry (LDA)	Single probe, optical, particles
	Laser-2-Focus Anemometry (L2F)	Single probe, optical, particles
	Laser Induced Fluorescence (LIF)	Scanning lightsheet, optical, fluorescin
	Particle Image Velocimetry (PIV)	Scanning lightsheet, optical, particles
	Particle Tracking Velocimetry (PTV)	Object volume, optical, particles

The thermo-electric velocity measurement techniques are common and more appropriate for the measurement of time series in one, two (2D) or three dimensional (3D) gas and liquid flows. The temporal resolution is high, whereas the spatial resolution is limited to the number of probe locations.

Hot Wire Anemometry or **Constant Temperature Anemometry** is a well-established technique which provides single point information about the flow velocity. The instantaneous velocity is measured by its cooling effect on a heated sensor (convective heat transfer), and the consequent voltage drop across the sensor. Depending on the setup, this technique can measure velocities ranging from a few cm/s to well above the speed of sound.

Pulsed Wire Anemometry works on the principle of measuring velocity by timing the flight of a passive tracer over a known distance. A pulsed wire emits a heated spot, a small slightly heated region of fluid, which is convected with the instantaneous flow and after a short time sensed by one of two sensor wires which is a positioned a known distance apart from the pulsed wire. PWA is similar to HWA / CTA in regard to having a limited spatial resolution, but differs by the restriction of yielding reasonable results only for small velocities (below 15 m/s).

In contrast, the optical velocity measurement techniques are non-intrusive and hence do not influence the flow directly; however, there may be indirect disturbances to the flow such as heating of the system due to the illumination facility leading to unwanted thermal forcing (of the flow).

Laser Doppler Anemometry is a very common technique used to study gas and liquid flow. Typically, liquids and especially gases are seeded with tracer particles. A photodetector receives light scattered from the tracer particles moving through the intersection volume and converts the light intensity into an electrical current. The scattered light contains a Doppler shift, the Doppler frequency, which is proportional to the velocity component perpendicular to the bisector of the two laser beams. Given the wavelength of the laser light and the angle between the intersecting laser beams, one can compute the Doppler frequency and the velocity using a conversion factor. Adding more beam pairs allows to measure all three velocity components. The advantages of LDA to alternative techniques is a very high temporal resolution and no need for calibration.

Laser-2-Focus Anemometry is a technique in which the velocity of extremely small particles is recorded. As in LDA, the light scattered by the particles when illuminated by a light source is the major measurement. Two highly focused parallel beams are projected; as a particle traverses the beams, it emits two scattering light pulses that are scattered back and are detected by two photodetectors each of which is assigned to a beam in the measuring volume. When the particle traverses both beams, it transmits two scattering signals whose time interval provides a value for the velocity component in the plane perpendicular to the beam axis. Like in LDA, the temporal resolution is relatively high, but the spatial resolution is limited to the number of probe locations.

Laser Induced Fluorescence is a technique suitable for studying mixing processes in turbulent flows. The basic idea of LIF is to add fluorescein to the fluid which absorbs light of a certain wavelength and emits light of a different (usually higher) wavelength as it traverses the laser light sheet. Images, which are synchronized with the scanning, are recorded layer by layer by a high-speed camera in order to generate volume image datasets. Three-dimensional least squares matching can then be used to determine the velocity fields [2].

1.2 *Particle Image Velocimetry*

Particle Image Velocimetry, or PIV, refers to a class of methods used in experimental fluid mechanics to determine instantaneous fields of the vector velocity by measuring the displacements of numerous fine particles that accurately follow the motion of the fluid [3]. Two velocity components are measured, but use of a stereoscopic approach permits all three velocity component to be recorded, resulting in instantaneous three dimensional velocity vectors for the whole target area. The use of modern digital cameras and dedicated computing hardware, results in real-time velocity maps. Figure 1 displays a schematic of a PIV system.

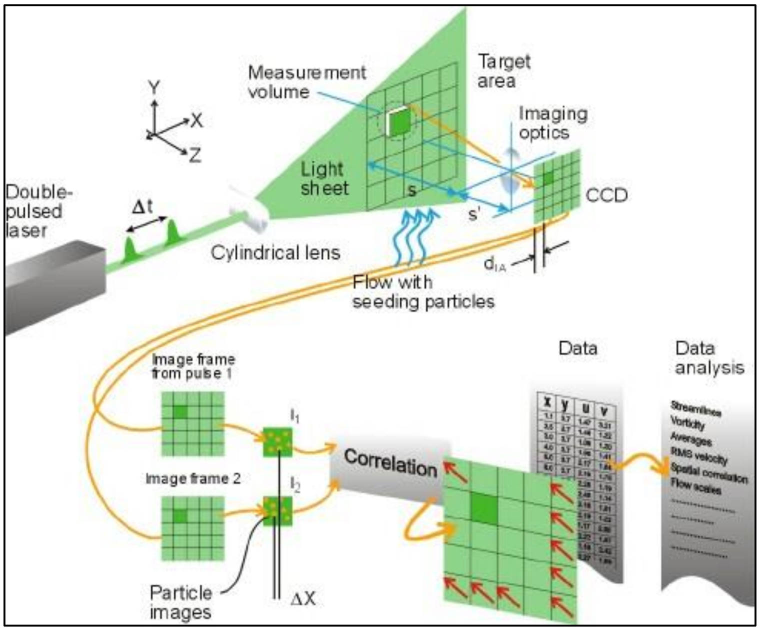


Figure 1: A standard PIV system [4]

The tracer particles are sufficiently small and of similar density as the fluid to accurately follow the fluid motion and not alter the fluid properties or flow characteristics. They are illuminated by means of a thin light sheet generated from a pulsed light source (usually a double-head pulsed laser system), and the light scattered by them is recorded onto two subsequent image frames by a digital imaging device, typically a CCD camera.

The recorded images are processed offline on a digital computer. Fundamentally, the process refers to a cross-correlation analysis of the particle-image patterns in small subdomains, called interrogation regions, between the first and second image frame. The particle-image pattern displacement divided by the image magnification and the time delay between the laser light pulses yields the local fluid velocity. This process is repeated for the entire image domain, which yields the instantaneous velocity in a planar cross section of the observed flow [5].

Some general aspects of PIV are [6]:

- Non-intrusive velocity measurement: PIV is an optical technique and is applicable to studies involving high-speed flows with shocks in boundary layers close to the wall (where the flow may be disturbed by the presence of probes as in the aforementioned techniques).
- Indirect velocity measurement: Like LDA, this technique measures the velocity of a fluid element indirectly by means of the measurement of the velocity of tracer particles within the flow. In two phase flows, particles, in the form of bubbles, are already present in the flow which allows the measurement of the bubble as well as the fluid velocity.
- Whole field technique: In PIV, images of large parts of the flow field(s) in a variety of applications in gaseous and liquid media are recorded. Therefore, the spatial resolution of PIV compared to other flow measurement techniques is higher.

- Velocity lag: Small particles with a density very close to the fluid density will follow the flow better than larger particles with different density. The choice of tracer particles depends on the type of experiment to be performed.
- Illumination: In liquids, "larger" particles are accepted because they scatter more light, thus requiring light sources of relatively low peak power. However, in gaseous flows, a high power source must illuminate the relatively small tracer particles. Moreover, the duration of the illumination light pulse must be short enough to "freeze" the motion of the particles during pulse exposure in order to avoid blurring of the image.
- Time delay between illumination pulses: The time delay must be long enough to be able to determine the displacement between the images of the tracer particles with sufficient resolution and short enough to avoid particles with an out-of-plane velocity component leaving the light sheet between subsequent illuminations.
- Distribution of tracer particles in the flow: A homogeneous distribution of medium density is desired for high quality PIV recordings in order to obtain optimal evaluation.
- Density of tracer particle images: Figure 2 shows the three modes of particle image density. In the case of low image density, case (a), the images of individual particles can be detected and images corresponding to the same particle originating from different illuminations can be identified. This mode requires methods for individual particle tracking, referred to as particle tracking velocimetry (PTV). Medium image density, case (b), is required to apply the standard statistical PIV

evaluation techniques. For high image density, case (c), it is impossible to detect individual images as they overlap and form speckles, here Laser Speckle Velocimetry (LSV) is the appropriate technique to use.

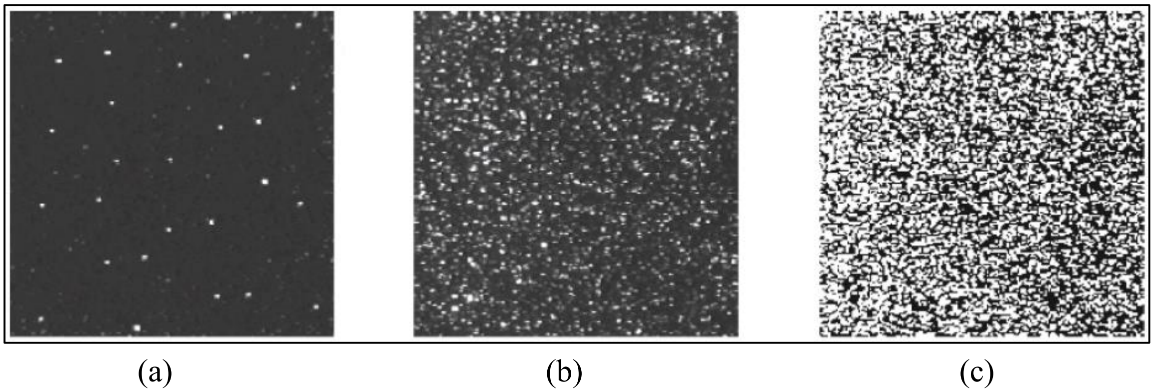
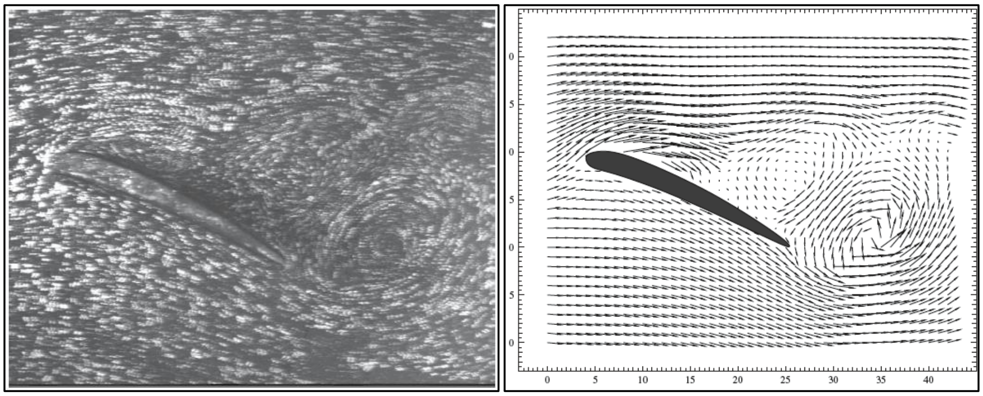


Figure 2: Three modes of particle image density: (a) low (PTV) (b) medium (PIV), and (c) high image density (LSV) [6]

- Number of components of the velocity vector: In the case of planar illumination of the flow field, only two components of the velocity vector can be determined in 2C-PIV. There are methods like stereo, dual-plane and holographic PIV which can extract the third component of the velocity vector; they are labeled 3C-PIV.
- Temporal resolution: Recent developments of high-speed lasers and cameras allow time resolved measurements of most liquid and low-speed aerodynamic flows.
- Spatial resolution: The size of the interrogation areas during evaluation must be small enough for the velocity gradients not to have significant influence on the results. Furthermore, the spatial resolution determines the number of independent

velocity vectors and therefore the maximum spatial resolution of the velocity map which can be obtained at a given spatial resolution of the sensor employed for recording.

An example of an early experimental fluid mechanics study case was done by Ludwig Prandtl, a renowned German engineer and a major contributor to the field of fluid mechanics. He studied the separated flow behind a wing; the corresponding PIV image for this flow (a), and the corresponding vector map (b) from a PIV study is shown in Figure 3.



(a) (b)

Figure 3: Separated flow behind a wing [6]

1.3 *Particle Tracking Velocimetry*

The spatial resolution in PIV evaluation can be even further increased by tracking the individual particle images. While standard PIV can only be used to determine Eulerian flow fields, with particle tracking velocimetry one can also obtain the Lagrangian representation of the flow field – Figure 4.

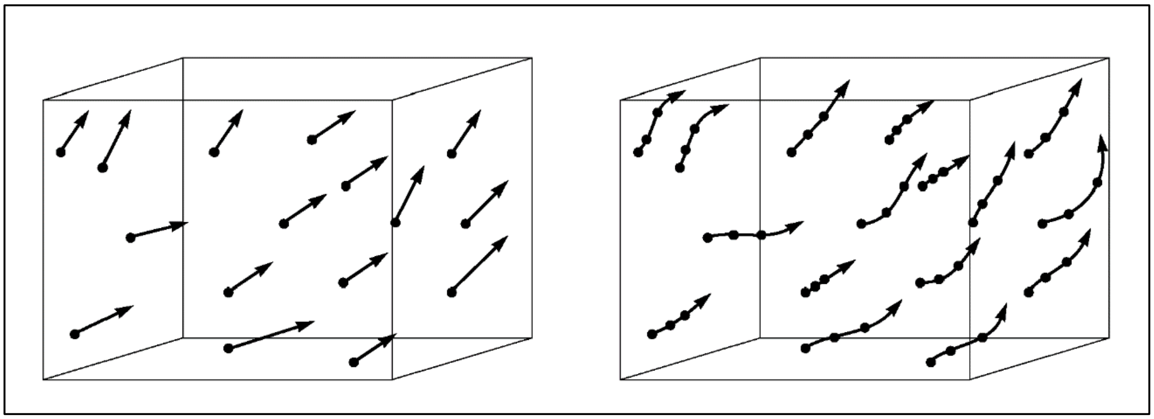


Figure 4: Eulerian (left) and Lagrangian (right) flow field illustration [2]

The advantages of PTV stand out when the flow is three dimensional, and thus requires a three dimensional analysis in order to obtain the velocity field and individual particle trajectories.

Three dimensional PTV, or 3D-PTV, is based on the acquisition of image sequences from different views recording the motion of particles. In contrast to PIV, the aim of PTV is to reconstruct the trajectories of individual particles in three dimensional object space. In general PTV setups, the cameras observe the flow from outside the object volume, thus being located in a different optical medium than the particles. If the motion in a fluid is observed through a glass plate, the optical ray passes through air, glass and fluid and is broken twice according to the refractive indices. Similarly to PIV, after an on- or offline digitization, the image sequence data is processed to segment the particle images and to extract the pixel coordinates of their centroids. If necessary, a high pass filter is applied to remove non-uniformities of the background intensity level. Next follows the assignment of corresponding particle images from different views using what is known as the epipolar constraint and will be explained in the next section. With knowledge of camera orientation data obtained from calibration, the particle correspondences can be established, and it is then possible to determine the 3D particle location in object space. From this point forward one can proceed with a tracking algorithm and reconstruct the particle trajectories. The spatio-temporal matching algorithm used in this study, performs the tracking procedure using information of image and object space simultaneously [2]. Figure 5 shows an illustration of the PTV processing scheme.

In summary, Table 2 presents performance characteristics of the aforementioned flow measurement techniques. Note that PTV goes beyond just yielding vectors, and provides the Lagrangian trajectories of individual particles.

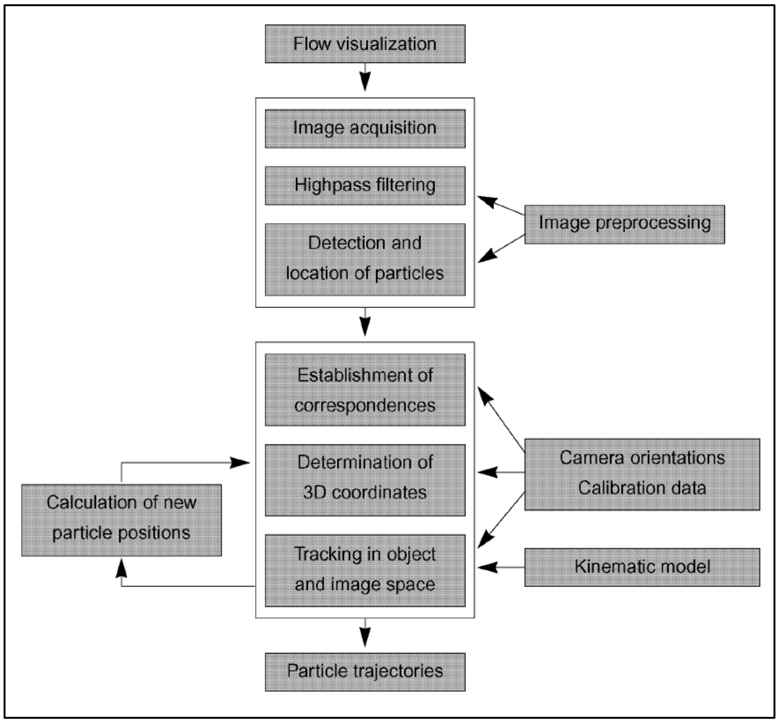

Flow visualization
Image acquisition
Highpass filtering
Detection and location of particles
Image preprocessing
Establishment of correspondences
Determination of 3D coordinates
Tracking in object and image space
Camera orientations Calibration data
Kinematic model
Calculation of new particle positions
Particle trajectories

Figure 5: PTV processing scheme [2]

Table 2: Performance characteristics of flow measurement techniques [2]

Method	Spatial Resolution	Temporal Resolution	Dimension of measurement	Accuracy potential	Results
CTA	Low	Very high	1-3	1-3 % of velocity scale	Vectors
PWA	Low	High	1-3	~ 1 % of velocity scale, 5 % for high turbulence	Vectors
LDA	Low	Very high	1-3		Vectors
L2F	Low	High	1-3	Signal to noise ratio ~ 0.3	Vectors
LIF	Very high	Very low	3	Systematic errors ~ 1-2 %, random errors of > 5-10 %	Vectors
PIV	Very high	Very low	2 (3)	< 1 % of mean flow velocity	Vectors
PTV	High	Low	3	Lateral 1:4000, depth 1:2000 of velocity vector	Trajectories

1.4 *Motivation*

Over the past few decades, the abovementioned measurement techniques have been developed further and improved in order to meet the increasing complexity of engineering systems. In particular, numerous academic and commercial institutions have worked on new and more accurate PIV and PTV algorithms for two and three dimensional studies. The majority of these algorithms are well developed and considered important intellectual property, thus the academic institutions do not usually share their codes. Commercial companies, on the other hand, ask a very high purchasing price for their softwares, and many of them are not as customizable as some laboratories would like in order to better use the product for their applications. It is thus of great interest to the author to study the 3D-PTV code OpenPTV provided by the OpenPTV consortium. The code is an open source software, meaning that anyone is legally permitted to obtain the code and use it for their own purposes. The author promotes the idea of open source research, and believes that it is the only progressive approach to develop better products for the benefit of humanity.

1.5 *OpenPTV*

OpenPTV is a 3D-PTV software based on the core algorithms developed at ETH Zurich. The branches of the Zurich code have been developed independently by the Eindhoven University of Technology (TU/e) group of Turbulence and Vortex Dynamics (C++ version with Tcl/Tk and few new algorithms of general coordinate transformation) and by the Turbulence Structure Laboratory at Tel Aviv University (Python version, PyPTV). Following a meeting at TU/e in October 2012, the three groups (ETH, TU/e and TAU) decided to release their software under open source licenses. The OpenPTV foundation has now become a collaborative effort of several research groups to join in order to develop a better software for 3D-PTV [7].

A sample result of 3D-PTV is shown in Figure 6. A model of an ascending aorta was created and studied using the OpenPTV software. Figure 7 shows a sample of the recovered trajectories.

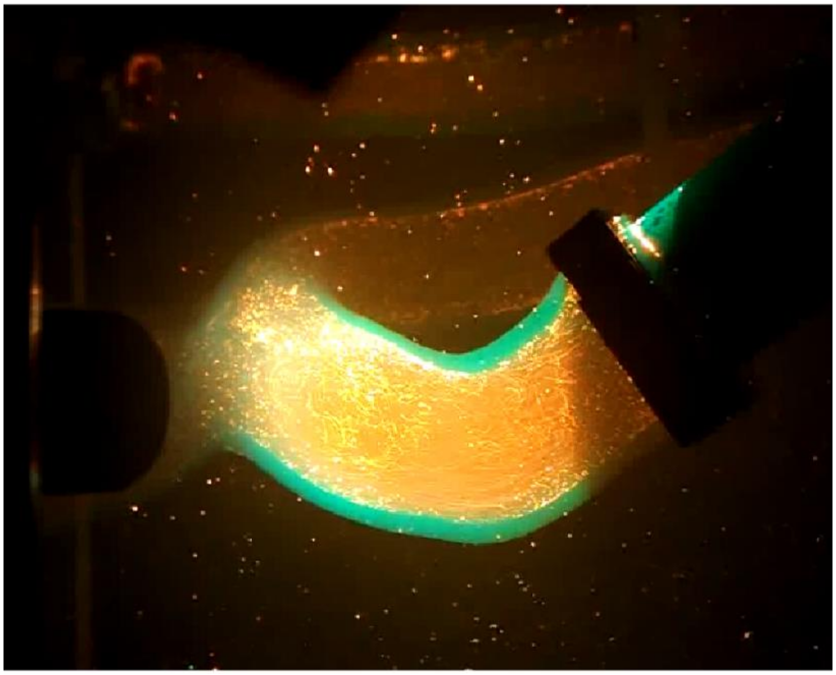


Figure 6: Aortic flow modelled in an ascending aorta [8]

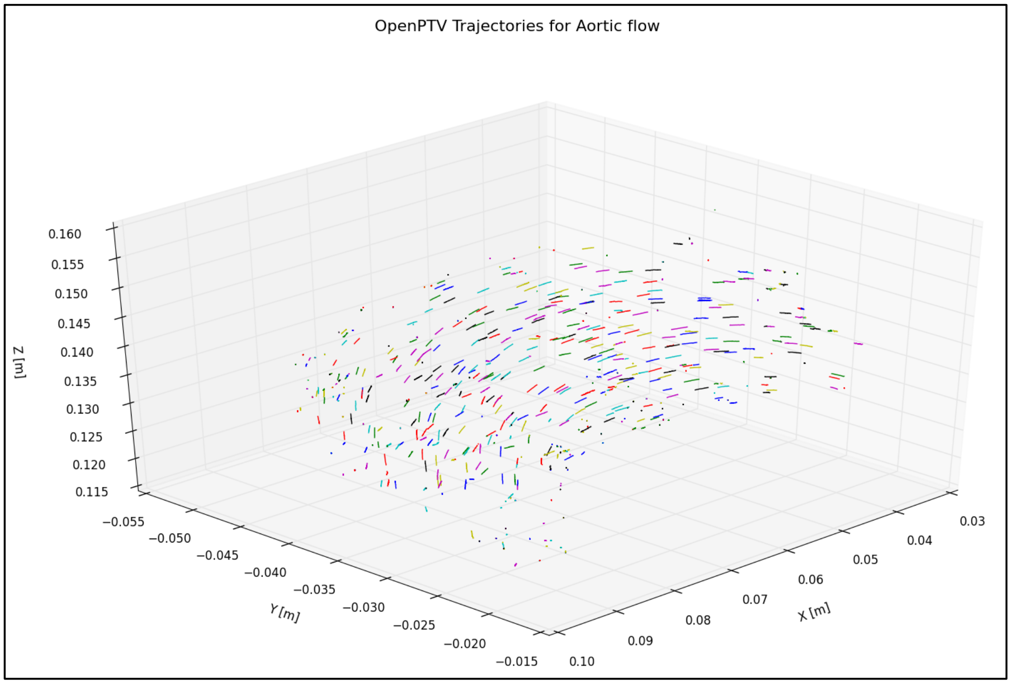


Figure 7: Recovered trajectories of aorta experiment

2. PHOTOGRAMMETRY OF 3D-PTV

2.1 *Pinhole camera model*

The first step to perform a particle tracking study is to develop a model of each camera. In the case of the OpenPTV software, the cameras recording the image sequence are mathematically modeled as a pinhole according to Tsai's calibration technique [9]. Each camera has its own intrinsic parameters, i.e. focal length and lens distortion, and extrinsic parameters corresponding to its orientation and position. The pinhole camera model relates the 3D coordinates of a point in object space to its corresponding 2D projection onto the image plane. This mapping is referred to as perspective projection. The center of the perspective projection is the point at which all rays intersect, and is denoted as the optical center or camera center. The line orthogonal to the image plane and passing through the optical center is called the optical axis. The principal point is the location where the image plane intersects with the optical axis. Figure 8 shows a schematic of the pinhole camera model.

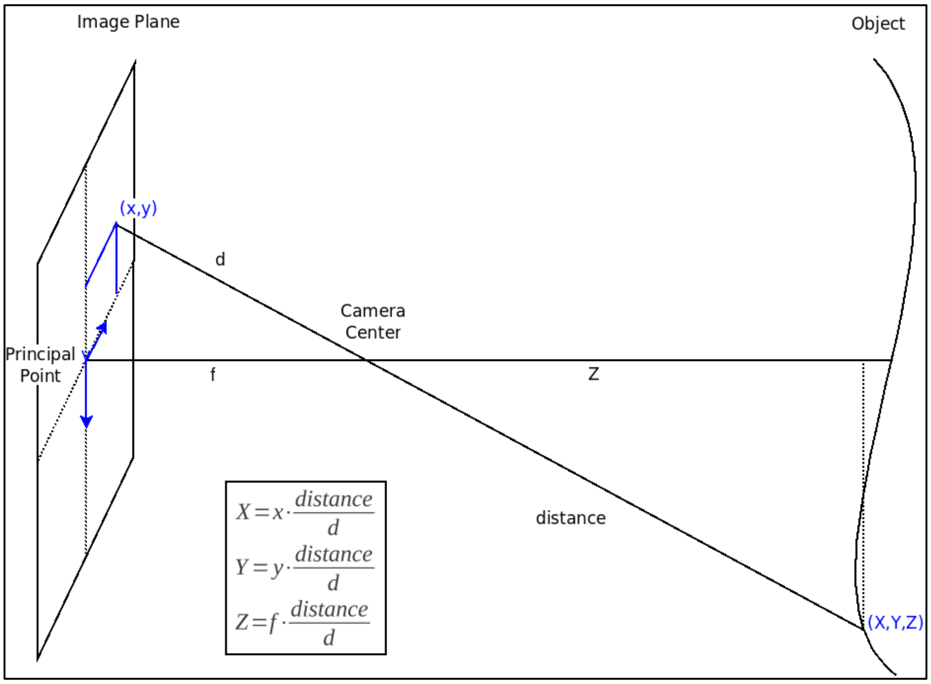


Image Plane
Object
(x,y)
d
Camera
Center
Principal
Point
f
Z
distance
(X,Y,Z)
$X = x \cdot \frac{distance}{d}$
$Y = y \cdot \frac{distance}{d}$
$Z = f \cdot \frac{distance}{d}$

Figure 8: Pinhole camera Model [10]

This model represents an idealized mathematical version of a camera. However, a general PTV image sequence will not be composed of ideal images due to disturbances in the system leading to noise in the image data. In regard to the imaging hardware, this noise comes from aberrations such as chromatic aberration, monochromatic aberration, spherical aberration, coma, astigmatism, Petzval curvature of Field, and distortion among others [3]. An example of distortion is shown in Figure 9. In (a) there is no image distortion, in (b) the image has pincushion distortion, and in (c) the image has barrel distortion. Cases (b) and (c) are caused by variable lateral magnification.

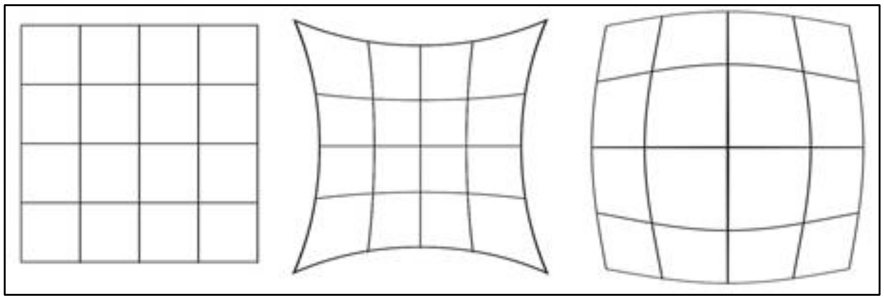


Figure 9: Image distortion example [11]

In addition to the abovementioned aberrations, in PIV and PTV experiments it is a common scenario that the object and image planes are not normal to the optical axis, in other words they are not facing each other. The case of non-parallel planes is called oblique imaging and is another cause of disturbance in the image data. However, a geometrical arrangement known as the Scheimpflug condition can be applied to remedy the non-uniform magnification. By attaching Scheimpflug adaptors to the cameras, the setup essentially satisfies the required geometric arrangement, and the image plane is effectively parallel with the object plane. Only when very high accuracy is desired, one should look for alternative solutions to the use of Scheimpflug mounts [3]. Based on the author's experience, this calibration technique requires information which is not easy to obtain, i.e. focal length, CCD or CMOS sensor offset relative to the projective center of the camera, and distortion parameters among others. An excellent alternative calibration technique is presented in the possible improvements section.

2.2 *Mathematical model*

A comprehensive review of the mathematical model of the OpenPTV software is found in [2] and [12]. The fundamental mathematical model of 3D particle coordinate determination is the collinearity condition: Object point, camera projective center, and image point must lie on a straight line. The formulation includes the three object coordinates X_O, Y_O, Z_O of the projective center, as well as the three angles ω, φ, κ describing the direction of the optical axis, and applies to the idealized pinhole camera model. Figure 10 illustrates the collinearity condition.

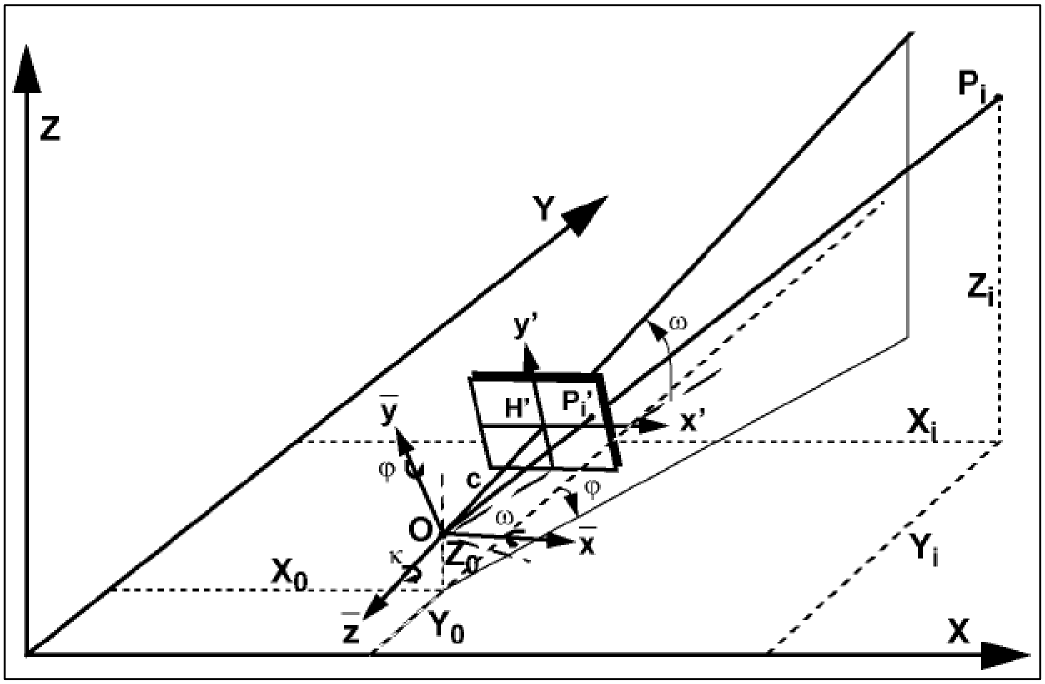


Figure 10: Collinearity condition (camera model inverted) [2]

The OpenPTV software is designed to model lens distortion according to the Brown model with radial symmetric lens distortion (parameters k_1, k_2, k_3) and decentering distortion (parameters p_1, p_2). In addition, the software accounts for multimedia geometry which occurs when there is one or more media (with different refractive indices) between the projective center of the camera and the particle position. The collinearity condition along with the intrinsic and extrinsic parameters of the camera, and the distortion parameters lead to a functional model of 16 parameters with x, y image coordinates as follows:

$$(x, y) = f(X_O, Y_O, Z_O, \omega, \varphi, \kappa, c, x_k, y_k, k_1, k_2, k_3, p_1, p_2, a_1, a_2, X_i, Y_i, Z_i) \qquad (3.1)$$

The parameters are determined through a calibration process, after which it is possible to apply the epipolar constraint to establish correspondences of the particle images from the different cameras [2].

2.3 *Epipolar constraint*

In three dimensional particle tracking velocimetry it is imperative to establish multi-image correspondences. As the tracer particles are of the same size, shape and color, the only available tool to establish the correspondence is through constraints in the epipolar geometry. The OpenPTV software uses this epipolar geometry to automatically establish the appropriate correspondences [12]. With knowledge of the aforementioned parameters one can make use of the coplanarity condition of the oriented image planes as shown in Figure 11; the geometric system can then be solved to obtain a mapping between the object and image coordinates of a particle.

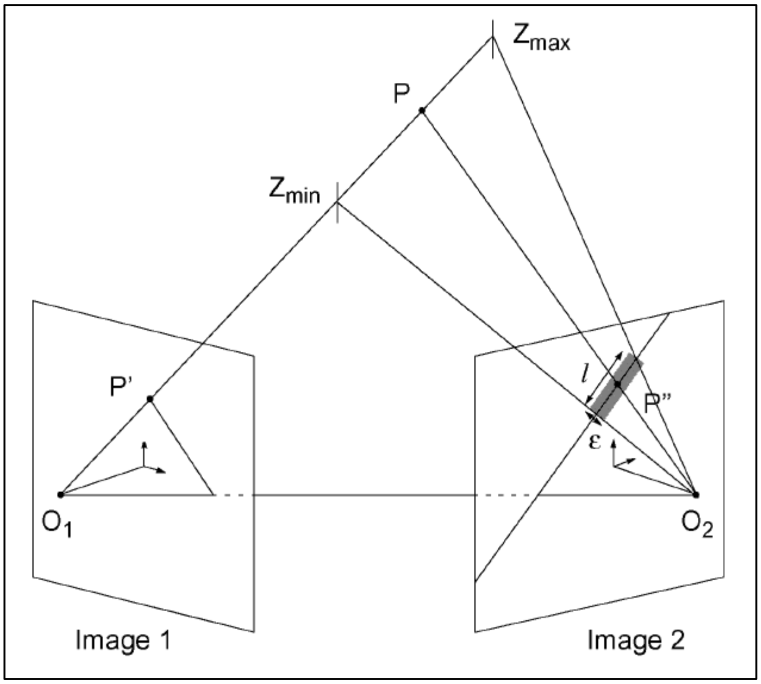


Figure 11: Epipolar geometry in a two-camera setup with intersecting epipolar line segments [2]

For this particular schematic, the coplanarity condition is

$$\overrightarrow{O_1O_2} \cdot (\overrightarrow{O_1P'} \times \overrightarrow{O_2P''}) = 0 \tag{3.2}$$

which yields the respective epipolar line in image space. By proceeding from an image point of the first image, the corresponding search area can be reduced to the epipolar line in the second image. In a general experimental scenario, the search area has to be extended by a tolerance ε to the epipolar line. This tolerance is strongly influenced by the quality of the calibration and the number of particles found in the epipolar search area. Therefore, it is recommended to use three or four cameras to more accurately determine the intersections of epipolar line segments, as the search area is reduced by the additional constraint(s). In the OpenPTV algorithm, the epipolar line intersection is implemented with a combinatorics algorithm to establish unambiguous quadruplets, triplets, and pairs of corresponding particle images [2].

2.4 *OpenPTV tracking Algorithm*

There are three methods to track individual particle trajectories, namely image space based, object space based, and spatio-temporal based tracking.

In image space based techniques the image sequences are acquired using a multi-camera setup and the tracking of particles is performed in two dimensional space with a particular approach (i.e. Fuzzy logic, neural network). The spatial correspondences between the resulting 2D tracks in each image are established in order to reconstruct the object space trajectories.

In object space based techniques the first step is to solve the correspondence of the particles to determine the 3D position, after which an algorithm performs the search of the temporal match in the following frame.

OpenPTV uses a spatio-temporal matching algorithm which combines the abovementioned approaches. The image sequences are recorded, after which the particle correspondences are established using the epipolar constraint. Next, the three dimensional particle positions are determined, and tracking is done in 3D point clouds, where additional positions of unused 2D detected particles are computed. Moreover, the tracking uses kinematic motion modeling with a time-dependent polynomial of order two to perform the tracking. As a result of the combined methods, the reconstructed trajectories are more complete [2]. Figure 12 illustrates the processing scheme of the OpenPTV software, and Figure 13 displays a comparison of using only image or object based techniques compared to the spatio-temporal algorithm.

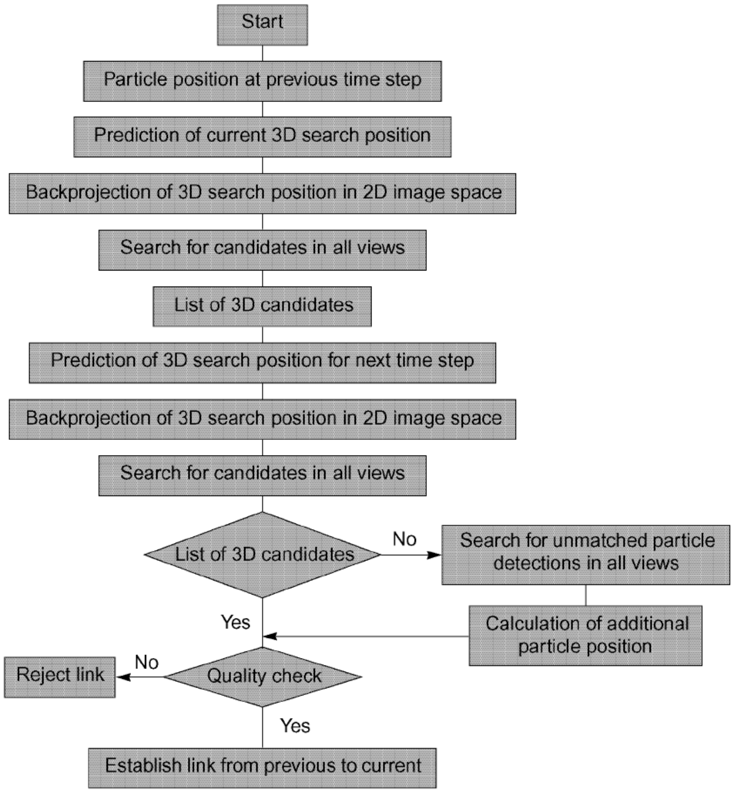

Start
Particle position at previous time step
Prediction of current 3D search position
Backprojection of 3D search position in 2D image space
Search for candidates in all views
List of 3D candidates
Prediction of 3D search position for next time step
Backprojection of 3D search position in 2D image space
Search for candidates in all views
List of 3D candidates
No
Search for unmatched particle detections in all views
Calculation of additional particle position
Yes
Reject link
No
Quality check
Yes
Establish link from previous to current

Figure 12: OpenPTV Processing scheme

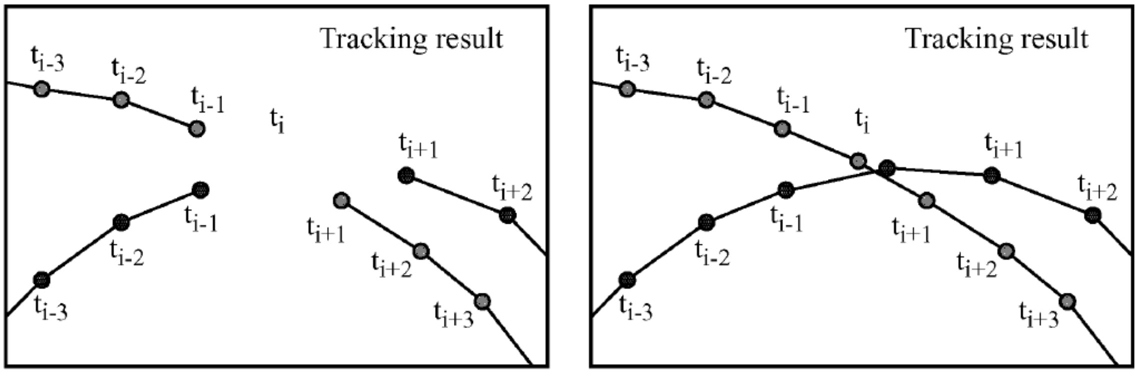


Figure 13: Incomplete and complete trajectories

2.5 *OpenPTV out-of-plane position and velocity component reconstruction*

When the three or four (depending on the number of cameras) particle pixel positions have been found, the three dimensional position of the particle can be determined through a mapping function

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = F^{-1}\left(X^{(N)}\right) \tag{3.3}$$

with

$$X^{(N)} = \left(X_1, Y_1, X_2, Y_2, ..., X_N, Y_N\right)^T \tag{3.4}$$

The mapping function is obtained from the calibration procedure. Note that mathematically only two positions are necessary to obtain the three dimensional position of the particle, however the redundancy by using all *N* image positions generally improves the accuracy of the reconstruction [3]. The complete mathematical reconstruction algorithm is presented in [12].

3. 3D-PIV STANDARD IMAGES

3.1 *Background*

In order to assess a software, the best approach is to test its performance on benchmark problems. In 1996, the Visualization Society of Japan (VSJ) initiated a project for PIV standardization and popularization, called JPIV. The activities of the JPIV are the following:

- Development of Standard images for 2D and transient 3D PIV
- Analysis using the Standard Experimental Problems
- Developments of PIV Database

Various test cases ranging from two component two dimensional (2C2D) data to three component three dimensional (3C3D) data were created using computer graphics and studied using a three dimensional Large Eddy Simulation (LES) to obtain the velocity vector [13]. The simulation was identical for all test cases, namely a jet impinging on a wall as shown in Figure 14.

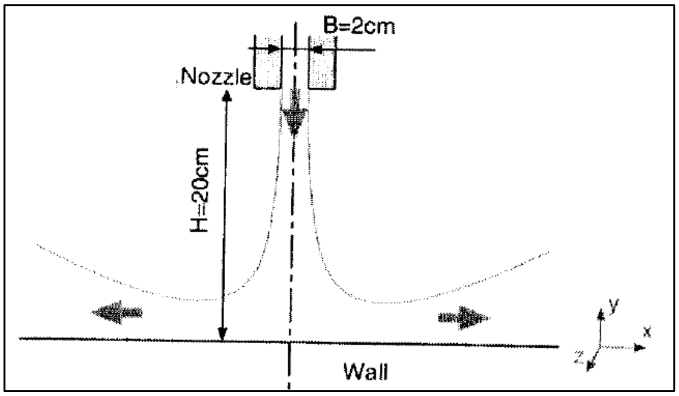
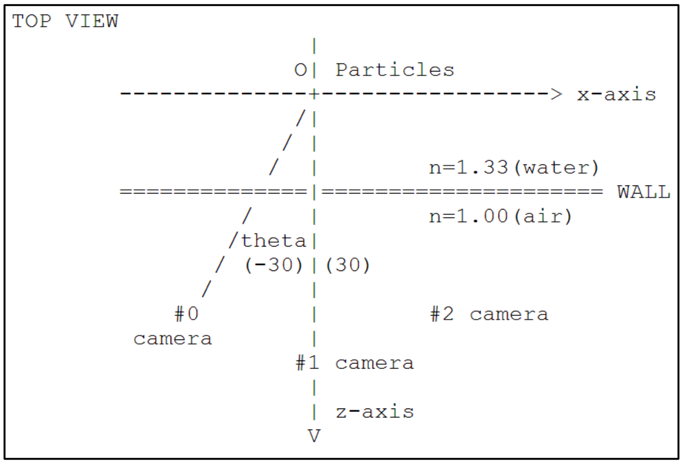


Figure 14: Schematic of general geometry

Parameters such as jet velocity and Reynolds number vary from case to case. In order to assess the OpenPTV software, the experimental or simulated data has to be 3C2D or 3C3D with a low image density. Therefore, test case 352 was chosen as the most suitable case to study. Table 3 shows relevant information about test case 352, and Figure 15 shows a schematic of the simulated camera setup.

Table 3: Standard PIV test case overview

Test Case No.	352
Number of cameras	3
Number of images	145
Image Size	256 x 256 pixel
Interval	0.005 sec
Reynolds Number	3000
Maximum velocity	0.12 m/s
Number of average particles per image	320
Particle Diameter	5 pixel
Water Refractive Index	1.33
Air Refractive Index	1.0



```
TOP VIEW
                       |
                      O|  Particles
        ---------------+-----------------> x-axis
                      /|
                     / |
                    /  |        n=1.33(water)
        ==============|===================== WALL
                  /    |        n=1.00(air)
                 /theta|
                / (-30)|(30)
               /       |
             #0        |        #2 camera
          camera       |
                      #1 camera
                       |
                       | z-axis
                       V
```

Figure 15: Schematic of camera setup

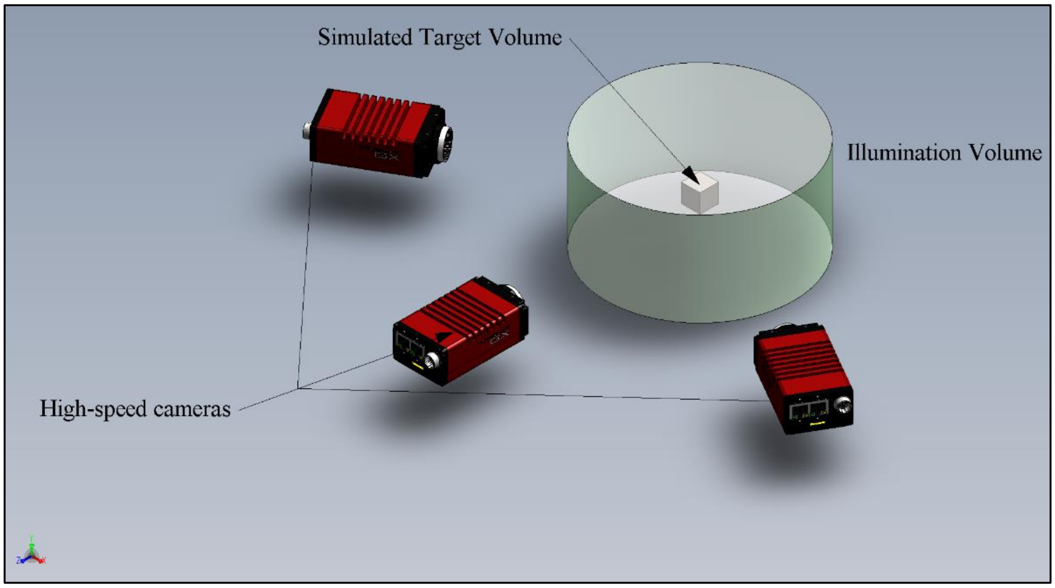


Figure 16: CAD model of camera setup - 1

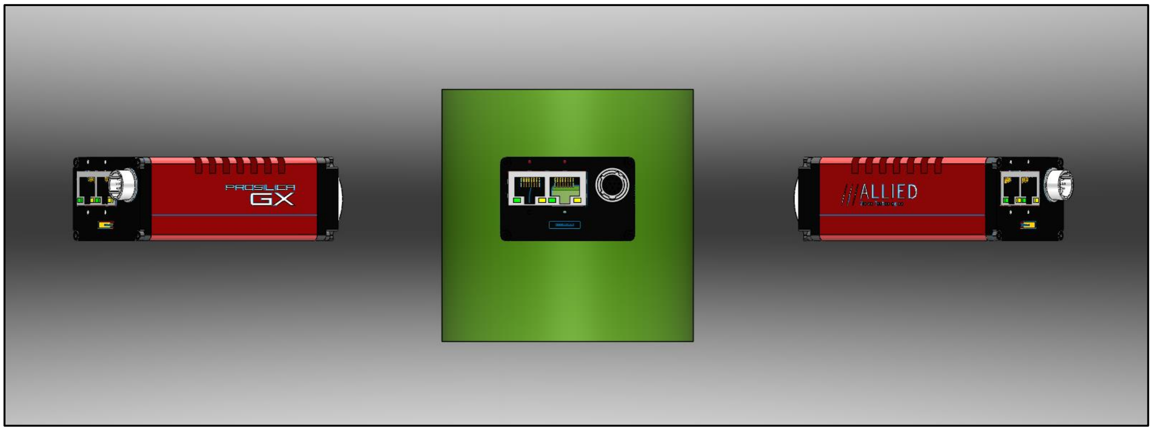


Figure 17: CAD model - 2

Figure 16 and Figure 17 show two different views of a CAD model of the camera orientation, as well as the illumination region and the target volume. The left and right cameras are at a 30 degree angle with respect to the Y-axis, and all three cameras are in the same horizontal plane (Y = 0). Note that the cameras in the images are not the cameras used in the study, but simply pre-built CAD models of a high speed camera model. As described in [12] this orientation is unfavorable due to the fact that no epipolar matches can be made at the center line and close to it.

In addition to the 145 images per camera, a calibration image for each camera was provided showing the respective view of the artificial calibration target. The target is a cube of length 16 mm, with target particle locations at each edge and at the midpoints, as shown in Figure 18.

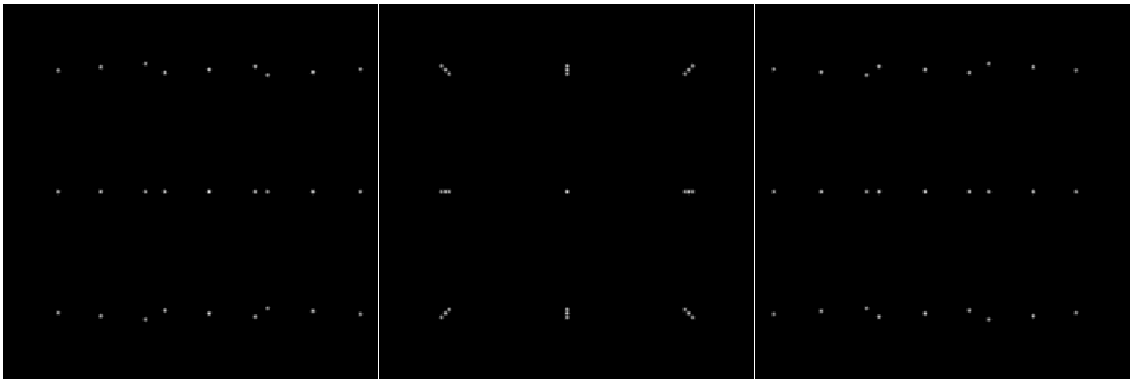

(a) (b) (c)

Figure 18: Calibration images for each camera: (a) left (b) center (c) right

3.2 *Image and data preparation*

Before the test case can be run in the OpenPTV software, the images have to be processed, that is, they have to be converted from their original .raw format to a .tiff format. Adobe Photoshop is the most popular image processing software on the market and was chosen by the author for simplicity and processing speed. Next the images were renamed from their original im0___.tiff to a file name readable by the OpenPTV code, namely cam1.10000, cam1.10001, cam1.10002, and similarly for cam2.10000, and cam3.10000, etc. This concluded the image preparation step.

An additional step was required to obtain the vectors for each individual particle in each frame. Due to the fact that the test cases were designed for PIV rather than PTV, the vector files provided as reference did not correspond to the particles themselves, but to systematically chosen grid locations as illustrated in Figure 19.

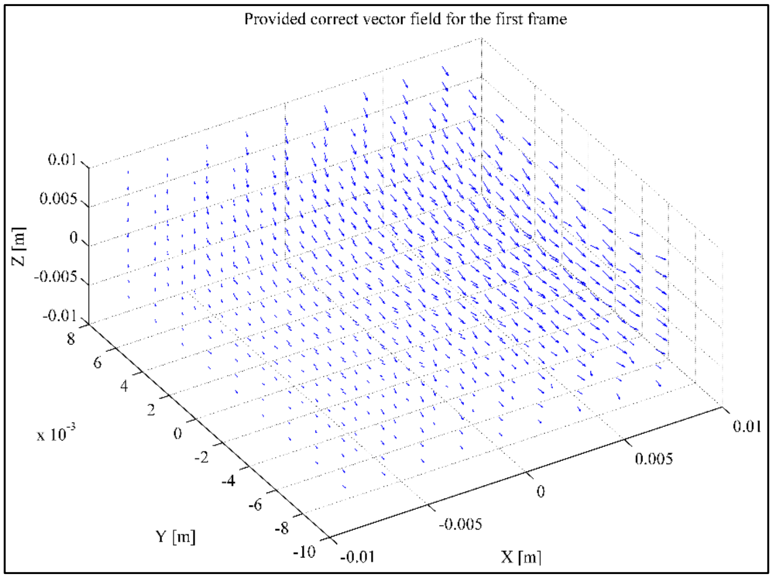


Figure 19: Illustration of vector field from provided vector files

This information is unsuitable for comparison with the output of the OpenPTV software which provides the vectors of individual particles. However, the creators of the test cases provided .dat files with the object space location of each particle in each image along with an ID number associated to it. Using this ID number, the author created vector files with vectors corresponding to individual particles by matching the particles from one frame to the next, determining the displacement and dividing by the time interval. This was done for all three dimensions, yielding 144 files with an ID for each particle as well as its 3D position and velocity for the respective frame. MATLAB was used to process and create the resulting files; the script used is attached in the Appendix. An illustration of a sample vector field with "correct" vectors is shown in Figure 20.

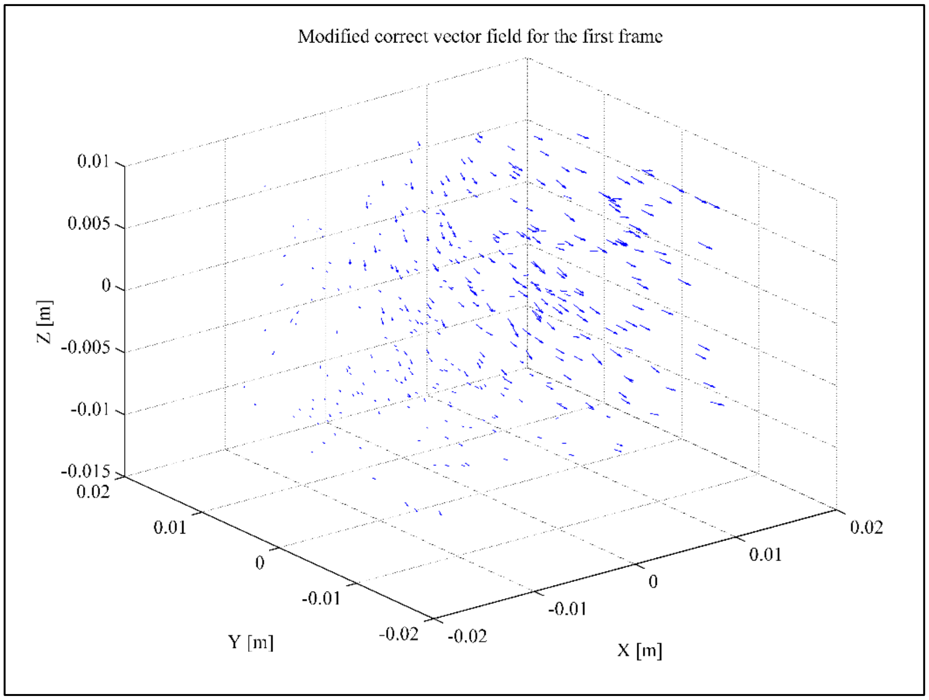


Figure 20: Correct vector field for individual particles

3.3 *OpenPTV calibration, parameter tuning, and post-processing*

The first step in the PTV process is the correct calibration of the software in order to obtain the appropriate camera parameters for the experiment and improve the epipolar matching algorithm to increase quality and quantity of correspondences among particles from the three different views. In accordance with the software, four points are selected from the calibration target and clicked in each camera calibration image to obtain the 16 parameters. The calibration process was kept as simple as possible, changing only those parameters which seem to affect the calibration results. It is assumed that there are no distortions or aberrations of any kind as the images are synthetic. Moreover, as no thickness of the wall in Figure 15 is provided, the thickness is set to 0 and the index of refraction is set to that of water, namely 1.33. This modification removed a shift in the Z-direction which is present if a wall thickness is set.

After the calibration results are deemed satisfactory, parameters relating to particle recognition and criteria for correspondences, and finally tracking were tuned to obtain optimal results. Over 100 runs, varying one parameter at a time were performed to determine the performance of the software. The observed trends are presented in the Results section.

When the optimal settings are determined, the code is run for the whole image sequence. Several different file types are output by the software, and have to be further post-processed in order to obtain the velocity and trajectory information of each particle. A post-processing script with several parameters of its own is provided with the software and can be modified to improve the accuracy of the results. For example, one can set the length of the fitting polynomial which filters the positions, velocities and accelerations; another important parameter is the radius of interpolation which sets how large the interpolation domain around a point should be. The script is run until the results can no longer be improved, according to the suggested criteria.

The final output of the OpenPTV software are files for each frame which include information about particle position, velocities in all three dimensions, accelerations in all three dimensions, as well as trajectory information. As shown in Table 2, no other flow measurement technique provides information about the full velocity gradient tensor and particle trajectories apart from photogrammetric PTV, or 3D-PTV.

4. TWIN JET EXPERIMENT

4.1 *Background*

Texas A&M University is currently conducting research on the University of Tennessee's twin-jet tank with the support of the Department of Energy. The outer dimensions of the tank base are 1016 mm length by 762 mm width. A CAD model of the tank is shown in Figure 21 [14]. The fluid used is water with seeding particles, and the cross sectional area of each jet is 508 mm^2. The facility uses two pumps which can create Reynolds numbers close to 25,000.

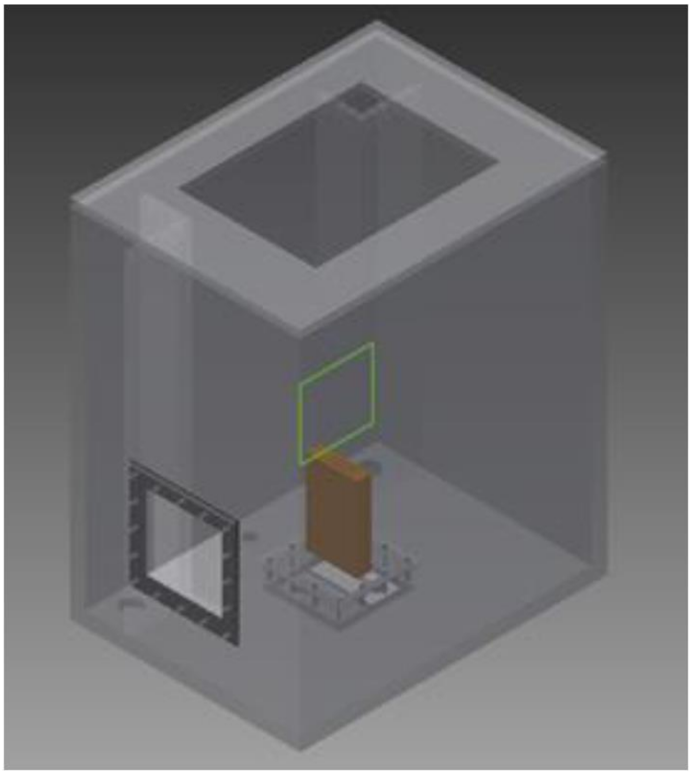


Figure 21: CAD model of twin-jet tank

This tank was chosen as a good set up to assess the performance of OpenPTV by comparing its results with a LDV, and a 2D-PTV analysis. In this study the pump flow rates were set to 10 $^{g}\!/_{\min}$ which yields an average velocity of 1.24 $^{m}\!/_{s}$ at the jet outlet.

4.2 *Camera set-up*

The author had access to three high-speed cameras which were set to run in a synchronized manner with each other as well as the laser illuminating the target volume. Figure 22 shows the camera set up from behind looking at the calibration target in the filled tank.

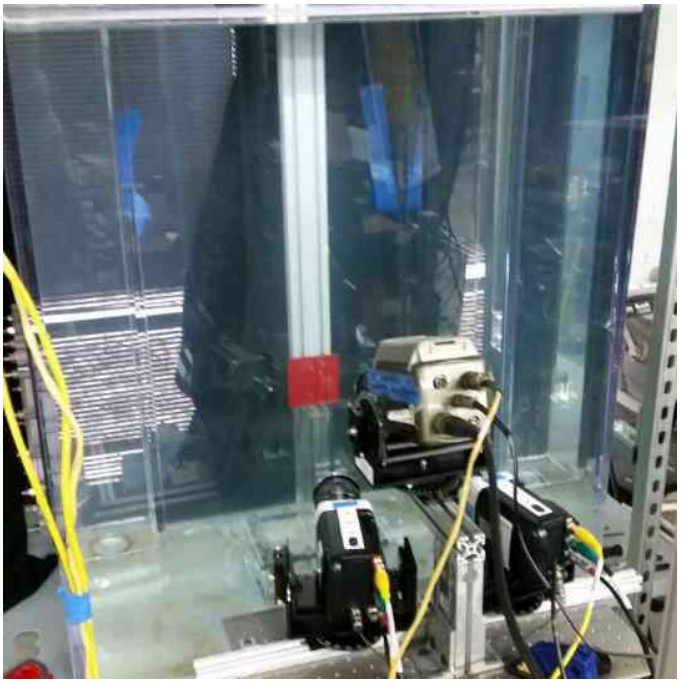


Figure 22: Camera set up with target – 1

The distance between the camera lenses and the center of the calibration target were measured to be 21.5 inches for the left and right cameras, and about 20 inches for the top camera. The angles of the left and right cameras with respect to the Y axis were -12 and 12 degrees respectively, while the top camera had an inclination of -20 degrees (X-axis). The triangular set-up was chosen as it is recommended by [12]. The laser used to illuminate the tracer particles is shown in Figure 23.

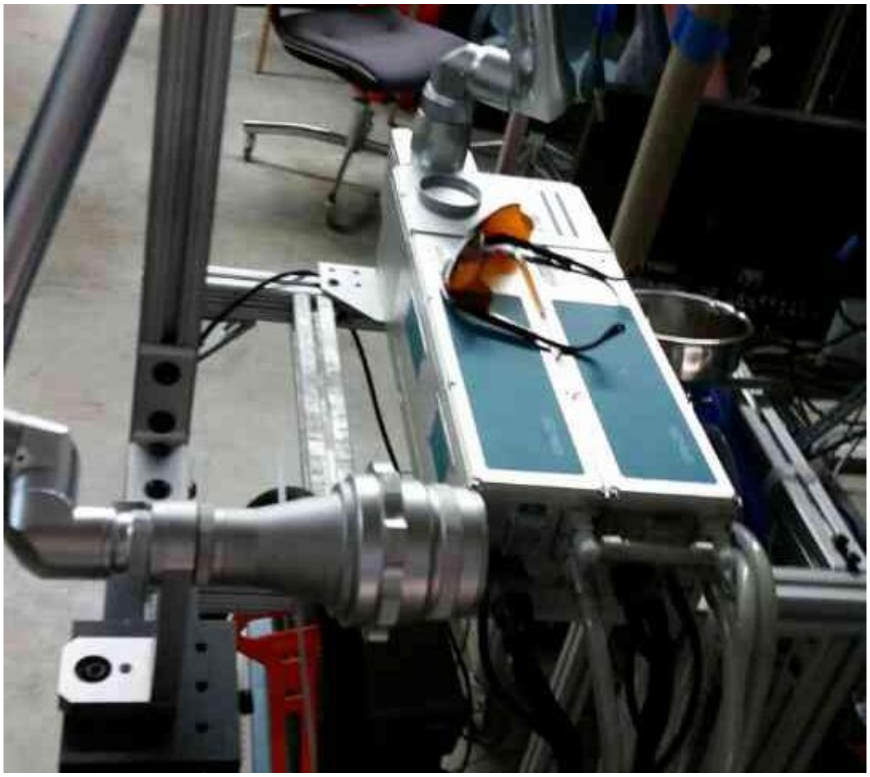


Figure 23: Laser

In order to reduce any background noise, the tank facility was covered up with black covers to stop any ambient light from entering the fluid domain, as shown in Figure 24.

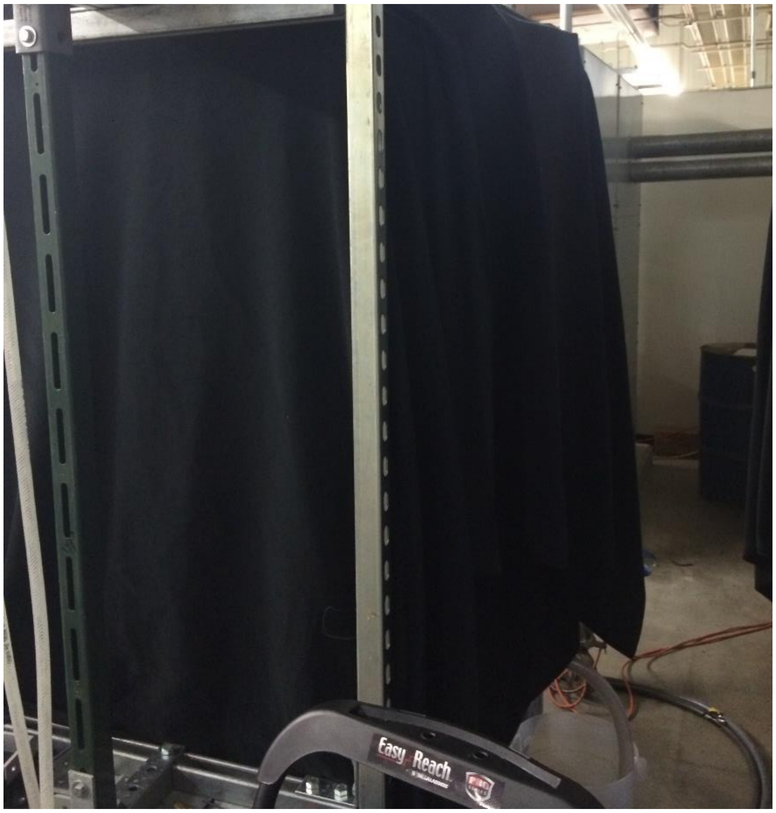


Figure 24: Covered tank

The cameras have a high sensitivity so disturbances have to be kept to a minimum. Initial images with the laser running were taken in order to determine the quality of the images and whether OpenPTV will be able to process them. Figure 25 shows the first images, and that a modification is needed to ensure uniform illumination.

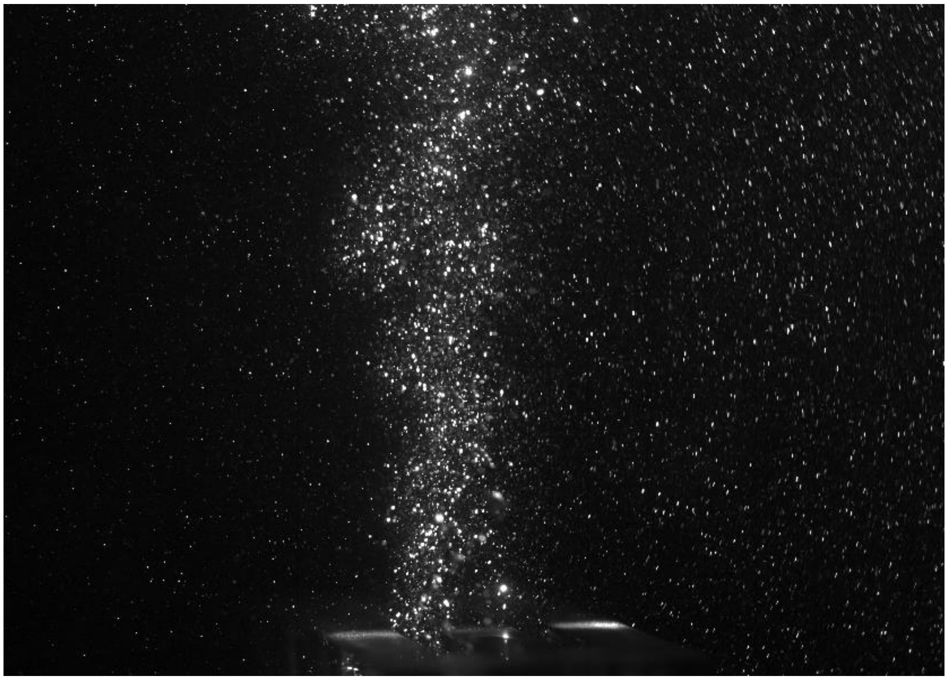


Figure 25: Frame from left camera

It is easy to observe that the right side of the image is brighter than the left side. Therefore, the tracking software will have trouble detecting and tracking particles from one half of the image in all three views. Moreover, the flow coming from the jet is very bright, but it is not due to the light scattered by the particles; the pumps were producing a relatively high flow rate (a tentative 18 $g/_{\min}$) which resulted in air being entrapped as the fluid comes down from the top which is then released through the jet as bubbles. Although this is perhaps favorable to study two-phase flow, this scenario is unfavorable for the current study as the research is to assess the code, and not to study two-phase flow. Therefore, the flow rate was reduced to the abovementioned 10 $g/_{\min}$ and the laser was moved to the top of the tank, and a plate was inserted at the top of the water level to prevent distortion of the laser light due to waves. Figure 26 presents the currently used set-up.

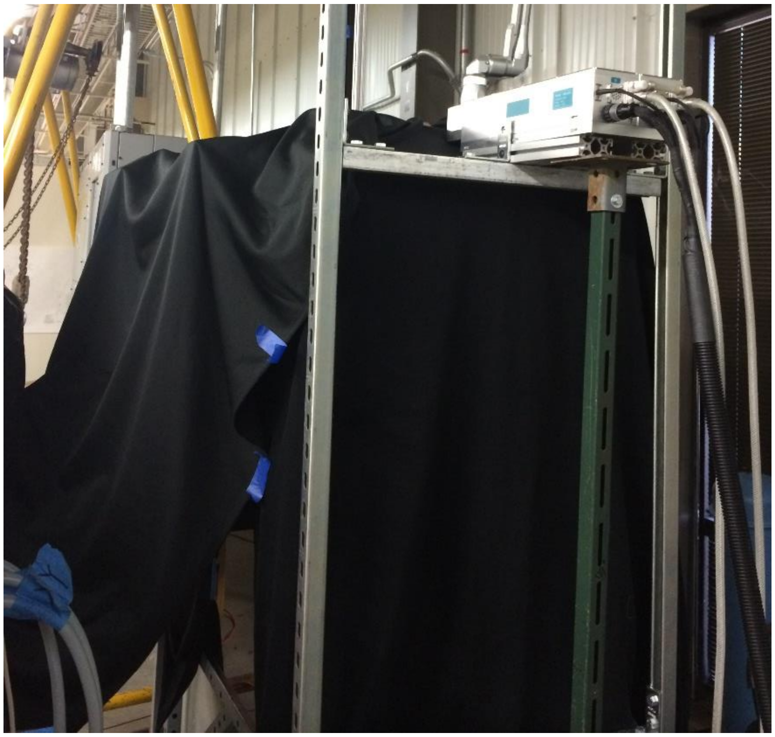


Figure 26: Laser on top of tank

The modification in set up improved the image quality dramatically as can be seen in Figure 27 which shows a frame from the right camera.

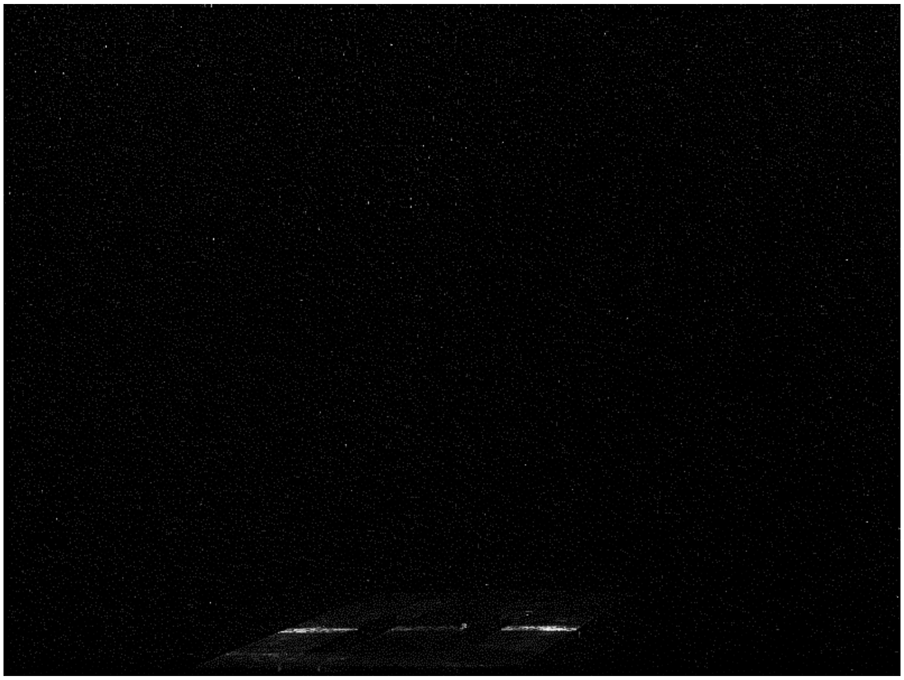


Figure 27: Uniform illumination

The image is darker due to the fact that the flow rate was reduced, thus removing the occurrence of bubbles. In addition, the aperture of the cameras was decreased in order to make use of the power of the laser, which works with a higher stability when run at higher power. The image density is held low, as required by a PTV study. More on the images is presented in the next subsection.

4.3 *Image pre-processing*

The camera data is sent to the provided Phantom software. The videos are set to a have an image size of 800 by 600, and are initially saved in the native .cine format, and later to .bmp files for the 2D-PTV and .tiff for the OpenPTV software, respectively. The first images taken were the calibration images. Figure 28 shows the 3D printed calibration target with dimensions 10 cm by 10 cm by 3/8 inches thickness, and three depth levels in order to be able to select target points at different Z locations with respect to the origin of the target area (in this study this was chosen to be the center point of the calibration target).

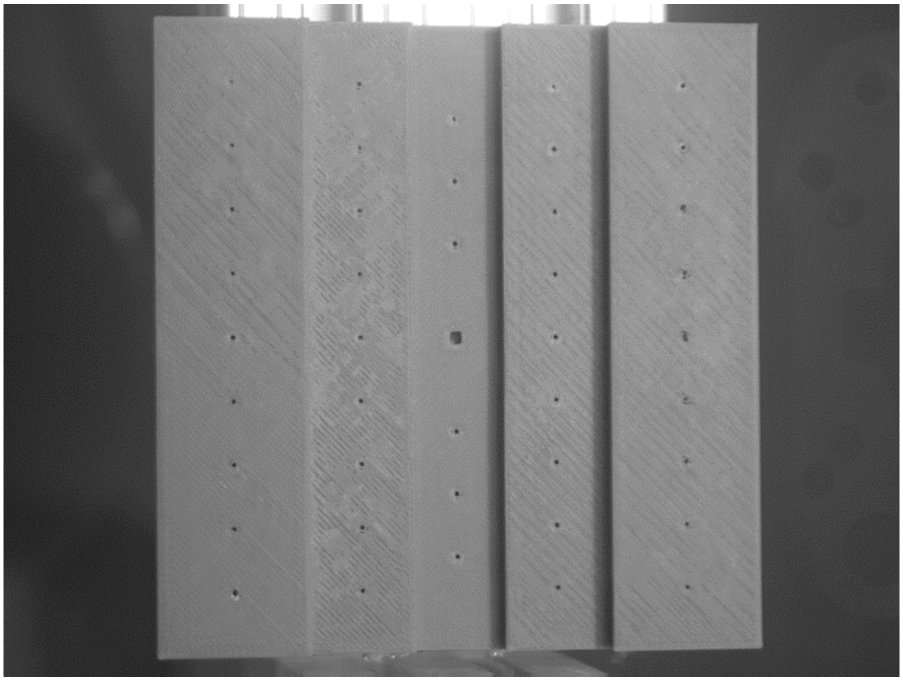


Figure 28: Calibration target as seen by left camera

The user can choose from a total of 43 target points with known object space coordinates. Each camera has one calibration image associated with it and is used in the calibration procedure of the OpenPTV processing. The camera orientation is necessary in order to determine a mapping between the object space coordinates and the image coordinates, which can then be used to reconstruct the particle positions and velocities. Moreover, the orientation information is used to create the epipolar geometry to find the correspondence between the three sets of pixel coordinates for each particle in an image. Figure 29 shows the beginning of the calibration process for the right camera, when the yellow "guessed" calibration points still do not match the positions of the target points.

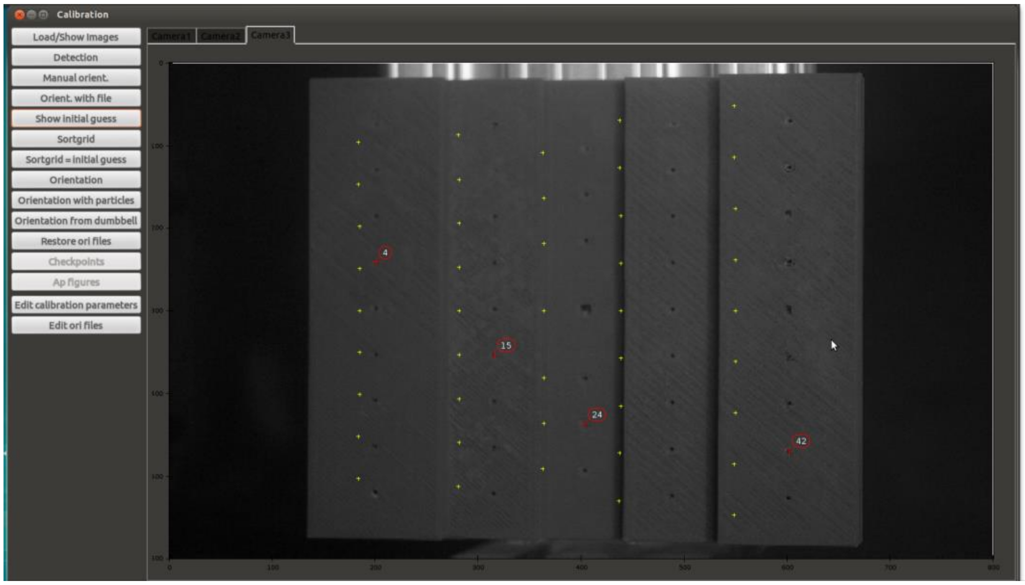
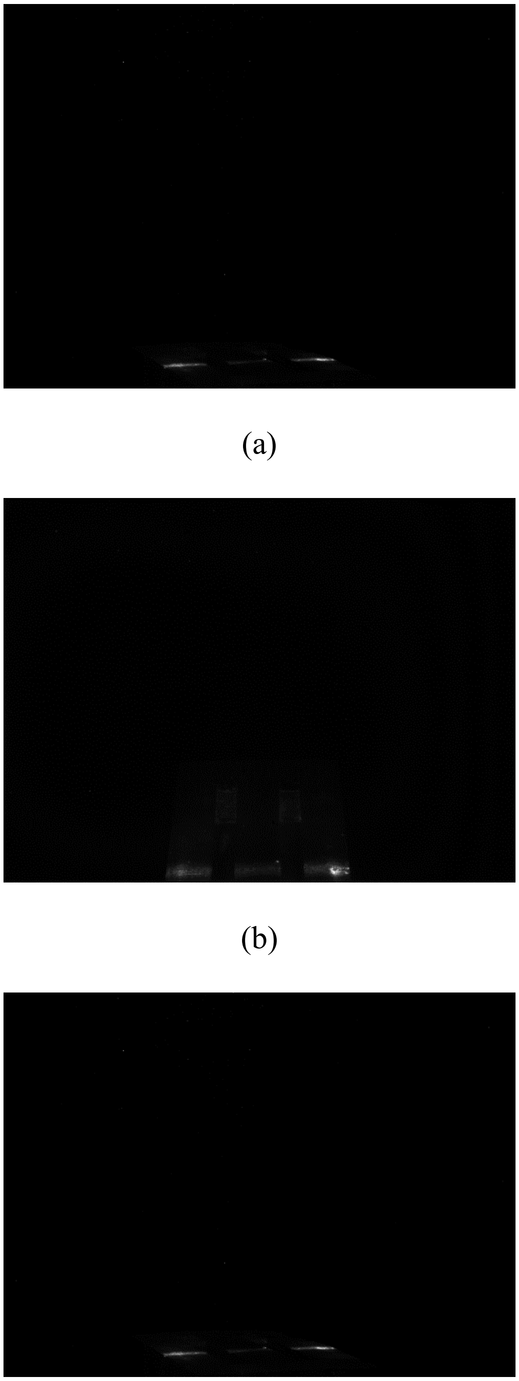


Figure 29: Calibration of right camera

After the dots match the correct locations in each view, the calibration is complete and the images can be processed. The first frame from all three views is shown in Figure 30.



(a)

(b)

(c)

Figure 30: First frame from all three views

5. RESULTS

5.1 *3D-PIV Standard Images*

5.1.1 **Pre-analysis**

Before the post-processing analysis, the author wanted to familiarize himself with the OpenPTV software. To do that, the author investigated the roughly forty parameters which have to be set in order to run the software. Figure 31 and Figure 32 show some of the resulting trends of two selected parameters.

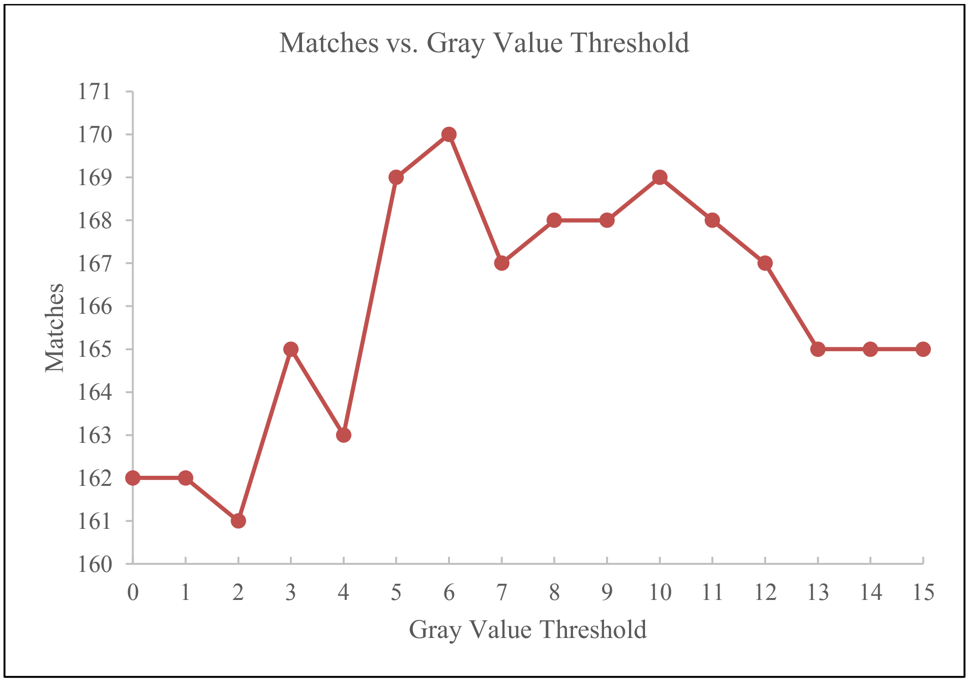


Figure 31: Matches versus. threshold

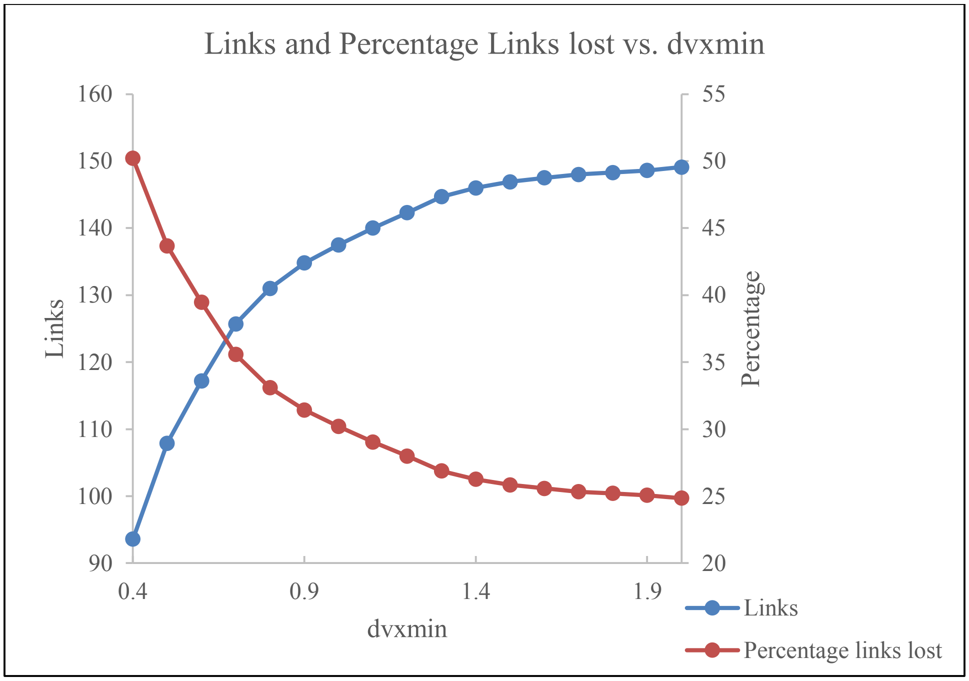


Figure 32: Links established versus. minimum particle displacement in X direction in mm

5.1.2 Particle centroids

A preliminary check whether the software is working, is to check the particle locations in image coordinates and verify them with the reference data. This confirms the effectiveness of the detection algorithm. The author wrote a MATLAB script in which the pixel locations from the OpenPTV output are matched with the correct data. The matching criteria was based on a maximum difference between the X and Y values, and the various epsilon values are shown in Figure 33.

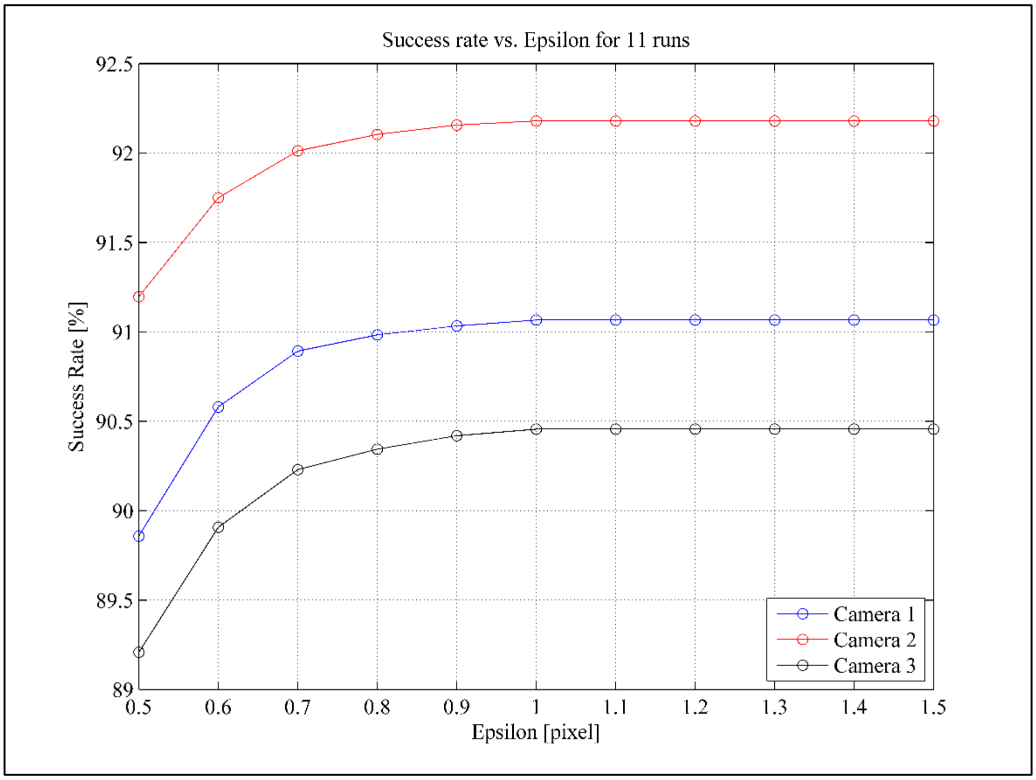


Figure 33: Matching success rate versus. maximum allowable difference

It is clear that at an epsilon of 1 pixel most of the particles are matched, with the highest success ratio for camera 2. The corresponding matching success per frame for an epsilon of 1 is shown in Figure 34. The matching is slightly higher in the first 80 frames which seems to be a trend in the further analysis. Table 4 shows the average percentage of found particles per frame using the OpenPTV code and the percentage matched particles compared to the correct data as well as the errors for X and Y in pixel. Figure 35 displays the particle pixel locations from OpenPTV overlaid on the correct data. One can easily observe that there are a number of unmatched particles at the boundaries. This might be due to the fact that the algorithm is not run for the complete 256 (depends on image size) arrays of information However, it is possible that with a different matching criteria or method, the performance could be improved.

Table 4: OpenPTV centroid detection results

Camera	Percent found [%]	Percent matched [%]	Error in x [pixel]	Error in y [pixel]
Camera 1	93.34	91.07	0.2518	0.2522
Camera 2	93.80	92.18	0.2528	0.2516
Camera 3	92.88	90.46	0.2522	0.2524

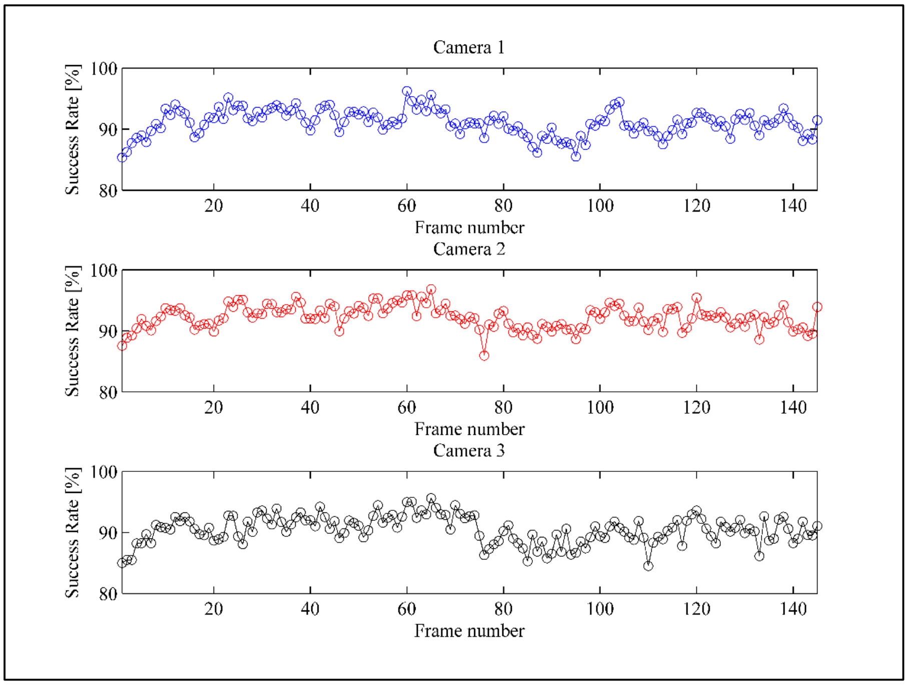


Figure 34: OpenPTV matching success per frame

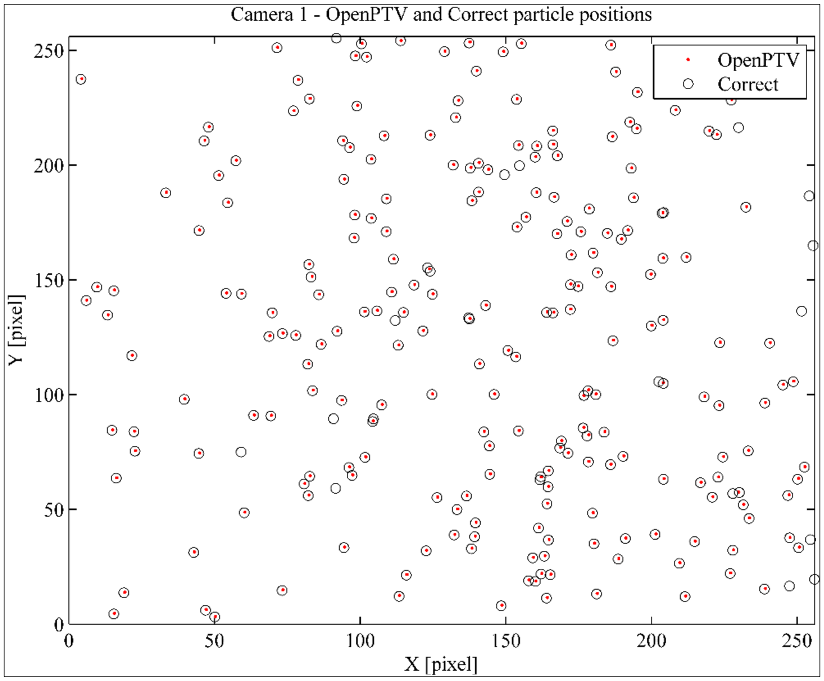


Figure 35: OpenPTV and correct pixel locations

The results seem satisfactory, as only about 9.5 % of the information is lost. Note that the synthetic images were modeled with a maximum (grayscale) intensity according to

$$I\left(x,y\right)=I_0\exp\left(-\frac{\left(x-x_p\right)^2+\left(y-y_p\right)^2}{\left({d_p}/{2}\right)^2}\right) \qquad 5.1$$

$$I_0=240\exp\left(-\frac{{z_p}^2}{{\sigma_p}^2}\right) \qquad 5.2$$

where $\left(x_p, y_p, z_p\right)$ is the particle location with size d_p and laser-sheet thickness $2\sigma_p$.

This is a model used to reconstruct the gray scale intensity and pixel locations of a particle with given object space coordinates. However, equation 5.2 may lead to inaccuracies in the reconstruction of the Z component as it is not a direct mapping. Moreover, some information might have been lost in the conversion from the .raw format to the .tif format, which could lead to difficulty in particle detection.

In addition to comparing the OpenPTV results to the correct data, the in-house two dimensional PTV code used in the laboratory was run with the same images.

Table 5: 2D PTV results

Camera	Percent found [%]	Percent matched [%]	Error in x [pixel]	Error in y [pixel]
Camera 1	88.97	74.67	0.2624	0.2813
Camera 2	80.92	75.65	0.2600	0.2792
Camera 3	89.02	74.98	0.2625	0.2827

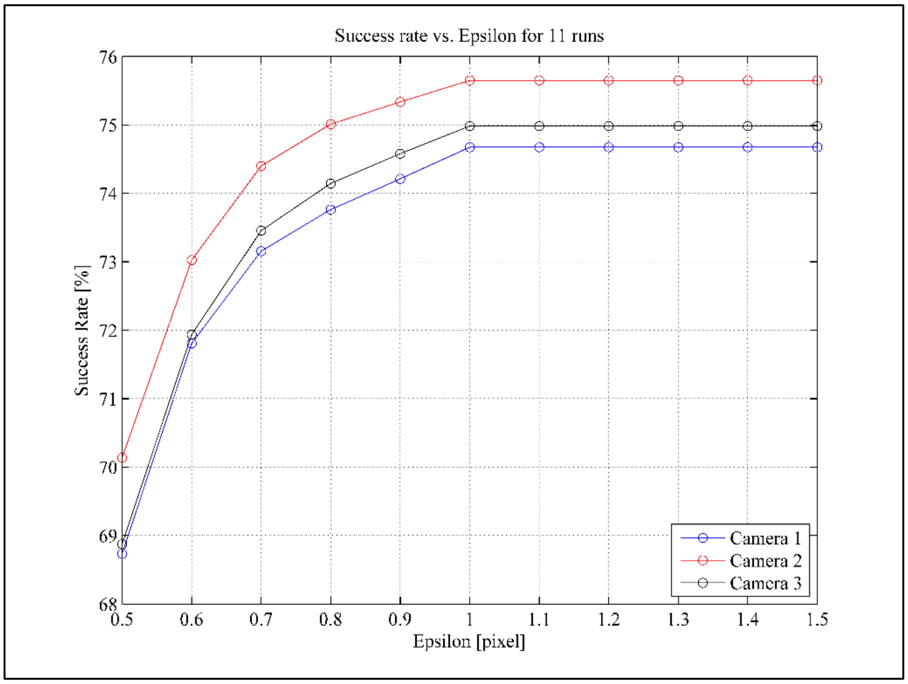


Figure 36: 2D PTV matching success rate vs. epsilon

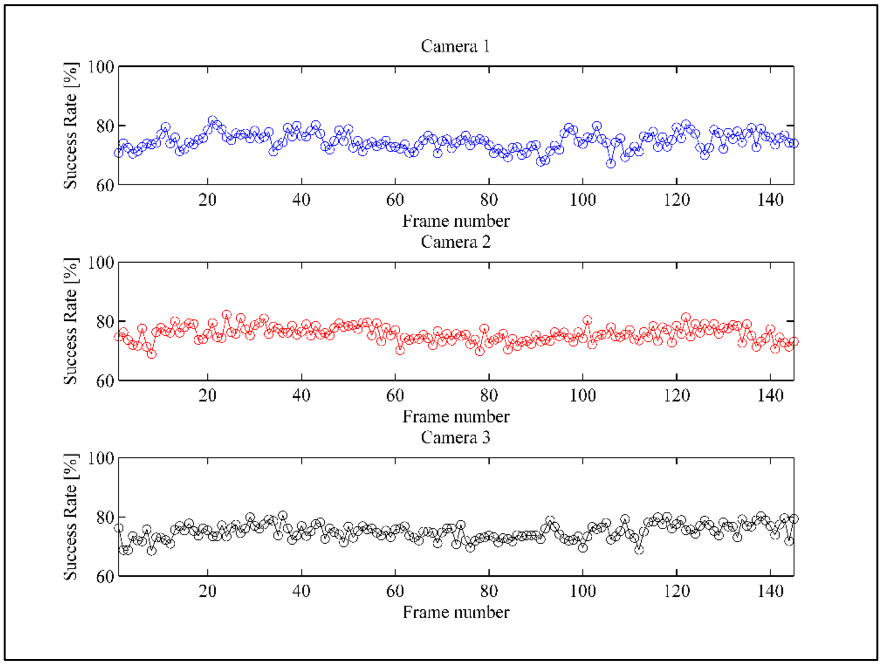


Figure 37: 2D PTV matching success per frame

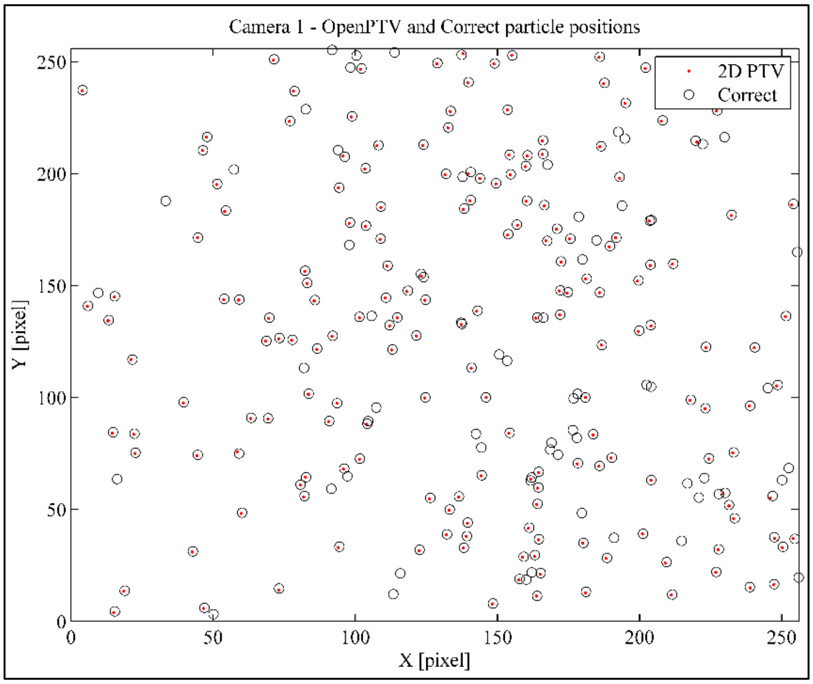


Figure 38: 2D PTV particle locations

As this code has fewer parameters, i.e. tolerable discontinuity and particle size in the two the two coordinates in pixels to name two, it is more difficult to find all particles, particularly when two particles are overlapping each other. Figure 36 shows the success rate for the same range of epsilon. Figure 37 shows the matching success rate frame, and Figure 38 shows the particle positions. There are fewer particles found, and code also does not see the particles on the boundaries.

Table 5 shows the corresponding percentages and errors. The author believes that both the OpenPTV and the 2D PTV codes do not compute the centroid locations at the lower and upper limits of the X and Y pixel directions. An important note is that the correct data includes particles which are not necessarily in all three views simultaneously. These particles have been removed for all statistical investigations.

5.1.3 **Object space coordinate matching**

The next step in the PTV procedure using the OpenPTV code is the determination of the particles object space coordinates. This is where the epipolar constraint is used in order to reconstruct the X, Y, and Z coordinates. Based on the camera orientation, it is clear that all the particles on the horizontal center line will not be processed, as all three cameras are in the same horizontal plane. From [12] one can gain an understanding why this is the case, and why the set up for the twin jet experiment was chosen as suggested from the reference.

The last output files from the OpenPTV software are ptv_is. files which contain the temporal and spatial information for each particle. From these files one can obtain the velocity, acceleration and trajectory information. The author ran another matching process to verify how many particles were found and lost from the correspondence procedure. This was done to confirm the effectiveness of the tracking and reconstruction algorithm(s). As mentioned previously, 9.5 % of the total information of all particles in the images was lost during the particle detection. Moreover, the unfavorable camera orientation is expected to decrease the number of successful matches. After choosing a range for the matching criteria, the author determined that the maximum number of successful matches with respect to the three object space coordinates is roughly 81.7 % as shown in Figure 39. This constitutes an 8.8 % loss from the centroid detection. An important trend is discovered in

Figure 40 which investigates the percentage of successful matches per frame for an epsilon of 0.0018 m.

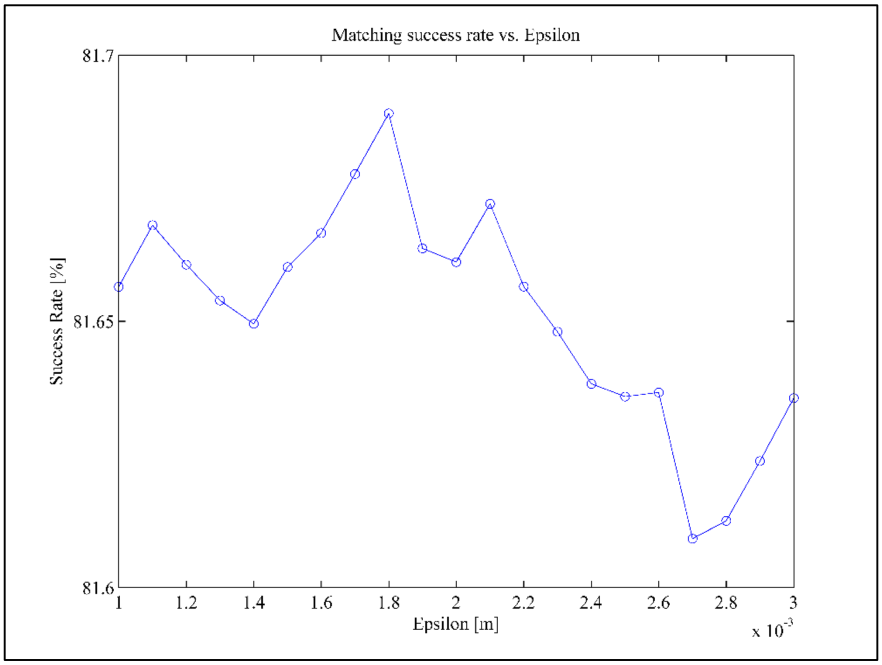


Figure 39: Matching success rate versus. epsilon for object space coordinates

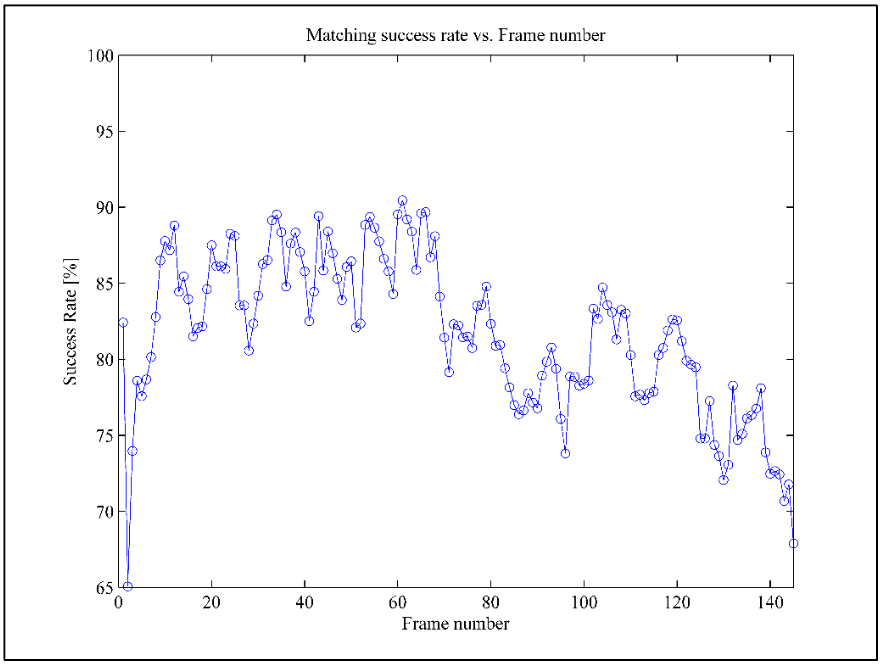


Figure 40: Matching success rate per frame

From Figure 40 it is clear that the OpenPTV algorithm reconstructs the object coordinates with lower success as the frame number increases. The same trend was found in the centroid determination, but to a much lesser extent. This suggests that the velocity reconstruction will show a similar trend. A scatter plot of the matched object coordinates is shown in Figure 41.

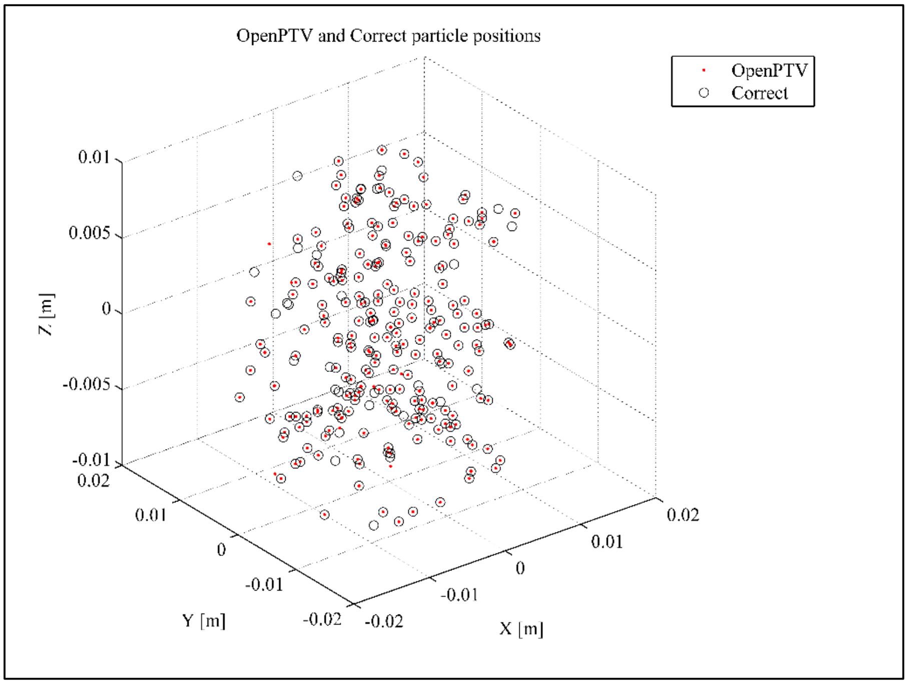


Figure 41: 3D Scatter plot of matched particle positions

From the decreasing number of matches, one might think that the error between the correct and reconstructed positions is rather large. However, in Table 6, the errors for each coordinate suggest that the algorithm works with excellent performance. The errors for position are on the orders of micrometers, while the errors for velocities are on the order of millimeters. For example, if the position of a particle is 0.3 cm in X, the algorithm is accurate within 1 percent. Considering scenarios where the study is involving turbulent flow, a 1 % error is an extremely low uncertainty.

Table 6: Object coordinate metrics

Metric	Value
Error in X [µm]	6.446
Error in Y [µm]	6.355
Error in Z [µm]	11.993
Number of Particles	28,942

Moreover, these values were determined for almost 30,000 instances. Despite, the expected difficulties in reconstruction from incomplete calibration information and inappropriate camera set-up, the OpenPTV algorithm reconstructs the particle positions with remarkable accuracy.

5.1.4 Velocity reconstruction

As mentioned in subsection 4.3, the final output files from the OpenPTV software hold information about the particle for the previous and following frame, as well as the object space coordinates. However, no velocity information is computed. Based on the author's understanding, the majority of the research being done using OpenPTV is Lagrangian statistics which focuses on the particle trajectories, rather than the velocity field. In order to obtain the velocity from the ptv_is. files, the OpenPTV consortium provides a post processing script which allows for parameter tuning to compute the velocity and acceleration data. The output files are xuap.* files which contain raw and filtered X, Y, and Z positions as well as filtered velocity and acceleration data. After successfully running the post processing script, the author performed another matching routine to determine whether the code maintained the 81.7 % volume of information or if there was further data loss, by matching purely based on the X, Y, and Z positions, then by setting varying epsilons for each direction, and finally by including the velocity information. No difference was found by using any of the three approaches. After analyzing the output files yielded from different parameter tuning and matching criteria, the author observes that there is a large data loss, on average about 20 %. Of particular interest is the percentage error in W velocity which should be on the order of 15 % as given in the test case information. The post processing script provides results which match roughly 53 % of all correct particles with a percent error of 15 %. The error can be decreased, but it would be at the expense of successfully matched particles. The question of whether it is really necessary to match the particles. In an experimental study there would be no "correct" data, so no matching routine can be performed. Since the research is about the assessment of the OpenPTV code, the author believed that a matching routine would be beneficial to provide the error for the particle positions and velocities. Due to the low matching success rate, the author wrote a script to obtain the velocity from the ptv_is. files to perhaps increase the match rate. Figure 42 shows the matching rate for the provided script, and Figure 43 shows the equivalent for the author's script.

Table 7 displays the average number of particles matched, as well as the errors for X, Y, Z, U, V, and W. Using the author's script, a total of 28,730 particles with velocities were found, and using the provided script the corresponding number is 24,796, out of a total of 47,345 based on the correct data.

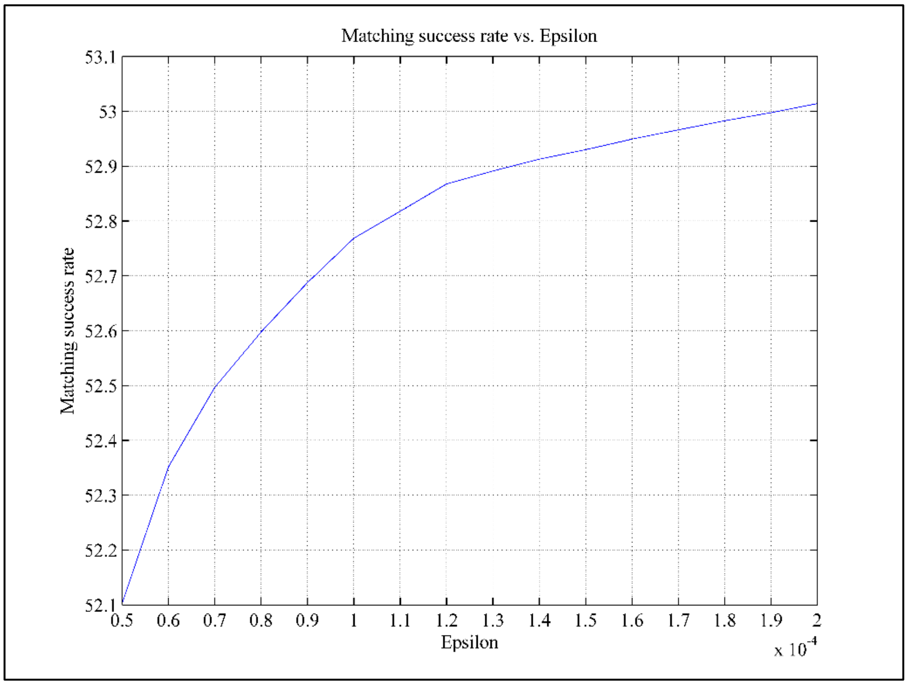


Figure 42: Matching rate for provided script

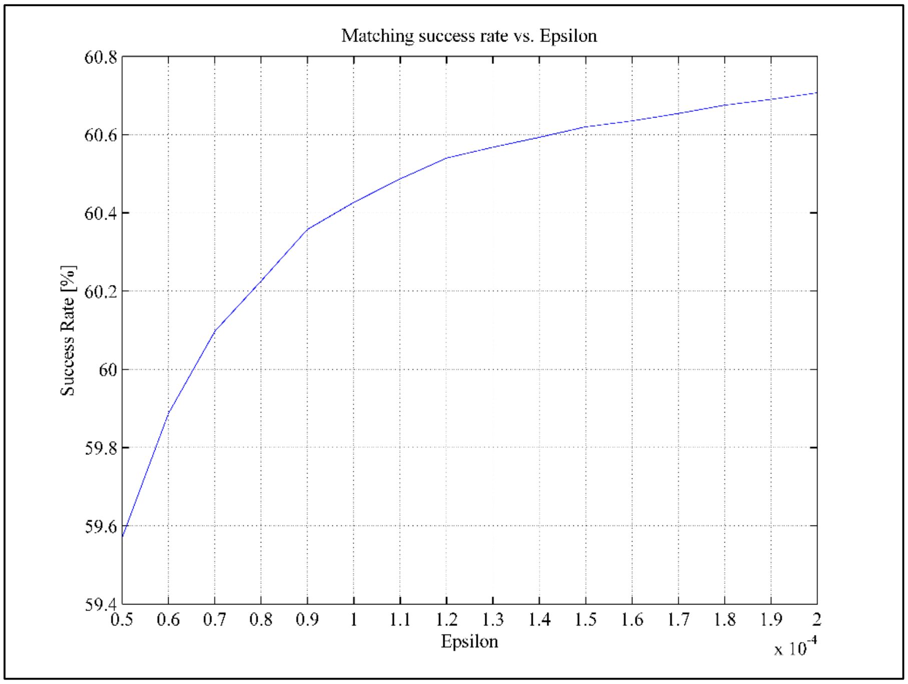


Figure 43: Matching rate for author's script

Both samples increase in successful matching as the epsilon is increased, with the difference being that the velocity script written by the author finds roughly 7 % more matches for all epsilons.

Table 7: Error and number of matches

Metric	Provided script	Author's script
Error in X [µm]	4.916	4.038
Error in X [%]	0.3545	0.2608
Error in Y [µm]	4.961	3.775
Error in Y [%]	0.3502	0.2485
Error in Z [µm]	8.369	7.840
Error in Z [%]	0.6187	0.5531
Error in U [mm]	1.075	1.073
Error in U [%]	8.9653	8.1657
Error in V [mm]	1.162	1.011
Error in V [%]	6.4261	5.0193
Error in W [mm]	1.407	2.207
Error in W [%]	15.0435	23.4375
Number of Particles	24,796	28,730

The author’s script obtains about 7 % more matches which translates to roughly 4,000 matches, and the errors for X, Y, Z, U, and V are lower. However, there is a 150 % increase in error in the W component error. Again, it is up to the analyst’s discretion which results have higher priority, and it must be kept in mind that these calculations are for statistical validation only, and have no bearing on how well the software calculates the desired metrics.

Plots of the vector field for a particular frame are displayed below. Figure 44 shows the matched results using the provided post processing script. Next, Figure 45 displays the results for all particles in the frame using the same script.

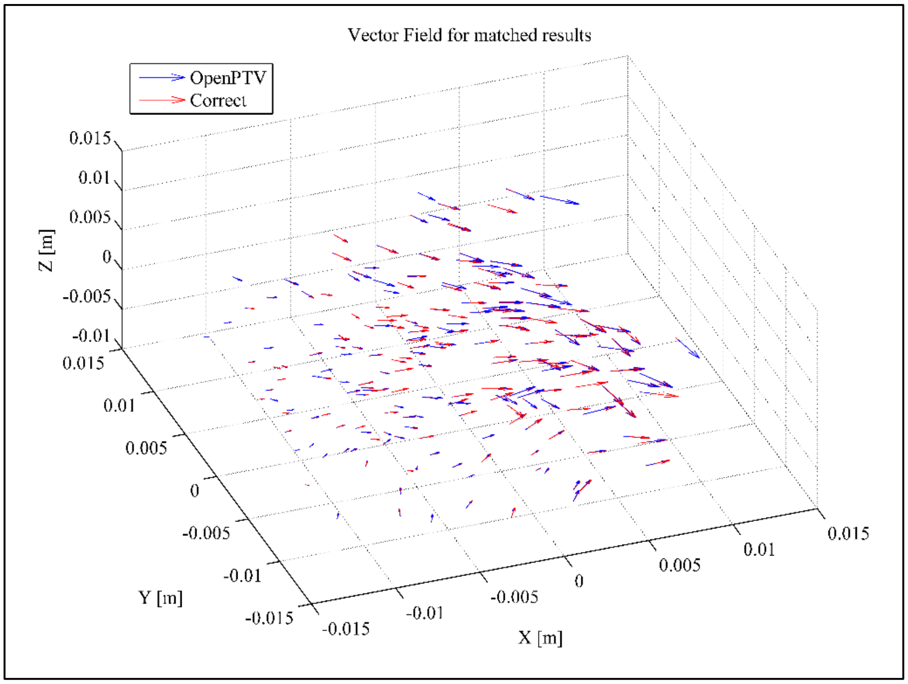


Figure 44: Matched vectors for provided script

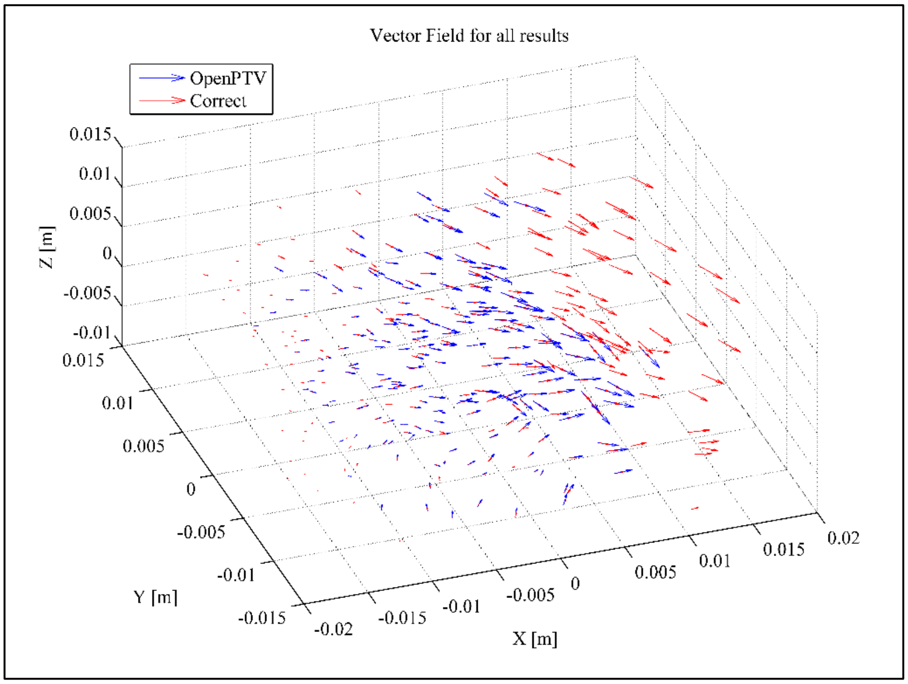


Figure 45: All vectors using provided script

One can easily notice that all vectors from X > 0.01 m were missed, but the matched vectors overlay precisely due to the relatively high accuracy. Figure 46 shows the matched vectors using the author’s script, and Figure 47 shows the full vector field for the frame. The result visibly looks slightly better, which suggests that the improvement of 7 % seems to be correct. From the modified vectors, there are a total of 47,345 particle positions and velocities. Losing about 40 % of the data is a significant loss in information, but it has to be remembered that particle tracking velocimetry is a measurement technique to analyze the overall flow in the target volume. Based on the results, it seems that despite the 40 % loss of data, the spatial resolution of the results is not significantly affected, and that the available data is sufficient for a quantitative statistical study of the simulated images.

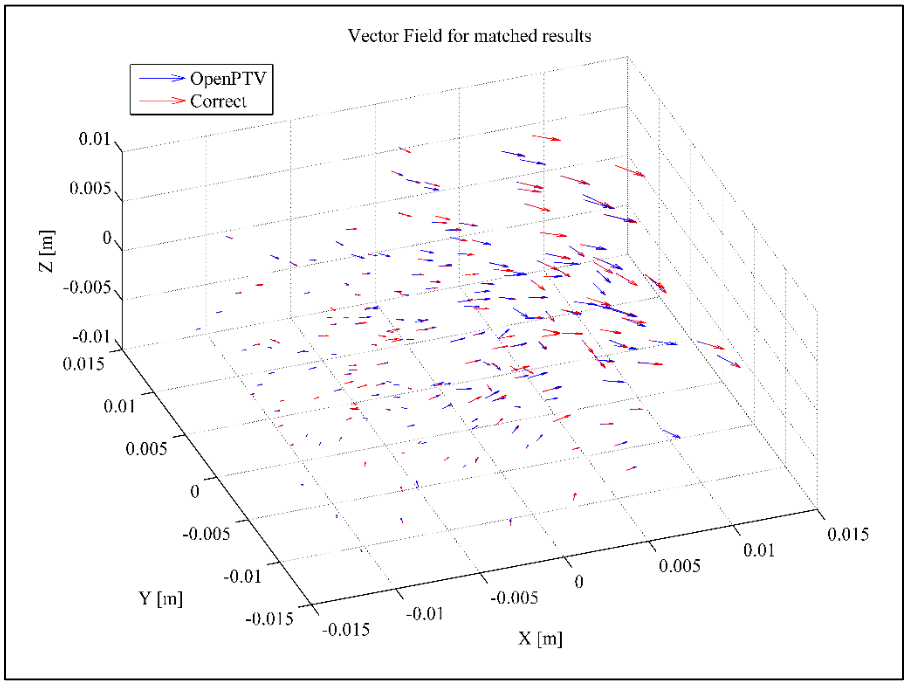


Figure 46: Matched vectors using author's script

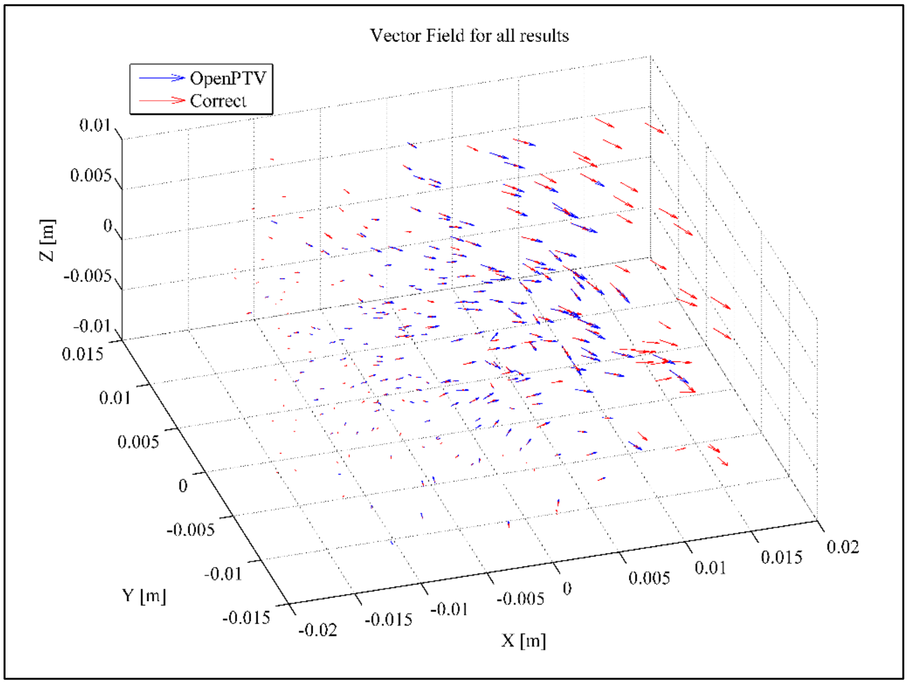


Figure 47: All vectors using author's script

5.1.5 **Statistical steady state analysis**

In PIV, the results are presented in a structured grid stretched throughout the fluid domain, as shown in Figure 19. In order to calculate important flow characteristics such as vorticity or Reynolds stress, the individual particle vectors have to be averaged and/or interpolated to systematic positions in cross-sectional planes, i.e. x - y, y - z, x - z. As currently the OpenPTV consortium is not studying flows according to these characteristics, the author wrote a script in MATLAB to average all vectors in one frame, multiple frames, or the complete image sequence in order to obtain a three dimensional grid where the vectors are assigned to specific locations, called voxels. The grid can be a cube, or stretched in one or all of the three coordinate directions, similar to a mesh in CFD simulations. The first grid size was chosen as 10 by 10 by 10 cubes inside the target volume. Figure 48 shows the averaged vector field with the provided script velocity, and Figure 49 shows the same for the author's obtained velocity.

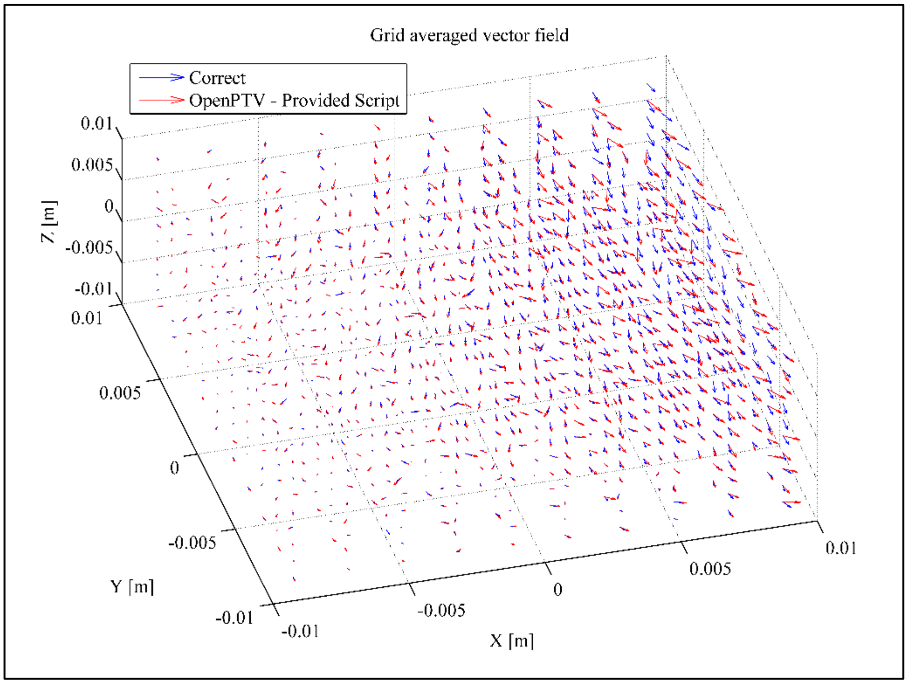


Figure 48: Provided script result

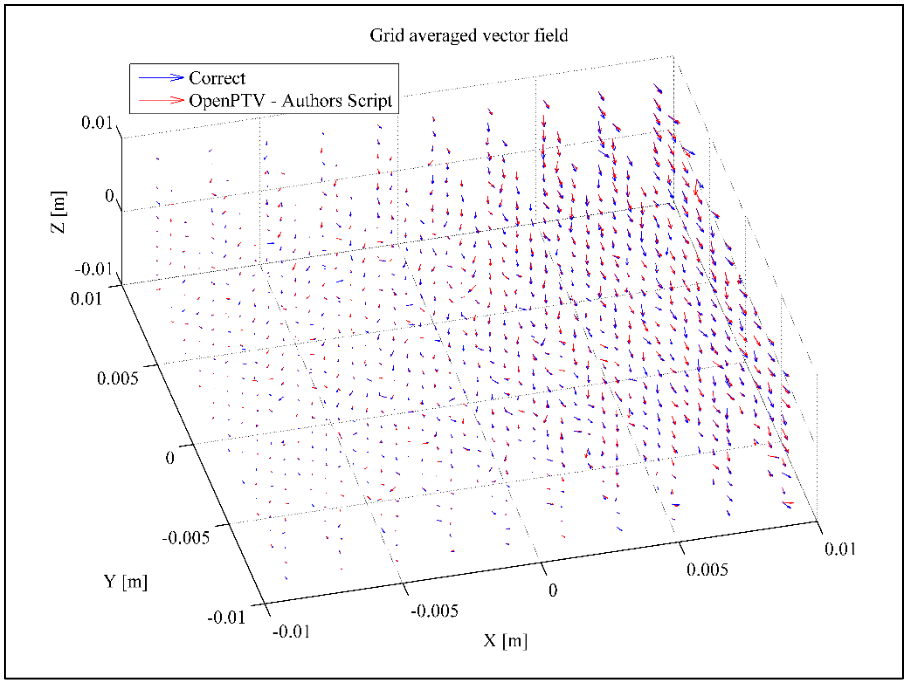
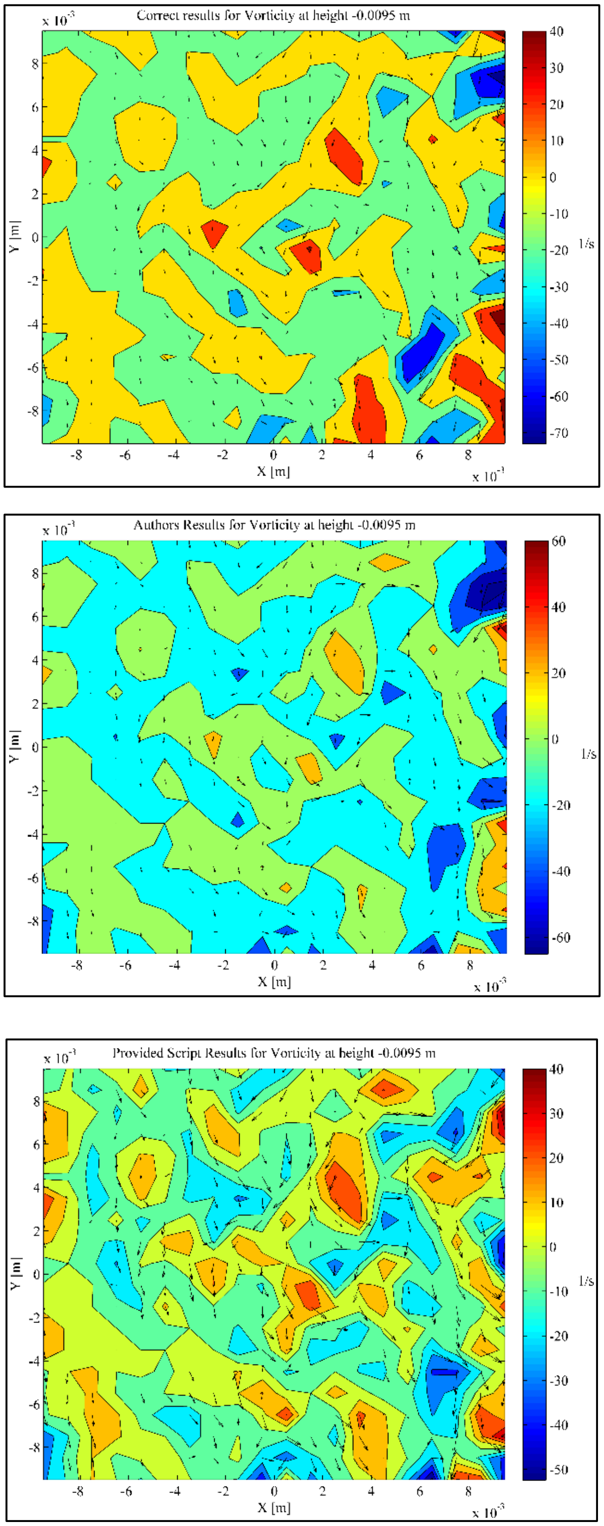


Figure 49: Author's solution

From a closer look, one can see that there are a lot of incorrect vectors in the velocity obtained by the provided script (PS). This trend is seen throughout the statistical steady state analysis. The author continued the analysis of the OpenPTV results by studying the vorticity behavior. In nuclear and mechanical systems, understanding the cause and strength of the vorticity is crucial. The author decided to use a finer grid in order to create smooth plots. A 20 by 20 by 20 grid size was chosen and the vorticity was plotted across the $x-y$ plane at three different heights. Figure 50 shows the vorticity at the bottom plane (z = -0.0095 m) using the correct, author's, and PS results. Figure 51 shows the vorticity close the center (z = -0.0005 m), and Figure 52 shows the vorticity at the top plane (z = 0.0095 m).

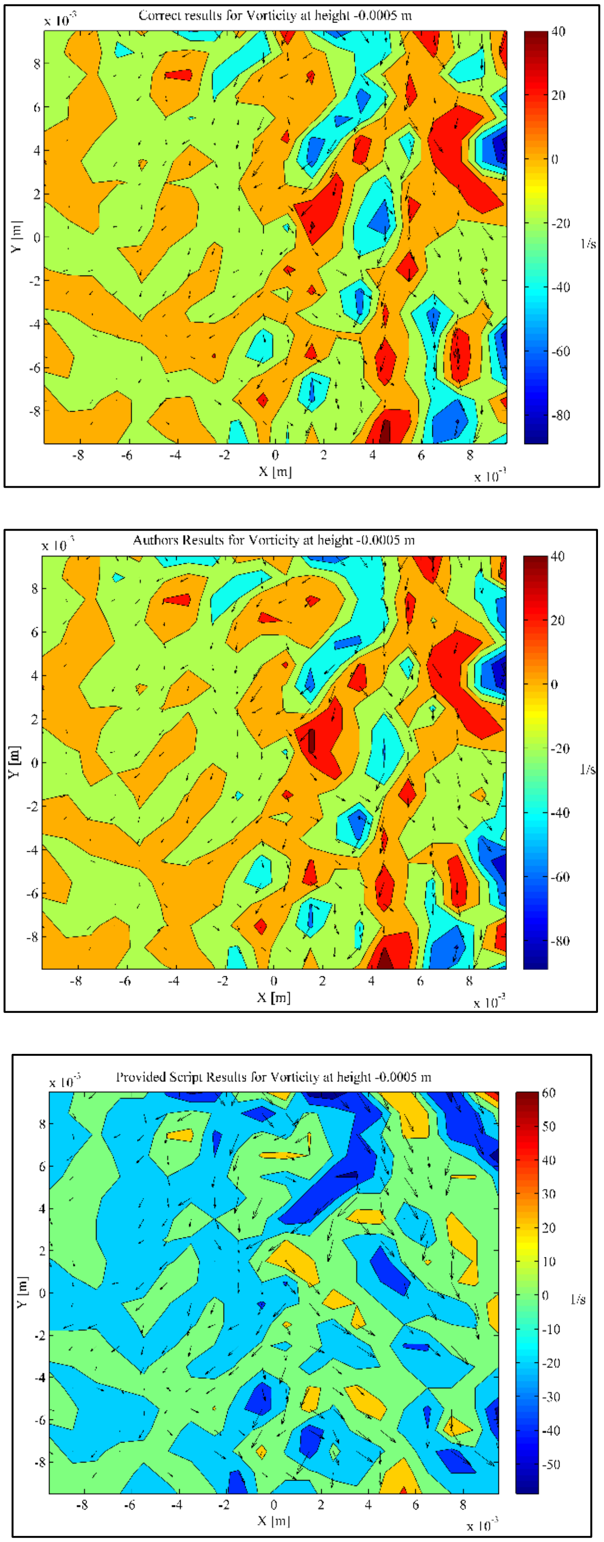


Correct results for Vorticity at height -0.0095 m

Authors Results for Vorticity at height -0.0095 m

Provided Script Results for Vorticity at height -0.0095 m

Figure 50: Vorticity at bottom plane


Correct results for Vorticity at height -0.0005 m

Authors Results for Vorticity at height -0.0005 m

Provided Script Results for Vorticity at height -0.0005 m

Figure 51: Vorticity at center plane

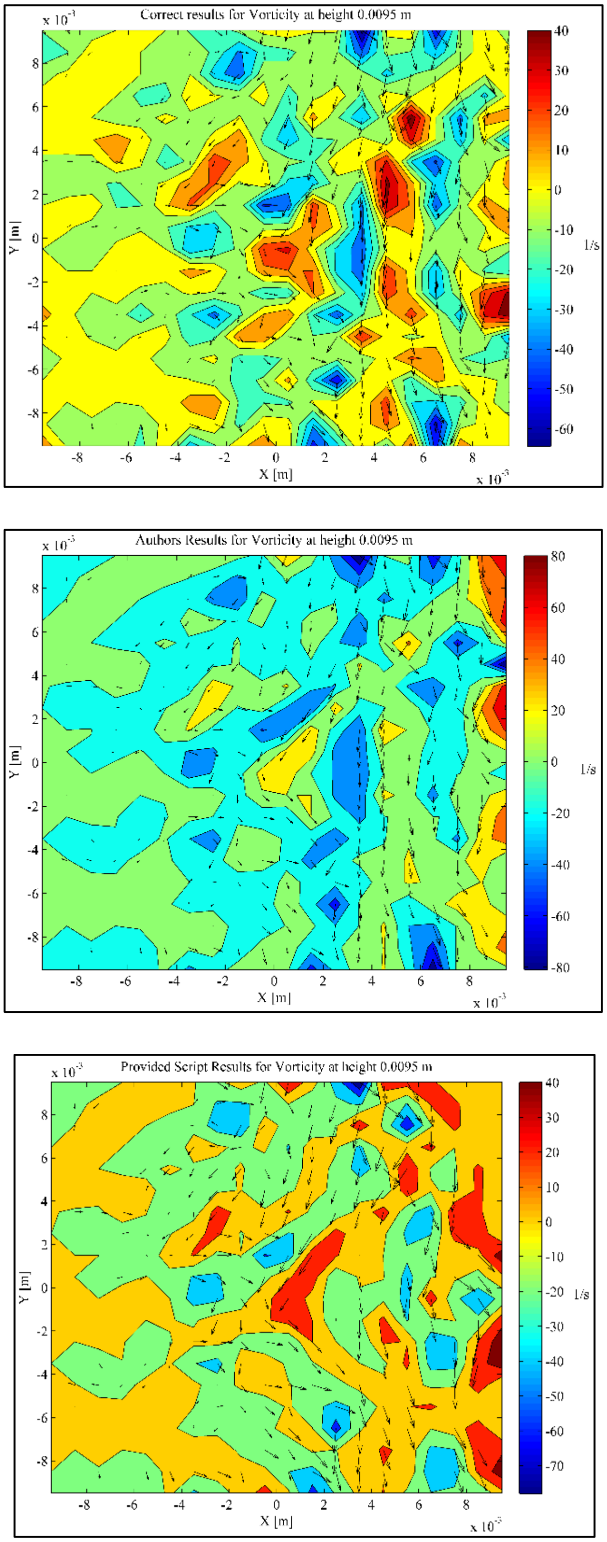


Figure 52: Vorticity at top plane

It seems that in the bottom and center planes, the author's result more closely matches the correct solution, and at the top plane the PS results match better. However, the overall behavior from the author's results still matches that of the correct solution. The vectors in the particular plane are overlaid to gain a better understanding of the flow velocity. As a last piece of study of the synthetic images, the author plotted the streamwise velocity at different heights and at different cross sections in the streamwise direction. The heights are identical as the vorticity plots; Figure 53 shows the bottom plane, Figure 54 the center plane, and Figure 55 shows the top plane. The results are overlaid so any major deviations can be observed easily.

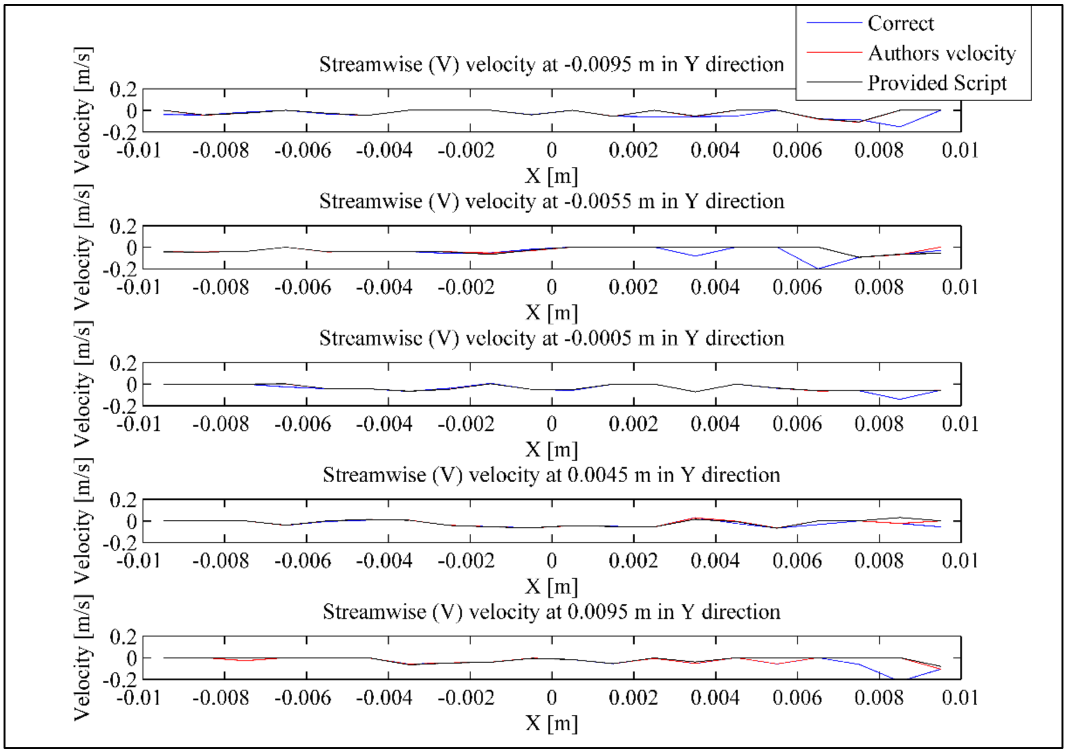


Figure 53: Streamwise velocity in *x-y* plane at bottom height

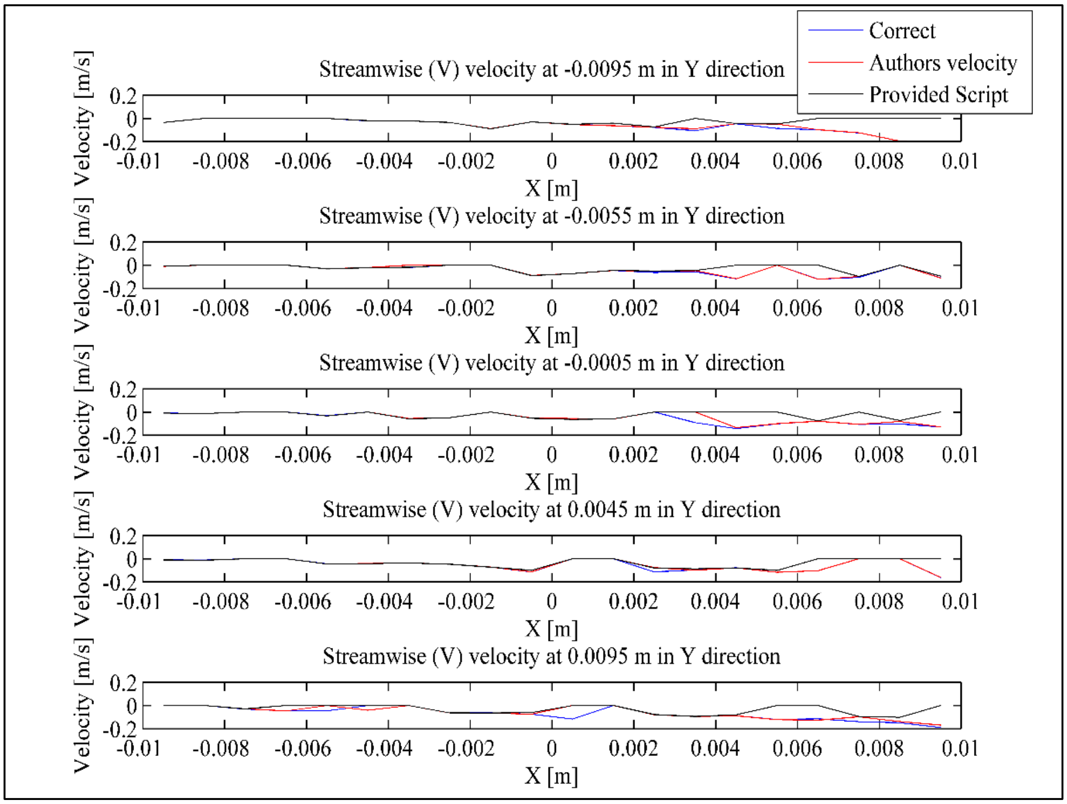


Figure 54: Streamwise velocity in *x-y* plane at center height

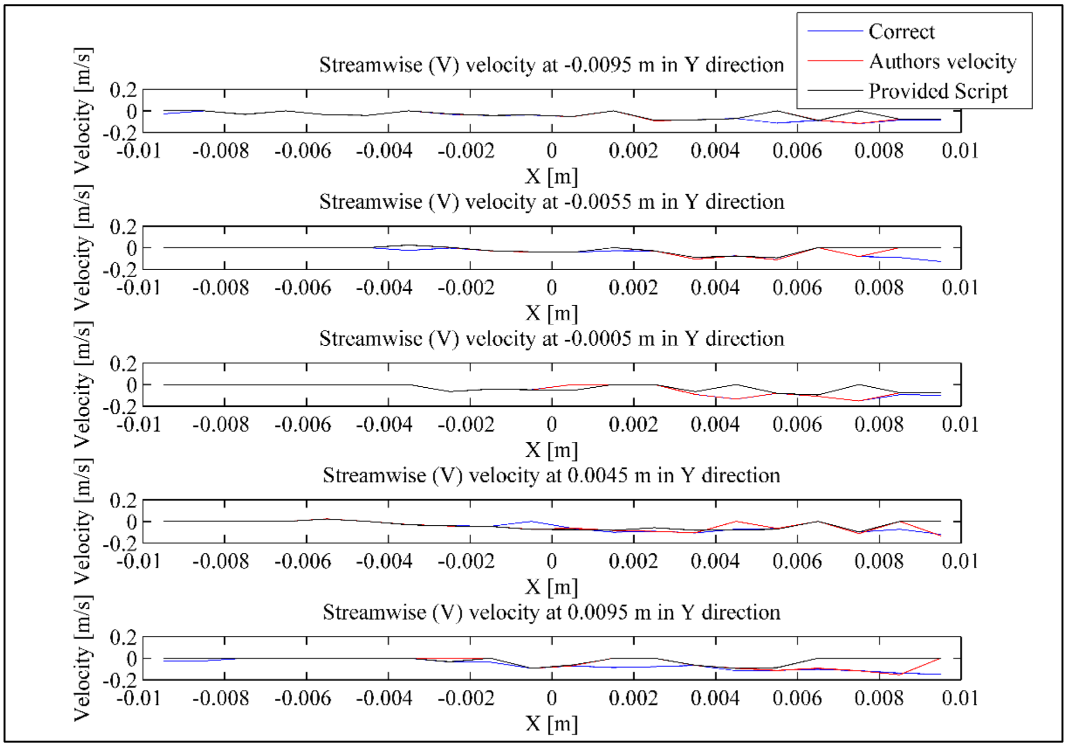


Figure 55: Streamwise velocity in *x*-*y* plane at top height

Again, the author’s solution more closely matches the correct data than the PS solution. Since the simulated target area is at the left edge of the jet (from the top, or $x - y$ plane), the streamwise velocity is expected to be small to zero going into the negative X direction, away from the jet. By the same logic, going closer to the jet (positive X) means that the streamwise velocity is increasing. One instance of the Reynolds stress tensor component $\overline{u'v'}$ was plotted in Figure 56 at the same cross sectional positions as the streamwise velocity. The plot shows that there are strong fluctuations going away from the jet stream until they level off at zero, as expected.

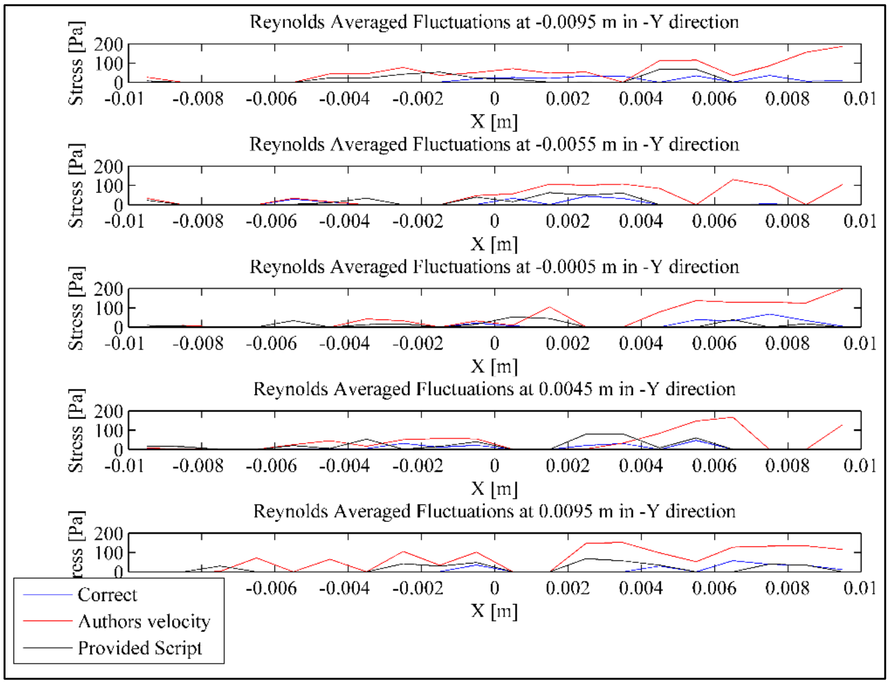


Figure 56: u'v' Reynolds fluctuations at various cross sections

The errors after averaging were computer to determine whether the averaging has an effect. Using grid sizes of 5x5x5, 10x10x10 and 20x20x20 cells the resulting errors for U, V, and W are displayed in Table 8: Errors after averaging.

Table 8: Errors after averaging

Metric	Provided script			Author's script		
	5	10	20	5	10	20
Error in U [mm]	7.504	9.521	10.074	1.929	3.732	4.644
Error in V [mm]	10.948	14.529	21.988	1.973	3.462	6.856
Error in W [mm]	3.245	5.324	6.415	1.422	3.098	3.717

The errors in U and V are larger than compared with the results in **Table 7**, but surprisingly the error in W is lower. It was expected that the averaging changes the error, but the author expected an increase in all velocity component errors, and not a decrease in the out-of-plane vector which is expected to have the largest error. Note that as the grid size increases, the error increases which is expected due to the large number of particles not detected by the OpenPTV software and consequent lack of vectors in a lot of cells.

5.1.6 **Computational Fluid Dynamics (CFD) study**

As a bonus study the author tried to confirm the quantitative results with a computational fluid dynamics simulation. The geometry was rebuilt from the available information in [13], and five different mesh sizes were used in the simulation. Figure 57 shows a plot of the mesh as well as the number of faces and cells. The total length is 1.06

m, the width 0.2 m, and the height 0.078 m. The inlet pipe was created as a 2 cm by 2 cm by 10 cm surface in order to ensure fully developed flow going into the large chamber.

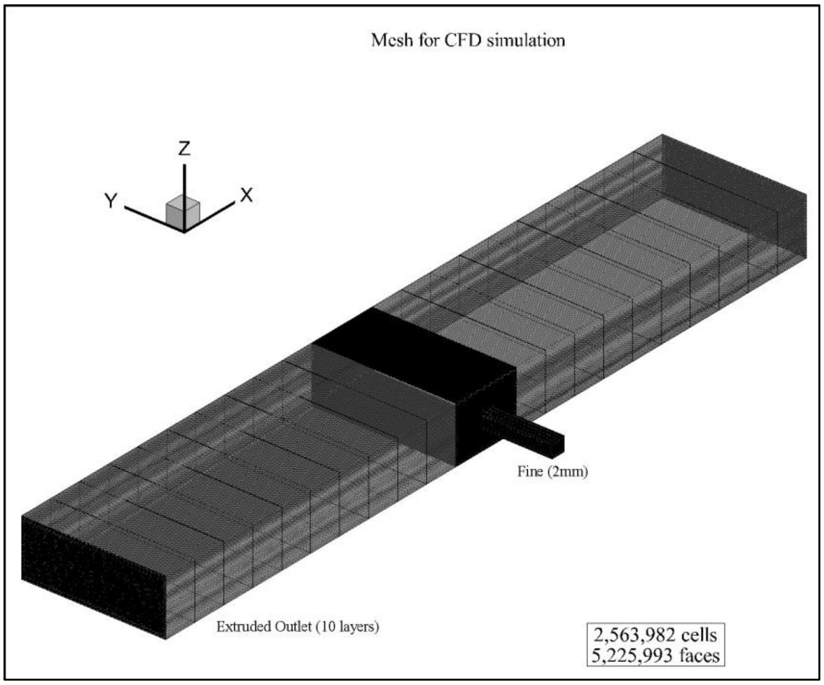


Figure 57: Mesh used for CFD

The five different mesh sizes had a different size mesh, namely 2, 3, 4, 5, and 6 mm. The author used more than one mesh in order to determine if the solution is grid independent. After each simulation was done, the author used the maximum velocity as a reference and computed the percent difference between two consecutive mesh sizes. Table 9 shows the mesh size, the maximum velocity, and the percent difference.

Table 9: Mesh size and percent difference in velocity

Mesh Size	Maximum Velocity [m/s]	Difference [%]
2 mm	0.17858	-
3 mm	0.17117	4.15
4 mm	0.16928	1.10
5 mm	0.16588	2.01
6 mm	0.16468	0.72

Since the differences are so small, it seems that either one of the solutions can be used as the grid independent solution. The author chose the finest mesh of 2 mm. The original standard images for the various test cases were done using transient LES simulations, however no time information was provided. Therefore, the author chose to run a steady state turbulent simulation with a k-epsilon turbulence model assuming constant density and three dimensional flow. The resulting streamline velocity at the center plane is shown in Figure 58 from a side view, and from the top in Figure 59.

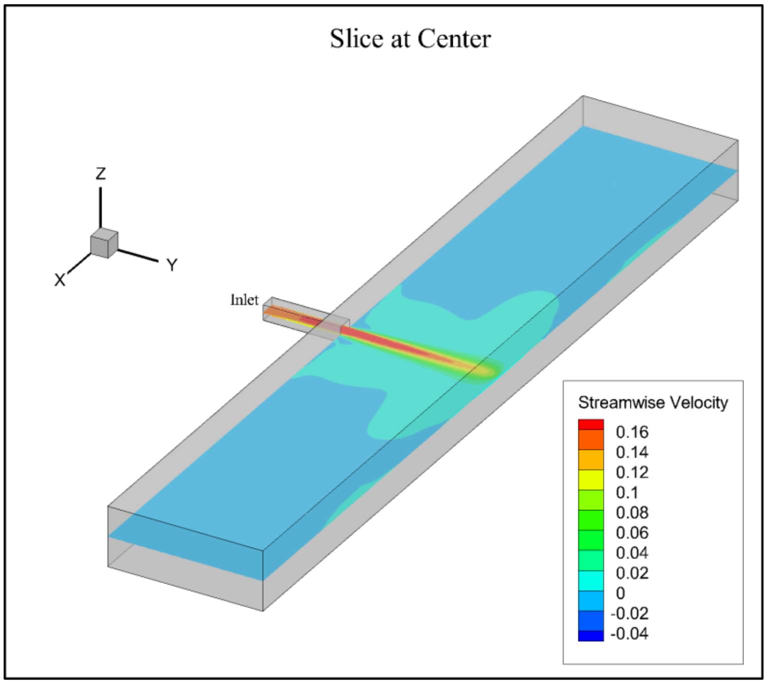


Figure 58: Side view of slice showing streamwise velocity

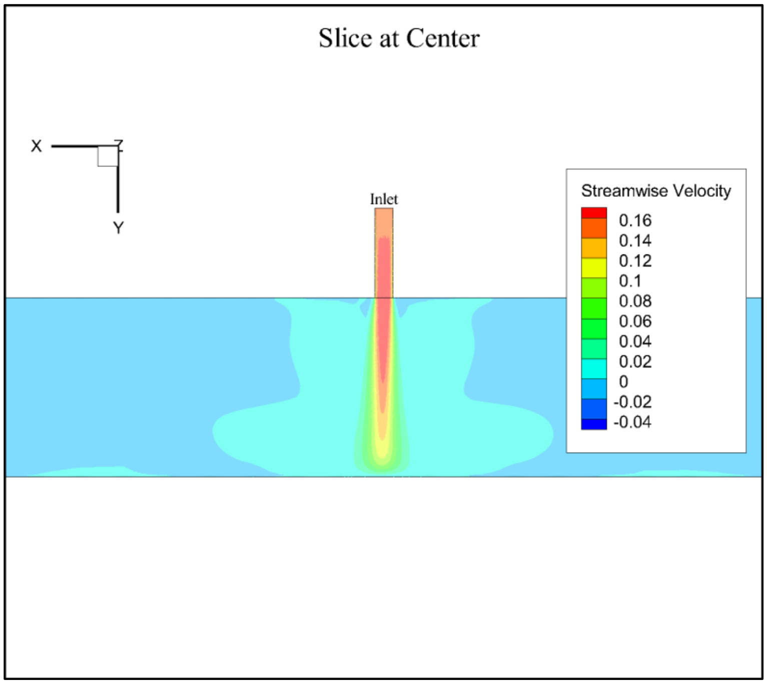


Figure 59: Top view of streamwise velocity profile

A contour plot of the streamwise velocity from the correct data and from the author’s solution is presented in Figure 60. Note that this contour plot represents only a small fraction of the information in Figure 59, as the target area is only 2 cm by 2 cm of the complete 1.06 m by 0.2 m geometry.

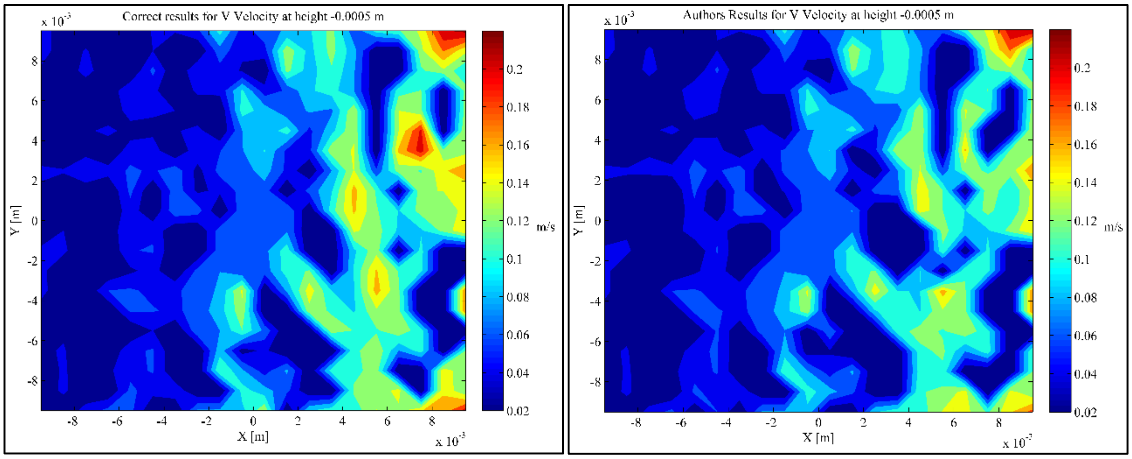


Figure 60: Streamwise velocity contour for correct data and author's solution

The magnitudes from the two figures match reasonably well, and the author’s script of the OpenPTV velocity reconstruction is very accurate in comparison to the correct data. Figure 61 shows a guess of where the target area might have been with the correct scale of the target area to the complete system geometry.

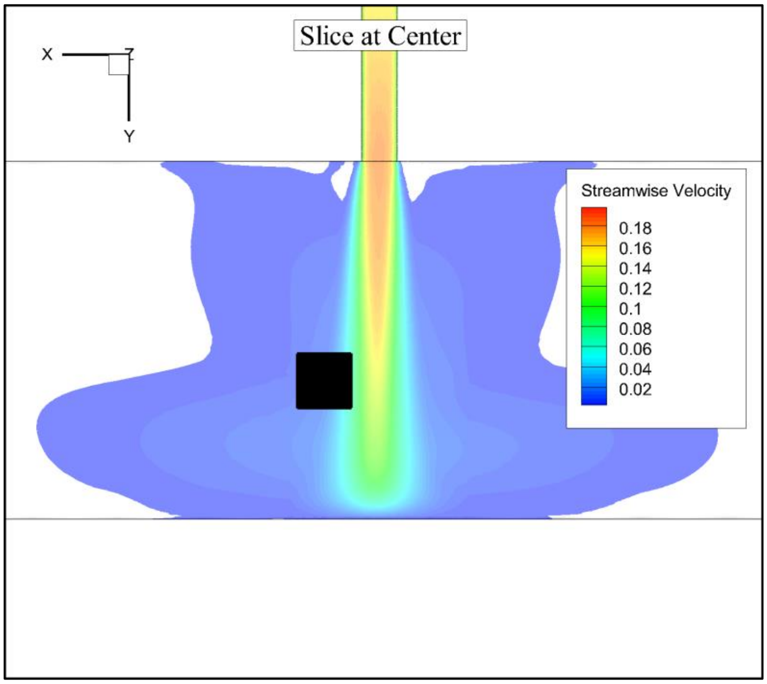


Figure 61: Velocity profile and target area from CFD

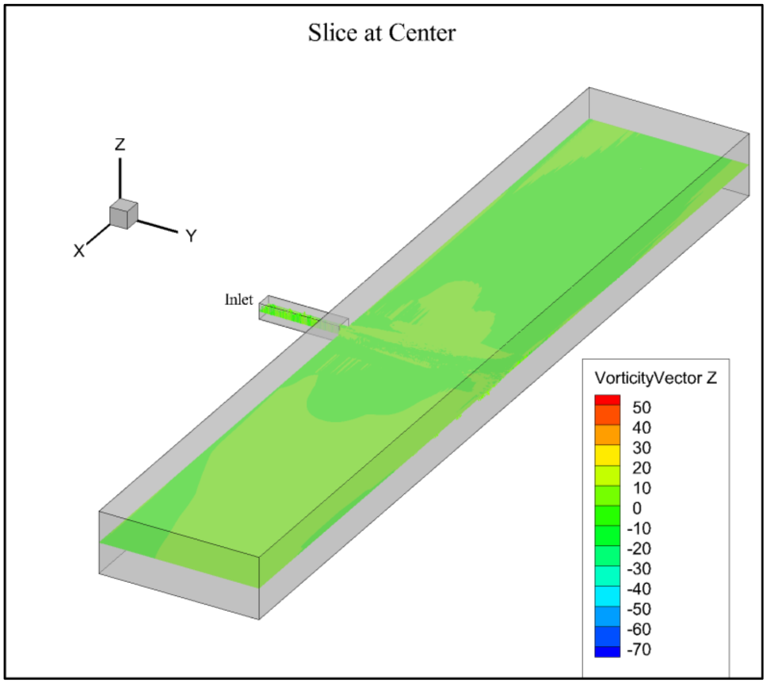


Figure 62: Vorticity slice at center from side

Since the vorticity was calculated previously and shown in Figure 50 through Figure 52, the author wanted to compare the results with the CFD results. Figure 62 shows the vorticity from a side view for a slice at the center plane, and Figure 63 shows the vorticity from above the system.

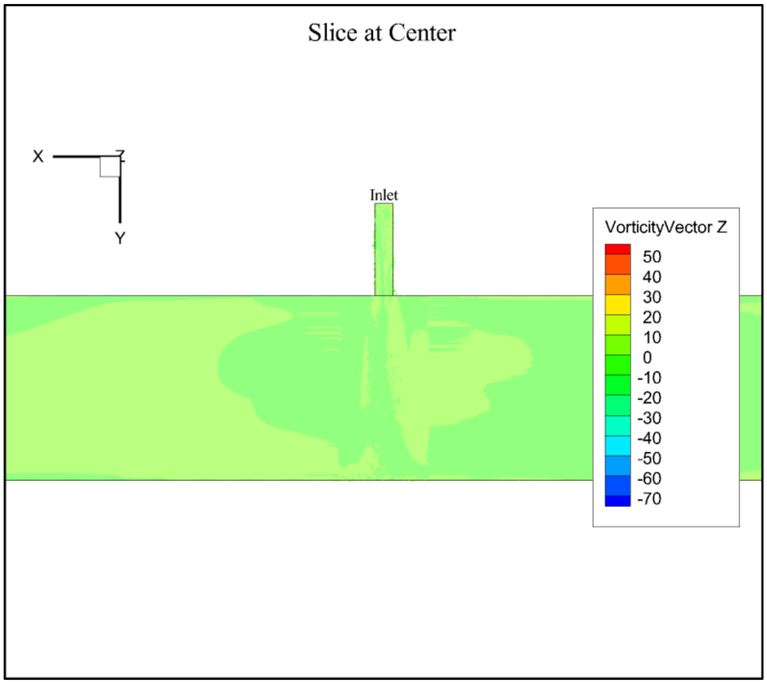


Figure 63: Top view of vorticity

The author is unsure why the automatically set scale is at is it considering that the contrast of color in the figure is not noticeable, however the scale basically matches that of Figure 51 which confirms that the images are from an experiment with identical/similar conditions and that the OpenPTV software correctly computed the velocity. The guessed target area is also plotted over the vorticity slice in Figure 64.

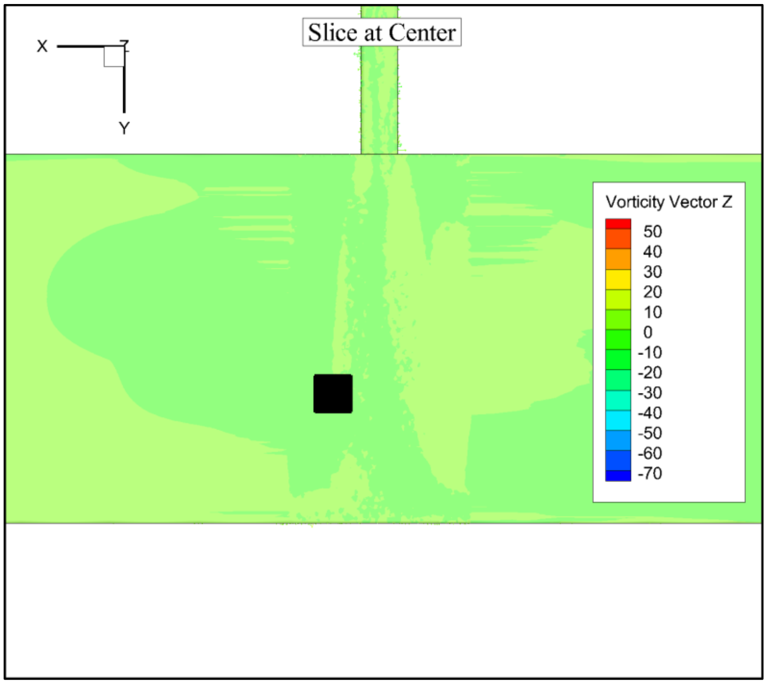


Figure 64: Target area in vorticity plot

While the author was not able to determine the correct position of the target area, the results of the CFD simulation are very close to those obtained from OpenPTV and compared with the correct data, thus further validating the performance of the OpenPTV software.

5.1.7 **Trajectories**

The OpenPTV code provides the Lagrangian trajectories of individual particles. For the assessment of the code there is not much information provided to verify the code's ability to successfully reconstruct the trajectories; therefore the author did not focus much attention to this aspect of OpenPTV. An attempt to obtain an idea of the performance of the code is shown in Figure 65 and Figure 66 which show the trajectories with a minimum length of 50 from the OpenPTV output, and from the correct data (generated by the author), respectively. The OpenPTV code found 91 trajectories with a minimum length of 50 points, while the correct data has 779 trajectories of this minimum length, yielding a success rate of only approximately 11.7 %. In the correct data, a total number of 988 unique particles were found which yields 988 unique trajectories. With the author's parameters, the OpenPTV code finds 1733 trajectories. It is thus clear that while the code is able to reconstruct and connect some of the "broken" trajectories, there are a lot of "additional" trajectories which are found due to the inability of the code to find all the links.

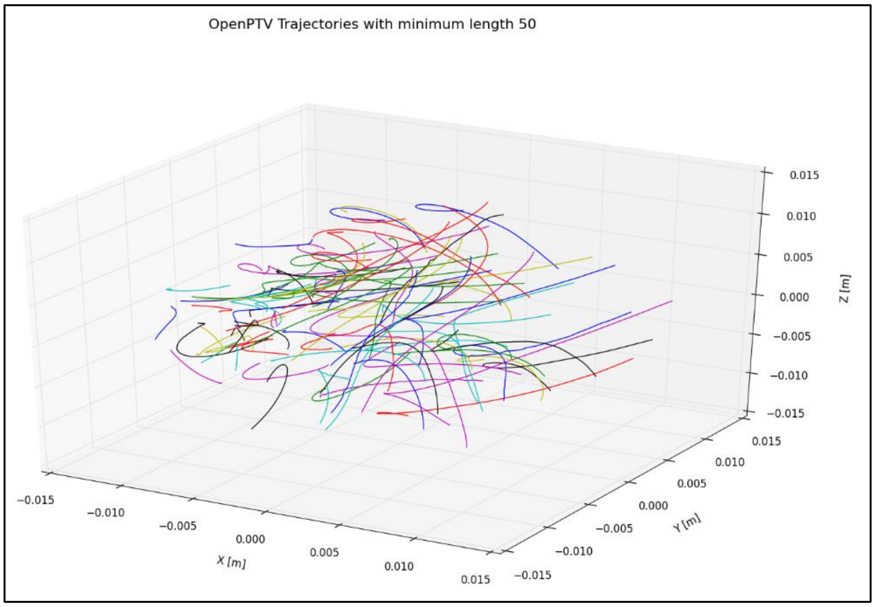


Figure 65: OpenPTV output for trajectories with length over 50

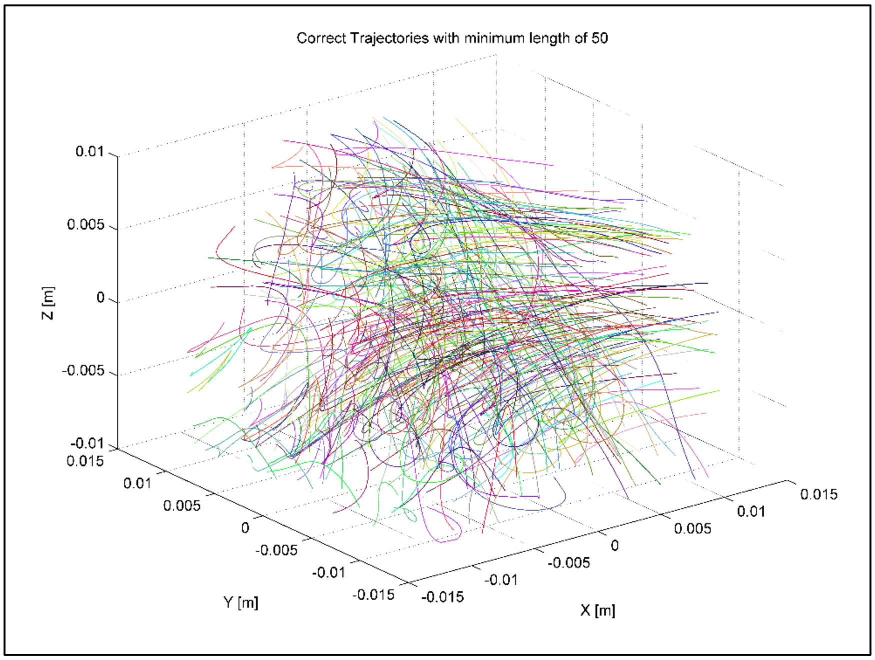


Figure 66: Correct data for trajectories with length over 50

Figure 67 and Figure 68 show the complete trajectories for all frames from OpenPTV, and the correct data, respectively.

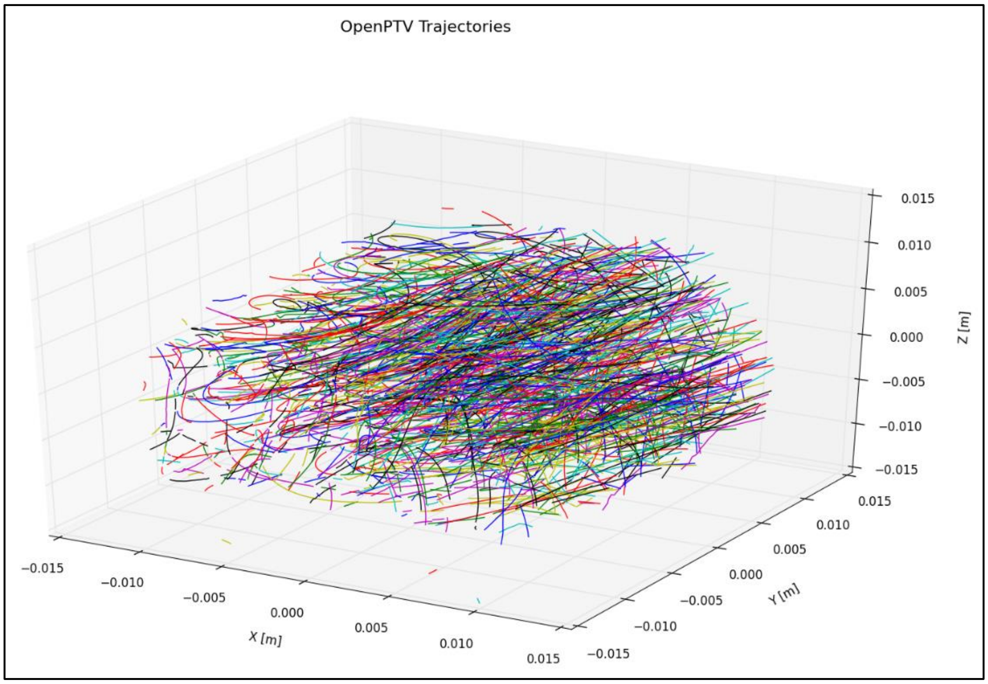


Figure 67: OpenPTV - all trajectories

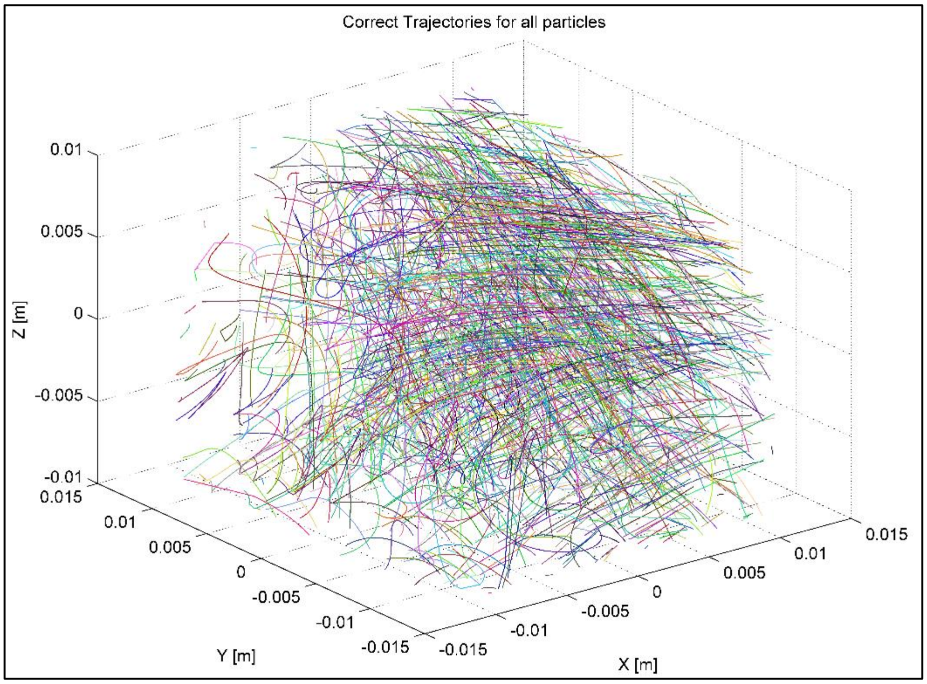


Figure 68: All correct trajectories

The author is convinced that under different conditions, i.e. laminar steady state flow, the OpenPTV software will more successfully reconstruct the complete particle trajectories. Tuning the parameters during the image processing and tracking procedure can improve the number of reconstructed trajectories. An example of a very successful trajectory reconstruction can be found in Dr. Liberzon's study of the aorta mentioned section 1.5 [14].

5.2 *Twin-jet experiment*

As mentioned in section 4, the twin jet geometry was studied using three different approaches, namely Laser Doppler Velocimetry, 2D-PTV as well as with the OpenPTV software.

5.2.1 **2D analysis**

Before the three dimensional performance of the OpenPTV code is tested, the author gathered data using only one camera looking straight at the jets (center), and from the side of the jets (rotated). The images were analyzed using the OpenPTV code as well as the lab's 2D-PTV code. Additionally, a Laser Doppler Velocimetry study was done for the center configuration with measurements taken at multiple locations.

5.2.1.1 <u>Laser Doppler Velocimetry</u>

The traverse system (partly seen in Figure 22) which holds and moves the cameras was moved to obtain data through a large part of the length and width of the tank. For the present study, only data from 48 mm to the left and to the right and 50 mm below and above the center of the target area is used.

Measurements within this specified area were taken at 45 mm below the center and at 50 mm above the center of the target area. 21 points ranging from -48 to 48 mm in the X direction were taken, and an additional 6 measurements were taken at 50 mm, which are called Main points in Figure 69 and correspond to points at the left and right edges of the jets as well as in their centers. The left jet outlet begins at -24.5 mm and end at -10 mm, while the right jet outlet begins at 12 mm and ends at 26 mm. The streamwise velocity is plotted and follows the expected parabolic behavior at the lower measurement, and then levels out between the two vertices which indicates mixing of the jet streams. The velocity was non-dimensionalized by the maximum velocity obtained, which for these measurements was 1.42 $^{m}/_{s}$.

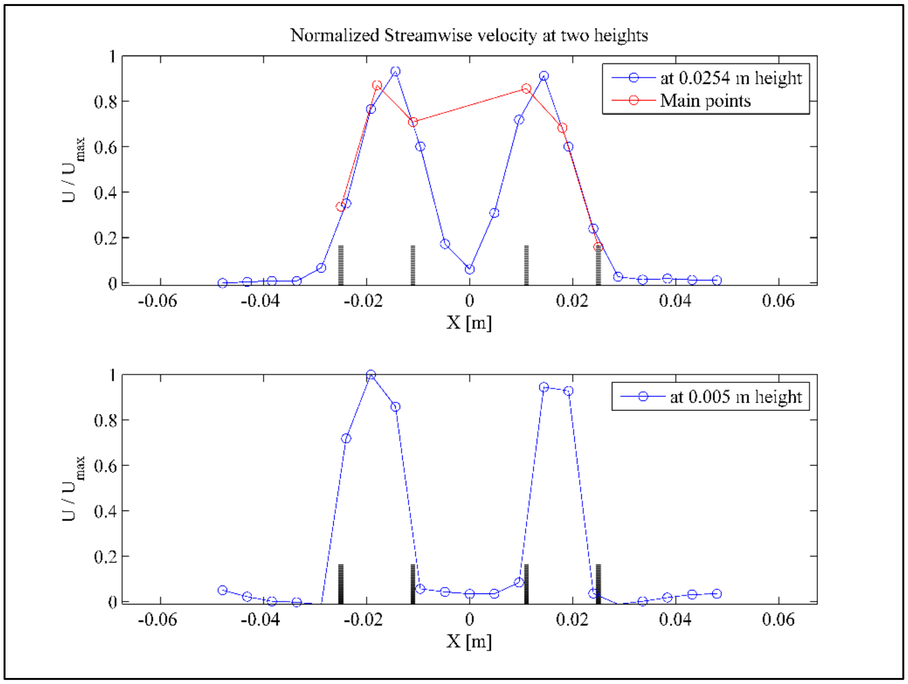


Figure 69: LDV streamwise velocity

The streamwise velocity profile is parabolic for each stream at the lower measurement location as expected. Note that no measurement was made in between the jets for the Main points readings, and MATLAB linearly interpolates the values in between the two recorded values.

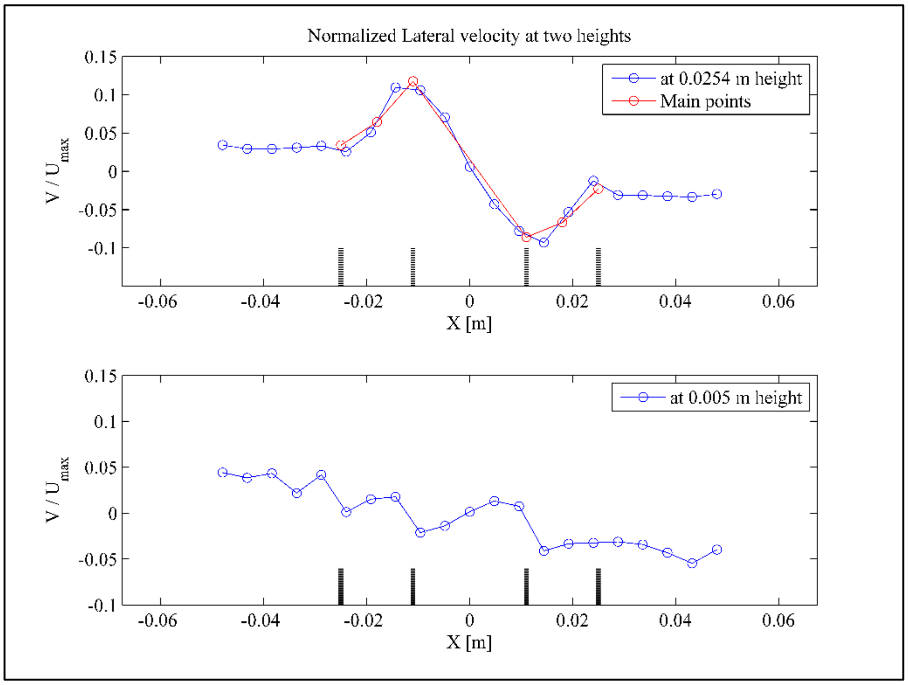


Figure 70: LDV lateral velocity

Figure 70 shows the corresponding non-dimensionalised lateral velocity. In the region between the two jets the profile is not yet developed for the lower reading, but clearly shows the expected behavior at the higher measurement. The streamwise and lateral fluctuations are also plotted for comparison in Figure 71 and Figure 72 respectively.

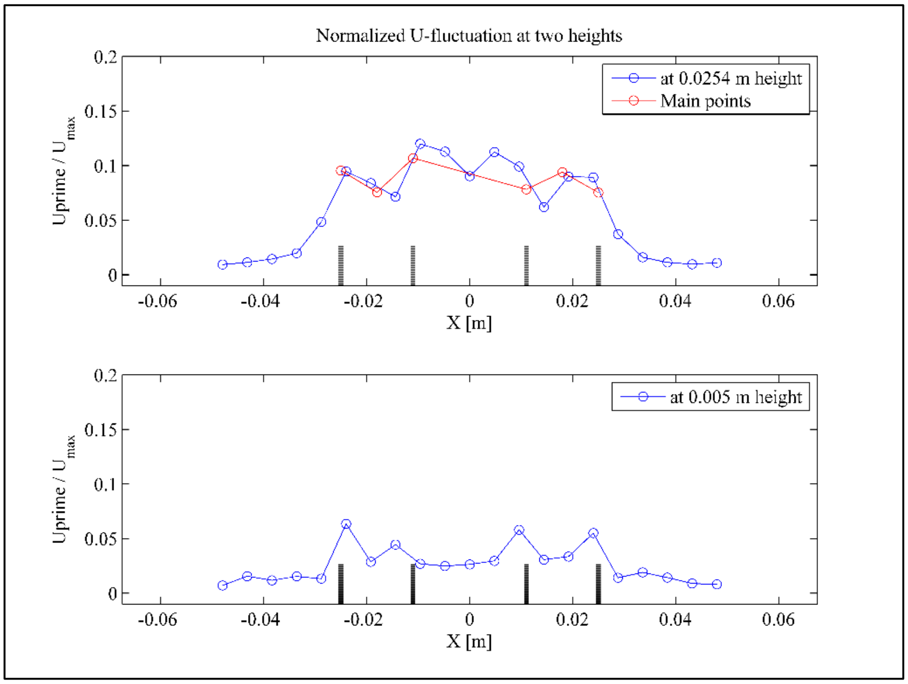


Figure 71: Streamwise fluctuation

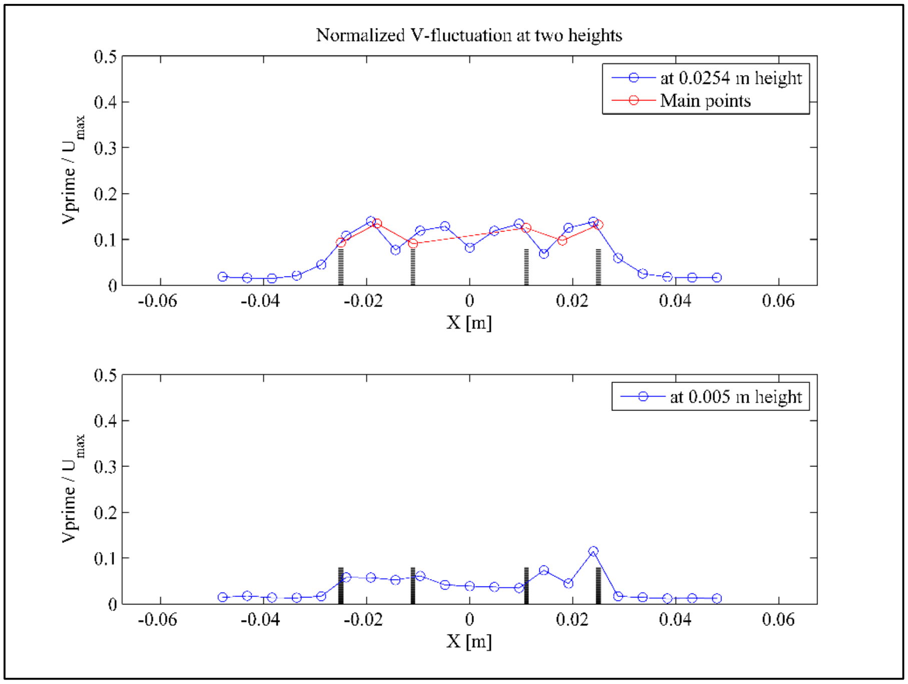


Figure 72: Lateral fluctuation

5.2.1.2 2D - Particle Tracking Velocimetry

The same images which were used in the OpenPTV study, were also processed using the labs two dimensional code. For comparison purposes, only three sets of data were used, namely the center view looking at the jets, and the rotated view once when the laser was directed at the back jet (further away from camera), and once when it was illuminating the front jet (closer to camera).

1. Center View

Figures Figure 73 and Figure 74 show the streamwise and lateral velocity contour plots. In Figure 73 one can also observe the cross sectional velocity at the two cross sections studied with LDV, and at a higher location.

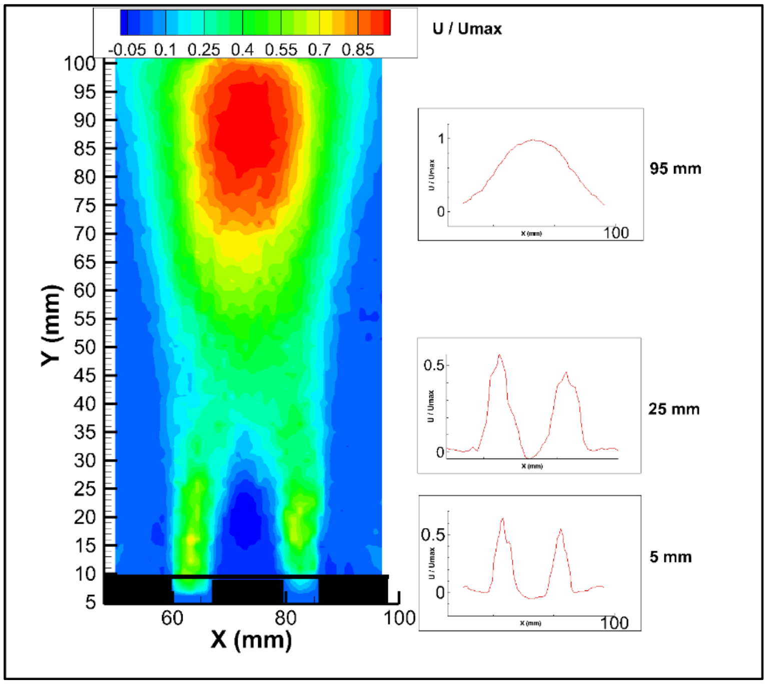


Figure 73: Streamwise velocity center

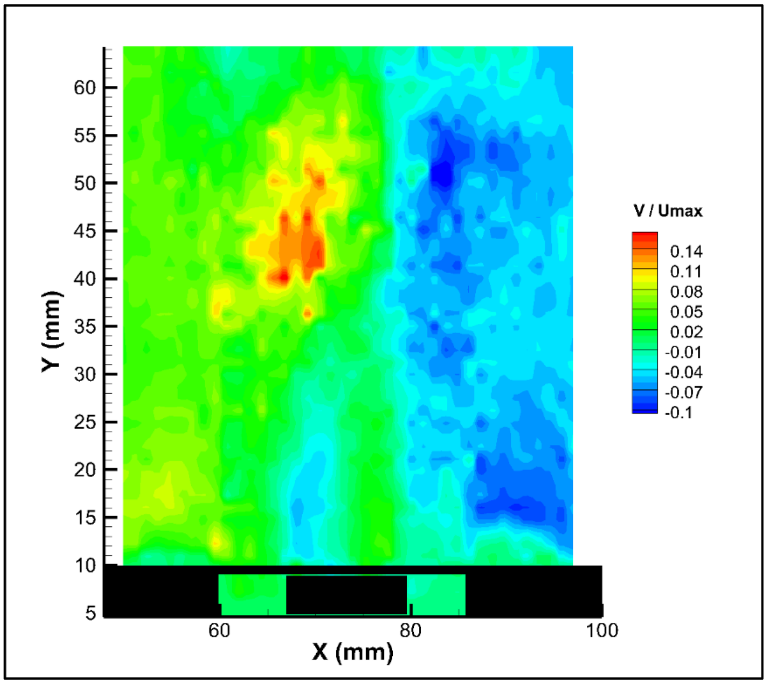


Figure 74: Lateral velocity center

One can clearly see the two jet streams separately right at the outlet, and then the mixing phenomena about 50 mm above the outlet. Moreover, the latter plot shows that the particles to the left of the stream are drawn towards it, and the same on the right where the velocity direction is reversed.

2. Back jet

Figures Figure 75 and Figure 76 display the velocity behaviour at the back jet. The streamwise velocity was found to be higher than the result obtained from the center view, but still below the LDV measurement.

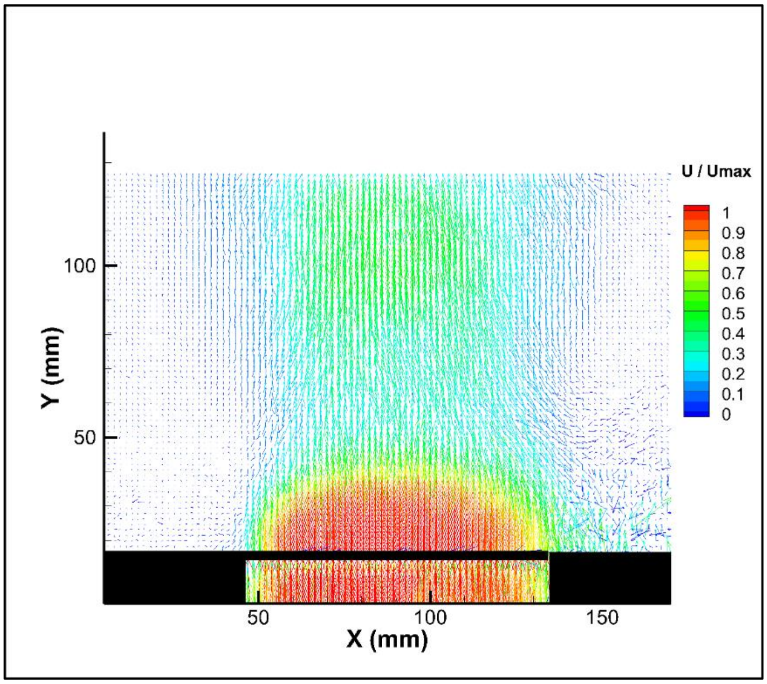


Figure 75: Streamwise velocity back jet

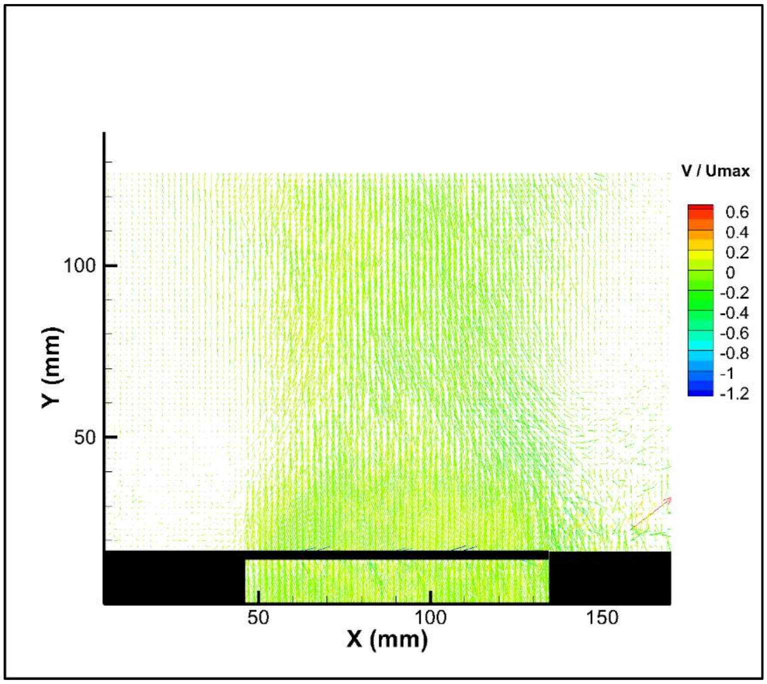


Figure 76: Lateral velocity back jet

3. Front jet

Figures Figure 77 and Figure 78 show the corresponding plots for the jet closer to the camera. The obtained results will be used to assess the performance of the three dimensional velocity reconstruction.

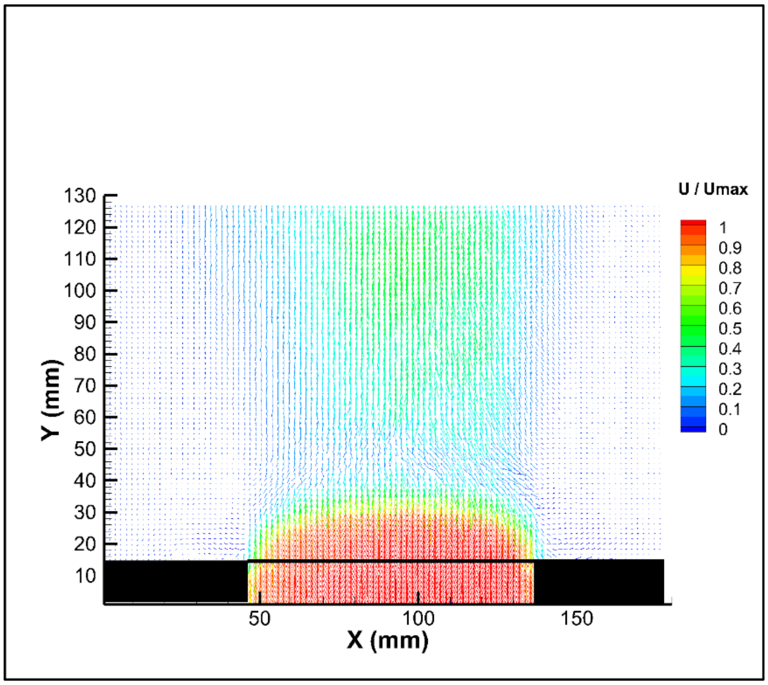


Figure 77: Streamwise velocity front jet

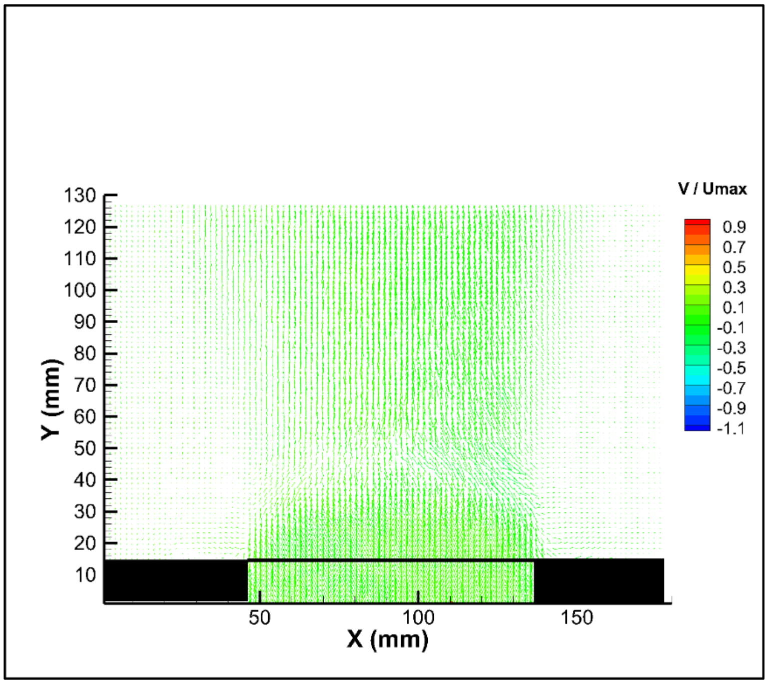


Figure 78: Lateral velocity front jet

5.2.1.3 OpenPTV

1. Pre-processing and calibration

After the images were converted to the required .tiff format and renamed, the author performed a calibration of the cameras, which although the code is used for 2D, is still necessary as the code needs to know where the origin of the target area is, and how much a displacement of 1 pixel translates to a displacement in world coordinates. The calibration was performed for the same three sets of data as used in the 2D-PTV code, and matched the expected orientation information, including the distance between the two jets from the rotated view.

2. Tracking procedure

The numerous parameters in OpenPTV allow the user to vary the resulting output data very significantly. In initial runs the author kept the number of detected particles similar to the standard PIV test case study. However, as the 2D-PTV code attempts to use all particles in each frame, the OpenPTV parameters were set to detect as many particles as possible. This was done by lowering the grayscale threshold and reducing the expected size of each particle. Figure 79 shows the detected particles in one frame for the front jet, and Figure 80 shows all detected particles in all frames overlaid in one image.

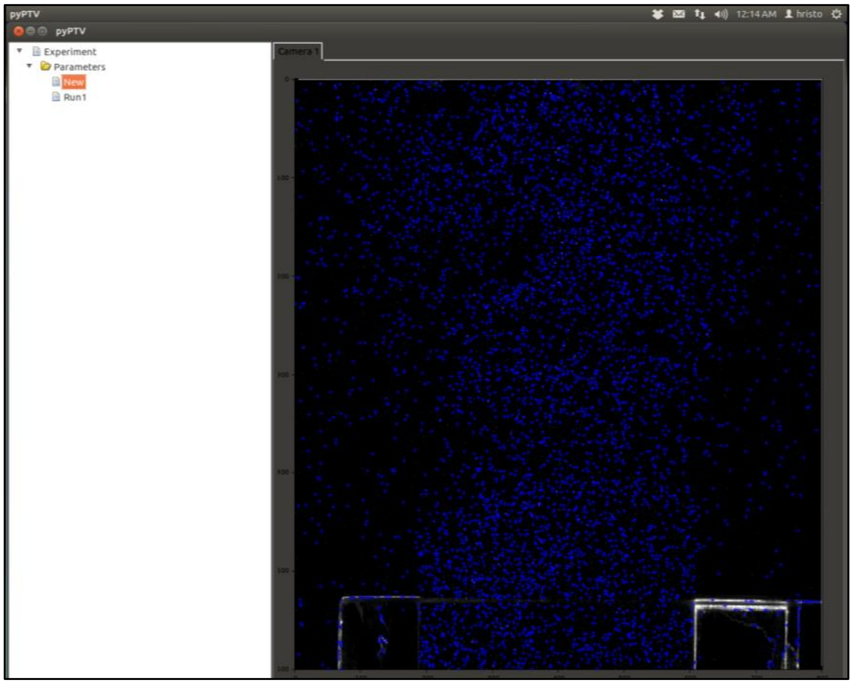


Figure 79: Detected particles in one frame for front jet study

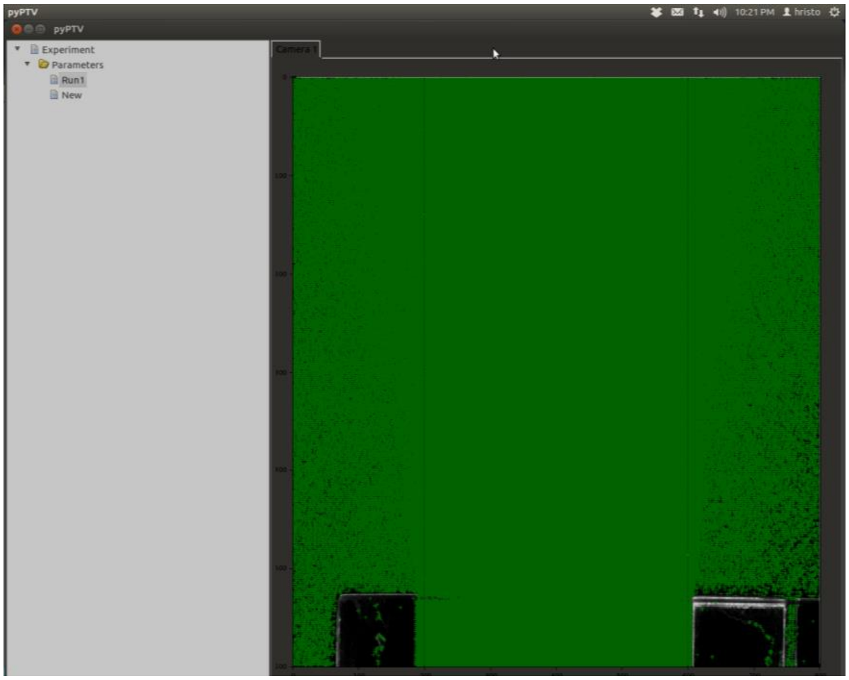


Figure 80: All detected particles in all frames for the front jet data

One can see that the laser light illuminated the inside of the jets as well, and that OpenPTV was able to detect those particles. Furthermore, the left and right side of the image in the latter figure have black spots which indicate that no particles were found there, and it is the author’s presumption that the code correctly identifies this region as relatively stagnant due to the lack of relative motion of the particles. Roughly one third of the detected particles were linked to each other.

3. Results

The results obtained from the OpenPTV do not compare well with the LDV and 2D-PTV data.

- **Frontal view - center**

Figure 81 shows the streamwise velocity normalized by the maximum streamwise velocity magnitude. Despite the author's attempts to improve the results by varying the parameters and the grid size, the two jet streams are not easily detectable, even with a coarse scale.

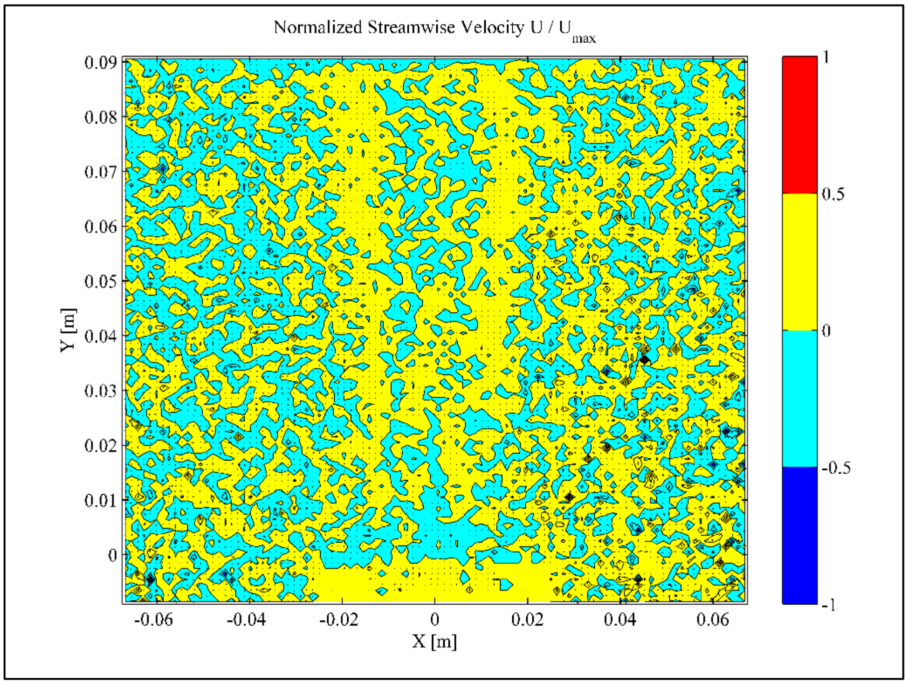


Figure 81: Streamwise velocity at center view

The result for the lateral velocity, shown in Figure 82 is slightly better as one can validate that the behavior matches that of the LDV and 2D-PTV, however the contours are not clearly defined.

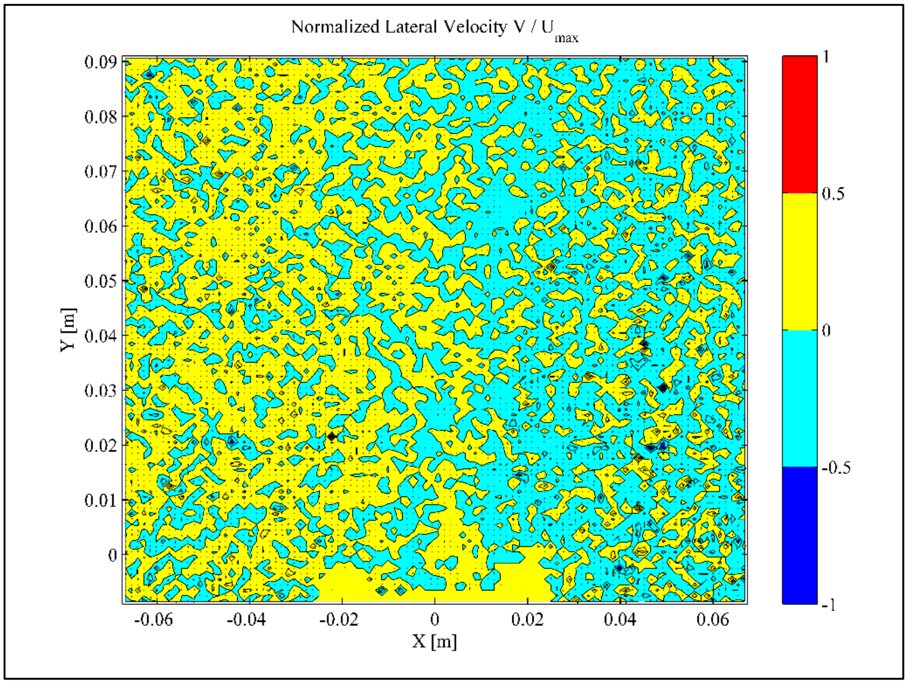


Figure 82: Lateral velocity at center view

The left side of image shows that the particles are being drawn to the right, while the reverse occurs on the right side. Corresponding cross sectional velocities at the previously used heights above the jet are plotted in Figures Figure 83 and Figure 84. Not only can one not see the expected humps representing the two streams, but there are also velocities found to the sides of the jets which is not physically accurate.

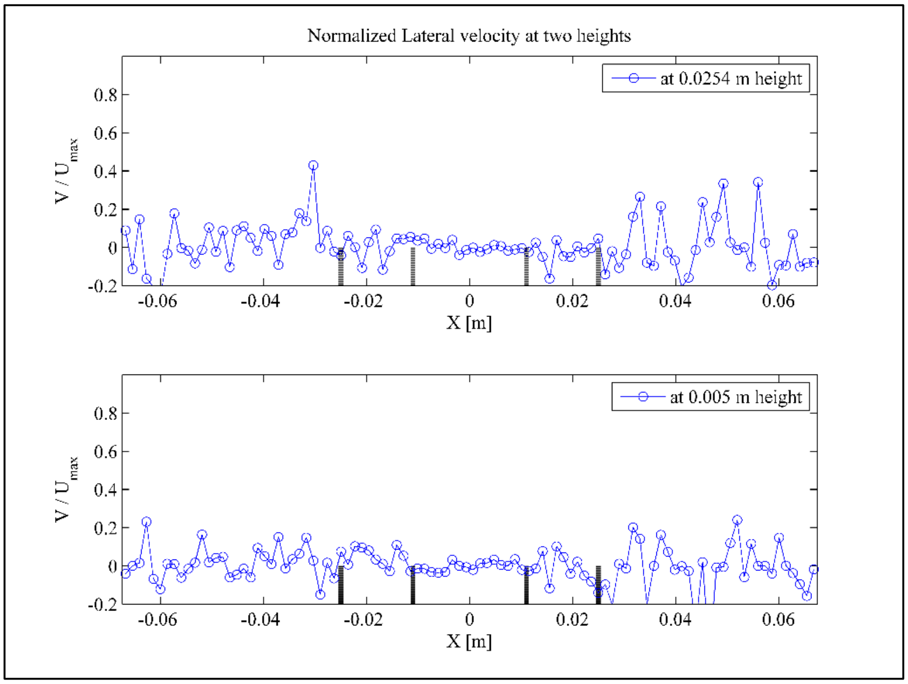


Figure 83: Lateral velocity at cross sections

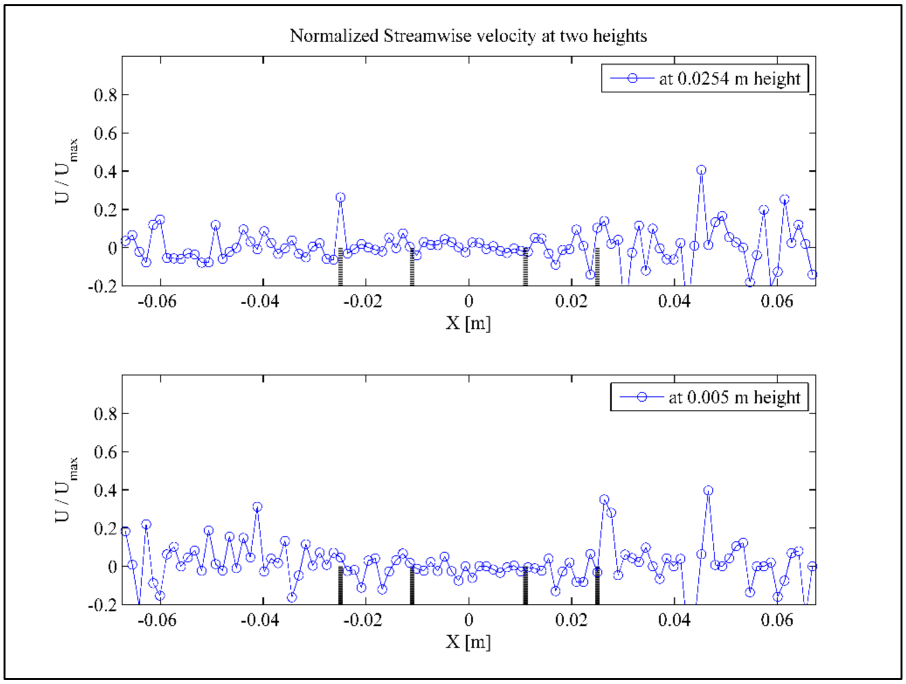


Figure 84: Streamwise velocity at cross sections

For comparison with the LDV results, the U and V fluctuations are also plotted in Figure 85.

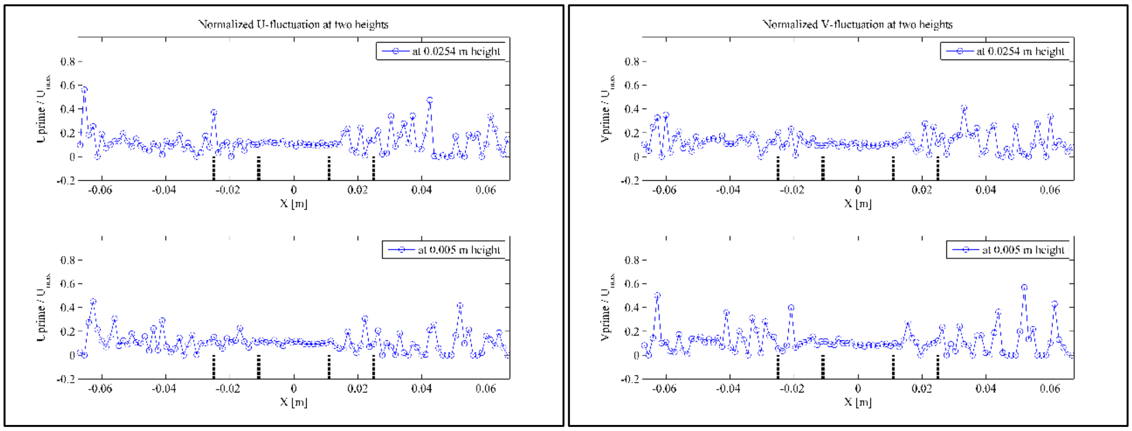
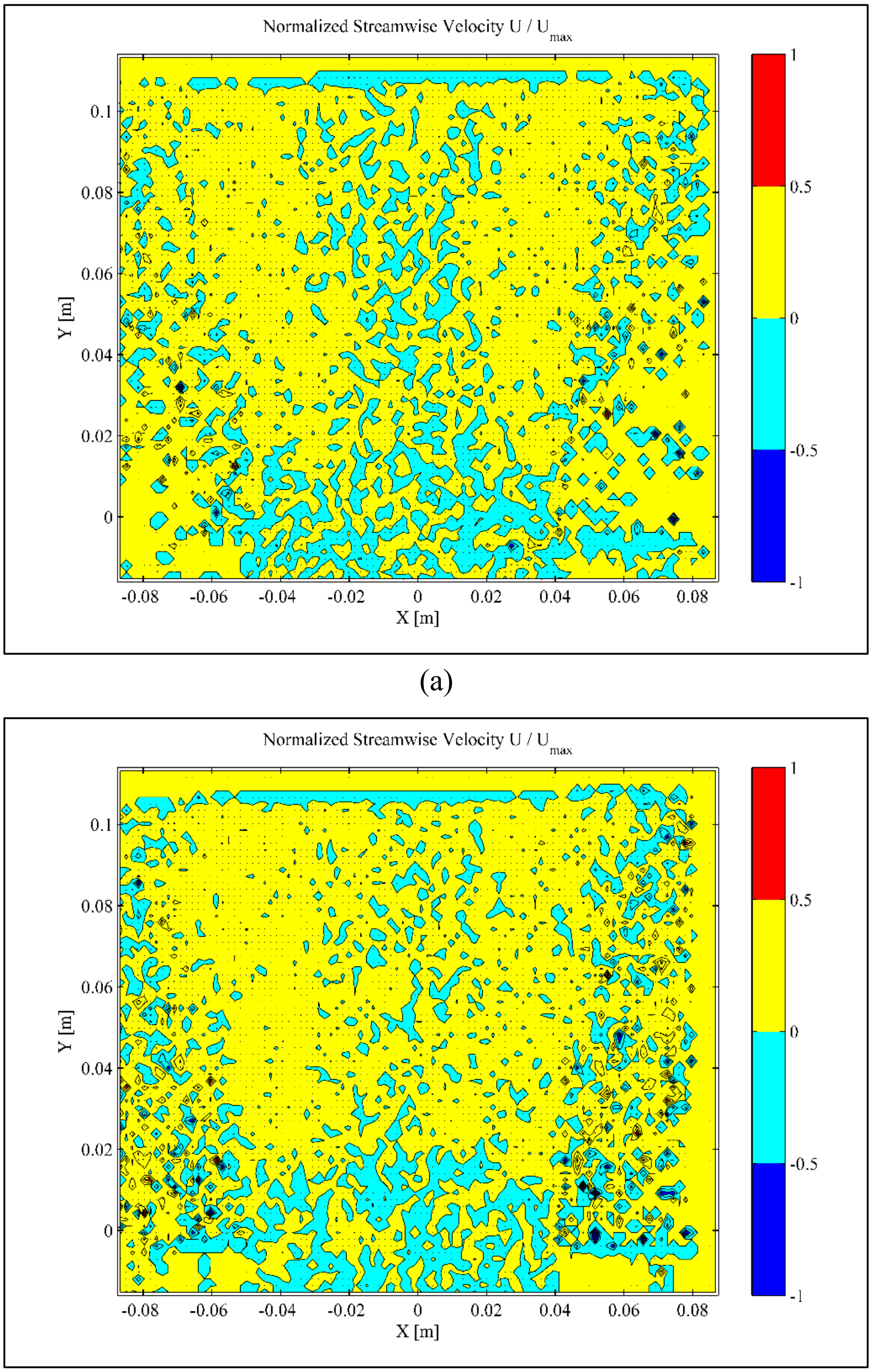


Figure 85: U and V fluctuations at cross sections

- **Rotated view - back and front jets**

The change of view did not improve the results in a noticeable way, but in Figure 86 which shows the streamwise velocity, one can see a shape, which the author assumes, resembles the side view of the jet streams. The lateral velocity plots and fluctuation results were not matching the LDV and 2D-PTV data well, and were thus omitted from this document, but can be provided at request. For informational purposes, the grid size used to average the vectors is 100 cells in the X direction, and 80 cells in the Y direction.



(a)

(b)

Figure 86: Streamwise velocity from (a) back and (b) front jet

After consultation with one of the members of the OpenPTV consortium, it was confirmed that the set-up of the equipment is not suitable for the OpenPTV code, or particle tracking in the literal sense. The laser used to illuminate the particles, shoots pulses at synchronized times for only 2 of 3 frames; between the first two frames the time interval is 2 ms, however between the second and third frames it is 33 ms. This discrepancy is too large for the code to handle, as the particle can move roughly 15 times farther during the longer interval. The concept of having this difference in time delay, is referred to as frame straddling, and is used in most PIV studies, as well as some PTV codes like the 2D-PTV code in the lab at Texas A&M. The author attempted to compensate for this discrepancy by only using the frames with 2 ms between them, however the results are not very good as shown in this section. A consequence of the frame straddling is that the tracking algorithm performs very poorly, as it is based on a kinetic model and not cross-correlation like PIV codes and the labs 2D-PTV code. Moreover, no trajectory reconstruction can be performed at these conditions, as particles will most likely leave the target area during the 33 ms.

In order to avoid this problem in the 3D study, the author used a halogen lamp and a continuous laser which provide continuous light, as was modeled in the standard test case data, and as used by the other members of the OpenPTV consortium.

5.2.2 **3D study**

In the first attempted configuration in which the laser was positioned to the side of the tank, no good data was obtained. When the laser was moved to the top of the tank, the obtained images looked much better in regard to brightness and focus. However, the unforseen difference in camera parameters among all three cameras has resulted in very difficult conditions for the OpenPTV software to perform to its potential.

The first step in the analysis is the calibration. Figure 87 shows the output of the sortgrid subroutine in OpenPTV which attempts to position the known target points in their respective image coordinates for each view. Despite a tedious trial-and-error process in which the author tried to match the crosses with the centroids of the points, the orientation setting could not be improved. After a careful investigation of the calibration images using ImageJ, it was determined that there is not enough grayscale contrast between the target points and the (rougher than expected) surface of the calibration target. For future experiments the author recommends designing and building a target with white target points printed on a black target, preferable made of some kind of metal and a CNC machine with very high accuracy. As the reconstructed position of each particle is directly related to the accuracy of the camera calibration, one needs to use a target with easily detectable points from all views. A possible alternative to the current design, one could print points on a transparency and laminate it to prevent water damage.

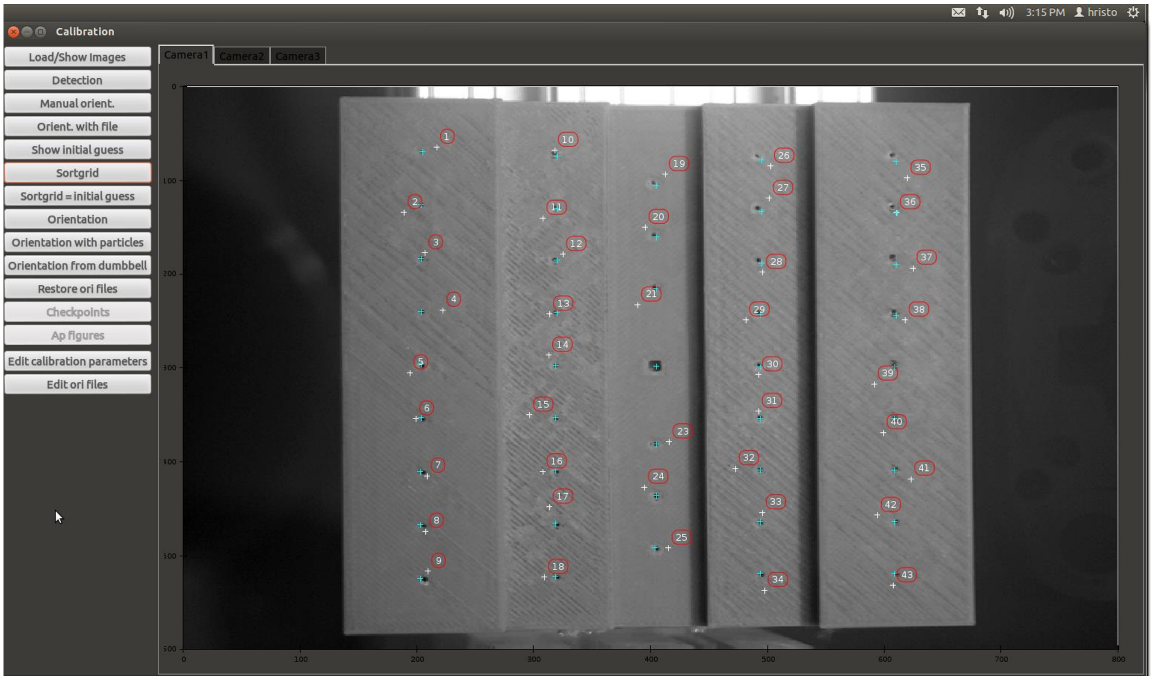


Figure 87: Bad sorting of target points for left camera

One can also observe that point 22 which represents the origin of the calibration target was not found. A quick analysis of the calibration images in the popular ImageJ showed that the target points have a very similar grayscale to the surface of the target. This means that either the current target has to be painted black to lower the surface grayscale, a new target with larger target holes will have to be manufactured, or that the images will have to be pre-processed such that the points can easily be detected. The orientation subroutine yielded a calibration with quite large errors, as seen in Figure 88.

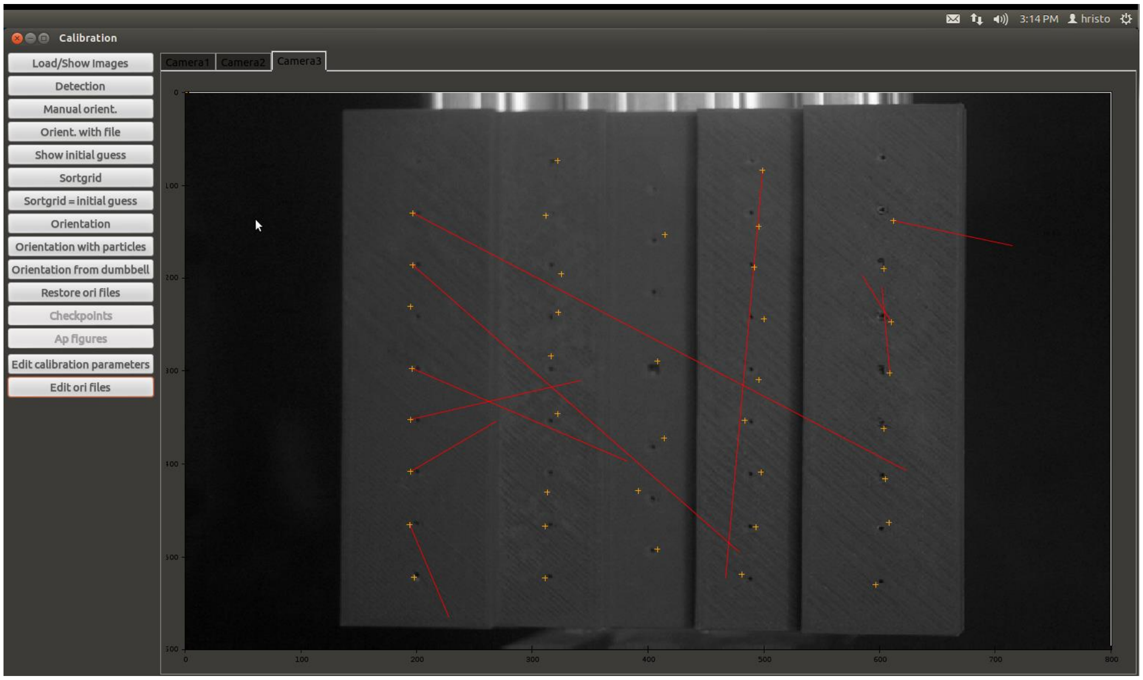
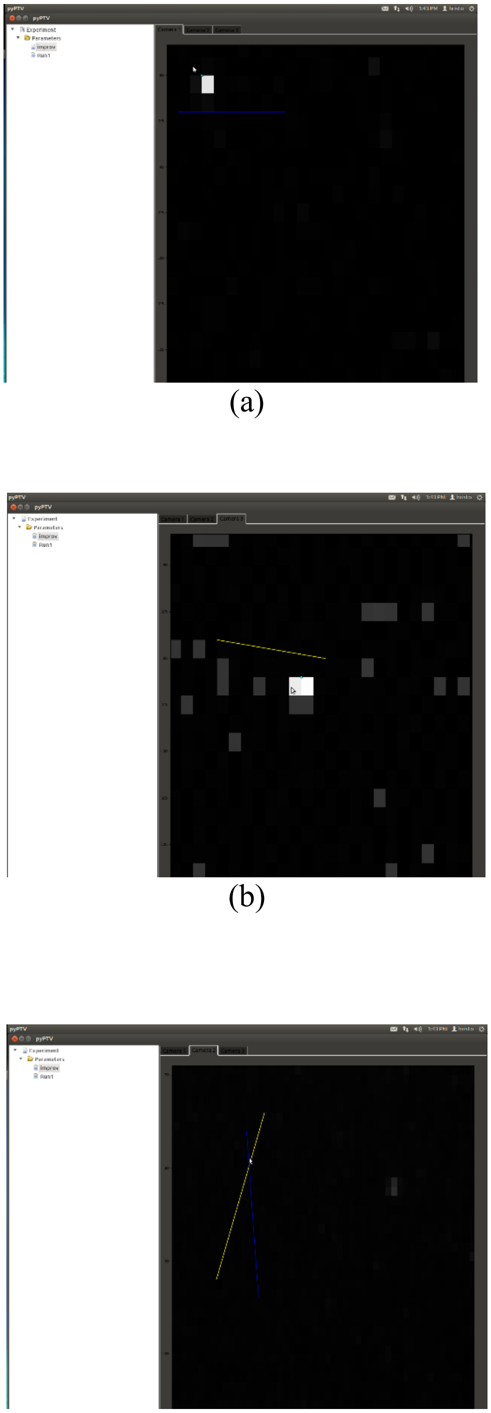


Figure 88: Bad orientation for right camera

The unsatisfactory calibration is expected to result in high reconstruction inaccuracies and significant mismatching of particles. An epipolar geometry analysis was performed to determine how much the bad calibration will affect the results. Figure 89 shows one particle which was seen in the left and right cameras, but not in the center camera.



(a)

(b)

(c)

Figure 89: (a) left, (b), right, and (c) center camera

In (a) a particle is chosen which creates a yellow line in the other two views. Then the view is changed to the right camera (b) where the same particle is selected by the user based on the distance to the epipolar line and an educated guess of which particle on the line is the same particle in (a). When the particle is chosen in the right view, the blue line is created in the left and center views. Looking at (c), one can not see a particle in the vicinity of the intersection between the two epipolar lines. This indicates that no match could be made, and thus this particle is either matched only in two views, or lost completely. Note that the epipolar visualization is for visualization purposes only, and does not have to be performed for analysis.

To show preliminary results, the author ran the OpenPTV software to determine whether any useful results could be obtained despite the unfavorable conditions. Figure 90 shows the detected locations of all particles throughout the image sequence for the left view. The jet stream is clearly visible which indicates that the particles detected in the left view should yield somewhat useful results.

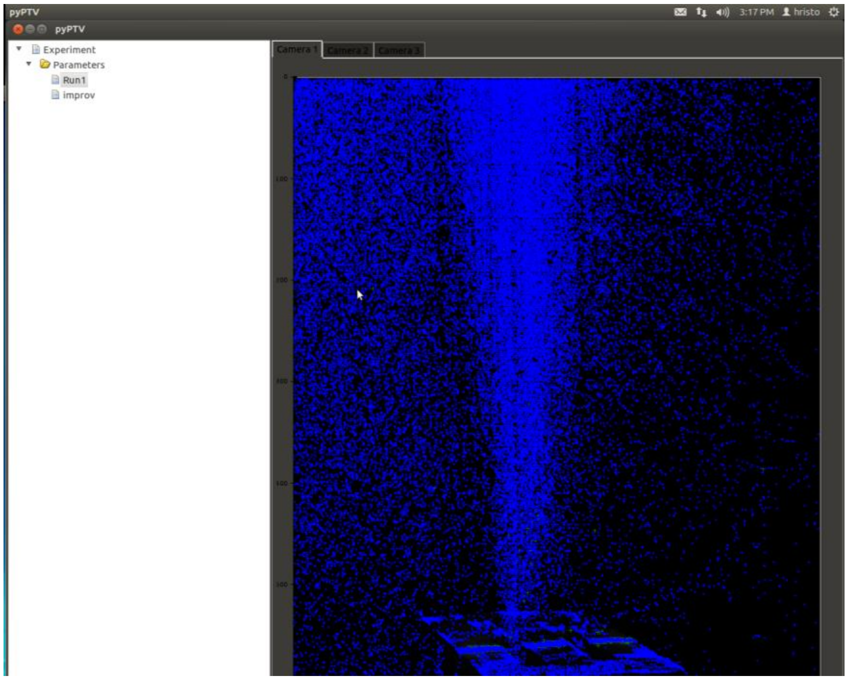


Figure 90: Left view detected particles

After a post-processing analysis, the data obtained from the current images and using the bad calibration parameters has resulted in unusable results. The grid averaged vector field using a grid size of 10 by 10 by 10 volumes is shown in Figure 91. Only 11 different vectors were found and they point in directions which are physically inconsistent with the flow. Similarly, the vorticity plot in Figure 92 is not very informative and seems to be incorrect compared with theory and the 2D-PTV result.

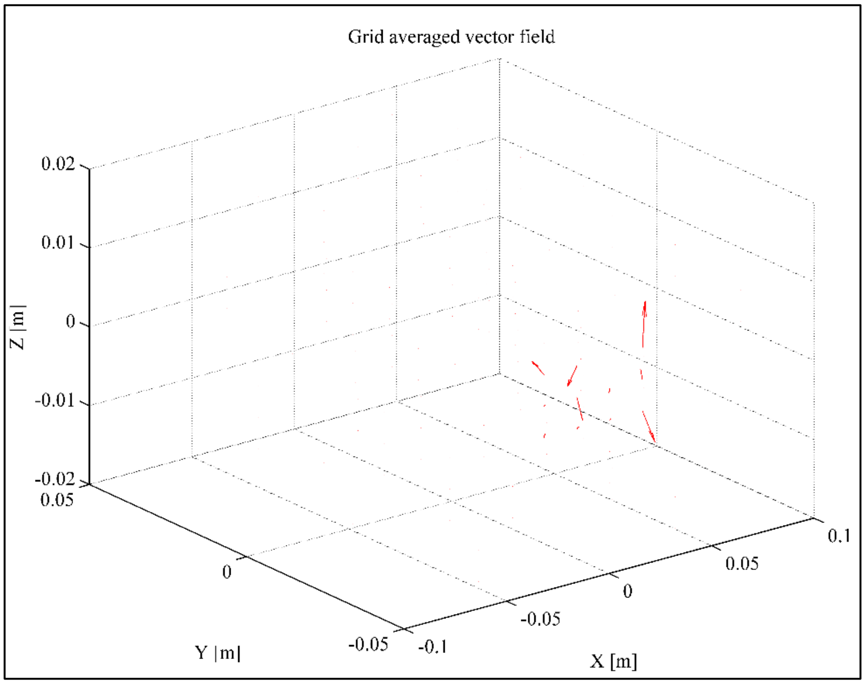


Figure 91: Bad velocity result

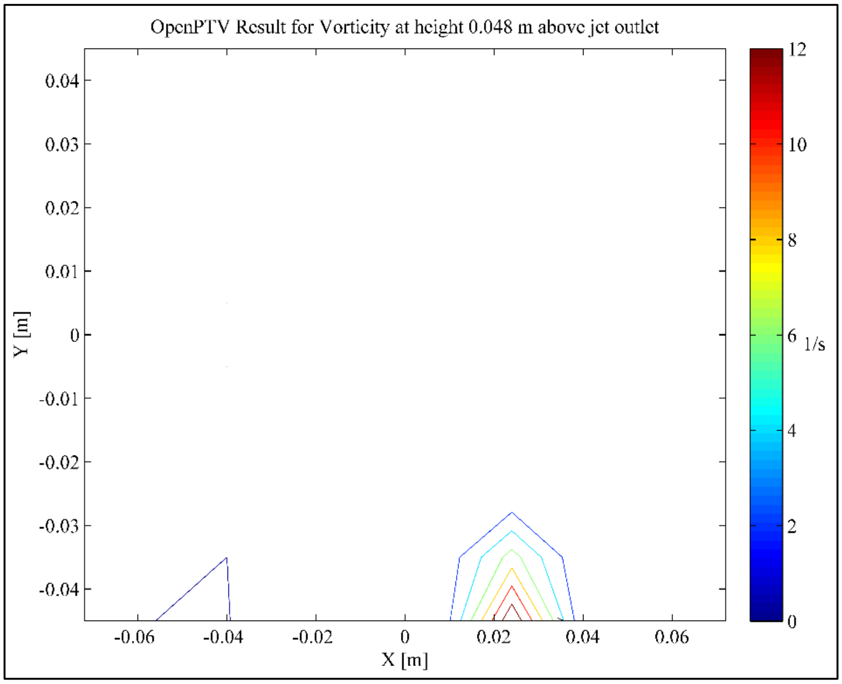


Figure 92: OpenPTV vorticity result

The only promising result from the preliminary analysis is the trajectory plot in Figure 93. A small, but still useful number of trajectories were reconstructed and point upward in the direction of the streamwise velocity as expected.

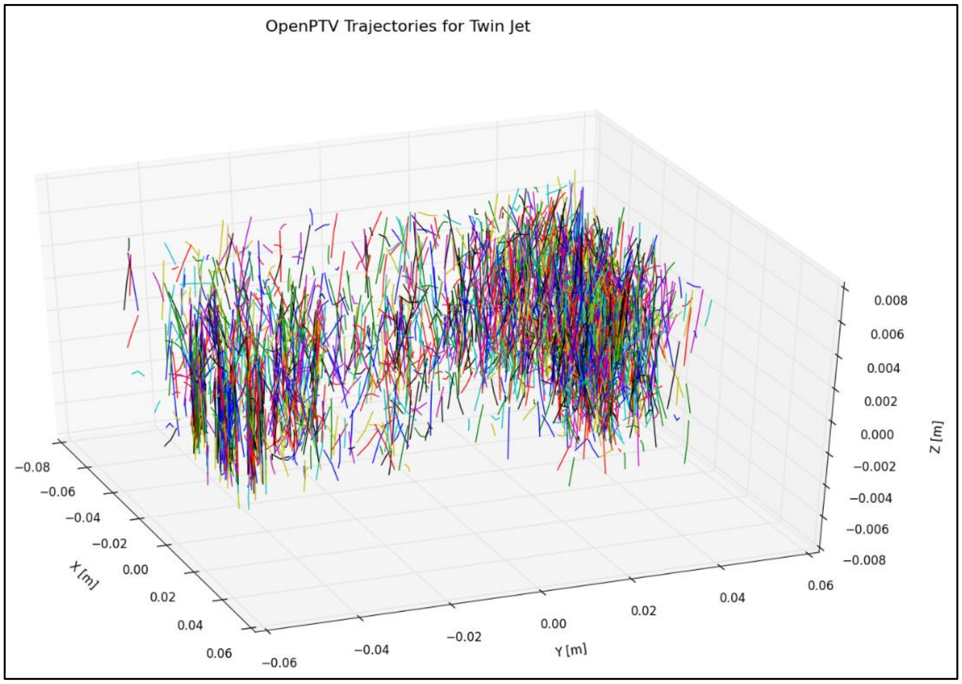


Figure 93: OpenPTV preliminary trajectories

As the calibration and image quality improve, the author expects to obtain much better results, and potentially see particle trajectories in the jet streams which merge indicating that the two jets mix.

Unfortunately, even after exchanging the equipment which provides continuous illumination and identical monochrome cameras, it was not possible to obtain good data to compare with the LDV and 2D-PTV study. Figure 94 shows the implementation of a halogen lamp, as well as a continuous laser to provide the required illumination.

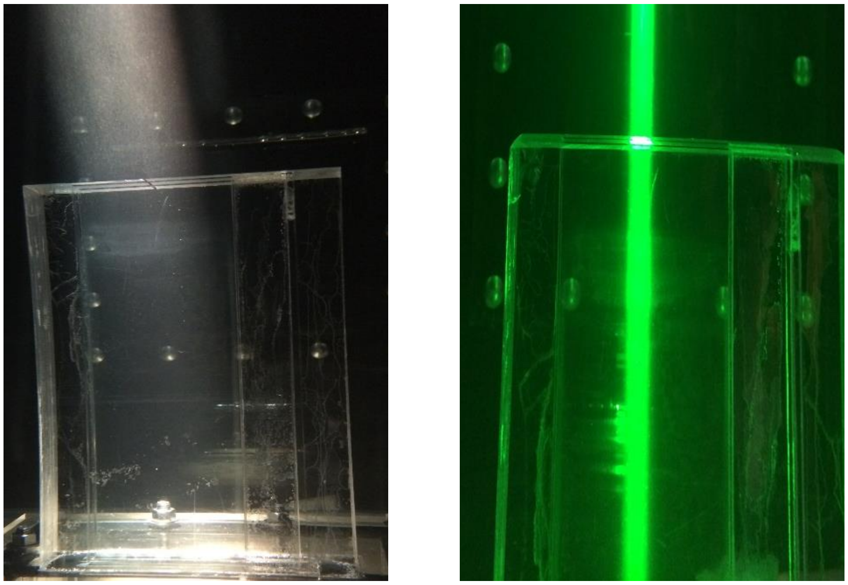


Figure 94: Halogen lamp and continuous laser

Inexperience with the cameras led to a reduced quality in the images in the two side cameras, however it was still better than with the color camera previously used. Figure 95 shows the resulting blurriness in the images which does not allow for correct particle identification, or tracking.

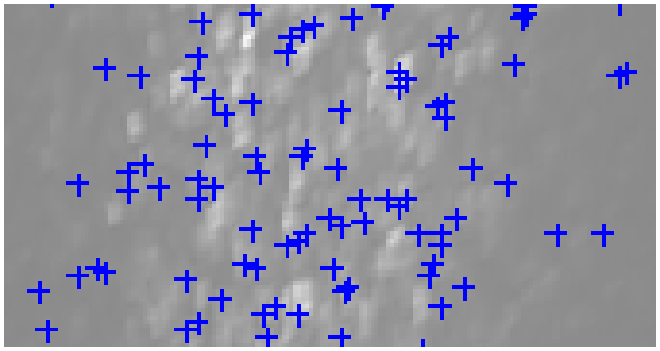


Figure 95: Blurry particles

No useful velocity plots could be obtained with the data, however the OpenPTV code was able to reconstruct 310 trajectories from the flow, as shown in Figure 96.

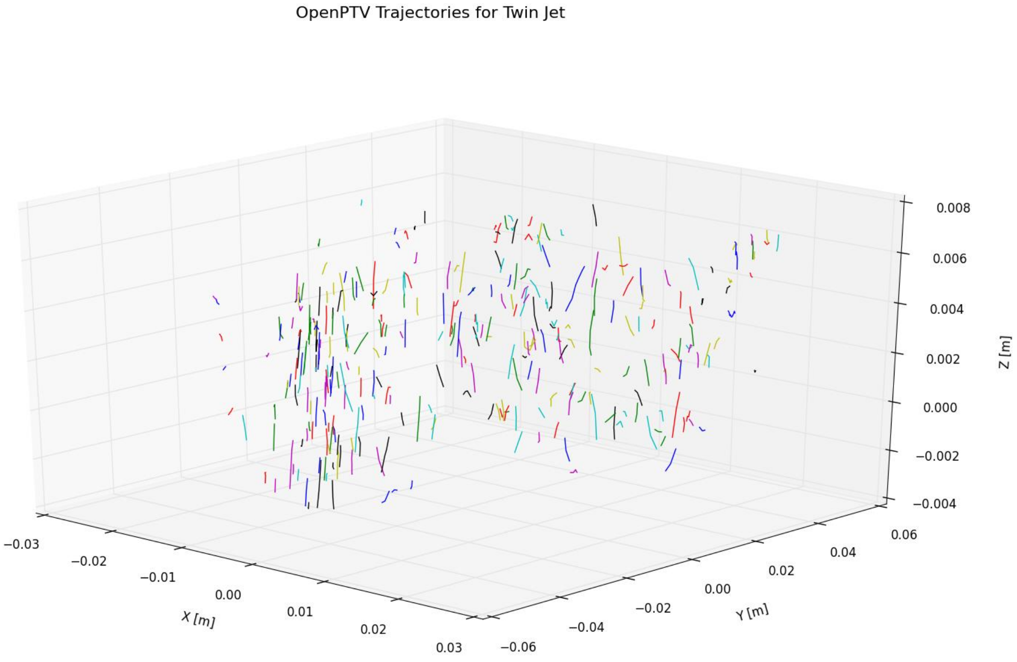


Figure 96: Trajectories from 3D twin jet study

Taking a close look at the vertical axis one can see that the OpenPTV code suggests large out-of-plane motion in the flow, which would be indicative of the mass of fluid which is pulled towards the jet stream by the negative pressure. This is expected to happen primarily at locations close to the surface of the jet surface, and to an increasingly smaller extent further up the stream. Chaotic turbulent motion is expected close to the jet surface as well as in the merging region of the jets where the separate masses of fluid hit each other and mix, before they begin exhibiting similar behavior to single jet flow. With better equipment and more time the author expects to capture a better image and representation of the flow characteristics and potentially match the results with the LDV, PIV, and CFD results.

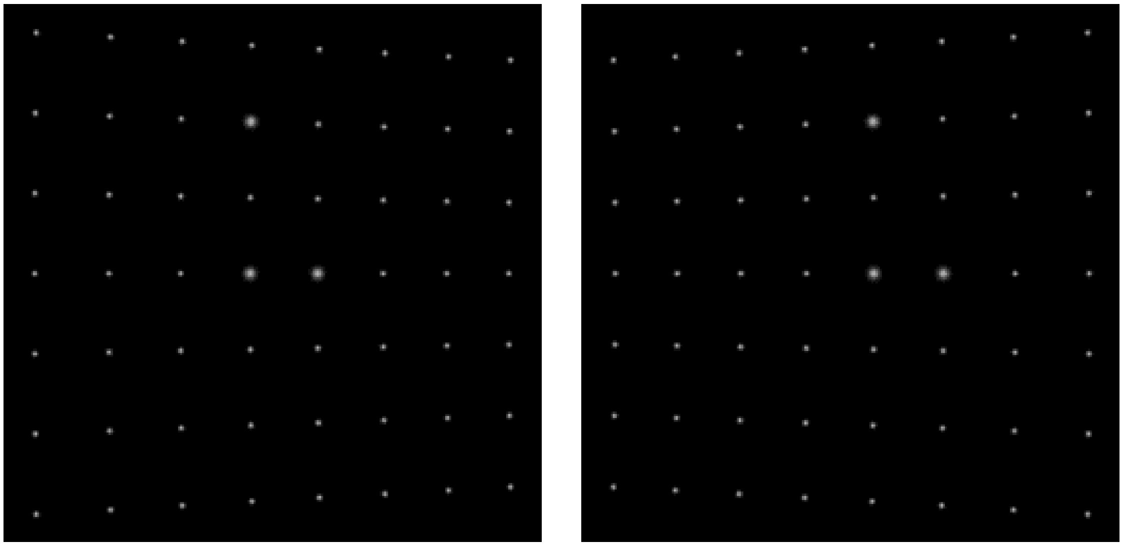
6. AUTHOR'S CONTRIBUTIONS

6.1 *Calibration – based reconstruction*

The OpenPTV software uses a geometric reconstruction according to the Tsai camera pinhole model. Before the images can be processed, the user has to obtain certain information about the system geometry, such as the focal length, thickness of the glass, refractive indices of all media between the camera lens and the tracer particles. Often in experiments it is very difficult or impossible to obtain some of these parameters due the complexity of the system. In addition, some geometries do not allow the insertion of a calibration target and make the necessary measurements, for example a (model of a) fuel rod bundle. A simple and very practical alternative calibration and reconstruction technique is to use an algebraic mapping function to relate the three object coordinates X, Y, and Z to the two pixel coordinates x, and y [16]. This technique was proposed by Soloff in 1997 [17]. The idea is to take images of calibration target points with known coordinates in three dimensional space, and select those points in the images using a software to obtain the pixel coordinates. The larger the number of points selected, the higher the accuracy of the mapping. The suggested function is a polynomial of order three in the *X* and *Y* directions and order two in the *Z* (out of plane) direction as follows:

$$\begin{aligned} F(x) &= a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_1^2 + a_5 x_1 x_2 + a_6 x_2^2 + a_7 x_2 x_3 + a_9 x_3^2 \\ &+ a_{10} x_1^3 + a_{11} x_1^2 x_2 + a_{12} x_1 x_2^2 + a_{13} x_2^3 + a_{14} x_1^2 x_3 + a_{15} x_1 x_2 x_3 + a_{16} x_2^2 x_3 + a_{17} x_1 x_3^2 + a_{18} x_2 x_3^2 \end{aligned} \tag{6.1}$$

where x is one of the two pixel coordinates, and x_1, x_2, x_3 are the respective object space coordinates. For each camera there will be two such equations with its own 18 parameters. This means that in order to work with 3D data, one will obtain a minimum of 72 separate coefficients, as at least two cameras are needed each with two equations. Due to the expected error in the reconstruction of the out of plane component, it is advised to use a high number of selected points in the calibration phase. The coefficients are computed through a least-squares algorithm. The author wrote a calibration script using this technique and verified it with data from another case provided by the Standard PIV challenge. Figure 97 shows two calibration images, one from the left and one from the right (the center camera image was left out for formatting reasons):



(a) (b)

Figure 97: Left and right calibration images of a sample test case

In this set, instead of using a three dimensional calibration target, there is only one plane which is moved to 5 different depths (*Z* coordinates). Each image has to be loaded and processed separately in order to obtain the appropriate coefficients for the 3D reconstruction. Figure 98 through Figure 100 shows the point selection screen in the author's MATLAB script.

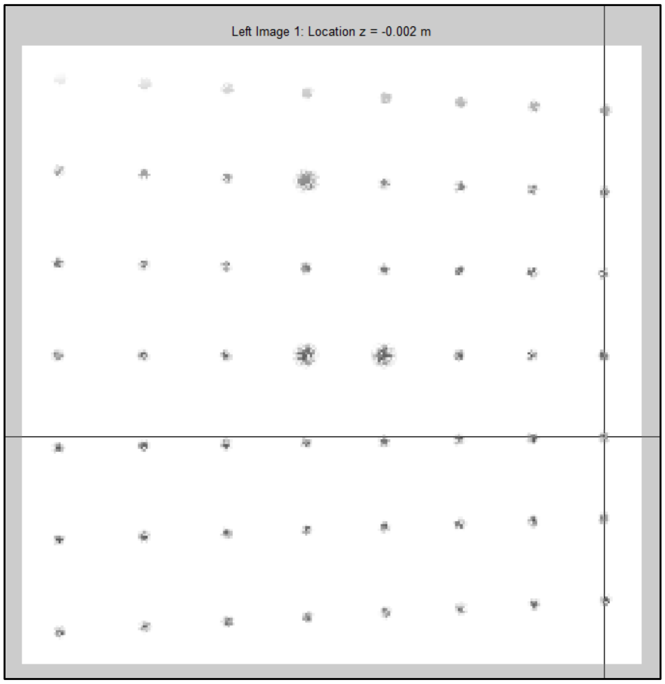


Figure 98: Left calibration image

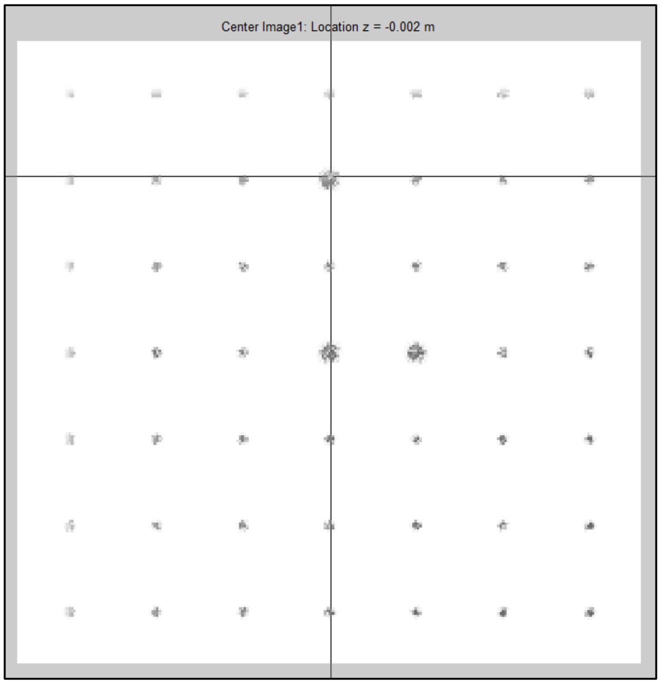


Figure 99: Center calibration image

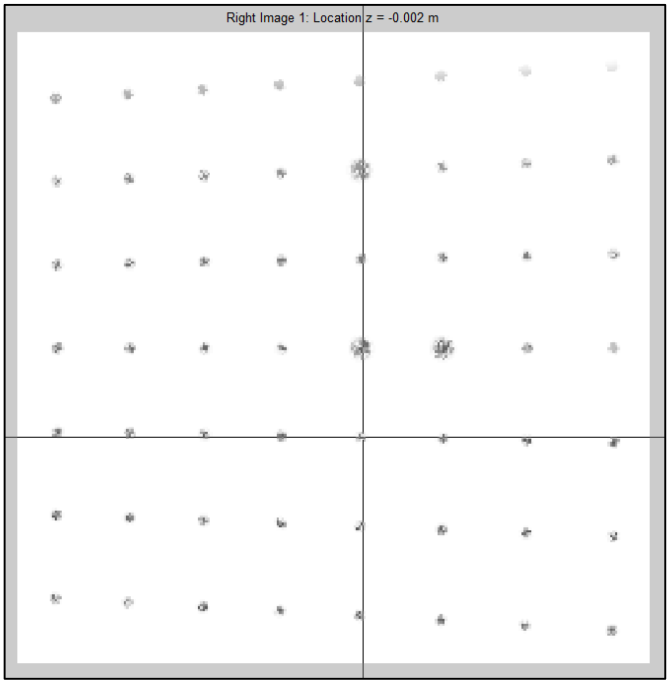
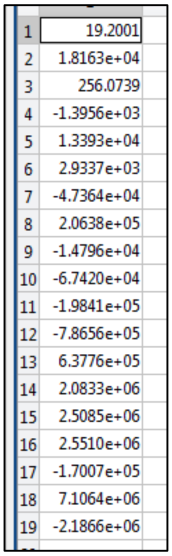


Figure 100: Right calibration image

In this test case, 49 points were selected from each of the 5 planes for each of the 3 cameras. This results to a total of 245 selected points per camera to be used for mapping using a least-squares algorithm. If desired, the number of points as well as the number of planes can be modified. The output of the script are arrays holding the coefficients for the mapping functions, and can be used for the epipolar constraint. Figure 101 shows the output of the coefficients for the *x* directions of the left camera.



1	19.2001
2	1.8163e+04
3	256.0739
4	-1.3956e+03
5	1.3393e+04
6	2.9337e+03
7	-4.7364e+04
8	2.0638e+05
9	-1.4796e+04
10	-6.7420e+04
11	-1.9841e+05
12	-7.8656e+05
13	6.3776e+05
14	2.0833e+06
15	2.5085e+06
16	2.5510e+06
17	-1.7007e+05
18	7.1064e+06
19	-2.1866e+06

Figure 101: Coefficients for mapping function

As a quick validation, the author chose 3 points with known three dimensional coordinates, and matched the predicted results with the output of the mapping functions. For this particular case the errors in x, and y directions were 0.929 and 0.624 pixels, respectively. This might seem a relatively high error, but one must keep in mind that the images are 256 by 256 pixels and it is impossible to select the exact position of the target point centroid which might lie at the edge of a pixel or elsewhere on the pixel resulting in non-integer pixel values. Figure 102 shows a case where the point is selected, but clearly does not lie at the centroid of the particle.

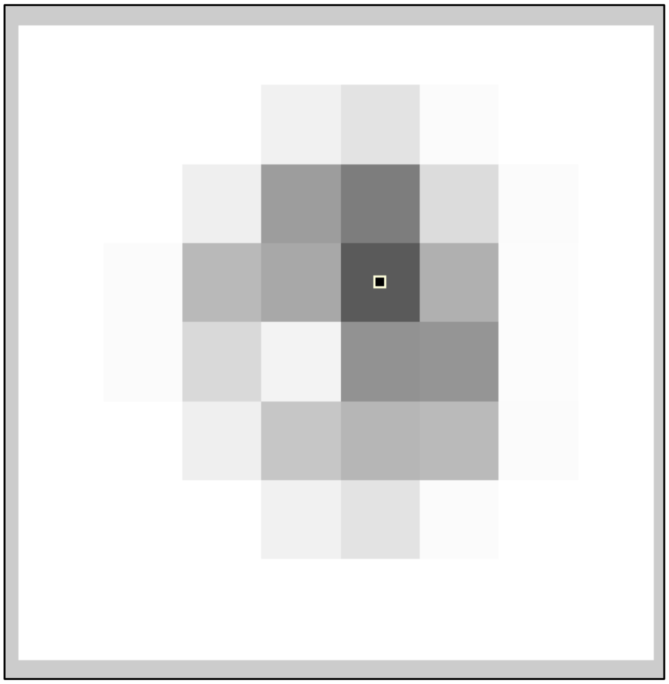


Figure 102: Misplaced centroid location restricted due to pixel coordinate

With a larger image size, this restriction will potentially be mitigated, but the added accuracy will result in high computation time and it is up to the user whether to increase the number of selected points / planes, use a larger image size, or use a particular least-squares method, or accept the result.

In order to use this technique in the OpenPTV software, the reconstruction algorithm will have to be changed according to [16], which fundamentally means to solve the inverse of the mapping function with a non-linear least-squares algorithm. The test case 352 images analyzed in this research were studied using the Tsai model, the Soloff model, as well as the Hall model [18]. Table 10 shows the reconstruction error of the calibration points for the test case, and Table 11 shows the reconstruction error for the seeded particles

Table 10: Reconstruction error of calibration points

Case	Mean [mm]			Standard deviation [mm]		
	Hall	Soloff	Tsai	Hall	Soloff	Tsai
I	0.046	0.034	0.046	0.012	0.021	0.012
II	0.116	0.033	0.105	0.069	0.020	0.062
III	0.116	0.034	0.105	0.069	0.020	0.062
IV	0.117	0.034	0.183	0.070	0.020	0.120
V	0.137	0.034	0.140	0.084	0.019	0.094

Table 11: Reconstruction error for first frame of test case 352 (273 particles)

Case	Mean [mm]			Standard deviation [mm]		
	Hall	Soloff	Tsai	Hall	Soloff	Tsai
I	0.043	0.039	0.047	0.010	0.019	0.017
II	0.166	0.047	0.112	0.127	0.037	0.102
III	0.166	0.047	0.112	0.127	0.037	0.102
IV	0.166	0.047	0.241	0.130	0.037	0.299
V	0.199	0.049	0.079	0.157	0.041	0.075

The error for the Tsai model is roughly three times higher than the Soloff model using 75 marker points. This means that the reconstruction of the object space coordinates of all particles will potentially be three times more accurate, and consequently the resulting velocity will be three times more accurate.

6.2 *Grid averaging*

The understanding of the author in regard to the current application of OpenPTV is that the consortium performs Lagrangian statistics studies, meaning that the interest lies in the Lagrangian field. In such studies there is no need to obtain averaged velocity vector results on structured grids to determine the average flow in a particular section of the flow domain. In contrast, PIV studies are used for the exact purpose of obtaining average vectors on specific grid locations, which potentially provide a better understanding of the flow distribution depending on the desired information or system behavior expectations. Fundamentally, grid averaging refers to loading the particles from all frames in one plane

(2D) or volume (3D) and selecting a certain size for the interrogation cells in which any present particle vectors will be averaged and assigned to the “cell” vector.

Given a steady state flow with a sufficiently high image density to cover the whole flow domain, but still within the available range for individual particle tracking, OpenPTV can be used to not only track particle trajectories, but using the author’s grid averaging script, the user can obtain averaged vector results with a specified grid size. As in CFD, the larger the grid size or mesh, the higher the resolution of the results and the “smoother” the results. Figure 103 shows the instantaneous vector field of the first frame as computer by the author’s velocity script, and Figure 104 and Figure 105 show the grid averaged result.

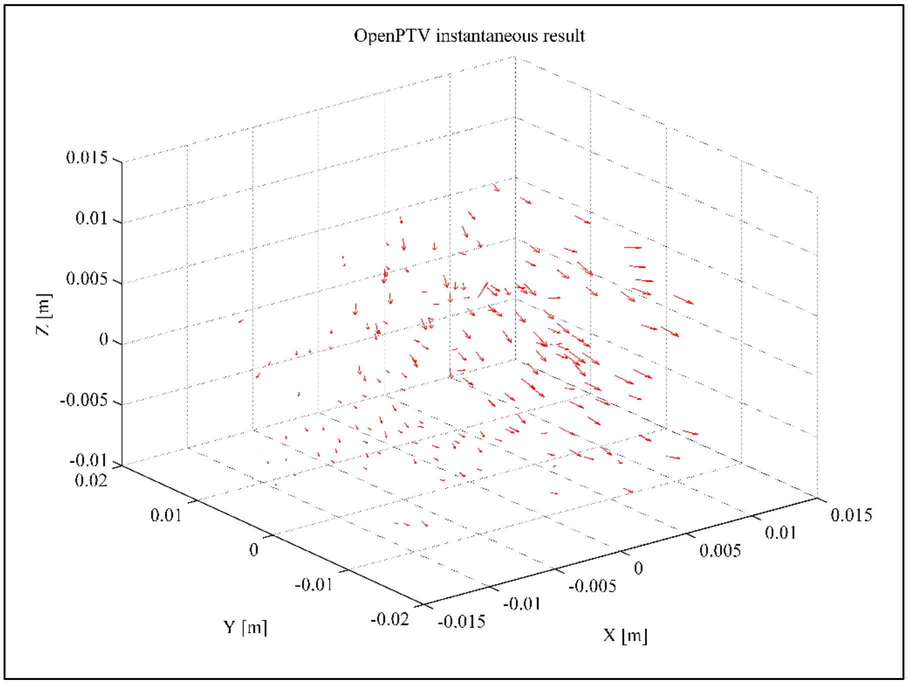


Figure 103: Instantaneous velocity

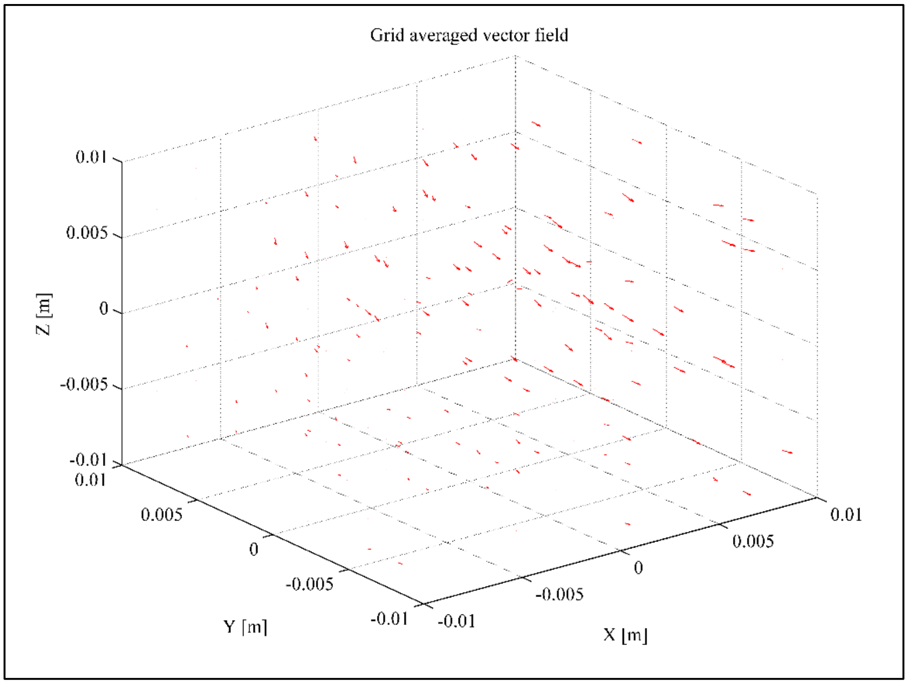


Figure 104: Grid averaged result

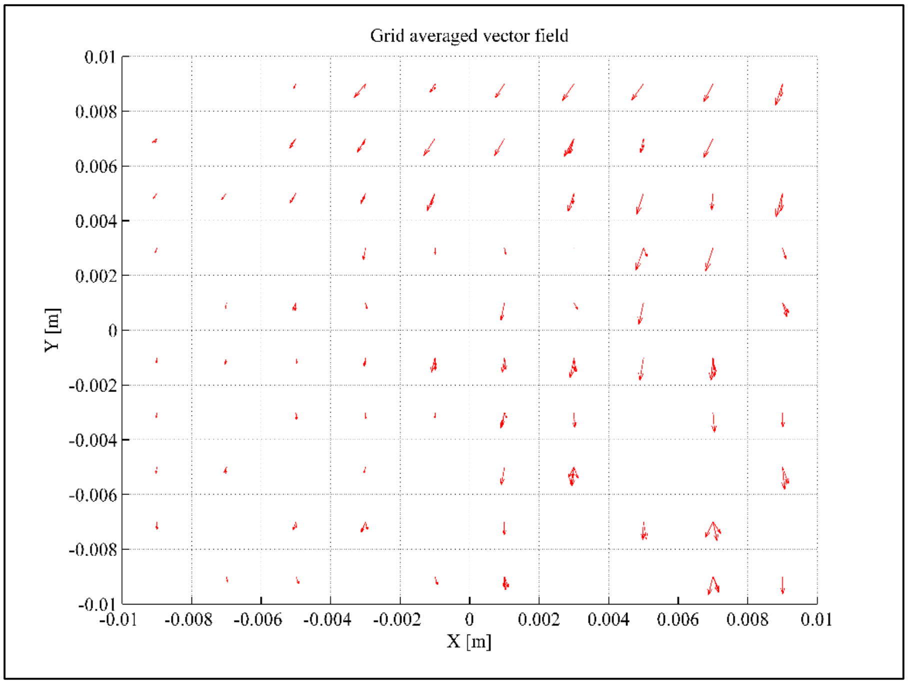


Figure 105: Better view of averaged grid (from top)

Looking at Figure 105 one notices some empty cells (regions). These empty cells indicate that no particle was found within the cell volume. Only one frame was used to generate the above figures, and considering the lower image density (compared to PIV) it is expected that one or more cells will be empty.

The grid allocated the averaged vector magnitudes and directions in the center position of each cell. If desired, a weighted averaged can be used instead and will only require a brief modification to the script. This grid is meant to both space and time average the information for studies like the determination of Reynolds stress components at specific locations.

7. CONCLUSION AND FUTURE WORK

7.1 *Conclusion*

Photogrammetric studies of flow distributions are becoming increasingly popular for various applications in the field of natural and applied sciences. As such, the development of more stable, accurate algorithms and softwares to perform three dimensional particle tracking velocimetry is gaining in interest and financial support. One 3D PTV code which is well developed, but free for all to use and contribute to, is OpenPTV. The original code was written and tested at ETH Zurich, and now has been released under open source licenses by what has now become the OpenPTV consortium, a collection of academic institutions which perform research in the field of experimental fluid mechanics.

The Laser Diagnostics Multiphase Flow Laboratory at Texas A&M University would like to become a new member of the consortium and help develop a better software. As an initial step, the author decided to provide the consortium with validation and for this purpose used available benchmark test data from the Standard PIV images created by the Visualization Society of Japan. The OpenPTV code was tested in regard to particle detection, particle position reconstruction, particle velocity reconstruction, as well as individual particle trajectory reconstruction. The performance of the code for each of these metrics is promising, and possible improvements were discovered.

As a second trial of validation, experimental data from the available twin-jet facility in the lab was gathered and analyzed using Laser Doppler Velocimetry, 2D-PTV and OpenPTV.

As a direct contribution to the OpenPTV consortium and the Laser Diagnostics Multiphase Flow Laboratory, the author wrote a script which performs the necessary calibration in a much more practical, efficient, and accurate manner than the current OpenPTV software. The results can be improved in accuracy by approximately 300 %, while the experimental set-up is simplified by mitigating the need to make measurements of focal length, camera angles, thickness of glass, or knowledge of individual media refractive indices. In addition, a script to ensemble average all vectors for statistically steady-state (or PIV) study was written for post-processing purposes and can be used to compute vorticity, and Reynolds stress.

7.2 *Future work*

The next step in the improvement of the OpenPTV code is to implement the author's calibration technique, and to expand it to also perform the reconstruction of the object space coordinates. Moreover, the interface of the code can be improved, and bugs which cause consistent crashes have to be removed in order to have a more stable working version. A possible direct collaboration with the various groups involved in the OpenPTV consortium should be considered, due to what appears to be similar interests among several groups.

REFERENCES

[1] Reyes, Denny L (2013). Three-Dimensional Velocity Measurement Reconstruction for a Rod Bundle Array using Matched Refractive Index Particle Tracking Velocimetry. Master's thesis, Texas A & M University, College Station, TX

[2] Willneff, Jochen. "A spatio-temporal matching algorithm for 3 D particle tracking velocimetry." *Mitteilungen- Institut fur Geodasie und Photogrammetrie an der Eidgenossischen Technischen Hochschule Zurich* (2003).

[3] Adrian, Ronald J., and Jerry Westerweel. *Particle image velocimetry*. Vol. 30. Cambridge University Press, New York, NY, 2011.

[4] Measurement Principles of PIV (Particle Image Velocimetry Measurement Principles) http://www.dantecdynamics.com/measurement-principles-of-piv

[5] Tropea, Cameron, Alexander L. Yarin, and John F. Foss, eds. *Springer handbook of experimental fluid mechanics*. Vol. 1. Springer, 2007.

[6] Raffel, Markus. *Particle image velocimetry: a practical guide*. Springer, Heidelberg, Germany 2007.

[7] "OpenPTV." . http://www.openptv.net/ (accessed July 21, 2014).

[8] "Hydromechanics Labs." http://www.ifu.ethz.ch/GWH/research/Facilities/Hydro_Lab/index_EN (accessed July 21, 2014).

[9] Tsai, Roger Y. "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses." *Robotics and Automation*, IEEE Journal of 3, no. 4 (1987): 323-344.

[10] "Pinhole Camera Model." . http://docs.mitk.org/nightly-qt4/PinholeCameraModel.png (accessed July 21, 2014).

[11] "Distortion." . http://fototips.ru/wpcontent/uploads/2008/09/distorton_barrel_and_pincushion.jpg (accessed July 21, 2014).

[12] Maas, H. G., A. Gruen, and D. Papantoniou. "Particle tracking velocimetry in three-dimensional flows." *Experiments in Fluids* 15, no. 2 (1993): 133-146.

[13] Okamoto, K., S. Nishio, T. Kobayashi, T. Saga, and K. Takehara. "Evaluation of the 3D-PIV standard images (PIV-STD project)." *Journal of Visualization* 3, no. 2 (2000): 115-123.

[14] Crosskey, Mark, and Arthur Ruggles. "UTK Twin Jet Water Facility Computational Fluid Dynamics Validation Data Set." *Proceedings of ICAPP 2014*.

[15] Gülan, Utku, Beat Lüthi, Markus Holzner, Alex Liberzon, Arkady Tsinober, and Wolfgang Kinzelbach. "Experimental study of aortic flow in the ascending aorta via Particle Tracking Velocimetry." *Experiments in fluids* 53, no. 5 (2012): 1469-1485.

[16] Prasad, Arun K. "Stereoscopic particle image velocimetry." *Experiments in fluids* 29, no. 2 (2000): 103-116.

[17] Soloff, Steven M., Ronald J. Adrian, and Zi-Chao Liu. "Distortion compensation for generalized stereoscopic particle image velocimetry." *Measurement science and technology* 8, no. 12 (1997): 1441.

[18] Joshi, Basanta, Kazuo Ohmi, and Kazuo Nose. "Comparative study of camera calibration models for 3D particle tracking velocimetry." *International Journal of Innovative Computing, Information and Control* 9, no. 5 (2013): 1971-1986.