

UVM ScholarWorks

Estimating particle velocity from dual-camera mixed reality video images using 3D particle tracking velocimetry

Item Type	thesis;article
Authors	Chivers, Thomas
Download date	2026-06-11 11:01:19
Link to Item	https://hdl.handle.net/20.500.14849/3614

ESTIMATING PARTICLE VELOCITY FROM DUAL-CAMERA MIXED REALITY
VIDEO IMAGES USING 3D PARTICLE TRACKING VELOCIMETRY

A Thesis Presented

by

Thomas Chivers

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Master of Science
Specializing in Mechanical Engineering

May, 2023

Defense Date: February 13, 2023
Thesis Examination Committee:

Jeffrey S. Marshall, Ph.D., Advisor
Safwan Wshah, Ph.D., Chairperson
William F. Louisos, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of the Graduate College

ABSTRACT

Mixed reality (MR) systems integrate diverse sensors, allowing users to better visualize and quantify surrounding environmental processes. Some existing mixed reality headsets include synchronized front-facing cameras that, among other things, can be used to track naturally occurring tracer particles (such as dust or snowflakes) to estimate particle velocity field in real time. The current work presents a 3D particle tracking velocimetry (PTV) method for use with MR systems, which combines various monocular cues to match particles between corresponding stereo images. Binocular disparity is used to estimate particle distance from an observer. Individual particles are tracked through time and used to construct the vector field of a scene. A digital display of velocity vectors can be broadcasted into a user's surrounding environment with the MR headset to be used as a flow visualization tool. The mixed reality particle tracking velocimetry (MR-PTV) approach was optimized to perform in natural conditions where particle size, particle color, and lighting are non-uniform. The approach was first tested using synthetic particle image data obtained by discrete element method simulations then experimentally validated for particles transported by a flume flow using the Microsoft HoloLens 2 MR headset. Uniform flow and flow around a body were considered experimentally. Experimental velocity measurements are compared to computational fluid dynamics results. The resulting MR-PTV system can be used for a variety of industry, scientific and recreational purposes for field-based measurement of particle velocities in real time.

ACKNOWLEDGEMENTS

Research accomplishments involve a great deal of resources and collaboration. Firstly, I would like to thank my advisor, Dr. Jeffrey Marshall, for his guidance and support throughout the entirety of the presented work. His wealth of knowledge in fluid mechanics, and talented ability in computer programming were invaluable assets to the work. His guidance as an advisor and timely responsiveness allowed me to be successful as a student and researcher.

I would also like to thank Sophie Gessman for her contributions to the research, and for becoming a HoloLens guru. Sophie was instrumental in experimental design and testing. Next, I would like to thank Scott Michael Slone for his assistance and access to the HoloLens 2 MR headset. Lastly, thank you to my family for everlasting support and trust.

This research is based upon work supported by the Broad Agency Announcement Program and Cold Regions Research and Engineering Laboratory (ERDC CRREL) under Contract No. W913E521C0003.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	ii
LIST OF FIGURES	vii
CHAPTER 1: MOTIVATION & OBJECTIVE	1
1.1. Motivation.....	1
1.1.1. Development of Augmented Reality Technology	2
1.1.2. Target Applications.....	4
1.2. Objective.....	5
CHAPTER 2: LITERATURE REVIEW	7
2.1. Particle Tracking Velocimetry (PTV) Method and Sources of Error	7
2.1.1. PTV Method	7
2.1.2. PTV Sources of Error	10
2.2. Volumetric PTV Methods.....	13
2.2.1. Tomographic PTV	14
2.2.2. Defocus PTV.....	17
2.2.3. Photogrammetric PTV	19
2.2.4. Holographic PTV	22
2.3. Depth Estimation in Computer Visualization.....	25
2.3.1. Single Camera Depth Estimation	26

2.3.2. Multiple Camera Depth Estimation	28
CHAPTER 3: METHODOLOGY	32
3.1. Particle Detection.....	34
3.2. Particle Linking.....	36
3.3. Stereo Matching Algorithm	40
3.4. Depth Estimations with Binocular disparity	43
3.5. Moving Least-squares Differentiation	49
CHAPTER 4: UNIFORM FLOW VALIDATION.....	52
4.1. Uniform Flow Computational Validation.....	52
4.1.1. Synthetic Image Generation.....	52
4.1.2. Synthetic Data results	54
4.2. Flume Experiment.....	55
4.2.1. Flume Method.....	55
4.2.2. Flume Results.....	58
4.3. Wind Tunnel Experiment.....	59
4.3.1. Wind Tunnel Method.....	59
4.3.2. Wind Tunnel Results.....	61
CHAPTER 5: ISOTROPIC TURBULENCE VALIDATION	64
5.1. Turbulence Computational Method	64

5.2. Turbulent Results 66

CHAPTER 6: SLUICE GATE VALIDATION..... 69

6.1. Sluice Gate Method..... 69

6.2. Sluice Gate Results 72

CHAPTER 7: CONCLUSION 75

REFERENCES 78

LIST OF TABLES

Table	Page
Table 1: Average and root-mean square values from MR-PTV measurements for the uniform flow flume validation test. Comparison values are the measured particle velocities based on time required to travel a prescribed distance.....	59
Table 2: Average and root-mean square values from MR-PTV measurements for the uniform flow wind tunnel validation test. Comparison values are the measured particle velocity based on time required to travel a prescribed distance for v and the measured fluid velocity for u	63
Table 3: Dimensionless simulation parameters and physical parameters of the fluid turbulence.....	66

LIST OF FIGURES

Figure	Page
Figure 1: Author wearing Microsoft HoloLens 2. Labels point to the location of front facing stereo cameras.	4
Figure 2: River currents determine areas of sedimentation deposit and erosion (St Mary's Fluvial Studies).	5
Figure 3: 2D PIV/PTV system including a laser to illuminate tracer particles seeded in a flow. A camera captures images of tracer particles (https://www.kanomax.co.jp/img_data/file_731_1454635706.pdf).	8
Figure 4: Determination of particle centroid method for subpixel accuracy. (a) shows pixel intensities in the x-direction, (b) shows pixel intensities in the y-direction (Gollin et al., 2017).	12
Figure 5: Pixel projections orthogonal to image planes form a 3D voxel (Kitzhofer and Brücker, 2010).	15
Figure 6: Ghost particles occur at positions where intensities of two different particles overlap (Elsinga et al., 2011).	16
Figure 7: a. Standard imaging system; b. defocusing arrangement (Pereira et al., 2000).	17
Figure 8: A target particle image in the image plane is compared to a set of calibration images with known out of plane depth, z (Barnkob et al., 2015).	19
Figure 9: Epipolar geometry of a stereo camera configuration (Lopez-Martinez and Cuevas, 2020).	21
Figure 10: a) Off-axis light configuration and (b) off-axis virtual and real image reconstruction. (c) Lighting configuration for in-line holography and (d) the real and virtual image reconstruction (Katz and Sheng 2010).	23
Figure 11: Schematic drawing of in-line holographic PTV system (Coletti et al., 2016).	24
Figure 12: Input image, ground truth laser scan depth map, transfer learning depth map, and regression loss CNN (Alshahim and Wonka, 2018).	28
Figure 13: Adaptive disparity mapping results. (a) The original input image, (b) is the ground truth disparity map, (c) results from Gu et al. (2008), (d) results from Jung et al. (2015).	31
Figure 14: Left (L) and right (R) cameras and the associated global coordinate system (x, y, z). The image plane is indicated by a red line, with left camera image plane coordinates (X^L, Y^L) and right camera image plane coordinates (X^R, Y^R).	32
Figure 15: Block diagram of MR-PTV velocity calculation algorithm.	33
Figure 16: A sequence of images illustrating the particle linking algorithm. (a) A field of particles in the image plane of either the left or right camera at time	

	t_n . The hard-filter region surrounding a particle c is indicated by a red box.	
	(b) A close-up of the particles in the hard-filter region surrounding particle c at t_n .	
	(c) Four particles in the hard-filter box surrounding the estimated position (X_c, Y_c) of particle c at time t_{n+1} , which together constitute a set Q_b . Particle c is found to link to particle 4 from this set, which minimizes the particle linking function G_{bc} .	38
Figure 17:	Illustration of the stereo matching algorithm, showing a pair of stereo images from the left and right cameras. In the left image, we identify a particle n and a hard filter search region, indicated by a red box. Particles $a_m, b_m,$ and c_m in the lower right-hand image that fall in the hard filter region constitute a subset Q_n , from which the stereo match of n is identified as particle c_m , which minimizes the function F_{mn} defined in Eq. (8).	41
Figure 18:	Definition of three sets of similar triangles -- APa and CPc, ABC and ADP, and acb and aPd -- used in distance estimation based on binocular disparity.	43
Figure 19:	Plot of the predicted particle depth z obtained using the MR-PTV algorithm versus the known particle depth. Each particle is represented by a black circle, and an exact match is indicated by the red line. The MR-PTV predictions were obtained with coefficient values $(c_1, c_2, c_3) = (0.100, 0.036, 0.018)$, set so that each term in the matching function in Eq. (8) has approximately equal weight.	45
Figure 20:	Effectiveness of each term in the stereo matching function given in Eq. (8). After application of the hard filter given in Eq. (7), the images in the figure compare the particle depth predicted by the MR-PTV algorithm with the known depth for cases using (a) only the particle brightness term, (b) only the particle diameter term, (c) only the particle velocity term, and (d) only the particle height Y term. The red dashed line represents an exact match between the predicted and known depth.	46
Figure 21:	Root-mean-square error divided by the computational domain size L for the estimated positions in the x -direction (red circles), y -direction (green deltas), and z -direction (blue squares). Plots are shown comparing root-mean-square error with (a) number of pixels across the particle standard image-diameter D_{image} , (b) ratio of camera baseline distance h to the domain size L , and (c) particle area concentration in the image plane c_{image} .	48
Figure 22:	Plot of the x -component of velocity versus time for a uniform flow with velocity components $(1, 0, 0)$. The effect of the number of points M used in the moving-least-square differentiation procedure on the smoothness and accuracy of the differentiation is demonstrated, using computations with $M = 1$ point (red squares), 2 points (blue circles), and 5 points (green gradients) on each side of the target point.	51
Figure 23:	In (a), the computed particle velocity components in the u (red circles), v (green deltas), and w (blue squares) directions are plotted as functions of the particle depth from the camera. The dashed lines are the exact values of	

known particle velocities. In (b), the root-mean square velocity error is plotted for each component as functions of particle depth.	55
Figure 24: (Right) Laboratory testing set-up for flume experiment. (Left) Schematic diagram showing experimental set-up used for flume validation experiments.	56
Figure 25: Particle velocity measurements for uniform flow in a flume from MR-PTV with three data frames. Red circles and green triangles represent particle x and y velocity components, respectively.	58
Figure 26: Wind tunnel experimental setup. HoloLens 2 positioned perpendicular to flow direction in wind tunnel. Particle hopper pictured in upper left corner.	60
Figure 27: Particle velocity measurements for uniform wind tunnel flow from MR-PTV with three data frames. Red circles, green triangles, and blue squares represent particle x, y, and z velocity components, respectively.	62
Figure 28: Contour plot of the x-component of velocity in the isotropic turbulent flow.	67
Figure 29: (a-c) Scatter plots comparing the x-, y- and z-velocity components of particles in an isotropic turbulent flow computed using the MR-PTV algorithm versus the known particle velocity. (d) Root-mean-square error in predicted velocity as a function of particle depth z.	68
Figure 30: Experimental configuration of sluice gate flume flow.	70
Figure 31: Domain boundary conditions used in CFD simulation, where the origin is where the Microsoft HoloLens 2 was positioned perpendicular to the domain.	71
Figure 32: MR-PTV measured velocity vectors for flow past a sluice gate. (a) Scatter plot showing experimental particles (yellow) with MR-PTV velocity direction (black arrows) superimposed onto CFD contour plot of velocity field for flow past a sluice gate, with computed particle pathlines indicated by black lines. (b) Comparison of velocity magnitude at particle locations as measured experimentally using the MR-PTV algorithm (y-axis) and as simulated by solving the particle momentum equation with the computed velocity field (x-axis).	73

CHAPTER 1: MOTIVATION & OBJECTIVE

1.1. Motivation

Knowledge of local flow fields is used in a variety of applications including targeting, obscurant dispersal, positioning for chemical release, and tracking river flows. Particle tracking velocimetry (PTV) is a method used to measure velocity at high spatial resolution by tracking tracer particles through time to compute individual velocity vectors (Maas et al., 1993). Typically, PTV is used together with a laser to generate a sheet of light which illuminates the tracer particles at a set distance from the camera. The standard set-up for using PTV in a laboratory setting requires precise multi-camera configurations (Doh et al., 2012), a coherent light source (e.g., a laser) (Bryanston-Cross et al., 1992), a synchronizer, and calibration image stacks (Barnkob et al., 2015). These requirements make standard planar PTV difficult to use in field settings. Dedicated large-scale PTV methods have been used in field research to measure flow around city building models (Zhao et al., 2020) and snowflakes have been used as tracer particles to resolve large vortex structures that form at the tips of wind turbine blades (Hong et al., 2014, and Hong and Abraham, 2020). Such systems are not easily moved from location to location.

Much of the difficulty in taking accurate PTV measurements lies in having to illuminate the flow field with a high-power laser, which serves to identify particles at a fixed distance from the camera. This difficulty can be removed by use of a three-dimensional PTV system, which uses particles in a three-dimensional volume in which the measurement requires an estimate of the distance between each particle and the PTV camera(s). This method described in the current work integrates a novel three-dimensional

particle tracking velocimetry (PTV) algorithm with a mixed reality (MR) headset and is termed MR-PTV. The system is designed to be capable of measuring both particulate flows in a laboratory, as well as naturally occurring tracer particles in field studies. For instance, natural particles such as snowflakes, rain drops, falling leaves, or river floaters can be tracked through time, and individual velocity vectors for each particle are computed. Individual velocity vector data can then be displayed as holograms using the capabilities of an MR headset. The output yields local flow measurements over a three-dimensional spatial region with minimum specialized instrumentation, aside from the MR headset and a source of tracer particles.

1.1.1. Development of Augmented Reality Technology

Advancing augmented reality (AR) systems are a key technology focus. Many companies, including Microsoft, Lenovo, and Google, have released wearable “Mixed Reality” (MR) headsets that merge computer generated holograms with the real-world environment where virtual and physical objects are seen and interacted with in real-time. In 2023, Apple plans to unveil their own MR headset, complete with its own operating system. MR headsets provide immersive experiences which can be used for recreational purposes such as gaming and entertainment, but primary applications include academia, industry, and scientific use. In academia, modeling and simulations using MR headsets provide memorable learning experiences for students and ways for researchers to visualize and interact with their models. For instance, medical students can view and interact with 3D anatomical structures, as if they were performing surgery, aspiring musical conductors can practice their skills with a virtual orchestra, and history students can witness

simulations of historical battles. In industry, companies can build training routines for new hires to become acquainted with their work environment. Mechanics can follow tutorials on how to fix vehicle components, and remote employees can work with their coworkers in an immersive environment.

While small in size, MR headsets integrate significant hardware and software components. Hardware components generally consist of multiple synchronized cameras used for hand tracking and spatial awareness. For instance, the MR-PTV algorithm developed in the current work uses the synchronized front-facing stereo cameras on the Microsoft HoloLens 2 MR headset, shown in Figure 1, to track naturally occurring tracer particles through time to compute velocity vectors for individual particles. Future evolution of this project aims to develop software for the MR headset with the capability of displaying velocity vectors as holograms anchored in the real-world environment.



Figure 1: Author wearing Microsoft HoloLens 2. Labels point to the location of front facing stereo cameras.

1.1.2. Target Applications

Real-time local velocity measurements can be used in a variety of applications in other industries, such as wind farm planning and water surveying. Wind energy is a growing industry with significant promise as a sustainable energy source. Developers position wind farms in high wind areas to maximize power output. A real-time flow visualization tool is a beneficial aid in positioning individual wind towers. The MR-PTV system can also be used to track the velocity of objects on and in water. For example, water flows are commonly surveyed to estimate areas of sedimentation deposit and erosion as shown in Figure 2.

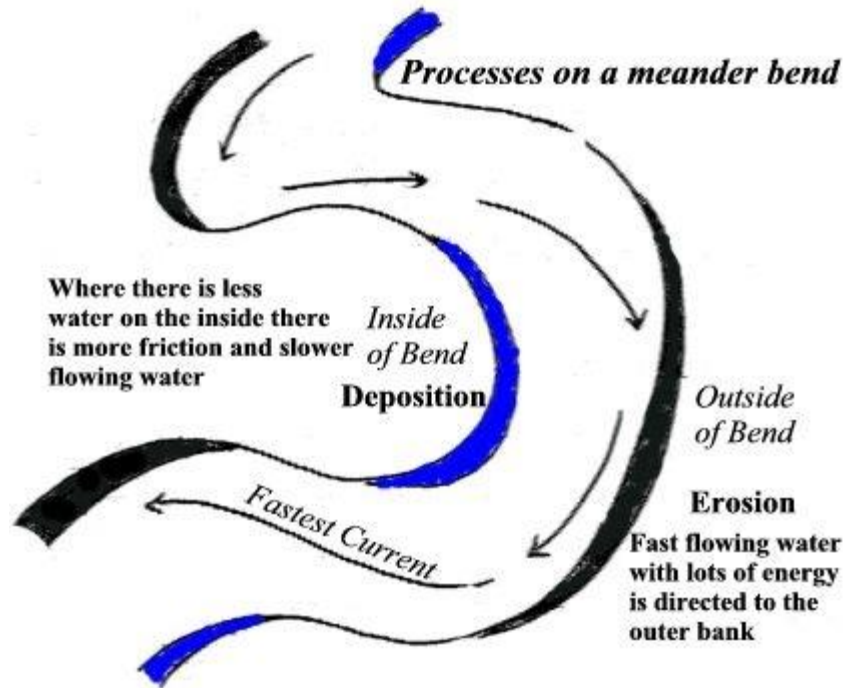


Figure 2: River currents determine areas of sedimentation deposit and erosion (St Mary's Fluvial Studies).

The MR-PTV system can measure floating objects, such as released floating tracer particles on the surface of moving water. By identifying areas of erosion and deposition scientists can predict areas vulnerable to erosion.

1.2. Objective

The objective of this study is to present the framework for a particle tracking velocimetry (PTV) system that is compatible for use with mixed reality (MR) headsets. The system is termed Mixed Reality Particle Tracking Velocimetry (MR-PTV) and uses two stereo cameras to track objects through space and time to compute the velocity field of a scene. After its development, the MR-PTV algorithm is validated in this work with velocity measurements from computational and experimental stereo image pairs. A

synthetic stereo camera code was developed that produces synchronized stereo image pairs of particles flowing through a three-dimensional volume. MR-PTV is used to compute three components of velocity in the computational volume containing particles flowing in both uniform and turbulent flow fields. Then, the Microsoft HoloLens 2 MR headset was used to experimentally capture stereo image pairs of particles flowing inside a flume. The MR-PTV system was used to compute velocity vectors of particles in the flume both for uniform flow and for flow under a submerged sluice gate.

The remaining sections of the thesis are summarized as follows. Chapter 2 contains a literature review covering existing PTV methods and sources of error, as well as depth estimation approaches used in the computer vision literature. The MR-PTV system methodology and algorithms are described in Chapter 3. Performance results for the MR-PTV system flow measurements for uniform flow for both computational and experimental data is presented in Chapter 4. Chapter 5 includes performance results for the MR-PTV system in synthetic data for isotropic turbulent flow. Experimental results from particles flowing under a sluice gate body in a flume are presented in Chapter 6. Conclusions are presented in Chapter 7.

CHAPTER 2: LITERATURE REVIEW

2.1. Particle Tracking Velocimetry (PTV) Method and Sources of Error

Particle tracking velocimetry (PTV) is a flow measurement technique used to measure velocity vectors of individual tracer particles seeded in a flow. PTV uses the same experimental setup as particle image velocimetry (PIV), a flow measurement technique which preceded PTV (Adrian 1986). Instead of tracking individual particles through time, PIV uses a double-pulsed laser to capture image pairs separated by a short interval of time. Image pairs are then processed using a spatial cross-correlation method to compute a single velocity vector for each interrogation window in an image. Section 2.1.1 describes the setup for general PIV/PTV systems. Sources of error in PTV measurements are described in Section 2.1.2.

2.1.1. PTV Method

PIV and PTV systems use images of particles to estimate flow field velocity components. Both methods use identical experimental setups for tracking two-dimensional velocity components. The setup, shown in Figure 3, consists of a camera oriented perpendicular to a thin laser sheet running through the test section of the flow, where the camera imaging and laser pulses are synchronized.

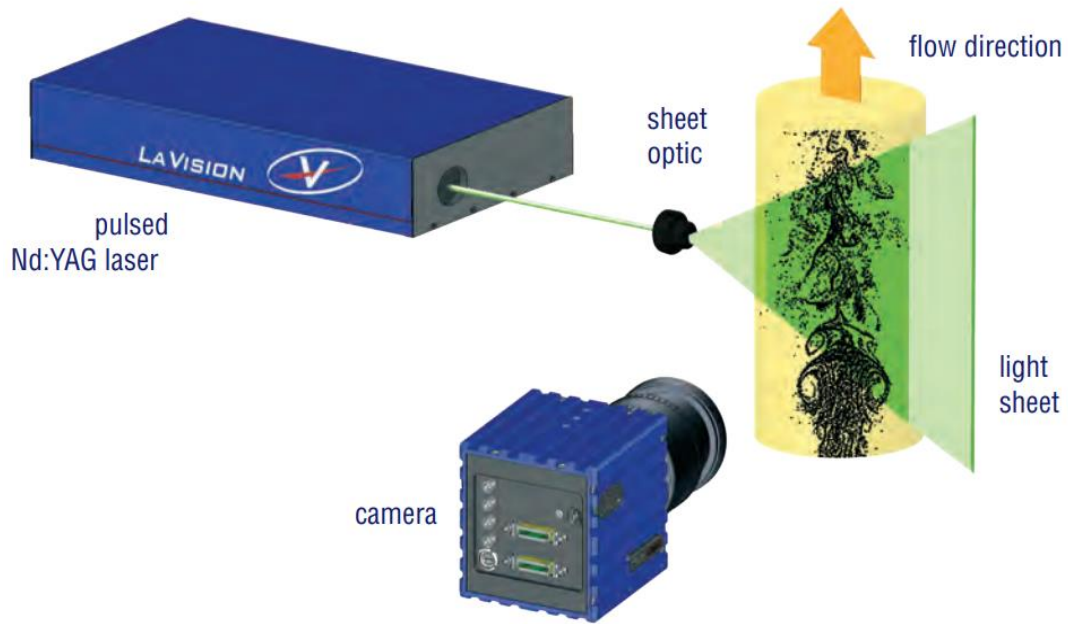


Figure 3: 2D PIV/PTV system including a laser to illuminate tracer particles seeded in a flow. A camera captures images of tracer particles (https://www.kanomax.co.jp/img_data/file_731_1454635706.pdf).

Tracer particles seeded in the flow are illuminated by the laser and appear as bright dots in images. The laser light sheet is an essential part of the system because it is impossible to calculate velocity of particle images without knowing the distance of the particles from the camera. Determining particle depth from a single image is extremely challenging. The laser light sheet illuminates particles at a known distance from the camera allowing for velocity calculations to be made.

While PIV and PTV methods share the same experimental setup, the post-processing approach and results of the two methods differ. The PIV approach segments an image into gridded “interrogation windows” and a spatial cross-correlation is used between image pairs from a double-pulsed laser, in which images are separated by a small time difference. The resulting output yields a single average velocity vector calculation for each

interrogation window. The theory behind the PIV algorithm makes two basic assumptions. First, the velocity field is assumed to be constant between two frames, and second, the velocity distribution within an interrogation window is assumed to remain constant. When measuring real flow, neither of these two conditions are easy to completely achieve. This leads to spatial resolution errors in sharp flow gradient measurements such as boundary layers (Scarano 2003).

PTV generally uses a single pulsed laser to track the displacement of individual particles through frames and compute a velocity measurement for each tracer particle. By tracking individual particles, PTV methods can, therefore, resolve flow fields at higher spatial resolution. Kähler et al., 2012 compared the spatial resolution limits of PIV and PTV methods for a localized step function flow gradient. Dynamic spatial range (DSR) is used to characterize the spatial resolutions for evaluating both PIV and PTV methods. The study found that PTV had a DSR nearly two and a half orders of magnitude greater than PIV measurements using 8x8 pixel interrogation windows and three orders of magnitude greater than PIV measurements that used 16x16 pixel interrogation windows. Due to the improved spatial resolution results using the PTV method, more effort has been made to enhance the development of PTV systems in recent years (Bao and Li, 2011, Doh et al., 2012, Kin et al., 2016, Lатычевская and Fink, 2014).

Raw data is acquired, for a PTV system, using a calibrated camera and synchronized laser, where the intrinsic and extrinsic camera parameters are known. Unprocessed data takes the form of an image stack where each image is separated by a known increment of time and the images contain illuminated particles. Post-processing is

a multi-step process that includes a particle centroid finding algorithm to determine the location of particles in each image, and a particle linking algorithm which links particles between consecutive frames. The velocity of an individual particle is then calculated by dividing the displacement of the centroid between frames by the known time increment between frames.

2.1.2. PTV Sources of Error

PTV algorithms use images to measure particle velocity. A PTV image is a bitmap of pixels where each pixel has an assigned intensity. For grayscale images, pixel intensity ranges from 0 (black) to 255 (white). When using particle centroid locations in an image plane, it is necessary to use a physical length scale to calculate velocity components. The following paragraphs discuss three main sources of PTV system errors - camera error, position uncertainty error, and particle overlap error.

Digital cameras are manufactured with a variety of different properties, including frame rate, focal length, sensor size, pixel resolution, and magnification. A camera calibration process is required to measure intrinsic parameters of a camera so that accurate measurements can be made from pixel positions. Typically, a camera calibration process selects points on a calibration plate or checkered board with known physical distances between points to calculate the intrinsic camera properties. If an image with low resolution is used, or the calibration selection points aren't precisely selected, the intrinsic camera properties will be incorrect, leading to calculation errors of physical centroid positions for particles in an image.

Many different types of cameras are used in PTV setups for different purposes. Frame rate is a critical property that must be selected to allow enough temporal resolution to resolve desired flow structures in the image flow field. If the frame rate is too low for a given flow velocity, particles will move large distances across the image between frames, making it difficult to link particles. In other cases, particles may only appear in a single frame making them impossible to link between consecutive frames. A frame rate that is too high compared to the required temporal resolution may increase computational cost by creating massive amounts of data storage and processing.

Hain et al. (2007) performed a detailed comparison of charged-coupled device (CCD) and active-pixel sensor (CMOS) camera sensors. CCD camera pixel sizes are nearly four times smaller than the pixels on the CMOS image sensor; however, the frame rate of the CCD is 8 Hz compared to the frame rate of the CMOS at 1024 Hz. The study showed that CCD cameras had the highest precision for spatial measurements while the CMOS is optimal for time-resolved data acquisition.

Images are bitmaps of pixels with assigned intensities. Precision for centroid finding is key to computing the particle location. Each particle appears in an image as a cluster of bright pixels. Depending on physical particle size and distance from the camera, a particle may only appear as a relatively small group of pixels requiring sub-pixel centroid finding precision. Figure 4 shows how a single particle may appear in a small image section.

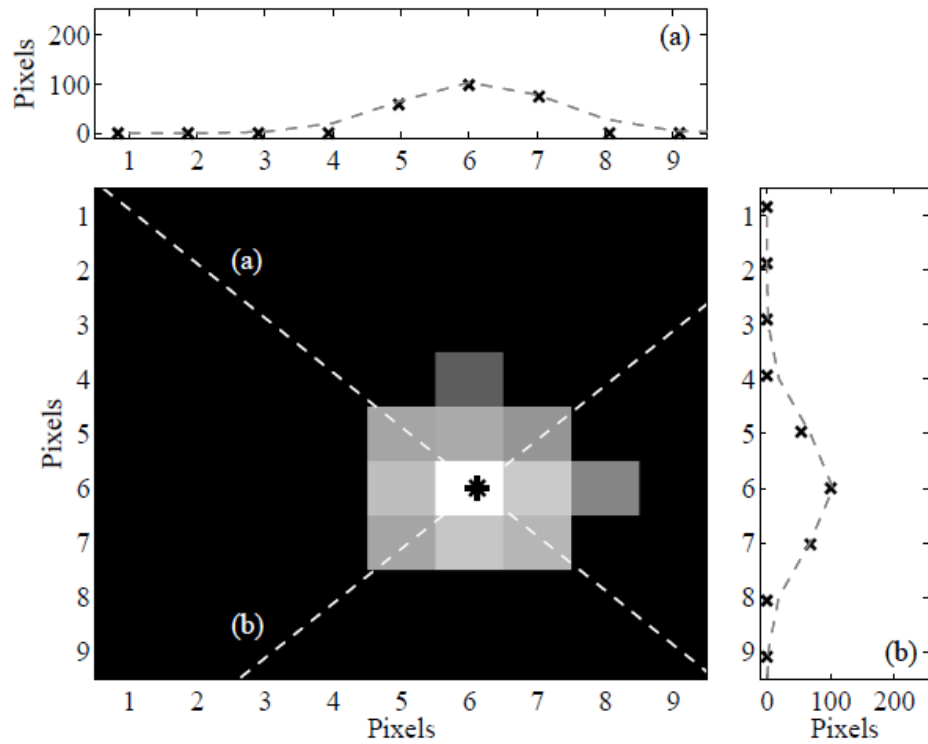


Figure 4: Determination of particle centroid method for subpixel accuracy. (a) shows pixel intensities in the x-direction, (b) shows pixel intensities in the y-direction (Gollin et al., 2017).

Feng et al., (2011) studied the effect of centroid uncertainty on PTV velocity results and found that velocity measurement errors are dominated by centroid uncertainty when the particle displacement between frames is small. They concluded that fast frame rates may not always be optimal in cases where particle displacement is negligible because velocity uncertainty is increased by centroid uncertainty with small particle displacements.

Two classes of ghost particles contribute to errors in PTV systems. The first class is reconstruction ghost particles which occur in 3D PTV methods. Reconstruction ghost particles occur when particle intensities intersect from multiple camera images: they are further discussed in Section 2.2.1. Clustered ghost particles occur when multiple particles

in an image appear as one cluster. In 2D PTV, a thin laser sheet is used. When particles pass by each other, they become indistinguishable within an image. PTV systems often identify the cluster as a single particle even though it is likely multiple overlapping particles. Centroid-finding algorithms cannot find the correct particle centroid if multiple particles are clustered together in an image. Not only do ghost particles have faulty centroids but may disrupt linking algorithms for the individual particle tracks of the clustered particles. The negative effect of ghost particles is minimized by seeding tracer particles at a lower density in the image. In some cases, the particle seeding density is not controllable; for these cases attempts have been made to identify overlapping particles. Methods to overcome this problem include anisotropic thresholding to find discontinuities within blobs (Maas et al., 1993), finding the inflection point between intensities within blobs to identify multiple particles (Mikheev and Zubtsov, 2008), neural networks for identifying overlapped particle images (Carosone et al., 1995), and particle mask correlation which correlates intensity to the expected shape of individual particles (Takehara and Etoh, 1999).

Sources of error in PTV systems vary depending on the algorithms used to identify and link particles. Optimal conditions for PTV exist when the image signal-to-noise ratio is high. Signal-to-noise ratio can be improved by reducing background noise and lowering the tracer particle seeding density.

2.2. Volumetric PTV Methods

The ability to track particles in three-dimensional space is critical for scientists to understand 3D flow structures. In a simple image containing particles spread throughout a

volume, there is insufficient information to measure particle spatial location because the distance from each particle to the camera is unknown. Various techniques have been developed to track the spatial location of a particle in three-dimensional space, and many instances exist where 3D-PTV methods have been used to study flow characteristics (e.g., Kim et al., 2016; Romano et al., 2021). 3D-PTV methods commonly use multi-camera configurations, image splitters, or structured light to calculate the depth of each particle in an image. A review of 3D-PTV methods follows in the remainder of Section 2.2.

2.2.1. Tomographic PTV

Tomographic PTV (Tomo-PTV) systems consist of 3 or 4 CCD cameras positioned at different angles of view around a laser illuminated test section. Instead of identifying particles in a 2D image plane, Tomo-PTV uses numerical pixel intensity data from the images of multiple cameras to construct a 3D voxel space using a tomographic approach (Doh et al., 2012). A simple illustration of the approach is shown in Figure 5, where two image planes are shown whose pixel projections in z_c intersect to form a 3D voxel space. Assembling a 3D voxel space requires an extensive calibration process.

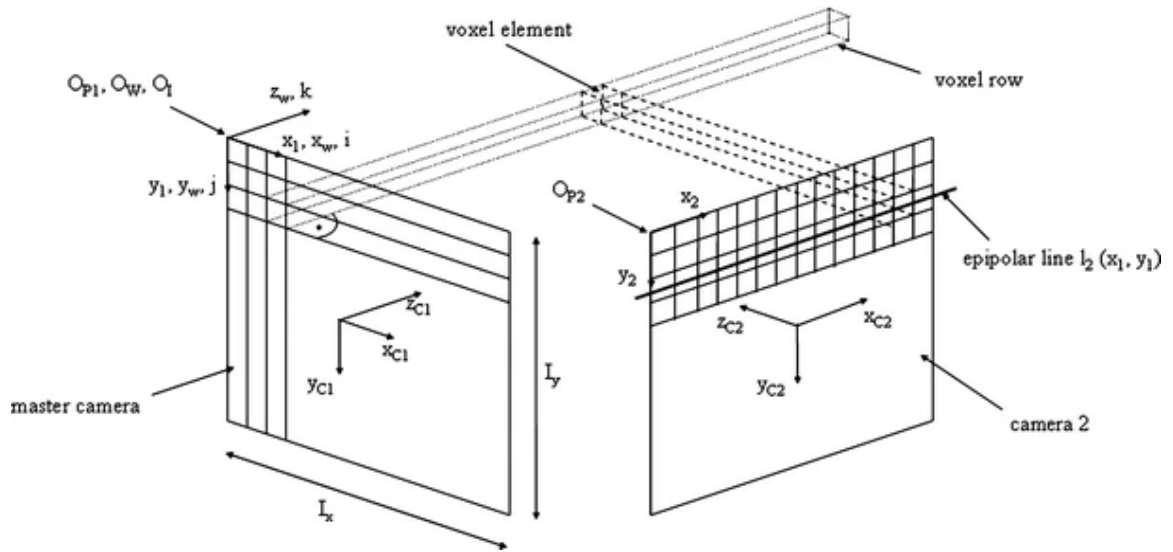


Figure 5: Pixel projections orthogonal to image planes form a 3D voxel (Kitzhofer and Brücker, 2010).

An image is a 2D bitmap of pixels where each pixel has a numerical intensity value. A particle intersected by a pixel row appears as a 1D Gaussian distribution of pixel intensities. The 3D position of a particle is identified at the intersection of particle projections from multiple cameras as a cluster of high intensity voxels. Solving for the centroid of a particle in voxel space requires at least three 1D Gaussian intensity profiles. The Multiplicative Algebraic Reconstruction Technique (MART) is the primary algorithm used to construct the 3D space (Elsinga et al., 2006). The MART method is an iterative method that uses a weighting matrix to relate pixel intensities from each camera to voxel intensities. Those intensities are projected onto a synthetic camera sensor, which is then compared to original images, and the process reiterates until the back-projected images match the original images. The method works well for a high particle seed density, but is computationally expensive; however, some improvements have been made to reduce the

computational cost (Atkinson and Soria, 2009). Once particles are constructed in 3D space, they are then tracked from frame-to-frame creating a velocity vector field.

Tomographic reconstruction is a challenging process, often leading to the appearance of ghost particles in voxel space. Ghost particles occur when particles are reconstructed in voxel space when there is no particle actually at that position. Ghost particles occur when the intensities of two different particles intersect in voxel space as shown by Figure 6.

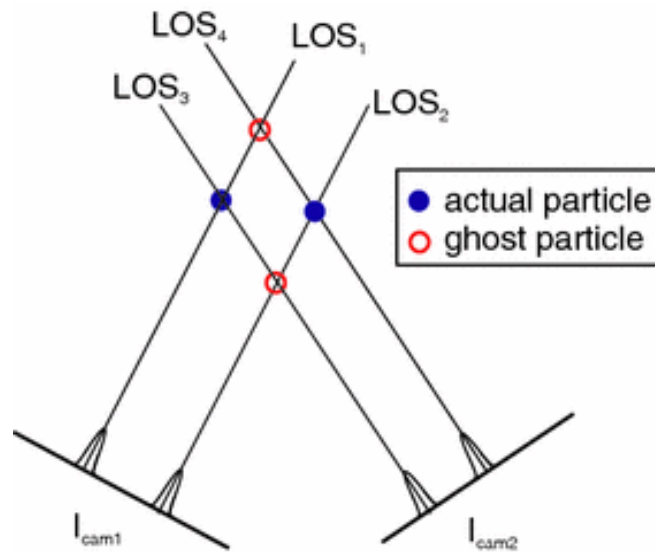


Figure 6: Ghost particles occur at positions where intensities of two different particles overlap (Elsinga et al., 2011).

Although a PTV algorithm identifies a particle as existing at the location of a ghost particle, there is no particle there in physical space. Ghost problems may interfere with linking and matching algorithms and are often hard to distinguish from real particles by the PTV system. Adding more cameras to the setup or decreasing particle seed density can help to control this problem. The main source of error for Tomo-PTV are ghost particles and

calibration errors. A two-camera Tomo-PTV configuration is insufficient, so generally 3-5 cameras, rotated at angles near 20-30 degrees, are used.

2.2.2. Defocus PTV

Defocusing PTV is a single camera particle tracking technique utilizing radial blur or image shift of particle in the image plan to determine particle depth. A traditional camera uses a single aperture opening to allow light to pass through a lens and land on an image sensor, as shown in Figure 7a. Defocusing PTV uses depth from a defocusing process, shown in Figure 7b, using a camera containing multiple aperture openings so that each particle in 3D space appears more than once on the image plane. Depth is then computed using the image shift, b , between the particle images of the same particle on the image plane.

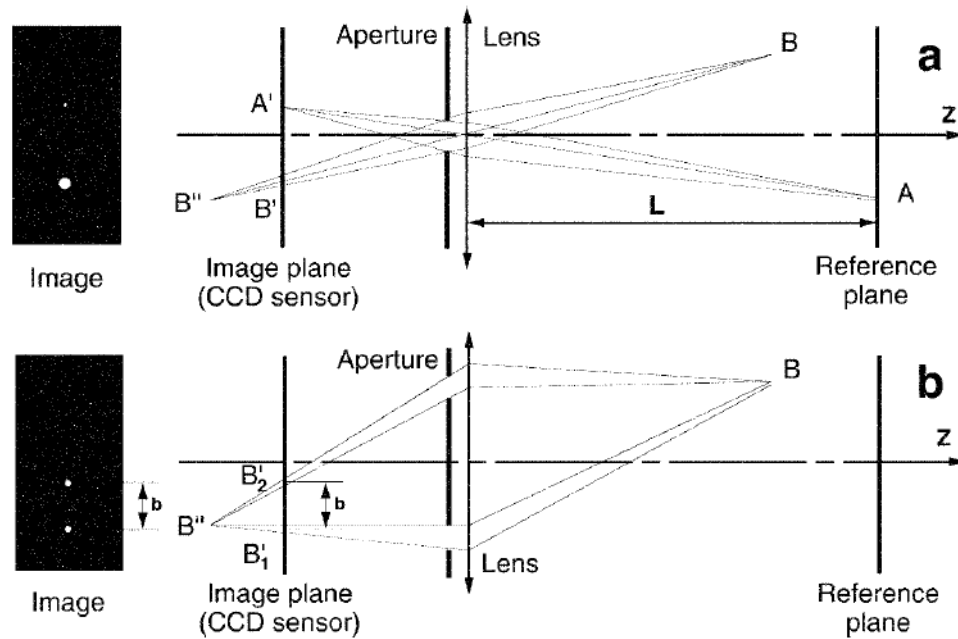


Figure 7: a. Standard imaging system; b. defocusing arrangement (Pereira et al., 2000).

Wilbert and Gharib (1992) used an aperture mask with three pin holes to decode three-dimensional positions of illuminated particles tracked throughout time to obtain particle trajectories in a flow field of a vortex ring impinging on a wall. Pereira et al. (2000) computed depth from defocusing with a PIV algorithm to map the velocity field around a propeller in a volume of one cubic foot using bubbles as tracer particles.

Other methods of defocusing PTV using cameras with traditional single-aperture openings are suited primarily for microscopic PTV (Park and Kihm 2006, Guo et al., 2019). These methods compare the radial blur or shape of a particle in the image plane to shapes or radial blur of particles at known distances. For instance, Particle A in Figure 7a lies on the reference plane and appears as a point source of light on the image plane; however, particle B in Figure 7a lies outside the reference frame, causing the radial projection on the image plane to appear larger and lower intensity. Barnkob et al. (2015) developed the General Defocus Particle Tracking (GDPT) method which compares the shape of a target particle in the image plane to a stack of calibration particle images at known depths, as shown by Figure 8.

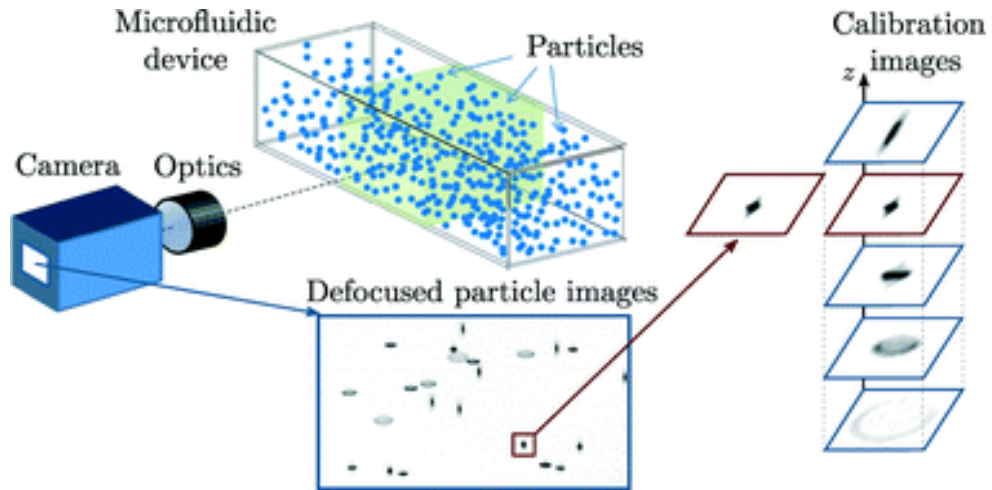


Figure 8: A target particle image in the image plane is compared to a set of calibration images with known out of plane depth, z (Barnkob et al., 2015).

The GDPT method successfully measures positions of $2\ \mu\text{m}$ particles with uncertainty of $0.1\ \mu\text{m}$ in the in-plane direction and $2\ \mu\text{m}$ in the depth direction in a measurement volume of $1510 \times 1270 \times 160\ \mu\text{m}^3$. This method requires uniform seed particles; however, it is capable of distinguishing overlapped particles due to the difference in appearance of particle image shapes in the image plane.

2.2.3. Photogrammetric PTV

Photogrammetry is the process of extracting 3D information from at least two overlapping images. Photogrammetric 3D-PTV employs multiple camera configurations to image particles inside a volume typically illuminated by a laser. Instead of building a voxel space, as with tomographic methods, photogrammetric PTV uses particle detection methods to find particle centroid locations in images, and then uses a matching algorithm to match like particles in each image to find instantaneous 3D particle positions. Particles

are then linked through time to calculate individual velocity vectors for each particle in the test volume.

Papantoniou and Dracos (1990) were the first to use photogrammetric analysis for PTV to compute flow characteristics of turbulent open channel flow using three synchronized cameras mounted in a stereoscopic arrangement with axes intersecting at 50 degrees. Many applications of photogrammetric PTV methods have since been developed using different numbers and types of cameras depending on the application. Two camera systems can calculate 3D spatial dimensions at low tracer particle density. At high tracer particle densities, particle images overlap to form clumped ghost particles, challenging to distinguish with only two cameras. Generally, for laboratory PTV experiments more than two cameras are preferable to improve 3D spatial resolution. Graff and Gharib (2008) studied error in the constructed scene and ghost particle generations for multiple configurations of three and four camera PTV systems and found a lower error factor for a four-camera system than a three-camera system.

Stereo PTV uses two cameras separated by a baseline distance with overlapping fields of view. The process involves a calibration procedure to determine 3D particle positions. Each camera is calibrated to extract intrinsic and extrinsic (rotation and translation) parameters used to form a 3x3 fundamental matrix relating a point in the left image, x , to an epipolar line in the right image on which a corresponding particle, x' , lies. In Figure 9, point X in 3D space, appears in the image plane of the left camera, C_1 , as x and appears in the image plane of the right camera, C_2 , as x' . The intersection of the right image

plane and the plane created by points C_1 , C_2 , and X in Figure 9 creates an epipolar line l' on which x' must lie.

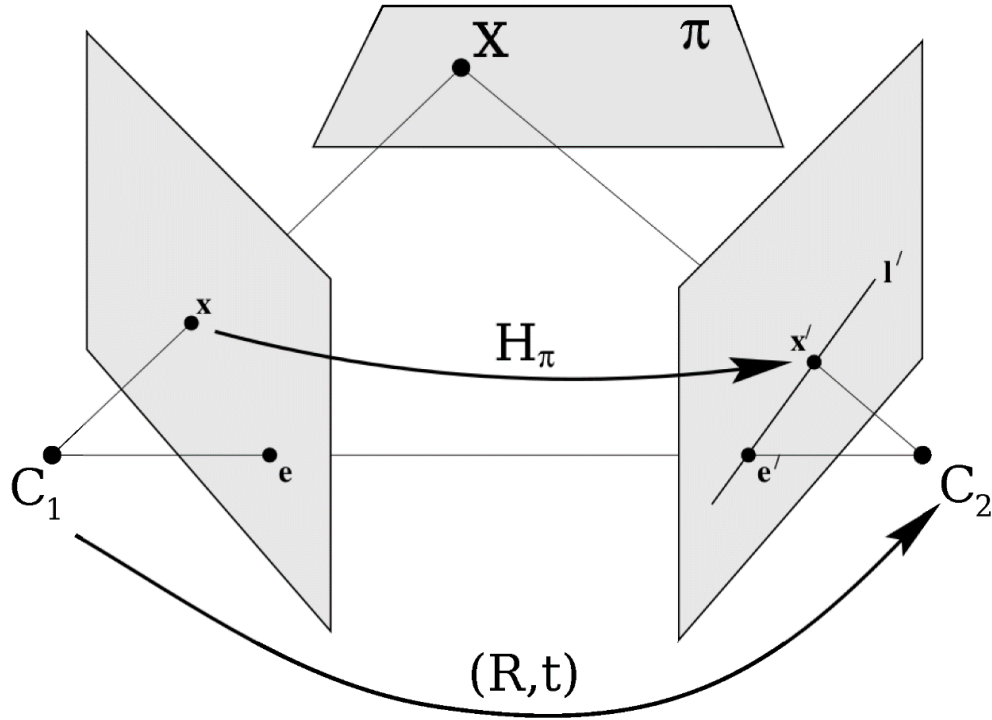


Figure 9: Epipolar geometry of a stereo camera configuration (Lopez-Martinez and Cuevas, 2020).

Epipolar lines can be used to determine particle matches between left and right frames (Lindken et al., 2008). It is possible that more than one particle may lie on the same epipolar line in a stereo configuration. Bao and Li (2011) proposed to use the radial blur of a particle image as an additional constraint if multiple particles passed the epipolar constraint. Their method used the radial blur from defocusing to estimate a range of depth for particle x in the left image so that the epipolar line can be snipped to a short segment in the right image. The corresponding particle match in the right image, x' , is only selected if it is within the depth tolerance calculated using radial blur from particle x . This method increased the correct pairing rate from 56.1% to 74.3% for a simulation using 2200

particles at a density of 0.0011 particles per pixel when compared to the epipolar nearest-neighbor method. Peterson et al. (2012) advanced the epipolar matching constraint by adding a constraint on image similarity defined by seven criteria: 1) peak intensity 2) summed intensity 3) total number of pixels 4) width, as measured in pixels 5) height, as measured in pixels 6) maximum value of a cross-correlation between zero-padded raw particle images 7) maximum value of a normalized cross-correlation between zero-padded binarized particle images. If multiple particles passed the epipolar constraint, then the particle pair with highest image similarity is considered a match.

2.2.4. Holographic PTV

A hologram is an apparent light structure formed by the interference between light diffracted from an object illuminated by a coherent light and a known reference beam. Holography is an imaging technique that uses light wave interference to reconstruct 3D positions of small objects, such as particles. Holograms form from a diffraction between a coherent light source, known as the object beam, and a known reference beam. Hologram images contain information on the phase and amplitude of diffracted waves making 3D position reconstruction possible using optical or computational techniques. There are generally two types of holographic reconstruction, off-axis and inline. With off-axis holography, the object beam and reference beams are at different angles, as seen in Figure 10a and in the corresponding 3D reconstruction of the virtual image is shown in Figure 10b. In-line holography, shown in Figure 10c, exists when the object and reference beams are parallel. In in-line reconstruction, real and virtual particle images are symmetrical, as shown in Figure 10d.

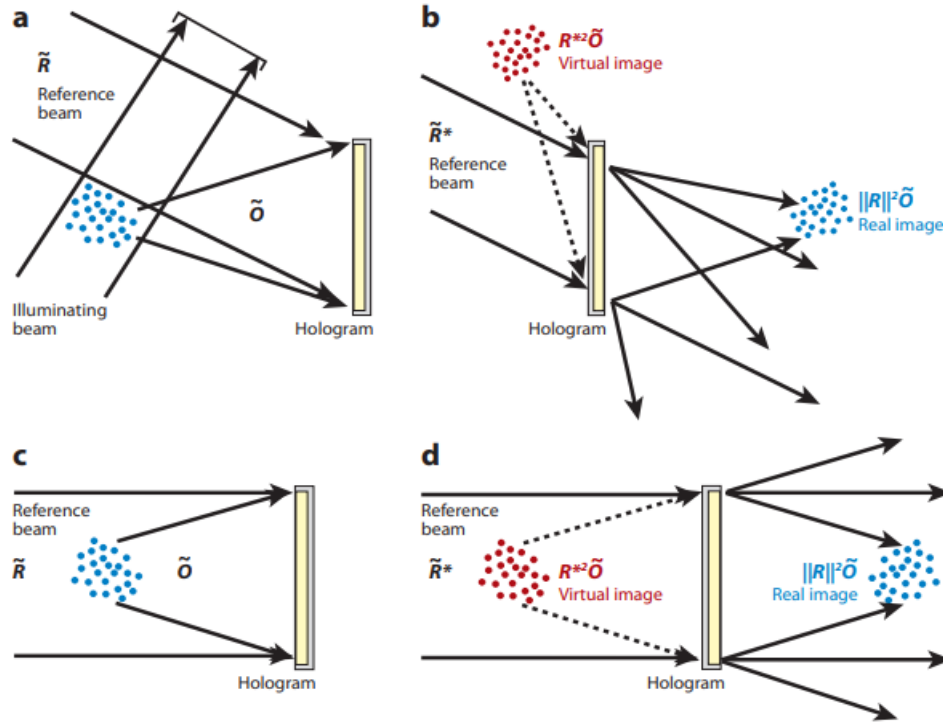


Figure 10: a) Off-axis light configuration and (b) off-axis virtual and real image reconstruction. (c) Lighting configuration for in-line holography and (d) the real and virtual image reconstruction (Katz and Sheng 2010).

In-line holography is typically used with PTV methods because the configuration is simple and requires only 0.1% of the light than an off-axis configuration (Toloui et al., 2017). However, off-axis configurations provide better depth resolution and have fewer overlapped holograms compared to in-line holography.

Figure 11 illustrates the experimental configuration for a holographic PTV system. In this case a coherent laser light source travels through a volume of flowing particles. Particle holograms, that are in focus, appear as points in an image, while holograms of particles that are not in focus appear to have large radius fringe patterns.

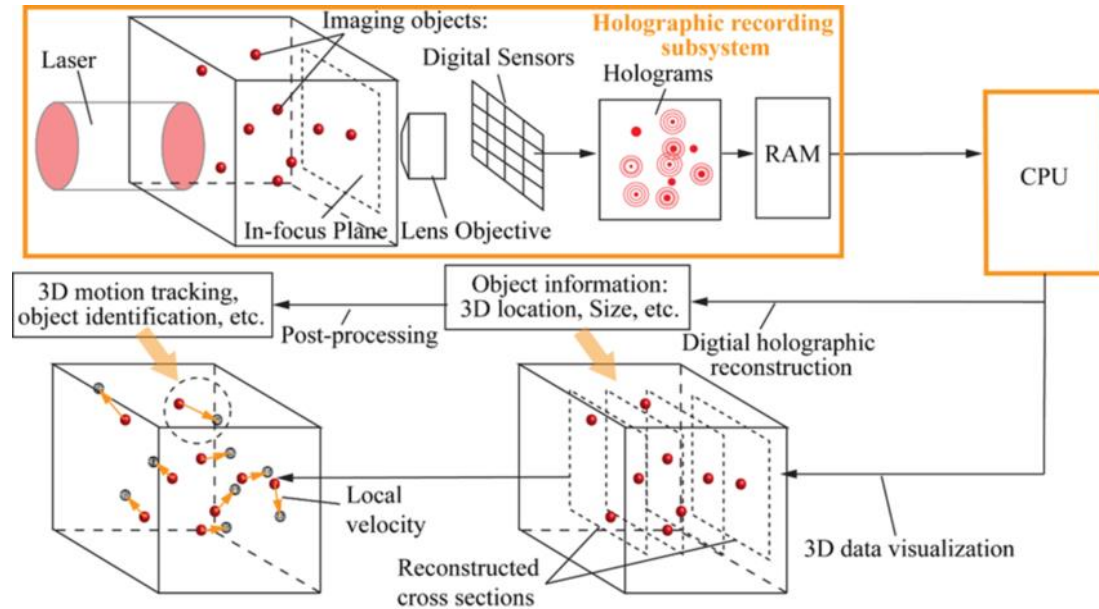


Figure 11: Schematic drawing of in-line holographic PTV system (Coletti et al., 2016).

Particles that are in focus exist at a known distance from the camera. Reconstruction methods then use the diffraction patterns of holograms to compute particle depth in voxel space. Holographic PTV methods have low spatial resolution in the depth direction because reconstruction methods elongate particles in voxel space (Adrian and Westerweel, 2011). Methods to reduce the elongation effect include digital deconvolution (Sarder and Nehorai, 2006; Latychevskaia et al., 2010; Latychevskaia and Fink, 2014) and inverse iterative prediction (Sun et al., 2020). Once particle positions are computed using hologram images, the same linking methods can be used as with other 3D PTV methods to track particles through time and compute velocity.

Satake et al. (2008) studied micro-channel flow using an in-line holographic PTV technique. To accomplish this, nylon particles were placed on a glass slide that traversed parallel to the camera image plane and depth from the camera was adjusted. The study

found that in-plane velocity components had an uncertainty average of roughly 1.6% while out of plane velocity uncertainty was 11.03%. Due to poor spatial resolution in the depth direction, stereo PTV methods are generally preferred over holographic PTV methods.

2.3. Depth Estimation in Computer Visualization

Image depth reconstruction is an important challenge for many different applications, including autonomous driving, robotics, and 3D reconstruction in augmented reality systems. The process by which a camera produces an image, when light travels through a small opening and hits an image sensor, is strikingly similar to the biological process of sight, when light passing through the cornea of an eye forms an image as the light hits the retina. This leads to the question; how do humans perceive depth from images formed by their eyes? The answer is through a combination of two different types of processing techniques -- *monocular depth cues*, such as texture, gradient, motion parallax and occlusion, and *binocular depth cues*, such as stereopsis. Monocular depth cues are object features the brain uses to determine object depth order from a single image. Binocular depth cues are formed from a combination of multiple images combined to perceive depth. Humans see with two eyes, and the brain combines the two distinct images to form a single image. Subtle differences in object positions between the two images are caused by the slight translation of the right eye to the left eye. The human brain uses these differences to calculate depth using a process called convergence (Kooi and Toet, 2004). The field of computer depth estimation seeks to mimic these biological processes related to sight to estimate depth within two-dimensional images. Section 2.3.1 discusses computer visualization techniques using monocular vision methods to map image depth. Section

2.3.2 reviews computer visualization methods developed to map depth using multiple cameras.

2.3.1. Single Camera Depth Estimation

Monocular depth perception cues involve the use of a single eye, or in the case of computer vision, a single camera. Monocular cues include size, interposition, lighting, texture, blur, and motion parallax. Interposition occurs when an object that is closer partially obstructs the view of an object at a greater depth. Motion parallax is a phenomenon that occurs when objects at different depths move different distances within an image. For example, a viewer standing in the middle of a snowstorm perceives nearer snowflakes to fall faster than snowflakes that are farther away, which appear to be falling relatively slower. The brain uses a combination of monocular cues to perceive depth (Landy et al., 1995; Mather and Smith, 2004; Ono and Wade, 2006). The following subsection provides a review of methods utilizing monocular cameras to measure depth from within an image.

It is important to note that monocular cues provide depth information relative to objects in a scene. Therefore, it is necessary to have a known physical length scale in a single image to compute real-world depth. Salih and Malik (2012) presented a method to compute object size and depth using simple geometric relationships between the points in an image compared to known camera position and angle. The method is intended for use where a camera is oriented at a given angle and height above the ground. The depth of an object on the ground can then be measured using the cameras' field of view in a pinhole camera model. Other models use lighting as a monocular cue to compute object depth by relating pixel intensities through a time series of images (Maki et al., 1998) or by using a

Hidden Markov Model (Nagai et al., 2002). Saxena et al. (2005) uses a combination of texture variations, texture gradients, and haze to compute an absolute depth for an image patch, which is then compared to other image patches resulting in a relative depth map for a single image. Often monocular cues, such as blur and motion parallax, do not provide enough information to compute depth maps on their own but are used as complimentary information to enhance the resolution of multi-camera depth maps (Kellnhofer et al., 2016, Kowdle et al., 2012).

Progress in machine learning in recent years has led to successful neural network algorithms for computing depth from a single camera. Fu et al. (2018) introduced a spacing-increasing discretization strategy to train a convolutional neural network (CNN) using an ordinary regression loss function as opposed to a conventional mean squared error loss function. This training method allows for fewer pooling operations resulting in a higher resolution depth map. Alhashim and Wonka (2019) utilized a transfer learning technique which used deep features from convolutional neural networks (CNN) previously trained on large datasets. They proposed that with transfer learning, their CNN can produce highly resolved depth maps. Figure 12 provides an example of depth maps from the image in the first column, with ground truth 3D laser scan in the second column, and Alhashim and Wonka (2019) transfer learning method in the fourth column.

Although depth estimation from single images has been improved by the enhanced capabilities of machine learning, single image methods are still significantly less accurate than multi camera systems. Many depth estimation methods using monocular cues also suffer from being image-dependent, in that what might work well for a recognizable

image, such as a road or train tracks extending into the distance or an image with shadows and light gradients, would not be expected to work well for an image consisting of a set of white dots (as is the case for PTV). The following section details how multi-camera depth perception can be used to provide additional information on image depth.

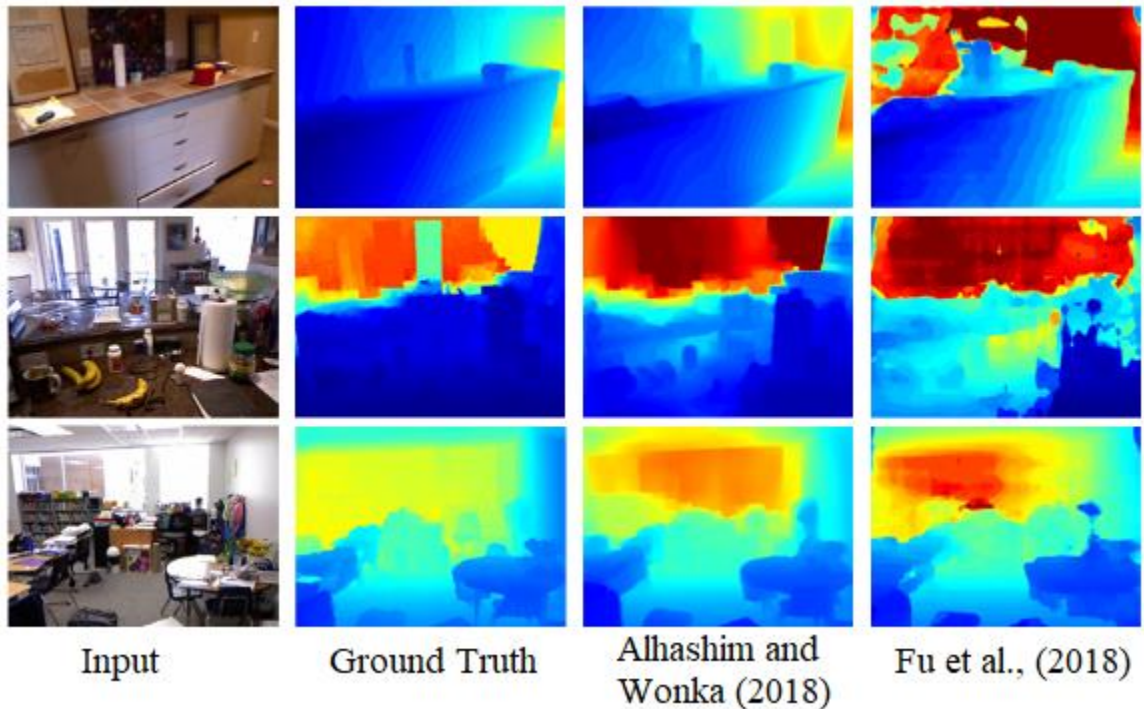


Figure 12: Input image, ground truth laser scan depth map, transfer learning depth map, and regression loss CNN (Alshahim and Wonka, 2018).

2.3.2. Multiple Camera Depth Estimation

The human brain perceives depth by forming a single image from the images of two individual eyes, a process known as stereopsis. Two eyes converge at different angles when focusing on objects at different distances. Cameras can be used in the same way to construct a 3D scene from similar points in images from cameras with known positions, a

technique which is known as disparity mapping. Disparity mapping is commonly used to compute the depth of field in applications including self-driving cars and robotics (Caesar et al., 2020; Geiger et al., 2013).

Disparity mapping involves stereo cameras with overlapping fields of view. Disparity is the distance between two corresponding points in a pair of stereo images, and is inversely proportional to depth. Disparity maps are created by iterating through each pixel in the first stereo image to locate the corresponding pixel, of the same real-world point, in the second image. Algorithms that match corresponding particles between stereo images generally use an epipolar geometry constraint. Epipolar geometry constraints limit the search area for a pixel in the first image to a single line of pixels in the second image. Epipolar geometry is based on the orientation of the stereo cameras and explained in more detail in Section 2.2.3. The epipolar geometry of co-directional stereo cameras is a horizontal line. For example, a particle image located at (X^R, Y^R) in the right image plane must lie on the same row of pixels as the corresponding image of the same particle in the left image, located at (X^L, Y^L) ; thus $Y^R = Y^L$, which significantly reduces the search area for corresponding particle images. Matching algorithms search epipolar geometry lines and use a cost computation function to match corresponding pixels between left and right images (Scharstein and Szeliski, 2002). The cost computation commonly uses *mean-squared intensity error* (Anandan, 1989) and *mean absolute intensity differences* (Kanade, 1994) of pixel intensities in a small window of pixels surrounding a pixel on the epipolar line to locate corresponding matches. After corresponding pixels are matched, the disparity d at that pixel is calculated by $d = X^L - X^R$, from which depth can be computed. Initial

disparity maps often have granulated appearances and require refinement for higher resolution maps. Gu et al. (2008) implemented adaptive disparity refinement that adjusts disparity values based on neighboring disparity values in a “support-window” that passes over the image in a refinement step. The support acts like a convolutional pass, where general regions are smoothed, but remain are blurred. Jung et al. (2015) implemented a single pixel adaptive disparity refinement by applying an adaptive support-weight to each pixel. The support-weight is generated from a combination of color-similarity, spatial distance, and color dissimilarity information from the original disparity map. Figure 13 shows the results, where (a) is the left image of a stereo input image pair, (b) is the ground truth disparity map, (c) are results from Gu et al. (2008) support-window refinement algorithm, and (d) are results from Jung et al. (2015) single pixel support-weight algorithm.

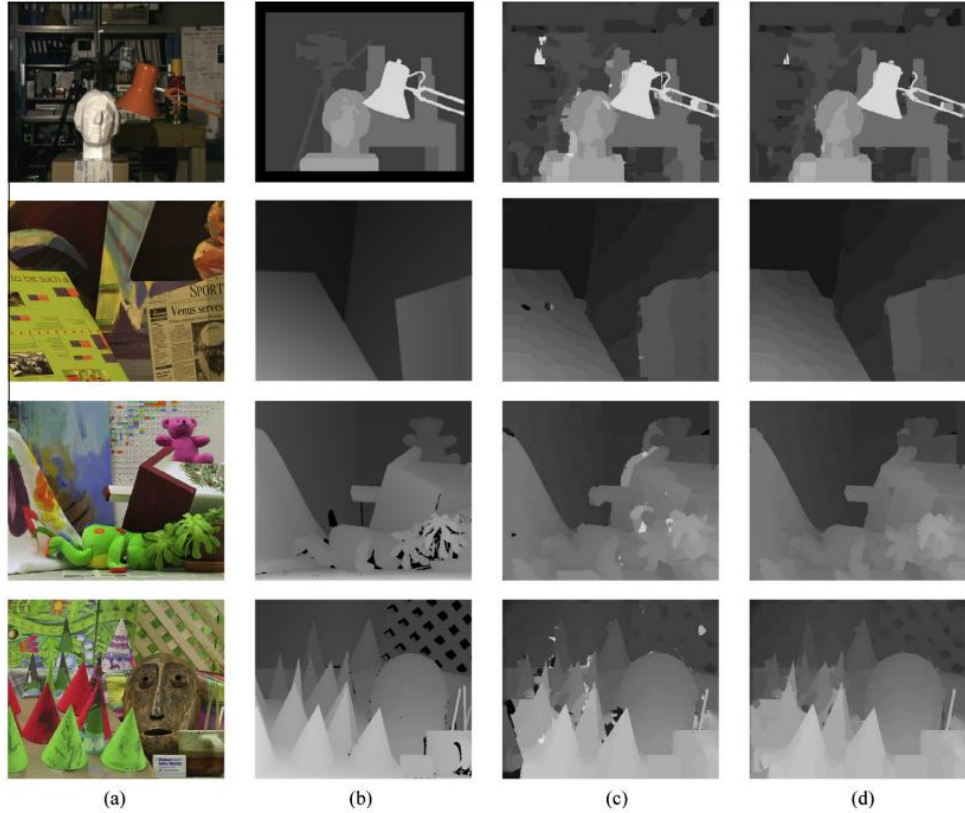


Figure 13: Adaptive disparity mapping results. (a) The original input image, (b) is the ground truth disparity map, (c) results from Gu et al. (2008), (d) results from Jung et al. (2015).

Disparity mapping from stereo cameras is a well-known method for computing depth of a scene. It is important to note that stereo cameras must be synchronized to resolve 3D positions for objects moving in time. The frame rate of stereo cameras must be high enough to resolve the characteristics of desired motion, yet low enough so that massive amounts of excess data is not generated. The resolution of a disparity map can be enhanced using monocular features such as color and spatial similarities (Jung et al., 2015) or motion parallax (Kellnhofer et al., 2016). Stereo correspondence is particularly effective for finding matches of objects that appear as clusters of pixels in an image, such as particles. The process can be simplified by finding corresponding matches for each cluster rather than each pixel.

CHAPTER 3: METHODOLOGY

This section presents a methodology for combining monocular and binocular cues for performing volumetric 3D PTV with a two-camera MR device. It is assumed that both cameras of the MR device are co-directional (front facing), take synchronized grayscale images at a fixed frame rate, are separated by a baseline distance h in the x -direction of a global Cartesian coordinate system, and have focal length f . The field of view of each camera substantially overlaps that of the other camera, with equal focal lengths for both cameras. The origin of the global Cartesian coordinates is located at the point half-way between the cameras, and the z -direction is taken as the distance away from the cameras in the direction of sight (Figure 14).

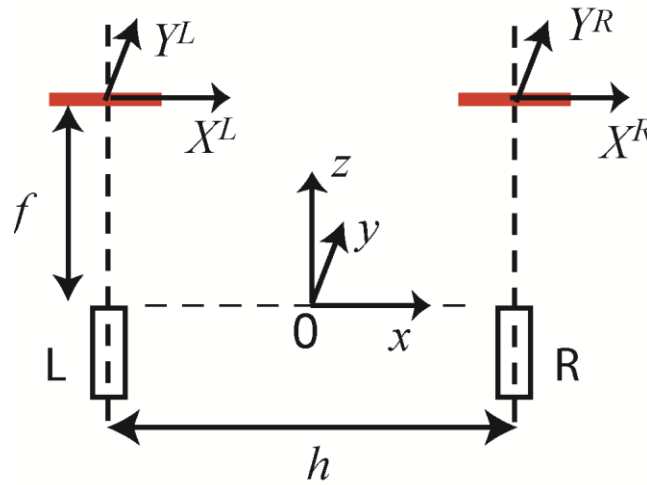


Figure 14: Left (L) and right (R) cameras and the associated global coordinate system (x, y, z) . The image plane is indicated by a red line, with left camera image plane coordinates (X^L, Y^L) and right camera image plane coordinates (X^R, Y^R) .

Image plane coordinates (X^L, Y^L) and (X^R, Y^R) are assigned to the left and right camera, respectively. The origin of each image plane coordinate system is the center of the respective image plane. To help avoid confusion, we use upper-case letters to denote

variables in the image plane and lower-case letters to denote variables in the global coordinate system throughout the paper.

Figure 15 provides a block diagram showing the series of steps required to track particles in 3D space using the two-camera MR device.

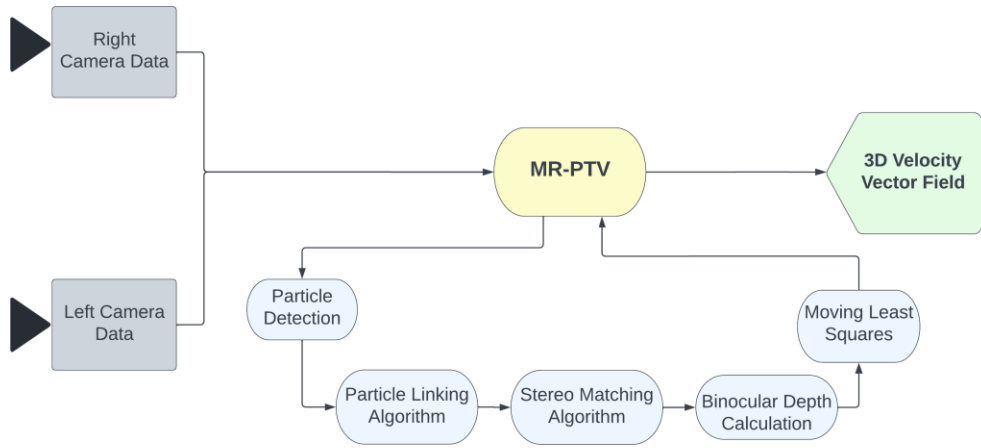


Figure 15: Block diagram of MR-PTV velocity calculation algorithm.

The first step, on the far left of Figure 15, is image acquisition by the two cameras. MR devices typically do not offer adjustable image acquisition rates (as is typical for cross-correlation cameras used for PIV and PTV), limiting the velocity of particle motion that can be acquired with the system. The output of the image acquisition step is two synchronized video feeds, one from each stereo camera. Each video frame is a pixel map taken at a constant time step, which are then processed using the MR-PTV software. The first step of the MR-PTV software involves particle detection in images by using a local adaptive binarization method. The output from the particle detection step includes measurement of image plane variables such as, centroid position (X^L, Y^L) , diameter D^L , and brightness B^L for each particle seen in the image plane of the left camera, with these

measurements repeated for the right camera. The particle linking step employs a dynamic linking algorithm, enhanced by monocular features, to link particles between successive frames. The particle matching step matches particles between left and right camera images. The matching algorithm, again, makes use of monocular cues, along with epipolar geometry constraints to match a particle between left and right stereo images. Binocular disparity is then used to estimate the distance of each particle from the origin. The previous steps result in a series of global position estimates for each particle over a series of frames. To reduce noise in 3D velocity estimates, we utilize the moving least-squares method to fit a polynomial to multiple points which can be differentiated resulting in smoother velocity measurements throughout time.

3.1. Particle Detection

The particle detection method involves an image segmentation technique where particles are distinguished from the background. When a grayscale image is input to the segmentation algorithm a binary image is returned with particles appearing as white and background appearing as black. Global segmentation techniques generally involve a pixel intensity threshold, l_T , that can be set manually or automatically (Sankur 2004) to determine whether each pixel in the image, with pixel intensity l_p , is part of a particle or background.

To reduce high spatial frequency noise, we first pre-process each image using an image convolution with a 5x5 pixel Gaussian kernel. The presented algorithm is intended for use with variable light conditions, therefore, we use an adaptive threshold binarization technique (Sauvola 2000). To determine the threshold, $l_T(X,Y)$, for a pixel, $P(X,Y)$, in an image, we position a square box, with 15 pixel sides, centered on pixel $P(X,Y)$. Next, the

mean pixel value inside the box is calculated. We then subtract a contrasting constant, C , from the mean pixel intensity inside the box, which results in the intensity threshold, $l_T(X, Y)$, for pixel, $P(X, Y)$ with intensity $l_p(X, Y)$, such that

$$\begin{aligned} l_p < l_T & \text{ then } p = \text{black} \\ l_p \geq l_T & \text{ then } p = \text{white.} \end{aligned} \tag{1}$$

The adaptive binarization method accounts for lighting conditions that may be nonuniform throughout the image by detecting particles that contrast the background in smaller subregions of the image. The method requires contrasting pixel intensities between particles and background, though the contrasting constant can be modified to compensate for small or large contrasts between particles and the background.

The segmented binary image is processed using an edge detection contouring method described in Suzuki and Abe (1985), where the output is the set of border points between the binary interface for each individual contour associated with a single particle. We find particle centroids, \bar{X} and \bar{Y} , from these lists by computing spatial moments M_{ij} for each contour as

$$M_{ij} = \sum_{X, Y} X^i Y^j I(X, Y) \tag{2}$$

where X and Y denote pixel location in the image plane, $I(X, Y)$ is the pixel intensity for all pixels within a contour, and the sum is over the pixels within a given contour. We calculate particle locations in the image as the centroids of the moments

$$\{\bar{X}, \bar{Y}\} = \{M_{10}, M_{01}\} / M_{00}. \tag{3}$$

The image-plane brightness B_n of particle n is set equal to the pixel intensity at the contour centroid. The image-plane particle diameter D_n is set equal to the equivalent contour diameter, defined as the diameter of the circle whose area is equal to the contour area.

3.2. Particle Linking

Since the PTV method estimates velocity based on particle displacement in time, it is necessary to track particles between consecutive video frames, a technique we refer to as "particle linking". Traditional planar 2D PTV systems link tracer particles undergoing Brownian diffusion between frames by a nearest neighbor search method (Crocker and Grier 1996), where a search radius is specified that should be larger than the largest displacement of a particle between frames but smaller than the separation between two neighboring particles in an image. When particles move at a larger displacement between frames than the scale of separation, it can be very difficult to link particles between frames. Two ways to handle this problem are by increasing the framerate of the camera so that particles are displaced a smaller amount between frames and by decreasing the tracer particle density so particles are more dispersed throughout the image. The MR-PTV system is designed to work with stereo cameras at a fixed framerate using seed particles whose density cannot be controlled. To meet these specifications, we take advantage of the fact that in 3D PTV systems, the particles in the image plane appear to have differences in their size and brightness, even when the physical particle size and field illumination are constant. These differences are examples of "monocular cues" for particle depth, which are used to distinguish particles between frames.

The monocular cues used for particle linking include the apparent particle diameter D_n and particle brightness B_n measured in the image plane. In general, velocity cannot be used as a monocular cue for the linking step because velocity cannot be calculated until after a particle has been linked between consecutive frames. We implement a predictive linking algorithm to account for effects of particle motion by developing an estimate $\{\tilde{X}(t_{n+1}), \tilde{Y}(t_{n+1})\}$ for the particle position in the image plane at time t_{n+1} based on the positions at the previous two time-steps as

$$\tilde{X}(t_{n+1}) = 2X(t_n) - X(t_{n-1}), \quad \tilde{Y}(t_{n+1}) = 2Y(t_n) - Y(t_{n-1}) \quad (4)$$

where $\{X(t_n), Y(t_n)\}$ denotes the particle position in the image plane at the current time t_n . Rather than simply linking to the particle that lies nearest to the estimated position, we evaluate the group of particles near to the estimated position using a set of monocular cues. The process used for linking is illustrated in Figure 16 for a sample particle denoted by 'c'.

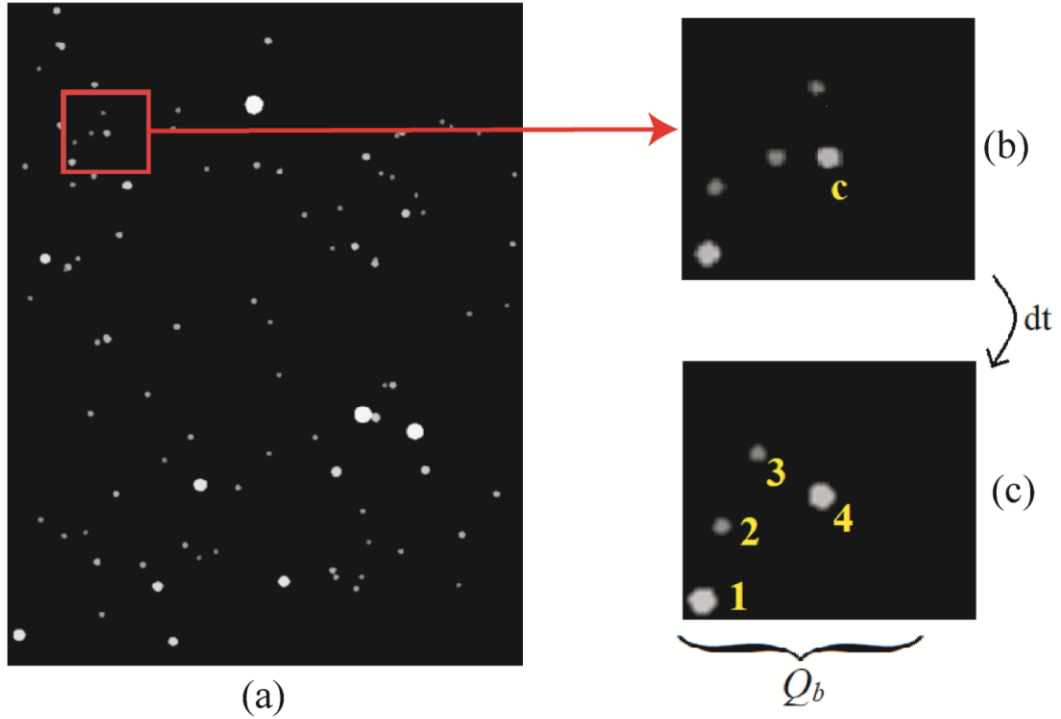


Figure 16: A sequence of images illustrating the particle linking algorithm. (a) A field of particles in the image plane of either the left or right camera at time t_n . The hard-filter region surrounding a particle c is indicated by a red box. (b) A close-up of the particles in the hard-filter region surrounding particle c at t_n . (c) Four particles in the hard-filter box surrounding the estimated position $(\widetilde{X}_c, \widetilde{Y}_c)$ of particle c at time t_{n+1} , which together constitute a set Q_b . Particle c is found to link to particle 4 from this set, which minimizes the particle linking function G_{bc} .

We desire to link particle c , in the current frame at time t_n , to a particle in the successive frame at time t_{n+1} . The first step after determining the estimated particle position from (4) is to identify a set of candidate target particles for linking using a hard-pass filter based on the location of particles in the image. The hard-pass filter is shown in Figure 16 by a box of side length $2H$, shown placed around the current position of particle c at time t_n in Figure 16a and then moved to be around the estimated position of particle c in Figure 16c.

A subset Q_b of particles in the successive frame at time t_{n+1} are considered as possible matches if the particles pass the spatial hard-filter condition, given by

$$x_n^L - K < x_m^R < x_n^L, \quad \left| \tilde{Y}(t_{n+1}) - Y_b(t_{n+1}) \right| \leq H. \quad (5)$$

It is ideal to set H large enough so that all particle links can be captured, but small enough so that relatively small subsets Q_b are considered in order to decrease computational time.

A linked particle is selected as the particle from subset Q_b that minimizes the linking function G_{bc} , defined by

$$G_{bc} = \ell_1 \left(\frac{D_b^L - D_c^L}{D_{ave}} \right)^2 + \ell_2 \left(\frac{B_b^L - B_c^L}{B_{ave}} \right)^2 + \ell_3 \left(\frac{X_b^L - \tilde{X}_c^L}{X_{image}} \right)^2 + \left(\frac{Y_b^L - \tilde{Y}_c^L}{Y_{image}} \right)^2. \quad (6)$$

The particle linking procedure is performed for each camera, although the linking function is shown in (6) for the left camera. The linking function G_{bc} incorporates the monocular cues of apparent diameter and brightness in the image plane, and it is defined for a given camera (denoted by the left camera L in (6)) as a function of the estimated position $(\tilde{X}_c^L, \tilde{Y}_c^L)$ of particle c from (4) and the measured position (X_b^L, Y_b^L) of a particle b from the set Q_b at time t_{n+1} . The coefficients ℓ_1 , ℓ_2 , and ℓ_3 are prescribed weights whose values are selected to try to make each term in (6) of approximately equal magnitude. The normalizing factors D_{ave} and B_{ave} are defined as the mean diameter and brightness in the image plane for all particles in the image. The normalizing factors X_{image} and Y_{image} are defined as half the image width and height as measured in the image plane, respectively. The optimal link for particle c is selected as the particle in the set Q_b that minimizes G_{bc} .

3.3. Stereo Matching Algorithm

Matching particles between two stereo cameras, at each time step, is necessary to triangulate the depth of each particle, which is a procedure that we refer to as 'stereo matching'. The proposed stereo matching algorithm describes how particles are matched between stereo image pairs at each time step. For a particle n at location (X_n^L, Y_n^L) in the image plane of the left camera, we seek to obtain the optimal matching particle m at position (X_m^R, Y_m^R) from the image plane of the right camera. A two-pronged approach is rapidly identifies the matching particle. First, we employ a hard filter to limit consideration of the candidate particle set for matching of particle n from the left camera to only those particles m from the right camera for which

$$X_n^L - K < X_m^R < X_n^L + K, \quad |Y_n^L - Y_m^R| \leq S, \quad (7)$$

where S denotes the half-height in the y -direction. The first of these equations requires the particle position along the X -axis of the image plane of the left camera to be greater than the image in the right camera of a candidate particle match, but less than some width K , as shown in Figure 17.

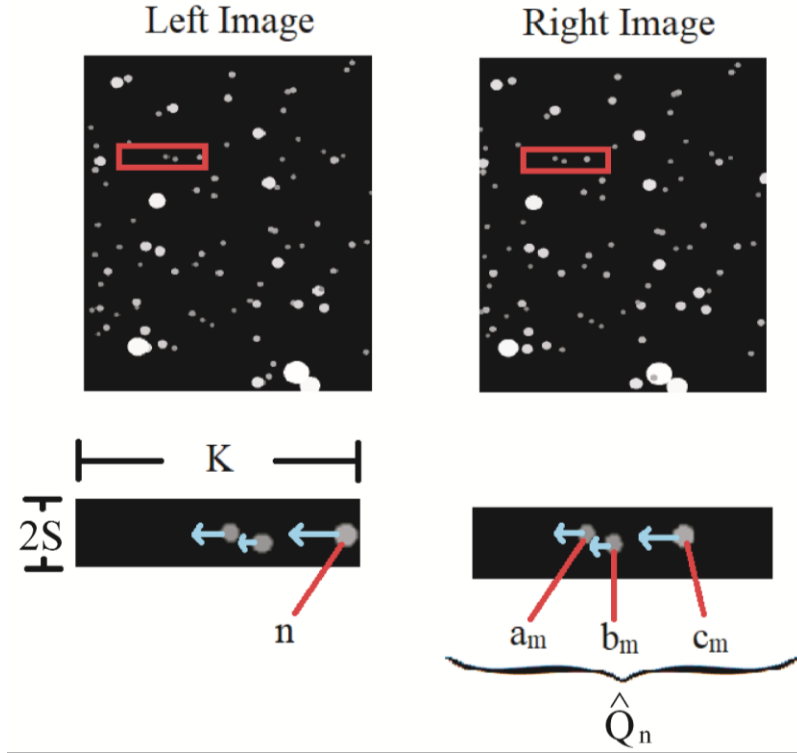


Figure 17: Illustration of the stereo matching algorithm, showing a pair of stereo images from the left and right cameras. In the left image, we identify a particle n and a hard filter search region, indicated by a red box. Particles a_m , b_m , and c_m in the lower right-hand image that fall in the hard filter region constitute a subset \hat{Q}_n , from which the stereo match of n is identified as particle c_m , which minimizes the function F_{mn} defined in Eq. (8).

For current computations, we set K equal to 0.25 times the image width. The second equation in (7) results from the requirement that a matching particle pair should have the same Y -coordinate value based on the epipolar geometry of a co-directional camera configuration, within some range of uncertainty resulting from centroid-finding or time synchronization imperfections. In the current computations, we set S equal to 0.01 times the image height to accommodate a small margin of error. We denote by \hat{Q}_n the set of particles from the right camera that satisfy the hard filter (7) for a given particle n in the

left camera image plane. If no particle is found satisfying the hard filter (7), the particle n is identified as having no match and eliminated from further consideration.

In much the same way that monocular cues were used in Section 3.2 to improve the particle linking algorithm, we employ various monocular cues as well as particle vertical position to determine the optimal selection from the set \hat{Q}_n of right camera particles satisfying the hard filter (7). The monocular cues make use of the fact that particles closer to the cameras appear to move faster and to be both larger and brighter than particles that are farther away from the cameras, even if in reality the particles have the same velocities, diameters and brightness. The diameter, brightness and velocity differences introduced by this distance effect can be used to distinguish between potential particle matches, since these distance-related effects should appear approximately the same when viewed by either camera. Specifically, we define a matching function F_{nm} for a particle n in the left camera image plane and a candidate particle m in the right particle image plane by

$$F_{nm} = c_1 \left(\frac{D_n^L - D_m^R}{D_{ave}} \right)^2 + c_2 \left(\frac{U_n^L - U_m^R}{U_{ave}} \right)^2 + c_3 \left(\frac{B_n^L - B_m^R}{B_{ave}} \right)^2 + \left(\frac{Y_n^L - Y_m^R}{Y_{image}} \right)^2. \quad (8)$$

The velocity U is the apparent velocity magnitude of a particle, measured as the distance a particle moves in the image frame between consecutive frames, divided by the time interval between frames. The X -position is not considered when performing particle stereo matching because a difference in X -position in the image planes of the two cameras is expected to occur as a function of the particle distance from the cameras. Here, c_1 , c_2 and c_3 are prescribed weighting coefficients, which are set so that each term in (8) has equal

order of magnitude. The normalizing factors B_{ave} , D_{ave} , and U_{ave} are the mean values of particle brightness, diameter and velocity magnitude in the image plane for all particles in a set of stereo images. The optimal particle match m from the right camera image plane is selected as the particle from the set \hat{Q}_n that minimizes F_{nm} .

3.4. Depth Estimations with Binocular disparity

A point P has a position (x_n, y_n, z_n) in the global coordinate system, (X_n^L, Y_n^L) in the image plane of the left camera, and (X_m^R, Y_m^R) in the image plane of the right camera, where m is the index of the matching particle from the right camera to the particle with index n from the left camera. The depth z_n of the particle is the same for both cameras. We seek to back out the particle's global coordinates based only on the coordinates measured in the two image planes. As illustrated in Figure 18, a triangle APa, is formed from center A of the left camera lens, the center a of the right camera lens, and the point P.

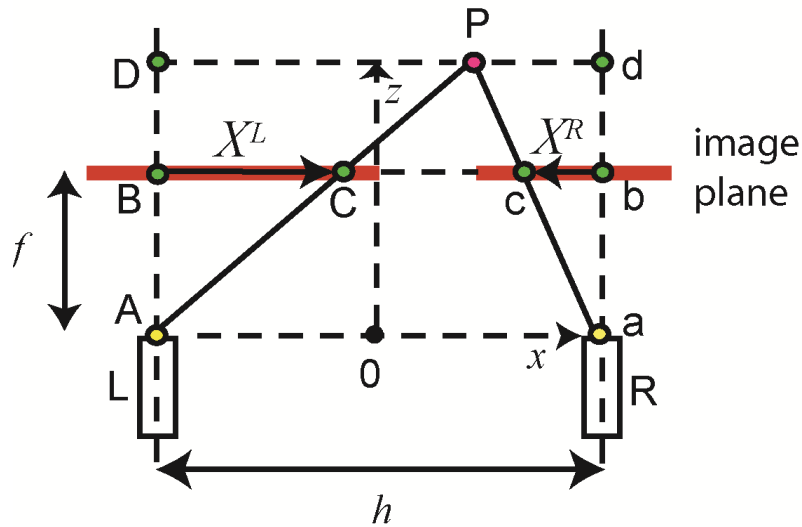


Figure 18: Definition of three sets of similar triangles -- APa and CPc, ABC and ADP, and acb and aPd -- used in distance estimation based on binocular disparity.

A second triangle, CPc, is formed from the particle apparent position a in the left camera image plane, the particle apparent position c in the right camera image plane, and the point P. Since these two triangles are similar, the ratio of base to height is equal, or

$$\frac{h}{z_n} = \frac{h + X_m^R - X_n^L}{z_n - f}. \quad (9)$$

Solving this equation for z_n gives the particle distance from the origin in the global coordinate system as

$$z_n = \frac{f h}{X_n^L - X_m^R}. \quad (10)$$

To calculate the global coordinates, x_n and y_n , for particle n , we note two similar triangles from the left camera denoted in Figure 18 by ABC and ADP, as well as two similar triangles from the right camera denoted in Figure 18 by acb and aPd. These similar triangles can be used to solve for the global particle positions in the x - and y -directions as

$$x_n = -h/2 + z_n X_n^L / f = h/2 + z_n X_m^R / f, \quad y_n = z_n Y_n^L / f = z_n Y_m^R / f. \quad (11)$$

Because we have measurements from each camera for the global coordinates x_n and y_n of each particle, we average the two measurements to reduce uncertainty, giving the global particle positions as

$$x_n = \frac{z_n (X_n^L + X_m^R)}{2f}, \quad y_n = \frac{z_n (Y_n^L + Y_m^R)}{2f}. \quad (12)$$

A test of the effectiveness of the stereo matching and binocular disparity methods for estimating particle depth was performed using 100 particles randomly distributed in a cubic computational domain with side length L . All length scales in the global coordinate system are non-dimensionalized by L in the remainder of the paper. The test was performed

with particles of physical diameter $d = 0.015$ and baseline distance $h = 0.02$, using 480×640 pixel resolution. Figure 19 shows a plot of the estimated particle depth from the origin obtained using the MR-PTV algorithm versus the known computational depth,

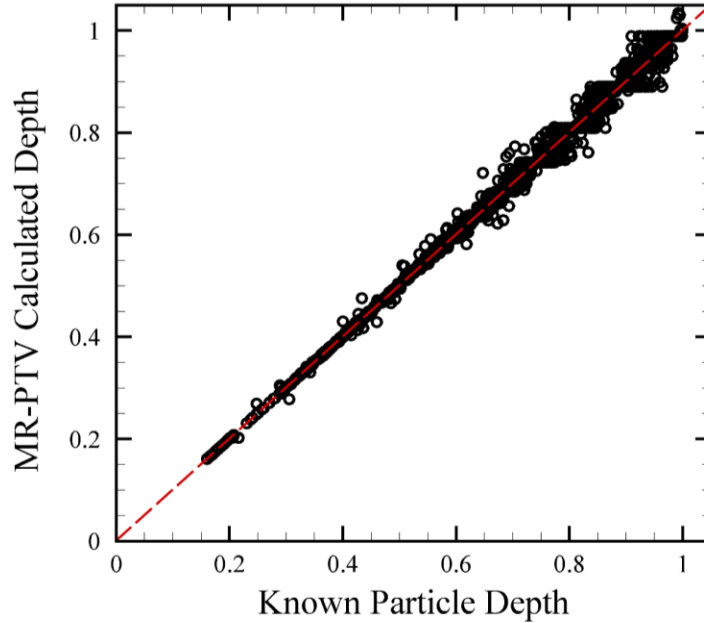


Figure 19: Plot of the predicted particle depth z obtained using the MR-PTV algorithm versus the known particle depth. Each particle is represented by a black circle, and an exact match is indicated by the red line. The MR-PTV predictions were obtained with coefficient values $(c_1, c_2, c_3) = (0.100, 0.036, 0.018)$, set so that each term in the matching function in Eq. (8) has approximately equal weight.

where each scatter point indicates the depth of a particle in the overlapped field of view. We observe that the MR-PTV estimates are grouped around the exact value, indicated by a dashed red line, and scatter points spread further from the known depth as depth increases. To understand the contribution of each of the terms in the equation (8) for the stereo matching function to the depth estimate, refer to Figure 20;

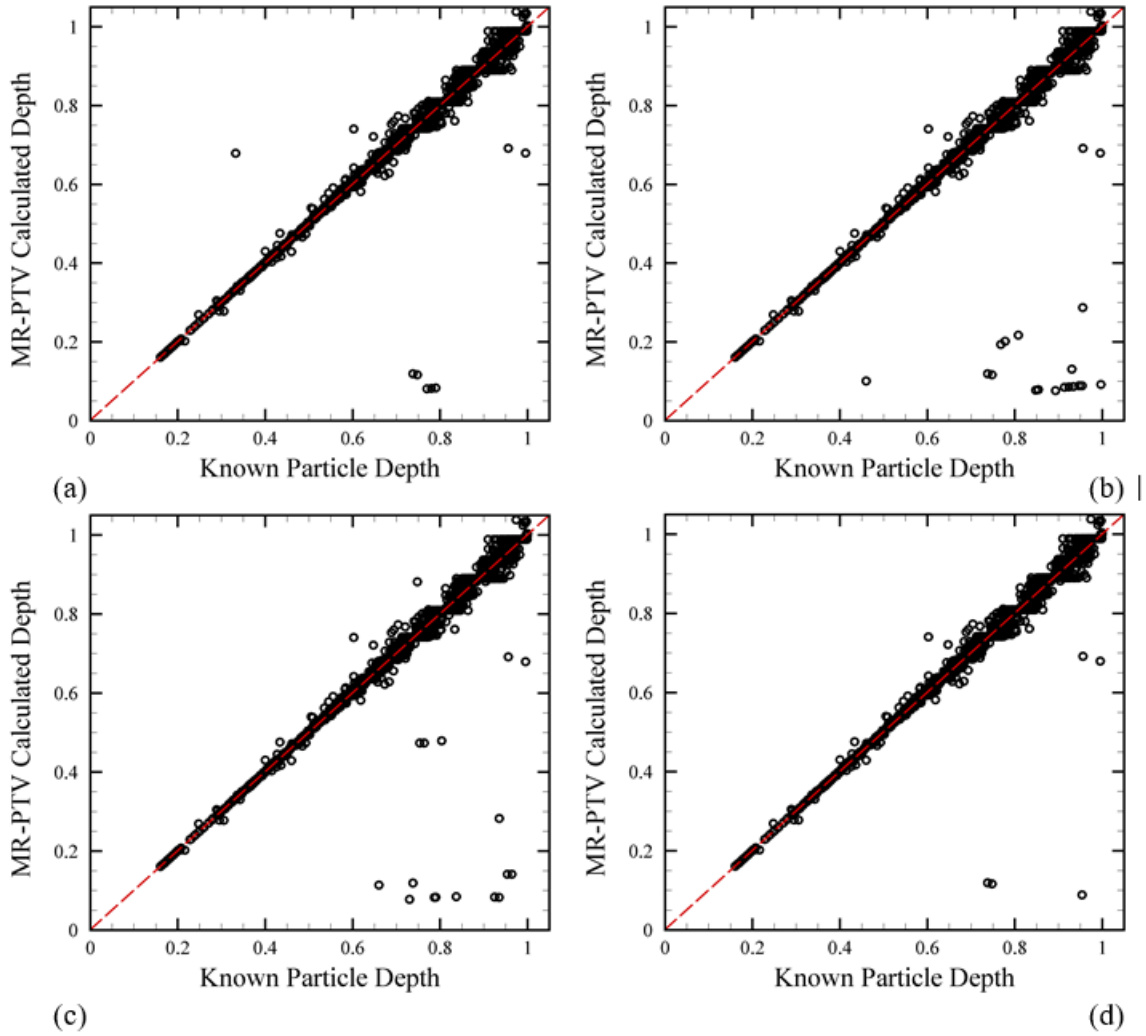


Figure 20: Effectiveness of each term in the stereo matching function given in Eq. (8). After application of the hard filter given in Eq. (7), the images in the figure compare the particle depth predicted by the MR-PTV algorithm with the known depth for cases using (a) only the particle brightness term, (b) only the particle diameter term, (c) only the particle velocity term, and (d) only the particle height Y term. The red dashed line represents an exact match between the predicted and known depth.

in which the depth estimate was repeated using only one term in the matching function for each image. All the test cases that use only one term in (8) result in some points, in the corresponding prediction comparison image in Figure 20, that are far from the exact values, which reinforces the need to include a variety of distance measures for stereo matching.

A second series of tests was conducted to examine how the overall root-mean-square error in prediction of the particle position in the global coordinate system varies with different variables in the computation. In general, the position prediction error is a function of variables characterizing the particles, such that particle diameter d , domain size L , and particle concentration c , as well as variables describing the camera system, such as baseline distance h , pixel width ΔX_{pixel} , focal length f and field of view FOV . In Figure 21, the dimensionless root-mean-square error in the predicted x , y and z particle positions is plotted against key combinations of these variables in order to characterize three specific sources of uncertainty in the MR-PTV predictions.

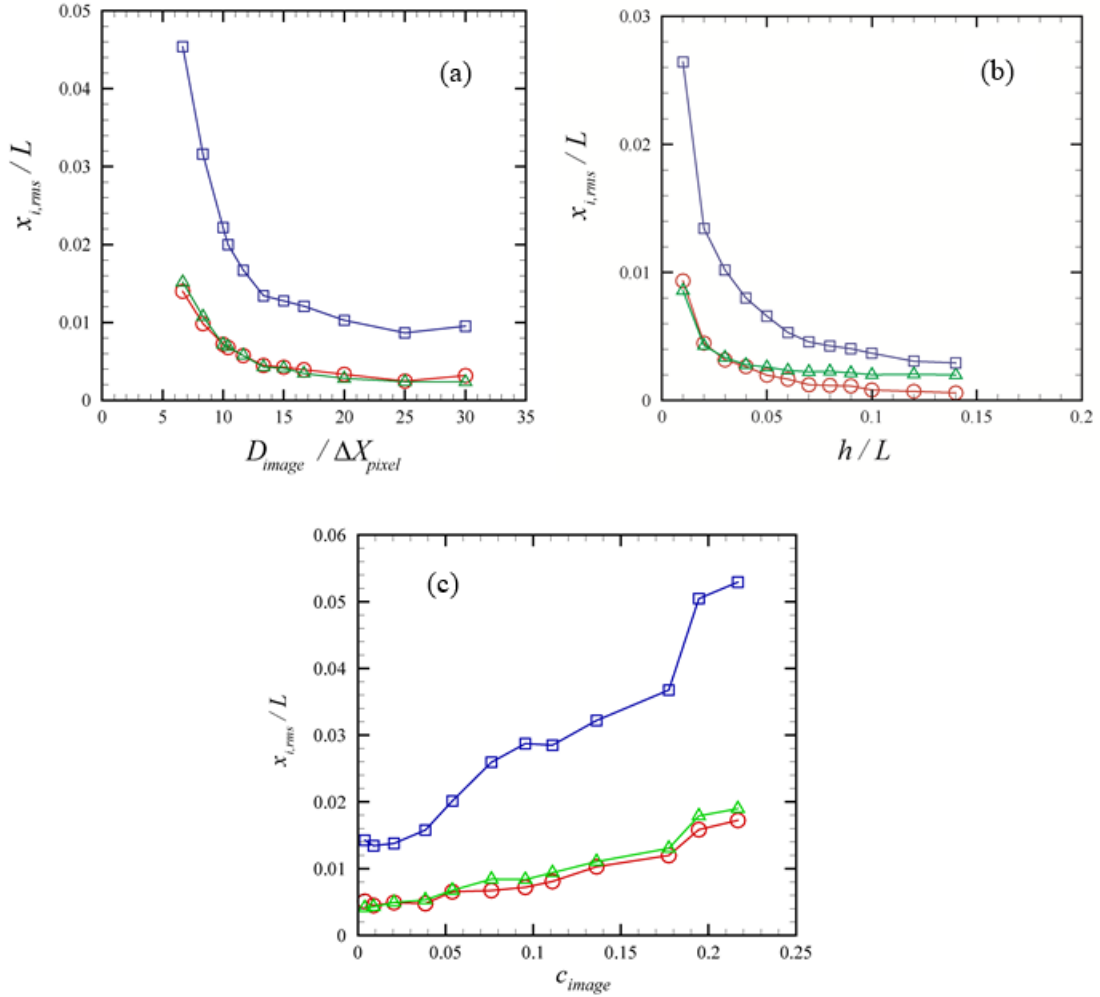


Figure 21: Root-mean-square error divided by the computational domain size L for the estimated positions in the x -direction (red circles), y -direction (green deltas), and z -direction (blue squares). Plots are shown comparing root-mean-square error with (a) number of pixels across the particle standard image-diameter D_{image} , (b) ratio of camera baseline distance h to the domain size L , and (c) particle area concentration in the image plane c_{image} .

In all these results, the uncertainty associated with predicting the x - and y -positions is very similar, and significantly smaller, than the uncertainty associated with the z -position prediction. One key source of error in the MR-PTV algorithm results from resolution limitations of the particle diameter by the pixel image. This error can be characterized by plotting the root-mean-square position error as a function of the ratio $D_{image}/\Delta X_{pixel}$, where

the image diameter D_{image} is defined as the apparent diameter in the image plane of a particle placed at the centroid of the computational domain. The ratio $D_{image} / \Delta X_{pixel}$ is a measure of the number of pixels spanning the characteristic particle diameter in the image plane. The plot, shown in Figure 21a, was generated using 10 different frames each holding 100 particles with diameter $d = 0.015$. The data was generated by gradually increasing the pixel resolution from 180×240 to 1080×1440 in a series of 12 steps while keeping the other variables fixed. A second resolution limitation is associated with the ratio of camera baseline distance h to the overall domain size L . In Figure 21b we show results from a series of tests with progressively increasing baseline distance, showing that the root-mean-square error decreases as the baseline distance to domain size ratio increases. A third source of error occurs when the particle concentration becomes so large that a significant amount of overlap occurs between the particles in the image plane. Overlap error can be expressed as a function of the image plane concentration c_{image} , defined as the ratio of the area covered by particles on the image plane to the total image plane area. A plot of the root-mean-square position error as a function of the image plane concentration is given in Figure 21c. Here, image concentration was varied by gradually increasing the number of particles in the computation. We observe, in Figure 21, a gradual increase in the root-mean-square position error with a corresponding increase in the image plane concentration.

3.5. Moving Least-squares Differentiation

The moving least-square method (Levin 1998) is used to obtain accurate derivatives of noisy measured data, denoted by C_m , $m = 1, N$. The basic idea of this method is to use a set of points in the vicinity of point t_n where the derivative is desired to obtain a low-order

polynomial fit $q_n(t)$ of the measured values C_m on a set of points surrounding t_n . We then estimate the function derivative at t_n by differentiating $q_n(t)$. For instance, by selecting a quadratic polynomial as a low-order fitting function, we can write

$$q_n(t) = a_n + b_n(t - t_n) + c_n(t - t_n)^2, \quad (13)$$

where a_n, b_n, c_n are undetermined coefficients. This quadratic function is fit to a set of M data points on each side of the point n at which the derivative is desired by minimizing a least-square error of the form

$$E_n = \sum_{i=n-M}^{n+M} [C_i - q_n(t_i)]^2, \quad (14)$$

which yields a 3×3 matrix equation for the coefficients a_n, b_n, c_n . After solution of the matrix equation, the time derivative can be estimated at t_n using

$$\frac{dC}{dt}(t_n) = b_n. \quad (15)$$

If $M = 1$, the moving least-square procedure is equivalent to the centered difference scheme for numerical differentiation. Setting $M > 1$ serves to smooth out data fluctuations. Figure 22 demonstrates the effect of selection of M on accuracy of the derivative for the x -component of velocity predicted using the MR-PTV algorithm.

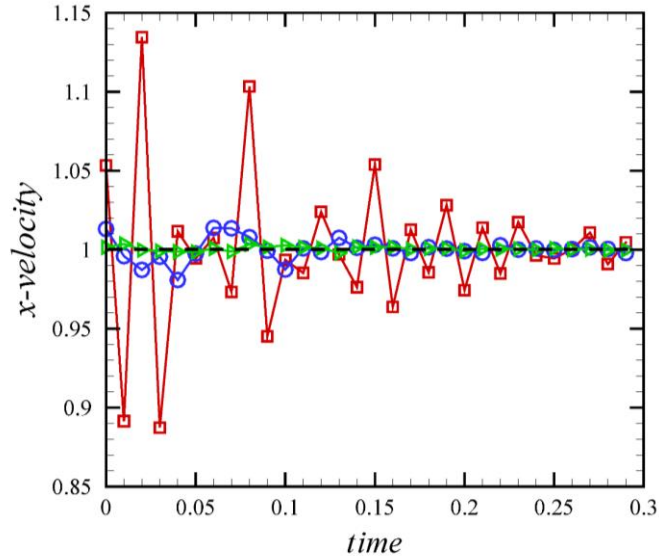


Figure 22: Plot of the x -component of velocity versus time for a uniform flow with velocity components $(1,0,0)$. The effect of the number of points M used in the moving-least-square differentiation procedure on the smoothness and accuracy of the differentiation is demonstrated, using computations with $M = 1$ point (red squares), 2 points (blue circles), and 5 points (green gradients) on each side of the target point.

Figure 22 compares the predicted velocity of a single neutrally-buoyant particle in a uniform flow for cases with $M = 1, 2$ and 5 particles on each side of the target particle in the moving-least-square procedure. Use of moving-least-square differentiation is shown to considerably smooth out fluctuations in the velocity calculation, yielding results close to the exact velocity value.

CHAPTER 4: UNIFORM FLOW VALIDATION

Section 4 presents findings from both computational and experimental tests of the MR-PTV algorithm for particles advected in a uniform flow field. In Section 4.1, synthetic computational data with a prescribed particle velocity is used to evaluate error in the MR-PTV velocity prediction algorithm. Section 4.2 describes an experimental test in which the MR-PTV algorithms are used in combination with the Microsoft HoloLens 2 mixed reality headset to measure the velocity of particles advected by a uniform flow in a flume. Similarly, Section 4.3 uses polystyrene foam particles advected in a wind tunnel to evaluate MR-PTV performance.

4.1. Uniform Flow Computational Validation

A synthetic camera code was developed to map 3D particle positions to a 2D image plane. Images of the 3D particle transport are produced as image pairs from a set of stereo “synthetic cameras”. Each camera has a prescribed focal length f , diagonal field of view FOV , and X and Y pixel resolutions $N_{pix,X}$ and $N_{pix,Y}$. The baseline distance between the stereo synthetic cameras is denoted by h .

4.1.1. Synthetic Image Generation

A cube of unit dimensionless side length was selected as the computational domain, consistent with our decision to non-dimensionalize distances by the computational domain depth L . The computational domain was randomly seeded with N particles with uniform diameter d . A particle with global position (x_n, y_n, z_n) was mapped to a two-dimensional bitmap with center at position (X_n^L, Y_n^L) in the left camera and (X_n^R, Y_n^R) in the right camera. Mapping each camera is performed by defining a 3D “camera” coordinate

system (x_c, y_c, z_c) and translating the global coordinate system such that the origin coincides with the camera lens center and rotating the global coordinate system such that the z -axis aligns with the camera lens axis. We use particle position in the camera coordinate system is used to determine the corresponding particle center position in the image plane using the mapping

$$X_n = \frac{fx_{c,n}}{z_{c,n}}, \quad Y_n = \frac{fy_{c,n}}{z_{c,n}}, \quad (16)$$

where f is the camera focal length. The apparent diameter D_n and brightness B_n of particle n in the image plane are given by

$$D_n = \frac{fd}{R_n}, \quad B_n = \frac{\ell}{4\pi R_n^2}, \quad (17)$$

where R_n denotes the distance between the centroid of particle n and the origin of the camera coordinate system (i.e., the center of the camera lens) and ℓ is a prescribed particle illumination coefficient. Particles closer to the camera appear both larger and brighter in the images than more distant particles, as shown in Figures 3 and 4.

Pixels in the synthetic camera bitmap that are completely covered by the image of a particle n have intensity set to the particle image brightness B_n , while pixels that are not covered by the particle images have pixel intensity 0. For pixels partially covered by a particle image, the pixel intensity is approximated using a two-dimensional version of the edge-length method proposed by Galindo-Torres (2013). In this algorithm, the fraction ε of the pixel cell area covered by a particle image, is estimated using the sum of the pixel

cell edge lengths $\ell_{p,i}$ covered by the particle image divided by the total pixel edge length,

or

$$\varepsilon \approx \sum_{i=1}^4 \ell_{p,i} / 4\Delta X_{pixel} . \quad (18)$$

The pixel intensity for partially-covered images is then set equal to εB_n .

4.1.2. Synthetic Data results

We performed computations using $N = 100$ particles of dimensionless diameter $d = 0.015$ in a uniform flow with prescribed particle velocity components $(u, v, w) = (1, 0, -1)$. Synthetic cameras with dimensionless baseline separation $h = 0.1$ and focal length $f = 0.01$ output stereo image pairs with a frame rate $\Delta t = 0.01$ and 720 x 960 pixel resolution. For any given frame, the velocity vector of a particle is only calculated if the particle appears in the image of both cameras for $M = 5$ frames on each side of the current frame, where M is the number of frames used on each side of the given frame in the moving least-squares differentiation method described in Section 5.5.

Particle velocity predictions from the MR-PTV algorithm are plotted in Figure 23a for a series of 10 frames as a function of particle depth z .

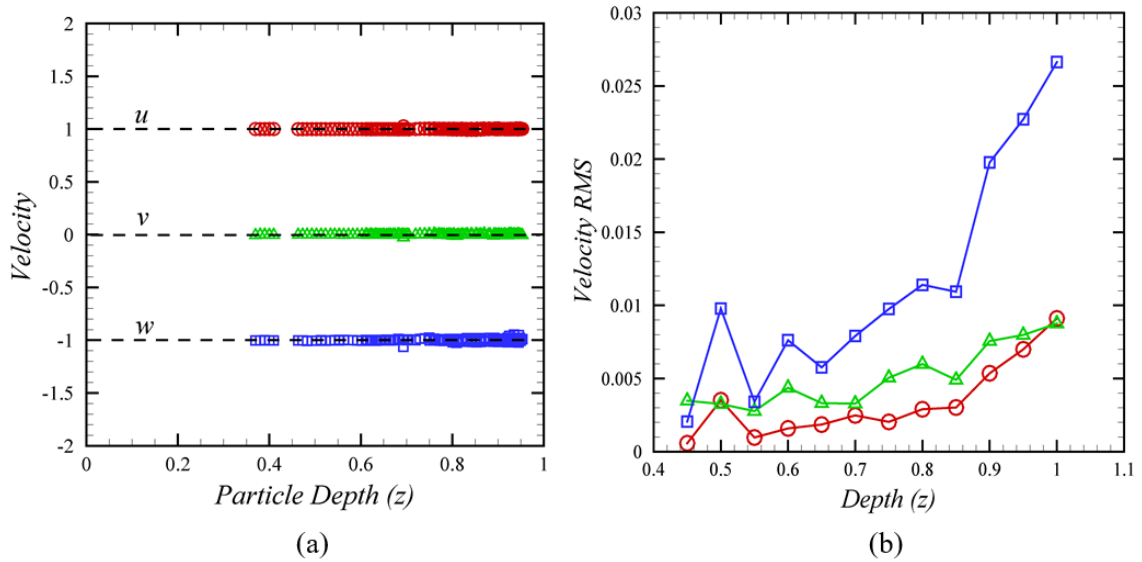


Figure 23: In (a), the computed particle velocity components in the u (red circles), v (green deltas), and w (blue squares) directions are plotted as functions of the particle depth from the camera. The dashed lines are the exact values of known particle velocities. In (b), the root-mean square velocity error is plotted for each component as functions of particle depth.

Resulting computational predictions compare well with the exact velocities, indicated in Figure 23, by dashed lines for each velocity component. Figure 23b shows the root-mean-square error in the velocity prediction as a function of particle depth z using a series of 20 bins on the z -axis. The data shows error for each velocity component increases with increase in the particle depth z . The root-mean-square error of the out-of-plane velocity component w is roughly 2-3 times greater than that of the in-plane velocity components u and v .

4.2. Flume Experiment

4.2.1. Flume Method

Experimental validation was performed by imaging particles advected by flow in an open channel hydraulic flume, with an outer rectangular footprint measuring 3.9×2.0 m and a motor-controlled water wheel to generate flow. A 50 cm wide flume is filled with

water to a height of 40 cm. The side and bottom flume walls are constructed with clear acrylic sheets for flow visualization. Rigid guide vanes were fixed in each corner of the flume to maintain flow uniformity. A honeycomb steel mesh, inserted upstream of the test section, served as a flow conditioner. The flume has a retractable sluice gate, with thickness 1.0 cm, extends to a depth of 26 cm below the water surface. In the uniform flow experiments discussed in the current section, the sluice gate was removed. Particles were introduced into the flume flow using a perforated aluminum hopper which spanned 33.7 cm across the width of the flume and 22.3 cm in the direction of flow. The hopper was positioned 2 mm above the water surface and the hopper midpoint was 63 cm upstream of the camera center. A schematic of the experimental configuration and coordinate system is shown in Figure 24.

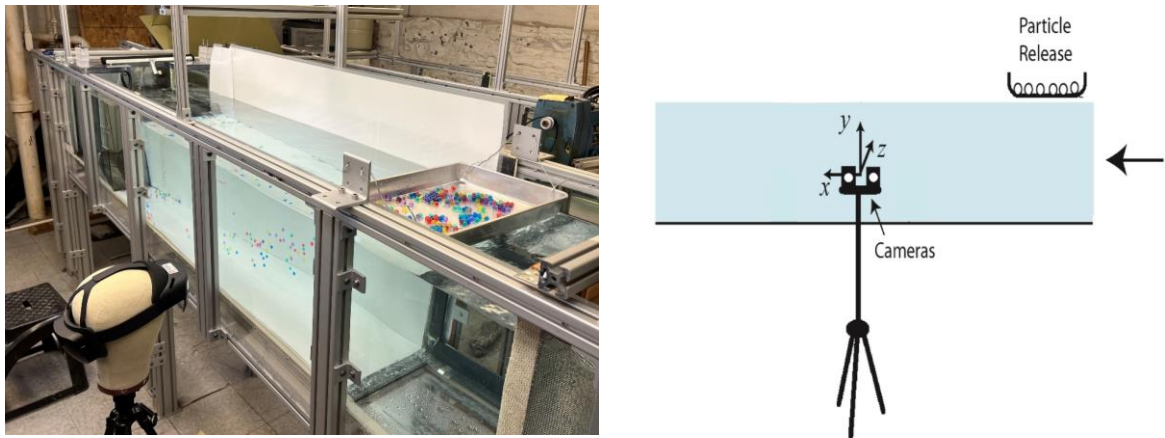


Figure 24: (Right) Laboratory testing set-up for flume experiment. (Left) Schematic diagram showing experimental set-up used for flume validation experiments.

We seeded the flow with multi-colored spherical water gel particles (Spin Master). Using digital calipers to measure particle diameter for a set of 50 particles, the average particle diameter was found to be $d = 10.6 \pm 1.3$ mm. The water wheel rotation rate

controlled the free-stream fluid velocity. The comparison particle velocity in the horizontal x -direction in the test section was determined by tracking the time required for a particle to traverse a fixed horizontal distance (0.9 m), yielding a horizontal particle velocity estimate of $u = 0.0983 \pm 0.005$ m/s. The particle sedimentation velocity in the y -direction was measured using a cylindrical tank with a height of 1.0 m. We tracked the time required for the particle to fall a fixed distance (0.7 m) at terminal velocity, yielding a particle vertical velocity of $v = 0.0372 \pm 0.003$ m/s. For both velocity components, the reported uncertainty values were based on the root-mean-square of 25 repeated tests. By comparing the measured particle sedimentation velocity to the theoretical terminal velocity of the particles, obtained using the Schiller-Naumann drag formula for a sphere (Schiller and Naumann, 1933), we estimated the effective density of the particles to be $\rho_p = 1.006$ g/ml.

Images were captured using the two front-facing visible light cameras onboard the Microsoft HoloLens 2 mixed reality headset (Figure 1). The headset was fixed to a mannequin head aligned perpendicularly to the direction of flow with 49 cm between the camera lens and the flume mid-section. The stereo cameras are co-directional, synchronized, and separated by a baseline distance $h = 9.9$ cm. Each camera has video resolution of 480×640 pixels and a framerate of 30 Hz. A checkered calibration board was used to verify the camera configuration. The MATLAB stereo camera calibrator app was used to validate that intrinsic and extrinsic camera parameters matched the manufacturing details. Next, we extracted particle images from the HoloLens 2 and processed externally using the MR-PTV algorithm presented in Section 3. Corrections to particle positions

accounted for refraction at the flume wall using the refraction correction algorithm of Bao and Li (2011).

4.2.2. Flume Results

Measured MR-PTV velocity components in uniform flow are presented in Figure 25, where each scatter point represents either the x or y particle velocity components of a particle for velocity readings taken from a set of three frames.

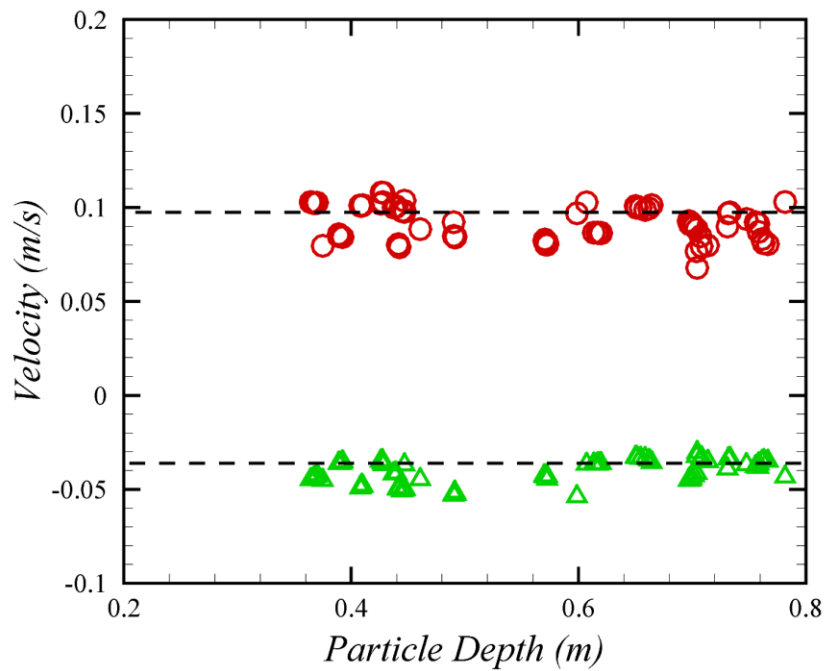


Figure 25: Particle velocity measurements for uniform flow in a flume from MR-PTV with three data frames. Red circles and green triangles represent particle x and y velocity components, respectively. The black dashed lines in Figure 25 indicate the measured particle settling velocity and the flow velocity in the flume for the y and x velocity components, respectively. Particle depth is defined as the distance from the cameras along the camera principal axis. It is expected that particles are flowing at a particle depth between 32.2 -65.8 cm. The average and root-mean square values of both the MR-PTV measured particle velocities and the comparison

values are given in Table 1. We found the MR-PTV measurements to be within the measurement uncertainty of the comparison values.

Table 1: Average and root-mean square values from MR-PTV measurements for the uniform flow flume validation test. Comparison values are the measured particle velocities based on time required to travel a prescribed distance.

Velocity Component	Comparison (m/s)		MR-PTV Measurements		Percent Difference (%)
	Average (m/s)	Root-mean square (m/s)	Average (m/s)	Root-mean square (m/s)	
u	0.0983	0.005	0.0965	0.012	3.3
v	-0.0372	0.003	-0.0390	0.007	6.7

4.3. Wind Tunnel Experiment

4.3.1. Wind Tunnel Method

A second experimental validation procedure was conducted to measure the velocity of expanded polystyrene foam particles flowing in an open return wind tunnel. The general process was similar to the experiment conducted in the flume, where the HoloLens 2 MR headset, oriented perpendicular to the direction of flow, was used to image particles advected in flow as shown in Figure 26.



Figure 26: Wind tunnel experimental setup. HoloLens 2 positioned perpendicular to flow direction in wind tunnel. Particle hopper pictured in upper left corner.

Particles were seeded in the flow using a particle hopper, which spanned the cross-section of the wind tunnel, pictured in the upper left corner of Figure 26. The hopper rested on top of the wind tunnel and the particle feed rate was controlled manually by sliding a cover plate that allowed particles to fall into the wind tunnel. Expanded polystyrene foam particles with diameter, $d = 4.18 \pm 0.38$ mm, were used in the experiment. Diameter measurements were obtained by measuring the diameter of 50 particles using digital calipers. The particle settling velocity, $v = -1.37 \pm 0.09$ m/s was calculated from a series of 50 foam particles falling 2.0 m at terminal velocity. Flow speed inside the wind tunnel was controlled by adjusting the fan speed at the end of open return wind tunnel. The fluid velocity component x -direction, $u_f = 1.52 \pm 0.10$ m/s, was measured using a hot wire anemometer positioned at the midline of the wind tunnel. Horizontal wind velocity

measurements were consistent throughout all areas of the observed test section. A black background curtain was placed at a depth of 110 cm from the camera.

4.3.2. Wind Tunnel Results

Particles in the wind tunnel were transported at a considerably higher velocity magnitude compared to particles in the water flume. A limitation of the current version of the Microsoft HoloLens 2 is a fixed framerate of 30 Hz. The fixed camera framerate makes it more difficult to track and link particles because a single particle appears in fewer frames and travels a greater distance between frames. Given this limitation, the wind tunnel results were performed using only $M = 2$ nearest points in the moving least-squares gradient filter described in Section 3.5.

Results from the wind tunnel experiment are shown in Figure 27, where u (x -component velocity) is indicated by red circles, v (y -component velocity) is indicated by green triangles, and w (z -component velocity) is indicated by blue squares. Velocity components are plotted against depth from camera on the horizontal axis in Figure 27.

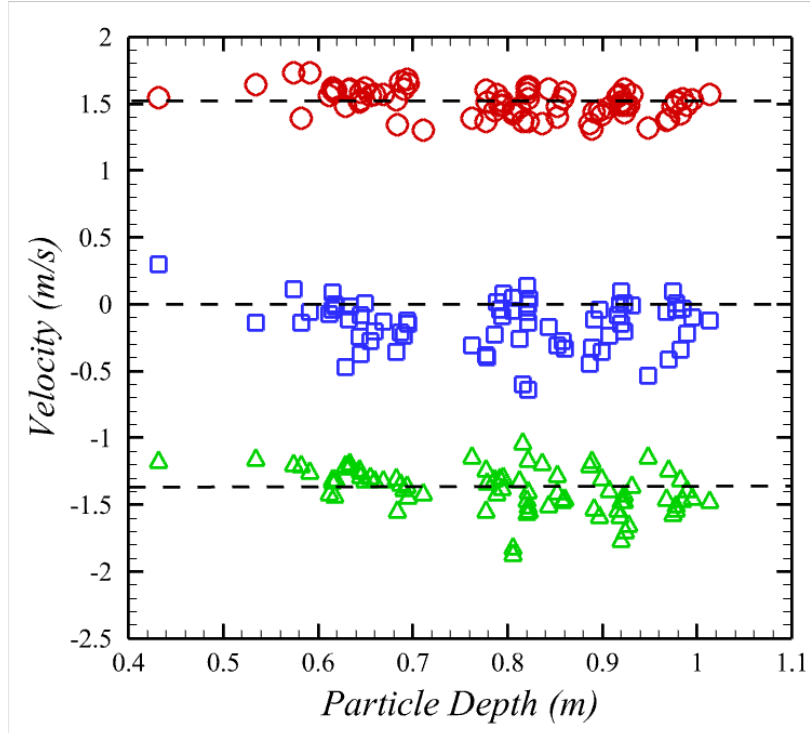


Figure 27: Particle velocity measurements for uniform wind tunnel flow from MR-PTV with three data frames. Red circles, green triangles, and blue squares represent particle x, y, and z velocity components, respectively.

Black dashed lines represent the expected velocity readings $(u, v, w) = (1.52, -1.37, 0)$ m/s, given by the fluid velocity, the particle sedimentation velocity and zero, respectively. The root-mean-square values of the velocity measurements, shown in Table 2, are significantly greater for the wind tunnel results than the hydraulic flume results shown in Table 1. This is results from a greater particle velocity-to-framerate ratio in the wind tunnel experiments, resulting in fewer images containing a given particle. When in-plane velocity measurements are compared, the vertical velocity component v was observed to have a larger RMS than the horizontal velocity component u . Our data shows the RMS error for the out-of-plane velocity, w , is nearly double the RMS of the in-plane velocity, u , consistent with our results for the flume experiments and synthetic data. Table 2 shows agreement

between the average MR-PTV velocity component measurements and experimentally measured comparison values of velocity.

Table 2: Average and root-mean square values from MR-PTV measurements for the uniform flow wind tunnel validation test. Comparison values are the measured particle velocity based on time required to travel a prescribed distance for v and the measured fluid velocity for u .

Velocity Component	Comparison (m/s)		MR-PTV Measurements	
	Average (m/s)	Root-mean square (m/s)	Average (m/s)	Root-mean square (m/s)
u	1.52	0.10	1.53	0.095
v	-1.37	0.09	-1.38	0.149
w	0.0	N/A	-0.14	0.184

CHAPTER 5: ISOTROPIC TURBULENCE VALIDATION

Validation tests for synthetic data using neutrally buoyant particles in an isotropic turbulent flow were conducted to examine a flow field in which nearby particles observed on the image plane have very different velocities.

5.1. Turbulence Computational Method

Direct numerical simulations of isotropic, homogeneous turbulence were obtained using a triply-periodic pseudo-spectral method with second-order Adams-Bashforth time stepping and exact integration of the viscous term (Vincent and Meneguzzi, 1991). In this approach, the spectral Navier-Stokes equations evolve in time after having been projected onto a divergence-free space using the operator $P_{ij} = k_i k_j / k^2 - \delta_{ij}$ according to the expression

$$\bar{\mathbf{u}}^{n+1} = \bar{\mathbf{u}}^n \exp(-\nu k^2 \Delta t) + \Delta t \mathbf{P} \cdot \left[\frac{3}{2} \bar{\mathbf{F}}^n \exp(-\nu k^2 \Delta t) - \frac{1}{2} \bar{\mathbf{F}}^{n-1} \exp(-2\nu k^2 \Delta t) \right], \quad (19)$$

where an overbar denotes Fourier transform in three space dimensions, a superscript indicates the time step, ν is the kinematic viscosity, and \mathbf{k} is the wavenumber vector with magnitude k . The force vector \mathbf{F} on the right-hand side has Fourier transform given by

$$\bar{\mathbf{F}} = \overline{\mathbf{u} \times \boldsymbol{\omega}} + \bar{\mathbf{f}}_F, \quad (20)$$

where $\bar{\mathbf{f}}_F$ is a small wavenumber forcing term required to maintain the turbulence with approximately constant kinetic energy. The velocity field was made divergence-free at each time step by taking its Fourier transform and using the spectral form of the continuity equation,

$$\mathbf{k} \cdot \bar{\mathbf{u}} = 0. \quad (21)$$

The forcing vector was assumed to be proportional to the fluid velocity (Lundgren, 2003; Rosales and Meneveau, 2005), such that

$$\bar{\mathbf{f}}_F = \begin{cases} C\bar{\mathbf{u}} & \text{for } k < k_{crit} \\ 0 & \text{for } k > k_{crit} \end{cases}, \quad (22)$$

where the coefficient C was set equal to $C = 0.0045 / E_{low}$ and $E_{low} = \frac{1}{2} \sum_{k < k_{crit}} \bar{\mathbf{u}} \cdot \bar{\mathbf{u}}$ is the kinetic energy in all modes with wavenumber amplitude $k < k_{crit}$. The current computations were performed with $k_{crit} = 5$, so that the forcing acts only on the large-scale eddies.

The turbulence kinetic energy q and dissipation rate ε_{diss} were obtained from the power spectrum $e(k)$ as

$$q = \int_0^{k_{max}} e(k) dk, \quad \varepsilon_{diss} = 2\nu \int_0^{k_{max}} k^2 e(k) dk. \quad (23)$$

Various dimensionless measures describing the turbulence in the validation computations are listed in Table 3, including the root-mean-square velocity magnitude u_0 , the average turbulence kinetic energy q , the integral length scale $\ell_0 = 0.5 u_0^3 / \varepsilon_{diss}$, the Taylor microscale $\lambda = (15\nu / \varepsilon_{diss})^{1/2} u_0$, and the Kolmogorov length scale $\eta = (\nu^3 / \varepsilon_{diss})^{1/4}$. The corresponding microscale Reynolds number is $\text{Re}_\lambda = u_0 \lambda / \nu = 99$.

Table 3: Dimensionless simulation parameters and physical parameters of the fluid turbulence.

Simulation Parameters		Turbulence Parameters	
Time step	0.002	Turbulent kinetic energy, q	0.122
Cycles	15000	Mean dissipation rate, ε_{diss}	0.015
Grid	128^3	Kinematic viscosity, ν	0.001
		Integral length, ℓ_0	0.771
		Taylor microscale, λ	0.285
		Kolmogorov length, η	0.016
		Integral velocity, u_0	0.285
		Integral time, T_ℓ	2.71

5.2. Turbulent Results

Performance of the MR-PTV system in turbulent flow was evaluated computationally using the same parameter values used in Section 3.1, with $N = 100$ computational particles, dimensionless camera baseline distance $h = 0.1$, image resolution 720×960 pixels, and time step $\Delta t = 0.01$ between frames. Particles are neutrally bouyant tracers advected by the flow field, with diameter $d = 0.015$. The velocity field was evaluated using the MR-PTV system for a series of 40 frames.

A visualization of the turbulent flow is shown by Figure 28, showing contours of the x-component of velocity.

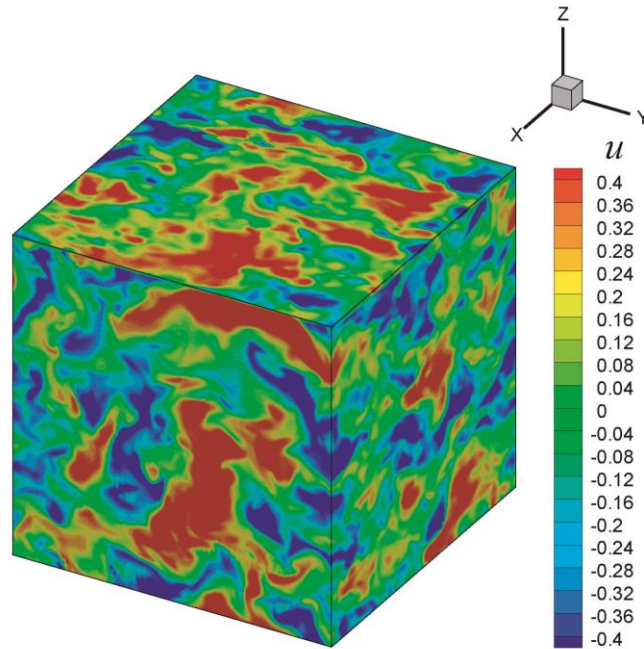


Figure 28: Contour plot of the x-component of velocity in the isotropic turbulent flow.

A comparison of the MR-PTV predictions for the three particle velocity components and the known particle velocities is given in Figure 29a-c, where each scatter point represents particle velocity at a given time.

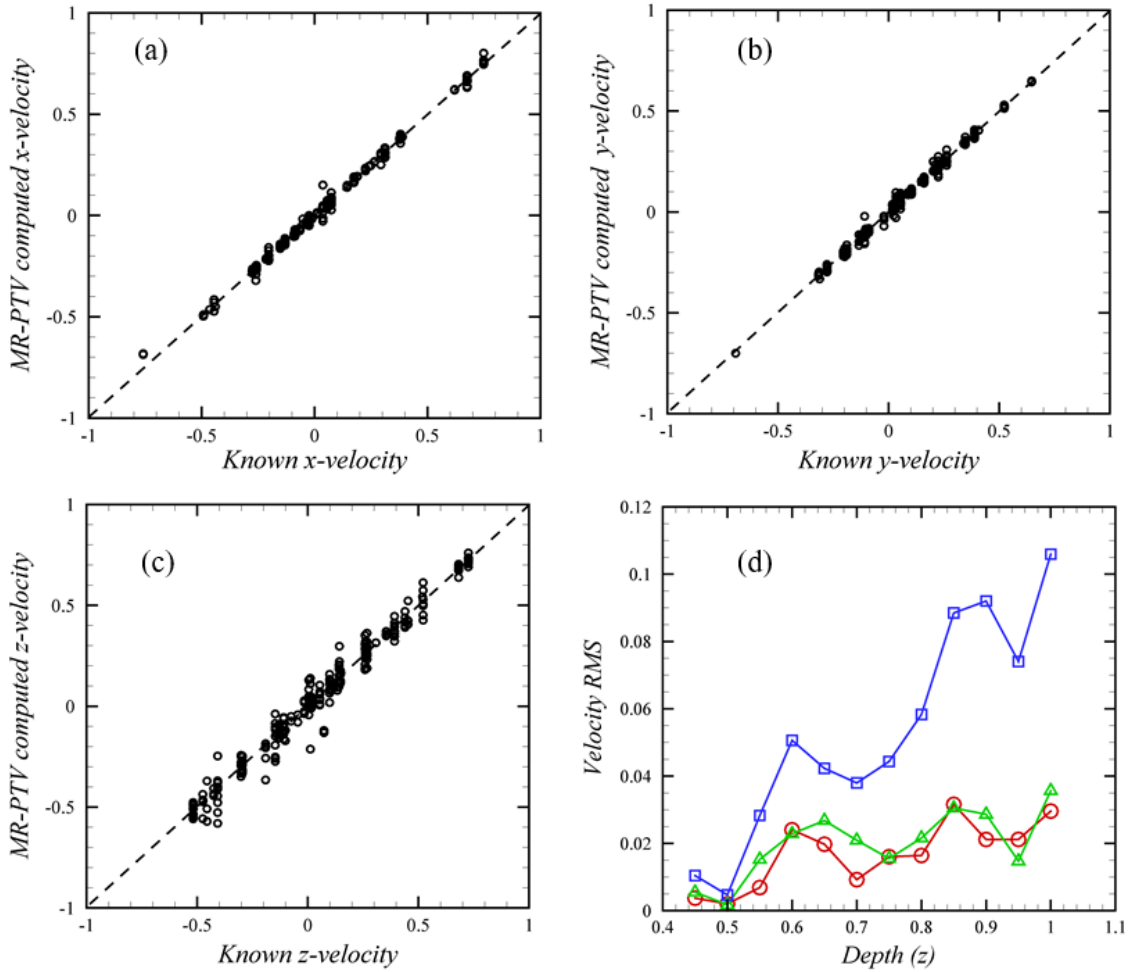


Figure 29: (a-c) Scatter plots comparing the x -, y - and z -velocity components of particles in an isotropic turbulent flow computed using the MR-PTV algorithm versus the known particle velocity. (d) Root-mean-square error in predicted velocity as a function of particle depth z .

The root-mean square error for each velocity component is plotted in Figure 29d as a function of particle depth z , where the root-mean square of the particle velocities was evaluated within each of a series of 20 bins on the z -axis. Similar to the uniform flow computational results in Section 4.1, we observe increased root-mean square error for the out-of-plane z -component of velocity compared to the in-plane x - and y -components, with the velocity error generally increasing with an increase in particle depth z .

CHAPTER 6: SLUICE GATE VALIDATION

In Section 6, performance of the MR-PTV algorithm is evaluated for flow past a sluice gate. A numerical solution of the fluid and particle velocity fields is compared to the MR-PTV measured particle velocities around a sluice gate submerged in flume flow. The computational method is described in Section 6.1 and the comparison with experimental results is given in Section 6.2.

6.1. Sluice Gate Method

In this section we describe the method used to obtain a numerical solution for velocity around a partially submerged sluice gate that matches the physical domain of the flume. The same general experimental procedure was followed as described in Section 4.2.1., where the HoloLens 2 is positioned perpendicular to the flow inside the flume. Particles, with diameter $d = 10.6 \pm 1.3$ mm and density $\rho_p = 1.006$ g/ml, are seeded into the flow using a perforated aluminum hopper fixed 2mm above water surface level. In this section we observe the velocity field around a submerged sluice gate body inside the flume as pictured in Figure 30.

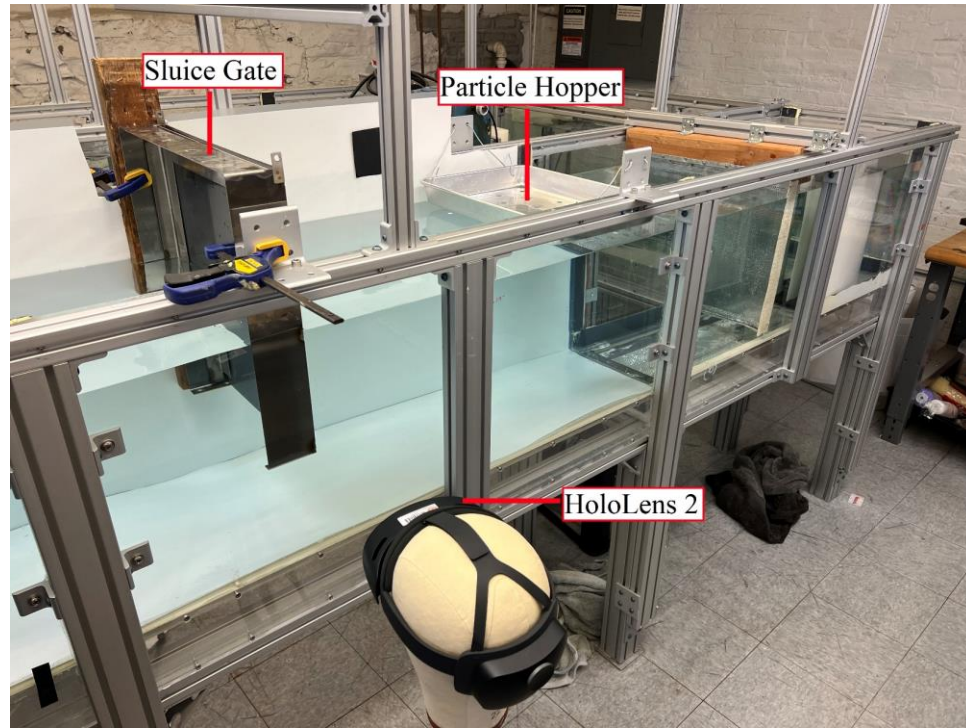


Figure 30: Experimental configuration of sluice gate flume flow.

Particles released from the hopper upstream flow under the submerged sluice gate body while imaged by the HoloLens 2. MR-PTV experimental results were compared to computational velocity results from a computational domain modeled to match the physical experiment.

We used ANSYS Fluent finite-volume computational fluid dynamics (CFD) code to model incompressible fluid flow in the physical flume domain and extract the steady-state fluid velocity field for a simulation of water flowing past a submerged sluice gate. The CFD simulation used a 2D domain, modeled to match the flow inside the flume. The origin of the computational domain is set to the real perpendicular position of the HoloLens origin during the experiment. The boundary conditions used are shown in Figure 31.

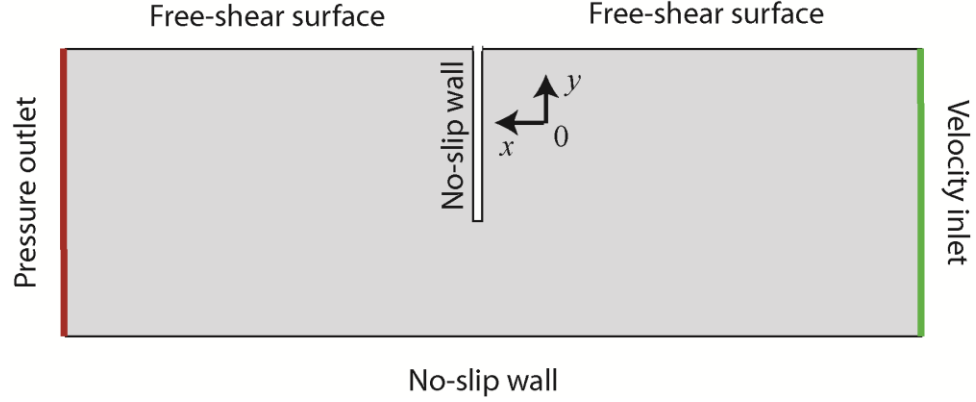


Figure 31: Domain boundary conditions used in CFD simulation, where the origin is where the Microsoft HoloLens 2 was positioned perpendicular to the domain.

The right wall of the domain is a flow velocity inlet with uniform velocity set to the free-stream flow velocity inside the flume ($u = 0.0983$ m/s), and the left wall is a pressure outlet. The bottom surface and the sluice gate are no-slip wall boundaries. The top wall of the domain is fixed, free-shear surface. Our experiments showed that at the low fluid flow rates used in the validation tests, the fluid height difference between the upstream and downstream sides of the side gate were negligible. A uniform Cartesian mesh contained 2020 elements in the x direction and 800 elements in y direction. Convergence with relative error of less than 10^{-6} was reached for both velocity components. Grid independence was demonstrated using both coarser and finer grids and comparing the velocity field.

Theoretical particle motion was computed by solving the particle momentum equation

$$(m + m_A) \frac{d\mathbf{v}}{dt} = -3\pi\mu d(\mathbf{v} - \mathbf{u})f_l, \quad (24)$$

where $m_A = 0.5\rho_f V$ is the particle added mass, m , V and d are the particle mass, volume and diameter, μ and ρ_f are the fluid viscosity and density, \mathbf{v} and \mathbf{u} are the particle and fluid velocity vectors, and

$$\text{Re}_p \equiv \frac{\rho_f d |\mathbf{v} - \mathbf{u}|}{\mu} \quad (25)$$

is the particle Reynolds number. The factor

$$f_I = 1 + 0.15 \text{Re}_p^{0.687} \quad (26)$$

is an inertial correction factor (Schiller and Naumann, 1933), which is valid for $\text{Re}_p < 800$. All values used in the particle momentum equations were set to match experimental conditions. Here we included the particle inertia and the added mass and drag forces. Other forces on the particles, such as lift, pressure gradient, and history forces, are negligible under the conditions used in the flow computations (Marshall and Li, 2014).

For the current sluice gate validation, computational particles were initialized in a horizontal rake on the right-hand side of the computational domain, matching the physical location where particles were released from the aluminum hopper during the experiment. Theoretical particle pathlines were determined by tracking the particle positions across the computational domain.

6.2. Sluice Gate Results

Velocity measurements results from the MR-PTV system and computed particle velocity fields are shown in Figure 32.

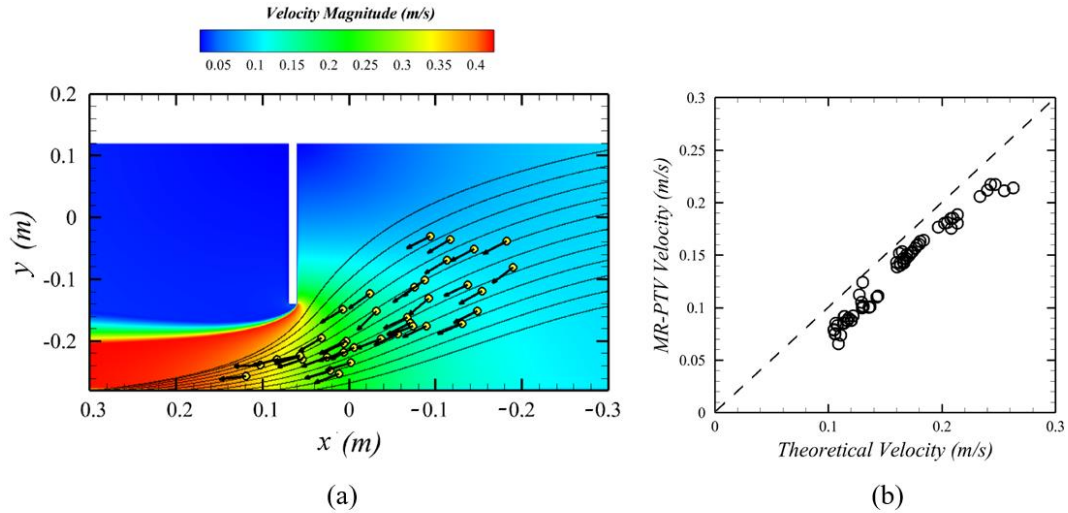


Figure 32: MR-PTV measured velocity vectors for flow past a sluice gate. (a) Scatter plot showing experimental particles (yellow) with MR-PTV velocity direction (black arrows) superimposed onto CFD contour plot of velocity field for flow past a sluice gate, with computed particle pathlines indicated by black lines. (b) Comparison of velocity magnitude at particle locations as measured experimentally using the MR-PTV algorithm (y-axis) and as simulated by solving the particle momentum equation with the computed velocity field (x-axis).

In Figure 32a, the direction of the experimental MR-PTV velocities at each measured particle position are compared to the particle pathlines computed from solution of (24). Particle experimental velocity direction vectors, and particle pathlines are plotted overlying a color plot of the computed fluid velocity field. The sluice gate is shown as a vertical white bar protruding into the flow. In Figure 32b, the velocity magnitude of the experimental MR-PTV velocities are compared to magnitudes of the computed particle velocities at the same locations.

The overall comparison of experimental MR-PTV data and computational data in Figure 29 show good agreement. Most of the experimental MR-PTV velocity vectors are oriented along the computed particle pathlines in Figure 29a. In few cases particle vectors are not perfectly tangent to computed pathlines but are agreeable knowing there are a distribution of particle diameters seeded in flume flow, while only the average particle

diameter is used to compute particle pathlines. The experimental particle velocity magnitudes in Figure 32b appear to be within about 10% of the predicted values, a promising agreement given the experimental uncertainties and various simplifications involved in both the CFD and particle tracking computations.

CHAPTER 7: CONCLUSION

We present a particle tracking velocimetry system compatible for use with a mixed-reality headset, such as the Microsoft HoloLens 2. The system uses video feed from two co-directional, synchronized, stereo cameras to detect and track particles through time to estimate the three-dimensional particle velocity vectors. Adaptive thresholding is used to detect particles in individual images. A stereo matching algorithm is optimized for three-dimensional flow fields using monocular cues, such as brightness, depth, and apparent velocity, to match particles between stereo images. A moving least-squares gradient filter uses a time series of particle images to reduce noise in performing time differentiation of the particle position in estimating particle velocity.

The approach was validated computationally using both synthetic data, generated by a discrete-element method, and using experimental data from flume and wind tunnel experiments. Synthetic data was used to perform a detailed error analysis for determination of particle depth from the camera. Three major sources of error were identified, including pixel resolution of the particle image, baseline distance between the cameras relative to the overall flow dimension, and particle concentration in the image. Synthetic data was then used to examine accuracy of MR-PTV velocity estimates for particles in both uniform flow and isotropic turbulence, and for both in-plane and out-of-plane particle velocity components.

The MR-PTV approach was experimentally validated using the Microsoft HoloLens 2 mixed reality headset to image particles flowing in a flume and a wind tunnel. Flume experiments tested the MR-PTV system for both uniform flow and flow around a

submerged sluice gate. Uniform flow in-plane velocity components exhibited an average error of 3.3% as compared to comparison values, which was within the measurement uncertainty of the comparison data. The direction of MR-PTV computed velocity vectors agrees with computed particle pathlines for particles flowing around a sluice gate body, although with some directional scatter, likely resulting from a distribution in tracer particle diameters. The magnitude of the experimental MR-PTV velocity vector measurements were within 10% of computed particle velocities, which was within estimated uncertainty of the numerical computations. Wind tunnel MR-PTV velocity measurements showed agreement within 1% of measured comparison values for in-plane velocity components. The out-of-plane velocity component had nearly double the RMS when compared to the in-plane velocity component, a result consistent with the findings from sluice gate experiments.

The MR-PTV algorithm presented in the paper is optimized to perform with the camera configuration limitations of a typical mixed reality headset. While many existing PTV systems are designed for precision measurements in controlled laboratory settings and require expensive specialized equipment (lasers, multiple cameras, synchronizers, etc), the MR-PTV algorithm requires only a mixed-reality headset and appropriate seeding medium. The algorithm uses monocular artifacts of non-uniformity, which may be present in a natural environment, to match particles between stereo images, which when combined with binocular disparity can generate reasonably accurate velocity estimates without requiring additional specialized equipment. The MR-PTV approach has significant promise for use

as a particle velocity visualization tool in a variety of industrial, entertainment, and strategic applications.

REFERENCES

- Adrian, R., Multi-Point Optical Measurements of Simultaneous Vectors in Insteady Flow - a Review. *Int. J Heat and Fluid Flow* **7**, 127-145 (1986).
- Adrian, R. J., and Westerweel, J., Particle Image Velocimetry. *New York: Cambridge University Press* (2011).
- Alhashim, I., and Wonka, P., High Quality Monocular Depth Estimation via Transfer Learning (2019).
- Anandan, P., A computational framework and an algorithm for the measurement of visual motion. *IJCV* **2**, (3) 283–310 (1989).
- Atkinson, C., and Soria, J., An efficient simultaneous reconstruction technique for tomographic particle image velocimetry. *Exp. Fluids* **47**, 553–68 (2009).
- Bao, X., and Li, M., Defocus and binocular vision based stereo pairing method for 3D particle tracking velocimetry. *Opt. Lasers Eng.* **49**, 623–31 (2011).
- Barnkob, K., Kahler, C. J., Massimiliano, R., General defocusing particle tracking. *Lab on a chip*, **15**, 3556-3560 (2015).
- Bryanston-Cross, P.J., Funes-Gallanzi, M., Quan, C., and Judge, T.R., Holographic particle image velocimetry (HPIV). *Optics & Laser Technology* **24**, 251-256 (1992).
- Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Yu., Baldan, G., and Beijbom, O., nuScenes: A multimodal dataset for autonomous driving. *IEEE/CVF conference on computer vision and pattern recognition* 11621–11631 (2020).
- Carosone, F., Cenedese, A., and Querzoli, G., Recognition of partially overlapped particle images using the Kohonen neural network. *Exp. Fluids* **19**, 225–32 (1995).
- Coletti, F., Toloui, M., Fong, K. O., Nemes, A., and Baker, L., Volumetric distribution and velocity of inertial particles in a turbulent channel flow. (2016)
- Crocker, J., and Grier, D., Methods of digital video microscopy for colloidal studies. *Journal of Colloid and Interface Science* **179**, 298–310 (1996).
- Doh, D. H., Cho, G. R., and Kim, Y. H., 2012 Development of a tomographic PTV. *J. Mech. Sci. Technol.* **26**, 3811–9 (2012).
- Elsinga, G. E., Scarano, L., Wieneke, B., and Van Oudheusden, B., Tomographic particle image velocimetry. *Exp. Fluids* **41**, 933–947 (2006).
- Elsinga, G. E., Westerweel, J., Scarano, F., Novara, M., On the velocity of ghost particles and the bias errors in tomographic-PIV. *Exp. Fluids* **50**, 825–38 (2011).

- Fu, H., Gong, M., Wang, C., Batmanghelich, N., and Tao, D., Deep ordinal regression network for monocular depth estimation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition 2002–2011* (2018).
- Galindo-Torres, S.A., A coupled discrete element lattice Boltzmann method for the simulation of fluid-solid interaction with particles of general shapes. *Computer Methods in Applied Mechanics and Engineering* **265**, 107-119 (2013).
- Geiger, A., Lenz, P., Stiller, C., and Urtasun, R., Vision meets robotics: The kitti dataset *The International Journal of Robotics Research* **32**, 11 1231–1237 (2013).
- Gu, Z., Su, X., Liu, Y., and Zhang, Q., Local stereo matching with adaptive supportweight, rank transform and disparity calibration. *Pattern Recogn. Lett.* **29**, 1230–1235 (2008).
- Guo, T., Ardekani, A.M. & Vlachos, P.P. Microscale, scanning defocusing volumetric particle-tracking velocimetry. *Exp Fluids* **60**, 89 (2019).
- Graff, E. C., and Gharib, M., Performance prediction of point-based three-dimensional volumetric measurement systems. *Meas. Sci. Technol.* **19**, 075403 (2008).
- Hain, R., Kähler, C. J., and Tropea, C., Comparison of CCD, CMOS, and intensified cameras. *Exp. Fluids* **42**, 403–11 (2007).
- Hong, J., Toloui, M., Chamorro, L.P., Guala, M., Howard, K., Riley, S., Tucker, J., and Sotiropoulos, F., Natural snowfall reveals large-scale flow structures in the wake of a 2.5-MW wind turbine. *Nature Communications* **5**, 4216 (2014).
- Hong, J., and Abraham, A., Snow-powered research on utility-scale wind turbine flows. *Acta Mechanica Sinica.* **36**, 339–355 (2020).
- Jung, C., Chen, X., Jiji, C., Lei, H., Yun, I., and Kim, J. Boundary-Preserving Stereo Matching with Certain Region Detection and Adaptive Disparity Adjustment. *Journal of Visual Communication and Image Representation* **33**, 1-9 10.1016/j.jvcir.2015.08.010 (2015).
- Kähler, C. J., Scharnowski, S., and Cierpka, C., On the resolution limit of digital particle image velocimetry. *Exp. Fluids* **52**, 1629–39 (2012).
- Kanade, T., Development of a video-rate stereo machine. *Image Understanding Workshop* 549–557 (1994).
- Katz, J., and Sheng, J., Applications of holography in fluid mechanics and particle dynamics. *Annu. Rev. Fluid Mech.* **42**, 531–55 (2010).

- Kellnhofer, P., Didyk, P., Ritschel, T., Masia, B., Myszkowski, K., Deidel, H. P., Motion Parallax in Stereo 3D: Model and Applications. *ACM Transactions on Graphics* **35**(6), 176 (12 pages) (2016).
- Kim, JT., Kim, D., Liberzon, A., and Chamorro, LP., Three-dimensional Particle Tracking Velocimetry for Turbulence Applications: Case of a Jet Flow. *J Vis Exp* (108), 53745. (2016).
- Kitzhofer, J., & Brücker, C., Tomographic particle tracking velocimetry using telecentric imaging. *Experiments in Fluids* **49**, (2010).
- Kooi, F. L., and Toet, A., Visual Comfort of Binocular and 3-D Displays. *Proceedings of SPIE — The International Society for Optical Engineering* **25**(2), 99–108 (2004).
- Kowdle, A., Gallagher, A., and Chen, T., Combining monocular geometric cues with traditional stereo cues for consumer camera stereo. In Fusiello, A., Murino, V., Cucchiara, R. (eds), *Computer Vision - EECV 2012. Workshops and Demonstrations. ECCV 2012. Lecture Notes in Computer Science* **7584**, (2012).
- Landy, M. S., Laurence, T. M., Elizabeth, B. J., and Young, M., Measurement and modeling of depth cue combination: in defense of weak fusion. *Vision Research* **35**, (3) 389-412 (1995).
- Latychevskaia, T., Gehri, F., and Fink, H., Depth-resolved holographic reconstructions by three-dimensional deconvolution. *Opt. Express* **18**, 22527–44 (2010).
- Latychevskaia, T., and Fink, H.W., Holographic time-resolved particle tracking by means of three-dimensional volumetric deconvolution. *Opt. Express* **22**, 1364 (2014).
- Levin, D., The approximation power of moving least-squares. *Mathematics of Computation* *S* 0025-5718(98)00974-0 (1998).
- Lindken, R., Esch, J. V., Westerweel, J., and Wieneke, B., 3D particle imaging for the quantitative characterization of advective microscale mixing. *14th Int. Symp. on Applications of Laser Techniques to Fluid Mechanics* (2008).
- López-Martínez, A.; Cuevas, F.J. Multiple View Relations Using the Teaching and Learning-Based Optimization Algorithm. *Computers* **9**, (4) 101 (2020).
- Lundgren, T.S., Linearly forced isotropic turbulence. *Annual Research Briefs 2003*, Center for Turbulence Research, Stanford, pp. 461-473 (2003).
- Maas, H.-G., Gruen, A., and Papantoniou, D., Particle tracking velocimetry in three-dimensional flows. *Experiments in Fluids* **15**, 133–46 (1993).
- Maki, A., Watanabe, B., and Wiles, C., Geotensity: Combining Motion and Lighting for 3D Surface Reconstruction. *Intl. Conf. Computer Vision* 1053-1060 (1998).

Marshall, J.S., and Li, S., *Adhesive Particle Flow – A Discrete-Element Approach*, Cambridge University Press, New York (2014).

Mather, G., and Smith, D., Combining depth cues: effects upon accuracy and speed of performance in a depth-ordering task. *Vision Research* **44**, (6) 557-562 (2004).

Mikheev, A. V., and Zubtsov, V. M., Enhanced particle-tracking velocimetry (EPTV) with a combined two-component pair-matching algorithm. *Meas. Sci. Technol.* **19**, 085401 (2008).

Nagai, T., Naruse, T., Ikehara, M., and Kurematsu, A., HMM-based surface reconstruction from single images. *Proceedings. International Conference on Image Processing* **2**, 11 (2002).

Ono, H., & Wade, N.J., Depth and motion perceptions produced by motion parallax. *Teaching of Psychology* **33**, 199-202 (2006).

Papantoniou, D., Dracos, T., Lagrangian statistics in open channel flow by 3-D particle tracking velocimetry. Rodi and Ganic (eds.) *Engineering Turbulence Modeling and Experiments. Elsevier* (1990).

Park, J., and Kihm, K., Three-dimensional micro-PTV using deconvolution microscopy. *Experiments in Fluids* 40:491 (2006).

Pereira, F., Gharib, M., Dabiri, D., and Modarress, D. Defocusing digital particle image velocimetry: a 3-component 3-dimensional DPIV measurement technique. Application to bubbly flows. *Experiments in Fluids* **29**(1), S078–S084 (2000).

Romano, M.G., Alberini, F., Liu, L., Simmons, M., and Stitt, E., 3D-PTV flow measurements of Newtonian and non-Newtonian fluid blending in a batch reactor in the transitional regime. *Chemical Engineering Science* **246**, 116969 (2021).

Rosales, C., and Meneveau, C., Linear forcing in numerical simulations of isotropic turbulence: physical space implementations and convergence properties. *Physics of Fluids* **17**(9), 095106 (2005).

Salih, Y., and Malik, A. S., Depth and Geometry from a Single 2D Image Using Triangulation. *IEEE International Conference on Multimedia and Expo Workshops* 511-515 (2012).

Sankur, B., Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging* **13**(1), 146 (2004).

Sarder, P., and Nehorai, A., Deconvolution methods for 3-D fluorescence microscopy images. *IEEE Signal Process Mag.* **23**, 32–45 (2006).

- Satake, S., Anraku, T., Kanamori, H., Kunugi, T., Sato, K., and Ito, T., Measurements of three-dimensional flow in microchannel with complex shape by micro-digital-holographic particle-tracking velocimetry. *J. Heat Transfer* **130**, 042412-1 (2008).
- Sauvola, J., and Pietikäinen, M., Adaptive document image binarization. *Pattern Recognit* **33**, 225–36 (2000).
- Saxena, A., Chung, S. H., and Ng, A. Y., Learning depth from single monocular images. *In NIPS* (2005).
- Scarano, F., Theory of non-isotropic spatial resolution in PIV. *Exp. Fluids* **35**, 268–77 (2003).
- Scharstein, D., Szeliski, R., A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int J Comput Vision* **47**(1), 7–42 (2002).
- Schiller L, and Naumann A. Über die grundlegenden Berechnungen bei der Schwerkraftaufbereitung. *Zeitschrift des Vereines Deutscher Ingenieure* **77**, 318-320 (1933).
- Sun, B., Ahmed, A., Atkinson, C., & Soria, J. A novel 4D digital holographic PIV/PTV (4D-DHPIV/PTV) methodology using iterative predictive inverse reconstruction. *Measurement Science and Technology*, **31**(10), 104002 (2020).
- Suzuki, S., and Abe, K., Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing* **30**(1), 32-46 (1985).
- Takehara, K., and Etoh, T., A study on particle identification in PTV—particle mask correlation method. *J. Visualization* **1**, 313–23 (1999).
- Toloui, M., Mallery, K., and Hong, J., Improvements on digital inline holographic PTV for 3D wall-bounded turbulent flow measurements. *Meas. Sci. Technol.* **28**, 044009 (2017).
- Vincent, A., and Meneguzzi, M., The spatial structure and statistical properties of homogeneous turbulence. *Journal of Fluid Mechanics* **225**, 1-20 (1991).
- Willert, C., and Gharib, M., Three-dimensional particle imaging with a single camera. *Experiments in Fluids* **12**(6), 353–358 (1992).
- Zhao, Y., Ma, X., Zhang, C., Wang, H., and Zhang, Y., 3D real-time volumetric particle tracking velocimetry—A promising tool for studies of airflow around high-rise buildings. *Building and Environment* **178**, 106930 (2020).