

Computational Imaging and Its Applications in Fluids

Dissertation by

Jinhui Xiong

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

King Abdullah University of Science and Technology

Thuwal, Kingdom of Saudi Arabia

Sep, 2021

EXAMINATION COMMITTEE PAGE

The dissertation of Jinhui Xiong is approved by the examination committee

Committee Chairperson: Prof. Wolfgang Heidrich

Committee Members: Prof. Bernard Ghanem, Prof. Peter Wonka, Prof. Konrad Schindler

©Sep, 2021

Jinhui Xiong

All Rights Reserved

ABSTRACT

Computational Imaging and Its Applications in Fluids

Jinhui Xiong

Computational imaging differs from traditional imaging system by integrating an encoded measurement system and a tailored computational algorithm to extract interesting scene features. This dissertation demonstrates two approaches which apply computational imaging methods to the fluid domain.

In the first approach, we study the problem of reconstructing time-varying 3D-3C fluid velocity vector fields. We extend 2D Particle Imaging Velocimetry to three dimensions by encoding depth into color (a “rainbow”). For reconstruction, we derive an image formation model for recovering stationary 3D particle positions. 3D velocity estimation is achieved with a variant of 3D optical flow that accounts for both physical constraints as well as the rainbow image formation model. This velocity field can be used to refine the position estimate by adding physical priors that tie together all the time steps, forming a joint reconstruction scheme.

In the second approach, we study the problem of reconstructing the 3D shape of underwater environments. The distortions from the moving water surface provide a changing parallax for each point on the underwater surface. We utilize this observation by jointly estimating both the underwater geometry and the dynamic shape of the water surface. To this end, we propose a novel differentiable framework to tie together all parameters in an integrated image formation model. To our knowledge, this is the first solution that is capable to simultaneously retrieve the structure of dynamic water surfaces and static underwater scene geometry in the wild.

ACKNOWLEDGMENT

Six years ago, I made one of my most significant and fortunate life-time decision – joining KAUST for my graduate study and working under the guidance of Prof. Wolfgang Heidrich. I am truly grateful that KAUST provides a worry-free environment where I can focus on my research and the rest are well taken good care of. I could not be luckier being advised by Prof. Wolfgang Heidrich, who made all the research outputs possible. With his valuable guidance and suggestions, I could feel full of support when stuck in academic obstacles, which makes my research journey much easier. I also thank Prof. Schindler Konrad, Prof. Bernard Ghanem and Prof. Peter Wonka for serving as the committee members, and providing me with fruitful feedback to my work.

Also, I would like to thank all past and current Imaging Lab members at KAUST, especial thanks to Qiang Fu, Ramzi Idoughi, Xiong Dun, who collaborated with me on a number of research projects. I also appreciate fruitful discussions with many other members in Wolfgang's lab and in KAUST Visual Computing Center. With all your participation, my doctoral life has been enriched.

Furthermore, I have been fortunate to collaborate with Prof. Sigurdur Thoroddsen, Prof. Peter Richtárik, Andres A.Aguirre-Pablo and Abdulrahman Aljedaani. With their guidance and contributions, I could have the opportunity to explore ideas in broader areas. I also wish to express my sincere thanks to Prof. Shree Nayar and Jian Wang for their mentorship when I was an intern in Snap research lab. Working on a wonderful project in Snap makes my doctoral life truly distinctive and memorable.

The last thanks are for my dearest families. I am not afraid of failures because I know you are always standing behind me with consistent support, care and love.

TABLE OF CONTENTS

Examination Committee Page	2
Copyright	3
Abstract	4
Acknowledgment	5
Table of Contents	8
List of Figures	9
List of Tables	11
1 Introduction	12
1.1 Dissertation Structure	15
2 Background and Related Work	17
2.1 Fluid Imaging	17
2.2 Particle Image Velocimetry	18
2.3 Particle Distribution Reconstruction	19
2.4 Velocity Estimation	21
2.5 Transparent Object Reconstruction	22
2.6 Fluid Surface Reconstruction	23
2.7 Structure from Distortion	24
2.8 Optimization Strategy	25
3 Rainbow Particle Imaging Velocimetry for Dense 3D Fluid Velocity Imaging	28
3.1 Introduction	28

3.2 System Overview	31
3.2.1 Optical Setup	31
3.2.2 Reconstruction	32
3.3 Particle Position Estimation	33
3.3.1 Image Formation Model	34
3.3.2 Optimization Problem	34
3.3.3 Particle Sparsity	38
3.3.4 Particle Motion Consistency	38
3.4 Velocity Field Reconstruction	38
3.4.1 Divergence Free	39
3.4.2 Temporal Coherence	40
3.4.3 Optimization Problem	40
3.5 Experimental Setup	43
3.5.1 Rainbow Light Generation	43
3.5.2 Acquisition Device	44
3.5.3 Measured Flows	47
3.6 Velocity Vector Fields Reconstruction Results	47
3.6.1 Synthetic Simulations	48
3.6.2 Experiments with a Ground Truth	50
3.6.3 Experiments without Ground Truth	54
3.7 Conclusion	57
4 Reconfigurable Rainbow PIV for 3D Flow Measurement	60
4.1 Introduction	60
4.2 System Overview	62
4.3 Particle Position Estimation	64
4.4 Volumetric Flow Reconstruction	68
4.5 Experimental Setup	69
4.6 Results	70
4.6.1 Calibration	70
4.6.2 Simulation Results	72
4.6.3 Refocusing	73
4.6.4 Fluid Flow Measurements	75
4.7 Conclusion	76
5 RainbowPIV with Improved Depth Resolution – Design and Comparative Study with TomoPIV	78

5.1 Introduction	78
5.2 System Overview	82
5.3 Depth Super-Resolution RainbowPIV	84
5.4 Experimental Results	87
5.4.1 Validation Experiments	87
5.4.2 RainbowPIV and Tomo-PIV	88
5.4.3 Low Particle Seeding Density	92
5.5 Discussions	94
5.6 Conclusion	95
6 In-the-Wild Single Camera 3D Reconstruction	
Through Moving Water Surfaces	97
6.1 Introduction	97
6.2 System Overview	99
6.3 Differentiable Framework	100
6.4 Method	101
6.4.1 Surface and Scene Representation	101
6.4.2 Multi-Time Triangulation	104
6.4.3 Integrating Ray Casting, Snell's Law and Triangulation	106
6.4.4 Loss Function	108
6.5 Results and Discussions	110
6.5.1 Synthetic Experiments	110
6.5.2 Experiments in the Lab	116
6.5.3 Experiments in the Wild	119
6.6 Discussion	122
6.6.1 Different Initializations	122
6.6.2 Failure Case	123
6.6.3 Relationship to Structure-from-Motion (SfM)	123
6.7 Conclusion	125
7 Concluding Remarks	126
7.1 Summary	126
7.2 Future Work	126
References	130
Appendices	139

LIST OF FIGURES

2.1 Schematic diagram for regular PIV system.	19
3.1 RainbowPIV system overview	31
3.2 The schematic diagram for RainbowPIV system.	32
3.3 Illustration of the experimental rainbowPIV setup.	44
3.4 Comparison of the images acquired using different lens.	45
3.5 Schema for hybrid refractive-diffractive lens.	46
3.6 Numerical comparisons with ground truth data for different algorithms.	48
3.7 Reconstructions for simple analytical flows	49
3.8 Calibrated PSFs for different layers along depth direction.	50
3.9 The reconstructed velocity vector fields.	52
3.10 5 successively captured images in a video frame.	54
3.11 Path line visualization of the vortex data.	55
3.12 Path line visualization for dripping and jet phenomena.	56
3.13 Path line visualization of a complex flow created by stirring the fluid.	57
4.1 Reconfigurable Rainbow PIV setup.	63
4.2 Illustration of the truncated consensus solver.	67
4.3 Illustration of rainbow generation using two parallel blazed gratings.	71
4.4 Reconstruction quality factor and runtime comparison.	73
4.5 Images captured using the encoded DOE.	74
4.6 Flow vector visualizations for the fluid flow in different size.	75
5.1 Schematic diagram for the experimental setup.	81
5.2 Schematic diagram for the vortex ring generator.	83
5.3 Proposed discretization scheme.	85
5.4 Calibrated curves for the normalized camera.	86
5.5 Averaged errors for reconstructed flow velocities.	87
5.6 Experimental setup for both Tomo-PIV and RainbowPIV.	89

5.7 A single captured RainbowPIV image with $\text{ppp} = 0.05$.	89
5.8 Reconstructed flow vectors and vorticity magnitude.	91
5.9 Isosurface visualization for the vorticity magnitude.	92
5.10 Reconstructed flow vectors and vorticity magnitude at low density..	93
5.11 The captured RainbowPIV image at low density.	94
6.1 Pipeline for underwater scene geometry estimation.	100
6.2 An illustration of our setup.	102
6.3 The flowchart of our proposed framework.	106
6.4 The evolution of the objective function versus iterations.	107
6.5 Average Euclidean distance between true and estimated point clouds.	112
6.6 Average Euclidean distance on synthetic data.	113
6.7 Qualitative comparisons between estimated geometry and ground truth.	115
6.8 Comparisons of the estimated underwater scenes.	117
6.9 Visual comparisons with the 3D scanned models.	118
6.10 Qualitative comparisons for water surface estimation.	119
6.11 Reconstructions of two scenes captured in the public fountain.	120
6.12 Qualitative comparisons on the reconstructed underwater scenes.	121
6.13 Reconstructions of two data sets collected in salt water environments.	121
6.14 The reconstructions from different initial points.	122
6.15 A failure case for the fountain scene.	123
7.1 A coral reef scene.	128

LIST OF TABLES

6.1	Average Euclidean distance between true and estimated point clouds.	111
6.2	Quantitative results compared with a multi-camera approach.	114
6.3	Quantitative results between the true and the estimated water surface.	116

Chapter 1

Introduction

Fluid imaging has many significant applications in scientific and engineering fields such as combustion research, design of airplanes and underwater vehicles, and development of artificial heart valves. Since 3D unsteady flows and turbulence are very common in such domains, the main task of the fluid imaging is to allow probing the fluid motions over a range of length scales. In other words, the ultimate goal is to be able to obtain 3D dense measurements of the three components of the velocity vector, known as 3D-3C. It usually needs tailored illumination and imaging systems, and reconstruction algorithms to reconstruct fluid data. These are essentially at the core of computational imaging, integrating the sensing system and the computation to extract information which is otherwise inaccessible.

In recent years, a great amount of effort has been invested into the development of methods for complete volumetric reconstruction of three-dimensional, three-component (3D-3C) velocity vector fields. Tomographic Particle Imaging Velocimetry (Tomo-PIV) [1, 2] has long been considered the standard technology for 3D measurement due to its ability to handle high particle seeding densities and high spatial resolution reconstruction, as well as its robustness to many types of flow phenomena. Recent advances of Tomo-PIV have included improved reconstruction accuracy [3, 4] or spatial and temporal resolution [5, 6] by exploiting temporal information, or reducing the cost of setup, e.g. using smartphones [7, 8]. Tomo-PIV typically makes use of 4-6 cameras capturing the volume of interest from different viewing angles. While Tomo-PIV has the above mentioned advantages, it also suffers from several

limitations that constrain its use. First, a considerable amount of effort is required to set up and calibrate a multi-camera system. Precise calibration is required, to avoid degradation of the reconstruction quality. Depth-of-field is another practical issue limiting its applications; achieving a large depth-of-field requires a small numerical aperture, which leads to low light efficiency (using a powerful light source is expensive and brings potential safety issue). Another severe limitation is that there are many experimental setups where optical access is limited, and thus setting up a multi-camera system becomes impractical. In such situation, a single-camera based 3D-3C technology would be desired.

This dissertation proposes a new approach, RainbowPIV, by combining a suitable setup for color-based encoding of the third dimension in volumetric PIV, as well as a powerful algorithm to retrieve both the particle positions and the velocity vector field. For the hardware part, a linear color filter is employed in order to obtain a continuous wavelength-gradation pattern, i.e. a rainbow illumination. Then, a diffractive optical element (DOE) is attached to the camera objective lens, in order to achieve a wavelength-selective focus that coincides with the rainbow illumination planes. With this setup, particles with different wavelengths (different depths) will be simultaneously in focus on the sensor plane.

In order to fully reconstruct dense 3D-3C velocity fields, prior work utilizes a pipeline approach that first estimates the particle distribution fields at successive time steps, and then reconstructs the corresponding flow fields using the estimated particle distributions. Separating particle distribution field and velocity field reconstructions neglects temporal coherence as a strong physical cue. Specifically, particles present at one time step should also be present at the next as well as the previous time steps (excluding a small number of particles that enter or leave the observation volume), and their location should be consistent with the estimated 3D-3C flow fields. This dissertation, for the first time, proposes a joint optimization framework for particle

distribution fields and fluid velocity vector fields reconstruction on a sequentially captured video frames. This joint optimization framework has also been utilized in the subsequent work, e.g. multi-camera 3D Fluid Flow Estimation [9] and X-ray computed tomographic applications [10], and it exhibits improved reconstruction quality in respective work.

Dynamic fluids are also common phenomena in the wild. Capturing the fluids remains an active research area in computer vision and computer graphics for decades. Little work has shown its ability to recover dynamic fluids in reality. Because of the transparency, complicated setups are always required to achieve the reconstruction. Tracer-free methods have attracted researchers attention as it will not change the appearance of interested fluids. Among tracer-free methods, a known reference pattern is usually placed behind or underneath the fluids, and fluid structure is reconstructed by analyzing the distortion of the pattern, where reflected light paths from the pattern obeys the Snell’s law at a refractive interface. In the literature, high frequency dot patterns are employed for gas flow reconstruction [11], checkboard or color patterns are applied for depth and normal reconstruction of water surface [12, 13, 14], and specific active illumination equipment is also proposed for fluid acquisition [15, 16]. Fundamental limitations restrict their out-of-lab applications in uncontrolled environments: (1) The calibration needs to be done without fluid disturbance, which could be fulfilled in a laboratory environment while always be impracticable outside the lab. (2) The background scene is required to be determined or to be a plane pattern for a reasonable water surface recovery , however it could be in arbitrary geometry in the wild.

Acquiring the 3D scene geometry behind fluids has broad applications in research fields like oceanography, remote sensing and astronomy. One practical application scenario is to monitor the 3D structure of coral reefs. Coral reefs have a global distribution, yet, they are experiencing significant changes due to the global warm-

ing and many other factors. Monitoring the coral reef system is of great ecological and economic values. However, reconstructing the topography of the underwater environments is rather challenging because of water surface undulation. The fluids, functioning as transmitting mediums, are unknown and usually non-stationary. The reflected rays from the scenes are continuously bent by the fluids, and captured images appear to be distorted and vary over time.

This dissertation presents a method for reconstructing the 3D shape of underwater environments from a single, stationary camera placed above the water. We propose a novel differentiable framework, which, to our knowledge, is the first single-camera solution that is capable of simultaneously retrieving the structure of dynamic water surfaces and static underwater scene geometry in the wild.

1.1 Dissertation Structure

The remainder of this dissertation will be structured as follows.

Chapter 2 introduces some background information on computational fluid imaging, transparent object reconstruction and structure-from-distortion related topics.

Chapter 3 introduces a new approach, RainbowPIV, which combines a suitable setup for color-based encoding of the third dimension in volumetric PIV, as well as a powerful algorithm to retrieve both the particle positions and the velocity vector field. The proposed system is evaluated with both simulations and an experimental prototype setup.

Chapter 4 introduces a reconfigurable rainbow PIV system that extends the volume size to a considerable range. We introduce a parallel double-grating system to improve the light efficiency for scalable rainbow generation. A varifocal encoded diffractive lens is designed to accommodate the size of the rainbow illumination, ranging from 15mm to 50mm. We also propose a truncated consensus ADMM algorithm to efficiently reconstruct particle locations. Our algorithm is 5× faster compared to

the state-of-the-art.

Chapter 5 addresses the issue of limited axial resolution, the major drawback of the previously introduced RainbowPIV system. We validate the improved RainbowPIV system with a direct, quantitative comparison to four-camera Tomo-PIV on experimental data. The reconstructed flow vectors of the two approaches exhibit a high degree of consistency, with the RainbowPIV results explicitly guaranteeing physical properties such as divergence free velocity fields for incompressible fluid flows.

Chapter 6 introduces a method for reconstructing the 3D shape of underwater environments from a single, stationary camera placed above the water. We propose a novel differentiable framework to simultaneously retrieve the structure of dynamic water surfaces and static underwater scene geometry in the wild. Experimental results show that our method is able to realize robust and detailed reconstructions on a variety of scenes, both in a laboratory environment and in the wild, and even in a salt water environment.

Chapter 7 concludes with some closing remarks and describes avenues of future research.

Chapter 2

Background and Related Work

2.1 Fluid Imaging

3D fluid imaging in graphics has mostly focused on independent scalar density fields for each time step. Examples for the physical properties recovered in this fashion include the distribution of light emission in flames [17, 18], scattering density in smoke using laser scanning [19] or structured light [20], density of a fluorescent dye in fluid mixtures [21, 22], as well as the refractive index distribution in hot air plumes using the technique of Background Oriented Schlieren [23, 24].

In the recent research work, Zang et al. [25] proposes a framework to recover fluids using sparse camera viewpoints with a dedicated regularizer based on reprojection consistency. Franz et al. [26] further proposes a global transport framework to reconstruct volumetric fluids from sparse views with a learned self-supervision for novel-view synthesis. While this data is sufficient for playback in graphics applications, other interesting applications such as re-simulation, or flow editing require velocity fields instead of just scalar densities. This requires some form of velocity estimation or flow tracking, which is difficult on this kind of input data. While there have been efforts to recover velocities from the captured scalar fields through optical flow-style approaches, and also some work tackles the problem by jointly recover the scalar fields and the fluid flows over a sequential 3D fluid volumes, these attempts have been limited by the relatively small amount of high-frequency texture in the recovered data [22].

2.2 Particle Image Velocimetry

Particle image velocimetry [27, 28, 29] has been widely adopted in the field of fluid mechanics to quantitatively evaluate fluid flow structures. PIV is an optical approach to visualize and measure the instantaneous and non-stationary fluid flow velocity fields within a planar slice of the volume, as seen in Figure 2.1. The fluids are seeded with sufficiently small tracking particles, yet, capable of scattering enough illuminating laser light, such that the movement of the particles will follow that of the fluids dynamics. By illuminating those small particles to make it visible, the velocity vector field of the fluids flow can be measured by detecting and calculating the motion of particles.

Typically, PIV consists of two steps: recording illuminated particles within the interested planar plane in a manner of single-frame multi-exposure or multi-frame single-exposure, and then applying auto-correlation or cross-correlation algorithms for calculating the spatial distribution of flow velocities. While such approach can only measure two components of the flow vectors of a planar slice of the volume, failing to detect out-of-plane motions.

Some examples of 3D extensions of PIV include holographic PIV [30, 31], which works with coherent light, and tomographic PIV (Tomo-PIV) [1, 2]. Tomo-PIV has long been considered the standard technology for 3D measurement. Tomo-PIV typically makes use of 4-6 cameras capturing the volume of interest from different viewing angles. In general, it is able to handle high particle seeding densities and high spatial resolution reconstruction, as well as robustness to many types of flow phenomena. Recent advances of Tomo-PIV have included improved reconstruction accuracy [3, 4] or spatial and temporal resolution [5, 6] by exploiting temporal information, or reducing the cost of setup, e.g. using smartphones [7, 8].

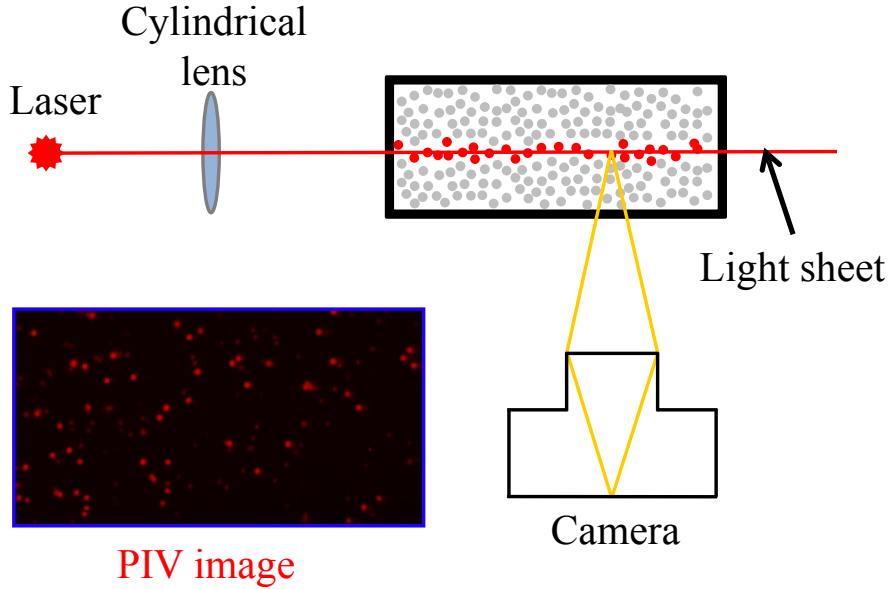


Figure 2.1: Schematic diagram for regular PIV system. A single plane of the volume is illuminated by a light sheet, and a camera focusing on this plane is able to track particles moving within it. This yields 2 components of the velocity field on a 2D slice of the volume.

2.3 Particle Distribution Reconstruction

3D particle reconstruction is an alternative to imaging continuous densities, and is used by 3D variants of PIV. The task of the particle reconstruction is to determine the 3D location of particles from one or more camera views. The total number of cameras in these settings is usually very limited due to space constraints, as well as occlusions by solids, and is typically orders of magnitude lower than the number of projections in x-ray tomography, for example. Another practical issue is depth of field of the cameras, since the whole volume needs to be in focus simultaneously, and the camera aperture usually has to be large to collect enough light to capture fast flows.

To get the 3D particle reconstruction, instead of using multiple cameras for perspective reconstruction, single-camera based approaches could drastically simplify the hardware setup. Willert et al. [32] used a three-pin-hole mask to decode illuminated

particles such that the three-dimensional positions of each particle can be retrieved from the image patterns on the observed image via a defocus analysis. Since three dots would appear in the image for each particle, this method is stuck with a low particle seeding density. Another group of approaches made use of plenoptic cameras [33, 34], which capture the full 4D light field. Particle positions can be reconstructed using ray tracing based algorithms. The idea of applying such technology for measuring volumetric particle distributions has been discussed by [35]. However, due to the existence of ghost particles originated from reconstruction algorithm and reduced spatial resolution, it becomes difficult to reveal particle locations with relatively high accuracy. This can also be seen as limited angle tomography with a very narrow cone of observations.

Instead of modifying the camera side, another class of volumetric particle reconstruction approaches relies on modifying the illumination method, providing additional information on the relative depth, as seen from the camera, by encoding it in color. For this purpose different illumination methods were used: prism [36], laser [37], color filter [38], LCD projector [39, 40]. Herein, the locations of the particles in the volume can be determined by their 2D spatial positions and by the colors in the captured image using a mapping between color and depth position. The primary advantage of this setup compared to other 3D PIV methods is its simplicity. Nevertheless, the presence of random noise, optical aberrations and focus issues, color contamination caused by secondary light scattering from the particles, and color mixing for overlapping particles severely complicate the identification of the representative colors for every possible particle in the observed image. McGregor et al. [37] used a method based on a calibration curve relating the hue of acquired images to the depth of particles within the imaged volume. Watamura et al. [39] proposed an algorithm to calculate particle's representative color by averaging hue values of the pixels where the particle is projected on in polar coordinate. Even though it revealed

promising results for Particle Tracking Velocimetry (PTV), where low density particles are seeded, it will fail for tasks of measuring a dense set of velocity fields, where sufficiently high density of particles is required.

2.4 Velocity Estimation

Velocity estimation from particle fields has been elaborately studied not only in the field of fluid mechanics, but also in the computer vision community. Literature from the fluid mechanics field mainly adopts correlation-based algorithms [41] for global velocity measurement, which computes the spatial auto-correlation or cross-correlation of successive images, extracting average motion at every single interrogation spot. Though significant improvements have been made on correlation methods [42], they still have issues in areas of low particle density, which is common in 3D measurements.

In a seminal result from computer vision, Horn and Schunck [43], proposed a global variational optical flow method based on the assumption of brightness constancy and smoothness in the flow vector. The connection between optical flow and fluid flow was investigated by Liu and Shen [44], which revealed that under certain conditions (mass conservation, inviscid), brightness constancy is equivalent to scalar transport equation for fluid flow. This connection lends a physical meaning to optical flow approaches for fluid tracking. Heitz et al. [45] gave an overview to the applications of optical flow based fluid motion estimation. The estimation accuracy between optical flow and correlation approaches applied to PIV system has been numerically evaluated [46].

Since the optical flow problem is physically connected to the continuity equation in fluid dynamics, it becomes feasible to introduce Navier-Stokes equations, which govern real-world fluid motions, as additional physical priors into the conventional Horn-Schunck algorithm. Some previous literature has taken divergence-free constraints into account [47, 48, 49], while most of them suffer from the complexity of solving higher order regularization terms. Gregson et al. [22] simplify this issue by

connecting the pressure projection method with the proximal operator, allowing it to be easily handled by a convex optimization framework. Ruhnau et al. [49] and Heitz et al. [50] also consider the equation for time evolution of fluid flow, imposing temporal consistency. In this dissertation, we adopt ideas from fluid simulation [51, 52, 53], which approximately solve the time-evolution of fluid flow. This enables us to integrate the temporal coherence regularization terms into the optical flow model, which can then be solved by a modular optimization framework.

2.5 Transparent Object Reconstruction

The reconstruction of transparent objects is complicated by the change in light direction at the object interface due to refraction [54]. Conventional multi-view stereo vision, designed for diffuse objects with Lambertian reflection, is not applicable to these types of objects. Recently, various approaches have been proposed for rigid transparent object reconstruction. Most of the work is realized with specialized hardware setups, for instance light field probes are proposed to capture the changes of the refractive index field [55], a Time-of-Flight camera is used to measure the distorted depth based on the varying speeds of light in transmission mediums with different refractive indexes [56], a tomographic camera system [57], variable illuminations [58], a specialized water tank setup to alter light paths [59], or coded patterns to illuminate the scene and a turntable to realize diverse viewpoints [60, 61] are proposed. Li et al. [62] propose a learning-based strategy for the transparent shape recovery. They use a rendering layer to model the imaging process of refraction and reflection with arbitrary environment maps, however, the background environments must also be measured ahead of time for correspondence estimation.

2.6 Fluid Surface Reconstruction

Many fluids are special types of transparent objects, and they are usually non-stationary. Time-resolved recovery of fluid structures can be realized by tracing the motions of the immersed tracers in the fluids. In the literature, the methods for reconstructing image phenomena, e.g. smoke [63, 64], dye [21, 22] and particles [65, 66], have been developed. The use of particles (known as particle imaging velocimetry), is a standard approach in the field of fluid mechanics to fully characterize the 2D or 3D fluid flows [27] as described in Sec. 2.2.

A variety of non-intrusive approaches have also been proposed to estimate the shape of fluid surface by analyzing the distortions of background patterns. The problem of reconstructing time-varying inhomogeneous refractive index distributions have been addressed in [11, 67]. Dedicated optical setups with active illuminations are presented for acquiring fluid surface structures [15, 16]. Morris et al. [12] extend the traditional multi-view triangulation to be appropriate for refractive scenes, and build up a stereo setup for water surface recovery. A learning-based single-image approach has recently been presented for recovering dynamic fluid surfaces [14]. Like reconstructing rigid transparent objects, the above mentioned work requires an undistorted reference image of the background patterns or known reference patterns to construct a ray-ray correspondence. Qian et al. [68] build a 3×3 camera array and exploit the correspondence information from multiple viewpoints to estimate both the water surface and the underwater scenes.

Reconstructing refractive surfaces is also related to specular object reconstruction [54, 69, 70, 71] and image restoration from refractive distortion [72, 73, 74, 75].

2.7 Structure from Distortion

Optical distortion can be seen in many places in reality. As previously described, transparent objects made of glasses or plastics, non-stationary water surfaces or hot airflow can bend the light rays passing through them and cause distorted patterns from the camera view. The shape of the transparent objects could be retrieved by measuring the ray deflection. Accordingly, this deflection provides different viewpoints of the background scenes, which allows for triangulation of the depth information.

By the fact that the transparent object itself is complicated to reconstruct, seminal work imposes strong assumptions when constructing depth cues from distorted images to 3D coordinates of the scene points. Tian et al. [76] extract the depth of the scenes from the fluctuation of projected image pixels measured by a fixed camera. Similarly, Alterman et al. [77] exploit refractive distortions of a stereo setup to yield a position likelihood of the object via stochastic triangulation. These statistical approaches assume that the fluctuation of the distorted patterns is random over time. Knowing that light paths are bent by the water surface via Snell’s law when crossing water-air interface, the time-varying fluid structures cannot be determined from their approaches. Zhang et al. [78] reconstruct fluid surface and immersed scene structures by analyzing the cues of distortion and defocus. Their method requires an undistorted reference image, which is inaccessible outside the lab. Moreover, they assume the surface normal to be the same for surface areas where the defocus patterns are back-projected to, which does not hold for real fluids. Julian et al. [79] propose to extract the scene depth by looking through a wetted window, where each water drop provides a distorted view of the scene. Their approach estimates the structure of water drops and pixel-to-ray mappings, while an assumption of a low-parameter model is imposed on the water drops. Fully characterizing the water drops is as challenge as reconstructing transparent objects as described in Sec. 2.5. In comparison,

we could realize full characterizations on both the background scene geometry and time-varying water surfaces.

Inferring depth from distorted views shares some similarities to structure-from-motion (SfM) [80]. SfM estimates the 3D structure from a sequence of images captured from different viewpoints. The parameters of the camera and the scene geometry are jointly refined by the scheme of bundle adjustment.

2.8 Optimization Strategy

It has been a consensus to formulate the inverse problem in computational imaging into a framework of convex optimization, consisting of data fitting term and regularization terms. A wide selection of optimization methods is available in the literature, including generic approaches such as Gradient Descent and Quasi-Newton. However, complex optimization problems involving a data term and sophisticated regularizers require a more tailored approach, especially if some terms of the objective function are non-smooth. Splitting-based approaches overcome this issue by decoupling joint minimization problem into multiple inner-connected subproblems, alternatively handling each of them.

Mathematically, the related minimization problem can be expressed as:

$$\mathbf{x} = \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x}) + h(\mathbf{G}\mathbf{x}), \quad (2.1)$$

where $f(\cdot)$ and $h(\cdot)$ are convex functions. The splitting strategy is implemented by introducing a slack variable \mathbf{z} , and formulate a constrained optimization problem:

$$\begin{aligned} \mathbf{x}, \mathbf{z} &= \operatorname{argmin}_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}) + h(\mathbf{z}) \\ \text{subject to} \quad \mathbf{G}\mathbf{x} &= \mathbf{z}. \end{aligned} \quad (2.2)$$

This splitting allows a variant of algorithms to solve it, such as Pock-Chambolle

algorithm [81], split Bregman method [82], half-quadratic splitting [83], alternating direction method of multipliers (ADMM) [84]. They essentially share the similar idea that is to construct a decomposition-coordinate procedure, where the solution to the global problem is guided by the solutions to small subproblems. In this dissertation, we will adopt ADMM, which is an augmented Lagrangian method accompanied with dual decomposition.

The augmented Lagrangian of Equation 2.2 can be written as:

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \lambda) = f(\mathbf{x}) + h(\mathbf{z}) + \lambda^T(\mathbf{G}\mathbf{x} - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{G}\mathbf{x} - \mathbf{z}\|_2^2, \quad (2.3)$$

where λ is the Lagrangian multiplier and $\rho > 0$. We can then formulate it into ADMM framework as following.

Algorithm 1 ADMM Framework for Equation 2.2

```

1: for  $k = 1$  to  $K$  do
2:   // x-minimization step
3:    $\mathbf{x}^{k+1} \leftarrow \operatorname{argmin}_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{z}^k, \lambda^k)$ 
4:   // z-minimization step
5:    $\mathbf{z}^{k+1} \leftarrow \operatorname{argmin}_{\mathbf{z}} \mathcal{L}(\mathbf{x}^{k+1}, \mathbf{z}, \lambda^k)$ 
6:   // scaled dual variables update
7:    $\lambda^{k+1} \leftarrow \lambda^k + \rho(\mathbf{G}\mathbf{x}^{k+1} - \mathbf{z}^{k+1})$ 
8: end for
```

The major benefit for this splitting can be revealed in Algorithm 1, where two functions in the original problem described in Equation 2.1 are separated, forming two unconstrained convex minimization problems, which can be alternatively solved in a relative easy way. The second subproblem can be further analyzed by defining the *proximal operator* [85]. The proximal operator of a function f is given by:

$$\operatorname{prox}_{\sigma f}(\mathbf{v}) = \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x}) + \frac{1}{2\sigma}\|\mathbf{x} - \mathbf{v}\|_2^2. \quad (2.4)$$

It seems to be trivial to introduce the proximal operator here, while it gives a better

generalization to the algorithm, as a new objective function can be directly applied by just deriving the proximal operator of it or a variety of priors can be evaluated by simply changing corresponding proximal operators.

Chapter 3

Rainbow Particle Imaging Velocimetry for Dense 3D Fluid Velocity Imaging

This chapter closely follows Xiong et al. [65].

Despite significant recent progress, dense, time-resolved imaging of complex, non-stationary 3D flow velocities remains an elusive goal. In this chapter we tackle this problem by extending an established 2D method, Particle Imaging Velocimetry, to three dimensions by encoding depth into color. The encoding is achieved by illuminating the flow volume with a continuum of light planes (a “rainbow”), such that each depth corresponds to a specific wavelength of light. A diffractive component in the camera optics ensures that all planes are in focus simultaneously. With this setup, a single color camera is sufficient for tracking 3D trajectories of particles by combining 2D spatial and 1D color information.

For reconstruction, we derive an image formation model for recovering stationary 3D particle positions. 3D velocity estimation is achieved with a variant of 3D optical flow that accounts for both physical constraints as well as the rainbow image formation model. We evaluate our method with both simulations and an experimental prototype setup.

3.1 Introduction

Fluid capture is an active research area in computer graphics. Recent works include efforts to image phenomena such as flames [17, 18], smoke [19, 20], transparent hot

air flows [23], and fluid mixtures [21]. While these methods recover dense volumetric reconstructions, they only yield independent scalar fields of the density of the respective phenomenon at each time step. To fully characterize the 3D flow and open up applications beyond simple play-back, 3D velocity fields need to be recovered as well. While there have been efforts to recover velocities from the captured scalar fields through optical flow-style approaches, these attempts have been limited by the relatively small amount of high-frequency texture in the recovered data [22]. The velocity field of the fluid flows are essentially needed. In other words, the ultimate goal is to be able to obtain 3D dense measurements of the three components of the velocity vector, known as 3D-3C.

Over the last decades, different imaging techniques have been developed to get closer to this goal. Particle Imaging Velocimetry (PIV) is the most commonly used of these techniques [86, 28]. For PIV, small density-matched tracer particles are inserted into the flow, and their advected motion is tracked with image correlation methods, i.e. optical flow. In basic 2D PIV [87], this tracking is made possible by illuminating the volume with a light sheet perpendicular to the camera line of sight (Fig. 2.1). Particles within that plane can be identified easily, and tracked over time, so long as the flow does not move them out of plane. This yields dense measurements of two components of the velocity field on a two-dimensional slice of the volume (2D-2C). Although 3D extensions such as holographic PIV [30] or tomographic PIV [1] exist, a dense reconstruction of all three components of the velocity field over the full 3D volume requires multiple cameras and remains elusive in practice. The densest volume measurements involve high-speed imaging in combination with scanning laser-volumes [88].

This chapter proposes a new approach, RainbowPIV, by combining a suitable setup for color-based encoding of the third dimension in volumetric PIV, as well as a powerful algorithm to retrieve both the particle positions and the velocity vector

field. For the hardware part, a linear color filter is employed in order to obtain a continuous wavelength-gradation pattern, i.e. a rainbow illumination. Then, a diffractive optical element (DOE) is attached to the camera objective lens, in order to achieve a wavelength-selective focus that coincides with the rainbow illumination planes (Fig. 3.2). With this setup, particles with different wavelengths (different depths) will be simultaneously in focus on the sensor plane.

The reconstruction algorithm utilizes a detailed image formation model for this setup to retrieve the 3D location of particles in each frame. From a sequence of successive frames, the velocity vector field is reconstructed using an optical flow approach [43, 89], where physical constraints (incompressibility and temporal consistency) are introduced. In order to improve the obtained results, we can iterate between position and velocity estimation, effectively solving a joint optimization problem for both. The specific contributions of this chapter are:

- We propose a simple PIV setup (RainbowPIV) for measuring time-varying 3D-3C fluid velocity vector fields using a single camera.
- We design a hybrid refractive-diffractive optical system in order to focus all wavelength on the same sensor plane, extending the depth-of-field while preserving high lateral resolution.
- We formulate an image formation model for 3D particle distribution reconstruction, and apply optimization strategies to tackle the ill-posed inverse problem.
- We introduce a physically constrained optical flow method for recovering the fluid velocity fields, and evaluate its effectiveness on synthetic data. Our approach allows having a good estimation of velocity over the measurement volume (high concentration of particles).
- We demonstrate our proposed hardware setup and algorithms on real fluid flows.

3.2 System Overview

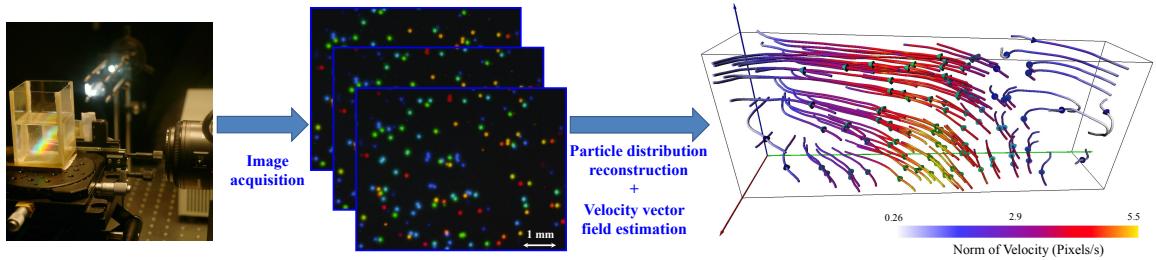


Figure 3.1: Using a rainbow color-coded PIV setup in combination with a hybrid diffractive/refractive camera optics(left) we can encode 3D particle positions in fluid flows into a single camera image, while keeping all particles in focus simultaneously (center). From a sequence of such images 3D particle positions and the dense fluid vector field can be reconstructed using an optimization-based approach. The vector field is represented as a regular grid covering the entire flow volume, and can be visualized by showing the path lines of new (synthetic) particles (right).

The RainbowPIV method consists of two components: a new optical setup that encodes particle depth into a color image with a large depth of field, and a matching new reconstruction algorithm that jointly optimizes particle position and velocity field. The general process are illustrated in Fig 3.1

3.2.1 Optical Setup

A schematic view of the RainbowPIV setup is shown in Figure 3.2, where the illumination is provided by a white light source that is collimated and filtered so that the wavelength varies linearly with the depth within the flow volume. In this setup, particles submersed in the fluid can be modeled as narrow-band point lights, whose wavelength varies linearly with depth. By comparing with the regular PIV setup as shown in Figure 2.1, the advantage of RainbowPIV is revealed that higher dimensional components of the velocity vector fields can be retrieved in the proposed setup.

The second part of the optical setup is a diffractive optical element, specifically a Fresnel phase plate, which provides a wavelength-selective focus. The optical system is designed such that the camera focus for each wavelength corresponds to the depth

at which that wavelength occurs in the rainbow illumination. This design achieves all-in-focus imaging of the particles in the interrogation volume.

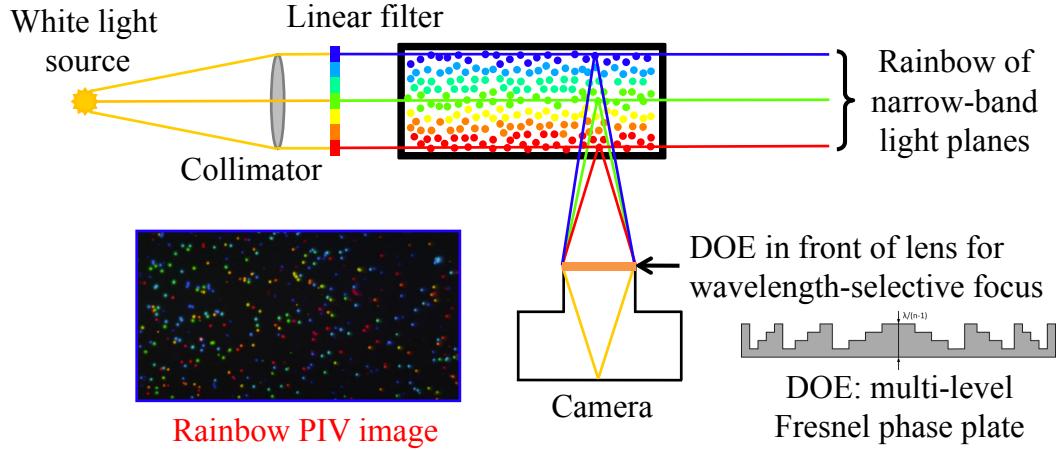


Figure 3.2: The schematic diagram for RainbowPIV system. The 3D volume is illuminated by a continuum of planes, where the wavelength of the illumination varies linearly with the depth. Particles in the volume can thus be modeled as narrow-band point sources, and a diffractive optical element attached to the camera lens ensures that for each wavelength the camera is focused at the appropriate depth.

3.2.2 Reconstruction

The reconstruction task is to estimate particle positions from the the observed color, and then track these particles over time to obtain a 3D velocity field to get a full 3D, 3 component (3D-3C) measurement. This task is made more complicated by the fact that the camera captures only RGB information, not a full hyperspectral image, which makes the position reconstruction less robust. To tackle this problem, we employ an iterative approach: an initial position estimate for each time step can be used to obtain a first estimate of the time-dependent velocity field. This velocity field can be used to refine the position estimate by adding physical priors that tie together all the time steps. These two iterative steps can be expressed in a joint-optimization problem:

$$\begin{aligned}
(\mathbf{p}^*, \mathbf{u}^*) = \operatorname{argmin}_{\mathbf{p}, \mathbf{u}} & \frac{1}{2} \left\| \mathbf{A} \begin{bmatrix} \mathbf{p}_1 | \dots | \mathbf{p}_T \end{bmatrix} - \begin{bmatrix} \mathbf{i}_1 | \dots | \mathbf{i}_T \end{bmatrix} \right\|_2^2 \\
& + \kappa_1 \| \operatorname{diag}(\mathbf{w}) (\mathbf{p}_1; \dots; \mathbf{p}_T) \|_1 + \Pi_{[0,1]} (\mathbf{p}_1; \dots; \mathbf{p}_T) \\
& + \kappa_2 \sum_{t=1}^T \int_{\Omega} \mathbf{p}_t \odot (\mathbf{p}_t - \mathbf{p}_{t+1}(\mathbf{u}_t, -\Delta t))^{\circ 2} d\Omega \\
& + \kappa_3 \sum_{t=1}^T \|\nabla \mathbf{u}_t\|_2^2 + \sum_{t=1}^T \Pi_{C_{\text{DIV}}}(\mathbf{u}_t) \\
& + \kappa_4 \sum_{t=1}^T \left(\|\mathbf{M}(\mathbf{u}_t - \Pi_{C_{\text{DIV}}}(\mathbf{u}_{t-1}(\mathbf{u}_{t-1}, \Delta t)))\|_2^2 \right. \\
& \left. + \|\mathbf{M}(\mathbf{u}_t - \Pi_{C_{\text{DIV}}}(\mathbf{u}_{t+1}(\mathbf{u}_t, -\Delta t)))\|_2^2 \right),
\end{aligned}$$

each term of which will be discussed in Section 3.3 and 3.4. The first three lines correspond to particle position estimation and the latter four lines correspond to velocity field reconstruction, and these two sub-problems are coupled with each other by the term of particle motion consistency (see Section 3.3.4), referring to term in the third line. In each sub-step of every iteration, one variable will be fixed such that the constructed bi-convex problem is reduced to simpler convex optimization problems. The following sections will explain the above two sub-problems respectively.

3.3 Particle Position Estimation

An inverse problem is proposed for recovering particle locations in 3D spatial domain. We start by introducing an image formation model that relates the obtained particle positions to the observed image. Three regularization terms are then added to formulate an optimization problem, which can be efficiently solved with guaranteed convergence, tackling our ill-posed inverse problem.

3.3.1 Image Formation Model

As mentioned above, the illumination in the volume is designed to consist of a continuum of light sheets (Figure 3.2) with a narrow-band spectrum, whose wavelength (denoted as λ) varies with depth (z coordinate). In this chapter, we restrict ourselves to a linear relationship between z and λ since this setting is easily implemented with off-the-shelf components. Therefore, the location of particles in the volume can be geometrically represented as the position of light plane specified by wavelength and pixel positions in that light plane $(\mathbf{x}, \lambda) = (x, y, \lambda)$. The presence of a particle at a specific point in the volume is modeled as an *occupancy probability* $P(\mathbf{x}, \lambda)$.

Since we are operating with incoherent light, the imaging process of the optical system can be modeled as a set of point spread functions (PSF), one for each color channel: $g_C(\mathbf{x}, \lambda)$, where $C \in \{\text{red, green, blue}\}$. With these definitions, the image formation model is

$$i_C(\mathbf{x}) = \int_{\Lambda} \int_{\mathbf{x}} g_C(\mathbf{x} - \mathbf{x}', \lambda) \cdot i_r(\mathbf{x}, \lambda) \cdot P(\mathbf{x}, \lambda) d\mathbf{x}' d\lambda, \quad (3.1)$$

where $i_C(\mathbf{x})$ are the color channels of the captured RGB image, and $i_r(\mathbf{x}, \lambda)$ is the corresponding spectral distribution incident on the image sensor. The spatial integral corresponds to a convolution representing potentially imperfect focus, while the wavelength integral represents the conversion from a spectral image to an RGB image encoding 3D particle positions.

3.3.2 Optimization Problem

After discretization, we can formulate the convolution of PSFs and reflected light intensity as a matrix $\mathbf{A} \in \mathbb{R}^{3N \times NL}$, where N is the number of image pixels, L is the number of discretization levels along the wavelength coordinate, and the value of 3 refers to three color channels. Moreover, $\mathbf{i}_t \in \mathbb{R}^{3N}$ represents the observed image at

time t and $\mathbf{p}_t \in \mathbb{R}^{NL}$ is the occupancy probability of a specific voxel, given \mathbf{i}_t . Hence, the distribution of particles at each time step of a video can be retrieved by solving the linear system

$$\mathbf{A}\mathbf{p}_t = \mathbf{i}_t. \quad (3.2)$$

However, this inverse problem is ill-posed as we have compressed the full spectral information encoding the particle position into just three color channels. To handle this ill-conditioned inverse problem, some prior knowledge of the distribution of particles is introduced as regularization terms, resulting the following minimization problem:

$$\begin{aligned} (\mathbf{p}^*) = \operatorname{argmin}_{\mathbf{p}} & \frac{1}{2} \left\| \mathbf{A} \begin{bmatrix} \mathbf{p}_1 | \dots | \mathbf{p}_T \end{bmatrix} - \begin{bmatrix} \mathbf{i}_1 | \dots | \mathbf{i}_T \end{bmatrix} \right\|_2^2 \\ & + \kappa_1 \| \operatorname{diag}(\mathbf{w}) (\mathbf{p}_1; \dots; \mathbf{p}_T) \|_1 + \Pi_{[0,1]} (\mathbf{p}_1; \dots; \mathbf{p}_T) \\ & + \kappa_2 \sum_{t=1}^T \int_{\Omega} \mathbf{p}_t \odot (\mathbf{p}_t - \mathbf{p}_{t+1}(\mathbf{u}_t, -\Delta t))^{\circ 2} d\Omega, \end{aligned} \quad (3.3)$$

where \odot and $(\cdot)^{\circ 2}$ respectively refer to the operators for the Hadamard (i.e. component-wise) product and square operator. The operator $\Pi_{[0,1]}$ projects all volume occupancy probabilities onto the convex set of valid probabilities $[0, 1]^{NL}$.

The first line in Equation 3.3 is a least-square data fitting term corresponding to Equation 3.2. The second line defines a weighted L_1 term that encourages sparse distributions of particles in the volume, and the indicator function enforces that occupancy probabilities are between zero and one. Finally, the term of the third line provides temporal coherence by mandating that occupancy probabilities of successive time frames are consistent with advection under a previously estimated flow field $\mathbf{u}_t = (u_t, v_t, w_t)$ by $-\Delta t$ units of time, expressed as $\mathbf{p}_{t+1}(\mathbf{u}_t, -\Delta t)$. We call this term the *particle motion consistency* term, and it allows for refining position estimates once a velocity field has been estimated, and ties the reconstruction of all frames together

into a single optimization problem. The particle motion consistency term is discussed in more detail below.

The above optimization problem is non-smooth because of the L_1 term and the indicator function, hence it cannot be solved by general optimization tools such as gradient descent. The strategy tackling this kind of issue is to decouple non-smooth terms from the original optimization problem, such that distinct parts can be handled separately. We apply this strategy using the ADMM framework which is systematically discussed in [84].

Algorithm 2 ADMM Framework of Computing Particle Distribution

```

1: procedure COMPUTEPARTICLELOCATION( $F_1, H_1$ )
2:   for from 1 to maximum ADMM iteration do
3:     // p-minimization step
4:      $\mathbf{p}^{j+1} \leftarrow \text{prox}_{\sigma_1 F_1}(\mathbf{z}^j - \mathbf{q}^j)$ 
5:     // z-minimization step
6:      $\mathbf{z}^{j+1} \leftarrow \text{prox}_{\tau_1 H_1}(\mathbf{p}^{j+1} + \mathbf{q}^j)$ 
7:     // scaled dual variables update
8:      $\mathbf{q}^{j+1} \leftarrow \mathbf{q}^j + \mathbf{p}^{j+1} - \mathbf{z}^{j+1}$ 
9:   end for
10: end procedure

```

The pseudo code for solving Equation 3.3 using ADMM is shown in Algorithm 2, where j is the iteration number, \mathbf{z} is a slack variable, \mathbf{q} is a dual variable, or Lagrange multiplier. $\text{prox}_{\sigma_1 F_1}$ and $\text{prox}_{\tau_1 H_1}$ are proximal operators [85] based on F_1 and H_1 respectively. F_1 and H_1 are defined as:

$$\begin{aligned}
F_1(\mathbf{p}) &= \frac{1}{2} \left\| \mathbf{A} \begin{bmatrix} \mathbf{p}_1 | \dots | \mathbf{p}_T \end{bmatrix} - \begin{bmatrix} \mathbf{i}_1 | \dots | \mathbf{i}_T \end{bmatrix} \right\|_2^2 \\
&\quad + \kappa_2 \sum_{t=1}^T \int_{\Omega} \mathbf{p}_t \odot (\mathbf{p}_t - \mathbf{p}_{t+1}(\mathbf{u}_t, -\Delta t))^{\circ 2} d\Omega
\end{aligned} \tag{3.4}$$

$$H_1(\mathbf{p}) = \kappa_1 \| \text{diag}(\mathbf{w})(\mathbf{p}_1; \dots; \mathbf{p}_T) \|_1 + \Pi_{[0,1]}(\mathbf{p}_1; \dots; \mathbf{p}_T) \tag{3.5}$$

The derivation of the proximal operators F_1 is given as follows:

$$\mathbf{p} = \mathbf{prox}_{\sigma_1 F_1}(\mathbf{d}) \Rightarrow (3.6)$$

$$\begin{bmatrix} \sigma_1(\mathbf{A}^T(\mathbf{A}\mathbf{p}_1) + f_{1,2}(\mathbf{p})) + \mathbf{p}_1 \\ \vdots \\ \sigma_1(\mathbf{A}^T(\mathbf{A}\mathbf{p}_t) + f_{t,t-1} + f_{t,t+1}) + \mathbf{p}_t \\ \vdots \\ \sigma_1(\mathbf{A}^T(\mathbf{A}\mathbf{p}_T) + f_{T,T-1}) + \mathbf{p}_T \end{bmatrix} = \begin{bmatrix} \sigma_1 \mathbf{A}^T \mathbf{i}_1 + \mathbf{d}_1^j \\ \vdots \\ \sigma_1 \mathbf{A}^T \mathbf{i}_t + \mathbf{d}_t^j \\ \vdots \\ \sigma_1 \mathbf{A}^T \mathbf{i}_T + \mathbf{d}_T^j \end{bmatrix}$$

where

$$\begin{aligned} f_{t,t+1}(\mathbf{p}) &= 2\kappa_2 \mathbf{p}_t \odot (\mathbf{p}_t - \hat{\mathbf{p}}_{t+1}^-) + \kappa_2 (\mathbf{p}_t - \hat{\mathbf{p}}_{t+1}^-)^{\circ 2}, \\ f_{t,t-1}(\mathbf{p}) &= 2\kappa_2 \mathbf{p}_{t-1} \odot (\hat{\mathbf{p}}_t^- - \mathbf{p}_{t-1}). \end{aligned}$$

To simplify the notations, we denote $\mathbf{z}^j - \mathbf{q}^j$ as \mathbf{d}^j , $\mathbf{p}^{j+1} + \mathbf{q}^j$ as \mathbf{e}^j , $\mathbf{u}_t^{j+1} + \mathbf{q}^j$ as \mathbf{h}^j , and $\mathbf{p}_{t+1}(\mathbf{u}_t, -\Delta t)$ as $\hat{\mathbf{p}}_{t+1}^-$. In this term, \mathbf{p} is represented by procedural operator on the left hand side of Equation 3.6, and it is solved by Conjugate Gradients.

The derivation of the proximal operators H_1 is given as follows:

$$\begin{aligned} \mathbf{z} &= \mathbf{prox}_{\tau_1 H_1}(\mathbf{e}^j) \Leftrightarrow \\ \mathbf{z} &= \Pi_{[0,1]}((\mathbf{e}^j - \tau_1 \kappa_1 \mathbf{w})_+ - (-\mathbf{e}^j - \tau_1 \kappa_1 \mathbf{w})_+). \end{aligned} (3.7)$$

This term is the point-wise shrinkage operation followed by a projection onto the domain of $[0, 1]$.

3.3.3 Particle Sparsity

The L_1 penalized term ensures a sparse distribution of particles in the volume. It is further weighted by a diagonal matrix $diag(\mathbf{w})$. Unlike the algorithm proposed in [90], which iteratively changes the weight coefficients based on previous results for enhancing sparsity, weights in our approach are fixed during iterations, but vary with particle depth. The motivation for this process is to compensate for different sensitivities of the camera to different wavelengths. For example, wavelengths in the yellow or in the blue-green part of the spectrum elicit a strong response in two or even three color channels, while wavelengths in the far blue or far red parts only trigger one channel. This can result in a non-uniform particle distribution, where particles are more likely to be placed on certain preferred depths. The weighting term allows us to eliminate this bias by compensating for the photometric non-uniformity.

3.3.4 Particle Motion Consistency

As mentioned, particle motion consistency ensures that estimated particle locations in successive frames are consistent with advection through a previously estimated flow field. This turns the position estimation from a set of independent problems, one for each time step, to a single joint estimation problem for the whole sequence. This term can be improved by adding a mask to suppress the impact of low confidence flow estimates.

3.4 Velocity Field Reconstruction

This section describes how we estimate the fluid flow vectors from reconstructed 3D particle distributions in a video frame. First, we introduce the physical properties of fluid flow formulated in Navier-Stokes equations, and then an optimization problem is constructed by combining conventional optical flow with those physical constraints.

3.4.1 Divergence Free

An incompressible flow can be described as a solenoidal flow vector field \mathbf{u}_{sol} , which is divergence free:

$$\nabla \cdot \mathbf{u}_{\text{sol}} = 0. \quad (3.8)$$

Based on the Helmholtz decomposition, any arbitrary vector field \mathbf{u} (in our case an intermediate flow vector obtained that does not satisfy the divergence-free constraints) can be decomposed into a solenoidal (divergence-free) part and an irrotational (curl-free) part. The irrotational flow vector is the gradient of some scalar function (pressure \mathcal{P} in our case), hence we can express the Helmholtz decomposition as

$$\mathbf{u} = \mathbf{u}_{\text{sol}} + \nabla \mathcal{P}/\rho, \quad (3.9)$$

where ρ defines density. Taking the divergence of both sides, we obtain

$$\nabla \cdot \mathbf{u} = \nabla^2 \mathcal{P}/\rho \quad (\text{since } \nabla \cdot \mathbf{u}_{\text{sol}} = 0). \quad (3.10)$$

With the intermediate vector field \mathbf{u} , the scalar function \mathcal{P} can be computed by solving the above Poisson equation, and then the solenoidal flow vector field can be simply retrieved as

$$\mathbf{u}_{\text{sol}} = \mathbf{u} - \nabla \mathcal{P}/\rho. \quad (3.11)$$

Equations 3.10 and 3.11 represent a pressure projection $\Pi_{\mathcal{C}_{\text{DIV}}}$ operation that projects an arbitrary flow field onto the space of divergence-free flows \mathcal{C}_{DIV} , and is widely used in fluid simulation. Mathematically, this step corresponds to an operator splitting method [22].

3.4.2 Temporal Coherence

The incompressible Navier-Stokes equation describes the time evolution of fluid velocity vector fields, given by:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla \mathcal{P}/\rho + (\nabla \cdot \bar{\tau})/\rho + \mathbf{f}, \quad (3.12)$$

where \mathcal{P} is the pressure, $\bar{\tau}$ is deviatoric stress and \mathbf{f} is an external force. For a non-viscous fluid in absence of external force and ignoring the unknown pressure gradient term, Equation 3.12 becomes

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = 0, \quad (3.13)$$

which refers to an approximated evolution of fluid velocity over time. On the basis of this equation, we can advect the fluid velocity at the current time step by itself, and then project it onto a space of divergence-free flows to generate an estimation of the subsequent velocity field, and vice versa. This time evolution equation will be introduced into the optimization problem discussed in the following as a soft constraint.

3.4.3 Optimization Problem

We aim to reconstruct the fluid flow velocity vector fields based on physically constrained optical flow model. The extended optical flow model is formulated as:

$$\begin{aligned} \mathbf{u}_t^* = \operatorname{argmin}_{\mathbf{u}_t} & \int_{\Omega} \mathbf{p}_t \odot (\mathbf{p}_t - \mathbf{p}_{t+1}(\mathbf{u}_t, -\Delta t)) \Big)^{\circ 2} d\Omega + \kappa_3 \|\nabla \mathbf{u}_t\|_2^2 \\ & + \kappa_4 \left(\|\mathbf{M}(\mathbf{u}_t - \Pi_{C_{DIV}}(\mathbf{u}_{t-1}(\mathbf{u}_{t-1}, \Delta t)))\|_2^2 \right. \\ & + \left. \|\mathbf{M}(\mathbf{u}_t - \Pi_{C_{DIV}}(\mathbf{u}_{t+1}(\mathbf{u}_t, -\Delta t)))\|_2^2 \right. \\ & + \left. \Pi_{C_{DIV}}(\mathbf{u}_t), \right) \end{aligned} \quad (3.14)$$

each line of which is explained hereafter:

- the first line describes the conventional Horn-Schunck optical flow model except that the brightness constancy constraint is replaced with the masked particle motion consistency as discussed in Section 3.3.4.
- the second and third lines describe the temporal coherence regularization as explained above: the fluid velocity at the current time step is approximated by either forward warping the flow vector at the previous time step by itself, followed by a projection operation, or by backward warping the flow vector at the next time step by the current flow, followed again by a projection operation. The binary mask \mathbf{M} is employed to ensure confidence-based weighting, giving 0 for the flow vectors near the boundary and 1 for vectors in the central region.
- the fourth line represents an indicator function of the projection method introduced above. Gregson et al. [22] found that the projection operation is equivalent to the proximal operator for the space of divergence-free velocity field. This allows us to integrate the divergence-free constraint into the original optical flow model, which can still be efficiently solved by well-known optimization frameworks.

We formulate this optimization problem in the ADMM framework in Algorithm 3, where the definitions of the functions F_2 and H_2 are given below.

$$\begin{aligned} F_2(\mathbf{u}_t) &= \mathbf{p}_t \odot \left(\mathbf{p}_t - \mathbf{p}_{t+1}(\mathbf{u}_t, -\Delta t) \right)^{\circ 2} + \kappa_3 \|\nabla \mathbf{u}_t\|_2^2 \\ &\quad + \kappa_4 \left(\|\mathbf{M}(\mathbf{u}_t - \Pi_{C_{DIV}}(\mathbf{u}_{t-1}(\mathbf{u}_{t-1}, \Delta t)))\|_2^2 \right. \\ &\quad \left. + \|\mathbf{M}(\mathbf{u}_t - \Pi_{C_{DIV}}(\mathbf{u}_{t+1}(\mathbf{u}_t, -\Delta t)))\|_2^2 \right) \end{aligned}$$

$$H_2(\mathbf{u}_t) = \Pi_{C_{DIV}}(\mathbf{u}_t)$$

The derivation of the proximal operators F_2 is given as follows:

$$\begin{aligned}
\mathbf{u}_t &= \text{prox}_{\sigma_2 F_2}(\mathbf{d}) \Leftrightarrow (\sigma_2 \mathbf{A} + \mathbb{I}) \mathbf{u}_t = \mathbf{d}^j - \sigma_2 \mathbf{b} \\
\mathbf{A} &= \mathbf{p}_t \odot (\nabla \hat{\mathbf{p}}_{t+1}^-)^{\circ 2} + \kappa_3 \nabla^2 + 2\kappa_4 \mathbf{M} \\
\mathbf{b} &= \mathbf{p}_t \odot ((\hat{\mathbf{p}}_{t+1}^- - \mathbf{p}_t) - \nabla \hat{\mathbf{p}}_{t+1}^- \mathbf{u}_t^k) \nabla \hat{\mathbf{p}}_{t+1}^- \\
&\quad - \kappa_4 \mathbf{M} (\Pi_{C_{\text{DIV}}}(\hat{\mathbf{u}}_{t-1}^{k+}) + \Pi_{C_{\text{DIV}}}(\hat{\mathbf{u}}_{t+1}^{k-})),
\end{aligned} \tag{3.15}$$

where

$$\begin{aligned}
\hat{\mathbf{u}}_{t+1}^{k-} &= \mathbf{u}_{t+1}(\mathbf{u}_t^k, -\Delta t), \\
\hat{\mathbf{u}}_{t-1}^{k+} &= \mathbf{u}_{t-1}(\mathbf{u}_{t-1}^k, \Delta t).
\end{aligned}$$

By applying the fixed-point theorem to tackle the nonlinear optimization problem, \mathbf{u}^k in this term refers to the result in the k^{th} iteration. We use Conjugate Gradients to solve this linear system in combination with an incomplete Cholesky factorization.

The derivation of the proximal operators H_2 is given as follows:

$$\mathbf{z} = \text{prox}_{\tau_2 H_2}(\mathbf{h}^j) \Leftrightarrow \mathbf{z} = \Pi_{C_{\text{DIV}}}(\mathbf{h}^j). \tag{3.16}$$

This term is a simple pressure projection step.

Algorithm 3 ADMM Framework of Computing Fluid Velocity Vector Fields

```

1: procedure COMPUTEVELOCITY( $F_2, H_2$ )
2:   for from 1 to ADMM iterations do
3:     // u-minimization step
4:      $\mathbf{u}_t^{j+1} \leftarrow \text{prox}_{\sigma_2 F_2}(\mathbf{z}^j - \mathbf{q}^j)$ 
5:     // z-minimization step
6:      $\mathbf{z}^{j+1} \leftarrow \text{prox}_{\tau_2 H_2}(\mathbf{u}_t^{j+1} + \mathbf{q}^j)$ 
7:     // scaled dual variables update
8:      $\mathbf{q}^{j+1} \leftarrow \mathbf{q} + \mathbf{u}_t^{j+1} - \mathbf{z}^{j+1}$ 
9:   end for
10: end procedure

```

In addition, a coarse-to-fine strategy is applied to deal with large displacements.

The algorithm begins from the coarsest level, and an initial guess of optical flow at the next finer level is obtained by scaling up the flow computed in the coarser level. It should be noted that in this case, the above optimization problem becomes non-linear in \mathbf{u}_t on account of the warping term $\mathbf{p}_{t+1}(\mathbf{u}_t, -\Delta t)$. To tackle this issue, the non-linear term is linearized by using first order Taylor expansion and \mathbf{u}_t is updated iteratively based on fixed-point theorem. More detailed descriptions about this approach are given in [89].

For a sequence of fluid velocity vector fields, each of them is solved independently in an iteration loop. The update of the flow at one time step will impact the subsequent flows in current iteration, and also the previous flows in the following iterations.

3.5 Experimental Setup

Figure 3.3 represents a picture of the experimental configuration used to evaluate the performance of the RainbowPIV algorithm.

3.5.1 Rainbow Light Generation

The experiments were performed using a high power plasma light source combined with a liquid light guide (HPLS245, Thorlabs) to generate a white light (output spectrum: [390, 730 nm]). A collimator was added to obtain a parallel light beam. It is important to have a parallel light beam, to guarantee that two particles having the same depth will be illuminated by the same colored light.

To split the white light in a rainbow beam, we employed a continuously linearly varying bandpass filter (LF103245, Delta *Optical Thin Film*). Other components (prism, blaze grating) were also considered for their ability to generate a rainbow beam. However after comparison, the linear filter appeared to us as the best solution for its effectiveness and simplicity. The generated beam encompasses a spectral range from 480 nm to 680 nm, and corresponding to a depth range of 18 mm in the z

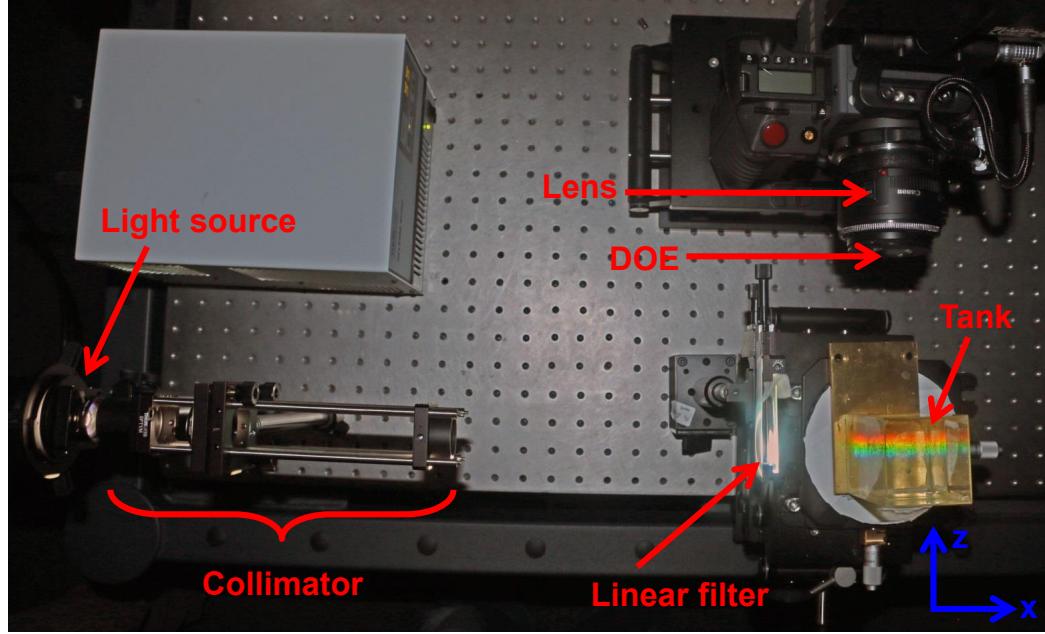


Figure 3.3: Illustration of the experimental setup. A combination of a white light source, a collimator and a linear bandpass filter yields a parallel rainbow beam. After reflection on the particles present in the tank, the light is acquired by a camera. A hybrid refractive-diffractive lens (lens+DOE) is used to ensure that all particles of the measurement volume are focused on the same sensor plane.

direction. Given the height of the beam and the length of the used tank, the two other dimensions of the measurement volume are 50.1 mm along the x axis and 25.6 mm along the y axis.

3.5.2 Acquisition Device

To record the particle images, a digital camera was used (RED SCARLET-X DSMC, sensor: MYSTERIUM-X [$30\text{ mm} \times 15\text{ mm}$], 4096×2160 pixels). A lens with a focal length equal to 50 mm , was mounted on the camera. As can be seen in Figure 3.4 (b, c, d), when a standard refractive lens is used alone, the depth of field is very shallow, and only a small depth range can be in focus. For these three cases the other wavelengths are out of focus, which makes it impossible to exploit these images to retrieve the velocity of particles.

With a standard refractive lens, the thickness of the measurement volume that is

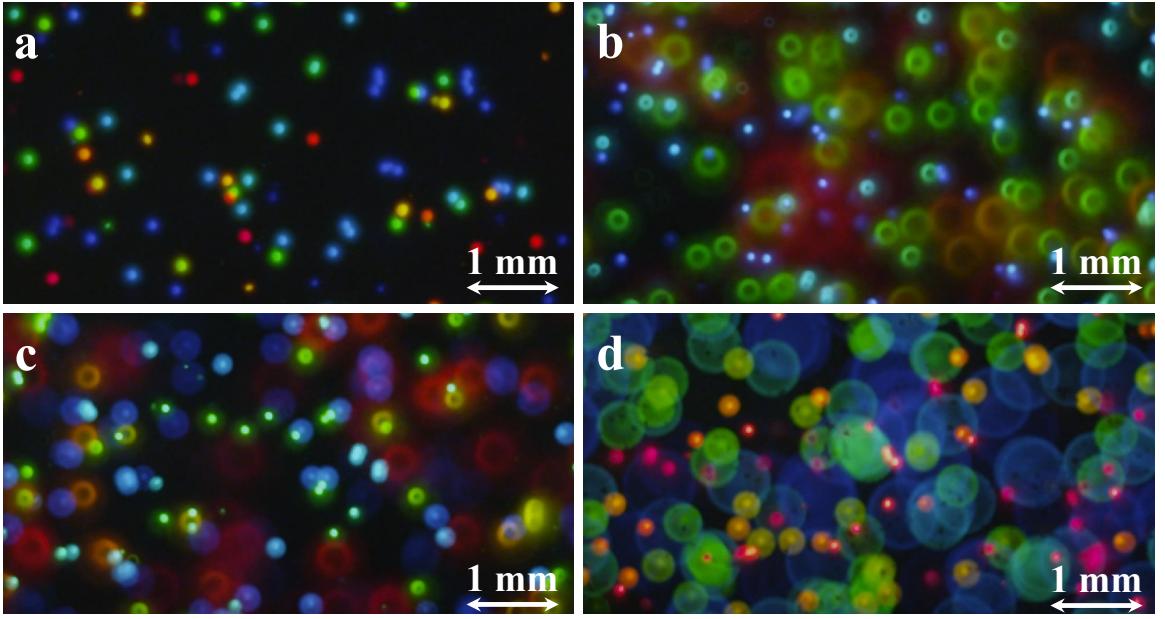


Figure 3.4: Comparison of subsections of the images acquired using a hybrid lens and a refractive lens. (a) Image obtained with the hybrid lens (DOE + lens). (b, c, d) Images obtained when using only the refractive lens. The focus is adjusted respectively for blue (b), green (c) and red (d) particles.

in focus is extremely small for many fluid imaging problems of practical interest. In our case, since the wavelength of the light that illuminates the particles varies linearly in the volume, the in-focus measurement volume can be easily extended by adding a DOE to the camera optics.

When designing the DOE we must ensure that all wavelengths are focused on the same sensor plane. Moreover, the aperture and the magnification of the hybrid lens should allow for an image of good quality.

Figure 3.5 shows a schematic for a hybrid refractive-diffractive lens. The aim is to design a DOE, that allows us to have all particles illuminated by a light, which wavelength is included in $[\lambda_1, \lambda_2]$ focused on the same sensor plane.

The DOE is a Fresnel phase plate, which is implemented as a height field with 16 discrete levels. A DOE is characterized by its phase, which can be expressed as follows:

$$\phi(r) = \frac{2\pi}{\lambda} \cdot \frac{r^2}{f_{\lambda}^{DOE}}, \quad (3.17)$$

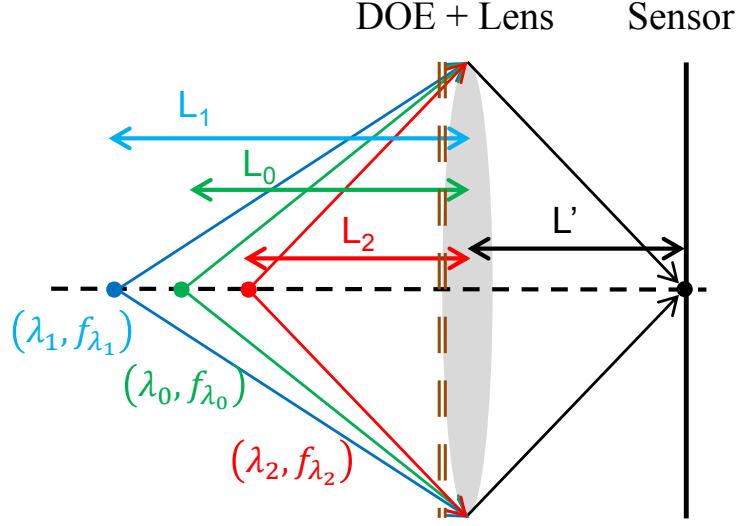


Figure 3.5: Schema for hybrid refractive-diffractive lens. All particles illuminated by a light, which wavelength is included in $[\lambda_1, \lambda_2]$, will be focused on the sensor plane.

where r is the radial distance to the center of the DOE, λ is a given wavelength, and f_λ^{DOE} is the focal length of the DOE associated to the wavelength λ . Indeed, for a DOE the focal length is spectral-dependent, and obeys the following relationship:

$$\lambda \cdot f_\lambda^{DOE} = \text{constant}. \quad (3.18)$$

Thereafter, the wavelength $\lambda_0 = \frac{2\cdot\lambda_1\cdot\lambda_2}{\lambda_1+\lambda_2}$ will be used to design the DOE. Thus, we only need to determine $f_{\lambda_0}^{DOE}$, in order to recover the phase of the DOE.

The image of particles acquired using the hybrid lens is presented in Figure 3.4 (a). One can notice that for this case all particles within the measurement volume are in focus. Their size on the image is almost the same, contrary to the defocused images obtained without using the DOE.

3.5.3 Measured Flows

Two types of experiments were realized using transparent, rectangular tanks made of glass plates placed on a brass metal support:

- Experiments with a ground truth were performed using a high viscosity transparent fluid (PSF-1,000,000 cSt Pure Silicone Fluid). Its viscosity is one million times higher than that of water. White particles (White Polyethylene Microspheres, with a diameter in the range [90, 106 μm]) were introduced to this liquid. This involved heating the liquid while stirring in the particles, followed by vacuum treatment to eliminate bubbles. After cooling the liquid, the particles become frozen in place. Then, experiments were conducted by applying a known movement (translations or rotation) to the tank using micro-meter stages. Therefore, the particle motion is known, since they are immobile with respect to the tank.
- Experiment without “ground truth” were realized using the same particles, after introducing them in a tank containing tap water. A small amount of surfactant (Polysorbate 80 Water) is added in order to reduce the surface tension of water. This is to avoid the agglomeration of particles in the tank. In this case, the particle motion is generated manually through stirring, pouring, and similar excitations.

3.6 Velocity Vector Fields Reconstruction Results

In this section, we first evaluate our proposed approaches based on synthetic examples for ground truth comparisons. Then, we conduct two types of experiments, where the first one is to move particles with known motion vector, verifying the accuracy of our methods on real data, the second one is to work on practical fluids.

3.6.1 Synthetic Simulations

To quantitatively assess our reconstruction method, we tested our algorithm on simulated data. A volume with the size of $100 \times 100 \times 20$ ($X \times Y \times Z$) was simulated and we randomly generated 1000 particles in the volume. The particles were advected over time by ground truth flow vectors that were generated using the method of Stam [53], such that we can obtain time evolved particle distributions from a forward simulation that is completely decoupled from our implementation. Using the image formation model from Equation 3.1, we simulated a time sequence of 5 captured images.

We compare our proposed velocity vector reconstruction algorithm, referred to "S-T div-H&S", with the general multi-scale Horn-Schunck algorithm "H&S" [89] and its extension by introducing divergence free constraint as a proximal operator "div-H&S" [22]. Note that the last two approaches compute the motion between one pair of frames independently, while our approach works on a sequence of frames simultaneously.

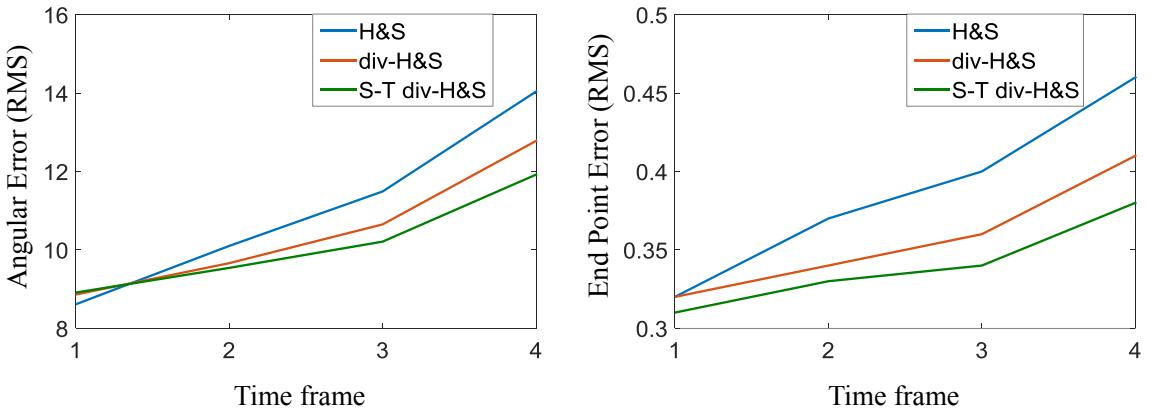


Figure 3.6: Numerical comparisons with ground truth data for different algorithms. Left: Average angular error (in degrees). Right: Average end-point error (in pixels).

For evaluation, we use two metric known from the optical flow literature: the average end-point error, i.e. the average Euclidean distance of the true and estimated particle positions after advection, and the average angular error, i.e. the average discrepancy in flow direction. In Figure 3.6 we show how both types of error accu-

mulate over multiple frames, which is a good indicator for the accuracy of path lines generated through forward integration in the estimated velocity fields. As expected, the reconstruction errors increase over time in all methods. However, by considering temporal coherence, our proposed method exhibits better performance compared to the other two approaches. We point out that a temporal smoothness regularizer may not necessarily result in improved reconstruction results at each particular time step, however, it conveys better estimations in the temporal domain. This is essential for video frames captured in real-world experiments.

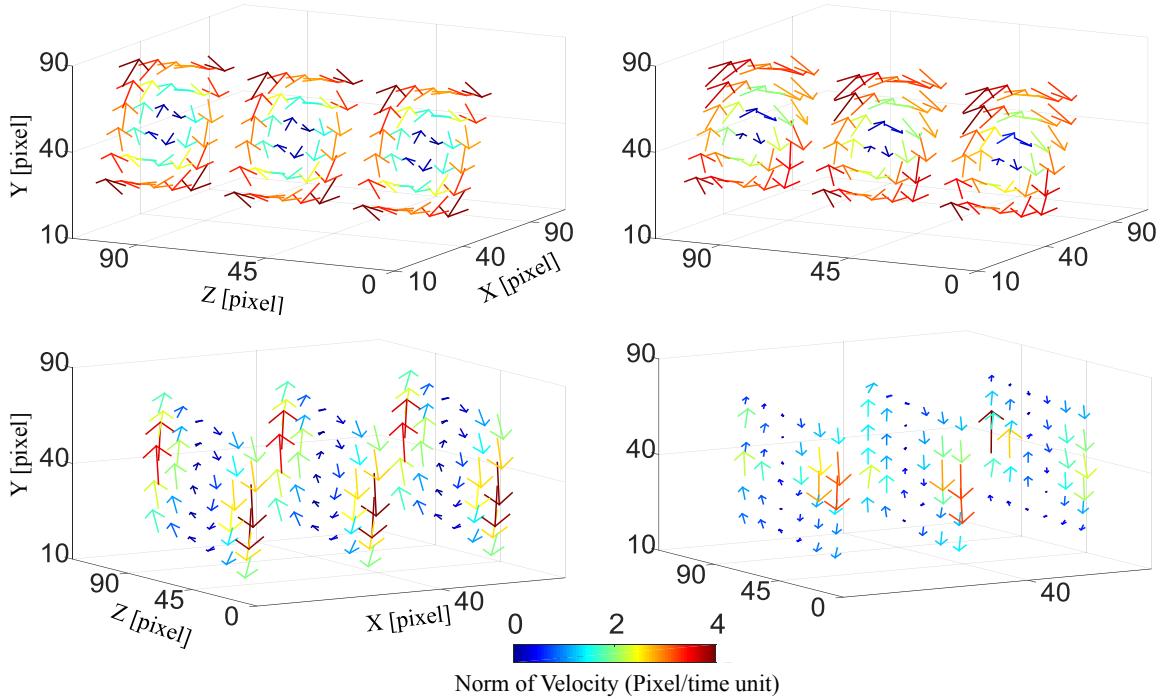


Figure 3.7: Ground truth (left) and reconstructed (right) results for simple analytical flows. Top: Rotation around axis aligned with the optical axis. Bottom: Rotation around axis orthogonal to optical axis.

We also ran experiments on simple analytical flows, specifically one vortex with the rotation axis aligned with the optical axis, and one with the rotation axis orthogonal to the optical axis. The results (shown Figure 3.7) are consistent with the above results on more complex simulated flows. In the first case, the mean of the average end-point error for a sequence of 5 frames are 0.54, 0.52 and 0.49 in pixels, and the

mean of the average angular error are 8.06, 7.77 and 7.08 in degrees respectively for "H&S", "div-H&S" and "S-T div-H&S" approaches. As for the latter one, the mean of the average end-point error are 0.79, 0.77 and 0.73 in pixels, and the mean of the average angular error are 17.15, 16.24 and 13.65 in degrees. These results verify that the temporal smoothness term truly boost the overall reconstruction results for a sequential frame data. Moreover, we could observe a better estimation for flows in the longitudinal plane than those in the transverse plane. This will be further discussed in the following section.

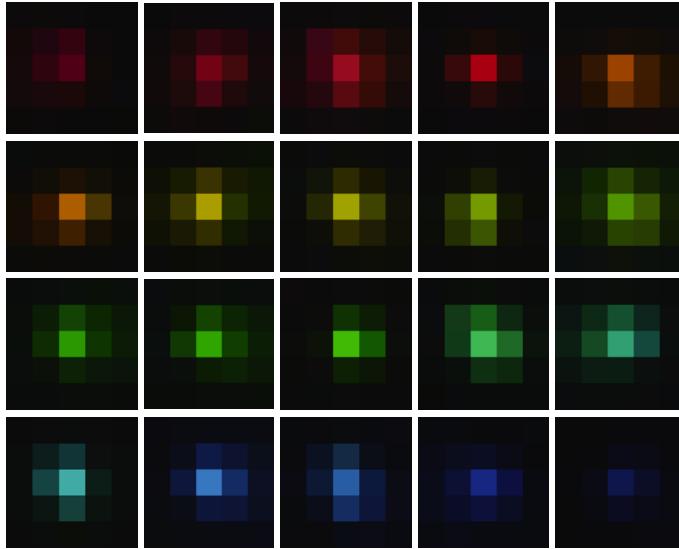


Figure 3.8: Calibrated PSFs for different layers along depth direction. From near camera side to far-end of the camera.

3.6.2 Experiments with a Ground Truth

To evaluate the effectiveness of our proposed methods on real captured data, we firstly conduct the experiments with a tank containing seeded particles in high viscosity liquid. The tank is put on a multi-dimensional translation/rotation stage such that reconstruction results of the algorithm can be compared with ground truth movements. Three independent tests are performed:

1. Translation in the x direction (i.e. perpendicular to camera line of sight): 5

frames were acquired. Between each two successive frames, a translation of 0.2 mm in the x direction was applied.

2. Translation in the z direction (i.e. along the camera line of sight): 5 frames were acquired. Between each two successive frames, a translation of 0.5 mm in the z direction was applied. In this case, the translation is larger, in order to observe easily the color change.
3. An approximation of rotation around the vertical (y) axis in a clockwise direction. With our setup of “frozen” particles in a volume, only an approximation of this rotational motion is possible, since it is not possible to tilt the tank relative to the camera line of sight to avoid distorting the volume by refraction. We therefore approximate rotational flow by rotating the rainbow illumination pattern relative to the tank. In practice, the tank and the camera are mounted together on a rotation table with fixed relative position, and the lighting setup is fixed. The rotations were performed from an angle of -8° to 8° (the reference is defined when the tank is aligned with the (x,z) directions). Between each two successive frames, a rotation of an angle equal to 4° was applied.

Before processing the captured images, we first pass them through a Gaussian filter and then downsample them by a factor of 8, hence the resolution for the downsampled image is about $100\text{ }\mu\text{m}/\text{pixel}$, approximately one particle per image pixel. We discretize the wavelength coordinate into 20 levels, corresponding to $900\text{ }\mu\text{m}/\text{layer}$. The calibrated point spread functions for each levels are shown in Figure 3.8. It should be noted that the resolution along the wavelength coordinate is about 9 times coarser than that in $x - y$ plane.

The reconstructed velocity vector fields are visualized in Figure 3.9. The overall structures of the reconstructed flow in all three cases reveal that a significant part of the real flow structures are reproduced.

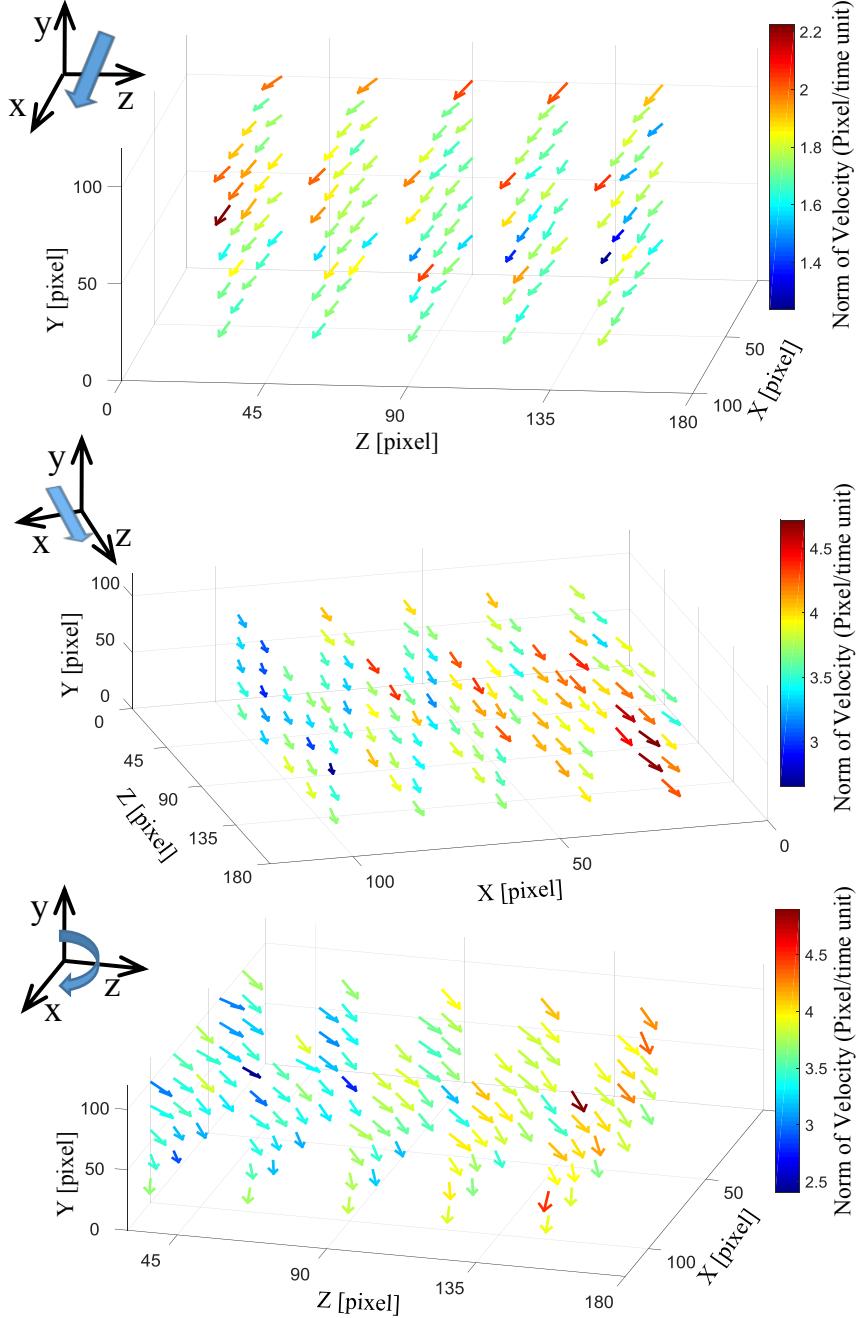


Figure 3.9: The reconstructed velocity vector fields induced by moving the measurement volume with a rotation stage. Top: Translation along x direction. Middle: Translation along z direction (towards the camera). Bottom: Rotate along y axis in clockwise direction. The magnitude of the vectors are coded by color.

Furthermore, we can numerically analyze the reconstructed results with respect to the ground truth movements. In the experiments, the x -axis and the z -axis translations move respectively $200 \mu m$ and $500 \mu m$ in one time step, which corresponds

in the captured images to 2 and 5 pixels. In the rotation test, the total rotation $\frac{\pi}{45} \text{ rad}$, and the 2D plane of the test section has the physical size of $10 \text{ mm} \times 18 \text{ mm}$ ($x \times z$), and distance from the center of the test section to the center of the disk is 10 mm , hence the practical magnitudes of the displacements are about $334 \mu\text{m}$ (3.3 pixel sizes) for the part at the near-end of the disk center and $506 \mu\text{m}$ (5.1 pixel sizes) for the part at the far-end of the disk center. The computed magnitudes of the flow vectors are encoded by color in our represented results.

The mean of the norm of the velocity in the left translational experiment is 1.75 pixel sizes with standard deviation of 0.15, while the mean of that in the experiment of translating towards camera is 3.48 pixel sizes with standard deviation of 0.79. We can see that reconstructed flow vectors reveal higher accuracy for the flow perpendicular to the optical axis with respect to the flow in longitudinal direction. This is reasonable since: (1) depth resolution is highly limited compared to lateral resolution as camera is much more sensitive to the spatial change of objects in 2D plane than the change of wavelength, which results in coarser reconstructed flow vectors along the wavelength coordinate. (2) the error may also come from a bias of reconstructed particle distributions. Determination of the spatial positions of the particles along z axis involves higher uncertainties. Moreover, distortion caused by the refractive effect of the applied high viscosity materials, arises when moving the tank along the z axis. As the thickness of the material between camera and illuminated particles changes, the PSFs are altered simultaneously. Fortunately, this issue does not exist when measuring practical fluid flow, where the particles move, instead of the light beam. Though facing the fact of relatively low reconstruction accuracy for flow in axial direction, not only flow in simple translational structures, but also vortical flows are reasonably reproduced, and the error in wavelength axis is within a certain tolerance, which, in general, is no more than half of length of the discretization intervals.

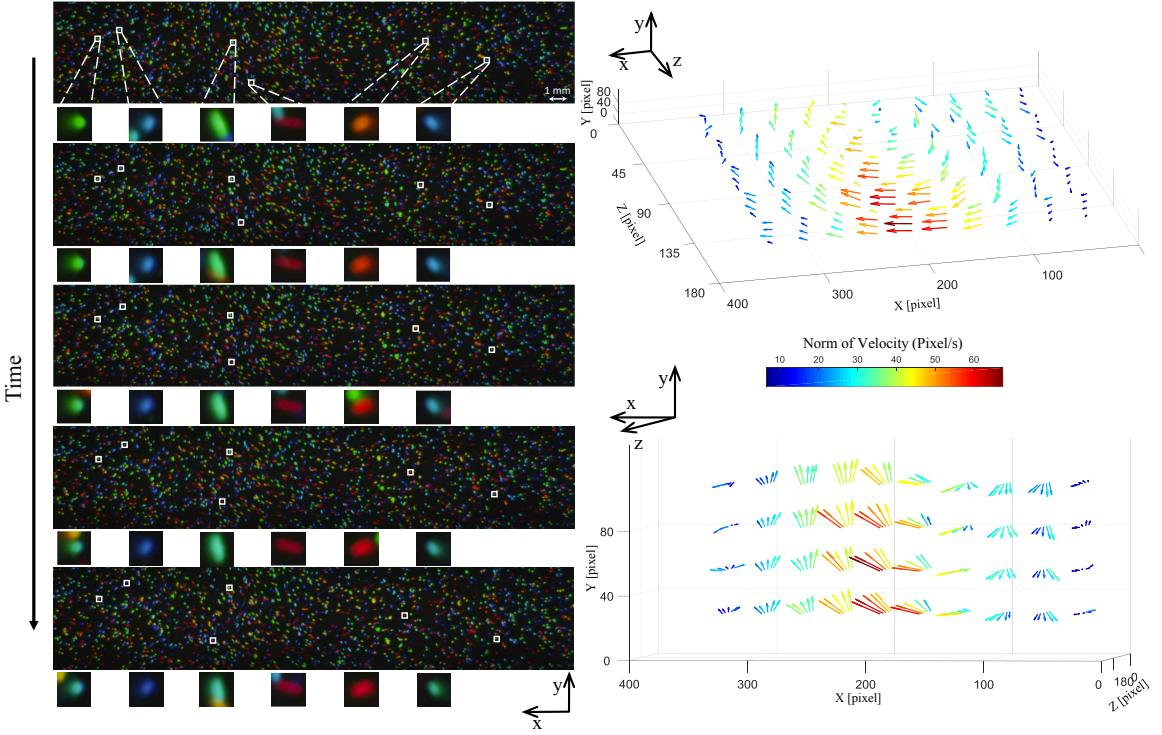


Figure 3.10: Left: 5 successively captured images (without post-processing) in a video frame. Six representative particles are tracked in the time sequential frames to verify the reconstructed flow structure. Right: Computed flow vectors according to the given frame data, viewing from different angles.

3.6.3 Experiments without Ground Truth

Finally, we test our RainbowPIV system on four different real flows of varying complexity (Figures 3.10–3.13). Using the setup described in Section 3.5, we captured image sequences of fluids at a frame rate of $30Hz$, and downsampled the images by a factor of 8 from an original resolution of 4096×2160 to 512×270 . The wavelength coordinate was discretized into 20 levels ($10nm/\text{level}$), hence the maximum grid resolution for any experiment was reach $512 \times 270 \times 20$, although additional cropping was performed on some datasets to only reconstruct regions with interesting flows. The voxel pitch in the (x, y) plane is $100\mu m$, while along the z axis it is $900\mu m$.

The parameters for the optimization method were kept the same for all datasets ($\kappa_1 = \kappa_2 = 0.01$, $\kappa_3 = 10^{-5}$, $\kappa_4 = 10^{-7}$). Only two outer loops were required for all datasets, with $30 - 50$ iterations in the inner loop of the position estimation

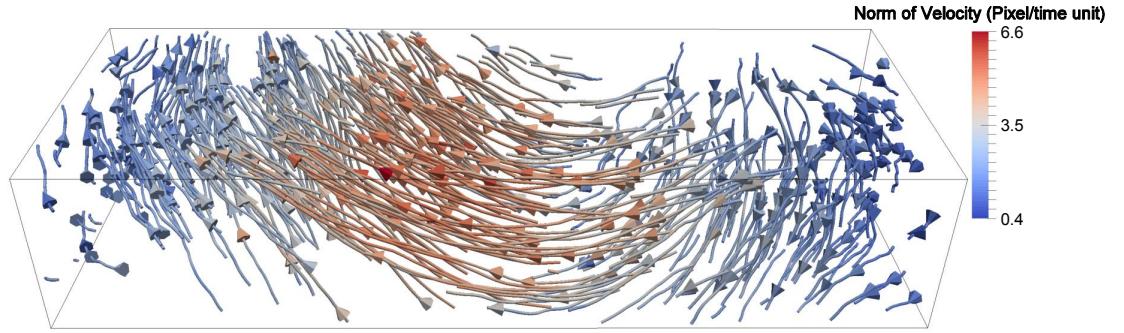


Figure 3.11: Path line visualization of the dataset from Figure 3.10.

subproblem, two inner loops for the velocity estimation problem, and finally five loops for each frame within each velocity estimation step. The reconstruction time for the largest dataset was 125 minutes on a 2.50Ghz Intel Xeon E5-2680 CPU with 128GB RAM. Roughly 1/3 of that time was spent on the position estimation, and the remaining 2/3 on the velocity estimation.

Five successive captured images are shown in the left hand side of Figure 3.10, and the reconstructed velocity vectors are visualized on the right hand side of the same figure. Six representative particles are manually selected to verify the accuracy of the computed flow vectors. The first particle moves upward in the image plane, and the color of it changes from green to cyan, which states that it moves away from the camera. The second particle moves upward and slightly to the right in the image plane and in the depth direction, it moves to the far-end of the camera. The third particle moves to the upper left, color changes from green to cyan. The fourth particle quickly moves to the left hand side with no significant color change. From the fourth particle, we can observe a certain amount of motion blur due to its large velocity. The fifth and sixth particle move downwards in the image plane and towards camera in the wavelength domain, while the orange one moves to the right and blue one moves to the left. Comparing the motion of these chosen particles with the corresponding flow vectors in the reconstructed results, it reveals that overall agreement is achieved. In addition, the actual stirred flow structure is supposed to be a vortex, rotating in

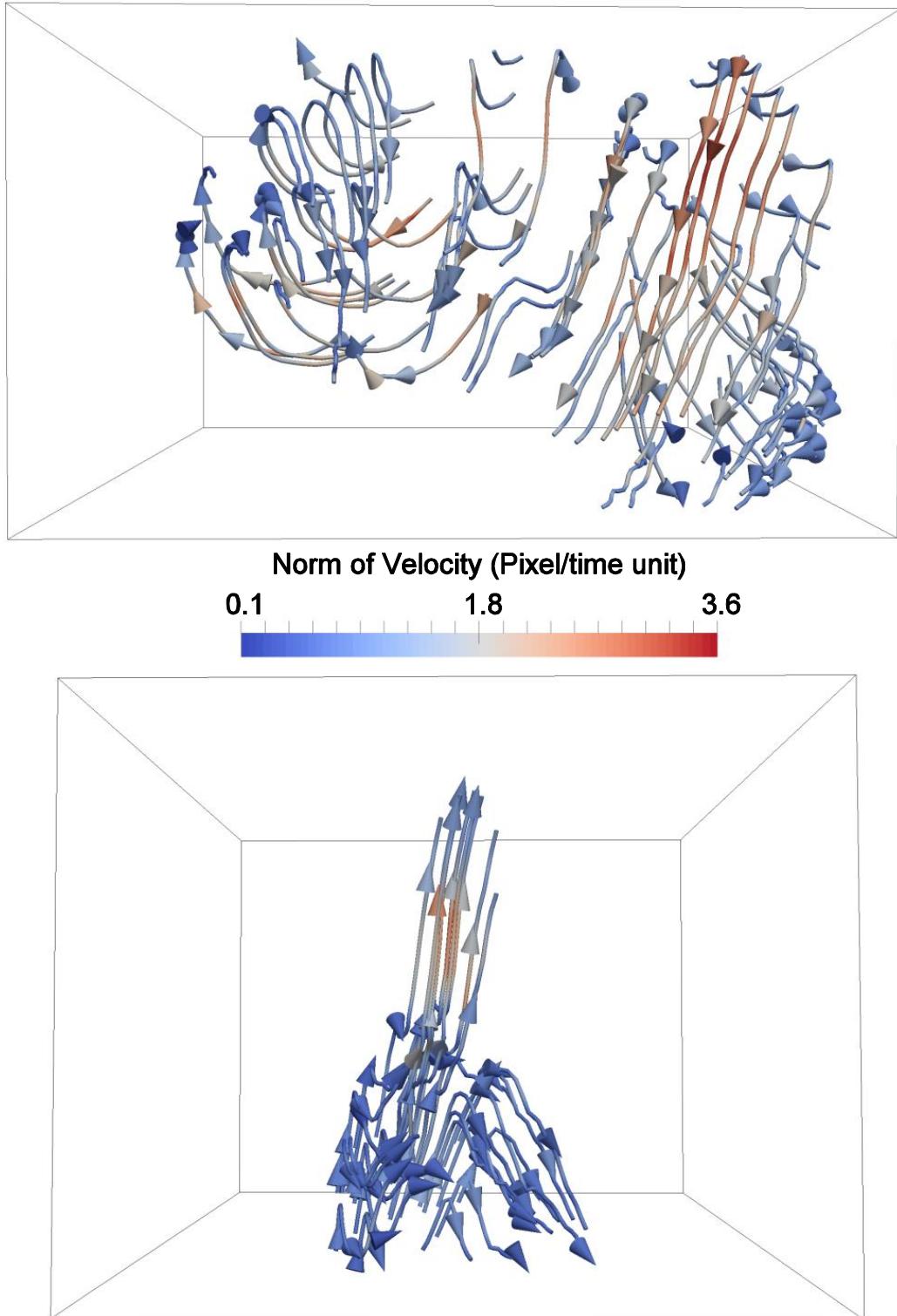


Figure 3.12: Path line visualization for two more datasets, corresponding to a drop of water being dripped into the volume from the top (top image), and a small jet of water being injected from the bottom (bottom image).

a clockwise direction. We observe that the key features of the vortex structure are well reconstructed by our developed methods. A path line visualization of the same velocity data is shown in Figure 3.11. Note that the particles in the visualization are seeded synthetically and do not directly correspond to RainbowPIV particles.

Figure 3.12 shows two more data sets, one with a drop of water being dripped into the volume from the top, and one where a small amount of liquid is injected into the volume form the bottom. The recovered flow field in both cases is consistent with both the expectations and the observed RainbowPIV frames.

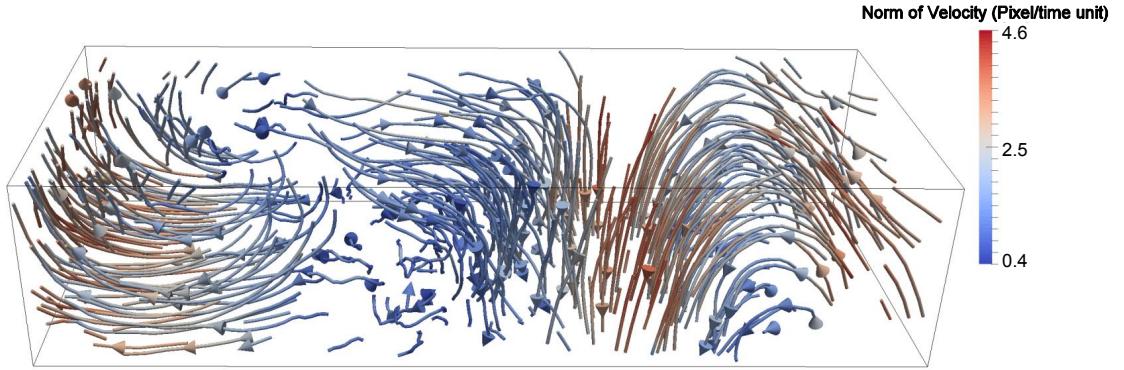


Figure 3.13: Path line visualization of a complex flow created by stirring the fluid. Note the two vortices that interact in a complex fashion. The visualization uses virtual particles that do not correspond directly to real particles imaged with RainbowPIV.

Finally, the most complex example is shown in Figure 3.13. This flow was generated by strongly stirring the fluid, and then letting it set. After a while, the pictured two-vortex structure can be observed. Like many fluid imaging methods, RainbowPIV has problems reconstructing flows with strong motion blur. This limits our ability to reconstruct the early stages of this experiment. To overcome this limitation, high speed cameras could be used in conjunction with stronger light sources.

3.7 Conclusion

In this chapter, we have introduced a novel RainbowPIV system coupled with optimization strategies, which enables us to recover the 3D fluid flow structures using

a single color camera, greatly reducing the hardware setup requirements and easing calibration complexity compared to the other approaches handling 3D-3C measurements. Our approach is implemented by illuminating particles in the volume with "rainbow" light such that the depth information of the particles is color-coded into the captured images, and the 3D trajectory of particles can be tracked by analyzing the 2D spatial motion in the image plane and the color change in the wavelength domain. A specially designed DOE helps to focus all the wavelength planes on the sensor plane simultaneously, to achieve high lateral resolution and relatively large depth of focus at the same time. We then formulate an inverse problem to reconstruct the particle positions in 3D using a sequence of frames to alleviate the ambiguity issues of identifying particle positions from a single frame. With the recovered particle locations at different time steps, a further step is taken to reconstruct the fluid velocity vector fields. An optimization problem integrating the conventional Horn-Schunck algorithm with physical constraints is proposed to compute the flow vectors.

We demonstrate our approach both on synthetic flows induced by moving a frozen particle volume and by using a real stirred flow. Overall, our method can robustly reconstruct a significant part of the flow structures, and also good accuracy.

The primary drawback of our system is the limited spatial resolution along the wavelength (depth) coordinate. Due to the existence of noise and light scattering issues, and relatively low sensitivity of the camera to the wavelength change, at current stage the wavelength coordinate is not allowed to be discretized any further. In the future this situation could be improved by making use of the IR end of the spectrum instead blue light, where camera sensitivity is rather low. Other possible improvements include the use of cameras with additional color primaries, or primaries that are optimized for this task.

Furthermore, our current system can only measure velocities within a flow volume of fixed dimensions, which are determined in the axial direction by the wavelength

spread of the generated rainbow volume and the matching chromatic aberration in the DOE camera optics. In the future we intend to address this issue by designing a dynamically reconfigurable rainbow light engine. In addition to having an adjustable depth range, this Rainbow light engine will use diffractive optics to provide better light efficiency than the currently used linear filter. On the camera side, a viable solution already exists in the form of encoded diffractive optics [91], which allows for diffractive lenses with dynamically changeable focal length. However, integration of the encoded DOE and the new light engine into a new RainbowPIV setup requires still a significant amount of system development.

We addressed part of the named limitations in this dissertation. Specifically, We propose a scalable and reconfigurable RainbowPIV system in Chapter 4, and propose a depth super-resolved RainbowPIV system in Chapter 5. Despite these issues, on account of the simple setup and good accuracy, our system can be easily implemented and applied to investigate new types of fluid flows in the future.

Chapter 4

Reconfigurable Rainbow PIV for 3D Flow Measurement

This chapter closely follows Xiong et al. [66].

In recent years, 3D Particle Imaging Velocimetry (PIV) has become more and more attractive due to its ability to fully characterize various fluid flows. However, 3D fluid capture and velocity field reconstruction remain a challenging problem. A recent rainbow PIV system encodes depth into color and successfully recovers 3D particle trajectories, but it also suffers from a limited and fixed volume size, as well as a relatively low light efficiency. In this chapter, we propose a reconfigurable rainbow PIV system that extends the volume size to a considerable range. We introduce a parallel double-grating system to improve the light efficiency for scalable rainbow generation. A varifocal encoded diffractive lens is designed to accommodate the size of the rainbow illumination, ranging from $15mm$ to $50mm$. We also propose a truncated consensus ADMM algorithm to efficiently reconstruct particle locations. Our algorithm is $5\times$ faster compared to the state-of-the-art. The reconstruction quality is also improved significantly for a series of density levels. Our method is demonstrated by both simulation and experimental results.

4.1 Introduction

A fully characterized fluid flow is essential for studying fluid properties in the field of fluid dynamics, and for flow editing and re-simulations in other applications. However, investigating complex and three-dimensional fluid phenomena in an easy way remains

unsolved.

Recently, single-camera approaches were proposed on a basis of the plenoptic camera (or light field camera) [34, 33], addressing part of the issues arisen from Tomo-PIV. Specifically, a plenoptic camera records the full 4D light fields, which are generated from the scattered light of seeded particles in the flow. Herein, one can digitally reconstruct the particle locations using ray-tracing based algorithms [35, 92, 93, 94]. Overall, by using a light field camera, it allows a dramatically simplified setup overcoming the optical access limitation. Furthermore, it can digitally refocus the images and thus allows a relatively large depth-of-field even for a large aperture lens. While as a compromise, the LF-PIV approach sacrifices spatial resolution for angular information, in order to achieve sufficient axial resolution. Furthermore, storing and processing 4D light field data is computationally expensive and exhibits heavy memory usage. Besides, light field camera commonly has low frame rates. All of the above mentioned drawbacks make time-resolved reconstruction of non-stationary fluid flows an intractable issue.

A further simplified setup using a single off-the-shelf RGB camera has been presented in Chapter 3. This RainbowPIV system simultaneously achieves high lateral resolution and all depths in focus without significant light loss, and also it is able to reconstruct time-resolved flows based on captured video frames. In short, it illuminates the volume using a rainbow light, which is produced by passing a white beam through a linear variable filter, such that the depth information is encoded by color. On the acquisition side, a custom designed diffractive optical element (DOE) manages a wavelength-selective focus so that all light planes are in focus at the same time. Afterwards, a joint optimization solver for particle distribution and velocity vector fields reconstruction is utilized for time-varying 3D-3C fluid velocity measurements. It also demonstrates its applications in different flow scenarios and reveals a compelling and robust reconstruction accuracy.

Nevertheless, the use of the linear variable filter for rainbow generation limits the depth range of the optical setup to a very specific size. Changing the depth range would require replacing the linear filter with a different version and changing the collimating optics, which involves rather large changes to the optical system. Also in this setup, the rainbow illumination is generated through absorption of unwanted wavelengths in the linear filter, which is a very energy inefficient process. These limitations restrict the above mentioned system from being adopted to an observing volume with various depth ranges. A flexible fluid measurement system would be favored for its applicability to flow phenomena occurred in various length scales. Considering this, we seek to use two off-the-shelf blazed gratings, and further design an encoded diffractive optics (see [91]) to construct an easily reconfigurable PIV system. In this chapter, we make the following contributions:

- We propose a single-camera 3D-3C PIV system with a scalable, reconfigurable rainbow illumination.
- We design and fabricate adjustable diffractive optics to focus all light planes for a reconfigurable rainbow volume.
- We propose a computationally efficient and memory friendly solver for high-precision 3D particle reconstruction.
- We demonstrate the scalability of our hardware setup on real fluid scenarios.

4.2 System Overview

we provide an overview of our designed system that meets the rainbow scalability requirement, and improves the particle position reconstruction performances.

To achieve reconfigurability of the rainbow volume, we require changes to both the illumination and the imaging system. In other words, the fluid volume containing

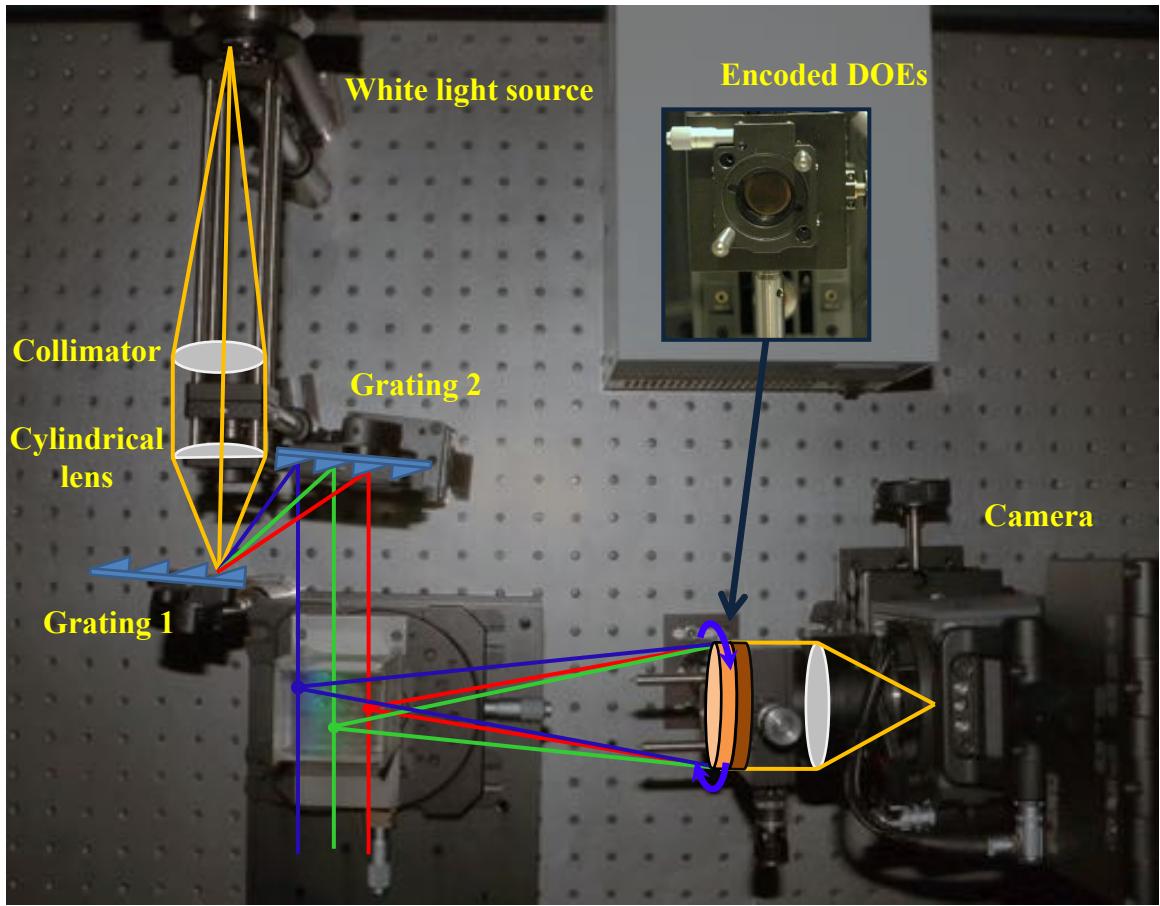


Figure 4.1: Reconfigurable Rainbow PIV setup. The illumination system, based on two parallel blazed gratings, allows the control of the rainbow's width. Meanwhile, a varifocal lens is designed to have a wavelength-selective focus adjustable according to the size of the illuminated volume.

seeded particles should be illuminated by a parallel rainbow beam with readily modifiable size. Furthermore, the imaging system should have an easily adjustable focal length, to ensure that for each rainbow size all particles are in-focus. Figure 4.1 illustrates our optical setup. The solution selected to generate a reconfigurable rainbow is based on the use of two parallel blazed gratings, combined to a white light source, a collimator and a cylindrical lens. Indeed, the size of the rainbow is controlled by the distance between the two gratings as illustrated in Fig. 4.1. On the other hand, an encoded lens composed of two diffractive optical elements and a refractive lens is used to ensure that all colored planes of the rainbow will be in-focus. The equivalent focal length of this encoded lens is governed by the relative angle between the two DOEs.

The captured images are then used to reconstruct 3D particle locations in spatial domain. Specifically, it solves an inverse problem of the image formation model which formulates the imaging process from particle locations to the observed image. After obtaining two consecutive particle distributions, volumetric flow reconstruction is performed, using a modified Horn-Schunck optical flow model. A modified solver (truncated consensus ADMM) for particle distribution retrieval is proposed in the consideration of computational efficiency and the reconstruction quality.

4.3 Particle Position Estimation

Following the image formation model from [65], particles in the volume are illuminated by wavelength-dependent (denoted as λ) light sheets, which are physically dependent on the depth of particle locations, such that the third dimension (z coordinate) of a specific illuminated particle can be determined from its spectral information. The occupancy probability of a specific voxel is described by $P(\xi, \lambda)$, where λ indicates on which light plane the voxel locates and $\xi = (x, y)$ indicates the 2D spatial position in that light plane. Moreover, a RGB camera is employed for capturing the regions of

interest, thus the response of the camera to the illuminated particles can be modeled as a series of point spread functions (PSFs), denoted as $K_c(\lambda)$, which vary from the color channels c ($c \in \{r, g, b\}$) and wavelengths. The convolution operator for the PSFs can be further formulated into matrix $\mathbf{A} = [\mathbf{A}_r; \mathbf{A}_g; \mathbf{A}_b]$. The objective is to reconstruct the particle probability distributions. Specifically, we solve the following minimization problem:

$$(\mathbf{p}^*) = \operatorname{argmin}_{\mathbf{p}} \frac{1}{2} \|\mathbf{Ap} - \mathbf{i}\|_2^2 + \alpha \|\mathbf{p}\|_1, \quad (4.1)$$

where $\mathbf{i} = [\mathbf{i}_r; \mathbf{i}_g; \mathbf{i}_b]$ is the stacked vector for the observed RGB image, \mathbf{p} is the vectorization of 3D particle distributions, and α regulates the sparsity of the reconstructed particles in spatial domain.

The above minimization problem refers to the basic LASSO algorithm, which can be solved by the ADMM framework [84], as in [65]. However this basic solution incurs a significant cost both on terms of computational effort and memory consumption when dealing with a large concatenated matrix \mathbf{A} . Although some matrix-free solvers, for instance conjugate gradient, can be applied to avoid explicitly storing the coefficient matrix, the computational cost is still significant. To ease both memory consumption and computational cost, we introduce a consensus based ADMM solver that allows us to split the problem into smaller chunks based on the spectral response of the camera sensor. These individual parts can be solved almost independently.

Consensus ADMM solver. Consensus ADMM has been recently used to solve distributed optimization problems [95, 84], and large-scale/high-dimensional feature learning based image processing tasks [96]. In particular, it solves an optimization problem involving a composite objective $\sum_j f_j(\mathbf{x}_j)$, where $f_j : \mathbb{R}^n \rightarrow \mathbb{R}$ refers to j^{th} component of the entire objective, and \mathbf{x}_j are the corresponding local variables. The local variables are all constrained by a common global variable \mathbf{y} , written as $\mathbf{x}_j = \mathbf{y}$.

For a simple variation of the objective function with an additional regularization term g , the problem can be expressed as:

$$\begin{aligned} & \text{minimize}_{\mathbf{x}_j} \sum_j f_j(\mathbf{x}_j) + g(\mathbf{y}) \\ & \text{subject to } \mathbf{x}_j = \mathbf{y}, \end{aligned} \tag{4.2}$$

We can fit our particle reconstruction model into the above consensus problem by decomposing it based on color channel, such that each subproblem corresponds to a single color channel. Specifically, for the problem in Eq. (4.1), $f_c(\mathbf{p}_c) = \frac{1}{2}\|\mathbf{A}_c \mathbf{p}_c - \mathbf{i}_c\|_2^2$ and $g(\mathbf{y}) = \alpha\|\mathbf{y}\|_1$. The consensus ADMM solver is expressed in Algorithm 4.

Algorithm 4 Consensus ADMM for solving Eq. 4.2

```

1: for  $k = 1$  to  $N$  do
2:   // p-update step (ridge regression)
3:    $\mathbf{p}_c^{k+1} \leftarrow (\mathbf{A}_c^T \mathbf{A}_c + \rho \mathbb{I})^{-1} (\mathbf{A}_c^T \mathbf{i}_c + \rho(\mathbf{y}^k - \mathbf{q}_c^k))$ 
4:   // y-update step (soft thresholding)
5:    $\mathbf{y}^{k+1} \leftarrow (\bar{\mathbf{p}}^{k+1} + \bar{\mathbf{q}}^k - \frac{\alpha}{3\rho})_+ - (-\bar{\mathbf{p}}^{k+1} - \bar{\mathbf{q}}^k - \frac{\alpha}{3\rho})_+$ 
6:   // scaled dual variables update
7:    $\mathbf{q}_c^{k+1} \leftarrow \mathbf{q}_c^k + \mathbf{p}_c^{k+1} - \mathbf{y}^{k+1}$ 
8: end for

```

Here, $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$ denote the averaged value of \mathbf{p}_c and \mathbf{q}_c over color channels respectively. Each subproblem tackles the reconstruction step in one color channel, and hence it becomes less computationally intensive and more memory friendly. While an issue referred to “over averaging” arises when averaging the local variables in y-update step of Algorithm 4. This issue comes from the fact that the generated rainbow covers a broadband visible spectrum, ranging from red to blue light. Herein, one color channel only covers part of the observing volume in axial direction, for instance red covers the front part of the volume, blue covers the rear part and green covers the middle part. Thus each subproblem can only reconstruct particles over the regions which are referred by the associated color channel. Therefore, averaging the solutions to the subproblems leads to degraded results, which causes the “over averaging” issue.

To overcome this issue, we present a truncated consensus ADMM solver, besides, it achieves higher computational efficiency.

Truncated Consensus ADMM solver. The general idea for the proposed truncated consensus solver is illustrated in Figure 4.2. Since one color channel provides partial information of the particle distributions in the volume along z -axis, we can discard those useless regions and retain the regions with the corresponding color information for each channel. Therefore, every modified subproblem only reconstructs partial particle distributions. Recall the notations in Figure 4.2, \mathbf{A}_c^{tr} denotes the truncated matrix of \mathbf{A}_c and \mathbf{p}_c^{tr} denotes the particle distributions referred by the color channel c .

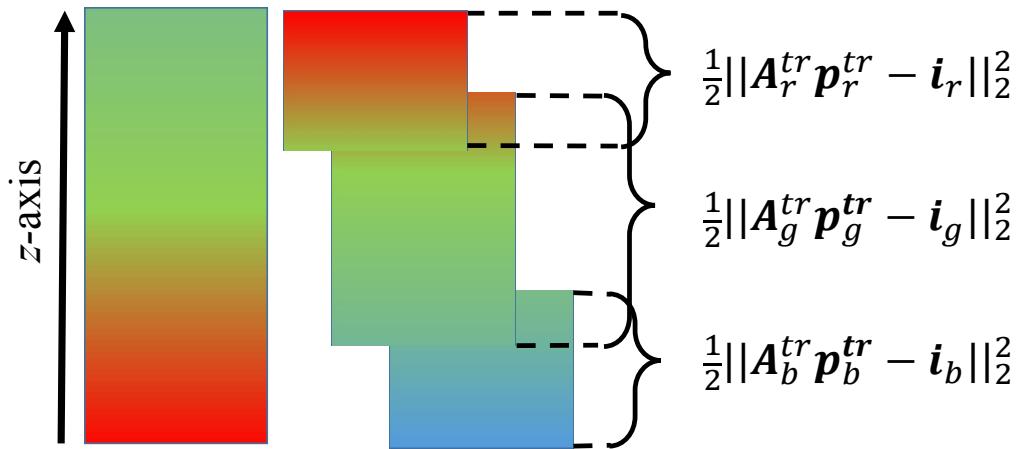


Figure 4.2: Illustration of the truncated consensus solver. Each subproblem only tackles part of the observing volume which is referred by the involved color channel.

Moreover, depth-dependent weights will be applied for the L_1 penalty term in order to compensate for the camera spectral sensitivity. For instance, the spectral response of the green channel of the camera sensor may be effective to wavelengths roughly ranging from $500nm$ to $600nm$, while it has the peak response at $530nm$. This will lead to a non-uniform distributions of reconstructed particles, which are preferred to be placed at wavelength levels exhibiting large response. The compensatory weighting term enables us to ease the biased reconstruction results, which is

expressed as:

$$w_c(\lambda_i) = \frac{\|K_c(\lambda_i)\|_2^2}{\max_i(\|K_c(\lambda_i)\|_2^2)} \alpha, \quad (4.3)$$

where $w_c(\lambda_i)$ refers to the weights of the c color channel at depth level of wavelength λ_i , and M is the number of discretized levels in axial direction. As we can observe, this formula assigns smaller penalties to the wavelength levels with less sensitive spectral response and larger penalties to levels with higher sensitivity.

4.4 Volumetric Flow Reconstruction

The algorithm used for tracking the fluid motion is adopted from Chapter 3, except that we only work on two frames, instead of a video sequence, to validate the scalability of our proposed PIV system. The tracking algorithm is a variant of standard Horn-Schunck model, in which the brightness constancy term is replaced by a particle presence consistency term, and also the divergence-free constraint is taken into account.

Specifically, we solve the following optimization problem

$$\begin{aligned} \mathbf{u}^* = \operatorname{argmin}_{\mathbf{u}} \quad & \int_{\Omega} \frac{\mathbf{p}_1}{2} \odot (\mathbf{p}_1 - \mathbf{p}_2(\mathbf{u}, -\Delta t))^{\circ 2} d\Omega \\ & + \beta \|\nabla \mathbf{u}\|_2^2 + \Pi_{C_{DIV}}(\mathbf{u}), \end{aligned} \quad (4.4)$$

where \odot and $.^{\circ 2}$ are component-wise (Hadamard) operators. The terms in this optimization problem are respectively a Horn-Schunck style “photoconsistency” term on the particle occupancy probabilities, an advection term, and an incompressibility term. We refer to Chapter 3 for detailed derivations of these terms. The method for solving this optimization problem is reproduced in Algorithm 5 for the sake of completeness. For lightening the notation, we define

$$\begin{aligned}\hat{\mathbf{p}}_2^k &= \mathbf{p}_2(\mathbf{u}^k, -\Delta t) \\ \mathbf{H} &= \mathbf{p}_1 \odot (\nabla \hat{\mathbf{p}}_2^k)^{\circ 2} + 2\beta \nabla^2 \\ \mathbf{t} &= \mathbf{p}_1 \odot (\hat{\mathbf{p}}_2^k - \mathbf{p}_1 - \nabla \hat{\mathbf{p}}_2^k \mathbf{u}^k) \nabla \hat{\mathbf{p}}_2^k\end{aligned}$$

Algorithm 5 ADMM for solving Eq. 4.4

```

1: for  $k = 1$  to  $K$  do
2:   // u-minimization step
3:    $\mathbf{u}^{k+1} \leftarrow (\rho \mathbf{H} + \mathbb{I})^{-1}(\mathbf{y}^k - \mathbf{q}^k - \rho \mathbf{t})$ 
4:   // pressure projection step
5:    $\mathbf{y}^{k+1} \leftarrow \Pi_{C_{DIV}}(\mathbf{u}^{k+1} + \mathbf{q}^k)$ 
6:   // scaled dual variables update
7:    $\mathbf{q}^{k+1} \leftarrow \mathbf{q}^k + \mathbf{u}^{k+1} - \mathbf{y}^{k+1}$ 
8: end for
```

4.5 Experimental Setup

In this subsection, we explain our approach to generate a rainbow beam to illuminate the tank. This rainbow volume should consist of a stack of parallel, nearly monochromatic planes, where the density of the planes (and therefore the thickness of the rainbow volume) is easily adjustable in size. Furthermore, the system should provide a good light efficiency. To meet these requirements we propose to use a setup based on two parallel gratings respecting the blaze condition, as shown in Figure 4.3.

A white light beam is generated by a plasma light source combined with a liquid light guide (HPLS245, Thorlabs). The beam becomes parallel after passing through the collimator. Then a cylindrical lens is employed to focus the light into a line that will reach the first blazed grating. After being diffracted by the first grating, the obtained rainbow beam is divergent and spreads out in the perpendicular direction to the rainbow plane. A second blazed grating, identical and parallel to the first one, will diffract the rainbow beam with the same angles as the previous one, yielding a parallel rainbow beam.

The blazed gratings are adopted in our setup because they concentrate the maximum optical power for a given diffraction order (the first order in our case), while the energy of other orders (including the zeroth order) is minimized. The selected blazed gratings have the following characteristics: blaze wavelength 500nm , blaze angle $17^\circ 27'$, 1200 grooves/mm, dimension $50\text{mm} \times 50\text{mm} \times 9.5\text{mm}$ and 60 – 80% grating efficiency at the blaze wavelength. The width of the obtained rainbow beam can be simply controlled by the distance between the two gratings.

Once the particles of interest are illuminated by the reconfigurable rainbow light generated by the double blazed gratings, a varifocal DOE lens is necessary to adapt to the changing volume length. This varifocal DOE lens should be able to continuously adjust to the color depth of field all over along the volume range. Thereby, a single DOE as used in [65] is not sufficient. An encoded diffractive lens as previously reported by Heide et al. [91] is applied to realize the varifocal DOE lens. An encoded diffractive optical element (EnDOE) consists of two DOEs that are optimized to form the phase function of an ideal lens when the two are aligned face-to-face. The focal lengths of the EnDOE are encoded in the relative rotation angles between the two components. Over a specific angle range $0 \leq \theta_{\min} \leq \theta_{\max} \leq \pi$, the focal length can be designed in the range of $[-f_{\max}, -f_{\min}] \cup [f_{\min}, f_{\max}]$.

4.6 Results

4.6.1 Calibration

Since wavelength changes approximately linearly in distance along longitudinal direction, we can uniformly discretize the volume in wavelength domain, and one discrete wavelength level is associated with a depth layer. It is essential to obtain the camera response to each of the wavelength level, and thus one can retrieve the depth information from captured data. In our setup, rainbow with spectrum ranging from 460nm to 660nm was generated. We discretize the spectral range into 20 levels, resulting in

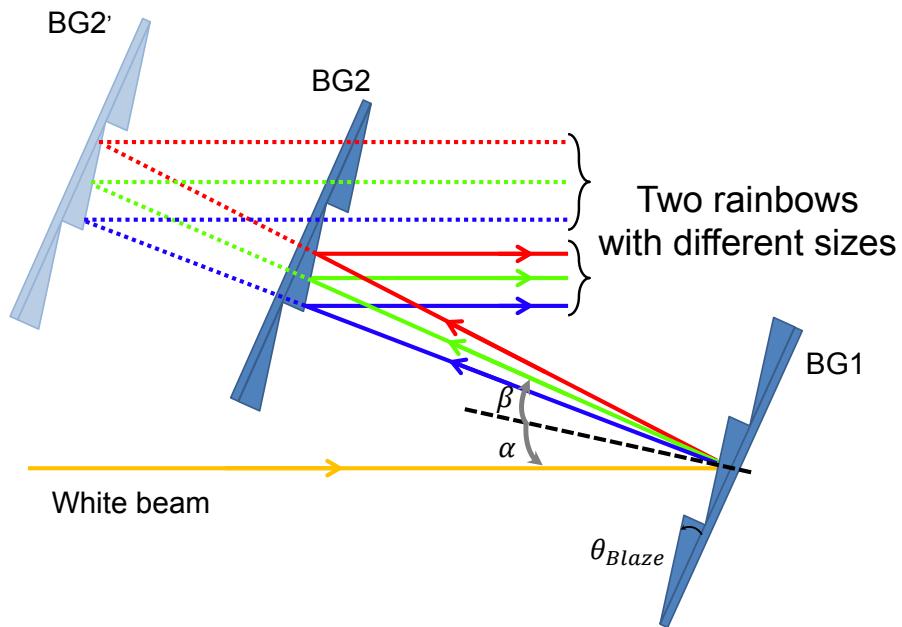


Figure 4.3: Illustration of rainbow generation using two parallel blazed gratings (BG1 and BG2 / BG2'). A parallel white beam reaching BG1 with an incidence angle (α) equal to the blaze angle (θ_{Blaze}), is diffracted at the same angle (β) for the blazed wavelength (green ray). The second diffraction occurs at BG2 / BG2' with the same angles for each wavelength. The obtained rainbow is then parallel, and its width is controlled by the distance between the two gratings.

a spectral resolution of $10\text{nm}/\text{layer}$. When calibrating the camera response at one particular wavelength level, one can either physically block the rest wavelengths or use a wavelength filter, but the former one is preferred in our experiments as it has no extra energy loss. Similar to the work in [65], a RGB camera is applied for calibration and frame data capture, such that each PSF refers to a color image. In our setup, however, the calibration should be re-performed after changing the rainbow size.

4.6.2 Simulation Results

In order to quantitatively evaluate the reconstruction accuracy of the particle retrieval algorithm, we synthetically generate a number of three-dimensional particles, employ the calibrated PSFs and add Gaussian white noise of variance 0.001 to produce a simulated observed image. The reconstruction accuracy is examined by the metric of quality factor (Q), which refers to the normalized correlation coefficient of the real and reconstructed probability fields, given by

$$Q = \frac{\sum \mathbf{p} \cdot \mathbf{p}^*}{\sqrt{\sum (\mathbf{p})^2 \cdot \sum (\mathbf{p}^*)^2}} \quad (4.5)$$

where \mathbf{p} and \mathbf{p}^* denote the reconstructed and ground truth probability fields respectively. The simulation is proceeded on a volume of dimensions $100 \times 100 \times 20$ with different seeding densities. In the truncated approach, color information with $\|K_c(\lambda_i)\|_2^2 \leq 0.25$ is discarded. We compare the proposed truncated ADMM solver (“Trun-ADMM”), with the standard ADMM solver “ADMM” and consensus ADMM solver “CADMM” under the same parameter settings ($\alpha = 0.05$, $\rho = 1$). The results are shown in Figure 4.4(a), where the solid horizontal line ($Q = 0.75$) defines the base line for a sufficiently accurate reconstruction result [1]. It suggests that the proposed method reveals a considerable improvement in reconstruction quality at all densities, while that of consensus solver is degraded significantly due to the “over averaging”

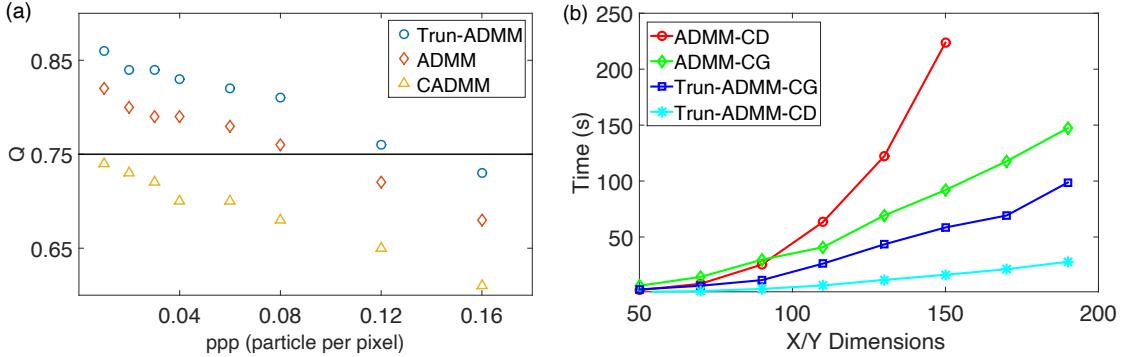


Figure 4.4: (a) Reconstruction quality factor (Q) versus a number of seeding densities (ppp) for different approaches. The solid line means the criterion for a reasonably accurate reconstruction. (b) Execution time for ADMM solver and Trun-ADMM solver using conjugate gradient (CG) and Cholesky decomposition (CD) respectively.

issue.

We then evaluate the execution time of ADMM solver and Trun-ADMM solver for various X/Y dimensions. Conjugate gradient (CG) and Choleskey decomposition (CD) are independently applied for those two solvers to deal with the linear system of Line 3 in Algorithm 4. Note that CG is implemented as a matrix-free solver and CD is preconditioning factorized, which is not counted in the running time. It takes 31 and 36 iterations to reach the same stopping criteria for “ADMM” and “Trun-ADMM”, respectively. The results of the execution time are shown in Figure 4.4(b). The processing time for “ADMM-CD” increases dramatically with the dimensions, and for CG methods it increases moderately, roughly in linear. Even though the proposed truncated consensus solver (“Trun-ADMM-CD”) takes more iterations, it still achieves higher computational efficiency, especially realizing about 5 \times speedup over “ADMM-CG”, which is a standard way for solving this type of problem.

4.6.3 Refocusing

We evaluate the reconfigurability of our system by two experiments with various rainbow sizes, 50mm and 30mm, the results for which are shown in Figure 4.5. We can observe that with the usage of EnDOE, the images are well focused in both cases

even with a large aperture, while the lens-only setup fails to focus on all particles with the same f-number. Although we can reduce the aperture size for a larger depth-of-field, as a tradeoff, the light efficiency is sacrificed and get poorer images. It should be noted that due to the diffraction efficiency, some light energy lost in the camera side, however, we can still obtain sufficiently high quality images comparable to reducing the aperture. We clarify that the size of our proposed system is not limited on the two test examples, but can be adjusted continuously from 15mm to 50mm.

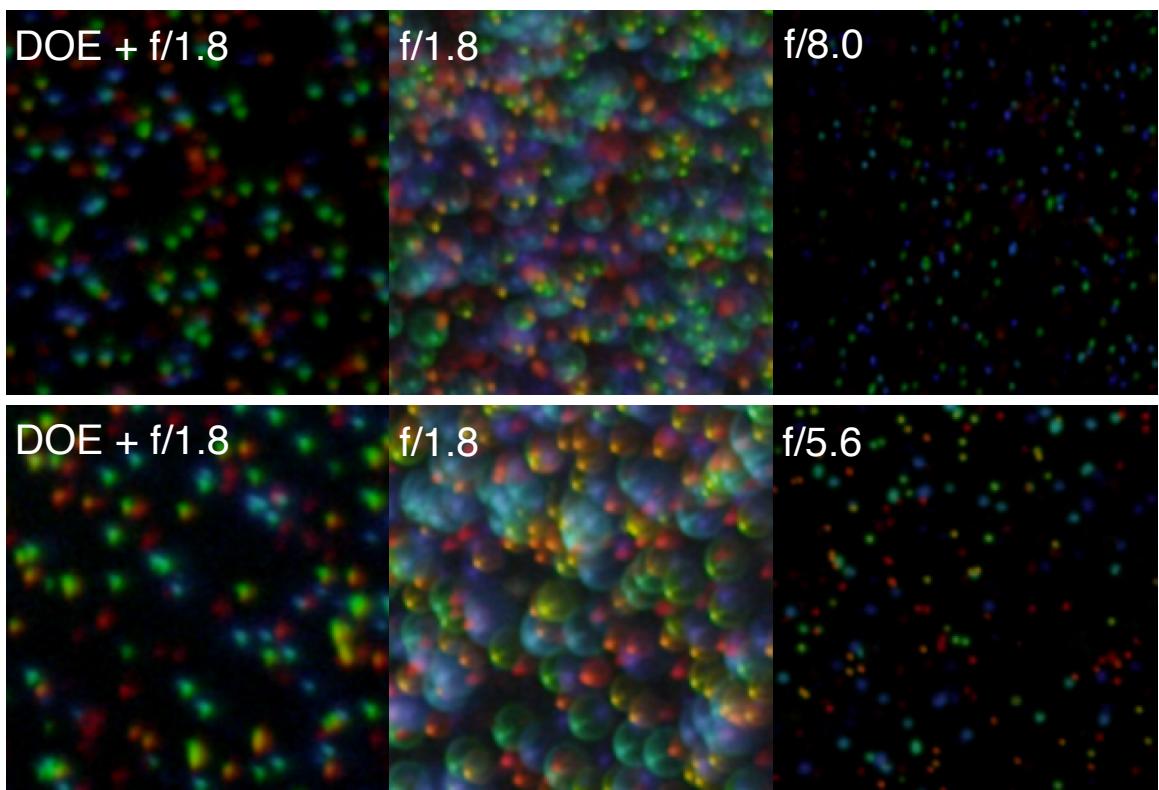


Figure 4.5: Images captured using the encoded DOE with lens (DOE+f/#) and lens-only (f/#) with different aperture settings. The first and second rows show the captured image with rainbow size of 50mm and 30mm respectively. Left: Encoded DOE and lens with f/1.8 since the depth-of-field of EnDOE setup is not affected by the aperture size, the largest available aperture was chosen to maximize the light efficiency. Middle: lens-only with an aperture of f/1.8. While the light efficiency matches that of our setup, depth-of-field blur is significantly worse. Right: lens-only with stopped down apertures of f/8.0 (top) and f/5.6 (bottom). Here the blur is approximately matched to our setup, but the light efficiency is decreased by a factor of 19.7 and 9.6, respectively.

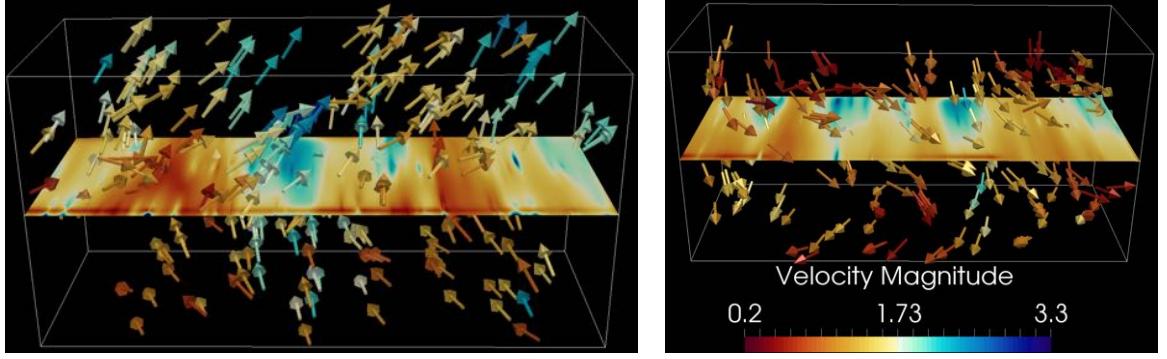


Figure 4.6: Flow vector visualizations for the fluid flow captured in rainbow sizes of 50mm (left) and 30mm (right).

4.6.4 Fluid Flow Measurements

At last we test our system and algorithms on real fluid flows to verify the ability for measuring flows within a volume with alterable depth ranges. The experiments take place on a test section in the size of $40\text{mm} \times 20\text{mm}$ ($x \times y$), and the size of z dimension varies with the generated rainbow. We proceed on the datasets as partially shown in Section 4.6.3, such that the size in axial direction is 30mm or 50mm. The working volume is discretized into grids with resolution of $400 \times 200 \times 20$, therefore, the voxel pitch in $x - y$ plane is $100\mu\text{m}$, and the pitch size in z -axis is 1.5mm or 2.5mm respectively for two generated rainbows. The pictures are captured at a frame rate of 30 fps, and the seeding density is about 0.03 ppp. The parameters for reconstructing the probability fields are the same as listed in Section 4.6.2. As for reconstructing volumetric flow, the parameters are common for all experiments: $\beta = 5e^{-6}$, $\rho = 1$, and the ADMM iteration is 3. The running time for the first step is about 60 seconds for each frame, and for flow reconstruction is roughly 40 minutes on a 2.50Ghz Intel Xeon E5-2680 CPU with 128GB RAM. The reconstructed flow vectors for different flow phenomena in an alterable volume size are visualized in Figure 4.6, and they coincide well with the real fluid flows. These results successfully demonstrate the feasibility of our proposed system in fully characterizing the 3D flow velocities for fluid scenarios in arbitrary volume size within designed depth range.

4.7 Conclusion

In this chapter, we have demonstrated a reconfigurable rainbow PIV system that can efficiently track particle flows in 3D with a considerable range of volume sizes. Compared to existing PIV methods, our system is easy to implement, and is much more flexible and re-configurable. We introduce a high efficiency parallel double-grating system to generate scalable rainbow illumination by simply adjusting the distance between the two. In the camera end, we exploit a varifocal encoded DOE lens to accommodate different sizes of the rainbow illumination, ranging from $15mm$ to $50mm$. Moreover, we propose a truncated consensus ADMM algorithm to reconstruct 3D particle distributions. Our algorithm is $5\times$ faster than the prior arts. The reconstruction quality factor is also improved significantly for a series of density levels.

Similar to [65], the axial resolution is still a limitation for the proposed system. Specifically, two factors determine the axial resolution: the spectral resolution of the illumination, and the spectral resolution of the camera. The illumination spectrum spreads continuously over the volume width, thus the wider the volume is, the finer spectral resolution it reveals. However, representing full spectral information by only three color channels in an RGB camera leads to metamerism, which adds to the ambiguity to resolve spectral features by the camera. The benefit of the increased illuminating spectral resolution could not compensate for the loss of spectral information, hence the axial resolution is decided by the camera spectral resolution. Capturing more color channels can enhance the spectral resolution on the camera side, however, high resolution, multi-spectral video cameras are not readily available for high-res, video-rate capture as required by PIV.

Our system could be further improved in future work. First, although the light efficiency has been significantly improved compared to existing rainbow generation (e.g. linear variable filter), the double-grating system still suffers from light loss in undesired diffraction orders. This could probably be addressed with the use of prisms

instead of blaze gratings, although the prisms would have to be quite large, which adds to the bulk of the system, as well as its cost. Second, instead of using plasma white light source, a supercontinuum white light laser could be employed to improve the SNR, and hence smaller and less reflective particles could be tracked. Because of the reconfigurability of our system, it is fairly straightforward to apply this technique to both large scale and microscopic PIV applications.

Chapter 5

RainbowPIV with Improved Depth Resolution – Design and Comparative Study with TomoPIV

This chapter closely follows Xiong et al. [97].

RainbowPIV is a recent imaging technology proposed for time-resolved 3D-3C fluid velocity measurement using a single RGB camera. It dramatically simplifies the hardware setup and calibration procedures compared to alternative 3D-3C measurement approaches. RainbowPIV combines optical design and tailored reconstruction algorithms, and earlier preliminary studies have demonstrated its ability to extract physically constrained fluid vector fields. This chapter addresses the issue of limited axial resolution, the major drawback of the original RainbowPIV system. We validate the new system with a direct, quantitative comparison to four-camera Tomo-PIV on experimental data. The reconstructed flow vectors of the two approaches exhibit a high degree of consistency, with the RainbowPIV results explicitly guaranteeing physical properties such as divergence free velocity fields for incompressible fluid flows.

5.1 Introduction

In recent years, a great amount of effort has been invested into the development of methods for complete volumetric reconstruction of three-dimensional, three-component (3D-3C) velocity vector fields. Tomographic Particle Imaging Velocimetry (Tomo-PIV) [1, 2] has long been considered the standard technology for 3D measurement due to its ability to handle high particle seeding densities and high spatial resolu-

tion reconstruction, as well as its robustness to many types of flow phenomena. As described in previous chapters, however, Tomo-PIV has the drawbacks of requiring complicated setups and calibration and limited depth-of-field. More importantly, there are many experimental setups where optical access is limited, and thus setting up a multi-camera system becomes impractical. In such situation, a single-camera based 3D-3C technology would be desired.

Two main types of single-camera approaches have been proposed for 3D flow measurement: encoding particle depth into the light path on the camera side, or into color or spectral information on the illumination side. The strategy of encoding particle depth by light path dates back to the work of Willert et al. [32], who place a three-pinhole mask in front of the objective, and recover 3D particle locations based on the position and length of captured image patterns (equilateral triangles) by means of a defocusing technique. This method has been applied to several flow scenarios [98, 99], although it has the disadvantage of low light efficiency (since most light is blocked by the pinhole mask), as well as low particle seeding density because of the difficulty of distinguishing overlapping patterns from nearby particles. More recently, single-camera PIV approaches based on plenoptic (or light-field) cameras have been proposed [92, 94]. These record the full 4D light field of the scattered light of seeding particles and digitally reconstructs particle locations by means of ray-tracing based algorithms. Quantitative comparisons between Tomo-PIV and Plenoptic-PIV have been conducted in recent work [100, 101]. Plenoptic-PIV dramatically simplifies the hardware setup and gets rid of the complicated calibration procedures required in Tomo-PIV. On the other hand, the significant spatial resolution is sacrificed for angular information in this approach. In addition, the available plenoptic cameras suffer from comparatively low frame rates, which in turn limits the usefulness of this approach for time-resolved non-stationary fluid flow measurements.

Another group of single-camera methods is based on the encoding of particle depth

by structured light (monochromatic or polychromatic light). This way, the 3D particle locations can be determined by 2D spatial location and 1D illumination information in the captured single image. A method for exploiting monochromatic illumination has been proposed in [40], which illuminates the volume by a spatially and temporally varying intensity profile, and achieves an axial resolution of 200 discrete depth levels. However, temporal resolution is sacrificed to a certain extend for the sake of high depth resolution, and the method also fails to separate overlapping particles. Most of the related work using polychromatic illumination is summarized in [39]. Early work extracts the particle depth based on a calibrated mapping function of RGB or hue value to depth. This naive method cannot deal with overlapping particles, and thus it is not suitable for dense fluid flow reconstructions. An additional disadvantage is that the method is sensitive to random noise, optical aberration and unpredictable secondary light scattering from the particles.

The recently proposed RainbowPIV [65] (presented in Chapter 3) tackles most of the existing issues by a joint design of the measurement setup (illumination and camera optics), and the reconstruction software. The measurement setup of RainbowPIV consists of an illumination module that generates a rainbow pattern which color-codes the distance of particles from the camera, as well as a diffractive optical element in front of the camera lens, which provides all-in-focus imaging of the color-coded particles. The reconstruction software is comprised of an integrated optimization framework to jointly reconstruct both of the particle distributions and velocity vector fields.

Many single-camera methods suffer from a limited depth-to-width ratio, i.e. they can image only shallow volumes. Indeed, the initial RainbowPIV system [65] had a limited depth-to-width ratio, which was fixed at 0.36. However, in principle the volume depth in RainbowPIV can be tuned by a) changing the thickness of the rainbow, and b) adjusting the design of the all-focus camera optics. This was demonstrated in

Chapter 4, which proposed a reconfigurable RainbowPIV system with an adjustable rainbow generation engine and a varifocal optical design, extending the depth range to (15-50 mm) while the lateral resolution is unaffected. This corresponds to a depth-to-width ratio of 0.3 - 1. Unfortunately, extending the depth range in this fashion reduces the depth resolution, as the same number of distinguishable depth layers get spread out over a larger range.

Therefore, despite advantages of Rainbow PIV in terms of simplicity of setup, it still suffers from a limited axial resolution compared to multi-camera methods like Tomo-PIV. In this chapter, we address this issue and propose an extension to the precedent RainbowPIV, by implementing a depth super-resolved RainbowPIV system. Furthermore, we carry out a direct comparison with the well established four-camera Tomo-PIV system to verify its applicability and reconstruction accuracy for flow measurement. Moreover, we demonstrate that RainbowPIV, unlike Tomo-PIV, reconstructs physically plausible flows e.g. divergence-free flow fields for incompressible fluid flow.

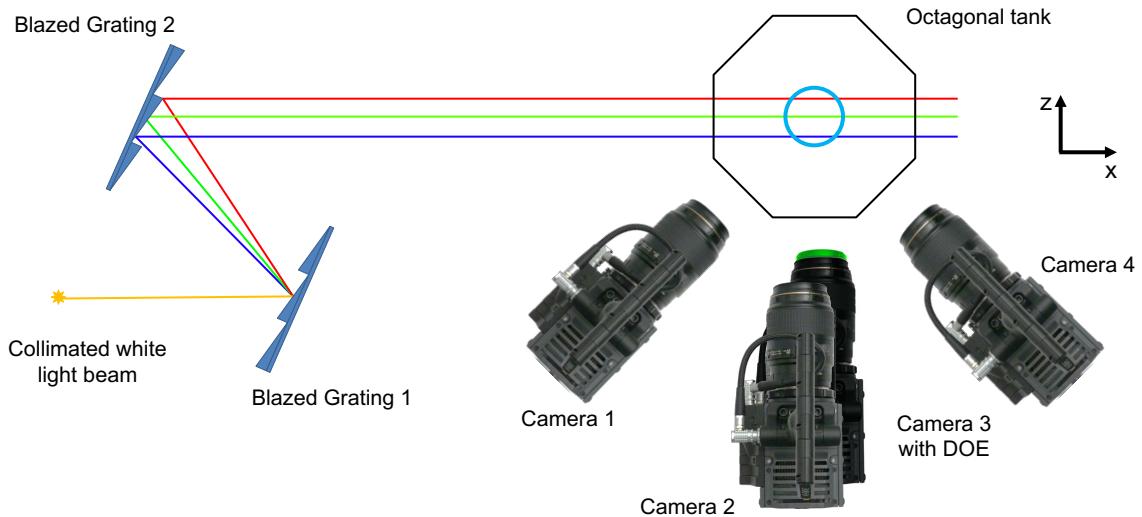


Figure 5.1: Schematic diagram for the experimental setup. Two blazed gratings are utilized for generating size-controlled rainbow. Four cameras are utilized for Tomo-PIV measurement, among which, the third camera with custom designed DOE (diffractive optical element) for all-in-focus imaging is also used for RainbowPIV.

5.2 System Overview

In order to realize a simultaneous measurement with RainbowPIV and Tomo-PIV, the schematic diagram for the designed experimental setup is shown in Fig. 5.1. Four identical color cameras (RED SCARLET-X DSMC with sensor MYSTERIUM-X (30 mm × 15 mm, 4096 × 2160 pixels) are arranged to view an octagonal tank from different perspectives for Tomo-PIV. The elevation angle between camera 2 and 3 is 15°. The internal clocks of the four cameras and triggering signals are produced with the time code generator “Lockit ACL 204”. The central bottom camera (camera 3), equipped with a specially designed diffractive optical element (DOE) to form a hybrid refractive/diffractive imaging system, is also utilized for RainbowPIV. The DOE is a Fresnel phase plate used for all-in-focus imaging in the RainbowPIV setup. Normally, the cameras used in Tomo-PIV setup use a relatively small aperture size in order to obtain a large depth-of-field. The resulting loss in light efficiency poses additional practical challenges, especially for high speed fluid imaging applications. In contrast, the depth-of-field of the utilized hybrid refractive/diffractive lens is not affected by the aperture size, hence RainbowPIV can make use of the largest available aperture for maximum light efficiency, while still maintaining an extended depth-of-field. Specifically, in our setup, the camera with DOE uses an aperture of f/1.8 (the maximum aperture) while the remaining cameras use f/5.6 for a good balance between depth-of-field and light efficiency. The design, fabrication and impact of the DOE are explained in detail in [65].

Another notable part is the illumination system (rainbow generation). The physical size of the measured volume is dependent on the field-of-view of the camera (x-y) and the illumination width (z), corresponding to the rainbow width. In order to generate a collimated rainbow beam with a controllable depth, we utilize two blazed gratings and place them parallel to each other at the blazing angle. A white light beam, produced by a Sumita LS M350 light engine with power consumption of 450W,

entering the setup is separated into a collimated set of monochromatic light sheets forming a rainbow. The thickness of this rainbow can be adjusted by altering the distance between two gratings. In the experiment, we work on a *25 mm* rainbow with wavelength ranging from *480 nm* to *680 nm*. The size of the x-y plane observed by the camera is *50 mm × 35 mm*.

White Polyethylene Microspheres, with a diameter in the range [90, 106 μm] is used in the experiments. We place a vortex ring generator (Fig. 5.2) beneath the tank to produce specific flow scenarios for measurements. Specifically, it is composed of a cylindrical chamber with an elastic latex membrane in the lower third. At the top, the cylinder is covered with a 3D printed cap that has a circular aperture. An air pulse enters the bottom chamber and upwards pressurizes the membrane, driving water out of the top chamber through the aperture, and finally generating the desired vortex rings.

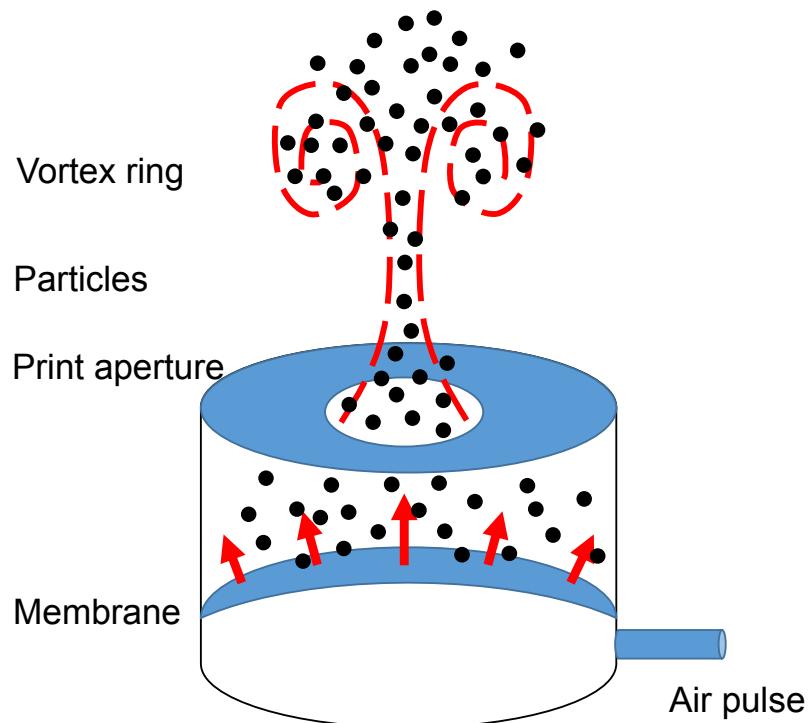


Figure 5.2: Schematic diagram for the vortex ring generator.

For Tomo-PIV processing, by converting the RGB images to 8-bit grayscale im-

ages, a $1120 \times 780 \times 560$ particle intensity field is reconstructed using the MART method, followed by a multi-pass cross-correlation with reduced interrogation window size and 75% overlap. The reconstructed velocity field has a size of $50 \times 35 \times 25$, resulting in a spatial resolution of 1 mm along all 3 axes. All of the utilized algorithms are built in the LaVision Davis software.

5.3 Depth Super-Resolution RainbowPIV

In order to fully reconstruct dense 3D-3C velocity fields, prior work utilizes a pipeline approach that first estimates the particle distribution fields at successive time steps, and then reconstructs the corresponding flow fields using the estimated particle distributions. Separating particle distribution field and velocity field reconstructions neglects temporal coherence as a strong physical cue. Specifically, particles present at one time step should also be present at the next as well as the previous time steps (excluding a small number of particles that enter or leave the observation volume), and their location should be consistent with the estimated 3D-3C flow fields. RainbowPIV for the first time proposed a joint optimization framework for particle distribution fields and fluid velocity vector fields reconstruction on sequentially captured video frames.

The depth resolution of RainbowPIV system is dependent on the number of discretized depth levels, which is basically the number of PSFs. In our implementation, PSFs are expressed by a batch of $3 \times 5 \times 5$ matrices, representing three color channels and a 5×5 spatial window. Owing to relatively low camera sensitivity of the camera sensor to subtle wavelength shifts, only a limited number of PSFs is calibrated in the original RainbowPIV system, resulting in a relatively low depth resolution.

Specifically, the axial reconstruction accuracy relies on two factors: the camera wavelength sensitivity and the quantization error. The sparser the signal is sampled, the larger quantization error it reveals. In this chapter, we show that it is feasible

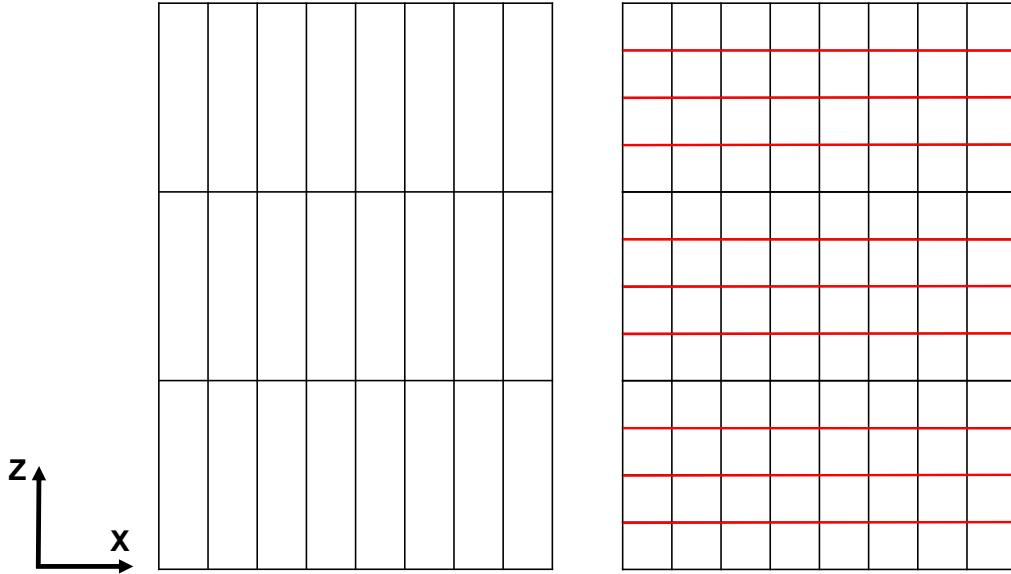


Figure 5.3: Discretization scheme of the precedent RainbowPIV (left) and the proposed strategy (right).

to reduce the quantization error by increasing the number of PSFs using a simple linear interpolation scheme to generate in-between PSFs. As indicated in Fig. 5.3, in the prior implementation, the sampling rate in the axial direction is far lower than that along the lateral directions, giving rise to the relatively large quantization error. With the proposed interpolation scheme, the same sampling rate can be achieved in all directions. While the PSF change with wavelength is not linear over a large spectral range, piecewise linear approximations like ours over small spectral bands are a good model of the true intensity change in each RGB color channel (see Fig. 5.4). With this simple digital adaptation (no hardware adjustment), we are able to generate super-depth particle distribution fields and flow vectors with reduced axial quantization error. A similar interpolation scheme is also applied to reconstruct subpixel flow vectors between two images [102].

We assess the effectiveness of this interpolation scheme on simulated data. We simulate a volume with a fixed lateral dimension size $20\text{ mm} \times 20\text{ mm}$, and various axial dimensions (2 mm, 4 mm, 8 mm, 16 mm), yielding different depth-to-width

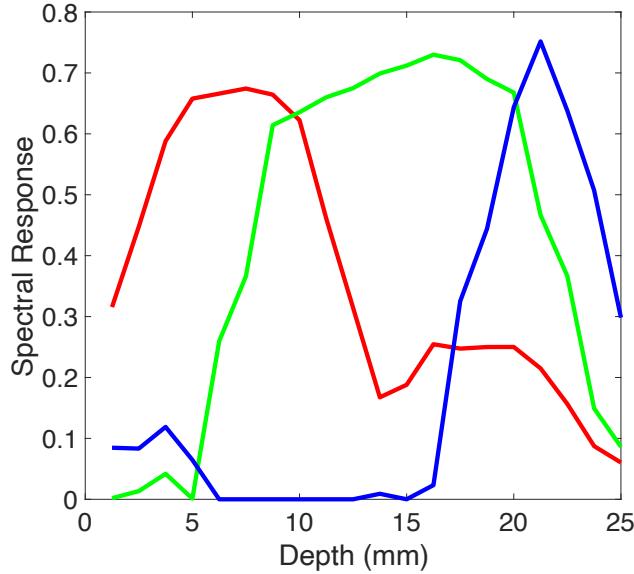


Figure 5.4: Calibrated curves for the normalized camera response to different depth levels (0 - 25mm) for red, green and blue channels. Spectral response of 20 depth levels are obtained from calibration and the curves are generated by linear fitting. These curves match well with camera spectral sensitivity curve with respect to wavelengths. Within a small spectral band, linear curve fitting is a simple, yet effective, approximation model to camera response.

ratio. We use the calibrated PSFs to model camera response to different depth levels. The lateral dimension is discretized by a size of 200×200 , yielding a spacing of 0.1 mm. A random particle distribution of $\text{ppp}=0.05$ is generated and is advected following simulated flow velocities as described in [65]. We compare the reconstruction error of the flow vectors in the axial direction (referring to \mathbf{u}_z) by RainbowPIV and the proposed depth super-resolved RainbowPIV. The results are shown in Fig. 5.5. When the depth-to-width ratio is 0.1, RainbowPIV and the new proposed approach are simply identical as they have the same axial spacing. When the depth-to-width ratio increases, RainbowPIV samples the volume sparser along the axial direction, and yields increased axial reconstruction error (almost linear to the depth-to-width ratio). Using the interpolated scheme, the axial reconstruction accuracy is significantly improved on this simulated results; Error is reduced by roughly a half when the depth-to-width ratio reaches 0.8. The accuracy of the reconstructed flow vectors

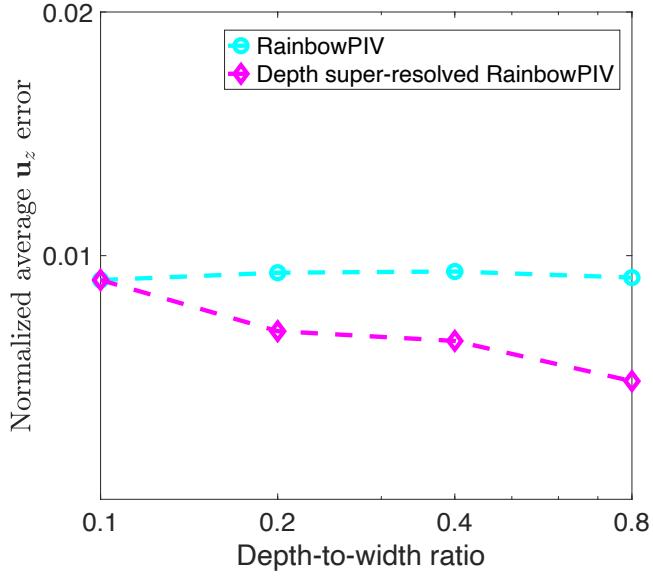


Figure 5.5: Averaged errors for reconstructed flow velocities in the axial direction with respect to various depth-to-width ratio. The error is normalized by dividing the volume depth.

in this fashion is further experimentally validated in the following.

5.4 Experimental Results

5.4.1 Validation Experiments

We validate the proposed strategy on experimental data with the information of ground truth motion as [65] did. We immerse the particles into a tank containing high viscosity liquid to "freeze" them. The tank is then put on a multi-dimensional rotation/translation stage. Therefore, the true movement of particles can be managed by manipulating the stage. We move the tank in the x and z direction respectively to compare the lateral and longitudinal reconstruction accuracy. A translation of 0.2 mm in the x direction and 0.5 mm in the z direction is applied. As a result, the precedent implementation [65] as described in Chapter 3 recovers a motion vector of 0.18 mm/time unit, the mean value of the norm of the reconstructed velocity in the lateral translation, with standard deviation of 0.015 (mm/time unit), and the

proposed method delivers similar reconstruction accuracy, 0.18 mm/time unit with standard deviation of 0.014 (mm/time unit). As for the longitudinal translation, 0.35 mm/time unit with standard deviation of 0.079 (mm/time unit) is achieved by original RainbowPIV, while our super-resolved approach achieves 0.42 mm/time unit with standard deviation of 0.045 (mm/time unit). We therefore observe a significant improvement with respect to axial reconstruction accuracy in this simple scenario, while the lateral reconstruction accuracy is unaffected.

In both implementations, the recovered lateral flows exhibits higher accuracy than the axial flows. As explained in Sec 5.3, the quantization error and camera wavelength sensitivity affect the axial resolution. In the original RainbowPIV, the sampling rate along the axial direction is lower than the lateral sampling rate, resulting in larger quantization error. This super-resolved approach achieves the identical sampling rate in lateral and axial directions, however, the camera is more sensitive to lateral particle motions (change of pixel positions) than axial motions (change of wavelength). This fact brings in higher uncertainties for the estimation of the axial flows.

5.4.2 RainbowPIV and Tomo-PIV

We further validate our method by directly comparing it with four-camera Tomo-PIV system on practical fluid flow phenomena. The simultaneous experimental setup for four-camera Tomo-PIV and single-camera RainbowPIV is shown in Fig. 5.6. The effective resolution of the captured RainbowPIV images is 1600×1120 (pixels). We downsample the images by a factor of 8 in both dimensions (no further image preprocessing required), resulting in 200×140 (pixels) images, and each particle occupies roughly 3×3 pixels. The depth dimension is discretized into 100 levels in order to get regular voxel, where the PSFs of 20 levels (1.25 mm per level) are obtained by calibration, and the intermediate PSFs are digitally generated by linear interpolation based on the calibrated neighbor PSFs. Therefore, the size of the reconstruction

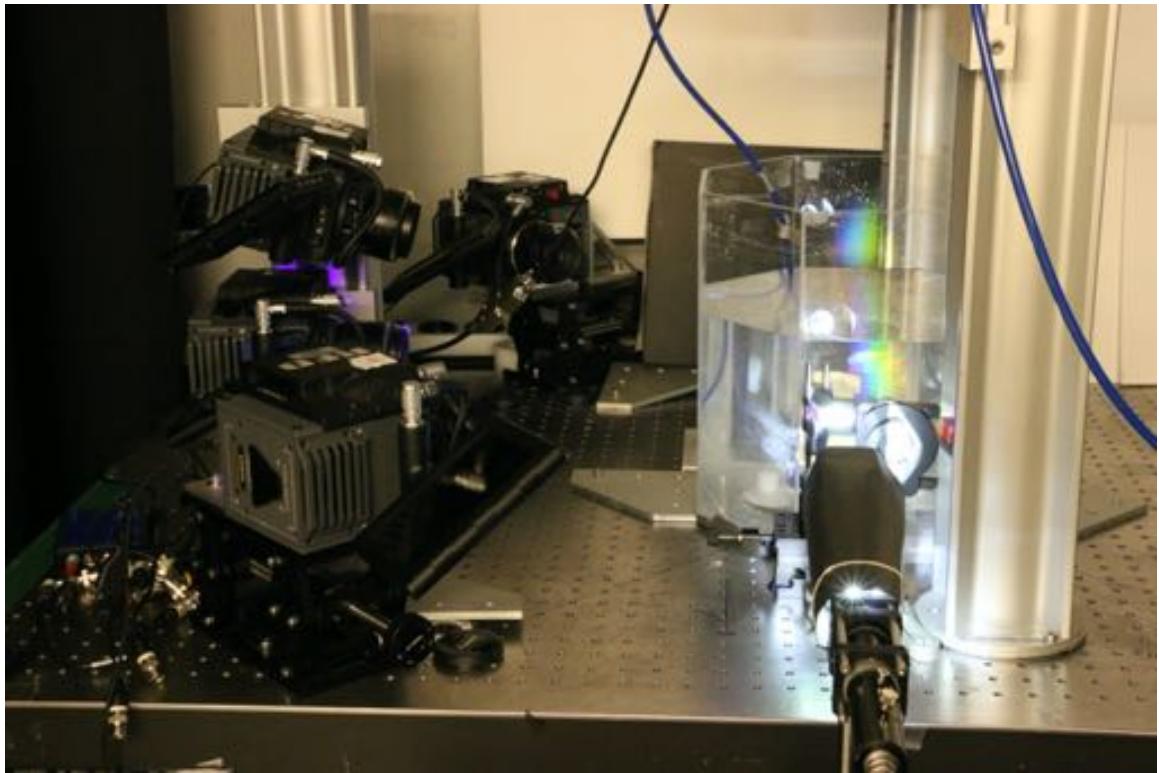


Figure 5.6: Experimental setup for both Tomo-PIV and RainbowPIV.

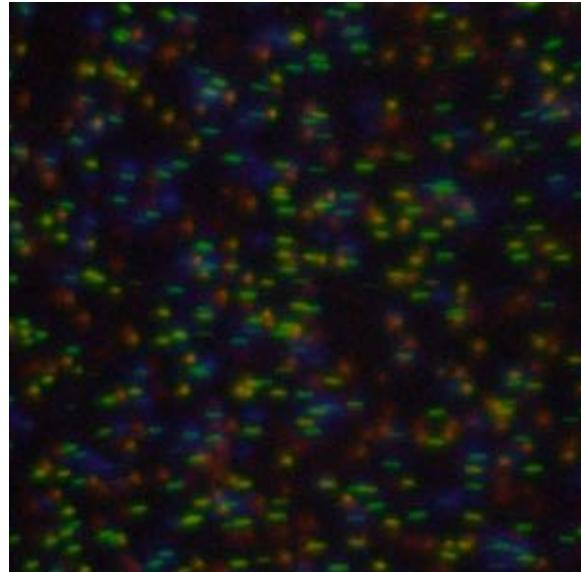


Figure 5.7: A single captured RainbowPIV image with $\text{ppp} = 0.05$. Notice that with the employment of the specifically designed diffractive optical element, different colored particles have similar level of focus with the maximum aperture ($f/1.8$), even though they have different object distance.

grid is $200 \times 140 \times 100$, with a voxel spacing of 0.25 mm along all 3 axes. One vector per voxel will be generated by the proposed algorithm, therefore, the reconstructed velocity field has the same dimension size and spatial resolution as the grids ($200 \times 140 \times 100$). The particle image density is roughly 0.05 ppp (particle per pixel), and one example of the captured RainbowPIV image at this particle density is shown in Fig. 5.7.

A qualitative comparison of the reconstructed velocity fields for the introduced vortex ring at one time step is presented in Fig. 5.8. Two sliced views (one is perpendicular and the other is parallel to the image plane) of the reconstructed flow vectors and the color-coded vorticity magnitude are visualized. The overall flow structure obtained by RainbowPIV agrees well with that computed by Tomo-PIV. Nevertheless, with multiple perspectives, Tomo-PIV has higher depth resolving capability than the proposed single view approach. We can observe that the reconstructed flow vectors from RainbowPIV with a large z component are noisier than flow vectors in the $x - y$ plane. As discussed in Sec. 5.3, on account of relatively lower sensitivity of the camera to the wavelength change, the camera is less sensitive to the particle motions in the axial direction than in-plane motions. Quantitative comparisons between Tomo-PIV and RainbowPIV show an average difference of about 0.05 m/s for flow vector components in the $x - y$ plane, and 0.1 m/s for vector component in the z direction with the maximum flow magnitude at 0.53 m/s . In comparison, the original RainbowPIV implementation has the same in-plane average difference, whereas the out-of plane difference is 0.21 m/s . Although the uncertainty of the axial flow vectors is still larger than the lateral flow vectors, these results affirm the improved axial resolution.

An additional comparison is conducted by visualizing the isosurface of the vorticity magnitude, as is shown in the top of Fig. 5.9. The figures reveal high similarity of the core structures reconstructed by these two measurement technologies. We further verify the mass conservation properties of the reconstructed flow fields which

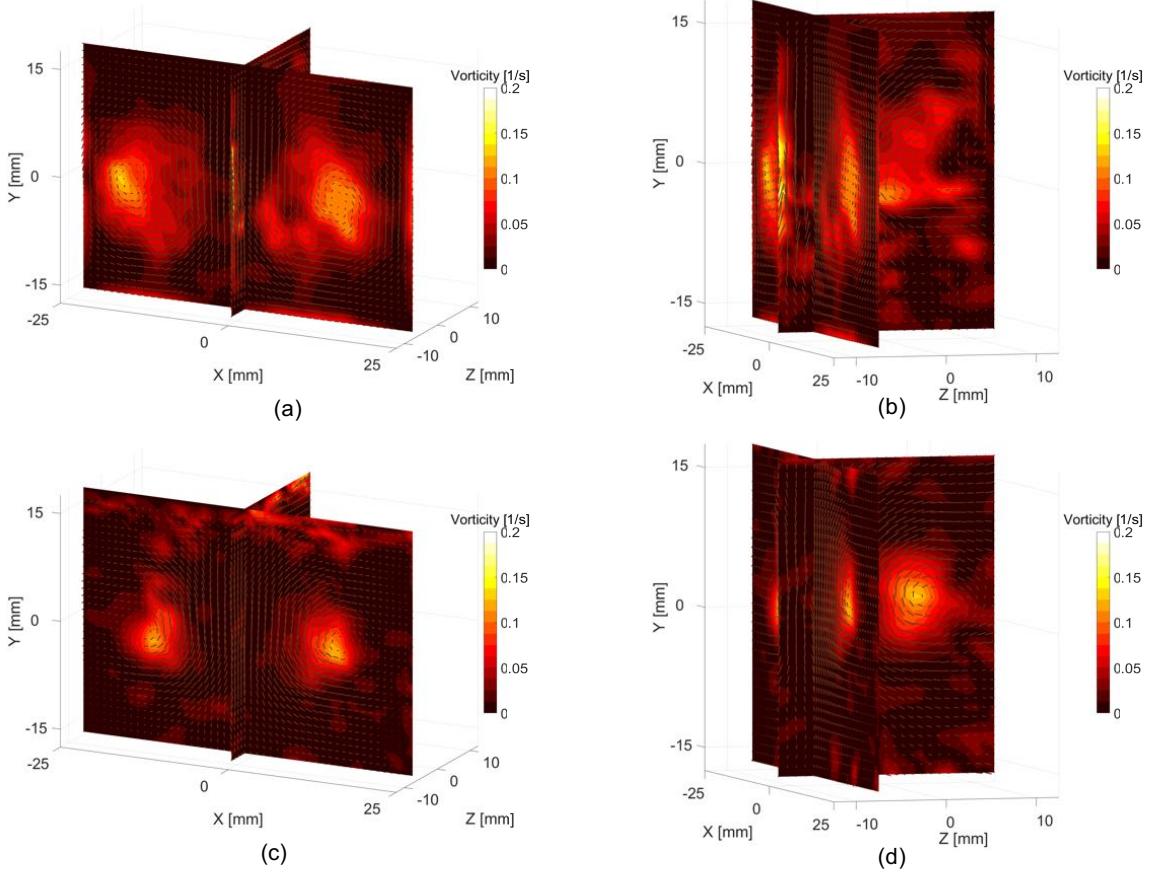


Figure 5.8: (a)-(b): Reconstructed flow vectors and vorticity magnitude by RainbowPIV (the vector fields are shown every other 4 vectors for visualization purpose); (c)-(d): Reconstructed flow vectors and vorticity magnitude by Tomo-PIV. This simultaneous measurement is conducted at $\text{ppp} = 0.05$. The length of the arrow indicates the magnitude of the flow vectors. All the plots are in the same length scale.

should be complied with the physical property. The divergence of the computed flows is shown in the bottom of Fig. 5.9. Zero divergence is expected everywhere for incompressible fluids. This divergence-free property is explicitly enforced by our reconstruction method whereas Tomo-PIV fails to generate flow fields obeying this physical property and also does not readily support to add this constraint during computation. This result demonstrates that, despite the overall high quality and detail of the Tomo-PIV solution, it can in fact *not* be considered as a ground-truth solution.

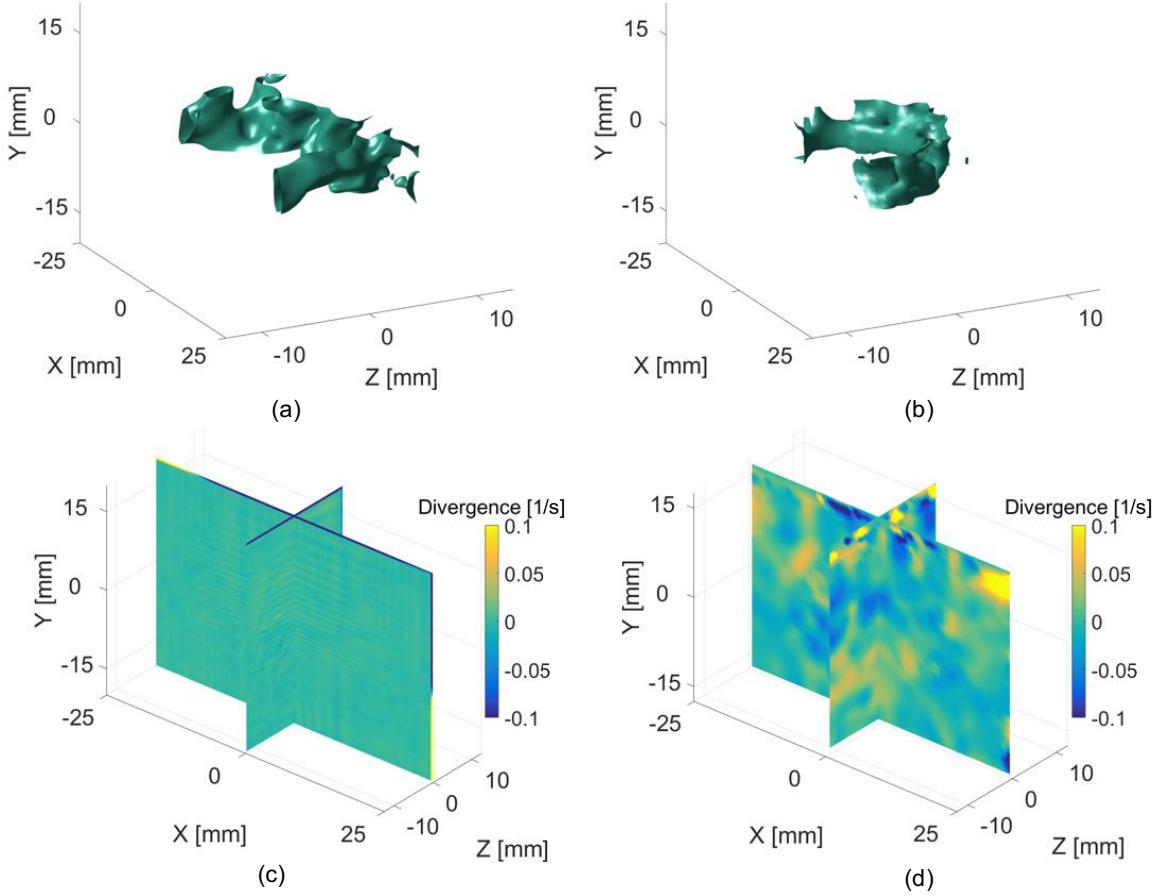


Figure 5.9: Isosurface visualization for the vorticity magnitude computed from RainbowPIV (a) and Tomo-PIV (b) at $\text{ppp}=0.05$; The divergence of the velocity fields ($\nabla \cdot \mathbf{u}$) by RainbowPIV (c) and Tomo-PIV (d).

5.4.3 Low Particle Seeding Density

Next, we evaluate the RainbowPIV system at low particle density situations, which are far below the desired particle density of Tomo-PIV. The correlation-based algorithms applied in Tomo-PIV require sufficiently dense seeding particles to extract accurate flow fields, and usually perform poorly at low particle densities. The presented variant of optical flow method, however, eases the issue by exploiting a *local* constraint (particle occupancy consistency) which ensures that at those regions with present particles, the reconstructed *local* flow vectors match with the practical par-

ticle motion, and two *global* constraints (*global* smoothness constraint and temporal coherence) transmitting the accurate *local* flow vectors to the regions without particle presence. Moreover, the physical properties are always satisfied regardless of the particle density.

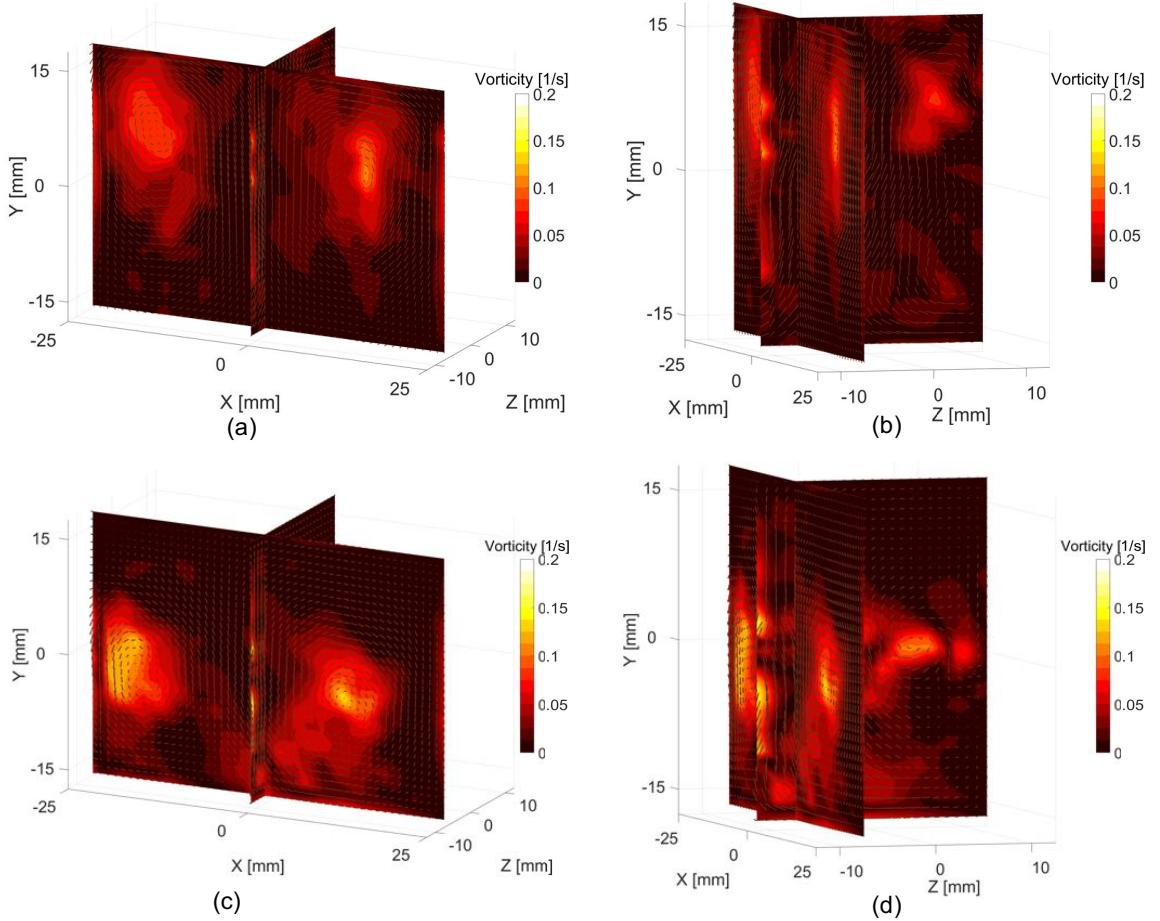


Figure 5.10: Reconstructed flow vectors and vorticity magnitude by RainbowPIV at $ppp=0.015$ (a)-(b) and $ppp=0.005$ (c)-(d).

The tested particle image densities are roughly at 0.005 and 0.015 ppp , the captured images for which are shown in Fig. 5.11. The reconstructed flow vectors and vorticity magnitude are also visualized in Fig. 5.10, which demonstrates that RainbowPIV successfully captures the expected flow structures for vortex rings at various low density levels. Owing to both of the *local* and the *global* constraints, the proposed

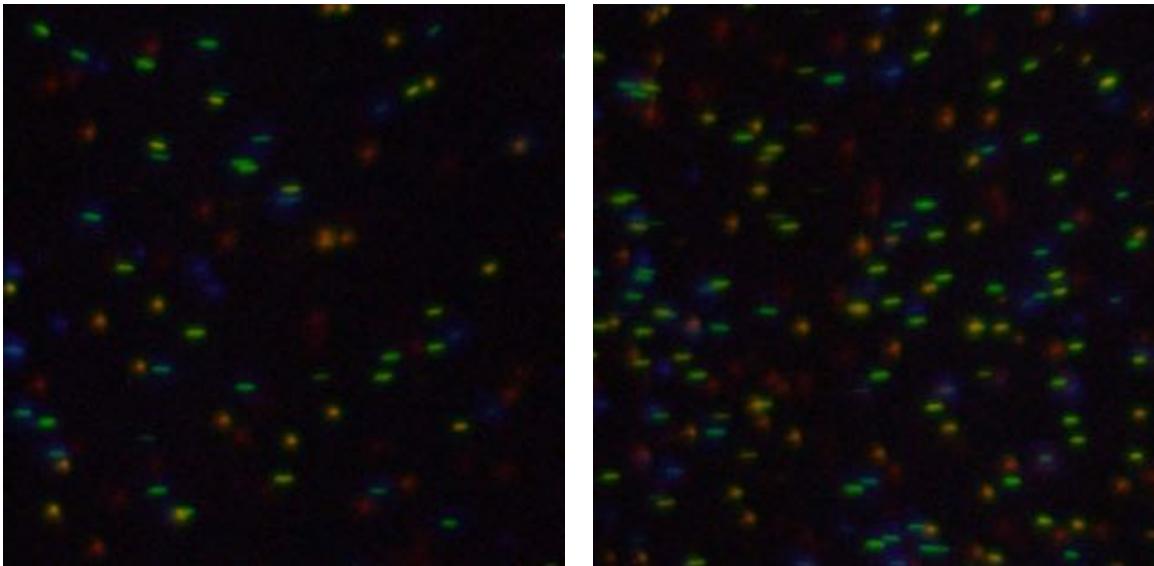


Figure 5.11: The captured RainbowPIV image with $ppp = 0.015$ (left) and $ppp = 0.005$ (right), respectively.

method delivers decent results at rather low particle densities. Under such conditions, particle tracking systems would be preferred over Tomo-PIV. However, this generates *Lagrangian* flow vectors with sparse descriptors, rather than desired *Eulerian* vector fields. As a result, we believe that RainbowPIV can also be very competitive in experiments where uniform particle seeding is difficult or impossible.

5.5 Discussions

Depending on the setup, there are a number of possible alternatives for illumination, including high power white LEDs or even super-continuum lasers. The volume size will determine the total light output required, but also the amount of spatial coherence or beam divergence of the rainbow. This then determines the choice of the light source.

At the moment, the reconstruction algorithm assumes a single exposure per video frame. Pulsed illumination at one pulse per frame could be used to suppress motion blur, without any changes to the software. However, multiple exposures per frame would require changes to the reconstruction; this may be a good avenue for future research.

The maximum flow velocity can be retrieved by our technique is constrained by two factors. From the algorithmic perspective, the maximum flow vectors can be reliably generated are 8 voxels (0.24 m/s) between consecutive time steps (a coarse-to-fine strategy is applied to tackle the issue of large displacements). From the image quality perspective like all other PIV measurement systems, fast-moving particles (along the lateral directions) would cause severe motion blur in the captured images, which downgrades the reconstruction accuracy.

As indicated in Fig. 5.4, the sensitivity of the color-based depth-encoding scheme is non-uniform along the depth axis. However, we did not observe correlations between the flow reconstruction accuracy and depth-encoding sensitivity. The main reason is that our flow estimation framework is composed of both *local* and *global* constraints (elaborated in Sec. 5.4.3). The sensitivity of the depth-encoding scheme only accounts for the *local* constraint, and the reconstructed flows are also managed by the other *global* constraints.

5.6 Conclusion

In this article, we have proposed a depth super-resolved RainbowPIV system which overcomes the limitation of axial resolution existed in precedent RainbowPIV system. A comprehensive study has been conducted for comparing the RainbowPIV with well developed four-camera Tomo-PIV using a simultaneous measurement setup. Both qualitative and quantitative results demonstrates a good agreement achieved by these two systems. Beyond the velocity consistency, due to the physical-constrained velocity estimation model, RainbowPIV delivers divergence-free velocity fields for the measured incompressible fluids, whereas Tomo-PIV fails. Moreover, with the employment of both local and global constraints, RainbowPIV successfully reconstructs velocity fields in rather low particle densities, which is restricted in Tomo-PIV. All the observations confirms the potential usage of RainbowPIV in 3D volumetric ve-

lacity measurements, specially in applications with limited optical access and low or non-uniform particle densities.

Chapter 6

In-the-Wild Single Camera 3D Reconstruction Through Moving Water Surfaces

This chapter closely follows Xiong et al. [103].

We present a method for reconstructing the 3D shape of underwater environments from a single, stationary camera placed above the water. We propose a novel differentiable framework, which, to our knowledge, is the first single-camera solution that is capable of simultaneously retrieving the structure of dynamic water surfaces and static underwater scene geometry in the wild. This framework integrates ray casting of Snell’s law at the refractive interface, multi-view triangulation and specially designed loss functions.

Our method is calibration-free, and thus it is easy to collect data outdoors in uncontrolled environments. Experimental results show that our method is able to realize robust and quality reconstructions on a variety of scenes, both in a laboratory environment and in the wild, and even in a salt water environment. We believe the method is promising for applications in surveying and environmental monitoring.

6.1 Introduction

Shallow waters in rivers, lakes, and oceanfronts are important sites both for their ecosystems, as well as for their economic significance. Environmental monitoring and surveying of these shallow water regions is therefore a task of comparable importance. Unfortunately, detailed 3D scanning of such environments is currently cumbersome,

since it requires placing cameras or 3D scanners under water, which incurs significant equipment costs, and results in slow acquisition time.

A more convenient solution would be to 3D image the environment directly from above water. This is a rather challenging problem, since the fluid, acting as a transmitting medium, is unknown and usually non-stationary. The refraction changes dynamically, and causes a time-varying distortion of the underwater scene. While there has been some work on this problem over the years [76, 78, 77, 79], the state of the art methods require extensive calibration and work primarily in laboratory settings. Tian et al. [76] derives a relation from the variance of distortions in the image plane to the physical scene depths. Alterman et al. [77] proposes a stochastic triangulation approach to recover a probability map of the scenes behind water-air interface. While these statistical approaches neglect the estimation of dynamic fluid structures. Zhang et al. [78] reconstructs fluid surface and immersed scene structures by analyzing the cues of distortion and defocus. Their method requires an undistorted reference image taken under flat water surface, which is inaccessible in the wild. Moreover, they assume the surface normal to be the same for surface areas where the defocus patterns are back-projected to, which does not hold for real fluids. More recently, Qian et al. [68] employs a multi-camera system for water surface and underwater scene reconstruction. Whereas the 3×3 camera array they build is bulky, expensive and complicated to be set up and be calibrated.

In contrast, our method requires no calibration and works “in the wild”. We are able to reconstruct underwater geometry up to a global scale factor, using a single, stationary camera. The distortions from the moving water surface provide a changing parallax for each point on the underwater surface. If this parallax is known, it can be used to triangulate the underwater geometry.

We utilize this observation by jointly estimating both the underwater geometry and the dynamic shape of the water surface (Fig. 6.1). To this end, we propose a

novel differentiable framework governed by ray casting, Snell’s law at the refractive interface, and multi-view triangulation, to tie together all parameters in an integrated image formation model. With our specifically designed loss function, we can progressively and simultaneously optimize the structures of water surfaces and scene geometry to fit the model. Our method is calibration-free and uses only a video sequence as input. Specifically, we make the following contributions:

- We establish a connection between the distorted patterns observed by a single camera and the time-varying fluid structures and the underwater 3D scene geometry.
- We formulate a differentiable framework to reconstruct unknown dynamic water surfaces and scene geometry simultaneously with a specially constructed objective function.
- We demonstrate our method on a variety of synthetic and real scenes. The real scenes are conducted both in the lab and in the wild. We even test the method over seawater.

6.2 System Overview

We propose a differentiable framework to estimate underwater scene geometry along with the time-varying water surface. The inputs to our model are a video sequence captured by a fixed camera. Dense correspondence from each frame to a world reference frame (selected from the input sequences) is pre-computed, ensuring the reconstruction is performed in a unified coordinate system. We feed the flow fields, together with initialized water surfaces and scene geometry (all are initialized as planar surfaces), into the framework, which incorporates ray casting, Snell’s law and multi-view triangulation. The gradients of the specially designed losses with respect to water surfaces and scene geometry are back-propagated, and all parameters are

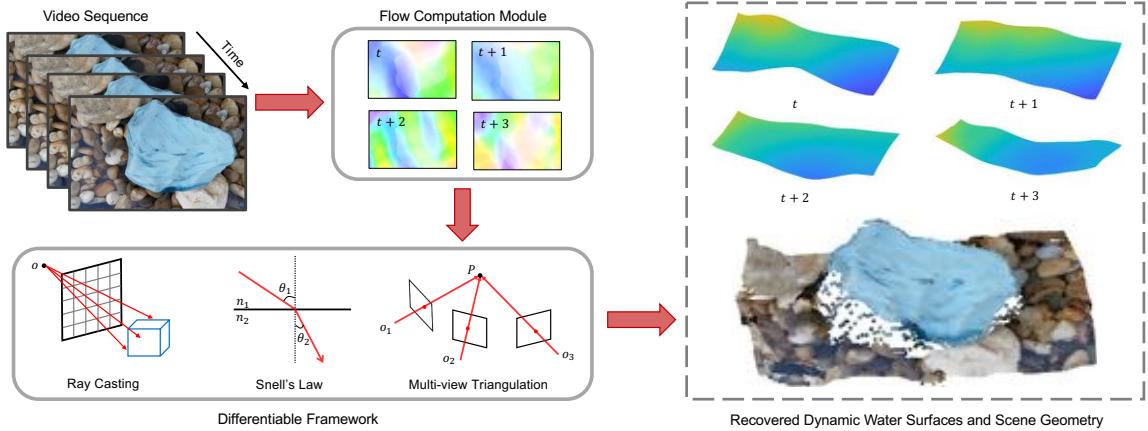


Figure 6.1: Pipeline for underwater scene geometry estimation. The inputs to our model are a sequence of frames captured by a single fixed camera. Dense correspondence matching from each frame to a world reference frame is precomputed, ensuring the reconstruction is performed in an unified coordinate system. We feed the flow fields, together with initialized water surfaces and scene geometry (all are initialized as planar surfaces), into the framework. The gradients of the specially designed losses with respect to water surfaces and scene geometry are back-propagated, and therefore all parameters could be simultaneously optimized. It finally yields quality reconstructions on the 3D structures of time-varying refractive surfaces and underwater geometry.

simultaneously optimized. The final result is a quality reconstruction of the underwater scene, along with an estimate of the time-varying water-air interface. The data shown here was captured in a public fountain environment.

6.3 Differentiable Framework

The reconstruction task is to estimate the underwater scene geometry from a single camera. In the meantime, a dynamic water surface needs to be estimated to establish multi-view triangulation. This task is challenging as any update of one of the two geometries also implies changes to the other. We propose a differentiable framework, integrating both ray casting based on Snell's law, and multi-view triangulation to estimate both geometries from the distortion patterns in the captured video frames. In this framework, the gradients with respect to the parameters of water surfaces and underwater scene geometry are computed through back-propagation from specifically

designed loss functions, and therefore they can be updated simultaneously. In the following, we describe how we parameterize the water surface and the underwater scene geometry, how to construct the framework and tailored loss functions.

Notation. Points and vectors are represented by bold letters, for instance \mathbf{o} denotes the nodal point, \mathbf{n} denotes the surface normal. Objects are represented by italic capital letters, for instance \mathcal{S} denotes the water surface, \mathcal{P} denotes the underwater scene. Scalar values are represented by italic letters, for instance B denotes a B-spline coefficient, κ denotes the weights compensate for different loss functions. (x, y) and t denotes the pixel position in the image plane and t -th frame in the video sequence, and are referenced with superscripts and subscripts, respectively.

6.4 Method

6.4.1 Surface and Scene Representation

In our setup as illustrated in Fig. 6.2 left, the camera is placed at the origin of the coordinate system, and its principle axis is aligned with the z -axis. The water surface \mathcal{S} is parameterized by image plane coordinate (x, y) . Suppose we work with a pinhole camera model, and the focal distance is 1, the emitted ray from image point (x, y) intersects with \mathcal{S} at:

$$\mathbf{s}^{x,y} = D^{x,y}(x, y, 1)^\top, \quad (6.1)$$

where $D^{x,y}$ is the vertical distance from the camera nodal point to its corresponding intersection point $\mathbf{s}^{x,y}$. This parameterization can model the shape of the water surface by finding the function of $D^{x,y}$ and explicitly tracing the rays where they are refracted. Moreover, this representation makes it straightforward to apply both spatial and temporal regularizers to the non-stationary water surfaces, as described in Sec. 6.4.4. $D^{x,y}$ is represented by a set of uniform cubic B-spline patches, making

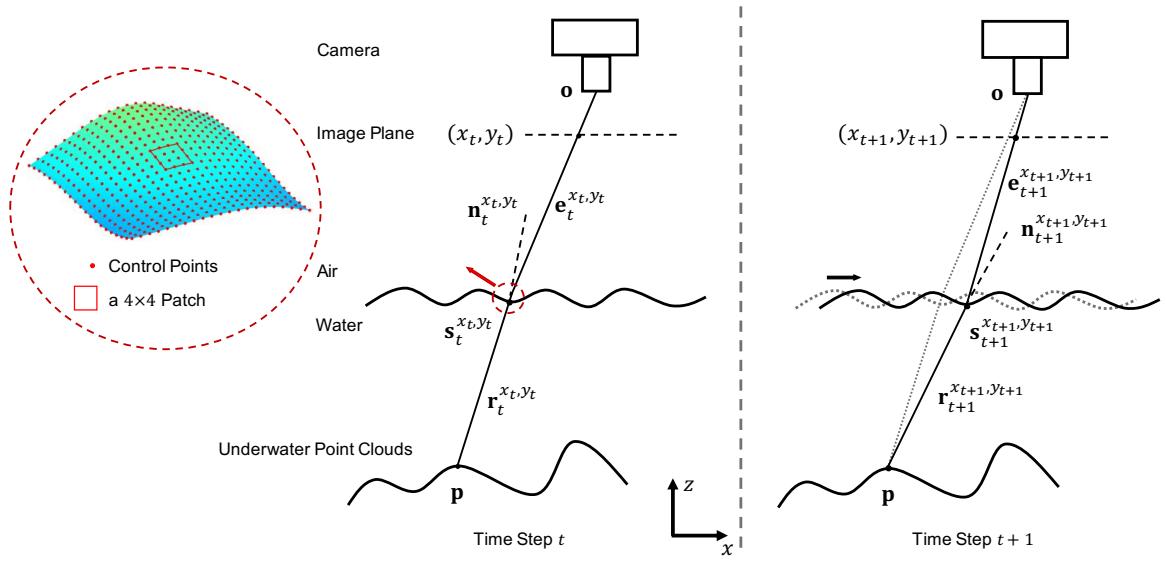


Figure 6.2: Left: An illustration of our setup. A camera is placed above the water. The rays from the camera are traced, which are refracted by the water-air interface following Snell’s law. The interface is represented by a set of control points corresponding to a uniform cubic B-spline surface. We show an example of a 4×4 cubic B-spline patch. Right: A schematic diagram for multi-time triangulation. At consecutive frames, a surface point p is observed at different pixel positions due to the time-varying distortion caused by refraction in the water surface. This effect provides the parallax needed to triangulate the depth of the surface point. The pixel position is referenced with superscripts, and time frame is referenced with subscripts.

the surface C^2 continuity. Specifically, for any point (x, y) within the image plane,

$$D^{x,y} = \sum_{i=0}^{m_x} \sum_{j=0}^{m_y} C_{i,j} B_i(x) B_j(y), \quad (6.2)$$

where $C_{i,j}$ is a control point in a $m_x \times m_y$ patch $\{C_{1,1}, C_{1,2}, \dots, C_{m_x, m_y}\}$. $B_i(x)$ and $B_j(y)$ are the cubic B-spline basis functions that can be derived knowing (x, y) . Fig. 6.2 illustrates how the surface is parameterized and an example of a 4×4 patch. For simplicity of notation, we rewrite Eq. 6.2 in its vector form:

$$D^{x,y} = \mathbf{b}^\top \mathbf{c}, \quad (6.3)$$

where $\mathbf{b} \in \mathbb{R}^{m_x m_y \times 1}$ and $\mathbf{c} \in \mathbb{R}^{m_x m_y \times 1}$ are constructed from vectorized basis functions and control points. The intersection point between the ray from image point (x, y) and water surface is then written as:

$$\mathbf{s}^{x,y} = \mathbf{b}^\top \mathbf{c}(x, y, 1)^\top. \quad (6.4)$$

The surface normal at $\mathbf{s}^{x,y}$ can also be computed in the form of a cross product of $\frac{\partial \mathbf{s}^{x,y}}{\partial x}$ and $\frac{\partial \mathbf{s}^{x,y}}{\partial y}$. Derived from Eq. 6.4, it yields:

$$\mathbf{n}^{x,y} = \left(\frac{\partial \mathbf{b}^\top}{\partial x} \mathbf{c}, \frac{\partial \mathbf{b}^\top}{\partial y} \mathbf{c}, -x \frac{\partial \mathbf{b}^\top}{\partial x} \mathbf{c} - y \frac{\partial \mathbf{b}^\top}{\partial y} \mathbf{c} - \mathbf{b}^\top \mathbf{c} \right)^\top. \quad (6.5)$$

$\frac{\partial \mathbf{b}^\top}{\partial x}$ and $\frac{\partial \mathbf{b}^\top}{\partial y}$ can be explicitly derived from the cubic B-spline basis functions. \mathbf{b} , $\frac{\partial \mathbf{b}^\top}{\partial x}$ and $\frac{\partial \mathbf{b}^\top}{\partial y}$ only need to be computed once, and are reused in the optimization procedure.

Given a camera ray $\mathbf{e}^{x,y}$ intersecting $\mathbf{s}^{x,y}$, where $\mathbf{e}^{x,y} = \mathbf{o} - \mathbf{s}^{x,y}$, and the corresponding surface normal at $\mathbf{n}^{x,y}$, from Snell's law, we can compute the refracted ray

at $\mathbf{s}^{x,y}$ as:

$$\begin{aligned}\mathbf{r}^{x,y} = & \left(\sqrt{1 - \left(\frac{1}{\eta}\right)^2(1 - \mathbf{n}^{x,y} \cdot \mathbf{e}^{x,y})^2} - \frac{1}{\eta} \mathbf{n}^{x,y} \cdot \mathbf{e}^{x,y} \right) \mathbf{n}^{x,y} \\ & + \frac{1}{\eta} \mathbf{e}^{x,y},\end{aligned}\tag{6.6}$$

where \cdot denotes dot product, and η is the refractive index of water (we let η be 1.33 and the refractive index of air be 1). The refracted ray $\mathbf{r}^{x,y}$ intersects with the underwater scene \mathcal{P} at the point defined by:

$$\mathbf{p}^{x,y} = (p_x, p_y, p_z)^\top,\tag{6.7}$$

where p_x, p_y, p_z are the x, y, and z-coordinates. Unlike the water surface, the underwater geometry is represented as a discrete point cloud, since the scene structure may not necessarily be smooth. We assume the underwater scene to be a Lambertian surface, so that brightness constancy holds (all surface points appear the same color from all observation angles).

6.4.2 Multi-Time Triangulation

Knowing only $\mathbf{s}^{x,y}$ and $\mathbf{r}^{x,y}$, we cannot determine the coordinate of $\mathbf{p}^{x,y}$. As in multi-view 3D reconstruction, the 3D position can be determined as the intersection of multiple projection rays. We exploit the property of dynamic water surfaces to establish a multi-time triangulation. The light rays from the underwater scene change direction when passing through the water-air interface, and thus the projection of the scene onto image plane varies over time. The variance of the projected positions relates to the structure of non-stationary water surface. Given two light paths at time step t and $t+1$ as an example as demonstrated in Fig. 6.2 right, the rays from a scene point \mathbf{p} intersect with the water surface at $\mathbf{s}_t^{x_t, y_t}$ and $\mathbf{s}_{t+1}^{x_{t+1}, y_{t+1}}$ at two consecu-

tive time steps, and they are observed by the same camera at image positions (x_t, y_t) and (x_{t+1}, y_{t+1}) , respectively. The image displacement $((x_t, y_t) - (x_{t+1}, y_{t+1}))$ can be obtained from computing the optical flow of those two frames.

Given the surface information, rays from image pixels (x_t, y_t) and (x_{t+1}, y_{t+1}) are traced, and we can obtain $\mathbf{s}_t^{x_t, y_t}$ and $\mathbf{r}_t^{x_t, y_t}$ for time step t , and $\mathbf{s}_{t+1}^{x_{t+1}, y_{t+1}}$ and $\mathbf{r}_{t+1}^{x_{t+1}, y_{t+1}}$ for time step $t + 1$ following Eq. 6.4-Eq. 6.6. Finding the 3D position of intersected underwater points is equivalent to solving a minimization problem for finding the point with the closest distance from both refracted rays. To generalize the model to a video frame with in total T frames, the objective function is formulated as:

$$dis(\mathbf{p}, \mathcal{S}_1, \dots, \mathcal{S}_T) = \sum_{t=1}^T \|\mathbf{p} - \mathbf{s}_t^{x_t, y_t} - ((\mathbf{p} - \mathbf{s}_t^{x_t, y_t})^\top \mathbf{r}_t^{x_t, y_t}) \mathbf{r}_t^{x_t, y_t}\|_2^2, \quad (6.8)$$

where $dis(\mathbf{p}, \mathcal{S}_1, \dots, \mathcal{S}_T)$ defines the summation of the distance of a particular underwater point cloud \mathbf{p} to its associated refractive rays generated from surface structures at various time steps (ranging from 1 to T). This term ties together all frames.

Confidence Mask. The computation of point cloud 3D positions relies on an accurate estimation of the image displacement. It is known that the computation of optical flow between two frames is prone to error in the presence of large motions, extreme distortions, and dramatic illumination changes. All of these issues may occur for captured underwater point clouds. In a global optimization, mis-estimated flows in one area may negatively impact the reconstruction accuracy everywhere. We introduce a confidence mask to suppress unreliable rays when finding the intersection point. The modified Eq. 6.8 is then expressed as:

$$dis(\mathbf{p}, \mathcal{S}_1, \dots, \mathcal{S}_T) = \sum_{t=1}^T M_t \|\mathbf{p} - \mathbf{s}_t^{x_t, y_t} - ((\mathbf{p} - \mathbf{s}_t^{x_t, y_t})^\top \mathbf{r}_t^{x_t, y_t}) \mathbf{r}_t^{x_t, y_t}\|_2^2, \quad (6.9)$$

where M_t is the confidence mask for that scene point at time step t . The mask is determined by backward warping the t -th frame to see whether the image pixels

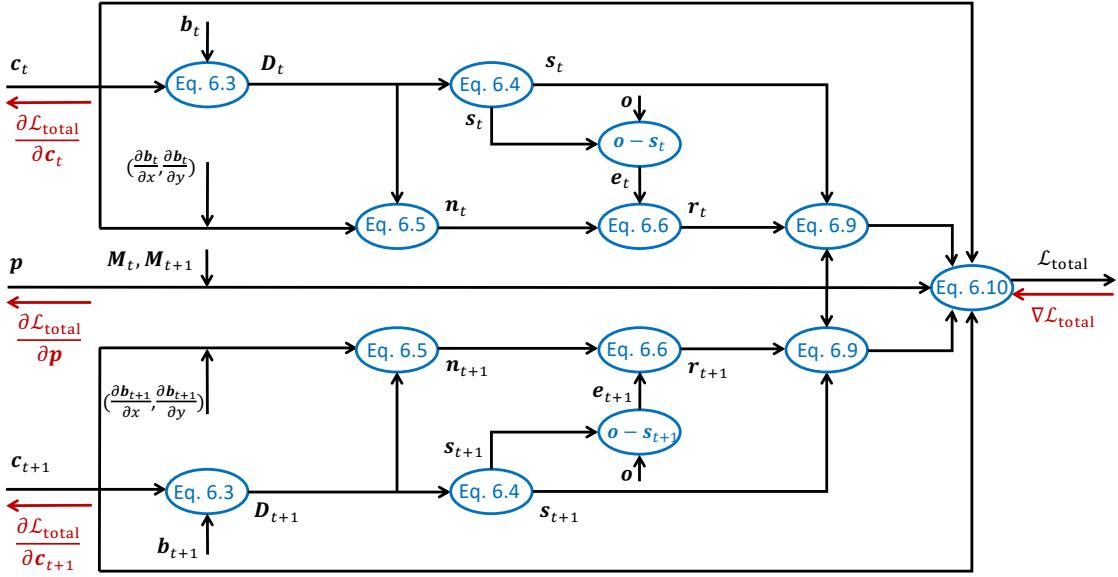


Figure 6.3: The flowchart of our proposed framework. After providing the framework with the structures of water surfaces (shown here are two time steps for illustration) and the underwater point cloud, the model loss can be computed in a fully-differentiable fashion through multi-stage procedures. The gradients can be effectively back-propagated in the framework, so that all parameters can be updated in the same iteration.

match. If the pixels match, let M_t be 1, otherwise, let M_t be 0. With the employment of the confidence mask, a false refractive rays will not be counted when computing the value of $\text{dis}(\mathbf{p}, \mathcal{S}_1, \dots, \mathcal{S}_T)$. This will enhance the robustness of the reconstruction method as demonstrated in Sec. 6.5.

6.4.3 Integrating Ray Casting, Snell's Law and Triangulation

In our setting, the estimation of the surface structure and underwater point clouds are codependent – updating one variable causes changes in another one. Previous work tackles this type of problem in an iterative scheme, alternating on these two subproblems and each of them is solved independently. We propose a novel strategy to integrate both factors into a differentiable framework as illustrated in Fig. 6.3. This framework integrates tracing the camera rays to find intersection points with water surfaces, refracting the rays passing through water-air interface following Snell's

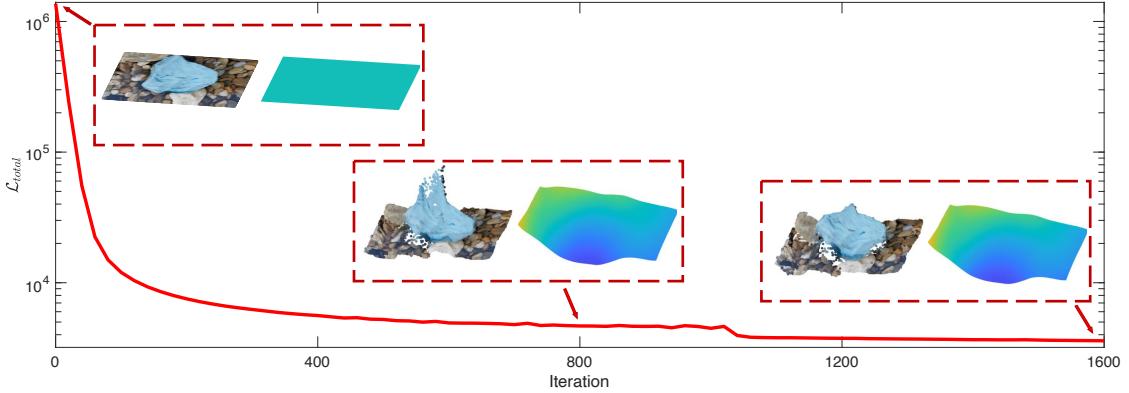


Figure 6.4: The evolution of the objective function versus iterations for the data shown in Fig. 6.1. The structures of water surfaces (one frame) and scene geometry are all initialized as planar surfaces. The objective function is effectively reduced, and accordingly, the parameters are progressively optimized, and yield a good representation of the scenes after 1600 iterations.

law and finding the underwater scene geometry via multi-time triangulation. Given the framework with underwater point clouds and time-varying water surfaces, the loss of the entire model is computed through forward propagation following the designed pipeline. Afterwards, the variables are simultaneously optimized from the back-propagated gradients from the model loss. The objective function of the framework is defined as:

$$\mathcal{L}_{\text{total}} = \kappa_1 \mathcal{L}_{\text{distance}} + \kappa_2 \mathcal{L}_{\text{curvature}} + \kappa_3 \mathcal{L}_{\text{temporal}} + \kappa_4 \mathcal{L}_{\text{projection}}, \quad (6.10)$$

which is a weighted summation of distance loss, curvature loss, temporal loss and projection loss. In the optimization process, the surfaces and the scene geometry are all initialized as planar surfaces. All parameters are progressively and simultaneously updated, and finally the model converges at stationary points. Fig. 6.4 shows an example of the progress of the loss function over time, as well as the corresponding geometries at the beginning, in the middle, and at the end of the optimization process. In the following, we discuss the components of the loss function in detail.

6.4.4 Loss Function

Distance Loss. The optimized water surfaces and underwater point clouds should be consistent with the input video in the sense that refracted rays corresponding to the scene point in different frames (as identified by the optical flow) should actually meet at the same 3D point, which also coincides with a point in the 3D point cloud. This is achieved by minimizing the defined distance loss function:

$$\mathcal{L}_{\text{distance}} = \sum_{\mathbf{p} \in \mathcal{P}} \text{dis}(\mathbf{p}, \mathcal{S}_1, \dots, \mathcal{S}_T). \quad (6.11)$$

This distance loss term is adopted from Eq. [6.9], which is applied to all underwater point clouds. The structures of the underwater scene and the time-varying water surfaces are integrated in this term, which makes them codependent. Notice that this term is non-convex since there always exists a single-view depth-normal ambiguity [12].

Curvature and Temporal Loss. Applying additional regularization terms on the water surface is a common strategy to encourage a smooth and temporal coherent reconstruction. Spatial and temporal smoothness are two basic features for dynamic water surfaces. We employ the mean curvature loss to govern its spatial smoothness, which is approximated as:

$$\mathcal{L}_{\text{curvature}} = \sum_{t=1}^T \left\| \frac{\partial^2 \mathbf{c}_t}{\partial x^2} \right\|_2^2 + \left\| \frac{\partial^2 \mathbf{c}_t}{\partial y^2} \right\|_2^2. \quad (6.12)$$

We further use the wave equation as a rough model governing the evolution of the water surface over time. Therefore, the temporal loss can be written as:

$$\mathcal{L}_{\text{temporal}} = \sum_{t=2}^{T-1} \left\| \frac{\partial^2 \mathbf{c}_t}{\partial t^2} - c^2 \left(\frac{\partial^2 \mathbf{c}_t}{\partial x^2} + \frac{\partial^2 \mathbf{c}_t}{\partial y^2} \right) \right\|_2^2, \quad (6.13)$$

where c is the magnitude of the velocity. The applied parameterization strategy makes these two loss functions easy to compute, and ensures that the gradients with respect to the time-varying surfaces can be propagated in the framework.

Projection Loss. Imposing regularization terms on the underwater scene geometry is not trivial as for the water surface. The rays originating from adjacent underwater scene points interlace after passing through wavy water surface, thus their projected image pixels may not be adjacent [68]. Imposing spatial smoothness simply on the captured image pixels is not effective.

Clearly, this adjacency relationship holds when the water surface is flat or there is no interference of water (a standard 3D-to-2D perspective projection). It would be feasible to enforce spatial smoothness on the virtually projected heightmaps of the point clouds – the projected heightmap synthesized from flat water surface or the projected heightmap synthesized from direct perspective projection. However, generating the first heightmap involves an iterative projection operation as bending of light paths occurs at the water-air interface. In contrast, generating the second heightmap is relatively easier, a linear operator projects the 3D point clouds to the image plane. We choose the second option in our implementation to regularize recovered underwater point clouds. Specifically, we define \mathbf{h} as the synthesized heightmap projected from the estimated point clouds, and the ℓ_1 norm of its gradient is defined as the projection loss, which could be written as:

$$\mathcal{L}_{\text{projection}} = \|\nabla \mathbf{h}\|_1. \quad (6.14)$$

This term proved effective in smoothing out the noise while preserving edge information in the recovered point clouds.

6.5 Results and Discussions

The cubic B-spline coefficients and confidence masks were pre-computed and were stored in sparse matrices. We implemented our proposed pipeline depicted in Fig. 6.3 in PyTorch. We used Adam [104] for optimization. The learning rate for underwater point clouds is set to $5e^{-2}$, and the learning rate for water surfaces is set to $1e^{-3}$ and is reduced to $1e^{-4}$ after 1000 iterations. The program takes around 2 hours to process a total of 120 frames with 30,000 reconstructed points, using 1600 iterations on a Nvidia 2080 Ti GPU.

6.5.1 Synthetic Experiments

We first conduct synthetic experiments to validate the proposed reconstruction framework. We use the Middlebury dataset [105] to model the 3D underwater scene (20 different scenes), and the dynamic water surfaces are represented as a sum of multiple waves from point sources. We set the focal length of the camera to 1 unit, and the pixel size to 0.01 units. The camera is vertically placed above the water at a distance of 20 units. The depth of the underwater scene ranges from 40 to 60 units from the camera. The refractive index of water is fixed at 1.33. A sequence of 120 consecutive distorted images is generated by ray tracing. Taking one frame from the sequence as a reference, we compute the optical flows to all other frames using the flow estimation model PWC-Net [106].

The single-view depth-normal ambiguity exists on the depth and normal of the water surface [12], and it forms a non-convex reconstruction problem, there could be a set of solutions satisfying the constraints. By fixing the time-varying water surfaces, the underwater point clouds become deterministic. Therefore, we can quantitatively evaluate the reconstruction accuracy on the point clouds with known water surface and study the effectiveness of the confidence mask and projection loss. This reconstruction problem can still be solved in the same framework.

Table 6.1: Average Euclidean distance between the true and the estimated point clouds on synthetic data with different parameter and experimental settings.

Number of Frames	30	60	120
w/o Projection Loss	0.286	0.265	0.255
w/o Confidence Mask	0.278	0.258	0.249
w/o both	0.306	0.284	0.271
Full model	0.254	0.233	0.227

Point Cloud Reconstruction

For evaluation, we use the metric of average Euclidean distance measured between the true and the estimated positions of the point clouds. We set $\kappa_1 = 1$, $\kappa_2 = \kappa_3 = 100$ and $\kappa_4 = Te^{-5}$ for all experiments. Table 6.1 shows the average Euclidean distance under different experimental settings on the synthetic experiments. In general, our model yields higher reconstruction accuracy for the point clouds when using more frames. This is similar to the behavior of multi-view 3D reconstruction methods. Erroneous optical flow estimates contribute to uncertainty in the point cloud, and using more input frames provides more diverse viewing angles of the scenes, which reduces the noise.

The use of a confidence mask and the total-variation regularizer on the projected heightmap of the point clouds also proves effective in addressing this uncertainty. The confidence masks filter out those erroneous viewing angles, and the regularizer further smooths out the depth of the estimated point clouds. We find that the error in the computed flow vectors mainly concentrates in the boundary areas. For some frames, the water surface refracts the rays outside the regular field of view of the camera, so that the computed flow vectors become unreliable. For these points, the camera can only provide one-sided viewing information, resulting in very small baselines for triangulation.

We conduct another synthetic experiments with varying averaged water depth. We put the underwater scene close to the steady water surface and then move the scene

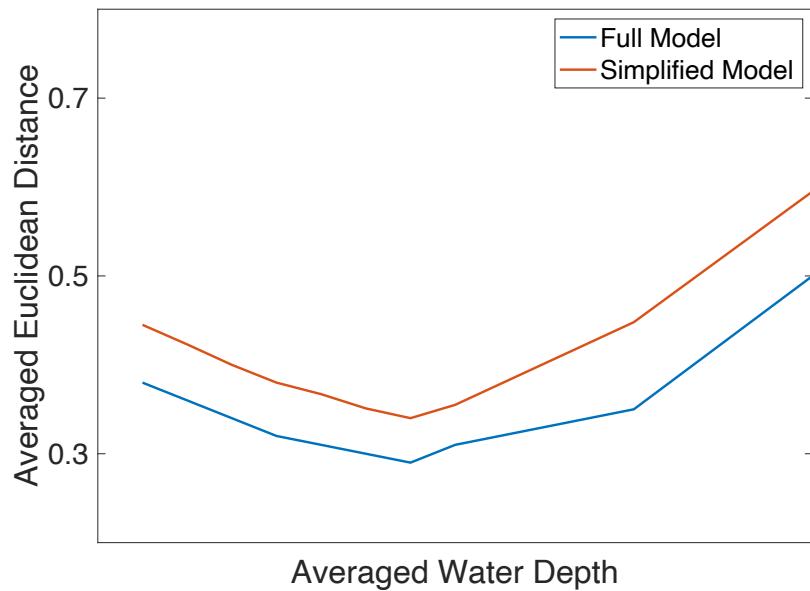


Figure 6.5: Average Euclidean distance between the true and the estimated point clouds on synthetic data with varying averaged water depth. When the water depth increases, camera observes stronger distortions, which yields a larger baseline for triangulation. Therefore, the average error of the reconstructed point clouds decreases at the early phase. However, when further increasing the water depth, the distortion becomes severe, in which case a precise image registration cannot be achieved. The reconstruction error will increase dramatically.

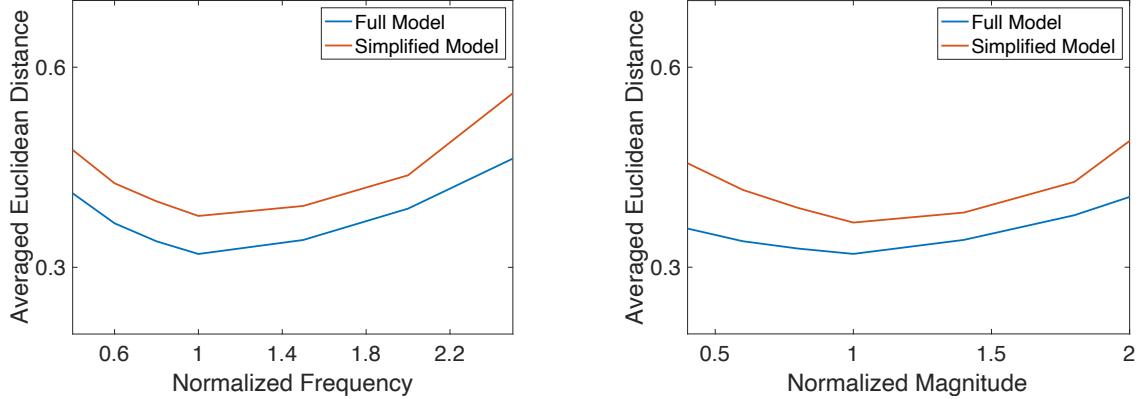


Figure 6.6: Average Euclidean distance between the true and the estimated point clouds on synthetic data with varying wave frequency and wave amplitude (the frequency and amplitude are normalized with respect to the frequency and magnitude which provide the best recovery, respectively).

gradually further away from the surface. The reconstruction error of the estimated point clouds is shown in Fig. 6.5. When the underwater scene is close to the water surface, the estimation error is reduced as increasing the water depth, as it yields a larger baseline for triangulation – a stronger distortion is observed at the camera side. When the water depth reaches a specific value, further increasing the water depth yields dramatically downgraded reconstruction error. When the camera captures rather strong distortions where a precise image registration cannot be fulfilled, the rays used for triangulation cannot be properly determined, which, therefore, breaks down the point cloud estimation.

Further, we conduct synthetic experiments with varying wave frequency and amplitude. The reconstruction error of the estimated point clouds is shown in Fig. 6.6. When the wave frequency or wave amplitude increases, camera observes stronger distortions, which yields a larger baseline for triangulation. Therefore, the average error of the reconstructed point clouds decreases at the early phase. However, when further increasing the frequency or amplitude, the distortion becomes severe, in which case a precise image registration cannot be achieved. The reconstruction error will increase accordingly. The above experiments also reveal that our proposed full model

Table 6.2: Quantitative results compared with a multi-camera approach [68] on point cloud estimation.

	Ours		[68]
Number of cameras	1		9
Number of frames	30	60	120
AED	0.254	0.233	0.227

consistently outperforms the simplified model without using the projection loss and confidence mask. This further verifies the effectiveness of these two terms at the point cloud reconstruction.

we also conduct a numerical comparison with a multi-camera system [68]. They use a 3×3 camera array to capture the scenes from different viewpoints. We implement their algorithm for point cloud estimation as no source code is publicly available. In their system, the parameters for the camera array are pre-calibrated and the reconstruction solves the scene geometry only. This is a standard multi-view 3D reconstruction approach which can provide a robust and accurate estimation. The baseline between adjacent cameras is set to 5 units (this could be 5-20 times larger than using water distortion for multi-view triangulation). Table 6.2 shows the numerical comparisons using average Euclidean distance (AED) as error metric. As expected, using the multi-camera system with a wide baseline yields a more accurate 3D geometry reconstruction, but it heavily relies on the acquisition system, which is expensive to build and calibrate such a system. We also show qualitative comparisons in Fig. 6.7.

Water Surface Reconstruction

The primary focus of our work is the reconstruction of the underwater scene. However, in the process of this reconstruction, we also estimate the shape of the deforming water surface. Here, we conduct a simulated quantitative evaluation of this aspect, and compare our method to a SOTA single-camera fluid reconstruction approach [14].

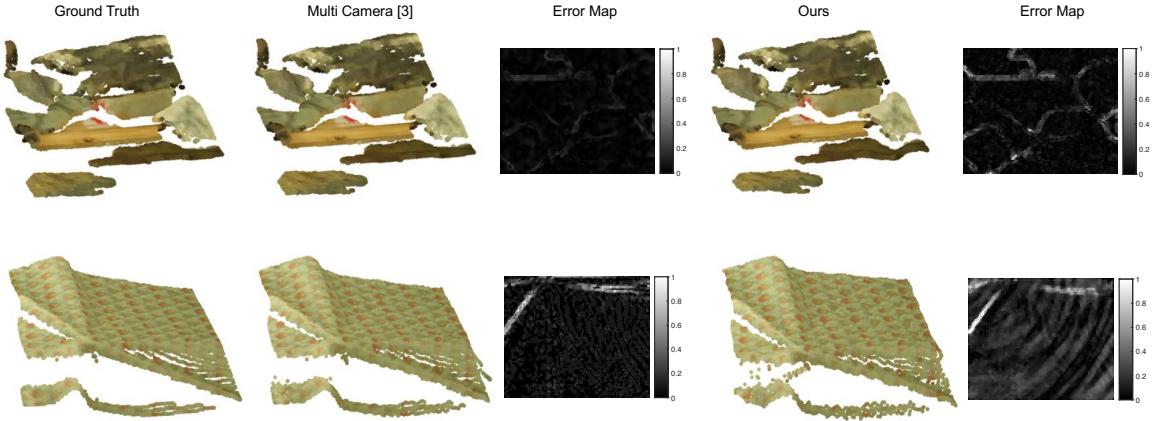


Figure 6.7: Qualitative comparisons between the reconstructed geometry and ground truth on synthetic data. Our recovered results exhibit a high degree of consistency with the ground truth with regard to geometric structures. The overall reconstruction error is comparably higher than using a multi-camera system (a 3×3 pre-calibrated camera array), but our acquisition system is simple and free of calibration.

Similar to our hardware setup, they use a single camera to capture refractive images of the background pattern, and the structure of fluid surfaces is estimated by a trained neural network. Their method simplifies the required equipment as compared to prior work, however, an undistorted frame is still required and serves as reference. For evaluation, the refractive images are rendered using the sample reference patterns as employed in [14] and synthetically generated time-varied water surfaces.

Table 6.3 shows the quantitative results on the recovered depth and surface normal. We use the root mean square error (RMSE) and absolute relative error (Abs Rel) as error metrics for estimated depth, and the root mean square error (RMSE) and average angular error (AAE) as error metrics for surface normal evaluation. Angular error (in degrees) measures the degrees between ground truth and the estimated surface normal. We find that our proposed model-based approach outperforms the existing single-camera fluid estimation method. It reveals that the temporal regularizer plays a significant role as it explicitly models a physical evolution of the water surface over time, and ties together all frames. The compared method employs a recurrent neural network module to encourage temporal consistency, and it may not

Table 6.3: Quantitative results between the true and the estimated water surface.

Method	Depth	
	Metric	
	RMSE	Abs Rel
FSRN [14]	0.103	0.087
w/o spatial Loss	0.097	0.079
w/o temporal Loss	0.129	0.112
w/o both	0.143	0.118
Ours	0.086	0.065

Method	Normal	
	Metric	
	RMSE	AAE
FSRN [14]	0.064	4.33°
w/o spatial Loss	0.038	3.01°
w/o temporal Loss	0.046	3.27°
w/o both	0.049	3.69°
Ours	0.030	2.21°

exactly retrieve the physical process of fluid flows. We also notice that even though the water surface is modeled as a cubic b-spline surface, which implicitly enforces the spatial smoothness, we do find an explicit spatial smoothness term further improves the reconstruction.

6.5.2 Experiments in the Lab

Next, we validate our method on real experiments conducted in a laboratory environment. We used a FLIR GS3-U3-41C6C camera with a 50 mm lens (the lens distortion should be calibrated to validate pinhole camera model). The camera was placed on top of a tank, pointing down vertically, at a distance of ca. 300 mm to the flat water surface. The water waves were introduced by pouring a cup of water into the tank. We used an aperture of f/6.0. The video was recorded at 60 fps with a resolution of 1024 × 1024 and we captured 120 frames in total for processing.

Fig. 6.8 visualizes one distorted image, and the recovered underwater point clouds.

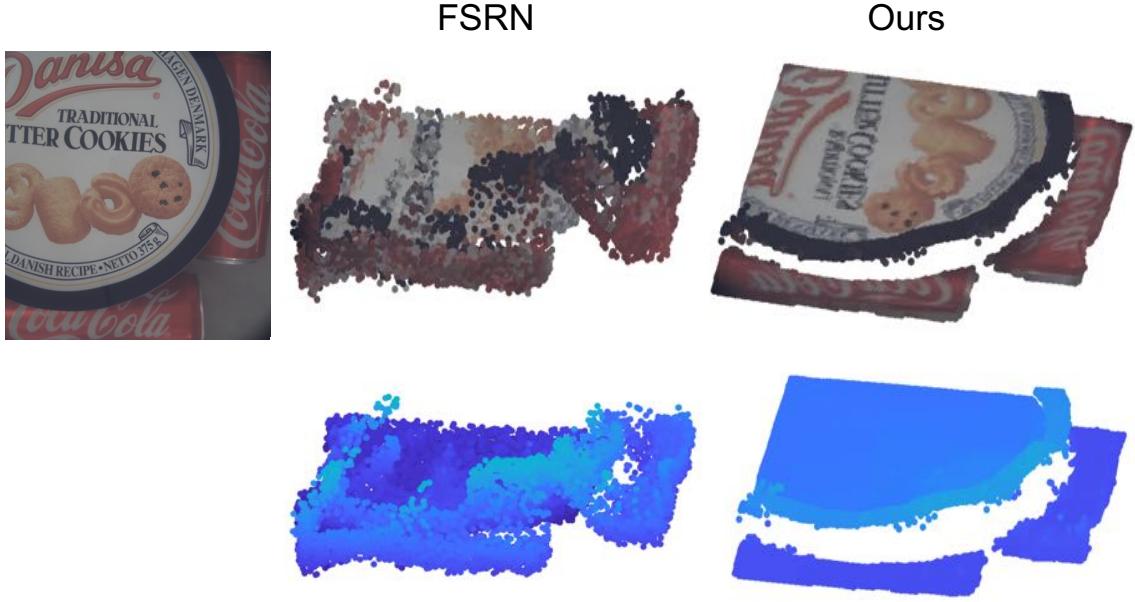


Figure 6.8: Comparisons of the estimated underwater scenes. A modified approach from [14] serves as a baseline. Our method could produce a reasonable recovery of the underwater scene. Notice that the baseline method still requires an additional undistorted frame as reference.

To the best of our knowledge, no existing work could retrieve the underwater geometry using the same hardware configuration as ours. We modify the SOTA single-camera fluid surface estimation method [14] as a baseline method. The time-varying surfaces are first estimated by their model, and then we feed the surfaces into the multi-time triangulation framework to estimate the underwater geometry. Fig. 6.8 shows that decoupling the estimation of water surfaces and underwater scene could not yield a reasonable reconstruction on the underwater environments, while our integrated model delivers an adequate recovery. It needs to point out that [14] exploits a simple setup (like ours), and requires a reference frame captured without distortion. This reference frame will be unavailable in an uncontrolled environment, e.g. in the wild.

We further show the comparisons with the underwater geometry from a 3D scan. Fig. 6.9 reveals the qualitative comparisons between the scanned models and our reconstructions. The scanned models were obtained without water interference. The reconstructions from a 3D scan exhibit finer recovery of its geometrical structure,

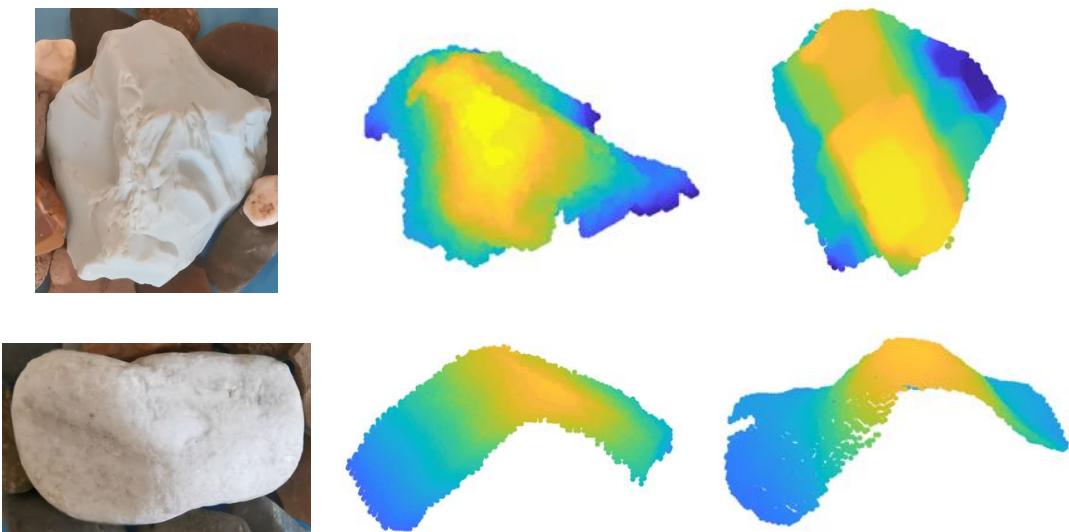


Figure 6.9: Visual comparisons between the 3D scanned models (serve as ground truth) and our reconstructions through moving water surfaces. From left to right, the distorted frames from the video for two stone scenes, the ground truth 3D structures measured using a 3D scanner and our reconstructed geometry. The averaged absolute error on the projected depth map is 4.24 mm for the first scene and 3.01 mm for the second scene. Notice that due to the non-convexity of the problem, our solution is a locally reasonable representation of the scenes. Therefore, we scale our reconstructions with respect to the scanning results.

however, our framework could generate an adequate representation of the scenes under such sophisticated conditions with a simple hardware setup.

We also show a qualitative comparison for the recovered water surface between ours and FSRN [14] in Fig. 6.10. The results demonstrate that the recovered structures from both methods are overall consistent. However, our method explicitly enforces smoothness, and the generated normal map exhibits fewer noise patterns. This makes our estimation to be more accord with the physical characteristics. Due to relatively lower estimation accuracy on the water surface for the compared method, inferring the 3D geometry of the underwater scene is prone to error. We also visualize the reconstructed point clouds with color-coded depth in Fig. 6.8.

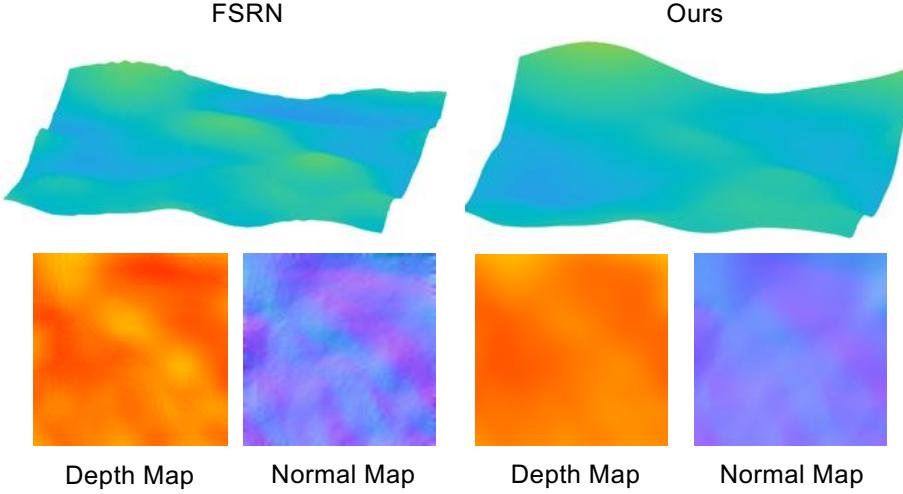


Figure 6.10: Qualitative comparisons with [14] for water surface estimation. The depths are normalized for fair comparisons. Their method requires an additional undistorted frame as reference. This reference frame will be unavailable in an uncontrolled environment, e.g. in the wild. The recovered shapes are overall consistent, while the reconstruction from ours is smoother with fewer noise-like features.

6.5.3 Experiments in the Wild

Finally, we tested our reconstruction model outside the lab. Our method neither requires a complicated hardware setup nor does it impose impractical assumptions like other approaches. We can easily capture data for processing in the wild. The first experiments were conducted for scenarios in a large public fountain. We captured 1080p videos at 60 fps using a smartphone held by a tripod, and downsampled the images by a factor of 8 yielding 240×135 underwater point clouds. The smartphone was placed above the water surface at a distance around 20 cm, and the depth of the underwater scenes roughly range from 20 – 35 cm. The data was captured under various weather conditions where the waves were driven by natural winds of different strengths.

Fig. 6.11 visualizes captured images at two frames, corresponding reconstructed surface structures, the side and top views of the recovered underwater scene geometry, which is represented by a set of discrete point clouds. Two different examples

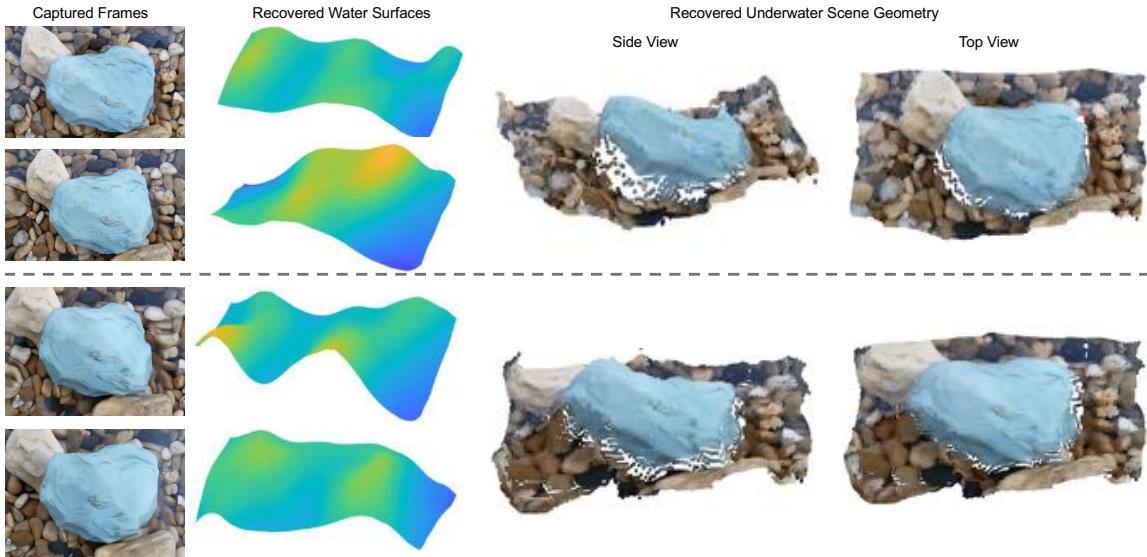


Figure 6.11: Reconstructions of additional two scenes captured in the public fountain environment. They were collected in different weather conditions with relatively mild winds (top) and strong winds (bottom), respectively. From left to right: Two captured frames and corresponding recovered surface shapes, the side and top views of the recovered underwater scene geometry. Both results exhibit an adequate representation, even with strong water distortions. We recommend to view the 3D geometry.

correspond to videos captured under a relatively mild (top) and strong (bottom) fluid disturbance, respectively. The recovered point clouds exhibit a faithful representation of the underwater scenes which are consistent with the expectation. The recovered time-varying water surfaces also agree with the observed distortion patterns even in conditions with rather strong fluctuations. Fig. [6.1] shows an additional reconstruction result, where data was collected in the same fountain environment.

Fig. [6.12] shows qualitative comparisons of the recovered scene geometry with and without using the confidence mask and projection loss. As expected, the recovered scene geometry using the full model tends to be more smooth and some fine details, e.g. edge of the objects, are better preserved. The projected heightmap also demonstrates a significant improvement when both strategies are applied.

Fig. [6.13] shows two more data sets which were captured by a sea shore. The captured images reveal that reconstructing the scenes under salt water is more chal-

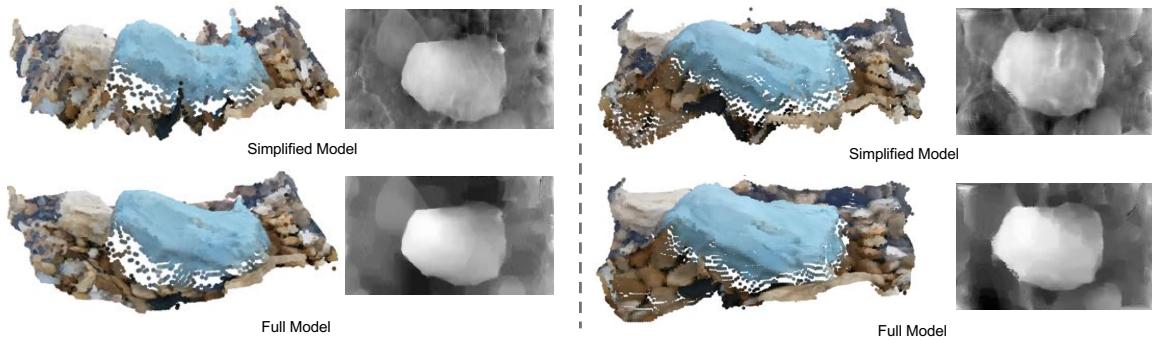


Figure 6.12: Qualitative comparisons on the reconstructed underwater scenes and projected heightmaps using (full model) and without using the mask strategy and projection loss (simplified model). Left refers to the data in Fig. 6.11 top, and right refers to the data in Fig. 6.11 bottom. In both cases, the full model delivers more smooth, yet finer-detail preserved, geometric structures.



Figure 6.13: Reconstructions of two data sets collected in salt water environments. The seawater becomes more turbid and the geometric structures under seawater is more complicated, while our method can still realize a reasonable reconstruction. This shows that our method is effective, and robustly handles reconstructions in rather complicated environments.

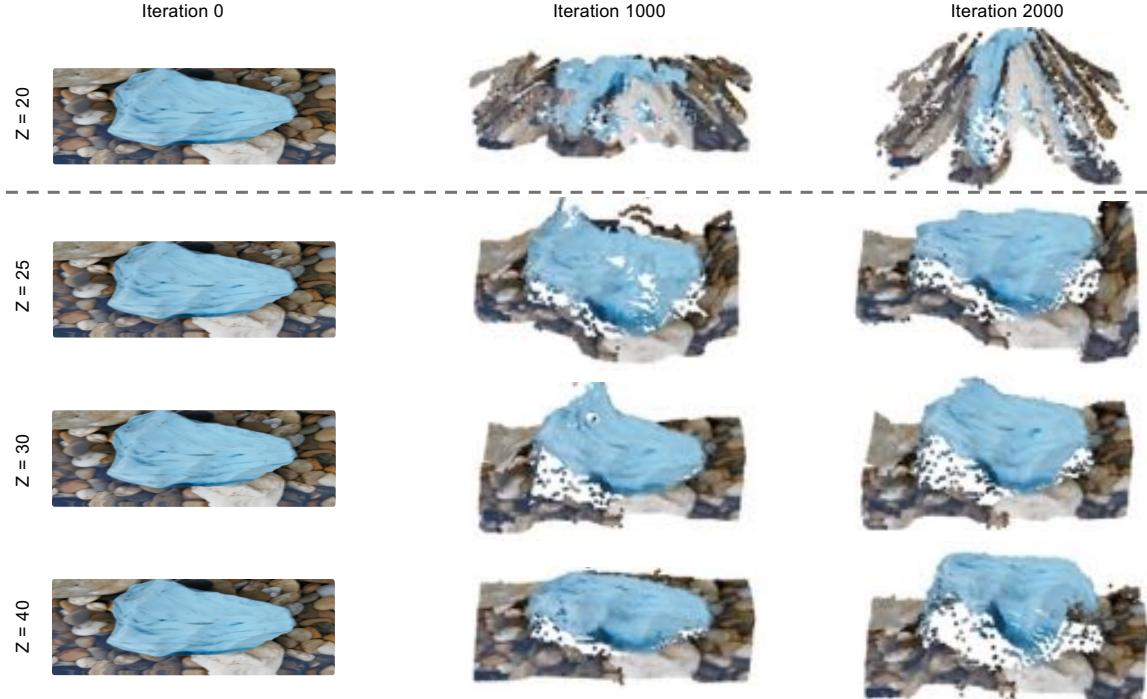


Figure 6.14: The reconstructions from different initial points and also the failure case at a degenerated initialization (top). The initialization of the underwater scene geometry is a planar surface with different axial depths.

lenging as the water is more turbid. However, our method can still realize a robust and adequately good recovery of the underwater scene geometry in this rather difficult experiment. This demonstrates that our method is robustly handling scenes with some level of turbidity, which is a common effect in natural bodies of water.

6.6 Discussion

6.6.1 Different Initializations

Because of the single-view depth-normal ambiguity on the recovered water surface, the proposed global optimization problem is non-convex. Different initializations will drive the framework to different local minimums. In most initial points, the framework finds a reasonable representation of the scenes. However, we did find degenerated cases with some initializations. As discussed above, the initialization



Figure 6.15: A failure case for the fountain scene as shown in Fig. 6.11. The water waves were driven by a rather strong wind, and it exhibits a vortex structure. Our method fails to produce a quality representation as a precise correspondence matching cannot be satisfied.

of the underwater scene geometry is a planar surface with different axial depths. In Fig. 6.14, we show the reconstructions with different initial values which yields similar adequate representations, and also a failure case with improperly selected initial value. The initial value should also vary with different reconstruction data.

6.6.2 Failure Case

Fig. 6.15 shows a failure case for the proposed method. The data was captured in the same fountain environment as shown in Fig. 6.11, but on a rather windy day. One frame of the images exhibits that the background scenes are hugely distorted by a vortex-like water wave. Under this condition, a precise image registration becomes problematic, and therefore the reconstruction of the scene geometry fails as well. Our reconstruction framework relies on a preprocessed dense and precise correspondence matching. When the waves are driven by excessively strong external force and become choppy, they are no longer in accord with the imposed smoothness regularizers, and then our method fails to recover geometry of adequate quality.

6.6.3 Relationship to Structure-from-Motion (SfM)

SfM utilizes a series of images taken from different viewpoints to reconstruct the 3D structure of the scene. The images are usually taken with a moving camera. The

reconstruction is realized with bundle adjustment, which jointly estimates camera parameters and scene geometry by solving a non-linear least square problem. By comparison, in our problem multiple viewpoints are introduced by the non-stationary water surface fluctuations, and we propose a novel differentiable framework to simultaneously estimate the structure of non-stationary water surfaces and underwater scene geometry.

The problem we studied differs from SfM mainly in the following three aspects:

- Camera parameters in SfM can be represented by a 4×4 matrix, a low-parameter model with respect to scene geometry. Since camera parameters are low-dimensional, they can be estimated by matching feature points, e.g. SIFT features [107]. On the other hand, to fully characterize a water surface, the degrees of freedom can approach the same size as the underwater scene for choppy water. To estimate water surfaces, a dense correspondence match is required, which is prone to error especially when the surface distortion is strong. Therefore, the estimation of camera parameters is more robust and less ill-conditioned.
- In SfM, the projection from 3D coordinates to the image plane is linear, and the problem could be solved via a relatively simple minimization formula. This projection becomes non-linear as the lights pass through a refractive interface, which makes the problem cannot be tackled with a scheme similar to bundle adjustment.
- In general, the baseline from moving a camera can be much larger than that caused by water surface fluctuation. Knowing that small baseline will reduce the depth estimation accuracy, which is approximately inversely proportional [108], employing water surface fluctuations for multi-view triangulation is more noise sensitive and theoretically produces scene geometry in lower depth accuracy.

Nonetheless, our proposed framework, which integrates ray casting, Snell’s law and multi-time triangulation, along with the regularization terms for time-varying water surfaces and underwater scene geometry, tackles the difficulties arising from a well-studied solution to a SfM problem. We demonstrate that this novel framework is capable of recovering fully characterized time-varying water surfaces and 3D background scenes using a single camera.

6.7 Conclusion

This chapter presents a novel approach to reconstruct the 3D shape of underwater scene via a single camera. This is realized by the time-varying distortions from moving water surface which provides a multi-time triangulation. We propose a dedicated differentiable framework accounting for the ray casting, ray refraction, and multi-time triangulation. This framework integrates the dynamic water surfaces and underwater scene geometry as inputs, such that both parameters, with planar surfaces as initialization, are progressively optimized from specially constructed and proven effective loss functions. Extensive in-the-wild experimental results, even tested in the salt water environments, validate the effectiveness and robustness of the proposed approach.

Chapter 7

Concluding Remarks

7.1 Summary

The contribution presented in this dissertation push the boundary of joint design of optics and tailored reconstructions for 3D-3C fluid velocity field estimation and simultaneously retrieving the structures of underwater surface and underwater environments. However, some open problems and challenges remain, as also acknowledged in the previous chapters. Furthermore, advances in measurement hardware and algorithmic development open up new possibilities for future research directions.

7.2 Future Work

When Deep Learning Meets Fluids. In recent years, deep learning has gathered huge popularity in the computer vision and computer graphics community. It has shown its prominent ability in extracting 3D representation of scenes and the registration map of consecutive motions. The cores of 3D PIV lie in extracting the time-dependent 3D particle distribution fields and the motion fields of the non-stationary particle fields. The 3D reconstruction of the particle fields and the computation of the motion fields are highly related to the successful usage of deep learning in the computer vision community.

Recent efforts have been devoted to applying the neural network architectures to extract velocity fields from 2D PIV data, yielding 2D-2C fluid flows [109, 110, 111]. While it exhibits specific characteristics when applying these types of technologies to

fluid imaging. The distributions of the particle fields are generally sparse, and the presence of the particle fields at different time steps is constrained by the inherent motion fields, which is governed by the Navier-Stokes equations. Deep learning offers plentiful techniques which are potentially capable of extracting information from data that is related to the underlying fluid mechanics. How to incorporate these physical models into the modern deep learning framework would be a very interesting future research direction. Besides, to fully characterize the fluid structures, 3D-3C velocity fields are expected. Extending the 2D PIV neural network to 3D is non-trivial as the underlying fluid flows exhibit more complicated structures in 3D and the size of the processing data rapidly increases which could easily runs out of the memory. Processing the PIV data in 3D with neural networks is still an open problem, and tackling this problem is also highly interested by the fluid mechanics community.

Furthermore, current approaches form a pipeline to reconstruct the fluid flows – computing the motion field on the pre-computed consecutive particle fields. As described before, the consecutive particle fields and the motion field between them are constrained by underlying physical constraints. An end-to-end deep learning framework for both particle fields and motion field estimation would be desired to integrate the inherent complex physical models. Owing to its powerful information extraction ability, applying the deep learning technique can enrich, and even transform the field of fluid imaging.

Differential Transparent Object Reconstruction. We have also presented a first attempt to recover the refractive surface and the background geometry in the wild using a single camera. It greatly simplifies the hardware setups and relaxes impractical assumptions as imposed by alternative approaches. However, our current approach is limited to settings where the light path is refracted only once by a refractive surface. Generalizing the model to more complicated conditions could be an interesting avenue for future work, for instance reconstructing glass or plastic objects with a minimum

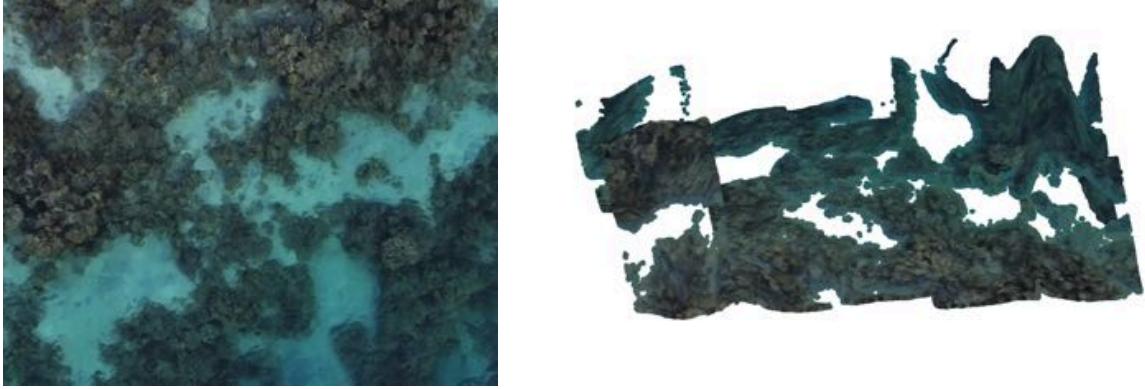


Figure 7.1: Left: Coral reefs captured using a drone. Right: preliminary reconstructions using the proposed framework. This is not an adequately good recovery of the underwater scenes because of the breeze ripples in the ocean surface. These ripples have high frequencies and complicated geometric structures, which are generally hard to be accurately recovered at current stage.

of two refractions [60, 62, 61], or reconstructing inhomogeneous fluids [11, 64]. For the time being, there are no reliable solutions to recover these kinds of transparent objects in the wild.

Implicit Representation. One major limitation for the proposed differentiable framework for reconstruction through moving water surfaces is that all the intermediate results need to be stored in order to be accessible when computing back-propagated gradients. To process 30,000 points in a sequence of 100 frames, it costs around 10 GB memory, which is close to the memory limit of modern GPUs. Fig. 7.1 shows a coral reef scene captured by a drone. In order to process these over 300,000 points in a 100-frame video, we have to split the entire scene into smaller regions and perform reconstruction on each of them independently to fit the scene into the GPUs. Using implicit shape representation [112, 113, 114] will be a promising solution to represent the dynamic water surfaces and underwater environment, as well as reducing the memory cost. The core idea is to use two fully connected neural networks to encode the water surface and underwater scene, respectively, by mapping the coordinate of the scenes to an implicit representation, e.g. signed distance field

or occupancy field. Furthermore, we can extend the representing of the time-varied water surfaces in a space-time encoding, by optimizing a neural network that encodes both the coordinate and the time. Such a representation has a great potential to model complicated geometric scenes in high resolution.

REFERENCES

- [1] G. Elsinga, F. Scarano, B. Wieneke, and B. van Oudheusden, “Tomographic particle image velocimetry,” *Experiments in Fluids*, 2006.
- [2] F. Scarano, “Tomographic PIV: principles and practice,” *Measurement Science and Technology*, 2012.
- [3] M. Novara, K. J. Batenburg, and F. Scarano, “Motion tracking-enhanced mte-mart for tomographic piv,” *Measurement science and technology*, 2010.
- [4] K. Lynch and F. Scarano, “An efficient and accurate approach to mte-mart for time-resolved tomographic piv,” *Experiments in Fluids*, 2015.
- [5] F. Scarano and P. Moore, “An advection-based model to increase the temporal resolution of piv time series,” *Experiments in fluids*, 2012.
- [6] J. F. Schneiders and F. Scarano, “Dense velocity reconstruction from tomographic ptv with material derivatives,” *Experiments in fluids*, 2016.
- [7] A. A. Aguirre-Pablo, M. K. Alarfaj, E. Q. Li, J. F. Hernández-Sánchez, and S. T. Thoroddsen, “Tomographic particle image velocimetry using smartphones and colored shadows,” *Scientific Reports*, 2017.
- [8] C. Cierpka, R. Hain, and N. A. Buchmann, “Flow visualization by mobile phone cameras,” *Experiments in Fluids*, 2016.
- [9] K. Lasinger, C. Vogel, T. Pock, and K. Schindler, “3d fluid flow estimation with integrated particle reconstruction,” *International Journal of Computer Vision*, 2020.
- [10] G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, and W. Heidrich, “Space-time tomography for continuously deforming objects,” *ACM Transactions on Graphics*, 2018.
- [11] B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel, “Time-resolved 3d capture of non-stationary gas flows,” *ACM transactions on graphics*, 2008.
- [12] N. J. Morris and K. N. Kutulakos, “Dynamic refraction stereo,” *IEEE transactions on pattern analysis and machine intelligence*, 2011.

- [13] Y. Qian, M. Gong, and Y.-H. Yang, “Stereo-based 3d reconstruction of dynamic fluid surfaces by global optimization,” in *CVPR*, 2017.
- [14] S. Thapa, N. Li, and J. Ye, “Dynamic fluid surface reconstruction using deep neural network,” in *CVPR*, 2020.
- [15] G. Wetzstein, D. Roodnick, W. Heidrich, and R. Raskar, “Refractive shape from light field distortion,” in *ICCV*, 2011.
- [16] J. Ye, Y. Ji, F. Li, and J. Yu, “Angular domain reconstruction of dynamic 3d fluid surfaces,” in *CVPR*, 2012.
- [17] S. Hasinoff and K. Kutulakos, “Photo-consistent Reconstruction of Semitransparent Scenes by Density-sheet Decomposition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
- [18] I. Ihrke and M. Magnor, “Image-based tomographic reconstruction of flames,” in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2004.
- [19] T. Hawkins, P. Einarsson, and P. Debevec, “Acquisition of Time-Varying Participating Media,” *ACM transactions on graphics*, 2005.
- [20] J. Gu, S. Nayar, E. Grinspun, P. Belhumeur, and R. Ramamoorthi, “Compressive Structured Light for Recovering Inhomogeneous Participating Media,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.
- [21] J. Gregson, M. Krimerman, M. B. Hullin, and W. Heidrich, “Stochastic tomography and its applications in 3d imaging of mixing fluids,” *ACM Transactions on Graphics*, 2012.
- [22] J. Gregson, I. Ihrke, N. Thuerey, and W. Heidrich, “From capture to simulation: connecting forward and inverse problems in fluids,” *ACM Transactions on Graphics*, 2014.
- [23] B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel, “Time-resolved 3D Capture of Non-stationary Gas Flows,” *ACM transactions on graphics*, 2008.
- [24] B. Atcheson, W. Heidrich, and I. Ihrke, “An evaluation of optical flow algorithms for background oriented schlieren imaging,” *Experiments in fluids*, 2009.
- [25] G. Zang, R. Idoughi, C. Wang, A. Bennett, J. Du, S. Skeen, W. L. Roberts, P. Wonka, and W. Heidrich, “Tomofluid: reconstructing dynamic fluid from sparse view videos,” in *CVPR*, 2020.

- [26] E. Franz, B. Solenthaler, and N. Thuerey, “Global transport for fluid reconstruction with learned self-supervision,” in *CVPR*, 2021.
- [27] R. J. Adrian, “Twenty years of particle image velocimetry,” *Experiments in fluids*, 2005.
- [28] R. Adrian and J. Westerweel, *Particle image velocimetry*. Cambridge University Press, 2011.
- [29] M. Raffel, C. E. Willert, F. Scarano, C. J. Kähler, S. T. Wereley, and J. Kompenhans, *Particle image velocimetry: a practical guide*. Springer, 2018.
- [30] K. Hinsch, “Holographic particle image velocimetry,” *Measurement Science and Technology*, 2002.
- [31] N. Chen, C. Wang, and W. Heidrich, “Snapshot space–time holographic 3d particle tracking velocimetry,” *Laser & Photonics Reviews*, 2021.
- [32] C. Willert and M. Gharib, “Three-dimensional particle imaging with a single camera,” *Experiments in Fluids*, 1992.
- [33] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, “Light field microscopy,” *ACM transactions on graphics*, 2006.
- [34] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” *Computer Science Technical Report CSTR*, 2005.
- [35] K. Lynch, T. Fahringer, and B. Thurow, *Three-dimensional particle image velocimetry using a plenoptic camera*. American Institute of Aeronautics and Astronautics (AIAA), 2012.
- [36] I. Kimura, Y. Kohno, T. Ogasawara, and T. Takamori, “Measurement of three dimensional velocity vectors in a flow field using a color spectrum,” *Transactions of the Society of Instrument and Control Engineers*, 1991.
- [37] T. McGregor, D. Spence, and D. Coutts, “Laser-based volumetric colour-coded three-dimensional particle velocimetry,” *Optics and Lasers in Engineering*, 2007.
- [38] S. Pick and F.-O. Lehmann, “Stereoscopic PIV on multiple color-coded light sheets and its application to axial flow in flapping robotic insect wings,” *Experiments in Fluids*, 2009.
- [39] T. Watamura, Y. Tasaka, and Y. Murai, “LCD-projector-based 3D color PTV,” *Experimental Thermal and Fluid Science*, 2013.

- [40] A. Aguirre-Pablo, A. Aljedaani, J. Xiong, R. Idoughi, W. Heidrich, and S. Thoroddsen, “Single-camera 3d ptv using particle intensities and structured light,” *Experiments in Fluids*, 2019.
- [41] A. Prasad, “Particle image velocimetry,” *Current science*, 2000.
- [42] M. Stanislas, K. Okamoto, C. Kähler, J. Westerweel, and F. Scarano, “Main results of the third international piv challenge,” *Experiments in Fluids*, 2008.
- [43] B. Horn and B. Schunck, “Determining optical flow,” *Artificial Intelligence*, 1981.
- [44] T. Liu and L. Shen, “Fluid flow and optical flow,” *J. Fluid Mechanics*, 2008.
- [45] D. Heitz, E. Mémin, and C. Schnörr, “Variational fluid flow measurements from image sequences: synopsis and perspectives,” *Experiments in Fluids*, 2010.
- [46] T. Liu, A. Merat, M. Makhmalbaf, C. Fajardo, and P. Merati, “Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images,” *Experiments in Fluids*, 2015.
- [47] I. Herlin, D. Béreziat, N. Mercier, and S. Zhuk, “Divergence-free motion estimation,” in *ECCV*, 2012.
- [48] J. Yuan, C. Schörr, and G. Steidl, “Simultaneous higher-order optical flow estimation and decomposition,” *SIAM Journal on Scientific Computing*, 2007.
- [49] P. Ruhnau, A. Stahl, and C. Schnörr, “Variational estimation of experimental fluid flows with physics-based spatio-temporal regularization,” *Measurement Science and Technology*, 2007.
- [50] D. Heitz, P. Héas, E. Mémin, and J. Carlier, “Dynamic consistent correlation-variational approach for robust optical flow estimation,” *Experiments in fluids*, 2008.
- [51] N. Foster and D. Metaxas, “Modeling the motion of a hot, turbulent gas,” in *ACM transactions on graphics*, 1997.
- [52] R. Fedkiw, J. Stam, and H. Jensen, “Visual simulation of smoke,” in *ACM transactions on graphics*, 2001.
- [53] J. Stam, “Stable fluids,” in *ACM transactions on graphics*, 1999.
- [54] I. Ihrke, K. N. Kutulakos, H. P. Lensch, M. Magnor, and W. Heidrich, “Transparent and specular object reconstruction,” in *Computer Graphics Forum*, 2010.

- [55] G. Wetzstein, R. Raskar, and W. Heidrich, “Hand-held schlieren photography with light field probes,” in *ICCP*, 2011.
- [56] K. Tanaka, Y. Mukaigawa, H. Kubo, Y. Matsushita, and Y. Yagi, “Recovering transparent shape from time-of-flight distortion,” in *CVPR*, 2016.
- [57] B. Trifonov, D. Bradley, and W. Heidrich, “Tomographic reconstruction of transparent objects,” in *ACM SIGGRAPH 2006 Sketches*, 2006.
- [58] S.-K. Yeung, T.-P. Wu, C.-K. Tang, T. F. Chan, and S. Osher, “Adequate reconstruction of transparent objects on a shoestring budget,” in *CVPR*, 2011.
- [59] K. Han, K.-Y. K. Wong, and M. Liu, “Dense reconstruction of transparent objects by altering incident light paths through refraction,” *International Journal of Computer Vision*, 2018.
- [60] B. Wu, Y. Zhou, Y. Qian, M. Gong, and H. Huang, “Full 3d reconstruction of transparent objects,” *ACM transactions on graphics*, 2018.
- [61] J. Lyu, B. Wu, D. Lischinski, D. Cohen-Or, and H. Huang, “Differentiable refraction-tracing for mesh reconstruction of transparent objects,” *ACM Transactions on Graphics*, 2020.
- [62] Z. Li, Y.-Y. Yeh, and M. Chandraker, “Through the looking glass: Neural 3d reconstruction of transparent shapes,” in *CVPR*, 2020.
- [63] T. Hawkins, P. Einarsson, and P. Debevec, “Acquisition of time-varying participating media,” *ACM Transactions on Graphics*, 2005.
- [64] J. Gu, S. K. Nayar, E. Grinspun, P. N. Belhumeur, and R. Ramamoorthi, “Compressive structured light for recovering inhomogeneous participating media,” *IEEE transactions on pattern analysis and machine intelligence*, 2012.
- [65] J. Xiong, R. Idoughi, A. A. Aguirre-Pablo, A. B. Aljedaani, X. Dun, Q. Fu, S. T. Thoroddsen, and W. Heidrich, “Rainbow particle imaging velocimetry for dense 3d fluid velocity imaging,” *ACM Transactions on Graphics*, 2017.
- [66] J. Xiong, Q. Fu, R. Idoughi, and W. Heidrich, “Reconfigurable rainbow piv for 3d flow measurement,” in *ICCP*, 2018.
- [67] Y. Ji, J. Ye, and J. Yu, “Reconstructing gas flows using light-path approximation,” in *CVPR*, 2013.
- [68] Y. Qian, Y. Zheng, M. Gong, and Y.-H. Yang, “Simultaneous 3d reconstruction for water surface and underwater scene,” in *ECCV*, 2018.

- [69] M. Liu, R. Hartley, and M. Salzmann, “Mirror surface reconstruction from a single image,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.
- [70] Y. Zhang, M. Ye, D. Manocha, and R. Yang, “3d reconstruction in the presence of glass and mirrors by acoustic and visual fusion,” *IEEE transactions on pattern analysis and machine intelligence*, 2017.
- [71] T. Whelan, M. Goesele, S. J. Lovegrove, J. Straub, S. Green, R. Szeliski, S. Butterfield, S. Verma, R. A. Newcombe, M. Goesele *et al.*, “Reconstructing scenes with mirror and glass surfaces.” *ACM Transactions on Graphics*, 2018.
- [72] Y. Tian and S. G. Narasimhan, “Seeing through water: Image restoration using model-based tracking,” in *ICCV*, 2009.
- [73] O. Oreifej, G. Shu, T. Pace, and M. Shah, “A two-stage reconstruction approach for seeing through water,” in *CVPR*, 2011.
- [74] Z. Li, Z. Murez, D. Kriegman, R. Ramamoorthi, and M. Chandraker, “Learning to see through turbulent water,” in *WACV*, 2018.
- [75] J. G. James, P. Agrawal, and A. Rajwade, “Restoration of non-rigidly distorted underwater images using a combination of compressive sensing and local polynomial image representations,” in *ICCV*, 2019.
- [76] Y. Tian, S. G. Narasimhan, and A. J. Vannevel, “Depth from optical turbulence,” in *CVPR*, 2012.
- [77] M. Alterman, Y. Y. Schechner, and Y. Swirski, “Triangulation in random refractive distortions,” *IEEE transactions on pattern analysis and machine intelligence*, 2016.
- [78] M. Zhang, X. Lin, M. Gupta, J. Suo, and Q. Dai, “Recovering scene geometry under wavy fluid via distortion and defocus analysis,” in *ECCV*, 2014.
- [79] J. Iseringhausen, B. Goldlücke, N. Pesheva, S. Iliev, A. Wender, M. Fuchs, and M. B. Hullin, “4d imaging through spray-on optics,” *ACM Transactions on Graphics*, 2017.
- [80] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *CVPR*, 2016.
- [81] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” *Journal of Mathematical Imaging and Vision*, 2011.

- [82] T. Goldstein and S. Osher, “The split bregman method for l1-regularized problems,” *SIAM journal on imaging sciences*, 2009.
- [83] D. Geman and C. Yang, “Nonlinear image recovery with half-quadratic regularization,” *IEEE Transactions on Image Processing*, 1995.
- [84] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, 2011.
- [85] N. Parikh, S. Boyd *et al.*, “Proximal algorithms.” *Foundations and Trends in Optimization*, 2014.
- [86] L. Lourenco, A. Krothapalli, and C. Smith, “Particle image velocimetry,” in *Advances in Fluid Mechanics Measurements*, 1989.
- [87] K. Okamoto, S. Nishio, T. Saga, and T. Kobayashi, “Standard images for particle-image velocimetry,” *Measurement Science and Technology*, 2000.
- [88] T. Casey, J. Sakakibara, and S. Thoroddsen, “Scanning tomographic particle image velocimetry applied to a turbulent jet,” *Phys. Fluids*, 2013.
- [89] E. Meinhardt-Llopis, J. Pérez, and D. Kondermann, “Horn-schunck optical flow with a multi-scale strategy,” *Image Processing on line*, 2013.
- [90] E. Candes, M. Wakin, and S. Boyd, “Enhancing sparsity by reweighted l1 minimization,” *J. Fourier analysis and applications*, 2008.
- [91] F. Heide, Q. Fu, Y. Peng, and W. Heidrich, “Encoded diffractive optics for full-spectrum computational imaging,” *Sci. Rep.*, vol. 6, 2016.
- [92] T. W. Fahringer, K. P. Lynch, and B. S. Thurow, “Volumetric particle image velocimetry with a single plenoptic camera,” *Measurement Science and Technology*, 2015.
- [93] E. A. Deem, Y. Zhang, L. N. Cattafesta, T. W. Fahringer, and B. S. Thurow, “On the resolution of plenoptic PIV,” *Measurement Science and Technology*, 2016.
- [94] S. Shi, J. Wang, J. Ding, Z. Zhao, and T. New, “Parametric study on light field volumetric particle image velocimetry,” *Flow Meas. & Instr.*, 2016.
- [95] R. Zhang and J. Kwok, “Asynchronous distributed ADMM for consensus optimization,” in *ICML*, 2014.
- [96] B. Choudhury, R. Swanson, F. Heide, G. Wetzstein, and W. Heidrich, “Consensus Convolutional Sparse Coding,” in *CVPR*, 2017.

- [97] J. Xiong, A. A. Aguirre-Pablo, R. Idoughi, S. T. Thoroddsen, and W. Heidrich, “Rainbowpiv with improved depth resolution—design and comparative study with tomopiv,” *Measurement Science and Technology*, 2020.
- [98] F. Pereira and M. Gharib, “Defocusing digital particle image velocimetry and the three-dimensional characterization of two-phase flows,” *Measurement Science and Technology*, 2002.
- [99] S. Y. Yoon and K. C. Kim, “3d particle position and 3d velocity field measurement in a microvolume via the defocusing concept,” *Measurement Science and Technology*, 2006.
- [100] S. Shi, J. Ding, C. Atkinson, J. Soria, and T. New, “A detailed comparison of single-camera light-field piv and tomographic piv,” *Experiments in Fluids*, 2018.
- [101] B. E. Rice, J. A. McKenzie, S. J. Peltier, C. S. Combs, B. S. Thurow, C. J. Clifford, and K. Johnson, “Comparison of 4-camera tomographic piv and single-camera plenoptic piv,” in *2018 AIAA Aerospace Sciences Meeting*, 2018.
- [102] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, “Fast cost-volume filtering for visual correspondence and beyond,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.
- [103] J. Xiong and W. Heidrich, “In-the-wild single camera 3d reconstruction through moving water surfaces,” in *ICCV*, 2021.
- [104] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.
- [105] H. Hirschmuller and D. Scharstein, “Evaluation of cost functions for stereo matching,” in *CVPR*, 2007.
- [106] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume,” in *CVPR*, 2018.
- [107] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, 2004.
- [108] J. Delon and B. Rougé, “Small baseline stereovision,” *Journal of Mathematical Imaging and Vision*, 2007.
- [109] S. Cai, S. Zhou, C. Xu, and Q. Gao, “Dense motion estimation of particle images via a convolutional neural network,” *Experiments in Fluids*, 2019.

- [110] M. C. Morrell, K. Hickmann, and B. Wilson, “Particle image velocimetry analysis with simultaneous uncertainty quantification using bayesian neural networks,” *Measurement Science and Technology*, 2021.
- [111] C. Lagemann, K. Lagemann, S. Mukherjee, and W. Schröder, “Deep recurrent optical flow learning for particle image velocimetry data,” *Nature Machine Intelligence*, 2021.
- [112] V. Sitzmann, M. Zollhoefer, and G. Wetzstein, “Scene representation networks: Continuous 3d-structure-aware neural scene representations,” *Advances in Neural Information Processing Systems*, 2019.
- [113] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, “Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision,” in *CVPR*, 2020.
- [114] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” in *ECCV*, 2020.
- [115] J. Xiong, P. Richtarik, and W. Heidrich, “Stochastic convolutional sparse coding,” in *International Symposium on Vision, Modeling, and Visualization*, 2019.
- [116] J. Xiong, J. Wang, W. Heidrich, and S. Nayar, “Seeing in extra darkness using a deep-red flash,” in *CVPR*, 2021.

APPENDICES

A Published Papers

This dissertation is based on the following published works.

- [65] Jinhui Xiong, Ramzi Idoughi, Andres A. Aguirre-Pablo, Abdulrahman B. Aljedaani, Xiong Dun, Qiang Fu, Sigurdur T. Thoroddsen, and Wolfgang Heidrich. “Rainbow particle imaging velocimetry for dense 3D fluid velocity imaging.” ACM Transactions on Graphics (TOG) 36, no. 4 (2017): 1-14. Referring to Chapter [3].
- [66] Jinhui Xiong, Qiang Fu, Ramzi Idoughi, and Wolfgang Heidrich. “Reconfigurable rainbow PIV for 3D flow measurement.” In 2018 IEEE International Conference on Computational Photography (ICCP), pp. 1-9. IEEE, 2018. Referring to Chapter [4].
- [97] Jinhui Xiong, Andres Alejandro Aguirre-Pablo, Ramzi Idoughi, Sigurdur T. Thoroddsen, and Wolfgang Heidrich. “RainbowPIV with improved depth resolution—design and comparative study with TomoPIV.” Measurement Science and Technology, no. 2 (2020): 025401. Referring to Chapter [5].
- [103] Jinhui Xiong, Wolfgang Heidrich. “In-the-Wild Single Camera 3D Reconstruction Through Moving Water Surfaces.” ICCV (Oral), 2021. Referring to Chapter [6].

B Published Papers Not included in This Dissertation

The following publications focus on other projects, which are not directly related to the dissertation.

- [40] Andres A. Aguirre-Pablo, Abdulrahman B. Aljedaani, Jinhui Xiong, Ramzi Idoughi, Wolfgang Heidrich, and Sigurdur T. Thoroddsen. "Single-camera 3D PTV using particle intensities and structured light." *Experiments in Fluids* 60, no. 2 (2019): 25.
- [115] Jinhui Xiong, Peter Richtarik and Wolfgang Heidrich. 'Stochastic convolutional sparse coding.' International Symposium on Vision, Modeling and Visualization (Best Paper Award), 2019
- [116] Jinhui Xiong, Jian Wang, Wolfgang Heidrich and Shree Nayar. "Seeing in Extra Darkness Using a Deep-Red Flash." CVPR (Oral), 2021