On the development of a Neural Radiance Field technique for tomographic reconstruction applied to flow diagnostics

by

Dustin Kelly

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 4, 2024

Keywords: Tomography, Neural Radiance Fields, Scalar Field Reconstruction, Volumetric Imaging, Flow Diagnostics

Approved by

Dr. Brian Thurow, Chair, W. Allen and Martha Reed Professor of Aerospace Engineering
Dr. Nek Sharan, Assistant Professor of Aerospace Engineering
Dr. David Scarborough, Associate Professor of Aerospace Engineering
Dr. Davide Guzzetti, Assistant Professor of Aerospace Engineering
Dr. Vrishank Raghav, Associate Professor of Aerospace Engineering
Dr. Chris Clifford, Research Mechanical Engineer at Air Force Research Laboratory

Abstract

In this work, a novel neural implicit representation tomography algorithm based on Neural Radiance Fields is developed and demonstrated for 3D flow diagnostics. Neural Radiance Fields (NeRF) originate from the computer vision community that uses a machine learning approach to approximate a scene of interest as a continuous function using a neural network. The NeRF machine learning concept provides some advantages over traditional tomography methods, including i) a continuous approximation of the volume that removes the inherent limitation on volume resolution that is present in discretized representations, ii) a reduction in memory requirements for the volume prediction, and iii) an adaptable tomography framework that can include additional inputs, outputs, imaging models, and constraints. The method developed in this work, FluidNeRF, predicts the intensity per unit volume as a continuous function of 3D spatial (static) or 4D spatial-temporal (time-resolved) coordinates. FluidNeRF trains similarly to other algebraic reconstruction techniques, where the volume approximation is updated by comparing predicted and captured images of the volume. The image rendering technique of FluidNeRF employs an emission-based imaging model. Static and time-resolved FluidNeRF was evaluated using both i) a DNS-generated turbulent mixing jet and ii) an experimental dataset of a low-speed, smoke-entrained jet flow. The synthetic datasets systematically investigated the hyperparameters, camera configuration, and image noise regarding reconstruction quality. Static FluidNeRF was also compared to a traditional ART-based tomography model. The results show that i) FluidNeRF is a viable technique for tomography of flow diagnostics, ii) FluidNeRF produces comparable or superior reconstruction accuracy and is more robust to noise than traditional tomography methods, and iii) the method can scale to larger problems. Additionally, the results proved the FluidNeRF can be expanded to time-resolved reconstructions, which further compresses the volume representation and implicitly constrains the problem in time.

Acknowledgments

During my graduate school studies at Auburn University, I worked with several outstanding individuals who allowed me to grow as a researcher and, more importantly, as a person. The doctoral journey was filled with many long nights and weekends; however, I also had many fun and exciting experiences. This dissertation would not be possible without the encouragement, support, and technical advice from influential people in my life, and this one goes out to all of you.

First, I would like to thank my PhD advisor, Dr. Brian Thurow, for allowing me to work in the aerodynamic measurement technique community. Coming from an undergraduate degree in Aerospace Engineering, I would not have predicted that I would be studying optics and metrology. Thank you for supporting and allowing me to pursue ideas for technical problems that might be slightly outside our knowledge, not knowing that these ideas could grow into this dissertation. Thank you for being my academic and professional advisors as I sought new opportunities. You have provided unique experiences that allowed me to explore the industry, including international conferences, collaborations, and internships. Thank you for taking the time to grow me as a technical writer, even though it required a lot of effort. I will cherish the time I was provided with you as my professor.

To Dr. Scarborough, thank you for being my unofficial second advisor. All the classes you have taught throughout my academic career have contributed to my technical knowledge. You opened my eyes to the aerodynamic and combustion research community during my undergraduate career, which led me to my position under Dr. Thurow. Also, thank you for taking the time to help me and others through the qualifying exams. To Dr. Sharan, thank you for the knowledge you provided via our conversations and for taking the time to generate the synthetic data for this dissertation. I wish I had the time to take a few classes from you due to your extensive knowledge of CFD. In addition, I would like to thank Dr. Guzzetti and Dr. Raghav for being on my advising committee and providing critical feedback during my candidacy.

To Dr. Chris Clifford, I want to start by saying thank you for being a terrific post-doc. You took the time to discuss and educate me on everything from plenoptic cameras to algorithm development, contributing to my research knowledge. Not only were you helpful during your time at Auburn, but you also provided an opportunity to have many great experiences with an internship at AFRL VKF Tunnel D. Those experiences helped me see the supersonic research community outside of plenoptic imaging, which has led to my current career endeavors. With the internship, I am grateful to Dr. Brian Rice for being my AFRL mentor at Tunnel D and allowing me to work on several fascinating projects.

To my office mates Mahi Moaven and Bibek Sapkota, thank you for never letting there be a dull moment while at the office or lab. Both of you were always there to help move equipment or open a door if needed. Mahi, thank you for the never-ending discussions of Auburn sports. We can confirm that you are the undisputed champion of darts. Bibek, thank you for the world saving discussions on world politics. I will always remember the videos you would leave in the background, even if you left the room. Lastly, I want to thank both of you for helping me move from Auburn. I could not have done this without the help of both of you.

To Giffin Eagan, thank you for the friendship and encouragement beginning with our undergraduate studies. I wouldn't have been able to make it through all of our classes without your help and encouragement. I also appreciate our time outside class, even though we managed to get in trouble a time or two. Also, thank you for the countless days during COVID, where we would work through math and aerodynamic problems in preparation for the qualifying exam. I wouldn't be who I am today without your friendship, so I thank you.

To the rest of the past and present AFDL members, I am thankful for every last one of you. All the interactions and help I have been granted through my years at school have led me to this point, and I am grateful for all the support and friendships.

To my dad and mom, David and Tami Kelly, there are no words to express my gratitude for the constant support and motivation. Without both of you providing for me over the years, I would not have been able to do any of this. To my brothers, thank you for always being there to play a game or other extracurricular activities to help me get through this challenging work. To the rest of the Kelly and Faulkner families, I wish some of you were here to see me complete this work I started 6 years ago. However, all of you have contributed to the person I have become and are the reason I have been able to come this far.

Last but not least, I want to thank my wife, Maddie Kelly, for constantly pushing me and believing that I can achieve a Ph.D. You were understanding and accepting of my late nights and early mornings. I am so grateful to have you by my side through this chapter of our lives, and I cannot wait for our future endeavors in life. Thank you for connecting me to your family, who have also been a significant advocate for my further education. I could not have asked for a better wife. I love you.

- *I believe in Auburn and love it.*

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Three-dimensional (3D) flow diagnostics have received increasing attention and interest due to the spatial complexity in many real-world applications. 3D tomography is a technique that reconstructs a 3D scene from a series of projections acquired by a distribution of 2D sensors surrounding a flow field. Image-based 3D flow diagnostic techniques that use tomography include fluorescence (LIF) [21], optical pyrometry [22], tomographic absorption spectroscopy (TAS) [23], particle image velocimetry (PIV) [24, 25], tomographic emission spectroscopy [26, 27], background oriented Schlieren (BOS) [28], and many more. The tomographic reconstruction method is generally an ill-posed and under-defined inverse problem, making it difficult to solve. In turn, larger or higher resolution volumes are required to experimentally capture the extensive range of space and time scales that can occur in complex flows, making the problem more challenging to solve due to the under-defined problem. Therefore, improvements to the tomographic approach are still needed.

Several reconstruction techniques have been developed over the past decades to solve the tomography problem including filtered back projection (FBP), deconvolution [29], iterative reconstruction (IR) techniques including algebraic reconstruction technique (ART) [30, 31, 32, 33, 34, 25, 35], and genetic algorithms (GA) [36, 37, 38]. FBP was the original standard for computed tomography (CT) in the medical field. However, it requires a large number of evenly spaced images that surround the object of interest. Iterative reconstruction (IR) techniques have been implemented for computed tomography (CT), where many variations of the method have emerged [39]. ART and its variants have been implemented for many flow diagnostic techniques such as PIV, BOS, and LIF due to the ability to reconstruct with a limited range of views [25, 40, 41]. Chetih and Messali [42] showed the advantages of ART over FBP for medical imaging. Grauer *et. al.* [1] has presented an extensive review article on volumetric emission tomography methods for combustion measurements. ART techniques require the volume to be discretized, which can lead to numerical artifacts, aliasing, or numerical instability during reconstruction. In addition, voxel-based approaches like ART require large memory (1+ GB) and computational costs that scale with the number of voxels used to represent the volume. Ultimately, this limits the ability to scale voxel-based reconstruction methods to very high resolutions, which could be achieved with modern improvements to imaging systems.

Most recently, machine learning techniques have rapidly expanded into tomographic reconstruction techniques, including flow diagnostics. Willemink and Noël [39] explain the recent advances in machine learning for tomographic reconstruction, where most work has focused on enhancing reconstruction quality in the medical field. For flow diagnostics, a machine learning technique was developed to reconstruct particle size and location via a U-net convolutional neural network architecture for digital holography [43, 44]. The U-net architecture increased extraction rates of particles at four times the concentrations and higher accuracy compared to traditional methods. Another technique reconstructs temperature and species concentrations in a flame using a neural network approach described by Ren *et al* [4]. Additionally, tomographic reconstruction has been attempted using convolutional neural networks [3, 45]; however, CNN methods require volume discretization and is limited to relatively small volumes. Unfortunately, these machine learning techniques require supervision/training data typically synthetically generated from CFD models. This approach makes it challenging to apply these techniques to new situations where training data is unavailable or the method has reduced generality. Determining the uncertainty when using these techniques for a new problem is problematic due to being trained with different datasets.

The computer vision community introduces a machine learning framework that approximates a volume as a *continuous spatial and temporal function*. The inspiration for this work originates from the neural radiance fields (NeRF) [6] method. NeRF represents a volume as a continuous function of 3D spatial coordinates and 2D view angles. The continuous function is generated by a feed-forward neural network that trains on the difference between the imaged perspectives and the rendered images of the approximated volume, similar to the traditional tomography methods. NeRF removes the inherent spatial resolution limitation of the discretized methods, compresses memory required for storage by orders of magnitude, and provides an easily adaptable framework for incorporating ray models, image rendering techniques, adjusting inputs, and physics-based learning models. Gao *et al.* [46] covers the advances of the NeRF technique with over 200 references in three years since the introduction of NeRF. A NeRF framework has been utilized for 3D tomography of flow fields for an instantaneous set of images [8, 47, 7]. Several techniques incorporate temporal information into the NeRF model [48, 49], but each possesses different training methods and network structures for incorporating time.

Tangential to the computer vision community, Physics-informed neural networks (PINNs) have been introduced to model complex flow fields [50], where feed-forward neural networks relate relevant flow parameters in space and time. Using automatic differentiation, PINNs can provide partial derivatives of the output variables (u, v, w, ρ). These partial derivatives are crucial for enforcing physics-based equations (e.g., Navier-Stokes, advection-diffusion, etc.) as loss functions for updating the model. PINNs have been shown to effectively solve inverse problems in fluid and solid mechanics [51, 52, 53, 54]. Raissi *et al.* [52] demonstrates that PINNs can calculate flow field measurements using flow visualization techniques. However, the measurements were limited to 2D (images), or the 3D scalar field output of the PINN was trained using volumetric intensities generated from traditional tomography methods. Molnar and Grauer [9] demonstrated that PINNs can be utilized for flow tomography of 2D flow fields by adding a "data loss" similarly used by NeRF methods. Furthermore, PINNs have also been proven for reconstructing 3D velocity and pressure fields using particle tracks [55] and BOS measurements [56]. Flow field measurements using a NeRF framework have been demonstrated in concurrent work by Chu *et al.* [7], where a velocity field was estimated from smoke visualization. PINNs and NeRF are two active areas of research where the developments are mainly application-agnostic. Therefore, deep learning-based tomography stands to benefit from the rapid advances occurring in both fields.

The previous paragraph illustrated some potential advantages that the NeRF technique brought to the computer vision community; however, NeRF-based techniques have been mainly applied to opaque natural scenes with limited investigation of flow tomography. To the best of the author's knowledge, there has yet to be a systematic study of the NeRF technique for 3D flow diagnostics, including hyperparameters of the network, camera configuration, and the introduction of time into the model. This work seeks to demonstrate a new machine learning architecture for flow tomography that can lead to a data assimilation model and characterize the framework to understand the method's limitations. This study performs a preliminary investigation to establish NeRF as a viable reconstruction technique for 3D flow diagnostics, termed FluidNeRF. The objectives of this work are as follows:

1. Design a viable NeRF-based 3D flow field reconstruction technique with consideration given to both the configuration of the multi-camera imaging system and the corresponding selection of NeRF hyperparameters.

2. Characterize the performance of NeRF for 3D flow field reconstruction by benchmarking it against an ART-based method for scalar field reconstructions.

3. Demonstrate the viability of incorporating temporal information into the reconstruction methodology that will be called time-resolved FluidNeRF (TR-FluidNeRF) using both synthetic and experimental volumes.

The objectives are achieved using synthetically generating images of a CFD-generated turbulent free jet volume and an experimental smoke-filled jet flow representative of typical measurements acquired for flow diagnostics. The synthetic CFD dataset provides a ground truth with several regions of varying complexities to evaluate the limits of the proposed reconstruction technique. The turbulent jet also resembles typical flow diagnostic volumes. An in-house image rendering code generates the perspective images to provide flexibility on camera parameters, layout, and noise levels. An experiment was designed to evaluate and demonstrate the reconstruction for real-world datasets, with the experiment using smoke as a passive scalar in the flow.

This dissertation focuses on supplying an initial understanding of a neural implicit representation method for flow tomography, including the influential hyperparameters on the reconstruction quality, computational efficiency, and limitations compared to a traditional tomography method. The first chapters will provide the necessary background information for the reader to understand the developed tomography method. Chapter 2 presents the conventional tomography techniques developed over the past decades, with most being applied to flow diagnostics. Chapter 3 introduces the fundamental concepts of the machine learning techniques that are utilized by this work. The chapter also illustrates the past developments for 3D tomography in computer vision and flow diagnostic communities. Chapter 4 finishes the background information by discussing typical scalar field measurements utilized for combustion and flow diagnostics. In this chapter, we emphasize the tomography algorithms and optical systems that have been applied to past work to show the current state of scalar field measurements. Chapter 5 introduces our formulation of the FluidNeRF technique, with a detailed description of each component of the NeRF-based architecture. Chapter 6 discusses the synthetic CFD dataset, the image rendering technique, and the ART-based technique employed to evaluate the FluidNeRF method. Chapter 7 presents the systematic results from the synthetic test cases, emphasizing the evaluation of the reconstruction quality for changing hyperparameters, camera layout, noise, and adding time to the model. Chapter 8 discusses the design, setup, and procedures of the smoke-filled jet experiment. During this work, a miniature camera system was developed to help reduce the overall costs and size of tomography systems, which this chapter also introduces. Chapter 9 presents the reconstructions and analysis of the smoke-entrained jet flow. Finally, Chapter 10 finishes the dissertation with this work's general conclusions while providing insight into the future work required for improving the method.

Chapter 2

Classical Tomography

Early developments of volumetric imaging for fluid diagnostics include planar or multi-planar methods that capture 3D information by stitching the planes together. These techniques consist of capturing various plane locations across different experimental runs, or the more popular method that comprises scanning the plane through the volume during the experiment. Combining the slices generates the 3D view. One of the first time-resolved 3D measurements of a gas jet was conducted using a scanning mirror to relocate the laser plane [57]. Since then, several 3D scanning methods have been conducted using PLIF and other measurements [58]. One advantage of scanning methods is the high spatial resolutions that can be acquired for a single slice of the volume. However, scanning techniques have lower resolution in the scanning direction as the number of planes is a trade-off between depth spatial resolution, temporal resolution, and number of planes. State-of-the-art MHz-rate cameras and lasers still lack the ability to capture the smallest structures and temporal resolution for turbulent flows. Additionally, the laser sheet has a finite thickness, and stitching slices together can create artifacts [1].

An alternative volumetric imaging technique is tomography. Tomography uses simultaneous line-of-sight (LoS) integrated projections of a volume to determine the original source of the projections as demonstrated in Figure 2.1. The LoS measurements have essential features of the volume convolved into a single image, where features could overlap one another. The convolution of features makes it difficult to separate these features. Therefore, multiple LoS measurements are needed to solve the problem. For flow diagnostics, multiple imaging sensors are distributed in a wide arc to capture 2D LoS projections of the volume. An alternative method is to rotate a single camera around the volume. To use the single camera method, the volume must be relatively static during the imaging process due to the temporal aspect of the capturing process. The sensors capture LoS projections of the volume, where each pixel is a single projection. These projections are inputs into a tomographic method, also called a reconstruction algorithm, to infer a source field. The source field could be subsets of the light field or features that alter the light-field, including absorption and density gradients. The term *light field* defines the dense array of light rays that fills space [59], where light rays are vectors that vary in space and time, with corresponding intensities that are functions of wavelength. The volume measurements can then be used to deduce additional quantities like concentrations and temperature. Reconstruction techniques require a camera model to approximate the imaging process. Ultimately, tomography attempts to deduce the inverse imaging model to determine the source field.

The process of determining a 3D source field from lower-order projections is a challenging problem. Tomography solves an inverse problem, where the only way to solve the source light-field exactly is with an infinite number of projections around the volume. The ambiguity of the under-determined inverse problem causes an endless number of reconstructions that produce the input projections. Additionally, the problem is typically ill-posed due to noise, model errors, and other uncertainties with the volume projections. Increasing the number of projections better constrains the inverse problem; however, the current cost of high-resolution imaging devices and the complexity of the experimental setup of several large cameras can limit the number of practical projections. Even with these challenges, diagnostics that have employed tomography have shown impressive results. The following sections provide an overview of the various methods developed to solve the inverse problem, where the methods can be broken into two categories: analytical and algebraic [1]. The rest of this chapter provides the reader with a general background on the classical tomography methods to provide an understanding on the development our new tomography method of this work. There have been several review articles for tomography [39, 60, 34, 1], where the author follows the definitions and terminology that was formulated in the review article by Grauer *et. al.* [1].

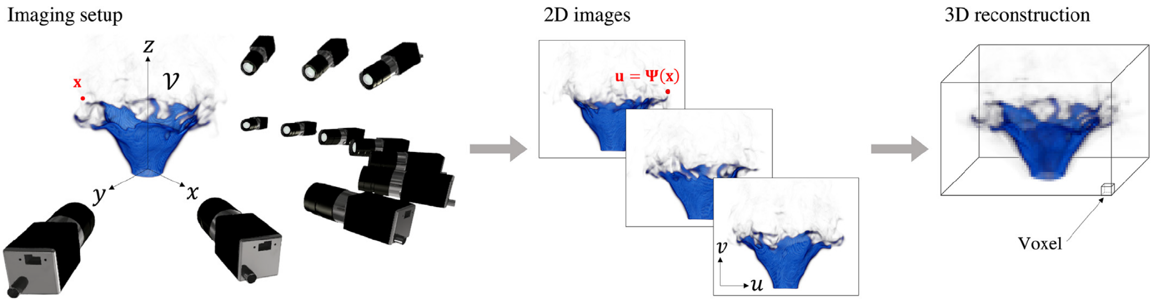


Figure 2.1: Schematic of the tomography process with cameras positioned around a volume of interest to capture 2D LoS projections of a volume of interest. The LoS projections are used to generate a predicted volume through a tomography algorithm. This figure is taken from the review article by Grauer *et. al.* [1].

2.1 Modeling 2D imaging

Let us start with understanding the 2D imaging process of volumetric imaging, as tomography must conduct the inverse to solve for the 3D scene/volume. Volumetric imaging acquires knowledge of the quantity of interest (QoI) in a 3D field from 2D images (perspectives), which is typically conducted with cameras. Images are captured simultaneously, averaged, or phase-averaged using one or more cameras or sensors. Taking the inverse of the imaging model to deduce a 3D field is called the reconstruction. Therefore, knowing the imaging model perfectly is essential to produce the best reconstructions. The rest of this section covers imaging models that have been developed.

The volumetric QoI, denoted as $g(\mathbf{x})$, represents the quantity of interest for 3D spatial locations within the volume of interest, where $\mathbf{x}$ is the 3D spatial coordinates. The images captured of the QoI are discrete measurements of intensity, with 2D discrete coordinates of pixels. The 2D unit vector ($\mathbf{u} = [u, v]$) of the sensor is an integer value that corresponds to the centroid of the pixels. A projection function (Ψ) is the relationship between $g(\mathbf{x})$ and $\mathbf{u}$ through

$$\mathbf{u} = \Psi(g(\mathbf{x})) \tag{2.1}$$

The projection function accounts for the optical system that captures light rays from the field of view, composed of lenses, mirrors, prisms, and filters. In addition to the optical system, the projection function resolves refraction and other phenomenon that changes the direction of light from the volume. For an ideal system, distortions and aberrations within the optical system are ignored when using a pinhole camera model. A pinhole camera model replaces the optics system with a single infinitesimal aperture that the sensor imagines, simplifying the imaging system. In a real optical system, decreasing the aperture size would increase diffraction effects. However, for most flow diagnostics, finite lenses, and apertures are employed to capture adequate accumulation of light on the sensor for short integration times. With a finite aperture, distortions should not be ignored. Therefore, assuming perfect thin lenses can lead to uncertainty about the location of light on the sensor. Grauer *et al.* [1] covers several imaging models that can account for these effects.

The projection function governs the location a ray will intersect with the image plane from a point in space. However, originating from a pixel going in the inverse direction corresponds to a continuous set of world coordinates along the LoS of that pixel. The path of the LoS from a pixel can be determined through the back-projection denoted as,

$$\Psi^{-1}(\mathbf{u}, l) = \mathbf{x} \tag{2.2}$$

where l is the distance the ray travels relative to the aperture. For a perfect system, the back-projection function can be determined by taking the inverse of the projection function. However, a real optical system with a finite aperture will produce a point spread function on the sensor. The point spread function relates a point source in space to the image shape of the point source as a function of the distance of the source relative to the focal plane of the optical system. The point spread function indicates that there is not a unique solution for the inverse of Ψ. Therefore, the central ray that passes through the center of the aperture is taken as the primary ray. Real optical systems' projection and back-projection functions are calculated via camera calibration methods [1].

The radiative transfer equation (RTE) describes the production, transmission, and attenuation for a single wavelength of light (λ) as described by Modest [61]. The RTE can be simplified by assuming a dark background, negligible absorption, and the light is not scattered by other molecules or objects in the volume. Using these simplifications, the spectral intensity that reaches the sensor is calculated via

$$I_\lambda(\mathbf{u}) \propto \int_0^\infty I'_\lambda[\Psi^{-1}(\mathbf{u}, l)]dl, \tag{2.3}$$

where I'_λ is the volumetric, monochromatic intensity source term. The field of interest can be found by spectrally integrating I'_λ for the QoI. Eqn. 2.3 becomes increasingly more complex when the optically thin assumption breaks down, which means absorption and out-scattering are not negligible and must be included in the equation. Absorption and out-scattering are known as extinction.

Using the assumption from Eqn. 2.3 and the field of interest is contained within the depth of field of the camera, the signal captured at each pixel (S_i) can be found using

$$S_i = \iint_{\mathcal{X}_i} \int_0^\infty \eta_\lambda \tau_{\lambda,f} \int_0^\infty I'_\lambda[\Psi^{-1}(\mathbf{u}, l)] \frac{\Omega_i(l)}{4\pi} .\mathrm{d}l\mathrm{d}\lambda\mathrm{d}u\mathrm{d}v \tag{2.4}$$

Not only is S_i a function of the transmittance of light, but it is also a function of the sensor's response to light. $\mathcal{X}_i$ is the sensitive area of a pixel for the i^{th} pixel. η_λ is the quantum efficiency of the sensor that is a function of wavelength and can vary with $\mathbf{u}$. The transmittance of light ($\tau_{\lambda,f}$) is the net amount of light that passes through all the optics and filters. Finally, Ω_i quantifies the amount of solid angular space subtended by a region as seen from a distance, l. The solid angle for volumetric imaging is the aperture area over l^2, which can be approximated using the f-number of the optical system $\left(\Omega \simeq f_\#^{-2}\right)$.

One simple model that has been used widely for emission-based measurements is the emission-only rendering model, which assumes that i) absorption, scattering, and refraction are negligible inside the field of interest, ii) the only sources of light originate from the QoI, and iii) the field of interest is within the depth-of-field of the camera. The emission-only model is provided as

$$p_i = \int_0^\infty \eta_\lambda \tau_{f,\lambda} \mathrm{d}\lambda \frac{\Omega}{4\pi} \int_{l_{in}}^{l_{out}} g\left[\Psi^{-1}(\mathbf{u}_i, l)\right] \mathrm{d}l, \tag{2.5}$$

where p_i the projection along the LoS of g at the i^{th} pixel, and l_{in} and l_{out} are the ray lengths originating from the lens to the entrance and exit of the volume, respectively. The integral begins at the aperture ($l = 0$) and proceeds to the beginning of the volume ($l = l_{in}$) before passing through the volume at $l = l_{out}$. Anything outside the volume g is assumed to be zero; therefore, the light generated is contained between l_{in} and l_{out}. The emission-only model is one of the simplest rendering techniques. A more realistic imaging model can be employed to account for finite, real optics,

$$p_i = \int_0^{\infty} \eta_\lambda \tau_{f,\lambda} \mathrm{d}\lambda \iint_{\mathcal{X}_i} \int_{l_{in}}^{l_{out}} g\left[\Psi^{-1}(\mathbf{u}_i, l)\right] \frac{\Omega_i(l)}{4\pi} \mathrm{d}l \mathrm{d}\mathbf{u}. \tag{2.6}$$

Eqn. 2.6 brings the solid angle inside the inner integral since the solid angle changes with the distance to the camera. The following sections cover methods that attempt to invert the measurements to determine g (reconstruct the volume).

2.2 Analytical methods

Analytical models provide an explicit function to solve the tomography problem, where the equation relates projections ($\mathbf{p}$) to the reconstruction (g). Two of the most common analytical models are the Fourier- and Radon-transform-based methods. These methods assume that the Fourier transform of the projection data corresponds to 1D lines in the Fourier space of the 2D or 3D field g. The Fourier slice theorem techniques are covered in the next section. In addition to this transform-based algorithm, the inverse Abel transform provides an analytical method for reconstruction. Abel's transform assumes a radially symmetric source field to derive an explicit formula for the inversion process.

2.2.1 Transform-based algorithms

Transform-based algorithms are derived from the Fourier Slice Theorem as explained by Kak *et. al.* [62]. Figure 2.2 shows the main components of the transform using a simple 2D example of the Fourier-based algorithm using a parallel projection of the volume ($g(x, y)$). The filtered back projection (FBP) algorithm is the most common closed-form solution to the inversion problem in tomography for real measurements. The Fourier slice theorem relates the Fourier transform of the projections to the Fourier transform of the volume. The algorithm was first

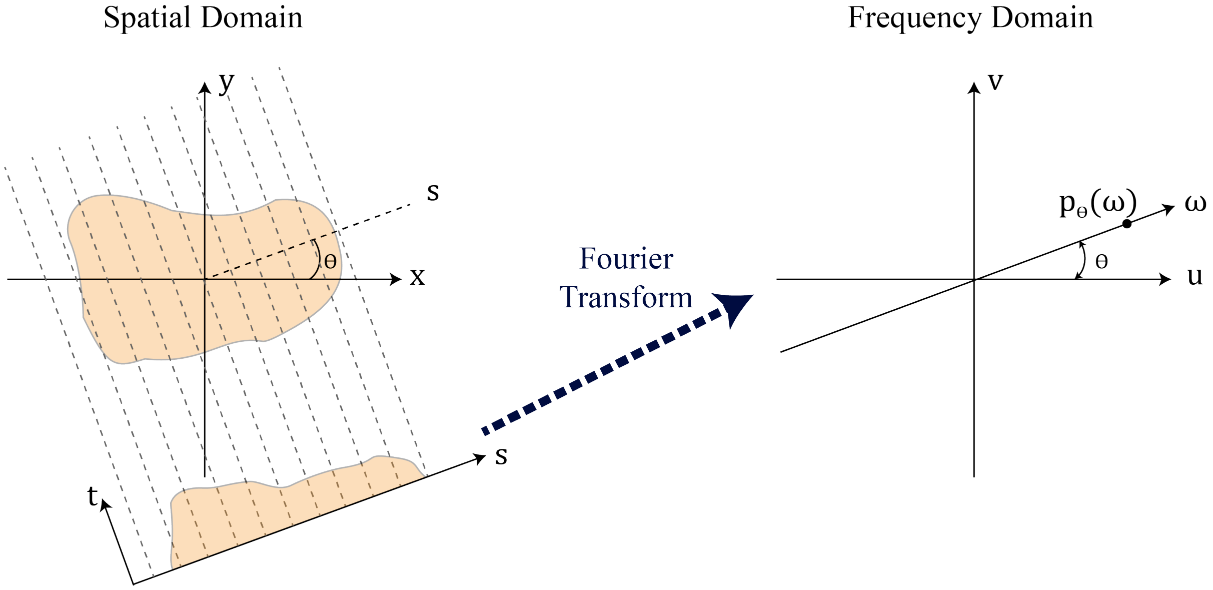


Figure 2.2: 2D schematic of the Fourier Slice theorem. The 1D projection of the volume using parallel LoS measurement through the QoI and transformed to the frequency domain using a 1D Fourier transform, also known as the Radon transform.

demonstrated for 2D tomography from 1D projections; however, the method has also been extended to 3D tomography with 2D projections.

To start the Fourier Slice Theorem, let's consider the 2D Fourier transform of the 2D object function (Figure 2.2) $g(x, y)$,

$$G(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-j2\pi(ux+vy)} dx dy \tag{2.7}$$

$$u = \omega \cos\theta, v = \omega \sin\theta \tag{2.8}$$

where j is the square root of -1 and u and v are the x- and y-direction wavenumbers of G. The wavenumbers can be calculated using Eqn. 2.8. The 1D Fourier transform of the projection ($P_\theta(s)$) of $g(x, y)$ use sinogram coordinates (s, θ), where θ is the angular position of the ray. The cartesian coordinates (x, y) are related to the projection coordinates using the following transformation,

$$s = x \cos\theta + y \sin\theta, \tag{2.9}$$

$$l = -x \sin\theta + y \cos\theta. \tag{2.10}$$

The projection of g using sinogram coordinates is found by integrating along the LoS of the rays in the l-direction as shown in Eqn. 2.11,

$$P_\theta(s) = \int_{-\infty}^{\infty} g(s, l) dl \tag{2.11}$$

which is referred to as the Radon transform [63]. This formulation assumes infinitesimally thin and parallel rays. The 1D Fourier transform of the projection function $P_\theta(s)$ assuming continuous data along s is

$$S_\theta(\omega) = \int_{-\infty}^{\infty} P_\theta(s) e^{(-j2\pi\omega s)} ds. \tag{2.12}$$

The projection $P_\theta(s)$ can be replaced with the Radon transform (Eqn. 2.11),

$$S_\theta(\omega) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} g(s, l) dl \right] e^{(-j2\pi\omega s)} ds. \tag{2.13}$$

Converting the sinogram coordinates to cartesian coordinates,

$$S_\theta(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-j2\pi\omega(x\cos\theta + y\sin\theta)} dx dy. \tag{2.14}$$

The 2D Fourier transform of g is on the right-hand side, where Eqn. 2.8 relates the cartesian coordinates and θ to the wavenumbers. The result indicates that the 1D Fourier transform of the projections equals the 2D Fourier transform of the object function $G(u, v)$. Assuming an infinite number of evenly-spaced projections were captured, then $G(u, v)$ would be known exactly. This formulation allows the original object $g(x, y)$ to be determined through the inverse Fourier transform of G, Eqn. 2.15, thus proving the Fourier slice theorem.

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(u, v) e^{j2\pi(ux+vy)} du dv \tag{2.15}$$

The equation can be converted to polar coordinates using Eqn. 2.8, which yields

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S_\theta(\omega) |\omega| e^{j2\pi\omega(x\cos\theta + y\sin\theta)} d\omega d\theta \tag{2.16}$$

While the Fourier slice theorem provides a way to analytically solve for an object from its projections, in practice, only a finite number of projections can be recorded. A finite number of projections leads to only resolving a limited number of radial lines (θ) in the frequency domain, as shown in Figure 2.2. Additionally, there are only a discrete number of samples in the frequency domain along the radial line corresponding to the pixel size. Since the radial lines will have finite angular separation, sampling points become increasingly sparse with frequency. Thus, the Fourier slice theorem poorly resolves the high-frequency content in a volume, which leads to increased error in the high-frequency components of g during back-projection. A filter can be applied to the back-projection equation, replacing $|\omega|$, to reduce the effects of the high-frequency error. This technique is referred to as "filtered back-projection" (FBP).

A wide range of filters have been developed for FBP, where the most common filter is the *Shepp-Logan filter* [64]. FBP was the first well-accepted method for CT scanners in the medical community. Even with filtering out the high-frequency, FBP requires a large number of evenly spaced projections (100+); otherwise, the reconstruction accuracy reduces significantly. FBP can quicly reconstruct volumes with high reconstruction quality but only when there are an adequate number of projections. Due to these requirements, FBP is only viable for limited flow applications due to the size and cost of modern high-speed cameras [1]. A more in-depth derivation and overview of FBP algorithms is found in the third chapter of the book by Kak and Slaney [62].

2.2.2 Abel Transform

The Abel transform technique assumes that the QoI in the imaged volume is radially symmetric, which forces a constraint on the reconstruction. In flow diagnostics, several steady flows, including jets, shocks, and laminar flames, can be assumed to be axisymmetric. The symmetry assumption reduces the complexity of the problem because axisymmetric objects/volumes will produce the same projection independent of θ, provided the projection is captured perpendicular to the axis of symmetry. A 2D example of this is demonstrated in Figure 2.3 [65]. The LoS projection of g is a function of the radial distribution given by,

$$p(y,z) = 2\int_{y}^{\infty} g(r,z)\frac{r}{\sqrt{r^2-y^2}}dr \tag{2.17}$$

where r is the distance to the central axis. The inversion of the projection equation to solve for g is termed Abel Transform (Eqn. 2.18)

$$g(r,z) = -\frac{1}{\pi}\int_{y}^{\infty} \frac{p'(y,z)}{\sqrt{y^2-r^2}}dy, \tag{2.18}$$

where $p' = \partial p/\partial y$. As explained by Dasch [65], the Abel transform is less common due to (i) the need for p' and (ii) the divergence of the solution at r. The finite difference approximation of p' using high-resolution images is sensitive to noise due to the solution requiring the derivative of the projection. Lower-order approximations can be made, but this decreases the resolution of g. Grauer *et. al.* [1] covers several numerical methods that were developed to stabilize the Abel transform, where the most popular was devised by Dasch [65] that uses a three-point Abel inversion technique. The disadvantage of the Abel transform is the limiting assumption that the projections are captured along thin, parallel rays, which is inconsistent with real optics. Thus leading to noticeable reconstruction errors. Abel's theorem should only be employed if the imaging model from Eqn. 2.17 is an accurate approximation and the volume can be assumed axisymmetric.

2.3 Algebraic methods

Alternative to analytical methods, algebraic techniques set up a series of equations relating the unknown volume to the projections. Algebraic techniques solve for the volume using a finite set of algebraic equations. Thus, the projections and volume must have a finite number of elements to form this finite set of equations, where the volume has to be discretized into n elements. The most popular method for volume discretization is partitioning the volume into evenly spaced and sized smaller volume elements. These are generally referred to as *voxels*. Voxels have a constant value throughout and do not affect the volume outside its area, similar to pixels of an image [1] given the name. The unknown volume/field g is represented by a finite set of equations called basis functions, φ_j. If voxels are used, the basis functions take the same

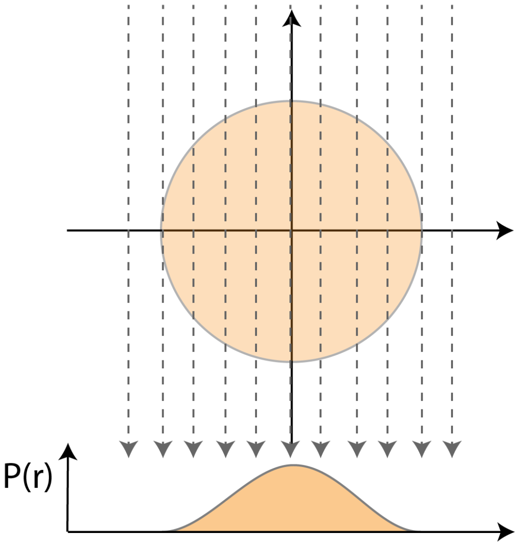


Figure 2.3: Demonstration of the forward Abel transform of a 2D axisymmetric object (circle) to form the projection using thin, parallel rays.

piece-wise shape for each, unity inside the voxel and zero outside. g can be approximated using n coefficients, g_j, multiplied by each basis function as provided by

$$g(\mathbf{x}) = \sum_{j}^{n} g_j \varphi_j(\mathbf{x}) \tag{2.19}$$

Similar to the back-projection methods described in the previous section, a more general model can be used to relate the unknown field g to the projections by using Eqn. 2.5. The projection function (Eqn. 2.5) can be discretized following the idea of voxels to produce a linear system of equations called ray-sums

$$\mathbf{A}g = p, \tag{2.20}$$

where $\mathbf{A}$ is a weight matrix that relates voxel coefficients g_j to the projection p_i via a matrix-vector multiplication. The projection and volume coefficients are arranged in column vectors with lengths m and n, respectively. $\mathbf{A}$ is a sensitivity matrix with a size of $m \times n$, where each value is a weight of the ith LoS measurement to the jth voxel coefficient. A more physical description is that $\mathbf{A}$ is an approximate imaging model to form p from g. The inversion of $\mathbf{A}$

will provide a solution for g. The next section covers algebraic methods to solve the inversion problem.

2.3.1 Algebraic reconstruction technique

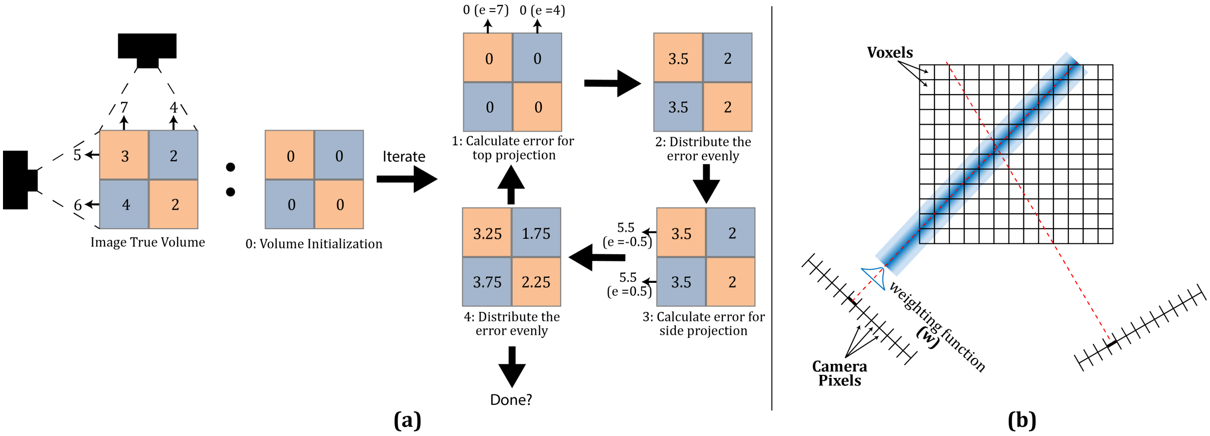


Figure 2.4: (a) A simple illustration of an iteration of an ART algorithm on a simple, defined, and well-posed 2×2 volume. (b) A larger volume grid showing the complexity of ART-based methods with a weighting function that relates voxels to pixels.

The combustion and flow diagnostic communities have widely adopted ART-based methods due to their performance using a small assortment of camera projections. ART models projections using ray-sums along the LoS through the volume (Eqn. 2.20), where the volume is discretized into cubic voxels. The scales that occur in many real-world flows can be very small, requiring dense volume discretization to capture those small structures [1]. Recent implementations have used voxel densities on the order of 10^6 to 10^8, requiring large amounts of memory storage. Couple the high voxel densities with modern scientific cameras with pixel densities reaching tens of megapixels, iterative algebraic reconstruction techniques have been introduced to solve for x of Eqn. 2.20 since $\mathbf{A}$ cannot be typically held in memory.

A simple example of the ART algorithm with parallel rays aligned with voxels is presented in Figure 2.4(a). The method assumes that ray projections are line integrals captured by a camera's pixels that can be represented as discrete linear ray-sums. The problem evolves to a system of linear equations for each projection and ray as first explained for tomography by Gordon *et al.* [66, 30]. However, the ART algorithm originates from the Kaczmarz method of projections [67] that is derived from linear algebra. The ART algorithm is updated via Eqn. 2.21 [30],

$$g^{(k+1)} = g^{(k)} + \mu \cdot \left(\frac{w_{ij}}{\sum_j w_{ij}^2} \left[p_i - \sum_j w_{ij} g_j^{(k)} \right] \right) \tag{2.21}$$

where w_{ij} is the weighting matrix between pixel i and voxel j that corresponds to **A** from Eqn. 2.20, and μ is a relaxation factor (generally, $0 < \mu \leq 2$) that can ensure numerical stability. Each ART iterates through each LoS measurement i to update g. The weight matrix w_{ij} represents the fractional area of the pixel size of each voxel. The relationship between pixel LoS and voxels can be approximated using a function w since the pixels and voxels do not align, unlike as presented in Figure 2.21(b). Several weight functions have been implemented for ART-based methods, where the selection depends upon computational cost and required accuracy. The weight matrix leads to a large computational overhead since each pixel has a corresponding weight for each voxel. Due to the increasing size of sensors and volumes, the weights typically cannot be stored in memory and have to be calculated as needed during each iteration. In addition to these steps, real-world volumes cannot have negative voxel values. Thus, a non-negativity constraint should be employed ($g_j = \max(g_j, 0)$).

The ART algorithm comprises two main parts: i) back-projection and ii) forward-projection (or the update step). The demonstration shown in Figure 2.4(a) has four projections with a well-defined system of only four unknowns. Like the original ART algorithm, the volume is initialized to all zeros due to its simplicity. The next step is to back-project along the LoS of the measured projections. For this case, the values of **A** are one along the LoS and zeros elsewhere, thus simply summing along the LoS since the projections align perfectly. The demonstration shows that the two projections are calculated simultaneously, but ART calculates them independently. Once the back-projection is calculated, the residual between the measured projection and the projection from the back-projection step are calculated ($e = p_i - \mathbf{A}_i * g$). The difference from the projections is distributed across the volume for the forward projection step along the LoS. For this case, with a relaxation of unity and unity of $\mathbf{A}_i$, it evenly distributes the error across the LoS. This will be done for each of the projections.

Once all projections are considered, ART will continue until an appropriate convergence criterion is satisfied. For the consistent case, where a subsequent g solves $\mathbf{A}_i g = p$, the solution will converge to the solution closest to the initial guess $g^{(0)}$. However, most applications use ART to solve ill-posed problems, where projections include measurement noise or image model errors. For these cases, each iteration of g has a unique minimum (known as the *limit cycle*) [68, 1]. To eliminate the issue of the limit cycle, the system of equations is squared to obtain the least-square solution.

$$g = \text{arg min}||\mathbf{A}^T(\mathbf{A}g - p)||_2^2 \tag{2.22}$$

As discussed by Grauer *et. al.* [1], this does not guarantee the best solution due to the limited number of projections. The least-squares solution has no direct physical relationship for volumes found within flow diagnostics like flames.

The original ART algorithm was adopted because of quick convergence with regard to the sum of squared errors. However, ART also results in pronounced salt-and-pepper noise distributed through the reconstruction [69, 70]. Convergence of ART depends on the order of the update sequence since ART updates g in a row-by-row basis. One way to overcome this is to randomly shuffle the order of the projections, which consistently outperforms sequential ordering in both accuracy and convergence speed [71]. The relaxation factor, μ, is a hyper-parameter that can increase convergence at higher values; however, μ increases sensitivity to noise or model error.

ART also suffers from *semi-convergence* [72, 73]. Semi-convergence is when g first approaches the best (error-free) solution but asymptotes to the least-squares solution. The closest approach to the best solution depends on the problem and the relaxation factor, thus making it difficult to determine the best stopping point. A simple stopping method is to halt the iterations after the difference between the projections hits a user-defined minimum error (δ_p) [74].

2.3.2 Multiplicative algebraic reconstruction technique

Another version of the ART algorithm is the multiplicative algebraic reconstruction technique (MART). As given by the name, MART consists of a set of linear equations given by Eqn. 2.20

like ART; however, the update step is a multiplicative factor rather than additive. A critical characteristic of the multiplicative step is that once voxels are zero, they remain at zero. The multiplicative factor improves convergence and reduces the number of updated voxels each iteration [1]. The MART algorithm was developed concurrently with ART by Godron *et. al.* [66]. In flow diagnostics, MART was first adopted for reconstructing PIV particle volumes as described by Elsinga [25]. The algorithm tends towards high entropy solutions that make it suitable for PIV. MART minimized line-of-sight artifacts that were found in ART and reduced the number of iterations required for convergence. MART is relevant for PIV due to the sparse nature of the volumes, where MART can generate satisfactory reconstructions with only four cameras. This is important for flow diagnostics due to the limited optical access and temporal requirements of different flows.

There are three main update step formulations for MART as evaluated by Verhoeven [60] and Mishra *et. al.* [35]:

$$g_j^{(k+1)} = g_j^{(k)} \left[1 - \mu \left(1 - \frac{p_i}{\mathbf{A}_i g_j^{(k)}} \right) \right] \tag{2.23}$$

$$g_j^{(k+1)} = g_j^{(k)} \left[1 - \mu \frac{\mathbf{A}_{i,j}}{\mathbf{A}_{i,max}} \left(1 - \frac{p_i}{\mathbf{A}_i g_j^{(k)}} \right) \right] \tag{2.24}$$

$$g_j^{(k+1)} = g_j^{(k)} \left(\frac{p_i}{\mathbf{A}_i g_j^{(k)}} \right)^{\mu \frac{\mathbf{A}_{i,j}}{\mathbf{A}_{i,max}}} \tag{2.25}$$

The last two formulations incorporate a weight normalization $\mathbf{A}_{i,max}$ to help stabilize reconstructions to noise and discretization errors. The relaxation parameter should be confined to (0, 1] because values greater than one can cause the solution to diverge due to the multiplicative property. Similar difficulties arise with MART, as was found in ART, where randomly ordering projections per iteration improves performance.

As presented in Eqn. 2.23, 2.24, 2.25, MART updates the volume using the ratio between the measured projection and the model back-projection instead of the difference between them like ART. This formulation inherently restricts the voxel values to $[0,\infty)$. The amount of update is proportional only to μ and the ratio of projections. MART's final solution and computational costs highly depend on the initial guess of g. Verhoeven *et. al.* [60] demonstrated that Eqn. 2.25

is the most robust MART formulation; therefore, it has been well accepted as the method of PIV [25].

Unlike ART, MART cannot be initialized to zero because the volume would remain zero due to the multiplicative update step. Hence, the most straightforward initial guess is to set all voxel values to one. Additionally, voxels that are not observed by any projections will not get updated and, thus, should be set to zero. While this initialization scheme is simple, initializing the volume to unity requires the algorithm to update every voxel until each voxel converges to zero. However, this can take many iterations, and the values will only approach zero. Therefore, other initialization techniques have been developed to improve MART's accuracy and computational time, where most techniques originated from PIV literature due to the sparse volumes. The three main procedures include multiplicative first guess (MFG) [75], multiplied LoS (MLoS) [40], and motion-tracking enhanced (MTE) MART [76]. While these have been demonstrated to improve performance for tomographic PIV, the effects will be less noticeable for a continuous emission or absorption light-field.

2.3.3 Simultaneous iterative reconstruction techniques

Thus far, the ART and MART methods update g going row-by-row through the projections, and conducting these algorithms this way is computationally expensive. Alternatively, the simultaneous iterative reconstruction techniques (SIRT) update g_j using all projections simultaneously rather than for each i projection. Like ART, several SIRT algorithms were first developed in the linear-algebra field before being implemented for volumetric imaging like Cimmino's method [77] and Landweber iteration [78]. A general form of a SIRT iteration, shown by Eqn. 2.26 was reported by Hansen and Jøgenson [73],

$$g^{(k+1)} = g^{(k)} + \mu_k \mathbf{T}\mathbf{A}^{\mathrm{T}}\mathbf{M}(p - \mathbf{A}g^{(k)}) \tag{2.26}$$

where the $\mathbf{T}$ is a $n \times n$ diagonal matrix that weights the back-projection for each basis function, and $\mathbf{M}$ is a $m \times m$ diagonal matrix that weights the residuals along the LoS of each projection. SIRT can be initialized in the same way as ART methods and iteratively updates the volume.

This section will be focus on SART, MART, and ASART. A detailed review of other SIRT algorithms is presented by Grauer *et. al.* [1].

Simultaneous ART

One of the more popular reconstruction algorithms for continuous scalar field flow field diagnostics is simultaneous ART (SART), first reported by Anderson and Kak [32]. SART combines the advantages of SIRT and ART without suffering the cost of SIRT. The SART algorithm is designed to produce smooth image back-projections after consideration of a ray-sum of a particular imaging direction, whereas SIRT does this after each full iteration. This allows SART to converge quickly like ART while reducing the salt and pepper noise [32]. SART weights the residuals using the length of each ray in the basis function [1]. The update step for SART is presented in Eqn. 2.27,

$$g_j^{(k+1)} = g_j^{(k)} + \frac{\mu}{\sum_i w_{ij}} \sum_i \left(\frac{w_{ij}}{\sum_j w_{ij}} \left[p - \sum w_{ij} g_j^{(k)} \right] \right) \tag{2.27}$$

The difference between ART and SART is that SART updates $g^{(k)}$ after the residuals from all projections are accumulated. Comparing Eqn. 2.21 to Eqn. 2.27, the big difference is the $\sum_i$ outside of the parenthesis that sums the residuals and is normalized by the column-sum of the weight matrix ($\sum_i w_{ij}$).

Simultaneous MART

Similar to SART, simultaneous MART (SMART) was developed to use a simultaneous iterative approach to reduce the effects of noise generated by th original MART algorithm. Mishra *et. al.* [35] devised the first SMART algorithm. Eqn. 2.28 presents the update step of SMART as

$$g_j^{(k+1)} = g_j^{(k)} \prod_{i \in \Upsilon_j} \left(\frac{p_i}{\mathbf{A}_i g_j^{(k)}} \right)^{\frac{\mu}{N_j} \frac{\mathbf{A}_{i,j}}{\mathbf{A}_{i,max}}} \tag{2.28}$$

where Υ_j is the set of LoS projections that interact with the jth voxel ($\mathbf{A}_{ij} > 0$) and N_j is the number of projections in Υ_j. Thus, SMART accumulates the product average of the ratio of measured projection to the back-projection. The average correction for SMART is less sensitive to projection noise than that of MART, making SMART more robust. The noise robustness, combined with MART algorithms yielding a maximum entropy solution, has made it a popular choice for tomographic PIV applications [40].

Adaptive SART

Another iteration of the SART algorithm called adaptive SART (ASART) was developed by Wan *et. al.* [70]. ASART aims to remove computational costs from SART while also weighting the residuals, not only with the length of the ray-sum, but also with the intensities of g along the ray-sum. ASART has been shown to decrease the computational time and improve performance with less projections compared to SART and FBP [70, 69]. The initial and updating steps of the reconstruction is given by Eqn. 2.29 and Eqn. 2.30,

$$g_j^{(0)} = \frac{\sum_i w_{ij} p_i}{\sum_i w_{ij}} \tag{2.29}$$

$$g_j^{(k+1)} = g_j^{(k)} + \frac{\mu}{4} \sum_i \left(\frac{w_{ij} g_j^{(k)}}{\sum_j w_{ij} g_j^{(k)}} \left[P_i - \sum_j w_{ij} g_j^{(k)} \right] \right) \tag{2.30}$$

where $E_j^{(k)}$ is the intensity of a volume element (voxel) j at iteration k, w_{ij} is the weighting between pixel i and voxel j, P_i is the intensity on the image for pixel i, S is the number of pixels per projection, and μ is a relaxation factor (generally, $0 < \mu \leq 2$) that can enforce numerical stability. The weighting function w_{ij} is an approximation of the complex relationship between voxels and pixels, and w_{ij} is calculated each instance of the iterations due to the memory requirement. ASART has been shown to improve accuracy and convergence rate for scalar field measurements compared to the family of ART algorithms [70, 69].

ASART algorithm incorporates a modified multilevel access scheme to arrange the order of projection data, adaptively correct the relaxation parameters to correct discrepancy between actual and computed projections, and a column-sum substitution to reduce memory and computation requirements compared to SART [70]. The column-sum substitution reduces memory by replacing an array of column sums with a scalar, which is important because it has shown that the column sums do not change during reconstruction and have no effect on the final solution from SART. SART and ASART have a view-by-view strategy, where the volume is updated simultaneously by all projections of each view direction. Overall, ASART improves reconstruction quality and has improved robustness to measurement noise and incomplete data [70].

2.3.4 Maximum likelihood expectation maximization

Another way of compensating for noise in the projections is to use an expectation maximization algorithm. These methods approach the tomographic problem statistically to determine the prediction of g that is more and less likely with respect to the measured projections. This algorithm allows for errors in the imaging model, such as noise. The projection model is presented in Eqn. 2.31

$$p = \mathbf{A}g + e \tag{2.31}$$

where e is the error vector. Expectation maximization algorithms are designed to calculate g to generate a resumed error distribution across the projections.

The maximum likelihood expectation maximization (MLEM) method attempts to maximize the probability of p given g, also known as the likelihood function ($\mathbb{P}(p|g)$). Since most image sensors capture the number of photons that interact with each pixel, the variation in this process (noise) can be modeled using a Poisson distribution [79]. The likelihood function can be formulated using the Poisson distribution as shown in Eqn.2.32 [1],

$$\mathbb{P}(p|g) = \prod_{i}^{m} \frac{(\mathbf{A}_i g)^(p_i)}{p_i!} \exp\left(-\mathbf{A}_i g\right) \tag{2.32}$$

where p_i is the photon count at the ith pixel of a sensor. Shepp *et. al.* [80] developed the first MLEM method. An updated version was reported by Lange and Carson [81], with the update step as provided in Eqn.2.33

$$g_j^{(k+1)} = \frac{g_j^{(k)}}{\sum_i w_{ij}} \sum_i \left(w_{ij} \frac{p_i}{\sum_j w_{ij} g_j^{(k)}} \right). \tag{2.33}$$

MLEM uses the same ratio of the measured and reconstructed projection as MART. However, instead of using that ratio to update the volume directly, MLEM weights the ratio with w_{ij}. Then MLEM calculates the sum across all projections and normalizes it by the column sum of w_{ij}. This provides a way of incorporating a physics model of noise into the reconstruction model. Unfortunately, Poisson-based MLEM is substantially costlier than ART while still producing similar results [82]. Therefore, MLEM is a less preferred tomography technique.

2.4 Regularization methods

As previously stated, the tomography is an ill-posed inverse problem for most flow diagnostics experiments. An ill-posed problem does not have a guaranteed unique solution, and the solution is sensitive to the initial volume prediction. One way to improve the robustness of the solution is to incorporate *a priori* information into the reconstruction algorithm. Incorporating physically motivated information into the algorithm is typically done using regularization schemes. Many classical regularization schemes have been applied to volumetric problems [83]. This section will be focused on Tikhonov, total variation (TV), and Bayesian regularization schemes as these are the most popular in the flow diagnostic community [1]. Of these, Tikhonov and TV regularization schemes favor smooth solutions.

Errors with the image model and noise can affect the measured projections, as demonstrated in Eqn. 2.31. The ART-based methods attempt to solve the inverse problem with a least squares solution, where errors and noise can cause large amplitudes of high spatial frequencies of the reconstructed volume [84]. The least squares minimization criteria include a regularization penalty term to avoid the noise problem. A general mathematical formulation with the new minimization problem is shown in Eqn. 2.34

$$\mathcal{J}(g) = ||p - \mathbf{A}g||_2^2 + \gamma\mathcal{R}(g), \quad 0 < \gamma < \infty \tag{2.34}$$

where the first term is the least squares criteria and the second term is the regularization criteria. The regularization criteria are composed of a regularization coefficient (γ) and a regularization function ($\mathcal{R}$) to enforce *a priori* information.

2.4.1 Tikhonov Regularization

Andrey Tikhonov developed Tikhonov regularization in 1963 [85]. This regularization method has $\mathcal{R}$ as the L2-norm as a function of the spatial intensities as shown in Eqn. 2.35 [1], where $\mathbf{L}_{\text{Tik}}$ is the Tikhonov matrix that operates on g to get a desired regularization. Tikhonov regularization can also be employed in matrix form using Eqn. 2.36, where 0 is an n x 1 vector of zeros.

$$g_{\text{Tik}} = \arg\min_{g} \left(||p - \mathbf{A}g||_2^2 + \gamma ||\mathbf{L}_{\text{Tik}} g||_2^2 \right) \tag{2.35}$$

$$g_{\text{Tik}} = \arg\min_{g} \left(\left|\left| \begin{bmatrix} \mathbf{A} \\ \gamma \mathbf{L}_{\text{Tik}} \end{bmatrix} g \; - \begin{bmatrix} \mathbf{p} \\ 0 \end{bmatrix} \right|\right|_2^2 \right) \tag{2.36}$$

There are two main versions used for flow diagnostics of Tikhonov regularization where the difference between them is $\mathbf{L}_{\text{Tik}}$. Zeroth-order Tikhonov regularization attempts to reduce the effects of noise by penalizing large magnitudes in the reconstruction. The zeroth-order Tikhonov regularization is employed by setting $\mathbf{L}_{\text{Tik}}$ as an identity matrix, which restricts the values of g. The restriction suppresses noise amplification. This technique is not advantageous for most scalar field measurements in flow diagnostics due to the limited number of projections used for most experiments [1] due to $\mathbf{A}$ being rank deficient.

Another Tikhonov regularization is the second-order method, where $\mathbf{L}_{\text{Tik}}$ is a discrete Laplacian operator (∇^2). Several finite difference schemes can be employed to approximate the Laplacian operator. In general, $\mathbf{L}_{\text{Tik}} g$ will be large when multiple high-magnitude gradients are present in g. Therefore, second-order Tikhonov will force g to a smoother solution as this will minimize $\mathbf{L}_{\text{Tik}} g$. The smoother approximation is useful prior information when the solution should follow the advection-diffusion equation that supports smooth scalar fields.

The regularization coefficient (λ) is critical for balancing the minimization of the measurement residual and Tikhonov regularization term. Several methods for selecting λ are presented by Idier [84], but a popular method is the L-curve first suggested by Lawson and Hansen [86] and further developed by Hansen [87]. The measurement residual and the regularization term are a function of λ, where $||\mathbf{L}_{\text{Tik}}g||_2^2$ versus $||p - \mathbf{A}g||_2^2$ are plotted at different λ forming the L-shaped curve. The curve is composed of two sections around an inflection point: i) the residual norm is relatively constant as the regularization term changes, and ii) the residual quickly rises from the inflection point with the regularization term remaining relatively unchanged [1]. The optimal λ occurs at the inflection point between these two regions of the curve.

2.4.2 Total Variation

Total variation regularization was first demonstrated by Rudin *et. al.* [88] for denoising of images and has been demonstrated for tomography [89, 90, 91, 26]. Total variation converges to a smooth solution while preserving sharp gradients in g. Eqn. 2.37 presents the TV regularization term ($\mathcal{R}(g) = ||g||_{\text{TV}}$) for a continuous function g, while the discrete version is Eqn. 2.38. ∇ is the 3D gradient operator, and ∇_x, ∇_y, ∇_z are the finite difference operator in the x, y, and z directions.

$$||g||_{\text{TV}} = \iiint_V |\nabla g| dV \tag{2.37}$$

$$||g||_{\text{TV}} = \sqrt{(\nabla_x g)_i^2 + (\nabla_y g)_j^2 + (\nabla_z g)_k^2} \tag{2.38}$$

TV has two main variations that utilize either the L1-norm ($||\nabla g||_1$) or the L2-norm ($||\nabla g||_2$). The L1-norm produces a piece-wise smooth function, allowing steep gradients to preserve boundaries. Alternatively, the L2-norm penalizes the high-magnitude gradients. Thus, fewer sharp gradients will be present in the solution. Unlike the Euclidean norm of Tikhonov, the L1-norm is not squared and thus does not penalize the larger gradients as harshly. Therefore, both TV variations are more accommodating to sharp edges or discontinuities in g. As explained by Grauer *et. al.* [1], the TV-norm minimization can be solved by nonlinear methods, allowing for Eqn. 2.34 to be nonlinear. Some examples include using a Gauss-Newton method [92], simulated annealing [26], and SIRT [89]. Grauer [89] showed that the SIRT algorithm can solve the minimization problem efficiently.

2.4.3 Bayesian formulation

Similar to Tikhonov and TV methods, Bayesian methods incorporate prior information to inverse tomography problems that are ill-posed and have limited projection information [93]. The Bayesian formulation incorporates a posterior that accounts for measurement and prior information. In contrast, MLEM attempted to use only a statistical model for the measurement model to account for imaging error. The Bayesian model parameters are taken as random variables that are characterized by a probability density function (PDF) [89]. The distribution of the PDF provides a measure of the confidence of the parameter value. The Bayes' equation, Eqn. 2.39, computes the posterior PDF ($\mathbb{P}(g|p)$) as a function of the likelihood and prior PDFs [1],

$$\mathbb{P}(g|p) = \frac{\mathbb{P}(p|g)\mathbb{P}_{pr}(g)}{\mathbb{P}(p)} \propto \mathbb{P}(p|g)\mathbb{P}_{p}(g) \tag{2.39}$$

where $\mathbb{P}(p|g)$ is the likelihood of the measurements given g, and $\mathbb{P}_{pr}(g)$ is the prior PDF of g. $\mathbb{P}(p)$ is a constant normalizing factor to conserve total probability. One difficulty with this formulation occurs due to the high dimensionality of the posterior PDF, where each element of g is a dimension. The final solution of g is the maximum posterior (MAP) estimate of $\mathbb{P}(g|p)$. The MAP can be calculated using only the likelihood and prior PDFs. A SIRT algorithm is typically used to solve these problems.

An accurate model of the measurement errors and *a priori* information is required to calculate the likelihood and prior probability. A similar approach to Eqn. 2.31 is generally used for the measurement error model. The Poisson model is a good approximation of a noise model, especially in low-light situations that require an intensified camera. However, with adequate signal levels, the Poisson distribution approaches a Gaussian shape. Thus, noise errors can be approximated using a centered Gaussian error vector [83]. Grauer *et. al.* [1] argues that the prior PDF is the most critical portion for the Bayesian reconstruction. For flames, a practical starting point for the prior function is a multivariate Gaussian prior as demonstrated by Kaipio *et. al.* [94].

That statistical formulation of the inversion problem provides a comprehensive solution since the posterior carries all information about g. The Bayesian formulation can give an uncertainty quantification of the solution that can be used to optimize the experimental system. Another advantage of this framework is that it can be formulated with other tomographic methods to provide an uncertainty quantification of the solution [9]. Cai [23] has proven that the Bayesian formulation is a tool to incorporate the *a priori* and *posterior* information effectively. A more detailed description of the Bayesian formulation for the tomography problem can be found in [89, 1].

2.5 Nonlinear Optimization

Tomography measurements become a nonconvex optimization problem when the field of interest has considerable in-scattering and self-absorption since the effects result in a highly nonlinear model. Even though Eqn. 2.20 is a system of linear equations that can be solved using the previous effectively with the previously described iterative methods, using nonlinear regularization functions like TV or a Bayesian prior produces nonlinear equations [1]. However, the iterative reconstruction methods described earlier cannot minimize the problem. Therefore, nonlinear or global optimization methods must be employed.

One of the simplest approaches to solving these convex problems is through an iterative gradient-based solver. These methods take iterative steps to approach a minimum using a general equation,

$$g^{(k+1)} = g^k + \alpha_k * s^k \tag{2.40}$$

For this equation, α_k and s^k is the size and direction of the step, respectively at iteration k. The step size controls the convergence rate and stability of the gradient-based solver. Where larger step sizes can approach the minimum quicker but can also cause the solution to diverge depending on the nonconvex function.

One method for determining the step direction is through the gradient of the nonconvex function. This technique is called steepest descent, where the step is calculated using

$$s^k = -\nabla F(g^k) \tag{2.41}$$

This step direction might be the simplest but generally leads to slow convergence and unstable solutions [1]. Neweton's method provides an alternative way to determine step direction by additionally incorporating the local curvature of F. The step direction can be generated through,

$$s^k = [\Delta_F(g^k)]^{-1}\nabla F(g^k) \tag{2.42}$$

where the Δ_F is the Hessian matrix of F evaluated at g^k. For Newton's method, the step size is generally set at unity. The quadratic fit of Newton's method stabilizes the performance but is only valid near the minimum. The Levenberg-Marquardt algorithm is a nonlinear optimization algorithm that combines the Newton method with gradient descent to provide a stable and robust minimization technique for nonlinear least square problems [95, 96]. This method initially begins with using the steepest descent method before transitioning to Newton's method to stabilize the approach to the minimum. Other nonlinear optimization methods, including conjugate gradient and Quasi-Newton methods, have been developed to solve the issues of steepest descent algorithms as described by Nocedal and Write [97].

Steepest descent and Newton's method-based algorithms have been developed for reconstructions [98, 99]. Unfortunately, these methods require matrix manipulation and inversion that can drastically increase computational time as the matrices (volume discretization) become significantly large. Therefore, the steepest descent or Newton's method, must use matrix inversion algorithms for each iteration. Press *et al.* [100] provides a comprehensive book on the different nonlinear optimization algorithms.

2.5.1 Genetic Algorithms

The main idea of genetic algorithms is to mimic natural selection or survival of the fittest process as a metaheuristic technique. The main components of a genetic algorithm are the individual coefficients (g_j), known as chromosomes, that make up the individual genotype ($\mathbf{g}$). The fitness of the genotype $\mathcal{F}(\mathbf{g})$ determines if the genotype will survive. These algorithms can be broken down into six main parts:

1. Initialization: To begin the algorithm, a population of individuals need to be generated, which is usually done with a random kernel. Initializing from a random kernel allows the algorithm to not get trapped in local minimum of the arbitrary function of interest.

2. Evaluation: Each chromosome is evaluated using $\mathcal{F}$ to determine a fitness score that is related to the suitability of the individual as a solution to the problem.

3. Selection: A subset of the previous generation is selected to progress to the next generation based on the fitness score. Individuals with the highest fitness are more likely to be selected.

4. Reproduction: The selected chromosome from the previous step are combined to produce a new set of individuals, where random changes are involved.

5. Replacement: The new individuals are then evaluated using $\mathcal{F}$ and replace genotypes from the current population to ensure evolution over time.

6. Termination: Reproduction and replacement are continued until termination conditions are met, which are typically based on $\mathcal{F}$.

Ultimately, the solution will have yielded the lowest value of $\mathcal{F}(\mathbf{g})$. Overall, Genetic Algorithms are a random search strategy to determine the global minima to an arbitrary function.

The previous description of a genetic algorithm has a limited description of the most critical part of the technique's performance: reproduction. After selecting the genotypes that will survive, the surviving individuals reproduce the new generation. The most common methods for reproduction are also influenced by natural processes that include mutation, crossover, and direct inheritance of chromosomes. First, mutation is the random perturbation of the chromosomes, where each chromosome has a chance to be mutated. Generally, the user specifies the number of mutations to use. The algorithm can modify mutations to fall within a range of values and base them on surrounding coefficients to impose certain qualities. Second, crossover involves mating two individuals from a mating pool of selected genotypes to produce offspring for the new generation. Mating is typically completed by randomly selecting an integer value k, where every chromosome (g_j) up to k is taken from one parent, and everything after is taken from the second, producing the offspring. Lastly, direct inheritance takes the exact genotype of the fittest individual from the previous generation. Genetic algorithms have many parameters that the user can adjust. Therefore, it can be challenging to optimize the selection of all the parameters as they affect the robustness and efficiency of the program [1]. For a more detailed description of a genetic algorithm, the authors highlight the work by Kramer *et. al.* [101].

Genetic Algorithms were first developed for tomography by Kihm *et al.* [36, 102]. Yang *et al.* [37] developed a hybrid method that combined an ART-based method with an evolutionary algorithm for flame reconstructions. Recently, a new genetic algorithm, the Evolutionary Reconstruction Technique (ERT), has been applied to several 3D combustion diagnostic measurements [38]. In this method, the individual chromosomes are an array of voxels representing the 3D scalar field. The fitness function used to qualify genotypes (3D scalar fields) uses the projection of the field and compares it to the measured 2D projections. Initially, ERT uses a pin-hole camera model to trace rays through the volume to render projections, while other camera models are expected in the future.

2.5.2 Simulated Annealing

Another integral metaheuristic approach applied to tomography is simulated annealing. Kirkpatric *et. al.* [103] developed the first simulated annealing algorithm for minimization of discrete problems. Simulated annealing algorithms are based on how materials are heated to a high temperature and then cooled slowly to produce a low-energy crystalline structure [104]. The structure might not reach the minimum energy state if the material is rapidly cooled. In this analogy, the material structure is the prediction of the volume, g. The free energy of g is calculated using $\mathcal{F}$. A heuristic parameter called temperature (T) is introduced to simulate the cooling process. There is no standard for initializing g, but a user could utilize a random perturbation method to calculate a candidate g'. Simulated annealing is a probabilistic method, where the candidate g' probability is calculated using

$$\mathbb{P}(g'|g^k) = 1 - \min\left\{1, \exp\left[\frac{\mathcal{F}(g^k) - \mathcal{F}(g')}{T_k}\right]\right\}. \tag{2.43}$$

T_k is the temperature at iteration k, that decreases with more iterations. To determine the next iteration structure g, the probability is compared to a random number $t \sim \mathcal{U}[0, 1]$

$$g^{(k+1)} = \begin{cases} g', & t \leq \mathbb{P}(g'|g^k) \\ g^k, & t > \mathbb{P}(g'|g^k) \end{cases} \tag{2.44}$$

Using this method, g' will always be accepted if $\mathcal{F}(g')$ ¡ $\mathcal{F}(g^k)$. Thus, selecting g' if it has the lower energy. However, g' still has a chance to be used for the next iteration, where the solution can have the energy increase after an iteration. The probability of g' being used depends on the free energy relative to g^k. This helps the solution escape shallow local minima. Generally, the initial iterations of simulated annealing are erratic, leading it to explore the solution space before transitioning to a monotonic progression towards a minimum as the system "cools".

Past literature has proven that metaheuristic algorithms can solve the tomography problem better than other derivative-free random search algorithms [104]. However, gradient-based solvers are more computationally efficient. Additionally, simulated annealing and genetic algorithms are highly sensitive to user-specific heuristics. Therefore, Grauer *et. al.* [1] recommends that users employ metaheuristic algorithms for problems with known nonlinearity.

2.6 Chapter Summary

This chapter covers the assortment of traditional tomographic reconstruction methods applied across combustion and flow diagnostics literature. The chapter summarizes the analytical methods, including FBP and Abel Transform. The basis of FBP is the Fourier slice theorem that proves that projections of g correspond to lines through the Fourier transform of g. Since real measurements can only capture a finite number of discrete projections, filters must be applied to remove non-physical noise during the inversion process. FBP is favorable for reconstructions of datasets that are composed of a large number of evenly spaced measurements. Abel transform method assumes that the flow field is radially symmetric about the primary flow axis. The inversion process is unstable and not commonly utilized. Iterative reconstruction methods like ART are the most common methods for 3D flow tomography as they are proven for limited projection data. Several ART-based techniques have been developed to improve the reconstruction accuracy and convergence rate. The simultaneous iterative methods are motivated by volumetric imaging, in which the reconstructions are simultaneously updated instead of updated row-by-row like other iterative methods. These iterative methods must filter out high-frequency artifacts that occur during the reconstruction. Regularization techniques are employed to remove artifacts from the reconstruction process; however, most of regularization methods drastically increase computational time and produce overly-smooth reconstructions.

The common goal of these methods is to use measured projections to determine the volume that produced those projections. These traditional tomography techniques require volume discretization, which is commonly implemented with uniform volume elements called voxels. Discretization leads to an inherent limitation on the spatial resolution of the reconstruction. Additionally, the computational cost of these methods scales with the number of volume elements. Therefore, traditional tomography methods have limitations on volume size or resolution caused by the current computational hardware. The following chapter presents deep learning-based methods for tomography, as this will provide the background and motivation behind the tomography method developed in this work.

Chapter 3

Deep Learning Algorithms

The fluid mechanics field has had many advances in analytical and CFD methods for numerical models of flows in the past century. These methods and the experimental data used for validation have produced immense amounts of data. The large datasets lead to the fluid mechanics field being a prime application for machine learning, which was first proposed by Kolmogorov [105]. Since then, there have been many advances in machine learning techniques that have increased the excitement for the technology, including the perceptron [106], neural networks (NN) with backpropogation [107], and convolutional neural networks (CNN) [108]. Although, until recently, many challenges have plagued these machine-learning methods, limiting their uses in fluid mechanics [109].

With the recent advances in hardware and algorithm development in the machine learning community, deep learning algorithms are becoming viable options for solving forward and inverse problems. Deep learning is a subset of machine learning that focuses on training artificial neural networks to learn and extract representations from data [110] automatically. One machine learning model is designed to mimic how the human brain processes information by utilizing neural networks composed of multiple layers. These neural networks, often referred to as deep neural networks (DNNs), consist of interconnected nodes, or “neurons,” arranged in many hidden layers. DNNs have been widely accepted for image classification, image segmentation, computational photography, and language modeling. In the flow diagnostic community, deep learning has generally been used for 2D PIV [111], super-resolution of PIV generated velocity fields [112], and dynamic masking using autoencoders [113]. Machine learning methods allow for a framework that can provide modular and flexible algorithms to address problems from CFD to diagnostics. These DNN methods have shown exceptional results and thus provide an appealing framework for tomography. The following sections of this chapter will cover i) a brief introduction to DNNs, ii) supervised tomography methods, iii) neural volume representations, iv) neural radiance fields, and v) physics-informed neural networks. These sections cover the current methods employed in the flow-solving and diagnostic communities while also providing the background for the current method developed for this work.

3.1 Machine Learning Background

Understanding the basic principles of machine learning is critical to comprehending deep learning methods. This section will focus on neural networks, a small piece of machine learning methods. For more comprehensive coverage of machine learning fundamentals, the author recommends books by Murphy [114] and Goodfellow *et. al.* [110]. Machine learning is a form of statistics that attempts to estimate complex functions using computers. Machine learning can be broken down into three main categories: i) supervised, ii) semi-supervised, and iii) unsupervised. The main difference between these methods is the way the models are trained and the data that is used to train the algorithm. Brunton [109] provides an overview of each and how they are applied in fluid mechanics. DNNs fall in the supervised category as they require a training data set to compare to the predicted output of the neural network that is used to update the network parameters.

3.1.1 Neural Networks

Neural networks are the quintessential deep learning supervised algorithm. Figure 3.1 shows the general structure of two of the most common neural networks, the original multi-layer perceptron (MLP) and the convolutional neural networks. Neural networks are fundamentally nonlinear function approximators as long as the network is sufficiently large. This originates from the Universal Approximation Theorem proposed by Hornik *et al.* [115] that states that any Borel measurable function can be approximated by a neural network of with at least one hidden layer, given the network has enough hidden units. The following section covers the description of these components, but the author recommends the book by Goodfellow [110] for

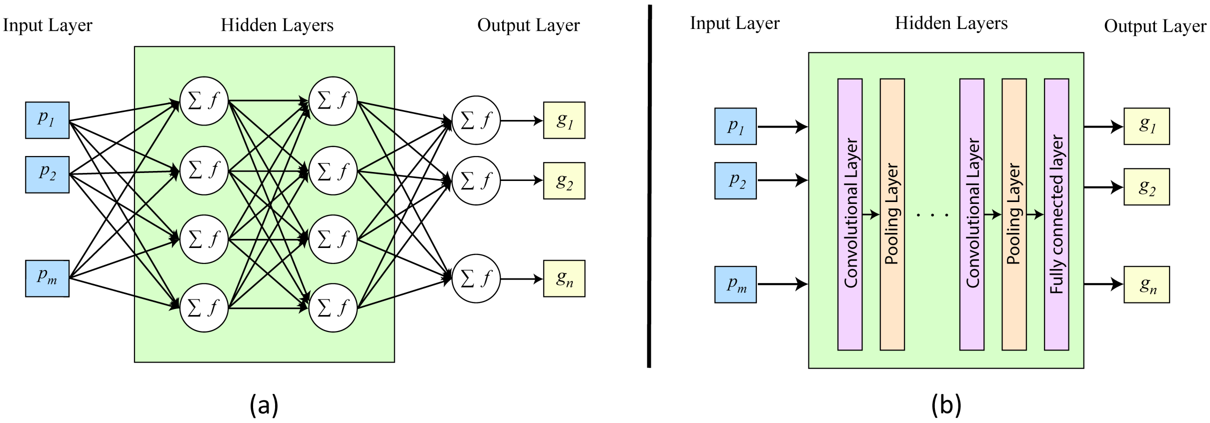


Figure 3.1: A simple schematic of a (a) two-layer, fully connected neural network with 4 nodes per layer and a (b) convolutional neural network with an arbitrary shape. Each method has a default m number of inputs and n number of outputs. The activation function for the neural network is indicated by f. (b) was adapted from [2].

a compressive explanation. The most common MLP is the feedforward algorithm, where the information flows from the input through the intermediate hidden layers before generating the output (Figure 3.1(a)).

Neural networks are called ’networks’ because they are composed of a string of connected smaller functions that attempt to approximate an overall function, as shown in Figure 3.1(a). The most common structure is for functions $(f_1, f_2, \text{and} f_3)$ to be arranged in a chain, such that $f = f_1(f_2(f_3(x)))$. Each function corresponds to a neural network layer, with f_1 as the first hidden layer, f_2 as the second, and f_3 as the third. The number of hidden layers determines the ’depth’ of an MLP. Depth refers to the number of layers in a neural network. Deeper networks improve the ability to fit a function, but they also increase the number of computations required for a prediction and can lead to issues like vanishing gradients.

The underlying unit of the neural network is called the perceptron, where an example schematic is displayed in Figure 3.2. The perceptron was developed to resemble the neurons of the brain, where a neuron takes in one or more inputs, processes them, and produces an output given the input [106]. The networks bear a resemblance to the neurons of the brain, hence the origin of the term “neural” for the network. Each node of Figure 3.1 is a perceptron linked with other perceptrons, constructing a fully connected network. The perceptron is a binary logistic

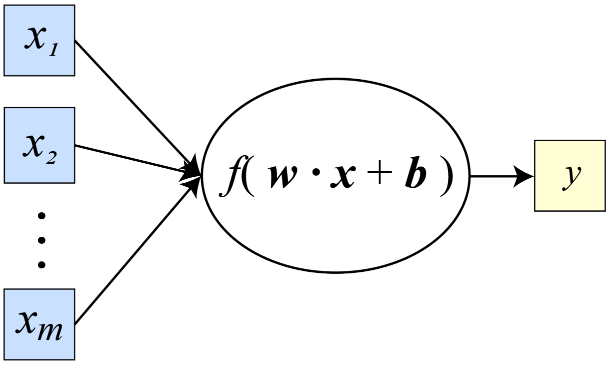


Figure 3.2: A schematic of a perceptron with m number of inputs and an activation function f imposed on the perceptron. $\mathbf{w}$ is the weights of the perceptron that is multiplied by the inputs $\mathbf{x}$, and b is the bias term.

model that accepts an input vector and manipulates it to a scalar "activation" value through

$$y = \sum_k (w_k x_k) + b, \tag{3.1}$$

where $\mathbf{w}$ is the weighting vector that is the linear operator on the input, and b is the bias. These are both unknowns that are "learned" from training data such that the perceptron learns to separate data for binary classification. The output of the perceptron is an "activation" value, but an activation function f can be added to a neuron to produce the ability to approximate nonlinear functions. The perceptron can only learn linearly separable functions since it is a linear array of weights with no activation function. For neural networks, the input into the MLP is the input layer, while the latter layers have input from the previous layers. The number of perceptrons per layer is the number of "hidden units" and is described as the height of the neural network. The weights and basis of hidden units are the trained parameters that allow the method to learn. The hyperparameters are the parameters chosen by the user, including the activation function, network size, and others.

The nonlinearity provided by the activation function applied to each hidden layer, as shown in Figure 3.2, allows the network to learn complex functions. Jarrett *et. al.* [116] emphasized that "using a rectifying nonlinearity is the single most important factor in improving the performance of a recognition system," among other hyperparameters of a neural network design.

Figure 3.3 presents five of the most common activation functions utilized for neural networks. The equations for each are provided below:

1. ReLU:

$$f(x) = \max(0, x) \tag{3.2}$$

2. Leaky ReLU:

$$f(x) = \begin{cases} a(x) & x < 0 \\ x & x \geq 0 \end{cases} \tag{3.3}$$

3. Sigmoid:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.4}$$

4. Hyperbolic Tangent

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.5}$$

5. Swish

$$f(x) = \frac{x}{1 + e^{-x}} \tag{3.6}$$

ReLU and Swish functions are unbounded in the positive x direction, while Leaky ReLU is also unbounded in the negative direction. Hyperbolic Tangent and Sigmoid bound the solutions, which can help normalize the output from those layers. Each of these activation functions provides different levels of complexity and advantages. Initially, activation functions that have discontinuous derivatives (ReLU and Leaky ReLU) were avoided due to the backpropagation method. However, they have been found to be useful [116] and are readily used in current computer vision methods. Activation functions have different pitfalls that are faced in deep learning methods, including sparse representation and vanishing gradients.

The most common way for MLPs to initialize the weights and biases is by assigning random values. The weights are randomized because the networks are trained in a gradient-based optimization strategy. The non-linearity of the neural networks produces a non-convex loss function. Stochastic gradient descent applied to non-convex functions is sensitive to the initial

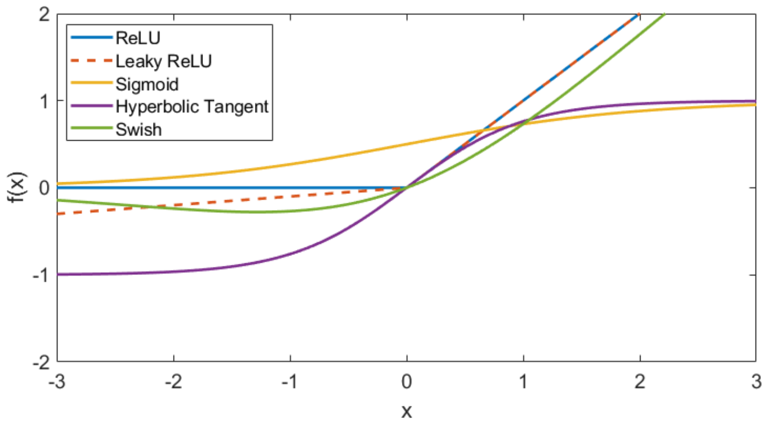


Figure 3.3: Visualization of common activation functions for neural networks.

parameters, and there is no convergence guarantee. Thus, small random values produce the best probability of convergence. For feedforward neural networks, training consists of providing an input with a known output. The predicted value from the network y is then compared to the known output $\hat{y}$ through the loss function

$$\mathcal{L} = f_{loss}(y, \hat{y}). \tag{3.7}$$

There are several loss function options, each impacting the method's performance. This method is supervised learning because a known output is supplied to update the network. The most common way of optimizing the network is through the back-propagation method [107]. The back-propagation algorithm is a stochastic gradient descent algorithm that uses the loss function to update the weights and biases after each iteration or epoch of the algorithm. Since feedforward neural networks are functions chained together as explained above, the gradients relative to $\mathcal{L}$ can be found and used to update the weights and basis of each layer and node. The training is repeated until the loss function converges to a minimum value or reaches the maximum number of iterations.

3.1.2 Convolutional Neural Networks

Convolutional neural networks are a subset of neural networks where each node uses convolution instead of linear vector multiplication, as developed by LeCun *et al.* [108]. CNNs are known for processing a grid-like topology, typical for many convolutional methods in flow diagnostics like PIV. CNNs are most common for processing 2D images or a 1D time series. CNNS, unlike traditional MLPs, typically have sparse interactions due to the convolutional kernel being smaller than the input (i.e., image). Sparse interactions allow the network to store fewer parameters and improve statistical efficiency [110]. Another advantage of CNNs is the ability to share parameters, which refers to the fact that a single set of parameters (kernel) is learned for each layer since the kernel is used at every input position. A standard MLP requires a different parameter set (w) for every input.

The convolution equation used for machine learning does not directly correspond to that used in other fields, such as engineering and mathematics. For CNNs, the general convolutional step for a 1D input can be calculated using

$$s(t) = \int x(a)w(t-a)da = (x * w)(t), \tag{3.8}$$

where w is the kernel of the CNN that is learned, and s is the output, also referred to as the feature map. The kernel of the CNN must be a probability density function to ensure the output is the weighted average of x. As discussed, one of the most practical use cases for CNNs is with 2D images. Therefore, a 2D discrete convolution equation must be employed as presented in Eqn. 3.9.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{3.9}$$

where I is the 2D image with a size of $m \times n$and K is the 2D kernel. While this has been implemented in CNNs, it is popular for many libraries to employ the cross-correlation method without using the commutative property to flip the kernel in the convolution.

After the convolution layer of the CNN, the output goes through a nonlinear function similar to neural networks. Finally, the last step is the pooling layer/function, also called the detector stage. The last step is a crucial step that sets CNNs apart from other models. The pooling layer produces invariance to small perturbations of the input (i.e., small shifts of a subject in an image). It divides the input feature map into non-overlapping regions, often called pooling regions or pooling windows, and computes a single output value for each region. By reducing the spatial dimensions, the pooling layer allows the network to focus on the most salient features and extract them regardless of their precise location. This is a key feature that enables the CNN to detect features that are globally present in the input. Some of the most popular pooling techniques are max pooling, L^2-norm, and weighted average [110]. Therefore, as the information is passed through the CNN, the input dimension into the next layer is reduced. An excellent review article for solving inverse problems using CNNs was presented by McCann *et al.* [117].

Many readily available libraries have been developed for all of the general DNNs covered above, including Tensorflow and PyTorch. These methods provide easy access to many of these features, empowering machine learning practitioners and researchers to create customizable networks without coding the underlying network structure and updating steps. These libraries enable the machine learning community to develop complex algorithms with just a couple hundred lines of code rather than thousands.

3.2 Deep Learning in Volumetric Imaging

Most of the deep learning for volumetric imaging consisted of DNNs with m input nodes corresponding to the pixels of a single image and n output nodes for each voxel of g, which is depicted by a simple schematic in Figure 3.1. This strategy requires the network to be trained using a synthetic data set with a known ground truth that is used to generate perspectives. For training, the perspectives are input into the DNN, and the predicted g is evaluated with the ground truth $\hat{g}$ that was used to generate the perspectives. Simple imaging models generate the training perspectives from the ground truth. The following will cover the DNN-based methods that have been developed for volumetric imaging.

One of the first demonstrations of tomography using DNNs was by Huang *et al.* [2] for tomographic absorption spectroscopy (TAS). In their work, a CNN was trained with input from the spectral perspective information. The output was a predicted 2D phantom g that was discretized into 60x60 voxels. Each output node was a different voxel. The CNN was trained with a known truth volume and perspectives, where the loss function was a normalized L^1-norm. The CNN was composed of two convolutional layers and one pooling layer. The activation function was Leaky ReLU. After the second convolutional layer, the 252 feature vector was fully connected to a single output vector with a length of 3600 that corresponded to the 60x60 voxel volume. The work was compared to a simulated annealing algorithm, where the CNN was proven to be more robust to noise and quicker to train with increasing input data. However, the CNN was trained on phantoms with similar characteristics, and the same imaging model was used to produce input perspectives.

Jin *et al* [45] developed a similar model for three-dimensional tomography. Again, the DNNs had the projections as the input for this work, and the predicted volume was the output. Instead of training with a known phantom, the CNN was paired with ART, where ART would be considered the ground truth for training the network. Training the network with ART reconstructions allowed the network to generate similar results as ART but be able to do it in fractions of the time. This method was demonstrated using a phantom and experimental chemiluminescence. The challenge with training with ART reconstructions is that CNN will learn the same issues that plague ART and will be limited to the imaging model used for ART.

After the work on Jin, Huang *et al.* [3] expanded to a 3D tomography method for combustion diagnostics called VT-Net. The CNN was similar to Jin's work; however, this work implemented a residual network. The residual network includes a skip connection that links the input to the output of the CNN as shown in Figure 3.4. The skip connection helps reduce the effect of the vanishing gradients problem to allow deeper networks (tens of layers deep). The residual network did not indicate accuracy improvement over the regular CNN but helps stabilize training. This work compared a residual network to a typical CNN and ART. At a low number of projections, both the CNN and the residual network outperformed ART; however, ART surpassed the DNNs after the number of projections was greater than nine.

With the progress of CNNs in the volumetric imaging field, several techniques continued with similar CNN structures [118, 119, 120, 4]. All of these methods require a training set before they can be applied to tomography problems. Thus far, the methods for generating the training set has been:

1. phantoms generated from CFD simulations,

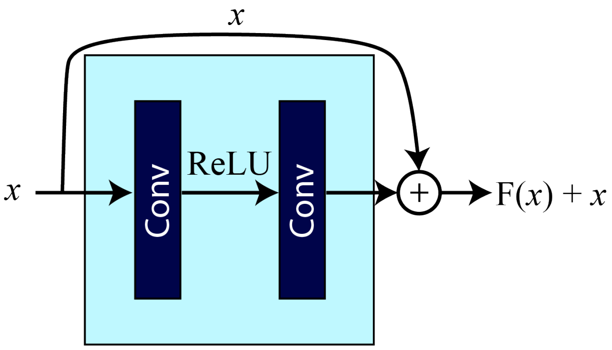


Figure 3.4: Schematic of a residual network with a skip connection. Adapted from [3].

2. generating phantoms from random Gaussian blobs,
3. using traditional tomographic methods (e.g. ART) to produce volumes as the ground truth.

Supervised DNN tomography methods, once trained, demonstrate impressive computational times for producing reconstructions, outperforming traditional tomography methods with times in the millisecond range. However, the effectiveness of these DNN tomography methods hinges significantly on the network's hyperparameters and the thoroughness of the training set. As one might anticipate, the network's structure, activation functions, and loss function all contribute to the DNN's ability to solve the inverse problem. Moreover, DNNs are recognized for their poor extrapolation capabilities. Consequently, it's crucial that supervised DNN methods accurately represent the physics of the volumetric imaging problem, which restricts the use of supervised DNN methods for volumetric imaging.

3.3 Neural Representations

Neural Representations have been critical in the past few years for shape representation, depth estimation, novel view synthesis, and other problems investigated by the computer vision community. All of these problems are related to tomography. This next section covers some neural representation methods for solving the above problems, including explicit and implicit representations.

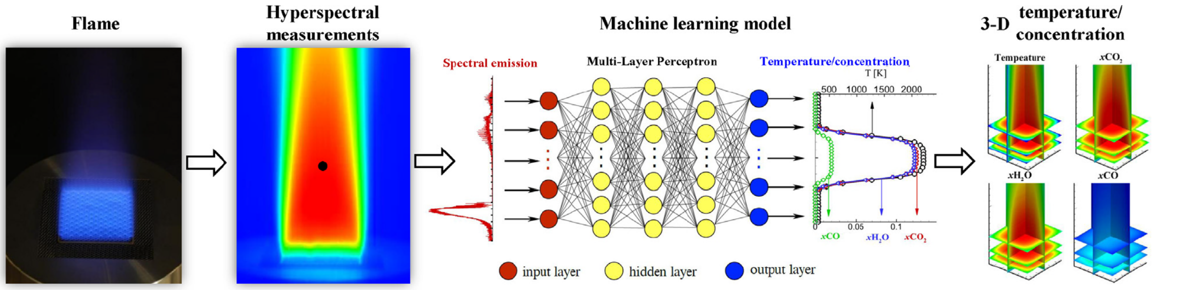


Figure 3.5: Schematic of a neural explicit representation that relates measurement data to QoI for each volume element. This figure was taken from the work by Ren *et. al.* [4].

3.3.1 Neural explicit representations

Neural explicit representations use DNNs to represent volumes for explicit size and discretization, including point cloud, meshes, or voxels. Figure 3.5 illustrates an explicit representation model that relates measurement data to voxel quantities. Most deep learning methods discussed in the previous section fall under explicit neural representations. Yifan *et al* [121] describes a differentiable surface splatting technique (DSS) that incorporates neural representations to conduct inverse techniques to produce multiple color views and depth maps of a scene. The splattering technique utilizes a point cloud-based representation, representing each point as a disk or ellipse. The DNN solves the inverse rendering problem, leading to better results than traditional DSS techniques.

Another neural explicit representation technique developed for novel view synthesis is Deep Voxels [122]. Deep Voxels aims to generate a 3D representation of a scene by learning to predict the occupancy or presence of objects within a voxel grid. Each voxel in the grid can either be empty (not occupied by an object) or occupied (part of an object). Thus, the 3D feature representation is based on occupancy for each voxel and predicted using a set of DNNs. The network takes a set of 2D views captured from different viewpoints and processes this input through several convolutional layers to extract meaningful features for each voxel. The technique indicates improved results compared to other methods but is limited to smaller images and volumes.

Neural Volumes [123] is another well-known method for view synthesis and shape representation using an explicit model. Neural volumes is similar to Deep Voxels, where images

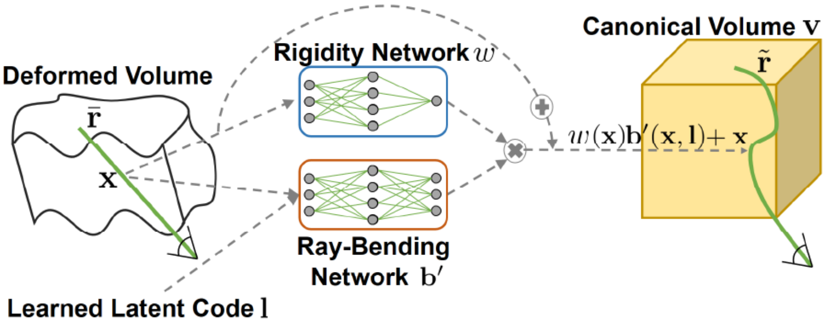

Deformed Volume
$\bar{\mathbf{r}}$
$\mathbf{x}$
Learned Latent Code $\mathbf{l}$
Rigidity Network w
Ray-Bending Network $\mathbf{b}'$
$w(\mathbf{x})\mathbf{b}'(\mathbf{x},\mathbf{l})+\mathbf{x}$
Canonical Volume $\mathbf{v}$
$\tilde{\mathbf{r}}$

Figure 3.6: Schematic of a neural implicit representation that relates points in space to QoI. Figure is taken from Tretschk [5].

are input into an encoder-decoder DNN structure to output feature mapping to a 3D volume discretization. This method has two main parts: i) an encoder-decoder network that solves the inverse problem to produce a 3D volume, and ii) a differentiable image rendering step given a set of camera parameters. The output of the ray marching step produces an RGB image, an alpha mask over the subject, and a background image.

All of these methods showed impressive results for novel view synthesis. However, since the output of these methods is discretized, they suffer from resolution limitations. These methods are similar to those developed for flow diagnostics discussed in the previous section. Additionally, these techniques struggle to expand to higher resolutions of volumes and images due to computational time and memory requirements [6].

3.3.2 Neural implicit representations

The computer vision community has recently developed implicit volume representation methods to overcome the limitations found for neural explicit reconstructions. Figure 3.6 depicts an implicit model that relates coordinates of the volume to a QoI. One of the first implicit methods, DeepSDF [124], models 3D geometries as a continuous function approximated by a neural network using a signed distance function (SDF). An SDF is a function that outputs a distance from a spatial location in the volume to the closest surface. DeepSDF employed an encoder-decoder network structure to handle an arbitrary number of shapes in a volume to provide a distance.

The DeepSDF method proved to smooth out surfaces drastically. Chabra *et al.* [125] expanded the technique by taking patches of the images rather than the whole pictures.

Additionally, Chibane [126] developed a similar technique that uses unsigned distance functions. Alternative to SDFs, occupancy fields have been used to represent 3D shapes/surfaces [127]. For these methods, the network approximates the occupancy field as a continuous function, where the network's output is the probability of occupancy ϵ {0, 1}. While these techniques have improved shape representations, they are limited by the need for ground truth 3D geometries for training. This leads to similar issues explained for supervised tomography.

Niemeyer *et al.* [128] and Sitzmann *et al.* [129] developed two techniques that overcome the supervised limitation. Niemeyer [128] introduced Differentiable Volumetric Rendering (DVR). DVR uses an implicit representation of the 3D occupancy field, where a numerical ray marching algorithm is employed to determine the location where the ray corresponds to a pixel of an image intersects the object. Once the ray intersection spatial point is chosen, the spatial point is input into another network that models a 3D texture field to predict a color. The predicted color is then used to update both networks. Sitzmann [129] demonstrated a method called Scene Representation Networks (SRNs) that outputs a feature vector and an RGB color for a 3D coordinate. This method uses a similar ray marching technique with a recurrent neural network structure to determine the location of the surface. These last two methods would be characterized as semi-supervised since the 3D field is learned from projections of the scene [109], where a ground truth 3D field is not known.

These implicit representation methods lead to the development of Neural Radiance Fields (NeRF) [6], where this method is a semi-supervised implicit representation approach. This method overcomes many of the challenges of both 3D shape representation and novel view synthesis problems. NeRF is the foundation of the FluidNeRF method developed for this work; therefore, we will describe Neural Radiance Fields in detail in the following section.

3.4 Neural Radiance Fields

The underlying novel concept of Neural Radiance Fields is that the light-field of a scene/volume can be represented as a continuous function approximated by a MLP as described by Mildenhall *et. al.* [6]. Most of the tomographic reconstruction techniques presented in the previous chapter used a voxel-based volume discretization method that inherently limits the topology of the reconstruction. NeRF provides a compact way to represent a volume without the same limitation as discretization methods while also not being limited by requiring ground truth volume information for training. The technique trains similarly to traditional algebraic tomography methods like ART, where the volume (neural network) is updated with the error between the measured and predicted projections.

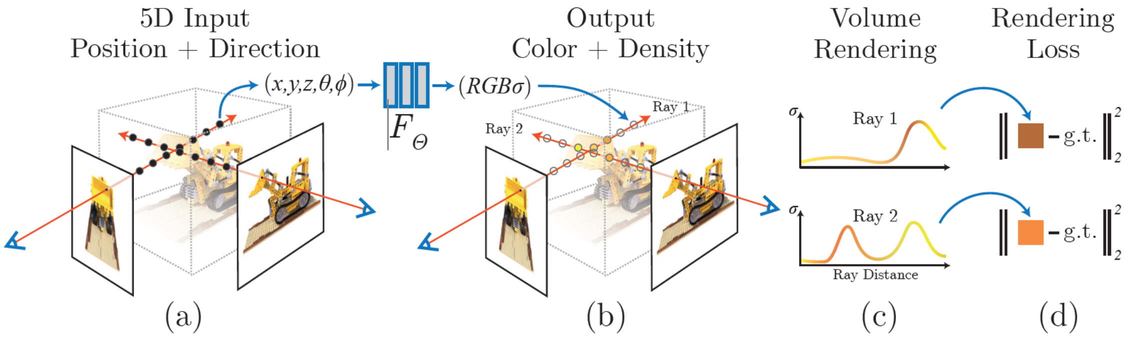


Figure 3.7: Neural Radiance field schematic showing (a) the input 5D sampling coordinates, (b) output from the neural radiance field from those samples, (c) rendering technique along that ray, and (d) the loss function used to update the neural radiance field using the ground truth observed images. This figure was taken from Mildenhall *et. al.* [6].

A Neural Radiance Field is a machine learning technique using a 9-layer fully connected neural network to represent a volume. Mildenhall *et. al.* [6] developed the method for novel view synthesis of complex scenes. The continuous volume/scene is characterized by a 5D vector-valued function with the inputs of i) a 3D location in the volume (x, y, z) and ii) a viewing direction ($\boldsymbol{d} = \theta,\phi$) as shown in Figure 3.7. The NeRF deep network used in this work outputs RGB color ($\boldsymbol{c}$ = <r, g, b>) and volume density (σ). The volume density can be referred to as the differential probability that a ray terminates at a 3D coordinate. The neural network can be thought of as a continuous function that represents the imaged volume,

$$[\boldsymbol{c}, \sigma] = F_{\Theta}(x, y, z, \theta, \phi) \tag{3.10}$$

where F_{Θ} is the continuous function defined by the neural network.

The neural network limits the prediction of volume density as only a function of location, while the color, $\boldsymbol{c}$, depends on the whole 5D input vector. Adding viewing direction helps the representation be multi-view consistent [6], where it can help with reflections and other lighting changes. The first eight fully connected layers process the input of 3D coordinates to output both σ and a 256-dimensional feature vector. Each layer comprises 256 hidden units with the ReLU activation function. The feature vector and viewing direction $\boldsymbol{d}$ are concatenated and passed into the last fully connected layer (ReLU activation and 128 channels) that outputs the view-dependent RGB color (intensity for monochromatic).

While this paper is focused on volume reconstruction and not view synthesis, view synthesis is a crucial step for updating the neural network via the loss function, as shown in Figure 3.7. View synthesis is the same as the projection step of ART-based methods, where ART projections are rendered using a complex weighting scheme accumulated through the volume. However, for NeRF, the images are rendered using rays that are propagated through the scene using classical volume rendering techniques as first described by Kajiya and Herzen [130]. The color or intensity of a ray through the volume is determined by

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))\boldsymbol{c}(r(t), \boldsymbol{d})dt \tag{3.11}$$

where $C(r)$ is the expected color for ray $r(t)$ = $\boldsymbol{o}$ + t$\boldsymbol{d}$, where the near and far bounds of the ray are t_n and t_f, respectively. The function $T(t)$ is the “line-of-sight” accumulation that occurs along the ray, where T(t) is given by

$$T(t) = \exp\left(-\int_{t_n}^{t_f} \sigma(\boldsymbol{r}(s))ds\right) \tag{3.12}$$

This function describes the probability that a ray will not terminate between t_n and t_f. The integral can be approximated using the numerical quadrature method. Instead of rendering on a discretized voxel grid, the NeRF method uses a stratified method not to limit the resolution of the neural network. The stratified sampling approach generates random ray samples from N evenly-spaced bins between t_n and t_f, where one random sample is selected from each bin. Compared to the ART-based methods in which the samples are fixed, this stratified approach can help get sub-samples inside a bin/voxel.

At this point, we have a deep network that provides intensity and volume density as a function of 3D coordinates and viewing direction. While this can represent a scene (volume), additional improvements can be incorporated to allow the neural network to capture the high spatial frequency and complexities of scenes. Neural networks are notorious for being biased towards a lower-frequency function as described by Rahaman *et. al.* [131]. One way to overcome this bias includes mapping the 5D input coordinates to higher dimensional space to allow the network to improve the approximation of the higher-frequency information [6]. This is referred to as positional encoding, where a Gaussian Fourier feature mapping function is utilized to take the input coordinates from $\mathbb{R}$ to $\mathbb{R}^{2L}$ as introduced by [6]

$$\gamma(p) = (\sin(2^0\pi p), \cos(2^0\pi p), ..., \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)) \tag{3.13}$$

where γ is applied separately to 3D position coordinate (x, y, z) and $\boldsymbol{d}$. L is 10 for x, y, z and L = 4 for $\boldsymbol{d}$.

One crucial step in rendering images is to use efficient discrete samples throughout the volume. NeRF introduces a hierarchical volume sampling method that generates informed sample locations in the volume by using a lower resolved set of stratified samples. For their method, new samples are generated from the accumulated transmittance. Hierarchical sampling can remove samples that do not contribute to a particular ray due to occlusion and increase samples in regions of higher intensity/optical density. For the NeRF method, hierarchical sampling is accomplished by simultaneously optimizing two networks that use “coarse” and “fine” sampling methods. The first set of course samples is generated using stratified locations along a ray. Stratified sampling splits a ray into N_c evenly spaced bins between a near and far field, where a random sample is generated inside each bin. This yields N_c course samples. The N_c

course coordinates are used to calculate the color and optical density along the ray for each sample. The color of the course ray is generated by weighting the color for each sample point, c_i, as computed using

$$\hat{C}_c(r) = \sum_{i=1}^{N_c} w_i c_i \tag{3.14}$$

where $w_i = T_i(1 - \exp(-\sigma_i \delta_i))$. After stratified sampling, a more informed sampling can be made from σ of the coarse network. The fine samples are found by normalizing the weights to produce a probability density function for the additional N_f samples. The fine network rendering uses all $N_c + N_f$ samples. The advantage of this technique is that more samples are allocated to regions with more visible content/features. Overall, the NeRF technique attempts to minimize the difference between the captured projections and the reprojections from the reconstructed volume, which is the same concept that is conducted by ART-based methods.

NeRF advancements have been pivotal in the realm of computer vision. A plethora of literature has been proposed to enhance the method, with techniques developed for various aspects such as view synthesis [132, 133, 134, 135], rendering speed [136, 137, 138, 139], generative image synthesis [140, 141], few-view novel view synthesis [138, 142], and dynamic scenes [143, 48, 144, 5]. Gao et al. [46] provides a comprehensive review of these developments. Notably, most techniques do not primarily focus on the accuracy of resembling the radiance field (g), but improvements to novel view synthesis can inherently enhance the volume approximation. A significant breakthrough in NeRF for 3D reconstructions using a single x-ray view, termed MedNeRF, was introduced by Corona-Figueroa *et al.* [145]. This technique utilizes the generative radiance field method, demonstrating impressive reconstructions with as minimal as a single x-ray view. Many of the methods require other networks that are trained for image classification or segmentation, which can require those other networks to be trained with alternative data. Using trained networks can lead to issues when applying to unknown problems. However, most of the developments in the NeRF community are application agnostic, allowing these improvements to be integrated into our model.

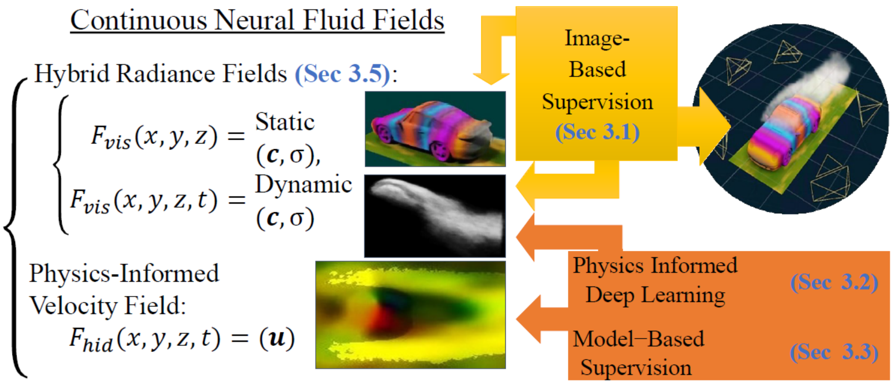

Continuous Neural Fluid Fields
Hybrid Radiance Fields (Sec 3.5):
$F_{vis}(x,y,z) =$ Static $(\boldsymbol{c},\sigma)$,
$F_{vis}(x,y,z,t) =$ Dynamic $(\boldsymbol{c},\sigma)$
Physics-Informed Velocity Field:
$F_{hid}(x,y,z,t) = (\boldsymbol{u})$
Image-Based Supervision (Sec 3.1)
Physics Informed Deep Learning (Sec 3.2)
Model–Based Supervision (Sec 3.3)

Figure 3.8: Schematic of the NN models and supervision that was employed by Chu *et al.* [7]. The method utilizes supervision from measurements, physics equations, and a pre-trained fluid network to calculate the radiance field and the velocity of the dynamic fluid.

3.4.1 Neural Implicit Representations in Flow Diagnostics

A few techniques for flow diagnostic techniques have been developed using a neural implicit volume representation [7, 146, 8]. Chu *et al.* [7] was the first to demonstrate the use of a NeRF-based technique for flow field measurements called physics-informed neural radiance fields (PI-NeRF). Their innovative hybrid method employs a two-network approach, with a static and a dynamic network, to allow each network to model the static and dynamic portions of the scene independently. The hybrid representation showed impressive results in automatically distinguishing the scene's dynamic fluid and static components. Like NeRF, their networks are functions of 3D spatial location (x, y, z) and time (t) for the dynamic network that outputs color and optical density. A third network is used to calculate the velocity of the fluid being imaged, called the hidden network. This network learns from physical priors (i.e., Navier-Stokes equations), which was inspired by Physics-Informed Neural networks, which will be described in more detail in Section 3.5.

The network structure was integrated similarly to NeRF, with the most significant changes being activation and loss functions. For the neural network that models the dynamic scene, Chu *et al.* [7] adopted a SIREN activation function that was proposed by Sitzmann *et. al.* [147]. The periodic function captures the continuous derivatives of the scene, which is essential for physics-informed loss. For their loss function, the PI-NeRF model adopts the same projection loss utilized by NeRF, which compares the measured and predicted projections. In addition to

MSE, a VGG-based image-based loss function was employed to improve the perceptual quality of the reconstructed image by capturing more of the high-frequency detail [148]. The VGG loss functions is given by

$$\mathcal{L} = \sum_{\Phi \epsilon \text{VGG layers}} \left(1 - \frac{\Phi(\hat{I}) * \Phi(I)}{||\Phi(\hat{I})|| * ||\Phi(I)||} \right). \tag{3.15}$$

The VGG uses the cosine similar loss (Eqn. 3.15), where it operates on small 40x40 patches between the measured and rendered images.

In addition to the image-based losses, the method includes two different loss functions to enforce physic priors on the flow field predictions, including the mass transport theorem and the Navier-Stokes equation. The first loss is the substantial derivative of the optical density, and the second is the simplified Navier-Stokes equations,

$$\mathcal{L}_{\frac{D\sigma}{Dt}} = \left(\frac{\partial \sigma}{\partial t} + \mathbf{u} \cdot \nabla \sigma \right)^2 \tag{3.16}$$

$$\mathcal{L}_{\text{NSE}} = \left\| \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right\|_2^2 + w_{div} \left\| \nabla \cdot \mathbf{u} \right\|_2^2 . \tag{3.17}$$

The substantial derivative of optical density is analogous to optical flow, which preserves the scalar (optical density) in the volume. This loss helps relate the NeRF network to the flow network. The Navier-Stokes equations include the momentum and continuity equations. Chu's work assumed that the right-hand side of the Navier-Stokes equations can be dropped off (assumed negligible), which can be limiting. In addition to these loss equations, the ray modeling was modified to account for the dynamic scene. The rays were morphed with time to ensure temporal consistency similar to Non-Rigid NeRF [5]. This warping is only done for the optical density since color is not a transported scalar [7]. The velocity field from the hidden network determines the location of the ray with time. Thus, the NeRF network does not include time as an input.

Chu initially found that vorticity was undetermined for their method due to the optimization of Eqn. 3.16 with a blurry signal of σ. Therefore, the vorticity approximation is improved by comparing the vorticity from a pre-trained flow network to the velocity prediction from F_{hid}

using an MSE loss function. The pre-trained network enhanced their vorticity approximation; however, it requires a pre-trained model. The pre-trained model can reduce the generality of the PI-NeRF method. In addition to the vorticity problem, NeRF has a well-known issue with producing non-physical optical densities when applied to datasets with limited projections ($<$ 20). They solve this problem by incorporating another loss function ($\mathcal{L}_{ghost}$) that uses regularization to penalize the "ghost" densities. With all these corrections, physics-informed neural fields demonstrated impressive results in distinguishing the flow field from static objects in the scene. However, there were a lot of additions that caused the method to be bulky and lack generality.

An additional method, conducted concurrently with the work of this dissertation, is the Neural Volume Reconstruction Technique (NVRT) demonstrated by Zhang *et al.* [8]. The NVRT technique was demonstrated for computed tomography of chemiluminescence (CTC), although it can be adapted for other emission-based flow diagnostics. The reconstruction method was inspired by NeRF as indicated by Figure 3.9. Similar to NeRF, NVRT encodes the 3D spatial coordinates using positional encoding before being passed into a fully connected network. The network consists of eight layers with 256 hidden units per layer that utilize leaky ReLU activation connected to four downsampling layers that decrease the height from 128 hidden units to one output node. The last down-sampling layer uses the soft-plus activation to predict a non-negative intensity output. Unlike NeRF, NVRT was developed for emission-based tomography. The model's output is the emitted intensity at the given 3D coordinate. The perspective p is rendered using an integrated line-of-sight emission model shown in Eqn. 3.18.

$$p = \sum_i w_i c_i \tag{3.18}$$

where w_i is the weight for each intensity c_i of the chemiluminescence at the i^{th} ray sample. The weight is calculated as the normalized length of each i^{th} segment $\left(w_i = \frac{l_i}{\sum_i l_i}\right)$.

The network is updated using the MSE between the rendered and measured projections. For this simplified NeRF model, Zhang utilizes a single network approach, unlike NeRF [6, 7]. NVRT removed the hierarchical sampling method, where global sampling was conducted using the semi-random sampling method. Semi-random sampling is the same as the first step of the NeRF method, where one random sample is generated for each bin along the ray. Similar to Chu [7], NVRT assumes diffuse emissions with no scattering that allows the model to drop the view direction dependence. Overall, NVRT was demonstrated to outperform ART using both synthetic and experimental datasets, and NVRT was more robust in terms of noise due to the neural implicit volume representation. The demonstration did not include hierarchical sampling and did not investigate hyperparameters.

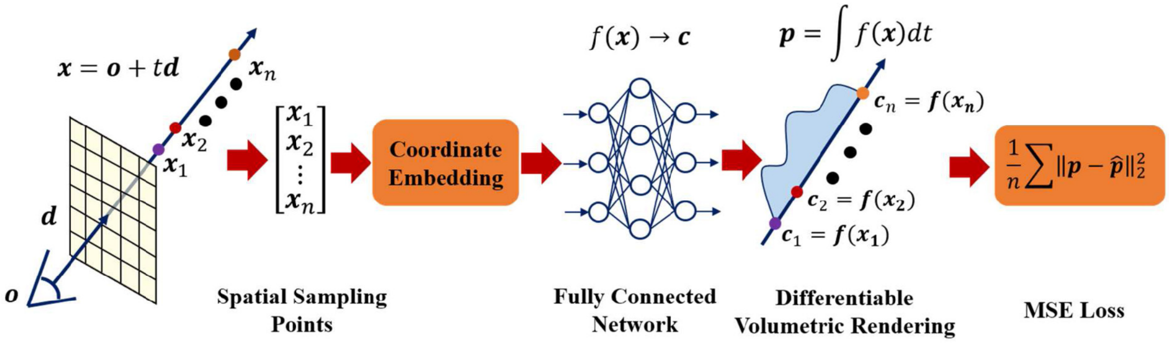


Figure 3.9: Schematic of the Neural Volume Reconstruction Technique. Taken from [8].

3.5 Physics-Informed Neural Networks

Physics-informed neural networks (PINNs) are a subclass of neural implicit representation methods where the volume is approximated by a neural network that outputs state variables of the flow. Physics-based priors using the governing flow equations help enforce physical representations as demonstrated by [149, 51]. While PINNs can solve many different dynamic systems, PINNs have proven to be a viable tool for solving inverse problems for flows by employing the Navier-Stokes equations. It should be noted that traditional CFD solvers have difficulty solving these inverse problems [53]. PINNs, like neural implicit representation methods, have an input of spatial-temporal coordinates (x, y, z, t) that are used to compute the relevant flow field variables (e.g. velocity $\mathbf{u}$, pressure p, density ρ, *etc*). Therefore, the neural network of the PINN approximates the flow field as a continuous function. The governing equations are a function of partial derivatives of the flow parameters. The partial derivatives of the input are calculated using automatic differentiation [150], which calculates the true derivate rather

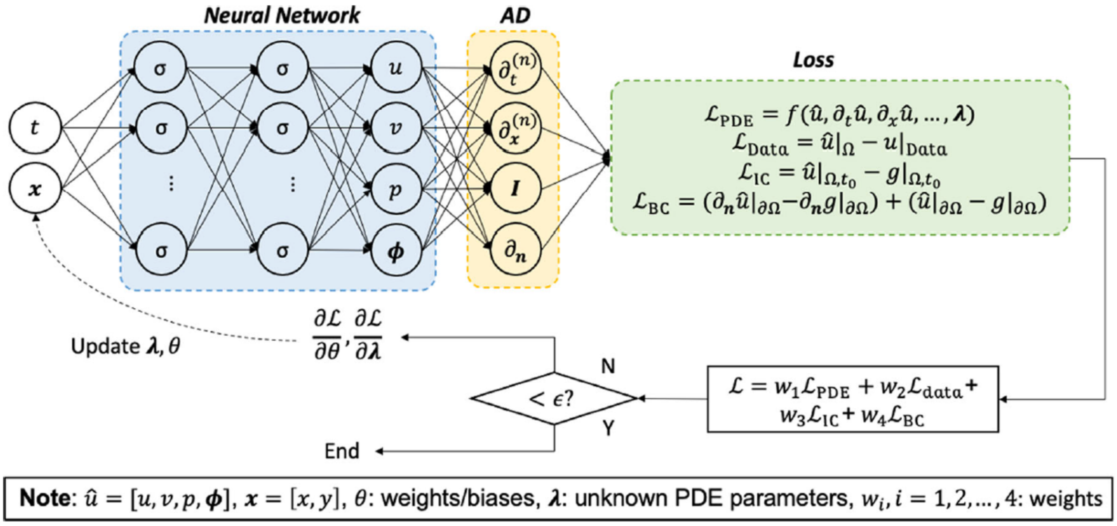

Neural Network
AD
Loss
t
x
σ
u
v
p
ϕ
$\partial_t^{(n)}$
$\partial_x^{(n)}$
I
∂_n
$\mathcal{L}_{\text{PDE}} = f(\hat{u}, \partial_t\hat{u}, \partial_x\hat{u}, \ldots, \boldsymbol{\lambda})$
$\mathcal{L}_{\text{Data}} = \hat{u}|_{\Omega} - u|_{\text{Data}}$
$\mathcal{L}_{\text{IC}} = \hat{u}|_{\Omega,t_0} - g|_{\Omega,t_0}$
$\mathcal{L}_{\text{BC}} = (\partial_n\hat{u}|_{\partial\Omega} - \partial_n g|_{\partial\Omega}) + (\hat{u}|_{\partial\Omega} - g|_{\partial\Omega})$
Update $\boldsymbol{\lambda}, \theta$
$\frac{\partial\mathcal{L}}{\partial\theta}, \frac{\partial\mathcal{L}}{\partial\boldsymbol{\lambda}}$
N
$< \epsilon?$
Y
End
$\mathcal{L} = w_1\mathcal{L}_{\text{PDE}} + w_2\mathcal{L}_{\text{data}} + w_3\mathcal{L}_{\text{IC}} + w_4\mathcal{L}_{\text{BC}}$
Note: $\hat{u} = [u, v, p, \boldsymbol{\phi}]$, $\boldsymbol{x} = [x, y]$, θ: weights/biases, $\boldsymbol{\lambda}$: unknown PDE parameters, $w_i, i = 1,2,\ldots,4$: weights

Figure 3.10: A diagram of a physics-informed neural network. For this figure $\hat{u}$ denotes the multi-physics solution $[u,v,p,\phi]$. AD stands for auto-differentiation, which solves for the partial derivatives indicated by the subscripts x and t.

than a numerical approximation. Therefore, the network can be constrained through losses that include the physics-based equations ($\mathcal{L}_{\text{PDE}}$), boundary conditions ($\mathcal{L}_{\text{BC}}$), initial conditions ($\mathcal{L}_{\text{IC}}$), and measurements ($\mathcal{L}_{\text{Data}}$). The general form of the loss functions for PINNs are given through [53]

$$\mathcal{L}_{\text{PDE}} = f(\hat{\mathbf{u}}, \hat{\mathbf{u}}_t, \hat{\mathbf{u}}_x, \ldots, \hat{\mathbf{u}}_{xx}) \tag{3.19}$$

$$\mathcal{L}_{\text{BC}} = (\hat{\mathbf{u}}_n|_{\text{BC}} - g_n|_{\text{BC}}) + (\hat{\mathbf{u}}|_{\text{BC}} - g|_{\text{BC}}) \tag{3.20}$$

$$\mathcal{L}_{\text{IC}} = (\hat{\mathbf{u}}|_{t_0} - g|_{t_0}) \tag{3.21}$$

where $\hat{\mathbf{u}}$ is the flow field variables, g is the known values of flow field variables for both boundary and initial conditions, and the subscripts of x and t denote the partial derivative of x and t, respectively. A general schematic of PINNs for flow predictions is shown in Figure 3.10. For fluid mechanics problems, different approximations of the Navier-Stokes equations can be used for the PDEs depending on the flow assumptions. Cai *et al.* [53] indicates $\boldsymbol{\lambda}$ as a learned parameter of the PDE, which is useful if properties of the flown are not known like kinematic viscosity. Data loss can also be included if other data points are measured or known, such as pressure measurements. The total loss utilized to update the neural network is the weighted sum of the individual losses. The weights help bias the solution towards the more important loss function. An introduction to the machine learning fundamentals and PINNs is provided by Kollmannsberger *et al.* [151].

$$e_1 = c_t + (\mathbf{u} \cdot \nabla)c - \mathrm{Pe}^{-1}(\nabla^2 c) \tag{3.22}$$

$$e_2 = \mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \mathrm{Re}^{-1}(\nabla^2 \mathbf{u}) \tag{3.23}$$

$$e_3 = \nabla \cdot \mathbf{u}, \tag{3.24}$$

where Pe is the Péclet number, and Re is the Reynolds number. These are the non-dimensionalized forms of the Navier-Stokes equations, although the dimensionalized equations can be utilized as others have demonstrated [55, 152, 153].

Cuomo *et al.* [154] has written a comprehensive review article on PINNs, focusing on the latest improvements, current unknowns, potential future directions, and the current pitfalls of PINNs. While PINNs are known well for fluid applications, they have been employed for various other applications due to their ability to solve forward and inverse problems with the same structure. Some examples are integro-differential equations [155], fractional equations [155], surface PDEs [156], heat transfer problems [157], point-source PDE [158], and stochastic differential equations [159]. One important advancement made for PINNs is the ability to calculate the prediction uncertainty [160, 153], where the Bayesian framework has been applied (B-PINNs) [161]. As was reviewed by Cai [53] and Cuomo [154], there has been recent improvements to the MLP architecture multi-fidelity framework [162, 54], activation functions [152, 163], dynamic weighting of the loss functions [164], PDEs enforced as hard constraints [165], and CNN-based network design [166]. Changes in the network architecture affect the accuracy of the approximation, convergence speed, and stability of the solution. Additionally, Wang *et al* has conducted theoretical investigations to explain when and why PINNs fail [167] and the eigenvector bias caused by Fourier feature networks [168]. Mishra *et* *al.* [169] investigates the generalization error of PINNs for inverse problems. Most of these improvements and results are application agnostic and, therefore, can be incorporated into other PINN implementations to improve their model.

3.5.1 Projection-based data assimilation with PINNs

One recent advancement in combining flow measurements and physics priors with PINNs was conducted by Molnar *et al.* [9], where a B-PINN approach was used to solve for the concentrations of a synthetic passive scalar and the flow field simultaneously using only the concentration measurements. Figure 3.11 shows the architecture of the B-PINN. Molnar's approach extends the method first proposed by Raissi *et al.* [52], where this work utilized the data loss function as shown in Figure 3.11. For 2D images with an infinitely thin volume, the concentration estimate is captured by the pixels of the image. However, extending to 3D fields requires traditional tomography methods to calculate an estimate. Using traditional methods can incorporate errors from reconstruction algorithms that can corrupt the flow field prediction. Molnar *et al.* [170] improves this by utilizing a projection loss that approximates the imaging model for LoS data, similar to what is employed in the NeRF community. Their method utilized a neural network with ten hidden layers containing 50 neurons per layer, with each node using the swish activation function. A comparison between the two data loss functions was investigated [9]. A three-phase training model was identified, which was composed of i) measurement-dominated, ii) hybrid, and iii) fully-coupled regimes. The measurement-dominated regime is the first phase of training, where concentrations are updated to match the LoS projections, but the velocity field loss increases. The hybrid section is the location where the physics loss levels and starts to diminish. Finally, the fully-coupled regime has both loss terms approach one another and continue to diminish. Molnar [9] showed the importance of a hyperparameter in balancing the physics and measurement loss functions. Noisy data produced semi-convergence; however, a stopping criterion was developed to locate the bottom of the semi-convergence accurately.

Recently, Molnar *et al.* has extended the projection loss-based PINNs to background-oriented Schlieren (BOS) measurements of supersonic-flows [170, 56]. Instead of predicting the concentration of a passive scalar in the flow field, density is added as an output, which is

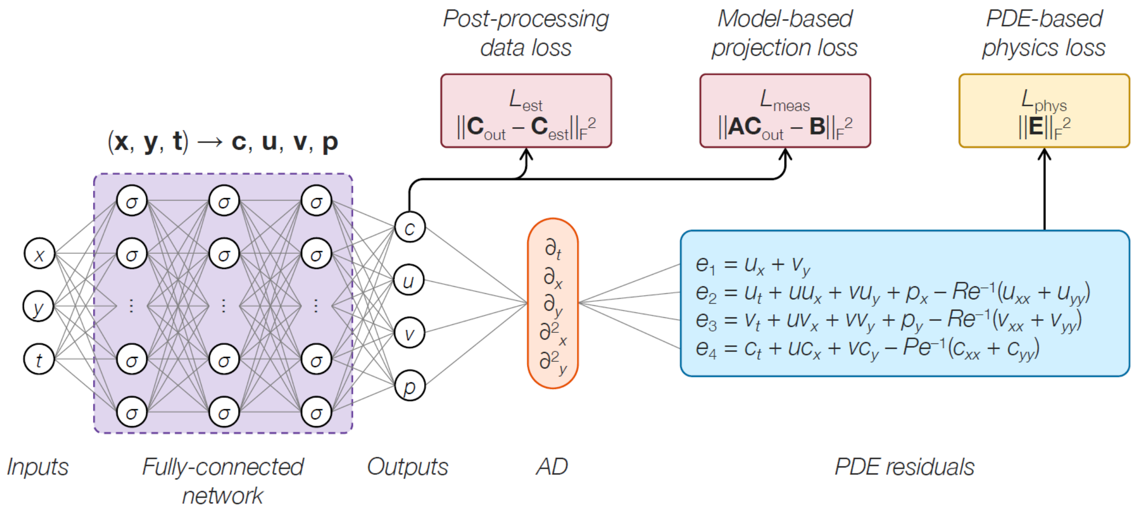

Post-processing data loss
Model-based projection loss
PDE-based physics loss
L_{est} $||\mathbf{C}_{out} - \mathbf{C}_{est}||_F^2$
L_{meas} $||\mathbf{AC}_{out} - \mathbf{B}||_F^2$
L_{phys} $||\mathbf{E}||_F^2$
(x, y, t) → c, u, v, p
x
y
t
σ
c
u
v
p
∂_t ∂_x ∂_y ∂^2_x ∂^2_y
$e_1 = u_x + v_y$
$e_2 = u_t + uu_x + vu_y + p_x - Re^{-1}(u_{xx} + u_{yy})$
$e_3 = v_t + uv_x + vv_y + p_y - Re^{-1}(v_{xx} + v_{yy})$
$e_4 = c_t + uc_x + vc_y - Pe^{-1}(c_{xx} + c_{yy})$
Inputs
Fully-connected network
Outputs
AD
PDE residuals

Figure 3.11: PINN architecture used for 2D flow field tomography as employed by Molnar *et al.* [9]. The data loss can be evaluated using two ways i) comparing the PINN output to an estimated concentration that is calculated through traditional tomography methods or ii) a projection based data loss similar to the update step of tomography methods. Figure adapted from [9].

used to calculate a predicted BOS measurement. Thus far, this method has been implemented for a single LoS projection of BOS measurement. Therefore, to generate a predicted BOS measurement from the density field, Abel inversion is utilized to solve the path integrals for the deflections. A unified BOS operator [28] is utilized to compare the predicted and measured projections. The work and equations shown thus far have utilized the incompressible Navier-Stokes equations. However, with this work, the compressible, steady, 2D Euler equations are utilized to account for density being spatially dependent. The 2D Euler equations also include the energy equation. Therefore, the total energy is another output. The network approximates $F(x, r) = (\rho, u, v, \mathrm{E})$, where F is the continuous function approximated by the neural network. Pressure is calculated from the polytropic equation of state to close the Euler equations. They added the inlet conditions to stabilize the training. Boundary conditions can help provide additional information on the flow to improve the PINN prediction, where inlet conditions are typically known for experimental data sets [56]. Other useful boundary conditions could be pressure taps or free stream conditions. The total loss is calculated using

$$\mathcal{L}_{total} = \omega_{meas}\mathcal{L}_{meas} + \omega_{phys}\mathcal{L}_{phys} + \omega_{in}\mathcal{L}_{in} \tag{3.25}$$

where ω is the weighting parameter for each loss. The optimal weights were found as $\omega_{meas}/\omega_{phys} = 10$ and $\omega_{in}/\omega_{phys} = 100$. For this work, Molnar employed a PINN with 5 hidden layers composed of 50 neurons with swish activation. He proved that with the inclusion of physics priors, PINNs yielded more accurate density approximations compared to conventional BOS algorithms [170].

3.5.2 PTV Data Assimilation with PINNs

Another essential implementation of PINNs for flow measurements is in the particle tracking velocimetry (PTV) field. PTV algorithms determine the velocities of tracer particles that seed the flow using tracking algorithms. For 3D PTV algorithms, the general framework starts with imaging particles in a flow that is illuminated by a laser with at least two different simultaneous views. Once the images are captured, a triangulation algorithm is employed to locate where each particle is in 3D space for each time step. Then, particles are tracked between frames using a tracking algorithm to determine velocities [171]. Since uncertainty accumulates during each step, regularization techniques are required to produce realistic flow fields.

One of the first to demonstrate PINNs for data assimilation of PTV measurements by introducing physics priors was Di Carlo *et al.* [172]. This method was developed for sparse 2D data. The data loss is the MSE between the measured and predicted velocities at particle locations. The physics loss was simplified by only utilizing the divergence of the velocity field, which assumes divergence-free, isotropic, and homogeneous flow. In addition to this, the subgrid self-similarity constraint was an added physics prior. This constraint enforces the scales below the elementary scale of the energy cascade to be driven by the power law ($S_2(l) = \gamma l^2$). Fourier features were employed in this work before connecting the inputs to the neural network with $L = 1$. The MLP consisted of 5 hidden layers with 256 nodes each, with hyperbolic tangent as the activation function.

Di Leoni *et al.* [55] in concurrent work employed a similar method for post-processing 3D PTV measurements. Leoni applied the same MSE loss for the data loss as Carlo; however, a more traditional physics loss is utilized using the incompressible continuity and momentum

Navier-Stokes equations similar to Raissi [52] discussed earlier. Unlike most of the PINNs discussed thus far, the authors apply normalization layers before and after the MLP as introduced by LeCun [108]. These normalization layers balance the magnitudes of both inputs and outputs. This helps keep the outputs of the MLP to be between (-1, 1) for efficient back-propagation. The input is normalized through a min-max function,

$$x = 2\frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} - 1 \tag{3.26}$$

where x_{max} and x_{min} are the maximum and minimum values of the volume domain of interest. The velocity and pressure are normalized with a z-score function

$$\hat{\mathbf{u}} = \sigma_{\hat{\mathbf{u}}}\hat{\mathbf{u}} + \mu_{\hat{\mathbf{u}}} \tag{3.27}$$

where $\mu_{\hat{\mathbf{u}}}$ and $\sigma_{\hat{\mathbf{u}}}$ are the mean and standard deviation of each flow field variable. The MLP for this work had eight hidden layers with 200 nodes, each with an activation function of hyperbolic tangent. For a case with a large number of time steps, multiple neural networks were trained for nonoverlapping subsets of the data. Increasing the number of time steps requires the complexity of the flow field approximation for unsteady flows, therefore requiring either a larger MLP or multiple smaller MLPs.

Finally, a technique for post-processing PTV data allowing for non-ideal tracer particles is stochastic particle advection velocity (SPAV), where Zhuo *et al.* [171] employed PINNs for the flow field predictions. Like other methods developed by Grauer's group, the technique learned through data, physics, and boundary condition losses. A deeper network was implemented, where the MLP was composed of 15 layers with 250 nodes each. As found by Cuomo [154], most methods discussed in this section utilize an Adam optimizer method for training the networks.

3.6 Chapter Summary

The goal of this chapter was to first discuss the basics of deep learning neural networks before covering deep learning methods utilized for tomography and computer vision. Deep neural networks are a subset of machine learning that consists of a network of neurons. Deep neural networks are considered a universal function approximator. They have shown remarkable ability in data processing, detection, classification, and production of probabilistic models. Most of the current demonstrated DNN methods for tomography use MLPs to relate input projections to an output reconstruction, where the method trains with a ground truth volume. Most methods use realistic models generated from CFD to train the model to resemble the experimental configuration that the model will apply. Most of the work for tomography thus far has used CNNs, where the output is the QoI for each volume element. The explicit representation leads to discretization errors, as found earlier for conventional tomography algorithms, and limits the applicability of the trained network.

In contrast to neural explicit representation, implicit representations have been recently implemented in the computer vision community. Neural implicit representations have been the most successful DNN framework for novel view synthesis, where the volume is approximated as a continuous function of spatial coordinates. One method that emerged from the implicit methods is neural radiance fields (NeRF). NeRF relates spatial and angular information to color and optical density within the volume to generate exceptional image rendering quality. NeRF integrated Fourier feature mapping, called positional encoding, that helps capture the high-frequency content of the light-field. The NeRF idea was implemented in two recent flow diagnostic techniques called Physics-Informed NeRF and Neural Volume Reconstruction Technique.

Another active area of research is physics-informed neural networks (PINNs). PINNs are a subclass of neural implicit volume representation that uses physics-based PDE equations to enforce physics priors on the solution. A PINN is a functional representation of the salient flow field variables. One of the most promising applications of PINNs is combining flow volumetric imaging measurements and physics priors, where PINNs can solve inverse flow problems. Recent work by Grauer's group [9, 56, 170] has attempted to combine tomography and PINNs to allow LoS projection data to assist data assimilation using both 2D passive scalars and BOS measurements. The advancements found in PINNs are generally application-agnostic and should be considered for tomography.

Chapter 4

Scalar Field Reconstructions

One of the most well-known 3D flow diagnostic techniques is PIV, where the volume of interest is composed of tiny particles entrained in the flow, which scatter light from a laser or other light sources. The points of light are reconstructed in 3D space, typically with MART, and used to determine the velocity of the flow. The point sources provide different challenges compared to the focus of this next chapter, scalar fields. Scalar fields are a continuous distribution of a QoI in space-time. In flow diagnostics, scalar fields include flow/fluid properties like temperature, velocity, pressure, density, and mass concentrations. One research area that is interested in measuring scalar fields is combustion. Various techniques have been developed to investigate the processes and products that occur during combustion. An established and frequently used method for gathering flow measurements involves employing a probe or rake system to acquire individual or multiple data points within the flow. However, these systems disturb the flow patterns and would have to withstand the harsh environments of combustion or high-speed flows.

The emergence of optical diagnostic techniques, including high-energy lasers and high-resolution cameras, overcomes many of the challenges of probes and intrusive measurements. Lasers and optical sensors can provide 1D, 2D, and 3D measurements of the QoI. Coherent anti-Stokes Raman scattering (CARS) [173] and laser-induced grating spectroscopy [174] provide 0D and 1D measurements of the temperature and concentration fields. Laser absorption spectroscopy can provide LoS measurements to acquire measurements of temperature, pressure, and species concentrations [23], where the LoS measurements can be used to get 2D and 3D measurements. As pulsed lasers become available, laser-based diagnostics have advanced to enable 2D [175] and 3D [1] measurements at increased temporal resolutions, particularly for turbulence analysis. Some measurements include planar laser-induced fluorescence and planar laser-induced incandescence, which provide multiple 2D scalar field measurements.

While lasers can make measurements through scattering, absorption, or fluorescence, combustion and flames have other emission modalities to provide self-illumination for optical sensors to capture. These self-emissions can provide information on the reactions, temperatures, species concentration, and more. An example of self-emission in combustion can be observed in a hydrocarbon flame, initially manifesting in the form of the color as a blue base transitioning into bright yellow/red regions as the flame advances and disperses away from the reaction zone. As elaborated in the tomography section, optical sensors can capture projections from various angles, either simultaneously or through scanning, enabling the generation of 3D measurements of these emissions. This chapter will cover four main scalar field measurements: i) chemiluminescence, ii) incandescence, iii) laser-induced incandescence, and iv) laser-induced fluorescence.

4.0.1 Emission modalities

To understand how these measurements are conducted, one must also understand the underlying physics of the emission process. Light emissions from flames and reaction zones of gases occur due to molecules releasing photons as they go from a higher energy level to a lower level, where the energy of the photon (wavelength) equals the change in energy levels. The energy levels of molecules have quantized states that include vibrational, rotational, and electronic. Since the energy levels are quantized, there are only a discrete number of wavelengths that can be emitted from a flame, each corresponding to a different transition. The emitted photons from the flame can be classified by the mechanism that caused the molecule to enter an excited state. Optical methods can single out a mechanism of interest through optical filters to isolate a particular portion of the spectrum corresponding to the specific mechanism that produced it. One mechanism of the transition of energy states is through random collisions, where higher energy molecules can pass energy to lower energy molecules. If the collisions are caused by the thermal motion of molecules, the mechanism is called thermal radiation or *incandescence*.

The incandescence of gas molecules is rarely used for combustion measurements. Another source of incandescence is solid-phase radiation through soot, one of the more common measurements. Solid-phase incandescence is emitted across a broad spectrum approximated by Planck's distribution [176]. If non-thermal mechanisms cause the radiation, it is referred to as luminescence. The most common luminescence for combustion diagnostics is *chemiluminescence*, where radiation is emitted by molecules from chemical reactions in the flame, where the molecules are generated at higher, non-stable energy states. The molecules or atoms in these states emit photons to transition to a more stable state. Finally, the last radiation type discussed here is *fluorescence*, where the molecules reach a higher energy state through absorption of other photons [1]. The molecules then transition to the same or alternative lower energy states, emitting photons.

4.0.2 Chapter road map

The following sections of this chapter provide an overview of four of the most common scalar field measurements for emission diagnostics, including chemiluminescence, incandescence, laser-induced incandescence, and laser-induced fluorescence. Emission diagnostics is a vast area of research, as indicated by the review article by Grauer *et al.* [1]; therefore, this chapter will focus on a small portion of the literature. This section will focus on past and present work that conducts mainly 3D scalar field emission measurements with emphasis on the tomography method and number of cameras utilized. Discussion on potential challenges for each technique will be covered, motivating the current work. The chapter will summarize the present techniques and discuss the motivation for this work as we shift to describing methodology and synthetic/experimental investigations. This review of scalar field measurement follows the descriptions and vocabulary that was presented by Grauer [1].

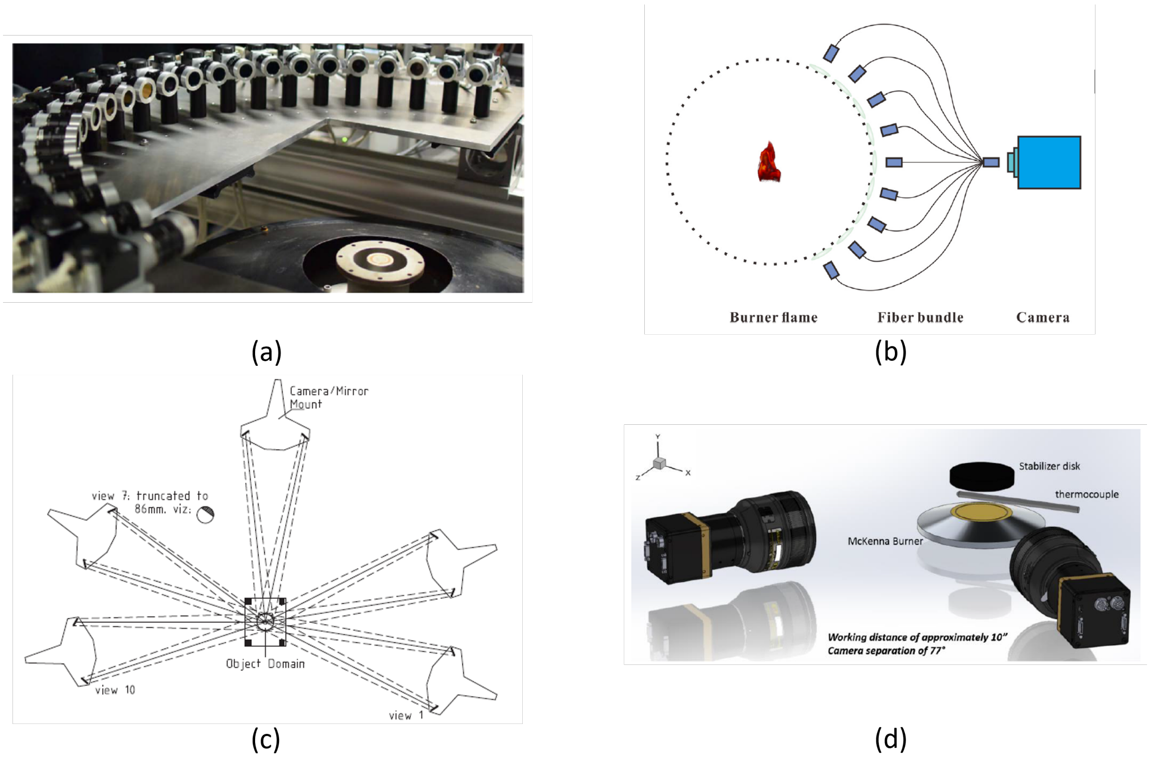
4.1 Chemiluminescence

During combustion, chemiluminescence involves intermediate reactions where molecules are initially formed in an excited state, which then relaxes, releasing photons. Chemiluminescence is typically located in narrow spectral bands corresponding to discrete energy state transitions.

Since these molecules are intermediate chemical reaction products, they provide information on the reaction zone. Optical chemiluminescence measurements are LoS integrated through the flame. The most common radicals are the OH* and CH*, where the asterisks (*) indicate the molecule is at a higher energy state. Tomography chemiluminescence measurements are generally called computed tomography chemiluminescence (CTC), where Hertz and Faris [177] first demonstrated CTC with 2D tomography. The implementation of 3D CTC methods has accelerated with the emergence of lower-cost optical sensors and improvements to computing power. Obertacke *et al.* [178] introduced a volumetric CTC method that captured different radicals (OH*, NH*, CN*, CH*, and C_2*). Grauer *et al.* [1] provides a detailed description of chemiluminescence, covering the spectral structure and reaction pathways to produce radiation.

Floyd and colleagues [179, 12] were among the first groups to systematically study 3D CTC. These first studies employed a relaxed ART method. Using the ART method, Floyd and Kempf [179] studied the resolution effects and number of views for CTC. A matrix burner of 21 laminar flames was constructed to evaluate the reconstruction quality of the flames using ten views from five cameras. They captured images of the burner with the matrix in different orientations to the cameras. They obtained ten views from five cameras using mirrors to split the sensor in half. In addition to the matrix burner, synthetic phantoms were used to investigate the effect of noise, where a signal-to-noise ratio of 20dB was the limit. Floyd *et al.* [12] explores the limits on spatial resolution using cosine phantoms. The results showed the complex relationship of 3D spatial resolution for tomography on image parameters and orientation relative to the flow features.

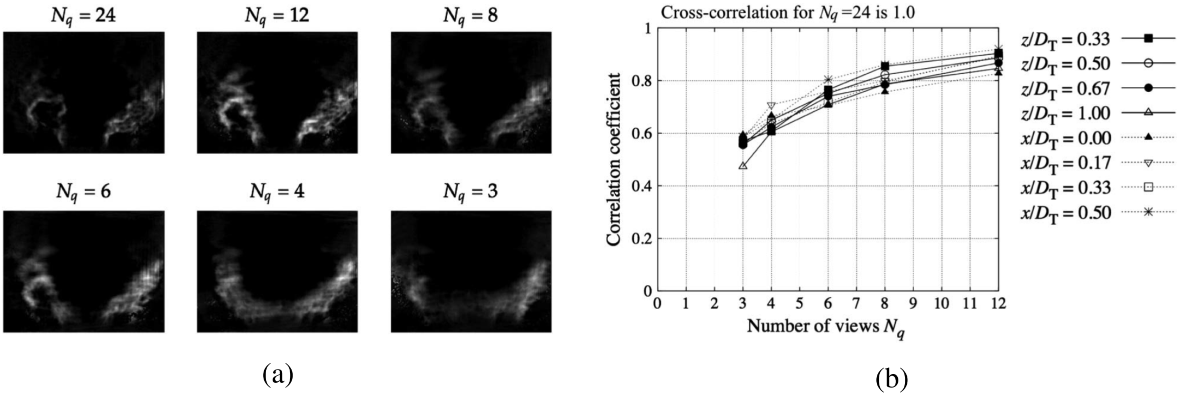
Mohri *et al.* [10] conducted an empirical experiment with highly turbulent swirl flame using CH_4 as fuel to understand the influence of the number of cameras and camera spacing on reconstruction quality. An array of 24 CCD cameras, shown in Figure 4.1(a), were arranged around the swirl burner to provide the ability to take subsets of cameras for comparisons and baselines. For the tomography algorithm, Mohri [10] employed ART to generate 3D reconstructions. Slices of the 3D reconstruction utilizing different numbers of projections are presented in Figure 4.2a. Reducing the number of views causes reduced reconstruction fidelity with artifacts appearing in the slices. Since there was no ground truth baseline, the baseline



(a) (b)

(c) (d)

Figure 4.1: Various optical systems for capturing multiple simultaneous projections for CTC: (a) using multiple cameras [10], (b) fiber-optic bundles that relay images to a single camera [11], (c) mirror and prisms to redirect a projection to a portion of the sensor [12], and (d) plenoptic cameras [13].



(a) (b)

Figure 4.2: (a) Slices through the center of the 3D reconstructed chemiluminescence corresponding to reconstructions with varying number of cameras, and (b) correlation coefficient of the 3D reconstructions for different number of projections compared to 24 projections. Taken from [10].

volume was a reconstruction using the maximum number of cameras. Figure 4.2b shows an appreciable difference between the 24-camera and 12-camera reconstructions ($\sim$ 85% correlation). Mohri [10] attempted reconstructions with as little as three cameras, where only a 60% correlation was found compared to the 24-camera reconstruction. The results discussed thus far used evenly-spaced views; however, Mohri extended the investigation to unequal spacing. It was found that unequal spacing had little effect on the reconstruction quality compared to the same number of evenly spaced cameras.

Thus far, all CTC papers presented have used ART. However, many algorithms have been demonstrated for CTC. Liu *et al.* [180] explored four different tomographic algorithms for CTC reconstructions, including ART, Landweber, Tikhonov Regularization, and Truncated Singular Value Decomposition algorithms using synthetic DNS-generated data. The projection data were specifically developed to test limited angular view configurations. Jin *et al* [181] demonstrated a hybrid ART algorithm with TV regularization for CTC and compared the reconstruction quality to ART, showing TV helped smooth reconstruction while preserving sharp gradients. In an earlier paper, Ishino and Ohiwa [182] employed an MLEM algorithm for reconstruction using as many as 40 cameras. Recently, Unterberger [38] introduced a genetic algorithm method (ERT) for CTC with results compared to ART. Another iteration of ART was developed for CTC by Zhang *et al.* [27]. Zhang [27] imposes non-negativity constraints on the ART equations termed the non-negative algebraic reconstruction technique (NNART). In addition to these methods, SMART has been employed for CTC using two plenoptic cameras [13]. Finally, deep learning methods have been developed by Jin *et al.* [45] and Zhang [8] that used an explicit and implicit neural representation, respectively. These examples show the vast number of tomography methods utilized for CTC. In general, ART-based methods were the most common.

With the magnitude of different tomography methods, there are different instrumentation configurations for CTC, as some examples are shown in Figure 4.1. One of the most common methods is distributing multiple cameras with lenses and filters oriented around a semicircle, as shown in Figure 4.1(a) [10]. However, a 24-camera array system of scientific-grade cameras can be very costly and lead to triggering complexities. Therefore, other methods have been developed to decrease the number of sensors for tomography. One method utilized by Liu [11] splits a single sensor into multiple projections through fiber optic cables with front lenses. As discussed earlier, Floyd [12] employed mirrors and prisms to generate two perspectives on a single sensor. The last example is the plenoptic camera by George [13], where a plenoptic camera captures angular information of light (perspectives) using a microlens array that is placed between the primary lens and the sensor. For plenoptic 1.0, the microlens array captures images of the primary lens such that the pixels behind each microlens correspond to a point on the primary lens. A more detailed description of instrumentation is discussed by Grauer *et. al.* [1].

4.2 Incandescence

Incandescence is radiation that is emitted from atoms/molecules caused by thermal energy stored in molecules as vibrational and rotational motion. In general, incandescence is a broadband radiation for opaque and translucent materials. Therefore, incandescence is typically generated from particle-laden flows for combustion and high-temperature applications. The radiation from soot can account for a significant amount of heat transfer in combustion devices, especially for large chambers [1]. As explained by Modest [61], the perceived color of incandescence from an object or particle-laden flow is dependent on the temperature. The relationship between the spectral intensity ($E_{b,\lambda}$) and temperature is specified through Planck's blackbody equation

$$E_{b,\lambda}(T) = \frac{2hc^2}{\lambda^5}\left[\exp\left(\frac{hc}{\lambda k_B T}\right) - 1\right]^{-1} \tag{4.1}$$

where h is Planck's constant, c is the speed of light, and k_B is Boltzmann's constant. Since the spectrum depends on temperature, incandescence can produce pyrometry measurements. An approximation of Eqn. 4.1 that allows for the inversion of the equation for easier calculation of temperature is the Wien Distribution

$$E_{b,\lambda}(T) \simeq \frac{2hc^2}{\lambda^5}\left[\exp\left(-\frac{hc}{\lambda k_B T}\right)\right], \tag{4.2}$$

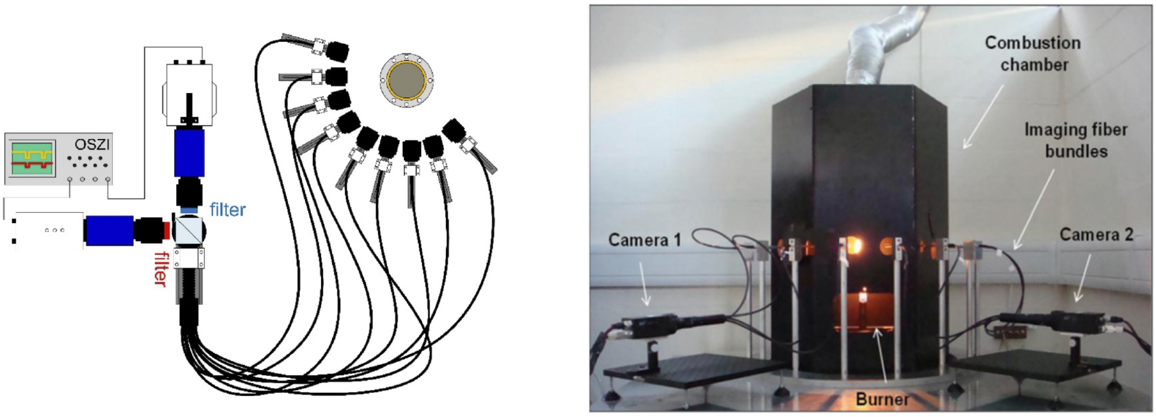
where the assumption best approximates Planck's equation for $\lambda T \leq 2898\mu$m K [183]. This approximation holds for wavelengths that are shorter than the spectral intensity peak at a given temperature. Real materials are non-ideal emitters and absorbers of radiation. Therefore, the absorption coefficient (κ_λ) or emissivity (ε_λ) is multiplied by the blackbody equation to account for the difference between the actual material response relative to the blackbody response. The coefficient depends on the molecular structure and other properties, and emissivity is a function of λ. If ε_λ is constant across λ of a material, the material is considered a graybody.

In addition to temperature, soot-volume-fraction can be calculated from incandescence [184]. Soot is a common byproduct of hydrocarbon combustion and, therefore, provides a passive source of incandescence measurements. 3D incandescence measurements are a crucial way of understanding the generation, agglomeration, and burning of hydrocarbon fuels, indicating the efficiency and stability of the combustion [1]. Hence, incandescence measurements are common in the combustion community, especially for hydrocarbons.

The chemiluminescence signal is located in a thin region in the reaction zone, as indicated in the previous section. Therefore, Chemiluminescence measurements can assume negligible absorption. Absorption can become an issue for most incandescence measurements, where the LoS through soot or other emitters is much longer. Incandescence measurements are mainly affected by self-absorption and out-scattering. Due to these effects, the imaging model (Eqn. 2.3) cannot assume negligible light loss. The imaging model is

$$I_\lambda \propto \int_0^\infty I'_\lambda|\Psi^{-1}(l)| \times \exp\left[-\int_{l^*}^\infty \beta_\lambda|\Psi^{-1}(l^*)|dl^*\right] dl. \tag{4.3}$$

Eqn. 4.3 accounts for the extinction through a coefficient (β_λ) that is a function of absorption and scattering coefficients (κ_λ and σ_λ), where $\beta_\lambda = \kappa_\lambda + \sigma_\lambda$. κ_λ and σ_λ are proportional to the soot-volume-fraction and inversely proportional to wavelength. Several methods have investigated the influence of both on incandescence measurements. The review article by Grauer [1] discusses both κ_λ and σ_λ in more detail.



(a)

(b)

Figure 4.3: Schematic of different experimental apparatuses utilized for incandescence measurements by (a) Yu [14] and (b) Hossain [15].

One of the most common techniques for determining the temperature of soot is dual-wavelength (two-color) pyrometry [176]. One demonstration of 3D reconstruction of temperature distribution of soot using two-color pyrometry was conducted by Yu *et al.* [14]. Yu utilized a fiber optic camera system surrounding a laboratory flame with nine fiber optic cables, as shown in Figure 4.3a. The fiber optic cables feed different projections to a beam splitter, where two camera sensors capture the light from each fiber optic. The cameras are outfitted with a narrow band filter to sample unique segments of the spectra, with one capturing 425 nm and the other 600 nm. Similar to chemiluminescence, ART-based tomography methods are common for 3D reconstruction, and Yu employed ART to reconstruct the flame at each wavelength. The temperature is then calculated using the ratio between the intensity for each voxel [176].

It is best to capture monochromatic emission measurements since chromatic nonlinearities arise due to absorption, scattering, and optical parameters like quantum efficiency. However, many works have used a standard color camera that utilizes a Bayer-style filter to capture three broadband signals corresponding to red (R), blue (B), and green (G) channels. For more information on spectral imaging techniques, the author suggests the review article by Hagen [185]. The broader spectral channels of these cameras encompass more significant uncertainty due to the spectral and temperature dependence of the imaging model. Some limitations can be removed if the radiation can be assumed to be gray, although this is not the case for most applications.

Hossain *et al.* [15] demonstrated the use of eight fiber optic bundles, where they are divided and captured by two RGB color CCD as depicted in Figure 4.3b. Each spectral channel is reconstructed independently using the SART algorithm, where the reconstructions for each channel are used to calculate temperature and emissivity at each voxel. Zhuo *et al.* [186] implemented an eight-view optical system using eight RGB detectors surrounding a pulverized coal flame in a boiler furnace. Zhuo employs a modified Tikhonov regularization method to reconstruct temperature directly from the images, where the temperature can be related to the image intensities through a linear matrix equation ($I = A_I T$). Huang *et al.* [187] employed a color light field camera that captured RGB images of a particle-laden flame. They introduced a new reconstruction method, which is a hybrid least-square QR decomposition with a conjugate gradient algorithm to reconstruct the flame's temperature, absorption, and scattering coefficients. Additionally, thermometry was conducted of a sooting flame by Hall *et al.* [188], where they employed the Fourier transform tomography method for capturing temperature distributions. As demonstrated with chemiluminescence, a wide variety of tomography methods are employed for volumetric incandescence measurements. A deep learning approach was employed by Ren *et. al.* [189, 4] to capture 2D and 3D temperature and molecular mole fractions. The machine learning method used a supervised approach by training an MLP with the spectral emission from each pixel as the input relating to temperature and concentrations to each voxel.

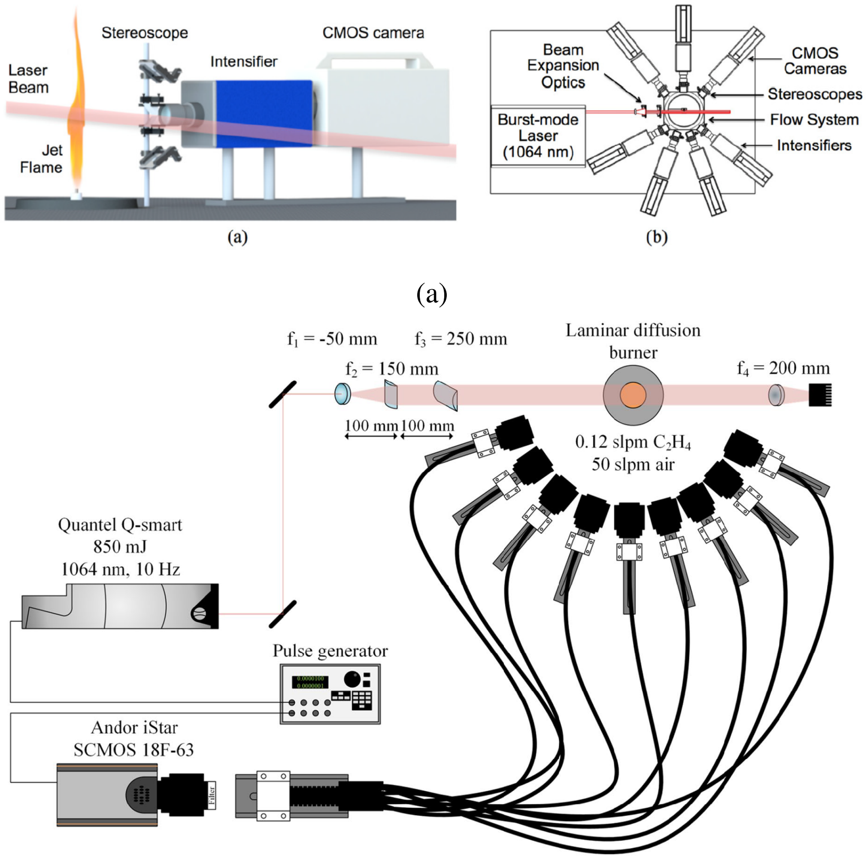
The main contributor of incandescence from flames is generated through soot. The concentration of soot affects the amount of radiance emitted while also being coupled to the amount of self-absorption and scattering that occurs. Soot measurement can be captured to understand the flame's temperature distribution and volume concentrations throughout the flame as shown by Eqn 4.3. Soot measurements provide knowledge of the soot production mechanisms so that soot can be controlled [1]. Early work in soot-volume-fraction measurements was demonstrated for 2D distributions [184, 190].

Those early works had to make some limiting assumptions, including optically thin and negligible absorption and scattering. Later developments were demonstrated to account for self-absorption for the reconstruction algorithm [191, 192, 193]. Some of the earlier correction techniques were developed by Freeman and Katz [194] and Hall and Bonczyk [188]. Liu [193] found that self-absorption was negligible for temperature estimates of sooting flames up to 10 mm in diameter and that scattering was insignificant in all cases. One recent development is hyperspectral imaging techniques, which can spectrally resolve the absorption and emission incandescence to simultaneously assess temperature, emissivity, and soot-volume fraction as demonstrated by Liu [195].

4.3 Laser-induced Incandescence

Thus far, the discussion has focused on naturally occurring incandescence due to the temperature of solid aggregates in flames. Heating particles with a characterized laser pulse can provide additional information through incandescence. After being heated, the particles will cool to thermal equilibrium. Cameras can capture the radiation during the cooling process, where this process is known as laser-induced incandescence (LII). Michelsen *et al.* [196] provides a very detailed review article of LII. Similar to incandescence measurements, particle temperature, volume fraction, size, and distribution can be captured from LII. Unlike natural incandescence, the particles and surroundings do not have to be at higher temperatures to image the particles since the laser heats the particles of interest. LII can provide quantitative measurements of the formation, distribution, and oxidation of particles in several different environments, including flames, exhaust, atmosphere, and liquids [1]. Measurements are similar to incandescence, with thermal radiation dominating the emission spectrum as described by Eqn. 4.1 and Eqn. 4.2. LII was first demonstrated as a point measurement, which was followed by planar and linear measurements [197] and recently volumetric [16, 17, 18]. Volumetric LII (VLII) has emerged recently due to the development of high-power, high-repetition-rate lasers. LII measures the cooling rate of the heated particles, which requires known heat transfer properties to get accurate quantities from the particles. The governing heat transfer equations are covered in detail by the two detailed review articles [1, 196], where readers are encouraged to review the basics of LII. This section will focus on the volumetric measurements that have been conducted.

The first demonstration of VLII was introduced by Meyer *et al.* [16]. The focus of the work by Meyer was to capture time-resolved soot distributions in an unsteady, turbulent ethylene/air flame. Due to the steep costs of high-speed cameras and intensifiers, VLII measurements were



(a)

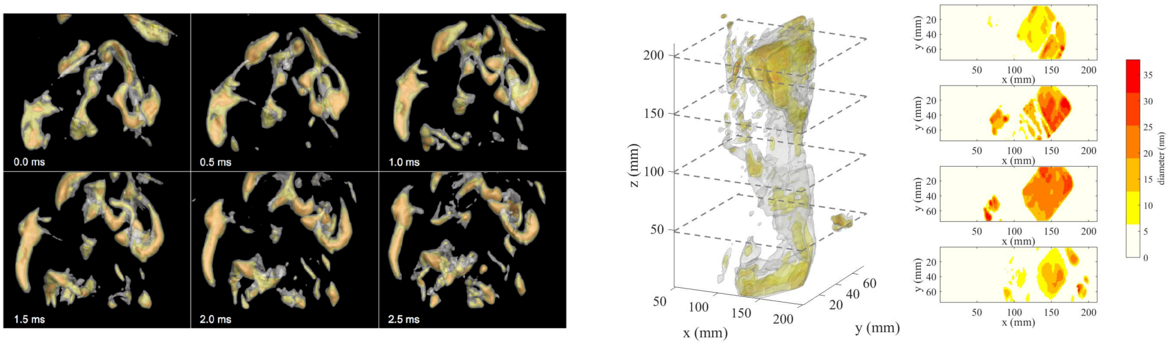
(b)

Figure 4.4: Schematic of different experimental apparatuses utilized for VLII measurements by (a) Meyer [16] and (b) Bauer [17]. Adapted from [16, 17].

captured using seven stereoscopic high-speed CMOS cameras to produce 14 views as shown in Figure 4.4a. The soot was illuminated and heated with a burst-mode Nd:YAG laser with pulse widths of about 10 nanoseconds. The excitation wavelength was the fundamental output of a Nd:YAG laser at 1024 nm. The laser pulse energy density was approximately 0.1 J/cm^2. The volumetric reconstruction was conducted using SMART. For the demonstration, only 8 of the 14 views were used for reconstruction, which is a limitation of the LaVision DaVis software that was utilized for this work. A critical assumption for the tomographic reconstruction was that extinction was negligible, which is not generally valid for sooting flames [1]. Extinction results from the volume becoming optically thick due to soot volume fraction. In addition to extinction, the method does not account for beam steering effects produced by the thermo-chemical gradients present in flames. Reconstructions were generated using different sets of the 14 views. Figure 4.5a shows the evolution of the soot reconstructions with time. The measurements by Meyer were qualitative, where work is required to account for LoS extinction, scattering, and beam profilometry to calculate quantitative measurements [16].

Quantitative time-resolved VLII diagnostics of a turbulent flame was first conducted by Hall *et al.* [18], where models were used to calculate soot particle sizes from volumetric intensities and cooling rates. Volumetric imaging was conducted with three HPV-X2 cameras that were configured with stereoscopes (similar to Meyer [16]), providing six views. The volume was illuminated using a pulse-burst Nd:YAG laser (1064 nm) with a pulse duration of 12 ns. The volume was 10 mm deep by 48 mm tall. Images were captured at a rate of 10 MHz. A 50 nm wide band-pass filter centered at 600 nm eliminated contaminating spectra from the surroundings. Again, LaVision Davis was used to reconstruct the volume using SMART. The particle diameter of the soot was calculated using the method by Cenker *et al.* [198], which relates particle size to the cooling rate. An average diameter of soot is calculated for each voxel location as shown in Figure 4.5b. Like Meyer, Hall neglected laser beam attenuation, LoS extinction, and scattering, which can significantly affect the reconstructions. Future works should study these effects, or the reconstruction method should account for them.

The most recent VLII study was conducted by Bauer *et al.* [17], where VLII measurements were conducted of a laminar diffusion flame using nine views. The volumetric imaging was captured using nine fiber bundles, each with a primary lens at different locations. Each fiber bundle produces an image on a single SCMOS sensor, as displayed in Figure 4.4b. With the use of a lower-speed laser (10Hz), quasi-time-resolved measurements were conducted by conducting several runs, where each experimental run had a different delay between the laser pulse and the camera trigger. The quasi-time-resolved technique is applicable because the flame is steady. The volumetric measurements were reconstructed using the ART algorithm. Bauer [17] explained that ART is more suitable for continuous scalar fields as compared to MART. Since the flame is steady, monochromatic measurements were conducted during separate runs. Bauer [17] calculated soot particle size distributions using the same technique as Hall [18]. Tomographic two-color pyrometry (incandescence measurements) was used to estimate the flame temperature. In addition, beam proliferation was conducted to account for beam



(a)

(b)

Figure 4.5: (a) Isocontours of the VLII reconstructions at time steps of 0.5 ms [16], and (b) Isocontours and slices of the soot particle diameter calculated by Hall [18]. Figures were adapted from [16, 17].

steering through the volume. Extinction was also calculated using LoS measurements through the flame with a broadband light source and a spectrometer.

4.4 Laser-induced Fluorescence

Fluorescence is the emission of a photon (energy) from an excited molecule that transitions to a lower energy state. The molecule enters the excited state through the absorption of an external photon. Laser-induced fluorescence (LIF), as the name indicates, generates the external photon by a laser, causing the molecule to enter an excited state. In fluorescence measurements, the wavelength of light is different from the laser light, meaning the molecule/atom returns to a different ground state than its initial state. Fluorescence has the same energetic structure and emission transitions as chemiluminescence. The population density is typically unknown for chemiluminescence, and the signal of interest overlaps with broadband emissions from other molecules. Laser excitation provides the ability to target specific species. However, laser excitation has challenges, including collisional quenching, predissociation, and photoionization, which requires LIF measurements to be accompanied by spectroscopic models to account for multiple energy levels and transfer mechanisms [1]. Phosphorescence operates similarly to fluorescence, the difference being that fluorescence emission occurs rapidly compared to phosphorescence. Phosphorescence requires a change in spin before emitting photons, whereas fluorescence has a constant spin. Several books discuss the physics behind LIF, where the author recommends the texts by Herzberg [199] and Hanson *et al.* [200] for detailed descriptions.

LIF has emerged as a powerful tool for flow visualization and quantitative measurements. In the field of combustion, LIF has been proven to determine species concentrations [201], temperature [202], and mixture fraction [203]. The diagnostic has been proven for quasi-point [204], 1D, planar [205], and volumetric [206]. Quasi-3D measurements were conducted using planar LIF (PLIF) measurements using scanning methods [207]. However, these measurements limit temporal resolution due to the scanning time. The signal generated by LII results from thermal radiation, where higher temperatures increase signal intensities. Alternatively, the LIF signal is a function of the local thermochemical state in combination with laser power and wavelength. The local thermochemical state is ultimately the energy state population distribution of the target species [1]. Since current lasers can tune a narrow wavelength, they can be used to target the particular absorption spectra of molecules in combustion, including OH [205] and CH [208] or flow tracers like acetone [209], rhodamine dye [210], or nitric oxide [211]. The speed and sensitivity of fluorescence have made it an appealing flow diagnostic for combustion and other flows. The rest of this section will focus on the recent literature on volumetric LIF (VLIF), which was first demonstrated in 2015 by Wu *et al.* [206].

Similar to LII, the emergence of VLIF in 2015 parallels the improvements of pulse-burst lasers, where high-powered lasers are necessary for adequate energy densities. Wu *et al.* [206] demonstrated VLIF for a non-reacting turbulent jet seeded with an iodine tracer. The LIF signal was generated by a Nd:YAG laser at a wavelength of 532 nm that excited iodine vapor. The signal generated from LIF was captured by five CMOS cameras distributed in a circular plane perpendicular to the flow direction. Each camera attenuated the laser wavelength using a notch filter. Wu utilized a hybrid ART method with TV regularization as described by Cai [26]. They accounted for absorption using a multi-step reconstruction, where species concentration is reconstructed in layers. During the process, the absorption of the signal from the previous

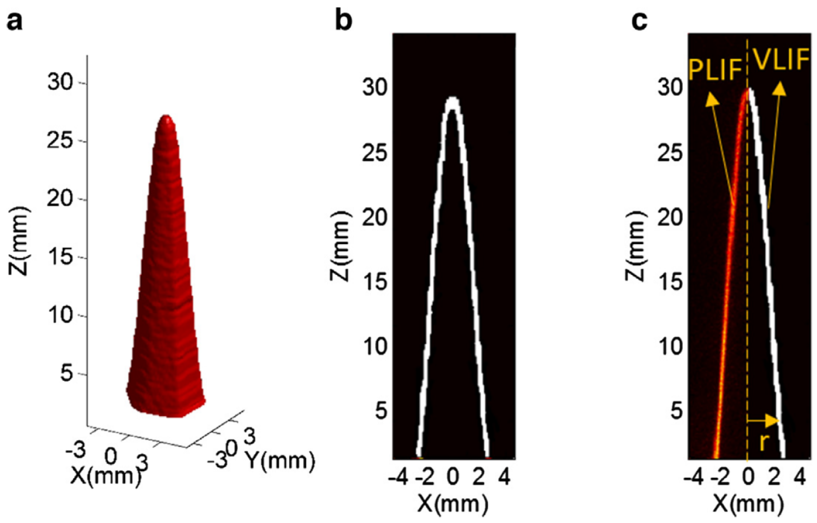
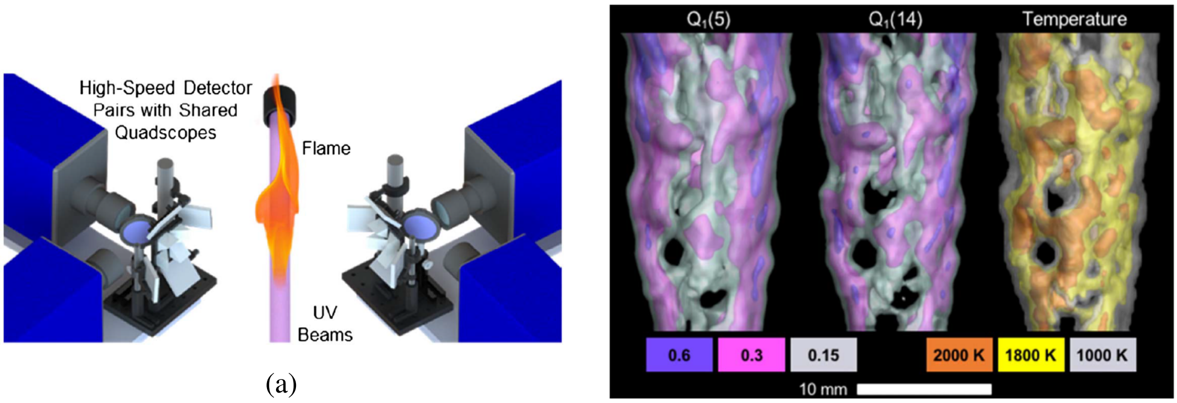


Figure 4.6: (a) Reconstruction of the conical flame, (b) central slice through the VLIF volume, and (c) side-by-side comparison of the PLIF and VLIF measurement. Adapted from [19].

layer is determined iteratively. The method starts with the layers closest to the camera, which will have little to no absorption.

Ma's group then applied the same setup as Wu for combustion measurements of CH VLIF in a conical flame [19]. Ma compared the VLIF measurements directly to PLIF measurements, showing close correspondence between the PLIF measurement and a slice of the LIF volume. Note that the PLIF laser pulse was 45 ns behind the VLIF laser. A comparison of the VLIF result and the PLIF results is shown in Figure 4.6. Ma further applied the measurement and comparison to a turbulent jet generating similar results from the conical flame [212, 208]. These techniques implemented the same hybrid ART tomographic reconstruction technique from [26]. Ma [19] discussed that the reconstruction method uses a screening process, similar to masking, that sets all voxels to zero along the LoS of a pixel that is zero. The authors would like to note that the PLIF measurements were made span-wise with the laser volume [206, 212, 19]; therefore, it has not been verified that VLIF provides additional resolution in the normal direction [1].

While most of the early work for VLIF was conducted by Ma's group, other groups quickly adopted the technique, including Meyer [21, 20] and Dreizler [213]. Additionally, the work by Li *et al.* [213] evaluated the effect of optical configuration (number of cameras and angular orientation) on reconstruction accuracy. Li employed a stereoscopic camera configuration with



(a)

(b)

Figure 4.7: (a) Two-color quadscope for VLIF measurements by Halls [20], and (b) reconstructions of $Q_1(5)$, $Q_1(14)$, and temperature distribution from VLIF. Figures adapted from [20].

four intensified cameras capturing eight views. The reconstruction was conducted with SMART using LaVision DaVis. The laser volume was systematically changed the volume depths to 2, 4, 10, and 30 mm. It was concluded that constricting the reconstruction volume decreases the complexity of the unknown intensity field, producing higher-quality reconstructions. The larger laser volume produced a blurring effect on the edge of the sharp OH intensity boundary captured by the PLIF.

In a paper by Yu *et al.* [214], an investigation of the empirical resolution of 3D emission tomography methods was conducted with a glass container filled with fluorescent dye. The glass container produces a sharp, circular boundary for the fluorescent dye. The boundary is a top hat style intensity distribution at the boundary. Reconstructions were generated using eight camera views, which we captured by moving a single camera to different angles around a semi-circle since the QoI is static. Yu [214] applied the ART tomography algorithm. A spatial resolution limit can be calculated from the blurred edge of the reconstruction compared to the expected top hat function. Calibration issues caused by the diffraction at the boundary of the glass produced several reconstruction artifacts.

In addition to qualitative concentration measurements, Halls *et al.* [20] captured temperature measurements of a laminar and turbulent H_2/air flame using OH VLIF. To conduct the quantitative measurement, Halls employed two quadscopes that captured two wavelengths using a beamsplitter and two sensors per quadscope, as shown in Figure 4.7a. Therefore, eight views were captured for each spectral band. Halls employed DaVis's SMART algorithm to reconstruct the two wavelengths. Temperature was calculated from the two wavelengths, where each wavelength corresponds to the two OH transitions ($Q_1(5)$ and $Q_1(14)$). The reconstructions of each OH transition and the resulting temperature distribution are presented in Figure 4.7b.

In addition to these snap-shot techniques, high-repetition-rate lasers and high-speed cameras have allowed the demonstration of time-resolved VLIF [215, 21, 216]. There are several methods for capturing multiple time-resolved views, including using seven high-speed cameras [215], four quad scopes generating 16 views [21], and a single camera with nine fiber-bundles [216]. All of these papers employed ART [215, 216] or SMART [21, 217] for volume reconstruction. Most reconstructions for LIF, as with SMART used by Halls, assume negligible beam steering, in-scattering, and absorption. In work by Halls [21], they were able to track the evolution of coherent structures within a turbulent flame using intermediates of formaldehyde (CHO) and hydrocarbon (PAH). Not only have the time-resolved VLIF measurements been demonstrated for combustion and gases, but Gomez demonstrated VLIF for reconstructing the shape of liquid droplets [217]. VLIF is a rapidly expanding flow diagnostic technique because of its versatility for combustion and other fluid dynamics analysis.

4.5 Chapter Summary

This chapter highlights some of the most widely used volumetric scalar field measurements in flow diagnostics, with active developments are still underway. The four measurements covered are chemiluminescence, incandescence, laser-induced incandescence, and laser-induced fluorescence. While these four measurements were chosen to demonstrate the applicability and execution of tomography methods for scalar field measurements, the author would like to emphasize that there are several other 3D scalar field measurements used in fluid dynamics analysis. These measurements are generally conducted for combustion diagnostics for quantities that include emission mechanisms, flame structure, reaction rates, mass concentrations, soot-volume-fraction, temperature, and turbulence. These methods have produced impressive results for combustion and flow diagnostics. However, extending measurements to 3D can provide some challenges, including quenching, self-absorption, scattering, and beam steering that occur along the LoS. It was shown that most 3D measurements have employed an ART-based method that assumes an optically thin volume. These lead to limitations for the current tomography methods, where accounting for these processes caused a redesign of the tomography algorithms.

Past work applied a wide range of image sensors and optical systems to capture a collection of 2D projections from different angles, where differences in the number and angle between projections were noted. The 3D measurements also utilized a wide array of volume sizes and resolutions. All of these factors can increase complexity and have an effect on the reconstruction quality, which some of the traditional tomography methods have characterized. This work proposes an implicit neural network approach to tomography that has only had limited concurrent work demonstrating CTC capabilities. Therefore, the literature should include an in-depth investigation of imaging configuration and network hyperparameters on the reconstruction quality for 3D flow diagnostics. The following sections cover a detailed description of the NeRF-based tomography method developed during this work and investigations using synthetic and experimental datasets.

Chapter 5

FluidNeRF Implementation

The following chapter will provide a detailed explanation of the neural implicit presentation tomography method based on NeRF, which was developed for flow diagnostics and is called FluidNeRF. FluidNeRF is a tomography method capable of handling spatial (instantaneous) or spatial-temporal coordinates (time-resolved). First, the instantaneous or static version will be described, followed by the time-resolved version. The static FluidNeRF section will describe the components of the reconstruction algorithm, and the time-resolved FluidNeRF section will include the differences that were added to handle the time input. While the current method was developed for flow diagnostics, the technique could be applied to other emission-based tomography problems.

5.1 Static FluidNeRF

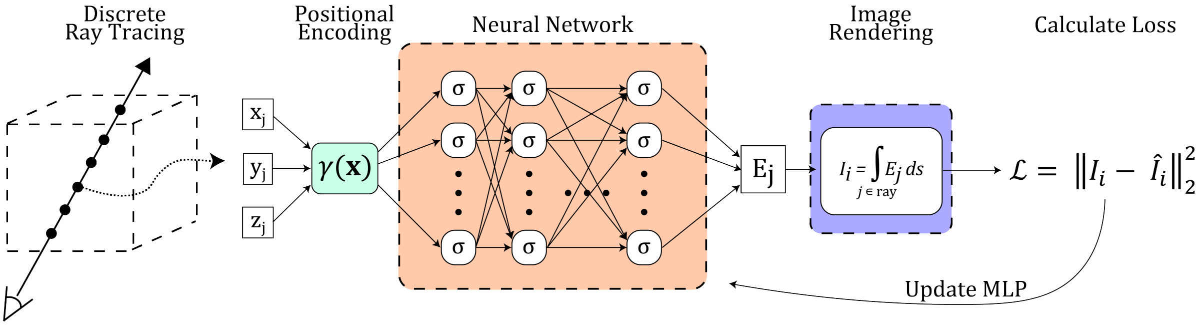


Figure 5.1: Schematic of the static FluidNeRF algorithm that uses a neural network to represent the volume.

The FluidNeRF reconstruction model is shown in Figure 5.1. First, a light ray is traced through the volume corresponding to an image pixel of the measurements, where a user-defined number of query points are taken along the ray. Each point has a corresponding 3D object space coordinate (x_j, y_j, z_j) passed through positional encoding (γ) to increase the dimensionality before connecting to the MLP. The activation function of the MLP nodes is indicated by σ. The MLP outputs the approximated volume luminance (E_j) at the queried 3D coordinate. Once all query points along the ray are interrogated, a pixel value is rendered by integrating the volume luminance along the ray. The measurement loss (L) that is used to update the MLP, where L is the mean-square-error between the rendered pixel value (I_i) and the measured pixel value ($\hat{I}_i$). The measurement loss is accumulated for a batch of rays before being used to update the MLP.

5.1.1 Discrete ray tracing

The volume sampling method is crucial in rendering images of the reconstructed volume, thus impacting the updated reconstruction quality. A multi-resolution technique is used to improve the sampling. First, FluidNeRF produces a semi-random set of samples along rays cast through the volume. Then, a second set of samples is calculated using information from the intensity values from the first set of samples. This two-stage sampling method is referred to as Hierarchical sampling [6]. This method helps increase samples in areas of interest while limiting samples to locations with constant or no intensity. Hierarchical sampling can be necessary in reducing the total number of samples required to capture the important features of a volume when rendering the projection, compared to a uniform or stratified sampling method.

This work uses a simple pinhole camera model to trace rays through the volume. Each ray has an origin (r_o), corresponding to the pinhole location in space relative to the volume, and a direction (r_d) given by a unit vector. Discrete samples along the ray can be generated using $r_o + sr_d$, where s is the distance of the sample point to the camera pinhole. The algorithm begins by dividing the ray into N_c evenly spaced regions between the user-defined near-field and far-field positions on the ray (s_{near}, s_{far}) to produce an initial set of sample regions. Then, a query point is randomly selected in each region as defined by Eqn. 5.1 [6]

$$s_i \sim U[s_{near} + \frac{i-1}{N_c}(s_{far} - s_{near}), s_{near} + \frac{i}{N_c}(s_{far} - s_{near})]. \tag{5.1}$$

This sampling technique produces semi-uniform sampling along the ray since a single sample is taken in each evenly-spaced region.

After the first set of semi-random locations have been interrogated, the intensity values for each location are used to produce another set of samples. This technique uses the spatial gradient of intensity along the ray. The gradient is found numerically using $dE = \frac{E_{j+1} - E_j}{s_{j+1} - s_j}$. While this method calculates an approximate gradient, it can include automatic differentiation to find the exact gradient at each point. This should be invested in the future. The feature sampling should be based on the amplitude of the gradient to position samples near boundaries or other features. Thus, the new samples are based on the absolute value of the gradient. The amplitude of the gradients is then normalized along the ray $\left(\hat{dE}_j = \frac{dE_j}{\sum_j^{N_c - 1} dE_j}\right)$, producing a piece-wise constant probability density function (PDF). Then, N_f feature samples are determined using the PDF. Thus, the final image will be rendered using all $(N_{tot} = N_c + N_f)$ samples.

5.1.2 Positional encoding

With the sampling along the ray completed, the 3D spatial coordinates are individually encoded to a higher dimension using Fourier feature mapping, as demonstrated by Tancik *et al.* [218]. Neural networks are notorious for converging to lower-order solutions, leading to poor performance in representing the high-frequency content in a volume. Thus, positional encoding uses a variant of Fourier feature mapping as shown in Eqn. 5.2

$$\gamma(\tau) = [\tau, \sin(2^0 \pi \tau), \cos(2^0 \pi \tau), \ldots, \sin(2^{L-1} \pi \tau), \cos(2^{L-1} \pi \tau)], \tag{5.2}$$

where L is an integer value that determines the dimensionality of the encoding. A higher value of L produces more values and increases the dimension of the input array. Not only does it improve the performance in capturing high-frequency information, but it has also been shown to improve training speed. Thus, positional encoding is incorporated into FluidNeRF.

5.1.3 Image rendering

Another critical step for an accurate reconstruction is the image rendering, as the imaging model should approximate the measurements. In flow diagnostics, the volumes of interest can be approximated as optically thin, therefore an emission-based imaging model can be applied. FluidNeRF incorporates a modified emission-based image rendering, where the standard model is described by [219]. For this model, the light only originates from the volume bounded by s_{near} and s_{far} for a particular ray. The light that reaches a pixel is calculated using Eqn. 5.3

$$I_i = \int_{s_{near}}^{s_{far}} \Omega(s)A(s)E(s)ds, \tag{5.3}$$

where I_i is the intensity of the ith pixel, Ω is the solid angle as a function of s, and A is the area of the ith pixel in object space as a function s. Ω is a measure of the spatial angle subtended by the lens or aperture of the camera, typically expressed in steradians (sr). Eqn. 5.3 needs to be estimated using the samples described previously. A simple schematic of the ray tracing and rendering is shown in Figure 5.2. In our method, this integral is calculated using the midpoint quadrature method as presented in Eqn. 5.4.

$$I_i = \sum_{j=1}^{N-1} \Omega(\bar{s}_j)A(\bar{s}_j)\Delta\bar{s}_j \frac{(E_{j+1} + E_j)}{2} \tag{5.4}$$

where $\bar{s}_j$ is the midpoint between the j and $j+1$ sample, and $\Delta\bar{s}_j$ is the distance between midpoints $(\bar{s}_{j+1} - \bar{s}_j)$. The area of a pixel can be found using the local magnification $\left(\frac{p_i}{-M_j}\right)$. For the quadrature method, Ω is the normalized solid angle relative to the focal plane that is calculated via $(s_o/s_j)^2$, where s_o is the object distance of the main lens. Note that the emission-based rendering model is only valid for optically thin volumes; however, the model allows for other imaging techniques to be incorporated easily.

5.1.4 Updating the network

The previous section conducts the back-projection step to render image pixels corresponding to the measurement images. The MLP that approximates the volume can be updated using a loss

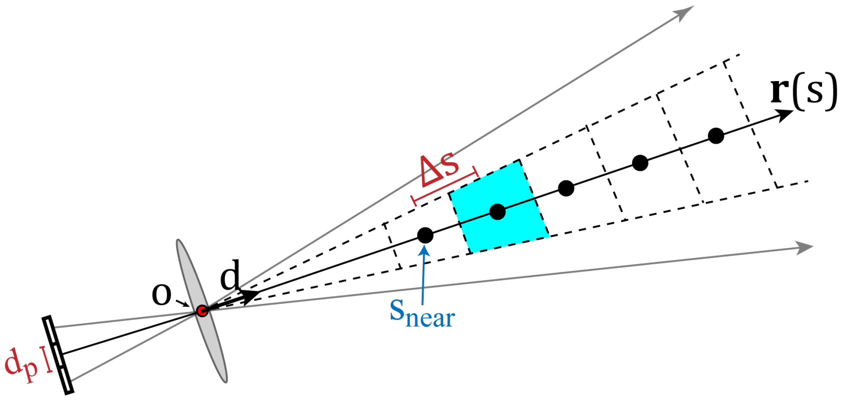


Figure 5.2: Schematic of the ray tracing and image rendering method. The solid rays correspond to rays cast from pixels and the dashed lines correspond to boundaries between the pixels. The query points are indicated by the points along the ray.

function that quantifies the difference between the predicted projection (I_i) and the measured projection ($\hat{I}_i$). In this work, the mean square error (MSE) loss function is used as shown in Eqn. 5.5

$$\mathcal{L} = \frac{1}{N_{pix}} ||I_i - \hat{I}_i||_2^2, \tag{5.5}$$

where N_{pix} is the number of pixels used per iteration. MSE is a standard loss equation that has been used for other NeRF variants [6, 8]. However, including other loss functions has been demonstrated to improve performance [7].

5.1.5 Implementation

FluidNeRF used for this work is implemented in Python utilizing the Tensorflow 2.9 library. The default hyperparameters of the method for the synthetic experiments are presented in Table 5.1 and used throughout this work unless otherwise indicated. The network size is a user-defined hyperparameter that is set at run-time. The neural network's activation function (σ) is the rectified linear unit (ReLU), which can also be set during run-time. An iteration of FluidNeRF is completed once all images have been considered, where a random batch of pixels from each image is queried. For this work, a user-specified 1024 pixels from each image are selected during each iteration for training. Once an iteration is finished, the network is updated using the Adam optimizer with an initial learning rate of 5×10^{-4} that decays exponentially to 5×10^{-5} after 25000 iterations. The Adam optimizer hyperparameters are kept as default as previously reported by [6]. A convergence criterion is used to stop training when the $\mathcal{L}$ varies less than 2% across a 2000-iteration stretch with a 200-iteration running average, similar to the implementation by Molnar *et al.* [170]. FluidNeRF is computed on a single NVIDIA Tesla T4 GPU from the Auburn University Easley Cluster.

Table 5.1: Default hyperparameters of static FluidNeRF unless specified otherwise.

Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
Depth	8	N_c	128	Optimizer	Adam
Height	256	N_f	128	Initial learning rate	0.0005
L	8	$N_{pix/img}$	1024	Decay	Exponential
# of cameras	15	Activation	ReLU		

5.2 Time-Resolved FluidNeRF

The NeRF concept allows the tomography method to be adaptable to alternative inputs and outputs since neural networks are universal function approximators. Currently, the instantaneous FluidNeRF method can reconstruct a volume under the assumption that the projections are not time-dependent. This assumption holds when the images are captured simultaneously or depict a static volume. This assumption is the same as traditional tomography methods described in Chapter 2, where the reconstructions are only a function of spatial coordinates. However, cameras can capture multiple snapshots across time of a dynamic scene, where the scene changes smoothly and continuously with time. Therefore, time can be included as an input to FluidNeRF, allowing the reconstruction technique to account for temporal information, similar to PINNs. The addition of time to the FluidNeRF model will be called time-resolved FluidNeRF (TR-FluidNeRF).

A schematic of the TR-FluidNeRF is displayed in Figure 5.1. The main idea behind TR-FluidNeRF is that a neural network is trained to approximate the volume of interest using captured images of the scene from different viewpoints, where the difference between the captured and predicted perspective updates the neural network. Therefore, the algorithm starts by attempting to render a volume projection from the same viewpoint as a captured projection. Rendering is conducted by tracing rays through the volume that corresponds to pixels of the

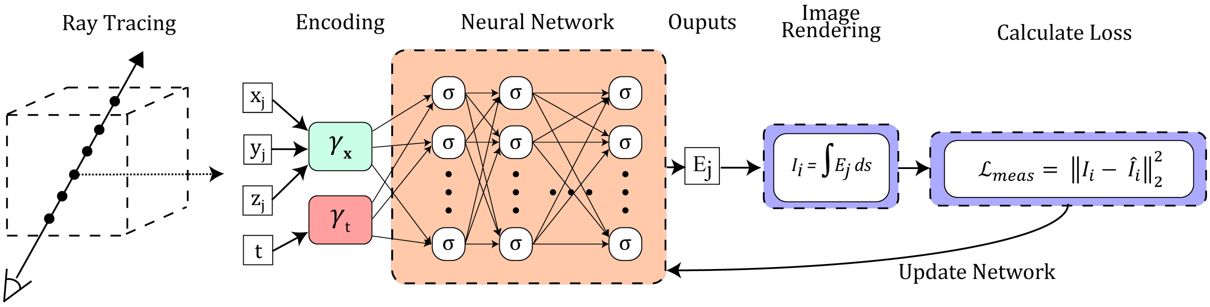


Figure 5.3: Schematic of the time-resolved FluidNeRF algorithm that uses a neural network to approximate the volume in space and time.

captured images for one of the measured time points. A semi-random set of spatial-temporal coordinates is generated along a ray to approximate a ray through space. The spatial and temporal points are first encoded using γ_x and γ_t, respectively. Similar to positional encoding discussed by Mildenhall [6], temporal encoding can improve the performance of capturing the higher-order components of the reconstruction with time. Once through encoding, the input array is passed into a feed-forward neural network that calculates the intensity density (E_j) for each spatial-temporal point along the ray. Like static FluidNeRF, the ray is rendered for one instance in time using a discrete emission-based image rendering technique. Once the pixel is rendered, the loss ($\mathcal{L}_{meas}$) is calculated by comparing the rendered and captured pixels. A backpropagation algorithm is utilized to update the weights of the neural network, equivalent to updating the volume approximation. An iteration accounts for all images at a specific time step.

A vital step for TR-FluidNeRF was adding a normalization step to the temporal coordinate before going through temporal encoding (Eqn. 5.2). With the smaller time step size of high-speed, time-resolved measurements, the temporal encoding function collapses the inputs since Fourier Feature mapping has a constant period of 2π. The collapsing of the temporal coordinate caused the neural network to have difficulty training and converging to an averaged volume across time steps. Therefore, the normalization creates a more general network, where L of Eqn. 5.2 should be the only thing required to change between different datasets or time step sizes. For our method, the normalized time coordinate is calculated by $t' = t/\Delta t$. An alternative method for countering this effect is adjusting the period of the sinusoidal function using a variable as discussed by Tancik *et. al.* [218], although this requires an additional hyperparameter that will need to be adjusted between spatial and temporal coordinates and for each dataset. Therefore, the normalization of the time coordinate reduces the number of hyperparameters.

Figure 5.4 presents an example of a fully connected neural network model that is employed for TR-FluidNeRF. We use a residual connection for the TR-FluidNeRF network structure that combines the inputs with the output of the middle layer. Each layer has a constant height (H) throughout the network. While the example model has eight layers, we will change the depth and height in the following results section. TR-FluidNeRF can utilize any of the activation functions as demonstrated in Section 3.1.1. Table 5.2 provides the default hyperparameters for TR-FluidNeRF reconstruction that will be used for the results unless specified otherwise. L_x and L_t are the encoding hyperparameters for spatial and temporal coordinates, respectively. The current version of TR-FluidNeRF is implemented using TensorFlow 2.9 in Python 3.9. TR-FluidNeRF employs the Adam optimizer built into Tensorflow to train the neural network. A convergence criterion is used to stop training when the $\mathcal{L}$ varies less than 2% across a 2000-iteration window with a 200-iteration running average. For the results, TR-FluidNeRF was conducted on a single NVIDIA Tesla T4. The current algorithm design provides the ability to reconstruct with and without temporal information, which is determined at run-time through user input. Where TR-FluidNeRF will be compared to the instantaneous FluidNeRF method. For the instantaneous FluidNeRF implementation, the network had eight layers with 256 nodes each, and all other parameters were kept constant.

Table 5.2: Default hyperparameters of TR-FluidNeRF.

Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
Depth	10	N_c	64	Optimizer	Adam
Height	200	N_f	64	Initial learning rate	0.0005
L_x	8	$BatchSize(N_{pix/img})$	1024	Decay	Exponential
L_t	0	# of cameras	15	Activation	Swish

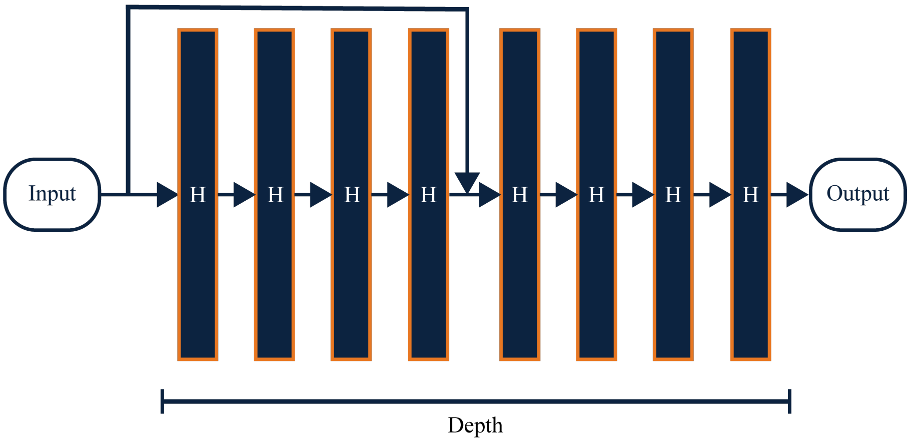


Figure 5.4: An example of the FluidNeRF network structure with 8 layers with a residual connection after four layers.

5.3 Chapter Summary

This chapter covers the NeRF-based technique that was developed during this work, named FluidNeRF. FluidNeRF is based on the idea of neural implicit representation, where a neural network approximates a continuous intensity field as a function of spatial coordinates. Two versions were developed during this work: i) instantaneous FluidNeRF without temporal dependence and ii) time-resolved FluidNeRF, including the time component of the measurements. An iteration of FluidNeRF commences by tracing rays through the volume of interest corresponding to pixels of the captured images of the volume. A semi-random sampling coordinates along each ray is generated for input into the neural network. Positional encoding is conducted to increase the dimensionality of the input coordinate vector and improve the performance of FluidNeRF. Time-resolved FluidNeRF also has an input of time, which goes through a separate encoding function. Those vectors are passed through the neural network to generate a volumetric intensity of each input coordinate. Another set of samples are generated using the intensity distribution along the ray. This method is referred to as hierarchical sampling. The new samples are passed through the neural network for additional volumetric intensity points along the ray. A predicted intensity captured by the pixel is generated through an emission-based image rendering technique, which is compared to the captured pixel value through a loss function. The loss between the predicted and captured pixel values is used to update the network. The update step of FluidNeRF is similar to that of traditional tomography methods, although FluidNeRF has the flexibility to include additional constraints. The following chapters cover the synthetic and experimental evaluations of FluidNeRF.

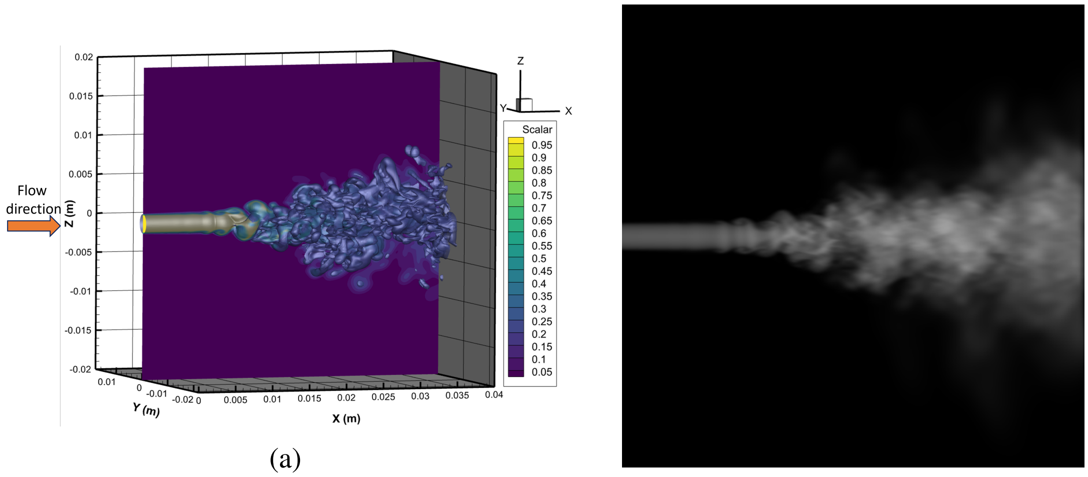
Chapter 6

Synthetic Experimental Design

After developing the FluidNeRF algorithm, it's essential to initially test it with synthetic data to establish its limitations before its application to experimental datasets. Synthetic data offers a reference point for assessing the accuracy of the method. For this purpose, synthetic data mimicking typical combustion and flow diagnostics scenarios was selected, ensuring the results obtained from simulation studies serve as a reliable benchmark for the flow diagnostics community. This section will first present the synthetic data, followed by a discussion on the image rendering algorithm that generates 2D perspectives of the synthetic volumes. Finally, we will introduce volumetric metrics designed to quantify the accuracy of the reconstruction.

6.1 Instantaneous CFD volume

The ground truth volume that is used for validation was generated from a direct numerical simulation (DNS) of a high-pressure, turbulent mixing jet as shown in Figure 6.1a. The jet flow field is represented as a scalar field with a value of 1.0 for pure fluid originating from the nozzle and 0.0 for the surrounding fluid. Scalar values of 0 and 1 indicate the mixture between the jet and the surrounding fluid. The jet originates from a circular nozzle with an exit diameter of 2.36 mm with a Reynolds number of 5000, as described by Sharan and Bellan [220]. The data is analogous to flow diagnostic techniques, including laser-induced fluorescence in combustion or passive scalar flow visualizations such as that achieved with dye or smoke injection. The current data set has several features of practical interest, including i) laminar region with a top hat cross-section, ii) transition region with large-scale asymmetric flow structures forming around



(a)

(b)

Figure 6.1: (a) A central x-y slice of the non-dimensional passive scalar volume originating from the nozzle and an isosurface of the scalar at a contour value of 0.2. The orange arrow indicates the flow direction (positive x-direction). (b) A line-of-sight integrated perspective image generated from the non-dimensional scalar value of the DNS volume.

the periphery, and iii) turbulent region consisting of low and high spatial frequency content. These characteristics offer a practical volume for assessing the impact of hyperparameters on the spatial frequency of FluidNeRF's reconstruction.

6.2 Time-Resolved CFD volume

The time-resolved validation data set was generated using the same DNS method of an atmospheric free jet. For this preliminary evaluation, the validation data set is composed of three time-steps of the initial start-up vortex as shown in Figure 6.1a. The DNS algorithm used for the simulation is described by Sharan and Bellan [220]. The free jet has an exit Mach number of 0.05 and a Reynolds number of 5000. The nozzle exit diameter is 2.36 mm. The time-resolved simulation had normalized time steps defined by Eqn. 6.1,

$$t^* = t/(D/U) \tag{6.1}$$

where D is the exit diameter, U is the exit velocity, and t is the iteration time. The iteration starts with t = 0, where no flow has exited the nozzle. The three time steps for the validation data set are t^* = 14, 15, and 16. The DNS model simulates a passive scalar using the compressible mass transport equation [220] with values of 1.0 for pure fluid originating from the nozzle and 0.0 for fluid originating from the surroundings. Values between the extremes correspond to a mixture of the two fluids caused by advection and diffusion. The simulated volume is analogous to different flow diagnostic techniques, including laser-induced fluorescence or flow seeded with dye or smoke. The time-resolved jet provides a smooth laminar region that develops more complex structures due to the shear layer and leading vortex. Although, these time steps are less spatially complex as compared to the flow from the static case. The camera field of view and layout that were used for the time-resolved perspective view generation are shown in Figure 7.7. The images were rendered using the method described in Section 6.3.

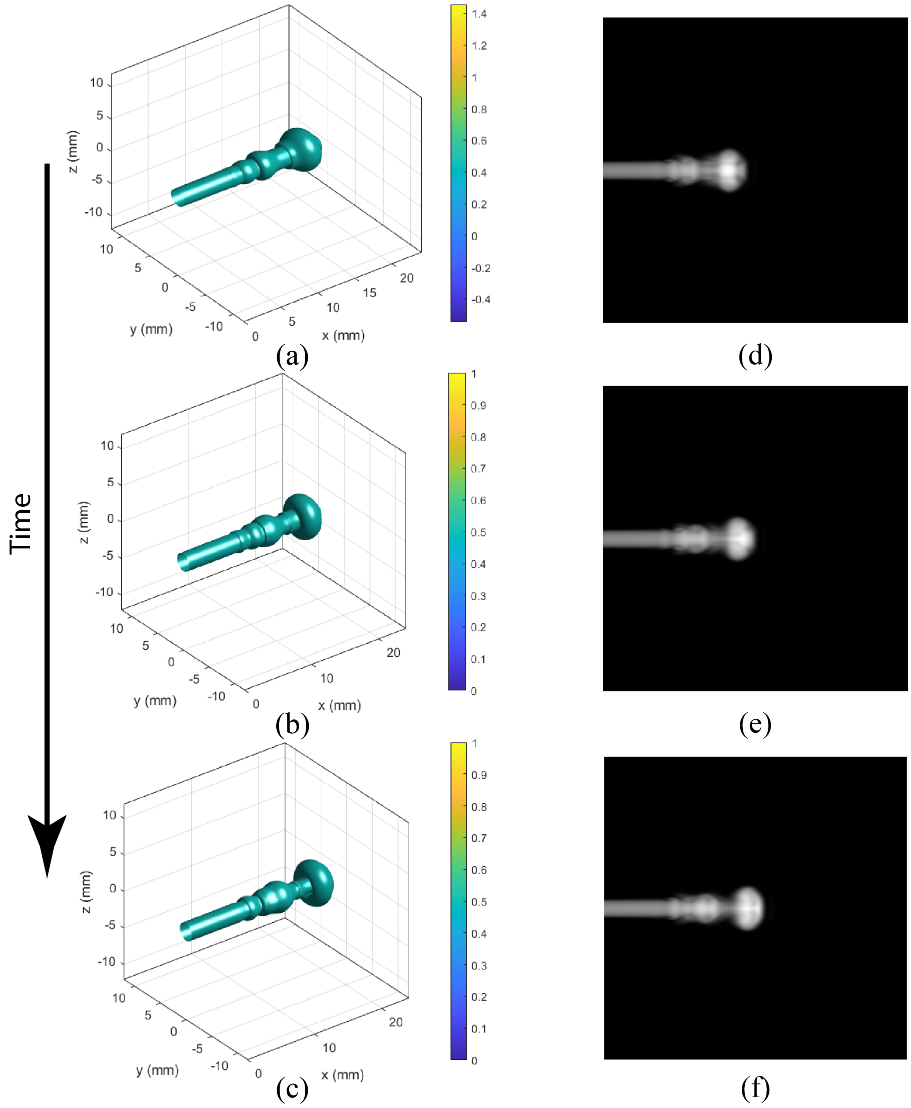


Figure 6.2: (a-c) Isosurface of a DNS solution of a free jet for t^* = (a) 14, (b) 15, and (c) 16. (d-f) A corresponding simulated image from the same perspective for all three time steps..

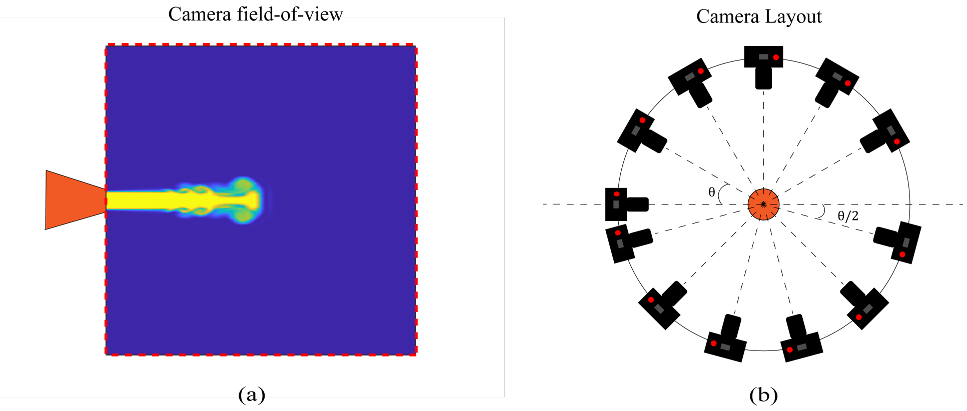


Figure 6.3: Schematic showing (a) camera field of view relative to the nozzle exit, and (b) the camera layout for maximum angular spacing within the plane.

6.3 Synthetic image generation

The synthetic perspectives of the DNS jet volume were generated using the Advanced Flow Diagnostics Laboratory's (AFDL) in-house tomography software [221, 222] as shown in Figure 6.1b. Before the CFD values could be used in the AFDL software, the values were interpolated from a variable-sized mesh grid onto a constant-sized voxel grid using a linear interpolation scheme. The jet volume used for rendering the images was discretized into 801×801×801 volume elements with a voxel size of 0.05 mm. The AFDL software assumes that the optics can be approximated with the thin lens equation and that the volume is optically thin, although the primary lens has a finite diameter. The synthetic images are formed by iteratively casting rays (N_{rays}) from each voxel position to the sensor using ray transfer matrices. Casting a discrete number of rays produces a type of shot noise that is typically found in imaging. This section discusses the method for simulating perspective images.

The synthetic imaging technique assumes a traditional camera configuration with an objective lens that focuses light onto the sensor plane. Figure 6.4 presents a simple 2D schematic of the ray tracing technique while showing variables within the software. For this schematic, the z-axis of the volume is aligned with the optical axis of the objective lens. All parameters of the simulation are measured relative to the optical axis. The volume is positioned such that the origin of the volume is located at the nominal focal plane of the objective lens, as indicated by the object distance (l_o). For this simulation, the main lens is modeled as a thin lens with a

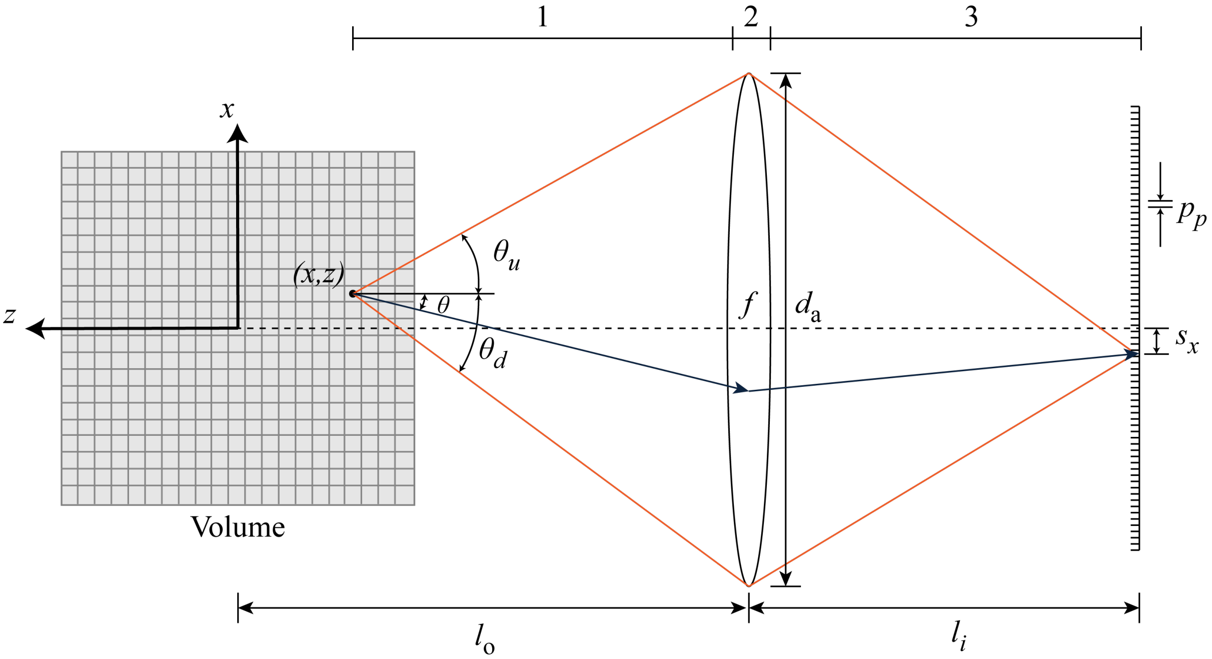


Figure 6.4: 2D schematic of the ray-tracing process used for generating images of the DNS volumes.

focal length, f, and an aperture diameter of d_a. The center of the sensor is aligned with the optical axis and is located at the image distance, l_i, from the primary lens. Each pixel of the sensor has a finite pitch, p_p, and the image resolution determines the size of the sensor. With the assumption of a thin lens, the object and image distance is related to f through the thin lens equation (Eqn. 6.2) and the magnification with Eqn. 6.3,

$$\frac{1}{l_i} + \frac{1}{l_o} = \frac{1}{f} \tag{6.2}$$

$$M = \frac{l_i}{l_o} = -\frac{h_i}{h_o} \tag{6.3}$$

where h_i and h_o is the height of a subject at l_i and l_o, respectively. The height in image space h_i is measured relative to the optical center as denoted by s_x and h_o relative to x from Figure 6.4. Note that modern camera lenses are composed of several optical elements; however, they can generally be modeled as thin lenses by measuring l_i and l_o relative to the principal plane of the compound lens. For a given lens, l_i and l_o can be determined through Eqn. 6.2 and Eqn. 6.3 if the focal length and magnification are known or measured.

Figure 6.4 presents a simple schematic of the ray tracing simulation. For a traditional camera, there are three main segments of ray tracing: i) emission from a point and traversing through object space to the primary lens, ii) interaction with the lens, and iii) projection of the focused image onto the sensor, where it is captured by individual pixels. We will discuss each segment of the rendering technique independently.

The first step of ray tracing is to convert the local volume coordinate system to the camera coordinate system, where the ground truth volume is generated such that the origin of the volume is located in the center of the volume. The conversion is completed by rotating the local volume coordinates to the camera coordinate frame to the orientation as shown in Figure 6.4. Since we are simulating 2D images of a 3D volume, the camera will be rotated around the x-axis (ϑ_x) and the y-axis (ϑ_y). The conversion between the local volume coordinates and the camera coordinates is calculated using

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{cam}} = \begin{bmatrix} \cos\vartheta_y & 0 & \sin\vartheta_y \\ \sin\vartheta_x \sin\vartheta_y & \cos\vartheta_x & -\sin\vartheta_x \cos\vartheta_y \\ -\cos\vartheta_x \sin\vartheta_y & \sin\vartheta_x & \cos\vartheta_x \cos\vartheta_y \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{vol}} \tag{6.4}$$

For our scalar field synthetic volume, the coordinates, $(x, y, z)_{\text{vol}}$, correspond to the center of each voxel since each voxel has a constant value throughout. For the voxel location, θ_u and θ_d are calculated to determine the maximum angle a ray to be captured by the primary lens using Eqn. 6.5 - 6.6.

$$\theta_u = \tan^{-1}\left(\frac{(d_a/2) - x}{l_o + z}\right) \tag{6.5}$$

$$\theta_d = \tan^{-1}\left(\frac{-(d_a/2) - x}{l_o + z}\right) \tag{6.6}$$

Each voxel is approximated as a sphere to simulate a voxel with finite size, with the sphere's diameter equal to the square voxel's side length. The origin of a ray that is traced from a voxel is randomly generated on the surface of the sphere. The number of rays generated for voxels located at the nominal focal plane (l_o) is N_{rays}. To account for the solid angle (Ω) of Eqn. 2.5, the number of rays cast from a voxel is determined from the ratio between the collection angle

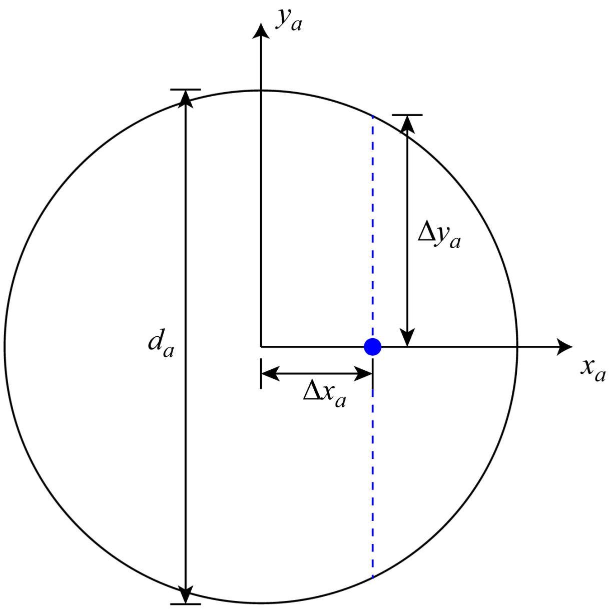


Figure 6.5: Schematic of a circular aperture utilized for the synthetic image generation.

of that voxel and the collection angle at the nominal focal plane (θ_{col}) as shown in Eqn. 6.7.

$$N_{\text{rays,vox}} = N_{\text{rays}} \left(\frac{\theta_u - \theta_d}{\theta_{col}} \right)^2 \tag{6.7}$$

Once a ray has a randomly generated origin on the sphere, the ray is given an initial, randomly generated propagation angle, θ, that falls between θ_u and θ_d, as shown in Figure 6.4. For a 2D thin-lens, a second angle in the y-direction (into or out of the page) is determined for propagating the ray in 3D space, where the angle is also randomly selected so that it is captured by the lens (Figure 6.4. For a circular aperture, the maximum angles (ϕ_u and ϕ_d) are calculated after randomly generating θ for the x-direction. Depending on θ, the size of the aperture in the y-direction Δy_a changes as presented in Figure 6.5. The x-location (x_a) on the aperture must be determined using Eqn. 6.8 to calculate ϕ_u and ϕ_d. Δy_a can be calculated using the equation of a circle (Eqn. 6.9), which calculates the y-coordinate on the edge of the circle. The y-coordinate corresponds to the edge of the circular aperture. Once the size of the aperture in the y-direction is determined for a given propagation angle, θ, ϕ_u, and ϕ_d are determined through Eqn. 6.10 - 6.11.

$$x_a = x + (z + l_o) \tan\theta. \tag{6.8}$$

$$\Delta y_a = \sqrt{\left(\frac{d_a}{2}\right)^2 - x_a^2} \tag{6.9}$$

$$\phi_u = \tan^{-1}\left(\frac{\Delta y_a - y}{l_o + z}\right) \tag{6.10}$$

$$\phi_d = \tan^{-1}\left(\frac{-\Delta y_a - y}{l_o + z}\right) \tag{6.11}$$

Similar to θ, an initial randomly generated propagation angle (ϕ) is given to each ray of a voxel.

Once a ray is given θ and ϕ for an origin point on the surface of the sphere of a voxel, the ray is propagated to the primary lens as indicated in segment 1 in Figure 6.4. The propagation of the ray is calculated using trigonometric relations given by

$$x_a = x_a = x + (z + l_o) \tan\theta \tag{6.12}$$

$$y_a = x_a = y + (z + l_o) \tan\phi. \tag{6.13}$$

The second segment for ray tracing is the deflection of a ray caused by the lens. The deflection assumes the thin lens approximation, where the ray instantaneously changes angles. The new ray propagation angles after the lens are given by

$$\theta_2 = \tan^{-1}\left(\frac{x_a - x}{z + l_o} - \frac{x_a}{f}\right) \tag{6.14}$$

$$\phi_2 = \tan^{-1}\left(\frac{y_a - y}{z + l_o} - \frac{y_a}{f}\right) \tag{6.15}$$

Finally, the third segment propagates the ray from the lens to the sensor. The propagation in image space uses trigonometric relations as given by

$$x_3 = x_a + l_i \tan\theta_2 \tag{6.16}$$

$$y_3 = y_a + l_i \tan\phi_2 \tag{6.17}$$

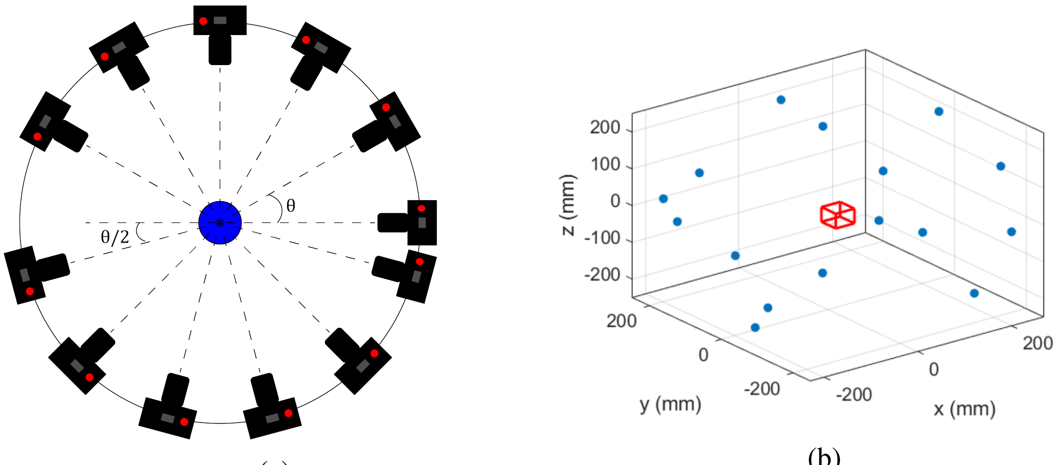
Once x_3 and y_3 are calculated, the pixel that the ray interacts with must be determined given the size of each pixel (p_p). The signal of the ray is accumulated on the sensor, with the intensity of each ray corresponding to the originating voxel value. The image rendering settings for this work are presented in Table 6.1. These settings produced a maximum intensity that would represent a 16-bit image.

Table 6.1: Camera settings for the free jet synthetic experiments.

Magnification	**Pixel pitch**	**Focal length**	**f/#**	**Resolution**	N_{rays}
-0.2	0.01 mm	60 mm	3.5	800 × 800	500

6.3.1 Camera layout

As shown in Figure 6.6a, a planar camera distribution and a spherical camera distribution are used to evaluate the impact of camera layout on reconstruction quality for the instantaneous data set. Alternatively, the time-resolved case will evenly distribute 60 cameras in a circular plane. The planar layout consists of up to 120 evenly-spaced perspectives, with the focal plane at the center of the volume. The camera plane is perpendicular to the flow direction. Subsets of the 120 cameras are generated to evaluate the effect of the number of cameras. Note that a few perspectives of the 120 cameras had distinct aliasing artifacts due to the near-perfect alignment of the voxel grid axes with the image sensor coordinates, a common issue with discretized schemes. Figure 6.7 shows an example of one such image, where the aliasing is present in the form of horizontal bands in the higher-intensity regions. Future work should try to eliminate the synthetic aliasing, but the images are retained here, where we note that both FluidNeRF and ASART used an identical set of images for reconstruction. The camera locations for the spherical data set were calculated using a Fibonacci lattice to distribute camera locations around a sphere (Figure 6.6b). After rendering the projections, the images were post-processed to apply varying levels of Gaussian image noise. The images' signal-to-noise (SNR) ratio was determined relative to the maximum intensity in the image, with levels of 1%, 2.5%, 5%, and 10%. The latter is an extreme case but will be a good test for experimental images.



(a) (b)

Figure 6.6: ((a) Planar camera layout with the jet marked in blue with the mean velocity coming into or out of the page, and (b) spherical camera positions calculated using Fibonacci lattice with the free jet volume indicated by the red box.

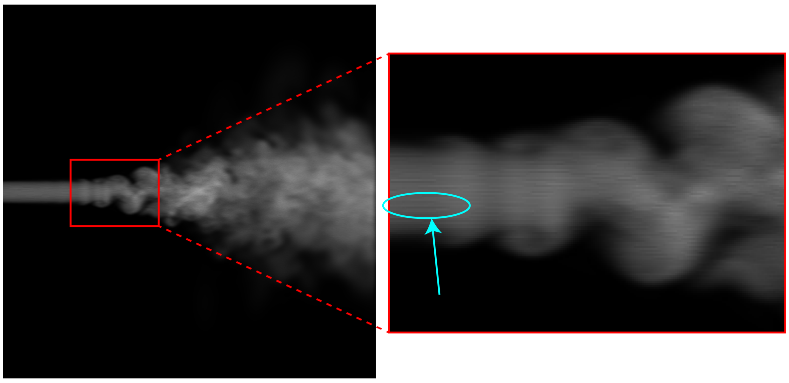


Figure 6.7: A line-of-sight integrated perspective image of the DNS free jet data. A small amount of aliasing, in the form of horizontal streaks, caused by the synthetic image rendering method is highlighted.

6.4 Numerical metrics

In addition to visual inspection of the reconstructed volumes, two different performance metrics are used to quantify the accuracy of the reconstructions. Volume metrics compare the volume reconstruction to a ground truth, which is possible due to the synthetic data set. One of the quantitative metrics is the normalized root-mean-square-error (NRMSE), given by

$$\varepsilon_2(E, \hat{E}) = \sqrt{\frac{\sum_j^{N_{vox}} (E_j - \hat{E}_j)^2}{\sum_j^{N_{vox}} (\hat{E}_j)^2}}. \tag{6.18}$$

E is the predicted volume value, $\hat{E}$ is the ground truth, and N_{vox} is the number of voxels or query points used. NRMSE is one of the most common accuracy metrics in volumetric imaging [1]. NRMSE is an averaged error quantification across the volume. While not as common in volumetric imaging, structural similarity index metric (SSIM) has been popular for comparing images in the computer vision community [6] but has been employed recently for tomography [8]. SSIM investigates the luminance, contrast, and structure the QoI to evaluate the overall structure and intensity of the image/volume as shown in Eqn. 6.19.

$$\text{SSIM} = \frac{(2\mu_E\mu_{\hat{E}} + c_1)(2\sigma_{E\hat{E}} + c_2)}{(\mu_E^2 + \mu_{\hat{E}}^2 + c_1)(\sigma_E^2 + \sigma_{\hat{E}}^2 + c_2)} \tag{6.19}$$

μ is the mean value across sample points, σ_E and $\sigma_{\hat{E}}$ is the variance of the intensity values for predicted and truth volumes, $\sigma_{E\hat{E}}$ is the covariance, and c_1 and c_2 are constant variables that stabilize SSIM. Luminance is compared to the ground truth using μ. The second parenthesis in the denominator of Eqn. 6.19 captures the contrast compared to the ground truth using σ_x and σ_y. The structural comparison is employed through σ_{xy}.

6.4.1 Grid-based reconstruction method

FluidNeRF is compared to a modern ART-based method called the adaptive, simultaneous algebraic reconstruction technique (ASART). As noted before, ART-based methods have been the most popular iterative tomography methods for combustion and flow diagnostics. Simultaneous ART-based (SART) methods improve the original ART method's performance by updating the predicted volume across all projections rather than a projection-by-projection update. The simultaneous methods increase computational efficiency and are motivated by volumetric imaging [1]. ASART incorporates a modified multilevel access scheme to arrange the order of projection data, adaptively correct the relaxation parameters that correct discrepancy between actual and computed projections, and a column-sum substitution [70]. The volume initialization and update equations for ASART are presented in Eqn. 2.29 and Eqn. 2.30.

Our ASART implementation uses a bilinear interpolation scheme to relate voxels to pixels (w_{ij}), where each voxel contributes intensity to a 2x2 grid of pixels for each camera. This work uses μ = 0.5 to ensure numerical stability while providing an adequate convergence rate. The volume was reconstructed at a resolution of 400^3 voxels, unless indicated otherwise. When making direct comparisons, FluidNeRF is interrogated at the exact spatial location as the center of each voxel. The ASART implementation allows for a maximum of 250 iterations of the technique before stopping; however, a convergence criterion was used for early stopping since ART-based methods are semi-convergent [1]. For the convergence criterion, the error is accumulated for each iteration $(P_i - \sum_j^N w_{ij} E_j^{(k)})$. The tomography algorithm stops the reconstructions when the iteration error varies less than 2% over the last 20 iterations using a sliding mean with a window of 4. We chose the convergence criterion to be similar to what is used by FluidNeRF. Our ASART method is implemented in C/C++ and parallelized using OpenMP. ASART was processed on Intel Xeon Gold 6248R processors provided by the Auburn University Easley Cluster.

6.5 Chapter Summary

This chapter outlines the synthetic data generation used to evaluate FluidNeRF and ASART with a ground truth. A DNS-generated CFD solution, provided by Sharan and Bellan [220], was utilized for the ground truth because the solution included a passive scalar for investigating mixing. The CFD volume provides regions of varying spatial complexity as the volume captures the transition from laminar to turbulent flow. Two datasets were generated for testing the static and time-resolved FluidNeRF methods: an instantaneous volume that captures the flow in a fully developed state and a time-resolved dataset that includes three-time steps during the transient of the leading vortex structure. After discussing the ground truth volume, the synthetic imaging rendering method that was utilized to generate perspectives of the volume was discussed. The image generation method assumed an optically thin volume but accounts for finite-sized optics. One point of interest is the effect of different camera layouts on reconstruction quality; therefore, planar and spherical camera layouts are introduced. The results will utilize the NRMSE and SSIM accuracy metrics for quantitative analysis of the reconstructions. Finally, a discussion on the grid-based traditional tomography method that is employed to provide a basis of comparison to the FluidNeRF method is provided.

Chapter 7

Synthetic Results and Discussion

The following chapter covers the initial investigation of the FluidNeRF tomography method using the synthetic dataset discussed in the previous chapter. The results generated in this section provide an error measurement relative to a ground truth. The chapter is split into two major sections that investigate the static FluidNeRF and TR-FluidNeRF methods, respectively. Each one of these sections begins by studying the effects of the hyperparameters on reconstruction accuracy, providing rationale for the selected default hyperparameters used in the remainder of the results. Both methods examine the result of the camera layout on the reconstruction. Static FluidNeRF includes the reconstruction accuracy as a function of image noise added to the projections. Static FluidNeRF results are compared to the ASART tomography method. Once this is established, TR-FluidNeRF is compared to the static FluidNeRF method. These studies should supply an initial understanding of the capabilities of the FluidNeRF method.

7.1 Static FluidNeRF

A preliminary comparison between FluidNeRF and ASART is made by visualizing a central slice along the jet axis as presented in Figure 7.1. The overall structure of the jet is clearly reconstructed using both methods, with the differences being most apparent in the error plots shown in (d) and (e). Both methods capture the laminar jet core, the transition region near the end of the jet core, and the diffuse, turbulent flow region downstream. The most significant difference between the two reconstructions is found in the laminar region, where ASART has difficulty capturing the sharp transition between the jet core and ambient fluid. Due to the

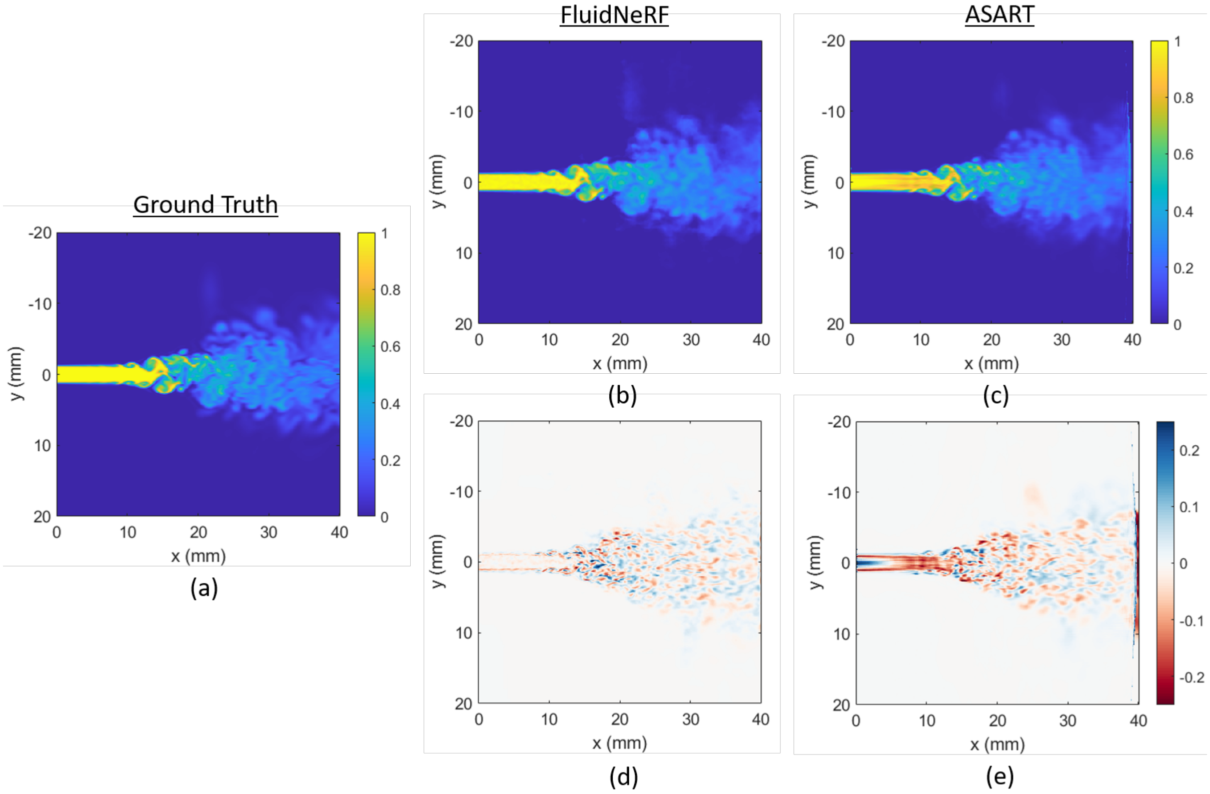


Figure 7.1: A central x-y slice of (a) the ground truth free jet, (b) FluidNeRF reconstruction, and (c) ASART reconstruction using 15 cameras in the planar configuration. The difference between the ground truth and (d) FluidNeRF and (e) ASART reconstructions.

necessary volume discretization, ASART has difficulty modeling the sharp gradient at the jet's edge and compensates by underpredicting the intensity in the core. This is also magnified by reconstruction artifacts that occur at the boundaries of the volume. FluidNeRF also appears to slightly outperform ASART in the transition region where sharp features are still prevalent; however, the differences between the two methods begin to diminish in the turbulent flow region where finer scale dominates the flow. In this region, both methods still reconstruct the larger scales and general distribution of the jet fluid throughout the volume but struggle to capture the finest details, thus indicating the limit of the spatial resolution.

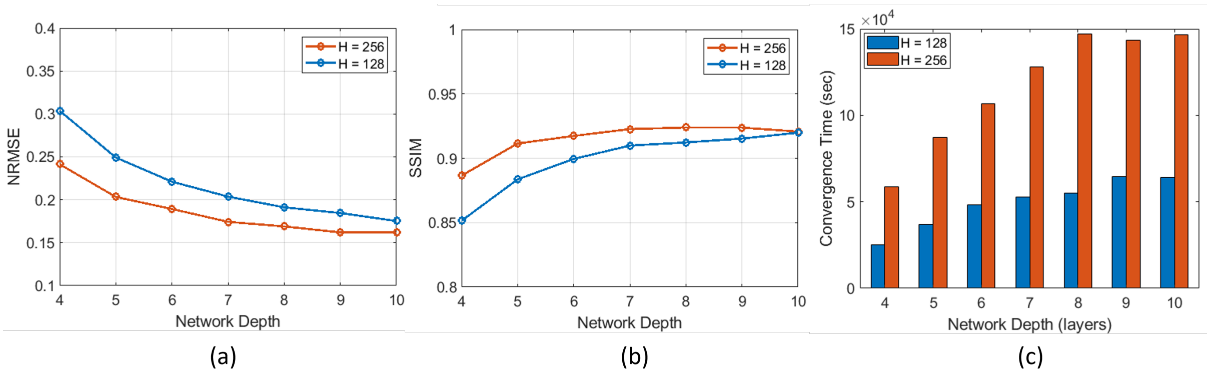
In the remainder of this section, a closer look at the performance of FluidNeRF is offered, beginning with the influence of the various hyperparameters. Once the hyperparameters have been selected, the impact of camera layout and image noise are investigated.

7.1.1 Hyperparameters

The choice of hyperparameters in the FluidNeRF reconstruction, such as network depth and height, can significantly impact the reconstruction quality, similar to voxel resolution in ART-based methods, as well as the computational time. The hyperparameter results presented here use the 15 evenly-spaced planar camera layout. This first investigation is the MLP network size. The other hyperparameters were held constant for this comparison, as shown in Table 5.1. The ability of the network to approximate the radiance field is affected by network size. Figure 7.2 illustrates the reconstruction accuracy as a function of network depth at two different network heights. In general, a deeper network improves the approximation of the volume. For a height of 256 nodes, the reconstruction had diminishing returns or leveled after approximately eight layers. Reducing the network height by half does decrease the accuracy for a given depth, but the accuracy of the shorter network converges to the taller network with increasing depth. SSIM is equal for both heights with a depth of ten layers. Therefore, height has less of an impact on the accuracy of the representation.

In addition to approximating the volume, the network depth also directly impacts computational efficiency, as the number of operations is directly proportional to the network size. For the case with ten layers, the taller network has 256*10 nodes compared to 128*10 nodes of the shorter network. Fewer operations lead to improved computational efficiency in reconstruction, as depicted by the convergence time for each network size in Figure 7.2(c). In almost every case, the computational time was $< 50\%$ with a height reduction of 50%. These results show that quicker reconstructions can be achieved with a measured trade-off in reconstruction accuracy. For the rest of this work, a network size of eight layers with 256 nodes each is chosen to maintain the focus on higher fidelity reconstructions.

The next hyperparameter investigated is positional encoding. As discussed, increasing L increases the dimensionality of the inputs, thus allowing the MLP to better reconstruct the higher frequency content in the volume. The free jet volume is composed of different regions (laminar, transition, and turbulent) that have varying levels of spatial frequencies, as shown in Figure 7.3(a). The reconstruction NRMSE for each positional encoding for the three regions



(a)

(b)

(c)

Figure 7.2: Reconstruction accuracy as a function of network depth at different heights (128 & 256) using (a) NRMSE and (b) SSIM. (c) Convergence time for each combination of network height and depth.

is presented in Figure 7.3(b). The turbulent and transition regions are characterized by higher spatial frequency content, and they are much more challenging to reconstruct, as was also shown in Figure 7.1. Thus, this region had higher NRMSE errors. The laminar region has much lower NRMSE, with L having little influence on the accuracy. While increasing L is not needed, it is also demonstrated that increasing L does not negatively affect the NRMSE of the reconstruction. The transition and turbulent regions show similar trends where increasing L causes an improvement in performance (reduced NRMSE) until $L = 8$, after which further increases in accuracy are not observed. Therefore, $L = 8$ is considered as the optimal encoding for this flow.

Another advantage of positional encoding is the convergence rate, as illustrated in Figure 7.3(c-e). The termination point of each line indicates the convergence rate as a function of L, where increasing L reduces the number of iterations for convergence. Note that FluidNeRF reconstructed the entire volume; hence, the termination point is the same across all regions. The laminar region converged quickly for all L. There is little difference between the trends of each encoding. The transition and turbulent regions show similar results as laminar, but the convergence rate is more apparent with increasing L. In these regions, the NRMSE exhibits a negative initial slope that progressively becomes steeper with increasing L until approximately $L = 8$. After $L = 8$, the NRMSE lines coalesce to the same trend. Increasing L causes $< 5\%$ increases in computational time per iteration between $L = 0$ and $L = 8$. However, approximately 30% fewer iterations are required for convergence. Thus, increasing L improves the ability to

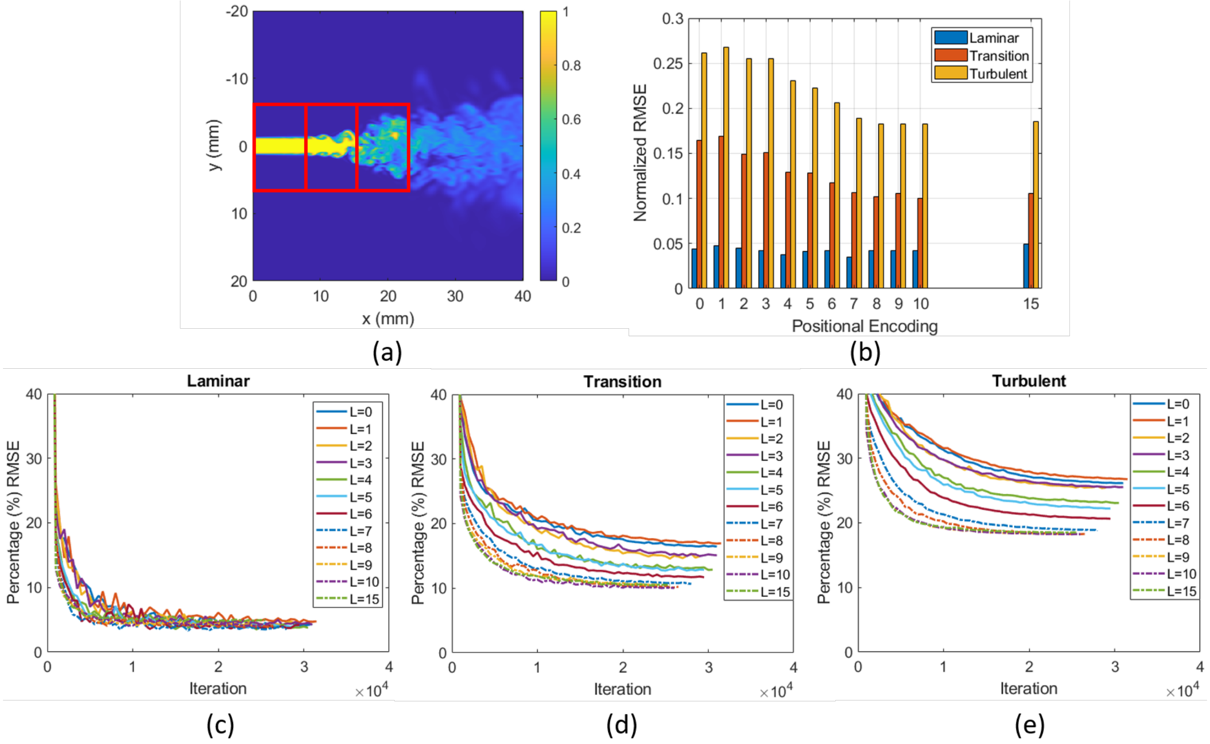
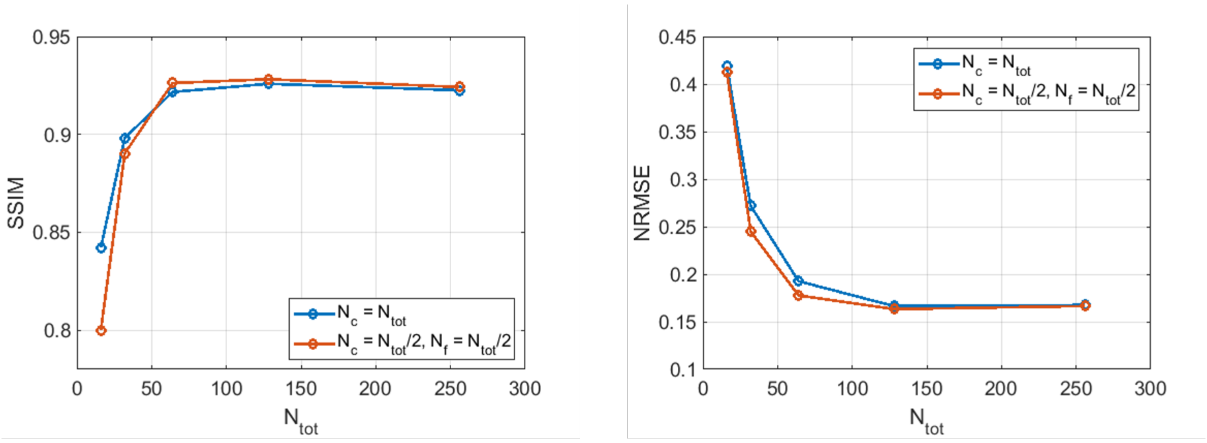


Figure 7.3: (a) x-y central slice of the ground truth volume with red boxes indicating the laminar, transition, and turbulent regions (left to right) utilized for the comparison. (b) NRMSE as a function of positional encoding for the three different regions. NRMSE as a function of reconstruction iterations for the (c) laminar, (b) transition, and (c) turbulent regions.

capture the volume accurately and reduces the training time. The authors would like to note that the optimal value for L depends on the spatial complexity of the flow field of interest; however, L values greater than the optimal do not cause a reduction in accuracy.

The final hyperparameter that is considered is the number of spatial samples along each ray used for rendering the perspective pixel. Figure 7.4 shows NRMSE and SSIM of the FluidNeRF reconstruction utilizing semi-random sampling ($N_c = N_{tot}$) and multi-resolution sampling ($N_c = N_{tot}/2, N_f = N_{tot}/2$) methods. The reconstruction accuracy quickly improves when N_{tot} equals or exceeds 64. With N_{tot} samples greater than this limit, the reconstruction quality appears to be limited by the MLP approximation of the volume rather than the discrete ray tracing method, as indicated by a constant accuracy for $N_{tot} > 64$. Figure 7.4(b) also illustrates that clustering sample points around the peak gradients in the volume only slightly improves the quality of the reconstruction, with the difference most notable at N_{tot} of 32 and 64. Alternatively, SSIM indicates a much more pronounced difference for $N_{tot} \leq 32$, with

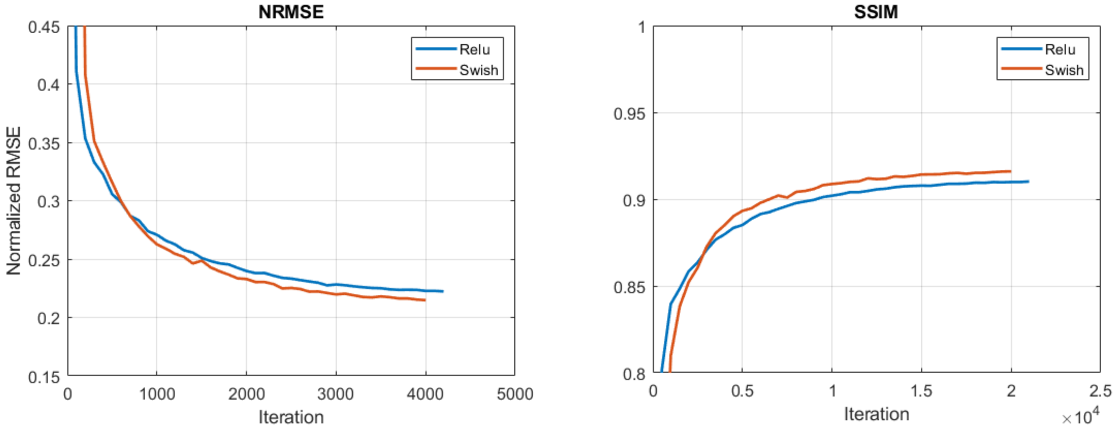


(a) (b)

Figure 7.4: (a) NRMSE and (b) SSIM as a function of N_{tot} samples using 15 cameras with and without hierarchical sampling.

the coarse sampling outperforming the multi-resolution sampling. The information provided by N_c with low sample densities does not seem to provide sufficient information to adequately determine where the fine samples should be located. Using these results, N_{tot} should be equal to or greater than 128 to maximize the reconstruction quality. Note that this is lower than traditional discretized methods require along the LoS. Therefore, the optimal settings for maximizing reconstruction accuracy and reducing computational time are N_c = 64 and N_f = 64, where increasing sampling past this will increase the computational time without improving accuracy.

Figure 7.5 presents the reconstruction accuracy of FluidNeRF using ReLU and Swish activation functions. The activation function is a critical component for the non-linearity of the function approximation of the neural network. The swish network exceeds the accuracy values of the ReLU network, indicated by both SSIM and NRMSE. However, the difference between the two is minimal. The swish activation function is more complex and has a smooth differential compared to the ReLU function, which will help the swish network achieve a higher accuracy value. Due to the increased computational complexity, the swish activation function increases the computational time of an iteration by approximately 50% while slightly increasing accuracy. It was determined that the increased complexity was not required for this demonstration. Therefore, the following results for static FluidNeRF will employ ReLU.



(a)

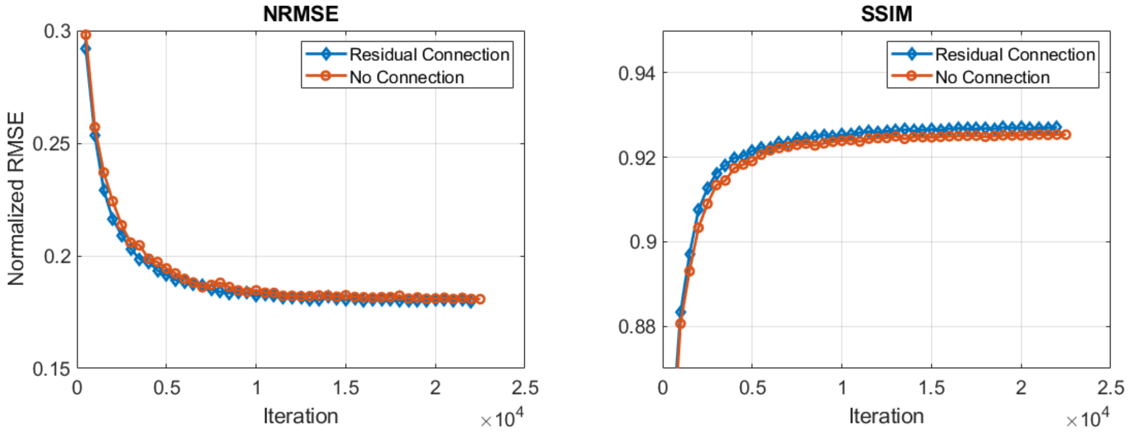
(b)

Figure 7.5: (a) NRMSE and (b) SSIM as a function of iteration for FluidNeRF utilizing ReLU and Swish activation functions for the 12-camera case.

A generalized network structure utilized for the results sections has a residual connection (Figure 5.4), taken from previous NeRF implementations. However, this connection has not been verified in the past literature. Therefore, we cover results comparing the convergence rate and final accuracy using a network with and without the residual connection. Figure 7.6 shows the quantitative metrics as a function of the iteration of both networks. The residual connection network trains similarly to the network without the connection. The residual network does converge to a negligibly higher value, where the final value is nominally the same. The convergence rate is also practically the identical. Therefore, both networks are adequate for the FluidNeRF problem. However, the residual neural network was developed to counter the issues of deep networks, including vanishing gradients. Thus, the results show that the residual connection does not impede in the training process and marginally improves accuracy. With these results, FluidNeRF will use the residual connection framework for the rest of the paper. The benefits will only become apparent as the depth of the network increases.

7.1.2 Camera layout

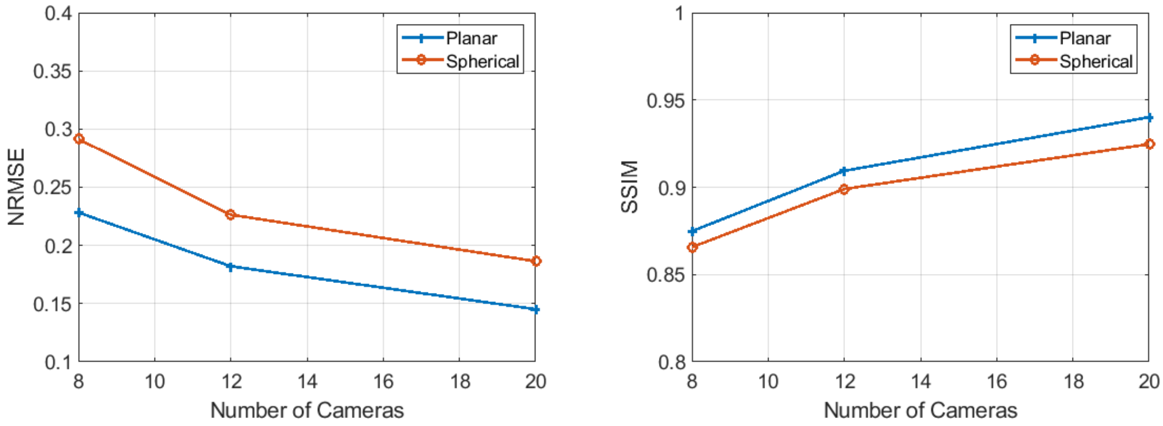
The next reconstruction variable that will be studied is camera layout. The volume was reconstructed with FluidNeRF using two camera configurations: a) a 360-degree evenly-spaced, planar layout representative of what might be expected in experiments and b) a more challenging spherical layout with the same number of cameras that provides better coverage of the



(a)

(b)

Figure 7.6: (a) NRMSE and (b) SSIM as a function of iteration for FluidNeRF's neural network with and without a residual connection. The reconstructions were conducted with 15 cameras with the same hyperparameters.



(a)

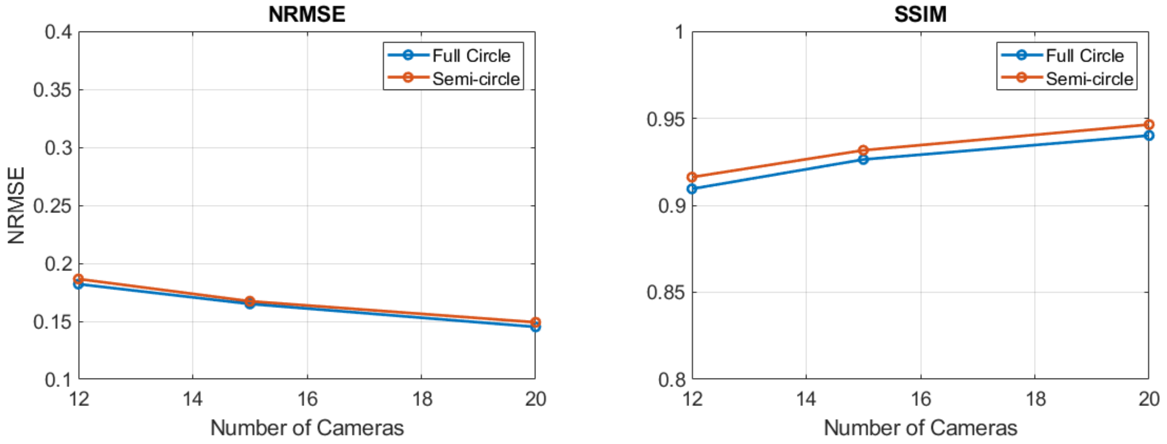
(b)

Figure 7.7: Reconstruction (a) NRMSE and (b) SSIM of FluidNeRF in the spherical and planar configurations with different number of perspectives.

whole angular space. Figure 7.7 shows the accuracy difference between the two layouts, where both have a similar trend, but the planar layout outperforms spherical for both metrics, NRMSE and SSIM. Planar outperforming spherical is perhaps counter-intuitive, as the increased angular separation of the spherical layout should improve the accuracy due to perspectives being more dissimilar, as discussed by [1]. This is the case for general tomography problems; however, the jet flow field utilized here is characterized by an axis of symmetry about which flow features are expected to be organized. Thus, the perspectives perpendicular to the jet axis best distinguish the flow features. Additionally, the spherical layout leads to perspective views with a relatively longer LoS through the jet, leading to less efficient use of image sensor resolution and dynamic range. Both configurations produce adequate reconstructions, but the planar configuration is superior for our case. Therefore, the rest of this work uses the planar configuration.

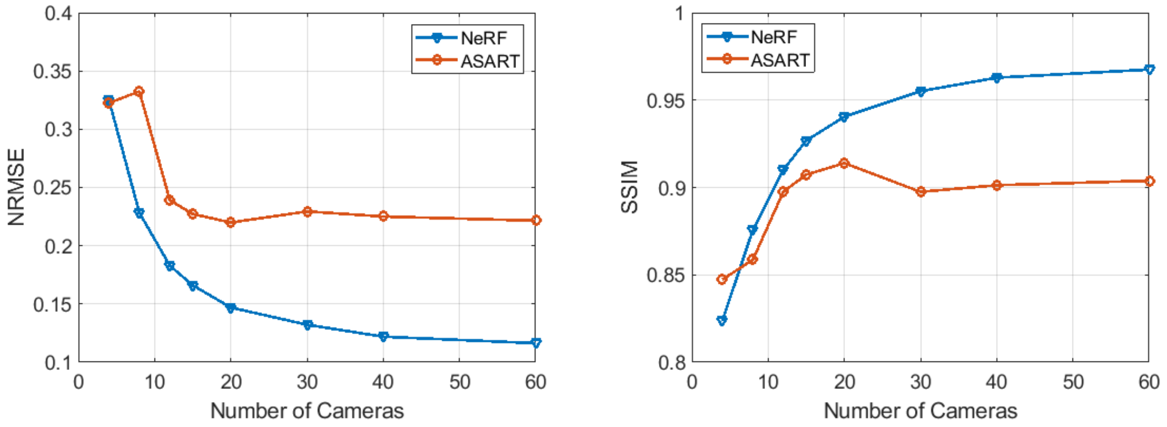
Figure 7.8 compares a full, 360°-planar layout to a 180°-planar layout (semi-circle) with various numbers of cameras. The semi-circle results suggest that minimal accuracy is lost when angular information is reduced. SSIM for the semi-circle was slightly higher than the full circle, but the difference is minor and could originate from convergence variation. The semi-circle layout is better suited for most flow diagnostics due to limited optical access, and this layout provides the ability to place blackout curtains or plates on the opposite side of the cameras to minimize background light. Therefore, FluidNeRF is expected to produce comparable results with the reduced angular information. Further limiting the total viewing angles would decrease reconstruction accuracy. A future systematic study should be conducted to understand the accuracy reduction for limiting view angles.

The following investigation compared the effect of the number of cameras on reconstruction accuracy with the planar layout to ASART, which is presented in Figure 7.9. For a practical camera layout with four cameras, typical for PIV/PTV, ASART and FluidNeRF performed about the same, with ASART performing slightly better as indicated by SSIM. Increasing the number of cameras to 8 and 12, common for camera configurations of scalar field reconstructions [1], drastically increases the accuracy of both methods. For these cases, FluidNeRF outperforms ASART. As expected, increasing the number of perspectives improves reconstruction quality for both cases but shows diminishing returns after approximately 20 cameras for ASART and 30-40 cameras for FluidNeRF. More importantly, FluidNeRF outperforms ASART for all cases, with 8-12 perspectives of FluidNeRF producing equal or better quality than ASART with the maximum number of viewpoints considered. We also note that the trends for ASART are not always monotonic. The authors believe this is most likely associated with aliasing issues related to the discrete nature of the ASART method, where the perfectly uniform distribution of cameras relative to the voxel grid can lead to numerical artifacts. Still, it is necessary to recognize that the same set of perspectives were used for FluidNeRF and ASART. Regularization techniques could be included in ASART at significant additional computational



(a) (b)

Figure 7.8: Reconstruction (a) NRMSE and (b) SSIM of FluidNeRF for full circle (360°) and semi-circle (180°) camera layout using different number of perspectives.



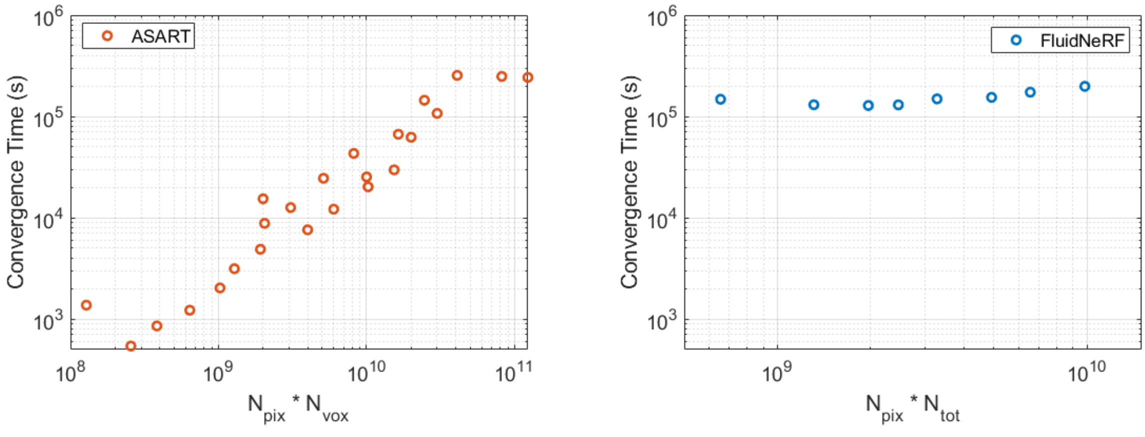
(a) (b)

Figure 7.9: (a) NRMSE and (b) SSIM for reconstructions using ASART and FluidNeRF using different numbers of cameras in the planar configuration.

expense to reduce the influence of these artifacts on the reconstruction. The lack of a discrete grid and random sampling through the volume enables FluidNeRF to avoid the same aliasing artifacts naturally.

7.1.3 Computational Efficiency

Given the disparity in programming languages and computational hardware utilized in the current implementations of ASART and FluidNeRF, a direct comparison of computational time is not appropriate. However, it is helpful to compare the trends of computational time for each algorithm as the overall scale (i.e., number of cameras, volume resolution) of the problem



(a) (b)

Figure 7.10: Convergence time for (a) ASART as a function of $N_{pix} * N_{vox}$ and (b) FluidNeRF as a function of $N_{pix} * N_{tot}$.

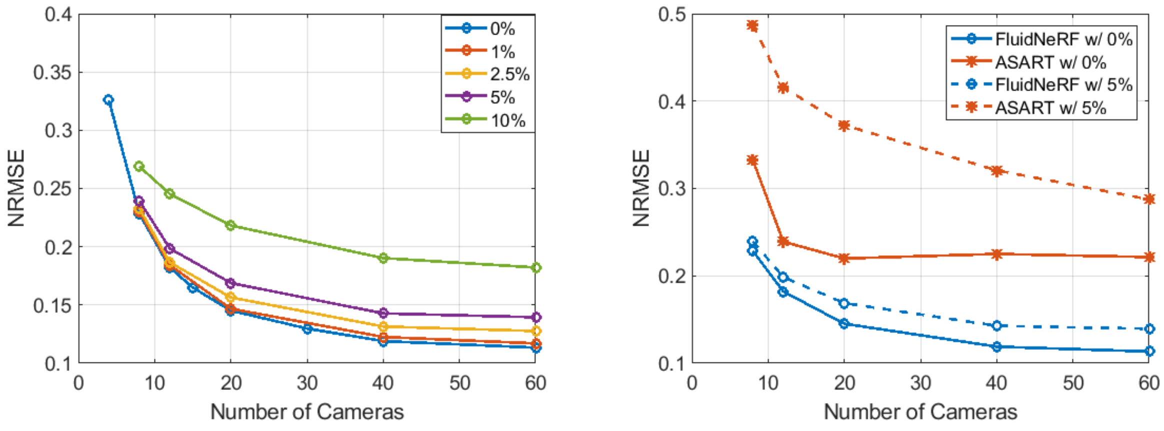
increases. The computational time associated with an iteration of each method is primarily expected to scale with O($N_{pix} * N_{vox}$) and O($N_{pix} * N_{tot}$) for ASART and FluidNeRF, respectively. However, the rate at which a method converges to a final solution may not scale according to these metrics. Both methods were run using 4, 8, 12, 20, 40, and 60 cameras, thus varying the amount of image data (i.e., N_{pix}) to compare trends in the overall reconstruction times. For each of the N_{pix} cases, ASART was reconstructed with different volume resolutions of 200^3, 400^3, 500^3, and 800^3 to represent a typical choice required by a user when using volume discretized techniques. An equivalent choice is unnecessary when using FluidNeRF as it is a gridless technique. However, one could reduce the size of the MLP or N_{tot} to achieve a similar effect. Figure 7.10 displays the convergence time for both FluidNeRF and ASART for various combinations. Note that the convergence time is provided for reference, but only the trends are compared here in the context of the scalability of each method to larger domains. For ASART, $N_{pix} = 4 * N_{cams}$ as we employ bilinear interpolation to relate image data to each voxel. As expected, the ASART convergence time scales linearly with $N_{pix} * N_{vox}$ due to the discrete relationship between voxels and pixels. FluidNeRF, conversely, is characterized by a convergence time that only modestly increases as the number of cameras or ray sampling rate is increased. The minimal increase in time indicates that FluidNeRF can effectively utilize the increased number of projections (N_{pix}), where the convergence time only slightly increased relative to the gain in accuracy with increasing N_{pix} (Figure 7.9).

The relationship between the density of the voxel grid and true spatial resolution in ASART is more complex, and a complete analysis is beyond the scope of this work. Nonetheless, a denser voxel grid will intrinsically reduce the limitation on spatial resolution. However, this increase in grid density can also lead to the tomography problem becoming more under-defined, meaning that multiple possible solutions could fit the data equally well. This under-defined nature can induce numerical instabilities, making the reconstruction process more challenging. Therefore, increasing resolution does not necessarily increase reconstruction quality. As a relevant aside, this was investigated by comparing the reconstruction accuracy for 400^3 and 800^3 volume resolution cases. ASART reconstruction accuracy stays relatively constant between the two resolutions. For example, the ASART reconstruction NRMSE using 15 cameras at 400^3 and 800^3 volume resolutions is 0.225 and 0.215, respectively. Therefore, increasing volume resolution slightly improved the reconstruction, but it did not reach the reconstruction accuracy of FluidNeRF (NRMSE = 0.165 using 15 cameras). Therefore, FluidNeRF has the ability to expand to higher-resolution cases while maximizing accuracy compared to discretized methods.

Another important distinction between ASART and FluidNeRF is the memory and storage requirements for each reconstruction. With ASART, the memory requirements for the solution scale with N_{vox}. For the 800^3 volume resolution case, each volume requires 2 GB of memory. This means that for larger volume resolutions, the memory requirements can quickly become significant, potentially limiting the applicability of ASART to high-resolution problems. Comparatively, FluidNeRF requires 1.9 MB to store the weights of the neural network for the default hyperparameters. This is significantly less than the memory requirements of ASART, making FluidNeRF a more memory-efficient option. It needs to be clarified if a deeper network would be needed to represent other flow fields; nonetheless, FluidNeRF holds distinct advantages for investigating problems with increasing volume size or resolution requirements.

7.1.4 Image Noise

This subsection evaluates the impact of image noise on reconstruction quality. Figure 7.11a shows the reconstruction quality at different levels of image noise for a varying number of



(a) (b)

Figure 7.11: (a) Reconstruction accuracy of FluidNeRF at different image noise levels as a function of number of cameras (color version required), and (b) comparison of NRMSE for FluidNeRF and ASART with 0% noise and 5% image noise using various numbers of cameras.

cameras. For FluidNeRF, the reconstruction quality slightly deteriorates for noise levels up to 5%, after which the impact of noise is more pronounced. Even at 10% noise, FluidNeRF can acquire adequate reconstructions. However, higher noise flattens the NRMSE curve, indicating that the noise reduces the effectiveness of increasing the number of perspectives.

Figure 7.11b compares ASART and FluidNeRF with and without 5% Gaussian noise applied to the perspectives. It is immediately apparent that FluidNeRF continues to outperform ASART even in the presence of noise, with ASART maintaining a lower NRMSE value throughout. Literature has shown that ASART is more robust to noise than most ART-based methods; however, it is still quite susceptible compared to FluidNeRF. As expected, noise has a much higher impact for cases with fewer cameras. Increasing the number of cameras allows ASART to overcome some of the noise-related challenges, but to the same degree as FluidNeRF, which is shown here to be more robust. Overall, FluidNeRF is more applicable for experiments since image noise is unavoidable.

The effect of noise on the ASART and FluidNeRF reconstructions is qualitatively illustrated in Figure 7.12 and Figure 7.13, which show 2D cross-sectional slices of the jet in the turbulent transition region at different noise levels for various number of cameras. For FluidNeRF, the differences in reconstructions are relatively subtle, with sharper features observed for low-noise, high-camera cases. The ASART reconstructions, on the other hand, show more

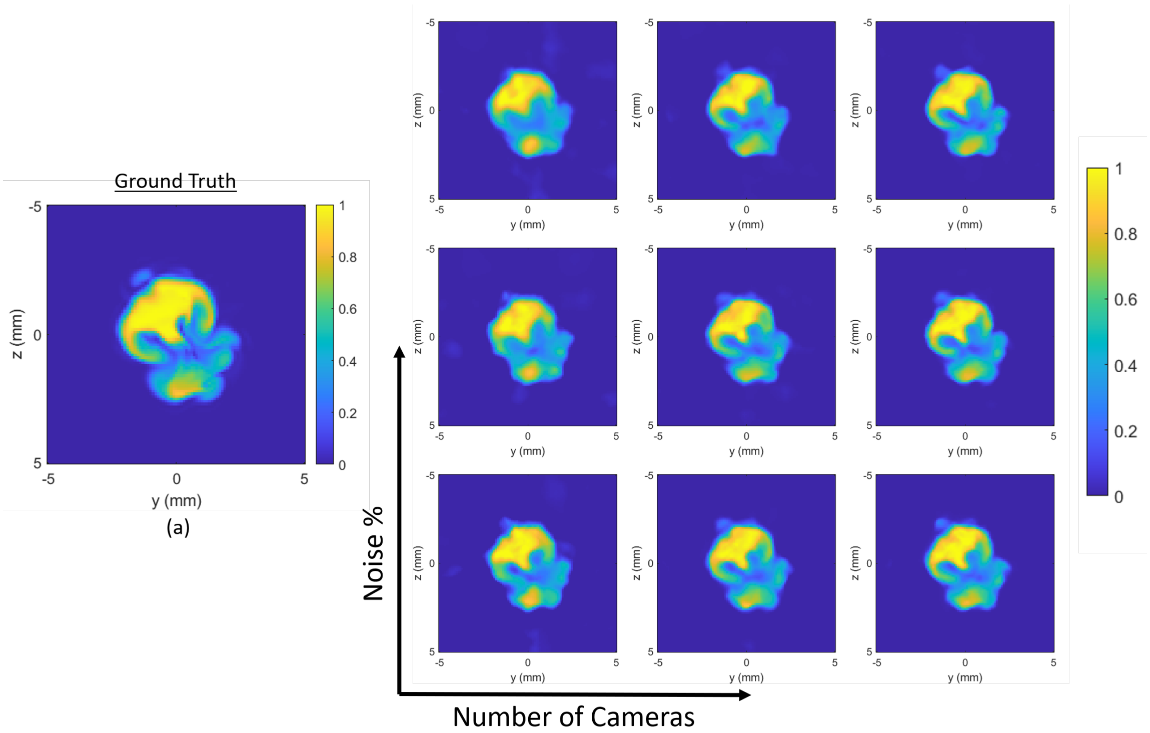


Figure 7.12: 2D slice of FluidNeRF reconstructions at x = 14.5mm downstream of the nozzle for noise levels of 1%, 2.5%, and 5% using 8, 20, and 60 cameras.

pronounced differences with reconstruction artifacts typically associated with discrete volume representations and aliasing effects. The artifacts are most notable for the lowest number of cameras where salt and pepper-type noise is prevalent.

7.2 TR-FluidNeRF

The preliminary results presented in this section focus on reconstructing the time-resolved free jet over three-time steps, as shown in Figure 6.2. Currently, the minimal time steps were chosen to provide the best-case scenario for TR-FluidNeRF by reducing the complexity of the temporal information to be fit. Additionally, it is the best-case scenario for employing physics models with only a limited subset of the flow dynamics required to be solved for future investigations. The smaller data set also allows for quicker debugging during the algorithm development. However, the model can be extended to include a large amount of time-resolved frames, where a future investigation is critical in understanding the limits on the quantity of time steps. This section covers an investigation of TR-FluidNeRF hyperparameters and compares

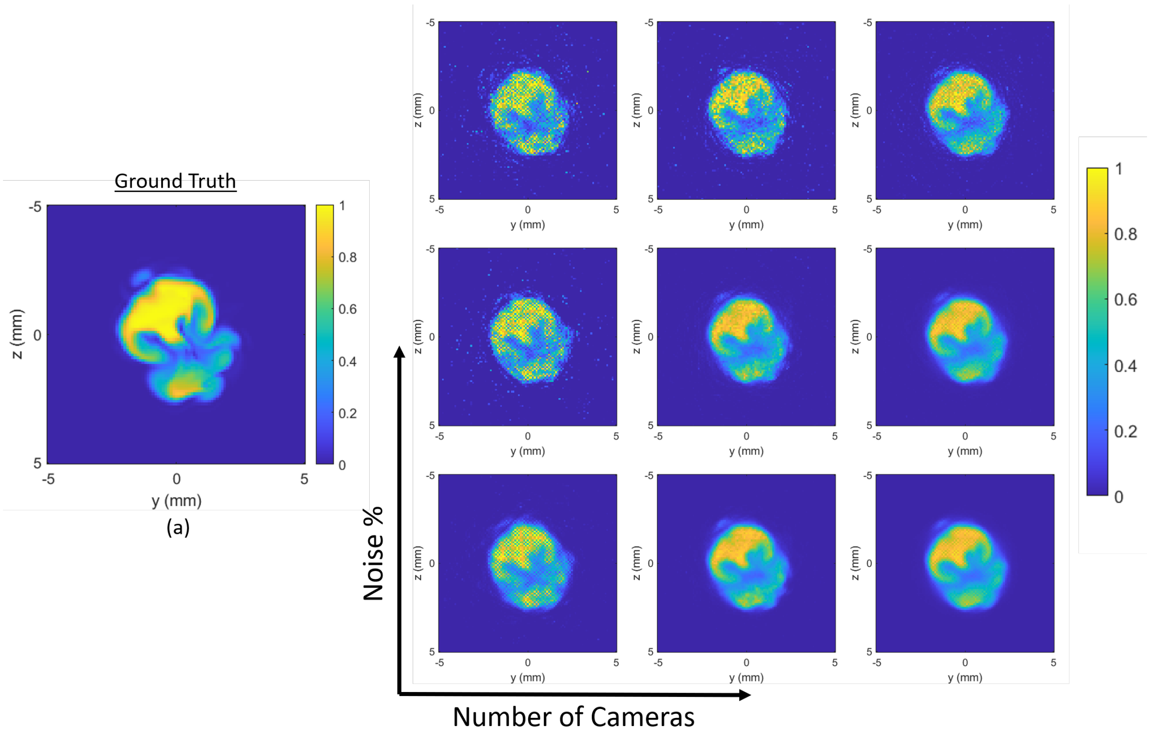


Figure 7.13: 2D slice of ASART reconstructions at x = 14.5mm downstream of the nozzle for noise levels of 1%, 2.5%, and 5% using 8, 20, and 60 cameras.

the TR-FluidNeRF to the FluidNeRF trained on individual time steps. The TR-FluidNeRF reconstruction of the free jet for each time step using 15 cameras is displayed in Figure 7.14. TR-FluidNeRF is able to capture the movement of the leading vortex and the overall structure of the jet for all three-time steps. The laminar region of the jet has a smooth top hat profile that matches the ground truth. The reconstruction also captured the small vortex structures forming at the shear layer of the flow as the jet transitions to turbulence. Overall, the FluidNeRF reconstruction is able to adequately capture the spatial and temporal information of the jet flow at the three times used for the training/reconstruction.

7.2.1 Hyperparameter Investigation

The first hyperparameter of interest is the activation function. While ReLU was determined to be adequate for static FluidNeRF, the added complexity of time could benefit from using the more complex swish activation function. Figure 7.15 shows the quantitative comparison between TR-FluidNeRF using swish and ReLU activation functions. For both cases, the only difference between the reconstruction was the activation function. The added complexity of the

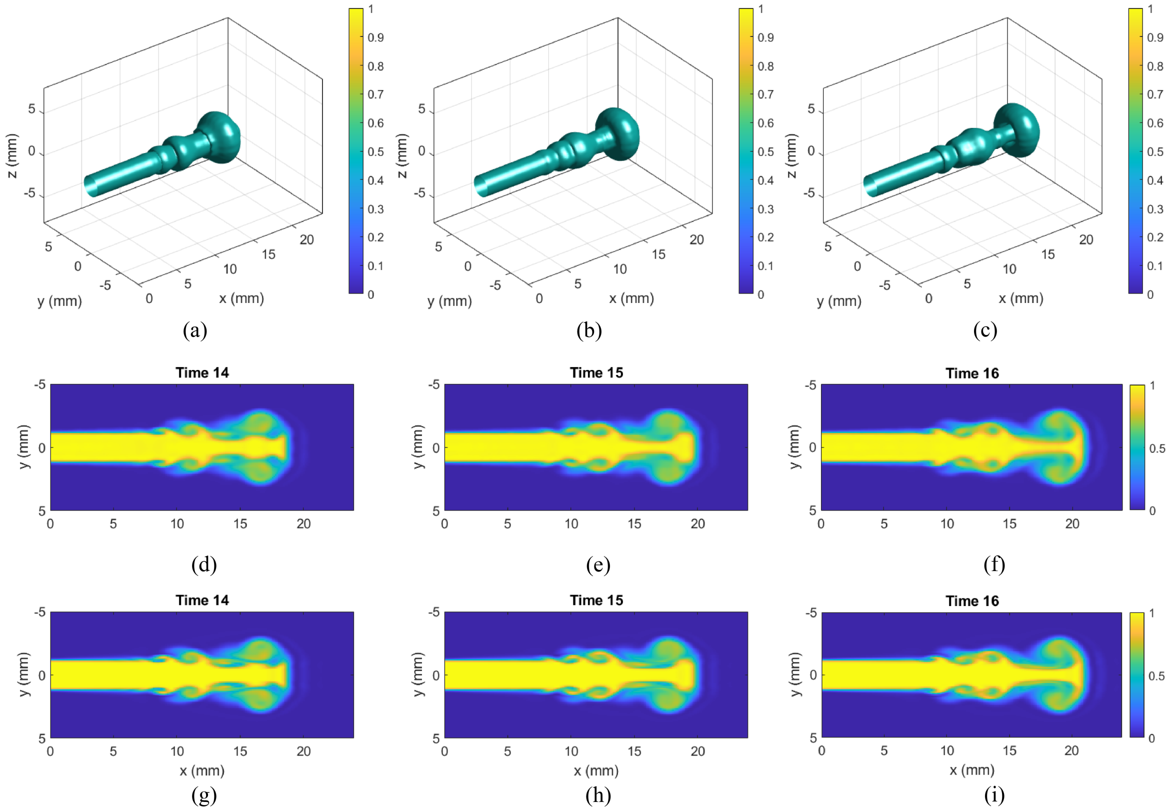


Figure 7.14: (a-c) Isocontour of the TR-FluidNeRF reconstruction at three different time steps. Central z-slice of the (d-f) reconstruction volume and (g-i) ground truth.

swish activation produces an increase in accuracy for TR-FluidNeRF compared to the results from Section 7.1.1. Additionally, the swish activation demonstrates a consistently balanced accuracy response across all time steps, implying an equal approximation across the three frames. Overall, the swish activation improves the neural implicit approximation for spatial-temporal coordinates.

Figure 7.16 displays the qualitative comparison between ReLU and swish. Both methods capture the general shape and vortex structures of the jet flow; however, ReLU has some blurring across the flow structures, causing the reconstruction to lose some details. The edges of the jet have higher errors for ReLU than swish, as best indicated in the difference between the truth slice and the truth slice. The accumulative errors in the swish activation function are lower than those found for ReLU. Therefore, the spatial-temporal representations of the flow should use swish activation for the increased complexity it adds to the NN. The following results for the TR-FluidNeRF method employ the swish activation function.

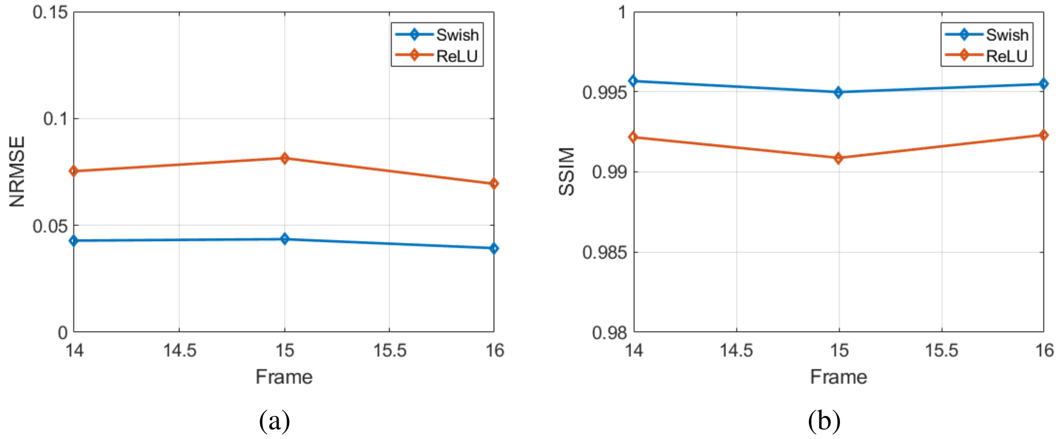


Figure 7.15: (a) NRMSE and (b) SSIM of the TR-FluidNeRF reconstructions employing swish and ReLU activation functions for three time steps.

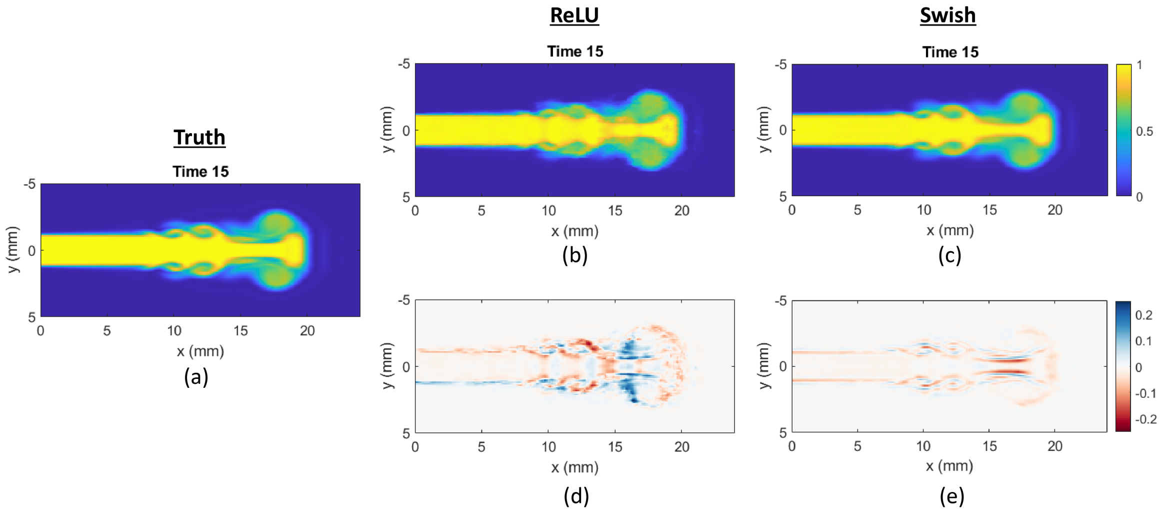
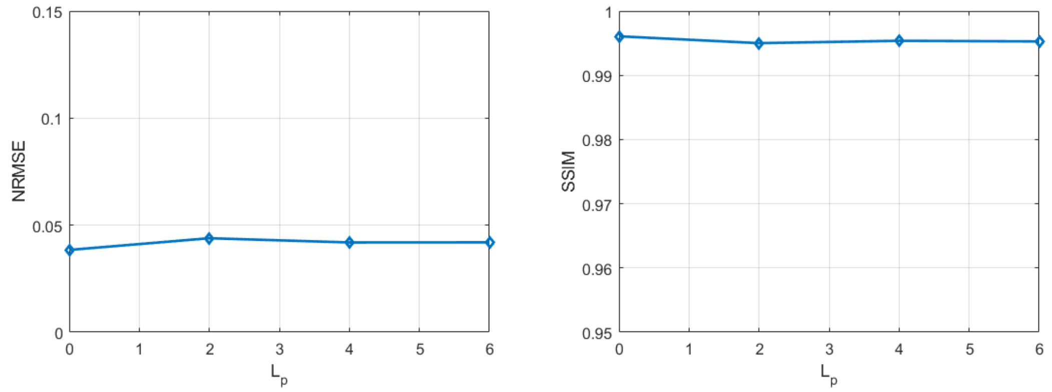


Figure 7.16: A Central z-slice of (a) the ground truth free jet and FluidNeRF reconstruction using the (b) ReLU and (c) swish activation function. The difference between the ground truth slice and the FluidNeRF reconstruction using (d) ReLU and (e) swish is provided.

Positional encoding was another crucial hyperparameter for the static FluidNeRF method, as demonstrated in Section 7.1.1. Figure 7.17 presents the accuracy metrics of the reconstructions using four different positional encoding values of $L_p = 0$, 2, 4, and 6. The results show that positional encoding has minimal influence on the reconstruction quality of the time-resolved passive scalar, as the accuracy of each reconstruction is essentially the same. This result could be a consequence of several factors, including the activation function and the spatial complexity of the flow for the current flow field of interest. The swish activation function, as shown in the previous result, does improve the approximation of the volume as compared to ReLU



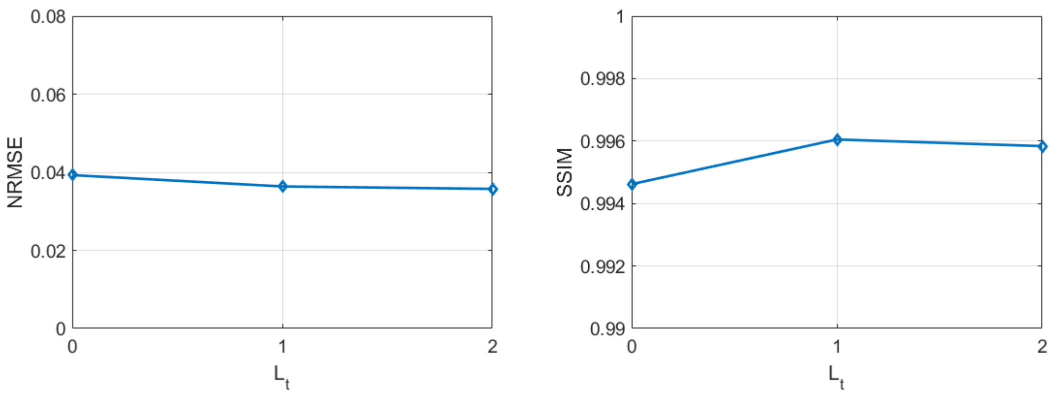
(a) (b)

Figure 7.17: Averaged (a) NRMSE and (b) SSIM of the TR-FluidNeRF reconstructions over three time-steps using L_p = 0, 2, 4, and 6.

because of the added complexity of the function. Therefore, positional encoding might have less impact on the method. Additionally, the flow field has less spatial complexity than the fully developed flow that was used for the static FluidNeRF synthetic experiments, which has three distinct regions and covers more of the field of view. Combining both effects could reduce the need for positional encoding for this synthetic dataset. However, future synthetic studies are vital in understanding the relationship between the complexity of different flows and positional encoding.

Next, we will consider temporal encoding. For the current three-time increments, temporal encoding is expected to have a more negligible effect since the temporal complexity is relatively limited. Figure 7.18 presents the NRMSE and SSIM of the TR-FluidNeRF reconstructions using L_t = 0, 1, and 2 averaged across the three-time steps. Similar to the results of positional encoding, L_t had limited effect on the reconstruction quality. The reconstruction accuracy improved slightly between $L_t = 0$ and $L_t = 1$, then leveled for $L_t = 2$. The minimal improvements of L_t were expected due to the limited temporal information for the three-time increments. The current work focuses on demonstrating the time-resolved FluidNeRF method, where a future study is required to understand the effect of temporal encoding for different time-resolved datasets. The rest of the results use $L_t = 1$ due to the slight increase in accuracy.

Network shape plays a crucial role in the performance and computation time of the time-resolved method. NRMSE and SSIM for three different heights and four different depths are

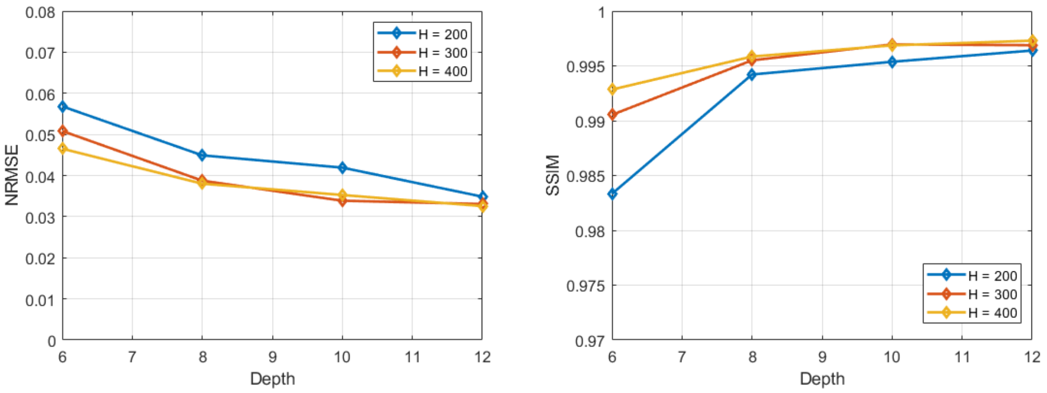


(a) (b)

Figure 7.18: (a) NRMSE and (b) SSIM of the TR-FluidNeRF reconstructions using L_t = 0, 1, and 3.

displayed in Figure 7.19. The depth of the network appears to have the most influence on the accuracy of the reconstruction. Height also improves the approximation of the volume, especially for shallower networks. However, as the depth is increased, the height has a reduced effect on the accuracy. The reconstruction accuracy had diminishing returns with increasing depth for every height case. Increasing depth also had diminishing returns, where each case started to level after ten layers. The propagation method better trains with deeper, thinner networks than shallower, taller ones.

Computation time is a crucial factor to consider alongside reconstruction accuracy, as the computational time is significantly influenced by the height of the network. Doubling the height of the network approximately doubled the iteration time, causing the convergence time to increase proportionally. This is due to the computational time of a fully-connected neural network being proportional to O(D*H). Let us consider two cases with the same number of hidden units in the neural network, a depth of 6 with a height of 400 and a depth of 12 with a height of 200. Both networks had similar iteration times, with the network with a depth of 12 having an iteration time roughly 10% quicker than the network with a depth of 6. Even with having the same number of hidden units, the network converges to a higher accuracy for the deeper network while taking comparable or slightly less time per iteration. Therefore, it is recommended to use a deeper, thinner network. The rest of the results use a depth of



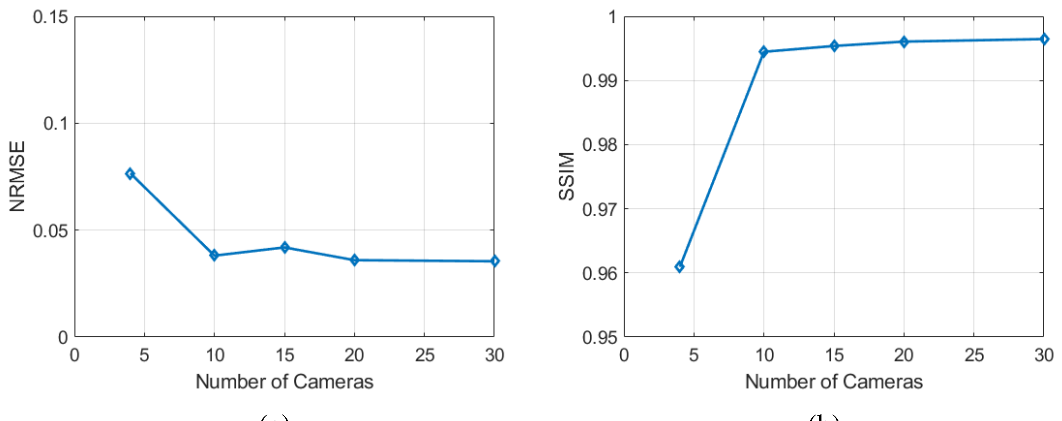
(a) (b)

Figure 7.19: (a) NRMSE and (b) SSIM of the TR-FluidNeRF reconstructions at four different depths and three different heights. The NRMSE and SSIM values are averaged over three time steps.

10 and a height of 200, where static FluidNeRF has a similar number of hidden units with a recommended height and depth of 256 and 8, respectively.

7.2.2 Camera Configuration

The subsequent evaluation is the effect of the number of cameras used for the reconstruction on the solution accuracy. Figure 7.20 shows the averaged metrics across the three-time steps for different subsets of cameras. For each case, the camera separation angles were optimized. As expected, the four-camera case performed the worst. Increasing the number of cameras does improve the accuracy in general, but the accuracy quickly approaches a constant value. The convergence to an accuracy with increasing number of cameras is contrary to the results that were shown in Section 7.1.1, where the accuracy was still noticeably increasing after 30 cameras. The TR-FluidNeRF has an additional time constraint on the reconstruction compared to static FluidNeRF, which only constrains the problem with an increased number of projections. The addition of time helps the available projections be related in time to improve the accuracy. Therefore, depending on the complexity of the flow, TR-FluidNeRF could require fewer cameras to produce reconstruction quality similar to static FluidNeRF. It should be noted that differences in flow complexity can affect these results.



(a)

(b)

Figure 7.20: Averaged (a) NRMSE and (b) SSIM of TR-FluidNeRF reconstructions over three different time steps utilizing five different subsets of cameras.

7.2.3 Static vs Time-Resolved FluidNeRF

Adding time to the reconstruction approximation increases the complexity of the implicit representation; however, time information can also constrain the problem implicitly since the neural network is approximating volume as a function of time. Since static FluidNeRF produces equivalent or superior reconstruction accuracy as an ART-based method, we will analyze TR-FluidNeRF compared to static FluidNeRF. Figure 7.21 presents a qualitative comparison of TR-FluidNeRF reconstruction to static FluidNeRF. The two methods produced similar quality of the major flow features, including vortex structures. The most notable differences between the two methods are illustrated in the plot, showing the disparity to the ground truth. TR-FluidNeRF struggles in the same areas as static FluidNeRF; however, the errors appear less for TR-FluidNeRF. The region with the highest error is the region of small vortex structures forming at the shear layer in the transition region.

The quantitative analysis between static and time-resolved FluidNeRF is illustrated in Figure 7.22. The findings from the qualitative analysis are confirmed as the time-resolved reconstruction has higher accuracy for all time steps than the static reconstructions. The reconstruction accuracy of TR-FluidNeRF exhibits uniformity throughout the time series, whereas the static case had lower accuracy for the middle reconstruction. This uniformity indicates that the time information improves the reconstruction by further constraining the inverse problem.

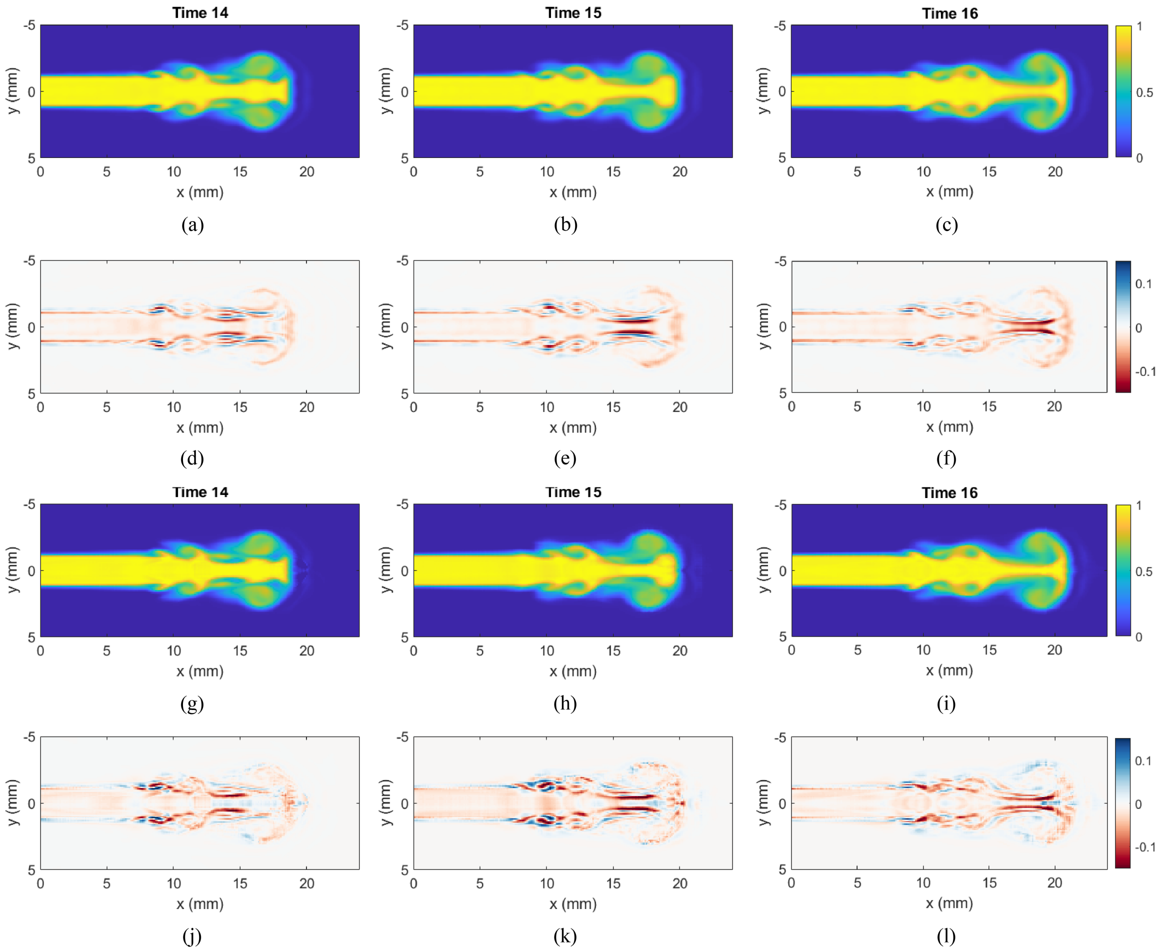
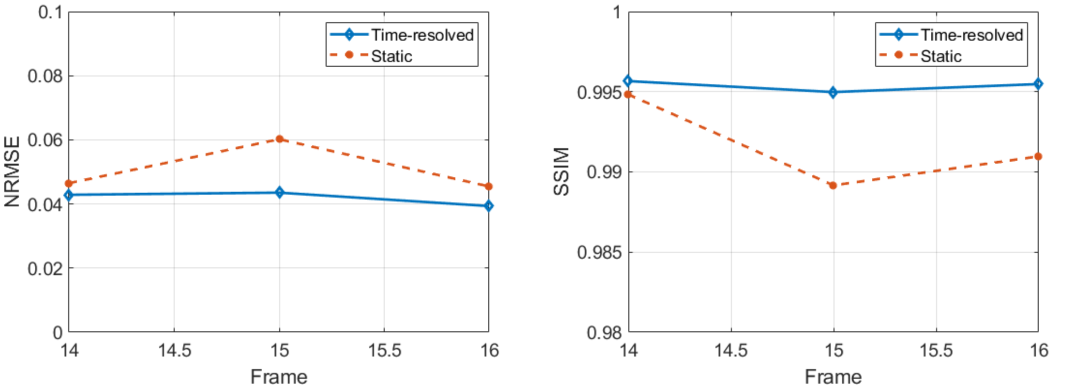


Figure 7.21: (a-c) A central z-slice of the TR-FluidNeRF reconstructions at three time steps, and (d-f) the difference between the TR-FluidNeRF reconstruction and the ground truth. (g-i) A central z-slice of the static FluidNeRF reconstructions at the three time steps, and (j-l) the difference between the static FluidNeRF reconstruction and the ground truth.

Further investigations are needed to understand the extent to which time can constrain the problem for different flow complexities. However, the current results suggest that TR-FluidNeRF produces higher accuracy than static FluidNeRF for time-resolved reconstructions.

7.3 Chapter Summary

In conclusion, a neural implicit representation tomography technique called FluidNeRF was introduced and compared to standard ART-based reconstruction methods. FluidNeRF's performance was demonstrated and characterized using synthetic images generated from a DNS volume of a turbulent jet, where it was found to perform comparably or better than ASART.



(a) (b)

Figure 7.22: (a) NRMSE and (b) SSIM of the TR-FluidNeRF compared to the instantaneous version of FluidNeRF for each time step.

FluidNeRF is similar to ART-based methods in that a volume is reconstructed iteratively by comparing reconstructed projects with measured projections; however, FluidNeRF is distinct in that it approximates the reconstructed volume as a continuous function using a neural network and that reconstruction is performed using standard machine learning training algorithms and volume rendering. Overall, the FluidNeRF technique has significant potential for 3D flow visualization, with this work being an essential first demonstration.

This work found that network size can impact reconstruction quality but also have an associated impact on computational time. Deep learning techniques have enabled the growth of network sizes to provide volume approximation fidelity that surpasses discretized methods. For the flow field investigated here, a deeper, thin network size has a negligible effect on reconstruction quality. A network with half the height and two layers deeper produced similar results while cutting computational time in half due to the reduction of computations per input and network unknowns. Positional encoding improved the ability of the network to capture high-frequency content in the volume. Increasing positional encoding maximizes reconstruction quality in regions with high-frequency content and preserves the quality in low-frequency areas, where an optimal encoding of $L = 8$ was found. Another critical step is the spatial sampling along a camera ray to maximize the image projection. A multi-resolution sampling technique enhances reconstruction accuracy for a given N_{tot}, although only slightly for the parameters considered here. In terms of camera layout, a planar configuration that is oriented perpendicular to the mean flow path outperformed the spherical configuration, emphasizing the importance of considering *a priori* knowledge of the flow during experimental set-up. Lastly, FluidNeRF proved more robust to noise than ASART, indicating that FluidNeRF could be better suited for experimental datasets.

Another version of FluidNeRF that represents the volume with spatial-temporal coordinates, TR-FluidNeRF, was evaluated with a DNS-generated time-resolved dataset. TR-FluidNeRF produced accurate reconstructions of a time-resolved scalar field across three-time increments. To the best of the author's knowledge, this method is the first to include time in the reconstruction model to help further constrain the problem. The time-resolved reconstruction was made possible due to the continuous spatial-temporal approximation of the volume through a neural network. The addition of time information further constrains the inverse tomography problem, where TR-FluidNeRF had comparable or increased accuracy and uniformity compared to static FluidNeRF. The authors would like to note that this is a limited study, requiring further investigations to understand the time constraint fully. In addition to these findings, the swish activation function improved the prediction accuracy for TR-FluidNeRF more than was found for the static FluidNeRF. Overall, TR-FluidNeRF is a viable option for reconstructing time-resolved tomography.

Chapter 8

Experimental Design

The set of experiments that are described in this chapter are developed to have a simple problem that represents typical flow diagnostic measurements based on the scalar field flow visualization. For the experiments, flow visualization is conducted on a free jet using smoke that is seeded in the flow and illuminated with a laser. The flow is composed of different-sized flow structures to examine the ability of FluidNeRF to capture experimental quantities of interest. The flow speeds of the jet were determined to ensure the images are time-resolved; therefore, this dataset can be used to test static and time-resolved FluidNeRF. The experiment can be modeled as a viscous, incompressible flow, where the incompressible Navier-Stokes and mass-transport equations can be utilized for physics constraints on TR-FluidNeRF.

The following sections present the experimental apparatus and methods. The first section describes the facility used to generate the jet flow. The second section discusses the miniature camera system developed to capture perspective views of the passive scalar in the flow. Then, we cover the flow visualization steps utilized for the jet experiments, including the volume calibration method used to determine the camera imaging model and 3D position around the volume. Finally, the last section covers the various experimental parameters and data processing steps used for the reconstruction.

8.1 Experimental Apparatus

The facility was originally designed to provide an open-air experiment with ample optical access. The experimental apparatus was composed of a tank filled with smoke, a nozzle, and a fan

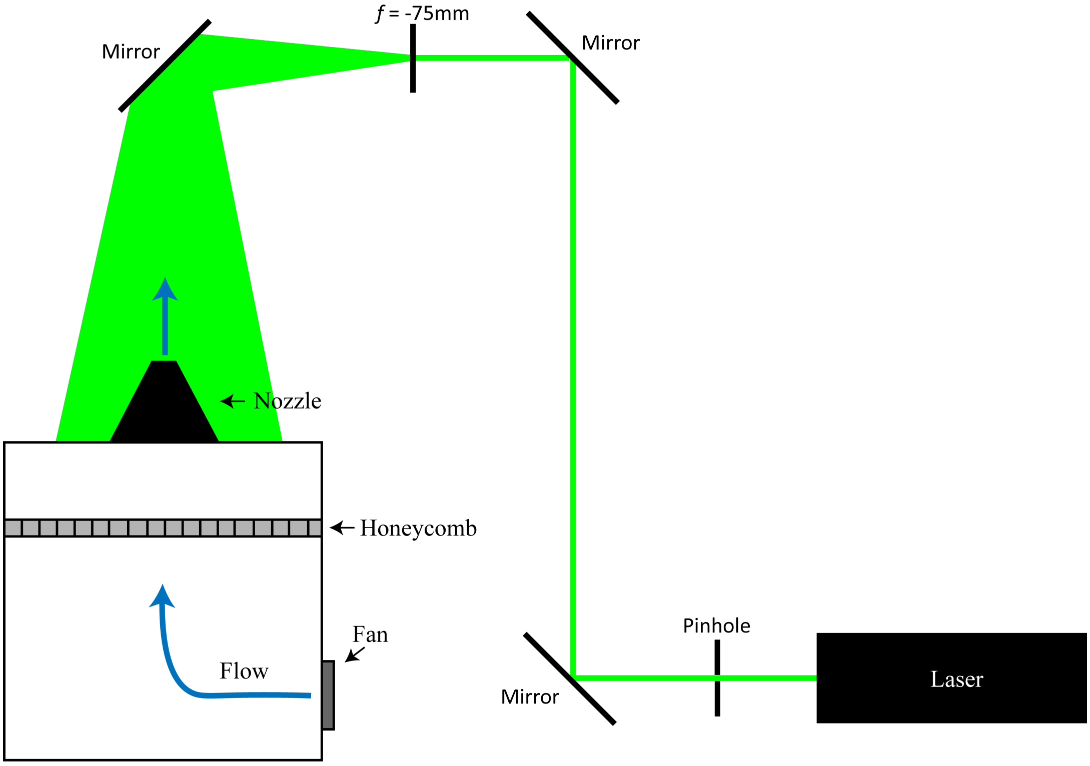


Figure 8.1: Schematic of the smoke chamber with the nozzle, honeycomb and fan locations indicated. The laser path is shown in green with each optical element labeled.

that increased pressure in the tank to push flow out the nozzle. A schematic of the low-speed jet facility is presented in Figure 8.1. The tank has a volume of approximately one cubic foot, with all tank walls composed of acrylic plates. An aluminum honeycomb plate is located approximately one inch from the top of the chamber to help straighten the flow before exhausting out a nozzle. The honeycomb structure has a thickness of half an inch with cell sizes of half an inch. Two conical nozzles were designed with an entrance diameter of 4.5 inches and an exit diameter of 1.5 inches and 1 inch, respectively. The nozzle was 3D printed using PLA. Air is forced through the system by a 40 mm Noctua computer fan with a maximum flow rate of 8.9 m^3/h. A PWM manual controller (NA-FC1) controls the speed of the fan. Smoke was generated using incense sticks that were located at the bottom of the tank, and the tank was filled with smoke between runs.

The jet flow is initiated and stopped using the PWM manual controller in an ad hoc manner. Therefore, the duration and timing of the fan operation has limited precision. For the experiments, the nozzle exit velocity was targeted to be approximately 0.25 m/s, which is a

Reynolds number of 620 for an exit diameter of 1.5 inches. This was calculated through mass conservation, which assumed a linear relationship between flow rate and PWM setting. To provide a more detailed understanding, the exit velocity was chosen to provide approximately 15 frames for flow to enter and exit the field of view. One set of images was taken to capture the passage of the leading vortex structure through the field of view. Another set of time-resolved images was taken when the flow has become fully-developed.

8.2 Flow Visualization: Smoke

Flow visualization was conducted by seeding the flow exiting the nozzle with smoke generated by incense sticks. The smoke was chosen as a passive scalar in the flow, which is typically used to understand mixing, mass transport, and visualization of flow features. The smoke was illuminated with a frequency-doubled Nd:YLF laser, where the smoke particulate scattered the light. The concentration of the smoke is important to the method, where high concentrations can make the jet optically thick, and low concentrations cause reduced SNR of the images. The concentration of the smoke was generated in an ad-hoc manner by blocking the openings in the tank and allowing the smoke smoke from the incense stick to diffuse in the chamber. When the density of the smoke was deemed qualitatively appropriate the entrances were uncovered and the experiment was initiated.

The multi-mode Nd:YLF laser that was utilized for this work was manufactured by Photonics Industries (Model No.: DM30-527DH). The laser produces a wavelength of 527 nm at a maximum pulse energy of 60 mJ. The laser has a pulse duration of 170 ns, which ensures a relatively instantaneous set of images. Figure 8.1 shows a 2D schematic of the laser path and optics used to illuminate the volume in the tank. The laser first passes through a pinhole to crop the elliptical shape into a circular shape, producing an approximate top-hat illumination distribution. Then, the laser is directed through a series of mirrors to have the laser aligned with the central axis of the flow from above. A circular laser volume is expanded using a convex spherical lens with a focal length of -75 mm. The lens produced a circular cross-section of approximately 3.5 inches in diameter at the exit of the nozzle. The author found that the laser

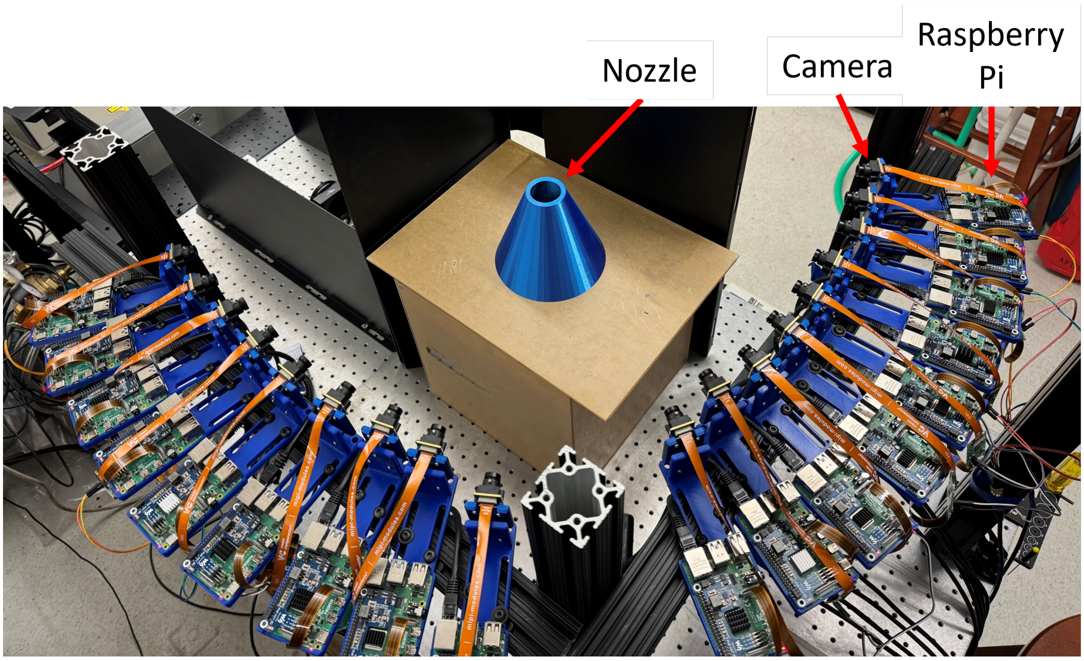


Figure 8.2: Miniature camera system layout around the nozzle that were secured using t-mount rail.

needed minimal pulse energies for adequate image SNR, which ensures the SNR of the images was not limited by laser energies.

8.3 Imaging System

Along with the development of FluidNeRF, the Advanced Flow Diagnostic Laboratory at Auburn University has been developing miniature camera systems that use commercially available board-mounted sensors to reduce the size and cost of multi-camera systems. The cameras were distributed around a plane perpendicular to the central nozzle axis, as revealed in Figure 8.2. For this work, the Vision Components OV9281 board-mounted image sensor was implemented, with the specifications of the sensor reported in Table 8.1. The OV9281 is a monochromatic CMOS sensor with a wavelength sensitivity range of 400 - 700 nm. When externally triggering the OV9281, the frame rate is limited to 60 fps. A critical feature of this sensor is the global shutter, which means all pixels are exposed simultaneously for the entire sensor integration time. The global shutter is important for flow diagnostics due to the flow's motion and the laser's pulse duration, where a rolling shutter could cause artifacts.

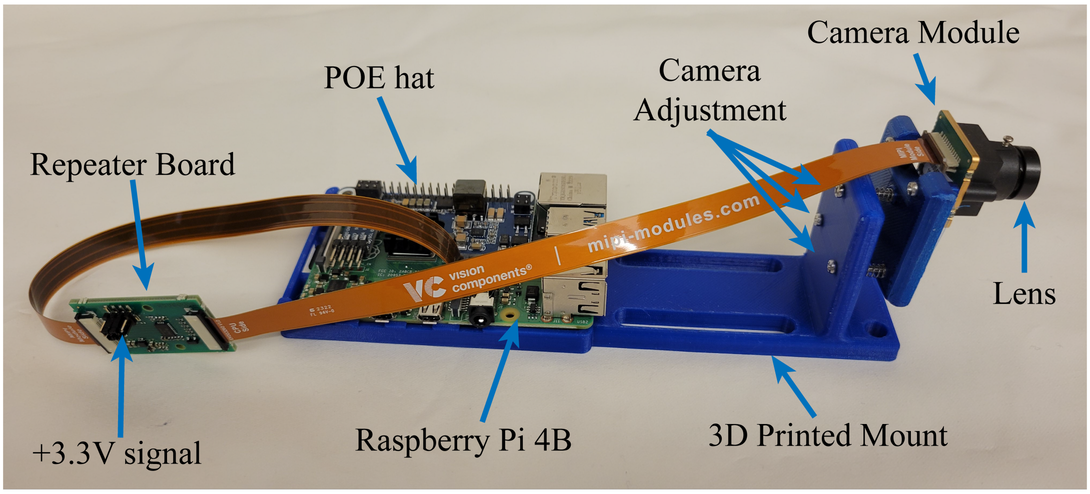


Figure 8.3: Image of a miniature camera system developed by AFDL with labeled components.

Table 8.1: Specifications of the Vision Components OV9281.

Parameter	**Value**	**Parameter**	**Value**
Sensor Type	CMOS	Sensor Size	1/4”
Resolution	1280×800	Pixel Size	3.00 μm
Shutter Type	Global	ADC Resolution	8 or 10 bit
Max FPS	120 fps (60 fps in trigger mode)	SNR	38 dB

Figure 8.3 presents an assembled camera system utilized for the experiments with labeled components. Each camera sensor is powered and operated through a Raspberry Pi 4B. A Vision Components repeater board is connected between the camera sensor and the Raspberry Pi 4B to provide the ability to hardware trigger the sensor externally. The camera was held in a custom 3D printed mount that held both the camera module and the Raspberry Pi. The camera mount provided three screw/spring actuators for fine adjustments of the viewing angle. To minimize the amount of cables required to power and control the Raspberry Pi, a Power over Ethernet (POE) HAT was included to allow communication and power through the Ethernet port. Not including the objective lens and other optical components, the camera system costs less than $300 per camera. The reduced cost and size of the imaging system allow for large-scale multi-camera systems, unlike earlier work that use beamsplitters and fiber bundles to minimize the number of camera sensors.

The Vision Components OV9281 board has the ability to use c-mount and s-mount lenses. The c-mount lenses have more flexibility, including focus and aperture; however, that flexibility comes at a cost. C-mount lenses are also larger in size, increasing the footprint of the system.

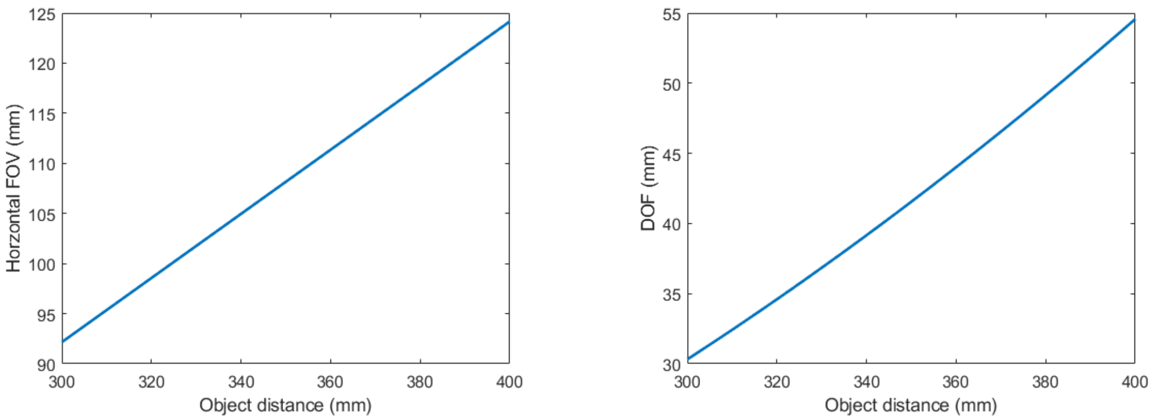
Therefore, s-mount lenses were utilized for our cameras. When determining the lenses, there are several considerations, including working distance, magnification (FOV), distortion, depth-of-field, and wavelength range. Most of these optical design parameters are correlated, as shown by Eqn. 6.2 and Eqn. 6.3.

For the jet experiment, the imaging lens for each camera was the Arducam (M1812ZH03S) 12 mm focal length M12 lens with a f/# of 2.8. The lens was selected to deliver a working distance between 300 mm and 400 mm, a horizontal FOV greater than 75 mm, and a depth-of-field (DOF) greater than 40 mm. Figure 8.4 presents the horizontal FOV and DOF for the 12 mm focal length Arducam lens for working distances of 300 - 400 mm. This result shows that the lens provides adequate FOV; however, it should be noted that too large FOV should be avoided because it reduces the spatial resolution of the QoI in the images. The DOF is not a hard limitation but rather a measure of the gradual blurring with distance. The DOF can be calculated using

$$\text{DOF} = \frac{d_a \cdot s_o}{d_a - c_o} - \frac{d_a \cdot s_o}{d_a + c_o}, \tag{8.1}$$

where c_o is the circle-of-confusion or the pixel size in object space, d_a is the aperture diameter, and s_o is the object distance. For the DOF in Figure 8.4b, c_o was approximated as 3 pixels. This assumption allows for up to 3 pixels worth of blur. Reducing d_a can increase the DOF but also reduce the light throughput. It should be noted that the current system has reduced resolutions due to blurring for regions away from the focal plane. However, these regions are different for each camera.

The signal for the cameras and laser is generated in LabVIEW with the frame rate being set by the frequency of the pulses. Figure 8.5 shows the communication connections for the camera and laser. The LabVIEW signal is sent to a signal generator as the external trigger source. One channel of the signal generator is then sent to the laser, while the other three are used to trigger the cameras as indicated by the red connections. The delay between the laser and the cameras is set through the signal generator to ensure the laser is illuminated during the exposure time of each camera. The cameras are synchronized by sending the signal to the header pins of each



(a) (b)

Figure 8.4: (a) Horizontal field of view and (b) depth-of-field for various working distances with the 12 mm focal length lens paired with the OV9281 sensor.

repeater board (Figure 8.3). Since only three ports are available for the cameras, the signal is split into six paths per signal generator port before connecting to each repeater board. A secondary cable was needed to adapt the BNC cable to header pins of the repeater board. The appropriate delay between the cameras and the laser signals was determined by filling the FOV with smoke and systematically changing the delay until the laser illumination was captured in the camera images. The cameras were controlled by a master computer through a network switch. The network switch provides PoE and connection to all the cameras as indicated by the black connection in Figure 8.5. The management of all the cameras is conducted through a custom MATLAB graphical user interface on the master computer.

8.4 Volume Calibration Method

A critical step of tomographic reconstruction is accurately mapping the object volume domain (3D) to the image space domain (2D). This mapping provides a projection function for the imaging model of FluidNeRF, which is required for the inverse problem. The reconstruction accuracy is closely coupled with the accuracy of the camera calibration method. There are several camera calibration techniques and models that are employed in computer vision applications, where an extensive overview is covered by Hartley and Zisserman [223]. The complexity

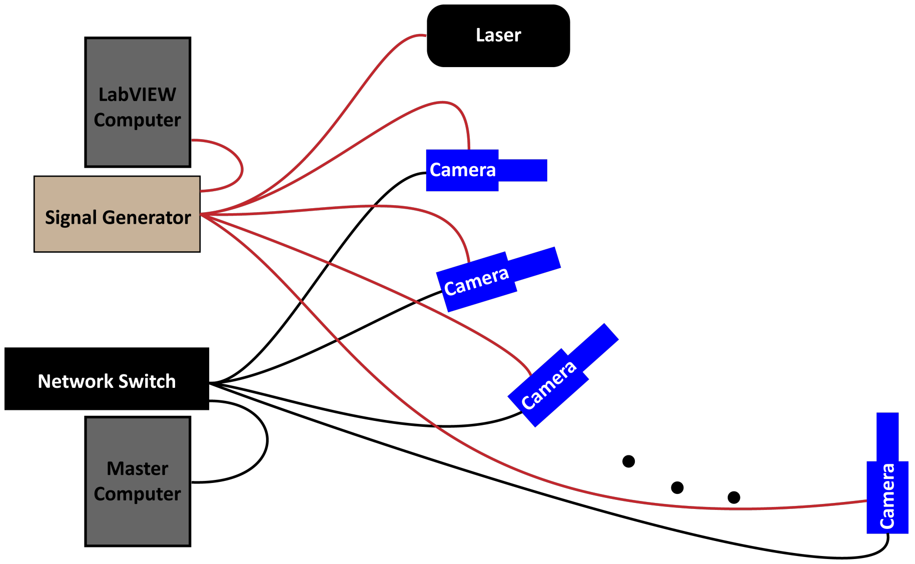


Figure 8.5: Schematic of the connections required for controlling the cameras and laser system. The system is composed of a network switch, master computer, and signal generator. The black wires indicate the Ethernet connections and red is 3.3V signal sent via BNC cables and header pins. LabVIEW dictated the timing of the images and laser

of the models varies widely, with the more complex models being able to account for distortions and other experimental aberrations that can occur. The most common imaging model is the pinhole camera with a nonlinear distortion correction that accounts for non-optimal lenses and mounting fixtures. Another adopted imaging model for tomography is the non-physical polynomial function that directly relates object space points to image points, which is useful when there are complex lens systems like for Plenoptic cameras [224]. Both of these methods have similar performance with traditional cameras and lens configuration, when the cameras can capture many features/points across both image and object space [1].

For our method, we employed a pinhole camera model with distortion because of the ability to directly trace rays from image space pixels to object space. The pinhole model is one of the simpler models, where the camera lens is represented as an infinitesimally small opening. Since light can only pass through the pinhole hole, the light captured by the sensor originates from the same location (pinhole). Thus, a ray tracing scheme can be generated with the rays originating from the same point corresponding to the pinhole. The camera parameters are split into two regions: i) "intrinsic" parameters that describe the image side of the pinhole and sensor and ii) "extrinsic" parameters that specify the pose of the camera (i.e., the camera location and orientation in world coordinates). The intrinsic parameters are composed of the camera focal length (f_u, f_v), the optical center (u_0, v_0), and the skew parameter (s). The focal length is the distance from the pinhole to the sensor. The optical center describes the location where the central axis of the pinhole aligns with the sensor and the skew parameter accounts for non-perpendicular image axes. The intrinsic parameters are generally defined in pixel dimensions. The extrinsic parameters are composed of a 3 × 3 rotation matrix ($\mathbf{R}$) and a 3 × 1 translation vector ($\mathbf{t}$). The extrinsic parameters translate and rotate world coordinates into the camera frame of reference. The intrinsic parameters can also be formed into a matrix ($\mathbf{K}$) that translates the relative world coordinates to the sensor. The intrinsic matrix is defined as

$$\mathbf{K} = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8.2}$$

The intrinsic and extrinsic matrices can be combined to form an overall camera matrix ($\mathbf{P}$) as shown in Eqn. 8.3

$$\mathbf{P} = \mathbf{K}[\mathbf{Rt}]. \tag{8.3}$$

The pinhole model does not consider a finite aperture or finite-sized optics, where these lenses and adapters can distort the image. The most common approach for addressing distortions is using a polynomial distortion model that incorporates radial and tangential distortions, as proposed by Brown [225]. The polynomials are non-physical models that relate undistorted pixel locations (u, v) to normalized image coordinates (u_{dist}, v_{ist}) for each distortion independently. The radially distorted pixel locations are determined using

$$u_{dist} = u\left(1 + \sum_{d=1}^{3} k_d r^{2k}\right) \tag{8.4}$$

$$v_{dist} = v\left(1 + \sum_{d=1}^{3} k_d r^{2k}\right), \tag{8.5}$$

where k_d are the radial distortion parameters and $r^2 = u^2 + v^2$. The radial distortion for this work is represented as a third-order polynomial but can be expanded as required. The tangential distortion is calculated from

$$u_{dist} = u + [2 * p_1 * u * v + p_2 * (r^2 + 2 * u^2)] \tag{8.6}$$

$$v_{dist} = v + [2 * p_1 * u * v + p_2 * (r^2 + 2 * v^2)], \tag{8.7}$$

where p_1 and p_2 are the tangential distortion coefficients.

The camera calibration process consists of solving for the unknown parameters of the intrinsic and extrinsic matrices and the distortion coefficients using a set of known real-world locations ($\tilde{\mathbf{X}}$) that correspond to sensor positions ($\tilde{\mathbf{U}}$). The world points, $\tilde{\mathbf{X}}$, can either be absolute locations or relative to a calibration target. The feature locations of the calibration target are determined in the images using dot or edge-finding algorithms. The target must have well-characterized features/points that can be identified in an image. The method that is used will determine the most appropriate target, including if every camera needs to view the same target points. Solving $\tilde{\mathbf{U}} = \mathbf{P}\tilde{\mathbf{X}}$ with matrix manipulation is not possible due to each pair of world and sensor points having an unknown and potentially unique scaling coefficient. Therefore, several nonlinear optimization techniques have been employed to solve the unknown parameters. For our method, we employ a two-step process using Zhang's method [226] and the Perspective-n-point (PnP) algorithm [227]. The two-step process helps relax the need for all cameras to image the same points, which is essential for our configuration, where the camera system is distributed around a wide arc.

For the first calibration step, we conduct Zhang's method, which supplies the intrinsic parameters of each camera. The technique calculates intrinsic parameters by capturing images of a planar calibration plate at random orientations and positions. The calibration is based on a target-specific coordinate system, where the z-component is assumed to be zero since the target is planar. The x and y coordinate system is determined by the size and spacing of features on the target plate. A homography matrix is formed to relate the planar target coordinates to the planar image coordinates using points on the plate. The homography matrix is a simplified version of the projection function (Eqn. 8.3, where z is assumed zero. A homography matrix must be formed for each image captured of the calibration target. Each homography matrix provides a unique rotation matrix and translation vector. Since the intrinsic matrix does not change between each image, the unknowns of the intrinsic matrix can be found for each homography matrix. Distortion parameters can also be calculated from the different equations that can be formed from each unique extrinsic matrix. An adequate number of images and target features are required to achieve a good approximation of intrinsic and extrinsic properties. A more in-depth review of Zhang's method can be found in [1].

Since the extrinsic matrices are only relative to the target frame of reference, our method acquires the intrinsic matrix and distortion parameters through Zhang's method. Fortunately, Zhang's method has been widely used in the computer vision and tomography communities [1]; therefore, it is a widely available code through the OpenCV library. Our implementation uses the camera calibration app through MATLAB. Before the experiments were conducted, we captured 30-40 images of a checkerboard calibration plate independently of each camera. Our checkerboard has a check size of 5 mm. Even with partial checkerboard detection built into MATLAB, it is recommended to keep the checkerboard within the field of view of each camera for the best results.

Now that we have the intrinsic parameters of each camera, the cameras must be determined in terms of global world coordinates. One way to determine each camera's global pose is to select a reference camera and then relate the other cameras to the camera using the planar calibration from Zhang's method. However, doing this can cause increasing uncertainty, primarily when some views cannot be directly related to the reference camera. Additionally, cameras that capture the plate at harsh angles have higher uncertainty in determining feature positions in the image. Therefore, we employ a cylindrical calibration target so that each view can capture the target simultaneously, and each image will have features that are approximately perpendicular to the camera. The unique global points on the calibration cylinder are specified using a ChArUco pattern affixed to the cylinder as first demonstrated by Zhang [27]. ChArUco combines a chessboard and an ArUco pattern, which is similar to a QR code. The ArUco pattern provides the ability to uniquely identify each code and, thus, each check location. The ChArUco check locations on the images are calculated using the Python implementation of

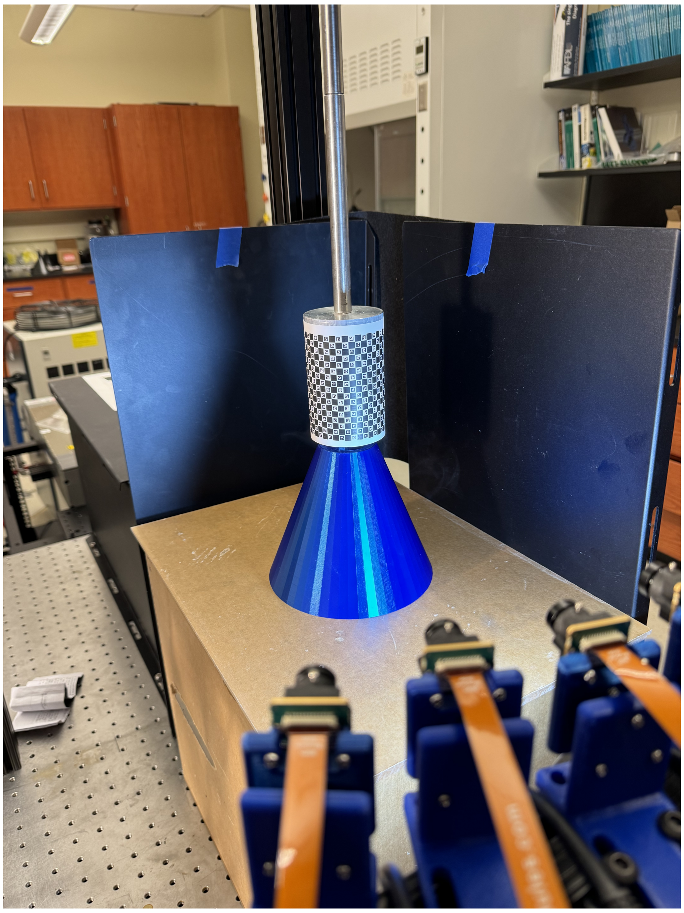


Figure 8.6: The cylindrical ChArUco calibration target that was utilized to determine the extrinsic calibration parameters with respect to the nozzle.

OpenCV. Once each camera captures the global points on the cylinder, the extrinsic parameters are independently approximated using the Perspective-N-Point (PNP) algorithm [227]. This algorithm is again readily available through OpenCV, where we utilized the MATLAB implementation.

The cylindrical ChArUco calibration target placed over the nozzle is shown in Figure 8.6. The cylinder had a diameter of 2.5 inches. Figure 8.7 presents the pattern that was wrapped around the cylinder. The ChAruCo calibration target had a check size of 5 mm, a marker size of 3 mm, and an ArUco dictionary of 5x5. For specifying the world coordinates, the top left checkerboard point corresponded to x = 0 and y = 0, with the y-axis aligned with the center of the cylinder. The x-z coordinates on the surface of the cylinder can be found using the circumference of the cylinder. Our method averages 10 images of the calibration cylinder to

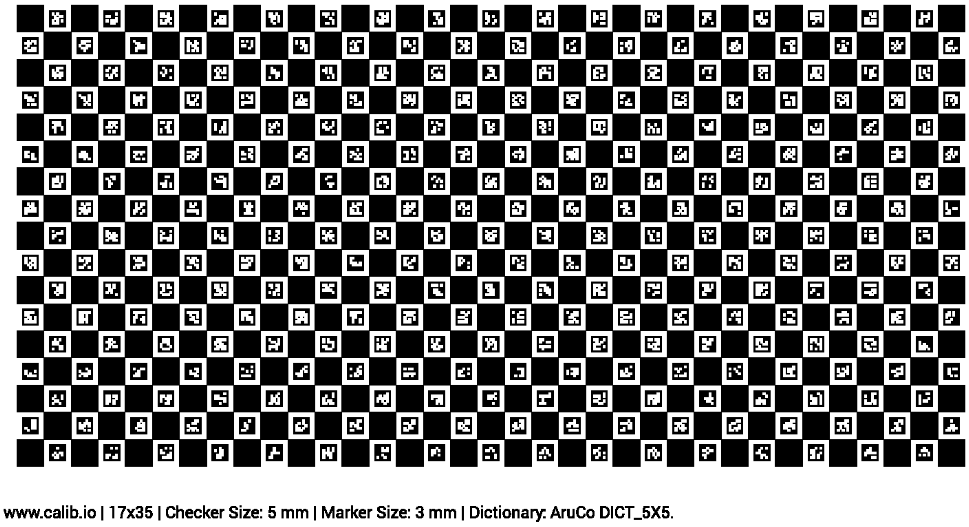

www.calib.io | 17x35 | Checker Size: 5 mm | Marker Size: 3 mm | Dictionary: AruCo DICT_5X5.

Figure 8.7: The ChArUco target that was wrapped around the cylinder for calibration. The pattern was generated using the website calib.io with the specified parameters below the ChArUco pattern.

minimize the effects of noise. The central axis of the cylinder was roughly colocated with the central axis of the nozzle.

8.5 Experimental procedures

Table 8.2 presents the combinations of nozzle diameter and fan settings that were used for different experimental runs. For each of these configurations, the system acquired 40 frames at a frame rate of 60 Hz. This frame rate provides time-resolved images for the designed flow speeds. The flow rates for the 1.5-inch nozzle with PWM settings of 45% and 60% produced exit velocities (Reynolds numbers) of approximately 0.98 m/s (2427) and 1.3 m/s (3220), respectively. The 1-inch nozzle with the PWM power set to 45% and 30%, produced exit velocities of roughly 2.2 m/s and 1.5 m/s, respectively. The Reynolds numbers for the 2.2 m/s and 1.5 m/s cases are 3630 and 2477, respectively. These were calculated assuming a linear response to flow rate with PWM relative to the maximum flow rate provided by the fan manufacturer. While these are higher than the targeted 0.25 m/s, the cameras were positioned such that the bottom of the FOV was approximately 30 mm above the nozzle. Therefore, the bottom h/D for the 1.5-inch and 1-inch diameter nozzles were roughly 0.79 and 1.18, respectively. Thus, the mean flow speeds decrease as the flow moves downstream of the nozzle. The FOV was positioned to view the development of vortex structures in the shear layer of the jet. The magnification at the focal plane of the cameras was approximately 0.033, producing a vertical FOV of 73 mm. Thus, the cameras viewed up to an h/D of roughly 4.1 and 2.7 for the 1-inch and 1.5-inch nozzles, respectively. The FOV varies slightly with the position of each camera.

Nozzle Diameter	**Fan PWM**	**Laser Current**	**Frame Rate**
1.5 inch	45%	20 A	60 Hz
1.5 inch	45%	19 A	60 Hz
1.5 inch	60%	19 A	60 Hz
1 inch	45%	19 A	60 Hz
1 inch	30%	19 A	60 Hz
1 inch	45%	20 A	60 Hz

Table 8.2: Experimental parameters used for smoke visualization of a free jet.

FluidNeRF was used to reconstruct the volumes, as provided in the next chapter, using the experimental images captured of the jet. The default hyperparameters used for static FluidNeRF reconstructions are shown in Table 8.3, unless noted otherwise. Each experimental image undergoes background subtraction to remove dark current from the images. Then, the images are undistorted, given the distortion parameters found during camera calibration. During the experiments, there was an angular intensity and polarization dependence on the laser illumination. Therefore, the image intensities were balanced to have the same total intensity for each image. Note that this could be accounted for by including view direction into the FluidNeRF model similar to NeRF [6]. For the time-resolved FluidNeRF reconstructions, the default hyperparameters are shown in Table 8.4. Only a few cases from the test matrix shown in Table 8.2 are presented in the next chapter.

Table 8.3: Default hyperparameters of static FluidNeRF for processing the smoke experiments.

Parameter	**Value**	**Parameter**	**Value**
Depth	8	N_c	64
Height	256	N_f	64
L_p	2	$N_{pix/img}$	1024
Activation	Swish		

Table 8.4: Default hyperparameters of TR-FluidNeRF for processing the smoke experiments.

Parameter	**Value**	**Parameter**	**Value**
Depth	10	N_c	64
Height	256	N_f	64
L_p	2	$N_{pix/img}$	1024
L_t	1	Activation	Swish

8.6 Chapter Summary

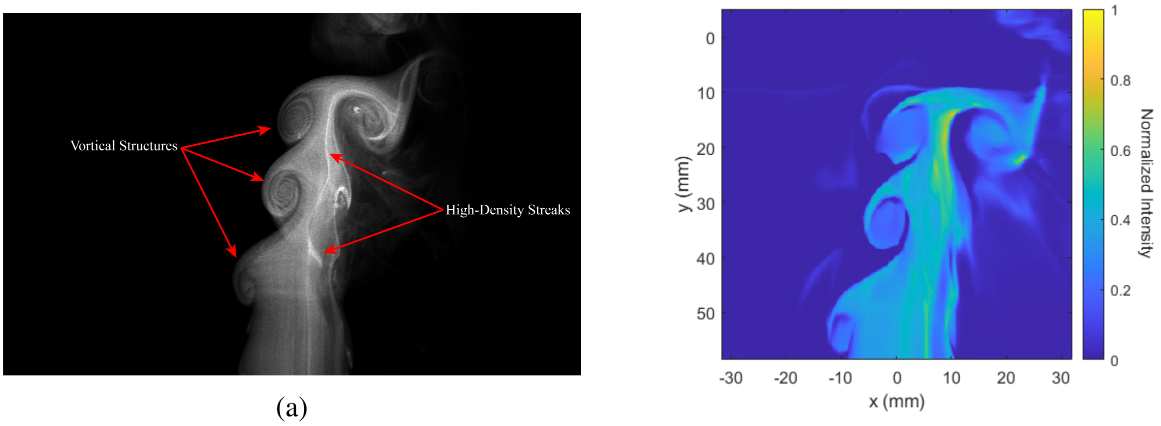
This chapter covers the design and procedures used to conduct 3D flow visualization of a subsonic flow from two different-sized nozzles. The flow was imaged using 18 different cameras, where smoke was entrained in the flow and illuminated by a Nd:YLF laser. These experiments were designed to demonstrate the capabilities of FluidNeRF for experimental datasets where image noise and imperfect camera calibrations are present. These experiments represent volumetric flow measurements performed using passive scalars, such as laser-induced fluorescence. The chapter started with covering the design of the experimental apparatus, flow visualization technique, and imaging system. The laser that was used to illuminate the smoke was hardware synced with the cameras to have the laser fired during the exposure time of all the cameras. The camera system was developed in-house using board-mounted cameras paired with Raspberry Pi 4Bs to control the camera. After covering the experimental setup, the volumetric calibration techniques were introduced. The calibration technique utilized a two-step process. The first step calculates the intrinsic parameters of each camera independently using Zhang's method, and then the extrinsic world pose is calculated for each camera using the Perspective-n-Point method. To have global coordinates for the extrinsic world pose, a cylinder with a ChArUco calibration pattern was employed to ensure each camera could view unique features in world coordinates. Finally, we discuss the experimental procedures conducted. The following chapter covers the volumetric reconstruction results of the experiments laid out in this chapter.

Chapter 9

Experimental Results and Discussion

Multiple line-of-sight images of a smoke-filled free jet were captured as discussed in Chapter 8. While the synthetic results revealed the advantages of FluidNeRF, it is vital to demonstrate the method with real-world experimental images. Real projections encompass noise and camera calibration uncertainty that are hard to fully model with synthetic generation. Therefore, it is necessary to evaluate the method for experimental datasets. One way to examine the quality of the reconstruction is to compare rendered to captured views that were not included during reconstruction, which is referred to as the validation set in the machine learning community. The captured images provide a ground truth for the volumetric reconstructions. However, it should be noted that several solutions can produce the same LoS image due to the ill-posed nature of the problem.

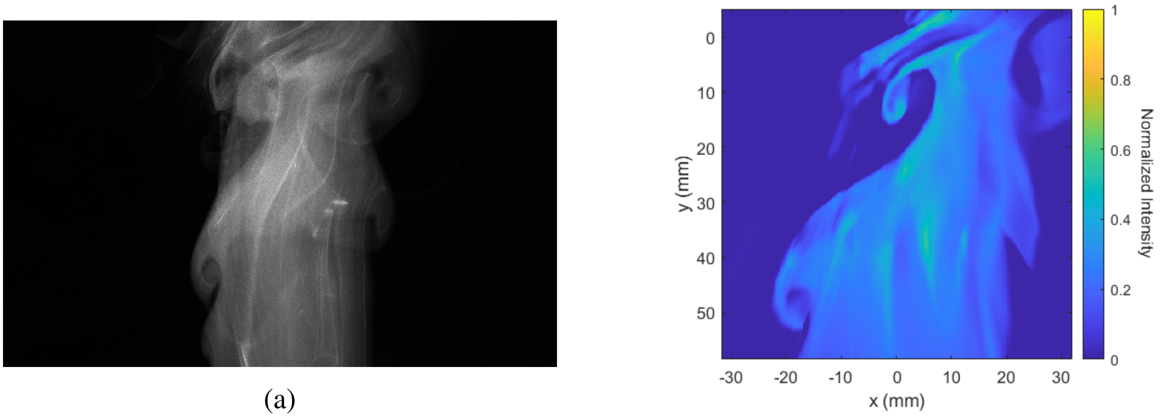
This chapter explores the reconstruction quality of static and time-resolved FluidNeRF for flow visualization of a passive scalar field in a free jet. First, the reconstructions will be qualitatively analyzed for the two different nozzle sizes for the static FluidNeRF method. Then, the effect of hyperparameters on reconstruction quality will be evaluated by using images that were not included in the reconstruction as the ground truth. We will also compare the TR-FluidNeRF method with the static FluidNeRF method, where we will additionally assess how the number of cameras affects reconstruction quality in each approach. Finally, the number of time steps included in thet time-resolved reconstruction will be evaluated. These results should provide a baseline using static and TR-FluidNeRF for experimental datasets.



(a)

(b)

Figure 9.1: (a) Experimental image of the jet filled with smoke originating from the 1-inch diameter nozzle. The static FluidNeRF reconstruction is demonstrated by the (b) central x-y slice. The reconstruction values were normalized by the maximum intensity in the volume. Note that the captured image was flipped to match the axis of the x-y slice.



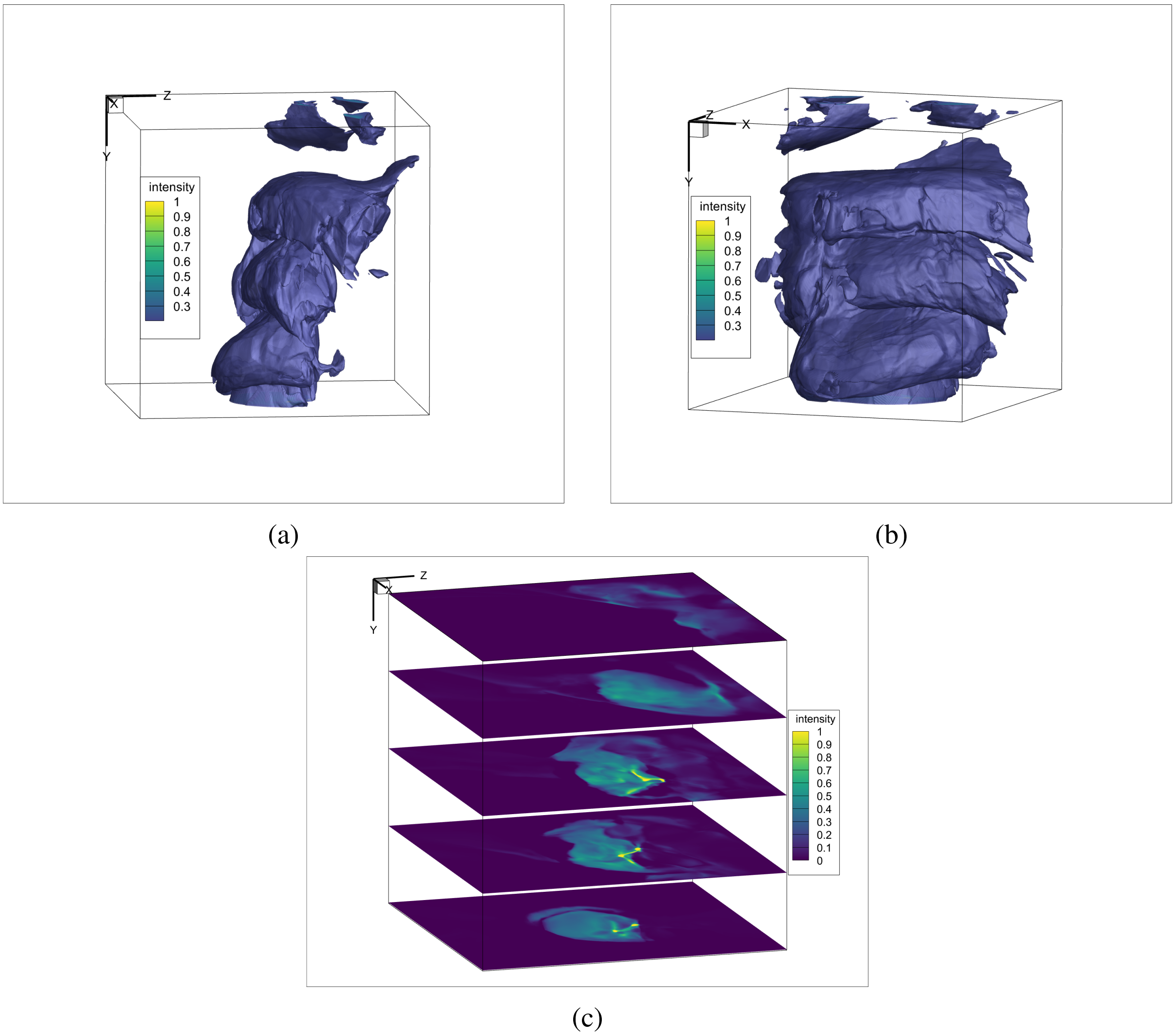
(a)

(b)

Figure 9.2: (a) Experimental image of the smoke filled jet originating from the 1.5-inch diameter. The static FluidNeRF reconstruction is presented by the (b) central x-y slice. The reconstruction values were normalized by the maximum intensity in the volume. Note that the captured image was flipped to match the axis of the x-y slice.

9.1 Static FluidNeRF

An example of an experimental image of the smoke-filled flow from a 1-inch and 1.5-inch diameter nozzle is presented in Figure 9.1a and Figure 9.2a, respectively. The images effectively capture the region of the jet flow where vortex roll-up occurs due to the shear layer. Within the vortical structures, layers of smoke swirls are discernible, each separated by lower-density regions of only a few pixels. Particularly noticeable are the higher density and thin lines of



(a)

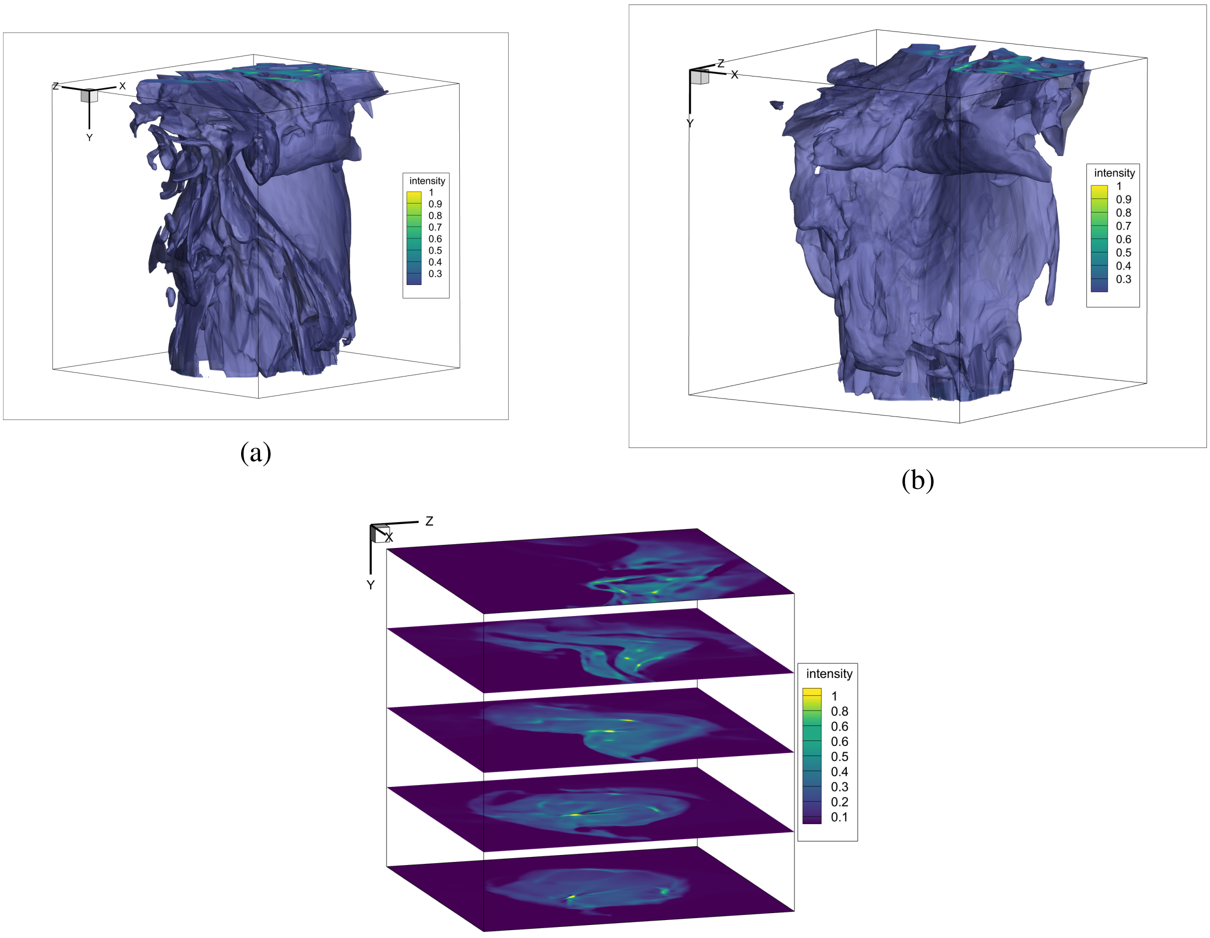
(b)

(c)

Figure 9.3: (a-b) Isosurface of the smoke-filled 1-inch diameter jet at two different views with a contour value of 0.2, and (c) x-z slices at different depths along the central nozzle axis. The reconstructed values were normalized by the maximum intensity in the volume.

smoke in the core, which contribute to high-intensity vertical streaks in the jet, which might pose challenges in reconstruction. The slice through the volume Given the complexities of the flow at this snapshot, depicted in Figure 9.1a for the 1-inch diameter case, this frame will serve as a focal point for evaluating the hyperparameters of static FluidNeRF.

The central x-y slice and isosurface of the 1-inch diameter nozzle volume reconstructed by static FluidNeRF with an activation function of ReLU and $L_p = 0$ are displayed in Figure 9.1. The overall structure of the reconstructed volume matches the perspective image, with multiple vortices forming at the shear layer. Note that the features depicted in the projection are accumulated along the LoS; therefore, comparing small details between the projection and



(a)

(b)

(c)

Figure 9.4: (a-b) Isosurface of the smoke-filled 1.5-inch diameter jet at two different views with a contour value of 0.2, and (c) x-z slices at different depths along the central nozzle axis. The reconstructed values were normalized by the maximum intensity in the volume.

a slice through the volume is not practical. A slice through the volume that aligns with the projection indicates that the general shape and features are present. Figure 9.2 presents a central x-y slice of the reconstructed volume from the 1.5-inch diameter. The general structure of the jet matches the perspective.

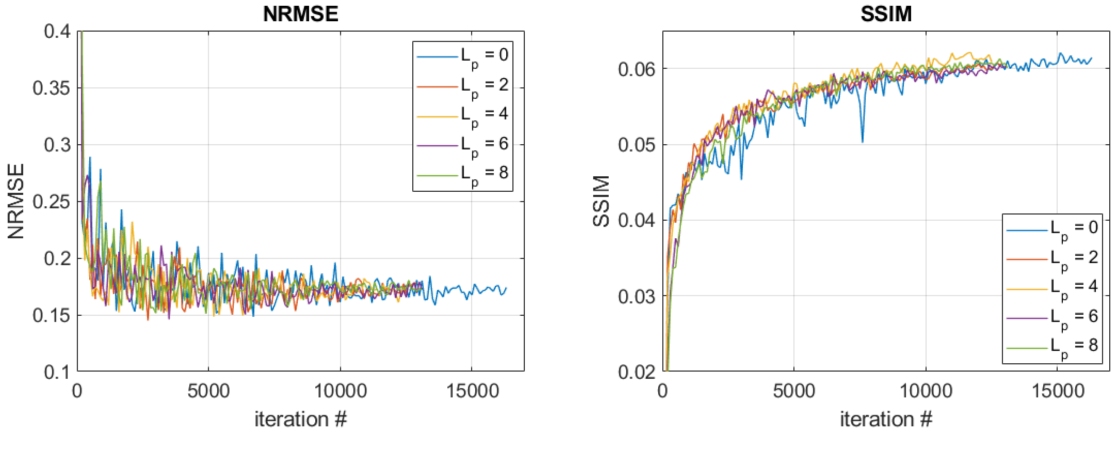
Figure 9.3 and 9.4 presents the views of an isosurface and x-z slices of the volume for the 1-inch and 1.5-inch nozzles, respectively. The diameter of the isosurface and x-y slice at the base of the 1-inch nozzle reconstruction was slightly smaller than the exit diameter of the nozzle, where the jet is circular at the bottom and expands in the x-direction more than the z-direction. Alternatively, the x-y slice of 1.5-inch jet volume has a slightly wider diameter than the nozzle exit. The x-z slices of the 1.5-inch diameter jet in Figure 9.4c show a similar deformation of the circular jet structure as the flow moves away from the nozzle as was noted for the 1-inch diameter nozzle. The experimental images of the 1.5-inch diameter nozzle were closer to fully developed flow as compared to the 1-inch diameter nozzle, where the initial vortex is transitioning the FOV.

The higher intensity vertical streaks in the jet are captured in the reconstruction, as labeled in Figure 9.1a. The higher intensity features are best presented in the x-z slices for both nozzle sizes (Figure 9.3c & Figure 9.4c). These features do not appear as narrow as presented in the perspective images, where some features in the projection are a couple of pixels in width. Although, some of the high-intensity streaks could be generated by overlapping features along the LoS. Static FluidNeRF lacks resolution in the high-intensity swirl of the vortical structures formed in the shear layer for the 1-inch diameter nozzle as displayed in Figure 9.1b. Overall, static FluidNeRF shows impressive results; however, the vortices and other structures appear to be smoothed out compared to the perspective image.

As provided from Chapter 7, the hyperparameters of the network are critical to the reconstruction quality of FluidNeRF. The following subsection covers the investigation of hyperparameters on the reconstruction quality, where noise and camera model uncertainty could cause instabilities. Our investigation will cover positional encoding and activation functions.

9.1.1 Hyperparameters: Positional Encoding and Activation

In both reconstructions shown above (Figure 9.1 and Figure 9.2), the static FluidNeRF with an activation function of ReLU and $L_p = 0$ appear to have an overly smooth reconstruction, which is expected given the results from Chapter 7. Increasing L_p should improve the high-frequency content in the reconstructed volume. For the sake of consistency, the following results of this subsection will utilize the 1-inch diameter images for the instant in time as shown by Figure 9.1a. The hyperparameters of interest are position encoding (L_p) and the activation function of the network. Both of these parameters can help increase the complexity of the neural network volume approximation. The synthetic experiments proved that FluidNeRF is more robust to noise than an ART tomography method. However, the projections from the smoke-filled jet also have camera model uncertainty coupled with measurement noise. Thus,



(a) (b)

Figure 9.5: (a) NRMSE and (b) SSIM of a rendered perspective relative a captured perspective for the 1 inch diameter nozzle experiments as a function of iteration using different L_p. The activation function was ReLU. The perspective utilized for this comparison was not included in the reconstruction set, therefore, only 17 cameras were utilized for these results.

the optimal parameters found in the synthetic section might be suboptimal for different flow regimes and experimental data.

Figure 9.5 presents the accuracy in terms of NRMSE and SSIM of a projection as a function of iteration, where the projection was not included in the set used for reconstruction. The reconstruction process employed five distinct positional encoding values ($L_p = 0, 2, 4, 6, 8$), ReLU activation, and 17 camera perspectives. Note that $L_p = 8$ was found to be the optimal value for the synthetic results. In each case, the accuracy follows a similar trend during training, converging to similar values. NRMSE converged much quicker to the final value than SSIM. The increased convergence is caused by FluidNeRF employing MSE as the loss function; therefore, the method is trained to minimize MSE. Alternatively, SSIM is still increasing, indicating it was not fully converged when the reconstruction was terminated. Positional encoding does reduce the fluctuations that occur during training, as the higher fluctuations are present for $L_p = 0$. The fluctuations in loss caused the case with $L_p = 0$ to require more iterations before converging compared to the reconstructions with positional encoding. Overall, there is little difference between the positional encoding values and the projection accuracy.

Both accuracy metrics are evaluated on a LoS projection of the volume. While these metrics provide valuable quantitative insights, they do not fully capture the influence of positional encoding on the volumetric reconstruction. Hence, a qualitative analysis is necessary to provide a more comprehensive understanding. This analysis involves a cross-sectional view of the reconstruction volume for $L_p = 0$, $L_p = 4$, and $L_p = 8$, as depicted in Figure 9.6. Without positional encoding, FluidNeRF produced a qualitatively smoother solution with minimal artifacts. However, it loses the finer details of the jet, including in the swirl of the vortices. Increasing positional encoding does improve the ability of FluidNeRF to capture the more detailed features in the volume, but high-frequency artifacts become apparent. The uncertainty of a ray projection from each image model is the most likely cause of the artifacts. In the case with $L_p = 0$, the impact of uncertainty and noise is smoothed out. Increasing positional encoding causes artifacts to appear, where positional encoding is added to help capture the higher frequency information in the volume. Note that these artifacts could arise due to the perspectives capturing the particulate nature of the smoke in the flow (Figure 9.1a), where FluidNeRF is trying to resolve this feature. In experimental data, a balance arises between minimizing reconstruction artifacts and accurately capturing the higher spatial frequency details.

Figure 9.6 shows a qualitative comparison between ReLU and Swish at three different L_p (0, 4, and 8). With $L_p = 0$, The swish activation appears to better resolve the features in the flow, including the vortex structures, with the edges being more defined. The higher-density smoke streams in the jet's core are also better resolved with the swish network. Like ReLU, the swish network produces a smoother solution without positional encoding, but the finer details seen in Figure 9.1a are smoothed out. As L_p is increased, both methods have high-frequency artifacts that start to appear and worsen with higher positional encoding. The swish activation network notably enhances the qualitative performance compared to ReLU.

Figure 9.7 shows the quantitative analysis between ReLU and swish for the different levels of L_p. The results from NRMSE do not show a clear separation between the two activation functions. From NRMSE, ReLU and swish have an optimal L_p of 4 and 2, respectively. However, the variation in NRMSE for both cases suggests that both methods converge to similar reprojection accuracy but different volumes (Figure 9.6). SSIM has a more definitive conclusion, with swish outperforming ReLU for every L_p. SSIM indicates that the best reprojection of the volume is swish without positional encoding. This conclusion is most likely due to SSIM

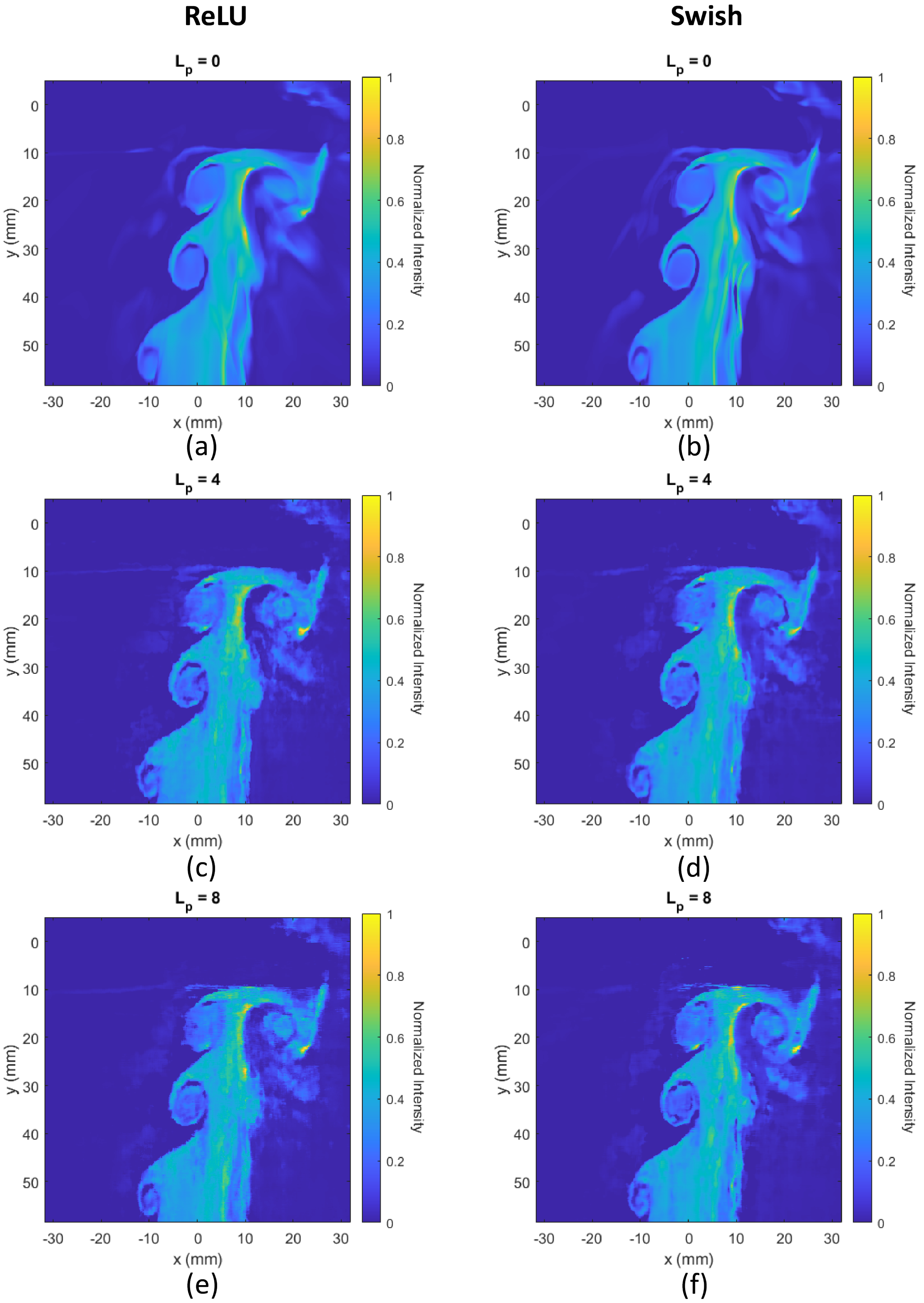
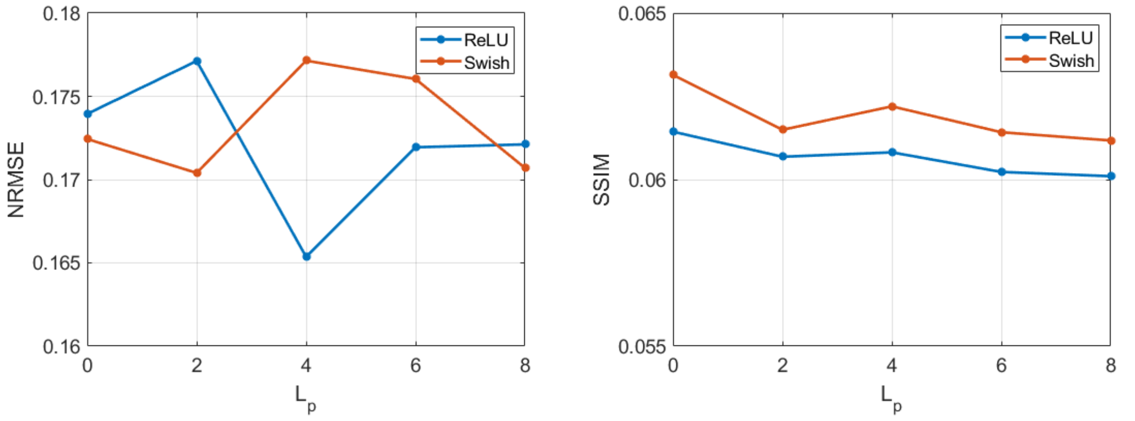


Figure 9.6: A x-y slice at z = 0 of the 1 inch diameter nozzle volume reconstructed using static FluidNeRF with an activation function of ReLU (left column) and Swish (right column) at three different L_p. Note the reconstructions were performed with the same 17 camera views.



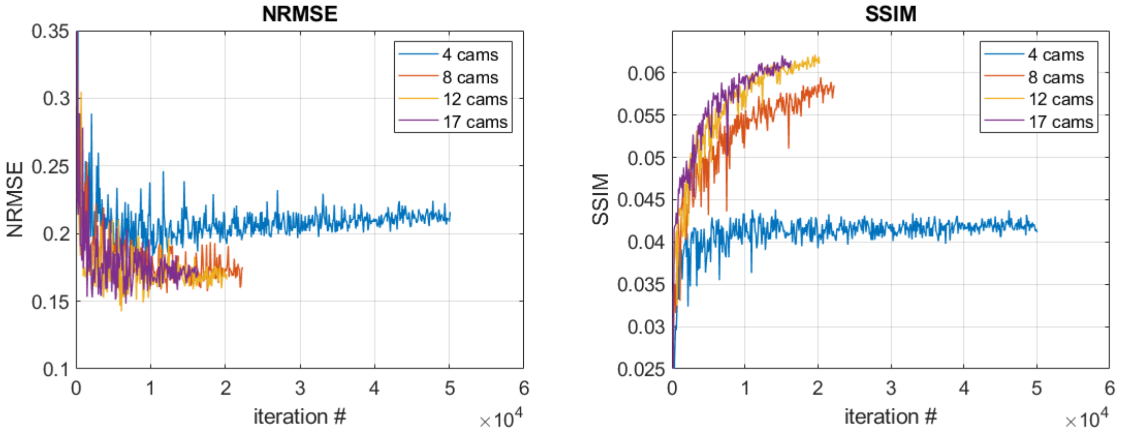
(a) (b)

Figure 9.7: (a) NRMSE and (b) SSIM of a rendered perspective relative a captured perspective for the 1 inch diameter nozzle experiments as a function of positional encoding. Each line corresponds to a different activation function, ReLU or Swish.

being a function of variation (Eqn.6.19), where positional encoding generates high-frequency artifacts not present in the captured images. Similar to other tomography methods, regularization can be added to limit the non-physical artifacts. Regularization can be included in FluidNeRF through the loss function. One potential option is SSIM, as it penalizes variation differences between the captured and rendered images. Swish with $L_p = 2$ was selected to process the following results because the swish function outperforms ReLU. The authors decided that a low value of positional encoding was desired to capture the smaller details without causing excessive high-frequency artifacts.

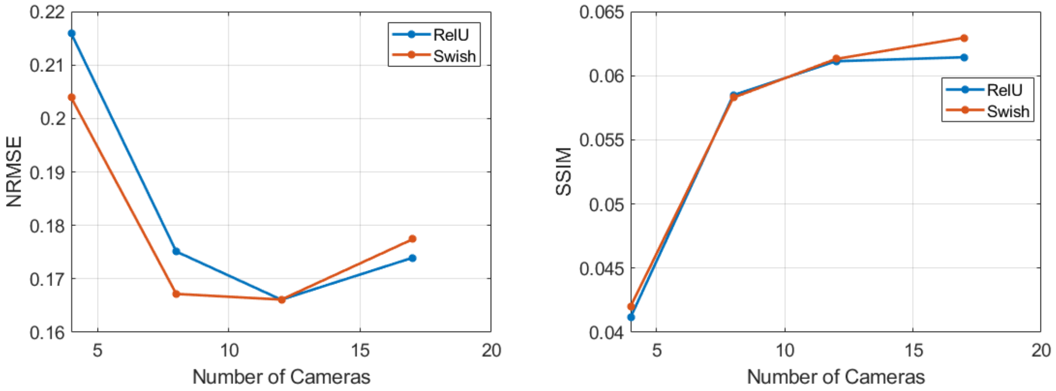
9.1.2 Number of cameras

The quantity of projections used for reconstruction is also essential for experimental setups, where large camera arrays can be costly or impractical due to limited optical access. Therefore, this section evaluates the reconstruction quality using various numbers of cameras. Figure 9.8 shows the convergence rate of a projection that was not included in the reconstruction sets. The projection subsets were selected to evenly distribute the camera without including the same projection for comparison. Although, Mohri *et. al.* [10] found that unequal spacing had a limited effect on the reconstruction quality. Since each iteration considers rays from all cameras, the cases with a higher number of cameras had increased convergence, as indicated



(a) (b)

Figure 9.8: (a) NRMSE and (b) SSIM of a rendered perspective relative a captured perspective for the 1 inch diameter nozzle experiments as a function of iteration for different subsets of cameras. The perspective utilized for this comparison was not included in the reconstruction projection set.



(a) (b)

Figure 9.9: (a) NRMSE and (b) SSIM of a rendered perspective relative a captured perspective for the 1 inch diameter nozzle experiments for different subsets of cameras using ReLU and Swish activation functions. Swish and ReLU used $L_p = 2$ and $L_p = 0$, respectively. The perspective utilized for this comparison was not included in the reconstruction projection sets.

by the length of each trace. With four cameras, the iterations approached a minimum with NRMSE before slowly diverging to higher values. SSIM quickly approached the final value and then leveled but still slightly increased. SSIM increasing indicates the solution was still approaching an optimal SSIM for the four-camera case. The 8, 12, and 17 camera solutions followed a similar trend to each other and approached a comparable NRMSE value. SSIM offers a slightly different trend, with the 8-camera case converging to a slightly lower value

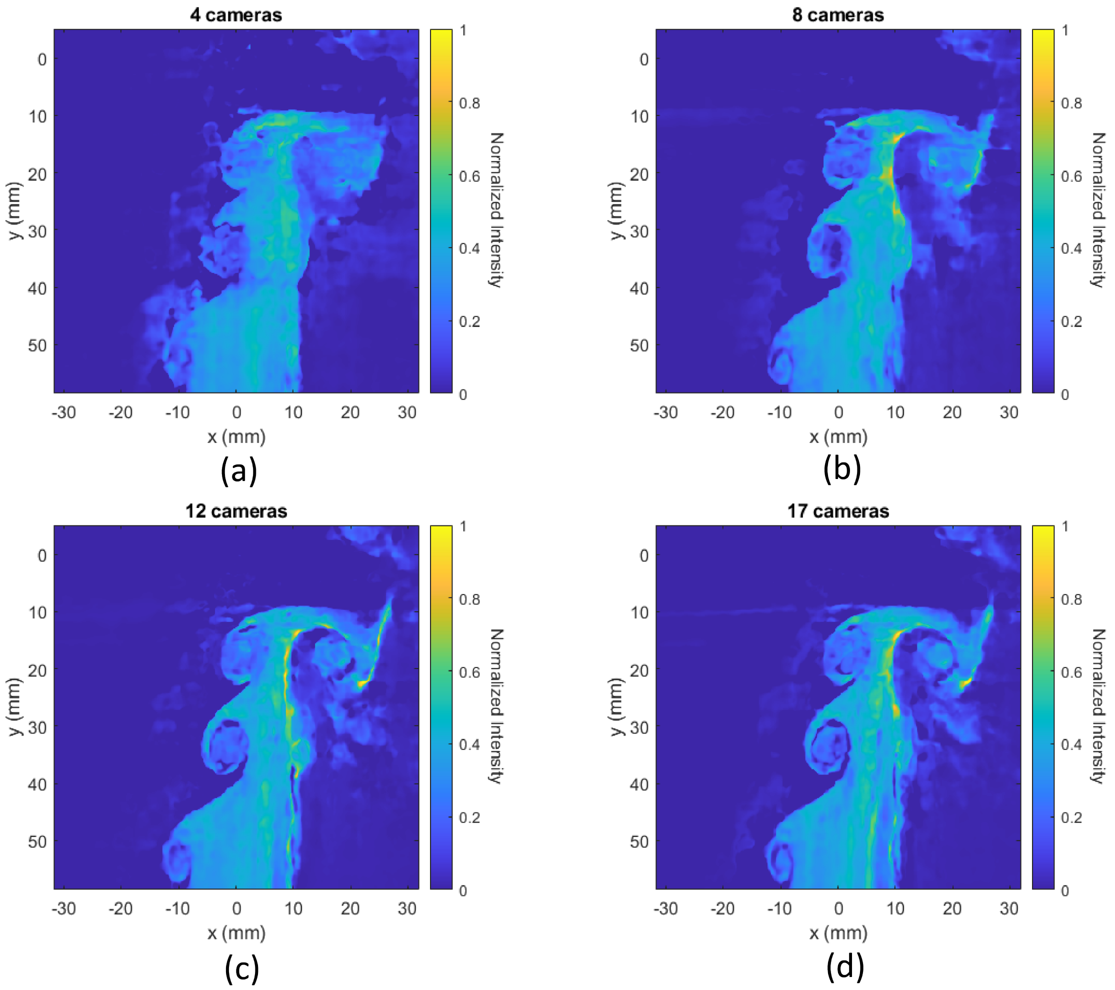


Figure 9.10: A x-y slice at z = 0 of the 1 inch diameter nozzle volume reconstructed using static FluidNeRF using for four subsets of cameras.

than 12 and 17. Overall, increasing the number of cameras improves the solution but is limited compared to the synthetic case.

The final solution accuracy metrics as a function of numbers of cameras is displayed in Figure 9.9b. The reconstructions were conducted with both ReLU and swish to examine if there is any difference between the two activation functions, where both functions exhibit improved NRMSE values up to 12 cameras, followed by an increase for the 17 cameras case, contrary to expectations. One possible explanation for the NRMSE increase could be caused by including a camera that has a higher imaging model uncertainty. Unlike NRMSE, SSIM has a monotonic increase when cameras are added to the reconstruction set with diminishing returns. These are similar trends as found in Section 7; however, the synthetic results showed the accuracy flattening after 20-30 cameras. The trend for the experimental image set shows diminishing returns at lower quantities of projections due to the addition of image model uncertainty. Future synthetic results should examine the imaging models as a source of uncertainty.

Figure 9.10 displays an x-y cross-section of the volume reconstructed by FluidNeRF using 4, 8, 12, and 17 cameras. While the four-camera case captures the jet's general shape, the flow's features could still be further resolved. Additionally, the higher intensity regions are not defined compared to the other camera counts. The 8-camera reconstruction has more defined vortex structures than the 4-camera case, and the higher-intensity regions are becoming more defined. Increasing the number of cameras further improves the reconstruction. The 17-camera case has the most defined vortex structures, and the vertical streaks in the core flow are more defined. Qualitatively, the improvements decrease with the added number of cameras, with the best solution being the 17-camera case. These qualitative results match the SSIM metric. Therefore, SSIM is a viable metric when comparing reprojection to capture images to examine tomography methods.

9.2 Time-Resolved FluidNeRF

The following section covers the results of TR-FluidNeRF for the experimental visualization of the jet flow. Figure 9.11 presents a LoS perspective for five consecutive time steps of the smoke-filled jet for the 1-inch diameter nozzle. The FOV of the jet flow captures vortex roll-up in the shear layer of the flow as the jet transitions to turbulence. The authors would like to point out the collision of two vortices in the shear layer in the central frame, which leads to the destruction of the two vortices in later frames. The slices from the volumes generated from static FluidNeRF and TR-FluidNeRF are provided next to the representative perspective in Figure 9.11. L_p is consistent between both cases, while TR-FluidNeRF employs $L_t = 1$. Similar to the results discussed in Section 9.1, the static FluidNeRF method captures the general shape and the vortex structures in the shear layer; however, some artifacts are present. TR-FluidNeRF's results are very similar to the static reconstruction, with the vortices' general structure and finer details preserved. Additionally, the artifacts in static FluidNeRF reconstructions are reduced for TR-FluidNeRF. Both methods captured the collision of the vortex pair and later the combining of the vortex pair. Overall, the qualitative analysis shows that TR-FluidNeRF has improved

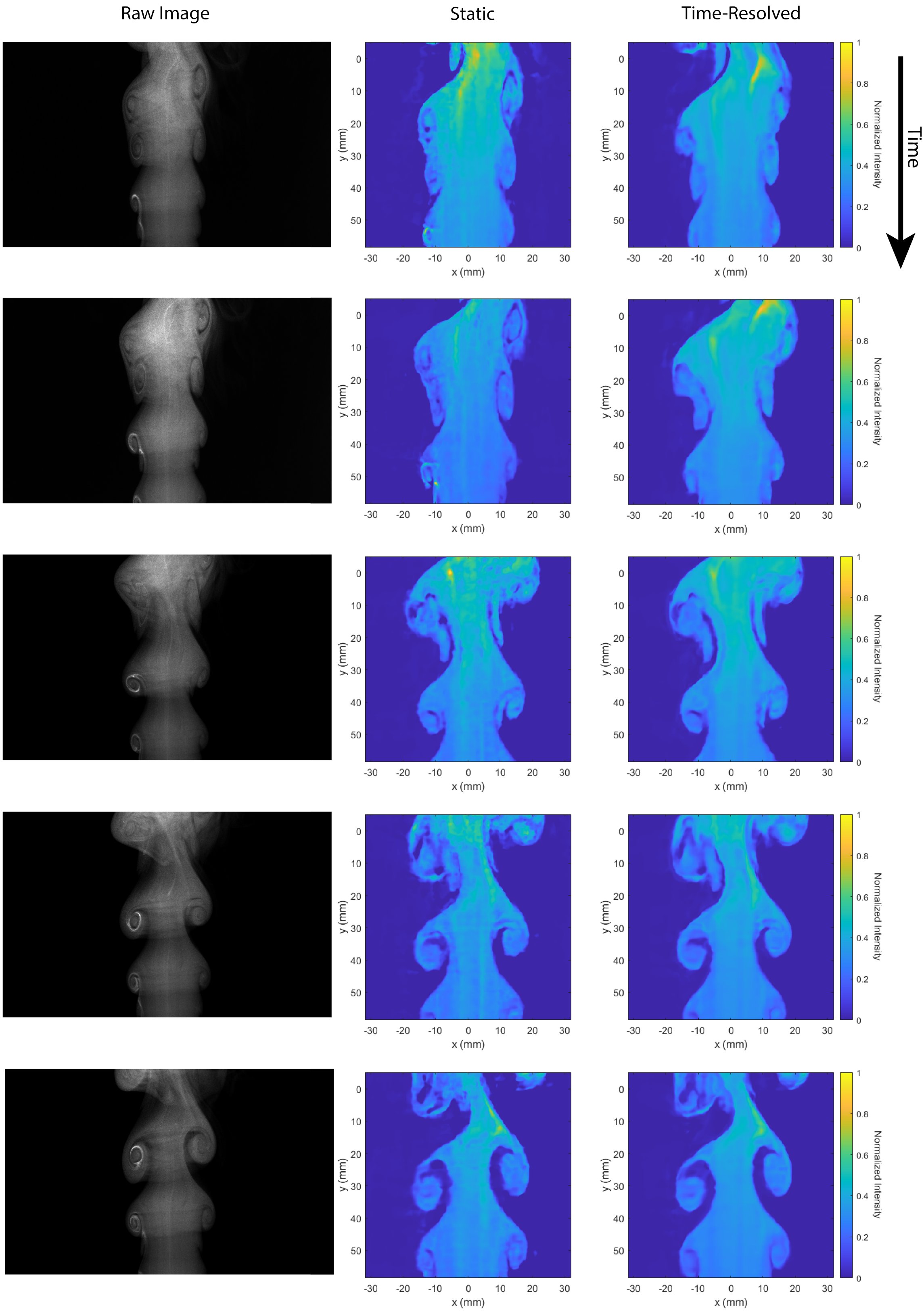


Figure 9.11: Five consecutive perspectives in time (left) that were used to reconstructed using static FluidNeRF (middle) and TR-FluidNeRF (right). Both reconstructions were conducted with 17 cameras.

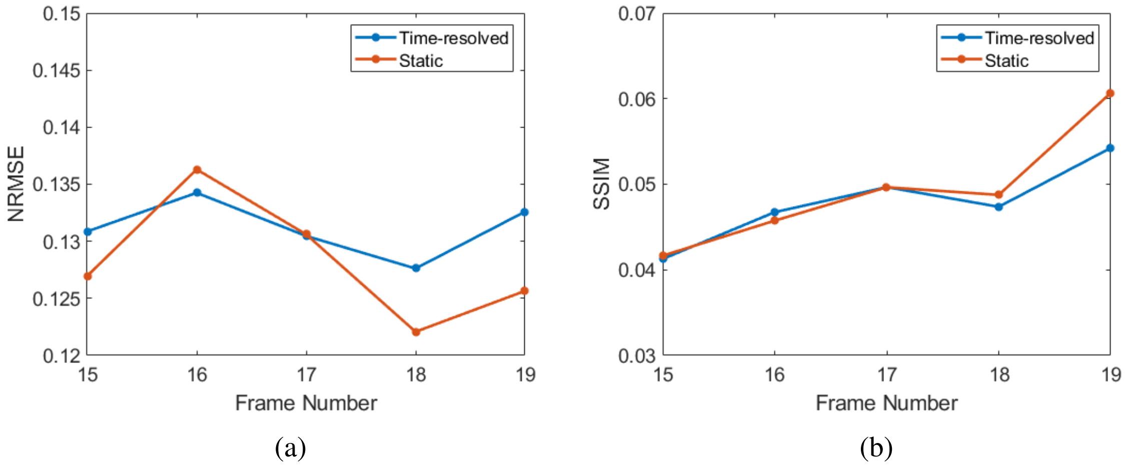


Figure 9.12: (a) NRMSE and (b) SSIM of a rendered perspective relative a captured perspective for both static and time-resolved FluidNeRF. Both methods used 17 cameras for each frame.

performance over static FluidNeRF regarding the non-physical artifacts caused by the imaging model uncertainty. The added time dimension of the model helps constrain the reconstruction.

A quantifiable comparison between static and time-resolved FluidNeRF is offered in Figure 9.15. The accuracy metrics are calculated for each rendered perspective from static and TR-FluidNeRF for the images shown in Figure 9.11. For both metrics, there are minor differences between the static and TR-FluidNeRF, with static showing slightly improved NRMSE and SSIM for the last two frames. Both methods converged to similar projection accuracy, but the reconstructed volumes had some differences (Figure 9.11). According to these results, the included time information does not reduce the method's performance.

TR-FluidNeRF helps compress the reconstructions further since the reconstructions for the five-time steps are integrated into the neural network-trained parameters. TR-FluidNeRF requires roughly double the time to converge for the five-time step data set as compared to static FluidNeRF for a single frame; however, TR-FluidNeRF converges much faster than static FluidNeRF given reconstruction time for each time step. Additionally, the network size is approximately the same, requiring the equivalent storage cost of five frames compared to a single frame for static FluidNeRF. Therefore, TR-FluidNeRF demands less computational time and storage costs than static FluidNeRF while producing similar projection accuracy.

Figure 9.13 presents the accuracy metrics of rendered perspectives from TR-FluidNeRF and static FluidNeRF as a function of the number of projections. TR-FluidNeRF outperformed

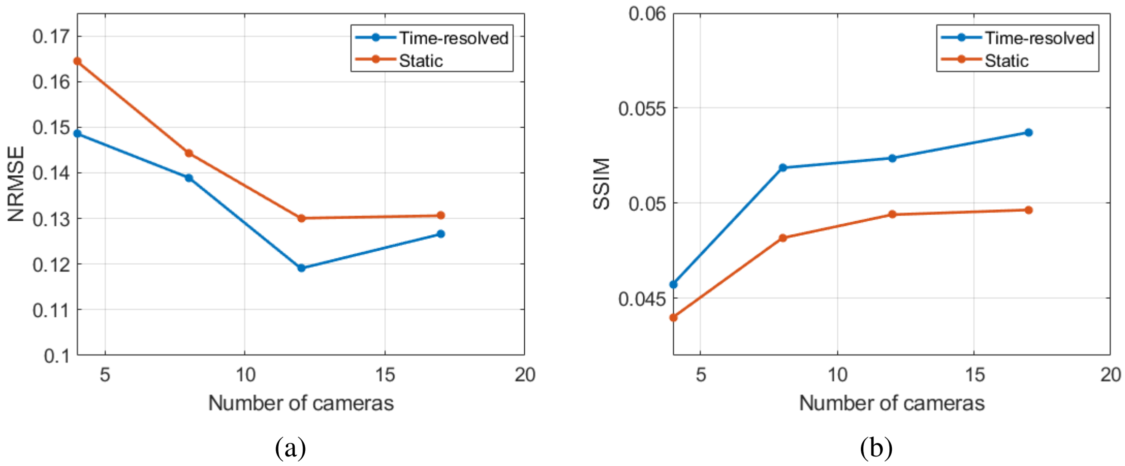
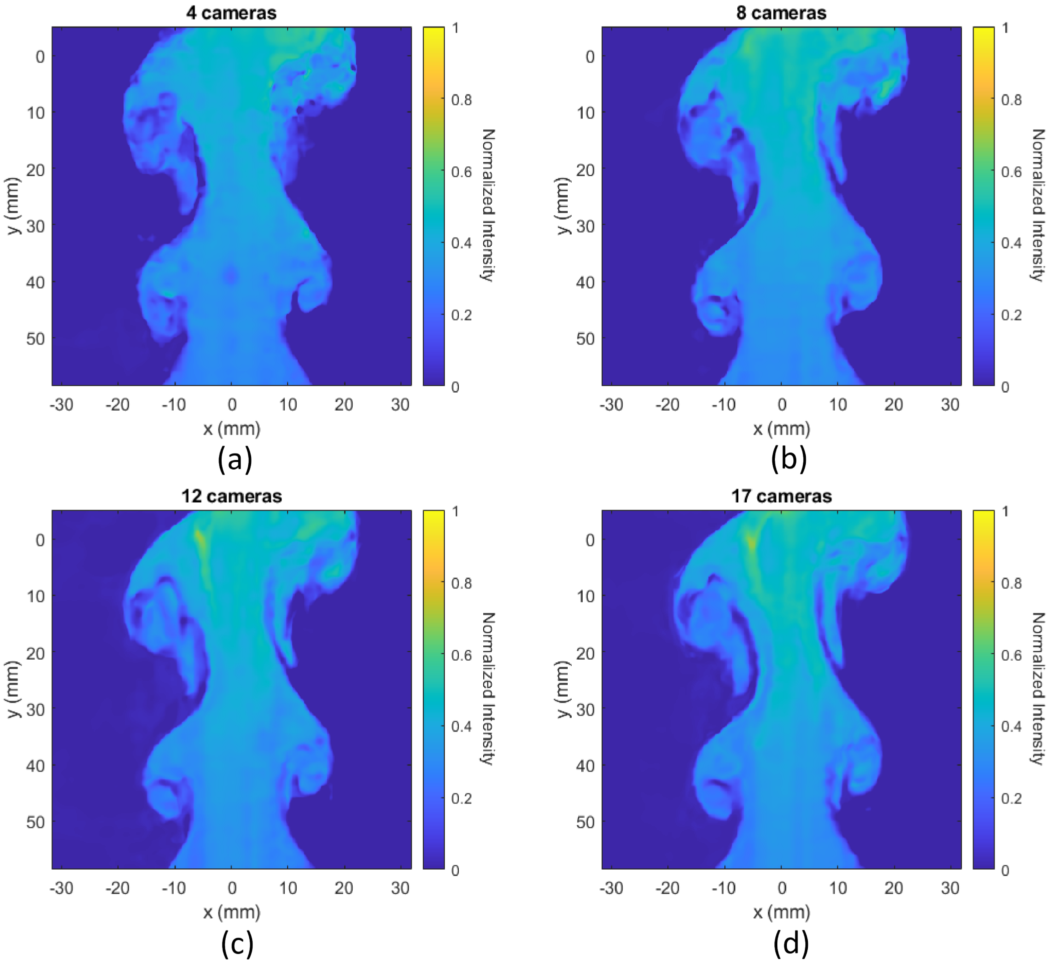


Figure 9.13: (a) NRMSE and (b) SSIM of a rendered perspective relative to a captured perspective of the central frame for the 1 inch diameter nozzle experiments for different number of cameras using TR-FluidNeRF and static FluidNeRF.

static FluidNeRF for each subset of perspectives. Both have the expected trend of a monotonic increase in accuracy with diminishing returns. The improved performance for all cases indicates that the addition of time in TR-FluidNeRF helps constrain the tomography problem, leading to fewer artifacts from the measurement uncertainty.

A qualitative comparison between the number of camera cases is illustrated in Figure 9.14. The qualitative analysis of the x-y slices follows the similar trend captured by Figure 9.13. The four camera case has the general shape of the jet, but the vortex structures are degraded and the higher intensity streaks in the core are lost. The higher intensity streaks in the core do not appear until the 12 cameras case. Adding more cameras enhances the apparent resolution and sharpens the details of the thin features close to the central core, which stem from the top vortex structure. Overall, the quality of the reconstructions improves with increasing numbers of cameras but with diminishing returns. This effect was also discovered in the synthetic results and static experimental results.

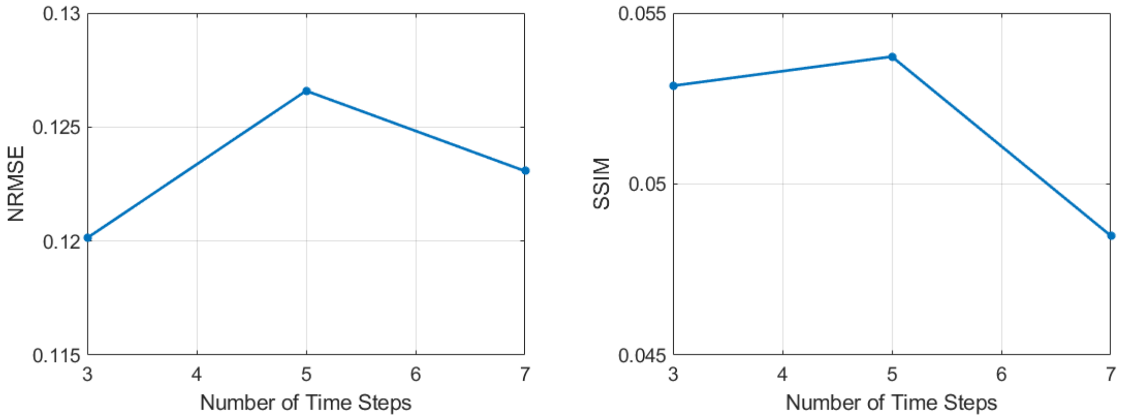
The TR-FluidNeRF can incorporate many time steps into the time-resolved reconstruction method. However, extending the duration over which the neural network must adapt raises the complexity of the continuous function approximation. Therefore, there will be a limit on the number of time steps that TR-FluidNeRF can consider before the reconstruction quality



(a) (b) (c) (d)

Figure 9.14: A central x-y slice of the TR-FluidNeRF reconstructed volume using (a) 4 cameras, (b) 8 cameras, (c) 12 cameras, and (d) 17 cameras.

diminishes for a given network structure. Figure 9.15 displays the projection metrics for rendered perspectives for TR-FluidNeRF that was trained using 3-, 5-, and 7-time steps of the time-resolved experimental images. Each set of frames was centered around the same frame (frame 17). For the three variations, there are limited differences, as indicated by NRMSE, with a general trend to lower accuracy values with an increasing number of steps. SSIM shows a similar trend with a more significant jump to a less accurate solution for the 7-time step case. Although, the variation is still minimal. Overall, for the limited investigation provided, the number of time steps has a negligible effect on the reprojection quality for the utilized network structure and the complexity of the flow. A more robust examination expanding to datasets with many more time steps will help understand this limitation. However, this will depend on the measurement uncertainty, flow properties, and NN framework.



(a) (b)

Figure 9.15: (a) NRMSE and (b) SSIM of a rendered perspective relative a captured perspective of the central time step as a function of number of time steps included in the TR-FluidNeRF reconstruction. Each reconstruction had the same central frame.

9.3 Chapter Summary

The previous chapter covers the results of the 3D reconstructions of the experimental jet visualization using FluidNeRF. Overall, FluidNeRF demonstrated impressive reconstructions of the scalar field measurements, where the method was able to distinguish the general shape, structure, and some minor details of the jet flow. This demonstration is a crucial verification of FluidNeRF for experimental measurements, as the previous results utilized synthetic datasets. The results comprised qualitative and quantitative analyses, with the quantitative analysis using a projection not included in the reconstruction image set. The first portion of this chapter investigated the effect of positional encoding, activation function, and number of cameras on the static FluidNeRF reconstruction. General conclusions from these results are provided in the following list:

1. Positional encoding had a minimal effect on the accuracy of volume reprojections compared to the captured projections. However, the qualitative analysis showed that the reconstructions using various L_p produced different volumes. There was a trade-off between non-physical reconstruction artifacts and resolving the finer details of the volume, where lower values of L_p produced smoother volumes but lacked the finer attributes of the jet.

2. The neural network's activation function proved important to the reconstruction quality. We investigated the swish and ReLU activation functions of the network. Qualitatively, swish produced more distinct edges of the jet structure and slightly reduced the artifacts caused by increasing L_p. Swish yielded higher values of SSIM than ReLU. Therefore, swish was chosen as the default function for the rest of the experimental results.

3. The static FluidNeRF quality was improved by increasing the number of cameras employed for reconstruction. However, increasing the number of cameras produced diminishing returns, where the quantitative analysis demonstrated the accuracy plateauing around 17 cameras for our dataset. This was slightly lower than what was determined for the synthetic results.

In addition to evaluating static FluidNeRF, a study of TR-FluidNeRF was conducted. The synthetic results demonstrated that the added time component increases the complexity of the approximated volume but also provides a continuous approximation in spatial and temporal coordinates that allows the method to relate perspectives in time. TR-Fluid was qualitatively and quantitatively compared to static FluidNeRF for a given network structure. Additionally, TR-FluidNeRF was utilized to reconstruct the experimental data with 3, 5, and 7-time steps. The following list includes the general conclusions from these results:

1. Adding time to the reconstruction through TR-FluidNeRF helped dampen the effect of artifacts that were found in the static FluidNeRF volume. TR-FluidNeRF yielded similar quantitative results compared to static while improving the appearance of the reconstruction while minimizing the artifacts. The time component helped implicitly constrain the tomography problem.

2. The number of time steps included in the reconstruction had minimal effect on the rendered perspective when moving from three-time steps to seven. However, TR-FluidNeRF is expected to get diminishing results as time-steps are increased for a given network architecture.

Chapter 10

Conclusions & Future Work

10.1 Conclusions

This work introduces a novel machine learning tomography technique to the flow diagnostic community, building upon the NeRF concept. The underlying idea of neural implicit representations is the representation of a volume with a continuous function of spatial and temporal coordinates using a neural network. The tomography method developed in this work, FluidNeRF, operates similarly to traditional tomography methods. The volume approximation is updated using the loss between captured and predicted projections, where the predicted projection is rendered using an emission-based image model. The novelty of FluidNeRF lies in its use of a neural implicit representation, a concept that is new in the 3D flow diagnostic community. This work evaluates FluidNeRF against a traditional ART-based tomography method, analyzing the effect of hyperparameters and optical layout on its reconstruction quality using both synthetic and real experimental volumes. The major conclusions from this work are:

1. A neural implicit representation is a viable technique to solve the inverse tomography problem for flow diagnostics.
2. FluidNeRF exhibits comparable or superior accuracy compared to traditional tomography methods.
3. FluidNeRF had increased resiliency to image noise as compared to traditional techniques.

4. Our investigation illustrates the scalability advantage of FluidNeRF over traditional tomography methods due to the gridless volume approximation, where the neural volume approximation improves data compression.

5. Our investigation illustrates that TR-FluidNeRF can reconstruct multiple time-resolved frames with a single network, reducing the computational expense and producing similar accuracy as compared to static FluidNeRF.

6. TR-FluidNeRF minimizes measurement uncertainty and noise for experimental data by constraining the problem with time.

The following will discuss the conclusions from this work and provide insight into the future avenues needed for improving the method.

Before the initiation of this work, the neural implicit representation of a volume was a relatively new concept employed by the PINN and computer vision communities. However, more investigations are needed to understand the limitations of neural implicit representations for flow field tomography problems. This work provides significant insight into the potential of machine learning to solve tomography problems for emission-based flow diagnostics using a neural implicit representation and represents an important demonstration of this capability. The tomography problem is ill-posed, so the best way to demonstrate its accuracy was by comparing the reconstruction to a ground truth with a synthetic dataset. It is important to note that FluidNeRF can be employed with different camera configurations and quantities, making it a viable technique for flow tomography in various applications with different optical access. This work concludes that machine learning techniques, similar to FluidNeRF, hold great potential for tomographic flow measurements.

Another crucial aspect of this research was comparing the machine learning method to a traditional tomographic technique adapted to flow diagnostics. As demonstrated in the literature review, ART-based methods are the most commonly used techniques for flow diagnostics. FluidNeRF produced reconstruction quality similar to or superior to ASART, where both methods used the same projections. This comparison is essential as it reassures the audience about the validity and reliability of the method. Even with other advantages of neural implicit representation tomography methods, a new tomography method must produce comparable accuracy to current methods to be considered a viable solution for flow tomography. A qualitative and quantitative comparison was conducted between FluidNeRF and ASART, where FluidNeRF captured the edges of the laminar flow region better than ASART and produced similar results in regions with high spatial frequencies. FluidNeRF had significant improvements in accuracy as the number of cameras increased compared to ASART. The accuracy improvement could be related to the continuous function approximation, which does not inherently limit the topology of the reconstruction in contrast to the discretized nature of traditional techniques. Therefore, the volume approximation can improve with an increased number of projections without necessitating a more dense grid.

In addition to the increase in accuracy, this work evaluated the effect of image noise on the reconstruction accuracy for both FluidNeRF and ASART. FluidNeRF was more robust to image noise than the ART-based technique. The accuracy of FluidNeRF using noisy images was greater than ASART using noiseless images. This is an important conclusion because flow measurements will have sources of noise; therefore, having higher resiliency to noise makes the method more applicable to real flow measurements.

The neural implicit representation of FluidNeRF provides a framework to extend to larger volumes of interest. Image sensors are rapidly improving to increase the resolution of projections, allowing the ability to increase the volume resolution and size. FluidNeRF will scale better than traditional voxel-based methods because those methods have an inherent expansion in memory and storage requirements as the number of voxels increases. Neural implicit representations do not have the same scaling since the storage requirements are related to the weights and basis of the neural network, which require approximately three orders of magnitude less memory costs. Therefore, FluidNeRF provides a more computationally memory-efficient method to scale to more substantial problems. Essentially, FluidNeRF reconstructs to the highest possible resolution that can be supported for the given camera layout, imaging model, perspectives resolution, and network parameters. The current feed-forward neural network design using back-propagation has some practical limitations, including vanishing or exploding gradients, inactive nodes, hardware limitations, and training time. Deeper networks require more sophisticated optimization algorithms to achieve good performance. Additionally, batch sizes must be sufficient to capture the general trends in the volume, where larger problems will require larger batch sizes. The current affordable GPU hardware has memory limitations that limit the batch size. However, these neural network approaches stand to benefit from the rapid advances in GPU technology. Further investigation is needed to understand the relationship between the scale of the flow diagnostic problem and the network framework and batch size.

The machine learning-based method was extended to show that the technique can produce a spatial-temporal approximation of the volume using time-resolved projections. Traditional tomography methods do not utilize temporal information for reconstruction. While adding time increases our method's complexity, time information can also constrain the problem, similar to increasing the number of cameras. For the perfect imaging model of the synthetic experiments, time-resolved FluidNeRF produced higher reconstruction quality of static FluidNeRF, where the limitation was the neural network approximation. TR-FluidNeRF had further improvements by using the swish activation function, where the network sizes between static and time-resolved where approximately the same. Adding time to the model further reduces the memory costs as multiple reconstructions in time are stored by one network. TR-FluidNeRF is a valid technique for creating a spatial-temporal approximation for flow tomography measurements.

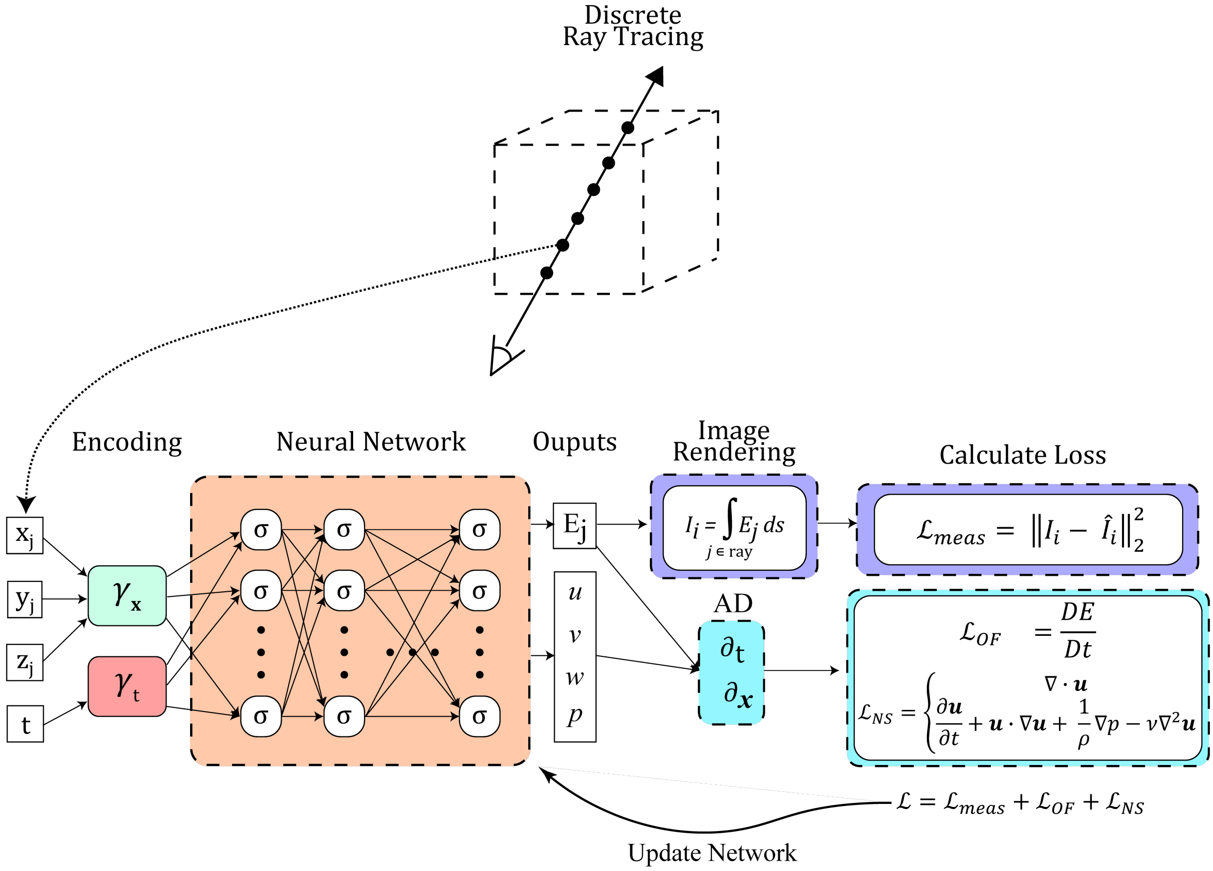
Adding time to the neural implicit representation tomography algorithm creates an additional constraint to improve the reconstruction quality, especially for real experiments. While TR-FluidNeRF does not explicitly impose a new constraint, the method approximates a continuous spatial-temporal function of the volume that requires the measurements to be consistent in time. Therefore, non-physical artifacts or noise that are developed in a reconstruction at a given time will be minimized across all time steps since a continuous function must approximate the volume. The reduction in artifacts due to the addition in time was demonstrated in the comparison between static and TR-FluidNeRF for the smoke-filled flow experiments. Therefore, TR-FluidNeRF is recommended for time-resolved experiments.

10.2 Future Work

This work has demonstrated FluidNeRF for tomography and provided an initial investigation of the neural implicit flow representation method for flow diagnostics. Additional studies of neural implicit representation methods are still required to understand the limitations of this model. Additionally, there are areas for improving the current FluidNeRF technique. This section will cover the suggestions for future work.

The neural implicit representation does not have the inherent limitation on volume resolution like discretized methods. However, the volume approximation still possesses limitations. The volume approximation is also a function of the spatial frequencies present in the volume, which will depend on the flow of interest. An analysis of a neural network's expressivity of a volume does not have to include the inverse tomography problem but rather direct training using a known volume. This investigation should consider encoding, network size, and network structure in terms of the accuracy of the representation. Pan *et. al* [54] conducted a similar analysis for PINNs of flow, showing that the network structure and shape are essential to the volume approximation. While these explore the expressivity of a neural network for direct training, the tomography problem will only be able to approach the accuracy of what is capable with direct training.

Other network-related parameters could improve the method. One of these parameters is the network architecture, where the current method uses a simple feed-forward network. In the PINN field, the neural implicit flow architecture demonstrated improved results [54] using a shape and parameter network structure to approximate the flow field. The recurrent neural network is another network style that has been used for temporal data. Other than network structure, activation and loss functions are other parameters that can improve the ability of the network to resolve the volume. For the work conducted in this dissertation, only two common activation functions were investigated; however, there are other functions that have been demonstrated to improve results in recent literature. The loss function is the metric that is used to train the network; therefore, other loss functions should be analyzed to determine the most



Discrete Ray Tracing
Encoding
Neural Network
Ouputs
Image Rendering
Calculate Loss
x_j
y_j
z_j
t
γ_x
γ_t
σ
E_j
u
v
w
p
$I_i = \int_{j \in ray} E_j \, ds$
$\mathcal{L}_{meas} = \|I_i - \hat{I}_i\|_2^2$
AD
∂_t
$\partial_{\boldsymbol{x}}$
$\mathcal{L}_{OF} = \frac{DE}{Dt}$
$\mathcal{L}_{NS} = \begin{cases} \nabla \cdot \boldsymbol{u} \\ \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho} \nabla p - \nu \nabla^2 \boldsymbol{u} \end{cases}$
$\mathcal{L} = \mathcal{L}_{meas} + \mathcal{L}_{OF} + \mathcal{L}_{NS}$
Update Network

Figure 10.1: Schematic of a conceptual physics-informed FluidNeRF algorithm for scalar-field flow diagnostics.

appropriate functions to improve the machine learning methods' ability to learn from the projections. Chu *et. al.* [7] demonstrated that additional loss functions can improve the training rate and accuracy.

The results showed the nature of the ill-posed, inverse problem, where infinite solutions exist for a given set of projections. Therefore, "prior" information must be enforced during the reconstruction to constrain the problem further to produce more realistic reconstructions. The imaging model is one portion of the algorithm that can be incorporated to make it more realistic. The current method utilized a pinhole ray model with infinitesimally thin rays, which was chosen due to its simplicity. However, infinitesimally thin rays are unrealistic as pixels and imaging optics have a finite size that are best approximated using point spread functions. One approach of interest would be the work by [132, 133] that improved the imaging model by accounting for a ray's finite size, where the ray is represented as a conical frustum in object space.

Additionally, the method can incorporate physics-based constraints through governing equations as demonstrated in Figure 10.1. This physics-informed framework reconstructs the scalar field emissions, similar to this dissertation, but it also implicitly predicts flow parameters, including velocity and pressure. Therefore, scalar field LoS measurements can directly i) calculate the velocity field in contrast to particles employed in PIV and ii) estimate the pressure gradients in the flow. The *a prior* information that links the flow parameters to the passive scalar reconstruction is the optical flow or advection-diffusion equation, where the Navier-Stokes equations constrain the flow parameters as demonstrated in previous PINN work [9, 52]. The inclusions of physics-based constraints do not limit the predictions to velocity and pressure but can also provide other estimates, including density, spectral intensities, temperature, and absorption in the volume. These flow parameters would require governing equations that relate those parameters to LoS measurements or additional measurements to be included in the method. The physics-informed model provides a unified framework to link multiple measurements together using governing equations. Overall, the current technique is adaptable and allows these new constraints and models to be incorporated, where the physics-informed method is poised to profit from the rapid advances of the PINN community.

References

[1] S. J. Grauer, K. Mohri, T. Yu, H. Liu, and W. Cai, “Volumetric emission tomography for combustion processes,” *Progress in Energy and Combustion Science*, vol. 94, p. 101024, 2023.

[2] J. Huang, H. Liu, J. Dai, and W. Cai, “Reconstruction for limited-data nonlinear tomographic absorption spectroscopy via deep learning,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 218, p. 187–193, 2018.

[3] J. Huang, H. Liu, Q. Wang, and W. Cai, “Limited-projection volumetric tomography for time-resolved turbulent combustion diagnostics via deep learning,” *Aerospace Science and Technology*, vol. 106, p. 106123, 2020.

[4] T. Ren, H. Li, M. F. Modest, and C. Zhao, “Machine learning applied to the retrieval of three-dimensional scalar fields of laminar flames from hyperspectral measurements,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 279, p. 108047, 2022.

[5] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhofer, C. Lassner, and C. Theobalt, “Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video,” *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[6] 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,” *Computer Vision – ECCV 2020*, p. 405–421, 2020.

[7] M. Chu, L. Liu, Q. Zheng, E. Franz, H.-P. Seidel, C. Theobalt, and R. Zayer, "Physics informed neural fields for smoke reconstruction with sparse data," *ACM Transactions on Graphics*, vol. 41, no. 4, p. 1–14, 2022.

[8] F. Zhang, W. Zhang, Q. Lei, X. Li, Y. Li, and M. Xu, "Voxel-free neural volume reconstruction technique for volumetric flame reconstructions," *Aerospace Science and Technology*, vol. 133, p. 108107, 2023.

[9] J. P. Molnar and S. J. Grauer, "Flow field tomography with uncertainty quantification using a bayesian physics-informed neural network," *Measurement Science and Technology*, vol. 33, no. 6, p. 065305, 2022.

[10] K. Mohri, S. Görs, J. Schöler, A. Rittler, T. Dreier, C. Schulz, and A. Kempf, "Instantaneous 3d imaging of highly turbulent flames using computed tomography of chemiluminescence," *Appl. Opt.*, vol. 56, pp. 7385–7395, Sep 2017.

[11] H. Liu, B. Sun, and W. Cai, "Khz-rate volumetric flame imaging using a single camera," *Optics Communications*, vol. 437, p. 33–43, 2019.

[12] J. Floyd, P. Geipel, and A. Kempf, "Computed tomography of chemiluminescence (ctc): Instantaneous 3d measurements and phantom studies of a turbulent opposed jet flame," *Combustion and Flame*, vol. 158, no. 2, p. 376–391, 2011.

[13] J. George, C. Clifford, T. P. Jenkins, and B. S. Thurow, "Volumetric spectral imaging and two-color pyrometry of flames using plenoptic cameras," *Applied Optical Metrology III*, 2019.

[14] T. Yu, F. J. Bauer, F. J. Huber, S. Will, and W. Cai, "4d temperature measurements using tomographic two-color pyrometry," *Opt. Express*, vol. 29, pp. 5304–5315, Feb 2021.

[15] M. M. Hossain, G. Lu, D. Sun, and Y. Yan, "Three-dimensional reconstruction of flame temperature and emissivity distribution using optical tomographic and two-colour pyrometric techniques," *Measurement Science and Technology*, vol. 24, no. 7, p. 074010, 2013.

[16] T. R. Meyer, B. R. Halls, N. Jiang, M. N. Slipchenko, S. Roy, and J. R. Gord, "High-speed, three-dimensional tomographic laser-induced incandescence imaging of soot volume fraction in turbulent flames," *Optics Express*, vol. 24, no. 26, p. 29547, 2016.

[17] F. J. Bauer, T. Yu, W. Cai, F. J. Huber, and S. Will, "Three-dimensional particle size determination in a laminar diffusion flame by tomographic laser-induced incandescence," *Applied Physics B*, vol. 127, no. 1, 2020.

[18] E. M. Hall, B. R. Halls, D. R. Richardson, D. R. Guildenbecher, E. Cenker, and M. Paciaroni, "Tomographic time resolved laser induced incandescence," *AIAA Scitech 2020 Forum*, 2020.

[19] L. Ma, Q. Lei, J. Ikeda, W. Xu, Y. Wu, and C. D. Carter, "Single-shot 3d flame diagnostic based on volumetric laser induced fluorescence (vlif)," *Proceedings of the Combustion Institute*, vol. 36, no. 3, p. 4575–4583, 2017.

[20] B. R. Halls, P. S. Hsu, S. Roy, T. R. Meyer, and J. R. Gord, "Two-color volumetric laser-induced fluorescence for 3d oh and temperature fields in turbulent reacting flows," *Optics Letters*, vol. 43, no. 12, p. 2961, 2018.

[21] B. R. Halls, N. Jiang, T. R. Meyer, S. Roy, M. N. Slipchenko, and J. R. Gord, "4D spatiotemporal evolution of combustion intermediates in turbulent flames using burst-mode volumetric laser-induced fluorescence," *Experiments in Fluids*, vol. 42, July 2017.

[22] H. Qi, J.-W. Shi, Y.-X. Su, B.-H. Gao, and Y.-T. Ren, "Soot temperature measurement within 3d flame by light-field imaging based on wave optics theory," *Optics and Lasers in Engineering*, vol. 138, p. 106419, 2021.

[23] W. Cai and C. F. Kaminski, "Tomographic absorption spectroscopy for the study of gas dynamics and reactive flows," *Progress in Energy and Combustion Science*, vol. 59, p. 1–31, 2017.

[24] F. Scarano, "Tomographic piv: Principles and practice," *Measurement Science and Technology*, vol. 24, no. 1, p. 012001, 2012.

[25] G. Elsinga, F. Scarano, B. Wieneke, and B. van Oudheusden, “Tomographic particle image velocimetry,” *Exp Fluids*, vol. 41, pp. 933–947, 2006.

[26] W. Cai, X. Li, F. Li, and L. Ma, “Numerical and experimental validation of a three-dimensional combustion diagnostic based on tomographic chemiluminescence,” *Optics Express*, vol. 21, March 2013.

[27] F. Zhang, H. Pan, X. Zhang, X. Bai, Z. Liu, X. Li, Y. Li, and M. Xu, “Three-dimensional reconstruction for flame chemiluminescence field using a calibration enhanced non-negative algebraic reconstruction technique,” *Optics Communications*, vol. 520, p. 128530, 2022.

[28] S. J. Grauer and A. M. Steinberg, “Fast and robust volumetric refractive index measurement by unified background-oriented schlieren tomography,” *Experiments in Fluids*, vol. 61, no. 3, 2020.

[29] T. Li, Y. Yuan, and Y. Suai, “Joint method for reconstructing three-dimensional temperature of flame using lucy-richardson and nearest neighbor filtering using light-field imaging,” *Science China Technological Sciences*, vol. 62, pp. 1232–1243, 2019.

[30] R. Gordon, “A tutorial on art (algebraic reconstruction techniques),” *IEEE Transactions on Nuclear Science*, vol. 21, no. 3, pp. 78–93, 1974.

[31] F.-F. Yin, “Adaptive algebraic reconstruction technique,” *Medical Physics*, vol. 31, no. 12, pp. 3222–30, 2004.

[32] “simultaneous algebraic reconstruction technique (sart): A superior implementation of the art algorithm,”

[33] A. Andersen, “Algebraic reconstruction in ct from limited views,” *IEEE Transactions on Medical Imaging*, vol. 8, no. 1, pp. 50–55, 1989.

[34] J. Qi and R. M. Leachy, “Iterative reconstruction techniques in emission computed tomography,” *Physics in Medicine & Biology*, vol. 51, pp. R541–R578, 2006.

[35] "A ROBUST MART ALGORITHM FOR TOMOGRAPHIC APPLICATIONS," *Numerical Heat Transfer: Part B: Fundamentals*, vol. 35:4, pp. 485–506, 1999.

[36] K. D. Kohm, K. Okamoto, D. Tsuru, and H. S. Ko, "A hybrid tomographic reconstruction algorithm for high speed X-ray tomography," *Experiments in Fluids*, vol. 22, pp. 137–143, May 1996.

[37] X. Yang, J. R. van Ommen, J. Schoormans, and R. F. Mudde, "A hybrid tomographic reconstruction algorithm for high speed X-ray tomography," *Computer Physics Communications*, vol. 196, May 2015.

[38] A. Unterberger, A. Kempf, and K. Mohri, "3D Evolutionary Reconstruction of Scalar Fields in the Gas-Phase," *Energies*, vol. 12, May 2019.

[39] M. J. Willemink and P. B. Noël, "The evolution of image reconstruction for ct-from filtered back projection to artificial intelligence," *Computed Tomography*, vol. 29, no. 5, pp. 2185–2195, 2019.

[40] C. Atkinson and J. Soria, "An efficient simultaneous reconstruction technique for tomographic particle image velocimetry," *Exp Fluids*, vol. 47, p. 553, 2009.

[41] J. K. Davis, C. J. Clifford, D. L. Kelly, and B. S. Thurow, "Tomographic background oriented schlieren using plenoptic cameras," *Measurement Science and Technology*, vol. 33, no. 2, p. 025203, 2021.

[42] N. Chetih and Z. Messali, "Tomographic image reconstruction using filtered back projection (fbp) and algebraic reconstruction technique (art)," in *2015 3rd International Conference on Control, Engineering & Information Technology (CEIT)*, pp. 1–6, 2015.

[43] S. Shao, K. Mallery, S. S. Kumar, and J. Hong, "Machine learning holography for 3d particle field imaging," *Optics Express*, vol. 28, no. 3, p. 2987, 2020.

[44] S. Shao, K. Mallery, and J. Hong, "Machine learning holography for measuring 3d particle distribution," *Chemical Engineering Science*, vol. 225, p. 115830, 2020.

[45] Y. Jin, W. Zhang, Y. Song, X. Qu, Z. Li, Y. Ji, and A. He, “Three-dimensional rapid flame chemiluminescence tomography via deep learning,” *Optics Express*, vol. 27, no. 19, p. 27308, 2019.

[46] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, “Nerf: Neural radiance field in 3d vision, a comprehensive review,” 2023.

[47] D. Kelly and B. Thurow, “Investigation of a neural implicit representation tomography method for flow diagnostics,” *Measurement Science and Technology*, vol. 35, p. 056007, Feb 2024.

[48] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, “D-nerf: Neural radiance fields for dynamic scenes,” *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[49] Z. Li, S. Niklaus, N. Snavely, and O. Wang, “Neural scene flow fields for space-time view synthesis of dynamic scenes,” *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[50] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, p. 686–707, 2019.

[51] M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, “Deep learning of vortex-induced vibrations,” *Journal of Fluid Mechanics*, vol. 861, p. 119–137, 2018.

[52] M. Raissi, A. Yazdani, and G. E. Karniadakis, “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations,” *Science*, vol. 367, no. 6481, p. 1026–1030, 2020.

[53] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, “Physics-informed neural networks (pinns) for fluid mechanics: A review,” *Acta Mechanica Sinica*, vol. 37, no. 12, p. 1727–1738, 2021.

[54] S. Pan, S. L. Brunton, and J. N. Kutz, "Neural implicit flow: a mesh-agnostic dimensionality reduction paradigm of spatio-temporal data," 2023.

[55] P. Clark Di Leoni, K. Agarwal, T. A. Zaki, C. Meneveau, and J. Katz, "Reconstructing turbulent velocity and pressure fields from under-resolved noisy particle tracks using physics-informed neural networks," *Experiments in Fluids*, vol. 64, no. 5, 2023.

[56] J. P. Molnar, S. J. Grauer, O. Léon, D. Donjat, and F. Nicolas, "Physics-informed background-oriented schlieren of turbulent underexpanded jets," *AIAA SCITECH 2023 Forum*, 2023.

[57] B. Yip, J. K. Lam, M. Winter, and M. B. Long, "Time-resolved three-dimensional concentration measurements in a gas jet," *Science*, vol. 235, no. 4793, p. 1209–1211, 1987.

[58] B. S. Thurow and K. P. Lynch, "Development of a high-speed three-dimensional flow visualization technique," *AIAA Journal*, vol. 47, no. 12, p. 2857–2865, 2009.

[59] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," *Computational Models of Visual Processing*, Oct 1991.

[60] D. D. Verhoeven, "Multiplicative algebraic computed tomographic algorithms for the reconstruction of multidirectional interferometric data," *Optical Engineering*, vol. 32, no. 2, p. 410, 1993.

[61] M. F. Modest, *Radiative heat transfer*. Academic Press, 2013.

[62] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. Society for Industrial and Applied Mathematics, 2001.

[63] S. Helgason, *The Radon transform*. Birkhauser, 1999.

[64] L. A. Shepp and B. F. Logan, "The fourier reconstruction of a head section," *IEEE Transactions on Nuclear Science*, vol. 21, no. 3, p. 21–43, 1974.

[65] C. J. Dasch, "One-dimensional tomography: A comparison of abel, onion-peeling, and filtered backprojection methods," *Applied Optics*, vol. 31, no. 8, p. 1146, 1992.

[66] R. Gordon, R. Bender, and G. T. Herman, “Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography,” *Journal of Theoretical Biology*, vol. 29, no. 3, p. 471–481, 1970.

[67] S. Karczmarz, “Angenaherte auflosung von systemen linearer glei-chungen,” *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.*, pp. 355–357, 1937.

[68] P. Eggermont, G. Herman, and A. Lent, “Iterative algorithms for large partitioned linear systems, with applications to image reconstruction,” *Linear Algebra and its Applications*, vol. 40, p. 37–67, 1981.

[69] Z. Chang, S. Huang, T. Wang, Y. Shang, Q. Zhang, C. Yan, and F. Zou, “Adaptive simultaneous algebraic reconstruction technique for retrieving refractive index profiles of optical fiber,” *Optical Engineering*, vol. 55, no. 9, 2016.

[70] X. Wan, F. Zhang, Q. Chu, K. Zhang, F. Sun, B. Yuan, and Z. Liu, “Three-dimensional reconstruction using an adaptive simultaneous algebraic reconstruction technique in electron tomography,” *Journal of Structural Biology*, vol. 175, Oct 2011.

[71] T. Strohmer and R. Vershynin, “A randomized kaczmarz algorithm with exponential convergence,” *Journal of Fourier Analysis and Applications*, vol. 15, no. 2, p. 262–278, 2008.

[72] F. Natterer, *The mathematics of Computerized Tomography*. SIAM, 2001.

[73] P. C. Hansen and J. S. Jørgensen, “Air tools ii: Algebraic iterative reconstruction methods, improved implementation,” *Numerical Algorithms*, vol. 79, no. 1, p. 107–137, 2017.

[74] T. Elfving and T. Nikazad, “Stopping rules for landweber-type iteration,” *Inverse Problems*, vol. 23, no. 4, p. 1417–1432, 2007.

[75] N. A. Worth and T. B. Nickels, “Acceleration of tomo-piv by estimating the initial volume intensity distribution,” *Experiments in Fluids*, vol. 45, no. 5, p. 847–856, 2008.

[76] M. Novara, K. J. Batenburg, and F. Scarano, “Motion tracking-enhanced mart for tomographic piv,” *Measurement Science and Technology*, vol. 21, no. 3, p. 035401, 2010.

[77] G. Cimmino, “Cacolo approssimato per le soluzioni dei systemi di equazioni lineari,” *La Ricerca Scientifica (Roma)*, vol. 1, pp. 326–333, 1938.

[78] L. Landweber, “An iteration formula for fredholm integral equations of the first kind,” *American Journal of Mathematics*, vol. 73, no. 3, p. 615, 1951.

[79] M. Fox, *Quantum Optics: An introduction*. Oxford University Press, 2013.

[80] L. A. Shepp and Y. Vardi, “Maximum likelihood reconstruction for emission tomography,” *IEEE Transactions on Medical Imaging*, vol. MI-1, p. 113–122, Oct 1982.

[81] K. Lange, R. Carson, *et al.*, “Em reconstruction algorithms for emission and transmission tomography,” *J Comput Assist Tomogr*, vol. 8, no. 2, pp. 306–16, 1984.

[82] G. Herman and L. Meyer, “Algebraic reconstruction techniques can be made computationally efficient (positron emission tomography application),” *IEEE Transactions on Medical Imaging*, vol. 12, no. 3, pp. 600–609, 1993.

[83] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*. Applied Mathematical Sciences, Springer New York, 2006.

[84] I. Jérôme, *Bayesian approach to inverse problems*. ISTE, 2010.

[85] A. N. Tikhonov, “Solution of incorrectly formulated problems and the regularization method,” *Soviet Math. Dokl*, vol. 4, pp. 1035–1038, 1963.

[86] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, 1995.

[87] P. C. Hansen, “Analysis of discrete ill-posed problems by means of the l-curve,” *SIAM Review*, vol. 34, no. 4, pp. 561–580, 1992.

[88] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[89] S. J. Grauer, A. Unterberger, A. Rittler, K. J. Daun, A. M. Kempf, and K. Mohri, “Instantaneous 3d flame imaging by background-oriented schlieren tomography,” *Combustion and Flame*, vol. 196, p. 284–299, 2018.

[90] H. Wang and L. Tang, “Total variation regularization used in electrical capacitance tomography,” *AIP Conference Proceedings*, 2007.

[91] M. Ertas, I. Yildirim, M. Kamasak, and A. Akan, “An iterative tomosynthesis reconstruction using total variation combined with non-local means filtering,” *BioMedical Engineering OnLine*, vol. 13, no. 65, 2014.

[92] G. González, V. Kolehmainen, and A. Seppänen, “Isotropic and anisotropic total variation regularization in electrical impedance tomography,” *Computers & Mathematics with Applications*, vol. 74, no. 3, pp. 564–576, 2017.

[93] K. J. Daun, S. J. Grauer, and P. J. Hadwin, “Chemical species tomography of turbulent flows: Discrete ill-posed and rank deficient problems and the use of prior information,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 172, pp. 58–74, 2016. Eurotherm Conference No. 105: Computational Thermal Radiation in Participating Media V.

[94] J. P. Kaipio, V. Kolehmainen, M. Vauhkonen, and E. Somersalo, “Inverse problems with structural prior information,” *Inverse Problems*, vol. 15, no. 4, p. 1111–1111, 1999.

[95] K. Levenberg, “A method for the solution of certain problems in least squares,” *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[96] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[97] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.

[98] W. Fang, “A nonlinear image reconstruction algorithm for electrical capacitance tomography,” *Measurement Science and Technology*, vol. 15, no. 10, p. 2124–2132, 2004.

[99] M. Molinari, S. J. Cox, B. H. Blott, and G. J. Daniell, “Comparison of algorithms for non-linear inverse 3d electrical tomography reconstruction,” *Physiological Measurement*, vol. 23, no. 1, p. 95–104, 2002.

[100] W. H. Press, S. A. Teukoisky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.

[101] O. Kramer, *Genetic Algorithm Essentials*, vol. 679 of *Studies in Computational Intelligence*. Springer Nature, 2017.

[102] K. D. Kihm and D. P. Lyons, “Optical tomography using a genetic algorithm,” *Optics Letters*, vol. 21, no. 17, p. 1327, 1996.

[103] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi *Science*, vol. 220, p. 671–680, May 1983.

[104] W. Cai and L. Ma, “Applications of critical temperature in minimizing functions of continuous variables with simulated annealing algorithm,” *Computer Physics Communications*, vol. 181, no. 1, p. 11–16, 2010.

[105] A. N. Kolmogorov, “The local structure of turbulence in incompressible viscous fluid for very large reynolds number,” *Dokl. Akad. Nauk SSSR*, vol. 30, no. 4, pp. 301–305, 1941.

[106] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[107] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[108] Y. LeCun, Y. Bengio, and G. Hinton, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[109] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, “Machine learning for fluid mechanics,” *Annual Review of Fluid Mechanics*, vol. 52, no. 1, pp. 477–508, 2020.

[110] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[111] C. Teo, K. Lim, G. Hong, and M. Yeo, “A neural net approach in analyzing photograph in piv,” *Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 1535–1538, 1991.

[112] K. Fukami, K. Fukagata, and K. Taira, “Super-resolution analysis via machine learning: A survey for fluid flows,” *Theoretical and Computational Fluid Dynamics*, 2023.

[113] B. Vennemann and T. Rösgen, “A dynamic masking technique for particle image velocimetry using convolutional autoencoders,” *Experiments in Fluids*, vol. 61, no. 7, 2020.

[114] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[115] K. Hornik, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[116] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?,” in *2009 IEEE 12th International Conference on Computer Vision*, pp. 2146–2153, 2009.

[117] M. T. McCann, K. H. Jin, and M. Unser, “Convolutional neural networks for inverse problems in imaging: A review,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, p. 85–95, 2017.

[118] C. Wei, K. K. Schwarm, D. I. Pineda, and R. M. Spearrin, “Deep neural network inversion for 3d laser absorption imaging of methane in reacting flows,” *Optics Letters*, vol. 45, no. 8, p. 2447, 2020.

[119] C. Wei, K. K. Schwarm, D. I. Pineda, and R. Mitchell Spearrin, “Physics-trained neural network for sparse-view volumetric laser absorption imaging of species and temperature in reacting flows,” *Optics Express*, vol. 29, no. 14, p. 22553, 2021.

[120] W. Cai, J. Huang, A. Deng, and Q. Wang, "Volumetric reconstruction for combustion diagnostics via transfer learning and semi-supervised learning with limited labels," *Aerospace Science and Technology*, vol. 110, p. 106487, 2021.

[121] W. Yifan, F. Serena, S. Wu, C. Öztireli, and O. Sorkine-Hornung, "Differentiable surface splatting for point-based geometry processing," *ACM Transactions on Graphics*, vol. 38, no. 6, p. 1–14, 2019.

[122] V. Sitzmann, J. Thies, F. Heide, M. NieBner, G. Wetzstein, and M. Zollhofer, "Deepvoxels: Learning persistent 3d feature embeddings," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[123] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes," *ACM Transactions on Graphics*, vol. 38, pp. 1–14, aug 2019.

[124] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[125] R. Chabra, J. E. Lenssen, E. Ilg, T. Schmidt, J. Straub, S. Lovegrove, and R. A. Newcombe, "Deep local shapes: Learning local SDF priors for detailed 3d reconstruction," *CoRR*, vol. abs/2003.10983, 2020.

[126] J. Chibane, A. Mir, and G. Pons-Moll, "Neural unsigned distance fields for implicit function learning," 2020.

[127] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[128] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[129] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[130] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *ACM SIGGRAPH Computer Graphics*, vol. 18, no. 3, p. 165–174, 1984.

[131] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," *In: ICML*, May 2018.

[132] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5835–5844, 2021.

[133] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," 2021.

[134] C. Lin, W. Ma, A. Torralba, and S. Lucey, "BARF: bundle-adjusting neural radiance fields," *CoRR*, vol. abs/2104.06405, 2021.

[135] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," 2020.

[136] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," 2020.

[137] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[138] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "Pixelnerf: Neural radiance fields from one or few images," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[139] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, "Fastnerf: High-fidelity neural rendering at 200fps," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[140] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "Graf: Generative radiance fields for 3d-aware image synthesis," 2021.

[141] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[142] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, "Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[143] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Deformable neural radiance fields," *CoRR*, vol. abs/2011.12948, 2020.

[144] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. Chaitanya, A. Kaplanyan, and M. Steinberger, "Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks," *Computer Graphics Forum*, vol. 40, no. 4, p. 45–59, 2021.

[145] A. Corona-Figueroa, J. Frawley, S. B. Taylor, S. Bethapudi, H. P. Shum, and C. G. Willcocks, "Mednerf: Medical neural radiance fields for reconstructing 3d-aware ct-projections from a single x-ray," *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2022.

[146] D. L. Kelly and B. S. Thurow, "Fluidnerf: A scalar-field reconstruction technique for flow diagnostics using neural radiance fields," *AIAA SCITECH 2023 Forum*, 2023.

[147] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," 2020.

[148] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and et al., “Photo-realistic single image super-resolution using a generative adversarial network,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[149] W. E and B. Yu, “The deep ritz method: A deep learning-based numerical algorithm for solving variational problems,” *Communications in Mathematics and Statistics*, vol. 6, no. 1, p. 1–12, 2018.

[150] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, “Automatic differentiation in machine learning: a survey,” 2018.

[151] S. Kollmannsberger, D. D’Angella, M. Jokeit, and L. Herrmann, *Deep learning in computational mechanics: An introductory course*. Springer, 2021.

[152] D. V. Dung, N. D. Song, P. S. Palar, and L. R. Zuhal, “On the choice of activation functions in physics-informed neural network for solving incompressible fluid flows,” *AIAA SCITECH 2023 Forum*, 2023.

[153] L. Sun and J.-X. Wang, “Physics-constrained bayesian neural network for fluid flow reconstruction with sparse and noisy data,” *Theoretical and Applied Mechanics Letters*, vol. 10, no. 3, p. 161–169, 2020.

[154] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, “Scientific machine learning through physics–informed neural networks: Where we are and what’s next,” *Journal of Scientific Computing*, vol. 92, no. 3, 2022.

[155] G. Pang, L. Lu, and G. E. Karniadakis, “fpinns: Fractional physics-informed neural networks,” *SIAM Journal on Scientific Computing*, vol. 41, no. 4, pp. A2603–A2626, 2019.

[156] Z. Fang and J. Zhan, “A physics-informed neural network framework for pdes on 3d surfaces: Time independent problems,” *IEEE Access*, vol. 8, pp. 26328–26335, 2020.

[157] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks for heat transfer problems,” *Journal of Heat Transfer*, vol. 143, no. 6, 2021.

[158] X. Huang, H. Liu, B. Shi, Z. Wang, K. Yang, Y. Li, B. Weng, M. Wang, H. Chu, J. Zhou, F. Yu, B. Hua, L. Chen, and B. Dong, “Solving partial differential equations with point source based on physics-informed neural networks,” 2021.

[159] D. Zhang, L. Guo, and G. E. Karniadakis, “Learning in modal space: Solving time-dependent stochastic pdes using physics-informed neural networks,” *SIAM Journal on Scientific Computing*, vol. 42, no. 2, pp. A639–A665, 2020.

[160] D. Zhang, L. Lu, L. Guo, and G. E. Karniadakis, “Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems,” *Journal of Computational Physics*, vol. 397, p. 108850, 2019.

[161] L. Yang, X. Meng, and G. E. Karniadakis, “B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data,” *Journal of Computational Physics*, vol. 425, p. 109913, 2021.

[162] X. Meng and G. E. Karniadakis, “A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems,” *Journal of Computational Physics*, vol. 401, p. 109020, 2020.

[163] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, “Adaptive activation functions accelerate convergence in deep and physics-informed neural networks,” *Journal of Computational Physics*, vol. 404, p. 109136, 2020.

[164] S. Wang, Y. Teng, and P. Perdikaris, “Understanding and mitigating gradient pathologies in physics-informed neural networks,” 2020.

[165] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, “Physics-informed neural networks with hard constraints for inverse design,” *SIAM Journal on Scientific Computing*, vol. 43, no. 6, pp. B1105–B1132, 2021.

[166] H. Gao, L. Sun, and J.-X. Wang, “Phygeonet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain,” *Journal of Computational Physics*, vol. 428, p. 110079, 2021.

[167] S. Wang, X. Yu, and P. Perdikaris, “When and why pinns fail to train: A neural tangent kernel perspective,” *Journal of Computational Physics*, vol. 449, p. 110768, 2022.

[168] S. Wang, H. Wang, and P. Perdikaris, “On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 384, p. 113938, 2021.

[169] S. Mishra and R. Molinaro, “Estimates on the generalization error of physics informed neural networks (pinns) for approximating a class of inverse problems for pdes,” 2021.

[170] J. P. Molnar, L. Venkatakrishnan, B. E. Schmidt, T. A. Sipkens, and S. J. Grauer, “Estimating density, velocity, and pressure fields in supersonic flows using physics-informed bos,” *Experiments in Fluids*, vol. 64, no. 1, 2023.

[171] K. Zhou, J. Li, J. Hong, and S. J. Grauer, “Stochastic particle advection velocimetry (spav): Theory, simulations, and proof-of-concept experiments,” *Measurement Science and Technology*, vol. 34, no. 6, p. 065302, 2023.

[172] D. Di Carlo, D. Heitz, and T. Corpetti, “Post-processing subsonic flows using physics-informed neural networks,” *20th International Symposium on Application of Laser and Imaging Techniques to Fluid Mechanics*, Jul 2022.

[173] A. D. Cutler, G. Magnotti, L. Cantu, E. Gallo, R. Rockwell, and C. Goyne, “Dual-pump coherent anti-stokes raman spectroscopy measurements in a dual-mode scramjet,” *Journal of Propulsion and Power*, vol. 30, no. 3, p. 539–549, 2014.

[174] F. De Domenico, T. F. Guiberti, S. Hochgreb, W. L. Roberts, and G. Magnotti, “Tracer-free laser-induced grating spectroscopy using a pulse burst laser at 100 khz,” *Optics Express*, vol. 27, no. 22, p. 31217, 2019.

[175] R. K. Hanson, “Combustion diagnostics: Planar imaging techniques,” *Symposium (International) on Combustion*, vol. 21, no. 1, p. 1677–1691, 1988.

[176] A. Araújo, “Multi-spectral pyrometry—a review,” *Measurement Science and Technology*, vol. 28, no. 8, p. 082002, 2017.

[177] H. M. Hertz and G. W. Faris, “Emission tomography of flame radicals,” *Optics Letters*, vol. 13, no. 5, p. 351, 1988.

[178] R. OBERTACKE, H. WINTRICH, F. WINTRICH, and A. LEIPERTZ, “A new sensor system for industrial combustion monitoring and control using uv emission spectroscopy and tomography,” *Combustion Science and Technology*, vol. 121, no. 1–6, p. 133–151, 1996.

[179] J. Floyd and A. Kempf, “Computed tomography of chemiluminescence (ctc): High resolution and instantaneous 3-d measurements of a matrix burner,” *Proceedings of the Combustion Institute*, vol. 33, no. 1, p. 751–758, 2011.

[180] H. Liu, T. Yu, M. Zhang, and W. Cai, “Demonstration of 3d computed tomography of chemiluminescence with a restricted field of view,” *Applied Optics*, vol. 56, no. 25, p. 7107, 2017.

[181] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, “Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations,” *Journal of Computational Physics*, vol. 426, p. 109951, 2021.

[182] Y. ISHINO and N. OHIWA, “Three-dimensional computerized tomographic reconstruction of instantaneous distribution of chemiluminescence of a turbulent premixed flame,” *JSME International Journal Series B*, vol. 48, no. 1, p. 34–40, 2005.

[183] T. L. Bergman, D. P. Dewit, A. S. Lavine, and F. P. Incropera, *Fundamentals of heat and Mass Trasnfer*. John Wiley & Sons, 2011.

[184] T. Jenkins and R. Hanson, “Soot pyrometry using modulated absorption/emission,” *Combustion and Flame*, vol. 126, no. 3, p. 1669–1679, 2001.

[185] N. Hagen and M. W. Kudenov, “Review of snapshot spectral imaging technologies,” *Optical Engineering*, vol. 52, no. 9, p. 090901, 2013.

[186] H.-C. Zhou, C. Lou, Q. Cheng, Z. Jiang, J. He, B. Huang, Z. Pei, and C. Lu, “Experimental investigations on visualization of three-dimensional temperature distributions in a large-scale pulverized-coal-fired boiler furnace,” *Proceedings of the Combustion Institute*, vol. 30, no. 1, p. 1699–1706, 2005.

[187] X. Huang, H. Qi, C. Niu, L. Ruan, H. Tan, J. Sun, and C. Xu, “Simultaneous reconstruction of 3d temperature distribution and radiative properties of participating media based on the multi-spectral light-field imaging technique,” *Applied Thermal Engineering*, vol. 115, pp. 1337–1347, 2017.

[188] R. J. Hall and P. A. Bonczyk, “Sooting flame thermometry using emission/absorption tomography,” *Appl. Opt.*, vol. 29, pp. 4590–4598, Nov 1990.

[189] T. Ren, H. Li, M. F. Modest, and C. Zhao, “Efficient two-dimensional scalar fields reconstruction of laminar flames from infrared hyperspectral measurements with a machine learning approach,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 271, p. 107724, Sep 2021.

[190] F. Cignoli, S. De Iuliis, V. Manta, and G. Zizak, “Two-dimensional two-wavelength emission technique for soot diagnostics,” *Applied Optics*, vol. 40, no. 30, p. 5370, 2001.

[191] N. J. Kempema and M. B. Long, “Effect of soot self-absorption on color-ratio pyrometry in laminar coflow diffusion flames,” *Opt. Lett.*, vol. 43, pp. 1103–1106, Mar 2018.

[192] D. R. Snelling, K. A. Thomson, G. J. Smallwood, O. L. Gulder, E. J. Weckman, and R. A. Fraser, “Spectrally resolved measurement of flame radiation to determine soot temperature and concentration,” *AIAA Journal*, vol. 40, p. 1789–1795, 2002.

[193] F. Liu, K. A. Thomson, and G. J. Smallwood, “Soot temperature and volume fraction retrieval from spectrally resolved flame emission measurement in laminar axisymmetric

coflow diffusion flames: Effect of self-absorption," *Combustion and Flame*, vol. 160, no. 9, pp. 1693–1705, 2013.

[194] M. P. Freeman and S. Katz, "Determination of the radial distribution of brightness in a cylindrical luminous medium with self-absorption," *Journal of the Optical Society of America*, vol. 50, no. 8, p. 826, 1960.

[195] H. Liu, S. Zheng, H. Zhou, and C. Qi, "Measurement of distributions of temperature and wavelength-dependent emissivity of a laminar diffusion flame using hyper-spectral imaging technique," *Measurement Science and Technology*, vol. 27, no. 2, p. 025201, 2015.

[196] H. Michelsen, C. Schulz, G. Smallwood, and S. Will, "Laser-induced incandescence: Particulate diagnostics for combustion, atmospheric, and industrial applications," *Progress in Energy and Combustion Science*, vol. 51, pp. 2–48, 2015.

[197] S. Will, S. Schraml, and A. Leipert, "Comprehensive two-dimensional soot diagnostics based on laser-induced incandescence (lii)," *Symposium (International) on Combustion*, vol. 26, no. 2, p. 2277–2284, 1996.

[198] E. Cenker, G. Bruneaux, T. Dreier, and C. Schulz, "Sensitivity analysis for soot particle size imaging with laser-induced incandescence at high pressure," *Applied Physics B*, vol. 119, no. 4, p. 745–763, 2015.

[199] G. Herzberg, S. J. W. T., and K. P. Huber, *Molecular spectra and molecular structure*. Reitell Press, 2007.

[200] R. K. Hanson, C. S. Goldenstein, and R. M. Spearrin, *Spectroscopy and optical diagnostics for gases*. Springer, 2016.

[201] I. van Cruyningen, A. Lozano, and R. K. Hanson, "Quantitative imaging of concentration by planar laser-induced fluorescence," *Experiments in Fluids*, vol. 10, no. 1, p. 41–49, 1990.

[202] J. M. Seitzman, G. Kychakoff, and R. K. Hanson, “Instantaneous temperature field measurements using planar laser-induced fluorescence,” *Optics Letters*, vol. 10, no. 9, p. 439, 1985.

[203] Y. Zhao, C. Tong, and L. Ma, “Demonstration of a new laser diagnostic based on photodissociation spectroscopy for imaging mixture fraction in a non-premixed jet flame,” *Applied Spectroscopy*, vol. 64, no. 4, p. 377–383, 2010.

[204] R. P. Lucht, N. M. Laurendeau, D. W. Sweeney, M. C. Drake, M. Lapp, and R. Pitz, “Single-pulse laser-saturated fluorescence measurements of oh in turbulent nonpremixed flames,” *Conference on Lasers and Electro-Optics*, 1983.

[205] M. J. Dyer and D. R. Crosley, “Two-dimensional imaging of oh laser-induced fluorescence in a flame,” *Optics Letters*, vol. 7, no. 8, p. 382, 1982.

[206] Y. Wu, W. Xu, Q. Lei, and L. Ma, “Single-shot volumetric laser induced fluorescence (vlif) measurements in turbulent flows seeded with iodine,” *Optics Express*, vol. 23, no. 26, p. 33408, 2015.

[207] G. Kychakoff, P. H. Paul, I. van Cruyningen, and R. K. Hanson, “Movies and 3-d images of flowfields using planar laser-induced fluorescence,” *Applied Optics*, vol. 26, no. 13, p. 2498, 1987.

[208] W. Xu, C. D. Carter, S. Hammack, and L. Ma, “Analysis of 3d combustion measurements using ch-based tomographic vlif (volumetric laser induced fluorescence),” *Combustion and Flame*, vol. 182, p. 179–189, 2017.

[209] T. R. Meyer, G. F. King, G. C. Martin, R. P. Lucht, F. R. Schauer, and J. C. Dutton, “Accuracy and resolution issues in no/acetone plif measurements of gas-phase molecular mixing,” *Experiments in Fluids*, vol. 32, no. 6, p. 603–611, 2002.

[210] B. Di Bella, M. Khatamifar, and W. Lin, “Experimental study of flow visualisation using fluorescent dye,” *Flow Measurement and Instrumentation*, vol. 87, p. 102231, 2022.

[211] C. T. Johansen, D. A. Lincoln, B. F. Bathel, J. A. Inman, and P. M. Danehy, “Simultaneous nitric oxide/atomic oxygen laser-induced fluorescence in an arcjet facility,” *Journal of Thermophysics and Heat Transfer*, vol. 30, p. 912–918, Oct 2016.

[212] L. Ma, Q. Lei, T. Capil, S. D. Hammack, and C. D. Carter, “Direct comparison of two-dimensional and three-dimensional laser-induced fluorescence measurements on highly turbulent flames,” *Optics Letters*, vol. 42, no. 2, p. 267, 2017.

[213] T. Li, J. Pareja, F. Fuest, M. Schütte, Y. Zhou, A. Dreizler, and B. Böhm, “Tomographic imaging of oh laser-induced fluorescence in laminar and turbulent jet flames,” *Measurement Science and Technology*, vol. 29, no. 1, p. 015206, 2017.

[214] T. Yu, H. Liu, and W. Cai, “On the quantification of spatial resolution for three-dimensional computed tomography of chemiluminescence,” *Opt. Express*, vol. 25, pp. 24093–24108, Oct 2017.

[215] Y. Wu, W. Xu, and L. Ma, “Kilohertz vlif (volumetric laser induced fluorescence) measurements in a seeded free gas-phase jet in the transitionally turbulent flow regime,” *Optics and Lasers in Engineering*, vol. 102, p. 52–58, 2018.

[216] Q. Wang, H. Liu, X. Liu, S. Wang, C. Fu, G. Wang, Y. Gao, W. Cai, and F. Qi, “Three-dimensional concentration field imaging in a swirling flame via endoscopic volumetric laser-induced fluorescence at 10-khz-rate,” *Science China Technological Sciences*, vol. 63, no. 10, p. 2163–2168, 2020.

[217] M. Gomez, B. W. Yant, M. N. Slipchenko, A. M. Braun, Z. D. Rancilio, T. R. Meyer, and S. Roy, “Four-dimensional laser-induced fluorescence and tomography of liquids,” *International Journal of Multiphase Flow*, vol. 166, p. 104501, 2023.

[218] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, “Fourier features let networks learn high frequency functions in low dimensional domains,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[219] N. Max, “Optical models for direct volume rendering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, p. 99–108, 1995.

[220] N. Sharan and J. Bellan, “Investigation of high-pressure turbulent jets using direct numerical simulation,” *Journal of Fluid Mechanics*, vol. 922, 2021.

[221] T. W. Fahringer and B. S. Thurow, “Plenoptic particle image velocimetry with multiple plenoptic cameras,” *Measurement Science and Technology*, vol. 29, no. 7, 2018.

[222] C. Clifford and B. Thurow, “Scalar-field reconstruction algorithms using plenoptic cameras,” *Applied Optical Metrology III*, 2019.

[223] R. Hartley and A. Zisserman, *Multiple view geometry in Computer Vision*. Cambridge University Press, 2019.

[224] E. M. Hall, T. W. Fahringer, B. S. Thurow, and D. R. Guildenbecher, “Volumetric calibration of a plenoptic camera,” *55th AIAA Aerospace Sciences Meeting*, 2017.

[225] D. C. Brown, “Decentering distortion of lenses,” *Photogrammetric Engineering*, 1966.

[226] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, p. 1330–1334, 2000.

[227] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, “Complete solution classification for the perspective-three-point problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.