

University of Kentucky

UKnowledge

Theses and Dissertations--Computer Science

Computer Science


2024

Flexible Attenuation Fields: Tomographic Reconstruction From Heterogeneous Datasets

Clifford S. Parker

University of Kentucky, c.seth.parker@uky.edu

Author ORCID Identifier:

 <https://orcid.org/0000-0002-3887-1237>

Digital Object Identifier: <https://doi.org/10.13023/etd.2024.71>

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

Recommended Citation

Parker, Clifford S., "Flexible Attenuation Fields: Tomographic Reconstruction From Heterogeneous Datasets" (2024). *Theses and Dissertations--Computer Science*. 143.

https://uknowledge.uky.edu/cs_etds/143

This Doctoral Dissertation is brought to you for free and open access by the Computer Science at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Computer Science by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Clifford S. Parker, Student

Dr. W. Brent Seales, Major Professor

Dr. Simone Silvestri, Director of Graduate Studies

FLEXIBLE ATTENUATION FIELDS: TOMOGRAPHIC RECONSTRUCTION
FROM HETEROGENEOUS DATASETS

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
C. Seth Parker
Lexington, Kentucky
Director: Dr. W. Brent Seales, Professor of Computer Science
Lexington, Kentucky
2024

Copyright © C. Seth Parker 2024
<https://orcid.org/0000-0002-3887-1237>

ABSTRACT OF DISSERTATION

FLEXIBLE ATTENUATION FIELDS: TOMOGRAPHIC RECONSTRUCTION FROM HETEROGENEOUS DATASETS

Traditional reconstruction methods for X-ray computed tomography (CT) are highly constrained in the variety of input datasets they admit. Many of the imaging settings – the incident energy, field-of-view, effective resolution – remain fixed across projection images, and the only real variance is in the detector’s position and orientation with respect to the scene. In contrast, methods for 3D reconstruction of natural scenes are extremely flexible to the geometric and photometric properties of the input datasets, readily accepting and benefiting from images captured under varying lighting conditions, with different cameras, and at disparate points in time and space. Extending CT to support similar degrees of flexibility would significantly enhance what can be learned from tomographic datasets. We propose that traditionally complicated or time-consuming tomographic tasks, such as multi-resolution and multi-energy analysis, can be more readily achieved with a reconstruction framework which explicitly accepts datasets with varied imaging settings. This work presents a CT reconstruction framework specifically designed for datasets with heterogeneous capture properties which we call Flexible Attenuation Fields (FlexAF). Built on differentiable ray tracing and continuous neural volumes, FlexAF accepts X-ray images captured from any position and orientation in the world coordinate frame, including images which differ in size, resolution, field-of-view, and photometric settings. This method produces reconstructions for regular CT scans which are comparable to those produced by filtered backprojection, demonstrating that additional flexibility does not fundamentally hinder the ability to reconstruct high-quality volumes. Our experiments test the expanded capabilities of FlexAF for addressing challenging reconstruction tasks, including automatic camera calibration and reconstruction of multi-resolution and multi-energy volumes.

KEYWORDS: Computed Tomography, Tomographic Reconstruction, Machine Learning, Implicit Neural Representations, Volumetric Imaging

C. Seth Parker

04/15/2024

Date

FLEXIBLE ATTENUATION FIELDS: TOMOGRAPHIC RECONSTRUCTION
FROM HETEROGENEOUS DATASETS

By
C. Seth Parker

Dr. W. Brent Seales

Director of Dissertation

Dr. Simone Silvestri

Director of Graduate Studies

04/15/2024

Date

For those who showed me the way,
Lawrence and Juanita and William and Frances.

ACKNOWLEDGEMENTS

Every writer faces the dilemma of fitting their infinite list of supporters into the limited space of a page, and this writer is no exception. This work would not have been possible without a strong network of colleagues, friends, and family who helped me push this boulder up the mountain. With humble apologies to those who I have failed to remember, I wish to thank...

The University of Kentucky Center for Computational Sciences and Information Technology Services Research Computing for their support and use of the Lipscomb Compute Cluster and associated research computing resources which generated most of the results you'll find in the following pages.

My computed tomography mentors over many long years: Benjamin Ache, Evi Bongaers, Leigh Connor, Frederik Coppens, Raj Manoharan, Alexander Sassov, and Nghia Vo.

My current and former VisCenter, Computer Science, DRI, and EduceLab colleagues, especially Christy Chapman, Mami Hayashida, Ankan Bhattacharyya, Beth Lutin, Paul Linton, Ben Corwin, Kristina Gessel, Jacob Chappell, Aaron Camenisch, and Julie Martinez for being major contributors to some of the most wonderfully interesting years of my life.

Stephen Parsons, whose ability to help my focus my technical ramblings is only overshadowed by his tremendous kindness and friendship.

Brent Seales, my advisor, colleague, and friend, who saw something inside this media nerd that he couldn't see in himself.

My massively supportive family, who provided me with a quiet place to work, plied me with coffee so that I would stay awake, and prayed me through the final days of this grueling trial. Particular attention must be paid to my wife and daughter, Elizabeth and Penny, my world above the world, whose continuous encouragement through every step of my unexpected journey carried me through to the end.

Finally, He who gives me strength in all things, without whom this work would simply be a pile of text.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
1 INTRODUCTION	1
2 BACKGROUND	7
2.1 Photogrammetry	7
2.2 Radiance fields	11
2.2.1 Neural radiance fields	11
2.2.2 Random Fourier features	14
2.2.3 Approximating ray integrals	16
2.2.4 Learned flexibility	18
2.2.5 Real-time rendering	20
2.3 X-ray imaging and CT reconstruction	20
2.3.1 Backprojection	23
2.3.2 Forward projection	26
2.3.3 Attenuation field CT reconstruction	29
2.3.4 X-ray camera geometry	31
2.4 A look back with optimism	35
3 FRAMEWORK	37
3.1 X-ray camera model	38
3.2 Ray sampling	42
3.2.1 Pencil approximation	44
3.2.2 Learned camera extrinsics	46
3.3 Volume model	47
3.3.1 Positional encoding	47
3.3.2 Standard network	52
3.3.3 Multi-energy network	52
3.4 Training and evaluation	55
3.4.1 Slice rendering	55
3.5 Attention mechanisms	56
3.5.1 Entropy pixels	56
3.5.2 Adjustable bounding volume	58
4 DATASETS	62
4.1 The Shepp-Logan phantom	62
4.2 The papyrus scroll dataset	64
4.3 The Multi* dataset	65
4.3.1 The Multi* proxy	66
4.3.2 Acquisition settings	70
4.4 Dataset formats	73
4.4.1 Heterogeneous dataset construction	73
5 EXPERIMENTS	75

5.1	Standard reconstruction	76
5.1.1	Shepp-Logan	77
5.1.2	Papyrus scroll	80
5.1.3	Multi* proxy	82
5.2	Automatic extrinsic calibration	87
5.2.1	Calibration of parallel geometries	91
5.3	Multi-resolution reconstruction	92
5.3.1	Combining regions of interest	92
5.3.2	ROI reconstruction	94
5.4	Multi-energy reconstruction	96
5.4.1	Interleaved energies	98
5.4.2	Alternative volume views	100
5.4.3	Energy wedges	101
5.5	On performance	102
6	DISCUSSION	106
6.1	The challenges of projective X-ray cameras	106
6.2	Freely-defined trajectories	107
6.3	Approximating ray integrals	108
6.4	Building a better model	109
6.5	Runtime performance	111
6.6	Spectral tomography	112
6.7	Low-dosage, high-resolution reconstruction	112
6.8	Unified volume models	113
6.9	Conclusion	114
	ENLARGED FIGURES	117
	BIBLIOGRAPHY AND FURTHER READING	119
	VITA	127

LIST OF TABLES

4.1	Scans and parameters in the Multi* dataset.	71
5.1	Standard reconstruction result metrics.	76
5.2	Estimated time required to train the full MS.01.02 dataset to convergence using various Nvidia GPUs.	104

LIST OF FIGURES

1.1	<i>Hand mit Ringen (Hand with Rings)</i> by Wilhelm Röntgen (1895). . .	3
2.1	Epipolar geometry and triangulation in photogrammetry.	9
2.2	The stages of photogrammetric reconstruction.	10
2.3	The neural radiance field (NeRF) training and evaluation process. . .	13
2.4	The Mip-NeRF ray sampling method.	17
2.5	X-ray mass attenuation coefficient as a function of incident energy. . .	21
2.6	The central slice theorem and its use for CT reconstruction.	24
2.7	A comparison of the photographic camera model and the geometry of cone beam X-ray imaging.	32
2.8	The geometry of parallel beam X-ray imaging.	33
3.1	The primary components of the FlexAF framework.	38
3.2	Cone and parallel beam X-ray camera models in FlexAF.	39
3.3	Initialization of X-ray cameras from a cylindrical CT scan.	40
3.4	A visualization of the FlexAF ray tracing system.	42
3.5	Plot of various interval schedules across training epochs.	43
3.6	Per-pixel pencil sampling strategies for parallel and cone beam geome- tries.	45
3.7	Misalignment artifacts in the papyrus scroll dataset.	47
3.8	The standard FlexAF neural volume architecture.	48
3.9	X-ray mass attenuation coefficients plotted against the atomic number (Z) for various incident energies.	53
3.10	The multi-energy FlexAF neural volume architecture.	54
3.11	Entropy pixels example for the Multi* dataset MS.01.02.	57
3.12	Plot of the potential training bounding volumes with respect to the area of the reconstruction.	59
3.13	The effect of adjusting the bounding volume on reconstruction quality.	60
4.1	Projections and slices of the Shepp-Logan phantom.	63
4.2	Images of the papyrus scroll dataset.	65
4.3	Images of the Multi* proxy and dataset.	66
4.4	A diagram of the faces and embedded materials of the Multi* proxy.	68
4.5	Objects of interest inside the Multi* proxy CT reconstructions. . . .	69
5.1	Evaluating Shepp-Logan reconstructions using FBP and FlexAF. . .	78
5.2	Comparing FlexAF reconstructions across various model configurations.	79
5.3	Comparison of FlexAF reconstructions for the papyrus scroll dataset.	80
5.4	Volume renderings of 12 slices from the papyrus scroll comparing FBP to FlexAF.	82
5.5	Comparison of FlexAF reconstructions for the MS.01.01 dataset. . . .	83
5.6	Volume renderings of 100 slices from MS.01.01 comparing FBP to FlexAF.	84

5.7	Comparison of FlexAF reconstructions for the MS.01.02 dataset. . . .	86
5.8	Testing the Gaussian frequency filter for automatic extrinsic calibration using a 1:4 scale papyrus scroll dataset.	88
5.9	Automatic extrinsic calibration of the papyrus scroll dataset at full resolution.	89
5.10	Automatic extrinsic calibration parallel projection images using the Shepp-Logan phantom.	91
5.11	Masking CT projection images to construct a reconstruction region of interest.	93
5.12	Resulting reconstructions from the multi-resolution ROI experiments.	95
5.13	Visualization of the dataset combinations for the multi-energy experiments.	97
5.14	Reconstructions for the five incident energies in the interleaved multi-energy experiment.	98
5.15	Comparing the reconstructed attenuation coefficients produced by FBP and FlexAF for the interleaved multi-energy experiments.	99
5.16	z -value slice for the interleaved multi-energy experiment.	101
5.17	Using the interleaved multi-energy model to interpolate between incident X-ray energies.	102
5.18	Multi-energy reconstruction results using the energy wedges training method.	103
A.1	Comparison of FBP and FlexAF slices for the MS.01.01 reconstructions.	117
A.2	Comparison of FBP and FlexAF slices for the MS.01.02 reconstructions.	117
A.3	Comparison of the multi-resolution reconstructions.	118

CHAPTER 1. INTRODUCTION

In late December 1895, Wilhelm Röntgen brought his wife, Anna Bertha, to his modest laboratory in Würzburg, Germany, for what surely must have seemed an unusual experiment. For many weeks, Röntgen had worked tirelessly in his lab to measure and explore the properties of an exciting new discovery, a new type of “light” emitted by a Crookes tube which could not be seen but the presence of which could be detected in the fluorescent glow it induced on barium platinocyanide paper and the shadows it left on photographic paper. In truth, Röntgen could not even be sure that his discovery should be called light, for the new phenomenon could easily penetrate paper, wood, and metal and could not be reflected or refracted by any known means. A true experimentalist who was unwilling to make claims without strong evidence, Röntgen had decided to call his discovery “X-rays,” choosing the “X” to stand for the many unknowns that surrounded his work.

Leading Anna Bertha to a small wooden table, Wilhelm placed a Crookes tube underneath the table’s surface, then placed her hand flat against its top [14]. Taking up a photographic glass plate wrapped in black paper from a nearby shelf, he placed it on top of Anna’s hand and initiated an electrical current to the Crookes tube. For half an hour or more, Anna’s hand rested on the table as the photographic plate captured the silent, invisible flight of the mysterious X-rays passing through her fingers. Little did Anna know that these uncomfortable moments, so strange and unassuming in their simplicity, would generate one of the most impactful experimental results of the late 19th century, one that would echo through scientific and medical study for generations.

Today, X-ray imaging is a standard practice in airports, hospitals, factories, and laboratories all over the world. The ubiquity of traditional 2D radiography in the healthcare system is almost incalculable, as X-ray images are regularly used to diag-

nose health issues such as tooth decay, bone fractures, and pneumonia. The world-wide use of medical X-ray computed tomography (CT) scans, the radiograph's three-dimensional (3D) counterpart, has grown steadily over the last decade [61], and in 2021, an estimated 84 million medical CT scans were performed in the United States alone.¹ Likewise, industrial CT is regularly deployed in the automotive, aerospace, and electronics industries for non-destructive testing (NDT) and inspection [16]. It is safe to say that there is very little of our modern life which has not been touched by X-rays.

Of course, radiography's first patient was not to know the part she had just played in history. Upon seeing the ghostly image of her skeletal hand (Figure 1.1), complete with a ring upon the fourth finger, Anna exclaimed, "I have seen my death," [54] and refused to enter her husband's laboratory again. It is hard to blame Frau Röntgen for her reaction. Even today, it can seem equal parts magic and miracle that we should be able to make bare the interiors of objects which normally remain hidden, or further, that those hidden spaces are revealed as incredibly detailed, 3D volumes at the push of a button. CT regularly enables us to pinpoint disease in the human body, inspect complex machinery for defects, and recover 2,000-year-old text from ancient scrolls, all without the risk of invasive damage to the subject being scanned. It is *science*, yes, but it is also *magic*.

For those who work regularly with modern CT systems, it is hard not to envy the freedom which Wilhelm Röntgen enjoyed during his early experiments. Röntgen's spartan imaging setups — a tube, a table, a photographic plate, and his wife's willingness — stand in stark contrast to the complexity of today's tomographic equipment. Commercial CT scanners are exactingly engineered to produce stunning images, but they remain large, bulky, and expensive devices that require stable environmental conditions and regular calibration to maintain their pristine image quality. While

¹254.6 CT scans per 1,000 population [61] with a total United States population of 331 million [8].



Figure 1.1: *Hand mit Ringen (Hand with Rings)* by Wilhelm Röntgen (1895). This print possibly depicts the hand of Anna Bertha Röntgen and is considered the first medical X-ray radiograph [82]. This work is licensed under the Creative Commons Attribution NonCommercial 4.0 International License (CC BY-NC 4.0).

Röntgen admittedly only worked in two dimensions and was unaware of the harmful ionizing effects of his discovery, the modern radiologist could be forgiven for looking at his methods and wishing for a CT scanner light enough to be carried to a patient's room or packed into a hand case and taken to a local clinic. And so the question must be asked: what's stopping us?

Part of the answer to this question can be found in the algorithms which enable CT in the first place. Today's frameworks for tomographic reconstruction provide strong guarantees for what can be recovered with tomography, but at the cost of strong constraints on the datasets which are supported. Many of the imaging settings — the incident energy, field-of-view, effective resolution — must remain fixed across projection images, and the only real variance is in the detector's position and orientation with respect to a known center of rotation. Unexpected subject movement during a scan, fluctuations in exposure, or too much mechanical misalignment can result in reconstruction artifacts in the best cases or unreconstructable scans in the worst.

As a consequence, the entire ecosystem of computed tomography has oriented itself around the quest for the “golden dataset,” that perfectly captured scan without deviation, blemish, or error. The scanners grow larger to realize more stability, the engineering becomes more exacting to guarantee more precision, and the scanning protocols become more critical to the success of the reconstruction. When the captured resolution is not sufficient, when the field-of-view needs to be widened, when the energy settings do not provide adequate contrast, or when the sample moves halfway through a scan, the solution is often to capture a completely new scan. Further, scans of the same object which are captured with different settings are reconstructed as independent entities, even though much of the structural information about the sample (i.e. the chemical composition) is shared across scans.

This restrictive approach to tomography seems curious when one considers that such constraints are not shared by similar methods for photographic 3D scene recon-

struction. Photogrammetry and neural radiance fields readily accept images captured under varying lighting conditions, with different cameras, and at disparate points in time and space, yet require practically no knowledge of the images’ extrinsic parameters and only minimal information about the cameras’ intrinsics. By their example, these methods beg the question of why similar degrees of flexibility cannot be extended into the realm of tomography.

When Röntgen first published his results, the world exclaimed it a success of *photography*. “The New Marvel in Photography,” declared McClure’s Magazine in the title to a work we will often quote in these pages [14]. “That a new photography has suddenly arisen which can photograph the bones, and before long, the organs of the human body...is news which cannot fail to startle everybody.” But now, more than a century later, we seem to have lost — or at least we downplay — the idea that X-ray imaging is akin to photography and that the X-ray source and detector form a *camera*. Perhaps this kinship should now be reevaluated in the light of promising new 3D reconstruction techniques being developed for the photographic realm.

The rise over the last decade of a new generation of general purpose machine learning and artificial intelligence methods has been meteoric and profound. It is only slight hyperbole to say that we have entered a new era of computation, one that allows us to review our oldest problems through a new lens.²

This dissertation argues that many of the accepted limitations regarding CT dataset homogeneity can now be lifted through the adaptation of neural methods originally developed for photographic scene reconstruction. We present Flexible Attenuation Fields (FlexAF), a data-centric CT reconstruction framework specifically designed to accommodate datasets with heterogeneous capture properties. Built on differentiable ray tracing and continuous neural volumes, FlexAF accepts X-ray images captured from any position and orientation in the world coordinate frame, including images

²Pun intended.

which differ in size, resolution, field-of-view, and photometric settings. By recasting CT reconstruction in terms of independent X-ray cameras within a common world coordinate frame, we can begin to widen the scope of computed tomography beyond the idealized datasets we pursue today. Indeed, intentional heterogeneity may be the *key* to unlocking traditionally complicated or time-consuming tomographic tasks, such as multi-resolution and multi-energy analysis. We demonstrate that this added flexibility does not fundamentally hinder our ability to reconstruct high-quality volumes, with our method producing reconstructions for regular CT scans which are comparable to or exceeding those produced by filtered backprojection. Further, we build upon this capability to experiment with creative new solutions to traditionally challenging reconstruction tasks, including automatic extrinsic calibration and reconstruction of multi-resolution or multi-energy volumes.

We begin in Chapter 2 with a review of the foundational literature on which our work is built, discussing the flexible camera models of photogrammetry and neural radiance fields, existing methods for CT reconstruction, and related methods for neural CT reconstruction. In Chapter 3, we discuss the principles and features of the FlexAF framework with an emphasis on those features which bring added flexibility to CT reconstruction. This is followed in Chapter 4 by a description of the core datasets used for this study, including a composite dataset designed specifically for testing reconstruction of heterogeneous inputs. Chapter 5 presents our results from applying FlexAF to the tasks of standard CT reconstruction, automatic geometric calibration, multi-resolution reconstruction, and multi-energy reconstruction. We conclude in Chapter 6 with a discussion of the challenges and limitations of our approach and a vision for tomography in the age of machine learning and neural networks.

CHAPTER 2. BACKGROUND

“I have been for a long time interested in the problem of the cathode rays from a vacuum tube as studied by Hertz and Lenard. I had followed theirs and other researches with great interest, and determined, as soon as I had the time, to make some researches of my own.”

– *Dr. Wilhelm Röntgen, The New Marvel in Photography, McClure’s Magazine, 1896*

A primary design goal for the FlexAF framework is to enable tomographic reconstruction from X-ray projection images with heterogeneous capture properties. Our inspiration derives from the observation that support for heterogeneous datasets expands the usefulness of photographic 3D scene reconstruction methods rather than hindering their application. Methods such as photogrammetry and neural radiance fields (NeRFs) are frequently applied across a wide range of multiscale and multispectral reconstruction tasks which might be difficult or impossible to approach with only homogeneous input datasets. In both cases, it is worth discussing how this heterogeneous support is enabled, what flexibility it provides, and how these methods compare to CT reconstruction algorithms.

2.1 Photogrammetry

Photogrammetry is a computational process for accurately reconstructing the structure and appearance of a 3D scene from photographs taken at different viewpoints. In photogrammetry, image formation is modeled by a general projective camera [29] represented by a homogeneous 3×4 matrix P which maps world coordinates \mathbf{X} to image coordinates \mathbf{x} :

$$\mathbf{x} = P\mathbf{X} \tag{2.1}$$

The functional meaning of matrix P can be better understood by its decomposition

into the camera's *extrinsic* parameters, $[R|\mathbf{t}]$, and *intrinsic* parameters, K :

$$\mathbf{x} = K[R|\mathbf{t}]\mathbf{X} \quad (2.2)$$

R is a 3×3 rotational matrix and \mathbf{t} is a translational 3-vector which together define a world-to-camera perspective transform. K is a 3×3 upper triangular matrix and has the form:

$$K = \begin{bmatrix} \alpha_u & s & c_u \\ & \alpha_v & c_v \\ & & 1 \end{bmatrix} \quad (2.3)$$

where c is the optical center, or *principal point*, of the camera in detector coordinates, α is the per-axis focal length in terms of pixel sizes, and s is a skew factor which is normally 0. The intuitive explanation for this decomposition is that the extrinsic parameters map world coordinates onto the *normalized* image plane, and the intrinsic parameters scale, shift, and (rarely) skew the points on that plane according to the construction of the specific camera (i.e. the internal optical properties of the lens or detector). Fully expanded, a homogeneous 3D world coordinate is projected to a homogeneous 2D image coordinate with the equation:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \alpha_u & s & c_u \\ & \alpha_v & c_v \\ & & 1 \end{bmatrix} \begin{bmatrix} R_{3 \times 3} & \mathbf{t}_x \\ & \mathbf{t}_y \\ & \mathbf{t}_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \quad (2.4)$$

Though this projective mapping reduces the dimensionality of the scene for each individual image, the underlying principle of photogrammetry is that the discarded dimension can be recovered from image features which appear in multiple views. Because the image content is formed by a strong structural relationship between camera

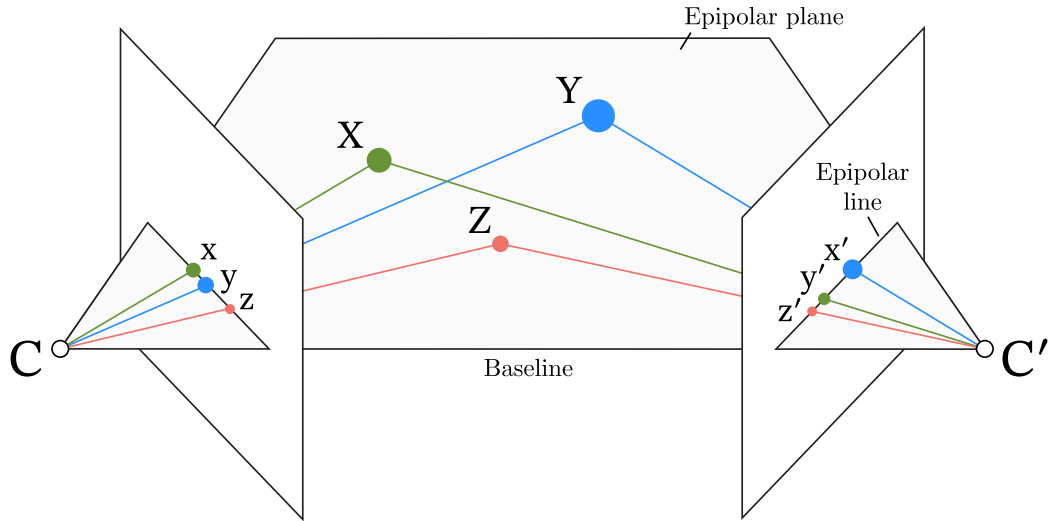


Figure 2.1: **Epipolar geometry and triangulation in photogrammetry.** In epipolar geometry, the baseline connecting the camera centers $\overline{CC'}$ forms a set of *epipolar planes* which project to *epipolar lines* in each image. Given 8 or more corresponding features, we can compute this geometry directly and recover the relative extrinsics of the cameras. Once the cameras are calibrated, we triangulate 3D feature coordinates by backprojecting rays from the camera centers through the projected features in the images. The epipolar constraints guarantee that these rays must intersect at the features' original 3D coordinates.

and scene, features which correspond across images inversely provide information about that structuring geometry. Specifically, when the camera intrinsics are known, corresponding features define an epipolar constraint by which the relative positions of the cameras can be determined up to a similarity transform [30]. With the extrinsics known, the further triangulation of world coordinates for scene features is relatively trivial. Rays are backprojected from the features in the image planes into the world coordinate frame, and the intersections of these rays determine the recovered 3D positions (Figure 2.1).

In practice, this process is not straightforward for real-world datasets which are subject to measurement error in the form of noise and lens distortion. Rather than directly factorizing solutions to the camera geometry using idealized projective relationships, structure-from-motion (SfM) algorithms address these issues by estimating

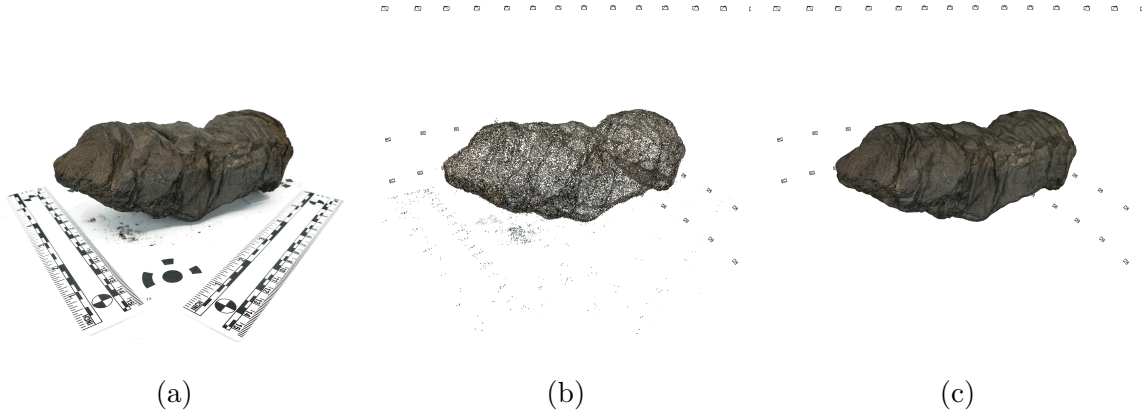


Figure 2.2: **The stages of photogrammetric reconstruction.** Demonstration of photogrammetry on a rolled Herculaneum scroll. (a) Photographs of the scroll are taken from arbitrary positions and orientations and with potentially varying photometric settings. Scale indicators are added to the field of view for the recovery of absolute scale. (b) MVG recovers the scene structure: the extrinsics of the cameras and the triangulated features of the scroll. (c) MVS transforms the feature points into a high-quality, textured model which reproduces the scroll’s natural appearance.

a set of optimized camera parameters which best explain the image set. The general SfM approach is to detect and match corresponding features across views using an image feature detector like SIFT [53], estimate the camera parameters using the pair-wise projective relationships, and then optimize these parameters by minimizing the *reprojection error* of the observed features across images [62]. This last step, known as *bundle adjustment*, is a nonlinear least-squares optimization problem, and is usually solved using the Levenberg-Marquardt algorithm [49, 55]. Notably, bundle adjustment optimizes both the camera extrinsics and intrinsics, thus only a close approximation of the intrinsics is required to initialize SfM.

The process described thus far has largely been concerned with the determination of scene structure (i.e. the positions of cameras and features in the world coordinate frame) from multiple views, otherwise known as multiple view geometry (MVG). However, MVG only forms one part of the photogrammetry equation (Figure 2.2), and the construction of a textured scene model is typically performed using methods from

the field of multi-view stereo (MVS). Broadly, MVS methods exploit the photometric consistency between multiple views in order to build a feature-rich model of the scene’s appearance properties, such as the lighting, surface geometry, and surface materials. This process involves the calculation of accurate depth maps for each image, from which a dense point cloud for the scene is extracted, meshed, and refined to construct an accurate surface mesh. Finally, material properties such as color and texture are projected onto the surface from the calibrated image set to generate the completed scene model. An in-depth description of the various MVS techniques can be found in [22].

2.2 Radiance fields

The past few years have seen the rapid development of *radiance field* methods for learned scene reconstruction using differentiable ray tracing and rendering. These methods derive their name from the nature of the scene representation itself, a continuous radiance function, or field, which can be evaluated at arbitrary points in the world coordinate frame. Recent work in this area has begun to move away from a fully continuous scene model for reasons of efficiency, but we will continue to use the term “radiance fields” as a convenience to describe the broad category of differentially-learned scene models.

Conceptually, radiance fields can be seen as a replacement for much of the MVS portion of photogrammetry. Usually, the images and cameras used for training radiance fields have already been calibrated using SfM, thus the radiance field is tasked with modeling the scene’s appearance properties given a well-defined scene structure. Rather than constructing a textured surface mesh, radiance fields produce an implicit scene model which is often stored in the weights of a neural network.

2.2.1 Neural radiance fields

The neural radiance field (NeRF) method presented by Mildenhall et al. was the first radiance field method to gain widespread attention [57]. Designed for 3D view

synthesis of photographic scenes, NeRF set a new bar for modeling complex, view-dependent scene characteristics such as occlusion, reflections, and specular highlights. Additionally, the NeRF model required far fewer training images than contemporary view synthesis methods and was significantly smaller than other synthesizing networks.

One of the most interesting aspects of the NeRF method is the way in which it intuitively combines principles from machine learning and volume rendering (Figure 2.3). The scene is modeled as a continuous vector-valued volume represented by a multi-layer perceptron (MLP). The input vectors to the volume are a 3D world coordinate and view direction, and the outputs are a view-dependent RGB color vector and density scalar. During training, a differentiable volume renderer synthesizes images of the neural volume from the viewpoint of each of the training images by casting rays through the volume and requesting color samples from the MLP. These samples are integrated along each ray into a single color value, using the learned density as an alpha compositing value to control the weighted contribution of each color sample. This process is fully differentiable, thus the MLP is updated through backpropagation using the residual error between the expected value and the rendered color. Networks which learn a function parameterized by its coordinates are often referred to as *coordinate-based networks* or *implicit neural representations*.

A significant problem in volumetric rendering is the question of how to avoid the computational expense that comes from sampling largely empty regions of the volume. This problem of sampling *attention* is particularly challenging for radiance fields, where the volume changes during training, and the sampling strategy may need to be adjusted across training iterations. NeRF addresses this issue by introducing a hierarchical sampling approach where two volumetric models are learned simultaneously, one “coarse” and one “fine.” The coarse network is provided ray samples using a method called *stratified sampling*, where the ray is divided into equal-sized bins,

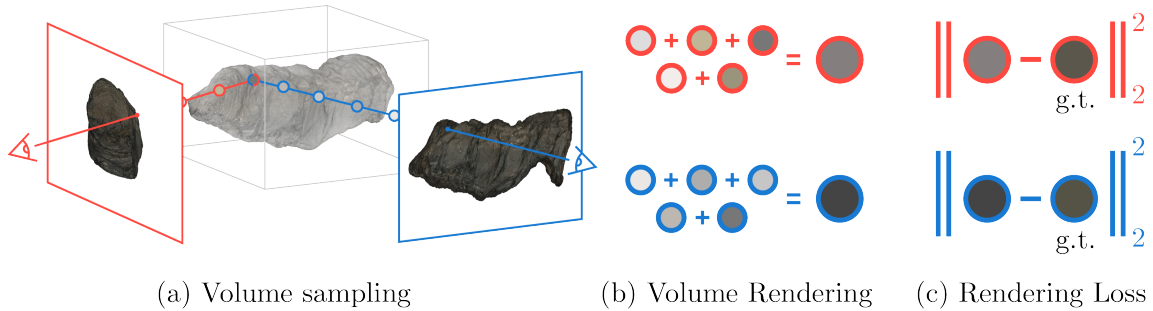


Figure 2.3: **The neural radiance field (NeRF) training and evaluation process.** (a) For each pixel in the set of training images, point samples are taken along rays drawn from the image plane through the volume. These point samples, along with the view direction, are passed to the MLP, which returns a volumetric density and view-dependent color value. Together these represent the color and opacity for the given coordinate. (b) The color and density samples along each ray are integrated to form a final, rendered color value for each input pixel. (c) The output color is compared to that of the input, and the MLP is updated based on the resulting loss.

and a random sample is drawn uniformly from each bin. The output densities from the coarse network are normalized to construct a piecewise-constant PDF along the ray. This PDF is used to sample a new set of points that are assumably closer to the content-bearing regions of the scene. These new points and the original coarse points are evaluated by the fine network and used to determine the final ray color used for rendering. To ensure that the coarse and fine networks do not diverge, the outputs for both networks are used to determine the loss value for the given ray.

A notable property of NeRF is its ability to quickly learn high-frequency features in the scene. While it is well-established that MLPs with nonlinear activation functions are universal function approximators [13, 34], recent work has shown that such networks exhibit a strong spectral bias which manifests during training as frequency-specific learning rates [77]. To provide faster convergence rates for the high-frequency details that occur in natural scenes, NeRF applies a form of *positional encoding* to the input vectors prior to passing them to the MLP which maps spatial vectors into

a high-dimensional, frequency-like space:

$$\gamma(\mathbf{v}) = [\sin(2^0\pi\mathbf{v}), \cos(2^0\pi\mathbf{v}), \dots, \sin(2^{L-1}\pi\mathbf{v}), \cos(2^{L-1}\pi\mathbf{v})] \quad (2.5)$$

In this equation, L is a hyperparameter which controls the dimensionality and bandwidth of the encoding and is set to $L = 10$ when encoding the sample coordinates and $L = 4$ when encoding the view direction.

2.2.2 Random Fourier features

Seeking to better understand the high-frequency learning properties of NeRF, Tan-cik et al. evaluated MLPs for low-dimensional coordinate regression tasks with analysis methods for kernel regression using a neural tangent kernel (NTK) [92]. They show that under NTK theory, the standard MLP has a sharp kernel falloff for parts of the learned function that correspond to high-frequency features. By first passing input coordinates through a Fourier feature mapping [46, 78] of the form

$$\gamma(\mathbf{v}) = [a_1 \cos(2\pi\mathbf{b}_1^T\mathbf{v}), a_1 \sin(2\pi\mathbf{b}_1^T\mathbf{v}), \dots, a_m \cos(2\pi\mathbf{b}_m^T\mathbf{v}), a_m \sin(2\pi\mathbf{b}_m^T\mathbf{v})]^T \quad (2.6)$$

this kernel falloff can be flattened, and learning high-frequency features becomes tractable for an MLP.

The authors observe that NeRF’s positional encoder is a special case of Fourier feature mapping, though one with a bias towards axis-aligned features. They propose an alternative *Gaussian encoding* which their experiments show performs better than positional encoding across a wide range of regression tasks:

$$\gamma(\mathbf{v}) = [\cos(2\pi\mathbf{B}\mathbf{v}), \sin(2\pi\mathbf{B}\mathbf{v})]^T \quad (2.7)$$

where each entry in $\mathbf{B} \in \mathbb{R}^{m \times d}$ is drawn from the normal distribution $\mathcal{N}(0, \sigma^2)$, d is the number of input dimensions, and m and σ are task-specific hyperparameters.

Intuitively, this mapping describes a sparse set of random Fourier features, with the feature sparsity controlled by m and the maximum bandwidth controlled by σ . For convenience, we refer to σ as the *scale* of the Gaussian encoder and m as the *number of features*. Unlike positional encoding, where increasing the bandwidth increases the size (and memory footprint) of the encoded feature, the authors show that keeping a fixed number of features and adjusting the scale is sufficient to control learning performance across a range of implicit modeling tasks.

Zheng et al. presented an alternative performance analysis for the broad category of positional encodings, including the Gaussian encoding [108]. Rather than looking at positional encodings with respect to their Fourier properties, the authors instead propose to view positional encodings as being systematically sampled from shifted continuous basis functions, a superset of encodings which includes Fourier feature mappings. Under this framework, they find that the dominant factors governing the performance of a specific positional encoding is the approximate matrix rank of the embedded coordinates and the distance preservation between coordinates after embedding. These two properties form a trade-off; a higher matrix rank correlates with better memorization of the training data, whereas distance preservation correlates with better generalization to unseen coordinates. They further show that increasing the scale of the Gaussian encoding decreases the distance preservation of the encoding and, given a sufficient number of features, linearly increases the embedded matrix rank up to a saturation point.

For our purposes, we take the following insights from these two works. First, increasing the scale of the Gaussian encoding will increasingly allow an MLP to learn high-frequency content. For maximum performance, the number of features should be increased commensurately to ensure a sufficient sampling density in feature space. Second, this consequently has the effect of decreasing the model’s ability to generalize between coordinates. As we will see, these insights will have important ramifications

for the FlexAF framework.

2.2.3 Approximating ray integrals

In all radiance field methods, the goal of the differentiable ray tracer is to convert the knowledge encoded in the model into projection images which accurately reconstruct the scene being observed. Ray tracing in general is formulated in terms of light transport theory, where the light rays that hit the image sensor are an integrated measure of their original brightness and color as well as the optical properties of the ray paths [75]. A particular challenge for radiance field ray tracers is that they must measure the ray integrals inside a continuous scene with an unknown set of objects. While traditional ray tracing methods operate in a continuous coordinate frame, the scene is largely composed of discrete scene objects (e.g. parameterized surfaces, triangulated meshes, discrete volumes), thus ray integration is at least structurally constrained to deploying efficient ray-object intersection algorithms.

As we have seen, the original NeRF method approximates the ray integral by taking color and density samples at discrete coordinates along each ray and using alpha compositing to integrate those values into a final color value. Hierarchical sampling, combined with the stochastic stratified sampling of the coarse network, is enough to ensure that samples are drawn from those portions of the scene which most contribute to the ray’s integral. An issue with this approach is that the image is approximated by infinitely small rays of sparse samples passing through the center of each pixel, while a natural image is instead formed from a continuous *pencil* of light striking the full surface area of each pixel. This leads to rendered images which are in some places blurry (due to the sample sparsity during training) or aliased (due to the infinitely small light pencil).

To address these issues, Mip-NeRF [2] approximates the conical frustum of each pixel with a set of multivariate Gaussians. The ray is first divided into n subintervals with $n + 1$ endpoints $[t_0, \dots, t_{n+1}]$, with each interval representing a conical section

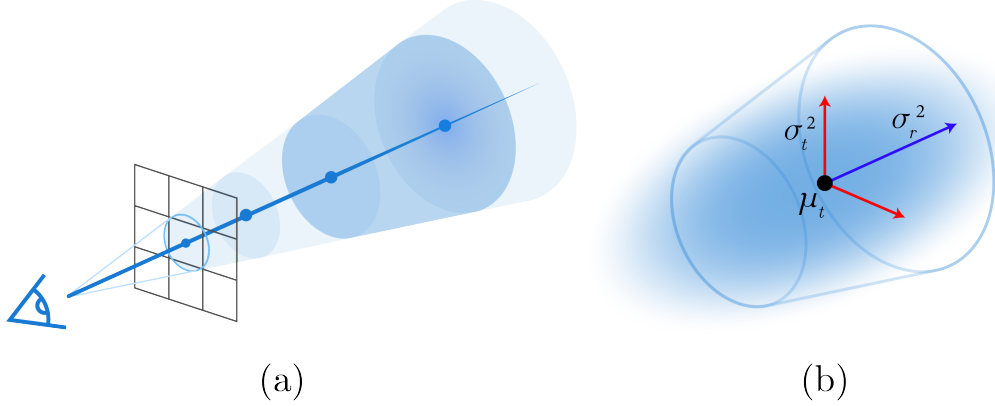


Figure 2.4: **The Mip-NeRF ray sampling method.** (a) Rather than drawing point samples from along the ray path, Mip-NeRF approximates conical sections of the pixel’s frustum. (b) The conical section for ray position t is modeled a 3D Gaussian distribution with mean μ_t , variance along the length of the ray σ_r^2 , and variance perpendicular to the ray σ_t^2 .

of the full frustum (Figure 2.4). These sections are then approximated by multivariate Gaussians with mean μ and covariance Σ , which replace the point samples from NeRF. Since the conical sections are circular and symmetric around the ray direction, these values can be computed from the mean distance along the ray, the variance along the length of the ray, and the variance perpendicular to the ray. To address positional encoding for multivariate Gaussians, the authors introduce *integrated positional encoding* as the expected sine and cosine for the Gaussian given as:

$$\mu_\gamma = \mathbf{P}\mu, \quad \Sigma_\gamma = \mathbf{P}\Sigma\mathbf{P}^T \quad (2.8)$$

$$\gamma(\mu, \Sigma) = \begin{bmatrix} \sin(\mu)_\gamma \circ \exp(-(1/2) \text{diag}(\Sigma_\gamma)), \\ \cos(\mu)_\gamma \circ \exp(-(1/2) \text{diag}(\Sigma_\gamma)) \end{bmatrix} \quad (2.9)$$

where \mathbf{P} is the 2^n basis function from NeRF rewritten as a 3D Fourier features matrix and \circ denotes element-wise multiplication. The encoded Gaussians, along with the encoded view direction, are passed to the MLP, and the resulting densities and color are integrated into final ray samples using the method described in NeRF.

Hierarchical sampling is still employed to encourage sampling near scene content, but now only a single MLP is used since the Gaussians implicitly sample content across multiple scales. Mip-NeRF shows slightly faster runtimes and moderate-to-significant improvements to error rates for the datasets tested.

2.2.4 Learned flexibility

We make note of two radiance field methods which were influential in our thinking around flexible CT reconstruction: Bundle Adjusting Radiance Fields and NeRF in the Dark.

The Bundle Adjusting Radiance Fields method, disconcertingly acronymized as BARF, extends NeRF with automatic camera extrinsic calibration even in the face of significant miscalibration [50]. Alongside scene reconstruction, the BARF method jointly learns a 6 degree-of-freedom transformation \mathbf{p} of the camera poses. The authors note that positional encoding interacts poorly with the task of smoothly learning a camera pose, as the gradients of the high-frequency components in early training produce erratic weight updates for the pose parameters. Their simple solution is to weight the components of the encoded coordinates in order to control the contribution of the various frequency bands during training:

$$\gamma_k(\mathbf{v}; \alpha) = w_k(\alpha) * [\sin(2^k \pi \mathbf{v}), \cos(2^k \pi \mathbf{v})] \quad (2.10)$$

where $\alpha \in [0, L]$ is set proportionally to the training progress and w_k smoothly interpolates between $[0, 1]$ as α increases:

$$w_k(\alpha) = \begin{cases} 0 & \text{if } \alpha < k \\ \frac{1 - \cos((\alpha - k)\pi)}{2} & \text{if } 0 \leq \alpha - k < 1 \\ 1 & \text{if } \alpha - k \geq 1 \end{cases} \quad (2.11)$$

In effect, BARF increasingly enables high-frequency components of the encoding as

training progresses, which allows the poses to learn from a smooth signal in early training and a detailed signal in late training.

NeRF in the Dark tackles the problem of learning scenes from high dynamic range (HDR) images [58]. Digital images can store an extremely large range of intensity values,¹ however display technologies have historically not been capable of reproducing such ranges of intensities in a way that matches the capabilities of human visual perception. As a result, images are often *tone mapped* into an 8-bit value range that approximates the perceptual contrast experienced by a human eye observing the same scene. Incorporating HDR information into a radiance field is one step towards enabling realistic, dynamic relighting of the encoded scene.

To enable HDR support, NeRF in the Dark applies exposure correction to the outputs of the differentiable ray tracer by multiplying the integrated color values, \hat{y} , by the camera shutter speed, t , and a per-color-channel corrective scalar, α_t^c . The corrective scalars are unique for each shutter speed and are learned jointly alongside the network. The final output for each rendered ray is given as:

$$\hat{y}_i = \min(\hat{y}_i^c \cdot t_i \cdot \alpha_{t_i}^c, 1) \quad (2.12)$$

where c is the color channel and the inner \hat{y}_i is the integrated output from the MLP. NeRF in the Dark also uses a relative mean-squared error (MSE) loss, which is a linear approximation to the L2 loss with a tonemap ψ applied to both the input and predicted images:

$$\tilde{L}_\psi(\hat{y}, y) = \sum_i \left(\frac{\hat{y}_i - y_i}{\text{sg}(\hat{y}_i) + \epsilon} \right)^2 \quad (2.13)$$

where $\text{sg}(\cdot)$ is a stop-gradient function which treats its argument as a constant during backpropagation.

¹While digital image sensors can capture 10 or 12-bits of dynamic range, it is not uncommon to encounter images with 16-bits of dynamic range. Such images are usually derived computationally or by combining photographs taken with multiple exposure parameters in a process called “bracketing”.

2.2.5 Real-time rendering

A significant limitation to neural radiance field methods is the computational expense of sampling and updating the neural network. NeRF and Mip-NeRF alike require many hours to train and many seconds to render a single image frame, making real-time interaction with the scene impossible. A significant amount of work has gone towards optimizing neural radiance methods for interactive rendering. These efforts often draw their inspiration from existing methods in computer graphics, employing variations on such techniques as Z-buffering [17, 51], scene baking and caching [24, 32], spatial partitioning [79, 93, 102], variable rate shading [80, 81], or some combination of the above [59]. Particularly interesting are the methods which avoid neural networks entirely [45, 99, 103].

Though our work is not directly influenced by these methods, we highlight them to demonstrate the vibrant, multi-faceted body of research which is developing around radiance fields. At the time of this writing, the original NeRF paper has garnered over 5,000 citations on Google Scholar since it was first presented in 2020 [27]. Research into radiance fields is advancing at an extraordinary pace and along multiple lines of inquiry, and the methods and techniques we have discussed will only grow more accurate, efficient, and interactive with time.

2.3 X-ray imaging and CT reconstruction

X-rays interact with matter in three primary ways: the photoelectric effect, Compton scattering, and Rayleigh scattering [36]. Of these, the photoelectric effect is the most important for X-ray imaging. If the X-ray photon’s energy is higher than that of a shell electron’s binding energy, the photon is absorbed by the atom, and the electron is ejected from the shell as a free electron, where the probability of this occurrence is proportional to the atom’s atomic number:

$$P_{photoelectric} \propto Z^3 \tag{2.14}$$

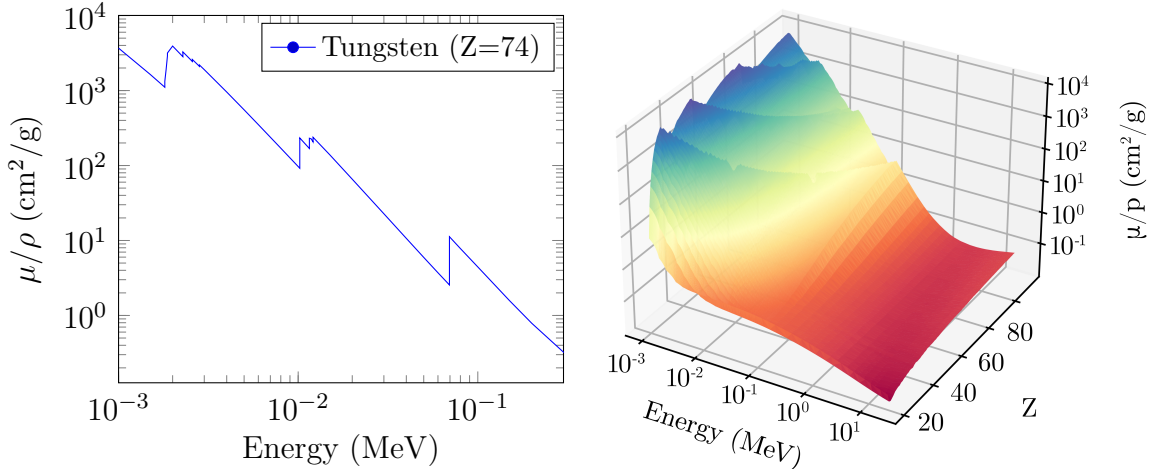


Figure 2.5: **X-ray mass attenuation coefficient as a function of incident energy.** (a) A 2D plot of the X-ray mass attenuation coefficients for Tungsten ($Z=74$). This plot demonstrates the complexity of the attenuation spectrum with respect to incident X-ray energy. As the incident energy increases, attenuation decreases as a mostly continuous function which is interrupted by discontinuous ridges at each element’s absorption edges. (b) A 3D plot for $Z \in [1, 92]$. When considered across a range of elements, the attenuation function forms a coherent topology with respect to energy and atomic weight. Data from NIST SRD 126 [40].

The net result of these interactions is that the X-ray beam intensity is attenuated as it passes through matter. The measure of an element’s likelihood to attenuate X-rays is known as the *attenuation coefficient* of the element, μ , and is measured as a function of the incident X-ray energy (Figure 2.5).

Following the Beer-Lambert law [4, 5, 36, 48], the intensity of a monochromatic X-ray beam that has passed through a uniformly attenuating material is determined by an exponential relationship between the original beam intensity and the attenuation coefficient of the traversed material:

$$I = I_0 e^{-\mu L} \quad (2.15)$$

where L is the thickness of the material. For composite materials, this formula can be rewritten in terms of the definite integral of attenuation coefficients along the length

of the beam's path:

$$I = I_0 e^{-\int_0^L \mu(\mathbf{x}) d\mathbf{x}} \quad (2.16)$$

where $\mathbf{x} \in \mathbb{R}^n$. To get a more convenient formula for CT reconstruction, we can divide both sides by I_0 to express attenuation in terms of the ratio of inputs and outputs and apply the negative logarithm to get:

$$p = -\ln\left(\frac{I}{I_0}\right) = \int_L \mu(\mathbf{x}) d\mathbf{x} \quad (2.17)$$

This final step is sometimes referred to as the *linearization* of the projection image, and in this form p represents the normalized (i.e. flatfielded), linearized X-ray projection image.

The mathematical foundations for computed tomography find their origin in the Radon integral transform [76]. Given a continuous function defined on a plane $f(\mathbf{x}) = f(x, y)$, the Radon transformed function Rf maps the original function to the space of line integrals of the plane:

$$Rf(L) = \int_L f(\mathbf{x}) d\mathbf{x} \quad (2.18)$$

We can see that (2.18) is equivalent to (2.17) for $\mathbf{x} \in \mathbb{R}^2$, and that the 1D projection measurement $p \in \mathbb{R}^1$ is a Radon-transformed observation of the attenuation coefficients $\mu(\mathbf{x})$. Thus, the central challenge in tomographic reconstruction is to develop a method for inverting the Radon transform in order to recover the attenuation coefficients.

Broadly, solutions to the Radon inversion problem fall into two categories: *backprojection* and *forward projection*. These names reference the general flow of information during the reconstruction process. In backprojection methods, each pixel in the reconstruction is backprojected to the set of projection measurements which pass through

that pixel, and the reconstructed attenuation coefficient is computed analytically. In forward projection methods, a computational Radon model is used to generate simulated projection measurements from the reconstruction, and the error between the simulated and captured projections is used to iteratively optimize the reconstructed pixels.

2.3.1 Backprojection

Backprojection reconstruction methods have been the most popular reconstruction methods for much of the history of computed tomography because they can be computed efficiently. These methods are built upon the *central slice theorem*, also known as the projection-slice theorem, which closely links the Radon and Fourier transforms [6, 37]. This theorem can be summarized as follows. Consider the function $f(x, y)$ and a corresponding rotated coordinate system $f'_\theta(t, s)$ at angle θ . A 1D projection of f'_θ is parameterized by the function $p(t, \theta)$, where t is simply a position along the rotated basis t . The central slice theorem states that the 1D Fourier transform of $p(t, \theta)$, taken with respect to t , is equal to a slice in the 2D Fourier transform of f taken at the same angle:

$$P(\omega, \theta) = F(\omega \cos \theta, \omega \sin \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi\omega(x \cos \theta + y \sin \theta)} dx dy \quad (2.19)$$

With this knowledge, a direct approach for reconstruction is conceptually straightforward. First, calculate the 1D Fourier transform of each projection (2.20). Next, fill a 2D Fourier image with the results from each of these transformations. Take care to place the data into the frequency domain at the angle at which the projection was captured in order to satisfy the central slice theorem (2.21). Finally, calculate the inverse 2D Fourier transform of the constructed Fourier space to recover the attenuation

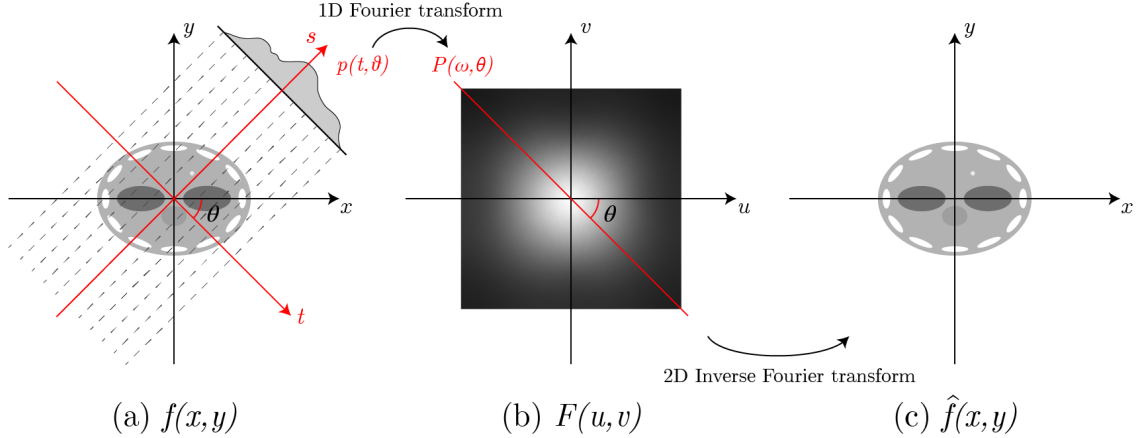


Figure 2.6: **The central slice theorem and its use for CT reconstruction.**

(a) The central slice theorem states that the 1D Fourier transform of projection $p(t, \theta)$ is equivalent to a line passing through the center of the 2D Fourier transform of $f(x, y)$ at angle θ . (b) By filling in the values of $F(u, v)$ with a sufficient number of Fourier-transformed projections and taking the inverse Fourier transform, (c) the function $\hat{f}(x, y)$ is recovered.

coefficients of the space (2.22). This process is visualized in Figure 2.6.

$$P(\omega, \theta) = \mathcal{F}[p(t, \theta)] \quad (2.20)$$

$$F(\omega \cos \theta, \omega \sin \theta) = P(\omega, \theta) \quad (2.21)$$

$$\hat{f}(x, y) = \mathcal{F}^{-1}[F] \quad (2.22)$$

Despite its simplicity, this approach suffers from practical issues. First, this method fills a Fourier image defined on a discrete grid using samples placed at radial coordinates. With a discrete set of projections, there will be empty regions of the Fourier grid which must be interpolated from observed samples, a non-trivial task to perform in Fourier space. Additionally, as the size of the reconstruction space grows, so too must the size of the Fourier space increase. At some point, calculating the inverse Fourier transform for increasingly larger images becomes computationally challenging.

Rather than constructing the Fourier image directly, the *filtered backprojection algorithm* (FBP) uses the central slice theorem to define the reconstruction space as a func-

tion of projection intensities [37]. Starting with the observation that the Fourier transform and the inverse Fourier transforms are reciprocal functions, $f(x, y) = \mathcal{F}^{-1}[\mathcal{F}[f]]$, and using the identity defined in (2.19), we can express the reconstruction space as the double integral:

$$f(x, y) = \int_0^\pi d\theta \int_{-\infty}^\infty P(\omega, \theta) |\omega| e^{j2\pi\omega(x \cos \theta + y \sin \theta)} d\omega \quad (2.23)$$

In this formulation, the inner integral represents the inverse Fourier transform of $P(\omega, \theta)$, which recovers a projection that has been filtered by the 1D bandpass filter $|\omega|$. An intuitive explanation for this equation is that the reconstructed value at $f(x, y)$ is the sum of all filtered projection samples which pass through the point (x, y) . By controlling the design of the bandpass filter $|\omega|$, one controls the frequency characteristics of the output reconstruction. Often, this means trading edge sharpness for reduced image noise. While the above formulation is only defined for parallel projections, there are many extensions to filtered backprojection which allow for fan-beam and cone-beam projectors [12, 19, 33, 107] and scan trajectories on a cylinder or sphere [43, 44, 60, 94].

FBP is easy to implement and can be computed efficiently on standard computing hardware using the fast Fourier transform (FFT), making it an extremely accessible method for a variety of CT applications. Further, since each reconstructed pixel value is simply the sum of contributions from independently computed filtered projections, the entire process can be easily parallelized and accelerated with GPUs.

It is important to note, however, that backprojection is an analytical approach to CT reconstruction which employs an idealized model of X-ray imaging. Equation (2.17) and all subsequent calculations rely upon the assumptions that the space is imaged with a monochromatic X-ray beam, that the X-rays striking a sensor pixel are locally collimated, and that the scan geometry has been recorded exactly. In

reality, these assumptions are frequently violated due to the simple practicalities of implementing X-ray imaging systems: the X-ray tubes used in most commercially available CT scanners produce polychromatic X-ray beams that break the linearity of (2.17) and introduce beam hardening and cupping artifacts; tube focal spots are large enough that a penumbra effect, which reduces image resolution, occurs at the sensor plane; and mechanical error in the scanning hardware or movement of the scan subject produce discrepancy between the actual and recorded scan geometry, introducing blurring or doubling artifacts in the reconstructed slices. Directly modeling these effects in the backprojection process is difficult, so solutions to these issues often involve conditioning of the scanner or projection images as an additional step prior to reconstruction [7, 9, 84].

2.3.2 Forward projection

Reconstruction from forward projection offers a flexible alternative to backprojection methods. Rather than analytically deriving the reconstructed pixel values, forward projection methods use a simulated X-ray projection system to optimize the reconstructed pixels with respect to the observations captured in the projection images. As the optimization process must often be applied iteratively, these methods are also known as iterative reconstruction methods. Historically, iterative reconstruction techniques were more expensive to compute than FBP and did not always produce reconstructions of higher quality. Today, computation times for iterative methods have significantly improved, and the reconstruction quality has begun to exceed that of FBP in many respects. As a result, iterative methods are becoming more widely available alongside commercially available CT scanners.

A simple forward projection algorithm models the projection process by the equation:

$$p = \mathbf{A}\mu + \epsilon \tag{2.24}$$

In this formulation, the projections, p , are a function of the projection system matrix

\mathbf{A} , the discrete vector of attenuation coefficients μ , and an additive noise vector ϵ . The system matrix models the transformation by which the attenuation coefficients are summed into the pixels of the projection images. Minimally, it applies the Radon transform to the coefficients, but it can also be constructed to account for other physical effects of X-ray interaction. Given this equation for projection, the reconstruction task can be defined as finding the values for μ which maximally match p after transformation by \mathbf{A} . Without prior information, all values of μ are equally likely, and this equation cannot be solved directly. Thus, iteration is employed to refine estimates of μ until convergence is achieved according to some loss function.

The prototypical example of a forward reconstruction method is the algebraic reconstruction technique (ART). Known in mathematics as the Kaczmarz method [41], but independently rediscovered and applied to image reconstruction by Gordon, Bender, and Herman in 1970 [28], ART was the first iterative method developed for image reconstruction and the method employed by Hounsfield to reconstruct data from the first commercially viable CT scanner [35].

During each iteration of ART, rays are cast from a simulated X-ray source position, through the pixels of the reconstruction space, towards each pixel on the X-ray detector. To simulate the effect of the Radon transform, the values of the pixels which intersect each ray are integrated into a single value which summarizes the ray's total attenuation. The i -th iteration for the estimated total attenuation along ray r is given as:

$$r^i = \sum_{j=0}^L \mu_j^i \tag{2.25}$$

where L is the number of pixels intersected by the ray and μ_j^i are the i -th density estimates for those pixels. The residual error between this simulated attenuation and that captured by the X-ray detector is then distributed evenly among the pixels which

intersect the ray. This translates to the per-pixel update function:

$$\mu_j^{i+1} = \max[\mu_j^i + \frac{p_r - r^i}{N}, 0] \quad (2.26)$$

While this algorithm is straightforward, it is worth noting that ART is a fairly simple model of X-ray projection. Each ray is considered independently and voxels are updated immediately, effectively ignoring the correlations between adjacent rays from the same projection image. Additionally, there is no compensation for system noise, which is unavoidable in real world datasets.

Statistical iterative reconstruction (SIR) methods were developed to address some of these issues through the incorporation of various types of prior information [20, 26, 85]. Usually, these methods form the reconstruction task in the Bayesian framework as maximizing the *a posteriori* probability estimate for the attenuation coefficients, $\hat{\mu}$:

$$\hat{\mu} = \underset{\mu}{\operatorname{argmax}}[\log \Pr(p|\mu) + \log \Pr(\mu)] \quad (2.27)$$

In this formulation, the log-likelihood term, $\log \Pr(p|\mu)$, defines the mapping of attenuation coefficients, μ , to the projections, p , while the prior term, $\log \Pr(\mu)$, models the properties of the scanned object and the reconstructed image [38]. These terms are converted into a mathematical reward function, and an optimization process iteratively computes a stable solution of the reconstructed image $\hat{\mu}$.

At their most basic, SIR methods only include a simple model of the X-ray projection system and a regularization term which describes how the reconstructed image should be formed. Functionally, the regularizer reduces noise in the output image by enforcing local consistency between adjacent voxels. Model-based iterative reconstruction (MBIR) methods are a more advanced version of this idea that further include a highly accurate system model and a statistical noise model alongside the regularization prior [52, 94, 104]. MBIR methods have been shown to significantly

improve reconstruction quality by modeling complex behaviors of the projection system, such as the blurring effects induced by the focal spot of cone-beam sources and the scintillators on flat-panel detectors [96, 97].

2.3.3 Attenuation field CT reconstruction

The application of neural networks, particularly deep neural networks, to CT reconstruction has been a well-studied topic for over a decade [1, 42, 95, 98, 100, 106]. We here restrict ourselves to a discussion of methods which employ coordinate-based networks to implement and/or inform the CT reconstruction task. We further note what differentiates these methods from FlexAF.

In the random Fourier features paper which introduced the Gaussian encoding [92], Tancik et al. compared various positional encodings across a range of 2D and 3D regression tasks. One of their experiments explores a 2D CT slice reconstruction task where an MLP is trained to predict the density values of a slice image when supervised on Radon-transformed projections of the slice. Designed to test the effectiveness of positional encodings for inverse learning problems, this experiment idealizes many challenges of real-world CT reconstruction and thus cannot be applied to practical CT reconstruction. For example, it only considers in-plane, parallel projections using a computational Radon transform of preexisting slice images. It is however notable as the first work, to our knowledge, which considers the application of coordinate-based networks for CT reconstruction.

Sun et al. applied a Fourier features network equipped with the Gaussian encoding to the task of synthesizing the missing rotational samples in sparse sinograms [91]. Unlike FlexAF, this work does not reconstruct a full volumetric model. Rather, the network is trained to reproduce sinograms parameterized by their coordinates (l, θ) , which correspond to each pixel’s sensor location and rotational angle respectively. Once trained, the network can then be queried at arbitrary sinogram coordinates in order to synthesize new sinogram samples. Synthesized sinogram samples are

combined with the original sparse sinograms to produce a more rotationally dense sinogram which is then reconstructed using an existing CT reconstruction algorithm.

Zang et al. propose a multi-stage reconstruction framework for ill-posed reconstruction problems, called IntraTom [105]. Like FlexAF, IntraTomo learns a volumetric attenuation field through the application of a differentiable projection system which is trained against X-ray projection images. The attenuation field and its synthesized projections are then used to iteratively optimize a final volume model as part of a model-based “geometry refinement” step. Unlike FlexAF, the learned attenuation field in IntraTomo is not the final output of the method and is instead used as an intermediate representation for projection synthesis.

Sitzmann et al. proposed SIREN, a coordinate-based network model based on an MLP with sinusoidal activation functions and no positional encoding. SIREN is notable for its improved ability to accurately learn both a function and its derivatives. Though the authors don’t apply SIREN to the radiance field task, they do demonstrate its ability to accurately solve waveform inversion tasks. In 2021, Koo et al. demonstrated a CT reconstruction method built on SIREN which employs a differentiable ray tracer similar to that of NeRF [47]. Rather than employing hierarchical sampling, they approximate the ray integrals in a single stage where the rays are divided into N intervals and samples are drawn from the midpoints of each interval. To correct for noise in their reconstructions, they augment the L2 loss function with a regularization term to control the spatial smoothness of the reconstruction. The method was tested on various phantoms and a real-world fan beam dataset, and the authors report comparable reconstruction quality to that of other model-based approaches. Early experiments with FlexAF evaluated the SIREN model for our neural volume, but we found that its stability was extremely sensitive to hyperparameter initialization, and we could not get it to converge for our micro-CT test datasets.

In parallel to the development of FlexAF, Rückert et al. introduced Neural Adaptive

Tomography (NeAT) [83], to date the most complete CT reconstruction framework employing radiance field methods. Noting the long computing times required by previous applications of implicit neural networks to CT reconstruction, they introduce an efficient hierarchical rendering pipeline, built on an adaptive octree decomposition of the volumetric space, that is capable of reconstructing full micro-CT volumes in a few hours. Further, they use the end-to-end differentiability of their pipeline to learn geometric and photometric system corrections which improve the signal-to-noise (SNR) of their reconstructions. The authors show that NeAT outperforms many of the leading existing reconstruction methods across a number of sparse and limited angle tasks. FlexAF shares many of the design goals of NeAT, in particular the interest in system calibration through automatic differentiation. However, our work is focused on how differentiable ray tracing and neural volumes allow us to expand the capabilities of CT reconstruction when combined with heterogeneous input datasets. Optimizing our work for at-scale deployment will be an important step for the wider application of FlexAF, and NeAT will be a valuable reference point during that development.

2.3.4 X-ray camera geometry

The structural relationships which map a point in 3D space onto the image plane of an X-ray sensor are very similar to those of the projective camera model used for photogrammetry, and in fact, the general projective model in (2.4) can be used to describe X-ray image acquisition for a cone beam X-ray source and flat panel detector.² Figure 2.7 illustrates the geometric equivalence between the general projective camera used in photogrammetry and the projection geometry of an X-ray sensor illuminated by a cone beam X-ray source.

An important difference between these two models is the location of the sensor with respect to the scene and the effect that placement has on the content of the projected

²While there are many other X-ray camera configurations besides those we discuss here, such as systems which use fan beam sources or curved detectors, most of these alternatives can be reduced to a cone or parallel beam geometry when considered at the pixel level.

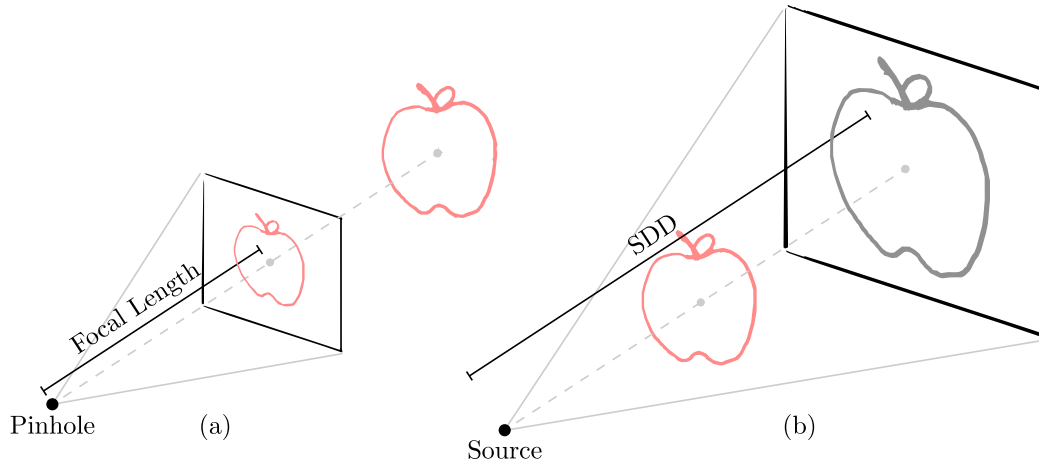


Figure 2.7: **A comparison of the photographic camera model and the geometry of cone beam X-ray imaging.** (a) The projective camera model for photography, showing the image plane in front of the pinhole for convenience. (b) The projective camera model for an X-ray sensor illuminated by a cone beam source. We can see that the focal length (i.e. pinhole-to-sensor distance) is analogous to the source-to-detector distance in determining the scale of features on the image plane.

image. In photography, the light from the scene is collected through the camera’s pinhole and projected such that scene features are “scaled down” to fit onto a smaller image sensor, with a magnification factor $f \leq 1$ determined by the camera’s intrinsic focal length. For cone beam X-ray projection, the light begins at a point light source and expands according to the inverse square law. At the image sensor plane, this results in the scene content having been “scaled up” by a magnification factor $f > 1$ which is determined by the source-to-detector distance (SDD). Thus, the SDD can be thought of as the X-ray system’s “focal length” and can be used to calculate the α parameter in (2.3). In CT, the magnification of the scene on the sensor plane is referred to as *geometric magnification* and is usually measured in terms of the ratio of the SDD to the source-to-sample distance (SSD).

Parallel beam X-ray sources provide a highly collimated beam of X-rays that are assumed to be perpendicular to all points on the image sensor plane (Fig. 2.8). Because of this perpendicularity, parallel beam geometries are more properly modeled through orthographic, rather than perspective, projection. In orthographic projec-

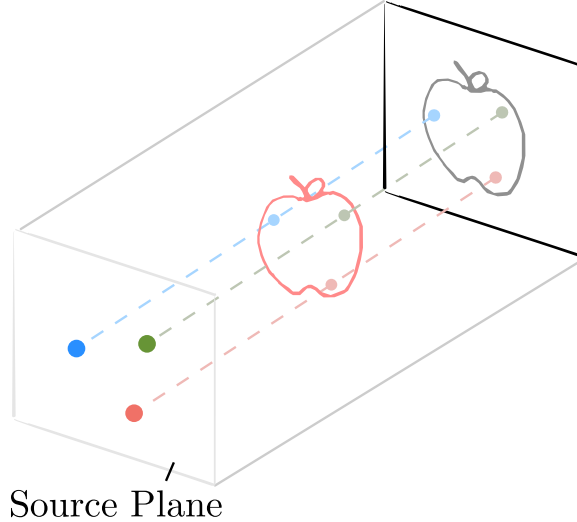


Figure 2.8: **The geometry of parallel beam X-ray imaging.** With a parallel beam, the image sensor is illuminated by a highly collimated beam of X-rays which are assumed to be perpendicular to the entire image plane, as if they proceeded from a distant *source plane*. As such, there is no observable magnification in the projections when altering the relative distances of the source or the detector planes. For purposes of reconstruction, the source plane placement is arbitrary as long as its distance from the center of rotation is sufficient to encompass the entire scan volume.

tion, a zero-scale is applied along the direction of the projection plane’s normal. For example, an orthographic projection of a 3D point onto an image plane at $Z = 0$ is given by the equation:

$$\begin{bmatrix} u \\ v \\ 0 \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \quad (2.28)$$

Orthographic projection can be generalized for arbitrarily positioned and oriented projection planes by right multiplying the orthographic projection matrix with a 4×4 homogeneous pose matrix composed of a rotation mapping the Z axis to the target plane’s normal and a translation to a point on the plane.

While these geometric camera models imply the ability to freely position X-ray cameras — both source and sensor — in the world coordinate frame, few CT recon-

struction algorithms use such a configuration in practice. For backprojection methods, this is due to those algorithms' relationship to the Fourier transform and the analytical derivation of volumetric attenuation as a function of projected intensities. By construction, FBP and its successors make a strong assumption about the structure of the X-ray cameras with respect to the scene, namely that the projections are captured along a radial path relative to known axis of rotation, or *isocenter*. This structure is so strongly determined that even small deviations in geometry, such as a pixel-length shift along the detector's horizontal axis, can produce a noticeable blur of the reconstructed volume.

For forward projection methods, the primary difficulty lies with calculating the reconstruction on a discrete grid. Though the simulated projection system can easily incorporate projective geometries, these methods must define an *explicit* update function to decide how the error between the simulated and real projections is to be distributed in the volume. Such a task is non-trivial in the case of freely positioned X-ray cameras, where significant variance in geometric magnification can produce irregular sampling patterns with respect to the 3D pixels, or *voxels*, in the grid. Additionally, the discrete grid is necessarily a bounded entity which becomes increasingly difficult to manage as the spatial extent of the volume grows and/or the scan resolution increases. Restricting camera geometries to views around a central volume makes sense simply as a matter of practicality, efficiency, and convenience.

For both forms of reconstruction, the transmissive nature of X-rays adds an ambiguity to the scene's projective geometry which is not present for photographic scenes. There is a strong rotational symmetry around the axis of rotation in X-ray imaging, and two X-ray projections captured with a rotational offset of 180° will appear to be horizontally flipped versions of each other.³ In the absence of any prior for the cam-

³This is not strictly true for cone beam X-ray sources, where objects that are nearer to the source will appear larger on the image sensor than objects which are further away. After a 180° rotation, these objects will have swapped positions relative to source, thus the projections will not be identical after a horizontal flip but only nearly so.

era’s structure, it is thus difficult, if not impossible, to disambiguate the orientations of two cameras based on their image contents alone. This makes camera calibration and reconstruction from a completely unorganized set of projection images a difficult proposition.

2.4 A look back with optimism

As we have seen, there is significant overlap in the projective camera models used by photogrammetry, radiance fields, and X-ray tomography, but tomography does not enjoy the same degrees of freedom as its photographic brethren. Before moving on, we make two observations of photogrammetry and radiance fields that we believe enables them to be more adaptable to heterogeneous inputs than tomography.

Our first observation is that photogrammetry introduces a separation of concerns between the scene *structure* and the scene *appearance*. Roughly, this delineation corresponds to a partition between MVG and MVS. The scene structure, computed with MVG, describes the global geometric relationships between the cameras and scene. The scene appearance, computed with MVS, describes the local geometric and photometric properties which govern our perception of the scene.

A similar distinction between structure and appearance is found in the way in which radiance fields implement the light transport model. In most radiance field methods, the scene is sampled to produce both density (structure) and view-dependent color (appearance). We see that here density also describes a global property of the scene, namely the visibility of objects from a particular view point, while view-dependent color describes a very local property of the surfaces.

We believe that this separation of concerns is a significant reason for the flexibility that these methods enjoy with respect to heterogeneous inputs. Field of view, depth of field, image resolution — these are structural properties which govern the geometry between camera and scene. Image exposure, surface color, and scene illumination are appearance properties which influence our *perception* of the scene but not its

fundamental structures.

We do not find this same separation of concerns in computed tomography. As we discussed in section 2.3.4, CT reconstruction algorithms have long been defined by a canonical scene structure of projections captured radially about an isocenter. While many modern CT pipelines compute camera calibration in order to improve the reconstructed volume, there is still an assumption that the initial structure is similar to the canonical form, and there is little attempt to define the attenuation coefficients separately from this structural assumption.

Our second observation is that both photogrammetry and radiance fields construct a continuous scene model.⁴ As a result, these methods can easily adapt to irregular spatial patterns found in the input dataset. For example, sparse images of a large scene can be combined with dense images of an object-of-interest to produce a point cloud with spatially varying point density. Such irregular sampling densities are not easily accomplished with CT because the discrete grid must have a fixed sample rate for the entire scene. Either dense regions will be oversampled, sparse regions will be undersampled, or some combination of both. The continuous volumetric representations used by radiance fields would appear to solve this issue. Regions of the scene can be learned with exactly the sample rate required to reconstruct that region’s content, making it an ideal approach for datasets with multiscale features.

⁴Though photogrammetry often produces a discretized output model (e.g. a cloud of points, a triangulated surface), the coordinate frame for these objects is defined continuously.

CHAPTER 3. FRAMEWORK

“I did not think; I investigated.”

– *Dr. Wilhelm Röntgen, The New Marvel in
Photography, McClure’s Magazine, 1896*

The FlexAF reconstruction framework falls into the broad category of forward projection algorithms. During training, our differentiable ray tracer queries our volumetric model (a neural network) to render simulated projection images of the current, learned reconstruction. These simulated images are then compared against a set of ground truth projection images. Since the entire process is end-to-end differentiable, the resulting loss is used to update the volumetric reconstruction through gradient-based optimization. Unlike most supervised machine learning tasks, where the goal is to train a model which generalizes to unseen inputs, our goal is to exactly learn the 3D function of volumetric attenuation coefficients $\mu(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^3$. As our volumetric model is continuous, we refer to this function as the volume’s *attenuation field*. After training, we view the reconstructed slices directly by querying the neural volume with a set of coplanar 3D coordinates.

In this chapter, we describe the key design choices, features, and process of the FlexAF framework in more detail. A visual overview of the FlexAF components is available in Figure 3.1. We begin with an explanation of the X-ray camera model we use during training and how this model maps onto our input datasets. Next, we describe our differentiable ray tracer and our process for X-ray image formation, highlighting its default operation and various optional features. We follow this with a discussion of our two neural volume representations: a standard model for reconstruction of a single attenuation coefficient and a multi-energy model which supports heterogeneous incident energies. Finally, we conclude with a description of various support mechanisms we employ to improve training times and control the learning attention.

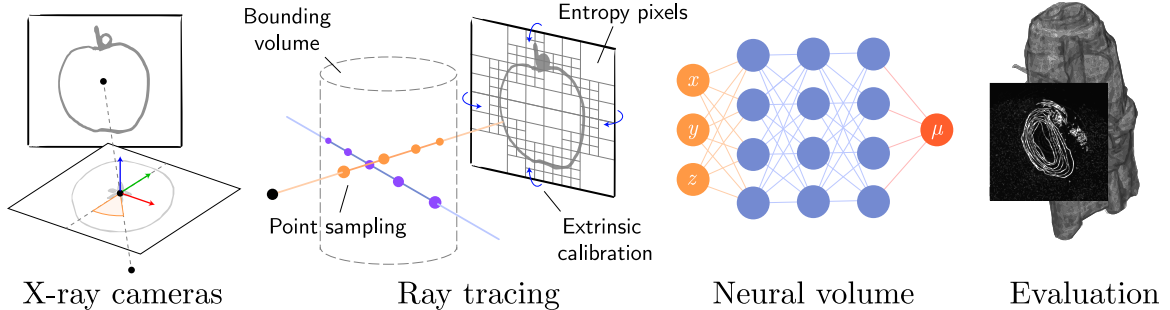


Figure 3.1: **The primary components of the FlexAF framework.** From left to right, these components generally map onto the reconstruction process. First, we initialize the *X-ray cameras* for each projection image using metadata provided by the scanner. Second, we *ray trace* the projection images and draw point samples from the world coordinate frame. Next, we pass the point samples to our *neural volume* and receive estimated attenuation coefficients. Finally, we *evaluate* our neural renderings against the captured projections to improve our reconstructions. Once learned, the neural volume can be queried directly to render slices and volumes.

3.1 X-ray camera model

In FlexAF, each projection image defines an X-ray camera which exists in the world coordinate frame and which captures the total attenuation for the space lying between the X-ray source and detector. Our geometric model for X-ray cameras build from the projection geometries described in 2.3.4. Each camera has a source position S , detector center position D , and detector basis vectors $(\vec{u}, \vec{v}, \vec{w})$, which correspond to the horizontal, vertical, and normal axes, respectively. Because we want to construct a volume which is measurable in real world units, we specify the positions of our X-ray cameras in millimeter units. Since we are only modeling the first-order effects of linear attenuation, the X-rays striking the detector surface point $D_{\mathbf{x}}$ can be modeled as a ray \vec{r} passing between the source and the surface point (Figure 3.2). For cone beam geometries, this ray is given as a directed line segment:

$$\vec{r} = SD_{\mathbf{x}} \quad (3.1)$$

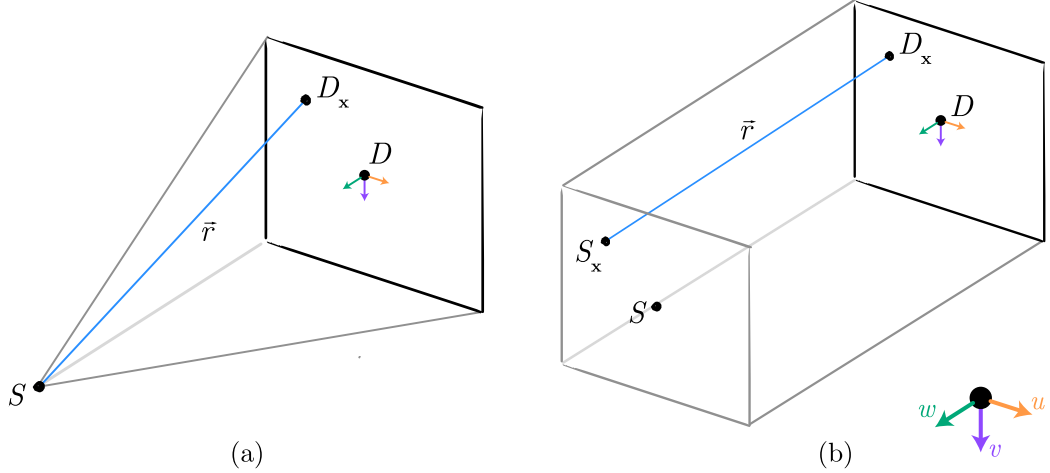


Figure 3.2: **Cone and parallel beam X-ray camera models in FlexAF.** (a) For a cone beam X-ray camera, all X-rays proceed from a single X-ray source point S . The projection ray which strikes point D_x on the detector is given by the ray $\vec{r} = SD_x$. (b) For a parallel beam X-ray camera, the ray which strikes D_x proceeds from a point S_x which lies on the source plane S along the detector normal w .

For parallel geometries, where X-rays are considered perpendicular to the detector surface, the ray proceeds from a point S_x which lies at a fixed distance along the detector normal. We set this value to be the distance between the source and detector centers:

$$S_x = D_x + \vec{w} * \|SD\| \quad (3.2)$$

$$\vec{r} = S_x D_x \quad (3.3)$$

For cases where the source position S is indeterminate or infinitely large (i.e. synchrotron light sources), the choice for this fixed distance is somewhat arbitrary and need only be large enough to cover the reconstruction space. A reasonable estimate is to use twice the sample-to-detector distance.

While our ultimate goal is to accommodate X-ray camera trajectories with arbitrarily positioned X-ray sources and detectors, the fact remains that few existing CT scanners cannot capture such trajectories. Since all the datasets in this study are de-

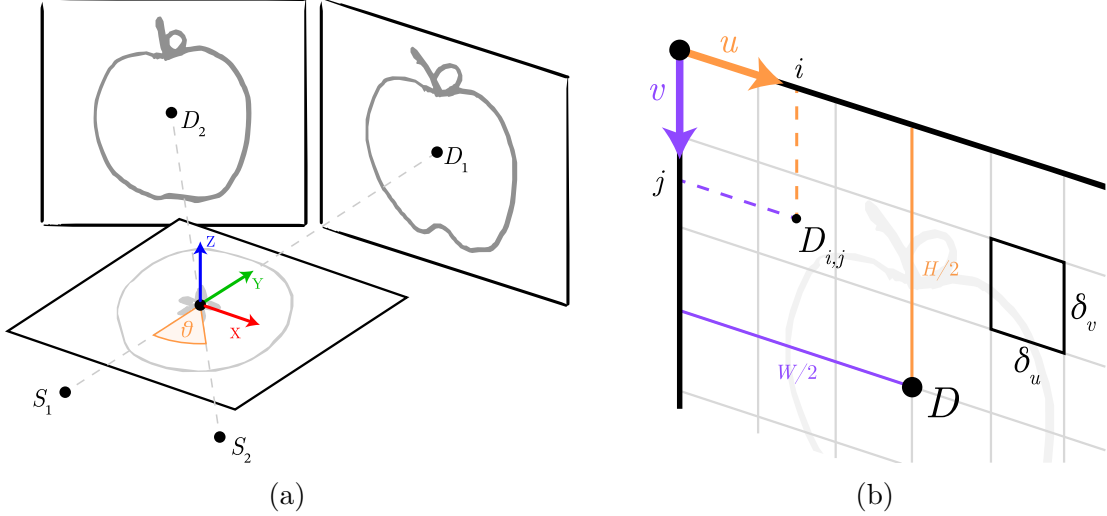


Figure 3.3: **Initialization of X-ray cameras from a cylindrical CT scan.** (a) Using metadata provided by the CT scan, we compute the X-ray camera positions for each projection image relative to the isocenter, which we place at the origin of the world coordinate frame. The scan’s rotational step size θ and the projection’s index determine its total rotational offset, while the source-to-sample and source-to-detector distances determine the source and detector placement respectively. (b) Once the detector is positioned in the world coordinate frame, we compute each pixel’s world coordinate $D_{i,j}$ relative to the detector center D as defined in (3.4).

rived from cylindrical CT scans with X-ray images captured at fixed rotational steps around an isocenter, we initialize the cameras for our projection images using this known trajectory (Figure 3.3a). We set the isocenter to be the origin of the world coordinate frame, and use the source-to-sample distance, source-to-detector distance, and total rotational offset θ for each projection image to calculate the initial positions and orientation vectors of S and D .

We follow the lead from radiance fields and train over individual pixels of the projections rather than full images or 2D subregions. This is a crucial property of the FlexAF framework as it enables us to easily support geometric heterogeneity in interesting ways. We can, as needed, train over full projection images, subregions of projection images, sparse samples from projection images, images captured from both parallel and cone beam geometries, images with different (physical and effective)

pixel sizes, etc.

Since we have defined the camera geometries in terms of individual rays striking the detector surface, we can easily calculate the world coordinates and orientation vectors for each pixel on the detectors (Figure 3.3b). Given the 2D pixel size in millimeters, δ , and the number of pixels along each axis, $H \times W$, the pixel’s center position can be calculated relative to the detector’s center position using the equation:

$$D_{i,j} = D + (i - \frac{H}{2})\delta_v\vec{v} + (j - \frac{W}{2})\delta_u\vec{u} \quad (3.4)$$

For flat panel detectors, the orientation vectors are identical to those of the detector.¹ We note that this construction provides inherent support for multi-resolution data, as the world coordinate offsets between adjacent pixels correspond to the physical pixel sizes and the effects of geometric magnification are encoded into the source and pixel positions in the world coordinate frame.

We construct a tensor T for each projection pixel p which is passed to our differentiable ray tracer during training:

$$T_p = \langle n, (d_x, d_y, d_z, 1), (s_x, s_y, s_z, 1), r_d, r_s, v \rangle \quad (3.5)$$

where d is the homogeneous world coordinate of the pixel center, s is the homogeneous world coordinate of the pixel’s X-ray source, r_d and r_s are the radii of the pixel’s frustum at d and s , and v is the pixel’s intensity value. Since we are training over randomized image pixels, we also track the global projection image index n so that we can apply image-level transformations during training. Our complete training set is a flattened list of pixel tensors from across all projection images.

¹Though we only consider flat panel detectors in this work, it is a minor extension to determine the position and orientation vectors for each pixel on a curved detector if the detector’s curvature is known *a priori*. The key takeaway here is that the detector’s position and orientation uniquely determines the position and orientation of each of its pixels.

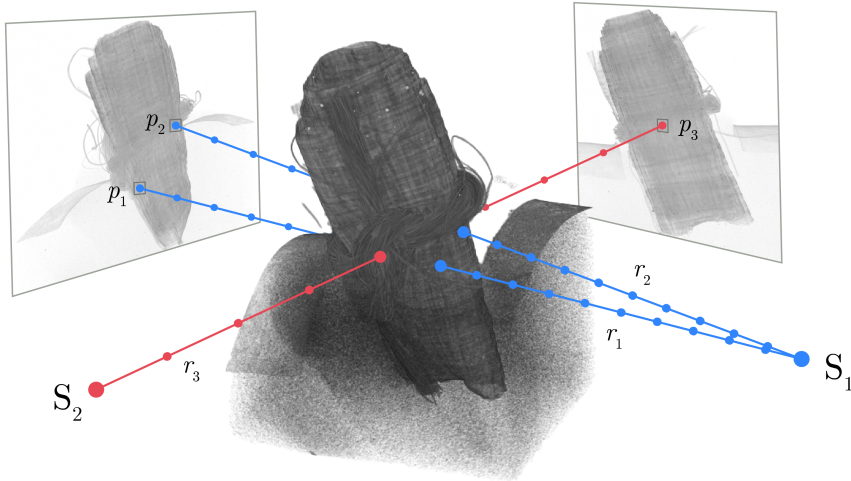


Figure 3.4: **A visualization of the FlexAF ray tracing system.** Every projection pixel p_x defines a ray r_x which travels from an X-ray source S_x to the detector. 3D points are drawn from the rays using stratified sampling and passed to the volume model to produce estimated attenuation coefficients. The coefficients are summed along each ray into single estimated projection value which is compared against the observed value at p_x .

3.2 Ray sampling

For each pixel p_x in our training set, there is a corresponding ray given by the directed line segment $\vec{r}_x = s_x d_x$. Our goal is to approximate the line integral of attenuation for this ray such that the final integrated value equals the observed pixel value in the projection image (Figure 3.4). We do this by drawing a discrete number of 3D point samples from along the ray which we then pass to our volume model for evaluation and integration. Similar to NeRF, we use a stratified sampling approach to generate point samples. The ray is divided into N equal-sized intervals, and we draw a new point sample uniformly at random from each interval. These samples are then combined with the ray end points to produce $N + 1$ sample points for evaluation. N is a hyperparameter which is selected according to the volume size and desired resolution, and is practically constrained by the computational limits of the host system. When N is small, the ray can be evaluated very quickly but will provide a poor approximation for the continuous integral of the Radon transform. As N

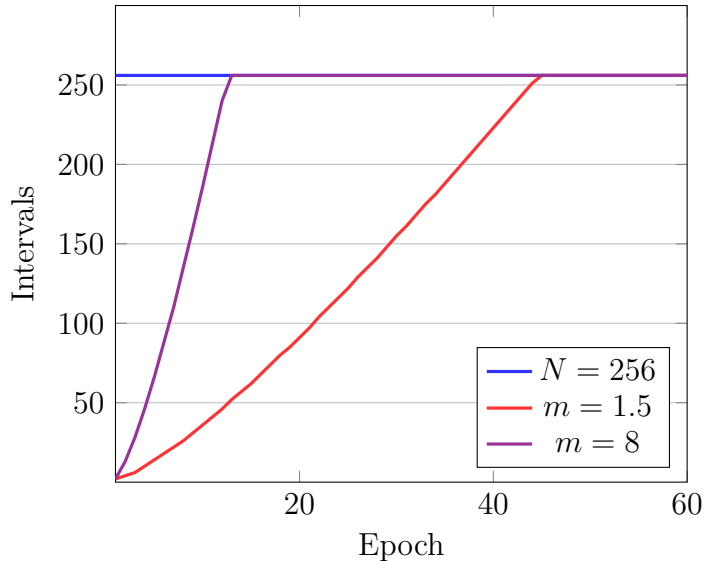


Figure 3.5: **Plot of various interval schedules across training epochs.** With a constant interval schedule (blue), the number of intervals do not vary across training epochs. In contrast, our log-linear schedule gradually increases the number of intervals in order to stabilize early learning and modestly improve total run times. When the schedule multiplier $m = 1.5$ (red), the number of intervals increase slowly. When the schedule multiplier $m = 8$ (purple), intervals increase quickly.

increases, the rays will more closely approximate the continuous integral, but at the expense of dramatically increased computation times.

During our development, we noted that immediately training with a dense number of sample intervals can occasionally result in poor learning. In the worst case, the gradients become unstable early in training and the model fails to converge. In the best case, the reconstruction converges rapidly in isolated regions of the volume to produce a locally sharp but globally sparse reconstruction which is then gradually “filled in” with more content as training progresses. Though this latter behavior is not necessarily problematic, it is sometimes preferable to initially learn a globally smooth reconstruction which becomes sharper as training progresses. This is particularly true when performing automatic extrinsic calibration, where supervising over high-frequency content early in training can lead to poor optimization of the extrinsic parameters.

To help stabilize early learning and provide more control over the learning behavior, we introduce an optional *interval schedule* which monotonically increases the number of sampling intervals from a lower bound to an upper bound according to a log-linear curve:

$$N(e) = \min(\lfloor L + m * (e + 1) * \ln(e + 1) \rfloor, U) \quad (3.6)$$

where e is the current training epoch, m is a hyperparameter which scales the rate of increase, and L and U are the lower and upper bounds on the number of sampling intervals respectively. In practice, we set $L = 2$ and select m and U according to the specific dataset. Figure 3.5 shows a plot of this schedule for common values of m and $U = 256$. We find no significant difference in reconstruction quality at convergence between using a constant number of intervals and the interval schedule. Though using a constant number of intervals does occasionally produce faster convergence than when using the interval schedule, the improvements are rarely dramatic, and we prefer the stability and predictability of using the interval schedule.

3.2.1 Pencil approximation

We have thus far modeled X-ray projection as individual rays following a linear path from the source to the center of detector pixels. While this is a useful geometric simplification, it misses the fact that the pixel has a physical surface area and is capturing a *pencil* of X-rays across its entire surface. For our continuous volumetric model, this can in principle lead to a scenario where we learn the attenuation for the precise coordinates along the ray and nowhere else, with the result being a volume model with noticeable “gaps” between the paths of rays.

This single ray problem is well-studied in computer graphics applications, where the result is a rendered image with noticeable aliasing artifacts. A typical solution is to employ a multi-sampling method where multiple rays are cast for each pixel such that the rays intersect the pixel at random locations over the pixel’s surface. Each ray is then traced independently, and the rendered color values are averaged together

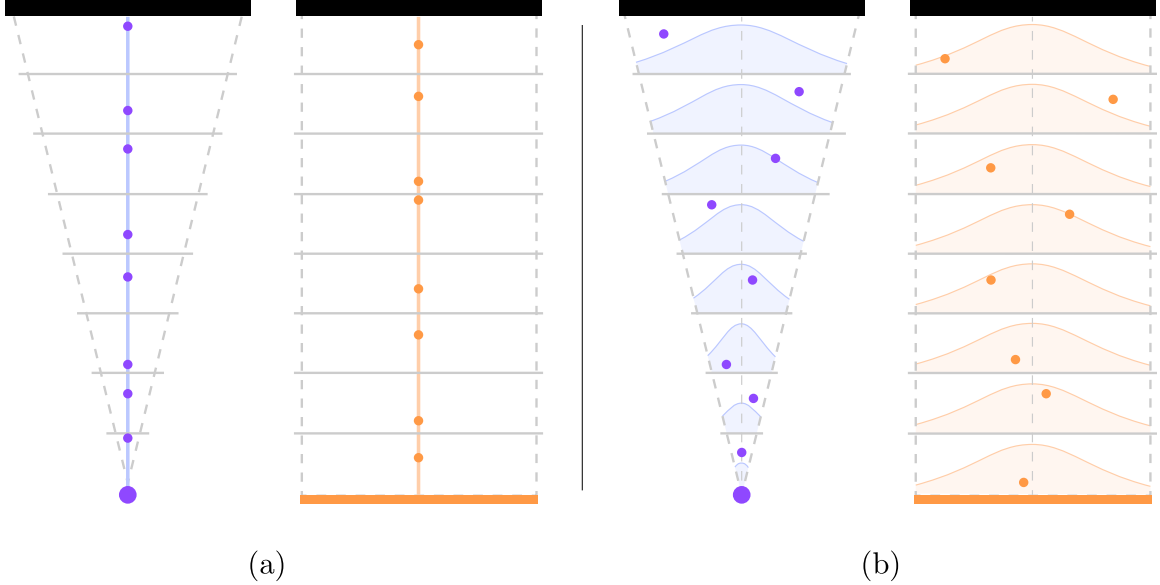


Figure 3.6: **Per-pixel pencil sampling strategies for parallel and cone beam geometries.** (a) In standard stratified sampling, the pixel frustum is divided into equal sized bins and samples are drawn from within each bin uniformly at random along the direction of the ray. (b) To approximate the full pencil beam, we additionally offset the sample perpendicular to the ray direction according to a normal distribution with standard deviation determined by the scaled pixel size. This deviation changes between each bin for cone beam geometries but does not change across bins for parallel beam geometries.

to produce the final pixel value. While very effective at reducing aliasing artifacts, this is a computationally expensive process due to the need to fully sample each pixel multiple times for every rendered frame.

We adapt this multi-sampling process in FlexAF by exploiting the inherent stochasticity of our training method. Because we are already sampling each pixel multiple times across training epochs, we can approximate multi-sampling by jittering the pixel’s center position d_x prior to ray sampling. This lets us avoid the large computational expense of multiple ray traces per pixel per iteration while still training over samples from the full volume of the X-ray pencil.

We also implement an alternative to this approach where we adjust the sample points provided by stratified sampling such that they more fully cover the volume of

the pencil rather than simply lying directly along the ray. This can be easily accomplished by locally jittering each point in the plane orthogonal to the ray direction. To keep samples within the bounds of the pencil, we draw the jitter offset value from a normal distribution with σ equal to the scaled pixel width times $\frac{2}{\sqrt{12}}$, which produces an in-plane variance equivalent to that of the pixel’s footprint [2].

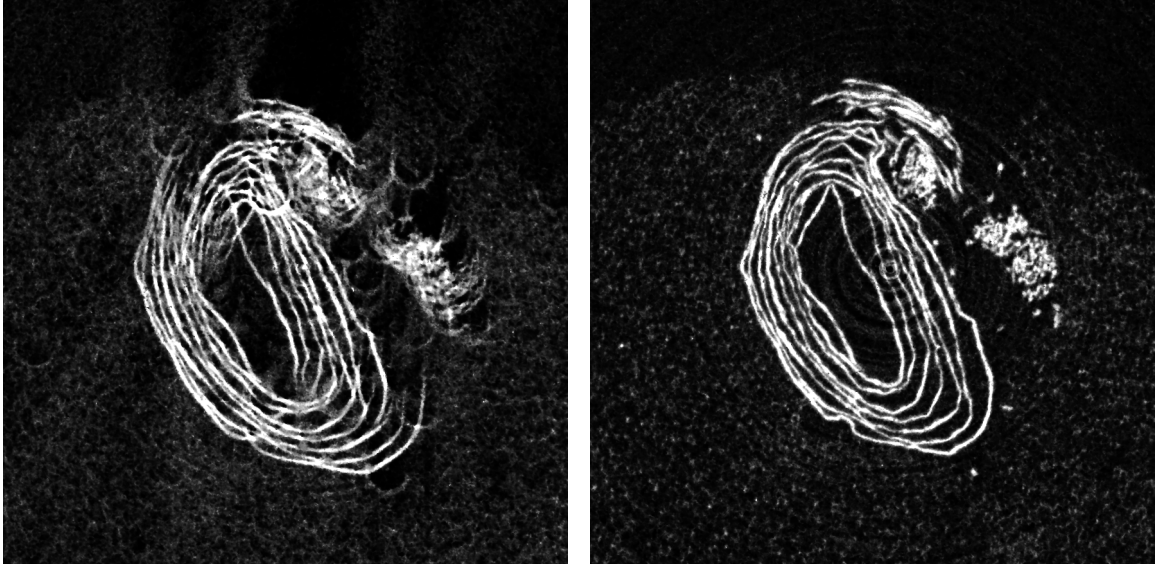
3.2.2 Learned camera extrinsics

X-ray camera misalignment, where the actual positions and orientations of the cameras (source and sensor combined) do not match the values recorded by the scanner, is a common problem for real-world CT acquisition. Misalignment can occur for a number of reasons, the most common of which is simply hardware limits on mechanical repeatability. It is a difficult proposition to precisely position the hardware for pixel perfect acquisition at every rotational step, particularly when the effective pixel sizes are at the micrometer and nanometer scale. As shown in Figure 3.7, camera misalignment can cause a range of reconstruction artifacts from minor to serious.

We implement an optional automatic misalignment correction system for the camera extrinsics which is learned jointly alongside the reconstruction process. For each projection image, we store two 4×4 tensors representing a homogeneous transform of the source and detector positions respectively and which are initialized to an identity transform. During training, we use each pixel’s projection image index to load and apply the corresponding transforms to the ray endpoints, s and d , prior to ray sampling. The transform matrices are set as learnable parameters and are updated during backpropagation alongside the volume parameters.² The transforms can be configured with either their own optimizer or they can share an optimizer with the volume network. In all of our experiments, we opt to use a standalone optimizer for the transforms as the shared optimizer typically leads to unstable transform learning.

For convenience and framework testing, we also implement manual post-alignment

²Since the final row of a homogeneous 3D transform should not be updated, we apply a stop gradient to this row.



(a) No post-alignment correction

(b) Manual post-alignment correction

Figure 3.7: **Misalignment artifacts in the papyrus scroll dataset.** Comparison of FBP slices without and with post-alignment correction. (a) Without correction, Point-like features become large crescent artifacts and the interior structure does not align. (b) With manual post-alignment correction of -30 pixels, the point-like features are resolved and the structure of the scroll is clearly defined and traceable. Learning the camera extrinsics removes the need to manually determine post-alignment corrections such as this.

correction. This is a commonly used technique where each detector’s position is shifted pixel-unit distances along its basis vectors (\vec{u}, \vec{v}) .

3.3 Volume model

Our reconstructed volume is modeled in the weights of a coordinate-based neural network which accepts a 3D point \mathbf{x} in the world coordinate frame and returns the learned attenuation coefficient $\mu(\mathbf{x})$. Our standard architecture, shown in Figure 3.8, is based off of that employed by many radiance field methods but adapted for CT reconstruction in a to-scale world coordinate frame.

3.3.1 Positional encoding

For our coordinate encoding $\gamma(\mathbf{x})$, we use the Gaussian encoding discussed in 2.2.2. This encoding is easy to implement, quick to evaluate, and tunable to each scan’s

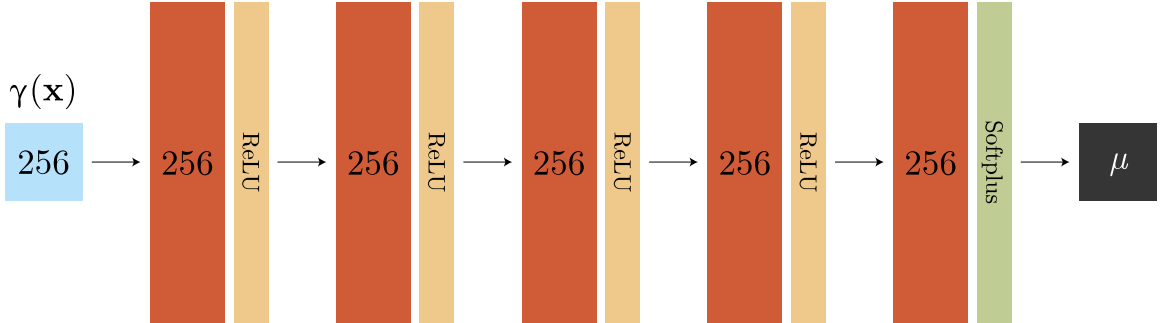


Figure 3.8: **The standard FlexAF neural volume architecture.** A single-energy FlexAF neural volume configured with a 5-layer MLP of width 256 and a Gaussian encoder with 256 features.

frequency content by varying the scale σ . As previously noted by Tancik et al., setting the scale to an arbitrarily large value decreases reconstruction quality [92]. Thus, we treat the scale as a hyperparameter which is manually tuned for each dataset.

In radiance field applications, the Fourier feature encoders are provided normalized coordinates in the range $[-1, 1]$ for positional encoding and $[0, 1]$ for Gaussian encoding. This is of little consequence for the view synthesis task as the absolute scale of the scene is largely unimportant for perspective rendering. We, however, wish to reconstruct in a world coordinate frame with physical units, where objects can be measured metrically and where we can reason about the size of the reconstructable volume.

Unfortunately, the straightforward solution of passing the raw world coordinates directly to the encoder does not work as well as one might hope. To understand why this might be, let us first consider that the raw world coordinate \mathbf{x} is simply the normalized world coordinate multiplied by a per-axis scalar vector \mathbf{g} which transforms the normalized coordinate to a specific unit of reference (e.g. millimeters): $\mathbf{x} = \mathbf{g}\hat{\mathbf{x}}$. If we substitute this into Eq. (2.7), we can see that the scale component of our raw world coordinates multiplies against \mathbf{B} , thus amplifying the Gaussian scale by some

volume-specific factor:

$$\gamma(\mathbf{x}) = [\cos(2\pi\mathbf{B}\mathbf{x}), \sin(2\pi\mathbf{B}\mathbf{x})]^\top = [\cos(2\pi\mathbf{B}\mathbf{g}\hat{\mathbf{x}}), \sin(2\pi\mathbf{B}\mathbf{g}\hat{\mathbf{x}})]^\top \quad (3.7)$$

While this is an operable solution in principle, it can be quite difficult to predict appropriate Gaussian scale values under such a scheme as it is inversely proportional to the volume’s size. An intuitive understanding of the Gaussian scale from radiance fields is that it controls the reconstruction’s frequency content with some approximately monotonic notion of resolution. That is, a larger Gaussian scale provides “more resolution.” It can be quite confusing, then, to observe that a volume with a 2 mm diameter would have a smaller Gaussian scale than a volume with a 1 mm diameter.

Our solution to this problem is to apply coordinate normalization dynamically during training and evaluation rather than when the data is loaded. Prior to training, we construct a coordinate scaling function from the scan’s minimum axis-aligned bounding box of all source and pixel positions in the world coordinate frame. This scaling function is stored as a framework parameter and is called immediately before coordinates are passed to the Gaussian encoder. This has the desired effect of providing an interpretable world coordinate frame for the user while still maintaining the intuitive properties of the Gaussian scale hyperparameter. We note that normalizing the coordinates on a per-volume basis only removes the inverse relationship between the volume size and the Gaussian scale and does not fix this value with respect to a particular spatial resolution. The scale must still be individually selected for each scan as before but with the advantage that the value now grows intuitively *with* the volume. For example, a volume with a 2 mm diameter would need twice the Gaussian scale of a volume with a 1 mm diameter to maintain a similar quality.

In line with the findings by Zheng et al. discussed earlier, we occasionally find

it necessary to increase the number of encoder features m alongside the Gaussian scale in order to improve reconstruction quality. Intuitively, as the scale increases for a fixed number of frequency features, so too does the sample sparsity increase in feature space. At some point, the feature space sparsity exceeds that required to learn the volumetric attenuation function, and the quality of the reconstruction is diminished. It is worth noting, however, that the number of features controls the size of the receptive field for the first layer in the neural network and thus also increases the network’s memory footprint. So while the scale hyperparameter can grow indefinitely, there is a practical limit to the range of frequencies which can be represented by the Gaussian encoding.

Learned camera extrinsics revisited

To control transform learning, we modify the frequency weighting method from BARF [50] for use with the Gaussian encoding. As discussed in 2.2.4, BARF applies a per-frequency weight $w_k(\alpha)$ to each frequency component of the encoded coordinate which progressively enables high-frequency learning as training progresses. This scheme is easy to implement for NeRF’s positional encoding as the frequency features are set to monotonically increasing values of 2^k and are identical for each spatial axis. In contrast, the Gaussian encoder uses frequency features $B \in \mathbb{R}^{m \times 3}$ which are drawn from a normal distribution, contain positive and negative values, and are independently selected along each spatial axis. We make a number of changes to the frequency weighting method to account for these feature differences.

Our first step is to redefine $\alpha \in [0, 1]$ so that it represents the percent of the total frequency range which has been fully enabled. When $\alpha = 0$, only the lowest frequency components are passed to the volume network, and when $\alpha = 1$, all frequency components will be passed. As before, we want to construct a weight function which smoothly enables the high-frequency components as α increases but only after a specific α threshold has been met. Though the Gaussian features can be both

positive and negative, only the absolute magnitude determines whether the feature corresponds to a low or high frequency. To easily filter our features according to their magnitude, we construct the normalized Gaussian feature matrix $\hat{\mathbf{B}} = [\hat{\mathbf{B}}_x, \hat{\mathbf{B}}_y, \hat{\mathbf{B}}_z]$ where $\hat{\mathbf{B}}_j \in [0, 1]^m$ is the normalized absolute value for axis j given by:

$$\hat{\mathbf{B}}_j = \frac{|\mathbf{B}_j| - \min |\mathbf{B}_j|}{\max |\mathbf{B}_j| - \min |\mathbf{B}_j|} \quad (3.8)$$

Intuitively, the values in $\hat{\mathbf{B}}$ represent the normalized positions of the original features within the per-axis range of absolute feature magnitudes. By thresholding the values of this matrix, we isolate features by frequency.

All that is left is to construct a modified weight function similar to (2.11) which is conditioned on $\hat{\mathbf{B}}$ rather than k . We use $0.5\hat{\mathbf{B}}$ as the lower threshold for each feature’s activation and linearly increase the weight for the range $\alpha \in [0.5\hat{\mathbf{B}}, \hat{\mathbf{B}}]$:

$$w_{\hat{\mathbf{B}}}(\alpha) = \begin{cases} 0 & \text{if } \alpha < 0.5\hat{\mathbf{B}} \\ \frac{1 - \cos(\frac{\alpha - 0.5\hat{\mathbf{B}}}{\hat{\mathbf{B}} - 0.5\hat{\mathbf{B}}} \pi)}{2} & \text{if } 0.5\hat{\mathbf{B}} \leq \alpha < \hat{\mathbf{B}} \\ 1 & \text{if } \alpha \geq \hat{\mathbf{B}} \end{cases} \quad (3.9)$$

Since the result of $w_{\hat{\mathbf{B}}}(\alpha)$ is a per-feature weight matrix of shape $m \times 3$, we take the average along the spatial dimensions to produce an m -length weight vector function $\bar{w}_{\hat{\mathbf{B}}}(\alpha)$. Finally, we apply the weight vector to the m -length coordinate components produced by the Gaussian encoding:

$$\gamma(\mathbf{v}; \alpha) = [\bar{w}_{\hat{\mathbf{B}}}(\alpha) * [\cos(2\pi\mathbf{B}\mathbf{v}), \sin(2\pi\mathbf{B}\mathbf{v})]]^T \quad (3.10)$$

During training, we linearly increase α after every mini-batch over a user-defined E number of epochs.

3.3.2 Standard network

Our standard neural volume network is an n -layer MLP with a fixed width w across all hidden layers. The input layer accepts m -length Gaussian encoded coordinate vectors $\gamma(\mathbf{x})$, and the output layer produces the estimated attenuation coefficient $\mu(\mathbf{x})$. It is important to note that negative output values from our network are illogical as they imply that the sample *amplifies* the X-rays rather than attenuating them. Thus, we want to make sure that we restrict the outputs of our final layer to positive values. In the appendices for Mip-NeRF [2], the authors introduce a shifted softplus activation $\log(1 + \exp(x - 1))$ to the density output of the MLP, replacing the ReLU activation used by NeRF. They note that this activation function improved training stability and led to slightly faster convergence rate during early training. We adopt the same approach and note a similar effect in our work. For all other layers, we apply ReLU activations.

3.3.3 Multi-energy network

Thus far, we have only considered CT projection images which were captured with the same incident X-ray energy. To design a multi-energy network, we start with the observation that scans across multiple incident energies share the same underlying structure. That is, the volume’s chemical composition does not change across scans, but the appearance of that composition in the X-ray projection images varies greatly across imaging parameters. To support heterogeneous X-ray energies in the same reconstruction, we want a network that models both this common volumetric structure and the scan-specific intensity functions which map that structure into the space of observed attenuation coefficients.

For inspiration, we look at how the X-ray attenuation coefficients vary across the chemical elements for a fixed incident X-ray energy. Figure 3.9 plots attenuation coefficients against the elemental atomic number, Z , for selected monochromatic X-ray energies across the 35 keV to 120 keV range. Except for a discontinuity from

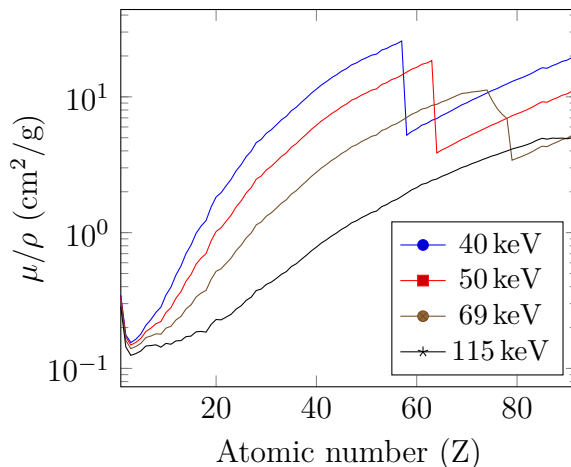


Figure 3.9: **X-ray mass attenuation coefficients plotted against the atomic number (Z) for various incident energies.** For multi-energy training, we’re interested in modeling the function which maps the atomic number (Z) to attenuation coefficients μ for a given incident X-ray energy. By slicing the 3D plot in Figure 2.5 along the energy axis, we gain insight into our desired mapping function for monochromatic beams. As the incident energy increases, the absorption edges shift to higher atomic numbers and the total attenuation across elements flattens. These plots use data from NIST SRD 126 [40].

an X-ray absorption edge that shifts across the elements as the energy increases, we can see from this that the attenuation coefficients represent a relatively smooth and well-behaved function.

Building from these observations, our multi-energy network is tasked with estimating the volumetric attenuation coefficients as a function of both world coordinates and the incident X-ray energy: $\mu(\mathbf{x}, k)$ (Figure 3.10). As before, the world coordinates are passed through a Gaussian encoder and n -layer MLP to produce a single output value, z . Since this value intuitively represents an uncalibrated estimate of the local elemental composition, we assign it the label z as a reference to the atomic number. The encoder and MLP are of almost identical construction to those in our single energy network, with the one difference being the use of a sigmoid activation on the final output layer which restricts z to the range $[0, 1]$.

As a first order approximation, we model the energy-specific intensity function

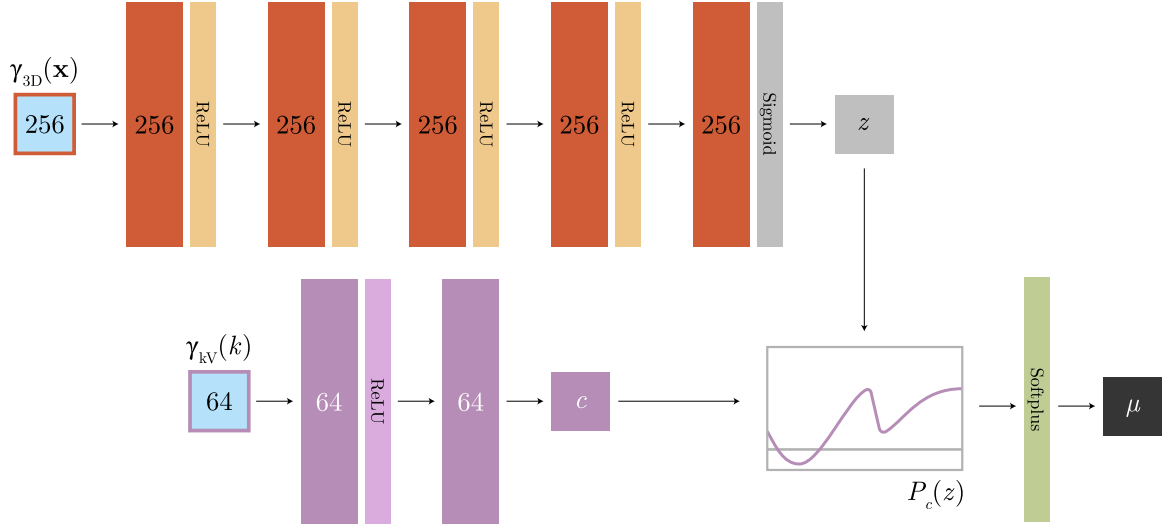


Figure 3.10: **The multi-energy FlexAF neural volume architecture.** Rather than learning the attenuation coefficients for a 3D coordinate directly, we model it as a function of common elemental structure z and the learned polynomial mapping P_c . We employ a slight modification of our standard architecture to generate z and a small Gaussian encoded-MLP to learn the energy-specific polynomial coefficients c .

which maps elemental composition to μ as an m -degree polynomial with coefficients that are learned for each incident energy in the training set. This is accomplished by passing k through a second Gaussian encoder and 2-layer MLP to produce the learned polynomial coefficients c . We then evaluate our polynomial with the elemental composition z from our first MLP:

$$P(z) = \sum_{i=0}^m c_i z^i \quad (3.11)$$

Here we again apply the shifted softplus to the polynomial’s output to ensure positive-valued attenuation coefficients from our model.

Practically, multi-energy training requires only one additional change to the FlexAF framework. On data load, we append the scan’s incident X-ray energy to the pixel tensor (3.5). It is thus passed into the ray tracer where it is appended to the ray samples prior to evaluation by the multi-energy volume model.

3.4 Training and evaluation

FlexAF follows a standard regression training loop. First, our datasets are loaded and converted into a flattened list of pixel tensors. From this list, a mini-batch of n pixels is drawn uniformly at random from the flattened list and passed to the differentiable ray tracer for evaluation. The ray tracer samples 3D coordinates along each ray and queries the neural volume for the given coordinates’ attenuation coefficients. The returned coefficients are integrated along the rays to produce a projection estimate \hat{y}_i for each ray in the mini-batch. We use the mean squared error between the estimated projection value and the pixel value from the projection image y_i to calculate the mini-batch loss for gradient backpropagation:

$$L_{\text{mse}}(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \tag{3.12}$$

After the learnable parameters of the neural volume and ray tracer have been updated, a new mini-batch is drawn without replacement from the flattened list of pixels. A training *epoch* occurs after all samples in the pixel list have been evaluated by the network. At this point, the list is refilled, randomized, and training proceeds as before.

3.4.1 Slice rendering

The most common view format for volumetric data is the *slice image* which is generated by plotting the volume’s attenuation coefficients on an intersecting plane. Though the intersection plane may be in any orientation and position with respect to the volume’s coordinate frame, often the plane is orthogonal to the Z axis of the world coordinate frame, and the full volume is exported as a stack of slice images which vary in Z. While our training method renders projection images via our differentiable ray tracer, we have specifically designed our volumetric models to support direct evaluation without the ray tracing framework. To render a slice image, we

sample a plane in the world coordinate frame to construct a regular, discrete grid of 3D coordinates. We pass these directly to our volumetric network to receive the attenuation coefficients for each coordinate. Our multi-energy model is unique in that we can render slice images for both the energy-dependent attenuation coefficients and the underlying z value which is shared across incident energies (Figure 5.16).

3.5 Attention mechanisms

Training a neural volumetric reconstruction with FlexAF is a computationally expensive task which only grows more demanding as the size and resolution of the volume increases: there are more pixels to train over in each epoch; the rays must be sampled at a finer rate; the scale and number of features of the Gaussian encoder must be increased proportionally; and the network size must grow to support a larger capacity. As such, training times for even moderately sized volumes can require multiple days of computation to reach convergence. It is therefore crucial that the training process be as efficient as possible, and that training attention is focused on the most important regions of the volume. In this section, we introduce two optional support features in FlexAF that we use to help control learning attention and improve reconstruction quality and runtimes.

3.5.1 Entropy pixels

The projection images in CT datasets often contain a significant amount of “empty space” where the X-rays pass only through air before striking the detector. While it’s important to accurately reconstruct all regions of the observed volume, even empty ones, such regions require far less computation to reach convergence than do the content-laden areas which capture the scan subject. To focus training attention on the most important regions of the projection images, we propose a hierarchical projection sampling approach based on image entropy known as *entropy pixels*.

Before training, we sweep a disk-shaped kernel of radius r across every projection

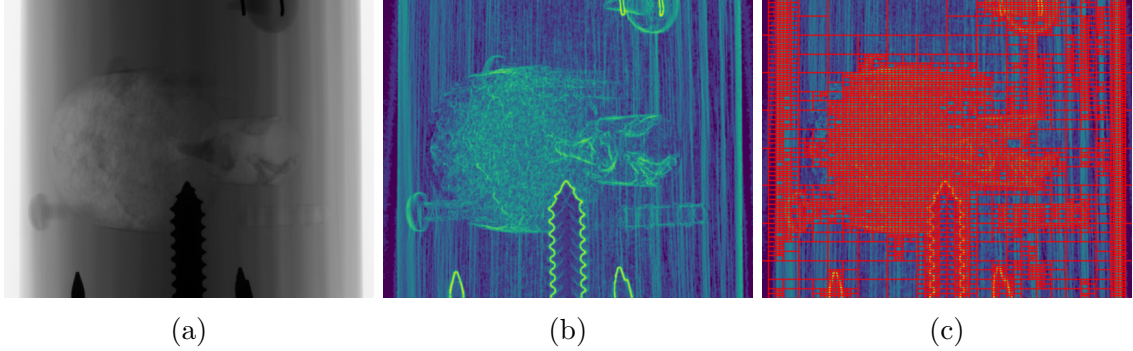


Figure 3.11: **Entropy pixels example for the Multi* dataset MS.01.02.** (a) The projection image is detailed but often low contrast. It is difficult to decide from projected attenuation alone which pixels need more attention. (b) After applying the image entropy filter, subtle details near the edges of the samples are enhanced. The wooden dowel in the lower right, which was hardly noticeable in the projection image, stands out clearly. (c) An entropy quadtree with restricted depth for illustrative purposes. At the beginning of each epoch, a single pixel sample is drawn from each tree leaf. By controlling the entropy threshold and tree depth, we effectively undersample low entropy regions and thus reduce the number of training iterations required to reach an acceptable reconstruction.

image and calculate the per-pixel entropy from the kernel region as:

$$e = - \sum_{i=0}^{255} p_i \log_2(p_i) \quad (3.13)$$

where p_i is the probability of the given gray value i computed from the full image histogram. This produces an *entropy image* which often highlights many of the subtle image features which are difficult to see in the raw projection images (Figure 3.11b).

Next, we build a quadtree [21] for the entropy image which splits leaf nodes when any pixel in the quadrant has entropy greater than the user-provided threshold E . To allow control over the surface area of the deepest nodes, the tree is bounded to a maximum depth D . When $D = \log_2(\max(W, H))$, where W and H are the image width and height respectively, the leaf nodes at depth D contain a single pixel. Intuitively, we have divided the image into a new grid-based structure where each leaf node in the quadtree represents a rectangular region, or *entropy pixel*, with a size

defined by the maximum entropy inside the region (Figure 3.11c). By construction, the smallest entropy pixels contain the “most important” regions of the image while the largest entropy pixels contain the “least important” regions.

Once we have computed the entropy pixels for each projection, we proceed to training. At the beginning of every epoch, we construct a new training set by drawing a single, original pixel uniformly at random from the bounds of every entropy pixel. This results in a reduced training set which is often significantly smaller than the full set of pixels — sometimes as much as 60% smaller — but which still captures the most detailed regions of the projection images.

3.5.2 Adjustable bounding volume

Often in CT reconstruction, the distance traveled by the X-rays between the source and detector is much larger than the diameter of the reconstructed scan volume. This is particularly true for micro-CT and nano-CT applications, where an extended source-to-detector distance (SDD) provides greater geometric magnification on the detector plane. As a point of reference, consider the datasets in our study. For the largest sample, the Multi* phantom (discussed in 4.3), our widest field of view is 10.75 cm while the SDD is 50 cm. Without a mechanism for focusing ray samples on the volumetric regions of interest near the world origin, we would waste valuable computing resources sampling the 80% of the ray length which passes through empty space.

Long ray lengths pose an additional difficulty for the Gaussian encoding used by our volume model. In 3.3.1, we noted that the Gaussian scale factor must be adjusted in order to accurately reconstruct high-frequency scene content, that is, increasing the scale factor enables the model to represent ever smaller features of the volume. However, as the bounds of the modeled volume grows, the relative size of our scene features grows smaller with respect to the normalized coordinate system that we provide the Gaussian encoding (Figure 3.12a). The effect of this is that, for a fixed

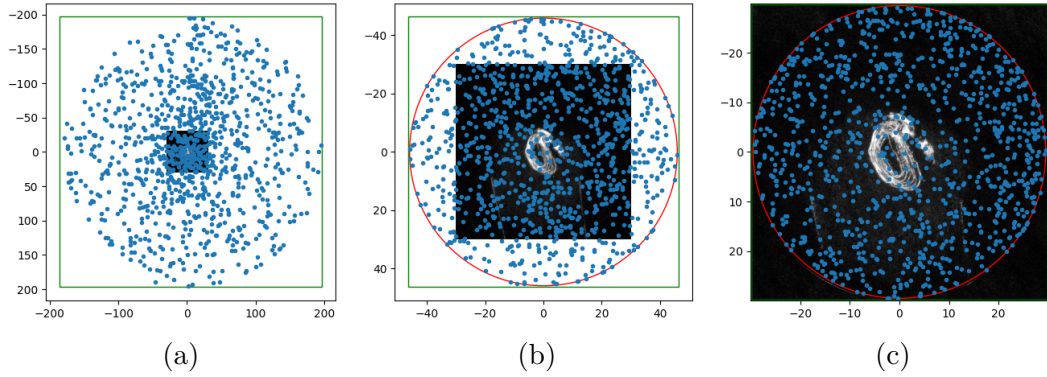


Figure 3.12: **Plot of the potential training bounding volumes with respect to the area of the reconstruction.** These plots visualize the size of various bounding volume configurations with respect to the reconstructable area of the papyrus scroll. The AABB of the Gaussian encoder is shown in green, the bounding cylinder is shown in red, and randomly selected ray samples are shown in blue. Coordinates are in millimeters. (a) No bounding volume is used, and we sample from the entire region between the X-ray source and detector for all projections. (b) We sample within the automatically-defined bounding cylinder and significantly reduce the size of the Gaussian AABB. (c) We sample within a manually-defined bounding cylinder inside the slice bounds and further reduce the size of the Gaussian AABB.

Gaussian scale, the quality of the reconstruction diminishes as the size of the volume grows. And as we have discussed, there is a practical limit to how large the Gaussian scale can grow without also increasing the neural volume’s size and computational requirements (see 2.2.2).

To address both of these issues, we implement a bounding volume system which focuses ray samples to the reconstructable region of the volume. During data load, we use the known scan geometries to calculate the reconstructable region of the volume as an axis-aligned bounding box (AABB) in terms of world coordinates. From this we construct a circular bounding cylinder which is centered on the world origin and parallel to the world Z axis. By default, we set the diameter of this cylinder to be slightly larger than the diagonal of the AABB, but the diameter may also be specified manually (Figure 3.12b). During ray tracing, the rays are clipped against either the

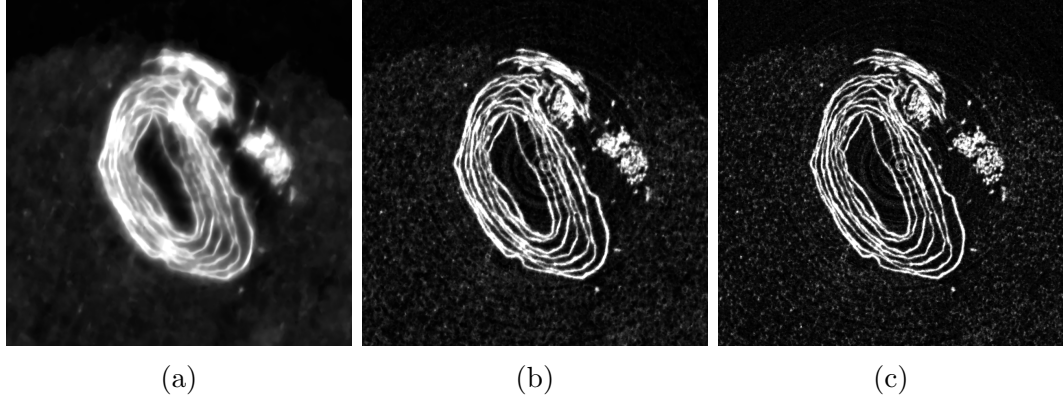


Figure 3.13: **The effect of adjusting the bounding volume on reconstruction quality.** (a) Bounds set to contain all source and detector positions. This is equivalent to a bounding cylinder with a diameter of 278.4 mm. (b) Bounds set to a 91.6 mm bounding diameter, just larger than the width of the projection images’ field of view. (c) Bounds set to a 59 mm bounding diameter, a bounding diameter fully contained in the projection images’ field of view.

bounding cylinder or the AABB prior to ray sampling. As a result, all training samples are drawn from within the bounds of a much smaller reconstructable region. Likewise, we may significantly reduce the range of the Gaussian encoder’s coordinate normalization function. As no training samples will ever be drawn from outside the bounding volume, we set the normalization bounds to be slightly larger than the bounding volume.

All the experiments in our study clip against the cylindrical bounding volume during ray sampling. Occasionally, we manually specify the diameter of the bounding cylinder in order to situationally improve the quality of the FlexAF reconstruction (Figure 3.12c). The effect from adjusting the size of the bounding volume can be quite dramatic. To demonstrate this point, we reconstruct a single slice from the papyrus scroll dataset using FlexAF, varying only the bounding volume diameter. The results of this experiment are shown in Figure 3.13.

The reconstruction for the largest bounding volume (i.e. the bounding box which contains all X-ray sources and detectors) captures only the most prominent signals:

the general structure of the scroll and the mere presence of the polyethylene foam (Figure 3.13a). As the size of the bounding volume decreases to a cylinder just larger than the projection field of view (Figure 3.13b), the scroll structure is clarified and one can begin to distinguish the cell structure of the foam. Finally, with a manually-defined bounding volume which fits fully inside the projection field of view (Figure 3.13c), the foam structure comes more clearly into focus.

CHAPTER 4. DATASETS

“It soon appeared from tests that the rays had penetrative power to a degree hitherto unknown. They penetrated paper, wood, and cloth with ease; and the thickness of the substance made no perceptible difference, within reasonable limits.”

– *Dr. Wilhelm Röntgen, The New Marvel in Photography, McClure’s Magazine, 1896*

Our work is predicated on the idea that heterogeneity in X-ray projection images is not a problem to be avoided but rather an opportunity for extracting more information with CT than we previously thought possible. Of course to test this idea, we need heterogeneous CT data. Our properties of inquiry here are straightforward: strong geometric misalignment of the X-ray cameras, projections captured at different effective resolutions, and projections captured with different incident energies and exposure settings. Since commercially available CT scanners do not explicitly support such heterogeneity, we test FlexAF on datasets which demonstrate these properties either individually or in composite. In this chapter, we discuss the micro-CT datasets used in this study and their properties. We begin with two datasets which are fairly conventional by CT standards, but which allow us to validate the correctness of our framework for reconstruction in general. We follow this with a description of a new composite dataset collected specifically for this study which we call the Multi* (pronounced *multi-star*) dataset.

4.1 The Shepp-Logan phantom

The Shepp-Logan phantom [89] is numerical phantom which is frequently used to validate and test the properties of CT reconstruction algorithms in a controlled manner. Originally constructed in 2D to emulate the shape and attenuation coefficients of the human head, the Shepp-Logan phantom is defined as the sum of gray levels from ten overlapping ellipses on the XY plane (Figure 4.1). We use a 3D variant which defines additional ellipses along the Z axis as well [56].

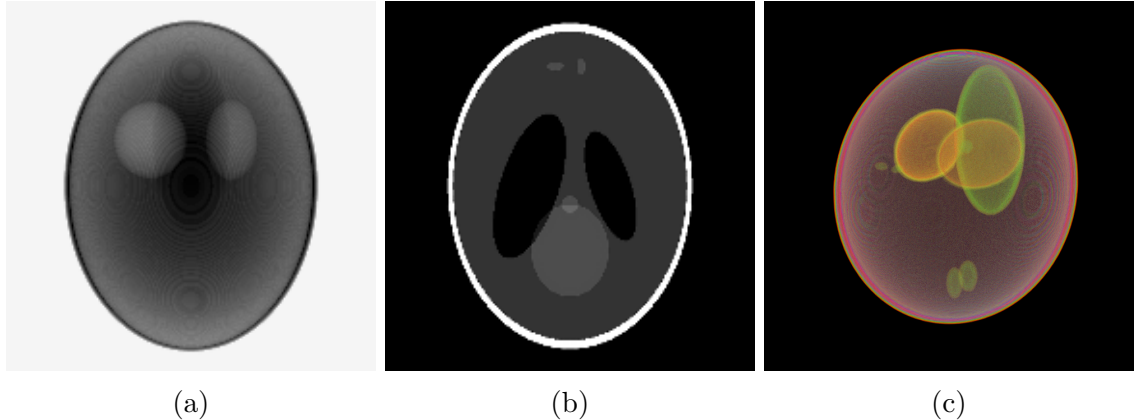


Figure 4.1: **Projections and slices of the Shepp-Logan phantom.** (a) A simulated, parallel projection used for training, rendered from the “end on” view of the largest ellipse. (b) A single slice which exemplifies the structural complexity and dynamic range of the phantom. (c) A volume render of the full phantom, taken slightly above and to the right. Color mapping is enabled to enhance interior feature visibility.

Since the Shepp-Logan phantom is a virtual object, we can render both the volume and projection images with selectable resolution. In our study, we generate the volume on a $512 \times 512 \times 512$ grid, then downscale it to $192 \times 192 \times 192$ to reduce aliasing artifacts from the generation process. We then simulate a cylindrical CT scan by rotating and resampling the volume around the volumetric Z axis. Projection images are formed by taking the line integral along the X axis of this new, rotated grid. This process is parameterized by the desired angular range of total rotation and the rotational step size between each projection image, enabling us to test FlexAF for both limited angle and sparse CT reconstruction tasks.

Since the Shepp-Logan phantom lacks a real-world coordinate system, we manually define a world coordinate frame in “millimeters” which we can use to initialize our X-ray cameras for training. To have a volume of approximately the same scale as our other micro-CT datasets, we first set the camera pixel size to $50 \mu\text{m}$. Due to our parallel projection geometry, this in turn defines the edge length of the volume as $192 \text{ px} * 0.05 \text{ mm/px} = 9.6 \text{ mm}$ and our reconstructable area as a 9.6 mm^3 cube centered on the world origin. To define our camera geometries, we set the X-ray

source and detector center positions to be on opposite sides of the origin and fully outside the volume for all rotational positions. This results in a source-to-detector distance of approx. 16 mm. Per-pixel center and source positions are generated using the equations for parallel geometries discussed in 3.1.

4.2 The papyrus scroll dataset

Our second dataset is a micro-CT scan of a papyrus scroll which was constructed as a test proxy for the virtual unwrapping software, Volume Cartographer [65, 87]. The scroll is formed of a single sheet of papyrus which has been rolled tightly and then wrapped and tied with natural fiber twine. Though the papyrus has writing on its surface, the ink has very low contrast against the papyrus and is not readily visible to the naked eye in the CT slices. Before scanning, the scroll was wrapped in open-cell polyethylene packing foam and affixed to the scanner’s sample stage with paper tape. Though these materials do appear in the reconstructed scan, they were specifically chosen for their relatively low attenuation. The scan was acquired with a prototype SkyScan 1173 micro-CT scanner at a $26.337\ \mu\text{m}$ pixel size and with an incident X-ray energy of 30 kV. Projections were captured with a 0.2° step size over a full 360° range for a total of 1800 projection images of size 2240×2240 .

The papyrus scroll is a multi-material sample, but many of these materials exhibit very similar attenuation, and the reconstruction generally has a relatively narrow dynamic range. Thus, our primary interest in this dataset lies in its varied and complex structural properties (Figure 4.2). Unlike the other samples in our study, the papyrus scroll scan shows many high-frequency features which are often separated by irregularly-sized gaps of air. The interior of the scroll itself is composed of papyrus wraps which are quite thin ($120\ \mu\text{m}$ to $200\ \mu\text{m}$) and have sharply-defined edges. In contrast, the polyethylene foam presents as a semicircular region of “noise” which surrounds the scroll for much of the volume. This “noise” is extremely irregular in size and shape, varies in edge definition, and changes structurally every 1 to 2 slices.

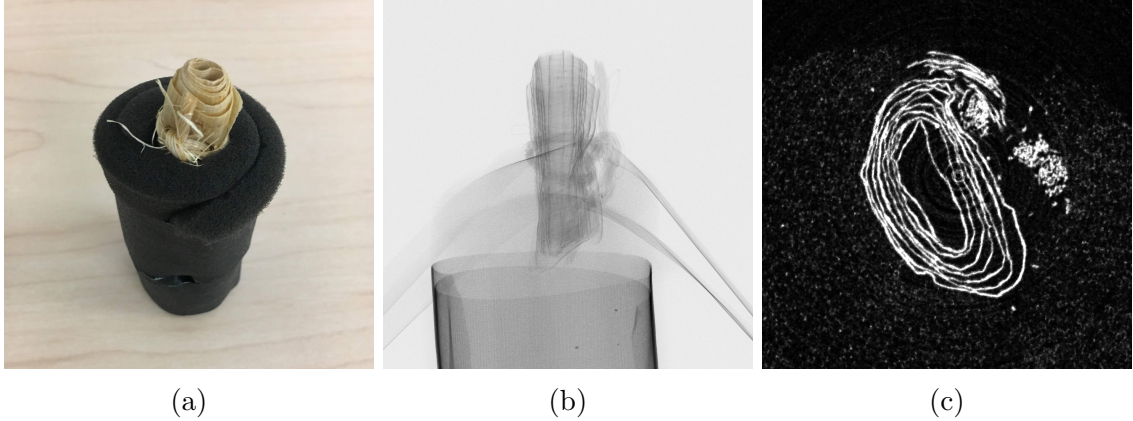


Figure 4.2: **Images of the papyrus scroll dataset.** (a) A photograph of the scroll and twine wrapped in the polyethylene packing foam. (b) An X-ray projection image from the CT scan. The scroll appears to float due to the low attenuation of foam. (c) A CT slice reconstructed with FBP, cropped to the central region containing the scroll. Despite the low attenuation, the structure of the foam is just visible, while the structure of the scroll and twine are crisp and clean.

Additionally, the papyrus scroll scan naturally exhibits extreme misalignment due to mechanical issues at the time of the scan (Figure 3.7). Generating an accurate reconstruction with FBP requires a post-alignment shift of -30 pixels, an extraordinary misalignment of almost 0.8 mm for this 26 μm scan. These properties combine to create a unique challenge for our automatic extrinsic calibration method. When the X-ray cameras are misaligned, the effect of that misalignment is readily apparent in the reconstruction (Figure 3.7). Even misalignment of a few pixels is enough to produce the tell-tale crescent artifacts which signify calibration issues. We use this dataset to test our framework’s ability to reconstruct a variety of high-frequency features and to do so in the face of extreme misalignment.

4.3 The Multi* dataset

We test our framework for multi-resolution and multi-energy reconstruction using a new dataset collection called the Multi* dataset (Figure 4.3). This collection was designed specifically for our work and contains 14 micro-CT scans of the same sample, scanned under varying conditions. The heterogeneity across the entire collection

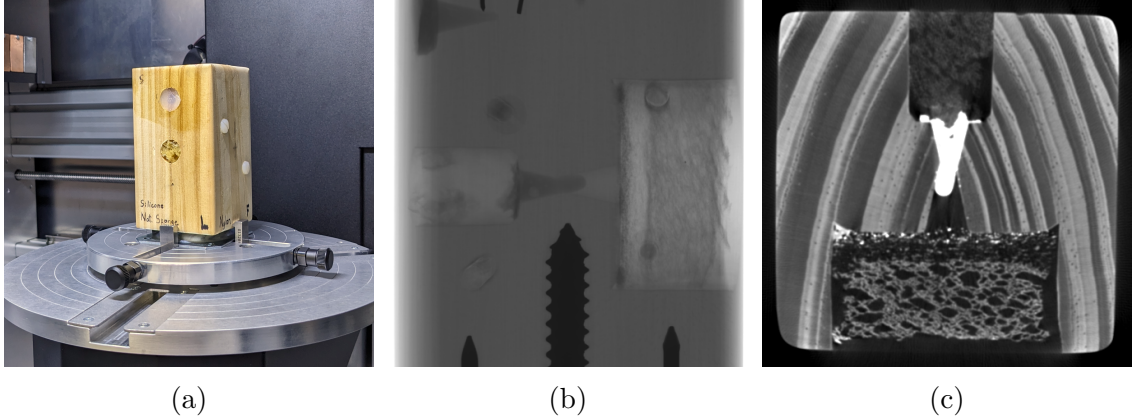


Figure 4.3: **Images of the Multi* proxy and dataset.** (a) A photograph of the proxy mounted in the SkyScan 1273 with the Left and Front faces visible. (b) A projection image from the MS.01.01 scan showing the complex inner structures of the proxy and the various embedded materials. (c) A CT slice from MS.01.01 reconstructed with FBP. Despite the relatively low resolution ($140\ \mu\text{m}$, the cell structure of both the synthetic and real sponges is quite visible. One can also just resolve the small pores in the base wood block.)

lends the dataset its name. Together the scans capture multiple resolutions, incident energies, X-ray filters, exposure settings, and capture positions. Additionally, the sample (the Multi* *proxy*) is a multi-material, multiscale object which is designed with extremely small features, extremely large features, features of low attenuation, and features of high attenuation. Though each scan is a standalone CT dataset which can be reconstructed on its own, we combine projections from across the collection at training time to build heterogeneous “scans” on-the-fly.

4.3.1 The Multi* proxy

The core structure of the Multi* proxy is a solid block of pine with dimensions of approx. $5.7\ \text{cm} \times 5.7\ \text{cm} \times 10.2\ \text{cm}$. Each of the six faces of this block is embedded with one or more materials representing a wide range of densities and structures and is labeled according to both its position on the block (F for front, B for back, L for left, etc.) and the embedded materials (Nylon, Aluminum, etc.). We provide here the layout of each face and a description of the embedded materials. For visual reference,

a diagram of each face is provided in Figure 4.4 and CT slices showing the various embedded materials are shown in Figure 4.5.

Front and back faces

The front face (F) features two 3.175 mm Nylon bolts of 25.4 mm length and with a 9 mm diameter bolt head. The bolts have been fully inserted into pre-drilled holes and are glued into place with a commercially available quick-dry adhesive to the end of the bolt shaft. The bolts are equally offset from the face's center, one to the top left and the other to the bottom right. During construction, the shaft of the right bolt was effaced by a recess which was drilled into the right block face.

The back face (B) is embedded with two 3.175 mm wooden dowels made of poplar. The dowels are 25.4 mm long, have been fully inserted into pre-drilled holes, and are glued into place with the quick-dry adhesive. Like the Nylon bolts, the dowels are equally offset from the face's center in an identical diagonal configuration.

In the CT slices, the bolts appear with near constant attenuation, the exception being small cavities of air on the interior of the shaft. The threads are easily distinguishable at our largest reconstructed pixel size of 140 μm . The wood grain of the dowels runs perpendicular to that of the base block and is much less varied in attenuation. The glue appears as a bright outer coating on the ends of both the bolts and the dowels.

Left and right faces

The left face (L) has two small, circular recesses which are approx. 14.4 mm in diameter and 18.7 mm deep. The first recess is perfectly centered in the block face and is filled by a piece of natural sponge. The second recess lies directly above the first by a vertical offset of approx. 3 cm and is entirely filled with silicone adhesive. The right face (R) has a single, large circular recess with a 38 mm diameter and a depth of 19 mm. This recess contains a piece of synthetic sponge made of polyester and an unknown plastic. Both sponges are glued into place with a silicone adhesive. A small,

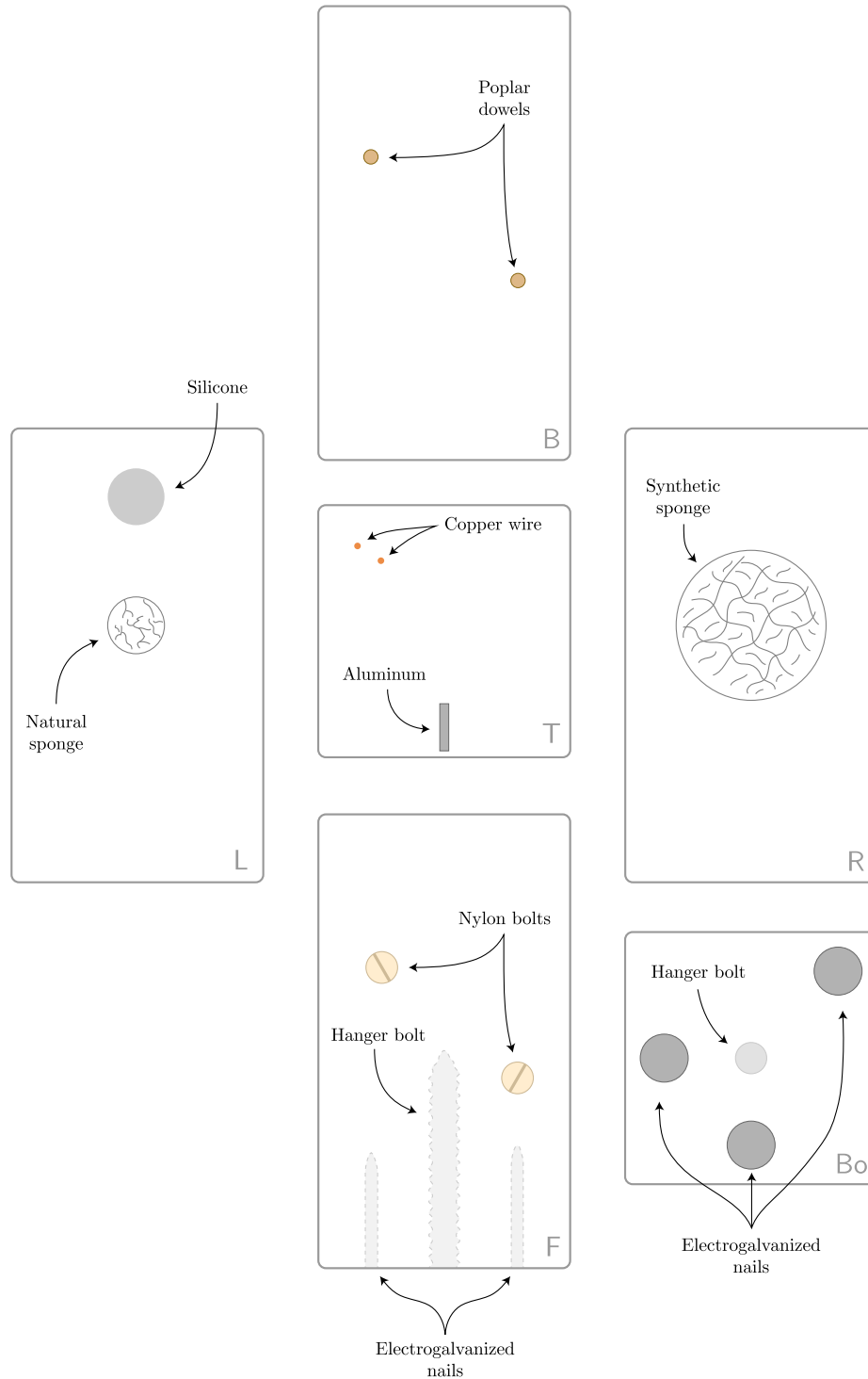


Figure 4.4: **A diagram of the faces and embedded materials of the Multi* proxy.** Each face of the Multi* proxy is embedded with materials which vary in feature size and chemical composition. This combination in a single object provides a comprehensive basis for evaluating both multi-resolution and multi-energy reconstruction tasks.

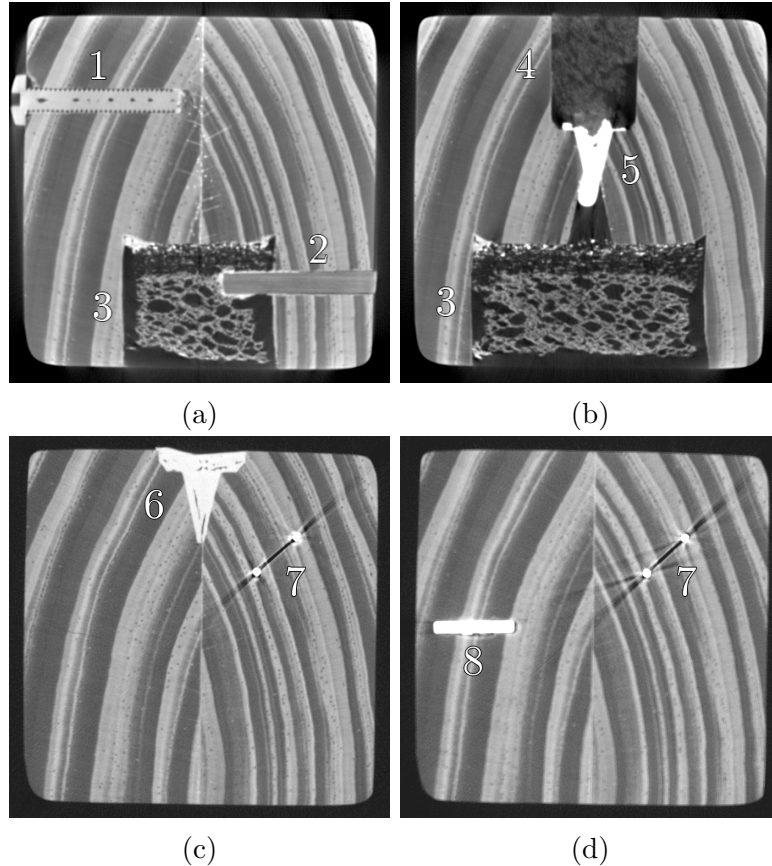


Figure 4.5: **Objects of interest inside the Multi* proxy CT reconstructions.** These four CT slices demonstrate the interior structures and relative attenuation for the various embedded materials. The numbers correspond to the following materials: (1) nylon bolt, (2) wooden (poplar) dowel, (3) synthetic sponge, (4) natural sponge, (5) silicone-filled interior channel, (6) silicone-filled exterior recess, (7) copper wire inserts, and (8) aluminum square insert.

cone-like channel created by the drill bit connects the left and right sponge recesses. This is filled with silicone adhesive to provide a high attenuation feature of interest on the interior of the sample which is entirely obscured from external observation.

Top and bottom faces

The top face (T) and bottom face (Bo) are embedded with our highest attenuating materials. The top face has two strands of solid core copper wire in the upper left corner of the face and a rectangular piece of aluminum in the bottom center. The strands are arranged in a diagonal pattern approx. 8.5 mm apart and are approx.

1.3 mm in diameter (16 AWG) and 25.4 mm long. They are glued into place with quick-dry adhesive applied to both ends of the wire. The aluminum is a thin, semi-square sheet which nominally measures $2.5\text{ cm} \times 2.5\text{ cm} \times 6.35\text{ mm}$. As this piece was cut by hand from a longer strip of aluminum, a small “point” extends from one edge of the piece, and thus the piece is slightly longer than 2.5 cm along this axis. The aluminum was hammered into the top face with the point extending down into the wood block. This resulted in deformation of the uppermost edge of the aluminum and the creation of microscopic fractures in the block which are visible in the CT reconstructions.

The bottom face contains three electrogalvanized steel nails on the edges of the face and a single steel hanger bolt in the middle of the face. The nails are 2.22 cm long and have a shaft diameter of 3 mm. They are arranged in a triangular pattern: one in the middle of the bottom edge, one in the middle of the left edge, and one in the upper right corner. The hanger bolt lies in the center of the face and has an 8 mm outer diameter. The bolt shaft on the interior of the block is approx. 5 cm long. The threaded shaft on the exterior of the block provide a means for mounting the block to a stable mounting plate for scanning.

4.3.2 Acquisition settings

We collected 14 micro-CT scans of the Multi* proxy in a single session using a Bruker SkyScan 1273. Collectively, these scans capture the proxy at two capture positions, four resolutions, and six peak incident energies. All scans were captured with the ultimate intent of constructing heterogeneous training sets for FlexAF by drawing projections from across multiple scans. As such, each scan is rotationally full and has a fixed rotational step size that, in most cases, is a multiple of 0.15° . This makes it easy to reason about how projections should be composed across incident energies (e.g. for each angle, draw projections by alternating between each energy) and scan resolutions (e.g. the higher resolution scan rotationally samples at exactly

	ID	Voxel size (μm)	Energy (kV)	Exposure (ms)	Filter	W \times H \times N
Multi-resolution	MS.01.01	140	45	560	–	768 \times 486 \times 601
	MS.01.02	70	45	560	–	1536 \times 972 \times 1201
	MS.01.03	35	45	560	–	3072 \times 1944 \times 2401
	MS.01.04	20	45	800	–	4992 \times 1944 \times 2401
	MS.01.05	20	45	100*	–	4992 \times 1944 \times 2401
	MS.01.06	35	45	100*	–	3072 \times 1944 \times 2881
	MS.01.07	35	45	50*	–	3072 \times 1944 \times 2881
Multi-energy	MS.02.01	70	35	1100	–	1536 \times 972 \times 1201
	MS.02.02	70	50	465	–	1536 \times 972 \times 1201
	MS.02.03	70	50	750	Al 0.5 mm	1536 \times 972 \times 1201
	MS.02.04	70	70	370	Al 0.5 mm	1536 \times 972 \times 1201
	MS.02.05	70	90	270	Al 1 mm	1536 \times 972 \times 1201
	MS.02.06	70	120	475	Cu 0.5 mm	1536 \times 972 \times 1201
	MS.02.07	35	120	475	Cu 0.5 mm	3072 \times 1944 \times 2401

*Dataset is intentionally underexposed.

Table 4.1: **Scans and parameters in the Multi* dataset.**

double the rate of the lower resolution scan). The SkyScan 1273 captures an inclusive range of 0° to 360° , thus each scan has one additional projection image than might otherwise be expected.

For purposes of comparison, we reconstructed all scans with the vendor-provided NRecon software. All scans come with a sidecar metadata file in an `.ini`-like format which describes all capture parameters and the NRecon settings used for reconstruction. Table 4.1 summarizes the most important scanning parameters for our study.

The scans are organized by their respective capture positions into two groups of seven and are labeled according to the pattern `MS.{POS}.{NUM}`, where `POS` is the group identifier and `NUM` is an index within the group. The field of view of the first capture position is centered on the vertical center of the Multi* proxy and chiefly captures the high frequency, low attenuation sponges. The field of view of the second capture position is centered 19.466 mm above the first and captures the high attenuation materials at the top of the proxy. As the proxy was never removed from the scanner during the session, scans acquired from the same capture position are inherently aligned, and the scans acquired at different capture positions are related

by a rigid transform.¹ Broadly, the capture settings used in each group correspond to our experimental goals; the first group is configured for multi-resolution experiments while the second is configured for multi-energy experiments.

Multi-resolution scans

The scans in the multi-resolution group begin with an effective pixel size of 140 μm and increase in resolution to an effective pixel size of 20 μm . The SkyScan 1273 has a fixed source-to-detector distance of 500 mm. All scans in the 35 μm to 140 μm range were captured with a source-to-sample distance of 234.151 mm, and pixel binning was applied at scan time to decrease the size of the captured image by a factor of 2 and 4. The 20 μm scans were captured at a source-to-sample distance of 133.801 mm. Due to the limited field of view of a single projection at this sample distance, projections were acquired and stitched with a 2x offset capture to ensure that the entire width of the proxy remained visible.

Additionally, this group contains three *underexposed* scans captured at 20 μm and 35 μm pixel sizes. These scans are included for the eventual testing of the effect of underexposure on FlexAF’s reconstruction quality. The two underexposed 35 μm scans differ from the other scans at this pixel size in that they were captured with a rotational step size of 0.125°. It is worth noting that SkyScan capture software automatically applies flatfield correction at capture time, thus the underexposed projections do not look dark as one might expect but rather have a “washed out” appearance.

Multi-energy scans

Our multi-energy scans capture six peak incident energies in the range 35 kV to 120 kV. As the energy increases, we accordingly lower the exposure times and add filters to the X-ray beam to mitigate beam hardening artifacts in the reconstructions. All scans have a 70 μm pixel size except for the final scan (MS.02.07) which has a 35 μm pixel size.

¹Assuming perfect mechanical calibration and alignment.

4.4 Dataset formats

The real world datasets in this study are all captured on Bruker SkyScan systems and are stored in the native SkyScan format. This format is a flat directory containing the flatfield-corrected projection images in a 16-bit grayscale TIFF series. A sidecar metadata file provides all scan settings in an easily parsable `.ini`-like format. Optionally, the SkyScan dataset format also provides a “post-scan,” a sparse set of projections captured at the end of the scan to assist in estimating misalignment effects from thermal drift or expansion of the X-ray source. We do not make use of this data in this study, but note its potential importance for future modeling of X-ray source properties.

Rather than conforming all input datasets to a common on-disk format, we provide a general purpose data loader API which detects the scan’s format and calls a format-specific backend loading function. To support a new scan format, developers add two functions to the library: one which detects the format and a second which can load the dataset into a FlexAF-specific in-memory structure. From the end user perspective, users simply provide the path to the dataset and FlexAF takes care of the rest. For the purposes of this study, projection loading is additionally parameterized by the subrange and stride of projections to be loaded as well as whether manual post-alignment correction should be applied, projections should be scaled by a user-provided factor, etc.

4.4.1 Heterogeneous dataset construction

We build upon our existing dataset loading functionality to enable on-the-fly construction of heterogeneous datasets from the Multi* collection at training time. We define a simple JSON batch file format that lists the scans and their respective loading parameters. Our dataset loader reads this batch file and returns an aggregate training set constructed from all listed scans. An example batch definition for drawing rotational samples from two scans is shown in Listing 4.1.

```
{
  "datasets": [
    {"data_dir": "data/MS.02.01", "start": 0, "skip": 2},
    {"data_dir": "data/MS.02.02", "start": 1, "skip": 2}
  ]
}
```

Listing 4.1: **Example definition for a batch file of two CT scans.** This batch file defines a heterogeneous scan which interleaves the projection images from two Multi* scans. In this example, FlexAF would be trained on dual incident energies of 35 kV (MS.02.01) and 50 kV (MS.02.02) which alternate between every rotational step.

CHAPTER 5. EXPERIMENTS

“‘Now, then,’ said [Röntgen], smiling, and with some impatience, when the preliminary questions at which he chafed were over, ‘you have come to see the invisible rays.’”

– *H.J.W. Dam, The New Marvel in Photography, McClure’s Magazine, 1896*

In this chapter, we test the capabilities of FlexAF across a wide range of CT reconstruction tasks. With our emphasis on dataset flexibility, it is tempting to evaluate FlexAF as a niche algorithm that’s only applied in those circumstances where existing algorithms won’t perform well. We are not interested in this sort of evaluation. Rather, we seek a general purpose method that is adaptable to heterogeneous datasets but which still performs well for traditional reconstruction tasks. Thus, we begin by exploring the abilities and limits of FlexAF for standard reconstruction tasks before moving on to the more challenging tasks of automatic extrinsic calibration, reconstruction from multi-resolution projections, and reconstruction from multi-energy projections.

As in similar studies, we quantitatively evaluate our methods by comparing our reconstructions against those produced by FBP. Using the process described in 3.4.1, we render a slice (or slice stack) with approximately the same world bounding box and sample rate as the FBP volume. We then compute the peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and normalized mutual information (NMI) comparison metrics between these two volumes. PSNR is reported using the logarithmic decibel (dB) scale, where good values lie in the 30 dB to 60 dB range and higher is better. SSIM is in the range $[-1, 1]$, where -1 represents anti-correlation and 1 represents perfect similarity between the compared images. When evaluating multiple slices, we take the mean SSIM across all slices. NMI is a commonly used metric for image alignment problems where the intensity ranges of the input

Dataset	PSNR	SSIM	NMI	Epochs	Time
Shepp-Logan (FBP)	23.578	0.945	1.541	–	4s
Shepp-Logan (Baseline)	30.076	0.974	1.523	200	67.1h
Shepp-Logan ($i = 215$)	30.092	0.973	1.520	200	67.63h
Shepp-Logan (E. Pix.)	29.972	0.973	1.519	200	37.29h
Shepp-Logan ($\sigma = 24$)	30.472	0.970	1.520	200	70.23h
Papyrus scroll (Baseline)	28.604	0.659	1.088	300	64.54h
Papyrus scroll ($i = 405$)	28.479	0.651	1.088	300	66.46h
Papyrus scroll (12 slices)	28.393	0.629	1.077	50	66.7h
MS.01.01 (Baseline)	37.785	0.916	1.422	1000	38.81h
MS.01.01 ($i = 512$)	37.838	0.914	1.421	1000	39.56h
MS.01.01 (100 slices)	27.080	0.769	1.284	34	7d10h
MS.01.02 (Baseline)	29.772	0.898	1.327	450	64.7h
MS.01.02 ($i = 512$)	29.628	0.946	1.365	450	68.01h

Table 5.1: **Standard reconstruction result metrics.** For each dataset, the best reported metric is in bold.

images may differ. Metric values are in the range $[1, 2]$, where 1 means the images are perfectly uncorrelated and 2 means the images are perfectly correlated.

Briefly, we note a few implementation details regarding our evaluation. First, the NRecon reconstruction software automatically applies a circular mask to its reconstructed slices in order to focus attention on the provably accurate central image region. To provide a fair comparison between our reconstructions and the reference volumes, we likewise mask our slices when calculating our comparison metrics. Second, we occasionally find that our continuous coordinate system is shifted by a few pixels from that of the FBP volume. In these cases, we shift our sample coordinates prior to sampling the neural volume in order to improve the volume alignment.

5.1 Standard reconstruction

For our standard reconstruction tasks, we evaluate FlexAF’s ability to reconstruct regular CT scans when little to no flexibility is required. For each dataset, we define a baseline FlexAF reconstruction which is trained on every pixel in the training set and which uses the interval schedule described in 3.2. We explore the effect of many

of FlexAF’s features (e.g. ray sampling patterns, entropy pixels, volume bounds) in comparison to both the FBP and baseline FlexAF reconstructions. Where applicable, we apply the same manual post-alignment correction values that were used to generate the FBP reconstructions. Table 5.1 lists our quantitative results for all standard reconstruction experiments.

5.1.1 Shepp-Logan

We reconstruct the entire volume of the Shepp-Logan phantom using 180 projection images of size 192×192 . The projections are rendered at a 1° angular offset over the rotational range $[0^\circ, 180^\circ)$. Our baseline FlexAF method uses a 6-layer MLP of width 256 and a Gaussian encoder with $\sigma = 8$ and 384 features. The interval scheduler is configured with a multiplier $m = 8$ and reaches a maximum of 215 ray intervals after 11 epochs. We experiment with three variations on the baseline FlexAF configuration: (1) we use a constant 215 intervals rather than the interval schedule, (2) we enable entropy pixels to speed up training, and (3) we increase the Gaussian encoding scale by a factor of 4 to $\sigma = 24$. We compare all FlexAF reconstructions against both the original phantom and the FBP reconstruction of the simulated projections.

Quantitatively, all FlexAF methods outperform FBP for both the PSNR and SSIM metrics but slightly underperform on the NMI metric. Figure 5.1 visually compares the original phantom against the FBP and baseline FlexAF reconstructions. The differences which explain these metrics are quite subtle. FBP does a better job at modeling the uniform intensities on the interiors of the ellipses, but the reconstruction overall appears to be blurrier than the FlexAF reconstruction. This blurring results in strong error on the edges of the ellipses, where the boundaries of intensities meet.

Though difficult to see in the raw slices, the FlexAF reconstructions all get brighter as you move closer to the center of the phantom (Figure 5.2). This reconstruction artifact, known as *cupping*, is usually caused by a dense surface in the sample absorbing all the low-energy X-rays from a polychromatic X-ray beam in a process called beam

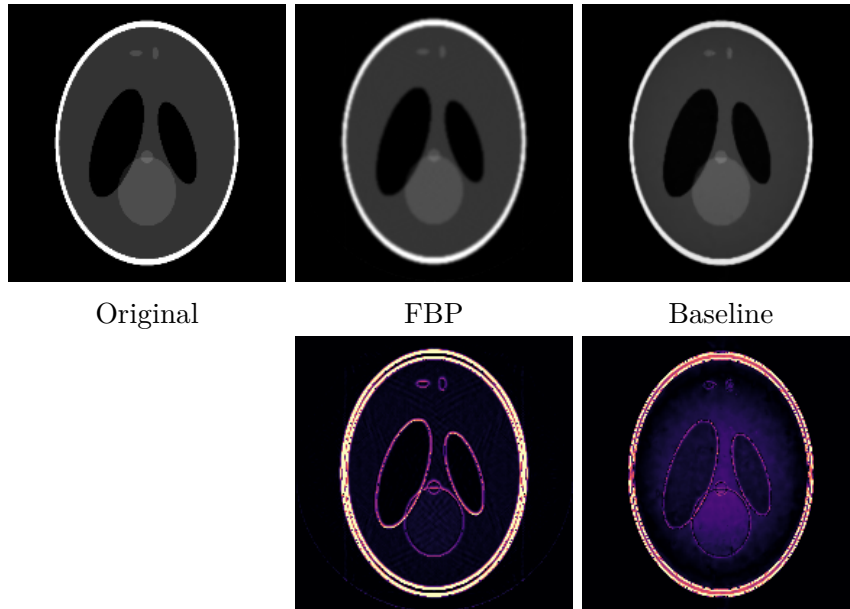


Figure 5.1: **Evaluating Shepp-Logan reconstructions using FBP and FlexAF.** Top row: reconstructed slices. Bottom row: Absolute difference images between the reconstructions and the original phantom. All difference images were windowed to $[0, 0.15]$ before color mapping in order to better visualize the error.

hardening. Though the Shepp-Logan phantom does have a dense outer surface, our simulated X-ray projections do not incorporate any polychromatic effects. This implies some subtle error in our training method which is in some way overemphasizing this central region.

Our baseline FlexAF configuration slightly edges out the constant intervals configuration, though the effect is admittedly quite small. Both configurations show extremely similar metrics and reconstructed slices. The baseline configuration demonstrates a slight performance advantage, finishing the 200 epochs approximately half an hour before the constant intervals configuration.

For our entropy pixels configuration, we used a kernel radius of 4, entropy threshold of 0.3, and a tree depth of 8, producing a 38.35% reduction in the number of training samples per epoch and a 44.4% reduction in overall runtime. Quantitatively, this configuration is comparable to the baseline and constant intervals reconstructions.

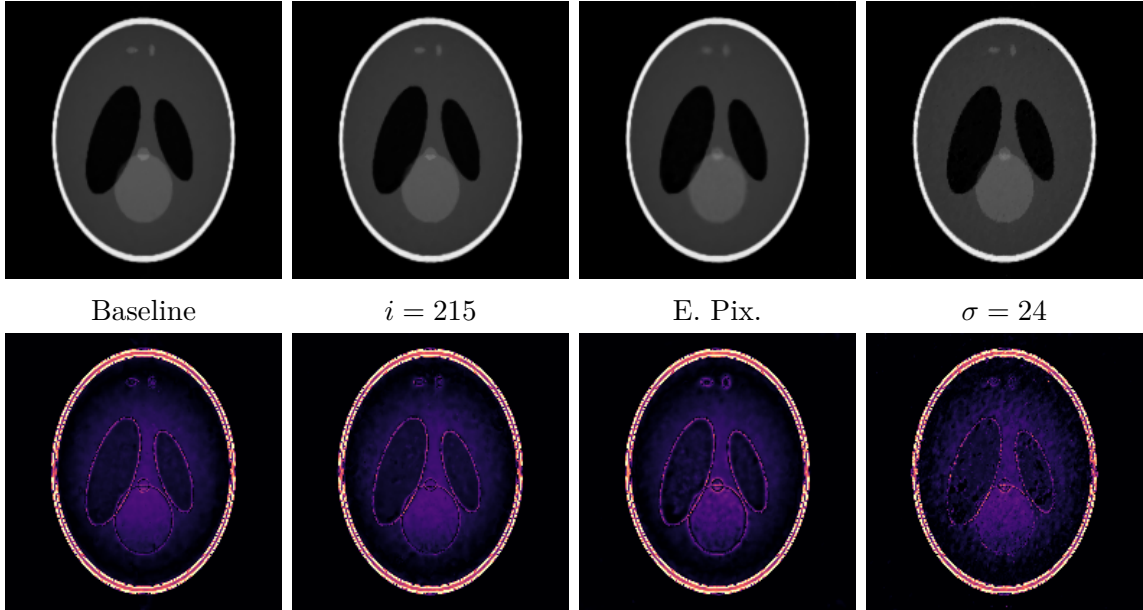


Figure 5.2: **Comparing FlexAF reconstructions across various model configurations.** Top row: reconstructed slices. Bottom row: Absolute difference images between the reconstructions and the original phantom. All difference images were windowed to $[0, 0.15]$ before color mapping in order to better visualize the error.

However, visual inspection of the slices shows a lack of definition around some of the smaller, low-contrast ellipses. This produces edge-effect errors for these features which are very similar to those errors seen in the FBP reconstruction.

Likewise, the quantitative difference between the baseline and $\sigma = 24$ configurations is very small, but the visual differences favor the baseline method. The increased Gaussian scale improves the definition around the edges of the ellipses and produces what appears to be the lowest edge error across all methods. However, it also amplifies a noise pattern which is only subtly visible in the other FlexAF reconstructions. As a result, the interiors of the ellipses appear mottled and do not demonstrate the expected uniformity. This is in line with the findings discussed in 2.2.2 that increasing the Gaussian scale can produce poor model generalization.

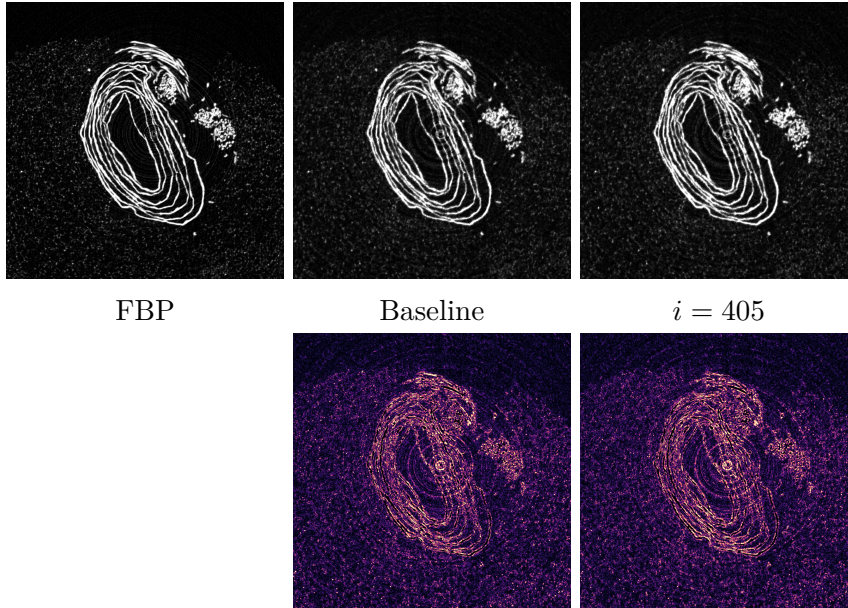


Figure 5.3: **Comparison of FlexAF reconstructions for the papyrus scroll dataset.** Top row: reconstructed slices. Bottom row: Absolute difference images between the reconstructions and the FBP reconstruction. All difference images were windowed to $[0, 0.3]$ before color mapping in order to better visualize the error.

5.1.2 Papyrus scroll

The papyrus scroll dataset is significantly larger than the Shepp-Logan phantom and contains 1361x the number of projection pixels. Training over the full dataset would require many days in order to evaluate even a single training epoch and thus is a practical impossibility. We make a number of reductions to the size in order to evaluate FlexAF on this dataset. First, we train over a “short scan” of 1073 projection images in the range $[0^\circ, 214.6^\circ)$, which approximately represents 180° plus two times the cone angle [68]. Second, we crop the projections to only the center-most rows and limit our evaluation to the slices which lie within this region.

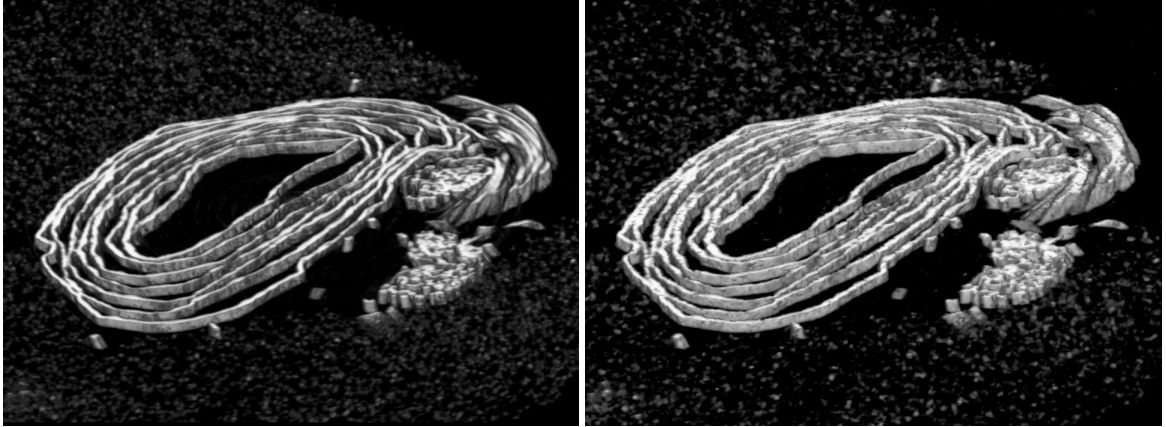
Our baseline FlexAF method is trained over the two central projection rows for 300 epochs, and we compute our evaluation metrics on the slice which lies between these two rows at $z = 0$. It is configured with an 8-layer MLP of width 256 and a Gaussian encoder with scale of $\sigma = 105$ and 384 features. We manually set the diameter of the

bounding cylinder to 59.04 mm, which is slightly larger than the width of the field of view. The interval scheduler is configured with a multiplier $m = 8$ and reaches a maximum of 405 ray intervals after 17 epochs. On data load, we apply a manual post-alignment of -30 pixels to our X-ray cameras to match the post-alignment used by FBP.

We experiment with two variations to the baseline configuration. The first disables the interval schedule and uses a constant 405 ray intervals. Like the baseline method, it is trained on two projection rows for 300 epochs and is evaluated on a single slice. The second is configured to reconstruct multiple slices from 12 rows of projection images. To account for the increased volume size, we adjust the width of the MLP to 334 and set the interval scheduler to a maximum of 512 intervals. We also enable entropy pixels to reduce the number of total training samples by 60.8%. Even still, the time-per-epoch is approximately six times that of the single slice configurations, thus we evaluate our metrics on the 12 slices after only 50 epochs.

All of our FlexAF configurations produce similar quantitative results, with the baseline method outperforming the fixed interval configuration in PSNR, SSIM, and training time. Despite comparison metrics which are overall quite low, the single slice FlexAF reconstructions are very similar in appearance to the FBP reconstruction (Figure 5.3). The wraps of the papyrus are generally well-defined and sharp, but FlexAF struggles to resolve the low contrast, irregular noise pattern of the polyethylene foam. As with the Shepp-Logan phantom, most of the error appears to be located around the boundaries of objects and in the background noise patterns. This is likely a significant contributor to our low comparison metrics as this dataset is mostly defined in terms of edges and noise patterns.

The 12 slice configuration scores the lowest across all metrics for the FlexAF configurations, though it's difficult to evaluate whether this is due to the many fewer training epochs or a fundamental capacity limit of the given configuration. We do



(a) FBP

(b) FlexAF

Figure 5.4: **Volume renderings of 12 slices from the papyrus scroll comparing FBP to FlexAF.** (a) The filtered backprojection (FBP) reconstruction. (b) The FlexAF reconstruction. Though FlexAF has some difficulty capturing the highest-resolution features of the polyethylene foam, the papyrus scroll’s structure is extremely close to that produced by FBP.

not note any significant falloff in visual quality, and the 3D structure in these 12 slices is of passable similarity to that of the FBP reconstruction (Figure 5.4).

5.1.3 Multi* proxy

As with the papyrus scroll, the size of the Multi* datasets makes evaluation at the highest resolutions a practical impossibility. Thus, our Multi* evaluations are performed on only two of the multi-resolution scans, MS.01.01 (140 μm) and MS.01.02 (70 μm). For both MS.01.01 and MS.01.02, we train over all but the final (repeated) projection image.

MS.01.01

The baseline configuration for MS.01.01 uses a 7-layer MLP with layer width 256 and a Gaussian encoder with scale $\sigma = 70$ and 384 features. The interval scheduler is configured with a multiplier $m = 8$ and reaches a maximum of 512 ray intervals after 20 epochs. A variant of this configuration disables the interval scheduler and uses a constant 512 intervals throughout training. On data load, we apply a manual post-alignment of -1.5 pixels to our X-ray cameras to match the post-alignment used by

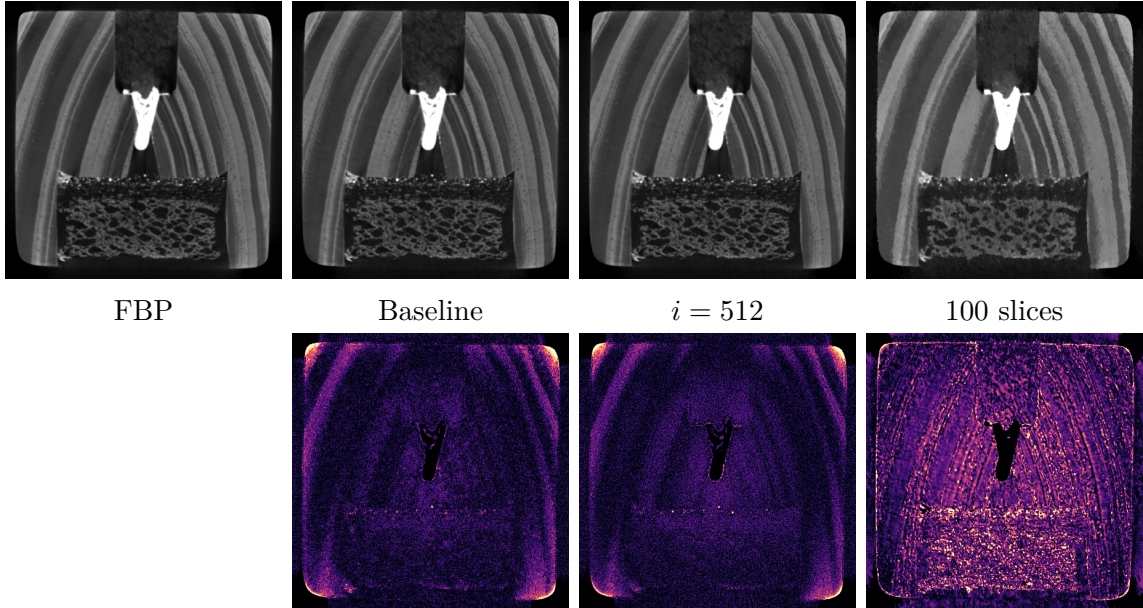
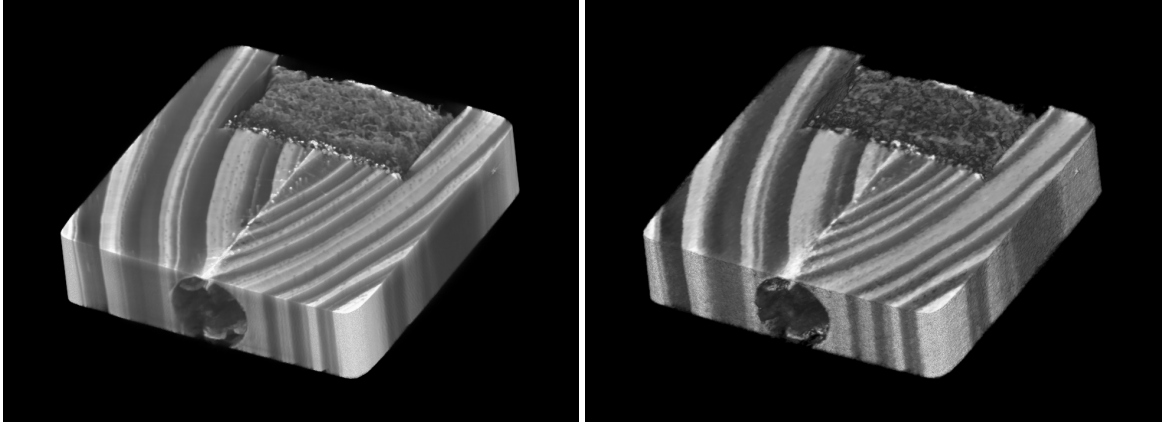


Figure 5.5: **Comparison of FlexAF reconstructions for the MS.01.01 dataset.** Top row: reconstructed slices. Bottom row: Absolute difference images between the reconstructions and the FBP reconstruction. The first two difference images were windowed to $[0, 0.1]$ and the final image was windowed to $[0, 0.2]$ before color mapping in order to better visualize the error. Enlarged versions of the FBP and baseline slices are available in Figure A.1.

FBP. We train both of these configurations on the two center rows of the projection images for 1000 epochs and evaluate the comparison metrics on a single slice at the plane $z = 0$.

The FlexAF reconstructions of MS.01.01 are some of the most accurate we observe in this study. Both configurations have a PSNR which is 7 dB higher than any other FlexAF reconstruction, and the SSIM and NMI metrics are among the highest we record for real-world datasets. As shown in Figure 5.5, this translates into rendered slices which are nearly identical to those produced by FBP.

Across both of these configurations, we again observe that there is an intensity gradient between the center and outer edges of the reconstruction in the difference images. Though less pronounced than that seen in the Shepp-Logan experiments,



(a) FBP

(b) FlexAF

Figure 5.6: **Volume renderings of 100 slices from MS.01.01 comparing FBP to FlexAF.** (a) The FBP reconstruction. (b) The FlexAF reconstruction. While the global structure of the wood block appears accurate in this rendering, closeups of the slices demonstrate that many of the high-frequency details are missing from the reconstruction (Figure 5.5).

the center of the reconstruction appears to be brighter than the outer edge, resulting in the largest error at the corners of the block. Unlike the Shepp-Logan results, we note that the FlexAF attenuation coefficients in these areas appear to be more consistent with those found in the rest of the wood block, thus we hypothesize that this measurement error represents FlexAF correcting for hardening artifacts found in the FBP reconstruction. Much of the remaining error is attributable to low intensity background noise.

Given the successful reconstruction of a single slice using the baseline configuration, we also test how our neural volume’s capacity is affected when growing the size of the volume to 100 slices (1.4 cm). We increase the network capacity from the baseline configuration by using an 8-layer MLP with layer width of 334 and enable entropy pixels to reduce the size of the training set by 32.8%. We train on the center 100 rows of the projection image for 34 epochs and evaluate the metrics across all 100 slices.

Though the metrics for this configuration are similar to or exceed those for the papyrus scroll, the qualitative assessment of the reconstruction tells a different story.

The model roughly captures the larger structures in the sample (i.e. the block, the silicone, the sponges), but completely fails to resolve any of the finely detailed features of the volume. Unlike a low resolution reconstruction where one expects edge features to lack clarity, here the edge features are well-defined but jagged and inaccurate. The effect is visually akin to what one sees when an image has been quantized with too few bits. Given the high quality result of our baseline configuration, it is likely that we have exceeded the MLP’s volumetric capacity. Though we have almost doubled the number of model parameters from the baseline configuration, we have also increased the size of our reconstructable volume by a factor of 100. The result is poor memorization of the dataset’s high frequency features. Nevertheless, viewing the reconstruction in 3D demonstrates that it is still globally consistent with the FBP reconstruction (Figure 5.6).

MS.01.02

For the MS.01.02 baseline configuration, we make two changes to the configuration used for MS.01.01 in order to account for the increased scan resolution. Rather than doubling the scale (and feature sparsity) of our Gaussian encoding, we instead halve the diameter of the bounding cylinder from 167.27 mm to 83.63 mm, which is just larger than the diameter of the Multi* proxy. Second, we increase the Gaussian scale to $\sigma = 90$ to account for high frequency features which were not observable in the 140 μm scan.

As before, we apply manual post-alignment of -3.5 pixels to our X-ray cameras to match the post-alignment used by FBP. We vary this configuration by disabling the interval scheduler and using a constant 512 intervals throughout training. We train both of these configurations on the two center rows of the projection images for 450 epochs and evaluate the comparison metrics on a single slice at the plane $z = 0$. Since the new bounding cylinder falls within the field of view of the projection images, we also crop the projections to the central 1192 columns which fall within the cylinder’s

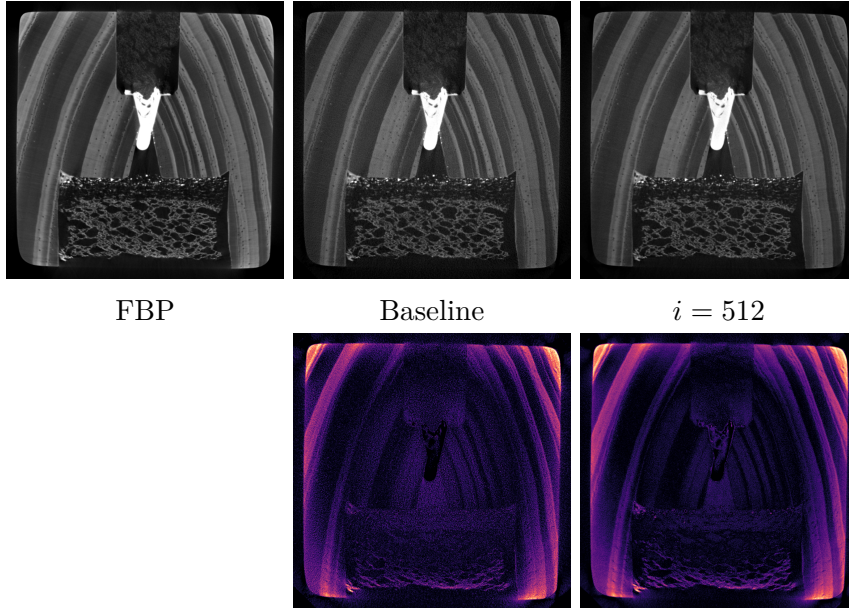


Figure 5.7: **Comparison of FlexAF reconstructions for the MS.01.02 dataset.** Top row: reconstructed slices. Bottom row: Absolute difference images between the reconstructions and the FBP reconstruction. All difference images were windowed to $[0, 0.3]$ before color mapping in order to better visualize the error. Enlarged versions of the FBP and $i = 512$ slices are available in Figure A.2.

bounds, further reducing the size of the training set by 825,600 samples.

Both configurations perform extremely well and again produce reconstructions which are visually very similar to those of FBP. Interestingly, the fixed interval configuration quantitatively outperforms the baseline configuration and provides the single highest SSIM score across all real-world datasets. Through visual inspection, we can see that the fixed interval configuration does a better job at reconstructing the smooth intensities of the wood grain and has less high-frequency noise than the baseline configuration (Figure 5.7). Both configurations demonstrate the intensity gradient where the center of the reconstruction is brighter than the edges, producing greater error at the corners of the block.

5.2 Automatic extrinsic calibration

We evaluate our automatic camera extrinsic calibration method on the papyrus scroll dataset due to its extreme misalignment. Since our calibration method ultimately needs to be effective across a range of resolutions, we validate the Gaussian frequency filter on both the full resolution dataset and a 1:4 scaled version. To construct the 1:4 scale dataset, we scale the dataset down rotationally and spatially. Rotationally, we skip every 7 rotational angles to load a total of 600 projection images. Spatially, we scale each projection image by 0.25x and crop to the central 12 rows of pixels. The 12 rows of pixels are intended to provide a region of support to the calibration task; with only one or two rows, there may not be enough structure in the projections to estimate good calibrations. The final dataset size is $560 \times 12 \times 600$. We also enable entropy pixels to further reduce the samples per epoch by 40.55%.

Using the 1:4 dataset, we evaluate FlexAF’s ability to learn the camera extrinsics both with and without the Gaussian frequency filter. For both configurations, we modify the baseline configuration from 5.1.2 by lowering the Gaussian scale to $\sigma = 26.25$ to account for the lower resolution. We have observed that the FlexAF model requires a few training epochs before the volume begins to converge on a discernible reconstruction. To avoid updating the extrinsics using wholly incorrect volume features, we wait some epochs before allowing the extrinsics to update. For the filter-enabled configuration, we train the detector extrinsics between epochs 4 and 30, and we set our frequency filter to linearly activate all frequencies over the first 21 training epochs. For the sans-filter configuration, we train both the source and detector extrinsics starting on the 4th epoch. Disabling the Gaussian filter tends to make extrinsic learning much less stable. To avoid catastrophic failure, we lower the learning rates for both the model and extrinsic optimizers in comparison to the filter-enabled configuration. Since this configuration now learns at a much slower rate, we do not set an epoch limit on extrinsic updates.

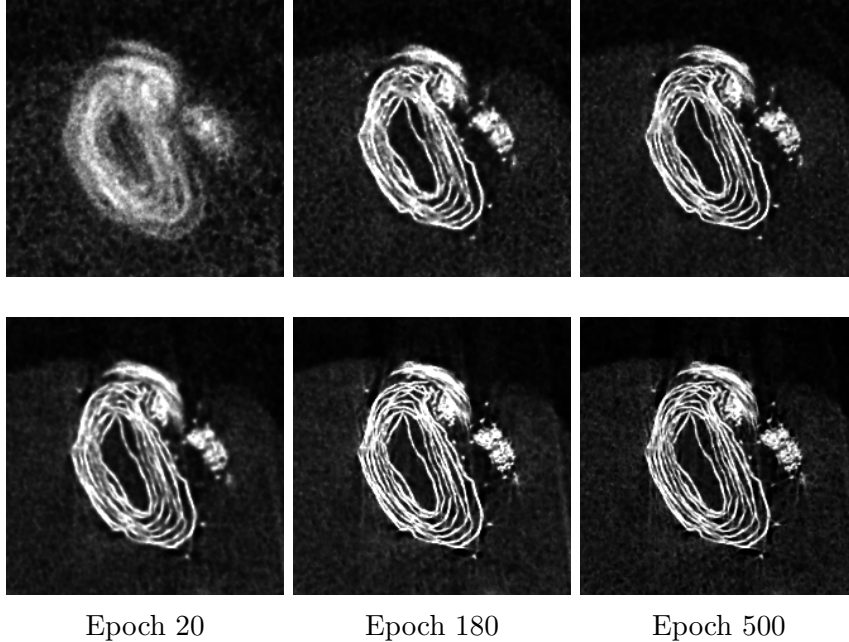


Figure 5.8: **Testing the Gaussian frequency filter for automatic extrinsic calibration using a 1:4 scale papyrus scroll dataset.** This figure depicts the effect of the Gaussian frequency filter on automatic extrinsic calibration when applied to the 1:4 scale papyrus scroll dataset. Training epochs increase from left to right. Top: FlexAF results with the frequency filter disabled. Bottom: FlexAF results with the frequency filter enabled. Training without the filter requires almost 6x more training iterations to reach a quality comparable to the filter-enabled configuration.

Figure 5.8 shows slices rendered from our two FlexAF configurations after 20, 180, and 500 training epochs. Both reconstructions are of acceptable quality given the scale and accurately capture the structure of the scroll and the blurred structure of the foam. Notably, the configuration without the frequency filter does learn the correct extrinsic calibrations and eventually converges to a reconstruction of similar quality as that of the filter-enabled configuration. However, this configuration is only stable because of its lower learning rates. In contrast, the model with the Gaussian frequency filter converges over many fewer iterations and produces a reasonable reconstruction after only 20 epochs.

We next test our automatic calibration method on the full resolution papyrus scroll dataset using the “short scan” of only 1073 projections. As in the previous experiment,

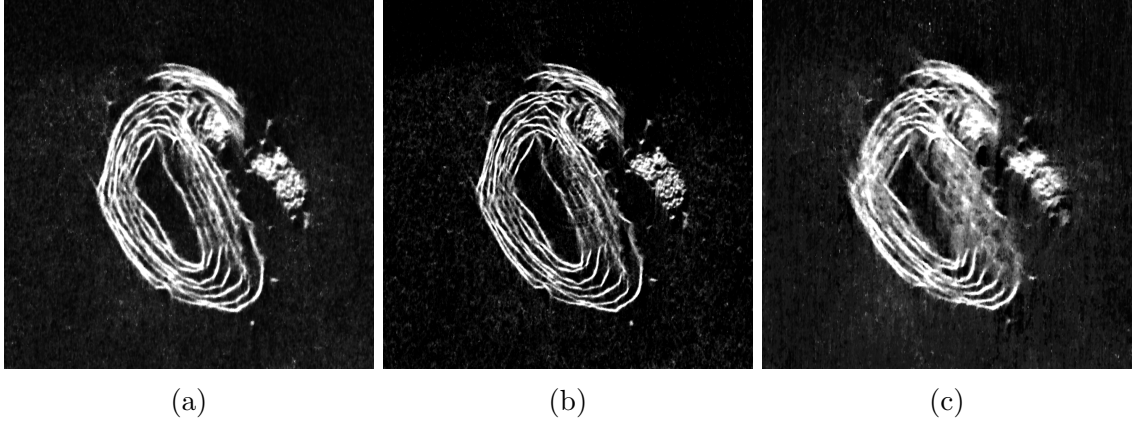


Figure 5.9: **Automatic extrinsic calibration of the papyrus scroll dataset at full resolution.** (a) Trained for 20 epochs. (b) Trained for 60 epochs. (c) Trained for 20 epochs with non-optimal hyperparameters.

we crop each projection to the 12 central rows of pixels to provide a region of support for extrinsic calibration and employ entropy pixels to reduce the samples per epoch. Empirically we observe that the model begins to converge on the full resolution dataset after only two epochs, so we train both the detector and source extrinsics between epochs 2 and 30, and we set our frequency filter to linearly activate all frequencies over the first 21 training epochs.

As in the low resolution experiment, the model converges quickly on both the extrinsics and the reconstruction, and the result is of reasonable quality after only 20 training epochs (Figure 5.9a). However, the extrinsic calibration is not perfect. Many of the interior wraps of the scroll are distinct but blurred, and a few of the smaller point-like features outside the scroll show the crescent-shape of misalignment. This residual misalignment does not improve as training proceeds. After 60 epochs (Figure 5.9b), extrinsic training has been disabled and the model has begun to alter the volume to best explain the residual error. The interior wraps of the scroll are even less well-defined than before, and the crescents have sharpened into distinct curves. When we compare this result to our baseline with manual post-alignment applied (Figure 5.3), it is easy to see that the calibration is almost but not quite correct.

This result highlights one of the difficulties in jointly learning X-ray camera calibration alongside reconstruction. Ideally, the relative convergence of the volume and the extrinsics should proceed in tandem, but our controls over this behavior — the learning rates, the time points at which we start and stop learning, the rate at which we adjust our Gaussian filter — are indirect, inexact, and numerous. Given this result, should the calibration’s learning rate be increased or decreased, or did calibration learning start too soon or too late? It is difficult to answer any of these questions from the hyperparameters alone, and seemingly small changes to the configuration can lead to outsized effects on the reconstruction.

To illustrate this point, consider the reconstruction for the alternative configuration shown in Figure 5.9c. This result was produced by a hyperparameter sweep that we ran to tune our configuration for extrinsic calibration. The given configuration varies only slightly from the one used in our experiment: calibration learning begins after the first epoch, the Gaussian frequency filter reaches its maximum after 14 epochs, and the calibration learning rate is increased by 2.18×10^{-4} . The reconstruction for this trial shows obvious signs of misalignment, yet it is unclear which of these hyperparameter adjustments (or perhaps all of them) led to such a dramatic difference in results.

Ultimately, we are heartened by the results of our extrinsic calibration experiment. In a relatively short amount of training time, FlexAF has converged on reasonable approximations of the extrinsics and the reconstruction, and our calibration method works across both low and high resolutions. This result was not a foregone conclusion when one considers the dramatic misalignment of the input dataset. We expect that perfect calibration of the full resolution dataset is attainable with the current method and is only a matter of finding the right combination of hyperparameters. The easy success of the low resolution method suggests that a hierarchical training approach may provide some means for improving this process.

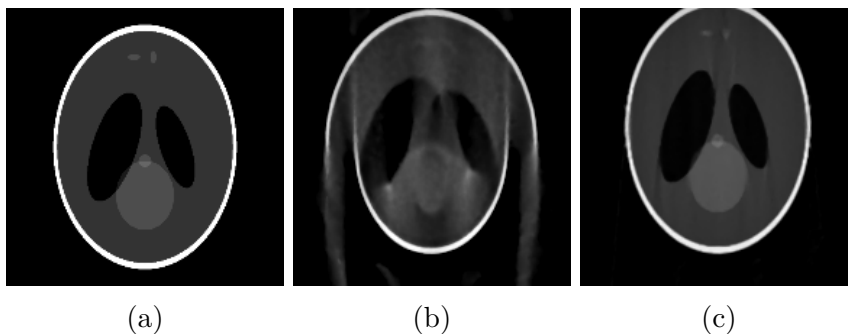


Figure 5.10: **Automatic extrinsic calibration parallel projection images using the Shepp-Logan phantom.** Reconstruction results for induced misalignment in the Shepp-Logan camera positions. (a) Ground truth phantom. (b) With calibration disabled. (c) With automatic calibration. The FlexAF calibration does correct for the induced misalignment and generates a structurally coherent reconstruction, however the scene is scaled and shifted relative to the world origin.

5.2.1 Calibration of parallel geometries

As discussed in 2.3.4, parallel beam X-ray cameras are modeled with orthographic projection and thus lack many of the perspective effects we might encounter when using a cone beam source. It is not immediately obvious whether our automatic calibration method should be expected to work for parallel beam geometries. To answer this question, we apply automatic calibration to a virtually misaligned version of the Shepp-Logan phantom. Creating the misaligned phantom is simple. We generate the phantom and projections as previously described, but we use the manual post-alignment functionality to shift the X-ray cameras so that their initial positions are horizontally offset from their correct locations.

For our experiment, we use the baseline Shepp-Logan configuration from 5.1.1, but augment it with automatic calibration settings similar to those used on the papyrus scroll phantom. To provide a comparison, we also reconstruct the misaligned Shepp-Logan phantom without automatic calibration. For both trials, we apply manual misalignment of -10 pixels to the Shepp-Logan X-ray cameras (0.5 mm in our virtual coordinate system). The resulting reconstructions are shown in Figure 5.10.

Our method does learn a calibration which accurately recovers the structure of the Shepp-Logan phantom. However, the calibrated reconstruction is shifted in the world coordinate frame and slightly larger than the original phantom. This result is promising in that it demonstrates the applicability of our calibration method to parallel beam X-ray cameras, but it also highlights a weakness in our current implementation. Our camera calibrations are modeled as 4×4 homogeneous transform matrices, and as such, they are capable of *scaling* the X-ray source and pixel positions with respect to the world coordinate frame. This experiment shows that a better solution would be to learn only a 3D translation and one rotation angle for each axis (i.e. roll, pitch, and yaw).

5.3 Multi-resolution reconstruction

The continuous coordinate system of our neural volume implies the ability to learn CT reconstructions with multiple levels of detail. Intuitively, volume regions that have only been trained on low resolution or sparse X-ray projections should be of low spatial resolution, while regions trained on high resolution or dense X-ray projections should be of high spatial resolution, and regions of overlap should have a spatial resolution somewhere in between. We experiment with FlexAF on a multi-resolution region of interest (ROI) reconstruction task to validate its ability to model different regions of a volume captured at various scales.

5.3.1 Combining regions of interest

Our experiment imagines a scenario where we would like to selectively improve upon the quality of the natural sponge from our MS.01.01 baseline experiment. The natural sponge is the lowest density material in the Multi* proxy and has an extremely fine cellular structure. At the $140 \mu\text{m}$ resolution of the MS.01.01 scan, the largest features of this structure are visible but lack any sort of clarity or sharpness (Figure A.1). At $70 \mu\text{m}$, however, we can begin to differentiate the cellular structure and sharp interior edges (Figure A.2). Our experimental goal is to reconstruct most of the Multi* proxy

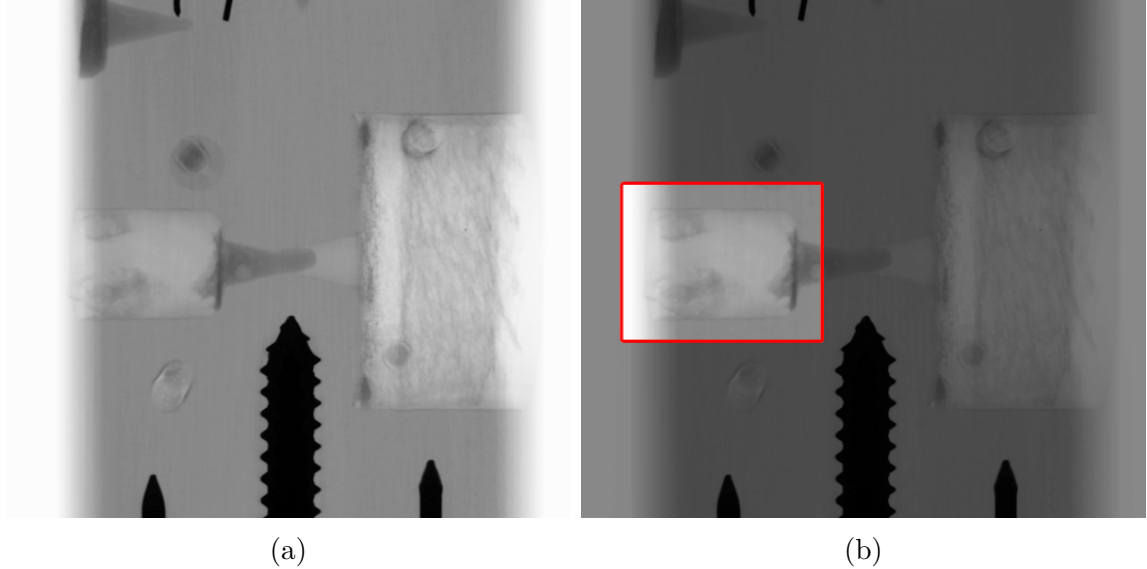


Figure 5.11: **Masking CT projection images to construct a reconstruction region of interest.** (a) A projection image from the Multi* dataset. (b) The masked ROI for the natural sponge. This mask tracks the sponge across all rotational projections.

at a $140\ \mu\text{m}$ resolution while selectively reconstructing the natural sponge ROI at a $70\ \mu\text{m}$ resolution.

We use three Multi* datasets for this study: MS.01.01 ($140\ \mu\text{m}$), MS.01.02 ($70\ \mu\text{m}$), and a scaled version of MS.01.04 ($20\ \mu\text{m}$). We include this latter scan because MS.01.01 is essentially just a binned down version of MS.01.02 and follows the same scan trajectory relative to the sample. In real world applications, it is highly likely that an ROI scan would vary the source-to-sample distance in order to maximize reconstruction quality. MS.01.04 does have a different source-to-sample distance and thus has a distinct scanning trajectory from MS.01.01. On data load, we scale the projections of MS.01.04 by $0.2857x$ and load every two projection images to achieve an approximate scan resolution of $70\ \mu\text{m}$.

Since the Multi* dataset does not contain a true ROI scan, we construct one by masking our high-resolution projections to only those regions containing the natural sponge. The natural sponge lies outside the center-of-rotation, thus we precompute a per-projection rectangular mask which tracks the motion of the sponge within the

projection images (Figure 5.11). This off-center moving mask poses no difficulty for FlexAF as we train on individual image pixels which store their individual extrinsic information. On data load, we initialize the X-ray cameras relative to the entire image, but only store the tensors for those pixels which lie within the bounds of the mask. This is similar to the cropping technique we already employ to improve reconstruction times.

5.3.2 ROI reconstruction

We test FlexAF by reconstructing with three different combinations of input datasets. First, we reconstruct an ROI-only dataset using only the masked region of MS.01.02. Second, we reconstruct a “same trajectory” dataset composed of the full width MS.01.01 projections and the masked MS.01.02 projections. Finally, we reconstruct an “alternate trajectory” dataset composed of the full width MS.01.01 projections and the masked MS.01.04 projections. The FlexAF configuration is identical across all reconstructions and is a combination of the baseline configurations in 5.1.3. We use the model and encoding parameters from the MS.01.01 baseline but apply the contracted bounding cylinder from the MS.01.02 baseline. For all reconstructions, we train on only the center most rows and evaluate on a single slice at the center of the volume. The reconstructed results are shown in Figure 5.12.

The ROI reconstruction looks largely as one might hope and expect. The cylindrical region surrounding the sponge has the same accuracy we saw in the MS.01.02 baseline experiment (Figure 5.7), while the areas outside this cylinder are of significantly degraded quality. This confirms our ability to independently reconstruct isolated subregions without a loss of quality within the region of interest.

The two multi-resolution configurations produce very similar results and, as expected, present reconstructions that combine the qualities of the low and high resolution datasets. The fine structures of the sponge which are so blurred in the 140 μm reconstruction are now distinguishable, though not quite to the same clarity as that

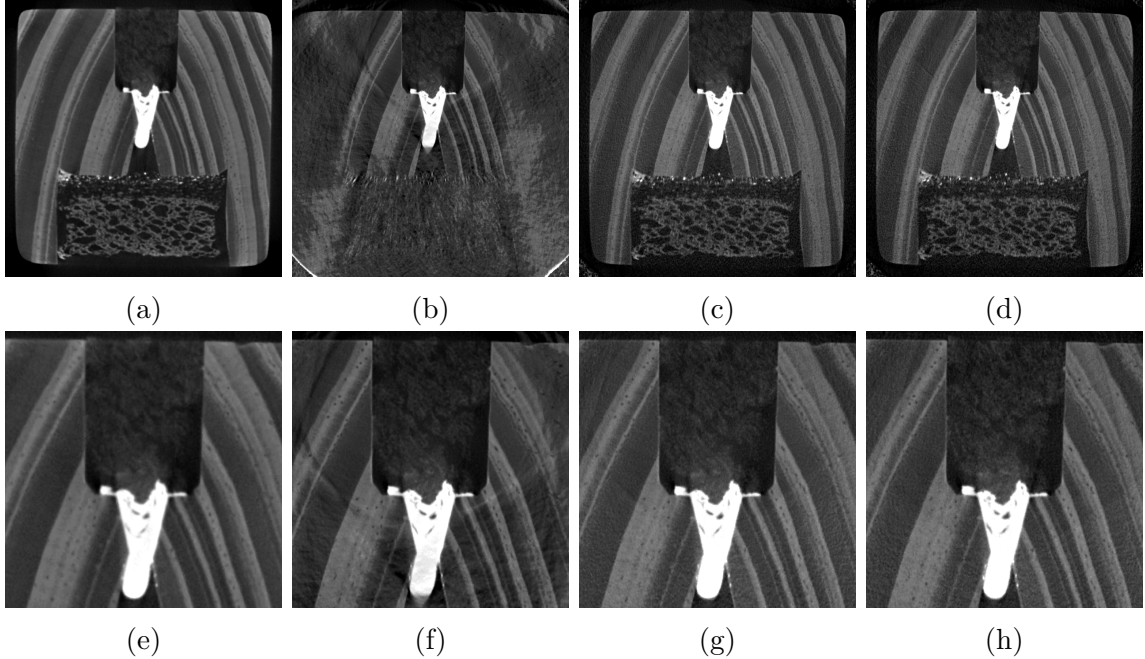


Figure 5.12: **Resulting reconstructions from the multi-resolution ROI experiments.** (a) Full FlexAF reconstruction of MS.01.01 ($140\mu\text{m}$). (b) ROI-only reconstruction. (c) ROI-enhancement reconstruction using projections from MS.01.01 and MS.01.02. (d) ROI-enhancement reconstruction using projections MS.01.01 and MS.01.04. (e–h) ROI crops of the same. Enlarged versions of the full slices are available in Figure A.3.

observed in the $70\mu\text{m}$ reconstruction. Likewise, many of the wood pores that fall within the ROI cylinder are also much sharper than their low resolution counterparts. The similarity between the two multi-resolution reconstructions is striking, and is a promising indicator for our desired ability to combine projections captured along drastically different scan trajectories.

These reconstructions do, however, show some artifacts that result from training on the ROI region. First, both multi-resolution reconstructions are generally much noisier than the baseline and ROI-only reconstructions. This noise is persistent throughout the entire sample area, but is most noticeable in the low resolution regions near the bottom of the slice view. A possible cause for the noise is that we are sampling from datasets with different bandwidths but encoding all coordinates with the same Gaussian features. We theorize that this produces aliasing in the volume’s

frequency domain which presents as noise in the reconstructions. We do not see the same noise in our baseline experiments or the ROI-only reconstruction because those results were trained on data of the same bandwidth.

The second artifact is a faint but distinct line that separates the ROI region from the rest of the volume. This line is also visible in the ROI-only reconstruction, where the distinction between the inner and outer regions is much more prominent. Including the low resolution data appears to have reduced this artifact but has not eliminated it. It is unclear from this experiment whether this is generally a problem for ROI reconstruction or a beam hardening artifact caused by the bright silicone in the core of the Multi* phantom.

5.4 Multi-energy reconstruction

We evaluate our multi-energy volume model using projections drawn from five of our Multi* datasets: MS.02.01 (35 kV), MS.02.03 (50 kV, Al 0.5mm filter), MS.02.04 (70 kV, Al 0.5mm filter), MS.02.05 (90 kV, Al 1mm filter), and MS.02.06 (120 kV, Cu 0.5mm filter). For convenience and clarity, we will refer to these datasets by their peak incident energies.

Our goal in this experiment is to generate a single reconstructed volume which captures the spectral information of all five scans. Crucially, we also want to avoid the theoretical X-ray dosage and capture times that would come from acquiring five complete CT scans in a real-world environment. Thus, we limit ourselves to training on a maximum of 1200 projection images, the same number of projections required to reconstruct any one of the five Multi* scans.

We consider two methods to combine our datasets into a single multi-energy scan (Figure 5.13). In the first, we *interleave* the incident energies from low to high after every rotational step such that the projections follow the sequence: *35, 50, 70, 90, 120, 35, 50, 70, ...* (Figure 5.13a). In this method, each individual scan is sparsely sampled at $1/5$ the rate of the full scan across the 360° range. Our second method

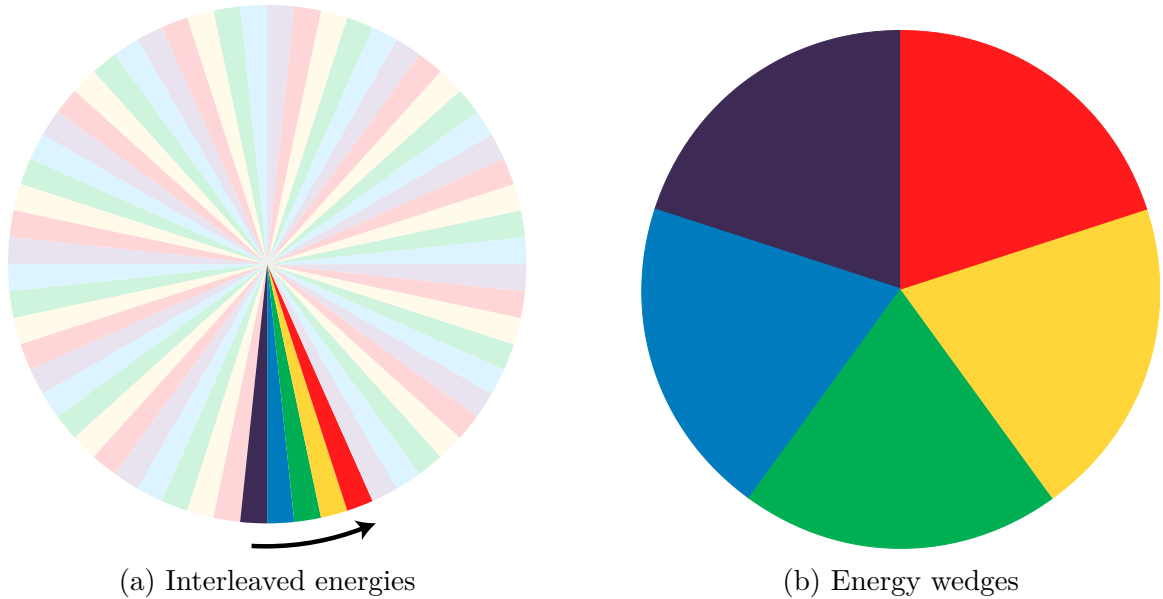


Figure 5.13: **Visualization of the dataset combinations for the multi-energy experiments.** (a) The incident energies are interleaved across projection images such that the energy changes between every rotational step. (b) The total rotational range is divided into five equal-sized wedges, one for each incident energy.

divides the 360° range into five equal-sized *wedges* which contain the projections from only one energy (Figure 5.13b). In this method, each individual scan is a limited angle dataset that covers only a 72° rotational range.

We evaluate both of these datasets with the baseline FlexAF configuration used for MS.01.01. The one variation we make is to use the multi-energy volume model described in 3.3.3. Though the full MS.02 scans contain many interesting spectral features within their fields of view (sponges, metals, silicone, etc.), most of these features are near the top and bottom of the volume. The cone beam geometries are such that reconstruction of even a single slice from these regions would require training over many rows of pixels from each projection to ensure sufficient sampling. As in our other experiments, we train over the central two rows of pixels and render on the slice plane at $z = 0$.

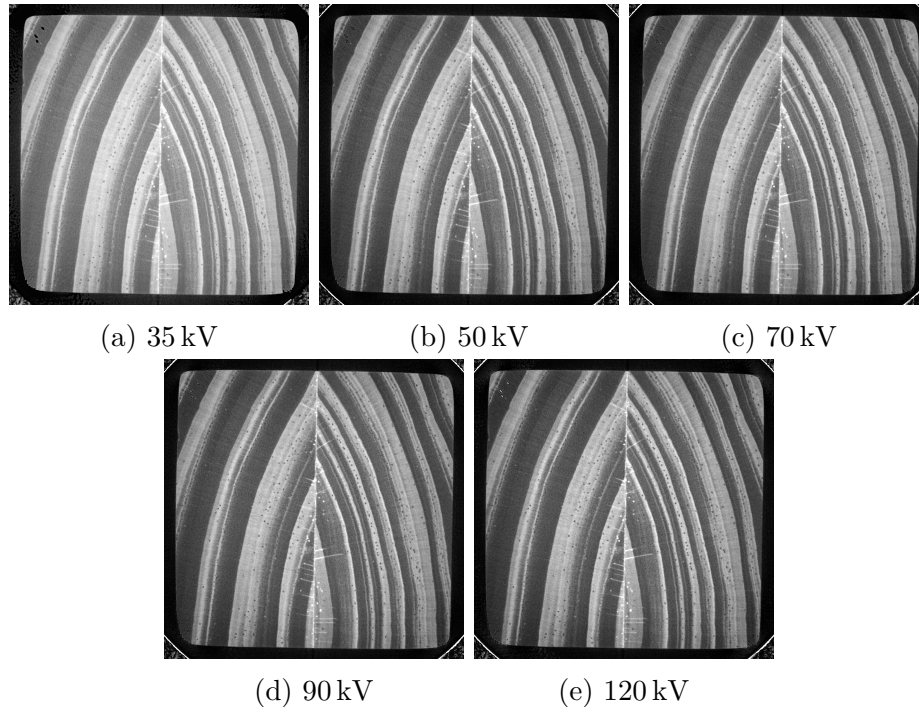


Figure 5.14: **Reconstructions for the five incident energies in the interleaved multi-energy experiment.** Slices rendered across the five training energies. Though the FlexAF framework was provided only sparse projection sets for each individual energy, the model has combined the shared information across energies to reconstruct a spatially accurate volume.

5.4.1 Interleaved energies

The reconstruction results for the interleaved multi-energy dataset are shown in Figure 5.14. Since our multi-energy volume is parameterized by both 3D coordinate and incident energy, we generate slices for each of our five training energies. Somewhat disappointingly, this slice only shows the wood grain of the Multi* base block. However, we can see that the spatial quality of the reconstruction compares favorably to that of the 70 μm MS.01.02 results. This is most noticeable in the clarity of the wood pores which are distributed in multiple places around the sample.

We do note a few places in which our multi-energy model fails to produce an accurate reconstruction. In the top left corner of the 35 kV slice, there are three “holes” in the reconstruction which should not be there. Looking at the same region across all of our rendered slices, we can see that the nature of these holes changes with

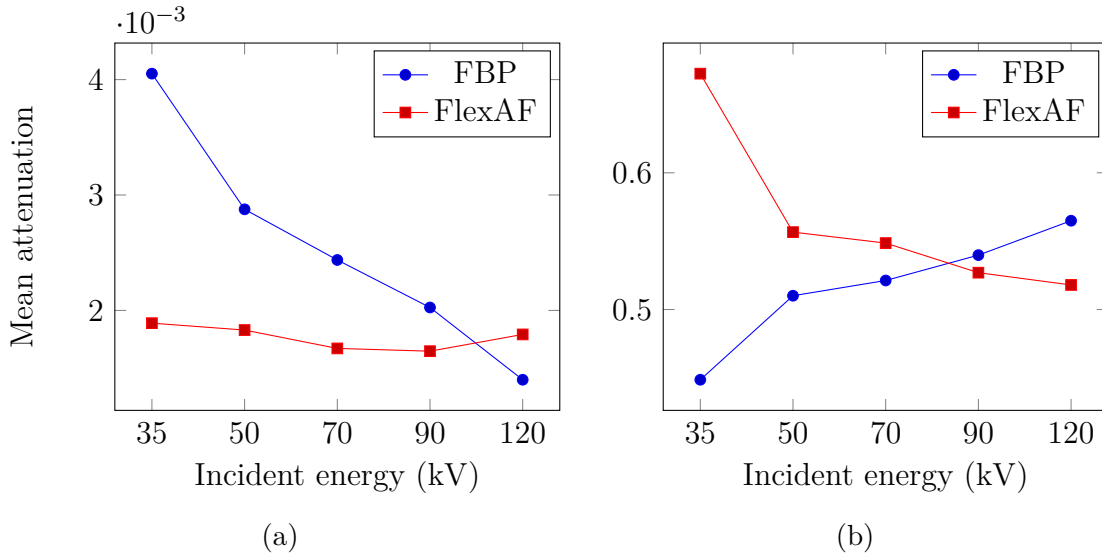


Figure 5.15: **Comparing the reconstructed attenuation coefficients produced by FBP and FlexAF for the interleaved multi-energy experiments.** Plots of the mean attenuation coefficients with respect to peak incident energy for a centered 600×600 slice ROI. (a) Comparison of the raw attenuation coefficients for the FBP and FlexAF reconstructions. (b) Comparison of the normalized attenuation coefficients for the same. As evidenced by the change in attenuation across incident energy, we have successfully separated attenuation from structural composition as two independent components in our reconstruction. However, the FlexAF attenuation coefficients appear to follow an inverse trend from those produced by FBP, implying that our polynomial mapping model requires further development.

the incident energy. In the other reconstructions, the holes are somewhat filled in but outlined in black. As the energy increases, so too does the brightness of the filled region inside the holes, yet the black outline remains. Likewise, the corners of the proxy are not continuous and are interrupted with “holes” of a similar appearance.

Evaluating the spectral accuracy of our model is not straightforward. There is no reason to believe that our reconstructed attenuation coefficients should fall into the same range as those returned by FBP. What we can say is that the relative coefficients across energy should follow the same trend: if the 50 kV slice is brighter than the 35 kV slice in FBP, the same should be true for our multi-energy reconstructions.

To this end, we plot the mean attenuation across incident energy between the FBP

and FlexAF reconstructions (Figure 5.15). To view the same values on a relative rather than absolute scale, we also plot the mean attenuation after normalizing the slices within the range of their own set. We specifically avoid including our reconstruction anomalies in this measure by calculating the mean attenuation from a 600×600 subregion centered on the middle of the slice. These plots show that our method does not accurately reconstruct the relative attenuation across energy. Instead, it appears as though the dynamic range of our raw attenuation coefficients is extremely narrow. When normalized, we also see that the relative intensities are almost exactly the inverse of those from FBP.

The exact reason for this error is currently unclear. Most likely is that our multi-energy volume and image formation method do not accurately model the effects of incident energy on the projection images. Our volume assumes a monochromatic incident energy, and we do not account for acquisition variables like exposure time. Without including terms for these parameters, our neural volume can only do so much to capture the complexity of the input datasets.

5.4.2 Alternative volume views

Our multi-energy volume provides us two ways to view our learned reconstruction. As we have seen, we can render slices with respect to a given incident energy, but we can also render slices using the learned z -value from which our energy-dependent attenuation coefficients are derived. Figure 5.16 shows the z -value slice for the interleaved multi-energy experiment. Though not shown, we have confirmed that the z -values do not change when we provide the neural volume with different incident energies. Notably, the z -values are inverted in comparison to the attenuation coefficients. This is one indication that we are not learning an uncalibrated atomic number, but rather a convenient shared structural representation across incident energies.

We also investigate what, if anything, our neural volume has captured for the incident energies which lie between those in our training set. There is generally

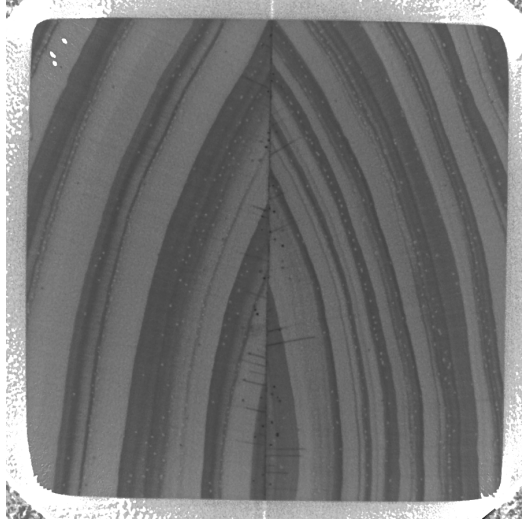


Figure 5.16: **z -value slice for the interleaved multi-energy experiment.** Despite the variations in observed attenuation coefficients across incident energy, the FlexAF multi-energy model captures the shared structure across all slices.

hope that the volume will interpolate reasonable attenuation coefficients between our trained energies. This is in keeping with the idea that the MLP learns a smooth function between its learned coordinates.

We densely render slices across the 35 kV to 120 kV range at an interval of 1 kV between each slice. Figure 5.17 shows the rendered 80 kV slice and plots the mean attenuation coefficients across all sampled energies. The slice, which is shown normalized to its own dynamic range, shares its structure with all the other slice images rendered from this volume. We can see in the coefficient plot that the attenuation function does smoothly transition between energies. However, the unsupervised regions of the X-ray spectrum are dramatically brighter than those over which we supervise. As such, these values do not appear to be meaningful interpolations between our trained energies.

5.4.3 Energy wedges

The reconstruction results for the energy wedges dataset are shown in Figure 5.18. These results are dramatically inferior to those from the interleaved dataset. While

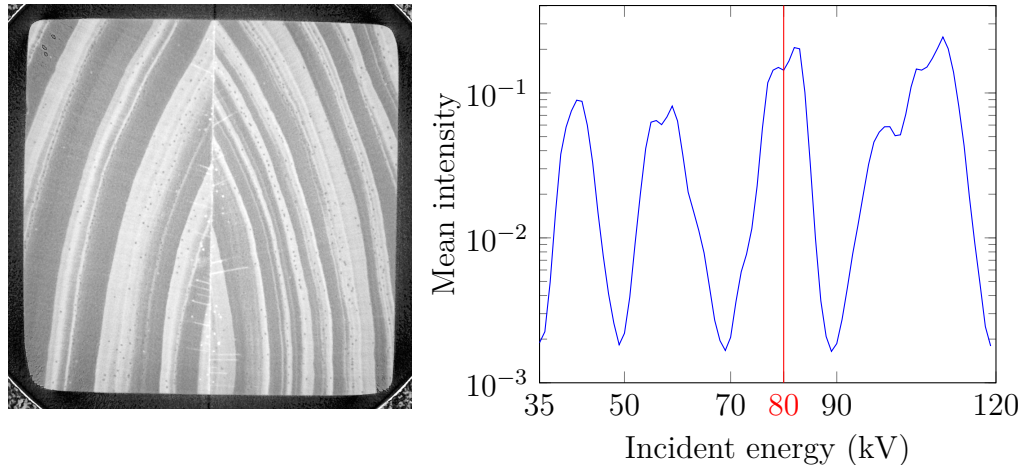


Figure 5.17: **Using the interleaved multi-energy model to interpolate between incident X-ray energies.** On the left, a slice rendered from the interleaved multi-energy model at an energy halfway between two of the training energies (80 kV). While the slice appears structurally reasonable, the plot on the right shows that the returned attenuation coefficients are dramatically out of range in comparison to those of the training energies.

FlexAF has clearly reconstructed the broad structure of the Multi* proxy, it has completely failed to learn the finely detailed features with any accuracy. Only the 35 kV slice looks reasonable, while all others suffer from significant errors. Looking at the z -value slice, we can see that this problem is not isolated to the attenuation coefficient outputs but is a feature of the volume’s learned structure. We do not know why this result should be so much worse than the interleaved dataset, but it is evident that there are significant differences between these two training methods that need to be understood.

5.5 On performance

As a final note, we briefly address a primary limitation for our method: the long training times required to converge to an accurate reconstruction. As we have noted, most of the reconstructions in this study have been for single slices at the center of the field of view, i.e. the slices which are the easiest to evaluate. Yet despite these relatively small reconstructions, our method often requires many hours, if not days,

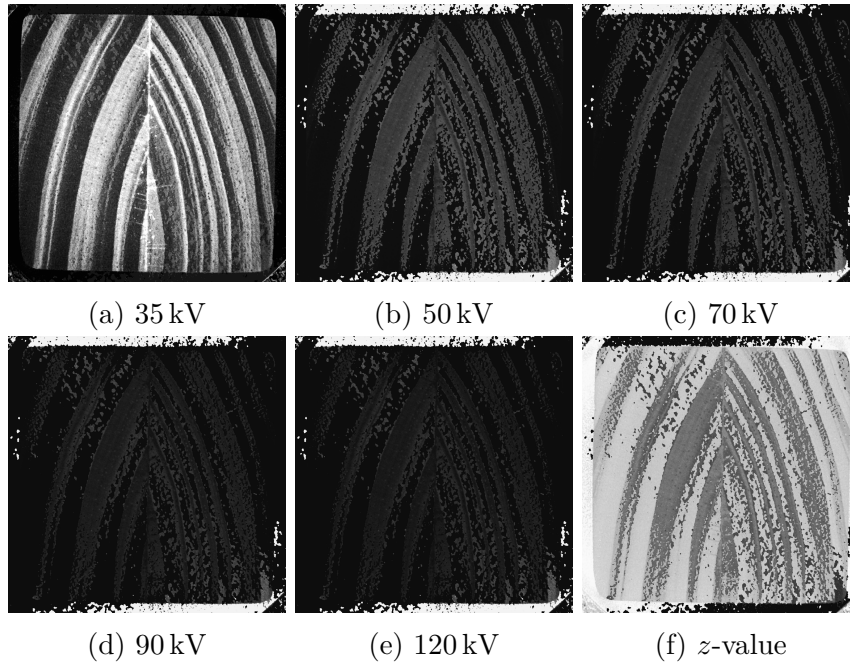


Figure 5.18: **Multi-energy reconstruction results using the energy wedges training method.** Slice images depicting the five training incident energies and the z -value slice. Though FlexAF appears to have converged on a reasonable reconstruction for the 35 kV slice, the slices for the other energies and z -values appear significantly degraded by comparison.

of GPU-accelerated computing time before it converges. Is a method such as ours even tractable in real-world applications?

The escapist answer to this question is to reference Moore’s Law and mumble something about quantum computing, the blockchain, and all things being possible as time approaches infinity. It is quite a different answer to say that this time may have already arrived. Before we consider potential optimizations to our method — and there are many *potential* optimizations — it is worth evaluating our current performance with respect to the acceleration hardware used in this study.

We consider our baseline experiment for MS.01.02, one of the largest full resolution datasets we attempt in this study. The evaluated size of our training set in this experiment was $1192 \times 2 \times 1200$, around 2.86 million pixels. Generously, this experiment converged to its final result after 250 training epochs. Training to 250 epochs took

GPU	F32	Convergence	TF32	Convergence
V100 (2017)	15.7 TFLOPS	2.5 years	–	–
A100 (2020)	19.5 TFLOPS	2.04 years	312 TFLOPS	46.5 days
H100 (2023)	67 TFLOPS	216.5 days	989 TFLOPS	14.67 days
B100 (2024)	–	–	14 PFLOPS	24.88 hours

Table 5.2: **Estimated time required to train the full MS.01.02 dataset to convergence using various Nvidia GPUs.** This table uses the single precision floating-point operations per second to estimate the time required to train the full MS.01.02 dataset to convergence. All experiments in this study were run on the Nvidia V100. Simply by switching to the recently announced B100 GPUs, FlexAF would potentially be able to reconstruct this entire dataset in a little more than 1 day of training time.

35.309 hours on an Nvidia Tesla V100 GPU, an amortized time-per-epoch of 8m30s. The full size of the MS.01.02 dataset is $1536 \times 972 \times 1200$,¹ around 1.79 billion pixels and a 626.25x increase in total dataset size. If we assume an identical model configuration with a runtime performance that increases linearly with dataset size, the estimated time-per-epoch for the full dataset would be 3.69 days, or 2.5 years to reach convergence at 250 epochs.

The Tesla V100 GPU was first released in June 2017, and needless to say, the ensuing 7 years have seen dramatic improvements to the runtime performance of GPU technologies. All operations in our framework are performed with single-precision floating point numbers, and thus we may roughly compare GPUs using their single-precision floating-point operations per second (FLOPS). We perform a simple estimation of training time to convergence for the full MS.01.02 dataset by comparing the FLOPS for the most recent three generations of Nvidia GPUs (Table 5.2). Starting with the A100 in 2020, Nvidia provides a *tensor float* type which significantly improves performance by automatically calculating many low precision operations with half-precision floats. Where available, we compare the FLOPS for both single-precision floats (F32) and tensor floats (TF32).

¹We again ignore the extra 1201st projection captured at the end of the full rotation.

As our table shows, the performance of GPU technology is increasing at almost exponential rates. At the time of this writing, the B100 has not yet been released, but the marketing materials claim an almost 1000x increase in performance over the V100 used in our study. We estimate a B100 reconstruction time for the full MS.01.02 dataset of 24.88 hours, 1.4x faster than the time we report for reconstructing a single slice. When we consider that this potential performance boost does not account for any optimizations we can make to the framework, we are extremely optimistic about the imminent applicability of FlexAF.

CHAPTER 6. DISCUSSION

“The discharge was in full force, and the rays were flying through my head, and, for all I knew, through the side of the box behind me. But they were invisible and impalpable. They gave no sensation whatever. Whatever the mysterious rays may be, they are not to be seen, and are to be judged only by their works.”

– *H.J.W. Dam, The New Marvel in Photography, McClure’s Magazine, 1896*

Before we conclude, we spend some time discussing the challenges and limitations that our method still faces and the immense opportunities that come with our increased capabilities.

6.1 The challenges of projective X-ray cameras

Our work is guided by the observation that X-ray imaging has a strong relationship to traditional photography, and that the projective camera models we use to understand the 3D structures of photographic spaces can be extended to tomographic applications. The decision to recast X-ray images as projective cameras presents a number of challenges which require further development and study.

First, today’s CT scanners do not directly record the position and orientation of the X-ray camera relative to some common world coordinate frame. This information is instead indirectly recorded in scan metadata as a pre-defined scan trajectory of known rotational step size, detector pixel sizes, source-to-sample-to-detector distances, etc. We construct the X-ray cameras we need on-the-fly using the metadata that has been made available, and we fill in any gaps with our understanding of how the scan was acquired. In the near term, using our framework across non-SkyScan datasets will require some effort parsing metadata formats, converting to our internal representation, etc. Going forward, we hope to see scanners which additionally provide per-projection extrinsic matrices similar to those found in many photogrammetry reconstruction frameworks.

Second, a limitation of our study is that we have not tested our framework on datasets with more interesting scan trajectories. Helical and spherical scanning paths are commonly employed across a wide range of industries and would provide an interesting challenge for our X-ray camera model and ray tracer. We see no fundamental reason why any well-posed scan trajectory should not work out-of-the-box with FlexAF.

6.2 Freely-defined trajectories

Throughout this text, we occasionally refer to the concept of freely-defined scan trajectories, by which we mean that reconstruction algorithms should accept X-ray cameras which are defined in arbitrary positions and orientations in the world coordinate frame. We do not mean that tomographic reconstruction is possible for arbitrary sets of X-ray projections, but rather that the algorithm should make a best-effort attempt at reconstruction using what information it has available. As with photogrammetry, the highest quality scan will likely follow the idealized rotational protocol. But also like photogrammetry, there are many use cases for tomography cannot be approached with traditional scanning hardware and well-established scan trajectories or which only require a best-effort reconstruction. These use cases have motivated our efforts to define X-ray cameras in FlexAF in a common world coordinate frame.

Chiefly, we desire a truly portable CT scanner much like the backpack model we briefly described in our introduction. Such a scanner would prove immensely valuable for medical CT applications in developing countries, particularly those places which do not possess the physical infrastructure required to transport large and delicate machinery. We are already beginning to see a shift towards low cost, low weight CT scanners within the medical industry [90], and we foresee that more flexible reconstruction algorithms will necessarily play a role in this development.

Likewise, there are many use cases within the sciences which would benefit from

a portable scanner. Recently, new virtual unwrapping technologies applied to CT volumes have provided a noninvasive means for recovering ancient and historical texts from inside badly damaged books, scrolls, manuscripts, and letters [3, 15, 67, 69, 87, 101]. As these materials are often extremely fragile and of a priceless nature, a chief difficulty in this work is transporting the materials from the collection to a laboratory environment so that a scan can take place. A more portable CT scanner provides an alternative to this paradigm where the object can stay safely in place and the scanner travels to the host institution.

6.3 Approximating ray integrals

The point-based ray sampling method we use in this study is simple and effective but ultimately at odds with the reality of X-ray imaging. If we wish to truly model an attenuation field across multiple volumetric scales, then we must be able to evaluate that model volumetrically as well. We believe that at least some of the frequency aliasing we experience in our multi-resolution experiments is due to all points being treated equally by the encoder and model, regardless of the spatial resolution of the original X-ray projection.

As discussed in 2.2.3, there are precedents for volumetric ray sampling for radiance fields. For example, Mip-NeRF approximates the conical frustums for each pixel as a discrete set of multivariate Gaussians. While we explored this approach during the development of FlexAF, our resulting reconstructions did not reproduce high-frequency features with the same accuracy as the point-based method. We believe that this may have been an implementation-specific issue and not a fundamental limitation of the Mip-NeRF method. This remains a promising avenue for future study.

As a practical concern, any discrete sampling method will eventually become challenging as the size of the modeled space grows with respect to a fixed resolution. For the sake of argument, we consider an 8 cm diameter sample which we wish to

reconstruct at a $1\ \mu\text{m}$ to $10\ \mu\text{m}$ resolution.¹ At these resolutions, the discrete grid for this sample easily approaches 10k pixels along a single slice axis. It seems unlikely that 512 ray intervals, the largest number of intervals we used in this study, would be of sufficient sampling density as to reconstruct the required details for such a scan.

An alternative method which may address this issue is not to sample on a per-ray basis at all. In our current sampling method, the areas near the center of rotation are sampled much more frequently than the areas near the edge of the reconstructable volume, a well-known byproduct of rotational geometries. As a result, many rays in the same batch are being independently sampled at nearly identical spatial locations. Removing this redundancy by evaluating rays with respect to a shared set of samples would significantly improve the volumetric coverage within each batch without requiring an increase to the total number of samples.

We noted in our Shepp-Logan experiments that the reconstructions show an intensity gradient which is brighter near the middle of the sample and darker at its edges. As both of these samples are centered in the field-of-view and contain relatively uniform internal structures, it is difficult to ascertain what exactly causes this gradient without more study. A likely cause is that the oversampled area near the center of the volume is likewise being overemphasized during our gradient updates. We believe that this represents an elusive bug in our implementation rather than a serious concern for the method at large.

6.4 Building a better model

Our volumetric model — the combination of the Gaussian encoding and the MLP — is crucial to the spatial accuracy of FlexAF. As we have seen, the current model operates well for modestly sized volumes and low-to-medium micro-CT resolutions, but has difficulty scaling to large volumes or high resolutions.

Many of the challenges which concern resolution can be traced to the Gaussian

¹Perhaps not so much a hypothetical as an actual challenge recently experienced by the author.

encoding we apply to our spatial coordinates. The encoding analyses by Tancik et al. and Zheng et al. (see 2.2.2) and our own experiences tuning the Gaussian encoder across multiple scales and resolutions strongly suggest a practical limit to the size and resolution which Gaussian encoding can support. While we were not able to measure a predictive relationship which consistently aided our hyperparameter selection, we did note an approximately inverse linear relationship between the sampled dimension sizes (i.e. the dimension’s size divided by the pixel size) and the Gaussian scale. For example, the MS.01.02 (70 μm) configuration presented in 5.1.3 maintains reconstruction quality by halving the size of the learnable volume from that used for MS.01.01 (140 μm) while keeping the Gaussian scale approximately the same.

A complicating factor in analyzing the Gaussian encoding’s effect on resolution is the tight relationship between the encoding and the MLP. As evidenced by our few multi-slice experiments, we frequently encountered encoder settings which produced high-quality, individual slices only to find that the quality deteriorated significantly as the size of the volume grew along the Z axis. We believe that this deterioration is purely a result of the MLP reaching its capacity limit and has very little to do with the Gaussian encoding. Further, it is clear that the MLP’s capacity is not fixed but is a function of the volume’s size, complexity, sparsity, and quality.

As we consider the next generation of neural volume models, it is obvious that we need to address the resolution and capacity questions that linger in FlexAF. Here we may look to the NeAT framework (see 2.3.3) for inspiration. It employs a dynamic hierarchical model built on differentiable features which purportedly adapts to the size and resolution of the reconstruction. Such a hierarchical structure would theoretically scale to scenes of arbitrary size, an important property as we consider the possibility of freely-defined scan trajectories.

6.5 Runtime performance

As we discussed in 5.5, the reconstruction times we report in this study are long in comparison to other CT reconstruction algorithms. Though improvements to computing hardware will continue to reduce the time spent in reconstruction, there are many opportunities for optimization which could dramatically improve the runtime performance of FlexAF.

We have already identified hierarchical space decompositions as pivotal for improving the quality and capacity of our neural volumes, but such decompositions could provide significant performance boosts as well. This benefit has already been shown for radiance fields, where a common insight among is that the deep, monolithic MLP is the dominant cost when sampling the neural volume. By using shallow MLPs, or removing them entirely in favor of alternative learned representations, one can achieve significant performance gains. This is perhaps best exemplified by Instant NGP [59], a highly optimized radiance field method which trains in seconds or minutes.

We additionally anticipate the importance of hierarchical decompositions in projection space. The entropy pixels method which we introduce in this study consistently reduces the number of training samples per epoch by 30-40% with only a slight (if not negligible) reduction in reconstruction quality. While these size reductions may appear modest, some experiments in this study would not have been feasible otherwise. Looking ahead, we see entropy pixels transitioning from a pre-calculated heuristic to one which dynamically proposes the most important training samples given the current state of the reconstruction. It seems obvious that the volume will not have learned enough of the reconstruction to make use of the full projection set until late in training, when we are most interested in refining high-frequency features in the reconstruction. Beyond entropy pixels, we foresee that adaptive sampling methods which intelligently allocate resources during training will be extremely important for reducing total training times.

6.6 Spectral tomography

The multi-energy volume we presented and tested in this work is largely a proof-of-concept meant to demonstrate the way in which a flexible framework can exploit dataset heterogeneity to great effect. Our existing polynomial model mapping structure and energy to attenuation is extremely simplistic and does not address practically any of the system variables which affect measured attenuation. However, we are extremely encouraged by our success modeling the shared structure across incident images, and we believe that this experiment proposes an exciting new approach for spectral tomography.

Notably, our method requires no changes to the scanning hardware and does not increase the total number of X-ray projections in the dataset. It is easy to imagine how such a method could be universally deployed into existing CT environments as a simple software upgrade. Future work should focus on developing a multi-energy model which more closely approximates the complexities of spectral X-ray attenuation, including terms for the energy distributions of the X-ray source, beam filters, exposure times, and scintillator and detector sensitivities.

6.7 Low-dosage, high-resolution reconstruction

Our multi-resolution ROI experiments in this study were motivated by a desire to decrease the scan times and X-ray dosage which are required to achieve to a high-resolution CT scan. As the pixel size of a scan gets smaller, the exposure times and number of rotational samples must increase to guarantee a high-quality reconstruction. Many existing reconstruction methods attempt to short circuit this exposure increase by optimizing reconstruction for a reduced number of rotational samples, the so-called *sparse* and *limited angle* reconstruction tasks.

We propose that similar improvements to X-ray dosage can be achieved by combining low-resolution scans with intentionally underexposed high-resolution data in a single FlexAF model. Our ROI experiments have already shown that FlexAF

naturally supports data captured across multiple scales in a single volume, and our multi-energy experiments demonstrate the ability to model shared structure across varying photometric settings. An obvious next step is to combine these features into a single model which also accounts for variance in exposure times. Such a modification will likely already be required in order to improve our multi-energy model, thus we see both mutli-resolution and multi-energy development continuing in tandem.

6.8 Unified volume models

As we discussed in our introduction, modern CT practices are characterized by the tendency to recapture entire scans when some perceived imperfection in the input dataset would produce reconstruction artifacts. Though the motivations of our work go well beyond error correction for CT scans, much of what we propose can in some ways alleviate those hard failure modes which lead to rescanning. By admitting projections which vary geometrically and photometrically, we enable at least partial, perhaps total, recovery from underexposed scans or unexpected sample movement.

However, full CT scans often do not represent the totality of X-ray images which are captured during a scan session. Numerous X-ray projections and test scans which are captured during scan setup are subsequently discarded because they “can’t be used” for reconstruction. Often, these images are perfectly acceptable projections in and of themselves but simply do not match the final scanning protocol. Armed with a reconstruction method which can account for dataset heterogeneity, we now have the means to use these images for more than just setup.

We imagine an online reconstruction process which begins *with the very first projection image*. At the heart of this process would be a unified attenuation field which is trained on every projection image captured during setup and which would provide instant feedback to the scan operator on the effects of their scan parameter selection. Optionally, this attenuation field could be used as the initial state for reconstructing the final scan, perhaps providing improvements to reconstruction times and/or

quality.

Further, we can extend this idea beyond a single scan session and into the space of multiple scans captured over long periods of time. We propose that attenuation fields could become “living models” which grow with time, and which integrate every new X-ray image into a single, unified reconstruction. Such a model could have important analytical advantages over discrete grids as it provides a simple method for combining all facets of radiography into a single frame of reference. By way of example, subtle changes to internal structure which are difficult to see directly in individual radiographs could be amplified by calculating the error between the radiograph and the existing unified volume.

6.9 Conclusion

In the preceding chapters, we have showed that neural reconstruction methods allow us to leave behind many of the limitations which have long governed computed tomography. Our data-centric reconstruction framework, FlexAF, adapts to and thrives on combinations of X-ray projection images which would produce significant errors in traditional reconstruction approaches. Our experiments produce high-quality reconstructions which are derived from standard, multi-resolution, and multi-energy projection image sets, sometimes in the face of extreme geometric misalignment. This flexibility is enabled by an X-ray camera model, differentiable ray tracer, and neural volume which implicitly model those complexities which would otherwise be challenging to formulate explicitly. We are aware of no other reconstruction method which unifies all of these concepts into a single framework, let alone a single volumetric model.

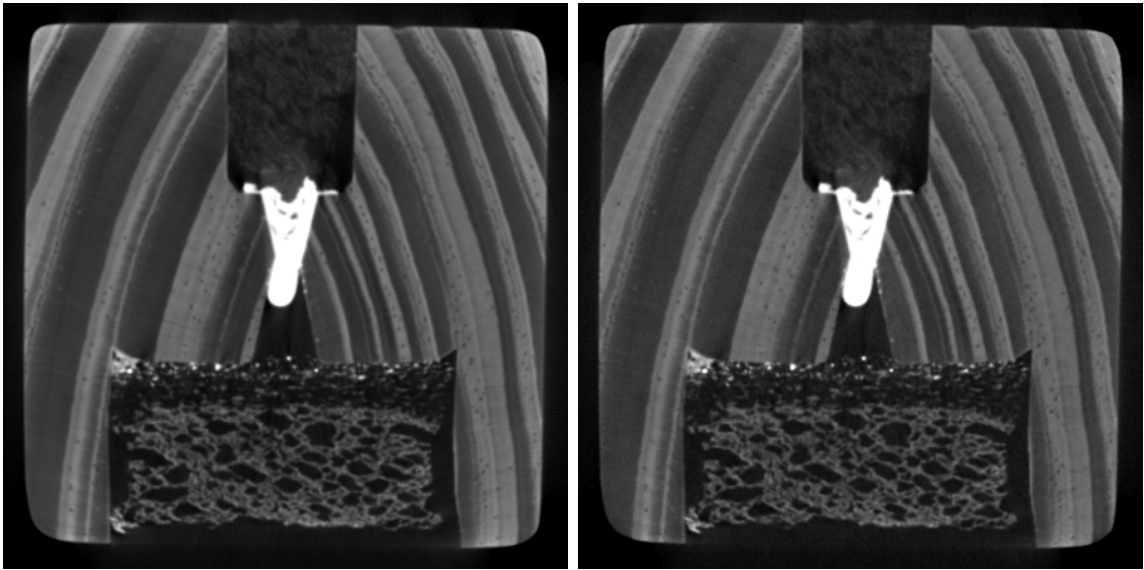
When H.J.W. Dam, the reporter from McClure’s Magazine, asked Wilhelm Röntgen whether he thought it would be possible to image the soft tissues of the body, Röntgen replied, “We shall see what we shall see. We have the start now; the developments will follow in time.” The opportunities presented by our framework are immense, but in

many ways, our work has only just begun. Whether through our methods described here, or through others, we believe that we are on the verge of a new, more flexible era for computed tomography. We have the start now, and it is difficult to predict what ideas will take hold, but the developments will follow in time.

“The most agreeable feature of the discovery is the opportunity it gives for other hands to help; and the work of these hands will add many new words to the dictionaries, many new facts to science, and, in the years long ahead of us, fill many more volumes than there are paragraphs in this brief and imperfect account.”

– *H.J.W. Dam, The New Marvel in Photography, McClure's Magazine, 1896*

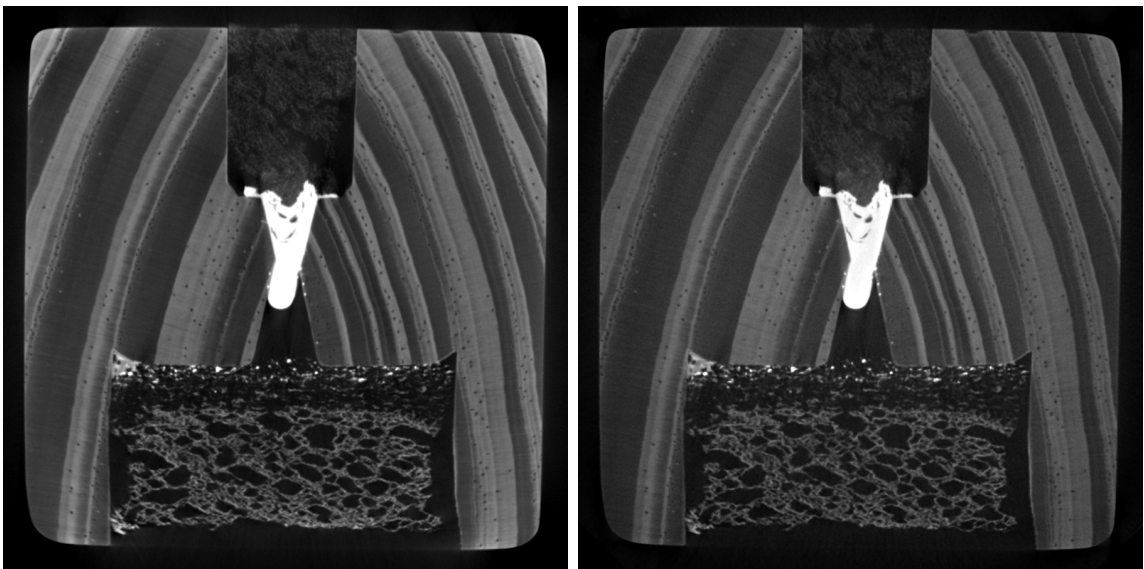
ENLARGED FIGURES



(a) FBP

(b) FlexAF

Figure A.1: **Comparison of FBP and FlexAF slices for the MS.01.01 reconstructions.** Enlarged version of the results depicted in Figure 5.5.



(a) FBP

(b) FlexAF

Figure A.2: **Comparison of FBP and FlexAF slices for the MS.01.02 reconstructions.** Enlarged version of the results depicted in Figure 5.7.

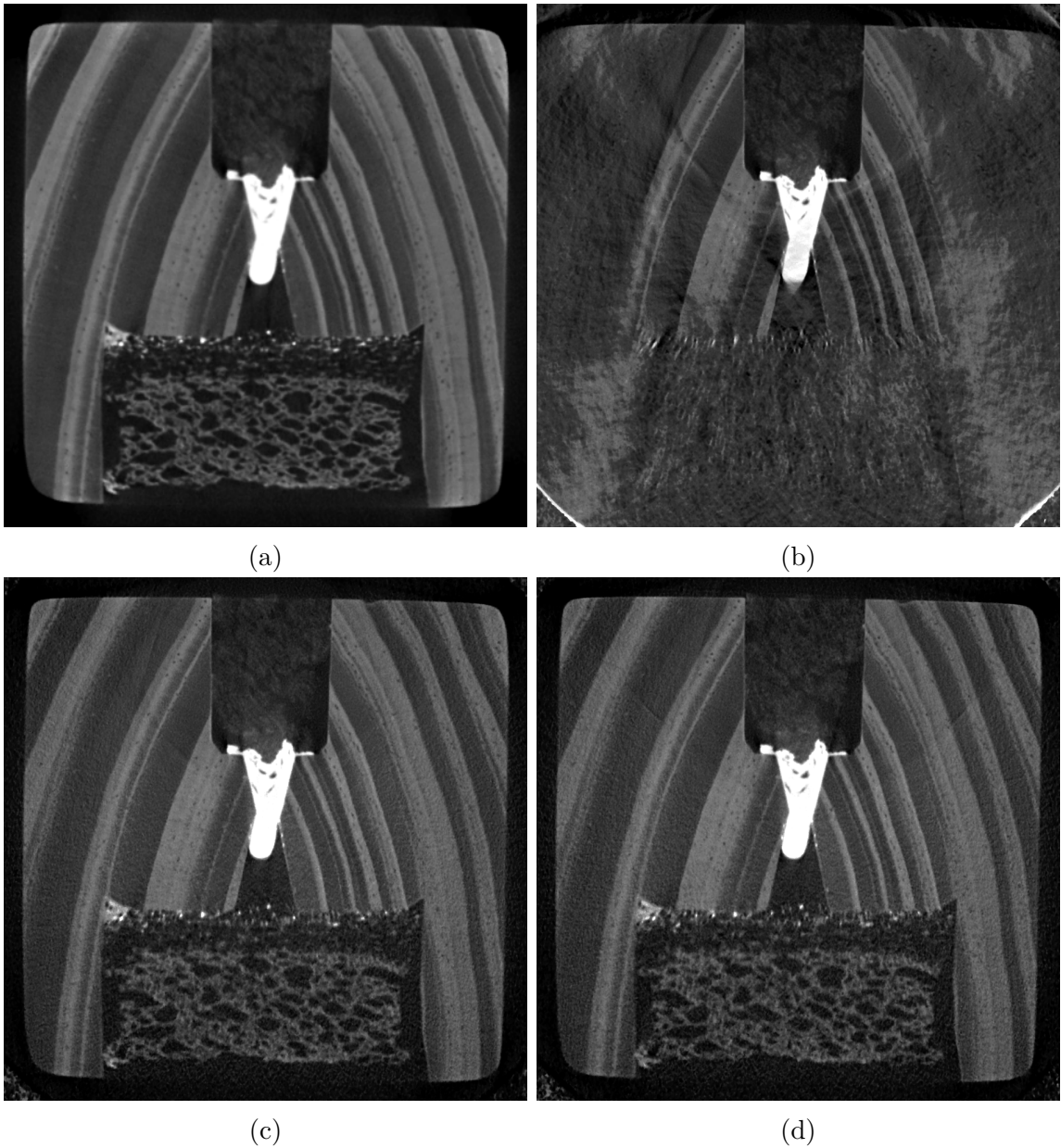


Figure A.3: **Comparison of the multi-resolution reconstructions.** Enlarged version of the results depicted in Figure 5.12. (a) Full FlexAF reconstruction of MS.01.01 ($140\mu\text{m}$). (b) ROI-only reconstruction. (c) ROI-enhancement reconstruction using projections from MS.01.01 and MS.01.02. (d) ROI-enhancement reconstruction using projections MS.01.01 and MS.01.04.

BIBLIOGRAPHY AND FURTHER READING

1. Ahishakiye, E. *et al.* A survey on deep learning in medical image reconstruction. *Intelligent Medicine* **1**, 118–127 (2021).
2. Barron, J. T. *et al.* Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. *ICCV* (2021).
3. Baum, D. *et al.* Revisiting the Jerash Silver Scroll: a new visual data analysis approach. *Digital Applications in Archaeology and Cultural Heritage*, e00186. ISSN: 2212-0548. <https://www.sciencedirect.com/science/article/pii/S2212054821000151> (2021).
4. Beer. Bestimmung der Absorption des rothen Lichts in farbigen Flüssigkeiten. *Annalen der Physik* **162**, 78–88 (1852).
5. Bouguer, P. *Essai d'optique sur la gradation de la lumière* (Claude Jombert, 1729).
6. Bracewell, R. N. Strip integration in radio astronomy. *Australian Journal of Physics* **9**, 198–217 (1956).
7. Brooks, R. A. & Di Chiro, G. Beam hardening in x-ray reconstructive tomography. *Physics in medicine & biology* **21**, 390 (1976).
8. Bureau, U. C. *RACE* U.S. Census Bureau. Accessed on 23 February 2024. <https://data.census.gov/table/DECENNIALPL2020.P1?g=010XX00US>.
9. Chang, M., Xiao, Y. & Chen, Z. Improve spatial resolution by Modeling Finite Focal Spot (MFFS) for industrial CT reconstruction. *Optics Express* **22**, 30641–30656 (2014).
10. Chapman, C. *et al.* *Using METS to Express Digital Provenance for Complex Digital Objects in Metadata and Semantic Research* (eds Garoufallou, E. & Ovalle-Perandones, M.-A.) (Springer International Publishing, Cham, Mar. 2021), 143–154. ISBN: 978-3-030-71903-6.
11. Chapman, C. Y. *et al.* The Digital Compilation and Restoration of Herculaneum Fragment P.Herc.118. *Manuscript Studies: A Journal of the Schoenberg Institute for Manuscript Studies* **6**, 1–32 (2021).
12. Chen, G.-H. *et al.* Development and evaluation of an exact fan-beam reconstruction algorithm using an equal weighting scheme via locally compensated filtered backprojection (LCFBP). *Medical Physics* **33**, 475–481 (2006).
13. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* **2**, 303–314 (1989).
14. Dam, H. J. The New Marvel in Photography. *McClure's Magazine* **6**, 403–415 (1896).

15. Dambrogio, J. *et al.* Unlocking history through automated virtual unfolding of sealed documents imaged by X-ray microtomography. *Nature Communications* **12**, 1184. ISSN: 2041-1723. <https://doi.org/10.1038/s41467-021-21326-w> (Mar. 2021).
16. De Chiffre, L. *et al.* Industrial applications of computed tomography. *CIRP Annals* **63**, 655–677. ISSN: 0007-8506. <https://www.sciencedirect.com/science/article/pii/S0007850614001930> (2014).
17. Deng, K. *et al.* *Depth-supervised NeRF: Fewer Views and Faster Training for Free* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022).
18. Dilley, P. C. *et al.* The X-Ray Micro-CT of a Full Parchment Codex to Recover Hidden Text: Morgan Library M.910, an Early Coptic Acts of the Apostles Manuscript. *Manuscript Studies: A Journal of the Schoenberg Institute for Manuscript Studies* **7**, 162–174 (2022).
19. Feldkamp, L. A., Davis, L. C. & Kress, J. W. Practical cone-beam algorithm. *Josa a* **1**, 612–619 (1984).
20. Fessler, J. A. Penalized weighted least-squares image reconstruction for positron emission tomography. *IEEE transactions on medical imaging* **13**, 290–300 (1994).
21. Finkel, R. A. & Bentley, J. L. Quad trees a data structure for retrieval on composite keys. *Acta informatica* **4**, 1–9 (1974).
22. Furukawa, Y., Hernández, C., *et al.* Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision* **9**, 1–148 (2015).
23. Ganio, M. *et al.* *Unbending light: new computational methods for the correction of 3D effects in scanning XRF* in *Optics for Arts, Architecture, and Archaeology VII Conference Proceedings of SPIE Volume 11058* (2019). <https://doi.org/10.1117/12.2525038>.
24. Garbin, S. J. *et al.* *Fastnerf: High-fidelity neural rendering at 200fps* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 14346–14355.
25. Gessel, K. *et al.* *Towards Automating Volumetric Segmentation for Virtual Unwrapping* in *Proceedings of the 25th International Conference on Cultural Heritage and New Technologies 2020*. (eds Börner, W. *et al.*) (Nov. 2020).
26. Geyer, L. L. *et al.* State of the art: iterative CT reconstruction techniques. *Radiology* **276**, 339–357 (2015).
27. Google Scholar. *Entry for Nerf: Representing scenes as neural radiance fields* Online. Accessed on 2 March 2024. <https://scholar.google.com/scholar?q=NeRF+Representing+scenes+as+neural+radiance+fields>.
28. Gordon, R., Bender, R. & Herman, G. T. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *Journal of theoretical Biology* **29**, 471–481 (1970).

29. Hartley, R. I. & Zisserman, A. in. *Multiple View Geometry in Computer Vision* 2nd ed., 153–177 (Cambridge University Press, 2004). ISBN: 0521540518.
30. Hartley, R. I. & Zisserman, A. in. *Multiple View Geometry in Computer Vision* 2nd ed., 262–278 (Cambridge University Press, 2004). ISBN: 0521540518.
31. Hartley, R. I. & Zisserman, A. *Multiple View Geometry in Computer Vision* 2nd ed. ISBN: 0521540518 (Cambridge University Press, 2004).
32. Hedman, P. *et al.* Baking Neural Radiance Fields for Real-Time View Synthesis. *ICCV* (2021).
33. Horn, B. K. Fan-beam reconstruction methods. *Proceedings of the IEEE* **67**, 1616–1623 (1979).
34. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural networks* **2**, 359–366 (1989).
35. Hounsfield, G. N. Computerized transverse axial scanning (tomography): Part 1. Description of system. *The British journal of radiology* **46**, 1016–1022 (1973).
36. Hsieh, J. eng. in. *Computed tomography : principles, design, artifacts, and recent advances* 3rd ed., 38–45 (SPIE Press, Bellingham, Washington, 2015). ISBN: 9781628416640.
37. Hsieh, J. eng. in. *Computed tomography : principles, design, artifacts, and recent advances* 3rd ed., 63–90 (SPIE Press, Bellingham, Washington, 2015). ISBN: 9781628416640.
38. Hsieh, J. eng. in. *Computed tomography : principles, design, artifacts, and recent advances* 3rd ed., 104–123 (SPIE Press, Bellingham, Washington, 2015). ISBN: 9781628416640.
39. Hsieh, J. *Computed tomography : principles, design, artifacts, and recent advances* 3rd ed. eng. ISBN: 9781628416640 (SPIE Press, Bellingham, Washington, 2015).
40. Hubbell, J. H. & Seltzer, S. M. *Tables of X-ray mass attenuation coefficients and mass energy-absorption coefficients 1 keV to 20 MeV for elements Z = 1 to 92 and 48 additional substances of dosimetric interest* tech. rep. (National Inst. of Standards and Technology, 1995).
41. Kaczmarz, S. Angenaherte auflosung von systemen linearer glei-chungen. *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.*, 355–357 (1937).
42. Kaji, S. & Kida, S. Overview of image-to-image translation by use of deep neural networks: denoising, super-resolution, modality conversion, and reconstruction in medical imaging. *Radiological physics and technology* **12**, 235–248 (2019).
43. Katsevich, A. Analysis of an exact inversion algorithm for spiral cone-beam CT. *Physics in Medicine & Biology* **47**, 2583 (2002).
44. Katsevich, A. An improved exact filtered backprojection algorithm for spiral computed tomography. *Advances in Applied Mathematics* **32**, 681–697 (2004).

45. Kerbl, B. *et al.* 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* **42**. <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/> (July 2023).
46. Konidakis, G., Osentoski, S. & Thomas, P. *Value function approximation in reinforcement learning using the Fourier basis* in *Proceedings of the AAAI Conference on Artificial Intelligence* **25** (2011), 380–385.
47. Koo, J. *et al.* *A Tomographic Reconstruction Method using Coordinate-based Neural Network with Spatial Regularization* in *Proceedings of the Northern Lights Deep Learning Workshop* **2** (2021).
48. Lambert, J. H. *Photometria sive de mensura et gradibus luminis, colorum et umbrae* (sumptibus viduae E. Klett, typis CP Detleffsen, 1760).
49. Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics* **2**, 164–168 (1944).
50. Lin, C.-H. *et al.* *BARF: Bundle-Adjusting Neural Radiance Fields* in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2021), 5741–5751.
51. Lin, H. *et al.* *Efficient Neural Radiance Fields for Interactive Free-viewpoint Video* in *SIGGRAPH Asia Conference Proceedings* (2022).
52. Liu, L. Model-based iterative reconstruction: a promising algorithm for today’s computed tomography imaging. *Journal of Medical imaging and Radiation sciences* **45**, 131–136 (2014).
53. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* **60**, 91–110 (2004).
54. Markel, H. ‘I Have Seen My Death’: How the World Discovered the X-Ray. <https://www.pbs.org/newshour/health/i-have-seen-my-death-how-the-world-discovered-the-x-ray> (2024) (2012).
55. Marquardt, D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics* **11**, 431–441. <https://doi.org/10.1137/01111030> (1963).
56. McKibben, N., Kärkkäinen, L. & GPH. *Phantominator. A Python package for easy generation of numerical phantoms*. comp. software. Accessed on 20 March 2024. <https://github.com/mckib2/phantominator>.
57. Mildenhall, B. *et al.* *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis* in *ECCV* (2020).
58. Mildenhall, B. *et al.* *NeRF in the Dark: High Dynamic Range View Synthesis from Noisy Raw Images*. *CVPR* (2022).
59. Müller, T. *et al.* Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* **41**, 1–15 (2022).
60. Noo, F., Pack, J. & Heuscher, D. Exact helical reconstruction using native cone-beam geometries. *Physics in Medicine & Biology* **48**, 3787 (2003).

61. OECD. *Diagnostic technologies in Health at a Glance 2023: OECD Indicators* 116–117 (OECD Publishing, 2023). <https://doi.org/10.1787/857d9cb0-en>.
62. Özyeşil, O. *et al.* A survey of structure from motion. *Acta Numerica* **26**, 305–364 (2017).
63. Parker, C. S. & Seales, W. B. *Enhanced CT Analysis Using Volume Flattening* in *Bruker Micro-CT User Meeting Abstract Book* (Brussels, Belgium, June 2017), 15–16.
64. Parker, C. S., Seales, W. B. & Heyworth, G. *Reading the Invisible Library* in *Bruker Micro-CT User Meeting Abstract Book* (Mondorf-les-Bains, Luxembourg, May 2016), 58–59.
65. Parker, C. S. *et al.* *Volume Cartographer. A cross-platform C++ library and toolkit for the recovery and restoration of damaged cultural artifacts* comp. software. Mar. 2021. <https://doi.org/10.5281/zenodo.4604881>.
66. Parker, C. S., Seales, W. B. & Shor, P. *Quantitative Distortion Analysis of Flattening Applied to the Scroll from En-Gedi* in *Art & Archaeology, 2nd International Conference* (2016). arXiv: 2007.15551 [cs.CV].
67. Parker, C. S. *et al.* From invisibility to readability: Recovering the ink of Herculaneum. *PLOS ONE* **14**, 1–17. <https://doi.org/10.1371/journal.pone.0215775> (May 2019).
68. Parker, D. L. Optimal short scan convolution reconstruction for fan beam CT. *Medical physics* **9**, 254–257 (1982).
69. Parsons, S. *Hard-Hearted Scrolls: A Noninvasive Method for Reading the Herculaneum Papyri* PhD thesis (University of Kentucky, 2023). <https://doi.org/10.13023/etd.2023.372>.
70. Parsons, S., Parker, C. S. & Seales, W. B. The St. Chad Gospels: Diachronic Manuscript Registration and Visualization. *Manuscript Studies: A Journal of the Schoenberg Institute for Manuscript Studies* **2**, 483–498 (2017).
71. Parsons, S. *et al.* *Revealing “Invisible” Signals in CT with Machine Learning* in *Bruker Micro-CT User Meeting Abstract Book* (Mechelen, Belgium, June 2019), 20–22.
72. Parsons, S. *et al.* *Deep Learning for More Expressive Virtual Unwrapping* in *Proceedings of the 25th International Conference on Cultural Heritage and New Technologies 2020*. (eds Börner, W. *et al.*) (Nov. 2020), 203–207. <https://doi.org/10.11588/propylaeum.1045.c14501>.
73. Parsons, S. *et al.* *Machine Learning Infrastructure on the Frontier of Virtual Unwrapping* in *Proceedings of International Symposium on Grids & Clouds 2021 (ISCG2021)* (Proceedings of Science, Academia Sinica Computing Centre (ASGC), Taipei, Taiwan (Online), Mar. 2021), 15.
74. Parsons, S. *et al.* *Educelab-Scrolls: Verifiable Recovery of Text from Herculaneum Papyri using X-ray CT* 2023. arXiv: 2304.02084 [cs.CV].

75. Pharr, M., Jakob, W. & Humphreys, G. *Physically based rendering: From theory to implementation* (MIT Press, 2023).
76. Radon, J. On the determination of functions from their integral values along certain manifolds. *IEEE Transactions on Medical Imaging* **5**, 170–176 (1986).
77. Rahaman, N. *et al.* On the spectral bias of neural networks in *International Conference on Machine Learning* (2019), 5301–5310.
78. Rahimi, A. & Recht, B. Random features for large-scale kernel machines. *Advances in neural information processing systems* **20** (2007).
79. Reiser, C. *et al.* Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 14335–14345.
80. Rolff, T. *et al.* Interactive VRS-NeRF: Lightning fast Neural Radiance Field Rendering for Virtual Reality in *Proceedings of the 2023 ACM Symposium on Spatial User Interaction* (2023), 1–3.
81. Rolff, T. *et al.* VRS-NeRF: Accelerating Neural Radiance Field Rendering with Variable Rate Shading in *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2023), 243–252.
82. Röntgen, W. C. The bones of a hand with a ring on one finger, viewed through x-ray. Photoprint from radiograph by W.K. Röntgen, 1895. *Wellcome Collection*. This work is licensed under the Creative Commons Attribution NonCommercial 4.0 International License (CC BY-NC 4.0). To view a copy of this license, visit: <https://creativecommons.org/licenses/by-nc/4.0>.
83. Rückert, D. *et al.* Neat: Neural adaptive tomography. *ACM Transactions on Graphics (TOG)* **41**, 1–13 (2022).
84. Sasov, A., Liu, X. & Salmon, P. L. Compensation of mechanical inaccuracies in micro-CT and nano-CT in *Developments in X-ray Tomography VI* **7078** (2008), 401–409.
85. Sauer, K. & Bouman, C. A local update strategy for iterative reconstruction from projections. *IEEE Transactions on Signal Processing* **41**, 534–548 (1993).
86. Seales, W. B., Parker, C. S. & Chapman, C. 4.1.1.7 Virtual Unwrapping: A Computational Approach for Reading Damaged Manuscripts in *Textual History of the Bible* (ed Lange, A.) chap. 1.1.7 (2017). http://dx.doi.org/10.1163/2452-4107_thb_COM_225869.
87. Seales, W. B. *et al.* From damage to discovery via virtual unwrapping: Reading the scroll from En-Gedi. *Science Advances* **2**. eprint: <http://advances.sciencemag.org/content/2/9/e1601247.full.pdf>. <http://advances.sciencemag.org/content/2/9/e1601247> (2016).
88. Segal, M. *et al.* An Early Leviticus Scroll from En-Gedi: Preliminary Publication. *Textus* **26**, 29–58 (2016).

89. Shepp, L. A. & Logan, B. F. The Fourier reconstruction of a head section. *IEEE Transactions on nuclear science* **21**, 21–43 (1974).
90. *Solving a global thrive for medical imaging* tech. rep. Accessed on 1 April 2024 (Nano X Imaging Ltd., 2021). <https://www.nanox.vision/publications/white-papers>.
91. Sun, Y. *et al.* CoIL: Coordinate-Based Internal Learning for Tomographic Imaging. *IEEE Transactions on Computational Imaging* **7**, 1400–1412 (2021).
92. Tancik, M. *et al.* Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. *NeurIPS* (2020).
93. Tancik, M. *et al.* Block-NeRF: Scalable Large Scene Neural View Synthesis. *arXiv* (2022).
94. Thibault, J.-B. *et al.* A three-dimensional statistical approach to improved image quality for multislice helical CT. *Medical physics* **34**, 4526–4544 (2007).
95. Thies, M. *et al.* Gradient-based geometry learning for fan-beam CT reconstruction. *Physics in Medicine & Biology* **68**, 205004 (2023).
96. Tilley II, S., Siewerdsen, J. H. & Stayman, J. W. Model-based iterative reconstruction for flat-panel cone-beam CT with focal spot blur, detector blur, and correlated noise. *Physics in Medicine & Biology* **61**, 296–319 (2016).
97. Tilley II, S. *et al.* Nonlinear statistical reconstruction for flat-panel cone-beam CT with blur and correlated noise models in *Medical Imaging 2016: Physics of Medical Imaging* **9783** (2016), 97830R.
98. Wang, G., Ye, J. C. & De Man, B. Deep learning for tomographic image reconstruction. *Nature Machine Intelligence* **2**, 737–748 (2020).
99. Wang, Q. *et al.* *IBRNet: Learning Multi-View Image-Based Rendering* in *CVPR* (2021).
100. Wang, T. *et al.* A Review of Deep Learning CT Reconstruction From Incomplete Projection Data. *IEEE Transactions on Radiation and Plasma Medical Sciences* **8**, 138–152 (2024).
101. Wilster-Hansen, B. *et al.* Virtual unwrapping of the BISPEGATA amulet, a multiple folded medieval lead amulet, by using neutron tomography. *Archaeometry*. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/arcm.12734>. <https://onlinelibrary.wiley.com/doi/abs/10.1111/arcm.12734> (Nov. 2021).
102. Yu, A. *et al.* *PlenOctrees for Real-time Rendering of Neural Radiance Fields* in *ICCV* (2021).
103. Yu, A. *et al.* Plenoxels: Radiance Fields without Neural Networks. *arXiv preprint arXiv:2112.05131* (2021).
104. Yu, Z. *et al.* Fast model-based X-ray CT reconstruction using spatially non-homogeneous ICD optimization. *IEEE Transactions on image processing* **20**, 161–175 (2010).

105. Zang, G. *et al.* *IntraTomo: Self-Supervised Learning-Based Tomography via Sinogram Synthesis and Prediction* in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2021), 1960–1970.
106. Zhang, M., Gu, S. & Shi, Y. The use of deep learning methods in low-dose computed tomography image reconstruction: a systematic review. *Complex & intelligent systems* **8**, 5545–5561 (2022).
107. Zhao, S.-R. & Halling, H. *A new Fourier method for fan beam reconstruction* in *1995 IEEE Nuclear Science Symposium and Medical Imaging Conference Record* **2** (1995), 1287–1291 vol.2.
108. Zheng, J. *et al.* Trading Positional Complexity vs. Deepness in Coordinate Networks. *Proceedings of the European Conference on Computer Vision (ECCV)* (2022).

VITA

C. Seth Parker

Education

- B.A. in Media and Communications from Asbury University. May, 2010.

Professional positions held

- 2011–2018: Video Production Coordinator, Center for Visualization and Virtual Environments, University of Kentucky, Lexington, KY.
- 2014–present: Researcher and Project Manager, Department of Computer Science, University of Kentucky, Lexington, KY.

Scholastic and professional honors

- Outstanding Student Paper Award, “From invisibility to readability: Recovering the ink of Herculaneum”, University of Kentucky, Department of Computer Science, April 2021

Publications

- Parsons, S., Parker, C. S., Chapman, C., Hayashida, M. & Seales, W. B. *EduceLab-Scrolls: Verifiable Recovery of Text from Herculaneum Papyri using X-ray CT* 2023. arXiv: 2304.02084 [cs.CV]
- Dilley, P. C., Chapman, C., Parker, C. S. & Seales, W. B. The X-Ray Micro-CT of a Full Parchment Codex to Recover Hidden Text: Morgan Library M.910, an Early Coptic Acts of the Apostles Manuscript. *Manuscript Studies: A Journal of the Schoenberg Institute for Manuscript Studies* **7**, 162–174 (2022)
- Parsons, S., Chappell, J., Parker, C. S. & Seales, W. B. *Machine Learning Infrastructure on the Frontier of Virtual Unwrapping* in *Proceedings of International Symposium on Grids & Clouds 2021 (ISCG2021)* (Proceedings of Science, Academia Sinica Computing Centre (ASGC), Taipei, Taiwan (Online), Mar. 2021), 15
- Chapman, C. Y., Parker, C. S., Bertelsman, A., Gessel, K., Hatch, H., Seevers, K., Brusuelas, J. H., Parsons, S. & Seales, W. B. The Digital Compilation and Restoration of Herculaneum Fragment P.Herc.118. *Manuscript Studies: A Journal of the Schoenberg Institute for Manuscript Studies* **6**, 1–32 (2021)
- Chapman, C., Parker, S., Parsons, S. & Seales, W. B. *Using METS to Express Digital Provenance for Complex Digital Objects* in *Metadata and Semantic Research* (eds Garoufallou, E. & Ovalle-Perandones, M.-A.) (Springer International Publishing, Cham, Mar. 2021), 143–154. ISBN: 978-3-030-71903-6

- Gessel, K., Parsons, S., Parker, C. & Seales, W. *Towards Automating Volumetric Segmentation for Virtual Unwrapping* in *Proceedings of the 25th International Conference on Cultural Heritage and New Technologies 2020*. (eds Börner, W., Rohland, H., Kral-Börner, C. & Karner, L.) (Nov. 2020)
- Parsons, S., Gessel, K., Parker, C. & Seales, W. *Deep Learning for More Expressive Virtual Unwrapping* in *Proceedings of the 25th International Conference on Cultural Heritage and New Technologies 2020*. (eds Börner, W., Rohland, H., Kral-Börner, C. & Karner, L.) (Nov. 2020), 203–207. <https://doi.org/10.11588/propylaeum.1045.c14501>
- Ganio, M., Parsons, S., Parker, S., Svoboda, M., Seales, B. & Patterson, C. S. *Unbending light: new computational methods for the correction of 3D effects in scanning XRF* in *Optics for Arts, Architecture, and Archaeology VII* Conference Proceedings of SPIE Volume 11058 (2019). <https://doi.org/10.1117/12.2525038>
- Parsons, S., Parker, C. S., Coppens, F. & Seales, W. B. *Revealing “Invisible” Signals in CT with Machine Learning* in *Bruker Micro-CT User Meeting Abstract Book* (Mechelen, Belgium, June 2019), 20–22
- Parker, C. S., Parsons, S., Bandy, J., Chapman, C., Coppens, F. & Seales, W. B. From invisibility to readability: Recovering the ink of Herculaneum. *PLOS ONE* **14**, 1–17. <https://doi.org/10.1371/journal.pone.0215775> (May 2019)
- Parker, C. S. & Seales, W. B. *Enhanced CT Analysis Using Volume Flattening* in *Bruker Micro-CT User Meeting Abstract Book* (Brussels, Belgium, June 2017), 15–16
- Seales, W. B., Parker, C. S. & Chapman, C. *4.1.1.7 Virtual Unwrapping: A Computational Approach for Reading Damaged Manuscripts* in *Textual History of the Bible* (ed Lange, A.) chap. 1.1.7 (2017). http://dx.doi.org/10.1163/2452-4107_thb_COM_225869
- Parsons, S., Parker, C. S. & Seales, W. B. The St. Chad Gospels: Diachronic Manuscript Registration and Visualization. *Manuscript Studies: A Journal of the Schoenberg Institute for Manuscript Studies* **2**, 483–498 (2017)
- Parker, C. S., Seales, W. B. & Shor, P. *Quantitative Distortion Analysis of Flattening Applied to the Scroll from En-Gedi* in *Art & Archaeology, 2nd International Conference* (2016). arXiv: 2007.15551 [cs.CV]
- Parker, C. S., Seales, W. B. & Heyworth, G. *Reading the Invisible Library* in *Bruker Micro-CT User Meeting Abstract Book* (Mondorf-les-Bains, Luxembourg, May 2016), 58–59

- Segal, M., Tov, E., Seales, W. B., Parker, C. S., Shor, P. & Porath, Y. An Early Leviticus Scroll from En-Gedi: Preliminary Publication. *Textus* **26**, 29–58 (2016)
- Seales, W. B., Parker, C. S., Segal, M., Tov, E., Shor, P. & Porath, Y. From damage to discovery via virtual unwrapping: Reading the scroll from En-Gedi. *Science Advances* **2**. eprint: <http://advances.sciencemag.org/content/2/9/e1601247.full.pdf>. <http://advances.sciencemag.org/content/2/9/e1601247> (2016)